ROBERT H. BISHOP

EBOOKBKMT.COM
Tài liệu kỹ thuật miễn phí

THE MECHATRONICS HANDBOOK

CƠ ĐIỆN TỬ

Tập 2

Biên dịch: Phạm Anh Tuấn

ĐHQG HÀ NỘI
NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

EBOOKBKMT.COM
Tài liệu kỹ thuật miễn phí

CƠ ĐIỆN TỬ

Người dịch:

Vũ Lê Bình	Nguyễn Cao Mệnh
Phùng Văn Hiệp	Phạm Văn Bạch Ngọc
Vũ Việt Hùng	Đỗ Thị Ngọc Oanh
Đinh Công Huân	Phan Mạnh Quân
Lê Xuân Huy	Nguyễn Trường Thanh
Vũ Minh Hùng	Dỗ Trần Thắng
Vương Thị Diệu Hương	Nguyễn Huy Thụy
Tạ Hồng Kỳ	Phạm Anh Tuấn

Robert H. Bishop

EBOOKBKMT.COM
Tài liệu kỹ thuật miễn phí

CƠ ĐIỆN TỬ

(Gồm hai tập)

Tập 2

Biên dịch: **Phạm Anh Tu ấn**

Nhà xuất bản Đại học Quốc gia Hà Nội

EBOOKBKMT.COM
Tài liệu kỹ thuật miễn phí

Lời giới thiệu

Cơ điện tử là một lĩnh vực khoa học công nghệ liên ngành, tích hợp trong mỗi sản phẩm của nó các thành tựu mới nhất của cơ khí chính xác, điện tử, điều khiển, công nghệ thông tin và các vật liệu mới. Với đặc điểm nổi bật này, vai trò của Cơ điện tử trở nên rất quan trọng và không thể thiếu trong sự phát triển của khoa học kỹ thuật hiện đại. Tạp chí "Technology Review" của MIT tháng 02/2003 đã đánh giá Mechatronics (Cơ điện tử) là một trong 10 công nghệ có triển vọng làm thay đổi thế giới trong thế kỷ 21.

Cơ điện tử đã được quan tâm phát triển và có những bước tiến nhất định ở Việt Nam. Ngày 31/12/2003, Thủ tướng Chính phủ đã ký quyết định phê duyệt "Chiến lược phát triển Khoa học và Công nghệ Việt Nam đến 2010" trong đó Cơ điện tử là một trong những hướng công nghệ trọng điểm phục vụ phát triển kinh tế xã hội.

Trong những năm qua, với tiềm lực của một trung tâm nghiên cứu KHCN liên ngành mạnh, Viện Khoa học và Công nghệ Việt Nam đã đạt được các thành tích đáng kể về trong lĩnh vực Cơ điện tử. Quan hệ hợp tác nghiên cứu và triển khai của Viện trong lĩnh vực Cơ điện tử cũng không ngừng được tăng cường. Giáo sư Robert H. Bishop, một chuyên gia hàng đầu trong lĩnh vực Cơ điện tử, là người có nhiều đóng góp tích cực trong việc thúc đẩy mối quan hệ hợp tác nghiên cứu và đào tạo giữa Viện Khoa học và Công nghệ Việt Nam và Trường Đại học Texas tại Austin (Mỹ). Ông đã giành được nhiều giải thưởng khoa học danh giá, đồng thời là chủ biên, tác giả của nhiều cuốn sách chuyên ngành có uy tín.

Cuốn *The Mechatronics Handbook* do Giáo sư Robert H. Bishop chủ biên, được biên soạn bởi tập thể 88 nhà khoa học xuất sắc đến từ 13 nước trên thế giới - không chỉ giới thiệu một cách toàn diện những kiến thức từ tổng hợp đến chuyên sâu về Cơ điện tử mà còn là một tài liệu hữu ích khi cần tra cứu các khái niệm kỹ thuật chuyên ngành hoặc sử dụng các kiến thức này vào các mục đích nghiên cứu hoặc triển khai ứng dụng.

Với mục đích giúp các nhà nghiên cứu, các kỹ sư, các sinh viên đại học và sau đại học trong nước có điều kiện tiếp cận với những kiến thức Cơ điện tử mới nhất trên thế giới một cách thuận lợi và hiệu quả, tập thể cán bộ phòng Cơ điện tử, Viện Cơ học, Viện Khoa học và Công nghệ Việt Nam dưới sự hỗ trợ chuyên môn của nhiều nhà khoa học, nhà giáo có uy tín, đã biên dịch toàn bộ nội dung cuốn *The Mechatronics Handbook* thành cuốn sách *Cơ điện tử* tiếng Việt. Cuốn sách được chia thành hai tập có nội dung sắp xếp khoa học theo chủ đề với 50 chương. Chủ tịch Viện Khoa học và Công nghệ Việt Nam đã dành sự quan tâm và hỗ trợ tài chính quý báu cho công việc rất có ý nghĩa này.

Tôi tin tưởng rằng cuốn *Cơ điện tử* sẽ là một tài liệu tham khảo thật sự bổ ích với các bạn đọc Việt Nam.

GS. TSKH. Nguyễn Khoa Sơn

Phó Chủ tịch Viện Khoa học và Công nghệ Việt Nam

AEROSPACE ENGINEERING & ENGINEERING MECHANICS
THE UNIVERSITY OF TEXAS AT AUSTIN

210 E. 24th Street, WRW · Tel: 512-471-7593 · Fax 512-471-3788
1 University Station C0600 · Austin, Texas 78712-0235

To my friends in Vietnam –

The Mechatronic Handbook represents the best efforts of researchers world-wide to describe the state-of-the-art in mechatronic design. I hope that students and faculty with a thirst for knowledge and a desire to expand their horizons will find this text valuable.

I wish to extend my congratulations to Dr. Pham Anh Tuan and his colleagues for succesfully completing this Vietnamese translation of the Mechatronics Handbook. I am thrilled that students in Vietnam will have access to this book in their native language.

My best wishes to all readers!

With warm regards,
Robert H. Bishop
August 28, 2006

Thư gửi độc giả Việt Nam

Các bạn Việt Nam thân mến!

Cuốn *The Mechatronics Handbook* là kết quả những nỗ lực làm việc hết mình của các nhà nghiên cứu trên thế giới nhằm mang đến cho người đọc những kiến thức mới nhất trong thiết kế các hệ cơ điện tử hiện đại. Tôi hy vọng rằng các sinh viên và cán bộ giảng dạy với khao khát tri thức và mong muốn mở rộng tầm nhìn sẽ tìm thấy giá trị của cuốn sách.

Tôi muốn gửi lời chúc mừng đến Tiến sĩ Phạm Anh Tuấn và các đồng nghiệp đã hoàn thành bản dịch tiếng Việt cuốn *The Mechatronics Handbook*. Tôi thật sự xúc động khi sinh viên tại Việt Nam sẽ được tiếp cận với cuốn sách bằng chính ngôn ngữ của mình.

Xin gửi lời chúc tốt đẹp nhất đến tất cả các bạn đọc cuốn sách này!

Thân ái,

Robert H. Bishop

21 - 08 - 2006

Lời cảm ơn

Cuốn sách *Cơ điện tử* tập II được tiếp tục biên dịch từ nguyên bản *The Mechatronics Handbook* bởi tập thể cán bộ phòng Động lực học vũ trụ và Cơ điện tử, Viện Công nghệ vũ trụ, Viện Khoa học và Công nghệ Việt Nam và sự hỗ trợ chuyên môn của nhiều Nhà khoa học, Nhà giáo có uy tín từ Viện Khoa học và Công nghệ Việt Nam, Đại học Công nghệ - Đại học Quốc gia Hà nội, Đại học Bách khoa Hà Nội và Trung tâm Khoa học và Công nghệ quân sự.

Cuốn *Cơ điện tử* này chứa đựng một khối lượng lớn kiến thức về Cơ điện tử và liên quan đến nhiều chuyên ngành khác nhau. Do vậy để thuận tiện cho việc tham khảo của bạn đọc, chúng tôi đã chia cuốn sách thành hai tập, với các nội dung chính sau:

Tập một: Phần I: Tổng quan về cơ điện tử

Phần II: Mô hình hóa hệ vật lý

Phần III: Cảm biến và cơ cấu chấp hành

Tập hai: Phần IV: Hệ thống và điều khiển

Phần V: Máy tính và hệ thống logic

Phần VI: Phần mềm và thu thập dữ liệu

Chúng tôi xin gửi lời cảm ơn đặc biệt tới GS. Đặng Vũ Minh, Nguyên Chủ tịch Viện Khoa học và Công nghệ Việt Nam vì sự quan tâm và hỗ trợ tài chính quý báu; GS. Robert H. Bishop, Chủ biên cuốn *The Mechatronics Handbook* vì sự giúp đỡ quý giá để cuốn *Cơ điện tử* được ra mắt.

Xin cảm ơn Viện Khoa học và Công nghệ Việt Nam, Viện Cơ học, Nhà xuất bản CRC Press (Mỹ), Nhà xuất bản Đại học Quốc gia Hà Nội đã tạo điều kiện giúp chúng tôi sớm đưa cuốn sách đến tay bạn đọc.

Chúng tôi chân thành cảm ơn các nhà khoa học, các nhà giáo trong và ngoài nước, bạn bè đồng nghiệp và người thân đã dành cho chúng tôi những lời đóng góp chân tình thẳng thắn về chuyên môn, sự quan tâm, động viên khích lệ tinh thần trong suốt quá trình biên dịch, hoàn thiện và xuất bản cuốn sách này.

Cuối cùng, chúng tôi xin chân thành cảm ơn toàn bộ thành viên phòng Động lực học vũ trụ và Cơ điện tử, đặc biệt là: Đỗ Thị Ngọc Oanh, Nguyễn Trường Thanh, Nguyễn Huy Thụy, Phùng Văn Hiệp, Đinh Công Huân... vì tinh thần làm việc nghiêm túc và tâm huyết trong quá trình biên dịch cuốn sách này.

Mặc dù đã rất cố gắng nhưng do khối lượng kiến thức chuyên sâu khá lớn, trong đó một số thuật ngữ kỹ thuật mới còn chưa được thống nhất về mặt dịch thuật ở Việt Nam, cuốn sách không thể tránh khỏi những sai sót. Chúng tôi rất mong nhận được sự phản hồi góp ý của bạn đọc.

Phạm Anh Tuấn

Viện Công nghệ vũ trụ

Viện Khoa học và Công nghệ Việt Nam

18 Hoàng Quốc Việt, Hà Nội, Việt Nam

Email: patuan@mechatronics.org.vn

Tel.: (+84-4) 37915106

Ban tư vấn khoa học

(Tập 2)

1) GS. TSKH. Nguyễn Đức Cương

Trung tâm Khoa học kỹ thuật và Công nghệ Quân sự

2) GS. TSKH. Nguyễn Cao Mệnh

Viện Cơ học, Viện Khoa học và Công nghệ Việt Nam

3) GS. TSKH. Đỗ Sanh

Đại học Bách khoa Hà Nội

4) GS. TSKH. Nguyễn Khoa Sơn

Viện Khoa học và Công nghệ Việt Nam

5) GS. TS. Ngô Diên Tập

Đại học Công nghệ, Đại học Quốc gia Hà Nội

6) GS. TSKH. Nguyễn Phú Thuỳ

Đại học Công nghệ, Đại học Quốc gia Hà Nội

7) GS.TS. Trần Mạnh Tuấn

Trung tâm Điều hành Ứng dụng Công nghệ Vũ trụ,
Viện Khoa học và Công nghệ Việt Nam

8) TS. Trần Quang Vinh

Đại học Công nghệ, Đại học Quốc gia Hà Nội

9) PGS. TS. Vương Đạo Vy

Đại học Công nghệ, Đại học Quốc gia Hà Nội

THE MECHATRONICS HANDBOOK

Editor-in-Chief

Robert H. Bishop

The University of Texas at Austin
Austin, Texas



ISA – The Instrumentation, Systems, and Automation Society



CRC PRESS

Boca Raton London New York Washington, D.C.

Lời nói đầu

Theo định nghĩa đầu tiên về cơ điện tử của Công ty Yasakawa Electric và những định nghĩa sau đó, nhiều sản phẩm công nghệ được thiết kế và sản xuất trong 25 năm gần đây tích hợp các hệ thống cơ khí và điện tử có thể coi là các *hệ cơ điện tử*. Tuy nhiên, nhiều kỹ sư và nhà nghiên cứu liên quan đến các sản phẩm này lại chưa từng được đào tạo chính thức về cơ điện tử. Cuốn *Cơ điện tử* là một tài liệu tham khảo cho các nhà thiết kế nhằm liên hệ những kinh nghiệm thực tế của họ với các lĩnh vực của cơ điện tử. Ngoài ra, cuốn sách này còn là tài liệu dùng trong các phòng nghiên cứu và phát triển của các đơn vị nghiên cứu, cơ quan quản lý, trong công nghiệp hay trong thư viện trong các trường đại học. Nó cũng được dùng cho những học giả có nhu cầu hiểu và giải thích quá trình thiết kế kỹ thuật. Khi mà sự phân chia truyền thống giữa các chuyên ngành khác nhau của kỹ thuật và khoa học máy tính trở nên khó xác định thì có thể thấy ngành cơ điện tử cung cấp một "định hướng" mới cho sinh viên kỹ thuật hiện đang được đào tạo theo các chương trình học truyền thống ở hầu hết các trường đại học. Các phòng thí nghiệm và các khóa học cơ điện tử ngày càng phát triển trong các trường đại học trên toàn thế giới. Thực tế đó được phản ánh qua danh sách các tác giả - 88 nhà khoa học và kỹ sư đến từ 13 nước - đã tham gia biên tập cuốn sách này. Hy vọng cuốn *Cơ điện tử* có thể được dùng như một tài liệu tham khảo chuẩn trong lĩnh vực cơ điện tử cho bạn đọc trên toàn thế giới.

Nội dung cuốn sách

Cuốn *Cơ điện tử* bao gồm 50 chương và bao quát các nội dung về các thành phần chính của cơ điện tử:

- Mô hình hóa hệ vật lý
- Cảm biến và cơ cấu chấp hành
- Tín hiệu và hệ thống
- Máy tính và hệ thống logic
- Phần mềm và thu thập dữ liệu

Phần I: Tổng quan về cơ điện tử

Phần mở đầu sẽ là phần định nghĩa và phân loại đối tượng chung của cơ điện tử. Nội dung của các chương nhằm đưa ra tổng quan và giới thiệu chung các thành phần chính của cơ điện tử. Với bạn đọc quan tâm tới vấn đề đào tạo liên quan đến cơ điện tử, phần này cũng đưa ra một thảo luận về định hướng mới trong chương trình giảng dạy về cơ điện tử. Các chương trong phần này bao gồm:

1. Cơ điện tử là gì?
2. Phương pháp thiết kế cơ điện tử
3. Giao diện, thiết bị đo và hệ thống điều khiển

4. Vi điện tử và các bộ điều khiển dựa trên vi xử lý
5. Giới thiệu về công nghệ micro và nano
6. Những hướng mới trong thiết kế các hệ cơ điện tử mini, micro, và nano và xây dựng chương trình đào tạo

Phần II: Mô hình hóa hệ vật lý

Các mô hình toán học, điện và cơ học cơ bản trong hầu hết các hệ cơ điện tử được trình bày trong phần này. Các thảo luận này nhằm mô tả chi tiết quá trình mô hình hệ vật lý, với các nội dung chính về cấu trúc và vật liệu, các hệ lưu chất, các hệ điện, các hệ nhiệt động lực học, các hệ tịnh tiến và quay, vấn đề mô hình liên quan đến MEMS và cơ sở vật lý của sự tương tự trong mô hình hệ thống. Các chương được liệt kê dưới đây:

7. Mô hình các hệ cơ điện
8. Cấu trúc và vật liệu
9. Mô hình hóa các hệ cơ học trong các ứng dụng cơ điện tử
10. Các hệ thống thủy lực
11. Kỹ thuật điện
12. Nhiệt động lực học
13. Mô hình hóa và mô phỏng hệ cơ điện tử micro (MEMS)
14. Tổng hợp, chế tạo, phân tích và tối ưu MEMS
15. Cơ sở vật lý của sự tương tự trong các mô hình hệ thống vật lý

Phần III: Cảm biến và cơ cấu chấp hành

Cơ sở của cảm biến và cơ cấu chấp hành được giới thiệu trong phần này. Nội dung của các chương đầu là các vấn đề cơ sở của thời gian, tần số, đặc tính của các cảm biến và cơ cấu chấp hành. Phần còn lại nói về hai nội dung chính đó là: cảm biến và cơ cấu chấp hành. Các chương này bao gồm các mối quan hệ vật lý cơ bản và các mô hình toán học liên quan đến các công nghệ cảm biến và cơ cấu chấp hành. Các chương của phần này được liệt kê dưới đây:

16. Giới thiệu về cảm biến và cơ cấu chấp hành
17. Cơ sở thời gian và tần số
18. Đặc tính của cảm biến và cơ cấu chấp hành
19. Cảm biến
    - 19.1. Cảm biến dịch chuyển thẳng và quay
    - 19.2. Cảm biến gia tốc
    - 19.3. Đo lực
    - 19.4. Đo mômen và công suất
    - 19.5. Đo lưu lượng
    - 19.6. Đo nhiệt độ
    - 19.7. Đo khoảng cách và cảm biến tiệm cận
    - 19.8. Hệ thống nhận biết ánh sáng, hình ảnh và nhận dạng
    - 19.9. Cảm biến micro tích hợp
20. Cơ cấu chấp hành
    - 20.1. Cơ cấu chấp hành cơ điện
    - 20.2. Máy điện
    - 20.3. Cơ cấu chấp hành áp-điện

20.4. Hệ thống chấp hành thuỷ lực và khí nén

20.5. MEMS: Phân tích, thiết kế và chế tạo các bộ chuyển đổi micro

Phần IV: Hệ thống và điều khiển

Phần này sẽ trình bày tổng quan về tín hiệu và hệ thống. Do đã có đầy đủ tài liệu cần thiết cho người đọc về các vấn đề chung của tín hiệu và hệ thống, ở đây sẽ không trình bày lặp lại. Mục đích của phần này là trình bày các vấn đề liên quan đến tín hiệu và điều khiển đặc biệt quan trọng đối với các nghiên cứu về cơ điện tử. Phần đầu tiên sẽ là các chương về vai trò của điều khiển trong cơ điện tử và vai trò của mô hình trong thiết kế cơ điện tử. Những chương này mở đầu cho các thảo luận cơ bản hơn về tín hiệu và hệ thống chiếm phần lớn trong phần này. Các vấn đề hiện đại của thiết kế điều khiển dùng các công nghệ tối ưu hóa từ lý thuyết H^2, điều khiển phi tuyến và thích nghi, mạng nơron và hệ thống mờ được trình bày với vai trò quan trọng trong thiết kế hệ thống kỹ thuật hiện đại. Phần IV cũng gồm một chương về tối ưu hóa thiết kế cho các hệ cơ điện tử. Phần này bao gồm các chương sau:

21. Vai trò của điều khiển trong cơ điện tử
22. Vai trò của mô hình hoá trong thiết kế cơ điện tử
23. Tín hiệu và hệ thống
    - 23.1. Tín hiệu thời gian liên tục và rời rạc
    - 23.2. Hệ thống số và biến đổi z
    - 23.3. Mô hình không gian trạng thái thời gian liên tục và rời rạc
    - 23.4. Hàm truyền và các biến đổi Laplace
24. Phân tích không gian trạng thái và các thuộc tính hệ thống
25. Đáp ứng của các hệ động
26. Phương pháp quỹ đạo nghiệm
27. Phương pháp đáp ứng tần số
28. Bộ lọc Kalman trong vai trò bộ quan sát trạng thái
29. Xử lý tín hiệu số trong các ứng dụng cơ điện tử
30. Thiết kế hệ điều khiển bằng phương pháp tối ưu H^2
31. Thiết kế điều khiển phi tuyến và thích nghi
32. Mạng nơron và hệ mờ
33. Điều khiển hiện đại một trục thuỷ lực-điện
34. Thiết kế tối ưu hệ cơ điện tử

Phần V: Máy tính và hệ thống logic

Sự phát triển của máy tính, sau đó là máy vi tính, các máy tính nhúng, và công nghệ thông tin liên quan, đã tác động sâu sắc đến thế giới. Nó càng đúng cho cơ điện tử, nơi tích hợp máy tính với các hệ cơ-điện dẫn đến việc sáng tạo ra các sản phẩm thông minh. Trong tương lai, các sản phẩm tốt hơn và thông minh hơn là kết quả của sự phát triển liên tục công nghệ máy tính và kỹ thuật phần mềm. Hai phần cuối cùng của cuốn *Cơ điện tử* dành cho máy tính và phần mềm. Phần V này sẽ tập trung vào phần cứng máy tính và các vấn đề liên quan tới logic, truyền thông, mạng, kiến trúc, phân tích lỗi, các máy tính nhúng và các bộ điều khiển khả trình. Phần này bao gồm các chương:

35. Giới thiệu về máy tính và hệ logic
36. Các khái niệm logic và thiết kế logic
37. Giao diện hệ thống

38. Truyền thông và mạng máy tính
39. Phân tích lỗi trong các hệ cơ điện tử
40. Thiết kế hệ logic
41. Các hệ thống dãy đồng bộ và không đồng bộ
42. Kiến trúc
43. Điều khiển với máy tính nhúng và bộ điều khiển logic khả trình

Phần VI: Phần mềm và thu thập dữ liệu

Do máy tính đóng vai trò trung tâm trong các sản phẩm cơ điện tử hiện đại, việc hiểu phương thức thu thập, xử lý và ghi dữ liệu của máy tính là khá quan trọng. Phần cuối của cuốn *Cơ điện tử* trình bày các vấn đề về phần mềm máy tính và thu thập dữ liệu. Các chương trong phần này là:

44. Giới thiệu về việc thu thập dữ liệu
45. Các kỹ thuật đo: cảm biến và bộ chuyển đổi
46. Biến đổi A/D và D/A
47. Chuẩn hóa tín hiệu
48. Các hệ thống thiết bị đo dựa trên máy tính
49. Thiết kế và phát triển phần mềm
50. Ghi dữ liệu

Lời cảm ơn

Tôi muốn gửi lời cảm ơn chân thành nhất của mình đến toàn bộ tập thể các tác giả của cuốn sách này. Những người đã dành những khoảng thời gian ít ỏi trong lịch làm việc dày đặc của mình để viết những bài viết hoàn hảo trong cuốn *Cơ điện tử* thật đáng được trân trọng. Tôi cũng muốn dành lời cảm ơn của mình đến toàn thể hội đồng tư vấn vì những giúp đỡ đáng quý trong giai đoạn ban đầu của các vấn đề được trình bày trong cuốn sách.

Cuốn sách là kết quả của sự cộng tác làm việc hiệu quả và chuyên nghiệp dưới sự điều hành của nhà xuất bản CRC. Tôi xin được cảm ơn tập thể biên tập và sản xuất cuốn sách:

Nora Konopka, biên tập viên

Michael Buso, biên tập viên

Susan Fox, tổng biên tập

Xin được cảm ơn những người bạn của tôi và đồng nghiệp Giáo sư Richard C. Dorf vì những giúp đỡ và chỉ dẫn nhiệt tình. Cuối cùng, tôi xin dành lời cảm ơn đặc biệt của mình đến Lynda Bishop, người đã quản lý những bản thảo trong thời gian đầu viết cuốn sách này. Sự giúp đỡ của bà trong việc hoàn thành và xuất bản cuốn sách là những đóng góp vô giá.

Chủ biên

Robert H. Bishop

Chủ biên



Robert H.Bishop là giáo sư tại khoa Công nghệ Vũ trụ và Cơ học kỹ thuật Trường đại học Tổng hợp Texas tại Austin (Mỹ), ông đã giành được vị trí giáo sư Joe J. King ngành kỹ thuật. Ông tốt nghiệp đại học và cao học tại khoa Công nghệ Vũ trụ, Trường đại học tổng hợp Texas A&M, ông hoàn thành luận án tiến sĩ tại Khoa Điện và Kỹ thuật Máy tính, Trường Đại học tổng hợp Rice. Trước khi giảng dạy ở Trường Đại học Tổng hợp Texas tại Austin, Ông là thành viên của Hội đồng kỹ thuật tại Phòng thí nghiệm MIT Charles Stark Draper. Tiến sĩ Bishop là chuyên gia trong lĩnh vực Công nghệ vũ trụ, các nghiên cứu tập trung chủ yếu về dẫn đường theo quỹ đạo và điều khiển tàu vũ trụ. Ông cùng hợp tác với Trung tâm Vũ trụ Johnson của NASA và Phòng thí nghiệm Động cơ phản lực nghiên cứu về kỹ thuật xác định độ chính xác khi hạ cánh trên Sao Hỏa. Ông là nhà nghiên cứu năng động, đồng tác giả của hơn 50 bài báo trên các tạp chí và báo cáo tại các Hội nghị khoa học. Ông đã hai lần được lựa chọn là thành viên Ban điều hành tại Phòng thí nghiệm Động cơ phản lực của NASA và thành viên Hiệp hội Welliver của Tập đoàn Boeing. Tiến sĩ Bishop là đồng tác giả cuốn *Modern Control Systems* với giáo sư R. C. Dorf, và là tác giả của hai cuốn *Learning with LabView* và *Design and Analysis Using Matlab và Simulink*. Ông mới nhận được huy chương John Leland Atwood của Hiệp hội các giảng viên kỹ thuật Mỹ và Học viện Hàng không vũ trụ Mỹ dành cho "Nhà lãnh đạo có những đóng góp quan trọng và bền vững trong đào tạo Công nghệ Vũ trụ".

Các tác giả

Maruthi R. Akella
University of Texas at Austin
Austin, Texas
Sami A. Al-Arian
University of South Florida
Tampa, Florida
M. Anjanappa
University of Maryland
Baltimore, Maryland
Dragos Arotaritei
Aalborg University Esbjerg
Esbjerg, Denmark
Ramutis Bansevicius
Kaunas University of
Technolo
Kaunas, Lithuania
Eric J. Barth
Vanderbilt University
Nashville, Tennessee
Peter Breedveld
University of Twente
Enschede, The Netherlands
Tomas Brezina
Technical University of Brno
Brno, Czech Republic
George T.-C. Chiu
Purdue University
West Lafayette, Indiana
George I. Cohn
California State University
Fullerton, California
Daniel A. Connors
University of Colorado
Boulder, Colorado
Neville Hogan
Massachusetts Institute of
Technology, Massachusetts
Rick Homkes
Purdue University
Kokomo, Indiana
Bouvard Hosticka
University of Virginia
Charlottesville, Virginia
Wen-Mei W. Hwu
Kevin C. Craig
Rennselaer Polytechnic
Institute
Troy, New York
Timothy P. Crain II
NASA Johnson Space Center
Houston, Texas
Jace Curtis
National Instruments, Inc.
Austin, Texas
K. Datta
University of Maryland
Baltimore, Maryland
Raymond de Callafon
University of California
La Jolla, California
Santosh Devasia
University of Washington
Seattle, Washington
Ivan Dolezal
Technical University of
Liberec
Liberec, Czech Republic
C. Nelson Dorny
University of Pennsylvania
Philadelphia, Pennsylvania
Stephen A. Dyer
Kansas State University
Manhattan, Kansas
M.A. Elbestawi
McMaster University
Hamilton, Ontario, Canada
Eniko T. Enikov
University of Arizona
Tuscon, Arizona
Thomas R. Kurfess
Georgia Institute of
Technology
Atlanta, Georgia
Kam Leang
University of Washington
Seattle, Washington
Chang Liu
University of Illinois
Halit Eren
Curtin University of
Technology
Bentley, Australia
H. R. (Bart) Everett
Space and Naval Warfare
Systems Center
San Diego, California
Jorge Fernando Figueroa
NASA Stennis Space Center
New Orleans, Louisiana
C. J. Fraser
University of Abertay Dundee
Dundee, Scotland
Kris Fuller
National Instruments, Inc.
Austin, Texas
Ivan J. Garshelis
Magnova, Inc.
Pittsfield, Massachusetts
Carroll E. Goering
University of Illinois
Urbana, Illinois
Michael Goldfarb
Vanderbilt University
Nashville, Tennessee
Margaret H. Hamilton
Hamilton Technologies, Inc.
Cambridge, Massachusetts
Cecil Harrison
University of Southern
Mississippi
Hattiesburg, Mississippi
Bonnie S. Heck
Georgia Institute of
Technology
Atlanta, Georgia
Ondrej Novak
Technical University of
Liberec
Liberec, Czech Republic
Cestmir Ondrusek
Technical University of Brno
Brno, Czech Republic

University of Illinois
Urbana, Illinois
Mohammad Ilyas
Florida Atlantic University
Boca Raton, Florida
Florin Ionescu
University of Applied Sciences
Konstanz, Germany
Stanley S. Ipson
University of Bradford
Bradford, West Yorkshire, England
Rolf Isermann
Darmstadt University of Technology
Darmstadt, Germany
Hugh Jack
Grand Valley State University
Grand Rapids, Michigan
Jeffrey A. Jalkio
Univeristy of St. Thomas
St. Paul, Minnesota
Rolf Johansson
Lund Institute of Technology
Lund, Sweden
J. Katupitiya
The University of New South Wales
Sydney, Australia
Ctirad Kratochvil
Technical University of Brno
Brno, Czech Republic
T. Song
University of Maryland
Baltimore, Maryland
Massimo Sorli
Politecnico di Torino
Torino, Italy
Andrew Sterian
Grand Valley State University
Grand Rapids, Michigan
Alvin Strauss
Vanderbilt University
Nashville, Tennessee
Fred Stolfi
Rennselaer Polytechnic Institute
Troy, New York
Job van Amerongen
University of Twente
Urbana, Illinois
Michael A. Lombardi
National Institute of Standards and Technology
Boulder, Colorado
Raul G. Longoria
University of Texas at Austin
Austin, Texas
Kevin M. Lynch
Northwestern University
Evanston, Illinois
Sergey Edward Lyshevski
Indiana University-Purdue University Indianapolis
Indianapolis, Indiana gy
Tom Magruder
National Instruments, Inc.
Austin, Texas
Francis C. Moon
Cornell University
Ithaca, New York
Thomas N. Moore
Queen's University
Kingston, Ontario, Canada
Michael J. Moran
The Ohio State University
Columbus, Ohio, es
Pamela M. Norris
University of Virginia
Charlottesville, Virginia
Leila Notash
Queen's University
Kingston, Ontario, Canada
Richard Thorn
University of Derby
Derby, England
Rymantas Tadas Tolocka
Kaunas University of Technology
Kaunas, Lithuania
M. J. Tordon
The University of New South Wales
Sydney, Australia
Mike Tyler
National Instruments, Inc.
Austin, Texas
Crina Vlad
Politehnica University of Bucharest
Bucharest, Romania
Hitay Özbay
The Ohio State University
Columbus, Ohio
Joey Parker
University of Alabama
Tuscaloosa, Alabama
Stefano Pastorelli
Politecnico di Torino
Torino, Italy
Michael A. Peshkin
Northwestern University
Evanston, Illinois
Carla Purdy
University of Cincinnati
Cincinnati, Ohio
M. K. Ramasubramanian
North Carolina State University
Raleigh, North Carolina
Giorgio Rizzoni
The Ohio State University
Columbus, Ohio
Armando A. Rodriguez
Arizona State University
Tempe, Arizona
Momoh-Jimoh Eyiomika Salami
International Islamic University of Malaysia
Kuala Lumpur, Malaysia
Mario E. Salgado
Universidad Tecnica Federico Santa Maria
Valparaiso, Chile
Jyh-Jong Sheen
National Taiwan Ocean University
Keelung, Taiwan
Bogdan M. Wilamowski
University of Wyoming
Laramie, Wyoming
Juan I. Yuz
Universidad Tecnica Federico Santa Maria, Chile
Qin Zhang
University of Illinois
Urbana, Illinois
Qingze Zou
University of Washington
Seattle, Washington
Enschede, The Netherlands

Mục lục

PHẦN IV: HỆ THỐNG VÀ ĐIỀU KHIỂN

21 Vai trò của điều khiển trong cơ điện tử *Job van Amerongen* 21-1

22 Vai trò của mô hình hoá trong thiết kế cơ điện tử *Jeffrey A. Jalkio* 22-1

23 Tín hiệu và hệ thống
Momoh-Jimoh Eyiomika Salami, Rolf Johansson, Kam Leang, Qingze Zou, Santosh Devasia và C. Nelson Dorny 23-1

24 Phân tích không gian trạng thái và các thuộc tính hệ thống
Mario E. Salgado và Juan I. Yuz 24-1

25 Đáp ứng của các hệ động *Raymond de Callafon* 25-1

26 Phương pháp quỹ đạo nghiệm *Hitay Ozbay* 26-1

27 Phương pháp đáp ứng tần số *Jyh-Jong Sheen* 27-1

28 Bộ lọc Kalman trong vai trò bộ quan sát trạng thái *Timothy P. Crain II* 28-1

29 Xử lý tín hiệu số trong các ứng dụng cơ điện tử
Bonnie S. Heck và Tomas R. Kurfess 29-1

30 Thiết kế hệ điều khiển bằng phương pháp tối ưu $\mathcal{H}^2$
Armando A. Rodriguez 30-1

31 Thiết kế điều khiển phi tuyến và thích nghi *Maruthi R. Akella* 31-1

32 Mạng nơron và hệ mờ Bogdan M. Wilamowski 32-1

33 Điều khiển hiện đại một trục thuỷ lực-điện
Florin Ionescu, Crina Vlad và Dragos Arotaritei 33-1

34 Thiết kế tối ưu hệ cơ điện tử
Tomas Brezina, Ctirad Kratochvil và Cestmir Ondrusek 34-1

PHẦN V: MÁY TÍNH VÀ HỆ THỐNG LOGIC

35 Giới thiệu về máy tính và hệ logic *Kevin Craig và Fred Stolfi* 35-1

36 Các khái niệm logic và thiết kế logic *George I. Cohn* 36-1
37 Giao diện hệ thống *M.J. Tordon và J. Katupitiya* 37-1
38 Truyền thông và mạng máy tính *Mohammad Ilyas* 38-1
39 Phân tích lỗi trong các hệ cơ điện tử *Leila Notash và Thomas N. Moore* 39-1
40 Thiết kế hệ logic *M. K. Ramasubramanian* 40-1
41 Các hệ thống dãy đồng bộ và không đồng bộ *Sami A. Al-Arian* 41-1
42 Kiến trúc *Daniel A. Connors và Wen-mei W. Hwu* 42-1
43 Điều khiển với máy tính nhúng và bộ điều khiển logic khả trình *Job van Amerongen* 43-1

PHẦN VI: PHẦN MỀM VÀ THU THẬP DỮ LIỆU

44 Giới thiệu về việc thu thập dữ liệu *Jace Curtis* 44-1
45 Các kỹ thuật đo: cảm biến và bộ chuyển đổi *Cecil Harrison* 45-1
46 Biến đổi A/D và D/A *Mike Tyler* 46-1
47 Chuẩn hóa tín hiệu *Stephen A. Dyer* 47-1
48 Các hệ thống thiết bị đo dựa trên máy tính *Kris Fuller* 48-1
49 Thiết kế và phát triển phần mềm *Margaret H. Hamilton* 49-1
50 Ghi dữ liệu *Tom Magruder* 50-1

IV

Hệ thống và điều khiển

21 Vai trò của điều khiển trong cơ điện tử *Job van Amerongen* 21-1

Mở đầu • Các thành phần chính của hệ cơ điện tử có điều khiển • Mô hình hóa, thiết kế và điều khiển • Các ví dụ hệ cơ điện tử hiện đại • Yêu cầu đặc biệt của các hệ cơ điện tử so với các hệ "cổ điển" và việc thiết kế

22 Vai trò của mô hình hoá trong thiết kế cơ điện tử *Jeffrey A. Jalkio* 22-1

Mô hình hóa là một phần của quá trình thiết kế • Mục đích của mô hình hóa • Mô hình hoá các hệ thống và các tín hiệu

23 Tín hiệu và hệ thống *Momoh-Jimoh Eyiomika Salami, Rolf Johansson, Kam Leang, Qingze Zou, Santosh Devasia và C. Nelson Dorny* 23-1

Tín hiệu thời gian liên tục và rời rạc • Biến đổi z và các hệ thống số • Các mô hình không gian - trạng thái thời gian rời rạc và liên tục • Hàm truyền và biến đổi Laplace

24 Phân tích không gian trạng thái và các thuộc tính hệ thống *Mario E. Salgado và Juan I. Yuz* 24-1

Mô hình: Những khái niệm cơ bản • Biến trạng thái: Những khái niệm cơ bản • Mô tả không gian trạng thái cho các hệ thời gian liên tục • Mô tả không gian trạng thái cho hệ thống thời gian rời rạc và dữ liệu lấy mẫu • Các mô hình không gian trạng thái cho các hệ kết nối Các thuộc tính của hệ thống Bộ quan sát trạng thái • Phản hồi trạng thái • Phản hồi trạng thái quan sát được

25 Đáp ứng của các hệ động *Raymond de Callafon* 25-1

Hệ thống và phân tích tín hiệu • Đáp ứng động học • Các thông số hoạt động của hệ thống động

26 Phương pháp quỹ đạo nghiệm *Hitay Özbay* 26-1

Giới thiệu • Các vị trí điểm cực mong muốn • Xây dựng quỹ đạo nghiệm • Quỹ đạo nghiệm bù • Quỹ đạo nghiệm cho hệ thống có trễ thời gian • Chú ý và tham khảo

27 Phương pháp đáp ứng tần số *Jyh-Jong Sheen* 27-1

Mở đầu • Đồ thị Bode • Đồ thị cực • Biên độ lôgarit đối với đồ thị pha • Xác định hàm truyền bằng thực nghiệm • Tiêu chuẩn ổn định Nyquist • Sự ổn định tương đối

28 Bộ lọc Kalman trong vai trò bộ quan sát trạng thái *Timothy P. Crain II* 28-1

Bộ lọc Kalman tuyến tính thời gian rời rạc • Các công thức bộ lọc Kalman khác • Tổng kết và tóm tắt các công thức • Xem xét sự thực hiện

29 Xử lý tín hiệu số trong các ứng dụng cơ điện tử

Bonnie S. Heck và Tomas R. Kurfess 29-1

Giới thiệu • Cơ sở về xử lý tín hiệu • Ánh xạ tín hiệu liên tục sang tín hiệu rời rạc • Thiết kế bộ lọc số • Thiết kế điều khiển số

30 Thiết kế hệ điều khiển bằng phương pháp tối ưu $\mathcal{H}^2$ *Armando A. Rodriguez* 30-1

Giới thiệu • Cấu trúc thiết kế hệ điều khiển tổng quát • Bài toán phản hồi đầu ra $\mathcal{H}^2$ • Bài toán phản hồi trạng thái $\mathcal{H}^2$ • Bài toán nội xạ đầu ra $\mathcal{H}^2$ • Kết luận

31 Thiết kế điều khiển phi tuyến và thích nghi *Maruthi R.Akella* 31-1

Giới thiệu • Lý thuyết Lyapunov cho các hệ dừng • Định lý Lyapunov đối với hệ không dừng • Lý thuyết điều khiển thích nghi • Các hệ điều khiển thích nghi phi tuyến • Ví dụ về điều chỉnh thích nghi tư thế bay cho tàu vũ trụ • Điều khiển thích nghi phản hồi đầu ra • Các bộ quan sát thích nghi và điều khiển phản hồi đầu ra • Nhận xét

32 Mạng nơron và hệ mờ *Bogdan M. Wilamowski* 32-1

Mạng nơron và hệ mờ • Tế bào nơron • Mạng nơron truyền thẳng • Các thuật toán học cho mạng nơron • Các mạng truyền thẳng đặc biệt • Mạng nơron hồi qui • Hệ mờ • Thuật toán di truyền

33 Điều khiển hiện đại một trục thuỷ lực-điện

Florin Ionescu, Crina Vlad và Dragos Arotaritei 33-1

Giới thiệu chung • Tổng quan về ROBI 3, rôbốt dạng tọa độ đề - các với ba trục thủy lực-điện • Mô hình toán học và mô phỏng các trục thủy lực-điện • Điều khiển trục thủy lực-điện bằng các bộ điều khiển truyền thống • Điều khiển trục thủy lực-điện với bộ điều khiển mờ • Điều khiển trục thủy lực-điện dùng kỹ thuật nơron • Điều khiển trục thủy lực-điện dùng kỹ thuật mờ-nơron • Phần mềm • Kết luận

34 Thiết kế tối ưu hệ cơ điện tử

Tomas Brezina, Ctirad Kratochvil và Cestmir Ondrusek 34-1

Giới thiệu • Các phương pháp tối ưu • Thiết kế tối ưu động cơ cảm ứng (IM) • Sử dụng mạng nơron cho việc nhận dạng tham số của một hệ thống động lực học

21

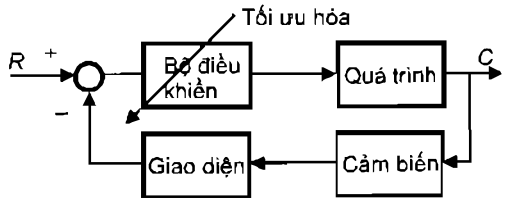
Vai trò của điều khiển trong Cơ điện tử

21.1 Mở đầu 21-1
21.2 Các thành phần chính của hệ cơ điện tử có điều khiển 21-3
21.3 Mô hình hóa, thiết kế và điều khiển 21-3
21.4 Các ví dụ hệ cơ điện tử hiện đại 21-11
21.5 Yêu cầu đặc biệt của các hệ cơ điện tử so với các hệ "cổ điển" và việc thiết kế 21-14

Job van Amerongen
University of Twente

21.1 Mở đầu

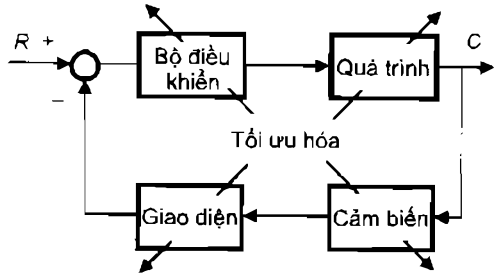
"Thiết kế Cơ điện tử liên quan tới thiết kế tối ưu và tích hợp một hệ cơ học và hệ điều khiển nhúng của nó". Định nghĩa này hàm ý rằng hệ cơ học được nâng cấp bằng các thành phần điện tử để đạt được hiệu suất tốt hơn, hệ thống mềm dẻo hơn hoặc giảm giá thành hệ thống. Trong nhiều trường hợp các thành phần điện tử có dạng một hệ điều khiển dựa trên máy tính. Điều này không có nghĩa rằng mọi hệ cơ học được điều khiển đều là hệ cơ điện tử bởi vì trong nhiều trường hợp, phần điều khiển chỉ đơn giản là được thêm vào hệ cơ học như một thủ tục thiết kế tiếp theo. Việc tiếp cận hệ cơ điện tử đòi hỏi sự lựa chọn tối ưu để thực hiện các đặc tính thiết kế trong các lĩnh vực khác nhau. Trong kĩ thuật điều khiển, thiết kế một hệ điều khiển tối ưu đã được nghiên cứu kỹ và đã có các phương pháp chuẩn cho các hệ tuyến tính. Vấn đề tối ưu được phát biểu như sau: cho một quá trình được điều khiển và cho một tiêu chí (ví dụ hàm giá), tìm các tham số điều khiển tối ưu sao cho hàm giá là tối thiểu. Với một bộ điều khiển phản hồi trạng thái và một hàm giá bậc hai, việc tìm giải pháp cho bài toán tối ưu được thực hiện nhờ phần mềm thiết kế bộ điều khiển chuẩn như Matlab [1] (hình 21.1).



HÌNH 21.1 Tối ưu hóa bộ điều khiển

Trái lại, thiết kế cơ điện tử đòi hỏi không chỉ bộ điều khiển là tối ưu. Nó đòi hỏi sự tối ưu của cả hệ thống. Trong trường hợp lý tưởng, tất cả các thành phần của hệ thống: đối tượng, bộ điều khiển, cảm biến và các cơ cấu chấp hành phải được tối ưu đồng thời (hình 21.2).

Nói chung điều này là không khả thi. Vấn đề đặt chưa rõ ràng và phải được tách thành các vấn đề nhỏ hơn để có thể tối ưu hóa riêng biệt. Sau đó các giải pháp cho từng phần được kết hợp lại và hoạt động của toàn bộ hệ thống phải được đánh giá. Việc điều chỉnh lần cuối cùng các thành phần của hệ thống sẽ đưa ra một giải pháp tối ưu khác.

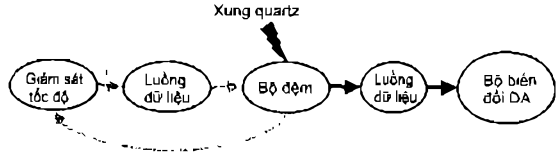


HÌNH 21.2 Tối ưu hóa các thành phần của hệ thống một cách đồng bộ

Trong giai đoạn thiết kế ý tưởng ban đầu, cần phải xác định đâu là vấn đề phải được giải quyết bằng cơ học, đâu là vấn đề phải được giải quyết bằng điện tử. Trong bước này, phải tìm ra các tính chất cơ học chính và đưa ra một mô hình đơn giản để thiết kế bộ điều khiển. Những ý tưởng ban đầu về các cảm biến, các cơ cấu chấp hành và các giao diện cần phải đưa ra trong giai đoạn này. Khi thiết kế của các phần khác nhau đã tương đối chi tiết, có thể sử dụng thông tin về những thiết kế đó để đánh giá toàn hệ thống. Chúng có thể được thay đổi để đạt được các thiết kế chi tiết và thực tế hơn cho các thành phần khác nhau.

Mặc dù "Cơ điện tử" là một thuật ngữ mới nhưng sản phẩm cơ điện tử đã xuất hiện một thời gian khá lâu. Thực tế, tất cả các hệ cơ học điều khiển bằng điện tử đều dựa trên ý tưởng cải tiến sản phẩm bằng cách thêm vào các đặc tính có được từ các lĩnh vực khác. Các thiết kế cơ điện tử tốt phải dựa trên sự tiếp cận hệ thống thực. Nhưng phần lớn các kỹ sư điện tử phải đối mặt với những thiết kế mà trong đó các tham số chính đã được cố định dựa trên những đánh giá tĩnh hoặc sự cân nhắc về mặt kinh tế. Điều đó cản trở sự tối ưu toàn bộ hệ thống, thậm chí cả khi phần điều khiển đã được tối ưu.

Trong giai đoạn cuối của máy quay đĩa, các thiết kế phức tạp hơn sử dụng phản hồi tốc độ kết hợp với một giá quay quang điện để đạt được tốc độ quay không đổi. Nhưng máy đọc đĩa compact thực sự là một thiết kế mới mẻ. Thay vì phải giữ số vòng quay đĩa không đổi, nó giữ tốc độ đầu đọc dọc theo rãnh của đĩa không đổi. Điều này có nghĩa là đĩa phải quay chậm hơn khi rãnh có bán kính lớn hơn được đọc. Các "bit" đọc từ đĩa CD được đưa vào trong bộ đệm (được điều khiển bởi một tinh thể thạch anh) để gửi các thông tin của nó tới bộ biến đổi DA. Điều này cho phép tốc độ đọc không đổi và loại trừ sự thay đổi bất thường của tốc độ. Một thiết bị cơ học thuần tuý không thể làm được như vậy ngay cả khi nó được trang bị một hệ thống điều khiển tốc độ tốt. Trong thực tế, vòng điều khiển tốc độ quay đĩa không cần có các đặc tính quá chính xác. Nó chỉ cần ngăn chặn tràn trên hoặc tràn dưới bộ đệm. Độ chính xác cao có thể đạt được trong chế độ vòng hở, được điều khiển bởi một tinh thể thạch anh (hình 21.3).



HÌNH 21.3 Tổ hợp của điều khiển kín và điều khiển hở trong một máy CD

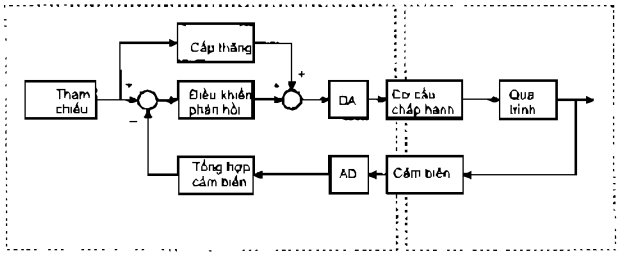
Tính mềm dẻo của sự kết hợp giữa các hệ cơ chính xác và bộ điều khiển điện tử cho phép phát triển các máy CD-ROM chạy ở tốc độ lớn hơn 50 lần so với tốc độ của của các máy đọc CD trước kia. Một cách nghĩ mới là cần thiết để cho ra một giải pháp mới. Mặt khác, máy đọc CD vẫn là một vấn đề phức tạp của cơ học chính xác. Không một thiết bị nhớ điện tử bán dẫn nào có thể cạnh

tranh một cách kinh tế với khả năng lưu trữ quang cơ của đĩa CD và thế hệ sau đó là DVD. Nhưng điều này có thể thay đổi rất nhanh.

21.2 Các thành phần chính của hệ cơ điện tử có điều khiển

Một hệ cơ điện tử bao gồm phần cơ thực hiện các dịch chuyển nhất định và phần điện tử (trong nhiều trường hợp là một hệ máy tính nhúng) tạo nên sự thông minh cho hệ thống.

Trong phần cơ của hệ thống, lực đóng vai trò quan trọng nhất. Trái lại, ở phần điện tử của hệ thống, xử lý thông tin là nhiệm vụ chính. Các cảm biến biến đổi các dịch chuyển cơ học thành các tín hiệu điện mà ở đó chỉ có nội dung thông tin là quan trọng hoặc thậm chí thành các thông tin thuần tuý ở dạng số (nếu cần thì qua một bộ biến đổi AD). Bộ khuyếch đại công suất biến đổi các tín hiệu thành năng lượng đã được điều chế. Trong hầu hết các trường hợp, nguồn năng lượng là điện nhưng cũng có thể là các nguồn khác như thủy lực, khí nén. Một hệ dịch chuyển cơ có điều khiển thường bao gồm một cấu trúc cơ, một hoặc nhiều cơ cấu chấp hành để tạo ra các dịch chuyển mong muốn và một bộ điều khiển để điều khiển các cơ cấu chấp hành bằng các thuật toán điều khiển thuận và điều khiển có phản hồi sử dụng cảm biến (hình 21.4).



HÌNH 21.4 Hệ cơ điện tử

21.3 Mô hình hóa, thiết kế và điều khiển

Mô hình hóa

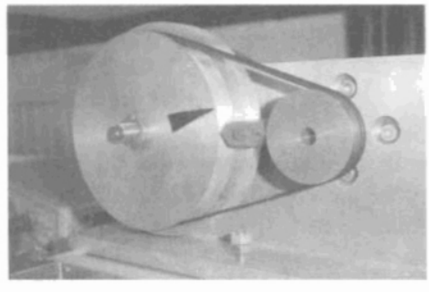
Khi thiết kế các hệ cơ điện tử, điều quan trọng là sự thay đổi trong cấu trúc và bộ điều khiển phải được đánh giá đồng thời. Mặc dù một bộ điều khiển phù hợp cho phép xây dựng một cấu trúc rẻ hơn, nhưng một hệ cơ học thiết kế kém sẽ không bao giờ có thể hoạt động tốt bằng cách thêm vào một bộ điều khiển phức tạp. Vì vậy, điều quan trọng là trong giai đoạn đầu của thiết kế phải có sự lựa chọn thích hợp đối với các tính chất cơ học cần thiết để hệ được điều khiển hoạt động tốt. Mặt khác, sử dụng các khả năng của bộ điều khiển để bù cho những khiếm khuyết về mặt cơ học có thể cho phép xây dựng cấu trúc cơ học với giá thành rẻ hơn. Điều này đòi hỏi trong giai đoạn đầu của thiết kế phải có một mô hình đơn giản để phát hiện những yếu tố hạn chế của hệ thống. Luôn luôn có một khoảng cách giữa phần mềm mô hình hóa và mô phỏng để đánh giá các cấu trúc cơ khí và phần mềm sử dụng cho việc thiết kế các bộ điều khiển. Các kỹ sư cơ khí đã quen với các chương trình phần tử hữu hạn để nghiên cứu các tính chất động lực học của các cấu trúc cơ học. Chỉ sau khi đơn giản xuống các mô hình bậc thấp, thì những mô hình này mới có thể sử dụng để thiết kế các bộ điều khiển. Mặt khác, phần mềm kỹ thuật điều khiển thường dùng chưa hỗ trợ trực tiếp quá trình thiết kế cơ điện tử. Trong quá trình mô hình hóa, các hàm truyền và những mô tả không gian trạng thái thường mất đi mối quan hệ với các tham số vật lý của cấu trúc cơ. Cần phải có các công cụ cho phép mô hình hóa hệ cơ học sao cho các tham số vật lý chính (như là khối lượng và độ cứng) được bảo toàn trong mô hình, đồng thời cung cấp một giao diện cho việc thiết kế bộ điều khiển và các công cụ mô phỏng mà các kỹ sư điều khiển đã quen dùng (Coelingh [2]; Coelingh, De Vries và Van Amerongen [3]).

Mô phỏng là một công cụ quan trọng để đánh giá thiết kế của các hệ cơ điện tử. Hầu hết các chương trình như Simulink [1] dùng các biểu diễn dưới dạng sơ đồ khối và không hỗ trợ mô hình

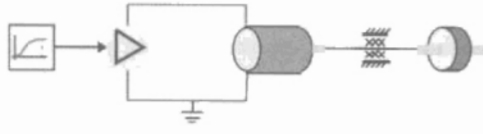
vật lý điều chỉnh trực tiếp các tham số vật lý của cấu trúc cơ và của các bộ điều khiển. Điều này rất cần thiết khi thiết kế các hệ cơ điện tử. Gần đây, đã có các chương trình cho phép mô hình hóa kiểu vật lý trong *các miền vật lý khác nhau*. Họ sử dụng phương pháp hướng đối tượng cho phép mô hình hóa theo thứ bậc và cho phép dùng lại các mô hình. Bậc tính toán chỉ được xác định sau khi tổ hợp các hệ con. Ví dụ: chương trình 20-sim [4], được mô tả bởi Broenink [5] cũng như CAMAS và Dymola [6].

Trong mục này, mô hình hóa và chương trình mô phỏng 20-sim sẽ được dùng để minh họa việc thiết kế đồng thời cấu trúc và bộ điều khiển trong một hệ cơ điện tử. 20-sim hỗ trợ việc mô hình hóa hướng đối tượng. Năng lượng và các cổng tín hiệu vào ra xác định mỗi đối tượng (Weustink, De Vries, and Breedveld [7]). Bên trong đối tượng có thể có các đối tượng khác, hoặc cấp thấp nhất là các phương trình. Những dạng khác nhau của một đối tượng có thể chứa các mô tả khác nhau hoặc chi tiết hơn miễn là xác định được giao diện (số lượng và kiểu của các cổng). Mô hình hóa có thể được bắt đầu bằng việc kết nối một cách đơn giản các mô hình con (hoặc rỗng). Sau đó chúng được bổ sung các mô tả thực với độ phức tạp khác nhau. De Vries [8] gọi đó là mô hình hóa kiểu *đa hình thái*. Các mô hình con có thể được xây dựng từ các mô hình con khác theo cấu trúc thứ bậc. Mô hình vật lý đạt được bằng cách kết hợp các mô hình con theo phương pháp *dòng năng lượng*, chứ không phải theo *tín hiệu* như là điện áp, dòng điện, lực và vận tốc. Cách mô hình hóa này rất thích hợp cho thiết kế hệ thống cơ điện tử. Điều đó được minh họa bằng ví dụ sau. Xét quá trình thiết kế một hệ servo đơn giản bao gồm một nguồn điện áp, một động cơ một chiều và một tải cơ học được điều khiển qua một bộ truyền lực (hình 21.5).

Bỏ qua bộ truyền lực, dây curoa được xem là không đàn hồi và tỉ số truyền nhận được bằng cách thay đổi hằng số của động cơ. Nếu một bộ khuyếch đại công suất được điều khiển bởi một bộ phát tín hiệu là nguồn điện áp, chúng ta có thể vẽ sơ đồ biểu diễn như trên hình 21.6. Ở giai đoạn này, các thành phần khác nhau trong mô hình này vẫn là rỗng. Nhưng tất cả các thành phần đều có các cổng điện tử hoặc cổng cơ. Với các giao diện (là các cổng) thích hợp đã được xác định, các thành phần có thể được nối với nhau.



HÌNH 21.5 Hệ servo-DC đơn giản



HÌNH 21.6 Sơ đồ biểu tượng của hệ servo đơn giản



HÌNH 21.7 Biểu tượng của động cơ được khai triển thành các thành phần vật lý lý tưởng

Ở bước tiếp theo chúng ta có thể chi tiết hóa mô hình động cơ DC. Hình 21.7 là một giải pháp mô tả động cơ DC. Động cơ được mô tả bằng một số các *phần tử vật lý lý tưởng*, mỗi phần tử mô tả một mối quan hệ vật lý cơ bản. Động cơ có một cổng điện (EL) cũng như một cổng cơ (MECH).

Mỗi *phần tử* trong hình này có thể được mô tả như là một phần tử với một cổng điện hoặc một cổng cơ. Khái niệm cổng tạo ra sự rõ ràng hơn trong các biểu đồ quan hệ [9-12]. Đối với các phần tử điện, đó là điện áp đặt lên mỗi phần tử và dòng qua nó. Đối với các phần tử cơ, đó là mô men và vận tốc góc. Tích của các biến liên hợp ($P=ui$ or $P=T\omega$) chính là công suất.

Nếu phát triển tiếp sơ đồ thứ bậc, ta sẽ được các phương trình. Ví dụ, một điện trở có thể được mô tả bằng phương trình:

$$p \cdot u = R * p \cdot i \tag{21.1}$$

trong đó các biến $p.u$ và $p.i$ chỉ ra các biến liên hợp u và i của p. Chú ý rằng đó là một phương trình chứ không phải một lệnh gán. Nó cũng có thể được viết ở dạng:

$$p \cdot i = 1/R * p \cdot u \tag{21.2}$$

Tương tự, cuộn cảm có thể được mô tả bằng các phương trình:

$$p \cdot u = L * \mathrm{ddt}(p \cdot i) \text{ hoặc } (p \cdot i) = 1/L * \mathrm{int}(p \cdot u) \tag{21.3}$$

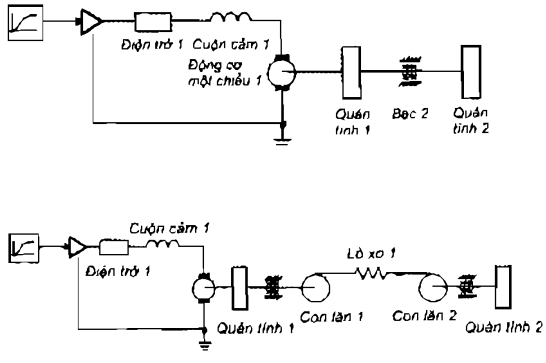
trong đó $\mathrm{ddt}(p \cdot i)$ biểu diễn di/dt và $\mathrm{int}(p \cdot u)$ biểu diễn $\int u dt$. Phần tử R thích hợp với cả hai dạng mô tả. Với phần tử I, dạng tích phân thích hợp hơn khi mô phỏng. Chương trình 20-sim xác định *dạng nhân quả mong muốn* và tự dẫn ra các phương trình.

Dòng năng lượng hoặc *công suất* P là tích của *hai tín hiệu liên hợp*. Các ví dụ của biểu thức này trong miền cơ và điện là:

$$P = Fv \text{ hoặc } P = T\omega \tag{21.4}$$

$$P = ui \tag{21.5}$$

trong đó F là lực, v là vận tốc, T là mô men, ω là vận tốc góc, u là điện áp và i là dòng điện.



HÌNH 21.9 Mô hình được triển khai với bộ truyền lực

Khi khai triển toàn bộ hình 21.6 chúng ta nhận được hình 21.8. Khi mô hình này được xử lý, một thông báo xuất hiện chỉ ra rằng *quán tính 2* ở trạng thái phụ thuộc. Hai *quán tính* trong mô

hình này luôn có cùng tốc độ và vì vậy chúng phụ thuộc lẫn nhau. Chúng không thể có các điều kiện đầu độc lập. Thông báo chỉ ra rằng phần tử này chỉ có thể được viết dưới dạng đạo hàm:

$$T = Jd\omega / dt \tag{21.6}$$

Có nhiều cách để giải quyết vấn đề này.

1. Hai quán tính có thể được tổ hợp thành một (chương trình sẽ làm điều này một cách tự động). Một thông báo xuất hiện chỉ ra rằng sự phụ thuộc của hai quán tính đã được giải quyết một cách tượng trưng.
2. Giải đạo hàm bằng thuật toán tích phân ẩn.
3. Bộ truyền lực có thể được đưa vào, bao gồm cả độ đàn hồi của dây curoa.

Nếu độ đàn hồi là không đáng kể, thì giải pháp 1 đưa ra mô hình đơn giản nhất. Ngoài ra, một câu hỏi được đặt ra là độ đàn hồi của dây curoa có thể bỏ qua được hay không. Nếu không thì mô hình phải được mở rộng với một phần tử lò xo. Nếu bộ truyền lực có độ cứng lớn thì sẽ gây ra vấn đề động lực học tần số cao và do đó mô phỏng chậm là không cần thiết. Mặt khác nếu độ đàn hồi lớn như trong hệ này thì cảnh báo sẽ nhắc nhở người thiết kế rằng mô hình có thể bị đơn giản quá. Trong hình 21.9 đã bổ sung bộ truyền lực bao gồm một phần tử lò xo. Việc xử lý mô hình này không gây ra một cảnh báo nào.

Ví dụ này minh họa khả năng một phần mềm hiện đại có thể trợ giúp xây dựng mô hình có độ phức tạp cần thiết cho một vấn đề cụ thể. Các mô hình vật lý ở dạng sơ đồ biểu tượng dựa trên việc nối các phần tử bằng các cổng năng lượng có thể hỗ trợ quá trình mô hình hóa này. Người dùng có thể chọn cách nhìn thích hợp như sơ đồ quan hệ, sơ đồ biểu tượng với các phần tử vật lý lý tưởng hay sử dụng các mô hình con mức cao hơn như trong hình 21.6. Phần tiếp theo sẽ giới thiệu cách dùng mô hình này để thiết kế các bộ điều khiển.

Phương pháp luận thiết kế hệ điều khiển

Nhiều quá trình có thể được điều khiển một cách hợp lý bằng các bộ điều khiển PID. Đó là do các quá trình này có thể được mô tả chính xác bằng một mô hình bậc hai. Các quy tắc điều chỉnh như những quy tắc của Ziegler Nichols, cho phép những người ít kinh nghiệm có thể điều chỉnh các bộ điều khiển như thế. Các mô hình tương đối đơn giản cũng có thể mô tả nhiều hệ cơ điện tử. Một hệ cơ điện tử thường bao gồm một cơ cấu chấp hành, một vài dạng của bộ truyền lực và tải. Một mô hình bậc bốn có thể mô tả chính xác một hệ như vậy. Hệ số giới hạn hoạt động trong các hệ này chính là tần số cộng hưởng. Sự phối hợp giữa phản hồi vị trí và phản tốc độ (về cơ bản là bộ điều khiển PD) cũng có thể được áp dụng trong trường hợp này. Nhưng do vấn đề điểm cực cộng hưởng nên việc lựa chọn tín hiệu phản hồi phù hợp là cần thiết. Đã có nhiều nghiên cứu (Groenhuis [13]; Coelingh [2]; Coelingh, De Vries, and Van Amerongen [3]) để tìm ra các phương pháp điều chỉnh các hệ như vậy, cũng như để chọn các tín hiệu phản hồi thích hợp. Các công cụ trợ giúp bằng máy tính là cần thiết cho phép người thiết kế ít kinh nghiệm vẫn sử dụng được các phương pháp này (Van Amerongen, Coelingh và De Vries[14]). Coelingh[2] và Coelingh, De Vries và Van Amerongen [3]) mô tả một phương pháp thiết kế cấu trúc cho các hệ cơ điện tử. Phương pháp bắt đầu bằng việc rút gọn thiết kế ý tưởng tới mô hình bậc bốn biểu diễn các tính chất chính của hệ thống theo khối lượng tổng cần di chuyển và độ cứng. Mô hình này vẫn có các tham số vật lý quan trọng. Trong mô hình này, các cảm biến thích hợp được lựa chọn. Trong giai đoạn thiết kế ý tưởng, một bộ điều khiển đơn giản được phát triển và các tính chất cơ có thể thay đổi nếu cần. Tiếp theo là giai đoạn thiết kế chi tiết hơn mà ở đó độ thiếu tin cậy của các tham số cũng được tính đến.

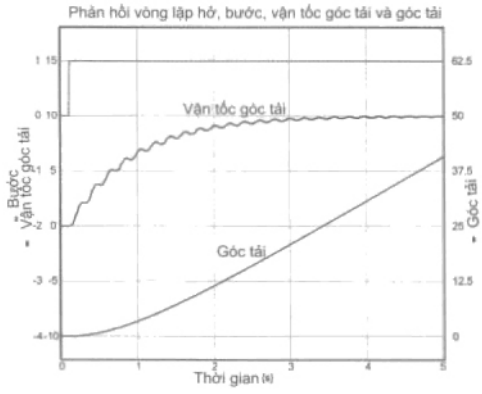
Thiết kế hệ servo

Chúng ta sẽ xem xét một vài khía cạnh đơn giản của thiết kế hệ servo để minh họa ưu điểm của việc sử dụng các mô hình vật lý và sự cần thiết của cách tiếp cận thiết kế tích hợp. Xét mô hình đã

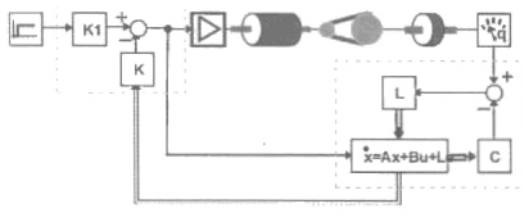
được đề cập ở trên, một tải được điều khiển bởi một động cơ điện qua một bộ truyền lực mềm. Sơ đồ biểu tượng của mô hình này đã được đưa ra trong hình 21.9. Trong ví dụ này, một bộ khuyếch đại dòng đã thay thế bộ khuyếch đại áp cho phép bỏ đi điện trở và điện cảm. Trong các đáp ứng bước ở hình 21.10, có thể thấy rõ sự cộng hưởng do bộ truyền lực mềm.

Từ các phương trình dùng để mô phỏng, 20-sim có thể tự động đưa ra một mô hình ở một dạng thích hợp cho việc thiết kế bộ điều khiển như mô tả không gian-trạng thái, hàm truyền hoặc các điểm cực và các điểm không. Một giao diện cung cấp các tiện ích của Matlab [1], ví dụ dùng các thuật toán của Matlab để tính toán hệ số khuyếch đại của các bộ điều khiển cấp cao như bộ điều khiển LQR (phản hồi trạng thái tối ưu) hoặc bộ điều khiển LQG (với một bộ lọc Kalman để xác định trạng thái và phản hồi trạng thái tối ưu). Sơ đồ quá trình cùng bộ điều khiển LQG được đưa ra trong hình 21.11. Hình 21.12. là một vài đáp ứng của hệ.

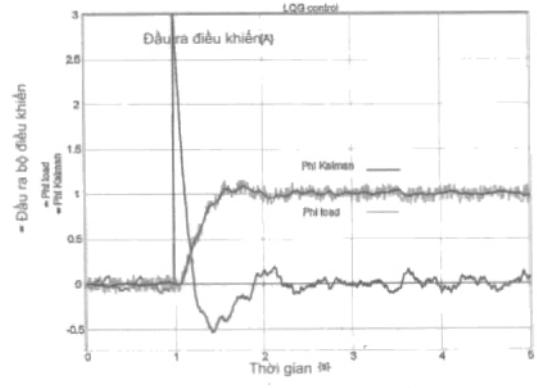
Bộ điều khiển PID được thiết kế thích hợp cũng có thể thực hiện được hầu hết các nhiệm vụ như vậy, đặc biệt là khi nhiễu là nhỏ. Thử nghiệm đầu tiên chỉ sử dụng để đo góc và tốc độ của tải.



HÌNH 21.10 Các đáp ứng vòng hở

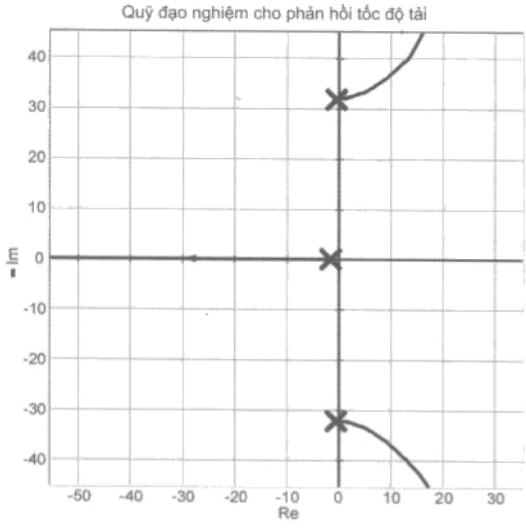


HÌNH 21.11 Thực hiện với bộ lọc Kalman và phản hồi trạng thái



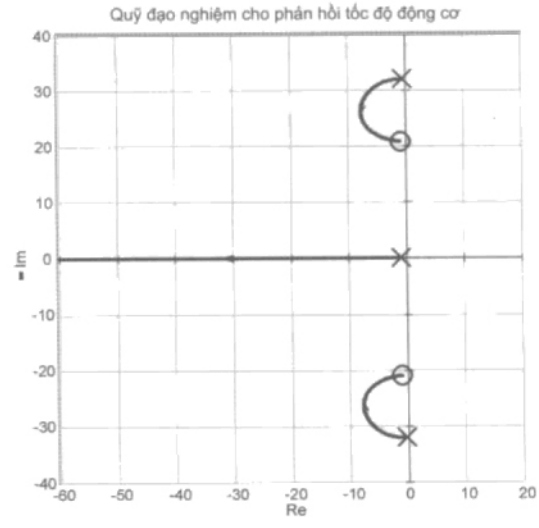
HÌNH 21.12 Đáp ứng của hệ điều khiển LQG

Phép thử này bị lỗi vì phản hồi tốc độ tải hầu như ngay lập tức làm hệ mất ổn định, điều này có thể thấy từ quỹ đạo nghiệm khi cho hệ số phản hồi tốc độ thay đổi. Từ các đáp ứng ở hình 21.10, 20-sim có thể dễ dàng xác định hàm truyền giữa dòng điện của động cơ và tốc độ tải và vẽ quỹ đạo nghiệm đó (hình 21.13).



HÌNH 21.13 Quỹ đạo nghiệm cho trường hợp phản hồi vận tốc của trục tải

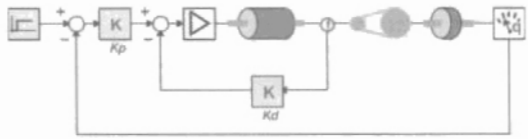
Hình 21.14 chỉ rõ rằng ngay cả một lượng nhỏ phản hồi tốc độ tải cũng dẫn đến một hệ mất ổn định. Một giải pháp tốt hơn là sử dụng phản hồi tốc độ động cơ. Dùng lại mô hình ở hình 21.9 và 21.10 để xác định hàm truyền từ dòng điện vào tới tốc độ động cơ sẽ cho quỹ đạo nghiệm ở hình 21.15.



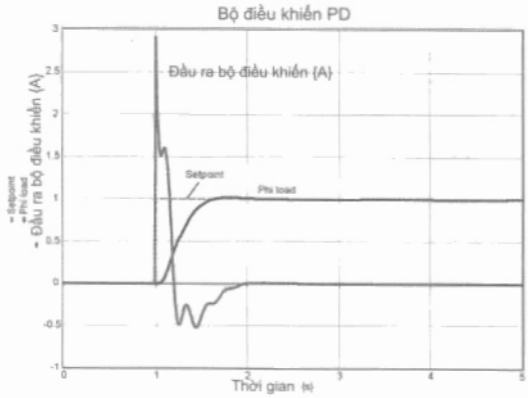
Hình 21.14 Quỹ đạo nghiệm cho trường hợp phản hồi vận tốc của trục động cơ

Bây giờ các điểm không phức luôn đi cùng với các điểm cực phức và bởi vì chúng gần nhau nên ảnh hưởng của chúng lên đáp ứng hầu như không đáng kể. Nhánh của quỹ đạo nghiệm trên trục thực chỉ ra đáp ứng mong muốn: chuyển điểm cực trội sang phía trái của mặt phẳng s. Kết hợp phản hồi tốc độ của động cơ với phản hồi góc của tải sẽ cho cấu trúc bộ điều khiển PD ở hình 21.15 và các đáp ứng ở hình 21.16. Ngoài nhiễu ra, không có sự khác biệt đáng kể so với đáp ứng của hệ có bộ lọc Kalman, mặc dù hệ điều khiển PD đơn giản hơn. Các nhận xét đưa ra ở đây có thể áp dụng một cách tổng quát. Một hệ với hai điểm cực cộng hưởng và không có điểm không như

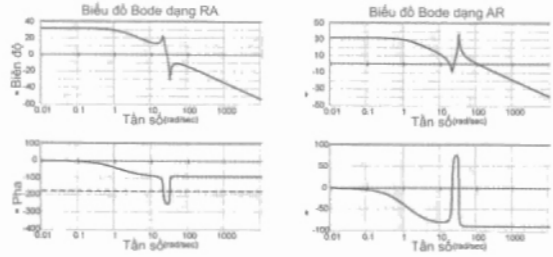
trong hình 21.13 là khó điều khiển bằng một bộ điều khiển đơn giản. Nếu các điểm không phức đi cùng với các điểm cực cộng hưởng có phần ảo nhỏ hơn phần ảo của cực, thì điều khiển ổn định là dễ dàng đạt được. Trong miền tần số, điều này giống như một phản cộng hưởng được theo sau bởi một cộng hưởng (kiểu *AR*). Ngược lại, một hệ kiểu *RA* mà ở đó tần số cộng hưởng thấp hơn tần số phản cộng hưởng (phần ảo của điểm cực nhỏ hơn phần ảo của các điểm không) thì khó điều khiển như trường hợp chỉ có một điểm cực cộng hưởng. Sự tồn tại và vị trí của các điểm không cộng hưởng được xác định (một cách hình học) hoàn toàn bởi vị trí của các bộ cảm biến trong hệ cơ. Vị trí của bộ cảm biến cần được lựa chọn cẩn thận nếu muốn ứng dụng thành công một bộ điều khiển. Cần chú ý rằng việc dùng các bộ phát điểm đặt (Set-Point) được thiết kế thích hợp có thể tránh được sự cộng hưởng như trong hình 21.10. Bộ phát điểm đặt có thể không kích hoạt các tần số cộng hưởng, ví dụ như việc sử dụng bộ lọc thông thấp với giải thông thấp hơn các tần số cộng hưởng. Tuy nhiên, bộ phát điểm đặt như vậy không giải quyết vấn đề ổn định nêu ra ở trên.



HÌNH 21.15 Hệ servo với bộ điều khiển PD



HÌNH 21.16 Các đáp ứng của hệ trong hình 21.15

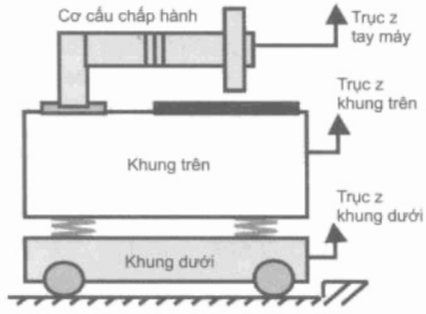


HÌNH 21.17 Các biểu đồ Bode của các hệ kiểu *RA* và kiểu *AR*

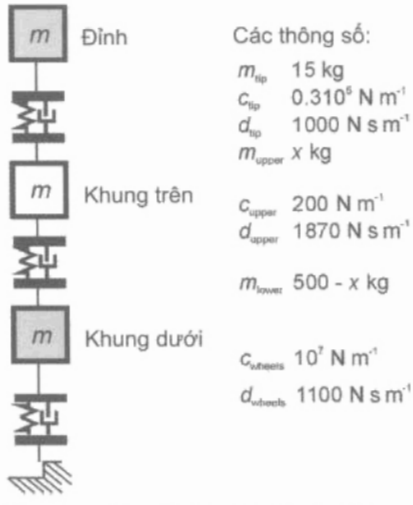
Thiết kế một rôbốt di động

Một ví dụ điển hình về thiết kế giai đoạn đầu là thiết kế ý tưởng của một rôbốt lắp ráp di động. Ngay trong giai đoạn đầu của quá trình thiết kế, các mâu thuẫn đã phải được giải quyết. Một rôbốt như vậy cần phải có khả năng thu thập các bộ phận ở xung quanh dây chuyền sản xuất và lắp ráp trong lúc vận hành. Điều quan trọng là phải đảm bảo cho độ nhấp nhô của mặt

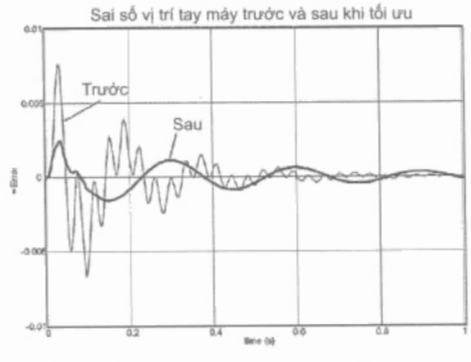
sàn và các kiểu dao động cấu trúc không gây ảnh hưởng đến độ chính xác khi lắp ráp. Mặt khác bộ điều khiển hành trình, một phần dựa trên sự xác định vị trí của rôbôt bằng cách đo vận tốc và hướng của bánh xe, yêu cầu các bánh phải rất cứng. Giảm chấn phải được thực hiện bằng các bộ giảm xóc khác. Điều này dẫn tới ý tưởng khung trên và khung dưới nối với nhau bởi các lò xo (hình 21.18).



HÌNH 21.18 Thiết kế ý tưởng của rôbốt di động



HÌNH 21.19 Mô hình đơn giản với các phần tử vật lý lý tưởng để tính sai số e_{tip}

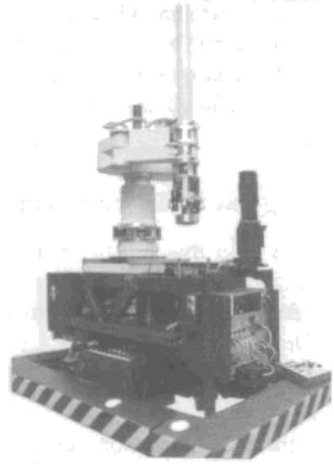


HÌNH 21.20 Các sai số của đầu tay kẹp trước và sau tối ưu hóa phân bố trọng lượng giữa khung trên và khung dưới

Rôbốt có thể được gắn ở khung trên và nên có không gian làm việc sao cho sai số vị trí ($e_{tay\ kẹp}=z_{\text{tay kẹp}}-z_{\text{khung trên}}$) giữa đầu kẹp của rôbốt ($z_{\text{tay kẹp}}$) và khung trên ($z_{\text{khung trên}}$) là đủ nhỏ.

Bước tiếp theo là đưa ra một mô hình đơn giản để tính các tham số cho việc phân bổ trọng lượng, độ cứng và độ giảm chấn của các lò xo. Trong mô hình ở hình 21.22, rôbốt phải chịu sự va chạm mạnh với nền ở tốc độ 1 m/s.

Khối lượng tổng của xe ước tính 500 kg (chủ yếu là trọng lượng của các pin). Độ cứng và độ giảm chấn của các bánh xe được đưa ra do yêu cầu về độ chính xác khi xác định vị trí. Khối lượng và không gian làm việc của tay máy được điều khiển (đã xác định từ các nghiên cứu khác) cho phép xác định độ cứng hiệu dụng và độ giảm chấn của đầu tay kẹp. Khi tính toán ban đầu về độ cứng và độ giảm chấn của các lò xo giữa khung trên và khung dưới đã được thực hiện thì chỉ một tham số duy nhất cần thay đổi là phân bố trọng lượng giữa khung trên và khung dưới. Bằng cách dùng tính năng tối ưu hóa của chương trình 20-sim, thì việc phân bố trọng lượng tối ưu có thể dễ dàng tìm được. Để tối thiểu hóa sai số giữa đầu tay kẹp và khung trên (hình 21.19) thì trọng lượng phải được đặt nhiều nhất có thể ở khung trên (hình 21.20). Ví dụ này minh họa cách xác định một kết cấu cơ học thông qua các yêu cầu của bộ điều khiển hành trình và độ chính xác của việc lắp ráp.



HÌNH 21.21 Rôbốt di động (MART) sau khi hoàn thiện

Bước tiếp theo có thể là tối ưu hóa các đặc tính của hệ thống giảm xóc giữa khung trên và khung dưới. Điều này nhằm cải thiện độ chính xác. Quyết định này được thực hiện ngay trong giai đoạn thiết kế ban đầu để định hướng các quyết định thiết kế khác. Sau khi hoàn thành, các tham số khác nhau của cấu trúc sẽ gần với các thông số đã được ước tính ban đầu (hình 21.21).

21.4 Các ví dụ hệ cơ điện tử hiện đại

Một vài ví dụ đã được xem xét trong các mục trước. Trong mục này sẽ đưa ra hai ví dụ nữa.

Ổn định sự lắc lư và bánh lái của tàu thủy

Ngày nay, các tàu thủy dùng một máy lái tự động để điều khiển hướng tàu. Bánh lái là cơ cấu chấp hành thông dụng nhất. Một số phương tiện như phà và các tàu hải quân cũng cần phải ổn định sự lắc lư tròng trành của tàu. Điều này có thể được thực hiện một cách bị động bằng cách nối các thùng đầy nước để tạo ra các lực ổn định ngược pha với các lực của sóng. Để tạo ra hệ thống phù hợp với các tần số khác nhau của các sóng, dòng nước giữa hai thùng chứa cần phải được điều khiển. Các tàu nhanh thường sử dụng các vây ổn định. Chúng là một loại "cánh" được điều khiển một cách chủ động để tạo ra mômen cần thiết chống lại mômen của sóng. Các vây không chỉ tác động tới sự lắc lư mà còn ảnh hưởng tới cả hướng tàu. Mặt khác, bánh lái không chỉ ảnh hưởng tới hướng tàu mà còn gây ra sự lắc lư. Trong kỹ thuật điều khiển, điều này dẫn đến một hệ đa biến đòi

hỏi phải thiết kế một bộ điều khiển đa biến để thực hiện tối ưu. Trong thực tế, một hệ đa biến như vậy hiếm khi thấy mà thường sử dụng hai hệ điều khiển riêng biệt.

Một phương pháp khác là chỉ dùng một trong các cơ cấu chấp hành kể trên (bánh lái hoặc vây) để điều khiển hành trình và giảm chuyển động quay. Vì các tần số của chuyển động lắc lư nằm ngoài dải tần làm việc của hệ thống điều khiển hành trình nên điều này là có thể. Bánh lái là cơ cấu phù hợp nhất. Ngoài ra, một thuận lợi cho tàu hải quân là bỏ các vây sẽ giảm độ ồn dưới nước.

Thiết kế lại các bộ điều khiển hành trình để ổn định sự lắc lư đồng thời chứng minh tính khả thi của phương pháp này, nhưng cũng làm sáng tỏ rằng tàu thủy cần phải được sửa đổi. Sự sửa đổi quan trọng nhất là cho máy lái. Tốc độ tối đa của máy lái đưa ra là hệ số giới hạn cho một hệ như vậy (nó phải được tăng từ các giá trị thông dụng từ 3–7°/s tới 20–25°/s). Bằng cách mô phỏng động lực, các yêu cầu cho máy lái có thể được xác định theo tốc độ cực đại của máy lái và thời gian cực đại cho phép để đạt được tốc độ này. Điều này đòi hỏi phải thiết kế lại máy lái thủy lực. Bước xa hơn là xem xét việc thay đổi hình dạng của tàu để tối ưu các tham số xác định hiệu quả của hệ thống ổn định sự lắc lư và bánh lái.

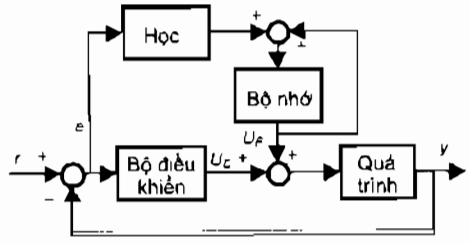
Để khẳng định giải pháp mới này có tốt hơn hay không, cần phải kiểm tra xem máy lái đã được thiết kế lại có giá thành thấp hơn hệ thống cũ (bánh lái và vây) hay không. Các vấn đề này phải được giải quyết ngay trong giai đoạn đầu của quá trình thiết kế. Ổn định sự lắc lư và bánh lái đã được áp dụng thành công trên các tàu hải quân cũng như các thương thuyền lớn (Van Amerongen, Van der Klugt, and Van Nauta Lemke [15]).

Bù các hiệu ứng phi tuyến trong các động cơ tuyến tính

Nhiều hệ cơ học do ảnh hưởng của hiệu ứng phi tuyến mà hạn chế độ chính xác có thể đạt được. Ma sát và sự ăn khớp là hai ví dụ. Một bộ điều khiển phản hồi tuyến tính có thể giảm ảnh hưởng của các tính chất phi tuyến nhưng rất khó bù hoàn toàn. Với các hệ thực hiện các dịch chuyển lặp đi lặp lại, một *bộ điều khiển học lặp* (Iterative Learning Controllor) có thể giúp cải thiện hoạt động (Arimoto [16]; De Vries, Velthuis, and Van Amerongen [17]). Ý tưởng cơ bản được giải thích trong hình 21.22.

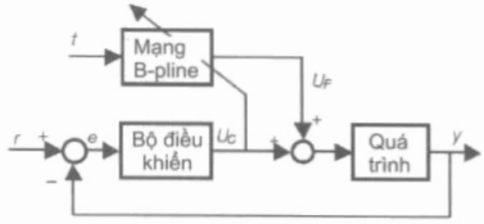
Chỉ khi có vòng phản hồi và với giả thiết rằng không có nhiễu, thì các sai số và tín hiệu điều khiển U_C sẽ giống nhau trong mỗi dịch chuyển lặp lại. Hiển nhiên rằng độ chính xác có thể được cải thiện khi trong dịch chuyển tiếp theo tín hiệu của bộ điều khiển (U_F) từ chu kỳ trước được dùng như tín hiệu *cấp thẳng* (feed-forward). Phản hồi sẽ tạo ra một tín hiệu bù sai số còn lại bằng cách cập nhật tín hiệu cấp thẳng U_F theo công thức:

$$U_F^{k+1} = U_F^k + LE^k \tag{21.8}$$



HÌNH 21.22 Nguyên tắc của điều khiển học lặp

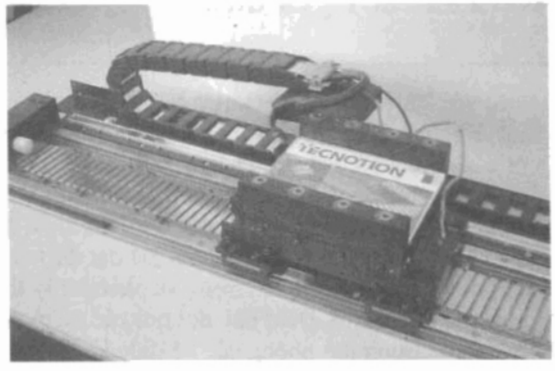
Trong đó L là hàm truyền của bộ lọc tự học. Chỉ số k kí hiệu dịch chuyển lặp thứ k. Tín hiệu U_F cần phải hội tụ tới tín hiệu cấp thẳng để bù tất cả các sai số lặp. Một ví dụ điển hình về các sai số như vậy là máy đọc CD được bù độ lệch tâm của đĩa. Một dạng khác của ý tưởng này là sử dụng bộ điều khiển LFFC (Learning Feed-Forward Controller) như trên hình 21.23.



HÌNH 21.23 Bộ điều khiển LFFC cho các dịch chuyển lặp

Khi tín hiệu cấp thẳng là hoàn hảo, tín hiệu ra của bộ điều khiển có thể bằng không. Điều này có nghĩa là tín hiệu ra có thể được dùng làm tín hiệu huấn luyện cho một mạng nơron. Một mạng B-spline thích nghi cho phép học các tính chất phi tuyến phức tạp. Tín hiệu vào của mạng B-spline là tín hiệu thời gian *t*. Nó được reset khi một dịch chuyển mới được bắt đầu. Bộ điều khiển này được gọi là LFFC *chỉ số thời gian*. Thay vì sử dụng thời gian, tín hiệu tham chiếu và các đạo hàm của nó có thể được dùng như làm chỉ số của mạng (LFFC *chỉ số dẫn*). Ưu điểm của cấu trúc này là sau khi huấn luyện, LFFC cũng có thể được ứng dụng thành công cho các dịch chuyển không lặp. Velthuis [19] đã đưa ra phân tích ổn định cho các bộ điều khiển LFFC chỉ số thời gian cũng như LFFC chỉ số dẫn. Phân tích ổn định là tương đối dễ cho trường hợp chỉ số thời gian. Với trường hợp chỉ số dẫn thì phức tạp hơn và cần phải đưa ra một số phương pháp thích hợp để đảm bảo hệ thống ổn định. Vấn đề chính là số lượng các B-spline không được quá lớn. Mặt khác, cần phải có sự phân bố B-spline thích hợp để có được xấp xỉ chính xác quá trình phi tuyến. LFFC đã được áp dụng để bù sự ăn khớp cho động cơ tuyến tính công nghiệp (Otten và cộng sự [20]) và bù ma sát (Coulomb) của động cơ tuyến tính dùng trong mô phỏng bay (Velthuis [19]). Nó cũng được áp dụng để điều khiển bám cho rôbốt di động mô tả trong mục Thiết kế rôbốt di động (Starrenburg và cộng sự [21]).

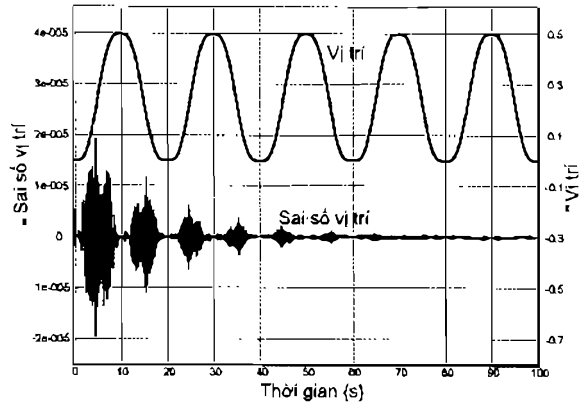
Ứng dụng bù ăn khớp của động cơ tuyến tính sẽ được mô tả chi tiết hơn. Một động cơ như vậy thường được sử dụng trong các máy lắp ráp. Thậm chí với các nam châm tốt nhất và lắp ráp chính xác thì sai số cũng không thể nhỏ hơn 100μ, với một bộ điều khiển PID phối hợp với điều khiển NLFFC (Nonlearning Feed-Forward Controller). Mục tiêu thiết kế là để cải thiện độ chính xác tối đa có thể đạt được từ 100 μ tới nhỏ hơn 10 μ. Hình 21.24 là một đông cơ tuyến tính.



HÌNH 21.24 Động cơ tuyến tính

Theo cấu trúc trên hình 21.23, động cơ tuyến tính được điều khiển bằng bộ điều khiển PID, trong khi mạng nơron B-spline được dùng để học mô hình động cơ đảo chiều, bao gồm cả thành phần phi tuyến do ăn khớp. Ăn khớp xảy ra trong các động cơ DC với các nam châm vĩnh cửu. Nó gây ra các lực dạng sin lớn hoặc nhỏ phụ thuộc vào vị trí của bộ chuyển đổi so với stato. Nếu các lực đó thực sự có dạng sin thì chúng có thể dễ dàng được bù bằng một bộ bù cấp thuận. Tuy nhiên điều này đòi hỏi các nam châm có các tính chất từ giống nhau hoàn toàn và khoảng cách của các nam châm phải rất chính xác. Một giải pháp khác là thiết kế một bộ điều khiển để học các kiểu nhiễu và bù nó bằng một bộ bù NLFF. Một hệ như vậy cũng có thể được dùng để bù cho các hiệu

ứng phi tuyến khác, ví dụ như ma sát. Điều này cũng đã được chứng thực trong bộ mô phỏng bay ở đó các lực ma sát làm giảm cảm giác của mô phỏng thực, đặc biệt ở tốc độ bằng không. Hình 21.25 chỉ ra rằng việc học gần như được hoàn thành sau sáu chu kỳ huấn luyện.



HÌNH 21.25 Tín hiệu vị trí và tín hiệu sai số trong quá trình học của LFFC

Điều khiển LFF là một phương pháp hấp dẫn để bù các phi tuyến tồn tại trong các hệ cơ điện tử, như sự ăn khớp và ma sát. Việc dùng mạng nơron B-spline sẽ dẫn đến hội tụ nhanh, tính toán tương đối ít và khả năng tổng quát hóa tốt. Với các kết quả thu được gần đây đối với độ ổn định của các hệ thống như vậy, các hệ điều khiển bền vững có thể được thiết kế.

Trên quan điểm cơ điện tử về vấn đề thiết kế này, một câu hỏi đặt ra là có thể dùng chính phương pháp này để xây dựng một động cơ tuyến tính rẻ hơn không, khi độ chính xác tối đa không phải là mục tiêu chính của thiết kế. Người ta đã chứng minh rằng một động cơ được xây dựng với các bộ phận có giá thành thấp và các yêu cầu lắp ráp không khắt khe nhưng với LFFC có thể cạnh tranh tốt với cấu trúc đắt tiền hơn. Độ chính xác có thể được cải thiện thường với hệ số là 10.

21.5 Yêu cầu đặc biệt của các hệ cơ điện tử so với các hệ "cổ điển" và việc thiết kế

Thiết kế hệ cơ điện tử khác với "thiết kế bộ điều khiển thông thường" ở chỗ thiết kế cơ điện tử là thiết kế một hệ thống nhất. Phương pháp này có thể được xem như việc tối ưu hóa tất cả các thành phần của hệ thống một cách đồng bộ, mặc dù không có thuật toán để làm điều này một cách tự động. Trong thực tế, vấn đề này thường được tách thành các vấn đề nhỏ hơn để có thể tối ưu hóa. Sau khi tích hợp tất cả các giải pháp thành phần, một hệ tối ưu khác đạt được và có thể được tiếp tục tối ưu bằng cách điều chỉnh lại các phần khác nhau. Để đạt được sự tối ưu của toàn bộ hệ thống, đòi hỏi phần cơ mà ở đó lực đóng vai trò quan trọng và phần xử lý thông tin (bộ điều khiển) phải được mô phỏng và hiệu chỉnh đồng thời. Điều này đòi hỏi các tham số cơ như khối lượng và các thông số khác có thể dùng được trong mô phỏng các hệ điều khiển. Các ví dụ đã được đưa ra về mô hình hóa và mô phỏng bằng 20-sim chứng tỏ 20-sim cho phép thực hiện như vậy.

Những người thiết kế cơ điện tử phải luôn nhận thức được vấn đề là các giải pháp có thể được tìm thấy ở các lĩnh vực khác nhau. Không phải bất cứ sự thiếu hiệu quả nào của phần cơ đều có thể giải quyết dễ dàng bằng điều khiển. Một thiết kế cơ khí tốt có thể thực hiện dễ dàng hơn và rẻ hơn. Mặt khác, một bộ điều khiển tốt có thể đạt được kết quả mong muốn rẻ hơn và dễ hơn nhiều so với những cấu trúc cơ phức tạp. Trong một số trường hợp, sự kết hợp có thể đạt được kết quả mà sẽ không thể có nếu thiếu thiết kế cơ điện tử.

Điều đó cũng tương tự đối với việc thiết kế các bộ cảm biến. Mỗi bộ cảm biến có thể được lắp một bộ lọc để loại nhiễu khỏi các phép đo. Nhưng nếu vài bộ cảm biến được tổ hợp thì tổ hợp của các cảm biến theo thuật toán lọc Kalman sẽ có hiệu quả với dữ liệu thô.

Việc thống nhất giữa các nhà thiết kế trong các lĩnh vực khác nhau để đưa ra được các quyết định rõ ràng là cần thiết cho sự thành công của một thiết kế cơ điện tử.

Tài liệu tham khảo

[1] Matworks (2001). The Mathworks: Developers of Matlab and Simulink, www.mathworks.com

[2] Coelingh, H.J., *Design Support for Motion Control Systems,* Ph.D. thesis, University of Twente, 2000, also www.rt.el.utwente.nl/clh/

[3] Coelingh, H.J., de Vries, T.J.A., van Amerongen, J., Design support for motion control systems—application to the Philips fast component mounter, in *Mechatronics Forum 7th Int. Conf., Mechatronics 2000,* Atlanta, Ga, USA.

[4] Controllab Products, *20-sim*, www.20sim.com

[5] Broenink, J.F., Computer-Aided Physical-Systems Modeling and Simulation: a Bond-Graph Approach, Ph.D. thesis, University of Twente, 1990.

[6] Dynasim, *Dymola,* www.dynasim.se/

[7] Weustink, P.B.T., de Vries, T.J.A., Breedveld, P.C., *Object Oriented Modeling and Simulation of Mechatronic Systems with 20-sim 3.0,* in *Mechatronics 98,* J. Adolfson and J. Karlsén (Eds.), Elsevier Science, 1998.

[8] De Vries, T.J.A., Conceptual Design of Controlled Electro-Mechanical Systems, Ph.D. thesis, University of Twente, 1994.

[9] Breedveld, P.C., Fundamentals of bond graphs, in IMACS Annals of Computing and Applied Mathematics, Vol. 3: Modelling and Simulation of Systems, Basel, 1989, pp. 7–14.

[10] Cellier, F.E., Elmqvist, H., Otter, M., Modeling from physical principles, in *The Control Handbook,* W.S. Levine (Ed.), CRC Press, pp. 99–108, 1996.

[11] Gawthrop, P., Lorcan Smith, L., *Metamodelling: Bond Graphs and Dynamic Systems,* Prentice-Hall, NJ, 1996.

[12] Van Amerongen, J., Modelling, simulation and controller design for mechatronic systems with 20-sim 3.0, in *Proc. 1st IFAC Conf. on Mechatronic Systems,* Darmstadt, Germany, September 2000, pp. 831–836.

[13] Groenhuis, H., *A Design Tool for Electromechanical Servo Systems,* Ph.D. thesis, University of Twente, 1991.

[14] Van Amerongen, J., Coelingh, H.J., de Vries, T.J.A., "Computer support for mechatronic control system design," *Robotics and Autonomous Systems,* vol. 30, no. 3, pp. 249–260, PII: SO921-8890 (99)00090-1, 2000.

[15] Van Amerongen, J., van der Klugt, P.G.M., van Nauta Lemke, H.R., Rudder roll stabilization for ships, *Automatica,* vol. 26, no. 4, pp. 679–690.

[16] Arimoto, S., A brief history of iterative learning control, in *Iterative Learning Control: Analysis, Design, Integration and Applications,* Kluwer Academic Publishers, pp. 3–7, 1988.

[17] De Vries, T.J.A., Velthuis, W.J.R., van Amerongen, J., Learning feed-forward control: a survey and historical note, in *1st IFAC Conf. on Mechatronic Systems,* Darmstadt, Germany, September 2000, pp. 949–954.

[18] De Kruif, Bas J., de Vries, T.J.A., On using a support vector machine in learning feed-forward control, in *Proc. 2001 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics,* Como, Italy, 8–12 July, 2001.

[19] Velthuis, W.J.*R., Learning Feed-Forward Control—Theory, Design and Applications,* Ph.D. thesis, University of Twente, 2000, also http://www.rt.el.utwente.nl/vts/

[20] Otten, G., de Vries, T.J.A., van Amerongen, J., Rankers, A.M., Gaal, E., Linear motor motion control using a learning feedforward controller, *IEEE/ASME Transactions on Mechatronics,* vol. 2, no. 3, ISSN.

[21] 1083-4435, pp. 179–187, 1997.

[22] Starrenburg, J.G., van Luenen, W.T.C., Oelen, W., van Amerongen, J., Learning feed-forward controller for a mobile robot vehicle, *Control Engineering Practice,* vol. 4, no. 9, pp. 1221–1230, 1996.

22

Vai trò của mô hình hóa trong thiết kế cơ điện tử

22.1 Mô hình hóa là một phần của quá trình thiết kế.....22-1
22.2 Mục đích của mô hình hóa.....22-6
22.3 Mô hình hoá các hệ thống và các tín hiệu.....22-9

Jeffrey A. Jalkio
University of St Thomas

Thiết kế cơ điện tử không đơn thuần là việc kết hợp các thiết kế điện tử, phần mềm và cơ khí. Người thiết kế cơ điện tử phải có khả năng tối ưu hóa giải pháp thiết kế theo các lĩnh vực khác nhau đó. Điều này đòi hỏi sự hiểu biết đầy đủ về mỗi lĩnh vực để có thể xác định được những phần nào của một vấn đề kỹ thuật sẽ được giải quyết tốt nhất ở mỗi lĩnh vực đó trong tình trạng công nghệ hiện thời. Tiếp theo, điều này đòi hỏi khả năng mô hình hóa vấn đề đã đặt ra và các giải pháp khả thi áp dụng các kỹ thuật ở các lĩnh vực độc lập hoặc ít nhất cũng cho phép dễ dàng so sánh các giải pháp và các công cụ của các lĩnh vực khác nhau.

Ví dụ hệ kiểm tra quang học được chỉ ra trong hình 22.1 phụ thuộc vào việc sắp xếp các thành phần quang học, các phần tử cơ có khả năng dịch chuyển chính xác, các bộ chuyển đổi, các hệ điện tử điều khiển dịch chuyển và lọc các tín hiệu cảm biến, phần mềm để phân tích ảnh và điều khiển dịch chuyển. Chỉ bằng cách phân chia các nhiệm vụ đó một cách thích hợp giữa điện tử, các thành phần cơ và phần mềm thì hệ thống mới có thể được tối ưu hóa. Điều này đòi hỏi sự hiểu biết tất cả các yêu cầu của hệ thống và các giới hạn cũng như các khả năng của mỗi thành phần trong các lĩnh vực khác nhau. Việc mô hình hóa các yêu cầu và các hệ thống là quan trọng để xác định lời giải nào có thể được chấp nhận cũng như để chứng minh các quyết định đó là đúng đắn. Trong mục này, chúng ta sẽ khảo sát các dạng mô hình được dùng ở các thời điểm khác nhau trong quá trình thiết kế, vai trò của các mô hình đó, mặt mạnh và yếu của chúng trong mỗi vai trò đó và cuối cùng là lựa chọn cụ thể các mô hình động học để phân tích tín hiệu và hệ thống.

22.1 Mô hình hóa là một phần của quá trình thiết kế

Các mô hình phục vụ các mục đích khác nhau tại các thời điểm khác nhau trong quá trình thiết kế; vì vậy để xác định các công cụ mô hình hóa nào là hiệu quả nhất trong các giai đoạn khác nhau, chúng ta cần phải khảo sát chính quá trình thiết kế. Các nhà nghiên cứu trên thế giới đã phát triển nhiều mô tả quá trình thiết kế [1-3]. Thông thường các mô tả đó dùng để hệ thống hóa quá trình nhằm cải thiện năng suất thiết kế hoặc để mô tả các kỹ thuật nhằm tạo ra các sản phẩm với chất lượng được cải thiện, giá thấp hơn hoặc các lợi ích khác. Tuy nhiên vì mục đích của chúng ta là khảo sát sự cần thiết của việc mô hình hóa quá trình thiết kế, nên chúng ta chỉ xét một mô hình đơn giản để phân biệt các giai đoạn của quá trình thiết kế theo các hoạt động thiết kế chứ không phải là một mô hình phức tạp hơn phù hợp với các mục đích khác. Với mục đích này, chúng ta có thể xét một quá trình bốn giai đoạn bao gồm phân tích các yêu cầu, đưa ra các ý tưởng, phân tích, lựa chọn và thiết kế chi tiết. Trong giai đoạn đầu của quá trình này người thiết kế tập trung phân tích vấn đề mà chưa cần xét tới các giải pháp. Trong giai đoạn thứ hai, các giải pháp ý tưởng được

tạo ra với hy vọng có thể tìm thấy một giải pháp có thể được chấp nhận từ các ý tưởng ban đầu nhờ việc tổng hợp hoặc sửa đổi các ý tưởng hoặc bằng cách thay đổi các tham số hiện thời của một trong những giải pháp ý tưởng đó. Trong giai đoạn thứ ba, các ý tưởng đó được đánh giá và một thiết kế được chọn để thực hiện. Giai đoạn thứ tư gồm việc xác định các vấn đề thiết kế cần được giải quyết để thực hiện các ý tưởng đã chọn và áp dụng quá trình thiết kế vào những vấn đề nhỏ hơn. Chúng ta sẽ xem xét hoạt động của mỗi giai đoạn một cách chi tiết.

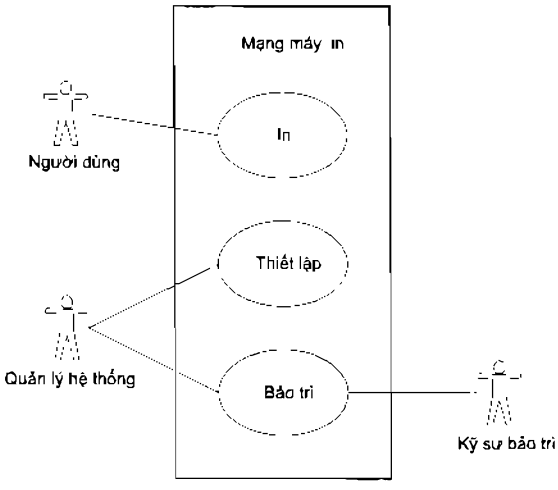


HÌNH 22.1 Hệ thống kiểm tra quang học cho các bảng mạch in
(Được đăng ký bởi Tập đoàn Cyber Optics, 2001)

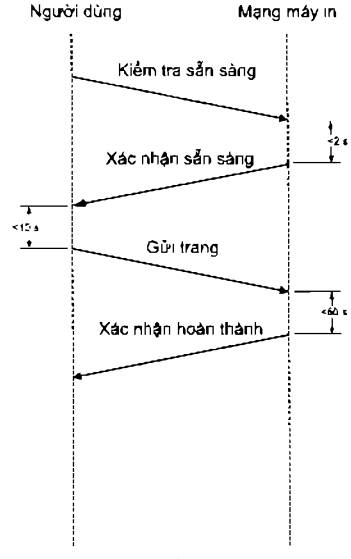
Giai đoạn 1

Giai đoạn phân tích các yêu cầu nhằm hiểu biết đầy đủ về vấn đề cần được giải quyết. Độ phức tạp của quá trình này tuỳ thuộc vào phạm vi của vấn đề, sự hiểu biết của người thiết kế về lĩnh vực của vấn đề, sự thay đổi của thị trường và sự xuất hiện của các yêu cầu mới ít được chú ý đến trong giai đoạn đặt vấn đề. Do phụ thuộc vào bản chất của vấn đề thiết kế, nên các yêu cầu được chỉ ra trong giai đoạn này có thể là nhu cầu của một khách hàng đơn lẻ, nhu cầu chung của một nhóm khách hàng tiềm năng được xác định qua nghiên cứu thị trường hoặc nhu cầu chung của xã hội được xác định bởi sự điều chỉnh của chính phủ. Hầu hết các vấn đề thiết kế đều bao gồm một vài kết hợp của các nhu cầu đó cũng như các yêu cầu nội bộ như nguyên tắc thiết kế, đường lối của công ty. Mục tiêu then chốt của giai đoạn này là có được đầy đủ thông tin để xác định khi nào một thiết kế đã giải quyết trọn vẹn vấn đề. Các mô hình trong giai đoạn này trước tiên phục vụ như là các công cụ dẫn chứng và truyền thông, vì vấn đề chính trong giai đoạn này là chứng minh và truyền đạt rõ ràng các tiêu chuẩn cho sự thành công của thiết kế. Ví dụ về các mô hình trong quá trình này bao gồm các bản liệt kê đặc tính kỹ thuật, sơ đồ các trường hợp sử dụng, các sơ đồ tuần tự và các sơ đồ tình huống. Hiện nay nhiều công cụ mô hình hóa đã được chuẩn hóa như là các phần của ngôn ngữ mô hình hóa hợp nhất UML (the Unified Modeling Language). UML là một công cụ phân tích quan trọng [4].

Sơ đồ trường hợp sử dụng dùng để mô hình hóa sự tương tác giữa hệ thống và người sử dụng ở mức trừu tượng hóa rất cao theo mục tiêu của tương tác đó. Hình 22.2 là một ví dụ về sơ đồ trường hợp sử dụng được dùng để thể hiện các hoạt động khác nhau cần phải có của một máy in mạng. Sơ đồ trường hợp sử dụng giúp chúng ta tránh việc bỏ sót những trường hợp quan trọng nhưng hiếm gặp như bảo dưỡng. Phải đặc biệt chú ý rằng các sơ đồ trường hợp sử dụng không đưa ra thông tin về bản chất của cách giải quyết, mà chỉ chứng tỏ sự tồn tại của nó. Nó giống như một mô hình mức cao nhất của một hệ thống.

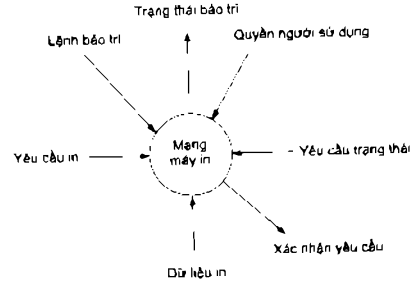


HÌNH 22.2 Sơ đồ trường hợp sử dụng cho một máy in mạng

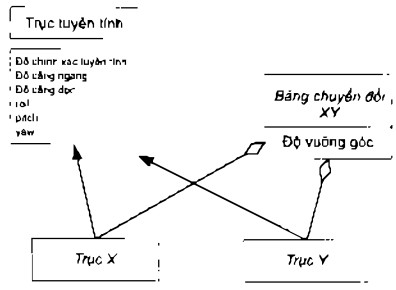


HÌNH 22.3 Sơ đồ tuần tự cho máy in mạng

Sơ đồ tuần tự dùng để mô tả chi tiết các trường hợp riêng biệt nhằm làm rõ các tương tác phải xảy ra, các đòi hỏi về thời gian, và tương tác giữa hệ và các yếu tố của môi trường. Hình 22.3 chỉ ra một ví dụ về sơ đồ tuần tự cho trường hợp dùng máy in ở hình 22.2.



HÌNH 22.4 Sơ đồ tình huống cho máy in mạng



HÌNH 22.5 Sơ đồ lớp cho các giai đoạn truyền tuyến tính

Các sơ đồ tình huống không kiểm soát các yếu tố thời gian nhưng kiểm soát các kiểu dữ liệu truyền thông giữa hệ thống và môi trường của nó [5]. Trong khi các sơ đồ trường hợp sử dụng tập trung vào các phương án sử dụng hệ thống thì sơ đồ tình huống tập trung chú ý vào các luồng thông tin phải tồn tại để tạo ra các phương án đó. Hình 22.4 chỉ ra một sơ đồ tình huống cho hệ kiểm tra giống như ví dụ đã nêu ở trên. Chú ý rằng sơ đồ tình huống này tóm tắt luồng thông tin trong các sơ đồ tương tác cho tất cả các trường hợp sử dụng. Các yêu cầu điều chỉnh và an toàn thường là giới hạn của các tương tác mà hệ có thể có với môi trường của nó. Các yêu cầu này phải được giữ để dùng trong các giai đoạn thiết kế sau này và phải được truyền thông ngược tới điểm xuất phát của các yêu cầu đó để kiểm tra độ chính xác. Sơ đồ tình huống cũng được dùng như mức cao nhất của sơ đồ luồng dữ liệu cho một hệ thống khi sử dụng phân tích cấu trúc [6].

Giống như sơ đồ tình huống liên quan chặt chẽ với các kỹ thuật phân tích cấu trúc, các kỹ thuật phân tích hướng đối tượng cung cấp sơ đồ quan hệ lớp như là công cụ thể hiện các mối quan hệ giữa một hệ với môi trường của nó và giữa các thành phần của hệ [7]. Các sơ đồ quan hệ lớp chỉ ra một hệ được cấu thành từ các hệ con như thế nào, các thành phần giống và khác nhau như thế nào. Ví dụ hình 22.5 chỉ ra một sơ đồ quan hệ lớp cho một hệ truyền hai trục bao gồm hai hệ con đơn trục. Sơ đồ này chứng minh sự tồn tại của 13 thành phần sai số phải được chỉ ra trong giai đoạn phân tích các yêu cầu. Điều này thực hiện bằng cách chỉ ra rằng các thành phần trục X và Y là các lớp ví dụ của các giai đoạn truyền trục đơn, mà mỗi giai đoạn có sáu thành phần sai số và chúng là các thành phần của một hệ thống nên phải cộng thêm một sai số đơn. Bằng việc chứng tỏ các quan hệ như là sự tổng hợp và không phải là sự kế thừa, các yêu cầu và sự phụ thuộc qua lại của các yêu cầu có thể được trình bày theo cách ngắn gọn và dễ hiểu.

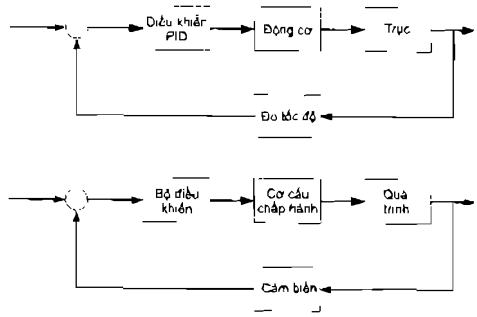
Một vấn đề chính của giai đoạn xác định các yêu cầu là việc xác định các yêu cầu được thực hiện mà không cần chỉ ra một giải pháp thích hợp. Vì vậy các phương pháp mô hình hoá cần phải được chọn để thể hiện các yêu cầu đó và các quan hệ bản chất của chúng mà không cần dẫn ra một giải pháp cụ thể.

Giai đoạn 2

Trong giai đoạn tạo ý tưởng, mục đích của chúng ta là tạo ra nhiều ý tưởng thiết kế để có thể thỏa mãn các yêu cầu đã được chỉ ra trong giai đoạn 1. Ở đây chúng ta cần các phương pháp mô hình hóa cho phép chúng ta mô tả các giải pháp có thể với các mức độ chi tiết khác nhau phụ thuộc vào độ chi tiết cần thiết để thể hiện các yếu tố then chốt của ý tưởng. Vì các ý tưởng riêng biệt được tạo ra trong giai đoạn này có thể chỉ thỏa mãn một phần của yêu cầu thiết kế, nên điều quan trọng là việc mô hình hóa ở thời điểm này phải cho phép mô tả từng phần các ý tưởng và cho phép kết hợp dễ dàng các ý tưởng thiết kế. Với lý do này, các mô hình của chúng ta phải thể hiện rõ ràng các phần của các yêu cầu đã được thỏa mãn cũng như bất kỳ tham số chưa được chỉ ra hoặc các yêu cầu bổ sung đã được giới thiệu trong ý tưởng. Với một số vấn đề, sơ đồ khối chỉ ra các mối liên kết bên trong giữa các thành phần dùng để giải quyết từng phần nhỏ của vấn đề, là một công cụ mô hình hóa hữu ích ở giai đoạn này. Hình 22.6 chỉ ra hai sơ đồ khối có thể mô tả một ý tưởng thiết kế đã được đưa ra. Sơ đồ đầu chỉ ra một thuật toán điều khiển cụ thể, cảm biến và cơ

cấu chấp hành, trong khi ở sơ đồ thứ hai các chi tiết đó không được đưa ra và chỉ mô tả đơn giản một bộ điều khiển vòng kín. Sơ đồ nào là một mô tả thích hợp còn tùy thuộc vào những tình huống cụ thể. Sơ đồ đầu đưa ra các chi tiết và gần với một thiết kế đầy đủ, trong khi sơ đồ thứ hai tổng quát hơn, dễ tổ hợp hơn với các ý tưởng khác để tạo ra các giải pháp lai.

Các sơ đồ khối không phải là công cụ mô hình hóa thích hợp duy nhất ở giai đoạn này. Với các vấn đề khác, ta có thể sử dụng các giản đồ để chỉ ra cách sắp xếp các thành phần, các phương trình hoặc các thuật toán đã được đưa ra. Tiếp theo, các ý tưởng thiết kế được kết hợp với nhau để tạo ra các giải pháp khả thi cho vấn đề thiết kế tổng thể. Vì vậy, sẽ là hữu ích nếu mô hình của mỗi ý tưởng có thể bao gồm mô tả các điều kiện tiên quyết cho việc sử dụng chúng, các kết quả và các tham số khác để giúp nhóm thiết kế xác định cách phối hợp các ý tưởng. Tương tự, khi các giải pháp khả thi được tạo ra bằng cách liên kết các ý tưởng với nhau, cần phải rút gọn ý tưởng cuối cùng bằng cách kết hợp các đặc tính từ các ý tưởng thành phần khác nhau hoặc loại bỏ các đặc tính chồng chéo không cần thiết [8]. Quá trình này được thực hiện dễ dàng bằng các phương pháp mô hình hóa cho phép mô hình hóa đồng thời các phần cơ khí, nhiệt, điện, phần mềm và các ý tưởng. Các phương pháp này bao gồm các mô hình đồ thị tuyến tính thông thường và đồ thị liên kết [9, 10].



HÌNH 22.6 Các sơ đồ khối ở hai mức độ chi tiết khác nhau

Giai đoạn 3

Trong giai đoạn thứ ba, chúng ta phải đánh giá các giải pháp tiềm năng theo các yêu cầu của vấn đề. Giai đoạn này sẽ đơn giản hơn nếu chúng ta có một mô hình cho phép so sánh các yêu cầu của vấn đề với các đặc tính của thiết kế. Một phương pháp luận thiết kế cụ thể có thể bao gồm các tiêu chuẩn chất lượng khác nhau cộng với các yêu cầu của khách hàng và môi trường, ví dụ như phương châm thiết kế của Suh hướng tới các thiết kế với nội dung thông tin tối thiểu và thỏa mãn mỗi yêu cầu bằng các đặc tính độc lập của thiết kế [11].

Một phương pháp để đánh giá là thử tìm ra các tiêu chuẩn số cho tất cả các yêu cầu và đánh giá các giải pháp bằng việc tối thiểu hóa một hàm giá trị chung. Phương pháp này đã mang lại kết quả trong một vài trường hợp mà các yêu cầu có thể định lượng. Ngay cả trong các trường hợp đó, việc xác định độ quan trọng của các yêu cầu gồm nhiều loại khác nhau trong công thức của hàm giá trị đó có thể gặp khó khăn. Ví dụ, một thiết kế có giá trị các phần là nhỏ nhất và trọng lượng nhỏ nhất thì một hàm giá trị thích hợp có thể là:

$$\psi = \sqrt[n]{\left(\frac{c}{c_0}\right)^p + \left(\frac{w}{w_0}\right)^q}$$

trong đó c và w tương ứng là giá và trọng lượng và c_0 và w_0 là các hệ số hiệu chỉnh chỉ ra mức độ quan trọng của hai hệ số, trong khi các số mũ p, q và n biểu diễn độ quan trọng tương đối của việc giảm thiểu hai hệ số đó. Rõ ràng, có nhiều lựa chọn có thể cho các tham số này và nhiều mô hình khác nhau có thể được dùng. Điều này nêu bật các khó khăn của phương pháp này. Một ví dụ của sự khó khăn này trong một trường hợp tương đối đơn giản là việc xác định giá trị của một thành phần ngoài khoảng cho phép. Các mô hình cho vấn đề này có thể là một hàm bước có giá trị bằng 0 trong

khoảng cho phép và giá trị bằng C ngoài khoảng cho phép hoặc là hàm giá trị bậc hai Taguchi [12]. Có thể thấy rõ giải pháp tối ưu tìm được phụ thuộc vào hàm giá trị đã chọn. Mặt khác, trong một vài trường hợp, tối ưu tìm được không phụ thuộc nhiều vào việc chọn hàm giá trị và có thể nhận được giá trị thỏa mãn mà không tốn nhiều công sức để tìm một hàm giá trị chính xác. Trong thực tế, nếu không có thiết kế để lựa chọn thì có thể làm theo cách ngược lại. Thay vì tìm cách tối thiểu một hàm giá trị cụ thể nào đó, người thiết kế có.thể tìm dải các tham số của hàm giá trị để dẫn đến một giải pháp tối ưu.

Giai đoạn 4

Giai đoạn thứ tư là thiết kế chi tiết, trong đó toàn bộ quá trình được thực hiện lại để tìm các chi tiết thiết kế chưa được giải quyết cho các thành phần riêng biệt của thiết kế. Quá trình này bản chất là đệ quy, mỗi quyết định thiết kế bậc cao đưa ra một vấn đề thiết kế đơn giản hơn ở một mức độ trừu tượng thấp hơn với các yêu cầu đơn giản hơn. Để thực hiện điều này, chúng ta cần có khả năng mô hình hóa thiết kế ở nhiều mức độ trừu tượng và cho phép định rõ giao diện giữa các thành phần. Các mô hình này phải cho phép chúng ta xác định các thành phần tĩnh cũng như động của giao diện.

Ngoài phương pháp đệ quy, dĩ nhiên cũng có thể sử dụng phép lặp. Để thực hiện phép lặp này, các mô hình thiết kế phải thể hiện không chỉ các quyết định đã được đưa ra mà còn cả lý do của các quyết định đó hoặc việc lập luận cần phải được kiểm tra lại trong mỗi bước lặp, ít ra cũng phải kiểm tra lại trước khi quá muộn.

22.2 Mục đích của việc mô hình hóa

Có thể thấy rằng các mô hình phục vụ rất nhiều mục đích trong thiết kế cơ điện tử. Đầu tiên, mô hình để thể hiện các ý tưởng, các giả định, các yêu cầu và cho phép truyền đạt các ý tưởng đó tới những người khác trong nhóm thiết kế và các nhà đầu tư trong quá trình thiết kế. Thứ hai là các mô hình cung cấp một cấu trúc theo thứ bậc cho phép phân chia công việc trong quá trình thiết kế và cho phép làm việc đồng thời trên các phần riêng biệt. Thứ ba là các mô hình cho phép chúng ta hiểu rõ ứng xử của một hệ thống mà nó có thể bị che khuất khi chúng ta cố gắng xem xét hệ đó trong toàn bộ sự phức tạp của nó. Thứ tư là cho phép chúng ta hiểu được sự tương tự của nó với các hệ khác và sử dụng kinh nghiệm trước đó để giúp giải quyết các vấn đề hiện tại. Thứ năm là các mô hình cho phép xác định và kiểm tra sự thiếu hiểu biết của chúng ta về hoạt động thực tế của hệ bằng cách xác định các tham số chưa biết và cung cấp các giả thuyết phù hợp cho các thử nghiệm. Trong mục này, chúng ta sẽ xem xét những mục đích khác nhau của mô hình hóa và nghiên cứu các yếu tố tác động đến một phương pháp mô hình hoá cụ thể.

Tài liệu và Giao tiếp

Mục đích đầu tiên của một mô hình mà chúng ta cần xem xét là sự thể hiện bằng tài liệu. Chúng ta thường quên rằng bên cạnh các mục đích khác, các mô hình thường là công cụ tốt nhất để chúng ta giao tiếp với đồng nghiệp, với khách hàng và với những người kế cận. Mỗi tài liệu được tạo ra trong quá trình thiết kế là một mô hình các thành phần của hệ thống. Tới một mức độ nhất định, chúng ta có thể tránh các sự thể hiện lặp lại cho cùng một ý tưởng, chúng ta không chỉ giảm tải và giảm thời gian đưa ra thị trường, mà chúng ta cũng giảm khả năng xảy ra mâu thuẫn trong các tài liệu. Mô hình cung cấp thông tin như thế nào? Xem sơ đồ mạch. Nó là sự thể hiện bằng tài liệu của thiết kế mạch. Nó cũng là một mô hình ứng xử mà chúng ta đang cố tìm kiếm từ mạch điện. Sự không lý tưởng và những thay đổi của các thành phần sẽ là nguyên nhân các mạch thực tế chạy khác so với mô hình, nhưng mô hình vẫn đáp ứng mục tiêu của nó. Ngay từ giai đoạn đầu của quá trình thiết kế, các tài liệu thể hiện các yêu cầu đối với sản phẩm hơn là các đặc tính thỏa mãn chúng.

Các tài liệu thiết kế là quan trọng đối với ít nhất bốn nhóm khác nhau. Nhóm kỹ thuật cần thông tin giữa các thành viên của mình để chắc chắn rằng công sức không bị lãng phí bởi các

thành phần hoạt động riêng biệt hoặc không được đóng góp cho các yêu cầu chung của sản phẩm. Việc trao đổi giữa đội thiết kế và khách hàng là cần thiết để chắc chắn rằng các yêu cầu được hiểu chính xác và được thực hiện đầy đủ trong thiết kế. Người quản lý cần hiểu tình trạng của dự án về các rủi ro, các phí tổn đang phải trả và sự thỏa hiệp có thể tác động đến thị trường. Cuối cùng, các đội thiết kế sau này có thể kế thừa các tài liệu thể hiện các ý tưởng, các phân tích và các quyết định từ giai đoạn thiết kế ban đầu. Vai trò này của tài liệu như là một sự thông tin với những nhà nghiên cứu trong tương lai đã được công nhận là một phần quan trọng của tất cả các lĩnh vực khoa học [14] nhưng thường bị bỏ qua do phải tập trung vào những báo cáo ngắn hạn.

Các đặc trưng nào của mô hình đưa ra tài liệu tốt và truyền thông rõ ràng? Rõ ràng rằng mô hình phải có nhiều người khác nhau hiểu được. Yêu cầu này khuyến khích việc dùng các phương pháp mô hình hóa chuẩn khi có thể. Nó cũng khuyến khích việc dùng các phương pháp mô hình hóa liên quan đến nhiều lĩnh vực khoa học ngay khi có thể. Khi chọn mô hình cho mục đích thể hiện bằng tài liệu, một điều phải nhớ rằng mỗi một mô hình có thể thể hiện một số đặc tính nào đó của thiết kế trong khi những đặc tính khác lại bị dấu đi. Ví dụ sơ đồ tuần tự thể hiện rõ sự truyền thông giữa hai đối tượng trong khi dấu đi chi tiết việc các đối tượng gửi thông báo đi hoặc dịch một thông báo đến như thế nào. Nếu nhiều loại mô hình của hệ thống được dùng để thể hiện các khía cạnh khác nhau của hệ thống thì phải có một số cách để liên hệ chéo giữa các mô hình đó. Ngoài ra, nếu sự thể hiện bằng tài liệu là có ích cho những người thiết kế sau này, thì có thể tham khảo chéo các mô hình thiết kế dựa trên các tiêu chuẩn khác nhau bao gồm các yêu cầu của hệ thống và các thành phần, sao cho các thiết kế sau này có thể sử dụng những tài liệu hiện tại. Rõ ràng, bất kỳ một kỹ thuật mô hình hóa nào đáp ứng một số mục đích khác khi thiết kế cũng được coi là tài liệu, nhưng mô hình đó phải được chọn để đáp ứng các tính chất mong muốn thể hiện.

Cấu trúc thứ bậc

Các hệ cơ điện tử thực bao gồm một số lượng lớn các thành phần ảnh hưởng lẫn nhau theo nhiều cách. Các mô hình của chúng ta sẽ đơn giản hóa các giao diện phức tạp của chúng và mô tả các sản phẩm và các quá trình theo các thành phần tương tác đơn giản với các giao diện và các ứng xử được định nghĩa rõ ràng. Việc đơn giản hóa này cho phép chúng ta chia nhỏ những vấn đề phức tạp thành tập hợp các vấn đề đơn giản hơn với các giải pháp có thể tích hợp dễ dàng hơn. Trong các hệ cơ điện tử phức tạp mà ở đó chúng ta phải xem xét các thành phần tương tác trong vài miền năng lượng thì việc đơn giản hoá là cần thiết. Khi chúng ta vẽ sơ đồ khối của một hệ thống, tức là chúng ta đang chia ứng xử của hệ thống thành tập hợp các phần tử xác định với ứng xử đã được mô hình hoá và một tập hợp các tương tác đã biết giữa các phần tử đó. Vai trò này của mô hình hoá có thể hướng chúng ta chọn các phương pháp mô hình hoá cho phép chia mô hình thành các phần có thể xử lý riêng biệt bởi các kỹ sư độc lập. Ví dụ hệ các phương trình vi phân tuyến tính có thể cho chúng ta nhiều thông tin về các chế độ cơ bản của một hệ thống, nhưng lại cho ít thông tin về cách chia nhiệm vụ thiết kế cho các thành viên của đội. Tuy nhiên, một sơ đồ khối, một sơ đồ lớp hoặc một sơ đồ luồng dữ liệu sẽ cung cấp các giao diện rõ ràng giữa các phần tử để cho phép chia nhỏ vấn đề.

Sự hiểu biết

Đối với các hiểu biết về ứng xử của hệ thống, các mô hình khác nhau cung cấp các hiểu biết khác nhau. Các biểu đồ dòng tín hiệu của các hệ cơ điện tử chỉ ra sự có mặt của các vòng phản hồi có thể làm ổn định hoặc mất ổn định hệ thống. Các phương trình vi phân cung cấp thông tin về các ứng xử theo thời gian và mức độ quan trọng của các thông số cũng như đưa ra các ước tính về sự đúng đắn của các giả thiết. Tương tự, các sơ đồ thời gian và tuần tự có thể cung cấp thông tin về khả năng xử lý cần thiết để thoả mãn các yêu cầu về thời gian cho các trường hợp khác nhau. Trong mỗi trường hợp, mô hình cung cấp thông tin về vấn đề hoặc các đặc trưng của một giải pháp đã được đề xuất bằng cách tập trung vào một số yếu tố chính của hệ và bỏ qua các yếu tố khác.

Sự tương tự

Liên quan tới các hiểu biết mà một mô hình có thể đem lại cho chúng ta về hệ thống đang được xem xét là các hiểu biết mà một mô hình có thể cung cấp, cho phép chúng ta nhận ra sự tương tự với các hệ thống khác mà chúng ta hoặc những người khác đã biết trước đó. Theo cách này, các phương pháp mô hình hoá sử dụng sự tương tự giữa các lĩnh vực khác nhau có thể trở nên hữu ích. Tuy nhiên sự tương tự cần phải được chọn một cách cẩn thận. Xét sự tương tự khả phổ biến giữa các đại lượng điện từ và các đại lượng cơ học được chỉ ra ở nửa trái của hình 22.7. Theo đó, vận tốc tương tự với điện áp và lực tương tự với dòng điện. Tuy nhiên, ở nửa phải của hình 22.7, các phương trình của hệ thống điện là không thay đổi trong hệ thống đối ngẫu khi dòng điện và điện áp được chuyển đổi qua lại và vai trò của điện cảm và điện trở được chuyển đổi qua lại tương ứng với điện dung và điện dẫn. Hệ thống đối ngẫu này dẫn đến một sự tương tự khác về mặt cơ học, trong đó vận tốc tương tự với dòng điện và lực tương tự với điện áp. Các tính chất tương tự đó rất có ích trong việc chuyển các tham số thiết kế từ vùng năng lượng này sang vùng năng lượng khác, cũng như tính đối ngẫu giữa hai mô hình điện rất có ích khi thiết kế mạch. Với bất kỳ tính chất tương tự nào, phải nhớ rằng có sự khác biệt cơ bản giữa các thành phần thực so với các mô hình lý tưởng của chúng và các tính chất tương tự không được tuân theo một cách chặt chẽ. Điều này có thể vừa có lợi cũng vừa có hại vì một thiết kế kém chất lượng do tính chất không lý tưởng của các thành phần có thể được thay thế bằng một thiết kế tương tự có các thành phần không lý tưởng khác nhưng ít bất lợi hơn.

Phần tử	Điện	Cơ	Điện	Cơ
Điện dung	$i = C\frac{dv}{dt}$	$f = M\frac{dv}{dt}$	$v = \frac{1}{C}\int i dt$	$f = k\int v dt$
Điện cảm	$i = \frac{1}{L}\int v dt$	$f = k\int v dt$	$v = L\frac{di}{dt}$	$f = M\frac{dv}{dt}$
Điện trở	$i = \frac{1}{R}v$		$v = Ri$	$f = Bv$
Điện dẫn	$i = Gv$	$f = Bv$	$v = \frac{1}{G}i$	

HÌNH 22.7 Sự tương tự điện - cơ

Nhận biết thiếu sót

Vai trò cuối cùng của việc mô hình hoá là để giúp chúng ta nhận biết những thiếu sót. Điều này được thực hiện dưới hai dạng. Đầu tiên, việc xây dựng mô hình thường đòi hỏi chúng ta định rõ các tham số mà giá trị số của chúng là chưa biết. Điều này buộc chúng ta phải ước lượng phạm vi có thể của các tham số đó hoặc phải thử nghiệm các thiết kế để xác định giá trị tham số. Việc nhận biết các thành phần còn thiếu là bước đầu tiên để xác định chúng. Theo cách này các mô hình sẽ giúp chúng ta nhận biết sự thiếu sót của vấn đề hoặc giải pháp đã đưa ra. Thứ hai, các dự đoán mà chúng ta đưa ra với sự trợ giúp của các mô hình là có thể kiểm tra. Một mô hình đầy đủ để mô tả ứng xử của hệ thống có thể được xác định bằng cách so sánh ứng xử của hệ thống thực với các dự đoán dựa trên mô hình đó. Sự khác nhau giữa kết quả của hệ thống thực và các dự đoán của mô hình chỉ ra rằng có nhiều sai sót trong mô hình đó. Bằng cách kiểm tra tính đúng đắn của các mô hình, chúng ta sẽ nhận biết được các thành phần của hệ thống thực đã được mô phỏng không chính xác trong mô hình hoặc các phần của mô hình không tồn tại trong hệ thống thực. Sự không nhất quán đó có thể chỉ ra một giả định không thích hợp hoặc sự hiểu sai cơ bản hệ thống vật lý. Trong cả hai trường hợp, nguồn gốc của sự không nhất quán đó phải được hiểu rõ để xác định mô hình của chúng ta có thích hợp với các mục đích đề ra hay không.

22.3 Mô hình hóa các hệ thống và các tín hiệu

Trong phạm vi của các hệ thống và các tín hiệu, chúng ta thường quan tâm tới việc mô hình hoá động lực học hệ thống và các luồng thông tin qua hệ thống và môi trường của nó. Như đã trình bày ở trên, đó là một phần nhỏ trong việc mô hình hoá cần thiết cho thiết kế hệ thống. Tuy nhiên, có nhiều quyết định cần phải đưa ra khi chọn một mô hình cho ứng xử của một hệ động lực. Điều này bao gồm việc lựa chọn giữa mô hình giải tích và mô hình số, tham số phân bố hoặc tập trung, lựa chọn phương pháp mô hình hoá các hệ số ngẫu nhiên và lựa chọn giữa mô hình tuyến tính và phi tuyến. Vấn đề này sẽ được trình bày trong các phần sau đây.

Mô hình giải tích và mô hình số

Trong lĩnh vực này có rất nhiều công cụ giải tích và số để hỗ trợ chúng ta, vì vậy có nhiều lựa chọn được đưa ra đối với mức độ phức tạp khác nhau của các mô hình được sử dụng cho các nhiệm vụ khác nhau. Các công cụ giải tích đưa ra sự hiểu biết về các ứng xử của toàn bộ hệ thống và cho phép chúng ta so sánh các hệ khác nhau dựa trên sự tương tự trong các mô hình của chúng. Các hệ tương tự đó rất hữu ích trong giai đoạn hình thành ý tưởng thiết kế bởi vì chúng cho phép ta đưa ra giải pháp cho các vấn đề khác nhau dựa trên giải pháp của vấn đề đã được giải quyết. Tuy nhiên như đã mô tả ở trên, các tính chất tương tự đó thường không đúng khi xem xét kỹ. Trong trường hợp sự tương tự động lực, chúng thường dựa trên các hệ được mô tả bằng các phương trình vi phân tương đương (ví dụ các máy tạo dao động điện và cơ). Tuy nhiên, các phương trình vi phân tuyến tính đó chỉ là sự đơn giản hóa khi bỏ qua các thành phần phi tuyến và ứng xử theo thời gian có thể ảnh hưởng lớn đến ứng xử của hệ thống. Vì vậy khi áp dụng một tính chất tương tự phải luôn cẩn thận đưa ra các giả thiết trạng thái trong quá trình mô hình hóa.

Mặt khác, các công cụ số không mang lại những hiểu biết giống như là các công cụ giải tích mà nó đưa ra những hiểu biết khác và chi tiết hơn về ứng xử và kết quả của hệ thực. Đặc biệt các công cụ đó trở lên hữu ích nhất ở giai đoạn cuối của quá trình thiết kế, cho cả phân tích tối ưu và công việc thiết kế chi tiết. Các công cụ này cho phép sử dụng các phương pháp mô hình hóa mà trước đây không được dùng và cho phép chúng ta xem xét các phương pháp được coi là rất khó trước đây. Ví dụ lời giải số của các phương trình vi phân phi tuyến cho phép mô hình hóa các hệ mà trước đó chỉ được tiếp cận bằng các phép xấp xỉ tuyến tính.

Phương trình vi phân đạo hàm riêng và phương trình vi phân thường

Chúng ta phải lựa chọn hoặc là mô hình hóa các tính chất động học của hệ thống bằng các phương trình vi phân thường hoặc là bằng các phương trình vi phân đạo hàm riêng. Các phương trình vi phân đạo hàm riêng được sử dụng khi các giá trị cần quan tâm thay đổi theo không gian cũng như thời gian. Ví dụ nếu chúng ta đang thiết kế một cánh tay robot, chúng ta có thể dùng các phương trình vi phân thường nếu coi cánh tay robot là vật cứng, các phương trình vi phân đạo hàm riêng nếu muốn mô tả một cách chi tiết khuỷu tay của một cánh tay mềm. Tuy nhiên, cái gì sẽ xảy ra nếu chúng ta công nhận sự mềm dẻo của cánh tay nhưng không cần biết chi tiết dịch chuyển của nó và chỉ quan tâm đến tác dụng của sự mềm dẻo đó lên cơ quan phản ứng lại kích thích cuối cùng? Trong trường hợp này chúng ta có thể tạo một mô hình tham số tập trung của cánh tay để tổng kết các tính chất động lực của nó theo dịch chuyển của cơ quan phản ứng lại kích thích cuối cùng và các lực lên bộ điều khiển và cơ quan phản ứng cuối. Khi nào mô hình các tham số tập trung này được dùng? Có hai hạn chế. Một là khi chi tiết sự thay đổi của ứng xử theo không gian được bỏ qua. Nếu cần biết các chi tiết đó (ví dụ để xác định sự tập trung ứng suất hoặc sự phân tán nhiệt) chúng ta không thể áp dụng mô hình đơn giản hóa. Thứ hai là khi phương trình vi phân đạo hàm riêng có thể không tuân theo dạng mô hình tham số tập trung do sự phức tạp của nó.

Mô hình thống kê và mô hình tiền định

Một yêu cầu đặt ra khi mô hình hóa ứng xử của hệ thống là thể hiện được những đại lượng biến thiên không được điều khiển trong mô hình [15]. Phương pháp được sử dụng phụ thuộc vào dạng và đặc tính của sự biến thiên đó. Ví dụ giá trị của một vài tham số hệ thống có thể không biết chính xác, nhưng có thể không đổi theo thời gian. Các tham số khác có thể thay đổi chậm theo thời gian theo một cách nào đó. Trong trường hợp đầu, độ nhậy của hệ thống đối với giá trị tham số có thể được xác định bằng cách dùng phương pháp giải tích hoặc phương pháp số. Trong trường hợp sau, cần phải nghiên cứu tốc độ biến thiên của tham số để xác định xem hệ có thể phân tích được ở trạng thái gần như bền vững không hoặc động học của sự biến thiên tham số đó có gắn với động học của hệ thống không. Với mục đích trên, các phương trình ở dạng không thứ nguyên thường rất hữu ích trong đó thời gian và các hằng số tỷ lệ chỉ ra các tốc độ tới hạn cho việc gắn kết.

Khi cần xem xét biến thiên của các tham số theo thời gian hoặc khi một vài tín hiệu vào hệ thống có sự biến thiên không được điều khiển, chúng ta phải lựa chọn mô hình hóa tín hiệu vào như một tín hiệu bất kỳ hoặc một tín hiệu với các tham số thống kê đã biết. Trong trường hợp đầu, chúng ta có ngay tín hiệu vào khác và có thể phân tích hệ thống về độ nhạy đối với tín hiệu vào đó. Trong trường hợp thứ hai, chúng ta có thể mô tả hệ thống theo các phép đo quang phổ tương quan hoặc tương đương và dùng các kỹ thuật lọc tối ưu để giảm độ không chắc chắn [16]. Một điểm quan trọng trong việc mô hình hóa thống kê của một hệ thống là mô hình dựa vào sự biến thiên đã biết của quá trình hơn là dựa vào giả thiết ồn phụ, ồn trắng, ồn Gaussian ở mọi chỗ [17].

Tuyến tính và phi tuyến

Đã có rất nhiều kỹ thuật phân tích các hệ tuyến tính bất biến theo thời gian. Nhưng trong thực tế các thành phần thường là không tuyến tính. Một thuận lợi của các kỹ thuật mô hình hóa cho phép chúng ta chia một hệ thống phức tạp thành các thành phần con đơn giản hơn là chúng ta có thể coi các thành phần phi tuyến như là các thành phần riêng biệt và xây dựng mô hình tuyến tính cho chúng. Với các biến thiên nhỏ quanh một điểm có thể lấy vi phân trên một đường cong không tuyến tính, chúng ta có thể tìm thấy một mô hình tuyến tính, nhưng trị số của mô hình này bị giới hạn chặt chẽ và phụ thuộc vào độ chính xác của nó trên một dải giới hạn các giá trị vào. Trong trường hợp phi tuyến gián đoạn hoặc phi tuyến không thể lấy đạo hàm, thì đó là cả một vấn đề lớn. Chỉ có thể tìm được một mô hình tuyến tính khi ở xa điểm gián đoạn hoặc khi khoảng gián đoạn là nhỏ so với phần tuyến tính của mô hình.

Tuy nhiên, tính phi tuyến tạo ra các ứng xử mới cho hệ thống mà những hệ hoàn toàn tuyến tính không có. Chính vì lý do này mà các mô hình phi tuyến là cần thiết trong một vài trường hợp. Ví dụ với một hệ tuyến tính, theo lý thuyết, các dao động ổn định chỉ có thể có trong các hệ phương trình vi phân với các giá trị riêng thuần ảo. Phần thực dương làm cho các dao động lớn dần, trong khi phần thực âm làm cho các dao động giảm dần. Rõ ràng, ứng xử của một hệ như vậy rất nhạy cảm đối với các tham số của hệ. Chỉ một thay đổi nhỏ sẽ làm tắt dao động hoặc đưa hệ vào vùng hoạt động phi tuyến. Ngoài ra, ngay cả với các dao động lý tưởng, biên độ của dao động cũng không bị ràng buộc. Tuy nhiên, một hệ phi tuyến như phương trình Van der Pol:

$$\ddot{y} = -\omega^2 y + \alpha\left[1 - \left(\frac{y}{y_0}\right)^2\right]\dot{y}$$

bao gồm một *hệ số suy giảm* tăng theo biên độ dao động. Các dao động nhỏ so với y_0 sẽ lớn lên, trong khi các dao động lớn hơn thì tắt dần. Ví dụ này cùng với những ứng xử phi tuyến cơ bản khác không thể được mô hình hóa một cách thỏa đáng bằng các mô hình tuyến tính. Ngay cả khi ứng xử này được mô tả bằng cách sử dụng mô hình phi tuyến thì hệ trạng thái bền vững (steady-state system) có thể được mô hình hóa cho nhiều mục đích bằng cách dùng hệ tuyến tính tương đương của nó.

Khi sử dụng các phương trình vi phân để mô hình hóa một hệ thống, chúng ta phải đặc biệt chú ý cách chọn các đơn vị đo và nhóm các tham số như thế nào. Ví dụ trong phương trình Van der Pol ở trên, chúng ta có thể chọn hệ số tỷ lệ khác cho y sao cho y_0 bằng 1 và hệ số tỷ lệ thời gian sao cho ω bằng 1. Trong trường hợp đó chúng ta có:

$$z'' = -z + \varepsilon\left(1 - z^2\right)z'$$

trong đó $z = y/y_0$, $\tau = \omega t$ và $\varepsilon = \alpha/\omega$. Biến đổi tỷ lệ này sẽ đưa tới một phương trình không thứ nguyên cũng như đưa ra hệ số thời gian thực của hệ thống, hệ số tỷ lệ của y và chỉ ra rằng độ lớn của tỷ số giữa α và ω là quan trọng. Nói chung, chúng ta thấy rằng việc viết các phương trình ở dạng không thứ nguyên có ba ưu điểm. Thứ nhất như đã nói ở trên, những hệ số tỷ lệ cho biết ý nghĩa của hệ số tỷ lệ thời gian và tầm quan trọng của các đặc tính ứng xử động học của hệ thống. Thứ hai, các tham số không thứ nguyên là kết quả của việc kết hợp các tham số của hệ vật lý đưa ra cách thức mô tả ứng xử trên một phạm vi rộng của hệ thống dưới dạng các tham số có thể xử lý dễ dàng. Thứ ba, bằng cách đưa các phương trình về dạng không thứ nguyên chúng ta có thể phân loại chúng và nhận dạng các phương trình tương tự trong các lĩnh vực khác.

Tài liệu tham khảo

[1] Hubka, V. and Eder, W.E., Engineering Design—General Procedural Model of Engineering Design, Heuista, Zurich, 1992.

[2] Pahl, G. and Beitz, W., Engineering Design: A Systematic Approach, Springer-Verlag, Berlin, 1988.

[3] Dym C.L., Engineering Design—A Synthesis of Views, Cambridge University Press, New York, 1994.

[4] Booch, G., Jacobson, I., Rumbaugh, J., and Rumbaugh, J., The Unified Modeling Language User Guide, Addison-Wesley, New York, 1998.

[5] Douglass, B., Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks and Patterns, Addison-Wesley, New York, 1999.

[6] Demarco, T., Structured Analysis and System Specification, Yourdon Press, NJ, 1978.

[7] Coad, P. and Yourdon, E., Object-Oriented Analysis, Yourdon Press, NJ, 1990.

[8] Glegg, G.L., The Design of Design, Cambridge University Press, London, 1969.

[9] Shearer, Murphy, and Richardson, Introduction to Dynamic Systems, Addison-Wesley, Reading, MA, 1967.

[10] Karnopp, D. and Rosenberg, R., System Dynamics: A Unified Approach, Wiley, New York, 1975.

[11] Suh, N.P., Principles of Design, Oxford University Press, NY, 1990.

[12] Taguchi, G. and Yokoyama, Y., Taguchi Methods: Design of Experiments, American Supplier Institute, Dearborn MI, 1994.

[13] Koen, B.V., Definition of the Engineering Method, American Society for Engineering Education, Washington, 1985.

[14] Lonergan, B., Method in Theology, University of Toronto, Toronto, 1990.

[15] Zhou, K., Essentials of Robust Control, Prentice-Hall, NJ, 1998.

[16] Papoulis, A., Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1991.

[17] Law, A. and Kelton, W., Simulation, Modeling and Analysis, McGraw-Hill, New York, 1991.

23

Tín hiệu và hệ thống

Momoh-Jimoh Eyiomika Salami
International Islamic University of Malaysia

Rolf Johansson
Lund Institute of Technology

Kam Leang
University of Washington

Qingze Zou
University of Washington

Santosh Devasia
University of Washington

C. Nelson Dorny
University of Pennsylvania

23.1 Tín hiệu thời gian liên tục và rời rạc 23-1
23.2 Biến đổi z và các hệ thống số 23-27
23.3 Các mô hình không gian - trạng thái thời gian rời rạc và liên tục 23-37
23.4 Hàm truyền và biến đổi Laplace 23-49

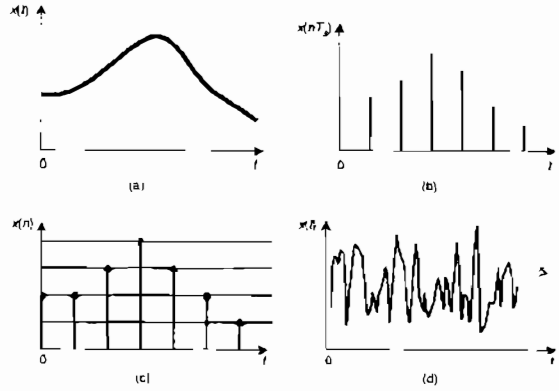
23.1 Tín hiệu thời gian liên tục và rời rạc

Tín hiệu là các biến vật lý hoặc các đại lượng đo được ở các phần khác nhau của một hệ thống, khi được xử lý nó cung cấp các thông tin mong muốn. Thực tế, có rất nhiều loại tín hiệu. Tín hiệu điện ở dạng dòng hoặc áp là đại lượng đo được dễ dàng nhất, vì vậy cần dùng các cảm biến và các bộ chuyển đổi để biến đổi các đại lượng không điện thành các tín hiệu điện. Các tín hiệu này phải được xử lý bằng các kỹ thuật phù hợp để thu được kết quả mong muốn. Một số phương pháp biểu diễn tín hiệu thích hợp để xử lý tín hiệu ở cả miền thời gian và tần số sẽ được trình bày trong phần này.

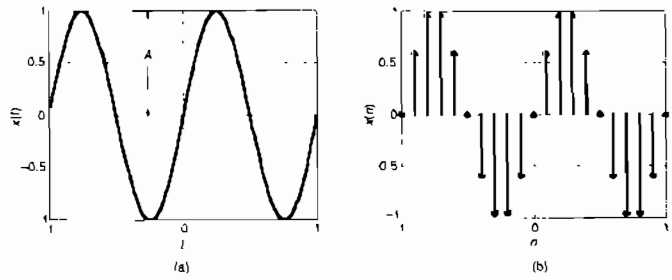
Phân loại tín hiệu [1 - 4]

Tín hiệu được phân loại thành tín hiệu thời gian liên tục (CT – Continuous Time) hoặc tín hiệu thời gian rời rạc (DT – Discrete Time), mỗi loại tín hiệu này lại có thể phân thành các tín hiệu xác định hoặc ngẫu nhiên. Một tín hiệu xác định luôn có thể biểu diễn ở dạng toán học, trong khi đó thời điểm xảy ra hoặc giá trị của một tín hiệu ngẫu nhiên là không thể đoán trước được. Một tín hiệu thời gian liên tục $x(t)$ có một giá trị ứng với mỗi giá trị thời gian t, trong khi một tín hiệu thời gian rời rạc $x(n)$ chỉ có giá trị ở các điểm rời rạc, là các giá trị nguyên n.

Ví dụ về các tín hiệu CT và DT là các tín hiệu tương tự và tín hiệu số tương ứng. Nếu biên độ của một tín hiệu là một giá trị bất kỳ trong một dải liên tục thì nó là một tín hiệu tương tự. Mặt khác, biên độ của một tín hiệu số chỉ có thể là một số hữu hạn các giá trị ở các điểm rời rạc. Hình 23.1 là các ví dụ về tín hiệu thời gian liên tục, thời gian rời rạc, tín hiệu số và ngẫu nhiên.



HÌNH 23.1 Một số lớp tín hiệu: (a) Tín hiệu tương tự thời gian liên tục, (b) Tín hiệu được lấy mẫu, (c) Tín hiệu số, (d) Tín hiệu ngẫu nhiên



HÌNH 23.2 Các tín hiệu có chu kỳ: (a) Thời gian liên tục (CT), (b) Thời gian rời rạc (DT)

Tín hiệu tiền định được phân thành hai loại chính là tín hiệu có chu kỳ và các tín hiệu không có chu kỳ. Tín hiệu có chu kỳ sẽ có cùng giá trị ở các thời điểm cách nhau một chu kỳ T, tức là $x(t)$ thỏa mãn mối quan hệ $x(t)=x(t+T)$, $-\infty<t<\infty$. Một ví dụ của tín hiệu thời gian liên tục có chu kỳ là tín hiệu hình sin với dạng $x(t)=A\sin(\Omega t+\theta)$ trong đó θ là pha tính theo radian, $\Omega = 2\pi F$ là tần số tính theo radian/s và F là tần số tính theo Hz. Chú ý rằng dải tần số của các tín hiệu hình sin tương tự là $-\infty<F<\infty$. Một tín hiệu thời gian rời rạc có chu kỳ được mô tả bởi $x(n)=x(n+N)$, $-\infty<n<\infty$, trong đó N là chu kỳ. Một ví dụ của tín hiệu loại này là sóng sin $x(t)= A\sin(2\pi rn+\theta)$, $-\infty<n<\infty$, trong đó r là tần số tín hiệu trên tần số lấy mẫu và có các giá trị trong dải $-\frac{1}{2}\le r\le\frac{1}{2}$. Các tín hiệu sin có tần số thuộc dải này là duy nhất và hoàn toàn khác nhau. Tuy nhiên các tín hiệu sin dạng DT có các tần số khác nhau một số nguyên lần của 2π là giống nhau. Ví dụ về các tín hiệu sin dạng DT và tương tự được mô tả trong hình 23.2.

Một tín hiệu tiền định bất kỳ không có chu kỳ được gọi là tín hiệu không tuần hoàn. Các tín hiệu hình sin tắt dần và các tín hiệu suy giảm theo hàm mũ là các ví dụ phổ biến của tín hiệu không theo chu kỳ. Đối với một vài ứng dụng, có thể phân loại các tín hiệu theo công suất hoặc năng lượng của chúng. Năng lượng tổng trên toàn dải $t\in(-\infty,\infty)$ của một tín hiệu CT được đưa ra bởi:

$$E=\lim_{T\to\infty}\int_{-T/2}^{T/2}\left|x(t)\right|^2 dt \tag{23.1}$$

và công suất trung bình được định nghĩa như sau:

$$P=\lim_{T\to\infty}\left[\frac{1}{T}\int_{-T/2}^{T/2}\left|x(t)\right|^2 dt\right] \tag{23.2}$$

Do vậy, $x(t)$ là tín hiệu năng lượng nếu và chỉ nếu $0<E<\infty$, điều này có nghĩa là $P=0$. Tương tự, $x(t)$ là tín hiệu công suất nếu và chỉ nếu $0<P<\infty$, chứng tỏ $E=\infty$. Vì vậy một tín hiệu không

thỏa mãn một trong hai định nghĩa trên sẽ không phải là tín hiệu năng lượng hoặc công suất. Các định nghĩa trên cũng có thể áp dụng cho các tín hiệu DT chỉ khác là phép tích phân trong các công thức (23.1) và (23.2) được thay thế bằng phép lấy tổng. Nói chung, các tín hiệu có chu kỳ tồn tại ở mọi thời điểm và được hiểu là có năng lượng vô hạn. Tuy nhiên, chúng có công suất trung bình hữu hạn vì vậy chúng là các tín hiệu công suất. Mặt khác, các tín hiệu được giới hạn trong một khoảng hữu hạn là các tín hiệu năng lượng. Việc phân loại một tín hiệu thành tín hiệu năng lượng hữu hạn, công suất hữu hạn hoặc không phải năng lượng cũng như công suất là quan trọng để chọn được cách phân tích phù hợp.

Các hàm đặc biệt

Khi mô hình hóa tín hiệu, các hàm đặc biệt rất có ích để đơn giản hóa các phép phân tích.

Hàm xung đơn vị (Unit Impulse)

Hàm xung hoặc hàm delta là mô hình toán để biểu diễn các hiện tượng vật lý xảy ra trong một khoảng thời gian rất nhỏ. Khoảng thời gian này có thể được giả thiết bằng 0. Hàm xung đơn vị này không phải là hàm toán học theo nghĩa thông thường, đúng hơn nó là hàm phân phối hoặc hàm suy rộng. Vì vậy hàm xung có thể được mô tả bởi tác dụng của nó lên *hàm thử* $\phi(t)$, đó là:

$$\int_{-\infty}^{+\infty} \phi(t)\delta(t)dt = \phi(0) \tag{23.3}$$

với điều kiện: $\phi(t)$ liên tục ở $t=0$. Công thức này chỉ ra tính chất thay đổi của hàm xung đơn vị. Biểu đồ của hàm xung đơn vị được chỉ ra trong hình 23.3(a). Bảng 23.1 mô tả các thuộc tính điều khiển của hàm xung đơn vị.

BẢNG 23.1 Các tính chất của hàm xung đơn vị

Tính chất	Biểu diễn toán
Lấy mẫu	$\int_{-\infty}^{+\infty} x(t)\delta(t-a)dt = x(a)$
Dịch	$x(t)\delta(t-a) = x(a)\delta(t-a)$
Cân chỉnh	$\delta(at \pm b) = \frac{1}{\lvert a\rvert}\delta\left(t \pm \frac{b}{a}\right)$
Tích chập	$x(t) * \delta(t-a) = \int_{-\infty}^{+\infty} x(\tau)\delta(t-\tau-a)d\tau = x(t-a)$

Hàm bậc thang đơn vị (Unit Step)

Hàm bậc thang đơn vị thường dùng để phân tích toán học các tín hiệu CT. Điều này đã được mô tả trong hình 23.3(b) và được định nghĩa như sau:

$$u(t) = \begin{cases} 1 & t > 0 \\ 0 & t < 0 \end{cases} \tag{23.4}$$

Hàm signum nhận được từ hàm nhảy như sau:

$$\text{sgn}(t) = 2u(t) - 1$$

Hàm dốc (Ramp Function)

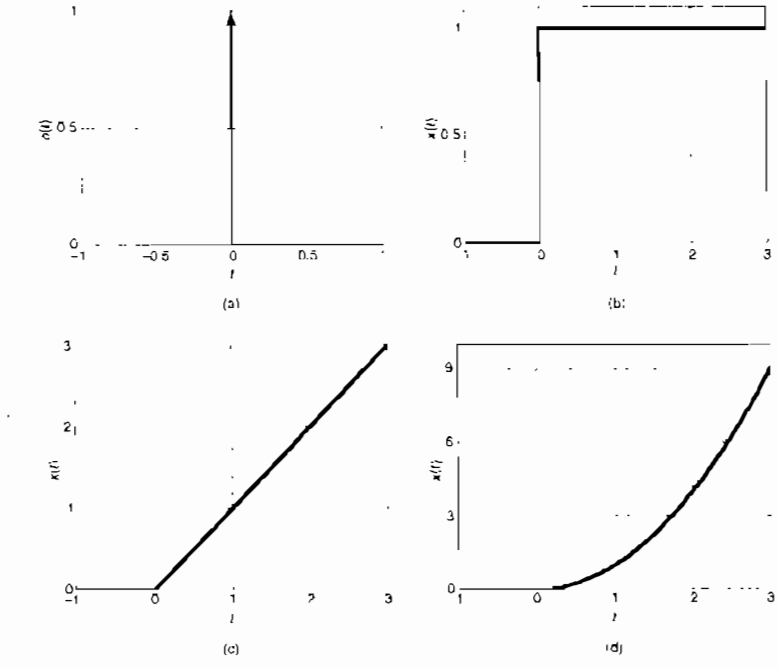
Tích phân (23.4) sẽ nhận được hàm dốc như chỉ ra trong hình 23.3(c). Hàm dốc $r(t)$ được biểu diễn như sau:

$$r(t) = \begin{cases} t, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{23.5}$$

Hơn nữa, tích phân *r(t)* sẽ nhận được tín hiệu parabol có dạng:

$$a(t) = \begin{cases} \frac{t^2}{2}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{23.6}$$

Tín hiệu parabol được mô tả trong hình 23.3(d).



HÌNH 23.3 Các hàm đặc biệt: (a) xung, (b) bậc thang, (c) dốc, (d) parabol

Các tín hiệu thời gian liên tục cơ bản

Hình 23.4 mô tả một số tín hiệu cơ bản thường gặp trong phân tích tín hiệu. Một số tín hiệu đó có thể nhận được trực tiếp từ các hàm đặc biệt đã được mô tả ở trên. Ví dụ, tín hiệu xung chữ nhật đơn vị từ $-\tau/2$ đến $\tau/2$ có thể được biểu diễn như sau:

$$\prod(t) = u\left(t + \frac{\tau}{2}\right) - u\left(t - \frac{\tau}{2}\right) \tag{23.7}$$

và điều này được mô tả trong hình 23.4(a).

Hàm tam giác ký hiệu là $\Lambda(t)$, được định nghĩa như sau:

$$\Lambda(t) = \begin{cases} 1 - |t|, & |t| < 1 \\ 0, & t \text{ còn lại} \end{cases} \tag{23.8}$$

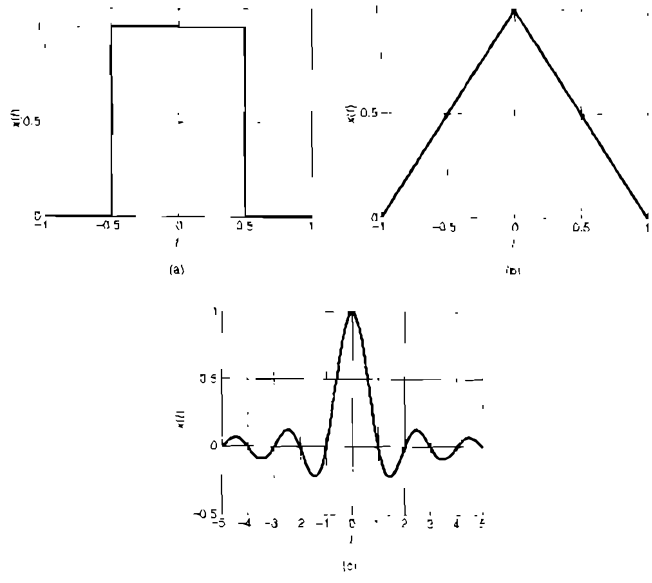
Sử dụng định lý tích chập, đã được giới thiệu trong mục "Phân tích các tín hiệu thời gian liên tục", có thể thấy rằng:

$$\Lambda(t) = \prod(t) * \prod(t)$$

Tín hiệu sin *c* được định nghĩa như sau:

$$\sin c(t) = \begin{cases} \dfrac{\sin(\pi t)}{\pi t}, & t \neq 0 \\ 1, & t = 0 \end{cases} \tag{23.9}$$

Tín hiệu tam giác và sin c đều được biểu diễn trong hình 23.4(b) và 23.4(c).



HÌNH 23.4 Biểu diễn các hàm CT cơ bản: (a) xung chữ nhật, (b) xung tam giác, (c) hàm sin c

Các tín hiệu thời gian rời rạc cơ bản

Chuỗi xung đơn vị ký hiệu là $\delta(n)$, được định nghĩa như sau:

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \text{ khác } 0 \end{cases}$$

Nó được gọi là chuỗi xung đơn vị hoặc hàm delta Kronecker. Các tính chất của chuỗi mẫu đơn vị tương tự như của $\delta(t)$ và chúng được biểu diễn như sau:

$$\sum_{n=-\infty}^{\infty} x(n)\delta(n-m) = x(m)$$

$$x(n)\delta(n-m) = x(m)\delta(n-m)$$

$$\delta(an \pm b) = \delta\left(n \pm \frac{b}{a}\right)$$

$$x(n) * \delta(n-m) = \sum_{n=-\infty}^{\infty} x(r)\delta(n-r-m) = x(n-m)$$

Chú ý rằng tính chất tỷ lệ chỉ được áp dụng khi cả a và b/a là các số nguyên. Hai tín hiệu cơ bản khác cũng thường được dùng để phân tích là tín hiệu bậc thang đơn vị và dốc đơn vị. Chuỗi nhảy đơn vị $u(n)$, được định nghĩa là:

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \tag{23.10}$$

Tín hiệu dốc đơn vị ký hiệu là *r(n)*, được định nghĩa như sau:

$$r(n) = \begin{cases} n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

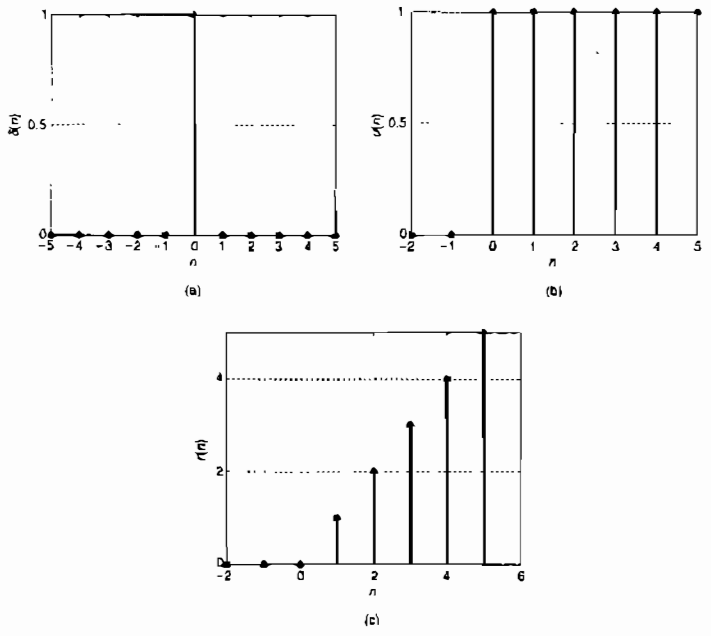
Ba chuỗi trên có quan hệ với nhau như sau:

$$\delta(n-k) = u(n-k) - u(n-k-1)$$

$$u(n) = \sum_{m=-\infty}^{n} \delta(n-m)$$

$$r(n) = u(n) * u(n-1)$$

Hình 23.5 biểu diễn các chuỗi DT trên.



HÌNH 23.5 Biểu diễn của các tín hiệu DT cơ bản: (a) chuỗi xung đơn vị, (b) chuỗi nhảy đơn vị, (c) chuỗi dốc

Phân tích các tín hiệu thời gian liên tục

Các phép toán cơ bản đối với các tín hiệu

Một số phép toán quan trọng thường được thực hiện đối với các tín hiệu để hiểu được các đặc trưng của chúng hoặc các hiện tượng vật lý sinh ra chúng. Ba phép toán chung nhất là: dịch, chia tỉ lệ thời gian và ánh xạ. Ví dụ về các phép toán này được minh họa trong hình 23.6, trong đó $x(t)$ được biểu diễn như sau:

$$x(t) = \begin{cases} t+1, & -1 \leq t \leq 3 \\ 3, & 3 < t \leq 6 \\ 0, & \text{t khác} \end{cases}$$

Tích chập và tương quan [2]

Mặc dù tích chập thường liên quan tới việc nghiên cứu các hệ thống, đôi khi phép toán này được dùng trong phân tích các tín hiệu nhận được từ một hệ thống vật lý. Tích chập của hai tín hiệu thời gian liên tục $x(t)$ và $y(t)$ sẽ cho $z(t)$, trong đó:

$$z(t) = x(t) * y(t) = \int_{-\infty}^{\infty} x(\tau) y(t-\tau) d\tau \tag{23.11}$$

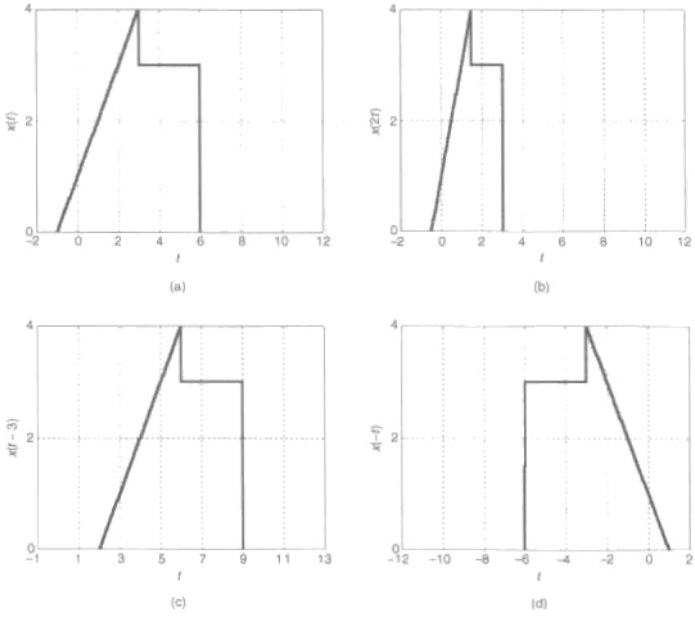
Tích chập không bị giới hạn trong miền thời gian vì nó cũng được dùng để xác định phổ tần số liên quan tới tích của hai tín hiệu thời gian. Hàm tương quan chéo giữa $x(t)$ và $y(t)$ ký hiệu là $R_{xy}(t)$, được định nghĩa như sau:

$$R_{xy}(t) = x(t) \oplus y(t) = \int_{-\infty}^{\infty} x(\tau) y^*(\tau - t) d\tau \tag{23.12}$$

Khác với tích chập, ở đây không có phép ánh xạ. Ngoài ra, trễ t là tham số để đo độ giống nhau giữa hai tín hiệu đó. Nếu $x(t) = y(t)$, thì (23.12) sẽ mô tả hàm tự tương quan. Một số tính chất của các hàm tương quan được trình bày trong bảng 23.2.

BẢNG 23.2 Các tính chất của hàm tương quan

Tính chất	Tự tương quan	Tương quan chéo
Chẵn	$R_{xx}(t) = R_{xx}(-t)$	$R_{xy}(t) = R_{yx}(-t)$
Giới hạn trên	$R_{xx}(0) \geq R_{xx}(t)$, cho mọi t	$\lvert R_{xy}(t) \rvert = \sqrt{R_{xx}(0) R_{yy}(0)}$



HÌNH 23.6 Các phép tính cơ bản của tín hiệu: (a) Tín hiệu gốc, (b) Phép tỷ lệ, (c) Phép dịch, (d) Phép ánh xạ

Tích phân của tích chập và tương quan đều có thể áp dụng cho các tín hiệu năng lượng cũng như công suất. Với các tín hiệu công suất, tích phân được thực hiện trên chu kỳ T và kết quả được chia cho T. Phân tích tương quan là quan trọng để tính mật độ phổ năng lượng cho các tín hiệu tức thời và mật độ phổ công suất cho cả các tín hiệu có chu kỳ và các tín hiệu ngẫu nhiên.

Phân tích Fourier của các tín hiệu CT

Ở đây mới chỉ bàn tới các phương pháp phân tích miền thời gian tín hiệu CT. Phép tích phân tích chập được quan tâm đặc biệt vì nó có thể được dùng để nghiên cứu cách một tín hiệu được biến đổi khi đi qua một hệ thống. Cũng cần phải xem xét các phương pháp phân tích miền tần số vì phân tích tích chập có thể là khó. Hơn nữa, công thức của tích chập dựa trên sự biểu diễn các tín hiệu bằng các hàm δ đã được dịch đi. Trong nhiều ứng dụng, phương pháp phân tích trong miền tần số sẽ thích hợp hơn để chọn một tập các hàm trực giao như các tín hiệu cơ bản vì phương pháp này làm giảm độ phức tạp trong tính toán số cũng như đưa ra cách biểu diễn bằng đồ họa cho các thành phần tần số của các tín hiệu.

Các hàm trực giao cơ bản [2, 3]

Sẽ thuận lợi về mặt toán học nếu biểu diễn các tín hiệu bất kỳ như một tổng có trọng số của các sóng trực giao, dẫn tới phép phân tích tín hiệu được đơn giản hóa rất nhiều cũng như chỉ ra sự giống nhau cơ bản giữa các tín hiệu và vectơ. Xét một tập hàm cơ bản $\phi(t)$, $i = 0, \pm1, \pm2, \ldots$ được gọi là trực giao trên khoảng (t_1, t_2) nếu:

$$\int_{t_1}^{t_2} \phi_m(t)\phi_k^*(t)dt = E_k\delta(m-k) \tag{23.13}$$

trong đó $\phi_k^*(t)$ là liên hợp phức của tín hiệu. Nếu E_k xấp xỉ giá trị đơn vị với mọi k thì $\phi(t)$ là một tập trực giao. Tương đối dễ dàng để xấp xỉ tín hiệu bằng một tập thích hợp các hàm trực giao, với điều kiện sai lệch là tối thiểu giữa tín hiệu thực và xấp xỉ của nó. Vì vậy, một tín hiệu $x(t)$ đã cho với năng lượng hữu hạn trong khoảng $t_1 < t < t_2$ có thể được biểu diễn như sau:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k\phi_k(t) \tag{23.14}$$

trong đó:

$$c_k = \int_{t_1}^{t_2} x(t)\phi_k^*(t)dt, \qquad k = 0, \pm1, \pm2, \ldots$$

Phương trình này được gọi là chuỗi Fourier tổng quát của $x(t)$, với các hằng số c_k, $k = 0, \pm1, \pm2, \ldots$, được gọi là các hệ số của chuỗi Fourier tương ứng với tập trực giao $\{\phi(t)\}$.

Ký hiệu M số hạng đầu tiên trong công thức (23.14) là $\hat{x}(t)$, thì hàm sai lệch sẽ là:

$$e_M(t) = x(t) - \sum_{k=0}^{M} c_k\phi_k^*(t) \tag{23.15}$$

Bằng cách tính công suất trung bình của hàm sai lệch này và đặt các đạo hàm của nó theo c_k bằng 0, ta nhận được một tập tối ưu $\{c_k\}$ để tối thiểu sai lệch năng lượng. Nếu $\lim_{M\to\infty}\hat{x}(t) = x(t)$ thì các hàm cơ bản là đầy đủ, có nghĩa là năng lượng sai lệch bằng 0. Với các tín hiệu có chu kỳ, khoảng thời gian (t_1, t_2) bằng chu kỳ T của tín hiệu. Ngoài ra, $\phi_n(t) = \exp\{jn\omega_0\}$ thường được chọn như tập các hàm cơ bản với $n = 0, \pm1, \pm2, \ldots$ và $\omega_0 = 2\pi/T$. Các phương pháp tính các hệ số của chuỗi Fourier được xem xét tiếp sau đây.

Chuỗi Fourier lũy thừa phức

Cho tín hiệu $x(t)$ thỏa mãn $x(t) = x(t + T)$ và thỏa mãn các điều kiện Dirichlet sau [3]:

1. $x(t)$ là khả tích trong khoảng chu kỳ của nó, có nghĩa là:

$$\int_{t_1}^{t_1+T} |x(t)|dt < \infty$$

2. Số cực đại và cực tiểu của $x(t)$ ở mỗi chu kỳ là hữu hạn,
3. Số các điểm gián đoạn của $x(t)$ trong mỗi chu kỳ là hữu hạn, do đó $x(t)$ có thể được khai triển như sau:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}, \quad \omega_0 = 2\pi f_0 \tag{23.16}$$

trong đó :

$$c_k = \frac{1}{T}\int_{t_1}^{t_1+T} x(t)e^{-jk\omega_0 t}\,dt \tag{23.17}$$

với mọi giá trị t_1.

Các hệ số c_k được gọi là các hệ số của chuỗi Fourier phức của tín hiệu $x(t)$, nói chung, chúng có thể là các số phức.

Các điều kiện Dirichlet chỉ là điều kiện đủ để tồn tại khai triển Fourier. Vẫn có thể nhận được khai triển Fourier của các tín hiệu không thỏa mãn các điều kiện này.

Các hệ số của chuỗi Fourier phức có thể được xác định bằng cách đánh giá trực tiếp tích phân trong (23.17) hoặc bằng cách dùng phương pháp vi phân. Phương pháp sau phụ thuộc vào việc lấy vi phân $x(t)$ với một số lần nào đó để tạo ra một dãy xung. Cả hai phương pháp xác định c_k này sẽ được minh họa ở dưới đây.

Xét các tín hiệu có chu kỳ đã được chỉ ra trong hình 23.7, chúng được biểu diễn như sau:

$$x(t) = \frac{1}{T}\sum_{n=-\infty}^{\infty} \Pi\left(\frac{t-nT}{\tau}\right)$$

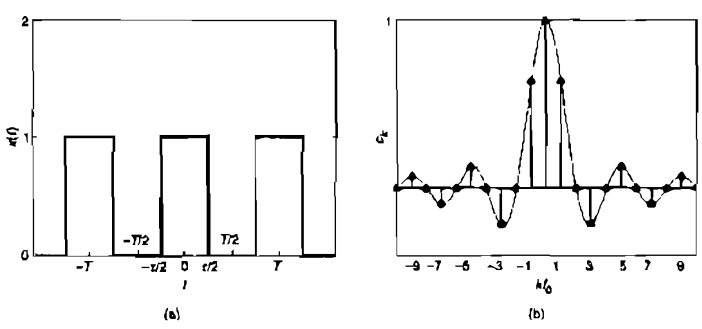
Thay vào công thức (23.17) chúng ta có:

$$\begin{aligned} c_k &= \frac{1}{T}\int_{-T/2}^{T/2} x(t)e^{-j2\pi kf_0 t}\,dt \\ &= \frac{1}{T}\int_{-T/2}^{T/2} e^{-j2\pi kf_0 t}\,dt \\ &= \frac{1}{\pi k}\sin(k\pi f_0\tau) \\ &= \tau f_0 \sin c(kf_0\tau) \end{aligned}$$

Vì vậy:

$$x(t) = \sum_{n=-\infty}^{\infty} \tau f_0 \sin c(kf_0\tau)e^{j2\pi kf_0 t}$$

Biểu đồ phổ biên độ và pha của các hệ số của chuỗi Fourier phức được chỉ ra trong hình 23.7(b) với τ/T thay đổi.



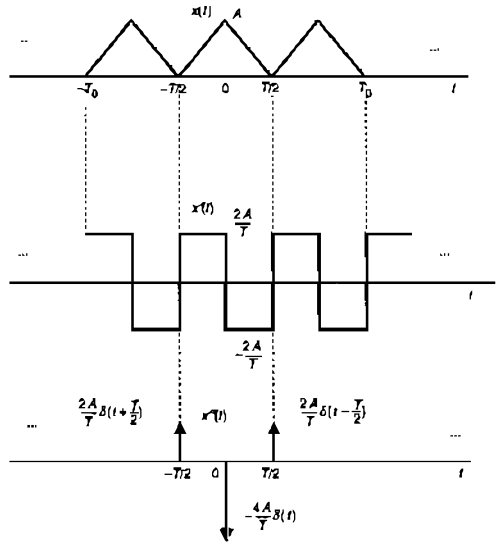
HÌNH 23.7 Tín hiệu chữ nhật có chu kỳ và các hệ số của chuỗi Fourier của nó

Tính các hệ số của chuỗi Fourier phức của tín hiệu đã được chỉ ra trên hình 23.8(a) bằng phương pháp vi phân. Tín hiệu này được khai triển theo công thức (23.16). Lấy vi phân phương trình này 2 lần theo t sẽ nhận được:

$$x''(t) = \sum_{k=-\infty}^{\infty} (j2\pi k f_0)^2 c_k e^{j2\pi k f_0 t}$$

có thể viết lại như sau:

$$x''(t) = \sum_{k=-\infty}^{\infty} \beta_k e^{j2\pi k f_0 t}$$



HÌNH 23.8 Minh họa phương pháp vi phân để tính hệ số của chuỗi Fourier

Hình 23.8(c) chỉ ra kết quả của phép vi phân $x(t)$. Chú ý rằng nếu tín hiệu có chu kỳ thì đạo hàm của nó cũng có chu kì. Điều này có nghĩa rằng β_k là hệ số của chuỗi Fourier phức của $x''(t)$ và có thể được tính từ:

$$\beta_k = \frac{1}{T}\int_{-T/2}^{T/2} x''(t) e^{-j2\pi k f_0 t}\, dt$$

trong đó:

$$x''(t) = 2\frac{A}{T}\left[\delta\left(t+\frac{T}{2}\right) - 2\delta(t) + \delta\left(t-\frac{T}{2}\right)\right]$$

Vì vậy:

$$\beta_k = -8\frac{A}{T^2}\sin^2\left(\frac{\pi k T f_0}{2}\right) = (j2\pi f_0 k)^2 c_k$$

Có nghĩa là:

$$c_k = \begin{cases} \dfrac{2A}{\pi^2 k^2}, & k \text{ lẻ} \\ 0, & \text{còn lại} \end{cases}$$

Và
$$c_0 = \frac{1}{T}\int_{-T/2}^{T/2} x(t)dt = \frac{A}{2}$$

Phương pháp xác định c_k này dễ hơn phương pháp trực tiếp.

Chuỗi Fourier cho các tín hiệu thực [3]

Một tín hiệu thời gian thực $x(t)$ thỏa mãn các điều kiện Dirichlet có thể khai triển theo các bước sau. Trong công thức (23.17), thay k bằng $-k$, chúng ta nhận được:

$$c_{-k} = \frac{1}{T}\int_{t_1}^{t_1+T} x(t)e^{j2\pi kf_0 t}dt = \left[\frac{1}{T}\int_{t_1}^{t_1+T} x(t)e^{-j2\pi kf_0 t}dt\right]^* = c_k^* \tag{23.18}$$

Điều này có nghĩa là với các tín hiệu thực, các hệ số âm và dương là liên hợp phức với nhau. Với các tín hiệu như vậy, $|c_k|$ có đối xứng chẵn và $\angle c_k$ có đối xứng lẻ với $k = 0$. Đặt $c_k=(a_k-jb_k)/2$, thì $c_{-k}=(a_k+jb_k)/2$, vì vậy:

$$c_k e^{j2\pi kf_0 t} + c_{-k}e^{-j2\pi kf_0 t} = \frac{a_k - jb_k}{2}e^{j2\pi kf_0 t} + \frac{a_k + jb_k}{2}e^{-j2\pi kf_0 t}$$

Vì c_0 là thực và được định nghĩa là $c_0=a_0/2$, nên chúng ta có thể viết:

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}[a_k\cos(2\pi kf_0 t) + b_k\sin(2\pi kf_0 t)] \tag{23.19}$$

Công thức này chỉ đúng cho tín hiệu thực có chu kỳ $x(t)$ và được gọi là khai triển Fourier lượng giác.

Vì

$$\begin{aligned} c_k = \frac{a_k - jb_k}{2} &= \frac{1}{T}\int_{t_1}^{t_1+T} x(t)e^{-j2\pi kf_0 t}dt \\ &= \frac{1}{T}\int_{t_1}^{t_1+T} x(t)[\cos(2\pi kf_0 t) - j\sin(2\pi kf_0 t)]dt \end{aligned} \tag{23.20}$$

Nên

$$a_k = \frac{2}{T}\int_{t_1}^{t_1+T} x(t)\cos(2\pi kf_0 t)dt \tag{23.21a}$$

$$b_k = \frac{2}{T}\int_{t_1}^{t_1+T} x(t)\sin(2\pi kf_0 t)dt \tag{23.21b}$$

và

$$a_0 = \frac{2}{T}\int_{t_1}^{t_1+T} x(t)dt$$

Phương pháp thứ ba để khai triển Fourier cho các tín hiệu thực được đưa ra bởi:

$$x(t) = c_0 + \sum_{k=1}^{\infty} c_k\cos(2\pi kf_0 t + \theta_k) \tag{23.22}$$

trong đó c_0 là thành phần đầu tiên, c_k và θ_k biểu diễn biên độ và góc pha của thành phần điều hòa thứ k tương ứng. Công thức (23.22) được gọi là dạng điều hòa của khai triển Fourier của $x(t)$. Các tham số c_k và θ_k liên quan với a_k và b_k như sau:

$$c_0 = \frac{a_0}{2}, \qquad c_k = \sqrt{a_k^2 + b_k^2}, \qquad \theta_k = \tan^{-1}\frac{b_k}{a_k}$$

Các tính chất của chuỗi Fourier [1, 4]

Hiểu được tính đối xứng của tín hiệu có thể đơn giản hóa việc tính toán các hệ số Fourier phức của nó. Trong các dạng đối xứng đã được thiết lập có một số kiểu đối xứng quan trọng thường gặp trong phân tích tín hiệu là:

- Đối xứng chẵn $x(t)= x(-t)$
- Đối xứng lẻ $x(t)= -x(t)$
- Đối xứng lẻ nửa sóng $x(t)= -x(t+T/2)$

Ảnh hưởng của tính đối xứng tới việc tính toán chuỗi fourier được giới thiệu trong bảng 23.3. Các tính chất khác của chuỗi Fourier được tóm tắt trong bảng 23.4, trong đó α_k và β_k là các hệ số phức của chuỗi Fourier của $x(t)$ và $y(t)$ tương ứng.

BẢNG 23.3 Các điều kiện đối xứng của chuỗi Fourier

Kiểu đối xứng	Hệ số của chuỗi Fourier thực	Hệ số của chuỗi Fourier phức	Ghi chú
Đối xứng chẵn $x(t)=x(-t)$	$a_k = \frac{4}{T}\int_0^{T/2} x(t)\cos(2\pi k f_0 t)dt$ $b_k = 0$	$c_k = \frac{1}{2}a_k$ c_k có giá trị thực	Pha của c_k là 0 hoặc π
Đối xứng lẻ $x(t)=-x(-t)$	$a_k = 0$ $b_k = \frac{4}{T}\int_0^{T/2} x(t)\sin(2\pi k f_0 t)dt$	c_k có các giá trị ảo	Pha của c_k là $\pi/2$ hoặc $-\pi/2$
Đối xứng chẵn nửa sóng $x(t) = x\left(t+\frac{T}{2}\right)$	a_{2k} và b_{2k} có thể có giá trị khác 0 nhưng $a_{2k-1}=0$, $b_{2k+1}=0$	$c_{2k} \neq 0$ $c_{2k+1} = 0$	
Đối xứng lẻ nửa sóng	a_{2k+1} và b_{2k+1} có thể có giá trị khác 0 nhưng $a_{2k}=0$, $b_{2k}=0$	$c_{2k} = 0$ $c_{2k+1} \neq 0$	

BẢNG 23.4 Các tính chất của chuỗi Fourier

Tính chất	Mô tả tín hiệu	Các hệ số của chuỗi Fourier
Tuyến tính	$ax(t) + by(t)$; a, b là hằng số	$a\alpha_k + b\beta_k$
Nhân	$x(t)y(t)$	$\alpha_k * \beta_k$
Tích chập	$x(t) * y(t)$	$\alpha_k\beta_k$
Định lý Parseval	$\frac{1}{T}\int_{t_1}^{t_1+T} \lvert x(t)^2 \rvert dt$	$\sum_{k=-\infty}^{\infty} \lvert\alpha_k\rvert^2$
Dịch theo thời gian	$x = (t \pm \tau)$	$\alpha_k e^{\pm j2\pi f_0 \tau}$
Vi phân	$\frac{d^n}{dt^n}x(t)$	$(j2\pi k f_0)^n \alpha_k$
Tích phân	$\int_{-\infty} x(\tau)d\tau$	$(j2\pi k f_0)^{-1}\alpha_k$, $\alpha_0 = 0$

Biến đổi Fourier

Phương pháp phân tích trong miền tần số các tín hiệu thời gian CT đã được trình bày trong mục trước. Một phương pháp khác được gọi là biến đổi Fourier, được dùng để phân tích các tín hiệu không có chu kỳ. Khai triển của biến đổi Fourier dựa trên các công thức (23.16) và (23.17). Thay (23.17) vào (23.16) sẽ có :

$$x(t) = \sum_{k=-\infty}^{\infty} \left(\frac{1}{T} \int_{T/2}^{T/2} x(\lambda) e^{-j2\pi k f_0 \lambda} d\lambda \right) e^{j2\pi k f_0 t} \tag{23.33}$$

Cho $T \to \infty$, thì $1/T \to df$, $kf_0 \to f$ và công thức (23.23) trở thành:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \tag{23.24a}$$

và

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \tag{23.24b}$$

Các tính chất của biến đổi Fourier [5, 6]

Một số tính chất cơ bản của biến đổi Fourier thường được dùng trong phân tích tín hiệu được đưa ra trong bảng 23.5.

BẢNG 23.5 Các tính chất của biến đổi Fourier

Tính chất	Mô tả tín hiệu	Biến đổi Fourier
Tuyến tính	$ax(t) + by(t)$; a, b là hằng số	$aX(f) + bY(f)$
Chẵn và lẻ	$x(-t) = x(t)$	$X(f) = 2\int_0^{\infty} x(t)\cos(2\pi ft)dt$
	$x(-t) = -x(t)$	$X(f) = -2\int_0^{\infty} x(t)\sin(2\pi ft)dt$
Dịch thời gian	$x(t-\tau)$	$e^{-j2\pi f\tau} X(f)$
Cân chỉnh thời gian	$x(at)$	$\frac{1}{\lvert a\rvert} X\left(\frac{f}{a}\right)$
Đảo thời gian	$x(-t)$	$X^*(f)$
Đối ngẫu	$X(t)$	$x(-f)$
Tích chập thời gian	$x(t) * y(t)$	$X(f)Y(f)$
Tích chập tần số	$x(t)y(t)$	$X(f) * Y(f)$
Điều biến	$x(t)e^{j2\pi f_0 t}$	$X(f - f_0)$
Vi phân thời gian	$\frac{d^n}{dt^n} x(t)$	$(j2\pi f)^n X(f)$
Vi phân tần số	$t^n x(t)$	$(\frac{j}{2\pi})^n \frac{d^n}{df^n} X(f)$
Tích phân	$\int_{-\infty}^{t} x(\tau)d\tau$	$\frac{1}{j2\pi f} X(f) + \frac{1}{2} X(0)\delta(f)$
Tương quan	$R_{xy}(\tau) = \int_{-\infty}^{\infty} y(t)x(t-\tau)dt$	$Y(-f)X(f)$
Định lý Parseval	$\int_{-\infty}^{\infty} \lvert x(t)\rvert^2 dt$	$\int_{-\infty}^{\infty} \lvert X(f)\rvert^2 df$

Cặp biến đổi Fourier của các tín hiệu cơ bản sau cũng thường được dùng để phân tích tín hiệu:

$$K \leftrightarrow K\delta(f)$$

$$\text{sgn}(t) \leftrightarrow \frac{1}{j\pi f}$$

$$u(t) \leftrightarrow \frac{1}{2}\delta(f) + \frac{1}{j2\pi f}$$

$$\cos(2\pi f_0 t) \leftrightarrow \frac{1}{2}\delta(f - f_0) + \frac{1}{2}\delta(f + f_0)$$

Để xác định biến đổi Fourier của một tín hiệu có chu kỳ cần thực hiện hai bước. Nếu $x(t)$ là một tín hiệu có chu kỳ với chu kỳ T, thì $x(t)$ có thể được khai triển Fourier như sau:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi f_0 t}, \qquad f_0 = \frac{1}{T} \tag{23.25}$$

Áp dụng tính chất tuyến tính với công thức này sẽ có:

$$X(f) = \sum_{k=-\infty}^{\infty} c_k \delta(f - kf_0) \tag{23.26}$$

Đó là biến đổi Fourier của tín hiệu $x(t)$ có chu kỳ bất kỳ. Công thức (23.26) giải thích sự khác nhau giữa phổ tần số được đưa ra bởi phép phân tích Fourier và mật độ phổ biên độ được đưa ra bởi phép biến đổi Fourier. Do đó, phổ tần số là một cách biểu diễn của c_k đối với kf_0, trong khi mật độ phổ biên độ là cách biểu diễn liên tục của phổ mật độ biên độ, mà trong trường hợp này nó có dạng xung, đúng hơn là số.

Tương tự biến đổi Fourier của một dãy xung có dạng:

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{23.27}$$

là:

$$P(f) = \frac{1}{T}\sum_{k=-\infty}^{\infty} \delta(f - kF_s), \qquad F_s = \frac{1}{T} \tag{23.28}$$

Mật độ phổ năng lượng và công suất [6]

Giả thiết $x(t)$ là một tín hiệu không có chu kỳ có biến đổi Fourier là $X(f)$, thì năng lượng của nó sẽ là:

$$E = R_{xx}(0) = \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df \tag{23.29}$$

Đây là định lý Parseval và nó chỉ ra rằng nguyên lý bảo tồn năng lượng theo thời gian và tần số là đúng. Phổ biên độ $X(f)$ có thể được biểu diễn là:

$$X = (f) = |X(f)| \angle X(f)$$

và ký hiệu

$$S_{xx}(f) = |X(f)|^2$$

thì năng lượng tổng của tín hiệu sẽ là:

$$E = \int_{-\infty}^{\infty} S_{xx}(f)df \tag{23.30}$$

trong đó $S_{xx}(f)$ biểu diễn sự phân bố của năng lượng tín hiệu như một hàm của tần số. $S_{xx}(f)$ được gọi là mật độ phổ năng lượng của tín hiệu năng lượng hữu hạn $x(t)$.

Xét một tín hiệu chu kỳ $x(t)$ có hàm tự tương quan là:

$$R_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)x^*(t-\tau)dt$$

thì:

$$R_{xx}(0) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 \, dt \tag{23.31}$$

là công suất của tín hiệu. Giống như với các tín hiệu năng lượng, ta định nghĩa $S_{xx}(f)$ là biến đổi Fourier của $R_{xx}(\tau)$, sao cho:

$$P = R_{xx}(0) = \int_{-\infty}^{\infty} S_{xx}(f)df \tag{23.32}$$

và $S_{xx}(f)$ được gọi là mật độ phổ công suất của tín hiệu chu kỳ $x(t)$.

Trong nhiều trường hợp thực tế, cần phải phân tích các tín hiệu ngẫu nhiên dừng. Tính chất của các tín hiệu như vậy có thể được suy ra từ các hàm tương quan của chúng. Ví dụ: hàm tự tương quan $\phi_{xx}(\tau)$ của một tín hiệu ngẫu nhiên dừng sẽ giảm tới 0 khi τ tăng vì các sự kiện trở nên không tương quan do khoảng thời gian lớn. Vì vậy, $\phi_{xx}(\tau) = \phi_{xx}(-\tau)$ và tồn tại biến đổi Fourier.

Do vậy chúng ta có thể viết:

$$\phi_{xx}(0) = \int_{-\infty}^{\infty} \Gamma_{xx}(f)df \tag{23.33}$$

trong đó $\Gamma_{xx}(f)$ và $\phi_{xx}(0)$ tương ứng biểu diễn mật độ phổ công suất và công suất trung bình của một quá trình ngẫu nhiên.

Các tín hiệu thời gian liên tục được lấy mẫu

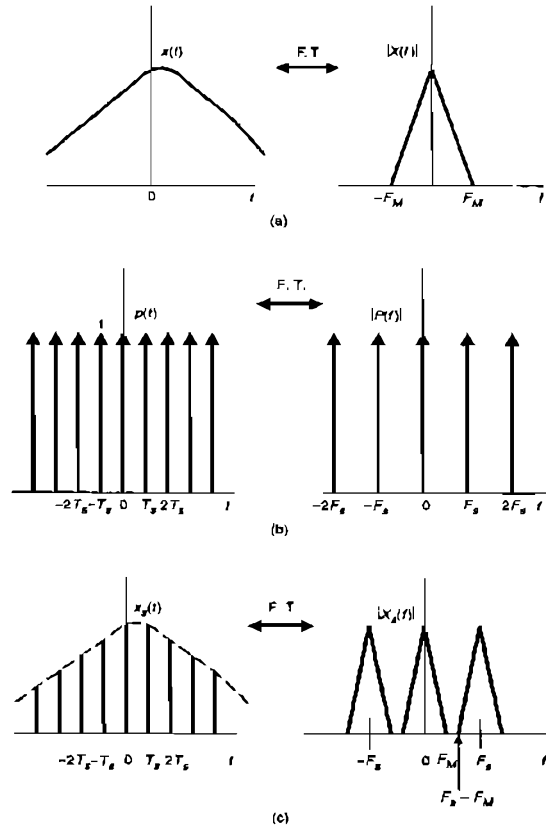
Các tín hiệu thời gian rời rạc (DT) xuất hiện một cách tự nhiên hoặc bằng cách lấy mẫu các tín hiệu thời gian liên tục (CT). Tuy nhiên, dạng sau thường gặp trong thực tế hơn. Trong trường hợp này, tín hiệu số đã được tạo ra từ một tín hiệu CT thông qua quá trình biến đổi tương tự-số. Giai đoạn đầu tiên của quá trình này là lấy mẫu tín hiệu tương tự, điều đó có nghĩa là biến đổi $x(t)$ thành $x(nT_s)$, trong đó T_s là chu kỳ lấy mẫu và nghịch đảo của nó $F_s = 1/T_s$ là tần số lấy mẫu tính theo số mẫu/s. Tần số lấy mẫu phải được chọn thích hợp để tránh méo phổ và như vậy đảm bảo $x(t)$ có thể được khôi phục từ các mẫu của nó. Để hiểu thêm về giai đoạn này, chúng ta sẽ khảo sát quá trình lấy mẫu ở miền tần số.

Lấy mẫu xung [6 - 9]

Xét quá trình lấy mẫu xung lý tưởng được mô tả trên hình 23.9, trong đó $x(t)$ được lấy mẫu bằng cách dùng một chuỗi xung $p(t)$, vì vậy:

$$x_s(t) = x(t)p(t) = x(t)\sum_{n=-\infty}^{\infty} \delta(t-nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t-nT_s) \tag{23.34}$$

Sự khác nhau chính giữa $x_s(t)$ và $x(nT_s)$ là $x_s(t)$ bản chất là tín hiệu CT bằng 0 tại tất cả các điểm trừ tại các giá trị nguyên của T_s, trong khi $x(nT_s)$ là mẫu của $x(t)$, là kết quả của việc lấy mẫu xung.



HÌNH 23.9 Quá trình lấy mẫu xung lý tưởng: (a) tín hiệu thời gian liên tục giới hạn và biến đổi Fourier của nó, (b) dãy xung và phổ của nó, (c) tín hiệu được lấy mẫu và phổ của nó

Biến đổi Fourier của công thức (23.34) sẽ cho:

$$X_s(f) = X(f) * P(f) = F_s \sum_{k=-\infty}^{\infty} X(f - F_s) \qquad (23.35)$$

trong đó $X(f)$ là phổ của $x(t)$.

Nhận thấy rằng $X_s(f)$ bao gồm một phiên bản của $X(f)$ đã được lặp lại có chu kỳ và tỷ lệ với chu kỳ F_s. Như chỉ ra trong hình 23.9(c), hiển nhiên rằng khi $F_s - F_M \geq F_M$ thì sẽ không có sự chồng nhau của phổ và tín hiệu $x(t)$ có thể được khôi phục hoàn toàn từ $x_s(t)$. Tuy nhiên, nếu $F_s - F_M < F_M$ thì các bản sao của $X(f)$ sẽ chồng lên nhau, dẫn tới phổ bị méo và vì vậy có thể không khôi phục được $x(t)$ từ mẫu của nó. Vì vậy, để khôi phục được $x(t)$ từ các mẫu của nó, thì tần số lấy mẫu phải là:

$$F_s - F_M \geq F_M$$

có nghĩa là:

$$F_s \geq 2F_M$$

Đây được gọi là định lý lấy mẫu Nyquist. Tần số lấy mẫu tối thiểu $F_s = 2F_M$ được gọi là tần số Nyquist. Việc lấy mẫu một tín hiệu ở tần số nhỏ hơn tần số Nyquist sẽ dẫn đến méo phổ được gọi là méo răng cưa. Ngoài ra, việc lấy mẫu một tín hiệu ở tần số Nyquist nhỏ nhất có nghĩa rằng một bộ lọc thông thấp lý tưởng (LPF) với một hệ số khuếch đại $1/F_s$ và tần số cắt F_c có thể được dùng để khôi phục phổ ban đầu của nó, trong đó $F_M \leq F_c \leq F_s - F_M$.

Giả sử chúng ta muốn khôi phục $x(t)$ từ các mẫu của nó. Giả thiết rằng $X(f)$ là phổ của $x(nT_s)$, không có méo răng cưa, như mô tả trong hình 23.9(c). Như vậy:

$$X_a(f) = \begin{cases} \dfrac{1}{F_s} X(f), & |f| \le \dfrac{F_s}{2} \\ 0, & |f| > \dfrac{F_s}{2} \end{cases} \tag{23.36}$$

Chú ý rằng:

$$X_a(f) = \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j2\pi fT_s} \tag{23.37}$$

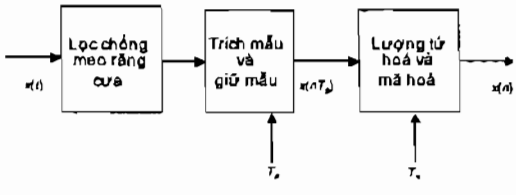
vì vậy biến đổi ngược Fourier sẽ là:

$$\begin{aligned} x_a(t) &= \frac{1}{F_s} \sum_{n=-\infty}^{\infty} x(nT_s) \int_{-F_s/2}^{F_s/2} e^{-j2\pi f(t-nT_s)} df \\ &= \sum_{n=-\infty}^{\infty} x(nT_s) \frac{\sin[\pi(t-nT_s)/T_s]}{\pi(t-nT_s)/T_s} \end{aligned} \tag{23.38}$$

Đây là công thức để khôi phục $x(t)$ từ các mẫu của nó. Điều đó có nghĩa là $x(t)$ được tạo ra bằng cách nhân hàm đã được dịch một cách thích hợp $g(t) = \text{sinc}(tF_s)$ với các mẫu tương ứng của $x(nT_s)$.

Việc lấy mẫu thực tế [8 - 10]

Việc lấy mẫu ở trên dựa vào các mô hình lấy mẫu xung có chu kỳ lý tưởng và nội suy dải giới hạn. Trong thực tế, các tín hiệu CT không được giới hạn dải một cách chính xác như các tín hiệu xung và các bộ lọc thông thấp lý tưởng không tồn tại. Hình 23.10 biểu diễn sơ đồ khối để chuyển đổi các tín hiệu thời gian liên tục thành dạng rời rạc. Tín hiệu thời gian liên tục đầu tiên được lọc, lấy mẫu, lượng tử hóa và cuối cùng được mã hóa thành các từ có độ dài hữu hạn được gọi là các bit *b*. Bộ lọc được gọi là lọc chống méo răng cưa (AAF), là một bộ lọc thông thấp để giới hạn độ rộng dải tần số của tín hiệu đầu vào tới $F_s/2$ trước khi lấy mẫu để loại méo tín hiệu. Trong thực tế bộ lọc này có các đặc tính không lý tưởng, vì vậy nó phải được thiết kế đủ để suy giảm các tín hiệu đầu vào ở tần số trên tần số Nyquist tới mức không thể phát hiện được bởi bộ biến đổi tương tự số (ADC).



HÌNH 23.10 Lấy mẫu thực các tín hiệu thời gian liên tục

Các tín hiệu đã lọc được đưa vào ADC để biến đổi thành tín hiệu DT. ADC có sẵn một mạch lấy mẫu và giữ mẫu hoạt động với tốc độ lấy mẫu F_s. Tuy nhiên, hàm lấy mẫu có độ rộng hữu hạn trái với việc lấy mẫu xung đã bàn ở trên. Phép lấy mẫu có thể được mô hình hóa bằng bộ lấy mẫu xung có độ rộng hữu hạn được mô tả trong hình 23.11(b), trong đó cổng lấy mẫu trong một chu kỳ T_s giây sẽ mở τ giây và ngắn mạch trong phần còn lại của khoảng lấy mẫu. Ở đây $p(t)$ được biểu diễn như sau:

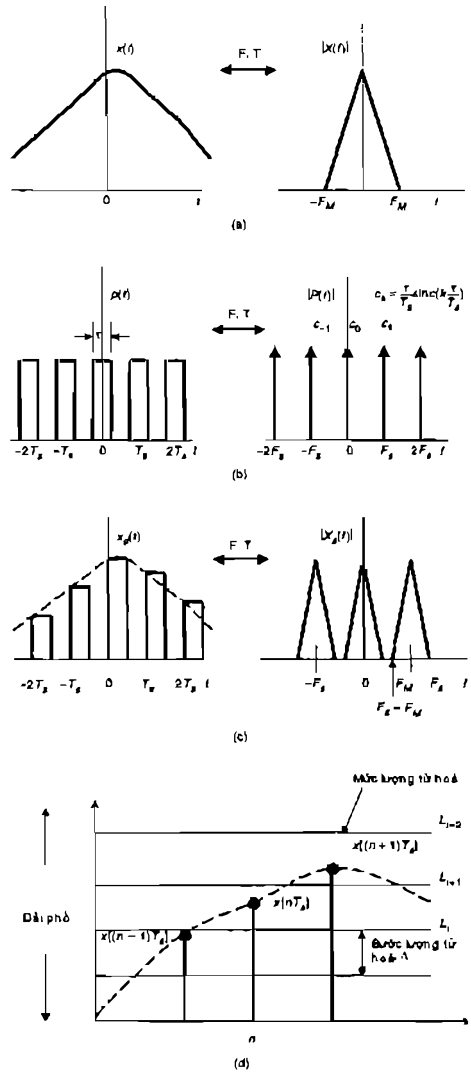
$$p(t) = \sum_{n=-\infty}^{\infty} \Pi\left(\frac{t-nT_s}{\tau}\right)$$

nó có thể được khai triển chuỗi Fourier như sau:

$$p(t) = \sum_{n=-\infty}^{\infty} c_k e^{j2\pi k f_s t}$$

trong đó:

$$c_k = \frac{\tau}{T_s} \sin c\left(k\frac{\tau}{T_s}\right)$$



HÌNH 23.11 Các tín hiệu lấy mẫu xung với độ rộng hữu hạn và phổ của nó: (a) Tín hiệu có dải giới hạn và phổ của nó, (b) Chuỗi xung có độ rộng hữu hạn và biến đổi Furier của nó, (c) Tín hiệu lấy mẫu và phổ của nó, (d) Quá trình lượng tử hóa

Biến đổi Fourier của các tín hiệu lấy mẫu có thể được viết như sau:

$$X_s(f) = \sum_{n=-\infty}^{\infty} c_k X(f - kF_s)$$

Chú ý rằng c_k không phải là hằng số trong biểu thức này (trái với việc lấy mẫu xung) vì giá trị của nó phụ thuộc vào số điều hòa (k) cũng như tỷ số τ/T_s. Tín hiệu rời rạc theo thời gian, $x(nT_s)$, được đưa tới bộ lượng tử hóa mà ở đó mỗi mẫu trích được biến đổi thành một trong các tập hữu hạn gần nhất của các giá trị quy định, đó là $\hat{x}(n) = Q(x(nT_s))$, trong đó $\hat{x}(n)$ là mẫu đã được lượng tử hóa. Quá trình lượng tử hóa được mô tả trong hình 23.11(d) cho bộ biến đổi tương tự-số trích và giữ mẫu bậc 0 (ADC), trong đó L_i là ký hiệu mức lượng tử và Δ là bước lượng tử. Sai số lượng tử (hoặc là ồn) của quá trình này là:

$$L_i - \frac{\Delta}{2} < x(nT_s) < L_i + \frac{\Delta}{2}$$

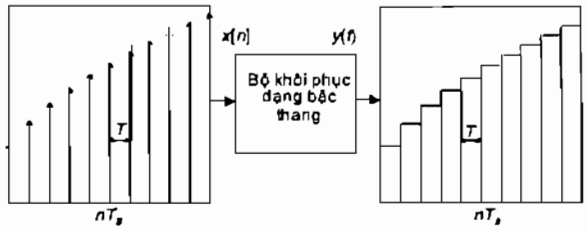
Từ những nghiên cứu thống kê cho thấy công suất nhiễu là $\Delta^2/12$ W. Hiệu suất của ADC là tỉ số của công suất tín hiệu với công suất nhiễu và được gọi là công suất của tín hiệu đối với nhiễu lượng tử hóa được biểu diễn theo decibel (dB) như sau:

$$\text{SQNR(dB)} = 1.76 + 6.02b$$

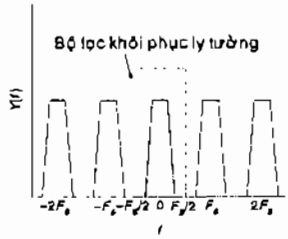
Biến đổi số - tương tự [8 - 12]

Việc khôi phục tín hiệu tương tự từ dạng lấy mẫu của nó gần giống với việc lọc thông thấp tín hiệu đã được lấy mẫu. Hình 23.12 chỉ ra một tín hiệu tương tự có thể được khôi phục bằng cách đổ đầy các khe giữa các mẫu và việc giữ giá trị hiện thời không đổi tới khi mẫu tiếp theo được nhận như thế nào. Xét một bộ lọc khôi phục lý tưởng với một hàm đáp ứng xung $h(t)$, thì đáp ứng của nó sẽ là:

$$y(t) = \sum_{n=-\infty}^{\infty} x(nT_s)h(t - nT_s)$$



HÌNH 23.12 Quá trình biến đổi số - tương tự



HÌNH 23.13 Đáp ứng tần số của bộ lọc khôi phục lý tưởng

Lấy biến đổi Fourier của công thức này sẽ có:

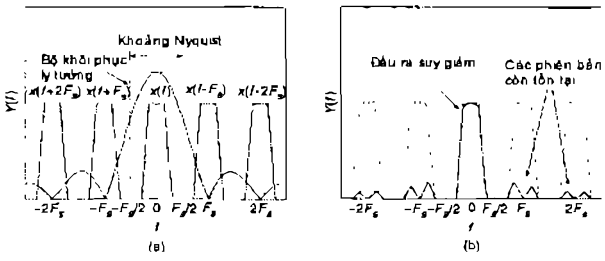
$$Y(f) = H(f)X(f)$$

trong đó $X(f)$ là phổ tuần hoàn của $x(nT_s)$ như chỉ ra trong hình 23.13 và $h(t)$=sinc$(F_s t)$. Chú ý rằng $h(t)$ là không nhân quả; vì vậy nó không thể được sử dụng cho các ứng dụng thời gian thực. Hơn nữa vì $h(t)$ không bị giới hạn về thời gian, nên một số vô hạn các đáp ứng xung được dùng để nội suy giữa các giá trị để nhận được các kết quả chính xác. Do vậy các bộ lọc khôi phục ví dụ như bộ giữ bậc 0 (bậc thang), bậc 1, hoặc các bộ giữ bậc phân số được sử dụng trong

thực tế. Tuy nhiên, bộ lọc khôi phục kiểu bậc thang nhìn chung là đơn giản nhất và được sử dụng rộng rãi nhất trong thực tế.

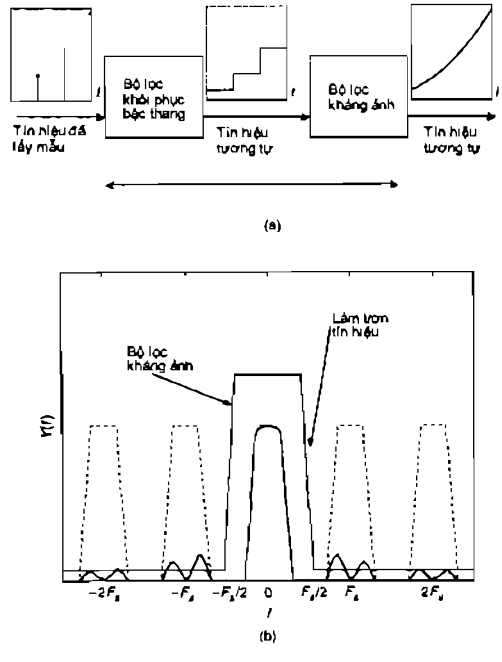
Đáp ứng xung của bộ lọc này được đưa ra như $h_0(t)=u(t)-u(t-T_s)$ vì nó phải có một khoảng T_s giây để đổ đầy vào khe hở giữa các mẫu. Vì vậy, trong thực tế bộ lọc này sẽ tạo ra một xấp xỉ bậc thang với tín hiệu tương tự gốc. Đáp ứng tần số của bộ lọc là:

$$H_0(f) = T\frac{\sin(\pi fT_s)}{\pi fT_s}e^{-j\pi fT_s}$$



HÌNH 23.14 (a) Khôi phục kiểu bậc thang ở miền tần số, (b) Kết quả khôi phục bậc thang ở miền tần số

Tuy nhiên đầu ra của bộ lọc khôi phục bậc thang trơn hơn dạng lấy mẫu của nó, xem hình 23.14(a), sẽ chứa các thành phần tần số cao giả mạo do các bước nhảy đột ngột ở các mức thang khi các giá trị lấy mẫu khác nhau được xét. Ngoài ra việc giữ mỗi xung của $x(nT_s)$ tới T_s giây sẽ dẫn tới sự thay đổi về thời gian $T_s/2$ đối với tín hiệu ra. Tuy nhiên, trễ thời gian này hầu như không ảnh hưởng tới chất lượng của tín hiệu ra. Hình 23.14(b) sẽ so sánh các phổ tín hiệu trước và sau khi qua bộ lọc khôi phục kiểu bậc thang. Chú ý rằng phổ đầu ra bị méo không đáng kể do các đặc tính không lý tưởng của $H_0(f)$ và các phiên bản đã bị làm méo và suy giảm đó của $X(f)$ vẫn ở vị trí trung tâm trong các bội khác không của F_s. Các bản sao của các phổ đó có thể bị xóa bằng cách dùng một bộ lọc kháng ảnh. Thực chất bộ lọc kháng ảnh sẽ làm trơn tín hiệu ra của bộ lọc khôi phục bậc thang như minh họa trong hình 23.15.



HÌNH 23.15 (a) Khôi phục của tín hiệu tương tự, (b) hiệu quả của lọc kháng ảnh sau

Phân tích tần số của các tín hiệu thời gian rời rạc

Phân tích các tín hiệu thời gian rời rạc (DT) trong miền tần số tương tự như phân tích các tín hiệu thời gian liên tục (CT). Như trong phân tích tín hiệu CT, các phương pháp phân tích phụ thuộc vào kiểu tín hiệu. Phân tích tín hiệu DT không có chu kỳ sẽ được xem xét trước.

Biến đổi Fourier của các tín hiệu thời gian rời rạc

Việc phân tích một tín hiệu DT không có chu kỳ thành các thành phần tần số được tiến hành bằng cách dùng biến đổi Fourier rời rạc (DTFT). Do đó, DTFT của $x(n)$ sẽ được đưa ra bởi:

$$X(f) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fn} \tag{23.39}$$

BẢNG 23.6 Các tính chất của DTFT

Tính chất	Mô tả tín hiệu	Miền tần số
Đối xứng chẵn (tín hiệu thực)	$x_e(n) = \frac{1}{2}\{x(n) + x(-n)\}$	$X_e(f) = \sum_{n=-\infty}^{\infty} x_e(n)\cos(2\pi nf)$
Đối xứng lẻ (tín hiệu thực)	$x_o(n) = \frac{1}{2}\{x(n) - x(-n)\}$	$X_o(f) = -\sum_{n=-\infty}^{\infty} x_o(n)\sin(2\pi nf)$
Tuyến tính	$ax(n) + by(n)$	$aX(f) + bY(f)$
Dịch thời gian	$x(n-m)$	$e^{-j2\pi fm}X(f)$
Đảo thời gian	$x(-n)$	$X(-f)$
Tích chập	$x(n) * y(n)$	$X(f)Y(f)$
Tương quan	$R_{xy}(n) = x(n) \oplus y(n)$	$S_{xy}(f) = X(f)Y(-f)$
Quan hệ Wiener-Khinchine	$R_{xx}(n)$	$S_{xx}(f)$
Dịch tần số	$e^{j2\pi f_0 n}x(n)$	$X(f - f_0)$
Điều biến	$x(n)\cos(2\pi nf_0)$	$\frac{1}{2}\{X(f + f_0) + X(f - f_0)\}$
Nhân	$x(n)y(n)$	$\frac{1}{F_s}\int_{F_s} X(\lambda)Y(f - \lambda)d\lambda$
Vi phân trong miền tần số	$nx(n)$	$\frac{j}{2\pi}\frac{dX(f)}{df}$
Sai khác trong miền thời gian	$x(n) - x(n-1)$	$(1 - e^{-j2\pi f})X(f)$
Tổng	$\sum_{m=-\infty}^{n} x(m)$	$\frac{X(f)}{(1 - e^{-j2\pi f})} + \frac{F_s X(0)}{2}\sum_{m=-\infty}^{\infty}\delta(f - mF_s)$
Liên hợp phức	$x^*(n)$	$X^*(-f)$
Định lý Parseval	$\sum_{n=-\infty}^{\infty} x(n)y^*(n)$	$\frac{1}{F_s}\int_{F_s} X(f)Y^*(f)df$

Không giống biến đổi Fourier của các tín hiệu tương tự, $X(f)$ là tín hiệu tuần hoàn với chu kỳ F_s; vì vậy, dải tần số của một tín hiệu DT là duy nhất trong miền tần số $(-F_s/2, F_s/2)$ hoặc, tương đương $(0, F_s)$. Chú ý, công thức (23.39) là tổng tuyệt đối để $X(f)$ tồn tại, có nghĩa rằng:

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty$$

Nếu phổ của một tín hiệu tồn tại thì chúng ta có thể tìm thấy tín hiệu từ phổ của nó thông qua DTFT ngược. DTFT ngược của $X(f)$ được đưa ra như sau:

$$x(n) = \frac{1}{F}\int_{-F_s/2}^{F_s/2} X(f)e^{j2\pi fn}df$$

Các tính chất của DTFT của một tín hiệu DT được chỉ ra trong bảng 23.6.

BẢNG 23.7 DTFT của một số tín hiệu phổ biến

$x(n)$	Biểu diễn trong miền tần số, $X(f)$
$\delta(n)$	1
$A, -\infty < n < \infty$	$AF_s\sum_{k=-\infty}^{\infty}\delta(f-mF_s), \quad F_s = \frac{1}{T}$
$u(n)$	$\frac{1}{1-e^{-j2\pi f}} + \frac{F_s}{2}\sum_{k=-\infty}^{\infty}\delta(f-kF_s)$
$\Pi\left(\frac{n}{2q+1}\right)$	$\frac{\sin[(2q+1)\pi f]}{\sin(\pi f)}$
$\Lambda\left(\frac{n}{q}\right)$	$\frac{\sin^2(\pi fq)}{q\sin^2(\pi f)}$
$\mathrm{sgn}(n)$	$\frac{2}{1-e^{-j2\pi f}}$
$\alpha^n u(n)$	$\frac{1}{1-\alpha e^{-j2\pi f}}, \lvert\alpha\rvert < 1$
$\alpha^{\lvert n\rvert}$	$\frac{1-\alpha^2}{1-2\alpha\cos(2\pi f)+\alpha^2}, \lvert\alpha\rvert < 1$
$n\alpha^n u(n)$	$\frac{\alpha e^{-j2\pi f}}{\left(1-\alpha e^{-j2\pi f}\right)^2}, \lvert\alpha\rvert < 1$
$e^{-an}u(n)$	$\frac{1}{1-e^{-(a+j2\pi f)}}, \lvert\alpha\rvert > 0$
$e^{-\alpha\lvert n\rvert}$	$\frac{1-e^{-2\alpha}}{1-2e^{-\alpha}\cos(2\pi f)+e^{-2\alpha}}, \lvert\alpha\rvert > 0$
$e^{j2\pi f_0 n}$	$F_s\sum_{k=-\infty}^{\infty}\delta(f-f_0+kF_s)$
$\cos(2\pi f_0 n+\theta)$	$\frac{F_s}{2}\sum_{k=-\infty}^{\infty}e^{j\theta}\delta(f-f_0+kF_s)+e^{-j\theta}\delta(f+f_0+kF_s)$
$\frac{\sin(2\pi f_c n)}{\pi n}$	$\Pi\left(\frac{f}{2f_c}\right)$

Với các tín hiệu DT, tích chập của hai chuỗi $x(n)$ and $y(n)$ được biểu diễn như sau:

$$x(n) * y(n) = \sum_{m=-\infty}^{\infty} x(m)y(n-m) \tag{23.40}$$

trong khi hàm tương quan chéo của $x(n)$ và $y(n)$ được đưa ra bởi:

$$R_{xy}(n) = x(n) \oplus y(n) = \sum_{m=-\infty}^{\infty} x(m)y^*(m-n)$$

Khi $x(n)=y(n)$, thì hàm tự tương quan của $x(n)$ là:

$$R_{xx}(n) = \sum_{m=-\infty}^{\infty} x(m)x^*(m-n)$$

Ngoài phép phản xạ (cho phép tích chập), thì qui trình để tính tích chập và tương quan là như nhau. Vì vậy hiệu quả tính toán sẽ cao hơn khi dùng cùng một thuật toán để đánh giá cả hai hàm. Để đạt được điều này thì một trong các chuỗi được phản xạ (chỉ cho phân tích tương quan) và sau đó là phép tích chập:

$$R_{xy}(n) = x(n) * y^*(-n)$$
$$R_{xx}(n) = x(n) * x^*(-n)$$

Năng lượng của một tín hiệu không tuần hoàn được tính như sau:

$$E = \sum_{n=-\infty}^{\infty} \left|x(n)^2\right| = \sum_{n=-\infty}^{\infty} x(n)x^*(n)$$

Thay thế (23.40) vào công thức trên sẽ nhận được:

$$E = \frac{1}{F_s}\int_{F_s} |X(f)|^2 df = \frac{1}{F_s}\int_{F_s} S_{xx}(f)df \qquad (23.41)$$

Biểu thức này liên quan đến sự phân bố năng lượng của một tín hiệu không tuần hoàn theo tần số. Đại lượng $|X(f)|^2$ được gọi là mật độ phổ năng lượng của $x(n)$. DTFT của một số tín hiệu thường gặp được trình bày trong bảng 23.7.

Chuỗi Fourier rời rạc [6-8]

Giả sử $x(n)$ là tín hiệu DT tuần hoàn với chu kỳ N, có thể nhận được khai triển chuỗi Fourier rời rạc (DFS) tương tự như cách tính chuỗi Fourier phức cho các tín hiệu CT. Hàm trực giao cơ sở của DFS là $W_N^{-n} = e^{j2\pi n/N}$, vì vậy việc phân tích $x(n)$ thành tổng của N điều hòa dưới dạng hàm mũ phức được biểu diễn như sau:

$$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} = \sum_{k=0}^{N-1} c_k W_N^{-kn} \qquad (23.42)$$

trong đó c_k là các hệ số của chuỗi Fourier rời rạc. Nhân cả hai vế của (23.42) với W_N^{mn}, lấy tổng trên toàn chu kỳ và sử dụng cơ sở lập luận rằng:

$$\sum_{k=0}^{N-1} W_N^{n(k-m)} = \begin{cases} N, & k = m \\ 0, & k \neq m \end{cases}$$

sẽ dẫn đến:

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n)W_N^{kn}, \qquad k = 0, 1, 2, \ldots, N-1 \qquad (23.43)$$

Các hệ số DFS $\{c_k\}$ mô tả $x(n)$ trong miền tần số do c_k biểu diễn phổ biên độ và pha của thành phần điều hòa thứ k. Chú ý rằng $\{c_k\}$ là một chuỗi tuần hoàn với chu kỳ cơ bản là N, có nghĩa là $c_k = c_{k-N}$. Vì vậy, N mẫu liên tiếp của tín hiệu hoặc các hệ số DFS của nó mô tả đầy đủ tín hiệu ở miền thời gian hoặc tần số.

Như chỉ ra trong bảng 23.8, các tính chất của DFS là gần giống của DTFT. Một trong những lợi thế của DFS so với DTFT là việc thay thế tổng vô hạn trong DTFT bằng một tổng hữu hạn trong DFS vì vậy có thể tính DFS và nghịch đảo của DFS bằng máy tính số.

Xét một tín hiệu DT tuần hoàn với chu kỳ N, năng lượng trung bình của nó là:

$$P = \frac{1}{N}\sum_{n=0}^{N-1}|x(n)|^2 \tag{23.44}$$

Thay công thức (23.42) vào (23.44) sẽ có:

$$P = \frac{1}{N}\sum_{n=0}^{N-1}x(n)x^*(n) = \frac{1}{N}\sum_{n=0}^{N-1}x(n)\left(\sum_{n=0}^{N-1}c_k^* W_N^{kn}\right) = \sum_{k=0}^{N-1}|c_k|^2 \tag{23.45}$$

Đây chính là quan hệ Parseval cho các tín hiệu tuần hoàn DT, nó chỉ ra rằng năng lượng trung bình của x(n) là tổng năng lượng điều hòa. Do vậy chuỗi |ck|2 là mật độ phổ năng lượng mô tả sự phân bố của năng lượng như một hàm của tần số.

BẢNG 23.8 Các tính chất của chuỗi Fourier rời rạc

Tính chất	Mô tả tín hiệu	Các hệ số của chuỗi Fourier rời rạc
Tuyến tính	$\sum_{m=0}^{Q}\beta_m x_m(n)$	$\sum_{m=0}^{Q}\beta_m c_{m,k}$
Dịch thời gian	$x(n-m)$	$c_k W_N^{km}$
Đảo thời gian	$x(-n)$	c_{-k}
Điều biến	$x(n)W_N^{-mn}$	c_{k-m}
$x(n)$ thực	$x(n)$	$c_k = c_k^*$
$x_e(n)$ là tín hiệu DT chẵn	$x_e(n)$	$\mathrm{Re}(ck)$
$x_o(n)$ là tín hiệu DT lẻ	$x_0(n)$	$j\{\mathrm{Im}(c_k)\}$

Biến đổi Fourier rời rạc [6, 8, 13]

Biến đổi Fourier rời rạc (DFT) là một phương pháp quan trọng và vô cùng mạnh để phân tích các tín hiệu DT. Trái với DTFT, DFT có thể được áp dụng cho các chuỗi hữu hạn và sẽ đưa ra phổ rời rạc hữu hạn. Vì vậy phép biến đổi này được sử dụng cho tính toán số và nó thích hợp với phần cứng số. Kết quả là DFT trở thành một công cụ rất hữu ích để phân tích các dạng sóng hoặc dữ liệu khác nhau sinh ra theo nhiều nguyên tắc.

DFT là phép ánh xạ của chuỗi N *mẫu*, $x(n)$, thành chuỗi $X(k)$ khác trong miền tần số, đó là:

$$X(k) = \sum_{n=0}^{N-1}x(n)W_N^{kn}, \quad k = 0,1,....,N-1 \tag{23.46}$$

$X(k)$ được gọi là điều hòa thứ k và nó tồn tại với điều kiện tất cả các mẫu của $x(n)$ có giới hạn. Để khôi phục $x(n)$ từ $X(k)$ chúng ta cần thực hiện phép biến đổi ngược. Biến đổi Fourier rời rạc ngược (IDFT) của $X(k)$ được định nghĩa như sau:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn}, \qquad n = 0,1,...,N-1 \tag{23.47}$$

DFT và DFS là các khái niệm khác nhau, DFS chỉ có thể áp dụng với các tín hiệu tuần hoàn trong khi DFT lại giả thiết rằng tín hiệu là tuần hoàn. Hệ số tỷ lệ được áp dụng với phương trình tổng hợp của DFT và phương trình phân tích của phép phân tích DFS.

Các tính chất của DFT cũng tương tự như của DFS và DTFT đã được giới thiệu trong bảng 23.9.

Khi tính toán DFT cần chú ý: phép dịch vòng có thể được xem như việc chuyển một phần của chuỗi nằm ngoài khoảng từ 0 tới N - 1 tới phía trước chuỗi, tức là $x[(-n) \bmod N]$ tương đương với $x(N - n)$.

Từ kết quả của phép dịch vòng có thể thấy rằng tích chập tuyến tính nhận được từ công thức (23.40) khác tích chập vòng đã được đưa ra trong bảng 23.9, vì vậy:

$$x(n) \otimes_N y(n) = \sum_{m=0}^{N-1} x(m) y[(n-m) \bmod N] = \sum_{m=0}^{N-1} x[(n-m) \bmod N] y(m)$$

ở đây $x[(n - m) \bmod N]$ là phiên bản phản xạ và dịch vòng của $x(n)$. Tuy nhiên, bằng cách lựa chọn thích hợp giá trị của N, tích chập vòng và tuyến tính có thể là như nhau. Vì vậy, nếu các tín hiệu $x(n)$ và $y(n)$ có chiều dài tương ứng là N_1 và N_2 thì tích chập vòng và tuyến tính là như nhau miễn là $N \geq N_1 + N_2 - 1$.

BẢNG 23.9 Các tính chất của DFT

Tính chất	Mô tả tín hiệu	Biến đổi Furier rời rạc
Tuyến tính	$\sum_{m=0}^{Q} a_m x_m(n)$	$\sum_{m=0}^{Q} a_m X_m(k)$
Dịch vòng	$x[(n-m) \bmod N]$	$W_N^{km} X(k)$
Điều biến	$W_N^{-qn} x(n)$	$X[(k-q) \bmod N]$
Đảo thời gian	$x^*[-n \bmod N]$	$x^*(k)$
Liên hợp phức	$x^*(n)$	$X^*[-k \bmod N]$
Tích chập vòng	$\sum_{m=0}^{N-1} x(m) y[(n-m) \bmod N]$	$X(k)Y(k)$
Nhân	$x(n)y(n)$	$\frac{1}{N}\sum_{m=0}^{N-1} X(m)Y[(k-m) \bmod N]$
Định lý Parseval	$\sum_{k=0}^{N-1} x(n)y^*(n)$	$\frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$
Phần thực của tín hiệu	$\mathrm{Re}\{x(n)\}$	$\frac{1}{2}\{X(k) + X^*(N-k)\}$
Phần ảo của tín hiệu	$j\,\mathrm{Im}\{x(n)\}$	$\frac{1}{2}\{X(k) - X^*(N-k)\}$
Chẵn phức	$x_{ce}(n) = \frac{1}{2}\{x(n) + x^*(N-n)\}$	$X_R(k)$
Lẻ phức	$x_{co}(n) = \frac{1}{2}\{x(n) - x^*(N-n)\}$	$jX_I(k)$
Tín hiệu thực bất kỳ	$x(n)$	$X(k) = X^*(N-k)$ $X_R(k) = X_R(N-k)$ $X_I(k) = -X_I(N-k)$ $\lvert X(k)\rvert = \lvert X(N-k)\rvert$ $\angle X(k) = -\angle X(N-k)$

Tương quan vòng có thể được thực hiện bởi tích chập vòng vì tương quan chéo của hai chuỗi $x(n)$ và $y(n)$ có thể được biểu diễn như sau:

$$R_{xy}(n) = x(n) \otimes_N y(n) = x(n) \otimes_N y^*[(-n) \bmod N]$$

Từ phương trình (23.46) và (23.47), thấy rằng để tính các hệ số DFT cần N_2 phép nhân phức và $N(N-1)$ phép cộng phức. Điều này có thể là gánh nặng cho máy tính nếu N rất lớn. Các biến đổi Fourier nhanh là kiểu thuật toán khác được phát triển để tăng tốc độ tính toán các hệ số DFT. Người đọc có thể tham khảo các tài liệu giới thiệu các dạng khác nhau của thuật toán FFT.[7,8] Có thể chỉ ra rằng số phép tính yêu cầu đối với thuật toán FFT là hằng số ($N*\log_2 N$). Do vậy, khối lượng tính đã giảm khi thuật toán FFT được sử dụng để tính DFT.

Nhận xét về việc xử lý DFT của các tín hiệu

Bù không (Zero Padding)

Dùng FFT để tính các hệ số DFT có một số ràng buộc với giá trị N, ví dụ N có thể là một lũy thừa của 2 với thuật toán FFT cơ số 2. Tích chập vòng vẫn là một giải pháp không mong muốn vì nó biểu diễn bằng tích biến đổi của hai chuỗi. Bù không là một kỹ thuật để khắc phục tình huống trên, bù không được dùng hoặc để làm tăng độ dài chuỗi để thuật toán FFT cơ số 2 có thể được sử dụng hoặc để chắc chắn rằng cả hai tích chập tuyến tính và vòng là như nhau. Ngoài ra, thủ tục này cũng được dùng để đưa ra cách biểu diễn phổ tín hiệu tốt hơn vì độ phân giải tần số của các mẫu trích FFT sẽ giảm khi N tăng.

Nguyên nhân của sai số

Phổ tổng hợp của tín hiệu đã được lấy mẫu bao gồm phổ tương tự được lặp lại ở các bội nguyên của tần số lấy mẫu. Sự chồng chéo phổ của tín hiệu tương tự với các phiên bản đã được dịch đi của nó dẫn đến méo tín hiệu. Trong thực tế, các sai số quá lớn do méo tín hiệu đã được hạn chế tối thiểu bằng cách tăng tốc độ lấy mẫu hoặc lọc trước tín hiệu để bỏ đi các thành phần phổ tần số cao. Nguyên nhân khác của sai số trong quá trình xử lý DFT các tín hiệu là sự *để hở phổ*. Đó là do dùng một cửa sổ cắt một tín hiệu dài vô hạn để thu được một dữ liệu có độ dài hữu hạn cho việc xử lý DFT. Việc phân tích tín hiệu với hàm cửa sổ trong miền thời gian được thay bằng phép nhân chập phổ của tín hiệu đã được lấy mẫu với phổ của cửa sổ trong miền tần số. Giả sử độ rộng cửa sổ không tương ứng với bội nguyên lần các chu kỳ của tất cả các thành phần tần số của một tín hiệu thời gian rời rạc, thì một thành phần tần số đơn sẽ rơi vào vùng tần số khác trong DFT của dữ liệu đã bị cắt. Hiện tượng này dẫn đến méo phổ và khó xác định được có hay không hai tần số liền kề nhau trong cùng một tín hiệu. Độ phân giải của phổ sẽ tốt hơn nếu độ rộng cửa sổ tăng hoặc bằng cách chọn một hàm cửa sổ với các *vấu* ở hai bên là thấp. *Hiệu ứng hàng rào* sẽ xuất hiện do chỉ một số hữu hạn điểm tần số của một phổ tần số liên tục tạo ra bởi DFT. Vì vậy có thể đỉnh của một thành phần tần số nào đó trong một tín hiệu bị mất do nó bị đặt giữa hai điểm tần số liền kề nhau trong phổ. Vấn đề này có thể được giải quyết bằng cách tăng số các điểm của DFT, N, trong khi duy trì cùng tốc độ lấy mẫu, dẫn đến có nhiều mẫu hơn trong DFT hoặc chọn phương pháp bù không.

Lựa chọn tham số DFT

Chu kỳ lấy mẫu T_s, độ phân giải tần số Δf và độ dài N là ba tham số phải được xác định khi thực hiện xử lý tín hiệu DFT. Vì $F_s = N\Delta f = 1/T_s$ nên các tham số này liên quan với nhau theo công thức $\Delta f = 1/NT_s$. Nếu ký hiệu T_x là độ dài của tín hiệu được lấy mẫu thì $T_x = MT_s$. Vì vậy, sẽ không có méo phổ nếu $N \geq M = T_x/T_s$. Tương tự sẽ không có méo phổ nếu:

$$\Delta f = \frac{1}{T} = \frac{1}{NT_s} \leq \frac{1}{MT_s} = \frac{1}{T_x}$$

Các tham số để xử lý DFT một tín hiệu liên tục đã được lấy mẫu cần phải được lựa chọn cẩn thận để tránh méo phổ do méo tín hiệu hoặc do việc cắt dữ liệu. Giả sử một cửa sổ với độ rộng T_x giây và rằng tín hiệu có độ rộng dải tần số cực đại là B Hz thì dựa trên lý thuyết lấy mẫu chúng ta sẽ có méo tín hiệu không đáng kể, miễn là $F_s = 1/T_s \geq 2B$. Để tránh sự hở phổ do việc cắt đột ngột dữ liệu, độ phân giải tần số được chọn phải thỏa mãn $1/\Delta f = T \geq T_x$. Do vậy méo phổ do méo tín hiệu và hở phổ có thể tránh được nếu độ dài của DFT được chọn thỏa mãn $N = F_s/\Delta f \geq 2BT_x$.

Tài liệu tham khảo

[1] Soliman, S.S., and Srinath, M.D., *Continuous and Discrete Signals and Systems*, Prentice-Hall, 1998.

[2] O'Flynn, M., and Moriarty, E., *Linear Systems: Time Domain and Transform Analysis*, Wiley, 1987.

[3] Lathi, B.P., *Modern Digital and Analog Communication Systems*, Oxford University Press, Third Edition, 1998.

[4] Proakis, J.G., and Salehi, M., *Communication Systems Engineering*, Prentice-Hall, 1994.

[5] Taylor, F.J., *Principles of Signals and Systems*, McGraw-Hill, 1994.

[6] Carlson, G.E., *Signal and Linear System Analysis*, Wiley, Second Edition, 1998.

[7] Proakis, J.G., and Manolakis, D.G., *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice-Hall, 1996.

[8] Oppenheim, A.V., and Schafer, R.W., with Buck, J.R., *Discrete-Time Signal Processing*, Prentice-Hall, Second Edition, 1999.

[9] Houts, R.C., *Signal Analysis in Linear Systems*, Saunders College Publishing, 1991.

[10] Ziemer, R.E., Tranter, W.H., and Fannin, D.R., Signals and Systems: *Continuous and Discrete*, MacMillan Publishing Company, Third Edition, 1993.

[11] Orfanidis, S.J., *Introduction to Signal Processing*, Prentice-Hall, 1996.

[12] Haykin, S., and Veen, B.V., *Signals and Systems*, Wiley, 1999.

[13] Taylor, F., and Mellot, J., *Hands-On Digital Signal Processing*, McGraw-Hill, 1998.

23.2 Biến đổi z và các hệ thống số

Rolf Johansson

Một hệ thống số (hoặc hệ thời gian rời rạc hoặc hệ dữ liệu trích mẫu) là một thiết bị ví dụ như bộ điều khiển số hoặc bộ lọc số, hoặc tổng quát là một hệ thống để bổ sung cho máy tính số tính số, thường có một số tương tác tuần hoàn với môi trường xung quanh và được hỗ trợ một phương pháp luận để phân tích và thiết kế. Các thuật toán lặp là đặc biệt quan trọng khi mô hình hóa và phân tích, ví dụ như các phương trình sai phân của dữ liệu vào/ra và biến đổi z là công cụ hữu ích để giải quyết những vấn đề này.

Biến đổi z được dùng để phân tích các hệ thời gian tuyến tính bất biến và các tín hiệu thời gian rời rạc (ví dụ cho bộ điều khiển hoặc lọc số) và có thể được so sánh với biến đổi Laplace dùng để phân tích các hệ thống và các tín hiệu thời gian liên tục. Tích chập của hai tín hiệu thời gian là một thuộc tính có ích tương đương với việc nhân các biến đổi z tương ứng của chúng. Biến đổi z là một công cụ mô tả một hệ thời gian tuyến tính bất biến theo các vị trí cực - không, hàm truyền, sơ đồ Bode và đáp ứng của nó đối với nhiều dạng tín hiệu khác nhau. Ngoài ra, nó đưa ra các mối quan hệ quan trọng giữa các thuộc tính thời gian và phổ của tín hiệu. Biến đổi z thường xuất hiện khi phân tích các phương trình sai phân được dùng nhiều trong kỹ thuật và toán ứng dụng.

Biến đổi z

Biến đổi z của chuỗi $\{x_k\}_{-\infty}^{+\infty}$ được định nghĩa như hàm tạo sau:

$$X(z) = \mathfrak{Z}\{x\} = \sum_{k=-\infty}^{\infty} x_k z^{-k} \tag{23.48}$$

trong đó biến z thể hiện bản chất của phép dịch về trước, sao cho:

$$\mathfrak{Z}\{x_{k-1}\} = z\mathfrak{Z}\{x_k\} = zX(z) \tag{23.49}$$

Biến đổi z là chuỗi lũy thừa vô hạn theo biến phức z^{-1} trong đó $\{x_k\}$ là các hệ số của chuỗi. Bởi vì biến đổi z là chuỗi lũy thừa vô hạn nên nó chỉ tồn tại với các giá trị của z để chuỗi này hội tụ và *miền hội tụ* của $X(z)$ là một tập z sao cho $X(z)$ nhận một giá trị hữu hạn. Điều kiện đủ để tồn tại biến đổi z là sự hội tụ của chuỗi lũy thừa:

$$\sum_{k=-\infty}^{\infty} |x_k| \cdot |z^{-k}| < \infty \tag{23.50}$$

BẢNG 23.10 Các tính chất của biến đổi z

Biến đổi z	$\mathfrak{Z}(\{f_k\}) = F(z)$
Tích chập	$\mathfrak{Z}(\{f_k * g_k\}) = \mathfrak{Z}(\{f_k\}) \cdot \mathfrak{Z}(\{g_k\})$
	$\mathfrak{Z}(\{f_k \cdot g_k\}) = \mathfrak{Z}(\{f_k\}) * \mathfrak{Z}(\{g_k\})$
Dịch trước	$\mathfrak{Z}(\{f_{k+1}\}) = z\mathfrak{Z}(\{f_k\}) = zF(z)$
Dịch sau	$\mathfrak{Z}(\{f_{k-1}\}) = \mathfrak{Z}(\{f_k\}) = z^{-1}F(z)$
Tuyến tính	$\mathfrak{Z}(\{af_k + bg_k\}) = a\mathfrak{Z}(\{f_k\}) + b\mathfrak{Z}(\{g_k\})$
Nhân	$\mathfrak{Z}(\{a^k f_k\}) = F(a^{-1}z)$
Giá trị cuối	$\lim_{k\to\infty} f_k = \lim_{k\to 1}(1 - z^{-1})F(z)$
Giá trị đầu	$f_0 = \lim_{z\to\infty} F(z)$

	Miền thời gian		Biến đổi z
Xung	$\delta_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$	$\rightleftharpoons$	$\mathfrak{Z}\{\delta_k\} = 1, \quad z \in C$
Hàm bậc thang	$\sigma_k = \begin{cases} 0, & k < 0 \\ 1, & k \geq 0 \end{cases}$	$\rightleftharpoons$	$\mathfrak{Z}(\sigma_k) = \dfrac{z}{z-1}, \quad \lvert z\rvert > 1$
Hàm dốc	$x_k = k \cdot \sigma_k$	$\rightleftharpoons$	$X(z) = \dfrac{z}{(z-1)^2}, \quad \lvert z\rvert > 1$
Hàm mũ	$x_k = a^k \cdot \sigma_k$	$\rightleftharpoons$	$X(z) = \dfrac{z}{z-a}, \quad \lvert z\rvert > \lvert a\rvert$
Hàm sin	$x_k = \sin \omega k \cdot \sigma_k$	$\rightleftharpoons$	$X(z) = \dfrac{z \sin \omega}{z^2 - 2z\cos\omega + 1}, \quad \lvert z\rvert > 1$

Miền hội tụ cho một tín hiệu hữu hạn là toàn bộ mặt phẳng z trừ $z = 0$ và $z = \infty$. Cho một chuỗi vô hạn một phía $\{x_k\}_{k=0}^{\infty}$, có thể tìm được một số r sao cho chuỗi lũy thừa hội tụ với $|z| > r$. Khi đó, *biến đổi ngược* có thể nhận được như sau:

$$x_k = \frac{1}{2\pi i}\oint X(z)z^{k-1}dz \tag{23.51}$$

Trong đó đường cong của tích phân sẽ bao tất cả các điểm kỳ dị của $X(z)$. Trong thực tế, đó là một quá trình chuẩn để sử dụng các kết quả dạng bảng. Một số cặp biến đổi z chuẩn được trình bày trong bảng 23.10.

Các hệ thống số và dữ liệu rời rạc

Việc lấy mẫu tuần hoàn các tín hiệu và tính toán tiếp theo hoặc việc lưu giữ các kết quả yêu cầu máy tính sắp sếp kế hoạch cho việc lấy mẫu và xử lý các chuỗi kết quả. $x(t)$ có thể đo được nếu chu kỳ quan sát bằng thời gian lấy mẫu T (chu kỳ lấy mẫu). Chuỗi mẫu có thể được biểu diễn như sau:

$$\{x_k\}_{-\infty}^{\infty}, \qquad x_k = x(kT) \qquad k = ..., -1, 0, 1, 2, ... \tag{23.52}$$



HÌNH 23.16 Tín hiệu thời gian liên tục $x(t)$ và bộ lấy mẫu tạo ra chuỗi mẫu $\{x_k\}$

và điều quan trọng là phải biết chắc rằng chuỗi mẫu biểu diễn đầy đủ biến gốc $x(t)$, xem hình 23.16. Để việc lấy mẫu là lý tưởng, yêu cầu khoảng cách lấy mẫu phải rất ngắn và hàm lấy mẫu phải được biểu diễn bằng một chuỗi các xung ngắn vô hạn $\delta(t)$ (xung Dirac). Vì vậy, hàm thời gian lấy mẫu được biểu diễn như sau:

$$x_\Delta(t) = x(t)\cdot T\sum_{k=-\infty}^{\infty}\delta(t-kT) = x(t)\cdot \text{Ш}_T(t) \tag{23.53}$$

$$\text{Ш}_T(t) \triangleq T\sum_{k=-\infty}^{\infty}\delta(t-kT) \tag{23.54}$$

và trong đó chu kỳ lấy mẫu T được nhân lên để chắc chắn rằng trung bình trên một chu kỳ lấy mẫu của biến gốc x và tín hiệu được lấy mẫu x_Δ tương ứng có cùng độ lớn. Một ứng dụng trực tiếp của biến rời rạc $x_\Delta(t)$ theo công thức (23.53) chứng tỏ rằng phổ của x_Δ liên quan tới biến đổi z, $X(z)$ như sau:

$$X_\Delta(i\omega) = \mathcal{F}\{x(t)\cdot \text{Ш}_T(t) = T\sum_{k=-\infty}^{\infty}x_k\exp(-i\omega kT) = TX(e^{i\omega T}) \tag{23.55}$$

Rõ ràng rằng, biến gốc $x(t)$ và dữ liệu lấy mẫu là không đồng nhất và vì vậy cần phải xem xét các hiệu ứng méo tín hiệu do quá trình rời rạc hóa gây ra. Xét phổ của tín hiệu lấy mẫu $x_\Delta(t)$ nhận được bởi biến đổi Fourier:

$$X_\Delta(i\omega) = \mathcal{F}\{x_\Delta(t)\} = \mathcal{F}\{x(t)\} * \mathcal{F}\{\text{Ш}_T(t)\} \tag{23.56}$$

trong đó:

$$\mathcal{F}\{\text{Ш}_T(t)\} = \sum_{k=-\infty}^{\infty}\delta\left(\omega - \frac{2\pi}{T}k\right) = \frac{T}{2\pi}\text{Ш}_{2\pi/T}(\omega) \tag{23.57}$$

sao cho:

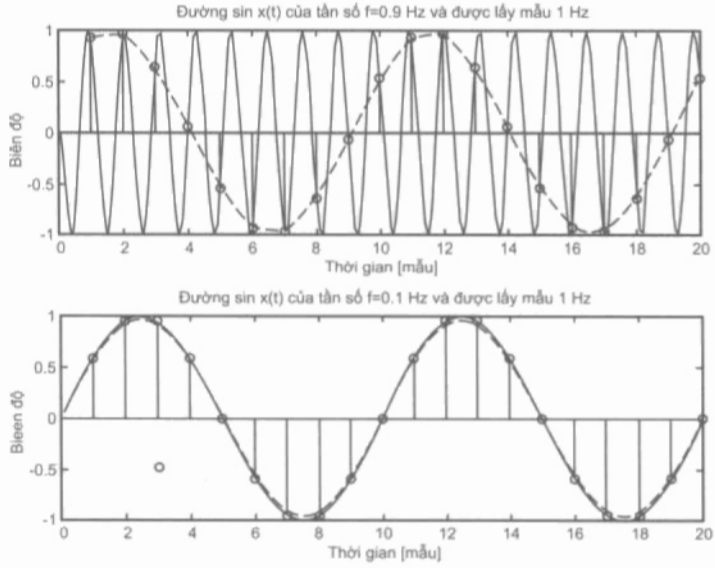
$$X_\Delta(i\omega) = \mathcal{F}\{x(t)\} * \mathcal{F}\{\text{Ш}_T(t)\} = \sum_{k=-\infty}^{\infty}X\left[i\left(\omega - \frac{2\pi}{T}k\right)\right] \tag{23.58}$$

Do đó, biến đổi Fourier X_Δ của biến được lấy mẫu là khai triển tuần hoàn của phổ gốc $X(i\omega)$ theo trục tần số với một chu kỳ bằng tần số lấy mẫu $\omega_s = 2\pi/T$. Một kết quả quan trọng dựa trên

những quan sát này là *thuyết lấy mẫu Shannon* phát biểu rằng biến thời gian liên tục $x(t)$ có thể được khôi phục lại từ các mẫu nếu và chỉ nếu tần số lấy mẫu ít nhất bằng hai lần tần số cao nhất để $X(i\omega)$ khác không. Vì vậy biến gốc $x(t)$ có thể được khôi phục là:

$$x(t) = \sum_{k=-\infty}^{\infty} x_k \frac{\sin\frac{\pi}{T}(t-kT)}{\frac{\pi}{T}(t-kT)} \tag{23.59}$$

Công thức (23.59) được gọi là *phép nội suy Shannon*, thường được trích dẫn mặc dù nó chỉ đúng với các chuỗi dữ liệu dài vô hạn và yêu cầu một bộ lọc không nhận quả để khôi phục tín hiệu thời gian liên tục $x(t)$ theo phép tính thời gian thực. Tần số $\omega_n=\omega_s/2=\pi/T$ được gọi là *tần số Nyquist* và chỉ ra giới hạn trên của việc lấy mẫu không làm méo tín hiệu. Một phổ khác không nằm ngoài giới hạn này sẽ dẫn đến sự giao thoa giữa tần số lấy mẫu và tín hiệu được lấy mẫu, xem hình 23.17.



HÌNH 23.17 Méo tín hiệu xuất hiện khi lấy mẫu tín hiệu sin $x(t)=\sin 2\pi.0.9t$ ở tần số lấy mẫu thiếu, 1 Hz (chu kỳ lấy mẫu $T = 1$) (đồ thị trên). Tín hiệu được lấy mẫu bị méo với các thành phần chính của nó giống như tín hiệu $x(t)=\sin 2\pi.0.1t$ được lấy mẫu với cùng tốc độ (đồ thị dưới)

Biến đổi Fourier rời rạc

Xét một chuỗi dài hữu hạn $\{x_k\}_{k=0}^{N-1}$ bằng 0 bên ngoài khoảng $0 \le k \le N - 1$. Ước lượng biến đổi z $X(z)$ tại N điểm cách đều nhau trên đường tròn đơn vị $z = \exp(i\omega_k T) = \exp\left[i(2\pi / NT)kT\right]$ với $k = 0, 1,\ldots, N - 1$ sẽ xác định *Biến đổi Fourier rời rạc* (DFT) của tín hiệu x với chu kỳ lấy mẫu h và N phép đo:

$$X_k = DFT\{x(kT)\} = \sum_{l=0}^{N-1} x_l \exp(-i\omega_k lT) = X(e^{i\omega_k T}) \tag{23.60}$$

Chú ý rằng biến đổi fourier rời rạc chỉ được xác định ở các điểm tần số rời rạc:

$$\omega_k = \frac{2\pi}{NT}k, \qquad k = 0,1,\ldots,N-1 \tag{23.61}$$

trong thực tế, biến đổi Fourier rời rạc thay đổi biến đổi Fourier và biến đổi z theo các yêu cầu thực tế của các phép đo hữu hạn. Các tính chất tương tự đều được giữ nguyên đối với biến đổi Laplace rời rạc với $z = \exp(sT)$, trong đó s là biến của biến đổi Laplace.

Hàm truyền

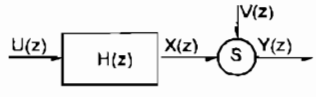
Xét hệ thời gian tuyến tính rời rạc với chuỗi vào $\{u_k\}$ (chuỗi kích thích) và chuỗi ra $\{y_k\}$ (chuỗi đáp ứng). Sự phụ thuộc của của tín hiệu đầu ra của một hệ tuyến tính được đặc trưng bởi phương trình tích chập và biến đổi z của nó:

$$\begin{aligned} y_k &= \sum_{m=0}^{\infty} h_m u_{k-m} + v_k = \sum_{m=-\infty}^{k} h_{k-m} u_m + v_k, \qquad k = \ldots,-1,0,1,2,\ldots \\ Y(z) &= H(z)U(z) + V(z) \end{aligned} \tag{23.62}$$

trong đó chuỗi $\{v_k\}$ tương ứng với một số đầu vào của sai số và nhiễu ngoài. $Y(z) = \mathcal{Z}\{y\}, U(z) = \mathcal{Z}\{u\}, V(z) = \mathcal{Z}\{v\}$ là tín hiệu đầu ra và các tín hiệu đầu vào. *Hàm trọng lượng* $h(kT) = \{h_k\}_{k=0}^{\infty}$ bằng 0 với k âm và theo thuyết nhân quả được gọi là *đáp ứng xung* của hệ thống số (so sánh với đáp ứng xung của hệ thời gian liên tục). Đáp ứng xung và biến đổi z của nó, *hàm truyền xung*:

$$H(z) = \mathcal{Z}\{h(kT)\} = \sum_{k=0}^{\infty} h_k z^{-k} \tag{23.63}$$

xác định đáp ứng của hệ đối với đầu vào $U(z)$, xem hình 23.18.



HÌNH 23.18 Sơ đồ khối có quan hệ hàm truyền được giả thiết là $H(z)$ giữa đầu vào $U(z)$, nhiễu $V(z)$, hàm trung gian $X(z)$ và đầu ra $Y(z)$

Hàm truyền xung $H(z)$ được xác định bởi tỉ số:

$$H(z) = \frac{X(z)}{U(z)} \tag{23.64}$$

và quy định quan hệ giữa đầu vào và đầu ra ở miền tần số của hệ thống. Đặc biệt sơ đồ Bode được đánh giá như $|H(z)|$ và arg $H(z)$ cho $z=\exp(i\varpi_k T)$ và với $|\omega_k| < \omega_n = \pi / T$, tức là khi $H(z)$ được đánh giá tại các điểm tần số Nyquist ϖ_n quanh đường tròn đơn vị.

Các hệ không gian - trạng thái

Thay thế cho cách biểu diễn quan hệ vào - ra bằng các hàm truyền đó là biểu diễn không gian trạng thái. Xét phương trình không gian trạng thái rời rạc với một vector trạng thái $x_k \in \mathbb{R}^n$, đầu vào $u_k \in \mathbb{R}^n$ và các biến quan sát $y_k \in \mathbb{R}^m$:

$$\begin{cases} x_{k+1} = \Phi x_k + \Gamma u_k \\ y_k = C x_k + D u_k \end{cases} \qquad k = 0,1\ldots \tag{23.65}$$

với hàm truyền xung:

$$H(z) = C(zI - \Phi)^{-1}\Gamma + D \tag{23.66}$$

và biến đầu ra:

$$Y(z) = C\sum_{k=0}^{\infty} \Phi^k z^{-k} x_0 + H(z)U(z) \tag{23.67}$$

trong đó các tác động có thể có của điều kiện đầu x_0 sẽ xuất hiện như số hạng đầu tiên. Chú ý rằng các điều kiện đầu x_0 có là các tác động của đầu vào trong khoảng thời gian $(-\infty, 0)$.

Các hệ thống số mô tả bằng các phương trình sai phân (mô hình ARMAX)

Một lớp quan trọng của các quá trình ngẫu nhiên không dừng là lớp có một số đáp ứng xác định đối với một đầu vào và một quá trình ngẫu nhiên dừng có tính chất xếp chồng. Điều này liên quan tới việc, ví dụ, đầu vào không thể được mô tả một cách hiệu quả bởi một vài phân bố xác suất. Một mô hình rời rạc có thể được biểu diễn theo dạng phương trình sai phân với tín hiệu vào $\{u_k\}$ thường được xem như đã biết:

$$y_k = -a_1 y_{k-1} - \ldots - a_n y_{k-n} + b_1 u_{k-1} + \ldots + b_n u_{k-n} + w_k + c_1 w_{k-1} + \cdots + c_n w_{k-n} \tag{23.68}$$

Biến đổi z của phương trình (23.68) là:

$$A(z^{-1})Y(z) = B(z^{-1})U(z) + C(z^{-1})W(z) \tag{23.69}$$

trong đó:

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + \cdots + a_n z^{-n} \\ B(z^{-1}) &= 1 + b_1 z^{-1} + \cdots + b_n z^{-n} \\ C(z^{-1}) &= 1 + c_1 z^{-1} + \cdots + c_n z^{-n} \end{aligned} \tag{23.70}$$

Các mô hình ngẫu nhiên bao gồm đa thức A, theo phương trình (23.69) và (23.70) được biết đến như các mô hình **AR (Autoregressive)** và các mô hình bao gồm đa thức C được biết đến như các mô hình **MA (Moving-Average)**, trong khi, đa thức B xác định các tác động của đầu vào từ ngoài (X). Một cách mô tả khác là *đáp ứng xung hữu hạn (finite impulse response)* hoặc *bộ lọc tất cả các điểm không (all-zero filter)*.

Vì vậy mô hình đầy đủ của phương trình (23.69) là một mô hình **ARMAX** (***Autoregressive Moving Average)*** với tín hiệu đầu vào và hàm truyền xung của nó $H(z)=B(z^{-1})/A(z^{-1})$ là ổn định nếu và chỉ nếu các điểm cực (tức là các số phức $z_1,\ldots, z_n$ là nghiệm của phương trình $A(z^{-1}) = 0$) hoàn toàn nằm trong đường tròn đơn vị (tức là $|z_i| < 1$). Các điểm không của hệ thống (tức là các số phức $z_1,\ldots, z_n$ là nghiệm của phương trình $B(z^{-1}) = 0$) có thể nhận giá trị bất kỳ mà không xuất hiện bất kỳ sự mất ổn định nào, mặc dù vậy các điểm không nên nằm bên trong đường tròn đơn vị, tức là $|z_i| < 1$. Vì $\{y_k\}$ tuyến tính nên có thể tách $\{y_k\}$ thành một quá trình xác định hoàn toàn $\{x_k\}$ và một quá trình ngẫu nhiên hoàn toàn $\{v_k\}$:

$$\begin{cases} A(z^{-1})X(z) = B(z^{-1})U(z) \\ A(z^{-1})V(z) = C(z^{-1})W(z) \end{cases} \qquad \begin{cases} y_k = x_k + v_k \\ Y(z) = X(z) + V(z) \end{cases} \tag{23.71}$$

Kiểu phân tách của phương trình (23.71) thành các quá trình xác định và các quá trình ngẫu nhiên được gọi là phân tách Wold (***Wold decomposition***).

Dự báo và khôi phục

Xét các vấn đề dự báo trước d bước của tín hiệu đầu ra khi tín hiệu đầu ra $\{y_k\}$ được tạo bởi mô hình ARMA:

$$A(z^{-1})Y(z) = C(z^{-1})W(z) \tag{3.72}$$

Mô hình này được điều khiển bởi ồn trắng trung bình không (zero-mean white noise) $\{w_k\}$ với hiệp phương sai $\mathcal{E}\{w_i w_j\} = \sigma_w^2 \delta_{ij}$. Nói cách khác, giả thiết rằng đã có các quan sát $\{y_k\}$ cho tới thời điểm hiện tại, phải dự báo trước một cách tối ưu d bước đầu ra như thế nào?. Khai triển đa thức C theo phương trình *Diophantin* có dạng như sau:

$$C(z^{-1}) = A(z^{-1})F(z^{-1}) + z^{-d}G(z^{-1}) \tag{23.73}$$

trong đó:

$$\begin{aligned} F(z^{-1}) &= 1 + f_1 z^{-1} + \cdots + f_{n_F} z^{-n_F}, \quad n_F = d-1 \\ G(z^{-1}) &= g_0 + g_1 z^{-1} + \cdots + g_{n_G} z^{-n_G}, \quad n_G = \max(n_A - 1, n_C - d) \end{aligned} \tag{23.74}$$

Biểu diễn của z^{-1} như là *toán tử dịch lùi (backward shift operator)*, áp dụng các phương trình (23.72) và (23.73) sẽ cho công thức sau:

$$y_{k+d} = F(z^{-1})w_{k+d} + \frac{G(z^{-1})}{C(z^{-1})} y_k \tag{23.75}$$

Ký hiệu các bộ dự báo tuyến tính d-bước là y_{k+d} dựa trên thông tin thu được tại thời điểm k. Vì số hạng trung bình không $F(z^{-1})w_{k+d}$ của phương trình (23.75) là không thể dự báo được tại thời điểm k, nên có thể đề xuất bộ dự báo d-bước như sau:

$$\hat{y}_{k+d|k} = \frac{G(z^{-1})}{C(z^{-1})} y_k \tag{23.76}$$

Sai số dự báo thỏa mãn:

$$\begin{aligned} \varepsilon_{k+d} &= (\hat{y}_{k+d|k} - y_{k+d}) \\ &= \frac{G(z^{-1})}{C(z^{-1})} y_k - \frac{A(z^{-1})F(z^{-1}) + z^{-d}G(z^{-1})}{C(z^{-1})} y_{k+d} \\ &= -F(z^{-1})w_{k+d} \end{aligned} \tag{23.77}$$

Ký hiệu $\mathcal{E}\{\cdot|\Psi_k\}$ là *kỳ vọng toán học có điều kiện* đối với thông tin đo được ở thời điểm k. Kỳ vọng toán học và hiệp phương sai của dự báo d-bước liên quan đến thông tin thu được ở thời điểm k là:

$$\begin{aligned} \mathcal{E}\{\hat{y}_{k+d|k} - y_{k+d} \,|\, \Psi_k\} &= \mathcal{E}\{-F(z^{-1})w_{k+d} \,|\, \Psi_k\} = 0 \\ \mathcal{E}\{(\hat{y}_{k+d|k} - y_{k+d})^2 \,|\, \Psi_k\} &= \mathcal{E}\{[F(z^{-1})w_{k+d}]^2 \,|\, \Psi_k\} \\ &= \mathcal{E}(w_{k+d} + f_1 w_{k+d-1} + \cdots + f_{d-1} w_{k+1})^2 \,|\, \Psi_k\} \\ &= (1 + f_1^2 + \cdots + f_{n_F}^2)\sigma_w^2 = 0 \end{aligned} \tag{23.78}$$

Suy ra bộ dự báo của phương trình (23.76) là không sai lệch và sai số dự báo chỉ phụ thuộc vào tương lai (các thành phần nhiễu không thể dự báo). Dễ dàng chỉ ra rằng bộ dự báo của phương trình (23.76) sẽ đạt được giới hạn dưới của phương trình (23.78) và bộ dự báo của phương trình (23.76) tối ưu theo nghĩa phương sai sai số dự báo là tối thiểu.

Ví dụ 23.1 - Bộ dự báo tối ưu cho mô hình bậc nhất

Xét mô hình ARMA bậc nhất:

$$y_{k+1} = -a_1 y_k + w_{k+1} + c_1 w_k \tag{23.79}$$

Phương sai của bộ dự báo trước một bước là:

$$\begin{aligned}\mathcal{E}\{(\hat{y}_{k+1|k}-y_{k+1})^2\Big|\mathfrak{F}_k\} &= \mathcal{E}\{(\hat{y}_{k+1|k}+a_1y_k-c_1w_k)^2\Big|\mathfrak{F}_k\}+\mathcal{E}\{w_{k+1}^2\Big|\mathfrak{F}_k\}\\ &= \mathcal{E}\{(\hat{y}_{k+1|k}+a_1y_k-c_1w_k)^2\Big|\mathfrak{F}_k\}+\sigma_w^2\ge\sigma_w^2\end{aligned} \tag{23.80}$$

Bộ dự báo tối ưu thỏa mãn giới hạn dưới của phương trình (23.80) nhận được từ phương trình (23.80) là:

$$\hat{y}^o_{k+1|k} = -a_1y_k + c_1w_k \tag{23.81}$$

nhưng nó không thể thực hiện khi nó ở trạng thái dừng bởi vì w_k không thể đo được. Vì vậy, chuỗi ồn $\{w_k\}$ đã được thay thế bởi một số hàm của biến quan sát được $\{y_k\}$. Bộ dự báo tuyến tính được chọn theo phương trình (23.76) là:

$$\hat{y}_{k+1|k} = \frac{G(z^{-1})}{C(z^{-1})}y_k = \frac{c_1-a_1}{1+c_1z^{-1}}y_k \tag{23.82}$$

Bộ lọc Kalman

Xét mô hình không gian trạng thái tuyến tính:

$$\begin{aligned}x_{k+1} &= \Phi x_k + v_k, \qquad x_k\in\mathbb{R}^n\\ y_k &= Cx_k + w_k, \qquad y_k\in\mathbb{R}^m\end{aligned} \tag{23.83}$$

trong đó $\{v_k\}$ và $\{w_k\}$ được giả thiết là các quá trình ồn trắng trung bình không độc lập với các hiệp phương sai $\sum_v$ và $\sum_w$ tương ứng. Giả thiết rằng $\{y_k\}$ là đo được và có thể dự báo $\{x_k\}$ từ các phép đo $\{y_k\}$.

Xét bộ dự báo trạng thái:

$$\begin{aligned}\hat{x}_{k+1|k} &= \Phi\hat{x}_{k|k-1}-K_k(\hat{y}_k-y_k), \hat{x}_{k|k-1}\in\mathbb{R}^n\\ \hat{y}_k &= C\hat{x}_{k|k-1}, \qquad y_k\in\mathbb{R}^m\end{aligned} \tag{23.84}$$

Bộ dự báo của phương trình (23.84) có ma trận động học Φ giống như mô hình không gian trạng thái của phương trình (101.83) và số hạng hiệu chỉnh $K_k(\hat{y}_k-y_k)$ với hệ số K_k được chọn. Sai số dự báo là:

$$\tilde{x}_{k+1|k} = \hat{x}_{k+1|k} - x_{k+1} \tag{23.85}$$

Động học của sai số dự báo là:

$$\tilde{x}_{k+1} = (\Phi - K_kC)\tilde{x}_k + v_k - K_kw_k \tag{23.86}$$

Sai số dự báo trung bình được tính bởi phương trình đệ quy:

$$\mathcal{E}\{\tilde{x}_{k+1}\} = (\Phi - K_kC)\mathcal{E}\{\tilde{x}_k\} \tag{23.87}$$

và sai số bình phương trung bình của sai số dự báo được tính bởi:

$$\begin{aligned}\mathcal{E}\{\tilde{x}_{k+1}\tilde{x}_{k-1}^T\} &= \mathcal{E}\{[\Phi-K_kC)\tilde{x}_k+v_k-K_kw_k][(\Phi-K_kC)\tilde{x}_k+v_k-K_kw_k]^T\}\\ &= (\Phi-K_kC)\mathcal{E}\{\tilde{x}_k\tilde{x}_k^T\}(\Phi-K_kC)^T+\sum_v+K_k\sum_wK_k\end{aligned} \tag{23.88}$$

Nếu chúng ta ký hiệu

$$P_k = \mathcal{E}\{\tilde{x}_k\tilde{x}_k^T\}, \qquad Q_k = \sum_w + CP_kC^T \tag{23.89}$$

thì phương trình (23.88) được đơn giản thành:

$$P_{k+1} = \Phi P_k\Phi^T - K_kCP_k\Phi - \Phi^TP_kC^TK_k^T + \sum_v + K_kQ_kK_k^T \tag{23.90}$$

Bằng cách bình phương toàn bộ các số hạng chứa K_k chúng ta sẽ nhận được:

$$P_{k+1} = \Phi P_k \Phi^T + \Sigma_v - \Phi P_k C^T Q_k^{-1} C P_k \Phi^T + (K_k - \Phi P_k C^T Q_k^{-1}) Q_k (K_k - \Phi P_k C^T Q_k^{-1})^T \qquad (23.91)$$

trong đó chỉ số hạng cuối cùng phụ thuộc vào K_k. Việc tối giản P_{k+1} có thể được thực hiện bằng cách chọn K_k sao cho số hạng phụ thuộc K_k phía nửa dương trong phương trình (23.91) biến mất. Vì vậy P_{k+1} đạt được giới hạn dưới của nó với:

$$K_k = \Phi P_k C^T (\Sigma_w + C P_k C^T)^{-1} \qquad (23.92)$$

và *bộ lọc Kalman* (hoặc *bộ lọc Kalman–Bucy*) sẽ có dạng:

$$\begin{aligned} \hat{x}_{k+1|k} &= \Phi \hat{x}_{k|k-1} - K_k(\hat{y}_k - y_k) \\ \hat{y}_k &= C\hat{x}_{k|k-1}, \qquad K_k = \Phi P_k C^T (\Sigma_w + C P_k C^T)^{-1} \\ P_{k+1} &= \Phi P_k \Phi^T + \Sigma_v - \Phi P_k C^T (\Sigma_w + C P_k C^T)^{-1} C P_k \Phi^T \end{aligned} \qquad (23.93)$$

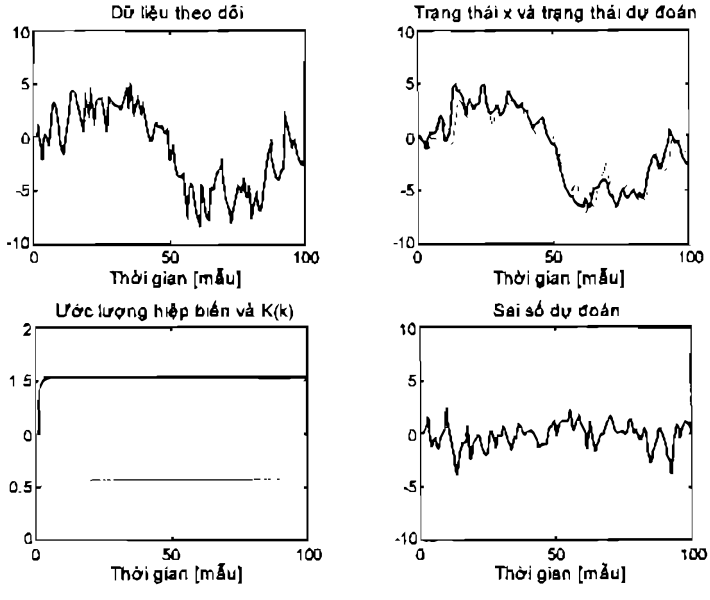
là bộ dự báo tối ưu theo nghĩa sai số bình phương trung bình (phương trình (23.88)) là tối thiểu theo mỗi bước.

Ví dụ 23.2—Bộ lọc Kalman cho một hệ bậc nhất

Xét mô hình không gian trạng thái

$$x_{k+1} = 0.95 x_k + v_k, \qquad y_k = x_k + w_k \qquad (23.94)$$

trong đó $\{v_k\}$ và $\{w_k\}$ là các quá trình ồn trắng trung bình không với các hiệp phương sai $\mathcal{E}\{v_k^2\} = 1$ và $\mathcal{E}\{w_k^2\} = 1$ tương ứng.



HÌNH 23.19 Bộ lọc Kalman được dùng để dự báo trước một bước x_{k+1} trong phương trình (23.94). Biến quan sát được $\{y_k\}$, trạng thái $\{x_k\}$ và trạng thái dự báo được $\{\hat{x}_k\}$, phương sai ước tính được $\{P_k\}$ và $\{K_k\}$, sai số dự báo trong 100 bước thực hiện của quá trình ngẫu nhiên.
(*Nguồn*: Johansson, R. 1993. *System Modeling and Identification*. Prentice-Hall, Englewood Cliffs, NJ.)

Các bộ lọc Kalman sẽ có dạng:

$$\hat{x}_{k+1|k} = 0.95\hat{x}_{k|k-1} - K_k(\hat{x}_{k|k-1} - y_k)$$
$$K_k = \frac{0.95P_k}{1+P_k} \tag{23.95}$$
$$P_{k+1} = 0.95^2 P_k + 1 - \frac{0.95^2 P_k^2}{1+P_k}$$

Kết quả được trình bày trong hình 23.19.

Định nghĩa các khái niệm

- **Mô hình AR (Autoregressive):** Một chuỗi thời gian tự lùi bậc n được định nghĩa bằng $y_k = -\sum_{m=1}^{n} a_m y_{k-m} + w_k$. Chuỗi $\{w_k\}$ thường được giả thiết bao gồm các biến ngẫu nhiên phân bố trung bình không giống nhau w_k.
- **Mô hình ARMA (Autoregressive moving average):** Một chuỗi thời gian trung bình động tự giảm bậc n được định nghĩa bằng $y_k = -\sum_{m=1}^{n} a_m y_{k-m} + \sum_{m=0}^{n} c_m w_{k-m}$. Chuỗi $\{w_k\}$ thường được giả thiết bao gồm các biến ngẫu nhiên phân bố trung bình không giống nhau w_k.
- **Biến đổi Laplace rời rạc:** Biến đổi Laplace rời rạc là một bản sao của biến đổi Laplace áp dụng cho các tín hiệu và các hệ rời rạc. Biến đổi Laplace rời rạc nhận được từ biến đổi z bằng cách thay thế $z = \exp(sT)$, trong đó T là chu kỳ lấy mẫu.
- **Quá trình MA (moving average):** Chuỗi thời gian trung bình động dịch bậc n được định nghĩa bởi $y_k = \sum_{m=0}^{n} c_m w_{k-m}$. Chuỗi $\{w_k\}$ thường được giả thiết bao gồm các biến ngẫu nhiên phân bố mean-zero giống nhau w_k.
- **Mô hình hữu tỷ:** AR, MA, ARMA, and ARMAX được gọi chung là mô hình hữu tỉ.
- **Chuỗi thời gian:** chuỗi của biến ngẫu nhiên $\{y_k\}$, trong đó k thuộc tập các số nguyên dương và âm.
- **Biến đổi z:** Một hàm tạo được áp dụng với các chuỗi dữ liệu và được đánh giá như một hàm của biến phức z theo tần số.

Tài liệu tham khảo

[1] Box, G. E. P. and Jenkins, G. M. 1970. *Time Series Analysis: Forecasting and Control.* Holden-Day, San Francisco, CA.

[2] Hurewicz, W. 1947. *Filters and servo systems with pulsed data. In Theory of Servomechanisms*, H. M. James, N. B. Nichols, and R. S. Philips, eds., McGraw-Hill, New York.

[3] Jenkins, G. M. and Watts, D. G. 1968. *Spectral Analysis and Its Applications.* Holden-Day, San Francisco, CA.

[4] Johansson, R. 1993. *System Modeling and Identification.* Prentice-Hall, Englewood Cliffs, NJ. Jury, E. I. 1956. Synthesis and critical study of sampled-data control systems. AIEE Trans. 75: 141–151.

[5] Kalman, R. E. and Bertram, J. E. 1958. *General synthesis procedure for computer control of single and multi-loop linear systems.* Trans. AIEE. 77: 602–609.

[6] Kolmogorov, A. N. 1939. *Sur l'interpolation et extrapolation des suites stationnaires.* C. R. Acad. Sci. 208: 2043–2045.

[7] Ragazzini, J. R. and Zadeh, L. A. 1952. *The analysis of sampled-data systems.* AIEE Trans. 71:225–234. Tsypkin, Y. Z. 1950. Theory of discontinuous control. Avtomatika i Telemekhanika. Vol. 5.

[8] Wiener, N. 1949. *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications.* John Wiley & Sons, New York.

23.3 Các mô hình không gian - trạng thái thời gian rời rạc và liên tục

Kam Leang, Qingze Zou, and Santosh Devasia

Giới thiệu

Trong mục này chúng tôi sẽ giới thiệu việc mô hình hóa các hệ thời gian liên tục và rời rạc sử dụng phương pháp không gian trạng thái. Phương pháp không gian trạng thái là phương pháp sử dụng một tập các phương trình vi phân bậc nhất để biểu diễn ứng xử của một hệ thống trong miền thời gian. Phương pháp không gian trạng thái có một thuận lợi hơn các phương pháp trong miền tần số như là phương pháp hàm truyền: có thể được dùng để mô hình hóa các hệ thống tuyến tính, phi tuyến, biến đổi theo thời gian và đa biến, trong đó phương pháp hàm truyền là phù hợp với các hệ tuyến tính bất biến theo thời gian (LTI) [1, chương 3]. Ngoài ra, các mô hình được biểu diễn ở dạng không gian trạng thái bậc nhất trong miền thời gian có thể dễ dàng được giải quyết bằng máy tính số hoặc bộ vi xử lý làm cho phương pháp này trở nên vô cùng hữu ích để thiết kế và điều khiển các hệ cơ điện tử hiện đại. Hơn nữa, có rất nhiều phần mềm máy tính như MATLAB [2], đã khai thác các lợi thế của dạng không gian trạng thái để phân tích và giải quyết các vấn đề của bài toán thiết kế. Vì vậy phương pháp không gian trạng thái có thể được dùng để nghiên cứu các ứng xử và thuận tiện khi thiết kế cả hệ thời gian liên tục và rời rạc, các nguyên tắc cơ bản của nó là tiêu điểm của mục này.

Sau đây, chúng ta bắt đầu bằng một ví dụ: Mô hình hóa một cơ cấu chấp hành piezoceramic và dùng ví dụ đó trong suốt mục này. Khái niệm trạng thái của hệ được đưa ra và chúng ta sẽ giải thích phương trình không gian trạng thái của các hệ tuyến tính và trình bày cách giải nó. Chủ đề tuyến tính hóa các hệ phi tuyến được đề cập một cách ngắn gọn. Mối quan hệ giữa các mô hình miền thời gian và mô hình miền tần số được đưa ra bàn luận và một quì trình để nhận được mô hình không gian trạng thái bằng cách dùng dữ liệu miền tần số thực nghiệm (đáp ứng tần số) cũng được trình bày. Mục này được kết thúc bằng việc xem xét mô hình hóa không gian trạng thái thời gian rời rạc và những nhận xét kết luận. Các lệnh thông dụng của MATLAB cũng được trình bày ở phần cuối.

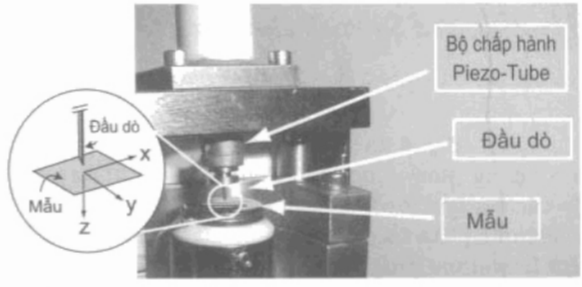
Trạng thái và không gian - trạng thái

Mô hình ví dụ của một cơ cấu chấp hành piezoceramic

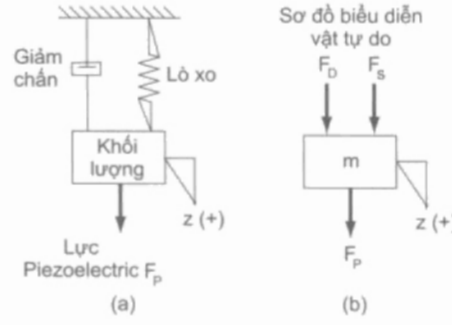
Chúng ta bắt đầu bằng việc mô hình hóa cơ cấu chấp hành piezoceramic, là một hệ cơ điện tử. Khi đặt một điện áp vào vật liệu piezoceramic thì kích thước của nó sẽ thay đổi. Sự thay đổi kích thước này có thể được dùng để xác định vị trí chính xác một đối tượng hoặc một công cụ (ví dụ như một đầu đo). Vì vậy tạo ra các cơ cấu chấp hành piezoceramic phù hợp cho nhiều ứng dụng rộng rãi khác nhau. Ví dụ nhờ khả năng xác định vị trí với độ chính xác cỡ nano, các cơ cấu chấp hành piezoceramic đã góp phần quan trọng để phát triển lĩnh vực công nghệ nano. Đặc biệt, một cơ cấu chấp hành ống piezo được dùng trong các kính hiển vi dò quét (Scanning Probe Microscopes - SPM, xem hình 23.20) để xác định vị trí chính xác đầu dò cho cơ khí chính xác cỡ nano, gia công bề mặt và thu thập ảnh của các nguyên tử [3].

Các đầu dò có thể được đặt trong hệ tọa độ x, y và z với mỗi dịch chuyển được điều khiển bởi một nguồn điện áp độc lập ($V_x(t)$, $V_y(t)$ và $V_z(t)$). Đầu dò thực hiện quét song song với bề mặt mẫu, dọc theo trục x và y; Sự dịch chuyển của trục x cho phép đầu dò dịch chuyển vuông

góc với bề mặt mẫu. Một mô hình toán chính xác cho động học của cơ cấu chấp hành ống piezo là cần thiết cho việc phân tích và thiết kế các hệ SPM. Người thiết kế có thể khai thác thông tin đã biết của hệ thống từ mô hình của nó để cải tạo hoặc tối ưu thiết kế sao cho việc xây dựng các SPM nhanh hơn và đáng tin cậy hơn. Ví dụ một phương pháp đã được thực hiện thành công là phương pháp điều khiển ngược. Để bám chính xác thì các đầu vào được xác định bằng cách đảo mô hình hệ thống [3]. Phương pháp này làm việc tốt nhất khi các động học của hệ được hiểu và mô tả tốt. Nói chung, phân tích và thiết kế các hệ thống điều khiển cũng yêu cầu một mô hình hệ thống. Vì vậy phân tích và thiết kế là then chốt để đạt được một mô hình toán chính xác mô tả ứng xử của một hệ thống. Việc mô hình hóa hệ ống piezo là ví dụ được xét sau đây.



HÌNH 23.20 Các thành phần chính của một kính hiển vi quét đầu dò (SPM) được dùng để phân tích bề mặt, bao gồm cơ cấu chấp hành ống piezo, đầu dò và mẫu. Cấu hình của đầu dò và mẫu tương ứng với các trục tọa độ (x, y và z) được phóng to trong hình.



HÌNH 23.21: (a) *Mô hình tập trung đơn giản* của cơ cấu chấp hành ống piezo dọc theo trục z bao gồm một khối vật, một lò so và một giảm chấn [4]. Hướng dương của z được chỉ ra bởi mũi tên và dấu " + ". (b) Các lực tác dụng lên vật (sơ đồ vật tự do).

Mô hình đơn giản của cơ cấu chấp hành ống piezo

Chúng ta sẽ mô hình hóa động học của cơ cấu chấp hành ống piezo dọc theo trục z trong đó đầu vào là điện áp cấp $V_z(t)$ và đầu ra của hệ thống là dịch chuyển của đầu dò $z(t)$. Mô hình hóa bắt đầu bằng việc đơn giản hóa hệ thống như là một khối vật độc lập, một lò xo lý tưởng và một giảm chấn như chỉ ra trên hình 23.21 (a). Toàn bộ khối lượng của ống piezo được quy về một phần tử khối lượng m, ứng xử đàn hồi bên trong của ống piezo được mô hình hóa là một lò xo và sự giảm chấn của cấu trúc trong ống piezo được mô hình hóa như một phần tử giảm chấn hoặc một phần tử ma sát nhớt (các mô hình như vậy được xem là các *mô hình tập trung* [4]). Mối quan hệ toán học giữa điện áp cấp $V_z(t)$ và dịch chuyển của đầu dò $z(t)$ có thể tìm được bằng cách sử dụng các định luật vật lý. Áp dụng định luật thứ 2 của Newton (tổng các ngoại lực đặt lên một vật bằng tích khối lượng m của nó với gia tốc $\ddot{z}(t)$) chúng ta có thể viết phương trình chuyển động như sau:

$$\sum_i F_i(t) = m\ddot{z}(t) \tag{23.96}$$

Như chỉ ra trong hình 23.21(b) (sơ đồ vật tự do), có 3 ngoại lực tác dụng lên ống piezo. Đầu tiên, lực được tạo bởi lò xo được giả thiết là tỷ lệ với dịch chuyển của đầu dò, có nghĩa là:

$$F_s(t) = -kz(t) \tag{23.97}$$

trong đó k là hằng số lò xo với đơn vị [N/m] theo hệ SI. Thứ hai, lực giảm chấn được xem là tỷ lệ với vận tốc của đầu dò $\dot{z}(t)$, có nghĩa là:

$$F_D(t) = -c\dot{z}(t) \tag{23.98}$$

trong đó c là hệ số ma sát dính với đơn vị là [Ns/m] theo SI. Thứ ba, là sức căng ε trong vật liệu áp gốm tỷ lệ với điện áp cấp $V_z(t)$ [5] và theo định luật Hook thì ứng suất σ tỷ lệ với sức căng ε. Vì vậy, lực $F_P(t)$ (ứng suất σ nhân với diện tích thiết diện) tỷ lệ với điện áp cấp $V_z(t)$, tức là:

$$F_P(t) = bV_z(t) \tag{23.99}$$

trong đó b là hằng số với đơn vị là [N/V] theo SI. Viết lại phương trình (23.96) theo ba ngoại lực thì phương trình trở thành:

$$\sum_{i=1}^{3} F_i(t) = F_s(t) + F_D(t) + F_P(t) = -kz(t) - c\dot{z}(t) + bV_z(t) = m\ddot{z}(t) \tag{23.100}$$

được gọi là mô hình *khối vật - lò xo - giảm chấn*. Chú ý rằng quan hệ giữa điện áp vào $V_z(t)$ và dịch chuyển $z(t)$ của đầu dò (tức là mô hình động lực học) là phương trình vi phân bậc hai. Đáp ứng của đầu dò (dịch chuyển của khối vật m) đối với điện áp $V_z(t)$ có thể tìm được trong miền tần số bằng cách dùng phương pháp biến đổi Laplace. Tuy nhiên phương pháp không gian trạng thái có thể được sử dụng để tìm ra lời giải trực tiếp trong miền thời gian. Trong các mục còn lại, phương pháp không gian trạng thái để mô hình hóa sẽ được trình bày và mô hình *khối vật - lò xo - giảm chấn* của cơ cấu chấp hành ống piezo được lấy làm ví dụ.

Các trạng thái của một hệ thống

Chúng ta bắt đầu bằng việc đưa ra khái niệm trạng thái, đó là cơ sở cho phương pháp không gian trạng thái. Nói chung, một trạng thái có thể được định nghĩa như sau:

Trạng thái $x(t_0)$ của một hệ thống động tại thời điểm t_0 là một tập các biến mà cùng với đầu vào $u(t)$, với $t \geq t_0$, xác định ứng xử của hệ đó ở tất cả thời điểm $t \geq t_0$.

Cơ sở của định nghĩa này là trạng thái sẽ phản ánh cấu hình hiện thời của một hệ thống. Vì vậy các dữ kiện về một hệ thống động được lưu giữ trong các biến trạng thái ở thời điểm hiện thời t_0 (được gọi là điều kiện đầu) và ứng xử trong tương lai của một hệ thống động được xác định bởi điều kiện đầu $x(t_0)$ và đầu vào $u(t)$, với $t \geq t_0$. Trạng thái của một hệ có thể được viết như sau:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{23.101}$$

trong đó n là số các trạng thái. Một tập bất kỳ các biến thỏa mãn định nghĩa trên có thể là một trạng thái, vì vậy trạng thái không phải là duy nhất.

Ví dụ

Các biến trạng thái để mô tả hệ khối vật - lò xo - giảm chấn có thể được chọn là vị trí $z(t)$ và vận tốc $\dot{z}(t)$ của khối lượng. Chúng ta có thể viết véc tơ trạng thái như sau:

$$x(t)=\begin{bmatrix}x_1(t)\\x_2(t)\end{bmatrix}=\begin{bmatrix}z(t)\\\dot{z}(t)\end{bmatrix} \tag{23.102}$$

trong đó số của các trạng thái là hai (n=2). Nếu vị trí $z(t)$ và vận tốc $\dot{z}(t)$ của khối vật là đã biết ở thời điểm t_0, cùng với điện áp $V_s(t)$ được xác định với $t \geq t_0$, thì ứng xử trong tương lai của hệ thống (tức là trạng thái $x(t)$) có thể được xác định bằng cách giải phương trình vi phân (23.100).

Phương trình không gian trạng thái tuyến tính và lời giải

Cho một hệ tuyến tính, sự biến thiên của các trạng thái của hệ theo thời gian có thể được mô tả bằng tập các phương trình vi phân tuyến tính bậc nhất sau:

$$\begin{aligned}
\dot{x}_1(t)&=\frac{dx_1(t)}{dt}=a_{11}(t)x_1(t)+\cdots+a_{1n}(t)x_n(t)+b_{11}(t)u_1(t)+\cdots+b_{1p}(t)u_p(t)\\
\dot{x}_2(t)&=\frac{dx_2(t)}{dt}=a_{21}(t)x_1(t)+\cdots+a_{2n}(t)x_n(t)+b_{21}(t)u_1(t)+\cdots+b_{2p}(t)u_p(t)\\
&\vdots\\
\dot{x}_n(t)&=\frac{dx_n(t)}{dt}=a_{n1}(t)x_1(t)+\cdots+a_{nn}(t)x_n(t)+b_{n1}(t)u_1(t)+\cdots+b_{np}(t)u_p(t)
\end{aligned} \tag{23.103}$$

trong đó n là số trạng thái (hoặc bậc của hệ) và p là số đầu vào. Định nghĩa véctơ đầu vào như sau:

$$u(t)=\begin{bmatrix}u_1(t)\\u_2(t)\\\vdots\\u_p(t)\end{bmatrix} \tag{23.104}$$

và véc tơ trạng thái $x(t)$ như được định nghĩa trong phương trình (23.101), tập các phương trình vi phân bậc nhất nhận được từ phương trình (23.103) có thể được viết lại theo dạng ma trận như sau:

$$\begin{aligned}
\dot{x}(t)&=\begin{bmatrix}a_{11}(t)&a_{12}(t)&\cdots&a_{1n}(t)\\a_{21}(t)&a_{22}(t)&\cdots&a_{2n}(t)\\\vdots&\vdots&\ddots&\vdots\\a_{n1}(t)&a_{n2}(t)&\cdots&a_{nn}(t)\end{bmatrix}x(t)+\begin{bmatrix}b_{11}(t)&b_{12}(t)&\cdots&b_{1p}(t)\\b_{21}(t)&b_{22}(t)&\cdots&b_{2p}(t)\\\vdots&\vdots&\ddots&\vdots\\b_{n1}(t)&b_{n2}(t)&\cdots&b_{np}(t)\end{bmatrix}u(t)\\
&=A(t)x(t)+B(t)u(t)
\end{aligned} \tag{23.105}$$

trong đó $A(t)$ là một ma trân $n\times n$ và $B(t)$ là một ma trân $n\times p$. Cho một hệ được xác định bằng q đầu ra $y(t)$ được giả thiết là một tổ hợp tuyến tính của trạng thái $x(t)$ và đầu vào $u(t)$, chúng ta có thể viết phương trình ra như sau:

$$\begin{aligned}
\dot{x}(t)&=\begin{bmatrix}c_{11}(t)&c_{12}(t)&\cdots&c_{1n}(t)\\c_{21}(t)&c_{22}(t)&\cdots&c_{2n}(t)\\\vdots&\vdots&\ddots&\vdots\\c_{q1}(t)&c_{q2}(t)&\cdots&c_{qn}(t)\end{bmatrix}x(t)+\begin{bmatrix}d_{11}(t)&d_{12}(t)&\cdots&d_{1p}(t)\\d_{21}(t)&d_{22}(t)&\cdots&d_{2p}(t)\\\vdots&\vdots&\ddots&\vdots\\d_{q1}(t)&d_{q2}(t)&\cdots&d_{qp}(t)\end{bmatrix}u(t)\\
&=C(t)x(t)+D(t)u(t)
\end{aligned} \tag{23.106}$$

trong đó $C(t)$ là một ma trận $q\times n$ và $D(t)$ là một ma trận $q\times p$. Nói chung, các ma trận $A(t)$, $B(t)$, $C(t)$ và $D(t)$ là thay đổi theo thời gian; tuy nhiên, trong chương này chúng ta sẽ chỉ xét đến

trường hợp bất biến theo thời gian, trong đó A, B, C và D là các ma trận không đổi. Khi đó phương trình (23.107) và (23.108) được gọi là các phương trình trạng thái và đầu ra tuyến tính bất biến theo thời gian (LTI).

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{23.107}$$

$$y(t) = Cx(t) + Du(t) \tag{23.108}$$

Đáp ứng của hệ đối với một đầu vào có thể được định lượng bằng biến thiên của trạng thái $x(t)$ của hệ và đầu ra $y(t)$. Phương trình không gian trạng thái (23.107) là tập các phương trình vi phân bậc nhất ở dạng ma trận có thể giải trong miền thời gian để nhận được điều kiện đầu $x(t_0)$ như sau:

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \tag{23.109}$$

Chú ý rằng nghiệm (23.109) là tổng của hai số hạng: số hạng đầu là hàm của điều kiện đầu $x(t_0)$ và số hạng thứ hai là hàm của đầu vào $u(t)$ giữa $t_0 \le \tau \le t$. Sử dụng phương trình đầu ra (23.108) và nghiệm trạng thái nhận được từ (23.109), thì đầu ra $y(t)$ có dạng

$$y(t) = Ce^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t) \tag{23.110}$$

Đáp ứng của hệ $y(t)$ đối với đầu vào $u(t)$ được đặc trưng bởi các ma trận của hệ (A, B, C, D). Ví dụ đầu ra $y(t)$ sẽ bị giới hạn bởi bất kỳ một đầu vào giới hạn nào nếu hệ thống ổn định và hệ thống là ổn định nếu phần thực của tất cả giá trị riêng của A nhỏ hơn 0 (âm hoàn toàn) [8, Chương 4, mục 4].

Ví dụ

Với ví dụ hệ khối vật - lò xo - giảm chấn, phương trình không gian trạng thái có thể tìm được bằng cách lấy vi phân trạng thái $x(t)$ được xác định trong phương trình (23.102) và dùng phương trình chuyển động (23.100) để nhận được:

$$\begin{aligned} \dot{x}_1(t) &= \dot{z}(t) = x_2(t) \\ \dot{x}_2(t) &= \ddot{z}(t) = -\left(\frac{k}{m}\right)z(t) - \left(\frac{c}{m}\right)\dot{z}(t) + \left(\frac{b}{m}\right)V_s(t) = -\left(\frac{k}{m}\right)x_1(t) - \left(\frac{c}{m}\right)x_2(t) + \left(\frac{b}{m}\right)u(t) \end{aligned} \tag{23.111}$$

Chúng ta chọn vị trí của khối vật $z(t)$ là đầu ra của hệ và viết phương trình không gian trạng thái và đầu ra nhận được từ các phương trình (23.107) và (23.108) như sau:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -(k/m) & -(c/m) \end{bmatrix}(t) + \begin{bmatrix} 0 \\ b/m \end{bmatrix}u(t) \tag{23.112}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}x(t) \tag{23.113}$$

Tuyến tính hóa các hệ phi tuyến

Một dạng tổng quát của phương trình không gian trạng thái (cho các hệ phi tuyến) là:

$$\dot{x}(t) = g(x,u) \tag{23.114}$$

$$y(t) = h(x,u) \tag{23.115}$$

trong đó g và h có thể là các hàm phi tuyến. Ứng xử của các hệ phi tuyến không thuộc phạm vi nghiên cứu của mục này. Tuy nhiên sẽ được đề cập chi tiết hơn ở [10]. Ứng xử của một hệ phi tuyến có thể được xấp xỉ bằng một mô hình tuyến tính ở lân cận điểm cân bằng. Vì vậy, việc

tuyến tính hóa có thể đơn giản hóa việc phân tích và thiết kế các hệ phi tuyến bởi vì các công cụ đã được phát triển cho các hệ tuyến tính có thể được sử dụng với những điều kiện nào đó [10]. Giả sử x_0 và u_0 là điểm cân bằng và đầu vào cân bằng sao cho:

$$g(x_0, u_0) = 0 \tag{23.116}$$

$$h(x_0, u_0) = y_0 \tag{23.117}$$

Xét các thay đổi nhỏ của điểm cân bằng $x(t) = x_0 + \bar{x}(t)$, đầu vào $u(t) = u_0(t) + \bar{u}(t)$ và đầu ra $y(t) = y_0 + \bar{y}(t)$. Nếu thay đổi $\bar{x}(t)$ là nhỏ với tất cả t thì chúng ta nhận được phương trình dưới bằng cách khai triển (23.114) thành chuỗi Taylor (bỏ qua các số hạng bậc cao của $\bar{x}(t)$ và $\bar{u}(t)$):

$$\begin{aligned} \dot{x}_0 + \dot{\bar{x}}(t) &= g(x_0 + \bar{x}(t), u_0 + \bar{u}(t)) \\ \dot{\bar{x}}(t) &= g(x_0, u_0) + \left.\frac{\partial g}{\partial x}\right|_{\substack{x=x_0 \\ u=u_0}} \bar{x}(t) + \left.\frac{\partial g}{\partial u}\right|_{\substack{x=x_0 \\ u=u_0}} \bar{u}(t) \end{aligned} \tag{23.118}$$

Thừa nhận rằng $g(x_0, x) = 0$ chúng ta nhận được:

$$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}\bar{u}(t) \tag{23.119}$$

trong đó:

$$\bar{\mathbf{A}} = \left.\frac{\partial g}{\partial x}\right|_{\substack{x=x_0 \\ u=u_0}} \qquad \bar{\mathbf{B}} = \left.\frac{\partial g}{\partial u}\right|_{\substack{x=x_0 \\ u=u_0}} \tag{23.120}$$

Các ma trận $\bar{A}$ và $\bar{B}$ là Jacobian tại x_0 and u_0. Phương trình (23.119) là phương trình trạng thái tuyến tính và đúng với các thay đổi nhỏ xung quanh x_0 và u_0. Kết quả tương tự có thể nhận được đối với thay đổi $\bar{y}(t)$ của đầu ra so với giá trị cân bằng y_0 như sau

$$\bar{y}(t) = \bar{C}\bar{x}(t) + \bar{D}\bar{u}(t) \tag{23.121}$$

trong đó

$$\bar{C} = \left.\frac{\partial h}{\partial x}\right|_{\substack{x=x_0 \\ u=u_0}} \qquad \bar{D} = \left.\frac{\partial h}{\partial u}\right|_{\substack{x=x_0 \\ u=u_0}} \tag{23.122}$$

Quan hệ giữa phương trình trạng thái và hàm truyền

Tính hàm truyền từ không gian trạng thái

Quan hệ vào ra của hệ động trong miền tần số được biểu diễn bằng một hàm truyền mà có thể tìm được bằng cách thực hiện biến đổi Laplace của (23.107) và (23.108) với các điều kiện đầu, như sau [8, Chương 3, mục 5]:

$$sX(s) = AX(s) + BU(s) \tag{23.123}$$

$$Y(s) = CX(s) + DU(s) \tag{23.124}$$

trong đó s là biến Laplace. Giải (23.123) để tìm $X(s)$ và thay vào (23.124), tỉ số của đầu ra $Y(s)$ với đầu vào $U(s)$ cho một hệ có một đầu vào và một đầu ra (SISO) có thể tìm được như sau:

$$\begin{aligned} G(s) &= \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \\ &= \frac{N(s)}{D(s)} \end{aligned} \tag{23.125}$$

trong đó I là một ma trận đơn vị $n \times n$. Trong phương trình (23.125), $N(s)$ và $D(s)$ được xem như là đa thức tử số và mẫu số của $G(s)$.

Tương tự phương trình không gian trạng thái, giới hạn của đáp ứng ra $y(t)$ đối với đầu vào giới hạn $u(t)$ được đặc trưng bởi các nghiệm của đa thức mẫu số $D(s)$, tức là các giá trị của s để $D(s) = 0$. Đặc biệt, đầu ra $y(t)$ sẽ bị giới hạn với bất kỳ đầu vào giới hạn nào, có nghĩa là hệ ổn định nếu phần thực của tất cả các biến của $D(s)$ nhỏ hơn không (hoàn toàn âm). Ngoài ra, một phương pháp thuận tiện để xác định tính ổn định mà không cần tìm nghiệm của $D(s)$ một cách tường minh là tiêu chuẩn ổn định Routh–Hurwitz.

Ví dụ

Với mô tả không gian trạng thái của hệ khối vật - lò xo - giảm chấn được xác định theo các phương trình (23.112) và (23.113) thì việc xây dựng hàm truyền bằng cách sử dụng phương trình (23.125) sẽ là:

$$G(s) = \frac{Y(s)}{U(s)} = \begin{bmatrix} 1 & 0 \end{bmatrix} \left[s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -(k/m) & -(c/m) \end{bmatrix} \right]^{-1} \begin{bmatrix} 0 \\ b/m \end{bmatrix} + [0]$$
$$= \frac{b/m}{s^2 + (c/m)s + k/m} \tag{23.126}$$

Đầu vào của hệ thống là điện áp $V_2(t)$ và đầu ra là sự dịch chuyển của khối vật $z(t)$.

Tính đáp ứng tần số bằng cách sử dụng hàm truyền

Xét một hệ ổn định với một đầu vào và một đầu ra (SISO) có hàm truyền $G(s)$. Khi hệ $G(s)$ được kích bằng một tín hiệu vào dạng sin :

$$u(t) = P\sin(\omega t) \tag{23.127}$$

với biên độ P và tần số ω, thì đáp ứng ra (sau khi kết thúc trạng thái quá độ) cũng sẽ có dạng sin:

$$y(t) = MP\sin(\omega t + \phi) \tag{23.128}$$

với cùng tần số ω và độ lệch pha ϕ [6, Chương 8]. Biên độ của tín hiệu ra là biên độ của tín hiệu vào nhân với M, là hệ số khuyếch đại cường độ. Hệ số khuyếch đại cường độ này nhận được bằng cách lấy độ lớn của $G(s)$ tại $s=j\omega$, tức là :

$$M = \left| G(s) \right|_{s = j\omega} \tag{23.129}$$

Thường hệ số khuyếch đại cường độ M được biểu diễn theo đơn vị (dB), tức M[dB] =20log$M$. Độ lệch pha ϕ là góc của $G(s)$ được lấy tại $s=j\omega$, tức là:

$$\phi = \angle G(s)\big|_{s = j\omega} \tag{23.130}$$

với đơn vị độ. Biểu đồ hệ số khuyếch đại cường độ M và độ lệch pha ϕ đối với tần số ω là sự biểu diễn bằng đồ thị đáp ứng tần số (các biểu đồ Bode) của một hệ thống. Các biểu đồ này có thể nhận được bằng thực nghiệm bằng cách đo hệ số khuyếch đại độ lớn và độ lệch pha giữa tín hiệu vào và đáp ứng ra của một hệ trên một dải tần số. Ngoài ra, hàm truyền của một hệ có thể nhận được từ dữ liệu đáp ứng tần số thực nghiệm bằng cách dùng phần mềm thích hợp. Trong mục "Mô hình hóa bằng thực nghiệm sử dụng đáp ứng tần số", chúng tôi trình bày phương pháp này để xác định mô hình cho một hệ dùng dữ liệu đáp ứng tần số thực nghiệm.

Tính không gian trạng thái từ hàm truyền

Trong mục "Tính hàm truyền từ không gian trạng thái", một mô hình hàm truyền nhận được cho một hệ thống dưới dạng không gian trạng thái. Phần này sẽ trình bày một phương pháp xác

định mô hình không gian trạng thái từ một hàm truyền $G(s)$. Cho một hàm truyền $G(s)$ của hệ SISO có dạng:

$$G(s)=\frac{b_0 s^n+b_1 s^{n-1}+\cdots+b_n}{s^n+a_1 s^{n-1}+\cdots+a_n} \tag{23.131}$$

dạng không gian trạng thái kinh điển điều khiển được viết theo các hệ số của $G(s)$ là :

$$\dot{x}(t)=\begin{bmatrix}-a_1 & -a_2 & \cdots & -a_{n-1} & -a_n\\ 1 & 0 & \cdots & 0 & 0\\ 0 & 1 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & 1 & 0\end{bmatrix}x(t)+\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}u(t) \tag{23.132}$$

$$y(t)=[(b_1-a_1b_0)\quad (b_2-a_2b_0)\quad \cdots\quad (b_n-a_nb_0)]x(t)+[b_0]u(t) \tag{23.133}$$

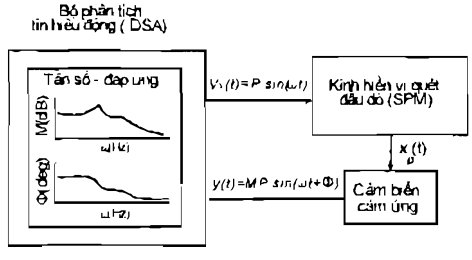
trong đó số các trạng thái n bằng lũy thừa cao nhất của mẫu số của $G(s)$. Cỡ ma trận nhỏ nhất có thể để mô tả một hệ thống, được gọi là mô tả tối thiểu, là một yếu tố quan trọng cần xem xét khi phân tích và thiết kế. Các mô hình với bậc tối thiểu đòi hỏi tính toán ít hơn khi mô phỏng và thực hiện so với các mô hình bậc cao hơn.

Mô hình hóa thực nghiệm sử dụng đáp ứng tần số

Phương pháp mô hình hóa sử dụng dữ liệu của đáp ứng tần số thực nghiệm sẽ được trình bày trong mục này. Bộ phân tích tín hiệu động (DSA) được sử dụng để đo đáp ứng tần số của các tín hiệu động dọc theo trục x của cơ cấu chấp hành ống piezo. Một điện áp đầu vào dạng sin $V_x(t)$ với tần số thay đổi trong khoảng từ 10 Hz đến 6 kHz được tạo ra bởi bộ DSA và được cấp cho hệ kính hiển vi quét đầu dò (SPM) (xem hình 23.22). Cảm biến kiểu cảm ứng được dùng để đo dịch chuyển $x_p(t)$ của ống piezo dọc theo trục x. Giá trị đó được đưa về DSA để tính đáp ứng tần số (các biểu đồ M và ϕ theo tần số ω). Trong hình 23.23 các biểu đồ Bode nhận được bằng DSA giữa điện áp cấp $V_x(t)$ và tín hiệu ra $y(t)$ của cảm biến cảm ứng. Việc tính toán mô hình của hệ thống từ dữ liệu của đáp ứng tần số có thể thực hiện bằng phần mềm MATLAB. Hàm truyền giữa điện áp vào $V_x(t)$ và tín hiệu ra $y(t)$ của cảm biến kiểu cảm ứng tìm được là:

$$G_1(s)=\frac{Y(s)}{V_x(s)}$$

$$=\frac{5.544\times10^5 s^4-7.528\times10^9 s^3+1.476\times10^{15}s^2-4.571\times10^{18}s+9.415\times10^{23}}{s^6+1.255\times10^4 s^5+1.632\times10^9 s^4+1.855\times10^{13}s^3+6.5\times10^{17}s^2+6.25\times10^{21}s+1.378\times10^{25}} \tag{23.134}$$

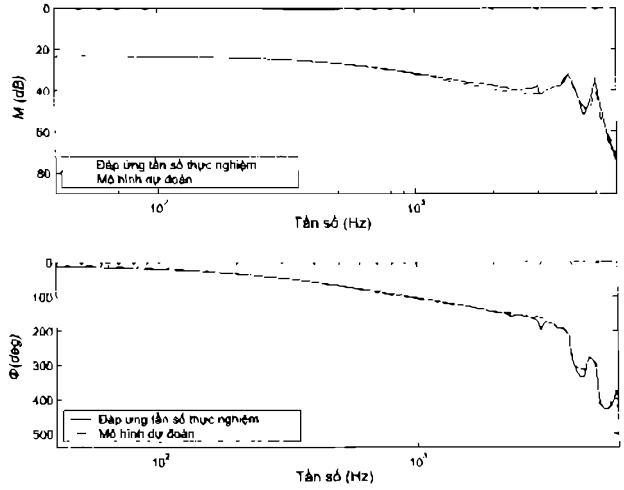


HÌNH 23.22 Một sơ đồ thực nghiệm được dùng để xác định đáp ứng tần số của cơ cấu chấp hành ống piezo

với đơn vị V/V. Phương trình (23.135) đã được nhân thêm hệ số khuyếch đại của cảm biến cảm ứng (30 Å/V). Hàm truyền giữa điện áp cấp $V_x(t)$ và dịch chuyển thực của đầu dò ống piezo $x_p(t)$ sẽ là:

$$G_2(s)=\frac{X_p(s)}{V_s(s)}$$

$$=\frac{1.663\times10^{7}s^{4}-2.258\times10^{11}s^{3}+4.427\times10^{14}s^{2}-1.371\times10^{20}s+2.825\times10^{25}}{s^{6}+1.255\times10^{4}s^{5}+1.632\times10^{9}s^{4}+1.855\times10^{13}s^{3}+6.5\times10^{17}s^{2}+6.25\times10^{21}s+1.378\times10^{25}} \tag{23.135}$$



HÌNH 23.22 Đáp ứng tần số-biên độ và pha thực nghiệm của cơ cấu chấp hành ống piezo

Chuyển đổi đơn vị thời gian của một mô hình hàm truyền

Sau đây chúng tôi trình bày một phương pháp để chuyển đơn vị thời gian cho $G_2(s)$ từ [s] sang [ms]. Chúng tôi tóm tắt lại tính chất chuyển đổi thời gian của hàm truyền Laplace đã được trình bày ở [1, Chương 3, mục 1.4]. Giả sử $F(s)$ là hàm truyền Laplace của $f(t)$, tức là:

$$f(t)\xrightarrow{L}F(s) \tag{23.136}$$

trong đó L ký hiệu toán tử hàm truyền Laplace. Bây giờ, xét một đơn vị thời gian mới được định nghĩa là $\hat{t}=at$ trong đó a là hằng số. Hàm truyền Laplace của $f(\hat{t})=f(at)$ nhận được là:

$$f(\hat{t})=f(at)\xrightarrow{L}\frac{1}{|a|}F\left(\frac{s}{a}\right)=\hat{F}(s) \tag{23.137}$$

Dùng công thức (23.137), chúng ta có thể giảm bớt các hệ số của $G_2(s)$ bằng cách thay đổi đơn vị thời gian của cả tín hiệu vào $u(t)$ và ra $y(t)$ như sau:

$$\hat{G}(s)=\frac{\hat{Y}(s)}{\hat{U}(s)}=\frac{Y(s/a)/|a|}{U(s/a)/|a|}=\frac{Y(s/a)}{U(s/a)}=G\left(\frac{s}{a}\right) \tag{23.138}$$

Vì vậy, để thay đổi đơn vị thời gian cho $G_2(s)$ từ [s] thành [ms], chúng tôi chọn $\hat{t}=at=0.001t$ và hàm truyền mới $\hat{G}_2(s)$ sau khi thay đổi sẽ là:

$$\hat{G}_2(s)=G_2\left(\frac{s}{a}\right)\Big|_{a=0.001}$$

$$=G_2(1000s) \tag{23.139}$$

$$\hat{G}_2(s)=\frac{16.63s^{4}-225.8s^{3}+4.427\times10^{4}s^{2}-1.371\times10^{5}s+2.825\times10^{7}}{s^{6}+12.55s^{5}+1.632\times10^{4}s^{4}+1.855\times10^{5}s^{3}+6.5\times10^{5}s^{2}+6.25\times10^{5}s+1.378\times10^{7}}$$

Chú ý rằng đơn vị thời gian của tín hiệu vào và ra của $\hat{G}_2(s)$ bây giờ là [ms]. Các hệ số của các đa thức tử và mẫu là nhỏ hơn và dạng này ($\hat{G}_2(s)$) sẽ ít xảy ra các lỗi tính toán hơn dạng $G_2(s)$ ở phương trình (23.135).

Mô hình không gian trạng thái

Mô hình không gian trạng thái cho $\hat{G}_2(s)$ biểu diễn trong các phương trình (23.132) và (23.133)) như sau:

$$x(t) = \begin{bmatrix} -12.55 & -1.632\times10^3 & -1.855\times10^4 & -6.50\times10^5 & -6.25\times10^6 & -1.378\times10^7 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \quad (23.140)$$

$$y(t) = [0 \quad 16.63 \quad -225.8 \quad 4.427\times10^4 \quad -1.371\times10^5 \quad 2.825\times10^7]x(t) \quad (23.141)$$

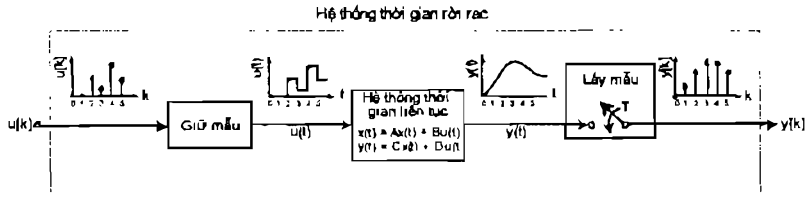
Đơn vị thời gian của phương trình (23.140) và (23.141) là [ms]. Nếu trạng thái ban đầu ở t_0 đã biết và điện áp $V_x(t)$ xác định với $t \geq t_0$ thì ứng xử tương lai của hệ tức là trạng thái $x(t)$ và đầu ra $y(t)$ có thể được xác định từ phương trình (23.140) và (23.141).

Mô hình hóa không gian trạng thái thời gian rời rạc

Giới thiệu

Việc nghiên cứu các hệ thời gian rời rạc là rất quan trọng để phân tích và thiết kế các hệ cơ điện tử hiện đại mà ở đó máy tính số hoặc các bộ vi xử lý nhỏ được sử dụng chủ yếu cho các hệ điều khiển. Các máy tính số hoặc các bộ vi xử lý đưa ra hoặc thu thập thông tin tại các thời điểm rời rạc. Ví dụ, tín hiệu vào cung cấp cho một máy tính số để điều khiển ống piezo sẽ thay đổi tại các thời điểm rời rạc. Tương tự, dịch chuyển của ống piezo chỉ có thể đo được ở các thời điểm xác định bằng cách dùng các máy tính số. Vì vậy, so với một hệ thống điều khiển thời gian liên tục mà ở đó các tín hiệu vào thay đổi liên tục theo thời gian thì tín hiệu vào của một hệ thời gian rời rạc sẽ thay đổi một lần trong một khoảng thời gian. Các hệ thời gian rời rạc như vậy sẽ được nghiên cứu tiếp trong các phần sau.

Xét một hệ thời gian liên tục với tín hiệu vào $u(t)$ và tín hiệu ra $y(t)$ như đã mô tả bởi các công thức (23.107) và (23.108). Dùng máy tính hoặc bộ vi xử lý để cung cấp tín hiệu vào $u[k]$ và đo tín hiệu ra $y[k]$ như mô tả trong hình 23.24 (các hệ thống với các tín hiệu liên tục và rời rạc như vậy được gọi là các hệ lấy mẫu dữ liệu).



HÌNH 23.24 Sơ đồ khối của một hệ thời gian rời rạc biểu diễn các tín hiệu ở dạng đồ thị. Chú ý rằng $u[k]=u(k.T)$ và $y[k]=y(k.T)$, với $k=0,1,2,...$ và chu kỳ lấy mẫu T được giả thiết là hằng số

Tín hiệu vào $u[k]$ và tín hiệu ra $y[k]$ của hệ thống này là tín hiệu rời rạc với $u[k]=u(k.T)$ và $y[k]=y(k.T)$ với $k=0,1,2,...$ trong đó chu kỳ lấy mẫu T là hằng số. Tín hiệu vào $u[k]$ được cấp

cho hệ thống liên tục từ một máy tính số hoặc bộ vi xử lý và được giữ không đổi trong khoảng thời gian T. Bộ lấy mẫu sẽ thu thập tín hiệu ra của hệ liên tục ở mỗi khoảng thời gian T tạo ra tín hiệu ra rời rạc $y[k]$. Hệ rời rạc là quan hệ giữa tín hiệu vào $u[k]$ và tín hiệu ra $y[k]$ [11, Chương 1]. Biểu diễn tương đương không gian trạng thái hệ thống thời gian rời rạc của mô hình không gian trạng thái hệ thời gian liên tục nhận được từ các phương trình (23.107) và (23.108) có dạng như sau:

$$x[k+1] = A_D x[k] + B_D u[k] \tag{23.142}$$

$$y[k] = C_D x[k] + D_D u[k] \tag{23.143}$$

trong đó

$$A_D = e^{AT}, \quad B_D = \left(\int_0^T e^{A\lambda} d\lambda \right) B, \quad C_D = C \quad D_D = D \tag{23.144}$$

và các ma trận C_D and D_D không bị thay đổi bởi việc lấy mẫu. Mô hình rời rạc này (các phương trình (23.142) và (23.143)) là biểu diễn của hệ thống lấy mẫu dữ liệu đã chỉ ra trên hình 23.24.

Nghiệm của phương trình không gian - trạng thái của hệ thời gian rời rạc

Nghiệm của mô hình rời rạc (các phương trình (23.142) và (23.143)) là

$$x[k] = A_D^k x[0] + \sum_{j=0}^{k-1} A_D^{k-j-1} B_D u[j] \tag{23.145}$$

$$y[k] = CA_D^k x[0] + C\sum_{j=0}^{k-1} A_D^{k-j-1} B_D u[j] + Du[k] \tag{23.146}$$

cho mỗi bước lấy mẫu k. Đáp ứng trạng thái $x[k]$ đối với tín hiệu vào $u[k]$ được đặc trưng bởi các ma trận hệ thống (A_D, B_D ,C_D, D_D). Đặc biệt, tín hiệu ra $y[k]$ sẽ bị giới hạn với bất kỳ tín hiệu vào giới hạn $u[k]$ nào nếu hệ thống là ổn định. Một hệ thống ở dạng phương trình (23.142) là ổn định nếu độ lớn của tất cả các giá trị riêng của A_D là nhỏ hơn đơn vị, tức là nằm trong đường tròn đơn vị mà tâm là gốc của mặt phẳng z.

Biến đổi z và mối quan hệ với không gian - trạng thái

Quan hệ vào-ra trong miền tần số của các hệ thống thời gian rời rạc được biểu diễn bằng một hàm truyền rời rạc được gọi là biến đổi z, được viết theo các số hạng của biến. Tương tự với hệ thời gian liên tục, mô hình một hệ động theo dạng hàm truyền rời rạc có thể được dùng trong thiết kế và điều khiển các hệ. Nếu mô hình hệ thống đã ở dạng hàm truyền rời rạc thì mô hình không gian trạng thái có thể xác định như sau. Cho một hệ rời rạc được mô tả bằng biến đổi z:

$$G(z) = \frac{d_0 + d_1 z^{-1} + \cdots + d_n z^{-n}}{1 + c_1 z^{-1} + \cdots + c_n z^{-n}} \tag{23.147}$$

dạng không gian trạng thái kinh điển có thể điều khiển của $G(z)$ là:

$$x[k+1] = \begin{bmatrix} -c_1 & -c_2 & \cdots & -c_{n-1} & -c_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} x[k] + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u[t] \tag{23.148}$$

$$y[k] = [(d_1 - c_1 d_0)\ (d_2 - c_2 d_0) \cdots (d_n - c_n d_0)] x[k] + [d_0] u[k] \tag{23.149}$$

Số trạng thái n tương đương với lũy thừa cao nhất của đa thức mẫu số của $G(z)$. Các thông tin về các dạng không gian trạng thái kinh điển khác tham khảo tại [11, Chương 5, mục 2].

Ví dụ

Xét mô hình không gian trạng thái thời gian liên tục của hệ ống piezo được mô tả bằng phương trình (23.140) và (23.141). Một máy tính số với tốc độ lấy mẫu 10 kHz ($T=1.0\times10^{-4}$) được dùng để đưa ra tín hiệu vào điều khiển $u[k]$ và đo dịch chuyển của nó dọc theo trục x (tín hiệu ra $y[k]$). Mô hình không gian trạng thái thời gian rời rạc với (A_D, B_D, C_D và D_D) nhận được từ phương trình (23.144) là:

$$x[k+1]=\begin{bmatrix} 0.999 & -0.163 & -1.85 & -65.0 & -624.5 & -1377.1 \\ 9.99\times10^{-5} & 0.999 & -9.26\times10^{-5} & -3.25\times10^{-3} & -3.12\times10^{-2} & -6.69\times10^{-2} \\ 5.00\times10^{-9} & 1.00\times10^{-4} & 1 & -1.08\times10^{-7} & -1.04\times10^{-6} & -2.30\times10^{-6} \\ 1.67\times10^{-13} & 5.00\times10^{-9} & 1.00\times10^{-4} & 1 & -2.60\times10^{-11} & -5.74\times10^{-11} \\ 4.17\times10^{-18} & 1.67\times10^{-13} & 5.00\times10^{-9} & 1.00\times10^{-4} & 1 & -1.15\times10^{-14} \\ 8.33\times10^{-23} & 4.17\times10^{-18} & 1.67\times10^{-13} & 5.00\times10^{-9} & 1.00\times10^{-4} & 1 \end{bmatrix} x[k] + \begin{bmatrix} 9.99\times10^{-5} \\ 4.99\times10^{-9} \\ 1.67\times10^{-13} \\ 4.17\times10^{-18} \\ 8.33\times10^{-23} \\ 1.39\times10^{-27} \end{bmatrix} u[k] \quad (23.150)$$

$$y[k]=\begin{bmatrix} 0 & 16.63 & -225.8 & 4.427\times10^{4} & -1.371\times10^{5} & 2.825\times10^{7} \end{bmatrix} x[k] \quad (23.151)$$

Phương trình (23.150) và (23.151) có thể tìm được bằng cách dùng lệnh 'c2d' của MATLAB.

Tóm tắt

Phần này đã trình bày các công cụ để mô hình hóa các hệ thời gian liên tục và rời rạc bằng cách dùng phương pháp không gian trạng thái. Phương pháp không gian trạng thái để mô hình hóa là một kỹ thuật mạnh để phân tích, thiết kế các hệ cơ điện tử và các hệ động lực và rất lợi thế khi tận dụng các công cụ có sẵn trong các máy tính số và các bộ vi xử lý. Các thảo luận về trạng thái của hệ và không gian trạng thái được minh họa bằng một hệ thống cơ cấu chấp hành ống piezo. Chúng ta đã xét việc mô hình hóa các hệ tuyến tính và kỹ thuật để tuyến tính hóa các hệ phi tuyến được giới thiệu một cách ngắn gọn. Đáp ứng tần số của một hệ thống và phương pháp mô hình hóa bằng cách sử dụng dữ liệu đáp ứng tần số thực nghiệm cũng đã được trình bày. Các mối quan hệ giữa các mô hình biểu diễn trong miền thời gian và tần số cho cả hệ thời gian liên tục và rời rạc cũng đã được bàn tới. Chi tiết về các khái niệm đã đề cập trong mục này và những nội dung liên quan bạn đọc có thể tham khảo trong các tài liệu sau.

Tài liệu tham khảo

[1] Franklin, G. F., et al., *Feedback Control of Dynamic Systems*, 3rd ed., Addison-Wesley, New York, 1994.

[2] Hanselman, D., and Littlefield, B., *The Student Edition of Matlab, Version 5, User's Guide*, Prentice-Hall, Upper Saddle River, NJ, 1997.

[3] Croft, D., et al., Creep, hysteresis, and vibration compensation for piezoactuators: atomic force microscopy application, ASME J. Dyn. Syst., Meas., Control, 123, 35, 2001.

[4] Dorny, C. N., *Understanding Dynamic Systems—Approaches to Modeling, Analysis, and Design*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[5] Locatelli, M., et al., *Easy method to characterize a piezoelectric ceramic tube as a displacer*, Rev. Sci. Instrum., 59, 4, 1988.

[6] Dorf, R. C., and Bishop, R. H., *Modern Control Systems*, 9th ed., Prentice-Hall, Upper Saddle River, NJ, 2001.

[7] Chen, T. C., *Linear System Theory and Design, Oxford University Press*, New York 1999.

[8] Friedland, B., *Control System Design. An Introduction to State-Space Methods* McGraw-Hill, New York, 1986.

[9] Gillis, J. T., *State space, in The Control Handbook*. Levine, W. S., CRC Press, Salem, MA, 1996, Chap. 5.

[10] Khalil, H. K., *Nonlinear Systems*, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 1996

[11] Ogata, K., *Discrete-Time Control Systems*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ. 1995.

[12] Franklin, G. F., et al., *Digital Control of Dynamic Systems*, 3rd ed., Addison-Wesley, Menlo Park, 1998.

23.4 Hàm truyền và biến đổi Laplace

C. Nelson Dorny

Chúng ta đã hiểu một cách căn bản một hệ thống qua ứng xử của nó. Vì vậy, hình ảnh trong đầu của chúng ta về một hệ thống thường bao gồm các tín hiệu đáp ứng điển hình. *Đáp ứng bước* (*step response*), là ứng xử khi chúng ta kích hoạt đột ngột hệ thống, vì vậy nó là một tín hiệu đặc trưng của hệ. *Đáp ứng xung* (*impulse response*) là một mô tả khác của hệ thống. Với một hệ được biểu diễn bằng các phương trình vi phân tuyến tính, đáp ứng bước đơn vị là tích phân của đáp ứng xung đơn vị.

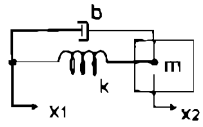
Ta biểu diễn vi phân theo thời gian (d/dt) bằng *toán tử đạo hàm theo thời gian p*. Khi đó chúng ta có thể kí hiệu đạo hàm theo thời gian của một tín hiệu y bằng py, đạo hàm bậc hai của nó là p^2y, tích phân của nó theo thời gian bằng $(1/p)y$. Kí hiệu toán tử như vậy làm đơn giản việc biểu diễn các phương trình vi phân. Chúng ta sẽ dùng cụm từ *các phương trình hệ thống* để chỉ tập hợp các phương trình vi phân xác định toàn bộ ứng xử của các biến phụ thuộc xuất hiện trong các phương trình đó. Chúng ta có thể giảm một tập tuyến tính các phương trình của hệ thành một **phương trình vào-ra của hệ** bằng cách loại bỏ tất cả khỏi tập đó trừ một biến phụ thuộc. *Hàm truyền* (*transfer function*) cùng với biến phụ thuộc đó là một biểu thức toán học chứa tất cả các thông tin cốt yếu thể hiện trong phương trình vi phân của hệ.

Phép biến đổi Laplace chuyển các tín hiệu (các hàm thời gian) thành các hàm của *một biến tần số phức* $s = \sigma + j\,\omega$. Đó là một sự tương ứng một - một giữa một tín hiệu và biến đổi Laplace của nó. Chúng ta có thể tìm lại được hàm thời gian bằng phép biến đổi ngược. Biến đổi Laplace sẽ tạo ra các ảnh có các tính chất thuận tiện hơn các tính chất của các tín hiệu gốc. Đặc biệt, vi phân theo thời gian một tín hiệu sẽ tương ứng với việc nhân biến đổi Laplace của nó với biến tần số phức s. Vì vậy, phép biến đổi sẽ biến các phương trình vi phân hệ số hằng tuyến tính thành các phương trình đại số tuyến tính. Việc đơn giản hóa như vậy của các phép toán miền thời gian làm cho biến đổi Laplace trở nên có ích. Biến đổi Laplace cũng biến đáp ứng xung của một biến hệ thống thành hàm truyền cho biến đó. Vì vậy chúng ta có thể thấy phương trình vi phân biểu diễn một hệ tuyến tính như là một biểu thức của đáp ứng của hệ đó đối với một tín hiệu vào dạng xung.

Các hàm truyền

Các dịch chuyển của nút x_1 và x_2 và các lực nén f_1 và f_2 bên trong các nhánh của mô hình tổng thể ở hình 23.25 liên quan tới nhau bằng phương trình lò xo, phương trình giảm chấn và sự cân bằng của các lực ở nút 2. Phương trình của lò xo là $f_2 = k(x_1 - x_2)$. Phương trình cho bộ giảm chấn là $f_1 = b(px_1 - px_2)$. Việc cân bằng các lực đòi hỏi $f_1 + f_2 = mp^2x_2$. Các phương trình này mô tả đầy đủ ứng xử của hệ khi lò xo và khối vật không được cung cấp có lượng. (Nếu khối vật

đang chuyển động và hoặc lò xo đang bị nén thì chúng ta có thể phải biểu diễn riêng rẽ các trạng thái năng lượng ban đầu của chúng để mô tả đầy đủ các quan hệ sau này giữa các biến.)



HÌNH 23.25 Mô hình tổng thể của hệ cơ học

Khử f_1 và f_2 khỏi các phương trình để nhận được phương trình hoạt động:

$$(mp^2 + bp + k)x_2 = (bp + k)x_1 \tag{23.152}$$

Phương trình vi phân này mô tả đầy đủ quan hệ của trạng thái không giữa x_1 và x_2. Sắp xếp lại phương trình (23.152) thành dạng tỉ số:

$$\frac{x_2}{x_1} = \frac{bp + k}{mp^2 + bp + k} \tag{23.153}$$

Chúng ta gọi (23.152) là *hàm truyền* từ x_1 đến x_2. Hàm truyền tập trung chú ý vào các phép toán đặc trưng cho các quan hệ ứng xử hơn là vào bản chất cụ thể của các biến. (Chú ý rằng hàm truyền từ v_1 đến v_2 trong đó $v_1 = px_1$ và $v_2 = px_2$ là giống hàm truyền nhận được bởi phương trình (23.153).)

Nói chung, giả thiết rằng y_1 và y_2 là hai biến liên quan (theo ký hiệu toán tử) bởi phương trình vi phân tuyến tính:

$$y_2 = G(p)y_1 \tag{23.154}.$$

Chúng ta định nghĩa *hàm truyền* từ y_1 tới y_2 bằng

$$G(p) = \left.\frac{y_2}{y_1}\right|_{ZS} \tag{23.155}$$

trong đó ký hiệu ZS là trạng thái không. Nếu y_1 là biến độc lập thì $G(p)$ là *hàm truyền vào-ra* cho biến y_2 và mô tả đầy đủ ứng xử của nó đối với tín hiệu vào y_1. Chúng ta có thể xác định từ hàm truyền đó ứng xử của hệ thống đối với bất kỳ tín hiệu nguồn dạng sóng và trạng thái ban đầu.

Biến đổi Laplace

Biến đổi Laplace một vế, $\mathcal{L}$, là toán tử tích phân để biến một tín hiệu $f(t)$ thành một hàm phức $F(s)$ có dạng như sau:

$$\mathcal{L}[f(t)] \equiv F(s) \overset{\Delta}{=} \int_{0^-}^{\infty} f(t)e^{-st}dt \tag{23.156}$$

Gọi hàm truyền $F(s)$ là *biến đổi Laplace* của tín hiệu $f(t)$. Hãy tưởng tượng giới hạn dưới 0^- của tích phân là một khoảng thời gian ngay trước khi t=0. Thông thường, dùng ký hiệu chữ thường (f) để biểu diễn dạng tín hiệu và chữ hoa (F) để biểu diễn biến đổi Laplace của nó. (mặc dù chúng ta đề cập ở đây là các tín hiệu thời gian, nhưng trong phương trình (23.156) không yêu cầu $f(t)$ là một hàm của thời gian. Hàm truyền có thể được dùng đối với các hàm của đại lượng t nào đó.)

Dùng biến đổi Laplace biến đổi tín hiệu của hệ thống tuyến tính bất biến theo thời gian. Ứng xử của một hệ như vậy tại $t \geq 0$ chỉ phụ thuộc vào tín hiệu vào tại $t \geq 0$ vào trạng thái trước của biến đầu ra (ở $t = 0^-$). Vì vậy, không có vấn đề gì khi biến đổi Laplace bỏ qua $f(t)$ với $t < 0^-$.

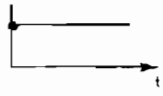
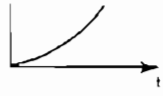
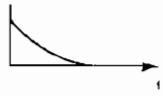
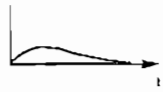
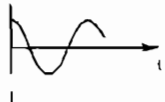
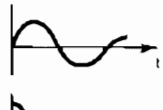
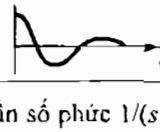
Quá trình tìm hàm thời gian $f(t)$ tương ứng với một biến đổi Laplace cụ thể $F(s)$ được gọi là *biến đổi Laplace ngược* và được ký hiệu là $\mathcal{L}^{-1}$. Chúng ta cũng gọi $f(t)$ là *biến đổi Laplace ngược* của $F(s)$. Vì biến đổi Laplace một vế bỏ qua $t < 0^-$, nên $F(s)$ không chứa thông tin về $f(t)$ tại $t < 0^-$. Vì vậy, biến đổi Laplace ngược không thể khôi phục $f(t)$ với $t < 0^-$. Chúng ta sẽ xét tất cả các tín hiệu như thể chúng chỉ được định nghĩa với $t \geq 0^-$. Vì vậy có quan hệ một - một giữa $f(t)$ và $F(s)$.

Để minh họa phép biến đổi Laplace, chúng ta tìm biến đổi Laplace của hàm mũ suy giảm $f(t) = e^{-\alpha t}$, $t \geq 0^-$. Biến đổi đó là

$$\begin{aligned} F(s) &= \int_{0^-}^{\infty} e^{-\alpha t} e^{-st}\,dt = \left.\frac{e^{-(s+\alpha)t}}{-(s+\alpha)}\right|_{0^-}^{\infty} \\ &= \left.\frac{e^{-(\sigma+\alpha)t} e^{-j\omega t}}{-(s+\alpha)}\right|_{0^-}^{\infty} = \frac{1}{s+\alpha} \qquad \mathrm{Re}[s] > -\alpha \end{aligned} \tag{23.157}$$

Chúng ta phải yêu cầu $\sigma > -\alpha$, trong đó σ là phần thực của s, để hệ số mũ thực hội tụ tới 0 ở giới hạn trên. (Độ lớn của hệ số mũ phức giữ nguyên là 1 với tất cả t). Vì vậy, biến đổi Laplace của hàm mũ suy giảm chỉ được xác định với $\mathrm{Re}[s] > -\alpha$. Giới hạn này trên miền F trong mặt phẳng phức s tương ứng với giới hạn $t \geq 0^-$ trên miền f.

BẢNG 23.11 Các cặp biến đổi Laplace

$f(t) = \mathcal{L}^{-1}[F(s)], t \geq 0^-$		$F(s) = \mathcal{L}[f(t)]$
$\delta(t)$		1
$u_s(t)$		$\frac{1}{s}$
$t^n, n = 1, 2, \ldots$		$\frac{n!}{s^{n+1}}$
$e^{-\alpha t}$		$\frac{1}{s+\alpha}$
$t^n e^{-\alpha t}, n = 1, 2, \ldots$		$\frac{n!}{(s+\alpha)^{n+1}}$
$\sin(\omega_0 t)$		$\frac{\omega_0}{s^2+\omega_0^2}$
$\cos(\omega_0 t)$		$\frac{s}{s^2+\omega_0^2}$
$e^{-\alpha t}\sin(\omega_d t)$		$\frac{\omega_d}{(s+\alpha)^2+\omega_d^2}$
$e^{-\alpha t}\cos(\omega_d t)$		$\frac{s+\alpha}{(s+\alpha)^2+\omega_d^2}$

Các đặc tính quan trọng của hàm tần số phức $1/(s + \alpha)$ là tồn tại một điểm cực và vị trí của cực đó, $s = -\alpha$ [rad/s]. (Cực này xác định giới hạn trái của miền mặt phẳng phức s mà ở đó hàm

truyền $1/(s + \alpha)$ xác định). Các đặc tính quan trọng của hàm thời gian tương ứng là tính suy giảm và tốc độ suy giảm với luỹ thừa $-\alpha$ [rad/s]. Có sự tương đương rõ ràng giữa các đặc tính của $f(t)$ và $F(s)$. Chúng ta nên nghĩ rằng toàn bộ hàm phức F tương ứng với toàn bộ tín hiệu thời gian f.

Như ví dụ biến đổi thứ hai, đặt $f(t) = \delta(t)$, xung đơn vị với khoảng thời gian rất ngắn. Nó tác động tại thời điểm $t = 0$. Nó có giá trị 0 ở $t = 0^-$. (Bởi vì chúng ta dùng 0^- là giới hạn dưới của tích phân xác định, nên nó không ảnh hưởng gì khi mà xung chưa kích ở t = 0 hoặc bắt đầu tăng ở t = 0). Xung chỉ khác 0 với $t \approx 0$ mà ở đó $e^{-st} \approx 1$. Vì vậy biến đổi Laplace là:

$$\Delta(s) = \int_{0^-}^{\infty} \delta(t)e^{-st}\,dt \approx \int_{0^-}^{\infty} \delta(t)(1)dt = 1 \tag{23.158}$$

Không cần phải dẫn ra biến đổi Laplace cho mỗi tín hiệu mà chúng ta sử dụng khi nghiên cứu hệ thống. Bảng 23.11 đưa ra các biến đổi cho một vài dạng tín hiệu thường gặp ở các hệ động lực.

Các thuộc tính của phép biến đổi

Một số thuộc tính có ích của biến đổi Laplace được tóm tắt trong bảng 23.12.

BẢNG 23.12 Các tính chất của biến đổi Laplace [4]

1. Khuyếch đại	$\mathcal{L}[af(t)] = aF(s)$
2. Cộng	$\mathcal{L}[f_1(t) + f_2(t)] = F_1(s) + F_2(s)$
3. Đạo hàm	$\mathcal{L}[\dot{f}(t)] = sF(s) - f(0^-)$
4. Các đạo hàm	$\mathcal{L}[\ddot{f}(t)] = s^2F(s) - sf(0^-) - \dot{f}(0^-)$
5. Tích phân	$\mathcal{L}\left[\int_{0^-}^{t} f(t)dt\right] = \dfrac{F(s)}{s}$
6. Tích chập	$\mathcal{L}\left[\int_{0^-}^{t} f_1(\lambda)f_2(t-\lambda)d\lambda\right] = F_1(s)F_2(s)$
7. Giá trị đầu	$f(0^+) = \lim_{t\to 0^+} f(t) = \lim_{s\to\infty} sF(s)$
8. Giá trị cuối	$f(\infty) = \lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$
9. Tích phân xác định	$\int_0^{\infty} f(t)dt = \lim_{s\to 0} sF(s)$
10. Suy giảm theo hàm mũ	$\mathcal{L}[e^{-\alpha t} f(t)] = F(s+\alpha)$
11. Suy giảm	$\mathcal{L}[f(t-t_0)u_s(t-t_0)] = e^{-t_0 s}F(s) \qquad t_0 \geq 0$
12. Nhân thời gian	$\mathcal{L}[tf(t)] = -\dfrac{dF(s)}{ds}$
13. Chia thời gian	$\mathcal{L}\left[\dfrac{f(t)}{t}\right] = \int_s^{\infty} F(s)ds$
14. Cân chỉnh thời gian	$\mathcal{L}[f(at)] = \dfrac{F(s/a)}{a}$

Theo tính chất đạo hàm, số nhân s tác động hoàn toàn giống toán tử đạo hàm thời gian, nhưng trong miền tín hiệu được biến đổi Laplace. Khi chúng ta biến đổi Laplace phương trình cho một phần tử lưu giữ năng lượng như là khối vật hoặc lò xo, thì tính chất đạo hàm kết hợp với trạng thái năng lượng trước đó của phần tử, là giá trị của biến tại $t = 0^-$.

Khi chúng ta biến đổi Laplace phương trình của hệ thống vào - ra của một biến riêng biệt thì tính chất đạo hàm sẽ liên kết với toàn bộ trạng thái trước đó của hệ. Vì vậy chúng ta có thể tìm thấy lời giải cho phương trình của hệ thống mà không cần xác định các điều kiện đầu (ở $t = 0^-$) - nó làm cho quá trình giải đơn giản đáng kể. Vì $F(s)$ chứa tất cả các thông tin về $f(t)$ với $t \geq 0^-$ nên có thể tìm thấy một số tính chất của tín hiệu $f(t)$ từ hàm truyền $F(s)$ mà không cần thực hiện phép biến đổi Laplace ngược. Các tính chất 7 – 9 của bảng 23.12 đưa ra ba trong số các tính chất đó, gọi là giá trị đầu ($t \to 0^+$), giá trị cuối ($t \to \infty$) và tích phân xác định. Các tính chất còn lại trong bảng chỉ ra các tác động lên hàm truyền khi có những biến đổi khác nhau ở các tín hiệu dạng sóng. Phương pháp thường dùng để tìm thấy các biến đổi ngược là sử dụng một bảng các cặp biến đổi. Bảng đó phải được lưu giữ trong các phần mềm như CC, MATLAB, MAPLE, v.v... Bảng 23.11 chứng minh rằng các hàm truyền của các tín hiệu điển hình của hệ thống tỉ lệ với các đa thức của s. Phân số của các đa thức có thể được tách thành tổng các phân số của đa thức *đơn giản* - quá trình này được gọi là *khai triển từng phần phân số*. Vì vậy quá trình biến đổi ngược có thể được hoàn thành bởi một chương trình máy tính kết hợp với một bảng tóm tắt các biến đổi.

Biến đổi và nghiệm phương trình hệ thống

Giả sử rằng một nguồn ngoài cấp một mẫu vận tốc riêng $v_1(t)$ tới nút 1 của hình 23.25. Để nhận được phương trình vào - ra của hệ thống liên hệ vận tốc v_2 của nút 2 với tín hiệu vào v_1, nhân phương trình (23.153) với p và thay v_1 cho px_1 và v_2 cho px_2. Kết quả là:

$$(mp^2 + bp + k)v_2 = (bp + k)v_1 \qquad (23.159)$$

Hai vế của phương trình (23.159) là các hàm đồng nhất của thời gian. Vì vậy, biến đổi Laplace của hai vế của phương trình (23.159) là bằng nhau. Vì biến đổi Laplace là tuyến tính (tính chất 1 và 2 của bảng 23.12) và vì các hệ số của phương trình vi phân là hằng số nên biến đổi Laplace có thể được áp dụng một cách riêng biệt cho các số hạng riêng của mỗi vế:

$$m[s^2V_2(s) - sv_2(0^-) - \dot{v}_2(0^-)] + b[sV_2(s) - v_2(0^-)] + kV_2(s) = b[sV_1(s) - v_1(0^-)] + kV_1(s) \quad (23.160)$$

trong đó các tính chất đạo hàm của biến đổi Laplace (tính chất 3 và 4 của bảng 23.12) đưa ra các giá trị đầu $v_1(0^-)$, $v_2(0^-)$ và $\dot{v}_2(0^-)$ của phương trình. Theo phương trình (23.160), để xác định đầy đủ biến đổi $V_2(s)$ của ứng xử $v_2(t)$, chúng ta phải xác định các giá trị đầu và $V_1(s)$. có thể thấy rằng việc xác định ba giá trị đầu là tương đương với việc chỉ ra các trạng thái năng lượng của lò xo và khối vật.

Giả thiết rằng nguồn độc lập cấp một vận tốc không đổi $v_1(t) = vc$ bắt đầu ở $t = 0$. Hàm truyền tương ứng, theo mục 2 của bảng 23.11 và tính chất 1 của bảng 23.12, là $V_1(s) = v_c / s$. Thay biến đổi $V_1(s)$ vào phương trình (23.160) và giải:

$$V_2(s) = \frac{(bs + k)v_c + ms\dot{v}_2(0^-) + bs[v_2(0^-) - v_1(0^-)] + ms^2v_2(0^-)}{s(ms^2 + bs + k)} \qquad (23.161)$$

Chúng ta có thể tìm được dạng tín hiệu ra $v_2(t)$ là một hàm của các tham số mô hình m, k, b, tham số của tín hiệu nguồn v_c và thông tin của các trạng thái ban đầu $v_1(0^-)$, $v_2(0^-)$ và $\dot{v}_2(0^-)$, nhưng cách biểu diễn nghiệm có thể lộn xộn. Thay vì vậy, chúng ta thực hiện quá trình giải cho các số cụ thể m = 2 kg, b = 4 N · s/m, k = 10 N/m, $\dot{v}_2(0^-) = 0\,m/s^2$, $v_1(0^-) = 0\,m/s$, $v_2(0^-) = -1\,m/s$ và v_c = 1 m/s. Khai triển từng phần phân số của biến đổi thuận và ngược, có thể tìm được bởi một chương trình máy tính, là:

$$Y_2(s) = \frac{1}{s} - \frac{2s+2}{(s+1)^2 + 2^2} \tag{23.162}$$

$$y_2(t) = 1 - 2e^{-t}\cos(2t), \qquad t \geq 0 \tag{23.163}$$

Chúng ta có thể thực hiện biến đổi Laplace của các phương trình hệ thống ở bất kỳ giai đoạn nào. Thậm chí, chúng ta có thể viết các phương trình trực tiếp theo các biến đã được biến đổi nếu chúng ta muốn. Ví dụ, toán tử $G(p)$ trong phương trình (23.154) biểu diễn một tỉ số của các đa thức của toán tử đạo hàm thời gian p. Vì vậy, biến đổi Laplace phương trình vi phân (23.154) sẽ đưa ra các giá trị đầu của các đạo hàm y_1 và y_2. Nếu các giá trị đầu của tất cả các đạo hàm đó là không thì phương trình biến đổi Laplace là:

$$Y_2(s) = G(s)Y_1(s) \tag{23.164}$$

toán tử p ở phương trình (23.154) được thay bởi biến tần số phức s ở phương trình (23.164). Vì vậy, sẽ là thích hợp để xác định hàm truyền trực tiếp theo các tín hiệu đã được biến đổi Laplace:

$$G(s) = \left.\frac{Y_2(s)}{Y_1(s)}\right|_{PV=0} \tag{23.165}$$

trong đó $Y_1(s)$ và $Y_2(s)$ là các biến đổi Laplace của các tín hiệu $y_1(t)$ và $y_2(t)$ và kí hiệu PV = 0 có nghĩa rằng các giá trị đầu (ở $t = 0$) của $y_1(t)$ và $y_2(t)$ và các đạo hàm đã đề cập ở trên có liên kết với phương trình (23.164) được đặt bằng không. Khái niệm *miền tần số*, phương trình (23.165) là tương đương với khái niệm *miền thời gian*, phương trình (23.155).

Giả sử rằng tín hiệu vào $y_1(t)$ là xung đơn vị $\delta(t)$. Thì tín hiệu đáp ứng $y_2(t)$ là đáp ứng xung đơn vị của hệ thống. Vì biến đổi Laplace của xung đơn vị là $Y_1(s) = \Delta(s) = 1$, mục 1 của bảng 23.11, phương trình (23.164) chỉ ra rằng biến đổi Laplace $Y_2(s)$ của đáp ứng xung đơn vị là đồng nhất với hàm truyền của trạng thái không. Hàm truyền của một hệ tuyến tính có hai cách thể hiện. Cả hai cách thể hiện đều mô tả hệ thống. Trong miền tần số, hàm truyền $G(s)$ là số nhân để đưa ra đáp ứng bằng việc nhân với hàm truyền của nguồn tín hiệu như trong phương trình (23.164). Trong miền thời gian, chúng ta dùng một tín hiệu đáp ứng điển hình - đáp ứng xung - để mô tả hệ thống. Hàm truyền $G(s)$ là biến đổi Laplace của đáp ứng đặc trưng đó.

Định nghĩa các khái niệm

- **Tín hiệu vào (Input)**: Biến độc lập.
- **Phương trình hệ thống vào ra (Input–output system equation)**: Một phương trình vi phân mô tả ứng xử của một biến phụ thuộc như là một hàm thời gian. Biến phụ thuộc được xem là đầu ra của hệ thống. Biến (các biến) độc lập là các đầu vào.
- **Tín hiệu ra (Output):** Một biến phụ thuộc.
- **Tín hiệu (Signal)**: Một biến có thể quan sát; Một đại lượng cho biết ứng xử của một hệ.
- **Trạng thái (State)**: Trạng thái của một hệ tuyến tính bậc n tương ứng với các giá trị của một biến phụ thuộc và n-1 đạo hàm theo thời gian đầu tiên của nó.
- **Bất biến theo thời gian (Time invariant)**: Một hệ có thể được biểu diễn bằng các phương trình vi phân với các hệ số hằng.
- **Trạng thái không (Zero state)**: là một trạng thái mà khi đó không có năng lượng được lưu giữ hoặc khi tất cả các biến có giá trị bằng không.

Tài liệu tham khảo

[1] Franklin, G. F., Powell, J. D., and Emami-Naeini, A. 1994. *Feedback Control of Dynamic Systems*, 3rd ed., Addison Wesley, Reading, MA.

[2] Kuo, B. C. 1991. *Automatic Control Systems*, 6th ed., Prentice-Hall, Englewood Cliffs, NJ.

[3] Nise, N.S 1992. *Control Systems Engineering*, Benjamin Cumming, Redwood City, CA.

24

Phân tích không gian trạng thái và các thuộc tính hệ thống

24.1 Mô hình: Những khái niệm cơ bản 24-1
24.2 Biến trạng thái: Những khái niệm cơ bản 24-2
24.3 Mô tả không gian trạng thái cho các hệ thời gian liên tục 24-4
24.4 Mô tả không gian trạng thái cho hệ thống thời gian rời rạc và dữ liệu lấy mẫu 24-13
24.5 Các mô hình không gian trạng thái cho các hệ kết nối 24-23
24.6 Các thuộc tính của hệ thống 24-25
24.7 Bộ quan sát trạng thái 24-39
24.8 Phản hồi trạng thái 24-44
24.9 Phản hồi trạng thái quan sát được 24-46

Mario E. Salgado
Universidad Técnica Federico Santa María

Juan I. Yuz
Universidad Técnica Federico Santa María

24.1 Mô hình: Những khái niệm cơ bản

Sự hiểu biết về một hệ thống của một kỹ sư hoặc một nhà khoa học thể hiện ở cách mô tả hệ thống đó để có thể xác định được ứng xử của nó.

Bất cứ sự mô tả nào hỗ trợ sự hiểu biết đó đều có thể gọi là một **mô hình**. Trong lý thuyết hệ thống, mô hình đóng vai trò cơ bản vì chúng cần thiết cho việc phân tích, tổng hợp và thiết kế tất cả các loại hệ thống có thể tưởng tượng được.

Không chỉ có một mô hình duy nhất cho một hệ thống cho trước. Thứ nhất, vì mô hình có thể tuân theo nhiều mục đích khác nhau. Thí dụ, khi xử lý với động cơ điện, chúng ta có thể quan tâm đến quá trình biến đổi năng lượng cơ-điện, hoặc có thể mô hình hoá động cơ giống như một hệ thống nhiệt hay hệ thống cơ để nghiên cứu sự rung động hoặc sức bền vật liệu vv... Thứ hai, vì trên thực tế các mô hình đó thường không chính xác vì các hệ thống thực thường vô cùng phức tạp. Một trong những quyết định then chốt đối với các kỹ sư khi thực hiện mô hình hoá hệ thống là phải quyết định đặc tính thiết yếu nào mà mô hình phải thể hiện và quyết định đó cũng liên quan mật thiết với mục đích của mô hình.

Lý thuyết này hỗ trợ việc mô hình hoá là một lĩnh vực rộng lớn, ở đó những học thuyết đầu tiên, lý thuyết tín hiệu, toán học và công cụ số hoá được kết hợp theo nhiều cách khác nhau để tạo ra các phương pháp luận phong phú. Một mô hình hiếm khi được xây dựng chỉ theo một hướng, quá trình xây dựng mô hình thường xuyên tương tác qua lại với nhau và các tiến trình của nó phụ

thuộc vào chất lượng kết quả thu được khi sử dụng mô hình trong một ứng dụng riêng rẽ. Các tương tác cũng có thể bao gồm việc thay đổi trong phương pháp luận của việc mô hình hoá.

Trong chương này, chúng ta dựa trên một lớp các mô hình để mô tả các *hệ thống động*. Hệ thống động là các hệ thống mà ở đó các biến hệ thống đều phụ thuộc lẫn nhau không chỉ theo toán học mà còn theo cách mà ở đó chúng ta quan sát sự can thiệp của các hiệu ứng tích luỹ và tốc độ thay đổi. Các mô hình cho hệ thống động có thể được xây dựng trong miền thời gian liên tục, rời rạc hoặc trong cấu trúc thời gian liên tục-rời rạc (cho các hệ thống hỗn hợp, các hệ thống lấy mẫu). Chúng ta sẽ đề cập đến cả ba trường hợp đó.

Ở đây chúng ta sẽ nhấn mạnh vào các khái niệm, những thuộc tính cơ bản, các giải thích vật lý và các ví dụ. Chúng ta cũng sẽ không đi sâu vào việc chứng minh và các lý thuyết phức tạp. Bạn đọc có thể tìm hiểu sâu hơn về lý thuyết thông qua các tài liệu [6, 8, 10-14].

24.2 Biến trạng thái: Những khái niệm cơ bản

Giới thiệu

Một trong những lớp mô hình được sử dụng thường xuyên nhất là những mô hình được xác định bởi các phương trình dựa trên một nhóm các biến nội tại của hệ thống. Những biến nội tại đó được hiểu là các **biến trạng thái**. Giá trị của chúng tại những khoảng thời gian tức thời nhất định cho biết **trạng thái hệ thống**, mặc dù chúng ta thường diễn đạt *biến hệ thống* và *trạng thái hệ thống* theo nghĩa giống nhau.

Định nghĩa trên trở nên không rõ ràng khi nó được đặt vào một tập biến hệ thống cụ thể. Đặc điểm nổi bật của một tập biến trạng thái được làm rõ trong định nghĩa sau.

Một tập biến trạng thái của hệ thống cho trước là một tập biến nội tại, do đó bất kỳ một biến hệ thống nào cũng có thể được tính toán như một hàm của trạng thái hiện tại và các đầu vào hiện tại và tương lai của hệ thống.

Trong định nghĩa này, chúng tôi muốn nhấn mạnh ý nghĩa vật lý của các biến trạng thái. Tuy nhiên, cũng có thể có thêm những định nghĩa trừu tượng khác. Định nghĩa này cho biết nếu chúng ta biết trạng thái tại thời gian t chúng ta có thể tính toán được năng lượng chứa trong hệ thống tại thời điểm đó. Năng lượng chứa trong hệ thống phụ thuộc vào một số biến hệ thống (tốc độ, điện áp, dòng điện, vị trí, nhiệt độ, áp suất...) và tất cả chúng, theo định nghĩa, đều có thể được tính toán từ trạng thái hệ thống.

Định nghĩa trên cho rằng có thể nghĩ về trạng thái theo một cách tổng quát hơn: các biến trạng thái có thể được chọn như một **hàm** (ví dụ một tổ hợp tuyến tính) của các biến nội tại của hệ thống. Sự tổng quát hoá này sẽ tạo nên một số khoảng cách giữa trạng thái và giải thích vật lý của nó. Tuy nhiên, cũng có thuận lợi khi tạo ra cấu trúc tổng quát hơn đó. Nó cũng làm rõ thêm: **sự lựa chọn các biến trạng thái là không duy nhất.**

Một quan sát quan trọng khác là thời gian tiến triển của trạng thái, bản thân biểu đồ trạng thái có thể được tính toán từ giá trị hiện tại của trạng thái và các đầu vào hiện tại và tương lai. Vì vậy các mô hình liên quan đều là phương trình vi phân bậc một (thời gian liên tục) hoặc phương trình sai phân một bước (thời gian rời rạc).

Các mô hình không gian trạng thái cơ bản

Nếu chúng ta biểu diễn **x** là vectơ tương ứng với một biến trạng thái riêng, dạng tổng quát của mô hình biến trạng thái là như sau:

Đối với hệ thống thời gian liên tục

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{24.1}$$

$$\mathbf{y}(t) = \mathbf{G}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{24.2}$$

trong đó $\mathbf{u}(t)$ là **vecto đầu vào** và $\mathbf{y}(t)$ là **vecto đầu ra** của hệ thống.

Đối với các hệ thống thời gian rời rạc

$$\mathbf{x}[t+1] = \mathbf{F}_d(\mathbf{x}[t], \mathbf{u}[t], t) \tag{24.3}$$

$$\mathbf{y}[t] = \mathbf{G}_d(\mathbf{x}[t], \mathbf{u}[t], t) \tag{24.4}$$

Tương tự như trường hợp thời gian liên tục, $\mathbf{u}[t]$ là **vecto đầu vào** và $\mathbf{y}[t]$ là **vecto đầu ra** của hệ thống.

Chú ý rằng, trong suốt chương này chúng ta sẽ sử dụng ký hiệu t để biểu diễn thời gian liên tục và rời rạc nhưng khác nhau ở chỗ sử dụng các ký hiệu [và] đối với thời gian rời rạc.

Để có được tiếp cận ban đầu về khái niệm không gian trạng thái, ta xem ví dụ sau.

Ví dụ 24.1

Trong hình 24.1, một ngoại lực $f(t)$ được đặt vào hệ khối lượng-lò xo. Vị trí $d(t)$ được đo tương quan với vị trí khối lượng khi lò xo ở trạng thái tự do không có ngoại lực tác động. Sự dịch chuyển của khối lượng bị cản bởi lực ma sát trượt tỉ lệ với tốc độ của vật nặng $v(t)$.

Chúng ta đã biết rằng tính toán vị trí và tốc độ vật nặng chúng ta phải biết tốc độ ban đầu của vật nặng và lực kéo ban đầu của lò xo. Vì vậy vectơ trạng thái phải có hai thành phần, ví dụ $x(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^T$, và lựa chọn trạng thái ban đầu là:

$$x_1(t) = d(t) \tag{24.5}$$

$$x_2(t) = v(t) = x_1(t) \tag{24.6}$$

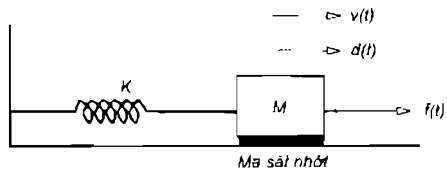
Với lựa chọn này áp dụng định luật Newton nhận được:

$$f(t) = M\frac{dv(t)}{dt} + Kd(t) + Dv(t) = M\dot{x}_2(t) + Kx_1(t) + Dx_2(t) \tag{24.7}$$

trong đó D là hằng số tỉ lệ ma sát nhớt. Chúng ta có phương trình trạng thái như sau:

$$\dot{x}_1(t) = x_2(t) \tag{24.8}$$

$$\dot{x}_2(t) = -\frac{K}{M}x_1(t) - \frac{D}{M}x_2(t) + \frac{1}{M}f(t) \tag{24.9}$$



HÌNH 24.1 Một hệ cơ

Ta cũng thấy năng lượng $w(t)$ được lưu giữ trong hệ thống được đưa ra bởi:

$$w(t) = \frac{1}{2}K(d(t))^2 + \frac{1}{2}M(v(t))^2 = x(t)^t \, \Lambda x(t) \tag{24.10}$$

trong đó Λ là ma trận đường chéo: $\Lambda = \text{diag}\left[\frac{K}{2}, \frac{M}{2}\right]$.

Cuối cùng, tính không duy nhất của vectơ trạng thái có thể được đánh giá nếu thay vì các lựa chọn được tạo ra theo (24.8) ta chọn trạng thái mới $\bar{\mathbf{x}}(t)$ liên quan với $\mathbf{x}(t)$ bởi một ma trận không suy biến:

$$\bar{\mathbf{x}}(t) = \mathbf{T}\mathbf{x}(t) \tag{24.11}$$

Vấn đề này sẽ được đề cập chi tiết ở phần "Biến đổi trạng thái tương đương".

Tín hiệu và mô tả không gian trạng thái

Không gian trạng thái cũng có thể được sử dụng để mô tả nhiều loại tín hiệu bằng cách sử dụng mô hình như sau:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t), \qquad y(t)=\mathbf{C}\mathbf{x}(t) \text{ đối với các tín hiệu thời gian liên tục} \tag{24.12}$$

$$\mathbf{x}[t+1]=\mathbf{A}_q\mathbf{x}[t], \qquad y[t]=\mathbf{C}_q\mathbf{x}[t] \text{ đối với các tín hiệu thời gian rời rạc} \tag{24.13}$$

Để minh hoạ ý tưởng này, chúng ta xét tín hiệu thời gian liên tục có dạng:

$$f(t) = 2 + 4\cos(5t) - \sin(5t) \tag{24.14}$$

Tín hiệu này có thể được giải thích như là nghiệm của phương trình vi phân thuần nhất:

$$\frac{d^3 f(t)}{dt^3} + 25\frac{df(t)}{dt} = 0 \text{ , giả thiết: } f(0) = 6,\ f'(0) = -5 \text{ và } f''(0) = -100 \tag{24.15}$$

Nếu ta chọn như biến trạng thái, $x_1(t) = f(t)$, $x_2(t) = f'(t)$ và $x_3(t) = f''(t)$, thì mô hình không gian trạng thái cho tín hiệu này là:

$$\frac{d\mathbf{x}(t)}{dt} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -25 & 0 \end{pmatrix} = \mathbf{x}(t), \quad y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\mathbf{x}(t) \tag{24.16}$$

Theo cách dùng này của mô hình không gian trạng thái, các biến trạng thái không có ý nghĩa vật lý cụ thể. Tuy nhiên, mô tả này đặc biệt có ích trong lý thuyết phục hồi tín hiệu và xử lý nhiễu khi tổng hợp hệ thống điều khiển.

24.3 Mô tả không gian trạng thái cho các hệ thống thời gian liên tục

Phần này giới thiệu mô tả không gian trạng thái cho hệ thống thời gian liên tục. Phân tích tập trung vào lớp các hệ thống **tuyến tính và bất biến** ; trước tiên ta sẽ xem xét phương pháp để xây dựng một mô hình tuyến tính từ các phương trình phi tuyến (24.1) và (24.2).

Ở giai đoạn này, có một giới hạn nữa là các hệ thống xem xét không có thời gian trễ thuần. Đặc tính này sẽ sinh ra một vectơ trạng thái vô hạn chiều. Tuy nhiên chúng ta sẽ thấy trong mục 24.4 rằng lớp hệ thống này có thể được xử lý bởi mô hình dữ liệu trích mẫu.

Tuyến tính hoá

Vì chúng ta sẽ tập trung vào các hệ thống thời gian bất biến, nên (24.1) và (24.2) có thể viết lại như sau:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t)) \tag{24.17}$$

$$\mathbf{y}(t) = \mathbf{G}(\mathbf{x}(t), \mathbf{u}(t) \tag{24.17}$$

Giả thiết rằng mô hình (24.17) và (24.18) có ít nhất một điểm cân bằng cho bởi $\{\mathbf{x}_Q, \mathbf{u}_Q, \mathbf{y}_Q\}$. Ba vectơ này thoả mãn:

$$\mathbf{0} = \mathbf{F}(\mathbf{x}_Q, \mathbf{u}_Q) \tag{24.19}$$

$$\mathbf{y}_Q = \mathbf{G}(\mathbf{x}_Q, \mathbf{u}_Q) \tag{24.20}$$

Chú ý rằng điểm cân bằng này được xác định bởi đạo hàm trạng thái bằng không.

Nếu ta xem xét vùng lân cận xung quanh điểm cân bằng, sau đó ta có thể tính xấp xỉ mô hình (24.17) và (24.18) bằng cách sử dụng chuỗi Taylor có dạng sau:

$$\dot{\mathbf{x}} \approx \mathbf{F}(\mathbf{x}_Q, \mathbf{u}_Q) + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\substack{x=x_Q \\ u=u_Q}} (\mathbf{x}(t) - \mathbf{x}_Q) + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\right|_{\substack{x=x_Q \\ u=u_Q}} (\mathbf{u}(t) - \mathbf{u}_Q) \tag{24.21}$$

$$\dot{\mathbf{y}}(t) \approx \mathbf{G}(\mathbf{x}_Q, \mathbf{u}_Q) + \left.\frac{\partial \mathbf{G}}{\partial \mathbf{x}}\right|_{\substack{x=x_Q \\ u=u_Q}} (\mathbf{x}(t) - \mathbf{x}_Q) + \left.\frac{\partial \mathbf{G}}{\partial \mathbf{u}}\right|_{\substack{x=x_Q \\ u=u_Q}} (\mathbf{u}(t) - \mathbf{u}_Q) \tag{24.22}$$

Phương trình (24.21) và (24.22) có thể viết như sau:

$$\frac{d\Delta\mathbf{x}(t)}{dt} = \mathbf{A}\Delta\mathbf{x}(t) + \mathbf{B}\Delta\mathbf{u}(t) \tag{24.23}$$

$$\Delta\mathbf{y}(t) = \mathbf{C}\Delta\mathbf{x}(t) + \mathbf{D}\Delta\mathbf{u}(t) \tag{24.24}$$

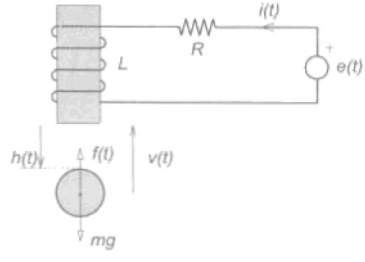
trong đó:

$$\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_Q \qquad \mathbf{Du}(t) = \mathbf{u}(t) - \mathbf{u}_Q \qquad \Delta\mathbf{y}(t) = \mathbf{y}(t) - \mathbf{y}_Q \tag{24.25}$$

và:

$$\mathbf{A} = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\substack{x=x_Q \\ u=u_Q}}, \qquad \mathbf{B} = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\right|_{\substack{x=x_Q \\ u=u_Q}}, \qquad \mathbf{C} = \left.\frac{\partial \mathbf{G}}{\partial \mathbf{x}}\right|_{\substack{x=x_Q \\ u=u_Q}}, \qquad \mathbf{D} = \left.\frac{\partial \mathbf{G}}{\partial \mathbf{u}}\right|_{\substack{x=x_Q \\ u=u_Q}} \tag{24.26}$$

Ví dụ sau đây sẽ minh họa cho việc tuyến tính hóa.



HÌNH 24.2 Hệ thống hút từ tính

Ví dụ 24.2

Xét hệ thống hút từ tính trong hình 24.2.

Quả cầu sắt chịu tác động bởi hai lực: sức nặng của bản thân vật, mg, và lực hút điện từ, $f(t)$. Điện từ được điều khiển bởi nguồn điện áp $e(t) > 0 \ \forall t$.

Lực hấp dẫn tác động lên khối cầu, $f(t)$, phụ thuộc vào khoảng cách $h(t)$ và dòng điện $i(t)$. Mối quan hệ này có thể mô tả gần đúng như sau:

$$f(t) = \frac{K_1}{h(t) + K_2} i(t) \tag{24.27}$$

trong đó K_1 và K_2 là hằng số dương.

Sử dụng quy tắc thứ nhất ta có:

$$e(t) = Ri(t) + \frac{Ldi(t)}{dt} \tag{24.28}$$

$$v(t) = -\frac{dh(t)}{dt} \tag{24.29}$$

$$f(t) = \frac{K_1}{h(t) + K_2} i(t) = mg + m\frac{dv(t)}{dt} \tag{24.30}$$

Tiếp theo ta chọn các đại lượng sau là các biến trạng thái: dòng điện $i(t)$, vị trí khối cầu $h(t)$, và tốc độ khối cầu $v(t)$:

$$x(t) = \left[x_1(t) \quad x_2(t) \quad x_3(t)\right]^T = [\,\mathrm{i(t)} \quad \mathrm{h(t)} \quad \mathrm{v(t)}\,]^T \tag{24.31}$$

Sau đó từ (24.28) - (24.30) ta có thể đặt mô tả hệ thống như trong (24.1):

$$\frac{di(t)}{dt} = \frac{dx_1(t)}{dt} = -\frac{R}{L}x_1(t) + \frac{1}{L}e(t) \tag{24.32}$$

$$\frac{dh(t)}{dt} = \frac{dx_2(t)}{dt} = -x_3(t) \tag{24.33}$$

$$\frac{dv(t)}{dt} = \frac{dx_3(t)}{dt} = \frac{K_1}{m(x_2(t) + K_2)}x_1(t) - g \tag{24.34}$$

Trước khi xây dựng mô hình tuyến tính, điểm cân bằng cần được tính toán. Đầu vào điều khiển của hệ thống này là nguồn áp $e(t)$. Điểm cân bằng nhận được khi $e(t) = E_Q$, vì vậy trạng thái cân bằng có thể tính toán nhờ (24.32) - (24.34), bằng cách đặt tất cả các phương trình đạo hàm bằng không, như sau

$$-\frac{R}{L}x_{1Q} + \frac{1}{L}E_Q = 0 \Rightarrow x_{1Q} = \frac{E_Q}{R} \tag{24.35}$$

$$-x_{3Q} = 0 \Rightarrow x_{3Q} = 0 \tag{24.36}$$

$$\frac{K_1}{m(x_{2Q} + K_2)}x_{1Q} - g = 0 \Rightarrow x_{2Q} = \frac{K_1}{mg}x_{1Q} - K_2 = \frac{K_1 E_Q}{mgR} - K_2 \tag{24.37}$$

Các thiết lập này đủ để xây dựng mô hình tuyến tính theo sự gia tăng của đầu vào $\Delta e(t)$ và gia tăng trạng thái $\Delta \mathbf{x}(t) = [\Delta x_1(t) \;\; \Delta x_2(t) \;\; \Delta x_3(t)]^T$. Cho kết quả như sau:

$$\frac{d\Delta x_1(t)}{dt} = -\frac{R}{L}\Delta x_1(t) + \frac{1}{L}\Delta e(t) \tag{24.38}$$

$$\frac{d\Delta x_2(t)}{dt} = -\Delta x_3(t) \tag{24.39}$$

$$\frac{d\Delta x_3(t)}{dt} = \frac{Rg}{E_Q}\Delta x_1(t) - \frac{Rmg^2}{K_1 E_Q}\Delta x_2(t) \tag{24.40}$$

Nếu ta định nghĩa vị trí quả cầu $h(t)$ là đầu ra của hệ thống, ta có thể so sánh các phương trình trên với (24.23) và (24.24) để có:

$$A=\begin{pmatrix} -\frac{R}{L} & 0 & 0 \\ 0 & 0 & -1 \\ \frac{R_g}{E_Q} & -\frac{Rmg^2}{K_1 E_Q} & 0 \end{pmatrix}, \qquad B=\begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix}, \qquad C=\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad D=0 \tag{24.41}$$

Trong kết quả trên chúng tôi bỏ qua kí hiệu Δ đằng trước, nhưng người đọc cần nhớ rằng mô hình trên đây là tuyến tính theo **sự gia tăng** của trạng thái, các đầu vào và đầu ra xung quanh điểm cân bằng đã chọn.

Các mô hình không gian trạng thái tuyến tính

Chúng ta bắt đầu từ mô hình không gian trạng thái tuyến tính bất biến:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t) \tag{24.42}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) - \mathbf{D}\mathbf{u}(t) \tag{24.43}$$

Với giả thiết $\mathbf{x}(t_0) = \mathbf{x}_0$, nghiệm của phương trình (24.42) là:

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \quad \forall t \geq t_0 \tag{24.44}$$

trong đó **ma trận chuyển tiếp** $e^{\mathbf{A}t}$ thỏa mãn:

$$e^{\mathbf{A}t} = \mathbf{I} + \sum_{k=1}^{\infty} \frac{1}{k!}\mathbf{A}^k t^k \tag{24.45}$$

Bạn đọc có thể chứng minh lại (24.44) thỏa mãn (24.43) bằng cách dùng quy tắc Leibniz cho đạo hàm của tích phân.

Với kết quả trên, lời giải cho (24.43) được cho bởi:

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \mathbf{C}\int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau + \mathbf{D}\mathbf{u}(t) \tag{24.46}$$

Động lực học hệ thống

Trạng thái của hệ thống gồm hai thành phần: thành phần **không cưỡng bức**, $\mathbf{x}_u(t)$, và thành phần **cưỡng bức** $\mathbf{x}_f(t)$, trong đó:

$$\mathbf{x}_u(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 \tag{24.47}$$

$$\mathbf{x}_f(t) = \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \tag{24.48}$$

Để hiểu rõ mô hình không gian trạng thái và lời giải của nó, ta xét trường hợp $t_0 = 0$ và $u(t) = 0$ $\forall t \geq 0$, nghĩa là trạng thái chỉ có thành phần không cưỡng bức:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 \tag{24.49}$$

Giả thiết rằng $A \in \mathcal{R}^n$ và để đơn giản, nó có các trị riêng λ_1, $\lambda_2 \ldots \lambda_n$ với n vectơ riêng (độc lập tuyến tính) $\mathbf{v}_1$, $\mathbf{v}_2 \ldots \mathbf{v}_n$. Khi đó luôn tồn tại một tập các hằng số α_1, $\alpha_2 \ldots \alpha_n$ sao cho:

$$\mathbf{x}_0 = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i, \quad \alpha_i \in \mathfrak{C} \tag{24.50}$$

Kết quả đã biết từ đại số tuyến tính cho thấy rằng các trị riêng của A^k là λ_1^k, $\lambda_2^k \ldots \lambda_n^k$ tương ứng với vectơ riêng tương ứng $\mathbf{v}_1$, $\mathbf{v}_2 \ldots \mathbf{v}_n$. Vì vậy:

$$\mathbf{x}(t) = e^{At}\mathbf{x}_0 = \mathbf{I} + \sum_{i=1}^{n} \alpha_i \sum_{k=1}^{\infty} \frac{1}{k!} \underbrace{\mathbf{A}^k \mathbf{v}_i t^k}_{\lambda_i^k \mathbf{v}_i} = \sum_{i=1}^{n} \alpha_i e^{\lambda_i t} \mathbf{v}_i \tag{24.51}$$

Phương trình này cho thấy thành phần không cưỡng bức của trạng thái là tổ hợp tuyến tính của các ***dạng tự nhiên*** $\{e^{\lambda_i t}\}$, mà mỗi dạng trong số chúng liên quan tới một trị riêng của **A**. Vì vậy, ma trận **A** xác định:

- cấu trúc của đáp ứng không cưỡng bức
- sự ổn định (hoặc những mặt khác) của hệ thống
- tốc độ đáp ứng

Khi ma trận **A** không có một tập n vectơ riêng độc lập, các dạng Jordan có thể được sử dụng (xem [9, 10]).

Cấu trúc của đáp ứng không cưỡng bức

Khi thiếu đầu vào, trạng thái là tổ hợp của các dạng tự nhiên phụ thuộc vào các lớp hàm đã được định nghĩa: tất cả được điều khiển bởi các số mũ với cả phần thực hoặc ảo. Vì vậy những mô hình đó bao gồm các hằng số, số mũ thực, các sóng sin thuần, các sóng sin bị điều biến theo hàm mũ, và một số các hàm đặc biệt khác được nội suy từ các trị riêng lặp.

Để minh họa lập luận trên và mối quan hệ vật lý của chúng, xét hệ thống trong ví dụ 24.1. Với hệ này:

$$A = \begin{bmatrix} 0 & 1 \\ -\dfrac{K}{M} & -\dfrac{D}{M} \end{bmatrix} \tag{24.52}$$

Vì vậy, các trị riêng của hệ thống là các nghiệm của phương trình:

$$\det(\lambda \mathbf{I} - \mathbf{A}) = \lambda^2 + \frac{D}{M}\lambda + \frac{K}{M} = 0 \tag{24.53}$$

có nghĩa là:

$$\lambda_{1,2} = -\frac{D}{2M} \pm \sqrt{\frac{D^2}{4M^2} - \frac{K}{M}} \tag{24.54}$$

Do đó, khi suy giảm bằng không ($D = 0$), các trị riêng của hệ thống đều là một cặp của các số phức liên hợp, và tổ hợp hai dạng tự nhiên (phức) đó sinh ra một dao động duy trì với tần số góc $\omega_0 = \sqrt{K/M}$. Điều này phù hợp với cảm nhận vật lý của chúng ta, vì chúng ta cho rằng dao động được duy trì khi hệ thống có điều kiện đầu khác không ngay cả khi ngoại lực $f(t)$ bằng không.

Khi hệ thống bị suy giảm nhẹ ($D^2 < 4KM$), các trị riêng của ma trận là các số phức liên hợp, và tổ hợp của các dạng phức tự nhiên sẽ sinh ra một sóng sin suy giảm theo hàm mũ. Điều này cũng phù hợp với cảm quan, vì năng lượng tích trữ ban đầu trong khối lượng và lò xo sẽ chạy từ khối

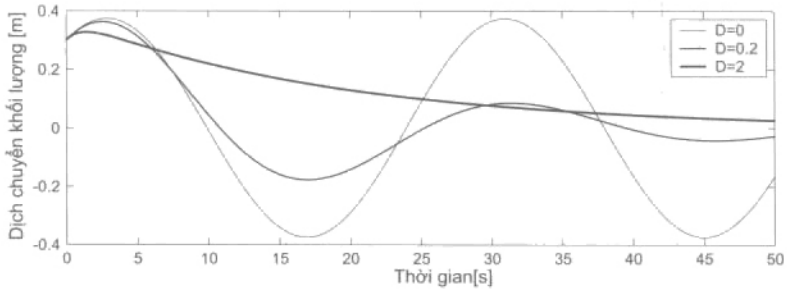
lượng tới lò xo theo chu kì và ngược lại nhưng cuối cùng nó sẽ bị hao tổn hoàn toàn, như nhiệt trong ma sát.

Cuối cùng, nếu suy giảm lớn ($D^2 > 4KM$), trị riêng của ma trận là các cặp số thực âm và mô hình ban đầu là hai hàm mũ suy giảm. Suy giảm mạnh sẽ ngăn cản dao động và năng lượng ban đầu sẽ suy giảm nhanh chóng.

Ba trường hợp khác nhau được biểu diễn trong hình 24.3. Để mô phỏng, chúng ta sử dụng ba giá trị khác nhau của hằng số ma sát D và:

$$M = 2 \text{ kg}, K = 0.1 \text{ N/m } d(0) = 0.3 \text{ m}, v(0) = 0.05 \text{ m/s} \tag{24.55}$$

Lưu ý rằng, khối lượng sẽ tiệm cận dừng trừ khi không có ma sát ($D = 0$).



HÌNH 24.3 Đáp ứng không cưỡng bức của hệ khối lượng - lò xo

Cấu trúc của đáp ứng cưỡng bức

Khi trạng thái ban đầu bằng không, nó sẽ chỉ thể hiện thành phần cưỡng bức. Thành phần cưỡng bức của trạng thái sẽ bao gồm các dạng tự nhiên và một số dạng cưỡng bức hoặc đặc biệt thêm vào, mà các dạng này phụ thuộc vào bản chất của đầu vào $\mathbf{u}(t)$ của hệ thống. Nói chung các dạng cưỡng bức của đầu vào cũng sẽ xuất hiện trong trạng thái. Tuy nhiên, một vài trường hợp đặc biệt xuất hiện khi một số dạng cưỡng bức của $\mathbf{u}(t)$ trùng với một số dạng tự nhiên của hệ thống.

Tính ổn định của hệ thống

Ổn định trong các hệ thống tuyến tính, bất biến cũng có thể được phân tích bằng cách sử dụng ma trận trạng thái **A**.

Tất cả các biến hệ thống có thể được biểu diễn như các hàm tuyến tính của trạng thái và đầu vào hệ thống. Khi đầu vào hệ thống $\mathbf{u}(t)$ là một vectơ của các hàm thời gian bị giới hạn, thì tính không giới hạn của các biến hệ thống phụ thuộc vào trạng thái bị giới hạn.

Vì vậy ta có kết quả sau:

Định lý 24.1 : *Xét hệ thống với mô tả trạng thái (24.42) và (24.43) trong đó **B, C** và **D** có các phần tử bị giới hạn. Khi đó trạng thái của hệ (và vì vậy đầu ra của hệ) bị giới hạn với tất cả các đầu vào bị giới hạn nếu và chỉ nếu các trị riêng của **A** có các phần thực âm.*

Để minh họa định lý này, chúng ta xét lại hệ trong ví dụ 24.2. Ma trận **A** (trong mô hình đã được tuyến tính hoá) được cho bởi:

$$\mathbf{A} = \begin{bmatrix} -\dfrac{R}{L} & 0 & 0 \\ 0 & 0 & -1 \\ \dfrac{R_g}{E_Q} & -\dfrac{Rmg^2}{K_1 E_Q} & 0 \end{bmatrix} \tag{24.56}$$

và trị riêng của nó là định thức của $\det(\lambda\mathbf{I} - \mathbf{A}) = 0$, trong đó:

$$\det(\lambda \mathbf{I} - \mathbf{A}) = \left(\lambda + \frac{R}{L}\right)\left(\lambda - \sqrt{\frac{Rmg^2}{K_1 E_Q}}\right)\left(\lambda + \sqrt{\frac{Rmg^2}{K_1 E_Q}}\right) \tag{24.57}$$

Có thể thấy rằng tập các trị riêng của ma trận bao gồm một phần là thực và lớn hơn không. Điều này cho thấy hệ thống không ổn định. Kết quả này cũng phù hợp với suy luận vật lý. Thật vậy, ít nhất về mặt lý thuyết, chúng ta có thể định vị được quả cầu trong quá trình cân bằng (điều này được miêu tả bởi x_{2Q} trong (24.37)). Tuy nhiên, đây là điểm cân bằng không ổn định, vì ngay khi ta tác động nhẹ vào quả cầu, nó tăng tốc cả về phía mặt đất hoặc về phía nam châm.

Tốc độ đáp ứng và cộng hưởng

Ngay cả nếu hệ thống ổn định vẫn còn một số vấn đề liên quan đến các thuộc tính cơ bản khác.

Đầu tiên, trong hệ thống ổn định, phần thực của trị riêng xác định tốc độ mà tại đó dạng liên quan hội tụ về không. Các dạng thấp nhất, các dạng trội xác định tốc độ mà tại đó đầu ra hệ thống đạt được giá trị trạng thái tĩnh của nó, có nghĩa là xác định tốc độ đáp ứng của hệ thống. Ví dụ nếu các trị riêng trội của hệ thống là $\lambda_{12} = -\sigma \pm j\omega_0, \sigma > 0$, thì các dạng tự nhiên được tổ hợp sẽ tạo ra một sóng sin suy giảm theo hàm mũ $y(t) = Ae^{\sigma t}\sin(\omega_0 t + \alpha)$. Ta thấy rằng tín hiệu này giảm nhanh hơn khi σ lớn hơn.

Vấn đề thứ hai, đặc biệt quan trọng cho các cấu trúc linh hoạt, là sự có mặt của cộng hưởng, liên quan đến các trị riêng phức. Trong các hệ thống vật lý, sự tồn tại của các trị riêng phức có liên hệ mật thiết tới sự hiện diện của hai dạng năng lượng. Cộng hưởng mô tả dao động (suy giảm chậm) giữa hai dạng năng lượng này. Trong các mạch điện, các dạng năng lượng đó là năng lượng tĩnh điện trong các tụ điện và năng lượng điện từ trong các cuộn cảm. Trong các hệ cơ, ta có động năng của các khối lượng chuyển động và thế năng trong các lò xo. Các cấu trúc linh hoạt có thể có nhiều dạng cộng hưởng. Một trong những vấn đề chính với cộng hưởng xảy ra khi đầu vào chứa năng lượng ở tần số gần với tần số cộng hưởng. Ví dụ nếu hệ thống có trị riêng $\lambda_{12} = -0.05 \pm j$, có nghĩa là tần số cộng hưởng là 1 rad/s và ngoài ra, một trong các thành phần đầu vào là một sóng sin với tần số 0.9 rad/s, thì đầu ra của hệ thống có dao động (cưỡng bức) rất lớn với biên độ ban đầu tăng gần như tuyến tính và sau đó ổn định ở một giá trị không đổi. Trong thực tế, hiện tượng này có thể phá huỷ hệ thống (trường hợp cầu Tacoma).

Biến đổi tương đương trạng thái

Chúng ta đã nói rằng việc chọn các biến trạng thái là không duy nhất. Ta có hệ thống với đầu vào $u(t)$, đầu ra $y(t)$ và hai lựa chọn khác nhau của các vectơ trạng thái: $\mathbf{x}(t) \in \mathbb{R}^n$ với 4 ma trận trạng thái $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, và $\bar{\mathbf{x}}(t) \in \mathbb{R}^n$ với $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}})$. Tồn tại ma trận không suy biến $\mathbf{T} \in \mathbb{R}^{n \times n}$ sao cho:

$$\bar{\mathbf{x}}(t) = \mathbf{T}\mathbf{x}(t) \Leftrightarrow \mathbf{x}(t) = \mathbf{T}^{-1}\bar{\mathbf{x}}(t) \tag{24.58}$$

Từ đó ta có các tương đương sau:

$$\bar{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}, \qquad \bar{\mathbf{B}} = \mathbf{T}\mathbf{B}, \qquad \bar{\mathbf{C}} = \mathbf{C}\mathbf{T}^{-1} \tag{24.59}$$

Các lựa chọn biến trạng thái khác nhau có thể hoặc không thể đáp ứng các phương pháp phân tích hệ thống khác nhau. Nhiều khi chỉ là một vấn đề đơn giản toán học, như ta thấy ở phần 24.6. Trong các trường hợp khác, quyết định được đưa ra bằng cách xem xét khả năng đo lường các biến hệ thống nào đó. Tuy nhiên, điều quan trọng là các đặc tính hệ thống cơ bản không thay đổi, không phụ thuộc vào mô tả trạng thái được chọn. Chúng được liên hệ với thực tế rằng các trị riêng của hệ thống là bất biến tương ứng với các biến đổi tương đương, vì:

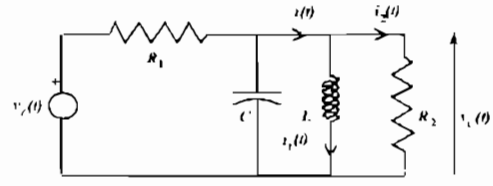
$$\det(\lambda\mathbf{I} - \overline{\mathbf{A}}) = \det(\lambda\mathbf{T}\mathbf{T}^{-1} - \mathbf{T}\mathbf{A}\mathbf{T}^{-1}) = \det(\mathbf{T})\det(\lambda\mathbf{I} - \mathbf{A})\det(\mathbf{T}^{-1}) \tag{24.60}$$

$$= \det(\lambda\mathbf{I} - \mathbf{A}) \tag{24.61}$$

Vì vậy, sự ổn định, bản chất của đáp ứng không cưỡng bức và tốc độ đáp ứng đều bất biến với các biến đổi tương đương.

Ví dụ 24.3

Xét mạch điện như hình 24.4.



HÌNH 24.4 Mạch điện

Chọn vectơ trạng thái $\mathbf{x}(t) = [x_1(t)\ \ x_2(t)]^T = [i_L(t)\ \ v_c(t)]^T$ đặt $u(t) = v_i(t)$. Sử dụng quy tắc thứ nhất ta có:

$$\frac{d\mathbf{x}(t)}{dt} = \begin{bmatrix} 0 & \dfrac{1}{L} \\ -\dfrac{1}{C} & -\dfrac{R_1 + R_2}{R_1 R_2 C} \end{bmatrix}\mathbf{x}(t) + \begin{bmatrix} 1 \\ \dfrac{1}{R_1 C} \end{bmatrix}u(t) \tag{24.62}$$

Vectơ trạng thái hoán chuyển là $\overline{\mathbf{x}}(t) = \left[\overline{x}_1(t)\ \ \overline{x}_2(t)\right]^T = \left[i(t)\ \ i_2(t)\right]^T$. Từ đó ta có:

$$\overline{\mathbf{x}}(t) = \underbrace{\frac{1}{R_2}\begin{bmatrix} R_2 & 1 \\ 0 & 1 \end{bmatrix}}_{\mathbf{T}}\mathbf{x}(t) \tag{24.63}$$

Không gian trạng thái và các hàm truyền

Mô tả không gian trạng thái của các hệ tuyến tính bất biến tuyến tính là một cách mô tả khác so với cách được đưa ra bởi các hàm truyền. Nói đúng ra, mô tả không gian trạng thái có phạm vi rộng hơn như sẽ thấy trong phần này.

Cho hệ tuyến tính bất biến với đầu vào $\mathbf{u}(t) \in \mathbb{R}^m$ và đầu ra $\mathbf{y}(t) \in \mathbb{R}^p$, hàm truyền $\mathbf{H}(s) \in \mathbb{C}^{p \times m}$, được định nghĩa bởi phương trình sau:

$$\mathbf{Y}(s) = \mathbf{H}(s)\mathbf{U}(s)\text{, trong đó } \left[\mathbf{H}(s)\right]_{ij} = \frac{\mathbf{Y}_i(s)}{\mathbf{Y}_j(s)} \tag{24.64}$$

thành phần (i, j) trong ma trận $\mathbf{H}(s)$ là biến đổi Laplace của đáp ứng theo đầu ra i^{th} khi xung đơn vị được đặt ở đầu vào j^{th}, với điều kiện đầu bằng không và với các đầu vào còn lại bằng không với mọi $t \geq 0$.

Mặt khác, nếu ta biến đổi Laplace (24.42) và (24.43) với các điều kiện đầu bằng không, ta có:

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) \tag{24.65}$$

$$\mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s) = \underbrace{(\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D})}_{\mathbf{H}(s)}\mathbf{U}(s) \tag{24.66}$$

Để đơn giản và để có thể phân tích kỹ hơn, trong phần còn lại của mục này ta tập trung vào lớp hệ thống vô hướng, có nghĩa là hệ thống với một đầu vào và một đầu ra (hệ SISO). Điều này có nghĩa là $m = p = 1$, $\mathbf{B}$ trở thành vectơ cột, $\mathbf{C}$ là vectơ hàng, và $D = H(\infty)$ (trong các hệ thống thực thường giữ sao cho $D = H(\infty) = 0$). Với hệ thống SISO, $H(s)$ là thương của các đa thức theo s, có nghĩa là:

$$H(s) = \frac{\mathbf{C}Adj(s\mathbf{I} - \mathbf{A})\mathbf{B} + \mathbf{D}\det(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})} \tag{24.67}$$

trong đó ký hiệu Adj(o) là ma trận liên hợp của (o).

Vấn đề quan trọng là các điểm cực của hàm truyền là các trị riêng của ma trận **A**. Tuy nhiên, điều này không đúng, nói chung, tập các điểm cực của hàm truyền là đồng nhất với tập các trị riêng của ma trận **A**. Điều này có thể được hiểu rõ qua ví dụ sau.

Ví dụ 24.4

Đặt:

$$\mathbf{A} = \begin{bmatrix} -2 & 1 \\ 0 & -3 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 1 \\ 0,5 \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \qquad \mathbf{D} = 0 \tag{24.68}$$

Khi đó:

$$H(s) = C(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \frac{1}{(s+2)(s+3)}\begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} s+3 & 1 \\ 0 & s+2 \end{bmatrix}\begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \tag{24.69}$$

$$= \frac{0.5(s+2)}{(s+2)(s+3)} = \frac{0.5}{(s+3)} \tag{24.70}$$

Vì vậy, hàm truyền chỉ có một điểm cực, mặc dù ma trận **A** có hai trị riêng. Chúng ta thấy rằng có một điểm cực - không bị loại ở $H(s)$. Hiện tượng này được gắn liền với vấn đề thuộc tính của hệ thống, là chủ đề trọng tâm của 24.6.

Xét hệ thống cảm ứng từ trong ví dụ 24.2. Nếu ta định nghĩa dòng điện $i(t)$ là đầu ra của hệ thống, ta có thể ngay lập tức thấy rằng hàm truyền từ đầu vào $e(t)$ tới đầu ra này chỉ có một điểm cực. Điều này trái với thực tế rằng chiều của trạng thái bằng ba. Để giải thích cho điều này, trong mô hình vật lý đã được đơn giản hóa, dòng điện $i(t)$ không bị tác động bởi vị trí và tốc độ của quả cầu kim loại (chú ý rằng, ta đã bỏ qua sự thay đổi của độ tự cảm do các thay đổi vị trí của quả cầu).

Kết quả quan trọng có được là hàm truyền không thể cung cấp cùng một lượng thông tin như mô hình không gian trạng thái cho cùng một hệ thống.

Vấn đề thú vị là làm thế nào để nhận được một mô tả không gian trạng thái từ một hàm truyền cho trước. Người đọc phải biết rằng mô hình không gian trạng thái kết quả không thể hiện được các sự triệt tiêu điểm cực - không. Vì lý do đó, mô tả nhận đó được biết như **một sự thực hiện tối giản**.

Có nhiều phương pháp để đi từ hàm truyền tới một mô hình không gian trạng thái. Sau đây, chúng ta sẽ trình bày một trong những phương pháp đó.

Xét hàm truyền cho bởi:

$$H(t) = \frac{B_0(s)}{A_0(s)} + H_f(\infty) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} + H_f(\infty) \tag{24.71}$$

Trước tiên chúng ta nhớ lại rằng $D = H_f(\infty)$. Do đó chúng ta tập trung vào hàm truyền $H(s) = H_f(s) - H_f(\infty)$ là hàm truyền thích hợp.

Xét biến tiếp theo $v_f(t) \in \mathbb{R}$ có biến đổi Laplace $V_f(s)$ thoả mãn:

$$V_i(t) = \frac{s^{i-1}}{A_0(s)}U(s), \qquad i \in \{1,2,.....n\} \tag{24.72}$$

Điều này có nghĩa rằng:

$$v_i(t) = \frac{dv_{i-1}(t)}{dt}, \qquad i \in \{2,.....n\} \tag{24.73}$$

$$Y(s) = \sum_{i=1}^{n} b_{i-1}V_i(s) \tag{24.74}$$

$$U(s) = \frac{A_0(s)}{A_0(s)}U(s) = \underbrace{\frac{s^n}{A_0(s)}U(s)}_{V_n(s)} + \sum_{i=1}^{n} a_{i-1}\underbrace{\frac{s^{i-1}}{A_0(s)}U(s)}_{V_i(s)} \tag{24.75}$$

Bây giờ chọn các biến trạng thái:

$$x_i(t) = v_i(t) \tag{24.76}$$

Từ các phương trình trên, ta có:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{24.77}$$

$$\mathbf{C}=[\,b_0 \quad b_1 \quad b_2 \quad \dots \quad b_{n-1}\,], \qquad \mathbf{D} = H_f(\infty) \tag{24.78}$$

Ví dụ 24.5

Hàm truyền của một hệ thống được cho bởi:

$$H(s) = \frac{4s-10}{(s+2)^2(s-1)} = \frac{4s-10}{s^3+3s^2-4} \tag{24.79}$$

Khi đó thực hiện tối thiểu cho hệ thống này là:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 4 & 0 & -3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{24.80}$$

$$\mathbf{C} = \begin{bmatrix} -10 & 4 & 0 \end{bmatrix}, \quad \mathbf{D} = 0 \tag{24.81}$$

Kết quả quan trọng được rút ra là hàm truyền của hệ thống là bất biến với các biến đổi trạng thái tương đương.

24.4 Mô tả không gian trạng thái cho hệ thống thời gian rời rạc và dữ liệu lấy mẫu

Phần này sẽ giới thiệu khái quát về mô tả không gian trạng thái cho các hệ thống thời gian rời rạc, chủ yếu dựa trên các kết quả đã được trình bày cho trường hợp thời gian liên tục.

Các mô hình thời gian rời rạc từ hai nguồn khác nhau:

- Từ hệ thống thời gian rời rạc thuần túy, thường là không tuyến tính, các biến được định nghĩa tại thời điểm xác định t_k. Các hệ thống như vậy có thể thấy trong các hệ thống kinh tế, lý thuyết xử lý stochastic...
- Từ việc rời rạc hóa hệ thống thời gian liên tục. Trong trường hợp này chúng ta chỉ quan tâm với giá trị của một vài biến hệ thống tại những thời điểm xác định. Các mô hình này thường được sử dụng khi các hệ thống số, như vi điều khiển, máy tính, PLC... tương tác với các hệ thống thực với miền thời gian liên tục, như các cấu trúc cơ khí, van, bồn chứa, mạch điện tương tự hoặc cả một quá trình xử lý công nghiệp. Chúng được gọi là các **hệ thống dữ liệu trích mẫu**.

Trong cả hai trường hợp trên, phân tích của chúng ta sẽ tập trung vào lớp các mô hình **tuyến tính và bất biến.**

Tuyến tính hóa các hệ thống thời gian rời rạc

Thời gian rời rạc tương đương với (24.3) và (24.4) được cho bởi các phương trình phi tuyến sau:

$$\mathbf{x}[t+1] = \mathbf{F}_d(\mathbf{x}[t], \mathbf{u}[t]) \tag{24.82}$$

$$\mathbf{y}[t] = \mathbf{G}_d(\mathbf{x}[t], \mathbf{u}[t]) \tag{24.83}$$

Tuyến tính hóa các mô hình của các hệ thống thời gian rời rạc sẽ cho ra đường giống với đường của hệ thời gian liên tục. Xét điểm cân bằng được cho bởi $\{\mathbf{x}_Q, \mathbf{u}_Q, \mathbf{y}_Q\}$:

$$\mathbf{x}_Q = \mathbf{F}_d(\mathbf{x}_Q, \mathbf{u}_Q) \tag{24.84}$$

$$\mathbf{y}_Q = \mathbf{G}_d(\mathbf{x}_Q, \mathbf{u}_Q) \tag{24.85}$$

Lưu ý rằng điểm cân bằng được xác định bởi một tập các giá trị không đổi của trạng thái và các giá trị không đổi của đầu vào thỏa mãn (24.82) và (24.83). Với các điều kiện ở trên tạo ra hệ đầu ra hệ thống không đổi. Vì vậy, mô hình rời rạc có thể được tuyến tính hóa xung quanh điểm cân bằng này. Định nghĩa:

$$\Delta\mathbf{x}[t] = \mathbf{x}[t] - \mathbf{x}_Q, \qquad \Delta\mathbf{u}[t] = \mathbf{u}[t] - \mathbf{u}_Q, \qquad \Delta\mathbf{y}[t] = \mathbf{y}[t] - \mathbf{y}_Q \tag{24.86}$$

chúng ta có mô hình không gian trạng thái:

$$\Delta\mathbf{x}[t+1] = \mathbf{A}_d\Delta\mathbf{x}[t] + \mathbf{B}_d\Delta\mathbf{u}[t] \tag{24.87}$$

$$\Delta\mathbf{y}[t] = \mathbf{C}_d\Delta\mathbf{x}[t] + \mathbf{D}_d\Delta\mathbf{u}[t] \tag{24.88}$$

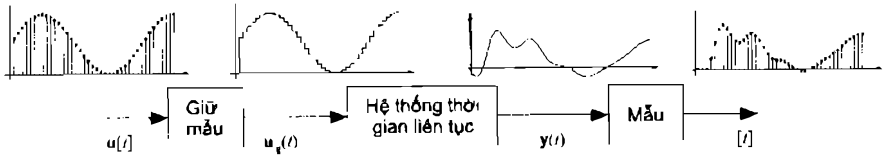
trong đó:

$$\mathbf{A}_d = \left.\frac{\partial \mathbf{F}_d}{\partial \mathbf{x}}\right|_{\substack{x=x_Q \\ u=u_Q}}, \quad \mathbf{B}_d = \left.\frac{\partial \mathbf{F}_d}{\partial \mathbf{u}}\right|_{\substack{x=x_Q \\ u=u_Q}}, \quad \mathbf{C}_d = \left.\frac{\partial \mathbf{G}_d}{\partial \mathbf{x}}\right|_{\substack{x=x_Q \\ u=u_Q}}, \quad \mathbf{D}_d = \left.\frac{\partial \mathbf{G}_d}{\partial \mathbf{u}}\right|_{\substack{x=x_Q \\ u=u_Q}} \tag{24.89}$$

Các hệ thống dữ liệu lấy mẫu

Như chúng ta thảo luận ở trên, các mô hình thời gian rời rạc thường đạt được nhờ lấy mẫu các đầu vào và các đầu ra của hệ thống thời gian liên tục. Khi thiết bị số được sử dụng để hoạt động dựa trên miền thời gian liên tục, các tín hiệu lệnh chỉ cần được định nghĩa ở những thời điểm xác định chứ không phải *toàn bộ* thời gian. Tuy nhiên, để có thể hoạt động trên hệ thống thời gian liên

tục, chúng ta cần một tín hiệu với thời gian liên tục. Điều này thường được xây dựng bằng việc *giữ mẫu bậc không*, sẽ tạo ra một tín hiệu bậc thang. Ngoài ra, khi ta muốn đo theo kiểu số một biến hệ thống, chúng ta đo tại những thời điểm xác định. Có nghĩa là chúng ta phải lấy mẫu các tín hiệu đầu ra. Hình (24.5) minh họa các khái niệm này. Nếu ta giả thiết việc lấy mẫu là tuần hoàn, với chu kì Δ, thì ta chỉ quan tâm tới các tín hiệu tại $k\Delta$. Tiếp theo ta sẽ bỏ Δ khỏi đối số, bằng cách dùng $\mathbf{u}(k\Delta) = \mathbf{u}[t]$ cho đầu vào, $\mathbf{y}(k\Delta) = \mathbf{y}[t]$ cho đầu ra, và $\mathbf{x}(k\Delta) = \mathbf{x}[t]$ cho trạng thái hệ thống.



HÌNH 24.5 Hệ thống dữ liệu lấy mẫu

Nếu ta xem xét mô hình không gian trạng thái tuyến tính bất biến với thời gian và liên tục, được định nghĩa bởi (24.42) và (24.43), với trạng thái ban đầu $\mathbf{x}(k_0\Delta) = \mathbf{x}_0$, chúng ta có thể sử dụng phương trình (24.44) để tính toán giá trị tiếp theo của trạng thái:

$$\mathbf{x}(k_0\Delta + \Delta) = e^{\mathbf{A}(k_0\Delta + \Delta - k_0\Delta)}\mathbf{x}(k_0\Delta) + \int_{k_0\Delta}^{k_0\Delta+\Delta} e^{\mathbf{A}(k_0\Delta+\Delta-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \tag{24.90}$$

Hơn nữa, nếu giữ mẫu bậc không được sử dụng, có nghĩa là $\mathbf{u}(\tau) = \mathbf{u}(k_0\Delta)$ cho $k_0\Delta \le \tau < k_0\Delta + \Delta$, chúng ta sẽ nhận được:

$$\mathbf{x}(k_0\Delta + \Delta) = e^{\mathbf{A}\Delta}\mathbf{x}(k_0\Delta) + \int_0^{\Delta} e^{\mathbf{A}\eta}d\eta\mathbf{B}\mathbf{u}(k_0\Delta) \tag{24.91}$$

Và, nếu ta biết trạng thái và đầu vào tại thời gian $k_0\Delta$, thì đầu ra được xác định bởi phương trình (24.43):

$$\mathbf{y}(k_0\Delta) = \mathbf{C}\mathbf{x}(k_0\Delta) + \mathbf{D}\mathbf{u}(k_0\Delta) \tag{24.92}$$

Bây giờ chúng ta có thể kết luận rằng một mô hình thời gian liên tục với các ma trận không gian trạng thái $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, và chúng ta lấy mẫu các đầu vào, các đầu ra, mỗi Δ giây thì hệ thống dữ liệu lấy mẫu tương đương sẽ được mô tả bởi mô hình không gian trạng thái rời rạc

$$\mathbf{x}(k\Delta + \Delta) = \mathbf{A}_d\mathbf{x}(k\Delta) + \mathbf{B}_d\mathbf{u}(k\Delta) \tag{24.93}$$

$$\mathbf{y}(k\Delta) = \mathbf{C}_d\mathbf{x}(k\Delta) + \mathbf{D}_d\mathbf{u}(k\Delta) \tag{24.94}$$

trong đó:

$$\mathbf{A}_d = e^{\mathbf{A}\Delta}, \quad \mathbf{B}_d = \int_0^{\Delta} e^{\mathbf{A}\eta}d\eta\mathbf{B}, \quad \mathbf{C}_d = \mathbf{C}, \quad \mathbf{D}_d = \mathbf{C} \tag{24.95}$$

Có nhiều phương pháp khác nhau để nhận được $\mathbf{A}_d$ đã được định nghĩa bởi (24.95), nhưng cách đơn giản để tính ma trận này là dùng biến đổi Laplace. Điều này sẽ dẫn ra:

$$\mathbf{A}_d = e^{\mathbf{A}\Delta} = \mathcal{L}^{-1}\{(s\mathbf{I} - \mathbf{A})^{-1}\}\Big|_{t=\Delta} \tag{24.96}$$

Ví dụ 24.6

Xét hệ cơ học trong ví dụ 24.1 đã được mô tả bởi mô hình không gian trạng thái:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{M}{D} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{M} \end{bmatrix} f(t) \tag{24.97}$$

trong đó $f(t)$ là ngoại lực, ta có thể chọn vị trí khối lượng, $x_1(t)$, hoặc tốc độ vật, $x_2(t)$, như là các đầu ra hệ thống.

Với mục đích minh họa số, chúng ta đặt $M = 1$kg, $D = 1.2$N s/m, và $K = 0.32$ N/m.

Ma trận $\mathbf{A}_d$ nhận được từ (24.96), bằng cách áp dụng biến đổi Laplace ngược:

$$A_d = \mathcal{L}^{-1}\left\{ \begin{bmatrix} s & -1 \\ 0.32 & s+1.2 \end{bmatrix}^{-1} \right\}\Bigg|_{t=\Delta} = \begin{bmatrix} 2e^{-0.4\Delta} - e^{-0.8\Delta} & 2.5(e^{-0.4\Delta} - e^{-0.8\Delta}) \\ 0.8(e^{-0.4\Delta} - e^{-0.8\Delta}) & -e^{-0.4\Delta} + 2e^{-0.8\Delta} \end{bmatrix} \tag{24.98}$$

ma trận $\mathbf{B}_d$ có được từ (24.95):

$$\mathbf{B}_d = \int_0^{\Delta} \begin{bmatrix} 2e^{-0.4\eta} - e^{-0.8\eta} & 2.5(e^{-0.4\eta} - e^{-0.8\eta}) \\ 0.8(e^{-0.4\eta} - e^{-0.8\eta}) & -e^{-0.4\eta} + 2e^{-0.8\eta} \end{bmatrix} d\eta \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{24.99}$$

$$\Rightarrow \mathbf{B}_d = \begin{bmatrix} -6.25e^{-0.4\Delta} + 3.125e^{-0.8\Delta} + 3.125 \\ 2.5(e^{-0.4\Delta} - e^{-0.8\Delta}) \end{bmatrix}$$

Chú ý rằng, cả $\mathbf{A}_d$ và $\mathbf{B}_d$ đều là hàm của Δ. Vì vậy, chu kì lấy mẫu Δ có vai trò quan trọng trong ứng xử động của hệ thống trích mẫu như chúng ta sẽ thấy trong các mục con theo sau đây.

Các mô hình không gian trạng thái tuyến tính

Chúng ta sẽ phân tích mô hình không gian trạng thái bất biến tuyến tính:

$$\mathbf{x}[t+1] = \mathbf{A}_d\mathbf{x}[t] + \mathbf{B}_d\mathbf{u}[t] \tag{24.100}$$

$$\mathbf{y}[t] = \mathbf{C}_d\mathbf{x}[t] + \mathbf{D}_d\mathbf{u}[t] \tag{24.101}$$

Đây có thể là mô hình thời gian rời rạc được tuyến tính hoá giống như (24.87) và (24.88), hoặc một hệ thống dữ liệu lấy mẫu giống như (24.93) và (24.94) trong đó Δ được bỏ khỏi đối số thời gian.

Lời giải của phương trình (24.100) và (24.101), $x[t_0] = x_0$, được cho bởi:

$$\mathbf{x}[t] = \mathbf{A}_d^{(t-t_0)}\mathbf{x}_0 + \sum_{i=0}^{(t-t_0)-1} \mathbf{A}_d^{(t-t_0)-i-1}\mathbf{B}_d\mathbf{u}[i+t_0] \qquad \forall t \geq t_0 \tag{24.102}$$

trong đó $\mathbf{A}_d^{(t-t_0)}$ là **ma trận chuyển tiếp**.

Bạn đọc có thể dễ dàng kiểm tra lại (24.102) thỏa mãn (24.100). Với kết quả trên, lời giải cho (24.101) được cho bởi:

$$\mathbf{y}[t] = \mathbf{C}_d\mathbf{A}_d^{(t-t_0)}\mathbf{x}_0 + \mathbf{C}_d\sum_{i=0}^{(t-t_0)-1} (\mathbf{A}_d^{(t-t_0)-i-1}\mathbf{B}_d\mathbf{u}[i+t_0]) + \mathbf{D}_d\mathbf{u}[t] \tag{24.103}$$

Hệ thống động lực

Trạng thái của hệ thống gồm hai thành phần: thành phần **không cưỡng bức**, $\mathbf{x}_u[t]$, và thành phần **cưỡng bức** $\mathbf{x}_f[t]$, trong đó:

$$\mathbf{x}_u[t] = \mathbf{A}_d^{(t-t_0)}\mathbf{x}_0 \tag{24.104}$$

$$\mathbf{x}_f[t] = \sum_{\tau=0}^{(t-t_0)-1} \mathbf{A}_d^{(t-t_0)-\tau-1}\mathbf{B}_d\mathbf{u}[\tau + t_0] \tag{24.105}$$

Để hiểu rõ hơn mô hình không gian trạng thái và lời giải của nó, xét trường hợp khi $t_0 = 0$ và $\mathbf{u}[t] = 0$, $\forall t \geq 0$, có nghĩa là trạng thái chỉ có **thành phần không cưỡng bức**, khi đó:

$$\mathbf{x}[t] = \mathbf{A}_d^t\mathbf{x}_0 \tag{24.106}$$

Hơn nữa, giả thuyết rằng $\mathbf{A}_d \in \mathbb{R}^{n\times n}$ và để đơn giản hơn, nó chứa n trị riêng khác nhau η_i, với n vectơ trị riêng độc lập tuyến tính $\mathbf{v}_i$. Khi đó luôn tồn tại n hằng số α_i thỏa mãn:

$$\mathbf{x}_0 = \sum_{i=1}^{n}\alpha_i\mathbf{v}_i, \quad \alpha_i \in \mathbb{C} \tag{24.107}$$

Từ một kết quả nổi tiếng của toán đại số cho chúng ta thấy rằng các trị riêng của $\mathbf{A}_d^k$ là η_i^k, với $k \in \mathbb{N}$ tương ứng với các vectơ riêng $\mathbf{v}_i$:

$$\mathbf{x}[t] = \mathbf{A}_d^t\mathbf{x}_0 = \mathbf{A}_d^t\sum_{i=1}^{n}\alpha_i\mathbf{v}_i = \sum_{i=1}^{n}\alpha_i\underbrace{\mathbf{A}_d^t\mathbf{v}_i}_{\eta_i^t\mathbf{v}_i} \tag{24.108}$$

$$\mathbf{x}[t] = \sum_{i=1}^{n}\alpha_i\eta_i^t\mathbf{v}_i \tag{24.109}$$

Công thức này cho thấy thành phần không cưỡng bức của trạng thái là tổ hợp tuyến tính của **các dạng tự nhiên**, $\{\eta_i^t\}$, và mỗi phần gắn với một trị riêng của $\mathbf{A}_d$, mà nó cũng được biết như là các **tần số cơ bản** của mô hình. Vì vậy, chúng ta lại biết rằng ma trận $\mathbf{A}_d$ xác định:

- Cấu trúc của đáp ứng không cưỡng bức
- Sự ổn định của hệ thống
- Tốc độ đáp ứng

Cấu trúc của đáp ứng không cưỡng bức

Khi không có đầu vào, trạng thái như là tổ hợp của các dạng tự nhiên thuộc một lớp hàm đã được định nghĩa: năng lượng của các trị riêng của *dạng*, gồm cả thực hoặc phức. Các *dạng* đó là các hàm rời rạc liên quan đến các hằng số, các hàm mũ thực, sóng sin cơ bản, các sóng sin đã bị điều chế theo hàm mũ, và một số hàm đặc biệt khác sinh ra từ các trị riêng lặp lại.

Để minh họa các vấn đề trên và để làm sáng tỏ ý nghĩa vật lý của chúng, xét hệ thống được lấy mẫu trong ví dụ 24.6. Nếu $\Delta = 1$, ma trận không gian trạng thái là:

$$\mathbf{A}_d = \begin{bmatrix} 0.8913 & 0.5525 \\ -0.1768 & 0.2283 \end{bmatrix}, \qquad \mathbf{B}_d = \begin{bmatrix} 0.3397 \\ 0.5525 \end{bmatrix} \tag{24.110}$$

Vì vậy, các trị riêng của hệ thống là các nghiệm của phương trình:

$$\det(\eta\mathbf{I} - \mathbf{A}_d) = \det\left(\begin{bmatrix} \eta - 0.8913 & -0.5525 \\ 0.1768 & \eta - 0.2283 \end{bmatrix}\right) \tag{24.111}$$

$$= (\eta - 0.6703)(\eta - 0.4493) = 0 \tag{24.112}$$

Tức là $\eta_1 = 0.6703$, $\eta_2 = 0.4493$ và đáp ứng không cưỡng bức là:

$$\mathbf{x}_n[t] = C_1(0.6702)^t + C_2(0.4493)^t \tag{24.113}$$

Trong đó C_1 và C_2 chỉ phụ thuộc vào các điều kiện đầu. Chúng ta có thể thấy rằng, khi t tiến tới vô cùng, $x_0[t]$ suy giảm về không, do $|\eta_{1,2}| < 1$. Ngoài ra, các trị riêng đó là các số thực dương, nên không có dao động ở các dạng tự nhiên. Điều này phù hợp với việc chọn tham số trong ví dụ 1.6 làm cho hệ khối lượng - lò xo tránh được rung quá mức.

Cấu trúc của đáp ứng cưỡng bức

Xét phương trình (24.102). Khi trạng thái đầu là không, trạng thái sẽ chỉ thể hiện thành phần cưỡng bức. Tuy nhiên, thành phần cưỡng bức này sẽ vẫn bao gồm các *dạng* tự nhiên cộng với một số *dạng* **cưỡng bức** hoặc **đặc biệt** phụ thuộc vào bản chất của đầu vào $\mathbf{u}[t]$ của hệ thống. Nói chung, các dạng cưỡng bức của đầu vào cũng sẽ xuất hiện trong trạng thái. Tuy nhiên các trường hợp đặc biệt sẽ xuất hiện khi một *dạng* cưỡng bức của $\mathbf{u}[t]$ trùng với dạng hệ thống tự nhiên.

Tính ổn định hệ thống

Ổn định trong hệ thống bất biến thời gian tuyến tính cũng có thể được phân tích bằng cách sử dụng ma trận trạng thái $\mathbf{A}_d$. Như chúng ta đã nói, tất cả các biến hệ thống đều có thể được biểu diễn như các hàm tuyến tính của trạng thái và đầu vào hệ thống. Khi đầu vào hệ thống $\mathbf{u}[t]$ là một vectơ của hàm thời gian bị giới hạn, thì giới hạn biên của các biến hệ thống phụ thuộc vào trạng thái bị giới hạn, chúng ta có kết quả sau:

Định lý 24.2 *Xét hệ thống với mô tả trạng thái (24.100) và (24.101) trong đó* $\mathbf{B}_d$, $\mathbf{C}_d$ *và* $\mathbf{D}_d$ *có các phần tử bị giới hạn. Khi đó trạng thái hệ thống được giới hạn theo tất cả các đầu vào giới hạn nếu và chỉ nếu các trị riêng của* $\mathbf{A}_d$ *nằm trong đường tròn đơn vị, ví dụ* $|\eta_\ell| < 1, \ \forall \ \ell$.

Tốc độ đáp ứng và các cộng hưởng

Ta nhớ lại rằng các dạng cơ bản của hệ thời gian rời rạc là luỹ thừa của các trị riêng η_ℓ. Vì các trị riêng đó có thể luôn được mô tả như các đại lượng phức, nên chúng ta có thể viết các dạng tự nhiên như sau:

$$(\eta_\ell)^t = (|\eta_\ell| e^{j\theta_\ell})^t = |\eta_\ell|^t e^{j\theta_\ell t}, \qquad \theta_\ell = \angle \eta_\ell \tag{24.114}$$

Vì vậy chúng ta có

- $0 < |\eta_\ell| < \infty$ xác định tốc độ, tại đó dạng suy giảm tới không với hệ thống ổn định ($|\eta_\ell| < 1$), hoặc lớn tới vô cùng với các hệ thống không ổn định ($|\eta_\ell| > 1$)
- $-\pi < \theta_\ell \le \pi$ xác định tần số của dạng tự nhiên, được đo theo radian.

Mặc dù các *dạng* tự nhiên của các hệ thống ổn định suy giảm tới không, nhưng các tính chất tự nhiên của chúng xác định đáp ứng quá độ của hệ thống.

Để minh họa những vấn đề này, **đáp ứng bước** với điều kiện đầu bằng không, thường được sử dụng.

Ví dụ 24.7

Xét hệ thống thời gian rời rạc một đầu vào-một đầu ra (SISO):

$$\mathbf{x}[t+1] = \eta_\ell \mathbf{x}[t] + \mathbf{u}[t] \tag{24.115}$$

$$\mathbf{y}[t] = (1 - \eta_\ell)\mathbf{x}[t] \tag{24.116}$$

Để nhận được đáp ứng bước ta sử dụng phương trình (24.103), trong đó $\mathbf{x}_0 = 0$, $\mathbf{u}(t) = 1$, $\forall \ge 0$:

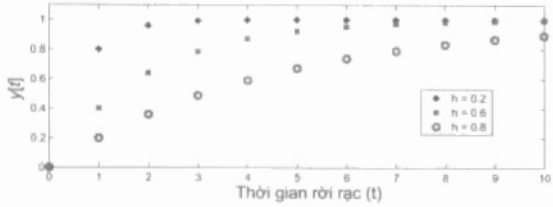
$$\mathbf{y}[t] = \mathbf{C}_d \left(\sum_{\tau=0}^{t-1} \mathbf{A}_d^{t-\tau-1} \right) \mathbf{B}_d \tag{24.117}$$

$$= (1-\eta_\ell)\left(\sum_{i=0}^{t-1}\eta_\ell^{t-i-1}\right) = (1-\eta_\ell)\eta_\ell^{t-1}\frac{1-\eta_\ell^{-t}}{1-\eta_\ell^{-1}} \tag{24.118}$$

$$= 1-\eta_\ell^t \tag{24.119}$$

Tín hiệu đầu ra, y[t] = y_h[t] + y_p[t], được biểu diễn trên hình (24.6), với các giá trị khác nhau của trị riêng η_ℓ. Quá độ nhận được bởi $\mathbf{y}_h[t] = -\eta_\ell^t$ và đáp ứng trạng thái ổn định bởi $\mathbf{y}_p[t] = 1$.

Chúng ta đã thấy trong phương trình (24.114), trị riêng hệ thống không những xác định sự tắt dần của đáp ứng quá độ mà còn xác định tần số dao động (khi trị riêng có thành phần ảo khác không). Vấn đề khả năng khi các dạng cộng hưởng tồn tại cũng là vấn đề chúng ta thấy trong nội dung của hệ thống thời gian liên tục, có nghĩa là, đầu vào hệ thống chứa một sóng sin hoặc một loại tín hiệu khác, với năng lượng ở một tần số gần với một trong các tần số tự nhiên của hệ thống. Đầu ra hệ thống vẫn còn bị giới hạn, mặc dù nó tăng tới biên độ không mong muốn.



HÌNH 24.6 Đáp ứng bước của hệ thống với các trị riêng khác nhau

Ví dụ 24.8

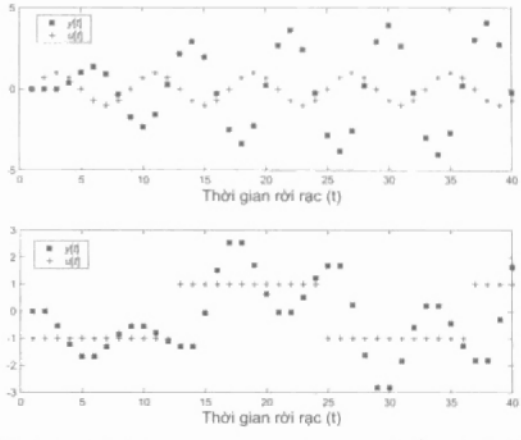
Xét hệ thống thời gian rời rạc được mô tả bởi mô hình không gian trạng thái:

$$\mathbf{x}[t+1] = \begin{bmatrix} 1.2796 & -0.81873 \\ 1 & 0 \end{bmatrix}\mathbf{x}[t] + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\mathbf{u}[t] \tag{24.120}$$

$$y[t] = [0 \quad 0.5391]\mathbf{x}[t] \tag{24.121}$$

Các trị riêng của hệ thống nhận được từ $\mathbf{A}_d$:

$$\eta_{1,2} = 0.6398 \pm j0.6398 = 0.9048(e^{j\pi/4}) \tag{24.122}$$



HÌNH 24.7 Ảnh hưởng của cộng hưởng tới đầu ra hệ thống

Và các dạng tự nhiên liên quan, có trong đáp ứng quá độ là:

$$\eta_{1,2}^t = 0.9048^t e^{j\frac{\pi}{4}t} = 0.9048^t \left[\cos\left(\frac{\pi}{4}t\right) \pm j \sin\left(\frac{\pi}{4}t\right) \right] \tag{24.123}$$

Các dạng tự nhiên đều suy giảm nhẹ, bởi vì $|\eta_{1,2}|$ gần với 1, và chúng sẽ đưa ra một dao động với tần số π/4.

Trong các đồ thị ở hình 24.7, ta thấy rõ ràng đầu ra cộng hưởng mạnh. Đồ thị trên tương ứng với đầu vào $u[t]=\sin(\frac{\pi}{4}t)$, có nghĩa là, tần số đầu vào trùng với tần số của các dạng tự nhiên. Trong đồ thị dưới, đầu vào là một sóng vuông của tần số π/12. Trong trường hợp này, hài **bậc ba** của đầu vào có tần số bằng với tần số của *dạng* tự nhiên.

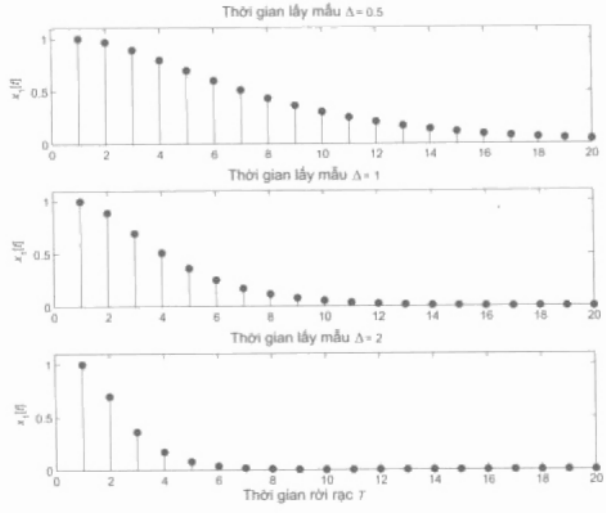
Ảnh hưởng của các chu kỳ lấy mẫu khác nhau

Phương trình (24.95) cho thấy rằng $\mathbf{A}_d$ và $\mathbf{B}_d$ phụ thuộc vào việc chọn chu kỳ lấy mẫu Δ. Việc lựa chọn này cũng quyết định vị trí của các trị riêng trong hệ thống. Nếu ta xem phương trình (24.96), giả thiết rằng **A** được đường chéo hóa, ta có:

$$\mathbf{A}_d = e^{diag[\lambda_1 \quad \lambda_n]\Delta} = diag\{e^{\lambda_1\Delta},...,e^{\lambda_n\Delta}\} \tag{24.124}$$

trong đó $\{\lambda_1,...,\lambda_n\}$ đều là các trị riêng của các hệ thống thời gian liên tục. Vì vậy những trị riêng đó được gắn với trị riêng của hệ thống dữ liệu lấy mẫu bằng phương trình:

$$n_\ell = e^{\lambda_\ell \Delta} \tag{24.125}$$



HÌNH 24.8 Ảnh hưởng của việc lấy mẫu tới các dạng tự nhiên

Trong hình 24.8 ta thấy đáp ứng của hệ thống được lấy mẫu trong ví dụ 24.6, chọn $x_1[t]$ là đầu ra của hệ thống, khi điều kiện đầu là $\mathbf{x}_0 = [1 \quad 0]^T$, với các giá trị khác nhau của Δ. Quan sát thấy rằng trục hoành tương ứng với t, do đó các thời điểm tức thời *thực* là tΔ.

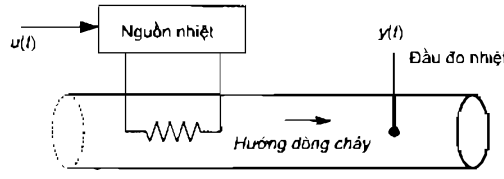
Một vấn đề cơ bản liên quan đến việc lấy mẫu tín hiệu trong miền thời gian liên tục là chu kì lấy mẫu được chọn đủ nhỏ để giữ được tính tự nhiên thiết yếu của tín hiệu được lấy mẫu. Để minh họa một lựa chọn tồi Δ, giả thiết rằng tín hiệu $f(t) = A\sin(\omega_0 t)$ được lấy mẫu mỗi Δ giây, với $\Delta = 2\ell\pi/\omega_0, \ell \in \mathbb{N}$. Khi đó tín hiệu thời gian rời rạc kết quả là $f(t) = 0, \forall t \in \mathbb{Z}$.

Hệ thống dữ liệu lấy mẫu và các trễ thời gian

Như chúng ta đã đề cập ở 24.3, không thể sử dụng mô hình không gian trạng thái trong miền thời gian liên tục để mô tả hệ thống với các trễ thời gian, vì chúng là các hệ thống có thứ nguyên vô tận. Cũng có thể nói rằng ta có thể thay đổi vấn đề này bằng cách sử dụng các tín hiệu lấy mẫu. Điều này được thực hiện như trong ví dụ sau.

Ví dụ 24.9

Xét hệ thống nhiệt trong hình 24.9.



HÌNH 24.9 Hệ thống nhiệt với trễ thời gian

Nhiệt độ được đo, $y(t)$, của dòng chảy phụ thuộc vào năng lượng tạo ra bởi nguồn nhiệt. Nguồn này được điều khiển bởi một tín hiệu điều khiển $u(t)$. Các thay đổi của $u(t)$ sẽ làm thay đổi nhiệt độ $y(t)$ với một trễ thời gian đáng kể. Vì vậy hệ thống được tuyến tính hóa có thể được biểu diễn bởi hàm truyền:

$$\frac{Y(s)}{U(s)} = H(s) = \frac{e^{-\tau s} K}{s + \lambda} \tag{24.126}$$

trong đó $U(s)$ và $Y(s)$ là biến đổi Laplace của $u(t)$ và $y(t)$.

Tiếp theo, chúng ta giả thiết rằng đầu vào và đầu ra được lấy mẫu mỗi $\Delta[s]$. Thời gian trễ τ theo giây, là hàm của vận tốc dòng chảy và để đơn giản hóa ta giả thiết rằng τ là bội số của khoảng thời gian lấy mẫu Δ, có nghĩa là $\tau = m\Delta$, $m \in \mathbb{Z}$. Các trễ này được thể hiện ở hệ số z^m trong mẫu số của biến đổi Z của hàm truyền. Nói cách khác, trễ làm xuất hiện một tập m điểm cực tại gốc tọa độ. Hơn nữa, trị riêng của hệ thời gian liên tục tại $s = -\lambda$ trở thành trị riêng của hệ n thời gian rời rạc tại $z = e^{-\lambda\Delta}$ (xem phương trình (24.125)). Kết quả cho hàm truyền có dạng:

$$\frac{Y(z)}{U(z)} = H(z) = \frac{K}{\lambda} \frac{1 - e^{-\lambda\Delta}}{z^m (z - e^{-\lambda\Delta})} \tag{24.127}$$

Và hàm truyền này có thể biểu diễn như mô hình không gian trạng thái rời rạc:

$$x_1[t+1] = x_2[t] \tag{24.128}$$

$$x_2[t+1] = x_3[t] \tag{24.129}$$

.....

$$x_m[t+1] = x_{m+1}[t] \tag{24.130}$$

$$x_{m+1}[t+1] = e^{-\lambda\Delta} x_{m+1}[t] + \frac{K}{\lambda}(1 - e^{-\lambda\Delta}) u[t] \tag{24.131}$$

$$y[t] = x_1[t] \tag{24.132}$$

Vì vậy chúng ta có thể xem các biến trạng thái $x_{m+1}[t],...,x_1[t]$ như nhiệt độ tại các điểm cách đều nhau, giữa nguồn nhiệt và cảm biến nhiệt.

Khi thời gian trễ τ không là bội số của chu kì lấy mẫu Δ, thì một điểm cực bổ sung ở hàm truyền ban đầu và một điểm "không" nữa xuất hiện ở hàm truyền rời rạc.

Biến đổi tương đương trạng thái

Ý tưởng biến đổi trạng thái thông qua một phép biến đổi tương đương cũng được áp dụng với các hệ rời rạc. Các thuộc tính của hệ thống vẫn không thay đổi.

Không gian trạng thái và các hàm truyền

Với các hệ thống rời rạc, mối quan hệ giữa các mô hình không gian trạng thái và mô hình hàm truyền về cơ bản là giống nhau như trong trường hợp thời gian liên tục (xem mục "Các hàm không gian trạng thái và hàm truyền"). Như chúng ta đã nói, mô tả không gian trạng thái của hệ tuyến tính bất biến là một cách mô tả khác tốt hơn cách được đưa ra bởi các hàm truyền, mặc dù trong một số trường hợp nó cung cấp nhiều thông tin hơn về hệ thống.

Cho hệ thống bất biến rời rạc tuyến tính với đầu vào $u[t] \in \mathbb{R}^m$ và đầu ra $y[t] \in \mathbb{R}^p$, hàm truyền, $H[z] \in \mathbb{C}^{p \times m}$, được định nghĩa bởi công thức:

$$\mathbf{Y}[z] = \mathbf{H}[z]\mathbf{U}[z], \qquad [\mathbf{H}[z]]_{i,j} = \frac{Y_i[z]}{U_j[z]} \tag{24.133}$$

ví dụ thành phần (i, j) trong ma trận $\mathbf{H}(z)$ là biến đổi Z của đáp ứng ở đầu ra thứ i khi một hàm đenta Kronecker đơn vị được đưa vào đầu vào thứ j, với điều kiện đầu bằng không và các đầu vào còn lại bằng không với $t \geq 0$.

Mặt khác, nếu ta dùng biến đổi Z với mô hình không gian trạng thái rời rạc (24.100) và (24.101) với điều kiện ban đầu bằng không, chúng ta có:

$$\mathbf{X}[z] = (z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d\mathbf{U}[z] \tag{24.134}$$

$$\mathbf{Y}[z] = \mathbf{C}_d\mathbf{X}[z] + \mathbf{D}_d\mathbf{U}[z] \tag{24.135}$$

dẫn tới:

$$\mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d = \mathbf{H}[z] \tag{24.136}$$

Trong phân tích sau đây, chúng ta sẽ tập trung vào lớp các hệ vô hướng, có nghĩa là $m = p = 1, \mathbf{B}_d, \mathbf{C}_d^{\mathrm{T}}$ là các véctơ cột và $D_d = H[\infty]$. Khi đó chúng ta có thể thấy $H[z]$ là thương của các đa thức theo z, có nghĩa là:

$$H[z] = \frac{\mathbf{C}_d Adj(z\mathbf{I} - \mathbf{A}_d)\mathbf{B}_d + \mathbf{D}_d \det(z\mathbf{I} - \mathbf{A}_d)}{\det(z\mathbf{I} - \mathbf{A}_d)} \tag{24.137}$$

trong đó Adj(o) là ma trận liên hợp của (o).

Tương tự trường hợp thời gian liên tục, ta lại có các điểm cực của hàm truyền là các trị riêng của $\mathbf{A}_d$. Tuy nhiên, nói chung là không đúng để tập các điểm cực của hàm truyền là đồng nhất với tập các trị riêng của ma trận. Nó là quan trọng để thực hiện các mô hình hàm truyền có thể giấu đi các sự triệt tiêu giữa các điểm cực và điểm không với các hệ quả đã được mô tả trong các mục con "Tính điều khiển được, đạt được và ổn định" và "Tính quan sát, khôi phục được và phát hiện được".

Kết quả quan trọng cho hệ thống rời rạc cũng giống như hệ thống liên tục là: **hàm truyền có thể không cung cấp cùng một lượng thông tin như mô hình không gian trạng thái** cho cùng một hệ thống.

Một cách để có được mô hình không gian trạng thái là sử dụng phương pháp được đề cập trong phần "Không gian trạng thái và các hàm truyền", bằng cách sử dụng biến đổi Z thay cho biến đổi Laplace và sử dụng:

$$F[z] = Z\{f[t]\} \Leftrightarrow zF[z] = Z\{f[t+1]\} \tag{24 138}$$

Ví dụ 24.10

Cho hàm truyền của hệ thống:

$$H[z] = \frac{2z^2 - z + 1}{(z-0.8)(z-0.6)} = \frac{1.8z + 0.04}{z^2 - 1.4z + 0.48} + 2 \tag{24.139}$$

Khi đó thực hiện tối thiểu cho hệ thống này là:

$$\mathbf{A}_d = \begin{bmatrix} 0 & 1 \\ -048 & -1.4 \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{24.140}$$

$$\mathbf{C}_d = [0.04 \quad 1.8], \quad \mathbf{D}_d = 2 \tag{24.141}$$

Trong các mô hình rời rạc, cũng xảy ra **hàm truyền của hệ thống là bất biến với biến đổi tương đương trạng thái.**

24.5 Các mô hình không gian trạng thái cho các hệ thống kết nối

Để xây dựng các mô hình không gian trạng thái cho các hệ thống phức tạp, đôi khi có thể biểu diễn chúng như là sự kết nối giữa các hệ thống đơn giản. Sự kết nối này thường là sự kết hợp của ba cấu trúc kết nối cơ bản: nối tiếp, song song và phản hồi. Trong ba trường hợp cơ bản đó, mục đích của chúng ta là làm sao để nhận được mô hình không gian trạng thái cho hệ thống hỗn hợp.

Trong phân tích dưới đây, chúng ta sẽ sử dụng hai hệ thống, được cho bởi:

$$\text{Hệ thống 1: } \frac{d\mathbf{x}_1(t)}{dt} = \mathbf{A}_1\mathbf{x}_1(t) + \mathbf{B}_1\mathbf{u}_1(t) \tag{24.142}$$

$$\mathbf{y}_1(t) = \mathbf{C}_1\mathbf{x}_1(t) + \mathbf{D}_1\mathbf{u}_1(t) \tag{24.143}$$

$$\text{Hệ thống 2: } \frac{d\mathbf{x}_2(t)}{dt} = \mathbf{A}_2\mathbf{x}_2(t) + \mathbf{B}_2\mathbf{u}_2(t) \tag{24.144}$$

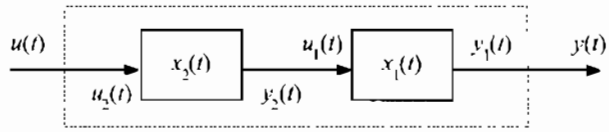
$$\mathbf{y}_2(t) = \mathbf{C}_2\mathbf{x}_2(t) + \mathbf{D}_2\mathbf{u}_2(t) \tag{24.145}$$

Kết nối nối tiếp

Hệ thống kết nối như biểu diễn trong hình 24.10 được biết như là một sự kết nối nối tiếp. Để xây dựng được mô hình không gian trạng thái mong muốn, trước tiên chúng ta quan sát thấy $\mathbf{y}_2(t) = \mathbf{u}_1(t)$. Cũng như vậy, đầu vào hỗn hợp của hệ thống là $\mathbf{u}(t) = \mathbf{u}_2(t)$, và đầu ra hỗn hợp của hệ thống là $\mathbf{y}(t) = \mathbf{y}_1(t)$. Nhờ vậy ta có:

$$\begin{bmatrix} \dot{\mathbf{x}}_1(t) \\ \dot{\mathbf{x}}_2(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1\mathbf{C}_2 \\ 0 & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1\mathbf{D}_2 \\ \mathbf{B}_2 \end{bmatrix} \mathbf{u}(t) \tag{24.146}$$

$$\mathbf{y}(t) = [\mathbf{C}_1 \quad \mathbf{D}_1\mathbf{C}_2] \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} + [\mathbf{D}_1\mathbf{D}_2]\mathbf{u}(t) \tag{24.147}$$



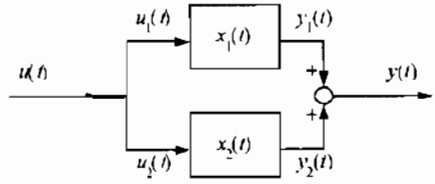
HÌNH 24.10 Kết nối nối tiếp

Kết nối song song

Kết nối hệ thống được chỉ ra trên hình 24.11 được biết như là kết nối song song. Để nhận được mô hình không gian trạng thái mong muốn, ta thấy rằng đầu vào là $\mathbf{u}(t) = \mathbf{u}_1(t) = \mathbf{u}_2(t)$ và đầu ra cho toàn bộ hệ thống là $\mathbf{y}(t) = \mathbf{y}_1(t) + \mathbf{y}_2(0)$. Ta có:

$$\begin{bmatrix} \dot{\mathbf{x}}_1(t) \\ \dot{\mathbf{x}}_2(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \mathbf{u}(t) \tag{24.148}$$

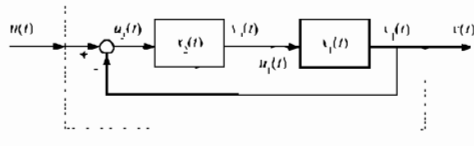
$$\mathbf{y}(t) = [\mathbf{C}_1 \quad \mathbf{C}_2] \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} + [\mathbf{D}_1 + \mathbf{D}_2]\mathbf{u}(t) \tag{24.149}$$



HÌNH 24.11 Kết nối song song

Kết nối phản hồi

Kết nối hệ thống được biểu diễn trên hình 24.12 là kết nối phản hồi (với phản hồi âm), tương ứng với cấu trúc cơ bản của một vòng điều khiển, trong đó S_1 là *đối tượng điều khiển* và S_2 là *bộ điều khiển*. Để xây dựng mô hình không gian trạng thái hỗn hợp chúng ta thấy rằng đầu vào của toàn hệ thống thỏa mãn phương trình $\mathbf{u}(t) = \mathbf{u}_2(t) + \mathbf{y}_1(t)$, và đầu ra của toàn hệ thống là $\mathbf{y}(t) = \mathbf{y}_1(t)$. Hơn nữa, chúng ta giả thiết rằng S_1 (đối tượng điều khiển) là hoàn toàn chính xác, có nghĩa là, $\mathbf{D}_1 = 0$. Từ đó ta có:



HÌNH 24.12 Kết nối phản hồi

$$\begin{bmatrix} \dot{\mathbf{x}}_1(t) \\ \dot{\mathbf{x}}_2(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 - \mathbf{B}_1\mathbf{D}_2\mathbf{C}_1 & \mathbf{B}_1\mathbf{C}_2 \\ \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1\mathbf{D}_2 \\ \mathbf{B}_2 \end{bmatrix} u(t) \tag{24.150}$$

$$\mathbf{y}(t) = [\mathbf{C}_1 \quad \mathbf{0}] \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} \tag{24.151}$$

Áp dụng cùng các kết quả, với những sửa đổi thích hợp, tới các hệ thời gian rời rạc kết nối. Các chi tiết hơn có thể tìm thấy ở những phần khác như trong [15].

24.6 Các thuộc tính của hệ thống

Tính điều khiển được, đạt được và ổn định

Một vấn đề rất quan trọng mà chúng ta phải quan tâm liên quan đến hệ thống điều khiển sử dụng các mô hình không gian trạng thái là chúng ta có thể lái hệ thống bởi đầu vào điều khiển tới một vị trí nhất định trong không gian trạng thái được hay không. Ta phải nhớ rằng các trạng thái của hệ thống thường là các biến nội tại như nhiệt độ, áp suất, mức của thùng chứa... Đôi khi, những biến đó là các biến tới hạn mà chúng ta muốn giữ giá trị của biến giữa các giá trị xác định.

Tính điều khiển được

Tính điều khiển được liên quan tới trạng thái đầu $\mathbf{x}_0$ có thể được lái tới điểm gốc trong thời gian hữu hạn bằng cách sử dụng đầu vào $\mathbf{u}(t)$ được hay không.

Ví dụ 24.11

Nếu ta xét mô hình đã được xác định trong (24.152), ta chú ý rằng đầu vào $u(t)$ không có tác động tới trạng thái $x_2(t)$:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \tag{24.152}$$

Cho trạng thái ban đầu $[x_1(0), x_2(0)]^T$, đầu vào $u(t)$ có thể được chọn để lái $x_1(t)$ về không, trong khi $x_2(t)$ không thay đổi.

Chúng ta có định nghĩa sau:

Định nghĩa 24.1 Trạng thái x_0 được gọi là **có thể điều khiển được** nếu tồn tại một khoảng hữu hạn $[0, T]$ và một đầu vào $\{u(t), t \in [0, T]\}$ sao cho $x(T) = 0$. Nếu tất cả các trạng thái đều có thể điều khiển được, thì hệ thống được gọi là **có thể điều khiển được hoàn toàn**.

Tính đạt được

Một khái niệm liên quan đó là **tính đạt được**, đôi khi được sử dụng trong các hệ thống rời rạc. Nó được định nghĩa một cách chính thức như sau:

Định nghĩa 24.2 Một trạng thái $\bar{\mathbf{x}} \neq 0$ được gọi là **có thể đạt được**, từ điểm gốc, nếu cho $\mathbf{x}(0) = 0$, tồn tại một khoảng thời gian hữu hạn $[0, T]$ và một đầu vào $\{\mathbf{u}(t), t \in [0, T]\}$ sao cho $\mathbf{x}(T) = \bar{\mathbf{x}}$. Nếu tất cả các trạng thái đều có thể đạt được thì hệ thống được gọi là **có thể đạt được hoàn toàn**.

Với các hệ tuyến tính bất biến liên tục, không có sự khác biệt giữa khả năng điều khiển được hoàn toàn và khả năng đạt được. Tuy nhiên, ví dụ sau đây minh họa rằng có một chút khác nhau trong các trường hợp thời gian rời rạc. Xét hệ thống và đầu ra:

$$\mathbf{x}[t+1] = \underbrace{\begin{bmatrix} 0.5 & 1 \\ -0.25 & -0.5 \end{bmatrix}}_{A} \mathbf{x}[t] \Rightarrow \mathbf{x}[t] = \begin{bmatrix} 0.5 & 1 \\ -0.25 & -0.5 \end{bmatrix}^t \mathbf{x}[0] \tag{24.153}$$

Chúng ta có thể thấy rằng hệ thống này hoàn toàn điều khiển được khi $\mathbf{x}[t] = 0$, $\forall t \geq 2$ và $\forall \mathbf{x}[0] \in \mathbb{R}^2$. Điều này chỉ ra rằng tất cả các trạng thái ban đầu đều điều khiển được. Tuy nhiên, không có trạng thái khác không nào là có thể đạt được.

Thông thường, trong các hệ tuyến tính bất biến, tính điều khiển được và tính đạt được có thể được sử dụng thay thế cho nhau.

Kiểm tra tính điều khiển được

Bây giờ chúng ta sẽ trình bày một cách phân loại để xác định tính điều khiển được hoàn toàn của một hệ thống.

Định lý 24.3 *Xét mô hình không gian trạng thái tuyến tính bất biến trong đó* $A \in \mathbb{R}^{n,n}$ *:*

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{24.154}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{24.155}$$

Tập hợp tất cả các trạng thái điều khiển được là ***không gian hàng*** *của ma trận điều khiển được* $\Gamma_c[\mathbf{A},\mathbf{B}]$ *trong đó:*

$$\Gamma_c[\mathbf{A},\mathbf{B}] \triangleq [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}] \tag{24.156}$$

Mô hình có thể điều khiển được hoàn toàn nếu và chỉ nếu $\Gamma_c[\mathrm{A},\mathrm{B}]$ *có* ***hạng cột đầy đủ.***

Ví dụ 24.12

Xét mô hình không gian trạng thái đưa ra trong (24.152), với các ma trận không gian trạng thái:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{24.157}$$

Ma trận có thể điều khiển được cho hệ thống này được cho bởi:

$$\Gamma_c[\mathbf{A}, \quad \mathbf{B}] = [\mathbf{B} \quad \mathbf{AB}] = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{24.158}$$

Rõ ràng, hạng của $\Gamma_c[\mathbf{A}, \quad \mathbf{B}] = 1$, do đó hệ thống là **không** thể điều khiển được hoàn toàn.

Kết quả trên đây áp dụng cho mô hình liên tục, và nó cũng như vậy với tính đạt được của các mô hình rời rạc.

Chúng ta cũng có thể thấy rằng tính điều khiển được của hệ thống không phụ thuộc vào việc chọn các biến trạng thái. Để thấy điều này, xét một phép biến đổi tương đương được định nghĩa trong mục nhỏ "Biến đổi tương đương trạng thái". Khi đó, thấy rằng $\overline{\mathbf{A}}' = \mathbf{T}^{-1}\mathbf{A}'\mathbf{T}$, chúng ta có:

$$\Gamma_c[\overline{\mathbf{A}}, \quad \overline{\mathbf{B}}] = \mathbf{T}^{-1}\Gamma_c[\mathbf{A}, \quad \mathbf{B}] \tag{24.159}$$

điều đó chỉ ra rằng $\Gamma_c[\overline{\mathbf{A}}, \quad \overline{\mathbf{B}}]$ và $\Gamma_c[\mathbf{A}, \quad \mathbf{B}]$ có cùng hạng.

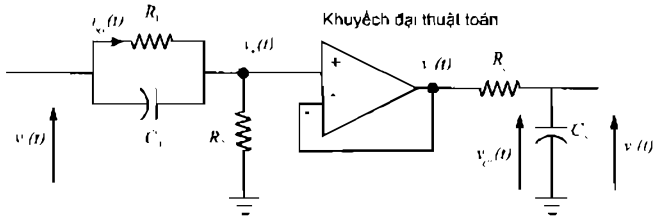
Bạn đọc có thể kiểm tra thấy mô hình không gian trạng thái được sử dụng để mô tả các tín hiệu trong phần "Tín hiệu và mô tả không gian trạng thái" là không thể điều khiển được. Thực vậy, luôn đúng rằng **bất kỳ mô hình không gian trạng thái nào có B = 0 đều không thể điều khiển được hoàn toàn.**

Mất tính điều khiển được

Sự thiếu tính điều khiển được đôi khi là một đặc tính của cấu trúc. Tuy nhiên, trong một số trường hợp, nó phụ thuộc vào giá trị của các thông số nhất định. Ta sẽ minh họa điều này trong ví dụ sau.

Ví dụ 24.13

Xét mạch điện tử trong hình 24.13.



HÌNH 24.13 Mạch điện tử

Trước tiên ta xây dựng mô hình không gian trạng thái cho mạch. Chọn $x_1(t) = i_{R1}(t)$ và $x_2(t) = v_{C3}(t)$ như là các biến trạng thái. Bằng cách dùng nguyên tắc đầu tiên trên nửa trái của mạch, ta có:

$$i_{C1} = C_1 \frac{d}{dt}(v_i - v_+), \quad i_{R1} = \frac{(v_i - v_+)}{R_1}, \quad i_{R2} = \frac{v_+}{R_2}, \; i_{C1} = i_{R2} - i_{R1} \tag{24.160}$$

Suy ra:

$$\frac{di_{R1}(t)}{dt} = -\frac{(R_1 + R_2)}{C_1 R_1 R_2} i_{R1}(t) + \frac{1}{C_1 R_1 R_2} v_i(t) \tag{24.161}$$

$$v_+(t) = -R_1 i_{R1}(t) + v_i(t) \tag{24.162}$$

Tương tự, từ nửa phải của mạch ta có:

$$\frac{dv_{C3}(t)}{dt} = -\frac{1}{R_3 C_3} v_{C3}(t) + \frac{1}{R_3 C_3} v_-(t) \tag{24.163}$$

$$v_0(t) = v_{C3}(t) \tag{24.164}$$

Bộ khuyếch đại thuật toán (lý tưởng) đảm bảo rằng $v_+(t) = v_-(t)$, vì vậy chúng ta có thể tổ hợp các mô hình không gian trạng thái đã cho trong (24.161) – (24.164) để nhận được:

$$\begin{bmatrix} \dfrac{di_{R1}(t)}{dt} \\ \dfrac{dv_{C3}(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\dfrac{(R_1 + R_2)}{C_1 R_1 R_2} & 0 \\ -\dfrac{R_1}{R_3 C_3} & -\dfrac{1}{R_3 C_3} \end{bmatrix} \begin{bmatrix} i_{R1}(t) \\ v_{C3}(t) \end{bmatrix} + \begin{bmatrix} \dfrac{1}{C_1 R_1 R_2} \\ \dfrac{1}{R_3 C_3} \end{bmatrix} v_i(t) \tag{24.165}$$

$$v_0(t) = [0 \quad 1] \begin{bmatrix} i_{R1}(t) \\ v_{C3}(t) \end{bmatrix} \tag{24.166}$$

Vì vậy ta có ma trận điều khiển được:

$$\Gamma_c[\mathbf{A}, \mathbf{B}] = [\mathbf{B} \quad \mathbf{AB}] = \begin{bmatrix} \dfrac{1}{R_1 R_2 C_1} & -\dfrac{(R_1 + R_2)}{(R_1 R_2 C_1)^2} \\ \dfrac{1}{R_3 C_3} & -\dfrac{(R_2 C_1 + R_1 C_3)}{(R_3 C_3)^2 R_2 C_1} \end{bmatrix} \tag{24.167}$$

và:

$$\det(\Gamma_c[\mathbf{A},\mathbf{B}]) = \frac{R_2}{(R_1R_2R_3C_1C_2)^2}(-R_1C_1 + R_3C_3) \tag{24.168}$$

ở đây chúng ta có thể thấy rằng hệ thống hoàn toàn điều khiển được nếu và chỉ nếu $R_1C_1 \neq R_3C_3$.

Điều này có một giải thích rất quan trọng nếu ta phân tích trên phương diện hàm truyền. Việc áp dụng biến đổi Laplace cho các phương trình (24.161) – (24.164), hàm truyền từ $v_i(t)$ tới $v_0(t)$ (cần nhớ là $V_i(t) = V(t)$) được cho bởi:

$$\frac{v_0(s)}{v_i(s)} = \frac{v_0(s)v_+(s)}{v_-(s)v_i(s)} = \frac{\dfrac{1}{R_3C_3}}{s + \dfrac{1}{R_3C_3}} \cdot \frac{s + \dfrac{1}{R_1C_1}}{s + \dfrac{R_1 + R_2}{R_1R_2C_1}} \tag{24.169}$$

ở đây, ta có thể thấy rằng sự mất tính điều khiển được hoàn toàn, khi $R_1C_1 \neq R_3C_3$, có được từ (24.168) nghĩa là có một **sự triệt tiêu điểm không - cực** trong hàm truyền, có nghĩa là, điểm không từ nửa trái của mạch trong hình (24.13) bị triệt tiêu bởi điểm cực từ phần kia của mạch. Vấn đề này sẽ được phân tích cụ thể trong mục "Khai triển chính tắc".

Gramian của tính điều khiển được

Kiểm tra tính điều khiển được cho ta câu trả lời *có* hay *không* về tính điều khiển được của mô hình hệ thống. Tuy nhiên, kết luận một hệ thống điều khiển được hoàn toàn không cho biết về *mức độ* của tính điều khiển được. Với hệ thống **ổn định**, chúng ta có thể xác định mức độ điều khiển trạng thái hệ thống qua năng lượng của tín hiệu đầu vào $\mathbf{u}(t)$ đã gây ra từ $t = -\infty$ để tiến tới trạng thái $x(0) = x_0$ tại $t = 0$:

$$J(u) = \int_{-\infty}^{0} \|u(t)\|^2 \, dt = \int_{-\infty}^{0} u(t)^T u(t) dt \tag{24.170}$$

Có thể thấy *năng lượng điều khiển* tối thiểu là:

$$J(u_{min}) = x_0^T P^{-1} x_0 \tag{24.171}$$

trong đó:

$$P = \int_0^{\infty} e^{At} BB^T e^{A^T t} dt \tag{24.172}$$

Ma trận **P** được gọi là **gramian của tính điều khiển được**, và nó đo lường khả năng điều khiển được của vectơ trạng thái $\mathbf{x}(0)$. Nếu ma trận này *nhỏ*, nghĩa là ta cần nhiều năng lượng ở đầu vào điều khiển $\mathbf{u}(t)$ để lái vectơ trạng thái đến $\mathbf{x}_0$. Thực vậy, ta có thể ước lượng được tác động cần thiết cho mỗi một trong các biến trạng thái, bằng cách tạo ra, ví dụ $x_0 = [0, \ \ldots, \ 0, \ 1, \ 0, \ \ldots, \ 0]^T$.

Cần nhấn mạnh rằng sự tồn tại của tích phân được định nghĩa ở (24.172) chỉ được đảm bảo nếu trị riêng của **A** có phần thực âm, tức là hệ thống phải được ổn định.

Cũng như vậy, gramian P của tính điều khiển được được định nghĩa ở (24.172) thỏa mãn phương trình Lyapunov:

$$AP + PA^T + BB^T = 0 \tag{24.173}$$

Với các hệ rời rạc chúng ta có các phương trình sau cho gramian của tính điều khiển được:

$$P_d = \sum_{k=0}^{\infty} A_d^k B_d B_d^k (A_d^T)^k \tag{24.174}$$

nó thỏa mãn:

$$A_d B_d A_d^T - P_d + B_d B_d^T = 0 \tag{24.175}$$

Tổng được định nghĩa trong (24.174) bị chặn nếu và chỉ nếu hệ thống rời rạc là ổn định, ví dụ trị riêng của nó nằm bên trong đường đơn vị.

Ví dụ 24.14

Ta có thể phân tích mô hình của ví dụ 24.13, mạch điện tử được mô tả bởi mô hình không gian trạng thái (24.165) và (2.166). Nếu chúng ta muốn đánh giá thông tin có được từ gramian của tính điều khiển được, được định nghĩa trong (24.172), khi mô hình gần như mất khả năng điều khiển được hoàn toàn, thì chúng ta có thể chọn các giá trị phù hợp cho các thông số để đảm bảo $R_1C_1 \neq R_3C_3$.

Nếu ta chọn:

$$R_1 = R_2 = R_3 = 10^3\,\Omega \quad C_1 = 0.9\times 10^3\,\mu F \tag{24.176}$$

mô hình sẽ được mô tả bởi:

$$\begin{bmatrix} i_{R_1}(t) \\ v_{C_1}(t) \end{bmatrix} = \begin{bmatrix} -\frac{20}{9} & 0 \\ -10^3 & -1 \end{bmatrix} \begin{bmatrix} i_{R_1}(t) \\ v_{C_1}(t) \end{bmatrix} + \begin{bmatrix} \frac{0.01}{9} \\ 1 \end{bmatrix} v_i(t) \tag{24.177}$$

$$v_0(t) = [0 \quad 1] \begin{bmatrix} i_{R_1}(t) \\ v_{C_1}(t) \end{bmatrix} \tag{24.178}$$

Nếu ta xét độ lớn tương đối của các phần tử của **B**, ta có thể có *quyền* nói rằng tác động của đầu vào $u(t)$ lên trạng thái $i_{R1}(t)$ sẽ yếu hơn nhiều so với tác động lên trạng thái $v_{C3}(t)$. Để kiểm tra điều này, ta có thể tính gramian của tính điều khiển được được định nghĩa ở (24.172), bằng cách giải:

$$0 = AP + PA^T + BB^T \tag{24.179}$$

$$0 = \begin{bmatrix} -\frac{20}{9} & 0 \\ -10^3 & -1 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} -\frac{20}{9} & -10^3 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} \frac{0.01}{9} \\ 1 \end{bmatrix} \begin{bmatrix} \frac{0.01}{9} & 1 \end{bmatrix} \tag{24.180}$$

Ta có:

$$= \begin{bmatrix} 0.28\times 10^{-6} & 0.000258620 \\ 0.000258620 & 0.99999948 \end{bmatrix}, \quad P^{-1} = \begin{bmatrix} 4736624.0 & -1224.9 \\ -1224.9 & 1.3 \end{bmatrix} \tag{24.181}$$

Vì vậy ta có thể nhận được năng lượng điều khiển tối thiểu để lái trạng thái $x(t)$, từ 0 ở $t = -\infty$ đến x_0 ở $t = 0$, từ phương trình (24.171).

$$x_0 = [1,0]^T \Rightarrow J(u_{opt}) = 4736624.0 \tag{24.182}$$

$$x_0 = [1,0]^T \Rightarrow J(u_{opt}) = 1.3 \tag{24.183}$$

Do đó ta có thể xác minh rằng, năng lượng điều khiển để đạt được $i_{k1}(0) = 1$ là bậc sáu của độ lớn hơn so với năng lượng cần thiết để đạt được $v_{c3}(0) = 1$.

Ngoài ra, nếu ta thay các giá trị của tham số trong phương trình (26.169), chúng ta có hàm truyền cho bởi:

$$\frac{V_0(s)}{V_i(s)} = \frac{1}{s+1} \cdot \frac{s+1+\frac{1}{9}}{s+\frac{20}{95}} \tag{24.184}$$

Từ đó chúng ta thấy được một sự triệt tiêu **điểm không - cực giả**.

Gramian đã được triển khai cho cả trường hợp không ổn định, xem [16].

Khai triển chính tắc và tính ổn định

Nếu một hệ thống không thể điều khiển được hoàn toàn, nó có thể khai triển thành một hệ thống con có thể điều khiển được và một hệ thống con không thể điều khiển hoàn toàn được theo cách sau.

Bổ đề 24.1 *Xét hệ thống có hạng* $\{\Gamma_c[A, B]\} = k < n$. Khi đó tồn *tại một biến đổi tương đương* ***T*** *sao cho* $\bar{x} = T^{-1}x$:

$$\bar{A} = T^{-1}AT \qquad \bar{B} = T^{-1}B \tag{24.185}$$

$\bar{A}$ *và* $\bar{B}$ *có dạng*:

$$\bar{A} = \begin{bmatrix} \bar{A}_c & \bar{A}_{12} \\ 0 & \bar{A}_{nc} \end{bmatrix}, \qquad \bar{B} = \begin{bmatrix} \bar{B}_c \\ 0 \end{bmatrix} \tag{24.186}$$

trong đó $\bar{A}_c$ *có k chiều và* $(\bar{A}_c, \bar{B}_c)$ *là có thể điều khiển được hoàn toàn.*

Kết quả trên đây cho thấy trạng thái nào ta có thể hoặc không thể lái tới không. Để đánh giá điều này, ta biểu diễn các phương trình trạng thái và đầu ra dưới dạng:

$$\begin{bmatrix} \dot{\bar{x}}_c \\ \dot{\bar{x}}_{nc} \end{bmatrix} = \begin{bmatrix} \bar{A}_c & \bar{A}_{12} \\ 0 & \bar{A}_{nc} \end{bmatrix} \begin{bmatrix} \bar{x}_c \\ \bar{x}_{nc} \end{bmatrix} + \begin{bmatrix} \bar{B}_c \\ 0 \end{bmatrix} u \tag{24.187}$$

$$y = \begin{bmatrix} \bar{C}_c & \bar{C}_{nc} \end{bmatrix} \begin{bmatrix} \bar{x}_c \\ \bar{x}_{nc} \end{bmatrix} + Du \tag{24.188}$$

Không gian con có thể điều khiển được của mô hình không gian trạng thái bao gồm tất cả các trạng thái được tạo ra thông qua mỗi tổ hợp tuyến tính có thể của các trạng thái trong $\bar{x}_c$. Sự ổn định của không gian con này được xác định bởi vị trí của các trị riêng của $\bar{A}_c$.

Mặt khác, **không gian con không thể điều khiển được** bao gồm tất cả các trạng thái được tạo ra thông qua mỗi tổ hợp tuyến tính có thể của các trạng thái trong $\bar{x}_{nc}$. Sự ổn định của không gian con này được xác định bởi vị trí của trị riêng của $\bar{A}_{nc}$.

Vì vậy, đầu vào sẽ không có tác động nào trên không gian con không điều khiển được, mong muốn của chúng ta là không gian con không điều khiển được này ổn định, trạng thái của nó sẽ tiến tới điểm ban đầu. Trong trường hợp này mô hình không gian trạng thái được gọi là **có thể ổn định được**.

Một đặc tính quan trọng của các mô tả (24.187) và (24.188) phát sinh từ thực tế là hàm truyền được cho bởi:

$$H(s) = \overline{C}_c (sI - \overline{A}_c)^{-1} \overline{B}_c + D \tag{24.189}$$

Phương trình (24.189) nói rằng các trị riêng của không gian con không điều khiển được không phụ thuộc vào tập các điểm cực của hàm truyền. Điều này chỉ ra rằng có một sự triệt tiêu tất cả các điểm cực tương ứng với các nghiệm của ($s\mathbf{I}$ - $\overline{A}_{nc}$).

Dạng chính tắc của tính điều khiển được

Bổ đề 24.2 *Xét mô hình không gian trạng thái có thể đạt được hoàn toàn cho một hệ SISO. Khi đó tồn tại một biến đổi tương đương, biến đổi mô hình không gian trạng thái sang* ***dạng chính tắc tính điều khiển được*** *như sau:*

$$A' = \begin{bmatrix} 0 & 0 & \dots & 0 & -\alpha_0 \\ 1 & 0 & \dots & 0 & -\alpha_1 \\ 0 & 1 & \dots & 0 & -\alpha_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\alpha_{n-1} \end{bmatrix}, \quad B' = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{24.190}$$

trong đó $\lambda^n + \alpha_{n-1}\lambda^{n-1} + \cdots + \alpha_1\lambda + \alpha_0 = \det(\lambda I - A)$ là đa thức riêng của **A**.

Bổ đề 24.3 Xét *mô hình không gian trạng thái có thể điều khiển được hoàn toàn cho một hệ SISO. Khi đó tồn tại một biến đổi tương đương, biến đổi mô hình không gian trạng thái thành* ***dạng chính tắc của bộ điều khiển*** *như sau:*

$$A' = \begin{bmatrix} -\alpha_{n-1} & -\alpha_{n-2} & \dots & -\alpha_1 & -\alpha_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad B' = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{24.191}$$

trong đó $\lambda^n + \alpha_{n-1}\lambda^{n-1} + \cdots + \alpha_1\lambda + \alpha_0 = \det(\lambda I - A)$ *là đa thức riêng của A.*

Tính quan sát được, tính khôi phục được, và tính xác định được

Nếu ta xét mô hình không gian trạng thái của một hệ thống, có thể phỏng đoán rằng nếu quan sát đầu ra trong một số quãng thời gian thì có thể biết được một số thông tin về trạng thái. Thuộc tính này của mô hình được gọi là tính quan sát được (hoặc tính khôi phục được).

Tính quan sát được

Tính quan sát được liên quan đến điều có thể nói về trạng thái nếu ta đo đầu ra đối tượng điều khiển.

Ví dụ 24.15

Nếu ta xét hệ thống định nghĩa bởi mô hình không gian trạng thái:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \qquad y(t) = [1 \quad 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{24.192}$$

Ta có thể thấy đầu ra $y(t)$ chỉ được xác định bởi $x_1(t)$, và biến trạng thái $x_2(t)$ không có ảnh hưởng đến đầu ra. Vì vậy hệ thống không là không thể quan sát được hoàn toàn.

Ta có định nghĩa sau:

Định nghĩa 24.3 Trạng thái $x_0 \neq 0$ được gọi là **không thể quan sát** được nếu cho $x(0) = x_0$ và $u(t) = 0$ với $t \geq 0$ thì có $y(t) = 0$ với $t \geq 0$, có nghĩa là, ta không thể thấy bất kỳ tác động nào của x_0 trên đầu ra hệ thống.

Hệ thống được gọi là **có thể quan sát được hoàn toàn** nếu không tồn tại trạng thái đầu khác không nào mà không quan sát được.

Tính khôi phục được

Có một khái niệm khác, liên quan mật thiết với quan sát được, được gọi là **tính khôi phục được**. Tính khôi phục được liên quan đến điều có thể nói về $x(T)$, nếu quan sát được các giá trị **quá khứ** của đầu ra **y**, với $0 \leq t \leq T$. Với các hệ thống liên tục, tuyến tính bất biến, sự khác biệt giữa khả năng quan sát được và khả năng khôi phục được là không cần thiết. Tuy nhiên, ví dụ minh họa sau đây cho thấy trong thời gian rời rạc, hai khái niệm đó là khác nhau. Xét:

$$x[t+1] = 0, \quad x[0] = x_0 \tag{24.193}$$

$$y[t] = 0 \tag{24.194}$$

Hệ thống này rõ ràng có thể khôi phục được cho mọi $T \geq 1$, vì ta biết chắc chắn rằng $x[T] = 0$ với $T \geq 1$. Tuy nhiên, nó không quan sát được hoàn toàn vì $y[t] = 0$, $\forall k$ mà không phụ thuộc vào x_0.

Kiểm tra tính quan sát được

Việc kiểm tra tính quan sát được của một hệ thống tuân theo định lý sau.

Định lý 24.4 *Xét mô hình không gian trạng thái tuyến tính, liên tục, bất biến với* $A \in \mathbb{R}^{n \times n}$:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{24.195}$$

$$y(t) = Cx(t) + Du(t) \tag{24.196}$$

Tập hợp tất cả các trạng thái không quan sát được là không gian rỗng của ma trận quan sát được $\Gamma_0[A,C]$ *trong đó*:

$$\Gamma_0[A,C] \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{24.197}$$

Hệ thống hoàn toàn quan sát được nếu và chỉ nếu $\Gamma_0[A,C]$ *có hạng cột n đầy đủ.*

Ví dụ 24.16

Xét mô hình không gian trạng thái sau:

$$A = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [1 \quad -1] \tag{24.198}$$

Ma trận quan sát được được cho bởi:

$$\Gamma_0[A, \quad C] = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -4 & -2 \end{bmatrix} \tag{24.199}$$

Vì vậy hạng $\Gamma_0[A, \quad C] = 2$, cho thấy hệ thống là có thể quan sát được hoàn toàn.

Ví dụ 24.17

Xét mô hình đã được định nghĩa trong (24.192), ta có:

$$A = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}, \quad C = [1 \quad 0] \tag{24.200}$$

Ma trận có khả năng quan sát được là:

$$\Gamma_0[A, \quad C] = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \tag{24.201}$$

Vì vậy hạng $\Gamma_0[A,C] = 1 < 2$ và hệ thống là không thể quan sát được hoàn toàn.

Kết quả trên đây cũng áp dụng cho các mô hình rời rạc.

Tính quan sát được của hệ thống không phụ thuộc vào sự lựa chọn các biến trạng thái. Có thể chứng minh rằng hạng của ma trận được định nghĩa trong phương trình (24.197) không thay đổi khi biến đổi tương đương $\boldsymbol{T}$ được sử dụng (xem mục "Biến đổi tương đương trạng thái").

Mất tính quan sát được

Sự mất tính quan sát được có thể xuất phát từ các các đặc tính cấu trúc của hệ thống. Tuy nhiên, nó cũng có thể xảy ra khi các thông số hệ thống nào đó nhận một vài giá trị đặc biệt. Cũng giống như tính điều khiển được đã được phân tích trong phần "Tính điều khiển được, quan sát được và tính ổn định".

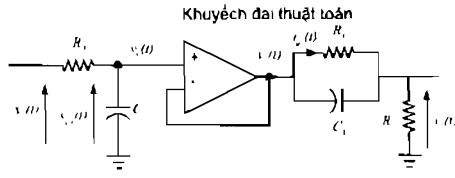
Ví dụ 24.18

Xét mạch điện tử trong hình 24.14. Ta có thể thấy giống như trong hình 24.13, trong đó phần trái và phải hoán đổi cho nhau, ta có thể sử dụng công thức tương tự để có mô hình không gian trạng thái. Các biến trạng thái được chọn là $x_1(t) = v_{C3}(t)$ và $x_2(t) = i_{R1}(t)$.

Với nửa trái của mạch, ta có:

$$\frac{dv_{C3}(t)}{dt} = -\frac{1}{R_3 C_3} v_{C3}(t) + \frac{1}{R_3 C_3} v_i(t) \tag{24.202}$$

$$v_x(t) = v_{C3}(t) \tag{24.203}$$



HÌNH 24.14 Mạch điện tử

Và với nửa phải, ta có:

$$\frac{di_{R1}(t)}{dt} = -\left(\frac{R_1 + R_1}{C_1 R_1 R_2}\right) i_{R1}(t) + \frac{1}{C_1 R_1 R_2} v_-(t) \tag{24.204}$$

$$v_0(t) = -R_1 i_{R1}(t) + v_-(t) \tag{24.205}$$

Bộ khuyếch đại thuật toán theo cách nối trên đảm bảo $v_+(t) = v_-(t)$, vì vậy ta có thể kết hợp các mô hình trạng thái đã đưa ra trong các phương trình (24.202) - (24.205):

$$\begin{bmatrix} \frac{dv_{C3}(t)}{dt} \\ \frac{di_{R1}(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_3 C_3} & 0 \\ \frac{1}{R_1 R_2 C_1} & -\frac{R_1 + R_2}{C_1 R_1 R_2} \end{bmatrix} \begin{bmatrix} v_{C3}(t) \\ i_{R1}(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{R_3 C_3} \\ 0 \end{bmatrix} v_i(t) \tag{24.206}$$

$$v_0(t) = [1 \quad -R_1] \begin{bmatrix} v_{C3}(t) \\ i_{R1}(t) \end{bmatrix} \tag{24.207}$$

Ma trận của khả năng quan sát được được cho bởi:

$$\Gamma_c[C, A] = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & -R_1 \\ -\frac{1}{R_3 C_3} & \frac{1}{R_2 C_1} \quad \frac{R_1 + R_2}{R_2 C_1} \end{bmatrix} \tag{24.208}$$

Để xác định khả năng quan sát được hoàn toàn ta cần tính định thức của ma trận:

$$\det(\Gamma_c[C, A]) = \frac{1}{R_3 C_3 C_1}(-R_1 C_1 + R_3 C_3) \tag{24.209}$$

từ đó ta kết luận rằng, hệ thống của mô hình này là có thể quan sát được hoàn toàn nếu và chỉ nếu, $R_1 C_1 \neq R_3 C_3$, giống với điều kiện ta thu được ở ví dụ 24.13.

Áp dụng biến đổi Laplace vào các phương trình (24.204) -- (24.203) ta có hàm truyền từ $V_i(s)$ tới $V_0(s)$:

$$\frac{V_0(s)}{V_i(s)} = \frac{V_-(s)}{V_i(s)} \frac{V_0(s)}{V_-(s)} = \frac{s + \frac{1}{R_1 C_1}}{s + \frac{R_1 + R_2}{R_1 R_2 C_1}} \cdot \frac{\frac{1}{R_3 C_3}}{s + \frac{1}{R_3 C_3}} \tag{24.210}$$

Điều kiện $R_1 C_1 = R_3 C_3$ tạo ra sự mất khả năng quan sát được, dẫn tới một **sự triệt tiêu điểm cực - không** trong mô hình hàm truyền, có nghĩa là, điểm cực ở nửa trái của mạch ở hình 24.14 bị triệt tiêu bởi điểm không ở nửa phải. Có một chút khác biệt giữa các hàm truyền trong (24.210) và (24.169). Kết quả cuối cùng là giống nhau, nhưng thứ tự sự triệt tiêu là khác nhau với mỗi trường hợp. **Sự triệt tiêu các điểm không - cực** liên quan đến sự mất tính quan sát được hoàn toàn và **sự triệt tiêu các điểm cực - không** liên quan đến sự mất tính điều khiển được hoàn toàn. Những vấn đề này sẽ được bàn một cách chi tiết hơn trong mục nhỏ "Khai triển chính tắc".

Gramian của tính quan sát được

Kiểm tra tính quan sát được theo định lý 24.4 sẽ trả lời "có" hay "không" về tính quan sát được hoàn toàn của một mô hình. Tuy nhiên, đôi khi ta quan tâm tới mức độ của tính quan sát được của một mô hình cụ thể. Từ đó ta có thể ước lượng năng lượng của tín hiệu đầu ra $y(t)$, khi không có đầu vào ($u(t) = 0$) và trạng thái là $x(0) = x_0$ tại $t = 0$:

$$E(x_0) = \int_0^\infty \|y(t)\|^2 \, dt = \int_0^T y(t)^T y(t) dt \tag{24.211}$$

Có thể chứng minh rằng *năng lượng đầu ra* là:

$$E(x_0) = \int_0^\infty \|y(t)\|^2 \, dt = x_0^T Q x_0 \tag{24.212}$$

trong đó:

$$Q = \int_0^\infty e^{A^T t} C^T C e^{At} dt \tag{24.213}$$

Ma trận **Q** được gọi là **gramian của tính quan sát được**, và nó đo lường tính quan sát được của vectơ trạng thái $x(0)$. Nếu ma trận này *nhỏ*, có nghĩa rằng chúng ta có một sự đóng góp yếu của trạng thái đầu x_0 đến năng lượng đầu ra $y(t)$. Thực vậy, ta có thể đánh giá tác động của mỗi biến trạng thái bằng cách lấy, ví dụ $x_0 = [0,\ldots,0,1,0,\ldots,0]^T$.

Chú ý rằng, sự tồn tại của tích phân được định nghĩa trong (24.213) được đảm bảo nếu và chỉ nếu hệ thống ổn định, có nghĩa là, nếu và chỉ nếu các trị riêng của **A** có phần thực âm.

Ngoài ra, gramian của tính quan sát được **Q** được định nghĩa trọng (24.213) thỏa mãn phương trình Lyapunov:

$$A^T Q + QA + C^T C = 0 \tag{24.214}$$

Với hệ thống rời rạc ổn định, gramian của tính điều khiển được được định nghĩa bởi:

$$Q_d = \sum_{k=0}^{\infty} (A_d^T)^k C_d^T C_d A_d^k \tag{24.215}$$

nó thỏa mãn:

$$A_d^T Q_d A_d - Q_d + C_d^T C_d = 0 \tag{24.216}$$

Ví dụ 24.19

Ta sẽ sử dụng mô hình trong ví dụ 24.18, được mô tả bởi mô hình không gian trạng thái (24.206) và (24.207), để đánh giá lợi ích của gramian của tính quan sát được (24.213), đặc biệt là khi mô hình gần mất tính quan sát được hoàn toàn, có nghĩa là, khi $R_1 C_1 \approx R_3 C_3$.

Giả thiết R_1, R_2, R_3, C_1, và C_3 có cùng các giá trị như trong ví dụ 24.14, chúng ta có:

$$\begin{bmatrix} \dot{v}_{C3}(t) \\ \dot{i}_{R1}(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 10^{-3} & -\dfrac{20}{9} \end{bmatrix} \begin{bmatrix} v_{C3}(t) \\ i_{R1}(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v_1(t) \tag{24.217}$$

$$v_0(t) = [1 \quad 10^{-3}] \begin{bmatrix} v_{C3}(t) \\ i_{R1}(t) \end{bmatrix} \tag{24.218}$$

Nếu ta xét độ lớn tương đối của các thành phần của ma trận C, chúng ta có thể *đoán được* đầu ra $v_0(t)$ sẽ chủ yếu được quyết định bởi trạng thái $i_{R1}(t)$. Để xác minh điều này, chúng ta tính gramian của tính quan sát được được định nghĩa trong (24.172) bằng cách giải:

$$0 = A^T Q + QA + C^T C \tag{24.219}$$

$$0=\begin{bmatrix}-1 & 10^{-3}\\ 0 & -\frac{20}{9}\end{bmatrix}\begin{bmatrix}q_{11} & q_{12}\\ q_{21} & q_{22}\end{bmatrix}+\begin{bmatrix}q_{11} & q_{12}\\ q_{21} & q_{22}\end{bmatrix}\begin{bmatrix}-1 & 0\\ 10^{3} & -\frac{20}{9}\end{bmatrix}+\begin{bmatrix}1\\ -10^{3}\end{bmatrix}\begin{bmatrix}1 & 10^{-3}\end{bmatrix} \quad (24.220)$$

Chúng ta có:

$$Q=\begin{bmatrix}0.57 & 69.83\\ 69.83 & 225000\end{bmatrix} \quad (24.221)$$

Từ đó ta có thể tính sự đóng góp của mỗi trạng thái với năng lượng tổng ở đầu ra. Bằng cách làm như vậy, chúng ta xác minh được rằng biến trạng thái $i_{R1}(t)$ có một tác động trên đầu ra lớn hơn tác động của $v_{C3}(t)$, như được định nghĩa ở phương trình (24.212):

$$x_0=[1,0]^T \Rightarrow E(x_0)=0.57 \quad (24.222)$$

$$x_0=[1,0]^T \Rightarrow E(x_0)=225000 \quad (24.223)$$

Hàm truyền là:

$$\frac{V_o(s)}{V_i(s)}=\frac{s+1+\frac{1}{9}}{s+\frac{20}{9}}\cdot\frac{1}{s+1} \quad (24.224)$$

Ta quan sát thấy rằng có một sự triệt tiêu điểm cực - không giả.

Nguyên tắc đối ngẫu

Chúng ta thấy một sự giống nhau giữa các kết quả ở định lý 24.3 và 24.4, và giữa các định nghĩa về gramian của (24.172) và (24.213). Đó là **nguyên tắc đối ngẫu**. nguyên tắc này được phát biểu như sau:

Định lý 24.5 (đối ngẫu) *Xét mô hình không gian trạng thái mô tả bởi một hệ thống cơ sở dữ liệu (4-tuple (A, B, C, D)). Khi đó hệ thống là có thể điều khiển được hoàn toàn nếu và chỉ nếu hệ thống đối ngẫu (A^T, B^T,C^T,D^T) là có thể quan sát được hoàn toàn.*

Khai triển chính tắc và Tính xác định được

Định lý ở trên thường được sử dụng để đi từ kết quả của tính điều khiển được sang kết quả của tính quan sát được và ngược lại. Đối ngẫu của bổ đề 24.1 là:

Bổ đề 24.4 *Nếu hạng* $[A, C]^T = n$, *thì tồn tại một biến đổi tương đương* $\bar{x} = Tx$, $\bar{A} = TAT^{-1}$, $\bar{C} = CT^{-1}$, *từ đó* $\bar{C}$ *và* $\bar{A}$ *có dạng:*

$$\bar{A}=\begin{bmatrix}\bar{A}_0 & 0\\ \bar{A}_{21} & \bar{A}_{no}\end{bmatrix},\quad \bar{C}=\begin{bmatrix}\bar{C}_0 & 0\end{bmatrix} \quad (24.225)$$

Trong đó $\bar{A}_0$ *có k chiều và cặp* ($\bar{C}_0$, $\bar{A}_0$) *là quan sát được hoàn toàn.*

Kết quả này có một tương đương thích hợp với thuộc tính điều khiển được và sự khai triển liên quan. Để đánh giá điều này, ta sử dụng bổ đề 24.1 để biểu diễn các phương trình trạng thái (đã được biến đổi) và phương trình đầu ra theo dạng được phân chia như sau:

$$\begin{bmatrix}\dot{\bar{x}}_0(t)\\ \dot{\bar{x}}_{no}(t)\end{bmatrix}=\begin{bmatrix}\bar{A}_0 & 0\\ \bar{A}_{21} & \bar{A}_{no}\end{bmatrix}\begin{bmatrix}\bar{x}_0(t)\\ \bar{x}_{no}(t)\end{bmatrix}+\begin{bmatrix}\bar{B}_0\\ \bar{B}_{no}\end{bmatrix}u(t) \quad (24.226)$$

$$y(y) = \begin{bmatrix} \overline{C}_0 & 0 \end{bmatrix} \begin{bmatrix} \overline{x}_0(t) \\ \overline{x}_{n0}(t) \end{bmatrix} + Du(t) \tag{24.227}$$

Mô tả trên đây nói lên tại sao có thể gặp khó khăn khi cố điều khiển hệ thống chỉ bằng đầu ra. Đầu ra không có thông tin trên trạng thái $\overline{x}_{n0}$.

Không gian con có thể quan sát được của một mô hình là không gian bao gồm tất cả các trạng thái được sinh ra qua mỗi tổ hợp tuyến tính có thể của các trạng thái của $\overline{x}_0$. Khả năng ổn định của không gian con này được quyết định bởi vị trí của các trị riêng của $\overline{A}_0$.

Không gian con không thể quan sát được của một mô hình là không gian bao gồm tất cả các trạng thái được sinh ra qua mỗi tổ hợp tuyến tính có thể của các trạng thái của $\overline{x}_{n0}$. Khả năng ổn định của không gian con này được quyết định bởi vị trí của các trị riêng của $\overline{A}_{n0}$.

Nếu không gian con không thể quan sát được là có thể ổn định thì chúng ta nói rằng hệ thống là **có thể xác định được.**

Một đặc tính quan trọng của các mô tả (24.226) và (24.227) xuất phát từ thực tế là hàm truyền được cho bởi:

$$H(s) = \overline{C}_0 (sI - \overline{A}_0)^{-1} \overline{B}_0 + D \tag{24.228}$$

Phương trình (24.228) nói rằng các trị riêng của không gian con không thể quan sát được không phụ thuộc vào tập các điểm cực của hàm truyền hệ thống. Điều này chỉ ra rằng có một sự triệt tiêu của tất cả các điểm cực tương ứng với các nghiệm của $(sI - \overline{A}_{n0})$.

Dạng chính tắc của tính quan sát được

Có các cặp đối ngẫu của các dạng chính tắc được đưa ra trong bổ đề 24.2 và 24.3. Ví dụ cặp đối ngẫu trong bổ đề 24.3 là:

Bổ đề 24.5 *Xét một hệ thống SISO quan sát được hoàn toàn. Khi đó tồn tại một biến đổi tương đương biến đổi mô hình về* ***dạng chính tắc của bộ quan sát****:*

$$\dot{x}_0(t) = \begin{bmatrix} -\alpha_{n-1} & 1 & & \\ \vdots & & \ddots & \\ \vdots & & & 1 \\ -\alpha_0 & 0 & 0 & \end{bmatrix} x(t) + \begin{bmatrix} b_{n-1} \\ \vdots \\ \vdots \\ b_0 \end{bmatrix} u(t) \tag{24.229}$$

$$y(t) = [1 \quad 0 \quad \cdots \quad 0]x(t) + Du(t) \tag{24.230}$$

Khai triển chính tắc

Sự hiểu biết hơn nữa về cấu trúc của các hệ thống tuyến tính động nhận được bằng cách chỉ xem xét những hệ thống chỉ quan sát hoặc điều khiển được từng phần. Những hệ thống này có thể được phân tách thành những hệ thống quan sát hoàn toàn và điều khiển được hoàn toàn.

Hai kết quả của bổ đề 24.1 và 24.4 có thể được kết hợp cho cả những hệ thống mà không phải là quan sát được hoàn toàn và cũng như điều khiển được hoàn toàn. Chúng ta có thể thấy như sau.

Định lý 24.6 (định lý khai triển chính tắc) *Xét hệ thống được mô tả bởi dạng không gian trạng thái. Khi đó luôn tồn tại một biến đổi tương đương T sao cho mô hình đã được biến đổi cho* $\overline{x} = T^{-1}x$ *có dạng:*

$$\overline{A} = \begin{bmatrix} \overline{A}_{co} & 0 & \overline{A}_{13} & 0 \\ \overline{A}_{21} & \overline{A}_{22} & \overline{A}_{23} & \overline{A}_{24} \\ 0 & 0 & \overline{A}_{33} & 0 \\ 0 & 0 & \overline{A}_{34} & \overline{A}_{44} \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} \overline{B}_1 \\ \overline{B}_2 \\ 0 \\ 0 \end{bmatrix}, \quad \overline{C} = \begin{bmatrix} \overline{C}_1 & 0 & \overline{C}_2 & 0 \end{bmatrix} \tag{24.231}$$

trong đó

i. *Hệ thống con* $[\ \overline{A}_{co}, \overline{B}_1, \overline{C}_1\ ]$ *là vừa có thể điều khiển được hoàn toàn và vừa có thể quan sát được hoàn toàn và có cùng hàm truyền như hệ thống ban đầu (xem bổ đề 24.6).*

ii. *Hệ thống con:*

$$\begin{bmatrix} \overline{A}_{co} & 0 \\ \overline{A}_{21} & \overline{A}_{23} \end{bmatrix}, \begin{bmatrix} \overline{B}_1 \\ \overline{B}_2 \end{bmatrix}, \begin{bmatrix} \overline{C}_1 & 0 \end{bmatrix} \tag{24.232}$$

là có thể điều khiển được hoàn toàn.

iii. *Hệ thống con:*

$$\begin{bmatrix} \overline{A}_{co} & \overline{A}_{13} \\ 0 & \overline{A}_{33} \end{bmatrix}, \begin{bmatrix} \overline{B}_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \overline{C}_1 & \overline{C}_2 \end{bmatrix} \tag{24.233}$$

là có thể quan sát được hoàn toàn.

Khai triển chính tắc được mô tả trong định lý 24.6 dẫn ra một hệ quả quan trọng cho hàm truyền của mô hình chỉ có không gian phụ có thể điều khiển được hoàn toàn và có thể quan sát được hoàn toàn.

Bổ đề 24.6 *Xét ma trận hàm truyền* **H***(s) cho bởi:*

$$Y(s) = H(s)U(s) \tag{24.234}$$

Khi đó:

$$H = C(sI - A)^{-1}B + D = \overline{C}_1(sI - \overline{A}_{co})^{-1}\overline{B}_1 + D \tag{24.235}$$

trong đó $\overline{C}_1$, $\overline{A}_{co}$, *và* $\overline{B}_1$ *đều giống như trong phương trình (24.231). Mô tả trạng thái này là một sự thực hiện tối thiểu của hàm truyền.*

Nếu **M** là ma trận vuông bất kỳ và ký hiệu $\Lambda\{M\}$ là tập các trị riêng của **M** thì:

$$\Lambda\{\overline{A}\} = \Lambda\{\overline{A}_{co}\} \cup \Lambda\{\overline{A}_{22}\} \cup \Lambda\{\overline{A}_{33}\} \cup \Lambda\{\overline{A}_{44}\} \tag{24.236}$$

trong đó:

$\Lambda\{\overline{A}\}$ = các trị riêng của hệ thống,

$\Lambda\{\overline{A}_{co}\}$ = các trị riêng của hệ thống con có thể điều khiển được và có thể quan sát được,

$\Lambda\{\overline{A}_{22}\}$ = các trị riêng của hệ thống con có thể điều khiển được nhưng không thể quan sát được,

$\Lambda\{\overline{A}_{33}\}$ = các trị riêng của hệ thống con có thể quan sát được nhưng không thể điều khiển được,

$\Lambda\{\overline{A}_{44}\}$ = các trị riêng của hệ thống con không thể quan sát và không thể điều khiển được.

Ta thấy rằng tính điều khiển được của một hệ thống cho trước phụ thuộc vào cấu trúc của các cổng vào, có nghĩa là, chỗ mà các đầu vào có thể điều chỉnh được được đặt vào. Do đó, các trạng thái của một hệ thống con cho trước có thể không điều khiển được với một đầu vào cho trước, nhưng có thể điều khiển được hoàn toàn với đầu vào khác. Sự khác biệt này là quan trọng khi thiết kế hệ thống điều khiển vì không phải tất cả các đầu vào nào của đối tượng điều khiển cũng điều khiển được (ví dụ các nhiễu) và vì vậy, không thể được sử dụng để lái đối tượng điều khiển tới những trạng thái nhất định.

Tương tự như vậy, tính quan sát được phụ thuộc vào đầu ra nào đang được xem xét. Những trạng thái nào đó có thể không quan sát được từ đầu ra cho trước, nhưng có thể hoàn toàn quan sát được từ một số đầu ra khác. Điều này cũng có tác động đáng kể đến các hệ thống điều khiển phản hồi đầu ra, vì một số trạng thái có thể không xuất hiện ở đầu ra đang được đo và phản hồi của đối tượng điều khiển Tuy nhiên, chúng có thể xuất hiện ở các biến nội tại cốt yếu và vì vậy là quan trọng với vấn đề điều khiển.

Kiểm tra PBH

Một cách kiểm tra khác cho tính điều khiển được và quan sát được cung cấp bởi bổ đề được biết đến như là phép kiểm tra **PBH** sau:

Bổ đề 24.7 *Xét một mô hình không gian trạng thái* (A, B, C). Khi đó

i. *Hệ thống là không thể quan sát được hoàn toàn nếu và chỉ nếu tồn tại một vectơ khác không* $x \in \mathbb{C}^n$ *và một đại lượng vô hướng* $\lambda \in \mathbb{C}$ *sao cho*:

$$Ax = \lambda x, \quad Cx = 0 \tag{24.237}$$

ii. *Hệ thống không thể điều khiển được hoàn toàn nếu và chỉ nếu tồn tại một vectơ khác không* $x \in \mathbb{C}^n$ *và một đại lượng vô hướng* $\lambda \in \mathbb{C}$ *sao cho*:

$$x^T A = \lambda x^T, \quad x^T B = 0 \tag{24.238}$$

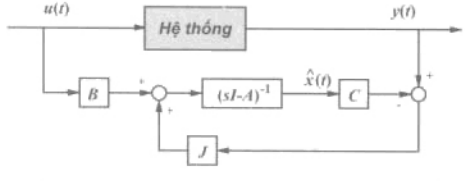
24.7 Bộ quan sát trạng thái

Các khái niệm cơ bản

Khi các biến trạng thái được đo lường để giám sát, thực hiện các hệ thống điều khiển, hoặc cho các mục đích khác mà phải đối mặt với những vấn đề khó khăn về công nghệ và kinh tế. Bộ quan sát là cách để ước lượng các biến trạng thái dựa trên mô hình hệ thống, đo lường đầu ra **y**(t), và đầu vào u(t). Đây là sự khái quát hóa của việc đo lường gián tiếp một biến hệ thống bằng cách sử dụng mô hình hệ thống và đo một vài biến khác dễ đo hơn.

Động học bộ quan sát

Giả thiết rằng hệ thống có mô hình không gian trạng thái cho bởi (24.42) và (24.43) với **D = 0** (một hệ thống chính xác như đã giả thiết). Khi đó, cấu trúc chung của một bộ quan sát trạng thái hệ thống kinh điển được chỉ ra trên hình 24.15, trong đó ma trận **J** là **hệ số khuyếch đại** của bộ quan sát.



HÌNH 24.15 Bộ quan sát trạng thái kinh điển

Phương trình của bộ quan sát là:

$$\frac{d\hat{x}}{dt} = A\hat{x}(t) + Bu(t) + J(y(t) - C\hat{x}(t)) \tag{24.239}$$

Một câu hỏi đặt ra là: nếu chúng ta biết mô hình chính xác của hệ thống và đầu vào hệ thống, thì tại sao ta cần phải cung cấp đầu ra hệ thống? Câu trả lời là cần phải đo đầu ra vì **chúng ta không biết trạng thái ban đầu của hệ thống**. Điều này có thể được đánh giá bằng phương trình sai số ước lượng trạng thái, $\tilde{x}(t) = x(t) - \hat{x}(t)$. Phương trình này có thể nhận được bằng cách trừ (24.42) cho (24.239). Từ đó ta có:

$$\frac{d\tilde{x}(t)}{dt} = (A - JC)\tilde{x}(t) \tag{24.240}$$

Từ (24.240) ta thấy sai số ước lượng sẽ hội tụ về không với một sai số ban đầu khác không nếu và chỉ nếu tất cả các trị riêng của ma trận $A - JC$ có các phần thực âm, có nghĩa là đa thức $E(s) = \det(sI - A + JC)$ của bộ quan sát là Hurwitz hoàn toàn.

Nhận xét

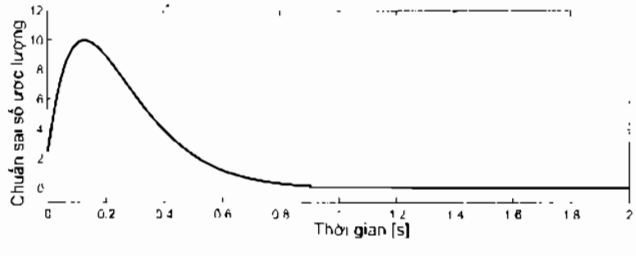
- Phương trình (24.240) chỉ hợp lệ nếu mô hình mô tả đúng hệ thống. Các sai số của việc mô hình hóa sẽ tác động đến bộ quan sát. Điều này thường dẫn đến các sai số ước lượng trạng thái khác không.
- Nếu cặp (**A, C**) là có thể quan sát được hoàn toàn, thì trị riêng của A-JC có thể được định vị tùy ý (trong miền ổn định). Vì vậy, tốc độ hội tụ ước lượng là do sự lựa chọn của người thiết kế. Các trị riêng này là các **điểm cực của bộ quan sát**.
- Nếu cặp (**A, C**) là có thể xác định, thì bộ quan sát sẽ dẫn ra một cách tiệm cận sai số trạng thái tĩnh không, mặc dù không phải tất cả các trị riêng của A-JC đều có thể được đặt theo ý muốn.
- Nếu hệ thống là không thể quan sát được hoàn toàn, và không gian con không thể quan sát được chứa các mô hình không ổn định, thì bộ quan sát sẽ không bao giờ hội tụ.

Để minh họa các phương pháp của bộ quan sát, chúng ta tham khảo ví dụ 24.5.

Ví dụ 24.20

Giả thiết rằng chúng ta muốn các cực của bộ quan sát với mô hình trạng thái trong ví dụ 24.5 được đặt ở s = -4, s = -6, và s = -8. Ta có thể tính hệ số khuyếch đại **J** của bộ quan sát, bằng cách sử dụng phần mềm như MATLAB chẳng hạn. Khi đó, ta có được:

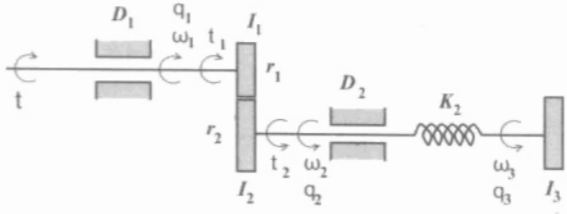
$$J = \begin{bmatrix} -4.5247 & -7.5617 & -4.1543 \end{bmatrix}^T \tag{24.241}$$



HÌNH 24.16 Sai số ước lượng trạng thái

Để đánh giá động học bộ quan sát, giả thiết rằng trạng thái ban đầu của hệ thống là $x(0) = [-1 \quad 2 \quad 1]^T$ và đầu vào hệ thống là một sóng vuông của biên độ 1 và tần số 1 rad/s. Bộ

quan sát được khởi tạo với $\hat{x}(0) = 0$. Khi đó chuẩn của sai số ước lượng, $\|\hat{x}(t)\|$, sẽ có dạng như chỉ ra trong hình 24.16.



HÌNH 24.17 Hệ thống quay

điều quan trọng được chỉ ra trong ví dụ này là đối tượng điều khiển không ổn định. Có nghĩa rằng trạng thái và ước lượng trạng thái tăng không giới hạn. Tuy nhiên, với giả thiết rằng mô hình là hoàn toàn đúng, thì sai số ước lượng sẽ hội tụ về không.

Ví dụ 24.21

Hình 24.17 biểu diễn sơ đồ của một hệ thống quay điều khiển bởi mô men $\tau(t)$. Năng lượng hệ thống được truyền qua hệ thống bánh răng với hai bánh xe có bán kính r_1 và r_2 và quán tính tương ứng I_1 và I_2. Chuyển động của hai trục quay bị giảm bởi ma sát với hệ số D_1 và D_2, đàn hồi xoắn đáng kể của trục 2 cũng được mô hình hóa. Tải của hệ thống được mô hình hóa là quán tính I_3. Ta muốn ước lượng tốc độ tải ω_3 dựa trên việc đo tốc độ ω_1 của trục 1.

Trước tiên ta cần xây dựng mô hình không gian trạng thái. Để làm việc đó, chúng ta chọn một tập tối thiểu các biến hệ thống xác định năng lượng được lưu giữ trong hệ thống. Hệ thống có bốn thành phần có thể chứa năng lượng: 3 quán tính và một đàn hồi. Năng lượng trong I_1 và I_2 có thể được tính từ hoặc ω_1 hoặc ω_2, có nghĩa là chúng ta chỉ cần một trong các tốc độ đó, vì chúng thỏa mãn:

$$\frac{\omega_1(t)}{\omega_2(t)} = \frac{r_2}{r_1} \quad and \quad \tau_1(t)\omega_1(t) = \tau_2(t)\omega_2(t) \tag{24.242}$$

Vì vậy, các biến trạng thái được chọn theo định hướng vật lý là:

$$x_1(t) = \omega_1(t) \tag{24.243}$$

$$x_2(t) = \theta_2(t) - \theta_3(t) \tag{24.244}$$

$$x_3(t) = \omega_3(t) \tag{24.245}$$

Từ các nguyên tắc đầu tiên ta có:

$$\tau(t) = D_1\omega_1(t) + I_1\frac{d\omega_1(t)}{dt} + \tau_1(t) \tag{24.246}$$

$$\tau(t) = \frac{r_2}{r_1}\tau_1(t) = D_2\omega_2(t) + I_2\frac{d\omega_2(t)}{dt} + K_2(\theta_2(t) - \theta_3(t)) \tag{24.247}$$

$$0 = K_2(\theta_3(t) - \theta_2(t)) + I_3\frac{d\omega_3(t)}{dt} \tag{24.248}$$

Vì chúng ta chọn $\omega_1(t)$ như là biến hệ thống có thể đo được, nên ta nhận được:

$$\frac{dx(t)}{dt} = \underbrace{\begin{bmatrix} \frac{r_1^2 D_2 + r_2^2 D_1}{r_1^2 I_2 + r_2^2 I_1} & -\frac{r_1 r_2 K_2}{r_1^2 I_2 + r_2^2 I_1} & 0 \\ \frac{r_1}{r_2} & 0 & -1 \\ 0 & \frac{K_2}{I_1} & 0 \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} \frac{r_2^2}{r_1^2 I_2 + r_2^2 I_1} \\ 0 \\ 0 \end{bmatrix}}_{B} \tau(t) \qquad (24.249)$$

$$\omega_1(t) = \underbrace{\begin{bmatrix} 1 & 0 & 1 \end{bmatrix}}_{C} x(t) \qquad (24.250)$$

Để đánh giá các thuộc tính quan sát được của hệ thống này, chọn các thông số như sau:

$$r_1 = 0.25m, \quad r_2 = r_3 = 0.50m, \quad D_1 = D_2 = 10Nms/rad \qquad (24.251)$$

$$K_2 = 30Nm/rad, \quad I_1 = 2.39Nms^2/rad, \quad I_2 = I_3 = 38.29Nms^2/rad \qquad (24.252)$$

Với các giá trị đó ta có:

$$A = \begin{bmatrix} -1.045 & -1.254 & 0 \\ 0.5 & 0 & -1 \\ 0 & 0.784 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.084 \\ 0 \\ 0 \end{bmatrix} \qquad (24.253)$$

Ta tiếp tục sử dụng phép kiểm tra đã được trình bày trong "tính quan sát được, tính khôi phục được và tính xác định được". Điều này sẽ dẫn đến:

$$\Gamma_0 = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 1000 & 0 & 0 \\ -1.0450 & -1.2540 & 0 \\ 0.4650 & 1.3104 & 1.2540 \end{bmatrix} \qquad (24.254)$$

Từ biểu diễn này ta thấy rằng Γ_0 là một ma trận có hạng đầy đủ. Do đó trạng thái của hệ thống là quan sát được hoàn toàn từ $\omega_1(t)$.

Trước chúng ta có một ước lượng trạng thái, $\hat{x}(t)$, một ước lượng $\omega_3(t)$ cho ω_3 nhận được từ:

$$\omega_3(t) = \underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}}_{K_3^T} \hat{x}(t) \qquad (24.255)$$

Trong đó $\omega_3(t)$ có thể nhận được từ (24.239) và do đó:

$$\frac{d\omega_3(t)}{dt} = K_3^T \frac{d\hat{x}(t)}{dt} = K_3^T (A - JC)\hat{x}(t) + \underbrace{K_3^T B}_{0} \tau(t) + K_3^T J\omega_1(t) \qquad (24.256)$$

Bộ quan sát và nhiễu đo lường

Theo lý thuyết trên, ta giả thiết rằng cả đầu vào $\mathbf{u}(t)$ của hệ thống và đầu ra $\mathbf{y}(t)$ đều không có sai số. Giả thiết này thường đúng với $\mathbf{u}(t)$ thường dùng cùng một $\mathbf{u}(t)$ để ước tính trạng thái. Tuy nhiên, giả thiết đó thường không hợp lệ với $\mathbf{y}(t)$ vì việc đo đo lường biến này thường bị sai lệch do bởi nhiễu. Để phân tích ảnh hưởng của sai số này, ta ký hiệu $\mathbf{y_m}(t)$ là đại lượng đo có nhiễu, có nghĩa là $\mathbf{y_m}(t) = \mathbf{y}(t) - \mathbf{v}(t)$, trong đó v(t) là nhiễu đo lường được cộng thêm. Vì vậy, sai số ước tính trạng thái thỏa mãn:

$$\frac{d\tilde{x}(t)}{dt} = (A - JC)\tilde{x}(t) + Jv(t) \tag{24.257}$$

Khi đó chúng ta có:

$$\tilde{X}(s) = (sI - A + JC)^{-1}\tilde{x}(0) + (sI - A + JC)^{-1}JV(s) \tag{24.258}$$

Vì vậy, sai số là nhỏ nếu hàm truyền $(sI - A + JC)^{-1}J$ lọc nhiễu. Xét ví dụ sau.

Ví dụ 24.22

Một hệ thống có mô hình không gian trạng thái được cho bởi:

$$A = \begin{bmatrix} -2 & -1 \\ 1 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & -1 \end{bmatrix}, \quad D = 0 \tag{24.259}$$

Giả thiết rằng, chúng ta muốn ước tính biến hệ thống $z(t) = \gamma^T x(t)$ trong đó $\gamma^T = [1\ 1]$. Ước lượng dựa trên cơ sở quan sát phù hợp là $\hat{z}(t)$ được cho bởi:

$$\hat{z} = \gamma^T \hat{x}(t) \tag{24.260}$$

Khi đó, nhiễu khi ước tính $z(t)$ là $z_v(t)$, có biến đổi Laplace thỏa mãn:

$$Z_v(s) = H_v(s)V(s), \quad \text{với} \quad H_v(s) = \gamma^T(sI - A + JC)^{-1}J \tag{24.261}$$

Tiếp theo ta xét hai lựa chọn khác nhau cho đa thức E(s) của bộ quan sát. Đó là:

$$E_1(s) = (s + 0.5)(s + 0.75) \text{ và } E_2(s) = (s + 10)(s + 20) \tag{24.262}$$

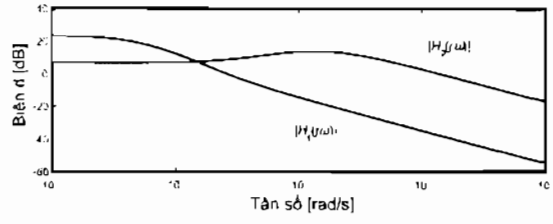
Người đọc có thể thấy rằng các bộ quan sát được dẫn ra sẽ có các tốc độ rất khác nhau. Bộ quan sát đầu chậm hơn nhiều bộ thứ hai.

Với các lựa chọn này chúng ta tính các hệ số khuếch đại J_1 và J_2, và các hàm lọc tương ứng của bộ quan sát:

$$H_1(s) = \gamma^T(sI - A + J_1C)^{-1}J_1 = \frac{1.875s + 5.625}{s^2 + 1.25s + 0.375} \tag{24.263}$$

$$H_2(s) = \gamma^T(sI - A + J_2C)^{-1}J_2 = \frac{144s + 432}{s^2 + 30s + 200} \tag{24.264}$$

Để so sánh cả hai trường hợp, ta tính và vẽ đáp ứng tần số của mỗi bộ lọc. Kết quả được chỉ ra trên hình (24.18). Từ hình 24.28 ta thấy rằng, với tần số cao, bộ lọc chậm kháng nhiễu tốt hơn bộ lọc nhanh.



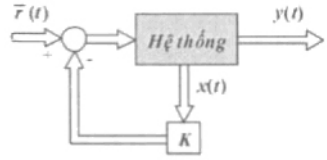
HÌNH 24.18 Đặc tuyến lọc của bộ quan sát

Ví dụ trên đây minh họa sự cân bằng giữa tốc độ của bộ quan sát và tính kháng nhiễu. Một cách hệ thống để giải quyết vấn đề khó xử này là sử dụng lý thuyết lọc tối ưu, như bộ lọc Kalman_Bucy. Người đọc quan tâm có thể tham khảo ở [2].

24.8 Phản hồi trạng thái

Các khái niệm cơ bản

Khi tất cả các trạng thái hệ thống đều có thể đo được và hệ thống có thể đạt được hoàn toàn (theo nghĩa đã được giải thích trong mục "Khả năng điều khiển, khả năng đạt được và khả năng ổn định"), thì chúng ta có thể điều khiển hệ thống bằng cách sử dụng phản hồi trạng thái để đạt được lệnh đầy đủ của động học vòng kín. Ý tưởng này được thể hiện trong hình (24.19).



HÌNH 24.19 Phản hồi trạng thái

Hình 24.19 biểu diễn dạng cơ bản nhất của phản hồi trạng thái: đầu vào của hệ thống là một thành phần tỉ lệ với trạng thái (thành phần còn lại là tín hiệu ngoài $\bar{r}(t)$).

Phản hồi trạng thái là một ý tưởng rất đơn giản và ngây thơ. Suy xét kĩ cho thấy ý tưởng này có một số thiếu sót và những đặc tính tiềm ẩn mối nguy hiểm như:

- Đòi hỏi nhiều biến như các biến trạng thái. Điều này không chỉ là rất đắt mà trong một số trường hợp, không thể thực hiện được.
- Mỗi phép đo trạng thái là một nguồn gốc của sai số vì độ chính xác của nó là hữu hạn.
- Mỗi phép đo sẽ đưa vào nhiễu với tác động xấu đến việc thực hiện hệ thống điều khiển.
- Thực hiện toàn bộ một cách chính xác dựa trên một hàm chính xác của một tập hợp phức tạp các thiết bị. Điều này sẽ đặt ra một số vấn đề liên quan đến sự thực hiện và khả năng tích hợp hệ thống.

Mặc dù có các điểm yếu, nhưng phản hồi trạng thái tự bản thân nó là một khái niệm mạnh, vì nó là cơ sở cho nhiều cấu trúc điều khiển mạnh và tinh vi hơn. Lý do chủ yếu cho điều này là bất kì bộ điều khiển tuyến tính nào cũng có thể được giải thích như một sự kết hợp của bộ quan sát trạng thái và phản hồi trạng thái.

Động học của phản hồi

Giả thiết rằng hệ thống được điều khiển có hàm truyền $\mathbf{H}(s)$ và không gian trạng thái được biểu diễn bởi (24.42) và (24.43) với $\mathbf{D} = \mathbf{0}$. Nếu đầu vào của đối tượng điều khiển được điều khiển theo:

$$u(t) = -Kx(t) + \bar{r}(t) \tag{24.265}$$

khi đó biểu diễn không gian trạng thái cho toàn bộ vòng điều khiển được cho bởi:

$$\frac{dx(t)}{dt} = Ax(t) + B(-Kx(t) + \bar{r}(t)) \tag{24.266}$$

$$Y(t) = Cx(t) \tag{24.267}$$

Có thể thấy rằng mối quan hệ giữa $\bar{R}(s)$ và $\mathbf{Y}(s)$ được cho bởi:

$$Y(s) = \underbrace{C'(sI - A)^{-1} B}_{H(s)}(I + K(sI - A)^{-1} B)^{-1} \overline{R}(s) \tag{24.268}$$

Điều này cho thấy vòng phản hồi trạng thái sẽ bảo toàn các điểm không của hệ thống và dịch các điểm cực tới các nghiệm của định thức ($s\mathbf{I}$ - **A** −**BK**).

Phản hồi trạng thái tối ưu. Điều khiển tối ưu

Xét hệ thống tuyến tính bất biến có không gian trạng thái được biểu diễn bởi (24.42) và (23.43) với **D** = **0**, với giả thiết trạng thái ban đầu $\mathbf{x}(0) = \mathbf{x}_0$.

Giả thiết rằng mục đích điều khiển là để lái hệ thống từ trạng thái ban đầu $\mathbf{x}_0$ tới giá trị nhỏ nhất có thể càng sớm càng tốt trong khoảng $[0, t_f]$. Chúng ta cần yêu cầu thêm rằng, quá trình lái không yêu cầu quá nhiều nỗ lực điều khiển. Khi đó vấn đề điều khiển tối ưu được xác định là vấn đề tìm điều khiển tối ưu $\mathbf{u}(t)$ trong khoảng thời gian $[0, t_f]$ sao cho một hàm giá bậc hai được tối thiểu. Hàm giá này được chọn là:

$$J_u(x_0) = \int_0^{t_f} [x(t)^T Qx(t) + u(t)^T Ru(t)]dt + x(t_f)^T Q_f x(t_f) \tag{24.269}$$

trong đó $Q \in \mathbb{R}^{n \times n}, Q_f \in \mathbb{R}^{n \times n}$ đều là các ma trận xác định, đối xứng, không âm và $R \in \mathbb{R}^{m \times m}$ là ma trận xác định, đối xứng, dương. Các yêu cầu đối với các ma trận trọng số là được đặt sao cho hàm giá trị có nghĩa. Nếu **Q** được phép là âm, thì giá trị *tốt ưu* có thể thậm chí là âm trong khi trạng trạng thái có thể tăng không giới hạn về độ lớn. Ngoài ra, nếu ta cho phép **R** có các trị riêng ở gốc (có nghĩa là **R** được phép là một ma trận xác định không âm, thay vì yêu cầu nó hoàn toàn dương) thì tín hiệu điều khiển $\mathbf{u}(t)$ cũng có thể tăng không giới hạn (theo các hướng của các vectơ trị riêng liên quan).

Một luật điều khiển tuyến tính bất biến nhận được bằng cách tiệm cận khi $t_f \to \infty$. Với điều kiện này, luật điều khiển tối ưu được cho bởi:

$$u^0(t) = -K^0 x(t) \tag{24.270}$$

với

$$K^0 = -R^{-1} B^T P_s \tag{24.271}$$

trong đó P_s là lời giải duy nhất không âm của phương trình đại số Riccacti:

$$0 = Q - P_s BR^{-1} B^T P_s + P_s A + A^T P_s \tag{24.272}$$

Để lời giải này tồn tại, cần thỏa mãn những điều kiện kĩ thuật nhất định (vấn đề này đã được bàn luận chi tiết, ví dụ trong [5]).

Nhận xét

- Lời giải cho vấn đề LQR sẽ tối thiểu hóa hàm giá trị (24.269) và, khi $t_f \to \infty$, hệ thống luôn ổn định.
- Vấn đề quan trọng là chọn các ma trận trọng lượng **Q** và **R** như thế nào. Lựa chọn thông thường cho **Q** là $Q=C^TC$. Với lựa chọn này, đầu ra của hệ thống được đặt trực tiếp vào hàm giá trị.
- Với **Q** cho trước, *kích thước* của **R** ảnh hưởng mạnh đến vị trí các điểm cực của vòng lặp kín. **R** càng lớn thì vòng lặp điều khiển càng chậm.

Để biết thêm về bộ điều khiên tối ưu bậc hai, có thể đọc các tài liệu ví dụ [1, 3, 4, 8, 9].

24.9 Phản hồi trạng thái quan sát được

Phương pháp phân ly

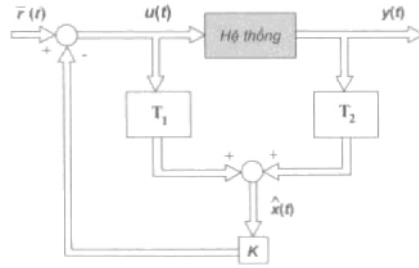
Do những hạn chế trong việc đo trạng thái, nên phản hồi của trạng thái ước tính có thể được sử dụng thay thế. Kết quả là hệ thống điều khiển tích hợp bộ quan sát và hệ cơ phản hồi cho các trạng thái quan sát được.

Sự kết hợp của bộ quan sát trạng thái và phản hồi của trạng thái ước tính phù hợp với cấu trúc trong hình 24.20.

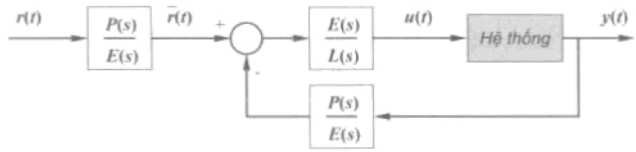
Trong hình 24.20, hàm truyền (ma trận) $\mathbf{T}_1(s)$ và $\mathbf{T}_2(s)$ có thể đạt được từ hình (24.15). Đó là:

$$T_1(s) = (sI - A_0 + JC_0)^{-1} B_0 \tag{24.273}$$

$$T_2(s) = (sI - A_0 + JC_0)^{-1} J \tag{24.274}$$



HÌNH 24.20 Phản hồi trạng thái ước tính



HÌNH 24.21 Vòng điều khiển tương đương

Giải thích hàm truyền cho trường hợp "một đầu vào một đầu ra"

Xét hệ thống SISO có hàm truyền:

$$G_0(s) = C(sI - A_0)^{-1} B = \frac{N_0(s)}{M_0(s)} \tag{24.275}$$

Trong đó $M_0(s)$ và $N_0(s)$ đều là đa thức của s.

Trước tiên, hệ số khuyếch đại **K** của phản hồi trạng thái được chọn để nhận được đa thức vòng kín $F(s)$, trong đó $F(s) = \det(s\mathbf{I} - \mathbf{A}_0 + \mathbf{B}_0\mathbf{K})$. Tiếp theo, hệ số khuếch đại **J** của bộ quan sát được tính toán để nhận được đa thức của bộ quan sát $E(s) = \det(s\mathbf{I} - \mathbf{A}_0 + \mathbf{JC}_0)$.

Nếu bộ quan sát và phản hồi trạng thái quan sát được được kết hợp, thì vòng điều khiển kết quả có thể được tạo ra tương đương (bằng một lựa chọn $\bar{r}(t)$) với vòng điều khiển cổ điển như trong hình 24.21.

Trong hình 24.21 các đa thức $P(s)$ và $L(s)$ thỏa mãn phương trình Diophantine:

$$M_0(s)L(s) + N_0(s)P(s) = E(s)F(s) \tag{24.276}$$

Kết quả này cho thấy các điểm cực của vòng kín là sự kết hợp của tập các cực của bộ quan sát và tập các cực của phản hồi trạng thái.

Tài liệu tham khảo

[1] Anderson, B.D.O. and Moore, J., *Linear optimal Control.* Prentice-Hall, Englewood Cliffs, NJ, 1971.

[2] Anderson, B.D.O. and Moore, J., *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, NJ, 1979.

[3] Athans, M. and Falb, P., *Optimal Control.* McGraw HIll, 1966.

[4] Dennis Bernstein and Wassim Haddad. LQG control with an H performance bound: A Riccati equation approach. *IEEE Transactions on Automatic Control,* 34(3): L293–305, 1989.

[5] Bittanti, S., Laub, A.J., and Willems, J.C., *The Riccati Equation.* Springer Verlag, Berlin, 1996.

[6] Dorf, R.C. and Bishop, R., *Dạngrn Control Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1997.

[7] Franklin, G.F. and Powel, J.D., *Digital Control of Dynamics Systems.* 2nd ed., Addison-Wesley, 1990.

[8] Goodwin, G.C., Graebe, S., and Salgado, M.E., *Control System Design.* Prentice-Hall, NJ, 2001.

[9] Kwakernaak, H. and Sivan, R., *Linear Optimal Control System.* Wiley–Interscience, New York, 1972.

[10] Ogata, K., *State Space Analysis of Control Systems.* Prentice-Hall, Englewood Cliffis, NJ, 1967.

[11] Rosenbrock, H.H., *State Space and Multivariable Theory.* John Wiley and Sons, New York, 1970.

[12] Schultz, D.G. and Melsa, J.L., *State Function and Linear Control Systems.* McGraw Hill, New York, 1967.

[13] Wiberg, D.W., *Theory and Problems of State Space and Linear Systems.* McGraw Hill, New York, 1971.

[14] Zadeh, L.A. and Desoer, C.A., *Linear System Theory: The State Space Approach.* McGraw Hill, New York, 1963.

[15] Zhou, K., *Essentials of Robust Control.* Prentice-Hall, Englewood Cliffs, NJ, 1998.

[16] Zhou, K., Salomon, G., and Wu, E., Balanced realization and dạngl reduction for unstable systems. *International Journal of Robust and Nonlinear Control,* 9:183–198, 1999.

25

Đáp ứng của các hệ thống động

Raymond de Callafon
University of California

25.1 Hệ thống và phân tích tín hiệu 25-1
25.2 Đáp ứng động học 25-6
25.3 Các thông số hoạt động của hệ thống động.....25-11

25.1 Hệ thống và phân tích tín hiệu

Khi phân tích và thiết kế hệ thống động, dự đoán và hiểu được các ứng xử động của hệ thống là rất quan trọng. Nghiên cứu ứng xử động có thể được thực hiện bằng cách sử dụng mô hình toán học để mô tả ứng xử động của hệ cần quan tâm. Thông thường, một mô hình được công thức hóa để mô tả ứng xử theo thời gian liên tục hoặc rời rạc của hệ thống. Các phương trình tương ứng biểu diễn mô hình được sử dụng để dự đoán và hiểu ứng xử động của hệ thống.

Với các mô hình khá đơn giản của hệ động, có thể phân tích một cách chặt chẽ bằng cách tính các phương trình của mô hình. Thông thường, phân tích này bị giới hạn ở mô hình tuyến tính bậc một và hai. Mặc dù bị giới hạn ở các mô hình bậc thấp nhưng từ các lời giải này vẫn có thể hiểu biết sâu hơn về các đáp ứng điển hình của một hệ thống động. Với các mô hình các bậc phức tạp hơn và có thể phi tuyến, các công cụ mô phỏng số có thể được dùng để phân tích hệ thống động.

Phần này sẽ xem xét lại việc phân tích mô hình tuyến tính của các hệ thống động liên tục và rời rạc. Các phương trình để rời rạc hóa và liên kết hóa ứng xử liên tục và rời rạc sẽ được trình bày. Để phân tích hệ thống thời gian liên tục, biến đổi Laplace đưa phương trình vi phân tuyến tính thành các biểu thức đại số được sử dụng rộng rãi. Tương tự, biến đổi z được dùng cho các hệ thống thời gian rời rạc.

Hệ thống thời gian liên tục

Các mô hình biểu diễn ứng xử động thời gian liên tục tuyến tính của một hệ thống thường được cho dưới dạng các phương trình vi phân để liên kết tín hiệu vào $u(t)$ với tín hiệu ra $y(t)$. Phương trình vi phân của mô hình thời gian liên tục tuyến tính bất biến có dạng tổng quát như sau:

$$\sum_{j=0}^{n_a} a_j \frac{d^j}{dt^j} y(t) = \sum_{j=0}^{n_b} b_j \frac{d^k}{dt^j} u(t) \tag{25.1}$$

trong đó tổ hợp tuyến tính nhận được bằng cách dùng đạo hàm theo thời gian bậc j d^j / dt^j của một đầu ra đơn $y(t)$ và một đầu vào đơn $u(t)$. Trong (25.1), các số thực vô hướng a_j với $j = 0, \ldots, n_a, a_{n_a} \neq 0$ và b_j với $j = 0, \ldots, n_b, a_{n_b} \neq 0$ được gọi là các hệ số của mẫu số và tử số tương ứng. Đầu vào $u(t)$ được phân biệt từ đầu ra $y(t)$ theo (25.1) bởi yêu cầu $n_a \geq n_b$. Kết quả là đạo hàm bậc n_a là đạo hàm bậc cao nhất của đầu ra $y(t)$ và n_a được sử dụng để chỉ ra bậc của phương tình vi phân.

Mô hình hệ thống thời gian liên tục có thể được biểu diễn theo cách khác bằng cách viết lại phương trình vi phân bậc n_a thành một tập các phương trình vi phân bậc một. Điều này có thể được thực hiện bằng cách tạo ra biến trạng thái $x(t)$ và viết lại phương trình vi phân bậc cao thành:

$$\frac{d}{dt}x(t) = Ax(t) + Bu(t) \tag{25.2}$$

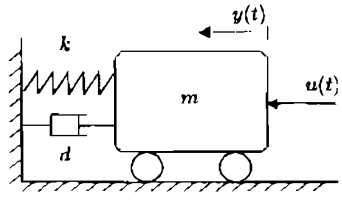
$$y(t) = Cx(t) + Du(t)$$

trong đó A, B, C và D là ma trận số thực. Tập các phương trình vi phân bậc một trong (25.2) được coi như là biểu diễn không gian trạng thái. Biến trạng thái $x(t)$ là vectơ cột và chứa n_a biến, trong đó n_a là bậc của phương trình vi phân.

Kích thước của các ma trận trong (25.2) tương ứng với bậc của phương trình vi phân (từ đó thu được không gian trạng thái). Để khái quát hóa, ta xét hệ nhiều đầu vào, nhiều đầu ra, được sắp xếp trong vectơ cột đầu vào $u(t)$ kích thước $m \times 1$ và vectơ cột đầu ra $y(t)$ kích thước $p \times 1$. Cho vectơ trạng thái kích thước $n_a \times 1$, ma trận trạng thái A có kích thước $n_a \times n_a$, ma trận đầu vào có kích thước $n_a \times m$, ma trận đầu ra C có kích thước $p \times n_a$ và ma trận D có kích thước $m \times p$. Từ đó có thể thấy rằng việc thực hiện không gian trạng thái theo (25.2) sẽ dễ dàng khái quát các mô tả mô hình của các hệ thống nhiều đầu vào nhiều đầu ra.

Để minh họa các khái niệm, ta xét phương trình vi phân:

$$m\frac{d^2}{dt^2}y(t) + c\frac{d}{dt}y(t) + ky(t) = u(t) \tag{25.3}$$



HÌNH 25.1 Biểu diễn hệ xe kéo là hệ động một vật với khối lượng xe m, hằng số lò xo k và hằng số giảm chấn c

Các mô tả ứng xử động của hệ thống xe kéo được đưa ra trong hình 25.1. Phương trình vi phân (25.3) dựa trên định luật 2 Newton cho khối lượng m của xe với đầu ra vị trí $y(t)$, lực lò xo $ky(t)$, lực giảm chấn $c(d/dt)y(t)$ và đầu vào lực $u(t)$. So sánh với (25.1), có thể thấy rằng $n_a = 2 \geq n_b = 0$, biến (25.3) thành phương trình vi phân bậc hai. Phương trình vi phân có thể được biến đổi thành biểu diễn không gian trạng thái (25.2) bằng cách định nghĩa biến trạng thái:

$$x(t) := \begin{bmatrix} y(t) \\ \frac{d}{dt}y(t) \end{bmatrix}$$

bao gồm vị trí và tốc độ của khối lượng. Với biến trạng thái này, (25.3) có thể viết lại thành:

$$\frac{d}{dt}x(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{bmatrix}x(t) + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}x(t) + 0u(t)$$

từ đó dẫn ra một mô hình không gian trạng thái tương đương với (25.2). Trong trường hợp này, kích thước của ma trận trạng thái A là 2x2, ma trận đầu vào B là 2x1, ma trận đầu ra C là 1x2và ma trận cấp thẳng $D = 0$ là vô hướng.

Hệ thống thời gian rời rạc

Mô hình thời gian rời rạc xấp xỉ và mô tả ứng xử dữ liệu lấy mẫu của hệ thống động thời gian liên tục. Trong một số ứng dụng, như điều khiển số, hệ thống điều khiển động đã là rời rạc. Trong trường hợp này, cần thiết phải phân tích với mô hình rời rạc tương đương.

Với mục đích phân tích, cả đầu vào $u(t)$ và đầu ra $y(t)$ giả thiết được lấy mẫu trong các khoảng thời gian rời rạc đều nhau

$$t = k\Delta T, \quad k = 0,1,2,3....$$

trong đó ΔT là thời gian lấy mẫu. Để thống nhất các kí hiệu trong suốt quá trình phân tích, thời gian lấy mẫu ΔT được chuẩn hóa tới Δ=1 và sự phụ thuộc thời gian t giả thiết được rời rạc với t = k = 0,1,2,....

Cho dữ liệu vào/ra rời rạc hoặc được lấy mẫu, thì mô hình rời rạc tuyến tính có thể được biểu diễn dưới dạng phương trình sai phân:

$$\sum_{j=0}^{n_c} c_k y(k+j) = \sum_{j=0}^{n_d} d_k u(k+j) \tag{25.4}$$

trong đó một tổ hợp tuyến tính được tạo thành từ các đầu vào $u(k)$ và các đầu ra $y(k)$ đã được dịch dịch theo thời gian dương. Để phân biệt phương trình vi phân từ các phương trình vi phân (25.1), các giá trị thực vô hướng c_j với $j = 0,\ldots,n_c, c_{n_c} \neq 0$ và d_j với $j = 0,\ldots,n_d, d_{n_d} \neq 0$ được sử dụng. Đầu vào $u(k)$ được phân biệt từ đầu ra $y(k)$ trong (25.1) bởi yêu cầu $n_c \geq n_d$. Kết quả là n_c là thời gian dịch lớn nhất của đầu ra $y(k)$ và n_c được sử dụng để chỉ ra bậc của phương trình sai phân.

Sự đơn giản mà với nó phương trình sai phân có thể được biểu diễn cũng cho phép biểu diễn (25.4) dưới dạng đại số. Việc đưa ra toán tử dịch thời gian:

$$qu(k) := u(k+1) \tag{25.5}$$

cho phép (25.4) có dạng biểu diễn khác dưới dạng đại số:

$$y(k)\sum_{j=0}^{n_c} c_j q^j = u(k)\sum_{j=0}^{n_d} d_j q^j$$

Tiếp theo việc phân tích này, đầu ra rời rạc $y(k)$ có thể được biểu diễn bằng mô hình sai phân:

$$y(k) = G(q)u(k) \text{ với } G(q) = \frac{\sum_{j=0}^{n_d} d_j q^j}{\sum_{j=0}^{n_c} c_j q^j} \tag{25.6}$$

trong đó các số thực vô hướng c_j với $j = 0,\ldots,n_c, c_{n_c} \neq 0$ và d_j với $j = 0,\ldots,n_d, d_{n_d} \neq 0$ chỉ ra từ số và mẫu số tương ứng.

Tương tự với hệ thống thời gian liên tục, phương trình sai phân bậc cao (25.4) cũng có thể viết lại thành tập các phương trình sai phân bậc nhất cho các mục đích phân tích. Điều này có thể thực hiện bằng cách đưa ra biến trạng thái $x(k)$ và phương trình được viết lại thành:

$$qx(k) = Fx(k) + Gu(k) \tag{25.7}$$

$$y(k) = Hx(k) + Ju(k)$$

trong đó $qx(k) = x(k+1)$, theo (25.5). Biến trạng thái $x(k)$ là một vectơ cột và chứa n_c biến, trong đó n_c là bậc của phương trình sai phân. Các ma trận không gian trạng thái trong (25.7) được kí hiệu khác đi để phân biệt chúng với mô hình không gian trạng thái thời gian liên tục.

Biến đổi Laplace và biến đổi z

Một khái niệm toán học quan trọng dùng trong phân tích mô hình được mô tả bởi phương trình vi phân tuyến tính như (25.1) và (25.2) là biến đổi Laplace. Như đã trình bày trước đó, biến đổi Laplace biến các phương trình vi phân tuyến tính thành các biểu thức đại số. Với phép biến đổi này, các thao tác đại số có thể sử dụng để khôi phục lời giải của phương trình vi phân. Tương tự, biến đổi z được sử dụng cho các mô hình rời rạc được mô tả bởi các phương trình sai phân. Mặc dù đã chỉ ra trong (25.6) rằng phương trình sai phân có thể được viết như một biểu thức đại số, biến đổi z cho phép phân tích phức tạp các mô hình rời rạc.

Biến đổi Laplace của tín hiệu $u(t)$ được định nghĩa là

$$\mathcal{L}\{u(t)\} := u(s) = \int_{t=0}^{\infty} u(t)e^{-st}dt \tag{25.8}$$

trong đó tích phân theo t sẽ loại trừ sự phụ thuộc thời gian và khi đó biến đổi $u(s)$ chỉ còn là hàm của biến Lapalce. Điều này được chỉ ra trong phép biến đổi u(s), khi sự phụ thuộc vào t bị bỏ qua và u(s) chỉ còn là hàm của biến Laplace (giá trị phức). Tích phân (25.8) tồn tại với hầu hết các tín hiệu thông dụng $u(t)$, với những điều kiện nào đó theo s. Để minh họa cho phép biến đổi ta xét tín hiệu bậc thang:

$$u(t) := \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

trong đó dạng của $u(t)$ giống với thay đổi bậc thang của tín hiệu đầu vào. Với định nghĩa của biến đổi Laplace trong (25.8) thì biến đổi của tín hiệu bậc thang trở thành:

$$u(s) = \int_{t=0}^{\infty} u(t)e^{-st}dt = \int_{t=0}^{\infty} e^{-st}dt = -\frac{e^{-st}}{s}\Big|_0^{\infty} = \frac{1}{s} \tag{25.9}$$

trong đó giả thiết là phần thực của s lớn hơn không, do đó $\lim_{t\to\infty} e^{st} = 0$

Nếu tín hiệu $u(k)$ có được tại các điểm thời gian rời rạc $k = 0,1,2\ldots$, thì công thức tích phân (25.8) không thể sử dụng được. Thay vào đó, một biến đổi tương tự với biến đổi Laplace có thể được dùng và được ký hiệu là biến đổi z. Biến đổi z của tín hiệu thời gian rời rạc $u(k)$ được định nghĩa là:

$$\mathcal{Z}\{u(k)\} := u(z) = \sum_{k=0}^{\infty} u(k)z^{-k} \tag{25.10}$$

Chuỗi (25.10) hội tụ nếu giả thiết rằng tồn tại giá trị r_l và r_u với $r_l < |z| < r_u$ là giới hạn biên độ của biến phức z.

Biến đổi z trong hệ thống thời gian rời rạc có cùng nguyên tắc như biến đổi Lapalace trong hệ thống thời gian liên tục. Trong trường hợp lấy mẫu, biến phức z của biến đổi z có liên quan với biến phức s của biến đổi Laplace qua:

$$z = e^{s\Delta T} \tag{25.11}$$

trong đó ΔT là thời gian lấy mẫu. Cả biến đổi Laplace và biến đổi z đều tuyến tính và thỏa mãn:

$$\mathcal{L}\{\alpha u(t) + \beta y(t)\} = \alpha L\{u(t)\} + \beta\{y(t)\} \tag{25.12}$$

Biến đổi của hầu hết các hàm thông dụng đã được tính toán trước và lập thành bảng bằng cách dùng định nghĩa trong (25.8) và tính chất tuyến tính trong (25.12).

Phân tích phương trình vi phân tuyến tính như (25.1) và (25.2) là biến đổi Laplace của đạo hàm:

$$\begin{aligned}\mathscr{L}\left\{\frac{d}{dt}u(t)\right\} &= \int_{t=0}^{\infty}\frac{d}{dt}u(t)e^{-st}dt \\ &= u(t)e^{-st}\Big|_0^{\infty} + \int_{t=0}^{\infty}u(t)e^{-st}dt \\ &= su(s) - u(0)\end{aligned}$$

Với $u(0) = 0$ có thể thấy rằng biến đổi Laplace của đạo hàm $u(t)$ chỉ đơn giản là s lần biến đổi Laplace của $u(s)$. Kết quả này có thể được triển khai với các đạo hàm bậc cao hơn và với đạo hàm bậc n thì kết quả là:

$$\mathscr{L}\left\{\frac{d^n}{dt^n}u(t)\right\} = s^n u(s) - \sum_{j=1}^{n} s^{n-j}\frac{d^{j-1}}{dt^{j-1}}u(t)\Big|_{t=0}$$

Trong trường hợp tín hiệu $u(t)$ thỏa mãn điều kiện ban đầu bằng không:

$$\frac{d^{j-1}}{dt^{j-1}}u(t)\Big|_{t=0} = 0 \text{ với } j = 1,\ldots,n$$

Công thức được rút gọn thành:

$$\mathscr{L}\left\{\frac{d^n}{dt^n}u(t)\right\} = s^n u(s)$$

và biến đổi Laplace của đạo hàm bậc n chỉ đơn giản là s^n lần biến đổi $u(s)$.

Với hệ thống thời gian rời rạc, biến đổi z của tín hiệu dịch theo thời gian sẽ được quan tâm. Tương tự như biến đổi Laplace, biến đổi z của tín hiệu dịch theo thời gian n lần có thể tính toán và được cho bởi:

$$\mathscr{Z}\{q^n u(k)\} = z^n u(z) - \sum_{j=0}^{n-1} z^{n-j}u(j)$$

Trong trường hợp tín hiệu rời rạc $u(k)$ thỏa mãn điều kiện ban đầu bằng không $u(j) = 0$ với $j = 0,\ldots,n-1$, thì công thức được rút gọn thành:

$$\mathscr{Z}\{q^n u(k)\} = z^n u(z)$$

và biến đổi z của tín hiệu rời rạc dịch theo thời gian n lần chỉ đơn giản là z^n lần biến đổi $u(z)$.

Mô hình hàm truyền

Kết quả biến đổi z và biến đổi Laplace có thể được sử dụng để rút gọn phương trình vi phân (25.1) và phương trình sai phân (25.4) thành các biểu thức đại số. Bắt đầu với phương trình vi phân cho mô hình liên tục và giả thiết điều kiện ban đầu bằng không cho cả đầu vào $u(t)$ và đầu ra $y(t)$, biến đổi Laplace của (25.1) sẽ là:

$$y(s)\sum_{j=0}^{n_a} a_j s^j = u(s)\sum_{j=0}^{n_b} b_j s^j$$

có thể viết lại ở dạng hàm truyền như sau:

$$Y(s) = G(s)u(s), \quad with\ G(s) = \frac{\sum_{j=0}^{n_b} b_j s^j}{\sum_{j=0}^{n_a} a_j s^j} \tag{25.13}$$

Trong (25.13), hàm truyền $G(s)$ là tỉ số của tử thức $\sum_{j=0}^{n_b} b_j s^j$ và mẫu thức $\sum_{j=0}^{n_a} a_j s^j$. Như đã chỉ ra ở trước, các số thực vô hướng a_j với $j = 0,\ldots n_a, a_{n_a} \neq 0$ và b_j với $j = 0,\ldots n_b, b_{n_b} \neq 0$, được gọi là các hệ số tử thức và mẫu thức tương ứng.

Tương tự với mô hình rời rạc, giả thiết điều kiện ban đầu bằng không với cả đầu vào $u(t)$ và đầu ra $y(t)$, biến đổi z cho (25.4) sẽ là:

$$y(z)\sum_{j=0}^{n_a} c_j z^j = u(z)\sum_{j=0}^{n_b} b_j z^j$$

có thể được viết lại ở dạng hàm truyền như sau:

$$y(z) = G(z)u(z), \text{ với } G(z) = \frac{\sum_{j=0}^{n_b} c_j z^j}{\sum_{j=0}^{n_a} a_j z^j} \tag{25.14}$$

Từ biểu diễn hàm truyền, các điểm cực và điểm không của hệ thống động có thể được tính để phân tích hệ thống. Các điểm cực của hệ thống được định nghĩa như là nghiệm của mẫu thức. Các điểm không của hệ thống được định nghĩa là nghiệm của tử thức.

Biến đổi Laplace và biến đổi z cũng có thể được sử dụng để rút gọn biểu diễn không gian trạng thái về một tập các biểu thức đại số bao gồm các đa thức bậc nhất. Giả thiết điều kiện ban đầu bằng không cho vectơ trạng thái $x(t)$, thì biến đổi Laplace cho (25.2) sẽ là:

$$sx(s) = Ax(s) + Bu(s)$$

$$y(s) = Cx(s) + Du(s)$$

trong đó vectơ trạng thái $x(t)$ có thể bị khử. Lời giải cho $x(s)$ đưa ra $x(s) = (sI - A)^{-1}Bu(s)$ và biến đổi trên đây có thể được viết lại thành một biểu diễn hàm truyền:

$$y(s) = G(s)u(s), \text{ với } G(s) = D + C(sI - A)^{-1}B \tag{25.15}$$

Với điều kiện kỹ thuật không khắt khe liên quan đến khả năng điều khiển được và quan sát được của mô hình không gian trạng thái, thì các biểu diễn hàm truyền trong (25.13) và (25.15) đều tương tự như trong trường hợp mô hình không gian trạng thái ở (25.2) nhận được từ phương trình vi phân (25.1) và ngược lại.

25.2 Đáp ứng động học

Biến đổi z và Laplace đưa ra khả năng tính toán đáp ứng động của hệ thống động học bằng các phép biến đổi đại số. Việc phân tích đáp ứng động mang lại sự hiểu biết sâu hơn về ứng xử động của hệ thống bằng cách phân tích đáp ứng với các tín hiệu kiểm tra điển hình như tín hiệu bậc thang (step), tín hiệu xung (impulse) và tín hiệu sin của hệ thống.

Có thể tính đáp ứng của các hệ thống thời gian liên tục tương đối đơn giản hoặc các hệ rời rạc được cho bởi các phương trình vi phân hoặc sai phân bậc thấp. Mô hình không gian trạng thái và mô tả hàm truyền đều đưa ra các cách biểu diễn tốt khi phân tích hệ thống động. Các kết quả sẽ được trình bày sau đây.

Đáp ứng xung và bậc thang

Một cách đánh giá đáp ứng của hệ thống động là sử dụng các tín hiệu kiểm tra cơ bản dạng xung và bậc thang. Đối với hệ liên tục, tín hiệu xung đầu vào được định nghĩa như là một hàm δ :

$$u_{imp}(t) = \delta(t) = \begin{bmatrix} \infty, & t = 0 \\ 0 & t \neq 0 \end{bmatrix}$$

với tính chất:

$$\int_{t=-\infty}^{\infty} f(t)\delta(t) = f(0)$$

trong đó $f(t)$ là hàm khả tích trên $(-\infty, \infty)$. Mặc dù tín hiệu xung là không thực tế theo quan điểm thực nghiệm nhưng tính toán hoặc mô phỏng của đáp ứng xung sẽ đưa ra những hiểu biết về ứng xử quá độ của hệ thống động.

Với các thuộc tính của hàm xung $\delta(t)$ được đề cập ở trên, biến đổi Laplace của hàm xung được cho bởi:

$$\mathcal{L}\{\delta(t)\} = \delta(s) = \int_{t=0}^{\infty} \delta(t)e^{-st}dt = e^{-s0} = 1$$

Vì vậy, đầu ra $y(s)$ do một đầu vào xung được cho bởi $y_{imp}(s) = G(s)u_{imp}(s) = G(s)\delta(s) = G(s)$. Từ đó có biến đổi Laplace ngược tức thời của hàm truyền liên tục $G(s)$:

$$y_{imp}(t) = \mathcal{L}^{-1}\{G(s)\}$$

cho phép tính đáp ứng động $y_{imp}(t)$ của hệ thống ứng với đầu vào xung.

Việc tính đáp ứng bậc thang được thực hiện theo cách tương tự. Theo (25.9), biến đổi Laplace của tín hiệu bậc thang là:

$$u_{step}(t) := \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

được cho như $u_{step}(s) = 1/s$.Vì vậy, với $y_{step}(s) = G(s)u_{step}(s) = G(s)/s$, biến đổi Laplace ngược của $G(s)/s$:

$$y_{step}(t) = \mathcal{L}^{-1}\left\{\frac{G(s)}{s}\right\}$$

sẽ tạo ra đáp ứng động $y_{step}(t)$ của hệ thống với đáp ứng đầu vào bậc thang.

Trên quan điểm thực tế, việc tính biến đổi Laplace ngược đã bị giới hạn ở các mô hình bậc thấp (bậc một hoặc hai). Tuy nhiên, kết quả đã đưa ra sự hiểu biết về ứng xử tổng quát của hầu hết các hệ thống động. Điều này được minh họa trong ví dụ sau.

- Xét mô hình liên tục bậc một cho bởi hàm truyền:

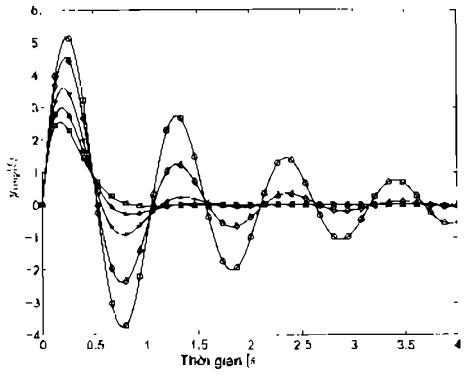
$$G(s) = \frac{K}{\tau s + 1}$$

trong đó K và τ tương ứng là hệ số khuếch đại tĩnh và hằng số thời gian của hệ thống. Một hàm truyền như vậy có thể có từ mạng RC đơn giản với $\tau = RC$. Để tính đáp ứng bậc

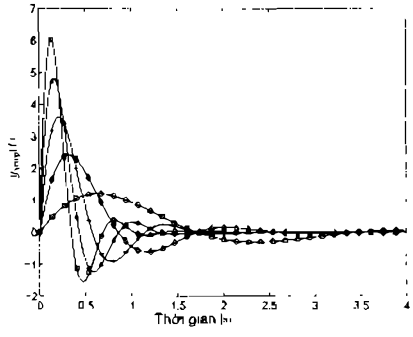
thang của hệ thống, biến đổi Laplace ngược của $G(s)/s$ cần được tính toán. Biến đổi Laplace ngược này được cho bởi:

$$y_{step}(t) = \mathcal{L}^{-1}\left\{\frac{G(s)}{s}\right\} = \frac{K}{\tau}(1 - e^{-t/\tau})$$

và có thể thấy rằng đáp ứng bậc thang là một hàm mũ. Để ổn định, hằng số thời gian τ cần thỏa mãn τ>0. Cũng có thể thấy rằng hằng số thời gian càng nhỏ thì tốc độ đáp ứng càng nhanh.



HÌNH 25.2 Biến thiên trong đáp ứng xung $y_{imp}(t)$ của hệ bậc hai với $\omega_n = 6$ và $\beta = 0.1(\bigcirc), 0.2(\diamond), 0.4(+), 0.6(*), 0.8(\square)$



HÌNH 25.3 Biến thiên trong đáp ứng xung $y_{imp}(t)$ của hàm bậc hai với $\beta = 0,4$ và $\omega_n = 2(\bigcirc), 4(\diamond), 6(+), 8(*), 10(\square)$

- Xét mô hình liên tục bậc hai cho bởi hàm truyền:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\beta\omega_n s + \omega_n^2} \tag{25.16}$$

trong đó ω_n và β tương ứng là tần số cộng hưởng không tắt dần và hệ số tắt dần của hệ thống. Mô hình này nhận được từ ứng xử động của hệ xe kéo đã được mô tả trong hình 25.1 và cho bởi (25.3). Với β<1 (dưới tắt dần), biến đổi Laplace ngược của $G(s)$ được cho bởi:

$$y_{imp}(t) = \frac{\omega_n}{\sqrt{1-\beta^2}} e^{-\beta\omega_n t} \sin(\omega_n \sqrt{1-\beta^2}\, t)$$

Từ công thức này có thể thấy rằng đáp ứng là một hình sin suy giảm với tần số cộng hưởng là $\omega_n\sqrt{1-\beta^2}$. Để ổn định, thì phải thỏa mãn $\omega_n > 0$ và $\beta > 0$, ω_n càng lớn thì tín hiệu hình sin càng giảm nhanh, tần số của đáp ứng $y_{imp}(t)$ càng cao. Đáp ứng xung của hệ bậc hai được mô tả trong hình 25.2 và 25.3 trong đó sự biến thiên của tần số cộng hưởng không tắt dần ω_n và hệ số tắt dần β minh họa ứng xử động của hệ thống.

Với hệ thống thời gian rời rạc, phân tích đáp ứng xung được dựa trên hàm xung rời rạc:

$$u_{imp}(k) := \delta(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$$

trong đó có giá trị bằng 1 tại $k = 0$ và bằng không với các giá trị khác của k. Tín hiệu bậc thang tương đương với tín hiệu liên tục và được cho bởi:

$$u_{step}(k) := \begin{cases} 1, & k < 0 \\ 0, & k \geq 1 \end{cases}$$

Để mô tả đáp ứng xung rời rạc và bậc thang, một qui trình tương tự như với mô hình liên tục có thể được dẫn ra bằng cách sử dụng biến đổi z. Có thể dễ dàng thấy biến đổi z của tín hiệu xung là $u_{imp}(z) = 1$ và biến đổi z của tín hiệu bậc thang là $u_{step}z = z/(z-1)$. Vì vậy đáp ứng của hệ thống thời gian rời rạc với một tín hiệu xung hoặc bậc thang là:

$$y_{imp}(k) = \mathcal{Z}^{-1}\{G(z)\}, \qquad y_{step}(k) = \mathcal{Z}^{-1}\left\{\frac{G(z)z}{z-1}\right\}$$

Ngoài ra, phương pháp sử dụng biến đổi z, tỉ số của các đa thức trong mô hình sai phân (25.6) có thể được viết theo một khai triển dạng chuỗi:

$$G(q) = \frac{\sum_{j=0}^{n_d} d_j q^j}{\sum_{j=0}^{n_c} c_j q^j} = \sum_{j=0}^{\infty} g_k q^{-k}$$

Với hàm xung rời rạc $u_{imp}(k)$ là đầu vào, có thể thấy rằng:

$$y_{imp}(k) = \sum_{j=0}^{\infty} g_k q^{-k}\delta(k) = g_k$$

và có thể kết luận rằng đáp ứng xung $y_{imp}(k)$ bằng các hệ số trong khai triển chuỗi của phương trình sai phân. Tương tự như vậy, với hàm bậc thang rời rạc $u_{step}(k)$ là đầu vào, có thể thấy rằng:

$$y_{imp}(k) = \sum_{j=0}^{\infty} g_k q^{-k} u_{step}(k) = \sum_{j=0}^{k} g_k$$

và có thể kết luận rằng các giá trị của đáp ứng bậc thang $y_{step}(k)$ được tính là tổng hữu hạn của các hệ số trong khai triển chuỗi của phương trình sai phân. Việc tính đáp ứng xung rời rạc cho mô hình rời rạc bậc nhất được đưa ra trong ví dụ sau.

- Xét mô hình rời rạc bậc nhất cho bởi mô hình sai phân:

$$G(q) = \frac{1}{q+d}$$

trong đó d là hằng số thời gian rời rạc của hệ thống. Khai triển chuỗi của mô hình sai phân được tính như sau:

$$G(q) = \frac{1}{q-d} = \sum_{j=0}^{\infty} d^j$$

và có thể thấy đáp ứng xung rời rạc:

$$y_{xung}(k) = d^k$$

là hàm mũ. Để ổn định, hằng số rời rạc d cần thỏa mãn $|d| < 1$. Tương tự như mô hình liên tục, thấy rằng hằng số thời gian càng nhỏ, đáp ứng càng nhanh. Hơn thế nữa, mô hình rời rạc bậc nhất có thể biểu diễn dao động trong trường hợp $-1 < d < 0$.

Tín hiệu sin và đáp ứng tần số

Ở trên chúng ta đã xét các hiệu ứng quá độ được tạo ra bởi các tín hiệu đầu vào bậc thang, xung rời rạc và xung đơn vị để nghiên cứu các thuộc tính động của một hệ thống động. Tuy nhiên, các đầu vào tuần hoàn thường xuyên gặp trong thực tế và sự phân tích hệ thống động với các đầu vào tuần hoàn và đặc biệt là đầu vào dạng sin có thể giúp ta hiểu sâu về ứng xử của hệ thống.

Đáp ứng của một hệ thống tuyến tính với một đầu vào hình sin được gọi là đáp ứng tần số của hệ thống. Tín hiệu đầu vào, $u(t) = U\sin\omega t$, có dạng sin với biên độ U và tần số ω_j, có biến đổi Laplace:

$$u(s) = \frac{U\omega}{s^2 + \omega^2}$$

Vì vậy, đáp ứng của hệ thống được cho bởi:

$$y(s) = G(s)\frac{U\omega}{s^2 + \omega^2}$$

và khai triển phân số từng phần của $y(s)$ sẽ đưa ra ứng xử quá độ của $y(s)$ và thành phần liên quan đến tín hiệu đầu vào $u(s)$ có dạng sin. Khử hiệu ứng quá độ và thực hiện biến đổi Laplace ngược sẽ tạo ra một đáp ứng tuần hoàn $y(t)$ với cùng một tần số ω_j:

$$y(t) = A\,U\sin(\omega t + \phi)$$

trong đó biên độ A và dịch pha ϕ được cho bởi:

$$A = G(s)\Big|_{s=\omega} \qquad \phi = \angle G(s)\Big|_{s=\omega} \tag{25.17}$$

Bằng việc đánh giá hàm truyền $G(s)$ theo trục ảo $s = i\omega, \omega \geq 0$, độ lớn $|G(j\omega)|$ đưa ra thông tin về độ khuếch đại tương đối của đầu vào dạng sin, trong khi đó pha $\angle G(j\omega)$ đưa ra thông tin về độ dịch pha tương đối giữa đầu vào và đầu ra.

Phân tích này có thể dễ dàng triển khai với các hệ thống thời gian rời rạc bằng cách thực hiện mối quan hệ giữa biến s của Laplace và biến của biến đổi z trong (25.11) để nhận được đáp ứng sin rời rạc:

$$y(k) = A\,U\sin(\omega k + \phi)$$

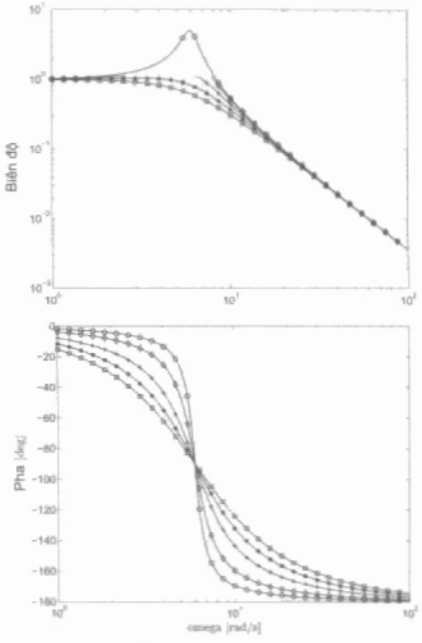
trong đó độ lớn của A và độ dịch pha ϕ cho bởi:

$$A = G(z)\Big|_{z=e^{i\omega T_s}} \qquad \phi = \angle G(z)\Big|_{z=e^{i\omega T_s}} \tag{25.18}$$

Do bản chất lấy mẫu của hệ thống thời gian rời rạc, nên bây giờ hàm truyền $G(z)$ được đánh giá trên vòng tròn đơn vị:

$$e^{i\Delta T\omega},\ 0 \le \omega < \frac{\pi}{\Delta T}$$

để có được thông tin về biên độ và dịch pha của đáp ứng sin.



HÌNH 25.4 Biến thiên trong đáp ứng tần số của hệ thống bậc hai $G(s)$ với $\omega_n = 6$ và $\beta = 0.1(\mathrm{O}), 0.2(\Diamond), 0.4(+), 0.6(*), 0.8(\square)$

Biểu diễn đáp ứng tần số của một hệ thống động cho phép hiểu rõ về vị trí điểm cực (chế độ cộng hưởng) và vị trí điểm không của hệ thống động. Như trong ví dụ, đáp ứng tần số của hệ thống bậc hai cho trong (25.16) được vẽ trong hình 25.4. Từ hình vẽ có thể thấy rằng, hệ thống bậc hai suy giảm ít hơn khi hệ số suy giảm β nhỏ hơn và điều này dẫn đến đáp ứng biên độ của hệ thống bậc hai sẽ lớn hơn tại tần số $\omega_n = 6$ rad/s. Cũng có thể thấy rằng thay đổi pha ở tần số cộng hưởng trở nên đột ngột hơn khi các hệ số suy giảm nhỏ hơn.

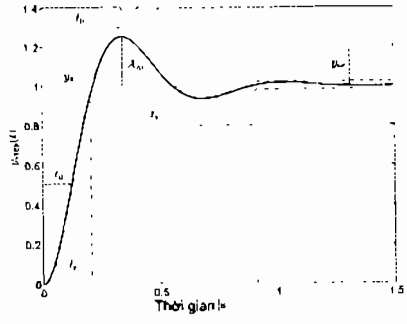
25.3 Các thông số hoạt động của hệ thống động

Các thông số của đáp ứng bậc thang

Các đặc tính của các hệ thống động thường bao gồm những yêu cầu về ứng xử quá độ của hệ thống. Các yêu cầu của ứng xử quá độ có thể được tính toán dựa trên một đáp ứng bậc thang và các thông số quan trọng nhất được tóm tắt sau đây và được minh họa trong hình 25.5.

- Giá trị trạng thái ổn định hoặc giá trị *DC* y_s của đầu ra đáp ứng bậc thang.
- Sai số trạng thái ổn định y_{se} là sai số giữa giá trị trạng thái ổn định y_s và giá trị đặt *DC* của đầu ra đáp ứng bậc thang.

- Độ quá điều chỉnh cực đại A_m là độ lệch tối đa của đầu ra đáp ứng bậc thang ở trên giá trị trạng thái ổn định y_s của nó.
- Thời gian đỉnh t_p là thời gian mà tại đó xuất hiện điểm quá điều chỉnh lớn nhất.
- Thời gian quá độ t_s là thời gian mà tại đó đầu vào đáp ứng bậc thang nằm trong khoảng vài phần trăm giá trị trạng thái ổn định y_s. Thông thường, phạm vi 2% hoặc 5% được chọn để xác định thời gian quá độ.
- Thời gian tăng t_r được định nghĩa là thời gian cần thiết để đầu ra đáp ứng bậc thang tăng từ 10% đến 90% của giá trị trạng thái ổn định y_s.
- Thời gian trễ t_d được định nghĩa là thời gian cần thiết để đạt được 50% giá trị trạng thái ổn định y_s.



HÌNH 25.5 Các thông số cho đáp ứng bậc thang: giá trị trạng thái ổn định y_s, sai số trạng thái ổn định y_{ss}, độ quá điều chỉnh A_m, thời gian đỉnh t_p, thời gian quá độ t_s, thời gian tăng t_r và thời gian trễ t_d

Hầu hết các giá trị trên đây có thể nhận được từ thực nghiệm. Nói chung, chúng không thể nhận được ở dạng giải tích, ngoại trừ với các mô hình bậc thấp. Với mô hình bậc hai của hệ thống một khối lượng cho trong (25.3), có thể nhận được một số kết quả giải tích như độ quá điều chỉnh lớn nhất A_m được xác định bởi:

$$A_m = 100e^{-\pi\beta}/\sqrt{1-\xi^2}, \text{trong đó } \xi = \frac{A}{\sqrt{\pi^2 + A^2}}, A = \ln\left(\frac{100}{A_m}\right)$$

Thời gian đỉnh t_p có thể được tính bởi:

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\xi^2}}$$

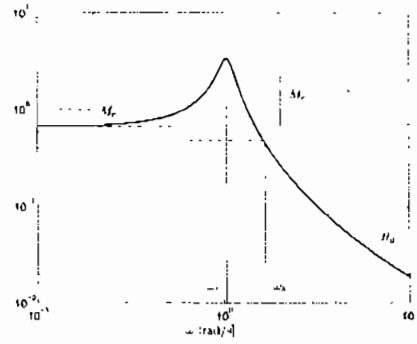
thời gian trễ t_d được xấp xỉ bởi:

$$t_d \approx \frac{1+0.7\xi}{\omega_n}$$

Khi độ quá điều chỉnh lớn nhất tăng lên với hệ số suy giảm β nhỏ hơn, thì độ quá điều chỉnh lớn nhất thường được sử dụng để chỉ ra độ ổn định tương đối của hệ thống.

Các thông số miền tần số

Với phân tích miền tần số của các hệ thống động, đặc tính của các thuộc tính động của một hệ thống cũng có thể được phát biểu ở miền tần số. Đặc tính tần số thường được chú ý khi thiết kế bộ lọc là các thông số về độ gợn sóng, dải thông và độ chậm pha. Các đặc tính tương tự cũng có thể được xác định cho hệ thống động trong trường hợp mô hình của hệ thống được phân tích trong miền tần số. Các thông số quan trọng nhất được tóm tắt dưới đây và được minh họa trong hình 25.6.



HÌNH 25.6 Các thông số cho đáp ứng tần số: dải thông ω_b, tần số cộng hưởng ω_r, đỉnh cộng hưởng M_r, sai số trạng thái tĩnh M_e

- Dải thông ω_b là một khái niệm có tần số lớn nhất mà tại đó đầu ra sẽ bám theo đầu vào dạng sin theo một cách mong muốn. Thông thường dải thông được định nghĩa là tần số mà tại đó đầu ra bị suy giảm -3dB (0,707).
- Tần số cộng hưởng ω_r là tần số đầu tiên mà tại đó xảy ra hiện tượng cộng hưởng đáng kể với suy giảm thấp. Nếu không được điều khiển, dạng cộng hưởng có thể có ảnh hưởng xấu tới thời gian quá độ của hệ thống động và nó giữ vai trò quan trọng trong việc mô tả quá trình.
- Đỉnh cộng hưởng M_r là chiều cao của dạng cộng hưởng. Đỉnh cộng hưởng là số đo sự suy giảm. Như minh họa trong hình 25.2 với mô hình bậc hai, dạng cộng hưởng tăng lên với các hệ số suy giảm thấp hơn.
- Sai số trạng thái tĩnh M_e cũng có thể được phân tích theo đáp ứng tần số của hệ thống. Bằng cách dùng định lý giá trị mong muốn cho hệ thống thời gian liên tục:

$$\lim_{t\to\infty} y(t) = y_s = \lim_{s\to 0} sy(s)$$

sự tồn tại của các sai số trạng thái tĩnh có thể được xem xét trong miền tần số bằng cách đánh giá $|G(s)|$ tại $s = i\omega = 0$ hoặc các giá trị nhỏ của vectơ tần số ω. Khi biến đổi Laplace $u_{step}(s)$ của đầu vào bậc thang $u_{step}(t)$ là $u_{step}(s) = 1/s$ thì:

$$\lim_{t\to\infty} y_{step}(t) = \lim_{s\to 0} sy_{step}(s) = \lim_{s\to 0} sG(s)\frac{1}{s} = \lim_{s\to 0} G(s)$$

Bằng cách đánh giá $|G(i\omega)|$ với tần số nhỏ ω, có thể nghiên cứu ứng xử trạng thái tĩnh của $G(s)$.

Tương tự với các hệ thống thời gian rời rạc, tại đó định lý giá trị mong muốn được phát biểu như sau:

Nếu $u(z)$ hội tụ với $|z|>1$ và tất cả các điểm cực của $(z-1)u(z)$ đều nằm trong vòng tròn đơn vị, thì:

$$\lim_{k\to\infty} u(k) = \lim_{z\to 1}(z-1)u(z)$$

Vì vậy, với các hệ thời gian rời rạc, ứng xử trạng thái tĩnh của hàm truyền $G(s)$ có thể được nghiên cứu bằng cách đánh giá $|G(i\omega)|$ với các tần số ω nhỏ.

26

Phương pháp quỹ đạo nghiệm

26.1 Giới thiệu 26-1
26.2 Các vị trí điểm cực mong muốn 26-4
26.3 Xây dựng quỹ đạo nghiệm 26-6
26.4 Quỹ đạo nghiệm bù 26-16
26.5 Quỹ đạo nghiệm cho hệ thống có trễ thời gian 26-16
26.6 Chú ý và tham khảo

Hitay Özbay
The Ohio State University

26.1 Mở đầu

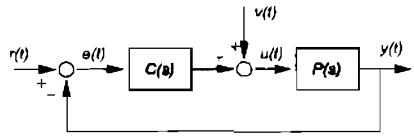
Quỹ đạo nghiệm là phương pháp đồ họa được sử dụng trong việc phân tích và thiết kế hệ thống phản hồi. Phương pháp này đã được công bố chính thức bởi W.R.Evans [3, 4], người đã nhận giải thưởng Richard E.Bellman về điều khiển từ Hội Đồng Điều Khiển Tự Động Mỹ năm 1988 cho đóng góp quan trọng này.

Để thảo luận về phương pháp quỹ đạo nghiệm, trước tiên ta phải xem lại định nghĩa cơ bản về tính ổn định của đầu vào hữu hạn và đầu ra hữu hạn (BIBO) của hệ thống phản hồi bất biến thời gian tuyến tính chuẩn đã được chỉ ra trong hình 26.1, trong đó đối tượng điều khiển và bộ điều khiển được biểu diễn tương ứng bởi các hàm truyền đạt $P(s)$ và $C(s)$. Đối tượng điều khiển $P(s)$ gồm quá trình vật lý được điều khiển, cũng như cơ cấu chấp hành và động lực học cảm biến.

Hệ thống phản hồi được gọi là ổn định nếu không có các hàm truyền kín từ các đầu vào ngoại vi r và v tới các tín hiệu nội tại e và u, có điểm cực nào trong nửa mặt phẳng đóng bên phải, $\overline{\mathbb{C}}_+ := \{s \in \mathbb{C} : \mathrm{Re}(s) \geq 0\}$. Điều kiện cần thiết để ổn định hệ thống có phản hồi là các điểm không của $P(s)$ (tương ứng với $C(s)$) ở nửa mặt phẳng đóng bên phải đều khác biệt so với các điểm cực của $C(s)$ (tương ứng với $P(s)$). Khi có điều kiện này, ta nói rằng không có sự mất ổn định do triệt tiêu điểm không trong việc thiết lập $P(s)C(s) =: G(s)$, khi đó kiểm tra tính ổn định của hệ thống có phản hồi trở thành tương đương với việc kiểm tra tất cả các nghiệm của:

$$1 + G(s) = 0 \tag{26.1}$$

đều nằm trong nửa mặt phẳng mở bên trái, $\mathbb{C}_- := \{s \in \mathbb{C} : \mathrm{Re}(s) < 0\}$. Các nghiệm của (26.1) là các điểm cực của hệ kín. Ta tìm hiểu xem vị trí của các điểm cực của hệ điều khiển vòng kín thay đổi như là hàm của một thông số thực của $G(s)$ như thế nào. Cụ thể hơn, giả thiết rằng $G(s)$ chứa thông số K, vì vậy ta sử dụng kí hiệu $G(s) = G_K(s)$ để nhấn mạnh sự phụ thuộc vào K. *Quỹ đạo nghiệm* là đồ thị các nghiệm của (26.1) trên mặt phẳng phức, khi thông số K thay đổi trong một khoảng xác định.



HÌNH 26.1 Hệ thống phản hồi đơn cơ bản

Ví dụ thường gặp về vấn đề quỹ đạo nghiệm gắn liền với hệ số khuếch đại không chắc chắn như là một thông số thay đổi: khi $P(s)$ và $C(s)$ là các hàm hữu tỷ trừ thành phần khuếch đại thì $G(s)$ có thể được viết là $G(s) = G_K(s) = KF(s)$, trong đó K là hệ số khuếch đại điều chỉnh được, và:

$$F(s) = \frac{N(s)}{D(s)} \quad \text{với} \quad N(s) = \prod_{j=1}^{m}(s - z_j) \qquad n \geq m \tag{26.2}$$

$$D(s) = \prod_{i=1}^{n}(s - p_i)$$

với $z_1, \ldots, z_m$ và $p_1, \ldots, p_n$ là điểm cực và điểm không của hệ kín. Trong trường hợp này, các điểm cực là nghiệm của phương trình đặc tính:

$$\chi(s) := D(s) + KN(s) = 0 \tag{26.3}$$

Quỹ đạo nghiệm thông thường nhận được bằng cách vẽ các nghiệm $r_1(K), \ldots, r_n(K)$ của đa thức đặc tính $\chi(s)$ trên mặt phẳng phức, với K thay đổi từ 0 đến $+\infty$. Tương tự đối với giá trị âm của K cho *quỹ đạo nghiệm bù*. Với sự trợ giúp của đồ thị quỹ đạo nghiệm, người thiết kế xác định các giá trị có thể chấp nhận được của thông số K dẫn đến một tập các điểm cực của hệ kín trong "vùng mong muốn" trên mặt phẳng phức. Có một vài yếu tố phải được xem xét theo khái niệm "vùng mong muốn" của mặt phẳng phức mà $r_1(K), \ldots, r_n(K)$ cần nằm ở đó. Điều này sẽ được đề cập đến trong phần sau. Phần 26.3 đưa ra các thủ tục xây dựng quỹ đạo nghiệm và các ví dụ thiết kế được đưa ra trong phần 26.4.

Ngoài hệ số K ra, quỹ đạo nghiệm cũng có thể được vẽ với các thông số hệ thống khác. Ví dụ phương trình đặc tính cho hệ thống $G(s) = G_\lambda(s)$ cho bởi:

$$G_\lambda(s) = P(s)C(s), \qquad P(s) = \frac{(1 - \lambda s)}{s(1 + \lambda s)}, \qquad C(s) = K_c\left(1 + \frac{1}{T_I s}\right)$$

cũng có thể được chuyển thành dạng cho trong (26.3). Ở đây K_c và T_I là các thông số của bộ điều khiển PI (tỉ lệ - tích phân) và $\lambda > 0$ là thông số thay đổi của đối tượng điều khiển. Chú ý rằng pha của đối tượng là:

$$\angle P(j\omega) = -\frac{\pi}{2} - 2\tan^{-1}(\lambda\omega)$$

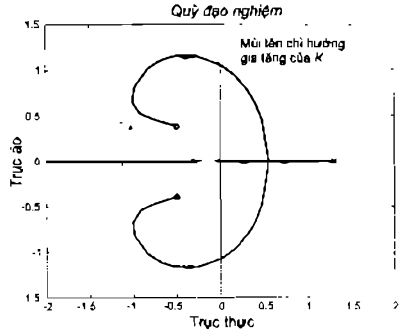
Do đó thông số λ có thể coi là hệ số trễ pha không chắc chắn (ví dụ, một trễ thời gian nhỏ không chắc chắn của đối tượng điều khiển có thể được mô hình hóa theo cách này, xem [9]). Dễ thấy rằng phương trình đặc tính là:

$$s^2(\lambda s + 1) + K_c(1 - \lambda s)\left(s + \frac{1}{T_I}\right) = 0$$

Và bằng cách sắp xếp lại các hệ số bằng việc nhân λ, phương trình này có thể biến đổi thành:

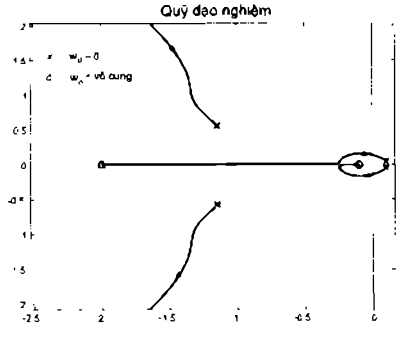
$$1+\frac{1}{\lambda}\frac{(s^2+K_c s+K_c/T_1)}{s(s^2-K_c s-K_c/T_1)}=0$$

Từ định nghĩa $K=\lambda^{-1}$, $N(s)=(s^2+K_c s+K_c/T_1)$ và $D(s)=s(s^2-K_c s-K_c/T_1)$, ta thấy phương trình đặc tính có thể viết ở dạng (26.3). Bây giờ, biểu đồ quỹ đạo nghiệm có thể nhận được từ dữ liệu $N(s)$ và $D(s)$ đã được định nghĩa ở trên; điều đó chỉ ra cách mà các cực của hệ kín di chuyển khi λ^{-1} thay đổi từ 0 tới $+\infty$, với tập cố định các thông số bộ điều khiển đã cho K_c và T_1. Ví dụ với K_c=1 và T_1=2,5, quỹ đạo nghiệm được biểu diễn ở hình 26.2.



HÌNH 26.2 Quỹ đạo nghiệm tương ứng với $K=1/\lambda$

Qui trình xây dựng quỹ đạo nghiệm sẽ được đưa ra ở phần 26.3. Hầu hết các tính toán liên quan trong mỗi bước của qui trình này đều có thể thực hiện bằng tay. Vì vậy một đồ thị xấp xỉ biểu diễn quỹ đạo nghiệm có thể vẽ dễ dàng. Có một số phần mềm có thể tự động tạo ra quỹ đạo nghiệm từ các số liệu $z_1,\ldots,z_m$ và $p_1,\ldots,p_n$.



HÌNH 26.3 Quỹ đạo nghiệm tương ứng với ω_0

Nếu có sẵn một chương trình số để tính nghiệm của một đa thức, thì ta cũng có thể nhận được quỹ đạo nghiệm tương ứng với một thông số sẽ được nhập vào trong phương trình đặc tính một cách không tuyến tính. Để minh họa, ta xét ví dụ sau: $G(s)=G_{\omega_0}(s)$ trong đó:

$$G_{\omega_0}(s)=P(s)C(s),\quad P(s)=\frac{(s-0,1)}{(s^2+1,2\omega_0 s+\omega_0^2)(s+0,1)},\ C(s)=\frac{(s-0,2)}{(s+2)}$$ ở đây, $\omega_0\geq 0$ là thông số có thể thay đổi của đối tượng. Chú ý rằng phương trình đặc tính:

$$1+\frac{\omega_0(1,2s+\omega_0)(s+0,1)(s+2)}{s^2(s+0,1)(s+2)+(s-0,2)(s-0,1)}=0 \qquad (26.4)$$

không thể biểu diễn được dưới dạng $D(s)+KN(s)=0$ với chỉ một thông số K. Tuy nhiên, với mỗi ω_0 ta có thể tính nghiệm của (26.4) theo phương pháp số và vẽ chúng trên mặt phẳng phức khi ω_0 thay đổi trong phạm vi cần thiết. Hình 26.3 minh họa cả bốn nhánh $r_1(K),\ldots,r_4(K)$ của quỹ đạo nghiệm cho hệ thống này khi ω_0 tăng từ không tới vô cùng. Hình nhận được bằng cách tính nghiệm của (26.4) với một tập các giá trị của ω_0, nhờ sử dụng MATLAB.

26.2 Các vị trí điểm cực mong muốn

Biểu diễn hệ thống phản hồi phụ thuộc nhiều vào vị trí các điểm cực của hệ điều khiển kín $r_i(K)=1,\ldots,n$. Trước hết, để ổn định ta muốn $r_i(K)\in\mathbb{C}_-$ với mọi $i=1,\ldots,n$. Rõ ràng, có một điểm cực "gần" trục ảo thì sẽ nguy hiểm, có nghĩa là những sự bất ổn nhỏ của đối tượng cũng có thể dẫn đến hệ thống phản hồi không ổn định. Do đó vị trí điểm cực mong muốn phải sao cho tính ổn định được duy trì trước những bất ổn như vậy (hoặc với sự tồn tại của những thông số để thay đổi) của đối tượng. Với các hệ bậc hai, ta có thể xác định các phép đo thô của tính ổn định chắc chắn theo các vị trí điểm cực có thể được gắn với đặc tính của đáp ứng bước. Với các hệ bậc cao hơn, cách tương tự chỉ có thể sử dụng với các điểm cực trội.

Với hệ điều khiển phản hồi cơ bản trong hình 26.1, giả thiết rằng hàm truyền đạt vòng kín từ $r(t)$ tới $y(t)$ có dạng sau:

$$T(s)=\frac{\omega_0^2}{s^2+2\zeta\omega_0 s+\omega_0^2},\quad 0<\zeta<1,\quad \omega_0\in\mathbb{R}$$

và $r(t)$ là hàm bước đơn vị. Khi đó, đầu ra là:

$$y(t)=1-\frac{e^{-\zeta\omega_0 t}}{\sqrt{1-\zeta^2}}\sin(\omega_d t+\theta),\qquad t\ge 0$$

trong đó $\omega_d := \omega_0\sqrt{1-\zeta^2}$ và $\theta := \cos^{-1}(\zeta)$. Với một số giá trị thông thường của ζ, đáp ứng nhảy bậc $y(t)$ có dạng giống như trong hình 26.4. Độ quá điều chỉnh được định nghĩa là:

$$\text{PO} := \frac{y_p-y_{ss}}{y_{ss}}\times 100\%$$

trong đó y_p là giá trị đỉnh. Bằng tính toán đơn giản có thể thấy giá trị đỉnh của $y(t)$ xuất hiện tại thời gian $t_p=\pi/\omega_d$ và:

$$\text{PO} := e^{-\pi\zeta/\sqrt{1-\zeta^2}}\times 100\%$$

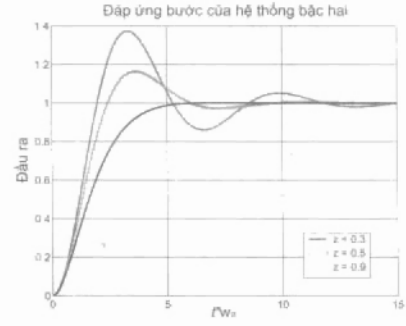
Hình 26.5 biểu diễn PO theo ζ. Thời gian xác lập được xác định là khoảng thời gian nhỏ nhất t_s, sau khi đáp ứng $y(t)$ nằm trong khoảng 2% giá trị mong muốn, có nghĩa là:

$$t(s) := \min\{t' : |y(t)-y_{ss}|\le 0{,}02y_{ss}\ \forall t\ge t'\}$$

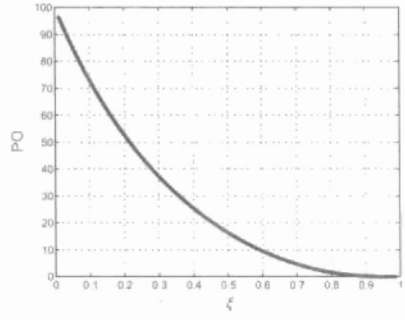
Đôi khi 1% hoặc 5% được sử dụng để xác định thời gian xác lập thay cho 2%, về cơ bản không có gì khác biệt. Với đáp ứng của hệ bậc hai, ta có:

$$t(s)\approx\frac{4}{\zeta\omega_0}$$

Vì vậy, để có một đáp ứng xác lập nhanh thì $\zeta\omega_0$ phải lớn.



HÌNH 26.4 Đáp ứng nhảy bậc của hệ bậc hai



HÌNH 26.5 PO và ζ

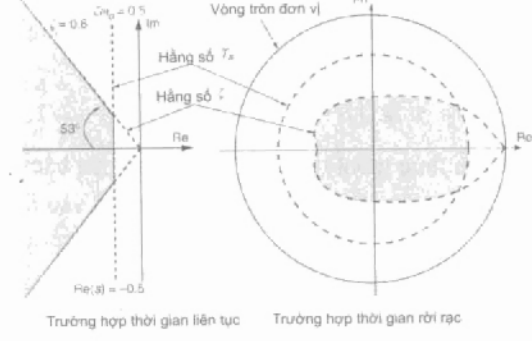
Các điểm cực của hệ kín là:

$$r_{1,2} = -\zeta\omega_0 \pm j\omega_0\sqrt{1-\zeta^2}$$

Vì vậy, khi thời gian xác lập lớn nhất cho phép và PO được xác định ta có thể xác định vùng các vị trí điểm cực mong muốn bằng cách xác định ζ và $\zeta\omega_0$ tối thiểu cho phép. Ví dụ, cho PO và t_s được giới hạn bởi:

$$\text{PO} \le 10\% \quad and \quad t_s \le 8s$$

Yêu cầu về PO ngụ ý rằng $\zeta \ge 6$, tương đương với $\theta \le 53^0$ (nhớ rằng $\cos(\theta) = \zeta$). Yêu cầu về thời gian xác lập được thỏa mãn nếu và chỉ nếu $\text{Re}(r_{1,2}) \le -0,5$. Vì vậy vùng các điểm cực của hệ kín là vùng đậm màu trong hình 26.6. Hình này cũng minh họa vùng các điểm cực mong muốn của hệ kín với các yêu cầu thiết kế tương tự trong trường hợp thời gian rời rạc.



HÌNH 26.6 Vùng của các điểm cực mong muốn của hệ kín

Nếu bậc của hàm truyền đạt vòng kín $T(s)$ lớn hơn hai, thì bậc đó sẽ phụ thuộc vào vị trí của các điểm cực và điểm không của nó, có thể xấp xỉ đáp ứng bước của hệ kín bằng đáp ứng của một hệ bậc hai. Ví dụ, xét hệ bậc ba:

$$T(s) = \frac{\omega_0^3}{(s^2 + 2\zeta\omega_0 s + \omega_0^2)(1 + s/r)} \text{ trong đó } r \gg \zeta\omega_0$$

Đáp ứng quá độ chứa số hạng e^{-rt}. So sánh với hình bao $e^{-\zeta\omega_0 t}$ của số hạng sin, e^{-rt} giảm rất nhanh và đáp ứng tổng thể tương tự như đáp ứng của hệ bậc hai. Vì vậy, tác động của điểm cực thứ ba $r_3 = -r$ là không đáng kể.

Xét ví dụ khác:

$$T(s) = \frac{\omega_0^2[1 + s/(r+\epsilon)]}{(s^2 + 2\zeta\omega_0 s + \omega_0^2)(1 + s/r)} \text{ trong đó } 0 < \epsilon \ll r$$

Trong trường hợp này, mặc dù r không cần lớn hơn nhiều so với $\zeta\omega_0$, điểm không tại $-(r+\epsilon)$ triệt tiêu tác động của điểm cực tại $-r$. Để thấy được điều này, xét khai triển từng phần $Y(s) = T(s)R(s)$ với $R(s) = 1/s$:

$$Y(s) = \frac{A_0}{s} + \frac{A_1}{s - r_1} + \frac{A_2}{s - r_2} + \frac{A_3}{s + r}$$

trong đó $A_0 = 1$ và:

$$A_3 = \lim_{s \to -r}(s + r)Y(s) = \frac{\omega_0^2}{2\zeta\omega_0 r - (\omega_0^2 + r^2)}\left(\frac{\epsilon}{r + \epsilon}\right)$$

Vì $|A_3| \to 0$ khi $\epsilon \to 0$ nên số hạng $A_3 e^{-rt}$ là không đáng kể trong $y(t)$.

Tóm lại, nếu có sự triệt tiêu điểm cực-không xấp xỉ ở nửa mặt phẳng trái, thì cặp điểm cực-không này có thể được lấy ra từ hàm truyền đạt $T(s)$ để xác định PO và t_s. Cũng như vậy, các điểm cực gần nhất với trục ảo có ảnh hưởng lớn tới đáp ứng quá độ của $y(t)$. Để khái quát nhận xét này, cho $r_1, \ldots, r_n$ là các điểm cực của $T(s)$, như vậy $\text{Re}(r_k) \ll \text{Re}(r_2) = \text{Re}(r_1) < 0$, với mọi $k \geq 3$ thì, cặp điểm cực liên hợp phức $r_{1,2}$ được gọi là *các điểm cực trội*. Ta đã thấy rằng các thuộc tính của đáp ứng quá độ mong muốn, như là PO và t_s, có thể chuyển đổi thành yêu cầu về vị trí của các điểm cực trội.

26.3 Xây dựng quỹ đạo nghiệm

Như đã nói ở trên, quỹ đạo nghiệm chủ yếu giải quyết việc tìm các nghiệm của đa thức đặc tính là một hàm của thông số K:

$$\chi(s) = D(s) + KN(s) \tag{26.5}$$

trong đó $D(s)$ và $N(s)$ là các đa thức monic (có nghĩa là hệ số của bậc cao nhất được chuẩn hóa về 1). Nếu N và/hoặc D không monic, hệ số cao nhất có thể nhập vào K.

Nguyên tắc quỹ đạo nghiệm

Cần nhắc lại rằng quỹ đạo nghiệm thông thường chỉ ra vị trí của các cực của hệ kín khi K thay đổi từ 0 đến $+\infty$. Nghiệm của $D(s)$, $p_1, \ldots, p_n$ là các cực và nghiệm của $N(s)$, $z_1, \ldots, z_m$ là các

điểm không, của hệ hở, $G(s) = KF(s)$. Vì $P(s)$ và $C(s)$ là chuẩn, $G(s)$ là chuẩn và vì vậy $n \geq m$. Do đó bậc của đa thức $\chi(s)$ là n và có chính xác n nghiệm.

Ký hiệu các cực của hệ kín là $r_1(K),\ldots,r_n(K)$. Chú ý rằng đó là các hàm của K; khi sự phụ thuộc vào K là rõ ràng, có thể viết đơn giản là $r_1,\ldots,r_n$. Các điểm trong $\mathbb{C}$ thỏa mãn (26.5) với một số $K>0$ nằm trên quỹ đạo nghiệm. Rõ ràng là một điểm $r \in \mathbb{C}$ nằm trên quỹ đạo nghiệm khi và chỉ khi:

$$K = -\frac{1}{F(r)} \tag{26.6}$$

Điều kiện (26.6) có thể được tách ra thành hai phần:

$$|K| = -\frac{1}{|F(r)|} \tag{26.7}$$

$$\angle K = 0 \ = -(2l+1)\times 180 \ - \angle F(r), \quad l = 0,\pm 1,\pm 2,\ldots \tag{26.8}$$

Quy tắc pha (26.8) xác định các điểm trong $\mathbb{C}$ nằm trên quỹ đạo nghiệm. Quy tắc về độ lớn (26.7) xác định hệ số khuếch đại $K>0$ mà tại đó quỹ đạo nghiệm là điểm r cho trước. Bằng cách sử dụng định nghĩa của $F(s)$, phương trình (26.8) có thể được viết lại như sau:

$$(2l+1)\times 180 \ = \sum_{i=1}^{n} \angle(r - p_i) - \sum_{j=1}^{m} \angle(r - z_j) \tag{26.9}$$

Tương tự, (26.7) tương đương với:

$$K = \frac{\prod_{i=1}^{n} |r - p_i|}{\prod_{j=1}^{m} |r - z_j|} \tag{26.10}$$

Xây dựng quỹ đạo nghiệm

Có một số phần mềm cho phép tạo quỹ đạo nghiệm tự động với $F = N/D$ cho trước. Cụ thể là lệnh `rlocus` và `rlocfind` trong MATLAB. Trong nhiều trường hợp, quỹ đạo nghiệm gần đúng được vẽ bằng tay sử dụng quy tắc được rút ra từ các định nghĩa cơ bản (26.5), (26.7) và (26.8) sau:

1. Quỹ đạo nghiệm có n nhánh: $r_1(K),\ldots,r_n(K)$.
2. Mỗi nhánh bắt đầu ($K \equiv 0$) tại điểm cực p_i và kết thúc (khi $K \to \infty$) tại một điểm không z_j, hoặc hội tụ về đường tiệm cận, $Me^{j\alpha_l}$, trong đó $M \to \infty$ và:

$$\alpha_l = \frac{2l+1}{n-m} \times 180 \ , \qquad l = 0,\ldots..,(n-m-1)$$

3. Có $(n-m)$ đường tiệm cận với các góc α_l. Tâm của đường tiệm cận (có nghĩa là điểm giao nhau trên trục thực của chúng) là:

$$\sigma_a = \frac{\sum_{i=1}^{n} p_i - \sum_{j=1}^{m} z_j}{n-m}$$

4. Một điểm $x \in \mathbb{R}$ nằm trên quỹ đạo nghiệm khi và chỉ khi tổng số điểm cực p_i và điểm không z_j ở bên phải của x (có nghĩa là tổng số điểm cực của p_i với $\mathrm{Re}(p_i) > x$ cộng

với tổng số điểm không z_i với $\mathrm{Re}(z_i) > x$) là lẻ. Vì $F(s)$ là hàm hữu tỷ với hệ số thực nên các điểm cực và điểm không xuất hiện trong liên hợp phức, do đó khi tính số điểm cực và điểm không ở bên phải của điểm $x \in \mathbb{R}$ ta chỉ cần xét các điểm cực và điểm không trên trục thực.

5. Các giá trị của K làm quỹ đạo nghiệm đi ngang qua trục ảo có thể xác định từ việc kiểm tra tính ổn định Routh-Hurwitz. Ta có thể đặt $s = j\omega$ trong (26.5) và giải với ω và K thực thỏa mãn:

$$D(j\omega) + KN(j\omega) = 0$$

Chú ý rằng ở đây có hai phương trình, một cho phần thực và một cho phần ảo.

6. Các điểm giao nhau của hai nhánh trên trục thực là các lời giải khả thi (thỏa mãn quy tắc 4) của:

$$\frac{d}{ds}F(s) = 0 \tag{26.11}$$

7. Góc của điểm xuất phát ($K \cong 0$) từ một điểm cực phức, hoặc điểm đến từ một điểm không phức ($K \to +\infty$), có thể xác định từ quy tắc pha. Xem ví dụ dưới đây.

Chúng ta từng bước xây dựng quỹ đạo nghiệm theo quy tắc trên cho:

$$F(s) = \frac{(s+3)}{(s-1)(s+5)(s+4+j2)(s+4-j2)}$$

Trước tiên, liệt kê các điểm cực và điểm không:

$$p_1 = -4 + j2,\ p_2 = -4 - j2,\ p_3 = -5,\ p_4 = 1,\ z_1 = -3;$$

Do đó, $n = 4$ và $m = 1$.

1. Quỹ đạo nghiệm có bốn nhánh.
2. Ba nhánh hội tụ về đường tiệm cận có góc là 60^0, 180^0 và -60^0 và một nhánh hội tụ về $z_1 = -3$.
3. Tâm của đường tiệm cận là $\sigma = (-12+3)/3 = -3$.
4. Các khoảng $(-\infty; -5]$ và [-3;1] đều nằm trên quỹ đạo nghiệm.
5. Trục ảo đi qua các nghiệm khả thi của:

$$(\omega^4 - j12\omega^3 - 47\omega^2 + j40\omega - 100) + K(j\omega + 3) = 0 \tag{26.12}$$

với ω và K thực. Các phần thực và ảo của (26.12) là:

$$\omega^4 - 47\omega^2 - 100 + 3K = 0$$

$$j\omega(-12\omega^2 + 40 + K) = 0$$

suy ra hai cặp lời giải khả thi là $(K = 100/3; \omega = 0)$ và $(K = 215,83; \omega = \pm 4,62)$.

6. Điểm giao nhau của các nhánh là nghiệm của phương trình:

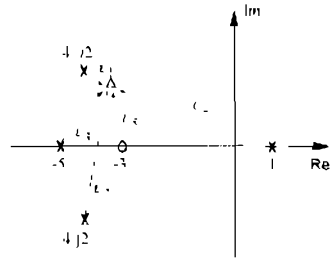
$$3s^4 + 36s^3 + 155s^2 + 282s + 220 = 0$$

Vì các nghiệm của phương trình là $-4,55 \pm j1,11$ và $-1,45 \pm j1,11$ nên không có lời giải nào trên trục thực, do đó không có điểm giao nhau.

7. Để xác định góc của điểm xuất phát từ điểm cực phức $p_1 = -4 + j2$, giả sử Δ biểu diễn một điểm trên quỹ đạo nghiệm gần điểm cực phức p_1 và đặt v_i với $i = 1,\ldots,5$ là các vectơ được vẽ từ p_i với $i = 1,\ldots,4$ và từ z_1 với $i = 5$, như biểu diễn trên hình 26.7.

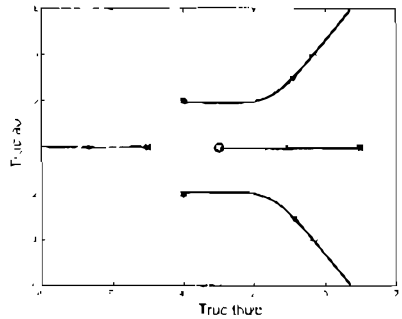
Gọi $\theta_1, \ldots \theta_5$ là góc của $v_1, \ldots, v_5$. Quy tắc pha chỉ ra rằng:

$$(\theta_1 + \theta_2 + \theta_3 + \theta_4) \quad \theta_5 = +180 \tag{26.13}$$



HÌNH 26.7 Góc của điểm xuất phát từ $-4 + j2$

khi Λ tiến tới p_1, θ_1 trở thành góc của điểm xuất phát và các θ_i còn lại được xấp xỉ bởi góc của các vectơ vẽ từ các điểm cực khác và điểm không tới điểm cực p_1. Do đó θ_1 có thể được giải từ (26.13), trong đó: $\theta_2 \approx 90^0$, $\theta_3 \approx \tan^{-1}(2)$, $\theta_4 \approx 180^0 - \tan^{-1}(2/5)$ và $\theta_5 \approx 90^0 + \tan^{-1}\left(\frac{1}{2}\right)$. Từ đó: $\theta_1 \approx -15^0$.



HÌNH 26.8 Quỹ đạo nghiệm cho $F(s) = \dfrac{(s+3)}{(s-1)(s+5)(s+4+j2)(s+4-j2)}$

Quỹ đạo nghiệm chính xác cho ví dụ này được biểu diễn trong hình 26.8. Từ kết quả của mục 5 nêu trên và dạng của quỹ đạo nghiệm, có thể kết luận rằng hệ thống phản hồi ổn định nếu:

$$33,33 < K < 215,83$$

Có nghĩa là bằng cách điều chỉnh đơn giản hệ số khuếch đại của bộ điều khiển, hệ thống có thể ổn định. Trong một số trường hợp, chúng ta cần sử dụng một bộ điều khiển động để thỏa mãn tất cả các yêu cầu thiết kế.

Các ví dụ thiết kế

Ví dụ 1

Xét hệ thống phản hồi cơ bản với đối tượng điều khiển:

$$P(s) = \frac{1}{0.72}\frac{1}{(s+1)(s+2)}$$

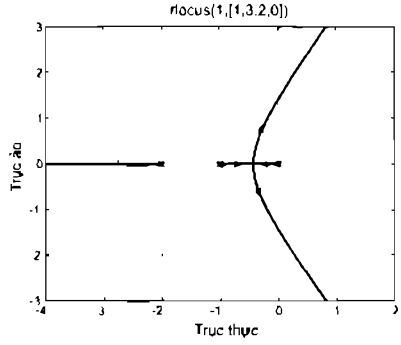
và thiết kế một bộ điều khiển sao cho

- Hệ thống phản hồi ổn định,

- PO ≤ 10%, $t_s \le 4s$ và sai số trạng thái ổn định bằng không khi $r(t)$ là hàm bậc thang đơn vị,
- Sai số trạng thái ổn định nhỏ khi $r(t)$ là hàm dốc đơn vị.

Rõ ràng là mục tiêu thiết kế thứ hai không thể đạt được bằng bộ điều khiển khuếch đại đơn giản. Để đạt được điều kiện này, bộ điều khiển phải có một điểm cực tại $s = 0$, ví dụ, nó cần phải có thành phần tích phân. Nếu ta thử điều khiển tích phân dưới dạng $C(s) = K_c / s$ với $K_c > 0$, thì quỹ đạo nghiệm có ba nhánh, khoảng [-1, 0] nằm trên quỹ đạo nghiệm, ba đường tiệm cận có các góc $\{60^0, 180^0, -60^0\}$ với trung tâm tại $\sigma_a = -1$; và chỉ có một điểm giao nhau tại $-1 + \frac{1}{\sqrt{3}}$, xem hình 26.9. Từ vị trí của các điểm giao nhau, tâm và góc của các đường tiệm cận, có thể suy ra là hai nhánh (một nhánh bắt đầu tại $p_1 = -1$ và nhánh còn lại bắt đầu tại $p_3 = 0$) luôn nằm bên phải của p_1. Mặt khác, điều kiện thời gian xác lập chỉ ra rằng phần thực của các điểm cực của hệ kín trội phải nhỏ hơn hoặc bằng -1. Do đó, một thành phần tích phân đơn giản không thể làm được điều này. Bây giờ ta thử bộ điều khiển PI có dạng:

$$C(s) = K_c\left(\frac{s - z_c}{s}\right), \qquad K_c > 0$$



HÌNH 26.9 Quỹ đạo nghiệm của ví dụ 1

Ta có thể chọn $z_c = -1$ để triệt tiêu điểm cực tại $p_1 = -1$ và hệ trở thành bậc hai. Quỹ đạo nghiệm cho $F(s) = 1/s(s+2)$ có hai nhánh và hai đường tiệm cận, với tâm $\sigma_a = -1$ và góc $\{90^0, -90^0\}$; điểm gãy cũng tại -1. Các nhánh -2 và 0, hướng về nhau, gặp nhau tại -1 và có xu hướng kéo dài vô tận dọc theo $\mathrm{Re}(s) = -1$. Các điểm cực của hệ kín là:

$$r_{1,2} = -1 \pm \sqrt{1 - K}, \quad \text{tại } K = K_c / 0.72$$

Sai số trạng thái ổn định, khi $r(t)$ là đơn vị độ dốc, $2/K$. K cần phải thật lớn để thỏa mãn điều kiện thiết kế thứ 3. Rõ ràng, $\mathrm{Re}(r_{1,2}) = -1$ với mọi $K \ge 1$, thỏa mãn yêu cầu về thời gian xác lập. Phần trăm độ quá điều chỉnh nhỏ hơn 10% nếu ζ của nghiệm $r_{1,2}$ lớn hơn 0.6. Một phép tính đơn giản cho thấy $\zeta = 1/\sqrt{K}$, vì vậy điều kiện thiết kế được thỏa mãn nếu K= 1/0.36, ví dụ $K_c = 2$. Do đó bộ điều khiển PI thỏa mãn vấn đề thiết kế sẽ là:

$$C(s) = 2\left(\frac{s+1}{s}\right)$$

Bộ điều khiển sẽ bỏ một điểm cực ổn định của đối tượng điều khiển (tại s = -1). Nếu có một sự không chắc chắn trong vị trí của điểm cực này, sự triệt tiêu hoàn toàn sẽ không xảy ra và hệ thống sẽ là bậc ba với điểm cực thứ ba tại $r_3 \cong -1$. Vì điểm không tại $z_0 = -1$ sẽ gần triệt tiêu hiệu ứng

của điểm cực này nên đáp ứng của hệ thống sẽ gần với đáp ứng của hệ bậc hai. Tuy nhiên, ta phải cẩn thận nếu các sự triệt tiêu điểm cực – không gần trục ảo, vì các nhiễu nhỏ ở vị trí điểm cực có thể dẫn đến sự thay đổi lớn trong đáp ứng hệ thống phản hồi, như minh họa trong ví dụ sau.

Ví dụ 2

Một cấu trúc linh hoạt với các điểm cực suy giảm nhẹ có hàm truyền đạt như sau:

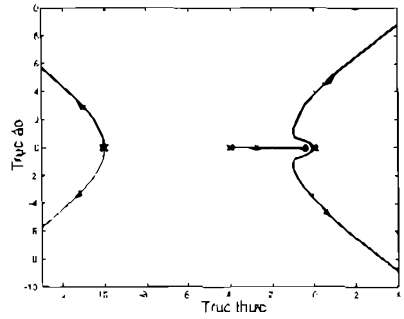
$$P(s) = \frac{\omega_1^2}{s^2(s^2 + 2\zeta\omega_1 s + \omega_1^2)}$$

Sử dụng quỹ đạo nghiệm, ta có thể thấy bộ điều khiển:

$$C(s) = K_c \frac{(s^2 + 2\zeta\omega_1 s + \omega_1^2)(s + 0.4)}{(s + r)^2(s + 4)}$$

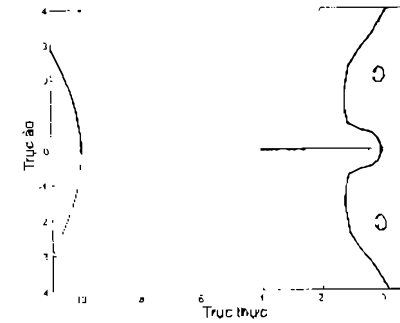
ổn định hệ thống phản hồi với r đủ lớn và một lựa chọn K_c phù hợp. Ví dụ, cho $\omega_1 = 2, \zeta = 0.1$ và $r = 10$. Thì quỹ đạo nghiệm của $F(s) = P(s)C(s)/K$ trong đó $K = K_c\omega_1^2$, được biểu diễn trong hình 26.10. Cho $K = 600$, các điểm cực của hệ kín sẽ là:

$$\{-10.78 \pm j2.57, -0.94 \pm j1.61, -0.2 \pm j1.99, -0.56\}$$



HÌNH 26.10 Quỹ đạo nghiệm cho ví dụ 2(a)

Vì các điểm cực $-0.2 \pm j1.99$ bị triệt tiêu bởi cặp điểm không tại cùng một điểm trong hàm truyền $T = G(1+G)^{-1}$ của hệ kín, nên các điểm cực trội tại -0,56 và $-0,94 \pm j1,61$ (có các phần thực âm tương đối lớn và tỉ lệ suy giảm khoảng 0.5).



HÌNH 26.11 Quỹ đạo nghiệm cho ví dụ 2(b)

Bây giờ, giả thiết rằng bộ điều khiển này được cố định và các điểm cực phức của đối tượng điều khiển được thay đổi một chút bằng cách đặt $\zeta = 0.09$ và $\omega_1 = 2.2$. Quỹ đạo nghiệm tương

ứng với hệ thống được biểu diễn trong hình 26.11. Vì các điểm cực phức suy giảm nhẹ không được triệt tiêu hoàn toàn, nên có hơn hai nhánh gần trục ảo. Hơn nữa, với cùng giá trị $K = 600$, các điểm cực của hệ kín là:

$$\{-10,78 \pm j2,57; -1.21 \pm j1,86; 0.05 \pm j1,93; -0,51\}$$

Trong trường hợp này, hệ thống phản hồi là không ổn định.

Ví dụ 3

Một trong những ví dụ quan trọng của hệ cơ điện tử là động cơ DC. Hàm truyền gần đúng của động cơ DC có dạng:

$$P_m(s) = \frac{K_m}{s(s+1/\tau_m)}, \quad \tau_m > 0$$

Nhận thấy rằng nếu τ_m lớn, thì $P_m(s) \approx P_h(s)$, trong đó $P_h(s) = K_h / s^2$ là hàm truyền của một dầm cứng. Trong ví dụ này, lớp tổng quát các đối tượng $P_m(s)$ sẽ được xét. Giả thiết rằng $p_m = -1/\tau_m$ và K_m được cho trước, bộ điều khiển bậc nhất:

$$C(s) = K_c \left(\frac{s - z_c}{s - p_c} \right) \tag{26.14}$$

sẽ được thiết kế. Mục đích là để đặt các điểm cực của hệ kín ra xa trục ảo. Vì bậc của $F(s) = P(s)_m C(s) / K_m K_c$ bằng ba, quỹ đạo nghiệm có ba nhánh. Giả thiết các điểm cực mong muốn được cho trước là p_1, p_2 và p_3. Sau đó, vấn đề đặt điểm cực rốt cuộc là tìm $\{K_c, z_c, p_c\}$ sao cho các phương trình đặc tính là:

$$\chi(s) = (s - p_1)(s - p_2)(s - p_3)$$

$$= s^3 - (p_1 + p_2 + p_3)s^2 + (p_1 p_2 + p_1 p_3 + p_2 p_3)s - p_1 p_2 p_3$$

Nhưng phương trình đặc tính thực, theo các thông số bộ điều khiển chưa biết, là:

$$\chi(s) = s(s - p_m)(s - p_c) + k(s - z_c)$$

$$= s^3 - (p_m + p_c)s^2 + (p_m p_c + K)s - Kz_c$$

trong đó K:=$K_m K_c$. Đặt các hệ số của $\chi(s)$ mong muốn bằng với các hệ số của $\chi(s)$ thực sẽ nhận được ba phương trình theo ba tham số chưa biết:

$$p_m + p_c = p_1 + p_2 + p_3$$

$$(p_m p_c + K) = p_1 p_2 + p_1 p_3 + p_2 p_3$$

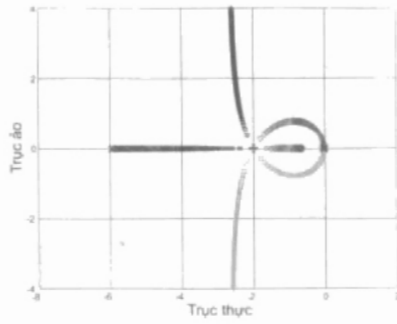
$$Kz_c = p_1 p_2 p_3$$

Từ phương trình thứ nhất, hai và ba lần lượt xác định được p_c, K và z_c.

Với các giá trị số khác của p_m, p_1, p_2 và p_3, dạng quỹ đạo nghiệm là khác nhau. Dưới đây là một số ví dụ, với các quỹ đạo nghiệm tương ứng trong hình 26.12-26.14:

(a) $p_m = -0,05; p_1 = p_2 = p_3 = -2 \Rightarrow$

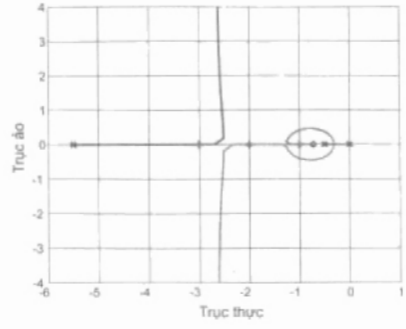
$$K = 11,70; \quad p_c = -5,95; \quad z_c = -0.68$$



HÌNH 26.12 Quỹ đạo nghiệm cho ví dụ 3(a)

(b) $p_m = -0,5; p_1 = -1; p_2 = -2; p_3 = -3 \Rightarrow$

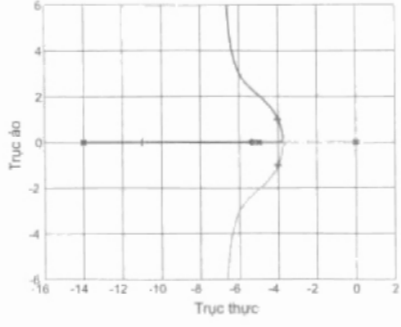
$$K = 8,25;\ p_c = -5,50;\ z_c = -0,73$$



HÌNH 26.13 Quỹ đạo nghiệm cho ví dụ 3 (b)

(c) $p_m = -0,5; p_1 = -1 1; p_2 = -4 + j1; p_3 = -4 - j1 \Rightarrow$

$$K = 35;\ p_c = -14;\ z_c = -5,343$$



HÌNH 26.14 Quỹ đạo nghiệm cho ví dụ 3(c)

Ví dụ 4

Xét hàm truyền của hệ hở: $P(s)C(s) = K_c \dfrac{(s^2 - 3s + 3)(s - z_c)}{s(s^2 + 3s + 3)(s - p_c)}$

trong đó K_c là hệ số khuếch đại của bộ điều khiển, z_c và p_c là điểm không và điểm cực tương ứng. Thấy rằng quỹ đạo nghiệm có bốn nhánh trừ trường hợp $z_c = p_c$. Cho các điểm cực trội mong muốn của hệ kín là $r_{1,2} = -0,4$. Sai số trạng thái tĩnh với tín hiệu vào tăng đơn vị:

$$e_{ss} = \frac{p_c}{K_c z_c}$$

Tùy theo chúng ta muốn tạo tỉ lệ $K_c z_c / p_c$ lớn như có thể.

Phương trình đặc tính là:

$$\chi(s) = s(s^2 + 3s + 3)(s - p_c) + K_c(s^2 - 3s + 3)(s - z_c)$$

và được dạng mong muốn là:

$$\chi(s) = (s + 0,4)^2 (s - r_3)(s - r_4)$$

cho các giá trị $r_{3,4}$ với $\mathrm{Re}(r_{3,4}) < 0$, có nghĩa là:

$$\chi(s)\big|_{s=-0,4} = 0, \qquad \frac{d}{ds}\chi(s)\big|_{s=-0,4} = 0 \tag{26.15}$$

Các điều kiện (26.15) đưa ra hai phương trình:

$$0,784(0,4 + p_c) - 4,36K_c(0,4 + z_c) = 0$$

$$4,36K_c - 0,784 - 1,08(0,4 + p_c) + 3,8K_c(0,4 + z_c) = 0$$

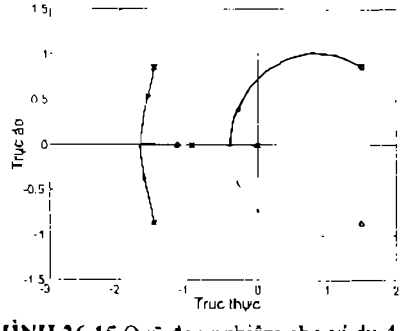
từ đó z_c và p_c được giải theo K_c, sau đó bằng các phép thay thế đơn giản, tỉ số được đặt lớn nhất, $K_c z_c / p_c$ có thể giảm tới:

$$\frac{K_c z_c}{p_c} = \frac{3,4776K_c - 0,784}{24,2469K_c - 3,4776}$$

Giá trị lớn nhất của K_c là 0.1297; từ đó $p_c = -0,9508$ và $z_c = -1,1637$. Cho bộ điều khiển này, các điểm cực của hệ phản hồi là:

$$\{-1,64 \pm j0,37; -1,64 - j0,37; -0,40; -0,40\}$$

Quỹ đạo nghiệm được biểu diễn trong hình 26.15.



HÌNH 26.15 Quỹ đạo nghiệm cho ví dụ 4

26.4 Quỹ đạo nghiệm bù

Ở phần trước, thông số K của quỹ đạo nghiệm được giả thiết là dương, pha và quy tắc độ lớn được thiết lập dựa trên giả thiết này. Có một số trường hợp mà hệ số khuếch đại của bộ điều khiển

cũng có thể âm. Vì vậy, hình ảnh hoàn chỉnh nhận được bằng cách vẽ quỹ đạo nghiệm thông thường (với $K>0$) và quỹ đạo nghiệm bù (với $K<0$). Các quy tắc của quỹ đạo nghiệm bù là:

$$l\times 360^{\circ} = \sum_{i=1}^{n} \angle(r-p_i) - \sum_{j=1}^{m} \angle(r-z_j), \quad l = 0, \pm 1, \pm 2, \ldots \tag{26.16}$$

$$|K| = \frac{\prod_{i=1}^{n} |r-p_i|}{\prod_{j=1}^{m} |r-z_j|} \tag{26.17}$$

Vì quy tắc pha (26.16) là phiên bản (26.9) dịch đi 180^0 nên quỹ đạo nghiệm bù có được bằng những thay đổi đơn giản của các quy tắc xây dựng quỹ đạo nghiệm. Cụ thể là số lượng các đường tiệm cận và tâm của chúng là như nhau, nhưng các góc α_l được cho bởi:

$$\alpha_l = \frac{2l}{(n-m)} \times 180^{\circ}, \quad l = 0, \ldots, (n-m-1)$$

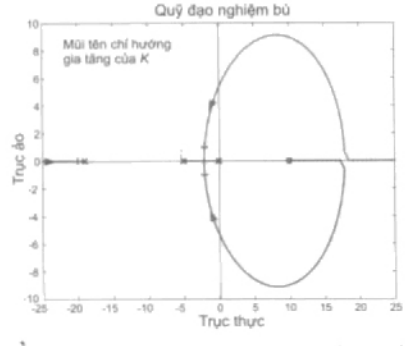
Cũng như vậy, một đoạn ở trên trục thực nằm trên quỹ đạo nghiệm bù nếu và chỉ nếu nó không ở trên quỹ đạo nghiệm thông thường.

Ví dụ 3

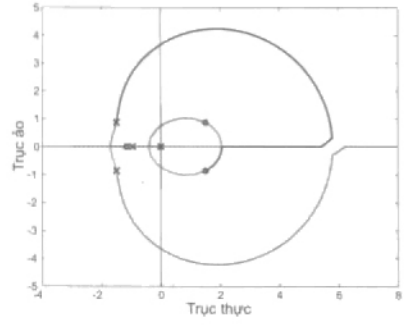
Trong ví dụ 3 đã cho ở trên, nếu số liệu được thay đổi thành $p_m = -5, p_1 = -20,$ và $p_{2,3} = -2 \pm j$ thì các thông số điều khiển trở thành:

$$K = -10, \qquad p_c = -19, \qquad z_c = 10$$

Chú ý rằng hệ số khuếch đại là âm. Các nghiệm của phương trình đặc tính khi K thay đổi giữa 0 và $-\infty$ tạo ra quỹ đạo nghiệm bù; xem hình 26.16.



HÌNH 26.16 Quỹ đạo nghiệm bù cho ví dụ 3



HÌNH 26.17 Quỹ đạo nghiệm thông thường và quỹ đạo nghiệm bù cho ví dụ 4

Ví dụ 4

Trong ví dụ này, nếu K tăng từ $-\infty$ đến $+\infty$, các điểm cực của hệ kín sẽ di chuyển dọc theo quỹ đạo nghiệm bù và sau đó là quỹ đạo nghiệm thông thường, như biểu diễn trong hình 26.17.

26.5 Quỹ đạo nghiệm cho hệ thống có trễ thời gian

Hệ điều khiển phản hồi chuẩn xét trong phần này được chỉ ra trong hình 26.18, trong đó bộ điều khiển C và đối tượng P có dạng:

$$C(s) = \frac{N_c(s)}{D_c(s)}$$

và:

$$P(s) = e^{-hs}P_0(s), \text{ với } P_0(s) = \frac{N_p(s)}{D_p(s)}$$

với (N_c, D_c) và (N_p, D_p) là các cặp đa thức cùng nguyên tố với các hệ số thực. Số hạng e^{-hs} là hàm truyền đạt của thành phần trễ (trong hình 26.18 đầu vào của đối tượng bị trễ h giây). Nói chung, đối tượng có trễ thời gian khi có:

- Trễ trong quá trình xử lý của cảm biến
- Trễ của phần mềm của bộ điều khiển
- Trễ truyền tải trong quá trình xử lý.

Trong trường hợp này, hàm truyền đạt hở là:

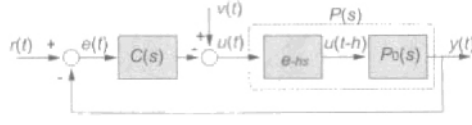
$$G(s) = G_h(s) = e^{-hs}G_0(s)$$

Trong đó $G_0(s) = P_0(s)C(s)$ tương ứng với trường hợp không trễ $h = 0$.

Chú ý rằng biên độ và pha của $G(j\omega)$ được xác định từ các đồng nhất thức:

$$|G(j\omega)| = |G_0(j\omega)| \tag{26.18}$$

$$\angle G(j\omega) = -h\omega + \angle G_0(j\omega) \tag{26.19}$$



HÌNH 26.18 Hệ thống phản hồi với trễ thời gian

Tính ổn định của các hệ thống trễ

Tính ổn định của hệ thống phản hồi đã được chỉ ra trong hình 26.18 tương đương với việc có tất cả các nghiệm của:

$$\chi(s) = D(s) + e^{-hs}N(s) \tag{26.20}$$

ở nửa mặt phẳng bên trái mở, $\mathbb{C}_-$, trong đó $D(s) = D_c(s)D_p(s)$ và $N(s) = N_c(s)N_p(s)$. Ta giả thiết rằng không có sự triệt tiêu các *điểm cực-điểm không* không ổn định khi tính $P_0(s)C(s)$ và rằng $\deg(D) > \deg(N)$. Thực ra $\chi(s)$ không phải là một đa thức vì nó là một hàm siêu việt của s. Các hàm có dạng (26.20) thuộc một lớp hàm đặc biệt gọi là *các giả đa thức*. Các điểm cực của hệ kín là nghiệm của (26.20).

Đã biết (xem 1, 10):

i. Nếu r_k là nghiệm của (20), thì $\overline{r_k}$ cũng thế (có nghĩa là các nghiệm xuất hiện theo dạng cặp liên hợp phức như thông thường).

ii. Có nhiều vô hạn điểm cực $r_k \in \mathbb{C}\, k = 1,2,\ldots$, thỏa mãn $\chi(r_k) = 0$.

iii. Và các r_k có thể được liệt kê theo cách mà $\mathrm{Re}(r_k + 1) \le \mathrm{Re}(r_k)$; hơn nữa $\mathrm{Re}(r_k) \to -\infty$ khi $k \to \infty$.

Ví dụ

Nếu $G_h(s) = e^{-hs}/s$, thì các điểm cực của hệ thống vòng hở r_k, với $k = 1,2,\ldots$, là các nghiệm của:

$$1 + \frac{e^{h\sigma_k} e^{jh\omega_k}}{\sigma_k + j\omega_k} e^{\pm j2k\pi} = 0 \tag{26.21}$$

trong đó $r_k = \sigma_k + j\omega_k$ với một số $\sigma_k, \omega_k \in \mathbb{R}$. Chú ý rằng $e^{\pm j2k\pi} = 1$ với mọi $k = 1,2,\ldots$. Phương trình 26.1 tương đương với tập các phương trình sau:

$$e^{-h\sigma_k} = |\sigma_k + j\omega_k| \tag{26.22}$$

$$\pm(2k-1)\pi = h\omega_k + \angle(\sigma_k + j\omega_k), \tag{26.23}$$

Điều này khá là thú vị vì với $h = 0$ chỉ có một nghiệm $r = -1$, nhưng thậm chí với $h > 0$ rất ít cũng có vô hạn nghiệm. Từ điều kiện về độ lớn (26.22), có thể biểu diễn:

$$\sigma_k \ge 0 \Rightarrow |\omega_k| \le 1 \tag{26.24}$$

Cũng như vậy, với $\sigma_k \ge 0$, pha $\angle(\sigma_k + j\omega_k)$ nằm giữa $-\pi/2$ và $+\pi/2$, vì vậy (26.23) đưa ra:

$$\sigma_k \ge 0 \Rightarrow h|\omega_k| \ge \frac{\pi}{2} \tag{26.25}$$

Bằng cách kết hợp (26.24) và (26.25), có thể chứng minh rằng hệ thống phản hồi không có nghiệm trong nửa mặt phẳng đóng bên phải khi $h < \pi/2$. Hơn nữa, hệ thống không ổn định nếu $h \ge \pi/2$. Cụ thể là, với $h = \pi/2$ có hai nghiệm trên trục ảo, tại $\pm j1$. Cũng có thể dễ dàng thấy rằng, với bất kì $h > 0$, khi $k \to \infty$, các nghiệm hội tụ về:

$$r_k \to \frac{1}{h}\left[-\ln\left(\frac{2k\pi}{h}\right) \pm j2k\pi\right]$$

Khi $h \to 0$, độ lớn của các nghiệm tiến tới ∞.

Như biểu diễn ở ví dụ trên, (iii) chứng tỏ rằng với số thực σ đã cho bất kỳ, chỉ có hữu hạn nghiệm r_k's trong miền của mặt phẳng phức:

$$\mathbb{C}_\sigma := \{s \in \mathbb{C}: \mathrm{Re}(s) \ge \sigma\}$$

Cụ thể với $\sigma = 0$, điều này có nghĩa là *giả đa thức* $\chi(s)$ có thể có hữu hạn nghiệm trong nửa mặt phẳng phải. Vì ảnh hưởng của các cực của hệ kín có các phần thực âm rất lớn bị bỏ qua (đến mức ứng xử vào – ra của hệ kín bị liên quan), nên chỉ có hữu hạn nghiệm "trội" r_k với $k = 1,\ldots,m$ phải được tính toán cho mọi mục đích thực tế.

Các nghiệm trội của một giả đa thức

Ta xét vấn đề sau: cho $N(s), D(s)$ và $h \geq 0$, tìm các nghiệm trội của giả đa thức:

$$\chi(s) = D(s) + e^{-hs} N(s)$$

Với mỗi giá trị cố định $h > 0$, có thể thấy rằng tồn tại σ_{max} sao cho $\chi(s)$ không có nghiệm nào trong vùng $\mathbb{C}_{\sigma_{max}}$, xem [11] với một thuật toán đơn giản để ước lượng σ_{max}, dựa trên tiêu chuẩn Nyquist. Cho $h > 0$ và một vùng mặt phẳng phức được định nghĩa bởi $\sigma_{min} \leq \mathrm{Re}(s) \leq \sigma_{max}$, vấn đề là tìm nghiệm của $\chi(s)$ trong miền này.

Rõ ràng, một điểm $r = \sigma + j\omega$ trong $\mathbb{C}$ là nghiệm của $\chi(s)$ nếu và chỉ nếu:

$$D(\sigma + j\omega) = -e^{-h\sigma} e^{-jh\omega} N(\sigma + j\omega)$$

Lấy bình phương độ lớn của cả hai vế của công thức trên, $\chi(r) = 0$ suy ra:

$$A_\sigma(x) := D(\sigma + x)D(\sigma - x) - e^{-2h\sigma} N(\sigma + x)N(\sigma - x) = 0$$

trong đó $x = j\omega$. Số hạng $D(\sigma + x)$ có mặt với hàm $D(s)$ được ước tính ở $\sigma + x$.

Các số hạng khác của $A_\sigma(x)$ cũng được tính tương tự. Với mỗi giá trị σ cố định, hàm $A_\sigma(x)$ là một đa thức của biến x. Nếu x là một điểm không của $A_\sigma(\cdot)$, thì $(-x)$ cũng là một điểm không.

Nếu $A_\sigma(x)$ có một nghiệm x_i có phần thực bằng 0, đặt $r_i = \sigma + x_i$. Tiếp theo, ước lượng độ lớn của $\chi(r_i)$; nếu bằng 0, thì n_i là nghiệm của $\chi(s)$. Ngược lại, nếu $A_\sigma(x)$ không có nghiệm trên trục ảo, thì $\chi(s)$ không thể có nghiệm với phần thực là giá trị cố định của σ mà từ đó $A_\sigma(\cdot)$ được xây dựng.

Thuật toán

Cho $N(s), D(s), h, \sigma_{min}$ và σ_{max}:

Bước 1. Lấy các giá trị $\sigma_1, \ldots, \sigma_M$ nằm trong khoảng σ_{min} và σ_{max} sao cho $\sigma_{min} = \sigma_1, \sigma_i < \sigma_i + 1$ và $\sigma_M = \sigma_{max}$. Với mỗi giá trị σ_i thực hiện các bước sau:

Bước 2. Xây dựng đa thức $A_i(x)$ theo:

$$A_i(x) := D(\sigma_i + x)D(\sigma_i - x) - e^{-2h\sigma_i} N(\sigma_i + x)N(\sigma_i - x)$$

Bước 3. Với mỗi nghiệm trục ảo x_i của A_i, thực hiện phép kiểm tra sau:

Kiểm tra nếu có $|\chi(\sigma_i + x_i)| = 0$, thì $r = \sigma_i + x_i$ là nghiệm của $\chi(s)$; nếu không thì loại bỏ x_i.

Bước 4. Nếu $i = M$ thì dừng, nếu không thì tăng i thêm 1 và nhảy tới bước hai.

Ví dụ

Ta sẽ tìm các nghiệm trội của:

$$1 + \frac{e^{-hs}}{s} = 0$$

với một tập các giá trị giới hạn của h. Nhớ rằng (26.26) có cặp nghiệm $\pm j1$ khi $h = \pi/2 = 1{,}57$. Hơn nữa, các nghiệm trội của (26.26) đều ở nửa mặt phẳng phải nếu $h > 1{,}57$ và chúng ở bên nửa

mặt phẳng trái nếu $h < 1,57$. Do đó, với, $h \in (1,2;2,0)$ ta cần các nghiệm trội ở gần trục ảo. Lấy $\sigma_{min} = -0,5$ và $\sigma_{max} = 0,5$, với $M = 400$ khoảng cách tuyến tính σ_i giữa chúng. Trong trường hợp này:

$$A_i(x) = \sigma_i^2 - e^{-2h\sigma_i} - x^2$$

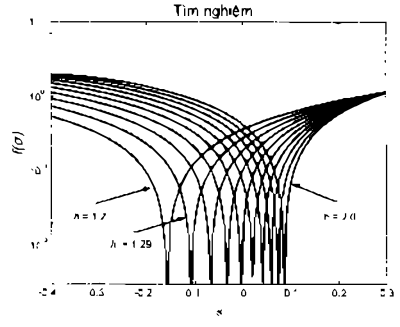
Ngay cả khi $e^{2h\sigma_i} \geq \sigma_i^2$, $A_i(x)$ có hai nghiệm:

$$x_i = \pm j\sqrt{e^{-2h\sigma_i} - \sigma_i^2}, \qquad \ell = 1,2$$

Với mỗi giá trị cố định σ_i thỏa mãn điều kiện này, đặt $r_i = \sigma_i + x_i$ (cần nhớ rằng x_i là hàm của σ_i, do đó r_i là một hàm của σ_i) và ước lượng:

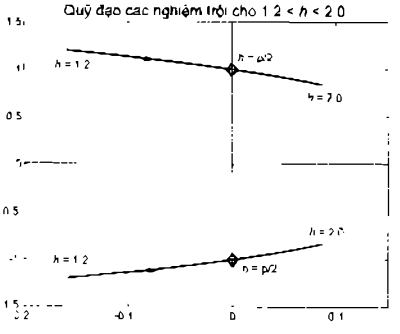
$$f(\sigma_i) := \left|1 + \frac{e^{-hr_i}}{r_i}\right|$$

Nếu $f(\sigma_i) = 0$, thì r_i là nghiệm của (26.26). Cho 10 giá trị khác nhau của $h \in (1,2;2,0)$, hàm $f(\sigma)$ được vẽ trên hình 26.19. Hình này chỉ ra các giá trị khả thi của σ_i với r_i (được định nghĩa từ σ_i) là một nghiệm của (26.26).



HÌNH 26.19. Tìm các nghiệm trội

Các nghiệm trội của (26.26), khi h thay đổi từ 1.2 đến 2.0, được chỉ ra trong hình 26.20. Với $h < 1.57$, tất cả các nghiệm đều nằm trong $\mathbb{C}_-$. Cho $h > 1,57$, các nghiệm trội đều ở trong $\mathbb{C}_+$ và với $h = 1,57$ thì các nghiệm trội ở $\pm j1$.



HÌNH 26.20 Các nghiệm trội khi h thay đổi từ 1.2 đến 2.0

Quỹ đạo nghiệm sử dụng xấp xỉ Padé

Trong phần này chúng ta giả thiết rằng $h > 0$ được cố định và ta cố gắng nhận được quỹ đạo nghiệm, tương ứng với hệ số khuếch đại K không chắc chắn/điều chỉnh được, tương ứng với các điểm cực trội. Vấn đề này có thể được giải quyết bằng việc tính theo phương pháp số các nghiệm trội của giả đa thức:

$$\chi(s) = D(s) + KN(s)e^{-hs} \tag{26.27}$$

với K thay đổi, bằng cách sử dụng phương pháp được giới thiệu trong phần trước. Trong phần này một phương pháp khác sử dụng xấp xỉ Padé của số hạng e^{-hs} của trễ thời gian sẽ được đưa ra. Cụ thể hơn là tìm đa thức $N_h(s)$ và $D(s)$ thỏa mãn:

$$e^{-hs} \approx \frac{N_h(s)}{D_h(s)} \tag{26.28}$$

do đó các nghiệm trội:

$$D(s)D_h(s) + KN(s)N_h(s) = 0 \tag{26.29}$$

gần đúng với các nghiệm trội của $\chi(s)$, (26.27). Chúng ta phải thực hiện xấp xỉ (26.28) như thế nào?

Bằng cách dùng các phép đo tính ổn định và bền vững từ tiêu chuẩn ổn định Nyquist, ta có thể chỉ ra rằng ta có thể xét giả hàm sau để xác định độ lớn của sai số xấp xỉ:

$$\Delta_h =: \sup_{\omega} \left| \frac{K_{\max} N(j\omega)}{D(j\omega)} \right| \left| e^{-jhw} - \frac{N_h(j\omega)}{N_h(j\omega)} \right|$$

trong đó K_{max} là giá trị tối đa của thông số K không chắc chắn/điều chỉnh được.

Xấp xỉ Padé bậc ℓ được định nghĩa như sau:

$$N_h(s) = \sum_{k=0}^{\ell} (-1)^k c_k h^k s^k$$

$$D_h(s) = \sum_{k=0}^{\ell} c_k h^k s^k$$

trong đó hệ số c_k's được tính từ:

$$c_k = \frac{(2\ell - k)!\ell!}{2\ell!k!(\ell - k)!}, \qquad k = 0, 1, ..., \ell$$

Xấp xỉ bậc một và bậc hai ở dạng:

$$\frac{N_h(s)}{D_h(s)} = \begin{cases} \dfrac{1 - hs/2}{1 + hs/2}, & \ell = 1 \\ \dfrac{1 - hs/2 + (hs)^2/12}{1 + hs/2 + (hs)^2/12}, & \ell = 2 \end{cases}$$

Với các số liệu đã cho $\{h, K_{max}, N(s), D(s)\}$, ta tìm bậc ℓ nhỏ nhất của xấp xỉ Padé, sao cho $\Delta_h \le \delta$ (hoặc $\Delta_h / K_{max} \le \delta'$) với sai số δ đã cho, hoặc sai số tương đối δ' đã cho như thế nào? Câu trả lời nằm ở kết quả sau đây [7]: với một bậc xấp xỉ ℓ cho trước, ta có:

$$\left| e^{-jh\omega} - \frac{N_h(j\omega)}{N_h(j\omega)} \right| \le \begin{cases} 2\left(\dfrac{eh\omega}{4\ell}\right)^{2\ell+1}, & \omega \le \dfrac{4\ell}{eh} \\ 2, & \omega \ge \dfrac{4\ell}{eh} \end{cases}$$

Từ đó ta có thể giải quyết vấn đề lựa chọn bậc xấp xỉ bằng cách dùng thủ tục sau đây:

1. Xác định tần số ω_s sao cho:

$$\left| \frac{K_{max} N(j\omega)}{D(j\omega)} \right| \le \frac{\delta}{2}, \text{ với mọi } \omega \ge \omega_s$$

và giá trị khởi tạo $\ell = 1$.

2. Với mỗi $\ell \ge 1$ xác định:

$$\omega_s = \max\left\{\omega_s, \frac{4\ell}{eh}\right\}$$

và vẽ hàm:

$$\Phi_\ell(\omega) := \begin{cases} 2\left|\dfrac{K_{max} N(j\omega)}{D(j\omega)}\right|\left(\dfrac{eh\omega}{4\ell}\right)^{2\ell+1}, & \omega \le \dfrac{4\ell}{eh} \\ 2\left|\dfrac{K_{max} N(j\omega)}{D(j\omega)}\right|, & \omega_s \ge \omega \ge \dfrac{4\ell}{eh} \end{cases}$$

3. Kiểm tra nếu:

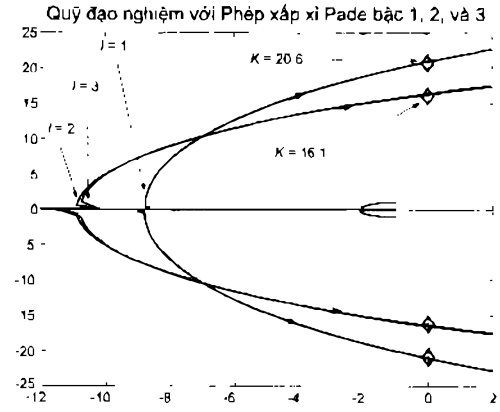
$$\max_{\omega \in [0, \omega_s]} \Phi_\ell(\omega) \le \delta \tag{26.30}$$

Nếu đúng thì dừng, giá trị này của ℓ thỏa mãn giới hạn sai số yêu cầu là $\Delta_h \le \delta$. Nếu không thì tăng ℓ lên 1 và nhảy đến bước 2. Chú ý rằng vế trái của bất đẳng thức (26.30) là giới hạn trên của Δ_h.

Vì ta giả thiết $Deg(D) > Deg(N)$, cuối cùng giải thuật sẽ qua bước 3 với một số hữu hạn $\ell \ge 1$. Tại mỗi lần lặp lại, ta phải vẽ hàm sai số $\Phi_\ell(\omega)$ và kiểm tra xem giá trị đỉnh của nó có nhỏ hơn δ hay không. Thông thường, khi δ giảm, ω_s tăng và kéo theo ℓ tăng lên. Mặt khác, với các giá trị ℓ rất lớn, độ lớn tương đối c_n / c_0 của các hệ số trở nên rất lớn, khi đó sẽ có những khó khăn trong phân tích và mô phỏng. Cũng như vậy, khi thời gian trễ h tăng lên, ℓ phải tăng lên để giữ mức sai số xấp xỉ δ cố định. Đây là khó khăn cơ bản liên quan đến các hệ có trễ thời gian.

Ví dụ

Đặt $N(s) = s + 1, D(s) = s^2 + 2s + 2$, $h = 0.1$ và $K_{max} = 20$. Khi đó, với $\delta = 0.5$, áp dụng thủ tục ở trên ta tính được $\ell = 2$ là độ xấp xỉ nhỏ nhất thoả mãn $\Delta_h / K_{max} < \delta$. Vì vậy, một xấp xỉ bậc hai của thời gian trễ là đủ để dự đoán các điểm cực trội với $K \in [0,20]$. Hình 26.21 biểu diễn các quỹ đạo nghiệm xấp xỉ nhận được từ xấp xỉ Padé của bậc ℓ=1,2,3. Có một sự khác biệt đáng kể giữa các quỹ đạo nghiệm của ℓ=1 và ℓ=2. Trong vùng $\mathrm{Re}(s) \ge -12$, các nghiệm trội dự đoán đều xấp xỉ giống nhau với ℓ=2,3, với $K \in [0,20]$. Do đó ta có nói rằng sử dụng xấp xỉ bậc cao sẽ không tạo ra bất kì sai khác đáng kể nào khi dự đoán ứng xử của các cực trội với dải hệ số K cho trước.



HÌNH 26.21 Nghiệm trội với $l = 1$

References

[1] Bellman, R. E., and Cooke, K. L., *Differential Difference Equations,* Academic Press, New York, 1963.

[2] Dorf, R. C., and Bishop, R. H., *Modern Control Systems,* 9th ed., Prentice-Hall, Upper Saddle River, NJ, 2001.

[3] Evans, W. R., "Graphical analysis of control systems," *Transac. Amer. Inst. Electrical Engineers,* vol. 67 (1948), pp. 547–551.

[4] Evans, W. R., "Control system synthesis by root locus method," *Transac. Amer. Inst. Electrical Engineers,* vol. 69 (1950), pp. 66–69.

[5] Franklin, G. F., Powell, J. D., and Emami-Naeini, A., *Feedback Control of Dynamic Systems,* 3rd ed., Addison Wesley, Reading, MA, 1994.

[6] Kuo, B. C., *Automatic Control Systems,* 7th ed., Prentice-Hall, Upper Saddle River, NJ, 1995.

[7] Lam, J., "Convergence of a class of Padé approximations for delay systems," *Int. J. Control,* vol. 52 (1990), pp. 989–1008.

[8] Ogata, K., *Modern Control Engineering,* 3rd ed., Prentice-Hall, Upper Saddle River, NJ, 1997.

[9] Özbay, H., *Introduction to Feedback Control Theory,* CRC Press LLC, Boca Raton, FL, 2000.

[10] Stepan, G., *Retarded Dynamical Systems: Stability and Characteristic Functions,* Longman Scientific & Technical, New York, 1989.

[11] Ulus, C., "Numerical computation of inner-outer factors for a class of retarded delay systems," *Int. J. Systems Sci.,* vol. 28 (1997), pp. 897–904.

27

Phương pháp đáp ứng tần số

27.1 Mở đầu 27-1
27.2 Đồ thị Bode 27-3
27.3 Đồ thị cực 27-7
27.4 Biên độ lôgarit đối với đồ thị pha 27-9
27.5 Xác định hàm truyền bằng thực nghiệm 27-9
27.6 Tiêu chuẩn ổn định Nyquist 27-12
27.7 Sự ổn định tương đối 27-16

Jyh-Jong Sheen
National Taiwan Ocean University

27.1 Mở đầu

Phương pháp đáp ứng tần số thường được dùng để phân tích và thiết kế các hệ điều khiển công nghiệp. Bằng phương pháp này chúng ta biết được đáp ứng trạng thái ổn định của một hệ với hệ số hằng tuyến tính có đầu vào là tín hiệu dạng hình sin. Chúng ta biết rằng đáp ứng của một hệ thống với đầu vào là tín hiệu hình sin cho tín hiệu đầu ra cũng dạng hình sin có cùng tần số với đầu vào. Tuy nhiên biên độ và pha của tín hiệu đầu ra khác tín hiệu đầu vào và sự khác biệt này là một hàm của tần số. Vì vậy chúng ta sẽ khảo sát mối quan hệ giữa hàm truyền và đáp ứng tần số của những hệ thống ổn định tuyến tính.

Xét một hệ thống có hệ số hằng ổn định tuyến tính như trên hình 27.1. Sử dụng công thức Euler, $e^{j\omega t} = \cos\omega t + j\sin\omega t$, với giả thiết tín hiệu đầu vào dạng hình sin được cho bởi công thức:

$$u(t) = U_0 e^{j\omega t} = U_0 \cos\omega t + jU_0 \sin\omega t \tag{27.1}$$

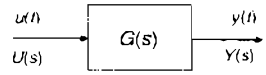
Biến đổi Laplace của $u(t)$ ta được:

$$U(s) = \frac{U_0}{s - j\omega} = U_0 \frac{s + j\omega}{s^2 + \omega^2} = \frac{U_0 s}{s^2 + \omega^2} + j\frac{U_0 \omega}{s^2 + \omega^2} \tag{27.2}$$

Biểu thức thứ nhất trong phương trình 27.2 là biến đổi Laplace của $U_0\cos\omega t$, trong khi biểu thức thứ hai, bỏ đi số ảo j, là biến đổi Laplace của $U_0\sin\omega t$.

Giả sử rằng hàm truyền $G(s)$ được viết dưới dạng sau:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{n(s)}{(s + p_1)(s + p_2)\cdots(s + p_n)} \tag{27.3}$$



HÌNH 27.1 Hệ thống hệ số hằng tuyến tính ổn định

Trong đó p_i, i=1,2...n là các điểm cực phân biệt. Biến đổi Laplace của đầu ra $Y(s)$ là:

$$Y(s) = G(s)U(s) = G(s)\frac{U_0}{s - j\omega} \tag{27.4}$$

Phương trình 27.4 tương đương với:

$$Y(s) = \frac{k_1}{s + p_1} + \cdots + \frac{k_n}{s + p_n} + \frac{\alpha}{s - j\omega} \tag{27.5}$$

Hệ số α được xác định bởi công thức:

$$\alpha = [(s - j\omega)Y(s)]\Big|_{s=j\omega} = [U_0 G(s)]\Big|_{s=j\omega} = U_0 G(j\omega)$$

Từ đó tính được biến đổi laplace ngược của $Y(s)$:

$$y(t) = k_1 e^{-p_1 t} + \cdots + k_n e^{-p_n t} + U_0 G(j\omega)e^{j\omega t}, t \geq 0 \tag{27.6}$$

Để hệ thống ổn định, tất cả các điểm cực phải có phần thực âm, do vậy tất cả các số hạng $k_i e^{-p_i t}$, i=1,2...n, tiến tới không khi t tiến tới vô cực. Trạng thái ổn định của đầu ra $y(t)$ là:

$$Y_{ss}(t) = \lim_{t \to \infty} y(t) = U_0 G(j\omega)e^{j\omega t} = U_0 |G(j\omega)| e^{j(\omega t + \phi)} \tag{27.7}$$

Hàm truyền dạng hình sin, $G(j\omega)$, được viết lại dưới dạng hàm mũ:

$$G(j\omega) = |G(j\omega)|e^{j\phi}$$

trong đó:

$$|G(j\omega)| = \sqrt{\{\mathrm{Re}[G(j\omega)]\}^2 + \{\mathrm{Im}[G(j\omega)]\}^2} \tag{27.8a}$$

và:

$$\phi = \angle G(j\omega) = \tan^{-1}\frac{\mathrm{Im}[G(j\omega)]}{\mathrm{Re}[G(j\omega)]} \tag{27.8b}$$

Phương trình (27.7) cho ta biết rằng đối với một hệ thống ổn định, đầu vào dạng hình sin thì đáp ứng trạng thái ổn định ở đầu ra cũng dạng hình sin với cùng tần số đầu vào. Biên độ đầu ra gấp $|G(j\omega)|$ lần đầu vào và góc pha khác nhau một lượng $\Phi = \angle G(j\omega)$.

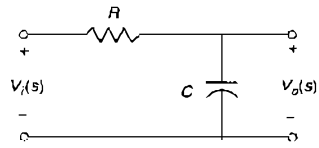
Ví dụ 1

Bộ lọc thông thấp bậc nhất trên hình 27.2, có hàm truyền đạt:

$$G(s) = \frac{V_0(s)}{V_i(s)} = \frac{1}{RCs + 1}$$

Hàm truyền dạng sin được cho bởi:

$$G(j\omega) = \frac{1}{j\omega(RC) + 1} = \frac{1}{j(\omega/\omega_1) + 1}$$

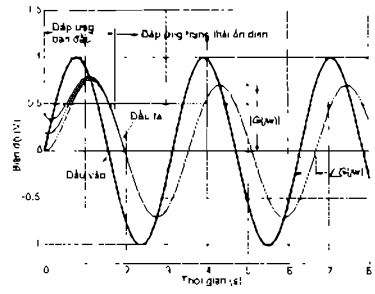


HÌNH 27.2 Bộ lọc thông thấp bậc nhất

trong đó $\omega_1=1/RC$, biên độ và góc pha của đáp ứng tần số là:

$$|G(j\omega)| = \frac{1}{\sqrt{1+(\omega/\omega_1)^2}} \qquad \phi(\omega) = -\tan^{-1}\frac{\omega}{\omega_1}$$

Hình 27.3 minh họa đáp ứng tần số của hệ thống khi RC=0.5 với đầu vào u=sin2t. Nó cũng cho thấy rằng đáp ứng trạng thái ổn định không liên quan đến điều kiện đầu vào và biên độ trạng thái ổn định của đầu ra là $1/\sqrt{2}$ và góc pha là -45°.



HÌNH 27.3 Đáp ứng tần số của G(s)=1/(0.5s+1) với u(t) = sin2t

27.2 Đồ thị Bode

Thông thường có 3 cách biểu diễn đáp ứng tần số của một hệ thống, đó là:

1. Biểu đồ Bode hoặc đồ thị lôgarit,
2. Đồ thị cực,
3. Đồ thị biên độ lôgarit đối với pha hoặc biểu đồ Nichols.

Trong phần này, chúng ta sẽ trình bày biểu đồ Bode của một hàm truyền dạng sin, sau đó là đồ thị cực và đồ thị biên độ lôgarit đối với pha.

Ưu điểm chính khi sử dụng đồ thị lôgarit là khả năng thể hiện đặc tính tần số cả bậc cao và thấp của hàm truyền trên cùng một biểu đồ và không bị ràng buộc khi thêm vào những số hạng khác nhau của một hàm truyền bậc cao. Các kiểu thừa số trong hàm truyền là:

1. Hệ số khuếch đại K,
2. Điểm cực (hoặc điểm không) tại gốc $(j\omega)^{\pm n}$,
3. Điểm cực (hoặc điểm không) trên trục thực $(j\omega\tau+1)^{\pm 1}$,
4. Điểm cực liên hợp phức (hoặc điểm không) $\left[(j\omega/\omega_n)^2+2\xi(j\omega/\omega_n)+1\right]^{\pm 1}$.

Đường cong biên độ lôgarit và góc pha cho bốn thừa số trên có thể dễ dàng vẽ được và thêm vào đồng thời trên đồ thị để đạt được những đường cong cho một hàm truyền. Quá trình vẽ đồ thị lôgarit có thể được đơn giản hóa bằng việc sử dụng phương pháp xấp xỉ tiệm cận cho những đường cong này và có được đường cong thực tế tại những tần số quan trọng cụ thể.

Hệ số khuếch đại K

Hệ số khuếch đại lôgarit là:

$$20\log|K| = \text{hằng số (decibel)},\ \angle K = \begin{cases} 0^\circ, & \text{if } K > 0 \\ -180^\circ, & \text{if } K < 0 \end{cases}$$

Đường cong pha và hệ số khuếch đại là những đường thẳng nằm ngang đơn giản trên biểu đồ Bode.

Điểm cực (hoặc điểm không) tại gốc $(j\omega)^{\pm n}$

Từ phương trình:

$$20\log|j\omega|^{\pm n} = \pm 20n\log\omega, \qquad \angle(j\omega)^{\pm n} = \pm n\times 90$$

độ dốc của đường cong biên độ là $\pm 20n$ dB/decade đối với thừa số $(j\omega)^{\pm n}$ và góc pha là hằng số bằng $\pm n\times 90^\circ$.

Điểm cực (hoặc điểm không) trên trục thực $(j\omega\tau+1)^{\pm 1}$

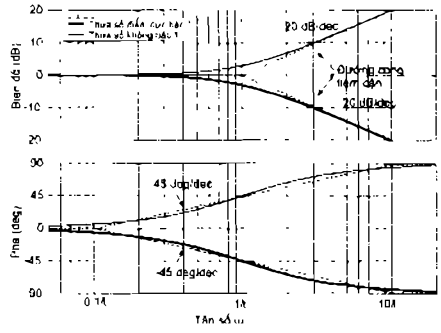
Cho một thừa số điểm cực $(j\omega\tau+1)^{-1}$:

$$\left|\frac{1}{j\omega\tau+1}\right| = \frac{1}{\sqrt{\omega^2\tau^2+1}}$$

Độ lớn của thừa số điểm cực là 1 khi $\omega << 1/\tau$ và bằng $1/(\omega\tau)$ khi $\omega >> 1/\tau$. Vì vậy có hai đường cong tiệm cận cho một thừa số điểm cực:

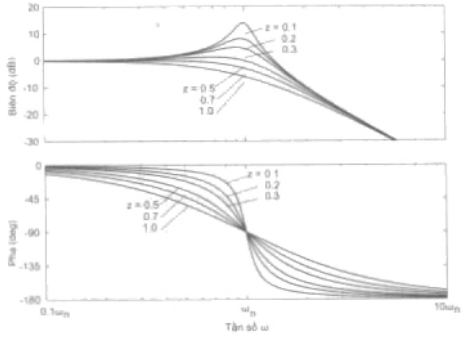
$$20\log\left|\frac{1}{j\omega\tau+1}\right| \approx \begin{cases} 0\,dB, & \omega \ll \dfrac{1}{\tau} \\ -20\log\omega\tau = -20\left(\log\omega - \log\dfrac{1}{\tau}\right), & \omega \gg \dfrac{1}{\tau} \end{cases}$$

Độ dốc của đường cong tiệm cận khi $\omega >> 1/\tau$ là -20 dB/decade cho một thừa số điểm cực. Hai tiệm cận giao nhau tại $\omega = 1/\tau$, gọi là tần số góc hoặc tần số gẫy. Hệ số lôgarit thực tế tại $\omega = 1/\tau$ là -3 dB. Góc pha là $\phi(\omega) = -\tan^{-1}\omega\tau$.



HÌNH 27.4 Biểu đồ Bode cho biểu thức bậc nhất $(j\omega\tau+1)^{\pm 1}$

Biểu đồ Bode của một thừa số không $(j\omega\tau+1)$ cũng có được bằng cách như trên. Tuy nhiên độ dốc đường cong tiệm cận biên độ khi $\omega >> 1/\tau$ là +20 dB/decade và góc pha là $\phi(\omega) = +\tan^{-1}\omega\tau$. Đồ thị Bode của thừa số bậc nhất được thể hiện trên hình 27.4. Sự xấp xỉ tuyến tính đường cong góc pha cũng được thể hiện.



HÌNH 27.5 Biểu đồ Bode cho biểu thức bậc hai $\left[(j\omega/\omega_n)^2 + 2\xi(j\omega/\omega_n)+1\right]^{\pm 1}$

Điểm cực liên hợp phức (hoặc điểm không) $\left[(j\omega/\omega_n)^2 + 2\xi(j\omega/\omega_n)+1\right]^{\pm 1}$

Biên độ và góc pha của điểm cực liên hợp phức $\left[(j\omega/\omega_n)^2 + 2\xi(j\omega/\omega_n)+1\right]^{\pm 1}$ là:

$$\left|\left(j\frac{\omega}{\omega_n}\right)^2 + 2\zeta\left(j\frac{\omega}{\omega_n}\right)+1\right|^{-1} = \left[\left(1-\frac{\omega^2}{\omega_n^2}\right)^2 + \left(2\zeta\frac{\omega}{\omega_n}\right)^2\right]^{-1/2}$$

$$\angle\left|\left(j\frac{\omega}{\omega_n}\right)^2 + 2\zeta\left(j\frac{\omega}{\omega_n}\right)+1\right|^{-1} = -\tan^{-1}\frac{2\zeta\omega/\omega_n}{1-\omega^2/\omega_n^2}$$

Biên độ của thừa số điểm cực liên hợp phức là 1 khi $\omega << \omega_n$ và bằng $(\omega/\omega_n)^{-2}$ khi $\omega >> \omega_n$. Vì vậy, hai đường cong tiệm cận cho thừa số điểm cực liên hợp phức là:

$$20\log\left|\left(j\frac{\omega}{\omega_n}\right)^2 + 2\zeta\left(j\frac{\omega}{\omega_n}\right)+1\right|^{-1} \approx \begin{cases} 0\,dB, & \omega \ll \omega_n \\ -40(\log\omega - \log\omega_n), & \omega \gg \omega_n \end{cases}$$

Độ dốc của đường cong tiệm cận khi $\omega >> \omega_n$ là -40 dB/decade cho thừa số điểm cực liên hợp phức. Những đường tiệm cận biên độ giao nhau tại $\omega = \omega_n$, tần số tự nhiên. Hệ số khuếch đại thực tế tại $\omega = \omega_n$ là $G(j\omega_n) = 1/2\xi$. Biểu đồ Bode của thừa số điểm cực liên hợp phức được thể hiện trên hình 27.5. Nhìn trên hình 27.5 ta thấy rằng sự khác nhau giữa đường cong biên độ thực tế và đường xấp xỉ tiệm cận là một hàm của hệ số suy giảm. Tần số cộng hưởng ω_r được định nghĩa là tần số mà ở đó có giá trị đỉnh của đáp ứng tần số M_r. Khi hệ số suy giảm tiến tới không, ω_r tiến tới ω_n. Tần số cộng hưởng có thể được xác định bằng cách lấy đạo hàm của biên độ đối với tần số và cho nó bằng không. Tần số cộng hưởng và giá trị đỉnh của biên độ được miêu tả bởi:

$$\omega_r = \omega_n \sqrt{1-2\zeta^2}, \zeta < 0.707 \tag{27.9a}$$

và:

$$M_r = \frac{1}{2\zeta\sqrt{1-2\zeta^2}}, \quad \zeta < 0.707 \tag{27.9b}$$

Ví dụ 2

Xét hàm truyền:

$$G(s) = \frac{10(s/5+1)}{s(s+1)[(s/10)^2 + (s/10)+1]}$$

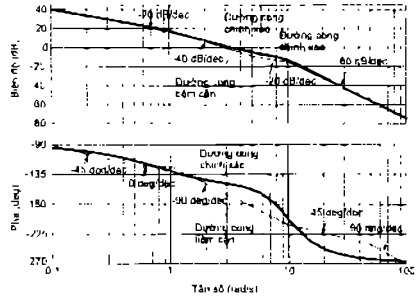
Trước tiên chúng ta liệt kê các thừa số cơ bản của hàm truyền $G(s)$ trong bảng 27.1, theo thứ tự tăng dần của tần số tự nhiên và tần số góc.

Đường cong biên độ tiệm cận hoàn thiện cho $G(j\omega)$ được tạo bằng cách thêm vào biên độ lôgarit tiệm cận của mỗi thừa số như thấy trên hình 27.6. Từ đó ta thấy rằng hệ số khuếch đại dc của mỗi thừa số là 1, những thừa số này không ảnh hưởng đến biên độ tiệm cận cho đến khi tần số tiến tới tần số tự nhiên hoặc tần số góc. Vì vậy, biên độ tiệm cận có thể đạt được rất nhanh bằng đồ thị mỗi tiệm cận theo sự tăng dần của tần số. Đường cong tiệm cận cắt đường thẳng 20 dB tại $\omega = 1$ với độ dốc là -20 dB/decade bởi vì điểm cực tại gốc toạ độ và hệ số khuếch đại K=10. Tại $\omega = 1$ độ dốc giảm -40 dB/decade bởi vì điểm cực tại $\omega = 1$. Sau đó tại $\omega = 5$ độ dốc tăng tới -20 dB/decade bởi vì điểm không tại $\omega = 5$. Cuối cùng tại $\omega = 10$ độ dốc là -60 dB/decade bởi vì điểm cực liên hợp phức tại $\omega_n = 10$.

Biên độ chính xác đạt được bằng cách tính toán biên độ thực tế tại những tần số quan trọng như tần số tự nhiên hoặc tần số góc của mỗi thừa số. Đường cong pha có thể đạt được bằng cách thêm vào pha của mỗi thừa số. Mặc dù xấp xỉ tuyến tính của đặc tính pha cho điểm cực hoặc điểm không đơn giản phù hợp cho việc phân tích lúc đầu nhưng lỗi giữa đường cong pha chính xác và xấp xỉ tuyến tính của điểm cực liên hợp phức có thể là lớn, như trên hình 27.6. Vì vậy, nếu yêu cầu có một đường cong góc pha chính xác thì một chương trình máy tính như Matlab hoặc C có thể được sử dụng để tìm ra đường cong pha thực tế.

BẢNG 27.1 Các thừa số cơ bản của $G(j\omega)$

Kiểu thừa số	Khuếch đại hằng	Điểm cực	Điểm cực	Điểm không	Điểm cực phức
Tần số góc	K=10	0	1	5	10
Bậc	0	-1	-1	+1	-2



HÌNH 27.6 Đồ thị Bode của hàm truyền trong ví dụ 2

27.3 Đồ thị cực

Đồ thị cực của một hàm truyền dạng sin $G(j\omega)$ là một đồ thị của cả biên độ và pha của đáp ứng tần số trong hệ tọa độ cực khi tần số ω thay đổi từ 0 tới vô cùng. Vì vậy, hàm truyền dạng sin có thể được miêu tả dưới dạng sau:

$$G(j\omega) = \mathrm{Re}\left[G(j\omega)\right] + j\,\mathrm{Im}\left[G(j\omega)\right] = \left|G(j\omega)\right| e^{j\phi}$$

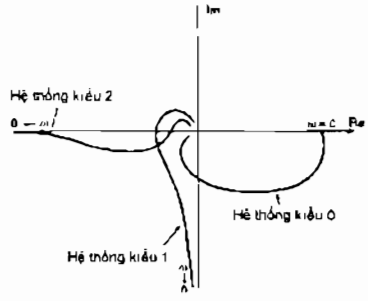
đồ thị cực của $G(j\omega)$ là đồ thị $\mathrm{Re}[\,G(j\omega)\,]$ trên trục hoành (trục nằm ngang) đối với $\mathrm{Im}[\,G(j\omega)\,]$ trên trục tung (trục thẳng đứng) trong mặt phẳng phức khi ω thay đổi từ 0 tới vô cực. Vì vậy, với mỗi giá trị của ω, một đồ thị cực của $G(j\omega)$ được định nghĩa bởi một vectơ có độ lớn là $\left|G(j\omega)\right|$ và góc pha là $\phi = \angle G(j\omega)$, như phương trình 27.8.

Chúng ta có thể khảo sát hình dáng tổng quát của đồ thị cực tùy thuộc vào kiểu của hệ thống và bậc tương đối của hàm truyền. Bậc tương đối của hàm truyền được định nghĩa là sự khác nhau giữa bậc của đa thức ở mẫu số và tử số. Xét một hàm truyền có dạng sau:

$$G(j\omega) = \frac{K(1+j\omega\tau_a)(1+j\omega\tau_b)\cdots}{(j\omega)^N(1+j\omega\tau_1)(1+j\omega\tau_2)\cdots} = \frac{b_0(j\omega)^m + b_1(j\omega)^{m-1}\cdots}{a_0(j\omega)^n + b_1(j\omega)^{n-1}\cdots}$$

BẢNG 27.2 $G(j\omega)$ đối với các kiểu hệ thống và bậc tương đối khi $\omega \to 0^+$ và ∞

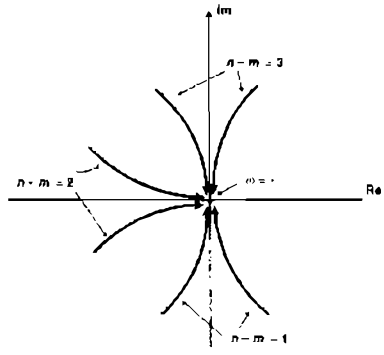
Kiểu hệ thống N	$\omega \to 0^+$	Bậc tương đối n - m	$\omega \to \infty$
0	$K\angle 0^0$	0	$b_0 / a_0 \angle 0^0$
1	$\infty\angle -90^0$	1	$0\angle -90^0$
2	$\infty\angle -180^0$	2	$0\angle -180^0$
3	$\infty\angle -270^0$	3	$0\angle -270^0$



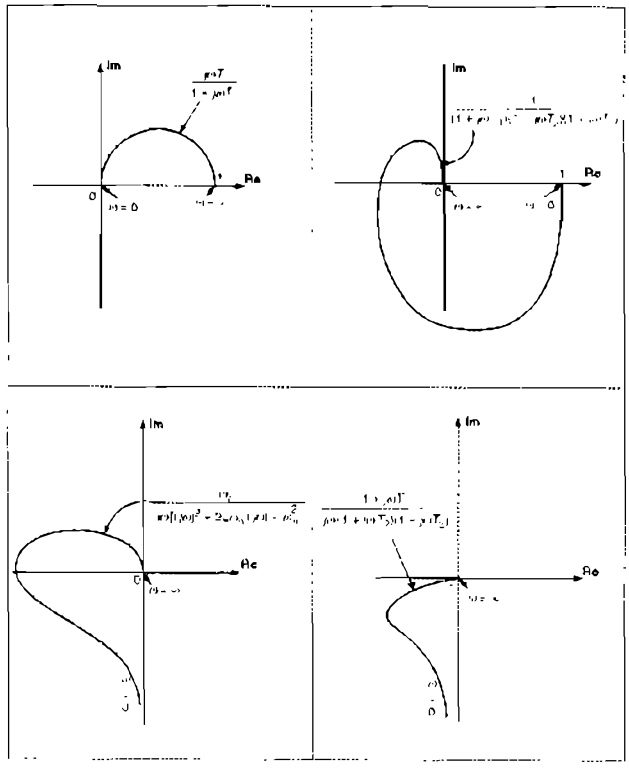
HÌNH 27.7 Đồ thị cực với các kiểu hệ thống khác nhau khi $\omega \to 0$

trong đó $K>0$ và bậc tương đối $n-m \geq 0$. Biên độ và góc pha của $G(j\omega)$ khi ω tiến tới 0 hoặc vô cùng được thể hiện trong bảng 27.2. Hình dáng tổng quát của đồ thị cực với các hệ thống khác nhau trong đoạn tần số thấp được thể hiện trên hình 27.7. Đoạn tần số cao của đồ thị cực với các

bậc tương đối khác nhau được thể hiện trên hình 27.8. Từ đó ta thấy rằng quĩ đạo nghiệm của $G(j\omega)$ song song với trục hoành hoặc trục tung, đồng thời biên độ tiến tới vô cùng khi ω tiến tới 0^+ cho những hệ thống lớn hơn không. Nếu bậc tương đối lớn hơn không, quĩ đạo nghiệm $G(j\omega)$ hội tụ về gốc theo chiều kim đồng hồ và tiếp xúc với trục tọa độ. Chú ý rằng đường cong tọa độ cực có thể rất phức tạp do đặc tính động học của tử thức và mẫu thức vượt quá dải tần số. Do vậy, đồ thị cực của $G(j\omega)$ trong dải tần số quan tâm phải được xác định chính xác.



HÌNH 27.8 Đồ thị cực với các bậc tương đối khác nhau khi $\omega \to \infty$

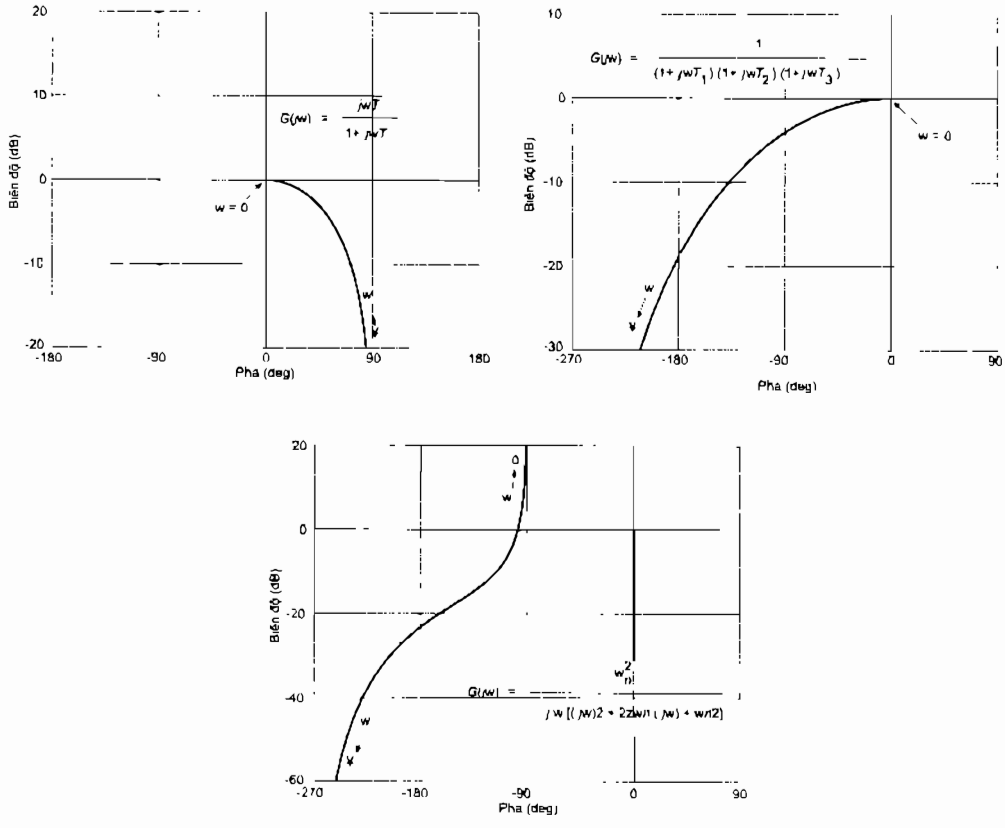


HÌNH 27.9 Đồ thị cực của một số hàm truyền đơn giản

Chúng ta thấy rằng đối với một hệ kín, đồ thị cực của một hàm truyền có ý nghĩa trong việc xác định sự ổn định của hệ thống. Đồ thị cực của một số hệ thống đơn giản được thể hiện trên hình 27.9.

27.4 Biên độ lôgarit đối với đồ thị pha

Một hướng khác để miêu tả đáp ứng tần số của hệ thống bằng một đồ thị đơn giản là vẽ đồ thị biên độ lôgarit của nó đối với góc pha trên dải tần số quan tâm. Đường cong có được là một hàm của tần số ω. Biên độ lôgarit đối với đồ thị pha được gọi là biểu đồ Nichols.



HÌNH 27.10 Biểu đồ Nichols của một số hàm truyền đơn giản

Ưu điểm của biểu đồ Nichols là sự ổn định tương đối của hệ kín có thể được xác định rất nhanh và quá trình bù vòng lặp kín được tìm ra dễ dàng. Biểu đồ Nichols của một hệ thống trên hình 27.9 được miêu tả trong hình 27.10 để tiện so sánh. Hình 27.11 hiển thị 3 đường cong đáp ứng tần số khác nhau của một hệ thống bậc hai:

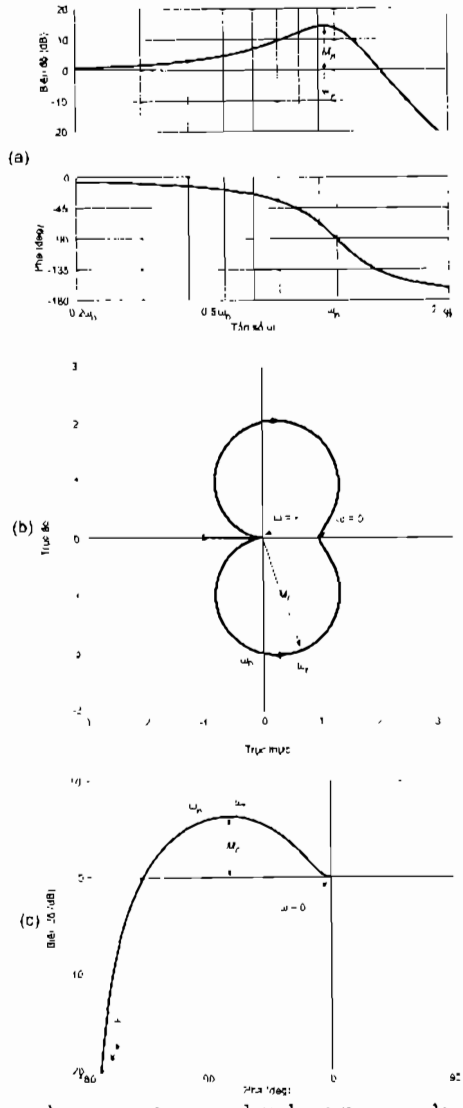
$$G(s) = \frac{\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2}$$

27.5 Xác định hàm truyền bằng thực nghiệm

Có thể thu được mô hình hàm truyền từ việc đo đáp ứng tần số của một hệ ổn định. Trước tiên, biểu đồ Bode của đáp ứng tần số được vẽ từ các giá trị đo. Sau đó hàm truyền vòng mở có thể suy ra từ đồ thị biên độ và pha dựa trên sự quan hệ giữa các thừa số điểm cực và điểm không cơ bản.

Một máy phân tích sóng là thiết bị đo biên độ và pha của đáp ứng trạng thái ổn định khi tần số của dạng sóng hình sin ở đầu vào thay đổi. Một máy phân tích hàm truyền có thể được sử dụng để đo hàm truyền hệ kín hoặc mở.

Chúng ta sẽ sử dụng một chương trình máy tính được kết hợp cùng với một card biến đổi số - tương tự và tương tự - số để phát ra tín hiệu đầu vào dạng sin và đo đáp ứng tần số của hệ thống. Xét một bộ lọc thông thấp Sallen-Key bậc hai như trên hình 27.12. Hàm truyền của bộ lọc được cho bởi:



HÌNH 27.11 Đáp ứng tần số của $G(s) = \omega_n^2 /(s^2 + 2\xi\omega_n s + \omega_n^2)$: (a) Biểu đồ Bode, (b) Đồ thị cực, (c) Biểu đồ Nichols

$$G(s) = \frac{V_0(s)}{V_i(s)} = \frac{K}{s^2/\omega_n^2 + 2\varsigma(s/\omega_n) + 1} \tag{27.10}$$

trong đó:

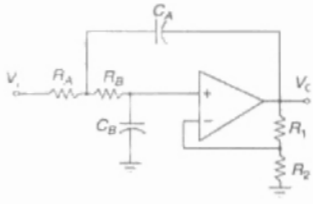
$$K = \frac{R_1 + R_2}{R_2} \qquad \omega_n = \frac{1}{\sqrt{R_A R_B C_A C_B}}$$

và:

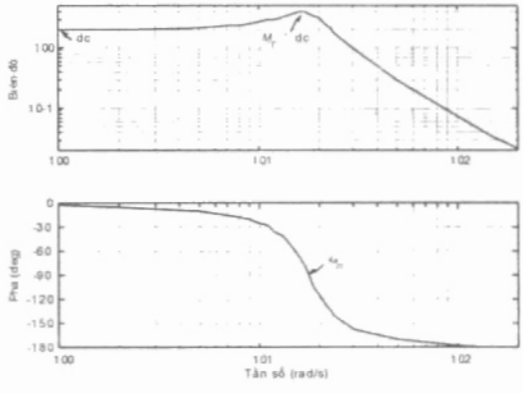
$$\varsigma = \frac{1}{2}\left[\sqrt{\frac{C_B}{C_A}}\frac{R_A + R_B}{\sqrt{R_A R_B}} + (1-K)\sqrt{\frac{R_A C_A}{R_B C_B}}\right]$$

Tín hiệu thời gian thực trên Windows trong Matlab được sử dụng cùng với card DA và AD Advantech PCL-818L. Thời gian lấy mẫu là 0.001s. Biên độ đo được và góc pha được thể hiện trên hình 27.13. Từ đồ thị Bode, chúng ta tìm ra hệ số khuếch đại dc bằng 1.995 và tần số tự nhiên $\omega_n = 17.90$ rad/s. Từ phương trình 27.9b và $M_r = 1.993$, chúng ta có $\xi = 0.26$.

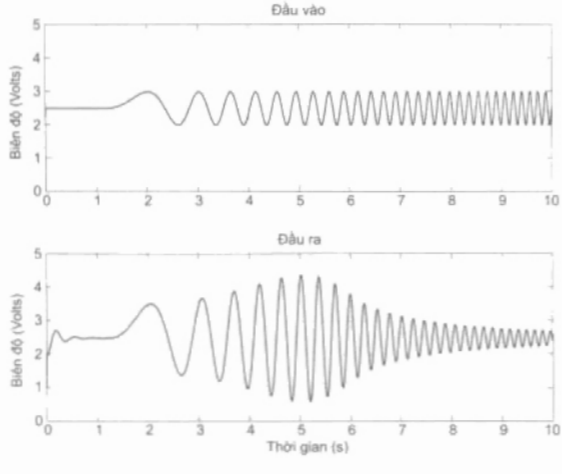
Một cách để đánh giá hàm truyền là sử dụng tín hiệu kích thích đủ mạnh trong dải tần số quan tâm và đo đầu ra tương ứng. Kỹ thuật nhận dạng hệ thống được áp dụng để tìm ra bậc và tham số của hàm truyền. Tín hiệu kích thích phù hợp là tín hiệu xung, tín hiệu quét hình sin, chuỗi ngẫu nhiên. Hình 27.14 là tín hiệu quét hình sin và đầu ra tương ứng. Công cụ nhận dạng hệ thống trên Matlab được sử dụng để đánh giá hàm truyền.



HÌNH 27.12 Bộ lọc thông thấp Sallen-Key



HÌNH 27.13 Đáp ứng tần số của bộ lọc thông thấp Sallen-Key từ thực nghiệm



HÌNH 27.14 Đáp ứng quét hình sin của bộ lọc Sallen-Key

BẢNG 27.3 Hàm truyền ước lượng cho bộ lọc thông thấp bậc hai

1. Mạch op-amp lí tưởng: $G(s) = \dfrac{1.997}{(s/18.09)^2 + 2x0.271(s/18.09) + 1}$ Trong đó giá trị đo được của điện trở và tụ điện được thay thế vào phương trình 27.10 là $R_1 = 98.4\Omega, R_2 = 98.7\Omega, R_A = 51.3\Omega, R_B = 98.5\Omega, C_A = 1.083\mu F, C_B = 0.564\mu F.$
2. Từ đồ thị Bode: $G(s) = \dfrac{1.995}{(s/17.90)^2 + 2x0.259(s/17.90) + 1}$
3. Nhận dạng hệ thống: $G(s) = \dfrac{1.997}{(s/17.78)^2 + 2x0.255(s/17.78) + 1}$

Hàm truyền của bộ khuyếch đại thuật toán lý tưởng được trình bày trong phương trình 27.10, đồ thị Bode và nhận dạng hệ thống được trình bày trong bảng 27.3. Có thể nhận thấy rằng sự khác nhau giữa ba hàm truyền là rất nhỏ. Tuy nhiên việc xác định hàm truyền từ đồ thị Bode là rất khó do các thừa số điểm cực và điểm không của tần số góc đóng có thể làm phức tạp đồ thị biên pha đối với các hệ thống bậc cao. Do vậy, kỹ thuật nhận dạng hàm truyền được sử dụng để xác định hàm truyền bậc cao.

27.6 Tiêu chuẩn ổn định Nyquist

Tiêu chuẩn ổn định Nyquist cung cấp một phương pháp đồ thị để xác định sự ổn định của hệ kín từ đường cong đáp ứng tần số hệ hở. Tiêu chuẩn này được dựa trên kết quả từ lý thuyết về biến số phức như là một nguyên tắc đối số Cauchy.

Giả sử $F(s)$ là một hàm hữu tỷ của s với hệ số thực dạng phân tích ở mọi điểm trên mặt phẳng phức (s-plane) ngoại trừ những điểm cực của nó. Đặt Γ_s là đường cong khép kín theo chiều kim đồng hồ trong mặt phẳng s và không đi qua điểm cực và điểm không của $F(s)$. Đường biên Γ_f được được định nghĩa là sự thay thế giá trị của s trên biên Γ_s cho s trong $F(s)$. Kết quả là được một đường biên liên tục khép kín trong mặt phẳng $F(s)$. Nguyên tắc của phương pháp này có thể bắt đầu như sau:

1. Biểu đồ đường biên Γ_f của hàm phức $F(s)$ được định nghĩa trên Γ_s trong mặt phẳng s sẽ chỉ được bao quanh điểm gốc của mặt phẳng $F(s)$ nếu chứa điểm cực và điểm không của hàm. Số vòng mà Γ_f bao quanh điểm gốc theo chiều kim đồng hồ là:

$$N = Z - P \tag{27.11}$$

2. Trong đó Z và P tương ứng là số điểm không và điểm cực của F(s) được bao quanh bởi đường biên Γ_s khép kín theo chiều kim đồng hồ trong mặt phẳng s.

Ví dụ 3

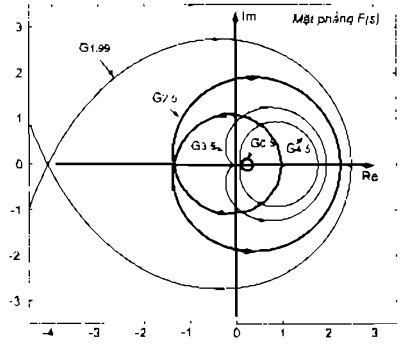
Để minh họa phương pháp, xét hàm truyền hữu tỷ sau:

$$F(s) = \frac{(s+3)(s+4)}{(s+1)(s+2)}$$

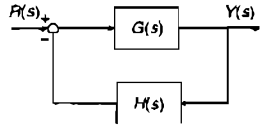
trong đó điểm không là s = -3, -4 và điểm cực là s = -1, -2. Các biểu đồ đường biên của $F(s)$ được thể hiện trên hình 27.15, ở đó Γ_f là biểu đồ đường biên của đường biên tròn theo chiều kim đồng

hổ với bán kính là r trong mặt phẳng s không bao quanh điểm cực và điểm không. Chúng ta rút ra các nhận xét sau đây từ đồ thị 27.15:

1. Đường biên Γ_{0s} không bao quanh gốc của mặt phẳng $F(s)$ khi đường biên trong mặt phẳng s không bao quanh điểm cực và điểm không.
2. Γ_{1s} bao quanh gốc một lần theo chiều ngược kim đồng hồ khi đường biên bao điểm cực tại s = -1 theo chiều kim đồng hồ trong mặt phẳng s và từ phương trình 27.11, $N = Z - P = 0 - 1 = -1$. Chú ý rằng Γ_{1s} là đường biên khép kín với hai vòng lặp và chỉ một bao quanh gốc như trên hình 27.15.
3. Γ_{2s} bao quanh gốc hai lần theo chiều ngược kim đồng hồ khi đường biên chứa hai điểm cực tại s = -1, -2 và $N = Z - P = 0 - 2 = -2$.
4. Khi bán kính của đường biên được tăng lên để chứa điểm cực tại s = -1, -2 và điểm không tại s = -3 thì $N = Z - P = 1\text{-}2 = -1$ và biểu đồ đường biên như Γ_{3s} bao quanh gốc một lần theo hướng ngược chiều kim đồng hồ.
5. Khi bán kính của đường biên được tăng xa hơn để bao quanh cả hai điểm cực và điểm không thì $N = 2 - 2 = 0$ và biểu đồ đường biên như Γ_{4s} không bao quanh điểm gốc.



HÌNH 27.15 Biểu đồ đường biên của F(s) trong ví dụ 3



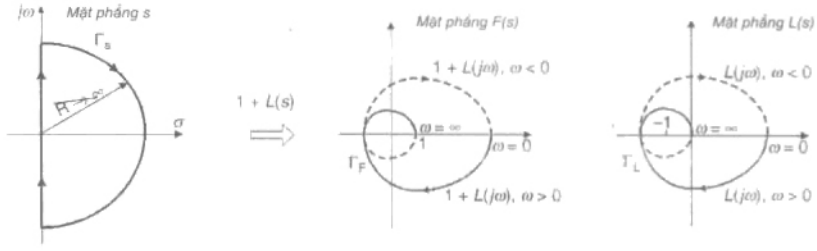
HÌNH 27.16 Hệ thống kín

Bây giờ chúng ta sẽ áp dụng nguyên tắc đối số Cauchy để phát triển tiêu chuẩn ổn định Nyquist. Giả sử rằng phương trình đặc tính của hệ thống vòng lặp hở trên hình 27.16 là:

$$F(s) = 1 + G(s)H(s) = 0$$

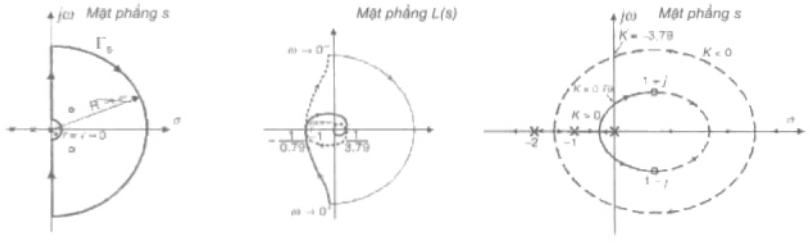
Đặt $L(s) = G(s)H(s)$, hàm truyền của vòng lặp. Sử dụng nguyên tắc đối số, giả sử rằng không có điểm cực hoặc điểm không của $F(s)$ nằm trên trục ảo trong mặt phẳng s. Chúng ta định nghĩa một quĩ đạo Nyquist, Γ_s, được tạo ra bởi trục ảo và nửa đường tròn bán kính vô cực. Đường biên này hoàn thành sẽ bao gồm toàn bộ nửa mặt phẳng phức bên phải như miêu tả trên hình 20.17 (a). Mặt phẳng đường biên tương ứng Γ_F như trên hình 27.17(b). Nguyên tắc đối số cho phép rằng N tương ứng với số vòng bao quanh chiều kim đồng hồ của điểm gốc trong mặt phẳng $1 + L(s)$ bởi Γ_1. P là số điểm cực của của $F(s)$ trong nửa mặt phẳng s bên phải và vì vậy là số điểm cực của hàm truyền $L(s)$ trong nửa mặt phẳng s bên phải. Z là số điểm không của phương trình đặc tính $F(s)$ của hệ thống kín trong nửa mặt phẳng s bên phải. Do vậy Z phải là 0 để cho hệ kín ổn định.

Trong thực tế, một sự sửa đổi được tạo ra để đơn giản hóa việc việc ứng dụng tiêu chuẩn Nyquist. Thay vì vẽ đồ thị Γ_F trong mặt phẳng $1+L(s)$, chúng ta chỉ vẽ đồ thị $L(s)$, dọc theo đường biên Γ_s. Mặt phẳng đường biên sinh ra Γ_L ở trong mặt phẳng $L(s)$ và có cùng hình dạng với Γ_F nhưng bị dịch chuyển một đơn vị về phía bên trái như trên hình 27.17(c). Vì vậy N là số vòng bao quanh điểm -1 trong mặt phẳng $L(s)$.



a) Đường Nyquist a) Mặt phẳng đường biên a) Biểu đồ Nyquist

HÌNH 27.17 Biểu đồ Nyquist



a) Đường Nyquist b) Biểu đồ Nyquist b) Quỹ đạo nghiệm

HÌNH 27.18 Biểu đồ Nyquist và quĩ đạo nghiệm trong ví dụ 4

Đến đây, tiêu chuẩn ổn định Nyquist được bắt đầu như sau:

Điều kiện cần và đủ cho sự ổn định của hệ kín được định nghĩa bởi hàm truyền $L(s)$ là:

$$Z = N + P \tag{27.12}$$

phải bằng 0, trong đó N là số vòng bao quanh điểm -1 trong mặt phẳng $L(s)$ và P là số điểm cực không ổn định của hàm truyền $L(s)$.

Ví dụ 4

Xét một hệ thống với hàm truyền là:

$$KL(s) = KG(s)H(s) = K\frac{s^2 - 2s + 2}{s(s+1)(s+2)} \tag{27.13}$$

Chúng ta hãy xác định dải giá trị của hệ số K để hệ kín ổn định. Do có một điểm cực tại $s = 0$, nên cần sửa đổi quĩ đạo Nyquist để đi vòng qua gốc tọa độ. Đường biên được vẽ trên hình 27.18(a), ở đó đường bao được lựa chọn là nửa đường tròn có bán kính tiến gần tới 0 trong một sự giới hạn. Chúng ta thực hiện các bước sau đây để phác thảo đồ thị Nyquist trên hình 27.18(b):

1. Xác định $L(j\omega)$ khi $\omega \to 0^+$: L(s) thuộc hệ thống kiểu 1 và vì vậy:

$$L(j\omega) \approx \frac{1}{j\omega} = \infty\angle -90$$

tương ứng với bảng 27.2.

2. Xác định $L(j\omega)$ khi $\omega \to \infty$: $L(s)$ có bậc tương đối là 1 và:

$$L(j\omega) \approx \frac{1}{j\omega} = 0\angle -90$$

tương ứng với bảng 27.2.

3. Từ đồ thị Bode, vẽ đồ thị cực của $L(j\omega)$ khi ω thay đổi từ 0^+ tới ∞. Mặc dù đường cong biên độ của $(s^2 - 2s + 2)$ giống với $(s^2 + 2s + 2)$, pha của $s^2 - 2s + 2$ thay đổi từ 0^o tới -180^o. Vì vậy một sự phác thảo biểu đồ Bode cho thấy đường cong biên độ thay đổi từ vô cực về 0 và góc pha thay đổi từ -90^o tới -450^o. Vì vậy có 2 điểm tại đó pha bằng -180^o và -360^o, ở đó sẽ có hai sự giao nhau của quĩ đạo $L(j\omega)$ với trục thực trong mặt phẳng $L(s)$.
4. Vẽ đồ thị cực của $L(j\omega)$ khi ω thay đổi từ 0^- tới $-\infty$ tương ứng với đường cong $L(j\omega)$ trong bước 3 và tương ứng với trục thực trong mặt phẳng $L(s)$.
5. Xác định biểu đồ đường biên của đường vòng nhỏ quanh gốc của mặt phẳng s để hoàn thiện đồ thị. Đường bao:

$$s = \lim_{\varepsilon \to 0} \varepsilon e^{j\theta}, \qquad -90 \le \theta \le -90$$

Biểu đồ đường biên được xác định bởi:

$$\lim_{\varepsilon \to 0} L(\varepsilon e^{j\theta}) = \lim_{\varepsilon \to 0} \frac{(\varepsilon e^{j\theta})^2 - 2\varepsilon e^{j\theta} + 2}{\varepsilon e^{j\theta}(\varepsilon e^{j\theta} + 1)(\varepsilon e^{j\theta} + 2)} = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon e^{j\theta}} = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \angle -\theta$$

Biểu đồ tạo được là một nửa đường tròn lớn có bán kính tiến gần tới vô cực. Nửa đường tròn này bắt đầu tại điểm $L(j0^-)$ và đi chuyển 180^o theo hướng ngược chiều kim đồng hồ để kết nối với điểm $L(j0^+)$ trong mặt phẳng $L(s)$.

6. Tính toán điểm giao nhau của quĩ đạo $L(j\omega)$ cùng với trục thực, những điểm này quan hệ với sự ổn định tương đối của hệ thống. Giả sử rằng quĩ đạo $L(j\omega)$ giao với trục thực ở những tần số tới hạn ω_{cr}. Thì:

$$L(j\omega_{cr}) = \begin{cases} 180^\circ + k360^\circ, & K > 0 \\ 0^\circ + k360^\circ, & K < 0 \end{cases}$$

Trong đó $k = 0, \pm 1, \pm 2, \pm 3, \ldots$ Điều kiện pha tại tần số tới hạn có quan hệ trực tiếp tới điều kiện góc của quĩ đạo nghiệm khi quĩ đạo nghiệm cắt trục ảo. Do đó, chúng ta có thể sử dụng tiêu chuẩn Routh-Hurwitz để xác định những điểm mà ở đó quĩ đạo $L(j\omega)$ cắt trục thực. Phương trình đặc tính của hệ thống (27.13) có thể được viết như sau:

$$s^3 + (K+3)s^2 + (2-2K)s + 2K = 0$$

Do vậy, mảng Routh là:

$$\begin{array}{c|cc} s^3 & 1 & 2-2K \\ s^2 & K+3 & 2K \\ s^1 & c & \\ s^0 & 2K & \end{array}$$

trong đó:

$$c = \frac{(K+3)(2-2K)-2K}{K+3}$$

Đặt $c = 0$ và giải tìm K, chúng ta được hệ số tới hạn:

$$K_{cr} = \frac{-3 \pm \sqrt{21}}{2} = 0.79, -3.79$$

Thay thế giá trị của K_{cr} vào trong phương trình phụ thuộc K_{cr} :

$$(K_{cr}+3)s^2 + 2K_{cr} = 0$$

chúng ta đạt được tần số tới hạn:

$$\omega_{cr} = \sqrt{\frac{2K_{cr}}{K_{cr}+3}} = \begin{cases} 0.65, & K_{cr} = 0.79 \\ 3.10, & K_{cr} = -3.79 \end{cases}$$

Tại tần số tới hạn, chúng ta có phương trình đặc tính:

$$1 + K_{cr} L(j\omega_{cr}) = 0$$

Do đó những điểm của quĩ đạo $L(j\omega)$ cắt trục thực là:

$$L(j\omega_{cr}) = -\frac{1}{K_{cr}} = -\frac{1}{0.79}, \frac{1}{3.79}$$

Đồ thị Nyquist hoàn thiện được thể hiện không theo tỷ lệ trên hình 27.18(b). Dải giá trị của hệ số K cho hệ thống ổn định có thể được xác định nhờ tiêu chuẩn Nyquist. Đối với những giá trị khác nhau của K, biểu đồ Nyquist cần vẽ lại để đếm số vòng bao quanh điểm -1. Chúng ta có thể tránh việc này bằng cách đếm số lần bao quanh điểm $-1/K$. Từ tiêu chuẩn Nyquist, $Z = N + P$, ở đó P=0. Từ hình 27.18(b), chúng ta thấy rằng có bốn trường hợp bao quanh điểm $-1/K$.

1. K>0 và $-1/K$<-1/0.79 $\Rightarrow$ 0<K<0.79 và N=0. Chúng ta có Z=0 và hệ thống ổn định.
2. K>0 và $-1/K$>-1/0.79 $\Rightarrow$ K>0.79 và N=2. Chúng ta có Z=2 và hệ thống có hai điểm cực không ổn định.
3. K<0 và $-1/K$<-1/3.79 $\Rightarrow$ K<-3.79 và N=3. Chúng ta có Z=3 và hệ thống có ba điểm cực không ổn định.
4. K<0 và $-1/K$>-1/3.79 $\Rightarrow$ -3.79<K<0 và N=1. Chúng ta có Z=1 và hệ thống có một điểm cực không ổn định.

Quĩ đạo nghiệm của hệ thống (27.13) cũng được thể hiện trên hình 27.18(c) để so sánh.

27.7 Sự ổn định tương đối

Trong quá trình thiết kế một hệ thống điều khiển, nó đòi hỏi hệ thống phải ổn định. Thêm nữa, có một sự liên quan quan trọng như đáp ứng tức thời được chấp nhận và khả năng xử lý được tính thay đổi của mô hình. Do mô hình được sử dụng trong thiết kế và phân tích của một hệ thống điều khiển thì không bao giờ chính xác, nó được giả thiết là một hệ ổn định, nhưng hệ thống vật lý làm cho nó không ổn định. Do vậy, nó đòi hỏi hệ thống không chỉ ổn định mà còn ở biên giới ổn định hoặc ổn định tương đối.

Giả sử rằng quĩ đạo của hàm truyền hình sin $L(j\omega)$ cắt điểm -1 tại tần số tới hạn ω_{cr}. Thì:

$$L(j\omega_{cr}) = G(j\omega_{cr})H(j\omega_{cr}) = -1$$

hoặc:

$$1 + L(j\omega_{cr}) = 1 + G(j\omega_{cr})H(j\omega_{cr}) = 0$$

Điều này chỉ ra rằng hệ thống kín có một cặp điểm cực phức tại $s = \pm\omega_{cr}$. Do vậy, hệ thống ở biên giới ổn định và dao động với với tần số ω_{cr}, được qui định là tất cả các điểm cực của hệ thống vòng lặp kín thì ở bên nửa trái mặt phẳng s. Một cách tổng quát quĩ đạo $L(j\omega)$ càng gần đến điểm -1+*j*0 trong đồ thị Nyquist thì đáp ứng của hệ thống dao động càng lớn. Vì lí do này mà sự tiếp xúc của quĩ đạo nghiệm $L(j\omega)$ tới điểm -1 có thể được sử dụng để đo biên giới ổn định. Hai cách đo truyền thống biên giới ổn định là biên khuếch đại và biên pha.

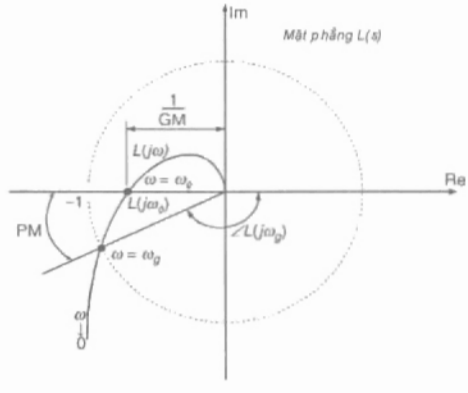
Biên khuếch đại và biên pha được định nghĩa thường cho hệ thống kín ổn định được đặc trưng bởi pha nhỏ nhất, hàm truyền vòng lặp là $L(s)$. Biên khuếch đại là thừa số mà khuếch đại hệ hở của hệ thống kín ổn định có thể được tăng lên trước khi đi đến không ổn định. Biên pha là lượng pha thêm vào tại tần số giao với khuếch đại được đòi hỏi để tạo cho hệ thống vòng lặp kín ổn định tới ổn định biên. Vì vậy chúng ta có các định nghĩa sau đây:

Biên khuếch đại (GM): Biên khuếch đại là sự nghịch đảo của độ lớn $|L(j\omega)|$ tại tần số giao với pha ω_ϕ, ở đó pha của $L(j\omega_\phi)$ tới -180°. Biên khuếch đại được cho bởi:

$$GM = \frac{1}{|L(j\omega_\phi)|} \qquad GM(dB) = -20\log|L(j\omega_\phi)|$$

Biên pha (PM): Biên pha được định nghĩa là góc giữa pha của hàm truyền tại tần số cắt với khuếch đại ở $|L(j\omega)| = 1$ và góc -180° hoặc $\zeta < 0.7$.

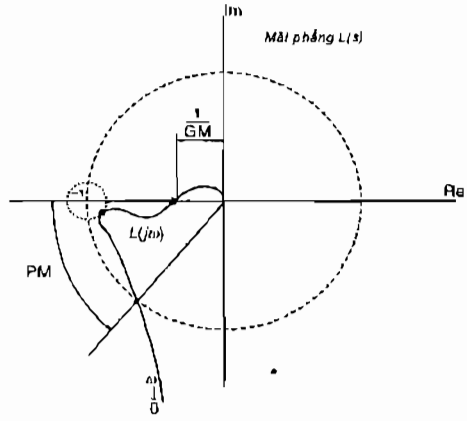
Biên pha và biên khuếch đại được thể hiện trên hình 27.19. Biên pha và biên khuếch đại là biên ổn định cho hệ thống một đầu vào một đầu ra (SISO). Chúng không thể áp dụng cho hệ thống nhiều đầu vào nhiều đầu ra (MIMO). Thêm vào nữa là sự thể hiện biên độ ổn định kém hơn trong mặt phẳng có sự kết hợp khác nhau giữa pha và khuếch đại, như trên hình 27.20. Điều này dẫn đến một thực tế là biên pha và biên khuếch đại là thước đo biên ổn định trong giới hạn chỉ có sự khác nhau giữa pha và khuếch đại, nhưng không có sự kết hợp cả hai. Kết quả là, một hệ thống có biên độ pha và biên độ khuếch đại tốt nhưng nó lại gần sát với trạng thái không ổn định, như chỉ ra trên hình 27.20. Để tạo ra sự thiếu biên độ khuếch đại và pha, biên ổn đinh thứ ba, hệ số khác biệt phản hồi được sử dụng trong lý thuyết điều khiển hiện đại. Chúng chỉ được áp dụng cho hệ thống SISO.



HÌNH 27.19 Biên pha và biên khuếch đại

Sự khác biệt phản hồi nhỏ nhất: là giá trị nhỏ nhất của $|L(j\omega)+1|$, khi $0 < \omega < \infty$. Nhìn trên hình 27.20 ta thấy rằng sự khác biệt phản hồi nhỏ nhất là khoảng cách nhỏ nhất từ đồ thị Nyquist tới điểm -1. Do vậy biên độ pha và biên độ khuếch đại là trường hợp đặc biệt của sự khác biệt

phản hồi nhỏ nhất. Biên độ khuếch đại có quan hệ trực tiếp tới trường hợp khi sự khác biệt phản hồi nhỏ nhất xảy ra tại tần số cắt với pha và biên độ pha thì tương ứng với trường hợp khi sự khác biệt phản hồi nhỏ nhất xảy ra tại tần số cắt với khuếch đại.



HÌNH 27.20 Thiếu biên pha và biên khuếch đại

Mặc dù sự khác biệt phản hồi nhỏ nhất là thước đo tốt cho biên độ ổn định hơn so với biên độ pha và biên độ khuếch đại, nhưng nó hiếm khi được sử dụng trong lý thuyết điều khiển cổ điển. Đây là lý do mà khi phân tích và thiết kế hệ thống điều khiển cổ điển thường sử dụng biểu đồ Bode và biểu đồ Nichols thay vì đồ thị Nyquist. Biên độ pha và khuếch đại được xác định dễ dàng hơn từ biểu đồ Bode hoặc biểu đồ Nichols so với đồ thị Nyquist. Mặc dù với thực tế là sự khác biệt phản hồi nhỏ nhất có thể được định giá một cách dễ dàng từ đồ thị Nyquist nhưng lại khó để xác định sự khác biệt phản hồi nhỏ nhất từ đồ thị Bode và biểu đồ Nichols.

Bây giờ chúng ta hãy tìm quan hệ giữa biên pha và hệ số suy giảm ζ của một hệ thống bậc hai. Xét một hệ thống bậc hai chuẩn:

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{27.14}$$

Chúng ta giả sử rằng hàm truyền $T(s)$ là của một hệ thống có phản hồi và được viết lại như sau:

$$T(s) = \frac{G(s)}{1 + G(s)}$$

ở đó hàm truyền của hệ hở $G(s)$ được cho bởi:

$$G(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)} \tag{27.15}$$

Biên độ pha xảy ra tại tần số cắt khuếch đại ω_c khi $|G(j\omega_c)| = 1$ hoặc:

$$\frac{\omega_n^2}{\omega_c(\omega_c^2 + 4\zeta^2\omega_n^2)^{1/2}} = 1$$

Phương trình này có thể được viết lại như sau:

$$(\omega_c^2)^2 + 4\zeta^2\omega_n^2(\omega_c^2) - \omega_n^4 = 0$$

Giải phương trình, với ω_c dương ta được:

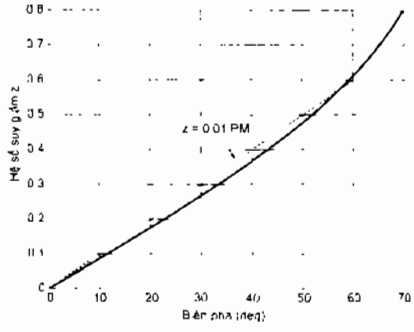
$$\frac{\omega_c^2}{\omega_n^2} = (4\zeta^4 + 1)^{1/2} - 2\zeta^2$$

Thay $s = j\omega_c$ vào phương trình 27.15, biên pha của hệ thống là:

$$\begin{aligned} PM &= 180^\circ + \angle G(j\omega_c) \\ &= 180^\circ - 90^\circ - \tan^{-1}\left(\frac{\omega_c}{2\zeta\omega_n}\right) \\ &= 90^\circ - \tan^{-1}\left(\frac{1}{2\zeta}[(4\zeta^4 + 1)^{1/2} - 2\zeta^2]^{1/2}\right) \\ &= \tan^{-1}\left(2\zeta\left[\frac{1}{(4\zeta^4 + 1)^{1/2} - 2\zeta^2}\right]\right) \end{aligned} \tag{27.16}$$

Phương trình (27.16) chỉ ra quan hệ giữa hệ số suy giảm của một hệ thống bậc hai chuẩn với biên pha của nó tương ứng với hệ thống hở (27.15) trong một hệ thống có phản hồi. Phương trình này cho biết mối quan hệ giữa đáp ứng tần số và đáp ứng thời gian. Đồ thị của ζ đối với PM được thể hiện trên hình 27.21. Đường cong ζ đối với PM có thể được xấp xỉ bằng đường thẳng nét đứt trên 27.21. Sự xấp xỉ tuyến tính có thể được diễn đạt bằng phương trình:

$$\zeta = 0.01PM$$



HÌNH 27.21 Hệ số suy giảm đối với biên pha của một hệ thống bậc hai

Sự xấp xỉ tuyến tính này có độ chính xác hợp lý khi $\zeta < 0.7$ và có ý nghĩa trong quan hệ giữa đáp ứng tần số và sự biểu diễn tức thời của một hệ thống bậc hai. Phương trình 27.17 cũng có thể được sử dụng cho hệ thống bậc cao nếu hệ thống được giả sử có một cặp điểm cực phức không suy giảm.

Tài liệu tham khảo

[1] Dorf, R.C., and Bishop, R.H., *Modern Control Systems*, 9th ed., Prentice-Hall, 2001.

[2] Ogata, K., *Modern Control Engineering*, 2nd ed., Prentice-Hall, 1990.

[3] Kuo, B.C., *Control Systems*, 7th ed., Prentice-Hall, 1995.

[4] Franklin, G.F., Powell, J.D., and Emami-Naeini, A., *Feedback Control of Dynamic Systems*, 3rd ed., Addison-Wesley, 1994.

[5] Phillips, C.L., and Harbor, R.D., *Feedback Control Systems*, 4th ed., Prentice-Hall, 2000.

[6] Wolovich, W.A., Automatic Control Systems: Basic Analysis and Design, Harcourt Brace College Publishing, 1994.

28

Bộ lọc Kalman trong vai trò bộ quan sát trạng thái các hệ động lực

28.1 Bộ lọc Kalman tuyến tính thời gian rời rạc 28-1
28.2 Các công thức bộ lọc Kalman khác 28-5
28.3 Tổng kết và tóm tắt các công thức 28-7
28.4 Xem xét sự thực hiện 28-8

Timothy P. Crain II
NASA Johnson Space Center

28.1 Bộ lọc Kalman tuyến tính thời gian rời rạc

Được chắt lọc từ các thành phần cơ bản, bộ lọc Kalman [1] là một thuật toán sử dụng một mô hình hệ động lực để dự báo các biến trạng thái và một mô hình đo để hiệu chỉnh việc dự báo này. Tuy nhiên, bộ lọc Kalman có nhiều tính năng hơn bộ quan sát trạng thái thông thường. Bằng cách tạo ra những giả thiết ngẫu nhiên, bộ lọc Kalman là thước đo độ tin cậy thống kê của sự ước lượng các biến trạng thái riêng biệt dưới dạng ma trận hiệp phương sai. Các thuộc tính cơ bản của bộ lọc Kalman được bắt nguồn từ các yêu cầu của ước lượng trạng thái như sau:

- Sự kết hợp tuyến tính của ước lượng trạng thái trước và thông tin đo hiện tại
- Không thay đổi so với trạng thái thực
- Tối ưu dưới dạng có phương sai cực tiểu đối với trạng thái thực

Bộ lọc Kalman xử lý chuỗi các giá trị đo theo thời gian để cập nhật ước lượng trạng thái hệ thống và sử dụng một mô hình động lực để khuếch đại ước lượng trạng thái giữa các giá trị đo. Giá trị đo quan sát được là một hàm trạng thái hệ thống và được biểu diễn như sau:

$$\mathbf{Y}(t) = h(\mathbf{X}(t), \beta, t) + \mathbf{v}(t) \tag{28.1}$$

trong đó $\mathbf{Y}(t)$ là giá trị quan sát được có m chiều, h là mô hình đo phi tuyến, $\mathbf{X}(t)$ là trạng thái hệ thống n chiều, β là một vectơ của các tham số mô hình, và $\mathbf{v}(t)$ là quá trình ngẫu nhiên của nhiễu đầu đo.

Hệ động lực chuẩn được miêu tả bởi phương trình vi phân thường, bậc nhất toàn phần là:

$$\dot{\mathbf{X}}(t) = f(\mathbf{X}(t), \alpha, t) + \mathbf{w}(t) \tag{28.2}$$

trong đó f là hàm động lực phi tuyến kết hợp tất cả các hiệu ứng tiền định có nghĩa của môi trường, α là một vectơ các tham số được sử dụng trong mô hình, và $w(t)$ là một quá trình ngẫu

nhiên gây ra nhiễu do sai mô hình f hoặc từ sai số lượng tử, phụ thuộc vào độ chính xác của mô hình tiền định đang dùng.

Với các mô hình tổng quát có sẵn, bộ lọc Kalman tuyến tính (LKF) có thể được biểu diễn dưới dạng công thức với biến số rời rạc theo thời gian. Các hàm động lực và hàm đo được tuyến tính hóa với trạng thái tham chiếu đã biết, $\tilde{X}(t)$, liên quan tới trạng thái môi trường chuẩn, $X(t)$ qua công thức:

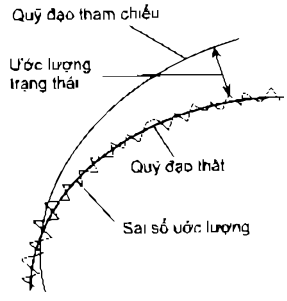
$$\tilde{\mathbf{X}}(t) + \mathbf{x}(t) = \mathbf{X}(t) \tag{28.3}$$

Ước lượng trạng thái LKF liên quan tới sai phân chuẩn:

$$\hat{\mathbf{x}}_k^{(\pm)} = \mathbf{x}_k + \delta\mathbf{x}_k^{(\pm)} \tag{28.4}$$

trong đó "^" thể hiện ước lượng trạng thái (hoặc trạng thái lọc), $\delta\mathbf{x}_k^{(=)}$ là sai số ước lượng, và "$\pm$" thể hiện ước lượng và sai lệch được định lượng ngay trước (-) hoặc ngay sau (+) giá trị đo cập nhật tại thời gian rời rạc t_k.

Chú ý rằng trạng thái bộ lọc LKF là ước lượng sai khác giữa trạng thái môi trường và trạng thái tham chiếu. Do vậy chế độ hoạt động LKF sẽ bám theo trạng thái tham chiếu và trạng thái lọc với giá trị đo cập nhật. Chỉ có trạng thái lọc là tại thời điểm cập nhật giá trị đo. Hình 28.1 miêu tả mối quan hệ giữa trạng thái thực, trạng thái tham chiếu, và trạng thái lọc trong ước lượng LKF với một quỹ đạo hai chiều.



HÌNH 28.1 Minh họa hoạt động của bộ lọc LKF cho quỹ đạo 2 chiều

Tuyến tính hóa các mô hình hệ động lực và mô hình đo lường

Các hàm động lực và đo lường có thể được tuyến tính hóa về trạng thái tham chiếu đã biết, $\tilde{X}(t)$, xuất phát từ:

$$f(\mathbf{X},\boldsymbol{\alpha},t) \simeq f(\tilde{\mathbf{X}}(t),\boldsymbol{\alpha},t) + F(\tilde{\mathbf{X}}(t),\boldsymbol{\alpha},t)x(t) + \mathbf{w}(t) \tag{28.5}$$

$$h(\mathbf{X},\boldsymbol{\alpha},t) = h(\tilde{\mathbf{X}}(t),\boldsymbol{\beta},t) + \mathbf{H}(\tilde{X}(t),\boldsymbol{\beta},t)\mathbf{x}(t) + \mathbf{v}(t) \tag{28.6}$$

trong đó **F** là ma trận đạo hàm riêng động lực và **H** là ma trận đạo hàm riêng đo lường được định nghĩa bởi:

$$\mathbf{F}(\tilde{\mathbf{X}}(t),\boldsymbol{\alpha},t) = \left.\frac{\partial f}{\partial \mathbf{X}}\right|_{X=\tilde{X}} \tag{28.7}$$

$$\mathbf{H}(\tilde{\mathbf{X}}(t),\boldsymbol{\beta},t) = \left.\frac{\partial h}{\partial \mathbf{X}}\right|_{X=\tilde{X}} \tag{28.8}$$

và $x(t)$ là trạng thái chuẩn được ước lượng bằng sự sai khác giữa trạng thái môi trường và trạng thái tham chiếu:

$$x(t) = \mathbf{X}(t) - \tilde{\mathbf{X}}(t) \tag{28.9}$$

Sau khi tuyến tính hóa các mô hình động lực và mô hình đo, giả sử không quan tâm đến các thành phần bậc cao của các quá trình ngẫu nhiên $\mathbf{w}(t)$ và $\mathbf{v}(t)$. Tuyến tính hóa là một xấp xỉ có thể chấp nhận được khi $\mathbf{x}(t)$ đủ nhỏ.

Các trạng thái tham chiếu và trạng thái lọc thay đổi theo quan hệ tuyến tính rời rạc:

$$\tilde{\mathbf{X}}_{k+1} = \boldsymbol{\Phi}(t_{k+1}, t_k)\tilde{\mathbf{X}}_k \tag{28.10}$$

$$\hat{\mathbf{x}}_{k+1}^{(-)} = \boldsymbol{\Phi}(t_{k+1}, t_k)\hat{\mathbf{x}}_k^{(+)} \tag{28.11}$$

trong đó $\Phi(t_{k+1}, t_k)$ là ma trận chuyển đổi trạng thái từ thời gian t_k tới t_{k+1} và có các tính chất sau:

$$\begin{aligned} &\boldsymbol{\Phi}(t_k, t_k) = \mathbf{I} \\ &\dot{\boldsymbol{\Phi}}(t_{k+1}, t_k) = \mathbf{F}(\tilde{\mathbf{X}}(t), \boldsymbol{\alpha}, t)\boldsymbol{\Phi}(t_{k+1}, t_k) \\ &\boldsymbol{\Phi}(t_{k+2}, t_k) = \boldsymbol{\Phi}(t_{k+2}, t_{k+1})\boldsymbol{\Phi}(t_{k+1}, t_k) \end{aligned} \tag{28.12}$$

Lưu ý rằng động lực hệ thống bây giờ được kết hợp với sự thay đổi của các trạng thái tham chiếu và trạng thái lọc bởi phép tích phân đạo hàm riêng động lực trong (28.13).

Về mặt toán học, trạng thái sai phân thực được truyền tương tự như việc cộng thêm một giá trị nhiễu ngẫu nhiên:

$$\mathbf{x}_{k+1} = \boldsymbol{\Phi}(t_{k+1}, t_k)\mathbf{x}_k + \mathbf{w}_k \tag{28.13}$$

Nói chung, mô hình động lực tham chiếu không cần phải chính xác giống như động học thực hoặc tham số mô hình $\boldsymbol{\alpha}$ cũng không cần tương đương với vectơ mô hình thực. Lưu ý này cho phép đơn giản hóa việc lấy đạo hàm của công thức bộ lọc Kalman. Một số cách tiếp cận mới đã được phát triển dựa theo các tham số mô hình tham chiếu để tăng độ tin cậy với mô hình thực chưa biết [2-6] và được sử dụng để nâng cao hoạt động của bộ lọc.

LKF cũng cần giá trị đo được tuyến tính hóa, $\mathbf{y}_k = \mathbf{Y}_k - \tilde{\mathbf{Y}}_k$, được mô hình bởi:

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{28.14}$$

Với sự phát triển của bộ lọc Kalman trình bày trên đây, phân bố ngẫu nhiên $\mathbf{v}_k$ và $\mathbf{w}_k$ được giả thiết là sự thể hiện rời rạc *quá trình Gaussian trung bình không* liên tục trong phương trình (28.1) và (28.2) và được định nghĩa như sau:

$$E\left[\mathbf{v}_k\mathbf{v}_i^T\right] = \mathbf{R}_k\delta_{ki} \tag{28.15}$$

$$E\left[\mathbf{w}_k\mathbf{w}_i^T\right] = \mathbf{Q}_k\delta_{ki} \tag{28.16}$$

Nói chung, có thể giả sử rằng nhiễu đo và nhiễu quá trình không tương quan, do vậy:

$$E\left[\mathbf{w}_k\mathbf{w}_i^T\right] = 0 \qquad \forall k, \forall i \tag{28.17}$$

Tuy nhiên, bộ lọc Kalman có thể được thay đổi để hoạt động với các hệ thống mà không cần giả thiết này [7].

Truyền phương sai lỗi của bộ lọc Kalman tuyến tính

Quá trình truyền các trạng thái lọc và tham chiếu trong LKF đã được đề cập ở phần trước ở phương trình (28.11) và (28.13). Tuy nhiên, tất cả các công thức của bộ lọc Kalman cũng phải

truyền một thước đo tin cậy cho ước lượng trạng thái dưới dạng của ma trận hiệp phương sai của sai số trạng thái. Hiệp phương sai của sai số trạng thái, **P**, được định nghĩa là kỳ vọng của tích ngoài vectơ sai số ước lượng:

$$\mathbf{P}_k^{(-)} = E\left[\delta\mathbf{x}_k^{-}\delta\mathbf{x}_k^{-T}\right] \tag{28.18}$$

Ma trận hiệp phương sai sai số trạng thái là ma trận đối xứng *n x n*, và phải xác định dương để tiếp tục ổn định bộ lọc. Cơ chế cho việc truyền hiệp phương sai nhận được bằng cách lấy các hiệp phương sai trước khi cập nhật phép đo tại thời điểm t_{k+1}:

$$\mathbf{P}_{k+1}^{(-)} = E\left[\delta\mathbf{x}_{k+1}^{(-)}\delta\mathbf{x}_{k+1}^{(-)T}\right] \tag{28.19}$$

Thay các định nghĩa và sai số ước lượng trong phương trình (28.4), (28.11), và (28.3) dẫn đến:

$$\mathbf{P}_{k+1}^{(-)} = E\left[\boldsymbol{\Phi}(t_{k+1},t_k)\left(\hat{\mathbf{x}}_k^{(-)} - \mathbf{x}_k\right)\left(\hat{\mathbf{x}}_k^{(+)} - \mathbf{x}_k\right)^T\boldsymbol{\Phi}(t_{k+1},t_k)^T + \mathbf{w}_k\mathbf{w}_k^T\right] \tag{28.20}$$

Sử dụng định nghĩa của hiệp phương sai nhiễu quá trình trong phương trình (28.16) và hiệp phương sai sai số trạng thái trong phương trình (28.18), phương trình truyền thu gọn thành:

$$\mathbf{P}_{k+1}^{(-)} = \boldsymbol{\Phi}(t_{k+1},t_k)\mathbf{P}_k^{(+)}\boldsymbol{\Phi}(t_{k+1},t_k)^T + \mathbf{Q}_k \tag{28.21}$$

Phương trình truyền có thể được hiểu như là tổng của ánh xạ hiệp phương sai sai số trước và sau khi cập nhật thông qua động lực hệ thống và nhiễu quá trình hệ thống bao gồm cả sai số. Do đó nhiễu quá trình làm tăng hiệp phương sai sai số trạng thái giữa các lần cập nhật giá trị đo.

Cập nhật bộ lọc Kalman tuyến tính

Bộ lọc Kalman tuyến tính (LKF – Linear Kalman Filter) tìm một lời giải phương sai nhỏ nhất cho trạng thái sai lệch, $\mathbf{x}_k$, bằng việc kết hợp thông tin trạng thái trước với các đo lường sẵn có. Do đó ước lượng trạng thái sau cập nhật phép đo được xem như là sự kết hợp tuyến tính của trạng thái trước khi cập nhật và thông tin đo được tuyến tính hóa:

$$\hat{\mathbf{x}}_k^{(+)} = \mathbf{K}_k'\hat{\mathbf{x}}_k^{(-)} + \mathbf{K}_k\mathbf{z}_k \tag{28.22}$$

Thay thế phương trình (28.4) và (28.14) vào phương trình (28.22) và xác định sai số ước lượng, ta được:

$$\delta\mathbf{x}_k^{(+)} = (\mathbf{K}_k' + \mathbf{K}_k\mathbf{H}_k - \mathbf{I})\mathbf{x}_k + \mathbf{K}_k'\delta\mathbf{x}_k^{(-)} + \mathbf{K}_k\mathbf{v}_k \tag{28.23}$$

Với việc xác định $E[\mathbf{v}_k] = \mathbf{0}$ và $E[\delta\mathbf{x}_k^{(-)}] = \mathbf{0}$ bằng cách giả sử ước lượng không lệch thì sai số ước lượng trạng thái cập nhật là không lệch:

$$E\left[\delta\mathbf{x}_k^{(+)}\right] = \mathbf{0} \tag{28.24}$$

chỉ khi:

$$\mathbf{K}_k' + \mathbf{K}_k\mathbf{H}_k - \mathbf{I} = 0 \tag{28.25}$$

Thay phương trình (28.35) vào phương trình (28.22) sẽ được một biểu thức cho ước lượng trạng thái cập nhật:

$$\hat{\mathbf{x}}_k^{(+)} = \hat{\mathbf{x}}_k^{(-)} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^{(-)}\right) \tag{28.26}$$

với sai số ước lượng:

$$\delta\mathbf{x}_k^{(+)} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\delta\mathbf{x}_k^{(-)} + \mathbf{K}_k\mathbf{v}_k \tag{28.27}$$

Hiệp phương sai sai số sau khi đo trong phương trình (28.18) có thể được mở rộng thành:

$$\mathbf{P}_k^{(+)} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{(-)}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{28.28}$$

bằng việc thay thế phương trình (28.27) và áp dụng nhiễu đo và quá trình không tương quan, nhiễu đo trung bình không, và việc xác định hiệp phương sai sai số ước lượng trạng thái trước khi đo. Ở đây, chỉ có yêu cầu bộ lọc Kalman là một bộ ước lượng không lệch được thoả mãn, do đó từ bây giờ chúng ta sẽ chọn hệ số khuếch đại Kalman $\mathbf{K}_k$ sao cho phương sai tổng nhỏ nhất trên sai số ước lượng trạng thái sau khi đo. Nói cách khác, chúng ta tìm hệ số khuếch đại cực tiểu hóa:

$$J_k = trace\left[\mathbf{P}_k^{(+)}\right] \tag{28.29}$$

Điều kiện cần để làm cho J_k nhỏ nhất là đạo hàm riêng đối với hệ số khuếch đại Kalman bằng không. Bằng việc sử dụng mối liên hệ sau:

$$\frac{\partial}{\partial \mathbf{A}}[trace(\mathbf{A}\mathbf{B}\mathbf{A}^T)] = 2\mathbf{A}\mathbf{B} \tag{28.30}$$

trong đó $\mathbf{B}$ là ma trận đối xứng. Áp dụng trên các thành phần của phương trình (28.28) với $\mathbf{K}_k$ dẫn đến:

$$(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{(-)}\mathbf{H}_k^T + \mathbf{K}_k\mathbf{R}_k = 0 \tag{28.31}$$

Hệ số khuếch đại tối ưu (khuếch đại Kalman) sẽ là:

$$\mathbf{K}_k = \mathbf{P}_k^{(-)}\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k^{(-)}\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1} \tag{28.32}$$

đôi khi còn được viết là:

$$\mathbf{K}_k = \mathbf{P}_k^{(-)}\mathbf{H}_k^T\mathbf{W}_k^{-1} \tag{28.33}$$

trong đó thành phần $\mathbf{W}_k$ là được coi như là hiệp phương sai mới

$$\mathbf{H}_k\mathbf{P}_k^{(-)}\mathbf{H}_k^T + \mathbf{R}_k \tag{28.34}$$

28.2 Các công thức bộ lọc Kalman khác

Cùng với bộ lọc Kalman tuyến tính (LKF), một số công thức khác của bộ lọc Kalman có thể được sử dụng để bám sát hơn theo đặc tính của đường quan sát trạng thái xác định. LKF có thể thay đổi theo bản chất tạm thời của hệ động lực và hệ đo là liên tục trong động lực và đo lường hoặc liên tục về động lực và rời rạc trong đo lường [12]. Mặc dù có nhiều ứng dụng khi hệ động lực mạnh hoặc chất lượng đo kém có thể làm trạng thái tham chiếu trong LKF rời nhanh khỏi miền tuyến tính của trạng thái môi trường. Trong những hệ thống như thế, trạng thái tham chiếu có thể được cập nhật bằng cách thêm các trạng thái lọc để trở thành bộ lọc Kalman mở rộng (EKF – Extended Kalman Filter). EKF rất phù hợp với các ứng dụng thời gian thực nhưng là phi tuyến trong trường hợp mà trạng thái tham chiếu được khởi tạo lại tại mỗi thời điểm cập nhật phép đo. LKF và EKF liên tục-rời rạc sẽ được phát triển trong các mục sau.

Bộ lọc Kalman tuyến tính rời rạc-liên tục

Thực tế, có thể nảy sinh một ứng dụng mà trạng thái tham chiếu, trạng thái bộ lọc và hiệp phương sai của sai số trạng thái được truyền hợp lý dưới dạng liên tục thông qua ứng dụng tuyến tính của ma trận dịch chuyển trạng thái. Hơn nữa, hệ thống đo thường có thể truyền các quan sát thời gian rời rạc ngay cả khi động lực được mô hình liên tục. Trong tình huống như thế, cơ chế cập

nhật không được thay đổi từ đạo hàm LKF trước, trong khi đó việc truyền giữa các lần cập nhật được tiến hành bằng cách tích hợp liên tục. Không mất tính tổng quát, có thể mô tả động lực tham chiếu của bộ lọc Kalman liên tục như sau:

$$\dot{\tilde{\mathbf{X}}}(t) = f(\tilde{\mathbf{X}}(t), \boldsymbol{\alpha}, t) \tag{28.35}$$

Hơn nữa, bằng việc lấy đạo hàm theo thời gian trạng thái lọc và hàm truyền hiệp phương sai (phương trình (28.11) và (28.21)) và thay vào phương trình (28.13) cho đạo hàm ma trận chuyển trạng thái, chúng ta tìm được mối liên hệ giữa hiệp phương sai và trạng thái lọc thời gian liên tục:

$$\dot{\hat{\mathbf{x}}}^{(-)}(t) = f(\tilde{\mathbf{X}}(t), \boldsymbol{\alpha}, t) + \mathbf{F}(\tilde{\mathbf{X}}(t), \boldsymbol{\alpha}, t)[\hat{\mathbf{x}}^{(-)}(t) - \tilde{\mathbf{X}}(t)] \tag{28.36}$$

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^{T}(t) + \bar{\mathbf{Q}}(t) \tag{28.37}$$

ở đây $\bar{Q}(t)$ là mật độ phổ của nhiễu quá trình động lực tại thời điểm t và sự phụ thuộc chức năng tương minh của **F** được bỏ qua nhằm thuận tiện cho việc ký hiệu. Theo cách này của LKF, ma trận chuyển trạng thái không cần phải tính toán do động lực được tính trực tiếp qua ma trận đạo hàm riêng và trạng thái tham chiếu, trạng thái lọc và hiệp phương sai sai số được truyền liên tục.

Nhiễu quá trình và đo trong công thức này là liên tục và rời rạc với các mô hình tương ứng, và vẫn được giả thiết là các quá trình trung bình không được chi phối bởi hiệp phương sai nhiễu quá trình động lực liên tục:

$$E[\mathbf{w}(t)\mathbf{w}^{T}(\tau) = \mathbf{Q}(t)\delta(t-\tau) \tag{28.38}$$

và hiệp phương sai nhiễu đo rời rạc:

$$\mathbf{E}\left[\mathbf{v}_k \mathbf{v}_l^{T}\right] = \mathbf{R}_k \delta_{kl} \tag{28.39}$$

Cũng cần giả sử nhiễu quá trình và đo là không tương quan, do đó:

$$E\left[\mathbf{w}(t)\right]\mathbf{v}_k^{T} = 0 \tag{28.40}$$

mặc dù có thể điều chỉnh công thức để bao gồm cả tương quan nhiễu quá trình và đo nếu cần thiết [7].

Bộ lọc Kalman mở rộng rời rạc-liên tục

Trong các ứng dụng mà trạng thái tham chiếu có thể nhanh chóng xa rời vùng tuyến tính của trạng thái môi trường, có thể cập nhật trực tiếp tham chiếu tại thời điểm cập nhật giá trị đo bằng cách thêm trạng thái bộ lọc LKF vào tham chiếu trong EKF. EKF tương tự như LKF, trong đó các giá trị đo được xử lý để cung cấp một ước lượng sai khác giữa trạng thái thật và trạng thái tham chiếu của miền không gian. EKF cũng đánh giá các phần riêng động lực và đo đối với trạng thái tham chiếu dưới dạng tương tự với LKF. Tuy nhiên, trang thái tham chiếu được hiệu chỉnh bằng cách thêm vào thông tin đo:

$$\tilde{\mathbf{X}}(t_k)^{(+)} = \tilde{\mathbf{X}}(t_k)^{(-)} + \hat{\mathbf{x}}(t_k) \tag{28.41}$$

Mô hình động lực trạng thái tham chiếu dùng trong công thức EKF được cho bởi phương trình (28.35), nhưng mô hình đo là dạng rời rạc cho bởi công thức (28.1). Trạng thái bộ lọc biểu diễn sai khác ước lượng giữa trạng thái thật và trạng thái tham chiếu chỉ được tính toán tại thời điểm cập nhật đo lường thông qua việc bỏ qua thành phần thông tin ước lượng trước từ phương trình (28.26):

$$\hat{\mathbf{x}}_k = \mathbf{K}_k \mathbf{Z}_k \tag{28.42}$$

trong đó có sự khác biệt là số dư giá trị đo thực:

$$\mathbf{Z}_k = \mathbf{Y}_k - \mathbf{h}\left(\tilde{\mathbf{X}}_k^{(-)}, \boldsymbol{\beta}, t_k\right) \tag{28.43}$$

Vì vậy, trong EKF không có trạng thái bộ lọc riêng biệt cần được truyền tới thời điểm của lần đo tiếp theo vì trạng thái bộ lọc đã được sát nhập vào trạng thái tham chiếu cập nhật.

Hiệp phương sai sai số tại mỗi giá trị đo được cập nhật bởi phương trình:

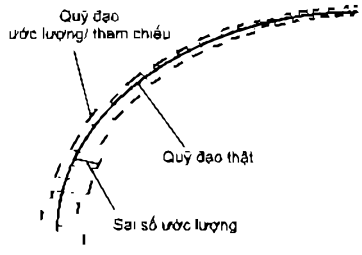
$$\mathbf{P}_k^{(+)} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{(-)}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{28.44}$$

và hệ số khuếch đại Kalman EKF và hiệp phương sai mới tương tự với bản sao LKF của nó

$$\mathbf{K}_k = \mathbf{P}_k^{(-)}\mathbf{H}_k\mathbf{W}_k^{-1} \tag{28.45}$$

$$\mathbf{W}_k = \mathbf{H}_k\mathbf{P}_k^{(-)}\mathbf{H}_k + \mathbf{R}_k \tag{28.46}$$

Sự khác biệt giữa hoạt động của EKF và hoạt động của LKF được minh họa bằng cách xem lại quỹ đạo hai chiều ở hình 28.2. Quỹ đạo tham chiếu bây giờ được xem như để đáp ứng thông tin đo lường và bám theo quỹ đạo môi trường thực.



HÌNH 28.2 EKF bám theo một quỹ đạo hai chiều

28.3 Tổng kết và tóm tắt các công thức

Công thức LKF rời rạc-rời rạc được đưa ra bởi các phương trình truyền sau:

$$\tilde{\mathbf{X}}_{k+1} = \Phi(t_{k+1}, t_k)\tilde{\mathbf{X}}_k$$
$$\hat{\mathbf{x}}_{k+1}^{(-)} = \Phi(t_{k+1}, t_k)\hat{\mathbf{x}}_k^{(-)}$$
$$\mathbf{P}_{k+1}^{(-)} = \Phi(t_{k+1}, t_k)\mathbf{P}_k^{(-)}\Phi(t_{k+1}, t_k)^T + \mathbf{Q}_k$$

và các phương trình cập nhật:

$$\hat{\mathbf{x}}_k^{(+)} = \hat{\mathbf{x}}_k^{(-)} + \mathbf{K}_k\left[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^{(-)}\right]$$
$$\mathbf{P}_k^{(+)} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{(-)}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T$$
$$\mathbf{K}_k = \mathbf{P}_k^{(-)}\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k^{(-)}\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1}$$

Trong cơ chế LFK thời gian rời rạc, trạng thái tham chiếu không bị ảnh hưởng bởi sự kết hợp của thông tin đo vào trạng thái bộ lọc.

Trong sự thay đổi nhỏ của cách tiếp cận này, động lực của LKF có thể sẽ liên tục, trạng thái bộ lọc, trạng thái tham chiếu, và hiệp phương sai được truyền mà không sử dụng ma trận chuyển trạng thái:

$$\dot{\tilde{\mathbf{X}}}(t) = f(\tilde{\mathbf{X}}(t), \alpha, t)$$
$$\dot{\hat{\mathbf{x}}}^{(-)}(t) = F(\tilde{\mathbf{X}}(t), \alpha, t)\hat{\mathbf{x}}^{(-)}(t)$$
$$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \bar{\mathbf{Q}}(t)$$

Khi ứng dụng yêu cầu trạng thái tham chiếu phải thay đổi để vẫn nằm lân cận tuyến tính của trạng thái môi trường, công thức EKF liên tục-rời rạc có thể phù hợp. Trong công thức EKF liên tục-rời rạc, sự lan truyền được tiến hành theo:

$$\dot{\tilde{\mathbf{X}}}(t) = f(\tilde{\mathbf{X}}(t), \mathbf{a}, t)$$
$$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^{T}(t) + \bar{\mathbf{Q}}(t)$$

và cập nhật đo lường theo:

$$\tilde{\mathbf{X}}(t_k)^{(-)} = \tilde{\mathbf{X}}(t_k)^{(-)} + \hat{\mathbf{x}}(t_k)$$
$$\hat{\mathbf{x}}_k = \mathbf{K}_k\left[\mathbf{Y}_k - \mathbf{h}\left(\tilde{\mathbf{X}}_k^{(-)}, \boldsymbol{\beta}, t_k\right)\right]$$
$$\mathbf{P}_k^{(+)} = (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{(-)}(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)^{T} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_K^{T}$$
$$\mathbf{K}_k = \mathbf{P}_k^{(-)}\mathbf{H}_k^{T}\left[\mathbf{H}_k\mathbf{P}_k^{(-)}\mathbf{H}_k^{T} + \mathbf{R}_k\right]^{-1}$$

Trạng thái tham chiếu sẽ thay đổi cùng với sự kết hợp của thông tin đo lường vào trong EKF và các bộ phận được đánh giá theo tham chiếu thay đổi này.

28.4 Xem xét sự thực hiện

Thực hiện các công thực liệt kê ở trên chỉ là một phần nỗ lực cần thiết để phát triển một ứng dụng bộ lọc Kalman bền vững và chính xác. Khi đã mã hóa động lực, các giá trị đo và đạo hàm riêng, nhiệm vụ còn lại là hiệu chỉnh cường độ nhiễu được thể hiện ở hiệp phương sai nhiễu quá trình **Q** và hiệp phương sai nhiễu phép đo **R**. Trong khi nhiễu phép đo được dựa trên sự xác định rõ khả năng thực hiện của phần cứng, thì nhiễu quá trình thường xuyên sử dụng như là tham số hiệu chỉnh để chắc chắn rằng bộ lọc hoạt động chính xác. Bản liệt kê của bộ lọc Kalman được đưa ra sau đây được dành cho những người mới nghiên cứu lĩnh vực này nhằm giảm thời gian thực hiện và hiệu chỉnh đường cong học:

- Do bộ lọc Kalman tuyến tính không thay đổi trạng thái tham chiếu khi có mặt của thông tin đo lương, nên trạng thái tham chiếu và các đạo hàm riêng của một ứng dụng LKF có thể được tính toán trước khi hoạt động. Điều này làm cho LKF dễ thực hiện đối với ứng dụng hạn chế về khả năng tính toán hoặc thử nghiệm giả thiết mà ở đó các tham số nhiễu quá trình và nhiễu đo lường khác đang được ước lượng song song [8].
- Nhiễu quá trình giúp bộ lọc tránh được tình trạng quá tin cậy vào ước lượng của nó (tức là một hiệp phương sai với các giá trị đường chéo gần 0) và hội tụ sớm. Bằng cách kiểm tra phương trình truyền cho bộ lọc Kalman đã được trình bày, có thể dễ dàng thấy được rằng nhiễu quá trình thêm vào làm tăng cường độ của hiệp phương sai sai số trạng thái giữa các giá trị đo như thế nào.
- Hiệp phương sai cải tiến nên hội tụ lý tưởng để miêu tả phương sai trong phần dư đo lường bộ lọc. Trong một bộ lọc lý tưởng, hiệp phương sai cải tiến cần tiệm cận hiệp phương sai nhiễu đo lường khi cường độ nhiễu quá trình tiến đến 0.
- Khi nhiều đo lường có sẵn tại cùng thời điểm, có thể xử lý chúng như một chuỗi các bộ quan sát vô hướng miễn là chúng không tương quan với nhau (tức là **R** là một ma trận đường chéo). Ảnh hưởng của quá trình đo vô hướng là việc hiệp phương sai cải tiến trở thành một đại lượng vô hướng, và cần một phép chia số học thay vì một phép đảo ma trận để tính hệ số khuếch đại Kalman.
- Có thể sử dụng sự hiệu chỉnh phép đo để ngăn chặn dữ liệu không chính xác từ nguyên nhân phân kỳ bộ lọc trong một số trường hợp. Một trong những cách thông thường nhất là loại bỏ các phép đo khi tỷ số của bình phương số dư phép đo trên hiệp phương sai cải tiến vô hướng:

$$\frac{r_k^2}{W_k} \tag{28.47}$$

lớn hơn giới hạn do người dùng đặt ra. Giá trị giới hạn này có thể là một hằng số hoặc biến đổi với thời gian sau các chu kỳ truyền dài nhằm cho phép một chuyển đổi nhẹ nhàng tới hiệp phương sai cải tiến có trạng thái ổn định:

- Hiệp phương sai luôn luôn xác định dương. Nếu sự phân kỳ bộ lọc là một bài toán thường xuyên trong một ứng dụng cụ thể, thì tính toàn vẹn số của hiệp phương sai cho phép hiểu rõ bản chất của sự phân kỳ. Cũng có vài cách giải quyết số cho phương trình cập nhật hiệp phương sai nhờ tính xác định dương và đối xứng để nâng cao tính ổn định trong khi giảm khối lượng tính toán [9].

Cuối cùng cần nhấn mạnh rằng bộ lọc Kalman không phải là thuật toán bộ quan sát trạng thái thích hợp cho tất cả các ứng dụng. Sức mạnh của nó là khả năng thực hiện thời gian thực ước lượng trạng thái với khối lượng tính toán ít trong khi thiếu thông tin đo chính xác. Tuy nhiên, các kĩ thuật ước lượng nhóm như ước lượng bình phương nhỏ nhất có thể thích hợp hơn trong ứng dụng mà ở đó quá trình động lực được mô hình hóa để có cấp chính xác cao hơn, các đo lường không hoàn toàn giống nhau, và hoạt động thời gian thực không phải là một vấn đề. Một số bài trong [10-12] viết về chủ đề ước lượng ngẫu nhiên nói chung và lọc Kalman nói riêng mà bạn đọc nên tìm hiểu để có các thông tin chi tiết hơn.

Tài liệu tham khảo

[1] Kalman, R. E., "A new approach to linear filtering and prediction problems," *Transactions of the ASME, Ser. D, Journal of Basic Equations*, March 1960, pp. 35–45.

[2] Burkhart, P. and Bishop, R., "Adaptive orbit determination for interplanetary spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1997, pp. 693–701.

[3] Chaer, W., Bishop, R., and Ghosh, J., "Hierarchical adaptive Kalman filtering for interplanetary orbit determination," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 3, 1998, pp. 1–14.

[4] Crain, T. and Bishop, R., "The mixture-of-experts gating network: integration into the ARTSN extended Kalman filter," Technical Memorandum CSR-TM-99-01, Center for Space Research, March 1999.

[5] Ely, T., Bishop, R., and Crain, T., "Adaptive interplanetary navigation using genetic algorithms," *The Journal of Astronautical Sciences*, 2000, Accepted for Publication.

[6] Crain, T. and Bishop, R., "Unmodeled impulse detection and identification during Mars pathfinder cruise," Technical Memorandum CSR-TM-00-01, Center for Space Research, March 2000.

[7] Chaer, W. and Bishop, R., "Adaptive Kalman filtering with genetic algorithms," *Advances in the Astronautical Sciences*, edited by R. Proulx, J. Liu, P. Siedelmann, and S. Alfano, Vol. 89, Univelt, San Diego, CA, 1995, pp. 141–156, Pt. 1.

[8] Gholson, N. and Moose, R., "Maneuvering target tracking using adaptive state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 13, No. 3, May 1997, pp. 310–317.

[9] Bierman, G., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, 1977.

[10] Brown, R. G. and Huang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley and Sons, 1992.

[11] Lewis, F., *Applied Optimal Control and Estimation*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[12] Gelb, A., *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, MA, 1974.

29

Xử lý tín hiệu số trong các ứng dụng Cơ điện tử

Bonnie S. Heck
Georgia Institute of Technology

Tomas R. Kurfess
Georgia Institute of Technology

29.1 Giới thiệu 29-1
29.2 Cơ sở về xử lý tín hiệu 29-1
29.3 Ánh xạ tín hiệu liên tục sang tín hiệu rời rạc 29-4
29.4 Thiết kế bộ lọc số 29-7
29.5 Thiết kế điều khiển số 29-16

29.1 Giới thiệu

Đa số các kỹ sư trong lĩnh vực Cơ điện tử chỉ làm với các hệ thống hoàn toàn là cơ khí hoặc điện tử. Có rất nhiều hệ thống điện hỗ trợ các hệ cơ khí và ngược lại. Ví dụ, hầu hết các bộ vi xử lý trong máy tính ngày nay đều có bộ tản nhiệt và quạt làm mát để máy tính hoạt động trong vùng nhiệt độ ổn định. Các hệ thống điện được sử dụng rộng rãi để giám sát và điều khiển rất nhiều hệ cơ khí. Với sự giảm giá của các chip xử lý số thì các bộ lọc số và điều khiển số cho các hệ cơ khí càng trở nên phổ biến. Các ví dụ về trường hợp này có thể thấy trong xe ô tô và hầu hết các thiết bị gia đình. Ví dụ, các tín hiệu cảm biến sử dụng trong giám sát và điều khiển các hệ cơ khí đòi hỏi vài dạng xử lý tín hiệu. Dạng xử lý tín hiệu này có thể từ tín hiệu "quét trên" đơn giản dùng một bộ lọc thông thấp đến việc phân tích cao cấp hơn như giám sát mô men và công suất trong động cơ servo một chiều. Chương này trình bày khái quát các phương pháp xử lý tín hiệu số phù hợp trong các hệ cơ khí. Trong khuôn khổ giới hạn của chương này, không cho phép phân tích tận gốc hoặc chi tiết. Nếu cần biết chi tiết hơn, bạn đọc có thể xem trong tài liệu tham khảo [1, 2].

29.2 Cơ sở về xử lý tín hiệu

Một vài khái niệm cơ bản về xử lý tín hiệu sẽ được giới thiệu trước khi thảo luận về vấn đề lọc hoặc điều khiển.

Tín hiệu liên tục

Biến đổi Laplace dùng để phân tích hệ thống liên tục theo thời gian, tìm đáp ứng hệ thống, và thiết kế điều khiển. Biến đổi Laplace một phía của tín hiệu liên tục, $x(t)$, được cho bởi công thức:

$$X(s) = \int_0^\infty x(t)e^{-st}dt$$

Một hàm truyền của hệ tuyến tính, $H(s)$, là tỷ số của biến đổi Laplace của đầu ra trên biến đổi Laplace đầu vào (với điều kiện ban đầu là 0).

Biến đổi Fourier được dùng để xác định phổ tần tín hiệu. Biến đổi Fourier của $x(t)$ được cho bởi công thức:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt \tag{29.1}$$

trong đó ω có đơn vị là radian/giây. Lưu ý rằng khi $x(t) = 0$ với $t \le 0$, biến đổi Laplace tương đương với biến đổi Fourier bằng cách đặt $s = j\omega$. Đáp ứng tần số của hệ thống được định nghĩa là tỷ số của biến đổi Fourier đầu ra trên biến đổi Fourier đầu vào của nó. Hoàn toàn tương đương, có thể nhận được từ hàm truyền bằng cách thay $H(\omega)$ bằng $H(\omega) \equiv H(j\omega) = H(s)\big|_{s=j\omega}$. Để đơn giản, j không được đưa vào danh sách các tham số, chỉ ghi kí hiệu là $H(\omega)$ để mô tả đáp ứng tần số. Băng thông của một hệ thống được định nghĩa là đáp ứng tại $H(\omega) = 0.707H(0)$.

Tín hiệu rời rạc

Biến đổi z được sử dụng để giải phương trình sai phân và phân tích hệ thống. Biến đổi z của tín hiệu rời rạc, x[n], được định nghĩa là:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

Biến đổi Fourier rời rạc (DTFT – discrete-time Fourier transform) được sử dụng để xác định phổ tần của tín hiệu. DTFT và DTFT ngược của một tín hiệu được định nghĩa như sau:

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} \tag{29.2}$$

và

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\Omega)e^{j\Omega n}\,d\Omega \tag{29.3}$$

Chú ý rằng DTFT có thể nhận được từ biến đổi z bằng cách đặt $z = e^{j\Omega}$. Do DTFT theo chu kỳ 2π, nên thường được biểu diễn trên khoảng $[-\pi, \pi]$ hoặc $[0, 2\pi]$, trong đó các tần số thông dụng là từ $\Omega = 0$ (tần số thấp) đến $\Omega = \pi$ (tần số cao). Đáp ứng tần số của một hệ rời rạc là tỷ số giữa DTFT của tín hiệu đầu ra và DTFT của tín hiệu đầu vào. Mặt khác, nó cũng có thể nhận được từ hàm truyền do $H(\Omega) \equiv H(e^{j\Omega}) = H(z)\big|_{z=e^{j\Omega}}$. Để đơn giản, ký hiệu $H(\Omega)$ hay được dùng hơn so với $H(e^{j\Omega})$. Cũng giống như trường hợp tín hiệu liên tục, băng thông được định nghĩa là tần số tại đó $H(\omega) = 0.707H(0)$.

Trong khi DTFT liên tục đối với biến số tần số Ω thì biến đổi Fourier rời rạc (DFT – discrete Fourier transform) chứa các điểm rời rạc đối với tham số k. Xét trong khoảng giới hạn $x[n]$, trong đó $x[n] = 0$ với $n < 0$ và với $n \ge N$. DFT và DTF ngược của $x[n]$ được định nghĩa là:

$$X_k = \sum_{n=0}^{N-1} x[n]e^{-j2\pi nk/N}, \qquad k = 0,1,\ldots,N-1 \tag{29.4}$$

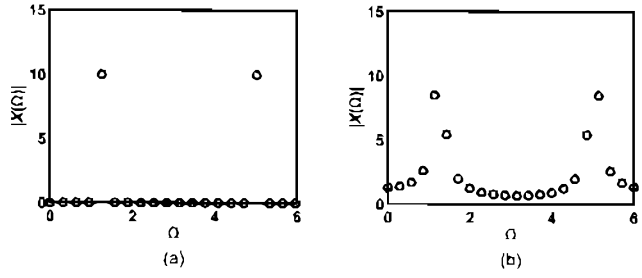
và

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j2\pi nk/N}, \qquad n = 0,1,\ldots,N-1$$

Chú ý rằng DFT là biến đổi được rời rạc hóa của DTFT với $X_k = X(\Omega)\big|_{\Omega=2\pi k/N}$ trong dải Ω từ 0 đến 2π. Việc tính công thức tổng quát cho DTFT có thể chỉ được làm với các tín hiệu đơn giản như là xung vuông hoặc xung tam giác. Vì vậy, DFT thường được sử dụng như là một phương pháp

số để tính DTFT tại các điểm rời rạc với tần số thuộc rải từ 0 đến 2π. Trong thực tế, để thu được một đồ thị của DTFT, phải vẽ đồ thị của X_k theo k, ở đây k tỷ lệ với $2\pi / N$. Với một tín hiệu bất kỳ, ví dụ như các tín hiệu đo từ một thiết bị, việc tính toán DFT thay vì DTFT là một phương pháp hoàn hảo hơn để tìm phổ tần của tín hiệu. Để tăng độ phân giải khi vẽ một DTFT từ các điểm được tính bởi DFT, các điểm không được thêm vào cuối của dãy làm cho giá trị N được tăng lên.

Với giả sử tín hiệu trong miền thời gian không bị hạn chế về mặt thời gian quan sát sao cho không có giá trị N làm cho $x[n] = 0$ với $n > N$. Để thực hiện DFT, tín hiệu cần được chia nhỏ. Có 2 trường hợp xảy ra: Trường hợp $x[n]$ suy giảm thành 0 và trường hợp x[n] có các thành phần chu kỳ. Trường hợp x[n] suy giảm về 0 thì N phải được chọn đủ lớn để các giá trị không đáng kể có thể bỏ qua. DFT nhận được là một xấp xỉ (không phải là phiên bản được rời rạc) của DTFT. Nếu tín hiệu là chu kỳ, DTFT không thể được tính bằng phương pháp số vì kết quả DTFT có các đáp ứng xung. Tuy nhiên, các tần số biểu diễn trong tín hiệu có thể vẫn được xác định nếu giá trị N dùng để cắt được chọn sao cho tín hiệu được cắt đi bằng một số nguyên lần chu kỳ. Nếu không DFT thu được sẽ bị "rò rỉ" trong đồ thị tần số khi so sánh với tín hiệu DTFT thực. Ví dụ, xét một tín hiệu $x[n] = \cos(0.4\pi n)$. Đây là tín hiệu tuần hoàn với chu kỳ $n = 5$ và có DTFT được cho bởi công thức $X(\Omega) = \pi[\delta(\Omega+0.4\pi)]+\delta(\Omega-0.4\pi)]$ với $-\pi \le \Omega \le \pi$. Tất cả phổ tần được đặt tại $\Omega = 0.4\pi$ và $\Omega = -0.4\pi$. DTFT có chu kỳ 2π, cũng có một xung tại $\Omega = 2\pi - 0.4\pi$. DFT được tính cho 2 điểm chặt của tín hiệu, một tại $N = 20$ (bằng 4 lần chu kỳ) và tại một điểm khác $N = 22$. DFT với $N = 20$ được vẽ trên hình 29.1(a) trong đó biến độc lập k được quét $2\pi / N$ cho mỗi điểm. Đồ thị này xuất hiện phổ tần là các điểm 0 ngoại trừ tại điểm $\Omega = 0.4\pi (= 1.2566)$ và $\Omega = 2\pi - 0.4\pi (= 5.0265)$, đây chính là vị trí chính xác của xung trong DTFT. Tương tự như vậy, DFT với $N = 22$ được vẽ như trên hình 29.1(b), chú ý kết quả có sự "rò rỉ" trong đặc tính tần.



HÌNH 29.1 DFT của tín hiệu tuần hoàn (a) cắt sau 4 chu kì (b) cắt sau 4.4 chu kì

Nếu một tín hiệu có phổ tuần hoàn, nhưng nó không phải là tín hiệu tuần hoàn, ví dụ như $x[n] = \cos(0.5n) + \cos(0.2n)$, thì việc rò rỉ là không thể tránh khỏi bằng cách chọn N. Một cách thay đổi để giảm rò rỉ là để đỉnh thứ nhất của tín hiệu bằng không tại điểm đầu và điểm cuối trước khi tính toán DFT. Quá trình này, được hiểu là *cửa sổ* dữ liệu, được thực hiện bằng việc nhân x[n] với một hàm cửa sổ $w[n]$ và sau đó thực hiện DFT tích số $x[n]w[n]$. Ba cửa sổ thường dùng là cửa sổ hình chữ nhật để chặt cụt đỉnh nhọn, cửa sổ Hanning và cửa sổ Hamming [1].

Cửa sổ chữ nhật:

$$w[n] = 1, \qquad 0 \le n \le N-1$$

Cửa sổ Hanning:

$$w[n] = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right), \qquad 0 \le n \le N-1$$

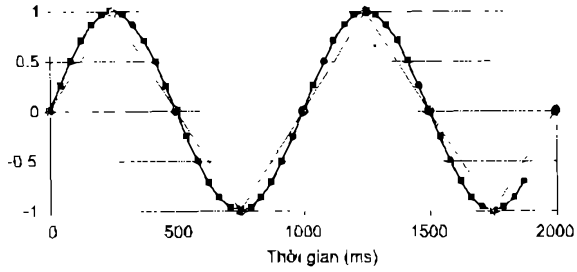
Cửa sổ Hamming:

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

Nếu giá trị của N là lũy thừa của 2, có một phương pháp để tính DFT được gọi là biến đổi Fourier nhanh (FFT – Fast Fourier Transform). Nếu giá trị N không là lũy thừa của 2, các điểm không có thể được thêm vào phần cuối của tín hiệu để dùng FFT. Điều này không ảnh hưởng đến độ chính xác của kết quả, nhưng nó làm tăng độ phân giải của đồ thị thu được khi DFT (hoặc FFT) được sử dụng để tính DTFT. Trong nhiều trường hợp, biểu thức được sử dụng trong (29.4) đủ để tính DFT vì khả năng tính toán của các bộ vi xử lý ngày nay đã giảm bớt yêu cầu số hóa FFT. Chi tiết về thuật toán của FFT không nằm trong nội dung của cuốn sách này. Bạn đọc có thể tham khảo trong [1] hoặc [2] để biết chi tiết.

29.3 Ánh xạ tín hiệu liên tục sang tín hiệu rời rạc

Trong khi hầu hết các hệ vật lý hoạt động trong thời gian liên tục, các máy tính hoạt động trong thời gian rời rạc. Vì vậy, để sử dụng máy tính trong xử lý các đo lường được lấy từ các hệ liên tục, cần phải có các phương pháp ánh xạ giữa thế giới liên tục và thế giới rời rạc.



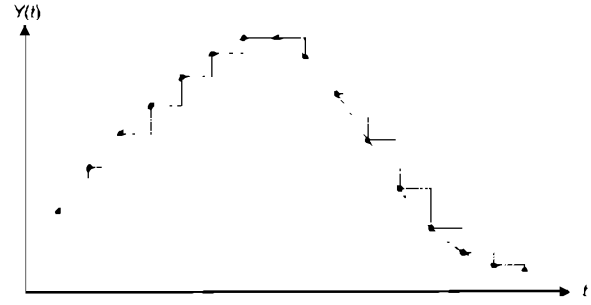
HÌNH 29.2 Tín hiệu 1Hz

Rời rạc hóa

Trước khi phân tích một tín hiệu tương tự bằng các kỹ thuật số, cần phải rời rạc hoá nó (nghĩa là chuyển thành tín hiệu rời rạc). Phương pháp lý tưởng để rời rạc hóa là *lấy mẫu*, ở đó các giá trị của tín hiệu được xác định tại các điểm rời rạc về mặt thời gian. Thông thường, tín hiệu được lấy mẫu với tốc độ cố định được gọi là *chu kỳ lấy mẫu*. Tốc độ lấy mẫu (đơn vị là Hz) là số nghịch đảo của chu kỳ lấy mẫu. Hình 29.2 mô tả tín hiệu 1Hz được lấy mẫu tại 2 tốc độ. Các điểm sẫm màu được lấy mẫu với chu kỳ 15ms, các điểm sáng màu hơn được lấy mẫu với chu kỳ 250 ms. Từ hình 29.2, có thể thấy rằng xấp xỉ dạng sóng suy biến rõ ràng khi giảm tần số lấy mẫu và nó tiệm cận đến tần số tín hiệu. Trong thực tế, có thể thấy rằng một tín hiệu phải được lấy mẫu tại một tần số lớn hơn 2 lần phổ tần lớn nhất của nó. Đó chính là nội dung Định lý lấy mẫu Nyquist. Ví dụ, nếu tín hiệu trong hình 29.2 được lấy mẫu tại 0.5 Hz, thì tất cả các mẫu sẽ có giá trị 0 vì tại 0, 500, 1000 ms ... giá trị của tín hiệu là 0. Hiện tượng biểu diễn không đúng tín hiệu do tần số lấy mẫu quá thấp được gọi là hiện tượng "răng cưa". Có hai phương pháp để thỏa mãn định lý lấy mẫu Nyquist. Phương pháp thứ nhất là dùng một tần số lấy mẫu lớn gấp 2 lần phổ tần cao nhất của tín hiệu đang được lấy mẫu. Tần số này được gọi là tần số Nyquist. Do không chắc chắn đó là phổ tần thực sự của tín hiệu thực, cần dùng một bộ lọc thông thấp để đảm bảo rằng một tín hiệu không bị ảnh hưởng bởi các tần số cao hơn một mức giới hạn nhất định. Bộ lọc đó được gọi là bộ lọc chống răng cưa. Đây là phương pháp thứ 2 và là phương pháp thực hiện nhiều trong thực tế, nó được sử dụng để thỏa mãn định lý lấy mẫu Nyquist. Vì vậy, sự kết hợp của một bộ lọc khử răng cưa được thiết kế tốt với một tần số lấy mẫu lớn hơn tần số giới hạn của bộ lọc sẽ đảm bảo rằng định lý lấy mẫu Nyquist được thỏa mãn.

Có 2 điểm quan trọng cần được chú ý khi sử dụng một bộ lọc chống răng cưa. Thứ nhất, bộ lọc khử răng cưa được dùng trước khi tín hiệu được lấy mẫu vì việc lấy mẫu gây ra hiện tượng răng cưa. Về cơ bản, điều này đòi hỏi bộ lọc chống răng cưa được thực hiện sử dụng một bộ lọc tương tự trước khi tín hiệu được số hóa. Khi tín hiệu bị răng cưa trong quá trình lấy mẫu thì không thể dùng bộ lọc số để hiệu chỉnh. Điểm thứ 2 là trong thực tế tần số giới hạn của bộ lọc khử răng cưa

nên chọn nhỏ hơn từ 5 đến 10 tần số Nyquist. Nên chú ý rằng bộ lọc khử răng cưa đưa thêm trễ pha vào giá trị đo, nó có thể làm giảm tính ổn định và khả năng làm việc trong một vòng phản hồi trừ khi băng thông của bộ lọc khử răng cưa lớn hơn băng thông của hệ kín rất nhiều. Các thiết bị trên thị trường hiện nay dùng để thực hiện lấy mẫu là các bộ biến đổi tương tự - số (ADC), và bộ lọc khử răng cưa được sử dụng trước thiết bị này.



HÌNH 29.3 Một tín hiệu được lấy mẫu và xây dựng lại bằng bộ giữ mẫu bậc không (ZOH)

Ngược lại của việc lấy mẫu là tái tạo khi một tín hiệu rời rạc được chuyển đổi sang một tín hiệu liên tục. Tốc độ lấy mẫu Nyquist đảm bảo rằng nếu một tín hiệu liên tục được lấy mẫu tại một tốc độ mà tốc độ đó ít nhất gấp 2 lần thành phần tần số cao nhất trong tín hiệu, thì tín hiệu liên tục có thể được tái tạo lại chính xác từ tín hiệu lấy mẫu. Tuy nhiên, định lý này giả thiết rằng mọi quá trình tái tạo là lý tưởng, điều này không thực tế. Hầu hết các cách thực hiện tái tạo một tín hiệu là giữ mẫu bậc không (ZOH). ZOH giả sử giá trị của tín hiệu là hằng số giữa các lần lấy mẫu. Sự xấp xỉ này là khá hợp lý nếu tín hiệu được lấy mẫu về cơ bản không thay đổi giữa các mẫu riêng biệt. Hình 29.3 là một ví dụ của tín hiệu và biểu diễn giữ mẫu bậc không của nó. Đường trơn biểu hiện tín hiệu tương tự ban đầu. Các điểm mẫu đen dọc theo tín hiệu chỉ ra các giá trị lấy mẫu của tín hiệu. Mỗi điểm đen được nối tới điểm tiếp theo bằng một đường nằm ngang sau đó là đường thẳng đứng. Đường nằm ngang mô tả việc giữ mẫu bậc không, giá trị của tín hiệu vẫn giữ là hằng số giữa các giá trị lấy mẫu. Đường thẳng đứng khẳng định rằng tín hiệu đó không giữ nguyên hằng số trong suốt chu kỳ lấy mẫu. Khi khoảng thời gian giữa các điểm lấy mẫu được tăng lên, độ chính xác của việc giữ mẫu bậc không bị giảm đi. Ngược lại, khi chu kỳ lấy mẫu giảm, nó sẽ cải thiện độ chính xác của giữ mẫu bậc không. Các thiết bị trên thị trường hiện nay thực hiện tái tạo là các bộ biến đổi số - tương tự DAC, thường sử dụng phương pháp giữ mẫu bậc không.

Ánh xạ từ mặt phẳng *s* sang mặt phẳng *z*

Một phương pháp liên hệ mặt phẳng s với mặt phẳng z bắt đầu từ biểu diễn toán học của tín hiệu liên tục được lấy mẫu $x(t)$ và tính toán biến đổi Laplace của nó. Biến đổi Laplace của tín hiệu đã lấy mẫu được liên hệ với biến đổi z của $x[n]$ bằng cách đặt $z = e^{sT}$ trong đó T là chu kỳ lấy mẫu. Biểu thức $z = e^{sT}$ thường được gọi là *ánh xạ chính xác* giữa mặt phẳng z và mặt phẳng s. (Để biết chi tiết hơn, xem [1]). Ví dụ, một biểu diễn số, $H_d(z)$, của một hệ liên tục, $H(s)$, có thể nhận được bằng cách dùng ánh xạ $H_d(z) = H(s)\big|_{z=e^{sT}}$. Tuy nhiên, ánh xạ này dẫn đến $H_d(z)$ là một hàm không hữu tỷ.

Ánh xạ xấp xỉ giữa mặt phẳng s và mặt phẳng z thường được sử dụng để dẫn đến một hàm hữu tỷ cho $H_d(z)$. Ba ánh xạ đó là biến đổi song tuyến, biến đổi thuận và biến đổi ngược.

Biến đổi song tuyến:

$$s = \frac{2(z-1)}{T(z+1)}$$

Biến đổi thuận:

$$s = \frac{1}{T}(z-1)$$

Biến đổi ngược:

$$s = \frac{1}{Tz}(z-1)$$

Biến đổi song tuyến (còn được gọi là quy tắc Tustin hoặc quy tắc hình thang) là biến đổi chính xác nhất trong các ánh xạ. Nó ánh xạ toàn bộ miền bên trái của mặt phẳng s sang vòng tròn đơn vị của mặt phẳng z, vì vậy nó duy trì được tính ổn định. Xét một ví dụ bậc nhất $H(s) = 1/(s+2)$. Biểu diễn rời rạc của hàm truyền này là:

$$H_d(z) = H(s)\Big|_{s=\frac{2(z-1)}{T(z+1)}} = \frac{T(z+1)}{2z-2+2T}$$

Chú ý rằng hàm truyền thu được là hữu tỷ đối với z.

Một phương pháp khác của hàm truyền ánh xạ giữa miền liên tục và miền rời rạc là ánh xạ "đáp ứng phù hợp".

Đáp ứng phù hợp: Giả sử $x(t)$ là đầu vào của một hệ thống $H(s)$ với kết quả đầu ra $y(t)$. Đặt $x[n]$ và $y[n]$ là sản phẩm lấy mẫu của $x(t)$ và $y(t)$. Sau đó, $H_d(z)$ là tỷ số giữa các biến đổi z của $x[n]$ và $y[n]$. Các đáp ứng phù hợp thông thường nhất là đáp ứng bước nhảy trong đó $x(t)$ là hàm bước nhảy, $x(t) = 1$ với $t > 0$ và $x(t) = 0$ với $t < 0$. Biểu thức $H_d(z)$ được cho bởi công thức sau:

$$H_d(z) = (1-z^{-1})Z\left[\frac{H(s)}{s}\right]$$

trong đó $Z[H(s)/s]$ biểu diễn biến đổi z đáp ứng bước nhảy của hệ liên tục. Dạng cho hệ bậc nhất tổng quát như sau:

$$H(s) = \frac{a}{s+a} \Leftrightarrow H_d(z) = \frac{(1-e^{-aT})z^{-1}}{1-e^{-aT}z^{-1}}$$

Phương pháp đáp ứng đầu ra (đặc biệt là đầu vào bước nhảy) thường được sử dụng để ánh xạ hệ liên tục sang hệ rời rạc khi thiết kế một bộ điều khiển số trong miền rời rạc. Do hầu hết các bộ điều khiển số được thực hiện bằng cách sử dụng giữ mẫu bậc không ZOH ở đầu ra của bộ điều khiển nên tín hiệu điều khiển là tín hiệu bước nhảy giống như là tổng các tín hiệu bước nhảy bị trễ. Vì vậy, phương pháp đáp ứng đầu ra bước nhảy là cách chính xác nhất để ánh xạ một hệ có giữ mẫu bậc không ZOH ở đầu vào.

Ánh xạ trong miền tần số

Biến đổi Fourier liên tục có thể được liên hệ với DTFT thông qua biểu thức:

$$X(\omega) = TX(\Omega)\Big|_{\Omega=\omega T} \quad \text{với} \quad -\pi \le \Omega \le \pi$$

ở đây $X(\omega)$ được định nghĩa trong phương trình (29.1) và biểu diễn biến đổi Fourier liên tục của $x(t)$, trong khi $X(\Omega)$ được định nghĩa trong (29.2) và biểu diễn DTFT của tín hiệu được lấy mẫu $x[n]$. Ánh xạ này rất hữu dụng trong việc tính toán biến đổi Fourier của dữ liệu đo. Nói một cách cụ thể, giả sử một tín hiệu liên tục được đo bằng cách lấy mẫu từ ADC và lưu trữ nó như là một tín hiệu rời rạc. Nếu tín hiệu $x(t)$ là giới hạn trong một khoảng thời gian, thì DTFT của $x[n]$ được tính tại các điểm rời rạc với tần số bằng việc sử dụng DFT, X_k, cho bởi phương trình (29.4). Sử dụng

các quan hệ $\omega = \Omega / T$, $\Omega = 2\pi k / N$, và $X_k = X(\Omega)|_{\Omega=2\pi k/N}$, trong đó N là chiều dài của chuỗi $x[n]$ và T là chu kỳ lấy mẫu, được cho bởi quan hệ sau:

$$X(\omega)|_{\omega=2\pi k/NT} = TX_k, \qquad 0 \le \omega \le \omega_s/2, \quad 0 \le k \le (N-1)/2$$

trong đó $\omega_s = 2\pi/T$ là tần số lấy mẫu có đơn vị rad/s. Để tăng độ chính xác có thể giảm chu kỳ lấy mẫu T, và tăng kết quả trên đồ thị bằng cách tăng NT.

Nếu tín hiệu $x(t)$ không bị giới hạn trong một khoảng thời gian, nó phải được cắt bỏ nhằm sử dụng phương pháp số để tính biến đổi Fourier liên tục. Như đã trình bày trong phần "Tín hiệu rời rạc", nếu tín hiệu được lấy mẫu giảm dần về 0, thì chọn số điểm được lấy mẫu N đủ lớn sao cho $x(t)$ đủ nhỏ và có thể bỏ qua kể từ giá trị đó. Nếu tín hiệu lấy mẫu $x[n]$ là tuần hoàn, chọn chu kỳ lấy mẫu T và số điểm N sao cho tín hiệu $x[n]$ bằng một số nguyên lần chu kỳ. Ví dụ, xét tín hiệu $x(t) = \cos(\pi t)$. Nếu chu kỳ lấy mẫu được chọn là T = 0.4s, thì tín hiệu được rời rạc hóa là $x[n]=x(nT)=\cos(0.4\pi n)$, đó là tín hiệu giống như tín hiệu đã được phân tích trong mục "tín hiệu rời rạc". Chọn N=5, 10, 15... sẽ cho kết quả chính xác khi tính DFT, trong khi các giá trị khác sẽ cho kết quả rò rỉ. Nếu tín hiệu $x(t)$ có phổ tần tuần hoàn, nhưng bản thân nó không phải là tín hiệu tuần hoàn, thì sử dụng một hàm cửa sổ như đã trình bày trong mục "Tín hiệu rời rạc" để giảm rò rỉ khi tính toán DFT.

Chú ý rằng DFT cũng có thể được sử dụng để xác định các hệ số Fourier của các tín hiệu tuần hoàn. Xét một chuỗi Fourier có dạng:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_s t}$$

Lấy mẫu tín hiệu $x(t)$ thứ nhất đảm bảo là tín hiệu được lấy mẫu đi qua một số nguyên lần chu kỳ. Các hệ số c_k với $k = 0,...(N-1)/2$ có thể được tính bằng $c_k = X_k/N$ trong đó X_k có thể nhận được từ DFT. Các hệ số còn lại được tính từ công thức $c_{-k} = c_k^*$.

Do đáp ứng tần số của một hệ thống là tỷ số của biến đổi Fourier đầu ra trên biến đổi Fourier đầu vào nên ánh xạ giữa một hệ liên tục, $H(\omega)$, và một hệ rời rạc tương ứng, $H_d(\Omega)$, có thể nhận được từ phép ánh xạ trước như sau:

$$H(\omega) = H_d(\Omega)|_{\Omega=\omega T} \quad \text{với} \quad -\pi \le \Omega \le \pi$$

Ánh xạ này rất hữu dụng khi thiết kế các bộ lọc số và các bộ điều khiển số.

29.4 Thiết kế bộ lọc số

Hàm đáp ứng tần số của hệ rời rạc mô tả cách hệ thống xử lý các tín hiệu đầu vào với các tần số khác nhau. Xét một tín hiệu đầu vào $x[n]=A\cos(\Omega_0 n)$ của một hệ thống với đáp ứng tần số $H(\Omega)$ trong đó $0 \le \Omega \le 2\pi$. Đầu ra tương ứng được tính bởi:

$$y[n] = |H(\Omega_0)|\cos(\Omega_0 n + \angle H(\Omega_0))$$

Với các tín hiệu không tuần hoàn, đặc tính lọc của biến đổi Fourier có mối quan hệ:

$$Y(\Omega) = H(\Omega)X(\Omega)$$

Vì vậy, nếu $|H(\Omega)|$ nhỏ trên một dải tần xác định thì các tín hiệu đầu vào với tần số nằm trong dải tần nói trên sẽ bị suy giảm khi chúng đi qua hệ thống.

Thường thì rất tiện lợi khi lọc một tín hiệu liên tục bằng một bộ lọc số như hình 29.4. Bộ biến đổi tương tự - số (ADC) lấy mẫu tín hiệu liên tục tạo ra một chuỗi các tín hiệu rời rạc để xử lý bằng máy tính hoặc bộ vi xử lý tín hiệu. Tín hiệu được lọc có thể được lưu trữ dưới dạng số hoặc

được gửi qua một bộ biến đổi số - tương tự (DAC). Bộ lọc số được thực hiện dưới dạng phần mềm bằng một phương trình đệ quy nhận được từ phương trình sai phân. Xét một bộ lọc số với hàm truyền:

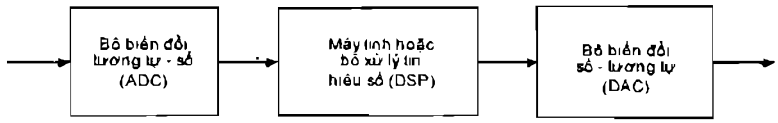
$$H(z)=\frac{b_1z^N+b_2z^{N-1}+\cdots+b_{N+1}}{a_1z^N+a_2z^{N-1}+\cdots+a_{N+1}}=\frac{b_1+b_2z^{-1}+\cdots+b_{N+1}z^{-N}}{a_1+a_2z^{-1}+\cdots+a_{N+1}z^{-N}}$$

Công thức đệ quy dùng để tính giá trị hiện thời của đầu ra $y[n]$ được cho bởi phương trình sai phân sau:

$$y[n]=\frac{1}{a_1}(b_1x[n]+b_2x[n-1]+\cdots+b_{N+1}x[n-N]-a_2y[n-1]-\cdots-a_{N+1}y[n-N]) \qquad (29.5)$$

Chú ý rằng các giá trị trước của y và x phải được giữ lại để sử dụng khi tính đệ quy.

Bây giờ xét đáp ứng xung của bộ lọc số, với $y[n]$ được tính từ một đầu vào $x[n]$ là một xung (nghĩa là $\delta[n]$=1 khi n=0 và còn lại $\delta[n]$=0). Phép tính đệ quy trình bày ở trên dẫn đến một đáp ứng của $y[n]$ là vô hạn (nghĩa là không có giá trị nào của M làm cho $y[n]$=0 với mọi $n>M$). Bộ lọc kiểu này được gọi là *bộ lọc đáp ứng xung vô hạn* (infinite impulse response – IIR).



HÌNH 29.4 Cấu hình của phần cứng xử lý tín hiệu

Bây giờ xét trường hợp các hệ số khuếch đại của bộ lọc $a_m = 0$ với $m > 1$. Biểu thức kết quả của $y[n]$ từ phương trình (29.5) không cần phải tính đệ quy nữa khi nó chỉ phụ thuộc vào giá trị hiện tại và các giá trị trước của x, và không phụ thuộc vào giá trị trước đó của y. Kết quả là đáp ứng xung có thể có trong khoảng N. Bộ lọc kiểu này được gọi là *bộ lọc đáp ứng xung hữu hạn* (FIR - finite impulse response filter). Các bộ lọc FIR đôi khi được chuộng hơn các bộ lọc IIR vì chúng có pha tuyến tính trong đáp ứng tần số. Pha tuyến tính nghĩa là góc của đáp ứng tần số là $-\theta\Omega$ khi θ là hằng số. Đáp ứng này sẽ là một khoảng trễ trong miền thời gian. Phương pháp thiết kế cho cả hai kiểu bộ lọc được mô tả trong hai mục tiếp theo.

Thiết kế bộ lọc IIR

Hai phương pháp để thiết kế các bộ lọc IIR được gọi là *mô phỏng tương tự* (thiết kế gián tiếp) và *thiết kế trực tiếp*. Thiết kế gián tiếp được thực hiện bằng cách thiết kế một bộ lọc tương tự trước và sau đó sử dụng một ánh xạ đã được mô tả trong mục "Ánh xạ từ miền s sang miền z" để chuyển đổi nó thành một bộ lọc số. Phương pháp này có các ưu điểm là sử dụng được nhiều các công nghệ thiết kế cho các bộ lọc tương tự dùng cho thiết kế bộ lọc số. Phương pháp thiết kế trực tiếp nói chung bao gồm các kỹ thuật số, và nó thường được ưu tiên hơn thiết kế gián tiếp khi chu kỳ lấy mẫu không quá nhỏ. Thiết kế trực tiếp không nằm trong phạm vi của cuốn sách này, bạn đọc có thể tham khảo [2] để biết chi tiết hơn.

Thiết kết bộ lọc tương tự bắt đầu bằng việc chọn dải thông, mẫu của bộ lọc và bậc của bộ lọc. Các thông số kỹ thuật khác được chọn dựa trên số lượng gợn sóng cho phép trong dải thông hoặc dải chặn. Hai mẫu thiết kế tương tự điển hình là bộ lọc Butterworth và bộ lọc Chebyshev.

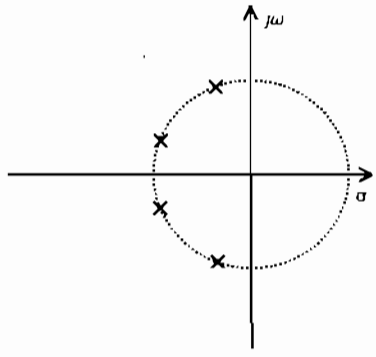
Bộ lọc Butterworth: Bộ lọc Butterworth không có điểm không và có các điểm cực nằm trên nửa đường tròn bên trái của mặt phẳng s. Khoảng cách từ các điểm cực đến gốc là tần số đài thông và được biểu thị bằng ω_h. Góc của các điểm cực được xác định bằng cách đặt chúng cách đều nhau với số lượng điểm cực gấp đôi quanh một đường tròn khép kín với bán kính ω_h, sau đó chỉ giữ lại các điểm cực nằm bên trái mặt phẳng, như hình 29.5. Bộ lọc bậc N được cho bởi công thức:

$$H(s) = \frac{\omega_h^N}{\prod_k (s - \omega_h p_k)}$$

ở đây:

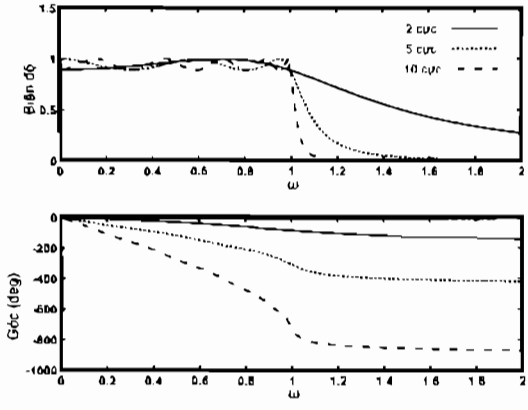
$$p_k = e^{j k\pi / N}, \text{ với } k = \frac{N+1}{2} \text{ đến } \frac{3N-1}{2} \text{ khi N lẻ}$$

$$p_k = e^{j(k-0.5)\pi / N}, \text{ với } k = \frac{N}{2} \text{ đến } \frac{3N-2}{2} \text{ khi N chẵn}$$



HÌNH 29.5 Phân bố điểm cực của bộ lọc Butterworth bậc 4

Bộ lọc này là thông thấp trong đó biên độ của đáp ứng tần số trong bộ lọc tương đối phẳng và gần giá trị 1 với $\omega < \omega_h$ và giảm ở phía xa tần số dải thông. Bậc của bộ lọc càng lớn thì giảm càng nhanh. So sánh 3 bộ lọc trong hình 29.6. Chú ý rằng quá độ đột ngột xuất hiện ở bộ lọc bậc cao hơn làm cho pha cũng giảm một cách đột ngột. Sự định pha trở lên quan trọng trong hệ thống đo thời gian thực như yêu cầu của các bộ điều khiển phản hồi.



HÌNH 29.6 So sánh các bộ lọc Butterworth tương tự

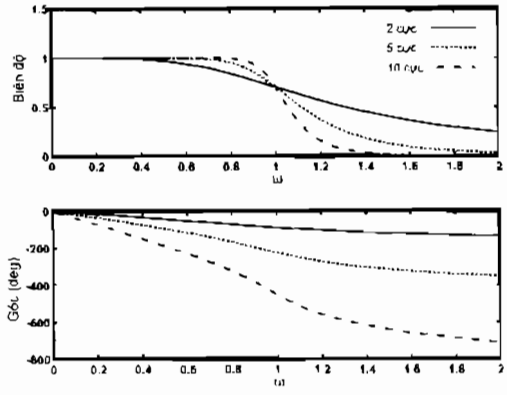
Bộ lọc Chebyshev: Không đơn điệu như bộ lọc Butterworth, bộ lọc Chebyshev cho phép một vài gợn sóng trong đồ thị biên độ cho dải thông hoặc dải chặn. Kiểu 1 của bộ lọc Cheyshev cho phép gợn sóng ở dải thông trong khi bộ lọc Chebyshev kiểu 2 cho phép gợn ở dải chặn. Việc cho phép một gợn sóng dẫn đến bộ lọc Chebyshev có quá độ đột ngột gần băng thông hơn bộ lọc Butterworth cùng bậc. Trong thiết kế Chebyshev, tần số cắt ω_c thường được xác định ngược với băng thông. Tần số cắt là tần số tại đó biên độ của bộ lọc suy giảm xuống một tỷ số đặt trước của giá trị DC. Khi tỷ số này là 0.707, tần số cắt là băng thông. Thông thường, trong thiết kế, tỷ số này được chọn để phù hợp với tổng số gợn sóng được phép trong dải thông. Một bộ lọc thông thấp kiểu 1 được định nghĩa bằng mối quan hệ:

$$|H(\omega)| = \frac{1}{\sqrt{1+\varepsilon^2 C_N^2(\omega/\omega_c)}}$$

và

$$C_N(x) = 2xC_{N-1}(x) - C_{N-2}(x)$$

Biểu thức $C_N(x)$ được gọi là đa thức Chebyshev bậc N, và nó được tính đệ quy bắt đầu với $C_0(x) = 1$ và $C_1(x) = x$. Giá trị $\varepsilon > 0$ xác định tổng số gợn sóng cho phép trong dải thông. Nói một cách cụ thể, gợn sóng nằm giữa giá trị 1 và $1/\sqrt{1+\varepsilon^2}$. Xét ví dụ bộ lọc Chebyshev kiểu 1 như trong hình 29.7. Các bộ lọc này đã được thiết kế để có độ gợn là 1dB trong dải thông ($\varepsilon = 0.51$). Chú ý rằng $\varepsilon = 1$ với độ gợn 3dB.



HÌNH 29.7 So sánh các bộ lọc Chebshev tương tự kiểu I

Như đã đề cập ở trên các bộ lọc mẫu này là thông thấp. Để thiết kế một kiểu bộ lọc khác (thông cao, thông một dải...), đầu tiên là thiết kế bộ lọc thông thấp, $H(s)$, với một tần số cắt ω_c (thông thường được chọn là 1). Sau đó sử dụng một phép biến đổi tần số để chuyển đổi bộ lọc này sang kiểu mong muốn. Các phép biến đổi trong miền tần số chuẩn sẽ được trình bày dưới đây.

Từ thông thấp tới thông thấp: Để thu được một bộ lọc thông thấp với tần số cắt ω_1, thay s trong $H(s)$ ban đầu bởi $s\omega_c/\omega_1$.

Từ thông thấp đến thông cao: Để thu được một bộ lọc thông cao với một dải thông từ ω_1 đến ∞, thay s trong hàm truyền $H(s)$ bởi $\omega_1\omega_c/s$.

Từ thông thấp tới thông dải: Để thu được một bộ lọc với dải thông từ ω_1 đến ω_2, thay s trong hàm truyền $H(s)$ bởi:

$$\frac{s^2 + \omega_2\omega_1}{s(\omega_2 - \omega_1)}$$

Từ thông thấp đến chặn dải: Để thu được một bộ lọc chặn dải với dải chặn từ ω_1 đến ω_2, thay s trong hàm truyền $H(s)$ bởi:

$$\frac{s(\omega_2 - \omega_1)}{s^2 + \omega_2\omega_1}$$

Thiết kết bộ lọc FIR

Một cách để có được bộ lọc FIR là chặt đáp ứng xung của một bộ lọc IIR lý tưởng. Ví dụ, một bộ lọc thông thấp IIR lý tưởng có đáp ứng tần số:

$$H(\Omega) = \begin{cases} A, & -\Omega_c \le \Omega \le \Omega_c \\ 0, & \text{trường hợp còn lại} \end{cases}$$

ở đó A là một hằng số và Ω_c là tần số cắt. Đáp ứng xung của bộ lọc này nhận được bằng cách nghịch đảo phương trình DTFT (29.3):

$$h[n] = \frac{A\Omega_c}{\pi} e^{jm\Omega_c/2} \operatorname{sinc}\left(\frac{\Omega_c n}{\pi}\right)$$

Chú ý rằng phương trình này có khoảng giới hạn trong cả $n < 0$ và $n > 0$. Việc tạo ra một bộ lọc FIR sẽ đòi hỏi việc chặt đáp ứng xung từ $n < -N$ và $n > N$. Tuy nhiên, bộ lọc IIR ban đầu và bộ lọc FIR bị chặt đều phi nhân quả, điều đó nghĩa là đáp ứng xung là khác 0 với $n < 0$. Các bộ lọc phi nhân quả cần các giá trị "tương lai" của đầu vào để tính giá trị đầu ra hiện tại, vì vậy nó không thể được thực hiện trong thời gian thực. Vì lý do này, thiết kế IIR chủ yếu sử dụng các bộ lọc không lý tưởng (thường dựa trên các mẫu tương tự) chúng xấp xỉ đáp ứng tần số lý tưởng. Khi lọc một tín hiệu off-line được lưu trữ, tính nhân quả là không cần thiết vì tất cả các giá trị của tín hiệu đã có sẵn (gồm cả các giá trị "tương lai").

Nhằm thực hiện khả năng thời gian thực của một bộ lọc FIR được tạo ra bằng cách chặt một bộ lọc IIR lý tưởng, bộ lọc phải được trễ sao cho tất cả thông tin có nghĩa của đáp ứng xung xuất hiện khi $n > 0$. Sự trễ trong miền thời gian tương đương với sự chậm pha phi tuyến trong miền tần số.

Vì vậy, một bộ lọc FIR được thiết kế trước tiên bằng cách chọn một bộ lọc IIR lý tưởng (thông thấp, thông cao...), sau đó thực hiện nghịch đảo DTFT để tìm đáp ứng xung, chặt đáp ứng xung, và cuối cùng làm trễ nó trong miền thời gian. Một một phương pháp tương đương và hoàn hảo hơn là sắp xếp lại các bước được mô tả ở trên. Thứ nhất, cộng thêm một độ lệch pha trong đáp ứng tần số của bộ lọc IIR lý tưởng. Việc này được thực hiện bằng cách nhân đáp ứng tần số với $e^{j(N-1)\cdot 2}$. Sau đó thực hiện nghịch đảo DTFT và chặt nó với $n < 0$ và $n > N - 1$. Kết quả là một bộ lọc FIR nhân quả với bậc N.

Sau đây là bộ lọc FIR tổng quát bậc N được tạo bằng phương pháp mô tả ở trên. Chọn $m = (N-1)/2$.

Bộ lọc FIR thông thấp với tần số cắt Ω_c:

$$h[n] = \begin{cases} \dfrac{\Omega_c}{\pi}, n = 0 \\ \dfrac{\Omega_c}{\pi} \operatorname{sinc}\left[\dfrac{\Omega_c (n-m)}{\pi}\right], 0 < n \le N-1 \\ 0, \text{trường hợp còn lại} \end{cases}$$

Trong đó $\operatorname{sinc}(x) = \sin(\pi x)/\pi x$.

Bộ lọc FIR thông cao với dải thông từ Ω_1:

$$h[n] = \begin{cases} 1 - \dfrac{\Omega_1}{\pi}, \text{với} \quad n = 0 \\ -\dfrac{\Omega_1}{\pi} \operatorname{sinc}\left[\dfrac{\Omega_1 (n-m)}{\pi}\right] 0, \text{với} \quad 0 < n \le N-1 \\ 0, \text{trường hợp còn lại} \end{cases}$$

Bộ lọc FIR thông một dải với dải thông Ω_1 đến Ω_2:

$$h[n] = \begin{cases} \dfrac{\Omega_2 - \Omega_1}{\pi}, & với \quad N = 0 \\ \dfrac{\Omega_2}{\pi}\sin c[\Omega_2(n-m)/\pi] - \dfrac{\Omega_1}{\pi}\sin[\Omega_1(n-m/\pi], & vii \quad 0 < n \le N-1 \\ 0, \text{ trường hợp còn lại} \end{cases}$$

Để thực hiện các bộ lọc này, các hệ số trong phương trình (29.5) được đặt để $b_m = h$[m-1], $a_1 = 1$, và $a_m = 0$ với $m > 1$.

Các bộ lọc FIR được thiết kế sử dụng phương pháp này có đáp ứng tần số quá độ đột ngột hơn giữa dải thông và dải chặn (bậc càng lớn thì quá độ càng đột ngột), nhưng chúng giữ cho đường dốc thành một đường gợn giữa dải thông và dải chặn. Đường gợn này nhận được từ việc chặt đáp ứng xung của bộ lọc IIR. Việc cắt dần dần bằng một hàm cửa sổ đã làm cho đáp ứng tần số trở lên trơn tru. Hàm cửa sổ đã được trình bày trong phần "Ánh xạ từ miền s sang miền z" dùng để thu thập dữ liệu cũng được sử dụng trong thiết kế bộ lọc FIR, khi đó các bộ lọc được thay đổi thành h[n]w[n]. Thiết kế FIR sử dụng các cửa sổ khác được trình bày chi tiết hơn trong [1, 2].

Thiết kế bộ lọc số với trợ giúp của máy tính

Matlab là một phần mềm phổ biến để phân tích xử lý tín hiệu và thiết kế. Toolbox xử lý tín hiệu chứa nhiều lệnh thiết kế và mô phỏng các bộ lọc số. Ví dụ, các lệnh butter và cheby1 thiết kế tự động một bộ lọc tương tự mẫu cho một bộ lọc IIR và sau đó sử dụng phép biến đổi song tuyến để ánh xạ tới các bộ lọc trong miền rời rạc. Các bộ lọc thông thấp, thông cao, chặn dải và thông dải có thể được thiết kế bằng các lệnh này miễn là các tần số cắt số, chuẩn hóa bởi π, được xác định. Để thiết kế một bộ lọc thông thấp số dựa trên bộ lọc Butterworth tương tự với tần số cắt $w1$, sử dụng lệnh [b, a]=butter(N, $w1$*T/pi). Trong đó N là số điểm cực, T là chu kỳ lấy mẫu, và $w1$*T là tần số cắt số. Lệnh này đặt các hệ số cho bộ lọc, được định nghĩa trong phương trình (29.5), trong các vectơ b và a theo thứ tự tăng dần. Để thiết kế một bộ lọc thông cao số với tần số cắt tương tự $w1$, sử dụng lệnh [b, a]=butter(N, $w1$*T/pi,'hight'). Để thiết kế một bộ lọc thông dải số với dải thông tương tự từ $w1$ đến $w2$, định nghĩa $w = [w_1, w_2]$ và sử dụng lệnh [b, a]=butter(N, w*T/pi). Để thiết kế bộ lọc chặn dải số với dải chặn từ $w1$ tới $w2$, định nghĩa w=[$w1$,$w2$] và sử dụng lệnh [b, a]=butter(N, w*T/pi,'stop'). Thiết kế một bộ lọc Chebyshev kiểu I bậc N được thực hiện giống như các phương pháp cho butter trừ việc "butter" được thay bằng "cheby1".

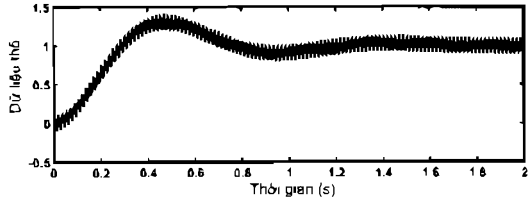
Toolbox xử lý tín hiệu cũng cung cấp các lệnh để thiết kế các bộ lọc FIR. Để có một bộ lọc FIR thông thấp độ dài N và tần số cắt tương tự $w1$, sử dụng lệnh h = *fir*1(N-1, w1*T/pi). Vectơ thu được h chứa đáp ứng xung của FIR trong đó $h(1)$ là giá trị của $h[0]$. Các giá trị trong vectơ h cũng bằng các hệ số b trong phương trình (29.5) với bậc tăng dần. (lưu ý, $a_1 = 1$ và $a_m = 0$ với $m > 1$). Một bộ lọc FIR thông cao độ dài N với tần số cắt analog $w1$ được thiết kế bằng cách sử dụng lệnh h = *fir*1($N-1$, $w1*T/pi$,'*high*'). Một bộ lọc FIR thông dải với dải thông từ $w1$ tới $w2$ được thiết kế bằng lệnh h = *fir*1($N-1$, $w1*T/pi$) trong đó $w = [w1, w2]$. Một bộ lọc FIR chặn dải với dải chặn từ $w1$ đến $w2$ được thiết kế bằng lệnh h = *fir*1($N-1$, $w1*T/pi$,'*stop*') trong đó $w = [w1, w2]$. Lệnh fir1 mặc định sử dụng cửa sổ Hamming. Để sử dụng cửa sổ khác đưa thêm lựa chọn "hanning" hoặc "boxcar" (đó là những cửa sổ hình chữ nhật) với các tham số; ví dụ, h = *fir*1($N-1$, $w1*T/pi$,'*high*',*boxcar*(N)) tạo bộ lọc FIR thông cao với tần số cắt tương tự $w1$ sử dụng một cửa sổ hình chữ nhật.

Lệnh bộ lọc trong Matlab được sử dụng để tính đầu ra của bộ lọc số được cho bởi dãy đầu vào. Một ví dụ về trường hợp này là $y = filter(b, a, x)$ trong đó b và a là các hệ số của bộ lọc và x là dãy đầu vào.

Các ví dụ về bộ lọc

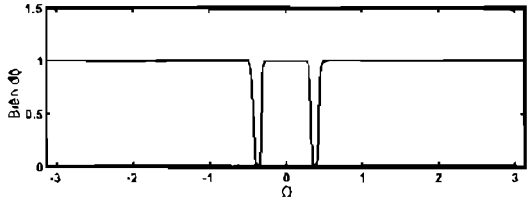
Chúng ta thường gặp nhiễu với tần số 60 Hz trong các giá trị đo của hệ cơ điện do điện áp đường chuẩn. (Lưu ý, nhiễu ở Châu Âu thường gặp ở tần số 50 Hz). Với mục đích minh họa, một tín hiệu với tần số 60 Hz được chồng vào tín hiệu có tần số thấp hơn như hình 29.8. Để giảm bớt sự ảnh hưởng không tốt của nhiễu 60 Hz, có thể sử dụng bộ lọc chặn dải. Thông thường, hầu hết các hệ thống có bộ lọc chặn dải đó được thiết kế quanh tần số 60Hz để tránh kiểu đáp ứng như hình 29.8. Các lệnh Matlab sau đây có thể được dùng để thiết kế bộ lọc số chặn một dải Butterworth bậc 8 mà các tần số ngắt là 50 và 70Hz. Vì vậy, bộ lọc này loại bỏ được nhiễu 60Hz.

```
T = 0.001; % Chu kỳ lấy mẫu
n = 4; % Một nửa bậc của bộ lọc
low_freq = 50 * (2*pi); %Tín hiệu chặn giữa 50 và 70 Hz
high_freq = 70 * (2*pi);
w1 = low_freq*(T/pi); % Tần số gẫy
w2 = high_freq*(T/pi);
w = [w1 w2];
b,a] = butter(n,w,'stop'); % Các hệ số của bộ lọc
W = -pi:pi/200:pi; % Định nghĩa một vecto tần số rời rạc
H = freqz(b,a,W); % Tính toán đáp ứng tần số
```



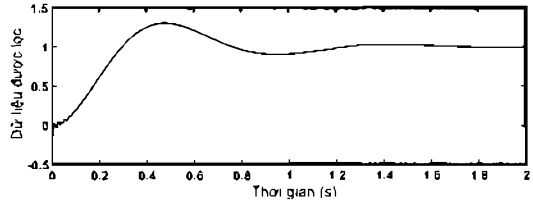
HÌNH 29.8 Giá trị đo bị lẫn nhiễu tần số 60 Hz

Hình 29.9 minh họa biên độ của đáp ứng tần số đối với bộ lọc IIR thu được. Chú ý rằng biến tần số được vẽ trong dải $[-\pi,\pi]$ ở đó tần số DC tương ứng với $\Omega = 0$ và tần số cao nhất cho phép là $\Omega = \pi$. Trong ví dụ này, các tần số gẫy tương ứng với $\Omega_1 = 50(2\pi)T = 0.314$ và $\Omega_2 = 70(2\pi)T = 0.44$ Hình 29.10 là kết quả áp dụng của bộ lọc này với tín hiệu nhiễu. Với tất cả các ứng dụng thực tiễn, nhiễu tần số 60Hz được loại bỏ hoàn toàn.



HÌNH 29.9 Bộ lọc chặn dải

Như thấy trên hình 29.10, chỉ có nhiễu trong khoảng thời gian quá độ 100ms lúc đầu của đáp ứng bước nhảy. Đây là sự kết hợp của bộ lọc Butterworth bậc bốn và quá độ ban đầu của hệ thống với tần số 60 Hz. Nên chú ý rằng tần số lấy mẫu của 1000kHz đủ nhanh để thực hiện chính xác tín hiệu 60Hz. Nếu sử dụng tần số lấy mẫu nhỏ hơn 120Hz, thì tín hiệu tần số 60Hz sẽ có hiện tượng răng cưa, và dù lọc đến mức nào vẫn không loại bỏ được các hiệu ứng của nhiễu tần số 60Hz.



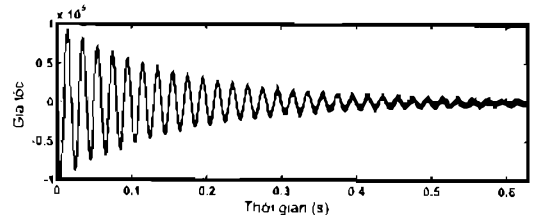
HÌNH 29.10 Giá trị đo đã được lọc

Một ứng dụng khác của bộ lọc số trong cơ điện tử được sử dụng khi thực hiện ước lượng độ dịch chuyển từ việc đo gia tốc. Một phương pháp đơn giản để tính độ dịch chuyển là tích phân 2 lần gia tốc. Trong miền s, bộ tích phân 2 lần này tương đương với việc nhân thêm $1/s^2$. Sử dụng biến đổi song tuyến để chuyển $1/s^2$ sang miền z sinh ra hàm truyền sau:

$$H(z) = H(s)\Big|_{s=\frac{2(z-1)}{T(z+1)}} = \frac{1}{s^2}\Big|_{s=\frac{2(z-1)}{T(z+1)}} = \left(\frac{T^2}{4}\right)\left(\frac{z^2+2z+1}{z^2-2z+1}\right) = \left(\frac{T^2}{4}\right)\left(\frac{1+2z^{-1}+z^{-2}}{1-2z^{-1}+z^{-2}}\right)$$

Phương trình sai phân tương đương dùng để tính độ dịch chuyển $y[n]$ từ phương trình $y_{dd}[n]$ là $4(y[n]-2y[n-1]+y[n-2])=T^2(y_{dd}[n]+2y_{dd}[n-1]+y_{dd}[n-2])$. Tuy nhiên, các gia tốc kế thường không đáp ứng tốt tại các tần số thấp. Trong thực tế, người ta thường chèn một độ lệch vào dữ liệu làm giảm hiện tượng trôi trong tính toán dịch chuyển. Chúng cũng rất nhạy cảm với các dao động ngẫu nhiên. Một phương pháp khác là xử lý dữ liệu gia tốc bằng bộ lọc thông dải trước khi sử dụng phương trình sai phân để tích phân nó dưới dạng số. Đoạn dải thông phải chứa các tần số riêng trong hệ thống.

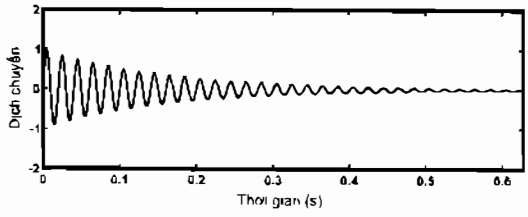
Xét ví dụ, dữ liệu gia tốc như hình 29.11, với một vài tín hiệu nhiễu ngẫu nhiên. Tín hiệu này được lấy mẫu với tốc độ 6400Hz, ở đó tần số riêng của hệ thống là 50Hz. Hình 29.12 minh họa độ dịch chuyển thực, còn hình 29.13 minh họa độ dịch chuyển ước lượng được tính bằng cách tích phân dữ liệu gia tốc dưới dạng số, dùng phương trình vi phân trên. Ước lượng này rất nghèo nàn. Thay vào đó, một bộ lọc thông dải Chebyshev kiểu 1 bậc 8 tương tự với dải thông 25 – 500Hz được thiết kế và sau đó được rời rạc hóa sử dụng biến đổi song tuyến. Dữ liệu gia tốc được xử lý bằng bộ lọc này trước, sau đó dữ liệu lọc được tích phân dưới dạng số với kết quả như hình 29.14.



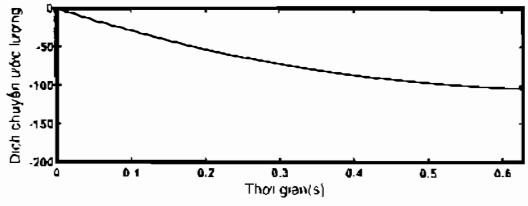
HÌNH 29.11 Đo gia tốc

Chú ý rằng ước lượng này tốt hơn nhiều so với kết quả không sử dụng bộ lọc thông dải. Một bộ lọc thông dải FIR bậc 500 cũng được thiết kế trong ví dụ này với băng thông 25 – 500

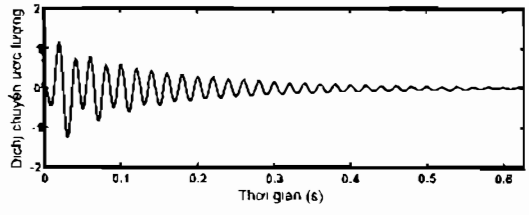
Hz. Sau khi dữ liệu qua bộ lọc FIR, nó được tích phân dưới dạng số dẫn đến độ dịch chuyển ước lượng như hình 29.15. Do có đặc tính pha tuyến tính của bộ lọc FIR, nó ít méo hơn bộ lọc IIR, nhưng nó có độ trễ lớn hơn. Bậc càng lớn, độ chính xác của kết quả càng cao, do càng ít thông tin có nghĩa bị mất trong quá trình chặt đáp ứng xung của bộ lọc thông dải IIR lý tưởng, nhưng độ trễ lớn hơn. Biên độ của bộ lọc IIR và các bộ lọc thông dải FIR như hình 29.16.



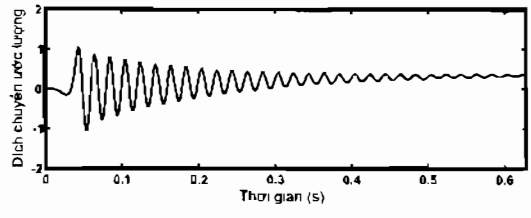
HÌNH 29.12 Độ dịch chuyển thực tế



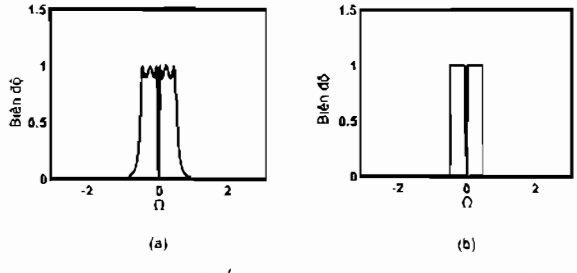
HÌNH 29.13 Độ dịch chuyển ước lượng không sử dụng bộ lọc trước



HÌNH 29.14 Độ dịch chuyển được ước lượng với bộ lọc IIR



HÌNH 29.15 Độ dịch chuyển được ước lượng với bộ lọc FIR



HÌNH 29.16 Các bộ lọc thông dải số: (a) bộ lọc IIR Chebyshev và (b) bộ lọc FIR

Hai điều được đề cập trong ví dụ này là:

1. Việc tính toán không lọc vô cùng nhạy với sai lệch trong dữ liệu (như đòi hỏi tích phân 2 lần). Vì vậy, sai lệch được loại bỏ khỏi gia tốc trước khi xử lý. Cả hai bộ lọc đã loại bỏ sai lệch một cách tương đối, do đó kết quả hầu như không thay đổi nếu có sự xuất hiện của sai lệch.
2. Bộ lọc FIR có một số hiện tượng trôi. Có lẽ, nguyên nhân của trôi là bộ lọc đó dường như vẫn có một vài khó khăn với vùng dải chặn nhỏ gần gốc. Tăng vùng dải chặn làm giảm trôi. Điều này có thể thực hiện được bằng cách giảm tần số lấy mẫu hoặc bằng cách tăng tần số dải thông. Cả hai biện pháp này đều giảm trôi nhưng tăng các sai số khác trong tín hiệu. Tăng chiều dài của bộ lọc giảm sai số trôi mà không xuất hiện các sai số khác.

Các lệnh Matlab sử dụng để thiết kế các bộ lọc và các kết quả là:

```
[num, den]=c2dm(1, [1 0 0], T, "tustin"); % Số hóa bậc 2
y1=filter(num, den, ydd); % 2 bộ tích phân của ydd

Wbreak=[2*pi*25*T, 2*pi*500*T]; % Các tần số gẫy
[b,a]=heby1(4,1,Wbreak); % Thiết kế bộ lọc IIR với 1dB độ gợn

W =pi:pi/200:pi; % định nghĩa dải tần số rời rạc cho đồ thị
H =reqz(b,a,W); % kết quả đáp ứng tần số
plot(W,abs(H)); % đồ thị biên của đáp ứng tần số

yddfilt=filter(b,a,ydd); % Tính toán đầu ra của bộ lọc ydd
y2=ilter(num,den,yddfilt); % 2 bộ tích phân của yddfilt

hfir =firl(500,Wbreak); % thiết kế bộ lọc FIR với bậc 500
yddfilt=_filter(hfir,1,ydd);% Tính toán đầu ra cho bộ lọc FIR
y3 =filter(num,den,yddfilt); % 2 bộ tích phân của yddfilt
```

29.5 Thiết kế điều khiển số

Cũng như trong trường hợp thiết kế bộ lọc số, có hai phương pháp chủ yếu để thiết kế một bộ điều khiển số: *phương pháp gián tiếp* được dựa trên rời rạc hóa một thiết kế tương tự, và *phương pháp trực tiếp* dựa trên việc rời rạc hóa hệ thống (thường sử dụng phương pháp thực hiện đáp ứng bước) và sau đó thiết kế bộ điều khiển trực tiếp trên miền rời rạc. Hầu hết các kỹ sư học điều khiển liên tục cổ điển, và học nhiều về thiết kế bộ điều khiển liên tục hơn là bộ điều khiển rời rạc hoặc thiết kế điều khiển số. Rất may, các công cụ thiết kế điều khiển liên tục thường cũng có thể được sử dụng để thiết kế các hệ điều khiển số. Để tạo các bộ điều khiển được thiết kế trong miền liên tục, cần sử dụng một ánh xạ từ miền s sang miền z. Các ánh xạ đã trình bày trong chương này có thể sử dụng cho các loại bộ điều khiển. Cách xác định bằng ánh xạ là cách tốt nhất để tìm từng bộ điều khiển hoặc bộ lọc. Mặc dù việc xắp xỉ song tuyến phức tạp hơn xắp xỉ thuận hoặc xắp xỉ ngược, nhưng nó vẫn được sử dụng cho hầu hết các hệ cơ điện tử. Điều này là do trong thực tế hầu hết các bộ điều khiển hiện đại đều có đủ sức tính toán để quản lý độ phức tạp lớn ở các dải thông yêu cầu của hệ cơ điện tử.

Một ví dụ về phương pháp thiết kế gián tiếp, xét một bộ điều khiển tỷ lệ vi phân PD được sử dụng để làm tăng chất lượng của hệ thống. các hệ số vi phân và tỷ lệ cho bộ điều khiển lần lượt là K_d và K_p. Bộ điều khiển PD, $K(s)$, được cho bởi công thức sau:

$$K(s) = K_d s + K_p \tag{29.6}$$

Phương trình (29.6) được chuyển sang miền số sử dụng bất kỳ ánh xạ nào từ miền s sang miền z đã trình bày trong phần trước của chương này. Một ví dụ biến đổi song tuyến sử dụng bộ điều khiển số tổng quát, $K(z)$.

$$K(z) = K(s)\Big|_{s=\frac{2(z-1)}{T(z+1)}} = \frac{(2K_d + TK_p)z + (TK_p - 2K_d)}{Tz + T} \tag{29.7}$$

Bên cạnh các hệ số khuếch đại điều khiển, chỉ có 1 thừa số cần trong phương trình (29.7) là T, thời gian lấy mẫu. Như đã trình bày ở trên, thời gian lấy mẫu phải ít nhất phải lớn gấp 5-10 lần hằng số thời gian hệ thống nhanh nhất. Tuy nhiên, thời gian lấy mẫu thì thường được chọn nhanh hơn vài trăm lần hằng số thời gian nhanh nhất. Một chiến lược thay đổi cho một hệ thống phản hồi là chọn tốc độ lấy mẫu tối thiểu là 20 lần đài thông hệ kín lý tưởng. Thời gian lấy mẫu về cơ bản nhanh hơn hệ thống thực để giảm bớt sự khác biệt giữa bộ điều khiển được thiết kế trong miền liên tục và thực hiện trong miền rời rạc. Nên chú ý rằng tần số lấy mẫu càng lớn, các hệ số khuếch đại của bộ điều khiển càng nhỏ. Ví dụ, trong phương trình (29.7), khi thời gian lấy mẫu càng nhỏ, T càng nhỏ yêu cầu độ phân giải số tốt hơn cho các hệ số khuếch đại của bộ điều khiển. Nếu T trở lên nhỏ hơn độ phân giải hệ số khuyếch đại số của bộ điều khiển, thì nó có thể thực hiện sai tại giá trị 0 dẫn đến luật điều khiển không chính xác.

Ví dụ về điều khiển số

Xét một động cơ được định vị tốc độ cao với động học động cơ cho bởi phương trình bậc nhất:

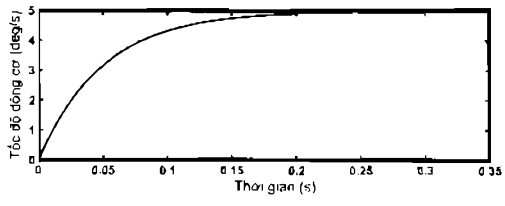
$$G(s) = \frac{\omega(s)}{V_m(s)} = \frac{K_m}{T_m s + 1}$$

trong đó K_m là hằng số khuếch đại của động cơ, T_m là hằng số thời gian của động cơ, $\omega(s)$ là biến đổi Laplace của vận tốc, và $V_m(s)$ là biến đổi Laplace của điện áp vào động cơ. Để xác định giá trị của T_m và K_m, sử dụng đáp ứng bước nhảy của vận tốc. Hình 29.17 là đáp ứng của động cơ với đầu vào là bước nhảy 1V. Hệ số khuếch đại động cơ, K_m, là giá trị ổn định của tốc độ động cơ và là 5. Kết quả này cũng có thể được xác định sử dụng các lập luận như sau

$$\lim_{t\to\infty}\omega(t) = \lim_{s\to 0} s\omega(s) = \lim_{s\to 0} sG(s)V_m(s)$$
$$= \lim_{s\to 0} sG(s)\frac{1}{s} = \lim_{s\to 0} s\frac{K_m}{T_m s + 1}\frac{1}{s} = K_m$$

Hằng số thời gian, T_m có thể được tính bằng cách xác định vận tốc động cơ với đáp ứng bước nhảy tại $t = T_m$ như sau:

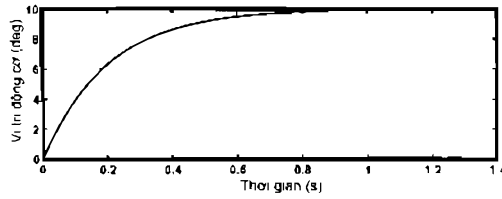
$$\omega(t = T_m) = K_m(1 - e^{-t/T_m}) = K_m(1 - e^{-1}) = 0.632K_m$$



HÌNH 29.17 Đáp ứng bước nhảy của vận tốc động cơ

Thời gian cần để động cơ đạt được 63.2% trạng thái ổn định của đáp ứng bước nhảy chính là hằng số thời gian và bằng 0.05s (theo hình 29.18) vì vậy hàm truyền của động cơ này là:

$$G(s) = \frac{\omega(s)}{V_{in}(s)} = \frac{K_m}{T_m s + 1} = \frac{5}{0.05s + 1} \tag{29.8}$$



HÌNH 29.18 Đáp ứng bước nhảy của vị trí động cơ

Trong ví dụ này, động cơ được sử dụng chế độ điều khiển vị trí. Do vị trí động cơ là tích phân của vận tốc, nên phương trình (29.8) được thêm với một bộ tích phân để tạo ra hàm truyền của đông cơ liên hệ điện áp đầu vào với vị trí đầu ra, $\theta(s)$:

$$G_p(s) = \frac{\theta(s)}{V_m(s)} = \frac{K_m}{s(T_m s + 1)} \tag{29.9}$$

Một bộ điều khiển PD được chọn dùng trong ví dụ này để làm tăng chất lượng của hệ thống. Để đạt được đáp ứng nhanh và không quá tải, hệ số vi phân, K_d, và hệ số tỷ lệ, K_p, được chọn lần lượt là 0.05 và 1, tạo ra luật điều khiển như sau:

$$K(s) = K_d s + K_p = 0.05s + 1 \tag{29.10}$$

Trên danh nghĩa, thiết kế này bỏ qua điểm cực tần số cao của động học động cơ trong phương trình (29.9).

Chu kỳ lấy mẫu 1 ms được chọn trong ví dụ này là nhanh hơn đáng kể so với hằng số thời gian của hệ thống, và nó không phải là giá trị không hợp lý cho các bộ điều khiển hiện đại. Như đã trình bày ở trên, sử dụng tần số trích mẫu 1kHz (1 ms) làm giảm bớt sự khác nhau giữa bộ điều khiển được thiết kế trong miền liên tục và sự thực hiện của nó trong miền rời rạc. Sử dụng phép biến đổi song tuyến như đã trình bày trong mục "ánh xạ từ miền s sang miền z" kết quả một bộ điều khiển số có dạng:

$$K_D(z) = \frac{101z - 99}{z + 1}$$

Trong thực tế, đáp ứng vòng kín của hệ thống sử dụng bộ điều khiển số không dễ phân biệt với hệ thống sử dụng bộ điều khiển tương tự cho bởi phương trình (29.10). Đáp ứng vị trí vòng kín của động cơ với đầu vào 100 như hình 29.18.

Như đã đề cập trong mục "Ví dụ về bộ lọc", tín hiệu nhiễu tần số 60 Hz thường xuất hiện trong đo lường các hệ cơ điện, vì vậy bộ lọc chặn dải thường được sử dụng để làm giảm đi tín hiệu nhiễu. Trong hoạt động của vòng kín, bộ lọc thông dải số được nối tầng với bộ điều khiển PD số.

Tài liệu tham khảo

[1] Kamen, E.W., and Heck, B.S., *Signals and Systems Using the Web and Matlab,* 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 2000.

[2] Britton Rorabaugh, C., *Digital Filter Designer's Handbook: with C++ Algorithms,* 2nd ed., McGraw-Hill, New York, 1997.

30

Thiết kế hệ thống điều khiển bằng tối ưu $\mathcal{H}^2$

30.1 Giới thiệu 30-1
30.2 Cấu trúc thiết kế hệ điều khiển tổng quát 30-2
30.3 Bài toán phản hồi đầu ra $\mathcal{H}^2$ 30-12
30.4 Bài toán phản hồi trạng thái $\mathcal{H}^2$ 30-42
30.5 Bài toán nội xạ đầu ra $\mathcal{H}^2$ 30-44
30.6 Kết luận 30-46

Armando A. Rodriguez
Arizona State University

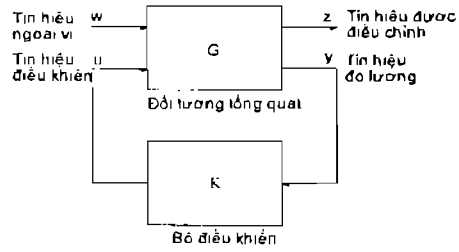
30.1 Giới thiệu

Chương này giới thiệu việc thiết kế hệ thống điều khiển thông qua tối ưu $\mathcal{H}^2$ (toàn phương). Một nguyên tắc thống nhất dựa trên khái niệm đối tượng tổng quát và tối ưu hóa theo trọng số cho phép nhà thiết kế hướng tới bài toán phản hồi trạng thái, ước lượng trạng thái, phản hồi đầu ra động lực và nhiều cấu trúc tổng quát tương tự. Nguyên tắc này cho phép người thiết kế kết hợp dễ dàng các tham số thiết kế và/hoặc các hàm trọng số có thể được sử dụng để tác động tới kết quả tối ưu, thỏa mãn các thông số kỹ thuật thiết kế mong muốn và hệ thống hóa quá trình thiết kế. Các lời giải tối ưu nhận được thông qua các phương trình Riccati nổi tiếng, ví dụ như phương trình Riccati đại số điều khiển (CARE - Control Algebraic Riccati Equation) và phương trình Riccati đại số lọc (FARE - Filter Algebraic Riccati Equation). Trong khi các hàm trọng số động lực làm tăng số chiều của phương trình Riccati thì các lời giải có thể thu được dễ dàng bằng cách sử dụng các phần mềm thiết kế như MATLAB, hộp công cụ điều khiển bền vững, hộp công cụ tổng hợp μ, *v.v.*

Nói tóm lại, tối ưu $\mathcal{H}^2$ tổng quát tất cả các hệ phương pháp thiết kế bộ lọc và điều khiển toàn phương nổi tiếng:

- Phương pháp thiết kế điều chỉnh toàn phương tuyến tính (LQR - Linear Quadratic Regulator) [7, 11],
- Phương pháp thiết kế bộ lọc Bucy-Kalman (KBF - Kalman Bucy Filter) [5, 6],
- Phương pháp thiết kế Gaussian toàn phương tuyến tính (LQG - Linear Quadratic Gaussian) [4, 10, 11].

Tối ưu $\mathcal{H}^2$ có thể được sử dụng để thiết kế một cách hệ thống các luật điều khiển phản hồi trạng thái hệ số hằng, các bộ ước lượng trạng thái, các bộ điều khiển đầu ra động lực, và nhiều bộ điều khiển khác.



HÌNH 30.1 Hệ thống phản hồi tổng quát

30.2 Nguyên tắc cơ bản để thiết kế hệ điều khiển

Trong phần này, chúng ta trình bày một nguyên tắc cơ bản để thiết kế hệ thống điều khiển (và bộ ước lượng). Với mục đích này, chúng ta nghiên cứu hệ thống phản hồi đã được tổng quát hóa như hình 30.1. Trong hình này, *G* biểu thị *đối tượng tổng quát*. *G* chứa mô hình cho đối tượng thực *P* (hệ vật lý) cần điều khiển. Nó cũng có thể chứa các hàm trọng số (tần số phụ thuộc) bổ sung được sử dụng để hướng tới các mục tiêu thiết kế vòng kín. *K* mô tả bộ điều khiển hoặc bộ bù cần thiết kế.

Ý tưởng chính: thiết kế thông qua tối ưu

Ý tưởng chính ở đây là các vấn đề quan trọng nảy sinh trong điều khiển, ước lượng, lọc, và các phạm vi khác của kỹ thuật được đưa ra dưới dạng một mô hình tổng quát *G* và bộ điều khiển *K* thiết kế bằng cách tối thiểu một số chuẩn (như $\mathcal{H}^2$) trên ma trận hàm truyền của hệ kín T_{wz} từ tín hiệu *w* đến tín hiệu *z*.

Tín hiệu

Để đánh giá sự linh hoạt của cấu trúc hệ phản hồi tổng quát đó, chỉ cần nghiên cứu đặc tính của các tín hiệu *z*, *u*, *w*, và *y* trong hình trên. Các tín hiệu này được mô tả như sau:

- *Tín hiệu được điều chỉnh.* Tín hiệu $z \in \mathcal{R}^{n_z}$ mô tả *tín hiệu được điều chỉnh* hoặc tín hiệu mà chúng ta muốn giữ nó "nhỏ" phụ thuộc vào ứng dụng và mục đích mong muốn. Các tín hiệu này bao gồm sai số bám, đầu vào cơ cấu chấp hành hoặc đầu vào điều khiển, sai lệch ước lượng tín hiệu, v.v
- *Tín hiệu điều khiển.* Tín hiệu $u \in \mathcal{R}^{n_u}$ mô tả*tín hiệu điều khiển* hoặc *các biến vận hành* là đầu ra của bộ điều khiển *K*. Các tín hiệu điều khiển có thể bao gồm luồng khí đốt vào máy, điện áp cung cấp cho một động cơ,... Chúng cũng có thể gồm các ước lượng trạng thái được cung cấp bởi *K*. Mục đích là để *K* tác động và phối hợp tín hiệu điều khiển *u* giữ cho tín hiệu cần hiệu chỉnh *z* "nhỏ". Trong thực tế, thường thì chúng ta có nhiều tín hiệu cần hiệu chỉnh hơn là các tín hiệu điều khiển ($n_z > n_u$). Nên chú ý rằng, mặc dù nói chung là như vậy, nhưng nếu chúng ta muốn điều khiển độc lập *m* biến, thì chúng ta cần ít nhất *m* biến điều khiển độc lập. Nguyên lý cơ bản này phải được duy trì trong thực tế. Càng nhiều biến điều khiển độc lập *u* (về nguyên tắc) càng dễ điều chỉnh tín hiệu *z*.
- *Tín hiệu ngoại sinh.* Tín hiệu $w \in \mathcal{R}^{n_w}$ biểu diễn *tín hiệu ngoại sinh* (hay *tín hiệu ngoài*) tác động vào hệ thống. Tín hiệu ngoại sinh bao gồm các lệnh tham chiếu được đưa ra để điều khiển hệ thống, các tác động của nhiễu lên hệ, nhiễu cảm biến, v.v
- *Tín hiệu đo.* Các tín hiệu $y \in \mathcal{R}^{n_y}$ mô tảcác giá trị đo hoặc các tín hiệu là các biến trực tiếp tới bộ điều khiển *K*. Các giá trị đo bao gồm một phần hoặc tất cả các biến trạng

thái thiết bị, các "đầu ra" của thiết bị có thể đo được, các tín hiệu điều khiển có thể đo được, các tín hiệu ngoại sinh có thể đo được,... Trong thực tế, chúng ta có nhiều tín hiệu ngoại sinh hơn các giá trị đo ($n_w > n_y$). Nói chung, càng nhiều các giá trị đo độc lập thì càng nhiều thông tin hữu dụng.

Chú thích 30.1 (Dẫn tới nguyên lý tách)

Việc kết hợp tín hiệu điều khiển u với tín hiệu cần chỉnh z là rất tự nhiên. Người ta cho rằng cặp này hoàn toàn có thể định nghĩa một bài toán điều khiển hoặc bài toán điều chỉnh. Điều này tương tự với tình huống đã bàn tới trong các bài toán LQR truyền thống. Trong bài toán đó, người ta thỏa hiệp giữa hành động điều khiển và tốc độ điều chỉnh.

Tương tự như vậy, việc kết hợp tín hiệu ngoại sinh w với tín hiệu đo y cũng rất tự nhiên. Người ta cho rằng cặp này hoàn toàn có thể định nghĩa một bài toán ước lượng hoặc bài toán phân tích tín hiệu. Điều này tương tự với tình huống đã bàn tới trong bài toán KBF truyền thống. Trong bài toán đó, người ta loại bỏ nhiễu thay cho tốc độ tạo ước lượng.

Các sự kết hợp đó như là một bài toán LQG truyền thống, cấu trúc tổng quát đưa ra một nguyên lý tách tự nhiên. Thật vậy, đây sẽ là một trường hợp của bài toán phản hồi đầu ra $\mathcal{H}^2$ mà chúng ta sẽ nghiên cứu.

Bài toán tối ưu $\mathcal{H}^2$ tổng quát

Bài toán tối ưu $\mathcal{H}^2$ tổng quát có thể được trình bày như sau:

Tìm bộ điều khiển hữu tỷ thực (có chiều hữu hạn) phù hợp K làm ổn định G sao cho chuẩn $\mathcal{H}^2$ của ma trận hàm truyền hệ kín là nhỏ nhất:

$$\min_{K} \|\mathcal{T}_{zw}(\mathcal{K})\|_{\mathcal{H}^2} \tag{30.1}$$

trong đó:

$$\|F\|_{\mathcal{H}^2} \overset{def}{=} \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} trace\{F^H(j\omega)F(j\omega)\}d\omega} \tag{30.2}$$

$$= \sqrt{\int_0^{\infty} trace\{f^H(t)f(t)\}dt} \tag{30.3}$$

$$= \|f\|_{\mathcal{L}^2(\mathcal{R}_+)} \tag{30.4}$$

và f là ma trận đáp ứng xung được kết hợp với ma trận hàm truyền F.

Chú thích 30.2 (Sử dụng chuẩn bình phương: các tín hiệu ngoại sinh dải rộng)

Chú ý rằng chuẩn bình phương đo năng lượng của đáp ứng xung và sự biến đổi của hàm đơn vị Dirac δ là duy nhất, nó cho phép chuẩn bình phương phù hợp khi các tín hiệu ngoại sinh w là dải rộng trong tự nhiên. Điều này có thể luôn luôn được lý giải bằng cách đưa ra các bộ lọc thông thấp thích hợp bên trong G. Cũng nên chú ý rằng các ý tưởng đó có các cách giải thích ngẫu nhiên. Thay vì các hàm δ, có thể dùng các nhiễu trắng với cường độ đơn vị.

Chú thích 30.3 (Bài toán điều khiển và bài toán ước lượng)

Mặc dù chúng ta đang cố tìm ra một bộ điều khiển tối ưu $\mathcal{H}^2$, nhưng phải chú ý rằng nguyên tắc đối tượng tổng quát sẽ cho phép thiết kế tối ưu trạng thái cũng như luật điều khiển với hệ số khuyếch đại động và không đổi.

Với vấn đề trình bày ở trên, rất thích hợp để nhắc lại kết quả cơ bản sau:

Bổ đề 30.1 (Chuẩn bình phương của một hệ thống ổn định)

Xét một hệ hợp thức chặt ổn định nhân quả LTI là $F = [A,\ B,\ C]$. *Ta có:*

$$\|F\|_{\mathcal{H}^2} = \|f\|_{\mathcal{L}^2(\mathbb{R}^+)} = \sqrt{CL_cC^H} = \sqrt{B^H L_o B} \tag{30.5}$$

trong đó L_c *là gramian điều khiển được của hệ thống và* L_o *là gramian quan sát được của hệ thống. Gramian điều khiển được là:*

$$L_c \overset{def}{=} \int_0^\infty e^{At} BB^H e^{A^H t}\, dt \tag{30.6}$$

là nghiệm bán xác định dương đối xứng duy nhất của phương trình Lyapunov đại số

$$AL_c + L_c A^H + BB^H = 0 \tag{30.7}$$

L_c *là nghiệm xác định dương khi và chỉ khi* (A, B) *điều khiển được. Gramian quan sát được là:*

$$L_o \overset{def}{=} \int_0^\infty e^{A^H t} C^H C e^{At}\, dt \tag{30.8}$$

là nghiệm bán xác định dương đối xứng duy nhất của phương trình Lyapunov đại số

$$A^H L_o + L_o A + C^H C = 0 \tag{30.9}$$

L_0 *là xác định dương khi và chỉ khi (A, C) quan sát được.*

Chú thích 30.4 (Chuẩn $\mathcal{H}^2$ có thể sai – Chuẩn L^∞ là quan trọng)

Một chú ý quan trọng là chuẩn $\mathcal{H}^2 / \mathcal{L}^2$ (hoặc năng lượng) của một hàm có thể rất nhỏ, trong khi bản thân hàm đó có thể rất lớn. Xét một xung nhỏ và cao, làm ví dụ. Sự quan sát này là tới hạn bởi vì có nhiều trường hợp quan trọng chúng ta đề cập nhiều đến độ cao của hàm hơn là năng lượng của nó. Một ví dụ điển hình của trường hợp này là lý thuyết ổn định Nyquist cổ điển [2, 8]. Nyquist phát biểu rằng cường độ đỉnh của hàm nhạy $S=1/(1+L)$ được kết hợp với vòng phản hồi âm chuẩn là rất quan trọng dưới dạng ổn định bền vững của vòng phản hồi. Độ nhạy lớn nghĩa là đồ thị Nyquist gần với điểm tới hạn -1, tức là một nhiễu loạn nhỏ (hoặc sai số mô hình không biết) có thể dẫn đến hệ kín không ổn định. Để giải quyết vấn đề cơ bản này chúng ta phải sử dụng hàm trọng số phụ thuộc tần số, nhưng những gì chúng ta cần là một chuẩn mà nó xác định trực tiếp các mối quan tâm này. Điều này dẫn đến cái gọi là chuẩn $\mathcal{H}^\infty$ và chuẩn $\mathcal{L}^\infty$ miễn là $\mathcal{H}^\infty / \mathcal{L}^\infty$ vẫn đúng với lý thuyết điều khiển [4, 11].

Chú thích 30.5 (Tính toán chuẩn $\mathcal{H}^2$ trong MATLAB)

Chuẩn $\mathcal{H}^2$ của hệ $F = [A,\ B,\ C,\ D]$ có thể được tính bằng cách sử dụng các lệnh MATLAB sau:

```
lc=lyap(a, b*b')
twonorm=sqrt(trace(c*lc*c'))
```

hoặc

```
lo=lyap(a', c*c')
twonorm=sqrt(trace(b'*lo*b))
```

Đối tượng tổng quát

Đối tượng tổng quát được giả thiết là có cấu trúc không gian trạng thái 2 cửa như sau:

$$G = \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right] = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0_{n_{z1}\times n_{w2}} & D_{12} \\ C_2 & D_{21} & 0_{n_{y2}\times n_{u2}} \end{array}\right] = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \tag{30.10}$$

trong đó:

$G_{ij}(s) = C_i(sI - A)^{-1}B_j$, $A \in \mathbb{R}^{n\times n}$, $B_1 \in \mathbb{R}^{n\times n_{w2}}$, $B_2 \in \mathbb{R}^{n\times n_{u2}}$, $C_1 \in \mathbb{R}^{n_{z1}\times n}$, $C_2 \in \mathbb{R}^{n_{y2}\times n}$, $D_{12} \in \mathbb{R}^{n_{z1}\times n_{u2}}$, $D_{21} \in \mathbb{R}^{n_{y2}\times n_{w2}}$

Chú thích 30.6 (Các hàm trọng số: thoả mãn các thông số thiết kế hệ kín)

Như đã trình bày ở trên, đối tượng tổng quát G có thể chứa các hàm trọng số phụ thuộc vào tần số cũng như mô hình hệ thống vật lý P (thiết bị) đang được điều khiển. Thông thường $P = G_{22} = [A,\ B_2,\ C_2]$. Các hàm trọng số trong G có thể được xem như là các tham số thiết kế (điểm nút toán học) được thực hiện bởi nhà thiết kế nhằm tác động tới bài toán $\mathcal{H}^2$ theo cách đem lại một bộ điều khiển không tối ưu – một khái niệm thường không phù hợp với các ứng dụng thực tế - nhưng thỏa mãn các thông số thiết kế hệ kín mong muốn. Các hàm trọng số có thể được sử dụng để gia trọng (phạt) các sai số bám, cơ cấu chấp hành và các mức tín hiệu khác, các sai số ước lượng trạng thái,... Bằng việc tạo ra trọng số trên tín hiệu lớn trong một dải tần xác định, chúng ta đang gián tiếp phát biểu bài toán tối ưu để đi tìm một bộ điều khiển tạo ra tín hiệu nhỏ trong dải tần đó. Bằng việc tạo ra trọng số trên tín hiệu nhỏ trong một dải tần xác định, chúng ta đang gián tiếp chuyển sang bằng lòng với việc chấp nhận một tín hiệu lớn trong dải tần đó. Ý tưởng này được minh họa qua ví dụ.

Chú thích 30.7 ($D_{11} = 0$ là cần thiết, $D_{22} = 0$ là không cần thiết)

$D_{11} = 0$ *cần thiết*. Lưu ý rằng chúng ta giả sử $D_{11} = 0$; nghĩa là không có liên hệ trực tiếp từ tín hiệu ngoại sinh w tới tín hiệu được điều chỉnh z. Giả thiết này cần cho chuẩn $\mathcal{H}^2$ của hàm truyền hệ kín T_{wz} giới hạn. Nếu $D_{11} \neq 0$ thì $\|T_{wz}\|_{\mathcal{H}^2}$ sẽ không giới hạn và bài toán $\mathcal{H}^2$ sẽ không có lời giải. Nếu chúng ta có $D_{11} \neq 0$, thêm bộ lọc hợp thức chặt trên w hoặc trên z (ví dụ $[1000/(s+1000)]I$) sẽ dẫn đến $D_{11} = 0$. Trong trường hợp này, giả thiết trên là không hạn chế.

$D_{22} = 0$ *không cần thiết*. Cũng đã giả sử rằng $D_{22} = 0$; tức là, ma trận hàm truyền D_{22} từ tín hiệu điều khiển u đến tín hiệu đo y là hợp thức chặt. Giả thiết này rất thực tế vì G_{22} (thiết bị P) thường là hợp thức chặt trong thực tế. Nếu không, động học tần số cao (chẳng hạn, động học của cơ cấu chấp hành, các chế độ linh hoạt, kỳ sinh...) có thể được đưa vào để tạo nên sự hợp thức chặt. Thậm chí còn có thể có một bộ lọc thông thấp với dải thông rộng (ví dụ, $1000/(s+1000)$) để làm G_{22} hợp thức chặt. Đây không phải là điều mong muốn bởi vì chiều của ma trận bị tăng lên, cần có một sự thay đổi mà không làm tăng chiều của G.

Người ta có thể bỏ qua D_{22} từ G_{22} để thu được một đối tượng tổng quát mới $\hat{G}$ với $\hat{D}_{22} = 0$. Sau đó, số hạng D_{22} được đưa vào trong bộ điều khiển K bằng cách lưu ý rằng u liên hệ với y như sau:

$$u = K[y + D_{22}u] = [I - KD_{22}]^{-1}Ky \tag{30.11}$$

Bộ điều khiển được thêm vào, gọi là $\hat{K}$, được biểu thị như sau:

$$\hat{K} = [I - KD_{22}]^{-1}K \tag{30.12}$$

Sau đó, bài toán $\mathcal{H}^2$ có thể được thực hiện với $\hat{G}$ và $\hat{K}$ (không cần quan tâm đến D_{22}). Khi tìm được bộ điều khiển tối ưu $\hat{K}$ cho $\hat{G}$, sẽ tính bộ điều khiển K cho G sử dụng quan hệ:

$$K = \hat{K}[I + D_{22}\hat{K}]^{-1} \tag{30.13}$$

Với cách này, giả thiết $D_{22} = 0$ không làm mất tính tổng quát.

Ma trận hàm truyền của hệ kín

Với cấu trúc đã cho của đối tượng tổng quát G, chúng ta có các mối quan hệ vòng kín sau:

$$u = Ky \tag{30.14}$$

$$= K(G_{21}w + G_{22}u) \tag{30.15}$$

$$= [I - KG_{22}]^{-1} KG_{21}w \tag{30.16}$$

$$= K[I - G_{22}K]^{-1} G_{21}w \tag{30.17}$$

$$y = [I - G_{22}K]^{-1} G_{21}w \tag{30.18}$$

$$z = G_{11}w + G_{12}u \tag{30.19}$$

$$= G_{11}w + G_{12}Ky \tag{30.20}$$

$$= [G_{11} + G_{12}K[I - G_{22}K]^{-1} G_{21}]w \tag{30.21}$$

Từ đó, ta có các ma trận hàm truyền vòng kín như sau:

$$T_{wu} = K[I - G_{22}K]^{-1} G_{21} \tag{30.22}$$

$$T_{wy} = [I - G_{22}K]^{-1} G_{21} \tag{30.23}$$

$$T_{wz} = G_{11} + G_{12}K[I - G_{22}K]^{-1} G_{21} \tag{30.24}$$

Chúng ta gọi mỗi ma trận là một *biến đổi phân đoạn tuyến tính* (LFT - linear fractional transformation) theo K.

Chú thích 30.8 (Đạt yêu cầu hệ kín)

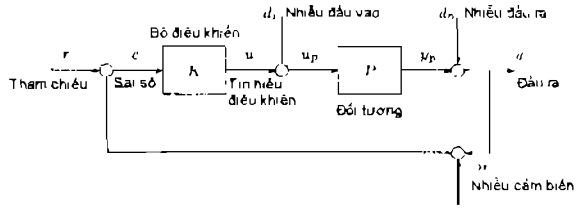
Trong phép biến đổi ở trên, đã giả sử rằng ma trận nghịch đảo $[I\text{-}G_{22}K]^{-1}$ tồn tại. Yêu cầu này được đảm bảo bởi giả thiết $D_{22} = 0$. Giả thiết này dẫn đến $G_{22}(j\omega) = D_{22} = 0$ và vì vậy tồn tại ma trận nghịch đảo.

Ví dụ sau đây trình bày cách tính bài toán độ nhạy hỗn hợp có trọng số $\mathcal{H}^2$ như thế nào để hướng tới kết quả thiết kế hệ thống điều khiển phản hồi.

Ví dụ 30.1 (Bài toán độ nhạy hỗn hợp có trọng số $\mathcal{H}^2$: nguyên lý thiết kế)

Ví dụ này xét việc thiết kế một bộ điều khiển K cho thiết bị $P = \left[\begin{array}{cc} A_p & B_p \\ C_p & D_p \end{array}\right]$ như trong hình 30.2. Để có K, chúng ta sẽ thiết lập công thức tối ưu $\mathcal{H}^2$ xem xét (một cách trực tiếp hoặc gián tiếp) các vấn đề khác nhau là điều quan trọng trong thiết kế của một vòng phản hồi tốt.

Các vấn đề thực thi hệ thống phản hồi: nói chung, trong thiết kế một bộ điều khiển phản hồi K như hình 30.2, người thiết kế phải xét từng vấn đề thực thực thi vòng kín sau:



HÌNH 30.2 Vòng phản hồi âm chuẩn

Tính ổn định vòng kín. Hệ kín phải ổn định. Điều này đòi hỏi tất cả các ma trận hàm truyền vòng kín phải ổn định. Một ma trận hàm truyền vòng kín hợp thức chặt là ma trận mà chuẩn $\mathcal{H}^2$ của nó là vô hạn, dẫn đến ma trận hàm truyền đó là không ổn định (hoặc giới hạn ổn định). Các ma trận hàm truyền hợp thức chặt ổn định cần phải có chuẩn $\mathcal{H}^2$ hạn chế.

Tuân theo điều khiển. Hệ kín phải thể hiện việc tuân theo điều khiển ở tần số thấp tốt, nghĩa là, đầu ra y (không bị lẫn với các giá trị đo của thiết bị chuẩn hóa) phải bám vào các điều khiển tham chiếu tần số thấp r được đưa ra với hệ thống phản hồi. Điều này đòi hỏi ma trận hàm truyền nhạy

$$S \overset{def}{=} [I + PK]^{-1} \tag{30.25}$$

phải nhỏ ở tần số thấp.

Giảm nhiễu. Hệ kín phải có sự suy giảm nhiễu ở tần số thấp tốt. Với nhiễu đầu ra d_0 được lấy mẫu tại đầu ra thiết bị, cần ma trận hàm truyền nhỏ ở tần số thấp. Với nhiễu đầu vào d_i được lấy mẫu tại đầu vào thiết bị, cần

$$T_{d_i y} \overset{def}{=} SP \tag{30.26}$$

phải nhỏ ở tần số thấp.

Giảm nhiễu cảm biến. Hệ kín phải giảm nhiễu đầu đo ở tần số cao n tốt. Điều này ma trận yêu cầu hàm truyền nhạy bù

$$T \overset{def}{=} I - S \tag{30.27}$$

phải nhỏ ở tần số cao.

Bền vững ổn định. Hệ kín phải tỏ ra bền vững đối với động lực phi mô hình tần số cao (ví dụ, các chế độ biến đổi, động lực ký sinh, thời gian trễ...); Điều này đòi hỏi "đỉnh" của ma trận hàm truyền hệ kín phải nhỏ ở tần số cao:

- Sai số mô hình nhân. Với một mô hình đã cho:

$$P_{act} = [I + \Delta]P \tag{30.28}$$

trong đó P_{act} biểu diễn thiết bị thật, P biểu diễn mô hình danh nghĩa, và Δ biểu diễn sai số nhân ổn định tại đầu ra thiết bị, ma trận hàm truyền hệ kín phù hợp (khi có thêm Δ) là T.

- Sai số mô hình cộng:

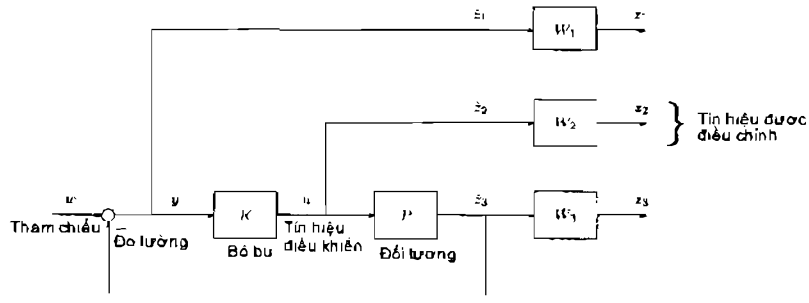
$$P_{act} = P + \Delta \tag{30.29}$$

trong đó P_{act} biểu diễn thiết bị thật, P biểu diễn mô hình danh nghĩa, và Δ tương ứng với sai số cộng ổn định tại đầu ra thiết bị, ma trận hàm truyền hệ kín phù hợp (khi có thêm) là KS.

Tác động điều khiển phù hợp. Hệ kín phải có tác động điều khiển phù hợp với các lệnh tham chiếu điển hình và nhiễu cảm biến. Điều này chủ yếu đòi hỏi "kích thước" của KS là điều khiển

được. Quá nhiều đầu vào K (như tác động đạo hàm) có thể giúp về mặt ổn định, để đạt được một giới hạn dải thông rộng và một giới hạn pha, nhưng nó có thể dẫn đến các điều khiển lớn không cần thiết với sự hiện diện của các lệnh tham chiếu r và nhiễu cảm biến n.

Các liệt kê ở trên có rất nhiều vấn đề quan trọng tác động quá trình thiết kế hệ thống điều khiển. Tuy nhiên, một phần công việc của người thiết kế là dành ưu tiên và chọn các vấn đề quan trọng nhất. Ở phần cuối, chúng ta sẽ quay lại hình 30.2 và xét thay vào đó là hệ thống "tưởng tượng" (toán học) được mô tả như hình 30.3.



HÌNH 30.3 Hệ phản hồi âm cho bài toán độ nhạy hỗn hợp có trọng số

Các hàm trọng số và ma trận hàm truyền vòng kín

Hình 30.3 gồm có các hàm trọng số xác định sẽ giúp chúng ta lập công thức tối ưu hóa $\mathcal{H}^2$ hướng tới (trực tiếp hoặc gián tiếp) các vấn đề đã đề cập ở trên. Hình trên cho thấy một hàm trọng W_1 trên tín hiệu y (sai số bám), một hàm trọng số W_2 trên tín hiệu điều khiển u, và một hàm trọng số W_3 trên đầu ra đối tượng $\hat{z}_3$. Từ hình vẽ, các tín hiệu cần điều chỉnh

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

liên hệ với các tín hiệu ngoại sinh w như sau:

$$z_1 = W_1\hat{z}_1 = W_1 S w \tag{30.30}$$

$$z_2 = W_2\hat{z}_2 = W_2 KS w \tag{30.31}$$

$$z_3 = W_3\hat{z}_3 = W_3 T w \tag{30.32}$$

Từ đó, ma trận hàm truyền vòng kín từ w tới z được cho bởi công thức

$$T_{wz} = \begin{bmatrix} W_1 S \\ W_2 KS \\ W_3 T \end{bmatrix} \tag{30.33}$$

Do T_{wz} gồm các ma trận hàm truyền "độ nhạy", chúng ta nói rằng, chúng ta có bài toán độ nhạy hỗn hợp có trọng số.

Chọn các hàm trọng số

Nói chung, các hàm trọng số W_1, W_2, W_3 được chọn để ổn định các ma trận hàm truyền thường có cấu trúc đường chéo (ít nhất là tại trạng thái ban đầu).

Trọng số nhạy. Người ta chọn trọng số nhạy W_1 trên S như sau:

$$W_1 = \left[\frac{k_1}{s+\varepsilon}\right] I_{n_y \times n_y} \tag{30.34}$$

trong đó k_1, $\varepsilon > 0$. Tham số k_1 thường được chọn lớn. Tham số ε thường được chọn nhỏ. Chọn sao cho S bị phạt nặng ở tần số thấp – ở đó chúng ta muốn K làm cho S nhỏ.

Trọng số điều khiển. Người ta chọn trọng số điều khiển W_2 trên KS như sau:

$$W_2 = k_2 I_{n_u \times n_u} \tag{30.35}$$

trong đó $k_2 > 0$ dẫn đến phạt không suy biến trên tín hiệu điều khiển u (tức là trên KS)

Trọng số đầu ra. Người ta chọn trọng số đầu ra trên W_3 trên T như sau:

$$W_3 = \left[\frac{k_3(s+z_3)}{s+p_3}\right] I_{n_y \times n_y} \tag{30.36}$$

với $k_3 > 0$ và $z_3 < p_3$. Trọng số này sẽ phạt T nặng hơn ở tần số cao.

Nói chung, phải thận trọng khi lựa chọn cấu trúc các hàm trọng số. Chọn không phù hợp có thể dẫn tới bài toán không khả thi và rất khó khăn trong quá trình thiết kế. Ví dụ, W_1 phải là phù hợp với bài toán $\mathcal{H}^2$ - ngược lại chuẩn $\mathcal{H}^2$ của $W_1 S$ là vô nghĩa (do S gần tới đồng nhất tại tần số cao). Trong khi chưa có một phương pháp thiết kế có hệ thống nào trong việc lựa chọn các hàm trọng số, thì các cấu trúc trên có thể coi là tốt (như điểm bắt đầu) trong các ứng dụng.

Mô tả vào/ ra cho đối tượng tổng quát G

Để thu được mô tả vào/ra (ma trận hàm truyền) cho đối tượng tổng quát, chúng ta phải biểu diễn các tín hiệu cần điều chỉnh z_1, z_2, z_3 và các giá trị đo y dưới dạng các tín hiệu ngoại sinh w và tín hiệu điều khiển u.

$$z_1 = W_1 \hat{z}_1 = W_1(w - Pu) = W_1 w - W_1 P u \tag{30.37}$$

$$z_2 = W_2 \hat{z}_2 = W_2 u \tag{30.38}$$

$$z_3 = W_3 \hat{z}_3 = W_3 P u \tag{30.39}$$

$$y = w - \hat{z}_3 = w - Pu \tag{30.40}$$

Từ đó, ta có mô tả vào/ra (ma trận hàm truyền) cho đối tượng tổng quát G:

$$\left[\begin{array}{c} z \\ \hline y \end{array}\right] = \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right] \left[\begin{array}{c} w \\ \hline u \end{array}\right] \tag{30.41}$$

$$\left[\begin{array}{c} z_1 \\ z_2 \\ \hline z_3 \\ \hline y \end{array}\right] = \left[\begin{array}{c|c} W_1 & -W_1 P \\ 0 & W_2 \\ \hline 0 & W_3 P \\ \hline I & -P \end{array}\right] \left[\begin{array}{c} w \\ \hline u \end{array}\right] \tag{30.42}$$

Mô tả không gian trạng thái cho đối tượng tổng quát G

Tiếp theo chúng ta thu được không gian trạng thái 2 cửa của G. Để làm được như vậy, giả sử việc mô tả không gian trạng thái như sau:

$$P = \left[A_p, \quad B_p, \quad C_p, \quad D_p \right] \quad \text{với trạng thái} \quad x_p \tag{30.43}$$

$$W_1 = \left[A_1, \quad B_1, \quad C_1, \quad D_1 \right] \quad \text{với trạng thái} \quad x_1 \tag{30.44}$$

$$W_2 = \left[A_2, \quad B_2, \quad C_2, \quad D_2 \right] \quad \text{với trạng thái} \quad x_2 \tag{30.45}$$

$$W_3 = \left[A_3, \quad B_3, \quad C_3, \quad D_3 \right] \quad \text{với trạng thái} \quad x_3 \tag{30.46}$$

Để thu được mô tả không gian trạng thái mong muốn cho G, chúng ta cần mô tả các tín hiệu $(\{\dot{x}_i\}_{i=1}^{3}, \dot{x}_p, \{z_i\}_{i=1}^{3}, y)$ theo các tín hiệu $(\{x_i\}_{i=1}^{3}, x_p, w, u)$. Đây chỉ là vấn đề tính toán và được làm như sau:

$$\dot{x}_1 = A_1 x_1 + B_1 y = A_1 x_1 + B_1 (w - C_p x_p - D_p u) = A_1 x_1 - B_1 C_p x_p - B_1 D_p u \tag{30.47}$$

$$\dot{x}_2 = A_2 x_2 + B_2 u \tag{30.48}$$

$$\dot{x}_3 = A_3 x_3 + B_3 z_3 = A_3 x_3 + B_3 (C_p x_p + D_p u) = A_3 x_3 + B_3 C_p x_p + B_3 D_p u \tag{30.49}$$

$$\dot{x}_p = A_p x_p + B_p u \tag{30.50}$$

$$z_1 = C_1 x_1 + D_1 y = C_1 x_1 + D_1 (w - C_p x_p - D_p u) = C_1 x_1 - D_1 C_p x_p + D_1 w - D_1 D_p u \tag{30.51}$$

$$z_2 = C_2 x_2 + D_2 u \tag{30.52}$$

$$z_3 = C_3 x_3 + D_3 z_3 = C_3 x_3 + D_3 (C_p x_p + D_p u) = C_3 x_1 + D_3 C_p x_p + D_3 D_p u \tag{30.53}$$

$$y = w - C_p x_p - D_p u = -C_p x_p + w - D_p u \tag{30.54}$$

Các phương trình trên có thể được viết theo dạng 2 cửa chuẩn:

$$\begin{bmatrix} \dot{x} \\ \hline z \\ \hline y \end{bmatrix} = \left[\begin{array}{c|c|c} A & B_{11} & B_{12} \\ \hline C_{11} & D_{11} & D_{12} \\ \hline C_{21} & D_{21} & D_{22} \end{array}\right] \begin{bmatrix} x \\ \hline w \\ \hline u \end{bmatrix} \tag{30.55}$$

như sau:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_p \\ \hline z_1 \\ z_2 \\ z_3 \\ \hline y \end{bmatrix} = \left[\begin{array}{cccc|c|c} A_1 & & & -B_1 C_p & & -B_1 D_p \\ & A_2 & & & & B_2 \\ & & A_3 & B_3 C_p & & B_3 D_p \\ & & & A_p & & B_p \\ \hline C_1 & & & -D_1 C_p & D_1 & -D_1 D_p \\ & C_2 & & & & D_2 \\ & & C_3 & D_3 C_p & & D_3 D_p \\ \hline & & & -C_p & I & -D_p \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_p \\ \hline w \\ \hline u \end{bmatrix} \tag{30.56}$$

Các giả thiết kiểm tra

Trong việc chọn các trọng số W_1, W_2, W_3, chúng ta phải đảm bảo rằng không có giả thiết nào của bài toán H^2 bị vi phạm. Để được như vậy, chúng ta cần $D_{11} = 0$ và $D_{22} = 0$. Để có $D_{11} = 0$, ta cần:

$$D_1 = 0 \tag{30.57}$$

Để có $D_{22} = 0$, ta cần:

$$D_p = 0 \tag{30.58}$$

Kết quả này dẫn đến:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_p \\ \hline z_1 \\ z_2 \\ z_3 \\ \hline y \end{bmatrix} = \left[\begin{array}{cccc|c} A_1 & & & -B_1C_p & B_2 \\ & A_2 & & & \\ & & A_3 & B_3C_p & \\ & & & A_p & B_p \\ \hline C_1 & & & & \\ & C_2 & & & D_2 \\ & & C_3 & \Delta_3 C_p & \\ \hline & & & -C_p & I \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_p \\ \hline w \\ \hline u \end{bmatrix} \tag{30.59}$$

Trong phần tiếp theo, sẽ phải chấp nhận thêm các giả thiết nữa trong việc mô tả không gian trạng thái 2 cửa cho đối tượng tổng quát G. Các giả thiết thêm phụ thuộc vào bài toán $\mathcal{H}^2$ cụ thể được xét.

Bài toán độ nhạy hỗn hợp tối ưu $\mathcal{H}^2$ có trọng số

Như đã nói ở trên, bài toán độ nhạy hỗn hợp tối ưu $\mathcal{H}^2$ có trọng số là đi tìm một bộ điều khiển ổn định trong hợp thức hữu tỷ-thực (có chiều xác định) K sao cho $\|T_{wz}\|_{\mathcal{H}^2}$ nhỏ nhất; tức là:

$$\min_K \|T_{wz}\|_{\mathcal{H}^2} = \min_K \left\| \begin{bmatrix} W_1 S \\ W_2 K S \\ W_3 T \end{bmatrix} \right\|_{\mathcal{H}^2} \tag{30.60}$$

Chúng ta sẽ trình bày bài toán điều khiển tối ưu này - và các bài toán tương tự như vậy - có thể được giải quyết bằng cách sử dụng phần mềm thiết kế được trợ giúp bởi máy tính (ví dụ như, MATLAB, robust control toolbox, $\mu - tools$).

Chú giải 30.9 (Tạo đối tượng tổng quát G)

Đối tượng tổng quát G rất dễ tạo với SIMULINK. Các khối đầu vào sử dụng để xác định các tín hiệu ngoại w và các tín hiệu điều khiển u. Các khối đầu ra sư dụng để xác định các tín hiệu điều chỉnh z và các giá trị đo y. Lệnh "linmod" sử dụng tạo biểu đồ khối (file SIMULINK) để đạt được không gian trạng thái 2 cửa cho G ($A, B = [B_1 B_2]$, $C = [C_1 : C_2]$, $D = [D_{11} D_{12}, D_{21}, D_{22}]$). Cú pháp của lệnh như sau:

```
[a,b,c,d]=linmod('filename')
```

Phương pháp này cho phép tạo nhanh mô hình đối tượng tổng quát.

Tổng quan về bài toán tối ưu $\mathcal{H}^2$ được xem xét

Ba bài toán cơ bản được xét trong chương này:

Bài toán phản hồi đầu ra $\mathcal{H}^2$. Nghiệm của bài toán này là một bộ bù động học dựa trên mô hình tối ưu có cấu trúc

$$K_{opt} = \left[\begin{array}{c|c} A - B_2 G_c - H_f C_2 & H_f \\ \hline -G_c & O_{n_u \times n} \end{array}\right] \tag{30.61}$$

trong đó G_c là ma trận hệ số điều khiển (phản hồi trạng thái) và H_f là một ma trận hệ số bộ lọc (quan sát). G_c được tìm bằng cách sử dụng nghiệm của phương trình Riccati đại số điều khiển (CARE – Control Algebraic Riccati Equation) – tương tự như bài toán điều chỉnh toàn phương tuyến tính (LQR – Linear Quadratic Regulator). H_f được xác định bằng nghiệm của phương trình Riccati đại số lọc (FARE) – tương tự như bài toán lọc Kalman-Bucy (KBF). Cấu trúc của K_{opt}, G_c và H_f được coi như là nghiệm của bài toán điều khiển Gaussian toàn phương tuyến tính, bài toán đem lại nguyên lý tách nổi tiếng: Các điểm cực của hệ kín là giá trị riêng của $A - B_2 G_c$, và giá trị riêng của $A - H_f C_2$.

Bài toán phải hồi trạng thái $\mathcal{H}^2$. Nghiệm của bài toán này là một bộ bù hệ số hằng tối ưu (phản hồi trạng thái) có cấu trúc

$$K_{opt} = -G_c \tag{30.62}$$

trong đó G_c là ma trận hệ số điều khiển (phản hồi trạng thái) được xác định bằng cách sử dụng nghiệm của CARE. Tương tự như tìm trong bài toán LQR. Các điểm cực của hệ kín là giá trị riêng của $A - B_2 G_c$. Nói tóm lại, bài toán này được coi như là một cách tính các ma trận hệ số điều khiển G_c có thể được sử dụng trong ứng dụng phản hồi trạng thái hoặc trong các ứng dụng bù dựa trên mô hình.

Bài toán nội suy đầu ra $\mathcal{H}^2$. Nghiệm của bài toán này là một bộ bù hệ số hằng tối ưu (tĩnh) có cấu trúc

$$K_{opt} = -H_f \tag{30.63}$$

Trong đó H_f là một ma trận hệ số bộ lọc (quan sát) được tìm bằng việc sử dụng nghiệm của phương trình *FARE*-tương tự như các bài toán KBF. Các điểm cực của hệ kín là giá trị riêng của $A - H_f C_2$. Nói tóm lại, bài toán này được coi như là một cách tính các ma trận hệ số lọc H_f có thể được sử dụng trong ứng dụng ước lượng trạng thái hoặc trong các ứng dụng bù dựa trên mô hình.

30.3 Bài toán phản hồi đầu ra $\mathcal{H}^2$

Trong phần này, chúng ta xét bài toán phản hồi đầu ra $\mathcal{H}^2$. Bài toán này đưa đến kết quả là bộ bù dựa trên mô hình (động) bao gồm một ma trận hệ số điều khiển (phản hồi trạng thái) G_c và một ma trận hệ số bộ lọc (quan sát) H_f. Bài toán này tổng quát các ý tưởng đã đưa ra trong lý thuyết LQG cổ điển.

Sau đây là giả thiết "chuẩn" của bài toán phản hồi đầu ra $\mathcal{H}^2$.

Giả thiết 30.1 (Bài toán phản hồi đầu ra $\mathcal{H}^2$)

Trong toàn bộ phần này, giả sử rằng:

Giả thiết đối tượng G_{22}. (A, B_2, C_2) ổn định và xác định được:

Giả thiết này là điều kiện cần và đủ để tồn tại một bộ điều khiển ổn định trong hợp thức K. Với giả thiết này, bộ điều khiển dựa trên mô hình (dựa trên bộ quan sát) làm ổn định hệ kín trong hình 30.1 là:

$$K = \left[\begin{array}{c|c} A - B_2 G_c - H_f (C_2 - D_{22} G_c) & H_f \\ \hline -G_c & O_{n_u \times n} \end{array}\right] \tag{30.64}$$

có $(A - B_2 G_c)$ và $(A - H_f C_2)$ ổn định, theo nguyên lý tách cổ điển của lý thuyết điều khiển tuyến tính.

Giả thiết này phát biểu rằng "*Tất cả các điểm cực của hệ hở "xấu" (bên phải trục thực và có phần ảo) phải điều khiển được bằng tín hiệu điều khiển u và quan sát được bằng các tín hiệu đo y*".

Giả sử là G thỏa mãn giả thiết. Xét độ tăng của bộ tích phân $1/s$ (hàm trọng). Độ tăng này có thể làm trái giả thiết. Ví dụ, việc hấp thụ một bộ tích phân $1/s$ (hàm trọng) lên tín hiệu ngoại sinh w vào trong G có thể trái với giả thiết về tính ổn định do nó sẽ đưa điểm cực hệ hở đến trục ảo và khi đó sẽ không điều khiển được bằng tín hiệu u. Việc hấp thụ một bộ tích phân lên tín hiệu điều chỉnh z vào trong G sẽ trái với giả thiết xác định được do nó tạo một điểm cực của hệ hở lên trục ảo dẫn đến không quan sát được bằng các các giá trị đo y. Ngược lại, việc sử dụng $1/(s+\varepsilon)(\varepsilon > 0)$ thay cho $1/s$ có thể làm cho G không trái với giả thiết.

Giả thiết trọng số điều khiển không suy biến $R = D_{21}^T D_{12} > 0$

Giả thiết này dẫn đến $D_{12} \in \mathcal{R}^{n_z \times n_u}$ có hạng cột đầy đủ (tức là, rank $D_{12} = n_u$) và vì vậy tất cả các tín hiệu u (tín hiệu điều khiển) đều ảnh hưởng tới tín hiệu điều chỉnh z thông qua D_{12} (tức là, D_{12} không có phần tử nào bằng không). Ma trận D_{12} phải "nhiều hàng" và "ít cột", tức là

$$\text{(số tín hiệu điều chỉnh) } n_z > n_u \text{ (số tín hiệu điều khiển)} \tag{30.65}$$

Ma trận $R = D_{21}^T D_{12} \in \mathcal{R}^{n_u \times n_u}$ cũng có thể được hiểu như là một trọng số trên tín hiệu điều khiển u - như ma trận trọng số điều khiển R trong bài toán LQR. Trong bài toán LQR ta nói trọng số điều khiển R trên u là không suy biến. R càng lớn, tín hiệu điều khiển càng nhỏ - bỏ qua tốc độ điều chỉnh. R lớn dẫn đến băng thông "điều chỉnh" thấp. R càng nhỏ, tín hiệu điều khiển u cho phép sẽ càng lớn – theo thứ tự tăng độ điều chỉnh. R nhỏ dẫn đến băng thông "điều chỉnh" cao.

Giả thiết bộ chỉnh định. $\begin{bmatrix} j\omega I - A & -B_2 \\ C_1 & D_{12} \end{bmatrix}$ có hạng cột đầy đủ $(n + n_u)$ với mọi ω.

Giả thiết này dẫn đến ma trận hàm truyền từ tín hiệu điều khiển u tới tín hiệu cần chỉnh z (A, B_2, C_1, D_{12}, R) không có điểm không nào bên phải trục ảo. Kết hợp (1) và (2), sẽ đảm bảo rằng Hamiltonian H_{com} gồm (A, B_2, C_1, D_{12}, R) được kết hợp với tín hiệu điều khiển u và tín hiệu cần chỉnh z – sẽ phụ thuộc vào *dom(Ric)*. Điều này đảm bảo lại rằng giải pháp của phương trình Ricati đại số điều khiển (CARE) đem lại ma trận hệ số điều khiển $G_c \in \mathcal{R}^{n_u \times n}$ sao cho $A - B_2 G_c$ ổn định.

Giả thiết này dẫn tới G không có các dạng ảo và không quan sát được bằng các tín hiệu điều chỉnh z, tức là tất cả các điểm cực của hệ hở trên trục ảo phải quan sát được bằng các tín hiệu z. Vì vậy, (A, C_1) không thể có dạng ảo không quan sát được. Đây là một điều kiện cần nhưng không đủ. Ví dụ, một bộ tích phân đưa thêm vào giá trị đo y sẽ trái với điều kiện này.

Từ điều kiện D_{12} có hạng cột đầy đủ, giả thiết này tương đương với cặp

$$(A - B_2 R^{-1} D_{12}^T C_1, (I - D_{12} R^{-1} D_{12}^T) C_1) \tag{30.66}$$

không có các dạng ảo không quan sát được.

Nếu D_{12} là ma trận vuông, thì nó có ma trận nghịch đảo và giả thiết tương đương với $A - B_2 R^{-1} D_{12}^T C_1$ không có các dạng ảo.

Nếu $D_{12}^T C_1 = 0$ (không có quan hệ chéo giữa các tín hiệu điều khiển và các trạng thái) thì giả thiết này tương đương với (A, C_1) không có các dạng ảo không quan sát được.

Giả thiết trọng số đo không suy biến. $\Theta = D_{21} D_{12}^T > 0$.

Giả thiết này dẫn đến $D_{21} \in \mathcal{R}^{n_y \times m_1}$ có hạng dòng đầy đủ (tức là hạng của $D_{21} = n_y$) và vì vậy các giá trị đo y độc lập tuyến tính với D_{21} (nghĩa là D_{21} không có không gian trống bên trái). Ma trận D_{21} vì vậy phải "ít hàng" và "nhiều cột" tức là,

$$(\text{số giá trị đo}) \; n_y < n \; (\text{số tín hiệu ngoại sinh}) \tag{30.67}$$

Ma trận $\Theta = D_{21} D_{12}^T \in \mathcal{R}^{n_y \times n_y}$ có thể được hiểu như là cường độ nhiễu cảm biến ảnh hưởng tới giá trị đo y – như là ma trận cường độ nhiễu cảm biến Θ trong bài toán KBF. Như trong bài toán KBF, chúng ta nói rằng ma trận cường độ Θ kết hợp với các giá trị đo y là không suy biến. Θ càng lớn, chúng ta càng muốn lọc thông thấp các giá trị đo y - hy sinh tốc độ ước lượng. Θ lớn dẫn đến độ rộng băng thông nhỏ để ước lượng kết hợp (quan sát). Θ càng nhỏ, chúng ta càng muốn các giá trị đo của bộ lọc thông thấp y càng nhỏ - để cân bằng sự miễn dịch nhiễu với tốc độ ước lượng. Θ nhỏ dẫn đến độ rộng dải thông lớn trong việc ước lượng kết hợp (quan sát).

Giả thiết về bộ lọc. $\begin{bmatrix} j\omega I - A & -B_1 \\ C_2 & D_{21} \end{bmatrix}$ có hạng dòng đầy đủ $(n + n_y)$ với mọi ω.

Giả thiết này hàm ý rằng ma trận hàm truyền từ các tín hiệu ngoại sinh w tới các giá trị đo y không có điểm không bên trái trục ảo. Kết hợp (1) và (3), có thể đảm bảo rằng ma trận Hamiltonian $\mathcal{H}_{fil}$ chứa $(A, B_1, C_2, D_{21}, \Theta)$ - gồm các tín hiệu ngoại sinh z và giá trị đo y - sẽ phụ thuộc vào *dom(Ric)*. Điều này đảm bảo lại rằng nghiệm của phương trình Ricati đại số bộ lọc (FARE) đem lại ma trận hệ số bộ lọc $H_f \in \mathcal{R}^{n \times n_y}$ sao cho $A - H_f G_1$ ổn định.

Giả thiết này hàm ý rằng G không có các dạng ảo không điều khiển được bằng các tín hiệu ngoại sinh w, hay tất cả các điểm cực hệ hở trên trục ảo phải điều khiển được bằng các tín hiệu ngoại sinh w. Vì vậy, (A, B_1) không thể có các dạng ảo không điều khiển được. Đây là điều kiện cần, nó chưa đủ. Một bộ tích phân đưa thêm vào điều khiển u sẽ trái với điều kiện này.

Từ điều kiện D_{21} có hạng dòng đầy đủ, giả thiết này tương đương với cặp

$$(A - B_1 D_{21}^T \Theta^{-1} C_2, \;\; B_1 (I - D_{21}^T \Theta^{-1} D_{21})) \tag{30.68}$$

không có các dạng ảo không điều khiển được.

Nếu D_{21} là ma trận vuông, thì nó có ma trận nghịch đảo và giả thiết tương đương với $A - B_1 D_{21}^T \Theta C_2$ không có các dạng ảo.

Nếu $B_1 D_{21}^T = 0$ (không có quan hệ chéo giữa nhiễu quá trình và nhiễu cảm biến), thì giả thiết này thì tương đương với (A, B_1) không có các dạng ảo điều khiển được.

Chú thích 30.10 (quan hệ hai chiều)

Trong phần trình bày ở trên, ta cần chú ý các quan hệ 2 chiều sau:

$$A \leftrightarrow A^T \tag{30.69}$$

$$B_2 \leftrightarrow C_2^T \tag{30.70}$$

$$C_1 \leftrightarrow B_1^T \tag{30.71}$$

$$D_{12} \leftrightarrow D_{21}^T \tag{30.72}$$

$$R = D_{12}^T D_{12} \leftrightarrow \Theta = D_{21} D_{21}^T \tag{30.73}$$

Dẫn đến:

- Tín hiệu điều khiển u quan hệ 2 chiều với giá trị đo y
- Tín hiệu điều chỉnh z quan hệ 2 chiều với tín hiệu ngoại sinh w.

Các ma trận Hamiltonian

Kết hợp với bài toán điều khiển tối ưu $\mathcal{H}^2$ là 2 ma trận Hamiltonian sau:

$$H_{con} = \begin{bmatrix} A & 0 \\ -C_1^T C_1 & -A^T \end{bmatrix} - \begin{bmatrix} B_2 \\ -C_1^T D_{12}^T \end{bmatrix} R^{-1} \begin{bmatrix} D_{12}^T C_1 & B_2^T \end{bmatrix} \tag{30.74}$$

$$= \begin{bmatrix} A - B_2 R^{-1} D_{12}^T C_1 & -B_2 R^{-1} B_2^T \\ -C_1^T (I - D_{12} R^{-1} D_{12}^T) C_1 & -(A - B_2 R^{-1} D_{12}^T C_1)^T \end{bmatrix} \tag{30.75}$$

$$H_{fil} = \begin{bmatrix} A^T & 0 \\ -B_1 B_1^T & -A \end{bmatrix} - \begin{bmatrix} C_2^T \\ -B_1 D_{21}^T \end{bmatrix} \Theta^{-1} \begin{bmatrix} D_{21} B_1^T & C_2 \end{bmatrix} \tag{30.76}$$

$$= \begin{bmatrix} (A - B_1 D_{21}^T \Theta^{-1} C_2)^T & -C_2^T \Theta^{-1} C_2 \\ -B_1 (I - D_{21}^T \Theta^{-1} D_{21}) B_1^T & -(A - B_1 D_{21}^T \Theta^{-1} C_2) \end{bmatrix} \tag{30.77}$$

Ma trận Hamiltonian thứ nhất được kết hợp với bài toán điều khiển phản hồi trạng thái tối ưu hoặc bài toán điều chỉnh. Ma trận thứ 2 được kết hợp với bài toán lọc tối ưu hoặc bài toán ước lượng tối ưu.

Lời giải của bài toán phản hồi đầu ra $\mathcal{H}^2$ được trình bày trong [11, trang 261-262].

Định lý 30.1 (Nghiệm của bài toán phản hồi đầu ra $\mathcal{H}^2$ đưa ra các giả thiết chuẩn)

Giả sử G thỏa mãn các giả thiết 30.1 – được gọi là các giả thiết bài toán phản hồi đầu ra $\mathcal{H}^2$. Chúng ta có:

Bộ điều khiển tối ưu $\mathcal{H}^2$ nhỏ nhất n chiều (như đối tượng tổng quát G) và cho bởi công thức:

$$K_{opt} = \left[\begin{array}{c|c} A - B_2 G_c - H_f C_2 & H_f \\ \hline -G_c & O_{n_u \times n_y} \end{array}\right] \tag{30.78}$$

trong đó ma trận khuếch đại điều khiển $G_c \in R^{n_u \times n}$ được cho bởi:

$$G_c = R^{-1}[B_2^T X + D_{12}^T C_1] \tag{30.79}$$

$X = Ric(H_{con}) \geq 0$ *là nghiệm xác định dương duy nhất của CARE:*

$$(A - B_2 R^{-1} D_{12}^T C_1)^T X + X(A - B_2 R^{-1} D_{12}^T C_1) + C_1^T (1 - D_{12}^T R^{-1} D_{12}^T) C_1 - X B_2 R^{-1} B_2^T X = 0 \tag{30.80}$$

và ma trận khuếch đại bộ lọc $H_f \in R^{n \times n_y}$ được cho bởi:

$$H_f = [Y C_2^T + B_1 D_{21}^T] \Theta^{-1} \tag{30.81}$$

$Y = Ric(H_{fil}) \geq 0$ *là nghiệm xác định dương duy nhất của FARE:*

$$(A - B_1 D_{12}^T \Theta^{-1} C_2) Y + Y (A - B_1 D_{21}^T \Theta^{-1} C_2)^T + B_1 (I - D_{21}^T \Theta^{-1} D_{21}) B_1^T - Y C_2^T \Theta^{-1} C_2 Y = 0 \tag{30.82}$$

Hơn nữa, chuẩn nhỏ nhất được cho bởi:

$$\left\|T_{wz}(K_{opt})\right\|_{\mathcal{C}} = \sqrt{\left\|M_c B_1\right\|_{\mathcal{C}}^2 + \left\|R^{1/2} G_c M_f\right\|_{\mathcal{C}}^2} \tag{30.83}$$

$$= \sqrt{trace(B_1^T X B_1) + trace(R G_c Y G_c^T)} \tag{30.84}$$

trong đó

$$M_c = [A - B_2 G_c, I_{n\times n}, C_1 - D_{12} G_c] \tag{30.85}$$

$$M_f = [A - H_f C_2, I_{n\times n}, B_1 - H_f D_{21}] \tag{30.86}$$

Cuối cùng, các điểm cực của hệ kín là giá trị riêng của $A - B_2 G_c$ *và* $A - H_f C_2$.

Chú thích 30.11 (Tính toán bộ điều khiển tối ưu $\mathcal{H}^2$ trong MATLAB)

Sau đây là chuỗi các lệnh MATLAB có thể được sử dụng để tính toán bộ điều khiển K_{opt} tối ưu $\mathcal{H}^2$ và kết quả ma trận hàm truyền hệ kín T_{wz}:

```
tss_g = mksys(a, [b1 b2], [c1; c2], [0*ones(nz, nw) d12; d21 0*ones(ny, nu)],'tss')
[ss_k ss_twz] = h2lqg(tss_g, 'schur')
[a_k, b_k, d_k] = branch(ss_k, 'a,b,c,d')
```

Lệnh "mksys" tập hợp dữ liệu không gian trạng thái 2 cửa cho đối tượng chuẩn hóa G đưa vào cấu trúc dữ liệu vectơ cột (được gọi là cây) xử lý cách chọn biến "tss" (không gian trạng thái 2 cửa). Tất cả các thông tin về chiều được giải mã vào vectơ cột. Lệnh "h2lqg" tính toán bộ điều khiển K_{opt} tối ưu $\mathcal{H}^2$ và hệ kín từ tín hiệu ngoại sinh w tới tín hiệu điều chỉnh z. Phương pháp vectơ riêng-giá trị riêng là phương pháp mặc định để giải 2 phương trình Riccati đại số liên quan. Phương pháp Schur - dựa trên biến đổi đơn nguyên Schur dưới dạng ma trận tam giác - có thể được sử dụng bằng cách đưa thêm lựa chọn "schur". Các kết quả được lưu theo thứ tự trong vectơ cây "ss_k" và "ss_twz". Lệnh "branch" sau đó được sự dụng để lấy lại biểu diễn không gian trạng thái cho K_{opt} từ vectơ cây "ss_k".

Chú thích 30.12 (Mối quan hệ LQG, giới hạn bền vững ổn định)

Định lý 30.1 giới thiệu bộ điều khiển phản hồi đầu ra tối ưu $\mathcal{H}^2$ giống với cấu trúc đã được tìm thấy trong các bài toán LQG cổ điển. Trong khi các công thức của bài toán LQR, KBF, và LQG/LTR đem lại kết quả trong hệ kín có giới hạn ổn định bền vững, thì bộ điều khiển LQG không cần có các giới hạn ổn định [3].

Điều này cũng đúng với thiết kế phản hồi đầu ra $\mathcal{H}^2$. Chúng ta sẽ trình bày các bước thao tác để giải các bài toán LQG/LTR sinh ra bộ điều khiển dựa trên mô hình với các giới hạn bền vững ổn định mong muốn – so với các thiết kế phản hồi đã tìm được đó với các bài toán LQR và KBF đã tính toán phù hợp (ví dụ, giới hạn hệ số tăng lên vô hạn, giới hạn của hệ số giảm xuống ít nhất là 6dB, giới hạn pha ít nhất là $\pm 60^0$).

Ví dụ sau trình bày tối ưu hóa độ nhạy hỗn hợp $\mathcal{H}^2$ có trọng số có thể được sử dụng để thiết kế một bộ điều khiển cho một hệ không ổn định với thời gian trễ.

Ví dụ 30.12 (Thiết kế độ nhạy hỗn hợp $\mathcal{H}^2$ có trọng số cho hệ không ổn định có thời gian trễ)

Trong ví dụ này, chúng ta xét một hệ không ổn định có thời gian trễ $\Delta = 0.05$ s (50ms). Hệ được mô hình (xấp xỉ) như sau:

$$P \approx \frac{1}{s-1}\left[\frac{2/\Delta - s}{2/\Delta + s}\right] = \frac{1}{s-1}\left[\frac{40-s}{40+s}\right] \tag{30.87}$$

Đặc điểm thiết kế. Mục đích là để thiết kế một bộ điều khiển K thỏa mãn các đặc điểm của hệ kín như sau: (1) tính ổn định hệ kín, (2) độ nhạy nhỏ hơn -60 dB với mọi tần số dưới 0.1 rad/s, (3) hệ số nhạy không vượt quá khoảng 2 và 3 rad/s, (4) độ nhạy đỉnh dưới 5 dB, (5) độ nhạy bù đỉnh dưới 10 dB.

Bài toán độ nhạy hỗn hợp $\mathcal{H}^2$ có trọng số. Để thiết kế được các đặc tính trên, chúng ta đưa ra một bài toán độ nhạy hỗn hợp $\mathcal{H}^2$ có trọng số – với hàm trọng W_1 trên độ nhạy S và hàm trọng W_2 trên KS; tức là:

$$\min_K \|T_{wz}\|_{\mathcal{H}^2} = \min_K \left\| \begin{bmatrix} W_1 S \\ W_2 KS \end{bmatrix} \right\|_{\mathcal{H}^2} \tag{30.88}$$

Các hàm trọng được sử dụng là:

$$W_1 = \frac{k_1}{s+p_1} = \frac{10}{s+0.01} \tag{30.89}$$

$$W_2 = \frac{k_2(s+z_2)}{s+p_2} = \frac{0.1(s+40)}{s+2} \tag{30.90}$$

W_1 phạt mạnh độ nhạy S ở tần số thấp (ví dụ dưới 0.001 rad/s). Trên 0.1 rad/s, W_1 nhỏ và W_2 phạt KS (với biên độ lớn hơn 1) cho tới khoảng 4 rad/s. Nghiệm tối ưu hóa $\mathcal{H}^2$ phụ thuộc một cách rất phức tạp vào các tham số định nghĩa W_1 và W_2, nên cần thời gian để xác định các tham số phù hợp.

Xây dựng đối tượng tổng quát. Đối tượng tổng quát G đã được xây dựng sử dụng SIMULINK và lệnh "linmod". Kết quả trình bày không gian trạng thái 2 cửa như sau:

$$G = \left[\begin{array}{c|c} 0 & -W_1 P \\ 0 & W_2 \\ \hline 1 & -P \end{array}\right] = \left[\begin{array}{c|c|c} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ \hline C_2 & D_{21} & D_{22} \end{array}\right] = \left[\begin{array}{cccccc} -0.01 & 0 & -40 & 1 & 1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 40 & -39 & 0 & 1 \\ \hline 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0.1 \\ \hline 0 & 0 & -40 & 1 & 1 & 0 \end{array}\right] \tag{30.91}$$

Tính bộ điều khiển tối ưu $\mathcal{H}^2$. Lệnh "mksys" được sử dụng để xây dựng không gian trạng thái 2 cửa ở trên vào một cấu trúc dữ liệu vectơ cây. Sau đó sử dụng lệnh "h2lqg" để đạt được bộ điều khiển tối ưu. Chú ý rằng đối tượng tổng quát có bậc 4 (2 cho đối tượng P, 1 cho trọng số nhạy W_1, 1 cho trọng số điều khiển W_2). Bộ điều khiển tối ưu là:

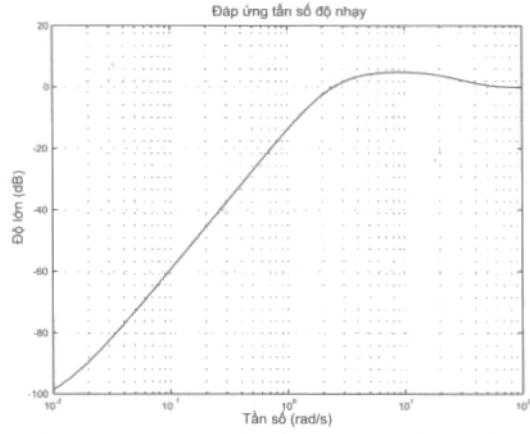
$$K_{opt} = \frac{191.0813(s+40)(s+2)(s+0.526)}{(s+1.915)(s+0.01)(s^2+84.15s+2133)} \tag{30.92}$$

cũng có bậc 4 - bậc của đối tượng chuẩn hóa G. Điểm cực tại $s = -0.01$ là một bộ tích phân xấp xỉ - một hệ quả của hàm trọng nặng W_1 đặt trên độ nhạy ở tần số thấp.

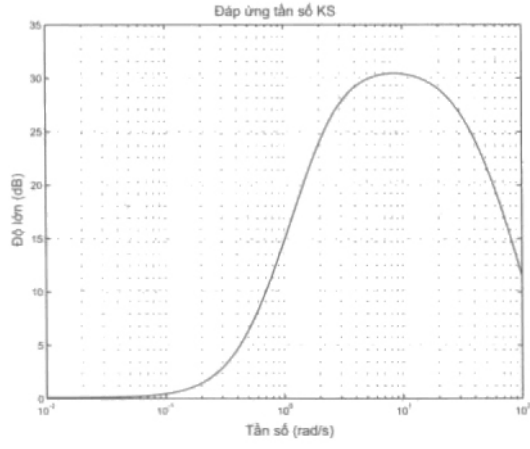
Phân tích hệ kín. Các điểm cực thu được của hệ kín (2 điểm cực của $P = G_{22}$, 4 điểm cực của bộ điều khiển K_{opt}) là:

$$s = -1, -2.0786 \pm j0.8302, -40, -40, -39.9216 \tag{30.93}$$

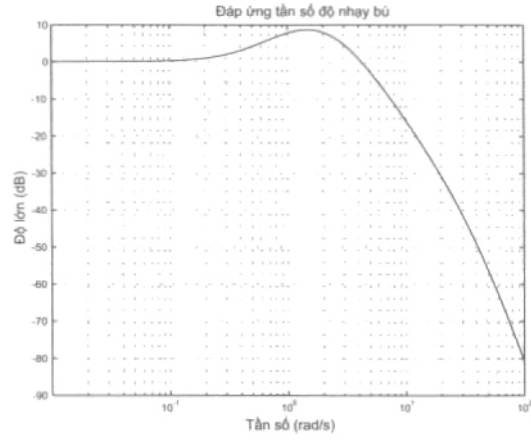
Độ nhạy thu được, *KS*, và các đáp ứng tần số độ nhạy bù theo thứ tự như hình 30.4-30.6. Các hình đó minh họa tất cả các đặc tính thiết kế ở trên. Đỉnh độ nhạy khoảng 4.855 dB. Đỉnh của độ nhạy bù khoảng 8.71 dB. Đáp ứng KS nằm gọn trong phần dò của bộ bù giữa 0.1 và 10 rad/s.



HÌNH 30.4 Đáp ứng tần số độ nhạy thiết kế $\mathcal{H}^2$



HÌNH 30.5 Đáp ứng tần số *KS* thiết kế $\mathcal{H}^2$



HÌNH 30.6 Đáp ứng tần số độ nhạy bù trong thiết kế $\mathcal{H}^2$

Tính chuẩn $\mathcal{H}^2$ nhỏ nhất. Chuẩn bình phương nhỏ nhất được tính sử dụng chuỗi các lệnh MATLAB sau:

```
lc = lyap(acl, bcl*bcl')
minnorm = sqrt(trace(ccl*c*c'))
```

Chuẩn bình phương nhỏ nhất được là 9.0648.

Ví dụ đơn giản sau minh họa lời giải của bài toán phản hồi đầu ra $\mathcal{H}^2$ có thể được sử dụng để giải bài toán LQG như thế nào.

Ví dụ 30.3 (Thiết kế LQG/LTR cho mô hình tên lửa không ổn định bậc nhất)

Ta xét một tên lửa không ổn định được mô tả bằng một mô hình bậc nhất với trạng thái x (tư thế phóng), đầu vào điều khiển u (độ lệch bánh lái thăng bằng), nhiễu quá trình (gió thổi) $w_1 = \zeta$, và nhiễu cảm biến $w_2 = \theta$.

Giả thiết là trọng tâm của tên lửa ở phía dưới tâm áp suất - ở đó sức nâng được tập trung. Giả thiết này dẫn đến sự không ổn định trong việc phóng tên lửa. Cũng giả thiết rằng mô men quán tính của tên lửa quanh trọng tâm là rất nhỏ. Giả thiết này dẫn đến một mô hình bậc nhất đơn giản. Vận tốc góc của tên lửa $\dot{x}$ được giả thiết là tỷ lệ với tư thế phóng x và nhiễu quá trình $w_1 = \zeta$. Tín hiệu cần chỉnh $z = [z_1\ z_2]^T$ gồm tư thế phóng $z_1 = x$ và đầu vào điều khiển trọng số $z_2 = \sqrt{\rho}u$. Ở đây, $\rho > 0$ là một tham số thiết kế được chọn dưới đây. Tư thế phóng của tên lửa được đo. Giá trị đo tư thế phóng y bao gồm cả nhiễu cảm biến $w_2 = \theta$.

Mô hình tên lửa. Mô hình tên lửa (được tổng quát) như sau:

$$\dot{x} = x + \xi + u \tag{30.94}$$

$$z = \begin{bmatrix} x \\ \sqrt{\rho}u \end{bmatrix} \tag{30.95}$$

$$y = x + \sqrt{\mu}\theta \tag{30.96}$$

trong đó $\mu > 0$ là tham số thiết kế được chọn ở dưới đây.

Đặc tính thiết kế. Mục đích là để thiết kế một bộ bù tối ưu $\mathcal{H}^2$ dựa trên mô hình hợp thức hữu tỷ-thực (nghĩa là, tối thiểu hóa $\|T_{w\to z}\|_{\mathcal{H}^2}$, điều này dẫn đến một hệ kín ổn định với một điểm cực trội tại $s = -5$ (thời gian đặt $t_s \approx 1s$).

Xây dựng đối tượng tổng quát. Mô hình trên có thể được viết lại như sau:

$$\dot{x} = x + [1 \quad 0]\begin{bmatrix} \xi \\ \theta \end{bmatrix} + u \tag{30.97}$$

$$z = \begin{bmatrix} 1 \\ 0 \end{bmatrix} x + 0_{2\times 2}\begin{bmatrix} \xi \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ \sqrt{\sigma} \end{bmatrix} u \tag{30.98}$$

$$y = x + \begin{bmatrix} 0 & \sqrt{\mu} \end{bmatrix}\begin{bmatrix} \xi \\ \theta \end{bmatrix} + 0_{1\times 1} u \tag{30.99}$$

Từ đó, dẫn đến

$$A = 1, \quad B_1 = [1\ 0], \quad B_2 = 1 \tag{30.100}$$

$$C_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad D_{11} = 0_{2\times 2} \quad D_{12} = \begin{bmatrix} 0 \\ \sqrt{\rho} \end{bmatrix} \tag{30.101}$$

$$C_2 = 1, \qquad D_{21} = \begin{bmatrix} 0 & \sqrt{\mu} \end{bmatrix}, \qquad D_{22} = 0_{1\times 1} \tag{30.102}$$

Các giả thiết bài toán $\mathcal{H}^2$. Bây giờ chúng ta kiểm tra từng giả thiết của bài toán phản hồi trạng thái $\mathcal{H}^2$, như đã phát biểu trong giả thiết 30.2. Từ dữ liệu ở trên, ta có $D_{11} = 0_{2\times 2}$, $D_{22} = 0_{1\times 1}$, và (A, B_2, C_2) ổn định và có thể xác định, và

$$R = D_{12}^T D_{12} = \rho > 0 \tag{30.103}$$

$$\Theta = D_{21}^T D_{21} = \mu > 0 \tag{30.104}$$

Do

$$D_{12}^T C_1 = 0 \tag{30.105}$$

$$B_1 D_{21}^T = 0 \tag{30.106}$$

nên điều kiện về hạng trục ảo gồm (A, B_2, C_1, D_{12}) và (A, B_1, C_2, D_{21}) trong giả thiết 30.2 trở nên tương đương với (A, C_1) có các dạng không quan sát được không ảo và (A, B_1) có dạng không điều khiển được không ảo. Điều này thỏa mãn hoàn toàn khi $A = 1$ có dạng không ảo. Và khi đó, tất cả các giả thiết bài toán phản hồi đầu ra $\mathcal{H}^2$ được thỏa mãn.

Đối tượng. Cuối cùng, chúng ta có hàm truyền đối tượng (tên lửa) $P = G_{22}$ là:

$$P = G_{22} = C_2 (sI - A)^{-1} B_2 \tag{30.107}$$

$$= \frac{1}{s-1} \tag{30.108}$$

G_{22} không ổn định với một điểm cực bên nửa phải của mặt phẳng tại $s = 1$. G_{22} cũng là pha cực tiểu (tức là, không có điểm không nào có Res>0).

Ma trận khuếch đại bộ lọc H_f. Khi $B_1 D_{21}^T = 0$, phương trình Riccati đại số lọc kết hợp như sau:

$$AY + YA^T + B_1 B_1^T - YC_2^T \Theta^{-1} C_2 Y = Y + Y + 1 - \frac{1}{\mu} Y^2 = 0 \tag{30.109}$$

hay

$$Y^2 - 2\mu Y - \mu = 0 \tag{30.110}$$

Áp dụng công thức toàn phương và chọn nghiệm dương (ổn định) của phương trình:

$$Y = \mu + \sqrt{\mu^2 + \mu} \tag{30.111}$$

Điều này dẫn đến ma trận khuếch đại bộ lọc:

$$H_f = YC_2^T \Theta^{-1} = 1 + \sqrt{1 + \frac{1}{\mu}} \tag{30.112}$$

Bây giờ chúng ta chọn μ để đạt được thông số điểm cực chính đã cho:

$$A = H_f C_2 = 1 - 1 - \sqrt{1 + \frac{1}{\mu}} = -5 \tag{30.113}$$

Suy ra

$$\mu = \frac{1}{24} \tag{30.114}$$

Hàm truyền hệ hở KBF kết hợp là:

$$G_{KF} = -C_2(sI - A)^{-1} H_f \tag{30.115}$$

$$= \frac{-6}{s-1} \tag{30.116}$$

Sau đây, chúng ta sẽ thấy nó là hàm truyền lặp hở xấp xỉ PK_{tpm} trong thiết kế cuối cùng. Trong trường hợp này, G_{KF} biểu diễn hàm truyền vòng hở cần đạt.

Ma trận hệ số điều khiển G_c: Từ phương trình $B_1 D_{21}^T = 0$, phương trình Riccati đại số điều khiển kết hợp được cho bởi:

$$A^T X + XA + C_1^T C_1 - XB_2 R^{-1} B_2 X = X + X + 1 - \frac{1}{\rho} X^2 = 0 \tag{30.117}$$

hay:

$$X^2 - 2\rho X - \rho = 0 \tag{30.118}$$

Áp dụng công thức toàn phương và chọn nghiệm dương (ổn định) của phương trình:

$$X = \rho + \sqrt{\rho^2 + \rho} \tag{30.119}$$

Điều này dẫn đến ma trận khuếch đại điều khiển sau:

$$G_c = R^{-1} B_2^T X = 1 + \sqrt{1 + \frac{1}{\rho}} \tag{30.120}$$

Từ đó ta có điểm cực hệ kín (bộ điều chỉnh) tại:

$$A - B_2 G_c = 1 - 1 - \sqrt{1 + \frac{1}{\rho}} = -\sqrt{1 + \frac{1}{\rho}} \tag{30.121}$$

Chú ý rằng với ρ lớn (tham khảo điều khiển "đắt" trong các bài LQR) chúng ta có một điểm cực hệ kín tại $s = -1$, và một tại $s = -1$. Chúng ta sẽ chọn tham số thiết kế ρ nhỏ (tham khảo điều khiển "rẻ" trong các bài toán điều khiển LQR) sao cho điểm cực hệ kín $s \approx -1/\sqrt{\rho}$ là nhanh và điểm cực bộ lọc hệ kín tại $s = 5$ là điểm cực hệ kín trội.

Bộ bù dựa trên mô hình phản hồi đầu ra tối ưu $\mathcal{H}^2$. Bộ bù dựa trên mô hình phản hồi đầu ra tối ưu $\mathcal{H}^2$ là:

$$K_{tpm} = \left[\begin{array}{c|c} A - B_2 G_c - H_f C_2 & H_f \\ \hline -G_c & 0_{n_u \times n_y} \end{array}\right] \tag{30.122}$$

trong đó:

$$A - B_2 G_c - H_f C_2 = 1 - 1 - \sqrt{1 + \frac{1}{\rho}} - 1 - \sqrt{1 + \frac{1}{\mu}} \tag{30.123}$$

$$= 1 - 1 - \sqrt{1 + \frac{1}{\rho}} - 1 - \sqrt{1 + 24} \tag{30.124}$$

$$= -6 - \sqrt{1 + \frac{1}{\rho}} \tag{30.125}$$

$$H_f = 1 + \sqrt{1 + \frac{1}{\mu}} \tag{30.126}$$

$$= 1 + \sqrt{1 + 24} \tag{30.127}$$

$$= 6 \tag{30.128}$$

$$G_c = 1 + \sqrt{1 + \frac{1}{\rho}} \tag{30.129}$$

Từ đó, hàm truyền của bộ bù là:

$$K_{opt} = -G_c (sI - A + B_2 G_c + H_f C_2)^{-1} H_f \tag{30.130}$$

$$= \frac{-6(1 + \sqrt{1 + 1/\rho})}{s + 6 + \sqrt{1 + 1/\rho}} \tag{30.131}$$

Với ρ nhỏ (điều khiển rẻ), điều này dẫn đến:

$$K_{opt} \approx \frac{-6(1 + \sqrt{\rho})}{s + 1/\sqrt{\rho}} \tag{30.132}$$

Hàm truyền hệ hở. Hàm truyền hệ hở kết hợp là:

$$PK_{opt} = -C_2 (sI - A)^{-1} B_2 \qquad G_c (sI - A + B_2 G_c + H_f C_2)^{-1} H_f \tag{30.133}$$

$$= \frac{1}{s - 1} \left[\frac{-6(1 + \sqrt{1 + 1/\rho})}{s + 6 + \sqrt{1 + 1/\rho}} \right] \tag{30.134}$$

Với ρ nhỏ (điều khiển rẻ), nó trở thành:

$$PK_{opt} \approx \frac{1}{s - 1} \left[\frac{-6(1/\sqrt{\rho})}{s + 1/\sqrt{\rho}} \right] \tag{30.135}$$

Khôi phục truyền lặp (LTR). Từ đây, chúng ta thấy rằng tham số ρ tiến tới 0 (điều khiển rẻ), hàm truyền hệ hở tiến tới hàm truyền hệ hở KBF G_{KF}; tức là:

$$\lim_{\rho \to 0^+} G_{22} K_{opt} = \frac{-6}{s - 1} \tag{30.136}$$

$$= G_{KF} \tag{30.137}$$

Điều này cho thấy rằng ρ tiến tới 0 (điều khiển rẻ), hàm truyền hệ hở thực tế PK_{opt} tiến tới hàm truyền vòng lặp cần đạt G_{KF}. Thủ tục khôi phục một hàm truyền vòng lặp cần đạt như trên (với các thuộc tính hệ kín mong muốn) sử dụng một bộ điều khiển LQG được gọi là *LQG với khôi phục truyền lặp hoặc LQG/LTR.*

Chọn điểm cực điều chỉnh vòng kín ở xa. Với ρ nhỏ, hệ kín ổn định với các điểm cực vòng kín là $s=-5$ và $s\approx -1/\sqrt{\rho}$. Chọn $\rho = 1/2500$ có thể sẽ tốt. Điều này dẫn đến một điểm cực hệ kín đáp ứng nhanh $s\approx -50$ và làm cho điểm cực bộ lọc hệ kín $s=5$ là điểm cực trội như yêu cầu.

Giới hạn bền vững ổn định. Các thiết kế LQG và $\mathcal{H}^2$ được biết đến vì không cần giới hạn bền vững ổn định tốt. Trong thực tế, đó là điều không tốt [3]. Các thiết kế LQG/LTR với đối tượng pha cực tiểu, ví dụ như $P=1/(s-1)$, đã được đảm bảo giới hạn bền vững ổn định. Các thiết kế LQG/LTR tạo ra các giới hạn gần với các giới hạn của thiết kế LQR và KBF; tức là không xác định với giới hạn hệ số khuếch đại trên, giới hạn hệ số khuếch đại dưới ít nhất là 6dB, giới hạn pha ít nhất là $\pm 60^0$. Thiết kế LQG/LTR cuối cùng:

$$PK_{opt} = \frac{-6}{s-1}\left[\frac{50}{s+50}\right] \tag{30.138}$$

cung cấp một giới hạn hệ số khuếch đại trên không xác định và giới hạn hệ số khuếch đại dưới là $1/6(-15.56\,dB)$. Tần số cắt hệ số khuếch đại duy nhất đạt được là $\omega_g = \sqrt{35} = 5.92\,rad/s$ và giới hạn pha kết hợp khoảng 99.59^0 là điều không tồi.

Ví dụ sau mở rộng ý tưởng LQG/LTR đã trình bày trong ví dụ 30.3 với hệ MIMO tổng quát – có thể thiết kế vòng phản hồi (với giới hạn bền vững không đáng kể) bằng tối ưu $\mathcal{H}^2$.

Ví dụ 30.4 (Thiết kế điều khiển MIMO LQG và LQG/LTR bằng tối ưu $\mathcal{H}^2$)

Chúng ta xét một đối tượng *P* (MIMO – Multi Input Multi Output) được định nghĩa bởi không gian trạng thái:

$$\dot{x} = Ax + Bu \tag{30.139}$$

$$y = Cx \tag{30.140}$$

Giả thiết là đối tượng $P=[A,B,C]$ là ổn định và xác định được.

Mục định là nhằm chứng minh lời giải của bài toán phản hồi đầu ra tối ưu $\mathcal{H}^2$ đã được trình bày có thể sử dụng để giải quyết bài toán điều khiển LQG dạng MIMO. Chúng ta sẽ trình bày một phương pháp đưa đến khái niệm LTR – nhờ đó chúng ta sử dụng một bộ điều khiển LQG dựa trên mô hình để khôi phục ma trận hàm truyền vòng mục tiêu với các thuộc tính vòng kín mong muốn. Mục tiêu của chúng ta không phải là một bài toán điều khiển LQG tối ưu ngẫu nhiên, mà là thiết kế các luật điều khiển với các đặc tính hệ kín mong muốn.

Xây dựng đối tượng tổng quát G. Với mục đích cuối cùng là một bộ bù dựa trên mô hình được định nghĩa bởi một ma trận khuếch đại điều khiển G_c và ma trận khuếch đại bộ lọc H_f, chúng ta xét đối tượng tổng quát sau:

$$\dot{x} = Ax + L\xi + Bu \tag{30.141}$$

$$z = \left[\frac{Mx}{\sqrt{\rho}I_{n_u \times n_u}}\right] \tag{30.142}$$

$$y = Cx + \sqrt{\mu}\theta \tag{30.143}$$

trong đó u là tín hiệu điều khiển, x là trạng thái đối tượng (tổng quát), $w_1 = \xi$ mô tảnhiễu quá trình trong phương trình trạng thái, $w_2 = \theta$ mô tảnhiễu cảm biến trong phương trình giá trị đo, $A \in R^{n\times n}$, $L \in R^{n\times n_w}$, $B \in R^{n\times n_u}$, $M \in R^{n_z \times n}$, $C \in R^{n_y \times n}$, $n_z = n_u, \rho > 0, \mu > 0$.

Các giả thiết tham số thiết kế. Phải có 1 trong 2 giả thiết sau:

(A, L) không có các dạng không điều khiển được ảo và (A, M) là xác định được, hoặc

(A, L) là ổn định và (A, M) không có các dạng không quan sát được ảo.

Ở đây, L, M, μ, và ρ được coi như là "các tham số thiết kế" được chọn theo thứ tự để thu được các ma trận khuếch đại điều khiển và khuếch đại bộ lọc G và H_f sao cho kết quả bộ bù dựa trên mô hình có được các đặc tính của hệ kín mong muốn.

Không gian trạng thái hai cửa cho đối tượng tổng quát G. Mô hình trên có thể được viết lại dưới dạng không gian trạng thái 2 cửa như sau:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ y \end{bmatrix} = \left[\begin{array}{c|c|c} A & \begin{bmatrix} L & 0_{n\times n_y} \end{bmatrix} & B \\ \hline \begin{bmatrix} M \\ 0_{n_u\times n} \end{bmatrix} & \begin{bmatrix} 0_{n_y\times n_u} & 0_{n_y\times n_y} \\ 0_{n_u\times n_u} & 0_{n_u\times n_y} \end{bmatrix} & \begin{bmatrix} 0_{n_y\times n_u} \\ \sqrt{\rho} I_{n_u\times n_u} \end{bmatrix} \\ \hline C & \begin{bmatrix} 0_{n_y\times n_u} & \sqrt{\mu} I_{n_y\times n_y} \end{bmatrix} & \end{array}\right] \begin{bmatrix} x \\ \xi \\ \theta \\ u \end{bmatrix} \tag{30.144}$$

Kiểm tra các giả thiết phản hồi đầu ra $\mathcal{H}^2$. Bây giờ chúng ta chắc chắn rằng tất cả các giả thiết của bài toán phản hồi đầu ra $\mathcal{H}^2$ trong giả thiết 30.2 được thỏa mãn.

- *Giả thiết* $P = G_{22}$. Do mô hình $P = G_{22} = [A, B, C]$ là ổn định và có thể xác định được, ta có $(A, B_2 = B, C_2 = C)$ là ổn định và có thể xác định được.
- *Giả thiết về bộ chỉnh định.* Do

$$D_{12} = \begin{bmatrix} 0_{n_y\times n_u} \\ \sqrt{\rho} I_{n_u\times n_u} \end{bmatrix}$$

có hạng cột đầy đủ, ta có ma trận trọng số điều khiển $R = D_{12}^T D_{12} = \rho I_{n_u\times n_u} > 0$ không suy biến.

Từ $D_{12}^T C_1 = 0$, dẫn đến điều kiện hạng (cột) trên trục ảo gồm (A, B_2, C_1, D_{12}) trong giả thiết 30.2 tương đương với $(A - B_2 R^{-1} D_{12}^T C_1, (I - D_{12} R^{-1} D_{12}^T C_1) = (A, C_1)$ không có các dạng không quan sát được ảo. Từ (A, M) hoặc có thể xác định được hoặc không có các dạng không quan sát được ảo, dẫn đến:

$$A, C_1 = \begin{bmatrix} M \\ 0_{n_u\times n} \end{bmatrix}$$

không có các dạng không quan sát được ảo. Do đó, ma trận Hamiltonian H_{cm} kết hợp dẫn tới nghiệm phương trình Riccati và ma trận khuếch đại điều khiển $A - BG_c$ là ổn định.

- *Giả thiết về bộ lọc.* Do $D_{21} = [0_{n_y\times n_u} \sqrt{\mu} I_{n_y\times n_y}]$ có hạng dòng đầy đủ, nên ma trận trọng số đo là không suy biến.

Từ $B_1 D_{21}^T = 0$, dẫn đến điều kiện hạng (dòng) trên trục ảo gồm (A, B_1, C_1, D_{21}) trong giả thiết 30.2 tương đương với $(A - B_1 D_{21}^T \Theta^{-1} C_2, B_1 (I - D_{21} \Theta D_{21}) = (A, B_1)$ không có các dạng không điều khiển được ảo. Từ (A, L) hoặc ổn định hoặc không có các dạng không điều được ảo, dẫn đến $(A, B_1 = \begin{bmatrix} L & 0_{n\times n_y} \end{bmatrix})$ không có các dạng ảo không điều khiển được ảo. Do đó, ma trận Hamiltonian H_{fl} sẽ dẫn tới nghiệm phương trình Riccati và ma trận khuếch đại bộ lọc H_f là $A - H_f C$ ổn định.

Từ những điều cho ở trên, dẫn đến là tất cả các giả thiết của bài toán phản hồi đầu ra $\mathcal{H}^2$ trong giả thiết 30.2 được thỏa mãn.

Ma trận khuếch đại điều khiển. Ma trận khuếch đại điều khiển G_c được cho bởi công thức

$$G_c = R^{-1}B^T X \tag{30.145}$$

Trong đó $X \geq 0$ là nghiệm dương duy nhất của CARE:

$$A^T X + XA + C^T C - XBR^{-1}BX = 0 \tag{30.146}$$

Thêm nữa, $A - BG$ ổn định.

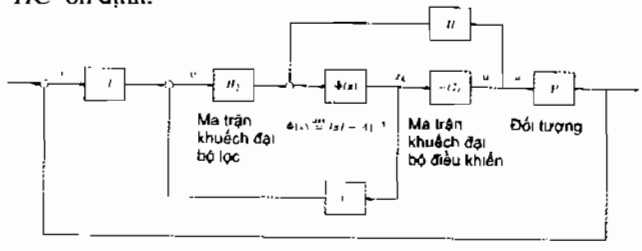
Ma trận khuếch đại bộ lọc. Ma trận khuếch đại bộ lọc H_f được cho bởi công thức

$$H_f = YC^T \Theta^{-1} \tag{30.147}$$

Trong đó $Y > 0$ là nghiệm dương duy nhất của FARE:

$$AY + YA^T + LL^T - YC^T\Theta^{-1}CY = 0 \tag{30.148}$$

Hơn nữa, $A - HC$ ổn định.



HÌNH 30.7 Vòng phản hồi âm với bộ bù dựa trên mô hình LQG và đối tượng

Bộ bù (LQG) tối ưu $\mathcal{H}^2$. Bộ bù tối ưu $\mathcal{H}^2$ tối thiểu hoá chuẩn $\mathcal{H}^2$ của ma trận hàm truyền từ các tín hiệu ngoại sinh $w = \begin{bmatrix} \xi \\ \theta \end{bmatrix}$ đến các tín hiệu cần chỉnh $z = \begin{bmatrix} C_x \\ \sqrt{\rho} I_{n_u \times n_u} \end{bmatrix}$ được cho bởi:

$$K_{opt} = \left[\begin{array}{c|c} A - BG_c - H_f C & H_f \\ \hline G_c & 0_{n_u \times n} \end{array}\right] \tag{30.149}$$

Chú ý rằng dấu trừ trên G_c (toàn bộ phần mẫu số bên tay trái của K_{opt}) được bỏ đi khi thực hiện hệ thống phản hồi âm như hình 30.7. Theo nguyên lý tách, các điểm cực của hệ kín là các trị riêng của $A - BG_c$ và $A - H_f C$.

Giới hạn bền vững ổn định. Phải nhấn mạnh rằng kết quả bộ điều khiển K_{opt}, mặc dù ổn định nhưng có các giới hạn bền vững ổn định không tốt [3]. Điều này bất chấp một thực tế là vòng điều chỉnh kết hợp

$$G_{LQ} = G_c(sI - A)^{-1}B \tag{30.150}$$

và vòng lọc

$$G_{KF} = C(sI - A)^{-1}H_f \tag{30.151}$$

khi coi như các ma trận hàm truyền hệ hở MIMO nằm trong vòng phản hồi âm, có các giới hạn bền vững ổn định rất tốt như sau: giới hạn khuếch đại trên không xác định, giới hạn khuếch đại dưới ít nhất là 6 dB, và giới hạn pha ít nhất là $\pm 60^0$. Điều này xảy ra một câu hỏi tự nhiên sau:

Có cách nào có thể chọn ma trận khuếch đại điều khiển G_c và ma trận khuếch đại bộ lọc H_f sao cho kết quả bộ bù dựa trên mô hình K_{mpc} đem lại vòng phản hồi có giới hạn tốt như trên?

Rất may, câu trả lời là có! Hai phương pháp đem lại các giới hạn ổn định có thể so sánh được tại đầu vào đối tượng và đầu ra đối tượng (nhưng không đồng thời) được trình bày ngay sau đây.

Phương pháp khôi phục truyền lặp (LTR – Loop Tranfer Recovery Methods). Là phương pháp thực hiện để đạt được một thiết kế phản hồi với giới hạn ổn định tốt. Quá trình này gồm 2 bước:

1. *Thiết kế vòng lặp cần đạt*: Thiết kế một ma trận hàm truyền vòng hở cần đạt có các đặc tính vòng kín mong muốn. Vòng lặp cần đạt này có thể kết hợp với đầu ra mô hình, ký hiệu là L_o. Trong trường hợp này, L_o biểu thị PK_{mpc} mong muốn. Nếu kết hợp với đầu vào mô hình, ký hiệu là L_i. Khi đó, L_i biểu thị $K_{mpc}P$ mong muốn. (Nói chung là $PK_{mpc}P \neq K_{mpc}P$).
2. *Khôi phục vòng lặp cần đạt bằng bộ bù dựa trên mô hình*: sử dụng bộ bù dựa trên mô hình $K_{mpc} = [A - BG_c - H_fC, H, G]$ để khôi phục vòng lặp mong muốn (L_o hoặc L_i).

   Nếu chúng ta muốn khôi phục L_o (tức là các đặc tính tốt tại đầu ra đối tượng), thì chúng ta muốn $PK_{mpc} \approx L_o$. Cách này gọi là khôi phục hàm truyền lặp tại đầu ra đối tượng (LTRO -Loop transfer recovery at the plant output).

   Nếu chúng ta muốn khôi phục L_i (tức là các đặc tính tốt tại đầu vào đối tượng), thì chúng ta muốn $K_{mpc} \approx L_i$. Cách này gọi là khôi phục hàm truyền lặp tại đầu vào đối tượng (LTRI -Loop transfer recovery at the plant input).

   Chú ý: Nói chung, các đặc tính được kết hợp với ngắt vòng lặp tại đầu ra đối tượng (các đặc tính của PK_{mpc}) khác (có lẽ rất khác) với các đặc tính kết hợp ngắt vòng lặp tại đầu vào đối tượng (các đặc tính của $K_{mpc}P$). Thường rất khó để cả PK_{mpc} và $K_{mpc}P$ đều có các đặc tính tốt (ví dụ, giới hạn, v.v..). Người thiết kế phải cân bằng các đặc tính tốt tại đầu ra đối tượng cho các đặc tính tốt tại đầu vào đối tượng, hoặc ngược lại.

Các phương pháp dựa trên $\mathcal{H}^2$ cho LTRO và LTRI sẽ được đưa ra dưới đây:

- Khôi phục hàm truyền vòng lặp tại đầu ra đối tượng (LTRO).

1. *Thiết kế vòng lặp cần đạt L_o*. Bước thứ nhất là để thiết kế một vòng lặp cần đạt $L_o = C(sI - A)^{-1}H_f$ với các đặc tính hệ kín mong muốn (ví dụ, tính ổn định, độ nhạy, độ nhạy bù, giới hạn bền vững ổn định...). Bước này có thể được thực hiện bằng cách sử dụng nhiều phương pháp (tuỳ thuộc phương pháp nào bạn cảm thấy phù hợp).

   Thủ tục cho các đặc tính tốt tại đầu ra đối tượng dựa trên phương pháp *KBF*. Ý tưởng này là để chọn ma trận thiết kế L sao cho các giá trị kỳ dị $G_{FOL} = C(sI - A)^{-1}H_f$ là tốt; ví dụ, giá trị kỳ dị nhỏ nhất lớn tại các tần số thấp, giá trị kỳ dị lớn nhất nhỏ tại các tần số cao, các giá trị kỳ dị đi qua 0 dB với độ dốc -20 dB/dec, v.v.

   Sau đó chúng ta giải phương trình FARE với $A, L, C, \Theta = \mu I_{n_y \times n_y}$ - sử dụng $\mu > 0$ để điều chỉnh dải thông của vòng lặp cần đạt $L_o = G_{KF} = C(sI - A)^{-1}H_f$. μ càng nhỏ (hoặc càng lớn) thì dải thông càng lớn (hoặc càng nhỏ).

   Nguyên tắc để hình thành vòng lặp cần đạt là $L_o = G_{KF}$.

- Đẳng thức miền tần số Kalman (KFDE) tạo ra thiết kế vòng lặp:

$$[I + G_{KF}(j\omega)][I + G_{KF}(j\omega)]^H = I + \left[\frac{1}{\sqrt{\mu}}G_{FOL}(j\omega)\right]\left[\frac{1}{\sqrt{\mu}}G_{FOL}(j\omega)\right]^H \qquad (30.152)$$

Từ đó, ta có:

$$\sigma_i[I + G_{KF}(j\omega)] = \sqrt{1 + \frac{1}{\mu}\sigma_i^2[G_{FOL}(j\omega)]} \tag{30.153}$$

Điều này gợi ý rằng bằng việc thiết kế G_{FOL}, ta có thể thiết kế vòng lặp cần đạt $L_o = G_{KF}$. Nếu G_{FOL} lớn tại tần số thấp thì ta cần:

$$G_{KI}(j\omega) \approx \frac{1}{\mu} G_{FOL}(j\omega) \tag{30.154}$$

tại tần số thấp. Nó cho thấy ma trận L nên được sử dụng để thiết kế vòng lặp cần đạt $L_o = G_{KF}$ trong khi $\mu > 0$ sử dụng để điều chỉnh dải thông vòng lặp cần đạt – tăng hay giảm μ dẫn đến giảm hay tăng dải thông đó. Kết quả $L_o = G_{KF}$ đảm bảo có các đặc tính hệ kín tốt được mô tả như bên dưới.

- Sự tương quan giá trị kỳ dị ở trên dẫn đến:

$$\sigma_{\min}[I + G_{KF}(j\omega)] \geq 1 \tag{30.155}$$

với mọi ω. Điều này dẫn đến các giá trị kỳ dị độ nhạy kết hợp thỏa mãn:

$$\sigma_{\max}[S_{KF}(j\omega)] = \frac{1}{\sigma_{\min}[S_{KF}(j\omega)]^{-1}} \leq 1 \tag{30.156}$$

với mọi ω, trong đó:

$$S_{KF}(j\omega) = [I + G_{KF}(j\omega)]^{-1} \tag{30.157}$$

- Từ mối quan hệ giá trị kỳ dị độ nhạy ở trên, chúng ta có được các giới hạn vòng lặp KBF đã biết như sau:
  - giới hạn hệ số khuếch đại trên là không xác định,
  - giới hạn hệ số khuếch đại dưới ít nhất là $\frac{1}{2}(6\,dB)$,
  - giới hạn pha ít nhất là $\pm 60^0$.

Các giới hạn nói trên thích hợp với sự thay đổi hệ số khuếch đại đồng thời và độc lập khi vòng lặp bị ngắt tại đầu ra. Nó giống như các giới hạn pha ở trên. Các giới hạn ở trên không đảm bảo các nhiễu hệ số khuếch đại và pha là đồng thời. Nên chú ý rằng các giới hạn đó có thể dễ dàng thay đổi bằng cách sử dụng tiêu chuẩn ổn định Nyquist.

- Từ quan hệ giá trị độ nhạy đơn ở trên, ta có mối quan hệ giá trị độ nhạy đơn như sau:

$$\sigma_{\max}[T_{KF}(j\omega)] = \sigma_{\max}[I - S_{KF}(j\omega)] \leq 1 + \sigma_{\max}[S_{KF}(j\omega)] \leq 2 \qquad (6\text{dB}) \tag{30.158}$$

với mọi ω, trong đó:

$$T_{KF} = I - S_{KF} = G_{KF}[1 + G_{KF}]^{-1} \tag{30.159}$$

2. *Khôi phục vòng lặp cần đạt L_o sử dụng bộ bù dựa trên mô hình.* Bước thứ 2 là sử dụng một bộ bù dựa trên mô hình $K_{opt} = [A - BG_c - H_f C, H_f, G_c]$ trong đó G_c được tìm bằng cách giải phương trình CARE với $A, B, M = C, R = \rho I_{m_u \times m_u}$ với ρ là một giá trị vô hướng dương rất nhỏ. Khi ρ nhỏ, chúng ta gọi là bài toán điều khiển rẻ.

- Nếu mô hình $P = [A, B, C]$ là pha cực tiểu thì:

$$\lim_{\rho \to 0^+} X = 0 \tag{30.160}$$

$$\lim_{\rho \to 0^+} \sqrt{\rho} G_c = WC \tag{30.161}$$

với các ma trận trực chuẩn W (tức là $W^T W = WW^T = I$)

$$\lim_{\rho \to 0^+} PK_{opt} = L_o \tag{30.162}$$

Trong trường hợp đó, $PK_{opt} \approx L_o$ với ρ nhỏ và vì vậy PK_{opt} sẽ có giới hạn bền vững gần với giới hạn bền vững của L_o (tại đầu ra mô hình) – bất kỳ phương pháp nào được dùng để thiết kế L_o. Phải chú ý rằng điều kiện pha cực tiểu trên mô hình P là điều kiện đủ. Nó là không cần thiết. Hơn nữa, G_c không cần tính bằng CARE. Trong thực tế, G_c thỏa mãn điều kiện giới hạn cho vài ma trận có thể nghịch đảo và chắc chắn rằng $A - BG_c$ là ổn định (với ρ nhỏ), sẽ dẫn đến khôi phục hàm truyền của hệ tại đầu ra đối tượng. Điều này đem lại kết quả của cấu trúc các bộ bù dựa trên mô hình và không còn gì để làm với các bài toán điều khiển tối ưu và lọc.

Giả thiết rằng điều kiện giới hạn $\lim_{\rho \to 0^+} \sqrt{\rho} G_c = WC$ giữ các ma trận khả đảo W, khôi phục truyền lặp của vòng lặp cần đạt $L_o = C(sI - A)H_f$, có thể được chứng minh như sau. Với ρ ta có:

$$G_c \approx \frac{WC}{\sqrt{\rho}} \tag{30.163}$$

từ đó dẫn đến:

$$PK_{opt} = PG_c(sI - A + BG_c + H_f C)^{-1} H_f \tag{30.164}$$

$$\approx P\frac{WC}{\sqrt{\rho}}\left(sI - A + B\frac{WC}{\sqrt{\rho}}\right)^{-1} H_f \tag{30.165}$$

$$\approx P\frac{WC}{\sqrt{\rho}}(sI - A)^{-1}\left[I + B\frac{WC}{\sqrt{\rho}}(sI - A)^{-1}\right]^{-1} H_f \tag{30.166}$$

$$\approx P\frac{WC(sI - A)^{-1}}{\sqrt{\rho}}\left[I + B\frac{WC(sI - A)^{-1}}{\sqrt{\rho}}\right]^{-1} H_f \tag{30.167}$$

$$\approx P\left[\frac{WP}{\sqrt{\rho}}\right]^{-1} W\frac{C(sI - A)^{-1} H_f}{\sqrt{\rho}} \tag{30.169}$$

$$\approx C(sI - A)^{-1} H_f = L_o \tag{30.170}$$

Ý tưởng chính (dưới dạng đại số) là ρ tiến tới 0, đường phản hồi C trong bộ bù $K_{opt} = [A - BG_c - H_f C, H_f, G_c]$ được ngắt (H.30.7) và các đặc tính tốt để giữ tại cái mới v (ví dụ ma trận hàm truyền hệ hở tại v là $L_o = C(sI - A^{-1})H_f$) trong hình 30.7 truyền tới tín hiệu sai lệch e (đầu vào bộ bù, hoặc đầu ra đối tượng) trong hệ phản hồi.

- Khôi phục truyền lặp tại đầu vào đối tượng (LTRI).

1. *Thiết kế vòng lặp cần đạt* L_t. Bước thứ nhất là để thiết kế một vòng lặp cần đạt $L_t = G_c(sI - A)^{-1}B$ với các đặc tính hệ kín mong muốn (ví dụ, tính ổn định, độ nhạy, độ nhạy bù, giới hạn bền vững ổn định...). Bước này có thể được thực hiện bằng nhiều phương pháp (tuỳ thuộc phương pháp nào bạn cảm thấy phù hợp).

   Một thủ tục đem lại các đặc tính tốt tại đầu vào đối tượng được dựa trên phương pháp LQG. Ý tưởng này là để chọn ma trận thiết kế M sao cho các giá trị kỳ dị $G_{OL} = M(sI - A)^{-1}B$ tốt; ví dụ, giá trị kỳ dị nhỏ nhất lớn tại các tần số thấp, giá trị kỳ dị lớn nhất nhỏ tại các tần số cao, các giá trị kỳ dị đi qua 0 dB với độ dốc -20 dB/dec...

   Sau đó chúng ta giải phương trình CARE với $A, B, M, \Theta = \rho I_{n_y \times n_y}$, sử dụng $\rho > 0$ để điều chỉnh dải thông của vòng lặp cần đạt $L_t = G_{LQG} = G_c(sI - A)^{-1}B$. ρ càng nhỏ (hoặc càng lớn) thì dải thông càng lớn (hoặc càng nhỏ).

- Nguyên tắc để thiết kế vòng lặp cần đạt là $L_t = G_{LQ}$.

  Đẳng thức miền tần số LQ (LQDE) cho thiết kế vòng lặp:

$$[I + G_{LQ}(j\omega)]^H [I + G_{LQ}(j\omega)] = I + \left[\frac{1}{\sqrt{\rho}} G_{OL}(j\omega)\right]^H \left[\frac{1}{\sqrt{\rho}} G_{OL}(j\omega)\right] \tag{30.171}$$

  Từ đó, ta có:

$$\sigma_i[I + G_{LQ}(j\omega)] = \sqrt{1 + \frac{1}{\rho}\sigma_i^2[G_{OL}(j\omega)]} \tag{30.172}$$

  Điều này thừa nhận rằng bằng việc thiết kế G_{OL}, ta có thể thiết kế vòng lặp cần đạt $L_t = G_{LQ}$. Cụ thể là, nếu G_{OL} lớn tại tần số thấp, thì ta cần :

$$G_{LQ}(j\omega) \approx \frac{1}{\sqrt{\rho}} G_{OL}(j\omega) \tag{30.173}$$

  tại tần số thấp. Từ đây cho thấy rằng ma trận M nên được sử dụng để thiết kế vòng lặp cần đạt $L_t = G_{LQ}$ trong khi $\rho > 0$ sử dụng để điều chỉnh dải thông vòng lặp cần đạt – tăng hay giảm ρ dẫn đến giảm hay tăng dải thông đó. Kết quả $L_t = G_{LQ}$ đảm bảo có các đặc tính hệ kín tốt được mô tả dưới đây.

- Sự quan hệ giá trị kỳ dị ở trên dẫn đến:

$$\sigma_{\min}[I + G_{LQ}(j\omega)] \geq 1 \tag{30.174}$$

  với mọi ω. Điều này dẫn đến các giá trị kỳ dị độ nhạy kết hợp thỏa mãn:

$$\sigma_{\max}[S_{LQ}(j\omega)] = \frac{1}{\sigma_{\min}[S_{LQ}(j\omega)^{-1}]} \leq 1 \qquad (0dB) \tag{30.175}$$

  với mọi ω, trong đó:

$$S_{LQ} = [I + G_{LQ}]^{-1} \tag{30.176}$$

- Từ quan hệ giá trị kỳ dị độ nhạy ở trên, ta có giới hạn vòng lặp LQR như sau:
  - Giới hạn hệ số khuếch đại trên là không xác định,
  - Giới hạn hệ số khuếch đại dưới ít nhất là $\frac{1}{2}(6\,dB)$,
  - Giới hạn pha ít nhất $\pm 60^0$.

Các giới hạn nói trên phù hợp với sự thay đổi hệ số khuếch đại đồng thời và độc lập khi ngắt đầu ra. Giống như vậy với giới hạn pha. Các giới hạn ở trên không đảm bảo là thay đổi hệ số và pha đồng thời. Chú ý rằng các giới hạn đó có thể dễ dàng thay đổi bằng cách sử dụng tiêu chuẩn ổn định Nyquist [2, 8].

- Từ quan hệ về giá trị độ nhạy đơn ở trên, ta có mối quan hệ giá trị độ nhạy đơn như sau:

$$\sigma_{\max}[T_{LQ}(j\omega)] = \sigma_{\max}[I - S_{LQ}(j\omega)] \le 1 + \sigma_{\max}[S_{LQ}(j\omega)] \le 2 \qquad (6\text{dB}) \qquad (30.177)$$

với mọi ω, trong đó:

$$T_{LQ} = I - S_{LQ} = G_{LQ}[I + G_{LQ}]^{-1} \qquad (30.178)$$

2. Khôi phục vòng lặp cần đạt L_t sử dụng bộ bù dựa trên mô hình: bước thứ 2 là sử dụng $K_{opt} = [A - BG_c - H_f C, H_f, G_c]$ trong đó H_f được xác định bằng các giải phương trình FARE với $A, L = B, C, \Theta = \rho I_{n_y \times n_y}$ với μ là một giá trị vô hướng dương rất nhỏ. Khi μ nhỏ, chúng ta gọi đây là bài toán cảm biến đắt.

- Nếu mô hình $P = [A, B, C]$ là pha cực tiểu, thì có thể thấy rằng:

$$\lim_{\mu \to 0^+} Y = 0 \qquad (30.179)$$

$$\lim_{\mu \to 0^+} \sqrt{\mu} H_f = BV \qquad (30.180)$$

với các ma trận trực chuẩn W (tức là $W^T W = WW^T = I$):

$$\lim_{\mu \to 0^+} K_{opt} P = L_t \qquad (30.181)$$

Trong trường hợp đó, $K_{opt} \approx L_t$ với μ nhỏ và vì vậy $K_{opt} P$ sẽ có giới hạn bền vững gần với giới hạn bền vững của L_t (tại đầu vào mô hình) – bất kỳ phương pháp nào được sử dụng để thiết kế L_t.

Phải chú ý rằng điều kiện pha cực tiểu trên mô hình P là điều kiện đủ. Nó là không cần thiết. Hơn nữa, H_f không cần tính bằng cách sử dụng FARE. Trong thực tế, (1) H_f thỏa mãn điều kiện giới hạn $\lim_{\mu \to 0^+} \sqrt{\mu} H_f = BV$ với một số ma trận khả đảo V và (2) chắc chắn rằng $A - H_f C$ là ổn định (với μ nhỏ), sẽ dẫn đến LTR tại đầu vào đối tượng, tức là $\lim K_{opt} P = L_t$. Kết quả này là hệ quả của cấu trúc các bộ bù dựa trên mô hình và không còn gì để làm với các bài toán điều khiển tối ưu và bộ lọc.

- Giả thiết rằng điều kiện giới hạn $\lim_{\mu \to 0^+} \sqrt{\mu} H_f = BV$ giữ các ma trận nghịch đảo V, sự khôi phục truyền lặp của vòng lặp đích $L_t = G_c(sI - A)^{-1} H_f$ có thể được chứng minh như ở dưới. Với μ chúng ta có:

$$H_f \approx \frac{BV}{\sqrt{\mu}} \qquad (30.182)$$

từ đó dẫn đến:

$$K_{opt} P = G_c(sI - A + BG_c + H_f C)^{-1} H_f P \qquad (30.183)$$

$$\approx G\left(sI - A + BG_c + \frac{BV}{\sqrt{\mu}} C\right)^{-1} \frac{BV}{\sqrt{\mu}} p \qquad (30.184)$$

$$\approx G_c\left(sI - A + \frac{BV}{\sqrt{\mu}}C\right)^{-1}\frac{BV}{\sqrt{\mu}}p \tag{30.185}$$

$$\approx G_c(sI-A)^{-1}_{\cdot}+\left[I+\frac{BV}{\sqrt{\mu}}C(sI-A)^{-1}\right]^{-1}\frac{BV}{\sqrt{\mu}}p \tag{30.186}$$

$$\approx G_c(sI-A)^{-1}\frac{BV}{\sqrt{\mu}}\left[I+C(sI-A)^{-1}\frac{BV}{\sqrt{\mu}}\right]^{-1}p \tag{30.187}$$

$$\approx G_c(sI-A)^{-1}\frac{BV}{\sqrt{\mu}}\left[C(sI-A)^{-1}\frac{BV}{\sqrt{\mu}}\right]^{-1}P \tag{30.188}$$

$$\approx G_c(sI-A)^{-1}B\frac{V}{\sqrt{\mu}}\left[P\frac{V}{\sqrt{\mu}}\right]^{-1}P \tag{30.189}$$

$$\approx G_c(sI-A)^{-1}B = L_t \tag{30.190}$$

Ý tưởng chính (dưới dạng đại số) là μ tiến tới 0, đường phản hồi B trong bộ bù $K_{opt} = [A - BG_c - H_fC, G_c, H_f]$ được ngắt (hình 30.7) và các đặc tính tốt để giữ tại cải mới $\hat{u}$ (chẳng hạn, ma trận truyền lặp tại $\hat{u}$ là $L_t = G(sI-A)^{-1}B$) trong hình 30.7 truyền tới đầu vào đối tượng u (đầu vào bù) trong hệ phản hồi.

Chú thích 30.13 (Các giới hạn ổn định và đỉnh độ nhạy)

Đỉnh trên đồ thị độ nhạy rất quan trọng trong thiết kế hệ thống phản hồi. Ví dụ, đỉnh lớn có thể do điểm cực hệ kín nằm gần trục ảo. Điều này rõ ràng là không mong muốn. Vì vậy chúng ta muốn đỉnh "nhỏ". Có thể thấy rằng đỉnh cần thiết lập giới hạn pha và hệ số khuếch đại.

Giả sử rằng độ nhạy tại đỉnh là được giới hạn bởi $\alpha \geq 1$; tức là $\sigma_{max}S(j\omega) < \alpha$ với mọi ω. Có thể thấy rằng khi đó vòng phản hồi có các giới hạn (pha và hệ số khuếch đại) bền vững ổn định đa biến như sau:

$$\uparrow GM > \frac{\alpha}{\alpha - 1} \tag{30.191}$$

$$\downarrow GM < \frac{\alpha}{\alpha + 1} \tag{30.192}$$

$$|PM| > 2\sin^{-1}\left(\frac{1}{2\alpha}\right) \tag{30.193}$$

Các giới hạn trên có thể dễ dành đạt được sử dụng tiên chuẩn Nyquist SISO [2, 8] như sau.

Nếu:

$$|S(j\omega)| < \alpha \tag{30.194}$$

với mọi ω, thì

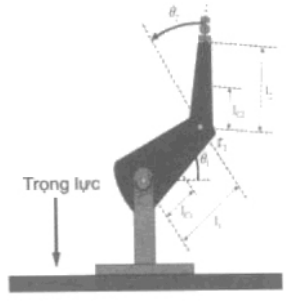
$$\frac{1}{\alpha} < |1 + L(j\omega)| \tag{30.195}$$

với mọi ω. Điều này dẫn đến đồ thị Nyquist kết hợp với L không thể đi xuyên qua đường tròn tâm -1 và bán kính $1/\alpha$, điểm xa nhất bên trái là $-[(\alpha+1)/\alpha]$, xa nhất bên phải là $-[(\alpha-1)/\alpha]$. Giới hạn trên của hệ số khuếch đại bắt đầu từ điểm phải nhất của đường tròn. Giới hạn dưới của hệ số khuếch đại bắt đầut điểm trái nhất của đường tròn. Giới hạn pha có thể đạt được với hình học nhỏ.

Ví dụ sau đây xét ứng dụng của lý thuyết $\mathcal{H}^2$ cho một tay máy rôbốt.

Ví dụ 30.5 (Thiết kế điều khiển TQG/LTR cho tay máy rôbốt 560 PUMA)

Trong ví dụ này, chúng ta trình bày tối ưu $\mathcal{H}^2$ được sử dụng để thiết kế một bộ điều khiển LQG/LTR cho một cánh tay rôbốt 560 như thế nào. Cánh tay rôbốt được minh họa như hình 30.8.



HÌNH 30.8 Cánh tay rôbốt PUMA 560 hai bậc tự do

Mô hình tuyến tính 2 bậc tự do $P=[A_p,B_p,C_p]$ được dung để bắt đầu quá trình thiết kế. Việc tuyến tính hóa mô hình phi tuyến PUMA [9] ở điểm cân bằng $\theta_1=90^0\ \theta_2=0^0$ (cả 2 khớp thẳng đứng), dẫn đến mô hình tuyến tính sau:

$$\dot{x}_p = A_p x_p + B_p \mu_p \tag{30.196}$$

$$y_p = C_p x_p \tag{30.197}$$

$$u_p = [\tau_1 \quad \tau_2]^T \tag{30.198}$$

$$x_p = [\theta_1 \quad \theta_2 \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T \tag{30.199}$$

$$y_p = [\theta_1 \quad \theta_2]^T \tag{30.200}$$

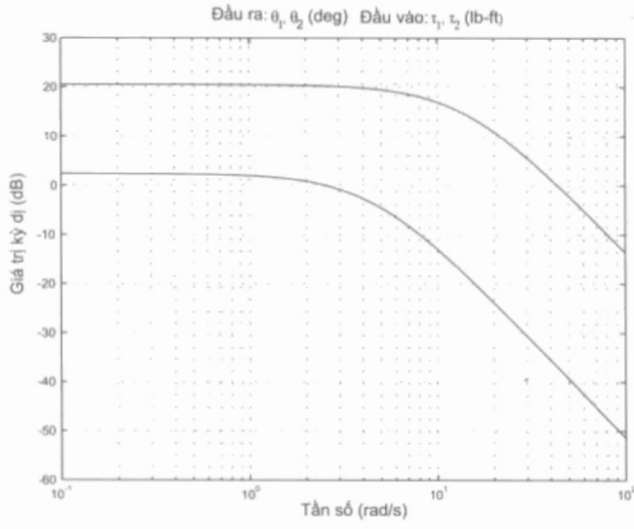
trong đó

$$A_p = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \\ 31.7613 & -33.0086 & 0.0000 & 0.0000 \\ -56.9381 & 187.7089 & 0.0000 & 0.0000 \end{bmatrix} \tag{30.201}$$

$$B_p = \begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \\ 1037.7259 & -3919.6674 \\ -3919.6674 & 2030.8306 \end{bmatrix} \tag{30.202}$$

$$C_p = [I_{2\times 2} \quad 0_{2\times 2}] \tag{30.203}$$

Các điểm cực của hệ thống là $s = \pm 14.1050, s = \pm 4.5299$. Phân tích vectơ riêng cho thấy tính không ổn định nhanh tại s = 14.1050 được kết hợp chủ yếu với khớp trên (khớp ngắn hơn), tính không ổn định chậm hơn được kết hợp chủ yếu với khớp thấp hơn (dài hơn). Hệ thống không có bộ tích phân nào (tức là không có trị riêng nào bằng không) và đúng như mong đợi các giá trị kỳ dị $\sigma_i[P(j\omega)]$ không thay đổi tại tần số thấp (xem hình 30.9).



HÌNH 30.9 Các giá trị kỳ dị của cánh tay rôbốt PUMA 560

Mục tiêu hệ kín

Một bộ điều khiển thực hiện trong một vòng phản hồi âm được xác định. Hệ phản hồi âm đó phải thỏa mãn các đặc tính sau: (1) tính ổn định vòng kín, (2) sai số giữa trạng thái ổn định với các lệnh tham chiếu bước (bước nhảy) bằng 0, (3) lệnh tham chiếu tần số thấp tốt sinh ra (đáp ứng bước nhảy với độ quá điều chỉnh nhỏ 3 s), (4) sự suy giảm nhiễu đầu vào ở tần số thấp tốt, (5) sự suy giảm nhiễu ở tần số cao tốt, (6) giới hạn bền vững ổn định tần số thấp ở đầu ra mô hình tốt.

Mỗi bước của quá trình thiết kế hệ thống điều khiển này sẽ được mô tả sau đây. Ý tưởng chính là hình thành một mô hình thiết kế P_d từ mô hình thật P. Mô hình thiết kế P_d là mô hình được đưa tới bộ máy thiết kế $\mathcal{H}^2$ – LQG/LTR của chúng ta.

Bước 1: Thêm vào mô hình P với các bộ tích phân để được mô hình thiết kế $P_d = [A,B,C]$

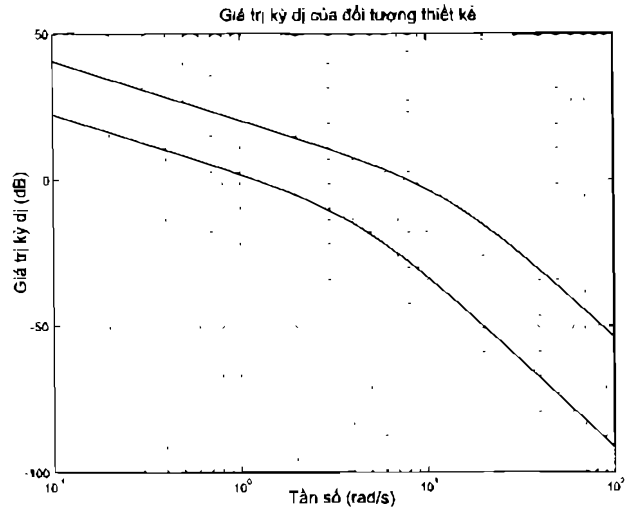
Để đạt được sai số giữa trạng thái ổn định với tín hiệu tham chiếu bước bằng 0, chúng ta bắt đầu bằng việc thêm vào mô hình $P = \left[\mathrm{A_p},\ \mathrm{B_p},\ \mathrm{C_p}\right]$ các bộ tích phân - mỗi bộ trên một kênh điều khiển - tạo thành mô hình thiết kế $P_d = [\mathrm{A},\ \mathrm{B},\ \mathrm{C}]$; tức là $P_d = P(I_{2X2}/s)$. Điều này được thực hiện như sau:

$$A = \begin{bmatrix} 0_{2\times 2} & 0_{2\times 4} \\ B_p & A_p \end{bmatrix} \tag{30.204}$$

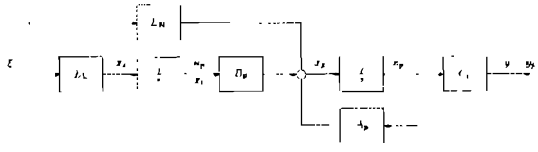
$$B = \begin{bmatrix} 1_{2\times 2} \\ 0_{4\times 2} \end{bmatrix} \tag{30.205}$$

$$C = \begin{bmatrix} 0_{2\times 2} & C_p \end{bmatrix} \tag{30.206}$$

Trạng thái của hệ thống này là $x = \begin{bmatrix} x_i \\ x_p \end{bmatrix}$ trong đó x_i là trạng thái tích phân và x_p là trạng thái mô hình. Các giá trị kỳ dị của mô hình tăng P_d có độ dốc -20 dB/dec tại tần số thấp như ta mong muốn (hình 30.10). Giá trị kỳ dị nhỏ nhất đi qua dB 0 tại 1 rad/s. Giá trị kỳ dị lớn nhất đi qua dB tại 8 rad/s.



HÌNH 30.10 Các giá trị kỳ dị của mô hình thiết kế cánh tay rôbốt PUMA 560



HÌNH 30.11 Hệ được ghép thêm sử dụng để thiết kế vòng lặp cần đạt

Bước 2: Thiết kế ma trận hàm truyền vòng lặp cần đạt $L_o = G_{KF} = C(sI - A)^{-1}H_f$

Tiếp theo chúng ta thiết kế một ma trận hàm truyền vòng lặp cần đạt $L_o = G_{KF} = C(sI - A)^{-1}H_f$ để có các đặc tính của hệ kín mong muốn (ví dụ, các giá trị kỳ dị nhạy, vị trí các điểm cực, giới hạn ổn định...) tại đầu ra. Để làm được điều này chúng ta sử dụng các ý tưởng của bộ lọc Kalman. Như vòng lặp LQR được thiết kế không cần phạt quan hệ chéo các trạng thái điều khiển, vòng lặp lọc Kalman được thiết kế tương tự như trình bày trong giới hạn bền vững ổn định mong muốn (giới hạn khuếch đại trên vô hạn, giới hạn khuếch đại dưới ít nhất là 6 dB, giới hạn pha ít nhất là $\pm 60^0$). Thiết kế vòng lặp lý tưởng này được thực hiện như sau:

Xét hệ đã được ghép thêm như hình 30.11

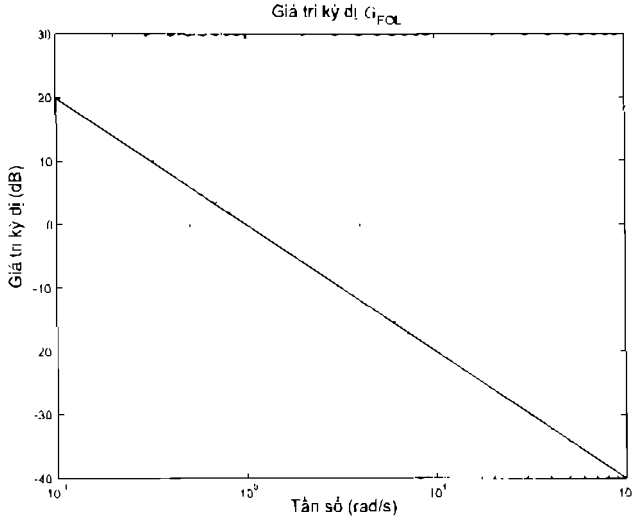
Nó được dung để để thiết kế ma trận hàm truyền mục tiêu $L_o = G_{KF}$ với các đặc tính hệ kín mong muốn tại đầu ra. Để làm được như vậy, ta bắt đầu thiết lập một hệ được ghép thêm $G_{FOL} = C(sI - A)^{-1}L$ với:

$$L = \begin{bmatrix} L_L \\ L_H \end{bmatrix} \tag{30.207}$$

$$L_I = [C_p(-A_p)^{-1}B_p]^{-1} \tag{30.208}$$

$$L_H = (-A_p)^{-1}B_pL_I \tag{30.209}$$

Ma trận L_I kết hợp với các giá trị kỳ dị của $G_{FOL} = C(sI - A)^{-1}L$ tại tần số thấp. Ma trận L_H kết hợp với các giá trị kỳ dị đó tại tần số cao. Kết hợp, L_L và L_H làm cho phù hợp với các giá trị kỳ dị của $G_{FOL} = C(sI - A)^{-1}L$ trong toàn miền tần số (xem hình 30.12). Tại sao như vậy?



HÌNH 30.12 Các giá trị kỳ dị G_{FOL} cánh tay rôbốt PUMA 560

Việc lựa chọn L_L và L_H này dẫn đến:

$$G_{FOL} = C_p(sI - A_p)^{-1}L_H + C_p(sI - A_p)^{-1}B_p\left(\frac{I}{s}\right)L_L \tag{30.210}$$

$$= C_p(sI - A_p)^{-1}\left[L_H + B_p\left(\frac{I}{s}\right)L_I\right] \tag{30.211}$$

$$= C_p(sI - A_p)^{-1}\left[(-A_p)^{-1}B_pL_I + B_p\left(\frac{I}{s}\right)L_I\right] \tag{30.212}$$

$$= C_p(-A_p)^{-1}B_pL_I\left(\frac{I}{s}\right) \tag{30.214}$$

$$= \frac{I}{s} \tag{30.215}$$

Tần số cắt hệ số khuếch đại thu được trong hình 30.12 là 1 rad/s, như mong muốn.

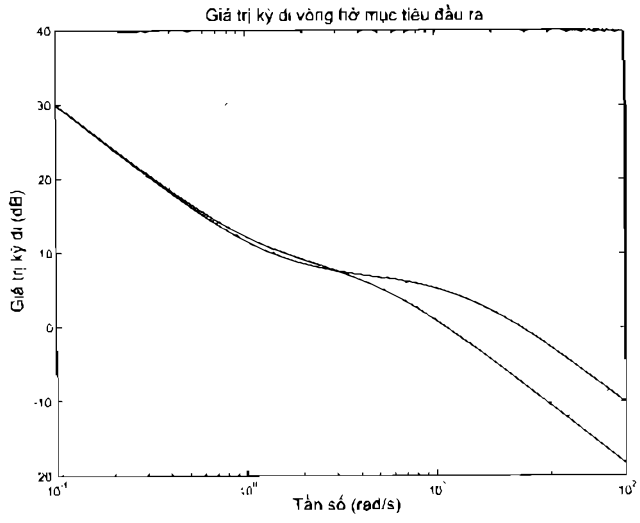
Tại sao lại kết hợp các giá trị kỳ dị của G_{FOL} theo cách này? Từ đẳng thức Kalman trong miền tần số (FKDE), ta có:

$$\sigma_i[I + G_{FOL}(j\omega)] = \sqrt{1 + \frac{1}{\mu}\sigma_i^2[G_{FOL}(j\omega)]} \tag{30.216}$$

Điều này gợi ý rằng bằng cách vẽ hình dạng G_{LOI}, chúng ta có thể định hình được mục tiêu $L_o = G_{KI}$. Đặc biệt là, nếu G_{LOI} lớn tại tần số thấp, thì chúng ta cần (từ đẳng thức miền tần số Kalman):

$$L_o(j\omega) = G_{KI}(j\omega) \approx \frac{1}{\sqrt{\mu}} G_{LOI}(j\omega) \tag{30.217}$$

tại tần số thấp. Điều này cho thấy rằng ma trận L được sử dụng để định hình thiết kế vòng lặp cần đạt $L_o = G_{KI}$ trong đó $\mu > 0$ được sử dụng để chỉnh dải thông của vòng lặp cần đạt – giảm/tăng μ sẽ tăng/giảm dải thông của vòng lặp cần đạt.



HÌNH 30.13 Các giá trị kỳ dị G_{KI} vòng lặp mục tiêu của cánh tay rôbốt PUMA 560

Chú ý rằng bằng cách chọn L, chúng ta đã làm tất cả các chế độ không ổn định của mô hình không thể điều khiển được bằng L. Vì vậy, (A, L) là không điều khiển được. Trong khi điều này xuất hiện một số bất lợi. Làm thế nào để ma trận Hamiltonian kết hợp phụ thuộc vào *dom* (Ric) sao cho tồn tại H_f ổn định. Một điều kiện cần và đủ cho điều này là (A, C) có thể xác định được và (A, L) có chế độ quan sát được trên trục ảo. Khi mỗi lệnh được thỏa mãn, chúng ta có thể sử dụng lệnh "are" để tìm nghiệm ổn định của FARE.

Tiếp theo chúng ta giải FARE với $\Theta = \mu I_{2\times 2} (\mu = 0.1)$:

$$AY + YA^T + LL^T - YC^T\Theta^{-1}CY = 0 \tag{30.218}$$

với $Y \geq 0$. Lệnh "are" được sử dụng để làm điều này, khi đó nó mang lại một nghiệm ổn định (tồn tại duy nhất một nghiệm). Sau đây chúng ta thành lập ma trận hệ số khuếch đại bộ lọc.

$$H_f = YC^T\Theta^{-1} \tag{30.219}$$

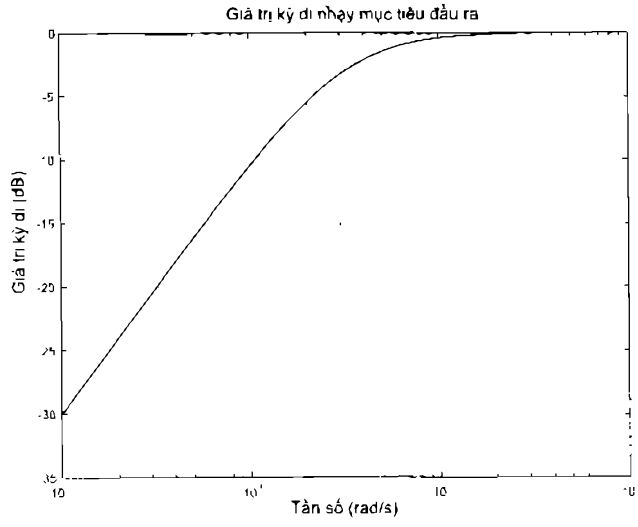
$$= \begin{bmatrix} 2.3635 & 0.0384 \\ 0.40585 & 0.3091 \\ 13.1371 & -4.2300 \\ -4.2300 & 30.4572 \\ 90.2377 & -83.4384 \\ -100.9668 & 467.7679 \end{bmatrix} \tag{30.220}$$

Dẫn đến các điểm cực vòng kín cần đạt như sau $(\lambda_i(A - H_f C))$:

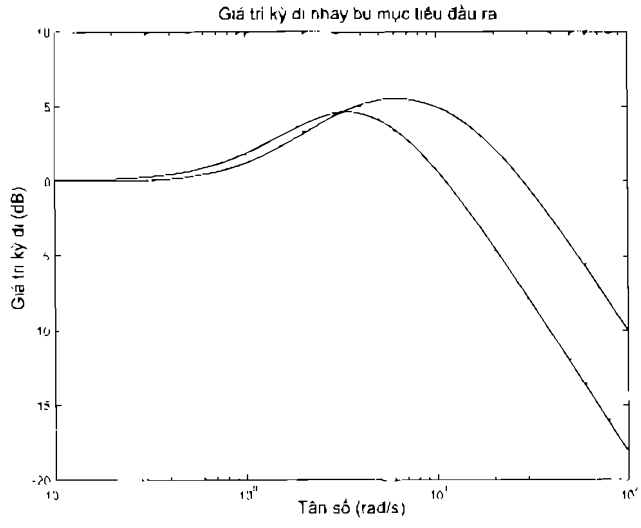
$$s = -3.1623,\ -3.1623,\ -4.5299,\ -4.5299,\ -14.1050,\ -14.1050 \qquad (30.221)$$

Các giá trị kỳ dị đối với ma trận truyền lặp mục tiêu thu được $L_o = G_{KF} = C(sI - A)^{-1} H_f$ như hình 30.13.

Các giá trị kỳ dị vòng lặp cần đạt – như mong muốn từ KFDE – được kết hợp ở tần số thấp với độ dốc -20 dB/dec. Chúng giữ lại khoảng 1 rad/s, sau đó tách ra. Đó là điều có thể thấy do $G_{FOL} = 1/s$ không phải là một vòng lặp có thể thực hiện được (không có tính ổn định của hệ kín). Ma trận hệ số khuếch đại bộ lọc thu được tạo ra dải thông cần thiết để làm ổn định cánh tay rôbốt không ổn định, với các tính ổn định vòng hở tại $s = 14.1050, 4.5299$. Một giá trị kỳ dị đi qua 0 dB trên 10 rad/s, một giá trị kỳ dị khác khoảng 30 rad/s. μ được sử dụng để điều chỉnh dải thông.



HÌNH 30.14 Các giá trị kỳ dị của độ nhạy mục tiêu $S_{KF} = [I + G_{KF}]^{-1}$ của cánh tay rôbốt PUMA 560



HÌNH 30.15 Các giá trị kỳ dị của độ nhạy bù mục tiêu $T_{KI} = G_{KI}[I + G_{KI}]^{-1}$ của cánh tay rôbốt

Các giá trị kỳ dị $S_{KJ} = [I + G_{kJ}]^{-1}$ của độ nhạy mục tiêu tương ứng và các giá trị kỳ dị của độ nhạy bù $T_{kJ} = G_{KJ}[I + G_{kJ}]^{-1}$ theo thứ tự như hình 30.14 và 30.15. Các giá trị kỳ dị của độ nhạy bù và độ nhạy kết hợp cần vòng lặp cần đạt có:

- Các đặc tính điều khiển theo lệnh ở tần số thấp tốt.
- Các đặc tính giảm thiểu nhiễu ở tần số thấp tốt.
- Các đặc tính giảm thiểu nhiễu cảm biến ở tần số cao, và
- Giới hạn ổn định MIMO tốt (giới hạn khuếch đại tăng gần vô hạn, giới hạn khuếch đại giảm ít nhất là 6 dB, và giới hạn pha ít nhất là $\pm 60^0$) tại đầu ra.

Các giá trị kỳ dị của độ nhạy bù cho thấy rằng một bộ lọc tiền xử lý W sẽ làm giảm độ quá điều chỉnh nhờ vào lệnh tham chiếu. Việc thiết kế bộ lọc đó sẽ được nghiên cứu dưới đây.

Bước 3: Giải bài toán điều khiển "rẻ" để khôi phục vòng lặp cần đạt tại đầu ra đối tượng

Tiếp theo chúng ta giải "bài toán điều khiển LQR rẻ" để có được ma trận hệ số khuếch đại điều khiển G_c sao cho bộ bù dựa trên mô hình tối ưu $\mathcal{H}^2$ $K_d = [A\text{-}BG_c\text{-}H_fC, H_f, G_c]$ với $P_d = [A, B, C]$ xấp xỉ ("tái tạo") ma trận hàm truyền vòng lặp mục tiêu $L_a = G_{KJ}$; tức là:

$$P_d K_d \approx L_a = G_{KJ} \tag{30.222}$$

Điều này đã được thực hiện bằng cách giải phương trình Riccati đại số điều khiển sau (sử dụng lệnh "lqr") với $R = \rho I_{2\times 2}(\rho = 10^{-13})$:

$$XA + A^T X + C^T C - XBR^{-1}B^T X = 0 \tag{30.223}$$

với $X > 0$ và dạng ma trận khuếch đại điều khiển:

$$G_c = R^{-1}B^T X \tag{30.224}$$

$$= \begin{bmatrix} 987.9832 & -543.0034 & 3162945.2928 & 56.9921 & 13941.9005 & 2069.8324 \\ -543.0034 & 3657.5891 & 11.7867 & 3162634.3919 & 2069.7987 & 3765.7978 \end{bmatrix} \tag{30.225}$$

Để có được kết quả đó các điểm cực điều chỉnh vòng kín như sau $(\lambda_i(A - BG_c))$:

$$s = -440.8808,\ -220.4404 \pm j381.7871,\ -1881.9053,\ -940.9527 \pm j1629.7168 \tag{30.226}$$

Có tất cả các thừa số suy giảm lớn hơn hoặc bằng $\xi = 0.5$. Như một chú ý trong thực tế để bộ điều khiển dễ dàng thực hiện trong thời gian thực, người ta có thể sử dụng các kỹ thuật lược bỏ mô hình [10] để bỏ đi các điểm cực ở tần số rất cao trong bộ bù. Làm được như vậy sẽ cho phép sử dụng bước tích phân lớn hơn trong hệ nhúng thời gian thực hoặc trong hệ thực hiện bằng vi xử lý.

Bước 4: Xây dựng bộ điều khiển K cuối cùng

Tiếp theo chúng ta thiết lập bộ điều khiển cuối cùng như sau:

$$K = \frac{K_d}{s} \tag{30.227}$$

$$= \frac{[A - BG_c - H_fC,\ \ H_f,\ \ G_c]}{s} \tag{30.228}$$

$$= [A_k,\ \ B_k,\ \ C_k] \tag{30.229}$$

Biểu diễn bộ điều khiển này dưới dạng không gian trạng thái là:

$$A_K = \begin{bmatrix} 0_{2\times 2} & G_c \\ 0_{6\times 2} & A - BG_c - H_f C \end{bmatrix}, \qquad B_K = \begin{bmatrix} 0_{2\times 2} \\ H_f \end{bmatrix} \tag{30.230}$$

$$C_K = \begin{bmatrix} I_{2\times 2} & 0_{2\times 6} \end{bmatrix} \tag{30.231}$$

Với cách chọn này, ta có:

$$PK = P\frac{K_d}{s} \tag{30.232}$$

$$= P\frac{I_{2\times 2}}{s}K_d \tag{30.233}$$

$$= P_d K_d \tag{30.234}$$

$$\approx L_o = G_{KF} \tag{30.235}$$

Thông qua việc lựa chọn K, chúng ta đã khôi phục ma trận hàm truyền vòng lặp cần đạt $L_o = G_{CK}$. Như vậy, K có chứa P nghịch đảo (từ về phải) trong thứ tự thiết kế $PK \approx L_o = G_{KF}$. Kiểm tra giá trị kỳ dị cho vòng lặp thực PK thấy rằng giá trị kỳ dị của vòng lặp thực phù hợp với giá trị kỳ dị mục tiêu tăng và lớn hơn 100 rad/s.

Khôi phục truyền lặp. Tại sao chúng ta có thể khôi phục được vòng lặp cần đạt? Việc khôi phục đã được cho phép bởi cấu trúc dựa trên mô hình của bộ bù K, các phương trình Riccati dùng để có được các ma trận G_c và H_f, và thực tế là mô hình $P = \begin{bmatrix} A_p, & B_p, & C_p \end{bmatrix}$ (và mô hình thiết kế $P_d = [A, \quad B, \quad C]$=P(l/s)) là pha cực tiểu. Điều kiện pha cực tiểu, nói một cách cụ thể, là một điều kiện cần để đảm bảo rằng tồn tại một ma trận trực chuẩn $U(U^TU = UU^T = I)$ tức là:

$$\lim_{\rho\to 0^+} \sqrt{\rho} G_c = UC \tag{30.236}$$

Giới hạn này liên quan với ma trận khuếch đại điều khiển và ma trận C của mô hình, tuy nhiên, có thể được sử để chứng tỏ rằng việc khôi phục ma trận truyền lặp có thể thực hiện được, tức là:

$$\lim_{\rho\to 0^+} P_d K_d = \lim_{\rho\to 0^+} PK = L_o = G_{KF} \tag{30.237}$$

Bước 5: Lệnh thiết kế bộ lọc tiền xử lý W

Lệnh MATLAB:

$$t_0 = (a - b*g - h*c, h, g) \tag{30.238}$$

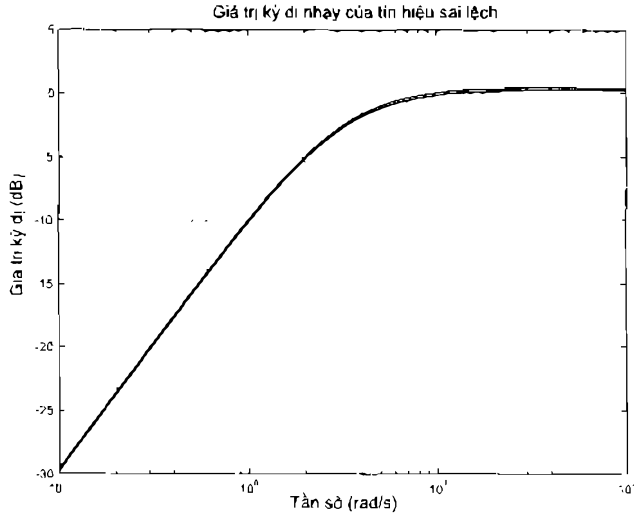
được sử dụng để tìm các điểm không của bộ bù. Cũng là các điểm không của ma trận hàm truyền hệ kín từ r đến y. Bộ bù cuối cùng (cũng là vòng lặp cần đạt G_{KF}) có các điểm không gần $s \approx -1.2$. Nhờ đó, bộ lọc tiền xử lý

$$W = \frac{1.2}{s+1.2} I_{2\times 2} \tag{30.239}$$

được thêm vào ngoài vòng lặp để lọc đầu vào tham chiếu. Bằng cách đó, chúng ta chắc chắn rằng đầu vào tham chiếu bước nhảy cho θ_1 và θ_2 là trạng thái ổn định (nhờ có bộ tích phân trong bộ điều khiển) không vượt quá độ quá điều chỉnh trong thời gian ngắn.

Đáp ứng tần số độ nhạy

Kết quả các giá trị kỳ dị độ nhạy được vẽ như hình 30.16. Đồ thị cho thấy rằng lệnh tham chiếu ở tần số thấp r sẽ được bám sát và các nhiễu đầu ra ở tần số thấp d_o sẽ bị suy giảm. Chính xác hơn, lệnh tham chiếu r với phổ tần dưới 0.3 rad/s được bám trong khoảng 20 dB; tức là, với một sai lệch trạng thái ổn định khoảng 10%. Tương tự như vậy, nhiễu đầu ra d_o với phổ tần dưới 0.3 rad/s sẽ được suy giảm trong khoảng 20 dB.



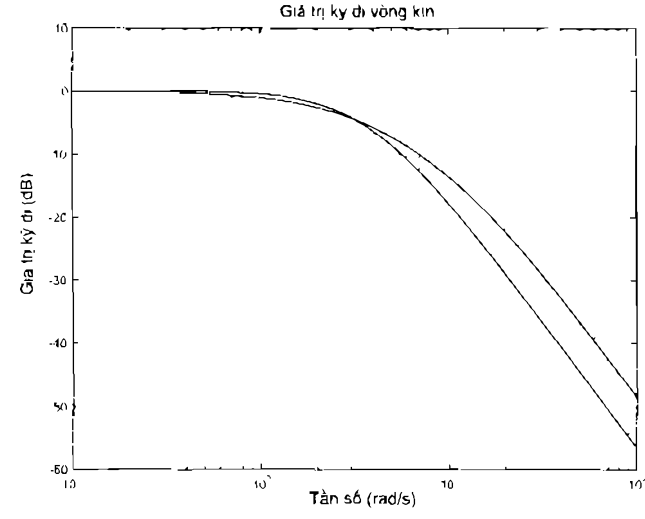
HÌNH 30.16 Đáp ứng tần số độ nhạy PUMA 560 của tín hiệu sai lệch

Tham chiếu tới đáp ứng tần số đầu ra

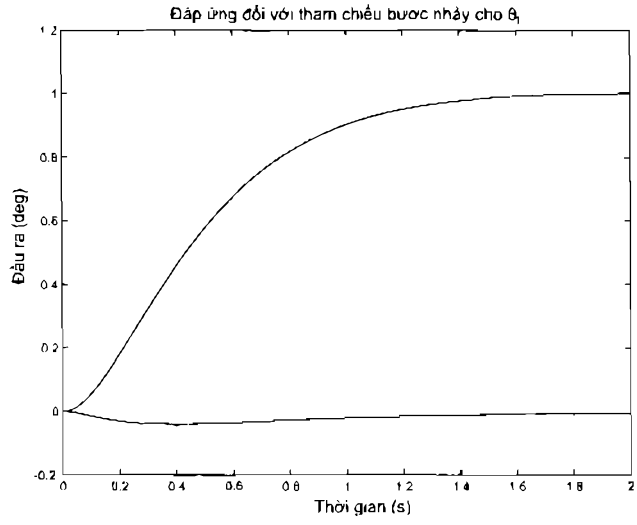
Ma trận hàm truyền từ lệnh tham chiếu r tới góc liên kết y là:

$$T_{ry} = [I + PK]^{-1} PKW \tag{30.240}$$

Giá trị kỳ dị của nó được vẽ như Hình 30.17. Đồ thị này cho thấy lệnh tham chiếu ở tần số thấp sẽ được bám trong trạng thái ổn định và độ quá điều chỉnh nhỏ sẽ đặt được trong một khoảng thời gian ngắn.



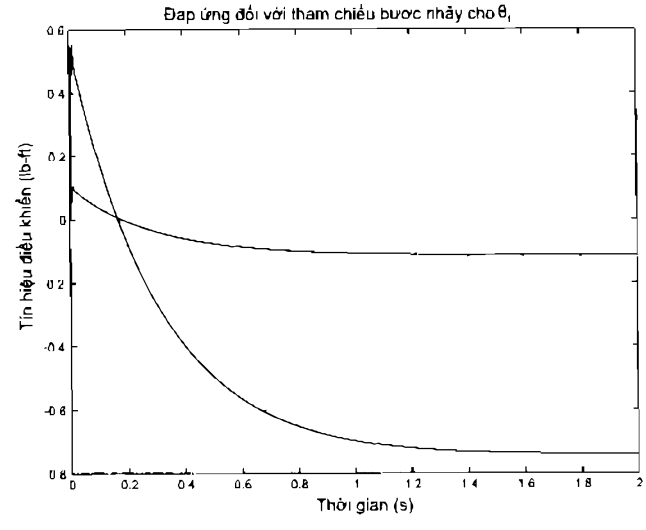
HÌNH 30.17 Tham chiếu với đáp ứng tần số đầu ra PUMA 560



HÌNH 30.18 Đầu ra PUMA 560: đáp ứng của lệnh tham chiếu θ_1

Đáp ứng lệnh tham chiếu bước nhảy θ_1

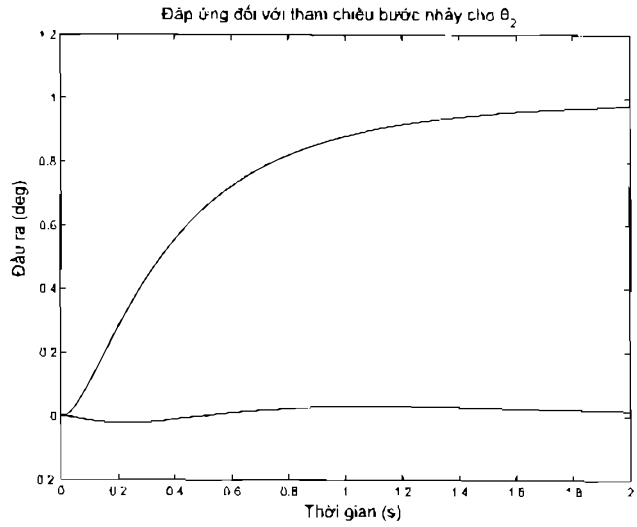
Đáp ứng lệnh bước đơn vị (unit bước nhảy) với θ_1 được vẽ như hình 30.18. Như mong đợi, θ_1 bám lệnh bước nhảy tốt, không có độ quá điều chỉnh và thời gian quá độ khoảng 1.6s. Đáp ứng kết hợp θ_2 nhỏ, chỉ ra một sự móc nối chéo trong hệ kín cuối cùng. Các tín hiệu điều khiển tương ứng được vẽ như hình 30.19 với kích thước chấp nhận được.



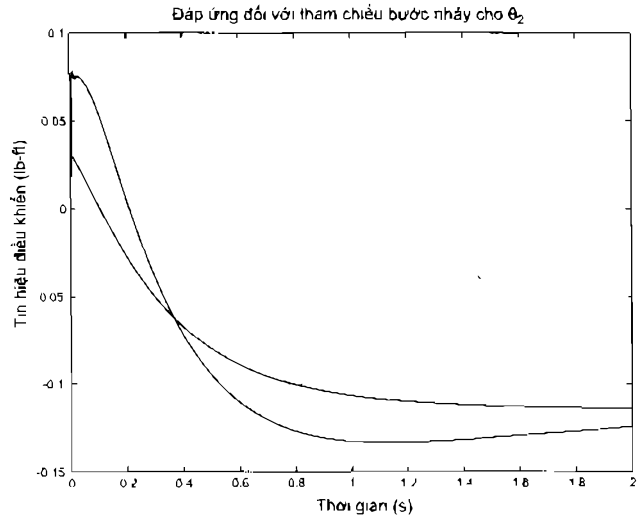
HÌNH 30.19 Điều khiển PUMA 560: đáp ứng của lệnh tham chiếu θ_1

Đáp ứng lệnh tham chiếu bước nhảy θ_2

Đáp ứng lệnh bước nhảy đơn vị với θ_2 được vẽ như hình 30.20. Như mong đợi, θ_2 bám lệnh bước nhảy tốt, không có độ quá điều chỉnh và thời gian quá độ khoảng 3s. Đáp ứng kết hợp θ_1 nhỏ, chỉ ra một sự móc nối chéo trong hệ kín cuối cùng. Tín hiệu điều khiển tương ứng được vẽ như hình 30.21 với kích thước chấp nhận được.



HÌNH 30.20 Đầu ra PUMA 560 đáp ứng của lệnh tham chiếu θ_2



HÌNH 30.21 Điều khiển PUMA 560: đáp ứng của lệnh tham chiếu θ_2

30.4 Bài toán phản hồi trạng thái $\mathcal{H}^2$

Phần này đưa ra các phương pháp đã trình bày trong bài toán phản hồi đầu ra cũng có thể sẵn sàng tiếp tục cho phép thiết kế các luật điều khiển phản hồi trạng thái hệ số hằng tốt ưu $\mathcal{H}^2$ (các ma trận hệ số hằng G_i).

Cấu trúc mô hình tổng quát của bài toán phản hồi trạng thái

Trong trường hợp này, mô hình tổng quát G (gồm mô hình P và các hàm trọng) có dạng sau:

$$G = \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right] = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0_{n_z \times n_w} & D_{12} \\ I_{n\times n} & 0_{n_y \times n_w} & 0_{n_y \times n_u} \end{array}\right] = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \tag{30.241}$$

Từ đây dẫn đến các tín hiệu đo y là các trạng thái x của mô hình tổng quát G. Như vậy, tất cả các dạng của A đều quan sát được nhờ $C_2 = I_{nxn}$.

Các giả thiết phản hồi trạng thái

Các giả thiết phản hồi trạng thái chuẩn là tập con của các giả thiết trong bài toán phản hồi đầu ra. Các giả thiết phản hồi trạng thái là như sau:

Giả thiết 30.2 (Bài toán phản hồi trạng thái $\mathcal{H}^2$)

Trong suốt mục này, chúng ta giả thiết như sau:

Giả thiết mô hình G_{22}. (A, B_2) ổn định

Giả thiết hàm trọng điều khiển không đơn. $R = D_{12}^T D_{12} > 0$ (D_{12} có hạng cột đầy đủ).

Giả thiết bộ chỉnh định. $\begin{bmatrix} j\omega I - A & -B_2 \\ C_2 & D_{12} \end{bmatrix}$ có hạng cột đầy đủ với mọi ω.

Nên chú ý rằng nếu $D_{12}^T C_1 = 0$, thì (3) tương đương với (A, C_1) không có các dạng ảo không quan sát được. Nếu (A, C_1) là có thể xác định được, thì điều này được thỏa mãn.

Luật điều khiển phản hồi trạng thái tối ưu $\mathcal{H}^2$

Bộ điều khiển tối ưu $\mathcal{H}^2$ như sau

$$K_{opt} = -G_c \tag{30.242}$$

trong đó ma trận hệ số điều khiển $G_c = R^{n_u \times n}$ được cho bởi

$$G_c = R^{-1}[B_2^T X + D_{12}^T C_1] \tag{30.243}$$

trong đó $X \geq 0$ là nghiệm bán xác định dương duy nhất của CARE

$$(A - B_2 R^{-1} D_{12}^T C_1)^T X + X(A - B_2 R^{-1} D_{12}^T C_1) + C_1^T (I - D_{12}^T R^{-1} D_{12}^T) C_1 - X B_2 R^{-1} B_2^T X = 0 \tag{30.244}$$

Các điểm cực hệ kín có được từ luật điều khiển phản hồi trạng thái hệ số hằng ở trên là trị riêng của $A - B_2 G_c$. Chuẩn hệ kín nhỏ nhất là:

$$\min_K \|T_{wz}\|_{\mathcal{H}^2} = \sqrt{trace(B_1^T X B_1)} \tag{30.245}$$

Thiết kế vòng phản hồi trạng thái

Nếu chọn

$$B_2 = B \tag{30.246}$$

$$C_1 = \begin{bmatrix} M \\ 0_{n_u \times n} \end{bmatrix} \tag{30.247}$$

$$D_{12} = \begin{bmatrix} 0_{n_y \times n} \\ \sqrt{\rho} I_{n_u \times n_u} \end{bmatrix} \tag{30.248}$$

$$R = \rho I_{n_u \times n_u} \tag{30.249}$$

thì $D_{12}^T C_1 = 0$ và ta có

$$G_c = \frac{1}{\rho} B_2^T X \tag{30.250}$$

trong đó $X \geq 0$ là nghiệm bán xác định dương duy nhất của CARE

$$A^T X + XA + M^T M - XB\frac{1}{\rho} B^T X = 0 \tag{30.251}$$

LQFDE sau đây có thể nhận được từ CARE

$$[I + G_{LQ}(j\omega)]^H [I + G_{LQ}(j\omega)] = I + \left[\frac{1}{\sqrt{\rho}} G_{OL}(j\omega)\right]^H \left[\frac{1}{\sqrt{\rho}} G_{OL}(j\omega)\right] \tag{30.252}$$

trong đó:

$$G_{OL} = M(sI - A)^{-1} B \tag{30.253}$$

$$G_{LQ} = G_c(sI - A)^{-1} B \tag{30.254}$$

Các ý tưởng thiết kế vòng lặp đã trình bày trước đây có thể áp dụng được. Người thiết kế có thể sử dụng ma trận M và đại lượng vô hướng $\rho > 0$ để thiết kế G_{OL} trong nỗ lực thu được một vòng lặp mong muốn G_{LQ}. Cụ thể là, ma trận M có thể được sử dụng để làm phù hợp các giá trị kỳ dị ở các tần số cao, tần số thấp, tất cả các tần số, v.v.. Giả thiết là (A, B) là ổn định và (A, M) không có các dạng không ảo không quan sát được, đảm bảo tồn tại một nghiệm ổn định. Thêm nữa, kết quả vòng lặp G_{LQ} sẽ có độ nhạy danh nghĩa và các đặc tính bền vững ổn định – hệ quả của LQFDE. Ma trận hệ số điều khiển G_c thu được có thể được sử dụng trong một vòng phản hồi trạng thái, một vòng phản hồi trạng thái biến đổi, hoặc trong một bộ bù dựa trên mô hình.

30.5 Bài toán nội xạ đầu ra $\mathcal{H}^2$

Phần này sẽ cho thấy làm thế nào để các phương pháp được trình bày trong bài toán phản hồi đầu ra có thể vẫn được tiếp tục cho phép trong thiết kế ước lượng trạng thái tối ưu $\mathcal{H}^2$ (các ma trận hệ số lọc H_f).

Cấu trúc đối tượng tổng quát của bài toán nội xạ đầu ra

Trong trường hợp này (đối lập với trường hợp phản hồi trạng thái), đối tượng tổng quát G (gồm đối tượng P và các hàm trọng) có dạng sau:

$$G = \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right] = \left[\begin{array}{c|cc} A & B_1 & I_{n\times n} \\ \hline C_1 & 0_{n_z\times n_w} & 0_{n_z\times n_u} \\ C_2 & D_2 & 0_{n_y\times n_u} \end{array}\right] = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \tag{30.255}$$

Từ đây dẫn đến các tín hiệu điều khiển u nằm gọn trong tất cả các trạng thái x của đối tượng tổng quát G. Được hiểu là tất cả các chế độ của A đều điều khiển được bằng $B_2 = I_{n\times n}$.

Các giả thiết nội xạ đầu ra

Các giả thiết nội xạ đầu ra là tập con của các giả thiết trong bài toán phản hồi đầu ra. Các giả thiết nội xạ đầu ra như sau.

Giả thiết 30.3 (Bài toán nội xạ đầu ra $\mathcal{H}^2$)

Trong suốt mục này, chúng ta giả thiết như sau:

Giả thiết đối tượng G_{22} . (A, C_2) có thể xác định được.

Giả thiết trọng số đo không suy biến. $\Theta = D_{12}^T D_{12} > 0$ (D_{21} có hạng dòng đầy đủ).

Giả thiết bộ lọc. $\begin{bmatrix} j\omega I - A & -B \\ C_1 & D_{21} \end{bmatrix}$ có hạng cột đầy đủ với mọi ω .

Nên chú ý rằng nếu $B_1 D_{21}^T = 0$, thì (3) tương đương với (A, B_1) không có các dạng ảo không điều khiển được. Nếu (A, B_1) là ổn định, thì điều này được thỏa mãn.

Luật điều khiển nội xạ đầu ra $\mathcal{H}^2$

Bộ điều khiển tối ưu $\mathcal{H}^2$ như sau:

$$K_{opt} = -H_f \tag{30.256}$$

Trong đó ma trận hệ số khuếch đại bộ lọc $H_f = R^{n \times n_y}$ được cho bởi:

$$H_f = [YC_2^T + B_1 D_{21}^T]\Theta^{-1} \tag{30.257}$$

trong đó $Y \geq 0$ là nghiệm bán xác định dương duy nhất của FARE

$$(A - B_1 D_{21}^T \Theta^{-1} C_2)Y + Y(A - B_1 D_{21}^T \Theta^{-1} C_2)^T + B_1(I - D_{21}^T \Theta^{-1} D_{21})B_1^T - YC_2^T \Theta^{-1} C_2 Y = 0 \tag{30.258}$$

Các điểm cực hệ kín có được từ luật nội xạ đầu ra ở trên là giá trị riêng của $A - H_f C_2$. Chuẩn hệ kín nhỏ nhất là:

$$\min_K \|T_{w_2}\|_{\mathcal{H}^2} = \sqrt{trace\left(C_1 Y C_1^T\right)} \tag{30.259}$$

trong đó Y là nghiệm của FARE

Thiết kế vòng ước lượng (lọc)

Nếu chọn

$$B_1 = \begin{bmatrix} L & 0_{n \times n_y} \end{bmatrix} \tag{30.260}$$

$$D_{21} = \begin{bmatrix} 0_{n_y \times n_u} & \sqrt{\mu} I_{n_y \times n_y} \end{bmatrix} \tag{30.261}$$

$$C_2 = C \tag{30.262}$$

$$\Theta = \mu I_{n_y \times n_y} \tag{30.263}$$

thì $B_1 D_{21}^T = 0$ và ta có:

$$H_f = YC_2^T \frac{1}{\mu} \tag{30.264}$$

trong đó $Y \geq 0$ là nghiệm bán xác định dương duy nhất của FARE:

$$YA^T + AY + LL^T - YC^T \Theta^{-1} CY = 0 \tag{30.265}$$

Khi đó, KFDE sau đây có thể nhận được từ FARE:

$$[I+G_{KF}(j\omega)][I+G_{KF}(j\omega)]^H = I+\left[\frac{1}{\sqrt{\mu}}G_{FOL}(j\omega)\right]\left[\frac{1}{\sqrt{\mu}}G_{FOL}(j\omega)\right]^H \tag{30.266}$$

trong đó:

$$G_{FOL} = C(sI-A)^{-1}L \tag{30.267}$$

$$G_{KF} = C(sI-A)^{-1}H_f \tag{30.268}$$

Các ý tưởng thiết kế vòng lặp đã trình bày trước đây có thể áp dụng được. Người thiết kế có thể sử dụng ma trận L và đại lượng vô hướng $\mu > 0$ để thiết kế G_{FOL} để cố gắng đạt được G_{KF} mong muốn. Cụ thể là, ma trận L có thể được sử dụng để làm phù hợp các giá trị kỳ dị ở các tần số cao, tần số thấp, tất cả các tần số, v.v.. Giả thiết là (A,C) là ổn định và (A,L) không có các dạng ảo không điều khiển được, thì chắc chắn tồn tại một nghiệm ổn định. Thêm nữa, vòng lặp G_{LQ} thu được sẽ có độ nhạy danh nghĩa (nominal sensitivity) và các đặc tính bền vững ổn định – hệ quả của KFDE. Ma trận hệ số bộ lọc thu được H_f (nội xạ đầu ra) có thể được sử dụng trong một vòng ước lượng (phản hồi), một vòng ước lượng (phản hồi) biến đổi, hoặc trong một bộ bù dựa trên mô hình.

30.6 Kết luận

Chương này đã trình bày nguyên tắc cơ bản để thiết kế hệ thống điều khiển bằng tối ưu hóa $\mathcal{H}^2$. Với các hệ thời gian liên tục LTI, các phương pháp này rất linh hoạt và được ứng dụng rộng rãi. Chúng có thể được sử dụng để thiết kế các luật điều khiển phản hồi trạng thái hệ số hằng, các ước lượng trạng thái hệ số hằng, các bộ điều khiển phản hồi đầu ra động, và hơn thế nữa. Các hàm trọng được sử dụng dễ dàng trong nguyên tắc đối tượng tổng quát đã trình bày. Chức năng này có thể được sử dụng để thực hiện các mục tiêu thiết kế cho hệ kín. Tất cả các tư tưởng đã được trình bày đó có thể được mở rộng với sự thay đổi khéo léo (rất quan trọng), để đem lại hệ thống điều khiển bằng tối ưu $\mathcal{H}^\infty$. Để biết thêm chi tiết bạn đọc có thể tìm trong [8, 11].

Các phương pháp đã trình bày trong chương này có thể được mở rộng sang hệ bất biến dịch tuyến tính (Linear Shift Invariant - LSI) thời gian rời rạc, cũng có thể mở rộng sang các hệ thống dữ liệu trích mẫu.

Tài liệu tham khảo

[1] Chen, T. and Francis, B., *Optimal Sampled-Data Control Systems*, Springer, London, 1995.

[2] Dorf, R.C. and Bishop, R.H., *Modern Control Systems*, Addison Wesley, 8th edition, CA, 1998.

[3] Doyle, J.C., "Guaranteed margins for LQG regulators," *IEEE Transactions on Automatic Control*, Vol. AC-23, No. 4, August 1978, pp. 756–757.

[4] Doyle, J.C., Glover, K., Khargonekar, P.P., and Francis, B.A., "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Transactions on Automatic Control*, Vol. AC-34, No. 8, 1989, pp. 831–847. Also see *Proceedings of the 1988 American Control Conference*, Atlanta, Georgia, June, 1988.

[5] Kalman, R.E., "A new approach to linear filtering and prediction problems," *ASME Journal of Basic Engineering*, Vol. 85, 1960, pp. 34–45.

[6] Kalman, R.E. and Bucy, R.S., "New results in linear filtering and prediction problems," *ASME Journal of Basic Engineering*, 1960, pp. 95–108.

[7] Kwakernaak, H. and Sivan, R., Linear Optimal Control Systems, Wiley-Interscience, New York, 1972.

[8] Rodriguez, A.A., A Practical Neo-Classical Approach to Feedback Control System Analysis and Design,Control3D, 2000.

[9] Spong, M.W. and Vidyasagar, M., *Robot Dynamics and Control*, John Wiley and Sons, New York, 1989.

[10] Zhou, K., Doyle, J.C., and Glover, K., *Robust and Optimal Control*, Prentice-Hall, NJ, 1996.

[11] Zhou, K. and Doyle, J.C., Essentials of Robust Control, Prentice-Hall, NJ, 1998.

31

Thiết kế điều khiển thích nghi và phi tuyến

31.1 Giới thiệu 31-1
31.2 Lý thuyết Lyapunov cho các hệ dừng 31-2
31.3 Định lý Lyapunov đối với hệ không dừng 31-3
31.4 Lý thuyết điều khiển thích nghi 31-4
31.5 Các hệ điều khiển thích nghi phi tuyến 31-6
31.6 Ví dụ về điều chỉnh thích nghi tư thế bay cho tàu vũ trụ 31-8
31.7 Điều khiển thích nghi phản hồi đầu ra 31-10
31.8 Các bộ quan sát thích nghi và điều khiển phản hồi đầu ra 31-11
31.9 Nhận xét 31-12

Maruthi R.Akella
The University of Texas at Austin

31.1 Giới thiệu

Thách thức lớn nhất đối với lý thuyết điều khiển hiện đại là phải đem lại hiệu năng có thể chấp nhận được khi giải quyết các mô hình yếu với độ phi tuyến cao và các cảm biến giá rẻ trong các điều kiện hoạt động khác nhau. Những khó khăn này xuất hiện hầu như trong mọi ứng dụng công nghiệp. Thường các hệ thống luôn chứa một số lượng lớn sai số mô hình và tham số nên các bộ điều khiển "cứng" không thể đạt được các yêu cầu về ổn định và hiệu năng. Để có được lời giải hợp lý cho bài toán này cần phải có sự kết hợp giữa lý thuyết điều khiển phi tuyến, các thành phần thích nghi và xử lý thông tin. Chính điều này cùng với một số ứng dụng thực tế như điều khiển quá trình hóa học và thiết kế thiết bị lái tự động cho máy bay hoạt động ổn định với nhiều tốc độ và độ cao đã thúc đẩy sự ra đời và phát triển của lý thuyết điều khiển thích nghi.

Một định nghĩa được chấp nhận rộng rãi đối với hệ thống thích nghi là: nó là một hệ vật lý được thiết kế trên quan điểm thích nghi [1]. Tất cả các kết quả ổn định và hội tụ hiện nay, trong lĩnh vực lý thuyết điều khiển thích nghi, xoay quanh giả thiết chính là các tham số chưa biết phải xuất hiện tuyến tính trong mô hình chứa các đặc tính phi tuyến đã biết. Cụ thể, quá trình tạo nên các ước lượng tham số như là các biến trạng thái, do đó sẽ mở rộng chiều không gian trạng thái của hệ thống ban đầu. Một cách tự nhiên, các lời giải điều khiển thích nghi đối với cả hệ động lực tuyến tính và phi tuyến đều dẫn đến công thức biến đổi theo thời gian phi tuyến trong đó các ước lượng tham số chưa biết được cập nhật bằng cách sử dụng dữ liệu vào ra. Một cơ chế thích nghi tham số (thường là phi tuyến) được sử dụng để cập nhật các tham số trong luật điều khiển. Sẽ nhận được đặc tính phi tuyến do có phản hồi thích nghi, nên cần chắc chắn rằng tính ổn định của hệ kín được duy trì. Vì vậy một thực tế hiển nhiên là điều khiển thích nghi và tính ổn định của hệ phi

tuyến thực chất là liên quan đến nhau và mọi kết quả nghiên cứu trong lĩnh vực này đều thuận lợi cho việc nghiên cứu lĩnh vực kia. Nhiều công thức trong lý thuyết ổn định phi tuyến có thể được tận dụng như là phương pháp trực tiếp Lyapunov và các phương pháp gián tiếp. Chúng ta sẽ trình bày các công cụ toán học và phân tích quan trọng trước để nghiên cứu tính ổn định của hệ động lực phi tuyến.

31.2 Lý thuyết Lyapunov cho các hệ dừng

Phương pháp trực tiếp Lyapunov là một trong những phương pháp phổ biến nhất được chấp nhận rộng rãi để chứng minh tính ổn định của hệ kín trong lĩnh vực điều khiển thích nghi. Nó không bị gới hạn đến ứng xử thành phần và nó xác định các đặc tính ổn định của hệ phi tuyến bằng cách xét tiến trình thời gian của các lời giải hệ thống đối với một hàm vô hướng "kiểu năng lượng", thường được biết đến là hàm Lyapunov.

Xét hệ động lực bất kỳ được biểu diễn bởi phương trình vi phân phi tuyến dưới đây:

$$\dot{x} = f(x) \quad , \quad f(0) = 0 \tag{31.1}$$

Rõ ràng $x(t) = 0$ là một nghiệm. Điều kiện đủ để tồn tại duy nhất nghiệm phương trình (31.1) là $f(x)$ là Lipschitz địa phương, tức là:

$$\|f(x) - f(y)\| \le L\|x - y\| \tag{31.2}$$

với mọi x và y trong một lân cận xác định của gốc. Chúng ta quan tâm tới tính ổn định của nghiệm phương trình (31.1) với sự hiện diện của nhiễu. Trước khi trình bày về các định lý ổn định Lyapunov chính, ta trình bày một số định nghĩa quan trọng.

Định nghĩa: **Ổn định Lyapunov**

Nghiệm x(t) = 0 của phương trình (31.1) được gọi là ổn định theo tiên chuẩn Lyapunov nếu với mọi $\varepsilon > 0$, luôn tồn tại $\delta(\varepsilon) > 0$ sao cho với mọi điều kiện ban đầu thỏa mãn $\|x(0)\| < \delta$, ta có $\|x(0)\| < \varepsilon$ với $t \in [0, \infty)$. Nghiệm là ổn định tiệm cận nếu nó là ổn định và tồn tại $\delta > 0$ sao cho tất cả các điều kiện đầu thỏa mãn $\|x(0)\| < \delta$ có tính chất:

$$\lim_{t \to \infty} \|x(t)\| = 0$$

Nghiệm là ổn định tiệm cận toàn cục nếu nó ổn định tiệm cận tại tất cả các điều kiện đầu. Các định nghĩa này liên quan đến sự ổn định của các nghiệm riêng của phương trình (31.1) với điều kiện đầu và không ổn định với các phương trình vi phân.

Định nghĩa: **Hàm xác định dương và hàm bán xác định dương**

Một hàm khả vi liên tục $V : R^n \to R$ được gọi là xác định dương nếu (i) $V(0) = 0$ và (ii) $V(x) > 0$ với mọi $x \neq 0$. Một hàm là bán xác định dương nếu điều kiện (ii) được thay bằng $V(x) \ge 0$ với mọi $x \neq 0$.

Định lý: **Định lý về tính ổn định của Lyapunov cho hệ dừng**

Nếu tồn tại một hàm xác định dương $V : R^n \to R$ sao cho đạo hàm theo thời gian của V với nghiệm $\dot{x} = f(x)$ được cho bởi:

$$\frac{d}{dt}V = \left[\frac{\partial V}{\partial x}\right]^T \dot{x} = \left[\frac{\partial V}{\partial x}\right]^T f(x)$$

là bán xác định âm, thì nghiệm $x(t) = 0$ của phương trình (31.1) là ổn định. Trong trường hợp này, nghiệm hội tụ trong miền $\{x \in R^n : \dot{V}(x) = 0\}$. Nếu $\dot{V}$ là xác định âm, thì nghiệm sẽ ổn định

tiệm cận. Hơn nữa, nếu $\dot{V}$ là xác định âm và $V(x) \to \infty$ khi $\|x\| \to \infty$, thì nghiệm là ổn định tiệm cận toàn cục. Hàm $V(x)$ được gọi là hàm Lyapunov cho hệ được mô tả trong phương trình (31.1).

Nhận xét: Định lý Lyapunov, mặc dù phát biểu đơn giản, nhưng có ứng dụng rất lớn trong việc phân tích tính ổn định của hệ phi tuyến. Tuy nhiên, vì định lý này chỉ cung cấp điều kiện đủ dưới dạng hàm Lyapunov, nên chúng ta thường gặp phải vấn đề khó trong việc đi tìm hàm Lyapunov phù hợp. Trong trường hợp đặc biệt khi phương trình (31.1) là một hệ tuyến tính ổn định:

$$\dot{x} = A_m x$$

một hàm Lyapunov toàn phương $V = x^T P x$ tồn tại khi P là một ma trận xác định dương đối xứng thỏa mãn phương trình được gọi là phương trình Lyapunov:

$$A_m^T P + P A_m = -Q \tag{31.3}$$

với Q là một ma trận xác định dương đối xứng bất kỳ. Ngoài ra, không còn cách tổng quát nào để xây dựng hàm Lyapunov cho hệ phi tuyến. Giống như quy tắc ngón tay cái, trong trường hợp là các hệ cơ khí, đầu tiên cần xác định các đại lượng "kiểu năng lượng".

31.3 Định lý Lyapunov đối với hệ không dừng

Xét tính ổn định của nghiệm phương trình vi phân không dừng (không tự điều khiển):

$$\dot{x} = g(x,t) \quad , \quad g(0,t) = 0 \quad \forall t \le 0 \tag{31.14}$$

Hàm g được giả thiết là liên tục trên từng đoạn đối với t và Lipschitz địa phương với lân cận nghiệm $x(t) = 0$. Điều này đảm bảo gốc tọa độ là trạng thái cân bằng của phương trình (31.4).

Để nghiên cứu về tính ổn định cân bằng của hệ thống không tự điều khiển, phải thấy rằng mọi nghiệm của phương trình (31.4) không những phụ thuộc vào thời gian t mà còn phụ thuộc vào thời gian khởi tạo t_0. Vì vậy, chúng ta cần xem lại định nghĩa về tính ổn định trước đây.

Định nghĩa: **Tính ổn định Lyapunov đồng nhất**

Nghiệm x(t) = 0 của phương trình (31.4) là ổn định đồng nhất nếu với mọi $\varepsilon > 0$, luôn tồn tại $\delta(\varepsilon) > 0$ không phụ thuộc vào thời gian khởi tạo t0 sao cho:

$$\|x(t_0)\| < \delta \quad \Rightarrow \quad \|x(t)\| < \varepsilon \text{ với } \forall t \ge t_0 \ge 0$$

Nghiệm là ổn định tiệm cận đồng nhất nếu nó là ổn định đồng nhất và tồn tại một hằng số dương ρ không phụ thuộc vào t_0 sao cho $\|x(t)\| \to 0$ khi $t \to \infty$ với mọi $\|x(t_0)\| < \rho$. Nghiệm là ổn định tiệm cận đồng nhất toàn cục nếu nó là ổn định tiệm cận đồng nhất với mọi điều kiện đầu.

Định lý về tính ổn định chính cho hệ phi tuyến không tự điều khiển yêu cầu xác định *các hàm lớp K* xác định.

Định nghĩa: **Các hàm lớp K**

Một hàm liên tục $\alpha : [0,a) \to [0,\infty)$ được gọi là thuộc *lớp K* nếu nó đơn điệu tăng và $\alpha(0) = 0$.

Nó được gọi là thuộc lớp K_∞, hay tiến tới vô cùng, nếu $a = \infty$ thì $\alpha(r) \to \infty$ khi $r \to \infty$.

Định lý: **Định lý về sự ổn định của Lyapunov cho hệ biến đổi với thời gian**

Xét miền $D = \{x \in R^n : \|x\| \le R\}$ với trạng thái cân bằng $x(t) = 0$ của phương trình (31.4). Nếu luôn tồn tại một hàm vô hướng $V : R^n \times R^- \to R$ với các đạo hàm riêng liên tục sao cho:

(i) $\alpha_1(\|x\|) \le V(x,t) \le \alpha_2(\|x\|)$ *xác định dương* và *giảm*

(ii) $\dot{V} = \dfrac{\partial V}{\partial t} + \left[\dfrac{\partial V}{\partial \boldsymbol{x}}\right]^T \boldsymbol{g}(\boldsymbol{x},t) \le \alpha_3(\|\boldsymbol{x}\|)$

với mọi $t \ge 0$, trong đó α_1, α_2 và α_3 là các hàm lớp K, thì điểm cân bằng $\boldsymbol{x} = 0$ là ổn định tiệm cận đồng nhất.

Nhận xét: Chú ý rằng để đưa ra tính ổn định của hệ phi tuyến không tự điều khiển, cần giới hạn hàm $V(\boldsymbol{x},t)$ bởi các hàm lớp K không phụ thuộc vào thời gian t. Một cách giải quyết chi tiết của tất cả các định nghĩa và chứng minh của định lý này bạn đọc có thể xem trong Slotine và Li [2] và Khalil [3].

Nhận xét: Trong những năm gần đây, định lý đảo Lyapunov đã đạt được một số kết quả tốt. Cụ thể là tất cả các hệ thống ổn định đồng nhất (hoặc ổn định tiệm cận đồng nhất), luôn tồn tại một hàm Lyapunov xác định dương có đạo hàm theo thời gian là bán xác định âm (xem Sastry và Bodson [4]). Các kết quả này rất hữu ích cho chất lượng của hệ kín bởi vì nó cho phép ước lượng một cách rõ ràng các tốc độ hội tụ cho nhiều trường hợp của hệ điều khiển thích nghi phi tuyến.

Ứng dụng của định lý ổn định của Lyapunov cho hệ không tự điều khiển vượt ra khỏi điều khiển thích nghi, nó thường dẫn đến đạo hàm bán xác định âm của hàm Lyapunov. Vì vậy, việc phân tích tính ổn định tiệm cận là một bài toán khó hơn nhiều và kết quả sau đây, được coi như là bổ đề Barbalat, rất hữu ích cho trường hợp này.

Bổ đề: **Barbalat**

Xét một hàm liên tục bất biến $\phi : R \to R$, tập xác định là tất cả các giá trị thực $t \ge 0$. Nếu:

$$\lim_{t\to\infty} \int_0^t \phi(s)ds$$

tồn tại và xác định, thì $\phi(t) \to 0$ khi $t \to \infty$.

Nhận xét: Một hệ quả của bổ đề này là nếu $\phi \in L_2$ và $\phi \in L_\infty$, thì $\phi(t) \to 0$ khi $t \to \infty$ (Xem Slotine và Li [2] và Tao [3] để xem thảo luận và chứng minh).

31.4 Lý thuyết điều khiển thích nghi

Khác với một bộ điều khiển thông thường, một bộ điều khiển thích nghi là một bộ điều khiển có các tham số có thể chỉnh được và một cơ chế điều chỉnh. Sau đây là các khái niệm cơ bản cần cho mọi trình bày về lý thuyết điều khiển thích nghi.

Bài toán điều chỉnh và bám

Mục đích của mọi bài toán điều khiển là duy trì giá trị đầu ra luôn giữ một giá trị mong muốn hoặc nằm trong dải xác định/có thể chấp nhận được của giá trị mong muốn. Nếu các giá trị mong muốn này là hằng số với thời gian, thì chúng ta có bài toán điều chỉnh, ngược lại là bài toán điều khiển bám.

Nguyên lý tương đương tuyệt đối

Nguyên lý này là nền tảng của hầu hết các phương pháp thiết kế điều khiển thích nghi và đã được quan tâm nhiều trong hai thập kỷ qua [4, 6, 7]. Các bộ điều khiển thích nghi dựa trên phương pháp này có được bằng cách thiết kế một luật điều khiển độc lập để đạt được mục đích điều khiển với giả thiết là đã biết tất cả các tham số của mô hình chưa biết (trường hợp tiền định), cùng với một luật cập nhật tham số, đó thường là một phương trình vi phân để tạo ra các ước lượng tham số trực tuyến sử dụng để thay thế các tham số chưa biết trong luật điều khiển. Đó như là một bộ điều khiển đã hoàn thiện với khả năng bám đầu ra trong trường hợp biết chính xác các tham số mô

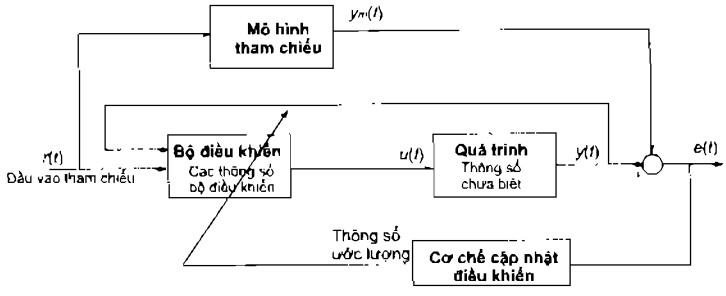
hình. Với sự hiện diện của các sai số tham số, cơ chế thích nghi sẽ chỉnh các tham số của bộ điều khiển sao cho tiếp cận đối tượng bám được thiết kế một cách tiệm cận. Như vậy, vấn đề chủ yếu trong thiết kế bộ điều khiển thích nghi là tạo ra cơ chế thích nghi (luật cập nhật tham số) nhằm đảm bảo rằng hệ thống điều khiển vẫn ổn định và sai lệch bám đầu ra hội tụ về 0 khi các giá trị tham số được cập nhật.

Điều khiển thích nghi trực tiếp và thích nghi gián tiếp

Có hai cách tiếp cận khác nhau trong điều khiển thích nghi cho các mô hình có chứa các tham số chưa biết hay chưa xác định. Cách thứ nhất được gọi là *phương pháp trực tiếp*, các tham số của bộ điều khiển được chỉnh trực tiếp bởi cơ chế thích nghi bằng cách tối ưu một số chỉ tiêu chất lượng đã xác định được trước trên cơ sở đầu ra. Cách thứ hai là *phương pháp gián tiếp*, các tham số mô hình được ước lượng trực tiếp và được cập nhật bằng luật thích nghi và các giá trị được ước lượng này sẽ được sử dụng để tính toán các tham số của bộ điều khiển. Điều khiển thích nghi trực tiếp không cần thêm sự tính toán này. Do đó, điều khiển thích nghi gián tiếp là thích nghi tham số mô hình, ngược lại điều khiển thích nghi trực tiếp là thích nghi kết quả đầu ra. Quá trình nhận dạng tham số mô hình trong phương pháp gián tiếp là tường minh trong khi đó trong phương pháp trực tiếp là không tường minh. Vì vậy, chúng còn được gọi là phương pháp tường minh và không tường minh. Trong cả hai trường hợp này thì cấu trúc bộ điều khiển vẫn giống nhau và được xác định từ nguyên tắc tương đương tuyệt đối.

Điều khiển thích nghi tham chiếu mô hình

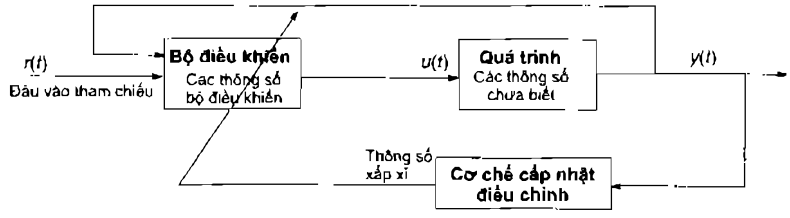
Cấu trúc bài toán điều khiển thích nghi tham chiếu mô hình (MRAC - Model Reference Adaptive Control) gồm bốn phần: (i) đối tượng chứa các tham số chưa biết, (ii) mô hình tham chiếu thích hợp để xác định các đặc tính đầu ra mong muốn, (iii) một luật điều khiển phản hồi chứa các tham số có thể điều chỉnh và (iv) một cơ chế thích nghi để cập nhật các tham số có thể điều chỉnh bằng luật điều khiển trên. Một phác thảo cho cấu trúc này được thấy trên hình 31.1.



HÌNH 31.1 Cấu trúc điều khiển thích nghi tham chiếu mô hình

Đối tượng được giả thiết là có cấu trúc đã biết với các tham số chưa biến. Trong trường hợp các hệ tuyến tính, điều này có nghĩa là số điểm cực và điểm không được giả thiết là đã biết, nhưng vị trí chính xác của các điểm cực và điểm không thì chưa biết. Đối với hệ phi tuyến, cấu trúc của các phương trình ảnh hưởng đến chuyển động được giả thiết là đã biết, nhưng một vài tham số xuất hiện tuyến tính trong các phương trình này thì chưa biết. Mô hình tham chiếu xác định hành vi đầu ra cần đạt từ mô hình như là một kết quả của đầu vào tham chiếu ngoài. Nó tạo ra đáp ứng đối tượng lý tưởng mà cơ chế thích nghi cần tìm để bám theo trong khi cập nhật tham số ước lượng. Việc chọn đối tượng tham chiếu là trọng tâm của thiết kế MRAC và bất kỳ lựa chọn có thể chấp nhận nào cũng phải thoả mãn hai yêu cầu cơ bản sau. Yêu cầu thứ nhất là mô hình tham chiếu phải phản ánh chính xác các chỉ tiêu chất lượng của hệ kín, như là thời gian tăng trưởng, thời gian đặt, độ quá điều chỉnh và các chỉ tiêu chất lượng khác. Yêu cầu thứ hai là với cấu trúc giả thiết của động học mô hình thì hành vi đầu ra của mô hình tham chiếu cần được thiết kế gần đúng

bởi hệ thống điều khiển thích nghi dẫn đến các điều kiện bổ sung về cấp độ quan hệ của mô hình tham chiếu và các điều kiện kích thích liên tục ở đầu vào tham chiếu. Cấu trúc bộ điều khiển được ra lệnh bằng nguyên lý tương đương tuyệt đối và cả 2 thủ tục cập nhật tham số trực tiếp và gián tiếp đều có thể được chọn với cấu trúc MRAC. Rất nhiều bài toán trong lĩnh vực này liên quan đến hệ liên tục.



HÌNH 31.2 Cấu trúc điều khiển tự chỉnh

Bộ điều khiển tự chỉnh (STC)

Trái với điều khiển MRAC, không có mô hình tham chiếu trong thiết kế bộ điều khiển tự chỉnh (Self-Tuning Controller – STC). Sơ đồ của thiết kế STC như hình 31.2. Trong công thức này, các thông số điều khiển của tham số hệ thống được ước lượng trong thời gian thực, phụ thuộc vào phương pháp trực tiếp hay gián tiếp. Sau đó, các ước lượng này được sử dụng nếu chúng bằng các tham số thực (thiết kế tương đương tuyệt đối). Việc ước lượng tham số đòi hỏi tìm ra bộ tham số *phù hợp nhất* dựa trên dữ liệu vào-ra của hệ thống. Nó khác với kiểu thích nghi tham số MRAC mà ở đó ước lượng tham số được cập nhật theo cách nào đó để thu được sai số bám tiệm cận giữa hệ thống và mô hình tham chiếu. Trong một số kiểu ước lượng STC, cũng có thể đánh giá tiêu chuẩn chất lượng của các ước lượng tham số được sử dụng trong thiết kế bộ điều khiển. Nhiều cách kết hợp khác nhau của các phương pháp ước lượng được chấp nhận và áp dụng cho cả hệ liên tục và hệ rời rạc. Do có sự "tách biệt" giữa ước lượng tham số và điều khiển trong STC, nên có sự linh hoạt cao hơn trong thiết kế. Tuy nhiên, tính ổn định và hội tụ là khó chứng minh và cần phải có các điều kiện khắt khe hơn đối với các tín hiệu đầu vào để đảm bảo hội tụ tham số. Như đã nói ở trên, thiết kế STC đã xuất hiện trong việc nghiên cứu bài toán điều chỉnh ngẫu nhiên và nhiều lý thuyết dành cho hệ rời rạc sử dụng phương pháp gián tiếp. Mặc dù có sự khác nhau giữa MRAC và STC, nhưng tồn tại sự tương ứng trực tiếp giữa các bài toán từ hai miền trên [8].

31.5 Các hệ điều khiển thích nghi phi tuyến

Với hầu hết các trường hợp tổng quát của hệ phi tuyến, tồn tại rất ít lý thuyết trong lĩnh vực điều khiển thích nghi. Mặc dù lĩnh vực này rất được quan tâm nhờ vào các ứng dụng tiềm năng trong sự đa dạng của các hệ cơ phức tạp nhưng vẫn tồn tại những khó khăn lý thuyết do thiếu các công cụ phân tích chung. Tuy nhiên, một số trường hợp đặc biệt quan trọng sẽ được trình bày ở đây và chúng ta tổng kết các điều kiện mà các lớp hệ thống đó thỏa mãn:

1. Các tham số chưa biết trong hệ phi tuyến được tham số hoá một cách tuyến tính.
2. Vectơ trạng thái hoàn toàn được xác định.
3. Khi các tham số chưa biết được giả thiết là đã biết, đầu vào điều khiển có thể bỏ qua tất cả các tính chất phi tuyến theo hướng tuyến tính hóa phản hồi và các thành phần động lực bên trong còn lại phải ổn định. Sau đó thiết kế thích nghi thực hiện được bằng nguyên lý tương đương tuyệt đối.

Bây giờ chúng ta xem xét một phương pháp MRAC phi tuyến điển hình để giải quyết trường hợp mô hình hệ phi tuyến không biết các tham số bên trong nó. Xét hệ phi tuyến:

$$\dot{x} = \theta f(x) + u \tag{31.5}$$

trong đó θ là một tham số ma trận hằng chưa biết và f là một hàm vécto phi tuyến khả vi và đã biết. Tương tự với phương pháp MRAC, chúng ta giả sử rằng cần có trạng thái x bám tiệm cận trạng thái x_m của hệ tham chiếu thỏa mãn:

$$\dot{x}_m = A_m x_m + r \tag{31.6}$$

trong đó $r(t)$ là đầu vào tham chiếu được giới hạn và liên tục trên từng đoạn và A_m là ma trận Hurwitz. Đưa vào một vectơ sai số $e = x - x_m$ sao cho động lực sai số có thể được thiết lập bằng cách trừ hai phương trình (31.5) và (31.6) như sau:

$$\dot{e} = \theta f(x) - A_m x_m - r + u \tag{31.7}$$

Nếu tham số θ được giả thiết là đã biết, thì việc chọn đầu vào điều khiển $u = A_m x + r - \theta f(x)$ sẽ diễn tả cấu trúc như sau cho động lực sai số tạo nên đối tượng điều khiển:

$$\dot{e} = A_m e$$

Tuy nhiên, không thể chọn được luật điều khiển vì θ chưa biết. Do đó, chúng ta giữ lại cấu trúc tương tự cho luật điều khiển, trừ việc thay θ bằng ước lượng không dừng $\hat{\theta}$ của nó, sao cho luật điều khiển thích nghi dựa trên nguyên lý tương đương tuyệt đối được cho bởi công thức:

$$u = A_m x + r - \hat{\theta} f(x) \tag{31.8}$$

Áp dụng cho luật điều khiển trong phương trình (31.7) cho ta động lực sai số vòng kín như sau:

$$\dot{e} = A_m e - \tilde{\theta} f(x) \tag{31.9}$$

trong đó chúng ta đã đưa ra biến $\tilde{\theta}(t)$ để mô tả sai lệch ước lượng tham số $\tilde{\theta}(t) - \theta$. Có hai vấn đề còn lại phải làm là: (i) chỉ rõ tính ổn định và hội tụ tiệm cận của $e(t)$ tiến tới 0 khi $t \to \infty$, (ii) đưa ra một cơ chế thích nghi tham số hợp lý cho $\tilde{\theta}(t)$. Chúng ta hoàn thành cả hai việc trên bằng cách chọn phương pháp Lyapunov. Giả thiết A_m là Hurwitz, với mọi cách chọn ma trận xác định dương và đối xứng Q, luôn tồn tại một ma trận xác định dương, đối xứng P thỏa mãn phương trình Lyapunov cho bởi (31.3). Chọn hàm Lyapunov dưới dạng một ma trận P:

$$V = e^T P e + tr[\tilde{\theta}^T \Gamma^{-1} \tilde{\theta}] \tag{31.10}$$

trong đó Γ là ma trận tốc độ học xác định dương đối xứng. Đạo hàm V theo thời gian và kết hợp với nghiệm của phương trình (31.9) ta có:

$$\dot{V} = e^T (P A_m + A_m^T P) e - 2 e^T P \tilde{\theta} f(x) + 2 tr[\tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}}] \tag{31.11}$$

Sử dụng đồng nhất thức vết ma trận [9], có thể thấy rằng:

$$e^T P \tilde{\theta} f(x) = tr[P \tilde{\theta} f(x) e^T] = tr[\tilde{\theta} f(x) e^T P] = tr[\tilde{\theta}^T P e f^T(x)]$$

sao cho chúng ta có thể kết hợp hai biểu thức bên vế phải của phương trình (31.11) như sau:

$$\dot{V} = e^T \underbrace{(P A_m + A_m^T P)}_{-Q} e + 2 tr[\tilde{\theta} \{\Gamma^{-1} \dot{\tilde{\theta}} - P e f^T(x)\}] \tag{31.12}$$

Do θ là hằng số, thì $\dot{\theta} = 0$ và $\dot{\tilde{\theta}} = \dot{\tilde{\theta}}$. Vì vậy, nếu luật thích nghi để cập nhật $\hat{\theta}$ được chọn là:

$$\dot{\tilde{\theta}} = \Gamma P e f^T(x) \tag{31.13}$$

thì đạo hàm của hàm Lyapunov trong phương trình (31.12) trở thành :

$$\dot{V} = -e^T Qe \tag{31.14}$$

Đây là hàm bán xác định âm, nhưng không xác định âm. Điều này dẫn đến $V(t) \le V(0)$ với mọi $t \ge 0$ và vì thế e và $\hat{\theta}$ phải được giới hạn. Điều này dẫn đến $x = e + x_m$ cũng được giới hạn. Ngoài ra $V(t) \ge 0$ và $V(0) \le 0$, nghĩa là $\lim_{t\to\infty} V(t) = V_\infty$ tồn tại và xác định :

$$\int_0^\infty \dot{V}(\tau)d\tau = -\int_0^\infty e^T(\tau)Qe(\tau)d\tau = V_\infty - V(0)$$

dẫn đến $e \in L_2 \cap L_1$. Từ phương trình (31.9), hiển nhiên rằng $\dot{e} \in L_\infty$. Vì vậy, chúng ta có thể dùng bổ đề Barbalat để chứng minh $e(t) \to 0$ khi $t \to \infty$. Lưu ý rằng sai lệch tham số $\tilde{\theta} = \hat{\theta} - \theta$ không cần thiết phải hội tụ về 0. Việc hội tụ của tham số thật có thể chỉ xuất hiện khi đầu vào tham chiếu $r(t)$ thỏa mãn tính quan sát bất biến và các điều kiện kích thích liên tục [4].

Nhận xét: Chú ý trong thiết kế MRAC ở trên, khi sự ổn định và sự hội tụ của sai lệch bám được đảm bảo cho mọi giá trị của A_m, Q và Γ, thì chất lượng của bộ điều khiển sẽ phụ thuộc rất nhiều vào tốc độ học Γ. Tốc độ học "nhỏ hơn" nghĩa là sự thích nghi sẽ chậm hơn dẫn đến sai lệnh bám lớn và quá độ lớn. Ngược lại, giới hạn trên của tốc độ học được giới hạn bởi động lực phi mô hình, nguyên nhân là do một giá trị tốc độ học quá lớn sẽ tạo ra ước lượng tham số dao động cao, dẫn đến kích thích bất lợi cho động lực đối tượng phi mô hình tần số cao.

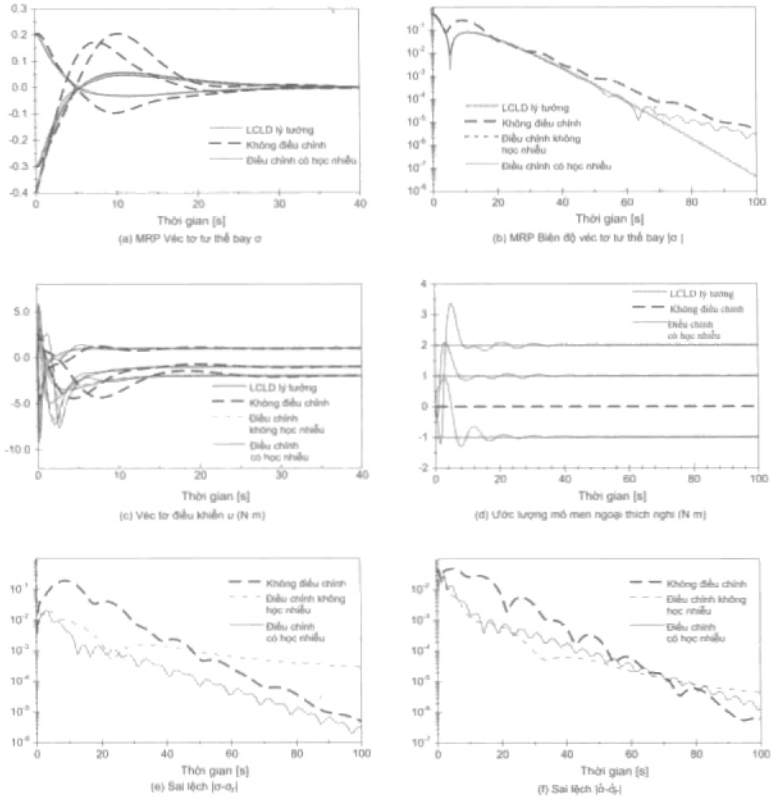
Nhận xét: Phương pháp thiết kế bộ điều khiển được dựa trên ba bước cơ bản: (i) tìm cấu trúc bộ điều khiển thích hợp với tư tưởng tuyến tính hóa phản hồi, (ii) đạo hàm động lực sai số bám phụ thuộc vào số hạng sai số tham số và (iii) tìm một hàm Lyapunov phù hợp có thể được dùng để tìm ra luật cập nhập tham số sao cho sai số bám tiến tới 0. Xác định cấu trúc bộ điều khiển trong trường hợp tham số đã biết là bước cơ bản nhất với bất kỳ thiết kế thích nghi nào do phần lớn quan điểm cho rằng tuyến tính hóa phản hồi thích nghi không thể luôn luôn áp dụng được vào các hệ thống đã tuyến tính hóa phản hồi trong trường hợp tham số đã biết. Điều này xảy ra bởi vì các đạo hàm bậc cao của ước lượng tham số xuất hiện trong luật điều khiển với hệ thống bậc cao hơn, gây khó khăn trong việc ứng dụng nguyên tắc tương đương tuyệt đối. Các phương pháp mới khác được thay đổi so với nguyên tắc tương đương tuyệt đối bằng cách sử dụng các hàm suy giảm phi tuyến và hiệu chỉnh [10]. Trong các phương pháp này, luật thích nghi ước lượng các tham số đối tượng chưa biết một cách trực tiếp, do đó cho phép sử dụng toàn bộ kiến thức trước và vì vậy loại trừ khả năng tràn tham số hóa như trong phương pháp MRAC trực tiếp truyền thống. Phương pháp thiết kế và kiểm chứng ổn định đạt được thông qua quá trình đệ quy [10, 11]. Tổng quan về vấn đề này có thể tìm thấy từ Kokotovic [12] và các kết quả của nhóm nghiên cứu của ông sau này.

Nhận xét: Trong thực tế là luôn tồn tại các sai số mô hình và các hiệu ứng của nhiễu chưa biết, cùng với các tham số chưa biết, các giải pháp điều khiển thích nghi phải hướng tới bài toán bền vững. Do sai số tham số luôn không biết, nên đạo hàm theo thời gian hàm Lyapunov luôn là bán xác định âm. Điều này dẫn đến phương trình hệ kín không ổn định theo hàm mũ, thậm chí cũng không ổn định tiệm cận bất biến. Nhiễu phi mô hình hoặc nhiễu ngoại có thể làm $\dot{V}$ bất định ngay lập tức và hầu hết các phương pháp chứng minh tính bền vững đều phải đảm bảo $\dot{V}$ là âm ngoài khoảng lân cận của trạng thái cân bằng. Bằng cách đưa vào thêm một số hạng trong luật điều khiển thích nghi trong phương trình (31.13), (được gọi là biến đổi σ), Ioannou [7] đã đạt được tính ổn định bền vững. Phương pháp này mặc dù rất phổ biến nhưng có nhiều hạn chế vì khi bỏ qua nhiễu thì sai số bám không thể hội tụ về 0. Để giải quyết vấn đề này, các phương pháp khác như biến đổi ε [6] đã được đề xuất để bảo đảm tính bền vững trong thiết kế thích nghi.

31.6 Ví dụ về điều chỉnh thích nghi tư thế bay cho tàu vũ trụ

Xét bài toán của tàu vũ trụ với một tư thế bay ban đầu khác không và vectơ vận tốc góc của vật đã được đưa về vectơ vị trí 0. Bài toán điều chỉnh tư thế thích nghi vật rắn này dựa trên phương

pháp tuyến tính hóa phản hồi đã được giải quyết bởi Schaub, Akella và Junkins [13]. Các phương trình chủ yếu được mô tả bởi phương trình chuyển động quay Euler và động lực học hệ kín tuyến tính mong muốn (LCLD – linear closed loop dynamics) có thể là dạng PD hoặc dạng PID [13, 14]. Giả thiết chỉ có ước lượng thô của ma trận mômen quán tính là đã biết. Luật điều khiển thích nghi được trình bày bao gồm số hạng phản hồi nguyên trong động lực học hệ kín mong muốn và đạt được độ ổn định tiệm cận ngay cả khi có nhiễu ngoại phi mô hình.



Hình 31.3 Sự ổn định vật rắn trong điều kiện quán tính lớn và bỏ qua nhiễu ngoài

Kết quả mô phỏng được thấy ở hình 31.3. Vectơ tư thế bay được xác định dưới dạng tham số Rodrigues sửa đổi (MRP – Modified Rodrigues Parameter) mà thành phần σ_i của nó được mô tả trong hình 31.3(a). Mặc dù không có bất kỳ sự thích nghi nào, hệ hở vẫn ổn định tiệm cận. Tuy nhiên, sai số tư thế bay tức thời không tốt bằng LCLD mong muốn. Khi có thích nghi, chất lượng rất gần với LCLD lý tưởng.

Hình 31.3(b) minh họa biên độ của vectơ sai lệch tư thế bay MRP σ trên đồ thị logarit. Ngược với sai số lớn của hệ hở, luật điều khiển không thích nghi trong 20 giây đầu của quá trình cho đặc tính hội tụ cuối cùng khá tốt. Hoạt động của LCLD lý tưởng được biểu thị bằng nét đứt. Hai phiên bản của luật điều khiển thích nghi được so sánh ở đây, chúng khác nhau ở chỗ có hay không có nhiễu ngoại được ước lượng thích nghi. Trong hình này, cả 2 luật thích nghi đều cho động lực học hệ kín tuyến tính rất tốt trong 40 giây đầu của quá trình. Sau đó luật thích nghi không học nhiễu bắt đầu suy yếu ở một tỷ lệ chậm hơn, thậm chí hơn cả giải pháp hệ hở (không thích nghi). Mặc dù có nhiễu ngoài, nhưng sự thích nghi cải thiện đáng kể tốc độ hội tụ cuối cùng. Tuy nhiên, không trường hợp thích nghi nào có sự khác biệt so với trường hợp LCLD lý tưởng cho đến khi biên độ sai số tư thế MRP bị suy giảm xuống xấp xỉ 10^{-3}. Điều này tương ứng với sai số góc quay cơ sở xấp xỉ khoảng 0.23^0. Với thích nghi nhiễu ngoài, sai số bám tại đó sai lệch LCLD xuất hiện là nhỏ hơn hai bậc biên độ.

Chất lượng của luật điều khiển thích nghi có thể thay đổi bằng cách chọn tốc độ học khác nhau. Tuy nhiên, do có sự xuất hiện của ma trận quán tính khởi tạo *lớn* và sai số mô hình nhiễu ngoài, nên tốc độ học được giảm bớt để tránh các mômen chuyển tiếp lớn. Các thành phần vectơ mômen điều khiển u_t cho một số trường hợp được mô tả như hình 31.3(c). Các mômen hệ hở không giống mômen LCLD lý tưởng trong chuyển động tức thời. Các mômen trong các trường hợp thích nghi là rất giống nhau. Sự khác nhau chỉ là trường hợp học nhiễu ngoài gây ra vài dao động của điều khiển so với trường hợp LCLD. Tuy nhiên, với tốc độ học nghi được chọn thì không một luật điều khiển nào đưa ra được các mômen từ trễ tức thời gần với mômen LCLD lý tưởng. Hình 31.3(d) minh họa ước lượng nhiễu ngoại thích nghi F_t tiệm cận nhiễu ngoại thực F_t^*. Bằng cách giảm tốc độ học thích nghi nhiễu ngoại, các sai lệch ước lượng thích nghi tức thời được giữ trong dải hợp lý.

Hình 31.3(f) chỉ ra sai lệch tuyệt đối của các tốc độ tư thế bay. Cả 2 trường hợp có thêm cơ chế thích nghi cho thấy sai lệch tốc độ tư thế bay giảm đáng kể so với khi không có cơ cấu thích nghi.

Mục đích của điều khiển thích nghi đã trình bày trong ví dụ này là nhằm tiến tới LCLD mong muốn. Các hình trên cho thấy rằng hệ tổng quát đạt được có tính ổn định tiệm cận. Hình 31.3(e) minh họa sai lệch tuyệt đối giữa chuyển động thật $\sigma(t)$ và chuyển động tham chiếu tuyến tính lý tưởng $\sigma_r(t)$. Hình này cho thấy sai lệch lớn do việc sử dụng luật điều khiển hệ hở với mô hình hệ thống chưa chính xác. Cơ cấu thích nghi đã cải thiện khả năng bám tức thời lên tới hai bậc biên độ. Khi không có học nhiễu ngoài thì tốc độ suy giảm sai lệch cuối cùng thay đổi rất ít. Sai lệch này sẽ suy giảm về 0. Tuy nhiên, với các hệ số học đã cho, sai lệch sẽ suy giảm với tốc độ chậm hơn nếu không dùng cơ cấu thích nghi. Khi có học nhiễu ngoài thì sự suy giảm sai lệch cuối cùng cải thiện đáng kể do hệ thống có mô hình chính xác của nhiễu hằng thực. Nếu ước lượng mô hình ban đầu càng chính xác thì có thể sử dụng nhiều tốc độ học thích nghi, kết quả thậm chí còn tốt hơn LCLD. Ví dụ này chứng tỏ rằng có thể bám theo LCLD mong muốn rất tốt ngay cả khi hệ thống có mô hình không rõ ràng.

31.7 Điều khiển thích nghi phản hồi đầu ra

Trái với các cách tiếp cận không gian trạng thái, phương pháp vào-ra coi đối tượng như là một hộp đen biến đổi các đầu vào sang không gian đầu ra tương ứng. Lý thuyết ổn định cho hệ phi tuyến nhìn từ quan điểm vào-ra rất quan trọng trong trường hợp thiết kế điều khiển phản hồi đầu ra thích nghi. Giải pháp cho bài toán thiết kế bộ quan sát thích nghi liên quan đến ước lượng trạng thái hệ thống với các tham số chưa biết là bước đi đầu tiên dẫn tới việc giải quyết bài toán điều khiển phản hồi đầu ra. Gần đây đã có những sự tiến bộ trong lĩnh vực này mà ở đó thủ tục thiết kế bộ quan sát thích nghi phi tuyến đã được mở rộng thành trường hợp tổng quát hơn với các hệ số của các tham số chưa biết phụ thuộc vào toàn bộ trạng thái chứ không chỉ trên thành phần được đo [15].

Để mở rộng hơn nữa, một số kết quả có thể được thực hiện bằng cách kết hợp các điều kiện "kiểu thụ động" với các điều kiện kích thích liên tục thông thường. Nội dung chủ yếu trong mục này là khái niệm của tính thụ động, nó gần như là một trình bày vắn tắt về ý tưởng tiêu thụ năng lượng trong cả hệ thống tuyến tính và phi tuyến. Hệ thụ độngcó hầu hết trong các ứng dụng cơ khí và kỹ thuật điện. Một hệ thống cơ khí gồm khối lượng, lò xo và giảm chấn nhớt là một ví dụ điển hình cho hệ thụ động. Sau đây là các định nghĩa.

Định nghĩa: **Chặt tín hiệu**

Cho Y là miền xác định của hàm giá trị thực được định nghĩa trong khoảng $[0, \infty)$. Cho x là một phần tử của Y. Khi đó *chặt* của x tại $T > 0$ được định nghĩa:

$$x_T(t) = \begin{cases} x(t) & 0 \le t \le T \\ 0 & t > T \end{cases}$$

Định nghĩa: **Không gian mở rộng**

Nếu X là một tập con tuyến tính của Y, thì *không gian mở rộng* X_e được định nghĩa bởi tập:

$$\{x \in Y : x_T \in X \text{ với một vài giá trị cố định } T > 0\}$$

Không gian mở rộng L_2 được biểu diễn là L_{2e}.

Định nghĩa: **Tích vô hướng của hai tín hiệu**

Tính vô hướng của hai tín hiệu biến đổi theo thời gian $x, y \in L_{2e}$ được định nghĩa là:

$$\langle x|y\rangle = \int_0^T x^T(\tau)y(\tau)d\tau = \int_0^T x^T(\tau)y(\tau)d\tau$$

Định nghĩa: **Hệ thống bị động**

Một hệ thống với đầu vào $u(t)$ và đầu ra $y(t)$ là *thụ động* nếu:

$$\langle y|u\rangle \geq 0$$

Hệ *là thụ động tuyệt đối đầu vào* nếu $\exists c > 0$ thỏa mãn:

$$\langle y|u\rangle \geq c\|u\|^2$$

Hệ được gọi là *thụ động tuyệt đối đầu ra* nếu $\exists c > 0$ thỏa mãn:

$$\langle y|u\rangle \geq c\|y\|^2$$

31.8 Các bộ quan sát thích nghi và điều khiển phản hồi đầu ra

Bây giờ chúng ta trình bày bài toán quan sát thích nghi phi tuyến theo công thức của Besancon [15]:

$$\begin{aligned} \dot{x} &= f(x,u,t) + g(x,u,t)\theta \\ y &= h(x) \end{aligned} \tag{31.15}$$

trong đó hàm f và g là C^k đối với tất các tham số của nó và θ là một tham số hằng chưa biết. Các biến x, u và y theo thứ tự biểu thị vectơ trạng thái, vectơ đầu vào, vectơ đầu ra. Các tín hiệu đầu vào có thể được giả sử là một số hàm đo được và bị giới hạn. Với khái niệm bộ quan sát thích nghi, chúng ta dẫn đến bài toán khôi phục ước lượng trạng thái $\hat{x}(t)$ sử dụng đầu vào u và đầu ra y với sự có mặt của tham số chưa biết θ sao cho $\lim_{t\to\infty}\|\hat{x}(t) - x(t)\| = 0$. Điều kiện để tồn tại một bộ quan sát như thế hiện đã có trong [15] và được phát biểu như sau. Nếu tồn tại một bộ quan sát tương ứng trong trường hợp mà θ đã biết, và nếu bộ quan sát đó được xác định khi có sai lệch tham số $\tilde{\theta} \doteq \hat{\theta} - \theta$ và hệ thống sai lệch ước lượng trạng thái là thụ động giữa "đầu vào" và sai lệch đầu ra $h(\hat{x}) - y$, thì có thể thiết kế bộ quan sát trạng thái tiệm cận ngay cả khi chưa biết θ. Cùng với yêu cầu thụ động này, hội tụ sai lệch tham số cần thêm kích thích liên tục đối với u.

Kết quả tuyệt vời này có các ứng dụng ngay lập tức trong bài toán theo dõi tư thế tàu không gian mà không có giá trị đo vận tốc góc [16]. Hiện nay được biết đến nhiều nhất là các phương trình của bài toán điều khiển tư thế vật rắn dưới dạng vectơ MRP [17, 18] thỏa mãn điều kiện thụ động giữa vectơ vận tốc góc và vectơ MRP. Một hệ quả rất quan trọng của tính thụ động trong trường hợp này là các luật điều khiển phản hồi cho điều khiển tư thế có thể được thực hiện theo cách xây dựng dựa trên Lyapunov mà không cần đo vận tốc góc.

31.9 Nhận xét

Như đã nói, việc phát triển và ứng dụng của lý thuyết điều khiển thích nghi hiện đại cho hệ phi tuyến tổng quát đã thông qua hướng tiếp cận triết học của các phương pháp luận hệ tuyến tính đang tồn tại rộng rãi. Trong một số trường hợp bị giới hạn như là lý thuyết điều chỉnh, cách tiếp cận của các phương pháp tuyến tính song song này đã thành công lớn. Tuy nhiên, các lĩnh vực nghiên cứu khác như bám quỹ đạo, tổng hợp bộ điều khiển và khôi phục trạng thái thì không có sự thành công như vậy.

Các hệ phi tuyến xuất hiện rất đa dạng và không phải tất cả chúng đều có thể điều khiển được bằng cách mở rộng đơn giản về phương pháp điều khiển thích nghi tuyến tính. Một phương pháp hứa hẹn cho mục đích nghiên cứu xa hơn là để chuẩn hóa việc nghiên cứu các hệ cơ khí, do đó giới hạn lớp các hệ phi tuyến đã nghiên cứu và vì vậy có thể giới thiệu "cấu trúc" và các điều kiện ràng buộc thêm. Trong trường hợp điều khiển phản hồi đầu ra cho hệ phi tuyến tổng quát, các thiết kế riêng của bộ quan sát trạng thái và bộ điều khiển không cần đảm bảo tính ổn định cho tổ hợp của chúng, một số phương pháp có cấu trúc sử dụng biến đổi trạng thái đã được đưa ra giúp lấy lại các tính chất riêng biệt trong nhiều trường hợp [15]. Với kết quả này, các phương pháp có cấu trúc đã nói ở trên cũng cho phép xây dựng các bộ điều khiển bám toàn cục và bán toàn cục dựa trên phản hồi đầu ra. Có thể thấy rằng việc theo đuổi cách tiếp cận tương tự có tiềm năng lớn trong việc cung cấp "chìa khóa" nhằm giải quyết một vài bài toán khác xuất hiện không thuộc các hệ cơ điện tử mà chúng ta còn chưa biết.

Tài liệu tham khảo

[1] Narendra, K. S., "Parameter adaptive control—The End ... or The Beginning?," *Proceedings of the 33rd Conference on Decision and Control.* Lake Buena Vista, Florida. December 1994.

[2] Slotine, J. E. and Li, W., *Applied Nonlinear Control.* Prentice-Hall, Englewood Cliffs, NJ, 1991.

[3] Khalil, H. K., *Nonlinear Systems.* Macmillan, New York, NY, 1992.

[4] Sastry, S. and Bodson, M., *Adaptive Control: Stability, Convergence and Robustness.* Prentice-Hall, 1989.

[5] Tao, G., "A simple alternative proof to the Barbalat Lemma," *IEEE Transactions on Automatic Control,* Vol. 42, No. 5, May 1997, p. 698.

[6] Narendra, K. S. and Annaswamy, A. M., *Stable Adaptive Systems.* Prentice-Hall, 1989.

[7] Ioannou, P. A. and Sun, J., *Stable and Robust Adaptive Control.* Prentice-Hall, Upper Saddle River, NJ, 1995, pp. 85–134.

[8] Astrom, K. J. and Wittenmark, B., *Adaptive Control.* Addison-Wesley, Reading, MA, 1995.

[9] Gantmacher. *The Theory of Matrices,* Vol 1. Chelsea Publishing Company, NY, 1977, pp. 353–354.

[10] Krsti, M., Kanellakopoulos, I., and Kokotovi, P. V., "Transient performance improvement with a new class of adaptive controllers," *Systems & Control Letters,* Vol. 21, 1993, pp. 451–461.

[11] Krtí, M., Kanellakopoulos, I., and Kokotovi, P. V., "Nonlinear design of adaptive controllers for linear systems," *IEEE Transactions on Automatic Control,* Vol. 39, 1994, pp. 738–752.

[12] Kokotovic, P. V., "The joy of feedback: nonlinear and adaptive control," *IEEE Control Systems Magazine,* Vol. 12, No. 3, 1992, pp. 7–17.

[13] Schaub, H., Akella, M. R., and Junkins, J. L., "Adaptive control of nonlinear attitude motions realizing linear closed loop dynamics," *Journal of Guidance, Control and Dynamics,* Vol. 24, No. 1, Jan.–Feb. 2001.

[14] Akella, M. R., Schaub, H., and Junkins, J. L., "Adaptive realization of linear closed loop tracking dynamics in the presence of large system model errors," *Journal of Astronautical Sciences,* Vol. 48, No. 4, 2000.

[15] Besançon, G., "Global output feedback tracking control for a class of Lagrangian systems," *Automatica,* Vol. 36, 2000, pp. 1915–1921.

[16] Akella, M. R., "Rigid body attitude tracking without angular velocity feedback," *Systems & Control Letters,* Vol. 42, No. 4, 2001.

[17] Lizarralde, F. and Wen, J. T., "Attitude control without angular velocity measurement: a passivity approach," *IEEE Transactions on Automatic Control,* Vol. 41, No. 3, 1996, pp. 468–472.

[18] Tsiotras, P., "Further passivity results for the attitude control problem," *IEEE Transactions on Automatic Control,* Vol. 43, No. 11, 1998, pp. 1597–1600.

32

Mạng nơron và hệ thống mờ

32.1 Mạng nơron và hệ mờ 32-1
32.2 Tế bào nơron 32-1
32.3 Mạng nơron truyền thẳng 32-4
32.4 Các thuật toán học cho mạng nơron 32-5
32.5 Các mạng truyền thẳng đặc biệt 32-10
32.6 Mạng nơron hồi qui 32-15
32.7 Hệ mờ 32-17
32.8 Thuật toán di truyền 32-21

Bogdan M. Wilamowski
University of Wyoming

32.1 Mạng nơron và hệ mờ

Sự ra đời của hàng loạt các thiết bị điện tử hiện đại đã làm cho các nhà nghiên cứu dám nghĩ tới việc xây dựng những cỗ máy thông minh hoạt động tương tự như hệ thần kinh của con người. Mục tiêu hấp dẫn này xuất hiện khi McCulloch và Pitts phát triển mô hình nơron tính toán cơ bản vào năm 1943 và khi Hebb giới thiệu *quy tắc học* vào năm 1949. Một thập kỷ sau (1958) Rosenblatt giới thiệu khái niệm **perceptron**. Vào đầu những năm 1960, Widrow và Holf đã phát triển hệ thống thông minh như ADALINE và MADALINE. Vào năm 1965, trong cuốn sách *Learning machines* của Nillson, ông đã tổng kết nhiều nghiên cứu trong thời gian này. Các ấn bản năm 1969 của Mynsky và Paper, với một vài kết quả không tốt, đã làm giảm mạnh sự hấp dẫn của mạng nơron nhân tạo, và các thành tựu trong cơ sở toán học của thuật toán **lan truyền ngược (back-propagation)** của Webos (1974) đã không gây được sự chú ý. Sự phát triển nhanh chóng trong lĩnh vực mạng nơron lại bắt đầu với Hopfield (1982, 1984) với mạng hồi quy, Kohonen (1982) với các thuật toán huấn luyện không giám sát, và thuật toán lan truyền ngược của Rumelhart vào năm 1986.

32.2 Tế bào nơron

Nơron sinh học là một cấu trúc phức tạp, nhận được sự huấn luyện từ các xung mạch trên hàng trăm đầu vào *kích thích* và *ức chế*. Các xung này được tổng hợp với các trọng số khác nhau (hoặc tính trung bình) trong khoảng thời gian của *tổng tiềm tàng*. Nếu giá trị được tổng hợp lớn hơn một ngưỡng, thì nơron tự phát ra một xung để gửi tới nơron lân cận. Vì các xung đến được tính tổng theo thời gian, nên nơron phát chuỗi xung ở tần số cao hơn đối với kích thích cao hơn. Hay nói cách khác, nếu tổng của các giá trị đầu vào có trọng số lớn hơn, thì nơron phát các xung nhanh hơn. Tại cùng một thời điểm, mỗi nơron được mô tả bởi khả năng phi kích thích trong thời gian xác định sau xung kích. Khoảng thời gian này cũng được gọi là *giai đoạn trơ* và được miêu tả chính xác hơn như là một hiện tượng mà sau khi kích thích giá trị ngưỡng tăng lên một giá trị rất cao và sau đó giảm từ từ với một hằng số thời gian xác định.

Giai đoạn trơ ì thiết lập các giới hạn mềm trên tần số của chuỗi xung đầu ra. Với nơron sinh học, thông tin được gửi dưới dạng chuỗi xung đã điều chỉnh tần số.

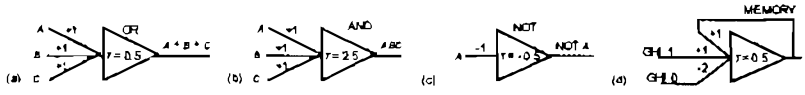
Mô tả hoạt động của nơron theo cách này dẫn đến một mô hình nơron rất phức tạp và không thực tế. Vào năm 1943 McCulloch và Pitts cho thấy rằng thậm chí với một mô hình rất đơn giản ta cũng có thể xây dựng các mạch nhớ và mạch logic. Hơn nữa, các nơron đơn giản với các ngưỡng này thường mạnh hơn các cổng logic thông thường được sử dụng trong máy tính. Mô hình nơron McCulloch-Pitts giả thiết rằng các tín hiệu vào và các tín hiệu ra có giá trị nhị phân 0 hoặc 1. Nếu các tín hiệu vào (được tổng hợp với trọng số dương hoặc âm) có giá trị lớn hơn ngưỡng thì đầu ra nơron được đặt là 1. Ngược lại, nó được đặt là 0.

$$T = \begin{cases} 1, & net \geq T \\ 0, & net \leq T \end{cases} \tag{32.1}$$

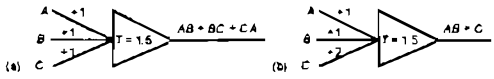
trong đó T là giá trị ngưỡng và giá trị *net* là tổng có trọng số của tất cả các tín hiệu vào.

$$net = \sum_{i=1}^{n} w_i x_i \tag{32.2}$$

Các ví dụ về nơron của McCulloch-Pitts thực hiện các toán tử OR (hoặc), AND (và), NOT (phủ định), và MEMORY (nhớ) được minh họa như hình 32.1. Chú ý rằng cấu trúc của cổng OR và AND có thể đồng nhất. Với các cấu trúc tương tự, các hàm logic khác có thể được thực hiện, như hình 32.2.



HÌNH 32.1 Các toán tử OR, AND, NOT, và MEMORY sử dụng các mạng với mô hình nơron McCulloch-Pitts



HÌNH 32.2 Hàm logic khác được thực hiện bằng mô hình nơron McCulloch-Pitts

Mô hình perceptron có cấu trúc tương tự. Các tín hiệu vào, các trọng số và các giá trị ngưỡng của nó có thể là các giá trị âm hoặc dương. Thông thường, thay vì sử dụng giá trị ngưỡng biến đổi, một đầu vào hằng số với trọng số âm hoặc dương được thêm vào mỗi nơron, như hình 32.2. Trong trường hợp này, giá trị ngưỡng luôn được đặt bằng 0 và giá trị *net* được tính như sau:

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1} \tag{32.3}$$

trong đó w_{n+1} có giá trị bằng giá trị ngưỡng yêu cầu và có dấu ngược lại. Perceptron đơn lớp đã được sử dụng thành công để giải quyết nhiều mẫu bài toán cổ điển. Các hàm kích hoạt ngưỡng cứng được cho bởi:

$$o = f(net) = \frac{\text{sgn}(net) + 1}{2} = \begin{cases} 1, & net \geq 0 \\ 0, & net < 0 \end{cases} \tag{32.4}$$

với các nơron **đơn cực** và

$$o = f(net) = \text{sgn}(net) = \begin{cases} 1, & net \geq 0 \\ -1, & net < 0 \end{cases} \tag{32.5}$$

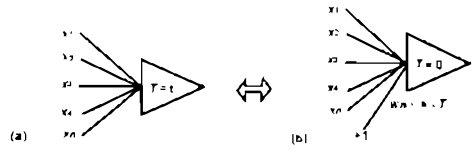
với nơron **lưỡng cực**. Với các kiểu nơron này, hầu hết các thuật toán huấn luyện đã biết chỉ có thể điều chỉnh trọng số trong các mạng đơn lớp.

Các mạng nơron đa lớp thường sử dụng các hàm kích hoạt liên tục, với đơn cực:

$$o = f(net) = \frac{1}{1+\exp(-\lambda net)} \tag{32.6}$$

hoặc lưỡng cực:

$$o = f(net) = \tanh(0.5\lambda net) = \frac{2}{1+\exp(-\lambda net)} - 1 \tag{32.7}$$



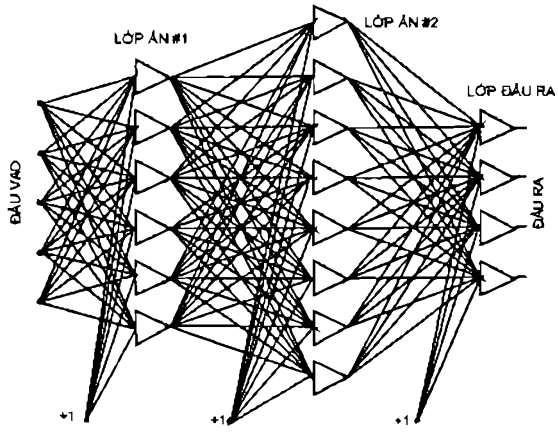
HÌNH 32.3 Tạo ngưỡng với trọng số bổ xung và đầu vào hằng số có giá trị +1: (a) nơron với giá trị ngưỡng T, (b) nơron sửa đổi với ngưỡng T=0 và thêm trọng số bằng –T

Các hàm kích hoạt liên tục cho phép áp dụng các thuật toán huấn luyện dựa trên gradien đối với mạng đa lớp. Các hàm kích hoạt điển hình được minh họa như hình 32.4. Trong trường hợp này, khi sử dụng đầu vào ngưỡng được thêm vào (Hình 32.3(b)), tham số λ có thể được loại bỏ khỏi phương trình (32.6) và (32.7) và độ dốc của đáp ứng nơron có thể được điều khiển bằng cách thay đổi trọng số. Vì vậy, không thực sự cần thiết phải sử dụng nơron với hệ số biến đổi.

Chú ý rằng ngay cả các mô hình nơron với các hàm kích hoạt liên tục cũng khác xa so với nơron sinh học thực tế hoạt động với các chuỗi xung điều chỉnh liên tục.



HÌNH 32.4 Các hàm kích hoạt điển hình: (a) hàm đơn cực ngưỡng cứng, (b) hàm lưỡng cực ngưỡng cứng, (c) hàm đơn cực liên tục, (d) hàm lưỡng cực liên tục



HÌNH 32.5 Một ví dụ của mạng nơron truyền thẳng 3 lớp, đôi khi cũng được hiểu là mạng lan truyền ngược

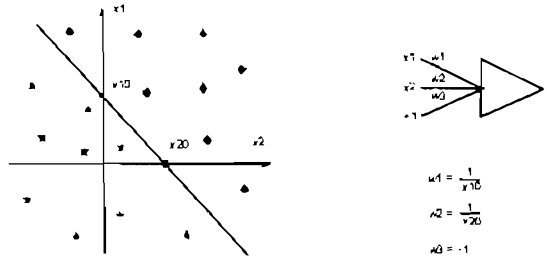
32.3 Mạng nơron truyền thẳng

Mạng nơron truyền thẳng chỉ cho tín hiệu đi theo một hướng. Hơn nữa, hầu hết các mạng nơron truyền thẳng được tổ chức theo các lớp. Một ví dụ của mạng nơron truyền thẳng 3 lớp được minh họa trong hình 32.5. Mạng này gồm các nút đầu vào, 2 lớp ẩn, và một lớp đầu ra.

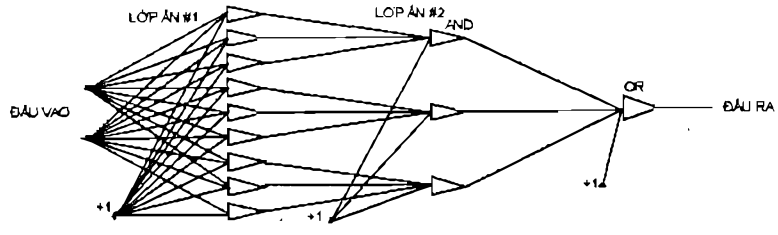
Một nơron đơn có thể tách các mẫu đầu vào thành 2 nhóm, và sự phân tách này là tuyến tính. Ví dụ, với các mẫu được minh họa như hình 32.6, đường phân tách cắt trục x_1 và x_2 tại điểm x_{10} và x_{20}. Sự phân tách này có thể được thực hiện với một nơron có trọng số như sau $w_1 = 1/x_{10}$, $w_2 = 1/x_{20}$, $w_3 = 1$. Trong trường hợp tổng quát với *n* chiều, trọng số là:

$$w_i = \frac{1}{x_{i0}} \text{ với } w_{n+1} = -1 \tag{32.8}$$

Một nơron chỉ có thể chia các mẫu tách biệt tuyến tính. Để chọn một vùng trong không gian đầu vào *n* chiều thì phải sử dụng nhiều hơn $n + 1$ nơron. Nếu nhiều nhóm đầu vào được chọn, thì số nơron trong lớp đầu vào (ẩn) phải tăng lên tương ứng. Nếu số nơron trong lớp ẩn không bị giới hạn thì tất cả các bài toán phân loại có thể được giải quyết bằng cách sử dụng mạng 3 lớp. Một ví dụ về mạng nơron, phân loại 3 nhóm trong không gian 2 chiều, được minh họa như hình 32.7. Các nơron ở lớp ẩn đầu tiên tạo ra các đường phân tách giữa các nhóm đầu vào.



HÌNH 32.6 Minh họa đặc tính phân tách tuyến tính trong không gian 2 chiều bằng một nơron đơn



HÌNH 32.7 Một ví dụ của mạng nơron 3 lớp với 2 đầu vào để phân loại 3 nhóm khác nhau. Mạng này có thể được tổng quát hóa để giải tất cả các bài toán phân loại

Các nơron trong lớp ẩn thứ 2 thực hiện toán tử AND, như hình 32.1(b). Các nơron đầu ra thực hiện toán tử OR, như hình 32.1(a), cho mỗi nhóm. Đặc tính phân tách tuyến tính của các nơron đã tạo ra các bài toán đặc biệt khó cho các mạng nơron, như là XOR, tính toán chẵn lẻ một vài bit, hoặc tách các mẫu nằm trên 2 đường xoắn ốc gần nhau.

Mạng nơron truyền thẳng cũng được sử dụng để biến đổi phi tuyến tính (ánh xạ) một biến đầu vào đa chiều thành một biến đầu ra đa chiều khác. Theo lý thuyết, mọi ánh xạ vào-ra đều có thể nếu mạng nơron có đủ số nơron trong các lớp ẩn (kích thước của lớp đầu ra được đặt bằng số chiều của đầu ra yêu cầu). Trong thực tế, đây không phải là một nhiệm vụ dễ dàng.

Hiện tại, không có phương pháp nào thỏa đáng để định nghĩa số nơron được sử dụng trong lớp ẩn như thế nào. Thông thường, việc này được làm bởi một phương pháp dò lỗi. Nói chung, ta vẫn biết rằng nếu nhiều nơron được sử dụng hơn, thì nhiều hình dạng phức tạp được ánh xạ hơn. Ngược lại, các mạng với số nơron lớn sẽ mất tính tổng quát hóa, và có vẻ như các mạng đó ánh xạ cả nhiễu của đầu vào.

32.4 Các thuật toán học cho mạng nơron

Tương tự như các nơron sinh học, các trọng số trong nơron nhân tạo được điều chỉnh trong suốt quá trình huấn luyện. Nhiều thuật toán học khác nhau đã được phát triển, và chỉ một số ít phù hợp với các mạng nơron đa lớp. Một số chỉ sử dụng các tín hiệu cục bộ trong nơron, một số khác yêu cầu thông tin từ phía đầu ra, một số thuật toán yêu cầu một bộ giám sát để biết các đầu ra là gì đối với các mẫu đã cho, và các thuật toán không giám sát khác không cần thông tin như vậy. Các quy tắc học sẽ được giới thiệu trong các phần sau.

Qui tắc học Hebbian

Quy tắc học Hebb (1949) dựa trên giả thiết rằng nếu 2 nơron gần nhau được kích thích và tắt kích thích tại cùng một thời điểm thì trọng số kết nối các nơron này phải tăng lên. Với các nơron hoạt động trong giai đoạn ngược nhau, trọng số giữa chúng phải giảm. Nếu không có tín hiệu tương quan, thì trọng số phải được giữ nguyên. Giả thiết này được miêu tả bởi công thức:

$$\Delta w_{ij} = c x_i o_j \tag{32.9}$$

trong đó:

w_{ij} = là trọng số từ nơron thứ i đến nơron thứ j

C = là hằng số học

x_i = tín hiệu trên đầu vào thứ i

o_j = tín hiệu đầu ra thứ j

Quá trình huấn luyện thường bắt đầu với tất cả trọng số có giá trị đặt là 0. Quy tắc học này được sử dụng cho cả các nơron ngưỡng cứng và nơron ngưỡng mềm. Do đáp ứng mong muốn của các nơron không được sử dụng trong quá trình học nên đây là **quy tắc học không giám sát**. Giá trị tuyệt đối của trọng số thường tỷ lệ với thời gian học, đó là điều không mong muốn.

Quy tắc học tương quan

Quy tắc học tương quan dựa trên nguyên tắc tương tự với quy tắc học Hebbian. Nó giả thiết rằng trọng số giữa các nơron đáp ứng đồng thời phải dương ở mức độ lớn, và trọng số giữa các nơron phản ứng ngược nhau phải âm ở mức độ lớn. Ngược lại với quy tắc Hebb, quy tắc tương quan là **học được giám sát**. Thay vì đáp ứng thực, o_j, đáp ứng mong muốn, d_j, được sử dụng để tính sự thay đổi của trọng số :

$$\Delta w_{ij} = c x_i d_j \tag{32.10}$$

Thuật toán huấn luyện này thường bắt đầu với giá trị khởi tạo của các trọng số là 0.

Quy tắc học Instar

Nếu các véc tơ đầu vào và các trọng số được chuẩn hóa, hoặc chúng chỉ có 2 giá trị nhị phân (-1 hoặc +1), thì *net* sẽ có giá trị dương lớn nhất khi các trọng số và các tín hiệu vào giống nhau. Do đó, các trọng số chỉ phải thay đổi khi chúng khác tín hiệu vào.

$$\Delta w_i = c(x_i - w_i) \tag{32.11}$$

Chú ý rằng, thông tin được yêu cầu cho các trọng số chỉ được lấy từ các tín hiệu đầu vào. Đây là một thuật toán học rất cục bộ và không giám sát.

Quy tắc WTA (Winner Takes All)

WTA là một sự cải biến của thuật toán instar trong đó các trọng số chỉ được thay đổi với nơron có giá trị *net* cao nhất. Trọng số của các nơron còn lại không thay đổi. Đôi khi thuật toán này được thay đổi theo cách mà một số nơron có giá trị *net* cao nhất được biến đổi tại cùng một thời điểm. Mặc dù đây là một thuật toán không giám sát do không biết đầu ra mong muốn, nhưng cũng cần một "trọng tài" hoặc "người giám sát" để tìm ra nơron *chiến thắng* có giá trị *net* lớn nhất. Thuật toán WTA được phát triển bởi Kohonen (1982), thường được sử dụng cho các ứng dụng tổng hợp hoặc phân tách thống kê dữ liệu đầu vào một cách tự động.

Quy tắc học Outstar

Trong quy tắc học Outstar, trọng số được kết nối tới một nút xác định phải bằng đầu ra mong muốn của các nơron liên kết tới các trọng số đó:

$$\Delta w_{ij} = c(d_j - w_{ij}) \tag{32.12}$$

trong đó d_j là đầu ra nơron mong muốn và c là hằng số học, hằng số này giảm dần trong quá trình học. Đây là quy tắc học có giám sát vì các đầu ra mong muốn phải được biết. Cả 2 quy tắc học Instar và Outstar đều được phát triển bởi Grossberg (1969).

Quy tắc học Widrow-Hoff LMS

Vào những năm (1960, 1962), Widrow và Hoff đã phát triển một thuật toán huấn luyện có giám sát, cho phép huấn luyện nơron với đáp ứng mong muốn. Quy tắc này được xây dựng sao cho bình phương của hiệu giá trị *net* và giá trị đầu ra được tối thiểu hóa.

$$Error_j = \sum_{p=1}^{P} (net_{jp} - d_{jp})^2 \tag{32.13}$$

trong đó:

$Error_j$ = là sai lệch của nơron thứ j

P = số mẫu sử dụng

d_{jp} = đầu ra mong muốn của nơron thứ j khi mẫu thứ p được cung cấp

net = được cho bởi phương trình (32.2)

Quy tắc này cũng được biết như là quy tắc bình phương trung bình nhỏ nhất (LMS- Least mean square). Bằng cách tính đạo hàm của phương trình (32.13) đối với w_{ij}, ta tìm được công thức cho sự thay đổi trọng số như sau:

$$\Delta w_{ij} = cx_i \sum_{p=1}^{P} (d_{jp} - net_{jp}) \tag{32.14}$$

Chú ý rằng sự thay đổi trọng số Δw_{ij} là tổng các thay đổi từ từng mẫu riêng biệt. Vì vậy, có thể sửa lại trọng số sau khi mỗi mẫu riêng biệt được áp dụng. Quá trình này được hiểu như là *cập nhật tăng dần*; *cập nhật tích lũy* là khi các trọng số được thay đổi sau khi tất cả các mẫu đã được áp dụng. Cập nhật tăng dần thường dẫn tới kết quả nhanh hơn, nhưng nó nhạy cảm với thứ tự các mẫu được thực hiện. Nếu hằng số học c được chọn là nhỏ, thì hai phương pháp cho cùng một kết quả. Quy tắc LMS làm việc tốt với tất cả các kiểu hàm kích hoạt. Quy tắc này cố gắng ép buộc giá trị *net* bằng giá trị mong muốn. Đôi khi đây không phải là vấn đề chính. Giá trị *net* bằng bao nhiêu không quan trọng nhưng giá trị *net* âm hay dương lại là điều quan trọng.

Ví dụ, một giá trị *net* lớn với dấu phù hợp sẽ cho đầu ra đúng và sai lệch lớn như định nghĩa bởi phương trình (32.13) và đây có thể là giải pháp hoàn hảo.

Hồi quy tuyến tính

Quy tắc học LMS yêu cầu hàng trăm hoặc hàng nghìn phép lặp, sử dụng công thức (32.14), trước khi nó hội tụ tới nghiệm phù hợp. Sử dụng quy tắc hồi quy tuyến tính, kết quả tương tự có thể đạt được chỉ sau một bước tính.

Xét một nơron và sử dụng ký hiệu véc tơ cho một bộ các mẫu đầu vào **X** được cung cấp thông qua véc tơ trọng số **w**, véc tơ **net** của các giá trị *net* được tính như sau:

$$\mathbf{Xw} = \mathbf{net} \tag{32.15}$$

trong đó

X = là mảng chữ nhật $(n+1)\,x\,p$

n = là số đầu vào
p = là số mẫu

Chú ý rằng kích thước của các mẫu đầu vào được tăng thêm một, và trọng số thêm này tương ứng với ngưỡng (xem hình 32.3 (b)). Phương pháp này, tương tự như quy tắc LMS, giả thiết một hàm kích hoạt tuyến tính, và véc tơ **net của** các giá trị *net* phải bằng giá trị đầu ra mong muốn **d**.

$$\mathbf{Xw} = \mathbf{d} \tag{32.16}$$

Thông thường $p > n+1$, và phương trình trước có thể được giải chỉ trong trường hợp sai lệch bình phương trung bình nhỏ nhất. Bằng cách sử dụng đại số véc tơ, nghiệm được cho bởi:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{d} \tag{32.17}$$

Khi phương pháp truyền thống được sử dụng, hệ p phương trình với $n+1$ biến, phương trình (32.16), phải được chuyển sang hệ $n+1$ phương trình với $n+1$ biến:

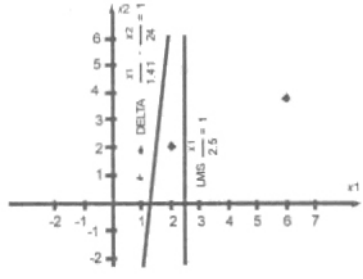
$$\mathbf{Yw} = \mathbf{z} \tag{32.18}$$

khi đó các thành phần của ma trận **Y** và vector **z** được cho bởi

$$y_{ij} = \sum_{p=1}^{P} x_{ip}x_{jp}, \quad z_i = \sum_{p=1}^{P} x_{ip}d_p \tag{32.19}$$

Các trọng số được cho bởi (32.17) hoặc tìm được bằng cách giải phương trình (32.18).

Quy tắc học Delta



HÌNH 32.8 Ví dụ với sự so sánh kết quả đạt được bằng cách sử dụng các thuật toán huấn luyện LMS và Delta. Chú ý rằng LMS không thể tìm được một giải pháp thích hợp

Phương pháp LMS giả sử có hàm kích hoạt tuyến tính **net=o**, và giải pháp đạt được đôi khi là không tối ưu, như minh họa trên hình 32.8 đối với trường hợp hai chiều đơn giản, với bốn mẫu phân loại thành hai nhóm. Trong kết quả có được sử dụng thuật toán LMS, một mẫu không được phân loại.

Nếu sai lệch được định nghĩa là:

$$Error_j = \sum_{p=1}^{P} (o_{jp} - d_{jp})^2 \tag{32.20}$$

thì đạo hàm của sai lệch đối với trọng số w_{ij} là:

$$\frac{d\,Error}{dw_{ij}} = 2\sum_{p=1}^{P} (o_{jp} - d_{jp}) \frac{df(net_{jp})}{d\,net_{jp}} x_i \tag{32.21}$$

do $o = f(net)$ và *net* được cho bởi phương trình 32.2. Chú ý rằng đạo hàm này tỷ lệ với đạo hàm của hàm kích hoạt $f'(net)$. Vì vậy phương pháp này chỉ có thể áp dụng đối với hàm kích hoạt liên tục và không thể sử dụng với hàm kích hoạt cứng trong phương trình (32.4) và (32.5). Ở khía cạnh này, phương pháp LMS tổng quát hơn. Đạo hàm của hàm kích hoạt liên tục phổ biến nhất là:

$$f' = o(1-o) \tag{32.22}$$

đối với trường hợp đơn cực, phương trình (32.6), và

$$f' = 0.5(1-o^2) \tag{32.23}$$

đối với trường hợp lưỡng cực, phương trình (32.7)

Sử dụng phương pháp tích lũy, trọng số nơron w_{ij} sẽ được thay đổi với hướng của gradien:

$$\Delta w_{ij} = cx \sum_{p=1}^{P} (d_{jp} - o_{jp}) f'_{jp} \tag{32.24}$$

Trong trường hợp huấn luyện tăng dần đối với mỗi mẫu được áp dụng

$$\Delta w_{ij} = cx_i f'_j (d_j - o_j) \tag{32.25}$$

sự thay đổi trọng số tỉ lệ với tín hiệu đầu vào x_i, sự khác nhau giữa đầu ra mong muốn và đầu ra thực tế d_{jp} - o_{jp}, và đạo hàm của hàm kích hoạt là f'_{jp}. Tương tự như luật LMS, trọng số có thể được cập nhật theo hai phương pháp tích lũy và tăng dần. So sánh với luật LMS, luật delta luôn dẫn tới một giải pháp gần tối ưu. Như minh họa trên hình 32.8, khi luật delta được sử dụng, cả bốn mẫu đều được phân loại chính xác.

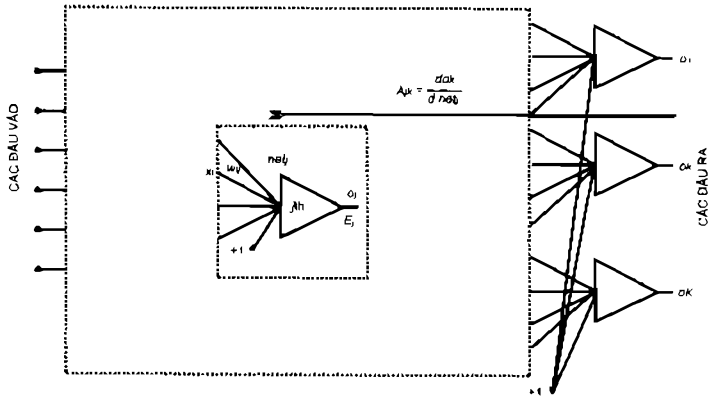
Học lan truyền ngược sai lệch

Quy tắc học delta có thể được tổng quát hóa cho các mạng đa lớp. Bằng cách sử dụng phương pháp tương tự với quy tắc học Delta, gradien của sai lệch toàn cục có thể được tính theo mỗi trọng số trong mạng:

$$\Delta w_{ij} = cx_i f'_j E_j \tag{32.26}$$

Trong đó:

c = hằng số học
x_i = tín hiệu đầu vào nơron thứ i
f'_j = đạo hàm của hàm kích hoạt



HÌNH 32.9 Minh họa khái niệm tính toán hệ số trong mạng nơron

Sai lệch tích lũy E_j của đầu ra nơron được cho bởi:

$$E_j = \frac{1}{f_j'}\sum_{k=1}^{K}(o_k - d_k)A_{jk} \tag{32.27}$$

trong đó K là số đầu ra của mạng và A_{jk} là hệ số khuếch đại tín hiệu nhỏ từ đầu vào của nơron thứ j tới đầu ra thứ k của mạng, như minh họa trên hình 32.9. Sự tính toán sai lệch lan truyền ngược bắt đầu tại lớp đầu ra và sai lệch tích lũy được tính ngược dần từng lớp một cho đến lớp đầu vào. Phương pháp này khó thực hiện bằng phần cứng. Do vậy, một phương pháp đơn giản hơn là tìm hệ số khuếch đại tín hiệu từ đầu vào của nơron thứ j tới mỗi đầu ra của mạng như minh họa trên hình 32.9. Trong trường hợp này, trọng số được hiệu chỉnh bằng cách sử dụng:

$$\Delta w_{ij} = cx_i\sum_{k=1}^{K}(o_k - d_k)A_{jk} \tag{32.28}$$

Chú ý rằng công thức này là tổng quát mà không cần quan tâm nơron có được sắp xếp trong các lớp hay không. Một cách để tìm hệ số khuếch đại A_{jk} là đưa ra một sự thay đổi tăng dần trên đầu vào của nơron thứ j và quan sát sự thay đổi ở đầu ra thứ k của mạng. Quá trình này chỉ yêu cầu lan truyền tín hiệu thuận, và dễ dàng để thực hiện trên phần cứng. Một cách khác là tính hệ số khuếch đại trên mỗi lớp và tìm tích của chúng. Cách này tính toán ít hơn so với việc tính toán sai lệch tích lũy trong thuật toán lan truyền ngược sai lệch.

Thuật toán lan truyền ngược có xu hướng dao động. Để làm mịn quá trình, trọng số tăng thêm Δw_{ij} có thể được hiệu chỉnh theo Rumelhart, Hinton, và Wiliams (1986):

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) + \alpha\Delta w_{ij}(n-1) \tag{32.29}$$

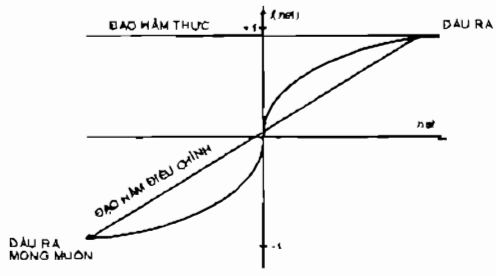
hoặc theo Sejnowski và Rosenberg (1987)

$$w_{ij}(n+1) = w_{ij}(n) + (1-\alpha)\Delta w_{ij}(n) + \alpha\Delta w_{ij}(n-1) \tag{32.30}$$

trong đó α là số hạng xung lượng.

Thuật toán lan truyền ngược có thể được tăng tốc rất nhanh khi tìm được thành phần của gradien, trọng số được hiệu chỉnh dọc theo hướng gradien đến khi tìm được giá trị nhỏ nhất. Quá trình này có thể được thực hiện mà không cần tính toán hướng gradien tại mỗi bước. Thành phần gradien mới được tính khi tìm thấy giá trị nhỏ nhất theo hướng của gradien liền trước. Quá trình này chỉ có thể được áp dụng đối với sự hiệu chỉnh trọng số tích lũy. Một phương pháp để tìm giá trị nhỏ nhất dọc theo hướng gradien là *quá trình leo cây tìm lối* đối với ba điểm dọc theo hướng gradien và sau đó sử dụng xấp xỉ parabol, nhảy trực tiếp tới giá trị nhỏ nhất. Một thuật toán học nhanh sử dụng phương pháp trên đã được phát triển bởi Fahlman (1988) và nó còn được gọi là thuật toán *Quickprop*.

Thuật toán lan truyền ngược có nhiều nhược điểm dẫn đến việc hội tụ rất chậm. Một trong những nhược điểm lớn nhất của thuật toán lan truyền ngược là quá trình học hầu như không áp dụng được đối với các nơron có đáp ứng với kết quả sai lớn nhất. Ví dụ, nếu giá trị ở đầu ra nơron gần bằng +1 và đầu ra mong muốn gần bằng -1 thì hệ số khuếch đại nơron $f'(net)= 0$ và tín hiệu sai lệch không thể truyền ngược và vì vậy quá trình học không hiệu quả. Để giải quyết khó khăn này, một phương pháp để tính đạo hàm đã được giới thiệu bởi Wilamowski và Torvik (1993). Đạo hàm được tính theo độ dốc của đường thẳng nối điểm giá trị đầu ra với điểm giá trị mong muốn, minh họa trên hình 32.10.



HÌNH 32.10 Minh họa tính toán đạo hàm hiệu chỉnh cho sự hội tụ nhanh của thuật toán lan truyền ngược sai lệch

$$f_{modif} = \frac{o_{desired} - o_{actual}}{net_{desired} - net_{actual}} \tag{32.31}$$

Chú ý rằng đối với sai lệch nhỏ, phương trình (32.31) hội tụ tới đạo hàm của hàm kích hoạt tại giá trị đầu ra. Nếu chiều của hệ thống tăng lên thì cực tiểu cục bộ sẽ giảm dần. Hiện tượng trên như một cái bẫy ở cực tiểu cục bộ và gây ra khó khăn cho bài toán hội tụ trong thuật toán lan truyền ngược sai lệch.

32.5 Các mạng truyền thẳng đặc biệt

Mạng lan truyền ngược đa lớp, như trên hình 32.5, là một một mạng truyền thẳng thường được sử dụng. Mạng gồm nhiều nơron với hàm kích thích liên tục kiểu chữ S được thể hiện trên hình 32.4(c) và 32.4(d). Trong hầu hết các trường hợp, chỉ cần một lớp ẩn và số nơron trong lớp ẩn được lựa chọn tỷ lệ với độ phức tạp của bài toán. Số lượng nơron trong lớp ẩn thường được tìm bởi quá trình dò lỗi. Quá trình huấn luyện được bắt đầu với tất cả các trọng số được lựa chọn ngẫu nhiên với giá trị nhỏ nhất và sử dụng thuật toán lan truyền ngược sai lệch để tìm ra đáp án. Khi quá trình học không hội tụ, quá trình huấn luyện được lặp lại với một bộ các trọng số được lựa chọn ngẫu nhiên mới. Nguyen và Widrow (1990) đề xuất một phương pháp thực nghiệm cho giá trị ban đầu của trọng số mạng hai lớp. Trong lớp thứ hai, trọng số được lựa chọn ngẫu nhiên trong dải từ -0.5 tới +0.5. Trong lớp thứ nhất, trọng số ban đầu được tính như sau:

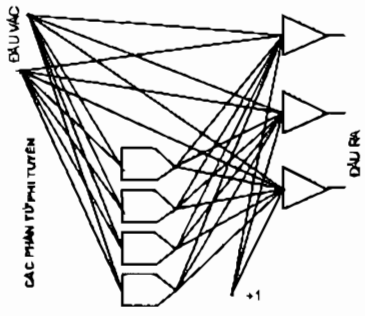
$$w_{ij} = \frac{\beta z_{ij}}{\left\| z_j \right\|}, \quad w_{(n+1)j} = random(-\beta, +\beta) \tag{32.32}$$

trong đó z_{ij} là số ngẫu nhiên từ -0.5 tới +0.5 và hệ số tỷ lệ β được cho bởi

$$\beta = 0.7 p^{1/N} \tag{32.33}$$

ở đây n là số đầu vào và N là số nơron ẩn trong lớp thứ nhất. Kiểu khởi tạo trọng số như này thường dẫn đến kết quả nhanh hơn.

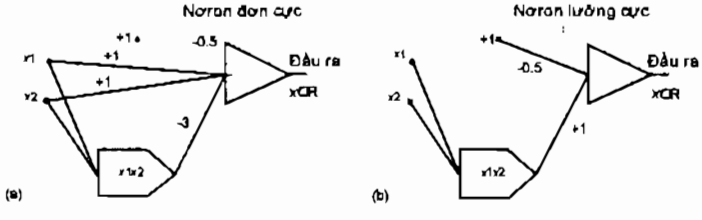
Để có giải pháp phù hợp với mạng lan truyền ngược thì tốn rất nhiều công sức với các cấu trúc mạng khác nhau và các trọng số ngẫu nhiên ban đầu khác nhau. Một điều quan trọng là mạng được huấn luyện thường có tính tổng quan. Điều đó có nghĩa rằng mạng được huấn luyện cũng có thể xử lý chính xác các mẫu không được sử dụng cho huấn luyện. Do vậy trong quá trình huấn luyện, một số dữ liệu được loại khỏi mẫu huấn luyện và những dữ liệu này được sử dụng cho việc kiểm tra. Kết quả của mạng lan truyền ngược thường phụ thuộc vào may rủi. Điều này đã khích lệ các nhà nghiên cứu phát triển mạng truyền thẳng, nó có độ tin cậy cao hơn. Một số mạng truyền thẳng sẽ được giới thiệu trong các phần sau.



HÌNH 32.11 Mạng liên kết hàm

Mạng liên kết hàm (Functional Link Network)

Mạng nơron đơn lớp khá dễ huấn luyện, nhưng những mạng này chỉ có thể giải quyết các bài toán tuyến tính riêng lẻ. Một giải pháp có khả năng ứng dụng cho các bài toán phi tuyến đã được giới thiệu bởi Nilsson (1965) và được phát triển thành công bởi Pao (1989) bằng cách sử dụng mạng liên kết hàm như trên hình 33.11. Sử dụng những số hạng phi tuyến với các hàm xác định ban đầu, số lượng đầu vào thực được cung cấp cho mạng nơron đơn lớp được tăng lên. Trường hợp đơn giản nhất, các yếu tố phi tuyến là những số hạng bậc cao của các mẫu đầu vào. Chú ý rằng mạng liên kết hàm có thể được xử lý như mạng đơn lớp, trong đó dữ liệu đầu vào tăng thêm được tạo ra bằng cách dùng các biến đổi phi tuyến. Quá trình học cho mạng đơn lớp dễ và nhanh. Hình 32.12 minh họa bài toán XOR được giải quyết bằng mạng liên kết hàm. Chú ý rằng khi sử dụng phương pháp liên kết hàm, bài toán khó này trở nên đơn giản. Vấn đề với mạng liên kết hàm là việc lựa chọn thành phần phi tuyến thích hợp không phải là một nhiệm vụ đơn giản. Trong nhiều trường hợp thực tế, không khó để dự đoán kiểu biến đổi dữ liệu đầu vào nào có thể tuyến tính hóa vấn đề, do vậy phương pháp liên kết hàm được sử dụng.



HÌNH 32.12 Mạng liên kết hàm cho giải pháp của bài toán XOR: (a) sử dụng tín hiệu đơn cực, (b) sử dụng tín hiệu lưỡng cực

Kiểu truyền thẳng của mạng lan truyền ngược

Mạng lan truyền ngược được đề xuất đầu tiên bởi Hecht-Nilsen (1987). Phần này sẽ giới thiệu một kiểu truyền thẳng được miêu tả bởi Zurada (1992). Mạng này được minh họa trên hình 32.13, đòi hỏi số nơron ẩn bằng số mẫu đầu vào hoặc chính xác hơn là số nhóm đầu vào.

Lớp thứ nhất được xem như là lớp Kohonen với các nơron đơn cực. Trong lớp này, duy nhất nơron chiến thắng được tích cực. Thứ hai là lớp Grossberg Outstar. Lớp Kohonen có thể được huấn luyện trong chế độ không giám sát nhưng điều này không quan trọng. Khi mẫu đầu vào nhị phân được xem xét, trọng số đầu vào phải chính xác bằng mẫu đầu vào. Trong trường hợp này:

$$net = \mathbf{x}'\mathbf{w} = [n - 2HD(\mathbf{x}, \mathbf{w})] \tag{32.34}$$

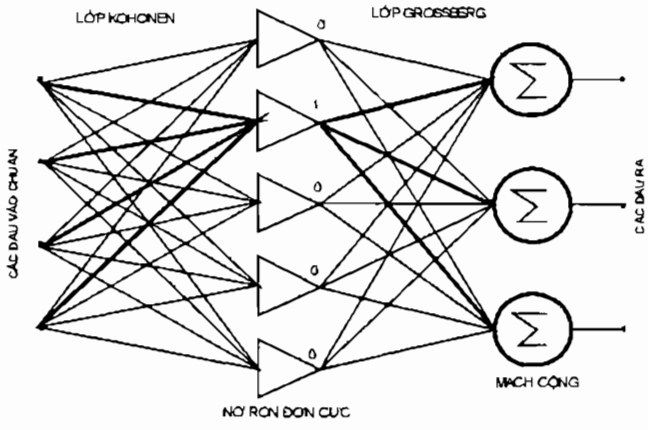
trong đó

n = số lượng đầu vào

$\mathbf{w}$= trọng số

$\mathbf{x}$= véc tơ đầu vào

HD($\mathbf{w}$, $\mathbf{x}$) = *khoảng cách Hamming* giữa mẫu và trọng số.



HÌNH 32.13 Mạng lan truyền ngược

Đối với nơron ở lớp đầu vào chỉ phản ứng với mẫu được tích trữ, giá trị ngưỡng đối với nơron này là:

$$w_{(n+1)} = -(n-1) \tag{32.35}$$

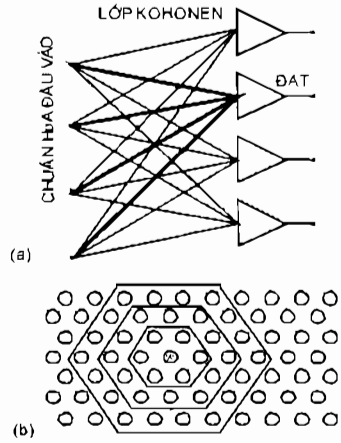
Nếu yêu cầu các nơron cũng phải phản ứng với những mẫu tương tự, thì ngưỡng nên là $w_{n+1} = -\left[n-(1+HD)\right]$, trong đó HD là khoảng cách Hamming định nghĩa một dải giống nhau. Do với một mẫu đầu vào chỉ có một nơron ở lớp thứ nhất có thể có giá trị 1 và những nơron còn lại có giá trị 0, nên trọng số ở lớp đầu ra bằng mẫu đầu ra được yêu cầu.

Mạng với hàm kích hoạt đơn cực ở lớp thứ nhất làm việc như một bảng tra cứu. Khi hàm kích hoạt tuyến tính (hoặc không có hàm kích hoạt nào) được sử dụng trong lớp thứ hai thì mạng cũng có thể được xem như là một bộ nhớ tương tự. Với một địa chỉ được áp dụng ở đầu vào như một véc tơ nhị phân thì tập các giá trị tương tự được lưu trữ, như các trọng số trong lớp thứ hai, có thể được phục hồi chính xác. Mạng lan truyền ngược truyền thẳng cũng có thể sử dụng các đầu vào tương tự nhưng trong trường hợp này tất cả dữ liệu đầu vào phải được chuẩn hóa.

$$\mathbf{w}_i = \hat{\mathbf{x}}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} \tag{32.36}$$

Mạng lan truyền ngược rất dễ thiết kế. Số nơron trong lớp ẩn bằng số lượng mẫu (nhóm). Trọng số ở lớp đầu vào bằng mẫu đầu vào và trọng số ở lớp đầu ra bằng mẫu đầu ra. Mạng đơn giản này có thể được dùng cho ứng dụng tạo mẫu nhanh. Mạng lan truyền ngược thường có

nhiều nơron ẩn hơn yêu cầu. Tuy nhiên, những nơron ẩn dư thừa cũng được sử trong mạng truyền thẳng phức tạp hơn như mạng nơron xác suất (Probabilistic Neural Network-PNN) Specht (1990) hoặc mạng nơron hồi qui tổng quát (General Regression Neural Networks-GRNN) Specht (1992).



HÌNH 32.14 Kiến trúc WTA đối với bài toán phân tách nhóm trong chế độ huấn luyện không giám sát: (a) kết nối mạng, (b) mạng đơn lớp được tổ chức thành hình sáu cạnh

Kiến trúc WTA

Mạng WTA (Winner Take All) được đề xuất bởi Kohonen (1988). Đây là mạng đơn lớp cơ bản được sử dụng trong thuật toán huấn luyện không giám sát để tìm tính chất thống kê của dữ liệu đầu vào, hình 32.14 (a). Ở bước đầu tiên, tất cả dữ liệu đầu vào được chuẩn hóa do vậy độ dài của mỗi véc tơ đầu vào là như nhau và thường bằng 1, phương trình 32.36. Hàm tích cực của nơron là luỡng cực và liên tục. Quá trình học bắt đầu với những giá trị nhỏ ngẫu nhiên của trọng số. Trong suốt quá trình học, trọng số chỉ thay đổi với các nơron có giá trị lớn nhất ở đầu ra – nơron chiến thắng.

$$\Delta \mathbf{w}_{w} = c(\mathbf{x} - \mathbf{w}_{w}) \tag{32.37}$$

trong đó:

$\mathbf{w}_{w}$ = trọng số của nơron chiến thắng
$\mathbf{x}$= véc tơ đầu vào
c = hằng số học

Thông thường, mạng đơn lớp này được tổ chức thành dạng lớp hai chiều, như trên hình 32.14(b). Hình sáu cạnh thường được chọn để bảo đảm sự tương tác giữa các nơron. Thuật toán cũng được được thay đổi để không chỉ các nơron chiến thắng mà những nơron lân cận cũng được phép thay đổi trọng số. Tại cùng một thời điểm, hằng số học c ở phương trình 32.37 giảm theo khoảng cách từ nơron chiến thắng. Sau quá trình huấn luyện không giám sát này, lớp Kohonen có thể tổ chức dữ liệu thành từng nhóm. Đầu ra của lớp Kohonen sau đó được kết nối tới mạng truyền thẳng một lớp hoặc hai lớp với thuật toán truyền ngược sai lệch. Tổ chức dữ liệu ban đầu trong lớp WTA thường dẫn đến quá trình huấn luyện nhanh của những lớp theo sau.

Kiến trúc tương quan tầng

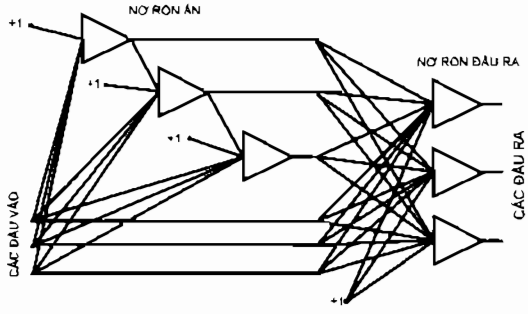
Kiến trúc tương quan tầng được đề xuất bởi Fahlman và Lebiere (1990). Quá trình xây dựng mạng bắt đầu với một mạng nơron đơn lớp và các nơron ẩn được thêm vào khi cần thiết. Kiến trúc mạng được minh họa trên hình 32.15. Ở mỗi bước huấn luyện, một nơron ẩn được

thêm vào và trọng số của nó được điều chỉnh để làm cực đại biên độ tương quan giữa đầu ra nơron ẩn mới và tín hiệu sai lệch dư ở đầu ra mạng cần loại trừ. Tham số tương quan S phải đạt được cực đại:

$$S = \sum_{o=1}^{O} \left| \sum_{p=1}^{P} (V_p - \bar{V})(E_{po} - \bar{E}_o) \right| \tag{32.38}$$

Trong đó:

O = số lượng đầu ra mạng
P = số lượng mẫu huấn luyện
V_p = đầu ra trên nơron ẩn mới
E_{po} = sai lệch trên đầu ra của mạng



HÌNH 32.15 Kiến trúc tương quan tầng

$\bar{V}$ và $\bar{E}_0$ là giá trị trung bình của V_p và E_{po}. Bằng cách tìm ra gradien, $\delta S / \delta w_i$, trọng số, sự điều chỉnh cho các nơron mới có thể tìm được như sau:

$$\Delta w_i = \sum_{o=1}^{O} \sum_{p=1}^{P} \sigma_o (E_{po} - \bar{E}_o) f'_p x_{ip} \tag{32.39}$$

trong đó:

σ_o = tín hiệu tương quan giữa giá trị đầu ra nơron mới và đầu ra nơron.

f'_p = đạo hàm của hàm tích cực cho mẫu p

x_{ip} = tín hiệu đầu vào

Nơron đầu ra được huấn luyện bằng thuật toán Delta hoặc Quickprop. Mỗi nơron ẩn chỉ được huấn luyện một lần và sau đó trọng số của chúng được cố định (đóng băng). Quá trình xây dựng mạng và học được hoàn thành khi đạt được kết quả mong muốn.

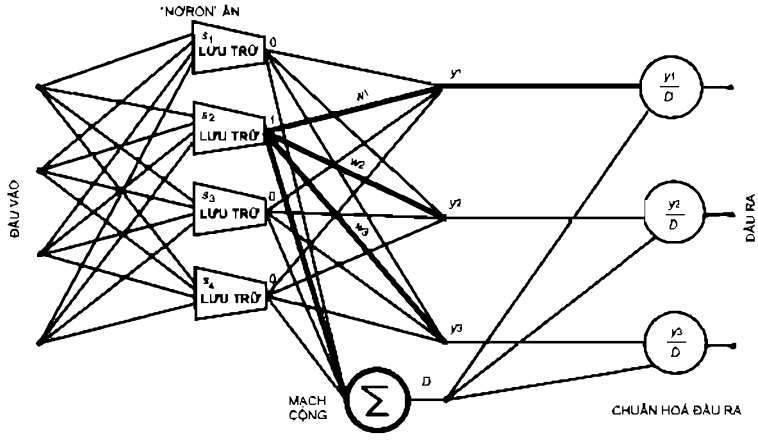
Mạng RBFN (Radial Basis Function Networks)

Cấu trúc của RBFN được thể hiện trên hình 32.16. Kiểu mạng này thường chỉ có một lớp ẩn với các nơron đặc biệt. Mỗi nơron này chỉ đáp ứng với tín hiệu đầu vào gần với mẫu lưu trữ. Tín hiệu đầu ra h_i của nơron ẩn thứ i được tính bằng công thức sau:

$$h_i = \exp\left(\frac{\|\mathbf{x} - \mathbf{s}_i\|^2}{2\sigma^2} \right) \tag{32.40}$$

trong đó:

$\mathbf{x}$ = véc tơ đầu vào
$\mathbf{s}_i$ = mẫu lưu trữ thể hiện tâm của nhóm i
σ_i = bán kính của nhóm



HÌNH 32.16 Một cấu trúc tiêu biểu của RBFN

Chú ý rằng cách hoạt động của "nơron" này rất khác so với nơron sinh học. Trong "nơron" này, sự kích thích không phải là một hàm tổng có trọng số của các tín hiệu đầu vào. Thay vào đó khoảng cách giữa đầu vào và mẫu lưu trữ được tính toán. Nếu khoảng cách này bằng không, nơron phản ứng với giá trị đầu ra lớn nhất bằng một. Nơron này có khả năng nhận dạng các mẫu và phát ra các tín hiệu đầu ra là hàm của sự tương tự. Đặc điểm của nơron này là có tác động mạnh hơn nơron được sử dụng trong mạng lan truyền ngược. Như một hệ quả, mạng tạo ra từ những nơron này cũng có tác động mạnh hơn.

Nếu các tín hiệu đầu vào giống mẫu tích trữ trong một nơron, thì nơron này đưa ra kết quả là 1 và các nơron còn lại có giá trị 0 ở đầu ra, như minh họa trên hình 32.16. Vì vậy tín hiệu đầu ra đúng bằng trọng số của nơron tích cực. Theo cách này, nếu số lượng nơron trong lớp ẩn là lớn thì ánh xạ đầu vào - đầu ra có thể đạt được. Thật không may, vẫn có thể xảy ra trường hợp với một số mẫu có nhiều nơron trong lớp thứ nhất sẽ đưa ra một tín hiệu khác không. Để có sự xấp xỉ thích hợp thì tổng của tất cả các tín hiệu từ lớp ẩn sẽ bằng 1. Để đạt được yêu cầu này thì tín hiệu đầu ra thường được chuẩn hóa, như minh họa trên hình 32.16.

Mạng RBFN có thể được thiết kế và huấn luyện. Việc huấn luyện thường được tiến hành theo hai bước. Trong bước thứ nhất, lớp ẩn được huấn luyện ở chế độ không quan sát bằng cách lựa chọn mẫu tốt nhất cho việc thể hiện nhóm. Phương pháp tương tự như trong kiến trúc WTA có thể được sử dụng. Cũng trong bước này, bán kính σ_i phải được tìm sao cho có sự trùng khớp phù hợp giữa các nhóm.

Bước huấn luyện thứ hai là thuật toán lan truyền ngược sai lệch được tiến hành chỉ cho lớp đầu ra. Vì đây là thuật toán quan sát cho một lớp nên việc huấn luyện rất nhanh, nhanh gấp 100-1000 lần so với mạng đa lớp lan truyền ngược. Điều này khiến mạng RBFN rất hấp dẫn. Mạng này có thể thiết lập dễ dàng bằng máy tính, tuy nhiên thực hiện trên phần cứng là rất khó.

32.6 Mạng nơron hồi qui

Trái ngược với mạng nơron truyền thẳng, đầu ra của mạng nơron hồi qui có thể được kết nối với đầu vào của chúng. Do vậy, tín hiệu trong mạng có thể được quay vòng liên tục. Cho đến bây giờ, chỉ có một số hữu hạn các mạng nơron hồi qui được giới thiệu.

Mạng Hopfield

Mạng hồi qui đơn lớp được phân tích bởi Hopfield (1982). Mạng này được thể hiện trên hình 32.17, có những nơron lưỡng cực ngưỡng cứng với đầu ra bằng 0 hoặc 1. Trọng số được

cho bởi một ma trận vuông đối xứng **W** với các thành phần trên đường chéo chính bằng không ($w_{ij} = 0$ với $i = j$). Sự ổn định của hệ thống thường được phân tích bởi hàm năng lượng:

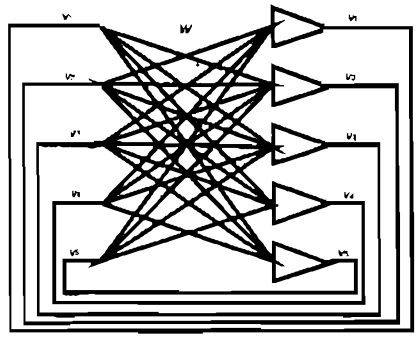
$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} v_i v_j \tag{32.41}$$

Có thể chứng minh rằng trong suốt quá trình quay vòng tín hiệu năng lượng E của mạng giảm và hệ thống hội tụ tới điểm ổn định. Điều này đặc biệt đúng khi giá trị đầu ra của hệ thống được cập nhật trong chế độ không đồng bộ. Điều này có nghĩa rằng tại một chu kỳ, chỉ có một đầu ra ngẫu nhiên có thể được thay đổi tới giá trị yêu cầu. Hopfield cũng chứng minh rằng những điểm ổn định mà hệ thống hội tụ về có thể được lập trình bằng cách điều chỉnh trọng số bằng cách sử dụng luật Hebbian sửa đổi.

$$\Delta w_{ij} = \Delta w_{ji} = (2v_i - 1)(2v_j - 1)c \tag{32.42}$$

Bộ nhớ này có khả năng chứa hữu hạn. Dựa trên thực nghiệm, Hopfield đã ước tính số lượng lớn nhất các mẫu được tích trữ là $0.15N$, trong đó N là số lượng nơron.

Sau này khái niệm hàm năng lượng được mở rộng bởi Hopfield (1984) thành mạng hồi qui đơn lớp có các nơron với hàm tích cực liên tục. Những kiểu mạng này được sử dụng để giải quyết các bài toán lập trình tuyến tính và tối ưu.



HÌNH 32.17 Mạng Hopfield hoặc bộ nhớ kết hợp tự động

Bộ nhớ kết hợp tự động

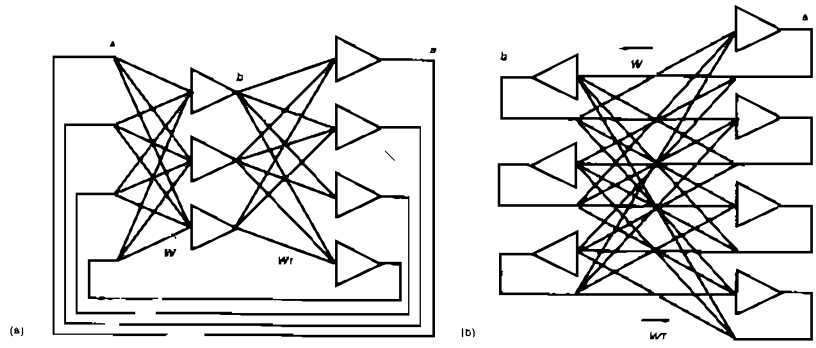
Hopfield (1984) đã mở rộng khái niệm mạng Hopfield thành bộ nhớ kết hợp tự động. Trong cấu trúc mạng tương tự như hình 32.17, những nơron ngưỡng cứng lưỡng cực được sử dụng với đầu ra bằng -1 hoặc +1. Trong mạng này, mẫu $\mathbf{s}_m$ được lưu trữ trong ma trận trọng số **W** bằng cách sử dụng thuật toán tương quan tự động:

$$\mathbf{W} = \sum_{m=1}^{M} \mathbf{s}_m \mathbf{s}_m^T - M\mathbf{I} \tag{32.43}$$

trong đó M là số lượng mẫu lưu trữ và **I** là ma trận đơn vị. Chú ý rằng **W** là ma trận vuông đối xứng với các thành phần trên đường chéo chính bằng không ($w_{ij} = 0$ với $i = j$). Bằng cách sử dụng công thức sửa đổi (32.42), những mẫu mới có thể được thêm vào hoặc trừ đi từ bộ nhớ. Khi bộ nhớ hướng tới một mẫu lưỡng cực nhị phân bằng cách ràng buộc những trạng thái mạng ban đầu thì sau khi quay vòng tín hiệu mạng sẽ hội tụ tới mẫu được lưu trữ gần nhất (giống nhất) hoặc phần bù của nó. Điểm ổn định này sẽ ở giá trị cực tiểu gần nhất của năng lượng:

$$E(\mathbf{v}) = -\frac{1}{2}\mathbf{v}^T\mathbf{W}\mathbf{v} \tag{32.44}$$

Cũng như mạng Hopfield, bộ nhớ kết hợp tự động cũng có khả năng lưu trữ hữu hạn, với giá trị lớn nhất ước tính là M_{max}=0.15N. Khi số lượng các mẫu lưu trữ lớn và gần với dung lượng của bộ nhớ, mạng có xu hướng hội tụ tới trạng thái sai (giả mạo) không được lưu trữ. Trạng thái sai này là những cực tiểu bổ sung thêm của hàm trọng lượng.



HÌNH 32.18 Một ví dụ của bộ nhớ kết hợp tự động hai chiều: (a) được mô tả như mạng hai lớp với tín hiệu quay vòng. (b) được mô tả như mạng hai lớp với luồng tín hiệu hai chiều

Bộ nhớ kết hợp hai chiều (BAM)

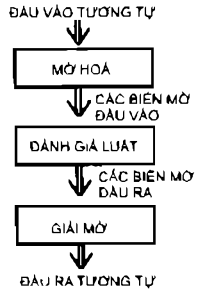
Khái niệm bộ nhớ kết hợp tự động được mở rộng thành bộ nhớ kết hợp hai chiều (BAM) bởi Kosko (1987, 1988). Bộ nhớ này, như trên hình 32.18, có khả năng kết hợp một cặp mẫu **a** và **b**. Đây là mạng hai lớp với đầu ra của lớp thứ hai được kết nối trực tiếp tới đầu vào của lớp thứ nhất. Ma trận trọng số của lớp thứ hai là $\mathbf{W}^T$ và **W** cho lớp thứ nhất. Ma trận trọng số vuông **W** là tổng các ma trận tương quan chéo:

$$\mathbf{W} = \sum_{m=1}^{M} \mathbf{a}_m \mathbf{b}_m \tag{32.45}$$

trong đó M là số lượng cặp được lưu trữ, $\mathbf{a}_m$ và $\mathbf{b}_m$ là các cặp véc tơ được lưu trữ. Nếu các nút **a** hoặc **b** được bắt đầu với một véc tơ tương tự như véc tơ được lưu trữ thì sau khi quay vòng tín hiệu cả $\mathbf{a}_m$ và $\mathbf{b}_m$ đều có thể tìm lại được. BAM có dung lượng nhớ hữu hạn và thường gặp vấn đề ngắt bộ nhớ giống như bộ nhớ kết hợp tự động. Khái niệm BAM có thể được mở rộng cho sự kết hợp ba hoặc nhiều hơn số véc tơ.

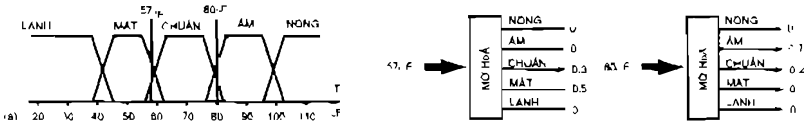
32.7 Hệ mờ

Những ứng dụng quan trọng của mạng nơron liên quan tới sự ánh xạ phi tuyến của biến đầu vào n chiều tới biến đầu ra m chiều. Chức năng đó thường được yêu cầu trong hệ điều khiển, ở đó biến điều khiển phải được tạo ra từ những biến đo lường.



HÌNH 32.19 Sơ đồ khối của bộ điều khiển mờ

Một cách tiếp cận khác cho phép ánh xạ phi tuyến của một tập các biến thành một tập các biến khác là *bộ điều khiển mờ*. Nguyên tắc hoạt động của bộ điều khiển mờ khác hẳn với mạng nơron. Sơ đồ khối của bộ điều khiển mờ được thể hiện trên hình 32.19. Ở bước thứ nhất, đầu vào tương tự được biến đổi thành tập các biến mờ. Trong bước này, đối với mỗi đầu vào tương tự, thường có 3 đến 9 biến mờ được tạo ra. Mỗi biến mờ có một giá trị tương tự giữa 0 và 1. Ở bước tiếp theo, logic mờ được áp dụng cho các biến mờ đầu vào và tạo ra một tập kết quả các biến đầu ra. Ở bước cuối cùng, được gọi là *giải mờ*, từ một tập các biến mờ đầu ra, một hoặc nhiều các biến tương tự được tạo ra và được sử dụng như là những biến điều khiển.



HÌNH 32.20 Quá trình mờ hoá: (a) hàm thuộc cho quá trình mờ hoá và giải mờ, (b) ví dụ về sự biến đổi nhiệt độ thành các biến mờ

Mờ hoá

Mục đích của mờ hoá là biến đổi một biến tương tự ở đầu vào thành một tập các biến mờ. Để có độ chính xác cao hơn thì nhiều biến mờ sẽ được lựa chọn. Để minh họa quá trình mờ hóa, xét một biến đầu vào là nhiệt độ và được mã hóa thành 5 biến mờ: cold (lạnh), cool (mát), normal (bình thường), warm (ấm), và hot (nóng). Mỗi biến mờ có giá trị giữa 0 và 1, mô tả *mức độ kết hợp* của đầu vào tương tự (nhiệt độ) trong biến mờ đã cho. Đôi khi, thay vì thuật ngữ *mức độ kết hợp*, người ta sử dụng thuật ngữ *mức độ liên thuộc*. Quá trình mờ hóa được minh họa trên hình 32.20. Sử dụng hình 32.20 chúng ta có thể tìm ra mức độ kết hợp của mỗi biến mờ với nhiệt độ đã cho. Ví dụ, đối với nhiệt độ là 57^0F, tập các biến mờ sau đây được tạo ra: [0, 0.5, 0.2, 0, 0], và đối với T = 80^0F là [0, 0, 0.25, 0.7, 0]. Thường chỉ có một hoặc hai biến mờ có giá trị lớn hơn không. Trong ví dụ này, hàm hình thang được sử dụng để tính toán mức độ kết hợp. Các hàm khác như tam giác hoặc Gaussian cũng có thể được sử dụng miễn là giá trị tính toán được nằm trong dải từ 0 đến 1. Mỗi hàm thuộc được mô tả bởi ba hoặc bốn tham số. Các tham số này phải được lưu trong bộ nhớ.

Để có sự thiết kế hợp lý bước mờ hoá thì các luật thực hành sau nên được sử dụng:

- Mỗi điểm của biến tương tự đầu vào nên thuộc về ít nhất một và không nhiều hơn hai hàm thuộc.
- Đối với những hàm chồng, tổng của hai hàm thuộc phải không lớn hơn một. Điều này có nghĩa là phần chồng chéo không được giao với những điểm có giá trị cực đại.
- Để có độ chính xác cao hơn thì nên sử dụng nhiều hàm thuộc. Tuy nhiên, mật độ hàm quá lớn sẽ dẫn đến tương tác hệ thống quá nhiều và hệ thống sẽ không ổn định.

Ước lượng luật (Đánh giá luật)

Trái với logic nhị phân, ở đó các biến chỉ có trạng thái nhị phân, trong logic mờ tất cả các biến có giá trị bất kỳ giữa 0 và 1. Logic mờ có các toán tử cơ bản giống nhau: ^ - AND, v – OR, và NOT:

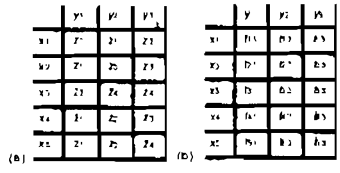
$A \wedge B \wedge C$ $\Rightarrow$ $\min\{A, B, C\}$: giá trị nhỏ nhất của A hoặc B hoặc C

$A \vee B \vee C$ $\Rightarrow$ $\max\{A, B, C\}$: giá trị lớn nhất của A hoặc B hoặc C

$\overline{A}$ $\Rightarrow$ $1 - A$: một trừ giá trị của A

Ví dụ, $0.1 \wedge 0.7 \wedge 0.3 = 0.1$, $0.1 \vee 0.7 \vee 0.3 = 0.7$, và $\overline{0.3} = 0.7$. Những luật này cũng được gọi là các toán tử *Zadeh* (Zadeh, 1965). Chú ý rằng những luật này cũng đúng đối với logic nhị phân cổ điển.

Các qui tắc mờ được xác định trong *bảng mờ* đối với hệ thống đã cho. Xét một hệ thống đơn giản với hai biến đầu vào tương tự x và y, và một biến đầu ra z. Mục đích là thiết kế hệ thống mờ tạo ra z là hàm $f(x, y)$. Sau khi mờ hoá, biến tương tự x được mô tả bởi 5 biến mờ: x_1, x_2, x_3, x_4, x_5 và biến tương tự y được mô tả bởi 3 biến mờ: y_1, y_2, y_3. Giả sử rằng biến đầu ra tượng tự được mô tả bởi 4 biến mờ: z_1, z_2, z_3, z_4 Vấn đề trọng tâm của quá trình thiết kế là thiết lập các biến mờ đầu ra thích hợp (z_k) cho mọi khả năng kết hợp của các biến mờ đầu vào như trên bảng trong hình 32.21. Người thiết kế phải chỉ ra nhiều qui tắc như: nếu đầu vào được mô tả bởi biến mờ x_i và y_j thì đầu ra được mô tả bởi biến mờ z_k,. Khi bảng mờ được xác định thì việc tính toán logic mờ được tiến hành trong hai bước. Đầu tiên mỗi ô của bảng mờ được điền với biến mờ trung gian t_{ij} thu được từ toán tử AND $t_{ij} = \min\{x_i, y_j\}$, như trên hình 32.21 (b). Bước này độc lập với các qui tắc cần thiết đối với hệ thống đã cho. Ở bước thứ hai, toán tử OR (max) được sử dụng để tính toán mỗi biến mờ đầu ra z_k. Trong ví dụ đã cho trên hình 32.12, $z_1 = \max\{t_{11}, t_{12}, t_{21}, t_{41}, t_{51}\}$, $z_2 = \max\{t_{13}, t_{31}, t_{42}, t_{52}\}$, $z_3 = \max\{t_{22}, t_{23}, t_{43}\}$, $z_4 = \max\{t_{32}, t_{34}, t_{53}\}$. Chú ý rằng các công thức phụ thuộc vào những thông số được cho trong bảng mờ trên hình 32.21.



HÌNH 32.21 Bảng mờ: (a) bảng với luật mờ. (b) bảng với các biến trung gian t_{ij}



HÌNH 32.22 Minh hoạ cho quá trình giải mờ

Giải mờ

Như một kết quả của ước lượng luật mờ, mỗi biến đầu ra tương tự được thể hiện bởi nhiều biến mờ. Mục đích của giải mờ là có được đầu ra tương tự. Điều này có thể được thực hiện bằng cách sử dụng hàm thuộc như trên hình 32.20. Trong bước thứ nhất, các biến mờ thu được từ ước lượng luật được sử dụng để sửa đổi hàm thuộc theo công thức:

$$\mu_k^*(z) = \min\{\mu_k(z), z_k\} \tag{32.46}$$

Ví dụ, nếu các biến mờ đầu ra là 0, 0.2, 0.7, 0.0, thì hàm thuộc được sửa đổi có hình dạng như đường đậm như trên hình 32.22. Giá trị tương tự của biến z được xác định như là *trọng tâm* của hàm thuộc được sửa đổi $\mu_k^*(z)$,

$$z_{analog} = \frac{\sum_{k=1}^{n} \int_{-\infty}^{+\infty} \mu_k^*(z) z dz}{\sum_{k=1}^{n} \int_{-\infty}^{+\infty} \mu_k^*(z) dz} \tag{32.47}$$

Trong trường hợp hình dạng của hàm thuộc đầu ra $\mu_k(z)$ giống nhau, phương trình trên được đơn giản hóa thành:

$$z_{\text{analog}} = \frac{\sum_{k=1}^{n} z_k zc_k}{\sum_{k=1}^{n} z_k} \tag{32.48}$$

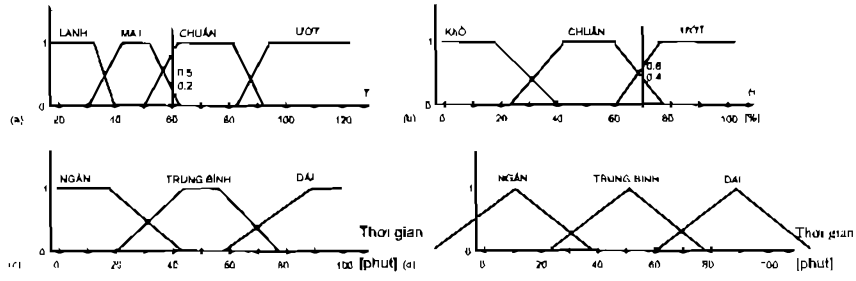
trong đó:

n = số hàm thuộc của biến đầu ra z_{analog}

z_k = biến đầu ra mờ thu được từ ước lượng luật

zc_k = giá trị tương tự tương ứng với tâm của hàm thuộc thứ k

Phương trình 32.47 thường quá phức tạp để sử dụng trong hệ thống dựa trên vi điều khiển, do vậy trong thực tế phương trình 32.48 được sử dụng nhiều hơn.



HÌNH 32.23 Hàm thuộc cho ví dụ được mô tả: (a) và (b) hàm thuộc cho biến đầu vào, (c) và (d) là hai hàm thuộc cho biến đầu ra

Ví dụ thiết kế

Xét việc thiết kế một bộ điều khiển mờ đơn giản cho hệ thống bình tưới. Thời gian tưới là một hàm của nhiệt độ và độ ẩm. Bốn hàm thuộc được sử dụng cho nhiệt độ, ba cho độ ẩm và ba cho thời gian tưới như minh họa trên hình 32.23. Bằng kinh nghiệm, bảng mờ có thể được trình bày như trên hình 32.24.

Giả sử nhiệt độ là 60^0F và độ ẩm là 70%. Bằng cách sử dụng hàm thuộc cho nhiệt độ và độ ẩm, ta thu được các biến mờ sau đây cho nhiệt độ: [0, 0.2, 0.5, 0] và cho độ ẩm: [0, 0.4, 0.6]. Sử dụng toán tử min, bảng mờ có thể được điền với các biến mờ tạm thời, như trên hình 32.24 (b). Chú ý rằng chỉ có bốn ô có giá trị khác không. Sử dụng luật mờ, như trên hình 32.24 (a), toán tử max có thể được áp dụng để thu được các biến đầu ra mờ: short → o_1 = max{0, 0, 0.2, 0.5, 0} = 0.5, medium → o_2 = max{0, 0, 0.2,0.4, 0} = 0.4, long → o_3 = max{0,0} = 0.

(a)

	KHÔ	CHUẨN	ẨM
LẠNH	M	S	S
MÁT	M	M	S
ẤM	L	M	S
NÓNG	L	M	S

(b)

	KHÔ	CHUẨN	ẨM
	0	0.4	0.6
LẠNH 0	0	0	0
MÁT 0.2	0	0.2	0.2
ẤM 0.5	0	0.4	0.5
NÓNG 0	0	0	0

HÌNH 32.24 Bảng mờ: (a) các luật mờ cho ví dụ thiết kế, (b) các biến mờ tạm thời cho ví dụ thiết kế

Sử dụng phương trình 32.47 và hình 32.23 (c), xác định được thời gian tưới là 28 giây. Phương pháp đơn giản hóa được sử dụng với phương trình 32.46 và hình 32.23 (d), thì thời gian tưới là 27 giây.

32.8 Thuật toán di truyền

Sự thành công của mạng nơron nhân tạo khuyến khích các nhà nghiên cứu tìm hiểu các mẫu khác trong tự nhiên. Khả năng của di truyền thông qua sự tiến hóa có thể tạo ra những bộ máy phức tạp như con người. Thuật toán di truyền dựa theo quá trình tiến hóa như trong tự nhiên để tìm ra giải pháp tốt hơn cho các vấn đề phức tạp. Cơ sở cho thuật toán di truyền được đưa ra bởi Holland (1975) và Goldberg (1989). Sau khi khởi tạo, các bước *chọn lọc*, *lai chéo* và *đột biến* được lặp lại cho mỗi thế hệ. Trong quá trình này, một chuỗi các ký hiệu được hiểu như là những nhiễm sắc thể, được ước lượng để có một giải pháp tốt hơn. Thuật toán di truyền bắt đầu với việc mã hóa và khởi tạo. Tất cả các bước quan trọng của thuật toán di truyền sẽ được mô tả bằng một ví dụ đơn giản để tìm ra cực đại của hàm $(\sin^2(x)\text{-}0.5*x)^2$ với dải của x từ 0 đến 1.6. Chú ý rằng trong dải này, hàm có giá trị cực đại toàn cục tại $x = 1.309$, và cực đại cục bộ tại $x = 0.262$.

Mã hóa và khởi tạo

Đầu tiên, biến x phải được mô tả như một chuỗi các ký hiệu. Với chuỗi dài hơn, quá trình thường hội tụ nhanh hơn, do vậy càng ít ký hiệu cho một chuỗi được sử dụng thì kết quả càng tốt hơn. Mặc dù chuỗi này có thể là một dãy các ký hiệu bất kỳ nhưng ký hiệu nhị phân 0 và 1 thường được sử dụng. Trong ví dụ này, các số nhị phân 6 bit được sử dụng cho việc mã hóa, có giá trị trong hệ thập phân là $40x$. Quá trình xử lý bắt đầu với một thế hệ ngẫu nhiên của mẫu khởi tạo như trong bảng 32.1.

Chọn lọc và sao chép

Chọn lọc những thành viên tốt nhất của mẫu là một bước quan trọng trong thuật toán di truyền. Nhiều phương pháp khác nhau có thể được sử dụng để đánh giá các cá thể. Trong ví dụ này, hàm đánh giá đã biết. Hạng cao nhất có số thành viên là 6, và hạng thấp nhất có số thành viên là 3. Những thành viên với hạng cao hơn sẽ có cơ hội hơn để lai tạo. Xác suất lai tạo cho mỗi thành viên được tính bằng một tỷ lệ của tổng tất cả các giá trị hàm đối tượng. Tỷ lệ này được thể hiện ở cột ngoài cùng của bảng 32.1.

BẢNG 32.1 Mẫu ban đầu

Số chuỗi	Chuỗi	Giá trị hệ thập phân	Giá trị biến	Giá trị hàm	Tỷ lệ
1	101101	45	1.125	0.0633	0.2465
2	101000	40	1.000	0.0433	0.1686
3	010100	20	0.500	0.0004	0.0016
4	100101	37	0.925	0.0307	0.1197
5	001010	10	0.250	0.0041	0.0158
6	110001	49	1.225	0.0743	0.2895
7	100111	39	0.975	0.0390	0.1521
8	000100	4	0.100	0.0016	0.0062
Tổng				0.2568	1.0000

Chú ý rằng để sử dụng phương pháp này, hàm đối tượng của chúng ta phải luôn là dương. Nếu không, cần đưa vào trước một sự chuẩn hóa thích hợp.

Sao chép

Những số trong cột ngoài cùng của bảng 32.1 thể hiện xác suất sao chép. Do vậy, hầu như số thành viên 3 và 8 sẽ không được lai tạo, số 1 và 6 có thể có hai hoặc nhiều hơn hai bản sao. Sử dụng quá trình lai tạo ngẫu nhiên, các mẫu sau đây (được tổ chức thành cặp) có thể được sinh ra:

101101 → 45 110001 → 49 100101 → 37 110001 → 49
100111 → 39 101101 → 45 110001 → 49 101000 → 40

Nếu kích cỡ của mẫu từ một thế hệ này tới một thế hệ khác là giống nhau, hai cha mẹ sẽ sinh ra hai con. Bằng cách kết hợp hai chuỗi, hai chuỗi khác sẽ được sinh ra. Cách đơn giản để thực hiện điều này là tách làm hai phần mỗi chuỗi cha mẹ và trao đổi những chuỗi con giữa cha mẹ. Ví dụ, từ chuỗi cha mẹ 010100 và 100111, những chuỗi con sau đây sẽ được sinh ra: 010111 và 100100. Quá trình này được hiểu như là *lai chéo*. Những chuỗi con sinh ra là:

101111 → 47 110101 → 49 100001 → 33 110000 → 48
100101 → 37 101001 → 41 110101 → 53 101001 → 41

Nói chung, các chuỗi không cần phải tách làm đôi. Thường thì chỉ một số bit được lựa chọn để trao đổi giữa cha mẹ. Điều quan trọng là vị trí các bit không bị thay đổi.

Đột biến

Trong quá trình tiến hóa, lai tạo được nâng cao nhờ đột biến. Cùng với tính chất thừa kế từ cha mẹ thế hệ sau có thêm một số tính chất ngẫu nhiên mới. Quá trình này được hiểu như là sự đột biến. Trong hầu hết trường hợp, sự đột biến sinh ra thế hệ sau có thứ hạng thấp hơn, và bị loại bỏ trong quá trình lai tạo. Tuy nhiên, thỉnh thoảng đột biến có thể tạo ra các cá thể tốt hơn với một tính chất mới. Điều này bảo vệ quá trình lai tạo khỏi sự thoái hóa. Trong thuật toán di truyền, đột biến thường đóng vai trò thứ yếu. Đối với mức độ biến tương đối cao, quá trình này giống như tạo mẫu ngẫu nhiên và thuật toán tìm kiếm như vậy rất không hiệu quả. Tỷ lệ biến đổi thường được giả sử tại mức độ nhỏ hơn 1%. Trong ví dụ này, đột biến tương đương với sự thay đổi bit ngẫu nhiên của mẫu đã cho. Trong trường hợp đơn giản, với những chuỗi ngắn và mẫu nhỏ, và với tỷ lệ biến đổi điển hình là 0.1%, các mẫu giữ hầu như không bị thay đổi bởi quá trình đột biến. Thế hệ thứ hai cho ví dụ này được thể hiện trong bảng 32.2.

BẢNG 32.2 Mẫu của thế hệ thứ hai

Số chuỗi	Chuỗi	Giá trị hệ thập phân	Giá trị biến	Giá trị hàm	Tỷ lệ
1	010111	47	1.175	0.0696	0.1587
2	100100	37	0.925	0.0307	0.0701
3	110101	53	1.325	0.0774	0.1766
4	010001	41	1.025	0.0475	0.1084
5	100001	33	0.825	0.0161	0.0368
6	110101	53	1.325	0.0774	0.1766
7	110000	48	1.200	0.0722	0.1646
8	101001	41	1.025	0.0475	0.1084
Tổng				0.4387	1.0000

Chú ý rằng hai thành viên có hạng cao nhất giống nhau của thế hệ thứ hai rất gần với nghiệm $x = 1.309$. Các cặp cha mẹ được lựa chọn ngẫu nhiên cho thế hệ thứ ba là:

$010111 \rightarrow 47$ $110101 \rightarrow 53$ $110000 \rightarrow 48$ $101001 \rightarrow 41$

$110101 \rightarrow 53$ $110000 \rightarrow 48$ $101001 \rightarrow 41$ $110101 \rightarrow 53$

Chúng tạo ra những cá thể (con) như sau:

$010101 \rightarrow 21$ $110000 \rightarrow 48$ $110001 \rightarrow 49$ $101101 \rightarrow 45$

$110111 \rightarrow 55$ $110101 \rightarrow 53$ $101000 \rightarrow 40$ $110001 \rightarrow 49$

Kết quả tốt nhất trong mẫu thứ ba giống như trong mẫu thứ hai. Bằng cách kiểm tra cẩn thận tất cả các chuỗi từ thế hệ thứ hai và thứ ba, có thể kết luận rằng sử dụng lai chéo, ở đó các chuỗi luôn được tách làm đôi, kết quả tốt nhất $110100 \rightarrow 52$ sẽ không bao giờ đạt được mặc dù có bao nhiêu thế hệ được tạo ra đi nữa. Lý do là không có mẫu nào trong thế hệ thứ hai có chuỗi con với phần đuôi là 100. Đối với sự lai chéo này, kết quả tốt nhất chỉ có thể đạt được do quá trình đột biến và phải trải qua nhiều thế hệ. Kết quả tốt nhất trong thế hệ tương lai cũng có thể đạt được nếu chuỗi được tách ra tại các vị trí ngẫu nhiên. Một giải pháp khả thi khác là chỉ những bit lựa chọn ngẫu nhiên được trao đổi giữa cha mẹ.

Thuật toán di truyền học rất nhanh, nó đưa ra một kết quả tốt sau một vài thế hệ. Kết qua này thường gần với cực đại toàn cục, nhưng không phải là tốt nhất.

Định nghĩa các thuật ngữ

Lan truyền ngược (Backpropagation): Kỹ thuật huấn luyện cho mạng nơron đa lớp.

Nơron lưỡng cực (Bipolar neuron): Nơron với đầu ra giữa -1 và +1.

Mạng truyền thẳng (Feedforward network): Mạng không phản hồi

Nhận thức (Perceptron): Mạng với các nơron ngưỡng cứng

Mạng hồi qui (Recurrent network): Mạng có phản hồi

Học có giám sát (Supervised learning): Thủ tục học khi đầu ra mong muốn đã biết

Nơron đơn cực (Unipolar neuron): Nơron với tín hiệu đầu ra giữa 0 và +1.

Học không giám sát (Unsupervised learning): Thủ tục học khi đầu ra chưa biết

Tài liệu tham khảo

[1] Fahlman, S.E. 1988. Faster-learning variations on backpropagation: An empirical study. *Proceedings of the Connectionist Models Summer School,* D. Touretzky, G. Hinton, and T. Sejnowski, Eds., Morgan Kaufmann, San Mateo, CA.

[2] Fahlman, S.E. and Lebiere, C. 1990. The cascade correlation learning architecture. *Adv. Ner. Inf. Proc. Syst.,* 2, D.S. Touretzky, ed., pp. 524–532. Morgan Kaufmann, Los Altos, CA.

[3] Goldberg, D.E. 1989. Genetic Algorithm in Search, Optimization and Machine Learning. Addison–Wesley, Reading, MA.

[4] Grossberg, S. 1969. Embedding fields: a theory of learning with physiological implications. *Journal of Mathematical Psychology* 6:209–239.

[5] Hebb, D.O. 1949. *The Organization of Behavior, a Neuropsychological Theory*. John Wiley, New York.

[6] Hecht-Nielsen, R. 1987. Counterpropagation networks. *Appl. Opt.* 26(23):4979–4984.

[7] Hecht-Nielsen, R. 1988. Applications of counterpropagation networks. *Neural Networks* 1:131–139.

[8] Holland, J.H. 1975. *Adaptation in Natural and Artificial Systems*. Univ. of Michigan Press, Ann Arbor, MI.

[9] Hopfield, J.J. 1982. Neural networks and physical systems with emergent collective computation abilities. *Proceedings of the National Academy of Science* 79:2554–2558.

[10] Hopfield, J.J. 1984. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Science* 81:3088–3092.

[11] Kohonen, T. 1988. The neural phonetic typerater. *IEEE Computer* 27(3):11–22.

[12] Kohonen, T. 1990. The self-organized map. *Proc. IEEE* 78(9):1464–1480.

[13] Kosko, B. 1987. Adaptive bidirectional associative memories. *App. Opt.* 26:4947–4959.

[14] Kosko, B. 1988. Bidirectional associative memories. *IEEE Trans. Sys. Man, Cyb.* 18:49–60.

[15] McCulloch, W.S. and Pitts., W.H., 1943. A logical calculus of the ideas imminent in nervous activity.*Bull. Math. Biophys.* 5:115–133.

[16] Minsky, M. and Papert, S. 1969. *Perceptrons*. MIT Press, Cambridge, MA.

[17] Nilsson, N.J. 1965. *Learning Machines: Foundations of Trainable Pattern Classifiers*. McGraw-Hill, New York.

[18] Nguyen, D. and Widrow, B. 1990. Improving the learning speed of 2-layer neural networks, by choosing initial values of the adaptive weights. *Proceedings of the International Joint Conference on Neural Networks* (San Diego), CA, June.

[19] Pao, Y.H. 1989. *Adaptive Pattern Recognition and Neural Networks*. Addison–Wesley, Reading, MA.

[20] Rosenblatt, F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psych. Rev.* 65:386–408.

[21] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. 1986. Learning internal representation by error propagation. *Parallel Distributed Processing*. Vol. 1, pp. 318–362. MIT Press, Cambridge, MA.

[22] Sejnowski, T.J. and Rosenberg, C.R. 1987. Parallel networks that learn to pronounce English text. *Complex Systems* 1:145–168.

[23] Specht, D.F. 1990. Probalistic neural networks. *Neural Networks* 3:109–118.

[24] Specht, D.F. 1992. General regression neural network. *IEEE Trans. Neural Networks* 2:568–576.

[25] Wasserman, P.D. 1989. *Neural Computing Theory and Practice*. Van Nostrand Reinhold, New York.

[26] Werbos, P., 1974. Beyond regression: new tools for prediction and analysis in behavioral sciences. Ph.D. diss., Harvard University.

[27] Widrow, B. and Hoff, M.E. 1960. Adaptive switching circuits. 1960 IRE Western Electric Show and Convention Record, Part 4 (Aug. 23):96–104.

[28] Widrow, B. 1962. Generalization and information storage in networks of adaline Neurons. In *Selforganizing Systems*, M.C. Jovitz, G.T. Jacobi, and G. Goldstein, eds., pp. 435–461. Sparten Books, Washington, D.C.

[29] Wilamowski, M. and Torvik, L., 1993. Modification of gradien computation in the back-propagation algorithm. *ANNIE'93 - Artificial Neural Networks in Engineering*. November 14–17, 1993, St. Louis, Missou.; also in C.H Dagli, ed. 1993. *Intelligent Engineering Systems Through Artificial Neural Networks* Vol. 3, pp. 175–180. ASME Press, New York.

[30] Zadeh, L.A. 1965. Fuzzy sets. *Information and Control* 8:338–353.

[31] Zurada, J. 1992. *Introduction to Artificial Neural Systems*. West Publ.

33

Điều khiển hiện đại cho một trục thủy lực-điện

33.1 Giới thiệu chung 33-1
33.2 Tổng quan về ROBI_3, rôbốt dạng tọa độ đề - các với ba trục thủy lực-điện 33-2
33.3 Mô hình toán học và mô phỏng các trục thủy lực-điện 33-4
33.4 Điều khiển trục thủy lực-điện bằng các bộ điều khiển truyền thống 33-8
33.5 Điều khiển trục thủy lực-điện với bộ điều khiển mờ 33-12
33.6 Điều khiển trục thủy lực-điện dùng kỹ thuật nơron 33-13
33.7 Điều khiển trục thủy lực-điện dùng kỹ thuật mờ-nơron 33-15
33.8 Phần mềm 33-21
33.9 Kết luận 33-22

Florin Ionescu
University of Applied Sciences

Crina Vlad
Politeknica University of Bucharest

Dragos Arotaritei
Aalborg University Esbjerg

33.1 Giới thiệu chung

Với sự phát triển của công nghệ trong những năm gần đây, rôbốt được xem như là một hệ cơ điện tử hiện đại đòi hỏi sự hiểu biết về cơ học, cơ cấu chấp hành và điều khiển để thực hiện các nhiệm vụ rất phức tạp. Các hệ thống servo khác nhau, đặc biệt là thủy lực-điện đã được sử dụng trong các cấp chấp hành của rôbốt. Để hiểu được các phương pháp điều khiển hiện đại, chương này sẽ đề cập đến việc thực hiện các bộ điều khiển hiện đại bên cạnh bộ điều khiển truyền thống đã được sử dụng trong hệ thống thủy lực-điện. Hệ thủy lực-điện được xét ở đây là một trong những trục của rôbốt. Những rôbốt này có ba trục thủy lực-điện giống như trục được nghiên cứu trong chương này hoặc nhiều hơn.

Trục thủy lực-điện có mô hình toán học được giới thiệu trong chương này là hệ phi tuyến. Những bộ điều khiển truyền thống ngày càng không phù hợp để điều khiển các hệ thống với mô hình không tường minh và có tính phi tuyến rõ rệt. Do vậy thường được sử dụng các kỹ thuật điều khiển hiện đại như mạng nơron và điều khiển mờ trong những hệ thống này. Mạng nơron được đề xuất đầu tiên bởi McCulloch và Pitts, Rosenblantt, Widrow, có nhiều giới hạn làm hạn chế phạm vi ứng dụng của chúng. Một sự thay đổi quan trọng diễn ra trong những năm 1980 khi lý thuyết của Hopfield về mạng nơron hồi qui, mô hình mạng nơron tự tổ chức được phát triển bởi Kohonen, và mạng nơron tế bào đã khởi động lại những nghiên cứu về ứng dụng của mạng nơron. Sự phát triển của những thuật toán hiệu quả, đặc biệt về cấu trúc và ứng dụng của mạng nơron trong điều khiển, đã cho thấy sự quan tâm nghiên cứu của giới khoa học đối với lĩnh vực này.

Hệ mờ cùng với mạng nơron giữ vị trí quan trọng trong kỹ thuật điều khiển nâng cao. Hệ thống này có nguồn gốc từ lý thuyết tập mờ được giới thiệu bởi L.Zadeh. Đặc tính chủ yếu của hệ mờ là sự suy luận xấp xỉ, trong đó các biến được mô tả một cách định tính. Do những khả năng của hệ

mờ đối với những thông tin không rõ ràng nên chúng được sử dụng để mô tả sự hiểu biết theo luật ngôn ngữ. Theo cách này, sự hiểu biết về hoạt động của con người dưới dạng ngôn ngữ hoặc số được sử dụng để tạo ra tập các qui tắc mờ dạng "nếu-thì" sẽ là cơ sở cho bộ điều khiển mờ. Hạn chế chính của hệ thống điều khiển mờ là chưa có một phương pháp chuẩn hoá chặt chẽ nào để thiết kế chúng theo các bước có tính hệ thống cao. Các thủ tục học sử dụng mạng nơron đã được áp dụng thành công để điều chỉnh tham số của các hàm thuộc nhằm khắc phục hạn chế trên.

Việc kết hợp hai lĩnh vực này tạo ra hệ mờ-nơron nổi bật với những ứng dụng đã mang lại kết quả đầy hứa hẹn trong lĩnh vực điều khiển. Bộ điều khiển mờ-nơron đã được sử dụng để cải thiện khả năng hoạt động của hệ thống phi tuyến.

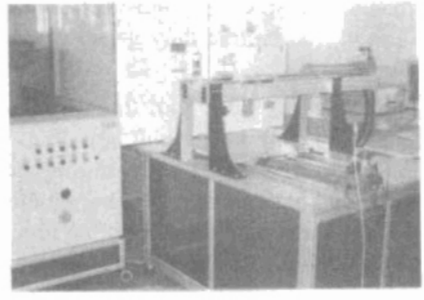
Chương này bao gồm: giới thiệu về hệ thống thủy lực-điện việc điều khiển các hệ thống đó trong mục 33.2; phần 33.3 đưa ra mô hình toán học của trục thủy lực-điện; phần 33.4 xét việc điều khiển các trục thủy lực-điện thông qua các phương pháp truyền thống; hệ mờ (mục 33.5); mạng nơron (mục 33.6), kỹ thuật mờ-nơron (mục 33.7) và phần kết luận được đưa ra ở mục 33.8.

33.2 Tổng quan về ROBI_3, rôbốt dạng tọa độ đề - các với ba trục thủy lực-điện

Hệ thống tự động sử dụng các trục thủy lực-điện có mô hình toán học được mô tả trong mục 33.3 là một rôbốt dạng tọa độ đề-các (Cartessian Robot) tên là ROBI_3. ROBI_3 có ba trục thủy lực-điện giống nhau và được làm từ nhôm [21]. Bộ phận trượt được kích hoạt bởi một cơ cấu chấp hành thủy lực. Bộ phận trượt di chuyển trong một đường dẫn tuyến tính với những quả cầu và hai quĩ đạo tròn khép kín. Hệ thống cung cấp thủy lực được đặt bên dưới bàn rôbốt và có một hệ thống điều khiển và làm mát bằng khí. Cấu trúc cơ khí của rôbốt được mô tả trên hình 33.1 Ba trục của ROBI_3 được điều khiển giống nhau bởi phần mềm điều khiển có tên là TORCH được chạy trên Windows. Phần cứng điều khiển dSPACE 32 bit có 10 giao diện A/D và D/A được cắm vào máy tính PC và là giao diện giữa máy tính PC và các trục.

Trục thủy lực-điện gồm có một xi lanh thủy lực và một van servo và có cấu trúc phi tuyến. Hệ thống điều khiển của một trục gồm có:

1. Bộ điều khiển: máy tính cá nhân và một card xử lý;
2. Bộ biến đổi thủy lực-điện;
3. Cơ cấu chấp hành: động cơ servo thủy lực tuyến tính (Linear Hydraulic Servomotor- LHM);
4. Quá trình cơ học được điều khiển: đặc trưng bởi vị trí của bộ phận trượt;
5. Bộ chuyển đổi vị trí.

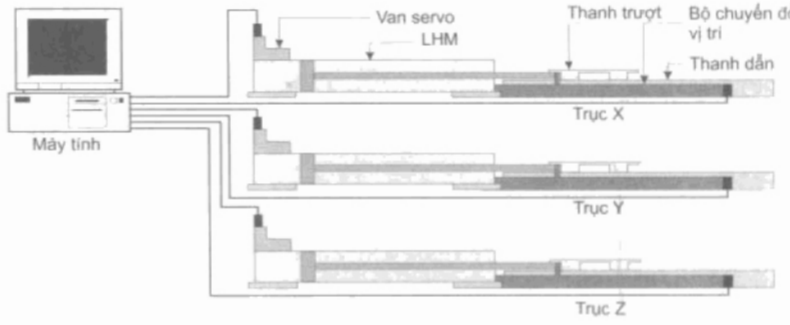


HÌNH 33.1 Rôbốt ROBI_3 với ba trục quay [21]

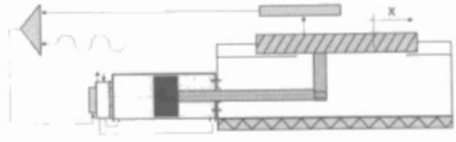
Hệ thống điều khiển của rôbốt ROBI_3 được minh họa trên hình 33.2, trên đó mô tả ba trục thủy lực-điện cùng với cấu trúc của từng trục, có thể thấy các trục là hoàn toàn giống nhau.

Mô hình toán học tương ứng cho mỗi trục, thực hiện dựa trên cơ sở điều khiển rôbốt, được miêu tả trong phần 33.3. Mặc dù mô phỏng 3 trục của rôbốt là mô phỏng số nhưng dữ liệu giao diện về cấu trúc cơ khí cần thiết đã đạt được.

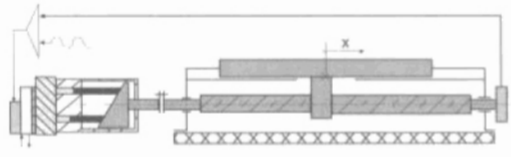
Quá trình thử nghiệm sơ bộ với các biến điều khiển: vị trí và vận tốc được tạo ra để làm thực nghiệm (hình 33.2 và 33.3). Sơ đồ của một hệ điều khiển vị trí vòng kín cùng với phép đo trực tiếp được thể hiện trên hình 33.3 và hệ kín điều khiển vị trí vòng kín với phép đo gián tiếp tại trục quay được thể hiện trên hình 33.4.



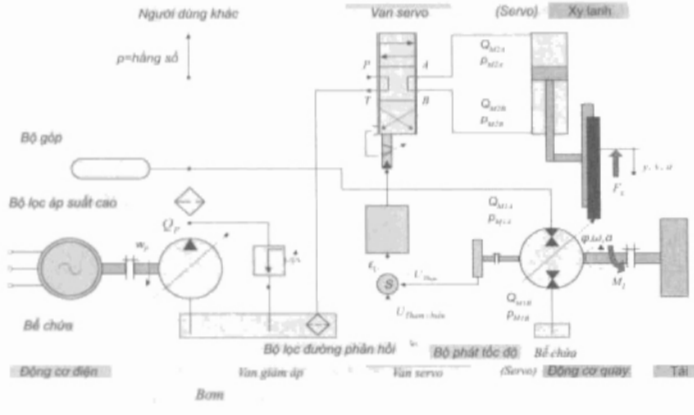
HÌNH 33.2 Hệ điều khiển của rôbốt [21, 24]



HÌNH 33.3 Sơ đồ của một hệ kín điều khiển vị trí với phép đo trực tiếp



HÌNH 33.4 Hệ kín điều khiển vị trí với phép đo trực tiếp



HÌNH 35.5 Điều chỉnh tỷ lệ thể tích Q với áp suất không đổi

Ví dụ điều chỉnh tỷ lệ thể tích Q với áp suất p là hằng số (Q thay đổi, p không đổi) được thể hiện trên hình 33.5. Mặc dù mô hình cổ điển này rất có ích cho ứng dụng trên nhưng chỉ được sử dụng cho các kết quả sơ bộ trong mô phỏng. Một trong những lý do chính phải sử dụng phương pháp này là vì chúng ta cần một mô hình toán học tốt và đã được nghiên cứu nhiều để kiểm tra các bộ điều khiển (ví dụ như nơron hoặc mờ-nơron).

33.3 Mô hình toán học và mô phỏng các trục thủy lực-điện

Phần 33.3 thực hiện việc phân tích để tìm ra mô hình toán học (Mathematical Model - MM) của các trục thủy lực-điện, là các phần cấu thành rôbốt ROBI_3. Phương pháp phân tích này có thuận lợi vì nó cho phép sử dụng mô hình toán học này cho những trục thủy lực-điện khác mà không cần quan tâm đến số trạng thái khác nhau, và cho phép kiểm tra các đặc tính động lực học của trục trong giai đoạn thiết kế.

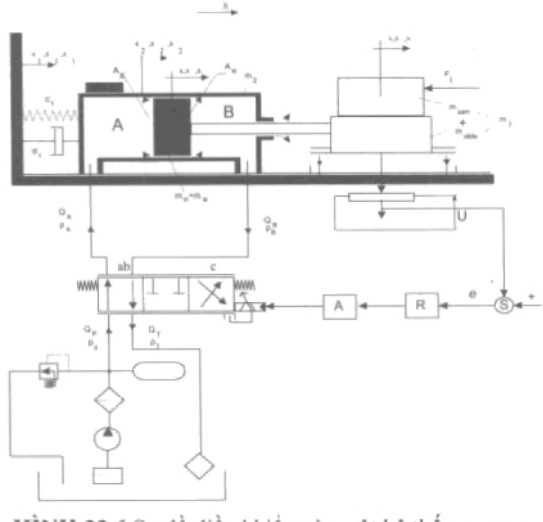
Trong phần này, các vấn đề sau đây được giới thiệu:

1. Mô hình tĩnh của các thành phần hệ thống thủy lực-điện (van servo, động cơ tuyến tính thủy lực);
2. Những tham số trong MM, dựa trên đặc tính chế độ lưu lượng và cấu trúc;
3. Mô hình toán học phi tuyến của hệ thống thủy lực-điện được đề xuất;
4. Xây dựng sơ đồ cấu trúc một trục thủy lực để mô phỏng hoạt động của chúng (SIMULINK);
5. Khảo sát MM.

Giá trị của các tham số mô tả MM là một tập hợp dựa trên đặc tính thủy lực và tham số cấu trúc của hệ thống được xét.

Mô hình toán học mở rộng

Hệ được nghiên cứu gồm có một van servo và một động cơ không đối xứng. Trong hầu hết trường hợp, việc điều khiển các trục thủy lực-điện là điều khiển trực tiếp nhằm mục đích điều khiển vị trí, vận tốc, áp suất hoặc điều khiển lực. Ở đây chúng ta sẽ nghiên cứu điều khiển vị trí. Vòng điều khiển vị trí cho hệ thống đề xuất được minh họa trên hình 33.6.



HÌNH 33.6 Sơ đồ điều khiển của một hệ thống servo

Mô hình toán học phi tuyến của van servo

Như đã đề cập ở phần trước, van servo sử dụng cho trục thủy lực-điện đề xuất có bốn đường dẫn và ba vị trí. Van servo có bốn đỉnh điều khiển tích cực tại trạng thái thủy lực thứ hai. Ba trạng thái có thể phân biệt bởi van servo này là: van cơ - điện, van thủy lực – cơ khí, và van cơ khí- thủy lực. Dưới góc độ là một hệ thống thì van servo rất phức tạp với các dạng phi tuyến khác nhau. Các tính chất phi tuyến động và tĩnh như miền chết, bão hoà, ma sát Coulombian và Newtonian với trễ và tính bất đối xứng xuất hiện trong ba mức này và trong cơ cấu chấp hành của động cơ thủy lực-điện. Những điểm này được xem xét khi mô hình hóa hoạt động của van servo và xi lanh.

Đối với các van servo được nghiên cứu, quá trình lưu chuyển chất lỏng được xem là trực tiếp từ bơm tới khoang nạp A (Q_A) và từ khoang xả B tới buồng chứa (Q_B). Hình 33.6 mô tả vòng điều khiển, trong đó bộ chuyển đổi được đặt trên đường phản hồi và bộ điều khiển R cùng bộ khuếch đại A ở trên đường thuận. Tín hiệu điện ($\pm 10\ V$ hoặc $\pm 300\ mA$) được biến đổi thành dịch chuyển x_v của van và cũng bằng cách này lưu lượng Q được truyền tới động cơ thủy lực tuyến tính.

Từ quan điểm về đặc tính điều khiển, số đỉnh tích cực được xem như là một phương pháp phân loại các van trượt [3]. Một van servo với bốn đỉnh điều khiển tích cực được xét. Một mô hình toán học đơn giản mô tả chức năng của van có các phương trình như sau:

$$u(t) = L \cdot \frac{d(t)}{dt} + R \cdot i(t) \tag{33.1}$$

trong đó L [H] là độ tự cảm, R [W] là điện trở, $u(t)$ [V] là điện áp điều khiển, $i(t)$ [A] là mật độ điều khiển.

$$m \cdot x_v(t) + d \cdot x_v(t) + c \cdot x_v(t) = \sum F \tag{33.2}$$

trong đó m [kg] là khối lượng của van, d [N/(m)/s] là građien tuyến tính hoá của ma sát nhớt đối với pít-tông của van; c [N/m] hệ số đàn hồi của chất lỏng; x_v [m] là dịch chuyển của ống cuộn; $\sum F$ [N] lực sinh ra, được thực hiện trên ống van.

$$\begin{aligned} &\ddot{x}_v(t) + 2 \cdot D_v \omega_{0v} \cdot \dot{x}_v(t) + \omega_{0v}^2 \cdot x_v(t) = \frac{\sum F}{m} \\ &\ddot{x}_v(t) = k^* - 2 \cdot D_v \cdot \omega_{0v} \cdot \dot{x}_v(t) - \omega_{0v}^2 \cdot x_v(t), \quad k^* = \frac{\sum F}{m} \end{aligned} \tag{33.3}$$

Độ dịch chuyển x_v thu được dựa trên những phương trình trên, được thực hiện trên SIMULINK bằng cách sử dụng sơ đồ khối trên hình 33.7. Môđun của mô hình toán học phi tuyến này bao gồm hai tầng của trục thủy lực-điện: tầng thủy lực- điện và tầng thủy lực- cơ.

Các phương trình tương ứng với bốn lưu lượng đi qua van servo là:

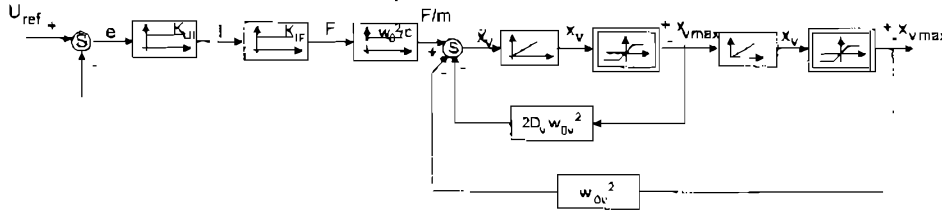
$$\begin{aligned} Q_{PA} &= \alpha_D \cdot \pi \cdot D_v \cdot \sqrt{\frac{2}{\rho}} \cdot (x_0 + x_v(t)) \cdot \sqrt{p_p - p_A(t)}, \ x_v \in [-x_0, x_{\max}] \\ Q_{AT} &= \alpha_D \cdot \pi \cdot D_v \cdot \sqrt{\frac{2}{\rho}} \cdot (x_0 - x_v(t)) \cdot \sqrt{p_A(t) - p_T}, \ x_v \in [-x_{\max}, x_0] \\ Q_{PB} &= \alpha_D \cdot \pi \cdot D_v \cdot \sqrt{\frac{2}{\rho}} \cdot (x_0 - x_v(t)) \cdot \sqrt{p_p - p_B(t)}, \ x_v \in [-x_{\max}, x_0] \\ Q_{BT} &= \alpha_D \cdot \pi \cdot D_v \cdot \sqrt{\frac{2}{\rho}} \cdot (x_0 + x_v(t)) \cdot \sqrt{p_B(t) - p_T}, \ x_v \in [-x_0, x_{\max}] \end{aligned} \tag{33.4}$$

trong đó Q_{PA} [m^3/s] là lưu lượng tới động cơ thủy lực, từ bơm tới khoang chứa A của động cơ; Q_{AT} [m^3/s] là lưu lượng từ khoang A tới buồng chứa; Q_{PB} [m^3/s] là lưu lượng từ bơm tới khoang B của động cơ; Q_{BT} [m^3/s] là lưu lượng từ khoang B tới buồng chứa; α_D [-] là hệ số xả; D_v [m] là đường kính của ống cuộn; x_0 [m] là độ quá hành trình của ống cuộn; x_v [m] là độ dịch chuyển của ống cuộn; p_A [N/m] là áp suất chất lỏng trong khoang A; p_B [N/m] là áp suất chất lỏng trong khoang B. Lưu lượng truyền tới LHM và đi ra là Q_A và Q_B được tính như sau:

$$Q_A = Q_{PA} - Q_{AT}, \quad Q_B = Q_{BT} - Q_{PB} \tag{33.5}$$

Giả sử x_0=0 (Zero-lap), do đó đặc tính tĩnh là tuyến tính xung quanh điểm gốc và trong phần còn lại. Với $Q_{PA} = \alpha_D \cdot \pi \cdot D_v \cdot \sqrt{2/p}$ ta có các phương trình sau:

$$Q_{PA} = Q_0 \cdot x_v \cdot \sqrt{p_p - p_A}, \quad Q_{AT} = Q_0 \cdot (-x_v) \cdot \sqrt{p_A - p_T}$$
$$Q_{PB} = Q_0 \cdot (-x_v) \cdot \sqrt{p_p - p_B}, \quad Q_{BT} = Q_0 \cdot x_v \cdot \sqrt{p_B - p_T} \tag{33.6}$$

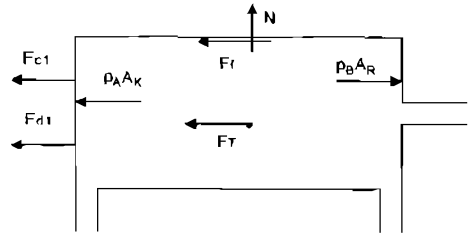


HÌNH 33.7 Mô hình toán học phi tuyến bậc nhất của van servo

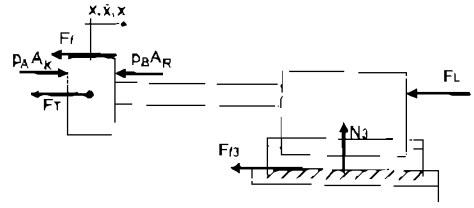
Mô hình toán học phi tuyến của động cơ thủy lực tuyến tính

Các phương trình vi phân trên cơ sở mô hình toán học của động cơ thủy lực tuyến tính (LHM) bao gồm:

i. Phương trình cân bằng động học của lực được quy về tay đòn của động cơ và
ii. Phương trình dịch chuyển và tính liên tục của lưu lượng.



HÌNH 33.8 Nguyên tắc D'Alembert được áp dụng cho xi lanh



HÌNH 33.9 Nguyên tắc D'Alembert được áp dụng cho tay đòn, pít-tông và tải

Khi mô hình hóa LHM, nhiều giả sử đơn giản hóa (phần lớn liên quan đến ma sát Coulomb và Newton) được sử dụng. Những lực tác dụng lên xi lanh của LHM được mô tả trên hình 33.8.

Áp dụng nguyên tắc D'Alembert, phương trình cân bằng động học của lực cho xi lanh của LHM là:

$$\dot{x}_2 = \frac{p_B \cdot A_R - p_A \cdot A_k + c_1 \cdot (x_1 - x_2) + d_1 \cdot (\dot{x}_1 - \dot{x}_2) + c_{fu} \cdot N \cdot \mathrm{sgn}(\dot{x} - \dot{x}_2) + d_z \cdot (\dot{x} - \dot{x}_2)}{m_2 \cdot s} \tag{33.7}$$

trong đó c_1 [N/m] là độ đàn hồi; d_1 [N/(m/s)] là hệ số tuyến tính hóa của ma sát chất lỏng Newton trong thành và xi lanh cơ cấu chấp hành kết nối; c_{fu} [-] là hệ số ma sát khô Coulomb trong xi lanh và cánh tay đòn; d_z [N/(m/s)] là hệ số ma sát Newton trong pít-tông và cánh tay đòn; m_2 [kg] là khối lượng của xi lanh; p_A [N/m^2] là áp suất chất lỏng trong khoang nạp A; p_B [N/m^2] là áp suất

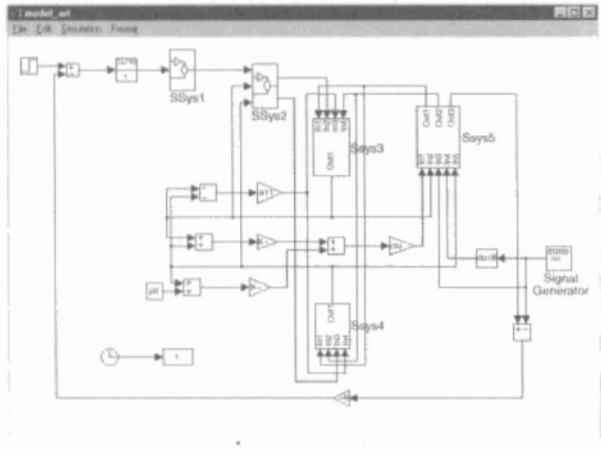
chất lỏng trong khoang xả B của cơ cấu chấp hành. A_k [m^2] là phạm vi hoạt động của pít-tông trong khoang chứa A; A_R [m^2] là phạm vi hoạt động của pít-tông trong khoang B; N [N] là lực chuẩn để xác định lực ma sát giữa pít-tông và xi lanh; x [m] là độ dịch chuyển của pít-tông; x_1 [m] là độ dịch chuyển của thành; x_2 [m^2] là độ dịch chuyển của xi lanh.

Những lực tác dụng lên cánh tay đòn, pít-tông và phần tử hoạt động được minh hoạ trên hình 33.9.

Vận tốc tương ứng của tay đòn, pít-tông và khối lượng m_3 (thanh trượt, đường dẫn và tải được xem như là giữ cố định) được suy ra từ phương trình cân bằng:

$$\dot{x} = \frac{p_A \cdot A_K - p_B \cdot A_R - F_L - c_{fu} \cdot N \cdot \mathrm{sgn}(\dot{x} - \dot{x}_2) - d_z \cdot (\dot{x} - \dot{x}_2) - c_{fu3} \cdot N_3 \cdot \mathrm{sgn}(\dot{x} - \dot{x}_1) - d_3 \cdot (\dot{x} - \dot{x}_1)}{(m_p + m_T + m_3) \cdot s} \quad (33.8)$$

trong đó m_p [kg] là khối lượng của pít-tông; m_T [kg] là khối lượng của tay đòn; m_3 [kg] là khối lượng của tải; c_{fu3} [-] là hệ số ma sát Coulomb giữa đường dẫn và thanh trượt (slade); N_3 [N] là lực chuẩn giữa tải và bàn; F_L [N] là lực của tải.



HÌNH 33.10 Trục thủy lực-điện được thực hiện trên SIMULINK

Từ những phương trình liên tục, áp suất p_A và p_B được xác định như sau:

$$\dot{p}_A = [Q_A - A_K \cdot (\dot{x} - \dot{x}_2) - a_{11} \cdot (p_A - p_B)] \cdot \frac{E_{ers}}{V_{0K} + A_K \cdot (x - x_2)} \quad (33.9)$$

$$\dot{p}_B = [A_R \cdot (\dot{x} - \dot{x}_2) + a_{11} \cdot (p_A - p_B) - a_{12} \cdot (p_B - p_0) - Q_B] \cdot \frac{E_{ers}}{(V_{0R} + A_R \cdot h) - A_R \cdot (x - x_2)} \quad (33.10)$$

trong đó a_{11}, a_{12} [(N/m^2)/(m^3/s)] là các gradien khe hở; $V_{0K,0R}$ [m^3] là thể tích trung bình ban đầu tương ứng với khoang chứa A và B; E_{ers} [N/m^2] là suất nén tương đương của dầu; $p_{A,B}$ [N/m^2] là áp suất chất lỏng tương ứng trong khoang chứa A và B; và h [m] là hành trình của tay đòn pít-tông.

Sự hoạt động của LHM dựa trên những phương trình đã giới thiệu ở trên, (33.7)-(33.10). Mô hình toán học được đề xuất trong phần này được thực hiện trên SIMULINK 2.1/MATLAB 5.1 và có cấu trúc được mô tả trên hình 33.10.

Biểu tượng máy phát tín hiệu từ hình trên tạo ra độ dịch chuyển của thành x_1 có dạng hình sin với tần số 0.5 Hz và biên độ 0.0001 m.

Các hệ con Ssys1 và Ssys2 có đầu ra là độ dịch chuyển của van x_v và lưu lượng Q_A, Q_B tương ứng. Sys3 là một khối thực hiện phương trình 33.9 và hệ con Ssys4 tạo mô hình cho phương trình 33.10. Những phương trình mô tả độ dịch chuyển của xi lanh và pít-tông LHM được thực hiện bởi hệ thống phụ Ssys5. Tín hiệu đầu vào là tín hiệu bước nhảy có dải giá trị trong khoảng 0-10V.

33.4 Điều khiển trục thủy lực-điện bằng các bộ điều khiển truyền thống

Phần này được tổ chức như sau: phần đầu tiên trình bày những nghiên cứu liên quan đến hướng truyền thống của hệ điều khiển, phần thứ hai bao gồm những nghiên cứu về cấu trúc điều khiển cổ điển (PID và các thuật toán điều khiển với bộ quan sát Luenberger) thông qua việc mô phỏng các trục thủy lực-điện sử dụng các bộ điều khiển này.

Việc kiểm tra các mô hình toán học được thực hiện với SIMULIMK nhằm mục đích đạt được các kết quả thực nghiệm để so sánh các bộ điều khiển cổ điển và cấu trúc điều khiển hiện đại được áp dụng đối với các trục thủy lực-điện.

PID, PI, PD với bộ lọc

Cấu trúc điều khiển truyền thống được sử dụng trong chương này là PI (Proportional-Integral, tỷ lệ-tích phân), PID (Proportional-Integral-Derivative, tỷ lệ-tích phân-vi phân) và PD (Proportional-Derivative, tỷ lệ-vi phân) cùng với hệ số lọc.

Hàm truyền của bộ điều khiển PI là:

$$H_{PI} = \frac{U(s)}{\varepsilon(s)} = K_R \cdot \left(1 + \frac{1}{T_i \cdot s}\right) = K_R \cdot \frac{T_i \cdot s + 1}{T_i \cdot s} \tag{33.11}$$

K_R là hệ số tỷ lệ, T_i là hằng số thời gian của thành phần tích phân.

Hàm truyền tương ứng với bộ điều khiển PID được mô tả bởi phương trình sau:

$$H_{PID} = \frac{U(s)}{\varepsilon(s)} = K_R \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right) = K_R \cdot \frac{T_i \cdot T_d \cdot s^2 + T_i \cdot s + 1}{T_i \cdot s} \tag{33.12}$$

K_R và T_i có cùng ý nghĩa như trên, T_d là hằng số thời gian của thành phần vi phân.

Hàm truyền của bộ điều khiển PD với bộ lọc là:

$$H_{PDF}(s) = \frac{U(s)}{\varepsilon(s)} = K_R \cdot \frac{1 + T_d \cdot s}{1 + \alpha \cdot T_d \cdot s} \tag{33.13}$$

Trong đó hệ số α có giá trị trong khoảng 0.1-0.125.

Nói chung, bộ điều khiển PID được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp và nó đã hoạt động tốt. Tuy nhiên, những kết quả đạt được khi sử dụng bộ điều khiển PID cho các vòng điều khiển phức tạp thì không thực sự thoả mãn bởi vì nó tốn quá nhiều thời gian để chỉnh lại bộ điều chỉnh. Bộ điều khiển PI phù hợp hơn trong các tình huống hoạt động vi phân không được sử dụng thường xuyên.

Bộ quan sát trạng thái

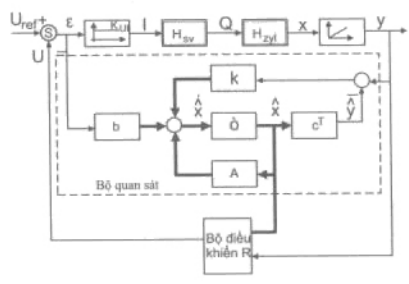
Lý thuyết về bộ quan sát, được bắt đầu bởi Luenberger và Ackermann, khá đầy đủ và hoàn thiện. Đối với trục được xét, cấu trúc bậc (n - m - 1) của bộ quan sát được chấp nhận, trong đó n = 5 thể hiện bậc của hệ thống và m = 1 là số đầu ra [25]. Mô hình điều khiển servo được mô tả bởi năm biến trạng thái: hai trong đó cho mô hình bậc hai của van servo và ba biến còn lại cho cơ cấu chấp hành servo bậc ba. Việc sử dụng bộ quan sát tuyến tính như một mô hình song song là để tổ chức lại các biến trạng thái của hệ thống và phân phối chúng tới bộ điều khiển. Hai khả năng có thể sau đây: tổ chức lại một phần hoặc toàn bộ. Giải pháp được chọn là tổ chức lại một phần [12]. Hệ thống hoàn thiện gồm có một hệ thống với tính chất phi tuyến, mô hình bậc hai song song cho van servo, động cơ servo tuyến tính bậc ba, một ma trận hiệu chỉnh cho bộ quan sát và một bộ điều khiển với năm vòng lặp cho năm biến trạng thái [23, 25, 26].

Sơ đồ khối của trục thủy lực-điện được điều khiển với bộ quan sát bậc ba thể hiện trên hình 33.11, ở đó **A**, **b**, $\mathbf{c}^T$ là ma trận đặc trưng của hệ tuyến tính (trục thủy lực-điện), **k** là ma trận hiệu chỉnh và **R** là ma trận tương ứng của bộ điều khiển.

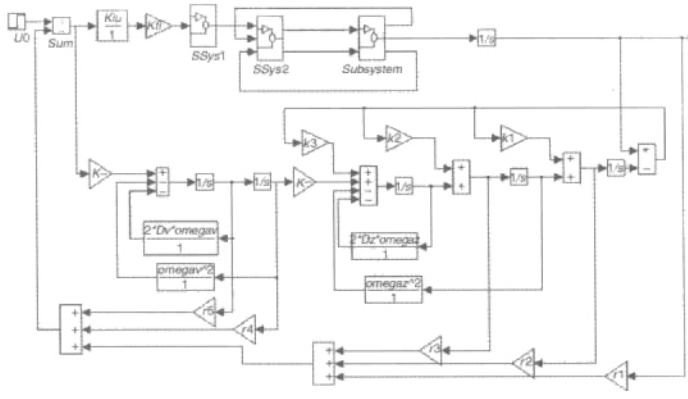
Sơ đồ mô phỏng trục thủy lực-điện được điều khiển bởi bộ quan sát bậc ba như trong SIMULINK được minh họa trên hình 33.12.

Thuật toán được sử dụng để tính toán ma trận **k** và **r** gồm có các bước sau:

i. Hoàn thiện mô hình toán học cho van servo và cơ cấu chấp hành servo;
ii. Thiết lập các biến trạng thái của quá trình;
iii. Xác định bộ điều khiển nhờ vào động học của hệ kín;
iv. Tính toán ma trận hiệu chỉnh bằng cách sử dụng các điểm cực mong muốn cho bộ quan sát [12].



HÌNH 33.11 Sơ đồ khối của vòng điều khiển sử dụng bộ quan sát



HÌNH 33.12 Cấu trúc điều khiển cùng với bộ quan sát cho trục thủy lực-điện sử dụng SIMULINK

Mô hình toán học tuyến tính (LMM)

Mô hình toán học tuyến tính của van servo và cơ cấu chấp hành servo đạt được bằng cách sử dụng các hàm truyền xấp xỉ H_{SV}, H_Z:

$$H_{SV}(s) \triangleq \frac{Q_V(s)}{I_V(s)} = \frac{k_V \omega_V^2}{s^2 + 2D_V\omega_V s + \omega_V^2} \tag{33.14}$$

$$H_Z(s) \triangleq \frac{Y_Z(s)}{Q_V(s)} = \frac{k_V \omega_Z^2}{(s^2 + 2D_Z\omega_Z s + \omega_Z^2)s} \tag{33.15}$$

với ý nghĩa sau đây: Q_v là lưu lượng của van servo; I_v là cường độ dòng điện; Y_Z là vị trí của tay đòn.

LMM trong không gian trạng thái

Các biến được sử dụng là: x_1, vị trí của tay đòn; x_2, vận tốc của tay đòn; x_3, gia tốc của tay đòn; x_4, vị trí của ống cuộn và x_5, vận tốc của ống cuộn:

$$\begin{aligned}
\dot{x}_1 &= x_2(t), \quad \dot{x}_2 = x_3(t), \qquad \dot{x}_4 = x_5(t) \\
\dot{x}_3 &= -\omega_Z^2 x_2(t) - 2D_Z\omega_Z x_3(t) + k_Z\omega_Z^2 x_4(t) \\
\dot{x}_5 &= -\omega_V^2 x_4(t) - 2D_V\omega_V x_5(t) + k_V\omega_V^2 u(t)
\end{aligned} \tag{33.16}$$

Vì vậy mô hình toán học của trục trong dạng không gian trạng thái trở thành:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \tag{33.17}$$

$$\mathbf{y}(t) = \mathbf{c}^T\mathbf{x}(t)$$

ở đó:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -\omega_Z^2 & -2D_Z\omega_Z & k_Z & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_V^2 & -2D_V\omega_V \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_V \end{pmatrix} \quad \mathbf{c}^T = (1 \quad 0 \quad 0 \quad 0 \quad 0) \tag{33.18}$$

Thiết kế bộ điều khiển

Đa thức đặc trưng thu được từ $\det\left[s\mathbf{I} - \left(\mathbf{A}_c - \mathbf{b}_c\mathbf{r}^T\right)\right] = 0$, ở đó $\mathbf{A}_c$ và $\mathbf{b}_c$ là dạng điều khiển được của ma trận **A** và **b**. Nếu $\mathbf{A} \neq \mathbf{A}_c$, sử dụng ma trận biến đổi T là thích hợp để đạt được $\mathbf{A}_c$ và $\mathbf{b}_c$. Vì vậy $\mathbf{A}_c = \mathbf{TAT}^{-1}$ và $\mathbf{b}_c = \mathbf{Tb}$. Ma trận $\mathbf{F} = \mathbf{A}_c - \mathbf{b}_c\mathbf{r}^T$ có dạng:

$$\mathbf{F} = \begin{pmatrix} 0 & 1 & .. & 0 \\ 0 & 1 & .. & 0 \\ . & . & .. & . \\ 0 & 0 & .. & 1 \\ -a_0 - r_1 & -a_0 - r_2 & .. & -a_{n-1} - r_n \end{pmatrix} \tag{33.19}$$

Đa thức đặc trưng của ma trận **F** là:

$$s^n + (a_{n-1} + r_n)s^{n-1} + \cdots + (a_1 + r_2)s + (a_0 + r_1) \tag{33.20}$$

Các điểm cực được lựa chọn cho hệ kín xác định đa thức:

$$s^n + p_{n-1}s^{n-1} + p_{n-2}s^{n-2} + \cdots + p_1 s + p_0 \tag{33.21}$$

Đa thức (33.20) và (33.21) là đồng nhất, do vậy các hệ số của ma trận $\mathbf{r}_R^T$ là:

$$r_v = p_{v-1} - a_{v-1}, \qquad v = 1, \ldots, n$$

nếu:

$$\mathbf{A} = \mathbf{A}c, \qquad \mathbf{r}^T = \mathbf{r}_R^T$$

ngược lại:

$$\mathbf{r}^T = \mathbf{r}_R^T\mathbf{T} \tag{33.22}$$

Thiết kế ma trận hiệu chỉnh

Ma trận **F** là $\mathbf{F} = \mathbf{A}^* - \mathbf{K}\mathbf{c}^T$, trong đó $\mathbf{A}^*$ là ma trận quan sát. Với **F** những điểm cực được lựa chọn là $s_1, s_2, \ldots, s_n$ và:

$$\det[s\mathbf{I}-\mathbf{F}]=(s-s_1)(s-s_2)\cdots(s-s_n) \tag{33.23}$$

$$\det[s\mathbf{I}-\mathbf{F}]=s^n+f_{n-1}s^{n-1}+\cdots+f_1s+f_0 \tag{32.24}$$

Từ hai phương trình trên, tìm được các hệ số $k_1, k_2, \ldots, k_n$.

Trong trường hợp này $\mathbf{c}^T=(1\quad 0\quad 0)$ và ma trận quan sát thứ ba là:

$$\mathbf{A}^*=\begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & -\omega_z^2 & -2D_z\omega_z\end{pmatrix} \tag{32.25}$$

Ma trận hiệu chỉnh ảnh hưởng đến cách hoạt động tức thời, các điểm cực của **F** càng xa các điểm cực của $\mathbf{A}^*$ thì đáp ứng càng nhanh.

Kết quả mô phỏng trục thủy lực-điện với bộ điều khiển truyền thống

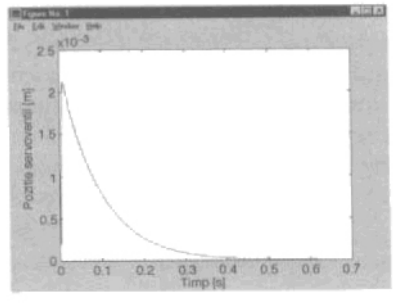
Dựa trên cơ sở các thuật toán ở trên để xác định ma trận hiệu chỉnh và ma trận của bộ điều khiển, việc thực hiện bộ quan sát trên SIMULINK bao gồm các giá trị sau:

$$r_1=19.95854,\qquad r_2=0.069481,\qquad r_3=-7.06024\times10^{-4}$$
$$r_4=-3.158688\times10^{2},\qquad r_5=-3.451209\times10^{-1}$$

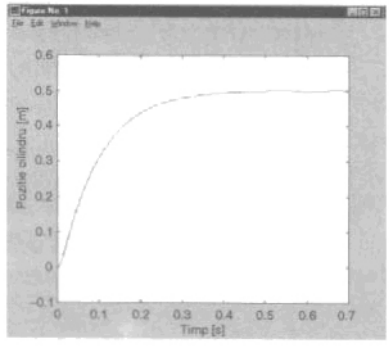
cho bộ điều khiển và:

$$k_1=1.67\times10^{-2};k_2=3.7028\times10^{4};k_3=-6.969698\times10^{6}$$ cho ma trận hiệu chỉnh

Khi tín hiệu đầu vào là bước nhảy với U = 10V, kết quả mô phỏng được chỉ ra trên hình 33.13 và 33.14.



HÌNH 33.13 Vị trí của van servo cho mô hình toán học với bộ quan sát

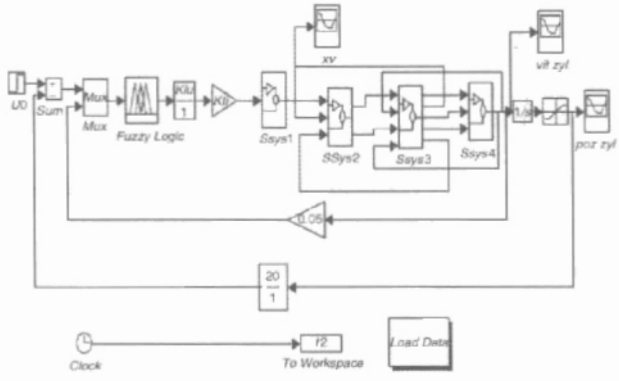


HÌNH 33.14 Vị trí của xi lanh

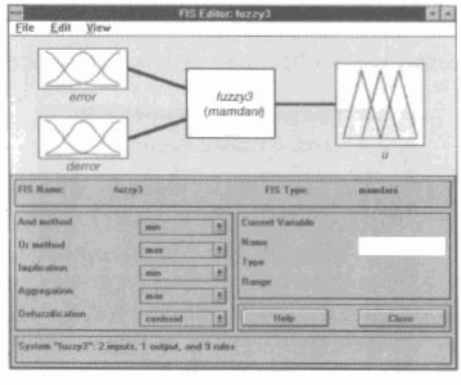
33.5 Điều khiển trục thủy lực-điện với bộ điều khiển mờ

Phần 33.5 này trình bày và kiểm tra những bộ điều khiển không truyền thống dựa trên tập mờ. Kết quả mô phỏng trục thủy lực-điện với bộ điều khiển SUGENO và MAMDANI sẽ được giới thiệu. Với cùng một số các qui tắc suy diễn được rút ra từ cơ sở kiến thức, kết quả mô phỏng chứng minh rằng đặc tính động học được cải thiện với bộ điều khiển mờ có hai đầu vào.

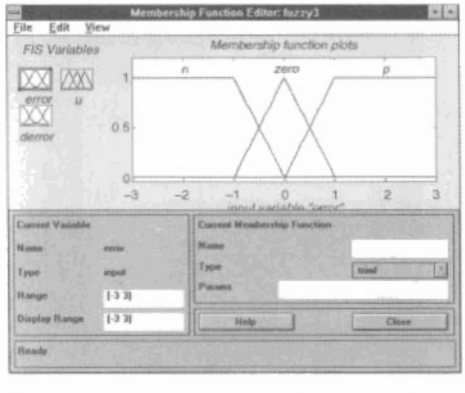
Biểu đồ được thực hiện trên SIMULINK để điều khiển trục thủy lực-điện với bộ điều khiển mờ hai đầu vào và suy diễn MAMDANI hoặc SUGENO được mô tả trên hình 33.15.



HÌNH 33.15 Điều khiển trục thủy lực-điện với bộ điều khiển mờ có hai đầu vào



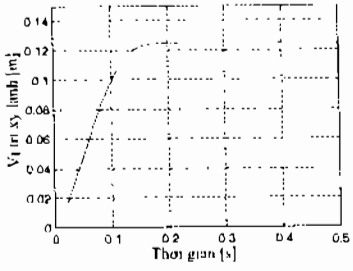
HÌNH 33.16 Bộ soạn thảo FIS cho hệ mờ dựa trên suy diễn MAMDANI



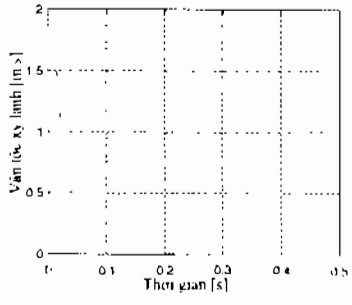
HÌNH 33.17 Hàm liên thuộc kết hợp với đầu vào

Kết quả thể hiện sự liên quan giữa mô phỏng trục thủy lực đã cho với một bộ điều khiển mờ, dựa trên suy diễn MAMDANI [59]. Hộp công cụ "Fuzzy Logic" cho phép người sử dụng có thể tạo ra hệ mờ MAMDANI và SUGENO bằng cách sử dụng giao diện đồ hoạ. Bộ soạn thảo hệ thống suy diễn mờ FIS (Fuzzy Inference System), bộ soạn thảo hàm thuộc (Membership Function Editor) và bộ soạn thảo qui tắc suy diễn (Inference Rules Editor) là những công cụ có sẵn trên SIMULINK. Ví dụ, bộ soạn thảo hệ thống suy diễn và hàm thuộc của mỗi đầu vào cho bộ điều khiển mờ cùng với suy diễn MAMDANI và hai đầu vào được minh họa trên hình 33.16 và 33.17.

Bộ điều khiển mờ này có chín hàm suy diễn được lựa chọn, có thể hình dung rõ hơn khi sử dụng bộ soạn thảo qui tắc suy diễn của SIMULIMK. Các kết quả mô phỏng của trục thủy lực-điện với bộ điều khiển mờ được trình bày trên hình 33.18 và 33.19 và mô tả vị trí và vận tốc xi lanh bằng đồ thị.



HÌNH 33.18 Vị trí LHM



HÌNH 33.19 Vận tốc LHM

33.6 Điều khiển trục thủy lực-điện dùng kỹ thuật nơron

Mục tiêu của phần này là: nhấn mạnh khả năng sử dụng các hộp công cụ mở rộng của MATLAB để thiết kế hệ thống điều khiển dựa trên kỹ thuật điều khiển hiện đại như mạng nơron; kiểm tra các thuật toán nơron này; thực hiện cấu trúc điều khiển nơron được áp dụng cho trục thủy lực-điện đã nghiên cứu.

Có hai hướng nghiên cứu chính trong điều khiển nơron. Thứ nhất là phát triển các bộ điều khiển từ mạng nơron. Thứ hai là nhúng các bộ điều khiển vào trong mạng nơron [50]. Phần này giới thiệu việc điều khiển trục thủy lực-điện bằng bộ điều khiển nơron được sử dụng rộng rãi, đó là mạng Perceptron đa lớp (Multilayer Perceptron - MLP).

Kỹ thuật điều khiển nơron

Học dựa trên sự bắt chước

Ý tưởng đến từ hệ sinh học, học bằng cách bắt chước được áp dụng vào hệ điều khiển. Mạng nơron có giám sát có thể bắt chước hoạt động của của một hệ thống khác. Phương pháp đầu tiên

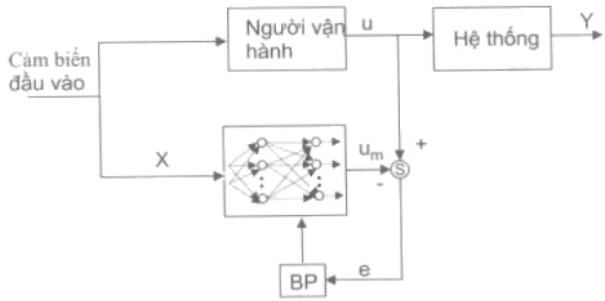
phát triển một bộ điều khiển nơron là để tái tạo một bộ điều khiển con người. Bộ điều khiển nơron cố gắng cư xử như hoạt động của con người. Huấn luyện nơron có nghĩa là học quan hệ giữa thông tin nhận được từ hoạt động của con người và đầu vào điều khiển (hình 33.20).

Học ngược

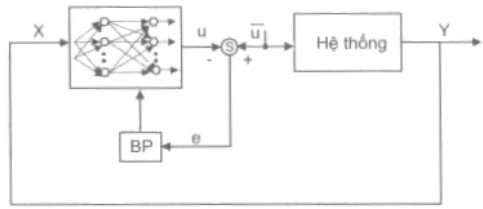
Mục đích của điều khiển ngược là nhằm điều khiển hệ thống bằng cách sử dụng động học ngược. Trong trường hợp này, mạng nơron nhận đầu ra của hệ thống làm đầu vào và đầu vào của hệ thống làm đầu ra. Hệ thống làm việc trong chế độ hệ hở và phải ở trong miền mà bộ điều khiển hoạt động. Học ngược (hình 33.21) là một hướng trực tiếp để giảm thiểu sai lệch đầu ra của mạng thay vì sai lệch của toàn hệ thống.

Học ngược chuyên sâu

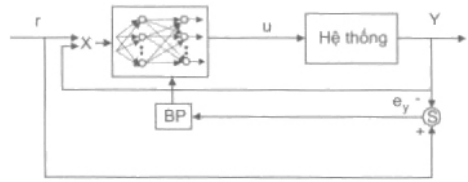
Theo Psaltis, người đề xuất *học ngược chuyên sâu* năm 1988, mạng nơron nên được huấn luyện tập trực tuyến để giảm thiểu sai lệch điều khiển $e_y = r - y$ (xem hình 33.22).



HÌNH 33.20 Học dựa trên sự bắt trước



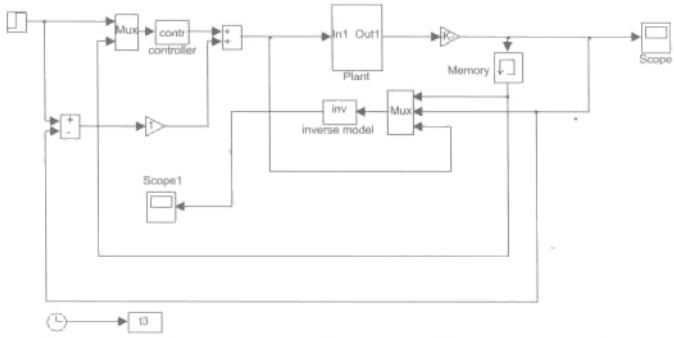
HÌNH 33.21 Giai đoạn huấn luyện trong học ngược



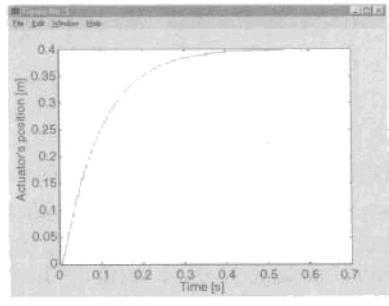
HÌNH 33.22 Kiến trúc điều khiển ngược chuyên sâu

Bộ điều khiển nơron được sử dụng để điều khiển vị trí của một trục thủy lực-điện là mạng nơron nhiều lớp truyền thẳng, thuật học là *lan truyền ngược*. Để hiệu chỉnh các trọng số của mạng nơron phải thực hiện hai bước: quá trình lan truyền thuận tín hiệu đầu và quá trình truyền ngược sai lệch đầu ra. Cấu trúc điều khiển được thực hiện trên SIMULINK như trên hình 33.23. Điều khiển nơron của trục thủy lực-điện và việc tính toán các tham số điều khiển được thực hiện trực tuyến.

Một mạng nơron 4 lớp với 2 nơron ở lớp thứ nhất, một nơron ở lớp cuối và 5 nơron ở mỗi lớp ẩn được đề xuất. Đáp ứng vị trí của trục khi sử dụng mạng nơron nói trên được mô tả trong hình 33.24.



HÌNH 33.23 Cấu trúc điều khiển cho bộ điều khiển được đề xuất



HÌNH 33.24 Vị trí của trục khi điện áp đầu vào U = 8 V

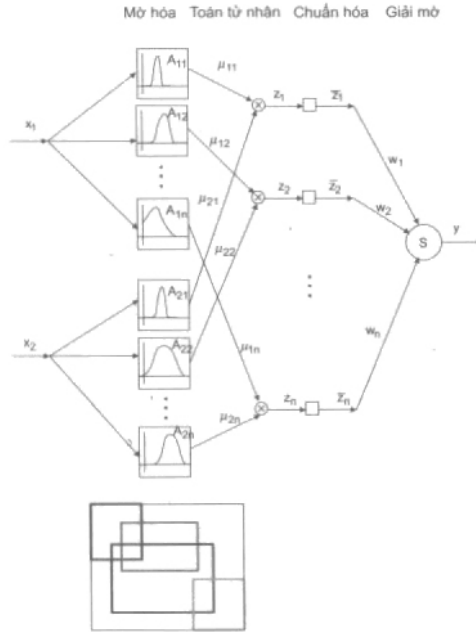
33.7 Điều khiển trục thủy lực-điện dùng kỹ thuật mờ-nơron

Chương này đề cập các kỹ thuật thiết kế có sự hỗ trợ của máy tính của thuật toán điều khiển lai. Phần này sẽ tập trung vào các dạng thuật toán trên vì mô phỏng trục thủy lực-điện với bộ điều khiển mờ-nơron cho thấy kết quả có thể bằng hoặc tốt hơn các thuật toán khác. Thuật toán mờ-nơron mang đến một sự mới lạ nhưng SIMULINK không có đủ các công cụ cho chúng, dẫn đến việc phát triển thư viện các chương trình C++ hỗ trợ cho SIMULINK trong thiết kế các bộ điều khiển. Do đó việc hiệu chỉnh trực tuyến các tham số của bộ điều khiển mờ được thực hiện. Quá trình nghiên cứu so sánh các thuật điều khiển truyền thống và hiện đại được thực hiện dựa trên cơ sở sai lệch bình phương tổng thể được tính toán từ giá trị quan sát tức thời.

Do hệ mờ có khả năng xử lý các thông tin không hoàn toàn chính xác nên chúng thường được sử dụng để biểu diễn kiến thức ở dạng qui tắc ngôn ngữ. Theo cách này thì kiến thức về hoạt động của con người ở dạng ngôn ngữ hoặc số được sử dụng để tạo ra tập các qui tắc mờ "nếu-thì" như là cơ sở cho bộ điều khiển mờ. Một trở ngại lớn của hệ mờ là rất khó để thiết kế chúng dựa trên một phương pháp luận có hệ thống. Để vượt qua trở ngại này thì quá trình học của mạng nơron được áp dụng để điều chỉnh tham số của hàm thuộc. Sự liên kết giữa mạng nơron và logic mờ đã tạo ra những bộ điều khiển mờ-nơron. Có thể khẳng định rằng bộ điều khiển mờ-nơron có được các đặc điểm cơ bản của cả hệ mờ và mạng nơron.

Bộ điều khiển mờ-nơron được đề xuất có cấu trúc dựa trên phương pháp Takagi-Sugeto và nó được mô tả trong hình 33.25.

Quá trình học trong thực tế thường là bài toán ước lượng tham số. Quá trình học cho bộ điều khiển mờ-nơron được đề xuất là suy giảm gradien. Phương pháp được áp dụng để thiết kế bộ điều khiển như vậy gọi là học ngược, trong đó kỹ thuật trực tuyến được sử dụng để tạo ra mô hình động học ngược của đối tượng. Mô hình mờ-nơron đạt được (động học ngược của đối tượng) được sử dụng để tạo ra các tín hiệu điều khiển.



HÌNH 33.25 Cấu trúc của một bộ điều khiển mờ-nơron

Bộ điều khiển mờ-nơron là một hệ thống nhiều lớp nối tiếp, một hệ thống logic mờ nhiều đầu vào và một đầu ra. Mạng có ba lớp: một lớp đầu vào với n x m đơn vị, một lớp ẩn với n đơn vị và một lớp đầu ra với một đơn vị [15]. Sự phân chia được sử dụng trong mô hình là sự phân chia phân tán [33].

Hình 33.25 giới thiệu một trường hợp đặc biệt, trong đó bộ điều khiển mờ có hai đầu vào và một đầu ra. Trong trường hợp tổng quát, bộ điều khiển mờ có m đầu vào và một đầu ra.

Cơ sở qui tắc mờ gồm một tập hợp n qui tắc ngôn ngữ có dạng như sau:

R_i: Nếu x_1 là A_{1i} và x_2 là A_{2i}

và…

và x_m là A_{mi}

thì y là w_i, i=1, 2…, n

trong đó i là chỉ số của qui tắc; A_{ji} là một tập mờ cho biến ngôn ngữ thứ j và qui tắc thứ i và w_i là một số thể hiện phần kết quả.

Hàm thuộc gán cho mỗi đầu vào là hàm Gaussian. Tâm của hàm thuộc được lựa chọn sao cho các hàm này là phân bố chuẩn trên toàn dải biểu diễn:

$$\mu_{ji} = e^{-(x_j - a_{ji})^2 / 2b_{ji}^2} \tag{33.26}$$

Suy diễn mờ trong bộ điều khiển mờ-nơron là một toán tử nhân T-norm được định nghĩa như là một hàm liên kết *and*. Giá trị đầu ra của mỗi qui tắc là:

$$z_i = \mu_{1i} \cdot \mu_{2i} \cdot \ldots \cdot \mu_{mi}, \quad i = 1,\ldots,n \tag{33.27}$$

Đầu ra là kết quả của việc xác định trọng tâm:

$$y = \frac{\sum_{i=1}^{n} z_i \cdot w_i}{\sum_{i=1}^{n} z_i} = \frac{z_1}{\sum_{i=1}^{n} z_i} \cdot w_1 + \frac{z_2}{\sum_{i=1}^{n} z_i} \cdot w_2 + \cdots + \frac{z_n}{\sum_{i=1}^{n} z_i} \cdot w_n = \sum_{i=1}^{n} \overline{z}_i \cdot w_i \tag{33.28}$$

Các tham số được xác định bằng cách tìm giá trị nhỏ nhất của hàm tục tiêu sau:

$$J(k) = \frac{1}{2} \cdot (y(k) - y_d(k))^2 \tag{33.29}$$

trong đó $y_d(k)$ là đầu ra mong muốn và $y(k)$ là đáp ứng đạt được tại thời điểm k.

Phương pháp xấp xỉ thống kê được sử dụng để tối ưu hàm mục tiêu này. Quá trình học có nghĩa là ước lượng các tham số và dựa trên thuật toán bình phương nhỏ nhất. Tham số được ước lượng là:

$$p = (a_{11}, \ldots, a_{nm}, b_{11}, \ldots, b_{nm}, w_1, \ldots, w_n) \tag{33.30}$$

Phương trình điều chỉnh tham số như sau:

$$\begin{aligned} a_{ji}(t+1) &= a_{ji}(t) - \lambda_a \frac{z_i}{\sum_{l=1}^{n} z_l} \cdot (y - y_d) \cdot (w_i - y) \cdot \frac{x_j - a_{ji}(t)}{b_{ji}^2} \\ b_{ji}(t+1) &= b_{ji}(t) - \lambda_b \frac{z_i}{\sum_{l=1}^{n} z_l} \cdot (y - y_d) \cdot (w_i - y) \cdot \frac{(x_j - a_{ji}(t))^2}{b_{ji}^3} \\ w_i(t+1) &= w_i(t) - \lambda_w \frac{z_i}{\sum_{l=1}^{n} z_l} \cdot (y - y_d) \end{aligned} \tag{33.31}$$

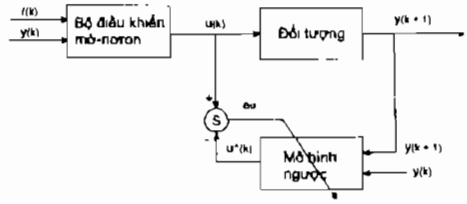
trong đó các hệ số học $\lambda_a, \lambda_b, \lambda_w$ được xác định trước.

Trong quá trình học, các tham số có thể được điều chỉnh là (a_{ji}, b_{ji}) mô tả hàm Gaussian và w_i là giá trị cuối cùng. Nếu cấu trúc của hàm thuộc được xác định thì chỉ có w_i được hiệu chỉnh.

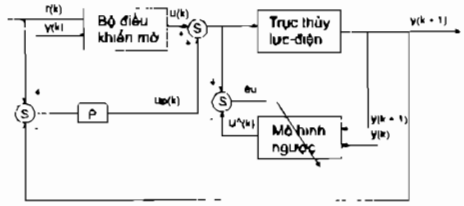
Cấu trúc điều khiển

Để thiết kế bộ điều khiển mờ-nơron đã được đề xuất ở trên, phương pháp học ngược được áp dụng. Điều khiển trục thủy lực-điện bao hàm việc sử dụng kỹ thuật trực tuyến để tạo mô hình động học ngược của đối tượng. Sơ đồ khối cho học ngược trực tuyến được mô tả trên hình 33.26.

Sơ đồ này là một hệ hở và nó cũng được tìm ra bởi phương pháp sai lệch đầu ra bộ điều khiển (Controller Output Error Method - COEM) [1] để điều chỉnh trực tuyến những tham số của bộ điều khiển mờ. Phương pháp này không đòi hỏi sai lệch đầu ra của đối tượng được phản hồi về đầu vào. Có một sự ràng buộc khác là bộ điều khiển phải có khả năng ổn định đối tượng trước khi bắt đầu hiệu chỉnh. Để tránh đòi hỏi này, phương pháp MCOEM (Modified COEM) [2] được sử dụng. Sơ đồ khối trong trường hợp này được mô tả trên hình 33.27.



HÌNH 33.26 Sơ đồ điều khiển dựa trên học ngược



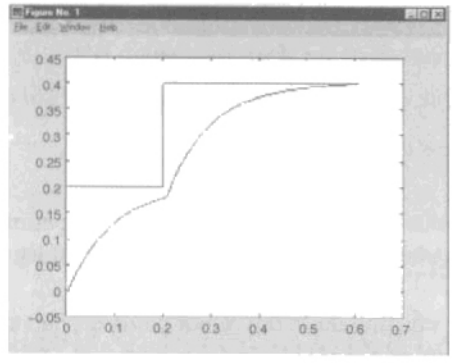
HÌNH 33.27 Sơ đồ khối của học ngược với bộ điều khiển tỷ lệ

Bộ điều khiển phản hồi tỷ lệ P được sử dụng và trong trường hợp này đầu vào của đối tượng là tổng của $u'(k)$ và $u_p(k)$. Bộ điều khiển P được lựa chọn sao cho có thể ổn định đối tượng. Cấu trúc và tham số của mô hình ngược và của bộ điều khiển mờ-nơron là đồng nhất.

Có hai giai đoạn trong thiết kế một bộ điều khiển là: điều khiển và hiệu chỉnh. Trong giai đoạn điều khiển, đầu ra của đối tượng và tín hiệu tham chiếu xác định lệnh điều khiển *u(k)*. Đầu vào của đối tượng trở thành *u(k)*, là tổng của *u(k)* và $u_p(k)$. Trong giai đoạn hiệu chỉnh, mô hình ngược (có đầu vào là *y(k+1)* và *y(k)*) tạo ra tín hiệu $\hat{u}(k)$ ở đầu ra. Tín hiệu này được sử dụng để tính toán sai lệch $e_u(k)$, từ đó xác định hàm mục tiêu *J(k)* cần tối thiểu hoá.

$$J(k) = \frac{1}{2} \cdot e_u^2(k) = \frac{1}{2} \cdot (u(k) - \hat{u}(k))^2 \tag{33.32}$$

Quá trình này được sử dụng trong điều khiển vị trí trục thủy lực-điện, ở đó tham số bộ điều khiển được xác định trực tuyến. Vị trí cơ cấu chấp hành đạt được khi tín hiệu tham chiếu thay đổi từ $U = 4$V đến $U = 8$V như mô tả trên hình 33.28.



HÌNH 33.28 Điều khiển vị trí với bộ điều khiển mờ-nơron

Để so sánh thuật toán điều khiển hiện đại với cấu trúc truyền thống, người ta sử dụng hai tiêu chuẩn tích phân: hệ số tích phân sai lệch tuyệt đối (IAE) và tích phân sai lệch bình phương (ISE). Kết quả thu được khi áp dụng tiêu chuẩn này thể hiện ở bảng 33.1.

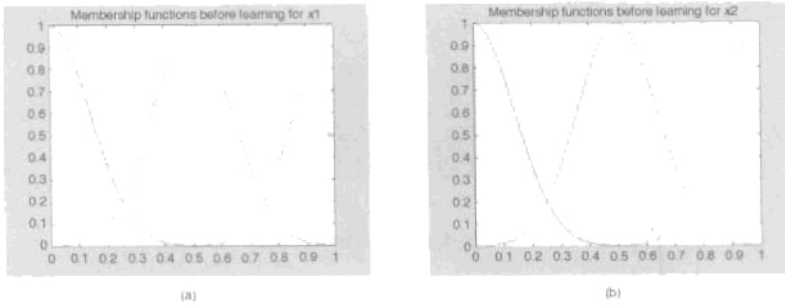
BẢNG 33.1

Bộ điều khiển	IAE	ISE
PID	0.8024	3.4754
PI	0.8006	3.4618
PD	0.7928	3.4537
Nơron	0.8027	3.4622
Mờ-nơron	0.7911	3.4501

Theo các kết quả trước đó, có thể suy ra rằng bộ điều khiển mờ-nơron được giới thiệu tỏ ra tốt hơn so với bộ điều khiển nơron dựa trên MLP hoặc các bộ điều khiển kinh điển (PID, PI, PD) được trình bày trong quyển sách này. Những kết quả mô phỏng nhấn mạnh rằng bộ điều khiển mờ-nơron là một công cụ hữu ích cho các ứng dụng cụ thể với nhiều thành phần phi tuyến.

Kết quả tối ưu đạt được thông qua thay đổi các tập dữ liệu và số bước lặp. Để kiểm tra hoạt động của bộ điều khiển mờ-nơron đã được đề xuất, một hàm phi tuyến cho bởi một phương trình giải tích được xấp xỉ. Hàm thuộc của biến đầu vào x_1 và x_2 trước khi học được mô tả trên hình

33.29(a,b). Bề mặt thu được sau khi mô phỏng được mô tả trên hình 33.30(c). Có thể quan sát độ chính xác của cấu trúc xây dựng lại sau 300 bước học bằng cách so sánh các bề mặt đạt được.

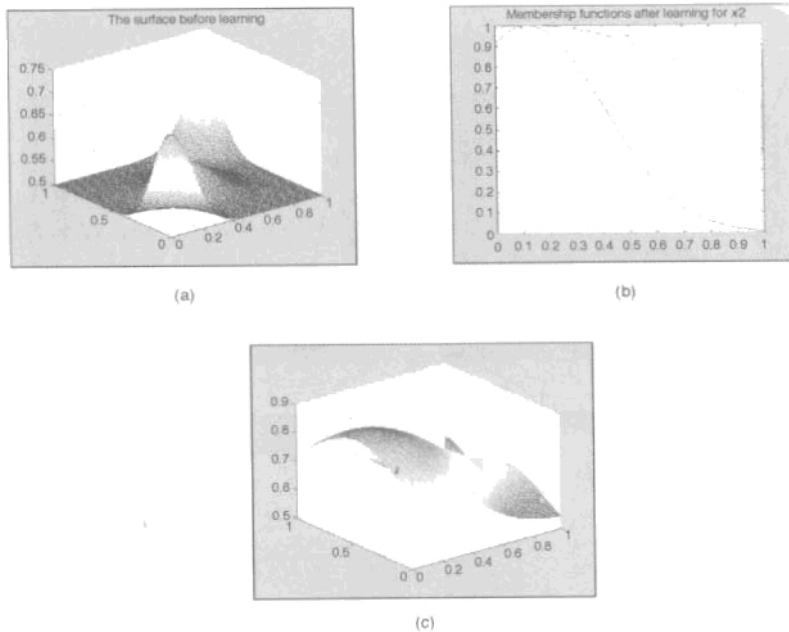


HÌNH 33.29 (a) Hàm liên trước khi học của biến x_1, (b) Hàm liên trước khi học của biến x_2

Các tập hợp kết quả trung gian thu được với các tập dữ liệu mô phỏng khác nhau được mô tả như sau đây. Các tập dữ liệu khác nhau của mô phỏng được sử dụng để đạt được kết quả tối ưu. Một vài kết quả trong số đó được thể hiện trong hình 33.30-33.34.

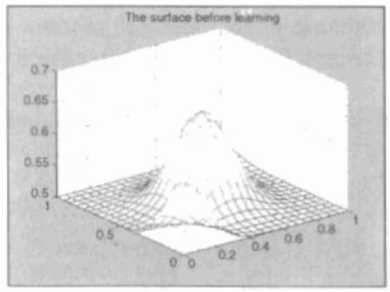
Để có được kết quả tốt từ mô hình, 10 hàm thuộc được sử dụng cho mỗi đầu vào. Các hệ số học $\lambda_a, \lambda_b, \lambda_w$ được lựa chọn bằng 0.01. Thuật toán điều khiển có khả năng quản lý sự thay đổi trong phạm vi hoạt động. Các kết quả mô phỏng trục thủy lực-điện với bộ điều khiển đề xuất đạt được với nhiều đầu vào. Các kết quả này trong miền thời gian và được thể hiện trong hình 33.35(a,b), tương ứng với điện áp đầu vào là 8 và 10V.

Tập thứ nhất: gw = 0.1; ga = 0.05; gb = 0.05; nepoc = 100; nesant = 100; niter = 200; threshold error = 0.001; Vmax = 0.04.



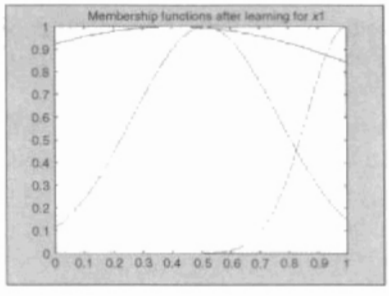
HÌNH 33.30 (a) Bề mặt đạt được sau bước lặp thứ nhất, (b) Hàm thuộc sau khi học cho biến x_2, (c) Bề mặt đạt được sau khi mô phỏng

Tập thứ hai: gw = 0.1; ga = 0.07; gb = 0.05; nepoc = 100; nesant = 21; niter = 200; threshold error = 0.001; Vmax = 0.0475.

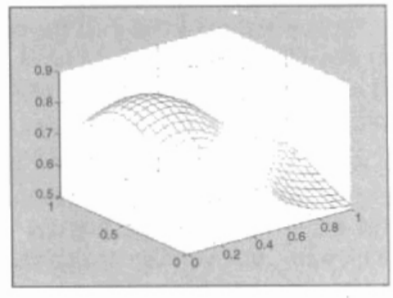


HÌNH 33.31 Bề mặt đạt được sau bước lặp thứ nhất

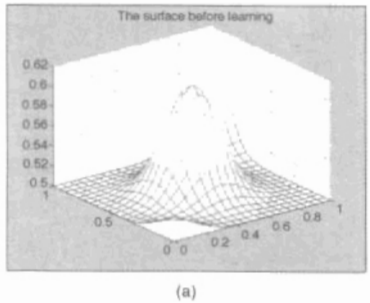
Tập thứ ba: gw = 0.5; ga = 0.07; gb = 0.03; nepoc = 200; nesant = 21; niter = 200; threshold error = 0.001; Vmax = 0.047515.



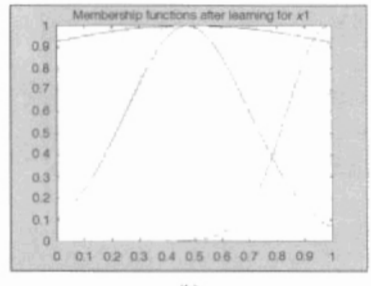
HÌNH 33.32 Hàm thuộc sau khi học cho biến x_2



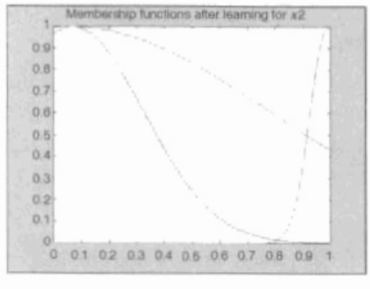
HÌNH 33.33 Bề mặt đạt được sau khi học



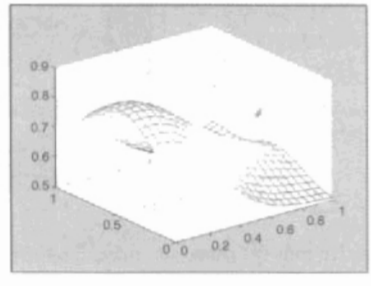
(a)



(b)

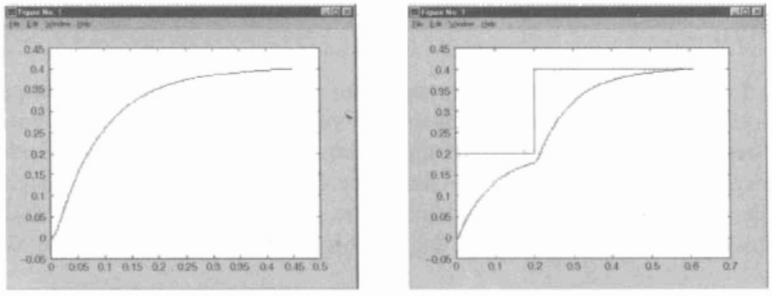


(c)



(d)

HÌNH 33.34 (a) Bề mặt đạt được sau bước lặp thứ nhất, (b) Hàm thuộc sau khi học cho biến x_1, (c) Hàm thuộc sau khi học cho biến x_2, (d) Bề mặt sau khi học



HÌNH 33.35 (a) Điều khiển vị trí với bộ điều khiển mờ-nơron (U = 8V), (b) Điều khiển vị trí với bộ điều khiển mờ-nơron (U = 10V)

33.8 Phần mềm

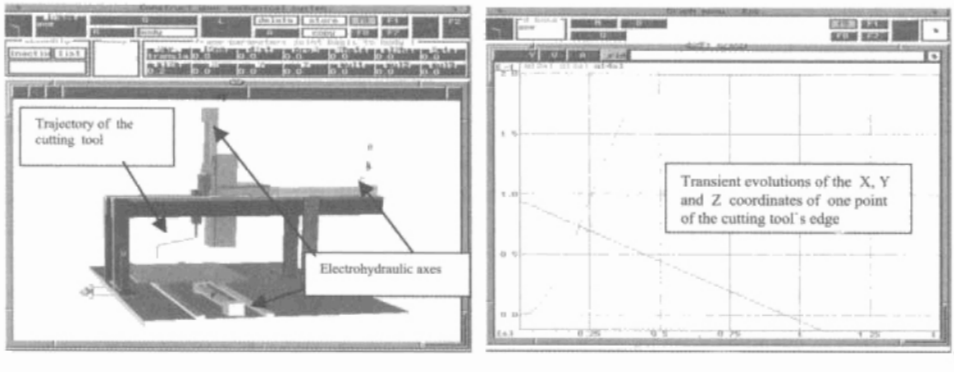
Mô hình toán học của trục thủy lực-điện được nghiên cứu trong chương này là của một hệ vật lý ở phòng thí nghiệm cơ điện tử của UAS-Konstanz (xem trên hình 33.1b). Hai biến thể của mô hình toán học phi tuyến được thiết lập trong phần 33.3 và làm tăng tính phi tuyến động và tĩnh trong hoạt động của trục thủy lực-điện [23, 58].

Mô hình toán học của truyền động thủy lực trong cấu trúc của ROBI_3 được thực hiện trên SIMULINK để nghiên cứu sự biến đổi động học của trục [26, 27]. Biến thể mở rộng của mô hình toán học các trục thủy lực được áp dụng cho chuyển động tương đối của các phần cấu thành động cơ servo.

Bộ điều khiển nơron và mờ-nơron (Takagi-Sugeno) được phát triển trên Borland C++ và thực hiện trên SIMULINK để điều khiển các trục thủy lực-điện. SIMULINK hỗ trợ người dùng một thư viện logic mờ, cho phép thiết kế và mô hình hóa hệ thống suy diễn mờ SUGENO và MAMDANI. Do không có phần mềm chuyên dụng để thiết kế các bộ điều khiển mờ-nơron nên chúng được phát triển trên C++ và sau đó sử dụng trong SIMULINK [26, 27, 28].

Với sự hỗ trợ việc mô phỏng bằng SIMULINK 2.1 và MATLAB 5.2 (dưới môi trường Window), cho phép thực hiện bộ điều khiển như một mô đun và biểu tượng tương ứng trong một toolbox đặc biệt. Quá trình thực nghiệm sử dụng các tiện ích cung cấp bởi S-function và C Mex kết hợp với Borland C++ 5.0 để biên dịch chúng. C S-function được chọn vì sự cần thiết về tốc độ để xử lý thông tin trong khối thực hiện bộ điều khiển. Khối thực hiện bộ điều khiển có hai đầu vào (thậm chí có thể có ba đầu vào mặc dù việc xử lý hiệu chỉnh sẽ phức tạp hơn) và một đầu ra.

Các tham số hiệu chỉnh (trọng số, tâm, phạm vi của hàm Gaussian) phải liên tục. Khai báo toàn cục, tĩnh hoặc sử dụng không gian làm việc để chứa là những kỹ thuật hữu ích để hoàn thành nhiệm vụ.



HÌNH 33.36 Rôbốt 3 trục Cartesian được tạo mô hình, điều khiển, mô phỏng trên SDS [21, 24]

33.9 Kết luận

Nghiên cứu được thực hiện trong chương này nhằm mục đích phát triển các cấu trúc điều khiển cải tiến dựa trên kỹ thuật điều khiển hiện đại (nơron và mờ) trong mối quan hệ với cấu trúc điều khiển truyền thống. Trục thủy lực-điện được nghiên cứu là một phần của rôbốt Cartesian ROBI_3 trong phòng thí nghiệm Cơ điện tử của UAS-Constance. Mô phỏng 3D với động học thuận và ngược đã được thực hiện, cùng với đó là bộ điều khiển được phát triển dựa trên SDS-Modelling và một phần mềm mô phỏng hệ thống thực [21, 23, 25, 28]. Mô hình và kết quả mô phỏng được minh họa trên hình 33.36.

Một cách tổng quát, phần 33.1 đưa ra sự giới thiệu của chương này. Trong phần 33.2, các đặc điểm quan trọng nhất của điều khiển hệ thống thủy lực-điện và tính chất phi tuyến của hệ thống được giới thiệu. Rôbốt ROBI_3 được trình bày như là một thành phần cũng như dưới quan điểm điều khiển.

Mô hình toán học trục thủy lực-điện của ROBI_3 được giới thiệu trong phần 33.3. Mô hình toán học phi tuyến được hoàn thành dựa trên dữ liệu kỹ thuật của các thành phần hệ thống. Kết quả mô phỏng mô hình toán học phi tuyến trong vòng lặp vị trí nhận được nhờ phần mềm SIMULINK/MATLAB. Lưu ý rằng tất cả các mô phỏng trong chương này được thực hiện nhờ phần mềm MATLAB/SIMULINK, trong khi đó các thuật toán điều khiển hiện đại (nơron và mơ-nơron) được phát triển trên Borland C++ 5.0.

Phần 33.4 là tổng quan ngắn gọn về lý thuyết các bộ điều khiển truyền thống PID, PD, PI và bộ quan sát, cùng với các kết quả mô phỏng trục thủy lực-điện với các bộ điều khiển trên.

Phần 33.5 nhắc lại về lý thuyết hệ mờ và giới thiệu kết quả mô phỏng các trục thủy lực-điện với bộ điều khiển mờ. Lý thuyết hệ mờ góp phần quan trọng vào việc mô hình hóa hệ thống và phát triển một khung lý thuyết phù hợp cho việc thực hiện những lập luận định tính của con người. Lập luận này rất hữu ích để mô hình hóa các hệ thống phức tạp có tính phi tuyến và thông tin không rõ ràng. Các kết quả mô phỏng trục thủy lực-điện đạt được bằng cách sử dụng bộ điều khiển mờ SUGENO và MAMDANI.

Sơ lược về lý thuyết mạng nơron, cấu trúc nơron phổ biến nhất và các kỹ thuật điều khiển nơron được trình bày trong phần đầu của mục 33.6. Mạng nơron làm việc dựa trên sự định lượng và số lượng. Nếu logic mờ có được một suy luận dựa trên các thông tin không rõ ràng thì mạng nơron học bằng cách huấn luyện và khi kết thúc quá trình đó mạng xấp xỉ một hàm mong muốn. Việc phân tích các mạng nơron được huấn luyện gặp nhiều khó khăn và kết quả là các luật thường không được rút ra từ các mạng được huấn luyện. Phần 33.6 là kết quả mô phỏng của mạng nơron nhiều lớp.

Hệ mờ-nơron giữ được đặc tính của cả hệ mờ và nơron và được sử dụng thành công trong lĩnh vực điều khiển những năm gần đây. Toàn bộ phần 33.7 trình bày về lý thuyết hệ mờ-nơron, bộ điều khiển mờ-nơron được thực hiện trên Borland C++ và áp dụng trong SIMULINK cho các trục thuỷ lực điện tử, một số kết quả mô phỏng và kết quả so sánh giữa các bộ điều khiển hiện đại và truyền thống.

Phần 33.8 trình bày ngắn gọn về chương này và một số kết quả chính trong phạm vi chủ đề cũng như triển vọng của hướng nghiên cứu này trong tương lai.

Sau đây là một danh sách các hướng nghiên cứu có thể và triển vọng phát triển trong tương lai:

- Ứng dụng các bộ điều khiển được thực hiện trên SIMULINK không chỉ để điều khiển các trục thủy lực-điện đã thảo luận mà còn áp dụng cho các hệ thống có cấu trúc phức tạp trong một hệ thủy lực lớn, cung cấp cho người sử dụng một thư viện bộ điều khiển mờ-nơron;
- Thực hiện bộ điều khiển mờ-nơron đã trình bày bằng phần cứng;
- Nghiên cứu phát triển bộ điều khiển tối ưu;
- Tích hợp các bộ điều khiển đã trình bày trong một gói phần mềm dành cho lĩnh vực thủy khí;

- Phát triển kỹ thuật thiết kế bộ điều khiển để cải tiến các bộ điều khiển này, tạo ra sự cộng sinh tốt hơn giữa phương pháp truyền thống và hiện đại (mờ-nơron và thuật toán di truyền);
- Mở rộng phạm vi nghiên cứu về điều khiển vận tốc, gia tốc, áp suất, lưu lượng, lực, mômen và công suất.

Tài liệu tham khảo

[1] Andersen, H.C., Lotfi, A., Tsoi, A.C. A new approach to adaptive fuzzy control: the controller output error method, *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-27-B(4), August 1997.

[2] Abonyi, J., Nagy, L., Szeifert, F. Indirect adaptive Sugeno fuzzy control, *Proceedings in Artificial Intelligence*, FNS'98, München, Germany,19–20 martie.

[3] Backé, W. *Systematik der hydraulischen Widerstandsschaltungen in Ventilen und Regelkreisen.* Krauskopf-Verlag, Mainz, 1974.

[4] Costa Branco, P.J., Dente, J.A. *Inverse-Model Compensation Using Fuzzy Modeling and Fuzzy Learning Schemes.* Intelligent Engineering Systems through Artificial Neural Networks, Smart Engineering Systems: Fuzzy Logic and Evolutionary Programming, Ed. C.H. Dagli, M. Akay et al. Vol. 6, ASME Press, New York, pp. 237–242, 1996.

[5] Brown, M., Harris, C. *Neuro-fuzzy Adaptive Modelling and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[6] Catana, I., Vasiliu, D., Vasiliu, N. *Servomecanisme electrohidraulice. Constructie, functionare, modelare, simulare si proiectare asistata de calculator.* U.P.B. Bucuresti, 1995.

[7] Cybenko, G. *Mathematical Problems in Neural Computing.* Signal Processing Scattering and Operator Theory and Numerical Processing, Vol. 3, Kashoek, M.A., van Schupper, J.H., Ram, A.C. Ed., 1989, pp. 47–64.

[8] Driankov, D., Hellendoorn, H., Reinfrank, M. *An Introduction to Fuzzy Control.* Springer-Verlag, Berlin, 1993.

[9] Dubois, D., Prade, H., Ughetto, L. Checking the coherence and redundancy of fuzzy knowledge bases, *IEEE Trans. on Fuzzy Systems*, 5(5):398–417, 1997.

[10] Dumitrache, I. sa. *Automatizari electronice.* Editura Didactica si Pedagogica, Bucuresti, 1993.

[11] Dumitrache, I., Catana, I., Militaru, A. *Fuzzy Controller for Hydraulic Servosystems.* IFAC International Workshop on Trends in H& P Components & Systems, Chicago, IL, 1994.

[12] Föllinger, O. *Regelungstechnick.* Dr. A. Hütig Verlag, Heidelberg, Germany, 1978.

[13] Friedrich, A. *Logik und Fuzzy-Logik.* Expert-Verlag,1997.

[14] Ghaoui, L. El. Reduced-order multimodel control using linear matrix inequalities: sufficient conditions, *Proc.* Od ACC 1993, 1993, pp. 633–634.

[15] Godjevac, J., Steele, N. Adaptive neuro-fuzzy controller for navigation of mobile rôbôt, *International Symposium on Neuro-Fuzzy Systems AT'96*, Conf. Report, EPFL-Lausanne, 1996.

[16] Gupta, M.M. *Fuzzy Logic and Neural Networks*, Proc. of the 2nd International Conference on Fuzzy Logic & Neural networks, Iizuka, Japan, 17–22 July, 1992, pp. 187–188.

[17] Healey, M. *Principles of Automatic Control.* The English Universities Press Ltd., 1975.

[18] Haykin, S. *Neural Networks*, MacMillan College Publishing Company, New York, 1994.

[19] Ionescu, Fl. Computer aided design of hydraulic and electrohydraulic drive installations, *Proceed. 9th Triennial World IFAC Congress*, Budapesta, Ungaria, Pergamon Press, Vol. 1, 1984, pp. 569–574.

[20] Ionescu, Fl., Stoffel, B. Contribution to the Automatic Generation of Mathematical Models for the Computer Assisted Analysis and Synthesis of Hydraulic Drive Systems. Proceed. of the 2nd Intern. Conf. on Fluid Power, 19–21 March 1991, Tampere, Finland, pp. 469–482.

[21] Ionescu, Fl., Haszler, Fl. TORCH: A *Control Software for Electrohydraulic Cartesians Róbóts*. Proceed. of the 6th Intern. IMEKO Symposium on Measurement and Control in Rôbôtics, ICMR'96, Bruxelles, Belgium, 9–11 May, 1996, pp. 484–489.

[22] Ionescu, Fl. *Non-Linear Problems in the Hydraulic Drive Systems.* 2nd World Congress of Nonlinear Analysts, Athena, Greece, 10–17 July 1996, Pergamon Press, Vol. 30, part 3, pp. 1447–1461.

[23] Ionescu, Fl., Vlad, C.I. Tools of HYPAS for the control of electrohydraulic drive installations, *Proc. of 7th Symposium on Computer Aided Control Systems Design*, Gent, Belgia, 1997, pp. 311–316.

[24] Ionescu, Fl., Borangiu, Th., Vlad, C.I. High integrated CAD strategies for control design of electrohydraulic systems, *Proc. 3rd IFAC Conference SSC*, Bucharest, 1997, pp. 390–395.

[25] Ionescu, Fl., Vlad, C.I. *Hypas tools for the control of electro-hydraulic drive installations Journal a*, Vol. 38, No. 3, Belgium, 1997, pp. 38–41.

[26] Ionescu, Fl., Vlad, C.I. Sugeno and hypas fuzzy-control solutions for electro-hydraulic drive installations, *Proceedings EUFIT'97*, Aachen, Germany, 8–11 Sept., Vol. 2, 1997, pp. 1238–1242.

[27] Ionescu, Fl., Vlad, C.I., Arotaritei, D. Fuzzy and neuro-fuzzy HYPAS controllers implemented for an electro-hydraulic axis, *International ICSC Symposium on Engineering of Intelligent Systems EIS'98*, Teneriffe, Spain, Feb. 11–13, 1998.

[28] Ionescu, Fl., Arotaritei, D., Vlad, C.I. *Modelling of Nonlinearities, Signal Reconstruction and Predictive Solutions Applied in Mechatronics Systems by Using Neuro-Fuzzy Systems,* Internal Report, Department of Mechatronics, FH-University of Applied Sciences-Konstanz, 1998.

[29] Ionescu,V., Varga, A. *Teoria Sistemelor. Sinteza robusta. Metode numerice de calcul.* Editura ALL, Bucuresti, 1995.

[30] Isermann, R. *Digitale Regelsysteme*, Springer-Verlag, Berlin, 1987.

[31] Isermann, R.*Zur Anwendung der Fuzzy-Logik in der Regelungstechnik.* Automatisierungstechnische Praxis (atp) Fuzzy-Control, 38, Oldenbourg Verlag, Germany, 1996.

[32] Isermann, R. On fuzzy logic application for automatic control, supervision, and fault diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, Vol. 28, No. 2, March 1998, pp. 221–235.

[33] Jang, J.-S.R., Sun, C.-T., Mizutani, E. Neuro-fuzzy and soft computing, *A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, Englewood Cliffs, NJ, 1997.

[34] Joh, J., Hong, S.K., Nam, Y., Chung, W.J. On the systematic design of Takagi-Sugeno fuzzy control systems, *International ICSC Symposium on Engineering of Intelligent Systems EIS'98*, Tenerife, Feb. 1998.

[35] Kandel, E.R. *Nerves Cell and Behavior*, Principles of Neural Sciences, 3rd ed., 1992, pp. 18–36.

[36] Knappe, H. Comparison of conventional and fuzzy-control of non-linear systems, In: Kruse, R., *Fuzzy Systems in Computer Science*, Verlag Vieweg, Wiesbaden, Germany, 1994.

[37] Kokotovic, P.V. *Lectures Notes in Control and Information Sciences*. Springer-Verlag, Berlin, 1991.

[38] Kosko, B. *Neural Networks and Fuzzy Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1992.

[39] Kovacic, Z., Balenovic, M., Bogdan, S. Sensitivity-based self-learning fuzzy logic for a servo-system, *IEEE Control Systems*, June, 1998.

[40] Lippmman, R. An introduction to computing with neural nets, *IEEE ASSPMagazine*, April 1987, pp. 4–22.

[41] MATLAB 5.2. MathWorks Corp, USA. 1998.

[42] Miller, Th., Sutton, R., Werbos, P.J. *Neural Networks for Control*, MIT Press, 1990.

[43] Nauck, D., Klawonn, F., Kruse, R. *Neuronale Netze und Fuzzy-Systeme*. Grundlagen des Konnektionismus, Neuronaler Fuzzy-Systeme und der Kopplung mit wissensbasierten Methoden, Vieweg, 1994, Germany.

[44] Nesterov, Y., Nemirovski, A. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, 1994.

[45] Pedrycz, W. *Fuzzy Control and Fuzzy Systems*, Wiley, New York, 2nd ed., 1993.

[46] Piechnik, M., Feuser, A. *Simulation mit Komfort* - HYVOS 4.0 und MOSIHS 1.0, Ö & P, 38, 1994.

[47] Postlethwaite, B.E. A model-based fuzzy controller, *Trans IChemE*, Vol. 72, Part A, Jan. 1994.

[48] Postlethwaite, B.E. Building a model-based fuzzy controller, *Fuzzy Sets and Systems*, 79(1996), Elsevier.

[49] Rehfeldt, K., Shöne, A., Büngener, N. *Einsatz von Fuzzy-Reglern zur Drehzahlregelung einer Hydraulikpumpe,Ölhydraulik und Pneumatic*, 36, Nr. 6, pp. 397–402, 1992.

[50] Ronco, E., Gawthrop, P.J. *Neural Networks for Modelling and Control*. Technical Report: csc97008, Centre for System and Control, Dept. of Mechanical Engineering, Univ. of Glasgow, 10 Nov. 1997.

[51] Simulink, Dynamic System Simulation for MATLAB, Writing S-functions, The Math Works Inc., 1998.

[52] Sontag, E.D. Mathematical control theory, *Deterministic Finite Dimensional Systems*. Springer-Verlag, Berlin, 1990.

[53] Takagi, T., Sugeno, M. Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Systems, Man, and Cybernetics*, Vol SMC-15, No. 1, pp. 116–132, 1985.

[54] Tanaka, K., Sugeno, M. Stability analysis and design of fuzzy control systems, *Fuzzy Sets and Systems*, Vol. 45, 1992, pp. 135–156.

[55] Teodorescu, H.N. *Sisteme Fuzzy si Aplicatii*. Institutul Politehnic Iasi, Romania,1989.

[56] Tertisco, M., Penescu, C., *Ionescu, G., Ceanga, E. Identificarea Experimentala a Proceselor Automatizate*. Editura Tehnica, Bucuresti, 1971.

[57] Viersma, T. J. *Analysis, Synthesis and Design of Hydraulic Servosystems and Pipelines*. Elsevier, Amsterdam- New York, 1980.

[58] Vlad, C.I. Contributions to the Direct Computer Control of Electrohydraulic Axes for Industrial Rôbôts. Technical University "Politehnica", Bucharest, Romania, 1998.

[59] Wang, L., Liu, G.P., Harris, C.J., Brown, M. *Advanced Adaptive Control*, Pergamon, 1997.

[60] Werbos, B. Overview of *Design and Capabilities*. In Neural Networks for Control, pp. 59–65, MIT Press, MA, 1990.

[61] Westcott, J.H. The minimum-moment-of-error-squared criterion: a new performance criterion for servo mechanisms, *Proc. of IEE.*, Measurements Section, pp. 471–480, 1954.

[62] Yager, R., Zadeh, L. *Fuzzy Sets, Neural Networks and Soft Computing*, 1994.

[63] Zadeh, Lotfi. *Fuzzy sets, Information & Control*, No. 8, pp. 338–353, 1965.

[64] Zadeh, L., King-Sun Fu, Tanaka, K., Shimura, M. *Fuzzy Sets and their Applications to Cognitive and Decision Processes*. Academic Press, 1975.

[65] Zimmermann, H.-J. *Fuzzy Sets Theory - and Its Applications*, Kluwer Academic Publishers. Dordrecht, The Netherlands, 1990.

34

Tối ưu thiết kế hệ cơ điện tử

Tomas Brezina
Technical University of Brno

Ctirad Kratochvil
Technical University of Brno

Cestmir Ondrusek
Technical University of Brno

34.1 Giới thiệu 34-1
34.2 Các phương pháp tối ưu 34-1
34.3 Thiết kế tối ưu động cơ cảm ứng (IM) 34-6
34.4 Sử dụng mạng nơron cho việc nhận dạng tham số của một hệ thống động lực học 34-10

34.1 Giới thiệu

Hệ cơ điện tạo nên một phần tích hợp của hệ cơ học và cơ điện tử. Việc tối ưu hoá chúng là điều kiện cần thiết cho sản phẩm có tính cạnh tranh. Trong thực tế kỹ thuật, còn rất nhiều bài toán tối ưu và nhận dạng không thể giải được nếu không sử dụng máy tính [5]. Mức độ phát triển công nghệ hiện nay được đặc trưng bởi sự tăng khả năng hoạt động của máy móc với chi phí sản xuất được giữ ở mức thoả đáng. Yêu cầu về độ tin cậy và an toàn vận hành của máy móc được thiết kế cũng cần phải quan tâm.

Từ kinh nghiệm thực tế chúng ta biết rằng tính chất động lực học của hệ cơ điện ảnh hưởng đáng kể đến độ tin cậy và an toàn. Mặt khác, xu hướng đẩy giá thành của máy giảm xuống thường dẫn đến các tính chất động lực học không thích hợp làm tăng dao động và nhiễu trong quá trình hoạt động. Các tính chất điện cũng giảm đáng kể khi số lượng vật liệu hoạt động trong máy giảm. Tải tăng lên dẫn đến sự quá nhiệt, ảnh hưởng tới cách điện và làm giảm tuổi thọ của máy.

34.2 Các phương pháp tối ưu

Nguyên lý tối ưu

Những tính chất của hệ cơ điện có thể được mô tả dưới dạng toán học bằng cách dùng các đại lượng vật lý. Khi đó cấp độ những tính chất này được mô tả bằng cách dùng hàm mục tiêu được thiết lập toán học. Các tham số cấu trúc trong phạm vi giữa các giá trị giới hạn đã cho thỏa mãn các điều kiện thứ cấp là các biến số độc lập của các hàm này. Dạng cụ thể của hàm phụ thuộc vào kiểu máy và kiểu mô tả toán học. Lời giải của bài toán tối ưu được thiết lập toán học với các phương pháp tối ưu cho phép một số lượng đáng kể các biến thể thiết kế khác nhau của máy có thể được tính toán trong thời gian tương đối ngắn. Chúng tạo ra khả năng thực hiện những tính toán này ở giai đoạn lập kế hoạch sản xuất cho một nguyên mẫu để có chất lượng theo hàm tiêu chuẩn lựa chọn. Theo cách này, thiết kế máy không chỉ được phân tích mà còn được sửa đổi và xây dựng lại về mặt tính chất cơ điện với mục tiêu cải thiện các tính chất này càng nhiều càng tốt (hoặc tối ưu chúng).

Từ quan điểm vật lý thì đây là các bài toán thực tế, ở một mức độ nào đó, ngược với các bài toán tính toán. Vấn đề tính toán giả sử là là một mô hình mô tả toán học và cố định của máy thực được sử dụng để tìm ra các tính chất kết quả của chúng. Trong vấn đề tính toán, chúng ta định nghĩa các tính chất và cố gắng tìm các tham số của mô hình toán học đã chọn có các thuộc tính đó. Trong vấn đề tối ưu tham số, chúng ta tìm các tham số này bằng hàm ưu tiên đã chọn, với những đặc tính tốt nhất. Rõ ràng là vấn đề tổng hợp và tối ưu phức tạp hơn nhiều so với tính toán.

Tối ưu tham số

Với mục tiêu là tìm các giá trị của tham số cấu trúc nào đó của máy, chúng ta sẽ bàn về khái niệm này cụ thể hơn. Bằng cách tối ưu tham số của hệ cơ điện, chúng ta muốn nói đến quá trình tìm những tham số này dưới dạng một mô tả toán học của hệ thống này (sắp xếp trong vectơ $\bar{p}$) từ tập P của các tham số có thể chấp nhận mà ở đó hàm mục tiêu $\psi(p)$ được lựa chọn thích hợp của tham số tiến tới cực trị của nó.

Hàm mục tiêu $\psi(p)$ xác định bậc của các tính chất hệ điện từ đã được tạo cực trị (chọn tham số có bậc tốt nhất của tính chất này). Khi định nghĩa một tập P có thể chấp nhận được, chúng ta được chỉ dẫn bởi khả năng cấu trúc của những thay đổi trong các tham số riêng lẻ (các biến) hoặc chúng ta có thể đưa ra một tiêu chuẩn thứ cấp: "bậc của các thuộc tính có thể không vượt qua giới hạn tiêu chuẩn". Khả năng thực hiện thay đổi cấu trúc tham số dẫn đến các ràng buộc thông thường (tự nhiên) của kiểu giới hạn $p_i^d \le p_i \le p_i^h$, trong đó p_i^d là giới hạn dưới và p_i^h là giới hạn trên của biến tối ưu thứ i. Việc đưa ra tiêu chuẩn thứ cấp dẫn đến sự định nghĩa các hàm giới hạn q_i của các biến tối ưu, ở đó chúng ta có $q_i^d \le q_i \le q_i^h$, trong đó q_i^d là giới hạn dưới và q_i^h là giới hạn trên của hàm liên quan.

Như vậy, từ quan điểm toán học, tối ưu tham số của hệ cơ điện được tính như bài toán tìm một điểm p trong tập P có thể chấp nhận được, mà ở đó hàm ưu tiên ψ tiến tới giá trị cực trị toàn cục (cực đại hoặc cực tiểu) liên quan đến P. Tập có thể chấp nhận được mô tả tổng quát bởi m bất đẳng thức được định nghĩa bởi hàm $q_j(p)$, trong đó $j=1,2,...m$. Nếu $P = R^s$, ở đây s là số biến số được tối ưu, chúng ta nói rằng tối ưu không điều kiện. Trong trường hợp khác chúng ta nói rằng tối ưu có điều kiện.

Để giải bài toán tối ưu các tính chất được chọn của một hệ thống, chúng ta làm như sau:

- Lập công thức mô tả toán học,
- Phân tích ở điểm bắt đầu,
- Xác định dạng mong muốn của hàm mục tiêu ψ,
- Lựa chọn các biến tối ưu,
- Xác định dạng mong muốn của hàm ràng buộc q_j,
- Chọn phương pháp tối ưu phù hợp,
- Giải bài toán tối ưu được thiết lập toán học và
- Sử dụng mô hình toán học, biến đổi ngược lại kết quả thành mô hình động lực học (chỉ với các bài toán động lực học).

Các khía cạnh chung của quá trình tối ưu

Nếu mục tiêu của quá trình tối ưu là tối ưu một số tính chất ảnh hưởng liên tục tới hệ thống (như tối thiểu các giá trị kích thước trong khi quan tâm tới các tính chất điện) thì chúng ta thu được một hàm mục tiêu đa tiêu chuẩn. Sau đó, hàm mục tiêu có dạng là tổng trọng số của các hàm tiêu chuẩn đơn. Mỗi hàm này thường nhận các cực tiểu địa phương của nó tại những điểm khác nhau của không gian tham số tối ưu. Đây là lý do tại sao hàm đa tiêu chuẩn có thể có số lượng lớn cực tiểu địa phương không sâu hoặc không nhạy với các thay đổi trong tham số tối ưu. Thực tế, việc

lựa chọn một phương pháp tối ưu thông số rất quan trọng. Kết quả được lấy trung bình trong phạm vi mà một vài tiêu chuẩn có thể tham gia đồng bộ trong dạng rút gọn của hàm đa tiêu chuẩn, trong khi một vài tiêu chuẩn khác có thể tăng.

Một phương pháp phù hợp hơn có thể là lựa chọn hàm mục tiêu tiêu chuẩn đơn, bao gồm tất cả tiêu chuẩn trong các ràng buộc. Chỉ có tiêu chuẩn quan trọng nhất được chọn cho hàm mục tiêu để định rõ trong quá trình sau đó. Tất cả tiêu chuẩn khác trong các ràng buộc được giữ trong giới hạn danh nghĩa mà không được tối ưu. Như vậy, kết quả của quá trình tối ưu là độc lập với bậc rút gọn của tập có thể chấp nhận bởi các ràng buộc kiểu bất đẳng thức.

Nói chung, chúng ta chỉ rõ các ràng buộc trong một dạng tương tự với hàm mục tiêu:

$$q_i(p) = f_i(p) - f_i^h, \ i = 1,2,...,m^* \tag{34.1}$$

Ở đây f_i là các hàm phù hợp của một biến vectơ và f_i^h là các giá trị có thể chấp nhận lớn nhất.

Sự lựa chọn các biến tối ưu nhận được cho bởi độ nhạy của hàm mục tiêu đối với các thay đổi của các biến tối ưu liên quan. Độ nhạy này được mô tả bởi vectơ gradien của hàm mục tiêu:

$$grad\,\psi(p) = \left[\frac{\delta\psi(p)}{\delta p_1},...,\frac{\delta\psi(p)}{\delta p_s}\right]^T \tag{34.2}$$

Các phương pháp tối ưu

Phương pháp tối ưu chuẩn

Hầu hết các bài toán thực tế dẫn đến các hệ phương trình phi tuyến (siêu việt). Chúng chỉ có thể được giải bằng các phương pháp tối ưu số. Tuỳ theo bậc của đạo hàm được sử dụng trong ứng dụng của phương pháp, mà ta có thể chia các phương pháp số tìm các cực tiểu cục bộ của hàm số nhiều biến thành:

1. Phương pháp bậc không (so sánh)
   - Phương pháp so sánh tọa độ
   - Phương pháp đơn hình
   - Phương pháp ngẫu nhiên
2. Phương pháp bậc nhất (gradien và gần gradien)
   - Phương pháp hướng liên kết
   - Phương pháp biến metric
3. Phương pháp bậc hai (phương pháp Newton)

Các phương pháp ngẫu nhiên

Các phương pháp này bao gồm tính toán giá trị của hàm mục tiêu tại số lượng lớn các điểm được chọn. Các điểm được chọn theo tiêu chuẩn mà mỗi điểm trong không gian có xác suất chọn bằng nhau. Sau đó, điểm tốt nhất được xác định bằng cách so sánh các giá trị hàm. Từ chiến lược được phác thảo, các phương pháp này sẽ dưa đến việc tính toán các giá trị hàm tại một số lượng lớn các điểm làm kéo dài thời gian tính toán. Mặt khác, chúng ta có thể dễ dàng tìm được tối ưu toàn cục của hàm được tối ưu. Các phương pháp này cũng so sánh với phương pháp tiến hóa khi mẫu kết quả đầu tiên được sinh ra hoàn toàn ngẫu nhiên. Chỉ khác nhau ở chiến lược chọn lời giải tốt hơn.

Các phương pháp tối ưu tiến hoá

Từ một số bài toán khó giải quyết bằng các phương pháp tối ưu số chuẩn, thậm chí nếu chúng hội tụ tới điểm tối ưu cho phép trong thời gian hợp lý, thì những nghiên cứu mới phải được phát triển. Một số phương pháp mới được thiết kế dựa trên các qui luật của di truyền tự nhiên sao chép

chúng ở các mức độ khác nhau. Các phương pháp này sử dụng sự tiến hoá ngẫu nhiên của tham số đầu vào và vì vậy có thể hiểu như phương pháp ngẫu nhiên. Nói chung, thuật toán tối ưu ngẫu nhiên (bao gồm hầu như tất cả các thuật toán tiến hóa) tối ưu bằng cách sử dụng hàm nhiều biến với hành vi "wild - chưa thuần" có nhiều cực tiểu hoặc một gradien chưa biết. Phương pháp tối ưu ngẫu nhiên tất nhiên là chậm hơn so với phương pháp kinh nghiệm có ưu điểm là biết kiểu hàm và mô tả của hàm được tối ưu. Trừ khi điều kiện cho tối ưu toàn cục được biết trước, chúng ta không bao giờ có thể đảm bảo rằng sẽ tiến tới tối ưu toàn cục hay không để kết thúc quá trình tối ưu. Tuy nhiên, phương pháp tối ưu ngẫu nhiên cũng mang đến nhiều thuận lợi. Chúng được xác định rất rõ ràng và vì vậy có thể áp dụng cho mọi bài toán và chúng có thể tránh được bẫy của cực tiểu cục bộ. Quá trình tiến hóa tìm kiếm không gian lời giải tiềm năng cần sự cân bằng của hai mục tiêu:

- Tìm cực tiểu gần nhất (hầu như là cục bộ) càng nhanh càng tốt
- Tìm không gian của tất cả các lời giải tiềm năng theo cách tối ưu.

Các phương pháp khác với định hướng đưa tới hai mục tiêu này và chúng có thể được đặt ban đầu trong một chuỗi bắt đầu từ các phương pháp hướng về cực tiểu cục bộ tới các phương pháp tìm kiếm số lượng lớn lời giải tiềm năng:

1. Thuật toán leo đồi ngẫu nhiên,
2. Thuật toán tìm kiếm cấm,
3. Thuật toán tôi luyện mô phỏng,
4. Giải thuật di truyền.

Thuật toán leo đồi

Đây là thuật toán tối ưu đơn giản nhất, là một dạng biến thể của phương pháp gradien "không gradien", ở đó hướng trèo dốc nhất được xác định bằng cách tìm lân cận. Thuật toán này cũng có tất cả những mặt hạn chế của phương pháp gradien, trong đó thuật toán gần như sẽ kết thúc ở một giới hạn cục bộ mà không tiến tới cực tiểu toàn cục. Ở đây, lời giải ban đầu được tạo ra một cách ngẫu nhiên. Với một lời giải được thiết kế hiện thời, một lân cận nào đó được sinh ra bằng cách dùng một tập biến đổi hữu hạn và cực tiểu tốt nhất được lựa chọn từ lân cận này. Sau đó, lời giải cục bộ thu được theo cách này được sử dụng như phần chính của lân cận mới mà trong đó lặp lại sự tối ưu. Quá trình này được lặp lại một số lần nhất định. Trong tiến trình này, các kết quả tốt nhất sau đó được ghi lại để sử dụng như là kết quả tìm được cuối cùng. Mặt hạn chế cơ bản của phương pháp này là, sau một số bước lặp, nó có thể trở lại cực tiểu cục bộ vừa trải qua trong bước trước (vấn đề vòng lặp). Vấn đề này có thể tránh được bằng cách thực hiện thuật toán nhiều lần với các giá trị ban đầu được tạo ngẫu nhiên khác nhau để lựa chọn kết quả tốt nhất.

Thuật toán tìm kiếm cấm

Cuối thập niên 1980, giáo sư Fred Glover đã thiết kế một cách tiếp cận mới để giải quyết bài toán tìm cực tiểu toàn cục, mà ông gọi là *tìm kiếm cấm*. Hiện nay, phương pháp này nằm trong số các phương pháp được sử dụng nhiều nhất để giải bài toán tổ hợp và bài toán tìm cực tiểu toàn cục. Dựa trên thuật toán leo đồi, nó cố gắng hạn chế vấn đề vòng lặp. Thuật toán leo đồi được trang bị một bộ nhớ tạm thời ghi nhớ các biến đổi ngược, đối với khoảng thời gian ngắn liền trước trong tiến trình thuật toán, thành các biến đổi giải pháp tối ưu địa phương được sử dụng để tìm tâm mới trong vòng lặp. Các biến đổi ngược này được ngăn chặn (cấm) khi lân cận mới được tạo ra với một kết quả hiện thời đã cho. Theo cách này, vòng lặp dẫn đến việc rơi vào bẫy cực tiểu cục bộ có thể giảm bớt. Thuật toán leo đồi được sửa đổi theo cách này sẽ tìm kiếm có hệ thống toàn bộ miền, trong đó có thể tìm ra cực tiểu toàn cục của hàm.

Thuật toán tôi luyện mô phỏng

Ngoại trừ các phương pháp ngẫu nhiên và các phương pháp dựa trên tiến hóa tự nhiên, có một khả năng khác trong việc mô phỏng sự tiến hóa của hệ thống dựa trên sự tiến hóa vật lý của hệ vĩ mô. Việc tôi luyện một vật rắn nhằm loại bỏ ứng suất bên trong là ví dụ đơn giản của kiểu tiến hóa

này. Để làm sáng tỏ về mặt vật lý của quá trình này, hãy xem một vật được nung nóng đến khi nó đạt tới nhiệt độ cao. Sau đó nhiệt độ giảm dần. Nguyên tử của vật được nung nóng tới nhiệt độ cao có thể dễ dàng vượt qua rào cản năng lượng cục bộ để tiến tới trạng thái cân bằng. Khi nhiệt độ thấp, nguyên tử được cố định trong trạng thái này và vật thể được làm lạnh sẽ không có ứng suất trong.

Nguyên tắc này được sử dụng để thiết kế phương pháp tôi luyện tái tạo. Trước tiên, ta đặt nhiệt độ ban đầu T_{max} giá trị của nó rất quan trọng để phương pháp có hiệu quả. Sau đó, thuật toán tôi luyện tái tạo sẽ tìm kiếm không gian của tất cả nghiệm tiềm năng bằng một cách thức ngẫu nhiên mạnh, kể cả việc chấp nhận các trạng thái tương ứng với kết quả kém hơn so với kết quả hiện thời. Tính chất của tôi luyện tái tạo là một đặc điểm đặc trưng của phương pháp này và cung cấp một cách để thoát khỏi bẫy của cực tiểu cục bộ, như vậy cho phép tìm những miền khác của không gian nghiệm. Tuy nhiên, khi nhiệt độ tôi luyện T thấp, khả năng chấp nhận trạng thái xấu cũng giảm bớt. Với giá trị nhiệt độ nhỏ, chỉ các nghiệm tốt hơn nghiệm hiện thời mới được quan tâm.

Thuật toán di truyền

Các thuật toán di truyền (Genetic Algorithm - GA) hầu hết được sử dụng để tối ưu hóa các tham số của hệ thống chưa biết có mô tả toán học quá phức tạp hoặc chưa biết [5]. Khi áp dụng một GA, hầu như nó có đủ khả năng để biết một hàm gán định giá cho mỗi cá thể trong tập mẫu. Điều này có thể là sai số so với kết quả của các thông số được lựa chọn ngẫu nhiên trong GA. Khi GA đang tìm một cực đại, thì ngược lại sai số đang được cực tiểu sẽ phải được biến đổi thành tìm cực đại. Điều này có thể được thực hiện theo nhiều cách khác nhau: bằng cách lấy sai số cực đại thu được trừ sai số đang xét, bằng cách tính toán giá trị nghịch đảo của sai số hoặc bằng cách sử dụng một hàm biến đổi khác tiến tới 0 khi sai số tiến tới 1. Nên tăng sự chú ý tới việc cài đặt chương trình thực thi hàm định giá do nó sử dụng thời gian tính toán nhiều nhất so với những thành phần GA khác.

Ngoài các bài toán tối ưu chung, GA còn được áp dụng cho hầu hết các mạng nơron. Ở đây xu hướng thực hiện GA ở hai mức khác nhau. Đầu tiên là tìm trọng số thích hợp cho mạng nơron và thứ hai, khi tối ưu hóa cấu trúc của mạng nơron, là lựa chọn thuật toán, số nơron đầu vào trong các lớp ẩn, số lớp ẩn, v.v... Sử dụng giải thuật di truyền để tối ưu các tham số của giải thuật di truyền khác (kích cỡ của mẫu, số lần sinh chéo, phạm vi đột biến, tần số đột biến) là một tư tưởng rất cách mạng (tối ưu thời gian tính toán, ở đó thời gian tính toán là hàm định giá của GA). Ngay khi áp dụng GA cho các bài toán gặp trong khi nghiên cứu về máy điện được quan tâm, GA đã được sử dụng để nhận dạng các tham số của sơ đồ thay thế cho động cơ cảm ứng.

Tóm lại, có thể nói thêm là các giải thuật di truyền thực thi tốt một cách ngạc nhiên trong khi tất cả các thuật toán khác thất bại, như các bài toán không đầy đủ khi thời gian tính toán là hàm mũ hoặc hàm giai thừa của số biến. Không có cách nào trong việc sử dụng GA để tối ưu các hàm đơn giản tương đối hoặc các hàm có thuật toán đặc biệt tồn tại để mô tả chúng. Khi xem xét sự cần thiết để tính toán các giá trị hàm cho hàng chục hoặc hàng trăm chuỗi di truyền trong một mẫu và sự cần thiết để đánh giá hàng trăm hoặc thậm chí hàng nghìn mẫu trong một lần chạy chương trình, ta sẽ thấy GA tiêu thụ nhiều thời gian hơn.

Mặc dù đạt được nhiều kết quả lạc quan bằng việc sử dụng GA, nhưng rõ ràng là tự nhiên cần sử dụng thậm chí sẽ phức tạp hơn và ở thời điểm này đây không phải là phương pháp quá phức tạp. GA mô tả ở trên chỉ tương ứng với ví dụ rất nguyên thủy được quan sát trong tự nhiên, đặc biệt chúng liên quan tới sự sinh sản vô tính với một nhiễm sắc thể đơn. Do tự nhiên đã phải mất hàng tỷ năm để kiểm tra thuật toán này nên sẽ hiệu quả cao nếu ta học nhiều hơn nữa về nó. Điều thú vị là không cần đến toán học để giải các bài toán tối ưu phức tạp. Tuy nhiên, có nhiều phương pháp tối ưu khác thích hợp để giải bài toán thiết kế [2-4].

Lựa chọn phương pháp tối ưu thích hợp

Phương pháp gradien chuẩn vẫn là một trong những phương pháp được sử dụng nhiều nhất. Các phương pháp gradien hoặc thậm chí các phương pháp phi gradien là không phù hợp nếu việc

tìm cực tiểu toàn cục cần một hàm với nhiều cực tiểu địa phương. Hầu hết những phương pháp này chỉ đạt tới điểm cực tiểu không quan trọng gần với điểm bắt đầu (lời giải ban đầu) ở đó chúng bị bẫy. Sự thiếu hụt này được khắc phục bằng việc lựa chọn lặp lại ngẫu nhiên lời giải ban đầu của bài toán tối ưu và lấy kết quả tốt nhất cho lời giải. Tính chất ngẫu nhiên của quá trình này có thể chỉ được thấy trong sự lựa chọn ngẫu nhiên của nghiệm ban đầu. Sau đó, thuật toán tối ưu tiếp tục mà không có bất kỳ sự ngẫu nhiên nào. Tiếp theo, các phương pháp tối ưu tiến hóa được xem như các phương pháp ngẫu nhiên bất chấp chúng thực hiện một chiến lược khi chọn các điểm thu được. Sau đây là những điểm khác nhau chính giữa giải thuật di truyền và phương pháp gradien được sử dụng thường xuyên hơn:

- GA thực hiện không có sự tính toán gradien để có thể gây khó khăn và tốn thời gian đặc biệt đối với hệ thống lớn
- GA làm việc với các lời giải được tạo ra ngẫu nhiên và có thể hội tụ nhanh hơn đến cực tiểu toàn cục.

Để tối ưu thiết kế ban đầu của một động cơ cảm ứng, người ta thực hiện một phương pháp tối ưu sử dụng giải thuật di truyền. Phương pháp này được mô tả cụ thể hơn trong phần sau.

34.3 Thiết kế tối ưu động cơ cảm ứng

Giới thiệu thiết kế động cơ cảm ứng (Induction Motor – IM)

Thiết kế thực của động cơ cảm ứng thường phụ thuộc vào các yêu cầu của từng khách hàng xác định cụ thể các thông số được một máy thiết kế thực hiện. Theo cách này, với cùng đầu ra của máy, chúng ta thu được nhiều cách thực hiện khác nhau khi gặp các điều kiện cụ thể tốt hơn hoặc kém hơn. Chỉ có thể lấy được chất lượng tốt cho một thông số khi làm giảm giá trị của những thông số khác. Chúng ta sẽ thực hiện thiết kế động cơ có đầu ra trong khoảng (0,6-200) kW. Động cơ được thiết kế cho tải dài hạn và với nhiệm vụ của dự án này thì cần các giá trị đầu vào sau đây:

Đầu ra của máy P_n [kW], điện áp U_{1n} [V], nối cuộn dây *Y/D*, số cực 2**p** hoặc tốc độ quay n [min^{-1}], tần số lưới f [Hz], hiệu suất η [%], hệ số công suất $\cos\varphi$, lớp cách nhiệt, thực thi IP và hình dạng của máy.

Chúng ta xem xét động cơ lồng sóc vòng kín với khung và các gân làm mát. Đó là lồng rôto bằng nhôm đúc. Để thiết kế, cần dữ liệu như dây dẫn và kích thước rãnh hoặc đặc tính từ suy ra từ bảng và đồ thị được cho bởi chuẩn hoặc đo lường của nhà sản xuất. Thiết kế thực là sự kết hợp giữa những thông số thiết kế riêng, để máy thu được có thể có các đặc tính hoạt động tốt nhất với nhiệt độ của máy thấp nhất và tiết kiệm nguyên liệu. Thiết kế động cơ thực được mô tả trong phần sau.

Đánh giá thiết kế IM cổ điển

Một thiết kế động cơ cảm ứng, khi thực hiện bằng tay, thể hiện hàng trăm phép tính có thể tiêu tốn hàng chục giờ thậm chí với nhà thiết kế có kinh nghiệm. Khi máy tính thực hiện phương thức của họ cho tất cả các nhánh phân tích và thiết kế trên thực tế, thì người ta đã tạo ra một chuỗi chương trình hợp tác với một nhà thiết kế trong một kiểu tương tác và tăng tốc độ tính toán.

Mặc dù có nhiều ưu điểm rõ rệt trong quá trình thiết kế này, nhưng chúng ta phải thấy rằng có một số lượng đáng kể cách thực hiện thiết kế cho một động cơ cho trước nhằm thu được nhiều hơn hay ít hơn các đặc tính hoạt động theo yêu cầu của động cơ. Điều này gần giống với cực tiểu toàn cục của hàm mục tiêu để đánh giá chất lượng thiết kế.

Do đó, cần phát triển ý tưởng để tạo ra một chương trình tìm kiếm toàn không gian trạng thái của tất cả các lời giải có thể và lựa chọn phương án phù hợp nhất cho hàm mục tiêu định giá (các đặc tính động cơ yêu cầu) bằng cách sử dụng một số phương pháp tối ưu. Phương pháp tiến hóa ngẫu nhiên và giải thuật di truyền được chọn, do nó tìm toàn bộ không gian trạng thái của tất cả các lời giải có thể theo cách tốt nhất.

BẢNG 34.1 Danh sách các tham số được tạo ra và thiết lập các giới hạn của chúng

Tên tham số	Ký hiệu	Đơn vị	Giới hạn trên	Giới hạn dưới
Đường kính ngoài của Stator	D_e	mm	Người dùng chọn	Người dùng chọn
Đường kính trong của Stator	D	mm	Người dùng chọn	Người dùng chọn
Độ dài sắt lí tưởng	l_i	mm	Người dùng chọn	Người dùng chọn
Cảm ứng khe hở không khí	Bδ	T	0.5	1.0
Độ điền đầy khe Stator	k_{dr1}	—	0.6	0.75(0.8)
Kích cỡ khe hở không khí	δ	mm	0.2	0.4
Mật độ dòng Stator	σ_1	A mm^{-2}	3.0	15.0
Mật độ dòng thanh Rôto	σ_t	A mm^{-2}	2.0	6.0
Mật độ dòng vòng Rôto	σ_k	A mm^{-2}	2.0	4.0
Cảm ứng từ của chân	B_Z	T	1.6	2.0
Số rãnh cho mỗi cực và pha Stator	q_1	—	2.0	5.0

Mô tả bài toán được giải

Các thông số được tạo ra

Các giá trị cho trong bảng 34.1 chỉ để giới thiệu đối với động cơ có công suất dưới 200 kW và chúng là những giá trị có giới hạn. Việc biến đổi các giá trị của thông số hoặc thay thế một thông số cho một thông số khác có thể chỉ bằng cách can thiệp vào chương trình nguồn. Các giới hạn đường kính D_e và D từ tệp đầu vào chỉ được xem xét trong trường hợp cần một thiết kế động cơ mà không quan tâm đến chiều cao trục tiêu chuẩn. Trong trường hợp mà chiều cao trục tiêu chuẩn được đưa vào, người ta sẽ tính toán giới hạn này. Giới hạn của chiều dài rôto lý tưởng thích hợp để đưa vào càng hẹp càng tốt nhằm hội tụ nhanh hơn tới giới hạn. Nhưng đây không phải là điều kiện cần thiết. Nói chung, dải thông số riêng càng thấp thì hội tụ tới cực tiểu toàn cục càng nhanh và số lượng cực tiểu cục bộ càng ít.

Hàm mục tiêu

Nó không chỉ là dạng của hàm mục tiêu, mà còn là việc lựa chọn các tham số tối ưu quan trọng cho các kết quả tối ưu tốt. Các tham số được lựa chọn phải mô tả đầy đủ một giải pháp chất lượng cho vấn đề đã cho. Trong trường hợp tối ưu hóa thiết kế động cơ cảm ứng, những thông số sau được lựa chọn:

- Thể tích động cơ V [dm3]
- Tăng trưởng nhiệt độ của động cơ ϑ_n [K]
- Hệ số công suất danh nghĩa của động cơ cos i 'n [–]
- Hiệu suất danh nghĩa của động cơ η_n [–]
- Khả năng chịu tải của mômen m_{pm} [–]

Đây là những tham số quan trọng nhất cho chất lượng thiết kế và mô tả thiết kế một cách đầy đủ.

Sai số tổng dựa trên quan hệ nhận được từ phương trình (34.1) là tổng sai số riêng của mỗi tham số điều khiển. Nếu chúng ta tập trung hơn vào một số tham số thì phải tăng trọng số tương ứng, như vậy, sẽ đạt được sự cải thiện trong thiết kế cuối cùng. Tại cùng thời điểm đó, giá trị của những tham số khác sẽ giảm. Tìm một cài đặt tối ưu của hệ số khuếch đại là một trong những vấn

để quan trọng và khó khăn nhất. Thuật ngữ "cài đặt tối ưu" có nghĩa là động cơ được thiết kế có hệ số công suất cao nhất, hiệu suất, khả năng chịu tải của mômen tại thể tích nhỏ nhất và đồng thời không vượt quá tăng trưởng nhiệt độ cho phép với một lớp cách nhiệt được chọn. Chúng ta có mối liên hệ như sau:

$$\begin{aligned}\varepsilon(GR_i) = abs(KV.V) + abs(k\vartheta(0.89\vartheta_{cl} - \vartheta_n)) + abs(k_{\cos\varphi}(1-\cos\varphi_n)) \\ + abs(k\eta(1-\eta_n) + abs(k_{mp}(m_p + 1 - m_{pn}))\end{aligned} \tag{34.1}$$

BẢNG 34.2 Các giá trị đầu vào của động cơ 5.5 kW, 380 V

Tên các đại lượng	Ký hiệu	Đơn vị	Giá trị
Công suất đầu ra danh nghĩa	P_n	W	5500
Điện áp danh nghĩa	U_{ln}	V	380
Hệ số công suất yêu cầu	*Cosj*	—	0.81
Hiệu suất yêu cầu	*H*	—	0.86
Tần số lưới	*F*	Hz	50
Chiều cao của trục động cơ	*H*	mm	132
Số đôi cực	*P*	—	3
Lớp cách ly nhiệt độ	*TT*	—	F
Khả năng chịu quá tải của mômen	m_p	—	2

ở đây, kV là trọng số thể tích, $k\vartheta_n$ là trọng số nhiệt độ, $k\cos\varphi_n$ là trọng số công suất, $k\eta$ là trọng số hiệu suất, k_{mp} là trọng số khả năng chịu tải của mômen.

Kết quả đạt được

Mô tả thiết kế động cơ 5,5 kW, 380 V

Khi phát triển và điều chỉnh chương trình, tối ưu đã được thực hiện trên động cơ mô tả trong bảng 34.2. Trong phần này chúng ta mô tả kết quả và các vấn đề gặp phải trong quá trình tối ưu. Tham số đầu vào động cơ được cho trong bảng 34.2.

Những kết quả khác

Nó bắt đầu từ nguyên lý vật lý liên hệ chặt chẽ các đại lượng tối ưu. Việc tăng hệ số khuếch đại của một đại lượng dẫn đến sự bất lợi của các đại lượng. Dựa trên sự tối ưu đã được thực hiện, có thể kết luận rằng tồn tại hai kiểu động cơ, phụ thuộc hàm lượng của thép và đồng:

1. Động cơ với hàm lượng thép phổ biến, mật độ dòng stator cao, hệ số công suất tốt, cho một giá trị hiệu suất động cơ xấu và với khả năng chịu quá tải của mômen kém hơn động cơ thứ hai.
2. Động cơ với hàm lượng đồng cao và ngược lại, mật độ dòng stator thấp, hiệu suất tốt, giá trị hệ số công suất kém. Khả năng chịu quá tải của mômen tốt.

Kiểu động cơ được xác định dựa trên thiết lập các hệ số khuếch đại. Tổng sai số hệ số nhiệt độ và công suất trên một mặt tác động tới tổng sai số thể tích và hiệu suất trên mặt khác. Khả năng chịu quá tải của mômen có thể là tốt cho cả hai loại loại động cơ.

Kết quả tối ưu cụ thể được liệt kê trong bảng 34.3, sắp xếp theo giá trị thể tích từ nhỏ nhất đến lớn nhất. Những loại động cơ khác nhau đã được sử dụng, phụ thuộc vào giá trị hệ số khuếch đại. Khó có thể xác định giải pháp nào tốt hoặc xấu, bởi vì sự lựa chọn phụ thuộc vào yêu cầu khách hàng. Một số kết quả cho giá trị tối ưu tốt nhất được quan tâm đặc biệt. Các kết quả số 1, 5, 8, 23 và 25 có thể được xem như thành công từ phương pháp này. Động cơ được mô tả ở trên (giải pháp số 2)

phục vụ cho việc mô tả nhiệm vụ. Các quan hệ được quan tâm trước đây giữa các đại lượng riêng lẻ có thể được thấy trong bảng 34.3. Chúng liệt kê kết quả tối ưu không giới hạn thông số khởi tạo, do đó sử dụng các yêu cầu trong bảng 34.1.

BẢNG 34.3 Danh sách các giải pháp cho động cơ có P=5.5kW, U=380 với các tham số khởi tạo không giới hạn

STT	$V[\mathrm{dm}^3]$	$\vartheta[\mathrm{K}]$	$\cos\varphi[-]$	$\eta[-]$	$m_p[-]$	Danh mục động cơ
1	**3.96**	88.1	0.798	0.834	1.72	Động cơ 1
2	4.20	86.9	0.818	0.843	1.90	Động cơ 2
3	4.31	74.9	0.787	0.865	1.77	Động cơ 3
4	4.32	88.8	0.836	0.817	1.78	Động cơ 4
5	4.33	75.1	0.690	**0.973**	1.07	Động cơ 5
6	4.5	89.0	0.836	0.834	1.79	Động cơ 6
7	4.51	86.8	0.818	0.818	1.93	Động cơ 7
8	4.54	**90.0**	0.884	0.812	1.98	Động cơ 8
9	4.56	84.6	0.857	0.816	1.74	Động cơ 9
10	4.58	86.5	0.836	0.817	1.77	Động cơ 10
11	4.63	68.2	0.792	0.858	2.10	Động cơ 11
12	4.69	88.4	0.862	0.808	1.80	Động cơ 12
13	4.70	73.4	0.845	0.830	2.25	Động cơ 13
14	4.73	61.0	0.499	0.871	1.90	Động cơ 14
15	4.78	78.1	0.853	0.858	1.67	Động cơ 15
16	4.78	71.0	0.767	0.870	1.80	Động cơ 16
17	4.81	70.6	0.703	0.934	1.28	Động cơ 17
18	4.97	54.5	0.804	0.883	1.90	Động cơ 18
19	5.08	55.5	0.762	0.877	2.20	Động cơ 19
20	5.12	88.2	0.879	0.806	2.05	Động cơ 20
21	8.96	44.0	0.784	0.870	2.55	Động cơ 21
22	6.35	42.4	0.803	0.882	2.69	Động cơ 22
23	6.40	87.5	**0.887**	0.853	2.27	Động cơ 23
24	6.57	59.0	0.747	0.956	1.16	Động cơ 24
25	7.05	42.3	0.793	0.865	**3.00**	Động cơ 25
26	7.39	52.9	0.714	0.986	1.03	Động cơ 26

Tiếp theo, một sự tối ưu động cơ được thực hiện chỉ với một tham số tối ưu, khi phần khuếch đại của những tham số khác được đặt bằng không.

1. *Tối ưu thể tích*: Trường hợp này, thuật toán lựa chọn động cơ có giải pháp tốt nhất với kích thước nhỏ nhất, khi các tham số D_e, D và l, hội tụ tới giá trị cực tiểu hiện thời.
2. *Tối ưu nhiệt độ*: Thuật toán đạt tới cực tiểu cục bộ trước tiên với nhiệt độ cực đại dựa trên lớp cách nhiệt yêu cầu và hầu như giữ ổn định ở giá trị này.
3. *Tối ưu hệ số công suất*: Một sự nỗ lực để đạt được kiểu động cơ thứ nhất (xem trình bày ở trên) với hàm lượng đồng thấp, mật độ dòng điện cao σ_1, giá trị hiệu suất thấp. Khả năng chịu quá tải của mômen tốt.
4. *Tối ưu hiệu suất*: Động cơ được thiết kế tương ứng với kiểu động cơ thứ hai (xem trình bày ở trên) với thành phần đồng phổ biến, mật độ dòng thấp, hiệu suất cao, tuy nhiên giá trị hệ số công suất thấp. Khả năng chịu quá tải của mômen tốt.

5. *Tối ưu khả năng chịu quá tải của mômen*: Động cơ được thiết kế có số rãnh cao đối với cực và pha, dẫn đến việc dây dẫn phải phân bố rộng theo chu vi. Động cơ có thể có hàm lượng đồng hoặc sắt phổ biến phụ thuộc vào giải pháp cục bộ, để nó hội tụ. Nó có thể có giá trị hệ số công suất và hiệu suất tốt khi giá trị của thể tích máy tăng lên.

34.4 Sử dụng mạng nơron để nhận dạng tham số của một hệ động lực học

Bước cơ bản được sử dụng để giải quyết các nhiệm vụ động lực học bằng bất kỳ loại phương tiện nào là tạo ra một tập các đại lượng quan trọng bao gồm các đại lượng mô tả cấu trúc, điều kiện, sự tương tác của các đối tượng kỹ thuật và các đại lượng thể hiện kết quả.

Các phương pháp để tạo mô hình toán học trong hệ truyền động, xa hơn là quá trình tương tác, sử dụng:

- Các ứng dụng của những định luật vật lý nổi tiếng để mô tả các hiện tượng trong hệ thống truyền động (như định luật Newton thứ hai, định luật Kirchhoff...)
- Các ứng dụng của những phương pháp dựa trên thuật toán trí tuệ nhân tạo (như giải thuật gen [1] và mạng nơron nhân tạo [6,7]).

Theo như lý thuyết đã biết ở phần trước, các phương pháp trí tuệ nhân tạo thay thế phương pháp số và giải tích "chuẩn" khi:

- Chỉ có các lý thuyết này mới giải quyết được vấn đề,
- Chúng thể hiện những tính chất tốt hơn từ quan điểm giải quyết vấn đề (chẳng hạn, một điều kiện tốt hơn khi xem xét các thay đổi của giá trị đầu vào),
- Chúng cho phép các vấn đề được giải quyết có hiệu quả hơn.

Trường hợp cuối cùng là điển hình khi chúng ta cần tiếp cận càng gần càng tốt tới các điều kiện hoạt động thực. Từ những phương pháp trí tuệ nhân tạo khác nhau, thuật toán tiến hóa ngẫu nhiên và mạng nơron nhân tạo đang được sử dụng tăng dần trong lĩnh vực mô hình hóa hệ thống tương tác truyền động. Trong phần tiếp theo, hai phương pháp trên được áp dụng cho bài toán phân tích tính chất động lực học của hệ truyền động.

Với thuật toán trí tuệ nhân tạo, lời giải động lực học chính là lời giải của các vấn đề sau:

- Định rõ một tập các đại lượng quan trọng (có liên quan),
- Lựa chọn lý thuyết phù hợp để giải quyết bài toán,
- Sắp xếp các quan hệ giữa những đại lượng liên quan để chúng cho phép sử dụng thuật toán trí tuệ nhân tạo đã chọn,
- Tạo ra dữ liệu huấn luyện và lựa chọn phương pháp dạy học, ví dụ trong các mạng nơron,
- Kiểm tra số lượng kết quả đạt được và sự tiến hóa của chúng.

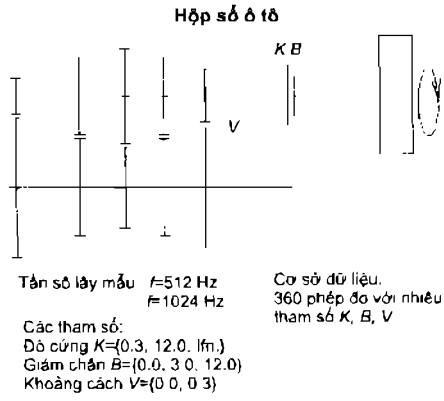
Ứng dụng thực tế

Nhiều phương pháp nhận dạng được biết và được thực hiện thường xuyên đem lại kết quả tốt trong các điều kiện thực tế. Tuy nhiên, các nhân tố giới hạn làm cho các phương pháp này trở nên khó khăn hơn (như các giả thuyết về tính tuyến tính của hệ thống, tính ổn định và tính chuẩn hóa của hiện tượng xảy ra trong hệ thống...). Do vậy, chúng ta sử dụng cách tiếp cận phi truyền thống tới bài toán *ƒ* nhận dạng các tính chất động lực học trong hệ cơ học, trong đó việc sử dụng hệ nơron dường như đầy hứa hẹn và ở thời điểm này có thể sẵn sàng được sử dụng trong tư duy kỹ thuật.

Một ứng dụng thực tế - Hộp số

Trường hợp đầu tiên được phân tích là hộp số của ô tô. Đầu vào (tải động cơ), đầu ra (phổ biên độ-tần số của dao động xoắn) và cấu trúc hộp số đã được biết. Thông tin này liên quan tới thông số

đã chọn về độ cứng và giảm chấn trong truyền động. Do điều kiện hoạt động thay đổi, biên độ giảm chấn có thể biến đổi đủ để ảnh hưởng tích cực đến đặc tính đầu ra. Nếu nhiều kết quả thực nghiệm được đánh giá, một số sai sót tiêu biểu có thể bị nhận dạng và sự cố có thể có của chúng được biết trước từ các tệp đầu ra.



HÌNH 34.1 Bánh răng năm cấp tốc độ

BẢNG 34.4 Tần số tự nhiên và kích thích mong muốn cho hộp số

Tần số mong muốn		[Hz]
Tần số thấp		5.0
Tần số hoạt động (OF) (tốc độ)	OF	14.16
	2 x OF	28.32
	3 x OF	42.48
Tần số điều hoà (IHF)	0.5 x OF	7.08
	1.5 x OF	21.25
	2.5 x OF	35.42
Tần số tự nhiên (NF)	I.NF	43.91
	0.5 x I.NF	21.96
	2 x I.NF	87.92
	II.NF	322.1
	0.5 x II.NF	161.1
	2.0 x II.NF	644.2
Tần số kết hợp (CF)	2 x OF + 0.5 x I.NF	50.28
	OF + I.NF	58.07
	2 x OF + I.NF	72.23
	2 x OF + 2 x I.NF	116.1
Tần số răng cưa (TF) {TF=z. OF}	Bánh răng tốc độ 1	184.1
	Bánh răng tốc độ 2	325.7
	Bánh răng tốc độ 3	424.9
	Bánh răng tốc độ 4	580.6

Chúng ta sử dụng dữ liệu được đo lường trong hệ thống thực. Phổ biên độ-tần số của dao động xoắn được đo ở trục hộp số đặt trên bốn ổ bi với năm bánh răng tốc độ (hình 34.1). Ban đầu, đo lường được thực hiện để xác định cộng hưởng của hệ thống tần số với mục đích giảm tiếng ồn. Những thông số sau đây được thiết lập trong hệ thống:

- Độ cứng $K \in \{0,3;12,0;\infty\}$,

- Giảm chấn $B \subset \{=:0:0.3:12.0\}$.

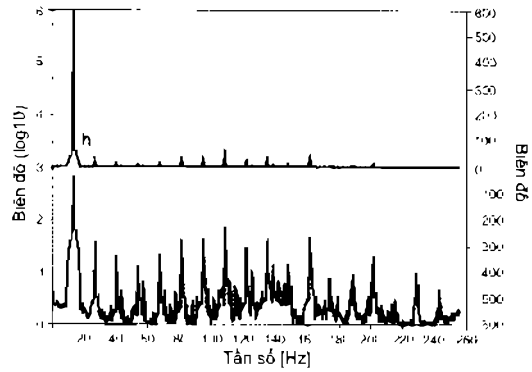
Các đo lường được thực hiện với tần số kiểm tra f = 512 Hz và f = 1024 Hz. Để ghi lại những điều hòa dao động quan trọng, tần số kích thích được linh hoạt trong cả hai trường hợp:

1. Từ 2,5 tới 14,0 Hz với bước 0,5 Hz và
2. Từ 14,0 tới 40,0 Hz với bước 1,0 Hz.

Phổ này luôn bao gồm 512 đường phổ. Đo lường được lặp lại 360 lần đối với các tham số khác nhau K và B (điều chỉnh hệ thống). Giá trị của tần số tự nhiên và kích thích mong muốn nhằm phân tích hộp số được thấy trong bảng 34.4.

Xác định nhiệm vụ

Nhiệm vụ ban đầu được xác định theo cách sau đây: để đánh giá biên độ của các tham số K và B tương ứng (điều này có nghĩa là, để nhận dạng sự điều chỉnh tham số được sử dụng trong hệ thống động lực học) bằng phương pháp mạng nơron nhân tạo trên cơ sở phổ biên độ-tần số của dao động trục xoắn trong hệ thống đã cho. Mạng perceptron đa lớp với ba lớp (tức là một lớp đầu vào, một lớp ẩn và một lớp đầu ra) được sử dụng như cấu hình của mạng nơron. Lựa chọn các hàm nơron tín hiệu như lớp tuyến tính, lớp ẩn và lớp đầu ra của hàm logistic, $f(x) = 1/(1+e^{-x})$



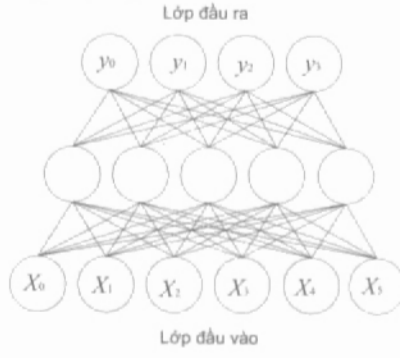
HÌNH 34.2 Chuẩn hóa vectơ kích thích (phần trên - trước khi chuẩn hóa; phần dưới - sau khi chuẩn hóa)

Theo định nghĩa này, đây là nhận dạng tham số hệ thống trên cơ sở phổ biên độ-tần số đo được. Tuy nhiên, tham số được lấy từ tập rời rạc (và rất thấp) và nhiệm vụ có thể được xác định lại như nhiệm vụ "chuẩn" của sự phân loại phổ theo bảy thuộc tính (mỗi thuộc tính tương đương với một giá trị có thể của thông số K và B). Áp dụng mạng nơron để giải quyết bài toán như thế thành công hơn khi so sánh với giải pháp ban đầu.

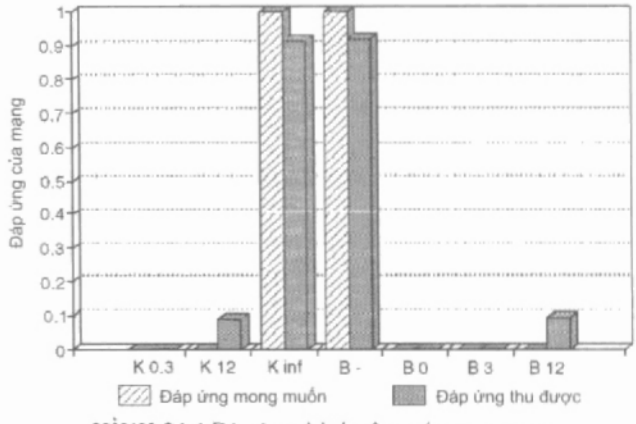
Biên độ của đường phổ được biểu diễn theo tỷ lệ lôgarit và sự suy giảm động lực học phổ với việc tăng chất lượng thông tin đã đạt được. Xét bản chất phi tuyến của hàm nơron kích hoạt được dùng, có sự mở rộng vượt quá dải tập trung đối với khoảng đầu vào (0,5; 0,95), **mạng không thể đáp ứng tốt đối với vectơ kích thích có dải giá trị** cao trong các thành phần riêng biệt. Điều này được minh họa trên hình 34.2. Lớp mạng đầu vào được cấu hình tới 512 nơron. Giá trị lôgarit biên độ của một đường phổ được đưa vào mỗi đầu vào. Những nơron riêng biệt ở lớp đầu ra tương ứng với thuộc tính phân loại. Do có bảy thuộc tính, nên bảy nơron được định cấu hình ở lớp đầu ra.

Chỉ một lớp ẩn được đặt như trung bình cộng của số nơron đầu vào và đầu ra. Hai trăm sáu mươi nơron được định cấu hình cho lớp ẩn duy nhất, như minh họa trên hình 34.3. Mỗi một phần tử tương ứng với một đo lường phổ tần số (vectơ kích thích) với vectơ thuộc tính tương ứng (vectơ đáp ứng yêu cầu). Biến đổi đặc điểm của tham số được biểu diễn bởi đáp ứng mạng yêu cầu tới hai nơron đầu ra tương ứng bằng 1 và nơron đầu ra còn lại bằng 0.

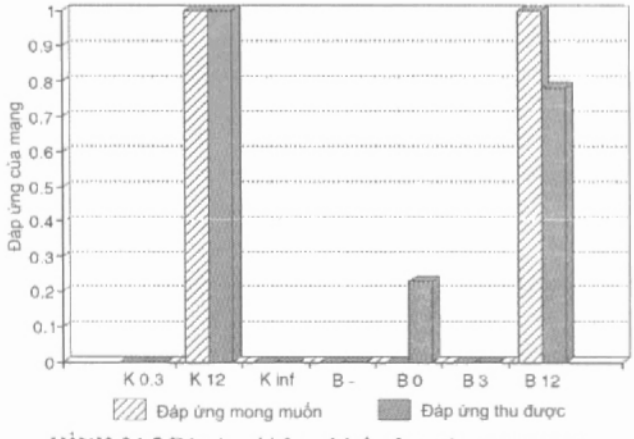
Từ 360 phần tử ban đầu, 36 phần tử được ngẫu nhiên lựa chọn (10% của tổng) cho việc kiểm tra sau này. Chúng ta đảm bảo rằng kiểm tra mạng có thể được tiến hành với các phần tử không vượt qua qui trình mạng huấn luyện (mạng không được huấn luyện trong tình huống này). Điều này cần thiết để kiểm tra các tính chất mô hình tổng quát. Một tập huấn luyện được thành lập bởi 324 phần tử còn lại. Chiến lược dạy học theo trình tự được sử dụng, tức là, những phần tử từ tập huấn luyện được sử dụng trong quá trình dạy học với trình tự cố định (những đoạn chu kỳ thông qua tập huấn luyện). Quan tâm tới kích thước của mạng nơron đã được định cấu hình, phương pháp truyền mômen phản hồi được lựa chọn như là phương pháp dạy học chỉ lấy thông tin kể cả từ bậc nhất (giá trị của hàm đặc biệt-người dạy và gradient của nó) và không sử dụng Hessian hoặc ước lượng của nó, mà trong trường hợp này có thể rất khắt khe.



HÌNH 34.3 Cấu hình mạng (lớp ẩn bao gồm các nơron không được ký hiệu)



HÌNH 34.4 Đáp ứng thành công của mạng nơron



HÌNH 34.5 Đáp ứng không thành công của mạng nơron

Kết quả

Mô hình nơron của hệ cơ học đã chứng minh tỷ lệ thành công cao trong khi thực hiện bởi các tập hợp kiểm tra. Mạng đã được dạy học với sự lựa chọn ngẫu nhiên các phần tử kiểm tra. Dưới sự kiểm tra của mô hình cụ thể, đáp ứng của mạng đã thành công 85-95% trong tất cả các trường hợp (xem hình 34.4 và hình 34.5). Hơn nữa, việc đánh giá các giá trị K và B tương ứng với các tham số có thể trong hai giây đối với phổ tần số trong chế độ hoạt động. Tuy nhiên, có thể là mô hình với chất lượng cao sẽ đạt được nếu kỹ thuật tối ưu đặc biệt được sử dụng trong tương lai.

Tóm lại, mô hình nơron của hệ cơ học được mô tả ở trên có thể được sử dụng trong điều kiện thực tế.

Tài liệu tham khảo

[1] Goldberg, D., *Genetic Algorithms in Searching, Optimisation and Machine Learning, Reading*, MA: Addison-Wesley, 1989.

[2] Glover, F., Lagunai, M., Marti, R., *Fundamentals of scatter search and path relinking, Control and Cybernetics*, pp. 653–684, 2000.

[3] Glover, F., *Scatter Search and Star-Paths—Beyond the Genetic Metaphor*, pp. 125–137, New York: Springer-Verlag, September 1995.

[4] Glover, F., Kelly, J.P., *Langunai, M., Genetic algorithm and tabu search—hybrids for optimization, Computers and Operations Research*, pp. 111–134, January 1995.

[5] Lee, J., Hajela, P., *Parallel genetic algorithm implementation in multidisciplinary rotor blade design*, Journal of Aircraft, Vol. 33, No.5, pp. 962–969, September–October 1996.

[6] Hagan, M.T., Demuth, H., Beale, M., *Neural Network Design*, Boston: PWS Publishing, 1996.

[7] Kosko, B., *Neural Networks and Fuzzy Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1992.

[8] Ye, X., Loh, N., Dynamic system identification using recurrent radial basis function network, Neural Networks Theory, Technology, and Applications, New York: IEEE Technology Update Series, 1996.

V

Máy tính và hệ thống logic

35 Giới thiệu về máy tính và hệ logic *Kevin Craig và Fred Stolfi* 35-1

Giới thiệu vai trò của máy tính trong cơ điện tử • Mô hình và mô phỏng Cơ điện tử trên máy tính • Các hệ thống đo lường, máy tính và cơ điện tử • Cơ điện tử và ứng dụng tính năng thời gian thực của máy tính • Tính tổng hợp của Mechatronics

36 Các khái niệm logic và thiết kế logic *George I. Cohn* 36-1

Giới thiệu • Biểu diễn thông tin dạng số • Các hệ thống số • Biểu diễn số • Số học • Chuyển đổi cơ số • Phần bù • Mã • Đại số Boolean • Các hàm • Boolean • Các mạch cổng • Các dạng khai triển • Thực hiện • Các lược đồ thời gian • Rủi ro • Các dạng giản đồ K • Giản đồ K và tối giản hóa • Tối thiểu hóa với giản đồ K • Tối giản hóa theo bảng Quine-McCluskey

37 Giao diện hệ thống *M.J. Tordon và J. Katupitiya* 37-1

Các khái niệm cơ bản • Các chuẩn giao diện nối tiếp TIA/EIA • IEEE 488-Bus giao diện đa năng (GPIB)

38 Truyền thông và mạng máy tính *Mohammad Ilyas* 38-1

Lịch sử tóm tắt • Giới thiệu • Mạng máy tính • Kỹ thuật cấp phát tài nguyên • Các thách thức và các vấn đề • Tóm tắt và kết luận

39 Phân tích lỗi trong các hệ cơ điện tử *Leila Notash và Thomas N. Moore* 39-1

Giới thiệu • Công cụ phân tích hỏng hóc/độ tin cậy • Phân tích hỏng hóc trong hệ Cơ điện tử • Các phương pháp phát hiện lỗi thông minh • Những vấn đề về phát hiện lỗi thông minh • Ví dụ về hệ Cơ điện tử: Tay máy/Máy công cụ song song • Kết luận và nhận xét

40 Thiết kế hệ logic *M. K. Ramasubramanian* 40-1

Giới thiệu lôgic số • Thiết bị bán dẫn • Các cổng lôgic • Thiết kế lôgic • Công nghệ cổng lôgic • Mạch tích hợp cổng lôgic • Thiết bị lôgic khả trình (PLD) • Ví dụ áp dụng Cơ điện tử

41 Các hệ thống dãy đồng bộ và không đồng bộ *Sami A. Al-Arian* 41-1

Tổng quan và định nghĩa • Tổng hợp hệ thống nối tiếp đồng bộ • Tổng hợp hệ thống nối tiếp không đồng bộ • Thiết kế mạch điều khiển và đường dữ liệu • Kết luận

42 Kiến trúc *Daniel A. Connors và Wen-mei W. Hwu* 42-1

Giới thiệu • Các loại vi xử lý • Các bộ phận chính của vi xử lý • Cấu trúc tập lệnh • Lệnh song song • Những xu hướng trong công nghiệp

43 Điều khiển với máy tính nhúng và bộ điều khiển logic khả trình

Job van Amerongen 43-1

Giới thiệu • Máy tính nhúng • Bộ điều khiển logic khả trình • Kết luận

35

Giới thiệu về máy tính và hệ thống lôgic

Kevin Craig
Rennselear Polytechnic Institute
Fred Stolft
Rennselear Polytechnic Institute

35.1 Giới thiệu vai trò của máy tính trong cơ điện tử 35-1
35.2 Mô hình và mô phỏng Cơ điện tử trên máy tính 35-3
35.3 Các hệ thống đo lường, máy tính và cơ điện tử.....35-4
35.4 Cơ điện tử và ứng dụng tính năng thời gian thực của máy tính.....35-5
35.5 Tính tổng hợp của Mechatronics.....35-10

35.1 Giới thiệu vai trò của máy tính trong cơ điện tử

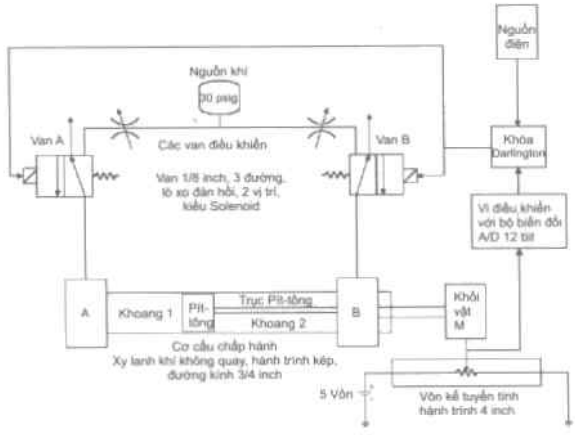
Cơ điện tử là sự kết hợp liên ngành giữa kỹ thuật cơ khí, điện tử, điều khiển, và máy tính. Đặc điểm chính của cơ điện tử chính là sự tích hợp các ngành trên trong một quy trình thiết kế. Sự bổ trợ và tích hợp trong thiết kế làm cơ điện tử tách hẳn ra khỏi các hệ thống mang tính truyền thống. Trong một hệ cơ điện tử thì máy tính, điện tử và công nghệ điều khiển đã làm thay đổi tư duy thiết kế, dẫn đến khả năng hoạt động của hệ thống được nâng cao với giá thành thấp hơn. Điều này có được là nhờ sự kế thừa tính chính xác và đáp ứng kịp thời của điều khiển, hiệu quả và tin cậy của thiết bị điện tử, sự chặt chẽ và linh hoạt của máy tính. Một ví dụ cụ thể là các hệ thống điều khiển động cơ ô tô. Ở đó có rất nhiều cảm biến dùng để đo thông tin về sự thay đổi của nhiệt độ, áp suất, tốc độ dòng chảy, tốc độ quay và sự tổng hợp các hoá chất. Các thông tin trên được gửi đến một vi máy tính. Máy tính sẽ tổng hợp các thông tin qua một mô hình động cơ được lập trình sẵn cùng các luật điều khiển và sau đó gửi lệnh điều khiển các van, cơ cấu chấp hành, vòi phun nhiên liệu và hệ thống đánh lửa sao cho động cơ đạt được hiệu quả tối ưu về gia tốc, nhiên liệu và lượng khí thải.

Trong cơ điện tử sự cân bằng mang ý nghĩa lớn nhất. Sự cân bằng giữa hai tập hợp các kỹ năng sau là một yếu tố cần thiết mang lại thành công trong thiết kế cơ điện tử:

- Lập mô hình (vật lý và toán học), phân tích (mô phỏng số hoá và mô phỏng hệ kín), thiết kế điều khiển (tương tự và số) các hệ vật lý động lực học.
- Kiểm tra bằng thực nghiệm các mô hình, phân tích và hiểu rõ các vấn đề cơ bản trong thiết kế phần cứng

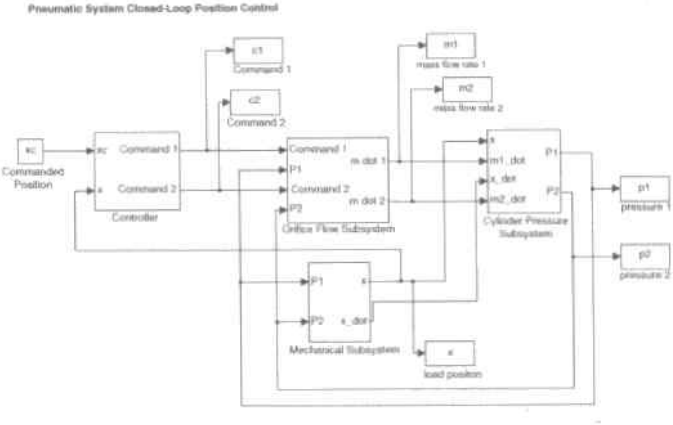
Trong các hệ cơ điện tử, máy tính có vai trò rất đa dạng. Trước hết, máy tính được sử dụng để lập mô hình, phân tích và mô phỏng các hệ cơ điện tử cũng như các thành phần của hệ cơ điện tử, do đó rất có ích cho quá trình thiết kế. Thứ hai, máy tính với vai trò là một phần của hệ thống đo, được dùng để đo sự hoạt động của các hệ cơ điện tử, xác định giá trị của các thông số thành phần và kiểm tra bằng thực nghiệm các mô hình. Cuối cùng, máy tính và vi máy tính là thành phần

trung tâm trong các hệ thống điều khiển số cho các thiết kế cơ điện tử. Do đó, máy tính đóng vai trò quan trọng vào sự cân bằng trong cơ điện tử và cũng là một thành phần chính trong thiết kế các hệ cơ điện tử. Điều này được minh họa bằng một ví dụ dưới đây.



HÌNH 35.1 Cơ cấu Servo khí nén

Xét một hệ thống điều khiển vị trí vòng kín bằng máy tính và cơ cấu servo khí nén, như trên hình 35.1. Cơ cấu servo khí nén có ưu điểm là giá thành thấp, tỷ số truyền lực lớn, dễ bảo trì, sạch, nguồn năng lượng rẻ và có sẵn. Tuy vậy, nhược điểm là lực ma sát lớn và phi tuyến, có vùng chết do ma sát và thời gian chết do tỷ số nén của khí. Mục tiêu thiết kế là tạo ra một hệ thống truyền động khí nén hoạt động nhanh, chính xác và giá thành thấp nhờ sử dụng các van từ on/off thay cho các van servo biến đổi liên tục đắt tiền. Để thực hiện nhiệm vụ này, trước hết phải hiểu rõ về hệ thống vật lý, phát triển một mô hình vật lý phục vụ việc phân tích, thiết kế, xác định và kiểm tra các tham số mô hình theo thực nghiệm. Công việc tiếp theo là phát triển mô hình toán học của hệ thống, phân tích hệ thống và so sánh các kết quả phân tích với kết quả đo đạc thực nghiệm để kiểm tra mô hình. Sau đó phải thiết kế một hệ thống điều khiển vị trí vòng kín có sử dụng điều khiển on/off hoặc điều khiển bằng phương pháp điều chế độ rộng xung. Sau cùng cần phải thực hiện việc điều khiển và kiểm tra bằng thực nghiệm các hoạt động của hệ thống.



HÌNH 35.2 Mô hình trên Matlab/Simulink của Cơ cấu Servo khí nén

Hình 35.2 là mô hình trên Matlab/Simulink của hệ thống này. Mô hình toán học có tính phi tuyến cao. Việc mô phỏng trên máy tính giúp chúng ta hiểu rõ về ứng xử của một hệ thống và các sơ đồ điều khiển khác nhau. Hệ thống thu thập dữ liệu được dùng để đo các giá trị đầu vào/ra của hệ thống và phục vụ việc mô phỏng số. Máy tính (hay vi điều khiển trong trường hợp này) dùng để thực hiện các sơ đồ điều khiển theo thời gian thực. Có nhiều công cụ hỗ trợ việc mô phỏng, một số

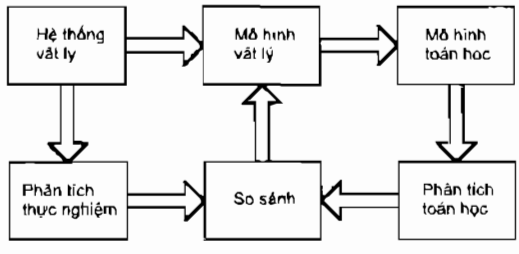
yêu cầu mô hình toán học chi tiết, trong khi một số khác cung cấp các mô hình ảo, ở đó các thành phần của hệ thống, với các mô hình toán học kèm theo, được ghép nối với nhau trên màn hình máy tính. Cũng có nhiều công cụ dựa trên nền máy tính dùng để chạy các thuật toán điều khiển, ví dụ như một máy tính sử dụng card DSP và một bộ tạo lệnh điều khiển theo thời gian thực, một vi điều khiển có khả năng lập trình bằng ngôn ngữ C hoặc Basic với bộ chuyển đổi A/D và các cổng vào/ra số, và một bộ vi mạch dùng cho việc phát triển sản phẩm.

35.2 Mô hình và mô phỏng Cơ điện tử trên máy tính

Trong thiết kế, sự cân bằng là chìa khoá của thành công, ví dụ như sự cân bằng giữa lý thuyết và thực hành, giữa kỹ năng lập mô hình/phân tích với kỹ năng thao tác phần cứng/đo đạc. Hình 35.3 minh họa các bước nghiên cứu một hệ thống động học, một quy trình được sử dụng trong thiết kế các hệ cơ điện tử. Sự phân biệt giữa mô hình vật lý và mô hình toán học được nhấn mạnh vì nó rất quan trọng đối với các phương pháp giải số, các phương pháp giải tích và các phương trình mô hình. Để tạo ra một mô hình vật lý người ta thường dùng phương pháp xấp xỉ cho hệ thống vật lý thực bằng cách bỏ qua các ảnh hưởng nhỏ, cũng như các ảnh hưởng từ môi trường. Các thành phần được giả sử là tập trung thay cho phân tán. Động lực học là tuyến tính. Các thông số là hằng. Nhiễu và sai số được bỏ qua. Việc xấp xỉ sẽ làm ảnh hưởng trực tiếp đến mô hình toán học. Bỏ qua các ảnh hưởng nhỏ nhằm hạn chế số lượng phương trình. Giảm các yếu tố ảnh hưởng từ môi trường giúp đơn giản hoá các phương trình. Việc xấp xỉ sẽ sinh các phương trình vi phân tuyến tính với hệ số hằng. Trong hầu hết các trường hợp, đánh giá thiết kế là để phát triển một mô hình đơn giản nhất mô tả đầy đủ sự phức tạp của động học hệ thống.

Đáp ứng động học dự đoán của hệ thống nếu không được kiểm tra bằng thực nghiệm sẽ không đáng tin cậy và không có giá trị. Việc so sánh các kết quả dự đoán với các kết quả thực nghiệm là một bước quan trọng trong quy trình nghiên cứu các hệ thống động học.

Các bước trong quy trình nghiên cứu hệ thống động học không chỉ được áp dụng khi tìm hiểu và dự đoán đáp ứng của một hệ thống vật lý thực tế mà còn được áp dụng ngay cả khi hệ thống vật lý là một khái niệm trong quy trình thiết kế cần được phân tích và đánh giá. Sau khi biết được các yêu cầu từ sản phẩm hoặc dịch vụ mới, người thiết kế sử dụng các kinh nghiệm đã có (của cá nhân hay của người khác), thiết bị sẵn có, hiểu biết về các luật vật lý, tính sáng tạo để tạo ra ý tưởng thiết kế. Mô hình hoá và phân tích trong quá trình thiết kế là rất quan trọng trong tình huống này. Hiện nay phương pháp xây dựng và kiểm tra không còn được sử dụng để đánh giá các ý tưởng thiết kế bởi vì nó quá tốn kém và mất nhiều thời gian. Việc đánh giá kết quả dự đoán khi không tồn tại một hệ vật lý phụ thuộc rất nhiều vào phần cứng và kinh nghiệm thực hành.



HÌNH 35.3 Qui trình nghiên cứu hệ thống động học

Trong mô hình vật lý, trước hết phải chỉ ra hệ thống vật lý sẽ được nghiên cứu cùng với giới hạn của nó và các biến vào/ra. Khi mô hình hóa các hệ thống động lực học, chúng ta sử dụng các phán đoán kỹ thuật và cần đơn giản hoá để phát triển một mô hình vật lý. Sự phức tạp của mô hình vật lý phụ thuộc vào các yêu cầu đặc biệt, ví dụ như sự lặp lại thiết kế, thiết kế hệ điều khiển, kiểm tra thiết kế điều khiển và hiểu biết về vật lý. Để sử dụng một mô hình yêu cầu chúng ta cần có những hiểu biết nhất định về những thành phần đã bỏ qua khi chọn lựa mô hình đơn giản thay cho các mô hình phức tạp. Thực hiện phép xấp xỉ một cách hợp lý khi bắt đầu nghiên cứu là vấn đề

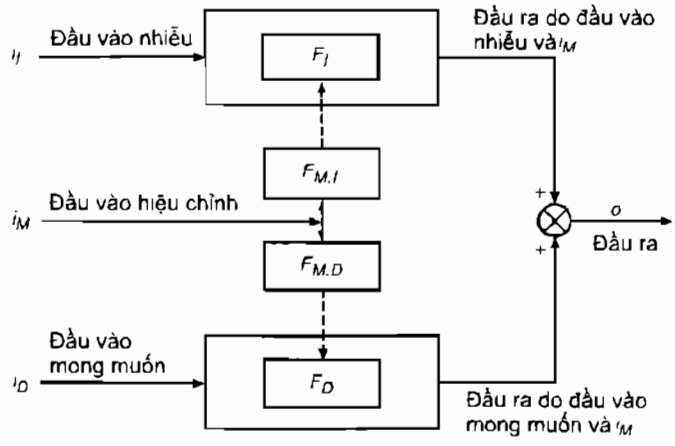
mấu chốt trong phân tích kỹ thuật. Rất nhiều mô hình kỹ thuật được phát triển dựa trên những yêu cầu đặc biệt. Một câu hỏi luôn được đặt ra là " Tại sao tôi đang mô hình hóa một hệ thống vật lý và đâu là giới hạn làm việc của mô hình?". Một *"mô hình thiết kế"* là cần thiết khi có sự lặp lại trong thiết kế hệ thống và thiết kế điều khiển, ví dụ như một mô hình vật lý có mô hình toán học là các phương trình vi phân tuyến tính với hệ số hằng, do vậy phù hợp với các kỹ thuật thiết kế tuyến tính. Một *"mô hình thật"* là cần thiết khi cần có sự kiểm tra trước khi chế tạo phần cứng, ví dụ như một mô hình vật lý gần giống thực tế, với các công cụ mô phỏng phi tuyến có sẵn, có thể mô phỏng hầu hết các mô hình toán học. Mô hình mới có thể được thực hiện bằng cách kế thừa các kết quả của mô hình cũ. Các mô hình chỉ cần hợp lệ trong giới hạn làm việc của vấn đề đang quan tâm, các mô hình bậc thấp có thể biểu diễn các mô hình bậc cao phức tạp rất hiệu quả. Trong thực tế, người thiết kế phải sử dụng nhiều loại mô hình với độ phức tạp khác nhau: một mô hình thật chi tiết cho lần đánh giá cuối cùng trước khi chế tạo phần cứng, các mô hình đơn giản hơn để kiểm tra một vài yêu cầu đặc biệt, và một hoặc một số mô hình thiết kế.

35.3 Các hệ thống đo lường, máy tính và cơ điện tử

Các hệ thống đo lường hoặc các hệ thống thu thập dữ liệu có thể được dùng cho nhiều mục đích, và ở đó máy tính đóng một vai trò to lớn.

1. *Theo dõi các quá trình.* Các ứng dụng cụ thể của thiết bị đo lường có thể thực hiện một số chức năng theo dõi như điện kế, áp kế, các công tơ đo điện, nước và khí.
2. *Điều khiển quá trình.* Một dụng cụ đo lường có vai trò như một bộ phận của hệ thống đo lường. Để điều khiển bất kỳ một biến nào trong mạch điều khiển phản hồi thì ta phải đo nó. Một hệ thống điều khiển đơn lẻ có thể nhận thông tin từ nhiều thiết bị đo, ví dụ như các máy móc công nghiệp và các bộ điều khiển quá trình, các hệ thống điều khiển máy bay.
3. *Phân tích kỹ thuật thực nghiệm.* Khi xử lý các vấn đề kỹ thuật có hai phương pháp truyền thống: lý thuyết và thực nghiệm. Nhiều vấn đề cần phải thực hiện đồng thời bằng cả hai phương pháp và chúng thường bổ xung cho nhau. Hơn thế nữa là việc cần những mô hình hợp lệ và các thiết bị đo lường để thu thập thông tin từ chúng.

Sự phân biệt giữa các chức năng theo dõi, điều khiển và phân tích là không được rõ ràng. Khi được đưa vào ứng dụng có thể phù hợp, có thể phụ thuộc vào quan điểm của người kỹ sư và sự khác nhau khi phân loại sẽ không gây ra bất kỳ khó khăn nào. Trong đó máy tính giống như trung tâm xử lý có thể phục vụ với nhiều chức năng trong quá trình xử lý các thông số đo được từ các hệ cơ điện tử và những chức năng xử lý này có thể liên quan hoặc không liên quan tới mô hình và phần điều khiển của hệ thống.



HÌNH 35.4 Cấu hình vào ra của một hệ thống đo lường

Bộ xử lý tín hiệu số tốc độ cao (DSP) được dùng để thu thập các tín hiệu đầu vào và đầu ra trong việc quyết định các chức năng truyền thông cho các hệ thống cơ điện tử. Tốc độ cao cho phép xử lý mô phỏng đơn giản của các đầu vào và đầu ra với lỗi nhỏ nhất. Tuy nhiên, ứng dụng chính của DSP trong các hệ thống cơ điện tử là điều khiển thời gian thực, sẽ được thảo luận ở phần dưới.

Hình 35.4 là cấu hình vào ra của một hệ thống đo lường. Đầu vào được phân chia ra thành:

1. *Các đầu vào mong muốn.* Số lượng phụ thuộc vào từng yếu tố cần đo.
2. *Các nhiễu đầu vào.* Số lượng của chúng tới các thiết bị là bất định. F_D và F_I có quan hệ vào - ra, ví dụ như các phép tính toán học cần có cả đầu ra từ mối quan hệ với đầu vào. Chúng mang đến các khái niệm khác nhau phụ thuộc vào từng đặc tính vào - ra cụ thể được mô tả (một hằng số, một hàm toán học, một phương trình vi phân, một hàm phân bố xác suất).
3. *Các đầu vào hiệu chỉnh.* Chúng tạo ra sự thay đổi trong quan hệ vào - ra của các lệnh đầu mong muốn và các nhiễu đầu vào. Ví dụ như chúng gây nên sự thay đổi cho F_D và/hoặc F_I, $F_{M,I}$ và $F_{M,D}$.

Có một số phương pháp nhằm làm mất hoặc giảm ảnh hưởng của các đầu vào lỗi. Phương pháp dựa trên xử lý tín hiệu của máy tính là phương pháp hiệu chỉnh các đầu ra đã được tính toán. Phương pháp này yêu cầu đo hoặc xác định độ lớn của các nhiễu đầu vào hoặc tín hiệu điều chỉnh đầu vào để biết một cách định lượng chúng ảnh hưởng thế nào đến đầu ra. Sau đó có thể tính phần hiệu chỉnh phải thêm vào hoặc bớt khỏi đầu ra đó sao cho (lí tưởng) chỉ duy nhất thành phần đó có liên quan với đầu vào mong muốn. Ngày nay, người ta có thể thêm vào máy tính nhiều hệ thống đo làm cho nó có các tính năng đa dạng. Nếu cung cấp các cảm biến để đo các đầu vào lỗi thì máy tính có thể thực hiện phương pháp hiệu chỉnh các đầu ra đã được tính toán một cách tự động.

35.4 Cơ điện tử và ứng dụng tính năng thời gian thực của máy tính

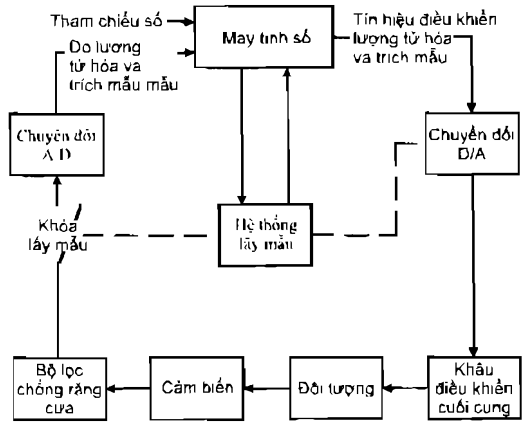
Chúng ta quay lại phần điều khiển vòng kín sử dụng máy tính số như là bộ điều khiển. Rất nhiều vấn đề sẽ được đề cập ở đây. Trước tiên, một hệ thống cơ điện tử thường liên quan đến các biến liên tục. Các thành phần quay hoặc tịnh tiến trong không gian. Dòng chảy của khí hoặc chất lỏng. Sự truyền nhiệt hoặc năng lượng. Trong khi đó máy tính là thành phần dạng số. Các biến được đưa vào máy tính dưới dạng rời rạc hoặc đơn giản bởi dãy các kí tự 0 và 1. Để máy tính hoạt động như một bộ điều khiển cho một hệ thống cơ điện tử thì các biến liên tục phải được chuyển đổi thành các biến rời rạc, sau quá trình xử lý nó lại được đưa ra ngoài dưới dạng biến liên tục. Điều này dường như là hiển nhiên. Cái không hiển nhiên ở đây là thuật toán trên máy tính đã tạo ra một sự tách biệt giữa việc xử lý tín hiệu và chính các tín hiệu (điều này không đúng cho thành phần của hệ cơ điện tử khác). Mặc dù các phần tử lôgic số được sử dụng, các tín hiệu số được chuyển sang dạng rời rạc nhưng dòng thông tin thì vẫn tiếp tục chảy qua các điểm. Khi một máy tính được sử dụng cho điều khiển điểm, dòng thông tin này bị huỷ và bị che mất trong thuật toán máy tính. Ví dụ đối khi các thuật toán máy tính bắt chước luật điều khiển PID liên tục. Khi tiến hành phân tích thuật toán này, mặc dù các hiệu ứng của việc trích mẫu và lượng tử hoá được đưa vào, chúng ta phải thừa nhận rằng các tín hiệu được xử lý như thể là chúng đã được xác định rõ bởi các phần tử xử lý liên tục. Trong thực tế, nếu mã máy tính được kiểm tra ở mức độ thấp (không phải là ngôn ngữ bậc cao), nó sẽ gần giống với một phương trình vi phân mô tả thuật toán PID. Điều này liên quan đến cả việc mô hình hoá hoạt động chính xác của máy tính giống như một phần tử điều khiển và việc hợp lý hoá mã máy tính tạo ra các tín hiệu có đáp ứng mong muốn.

Một vấn đề khác là bộ điều khiển của hệ thống cơ điện tử có thể thực hiện trên phần mềm. Việc thực hiện trên phần mềm thường là không đồng bộ với các hằng số thời gian khác trong hệ thống. Phần mềm có thể được đồng bộ bởi việc đồng bộ hóa nó với một chu kì trích mẫu, nhưng việc này là khó khăn và bị hạn chế khi máy tính được sử dụng cho nhiệm vụ khác ngoài điều khiển. Trước tiên một máy tính được xem như là một phần tử trong hệ thống cơ điện tử, có một xu hướng là sử dụng khả năng xử lý để cung cấp các chức năng đi kèm hoặc để sử dụng sản phẩm dễ

dàng hơn. Mã thêm vào này có thể ảnh hưởng (có thể là ngược lại) tới quá trình thực hiện của bộ điều khiển thời gian thực. Việc kiểm tra các mã và sự an toàn của các mã cũng là một vấn đề. Người kỹ sư phải xác định rằng hệ thống của anh ta có điều hành theo như dự định và an toàn cho mọi khả năng tổ hợp có thể của tín hiệu đầu vào và cho mọi trạng thái có thể trong khi thực hiện các thuật toán. Trong hệ thống thời gian thực, thứ tự tiến hành cho mã lệnh thường không được dự đoán trước vì nó bị phụ thuộc vào sự tổng hợp của các tín hiệu đầu vào. Sự đơn giản của mã lệnh làm nó dễ dàng hơn khi kiểm tra, việc sử dụng các phần mềm viết sẵn sẽ bảo đảm chất lượng thực hành và việc lưu hành rộng rãi các mã nguồn và tài liệu là cách để phát triển một hệ thống nhanh hơn. Thông thường, để đảm bảo an toàn người ta sử dụng một khóa liên động (hệ thống an toàn dùng điện và các phần cứng cơ khí) trong các hệ thống được điều khiển bằng phần mềm.

Hoạt động của mã lệnh phải được xác minh cẩn thận khi chúng bị thay đổi hoặc khi được sử dụng lại. Không giống như các bộ điều khiển khác, lệnh máy tính khá linh động nên cần phải suy nghĩ kỹ hơn về khả năng lặp lại của nó. Việc dùng các gói phần mềm chuẩn, các bộ xử lý chuẩn, các mô đun lệnh và các môi trường thời gian thực sẽ làm tăng khả năng tái sử dụng.

Bên cạnh những yếu tố được hỗ trợ khi sử dụng mã máy tính như là một bộ điều khiển, còn tồn tại một số vấn đề khi các bộ xử lý số được kết hợp vào hệ cơ điện tử. Vì thế cần phải tính toán khi xử lý tín hiệu số. Hình 35.5 là một cấu hình phù hợp cho việc này. Máy tính là quan trọng, nhưng "bộ phận" máy tính trong nhiều máy móc cơ điện tử và các quy trình thường không đạt được hiệu quả tốt nhất xét trong cả yếu tố kỹ thuật và kinh tế. Dĩ nhiên là các thành phần bên ngoài máy tính như các cơ cấu chấp hành, các cảm biến, hệ thống trích mẫu và bộ lọc chống răng cưa là các nhân tố giới hạn trong thiết kế hệ thống.



HÌNH 35.5 Cấu hình máy tính - điều khiển tổng quát

Phân chia các tín hiệu tín hiệu số và tín hiệu tương tự dùng trong hệ thống điều khiển bằng máy tính như sau:

Phân loại tín hiệu	Rời rạc theo thời gian	Liên tục theo thời gian
Rời rạc theo biên độ	D – D (số)	D – C
Liên tục theo biên độ	C – D	C – C (tương tự)

Với các tín hiệu tương tự, giá trị chính xác của các đại lượng mang thông tin như điện thế, góc quay là rất quan trọng. Nghĩa là các tín hiệu tương tự và số có dạng sóng đặc biệt là cực kỳ quan trọng. Ngược lại, các tín hiệu nhị phân (on, off) trong tự nhiên và sự biến đổi trong giá trị dạng số có liên quan tới sự thay đổi trong các trạng thái lôgic (true/ false) của các công tắc. Ví dụ như +2 hoặc +5 V thể hiện trạng thái đóng, 0 V hoặc +0.8 V thể hiện trạng thái mở.

Trong các thiết bị số, các mức lôgic 0 và lôgic 1 được biểu diễn bằng điện thế trong các dải rộng, giá trị chính xác của các tín hiệu là không cần thiết. Do đó các thiết bị số hiệu chỉnh được các sai lệch của điện thế mà không cần chính xác về giá trị của nó mặc dù toàn bộ hệ thống là tuyệt đối chính xác. Khi kết hợp hai hệ thống liên tục và số, các thành phần số không cần giới hạn độ chính xác hệ thống. Nói chung các giới hạn này thường được kết hợp vào các thành phần liên tục và/ hoặc các thiết bị chuyển đổi từ liên tục sang số (A/D). Trong thực tế, các hệ cơ điện tử là hệ thống liên tục, do đó phải có các bộ chuyển đổi A/D và D/A hoạt động như một phiên dịch giúp máy tính có thể trao đổi với thế giới liên tục bên ngoài.

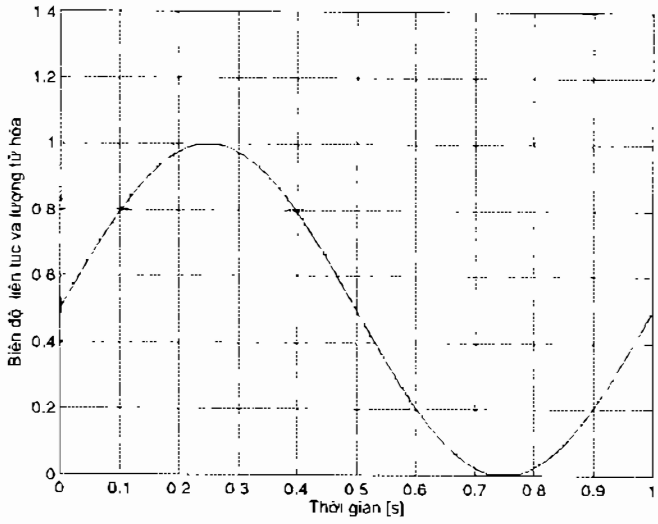
Trong hầu hết các trường hợp, cảm biến và các phần tử điều khiển đều là các thiết bị tương tự, do vậy cần phải có các bộ chuyển đổi A/D và D/A tại cổng vào/ra của máy tính. Dĩ nhiên là có cả động cơ bước và cả bộ mã hoá quang. Tuy nhiên trong nhiều trường hợp các cảm biến có đầu ra tương tự và các phần tử điều khiển cuối cùng vẫn chấp nhận tín hiệu vào là tín hiệu tương tự.

Xu hướng hiện nay là sử dụng các hệ thống điều khiển số phân tán dựa trên máy tính trong các ứng dụng cơ điện tử có thể phù hợp với các tiến bộ cơ bản của điều khiển số:

- Điều khiển số ít nhạy cảm hơn với sự không ổn định của các thiết bị bởi dữ liệu được đo, xử lý, chuyển đổi thông qua hệ nhị phân với các bit có hai trạng thái.
- Quá trình xử lý số chính xác và nhanh. Tuy nhiên việc thiết kế phần cứng thường nhanh hơn thiết kế phần mềm. Việc xác định thời gian để thiết kế phần mềm là cực kì khó.
- Điều khiển số có thể giải quyết các nhiệm vụ lặp lại rất tốt thông qua các chương trình.
- Các luật điều khiển phức tạp và các phương pháp xử lý tín hiệu có thể được lập trình, điều này khó thực hiện bằng các thiết bị tương tự. Các thuật toán rất phức tạp có thể được thực hiện theo kiểu số.
- Sản phẩm có độ tin cậy cao có thể đạt được bằng cách tối thiểu hóa các bộ phận phần cứng tương tự và thông qua sự phân tán bằng cách sử dụng máy tính cho nhiều nhiệm vụ điều khiển.
- Các hệ thống số dễ lập trình hơn và cung cấp khả năng chia sẻ một đơn vị xử lý giữa các chức năng khác nhau.
- Một khối lượng lớn dữ liệu có thể được lưu trữ bằng phương pháp lưu trữ dữ liệu nén mật độ cao.
- Dữ liệu có thể được lưu trữ hoặc bảo tồn trong thời gian dài mà không bị mất và không bị ảnh hưởng bởi các yếu tố môi trường bất lợi bên ngoài, khả năng phục hồi dữ liệu nhanh và dễ dàng.
- Khả năng truyền dữ liệu với khoảng cách xa nhanh hơn và không bị trễ động học đầu vào giống như hệ thống tương tự.
- Nguồn dùng cho các thiết bị số có hiệu điện thế thấp (0 – 12 VDC).
- Các thiết bị điều khiển số thường có giá thành thấp.

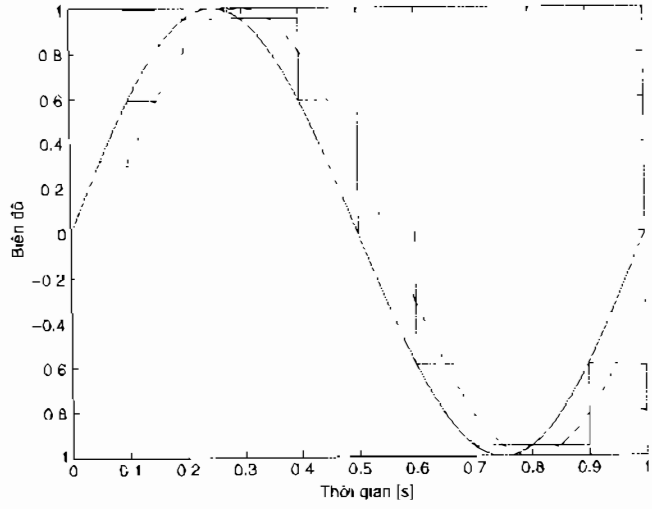
Hơn thế nữa, đứng trên quan điểm cơ điện tử, việc áp dụng máy tính vào hệ thống sẽ đem đến nhiều tính năng mới. Người sử dụng có thể lựa chọn cho mình từ một loạt các hệ điều hành với các tính năng riêng biệt. Giao diện sử dụng với các chỉ dẫn thuận tiện mà giá thành lại thấp.

Nhưng thực tế cũng có một số vấn đề khi phân tích và thiết kế các hệ điều khiển số (như vấn đề phần mềm, một vấn đề mang tính cố hữu) có liên quan tới việc tính toán các hiệu ứng của thời gian trích mẫu (T) và tần số trích mẫu (q). Nếu T và q là rất nhỏ thì các tín hiệu số gần như liên tục và có thể dùng phương pháp phân tích và thiết kế trong trường hợp này. Hiểu được các ảnh hưởng của *tốc độ trích mẫu* (nhanh hay chậm) và *tần số trích mẫu* (rộng hay hẹp) là rất quan trọng. Các máy tính giá thành thấp thường chậm và có kích thước từ nhỏ. Hình 35.6 minh họa các ảnh hưởng của việc lượng tử hoá với kích thước rất nhỏ và độ rộng một từ đơn cũng rất nhỏ. Các tín hiệu được xử lý bởi bộ điều khiển có sai lệch lớn hơn tín hiệu tương tự ban đầu.



HÌNH 35.6 Mô phỏng của một tín hiệu liên tục và lượng tử hóa 4 bit

Hình 35.7 minh họa các ảnh hưởng của việc trích mẫu. Một điều cần chú ý khi thực hiện hệ thống điều khiển số đó là thường có trễ trong các thiết bị chuyển đổi D/A. Trễ gây ra ảnh hưởng đáng kể trong các hệ thống phản hồi mạch kín và thường giới hạn phạm vi điều khiển (ở đây độ trễ là $T/2$).

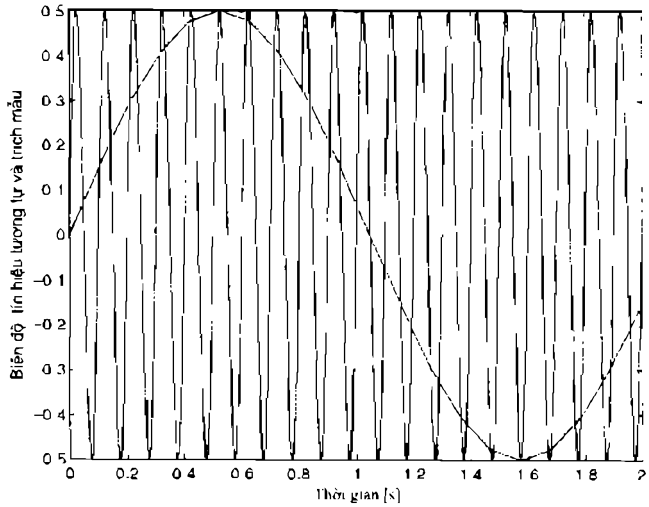


HÌNH 35.7 Tín hiệu liên tục và đầu ra của bộ chuyển đổi D/A

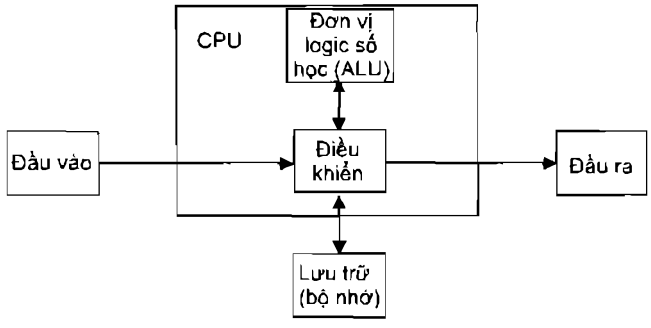
Tín hiệu tương tự trong một hệ thống phản hồi đến từ cảm biến chứa các thông tin hữu ích liên quan tới nhiễu có thể điều khiển (tần số tương đối thấp), nhưng nó cũng có thể chứa các "nhiễu" không thể điều khiển có tần số cao hơn (quá nhanh để điều khiển chính xác). Giống như trong hệ tương tự, để hạn chế các tín hiệu nhiễu và cho phép việc điều khiển tốt hơn người ta sử dụng bộ lọc thông thấp. Nếu bỏ qua bộ lọc thì các tín hiệu này sẽ làm mất tính ổn định của hệ thống điều khiển.

Cuối cùng, trong các hệ thống số, một hiện tượng gọi là tín hiệu răng cưa gây ra một vài vấn đề về nhiễu. Nếu một tín hiệu tần số cao được trích mẫu quá ít, tín hiệu đầu ra của mẫu sẽ chứa các thành phần tần số thấp. Hình 35.8 minh họa điều này. Nếu trích mẫu quá ít một tín hiệu có chứa nhiều thành phần tần số cao thì kết quả sẽ giống như tín hiệu tần số thấp. Trên quan điểm điều khiển, rất khó để xác định đâu là tín hiệu hiện thời. Nếu chúng ta lại dùng các tín hiệu điều khiển sai có tần số thấp thì kết quả điều khiển đem lại rất xấu. Theo lý thuyết thì tỷ lệ trích mẫu tối thiểu để không gây ra hiện tượng răng cưa là hai mẫu trên một chu kỳ trích mẫu, tuy nhiên trong thực tế

người ta thường dùng là trên 10 mẫu cho một chu kỳ. Một tín hiệu tần số cao, được trích mẫu không đầy đủ, có thể sinh ra một tín hiệu có tần số thấp hơn nhiều và nó không thể phân biệt với tín hiệu được sinh ra bằng cách trích mẫu đầy đủ tín hiệu tần số thấp.



HÌNH 35.8 Mô phỏng của tín hiệu liên tục và được trích mẫu: méo răng cưa

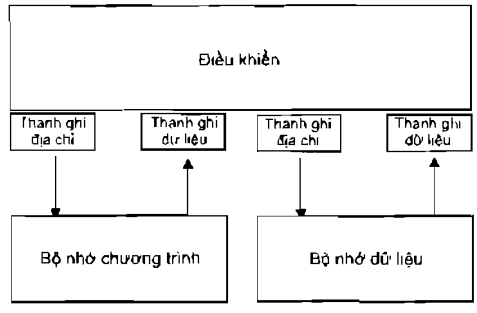


HÌNH 35.9 Các phần tử của máy tính

Trong phần trên, khái niệm máy tính được dùng cho các phần tử xử lý số. Trong lĩnh vực điện tử, người ta đưa ra sự phân biệt giữa một bộ vi xử lý, vi máy tính, DSP và máy tính. Hình 35.9 là sơ đồ khối tổng quát cho một máy tính. Tất cả các máy tính đều có đầu vào, đầu ra, bộ phận điều khiển luồng dữ liệu, bộ nhớ lưu trữ dữ liệu, và một đơn vị logic thuật toán (ALU) để thực hiện các lệnh. ALU và các phần tử điều khiển thường được gọi chung là bộ xử lý trung tâm (CPU). Các máy tính nhỏ chỉ có một CPU thì gọi là bộ vi xử lý. Bộ nhớ cho các máy tính này thường được gắn với vi xử lý nhưng trong khối điện tử khác. Vào/ra của vi xử lý được thực hiện bởi các thiết bị điện tử gọi là thiết bị ngoại vi. Nếu bộ nhớ được tích hợp trên cùng một khối thì có thể được gọi là máy vi tính hoặc hoặc máy tính tùy thuộc vào kích thước vật lý của nó. CPU và bộ nhớ trên một chip đơn thường được gọi là vi máy tính. Bạn đọc lưu ý rằng một khối điện tử đơn có thể chứa nhiều chip được nối với nhau thông qua các dây dẫn và toàn bộ khối vẫn được gọi là chip. Cuối cùng nếu máy tính có thêm chức năng chuyển đổi A/D và D/A thì nó được gọi là DSP. Tuy nhiên những chức năng trên cũng có thể có trong vi máy tính. DSP cũng có thể coi là máy tính có lệnh đặc biệt ở ALU và gọi là lệnh tích luỹ đa năng (MAC) ngay cả khi nó không có chức năng chuyển đổi D/A, A/D. Các thuật toán xử lý tín hiệu số thường liên quan tới các lệnh MAC và một máy tính có thể xử lý các lệnh này một cách hiệu quả thì người ta gọi đó là DSP. Trường hợp phức tạp hơn là các thiết bị điện tử ASIC. Các thiết bị này có thể thực hiện các chức năng đặc biệt (như thuật toán PID). ASIC có thể chứa một CPU hoặc một bộ nhớ

hoặc các chức năng ngoại vi hoặc thậm chí là một tế bào MAC như một phần cấu trúc của nó. Nếu bạn đọc không bị rối tung trong cách lý giải trên thì hoàn toàn có khả năng hiểu thấu đáo vấn đề này.

Trước khi kết thúc phần này, một điểm nữa bạn cũng nên biết là bộ nhớ của máy tính có thể được chia thành không gian lập trình và không gian dữ liệu như trong hình 35.10. Kiểu diễn đạt này tốt hơn là định nghĩa một kiến trúc máy tính đặc biệt. Trong kiến trúc của Von Neumann, bộ nhớ lập trình và bộ nhớ dữ liệu dùng chung không gian nhớ và các bus thông tin. Ngược lại trong kiến trúc Harvard, bộ nhớ chương trình và bộ nhớ dữ liệu là riêng biệt. Trong cả hai trường hợp, đối với một hệ cơ điện tử, có thể hiểu chương trình (trong bộ nhớ lập trình) như là một tập lệnh để CPU biết cách chế tác dữ liệu (trong bộ nhớ dữ liệu) tạo ra đầu ra. Điều này nhấn mạnh quan điểm trước đây rằng dòng tín hiệu trong hệ cơ điện tử trở nên lẫn lộn nếu máy tính sử dụng điều khiển thời gian thực.



HÌNH 35.10 Tổ chức bộ nhớ máy tính

Bởi vì giá thành của các bộ vi máy tính hiện đại là khá rẻ, do đó việc sử dụng các phần tử logic như là những thành phần rời rạc trong hệ cơ điện tử đã giảm dần. Vi máy tính thường được lập trình để thực hiện các chức năng lôgic mà ưu điểm của nó là có thể thay đổi việc lập trình phần mềm thay cho việc thay đổi phần cứng. Theo lôgic này việc phân tích cũng có thể áp dụng một vài phương pháp truyền thống. Ví dụ, có thể tối giản logic theo bản đồ Karnaugh. Điểm khác biệt duy nhất nằm trong việc thực hiện các thuật toán. Người ta luôn sử dụng ASIC để thực hiện các chức năng lôgic.

35.5 Tính tổng hợp của Cơ điện tử

Như khẳng định ngay từ đầu chương, cơ điện tử là sự kết hợp hài hòa của kỹ thuật cơ khí, điện tử, các hệ thống điều khiển, và máy tính. Yếu tố chính của cơ điện tử là sự tích hợp các phần trên nhờ một quy trình thiết kế. Việc sử dụng máy tính và các phần tử lôgic như là thành phần của các hệ cơ điện tử chỉ mang lại những thiết kế thành công khi có sự kết hợp trên. Máy tính phải được ghép đồng thời trong quá trình thiết kế chứ không phải là một thành phần thêm vào khi hoàn tất thiết kế. Khi không có sự kết hợp của máy tính vào hệ thống thì khả năng của một hệ cơ điện tử sẽ bị hạn chế.

36

Các khái niệm về lôgic số và thiết kế lôgic tổ hợp

36.1 Giới thiệu 36-1
36.2 Biểu diễn thông tin dạng số 36-1
36.3 Các hệ thống số 36-2
36.4 Biểu diễn số 36-2
36.5 Số học 36-2
36.6 Chuyển đổi cơ số 36-3
36.7 Phần bù 36-6
36.8 Mã 36-8
36.9 Đại số Boolean 36-9
36.10 Các hàm Boolean 36-10
36.11 Các mạch cổng 36-12
36.12 Các dạng khai triển 36-12
36.13 Thực hiện 36-13
36.14 Các lược đồ thời gian 36-14
36.15 Rủi ro 36-14
36.16 Các dạng giản đồ K 36-14
36.17 Giản đồ K và tối giản hóa 36-17
36.18 Tối thiểu hóa với giản đồ K 36-17
36.19 Tối giản hóa theo bảng Quine-McCluskey 36-19

George I. Cohn
Califonia State University, Fullerton

36.1 Giới thiệu

Lôgic số giải quyết các vấn đề về biểu diễn, truyền, thao tác, lưu trữ thông tin số. Một đại lượng ở dạng số chỉ có các giá trị rời rạc nhất định, ngược lại với dạng tương tự có thể có giá trị bất kỳ nằm trong dải cho phép. Ưu điểm nổi trội của dạng số so với dạng tương tự là nó làm giảm ảnh hưởng của nhiễu nếu nhiễu không vượt quá ngưỡng cho phép.

36.2 Biểu diễn thông tin dạng số

Thông tin có thể được mô tả về mặt số lượng hoặc chất lượng. Thông tin số lượng cần một hệ số để biểu diễn cho nó. Thông tin chất lượng thì không cần. Thông tin đã số hóa được biểu diễn bởi một tập hợp hữu hạn các ký tự khác nhau. Mỗi ký tự sẽ là một định lượng rời rạc của thông tin. Tập hợp các ký tự này được sử dụng để tạo thành bảng chữ cái.

BẢNG 36.1 Ký hiệu cho các số

	Dạng liền kề	Dạng đa thức
Số nguyên	$N = N_{n-1}N_{n-2}{}^{n}N_1N_0$	$N = \sum_{k=0}^{n-1} N_k R^k$
Số thập phân	$F = F_{-1}F_{-2}{}^{o}F_{-m+1}F_{-m}$	$F = \sum_{k=-m}^{-1} F_k R^k$
Số thực	$X = X_{n-1}X_{n-2}{}^{a}X_1X_0\Diamond X_{-1}X_{-1}X_{-m+1}X_{-m}$	$X = \sum_{k=-m}^{n-1} X_k R^k$

36.3 Các hệ số

Thông tin số lượng được biểu diễn bởi một hệ số. Một ký tự biểu diễn cho thông tin về số lượng được gọi là một **chữ số**. Số các giá trị khác nhau mà một chữ số có được gọi là **cơ số**, được ký hiệu là R. Những ký hiệu biểu diễn các giá trị khác nhau của một chữ số có được gọi là các ký tự số. Các ký tự số thường được sử dụng nhiều nhất là 0, 1, 2, v.v, trong đó 0 biểu diễn cho giá trị nhỏ nhất. Giá trị lớn nhất có thể có trong một hệ số là **cơ số được rút gọn**, r = R – 1. Các cơ số có giá trị khác nhau thì đại diện cho các hệ số khác nhau. Ví dụ như R = 2 thì hệ cơ số đó gọi là hệ nhị phân, với R = 3 thì ta gọi là hệ tam phân, với R = 8 ta gọi là hệ bát phân, với R = 10 thì ta gọi là hệ thập phân và với R = 16 thì ta gọi đó là hệ cơ số hexa.

Bất kỳ một giá trị nào được biểu diễn đơn lẻ bằng các chữ số thì gọi là số nguyên. Một số nguyên âm là một số nguyên có được bằng cách lấy một số nguyên trừ đi số nguyên có giá trị lớn hơn và ta ký hiệu dấu trừ đứng trước số nguyên đó. Lấy một số nguyên bất kỳ chia cho số nguyên lớn hơn nó ta sẽ được một **phân số**. Một số có cả phần nguyên và phần phân số gọi là một **số thực**.

Tất cả các số trong một hệ số có cùng một cơ số. Cơ số là cơ sở cho một hệ số. Có lẽ con người có mười ngón tay nên người ta dùng hệ cơ số 10 cho dễ sử dụng. Mỗi ký tự đại diện cho một số khác nhau bao gồm 0, 1, 2, 3, 4, 5, 6, 7, 8 và 9. Hệ cơ số 2 thể hiện thực chất nhất cho các hệ điện tử số bởi vì các trạng thái rời rạc đáng tin trong một hệ thống có thể là hiệu quả nhất và nó được thực hiện với hai trạng thái ổn định của các phần tử. Bình thường hai ký tự được dùng để biểu diễn các số trong hệ 2 là 0 và 1. Hệ 16 (bao gồm các ký tự: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E và F với cơ số R = 16) rất quan trọng bởi vì có thể chia thành một cụm bao gồm chuỗi 4 số mang thông tin nhị phân được lưu trữ và xử lý trong các máy tính số.

36.4 Biểu diễn số

Các chữ số cần một hoặc nhiều ký tự có thể biểu diễn theo các cách nhau khác như trong bảng 36.1. Các dạng khác nhau sẽ thuận tiện hơn khi xử lý bằng các thủ tục khác nhau. Các thuật toán dễ dùng nhất với các số viết theo dạng liền kề. Dạng đa thức phù hợp trong nghiên cứu lý thuyết.

36.5 Số học

Các thuật toán xử lý thông dụng nhất: cộng, trừ, nhân và chia được thực hiện thuận tiện với các ký hiệu. Các đa thức rất có ích khi phát triển các công thức mang tính hệ thống. Kể từ khi áp dụng các chữ số vào trong kỹ thuật số, lôgic được sử dụng để tính toán các số được gọi là *lôgic số*. Lôgic số thường được thực hiện bởi phần cứng. Tuy nhiên nó khác với lôgic của đại số boolean. Bốn phép tính cơ bản là hình thức để tạo ra phương trình hoặc một dạng xử lý thuật toán, như trong bảng 36.2.

Việc xử lý thuật toán trong hệ cơ số 2 được dựa vào hai phép tính là phép cộng và phép nhân nhị phân như trong bảng 36.3. Bảng 36.4 đưa ra ví dụ cho mỗi phép tính cơ bản.

BẢNG 36.2 Các phép toán số học

	Dạng đại số	**Dạng số học**
Phép cộng	Tổng = Số bị cộng + Số cộng	Số bị cộng + Số cộng Tổng
Phép trừ	Hiệu = Số bị trừ - Số trừ	Số bị trừ -Số trừ Hiệu
Phép nhân	Tích = Số bị nhân x số nhân	Số bị nhân + Số nhân Tích
Phép chia	Số bị chia/Số chia = Thương + Số dư/Số chia	Thương Số dư Số chia | Số bị chia

BẢNG 36.3 Bảng số nhị phân đơn nguyên

(a) Phép cộng

	0	1
0	0	1
1	1	10

(b) Phép nhân

	0	1
0	0	0
1	0	1

36.6 Chuyển đổi hệ số

Việc chuyển đổi một số từ hệ này sang hệ khác được thực hiện bằng cách sử dụng các khai triển đa thức chuỗi, minh họa trong bảng 36.5.

Phân tích các đa thức được thực hiện dễ dàng hơn với *dạng ghép* của chúng. Dạng ghép thu được từ dạng chuỗi khi phân tích các biến thành thừa số liên tiếp, như trong bảng 36.6. Số lượng các phép nhân để thực hiện dạng ghép tăng theo bậc của các đa thức một cách tuyến tính, trong khi đó số lượng các phép nhân để thực hiện dạng chuỗi tăng theo bình phương bậc của đa thức.

Ta có thể dùng phương pháp giảm bậc của đa thức để chuyển đổi các số nguyên từ cơ số này sang cơ số khác, như trong bảng 36.6(b). Ký tự có ít ý nghĩa nhất của số trong hệ cơ số mới là phần nhớ thu được sau khi chia số cần đổi cho cơ số mới. Ký tự ít ý nghĩa tiếp theo là phần nhớ thu được sau khi chia đa thức đã làm giảm đầu tiên cho cơ số mới. Quy trình này được lặp lại cho đến khi thu được ký tự quan trọng nhất trong hệ cơ số mới. Bảng 36.7 là ví dụ minh họa phương pháp trên.

Để chuyển đổi một số thập phân từ cơ số này sang cơ số khác ta nhân liên tiếp số thập phân đó với cơ số của hệ số mà ta đang định chuyển sang. Mỗi phép nhân với cơ số sẽ cho một kết quả, nó có các ký tự được chuyển qua phía trái một vị trí. Phép nhân đó chuyển ký tự quan trọng nhất của số thập phân tới vị trí bên trái dấu chấm cơ số, dấu phân chia phần nguyên và phần thập phân, do vậy ta tách nó ra khỏi phân số. Quy trình này được minh họa ở dạng đại số ở cột bên trái của bảng 36.8 và ở dạng đại số ở cột kế tiếp. Hai ví dụ minh họa cho phương pháp chuyển hệ cơ số được trình bày trong bảng 36.8.

BẢNG 36.4 Các ví dụ về toán tử với số nhị phân

Phép cộng:

```
 1 1 0 0     Nhớ
 1 1 1 0 0   Số bị cộng
+1 1 0 1     Số cộng
1 0 1 0 0 1  Tổng
```

Phép nhân:

```
    1 1 0 1     Số bị nhân
  x 1 1 0       Số nhân
    0 0 0 0
  1 1 0 1
1 1 0 1
1 0 0 1 1 1 0   Tích
```

Phép trừ, phương pháp mượn:

```
  1 0     Mượn
1 0 1     Số bị trừ
- 1 0     Số trừ
  1 1     Hiệu
```

Phép chia với phần dư phân số:

```
                    Phần dư
          Thương      1 1 1
            1 0 0 1 0   1 0 1 1
1 0 1 1 | 1 1 0 0 1 1 0 1   Số bị chia
Số chia   1 0 1 1
          0 1 1 1 0
            1 0 1 1
              0 1 1 1
```

Phép trừ, phương pháp trả ngược:

```
  1 0     Mượn
1 0 1     Số bị trừ
- 1       Trả ngược
  - 1 0   Số trừ
    1 1   Hiệu
```

Phép chia với phần dư trong thương số:

```
          Thương
          .1 0 1 0 0 0 1 ...
1 0 1 1 | 1 1 1.0 0 0 0 0 0 0   Số bị chia
Số chia   1 0 1 1
            1 1 0 0
            1 0 1 1
              0 1 0 0 0 0
                1 0 1 1
```

BẢNG 36.5 Biến đổi giữa các hệ số bằng phương pháp đa thức chuỗi

	Ví dụ	
Phương pháp	Từ hệ số thấp về cơ số cao	Từ hệ số cao về cơ số thấp
1. Biểu diễn số ở dạng đa thức theo hệ số đã cho	101.12=1x22+0x21+1x20+1x2-1	36.510=3x101+6x100+5x10-1
2. Biến đổi cơ số và các hệ số về hệ số mới	=1x4+0x2+1x1+1x0.5	=11x10101+110x10100+101x1010-1
3. Các thành phần xác định giá trị trong hệ số mới	4+0+1+0.5	=11x1010+110+101/1010 =11110+11+.1
4. Cộng các thành phần	101.12=5.510	36.510=100100.12

BẢNG 36.6 Các đa thức dạng ghép

(a) Đa thức dạng ghép thông qua các hệ số lặp	(b) Các đa thức bậc thấp hơn
$N = N_{n-1}R^{n-1} + N_{n-2}R^{n-2} + \cdots + N_2R^2 + N_1R + N_0$	$N = N^{(1)}R + N_0$
$N = (N_{n-1}R^{n-2} + N_{n-2}R^{n-3} + \cdots + N_2R + N_1)R + N_0$	$N^{(1)} = N^{(2)}R + N_1$
$N = ((N_{n-1}R^{n-3} + N_{n-2}R^{n-4} + \cdots + N_2)R + N_1)R + N_0$	$N^{(2)} = N^{(3)}R + N_2$
$\vdots$	$\vdots$
	$N^{(n-2)} = N^{(n-3)}R + N_{n-2}$
$N = (\cdots(N_{n-1})R + N_{n-2})R + \cdots + N_2)R + N_1)R + N_0$	$N^{(n-1)} = N_{n-1}$

BẢNG 36.7 Biến đổi giữa các hệ số bằng phương pháp chia cơ số

Phương pháp	Ví dụ: Từ hệ số cao về cơ số thấp	Ví dụ: Từ hệ số thấp về cơ số cao
$R \mid N$	$2 \mid 36_{10}$	
$R \mid N^{(1)}$ N_0	$2 \mid 18$ 0	
$R \mid N^{(2)}$ N_1	$2 \mid 9$ 0	$12 \mid 2012_3$
$R \mid N^{(3)}$ N_2	$2 \mid 4$ 1	$12 \mid 102$ 11
$\vdots$	$2 \mid 2$ 0	$12 \mid 2$ 1
	$2 \mid 1$ 0	$12 \mid 0$ 2
$R \mid N^{(n-1)}$ N_{n-2}	0 1	
$R \mid 0$ N_{n-1}	$36_{10} = 100100_2$	$2012_3 = 214_5$

Bảng 36.8 chỉ thực hiện với các phân số hữu hạn, các phân số mà phần thập phân sẽ biến mất sau một số bước nhất định. Trong trường hợp phân số vô hạn người ta chỉ tính sau một số bước đủ lớn để thu được một số, với kết quả gần chính xác. Trong bảng 36.9 minh họa trường hợp vô hạn. Một dãy các số lặp đi lặp lại vô hạn và kết thúc bằng đường gạch.

BẢNG 36.8 Phương pháp biến đổi nhân cơ số

Biểu diễn: Đại số	Biểu diễn: Số học	Ví dụ: Cao xuống thấp	Ví dụ: Thấp lên cao
$F = F_{-1}F_{-2}F_{-3}\cdots F_{-m}$	$F_{-1}F_{-2}F_{-3}\cdots F_{-m}$		0.100101_2
	$\times R$		$\times 1010$
$R^*F = F_{-1}\cdot F_{-2}F_{-3}\cdots F_{-m}$	$F_{-1} \mid \cdot F_{-2}F_{-3}\cdots F_{-m}$	0.125_{10}	$101 \mid .11001$
$= F_{-1}\cdot F^{(1)}$	$\times R$	$\times 2$	$\times 1010$
$R^*F^{(1)} = F_{-2}\cdot F_{-3}F_{-4}$	$F_{-2} \mid \cdot F_{-3}\cdots F_{-m}$	$0 \mid .25$	$111 \mid .1101$
$\cdots F_{-m} = F_{-2}\cdot F^{(2)}$	$\times R$	$\times 2$	$\times 1010$
		$0 \mid .5$	$1000 \mid 001$
$R^*F^{(2)} = F_{-3}\cdot F_{-4}F_{-5}$	$\vdots$	$\times 2$	$\times 1010$
$\cdots F_{-m} = F_{-2}\cdot F^{(3)}$		$1 \mid .0$	$1 \mid .10$
$\vdots$			$\times 1010$
			$10 \mid .1$
$R^*F^{(m-2)} = F_{-m+1}\cdot F_{-m}$	$F_{-m+1} \mid \cdot F_{-m}$		$\times 1010$
$= F_{-m+1}\cdot F^{(m-1)}$	$\times R$		$101 \mid .$
$R^*F^{(m-1)} = F_{-m}$	$F_{-m} \mid \cdot$	$.125_{10} = .001_2$	$.100101_2 = .578125_{10}$

BẢNG 36.9 Phương pháp biến đổi thập phân vô hạn

$0.1_{10} = 0.0001\,10011\ldots_2$

×2	
0.2	0
×2	
0.4	0
×2	
0.8	0
×2	
1.6	1
×2	
1.2	1
×2	
0.4	0
×2	
0.8	0
×2	
1.6	1

BẢNG 36.10 Chuyển đổi giữa các hệ ở đó một cơ số là một lũy thừa nguyên của cơ số khác

(a) Chuyển đổi từ cơ số cao tới thấp

$B2.C5_{16} = 1011\ 0010\ .1100\ 0101_2$

$62.75_8 = 110\ 010\ .111\ 101_2$

(b) Chuyển đổi từ cơ số thấp tới cao

$11\ 0010\ 0100.0001\ 1100\ 01_2 = 324.1C4_{16}$

$10\ 110\ 001.011111\ 01_2 = 261.372_8$

Việc chuyển sang hệ cơ số 2 từ một cơ số là bội số của 2 có thể được thực hiện đơn giản nhất do việc chuyển đổi độc lập của mỗi ký tự liên tiếp, như minh họa trong bảng 36.10(a). Việc chuyển ngược lại từ cơ số 2 sang cơ số 2^k có thể thực hiện đơn giản bằng cách nhóm k bit trong dãy số thành từng nhóm, mỗi nhóm bắt đầu với bit có ý nghĩa ít nhất cho phần nguyên và với bit có ý nghĩa nhất cho phần thập phân, như minh họa trong bảng 36.10(b).

36.7 Phần bù

Mỗi hệ cơ số đều có hai **phần bù** được sử dụng thuận tiện:

$$\text{Bù cơ số của } N = N^{RC} = R^n - N$$

$$\text{Phần bù hệ cơ số giảm của } N = N^{rC} = N^{RC} - 1$$

Ở đó R là cơ số và n là số các ký tự trong hệ N. Các phương trình trên đưa ra phần bù của những số có độ lớn là N.

Một số dương có thể được biểu diễn bởi một mã trong ngôn ngữ máy 2 ký tự là 0 và 1. Trong hệ cơ số 2, nó biểu diễn đúng giá trị của nó. Một số âm thì cần một bit để mã hóa dấu trong hệ cơ số hai. Ta có thể thực hiện việc này bằng cách mã hóa riêng bit dấu và các bit độ lớn hoặc bằng cách mã hóa một số âm như một thực thể đơn. Bảng 36.11 minh họa bốn dạng mã hóa khác nhau cho một số âm. Các số âm có thể được biểu diễn bằng cách sử dụng bit đầu tiên để thể hiện bit dấu (0 cho dấu + và 1 cho dấu -) và các ký tự còn lại mô tả độ lớn của số. Phần bù và phép dịch rất có

ý nghĩa trong việc mã hóa các số âm như là một thực thể đơn thay vì phải dùng các ký tự rời rạc để mã hóa bit dấu của chúng. Việc sử dụng phần bù cung cấp các dãy giá trị bằng nhau cho cả số âm và số dương. Biểu diễn dịch cũng có thể cung cấp các dãy bằng nhau cho cả số âm và số dương bằng cách chọn lựa giá trị dịch chuyển bằng một nửa số nhị phân lớn nhất có thể được biểu diễn bằng các ký tự sẵn có. Mã dịch chuyển thu được cách lấy mã được xét trừ đi giá trị dịch chuyển.

BẢNG 36.11 Biểu diễn số

		Các số có dấu với các mã đặc biệt			
Mã sẵn có	Các số dương	Độ lớn của dấu	Bù 1	Bù 2	dịch 111
1111	1111	-111	-000	-001	+1000
1110	1110	-110	-001	-010	+111
1101	1101	-101	-010	-011	+110
1100	1100	-100	-011	-100	+101
1011	1011	-011	-100	-101	+100
1010	1010	-010	-101	-110	+011
1001	1001	-001	-110	-111	+010
1000	1000	-000	-111	-1000	+001
0111	0111	+111	+111	+111	000
0110	0110	+110	+110	+110	-001
0101	0101	+101	+101	+101	-010
0100	0100	+100	+100	+100	-011
0010	0010	+010	+010	+010	-101
0001	0001	+001	+001	+001	-110
0000	0000	+000	+000	+000	-111

BẢNG 36.12 So sánh số học có bù và không bù

Ví dụ minh họa	N = 7-5 = 2	N = 5-7 = -2	N = 5+7 = 12	N= -5-7 = -12
Tính toán thủ công (không dùng phần bù)	111 -101 10	(-111) -(-101) -10	101 +111 1100	(-101) +(-111) -1100
Tính toán trên máy với mã bù 2	0111 +1011 10010	0101 +1001 1110	0101 +0111 1100	1011 +1001 10100
	↓	↓	↓	↓
Trường làm việc với ký tự 4-bit	Số dương	Số âm	Số âm	Số dương
Kết quả	+010	-010	-100	-100
Độ chính xác	Đúng	Sai	Sai	Đúng
Nhớ đặc biệt	$C_0 = 1$ $C_1 = 1$	$C_0 = 0$ $C_1 = 0$	$C_0 = 0$ $C_1 = 1$	$C_0 = 1$ $C_1 = 0$
Điều kiện chính xác hoặc tương đương	$C_0 \equiv C_1$ $C_0 \odot C_1 = 1$	$C_0 \equiv C_1$ $C_0 \odot C_1 = 1$	$C_0 \neq C_1$ $C_0 \odot C_1 = 0$	$C_0 \neq C_1$ $C_0 \odot C_1 = 0$

Với phần bù, phép trừ được thực hiện bằng cách cộng với phần bù của nó. Nếu kết quả thu được vừa đủ trong số lượng ký tự tối đa cho phép thì số đó là hợp lệ. Ta phải dự đoán khi kết quả không đúng trong trường hợp hiện tượng tràn số xảy ra, số đó có nhiều kỷ tự hơn so với mức độ cho phép. Bảng 36.12 minh họa các phép tính dùng và không dùng phần bù. Hai cột ở tận cùng bên phải minh họa các trường hợp mà các kết quả tràn so với kích cỡ cho phép của độ lớn. Điều kiện tràn có thể được biểu diễn bởi hai thông số nhớ sau đây:

- C_0, kết quả được nhớ từ kí tự ở vị trí bên trái nhất
- C_1, kết quả được nhớ từ kí tự ở vị trí thứ hai tính từ bên trái sang

Nếu cả hai thông số nhớ trên là đồng nhất (ví dụ như cùng giá trị) thì kết quả vừa vặn với số kí tự cho phép và ta thu được kết quả đúng. Ngược lại, nếu hai thông số là không đồng nhất thì kết quả là sai.

36.8 Mã

Người ta đã phát triển nhiều dạng mã để phục vụ cho nhiều mục đích khác nhau. Một số mã cho phép các kỷ tự trong bảng chữ cái có thể được biểu diễn độc lập bởi các chữ cái nhỏ hơn. Ví dụ như các kỷ tự số thập phân có thể được biểu diễn dưới dạng chữ cái nhị phân của hệ BCD (Binary Coded Decimal) hoặc mã 8421 như trong bảng 36.13.

BẢNG 36.13 Ví dụ về các mã

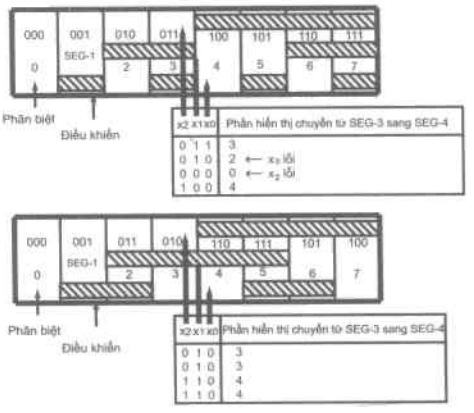
Các mã thập phân	BCD		2 trên 5	Chẵn lẻ		Gray		
	8421	2421		Chẵn	Lẻ	1-bit	2-bit	3-bit
0	0000	0000	00011	0000 1	0000 1	0	00	000
1	0001	0001	00101	0001 1	0001 0	1	01	001
2	0010	0010	00110	00101	0010 0		11	011
3	0011	0011	01001	0011 0	0011 1		10	010
4	0100	0100	01010	0100 1	0100 0			110
5	0101	1011	01100	0101 0	0101 1			111
6	0110	1100	10001	0100 1	0110 0			101
7	0111	1101	10010	0111 1	0111 0			100
8	1000	1110	10100	1000 1	1000 0			
9	1001	1111	11000	1001 0	1001 1			

Một số mã giúp cho quá trình giải toán được thuận tiện hơn. Mã 2421 có thể được dùng để biểu diễn hệ thập phân. Mã 2421 có ưu điểm là mã cho phần bù cơ số rút gọn cũng giống như phần bù cơ số rút gọn của mã và điều này không đúng trong mã BCD. Do đó mã 2421 thuận tiện hơn trong việc nhân các nhóm kỷ tự đã được tách riêng thành từng cụm.

Người ta còn đưa ra các mã nhằm phát hiện lỗi xảy ra trong lưu trữ cũng như trong khi thu phát. Ví dụ như các mã chẵn lẻ và mã 2 trong 5 được biểu diễn trong bảng 36.13. Mã phát hiện lỗi 2 trong 5 là mã mà mỗi giá trị thập phân đều có chính xác hai giá trị trọng số. Mã chẵn lẻ được gắn thêm một bit có giá trị sao cho tổng số các trọng số là lẻ (nếu mã lẻ được sử dụng) và tổng số các trọng số có giá trị chẵn (nếu mã chẵn được sử dụng). Mã chẵn lẻ đơn bit không có khả năng phát hiện một số chẵn các bit lỗi. Do đó, các mã chẵn lẻ đơn bit chỉ phù hợp khi số bit lỗi đủ thấp. Nếu sử dụng các mã chẵn lẻ có số bit đủ lớn thì có thể phát hiện và sửa chữa tất cả các lỗi.

Một số mã được thiết kế để ngăn chặn việc biểu diễn sai kết quả đo do những lỗi nhỏ của các cảm biến. Các mã Gray được sử dụng cho mục đích này. Mã Gray là mã mà các vị trí liền kề về mặt vật lý cũng liền kề về lôgic, chúng chỉ khác nhau ở một kỷ tự nhị phân. Mã Gray có thể được tạo thành với số lượng bất kỳ các bit bằng cách biểu diễn mã Gray theo các mã Gray có ít hơn một

bit, ví dụ như trong hình 36.13 với các trường hợp 1, 2 và 3 bit. Ưu điểm của việc mã hóa cảm biến vị trí tuyến tính bằng mã Gray được minh họa trong hình 36.1 cho trường hợp 8 phần.



HÌNH 36.1 Cảm biến vị trí 8-phần với tiếp xúc không thẳng hàng: (a) cấu hình vật lý mã nhị phân, (b) cấu hình vật lý mã Gray

36.9 Đại số Boolean

Đại số Boolean cung cấp phương tiện để phân tích và thiết kế các hệ nhị phân và dựa trên cơ sở 7 định đề toán học trong bảng 36.14. Tất cả các quan hệ khác trong đại số Boolean được lấy ra từ 7 định đề này. Các định đề này được biểu diễn theo dạng hình học gọi là *sơ đồ Venn* và chúng xuất hiện nhiều hơn trong tự nhiên và trong lôgic. Lợi ích này thu được từ việc thể hiện hình ảnh hai chiều, giải phóng việc thể hiện theo một chiều bởi định dạng ngôn ngữ tuyến tính.

BẢNG 36.14 Các định đề Boolean

Định đề	Tên	Ý nghĩa	Dạng (a)	Dạng (b)
1	Định nghĩa	∃ một tập $\{K\}=\{a,b,\ldots\}$ của hai hoặc nhiều phần tử và hai toán tử nhị phân	OR	AND
			+	·
		∋ $\{K\}=\{a,b,a+b,a\cdot b,\ldots\}$	∨	∧
			∪	∩
2	Luật thay thế	Biểu thức$_1$=Biểu thức$_2$ Nếu một biểu thức được thay thế bằng biểu thức khác thì giá trị không thay đổi		
3	Phần tử đồng nhất	∃ các phần tử đồng nhất cho mỗi toán tử	$a+0=a$	$a\cdot 1=a$
4	Giao hoán	Cho mọi a và b trong K	$a+b=b+a$	$a\cdot b=b\cdot a$
5	Kết hợp	Cho mọi a,b và c trong K	$a+(b+c)=(a+b)+c$	$a\cdot(b\cdot c)=(a\cdot b)\cdot c$
6	Phân bố	Cho mọi a,b và c trong K	$a+(b\cdot c)=(a+b)\cdot(a+c)$	$a\cdot(b+c)=(a\cdot b)+(a\cdot c)$
7	Phần bù	Cho mọi a trong K ∃ một phần bù trong K ∋	$a+\bar{a}=1$	$a\cdot\bar{a}=0$

BẢNG 36.15 Định lý Boolean

Định lý		Dạng (a)	Dạng (b)
8	Không thay đổi giá trị	$a+a=a$	$a \Diamond a = a$
9	Định lý phần bù	$a+1=1$	$a \Diamond 0 = 0$
10	Sự hấp thụ	$a+ab=a$	$a(a+b)=a$
11	Khử phần tử thừa	$a+\bar{a}b=a+b$	$a(\bar{a}+b)=ab$
12	Định lý De Morgan Concesus	$a+b=\bar{a} \Diamond \bar{b}$	$\overline{ab}=\bar{a}+\bar{b}$
13	Concesus	$ab+\bar{a}c+bc=ab+\bar{a}c$	$(a+b)(\bar{a}+c)(b+c)=(a+b)(\bar{a}+c)$
14	Định lý phần bù 2	$ab+a\bar{b}=a$	$(a+b)(a+\bar{b})=a$
15	Concesus 2	$ab+a\bar{b}c=ab+ac$	$(a+b)(a+\bar{b}+c)=(a+b)(a+c)$
16	Concesus 3	$ab+\bar{a}c=(a+c)(\bar{a}+b)$	$(a+b)(\bar{a}+c)=ac+\bar{a}b$

Các toán tử OR và AND bình thường được biểu diễn bởi các ký hiệu toán tử thuật toán + và ◊ và được hiểu như là tổng và tích số các toán tử trong lôgic số cơ bản. Tuy nhiên, trong các hệ thống số thực hiện các phép tính thuật toán thì ký hiệu này không rõ ràng và các ký hiệu ∨ cho OR và ∟ cho AND loại bỏ sự không rõ ràng giữa thuật toán và các toán tử boolean. Ý nghĩa của các toán tử boolean này được hiểu một cách tốt nhất dựa trên khái niệm tập hợp. Tập này sử dụng toán tử hợp ∪ cho OR và toán tử giao ∩ cho AND.

Một tập các định lý được lấy ra từ các định đề cho phép các phát triển sau này được thực hiện dễ hơn. Các định lý được tổng hợp trong bảng 36.15. Việc sử dụng các định đề được minh họa bằng sự kiểm chứng một định lý trong hình 36.2.

36.10 Các hàm Boolean

Các hàm Boolean có thể được định nghĩa và biểu diễn theo các biểu thức boolean và bảng chân lý như minh họa trên hình 36.3(a, c). Ta có thể chuyển từ dạng này sang dạng khác. Các giá trị hàm để xây dựng bảng chân lý có thể thu được bằng cách phân tích các hàm như minh họa trong hình 36.3(b).

BẢNG 36.16 Số của các hàm Boolean khác nhau

Các biến	Các đối số	Các hàm
n	2^2	2^{2^n}
0	1	2
1	2	4
2	4	16
3	8	256
4	16	65.536
5	32	4.194.304
⋮	⋮	⋮

Với một số các biến cho trước sẽ có một số giới hạn các hàm boolean. Do mỗi biến boolean có thể có hai giá trị 0 hoặc 1 nên một tập n biến sẽ có 2^n giá trị khác nhau. Hàm boolean có một giá

trị xác định đối với các giá trị mà các biến độc lập có thể có. Do các biến độc lập có thể có hai giá trị nên sẽ có 2^{2^n} hàm boolean khác nhau của n biến. Số lượng các hàm tăng lên rất nhanh khi số biến độc lập tăng, như minh họa trong bảng 36.16.

1)	$x + x = x + x$	Nhận dạng
2)	$= (x + x).1$	P3b Sự tồn tại của phần tử nhận dạng
3)	$= (x + x).(x + \bar{x})$	P7a Sự tồn tại của phần bù
4)	$= x + x.\bar{x}$	P6a Sự phân bố
5)	$= x + 0$	P7b Sự tồn tại của phần bù
6)	$= x$	P3a Sự tồn tại của phần tử nhận dạng

HÌNH 36.2 Chứng minh định lý 8: Không thay đổi giá trị (a): x + x = x

(a) $f(A, B, C) = AB + A\bar{C} + \bar{A}C$

(b) $f(0, 0, 1) = 0.0 + 0.\bar{1} + \bar{0}.1$
$= 0.0 + 0.0 + 1.1$
$= 0 + 0 + 1$
$= 1$

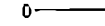
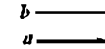
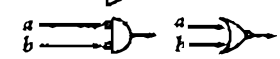
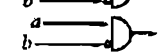
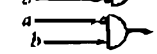
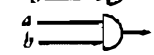
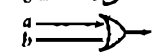
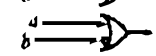
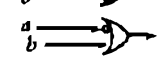
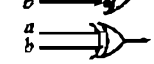
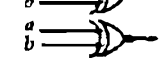
(c)

A	B	C	f(A, B, C)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

HÌNH 36.3 Ví dụ định nghĩa các hàm Boolean: (a) Định nghĩa biểu thức Boolean, (b) phân tích biểu thức Boolean, (c) định nghĩa bảng chân lý

Mười sáu hàm boolean khác nhau của hai biến độc lập được định nghĩa dưới dạng toán học trong bảng 36.17 và dạng bảng chân lý trong bảng 36.18.

BẢNG 36.17 Các hàm hai biến được định nghĩa như các biểu thức Boolean

	Tên	Biểu thức	Mạch biểu diễn
0	LUÔN LUÔN	1	
1	KHÔNG BAO GIỜ	0	
2	BIẾN 1	a	
3	BIẾN 2	b	
4	PHỦ ĐỊNH BIẾN 1	$\bar{a}$	
5	PHỦ ĐỊNH BIẾN 2	$\bar{b}$	
6	MIN-0/NOR	$\bar{a} \wedge \bar{b} = a \downarrow b$	
7	MIN-1	$a \wedge \bar{b}$	
8	MIN-2	$\bar{a} \wedge b$	
9	MIN-3/AND	$a \wedge b$	
10	MAX-0/OR	$a \vee b$	
11	MAX-1	$a \vee \bar{b}$	
12	MAX-2	$\bar{a} \vee b$	
13	MAX-3/NAND	$\bar{a} \vee \bar{b} = a \uparrow b$	
14	HỘI	$A \oplus b = a \wedge \bar{b} \vee \bar{a} \wedge b$	
15	TUYỂN	$a \Theta b = \overline{a \oplus b}$	

36.11 Các mạch cổng

Có thể tạo thành các hàm boolean từ các mạch số. Các mạch thực hiện các hàm boolean phức tạp có thể được chia nhỏ thành các mạch đơn giản hơn, thực hiện các hàm boolean đơn giản hơn. Các mạch thực hiện các hàm boolean đơn giản nhất gọi là các phần tử cơ bản hay còn được gọi là *các cổng* và được biểu diễn bằng các ký tự đặc biệt. Các cổng thực hiện các hàm với hai biến độc lập được biểu diễn trong bảng 36.17. Các cổng được xác định dựa vào tính chất kết quả phép toán mà chúng thực hiện. Các cổng cơ bản nhất là AND, OR, NAND, NOR, XOR và COIN. Cổng một đầu vào hay được dùng là cổng đảo hay còn gọi là cổng NOT. Các cổng là các phần tử cơ bản giúp chúng ta xây dựng các mạch lôgic số phức tạp hơn.

BẢNG 36.18 Bảng chân lý cho các hàm hai biến

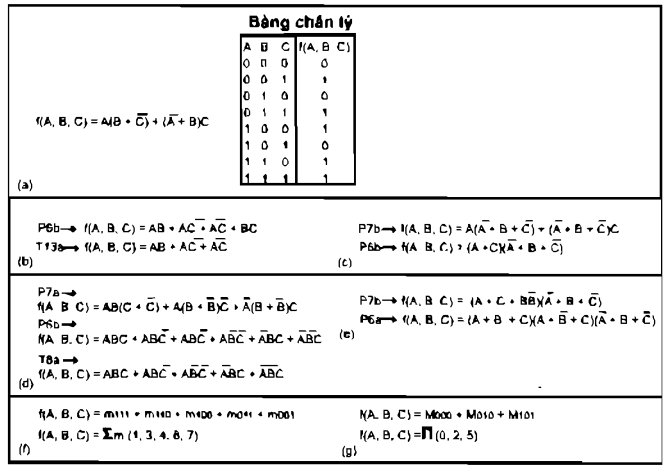
		NOR			AND	OR			NAND							
a	b	m_0	m_1	m_2	m_3	M_0	M_1	M_2	M_3	XOR	COIN	A	B	$\bar{a}$	$\bar{b}$	LO	HI
0	0	1	0	0	0	0	1	1	1	0	1	0	0	1	1	0	1
0	1	0	1	0	0	1	0	1	1	1	0	0	1	1	0	0	1
1	0	0	0	1	0	1	1	0	1	1	0	1	0	0	1	0	1
1	1	0	0	0	1	1	1	1	0	0	1	1	1	0	0	0	1

Một mạch lôgic có đầu ra ổn định chỉ phụ thuộc vào đầu vào ổn định (và không phụ thuộc vào các tín hiệu vào trước đó) được gọi là *mạch lôgic tổ hợp*. Để có thể phụ thuộc vào các tín hiệu đầu vào trước đó thì cần phải có bộ nhớ, nhưng ở đây các mạch lôgic tổ hợp lại không có các phần tử nhớ.

Đại số Boolean cho phép xây dựng bất kỳ một mạch lôgic tổ hợp nào dựa trên các phần tử cổng cơ bản là AND, OR và NOT. Ta cũng có thể xây dựng bất kỳ một mạch lôgic tổ hợp nào dựa trên phần tử NAND cũng như phần tử NOR.

36.12 Các dạng khai triển

Tổng của các tích (SP) là một dạng cơ bản, nó có thể chứa tất cả các hàm boolean. Tích của các tổng (PS) lại là một dạng khác mà nó cũng có thể chứa tất cả các hàm boolean. Các ví dụ minh họa được đưa ra trong hình 36.4 (b,c) là khai triển của ví dụ trong hình 36.4(a).



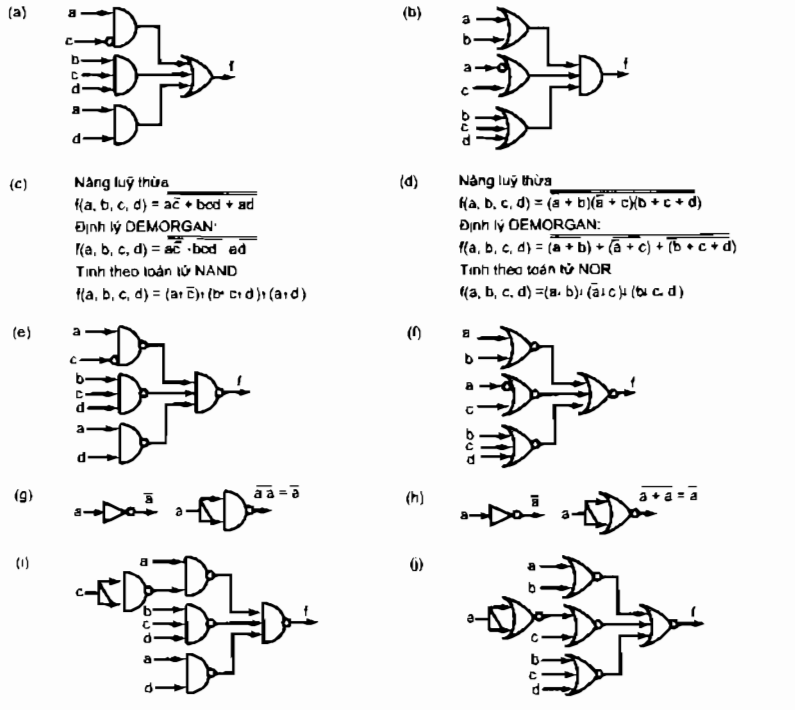
HÌNH 36.4 Chuyển đổi giữa các dạng của hàm Boolean: (a) ví dụ cho trước, (b) dạng SP, (c) dạng PS, (d) dạng SP quy chuẩn, (e) dạng PS quy chuẩn, (f) dạng SP tối giản quy chuẩn/tối giản ký hiệu, (g) dạng PS cực đại quy chuẩn/cực đại ký hiệu

Các tối giản là một tập đặc biệt các hàm. Mỗi tối giản có các biến ở dạng bù hoặc không bù kết hợp với nhau bởi phần tử AND. Khai triển SP chỉ có các tối giản được gọi là khai triển SP quy chuẩn. Hình 36.4(d) minh họa khai triển SP quy chuẩn cho những ví dụ trước đó. Ta có thể biểu diễn khai triển SP quy chuẩn một cách đơn giản bằng cách liệt kê các phần tử tối giản như trong hình 36.4(f). So sánh bảng chân lý với dạng khai triển các hàm tối giản chỉ ra rằng mỗi giá trị 1 của hàm biểu diễn một tối giản của hàm và ngược lại. Tất cả các giá trị hàm khác là 0.

Các cực đại có tính chất giống như các tối giản nhưng chúng được hình thành bởi các phần tử OR. Khai triển PS chỉ có các cực đại trong đó được gọi là khai triển PS quy chuẩn, như hình 36.4(e). Chúng cũng có thể được biểu diễn đơn giản bằng cách liệt kê các phần tử cực đại như trong hình 36.4(g). So sánh bảng chân lý với dạng khai triển các hàm cực đại chỉ ra rằng mỗi giá trị 0 của hàm biểu diễn dạng cực đại của hàm và ngược lại. Tất cả các giá trị hàm khác là 1.

36.13 Thực hiện

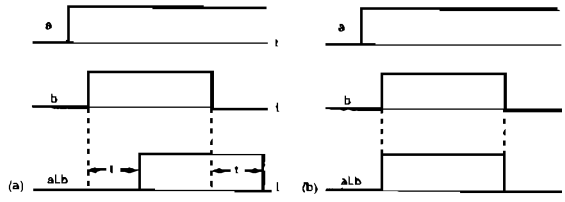
Các dạng khác nhau của khai triển boolean cung cấp các mạch khác nhau để thực hiện các hàm. Một hàm biểu diễn theo dạng SP được thực hiện trực tiếp bằng các phần tử AND–OR như trong hình 36.5(a). Một hàm biểu diễn theo dạng PS được thực hiện trực tiếp bằng các phần tử OR–AND như trong hình 36.5(b).



HÌNH 36.5 Ví dụ thực hiện các hàm dựa trên các dạng khai triển:

(a) thực hiện hàm $f(a,b,c,d) = a\bar{c} + bcd + ad$ bằng AND–OR, (b) thực hiện hàm $f(a,b,c,d) = (a+b)(\bar{a}+c)(b+c+d)$ bằng OR–AND, (c) biến đổi AND-OR thành NAND–NAND, (d) biến đổi OR–AND thành NOR–NOR, (e) thực hiện hàm $f(a,b,c,d) = a\bar{c} + bcd + ad$ bằng NAND–NAND, (f) thực hiện hàm $f(a,b,c,d) = (a+b)(\bar{a}+c)(b+c+d)$ bằng NOR–NOR, (g) thực hiện cổng NOT bằng cổng NAND, (h) thực hiện cổng NOT bằng cổng NOR, (i) thực hiện của hàm $f(a,b,c,d) = a\bar{c} + bcd + ad$ bằng NAND–NAND–NAND, (j) thực hiện hàm $f(a,b,c,d) = (a+b)(\bar{a}+c)(b+c+d)$ bằng NOR–NOR–NOR

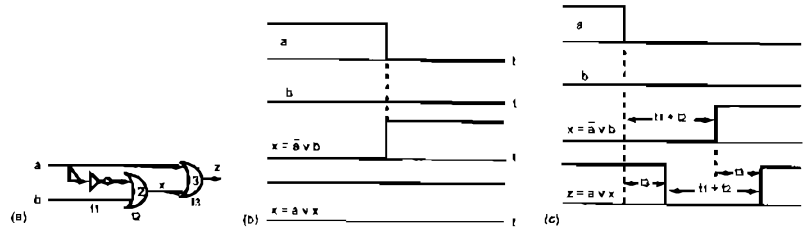
Bằng cách sử dụng các biến đổi luỹ thừa và định lý deMorgan, ta có thể khai triển dạng SP theo các phần tử NAND–NAND và dạng PS theo các phần tử NOR–NOR như trong hình 36.5(c,d). Các giá trị đảo ngược ở đầu vào có thể được cung cấp từ các phần tử NAND hoặc NOR, như hình 36.5(g,h), sau đó chúng tạo thành các mạch NAND–NAND–NAND và NOR–NOR–NOR như trong hình 36.5(i, j).



HÌNH 36.6 Lược đồ thời gian cho mạch cổng NAND: (a) lược đồ thời gian nhỏ, (b) lược đồ thời gian lớn

36.14 Các lược đồ thời gian

Lược đồ thời gian có hai dạng chính là *lược đồ thời gian nhỏ* và *lược đồ thời gian lớn*. Lược đồ thời gian nhỏ có phạm vi thời gian mở rộng đủ để hiển thị rõ ràng độ trễ của cổng, như trong hình 36.6(a) cho một cổng AND. Lược đồ thời gian lớn có phạm vi thời gian co lại đủ nhỏ để độ trễ của cổng là không đáng kể, như trong hình 36.6(b) cho một cổng AND.



HÌNH 36.7 Ví dụ về lỗi (của đầu ra) do độ trễ không đều của các đầu vào: (a) mạch minh họa lỗi, (b) trường hợp lý tưởng, không trễ, $\tau_1 = \tau_2 = \tau_3 = 0$, không có lỗi ở đầu ra, (c) các đầu vào với độ trễ khác nhau, $\tau_1 + \tau_2 + \tau_3 > \tau_3$, có lỗi

Ưu điểm của lược đồ thời gian lớn là những khoảng thời gian lớn hơn có thể được biểu diễn trong một kích cỡ không gian cho phép và chúng có thể được phát triển nhanh hơn. Nhưng nhược điểm của nó là không thể hiện được thông tin cần thiết để kiểm tra về giới hạn tốc độ.

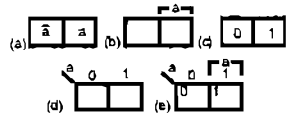
36.15 Rủi ro

Sự biến thiên thời gian trễ của các tín hiệu khi chúng qua các phần tử mạch khác nhau trong các đầu vào khác nhau có thể khiến các tín hiệu ra thay đổi bất thường so với dự đoán của bảng chân lý không phụ thuộc vào thời gian cho các phần tử. Sự thay đổi bất thường có thể gây ra một kết quả không mong muốn và đó chính là một *rủi ro*. Điều này được minh họa trong hình 36.7.

36.16 Các dạng giản đồ K

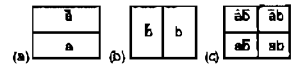
Ở bảng chân lý các giá trị của một hàm boolean được biểu diễn theo dãy một chiều. Một **giản đồ *K*** chứa đựng thông tin tương tự được sắp xếp theo số chiều tương ứng với các biến độc lập có trong một hàm. Cách biểu diễn đặc biệt như vậy cung cấp một thủ tục đơn giản để tối giản hóa các biểu thức và do đó giới hạn số lượng các thành phần cần thiết cho việc thực hiện một hàm theo

dạng đã cho. Hàm biểu diễn trong không gian của biểu đồ Venn được gọi là *một tập vạn năng*. Giản đồ *K* là một dạng đặc biệt của biểu đồ Venn. Không gian được chia làm hai phần khác nhau đối với mỗi biến độc lập. Với mỗi biến độc lập, không gian được chia làm hai vùng có kích thước đồng nhất, mỗi vùng đại diện cho một tối giản của hàm. Với n biến độc lập ta sẽ có 2^n vùng có kích thước đồng nhất, mỗi vùng đại diện cho một tối giản trong 2^n tối giản của hàm. Điều này được minh họa trong các hình từ 36.8 – 36.15.

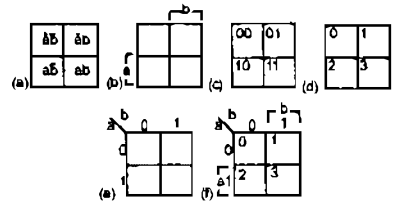


HÌNH 36.8 Các dạng giản đồ *K* một biến: (a) nhãn tối giản bên trong, (b) nhãn ở ngoài, (c) nhãn số tối giản bên trong, (d) nhãn tỷ lệ bên ngoài, (e) nhãn hỗn hợp

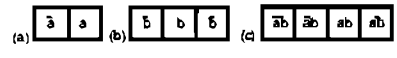
Hình 36.8 mô tả dạng giản đồ *K* một biến. Hình 36.8(a) minh họa việc chia không gian thành hai vùng bằng nhau, mỗi vùng dành cho một trong hai tối giản có thể của mỗi biến độc lập. Ta cũng có thể xác định các ô vuông bằng cách đặt các biến ở ngoài khoảng trống để định rõ vùng thuộc phạm vi của biến, cùng với khoảng trống chưa định tên thuộc phạm vi phần bù của biến, như trong hình 36.8(b). Cách khác trong việc xác định các khu vực đó là dùng số tối giản cho các vùng, như hình 36.8(c). Một cách khác nữa là đặt các giá trị có thể có của biến theo một tỷ lệ dọc theo khoảng trống như trong hình 36.8(d). Nhãn hỗn hợp, như hình 36.8(e), có vẻ dư thừa nhưng chúng rất hữu ích vì các dạng khác nhau có thể sử dụng với các loại nhãn khác nhau tuỳ theo từng cách hiểu. Cách đặt các biểu thức tối giản vào trong mỗi hình vuông là quá lộn xộn và ít khi được sử dụng ngoại trừ mục đích giảng dạy về lý thuyết cho giản đồ K. Mặc dù được sử dụng rộng rãi nhưng các số tối giản thường làm lộn xộn lược đồ và phương pháp luận được đưa ra ở đây sẽ là không cần thiết khi các khái niệm được trình bày rõ ràng.



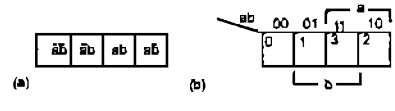
HÌNH 36.9 Cấu trúc của giản đồ *K* hai biến: (a) các vùng cho a, (b) các vùng cho b, hỗn hợp



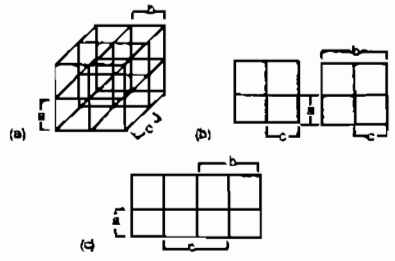
HÌNH 36.10 Các dạng giản đồ *K* hai biến: (a) nhãn tối thiểu bên trong, (b) nhãn ở ngoài, (c) nhãn số tối giản bên trong, (d) nhãn tỷ lệ bên ngoài, (e) nhãn hỗn hợp



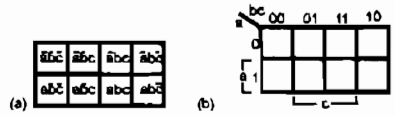
HÌNH 36.11 Cấu trúc khác của giản đồ *K* hai biến: (a) các vùng cho a, (b) các vùng cho b, (c) hỗn hợp



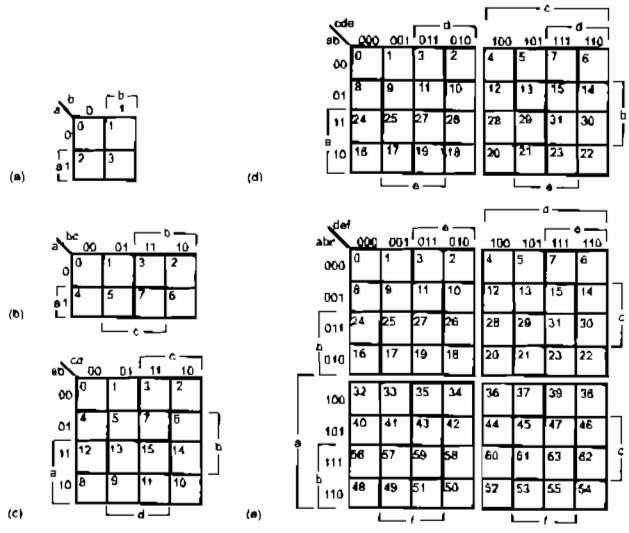
HÌNH 36.12 Cấu trúc khác của giản đồ *K* hai biến: (a) nhãn tối giản, (b) nhãn hỗn hợp



HÌNH 36.13 Các dạng giản đồ K ba biến: (a) ba chiều, (b) hai nửa trái, phải 3 chiều, (c) hai chiều



HÌNH 36.14 Các dạng giản đồ *K* hai chiều ba biến: (a) nhãn tối giản, (b) nhãn tỷ lệ hỗn hợp



HÌNH 36.15 Các giản đồ *K* cho các hàm từ 2-6 biến độc lập: (a) trường hợp 2 biến *f(a, b)*, trường hợp ba biến *f(a, b, c)*, (c) trường hợp 4 biến *f(a, b, c, d)*, (d) trường hợp năm biến *f(a, b, c, d, e)*, trường hợp 6 biến *f(a, b, c, d, e, f)*

Việc sắp xếp giản đồ *K* dạng hai biến được minh họa trong hình 36.9. Không gian được chia theo cột thành hai vùng cho biến a và phần bù của nó và được chia nhỏ theo hàng cho biến b và phần bù của nó, như hình 36.9(a,b). Các vùng hỗn hợp cho hai biến và các biểu thức tối giản hai biến được mô tả trong hình 36.9(c). Các phương pháp khác nhau để xác định các vùng cho trường hợp hai biến được mô tả trong hình 36.10. Điều cần chú ý ở đây là sự so sánh giữa nhãn số tối giản trong hệ nhị phân và hệ thập phân. Số tối giản nhị phân thì đơn giản trong việc nối ghép giữa tỷ lệ cột và hàng cho các vị trí. Cách xác định này làm các nhãn trong các hình vuông trở nên không cần thiết.

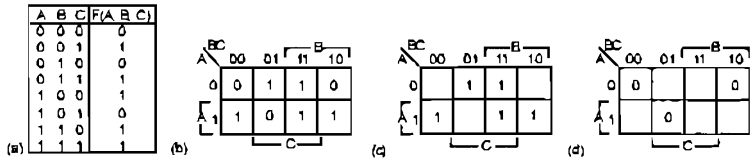
Một cách phân chia khác cho trường hợp hai biến được minh họa trong hình 36.11 và cách đánh nhãn được minh họa trong hình 36.12. Cấu hình trong hình 36.9 sử dụng hai chiều cho hai biến, trong khi đó cấu hình trong hình 36.11 sử dụng một chiều cho hai biến. Cấu hình hai chiều lôgic hơn và thuận tiện hơn so với cấu hình một chiều cho trường hợp hai biến. Trong trường hợp có nhiều biến hơn thì ta phải dùng cấu hình đặc biệt khác.

Giản đồ *K* ba biến được minh họa trong hình 36.13. Sẽ hợp lý hơn nếu đưa thêm một chiều không gian cho mỗi biến độc lập như mô tả trong hình 36.13(a). Tuy nhiên những nhược điểm khi làm việc với dạng này khiến chúng trở nên không thực tế. Để tạo có thể ứng dụng trong thực tế thì chúng phải được chuyển sang dạng hai chiều. Có thể thực hiện điều này theo hai cách. Cách thứ nhất là tách cấu hình 3D thành các lát cắt riêng lẻ và đặt chúng liền kề nhau như minh họa trong hình 36.13(b). Cách thứ hai là sử dụng cấu hình một chiều cho hai biến như trong hình 36.13(c). Trong trường hợp có 3 đến 4 biến độc lập thì dạng đưa ra như trong hình 36.12 có nhiều ưu điểm hơn và nó cũng được sử dụng thuận tiện trong trường hợp có nhiều biến hơn nữa. Hình 36.15 minh họa các dạng trên. Hình 36.14 minh họa việc gán nhãn cho 3 biến độc lập.

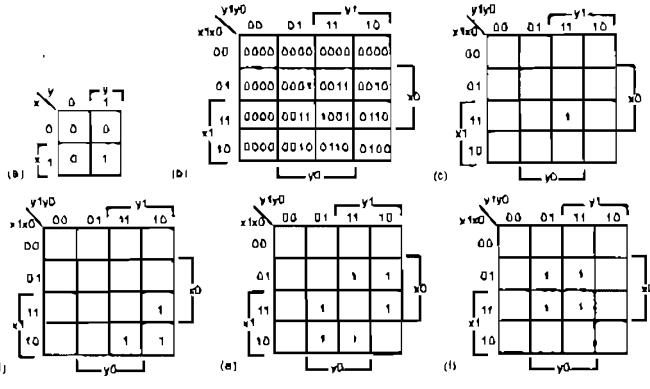
Các biến boolean độc lập trong giản đồ *K* có thứ tự chính xác như trong danh sách đối số của hàm boolean, như mô tả trong hình 36.15. Sắp xếp đúng thứ tự các biến độc lập theo giản đồ *K* làm cho tọa độ vị trí của một tối giản (hoặc một cực đại) đồng nhất với số tối giản (hoặc số cực đại). Với sự đồng nhất này việc sắp xếp các số xác định tối giản trong mỗi ô vuông hoặc sử dụng bảng nhận dạng vị trí là không cần thiết. Đặc điểm này làm giảm thời gian cần thiết để xây dựng một giản đồ *K* và tránh được các lỗi có thể xảy ra trong khi xây dựng. Số tối giản rất rõ ràng khi sử dụng các hệ nhị phân hay hệ thập phân.

36.17 Giản đồ *K* và tối giản hóa

Các hàm được chuyển về dạng giản đồ *K* bằng cách nhập giá trị cho mỗi tối giản trong khoảng trống dành cho tối giản đó. Các giá trị hàm có thể thu được theo nhiều cách như từ bảng chân lý, từ biểu thức Boolean hoặc từ các cách định nghĩa khác của hàm. Hình 36.4(a) là ví dụ sử dụng bảng chân lý, giản đồ *K* của nó được biểu diễn theo nhiều dạng trong các hình 36.16(b-d).



HÌNH 36.16 Giản đồ K 3 biến: (a) bảng chân lý, (b) giản đồ *K* với tất cả các giá trị, (c) giản đồ *K* dạng tối giản, (d) giản đồ *K* dạng cực đại



HÌNH 36.17 Giản đồ *K* dạng bảng hàm: (a) Giản đồ *K* cho tích của hai số dạng 1-b, $P = x * y$; (b) Giản đồ *K* cho tích của hai số 2-b, $P_3P_2P_1P_0 = x_1x_0 * y_1y_0$; (c) Giản đồ *K* cho thành phần P_3 của tích hai số 2-b; (d) giản đồ *K* cho thành phần *P*2 của tích hai số 2-b; (e) Giản đồ *K* cho thành phần *P*1 của tích hai số 2-b; (f) Giản đồ *K* cho thành phần *P*0 của tích hai số 2-b

Các hàm cũng có thể được chuyển về dạng giản đồ *K* một cách trực tiếp từ khai triển quy chuẩn, quá trình này cũng giống như việc nhập các tối giản (hoặc cực đại) từ bảng chân lý. Các

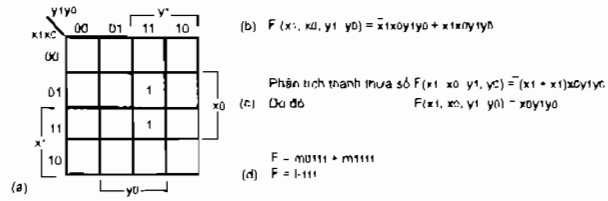
hàm có thể được chuyển về dạng giản đồ *K* từ bất kỳ dạng khai triển PS hay SP nào. Một cách khác để xây dựng giản đồ *K* là coi nó như một bảng hàm, như minh họa trong phép nhân các số có một chữ số và các số có hai chữ số trong hình 36.17.

36.18 Tối thiểu hóa với giản đồ K

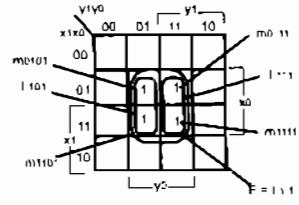
Đặc trưng quan trọng của giản đồ *K* khiến chúng rất thuận tiện trong quá trình tối thiểu hóa là các tối giản (liền kề không gian theo các chiều dọc và ngang) có tính chất liền kề logic. Dạng liền kề logic của các tối giản là đồng nhất trong tất cả các biến trừ một biến. Điều này cho phép kết hợp hai tối giản thành một với một biến ít hơn (minh họa trong hình 36.18). Hai tối giản liền kề kết hợp thành một liên kết bậc nhất. Một liên kết bậc nhất là sự kết hợp của tất cả các biến độc lập trừ một biến. Trong ví dụ này, liên kết bậc nhất biểu diễn theo các tối giản chứa 8 chữ nhưng biểu thức tối giản thì chỉ chứa có 3 chữ. Mạch thực hiện phép OR của hai tối giản có hai cổng AND và một cổng OR, trong khi đó để thực hiện liên kết bậc nhất tương đương chỉ cần một cổng AND.

Việc tổ hợp các tối giản thành một liên kết bậc nhất có thể được biểu diễn ngắn gọn hơn nữa bằng cách sử dụng các ký hiệu biểu diễn dạng tối giản đơn với các chỉ số để xác định dạng tối giản đặc biệt biểu diễn trong hệ nhị phân, minh họa trong hình 36.18(d).

Có thể kết hợp hai liên kết bậc nhất liền kề thành một liên kết bậc 2 như hình 36.19. Một liên kết bậc hai chứa tất cả các biến độc lập trừ hai biến. Nhìn chung một liên kết bậc *n* chứa tất cả các biến độc lập trừ *n* biến và cần một tập gồm 2^n tối giản.

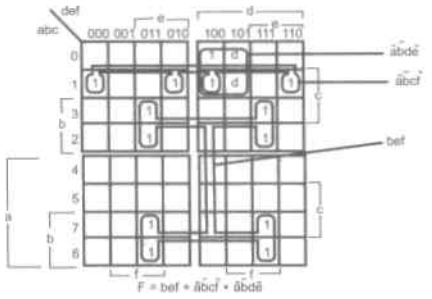


HÌNH 36.18 Ví dụ về tối giản hóa với giản đồ K: (a) giản đồ K, (b) biểu diễn theo dạng tối giản của định nghĩa hàm trong (a), (c) đơn giản hóa biểu thức trong (b), (d) đơn giản hóa biểu thức trong (b) bằng cách sử dụng dạng tối giản ký hiệu đơn và ký hiệu liên kết



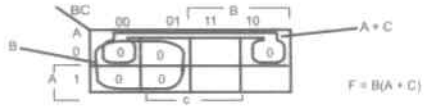
Hình 36.19 Ví dụ của tối giản hóa với giản đồ *K*

Các tối giản ở các cạnh đối diện trên cùng một cột hoặc một hàng là liền kề logic vì chúng chỉ khác nhau 1 biến. Nếu không gian mặt phẳng được cuộn tròn thành một hình trụ với các cạnh đối diện tiếp xúc với nhau, thì các cặp cạnh liền kề logic trở thành liền kề vật lý. Khi số lượng biến số là lớn, ta sử dụng giản đồ *K* với các màng song song, các vị trí tương đương trên các màng khác nhau là liền kề logic với nhau. Nếu các màng được coi là che phủ nhau thì các ô vuông tương đương là liền kề vật lý. Biểu thức tối giản thu được bao gồm các tối giản với số lượng ít nhất các liên kết lớn nhất. Một tối giản là một liên kết bậc không. Hình 36.20 minh họa nhiều ví dụ. *Giá trị bỏ qua* là một giá trị không bao giờ xảy ra hoặc nếu có xảy ra thì cũng không được sử dụng, vì vậy không phải quan tâm đến giá trị đó. Giá trị bỏ qua đôi khi cũng được biểu diễn để minh họa rằng chúng có thể được sử dụng để tối giản hóa các hàm bằng cách dùng giá trị của chúng để cực đại bậc của các liên kết.



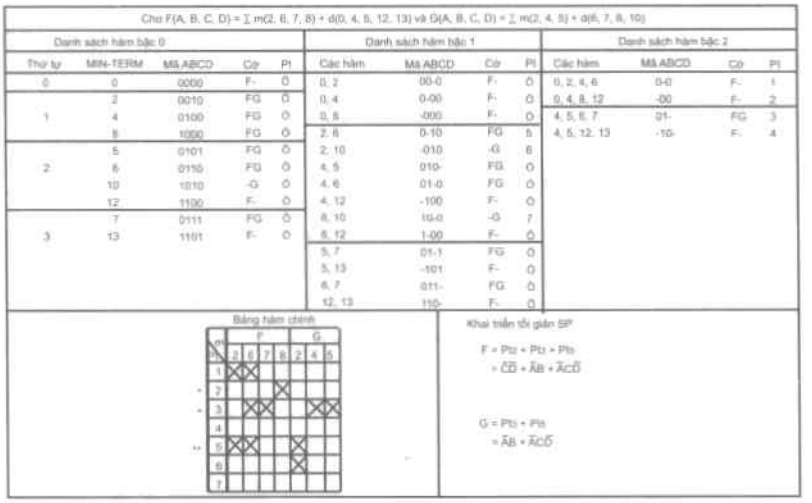
Hình 36.20 Ví dụ tối giản với giản đồ *K* 6 biến

Có thể sử dụng các giản đồ *K* dạng cực đại để thu được các biểu thức tối giản bằng cách kết hợp các dạng cực đại thành các liên kết bậc cao hơn, minh họa trong ví dụ hình 36.21.



HÌNH 36.21 Ví dụ giản đồ *K* dạng cực đại 3 biến

36.19 Tối giản hóa theo bảng Quine-McCluskey



HÌNH 36.22 Minh họa phương pháp Quine-McClusky cho trường hợp tối giản đồng thời

Tối giản hóa theo giản đồ *K* là quá phức tạp khi có từ 6 biến trở lên và rất khó để lập trình. Phương pháp bảng có thể thực hiện với bất kỳ số lượng biến và việc lập trình cho nó cũng dễ dàng hơn, bao gồm các bước sau:

1. Liệt kê tất cả các tối giản trong hàm boolean (với mã nhị phân) theo các nhóm có số lượng chữ số 1 như nhau. Các nhóm phải được liệt kê theo thứ tự liên tiếp của số lượng chữ số 1.
2. Xây dựng danh sách các liên kết bậc nhất. Sử dụng cờ để đánh dấu các tối giản, giá trị bỏ qua hoặc các liên kết đi kèm theo hàm nào (chỉ các dạng tối giản trong các nhóm liền kề có khả năng trở thành liền kề và do đó phương pháp thứ tự này đã giảm bớt công sức biên dịch các liên kết).
3. Xây dựng danh sách các liên kết bậc hai và liệt kê tất cả các liên kết bậc cao hơn, cho đến khi không còn liên kết cao hơn nữa được tạo ra.

4. Xây dựng *bảng liên kết chính*. Bảng liên kết biểu diễn liên kết chính bao phủ các tối giản của nó.
5. Lựa chọn số nhỏ nhất các liên kết chính lớn nhất bao phủ các tối giản.

Quá trình này được minh họa ở hình 36.2 cho trường hợp tối giản hóa đồng thời hai hàm boolean.

Các thuật ngữ

Cơ số: Số các giá trị khác nhau mà một con số có thể có.

Thập phân mã hóa theo nhị phân (BCD): Các con số theo hệ thập phân được biểu diễn ở dạng nhị phân.

Chuỗi ghép nối: Các ký tự nối với nhau để tạo thành một chuỗi lớn hơn, như là các ký tự trong một từ và các con số trong một số.

Mã: Chuyển đổi biểu diễn từ hệ thống bảng chữ cái này sang bảng chữ cái khác.

Phần bù: Đại lượng đạt được bằng cách lấy đại lượng lớn nhất trừ đi số đó trong một hệ thống số.

Giá trị bỏ qua: Giá trị có thể biểu diễn là tối giản hoặc cực đại

Mã gray: Là một tập hợp các mã có thuộc tính kề liền lôgic.

Liên kết: Liên kết bậc nhất là một cặp tối giản liền kề logic. Liên kết bậc hai là một tập hợp các liên kết bậc nhất liền kề logic.

Liền kề lôgic: Hai mã nào đó có cùng số lượng các con số mà giá trị của chúng chỉ khác nhau ở một con số.

Lược đồ thời gian lớn: Biểu diễn đồ họa thể hiện sự biến đổi của các dạng sóng theo thời gian nhưng với một thang đo thời gian không đủ độ phân giải để thể hiện độ trễ được tạo ra bởi các phần tử cơ sở riêng của mạch số.

Dạng cực đại: Là hàm của một tập hợp các biến Boolean mà sẽ chỉ cho một giá trị thấp với một tổ hợp các giá trị biến và cho giá trị cao đối với tất cả các tổ hợp khác của các giá trị biến.

Lược đồ thời gian nhỏ: Biểu diễn đồ họa thể hiện sự biến đổi của các dạng sóng theo thời gian nhưng với một thang đo thời gian đủ độ phân giải để thể hiện một cách rõ ràng độ trễ được tạo ra bởi các phần tử cơ sở riêng của mạch số.

Dạng tối giản: Là hàm của một tập hợp các biến Boolean sẽ chỉ cho một giá trị cao với một tổ hợp các biến và có giá trị thấp với tất cả các tổ hợp khác của các giá trị biến.

Tràn số: Là một phần của phép toán số tính toán ra kết quả ra không nằm trong trường được chỉ định.

Bít chẵn lẻ: Một bít thêm vào được tạo thành một chuỗi của một mã mà giá trị đưa ra thoả mãn sao cho tổng của các bít cao là chẵn đối với trạng thái chẵn và là lẻ đối với trạng thái lẻ.

Tích của các tổng (PS): Phép tổ hợp AND các đại lượng mà đã được tổ hợp theo luật OR từ các biến Boolean.

Số thực: Một số có cả phần phân số và phần số nguyên.

Tổng của các tích (SP): Phép tổ hợp OR của các đại lượng mà đã được tổ hợp theo luật AND từ các biến Boolean.

Bảng chân lý: Là một bảng các giá trị mà một hàm Boolean có thể có với các biến độc lập được xem xét như là một số nhiều cơ số và được sắp xếp theo thứ tự liên tiếp.

Tài liệu tham khảo

[1] Hayes, J.P. 1993. *Introduction of Digital Logic Design*. Addison-Wesley, Reading, MA.

[2] Humphrey, W.S., Jr. 1958. *Switching Circuits with Computer Applications*. McGraw-Hill, New York.

[3] Hill and Peterson. 1974. *Introduction to Switching Theory and Logical Design*, 2nd ed. Wiley, New York.

[4] Johnson and Karim. 1987. *Digital Design a Pragmatic Approach. Prindle*, Weber and Schmidt, Boston.

[5] Karnaugh, M. 1953. *The map method for synthesis of combinational logic circuits*. AIEE Trans. Comm.

[6] Elec. 72 (Nov.): 593–599.

[7] Mano, M.M. 1991. *Digital Design*. Prentice-Hall, Englewood Cliffs, NJ.

[8] McClusky, E.J. 1986. *Logic Design Principles*. Prentice-Hall, Englewood Cliffs, NJ.

[9] Mowle, F.J. 1976. *A Systematic Approach to Digital Logic Design*. Addison-Wesley, Reading, MA.

[10] Nagle, Carrol, and Irwin. 1975. *An Introduction to Computer Logic*, 2nd ed. Prentice-Hall, Englewood

[11] Cliffs, NJ.

[12] Pappas, N.L. 1994. *Digital Design* West, St. Paul, MN.

[13] Roth, C.H., Jr. 1985. *Fundamentals of Logic Design*, 3rd ed. West, St. Paul, MN.

[14] Sandige, R.S. 1990. *Modern Digital Design*. McGraw-Hill, New York.

[15] Shaw, A.W. 1991. *Logic Circuit Design*. Saunders, Fort Worth, TX.

[16] Wakerly, 1990. *Digital Design Principles and Practices*. Prentice-Hall, Englewood Cliffs, NJ.

37
Giao diện hệ thống

M.J. Tordon
The University of New South Wales

J. Katupitiya
The University of New South Wales

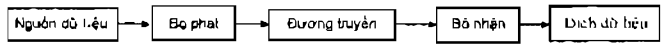
37.1 Các khái niệm cơ bản 37-1
37.2 Các chuẩn giao diện nối tiếp TIA/EIA 37-6
37.3 IEEE 488-Bus giao diện đa năng (GPIB) 37-9

Chương này trình bày các vấn đề liên quan đến giao diện nối tiếp không đồng bộ được mô tả bởi các chuẩn RS-232, RS-422, RS-485 và bus giao diện song song đa năng với chuẩn IEEE-488. Phần này cũng cung cấp những thông tin cơ bản, các thuật ngữ và thông số quan trọng cho việc thiết kế các giao diện hệ thống trong hệ cơ điện tử.

37.1 Các khái niệm cơ bản

Các hệ cơ điện tử hiện đại bao gồm một số hệ thống phụ dựa trên truyền thông dữ liệu số. Các mức độ phức tạp của các hệ thống này cho biết yêu cầu của truyền thông dữ liệu, từ dạng đơn giản giữa hai thiết bị đến các hệ thống có nhiều hệ phụ mà mỗi hệ thống phụ lại truyền thông trực tiếp hoặc gián tiếp với các hệ thống phụ khác bằng cách sử dụng một mạng truyền thông. Tuỳ thuộc vào khoảng cách của các hệ thống phụ mà các yêu cầu khác nhau sẽ được đặt ra cho các kênh truyền thông, cũng như việc thực hiện về mặt vật lí của các kênh và giao diện giữa các thiết bị. Hình 37.1 giới thiệu sơ đồ của một hệ truyền thông dữ liệu đơn giản giữa hai thiết bị.

Một nguồn dữ liệu sẽ tạo ra dữ liệu để truyền tới hệ thống đích và có thể chuyển đổi dữ liệu sang một dạng dữ liệu riêng. Hệ thống ban đầu thường không tạo ra dạng dữ liệu phù hợp với các đường truyền. Vì vậy bộ chuyển đổi phải biến đổi dữ liệu thành một dạng tín hiệu phù hợp với một kiểu đường truyền nhất định. Đường truyền thường được thực hiện bằng cách sử dụng mạng điện, có thể bao gồm các môi trường vật lý như: các tín hiệu tần số radio, hồng ngoại và âm thanh. Một đường truyền tín hiệu cung cấp một môi trường vật lý nối liền hai hệ thống. Một thiết bị thu sẽ nhận tín hiệu và biến đổi nó thành một dạng dữ liệu phù hợp để đi vào được hệ thống đích. Hệ thống đích sẽ xử lý dữ liệu để lấy lại thông tin gốc. Từ đó chúng ta thấy rằng ngay cả trong một hệ truyền thông đơn giản cũng có nhiều hệ thống phụ cùng tham gia vào nhiệm vụ truyền thông.



HÌNH 37.1 Sơ đồ của một hệ truyền dữ liệu đơn giản

Thuật ngữ và định nghĩa

Giao diện: Ranh giới chung giữa hai hệ thống phụ được gọi là giao diện, hình 37.1 cho thấy ngay trong một hệ truyền thông đơn giản cũng có nhiều giao diện.

Bit: Dạng đơn giản nhất của dữ liệu là một bit. Một bit có thể nhận giá trị 0 hoặc 1 và do đó được gọi là dạng nhị phân. Thông tin trong máy tính số đều được lưu trữ ở dạng nhị phân.

Byte: Một số lượng cố định các bit (thường là 8) có thể được máy tính coi như một đơn vị.

Ký tự: Từ trước đến nay, thông tin được thể hiện ở dạng ký tự. Một ký tự là một phần của một tập hợp ký tự. Một ví dụ về tập hợp ký tụ là bảng chữ cái trong tiếng Anh.

Mã ký tự: Mỗi ký tự trong một tập hợp ký tự được mã hóa trong các máy tính số dưới dạng các số nhị phân. Một trong những bộ mã ký tự được sử dụng rộng rãi nhất là ASCII (American Standard Code for Information Interchange).

Nối tiếp và song song

Đơn vị thông tin cơ bản được truyền giữa các hệ thống phụ thường là một ký tự. Đối với khoảng cách ngắn, người ta có thể sử dụng nhiều đường truyền song song để truyền đồng thời tất cả các bit của một ký tự. Đối với khoảng cách xa hơn, chi phí của các đường truyền dữ liệu thường cao hơn và các dữ liệu được chuyển thành nối tiếp để có thể truyền qua một đường truyền dữ liệu đơn như một chuỗi bit.

Tốc độ bit và tốc độ baud

Tốc độ truyền dữ liệu được biểu diễn bằng số bit dữ liệu được truyền trên một giây và gọi là tốc độ bit hiệu dụng, với đơn vị là bps, những đơn vị lớn hơn là kbps (1 kbps = 1.000 bps) và Mbps (1 Mbps = 1.000.000 bps) cũng được sử dụng phổ biến. Tốc độ baud là tốc độ tín hiệu được tính bằng số lần tín hiệu trên đường truyền thay đổi trạng thái trong một giây. Đối với các hệ thống chỉ sử dụng hai trạng thái, tốc độ bit tương đương với tốc độ baud. Trong truyền thông nối tiếp không đồng bộ, tốc độ bit truyền dữ liệu hiệu dụng có thể nhỏ hơn đáng kể so với tốc độ bit tín hiệu vì sự có mặt của các bit start (bắt đầu), stop (kết thúc) và parity (chẵn lẻ). Để tối đa hoá tốc độ truyền trên đường truyền nối tiếp, các hệ thống truyền thông hiện đại sử dụng các tín hiệu với nhiều hơn hai trạng thái, do đó đạt được tốc độ bit tín hiệu cao hơn. Ví dụ: Nếu sử dụng tín hiệu truyền 16 trạng thái thì tốc độ bit lớn hơn 4 lần so với tốc độ baud. Các thuật ngữ tốc độ baud, tốc độ bit tín hiệu, và tốc độ truyền dữ liệu hiệu dụng thường được sử dụng thay thế cho nhau nên rất dễ nhầm lẫn.

Đồng bộ và không đồng bộ

Với cả giao diện nối tiếp và song song, vấn đề đồng bộ phải được giải quyết. Việc truyền thông trên một đường truyền có thể được thực hiện theo chế độ đồng bộ hoặc không đồng bộ. Trong chế độ truyền đồng bộ việc truyền dữ liệu được đồng bộ với một đồng hồ, do đó việc truyền xảy ra tại các khoảng thời gian đều đặn. Bởi vì việc truyền dữ liệu xảy ra tại thời điểm xác định, nên không cần biết việc truyền thông được hoàn thành khi nào. Trong chế độ truyền không đồng bộ, hai hệ thống đang sử dụng các đồng hồ không được đồng bộ và có thể chạy ở các tần số khác nhau. Do đó đối với các hệ không đồng bộ, để dữ liệu truyền đúng yêu cầu cần có sự phối hợp riêng biệt gọi là bắt tay.

Điều khiển luồng dữ liệu

Một vấn đề khác trong các hệ truyền thông không đồng bộ đó là tốc độ xử lý dữ liệu. Nếu một hệ thống xử lý dữ liệu chậm thì phải điều khiển luồng dữ liệu để tránh mất dữ liệu. Điều khiển luồng dữ liệu cũng có thể yêu cầu thêm giao thức bắt tay. Các vấn đề tương tự cũng có thể xuất hiện trong các hệ thống đa nhiệm, do các nhiệm vụ khác, hệ thống không thể điều khiển dữ liệu đến trong chu kỳ có khối lượng công việc lớn.

Bắt tay

Để đảm bảo truyền dữ liệu hiệu quả mà không có lỗi, hệ thống gửi sẽ sử dụng một tín hiệu riêng biệt để chỉ ra rằng dữ liệu phù hợp đã được xuất hiện ở giao diện. Bởi vì có thể cùng lúc đó

tại thiết bị thu xử lý dữ liệu mà nó không biết, thiết bị truyền phải chờ một tín hiệu phản hồi trước khi truyền tiếp dữ liệu mới ra giao diện.Việc bắt tay có thể được thực hiện bằng phần cứng hoặc phần mềm.

Giao thức truyền thông

Hoạt động của các hệ thống truyền thông bị chi phối bởi một tập các quy luật đảm bao truyền dữ liệu tin cậy không lỗi và không bị mất dữ liệu. Một tập các quy luật như thế được gọi là một giao thức truyền thông.

Xử lý lỗi

Dữ liệu được truyền trên một đường truyền có thể bị nhiễu và do đó gây ra lỗi. Để duy trì tính toàn vẹn của dữ liệu, một số phương pháp đã được sử dụng. Một phương pháp đơn giản nhất là sau khi phát hiện lỗi truyền lại dữ liệu bị hỏng. Rất nhiều giao thức truyền thông tinh vi đòi hỏi các phương pháp sửa lỗi phức tạp được thực hiện tại mức giao thức.

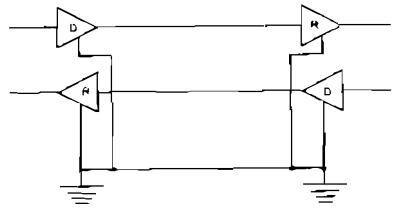
Truyền một chiều, hai chiều không đồng thời và hai chiều đồng thời

Theo cách đơn giản nhất, truyền thông có thể được thiết lập với một cặp dây đơn. Chế độ truyền dữ liệu, ở đó dữ liệu chỉ có thể truyền theo một hướng được gọi là truyền một chiều (simplex) hoặc kênh truyền đơn hướng. Trong hầu hết các ứng dụng, truyền thông được thực hiện theo cả hai hướng. Nếu chi phí của đường truyền dữ liệu cao, có thể tổ chức cho tín hiệu truyền theo cả hai hướng trên một đường truyền đơn bằng cách thêm mạch vào cả hai đầu cuối đường truyền nhưng chỉ theo một hướng tại một thời điểm. Chế độ truyền dữ liệu này được gọi là chế độ truyền hai chiều không đồng thời (half-duplex). Ngoài ra còn sử dụng, giao thức bắt tay để thực hiện chia sẻ thời gian của đường truyền.

Nếu tín hiệu có thể truyền theo cả hai hướng đồng thời trên một đường truyền thì chế độ truyền thông này được gọi là chế độ truyền hai chiều đồng thời (full-duplex). Một ví dụ của chế độ hai chiều đồng thời là đường dây điện thoại, ở đó hai kênh được tạo ra tại các băng tần khác nhau. Nếu kinh phí cho phép, có thể thiết lập hai đường truyền riêng biệt trong đó truyền thông hai chiều đồng thời được thực hiện trên hai kênh một chiều. Điều này yêu cầu nhân đôi tất cả các chức năng của một hệ thống truyền thông đơn giản như chỉ ra trong hình 37.1.

Truyền cân bằng và truyền không cân bằng

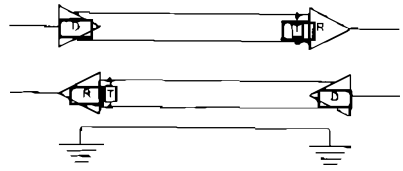
Việc thực hiện trên đường truyền điện có hai dạng cơ bản: cân bằng (single-ended) hoặc không cân bằng (differential). Đối với truyền không cân bằng, một đường dẫn đơn được sử dụng để mang các tín hiệu điện áp được tham chiếu với một tín hiệu đất. Đất thường là đường quay về chung cho tất cả các tín hiệu trên giao diện. Hình 37.2 chỉ ra một ví dụ của hệ thống truyền dữ liệu không cân bằng với hai kênh và ba dây.



HÌNH 37.2 Ví dụ truyền dữ liệu không cân bằng

Ký hiệu D (driver) biểu diễn bộ phát và R (receiver) biểu diễn bộ thu. Truyền dữ liệu không cân bằng thì không quá đắt bởi vì đối với các đường tín hiệu phức tạp chỉ cần duy nhất một đường truyền chung; tuy nhiên loại giao diện này dễ bị nhiễu cảm ứng và nhiễu đất, không thích hợp cho

việc truyền dữ liệu tốc độ cao với khoảng cách dài. Nhiễu đất liên quan đến việc giảm điện áp trong đường quay về chung, trong khi nhiễu cảm ứng xuất hiện do nhiễu trường điện từ. Cả hai loại nhiễu này đều là các nguồn nhiễu ngoài hoặc từ các mạch truyền dẫn bên cạnh. Một phương pháp giải quyết là có thể sử dụng cáp đồng trục, cáp bọc và/hoặc sử dụng các đường truyền ngược cho từng tín hiệu. Những cách giải quyết này có khuynh hướng làm tăng chi phí của giao diện.



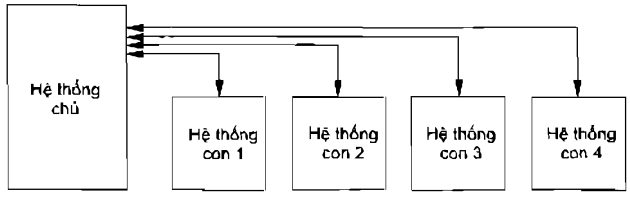
HÌNH 37.3 Ví dụ truyền dữ liệu cân bằng

Chế độ truyền cân bằng chống nhiễu tốt hơn nhiều so với chế độ truyền không cân bằng. Hai đường truyền bổ sung mang các tín hiệu dữ liệu. Sự thực hiện này thường bao gồm hai bộ phát sử dụng một đường truyền đôi dây xoắn. Hình 37.3 chỉ ra một ví dụ truyền dữ liệu cân bằng sử dụng hai kênh và 5 dây. Ký hiệu D biểu diễn bộ phát, R biểu diễn bộ thu, T là trở đầu cuối. Việc sử dụng một trở đầu cuối tại cuối bộ thu của đường truyền là mấu chốt của truyền thông tốc độ cao với khoảng cách dài vì các đường truyền không kết thúc có thể gây ra việc suy giảm tín hiệu. Cả nhiễu cảm ứng và nhiễu đất xuất hiện trên các đường dẫn như là các tín hiệu thông thường và bị loại bỏ bởi bộ thu vi sai. Các tín hiệu vi sai mang dữ liệu được khuyếch đại trong khi tín hiệu nhiễu bị loại bỏ. Do đó các đường truyền dữ liệu cân bằng có thể được sử dụng với khoảng cách xa và tốc độ truyền cao. Cả giao diện truyền cân bằng và không cân bằng được chỉ ra trong hình 37.2 và 37.3 thể hiện hai giao diện một chiều có thể tạo ra một kênh truyền thông điểm-điểm hai chiều toàn phần (xem truyền thông điểm-điểm ở phần sau).

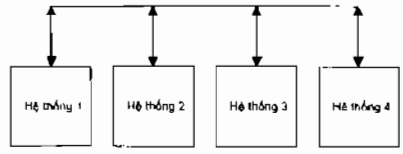
Các nguồn thông tin chi tiết về các bộ phát và bộ thu được cung cấp trong các tài liệu của nhà sản xuất bán dẫn [1], [2].

Truyền điểm-điểm và truyền đa điểm

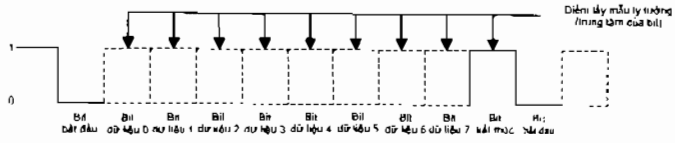
Nếu việc truyền dữ liệu xảy ra giữa hai thiết bị ta gọi đó là đường truyền điểm-điểm. Trong hệ cơ điện tử, thường yêu cầu hệ thống chủ liên lạc với một số hệ con. Nếu kinh phí cho phép thì một số đường truyền điểm-điểm có thể được thực hiện. Trong thiết lập điểm-điểm, hệ thống chủ có kết nối điểm-điểm tới mỗi hệ thống con riêng biệt, nghĩa là có một cổng và đường truyền riêng biệt cho mỗi hệ con. Kiểu thiết lập này được chỉ ra ở hình 37.4. Kết nối cũng có thể được thiết lập như là một kết nối đa điểm trong đó tất cả các thiết bị được kết nối tới một đường truyền đơn như hình 37.5. Đây là sự thiết lập một mạng truyền thông dữ liệu mà ở đó dữ liệu có thể được truyền từ bất kỳ một thiết bị nào đến bất kỳ một thiết bị khác trên mạng. Tất cả các thiết bị trên mạng phải có một bộ phát và một bộ thu. Bộ phát phải có 3 trạng thái (trở kháng đầu ra cao) để chúng không thể cung cấp thêm tải trên đường dây. Khi bộ phát không phát chúng được ngắt ảo kết nối khỏi đường truyền. Phương thức truyền phức hợp (complex) được yêu cầu để quản lý các bộ truyền riêng lẻ trên mạng. Ưu điểm chính của thiết lập đa điểm là chi phí thấp hơn so với kết nối truyền thông riêng biệt. Nhược điểm là phương thức truyền phức tạp hơn nhiều (phải nhận biết được những thiết bị nào đang truyền và đang nhận) và giao diện cũng phức tạp hơn.



HÌNH 37.4 Một hệ thống truyền thông điểm-điểm với 4 hệ con



HÌNH 37.5 Một hệ thống truyền thông đa điểm



HÌNH 37.6 Dạng dữ liệu nối tiếp không đồng bộ

Truyền thông nối tiếp không đồng bộ

Trong truyền thông nối tiếp không đồng bộ, dữ liệu được truyền tại các khoảng thời gian không đồng đều như là một dòng bit. Các ký tự riêng lẻ được mã hoá như các số nhị phân được chuyển thành luồng dữ liệu nối tiếp, được đóng khung với các bit start và stop. Một bit chẵn lẻ parity có thể được tùy chọn thêm vào dòng bit. Nói chung máy tính sẽ thể hiện thông tin dưới dạng song song chẳng hạn như là các byte và các word trong khi việc truyền thông với thiết bị ngoại vi chủ yếu là theo kiểu nối tiếp. Việc chuyển đổi song song thành nối tiếp và nối tiếp thành song song được thực hiện bởi một mạch tích hợp đặc biệt và được gọi là bộ thu phát không đồng bộ đa năng UART (universal asynchronous receiver transmitter) sẽ được miêu tả sau (hình 37.7).

Hình 37.6 đưa ra một dòng dữ liệu điển hình cho truyền không đồng bộ. Trong suốt thời gian không làm việc đường truyền ở trạng thái lôgic 1 (vì các lý do lịch sử nên cũng được gọi là 'MARK'). Điểm bắt đầu của một dòng dữ liệu luôn luôn được chỉ ra bởi bit start có giá trị lôgic 0 (cũng được gọi là "SPACE"). Theo sau bit start là 5 đến 8 bit dữ liệu thể hiện một ký tự. Tiếp theo là một bit parity tuỳ chọn. Dòng dữ liệu được kết thúc bởi 1 hoặc 2 bit stop với giá trị lôgic 1, theo sau có thể là đường truyền ở trạng thái không làm việc hoặc là bit start của một kỳ tự tiếp theo. Đường truyền ở trạng thái không làm việc tương ứng với lôgic 1. Bit parity là một bit được chèn thêm vào sau các bit dữ liệu và trước các bit stop. Nó được đặt theo các thông tin chẵn lẻ của dữ liệu trong dòng. Ví dụ, nếu trạng thái chẵn được sử dụng thì bit parity sẽ được đặt sao cho tổng số các bit 1 trong luồng dữ liệu bao gồm cả bit parity là chẵn. Bộ thu sẽ sử dụng bit parity để soát lỗi. Nhiệm vụ của bộ thu là phát hiện điểm bắt đầu của dòng dữ liệu và lấy mẫu chính xác từng bit riêng lẻ trong dòng. Sau khi phát hiện bit start, bộ thu phải lấy mẫu từng bit riêng lẻ, lý tưởng là tại điểm giữa của mỗi bit, như chỉ ra trong hình 37.6. Trong trường hợp lấy mẫu lý tưởng, như chỉ ra trong hình 37.6, bộ thu được cho là có sai số cho phép 50%. Trong thực tế, bộ thu của một UART lấy mẫu tín hiệu tới bằng cách sử dụng tần số của bộ phát tốc độ baud, là tần số cao hơn 16 lần so với tốc độ baud tương ứng được sử dụng cho việc truyền dẫn. Sự không chắc chắn trong việc phát hiện bit start sẽ giảm sai số cho phép đi từ 6,25% (1/16) tới 43,75% [3].

Ví dụ: Nếu đồng hồ bộ thu chậm hơn 1% so với đồng hồ của bộ phát tương ứng thì thời gian trích mẫu của bit dữ liệu đầu tiên sẽ bị trễ 1,5% thời gian của bit và thời gian trích mẫu của bit stop sẽ bị trễ 9,5%. Trong trường hợp này sai số cho phép có thể được giảm xuống 34,25%. Nếu đồng hồ bộ thu chậm hơn 5% thì bộ thu có thể phát hiện được bit start của kỳ tự tiếp theo thay vì bit stop của ký tự hiện tại. Điều này sẽ dẫn đến các lỗi khung. Ví dụ trên chỉ ra tầm quan trọng của độ chính xác của tốc độ đồng hồ và lý do tại sao dòng dữ liệu cần phải ngắn trong chế độ truyền không đồng bộ.

Các nhân tố khác ảnh hưởng đến chất lượng truyền thông bao gồm chiều dài và loại đường truyền, tốc độ truyền thông, các tham số của bộ phát, giới hạn của đường truyền và nhiễu trong hệ thống truyền thông.

Bộ thu phát không đồng bộ đa năng (UART)

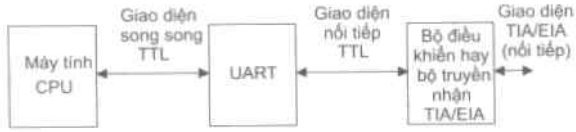
Chức năng cơ bản của UART là để thuận tiện biến đổi dữ liệu song song thành nối tiếp và nối tiếp thành song song. UART thường bao gồm một bộ phát và một bộ thu. Bộ thu và bộ phát có thể hoạt động đồng thời hoặc độc lập. UART có thể hoạt động ở chế độ hai chiều đồng thời hoặc hai chiều không đồng thời.

Dữ liệu song song từ máy tính được chuyển đổi thành dòng bit nối tiếp không đồng bộ. UART tự động thêm bit start, có thể một bit parity và một số lượng được lập trình các bit stop, rồi gửi dữ liệu tới chân (TxD) của đầu ra dữ liệu nối tiếp của bộ phát. Dữ liệu song song được chuyển đổi thành nối tiếp với bit có trọng số thấp nhất được dịch ra trước. Hình 37.7 chỉ ra một cách tổ chức điển hình cho UART. Như ta biết UART sử dụng giao diện tương thích TTL (transistor transistor lôgic). Bộ phát và bộ thu của đường truyền TIA/EIA (trình bày phần sau) được chỉ tới một giao diện đặc biệt, do đó thay đổi giao diện hệ thống có nghĩa là thay đổi môi trường truyền dẫn và các bộ thu và phát liên quan. Việc sử dụng UART là độc lập với môi trường truyền dẫn.

Dữ liệu nối tiếp được nhận ở chân (RxD) của đầu vào nối tiếp của bộ thu được chuyển thành dữ liệu song song. Trong quá trình xử lý, UART kiểm tra bit start, bit parity (nếu cần), các bit stop và thông báo mọi tình trạng lỗi. Chú ý rằng UART có khả năng tạo ra tất cả các tín hiệu được yêu cầu cho việc truyền thông nối tiếp không đồng bộ thành công.

UART cũng có thể thông báo một số tình trạng lỗi, bao gồm sự tràn bộ thu, lỗi parity, lỗi khung và lỗi ngắt. Lỗi tràn bộ thu xảy ra khi các byte được nhận nhanh hơn so với máy tính xử lý chúng. Lỗi parity được chỉ ra nếu tính chẵn lẻ của dòng bit thay đổi trong quá trình truyền thông. Lỗi khung được thông báo nếu bit stop được lấy mẫu không phải tại mức lôgic 1. Lỗi ngắt xảy ra nếu đường truyền không được dùng trong khoảng thời gian tương đương với khoảng của ít nhất một ký tự.

Các kiểu thiết bị UART cũ như như 8250, 16450 chỉ có một byte bộ đệm FIFO (First In First Out) và do đó dễ dàng bị tràn bộ đệm của bộ thu. Các thiết bị ngày nay có bộ đệm lớn hơn cung cấp khả năng truyền thông hiệu quả hơn. Chẳng hạn, thiết bị 16550D của National Semiconductor có một bộ đệm thu 16 byte và một bộ đệm phát 16 byte, có thể hoạt động với tốc độ lên tới 1.5 Mbps. Các thiết bị UART hiện đại cũng có thể tự động xử lý các nhiệm vụ liên quan tới các hệ thống điểm-nhiều điểm trên một mạng.



HÌNH 37.7 Cấu trúc điển hình của UART

37.2 Các chuẩn giao diện nối tiếp TIA/EIA

Giao diện nối tiếp RS-232

RS-232 được phát triển vào năm 1962 bởi Hiệp hội công nghiệp điện tử EIA (Electronic Industries Association) ban đầu như là một giao diện giữa máy tính và thiết bị truyền thông. Ngày nay nó được duy trì bởi Hiệp hội công nghiệp Viễn Thông TIA (Telecommunication Industries Association) và EIA. Phiên bản hiện tại được gọi là TIA/EIA 232-F [4]. CCITT (The Consulting Committee for International Telegraphs and Telephones) đưa ra các khuyến khích sử dụng giao diện tương đương hoặc tương tự với các đặc tính đã được đưa ra bởi TIA/EIA.

Sự phát triển nhanh chóng của máy tính tạo ra nhu cầu truyền thông máy tính-máy tính với khoảng cách xa. Mạng điện thoại công cộng là cơ sở hạ tầng ổn định cho nhiệm vụ truyền thông. Bởi vì các máy tính tạo ra các dữ liệu số trong khi mạng điện thoại được thiết kế cho việc truyền tín hiệu tiếng nói, nên tín hiệu số từ máy tính phải được chuyển thành một tín hiệu điều chế để có thể truyền trên mạng tương tự. Môđem (bộ điều chế/ giải điều chế) được sử dụng để chuyển đổi tín

hiệu số thành tín hiệu điều chế tương tự để truyền trên đường dây điện thoại và được chuyển đổi ngược lại thành tín hiệu số bởi môđem tại đầu kia của đường dây điện thoại. RS-232 được thiết kế như là một giao diện giữa máy tính và môđem. Tên hình thức của chuẩn RS-232 là "Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange", trong đó thiết bị đầu cuối dữ liệu DTE (Data Terminal Equipment) là máy tính và thiết bị truyền thông dữ liệu DCE (Data Communication Equipment) là môđem. Hình 37.8 chỉ ra một ví dụ giao diện RS-232 trong hệ thống truyền thông máy tính – máy tính qua mạng điện thoại. Các máy tính ở mỗi đầu cuối là DTE và các môđem là DCE.

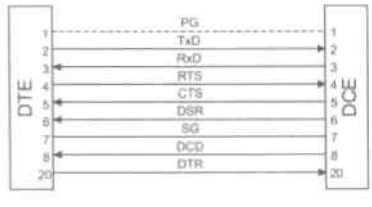


HÌNH 37.8 Truyền thông dữ liệu trên mạng điện thoại

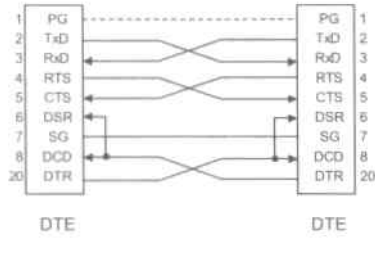
Chuẩn giao diện RS-232 chỉ rõ các đặc tính điện, cơ khí và chức năng của giao diện DTE/DCE. Giao diện CCITT V.24 mô tả các đặc tính chức năng tương đương và dựa trên các chuẩn khác về đặc tính cơ và điện của giao diện. Chuẩn RS-232 được sử dụng rộng rãi trong các ứng dụng cung cấp một kết nối điểm-điểm trực tiếp giữa hai máy tính hoặc giữa máy tính và các thành phần của hệ cơ điện tử, trong những trường hợp đó chúng ta đang thực hiện giao diện DTE tới giao diện DTE. Đây là một tình huống mà môđem không được yêu cầu, cáp được sử dụng cho việc kết nối một DTE đến một DTE khác gọi là cáp "null modem", cáp này đã có sẵn các kết nối bên trong để tạo lên một môđem ảo.

Đặc tính cơ được đề cập với các kết nối vật lý thực của DTE và DCE liên quan đến các đặc điểm kỹ thuật về các nhiệm vụ của chân và giống của giắc nối. Chuẩn RS-232 không chỉ rõ một loại giắc nối, nhưng nó là tuỳ chọn để sử dụng giắc nối loại D (DB-25) với 25-chân, được liệt kê trong tiêu chuẩn. Trong thực tế chỉ một số nhỏ các chân được sử dụng, do đó giắc nối loại D 9 chân thường được sử dụng thay thế. Chú ý rằng nhiệm vụ các chân của giắc nối DB-9 thì không được chỉ ra bởi RS-232 và khác với nhiệm vụ các chân của DB-25.

Hình 37.9 chỉ ra nhiệm vụ các chân của giắc DB-25 và kết nối bên trong của mạch được lựa chọn giữa một DTE và DCE. Hình 37.10 chỉ ra một ví dụ về nhiệm vụ các chân của giắc DB-25 và kết nối bên trong của các mạch được lựa chọn giữa hai máy tính có nghĩa là hai DTE.



HÌNH 37.9 Nhiệm vụ chân giữa một DTE và DCE



HÌNH 37.10 Nhiệm vụ chân của cáp "null modem"

Mô tả chức năng của các mạch giao tiếp được chọn

Chương này không mô tả tất cả các tín hiệu được chỉ ra bởi chuẩn RS-232. Người đọc có thể tham khảo các chuẩn liên quan ở [4]. Chúng tôi sẽ mô tả hầu hết các tín hiệu thông dụng được sử dụng trong các giao diện DTE/DCE và DTE/DTE. Chú ý rằng ngoại trừ mạch nối đất bảo vệ (Protective Ground) và mạch nối đất tín hiệu (Signal Ground), những mạch mang tín hiệu chỉ theo một hướng, như được chỉ ra bởi các mũi tên trong hình 37.9. Các đặc tính chức năng xác định các chức năng được thực hiện bởi từng mạch giao tiếp riêng biệt.

Protective Ground (*PG*): Đường dây này đảm bảo rằng vỏ máy của DTE và DCE có cùng điện thế.

Transmitted Data (*TxD*): Tín hiệu truyền được tạo ra từ DTE truyền tới DCE.

Received Data (*RxD*): Tín hiệu nhận tạo ra từ DCE truyền tới DTE.

Request to Send (*RST*): Tín hiệu này được sử dụng để quy định DCE cho việc truyền dữ liệu. Trên các kênh hai chiều không đồng thời tín hiệu điều khiển hướng truyền dữ liệu của DCE (truyền hay nhận). Trên kênh một chiều hoặc trên các kênh hai chiều đồng thời, tín hiệu này điều khiển trạng thái truyền của DCE (truyền hay không truyền). Tín hiệu này được tạo ra từ DTE truyền tới DCE.

Clear to Send (*CTS*): Chỉ ra rằng DCE đã sẵn sàng để nhận và đáp ứng với tín hiệu yêu cầu RTS. Tín hiệu CTS được tạo ra từ DCE truyền tới DTE.

Data Set Ready (*DSR*): Chỉ ra rằng DCE sẵn sàng để hoạt động. Tín hiệu này được tạo ra từ DCE và truyền tới DTE.

Signal Ground (*SG*): Là một đường đất chung cho tất cả các tín hiệu khác.

Data Carrier Detect (*DCD*): Tín hiệu này chỉ ra rằng DCE đang nhận một tín hiệu điều chế hợp lệ từ DCE ở đầu bên kia. Tín hiệu này được tạo ra từ DCE và truyền tới DTE.

Data Terminal Ready (*DTR*): Tín hiệu này chỉ ra rằng DTE đã được cấp nguồn và sẵn sàng hoạt động. Tín hiệu này được tạo ra từ DTE và truyền tới DCE.

RS-232 xác định giao diện điểm-điểm, đơn hướng, không cân bằng. Các kết nối bên trong được thực hiện qua một tập các dây dẫn được xem như các mạch giao tiếp. Các đặc tính về điện chỉ ra mức điệp áp của tín hiệu, tốc độ thay đổi của tín hiệu, trở kháng đường dây của các mạch trao đổi. Tiêu chuẩn này đưa ra việc mã hoá NRZ (Nonreturn to Zero) các tín hiệu số.

Chuẩn này yêu cầu các bộ phát phải được thiết kế sao cho các trở kháng đầu cuối giữa 3 và 7 KW, các bộ phát phải có khả năng truyền các điện áp mức cao giữa +5 và +15 V và điện áp thấp giữa –5 và –15 V. Các tín hiệu điện được thiết kế với một số dư 2V theo các mức tín hiệu. Tín hiệu bộ thu được định nghĩa từ +3 đến +15 V cho điện áp cao và –3 đến –15 V cho điện áp thấp.

Chú ý rằng đối với các mạch trao đổi dữ liệu, điện áp mức cao được định nghĩa là mức lôgic 0 (SPACE), trong khi điện áp mức thấp được định nghĩa là mức lôgic 1 (MARK). Mặt khác, với các tín hiệu điều khiển, điện áp mức cao xác định trạng thái ON trong khi điện áp mức thấp xác định trạng thái OFF. Tốc độ thay đổi của tín hiệu tối đa được cho phép cho cả đường truyền tín hiệu và dữ liệu là 30V/μs.

Chuẩn ban đầu cũng đã chỉ rõ chiều dài tối đa của cáp là 15m. Đặc tính kỹ thuật này được thay thế bởi chỉ tiêu kỹ thuật về tải điện dung tối đa cho phép là 2500 pF trong EIA/TIA 232-D. Chiều dài cáp tối đa được xác định bởi điện dung của cáp trên một đơn vị chiều dài; do đó tham số này xác định gián tiếp chiều dài của cáp giao diện. Giao diện RS-232 có tốc độ tín hiệu trong phạm vi 0 đến 20 Kbps. Chú ý rằng trong thực tế, một thiết kế tốt sẽ cho phép các khoảng cách lớn hơn và các tốc độ dữ liệu lớn hơn so với các giá trị đã được chỉ ra trong tiêu chuẩn.

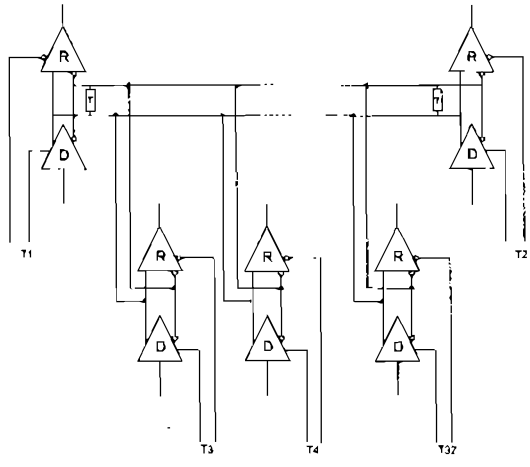
Giao diện RS-422 và RS-485

Chuẩn TIA/EIA-422-B về đặc tính điện của các mạch giao diện số điện áp cân bằng [5], định nghĩa các đặc tính điện của giao diện RS-422. RS-422 xác định một giao diện cân bằng đầu cuối, bộ phát đơn, đơn hướng. Tiêu chuẩn này cho phép nhiều bộ thu (tối đa là 10) trên một đường

truyền. Hình 37.3 minh hoạ một ứng dụng điểm-điểm điển hình của RS-422. Do việc cải thiện khả năng chống nhiễu, giao diện RS-422 hỗ trợ tốc độ dữ liệu lên đến 10 Mbps và chiều dài cáp lên đến 1200m, cho dù không đồng thời. Tốc độ tối đa 10 Mbps được hỗ trợ trên cáp có chiều dài tối đa 12m, trong khi cáp có chiều dài 1200m tốc độ tối đa là 100 Kbps. Nhận thấy rằng chiều dài cáp và tốc độ dữ liệu là một tham số bị giới hạn của giao diện. Môi trường truyền dẫn là một đường truyền đôi dây xoắn.

Chuẩn TIA/EIA-485-A về các đặc tính điện của các máy phát và máy thu để sử dụng trong các hệ thống đa điểm số [6], TIA/EIA-485-A định nghĩa các đặc tính điện của giao diện RS-485. RS-485 là chuẩn duy nhất cho phép nhiều nút liên kết theo hai hướng trên một đường truyền đôi dây xoắn đơn. Chuẩn RS-485 xác định một giao diện cân bằng đa điểm, giá thấp với các đặc tính điện, được hỗ trợ các loại đường dây, chiều dài cáp và tốc độ dữ liệu tương đương được chỉ ra trong chuẩn RS-422. Các phần của RS-485 tương thích và có khả năng thay thế các phần tương đương của RS-422, tuy nhiên các phần của RS-232 không nên sử dụng trong các hệ thống RS-485. RS-422 thường được sử dụng trong các hệ thống truyền thông hai chiều đồng thời điểm-điểm, trong khi RS-485 được sử dụng trong các hệ thống hai chiều không đồng thời đa điểm. Đặc điểm khác biệt của bộ phát RS-485 là khả năng 3 trạng thái, cho phép sử dụng nhiều bộ phát. RS-485 đã cải thiện khả năng bộ phát và dải điện áp đầu vào và hỗ trợ tới 32 thiết bị (các bộ phát và/hoặc các bộ thu) trên một đường truyền đơn. Hình 37.11 chỉ ra một ứng dụng đa điểm điển hình RS-485.

Chú ý rằng cả RS-422 và RS-485 chỉ là các chuẩn về điện. Chúng không chỉ ra các yêu cầu về cơ khí hay chức năng.



HÌNH 37.11 Ví dụ một ứng dụng đa điểm RS-485

Như đã nói ở trên, các tài liệu của các nhà cung cấp thiết bị sẽ cung cấp các nguồn thông tin rất hữu ích. Tổng quan về truyền thông dữ liệu thực tế và giao diện cho các thiết bị và điều khiển được cung cấp trong [7]. Thảo luận chi tiết về các khía cạnh thiết kế truyền thông nối tiếp và giao diện dựa trên RS-232 và RS-485 được đưa trong [8]. Người thiết kế nên tham khảo các tiêu chuẩn liên quan [4]-[6] cho toàn bộ các chỉ tiêu kỹ thuật. Ngoài ra, các thông tin khác bao gồm các lời khuyên trong thiết kế có thể tìm thấy trong [9]-[11]. Cơ sở lý thuyết và tổng quan về truyền thông dữ liệu và mạng máy tính có thể tìm thấy trong [12].

37.3 IEEE 488-Bus giao diện đa năng (GPIB)

Giới thiệu

Giao diện được mô tả bởi chuẩn IEEE 488 và được coi như là GPIB ở chương này, được sử dụng để kết nối các thiết bị với các hệ thống kiểm tra và đo. Ví dụ các thiết bị đo như đồng hồ đo điện áp số, máy hiện sóng, máy in, máy vẽ. Nói chung các thiết bị này được gọi là các thiết bị

GPIB. Các thiết bị này hoạt động dưới sự điều khiển của bộ điều khiển. Hầu hết các hệ thống hiện đại chứa một tập hợp các thiết bị như thế được kết nối đến một hoặc nhiều máy tính. Trong một hệ thống như thế, một trong các máy tính sẽ trở thành bộ điều khiển.

Giao diện được phát triển bởi Hewlett-Packard vào năm 1965. Tại thời điểm đó giao diện được gọi là HPIB, và không có một chuẩn chung. Năm 1975, IEEE 488 đã được mô tả một cách có hệ thống và được gọi là giao diện số chuẩn IEEE cho các thiết bị lập trình được. Chuẩn chỉ rõ các khía cạnh về phần điện, cơ khí và phần cứng có nghĩa là các tín hiệu, chức năng và mục đích của chúng. Các nhà sản xuất thiết bị sử dụng giao diện tự do, không tuân theo một giao thức chuẩn nào trong truyền thông với các thiết bị. Các thiết bị có cùng một mục đích chung, của các nhà sản xuất khác nhau yêu cầu các lệnh rất khác nhau. Một vài thiết bị đo đáp ứng với một câu lệnh, trong khi một vài thiết bị đo tương tự khác lại không phản ứng với câu lệnh đó một chút nào. Hơn nữa không có dạng dữ liệu phù hợp nào giữa thiết bị gửi dữ liệu và thiết bị nhận dữ liệu. Tình trạng này dẫn đến việc phải mở rộng tiêu chuẩn 488. Tiêu chuẩn mới đã được ban hành năm 1987 và được gọi là chuẩn IEEE 488.2 về các mã, các dạng, các giao thức, và các lệnh thông thường để sử dụng với IEEE 488.1 (1987) [13], trong đó IEEE 488.1 là tên mới cho chuẩn IEEE 488 ban đầu.

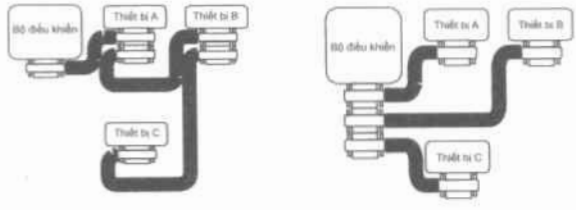
Các thiết bị tương thích với IEEE 488.2 phải biểu diễn dữ liệu qua các dạng dữ liệu và các mã hóa được chỉ ra trong tiêu chuẩn. Chuẩn cũng xác định một tập tối thiểu các thứ tự điều khiển bắt buộc hoặc các lệnh và các đề nghị về một vài các lệnh tuỳ chọn khác. Nó cũng cung cấp một mô hình báo cáo trạng thái chuẩn phải được thực hiện bởi các nhà sản xuất thiết bị vì vậy việc xác định trạng thái của thiết bị sẽ dễ dàng hơn cho các lập trình viên.

Mặc dù vẫn chưa phải là một chuẩn, nhưng các lệnh chuẩn cho các thiết bị có thể lập trình được (SCPI) được tập hợp lại trong các thỏa thuận của năm 1990 dựa trên một tập các lệnh chuẩn cho các loại thiết bị khác nhau. Theo đó tất cả các vôn kế số được sản xuất bởi các nhà sản xuất khác nhau sẽ đáp ứng với cùng lệnh GPIB.

Phần cứng GPIB

Phần này sẽ nói về các tiêu chuẩn kỹ thuật cơ điện của giao diện GPIB cũng như mô tả về tín hiệu và mục đích của chúng.

Tất cả các thiết bị GPIB được kết nối bằng cách sử dụng cáp đặc biệt có một đầu là giắc nối đực và đầu kia là giắc nối cái. Điều này cho phép kết nối các cáp chồng lên nhau. Các thiết bị có thể được kết nối theo dạng mắt xích (ví dụ thiết bị B kết nối với thiết bị A, thiết bị C kết nối với thiết bị B...) hoặc là theo hình sao (ví dụ thiết bị A, B, C... được kết nối tới một điểm chung). Các cấu hình kết nối được chỉ ra trong hình 37.12.



HÌNH 37.12 Cấu hình mắt xích và cấu hình sao kết nối các thiết bị GPIB tới bộ điều khiển

Tối đa 15 thiết bị có thể nối vào một bus. Khoảng cách tối đa giữa hai thiết bị là 4m và khoảng cách trung bình là hơn 2 m. Ít nhất 2/3 thiết bị kết nối phải được cấp nguồn.

Cáp GPIB gồm 24 dây. Tám trong số các dây này là các đường dữ liệu, 3 đường dây khác được sử dụng cho việc bắt tay. 5 đường dây nữa được sử dụng cho việc quản lý giao diện và 8 đường dây còn lại là các đường đất. Trong số các đường đất là một đường cáp bảo vệ, một đường đất tín hiệu, 3 đường đất ngược lại cho các tín hiệu bắt tay, và 3 đường đất ngược lại khác cho 3 đường dây quản lý giao diện. Tất cả các tín hiệu sử dụng mức tín hiệu TTL chuẩn với mức lôgic âm. Các đường bắt tay và các đường quản lý giao diện được đưa trong bảng 37.1.

BẢNG 37.1 Các đường bắt tay và các đường quản lý ghép nối

Các đường bắt tay		Các đường quản lý ghép nối	
NRFD	Không sẵn sàng với dữ liệu	ATN	Chuẩn bị
NDAC	Không chấp nhận dữ liệu	IFC	Xóa ghép nối
DAV	Dữ liệu hợp lệ	REN	Cho phép điều khiển từ xa
		SRQ	Đề nghị phục vụ
		EOI	Kết thúc nhận dạng

Hoạt động của từng đường riêng lẻ thì không quan trọng đối với người sử dụng bình thường hay những lập trình viên bởi vì việc sử dụng của họ liên quan tới các bộ điều khiển và các công cụ tuân theo chuẩn IEEE 488.2.

Bộ điều khiển, thiết bị nói và thiết bị nghe

Bộ điều khiển quản lý bus nói chung. Trong khi có thể có nhiều bộ điều khiển kết nối tới mạng GPIB, nhưng chỉ có một bộ điều khiển chịu trách nhiệm (CIC - controller-in-charge) quản lý bus tại một thời điểm xác định. Tất cả các thông tin gửi ra bởi bộ điều khiển trên đường dữ liệu được gọi "các lệnh" và tất cả các thông tin được gửi ra bởi các thiết bị khác được coi là "dữ liệu". Các thiết bị GPIB gửi dữ liệu bất cứ thời điểm nào được gọi là "các thiết bị nói" (talkers) và các thiết bị nhận dữ liệu được gọi là "các thiết bị nghe" (listeners). Trong khi có thể có nhiều thiết bị nghe đang hoạt động tại một thời điểm cho trước bất kỳ, thì chỉ có duy nhất một thiết bị nói hoạt động tại một thời điểm xác định. Một hệ thống có thể có các thiết bị nói cố định và các thiết bị nghe cố định, tuy nhiên có khả năng là một thiết bị GPIB có thể là thiết bị nghe ở một thời điểm và ở thời điểm khác là thiết bị nói. Tổng quan về các đường dây tín hiệu được đưa ra dưới đây sẽ cho chúng ta hiểu hơn về hoạt động của giao diện GPIB.

Các đường quản lý giao diện

Attention (ATN)

Đường ATN được điều khiển bởi CIC. Khi được xác nhận, các tín hiệu trên đường dữ liệu tạo thành một tín hiệu lệnh và tất cả các thiết bị phải nghe. Khi không được xác nhận, các tín hiệu trên đường dữ liệu vẫn là dữ liệu và nói chung đã được gửi bởi một thiết bị nói tới một hoặc nhiều thiết bị nghe.

Interface Clear (IFC)

Đường IFC được xác nhận bởi CIC để reset bus GPIB. Tuỳ thuộc vào việc nhận dữ liệu này, tất cả các thiết bị GPIB trên bus sẽ tự khởi tạo lại.

Remote Enable (REN)

Các thiết bị có thể được điều khiển hoặc là ở gần hoặc từ xa. CIC xác nhận đường REN để đặt tất cả các thiết bị dưới chế độ lập trình từ xa. Do đó, chẳng hạn sự thay đổi của một DVM có thể được đưa ra bởi một câu lệnh GPIB thay vì dùng bảng điều khiển.

Service Resquest (SRQ)

Bất kỳ thiết bị nào trừ bộ điều khiển đều có thể đòi hỏi một cách không đồng bộ dịch vụ yêu cầu đường dây SRQ từ bộ điều khiển. Bộ điều khiển giám sát đường SRQ và thu thập tất cả các thiết bị để xác định xem những thiết bị nào đang yêu cầu dịch vụ.

End of Identity (EOI)

Tín hiệu EOI được sử dụng bởi thiết bị nói để chỉ ra điểm cuối của bản tin dữ liệu. Nó chỉ cho các thiết bị nghe điểm cuối của bản ghi dữ liệu đang nhận.

Các đường bắt tay

Nói chung, truyền dữ liệu với chế độ bắt tay phải trải qua ba trạng thái: trạng thái yêu cầu hay trạng thái sẵn sàng, trạng thái truyền dữ liệu, và trạng thái thông báo đã nhận được. Ở một vài hệ thống mà sự ổn định của dữ liệu trên đường bus dữ liệu là đáng nghi ngờ, thì phải đưa ra một tín hiệu báo dữ liệu hợp lệ. Trên bus GPIB, khi một thiết bị nói phải gửi dữ liệu tới cho một thiết bị nghe, bộ điều khiển phải đánh địa chỉ một thiết bị và chỉ định nó là thiết bị nói, rồi sau đó đánh địa chỉ cho một hoặc nhiều thiết bị khác và chỉ định chúng là các thiết bị nghe. Xem mục "định địa chỉ các thiết bị GPIB".

Ready for Data (NRFD)

Đường NRFD được điều khiển bởi bộ điều khiển khi gửi các lệnh hoặc bởi thiết bị nói khi gửi dữ liệu. Một thiết bị đã được chỉ định là thiết bị nghe sẽ không xác nhận NRFD để chỉ cho thiết bị nói rằng nó đã sẵn sàng nhận dữ liệu. Thiết bị chậm nhất trong số các thiết bị nghe là thiết bị cuối cùng không xác nhận NRFD và do đó điều khiển tốc độ truyền dữ liệu.

Data Valid (DVA)

Khi tất cả các thiết bị nghe cho biết đã sẵn sàng nhận dữ liệu bằng cách không xác nhận NFRD, thì thiết bị nói (hoặc bộ điều khiển khi gửi lệnh) sẽ xác nhận tín hiệu DAV để thông báo cho tất cả các thiết bị nghe rằng dữ liệu trên đường truyền DIO1-DIO8 là ổn định và có thể được đọc bởi thiết bị nghe. Để đáp ứng với tín hiệu DAV, các thiết bị nghe có thể xác nhận NRFD để tạm dừng bất cứ sự truyền dữ liệu thêm nào của thiết bị nói cho đến khi dữ liệu đã truyền được nhận.

Not Data Accepted (NDAC)

Đường NDAC được điều khiển bởi tất cả các thiết bị nghe là tín hiệu xác nhận (Acknowledgment Signal). Khi tất cả các thiết bị nghe đã nhận dữ liệu, đường NDAC sẽ không được xác nhận. Sau đó thiết bị nói có thể xóa dữ liệu và không xác nhận tín hiệu DAV.

Các đường dữ liệu DIO1-DIO8 (8 đường)

Các đường dữ liệu được điều khiển bởi bộ điều khiển khi phát lệnh hoặc bởi thiết bị nói. Ngay sau khi bộ điều khiển ra lệnh cho một thiết bị nào đó trở thành thiết bị nói, thì thiết bị đó sẽ đặt dữ liệu lên các đường truyền DIO1-DIO8 và sẽ chờ ít nhất T1 giây.

Định địa chỉ cho các thiết bị GPIB

Tất cả các thiết bị GPIB kết nối tới bus GPIB phải có một địa chỉ GPIB duy nhất. Một thiết bị có thể có một địa chỉ chính cũng như một địa chỉ phụ. Hầu hết các thiết bị chỉ sử dụng địa chỉ chính. Các địa chỉ nằm trong dải từ 0 đến 30 (số thập phân). Nói chung, các địa chỉ của các thiết bị cũng như bộ điều khiển được thiết lập bằng các chuyển mạch. Bộ điều khiển chỉ định một thiết bị với một địa chỉ nào đó là thiết bị nói hoặc thiết bị nghe bằng cách gửi một mẫu bit trên bus dữ liệu. Mẫu bit này được định dạng theo bảng 37.2. Các bit dữ liệu được đánh số từ D0 đến D7. Giá trị của mỗi bit được liệt kê chính xác ở dưới. Ký tự A có nghĩa là 0 hoặc 1. Năm bít D4-D0 sẽ định dạng một mẫu bit thể hiện địa chỉ của thiết bị. Ký tự X có nghĩa là một bít không cần quan tâm, không được sử dụng. TA sẽ được thiết lập bằng 1 nếu bộ điều khiển đang ra lệnh cho một thiết bị trở thành thiết bị nói. LA sẽ được thiết lập bằng 1 nếu bộ điều khiển ra lệnh cho một thiết bị thành thiết bị nghe. Chẳng hạn, nếu một thiết bị đặc biệt nào đã có địa chỉ 15 (thập phân) và nếu bộ điều khiển đang ra lệnh cho thiết bị đó là thiết bị nói, thì bộ điều khiển phải gửi mẫu bít dưới đây qua đường dữ liệu DIO8-DIO1.

Thiết bị với địa chỉ 15 là thiết bị nói 0 1 0 0 1 1 1 1 = 4 F (hex)

Tương tự:

Thiết bị với địa chỉ 0 là nghe 0 0 1 0 0 0 0 0 = 20 (hex)

Thiết bị nói hiện tại "Untalk" 0 1 1 1 1 1 1 1 = 5F (hex)

Tất cả thiết bị nghe "Unlisten" 0 0 1 1 1 1 1 1 = 3F (hex)

Chú ý rằng các câu lệnh Untalk và Unlisten trông rất giống câu lệnh Talk và Listen, tuy nhiên địa chỉ 31 thập phân không tồn tại. Do đó địa chỉ 31 được sử dụng để tác động đến Untalk và Unlisten.

Đối với bộ điều khiển, chuẩn IEEE 488.2 cung cấp các chuỗi điều khiển tuỳ chọn và yêu cầu. Tất cả các bộ điều khiển tuân theo IEEE 488.2 phải hỗ trợ tất cả các lệnh bắt buộc. Chuẩn này cũng cung cấp giao thức cho bộ điều khiển. Các giao thức được hình thành bằng cách tổ hợp một tập các trình tự điều khiển. Ví dụ, câu lệnh FINDLSTN sẽ phát ra một tập các trình tự điều khiển để xác định các thiết bị nghe đang tồn tại.

Đối với các thiết bị, IEEE 488.2 đưa ra một tập các lệnh bắt buộc và các truy vấn. Ví dụ, khi câu lệnh "RST" được nhận bởi các thiết bị tuân theo IEEE 488.2, tất cả chúng phải đưa ra một reset thiết bị. Tương tự, khi nhận câu lệnh "STB?" thiết bị sẽ gửi byte trạng thái tới bộ điều khiển.

BẢNG 37.2 Các lệnh định địa chỉ thiết bị nói/thiết bị nghe

D7	D6	D5	D4	D3	D2	D1	D0
X	TA	LA	A	A	A	A	A

Tất cả các lệnh bắt buộc, các truy vấn, các trình tự điều khiển và tuỳ chọn cũng như các giao thức bộ điều khiển có thể tìm thấy trong [14], [15].

Tài liệu tham khảo

[1] Goldie, J., Summary of well known interface standards, Application note AN-216, National Semiconductor, 1998, www.national.com.

[2] Goldie, J., Comparing EIA-485 and EIA-422-A line drivers and receivers in multipoint applications, Application note AN-759, National Semiconductor, 1998.

[3] McNamara, J.E., *Technical Aspects of Data Communication,* 3rd edition, Digital Press, 1988.

[4] TIA/EIA-232-F, Interface between data terminal equipment and data communication equipment employing serial binary data interchange, TIA, EIA, 1997.

[5] TIA/EIA-422-B, Electrical characteristics of balanced voltage digital interface circuits, TIA, EIA, 1995.

[6] TIA/EIA-485-A, Standard for electrical characteristics of generators and receivers for use in digital multipoint systems, TEI, EIA, 1998.

[7] Mackay, S.G., et al., *Data Communications for Instrumentation and Control,* IDC Techbooks, 2000.

[8] Axelson, J., *Serial Port Complete,* Lakeview Research, Madison, 1998.

[9] Goldie, J., Ten ways to bulletproof RS-485 interfaces, Application note AN-1057, National Semiconductor, 1996.

[10] RS-422 and RS-485 Application Note, B&B Electronics Manufacturing Co., 1997, www.bb-elec.com.

[11] DALLAS SEMICONDUCTOR, Application Note 83, Fundamentals of RS-232 Serial Communications, 1998.

[12] Stallings, W., *Data and Computer Communications,* 6th ed., Prentice-Hall, Upper Saddle River, NJ, 2000.

[13] ANSI/IEEE 488.1-1987 IEEE standard digital interface for programmable instrumentation institution of electrical and electronic engineers, New York, 1987.

[14] ANSI/IEEE 488.2-1987 IEEE standard codes, formats, protocols and common commands, institution of electrical and electronic engineers, New York, 1987.

[15] ANSI/IEEE 488.2-1992 IEEE standard codes, formats, protocols and common commands, and standard commands for programmable instruments. Institution of Electrical and Electronic Engineers, New York, 1992.

38

Truyền thông và mạng máy tính

38.1 Lịch sử tóm tắt 38-1
38.2 Giới thiệu 38-2
38.3 Mạng máy tính 38-4
38.4 Kỹ thuật cấp phát tài nguyên 38-10
38.5 Các thách thức và các vấn đề 38-11
38.6 Tóm tắt và kết luận 38-11

Mohammad Ilyas
Florida Atlantic University

Lĩnh vực truyền thông và mạng máy tính giải quyết việc truyền thông tin linh hoạt và tin cậy từ điểm này đến điểm khác. Sự cần thiết trao đổi thông tin không còn là điều mới mẻ nhưng công nghệ dùng để thiết kế trao đổi thông tin đã có những bước phát triển vượt bậc đặc biệt trong vài thập kỷ qua. Nhiều yếu tố đã và đang tiếp tục thúc đẩy sự phát triển này. Internet là sản phẩm điển hình nhất, nó đã đi vào cuộc sống của mỗi chúng ta. Chương này mô tả các đặc tính nổi bật và chi tiết của hệ thống truyền thông và mạng máy tính.

Nội dung của chương này được chia thành các phần sau. Phần 38.1 mô tả tóm tắt về lịch sử của lĩnh vực truyền thông. Phần 38.2 sẽ giới thiệu về hệ thống truyền thông và mạng máy tính. Phần 38.3 mô tả chi tiết về mạng máy tính. Phần 38.4 trình bày về cơ chế cấp phát tài nguyên. Phần 38.5 mô tả tóm tắt những khó khăn và các vấn đề trong truyền thông và mạng máy tính. Cuối cùng, phần 38.6 tóm tắt nội dung của chương.

38.1 Lịch sử tóm tắt

Trao đổi thông tin (truyền thông) giữa hai hay nhiều thực thể là một nhu cầu cần thiết trong đời sống của con người. Nó được bắt đầu từ một vài dạng và hình thái của giọng nói, được tạo ra từ một thực thể và thực thể khác có thể nghe và hiểu. Qua một vài thế kỷ, các giọng nói đó đã trở thành ngôn ngữ. Theo sự phát triển dân số của thế giới, ngày càng nhiều ngôn ngữ được sinh ra. Trải qua một thời gian dài, các ngôn ngữ đã được sử dụng trong giao tiếp giữa mọi người với nhau. Nếu cần truyền thông tin (bản tin) đi xa, người ta sẽ làm gọn nó và gửi để phân phát bản tin đến một vị trí khác. Dần dần, ngày càng có nhiều phương pháp được phát triển để biểu diễn và trao đổi thông tin. Các phương pháp đó gồm các ký hiệu, các hình vẽ và thậm chí các bảng chữ cái. Sự phát triển này đã đem đến sự linh hoạt trong việc lưu giữ thông tin và sử dụng cách "không phát âm" để trao đổi thông tin. Vì vậy, việc lưu giữ, phân bố, chia sẻ và trao đổi hiểu biết trở nên dễ dàng.

Từ khoảng 150 năm trước đây, việc truyền thông qua các phương tiện không dây và gồm cả các tín hiệu khói, đánh trống và sử dụng các mặt phản chiếu do các tín hiệu ánh sáng mang lại (không dây quang học). Hiệu suất của các công nghệ này đã bị ảnh hưởng rất lớn bởi các điều kiện môi trường. Ví dụ, sử dụng tín hiệu khói sẽ không hiệu quả trong điều kiện gió. Trong một số trường hợp mà sau này chúng ta không để ý, một vài công nghệ đã được sử dụng trong các thế kỷ trước để truyền thông tin đi xa giống với các công nghệ mà chúng ta đang sử dụng. Chỉ khác ở chỗ việc thực hiện các công nghệ ngày nay có độ phức tạp hơn so với trước kia.

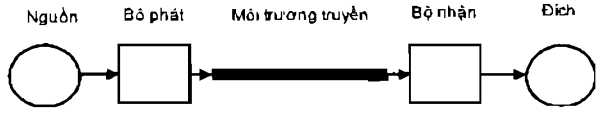
Với sự phát triển liên tục của công nghệ và sự xuất hiện của thiết bị điện tử, lĩnh vực truyền thông cũng bắt đầu sử dụng các công nghệ mới. Các chữ cái đã truyền được vào trong các biểu diễn điện tử vì vậy thông tin được truyền điện tử. Sự mã hoá đã được phát triển để trao đổi bức điện. Phát triển xa hơn nữa là cho phép sử dụng điện thoại. Một lưu ý rất quan trọng là thời kỳ đầu của công nghệ này, người sử dụng có thể đi đến các nơi để gửi các bức điện đi xa hoặc nói chuyện điện thoại với những người ở xa. Đây là một ví dụ cổ điển về chia sẻ tài nguyên. Tất nhiên phải thiết lập các kết nối với các vị trí ở xa.

Việc cải tiến các công nghệ truyền thông đã và đang thu được kết quả tích cực. Máy tính điện tử cũng phát triển mạnh và các bản tin đã ra đời. Trước đó máy tính không chỉ đắt và kém tin cậy mà còn có kích thước khổng lồ. Ví dụ, Các máy tính sử dụng đèn chân không thì cỡ khoảng bằng một phòng rộng và sử dụng khoảng 10000 ống chân không. Các máy tính này sẽ ngừng làm việc nếu một trong các đèn này bị cháy và cần được thay thế, khi đó phải dùng đến cả thang. Trung bình, những máy tính đó phải mất vài phút để thay thế đèn khác. Một vài phút không đủ để máy tính thực hiện chương trình tính toán lớn. Với sự phát triển của transistor, máy tính không chỉ trở nên nhỏ hơn về kích thước và rẻ hơn về giá thành mà còn tin cậy hơn. Ảnh hưởng của máy tính là đem đến các ứng dụng phổ biến. Với sự phát triển của máy tính cá nhân, gần như không có khía cạnh nào trong đời sống của con người mà không được sử dụng máy tính. Truyền thông không phải là ngoại lệ và người sử dụng máy tính đã đưa khả năng truyền thông lên một tầm cao mới.

38.2 Giới thiệu

Việc truyền thông tin từ điểm này tới điểm khác một cách hiệu quả và tin cậy luôn là điều cần thiết. Một hệ thống truyền thông điển hình bao gồm các thành phần như hình 38.1 sau:

- Nguồn phát hoặc nơi có thông tin để truyền.
- Bộ truyền để chuẩn bị thông tin truyền đi.
- Môi trường truyền dẫn để mang thông tin từ nơi này sang nơi khác
- Bộ nhận thông tin và chuẩn bị nó để giao cho nơi nhận
- Đích để lấy thông tin từ bộ nhận và sử dụng nó khi cần thiết



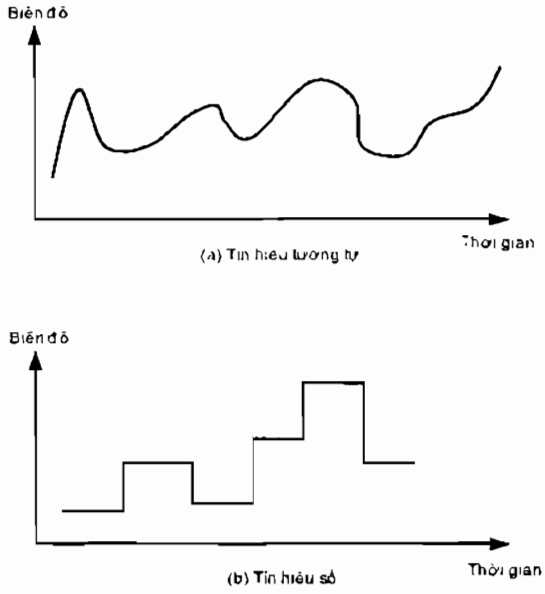
HÌNH 38.1 Một hệ thống truyền thông điển hình

Thông tin được phát dưới dạng tương tự hoặc số. Thông tin tương tự được biểu diễn bằng một tín hiệu liên tục, tín hiệu biến đổi trơn tru theo thời gian. Khi giọng nói vào micro, một tín hiệu âm tương tự được phát ra. Thông tin dạng số biểu diễn bởi một tín hiệu mà nó giữ một giá trị nhất định trong một khoảng thời gian sau đó thay đổi sang một giá trị cố định khác. Máy tính làm việc với thông tin dạng số ở 2 mức (tín hiệu số nhị phân). Hình 38.2 là một ví dụ về tín hiệu số và tín hiệu tương tự. Việc truyền thông tin cũng có thể là dạng tương tự hoặc dạng số. Do đó hệ thống truyền thông có 4 khả năng sau [21]:

- Thông tin tương tự được truyền như là một tín hiệu tương tự
- Thông tin tương tự được truyền như là một tín hiệu số
- Thông tin số được truyền như là một tín hiệu tương tự
- Thông tin số được truyền như là một tín hiệu số

Không có nhiều sự lựa chọn về dạng thông tin (tương tự hoặc số) phát ra từ một thiết bị. Ví dụ, một tín hiệu giọng nói như người nói, một tín hiệu video như tín hiệu được phát ra bởi một máy camera, một tín hiệu tốc độ được phát ra bởi một phương tiện và một tín hiệu độ cao được phát ra

bởi thiết bị trên máy bay sẽ luôn luôn là tín hiệu tương tự trong tự nhiên. Tuy nhiên có một lựa chọn về dạng thông tin (tương tự hoặc số) để truyền qua môi trường truyền dẫn. Thông tin được truyền có thể là tương tự hoặc số trong tự nhiên và thông tin có thể dễ dàng chuyển đổi từ dạng này sang dạng khác.



HÌNH 38.2 Tín hiệu tương tự và tín hiệu số điển hình

Mỗi khả năng có những ưu điểm và nhược điểm riêng. Khi tín hiệu mang thông tin được truyền, nó sẽ mất năng lượng, cường độ và nhận thêm các tín hiệu nhiễu ngay khi nó phát ra khỏi bộ truyền. Nếu năng lượng của tín hiệu không được gia tăng tại các điểm trung gian, nó sẽ bị tắt ở nơi nào đó trước khi tới đích theo dự kiến. Tất nhiên là tín hiệu sẽ bị mất thông tin. Để tăng năng lượng và cường độ của tín hiệu, nó phải được khuyếch đại (với tín hiệu tương tự) và xây dựng lại (với tín hiệu số). Khi một tín hiệu tương tự được khuyếch đại, thì nhiễu cũng được khuyếch đại và đây là điều không muốn ở nơi nhận, tín hiệu nhận được không giống (hoặc không gần) với dạng tín hiệu ban đầu. Ngược lại, tín hiệu số được xử lý và khôi phục lại tại các điểm trung gian, Do đó nhiễu về cơ bản được lọc đi. Hơn nữa, truyền thông tin dạng số có nhiều ưu điểm khác, gồm xử lý thông tin để phát hiện lỗi và sửa lỗi, áp dụng các công nghệ mã hoá và giải mã để thông tin bảo mật và nhiều ưu điểm nữa. Vì vậy công nghệ truyền thông số đã trở thành công nghệ vượt trội trong các lĩnh vực truyền thông [9, 18].

Như đã chỉ ra ở trên, công nghệ truyền thông đã có bước phát triển vượt bậc trong những thập kỷ qua. Hai nhân tố luôn luôn đóng vai trò then chốt trong việc định hướng tương lai của hệ thống truyền thông [20] đó là:

- Yêu cầu khắt khe của người sử dụng cần để trao đổi thông tin
- Tình trạng của công nghệ liên quan đến truyền thông

Trong lịch sử, các phát minh luôn được thúc đẩy bởi tính khắc nghiệt của nhu cầu. Điều này đã rất đúng với các lĩnh vực truyền thông. Thêm nữa luôn có một sự thúc đẩy và tò mò để cho mọi thứ xảy ra nhanh hơn. Khi điện được phát minh và con người (mới chỉ là thưa thớt) muốn trao đổi thông tin đi xa và mất ít thời gian, điện báo đã được ra đời. Mã hóa đã được phát triển với các chuỗi ngắn hơn (với các chấm và các gạch) để biểu diễn ký tự nhanh hơn. Điều đó đã dẫn đến việc truyền các bản tin nhanh hơn. Sự có mặt của điện và khả năng truyền thông tin đi xa bằng dây dẫn đã thúc đẩy sự phát triển của các thiết bị chuyển đổi giọng nói con người sang các tín hiệu điện và

vì vậy dẫn đến sự phát triển của hệ thống điện thoại. Sau phát minh này là mong muốn thiết lập hệ thống truyền thông hai chiều với giọng nói của con người. Khi mà sử dụng điện thoại đã trở nên phổ biến, người sử dụng điện thoại phải được kết nối tới mọi người khác và điều đó đã dẫn đến sự phát triển của các trạm điện thoại. Ngay trong thời gian này, các trạm điện thoại vẫn còn được vận hành bởi con người. Khi công nghệ đã phát triển, các trạm điện thoại này được thay thế bởi các trạm điện thoại tự động. Mỗi người sử dụng điện thoại được gán một số điện thoại với mục đích nhận dạng và người khác có thể quay số để thiết lập kết nối với người được gọi. Với sự phát triển của công nghệ máy tính và việc máy tính trở nên nhỏ gọn hơn, đã tạo ra vô số các ứng dụng trong đó có lĩnh vực truyền thông. Máy tính không chỉ thay thế trạm điện thoại tự động, chúng còn thực hiện rất nhiều nhiệm vụ khác của hệ thống truyền thông. Ví dụ chuyển đổi thông tin từ tương tự sang số và ngược lại, xử lý thông tin để phát hiện lỗi và sửa lỗi, nén và mã hoá/giải mã thông tin.

Khi máy tính trở nên hữu dụng, có nhiều ứng dụng được thực hiện dựa trên nó. Ứng dụng rõ nhất là những người sử dụng máy tính chia sẻ thông tin với nhau. Khối lượng thông tin được trao đổi với những người sử dụng xung quanh đã tăng lên theo hàm mũ qua 3 thập kỷ gần đây. Đến khi người sử dụng cần trao đổi một lượng thông tin khổng lồ, các công nghệ mới đã được phát minh để xử lý linh hoạt. Không chỉ những người sử dụng trao đổi thông tin với những người khác theo một cách nhất định, mà còn cần các máy tính trao đổi thông tin với nhau. Các thông tin được trao đổi theo cách này có nhiều điểm khác biệt hơn với các thông tin trao đổi qua điện thoại. Điều này dẫn đến việc phải kết nối các máy tính với nhau và dẫn tới cái được gọi là mạng máy tính.

38.3 Mạng máy tính

Mạng máy tính là một sự kết nối của nhiều máy tính. Các kết nối tạo nên sự linh hoạt để cung cấp các phương tiện tin cậy và hiệu quả trong việc truyền thông với các máy tính và các thiết bị khác. Truyền thông sử dụng trong mạng máy tính được hỗ trợ bởi các máy tính và các tính năng truyền thông cũng được cung cấp cho các máy tính đó. Mạng máy tính cũng dựa trên mạng truyền thông máy tính. Việc kết nối các máy tính có thể qua môi trường truyền dây hoặc môi trường không dây [5, 6, 10, 13, 18].

Có hai cách tổ chức chính của mạng máy tính:

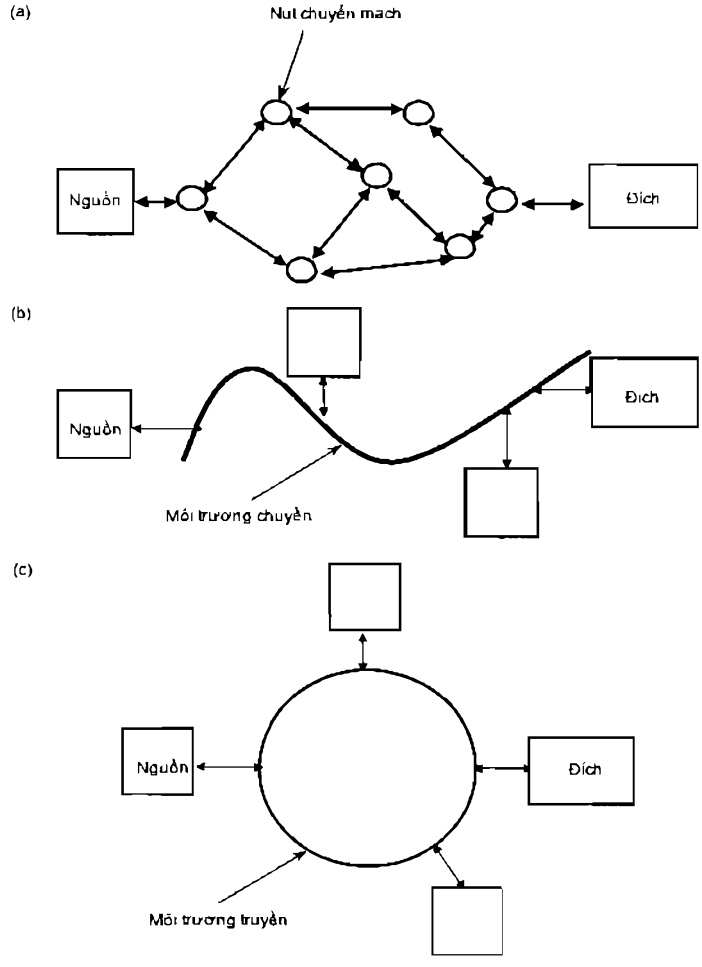
- Mạng diện rộng
- Mạng cục bộ/mạng đô thị

Mạng máy tính diện rộng, như tên gọi của nó, có phạm vi địa lý rộng hơn và về cơ bản có thể có phạm vi toàn cầu. Ngược lại, mạng nội bộ và mạng đô thị bị giới hạn về khoảnh cách. Mạng cục bộ thì bị giới hạn trong một toà nhà, một nhà máy hoặc một cơ quan. Mạng đô thị cũng bị giới hạn về phạm vi địa lý nhưng nó lớn hơn mạng cục bộ [19]. Mạng cục bộ và mạng đô thị điển hình được minh họa như hình 38.3.

Một người sử dụng được kết nối tới mạng máy tính, người đó có thể giao tiếp với bất kỳ người sử dụng nào được nối tới mạng đó tại một số điểm. Không đòi hỏi người sử dụng phải kết nối trực tiếp với người sử dụng kia. Trên thực tế, với mạng diện rộng, 2 người sử dụng truyền thông sẽ rất ít khi được kết nối trực tiếp với nhau. Nghĩa là người sử dụng sẽ phải chia sẻ đường truyền để trao đổi dữ liệu. Đây là điều quan trọng nhất của mạng máy tính. Chia sẻ tài nguyên để cải thiện việc sử dụng tài nguyên và tất nhiên là có nhiều lợi nhuận. Hơn nữa chia sẻ đường truyền, người sử dụng cũng sẽ chia sẻ nguồn xử lý của các máy tính tại các nút chuyển mạch, khả năng đệm để lưu trữ thông tin tại các trạm chuyển đổi và các nguồn tài nguyên khác được kết nối tới mạng máy tính. Một người sử dụng được nối tới mạng máy tính tại bất kỳ nút chuyển mạch nào đều có thể truy cập tới tất cả tài nguyên ngay lập tức (các cơ sở dữ liệu, các bài báo nghiên cứu, các bài điều tra và các thứ khác nữa) mà các nguồn tài nguyên đó cũng được kết nối tới mạng máy tính. Tất nhiên, để truy cập các thông tin xác định có thể bị hạn chế và người sử dụng phải có bản quyền thích hợp mới truy cập được thông tin.

Thông tin từ một người sử dụng tới người khác cần phải đi qua các nút chuyển mạch và các đường truyền trước khi tới nơi nhận. Điều này dẫn đến việc một người sử dụng có thể có nhiều lựa

chọn để chọn một trong nhiều thứ tự các đường truyền và các nút chuyển mạch để trao đổi thông tin. Điều đó làm tăng thêm sự tin cậy của quá trình trao đổi thông tin. Nếu một đường dẫn không sẵn sàng, không khả thi hoặc không thực hiện được, các đường dẫn khác có thể được sử dụng. Thêm nữa, để việc chia sẻ thông tin tốt hơn và hiệu quả hơn cho mỗi thành viên sử dụng, không cho phép bất kỳ người sử dụng nào trao đổi một lượng thông tin lớn tại một thời điểm. Tuy nhiên, không lấy làm lạ khi mọi người có lượng lớn thông tin đều được trao đổi. Trong trường hợp đó, thông tin được chia thành các đơn vị thông tin nhỏ gọi là gói thông tin. Mỗi gói được gửi đi tới nơi nhận như là các thực thể riêng biệt và sau đó tất cả các gói được ghép lại với nhau ở nơi nhận để tái tạo lại thông tin ban đầu [2].



HÌNH 38.3 (a) Một mạng truyền thông máy tính diện rộng điển hình. (b) một mạng đường trục truyền thông cục bộ/đô thị điển hình. (c) Một mạng vòng truyền thông cục bộ/đô thị điển hình

Vì môi trường tài nguyên bị chia sẻ, nên người sử dụng có thể sẽ không trao đổi được thông tin của họ tại một thời điểm họ muốn bởi vì các nguồn tài nguyên (các nút chuyển mạch, các đường truyền) có thể bận để phục vụ người sử dụng khác. Trong trường hợp đó, vài người sử dụng có thể phải chờ một thời gian trước khi họ bắt đầu truyền thông. Những người thiết kế mạng máy tính đã thiết kế sao cho tổng số thời gian trễ (bao gồm cả thời gian đợi) nhỏ nhất có thể và tổng số thông tin trao đổi thành công là lớn nhất có thể.

Rất nhiều bước phải được thực hiện để cho mạng có thể truyền thông tin của người sử dụng từ nơi này sang nơi khác. Các bước chính là:

- Cơ chế địa chỉ để xác định người sử dụng
- Cơ chế địa chỉ cho các gói thông tin để xác định nguồi và đích của chúng
- Thiết lập kết nối giữa bên gửi và bên nhận và xác nhận nó
- Chọn đường dẫn hoặc lộ trình (thứ tự các nút chuyển mạch và đường truyền) để mang thông tin từ nơi gửi đến nơi nhận
- Thực hiện một lộ trình hoặc đường dẫn đã lựa chọn
- Kiểm tra lỗi và sửa lỗi các gói thông tin
- Mã hóa và giải mã thông tin
- Điều khiển luồng thông tin để các tài nguyên chia sẻ không bị vượt quá
- Thông báo với nơi gửi là thông tin đã gửi thành công tới nơi nhận (xác nhận bản tin)
- Thông báo với người sử dụng nguồn tài nguyên
- Cho phép các máy tính khác đang chạy các ứng dụng khác và hệ điều hành có thể trao đổi thông tin
- Chuẩn bị thông tin thích hợp để truyền qua môi trường truyền dẫn đã cho

Đây không phải là danh sách chi tiết các bước cần thực hiện trong mạng máy tính. Trong nhiều trường hợp, tất cả các vấn đề được đề cập bằng các thủ tục chi tiết và rất có hệ thống. Các thủ tục đó gọi là các giao thức truyền thông. Các giao thức được thực hiện ở các nút chuyển mạch bằng sự kết hợp của cả phần cứng và phần mềm. Thực hiện tất cả các phần đó trên một môđun phần cứng hoặc phần mềm thì đều không phù hợp bởi vì nó sẽ trở nên rất khó quản lý. Một cách thực hiện chuẩn là chia các đặc tính này thành nhiều môđun nhỏ và sau đó các mô đun này được giao diện cùng nhau để cùng thực hiện các đặc tính đó. Tổ chức chuẩn hóa quốc tế (ISO –International Standards Organization) đã chia các đặc tính này thành 7 môđun riêng biệt được gọi là các lớp. Mô hình được đề xuất được gọi là mô hình tham chiếu kết nối hệ thống mở (OSI-Open System Interconnection). Mô hình này được đề xuất có 7 lớp [2]:

- Lớp ứng dụng
- Lớp biểu diễn dữ liệu
- Lớp kiểm soát nối
- Lớp vận chuyển
- Lớp mạng
- Lớp liên kết dữ liệu
- Lớp vật lý

Lớp vật lý thực hiện việc truyền thông tin trong môi trường truyền dẫn. Lớp liên kết dữ liệu điều khiển thông tin trong mỗi đường truyền. Lớp mạng thực hiện chức năng tìm đường hoặc đường dẫn cho dữ liệu từ nút chuyển mạch nơi mà cả nguồn và đích cùng được nối tới. Nó cũng theo dõi luồng dữ liệu từ đầu đến cuối. Còn lại 4 lớp thuộc về thiết bị của người sử dụng. Lớp vận chuyển liên quan đến trao đổi thông tin từ nguồn đến nơi nhận. Lớp kiểm soát nối điều khiển thiết lập liên kết truyền thông giữa bên gửi, bên nhận và duy trì nó. Lớp biểu diễn dữ liệu thực hiện biểu diễn các thông tin được trình bày ở các lớp dưới. Mã hóa và giải mã thông tin cũng có thể được thực hiện ở lớp này. Lớp ứng dụng thực hiện cho các chương trình ứng dụng, chương trình ứng dụng phát thông tin tại nguồn phát và làm việc với dữ liệu khi nhận được thông tin từ nơi nhận.

Thông tin bắt đầu từ lớp ứng dụng tại nơi gửi đi, nó được xử lý ở các lớp tùy thuộc vào các giao thức xác định được thực hiện trên lớp đó. Mỗi lớp xử lý thông tin và xác nhận đầu và cuối bản tin trước khi chuyển qua lớp tiếp theo. Đầu và cuối bản tin được xác nhận bởi các lớp tạo nên tiêu đề bản tin, điều này là cần thiết cho việc truyền thông tin. Cuối cùng, tại lớp vật lý, các bit của các gói thông tin được chuyển đổi sang tín hiệu phù hợp và truyền qua môi trường truyền dẫn. Tại phía nhận, lớp vật lý sẽ nhận các gói thông tin từ môi trường vật lý và chuẩn bị chúng để gửi lên các lớp cao hơn. Một gói thông tin được xử lý bởi các lớp giao thức tại nơi nhận, phần đầu và cuối

của mỗi gói được dỡ bỏ trước khi sang lớp tiếp theo. Đến lúc thông tin nhận được ở lớp ứng dụng, nó có dạng giống như lúc nó truyền đi ở nơi phát.

Một người sử dụng sẵn sàng gửi thông tin cho người sử dụng khác với hai lựa chọn: có thể thiết lập một sự giao tiếp với đích trước để trao đổi thông tin hoặc có thể gửi thông tin tới nút mạng để nút mạng phân phát thông tin tới đích. Nếu sự giao tiếp được thiết lập trước để trao đổi thông tin, quá trình gọi là dịch vụ hướng kết nối và được thực hiện bởi các kết nối mạch ảo. Ngược lại, nếu không kết nối nào được thiếp lập trước để gửi thông tin, quá trình được gọi là dịch vụ không kết nối. Quá trình này được thực hiện bằng việc sử dụng một môi trường dữ liệu. Trong dịch vụ hướng kết nối, tất cả các gói dữ liệu giữa 2 người sử dụng được đưa qua cùng một đường dẫn của mạng máy tính và vì vậy, chúng đến đích cùng thứ tự như lúc đã gửi đi bởi nguồn. Trong dịch vụ không kết nối, mỗi gói sẽ tìm một đường đi riêng qua mạng để tới đích. Vì vậy các gói dữ liệu sẽ đến đích không theo thứ tự. Trong trường hợp đó, ở nơi nhận sẽ phải đặt tất cả các gói theo thứ tự trước khi chúng được ghép lại thành bản tin [2, 10, 13]

Với tất cả các hệ thống chia sẻ tài nguyên, sự phân phối nguồn tài nguyên trong mạng máy tính phải hết sức lưu ý. Ý tưởng chủ yếu là các nguồn tài nguyên nên được chia đều cho mọi người sử dụng trong mạng máy tính. Tại một thời điểm, mạng cần được duy trì thực hiện càng tối ưu càng tốt. Hai tham số thực hiện để đánh giá mạng máy tính đó là thời gian trễ và thông lượng. Thời gian trễ là khoảng thời gian từ lúc thông tin được đưa ra bởi một người sử dụng đến lúc truyền thành công đến nơi nhận. thông lượng là tổng số thông tin được truyền thành công tới nơi nhận trên một đơn vị thời gian. Do môi trường chia sẻ tài nguyên trong mạng máy tính, hai tham số thực hiện này lại trái ngược với nhau. Người ta luôn mong muốn tham số thời gian trễ nhỏ nhất và tham số thông lượng lớn đến mức có thể. Để tăng thông lượng, mạng máy tính phải tăng lượng thông tin vận chuyển, nhưng tăng vận chuyển thông tin dẫn đến phải cần bộ đệm lớn hơn ở các nút chuyển mạch và vì vậy sẽ làm tăng thời gian đợi cho các gói thông tin khác. Điều này sẽ làm tăng thời gian trễ, ngược lại nếu giản thời gian trễ cũng làm giảm thông lượng. Một sự thỏa hiệp hợp lý giữa thông lượng và thời gian trễ là cần thiết để thỏa mãn hoạt động của mạng máy tính [10, 11].

Mạng máy tính diện rộng

Một mạng diện rộng bao gồm các nút chuyển mạch và các đường truyền như hình 38.3(a). Sự sắp xếp của các nút chuyển mạch và các đường truyền được dựa trên các mẫu vận chuyển sao cho khối lượng vận chuyển của luồng dữ liệu từ nơi này đến nơi khác là lớn nhất. Các nút chuyển mạch cung cấp cho người sử dụng để truy cập tới mạng máy tính và thực hiện các giao thức truyền thông. Khi người sử dụng muốn truyền dữ liệu, nút chuyển mạch nơi được người sử dụng kết nối tới sẽ thiết lập một kết nối nếu dịch vụ hướng kết nối đã được chọn. Ngược lại, thông tin sẽ được truyền trong môi trường không kết nối. Trong trường hợp này, các trạm chuyển đổi thực hiện vai trò tìm đường dẫn cho luồng dữ liệu phụ thuộc vào các tiêu chuẩn phân tuyến đang sử dụng. Tiêu chuẩn này gồm các các tham số mục tiêu (thời gian trễ và thông lượng) và các nhân tố xung quanh dựa trên sự cần thiết của người sử dụng. Để duy trì mạng hoạt động với một dải phù hợp, phải có các cơ chế điều khiển luồng dữ liệu. Vào cuối những năm 60 và đầu những năm 70, khi mà dải truyền dữ liệu sử dụng trong mạng máy tính thấp (vài nghìn bit trên giây), các cơ chế này là khá đơn giản. Một phương pháp được sử dụng phổ biến để điều khiển dữ liệu qua đường truyền là: bên gửi sẽ liên tục gửi dữ liệu cho đến khi bên nhận gửi một thông báo yêu cầu dừng. Luồng dữ liệu sẽ lặp lại ngay khi bên nhận gửi một yêu cầu tiếp tục truyền. Về cơ bản bên nhận được gọi là điều khiển luồng dữ liệu qua đường truyền. Khi dải dữ liệu của môi trường truyền bắt đầu tăng, phương pháp này được cho là không hiệu quả. Để điều khiển luồng dữ liệu truyền nhanh hơn, cơ chế cửa sổ trượt đã được sử dụng. Theo cơ chế này, bên gửi sẽ liên tục gửi các gói thông tin nhưng với một giới hạn xác định. Khi đạt đến giới hạn, bên gửi sẽ dừng gửi các gói thông tin và đợi để xác nhận các gói đó đã được truyền hay chưa. Ngay khi có một xác nhận, bên gửi sẽ gửi gói thông tin khác. Phương pháp này đảm bảo rằng số gói thông tin gửi đi không nhiều hơn giới hạn trong một khoảng thời gian cho trước. Ngược lại bên nhận có toàn bộ thông tin mà bên gửi gửi đi. Các kỹ thuật để điều khiển truyền dữ liệu này là kỹ thuật phản ứng hoặc kỹ thuật dựa trên phản hồi bởi vì việc quyết định truyền hay không được dựa trên điều kiện vận chuyển hiện tại.

Kỹ thuật phản ứng có thể chấp nhận được trong các dải truyền thấp đến trung bình. Khi dải truyền tăng từ kilobit trên giây lên megabit và gigabit trên giây, tình hình sẽ thay đổi. Qua vài năm, dải truyền dữ liệu đã tăng lên đáng kể. Các sợi quang học cung cấp tốc độ truyền dữ liệu vô cùng lớn. Kích thước của mạng máy tính cũng đã được tăng lên đáng kể. Lượng thông tin vận chuyển qua các mạng này cũng được tăng lên theo hàm mũ. Giả thiết rằng, các công nghệ điều khiển truyền thông được sử dụng trong mạng trước đây thì không hiệu quả nữa [11, 12, 22]. Nhiều yếu tố làm tăng độ phức tạp đó là hiện nay người sử dụng đang trao đổi các kiểu thông tin khác nhau qua cùng một mạng. Xét ví dụ về Internet. Phạm vi địa lý của Internet về cơ bản là toàn cầu. Sử dụng sợi quang học là môi trường truyền cung cấp dải truyền cao để trao đổi thông tin. Thêm nữa, mọi người đang sử dụng để trao đổi các kiểu thông tin, bao gồm âm thanh, hình ảnh và dữ liệu. Tất cả các yếu tố này về cơ bản là cần thiết để thay đổi các tiếp cận trong quản lý lưu thông trong mạng máy tính. Nguyên nhân chính dẫn đến sự thay đổi này là các gói thông tin đó sẽ vận chuyển rất nhanh qua mạng máy tính mà không điều khiển dựa trên phản hồi sẽ rất chậm. Vì vậy, một số cơ chế ngăn chặn đã được phát triển để duy trì truyền thông bên trong một mạng máy tính để đạt được mức độ tin cậy cao hơn. Bên gửi thực hiện các kỹ thuật để đảm bảo rằng chỉ có lưu thông về thông tin mới được cho phép vào mạng được điều khiển bởi mạng máy tính [1, 20, 22]. Dựa trên yêu cầu của người sử dụng và trạng thái công nghệ, việc cung cấp truyền đã nhanh hơn với các kiểu dữ liệu khác nhau của các dịch vụ (tiếng nói, hình ảnh, dữ liệu và các kiểu khác), trong cùng một mạng máy tính thì các tích hợp và đồng nhất là điều rất cần thiết. Các mạng máy tính này được coi như là mạng số tích hợp dịch vụ băng thông rộng (broadbband integrated services digital networks - BISDNs). BISDNs cung cấp khả năng kết nối số từ đầu đến cuối và người sử dụng có thể truy cập các kiểu dịch vụ truyền thông từ một điểm truy cập đơn giản. BISDNs sử dụng chế độ truyền không đồng bộ (Asynchronous transfer mode - ATM). ATM về bản chất là một kỹ thuật chuyển gói nhanh khi thông tin được truyền có kích thước cố định và nhỏ, được gọi là các "tế bào". Mỗi tế bào thông tin là 53 byte trong đó có 5 byte đầu. Thông tin chủ yếu được truyền sử dụng môi trường hướng kết nối (mạch ảo) [3, 4, 8, 12, 17].

Một đặc điểm nữa của mạng diện rộng là tốc độ xử lý của các trạm chuyển mạch. Khi dải truyền tăng, trạm chuyển mạch cần có khả năng xử lý nhanh hơn. Nếu không, các trạm chuyển mạch sẽ trở nên tắc nghẽn và môi trường truyền nhanh có thể là không hoàn toàn hữu ích. Khi môi trường truyền dẫn là các sợi quang học, thông tin đến các trạm chuyển mạch được chuyển đổi từ dạng quang sang dạng điện tử vì vậy nó có thể được xử lý và chuyển tới liên kết tương ứng. Trước khi nó được truyền, thông tin lại được chuyển từ dạng điện tử sang dạng quang. Điều này làm giảm việc xử lý thông tin truyền và tăng thời gian trễ. Để làm giảm tình trạng này, việc nghiên cứu đang được tiến hành để phát triển các bộ chuyển mạch quang học sử dụng như các trạm chuyển mạch. Các bộ chuyển mạch quang học sẽ không cần phải chuyển đổi thông tin từ quang sang điện tử và ngược lại tại các trạm chuyển mạch; tuy nhiên, các bộ chuyển mạch này cũng phải có khả năng xử lý quang cho thông tin. Khi các bộ chuyển mạch quang học có kích cỡ vừa phải trở nên có giá trị, sử dụng sợi quang học làm môi trường truyền dẫn cùng với các bộ chuyển mạch quang học sẽ dẫn đến máy tính và mạng truyền thông toàn bộ là quang học. Các gói thông tin sẽ không cần được lưu trữ để xử lý tại các trạm chuyển đổi và điều đó tất nhiêu là giảm được thời gian trễ. Thêm nữa, các kỹ thuật hợp kênh chia độ dài sóng đang mang lại việc sử dụng trong môi trường truyền quang học với khả năng trọn vẹn nhất [14].

Mạng cục bộ và mạng đô thị

Một mạng cục bộ có một phạm vi địa lý giới hạn (không lớn hơn vài ki lô mét) và thường được giới hạn trong một tòa nhà và một cơ quan. Nó sử dụng một môi trường truyền đơn giản và tất cả mọi người sử dụng được kết nối tới cùng một môi trường tại vài điểm truy cập. Môi trường truyền có thể là bus như hình 38.3(b) hoặc có thể là dạng vòng (ring) như hình 38.3(c). Mạng đô thị cũng có môi trường truyền đơn giản, nó được chia sẻ bởi tất cả mọi người sử dụng được kết nối tới mạng, nhưng khoảng cách truyền lớn hơn, có thể tới 150 km. Nó cũng sử dụng một môi trường truyền với tốc độ truyền cao hơn. Mạng cục bộ và mạng đô thị cũng sử dụng một sự thực thi lớp của các giao thức truyền thông như mạng diện rộng; tuy nhiên, các giao thức này đơn giản hơn bởi

vì cấu trúc mạng đơn giản, trạm chuyển mạch và khoảng cách bị giới hạn giữa bên gửi và bên nhận. Tất cả người sử dụng chia sẻ cùng môi trường truyền để trao đổi thông tin. Tất nhiên, nếu 2 hoặc nhiều người sử dụng truyền thông tin tại cùng một thời điểm, các thông tin từ các người sử dụng khác nhau sẽ cản trở nhau và dẫn đến xung đột. Trong các trường hợp đó, thông tin của tất cả các bên đều bị hủy và sẽ phải truyền lại. Vì vậy, phải có các thủ tục rõ ràng để mọi người có thể dùng chung một đường truyền theo một cách nhất định và có sự trao đổi thành công của các thông tin. Các thủ tục này được gọi là các giao thức điều khiển truy cập đường truyền (MAC - Medium Access Control).

Có 2 phương pháp chính của giao thức điều khiển truy cập đường truyền:

- Các giao thức truy cập có điều khiển
- Các giao thức truy cập dựa trên sự tranh chấp

Với các giao thức truy cập có điều khiển, người sử dụng nhận lệnh truyền thông tin của mình và chỉ một người sử dụng được phép truyền thông tin tại một thời điểm. Khi một người kết thúc việc truyền, người tiếp theo bắt đầu truyền. Việc điều khiển có thể là tập trung hoặc phân tán. Không thể xung đột thông tin và vì vậy không bị mất thông tin do 2 hay nhiều người truyền cùng một thời điểm. Ví dụ các giao thức truy cập có điều khiển gồm mạng cục bộ dạng truyền thẻ bài và mạng cục bộ dạng vòng truyền thẻ bài. Trong đó, một thẻ bài luân chuyển qua các trạm. Một trạm có thẻ bài được phép truyền thông tin và các trạm khác đợi tới khi chúng nhận được thẻ bài [19].

Với giao thức dựa trên xung đột, người sử dụng không truyền thông tin xoay vòng. Một người sẽ tự quyết định gửi thông tin và cũng chấp nhận mạo hiểm nếu xảy ra xung đột với các trạm khác cũng gửi dữ liệu tại thời điểm đó. Nếu không có xung đột xuất hiện, thông tin được truyền thành công tới trạm nhận. Ngược lại, nếu có một xung đột xảy ra, thông tin từ các trạm sẽ cần phải truyền lại. Một ví dụ của giao thức MAC dựa trên tranh chấp là CSMA/CD (Carier Sense Multiple Access with Collision Detection) giao thức này được sử dụng trong mạng Ethernet. Trong CSMA/CD. Một người sử dụng nghe đường dẫn trước khi truyền. Nếu đường truyền bận trạm sẽ đợi một thời gian. Tuy nhiên, nếu môi trường rỗi, trạm đó sẽ truyền thông tin. Nhờ việc cảm nhận, giao thức MAC này có thể tránh được các xung đột, nhưng các xung đột vẫn xảy ra. Lý do là sự truyền thông dọc theo đường truyền có tốc độ hạn chế. Nếu một trạm nghe được đường truyền là rỗi tại một điểm, điều đó không có nghĩa là một trạm khác ở vị trí khác chưa bắt đầu truyền. Điều này phản ánh ảnh hưởng thời gian trễ của tín hiệu điện từ trong môi trường truyền. Đây là tham số quan trọng nhất quyết định hiệu suất của mạng cục bộ dựa trên sự tranh chấp [11,19].

Mạng cục bộ cũng đã được thiết kết gói gọn trong khả năng truyền với tốc độ cao hơn. Khi dài dữ liệu của môi trường truyền tăng, hiệu ứng của thời gian trễ lan truyền sẽ được thể hiện rõ rệt. Với mạng cục bộ tốc độ cao hơn như Gigabit Ethernet và 100-BASE-FX, các giao thức truy cập đường truyền được thiết kế để giảm hiệu ứng thời gian trễ lan truyền. Nếu không chú ý thích đáng đến ảnh hưởng của trễ lan truyền thì hiệu suất của mạng cục bộ tốc độ cao sẽ rất kém [15,19].

Mạng đô thị về cơ bản giải quyết giống như mạng cục bộ. Các mạng này được sử dụng làm xương sống cho các mạng cục bộ kết nối khác nhau. Có các mạng tốc độ cao và phạm vi địa lý lớn hơn. Các giao thức điều khiển truy cập đường truyền để chia sẻ môi trường truyền dẫn dựa trên điều khiển truy cập. Hai ví dụ điển hình của mạng đô thị là FDDI (Fiber Distributed Data Interface) và DQDB (Distributed Queue Dual Bus). Với FDDI, môi trường truyền là dạng 2 vòng, ngược lại DQDB sử dụng 2 đường. Các vòng FDDI mang thông tin trên một sợi và một sợi để dự phòng, sự sắp xếp này làm tăng độ tin cậy của truyền thông. Với DQDB, 2 bus cũng mang thông tin trên 1 bus. Giao thức MAC cho FDDI dựa trên truyền thẻ bài và hỗ trợ giọng nói và truyền thông dữ liệu xung quanh các trạm. DQDB sử dụng cơ chế truy cập riêng và cũng hỗ trợ giọng nói và truyền thông dữ liệu xung quanh các trạm [19].

Mạng không dây và mạng di động

Truyền thông không sử dụng dây dẫn vật lý đã được khám phá và các mạng truyền thông di động và không dây đảm bảo điều đó. Trong mấy năm gần đây đã có sự phát triển chưa từng có

trong mạng truyền thông không dây. Sự tiến bộ đầy ý nghĩa đã làm nên các công nghệ, chúng hỗ trợ môi trường truyền thông không dây và sẽ rất phát triển trong tương lai. Các thiết bị sử dụng cho truyền thông không dây đòi hỏi các đặc tính cụ thể mà các thiết bị truyền thông nối dây không cần phải có. Các đặc tính này bao gồm tiêu tốn năng lượng ít, trọng lượng nhỏ và có khả năng truyền thông xa.

Với mạng truyền thông không dây và mạng di động, việc truy cập mạng truyền thông là không dây vì vậy người sử dụng có thể tự do dịch chuyển. Phần cuối của đường truyền có thể là dây, không dây hoặc kết hợp cả hai. Nói chung một người sử dụng di động, trong khi truyền thông, có một kết nối không dây với một khả năng truyền thông cố định và phần cuối của đường truyền là dây. Dải truyền thông không dây thì luôn luôn bị giới hạn và vì vậy dải của người sử dụng di động cũng bị giới hạn. Để vượt qua giới hạn này, môi trường truyền thông tế bào đã được phát minh. Trong môi trường truyền thông tế bào, một vùng địa lý được chia ra thành các miền nhỏ được gọi là các tế bào. Mỗi tế bào có một thiết bị truyền thông nhất định nó sẽ đáp ứng tất cả các thiết bị di động trong vùng đó. Tuy nhiên, với một di động, trong khi nó hoạt động mà nó di chuyển ra ngoài một vùng nào đó và vào vùng khác, thì dịch vụ của kết nối đó sẽ chuyển từ vùng này sang vùng khác. Việc này gọi là quá trình đẩy [7,16].

Sự sắp xếp vùng có rất nhiều ưu điểm. Khi kích thước vùng nhỏ, các thiết bị di động không cần công suất truyền cao. Điều này dẫn đến các thiết bị tiêu tốn ít năng lượng hơn. Thêm nữa, như ta đã được biết là phổ tần có thể sử dụng cho truyền thông không dây đã bị giới hạn và có thể vì vậy hỗ trợ số lượng nhỏ các kết nối truyền thông không dây tại một thời điểm. Việc chia các vùng truyền thông thành các tế bào cho phép người sử dụng cùng tần số ở các tế bào khác nhau miễn là chúng đủ xa để tránh xung đột. Sự tăng thêm số thiết bị di động này sẽ được hỗ trợ. Sự cải tiến trong các thuật toán xử lý tín hiệu số và điện tử đã làm cho các thiết bị truyền thông di động mạnh hơn, nhỏ hơn, đẹp hơn và đa năng. Các thiết bị này có khả năng truyền thông di động rất lớn bao gồm truy cập internet không dây, e-mail không dây và truyền hình ảnh video không dây qua các thiết bị cầm tay. Điện thoại di động luôn sẵn sàng và hoạt động trong môi trường truyền thông khác nhau xuyên qua các đại lục. Một ngày không xa, một con số truyền thông đơn giản duy nhất sẽ được gán vào mỗi người từ khi mới sinh và khi đó không cần quan tâm người đó ở đâu.

Một lĩnh vực khác đang phát triển rất nhanh là lĩnh vực mạng truyền thông không dây không cơ sở hạ tầng. Các mạng này có tính chất tạm thời và được thiết lập cho việc cần thiết nhất định và trong một khoảng thời gian nhất định. Để thiết lập các mạng này không cần phải cài đặt gì phức tạp. Khi một vài thiết bị viễn thông di động tiến lại gần nhau, chúng có thể thiết lập một mạng truyền thông xung quanh chúng. Nơi mà mạng không dây không cơ sở hạ tầng có thể được sử dụng là phòng học, gặp gỡ, hội thảo, các hoàn cảnh khôi phục thảm họa, v.v. Khi sự cần thiết của mạng được thỏa mãn, việc thiết lập mạng không cơ sở hạ tầng sẽ biến mất.

38.4 Kỹ thuật cấp phát tài nguyên

Như đã trình bày ở trên, mạng máy tính là các hệ thống chia sẻ tài nguyên. Người sử dụng chia sẻ hầu hết các tài nguyên như: môi trường truyền dẫn, công suất xử lý và vùng nhớ đệm tại các trạm chuyển mạch và các nguồn tài nguyên khác, chúng là một phần của mạng. Bí quyết hoạt động thành công của mạng máy tính là cấp phát tài nguyên công bằng và hiệu quả cho người sử dụng. Từ trước đến nay, đã có 2 phương pháp để cấp phát nguồn tài nguyên cho người sử dụng trong mạng máy tính:

- Cấp phát tài nguyên tĩnh
- Cấp phát tài nguyên động

Cấp phát tài nguyên tĩnh nghĩa là số lượng tài nguyên đòi hỏi được cấp phát cho mỗi người sử dụng để sử dụng nó bất kỳ khi nào họ cần. Nếu người sử dụng không sử dụng các nguồn tài nguyên được cấp phát này thì người khác không được sử dụng. Ngược lại, cấp phát tài nguyên động nghĩa là số lượng tài nguyên mong muốn được cấp pháp cho người sử dụng trên cơ sở người đó yêu cầu và trong quá trình người đó cần. Khi đã thoả mãn sự cần thiết, sự cấp phát được lấy lại.

Trong trường hợp này, người khác có thể sử dụng các nguồn tài nguyên này nếu họ cần. Việc cấp phát tĩnh đem lại sự lãng phí của tài nguyên, nhưng không đòi hỏi kết hợp toàn bộ như cấp phát động. Công nghệ này được sử dụng trong tình huống cho phép tùy thuộc vào cung cấp và yêu cầu. Nếu nguồn tài nguyên phong phú và yêu cầu không quá cao, có thể nên sử dụng cấp phát tài nguyên tĩnh. Tuy nhiên, khi nguồn tài nguyên khan hiếm và yêu cầu cao, cấp phát động lại là sự cần thiết để tránh lãng phí tài nguyên.

Trước kia, truyền thông và mạng máy tính đã giải quyết cả 2 trường hợp. Các môi trường truyền thông trước kia đã sử dụng cấp phát tài nguyên động khi người sử dụng đến văn phòng công cộng để gọi điện thoại hoặc gửi điện báo. Sau đó vài năm, cấp phát tài nguyên tĩnh được chọn, khi người sử dụng đã cấp phát tài nguyên của riêng họ để chỉ ra các kênh truyền thông và các kênh này không chia sẻ cho những người sử dụng khác. Vào cuối những thập kỷ 60 thời đại của mạng máy tính đã bắt đầu hé nở với cấp phát tài nguyên động và toàn bộ mạng truyền thông và mạng máy tính đã tiếp tục với cách truyền thống này. Với sự phát triển của sợi cáp quang, các nguồn tài nguyên truyền dẫn phong phú và có thể thỏa mãn tại mọi thời điểm. Nhiều nhà nghiên cứu và các hãng sản xuất đã có thiện ý quay lại với cấp phát tài nguyên tĩnh, nhưng cấp phát tài nguyên động vẫn tiếp tục được dùng và sẽ tồn tại trong nhiều năm tới [10].

38.5 Các thách thức và các vấn đề

Nhiều thách thức và vấn đề liên quan đến truyền thông và mạng máy tính vẫn còn phải vượt qua. Ở đây chỉ mô tả những vấn đề quan trọng nhất.

Tốc độ truyền dữ liệu cao được cung cấp bởi sợi cáp quang và việc xử lý tốc độ cao có thể thực hiện ở các trạm chuyển mạch đã dẫn đến thời gian để thông tin truyền từ điểm này đến điểm khác giảm đi. Tuy nhiên, thời gian trễ lan truyền về cơ bản thì vẫn chưa được thay đổi. Thời gian trễ này chỉ phụ thuộc vào khoảng cách và không phụ thuộc vào tốc độ truyền hoặc kiểu môi trường truyền dẫn. Vấn đề này liên quan đến một sự tiềm ẩn chống lại vấn đề thời gian [11]. Trong trường hợp này, các công nghệ quản lý vận chuyển dựa trên phản hồi truyền thống trở nên không hiệu quả. Công nghệ ngăn chặn mới để quản lý vận chuyển và điều khiển linh hoạt là cần thiết để thu được tất cả các khả năng của truyền thông và mạng máy tính [22].

Việc tích hợp các dịch vụ khác nhau trên cùng một mạng cũng đã đặt ra nhiều thách thức mới. Mỗi dịch vụ đều có những đòi hỏi riêng để tạo ra mức chất lượng của dịch vụ mong muốn (QoS - Quality of Service). Trong mỗi mạng mà cố gắng thỏa mãn QoS cho từng dịch vụ thì sẽ gây ra những khó khăn cho các mạng khác. Vì vậy phải cố gắng thiết kế chất lượng dịch vụ sao cho có thể cung cấp đồng đều tới bên trong hệ thống truyền thông và mạng máy tính. Đó là một thách thức cần được lưu ý cẩn thận và giải quyết triệt để [13].

Duy trì bảo mật và nguyên vẹn thông tin đang là một thách thức khác. Nguy cơ thông tin hỏng trực tiếp hoặc gián tiếp đều rất lớn. Hơn nữa, những vi phạm bản quyền để đạt mục đích truy cập vào cơ sở dữ liệu cũng rất dễ xảy ra. Những vấn đề này cần được giải quyết để người sử dụng có thể sử dụng trong các công nghệ truyền thông và mạng máy tính một cách hiệu quả nhất.

38.6 Tóm tắt và kết luận

Chương này đã mô tả cơ bản về truyền thông và mạng máy tính và sự phát triển mới nhất liên quan đến các lĩnh vực này. Truyền thông và mạng máy tính đã chứng kiến sự phát triển rất lớn và cải tiến vượt bậc trong những thập kỷ gần đây.

Mạng máy tính là hệ thống chia sẻ tài nguyên cơ bản nơi mà người sử dụng chia sẻ đường truyền và các trạm chuyển mạch. Nó được sử dụng để trao đổi thông tin giữa những người sử dụng mà không cần kết nối trực tiếp. Có nhiều sự tăng trưởng về tốc độ truyền của môi trường truyền và khả năng xử lý của các trạm chuyển mạch cũng đã được nhân lên. Các mạng máy tính đang hỗ trợ việc truyền thông các kiểu dịch vụ khác nhau trên cùng một mạng. Các kiểu thông tin, bất kể kiểu và nguồn nào đang được vận chuyển dưới dạng các gói (ví dụ gói ATM). Các tài nguyên đang

được cấp phát cho người sử dụng theo cách cấp phát động để sử dụng tốt hơn. Mạng truyền thông không dây mở ra để cung cấp khả năng kết nối không dây và trao đổi thông tin ở mọi thời điểm.

Sự phát triển này cũng đặt ra một số vấn đề. Các công nghệ quản lý giao thông thuận tiện, các đòi hỏi về chất lượng dịch vụ và bảo mật thông tin là những thách thức chủ yếu cần được khắc phục để chinh phục sự tin tưởng của người sử dụng.

Tài liệu tham khảo

[1] Bae, J., and Suda, T., "Survey of traffic control schemes and protocols in ATM networks," *Proceedings of the IEEE,* Vol. 79, No. 2, February 1991, pp. 170–189.

[2] Beyda, W., "Data communications from basics to broadband," Third Edition, 2000.

[3] Black, U., "ATM: foundation for broadband networks," Prentice-Hall, Englewood Cliffs, NJ, 1995.

[4] Black, U., "Emerging communications technologies," Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1997.

[5] Chou, C., "Computer networks in communication survey research," *IEEE Transactions on Professional Communication,* Vol. 40, No. 3, September 1997, pp. 197–208.

[6] Comer, D., "Computer networks and internets," Prentice-Hall, Englewood Cliffs, NJ, 1999.

[7] Goodman, D., "Wireless personal communication systems," Addison-Wesley, Reading, MA, 1999.

[8] Goralski, W., "Introduction to ATM networking," McGraw-Hill, New York, 1995.

[9] Freeman, R., "Fundamentals of telecommunications," John Wiley & Sons, New York, 1999.

[10] Ilyas, M., and Mouftah, H.T., "Performance evaluation of computer communication networks," *IEEE Communications Magazine,* Vol. 23, No. 4, April 1985, pp. 18–29.

[11] Kleinrock, L., "The latency/bandwidth tradeoff in gigabit networks," *IEEE Communications Magazine* Vol. 30, No. 4, April 1992, pp. 36–40.

[12] Kleinrock, L., "ISDN-The path to broadband networks," *Proceedings of the IEEE,* Vol. 79, No. 2, February 1991, pp. 112–117.

[13] Leon-Garcia, A., and Widjaja, I., "Communication networks, fundamental concepts and key architectures," McGraw Hill, New York, 2000.

[14] Mukherjee, B., "Optical communication networks," McGraw-Hill, New York, 1997.

[15] Partridge, C., "Gigabit networking," Addison-Wesley, Reading, MA, 1994.

[16] Rappaport, T., "Wireless communications," Prentice-Hall, Englewood Cliffs, NJ, 1996.

[17] Schwartz, M., "Broadband integrated networks," Prentice-Hall, Englewood Cliffs, NJ, 1996.

[18] Shay, W., "Understanding communications and networks," Second Edition, PWS, 1999.

[19] Stallings, W., "Local and metropolitan area networks," Sixth Edition, Prentice-Hall, Englewood Cliffs, NJ, 2000.

[20] Stallings, W., "ISDN and broadband ISDN with frame relay and ATM," Fourth Edition, Prentice-Hall, Englewood Cliffs, NJ, 1999.

[21] Stallings, W., "High-speed networks, TCP/IP and ATM design principles," Prentice-Hall, Englewood Cliffs, NJ, 1998.

[22] Yuan, X., "A study of ATM multiplexing and threshold-based connection admission control in connection-oriented packet networks," Doctoral Dissertation, Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, Florida 33431, August 2000.

39

Phân tích lỗi trong hệ cơ điện tử

Leila Notash
Queen's University

Thomas N. Moore
Queen's University

39.1 Giới thiệu 39-1
39.2 Công cụ phân tích hỏng hóc/độ tin cậy 39-2
39.3 Phân tích hỏng hóc trong hệ Cơ điện tử 39-2
39.4 Các phương pháp phát hiện lỗi thông minh 39-3
39.5 Những vấn đề về phát hiện lỗi thông minh 39-5
39.6 Ví dụ về hệ Cơ điện tử: Tay máy/Máy công cụ song song 39-5
39.7 Kết luận và nhận xét 39-10

39.1 Giới thiệu

Khi ngành tự động hóa phát triển đến mức độ cao, đặc biệt là tự động hóa thông minh, thì độ tin cậy cao, độ an toàn với lỗi và độ chịu đựng hỏng hóc trở thành một phần thiết yếu của thiết kế hệ thống cơ điện tử. Một hệ cơ điện tử là tin cậy nếu không một hỏng hóc hoặc sự cố nào có thể làm hệ thống mất an toàn. Một hệ cơ điện tử là an toàn nếu nó không gây ra thương vong hoặc nguy hiểm cho người vận hành, môi trường và chính bản thân nó. Một hệ cơ điện tử là an toàn với lỗi nếu hệ có thể dừng hoạt động một cách an toàn sau khi có hỏng hóc xảy ra và gọi là có khả năng chịu đựng hỏng hóc nếu hệ thống này có thể hoàn thành nhiệm vụ của mình một cách an toàn sau bất cứ hỏng hóc nào.

Hỏng hóc/lỗi tương ứng với bất kỳ tình trạng hay sự xuống cấp của các thành phần/hệ thống con (xuống cấp đột ngột hoặc từ từ) ảnh hưởng đến sự vận hành của một hệ thống đều sẽ khiến hệ thống không thể hoạt động được như mong muốn. Khi các hệ cơ điện tử được triển khai ứng dụng trong các môi trường động lực học phức tạp/môi trường phi cấu trúc hoặc các môi trường không gian/môi trường phân tán, các ứng dụng trong y học và ứng dụng có tốc độ cao thì hệ cơ điện tử cần phải an toàn hơn (có thể dừng hoạt động mà không làm tổn hại tới môi trường, cho người vận hành và cho chính bản thân thiết bị) và có khả năng chịu đựng hỏng hóc hơn (khắc phục hỏng hóc và hoàn thành công việc đã được thiết lập sẵn).

Một hệ cơ điện tử được gọi là chịu đựng được hỏng hóc nếu hệ vẫn tiếp tục nhiệm vụ/quá trình hoạt động sau bất kỳ hỏng hóc nào. Khả năng chịu đựng hỏng hóc và độ tin cậy cao chỉ có thể đạt được bằng cách sử dụng các phần tử có chất lượng cao, thông qua việc thiết kế và điều khiển bền vững và bằng cách kết hợp chặt chẽ các biện pháp dự phòng trong thiết kế các hệ cơ điện tử. Một hệ cơ điện tử gồm các hệ con như: cơ, điện, máy tính và điều khiển (phần cứng và phần mềm). Bởi vậy, các dự phòng của chúng có thể ở dạng phần cứng (cảm biến, cơ cấu chấp hành, truyền động, truyền thông và tính toán), phần mềm, phân tích, thông tin và dự phòng về thời gian.

39.2 Công cụ phân tích hỏng hóc/độ tin cậy

Có thể chia các phương pháp phân tích lỗi thành các phương pháp phân tích quy nạp và các phương pháp phân tích suy diễn (Wolfe, 1978). Các phương pháp quy nạp, ví dụ như cây sự kiện hoặc quyết định và phân tích các dạng ảnh hưởng và hỏng hóc (FMEA – Failure mode and effects analysis), xét các trạng thái có thể của các thành phần/hệ thống con và xác định ảnh hưởng của chúng lên hệ thống, có nghĩa là chỉ ra những trạng thái không mong muốn. Các phương pháp phân tích suy diễn, ví dụ như phân tích cây lỗi (FTA – Fault tree analysis), bao gồm kiểm tra các trạng thái mong muốn có thể có của toàn bộ hệ thống và chỉ ra các trạng thái thành phần tạo ra trạng thái không mong muốn, tức là mô tả cách thức gây ra trạng thái không mong muốn.

Phương pháp cây sự kiện là diễn tả bằng hình ảnh tất cả các sự kiện (thành công hoặc hỏng hóc) có thể xảy ra trong hệ thống. Tương tự với các phương pháp khác, phương pháp cây sự kiện có thể sử dụng cho các hệ thống có tất cả các hệ con/các thành phần hoạt động liên tục. Phương pháp này cũng được ứng dụng rộng rãi cho các hệ thống có một vài hoặc tất cả các hệ thống con/các thành phần ở chế độ chờ với các phép tính lôgic và chuyển mạch tuần tự như là các hệ thống hướng an toàn (Billinton và Allan, 1983).

Phương pháp FMEA là phương pháp định lượng từ dưới lên được sử dụng để đánh giá thiết kế bằng cách nhận dạng các lỗi có thể có và ảnh hưởng của chúng tới hệ thống, sự xuất hiện của các lỗi và các cách xác định chúng. Phương pháp FMEA đã có từ những năm 1950, khi đó nó được sử dụng cho việc thiết kế và phát triển hệ thống điều khiển thiết bị bay (Dhillon, 1983). Từ đó nó được ứng dụng rộng rãi trong công nghiệp để thiết kế hệ thống với sự hiểu biết về các bộ phận, các hệ thống con, chức năng, thực hiện yêu cầu và các đặc trưng kỹ thuật của chúng, v.v... Phân tích tới hạn (CA - Criticality analysis) là phương pháp định lượng được sử dụng để phân loại các ảnh hưởng tới hạn của các dạng hỏng hóc bằng cách xem xét khả năng xuất hiện của chúng. FMECA là một phương pháp thiết kế được kết hợp từ phương pháp FMEA và CA, đồng thời chỉ ra cách tiếp cận có hệ thống để loại bỏ các hỏng hóc của phần cứng.

Phân tích FTA là một quá trình xét từ trên xuống các thành phần ở trạng thái làm việc hoặc trạng thái lỗi, và nó gặp khó khăn trong việc xử lý các trạng thái của thành phần bị xuống cấp. FTA có thể sử dụng để nhận được các tập cắt tối thiểu, các tập cắt tối thiểu này xác định dạng lỗi của hệ thống và nhận dạng các thành phần tới hạn. Các phép đo độ tin cậy cho các sự kiện trên cùng với phương pháp FTA có thể nhận được từ dữ liệu lỗi của các sự kiện/hỏng hóc nguyên thủy.

39.3 Phân tích hỏng hóc trong hệ Cơ điện tử

Các dạng hỏng hóc của một hệ cơ điện tử bao gồm các hỏng hóc của phần cơ khí, phần điện tử, máy tính và các hệ thống con của phần điều khiển, chúng cũng có thể chia thành các hỏng hóc phần cứng và phần mềm. Việc phân tích những hỏng hóc của các hệ cơ điện tử bao gồm việc phát hiện, nhận dạng (chẩn đoán) và cách ly lỗi của phần cứng, phần mềm và phục hồi (tức thời hoặc từ từ) chúng đòi hỏi điều khiển thông minh.

Lỗi phần cứng có thể được nhận ra một cách dễ dàng bởi thông tin dự phòng của hệ thống và/hoặc bởi việc giám sát thực hiện một nhiệm vụ đã được xác định sẵn của hệ thống. Việc dự phòng thông tin đòi hỏi phải có sự hợp nhất của hệ thống cảm biến và có thể đưa ra thông tin về các trạng thái của hệ thống và các thành phần của nó, về nhiệm vụ đã được ấn định của hệ thống và sự hoàn thành trọn vẹn nhiệm vụ trong trường hợp có lỗi vận hành hoặc bất kỳ thay đổi không mong muốn nào trong môi trường hoặc cho môi trường động lực.

Phương pháp giám sát đơn giản nhất phân biệt hai trạng thái (bình thường hoặc không bình thường) bằng cách sử dụng thông tin/tín hiệu của cảm biến: nếu tín hiệu cảm biến thấp hơn một mức ngưỡng thì trạng thái đó là bình thường và ngoài ra sẽ là không bình thường. Trong hầu hết các ứng dụng thực tế, tín hiệu này dễ bị ảnh hưởng bởi những thay đổi của điều kiện làm việc của hệ thống/quá trình và các tác động của nhiễu với đòi hỏi các phương pháp đưa ra quyết định hiệu quả hơn. Thông thường, các phương pháp giám sát có thể được chia thành hai loại: phương pháp dựa trên mô hình và phương pháp dựa trên đặc tính. Với các phương pháp dựa trên mô hình, việc

giám sát được tiến hành trên cơ sở mô hình hóa hệ thống và đánh giá mô hình. Các hệ tuyến tính bất biến đã được hiểu rõ và có thể được mô tả bằng một số mô hình như mô hình không gian trạng thái, mô hình hàm truyền vào-ra, mô hình tự hồi quy và mô hình tự hồi quy trung bình động (ARMA - Autoregressive Moving Average). Khi một mô hình đã được xây dựng, giám sát có thể được thực hiện bằng việc phát hiện ra những thay đổi của các tham số mô hình (ví dụ: suy giảm và các tần số dao động tự do) và/hoặc những thay đổi của đáp ứng hệ thống yêu cầu (ví dụ: sai số dự báo). Các phương pháp giám sát dựa trên mô hình cũng được gọi là các phương pháp phát hiện lỗi.

Các hệ thống dựa trên mô hình có hai hạn chế đáng kể. Thứ nhất là nhiều hệ thống/quá trình là các hệ phi tuyến, hệ biến đổi với thời gian. Thứ hai là tín hiệu cảm biến thường phụ thuộc nhiều vào điều kiện làm việc. Vì vậy khó mà xác định được thay đổi của tín hiệu cảm biến là do thay đổi điều kiện làm việc hay do nhiễu của quá trình.

Các phương pháp giám sát dựa trên đặc tính sẽ dùng các đặc tính phù hợp của các tín hiệu cảm biến để xác định điều kiện hoạt động. Các đặc tính của tín hiệu cảm biến (thường được gọi là chỉ số giám sát) có thể là các đặc tính miền thời gian và/hoặc tần số của các tín hiệu cảm biến như là giá trị trung bình, sự dao động, độ lệch, hệ số nhọn, hoặc năng lượng của một dải tần số xác định. Lựa chọn các chỉ số giám sát thích hợp thì có tính quyết định. Trường hợp lý tưởng thì các chỉ số giám sát phải là: (i) Độ nhạy với các điều kiện sức khỏe của hệ thống/quá trình, (ii) Độ không nhạy với các điều kiện làm việc và (iii) ảnh hưởng của giả. Khi một chỉ số giám sát đã nhận được thì chức năng giám sát sẽ được hoàn thành bằng cách so sánh những giá trị thu được trong suốt quá trình hoạt động của hệ thống với những giá trị ngưỡng trước đó hoặc trên đường cơ sở đã được xác định trước. Trong thực tế quá trình so sánh này khá phức tạp. Có một số phương pháp giám sát dựa trên đặc tính bao gồm xác nhận mẫu, các hệ thống mờ, các cây quyết định, các hệ chuyên gia và các mạng Nơron.

Quá trình phát hiện và nhận dạng lỗi (FDI – Fault Detection and Identification) trong hệ động có thể đạt được bằng các phương pháp phân tích như bộ lọc dò, tỉ số có khả năng tổng quát hóa (sử dụng cho bộ lọc Kalman để phân biệt sự khác nhau trong đáp ứng của hệ thống), và phương pháp đa chế độ (yêu cầu mô hình động lực học hệ thống và có thể là một vấn đề do tính không chắc chắn của mô hình động lực học hệ thống) (Chow và Willsky, 1984).

Như đã đề cập ở trên, các lỗi hệ thống có thể được xác định và nhận dạng bằng cách nghiên cứu sự khác nhau giữa các chức năng của thông tin đo được từ các cảm biến và các giá trị mong muốn của những chức năng này. Trong trường hợp lỗi, sẽ có sự khác nhau giữa phản ứng quan sát được và phản ứng mong đợi của hệ thống, mặt khác chúng sẽ chấp nhận trong khoảng một ngưỡng xác định. Các phép kiểm tra ngưỡng có thể dựa trên các giá trị tức thời của cảm biến, hoặc trên giá trị trung bình động của các số liệu thu được để giảm nhiễu.

Trong một hệ thống *lựa chọn theo số đông cảm biến*, sự khác nhau ở đầu ra của các cảm biến và của mỗi thành phần (cảm biến hoặc cơ cấu chấp hành) được bao gồm trong ít nhất một quan hệ đại số. Khi một thành phần bị hỏng, quan hệ của thành phần đó sẽ bị thay đổi và những quan hệ không bao gồm nó sẽ được giữ nguyên. Với một hệ thống lựa chọn kiểu này, để an toàn với lỗi và phát hiện được lỗi thì cần phải có ít nhất hai thành phần. Với hệ thống bỏ phiếu hoạt động được với lỗi và nhận dạng được lỗi thì cần phải có ít nhất ba thành phần, ví dụ: ba cảm biến để đo cùng một đại lượng (trực tiếp hoặc gián tiếp). Như Chow và Willsky (1984) đã chỉ ra, để hệ thống phát hiện và nhận biết một lỗi đơn trong m thành phần thì phải biết ít nhất $(m-1)$ mối quan hệ (càng nhiều mối quan hệ càng thực hiện tốt khi có nhiễu).

39.4 Các phương pháp phát hiện lỗi thông minh

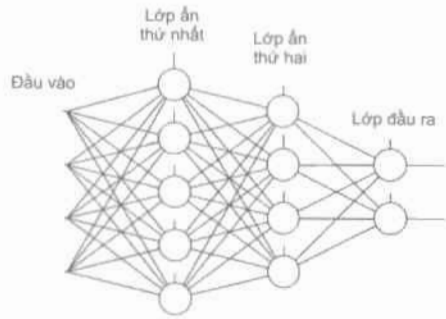
Điều khiển có khả năng chịu được lỗi (điều khiển bền vững và quy trình đưa ra quyết định) phải tính đến sự suy giảm khả năng thực hiện cho phép trong trạng thái lỗi, giá trị tới hạn và khả năng xảy ra hư hỏng, đáp ứng khẩn cấp với lỗi, sự thỏa hiệp giữa việc sửa chữa và tốc độ đáp ứng, phạm vi tiêu chuẩn của tình trạng không chắc chắn của hệ thống, môi trường có nhiễu, độ tin cậy thành phần khi có dự phòng, các mục tiêu duy trì (thời gian trung bình xảy ra lỗi, thời gian trung

bình để sửa lỗi, thời gian bảo dưỡng/thời gian hoạt động, v.v.), kiến trúc hệ thống, các giới hạn can thiệp bằng tay và giá thành (Stengel, 1991).

Xác định lỗi có thể đạt được bằng dự phòng cảm biến (đo lường) và cơ cấu chấp hành, dự phòng song song (ví dụ: sử dụng 2 cảm biến, 2 cơ cấu chấp hành), dự phòng phân tích, và trí tuệ nhân tạo (hệ chuyên gia, mạng Nơron nhân tạo hoặc tích hợp cả hai kỹ thuật) kết hợp với dự phòng.

Stengel (1991) phân loại *dự phòng phân tích* thành *dự phòng trực tiếp* và *dự phòng theo thời gian*. Dự phòng trực tiếp gồm các mối quan hệ đại số giữa các đầu ra tức thời của các cảm biến và nó được dùng để phát hiện lỗi cảm biến, nhưng không để phát hiện lỗi cơ cấu chấp hành. Dự phòng theo thời gian gồm các mối quan hệ giữa giá trị quá khứ của các đầu ra cảm biến và các đầu vào cơ cấu chấp hành (cũng có thể so sánh các đầu ra của cảm biến khác nhau tại các thời điểm khác nhau). Dự phòng theo thời gian được sử dụng để phát hiện và nhận dạng lỗi cho cả cảm biến và cơ cấu chấp hành, ví dụ một hệ lựa chọn cảm biến với sự kết hợp của cảm biến dịch chuyển và cảm biến vận tốc có thể phát hiện lỗi của cả hai loại cảm biến này. Khi yêu cầu động lực học hệ thống thì việc tính toán trong trường hợp dự phòng theo thời gian sẽ phức tạp hơn so với trường hợp dự phòng trực tiếp.

Một hệ chuyên gia trong máy tính là thành phần tri thức của một kỹ năng chuyên nghiệp mà với nó hệ thống có thể đưa ra các quyết định và các lời khuyên thông minh và khi cần, có thể giải thích cho người sử dụng các quy tắc lập luật của nó. Thông thường, một hệ chuyên gia gồm có ba phần sau: một bộ suy diễn, một giao diện người – máy, và một cơ sở tri thức. Bộ suy diễn là một bộ xử lý tri thức và được mô hình hóa khi có các lập luận của chuyên gia. Bộ suy diễn hoạt động với các thông tin sẵn có về một vấn đề cụ thể. Kết hợp với những kiến thức được lưu giữ trong cơ sở tri thức để đưa ra các kết luận hoặc các lời khuyên.



HÌNH 39.1 Cấu trúc điển hình của một mạng nơron truyền thẳng đa lớp

Cơ sở tri thức bao gồm những kiến thức chuyên sâu về các lĩnh vực được cung cấp bởi chuyên gia dưới dạng các phân tích thống kê, các quy tắc thực nghiệm hoặc bán thực nghiệm, các nghiên cứu lý thuyết và mô phỏng trên máy tính và kiểm tra bằng thực nghiệm. Nó bao gồm các cơ sở lập luận, các quy tắc, các khái niệm và các mối quan hệ.

Các hệ chuyên gia có cách trình bày các hiểu biết một cách rõ ràng để dễ dàng cho việc quản lý, có khả năng giải thích ứng xử của chúng, và có thể chẩn đoán những lỗi mới bằng cách sử dụng các cơ sở tri thức của chúng. Trong khi đó, việc tự học vẫn còn là một vấn đề và thời gian tính toán có thể khá dài đối với các nhiệm vụ phức tạp.

Mạng Nơron là một hệ phi tuyến cao với các khả năng thích nghi và tổng quát hóa. Có nhiều cấu trúc mạng Nơron khác nhau, mặc dù vậy mạng Nơron đa lớp truyền thẳng (xem hình 39.1) vẫn là một trong những dạng phổ biến nhất. Bởi vì nó đơn giản, thích hợp với các phương pháp học có hiệu quả, có khả năng tổng quát hóa và khả năng kháng nhiễu. Mạng này là tập hợp các nút kết nối đơn giản, cũng được biết đến như là các nơron, chúng hoạt động song song và lưu trữ dữ liệu bằng cách kết nối giữa các nút độc lập. Giống như mạng tính toán song song, dựa trên cấu trúc tính toán của bộ não con người, đã được ứng dụng thành công với các nhiệm vụ thông minh như: học tập, diễn thuyết tổng hợp và xác nhận mẫu. Véc tơ đầu vào cấp cho mỗi nơron của lớp thứ nhất, các

đầu ra của mỗi lớp này lại cấp cho mỗi nơron của lớp thứ hai và cứ như vậy. Lớp cuối cùng đưa ra đầu ra thế giới bên ngoài, được gọi là lớp ra. Các lớp ẩn là các lớp không được nối với thế giới bên ngoài. Thường thì các nơron được kết nối hết giữa các lớp, có nghĩa là mỗi nơron trong lớp l được nối tới mọi nơron trong lớp l+1.

Huấn luyện một mạng nơron bao gồm quá trình tìm bộ trọng số kết nối (có một trọng số kết nối liên quan với mỗi nơron để thay đổi tín hiệu đầu vào của nơron đó theo một cách nào đó), làm cho đầu ra của mạng thỏa mãn một tiêu chuẩn đặt trước. Các mạng nơron truyền thẳng được huấn luyện bằng cách sử dụng thuật toán lan truyền ngược. Đây là một phương pháp huấn luyện có giám sát. Nghĩa là các đầu vào mẫu và các đáp ứng chính xác, được gọi là mẫu huấn luyện được đưa vào mạng. Sau đó mạng được huấn luyện để chép lại các đáp ứng chính xác đó.

Các mạng nơron có khả năng kết hợp, khả năng nhớ, khả năng sửa lỗi, khả năng tự thích nghi và khả năng xử lý các đa mẫu phức tạp. Tuy nhiên, chúng không thể giải thích các ứng xử của bản thân chúng và không thể chẩn đoán các lỗi mới (những lỗi chưa từng gặp trước đó khi huấn luyện mạng).

39.5 Những vấn đề về phát hiện lỗi thông minh

Giản đồ xác định lỗi phải có khả năng phát hiện và nhận dạng lỗi một cách chính xác và nhanh với thời gian trễ nhỏ nhất. Điều này đòi hỏi một bộ điều khiển bền vững có khả năng tái cấu hình. Tức là, bộ điều khiển có khả năng phân biệt được các lỗi, các điểm sai/những điều không chắc chắn trong mô hình hệ thống, và các nhiễu như là nhiễu của bộ cảm biến, và giảm ảnh hưởng của sai số đo và nhiễu đo, những điều không chắc chắn trong mô hình hệ thống, và các nhiễu (thậm chí lỗi thành phần) ở đầu ra hệ thống.

Nhiễu của các bộ cảm biến có thể được theo dõi bằng phép phân tích thống kê trên các số liệu đọc được từ cảm biến. Có thể giảm thiểu những điều không chắc chắn trong mô hình hệ thống bằng cách ước tính ảnh hưởng của những điều không chắc chắn của tham số và bù trừ cho nó trong hệ thống FDI, hoặc bằng cách tối thiểu hóa độ nhạy của hệ thống FDI đối với những thông số thiếu tin cậy đó. Giản đồ phát hiện lỗi cũng có khả năng giám sát sự xuống cấp của hệ thống cũng như sự phát sinh và phát triển lỗi theo thời gian (và dự đoán lỗi), và đáp ứng một cách phù hợp với từng loại.

39.6 Ví dụ về hệ Cơ điện tử: Tay máy/Máy công cụ song song

Các máy công cụ có kết cấu song song bao gồm nhiều nhánh/chân nối tiếp hoạt động song song trên một tấm di động chung với trục quay được nối với tấm di động. Các tay máy song song có lợi thế là không đòi hỏi cơ cấu chấp hành của các khớp ngoại biên của tấm đế (tấm cố định) và có các khớp chủ động hoạt động song song trên bề mặt di động. Những lợi thế này, có thể khiến cho các máy công cụ song song có thể có độ cứng, độ chính xác và đặc tính động lực học như mong muốn, vì vậy có thể giúp tạo ra tốc độ di chuyển với tỷ lệ loại bỏ vật liệu cao hơn (khối lượng sản phẩm lớn) với dung sai thấp và có khả năng giám sát trong quá trình vận hành (việc đo đạc các thông số trên máy của nguyên liệu mà không phải dừng máy và các công cụ trong và sau quá trình sản xuất sẽ không ảnh hưởng đến quá trình cài đặt ban đầu).

Việc phân tích lỗi của các máy công cụ song song phải bao gồm việc phân tích lỗi của cấu trúc song song cũng như của dụng cụ cắt và những lỗi của phần mềm.

Các tay máy có cấu trúc song song (dựa trên bài báo của Huang và Notash, 1999)

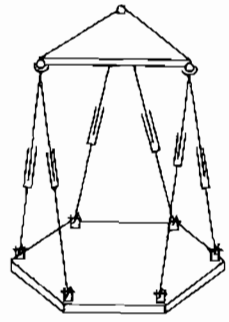
Những bàn luận sau sẽ tập trung vào việc phân tích lỗi thiết kế của hệ thống cơ khí trong các tay máy/máy song song.

Tay máy song song bao gồm một tấm cố định, một tấm di động, và nhiều nhánh/chân liên kết tấm cố định và tấm di động. Hình 39.2 minh họa một ví dụ về tay máy song song sáu chân.

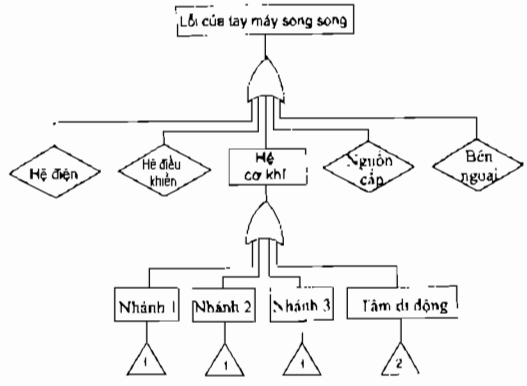
Các dạng lỗi cơ khí của tay máy có thể phân loại thành lỗi khớp (lỗi thành phần), lỗi liên kết/gãy (lỗi thành phần), lỗi chân (lỗi hệ con), lỗi tấm di động và lỗi thiết bị (lỗi hệ thống). Hình 39.3 trình bày FTA mức độ cao của một tay máy/máy song song ba chân.

Hỏng hóc của các bộ phận thành phần

Cơ cấu song song (vòng kín) có cả các khớp chủ động (các khớp đo được và phát động được) và các khớp bị động (các khớp không phát động và có thể đo được hoặc không). Vì vậy lỗi của chúng có thể do lỗi của các khớp chủ động, lỗi của các khớp bị động đo được và lỗi của các khớp bị động không đo được. Lỗi của bất kỳ khớp nào cũng là nguyên nhân dẫn đến lỗi của cơ cấu song song, trừ khi là khớp dự phòng (dự phòng đo hoặc dự phòng chấp hành) cho các nhiệm vụ xác định.



HÌNH 39.2 Ví dụ về tay máy/máy song song 6 chân



HÌNH 39.3 Sơ đồ cây lỗi mức cao của tay máy song song ba chân

Thông thường dạng hỏng hóc của cảm biến chủ động, cảm biến bị động và khớp chủ động không có cảm biến là khớp bị vỡ và khớp bị kẹt. Chỉ duy nhất dạng hỏng hóc của khớp chủ động không có cảm biến là dạng hỏng hóc thông thường của các khớp. Dạng hỏng hóc của các khớp chủ động có cảm biến bao gồm hỏng hóc của cảm biến, còn có các nguyên nhân khác gây nên những dạng hỏng hóc thông thường. Trong trường hợp này, chuyển động của khớp sẽ không thể đo được và khớp có thể chuyển thành khớp chủ động không có cảm biến. Dạng hỏng hóc nguy hiểm nhất của khớp chủ động có thể tương đương với hỏng hóc của động cơ dẫn động, hỏng hóc của bộ truyền động và hỏng hóc của cảm biến. Cũng như kết quả của hỏng hóc cơ cấu truyền động, khớp chủ động suy biến thành khớp bị động có cảm biến, điều đó có nghĩa là khớp có thể phục hồi. Mặt khác, khớp phải được khóa và tương ứng với các nhánh (chân) và tay máy song song có thể mất một bậc tự do và sự hoạt động. Vì sự hỏng hóc của bộ truyền động, hỏng hóc của động cơ dẫn động dẫn động đến các khớp, mà khớp chủ động chỉ có thể sử dụng tương đương như khớp bị động có cảm biến. Khi cảm biến của khớp chủ động bị hỏng, mặc dù hệ thống dẫn động có thể hoạt động tốt, nhưng chuyển động của khớp không thể điều khiển được giống như nó không có sự cung cấp thông tin đáng tin cậy trong chuyển động của khớp, vì vậy khớp chủ động chuyển thành khớp bị động không có cảm biến.

BẢNG 39.1 Các dạng lỗi của tay máy song song và hậu quả của chúng

Mức độ lỗi			Dạng lỗi	Nguyên nhân lỗi	Hậu quả
Bộ phận	Liên kết		Vỡ	Quá tải, mỏi, va chạm, nứt của vật liệu.	Giảm số lượng nhánh
	Khớp	Phổ biến	Vỡ khớp	Quá tải, mỏi, va chạm, nứt của vật liệu.	Giảm số lượng nhánh
			Kẹt khớp	Sự hỏng hóc, tác động bên ngoài.	Giảm số bậc tự do của nhánh tương ứng
		Chủ động	Lỗi cơ cấu chấp hành	Phụ thuộc vào loại cơ cấu chấp hành	Giảm khả năng dẫn động, bậc tự do
			Lỗi truyền động	Phụ thuộc vào kiểu truyền động	Giảm khả năng dẫn động, bậc tự do nếu khớp không thể chuyển động quay trở lại
			Lỗi cảm biến	Phụ thuộc dạng cảm biến.	Giảm khả năng đo, cơ cấu chấp hành có thể bậc tự do.
		Đo được bị động	Hỏng cảm biến	Phụ thuộc dạng cảm biến	Giảm khả năng đo.
		Không đo bị động	Lỗi thông thường (vỡ, kẹt)	Quá tải, mỏi, va chạm, nứt vật liệu, hỏng hóc, do các nguyên nhân bên ngoài	Giảm số lượng các nhánh, giảm số bậc tự do.
Nhánh	Thông thường		Vỡ	Vỡ khớp/liên kết	Giảm số lượng nhánh, có thể là cơ cấu chấp hành và bậc tự do, ảnh hưởng đến các nhánh khác.
			Mất bậc tự do	Kẹt khớp, khóa khớp chủ động, nhánh kỳ dị	Giảm bậc tự do của tay máy
Tay máy			Mất bậc tự do	Kẹt khớp, nhánh kỳ dị	Thiếu bậc tự do
			Mất sự phát động	Lỗi của khớp chủ động/nhánh	Giảm khả năng tải và khả năng chuyển động
			Mất độ hãm	Giảm số lượng của nhánh chủ động.	Không thể điều khiển chuyển động của tay máy
				Nhánh không đo bị động bị gẫy	Không thể điều khiển chuyển động của tay máy
				Cấu hình không tin cậy	Không thể điều khiển được chuyển động của tay máy.

Hỏng hóc của các hệ con

Những nhánh của cơ cấu song song có thể phân loại thành nhánh chủ động hoặc nhánh bị động. Một nhánh chủ động có ít nhất một khớp chủ động để cung cấp lực và giúp cho nhánh cuối cùng dễ dàng phù hợp với quỹ đạo chuyển động. Một nhánh chủ động có cảm biến có ít nhất một khớp có cảm biến và nó có chức năng chính là hạn chế quỹ đạo vị trí cuối cùng của nhánh. Khớp của nhánh bị động không có cảm biến thì sẽ không phát hiện được và nhánh này chủ yếu được sử dụng để hạn chế sự chuyển động của bề mặt di động. Một nhánh của tay máy song song có thể bị lỗi do sự hỏng hóc bộ phận (mối nối, khớp). Cũng như, một nhánh sẽ không thể đi theo quỹ đạo đã được xác định trước nếu nó nằm trong đường bao của không gian làm việc, hoặc nằm tại vị trí kỳ dị (nơi mà mất một hoặc nhiều bậc tự do). Vì vậy, dạng hỏng hóc của phần cơ khí của các nhánh bao gồm vỡ nhánh, mất bậc tự do và mất cảm biến/cơ cấu chấp hành.

Lỗi của hệ thống cơ khí

Một tay máy song song có thể bị lỗi do các lỗi thành phần và/hoặc lỗi hệ con. Vì vậy lỗi phần cơ khí của tay máy song song bao gồm mất bậc tự do, mất phát động, mất giới hạn chuyển động và cấu hình thiếu tin cậy. Tổng kết mức độ, dạng, hậu quả và nguyên nhân của lỗi cơ khí của tay máy song song được trình bày trong bảng 39.1.

Nhận dạng hỏng hóc

Một tay máy có khả năng chịu đựng lỗi là phải có khả năng nhận dạng lỗi cũng như chấp nhận lỗi. Hỏng hóc của bộ phận cấu thành (hệ thống cơ khí) của tay máy song song, ví dụ như hỏng cảm biến khớp, có thể nhận ra thông qua sự điều khiển tay máy bằng cách sử dụng các thông tin được cung cấp bởi các cảm biến của thiết bị. Phối hợp nhận ra hỏng hóc của cảm biến khớp cho các mức hỏng hóc chấp nhận được của tay máy song song, dựa trên việc đo dự phòng của các khớp thay thế và sự so sánh các giải pháp thay thế, được trình bày trong tài liệu của Notash, 2000.

Trong khi hỏng hóc của các khớp chủ động có thể được nhận dạng dựa trên các thông tin do cảm biến cung cấp về khớp tương ứng thì hỏng hóc của các khớp bị động có thể được nhận dạng bằng giám sát toàn bộ quá trình vận hành của tay máy trong phần mềm. Với một tay máy song song cụ thể, tiêu chuẩn xác định lỗi cần được kết hợp trong phần mềm mô phỏng. Ví dụ: sự mất bậc tự do phụ thuộc vào đường bao của không gian làm việc có thể được giám sát (tương tự với sự giới hạn của khớp và can thiệp của các nhánh) và tay máy có thể dừng hoạt động trước khi nó chạm tới đường bao để ngăn cản lỗi tiềm tàng và nguy hiểm đến thiết bị. Cũng như tất cả các cấu hình đặc biệt (loại thiếu chắc chắn) của tay máy cần được nhận biết và sự tiếp cận với những cấu hình đặc biệt này cần được giám sát chặt chẽ khi nó dịch chuyển xung quanh không gian làm việc.

Hỏng hóc có thể chấp nhận được bằng cách dự phòng

Khả năng chấp nhận lỗi của tay máy song song có thể được cải tiến bằng cách sử dụng dự phòng một cách hợp lý. Cảm biến dự phòng có thể được kiểm tra nhằm nâng cao khả năng chấp nhận lỗi của tay máy, đơn giản hóa phân tích thay thế của tay máy này và đơn giản hóa sự hiệu chỉnh của thiết bị này. Dự phòng trong cơ cấu chấp hành có thể cân nhắc để loại trừ những cấu hình không chắc chắn của tay máy song song. Nhiều công việc yêu cầu phương pháp phát triển để nhận ra hỏng hóc thành phần của tay máy song song với các yếu tố của dự phòng và bù những hỏng hóc đó. Đối với tay máy song song, dự phòng có thể kết hợp như là bậc tự do thừa (có thể di động), dự phòng cảm biến và dự phòng cơ cấu chấp hành.

Dự phòng bậc tự do có thể đạt được bằng cách thêm các khớp vào trong tay máy song song. Bậc tự do dự phòng yêu cầu một hoặc nhiều cơ cấu chấp hành trên tay máy song song. Cơ cấu chấp hành được thêm vào không bị coi là thừa bởi hỏng hóc của nó sẽ dẫn đến hỏng hóc của tay máy song song. Dự phòng cảm biến có thể đạt được bằng cách phán đoán các khớp bị động không có cảm biến hiện tại của tay máy song song và thêm vào một nhánh bị động cảm biến dự phòng, hoặc sử dụng một cảm biến ngoại vi như một hệ thống quan sát. Cần chú ý rằng dự phòng thông tin có được bằng cách dự phòng cảm biến, cũng như bằng cách mô tả nhiệm vụ của tay máy, ví dụ như vị trí quỹ đạo cuối cùng trong hệ Đêcác (để xây dựng quỹ đạo của robot và quá trình hoạt động của máy). Dự phòng trong cơ cấu chấp hành có thể được bù bằng cách kích hoạt các khớp bị động của tay máy hoặc thêm vào một nhánh chủ động (cùng việc áp dụng hai cơ cấu chấp hành).

Công cụ giám sát trạng thái

Một yếu tố quan trọng của chức năng điều khiển quá trình tự động là việc nhận dạng lỗi trong thời gian thực của máy cắt, bao gồm cả hao mòn và tình trạng nứt gãy trong khi vận hành máy. Khả năng nhận dạng trực tuyến những hỏng hóc như vậy giúp đưa ra những biện pháp sửa chữa kịp thời, do vậy sẽ đảm bảo được một cách chắc chắn chất lượng cao của sản phẩm (chất lượng về độ chính bề mặt gia công và độ chính xác về kích thước) và ngăn ngừa những nguy hiểm tiềm tàng đối với quá trình vận hành của máy. Cơ sở lý luận của bất kỳ hệ thống giám sát điều kiện công cụ

tự động hóa thời gian thực là tồn tại hoặc các thông số đo lường trực tiếp, gián tiếp liên quan đến việc làm giảm hao mòn hoặc tình trạng gẫy vỡ của công cụ. Nếu thu thập trực tuyến được những thông tin về hao mòn dụng cụ thì nó có thể được dùng để thiết lập chính sách thay đổi công cụ, điều khiển thích nghi, tiết kiệm chi phí của quy trình gia công và tự động hóa hoàn toàn quá trình gia công. Trong trường hợp lý tưởng, hệ thống cần có khả năng phát hiện các cấp độ hao mòn, mà ở dưới những mức đó công cụ cần được thay thế và cũng cần phải có khả năng thích nghi với những thay đổi khá nhỏ ở cấp độ hao mòn. Những tính chất sau cùng sẽ cung cấp cho hệ thống với những khả năng, xu hướng hao mòn của mô hình và dự đoán trước thời gian hoạt động hữu ích còn lại của công cụ (giới hạn hao mòn có thể chấp nhận được). Cùng với sự quan tâm đến vết nứt của công cụ, hệ thống có thể nhận dạng cả những vết nứt nhỏ, hiện tượng sứt mẻ và những hỏng hóc nặng của công cụ. Mặc dù có thể dự báo trước được các hư hỏng đó, nhưng nó khó khắc phục dù đó là mục tiêu trong thực tế, ít nhất là trong khoảng thời gian tới. Số lượng có thể biến thiên được, cái đó quyết định đến sự cố thực sự xảy ra của sự nứt gãy công cụ cũng như tác động lẫn nhau của chúng, và trong nhiều trường hợp thì dụ nó tạo ra khả năng dự báo đáng tin cậy, tốt nhất, và có cái nhìn tổng quan toàn cảnh trong giám sát công cụ.

Kỹ thuật giám sát hỏng hóc dụng cụ cắt

Công cụ giám sát tình trạng hệ thống dựa trên phương pháp trực tiếp hoặc phương pháp gián tiếp của số lượng biên độ hỏng hóc của công cụ. Phương pháp trực tiếp là sử dụng những nguyên nhân gây ra hỏng hóc trực tiếp đến công cụ. Phương pháp trực tiếp thường xuyên thực hiện bằng thị giác, phóng xạ, khí nén hoặc các cảm biến tiếp xúc có thể ứng dụng trong thực tế để đo được các thông số không trực tuyến về hao mòn hoặc gẫy vỡ của công cụ. Tuy nhiên những phương pháp đo lường mức độ hỏng hóc của công cụ một cách trực tiếp như vậy nhìn chung khó có thể áp dụng trong các phân xưởng. Điều này đặc biệt đúng trong những tình huống đòi hỏi khả năng giám sát trực tuyến (thời gian thực). Các phương pháp nhận dạng mức độ hỏng hóc của các công cụ một cách gián tiếp phụ thuộc vào việc đo đạc các thông số gián tiếp liên quan đến điều kiện của lưỡi cắt. Ví dụ lực cắt trong quá trình vận hành máy phụ thuộc vào điều kiện lưỡi cắt của công cụ. Một cách tổng quát, sự hao mòn của dao cắt tăng thêm lực cắt. Vì vậy, việc đo lực cắt của máy trong suốt quá trình hoạt đông cung cấp chỉ số của trạng thái công cụ, ví dụ sự tăng của lực cắt chỉ ra rằng tăng quá trình mòn. Trong thực tế mối quan hệ này rất phức tạp. Các thông số khác đã được nghiên cứu sự thích hợp của nó giống như dụng cụ chỉ ra sự hỏng hóc của máy cắt bao gồm trục mômơ, sự phát ra âm thanh, nhiệt độ của dụng cụ cắt, nhiễu và tín hiệu dao động. Nó cũng có thể trực tiếp đo lực cắt và liệt kê lại những giá trị trạng thái của máy cắt. Trong thực tế, đó là một trong số nhiều phương pháp giám sát không trực tiếp máy công cụ. Theo các thống kê, các tín hiệu lực cắt nhạy cảm với sự hao mòn công cụ hơn so với tín hiệu dao động hoặc thông số công suất. Độ tin cậy của các thông số lực là một lý do khác cho sự áp dụng phổ biến thiết bị giám sát điều kiện làm việc của công cụ. Để sử dụng các số đo lực cắt trong các hệ thống giám sát công cụ thì cần phải gắn các lực này với điều kiện làm việc trực tuyến của công cụ. Tuy nhiên, đo lực cắt bị ảnh hưởng bởi lưỡi cắt và những thay đổi trong điều kiện cắt (tốc độ cung cấp nguyên liệu, tốc độ cắt, và độ sâu gia công), thì trên thực tế, việc phát hiện hỏng hóc máy công cụ thông qua các số đo lực cắt này gặp nhiều thách thức.

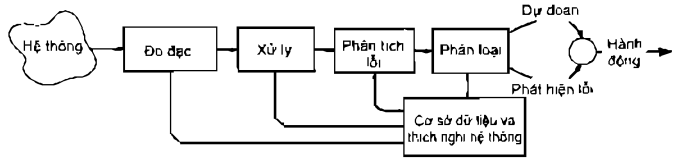
Đặc điểm của hệ thống

Cho dù một hệ thống giám sát điều kiện làm việc của công cụ áp dụng phương pháp đo sự hỏng một cách trực tiếp hoặc gián tiếp (tuỳ thuộc hệ thống tự động dựa trên máy tính hay không), thì nó phải bao gồm một số các đặc trưng thông dụng nếu nó thực sự mang tính khả thi. Hình 39.4 cho thấy sơ đồ khối của một hệ thống giám sát điều kiện làm việc của công cụ phổ biến (Braun, 1986).

Trong phần *đo,* các thông số vật lý mong muốn được chuyển sang một dạng phù hợp với sự vận hành hệ thống một cách thủ công (thường là sự số hóa các tín hiệu tương tự).

Trong phần *xử lý* các kỹ thuật khác nhau được áp dụng nhằm giảm nhiễu, nén thông tin và nhấn mạnh các đặc trưng quan trọng của tín hiệu thu được. Các kỹ thuật điển hình bao gồm kỹ

thuật lọc số hoặc tương tự, trung bình hóa miền thời gian, biên đổi Fourier, nhận dạng thông số dựa trên mô hình ARMA, v. v...



HÌNH 39.4 Sơ đồ khối của một hệ thống giám sát điều kiện làm việc của công cụ phổ biến

Mục đích của phần tách các *đặc trưng* là đạt được một đặc trưng hoặc các đặc trưng cụ thể (thường là các vectơ đặc trưng). Các đặc trưng đó có thể được sử dụng bởi *phần phân lớp* để quyết định dạng hỏng hóc cụ thể hoặc đưa ra phương pháp sửa chữa phù hợp.Ví dụ về các đặc trưng bao gồm: tín hiệu công suất tổng thể, giá trị hệ số nhọn, công suất trong một dải tần số nhất định, tần số đỉnh, biên độ của tần số đỉnh, thông số tự động hồi quy của mô hình ARMA, v.v. Nếu nhiều đặc trưng được áp dụng, chúng cần phải được tách rời nhau để có thể đưa ra các thông số độc lập về tình trạng hỏng hóc của công cụ. Khi được áp dụng kèm với cấu trúc quyết định thứ bậc hình cây (hoặc cấu trúc phù hợp khác) trong *phần phân lớp* thì các vectơ đặc trưng như vậy có thể cải tiến phần lớn độ tin cậy của hệ thống giám sát công cụ.

Mục thích nghi và cơ sở dữ liệu không chỉ quản lý một cách hiệu quả các dữ liệu và các yêu cầu vận hành thủ công mà còn cung cấp cho hệ thống ở mức tối đa có thể khả năng rút ra bài học kinh nghiệm.

39.7 Kết luận và nhận xét

Cần lưu ý rằng bước mang tính thực tiễn nhất trong việc làm tăng độ tin cậy và khả năng chấp nhận lỗi của hệ thống cơ điện tử như máy song song chẳng hạn là cải tiến thiết kế hiện tại hoặc làm tăng sự thích nghi của thiết kế, ví dụ như sử dụng các khớp nối trong khi thiết kế cấu trúc của tay máy. Việc dự phòng trong tái thiết kế được khuyến dùng trong các ứng dụng mà ở đó hệ thống dự phòng đảm bảo an toàn đóng vai trò quan trọng hoặc thời gian hư hỏng cần phải được tối thiểu và lên kế hoạch trước, ví dụ như các ứng dụng trong y học hoặc trong không gian. Điều đáng lưu tâm là không phải dự phòng nào cũng có thể làm gia tăng khả năng chấp nhận lỗi của những hệ thống không có những bổ sung cho cấu trúc của thiết bị.

Cũng cần nhấn mạnh tầm quan trọng của phần mềm mô phỏng dự phòng đảm bảo an toàn và bộ điều khiển cho hệ thống cơ điện tử có khả năng chấp nhận lỗi. Những hệ thống này thường đòi hỏi những phần mềm có độ tin cậy cao có khả năng giám sát sự vận hành của hệ thống và phản hồi lại bất kỳ sự hỏng hóc nào của nó (bao gồm các hệ cơ, điện và điều khiển).

Tài liệu tham khảo

[1] Billinton, R., and Allan, R.N., *Reliability Evaluation of Engineering Systems: Concepts and Techniques*, Plenum Press, 1987.

[2] Braun, S. ed., *Mechanical Signal Analysis - Theory and Applications*, Academic Press, 1986.

[3] Chow, E.Y., and Willsky, A.S., "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. Automatic Control*, 29(7), 603–614, 1984.

[4] Dhillon, B.S., *System Reliability, Maintainability and Management*, Petrocelli Books, 1983.

[5] Huang, L., and Notash, L., "Failure analysis of parallel manipulators," *Proc. 10th IFToMM Congress on Theory of Machines and Mechanisms*, pp. 1027–1032, June, 1999.

[6] Notash, L., "Joint sensor fault detection for fault tolerant parallel manipulators," *J. Robotic Systems*, 17(3), 149–157, 2000.

[7] Stengel, R.F., "Intelligent failure-tolerant control," *IEEE Control Systems*, pp. 14–23, June 1991.

[8] Wolfe, W.A., "Fault tree analysis," Atomic Energy of Canada, Report, 1978.

40

Thiết kế hệ thống lôgic

40.1 Giới thiệu lôgic số 40-1
40.2 Thiết bị bán dẫn 40-3
40.3 Các cổng lôgic 40-5
40.4 Thiết kế lôgic 40-5
40.5 Công nghệ cổng lôgic 40-10
40.6 Mạch tích hợp cổng lôgic 40-14
40.7 Thiết bị lôgic khả trình (PLD) 40-15
40.8 Ví dụ áp dụng Cơ điện tử 40-15

M. K. Ramasubramanian
North Carolina State University

40.1 Giới thiệu về lôgic số

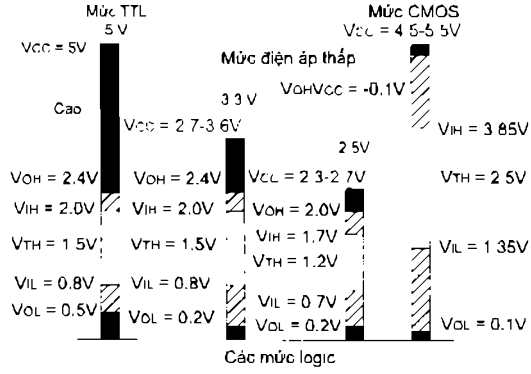
Trong điện tử tương tự, các biến số biểu diễn điện áp và dòng điện biến thiên liên tục từ giá trị thấp nhất đến giá trị cao nhất cho phép. Những biến số này được đo đạc, khuyếch đại, cộng vào và trừ đi thông qua các mạch tương tự để có được kết quả mong muốn. Ví dụ, khi đo nhiệt độ sử dụng cặp nhiệt độ, thì cần phải khuyếch đại điện áp đến một dải đo phù hợp, xác định mức điện áp tương ứng với nhiệt độ đo và đưa kết quả ra vôn kế để biểu thị nhiệt độ. Trong thiết kế này thì cần phải trừ đi một điện áp sai lệch, rồi nhân với một hệ số khuếch đại phụ thuộc vào dải nhiệt độ. Điện áp và dòng điện được khuyếch đại nhờ các bộ khuyếch đại thuật toán và tranzito. Nhiệt độ đo được có thể được sử dụng làm tín hiệu phản hồi trong vòng điều khiển kín của một hệ Cơ điện tử điều khiển nhiệt độ. Trong điện tử số, các biến được biểu thị dưới dạng nhị phân nhận giá trị 0 hoặc 1. Trong ví dụ nói trên, ta cần đóng van sôlênôit lại nếu nhiệt độ xuống dưới giá trị mong muốn và mở van nếu nhiệt độ vượt trên giá trị ngưỡng. Trong trường hợp này, đơn giản là ta cần trả lời ĐÚNG hoặc SAI đối với câu hỏi "nhiệt độ đang ở trên hay dưới ngưỡng?". Chương này sẽ trình bày về các dạng thay đổi này, giả thiết là ở dạng nhị phân và các thao tác để đạt được kết quả mong muốn.

Các mức chuyển đổi Lôgic

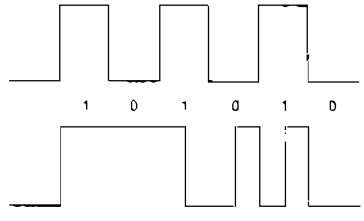
Trong các mạch số, mức điện áp được biểu thị bằng các trạng thái nhị phân. Trạng thái CAO (HIGH) hoặc ĐÚNG (TRUE) được biểu diễn bằng giá trị điện áp lớn nhất, thông thường là 5V và trạng thái THẤP (LOW) hoặc SAI (FALSE) được biểu diễn bằng giá trị điện áp thấp nhất thông thường là 0V. Trong lôgic Boolean, "1" biểu thị ĐÚNG và "0" biểu thị SAI. Trên thực tế, bất kỳ mức điện áp nào trên ngưỡng đầu vào tối thiểu, V_{IH}, thì được coi là lôgic CAO và bất kỳ mức điện áp nào dưới ngưỡng đầu vào tối đa, V_{IL} thì được coi là lôgic THẤP. Đầu ra tối thiểu từ một thiết bị lôgic ở mức lôgic CAO được biểu thị bởi V_{OH} (khác với V_{IH}) và đầu ra tối đa ở mức lôgic THẤP được biểu thị bởi V_{OL} (khác với V_{IH}). Những giá trị này phụ thuộc vào loại thiết bị lôgic, đồ thị biểu diễn giá trị của những thông số này được thể hiện trong hình 40.1 [1]. V_{CC} là điện áp cung cấp. Khoảng giữa V_{OH} và V_{IH} hoặc giữa V_{OL} và V_{IL}, được gọi là giới hạn nhiễu. Điều quan trọng là

phải thiết kế mạch lôgic sao cho điện áp không bao giờ rơi vào vùng giữa V_{IH} và V_{IL}. Vùng này được gọi là vùng cấm mà ở đó thiết bị lôgic không có khả năng nhận dạng tín hiệu. Sự chênh giữa mức chuyển của các công nghệ khác nhau như lôgic 5V, lôgic 3.3V, CMOS (chất bán dẫn oxit kim loại tương hợp) và TTL (lôgic tranzito-tranzito) cần phải được cân nhắc khi ghép những hệ thống này với nhau.

Một biến lôgic có thể thay đổi các trạng thái một cách nhanh chóng như chuỗi xung lý tưởng được biểu diễn trong hình 40.2. Các biến có thể dao động giữa 0 và 1 theo chu kỳ hoặc không theo chu kỳ. Các cổng lôgic nhận các tín hiệu này làm đầu vào, thực hiện các phép tính Boolean giữa chúng và đưa ra đầu ra chính xác với một tốc độ tính toán mong muốn. Thiết kế hoàn chỉnh, sử dụng các hàm lôgic và ứng dụng chúng trong các mạch là một phần không thể thiếu trong thiết kế hệ cơ điện tử.



HÌNH 40.1 Mức chuyển mạch của các cổng lôgic [1]



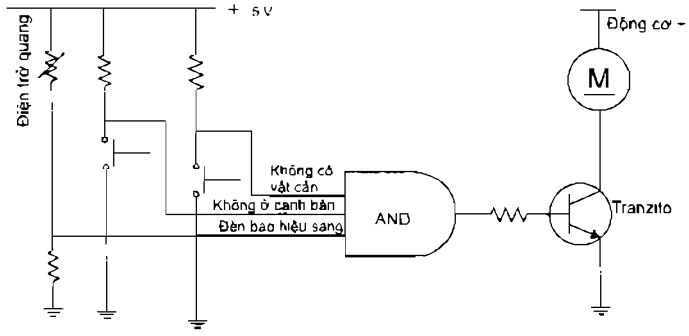
HÌNH 40.2 Các mức lôgic tín hiệu chu kỳ và không chu kỳ

Ứng dụng cổng Lôgic

Hãy xem xét ví dụ về một con rôbôt tự động di chuyển trên bề mặt một chiếc bàn. Con rôbôt phải di chuyển về phía đích được đánh dấu bởi một nguồn ánh sáng và phải tránh các chướng ngại vật cũng như không được rơi ra khỏi bàn. Giả định rằng chúng ta có ba cảm biến số là cảm biến phát hiện chướng ngại vật, cảm biến phát hiện ranh giới và cảm biến phát hiện đích thì chúng ta có thể viết được một hàm lôgic đơn giản biểu thị sự di chuyển về phía trước, như thấy ở phương trình (40.1). Tất nhiên, đây không phải là một hàm lôgic hoàn thiện giúp rôbôt di chuyển hợp lý. Tuy nhiên, ở đây chúng ta chỉ tập trung vào một khía cạnh của vấn đề đó là để minh hoạ cho tính ứng dụng của các hàm.

DI CHUYỂN LÊN PHÍA TRƯỚC = (KHÔNG PHÁT HIỆN CHƯỚNG NGẠI VẬT) AND (KHÔNG PHÁT HIỆN RANH GIỚI) AND (NHÌN THẤY ĐÈN BÁO HIỆU) (40.1)

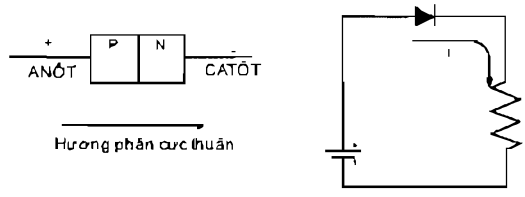
Ba cảm biến này được nối với các đầu vào của một mạch lôgic bao gồm các cổng lôgic, trong ví dụ này là một cổng AND ba đầu vào và đầu ra sẽ điều khiển động cơ. Tất nhiên, các trường hợp khác về hoạt động của rôbôt khi phát hiện ranh giới hoặc chướng ngại vật hoặc khi không có đèn hiệu cần phải được xem xét để làm cho mạch này trở nên hoàn chỉnh và hữu ích. Hình 40.3 là sơ đồ thực hiện các lệnh lôgic được mô tả trong phương trình (40.1).



HÌNH 40.3 Sơ đồ hàm lôgic điều khiển di chuyển lên phía trước của một rôbôt trên bàn

40.2 Các linh kiện bán dẫn Điôt

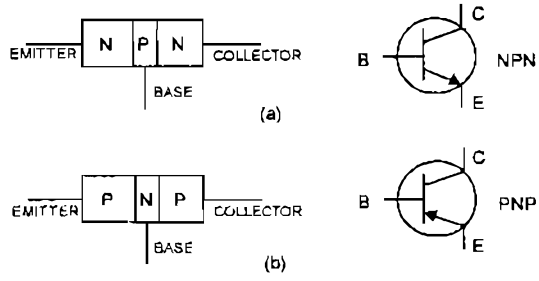
Để hiểu được các cổng lôgic, điều quan trọng là phải nắm bắt được một cách cơ bản nhất về các linh kiện bán dẫn, đặc biệt là điôt và tranzito. Điôt là một lớp tiếp giáp pn, tức là điôt được tạo thành bởi một lớp vật liệu dạng p (thiếu các electron) và một lớp vật liệu dạng n (nhiều các electron) được ghép lại với nhau. Khi cực dương của pin được nối với mặt p của điôt (cực dương - anôt) và cực âm nối với mặt n của điôt (cực âm – catôt) thì điôt được phân cực thuận khi điện áp xuyên qua lớp tiếp giáp phải vượt mức 0.7 V. Khi đảo các cực lại, điôt phân cực ngược và không dẫn điện cho tới khi xuất hiện mức điện áp rất cao xuyên qua lớp tiếp giáp, mức điện áp này được gọi là điện áp đánh thủng. Trong thực tế, ta giả thiết rằng một điôt phân cực ngược thì không dẫn điện. Sơ đồ, ký hiệu và mạch phân cực thuận của điôt được thể hiện trong hình 40.4. Khi phân cực thuận, điôt có thể được coi như một khóa chuyển mạch đóng đơn giản với mức điện áp rơi trên nó là 0,7 V và khi điôt phân cực ngược thì nó đóng vai trò như một khóa chuyển mạch mở.



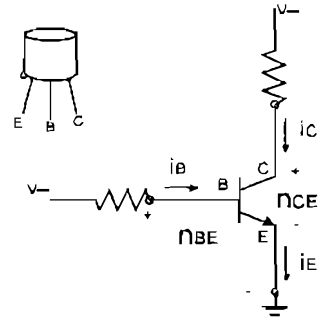
HÌNH 40.4 Điôt và các trạng thái của nó

Tranzito lưỡng cực

Một tranzito lưỡng cực có ba lớp bán dẫn. Trong một tranzito npn, Một lớp p rất mỏng được kẹp giữa hai lớp n. Các loại tranzito và ký hiệu của chúng được thể hiện trong hình 40.5(a, b).

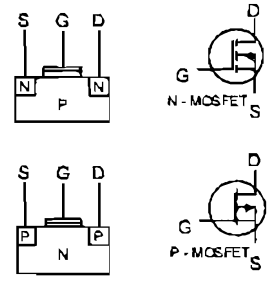


HÌNH 40.5 (a) Ký hiệu tranzito npn, (b) ký hiệu tranzito pnp



HÌNH 40.6 Sơ đồ biểu diễn nguyên lý hoạt động của tranzito npn

Một tranzito có ba cực là collector (C), emiter (E) và base (B). Đối với một tranzito npn như trong mạch hình 40.6, lớp tiếp giáp B-E phân cực thuận, tranzito này sẽ dẫn điện nếu điện áp V_{BE} vượt quá điện áp phân cực thuận của liên kết pn, thông thường là 0.7 V. V_{BE} sẽ tăng lên bằng cách tăng điện áp tại B. Tuy nhiên, lớp tiếp giáp B-C sẽ phân cực ngược khi cực C có điện thế cao hơn. Khi dòng điện chạy trong vòng B-E, các electron từ cực E sẽ di chuyển sang cực B bằng cách lấp đầy các "lỗ trống" của lớp p và sau đó sẽ giải phóng các electron của lớp p ra khỏi cực B. Tuy nhiên, do số lượng các "lỗ trống" trong lớp p (rất mỏng) có hạn, nên các electron di chuyển khỏi cực E sẽ có điện thế trong đường dẫn C-E lớn hơn và sẽ nhảy khỏi lớp tiếp giáp. Kết quả là tạo ra một dòng điện lớn I_C chạy trong vòng C-E. Do đó, tranzito là một bộ khuyếch đại dòng điện. Một dòng điện nhỏ chạy trong vòng B-E, I_B được khuyếch đại với một hệ số khoảng 100 trong đường dẫn B-E. Khi dòng điện trong vòng B-E được tăng lên bằng cách tăng V_{BE}, thì dòng điện C-E tăng lên bằng cách giảm V_{CE}. Do cực C được nối với nguồn điện V_{CC} và cực E được nối với đất, nên tranzito điều khiển dòng điện này bằng cách kiểm soát sự sụt giảm điện áp trên lớp tiếp giáp C-E và tiếp tục giảm điện áp khi dòng điện B-E tăng lên. Rõ ràng là điện áp không thể giảm xuống dưới 0V. Trên thực tế nó không thể giảm xuống dưới mức 0.2-0.35V. Trong những trường hợp như vậy, tranzito được coi là bị bão hòa và đóng vai trò như một khoá đóng. Các mạch chứa tranzito ở trạng thái bão hòa được gọi là mạch bão hòa; ví dụ như các cổng lôgic họ TTL. Các mạch không cho phép các tranzito ở trạng thái bão hòa và tìm được một điểm hoạt động ổn định trong vùng tích cực của tranzito được gọi là các mạch không bão hòa, ví dụ như các cổng lôgic hai cực phát (ELC emitter-coupled logic). Ưu điểm lớn nhất của mạch không bão hòa là ở chỗ so với mạch bão hòa, trạng thái có thể được thay đổi cùng với tốc độ.

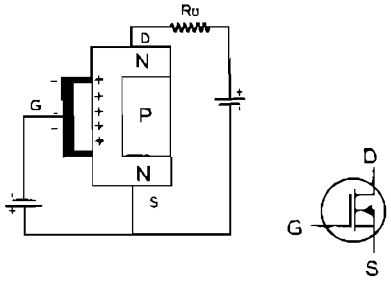


HÌNH 40.7 Các ký hiệu MOSFET dạng kênh n và p

Tranzito hiệu ứng trường (FET)

Những thiết bị này được chế tạo dễ dàng hơn và tiêu tốn ít silicon hơn. Có hai loại FET chủ yếu là FET dạng tiếp giáp (JFET) và FET bán dẫn oxit kim loại (MOSFET). Trong cả hai loại này, dòng điện ra sẽ được điều khiển bởi một điện áp đầu vào nhỏ, gần như không có dòng điện vào. Ba cực được gọi là cực nguồn (S), cực máng (D) và cực cổng (G). Hình 40.7 là các ký hiệu của

MOSFET dạng kênh n và p. MOSFET là một loại tranzito phổ biến nhất. Cổng MOSFET không tiếp xúc điện với cực nguồn và cực máng. Lớp silicon-đioxit cách ly cổng này. Điện áp ở cổng sẽ hút các electron vào vùng dưới của cổng và cung cấp kênh dạng n trong chất nền dạng p để dẫn điện giữa cực máng và cực nguồn. Đây được gọi là dạng mở rộng của MOSFET. Loại thứ hai có kênh n giữa cực máng và cực nguồn, tuy nhiên điện trở kênh có thể được tăng lên hoặc giảm đi bằng cách cấp điện áp âm hoặc dương tương ứng cho cổng. Các chức năng và ký hiệu của MOSFET dạng này được mô tả trong hình 40.8. Thiết bị sử dụng MOSFET có tốc độ chậm hơn các thiết bị lưỡng cực nên được dùng trong các mạch tốc độ thấp nhưng được sử dụng nhiều do dễ chế tạo và tiêu tốn ít silicon.



HÌNH 40.8 MOSFET loại thứ hai

40.3 Các cổng lôgic

Các cổng lôgic là các khối cơ bản trong mạch lôgic và máy tính. Các hệ Cơ điện tử có một khối tính toán trung tâm thực hiện các hàm lôgic đặc trưng trong phần cứng. Một mạch lôgic gồm một vài cổng lôgic cùng hoạt động. Ta sẽ bàn về các cổng lôgic nói chung và các khối cấu thành của hệ Cơ điện tử nhỏ. Các phép toán lôgic có thể được phân thành hai loại là lôgic tổ hợp và lôgic dãy. Trong trường hợp mạch lôgic tổ hợp, kết quả đầu ra của các cổng lôgic chỉ dựa trên giá trị tức thời của các đầu vào, còn ở mạch lôgic dãy, sự thay đổi của đầu ra phụ thuộc vào trạng thái hiện tại cũng như trạng thái trước khi có sự thay đổi giá trị của đầu vào, do đó nó có thuộc tính nhớ. Hơn nữa, các mạch lôgic dãy có thể đồng bộ hoặc không đồng bộ. Khi đầu ra thay đổi một cách đồng nhất với đầu vào đồng hồ thì nó được coi là đồng bộ. Khi các đầu vào được đọc ngay khi mạch có bất kỳ thay đổi nào, thì mạch này được gọi là mạch lôgic không đồng bộ.

Có ba phép toán cơ bản là hàm AND, OR và NOT. Các phép toán lôgic khác là các phép toán bắt nguồn từ các phép toán cơ bản. Ký hiệu cổng AND và bảng chân lý của nó được thể hiện trong hình 40.9. Cổng AND có thể có nhiều hơn hai đầu vào.

Hình 40.10 biểu thị cổng OR. Ở đây, đầu ra có giá trị CAO (HIGH) khi một trong hai đầu vào hoặc cả hai đầu vào CAO. Cổng OR cũng có thể có nhiều hơn hai đầu vào. Hình 40.11 là một cổng đảo, hay còn gọi là cổng NOT. Loại cổng này đảo giá trị nhận được ở đầu vào, ví dụ đầu vào CAO được chuyển thành đầu ra THẤP và ngược lại.

Các cổng lôgic phổ biến khác bắt nguồn từ những phép toán cơ bản này là NAND, NOR và cổng Exclusive OR. Cổng NAND là sự kết hợp của cổng AND và cổng NOT; NOR là sự kết hợp của cổng OR và cổng NOT và Exclusive OR có thể được tạo thành nhờ sự kết hợp của cổng OR, NAND và AND. Từ hình 40.12 đến hình 40.14 chỉ cho chúng ta thấy các dạng cổng NAND, NOR và XOR cũng như các bảng chân lý tương ứng của chúng. Phần cứng sử dụng các dạng cổng này và các hàm lôgic là các khối cơ bản cấu thành của một máy tính số.

40.4 Thiết kế lôgic

Trong bất kỳ thiết kế nào, điều quan trọng là phải làm cho sản phẩm trở nên đơn giản, có tính năng mạnh và mang lại hiệu quả kinh tế. Thiết kế hệ Cơ điện tử và mạch lôgic cũng không là

trường hợp ngoại lệ. Khi một hàm lôgic của một hệ thống được thể hiện dưới dạng các mối quan hệ giữa các đầu vào và đầu ra, thì chúng ta sẽ không biết chắc được rằng liệu số lượng các nhân tố tham gia việc thiết kế có tối thiểu không hay liệu chúng ta có thể đơn giản hoá thiết kế hơn nữa không. Nếu độ phức tạp được xác định bằng số lượng các cổng lôgic được sử dụng, thì vấn đề đặt ra ở đây là cần phải giảm thiểu các hàm lôgic về mặt toán học. Tuy nhiên, nếu độ phức tạp được xác định bằng số lượng các ICs được sử dụng trong mạch (diện tích chiếm trong mạch) thì cần phải cân nhắc thêm, chẳng hạn như cần phải sử dụng các dạng cổng cùng loại càng nhiều càng tốt, mặc dù nó có thể không tối thiểu nếu xét về số lượng cổng. Việc này được ưu tiên hơn việc giảm bớt các dạng cổng khác nhau, do nếu sử dụng nhiều dạng cổng đòi hỏi phải có nhiều IC hơn, mà đôi khi lại không sử dụng hết số cổng.

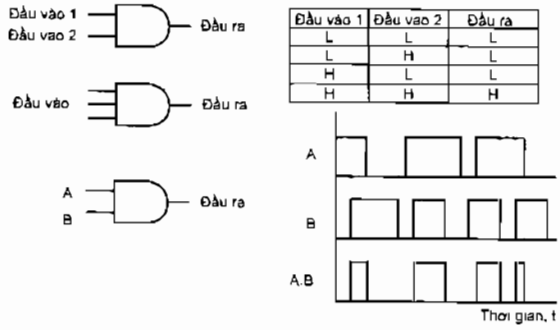
Tối thiểu hoá

Một phương pháp để tối thiểu hoá các hàm Boolean là dùng giản đồ các nô (giản đồ K). Từ những mô tả chức năng vật lý, các lệnh lôgic dùng cho việc điều khiển rôbôt trên bàn được viết như phương trình (40.1). Bảng chân lý thể hiện mối quan hệ giữa các đầu vào và các đầu ra. Chúng ta lập được bảng chân lý cho thiết kế dùng ba biến số, như hình 40.15.

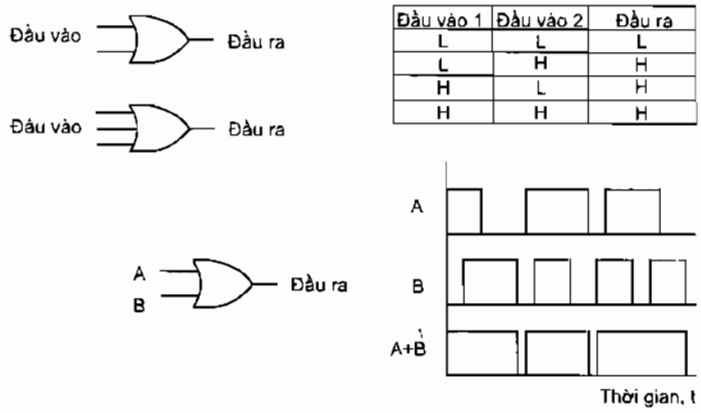
Hàm lôgic có thể được viết như sau:

$$X = A'B'C' + A'BC' + ABC' + A'BC \tag{40.2}$$

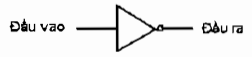
Để thực hiện hàm lôgic trên mà chưa tính đến các yêu cầu khác thì cần phải có bốn cổng AND 3 đầu vào, một cổng OR 4 đầu vào và ba cổng đảo. Nếu giả thiết rằng có thể sử dụng cả hai dạng bù và không bù của tín hiệu cho mỗi biến, chúng ta sẽ thiết kế một mạch tổ hợp hai tầng để thực hiện mục đích đó sử dụng giản đồ các nô, chúng ta có thể tối thiểu hàm Boolean từ đó đơn giản hóa dạng và số lượng của các cổng lôgic cần thiết mà vẫn đảm bảo thực hiện đúng chức năng của mạch lôgic.



HÌNH 40.9 Ký hiệu và đặc tính của cổng AND

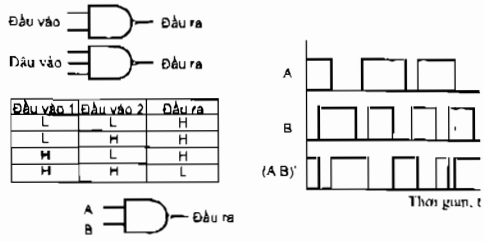


HÌNH 40.10 Ký hiệu và đặc tính của cổng OR



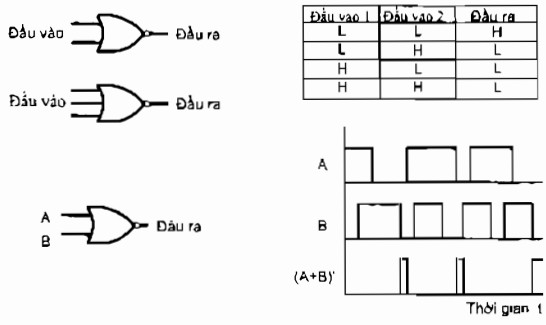
Đầu vào	Đầu ra
L	H
H	L

HÌNH 40.11 Ký hiệu và đặc tính của cổng đảo hay cổng NOT



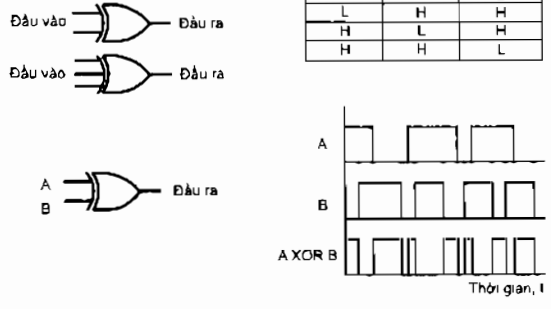
Đầu vào 1	Đầu vào 2	Đầu ra
L	L	H
L	H	H
H	L	H
H	H	L

HÌNH 40.12 Ký hiệu và đặc tính của cổng NAND



Đầu vào 1	Đầu vào 2	Đầu ra
L	L	H
L	H	L
H	L	L
H	H	L

HÌNH 40.13 Ký hiệu và đặc tính của cổng NOR



Đầu vào 1	Đầu vào 2	Đầu ra
L	L	L
L	H	H
H	L	H
H	H	L

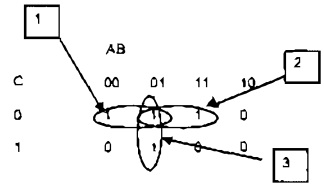
HÌNH 40.14 Ký hiệu và đặc tính của cổng XOR

Giản đồ các nô được xây dựng dựa trên bảng chân lý trong hình 40.15. Hai biến AB được gán cho các cột và biến thứ ba được gán cho hàng. Các giá trị được sắp xếp sao cho các giá trị trên cột và hàng liền kề chỉ khác nhau 1 bit.

Hình 40.16 thực chất là các hàm lôgic tương ứng đã được mô tả trong bảng chân lý trong hình 40.15. Do các ô liền kề trong giản đồ K chỉ khác nhau 1 bit nên sự thay đổi bit là không quan trọng trong một nhóm các ô liền kề. Để tối thiểu hóa hàm lôgic, xác định các nhóm ô liền kề nhau trong giản đồ K và trong mỗi nhóm đó một biến không có vai trò quan trọng đối với giá trị đầu ra X. Biến số được loại trừ và quá trình này được tiếp tục cho đến khi tất cả các nhóm được xem xét.. Cuối cùng, kết hợp các nhóm đã xác định nhờ hàm OR, ta sẽ thu được hàm tối thiểu.

Các đầu vào			Đầu ra
A	B	C	X
0	0	0	1
0	1	0	1
1	0	0	0
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	0

HÌNH 40.15 Bảng chân lý dùng cho thiết kế và tối thiểu hóa mạch lôgic



HÌNH 40.16 Lược đồ Các nô dùng cho thiết kế mạch lôgic có 3 đầu vào và một đầu ra

Hình 40.16 gồm ba nhóm liền kề nhau theo hàng và cột được xác định bởi vòng tròn bao quanh chúng và có thể dẫn tới những nhận xét sau đây :

Nhóm 1: Chỉ một biến B là biến đổi trạng thái, do đó nó có thể được giảm thiểu và dạng tối thiểu cho nhóm là A'C'.

Nhóm 2: Chỉ một biến A là biến đổi trạng thái, do đó nó có thể được giảm thiểu và dạng tối thiểu cho nhóm là B'C'.

Nhóm 3: Chỉ một biến C là biến đổi trạng thái, do đó nó có thể được giảm thiểu và dạng tối thiểu cho nhóm là A'B'.

Do đó dạng tối thiểu cho hàm lôgic là:

$$X = A'C' + B'C' + A'B'$$

Hàm này có thể được thực hiện với một IC cổng AND 2 đầu vào và một cổng 3 đầu vào OR. Phương pháp giản đồ K đã giúp giảm thiểu tới sáu biến.

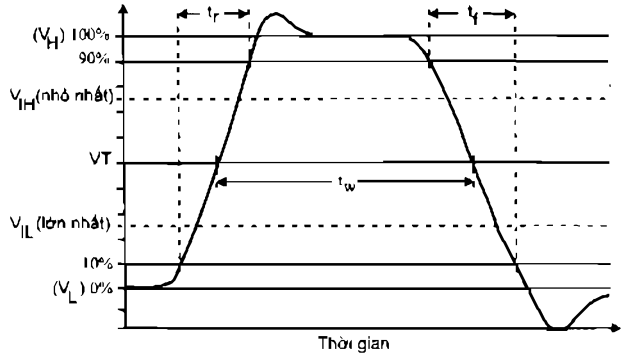
Các đặc tính động học

Bằng việc nghiên cứu hàm lôgic và tìm được hàm tối thiểu hóa, chúng ta có thể xây dựng được mạch lôgic. Tuy nhiên để đảm bảo mạch hoạt động được như mong muốn thì cần cân nhắc kỹ càng các đặc điểm động học của mạch lôgic, toàn bộ vùng hoạt động. Như đã nói ở trên, tín hiệu đầu vào có thể thay đổi nhanh trong một hệ thống và mạch lôgic cần hoạt động như mong muốn ở tần số vận hành của hệ thống.

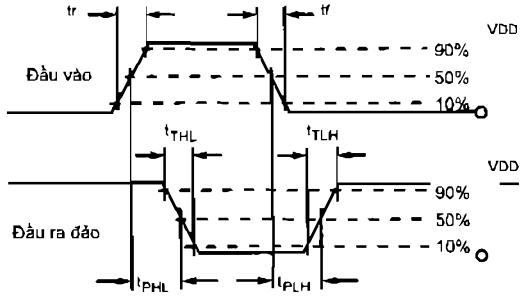
Chức năng hiệu chỉnh của mạch lôgic khi các đầu vào thay đổi nhanh cần phải rất lưu ý trong thiết kế. Như đã trình bày ở trên, chúng ta giả thiết rằng tín hiệu lôgic là một xung vuông lý tưởng và các cổng lôgic không có trễ. Chúng ta hãy kiểm tra tác động của hai giả thiết này bằng cách xem xét kỹ các thuộc tính động học của mạch lôgic.

Một dạng xung thực được thể hiện trong hình 40.17 [2]. Thời gian tăng (rise time) được ký hiệu là t_r và thời gian sụt(fall time) được ký hiệu là t_f. Ngoài ra, xung còn có thời gian quá độ, độ quá điều chỉnh và dưới điều chỉnh khi thay đổi trạng thái. Biên độ tín hiệu là hiệu giữa hai mức ổn định của tín hiệu là mức cao (V_H) và mức thấp (V_L), ví dụ từ 100% đến 0% và t_w là bề rộng xung của tín hiệu được tính ở mức 50% biên độ. t_{THL}, t_{TLH} lần lượt là khoảng thời gian chuyển đổi từ cao xuống thấp và từ thấp lên cao của tín hiệu đầu ra. t_{PHL} và t_{PLH} lần lượt là thời gian trễ truyền dẫn khi chuyển đổi trạng thái từ cao xuống thấp và từ thấp lên cao. Với tốc độ hoạt động trung bình, t_{PHL} và t_{PLH} thường phổ biến ở mức 30 ns.

Khi một đầu vào của cổng lôgic thay đổi trạng thái đầu ra sẽ bị trễ sau bởi thời gian trễ đặc trưng được gọi là thời gian truyền dẫn, nó được đo bằng hiệu thời gian tính từ điểm tương ứng ở mức 50% của biên độ đầu vào đến 50% của biên độ đầu ra. Mô hình đơn giản hoá của xung thực cho một bộ nghịch đảo được thể hiện ở hình 40.18 [3]. Các giá trị truyền dẫn thông thường được tính theo nanô giây, có thể được thấy trong bảng dữ liệu của một số thiết bị.



HÌNH 40.17 Một xung thực và định nghĩa các thông số đặc trưng[2]



HÌNH 40.18 Định nghĩa trễ lan truyền

Đối với mạch lôgic thì việc phân tích thời gian trễ truyền dẫn được thực hiện bằng cách tính tổng thời gian trễ từ đầu vào đến đầu ra khi đầu vào thay đổi trạng thái và xác định với bất kỳ thành phần tĩnh (tần số độc lập) và các thành phần động. Khi thêm các cổng vào mạch có thể sẽ nảy sinh những vấn đề khác.

Các lưu ý khác khi thiết kế

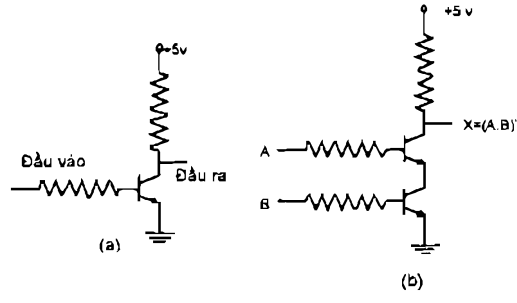
Số lượng cổng lôgic của một cổng cho trước có thể điều khiển được gọi là *fan-out* và số lượng cổng có thể nối đến đầu vào của cổng đó được gọi là *fan-in*. Những số liệu được trình bày trong tài liệu cần được lưu ý. Hơn nữa, tối thiểu số lượng các IC cần thiết trong mạch lôgic thì cần phải tính toán dù có thể yêu cầu phải thay đổi thiết kế và phải dùng nhiều cổng cùng loại hơn so với kết quả tối thiểu hóa giản đồ K. Sử dụng các cổng cùng dạng để tích hợp các IC cũng là một điểm cần cân nhắc khi thiết kế mạch lôgic.

40.5 Các công nghệ cổng lôgic

Họ lôgic đầu tiên được bán trên thị trường là loại lôgic điện trở-tranzito (RTL), tranzito được sử dụng trong mạch là các chuyển mạch tốc độ cao. Tiếp theo là lôgic điôt-tranzito (DTL) và lôgic tranzito-tranzito (TTL) được phát triển. Trong khi loại RTL và DTL lỗi thời thì loại TTL được sử dụng rộng rãi. Có nhiều loại lôgic TTL, cụ thể là: tốc độ cao (H), tốc độ thấp (L), Shottkey (S) và Shottkey công suất thấp (LS). Cổng CMOS lôgic là sự bổ sung hoàn thiện cổng lôgic trên cơ sở công nghệ thiết bị bán dẫn oxit-kim loại (CMOS). Những thiết bị này yêu cầu công suất thấp và cải thiện được đặc tính nhiễu. Những thiết bị này có độ nhạy tĩnh cao và do đó dễ bị hư hại hơn. Sự kết hợp của CMOS với các quá trình lưỡng cực cho ra kết quả là công nghệ BiCMOS, sử dụng các thành phần CMOS bên trong và các đầu ra lưỡng cực công suất lớn. Có nhiều họ BiCMOS khác nhau phát triển từ những quá trình BiCMOS ban đầu [1].

Lôgic điện trở-tranzito (RTL)

Hình 40.19(a) giới thiệu một cổng RTL đảo và hình. 40.19(b) giới thiệu cổng NAND. Tranzito được giả thiết là hoạt động ở chế độ bão hòa. Do bản chất của tranzito, có một điện áp rất nhỏ tổn hao trên lớp tiếp giáp cực góp-cực phát ở chế độ bão hòa và do đó yêu cầu dòng điện rất nhỏ trong vòng cực gốc-cực phát để bão hòa tranzito. Trong hình 40.19(a), điện áp đầu ra nằm trong khoảng 0.3-5.0 V. Đối với cổng NAND được mô tả trong hình, mức lôgic thấp của đầu ra là 0.6 V tương ứng với độ sụt giảm trên cực thu-cực phát của 2 tranzito nối tiếp. Nếu thêm một tranzito để thêm đầu vào, năng lượng tổn hao sẽ tăng lên và mức lôgic THẤP ở đầu ra của mạch là 0.9 V (LOW), đó là nguyên nhân gây ra các vấn đề với các thiết bị lôgic mà nó điều khiển. Hoạt động của hàm lôgic trở nên không tin cậy khi tín hiệu đầu ra có thể rơi vào vùng cấm. Hơn nữa, sự có mặt của điện trở trong vòng cực gốc–cực phát có xu hướng làm chậm thiết bị. Do các đặc tính hạn chế này các cổng RTL trở nên lỗi thời.

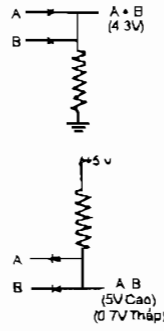


HÌNH 40.19 (a) Cổng điện trở-tranzito NOT, (b) Cổng điện trở - tranzito NAND

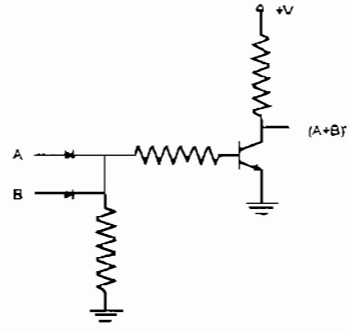
Lôgic Điôt-Tranzito (DTL)

Bản thân các điôt có thể được dùng để xây dựng các cổng lôgic trong các ứng dụng đơn giản như thấy trên hình 40.20. Khi điôt dẫn điện thì điện áp rơi trên lớp tiếp giáp pn thường khoảng 0,7V, kết quả mức lôgic THẤP ở đầu ra là 0.7 V và mức lôgic CAO 4,3 V. Như đã biết, sự nối tầng của một số mạch này sẽ đẩy mạch vào vùng cấm, gây ra lỗi lôgic. Hơn nữa, lôgic điện trở điôt không thể thực hiện được hàm đảo (NOT) và việc tạo ra các IC với mật độ các điôt và điện trở lớn là điều không thể thực hiện được. Do những hạn chế này mà cổng lôgic điôt-điện trở đã trở lên lỗi thời.

Cổng DTL được minh họa trong hình 40.21. Trong đó, các điôt thực hiện hàm OR, tranzito thực hiện hàm NOT để tạo ra cổng NOR. Sự có mặt của điện trở tại cực gốc của tranzito gây ra sự tổn hao năng lượng và làm giảm tốc độ hoạt động. Hình 40.22 minh hoạ một thiết kế điôt-tranzito đã được cải tiến loại trừ điện trở phân cực, nhờ đó tăng tốc độ hoạt động. Các thiết bị DTL đã bị lỗi thời do những hạn chế nói trên.



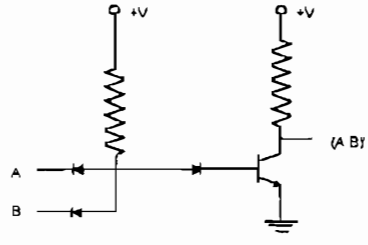
HÌNH 40.20 Lôgic điôt-điện trở



HÌNH 40.21 Cổng lôgic NOR điôt-tranzito

Lôgic Tranzito–Tranzito (TTL)

Từ hình 40.22, chúng ta có thể thấy rằng khi đầu vào ở mức THẤP, các điôt tại đầu vào phân cực thuận trong khi đó điôt tại cực gốc của tranzito phân cực ngược. Nói cách khác, khi tất cả các đầu vào ở mức CAO, điôt tại cực gốc được phân cực thuận và tranzito dẫn điện ,thực hiện chức năng của hàm NAND.

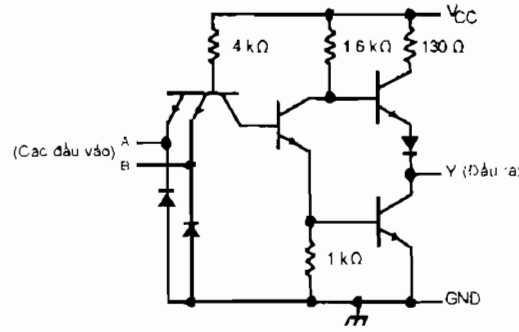


HÌNH 40.22 Cổng lôgic NAND điôt-tranzito có cải tiến

Chẳng hạn, khi một tranzito npn dẫn, cực gốc-cực phát sẽ phân cực thuận còn cực góp-cực gốc phân cực ngược. Nếu chúng ta có một tranzito với nhiều cực phát, thì chúng ta có thể sử dụng lớp tiếp giáp cực phát-cực gốc cho các điôt đầu vào. Lớp tiếp giáp cực gốc-cực gop được dùng cho cực gốc-điôt trong cổng DTL. Kết quả chúng ta có cổng TTL thực hiện chức năng của hàm NAND như thấy ở hình 40.23 [4].

Tại đây, khi có bất kỳ đầu vào nào ở mức THẤP, vòng cực gốc-cực phát sẽ dẫn điện và điện thế tại cực phát của tranzito tầng thứ nhất là 0,2 V, đầu ra của bộ đảo ở mức CAO. Khi tất cả các đầu vào ở mức CAO, thì tranzito nhiều cực phát (tầng thứ nhất) ngắt. Do đó, tất cả những

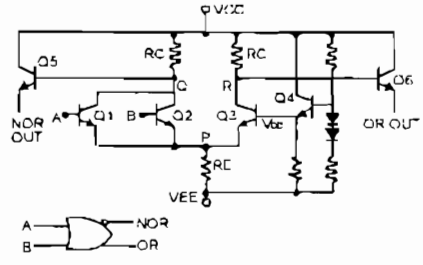
tranzito khác sẽ dẫn điện với đầu ra ở mức lôgic THẤP. Các tài liệu đối với từng thiết bị do nhà sản xuất cung cấp cho chúng ta sơ đồ mạch và tất cả các thông số kỹ thuật bao gồm giá trị đầu vào tối đa và tối thiểu, độ trễ truyền dẫn, thời gian tăng và thời gian sụt, giới hạn đầu vào, giới hạn đầu ra, sự tổn hao công suất và đề suất ứng dụng. Đây là những nguồn thông tin có ích cho các nhà thiết kế.



HÌNH 40.23 Sự thực thi lôgic Tranzito -tranzito của một cổng NAND [4]

Lôgic hai cực phát (ECL)

Thiết bị lôgic hai cực phát (ECL) là thiết bị lưỡng cực trong đó tranzito không bao giờ bão hòa hoặc bị ngắt hoàn toàn. Kết quả là tốc độ của nó cao hơn nhiều so với TTL hoặc CMOS. Các cổng ECL được dùng trong một vài ứng dụng cần tốc độ cao như bộ nhớ lưu trữ trong máy tính. Hình 40.24 minh hoạ cổng NOR/OR [5]. *Vcc* được nối đất (0V) trong khi đó V_{TT} lại được nối với điện áp cung cấp điện -5.2 V để giảm bớt nhiễu. Thời gian chuyển đổi từ trạng thái này sang trạng thái khác là khoảng dưới 1 ns, do đó tần số hoạt động là khoảng một vài gigahec khi sử dụng các cổng ECL.



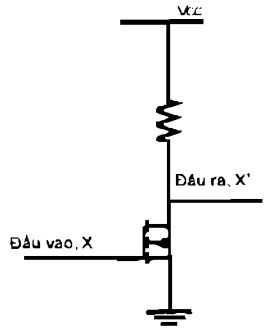
HÌNH 40.24 Cổng lôgic NOR/OR cực phát-cặp đôi [5]

Do các tranzito không bão hòa hoàn toàn, nên các đầu ra của cổng ELC ở mức CAO và mức THẤP tương ứng là khoảng –0.75 và –1.6 V. Điện áp phân cực được đặt tại cực gốc của tranzito Q_4, giá trị của nó là trung bình cộng của giá trị CAO và giá trị THẤP của cổng; trong trường hợp này, giá trị của nó là vào khoảng –1.175 V. Tiếp theo, các điện trở được lựa chọn để điều khiển dòng điện và ngăn chặn tranzito bão hòa. Khi tín hiệu ở A hoặc B là CAO (–0.75 V), thì các tranzito sẽ dẫn. Điện áp tại điểm P sẽ là –1.5 V (V_{BE} của Q_2 + V_4). Điều này làm giảm hiệu giữa V_{bb} và điện áp ở điểm P dưới ngưỡng cho phép để tranzito Q_3 có thể dẫn điện , vì thế nó bị ngắt và điện áp ở điểm R tăng đến 0 V. Tranzito Q6 mở. Với ngưỡng 0,75 V của V_{BE}, đầu ra đo được OR là –0.75 V, đây là mức lôgic CAO. Giá trị của điện trở R_C và R_E được lựa chọn sao cho điện áp ở điểm Q là -0.85 V khi tranzito Q_1 hay Q_2 đang dẫn. Điều này đúng khi B CAO và A THẤP hoặc khi cả *A* và *B* đều cao. Khi cả *A* và *B* đều THẤP, tranzito Q_1 và Q_2 đều bị ngắt và tranzito Q_3 dẫn điện làm giảm điện áp ở điểm *P* xuống -1.925 V. Điện áp ở điểm *R* là –0.85 V khiến cho điện áp ở đầu ra OR là -1.6 V và điện áp ở đầu ra NOR là -0.75 V.

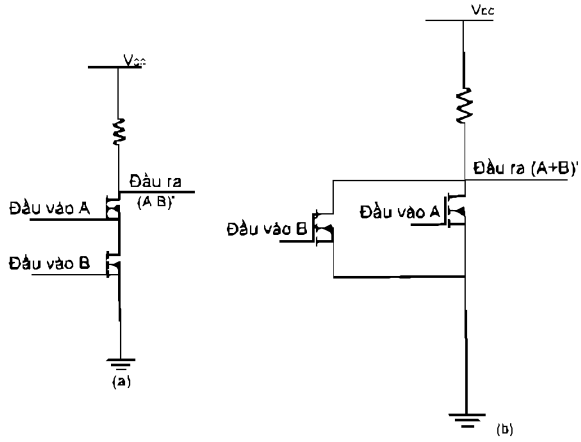
Bởi vì quá trình hoạt động của tranzito là không đổi trong vùng tích cực, ở đó có dòng điện liên tục và gây ra sự tổn hao nhiệt. Thiết bị ECL bằng 4 đến 5 lần công suất so với thiết bị TTL. Vì lý do đó, cần thận trọng khi dùng nó như thiết bị đầu cuối nơi mà tốc độ là yếu tố cốt yếu trong khi sử dụng cổng HCMOS hoặc TTL ở bất kỳ nơi nào khác. Để tích hợp cổng ECL với thiết bị TTL hoặc CMOS, cần sử dụng bộ chuyển dịch mức riêng, ví dụ bộ chuyển đổi ECL sang TTL 100325 công suất thấp của hãng bán dẫn National. Thiết bị chuyển đổi này có đầu vào (-0.75 (H) và –1.6 (L) tới đầu ra TTL (2.4 V min. đối với mức CAO (H) và 0.5 V max. đối với mức THẤP (L)). trong khi duy trì thời gian tăng và thời gian sụt có thể nhỏ hơn 1ns.

CMOS Lôgic

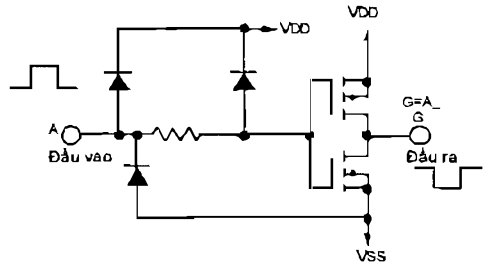
Như đã trình bày ở trên, MOSFET có thể được sử dụng giống như một chuyển mạch tranzito không tổn hao công suất. Cổng lôgic NMOS được thiết kế với MOSFET dạng n và PMOS được thiết kế với tranzito MOSFET dạng p. Ví dụ, một bộ đảo NMOS được giới thiệu trên hình 40.25. Hình 40.26(a) giới thiệu một hàm lôgic NAND với hai MOSFET dạng n nối tiếp và hình 40.26(b) là cổng NOR với hai MOSFET dạng n nối song song. Mạch NMOS có điện trở đẩy và n-MOSFET kéo. Để loại trừ điện trở, phía đẩy của mạch được thay thế bằng p-MOSFET. Mạch đảo hay NOT được biểu diễn trên hình 40.27 [3]. Các điôt thêm vào nhằm đảm bảo sự ổn định của thiết bị. Mạch này được gọi là mạch CMOS do các phần đẩy và kéo của mạch có các MOSFET bổ sung. Khi hai MOSFET dạng n được nối nối tiếp ở phía kéo của mạch, điện trở đẩy được thay thế bằng hai p-MOSFET song song và ngược lại. Cổng NOR dùng CMOS được trình bày trên hình 40.28.



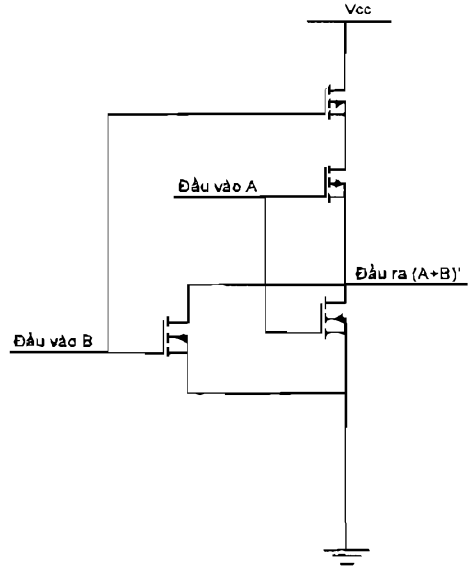
HÌNH 40.25 Bộ đảo NMOS



HÌNH 40.26 (a) Một cổng NMOS NAND, (b) Một cổng NMOS NOR



HÌNH 40.27 Một cổng đảo CMOS [3]

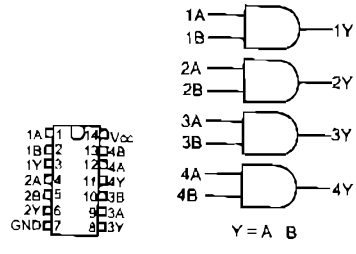


HÌNH 40.28 Một cổng CMOS NOR

Một tính chất quan trọng của cổng CMOS đó là sự tiêu thụ ít năng lượng và thực tế không có dòng điện đi qua ở cả hai trạng thái CAO và THẤP. Tuy nhiên, thiết bị này chậm hơn tranzito lưỡng cực. Với sự tiến bộ của công nghệ sản xuất ngày nay kích thước của tranzito được giảm xuống và tốc độ của CMOS tiếp tục được cải thiện.

40.6 Mạch tích hợp cổng lôgic

Một cổng IC lôgic được bán trên thị trường bao gồm nhiều cổng cùng dạng. Ví dụ, trên hình 40.29 giới thiệu một IC lôgic có 4 cổng AND.



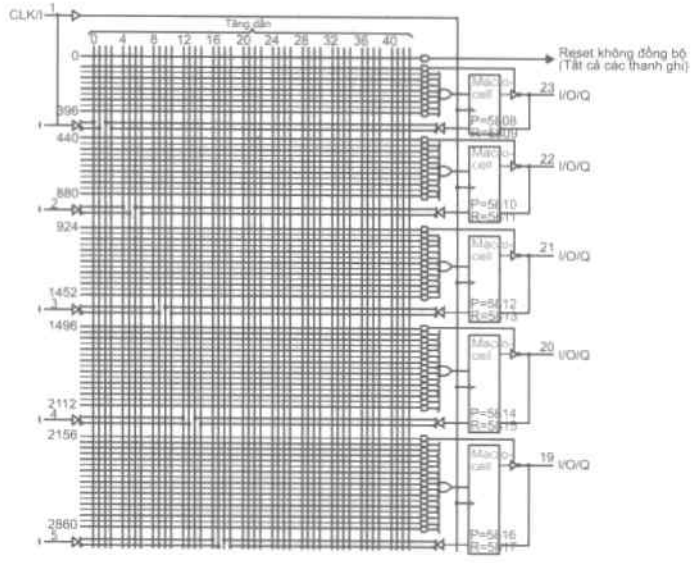
HÌNH 40.29 IC bốn cổng AND dạng DIP

Nguồn nuôi chip được cấp qua các chân V_{cc} và chân nối đất GND, *A* và *B* là các chân đầu vào và Y là chân đầu ra tương ứng. Bạn có thể sử dụng một hoặc tất cả các cổng trên chip nếu cần thiết. Khi lựa chọn chip cần một số lưu ý. Ví dụ khi chọn một cổng AND của Texas Instrumentsta thấy có các loại SN5408, SN54LS08, SN54S08, SN7408, SN74LS08 và SN74S08. Trên bảng thông số của thiết bị có nhiều thông tin hơn những điều được trình bày ở đây, các thông tin cơ bản mà chúng ta cần quan tâm đó là chức năng lựa chọn (00 = cổng NAND, 02 = NOR, 04 = Bộ chuyển đổi, 08 = AND) và loại công nghệ (HC đối với CMOS tốc độ cao, LS đối với Schottky công suất thấp). Đối với các số liệu khác của chip thì có thể tham khảo tài liệu hướng dẫn lựa chọn của hãng Texas Instruments[1].

40.7 Linh kiện lôgic khả trình (PLD)

Các linh kiện lôgic khả trình chính là các IC với một vài cổng lôgic không ràng buộc, các liên kết giữa những cổng này được lập trình dựa trên thiết kế mạch lôgic. Điều đó đặc biệt có ý ích khi xây dựng và kiểm tra một mạch lớn với hàng nghìn cổng lôgic. Để thiết kế và kiểm tra mạch lớn, sẽ là không thực tế khi sử dụng các cổng IC lôgic chuẩn vì mỗi một IC chỉ có tối đa 4 hoặc 6 cổng lôgic nên phải sử dụng một bản mạch lớn kết nối chúng với nhau. Một PLD bao gồm hàng trăm cổng lôgic và thiết kế trên thiết bị này có thể lập trình được nhờ phần cứng lập trình đặc biệt. Khi sử dụng nhiều hơn một PLD cho việc thiết kế, cần lập trình kết nối các PLD. Một dạng trọn vẹn của PLD là mảng lôgic khả trình (PLA) bao gồm một lớp AND ở giữa và một lớp OR ở đầu ra, tương tự như một cấu trúc cổng lôgic đơn TTL, với cả hai lớp có thể lập trình được. Tất cả các tín hiệu đầu vào đều được đi qua một lớp đảo, lớp này sẽ cung cấp cả giá trị chuẩn và giá trị bù của biến đầu vào tới lớp AND. Các kết nối thích hợp được thực hiện ở lớp AND và lớp OR để thu được các kết quả lôgic mong muốn ở đầu ra. Trong thiết bị này tất cả các lớp đều có thể lập trình được..

Một phiên bản đơn giản của PLD được gọi là thiết bị dãy lôgic khả trình(PAL), bao gồm một lớp AND khả trình và một lớp OR cố định. Thiết bị này đơn giản và giảm chi phí chế tạo, mặc dù nó không linh hoạt như một PLA. Có thể kết hợp linh hoạt để cho phù hợp với mọi yêu cầu khác nhau. Sơ đồ của một PLA được giới thiệu trong hình 40.30 [6], ở đó các liên kết phần cứng được đánh dấu X. Khi lập trình các kết nối này được thiết lập hoặc ngắt và được kiểm tra bằng chương trình lập trình cho phần cứng.



HÌNH 40.30 Mảng lôgic khả trình (PLA) [6]

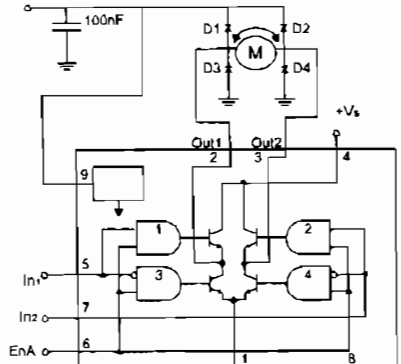
40.8 Ví dụ về ứng dụng thiết kế lôgic trong Cơ điện tử

Mạch điều khiển cho động cơ một chiều là một ví dụ điển hình trong việc sử dụng tranzito và cổng lôgic. Yêu cầu của đối tượng cần thiết kế là:

1. Động cơ có thể quay thuận hoặc ngược ở các vận tốc khác nhau.
2. Động cơ có thể dừng từ từ hoặc có thể hãm đột ngột
3. Có thể điều khiển động cơ quay với những tốc độ khác nhau nhờ bộ vi xử lý.

Một giải pháp thực hiện mạch lôgic và mạch công suất hoàn chỉnh để thiết kế đáp ứng những yêu cầu ở trên được trình bầy trên hình 40.31, đó là một mạch cầu H. Động cơ được nối với các chân đầu ra (đầu ra 1, đầu ra 2) [7]. Chân EN (enable), IN1 (đầu vào 1) và IN2 (đầu vào 2) là các đầu vào. Hoạt động của hệ thống này được mô tả trên bảng tiếp theo hình 40.31. Khi mà tín hiệu cho phép ở mức THẤP, không cần xét đến các trạng thái đầu vào, tất cả các cổng AND đều ở mức THẤP, các tranzito công suất đóng và động cơ sẽ không hoạt động. Nếu động cơ đang hoạt động mà có tín hiệu chuyển mạch thấp thì động cơ sẽ từ từ dừng lại. Khi tín hiệu cho phép ở đầu vào CAO, có thể thấy rằng nếu IN1 CAO và IN2 THẤP thì tranzito 1 và 4 sẽ mở còn 2 và 3 sẽ đóng. Lúc này động cơ sẽ được điều khiển quay theo một chiều và dòng điện chạy qua động cơ xuống đất thông qua 2 tranzito chéo nhau. Do tranzito 2 và 3 đóng nên tránh được việc ngắn mạch nguồn. Việc này được thiết kế bằng cách đưa phần bù của IN1 và IN2 đến cổng AND để điều khiển tranzito 3 và 4. Khi IN2 CAO và IN1 THẤP, động cơ sẽ quay ngược lại (tín hiệu cho phép ở mức CAO),tranzito 2 và 4 đóng và tranzito 1 và 3 mở, dòng điện qua động cơ theo chiều ngược lại.

Khi tín hiệu cho phép ở mức CAO và đầu vào IN1 và IN2 lần lượt CAO hoặc THẤP trong cùng một thời điểm khi động cơ chuyển động, các cực của động cơ cũng được nối với nguồn V_{cc} hoặc đất. Tuy nhiên, động cơ sẽ không hoạt động nếu cả IN1 và IN2 cùng THẤP. Khi đó, các động cơ được coi như một máy phát điện duy trì hiệu điện thế qua các cực của nó khi rôto chuyển động trong từ trường. Sức điện động sinh ra được đưa tới nguồn hoặc suy giảm điện áp. Điều đó làm cho động cơ dừng lại một cách nhanh chóng, việc đó giống như việc hãm nhanh hay phanh. IN1 và IN2 cũng có thể được sử dụng để đổi hướng và phanh trực tiếp, khi tín hiệu cho phép có thể là các xung ở các vòng quét khác nhau (điều chế độ rộng xung) để đạt được các tốc độ khác nhau. Khi động cơ quay tự do tín hiệu cho phép thấp không xét đến đầu vào, khi EN chuyển mạch nhanh, quán tính của roto giúp cho chuyển động đầu ra êm. Việc lựa chọn thời gian lặp lại (PRT) và chọn xung với một PRT nhất định sẽ có được xung PWM điều khiển hoạt động của hệ như mong muốn



Các đầu vào			Chức năng
ENA=H	In1=H	In2=L	Thuận
	In1=L	In2=H	Ngược
	In1=In2		Dừng động cơ nhanh
ENA=L	In1=X	In2=X	Dừng động cơ chạy tự do

HÌNH 40.31 Mạch điều khiển động cơ dạng cầu H [7]

Tài liệu tham khảo

[1] "Logic Selection Guide, First Half 2001," Texas Instruments, Document sdyu001o.pdf. Source: www.ti.com.

[2] "Designing with Logic," Texas Instruments, Document sdya009C.pdf. Source: www.ti.com

[3] "CD4069UB Types- Quad-Inverter," Texas Instruments, Datasheet, schs054.pdf, 1998. Source: www.ti.com.

[4] "SN5400 Quadruple 2-Input Positive NAND-Gates," Texas Instruments, Datasheet, sdls025.pdf, March 1988, Source: www.ti.com.

[5] Koga, R., Crain, W.R., Hansel, S.J., Crawford, K.B., Pinkerton, S.D., Peozin, S.H., Moses, S.C., and Maher, M., "Ion Induced Charge Collection and SEU Sensitivity of Emitter Coupled Lôgic (ECL) Devices," *IEEE Trans on Nuclear Science,* 42(6), 1823–1828, 1995.

[6] High-performance *Impact-X* ™ Programmable Array Lôgic Circuits, TIBPAL22V10-7C, TI, 1995. Product datasheet.

[7] "Dual Full-Bridge Driver L298," SGS Thomson Microelectronics Datasheet. Source: www.st.com.

41

Hệ thống nối tiếp đồng bộ và không đồng bộ

41.1 Tổng quan và định nghĩa 41-1
41.2 Tổng hợp hệ thống nối tiếp đồng bộ 41-3
41.3 Tổng hợp hệ thống nối tiếp không đồng bộ 41-6
41.4 Thiết kế mạch điều khiển và đường dữ liệu 41-10
41.5 Kết luận 41-11

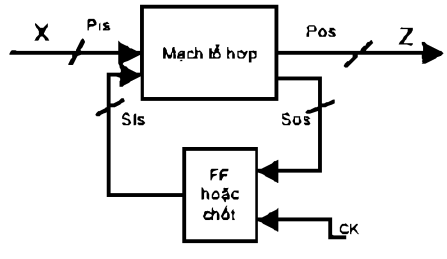
Sami A. Al-Arian
University of South Florida

41.1 Tổng quan và định nghĩa

Trước đây, các hệ thống số được chia làm hai loại chính: *tổ hợp* và *nối tiếp*. Hệ thống tổ hợp là mạch lôgic với các giá trị đầu ra được xác định bởi giá trị của các đầu vào tại thời điểm đó. Mặt khác, hệ thống nối tiếp là các mạch điện mà ở đó đầu ra của tín hiệu không chỉ phụ thuộc vào các giá trị tại thời điểm đó của đầu vào mà nó còn phụ thuộc vào cách ứng xử của mạch điện trước đó. Trong phần lớn các hệ thống, một tín hiệu xung đồng hồ được sử dụng để điều khiển quá trình hoạt động của mạch lôgic nối tiếp. Một hệ thống như thế được gọi là mạch nối tiếp *đồng bộ* và nếu không sử dụng tín hiệu xung đồng hồ thì hệ thống được gọi là *không đồng bộ*.

Hệ thống nối tiếp đồng bộ

Hình 41.1 giới thiệu sơ đồ tổng quát của một hệ thống nối tiếp đồng bộ. Mạch đó có một tập hợp các đầu vào chính $\underline{X}$ và một tập hợp các đầu ra $\underline{Z}$. Thêm vào đó, nó còn có một tập hợp các đầu vào và đầu ra thứ cấp là $\underline{Q}^+$ và $\underline{Q}$. Các tín hiệu này là đầu vào và đầu ra của phần tử trạng thái hoặc của các thiết bị được gọi là *flip-flop* (FF) hoặc *các chốt*. Đầu ra của những thiết bị đó thiết lập trạng thái hiện tại $\underline{Q}$, trong khi đầu vào thiết lập trạng thái tiếp theo $\underline{Q}^+$. Có nhiều loại thiết bị như vậy, cũng như có nhiều biến thể của chúng, cụ thể là set-reset (SR), delay (D), trigger (T) và JK (một sự kết hợp của SR và T) FF và chốt. Bảng 41.1 giới thiệu nguyên tắc hoạt động của chúng.



HÌNH 41.1 Dạng tổng quát của mạch nối tiếp

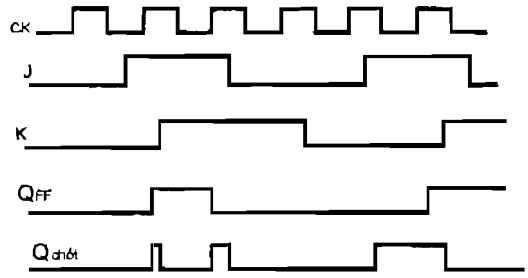
BẢNG 41.1 Nguyên tắc hoạt động của FF đối với các loại SR, D, T và JK

Các đầu vào FF				SR		D		T		JK	
SR	D	T	JK	Q	Q+	Q	Q+	Q	Q+	Q	Q+
00	0	0	00	0	0	0	0	0	0	0	0
00	0	0	00	1	1	1	1	1	1	1	1
01	1	1	01	0	0	0	1	0	1	0	0
01	1	1	01	1	0	1	1	0	0	1	0
10			10	0	1					0	1
10			10	0	1					1	1
11			11							0	1
11			11	Không cho phép						1	0

Lưu ý: Q là trạng thái hiện tại, Q^+ là trạng thái kế tiếp

Flip-Flop và chốt

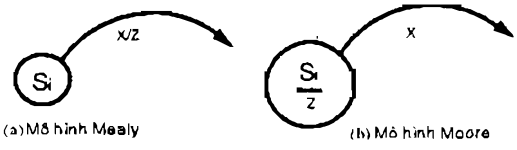
Đầu ra của FF hoặc chốt (là các thiết bị nối tiếp) được xác định bởi các giá trị đầu vào tại thời điểm hiện tại cũng như giá trị trạng thái hiện tại của chúng. Mặc dù vậy, FF là các thiết bị edge-triggered, điều đó có nghĩa là sự thay đổi trạng thái chỉ có thể xảy ra trong vòng một xung nhịp đồng hồ. Sự chuyển đổi này hoặc là sườn dương (chuyển L thành H) hoặc là sườn âm (chuyển đổi H thành L). Tín hiệu xung nhịp đồng hồ tạo ra sự thay đổi trạng thái thường được coi là sườn xung nhịp chủ động. Mặt khác, chốt là thiết bị nối tiếp có thể thay đổi trạng thái bên trong của thiết bị mỗi khi tín hiệu đầu vào (hoặc bộ điều khiển đầu vào) được kích hoạt (có thể là tích cực cao hoặc thấp). Đặc tính này kết hợp với chốt được gọi là *thuộc tính trong suốt*. Hình 41.2 giới thiệu một ví dụ về đồ thị thời gian của JK, FF và chốt JK.



HÌNH 41.2 Đồ thị thời gian của JK, FF và chốt JK (lưu ý thuộc tính trong suốt của chốt)

Mô hình Mealy và mô hình Moore

Mạch nối tiếp cũng có thể được coi là các *máy có trạng thái giới hạn* (FSMs), tức là những mạch như vậy có một số giới hạn các trạng thái để thể hiện những ứng xử của nó. Hơn thế nữa, FSMs được chia làm hai dạng là: *Mealy* và *Moore*. Mạch Mealy là các mạch với đầu ra là $\underline{Z_m}$ phụ thuộc vào trạng thái hiện tại (Q) và đầu vào chính (X). Mặt khác, mạch Moore là mạch mà đầu ra $\underline{Z_m}$ chỉ phụ thuộc vào trạng thái hiện tại (Q). Một FSM tất nhiên có thể có cả hai dạng đó trong một hệ thống.



HÌNH 41.3 Biểu đồ trạng thái cho FSM

Đầu vào dạng mức và xung

Đầu vào của bất kỳ một hệ thống nối tiếp nào cũng có hai loại: *xung* hoặc *mức*. Một xung đầu vào (dù tích cực thấp hoặc cao) là đầu vào tạo ra sự chuyển đổi trạng thái (L thành H hoặc H thành L) và sau đó trở lại về trạng thái không tích cực. Một đầu vào dạng mức là đầu vào tạo ra chuyển đổi trạng thái đơn (L thành H hoặc H thành L) và giữ nguyên trạng thái đó cho đến khi đầu vào thay đổi giá trị. Số lượng trạng thái của hệ thống phụ thuộc vào dạng đầu vào của hệ thống là xung hay mức. Vì vậy, mạch nối tiếp có bốn loại chính sau:

1. *Xung đồng bộ:* Hệ thống nối tiếp có tín hiệu đầu vào dạng xung và thành phần tạo xung nhịp.
2. *Mức đồng bộ:* Hệ thống nối tiếp có tín hiệu đầu vào dạng mức và thành phần tạo xung nhịp.
3. *Xung không đồng bộ:* Hệ thống nối tiếp có tín hiệu đầu vào dạng xung và không có thành phần tạo xung nhịp.
4. *Mức không đồng bộ:* Hệ thống nối tiếp có tín hiệu đầu vào dạng mức và không có thành phần tạo xung nhịp. Đồ thị trạng thái.

Biểu đồ trạng thái

Một *biểu đồ trạng thái* là một công cụ được dùng trong việc tổng hợp mạch nối tiếp. Nó biểu diễn đồ thị chuyển đổi trạng thái của FSM. Mỗi trạng thái được biểu diễn bởi một đường tròn. Nếu máy là loại Moore, thì giá trị đầu ra là sự kết hợp với trạng thái hiện tại. Tuy nhiên, nếu máy là loại Mealy thì giá trị đầu ra là sự kết hợp giữa trạng thái hiện tại và tín hiệu đầu vào. Cả hai loại đó được minh họa ở hình vẽ 41.3. Đầu vào được biểu diễn bằng mũi tên bắt đầu từ trạng thái này đến trạng thái khác. Đối với n đầu vào, số lượng mũi tên hướng ra khỏi mỗi một trạng thái là 2^n đối với đầu vào loại mức và n đối với loại đầu vào dạng xung. Ví dụ, nếu hệ thống nối tiếp có hai đầu vào dạng mức là X_1, X_2, thì sẽ có bốn mũi tên đi ra khỏi mỗi trạng thái biểu diễn các đầu vào 00, 01, 10 và 11. Mặt khác, trong một hệ đầu vào dạng xung, ví dụ trong thiết kế máy bán hàng tự động có các tín hiệu đầu vào là 25 xen (Q), 10 xen (D) và 5 xen (N), thì số lượng mũi tên ra khỏi mỗi trạng thái là 3 biểu diễn các đầu vào Q, D và N.

41.2 Tổng hợp hệ thống nối tiếp đồng bộ

Giả sử phải thiết kế một hệ thống nối tiếp đồng bộ thỏa mãn các yêu cầu sau:

1. Một mạch có bốn đầu vào dạng xung X_1, X_2, X_3 và X_4 và có một đầu ra dạng mức Z.
2. Tất cả các sự thay đổi trong mạch xảy ra trên sườn dương của xung đồng hồ.
3. Đầu ra dạng mức (Z=1) tích cực nếu như các đầu vào tích cực theo trình tự $X_2 X_4 X_3 X_1$.
4. Nếu hai xung liên tiếp của cùng một xung đầu vào xảy ra, thì mạch quay trở lại trạng thái ban đầu.

Các bước thiết kế

Có 6 bước thiết kế đơn giản như sau:

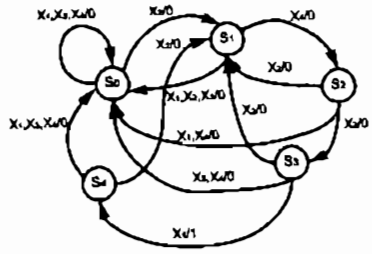
1. Với các yêu cầu đã đặt ra của hệ, bước đầu tiên là xây dựng *biểu đồ trạng thái* (SD). Hình 41.4 giới thiệu đồ thị trạng thái cho bài toán này. Lưu ý rằng có 5 trạng thái (S_0 - S_4). Mỗi trạng thái có bốn mũi tên biểu diễn *n* xung đầu vào (X_1 - X_4). Thêm vào đó, S_0 biểu diễn trạng thái ban đầu. Cũng lưu ý rằng các trạng thái mới được tạo ra tùy theo yêu cầu của hệ thống. Không nhất thiết phải tối ưu hóa số lượng các trạng thái trong giai đoạn này.
2. Bước tiếp theo là chuyển đổi đồ thị trạng thái sang *bảng trạng thái* (ST), như mô tả ở bảng 41.2. Lưu ý rằng bước này là chuyển đổi một-một.

3. Bước tiếp theo là tối thiểu hóa số lượng các trạng thái bằng cách tạo ra *bảng trạng thái rút gọn* (RST). Rất nhiều kỹ thuật có thể được dùng trong bước này, bao gồm kiểm tra, phân chia và bảng tham chiếu. Hai trạng thái được gọi là tương đương (và vì vậy có thể kết hợp lại) nếu (1) chúng cùng tiến đến một trạng thái với tất cả các tín hiệu đầu vào và (2) chúng có những đầu ra giống nhau với tất cả các đầu vào. Khi một trạng thái dư thừa hoặc một trạng thái tương đương được xác định, người ta có thể sử dụng một biểu đồ hợp nhất để kết hợp các trạng thái dư thừa, nơi mà mỗi trạng thái ở trong tập hợp tương đương với tất cả các trạng thái khác trong cùng một tập hợp trạng thái. Trong ví dụ này, trạng thái S_4 là tương đương với trạng thái S_0, như mô tả trong bảng tham chiếu trên hình 41.5. Lưu ý rằng điểm kiểm tra được đưa vào hộp S_0 - S_4 do cả hai trạng thái có trạng thái tiếp theo giống nhau và có các đầu ra giống nhau dưới tất cả các đầu vào. (hình 41.6 giới thiệu một ví dụ biểu đồ hợp nhất trong đó rất nhiều trạng thái được coi là tương đương vì mỗi trạng thái đều tương đương với tất cả các trạng thái khác).
4. Bước tiếp theo là *chỉ định trạng thái* (SA). Chỉ định trạng thái là bước rất quan trọng bởi vì sự chỉ định khác nhau tạo ra việc thực hiện khác nhau và vì thế có giá thành khác nhau. Số lượng chỉ định riêng (N_D) được tính như sau:

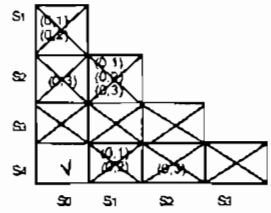
$$N_D = \frac{(2^{N_{FF}} - 1)!}{(2^{N_{FF}} - N_S)!\, N_{FF}!}$$

BẢNG 41.2 Bảng trạng thái của ví dụ thiết kế hệ thống nối tiếp đồng bộ

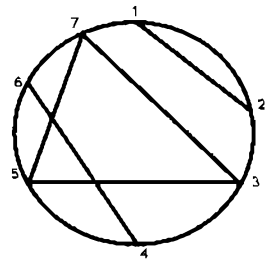
Trạng thái hiện tại	Trạng thái tiếp theo/ đầu ra			
	X_1	X_2	X_3	X_4
S_0	$S_0/0$	$S_1/0$	$S_0/0$	$S_0/0$
S_1	$S_2/0$	$S_0/0$	$S_0/0$	$S_0/0$
S_3	$S_0/0$	$S_1/0$	$S_1/0$	$S_0/0$
S_4	$S_0/0$	$S_1/0$	$S_0/0$	$S_0/0$



HÌNH 41.4 Biểu đồ trạng thái cho ví dụ nối tiếp đồng bộ



HÌNH 41.5 Bảng tham chiếu cho ví dụ đồng bộ



HÌNH 41.6 Ví dụ về biểu đồ hợp nhất: các trạng thái tương đương (1, 2), (3, 5, 7) và (4,6), 7 trạng thái được thu gọn thành 3 trạng thái riêng biệt

BẢNG 41.3 Giá trị đầu vào FF cho các chuyển đổi trạng thái

Q	Q+	S	R	D	T	J	K
0	0	0	d	0	0	0	d
1	0	0	1	0	1	d	1
1	1	d	0	1	0	d	0

Lưu ý: Q là trạng thái hiện tại, Q+ là trạng thái kế tiếp

với N_S là số lượng trạng thái trong RST và N_{FF} là số lượng FF. Cũng cần ghi nhớ rằng $2^{N_{FF}-1} < N_s < 2^{N_{FF}}$, do đó số lượng của trạng thái riêng biệt với 9 trạng thái và 4 FF sẽ vượt quá 10 triệu. Bởi vậy, việc chỉ định trạng thái phải tuân theo một số hướng dẫn để làm giảm việc thực hiện, do đó tối ưu giá thành và độ tin cậy. Sau đây là ba nguyên tắc hướng dẫn được liệt kê theo thứ tự ưu tiên. Mức quan trọng của các nguyên tắc được phân cấp như sau: 5 cho nguyên tắc A, 3 cho nguyên tắc B và 1 cho nguyên tắc C.

Nguyên tắc A: Các trạng thái hiện tại, nếu giống nhau ở những trạng thái tiếp theo với cùng một đầu vào cho trước, được gán những chỉ định liền kề.

Nguyên tắc B: Các trạng thái là những trạng thái tiếp theo của một trạng thái hiện tại với các đầu vào khác nhau, được gán những chỉ định liền kề.

Nguyên tắc C: Các trạng thái hiện tại có các đầu ra tương tự nhau với tất cả các đầu vào, được gán những chỉ định liền kề.

Mục đích của các nguyên tắc này thỏa mãn càng nhiều trạng thái liền kề càng tốt tùy theo những mức quan trọng cho trước. Trong ví dụ được trình bày ở đây, các tập liền kề sau đây đạt được từ nguyên tắc A và B (ở đây bỏ qua nguyên tắc C).

Nguyên tắc A: (S0, S1) × 2 (có nghĩa là 2 lần), (S0, S2) × 3, (S1, S2), (S0, S3) × 3, (S2, S3) × 2, (S1, S3). Nguyên tắc B: (S0, S1) × 3, (S0, S2), (S0, S3), (S1, S3).

Do đó, tổng mức quan trọng là (S0, S1): 19, (S0, S2): 18, (S0, S3): 18, (S1,S2): 5, (S2, S3): 10, (S1, S3): 8. Do đó, sự chỉ định sẽ là: S0 = 00, S1 = 01,S2 = 10 và S3 = 11, trong đó các liền kề (S0, S1), (S0, S2), (S1, S3) và (S2, S3) được thỏa mãn. Mặc dù việc chỉ định trạng thái này không phải là duy nhất, nhưng nó đã đưa ra một phương pháp thực hiện hiệu quả.

5. Trong bước này loại FF được chọn và trạng thái tiếp theo cũng như phương trình toán học đầu ra được xác định. Các phương trình trạng thái tiếp theo được xây dựng từ phương trình đặc trưng hoặc từ bảng chuyển đổi trạng thái, trong đó mỗi đầu vào FF được xác định cho một chuyển đổi trạng thái từ trạng thái hiện tại đến trạng thái tiếp theo. Các phương trình đặc trưng của các FF được giới thiệu trên hình 41.7 và bảng chuyển đổi trạng thái cho mỗi loại được giới thiệu trong bảng 41.3. Trạng thái tiếp theo và phương trình đầu ra cho ví dụ sử dụng JK FF được giới thiệu trong hình vẽ 41.8.
6. Bước tiếp theo là thực hiện. Hình 41.9 giới thiệu việc thực hiện của thiết kế sử dụng JL FF. Cần nhớ rằng mỗi số hạng trong phương trình trạng thái tiếp tương ứng với một đầu

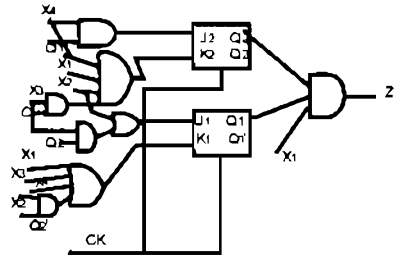
vào dạng xung bằng cách sử dụng các ánh xạ riêng biệt K. Nếu như đầu vào là dạng mức, thì ánh xạ K cho mỗi đầu vào FF bao gồm tất cả các đầu vào chính.

$$Q_{SR}^{+} = S + R'Q \qquad Q_{D}^{+} = D$$
$$Q_{T}^{+} = TQ' + T'Q \quad Q_{JK}^{+} = JQ' + K'Q$$

HÌNH 41.7 Các phương trình đặc trưng FF

$$Q_2^+ = Q_2Q_1'X_3 + Q_2Q_1X_4$$
$$Q_1^+ = Q_1X_2 + Q_2'X_2 + Q_2Q_1'X_3$$
$$Z = Q_2Q_1X_1$$
$$J_2 = Q_1X_4 \qquad K_2 = X_1 + X_2 + Q_1X_3 + X_4$$
$$J_1 = X_2 + Q_2X_3 \qquad K_1 = X_1 + Q_2'X_2 + X_3 + X_4$$

HÌNH 41.8 Trạng thái tiếp theo (Q_2^+, Q_1^+), đầu ra (Z) và phương trình đầu vào JK FF



HÌNH 41.9 Sơ đồ mạch thực hiện ví dụ thiết kế đồng bộ

41.3 Tổng hợp hệ thống nối tiếp không đồng bộ

Các mạch nối tiếp đồng bộ sử dụng xung nhịp đồng hồ để điều khiển quá trình hoạt động toàn bộ của hệ thống. Những mạch nối tiếp đồng bộ này được gọi là hoạt động ở *chế độ xung*. Ngược lại, trong hệ nối tiếp không đồng bộ, sự thay đổi trạng thái của hệ thống không phải do các xung nhịp đồng hồ gây ra. Thay vào đó, sự thay đổi trạng thái của hệ thống phụ thuộc vào sự thay đổi của các đầu vào chính. Tuy nhiên, để có được các thiết kế tốt và tin cậy thì đầu vào của mạch chỉ được thay đổi 1 lần tại một thời điểm, vì vậy những thay đổi đó phải diễn ra đủ nhanh để đạt được một *trạng thái ổn định*. Trạng thái ổn định đạt được khi mà tất cả các phần tử ở bên trong không thay đổi giá trị nữa. Một mạch thỏa mãn các yêu cầu như vậy được gọi là dạng mạch *cơ bản*.

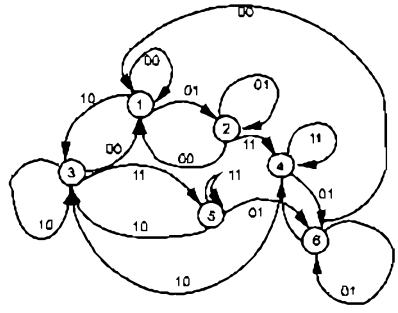
Một trong những thuận lợi chính của mạch không đồng bộ đó là tốc độ làm việc của nó. Do không sử dụng xung nhịp đồng hồ nên tốc độ có thể bằng trễ đường truyền trong một phần của mạch. Vì vậy, hiệu suất của hệ thống được nâng cao. Tuy nhiên, bất lợi chính của hệ không đồng bộ là sự cạnh tranh và rủi ro, cả trạng thái động và tĩnh. Trạng thái cạnh tranh và rủi ro khiến các mạch không đồng bộ rất khó để giải quyết và vì vậy nó phải được thiết kế rất thận trọng.

Việc thiết kế một hệ không đồng bộ được trình bày thông qua các ví dụ sau. Giả sử cần thiết kế một mạch cơ bản với hai đầu vào (X_1, X_2) và một đầu ra Z. Đầu ra Z có thể thay đổi giá trị từ 0 đến 1 khi giá trị của X_2 thay đổi từ 1 đến 0 và X_1=1. Tương tự đầu ra Z có thể thay đổi giá trị từ 1 đến 0 khi giá trị của X_1 thay đổi từ 1 đến 0 và X_2=1. Lưu ý rằng chỉ duy nhất một đầu vào có thể thay đổi giá trị tại một thời điểm. Cũng cần lưu rằng đầu ra chỉ được quan tâm khi trạng thái đã ổn định. Ngược lại, đầu ra sẽ được bỏ qua.

Các bước thiết kế

Tương tự với việc thiết kế hệ đồng bộ, cũng có 6 bước để thiết kế hệ không đồng bộ:

1. Bước đầu tiên là tạo ra *biểu đồ trạng thái ban đầu* (SD) và *lưu đồ gốc* (PFT) của hệ không đồng bộ. Hình 41.10 và bảng 41.4 giới thiệu SD và PFT cho ví dụ này. Ghi nhớ rằng các trạng thái ổn định được khoanh tròn. Thêm vào đó, PFT có thể chỉ có duy nhất một trạng thái ổn định trên một hàng. Cũng cần lưu ý những thuật ngữ mới của mạch không đồng bộ. Bảng trạng thái trong hệ đồng bộ được gọi là bảng dịch chuyển trong hệ không đồng bộ. Do chỉ có một đầu vào được phép thay đổi tại một thời điểm, nên trường hợp nhiều đầu vào cùng thay đổi tại một thời điểm không được khảo sát. Trong ví dụ này, PFT có 6 trạng thái ổn định 1-6.
2. Bước tiếp theo là sử dụng bảng tham chiếu cho PFT, được giới thiệu trên hình 41.11. Bảng tham chiếu đó cho thấy (1,2) (1, 3), (3, 5) và (4, 6) là các mũi tên tương thích. Điều đó có nghĩa là ứng với mỗi đầu vào các trạng thái ổn định sẽ dẫn đến trạng thái tiếp theo và có những đầu ra tương tự nhau và ít nhất chúng cũng không mâu thuẫn với nhau. Do đó, biểu đồ hợp nhất tương ứng chỉ ra rằng PFT có thể được giảm về bảng dịch chuyển 3 trạng thái. Các trạng thái cuối cùng sau phép rút gọn trong bảng flow khi đó là 3 trạng thái không phải nhị phân A, B và C. Trạng thái A là (1,2), trạng thái B là (3,5) và trạng thái C là (4,6). Bảng 41.5 giới thiệu bảng dịch chuyển rút gọn. Trong bảng dịch chuyển này, chúng ta có nhiều hơn một trạng thái ổn định trên một hàng. Lưu ý rằng khi các trạng thái được kết hợp, các đầu vào không được quan tâm được thay thế bởi các trạng thái thực tế ứng với một đầu vào cho trước.



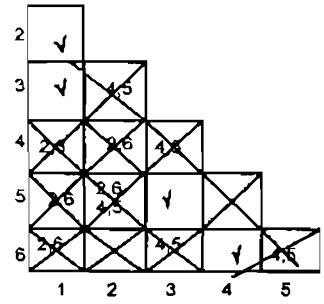
HÌNH 41.10 Bảng dịch chuyển gốc cho một ví dụ thiết kế không đồng bộ

BẢNG 41.4 Bảng dịch chuyển gốc (PFT) cho ví dụ thiết kế không đồng bộ

Đầu vào trước	Trạng thái hiện tại	Trạng thái kế tiếp / Đầu ra			
		X_2X_1 00	01	11	10
00	(1)	(1) /0	2 /-	- /-	3 /-
01	(2)	1 /-	(2) /0	4 /-	- /-
10	(3)	1 /-	- /-	5 /-	(3) /0
11	(4)	- /-	6 /-	(4) /0	3 /-
11	(5)	- /-	6 /-	(5) /1	3 /-
01	(6)	1 /-	(6) /1	4 /-	- /-

BẢNG 41.5 Bảng dịch chuyển rút gọn (RFT) cho ví dụ thiết kế không đồng bộ

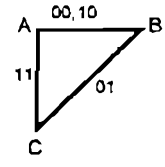
Trạng thái hiện tại	Trạng thái kế tiếp / Đầu ra			
	X_2X_1 00	01	11	10
(A)	(A) /0	(A) /0	C /-	B /-
(B)	A /-	C /-	(B) /1	(B) /0
(C)	A /-	(C) /1	(C) /0	B /-



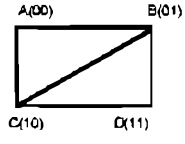
HÌNH 41.11 Bảng tham chiếu cho PFT

BẢNG 41.6 Bảng dịch chuyển (FT) với vòng tròn D được thêm vào để loại bỏ cạnh tranh tới hạn A

Chỉ định trạng thái	Trạng thái hiện tại	Trạng thái kế tiếp/ Đầu ra			
		X_2X_1 00	01	11	10
00	(A)	(A) /0	(A) /0	C /-	B /-
01	(B)	A /-	D /-	(B) /1	(B) /0
10	(C)	A /-	(C) /1	(C) /0	B /-
11	(D)	- /-	C /-	- /-	- /-



HÌNH 41.12 Chuyển đổi trạng thái. AB là chuyển đổi không giới hạn; AC, BC là chuyển đổi có giới hạn



HÌNH 41.13 Chuyển đổi trạng thái thông qua vòng tròn D. Lưu ý rằng B đến C thông qua vòng tròn D

3. Bước tiếp theo là chỉ định trạng thái. Ở đây một lần nữa, mỗi trạng thái phải được gán các chỉ định liền kề nếu có sự chuyển đổi thái giữa bất kỳ hai trạng thái ổn định nào. Nếu có

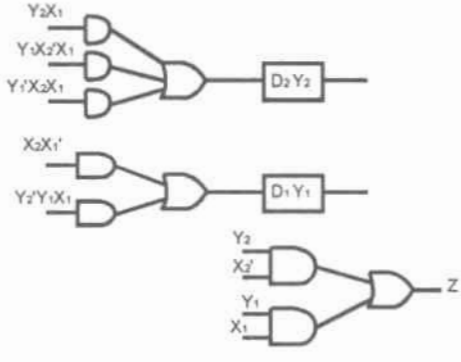
nhiều hơn hai trạng thái ổn định trên một hàng, thì tất cả sự chuyển đổi giữa các trạng thái phải được xem xét cẩn thận. Hình 41.12 giới thiệu tất cả các chuyển đổi tới hạn giữa các trạng thái ổn định. Mỗi đường biểu diễn một sự chuyển đổi với giá trị đầu vào tương ứng của nó trên đường biểu diễn. Lưu ý rằng đầu vào 00 không phải là chuyển đổi tới hạn bởi vì nó chỉ có một trạng thái ổn định (A). Nhưng nếu một chuyển đổi tới hạn tồn tại, chúng ta phải có các chỉ định liền kề để tránh vấn đề của sự cạnh tranh, do chúng ta có thể kết thúc ở nhiều trạng thái ổn định khác nhau nếu xảy ra sự thay đổi của nhiều đầu vào. Chỉ định trạng thái trên hình 41.12 nói lên rằng chúng ta phải có 3 liền kề (A, B), (A, C) và (B, C). Nhưng vì chúng ta chỉ có 2 liền kề với 2 biến, nên chúng ta hoặc đưa ra chỉ định trên trạng thái ổn định hoặc tạo các vòng tròn. Nhược điểm của phương pháp thứ nhất là trong thực tế chúng ta có thể có nhiều mối quan hệ lôgic gây ra bởi các trạng thái mới thêm vào, điều đó có nghĩa là làm tăng giá thành và giảm hiệu suất (ví dụ như tốc độ). Phương pháp thứ 2 tạo ra các vòng tròn. Phương pháp này cũng ảnh hưởng đến hiệu suất với trễ tăng thêm của các vòng tròn. Trong ví dụ này, một vòng tròn cùng với các trạng thái không ổn định được tạo ra giữa các trạng thái B và C để đảm bảo sự chuyển đổi của hai trạng thái. Trong bài toán này, trạng thái D là vòng tròn được tạo ra giữa các trạng thái B và C, như mô tả trên hình 41.13. Do đó, trạng thái B và C chỉ có thể tạo ra sự chuyển đổi giữa chúng thông qua vòng tròn mới tạo ở trạng thái D. Bảng dịch chuyển mới được trình bày trong bảng 41.6.

BẢNG 41.7 Mã hóa kích thích và bảng đầu ra

Trạng thái hiện tại	Trạng thái kế tiếp/ Đầu ra			
	X_2X_1 00	01	11	10
(00)	(00) / 0	(00) / 0	10 / -	01 / -
(01)	00 / -	11 / -	(01) / 1	(01) / 0
(10)	00 / -	(10) / 1	(10) / 0	01 / -
(11)	- / -	10 / -	- / -	- / -

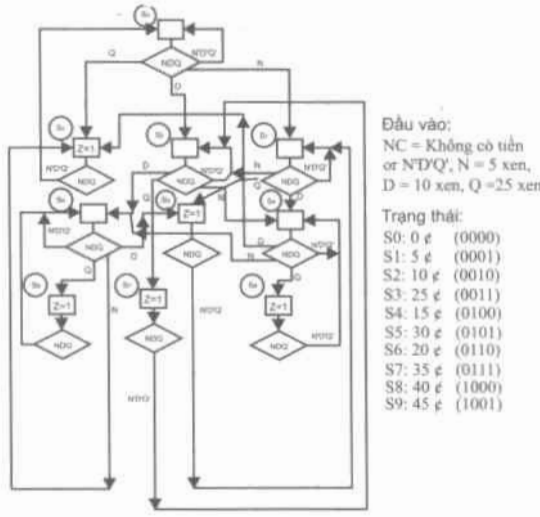
$$Y_2^+ = D_2 = Y_2X_1 + Y_1X_2^+X_1 + Y_1^+X_2X_1$$
$$Y_1^+ = D_1 = X_2X_1' + Y_2'Y_1X_1$$
$$Z = Y_2X_2' + Y_1X_1$$

HÌNH 41.14 Phương trình kích thích và phương trình đầu ra



HÌNH 41.15 Mạch thực hiện ví dụ không đồng bộ

4. Bước tiếp theo là lấy đạo hàm của kích thích mã hóa và bảng đầu ra, được trình bày trong bảng 41.7. Các trạng thái ổn định lại được khoanh tròn. Y_2Y_1 biểu diễn trạng thái hiện tại và $Y_2^+ Y_1^+$ biểu diễn trạng thái tiếp theo.
5. Bước này xác định kích thích tương ứng (hoặc trạng thái tiếp theo) cũng như là phương trình đầu ra, được giới thiệu trên hình 41.14.
6. Mạch lôgic của phương trình trên được trình bày ở hình 41.15.



$$D_4 = Y_4^+ = Y_3Q$$

$$D_3 = Y_3^+ = (Y_4 + Y_3Y_1')NC + (Y_3Y_2' + Y_3'Y_2)N + (Y_2 + Y_1)D + (Y_1 + Y_2Y_3')Q$$

$$D_2 = Y_2^+ = (Y_2Y_1' + Y_3Y_2 + Y_4Y_1)NC + (Y_3 + Y_1)N + (Y_2'Y_1' + Y_3'Y_2)D + Y_3'Y_1'Q$$

$$D_1 = Y_1^+ = (Y_4'Y_2'Y_1)NC + (Y_3Y_2 + Y_3'Y_2'Y_1')N + Y_3D + (Y_2 + Y_3')Q$$

$$Z = Y_4 + Y_2Y_1 + Y_3Y_1$$

HÌNH 41.16 Biểu đồ ASM và đường dữ liệu của thiết kế đồng bộ cho bộ điều khiển máy bán hàng tự động và phương trình đầu vào/ra FF

41.4 Thiết kế mạch điều khiển và đường dữ liệu

Mạch điều khiển có thể được thiết kế bằng cách sử dụng mạch đồng bộ cũng như mạch không đồng bộ. Trong mọi trường hợp, các thiết kế đồng bộ được ưu tiên lựa chọn nhằm tránh sự cạnh tranh và rủi ro. Mạch không đồng bộ không được khuyến khích do rất khó để kiểm soát được các trễ. Tuy nhiên, đôi khi phải sử dụng các mạch không đồng nhất do chúng có tốc độ cao và thỉnh thoảng chúng là một phần của các hệ thống đồng bộ lớn hơn. Một phương pháp thiết kế truyền thống cho bộ điều khiển hoặc xử lý là sử dụng *máy trạng thái thuật toán* (ASM). Một đồ thị ASM là một dạng lưu đồ được sử dụng để mô tả các chuyển đổi trạng thái và các đầu ra của một FSM. Mỗi trạng thái được miêu tả bởi một hình chữ nhật, với các đầu vào được kiểm tra bởi một hộp kim cương. Các đầu ra được hiểu là có điều kiện với việc sử dụng một hộp hình ovan (đầu ra dạng Mealy) hoặc không điều kiện bên trong hộp trạng thái (đầu ra dạng Moore). Các đường dữ liệu của hệ thống được mô tả như là những chuyển đổi từ trạng thái này sang trạng thái khác. Trong một hệ đồng bộ, các chuyển đổi trạng thái xảy ra với các chuyển đổi của xung đồng hồ. Mặt khác, các hệ không đồng bộ có thể có các chuyển đổi trạng thái khi xảy ra sự thay đổi của các đầu vào. Người thiết kế phải phân tích mạch một cách cẩn thận để đảm bảo rằng mạch có thể hoạt động theo các thông số kỹ thuật của nó, đặc biệt khi không thể tránh một số đầu vào không đồng bộ.

Giả sử cần thiết kế mạch điều khiển cho máy bán cà phê. Giá của cà phê là 25 xen. Các mệnh giá 5 xen (N), 10 xen (D) và 25 xen (Q) được chấp nhận. Tuy nhiên không được trả tiền thừa bằng đồng xu. Đầu ra được phân phối ngay lập tức sau khi 25 xen được đưa vào. Hình 41.16 giới thiệu các bước thiết kế, đường dữ liệu và sự thực hiện của một mạch điều khiển đồng bộ.

41.5 Kết luận

Hầu hết các thiết kế số là những hệ thống nối tiếp. Hệ thống đó có thể là đồng bộ hoặc không đồng bộ. Hệ thống đồng bộ có bộ đếm thời gian để điều khiển quá trình hoạt động của hệ thống. Hoạt động của hệ thống này phụ thuộc vào tốc độ của bộ đếm thời gian đó. Nhưng các hệ thống đồng bộ tránh được những vấn đề rủi ro và cạnh tranh. Mặt khác, các mạch và bộ điều khiển không đồng bộ thì nhanh hơn nhưng lại tồn tại vấn đề cạnh tranh. Sự cạnh tranh có thể xuất hiện khi một chuyển đổi trạng thái yêu cầu có sự thay đổi đồng thời của 2 hoặc nhiều các biến trạng thái. Cạnh tranh xảy ra giữa các biến để xác định biến nào thay đổi trước. Một sự cạnh tranh khốc liệt có thể buộc mạch kết thúc ở các trạng thái ổn định khác nhau. Có thể loại trừ cạnh tranh khốc liệt bằng cách nghiên cứu và phân tích mạch một cách cẩn thận.

42
Kiến trúc

Daniel A. Connors
University of Colorado at Boulder

Wen-mei W. Hwu
University of Illinois at Urbana-Champaign

42.1 Giới thiệu 42-1
42.2 Các loại vi xử lý 42-1
42.3 Các bộ phận chính của vi xử lý 42-2
42.4 Cấu trúc tập lệnh 42-13
42.5 Lệnh song song 42-14
42.6 Những xu hướng trong công nghiệp 42-18

42.1 Giới thiệu

Công nghiệp vi xử lý được chia thành công nghiệp máy tính và công nghiệp sản phẩm nhúng. Máy tính và bộ vi xử lý nhúng có nhiều điểm thiết kế giống nhau như cấu trúc tập lệnh, cấu tạo và phần cứng. Thuật ngữ "cấu trúc máy tính" được sử dụng để mô tả các điểm cơ bản đó mà cụ thể là các bộ phận phần cứng trong một hệ thống máy tính, dòng dữ liệu và thông tin điều khiển giữa chúng. Chương này sẽ mô tả các dạng khác nhau của bộ vi xử lý, các cơ cấu kiến trúc cơ bản liên quan đến sự vận hành của tất cả các bộ vi xử lý và bàn về xu hướng phát triển của ngành công nghiệp vi xử lý.

42.2 Các loại vi xử lý

Bộ vi xử lý máy tính được thiết kế để làm đơn vị xử lý trung tâm (CPU) của hệ thống máy tính, ví dụ như: máy tính cá nhân, trạm làm việc, máy chủ và các siêu máy tính. Mặc dù vào những thập niên 1970, bộ vi xử lý giống như là một bộ điều khiển có thể lập trình được, nhưng gần như tất cả các hệ thống máy tính được chế tạo vào những thập niên 1990 đều sử dụng bộ vi xử lý như là đơn vị trung tâm xử lý của chúng. Một cấu trúc nổi bật trong lĩnh vực vi xử lý của máy tính ngày nay là cấu trúc Intel 32-bit được biết đến như là IA-32 hoặc X86. Các cấu trúc ưu việt khác trong lĩnh vực vi xử lý máy tính là Compaq-Digital Alpha, HP PA-RISC, Sun Microsystems SPARC, IBM/Motorola PowerPC và MIPS.

Bộ vi xử lý nhúng được sử dụng rất nhiều do nhu cầu của người dùng và các sản phẩm viễn thông nhằm đáp ứng những yêu cầu chất lượng và chức năng. Các sản phẩm yêu cầu bộ vi xử lý nhúng bao gồm tivi kỹ thuật số, máy ảnh kỹ thuật số, bộ chuyển mạch, môđem tốc độ cao, điện thoại cầm tay kỹ thuật số, máy in laze và ôtô. Những cải tiến trong tương lai về tiêu thụ năng lượng, giá thành sản xuất và khả năng hoạt động có thể mang đến nhiều ứng dụng khác, ví dụ như máy trợ thính. Các chuyên gia hy vọng rằng bộ vi xử lý nhúng sẽ là một lĩnh vực phát triển nhanh nhất của ngành công nghiệp bán dẫn trong thập niên tới [1].

Bộ vi xử lý được chia thành bộ xử lý DSP và CPU. Bộ xử lý DSP được thiết kế và đưa ra thị trường như là một thiết bị có ý nghĩa đặc biệt, nó phần lớn đã được lập trình bằng tay để thực hiện

công việc xử lý các tín hiệu số. Xu thế hiện nay trong lĩnh vực DSP là sử dụng trình biên dịch để giảm bớt công việc mã hóa bằng tay. Một xu thế khác là tích hợp DSP với các ứng dụng lôgic đặc biệt tạo thành một giải pháp chip đơn lẻ. Xu thế này là khả thi nhờ sự phát triển nhanh chóng của công nghệ tích hợp chip. Lợi ích chính là giảm bớt giá thành của hệ thống và khả năng tiêu hao năng lượng. Ngày nay, có hai dạng lõi DSP thường dùng. Foundry-Captive DSP và dịch vụ thiết kế lôgic ứng dụng có liên quan được cung cấp bởi các hãng như Texas Instruments, Lucent Technologies và SGS-Thompson. Các công ty cỡ nhỏ hoặc trung bình thường cung cấp Licensable DSP cho các nhà phát triển ứng dụng, những người muốn có quyền lựa chọn giữa các kênh (nhà sản xuất) khác nhau.

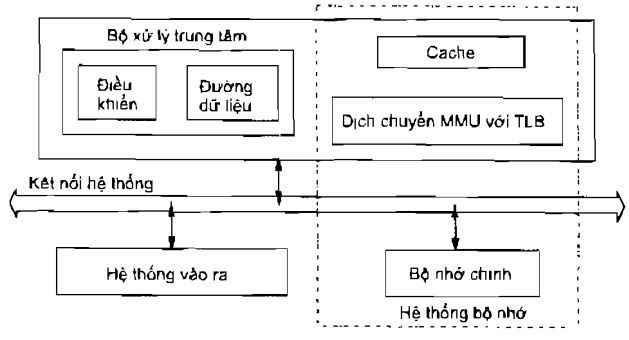
Có rất nhiều lý do khiến nhu cầu về máy tính nhúng khác với nhu cầu về các hệ thống đa năng truyền thống. Nhu cầu sử dụng bộ vi xử lý nhúng bị ràng buộc nghiêm ngặt bởi số lượng mã, khối lượng và khả năng tiêu thụ năng lượng. Những yêu cầu về thời gian thực cũng thường được cân nhắc trong nhiều hệ thống nhúng. Và giá thành cũng là một ràng buộc trong bộ vi xử lý nhúng.

CPU nhúng được sử dụng trong các sản phẩm mà việc tính toán phức tạp tương tự như các ứng dụng đa năng và các hệ điều hành. CPU nhúng thường được lấy từ các vi xử lý máy tính đã lỗi thời. Chúng thường tái sử dụng trình biên dịch và các phần mềm hỗ trợ vốn được phát triển cho các dòng máy tính tương đương. Tái sử dụng thiết kế vi xử lý và phần mềm biên dịch sẽ làm giảm giá thành chế tạo. Một xu hướng trong lĩnh vực CPU nhúng tương tự với lĩnh vực DSP, đó là cung cấp lõi CPU nhúng và thiết kế ứng dụng lôgic đặc biệt để tạo thành các giải pháp chip đơn lẻ. Ví dụ như MPI sử dụng lõi CPU nhúng trong Nintendo64 để bù đắp cho các chi phí liên quan đến kỹ thuật và chi phí bản quyền. ARM, NEC và Hitachi cung cấp các dịch vụ và sản phẩm tương tự. Do nhu cầu thực hiện tính toán DSP trong các sản phẩm tiêu dùng và sản phẩm viễn thông ngày càng gia tăng nên số lượng CPU nhúng cũng tăng lên nhằm đảm bảo việc tính toán DSP hiệu quả hơn.

Máy tính và bộ vi xử lý nhúng đều dùng các yếu tố truyền thống của cấu trúc máy tính. Ngoài ra, hơn một thập kỷ qua, các nghiên cứu lớn đã tập trung vào thiết kế bộ vi xử lý nhúng song song, tối ưu hóa trình biên dịch linh hoạt và kỹ thuật phân tích để khai thác các cơ hội đó. Những nỗ lực này đã được công nhận cùng với sự phát triển của các máy tính đa năng được xây dựng dựa trên những công nghệ này. Tuy nhiên, những yêu cầu về khả năng hoạt động của hệ thống nhúng đã tạo ra những cơ hội mới để thúc đẩy việc ứng dụng những công nghệ này trong những miền ứng dụng đặc biệt. Những nghiên cứu về ILP (Instruction-Level Parallelism) đã phát triển một phương pháp luận mới gọi là EPIC (Explicitly Parallel Instruction Computing). Nhìn chung, những kỹ thuật này thể hiện những thay đổi căn bản trong cấu trúc máy tính.

42.3 Các bộ phận chính của vi xử lý

Phần cứng chính của bộ vi xử lý được chia thành các phần theo chức năng của chúng. Cách phổ biến nhất là chia một hệ thống thành 4 hệ thống con: bộ xử lý trung tâm, bộ nhớ hệ thống, hệ thống đầu vào/ra (I/O) và liên kết hệ thống. Hình 42.1 là sự kết nối giữa các hệ thống con này. Các thành phần chính và đặc tính của những hệ thống con này sẽ được trình bày trong các phần sau.



HÌNH 42.1 Cấu trúc các hệ thống con của một hệ thống máy tính

Bộ xử lý trung tâm

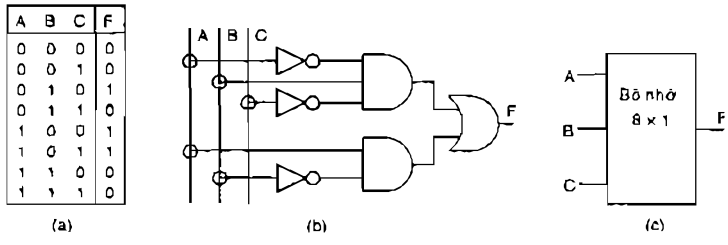
Hệ thống xử lý trung tâm của một bộ vi xử lý hiện đại thường được chia nhỏ hơn nữa thành: phần điều khiển (control), đường dẫn dữ liệu (data path), đường ống dẫn (pipelining) và phần cứng dự đoán lệnh nhánh (branch prediction).

Đơn vị điều khiển

Đơn vị điều khiển của một bộ vi xử lý tạo ra các tín hiệu để điều khiển các hoạt động trong đường dữ liệu. Có hai loại đường kết nối chính giữa đơn vị điều khiển và đường dữ liệu: các *đường điều khiển* và các *đường trạng thái*. Các đường điều khiển truyền tín hiệu điều khiển từ đơn vị điều khiển tới đường dữ liệu. Các giá trị tín hiệu khác nhau trên những đường này tạo ra các hoạt động khác nhau trong đường dữ liệu. Đường trạng thái mang các trạng thái thực hiện từ đường dữ liệu tới đơn vị điều khiển. Những đường này kiểm tra các trạng thái (bao gồm cả các thanh ghi) trong đường dữ liệu để tạo ra các quyết định điều khiển tương lai. Chú ý rằng các quyết định được thực hiện trong đơn vị điều khiển nhưng các thanh ghi là ở trong đường dữ liệu. Do đó, các trạng thái của giá trị thanh ghi được thiết lập trong đường dữ liệu và sau đó được đưa tới đơn vị điều khiển để tạo ra các quyết định. Một đơn vị điều khiển có thể được thực hiện bằng các mạch cứng, vi lập trình hoặc kết hợp cả hai phương pháp trên.

Trong thiết kế mạch cứng, mỗi đơn vị điều khiển được xem như một mạch tuần tự thông thường. Mục đích thiết kế là để tối thiểu hóa các thành phần và tối đa hóa tốc độ hoạt động. Khi đã thực hiện thì thiết kế chỉ có thể được thay đổi bằng cách đi lại dây. Do đó, các mạch được gọi là *đơn vị điều khiển mạch cứng*. Để việc thiết kế được tối ưu các mạch thường có cấu trúc nhỏ. Việc thiếu cấu trúc gây ra nhiều khó khăn khi thiết kế và gỡ lỗi các đơn vị điều khiển phức tạp với công nghệ này. Do đó, mạch cứng thường được sử dụng khi đơn vị điều khiển tương đối đơn giản.

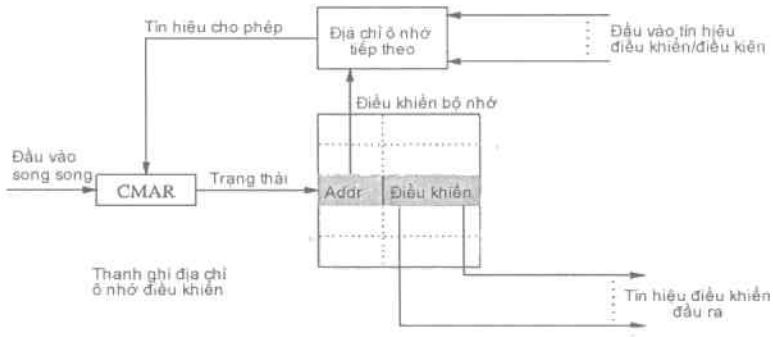
Hầu hết những khó khăn trong việc thiết kế các đơn vị điều khiển mạch cứng là các nỗ lực để tối ưu hóa các mạch tổ hợp. Nếu có một phương pháp mà không cần cố gắng tối ưu hóa mạch tổ hợp thì sự phức tạp của quá trình thiết kế có thể giảm đi đáng kể. Để thực hiện mạch tổ hợp có thể sử dụng bộ nhớ chỉ đọc (ROM) hoặc bộ nhớ truy cập ngẫu nhiên (RAM). Một đơn vị điều khiển có mạch tổ hợp được đơn giản bằng cách sử dụng ROM hoặc RAM được gọi là *đơn vị điều khiển được vi lập trình*. Bộ nhớ được sử dụng khi đó gọi là *bộ nhớ điều khiển*. Việc thực hiện mạch tổ hợp trong một đơn vị điều khiển với ROM/RAM được gọi là *vi lập trình*. Khái niệm vi lập trình lần đầu tiên được giới thiệu bởi Wilkes.



HÌNH 42.2 Sử dụng bộ nhớ để đơn giản hóa thiết kế lôgic: (a) lược đồ Karnaugh, (b) lôgic, (c) bộ nhớ

Ý tưởng sử dụng một bộ nhớ để thực hiện một mạch tổ hợp được mô tả bằng một ví dụ đơn giản. Giả sử rằng chúng ta thực hiện một hàm lôgic với ba biến đầu vào như được mô tả trong hình 42.2(a). Một cách phổ biến để thực hiện hàm này là sử dụng lược đồ Karnaugh để có sự tối ưu về lôgic và cách ghép nối. Kết quả này được chỉ ra trong hình 42.2(b). Một hàm tương tự cũng có thể được thực hiện trong bộ nhớ. Trong phương pháp này một bộ nhớ với tám vị trí 1 bit được dùng để nhớ tám tổ hợp có thể có của ba biến đầu vào. Vị trí *i* chứa giá trị F tương ứng với sự tổ hợp đầu vào thứ *i*. Ví dụ vị trí 3 chứa giá trị F (0) đối với sự liên kết đầu vào 011. Ba biến đầu vào được nối với các đầu vào địa chỉ của bộ nhớ để hoàn thành việc thiết kế này (hình 42.2(c)).

Xét lôgic giải mã và ô nhớ trong một bộ nhớ 8x1, rõ ràng là phương pháp sử dụng bộ nhớ cần nhiều thiết bị phần cứng hơn so với lược đồ Karnaugh nhưng việc thiết kế lại đơn giản hơn nhiều.



HÌNH 42.3 Mô hình cơ bản của đơn vị điều khiển được vi lập trình

Hình 42.3 mô tả mô hình tổng quát của một đơn vị điều khiển được vi lập trình. Mỗi vị trí bộ nhớ điều khiển có một trường địa chỉ và vài trường điều khiển. Trường địa chỉ kết hợp với địa chỉ lôgic kế tiếp thực hiện mạch lôgic tổ hợp để tạo ra giá trị trạng thái kế tiếp. Các trường điều khiển thực hiện mạch tổ hợp để tạo ra các tín hiệu điều khiển. Bộ nhớ điều khiển và địa chỉ lôgic kế tiếp sẽ được nghiên cứu một cách chi tiết trong phần này. Thanh ghi-bộ đếm trạng thái được đổi tên thành *thanh ghi địa chỉ bộ nhớ điều khiển* (CMAR-Control Memory Address Register) bởi lý do nội dung của thanh ghi được sử dụng như là địa chỉ đầu vào cho bộ nhớ điều khiển. Một điều quan trọng là CMAR chứa trạng thái của đơn vị điều khiển.

Đường dữ liệu

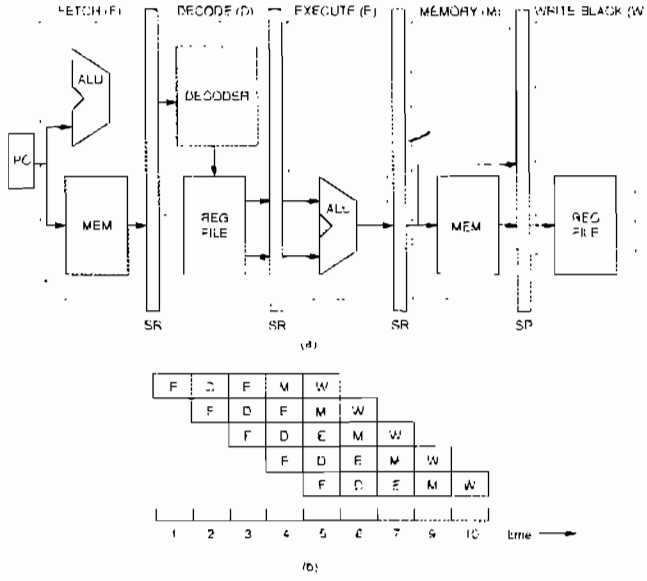
Đường dữ liệu của bộ vi xử lý chứa đựng các đơn vị thực hiện lôgic và số học để thực hiện các lệnh. Thiết kế đường dữ liệu bao gồm việc phân tích các chức năng sẽ được thực hiện, sau đó xác định một tập các thanh ghi phần cứng để giữ trạng thái tính toán và thiết kế các bước tính toán để chuyển đổi nội dung các thanh ghi thành kết quả cuối cùng. Nói chung các chức năng sẽ được thực hiện được chia thành nhiều bước, mỗi bước có thể được thực hiện với một số lượng lôgic hợp lý trong một chu kỳ đồng hồ. Mỗi bước sẽ đưa nội dung của thanh ghi đến gần kết quả cuối cùng hơn. Đường dữ liệu phải được trang bị với một số lượng phần cứng đáng kể để cho phép thực hiện những bước tính toán này trong một chu kỳ đồng hồ. Đường dữ liệu của một bộ vi xử lý điển hình chứa các file thanh ghi nguyên và thập phân, mười hoặc nhiều hơn các đơn vị chức năng cho việc tính toán và truy cập bộ nhớ và các thanh ghi đường ống dẫn. Để hiểu rõ về đường dữ liệu của bộ vi xử lý ngày nay chúng ta phải hiểu khái niệm về đường ống dẫn.

Đường ống dẫn

Vào những năm 1970 chỉ có những siêu máy tính và máy chủ được thiết kế với đường ống dẫn. Ngày nay hầu hết các bộ vi xử lý thương mại đều được thiết kế như vậy. Thực tế, đường ống dẫn chính là lý do khiến các bộ vi xử lý ngày nay hoạt động tốt hơn so với các siêu máy tính được xây dựng cách đây gần chục năm. Đường ống dẫn là một công nghệ phối hợp xử lý các hoạt động song song [2]. Công nghệ này đã được sử dụng trong các dây chuyền lắp ráp của nhiều ngành công nghiệp cách đây hơn một thế kỷ. Ý tưởng xuất phát từ mong muốn có một đội ngũ công nhân làm việc chuyên môn hóa trong những phần khác nhau của công việc để tạo ra một sản phẩm cuối cùng. Một dây chuyền mang mỗi sản phẩm qua một hàng công nhân. Mỗi một công nhân sẽ làm một phần nhỏ công việc trên mỗi sản phẩm. Mỗi sản phẩm sẽ hoàn thiện sau khi nó được xử lý bởi toàn bộ các công nhân trong hệ thống.

Ưu điểm của đường ống dẫn là cho phép một công nhân có thể ngay lập tức bắt đầu một công việc trên một sản phẩm mới sau khi kết thúc công việc ở sản phẩm hiện tại. Một phương pháp tương tự cũng được ứng dụng để thực hiện quá trình xử lý trong vi điều khiển. Hình 42.2(a) chỉ ra một đường ống dẫn 5 tầng chia nhỏ các lệnh thực hiện thành Fetch(F), Decode (D), Execute (E), Memory (M), Write-back (W). Giữa các tầng là một tầng thanh ghi được sử

dùng để giữ các thông tin về câu lệnh phục vụ cho việc điều khiển các câu lệnh. Một nguyên tắc cơ bản của đường ống dẫn là công việc của mỗi tầng phải được thực hiện trong cùng một lượng thời gian. Nếu không hiệu quả sẽ giảm đi đáng kể vì một tầng sẽ trở thành nút cổ chai của toàn bộ đường ống. Tương tự, khoảng thời gian của tầng đường ống chậm nhất xác định tần số xung nhịp của bộ xử lý. Do những yêu cầu trên và các đặc tính về tốc độ bộ nhớ, một vài tầng trong số 5 tầng cơ bản trên có thể chia thành các tầng nhỏ hơn. Ví dụ tầng bộ nhớ được chia thành ba tầng nhỏ, cho phép sự truy nhập bộ nhớ được đưa vào đường ống và tốc độ xung nhịp bộ xử lý là một hàm của độ trễ truy nhập bộ nhớ.



HÌNH 42.4 Cấu trúc của một đường ống: (a) máy, (b) các câu lệnh xếp chồng

Thời gian để thực hiện N câu lệnh trong một đường ống có K tầng có thể xác định được. Giả sử cần một chu kỳ thời gian T để hoàn thành toàn bộ câu lệnh và độ trễ xử lý mỗi tầng bằng T/K. Với một lược đồ đường ống, câu lệnh đầu tiên hoàn thành đường ống sau thời gian T và sẽ có một câu lệnh mới ra khỏi đường ống sau thời gian T/K. Do đó, độ trễ của việc thực hiện N câu lệnh có và không có đường ống tương ứng là:

$$T*(N) \tag{42.1}$$

$$T+(T/k)*(N-1) \tag{42.2}$$

Có một độ trễ khởi tạo trong mô hình thực hiện đường ống trước khi mỗi tầng hoạt động. Độ trễ khởi tạo được gọi là *độ trễ khởi động đường ống* (P) và bằng tổng thời gian thực hiện của một câu lệnh. Tốc độ tương đối của một máy có đường ống so với một máy không có đường ống được tính như sau:

$$\frac{P*N}{P+(N-1)} \tag{42.3}$$

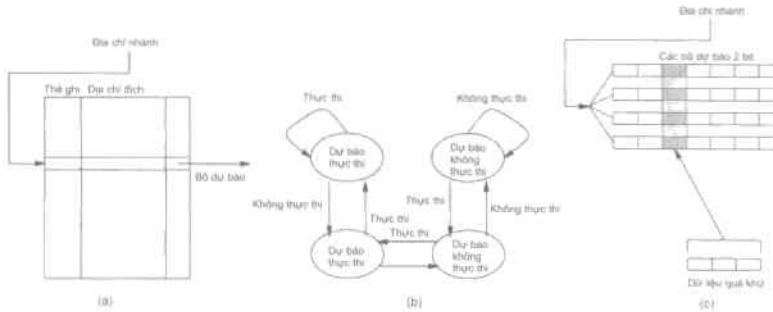
Khi N lớn hơn P rất nhiều, tốc độ lý tưởng sẽ tiến tới P. Kết quả này có thể dự đoán được vì có P bộ phận của máy đang làm việc song song cho phép việc thực hiện nhanh hơn P lần trong điều kiện lý tưởng.

Sự xếp chồng của các câu lệnh nối tiếp trong một bộ xử lý đường ống được minh họa trong hình 42.4(b). Đường ống câu lệnh sẽ đầy sau khi độ trễ đường ống $P = 5$ chu kỳ. Mặc dù cấu hình đường ống thực hiện việc xử lý ở mỗi tầng của bộ xử lý nhưng hai cơ cấu quan trọng được xây dựng để đảm bảo sự chính xác trong hoạt động chức năng của các câu lệnh độc lập trong trường

hợp có rủi ro về dữ liệu. Rủi ro dữ liệu xảy ra khi các câu lệnh trong đường ống tạo ra các kết quả cần thiết cho các câu lệnh sau nhưng các câu lệnh sau đó đã được bắt đầu trong đường ống. Trong cấu hình đường ống ở hình 42.4(a), các toán hạng thanh ghi được phục hồi trong tầng giải mã. Tuy nhiên tầng bộ nhớ và tầng thực hiện có thể xác định toán hạng thanh ghi và chứa giá trị hiện thời chính xác nhưng không có khả năng cập nhật file thanh ghi cho đến khi tầng thực hiện write-back sau đó. *Chuyển tiếp* là hành động nhận giá trị toán hạng chính xác cho câu lệnh thực hiện giữa truy nhập file thanh ghi khởi tạo và cập nhật file thanh ghi của bất kỳ câu lệnh kế tiếp nào. *Cài khóa* là hành động dừng hoạt động trong đường ống khi các điều kiện tạo ra các kết quả thanh ghi cần thiết bị trễ. Việc dừng các tầng trước của máy là cần thiết để các kết quả chính xác được sử dụng và máy không phải xử lý các kết quả sai. Nguyên nhân chủ yếu gây ra trễ hoạt động đường ống là do trễ của các câu lệnh và trễ bộ nhớ.

Dự đoán lệnh nhánh

Các lệnh nhánh gây ra nhiều vấn đề phức tạp cho các bộ xử lý đường ống do phần cứng phải tìm kiếm và thực hiện các câu lệnh cho đến khi các lệnh nhánh được hoàn thành. Việc thực hiện các lệnh không chính xác có thể gây ra sự suy giảm nghiêm trọng khả năng hoạt động do có các chu kì thừa trong dòng câu lệnh.



HÌNH 42.5 Dự báo lệnh rẽ nhánh

Có nhiều phương pháp để giải quyết các vấn đề dừng đường ống gây ra bởi các lệnh rẽ nhánh. Phương pháp đơn giản nhất quản lý các nhánh bằng cách coi các nhánh là được thực hiện hoặc không được thực hiện. Phương pháp này có thể được thiết lập cho mỗi nhánh hoặc được xác định bằng mã rẽ nhánh. Thiết kế này cho phép đường ống tiếp tục tìm các câu lệnh như thể các lệnh nhánh là các câu lệnh bình thường. Tuy nhiên, các câu lệnh nhận được có thể phải loại bỏ và việc tìm các câu lệnh cần được thực hiện lại khi kết quả nhánh là sai. *Nhánh trễ* là một phương pháp khác, phương pháp này coi tập các câu lệnh theo sau một nhánh như là các khe trễ. Các câu lệnh khe trễ có được thực hiện hay không là phụ thuộc vào lệnh nhánh có được thực hiện hay không. Các hạn chế của nhánh trễ là do trình biên dịch và đặc tính chương trình không thể hỗ trợ nhiều câu lệnh thực hiện độc lập với hướng nhánh. Rất nhiều cải tiến đã được thực hiện để đưa ra khái niệm *nhánh hủy bỏ*, bao gồm một hướng dự báo trước cho nhánh. Khi dự báo bị sai, các câu lệnh khe trễ sẽ không được thực hiện.

Một phương pháp hiện đại hơn để giảm các hạn chế của lệnh nhánh là sử dụng phần cứng để dự báo động kết quả của một nhánh. Các chiến lược dự báo nhánh cho phép phần cứng tiếp tục xử lý câu lệnh cùng với đường điều khiển được dự báo, do đó loại bỏ được các chu kỳ thừa. Sự hoạt động hiệu quả có thể được duy trì trong khi các mục tiêu nhánh được dự báo chính xác. Tuy nhiên, sai sót lớn sẽ xảy ra khi một nhánh bị dự báo nhầm. Bộ đệm mục tiêu nhánh là một cấu trúc lưu trữ được truy cập song song cùng với quá trình tìm các câu lệnh. Nó ghi lại quá khứ của các lệnh nhánh để thực hiện một dự báo mới khi các lệnh nhánh tiếp tục được thực hiện. Phương pháp dự báo này điều chỉnh dự báo nhánh theo các ứng xử của chương trình trong khi thực hiện, tạo ra sự dự báo chính xác. Địa chỉ đích của các nhánh cũng được nhớ vào trong bộ đệm để các câu lệnh đích có thể tìm được ngay tức khắc nếu một nhánh dự báo được thực hiện.

Nhiều phương pháp dự báo nhánh đích đã được xây dựng [3]. Hình 42.5 mô tả một vài phương pháp nhánh dự báo. Các phương pháp thông dụng nhất giữ lại các thông tin quá khứ cho mỗi nhánh như mô tả trong hình 42.5(a). Thông tin quá khứ bao gồm các hướng nhánh trước đó phục vụ cho việc dự báo các hướng nhánh tương lai. Thông tin quá khứ đơn giản nhất là lần thực hiện cuối cùng. Có một phương pháp dự báo nhánh hiệu quả hơn sử dụng bộ đếm quá khứ trạng thái bão hòa 2-bit để xác định các kết quả nhánh dự báo tương lai, phương pháp này được mô tả trong hình 42.5(b). Mỗi dự báo chính xác sẽ củng cố sự dự báo, trong khi dự báo sai sẽ làm suy giảm nó. Cần hai lần dự báo sai liên tiếp mới cho phép khôi phục lại hướng dự báo (hoặc là xảy ra hoặc là không).

Ngày nay, các lược đồ dự báo nhánh thích nghi hai mức phức tạp hơn đã được xây dựng. Chúng sử dụng hai mức quá khứ nhánh để tạo ra dự báo, như mô tả trong hình 42.5(c). Mức thứ nhất là kết quả quá khứ của các nhánh cuối cùng được thực hiện. Mức thứ hai là ứng xử của một mẫu cụ thể các quá khứ nhánh xảy ra lần cuối cùng. Có một vài phương pháp khác để xây dựng hai mức của lược đồ dự báo nhánh thích nghi, cấu trúc có thể chứa thông tin dựa trên các nhánh riêng lẻ, nhóm hoặc toàn cục. Thông tin riêng lẻ chứa quá khứ nhánh của mỗi lệnh nhánh. Thông tin nhóm tập hợp các nhánh theo địa chỉ câu lệnh của chúng, do đó hình thành các tập quá khứ nhánh. Thông tin toàn cục sử dụng tất cả các kết quả nhánh quá khứ. Mức thứ hai chứa các hành vi của nhánh cũng có thể được xây dựng bằng cách sử dụng một trong ba loại kể trên. Nói chung, mẫu quá khứ nhánh mức một được sử dụng như là bảng chú dẫn cho quá khứ nhánh mức hai.

Hệ thống bộ nhớ

Hệ thống bộ nhớ đóng vai trò như là bộ chứa thông tin trong hệ thống vi xử lý. Đơn vị xử lý nhận các thông tin được lưu trữ trong bộ nhớ, xử lý các thông tin và trả lại các thông tin mới về bộ nhớ. Hệ thống bộ nhớ được cấu tạo từ các đơn vị DRAM bán dẫn cơ bản được gọi là môđun hoặc các khối.

Các đặc tính của bộ nhớ như tốc độ, dung lượng và giá trị đóng vai trò quan trọng đến khả năng hoạt động của hệ thống. Tốc độ của một hệ thống bộ nhớ là tham số hoạt động chính trong thiết kế hệ thống vi xử lý. *Độ trễ* (L) của bộ nhớ được định nghĩa là thời gian trễ từ khi bộ xử lý lần đầu yêu cầu dữ liệu từ bộ nhớ cho đến khi bộ xử lý nhận được dữ liệu. *Băng thông* (BW) được định nghĩa là tốc độ mà thông tin có thể được truyền từ hệ thống bộ nhớ. Băng thông và độ trễ của bộ nhớ liên quan đến số lượng yêu cầu (R) mà bộ nhớ có thể cung cấp:

$$BW=\frac{L}{R} \tag{42.4}$$

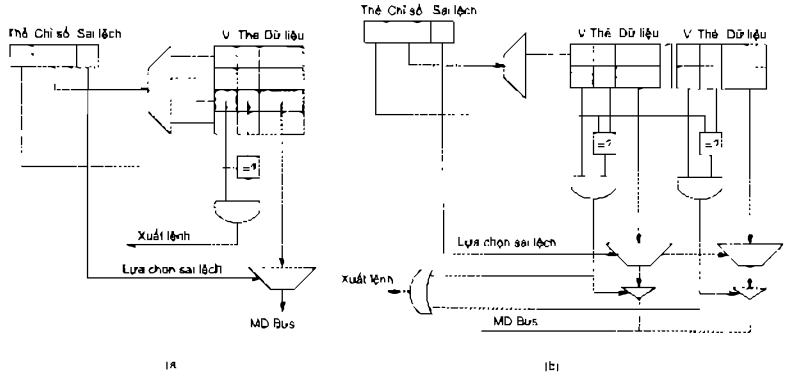
Băng thông đóng vai trò quan trọng đối với khả năng hoạt động của bộ xử lý. Tuy nhiên, tối ưu độ trễ và cải thiện băng thông thường mâu thuẫn với sự cần thiết của việc tăng dung lượng và giảm giá thành hệ thống.

Bộ nhớ Cache

Bộ nhớ Cache là một bộ nhớ nhỏ và nhanh, được cấu tạo từ các SRAM bán dẫn. Trong các hệ thống máy tính hiện đại luôn có sự phân cấp của bộ nhớ cache. Cache mức cao nhất là gần với bộ xử lý nhất và mức thấp nhất là gần với bộ nhớ chính. Mỗi một cache mức cao hơn nhanh hơn khoảng 5-10 lần so với mức tiếp theo. Mục đích của phân cấp cache là để thoả mãn việc truy cập bộ nhớ trong một hoặc một vài chu kỳ đồng hồ. Cache mức cao nhất thường được chia thành cache câu lệnh và cache dữ liệu để cho phép bộ xử lý truy nhập đồng thời các câu lệnh và dữ liệu. Các bộ nhớ cache lần đầu tiên được sử dụng trong các máy chủ IBM vào những năm 1960. Kể từ năm 1985, các bộ nhớ cache đã trở thành một đặc trưng tiêu chuẩn cho các bộ vi xử lý.

Các bộ nhớ cache sử dụng nguyên tắc tham chiếu cục bộ. Theo nguyên tắc này thì một số vị trí bộ nhớ được tham chiếu thường xuyên hơn so với những vị trí khác, dựa trên hai tính chất. *Vị trí không gian* là tính chất mà sự truy cập tới một vị trí bộ nhớ làm tăng khả năng vị trí bộ nhớ bên cạnh cũng sẽ bị truy cập. Vị trí không gian chủ yếu dựa trên truy cập tuần tự tới mã chương trình

và dữ liệu cấu trúc. *Vị trí thời gian* là tính chất mà sự truy nhập tới một vị trí bộ nhớ làm tăng khả năng vị trí đó sẽ được truy nhập lại trong tương lai. Hai tính chất này cùng nhau đảm bảo rằng hầu hết các tham chiếu bộ nhớ sẽ được thoả mãn bởi bộ nhớ cache.



HÌNH 42.6 Bộ nhớ cache: (a) ánh xạ trực tiếp, (b) liên hợp vùng hai đường

Có nhiều thiết kế bộ nhớ cache khác nhau: ánh xạ trực tiếp, liên hợp toàn bộ và liên hợp vùng. Hình 42.6 mô tả hai lược đồ cơ bản của bộ nhớ cache, ánh xạ trực tiếp và liên hợp vùng.

Cache ánh xạ trực tiếp, được mô tả trong hình 42.6(a), cho phép mỗi khối bộ nhớ có một nơi để lưu ở trong cache. Cache liên hợp toàn bộ, trong hình 42.6(b), cho phép một khối được đặt ở bất kỳ đâu trong cache. Cache liên hợp vùng giới hạn một khối ở một tập vị trí nhất định trong cache.

Lỗi cache là tình huống xảy ra khi dữ liệu được yêu cầu không có trong bất cứ vị trí nào của cache. Lỗi cache thường được phân làm ba loại: xung đột, ép buộc và dung lượng. Lỗi xung đột là các lỗi không xảy ra trong cache liên hợp toàn bộ với sự thay thế LRU (Least Recent Used). Lỗi ép buộc là lỗi xảy ra khi tham chiếu lần đầu đến một vị trí bộ nhớ. Lỗi dung lượng xảy ra khi kích thước cache không đủ lớn để chứa dữ liệu giữa các tham chiếu. Định nghĩa đầy đủ về lỗi cache có thể tìm thấy trong [4].

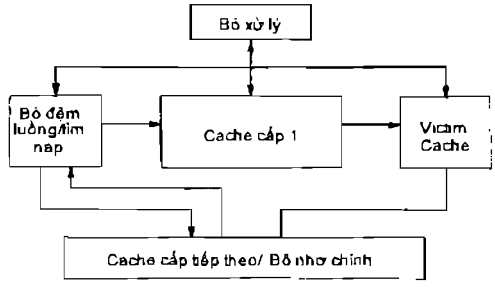
Không giống như tính chất của hệ thống bộ nhớ, trễ trong bộ nhớ cache là không cố định và phụ thuộc vào độ trễ và tần số của lỗi cache. Một đại lượng dùng để đánh giá lỗi cache là *độ trễ hiệu quả*. Trễ hiệu quả phụ thuộc vào hai loại trễ: L_{HIT} - trễ gây ra do sự truy cập dữ liệu bên trong cache và L_{MISS} - trễ do sự truy nhập dữ liệu không ở trong cache. Trễ hiệu quả cũng phụ thuộc vào H - tỷ lệ phần trăm truy cập bộ nhớ có trong cache và M (hoặc 1-H) - tỷ lệ phần trăm truy cập bộ nhớ không có trong cache. Trễ hiệu quả trong một hệ thống cache được tính như sau:

$$L_{effective} = L_{HIT} * H + L_{MISS} * (1 - H) \tag{42.5}$$

Cùng với thiết kế cache cơ bản và các vấn đề về kích thước, một vài tham số khác cũng ảnh hưởng đến toàn bộ hoạt động của cache và M trong một hệ thống. Phương pháp cập nhật bộ nhớ chính cho biết khi nào bộ nhớ chính sẽ được cập nhật bởi các hoạt động lưu trữ. Trong *write-through cache*, mỗi lần ghi tác động ngay lập tức đến bộ nhớ chính. Trong *write-back cache*, các lần ghi tác động tới bộ nhớ chính chỉ khi khối cache tương ứng bị thay thế. Sự định vị các khối cache là một tham số khác và xác định khi nào khối cache là ghi hay đọc. Cuối cùng, các giải thuật thay thế khối cache cho các cấu trúc liên hợp có thể được thiết kế theo nhiều cách khác nhau để tăng thêm hiệu suất của cache. Chúng bao gồm LRU, LFU (Least Frequently Used), ngẫu nhiên và FIFO (First In, First Out). Chiến thuật quản lý cache cố gắng khai thác các tính chất về vị trí. Vị trí không gian được khai thác bằng cách quyết định khối bộ nhớ nào được đặt vào trong cache, vị trí thời gian được khai thác bằng cách quyết định khối cache nào bị thay thế. Theo nguyên tắc, khi dịch vụ cache bị lỗi, chúng sẽ khóa tất cả các yêu cầu mới. Tuy nhiên các cache không bị khóa có

thể được thiết kế để phục vụ trường hợp các yêu cầu lỗi đồng thời xảy ra, do đó làm giảm độ trễ truy cập dữ liệu bộ nhớ.

Các bộ nhớ đệm có thể được sử dụng để cải thiện khả năng hoạt động của cache. Hai bộ đệm như thế là *streaming/prefetch buffer* và *victim cache* [2]. Hình 42.7 mô tả mối quan hệ giữa streaming buffer và victim cache với cache chính của hệ thống bộ nhớ.



HÌNH 42.7 Hệ thống bộ nhớ cache hiện đại

Streaming buffer được sử dụng như là một cơ chế nhận trước các lỗi cache. Khi một lỗi cache xảy ra, streaming buffer bắt đầu nhận các dòng liên tiếp bắt đầu từ điểm lỗi. Victim cache thường là một cache nhỏ, liên kết hoàn toàn, được tải chỉ với các dòng cache bị xoá khỏi cache chính. Khi có một lỗi trong cache chính, victim cache có thể lưu trữ các dữ liệu bổ sung. Việc sử dụng victim cache có thể cải thiện khả năng hoạt động vì đã làm giảm số lượng các lỗi xung đột. Hình 42.7 mô tả cách truy nhập cache được xử lý qua streaming buffer vào trong cache chính trên các yêu cầu cache và từ cache chính qua victim cache tới mức thứ hai của bộ nhớ trên lỗi cache.

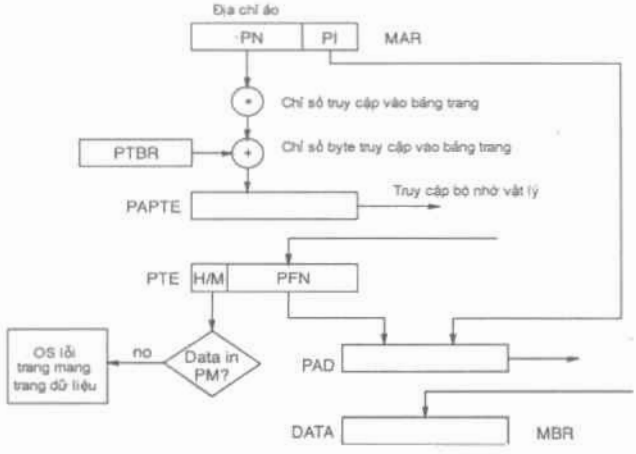
Tóm lại bộ nhớ cache được cấu tạo để giữ hầu hết các phần quan trọng của bộ nhớ. Các kỹ thuật phần cứng hoặc phần mềm có thể được sử dụng để lựa chọn phần nào của bộ nhớ chính được lưu vào trong cache. Tuy nhiên khả năng hoạt động của cache bị ảnh hưởng mạnh bởi cách thức lập trình và các thiết kế phần cứng khác nhau.

Bộ nhớ ảo

Bộ nhớ cache đã chứng minh nguyên tắc địa chỉ bộ nhớ dữ liệu có thể được tách riêng khỏi vị trí lưu trữ cụ thể. Bộ nhớ chính và đĩa lưu trữ phân cấp hai mức cũng sử dụng cách đánh địa chỉ tương tự. Địa chỉ được tạo ra bởi một chương trình gọi là *địa chỉ ảo*. Địa chỉ này cần được chuyển thành *địa chỉ vật lý* hoặc vị trí trong bộ nhớ chính. Hệ thống quản lý bộ nhớ ảo là một cơ cấu cung cấp cho người lập trình những phương pháp đồng nhất đơn giản để truy nhập vào bộ nhớ chính và bộ nhớ phụ. Với sự quản lý bộ nhớ ảo, người lập trình được cung cấp không ảo gian để lưu giữ tất cả câu lệnh và dữ liệu. Không gian ảo được tổ chức như là một dãy tuyến tính các vị trí. Mỗi vị trí có một địa chỉ để truy cập thuận tiện. Các câu lệnh và dữ liệu phải được lưu trữ ở một nơi nào đó trong hệ thống thật. Những vị trí không gian ảo này phải tương ứng với những vị trí vật lý trong bộ nhớ chính và bộ nhớ phụ. Hệ thống quản lý bộ nhớ ảo ánh xạ các vị trí không gian ảo tới các vị trí trong bộ nhớ chính và bộ nhớ phụ. Quá trình ánh xạ trên được theo dõi bởi hệ thống quản lý bộ nhớ ảo. Những người lập trình không phải quan tâm tới vấn đề trên.

Hầu hết các hệ thống quản lý bộ nhớ thông dụng ngày nay là các hệ thống đánh số trang, trong đó mỗi không gian ảo được chia thành các trang được đánh số trang (PN). Mỗi trang chứa các vị trí liên tiếp trong không gian ảo được đánh số bởi chỉ số trang (PI). Số lượng vị trí trong mỗi trang là một thông số thiết kế quan trọng được gọi là kích thước trang. Kích thước trang thường được định nghĩa là số mũ của 2 để không gian ảo có thể được chia thành một số nguyên các trang. Các trang là đơn vị cơ bản của quản lý bộ nhớ ảo. Nếu bất kỳ vị trí nào trong một trang được ánh xạ tới bộ nhớ chính, các vị trí khác trong trang đó cũng được ánh xạ tới bộ nhớ chính. Điều này đã làm giảm kích thước thông tin ánh xạ.

Thành phần của bộ nhớ phụ dùng để chứa các trang của không gian ảo được gọi là *không gian trao đổi*. Bộ nhớ chính và không gian trao đổi được chia thành các khung trang. Mỗi khung trang có thể lưu trữ một trang của bộ nhớ ảo. Nếu một trang được ánh xạ vào trong bộ nhớ chính, nó cũng được lưu giữ bởi khung trang trong bộ nhớ chính. Việc ghi ánh xạ trong quản lý bộ nhớ ảo cho phép giữ lại các đường liên kết giữa trang và khung trang.



HÌNH 42.8 Chuyển đổi bộ nhớ ảo

Khi một vị trí không gian ảo được yêu cầu, hệ thống quản lý bộ nhớ ảo sẽ tìm kiếm bản ghi ánh xạ. Nếu bản ghi ánh xạ chỉ ra rằng trang lưu giữ vị trí không gian ảo được yêu cầu là trong bộ nhớ chính, hệ thống quản lý sẽ thực hiện việc truy nhập mà không có bất cứ rắc rối nào. Ngược lại, việc truy cập bộ nhớ phụ sẽ được thực hiện. Truy nhập bộ nhớ phụ luôn là một nhiệm vụ phức tạp và được thực hiện như là một dịch vụ hệ điều hành. Để truy cập thông tin lưu trong bộ nhớ phụ, một dịch vụ hệ điều hành luôn phải được yêu cầu để chuyển các thông tin vào trong bộ nhớ chính. Điều này cũng áp dụng cho việc quản lý bộ nhớ ảo. Khi một trang được ánh xạ vào bộ nhớ phụ, hệ thống quản lý bộ nhớ ảo phải yêu cầu một dịch vụ trong hệ điều hành để chuyển vị trí không gian ảo được yêu cầu vào trong bộ nhớ chính, cập nhật bản ghi ánh xạ của nó và sau đó thực hiện truy cập.

Quá trình xử lý chính của quản lý bộ nhớ ảo là một giải thuật truy cập bộ nhớ. Giải thuật chuyển đổi địa chỉ ảo một mức được mô tả trong hình 42.8. Tại lúc bắt đầu chuyển đổi, giải thuật truy cập bộ nhớ nhận một địa chỉ ảo trong thanh ghi địa chỉ bộ nhớ (MAR), tìm kiếm bản ghi ánh xạ, yêu cầu một dịch vụ hệ điều hành để chuyển các trang được yêu cầu nếu cần thiết và thực hiện truy cập bộ nhớ chính.

Quá trình ánh xạ được ghi vào một cấu trúc dữ liệu được gọi là bảng trang, bảng này được định vị trong bộ nhớ chính tại một vị trí được đánh dấu bởi thanh ghi cơ sở bảng trang (PTBR).

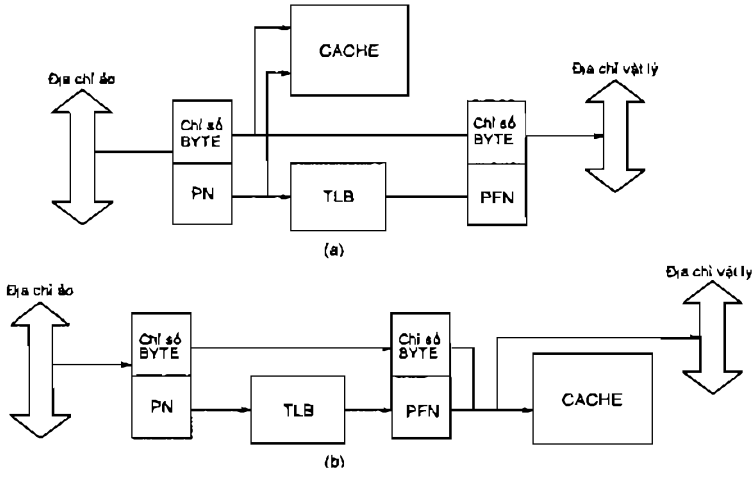
Chỉ số bảng trang và PTBR tạo thành địa chỉ vật lý (PAPTE) của các mục bảng trang tương ứng (PTE). Mỗi PTE giữ dấu ánh xạ của một trang trong không gian ảo. Nó bao gồm 2 trường: 1-bit hit/miss (H/M) và số khung trang. Nếu bit H/M được tích cực (hit) thì trang tương ứng nằm trong bộ nhớ chính. Trong trường hợp này khung trang chứa trang yêu cầu được trỏ tới bởi số khung trang (PFN). Địa chỉ vật lý cuối cùng (PAD) của dữ liệu yêu cầu được hình thành bằng cách sử dụng PFN và PI. Dữ liệu được trả về và chứa trong thanh ghi bộ nhớ đệm (MBR) và bộ xử lý được thông báo rằng việc truy cập bộ nhớ đã thành công. Ngược lại (miss), việc truy cập bộ nhớ phụ sẽ được thực hiện. Trong trường hợp này, số khung trang nên bị bỏ qua. Bộ xử lý lỗi phải được xử dụng để truy cập tới bộ nhớ phụ. Phần cứng thực hiện giải thuật chuyển đổi địa chỉ được gọi là đơn vị quản lý bộ nhớ (MMU).

Độ phức tạp của giải thuật phụ thuộc vào cấu trúc ánh xạ. Một cấu trúc ánh xạ đơn giản sẽ được sử dụng trong phần này để giới thiệu các nguyên tắc cơ bản của các giải thuật truy cập bộ nhớ. Tuy nhiên, các phương pháp hai mức phức tạp hơn thường được sử dụng do kích thước của không gian địa chỉ ảo. Kích thước của bảng trang được chọn có thể rất lớn cho kích thước bộ nhớ chính. Do đó cần phải ánh xạ các phần của bảng trang vào trong một bảng trang thứ hai. Trong một thiết kế như vậy, chỉ có bảng trang mức hai được lưu trữ trong bộ nhớ chính, trong khi bảng trang đầu tiên được ánh xạ giống như dữ liệu trong không gian ảo. Cũng có các yêu cầu cho các thiết kế như thế trong một hệ thống đa lập trình, ở đó có nhiều quá trình xử lý cùng được thực hiện tại một thời điểm. Mỗi bộ điều khiển có không gian ảo riêng và do đó có bảng trang của chính nó. Vì vậy những hệ thống này cần giữ nhiều bảng trang cùng một lúc. Thông thường cần nhiều bộ nhớ chính để lưu tất cả các bảng trang tích cực. Điều này dẫn đến một giải pháp là cung cấp các mức khác của ánh xạ.

TLB (Translation Lookaside Buffer)

Phần cứng hỗ trợ hệ thống bộ nhớ ảo thường có một cơ chế chuyển đổi địa chỉ ảo thành địa chỉ vật lý được sử dụng để truy nhập bộ nhớ chính. TLB là một cấu trúc lưu trữ chứa các PTE được sử dụng thường xuyên cho việc chuyển đổi địa chỉ. Với TLB, chuyển đổi địa chỉ có thể được thực hiện trong một chu kỳ đồng hồ khi TLB có chứa các PTE yêu cầu (TLB hit). Giải thuật chuyển đổi địa chỉ chỉ được thực hiện khi các PTE yêu cầu không có trong TLB (TLB miss).

Độ phức tạp tăng lên khi hệ thống bao gồm cả quản lý bộ nhớ ảo và bộ nhớ cache. Vấn đề chính là liệu chuyển đổi địa chỉ có được thực hiện trước truy nhập bộ nhớ cache hay không. Trong hệ thống *cache ảo*, địa chỉ ảo truy cập trực tiếp vào cache. Trong một hệ thống *cache vật lý*, địa chỉ ảo được chuyển đổi thành địa chỉ vật lý trước khi truy nhập cache. Hình 42.9 mô tả chuyển đổi cache vật lý và cache ảo.



HÌNH 42.9 TLB, (a) cache ảo (b) cache vật lý

Một hệ thống cache ảo thường truy cập đồng thời bộ nhớ cache và TLB. Truy cập đồng thời khả thi khi kích thước của trang bộ nhớ ảo lớn hơn dung lượng cache (được chia theo mức độ liên kết cache). Thực chất, do chỉ số trang ảo là giống chỉ số địa chỉ vật lý, vì vậy không cần thiết phải chuyển đổi các địa chỉ ảo có chỉ số thấp hơn. Do đó cache có thể truy cập song song với TLB. Đặc biệt khi không có lôgic TLB giữa bộ điều khiển và cache, truy nhập cache có thể được thực hiện với chi phí thấp hơn trong các hệ thống cache ảo. Khi đó các hệ thống cache đa truy nhập trên mỗi chu kỳ sẽ thay thế các TLB đa cổng. Tuy nhiên, chuyển đổi cache ảo lại có vấn đề với bộ nhớ ảo. Đó là trường hợp hai quá trình truy cập đến hai ô nhớ vật lý khác nhau có cùng một địa chỉ ảo. Giải pháp cho vấn đề này là gắn công cụ nhận dạng quá trình xử lý vào địa

chỉ ảo. Một vấn đề khác là các địa chỉ ảo khác nhau của cùng quá trình xử lý có thể được ánh xạ tới cùng một địa chỉ vật lý. Nói chung không có cách giải quyết dễ dàng và nó liên quan đến vấn đề chuyển đổi ngược.

Các thiết kế cache vật lý không phải lúc nào cũng bị giới hạn bởi độ trễ của TLB và truy cập cache. Nói chung có hai giải pháp cho phép thiết kế cache vật lý lớn. Giải pháp thứ nhất tăng sự liên kết của một tập các cache, nó cho phép phần chỉ số cache của địa chỉ được sử dụng ngay lập tức bởi cache song song với quá trình chuyển đổi địa chỉ ảo. Tuy nhiên chi phí để thực hiện một tập liên kết lớn là rất cao. Giải pháp thứ hai là để sử dụng kích thước trang lớn hơn. Cache có thể được truy cập song song với truy cập TLB. Trong giải pháp này, các chỉ số địa chỉ được chuyển đổi qua TLB giảm đi đáng kể, do đó trễ cũng nhỏ hơn. Với kích thước trang lớn hơn, các cache ảo không có ưu điểm so với cache vật lý về thời gian truy cập.

Hệ thống vào/ra

Hệ thống vào/ra (I/O) truyền dữ liệu giữa các thành phần bên trong (CPU và bộ nhớ chính) và các thiết bị ngoại vi (đĩa cứng, các thiết bị đầu cuối, máy in, bàn phím và máy quét).

Bộ điều khiển ngoại vi

CPU thường điều khiển hệ thống I/O bằng cách đọc và viết các thanh ghi. Có hai phương pháp thông dụng cho phép CPU truy nhập vào các thanh ghi I/O là các tập lệnh I/O và I/O được ánh xạ bộ nhớ. Trong phương pháp tập lệnh I/O, các câu lệnh đặc biệt được thêm vào tập lệnh để truy nhập cờ trạng thái I/O, các thanh ghi điều khiển và các thanh ghi bộ đệm dữ liệu. Trong phương pháp I/O ánh xạ bộ nhớ, các thanh ghi điều khiển, các cờ trạng thái và các thanh ghi bộ đệm dữ liệu được ánh xạ như vị trí bộ nhớ vật lý. Do diện tích chip và số lượng chân đã tăng lên đáng kể nên vi xử lý thường có các bộ điều khiển ngoại vi trên chip. Khuynh hướng này thể hiện rất rõ trong bộ vi xử lý nhúng.

Bộ điều khiển truy cập bộ nhớ trực tiếp

Bộ điều khiển truy cập bộ nhớ trực tiếp (DMA) là một bộ điều khiển ngoại vi có thể điều khiển trực tiếp địa chỉ của bus hệ thống. Dữ liệu được chuyển trực tiếp từ bộ đệm dữ liệu đến bộ nhớ chính chứ không phải từ bộ đệm dữ liệu tới một thanh ghi CPU, sau đó từ thanh ghi CPU tới bộ nhớ chính.

Kết nối hệ thống

Kết nối hệ thống cho phép các thành phần trong một hệ thống máy tính giao tiếp với nhau.

Đường kết nối chuyên dụng hay kết nối điểm-điểm cho phép truyền thông chuyên dụng giữa các thành phần. Có nhiều cấu hình kết nối hệ thống khác nhau dựa trên sự kết nối của các thành phần hệ thống. Một cấu hình kết nối hoàn chỉnh, yêu cầu $N(N\text{-}1)/2$ đường liên kết, được tạo khi có một đường liên kết giữa tất cả các cặp thành phần. Cấu hình *hypercube* sử dụng một *n-tuple* {1, 0} làm tọa độ cho mỗi thành phần và xây dựng một đường liên kết giữa các thành phần có tọa độ chỉ khác nhau trong một chiều, do đó yêu cầu $N\log N$ đường liên kết. Kết nối *mesh* bố trí các thành phần hệ thống trong mảng kích thước N chiều, các thành phần liền kề được kết nối với nhau, do đó cần $2N$ đường liên kết.

Mạng chuyển mạch là một nhóm các khóa xác định sự tồn tại của các đường kết nối giữa các thành phần. Mạng *cross-bar*, là mạng chuyển mạch phổ biến nhất, sử dụng một mảng hai chiều gồm $N+M$ các khóa để cung cấp một kết nối bất kỳ giữa N thành phần ở một phía và M thành phần ở phía bên kia. Mạng này sử dụng NM khóa và $N+M$ đường kết nối. Một mạng chuyển mạch khác là mạng nhiều tầng, sử dụng *mạng xáo trộn* nhiều tầng để cung cấp các kết nối giữa N thành phần ở mỗi phía bằng cách sử dụng $N\log N$ khóa và N log N liên kết.

Bus chia sẻ là đường kết nối tất cả các thành phần với nhau, đây là cấu trúc kết nối thông dụng nhất. Chia sẻ bus giữa các thành phần của một hệ thống đòi hỏi một số yêu cầu đối với

điều khiển bus. Trước hết, có một sự phân biệt giữa các bus chủ (master), các đơn vị điều khiển chuyển bus (CPU, DMA, IOP) và các bus phụ (slave), và các đơn vị khác (bộ nhớ, các cổng I/O lập trình được).

Giao diện bus và địa chỉ bus là các phương tiện để kết nối và huỷ các kết nối các đơn vị trên bus. Phân xử bus là quá trình cấp tài nguyên bus cho một trong những yêu cầu. Phân xử bus thường sử dụng một lược đồ lựa chọn tương tự như ngắt, tuy nhiên có nhiều phương pháp thiết lập lựa chọn cố định. Phân xử ưu tiên cố định cấp cho mọi yêu cầu một độ ưu tiên như nhau và đảm bảo mỗi yêu cầu được ưu tiên tại một thời điểm. Định thời bus là phương pháp kết nối giữa các đơn vị hệ thống và có thể được phân loại là đồng bộ hoặc không đồng bộ. Định thời bus đồng bộ sử dụng một đồng hồ chia sẻ để định nghĩa thời gian cho các bus khác thay đổi và ổn định tín hiệu. Đồng hồ chia sẻ cho phép bus được quan sát tại những khoảng thời gian mong muốn và các hành động được thực hiện theo yêu cầu. Tuy nhiên, bus hệ thống đồng bộ phải hoạt động tại tốc độ của thành phần có tốc độ thấp nhất. Định thời bus không đồng bộ cho phép các đơn vị sử dụng các đồng hồ khác nhau, nhưng do không có đồng hồ chia se nên phải sử dụng các tín hiệu khác để xác định sự phù hợp của các tín hiệu bus.

42.4 Cấu trúc tập lệnh

Có nhiều phần tử đặc trưng cho một cấu trúc tập lệnh như kích thước từ (word size), mã hóa câu lệnh và mô hình cấu trúc.

Kích thước từ

Các chương trình thường khác nhau về kích thước dữ liệu. Các chương trình xử lý từ hoạt động trên các dữ liệu 8-bit hoặc 16-bit tương ứng với các ký tự trong văn bản. Rất nhiều ứng dụng yêu cầu dữ liệu nguyên 32-bit để tránh hiện tượng tràn trong khi tính toán giải thuật. Các ứng dụng khoa học thường yêu cầu dữ liệu 64-bit thập phân để có được kết quả chính xác mong muốn. Do đó các bộ điều khiển thường được thiết kế để truy cập các dữ liệu nhiều byte từ hệ thống bộ nhớ. Đây là một vấn đề rất phức tạp trong thiết kế bộ vi xử lý.

Một quy ước chỉ ra số lượng các byte của một từ bộ nhớ. Trong quy ước đó byte có trọng số thấp nhất trong một từ được đánh số là byte 0. Chỉ số này tăng lên khi các vị trí tăng lên tương ứng theo trọng số. Các cấu trúc DEC VAX và X86 tuân theo các quy ước này. Trong các quy ước lớn, byte trọng số lớn nhất trong một từ được đánh số là 0. Chỉ số này giảm xuống khi vị trí giảm theo trọng số. Các cấu trúc IBM 360/3670 HP PA-RISC, Sun SPARC và Motorola 680X0 tuân theo quy ước lớn này. Người sử dụng sẽ gặp nhiều khó khăn khi cố gắng truyền byte nhị phân giữa các máy sử dụng các quy ước khác nhau.

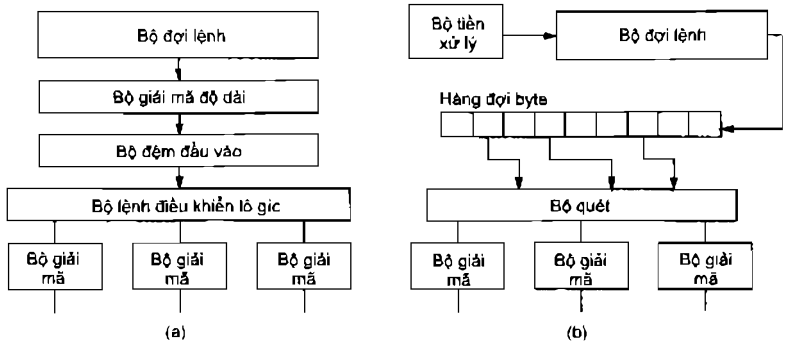
Mã hóa tập lệnh

Mã hóa lệnh đóng một vai trò rất quan trọng trong mật độ mã và khả năng hoạt động của bộ vi xử lý. Thông thường, giá thành dung lượng bộ nhớ là một nhân tố được xem xét trong việc thiết kế tập lệnh có kích thước cố định hoặc kích thước thay đổi. Mã hóa kích thước cố định tạo ra kích thước mã hóa như nhau đối với mọi câu lệnh. Mã hóa kích thước cố định là một đặc tính chung của các bộ vi xử lý hiện đại và là sản phẩm của sự nâng cấp dung lượng bộ nhớ.

Tập lệnh kích thước thay đổi là một thuật ngữ mô tả kiểu mã hóa câu lệnh sử dụng chiều dài câu lệnh khác nhau tuỳ theo phương pháp định địa chỉ các toán tử. Các phương pháp định địa chỉ thông thường là thanh ghi hoặc bộ nhớ chỉ số. Hình 4.20 mô tả hai thiết kế thường ứng dụng trong giải mã các câu lệnh kích thước thay đổi. Thiết kế đầu tiên, hình 42.10(a), có thêm một tầng giải mã câu lệnh so với thiết kế đường ống ban đầu. Trong mô hình này, tầng thứ nhất được sử dụng để xác định chiều dài câu lệnh và điều khiển các câu lệnh tới tầng thứ hai, ở đó việc giải mã câu lệnh thực sự được thực hiện.

Thiết kế thứ hai, hình 42.10(b), sử dụng tiền giải mã và ghi nhớ chiều dài câu lệnh trong cache câu lệnh. Phương pháp thiết kế này được sử dụng hiệu quả trong giải mã tập lệnh độ dài thay đổi

của X86 [5]. Ưu điểm lớn nhất của lược đồ này là sự đơn giản về số lượng các tầng giải mã trong thiết kế đường ống. Tuy nhiên phương pháp này yêu cầu cấu trúc cache câu lệnh lớn hơn để giữ các thông tin câu lệnh.



HÌNH 42.10 Giải mã câu lệnh kích thước thay đổi: (a) tầng, (b) tiền giải mã

Mô hình cấu trúc

Rất nhiều mô hình cấu trúc tập lệnh đã tồn tại hơn ba thập kỷ. Trước tiên phải kể đến CISC (Complex Instruction Set Computers) là thiết kế đặc trưng có dạng câu lệnh thay đổi, nhiều phương thức định địa chỉ bộ nhớ và nhiều kiểu lệnh. Mục tiêu ban đầu CISC là tạo ra các tập lệnh cho phép ghép các ngôn ngữ lập trình bậc cao để đơn giản hóa công nghệ biên dịch. Thêm vào đó, việc thiết kế dung lượng bộ nhớ nhỏ cũng dẫn đến sự phát triển của hệ CISC. Hai ví dụ kiến trúc chủ yếu của mô hình CISC là Digital VAX và Intel X86.

RISC (Reduced Instruction Set Computers) có kích thước câu lệnh thống nhất, ít phương thức định địa chỉ và giảm bớt chế độ hoạt động. Khái niệm RISC cho phép thiết kế vi kiến trúc của máy thực hiện theo kiểu đường ống dễ dàng hơn, giảm tần số xung nhịp của bộ xử lý và tốc độ của máy. Khái niệm RISC là kết quả của sự nâng cấp ngôn ngữ lập trình, công nghệ biên dịch và kích thước bộ nhớ. HP PA-RISC, Sun SPARC, IBM Power PC, MIPS và DEC Alpha là các ví dụ của kiến trúc RISC.

Các mô hình kiến trúc cho phép nhiều câu lệnh được thực hiện trong một chu kỳ xung nhịp được gọi là VLIW (Very Long Instructure Word). VLIW chuyển một số lượng nhất định các hoạt động thành một câu lệnh đơn dài và trách nhiệm tạo các gói câu lệnh song song được thực hiện bởi trình biên dịch. Các bộ xử lý VLIW thế hệ đầu gặp vấn đề với sự mở rộng mã các câu lệnh. Các máy Multiflow Trace và Cydrome Cydra là ví dụ về công nghệ VLIW. EPIC (Explicitly Parallel Instruction Computing) là một khái niệm tương tự như VLIW, cả hai sử dụng trình biên dịch để tập hợp các câu lệnh cho các ứng dụng song song. Trên thực tế nhiều ý tưởng về cấu trúc EPIC xuất phát từ RISC và VLIW. Nói chung EPIC giải quyết vấn đề mở rộng mã quá mức và các vấn đề của VLIW bằng cách không hoàn toàn loại bỏ chức năng của nó. Khuynh hướng bộ biên dịch được điều khiển bởi cơ chế cấu trúc được coi là một phần của lĩnh vực kiến trúc kiểu EPIC. Intel IA64, Philips Trimedia, và Texas Instruments' C6X là các ví dụ của máy EPIC.

42.5 Mức lệnh song song

Các bộ xử lý hiện đại đang được thiết kế với khả năng thực hiện nhiều hoạt động song song tại mức lệnh. Các bộ xử lý như thế được gọi là khai thác ILP (Instruction-Level Parallelism). *Sử dụng ILP là một khái niệm cấu trúc cơ bản mới trong việc cải thiện khả năng hoạt động của bộ vi xử lý. Hiện nay có rất nhiều công nghệ cấu trúc để định nghĩa cách một cấu trúc có thể khai thác ILP.*

Thực thi động câu lệnh

Giới hạn chủ yếu của công nghệ đường ống là xử lý câu lệnh theo thứ tự. Khi có một câu lệnh trong đường ống thì không có câu lệnh khác được phép xử lý nhằm đảm bảo thực hiện chính xác câu lệnh. Vấn đề này đặc biệt nghiêm trọng đối với các máy đa nhiệm vụ, ở đó mỗi chu kỳ sẽ làm việc với nhiều câu lệnh. Tuy nhiên trong rất nhiều trường hợp một câu lệnh có thể được thực hiện chính xác nếu không có sự phụ thuộc dữ liệu giữa câu lệnh đang được thực hiện và câu lệnh chờ được xử lý tiếp theo. Lập lịch tĩnh là cách tiếp cận hướng trình biên dịch cho các câu lệnh lập lịch để tách riêng các câu lệnh phụ thuộc và tối thiểu hóa các rủi ro. Lập lịch động là một cách tiếp cận khác sử dụng phần cứng để sắp xếp lại việc thực thi các câu lệnh. Khái niệm về thực thi động sử dụng phần cứng để phát hiện sự phụ thuộc giữa các câu lệnh trong dòng câu lệnh nối tiếp và sắp xếp lại chuỗi câu lệnh khi tìm thấy sự phụ thuộc.

Ngày nay hầu hết các vi xử lý hiện đại sử dụng công nghệ lập lịch động không theo thứ tự để tăng số lệnh thực hiện trong một chu kỳ. Các bộ vi xử lý như thế sử dụng khái niệm đường ống lập lịch động. Các câu lệnh đi qua một tầng theo thứ tự, được xử lý không theo thứ tự, sau đó được ghép lại theo thứ tự. Có rất nhiều thành phần chức năng (đã được phát triển trong kiến trúc máy tính) trong chuỗi thông dụng này. Khái niệm chức năng đầu tiên là *scoreboarding*. Scoreboarding là kỹ thuật cho phép các câu lệnh được thực hiện không theo thứ tự khi có các tài nguyên và không có sự phụ thuộc dữ liệu. Scoreboarding xuất phát từ vấn đề lôgic của máy CDC 6600. Mục đích của scoreboarding là xử lý mỗi câu lệnh càng sớm càng tốt.

Một phương pháp thực hiện động ưu việt hơn là phương pháp của Tomasulo. Lược đồ này được sử dụng trong bộ xử lý IBM 360/91. Mặc dù lược đồ này có rất nhiều dạng nhưng khái niệm tránh sự phụ thuộc WAR (Write After Read) và WAW(Write After Write) trong quá trình thực hiện động là đặc điểm chính của phương pháp Tumasulo. Trong lược đồ Tomasulo chức năng của Scoreboarding được cung cấp bởi các *trạm dự phòng*. Các trạm dự phòng là bộ đệm cho các toán hạng lệnh chờ được thực hiện ngay khi có thể. Khái niệm này nhằm thực hiện các câu lệnh ngay khi tất cả các toán hạng nguồn đã sẵn sàng thay vì truy nhập vào các toán hạng đó thông qua thanh ghi. Nghĩa là các câu lệnh chờ xử lý chỉ rõ trạm dự trữ sẽ cung cấp các toán hạng đầu vào của chúng. Hành động này xóa bỏ sự phụ thuộc WAW gây ra bởi các lệnh ghi liên tiếp vào cùng một thanh ghi bằng cách buộc các câu lệnh phải có liên quan với nhau bởi sự phụ thuộc thay vì bởi thanh ghi. "*An Efficient Algorithm for Exploring Multiple Arithmetic Units*" [6] giới thiệu chi tiết về lược đồ Tomasulo.

Dự báo trước

Các lệnh nhánh là trở ngại chính đối với việc thực hiện ILP. Lệnh nhánh buộc trình biên dịch và phần cứng phải dự báo thường xuyên về hướng lệnh nhánh để tìm ra mức song song phù hợp. Dự báo nhằm có thể làm suy giảm khả năng hoạt động do lãng phí một số chu kỳ trong dòng câu lệnh. Chiến thuật dự báo lệnh nhánh giải quyết vấn đề này bằng cách cho phép trình biên dịch và phần cứng tiếp tục xử lý các câu lệnh theo các đường điều khiển đã được dự báo trước, do đó loại bỏ được các câu lệnh thừa.

Hỗ trợ thực hiện dự báo trước cung cấp một phương tiện hiệu quả để loại bỏ các lệnh nhánh khỏi dòng câu lệnh. Thực hiện dự báo trước là sự thực hiện có điều kiện của một câu lệnh dựa trên giá trị của toán hạng nguồn Boolean, gọi là thuộc tính của câu lệnh. Kiến trúc này cho phép trình biên dịch sử dụng giải thuật *if-conversion* để chuyển đổi các lệnh nhánh có điều khiển thành các câu lệnh được định nghĩa thuộc tính và các câu lệnh theo các hướng khác của mỗi nhánh thành các câu lệnh được dự báo trước [7]. Các câu lệnh dự báo trước được thực hiện mà không cần quan tâm đến giá trị dự báo của chúng. Các câu lệnh có giá trị dự báo trước là TRUE thường được thực hiện một cách bình thường. Ngược lại, các câu lệnh có giá trị dự báo là FALSE bị loại bỏ. Thực hiện dự báo trước cho phép trình biên dịch nhận được dự báo trước hiệu quả để đưa ILP tới phần cứng theo các đường thực hiện.

Thực hiện dự báo trước đưa ra một cơ hội để cải thiện việc quản lý lệnh nhánh trong bộ vi xử lý. Loại bỏ thường xuyên các nhánh dự báo sai sẽ làm giảm lỗi dự báo nhánh. Do đó loại bỏ

các hoạt động liên quan đến các nhánh đó. Loại bỏ các lệnh nhánh cũng làm giảm bớt nhu cầu quản lý nhiều lệnh nhánh trong một chu kỳ cho các bộ xử lý. Cuối cùng, thực hiện dự báo trước cung cấp một giao diện hiệu quả cho các trình biên dịch để đưa nhiều đường thực hiện tới phần cứng. Nếu không có sự hỗ trợ trình biên dịch, chi phí để duy trì nhiều đường thực hiện trong phần cứng sẽ rất lớn.

Bản chất của việc thực hiện dự báo trước là khả năng loại bỏ sự thay đổi trạng thái của bộ xử lý dựa trên một số điều kiện thực hiện. Dự báo đầy đủ hỗ trợ điều này thông qua sự kết hợp giữa một tập lệnh và mở rộng vi kiến trúc. Những mở rộng này có thể phân loại thành hỗ trợ cho việc loại bỏ sự thực hiện và các biểu thức trạng thái. Kết quả của trạng thái, được xác định nếu một câu lệnh thay đổi trạng thái, sẽ được lưu trong một tập các thanh ghi 1-bit. Các thanh ghi này được xem như là các file thanh ghi thuộc tính. Giá trị trong các file thanh ghi thuộc tính được kết hợp với mỗi câu lệnh trong tập câu lệnh mở rộng bằng cách sử dụng các toán hạng nguồn bổ sung. Toán hạng này chỉ ra thanh ghi thuộc tính nào sẽ xác định hoạt động có nên thay đổi trạng thái bộ xử lý hay không. Nếu giá trị trong một thanh ghi cụ thể là 1 hoặc true thì câu lệnh sẽ được thực hiện một cách bình thường, nếu giá trị là 0 hoặc false thì câu lệnh sẽ bị loại bỏ.

Giá trị thanh ghi thuộc tính có thể được thiết lập bằng cách sử dụng các câu lệnh định nghĩa dự báo. Ngôn ngữ định nghĩa dự báo được sử dụng là kiến trúc HPL Playdoh [8]. Có một câu lệnh định nghĩa dự báo cho mỗi so sánh trong tập lệnh nguồn. Sự khác nhau chủ yếu với các câu lệnh so sánh truyền thống là những định nghĩa dự báo này phải có hai thanh ghi đích, và các thanh ghi đích của chúng là thanh ghi thuộc tính. Mẫu câu lệnh dự báo có dạng như sau:

$$pred_ < cmp > Pout1_{<type>}, Pout2_{<type>}, scr1, scr2(P_{in})$$

Câu lệnh này xác định giá trị *Pout1* và *Pout 2* tùy theo sự so sánh giữa *src1* và *src2* được chỉ ra bởi <*cmp*>. Sự so sánh <*cmp* > có thể là: bằng nhau (eq), không bằng nhau (nq), lớn hơn (gt). Thuộc tính <*type*> được xác định cho mỗi thuộc tính đích. Các câu lệnh định nghĩa thuộc tính cũng có thể được dự báo (được khai báo bởi P_{in}).

Thuộc tính <*type*> xác định giá trị được viết tới thanh ghi thuộc tính đích dựa trên kết quả của so sánh và thuộc tính đầu vào P_{in}. Đối với mỗi sự liên kết giữa kết quả và P_{in}, một trong ba hành động sau có thể thực hiện trên thuộc tính đích: Nó có thể ghi 1, ghi 0 hoặc không đổi. Có sáu loại thuộc tính rất hữu ích, đó là không điều kiện (*U*), *OR* và *AND* và các phần bù của chúng. Bảng 42.1 là bảng chân lý cho các loại định nghĩa thuộc tính này.

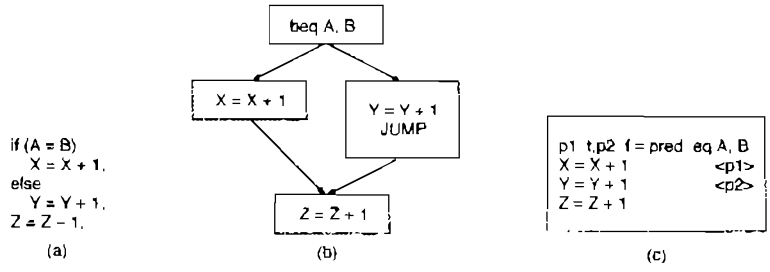
BẢNG 42.1 Bảng chân lý định nghĩa thuộc tính

		P_{out}					
P_{in}	So sánh	*U*	$\overline{U}$	*OR*	$\overline{OR}$	*AND*	$\overline{AND}$
0	0	0	0	—	—	—	—
0	1	0	0	—	—	—	—
1	0	0	1	—	1	0	—
1	1	1	0	1	—	—	0

Các thanh ghi thuộc tính đích không điều kiện luôn được xác định mà không cần quan tâm đến giá trị của P_{in} và kết quả so sánh. Nếu giá trị P_{in} bằng 1, kết quả so sánh được đặt vào trong thanh ghi thuộc tính (hoặc phần bù của nó). Ngược lại, 0 sẽ được ghi vào thanh ghi thuộc tính. Các thuộc tính không điều kiện được sử dụng cho các khối (được thực hiện dựa trên một điều kiện đơn).

Thuộc tính *OR* rất hữu ích khi thực hiện một khối mà khối đó có thể được kích hoạt bởi nhiều điều kiện. Các thanh ghi thuộc tính đích loại *OR* được thiết lập nếu P_{in} bằng 1 và kết quả so sánh bằng 1. Ngược lại, thanh ghi thuộc tính đích không thay đổi. Chú ý rằng thuộc tính loại *OR* phải được khởi tạo bằng 0 trước khi chúng được định nghĩa và sử dụng. Tuy nhiên, sau khi chúng được

khởi tạo, các định nghĩa thuộc tính loại *OR* có thể được sử dụng một cách đồng thời và theo bất kỳ thứ tự nào trên cùng một thanh ghi thuộc tính. Điều này là đúng bởi vì thuộc tính loại *OR* hoặc là ghi 1 hoặc để nguyên giá trị thanh ghi không đổi, do đó cho phép việc thực hiện các điều kiện lôgic *OR*. Thuộc tính loại *AND* tương tự như thuộc tính loại *OR*. Các thanh ghi thuộc tính đích loại *AND* bị xóa nếu P_{in} bằng 1 và kết quả so sánh là 0. Ngược lại, thanh ghi thuộc tính đích không bị thay đổi.



HÌNH 42.11 Chuỗi câu lệnh: (a) mã chương trình, (b) phương pháp truyền thống, (c) thực hiện được dự báo

Hình 42.11 là một ví dụ đơn giản mô tả khái niệm về thực hiện được dự báo. Hình 42.11(a) chỉ ra một chương trình thông dụng" if-then-else". Dòng điều khiển của mã chương trình được thể hiện ở hình 42.11(b). Bằng cách sử dụng if-conversion, mã trong hình 42.11(b) được chuyển đổi thành mã trong hình 42.11(c). Nhánh điều kiện ban đầu được chuyển đổi thành các câu lệnh *pred-eq*.

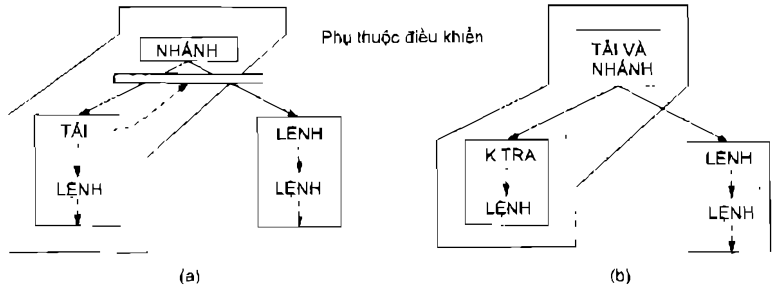
Thanh ghi thuộc tính *p1* được thiết lập để chỉ rằng nếu điều kiện (*A=B*) là đúng và *p2* được thiết lập nếu điều kiện là sai. Thành phần "then" của câu lệnh "if" được xác định trên *p1* và thành phần else được xác định trên *p2*. *Pred-eq* quyết định khi nào lệnh cộng hoặc lệnh trừ được thực hiện và đảm bảo rằng một trong hai thành phần không được thực hiện. Có rất nhiều tiện lợi đối với các mã được dự báo trước. Trước hết, bộ vi xử lý không phải tạo bất kỳ dự báo nhánh nào vì tất cả các nhánh trong mã đã bị loại bỏ. Do đó loại bỏ các rủi ro về dự báo nhánh nhầm. Nhưng quan trọng hơn các câu lệnh được dự báo có thể sử dụng khả năng thực hiện nhiều câu lệnh của bộ vi xử lý hiện đại và tránh được các rủi ro về dự báo nhánh nhầm.

Suy đoán

Số lượng ILP của các khối cơ bản là có hạn trong các chương trình không số. Bởi vì bộ xử lý phải tối ưu hóa và sắp đặt các câu lệnh qua các vùng mã khối cơ bản để đạt được khả năng hoạt động cao hơn. Hơn nữa, các bộ xử lý phải đấu tranh với sự hoạt động tải trễ và các lỗi cache trễ. Khi tải dữ liệu là cần thiết bởi các câu lệnh phụ thuộc, việc thực hiện bộ xử lý phải chờ cho đến khi truy cập cache hoàn thành.

Trong những trường hợp này, các máy không theo thứ tự sẽ sắp xếp lại dòng câu lệnh theo phương pháp động để thực hiện các câu lệnh độc lập. Hơn nữa, các máy không theo thứ tự có nhiều lợi thế khi thực hiện các câu lệnh mà theo sau là các câu lệnh nhánh được dự báo chính xác.Tuy nhiên cách tiếp cận này yêu cầu các mạch phức tạp, do đó làm mất không gian của chip. Hoạt động tương tự cũng có thể đạt được bằng cách sử dụng phương pháp suy đoán thời gian biên dịch tĩnh mà không cần lôgic không theo thứ tự phức tạp. Suy đoán là một công nghệ quan trọng để khai thác ILP trong chương trình. Hoạt động suy đoán là cách tốt nhất để loại bỏ trễ bộ nhớ. Phương pháp này sử dụng cấu trúc tập lệnh hỗ trợ bởi các câu lệnh suy đoán đặc biệt.

Trình biên dịch sử dụng dịch chuyển mã suy đoán để đạt được khả năng hoạt động tốt hơn theo nhiều cách. Trước hết, trong vùng mã không có nhiều ILP, để sử dụng toàn bộ các tài nguyên của bộ xử lý thì các câu lệnh hữu ích có thể được thực hiện. Thứ hai, các câu lệnh tại vị trí ban đầu của các chuỗi phụ thuộc dài có thể được thực hiện sớm để giảm việc tính toán. Cuối cùng các câu lệnh trễ lớn có thể được khởi tạo sớm để ghép việc thực hiện chúng với các hoạt động hữu ích khác. Hình 42.12 mô tả một ví dụ đơn giản về mã trước và sau khi thực hiện quá trình chuyển đổi thời gian biên dịch được suy đoán để thực hiện lệnh tải trên một nhánh điều kiện.



HÌNH 42.12 Chuỗi câu lệnh: (a) thực hiện bình thường, (b) thực hiện có xem xét

Hình 42.12(a) chỉ ra cách câu lệnh nhánh và luồng điều khiển của nó định nghĩa một sự phụ thuộc điều khiển làm giới hạn hoạt động tải khỏi việc sắp đặt theo thứ tự sớm trong mã. Trễ lỗi cache sẽ dừng bộ xử lý trừ khi cơ chế thực hiện không theo thứ tự được sử dụng. Tuy nhiên với hỗ trợ suy đoán này, hình 42.12(b) có thể được sử dụng để loại bỏ độ trễ hoạt động tải.

Để giải quyết vấn đề này thì tải phải được suy đoán hoặc không lỗi. Một tải suy đoán sẽ không báo hiệu một sự ngoại trừ về lỗi, chẳng hạn như đồng bộ địa chỉ hoặc lỗi truy nhập khoảng trống địa chỉ. Thực chất các tải được coi như là *câm* đối với những sự kiện này. Câu lệnh kiểm tra bổ sung trong hình 42.12(b) cho phép những tín hiệu này được phát hiện khi hành động gốc tìm thấy vị trí ban đầu của tải.

Khi những đường thực hiện nhánh khác xảy ra, các tín hiệu *câm* không có nghĩa, có thể bỏ qua. Với cơ chế này, tải có thể được đặt ở trên tất cả sự phụ thuộc điều khiển đang tồn tại, và cung cấp cho trình biên dịch khả năng loại bỏ trễ tải. Chi tiết về suy đoán trình biên dịch có thể thấy ở [9].

42.6 Những xu hướng trong công nghiệp

Ngày nay, công nghiệp vi xử lý là một trong những ngành phát triển mạnh nhất. Nhu cầu thị trường đã khuyến khích sự cạnh tranh, lần lượt mang lại kết quả cao trong việc đổi mới công nghệ.

Những xu hướng của bộ vi xử lý máy tính

Xu hướng hiện tại của bộ vi xử lý máy tính bao gồm đường dẫn tốc độ, tần số đồng hồ cao, đưa ra số lệnh nhiều hơn, suy đoán và thực hiện không theo thứ tự, thực hiện dự đoán, các dạng dữ liệu tự nhiên, cache trên chip lớn hơn, con trỏ động và hỗ trợ bộ đa xử lý. Trong phạm vi của đường ống dẫn, bộ vi xử lý Intel Pentium II có đường ống dẫn sâu gấp gần hai lần so với những Pentium trước đó. Đường ống dẫn sâu của bộ vi xử lý Pentium II cho phép tần số đồng hồ có thể chạy nhanh hơn bộ Pentium.

Trong phạm vi xử lý lệnh, bộ vi xử lý Pentium II có thể giải mã và đưa ra 3 lệnh X86 trong một chu kỳ so với 2 lệnh của Pentium. Pentium II đã cung cấp một lượng lớn diện tích chip cho bộ đệm nhánh mục tiêu, trạm dữ trữ và bộ đệm sắp xếp lại để hỗ trợ thực hiện không theo thứ tự và suy đoán. Những cấu trúc đó cho phép bộ vi xử lý Pentium II có thể thực hiện nhiều lệnh không theo thứ tự, suy đoán và linh hoạt hơn so với bộ Pentium thông thường. Đặc biệt, Pentium II có thể thực hiện đến 40 lệnh X86, lớn hơn Pentium thông thường rất nhiều.

Trong phạm vi thực hiện dự đoán trước, Pentium II hỗ trợ lệnh di chuyển có điều kiện, trong khi đó Pentium thông thường chưa có. Khuynh hướng đó được đẩy mạnh bởi cấu trúc IA-64 thế hệ tiếp theo, trong đó tất cả các lệnh có thể được thực hiện theo điều kiện dưới quá trình điều khiển của các thanh ghi thuộc tính. Chính khả năng đó cho phép trong tương lai bộ vi xử lý thực hiện điều khiển nhiều chương trình ở mức độ cao nhanh hơn những bộ vi xử lý trước đây.

Trong phạm vi của dạng dữ liệu, tập lệnh MMX của Intel đã trở thành chuẩn cho tất cả các bộ vi xử lý X86 hiện nay. Những lệnh đó có nhiều lợi thế do trên thực tế dữ liệu đa phương tiện thường được biểu diễn bởi số lượng bit nhỏ hơn (8-16 bit) so với bề rộng của đường dữ liệu dạng

integer (32-64 bit). Dựa trên quan sát, hoạt động tương tự thường được lặp lại trên tất cả các mục dữ liệu trong các ứng dụng đa phương tiện. Cấu trúc của MMX cho thấy rằng mỗi lệnh MMX đều thực hiện hoạt động giống nhau trên nhiều mục dữ liệu đa phương tiện được đóng gói trong một từ nguyên. Điều này cho phép mỗi lệnh MMX đồng thời xử lý nhiều mục dữ liệu để đạt được tốc độ cao hơn trong các ứng dụng quan tâm. Năm 1998, AMD đưa ra tập lệnh 3DNow! để thực hiện những thao tác cần thiết của các ứng dụng đồ họa 3D. Những lệnh 3Dnow! được thiết kế dựa trên quan niệm cho rằng các mục dữ liệu đồ họa 3D thường được biểu diễn với định dạng con trỏ động chính xác đơn và chúng không yêu cầu việc làm tròn số phức tạp. Do đó có thể ghép hai dữ liệu đồ họa con trỏ động thành một thanh ghi con trỏ động có độ chính xác gấp đôi để dùng trong các ứng dụng đồ họa yêu cầu xử lý con trỏ động hiệu quả hơn.

Trong lĩnh vực cache trên chip lớn, những kỹ thuật thông thường được sử dụng trong bộ vi xử lý máy tính là nhằm mở rộng các cache mức một hoặc để ghép cache mức hai và đôi khi cache mức ba trên chip. Ví dụ, bộ vi xử lý ADM K7 có một cache lệnh mức một 64 Kb và cache dữ liệu mức một 64 Kb. Những cache mức một này có kích thước lớn hơn đáng kể so với cache của thế hệ trước. Một ví dụ khác, bộ vi xử lý Intel Celeron có một cache kết hợp lệnh và dữ liệu mức hai 128 Kb. Những cache lớn này được hỗ trợ bởi mật độ chip ngày càng cao cho phép xây dựng nhiều tranzito bán dẫn trên chíp. Bộ vi xử lý Compaq Alpha 21364 có cả một cache lệnh bậc nhất 64Kb, một cache dữ liệu bậc nhất 64Kb và một cache kết hợp bậc hai 1,5Mb.

Xét về mặt khả năng con trỏ động, nhìn chung các bộ vi xử lý máy tính có thể thực hiện các thao tác con trỏ động mạnh hơn so với những thế hệ vi xử lý trước đó. Ví dụ, thế hệ Pentium II đạt được tốc độ xử lý con trỏ động cao hơn vài lần so với thế hệ Pentium trước đó. Một ví dụ khác, phần lớn các bộ vi xử lý RISC ngày nay có được tốc độ xử lý con trỏ động mà các siêu CPU đối thủ đã có được chỉ cách đây vài năm.

Do nhu cầu ngày càng gia tăng của các máy chủ đa xử lý, rất nhiều bộ vi xử lý ngày nay đã hỗ trợ giao thức giao thức liên kết cache. Ví dụ bộ vi xử lý AMD K7 cung cấp hỗ trợ trực tiếp cho những vận hành đa xử lý liên tục khi nhiều bộ vi xử lý K7 được nối với bus hệ thống. Đây là khả năng mà phiên bản AMD K6 không có được.

Xu hướng của vi xử lý nhúng

Có ba xu hướng phát triển tiêu biểu của bộ vi xử lý nhúng. Xu hướng thứ nhất là tích hợp lõi DSP với một lõi CPU/Bộ điều khiển nhúng. Các ứng dụng nhúng ngày càng cần các chức năng DSP như chức năng mã hóa dữ liệu trong đĩa và cân bằng tín hiệu trong các giao tiếp không dây. Những chức năng này làm tăng chất lượng phục vụ của các sản phẩm máy tính. Tại *diễn đàn vi xử lý nhúng năm 1998*, các hãng ARM, Hitachi và Siemens đều công bố sản phẩm ứng dụng DSP và vi xử lý nhúng [10].

Có ba cách tiếp cận tích hợp DSP và CPU nhúng. Cách thứ nhất là chỉ đơn giản là ghép hai bộ phận tách rời nhau vào một chip. Ưu điểm của cách này là nó đơn giản hóa quá trình phát triển của vi xử lý. Hai bộ phận này thường được lấy ra từ các thiết kế có sẵn. Công cụ phát triển phần mềm có thể được lấy trực tiếp từ môi trường hỗ trợ phần mềm tương ứng của mỗi đơn vị. Nhược điểm là các nhà thiết kế cần phải xử lý hai đơn vị phần cứng độc lập và hai môi trường phát triển phần mềm. Điều này thường làm phức tạp sự phát triển phần mềm và sự kiểm định.

Cách tiếp cận thứ hai là coi DSP như là bộ đồng xử lý với CPU. CPU này tìm tất cả các lệnh và chuyển các lệnh DSP tới bộ đồng xử lý. Thiết kế phần cứng phức tạp hơn so với cách tiếp cận thứ nhất do nhu cầu ghép nối 2 đơn vị gần nhau hơn, đặc biệt trong việc truy cập bộ nhớ. Môi trường phát triển phần mềm cũng cần được chỉnh sửa để hỗ trợ mô hình tương tác đồng xử lý. Ưu điểm là các nhà thiết kế phần mềm ngày nay được tiếp cận với một môi trường liên kết hơn rất nhiều.

Cách tiếp cận thứ 3 là tích hợp các lệnh DSP vào một cấu trúc tập lệnh CPU. Việc này thường yêu cầu phải có một thiết kế mới nhằm thực hiện cấu trúc tập lệnh tích hợp hoàn toàn.

Xu hướng thứ hai trong bộ vi xử lý nhúng là hỗ trợ sự phát triển của các giải pháp chip đơn lẻ cho các thị trường rộng lớn hơn. Rất nhiều nhà cung cấp bộ vi xử lý nhúng đưa ra các thiết kế

được cấp phép và được tích hợp trong một chip lớn hơn bao gồm thiết bị ngoại vi I/O và các thiết kế mạch tích hợp ứng dụng đặc biệt (ASIC). Mô hình này được nhắc tới như một thiết kế chip trong hệ thống. Một bộ vi xử lý được thiết kế để thực hiện các chức năng trong một hệ thống như vậy thường được coi như một lõi được cấp phép.

Xu hướng phát triển thứ ba của bộ vi xử lý nhúng là ứng dụng các kỹ thuật vận hành cao. Trước đây các bộ vi xử lý nhúng rất chậm trong việc chấp nhận cấu trúc vận hành cao và kỹ thuật thực thi. Chúng cũng có xu hướng tái sử dụng các công cụ phát triển phần mềm như trình biên dịch trong lĩnh vực vi xử lý máy tính. Tuy nhiên, do sự gia tăng nhanh chóng về các chức năng mong muốn trong thị trường nhúng, các nhà cung cấp bộ vi xử lý nhúng đang khẩn trương ứng dụng các công nghệ vận hành cao. Xu hướng này đặc biệt được thấy rất rõ trong các bộ vi xử lý DSP. Các hãng Texas Instruments, Motorola/Lucent và Analog Devices đều công bố bộ vi xử lý EPIC DSP trước bộ vi xử lý Intel/HP IA-64 EPIC.

Xu hướng thị trường của bộ vi xử lý

Bạn đọc có quan tâm đến những xu hướng thị trường của bộ vi xử lý có thể tham khảo trong các tài liệu được xuất bản thường kỳ *Microprocessor Report,* do MicroDesign Resources (www.MDRonline.com) phát hành. Trong mỗi lần xuất bản đều tổng kết các đặc trưng vi kiến trúc, tính chất vật lý, và giá thành của bộ vi xử lý.

Tài liệu tham khảo

[1] Turley, J., RISC volume gains but 68K still reigns, *Microprocessor Report,* vol. 12, pp. 14–18, Jan. 1998.

[2] Hennessy, J.L. and Patterson, D.A., *Computer Architecture A Quantitative Approach,* Morgan Kaufman, San Francisco, CA, 1990.

[3] Smith, J.E., A study of branch prediction strategies, *Proceedings of the 8th International Symposium on Computer Architecture,* pp. 135–14, May 1981.

[4] Hwu, W.W. and Conte, T.M., The susceptibility of programs to context switching, *IEEE Transactions on Computers,* vol. C-43, pp. 993–1003, Sept. 1994.

[5] Gwennap, L., Klamath extends P6 family, *Microprocessor Report,* Vol. 1, pp. 1–9, February 1997.

[6] Tomasulo, R.M., An efficient algorithm for exploiting multiple arithmetic units, *IBM Journal of Research and Development,* vol. 11, pp. 25–33, Jan. 1967.

[7] Allen, J.R. et al., Conversion of control dependence to data dependence, *Proceedings of the 10th ACM Symposium on Principles of Programming Languages,* pp. 177–189, Jan. 1983.

[8] Kathail, V., Schlansker, M.S., and Rau, B.R., HPL PlayDoh architecture specification: Version 1.0,Tech. Rep. HPL-93-80, Hewlett-Packard Laboratories, Palo Alto, CA, Feb. 1994.

[9] Mahlke, S.A. et al., Sentinel scheduling: a model for compiler-controlled speculative execution, *ACMTransactions on Computer Systems,* vol. 11, Nov. 1993.

[10] Embedded Microprocessor Forum (San Jose, CA), Oct. 1998.

43

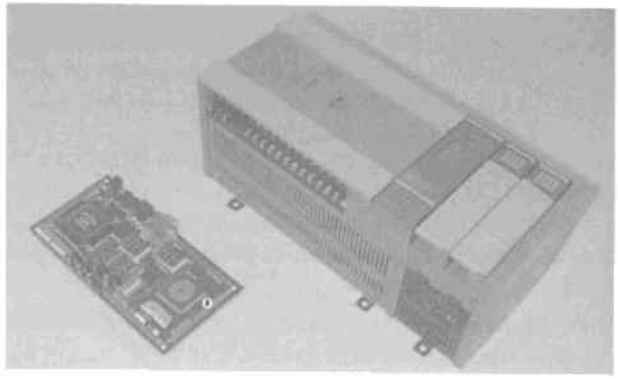
Điều khiển với máy tính nhúng và bộ điều khiển logic khả trình

43.1 Giới thiệu 43-1
43.2 Máy tính nhúng 43-2
43.3 Bộ điều khiển logic khả trình 43-6
43.4 Kết luận 43-12

Job van Amerongen
University of Twente

43.1 Giới thiệu

Hệ thống điều khiển hiện đại thường chứa một vài dạng máy tính, điển hình nhất là máy tính nhúng hoặc bộ điều khiển logic khả trình (PLC). Máy tính nhúng là một hệ thống dựa trên nền vi xử lý hoặc vi điều khiển dùng cho những nhiệm vụ cụ thể không giống như các loại máy tính thông thường khác. Người sử dụng thường không phân biệt được chúng ngoại trừ giao diện điều khiển. PLC là một dạng của bộ điều khiển nhúng được thiết kế để điều khiển các thiết bị công nghiệp.

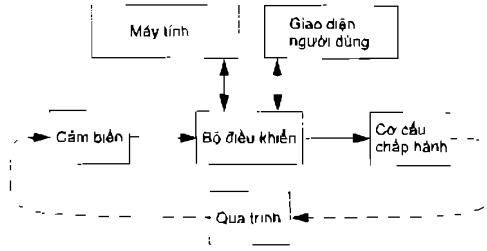


HÌNH 43.1 Một máy tính nhúng với Altera FPGA (phía bên trái) và một bộ điều khiển logic khả trình SLC500 của hãng Allen Bradley (phía bên phải)

Sơ đồ khối của một hệ thống điều khiển điển hình được minh họa trong hình 43.2. Bộ điều khiển giám sát quá trình bằng các cảm biến và tác động vào quá trình bằng các cơ cấu chấp hành. Một giao diện cho phép người sử dụng hay người điều khiển quản lý và theo dõi hệ thống điều khiển. Các giao diện với các máy tính khác được sử dụng với nhiều mục đích như lập trình, theo dõi từ xa, hoặc kết hợp với bộ điều khiển khác.

Khi máy tính được sử dụng trong các ứng dụng điều khiển nó phải thỏa mãn một số yêu cầu kỹ thuật. Hệ thống phải luôn có khả năng đáp ứng và được điều khiển trong suốt quá trình. Điều này

đòi hỏi phần mềm điều khiển phải là phần mềm thời gian thực để nó có thể đáp ứng các sự kiện trong một khoảng thời gian nhất định hoặc tại những khoảng thời gian không đổi. Hệ thống cũng phải được đảm bảo an toàn. Điều này được thực hiện nhờ quá trình giám sát nhiệt độ để tránh quá nhiệt và giám sát mức năng lượng để phát hiện mất nguồn, hoặc các bộ định thời giám sát cho các chương trình không đáp ứng kịp.



HÌNH 43.2 Một sơ đồ khối minh họa ứng dụng máy tính điều khiển

43.2 Máy tính nhúng

Máy tính nhúng là một hệ thống dựa trên nền vi xử lý hoặc vi điều khiển được thiết kế với những chức năng chuyên dụng. Ví dụ điển hình về máy tính nhúng là điện thoại di động, lò vi sóng, thiết bị tính toán cầm tay, các hệ tự động và nhiều hệ thống khác.

Yêu cầu thiết kế và tham số của máy tính nhúng thường khác máy tính đa năng. Máy tính đa năng được thiết kế cho các ứng dụn tính toán lớn và hỗ trợ các chuẩn giao tiếp và chuẩn ngoại vi mới nhất. Trong khi đó máy tính nhúng được thiết kế với sức mạng vừa đủ và chỉ hỗ trợ các giao diện và giao thức cần thiết. Các yêu cầu để thiết kế một máy tính nhúng bao gồm: kích cỡ, nguồn tiêu thụ, tản nhiệt và giá thành.

Cơ sở phần cứng

Hệ thống dựa trên vi điều khiển

Vi điều khiển rất giống với với vi xử lý sử dụng trong máy tính đa năng ngày nay. Nhìn chung, chúng khác với vi xử lý ở độ tích hợp cao, các ngoại vi tích hợp bên trong để giảm thiểu các bộ phận của hệ thống, tiêu thụ năng lượng thấp, cung cấp RAM và ROM trên chip và các đầu vào/ra cho các cảm biến và điều khiển. Do đó, hệ thống nhúng dựa trên vi điều khiển được thiết kế với rất ít các bộ phận ngoại vi. Trái lại, một hệ thống dựa trên vi xử lý lại đòi hỏi RAM ngoài, các thiết bị ngoại vi, các giao diện vào/ra và thường tản rất nhiều nhiệt đến mức cần phải làm mát trong quá trình hoạt động.

Ngoại vi tích hợp trong vi điều khiển bao gồm các giao diện truyền thông nối tiếp (ví dụ như RS232), bộ định thời, bộ phát xung, các bộ đếm sự kiện. Các ngoại vi này hỗ trợ nhiều chức năng điều khiển chấp hành và cảm biến. Ví dụ, các bộ phát xung và định thời có thể được sử dụng để điều khiển động cơ bước. Vi điều khiển ngày càng chuyên môn hoá với các giao diện truyền thông mà chúng hỗ trợ. Trong khi hỗ trợ rất nhiều giao thức quen thuộc như RS232, SPI và I2C, các vi điều khiển gần đây còn hỗ trợ các giao diện khác như là USB.

Để tối thiểu hoá việc tiêu tốn năng lượng, hầu hết các vi điều khiển có thời gian nghỉ hoặc chế độ tạm dừng. Trong thời gian đó không có lệnh nào được thực hiện và tiêu tốn rất ít năng lượng. Vi điều khiển được lập trình để đáp ứng các sự kiện bên ngoài sao cho mã chương trình được thực hiện và năng lượng bị tiêu tốn chỉ khi cần thiết.

Vi điều khiển là thị trường bán dẫn lớn do nhiều thiết bị sử dụng chúng. Có rất nhiều nhà sản xuất và nhiều loại vi điều khiển, từ các thiết bị nhỏ 8 chân với ít chức năng và giá thành rẻ tới các thiết bị lớn với hàng trăm chân nhiều đặc tính và giá thành cao. Sự phân bố rộng này phản ánh tính chuyên dụng cao của máy tính nhúng và thiết kế của nó.

Hệ thống dựa trên FPLD (Field-Programmable Logic Device)

FPLD-Thiết bị logic khả trình cấp trường, ví dụ như CPLD (Complex Programmable Logic Device) và FPGA (Field Programmable Gate Array), gần đây được dùng để thay thế vi điều khiển trong thiết kế máy tính nhúng. FPLD tiêu biểu cho thiết bị phần cứng lập trình được. Ngược lại, vi điều khiển có các chức năng phần cứng cố định và được lập trình bằng phần mềm. Tuy nhiên, cũng có thể thiết kế FPLD giống như vi điều khiển và được lập trình thêm bằng phần mềm. Chức năng phần cứng lập trình được cung cấp cho người thiết kế sự linh hoạt hơn so với giải pháp phần cứng cố định. Nhưng để có được sự linh hoạt này cũng rất phức tạp.

FPLD có thể được thiết kế từ đầu hoặc có thể gồm một hay nhiều lõi thiết kế sơ bộ và các khối ngoại vi. Ví dụ như có thể mua các khối chức năng vi điều khiển cơ bản, các khối chức năng ngoại vi.và tích hợp chúng thành vi điều khiển dựa trên FPLD mà không cần thiết kế lại. Giá của nó thấp hơn thiết kết một chíp từ đầu.

FPLD có thể được lập trình trong quá trình hoạt động, cho phép tính toán tái cấu hình. Đây là một kiểu tính toán lập trình lại hệ thống ở cấp độ phần cứng trong khi hoạt động theo yêu cầu hệ thống. Điều này nghĩa là một thiết bị phần cứng có thể thực hiện nhiều giao thức bus, giao diện hoặc các thuật toán cần thiết.

Hệ thống xử lý tín hiệu số

Trong nhiều trường hợp, thiết bị xử lý tín hiệu số (DSP - Digital Signal Processing) tương tự như vi điều khiển về mặt tích hợp ngoại vi, tiêu thụ năng lượng. Tuy nhiên nó cũng có phần cứng chuyên dụng hỗ trợ hoạt động của DSP, ví dụ như bộ lọc. Các thiết bị DSP rất phù hợp trong các hệ thống xử lý giọng nói và âm nhạc, hoặc trong điều khiển bền vững và các ứng dụng truyền thông. Hỗ trợ phần cứng chuyên dụng của thiết bị nghĩa là cung cấp khả năng tính toán nhanh hơn hơn nhưng có cùng tốc độ xung nhịp và tiêu tán năng lượng như vi xử lý đa năng.

Hệ thống thời gian thực

Hầu hết các hệ thống nhúng phải hoạt động theo thời gian thực, nghĩa là nó phải đáp ứng được các sự kiện bên ngoài, ví dụ như các lệnh của người sử dụng và đọc giá trị của các cảm biến. Khi có yêu cầu một giới hạn trên tuyệt đối về đáp ứng thời gian thì hệ thống là *hệ thời gian thực cứng*, ngược lại là *hệ thời gian thực mềm*. Các hệ thống yêu cầu độ an toàn cao, ví dụ như các hệ tự động và các hệ điều khiển công nghiệp, thường là hệ thống thời gian thực cứng có khả năng tính toán và kiểm tra thời gian trễ tối đa cho các sự kiện.

Tính toán thời gian thực được thực hiện bằng cách sử dụng *ngắt*. Các cơ chế này được hỗ trợ bởi tất cả các vi điều khiển thông thường. Chúng thay đổi luồng thực hiện chương trình khi ngắt xuất hiện. Ví dụ, ngắt có thể xuất hiện khi mức logic thay đổi ở chân thiết bị, thể hiện một trạng thái của cảm biến, hoặc có thể xuất hiện khi người sử dụng ấn một nút trên bàn phím để thực hiện một hành động nào đó.

Hệ thống phần mềm thời gian thực có thể được thiết kế theo yêu cầu của khách hàng hoặc có thể sử dụng một hệ điều hành thời gian thực thương mại (RTOS - Realtime Operating System). Do việc thiết kế hệ thời gian thực điều khiển bằng ngắt có nhiều khó khăn nên sử dụng một hệ RTOS hoàn chỉnh sẽ làm giảm thời gian phát triển.

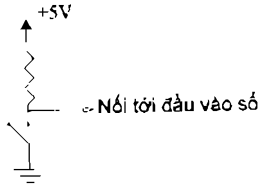
Các mô đun nhúng

Chức năng của các môđun tính toán nhúng ngày càng tăng một cách rõ rệt. Có thể dễ dàng tìm thấy sự tích hợp các vi điều khiển mạnh, giao diện Ethernet và giao thức Internet cơ bản trong cùng một môđun nhỏ. Khả năng tích hợp này làm tăng tốc độ phát triển các ứng dụng điều khiển qua mạng và ứng dụng viễn thám.

Giao diện phần cứng

Các công tắc cơ

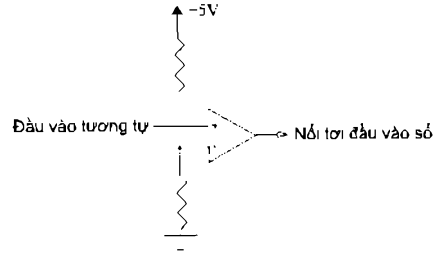
Các công tắc được nối tới đầu logic số bằng điện trở như hình 43.3. Tuy nhiên, bản chất cơ khí của công tắc có thể sinh ra nhiễu của tín hiệu số trong quá trình đóng/mở công tắc. Việc loại bỏ nhiễu có thể được thực hiện bằng phần cứng hoặc phần mềm.



HÌNH 43.3 Công tắc cơ khí nối với đầu vào số trên vi điều khiển thông qua điện trở

Các đầu vào tương tự

Đầu vào tương tự chỉ ra một trong 2 trạng thái được nối với một đầu vào logic số qua một bộ so sánh đơn giản (hình 43.4). Điện áp ngưỡng được thiết lập bằng mạch chia điện trở. Bộ so sánh phát ra một tín hiệu số cho biết điện áp đầu vào lớn hơn hay nhỏ hơn điện áp ngưỡng. Phương pháp này có thể được sử dụng cho các cảm biến, ví dụ như các bộ ngắt quang học (bộ đếm, bộ định vị dịch chuyển động cơ), cảm biến giới hạn nhiệt độ và nhiều thiết bị khác.



HÌNH 43.4 Đầu vào số điều khiển bởi một bộ so sánh để xác định điện áp tương tự lớn hơn hay nhỏ hơn điện áp ngưỡng

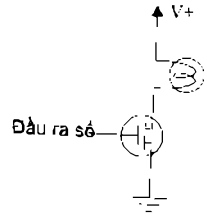
Khi cần đo giá trị điện áp, có thể sử dụng bộ chuyển đổi tương tự-số (ADC) để biểu diễn nối tiếp hoặc song song giá trị điện áp với dải chính xác bất kỳ từ 8 bit tới 16 bit hoặc lớn hơn. Bộ chuyển đổi nối tiếp có thể chỉ cần hai chân vào/ra số trên vi điều khiển để dữ liệu truyền. Trong khi đó bộ chuyển đổi song song đòi hỏi số chân tương đương với độ phân giải, nhưng có thể truyền toàn bộ giá trị điện áp trong một lần truyền để tăng tốc quá trình chuyển đổi.

Một số vi điều khiển có các ngoại vi ADC tích hợp với nhiều kênh đầu vào tạo ra các hệ cảm biến tương tự với giá thành thấp.

Các điện áp tương tự rất nhỏ có thể được khuyếch đại bằng một bộ khuyếch đại trước khi chuyển đổi tương tự-số. Các bộ khuyếch đại có hệ số khuyếch đại cao, trở kháng cao và vì vậy phù hợp với các cảm biến có dòng và áp điều khiển nhỏ.

Các cơ cấp chấp hành đơn giản

Các máy tính nhúng thường không có khả năng điều khiển trực tiếp các cơ cấu chấp hành, do các các cơ cấu chấp hành yêu cầu dòng và áp không tương thích với mạch số. Tuy nhiên, một số cảm biến chỉ cần vài mạch giao diện đơn giản. Các cơ cấu chấp hành on/off đơn giản như đèn LED, cuộn rơle... có thể được điều khiển bằng đầu ra số sử dụng transistor làm công tắc (hình 43.5). Đầu ra số điều khiển trạng thái on/off của tranzito cho phép hoặc không cho phép dòng đi qua cơ cấu chấp hành.



HÌNH 43.5 Đầu ra số có thể điều khiển một thiết bị dòng cao thông qua tranzito

Các động cơ cũng có thể được điều khiển bằng cách sử dụng transistor nối các đầu ra số với các cuộn dây động cơ. Bóng đèn trong hình 43.5 có thể thay thế bằng một động cơ một chiều để điều khiển đóng mở đơn giản động cơ. Một bộ 4 tranzito thiết kế theo kiểu cầu H (H-bridge) cho phép điều khiển động cơ quay theo chiều bất kỳ. Cấu hình 2 mạch cầu H có thể được sử dụng để điều khiển động cơ bước. Trong tất cả các trường hợp, tốc độ và chiều quay được điều khiển trực tiếp bằng máy tính nhúng thông qua các đầu ra số của nó.

Đầu ra tương tự

Với các cơ cấu chấp hành đòi hỏi điện áp hoặc dòng tương tự, bộ chuyển đổi số-tương tự (DAC) có thể được dùng làm giao diện giữa máy tính nhúng và cơ cấu chấp hành. So với ADC, DAC rất đa dạng về độ phân giải, tốc độ chuyển đổi, số kênh, v.v. Thông thường, dòng điện điều khiển các thiết bị này là không đủ, do đó cần thêm bộ khuyếch đại đệm để thoả mãn yêu cầu về dòng cho các cơ cấu chấp hành.

Ngôn ngữ lập trình

Hầu hết các máy tính nhúng được lập trình bằng ngôn ngữ bậc thấp để điều khiển tối đa các tài nguyên phần cứng. Những phần chương trình có yêu cầu chặt chẽ về thời gian thường được lập trình bằng ngôn ngữ Assembly, đó là ngôn ngữ bậc thấp nhất mà vi điều khiển có thể hiểu được. Ngôn ngữ C thường được sử dụng cho lập trình cấu trúc bậc cao. Các ngôn ngữ bậc cao hơn như C++ hoặc Java không phù hợp lắm cho lập trình hệ nhúng vì chúng đòi hỏi vùng nhớ lớn hơn và không được thiết kế để truy nhập trực tiếp các tài nguyên phần cứng.

Hình 43.6 minh họa một đoạn chương trình ngôn ngữ Assemlby được viết cho vi điều khiển PIC16F84A. Nó cấp nguồn cho động cơ một chiều (qua một mạch giao diện ngoài) khi 2 đầu vào số cùng đạt mức logic 1. Chương trình chạy liên tục, luôn kiểm tra trạng thái của 2 đầu vào số đó.

```
loop:
        btfss   PORTA, 0        Kiểm tra đầu vào số: bit 0 của cổng A
        goto    turnoff         Nhảy đến turnoff nếu tín hiệu bằng 0
        bsf     PORTA, 1        Kiểm tra đầu vào số: bit 1 của cổng A
        goto    turnoff         Nhảy đến turnoff nếu tín hiệu bằng 0

        bsf     PORTB, 5        Cho phép động cơ hoạt động bằng việc thiết lập bit 5
                                của cổng B
        goto    loop            Quay lại kiểm tra đầu vào

turnoff:
        bcf     PORTB, 5        Xoá bit 5 ở cổng B, động cơ không hoạt động
        goto    loop            Quay lại kiểm tra đầu vào
```

HÌNH 43.6 Một đoạn mã Assembly cho vi điều khiển PIC 16F84A

Một đoạn chương trình tương tự được viết bằng C như hình 43.7. Mã nguồn ngắn gọn và dễ đọc hơn vì C là một ngôn ngữ bậc cao hơn Assembly.

```
while (1){
        if(PA0&& PA1){      // Kiểm tra trạng thái của bit 0 và 1 của cổng A
            PB5=1;          // Set bit 5 của cổng B
        }else{
            PB5=0;          // Xoá bit 5 của cổng B
        }
}
```

HÌNH 43.7 Đoạn chương trình bằng C thực hiện chức năng như chương trình trong hình 43.6

43.3 Bộ điều khiển logic khả trình

Bộ điều khiển logic khả trình hiện đại (PLC) là sự phát triển của điều khiển dựa trên rơle. Sự thay đổi công nghệ đã bắt đầu từ những năm 60 khi bộ điều khiển rơle cơ điện có nhiều hạn chế, hãng General Motors đã nghiên cứu để thay thế bằng điện tử. Câu trả lời đã được Modicon, người chuyên cung cấp hệ điều khiển dựa trên nền vi xử lý, đưa ra vào năm 1970. Ngôn ngữ lập trình được mô hình theo các biểu đồ logic hình thang để dễ dàng chuyển giao cho người thiết kế, xây dựng và người bảo dưỡng các bộ điều khiển mới này. Trong những năm 70, công nghệ này đã được cải tiến và chứng thực, và từ những năm 80 chúng đã trở nên phổ biến trong các nhà máy.

Hầu hết các bộ phận của PLC đều là dạng card để có thể thay thế nhanh khi có lỗi. Một ứng dụng PLC điển hình có khoảng 100 các đầu vào/ra. Tuy nhiên, phạm vi ứng dụng của nó còn lớn hơn nhiều. Một PLC nhỏ giá 200 USD có thể có 6 đầu vào và 4 đầu ra. Một ứng dụng lớn có thể bao gồm nhiều PLC làm việc cùng nhau để điều khiển toàn nhà máy và tổng số có thể đến 10 nghìn các đầu vào/ra. Nói chung giá của phần cứng PLC trên mỗi đầu vào/ra khoảng 10 đến 15 USD. Đấy là chưa gồm các cảm biến (giá khoảng 50-100 USD), các cơ cấu chấp hành (giá khoảng 50-200 USD), lắp đặt (giá khoảng 10-100), thiết kế hoặc lập trình.

Hệ thống điều hành sản xuất luôn luôn đòi hỏi điều khiển logic và đôi khi cần điều khiển liên tục. Điều khiển logic gồm kiểm tra các tín hiệu nhị phân đầu vào (on hoặc off) từ các cảm biến và xuất các tín hiệu đầu ra để điều khiển các cơ cấu chấp hành. Một ví dụ đơn giản là cảm biến quang Dùng để phát hiện hộp trên băng tải và tác động đến xi lanh thủy khí để điều khiển hộp. Hệ điều khiển liên tục thường ít được sử dụng hơn vì giá thành đắt và phức tạp hơn. Một bộ điều khiển liên tục điển hình có thể sử dụng một card đầu ra tương tự (1000 USD) để xuất ra điện áp điều khiển vận tốc băng tải.

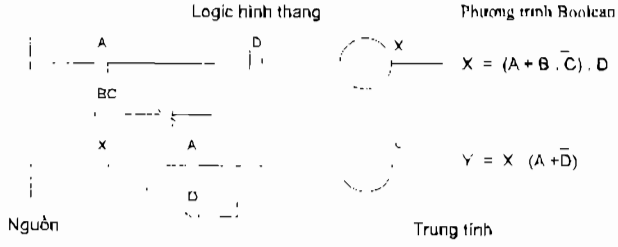
Ngôn ngữ lập trình

Tất cả các PLC đều có thể lập trình được với biểu đồ logic hình thang. Ngôn ngữ hình thang sử dụng các công tắc đầu vào (ký hiệu bằng 2 gạch thẳng đứng) và các cuộn dây đầu ra (ký hiệu bằng một vòng tròn). Công tắc có thêm một ký hiệu gạch chéo là công tắc thường đóng. Trong logic hình thang này đường bên trái là nguồn. Khi các công tắc được đóng theo trình tự phù hợp, dòng điện sẽ chạy qua cuộn dây đến đường trung tính ở phía bên phải.

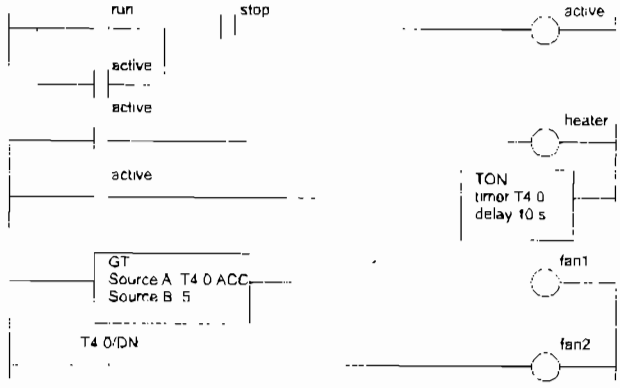
Xét ví dụ trong hình 43.8. Giả sử đường bên trái là đường *dây nóng* (có nguồn) và đường bên phải là *trung tính*. Khi các công tắc đóng và mở theo trình tự phù hợp thì đường dây sẽ thông và cho phép dòng đi tới cuộn dây đầu ra và do đó kích hoạt chúng. Logic chương trình được thực hiện từ trái sang phải. Ở bậc thang thứ nhất nếu A và D tích cực thì đầu ra X sẽ tích cực. X cũng có thể được tích cực khi B tích cực, C không tích cực và D tích cực. Ở bậc thang thứ hai, đầu ra Y sẽ được tích cực nếu X tích cực và A tích cực hoặc X tích cực và D không tích cực. Lưu ý rằng các nhánh là hàm OR và các công tắc nối tiếp là hàm AND. Có thể viết các bậc thang logic bằng biểu logic như bên phải của hình vẽ.

Ví dụ trong hình 43.8 chỉ chứa logic trạng thái, nhưng hình 43.9 là một ví dụ phức tạp hơn của chương trình logic hình thang sử dụng các thiết bị định giờ và các biến nhớ. Khi đầu vào *run* tích cực thì đầu ra *heater* sẽ được tích cực, 5 giây sau *fan1* sẽ tích cực, *fan2* được tích cực sau 10 giây. Bậc thang thứ nhất của chương trình cho phép hệ thống được bắt đầu với một nút bấm thường mở *run*,

hoặc dừng với một nút bấm thường đóng *stop*. Tất cả các đầu vào stop là khóa thường đóng, do đó các công tắc trong bậc thang này phải là thường mở để giữ nguyên trạng thái logic. Đầu ra *active* được nối phân nhánh với công tắc *run* để *tự giữ* trạng thái tích cực. Bậc thang tiếp theo sẽ kích hoạt đầu ra *heater* khi hệ thống tích cực. Bậc thang thứ 3 sẽ kích hoạt timer khi *active* được tích cực. Khi đầu vào của bộ định thời TON tích cực, bộ định thời T4:0 sẽ bắt đầu đếm và thành phần T4:0.ACC sẽ bắt đầu tăng cho đến đến khi đạt 10 giây thì bit T4:0/DN được tích cực và giữ nguyên trạng thái đó cho tới khi đầu vào của timer hết tích cực. Ở bậc thang thứ 4, khi thời gian của timer lớn hơn 5 thì đầu ra *fan1* sẽ được tích cực. Bậc thang cuối cùng sẽ tích cực *fan2* khi timer đã đếm được 10 giây.



HÌNH 43.8 Một chương trình logic hình thang đơn giản



HÌNH 43.9 Một chương trình điều khiển logic hình thang phức tạp

Bộ PLC thực hiện (quét) một chương trình logic hình thang liên tục. Dải thời thực hiện điển hình là từ 5 đến 100 ms. Thời gian thực hiện nhanh hơn đòi hỏi vi xử lý có tốc độ cao hơn.

Các ký hiệu và mẫu hàm sử dụng trong hình 43.9 được phát triển theo nhà sản xuất PLC. Trong thực tế, các hãng phát triển các phiên bản logic hình thang khác nhau.

Các ngôn ngữ lập trình IEC 61131-3

Chuẩn IEC 61131 (trước đây là chuẩn IEC 1131) ra đời để thống nhất các PLC [3,5]. Các phần chính của chuẩn được liệt kê dưới đây.

IEC 61131-1 Tổng quan

IEC 61131-2 Yêu cầu và phương pháp kiểm tra

IEC 61131-3 Kiểu dữ liệu và ngôn ngữ lập trình

IEC 61131-4 Hướng dẫn sử dụng

IEC 61131-5 Truyền thông

IEC 61131-7 Điều khiển mờ

Phần quan trọng nhất của chuẩn là nguyên tắc lập trình, IEC 61131-3. Nó mô tả 5 mô hình lập trình cơ bản bao gồm: biểu đồ hình thang (LD - ladder diagrams), liệt kê lệnh (IL - instruction list), văn bản cấu trúc (ST - structured text), biểu đồ chức năng tuần tự (SFC - sequential function

charts) và biểu đồ khối chức năng (FBD - function block diagrams). Các ngôn ngữ này được thiết kế để làm việc cùng nhau. Có thể thực hiện một hệ thống bằng cách kết hợp các ngôn ngữ đó, hoặc thực hiện các chức năng giống nhau trong các ngôn ngữ khác nhau. Các phần sau sẽ mô tả các chương trình ST, SFC và FDB.

Ngôn ngữ văn bản cấu trúc

Một chương trình văn bản cấu trúc được mô tả như hình 43.10. Chương trình này có chức năng giống như ví dụ hình thang logic ở trên. Dòng thứ nhất định nghĩa tên chương trình. Sau đó là các định nghĩa biến. Các biến *run* và *stop* là các biến đầu vào của bộ điều khiển từ cảm biến và các công tắc. Các biến *heater*, *fan1* và *fan2* là các biến đầu ra tới các cơ cấu chấp hành của hệ thống. Các biến *active* và *delay* là các biến nội, chỉ sử dụng trong chương trình.

```
PROGRAM example
VAR_INPUT
    run: BOOL;
    stop: BOOL;
END_VAR
VAR_OUTPUT
    heater: BOOL;
    fan1: BOOL;
    fan2: BOOL;
END_VAR
VAR
    active: BOOL
    delay: TON
END_VAR

    active:= (run OR active) & stop;

    heater:=active;
    delay(EN:=active, PRE:=10);
    IF (delay.ACC>5) THEN
                        fan1:=1;
    ELSE
                        fan1:=0;
    END_IF;
    fan2:=delay.DN;
END_PROGRAM;
```

HÌNH 43.10 Một chương trình văn bản cấu trúc tương đương với hình 43.9

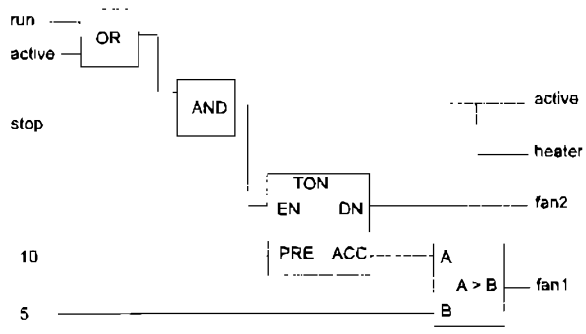
Ngay sau các khai báo biến là chương trình chính. Trong chương trình, 2 dòng đầu tiên là đặt các biến *active* và *heater*. Lệnh *delay* gọi timer, tham số EN:=active kích hoạt timer và PRE:=10 đặt thời gian trễ là 10s. Dòng sau đó sử dụng lệnh "*if*" để đặt giá trị cho *fan1* bằng cách sử dụng giá trị tích luỹ *delay.ACC*. Giá trị của *fan2* được đặt khi timer đã đạt thời gian trễ mong muốn.

Ngôn ngữ văn bản có cấu trúc rất thông dụng và có khả năng thay thế hoàn toàn ngôn ngữ hình thang logic để trở thành ngôn ngữ lập trình phổ biến nhất.

Ngôn ngữ biểu đồ khối chức năng

Mô hình luồng dữ liệu là mô hình cơ bản của biểu đồ khối chức năng. Trong các chương trình này, luồng dữ liệu đi từ các đầu vào bên trái đến các đầu ra bên phải. Ví dụ trong hình 43.11 tương đương với ví dụ hình thang logic ở trên. Hàm OR và AND được sử dụng để đặt các biến *active* và *heater*. Timer TON sử dụng đầu vào EN để tích cực và đầu vào PRE đặt thời gian trễ để điều khiển biến tích luỹ ACC và đầu ra DN. Đầu ra DN điều khiển *fan2*, còn giá trị ACC được so sánh với giá trị 5 để đặt đầu ra *fan1*.

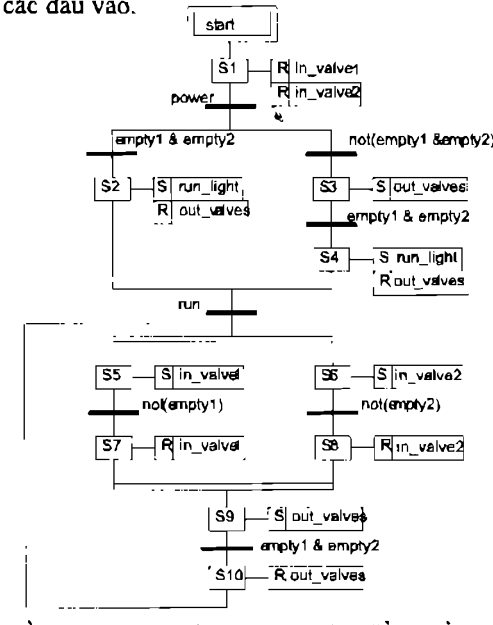
Biểu đồ luồng dữ liệu rất hữu dụng cho thiết kế hệ thống điều khiển cấp cao.



HÌNH 43.11 Chương trình FBD tương đương với hình 43.9

Ngôn ngữ biểu đồ chức năng tuần tự

SFC được sử dụng để miêu tả hệ thống trong trường hợp chỉ có các *bước* (step) và các *chuyển tiếp* (transition). *Bước* mô tả một chế độ vận hành hoặc trạng thái mà một *hành động* được thực hiện. Thông thường là để đặt các đầu ra. *Chuyển tiếp* xác định sự thay đổi của các trạng thái, thông thường bằng cách kiểm tra các đầu vào.



HÌNH 43.12 Một chương trình SFC điều khiển bình mức

Hình 43.12 là ví dụ về một chương trình SFC điều khiển bình chứa. Bình chứa này được xả cạn khi bộ điều khiển bắt đầu hoạt động và *power* được kích hoạt. Sau đó đầu vào *run* sẽ bắt đầu chu kỳ làm việc, các bình được đổ đầy và sau đó bị bơm cạn, quá trình này lặp đi lặp lại.

Trong ví dụ này, luồng điều khiển bắt đầu bằng bước đầu tiên là *start* và sau đó đến bước *S1*. Hành động được thực hiện trong bước *S1* là R, reset đầu ra *in_valve1* và *in_valve2*. Hệ thống sẽ ở bước *S1* cho tới khi có điều kiện chuyển đổi được thỏa mãn bởi đầu vào *power*. Sau đó sẽ có 2 khả năng xảy ra. Nếu cả *empty1* và *empty2* cùng tích cực, nhánh bên trái sẽ được tích cực, ngược lại điều kiện chuyển đổi bên phải được thỏa mãn và nhánh đó được phép hoạt động. Nhánh trái tích cực *run_light* (với lệnh S) và reset *outlet_valve*. Nhánh phải sẽ tích cực *outlet_valves* cho tới khi *empty1* và *empty2* cùng tích cực. Lúc này *run_light* sẽ được tích cực và *outlet_valves* bị reset. Cho dù là nhánh nào đã thực hiện, luồng thực hiện sẽ phải dừng tại điều kiện chuyển tiếp sau đó cho đến khi đầu vào *run* tích cực.

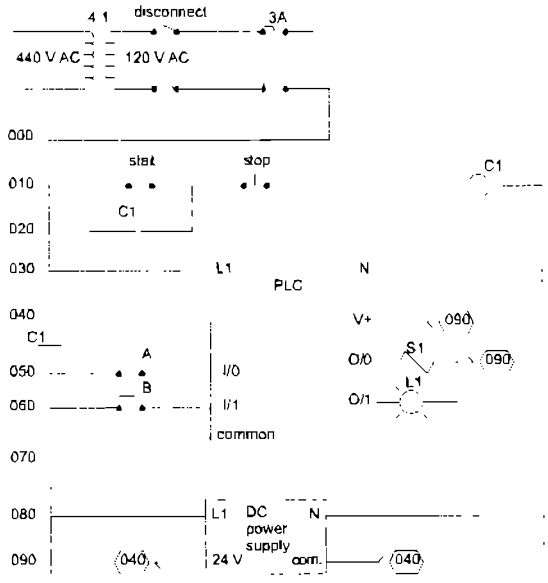
Sau khi điều kiện chuyển tiếp *run* được kích hoạt, luồng thực hiện sẽ vào cả 2 nhánh trái và phải, được thể hiện bằng 2 đường gạch ngang. Nhánh trái đổ đầy vào một bình, nhánh phải đổ đầy vào bình còn lại một cách độc lập. Khi cả 2 nhánh được hoàn thành thì luồng thực hiện sẽ nối lại tại đường ngang thứ 2 và sau đó tích cực S9. Sau S10 luồng thực hiện sẽ trở về vị trí sau điều kiện chuyển tiếp *run*.

Phương pháp lập trình SFC khác với các phương pháp lập trình khác ở chỗ chương trình không thể hoàn thành trong một vòng quét, trong khi đó các chương trình khác phải hoàn thành trong một vòng quét.

Giao diện

Các yêu cầu về lắp đặt và cách ly cho PLC xuất phát từ nhu cầu đảm bảo an toàn cho con người và thiết bị. Hình 43.13 minh hoạ một biểu đồ đi dây điển hình cho ứng dụng PLC. Trên đỉnh biểu đồ là máy biến thế dùng để hạ thấp nguồn điện áp cao. Tiếp theo là cầu dao và cầu chì. Sau đó nguồn được chia làm 2 nhánh trái và phải, tương tự như biểu đồ hình thang đã trình bày ở trên. Dòng 10 là hệ thống quản lý nguồn chính. Nó bao gồm một nút bấm start thường mở và một nút bấm stop thường đóng. Các công tắc này điều khiển rơle C1. Lưu ý rằng nếu nguồn được cung cấp cho cuộn dây C1, nó sẽ đóng công tắc C1 và giữ C1 cho đến khi nút stop được nhấn. Các công tắc khác được sử dụng trên nhánh bên trái để ngắt PLC và nguồn cung cấp một chiều khỏi các nguồn đầu vào. Các công tắc này là cần thiết để nút stop của hệ điều khiển có thể ngắt trực tiếp nguồn cung cấp. Đây là nguyên tắc thiết kế.

Trong ví dụ này PLC được cấp nguồn 120 V AC. Có 2 đầu vào 120V AC từ các nút bấm thường mở. Nguồn 24 V DC là đầu vào V+ của PLC, nó sẽ đưa nguồn tới cuộn dây *S1* và đèn hiển thị *L1*.



HÌNH 43.13 Một ví dụ đi dây PLC

Các tính năng nâng cao

PLC thường được dùng trong các ứng dụng khác ngoài các bài toán điều khiển logic. Sau đây là một vài chức năng lập trình và vào/ra nâng cao:

Khả năng tính toán: PLC có khả năng làm các phép tính đơn giản. Các PLC nhỏ chỉ có thể tính toán các số nguyên, trong khi các PLC lớn hơn có thể tính toán các phép tính số thực.

Vào/ra tương tự: Các giá trị vào/ra của dòng và áp có là liên tục.

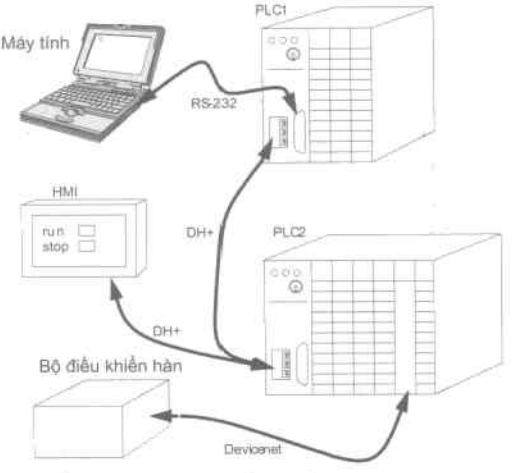
Điều khiển phản hồi: Bộ điều khiển tỷ lệ, tích phân, vi phân (PID) được cung cấp như là một khối hàm và được sử dụng với các cổng vào/ra tương tự.

Truyền thông: Có khả năng truyền dữ liệu bằng các chuỗi ký tự qua cổng nối tiếp hoặc truyền các vùng nhớ của PLC bằng các giao thức phù hợp.

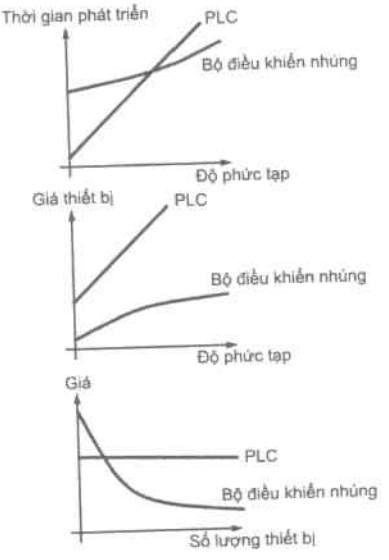
Chuỗi ký tự ASCII: Có các hàm xử lý các chuỗi ký tự ASCII.

Hệ thống: Xác định lỗi, theo dõi các trạng thái, phục vụ ngắt...

Điều khiển mờ: Một vài PLC có chức năng điều khiển mờ cho các đối tượng điều khiển phi tuyến.



HÌNH 43.14 Ví dụ về truyền thông PLC



HÌNH 43.15 So sánh các giải pháp điều khiển

Ở các PLC nhỏ, truyền thông chỉ để lập trình. Nhưng trong nhiều ứng dụng PLC được sử dụng để truyền thông với các thiết bị khác. Trước kia, hầu hết việc truyền thông là độc quyền, đóng, hoặc theo chuẩn. Gần đây nhiều chuẩn truyền thông mở đã được phát triển và được cung cấp bởi các hãng, ví dụ như Profbus, DeviceNet, CanBus và ModBus. Cũng có xu hướng sử dụng các chuẩn truyền thông phổ biến như RS-232, RS-422, RS-485 và Ethernet. Một ví dụ về hệ thống tự động hoá được minh hoạ trọng hình 43.14. Kết nối RS-232 được sử dụng giữa laptop (ví dụ dùng COM1) và PLC1 để lập trình. Dùng DH+ để nối PLC1, PLC2 và HMI, đây là chuẩn truyền thông riêng của Allen-Baradley. Người điều khiển có thể sử dụng giao diện người máy (HMI) để hiển thị dữ liệu, truy nhập các biến đầu vào và truyền các biến này trực tiếp xuống cả 2 PLC. Devicenet, một chuẩn truyền thông mở, được sử dụng để kết nối PLC2 tới bộ điều khiển hàn.

43.4 Kết luận

PLC và các bộ điều khiển nhúng là các công nghệ hỗ trợ nhau, khi được ứng dụng một cách hợp lý chúng cung cấp các giải pháp tin cậy có chi phí thấp cho các bài toán điều khiển. Hình 43.15 mô tả mối quan hệ tương quan giữa các bộ điều khiển này. Nói chung, trong các hệ đơn giản, thời gian phát triển ban đầu của bộ điều khiển nhúng lâu hơn bộ PLC. Với các hệ thống lớn hơn, phức tạp hơn, bộ điều khiển nhúng có lợi thế hơn khi tận dụng thư viện phần mềm và công cụ thiết kế có sẵn. Khi sử dụng PLC, giá thành phần cứng trên mỗi thiết bị là đắt hơn. Giá thành để phát triển một máy tính nhúng là rất cao, nhưng khi số lượng thiết bị lớn sẽ giảm tối đa chi phí. Do đó, bộ điều khiển nhúng thường được lựa chọn cho các ứng dụng được sản xuất hàng loạt và cho phép thời gian phát triển lâu hơn, ví dụ như robot đồ chơi. PLC thường được lựa chọn cho các ứng dụng chỉ đòi hỏi một vài bộ điều khiển và được hoàn thành trong thời gian tương đối ngắn, ví dụ như các máy sản xuất đồ chơi.

Tài liệu tham khảo

[1] Bryan, L.A., Bryan, E.A., *Programmable Controllers*, Industrial Text and VideoCompany, 1997.

[2] Filer, R., Leinonen, G., *Programmable Controllers and Designing Sequential Logic*, Saunders College Publishing, 1992.

[3] Lewis, R.W., *Programming Industrial Control Systems using IES1131-3*, The Institution of Electrical Engineers, 1998.

[4] Petruzella, F., *Programmable Logic Controllers*, Second Edition, McGraw-Hill, 1998.

[5] *Programmable Controllers—Part 3: Programming Languages*, IEC 61131-3 Ed. 1.0, 1993.

[6] Stenerson, J., *Fundamentals of Programmable Logic Controllers, Sensors and Communications*, Prentice- Hall, 1998.

[7] Webb, J.W., Reis, R.A., *Programmable Logic Controllers, Principles and Applications*, Prentice-Hall, 1995.

VI

Phần mềm và thu thập dữ liệu

44 Giới thiệu về việc thu thập dữ liệu *Jace Curtis* 44-1

45 Các kỹ thuật đo: cảm biến và bộ chuyển đổi *Cecil Harrison* 45-1

Giới thiệu • Bộ chuyển đổi lực và chuyển động • Bộ chuyển đổi quá trình • Hoạt động của bộ chuyển đổi • Sự phù hợp giữa tải và bộ chuyển đổi

46 Biến đổi A/D và D/A *Mike Tyler* 46-1

Giới thiệu • Lấy mẫu • Đặc điểm kỹ thuật bộ ADC • Đặc điểm kỹ thuật bộ DAC

47 Chuẩn hóa tín hiệu *Stephen A. Dyer* 47-1

Tác động tuyến tính • Tác động phi tuyến

48 Các hệ thống thiết bị đo dựa trên máy tính *Kris Fuller* 48-1

Sức mạnh của phần mềm • Số hóa thế giới tương tự • Triển vọng

49 Thiết kế và phát triển phần mềm *Margaret H. Hamilton* 49-1

Khái niệm phần mềm • Bản chất của kỹ thuật phần mềm • Sự phát triển trước thực tế • Thực nghiệm với công nghệ DBTF • Kết luận

50 Ghi dữ liệu *Tom Magruder* 50-1

Tổng quan • Bối cảnh lịch sử • Yêu cầu chức năng của thu thập dữ liệu • Hệ thống thu thập dữ liệu • Kết luận

44

Giới thiệu hệ thống thu thập dữ liệu

Jace Curtis

National Instruments, Inc

Mục đích của một hệ thống thu thập dữ liệu là nắm bắt và phân tích một số hiện tượng vật lý trong thực tế. Ánh sáng, nhiệt độ, áp suất và mômen là một trong nhiều dạng tín hiệu khác nhau có thể tương tác với hệ thống thu thập dữ liệu. Một hệ thống thu thập dữ liệu cũng có thể sinh ra các tín hiệu điện một cách đồng thời. Những tín hiệu này có thể điều khiển các hệ cơ học một cách thông minh hoặc tạo ra kích thích để hệ thống thu thập dữ liệu có thể đo được các đáp ứng. Một hệ thống thu thập dữ liệu cung cấp phương pháp kiểm tra bằng thực nghiệm các thiết kế, lý thuyết và các hệ thống thực tế phục vụ cho việc kiểm nghiệm hay nghiên cứu. Hình 44.1 minh họa một môđun thu thập dữ liệu dựa trên máy tính điển hình.

Chẳng hạn, việc thiết kế và sản xuất một chiếc ô tô hiện đại phần lớn phụ thuộc vào quá trình thu thập dữ liệu. Trước hết, các kỹ sư sẽ sử dụng những dữ liệu thu thập được để kiểm tra phần thiết kế của các bộ phận của ô tô. Khung xe có thể được kiểm tra ứng suất cơ học, độ bền và độ ồn. Độ rung và nhiệt độ của động cơ cũng là thông tin cần thiết để đánh giá chất lượng thiết kế. Các nhà nghiên cứu và các kỹ sư sau đó có thể sử dụng những thông tin này để tối ưu hoá thiết kế của chiếc xe nguyên mẫu. Tiếp đó, nguyên mẫu này có thể được kiểm tra dưới nhiều điều kiện khác nhau để lấy những thông tin cần thiết thông qua quá trình thu thập dữ liệu. Sau một vài lần sửa đổi thiết kế và thu thập dữ liệu, chiếc xe đã sẵn sàng đưa vào sản xuất. Các thiết bị thu thập dữ liệu có thể giám sát quá trình lắp ráp và kiểm tra xem xe có được lắp ráp theo đúng thông số kỹ thuật hay không.

Ban đầu, các thiết bị thu thập dữ liệu đứng độc lập và được điều khiển thủ công bởi người vận hành. Khi PC xuất hiện, các thiết bị thu thập dữ liệu và các thiết bị đo được nối với máy tính thông qua cổng nối tiếp, cổng song song, hoặc một số giao diện khác. Chương trình máy tính có thể điều khiển các thiết bị này một cách tự động và thu thập dữ liệu từ chúng để lưu trữ, phân tích hoặc hiển thị. Hiện nay, các thiết bị đo và thiết bị thu thập dữ liệu có thể được tích hợp vào máy tính thông qua các kết nối truyền thông tốc độ cao để có được sự kết hợp chặt chẽ hơn giữa sức mạnh và tính linh hoạt của máy tính với các thiết bị này.

Do các thiết bị thu thập dữ liệu chỉ làm việc với tín hiệu điện, nên phải có một bộ chuyển đổi hoặc cảm biến để chuyển các hiện tượng vật lý về dạng tín hiệu điện. Cặp nhiệt điện là một bộ chuyển đổi thường gặp. Nó sử dụng các đặc tính vật chất của các kim loại khác nhau để chuyển đổi nhiệt độ sang điện áp. Khi nhiệt độ tăng, điện áp do cặp nhiệt điện sinh ra cũng tăng lên. Sau đó, một chương trình phần mềm có thể chuyển đổi điện áp ngược lại sang nhiệt độ để phân tích, hiển thị và ghi chép dữ liệu. Nhiều cảm biến sinh ra dòng điện thay vì điện áp. Dòng điện thường

có nhiều ưu điểm hơn vì tín hiệu sẽ không bị gián đoạn gây ra bởi một lượng nhỏ điện trở trong dây dẫn nối bộ chuyển đổi với thiết bị thu thập dữ liệu. Mặc dù vậy, nhược điểm của các bộ chuyển đổi có đầu ra dạng tín hiệu dòng điện là hầu hết các thiết bị thu thập dữ liệu đều đo điện áp chứ không đo dòng điện. Nhìn chung, các thiết bị thu thập dữ liệu muốn đo được dòng điện phải sử dụng một giá trị điện trở rất nhỏ đã biết để chuyển đổi dòng điện cho trước sang điện áp có thể đọc được. Cuối cùng, thiết bị này vẫn đòi hỏi phải có một điện áp.



HÌNH 44.1

Các tín hiệu tương tự phục vụ quá trình thu thập dữ liệu có thể được phân loại thành hai nhóm cơ bản: ngẫu nhiên và tiền định. Các thiết bị thu thập dữ liệu có thể vừa thu nhận và vừa sinh ra các tín hiệu dạng này. Các tín hiệu ngẫu nhiên không bao giờ lặp lại và có phổ tần số phẳng. Nhiễu tĩnh điện của micrô là một ví dụ về tín hiệu ngẫu nhiên. Không giống như tín hiệu ngẫu nhiên, một tín hiệu tiền định có thể được biểu diễn bằng tổng các đường hình sin. Các tín hiệu tiền định có thể được chia nhỏ thành các tín hiệu tuần hoàn và tạm thời. Các tín hiệu tuần hoàn liên tục lặp lại hình dạng ban đầu sau những khoảng thời gian bằng nhau, trong khi tín hiệu tạm thời bắt đầu và kết thúc ở một trạng thái không đổi và không xuất hiện sau những khoảng thời gian bằng nhau. Tín hiệu tạm thời là dạng không tuần hoàn thể hiện một phản ứng tức thời đối với một số kích thích.

Các đầu vào/ra số thông thường được gắn với phần cứng thu thập dữ liệu nhằm kiểm tra các tiếp điểm, điều khiển các rơle và kiểm tra các thiết bị số. Các mức tín hiệu số được sử dụng phổ biến nhất là TTL và CMOS tương thích TTL. Cả hai đều được xây dựng dựa trên chuẩn 5V rất phổ biến cho các phần cứng số. Có nhiều kiểu truyền tín hiệu số đi và đến phần cứng thu thập dữ liệu, từ truyền tín hiệu tĩnh (ở đây được hiểu là phương pháp truyền không cần các tín hiệu điều khiển và tín hiệu trạng thái) cho đến truyền tốc độ cao. Các tín hiệu vào/ra số dạng tĩnh (hay còn gọi là tín hiệu tức thời) thường được sử dụng cho việc thiết lập đường truyền, và giám sát trạng thái thông qua các lệnh phần mềm. Tốc độ tối đa của một tín hiệu vào/ra số dạng tĩnh phụ thuộc chủ yếu vào phần cứng máy tính, hệ điều hành và các chương trình ứng dụng. Tốc độ truyền và nhận tín hiệu phụ thuộc vào lượng dữ liệu, phần cứng thu thập dữ liệu và tốc độ máy tính.

Dạng đầu vào/ra cuối cùng của các phần cứng thu thập dữ liệu dựa trên máy tính là đầu vào/ra của bộ đếm/bộ định thời. Bộ đếm/bộ định thời có khả năng đo hoặc tạo ra các xung nhịp. Giống như tín hiệu vào/ra số, những xung này đa phần là TTL hoặc CMOS tương thích với TTL. Bộ đếm/bộ định thời thường được dùng để đo và tạo ra một loạt các tín hiệu bao gồm đếm sự kiện, tạo chuỗi xung, điều biến dịch tần số và giám sát các bộ encoder. Hai đặc trưng cơ bản của một bộ đếm/bộ định thời là kích cỡ bộ đếm và tần số nguồn tối đa. Kích cỡ bộ đếm thường được biểu thị dưới dạng bit và nó xác định giá trị lớn nhất mà bộ đếm có thể đếm được. Ví dụ, một bộ đếm 32 bit có thể đếm được $2^{32} - 1 = 4{,}294{,}967{,}295$ sự kiện trước khi nó chuyển giá trị đếm về 0. Tần số nguồn tối đa biểu thị tốc độ nhanh nhất của tín hiệu mà bộ đếm có thể đếm được. Một bộ đếm 80-MHz có thể đếm các sự kiện cách nhau 12,5 ns. Thực chất "sự kiện" là một cạnh lên hoặc một cạnh xuống của tín hiệu số.

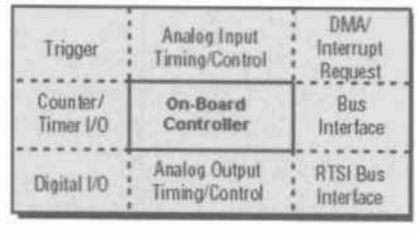
Trong thực tế không có các tín hiệu hoàn toàn không có nhiễu. Xử lý tín hiệu là phương pháp loại bỏ càng nhiều càng tốt các thành phần không mong muốn của một tín hiệu số hoặc tương tự.

Một tín hiệu tương tự thường bao gồm cả tín hiệu tiền định và tín hiệu ngẫu nhiên. Trong khi đó tín hiệu số thì không hoàn toàn vuông. Phần cứng đo đạc, đặc biệt được dùng cho các tín hiệu tương tự tần số cao, thường được trang bị thêm một bộ lọc khử răng cưa. Đây là bộ lọc thông thấp có chức năng loại bỏ các tần số lớn hơn dải tần số mong muốn và làm tăng độ chính xác của các phép đo. Đường truyền số và đường truyền của bộ đếm/bộ định thời thường đi kèm với những bộ lọc có khả năng loại bỏ các xung nhọn khỏi tín hiệu, nếu không những xung này rất dễ bị đếm nhầm như một cạnh lên hoặc cạnh xuống. Cách ly là một phương pháp xử lý tín hiệu khác. Nó tách rời mạch điện phần cứng của thiết bị đo khỏi tín hiệu cần đo nhằm loại bỏ sự chênh lệch điện thế giữa phần cứng của thiết bị đo và tín hiệu, cũng như bảo vệ phần cứng của thiết bị đo trong trường hợp điện áp hoặc dòng điện tăng vọt.

Phần trung tâm của một thiết bị thu thập dữ liệu là bộ chuyển đổi số-tương tự (DAC) và một bộ chuyển đổi tương tự-số (ADC) hoặc sự kết hợp của cả hai. Bộ ADC có một tập hợp hữu hạn các giá trị biểu diễn điện áp. Chức năng của bộ ADC là lựa chọn một giá trị gần với điện áp thực nhất trong tập hợp này tại một thời điểm nhất định. Giá trị này sau đó được chuyển tới máy tính dưới dạng nhị phân. Ngược lại, một bộ DAC có thể tạo ra một điện áp tương tự từ một tập hợp các giá trị nhị phân. Điện áp do bộ DAC cơ bản tạo ra không thay đổi cho tới khi nó nhận được một giá trị khác từ máy tính. Để thu nhận và tạo ra các các dạng tín hiệu tương tự, bộ DAC và ADC phải làm việc ở những khoảng thời gian chính xác. Do đó, phần cứng của thiết bị đo có mạch định thời nhằm tạo ra một chuỗi xung có tần số không đổi để điều khiển bộ ADC và DAC.

Máy tính trao đổi dữ liệu với ADC và DAC thông qua bus. Bus là một tập hợp các dây dẫn điện có chức năng truyền thông tin bên trong máy tính. Một số ví dụ điển hình về bus là PCI và USB. Bus có thể truyền cả thông tin điều khiển lẫn dữ liệu đo lường nhị phân đến và đi từ phần cứng của thiết bị đo. Một trong những vấn đề quan trọng nhất khi lựa chọn bus là tốc độ truyền của bus, thường được biểu diễn bằng Mbyte/s. Một giá trị tương tự có thể chỉ cần dưới 1 byte hoặc tối đa là 4 byte, phụ thuộc vào từng loại phần cứng của thiết bị đo. Bus được dùng chung bởi nhiều thiết bị, do đó các thiết bị thu thập dữ liệu thường có bộ nhớ gắn trong với vai trò là nơi lưu trữ dữ liệu khi không có bus. Với những quá trình thu thập dữ liệu ở tốc độ cao, bộ nhớ có thể lưu tất cả các dữ liệu và đến cuối quá trình thu thập dữ liệu tất cả những dữ liệu này sẽ được chuyển sang máy tính để xử lý.

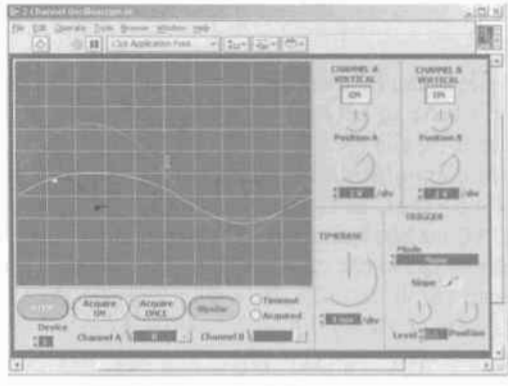
Khi dữ liệu được thu nhận với tốc độ cao trên nhiều kênh, điều quan trọng là phải nắm bắt được quan hệ pha giữa các tín hiệu. Nếu các tín hiệu được thu nhận và tạo ra từ nhiều thiết bị thu thập dữ liệu, sẽ có rất nhiều cách đồng bộ hoá các hệ thống và duy trì các quan hệ pha. Một trong các cách đó là chia sẻ xung nhịp của ADC và DAC giữa các thiết bị thu thập dữ liệu. Bus tích hợp hệ thống thời gian thực (RTSI - Realtime System Integration) là một bus liên kết nhiều thiết bị với nhau để chia sẻ mạch định thời giữa các thiết bị này. Phương pháp đồng bộ hóa phức tạp hơn là vòng khóa pha (PPL - Phase Lock Looping). Một tín hiệu tham chiếu được truyền đến tất cả các thiết bị thu thập dữ liệu và các xung nhịp của chúng sẽ cùng pha với tín hiệu tham chiếu. Do đó, quan hệ pha có thể được duy trì, ngay cả khi các phần cứng đo có tốc độ lấy mẫu và cập nhật thông tin khác nhau. Hình 44.2 giới thiệu các thành phần của một phần cứng thu thập dữ liệu điển hình.



HÌNH 44.2

Sai lệch giữa giá trị thực của một điện áp tương tự với một điện áp gần nhất trong tập hợp các giá trị nhị phân được gọi là sai số lượng tử hoá. Trong một hệ thống đo lường số hoàn toàn không

có nhiễu, sai số lượng tử hoá biểu thị sự chênh lệch giữa điện áp thực tế và điện áp đo được. Tuy nhiên, trên thực tế không có một phần cứng đo lường cũng như môi trường nào là hoàn hảo. Độ chính xác của một thiết bị thể hiện lượng bất định khi xem xét sai số lượng tử, nhiễu hệ thống không tránh khỏi và sự không hoàn chỉnh của phần cứng. Một thông số khác cũng rất quan trọng khi đánh giá thiết bị đo là khả năng lặp lại kết quả của phép đo với một nguồn tín hiệu không đổi. Thậm chí nếu một thiết bị được coi là chính xác thì cũng có thể cho kết quả sai nếu kết quả của phép đo vẫn chênh lệch so với giá trị thực của tín hiệu.



HÌNH 44.3

Độ chính xác của một hệ thống thu thập dữ liệu có thể thay đổi theo nhiệt độ, thời gian và cách sử dụng. Phần cứng thu thập dữ liệu có thể lưu các hằng số hiệu chỉnh trên bo mạch cho các sai số dịch và sai số khuếch đại. Sai số dịch là sự chênh lệch không đổi giữa điện áp thực và điện áp đo được mà không quan tâm đến mức điện áp. Sai số khuếch đại tăng lên một cách tuyến tính khi điện áp cần đo tăng. Một số phần cứng thu thập dữ liệu có một nguồn điện áp chuẩn trên bo mạch được dùng định kỳ như một tham chiếu để hiệu chỉnh các sai số khuếch đại và sai số dịch.

Vấn đề cuối cùng cần phải hiểu rõ trong hệ thống thu thập dữ liệu là phần mềm. Phần mềm driver là một tập hợp lệnh được người lập trình đưa vào một chương trình. Phần mềm driver thường do nhà sản xuất phần cứng cung cấp và có thể được sử dụng trong nhiều ngôn ngữ lập trình. Nhà lập trình có thể sử dụng một ngôn ngữ lập trình để xây dựng một ứng dụng dựa trên phần mềm driver như mô tả trong hình 44.3. Sau đó, người dùng có thể dễ dàng điều khiển và thu nhận dữ liệu từ phần cứng. Đây được coi là một thiết bị được sản xuất theo nhu cầu của người sử dụng.

45

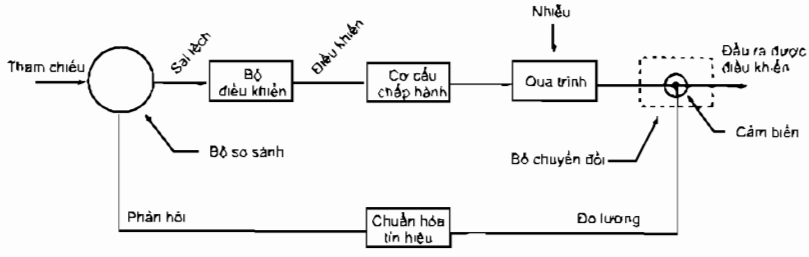
Kỹ thuật đo: Cảm biến và bộ chuyển đổi

45.1 Giới thiệu 45-1
45.2 Bộ chuyển đổi lực và chuyển động 45-2
45.3 Bộ chuyển đổi quá trình 45-9
45.4 Hoạt động của bộ chuyển đổi 45-20
45.5 Sự phù hợp giữa tải và bộ chuyển đổi 45-20

Cecil Harrison
University of Southern Mississippi

45.1 Giới thiệu

Các thành phần **hướng thuận** của một hệ thống điều khiển tự động (*bộ so sánh, bộ điều khiển, cơ cấu chấp hành, thiết bị hoặc quá trình*) thường phản ứng lại tín hiệu sai số (hình 45.1). Tín hiệu sai số được xác định bằng cách so sánh giá trị **đầu ra điều khiển** với một số **đầu vào tham chiếu**, vì vậy độ chính xác và khả năng lặp lại của đầu ra điều khiển phụ thuộc phần lớn vào độ chính xác và khả năng lặp lại của phép đo đầu ra điều khiển. Kết quả đo của đầu ra điều khiển được đưa qua một bộ phận của hệ thống gọi là **bộ chuyển đổi**, đây được coi là chức năng quan trọng nhất trong một hệ thống điều khiển tự động.



HÌNH 45.1 Sơ đồ chức năng của hệ điều khiển tự động tiêu chuẩn

Bộ chuyển đổi có chức năng đo biên độ hoặc cường độ của đầu ra điều khiển và sinh ra một tín hiệu tương ứng dưới dạng năng lượng thích hợp với việc truyền dẫn theo đường phản hồi đến bộ so sánh. Thuật ngữ "tương ứng" được sử dụng ở đây mang tính chất tương đối bởi vì đầu ra của bộ chuyển đổi không phải lúc nào cũng tương ứng với đầu ra điều khiển, nghĩa là bộ chuyển đổi có thể không phải là một thành phần tuyến tính. Trong các hệ tuyến tính, nếu đầu ra của bộ chuyển đổi không tuyến tính thì nó sẽ được tuyến tính hóa bằng bộ chuẩn hóa tín hiệu. Phần tử của bộ chuyển đổi đo đầu ra điều khiển được gọi là cảm biến, các phần tử còn lại của bộ chuyển đổi có

chức năng biến đổi đầu ra của cảm biến sang dạng năng lượng cần cho **đường phản hồi**. Các dạng có thể của đường phản hồi bao gồm:

- Liên kết cơ học
- Lưu chất (Chất khí hoặc Chất lỏng)
- Điện (bao gồm nối quang, truyền RF, nối từ, hoặc truyền âm)

Những tín hiệu điện phù hợp để mô tả kết quả đo bao gồm:

- Điện áp hoặc dòng điện DC
- Điện áp hoặc dòng điện AC, tần số hoặc pha (điều biến CW)
- Xung áp hoặc xung dòng (tín hiệu số)

Trong một số trường hợp, tín hiệu phản hồi có thể thay đổi (ví dụ từ biên độ DC sang dạng xung số).

Phần tiếp theo sẽ trình bày về các hệ thống điều khiển tự động trong đó tín hiệu phản hồi là tín hiệu điện và đường phản hồi bao gồm các kết nối dây dẫn hoặc dây cáp giữa các bộ phận của đường phản hồi. Những bộ chuyển đổi được xem xét dưới đây đo đầu ra điều khiển và tạo ra tín hiệu điện mô tả biên độ, cường độ hoặc hướng của đầu ra điều khiển.

Bộ chuẩn hóa tín hiệu nhận đầu ra của bộ chuyển đổi và truyền tín hiệu đó đến bộ so sánh dưới dạng tương thích với đầu vào tham chiếu. Chức năng của bộ chuẩn hóa tín hiệu bao gồm:

- Khuếch đại/suy giảm (tỷ lệ)
- Cách ly
- Lấy mẫu
- Khử nhiễu
- Tuyến tính hóa
- Dịch nhịp và tham chiếu
- Thao tác toán học (ví dụ như vi phân, chia, tích phân, nhân, tìm nghiệm, bình phương, trừ, hoặc cộng)
- Chuyển đổi tín hiệu (ví dụ, DC–AC, AC–DC, tần số–điện áp, điện áp–tần số, số–tương tự, tương tự–số, v.v...)
- Bộ đệm
- Số hóa
- Lọc
- Đồng bộ trở kháng
- Tạo dạng sóng
- Dịch pha

Trong trường hợp các tín hiệu cần chuẩn hóa đã được thực hiện trong bộ chuyển đổi thì đầu ra của bộ chuyển đổi có thể được nối trực tiếp với bộ so sánh. Nối trực tiếp đầu ra của bộ chuyển đổi đến bộ so sánh khác với phản hồi đơn vị. Phản hồi đơn vị xảy ra khi các thành phần của đường phản hồi (bộ chuyển đổi và bộ xử lý tín hiệu) có hàm truyền tổng bằng 1. Trong hệ thống điều khiển số, nhiều chức năng của bộ chuẩn hóa tín hiệu có thể được thực thiện bằng phần mềm.

Bộ chuyển đổi thường được chia thành hai nhóm:

- **Bộ chuyển đổi chuyển động và lực**, loại này chủ yếu kết hợp với **cơ cấu servo**
- **Bộ chuyển đổi quá trình**, loại này chủ yếu kết hợp với hệ **điều khiển quá trình**

45.2 Bộ chuyển đổi lực và chuyển động

Phần này đề cập đến các bộ chuyển đổi được sử dụng trong hệ thống điều khiển chuyển động (ví dụ: *dịch chuyển, vận tốc và gia tốc*). Lực liên quan mật thiết với chuyển động, bởi vì

chuyển động là kết quả của các lực mất cân bằng, và do vậy các bộ chuyển đổi lực cũng được trình bày trong phần này. Tuy nhiên trong phần này chúng ta chỉ đề cập đến những bộ chuyển đổi đo chuyển động thẳng (chuyển động theo đường thẳng trong một hệ quy chiếu cố định) hoặc chuyển động góc (chuyển động tròn quanh một trục cố định). Chuyển động thẳng đôi khi được gọi là chuyển động *tuyến tính*, nhưng điều này dẫn đến sự nhầm lẫn với những trường hợp chuyển động theo một đường thẳng nhưng có đáp ứng phi tuyến về mặt toán học đối với các lực đầu vào. Chuyển động góc cũng được gọi là *chuyển động quay tròn* hoặc chuyển *động xoay tròn*.

Cơ sở lý thuyết chính của bộ chuyển đổi chuyển động nằm trong lĩnh vực cơ học vật rắn. Từ phương trình chuyển động của vật rắn (bảng 45.1), rõ ràng là nếu đo bất kỳ một trong các thông số: dịch chuyển, vận tốc hoặc gia tốc thì có thể xác định hai thông số còn lại.

Vị trí đơn giản là một tọa độ trong một hệ qui chiếu, vì vậy bất kỳ phép đo dịch chuyển nào liên quan đến hệ qui chiếu này đều là phép đo vị trí, và bất kỳ bộ chuyển đổi dịch chuyển nào có đầu vào được tham chiếu tới hệ qui chiếu này đều có thể được dùng như một bộ chuyển đổi vị trí.

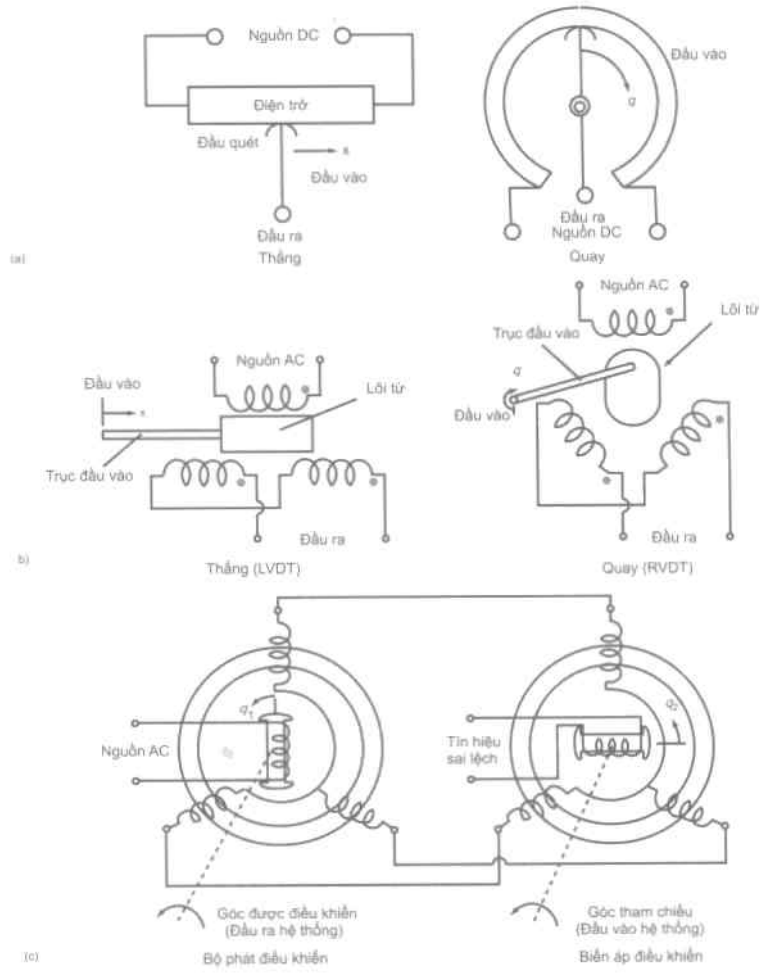
BẢNG 45.1 Các phương trình chuyển động

Liên tục	Rời rạc $\Delta t = t_i - t_{i-1}$
Chuyển động thẳng:	
$x(t) = \int v(t)dt = \iint a(t)dt$	$x_i = x_{i-1} + \frac{v_i + v_{i-1}}{2} \cdot (\Delta t) = 2x_{i-1} - x_{i-2} + \frac{a_i + 2a_{i-1} + a_{i-1}}{2} \cdot \frac{(\Delta t)}{2}$
Chuyển động góc:	
$\theta(t) = \int \omega(t)dt = \iint \alpha(t)dt$	$\theta_i = \theta_{i-1} + \frac{\omega_i + \omega_{i-1}}{2} \cdot (\Delta t) = 2\theta_{i-1} - \theta_{i-2} + \frac{\alpha_i + 2\alpha_{i-1} + \alpha_{i-2}}{2} \cdot \frac{(\Delta t)}{2}$
Vận tốc thẳng:	
$v(t) = \frac{d}{dt}x(t) = \int a(t)dt$	$v_i = \frac{x_i - x_{i-1}}{\Delta t} = v_{i-1} + \frac{a_i + a_{i-1}}{2} \cdot (\Delta t)$
Vận tốc góc:	
$\omega(t) = \frac{d}{dt}\theta(t) = \int \alpha(t)dt$	$\omega_i = \frac{\theta_i - \theta_{i-1}}{\Delta t} = \omega_{i-1} + \frac{\alpha_i + \alpha_{i-1}}{2} \cdot (\Delta t)$
Gia tốc thẳng:	
$a(t) = \frac{d}{dt}v(t) = \frac{d^2}{dt^2}x(t)$	$a_i = \frac{v_i - v_{i-1}}{\Delta t} = \frac{x_i - 2x_{i-1} + x_{i-2}}{(\Delta t)^2}$
Gia tốc góc:	
$\alpha(t) = \frac{d}{dt}\omega(t) = \frac{d^2}{dt^2}\theta(t)$	$\alpha_i = \frac{\omega_i - \omega_{i-1}}{\Delta t} = \frac{\theta_i - 2\theta_{i-1} + \theta_{i-2}}{(\Delta t)^2}$

Bộ chuyển đổi dịch chuyển (vị trí)

Các bộ chuyển đổi dịch chuyển có thể phân loại theo tính năng như bộ chuyển đổi dịch chuyển lớn hoặc bộ chuyển đổi nhỏ. Ranh giới giữa dịch chuyển lớn và nhỏ đôi khi là tùy ý, nhưng để thuận tiện có thể lấy xấp xỉ 1 mm đối với chuyển động thẳng và xấp xỉ 1 cung 10′ (1/6°) đối với chuyển động góc. Các dạng điển hình của bộ chuyển đổi dịch chuyển lớn (hình 45.2) là:

- Bộ chia áp [hình 45.2(a)]
- VDT (Variable Differential Transformer) [hình 45.2(b)]
- Bộ đồng tốc [hình 45.2(c)]
- Resolver [hình 45.2(d)]
- Bộ mã hóa vị trí [hình 45.2(e)]



HÌNH 45.2 Bộ chuyển đổi vị trí: (a) Bộ chia áp, (b) VDT, (c) Bộ đồng tốc, (d) Resolver, (e) Encoder tuyệt đối, (f) Rãnh mã hóa cho encoder tương đối

Bộ chia áp rất dễ sử dụng và việc xử lý tín hiệu là đơn giản nhất nhưng bộ chia áp dễ bị hao mòn do có tiếp xúc trượt giữa đầu đo với phần tử điện trở và có thể sinh ra nhiễu do sự dao động của đầu đo. Hiện nay bộ chia áp có hành trình biến đổi từ nhỏ hơn 1cm đến lớn hơn 50 cm (đường thẳng) hoặc từ vài độ đến lớn hơn 50 vòng (quay).

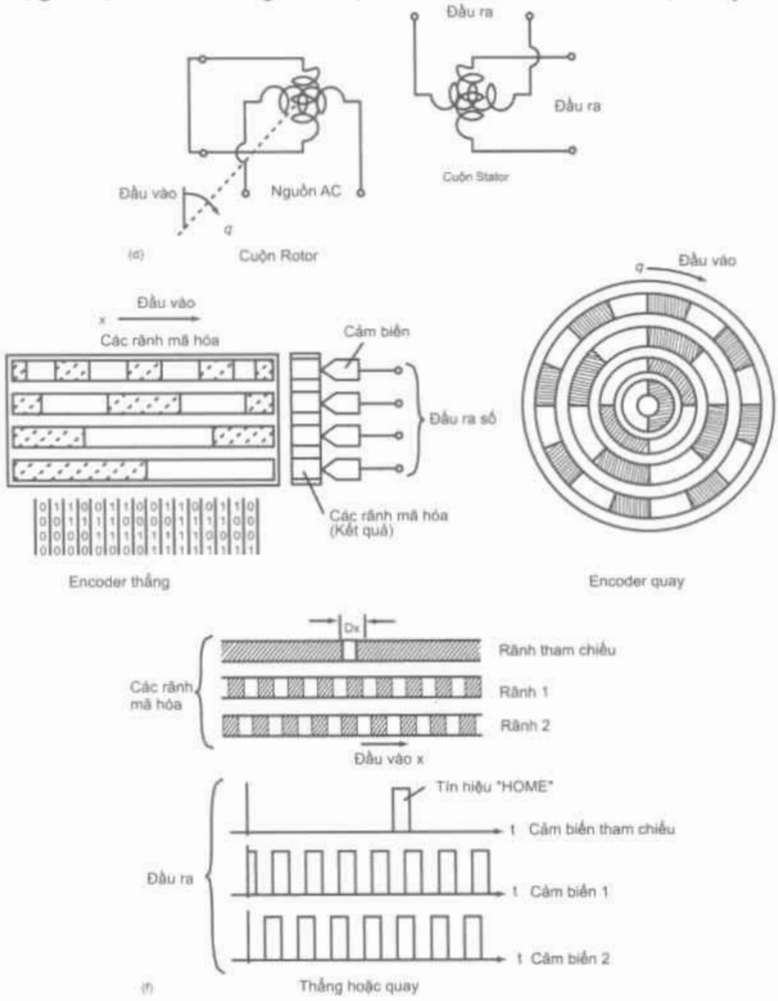
VDT không bị hao mòn như bộ chia áp nhưng độ dài tối đa của hành trình rất nhỏ, xấp xỉ 25cm hoặc thấp hơn đối với VDT tuyến tính (LVDT) và xấp xỉ 60° hoặc nhỏ hơn đối với VDT quay (RVDT). Việc xử lý tín hiệu dưới dạng giải điều chế nhạy pha tín hiệu xoay chiều của VDT rất phức tạp; tuy nhiên, tính sẵn có của bộ giải điều chế VDT chuyên dụng trong các mạch tích hợp (IC) đã góp phần làm giảm bớt những nhược điểm của VDT.

Bộ đồng tốc là thiết bị xoay chiều ba pha khá phức tạp và đắt đỏ, chúng được sản xuất với độ chính xác cao và bền. Bộ đồng tốc có khả năng đo sự sai khác về góc (có thể lên tới ±180°) giữa các

vị trí của hai trục quay liên tục. Thêm vào đó, bộ đồng tốc có thể thực hiện đồng thời nhiều chức năng như: đầu vào tham chiếu, thiết bị đo đầu ra, đường phản hồi và bộ so sánh [hình 45.2 (c)].

Resolver đơn giản hơn và rẻ hơn bộ đồng tốc, và chúng có ưu điểm hơn RVDT nhờ khả năng đo chuyển động góc trong phạm vi 360°. Hình 45.2(d) minh họa một trong nhiều khả năng ứng dụng của Resolver, biên độ tín hiệu tỷ lệ với cosin của góc đo được tại một cuộn dây đầu ra và sin của góc đo được tại đầu còn lại. Các IC chuyên dụng được dùng để xử lý tín hiệu và biến đổi đầu ra của Resolver sang dạng số. Các IC đó khi kết hợp với *bộ chuyển đổi Scott-T*, có thể được dùng để biến đổi đầu ra của bộ đồng tốc sang dạng số.

Bộ mã hóa vị trí có khả năng tương thích cao với các ứng dụng điều khiển số do chúng loại bỏ yêu cầu biến đổi số-tương tự của tín hiệu phản hồi. Các rãnh mã hóa được đọc bởi các cảm biến, thông thường là đầu đọc hoặc các thiết bị quang điện (ví dụ như hồng ngoại hoặc laze). Bộ mã hóa vị trí được dùng cho cả ứng dụng chuyển động thẳng và chuyển động quay, tuy nhiên chúng thường được dùng như bộ mã hóa trục trong các ứng dụng quay. Việc xử lý tín hiệu đối với bộ mã hóa tuyệt đối (Absolute encoder) rất đơn giản [hình 45.2(e)] do chỉ yêu cầu một bộ giải mã, nhưng độ phân giải vị trí phụ thuộc vào số lượng rãnh, và số lượng rãnh càng nhiều độ phức tạp của bộ giải mã càng tăng. Bộ mã hóa tương đối (Incremetal encoder) xử tín hiệu phức tạp hơn, dưới dạng bộ đếm và bộ xử lý để tính toán vị trí. Số lượng rãnh thường cố định là 3 [hình 45.2(f)]. Độ phân giải vị trí chỉ bị giới hạn bởi khả năng chia mịn của rãnh mã hóa trên bề mặt chuyển động.

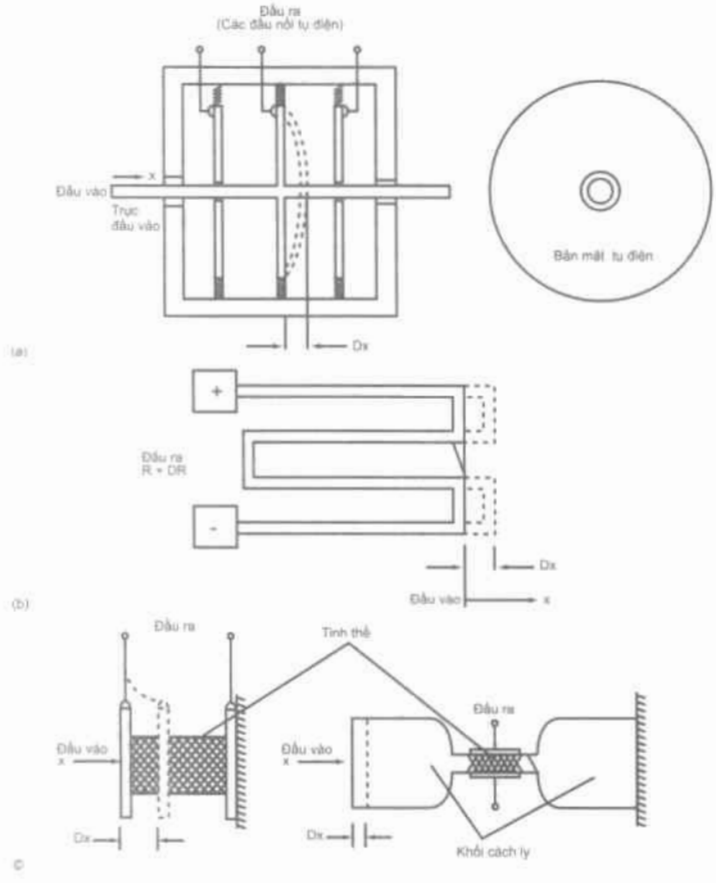


Hình 45.2 (tiếp theo)

Mặc dù bộ chuyển đổi chuyển động lớn được thiết kế riêng cho chuyển động thẳng hoặc chuyển động quay, nhưng cơ cấu *thanh răng* và *bánh răng* hoặc một bộ biến đổi chuyển động tương tự thường được dùng để điều chỉnh các bộ chuyển đổi thiết kế cho chuyển động thẳng để đo chuyển động quay và ngược lại.

Các dạng điển hình của bộ chuyển đổi chuyển động nhỏ [hình 45.3] là:

- Tụ vi phân
- Bộ điện trở biến dạng kế
- Tinh thể áp điện

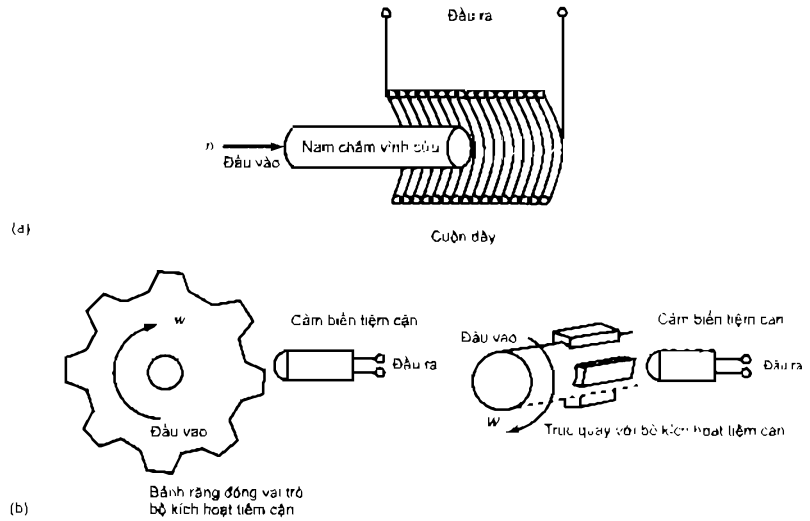


HÌNH 45.3 Bộ chuyển đổi chuyển động nhỏ: (a) Tụ vi phân, (b) điện trở biến dạng kế, (c) tinh thể áp điện

Hình 45.3 (a) mô tả một cách đơn giản nhất ứng dụng của tụ vi phân dùng cho các đo lường chuyển động nhỏ. Chuyển động của thanh đầu vào sẽ uốn cong bản mặt chung (hay còn gọi là bản mặt ở giữa), do đó làm tăng điện dung của một tụ điện và làm giảm điện dung của tụ điện còn lại. Trong một kỹ thuật đo, hai tụ điện là một bộ phận của cầu đo trở kháng (ví dụ như cầu Schering), và sự thay đổi đầu ra của cầu biểu thị sự chuyển động của bản mặt chung. Trong một kỹ thuật đo khác, mỗi tụ điện được kết nối để hoạt động như một tụ điều chỉnh cho bộ tạo dao động và sự chênh lệch tần số giữa hai bộ tạo dao động biểu thị chuyển động.

Điện trở biến dạng kế được dùng để đo biến dạng đàn hồi (sức căng) của vật liệu bằng cách gắn điện trở lên vật liệu [hình 45.3(b)] sao cho nó cùng chịu sức căng như vật liệu. Điện trở thường được gắn vào một trong các nhánh mạch cầu và đầu ra của mạch cầu được lấy là biểu thị của sức căng.

Hiệu ứng áp điện được ứng dụng trong nhiều kỹ thuật đo lường dịch chuyển nhỏ [hình 45.3(c)]. Trong một kỹ thuật, chuyển động đầu vào làm biến dạng tinh thể bằng cách tác động trực tiếp lên một điện cực. Ở một kỹ thuật khác, tinh thể được gia công như một phần của một cấu trúc lớn hơn, được định hướng sao cho chuyển động đầu vào làm cong cấu trúc và làm biến dạng tinh thể. Biến dạng của tinh thể tạo ra điện áp đầu ra và cũng làm thay đổi tần số cộng hưởng của tinh thể. Trong một vài trường hợp, điện áp đầu ra được lấy trực tiếp làm chỉ số của dịch chuyển, tuy nhiên tinh thể thường được dùng để điều khiển bộ tạo dao động và tần số của bộ tạo dao động biểu thị sức căng.



HÌNH 45.4 Bộ chuyển đổi vận tốc: (a) Nam châm và cuộn dây. (b) cảm biến tiếp cận

Bộ chuyển đổi vận tốc

Như đã được trình bày ở trên, kỹ thuật chuẩn hóa tín hiệu cho ta biết tất cả các thông số về dịch chuyển, vận tốc, hoặc gia tốc từ phép đo bất kỳ một trong ba thông số trên. Tuy nhiên, đôi khi đo trực tiếp vận tốc sẽ thuận lợi hơn, đặc biệt trong trường hợp chuyển động thẳng có hành trình ngắn hoặc trục quay có vận tốc cao. Các bộ chuyển đổi tương tự thường được dùng để đáp ứng hai yêu cầu này là:

- Bộ chuyển đổi vận tốc nam châm-cuộn dây [hình 45.4(a)]
- Máy đo tốc độ

Loại thứ ba của bộ chuyển đổi vận tốc là *bộ chuyển đổi vận tốc kiểu bộ đếm* [hình 45.4(b)]. Loại này dễ sử dụng và tương thích với các bộ điều khiển số.

Hoạt động của bộ chuyển đổi vận tốc nam châm-cuộn dây dựa trên định luật cảm ứng của Faraday. Đối với cuộn dây sôlênôit có tỷ lệ độ dài/đường kính cao, điện áp cảm ứng trong cuộn dây tỷ lệ với vận tốc của nam châm. Bộ chuyển đổi vận tốc nam châm-cuộn dây có hành trình nằm trong khoảng từ nhỏ hơn 10mm tới xấp xỉ 0,5m.

Máy đo vận tốc là một máy phát AC hoặc DC nhỏ mà điện áp đầu ra của nó tỷ lệ thuận với vận tốc góc của roto. Máy đo vận tốc góc có thể có tốc độ trục lên tới 5000 vòng/phút hoặc lớn hơn nhưng đầu ra có thể là phi tuyến và điện áp đầu ra nhấp nhô không thể sử dụng được ở tốc độ thấp.

Các máy đo vận tốc góc AC có giá rẻ hơn và dễ bảo dưỡng hơn các máy đo vận tốc góc DC, nhưng máy đo vận tốc góc DC lại có khả năng tương thích trực tiếp với các bộ điều khiển tương tự và chiều phân cực của đầu ra trực tiếp biểu thị hướng quay. Đầu ra của máy đo vận tốc góc AC phải được giải điều chế (ví dụ như chỉnh lưu và lọc) và bộ giải điều chế phải nhạy cảm với pha để biểu thị hướng quay.

Bộ chuyển đổi vận tốc kiểu bộ đếm hoạt động dựa trên nguyên tắc đếm các xung điện trong một khoảng thời gian cố định, sau đó chuyển đổi số xung đếm được trong một đơn vị thời gian sang vận tốc. Bộ chuyển đổi vận tốc kiểu bộ đếm hoạt động dựa trên các cảm biến tiệm cận hoặc incremental encoder [hình 45.2(f)]. Cảm biến tiệm cận có các dạng sau:

- Quang điện
- Từ trở biến thiên
- Hiệu ứng Hall
- Tự cảm
- Điện dung

Hai ứng dụng phổ biến của bộ chuyển đổi vận tốc kiểu bộ đếm được minh họa trong hình 45.4.

Do bộ điều khiển số đã có đồng hồ điện tử rất chính xác nên việc đếm và biến đổi xung sang vận tốc có thể được thực hiện bằng phần mềm. Tuy nhiên, sử dụng phần cứng để đếm xung có thể cần thiết nếu việc đếm xung ảnh hưởng tới những chức năng điều khiển khác của bộ điều khiển. Một IC đặc biệt, gọi là *giao diện đếm/giải mã cầu phương* (Quadrature Decoder/Counter Interface), có thể thực hiện các chức năng giải mã và đếm, đồng thời truyền kết quả tới bộ điều khiển dưới dạng từ dữ liệu.

Bộ chuyển đổi gia tốc

Giống như đo vận tốc, đôi khi người ta đo gia tốc một cách trực tiếp hơn là xác định gia tốc từ kết quả đo dịch chuyển hoặc vận tốc. Phần lớn bộ chuyển đổi gia tốc được gọi là gia tốc kế dạng *địa chấn* do chúng hoạt động dựa trên việc đo chuyển động của một khối lượng được gọi là *phần tử địa chấn* (hình 45.5). Các cấu hình trong hình 45.5 (a,b) yêu cầu phải có sự bố trí các lò xo khá chính xác cho hệ thống treo và định tâm cho phần tử địa chấn. Một trong những nhược điểm của gia tốc kế dạng địa chấn là phần tử địa chấn bị chuyển động trong quá trình đo gia tốc và gây ra phi tuyến và sai lệch kết quả đo. Cấu hình cân bằng lực trong hình 45.5 (c) sử dụng lõi nam châm điện như là phần tử địa chấn. Một cảm biến chuyển động nhạy phát hiện ra sự chuyển động của lõi và sử dụng tín hiệu chuyển động dưới dạng phản hồi âm để điều khiển cuộn dây đưa lõi trở về vị trí trung tâm của nó. Đầu ra của gia tốc kế lực cân bằng là sự phản hồi cần thiết để cản trở sự chuyển động hơn là nguyên nhân gây ra sự chuyển động.

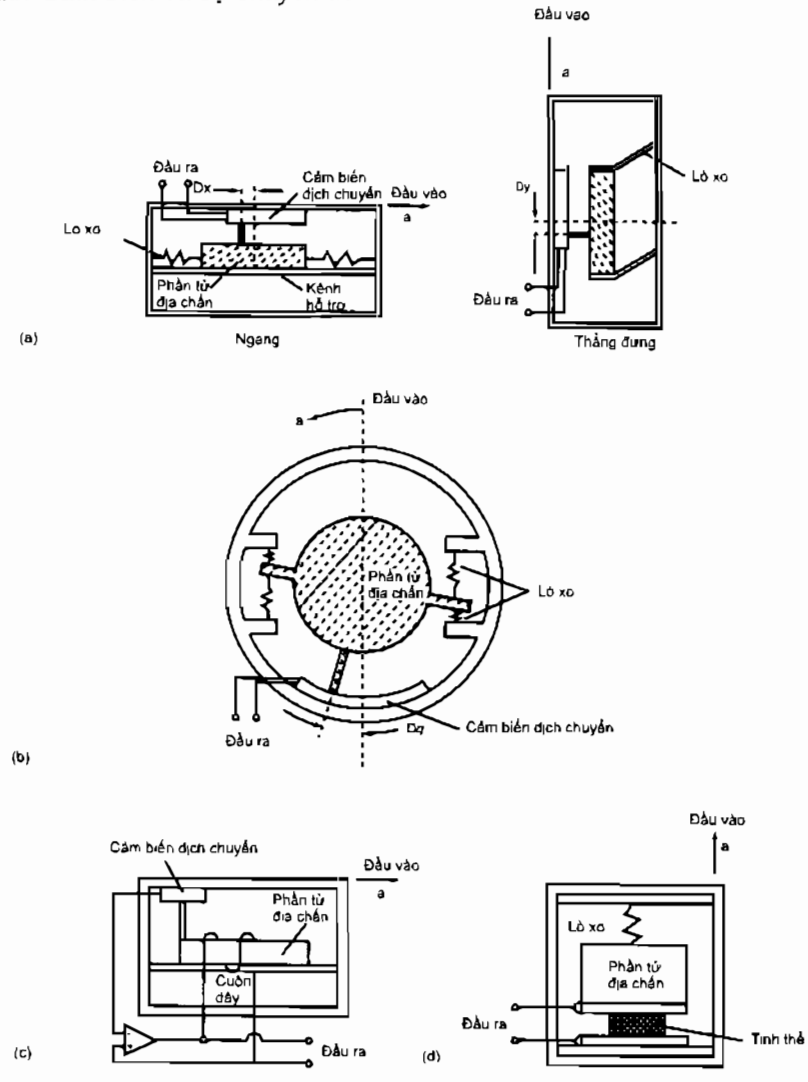
Một dạng gia tốc kế dạng địa chấn đơn giản hơn dùng điện cực của tinh thể áp điện làm phần tử địa chấn [hình 45.5(d)]. Tương tự như vậy, một dạng gia tốc kế đơn giản khác lại coi bản mặt chung của tụ vi phân [hình 45.3(a)] là phần tử địa chấn.

Bộ chuyển đổi lực

Đo lực thông thường được dựa trên đo chuyển động, là kết quả tác dụng của ngoại lực. Nếu ngoại lực gây ra chuyển động của đầu ra điều khiển và khối lượng của phần tử đầu ra đã biết thì bất kỳ gia tốc kế nào được gắn với đầu ra điều khiển đều sinh ra một đầu ra tỷ lệ với ngoại lực ($F=Ma$). Một thang chia cân bằng lò xo đơn giản [hình 45.6(a)] dựa trên đo chuyển động, là kết quả của ngoại lực (trọng lượng) kéo dãn lò xo.

Các phép đo lực có độ chính xác cao trong các cơ cấu servo thường dựa trên con quay hồi chuyển (Gyroscope) đóng vai trò nhận biết ngoại lực tác động. Sơ đồ trình bày trên hình 45.6(b) là một con quay hồi chuyển có khớp các đăng và phần tử quay. Bộ chuyển đổi chuyển động (hoặc dịch chuyển và vận tốc) trên trục quay cung cấp đầu ra tương ứng với ngoại lực tác dụng. Các dạng khác của con quay hồi chuyển và cảm biến trục quay cũng được sử dụng trong kỹ thuật đo lực.

Đo lực tĩnh (chuyển động không rõ ràng) thường dựa trên đo sự căng do ngoại lực tác dụng. Hình 45.6(c) minh họa một cấu trúc phổ biến của bộ chuyển đổi lực gọi là *load cell*. Lực tác dụng tạo ra biến dạng tỷ lệ trong thành phần cấu trúc hình chữ S. Biến dạng này được đo bằng bộ chuyển đổi dịch chuyển nhạy, thường là một điện trở biến dạng kế hoặc tinh thể áp điện.



HÌNH 45.5 Gia tốc kế dạng địa chấn: (a) Bộ chuyển đổi gia tốc thẳng, (b) Gia tốc kế quay, (c) Gia tốc kế cân bằng lực, (d) Gia tốc kế áp điện

45.3 Bộ chuyển đổi quá trình

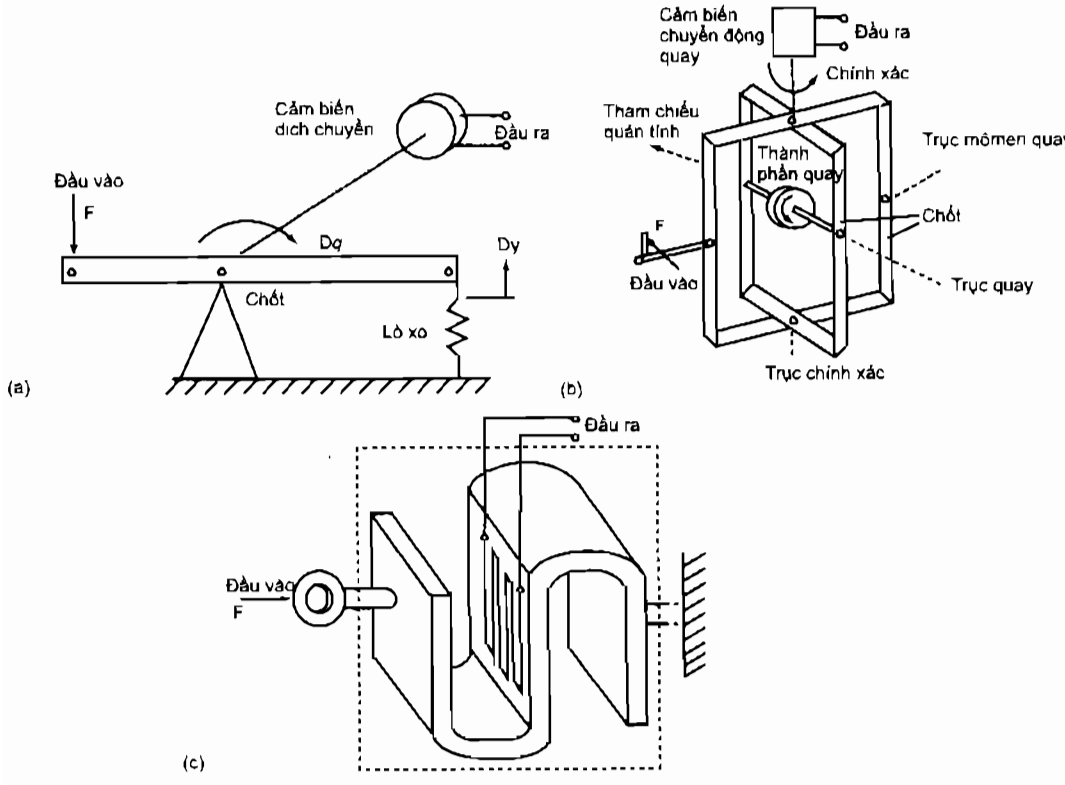
Phần này thảo luận về các bộ chuyển đổi sử dụng trong đo lường và điều khiển các *tham số quá trình* thường gặp trong các công nghiệp, đó là

- Áp suất lưu chất
- Dòng lưu chất
- Độ nhớt
- Nhiệt độ

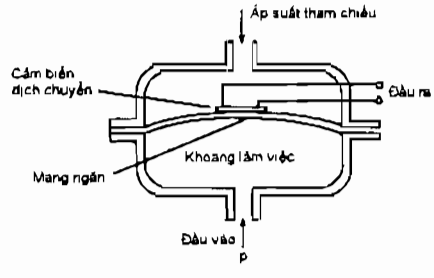
Bộ chuyển đổi áp suất lưu chất

Phần lớn bộ chuyển đổi áp suất lưu chất thuộc dạng *đàn hồi*, trong đó lưu chất được giữ trong một khoang có ít nhất một vách đàn hồi và sự biến dạng của vách đàn hồi biểu thị áp suất. *Ống Bourdon* và các *ống xếp* là những ví dụ về bộ chuyển đổi áp suất dạng đàn hồi được sử dụng trong phòng thí nghiệm và một số ứng dụng điều khiển quá trình công nghiệp. Bộ chuyển đổi áp suất lưu

chất mô tả trong hình 45.7 sử dụng *màng ngăn* đàn hồi để phân tách hai khoang là dạng thường thấy trong điều khiển quá trình công nghiệp. Các màng ngăn được tạo bởi nhiều vật liệu đàn hồi, từ kim loại mỏng đến vải polyme hóa.



HÌNH 45.6 Chuyển đổi vận tốc: (a) Thang đo lò xo, (b) Con quay hồi chuyển, (c) Load cell

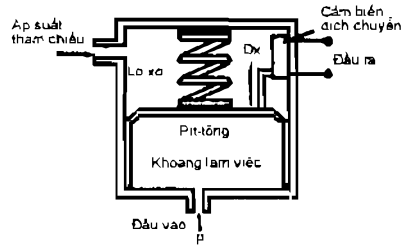


HÌNH 45.7 Bộ chuyển đổi áp suất dạng màng ngăn

Với đo lường áp suất lớn, dịch chuyển của màng ngăn được đo bằng bộ chia áp hoặc LVDT. Với các đo lường áp suất nhạy hơn, có thể dùng bất cứ cảm biến nào trong ba cảm biến dịch chuyển nhạy được mô tả ở trên. Trong hầu hết các dạng thông thường của cảm biến áp suất nhạy, điện trở biến dạng kế hình hoa thị thường được gắn vào màng ngăn. Trong một dạng khác, các vách ngoài của cảm biến áp suất đóng vai trò là bản tụ và màng ngăn đóng vai trò là bản mặt chung của tụ vi phân. Trong một cấu hình rất nhạy và tích hợp cao, màng ngăn là màng silicon với một biến dạng kế áp trở và các mạch chuẩn hóa tín hiệu được tích hợp trong silicon.

Các đo lường có độ chân không cao (áp suất rất thấp) thường dựa trên các quan sát về độ nhớt, tính dẫn nhiệt, thuộc tính âm thanh, hoặc điện thế ion hóa của chất lỏng, sẽ không

được giới thiệu trong phần này. Các bộ chuyển đổi được dùng trong các hệ thủy lực áp suất cao [70 MPa (10,000 psi) hoặc cao hơn] thường có dạng *piston-lò xo* [hình 45.8].



HÌNH 45.8 Bộ chuyển đổi Piston-lò xo

Trong cả hai bộ chuyển đổi áp suất này, đầu ra là phép đo chênh lệch áp suất giữa khoang làm việc và khoang tham chiếu của bộ chuyển đổi (tức là, $p_{OUT} = p - p_{REF}$). Do lường này được gọi là:

- *Áp suất tuyệt đối* nếu khoang tham chiếu được đóng kín và tạo chân không (tức là $p_{REF} = 0$ và $p_{OUT} = p$)
- *Áp suất tiêu chuẩn* nếu khoang tham chiếu được mở thông với không khí (tức là $p_{OUT} = p - p_{ATM}$)
- *Áp suất vi sai* nếu bất kỳ áp suất nào khác được đưa vào khoang tham chiếu

Bộ chuyển đổi dòng lưu chất (Lưu tốc kế)

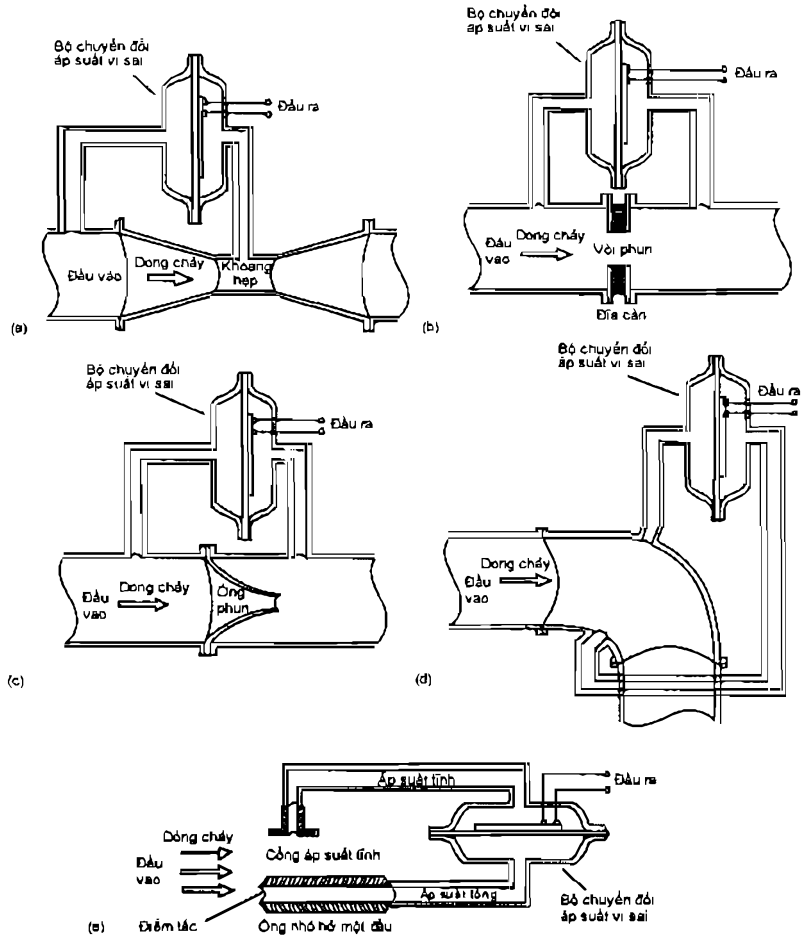
Đo lưu tốc được sử dụng rộng rãi trong nhiều ứng dụng và công nghệ đo. Trong các quá trình công nghiệp, thuật ngữ *lưu chất* không chỉ được áp dụng cho chất khí và chất lỏng mà còn cho các hỗn hợp dạng dòng (thông thường gọi là các *chất sền sệt* hoặc *bùn đặc*) như bê tông, nước thải hoặc bột giấy. Do đó việc điều khiển dòng chất lỏng và các dạng đo lường cần thiết có thể liên quan đến *thể tích*, *khối lượng* và hướng dòng chảy. Các dòng khí có *khả năng nén* cũng tác động đến kỹ thuật đo. Ngoài ra, trạng thái của dòng chảy, dù có đồng nhất và sạch hay không, cũng ảnh hưởng đến công nghệ lưu tốc kế. Một nhân tố khác cần quan tâm là vận tốc dòng chảy. Dòng chất nhớt dạng lớp chuyển động cần có các kỹ thuật đo khác so với kỹ thuật đo dòng chảy rối tốc độ cao. Một vấn đề khác cần lưu ý là sự giam hãm dòng chảy. Trong khi hầu hết các đo lường dòng chất lỏng liên quan đến dòng chảy *đầy đủ* qua các *kênh kín* như là ống, thì một số ứng dụng lại yêu cầu các đo lường *dòng chảy từng phần* qua các *kênh mở* như máng. Ở đây chúng ta chỉ quan tâm đến những lưu tốc kế được sử dụng rộng rãi .

Lưu tốc kế được chia thành các loại chính sau:

- Dạng áp suất vi phân, dạng thắt (venturi, vòi phun, ống phun đo lưu lượng, ống khuỷu và ống pitot tĩnh) (hình 45.9)
- Áp suất dòng chảy (hộp số, tua bin, và guồng quay) (hình 45.10)
- Sóng siêu âm (hình 45.11)
- Thả xoáy (hình 45.12)
- Phong tốc kế nhiệt (hình 45.13)
- Điện từ (hình 45.14)
- Lưu tốc kế kiểu phao (lưu tốc kế có tiết diện thay đổi trên đường đi) (hình 45.15)

Lưu tốc kế áp suất vi phân phù hợp với dòng chảy khí và chất lỏng sạch, độ nhớt thấp có vận tốc cao và trung bình. Lưu tốc kế Venturi [hình 45.9(a)] có độ chính xác cao nhất nhưng chúng có kích thước lớn và đắt. Lưu tốc kế dạng vòi phun [hình 45.9(b)] có kích thước nhỏ hơn, giá rẻ hơn nhưng lại không chính xác bằng lưu tốc kế Venturi. Lưu tốc kế ống phun [hình 45.9(c)] là sự kết

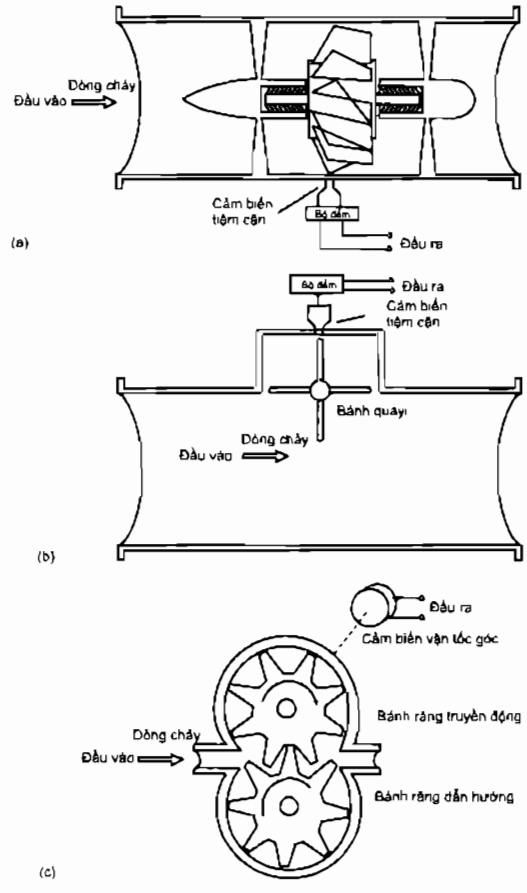
hợp giữa lưu tốc kế Venturi và dạng vòi phun. Lưu tốc kế ống khuỷu [hình 45.9(d)] có thể lắp đặt ở bất kỳ chỗ cong nào trong hệ thống ống dẫn, chủ yếu được sử dụng trong các đo lường lưu lượng thô. Lưu tốc kế pitot tĩnh [hình 45.9(e)] được dùng với những đòng chảy có tiết diện lớn như trong hầm gió. Lưu tốc kế pitot tĩnh cũng được sử dụng trong các ứng dụng dòng tự do như bộ hiển thị vận tốc gió trong máy bay.



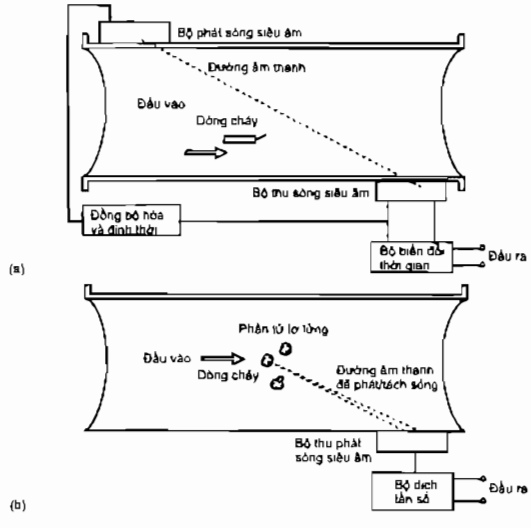
HÌNH 45.9 Lưu tốc kế áp suất vi phân: (a) Lưu tốc kế Venturi, (b) Lưu tốc kế dạng vòi phun, (c) Lưu tốc kế ống phun, (d) Lưu tốc kế dạng ống nối cong (ống khuỷu), (e) Lưu tốc kế ống pitot tĩnh

Lưu tốc kế công suất chất lỏng được dùng trong những dòng chảy tốc độ thấp, độ nhớt trung bình. Ngoài các ứng dụng điều khiển trong công nghiệp, đôi khi các lưu tốc kế tua bin [hình 45.10(a)] còn được dùng làm đồng hồ chỉ tốc độ cho tàu thuyền. Lưu tốc kế guồng quay [hình 45.10(b)] được sử dụng trong các ứng dụng dòng chảy kín và mở như dòng chất lỏng trong máng. Do lưu tốc kế động cơ bánh răng [hình 45.10(c)] là thiết bị có thể tích không đổi nên tốc độ trục quay luôn luôn biểu thị trực tiếp tốc độ dòng chất lỏng.

Lưu tốc kế siêu âm dạng truyền [hình 45.11(a)] dựa trên nguyên lý tốc độ âm thanh tăng theo tốc độ dòng chất lỏng. Loại lưu tốc kế này được dùng cho các dạng dòng chảy sạch và dưới tốc độ âm thanh. Lưu tốc kế Doppler [hình 45.11(b)] lại phụ thuộc vào sự đội lại bên trong dòng chảy và do đó chỉ hữu ích với dòng chảy bẩn mang theo phần tử lơ lửng hoặc dòng chảy rối tạo ra bong bóng. Lưu tốc kế siêu âm là thiết bị không xâm nhập (không cần chỉnh sửa hoặc can thiệp vào ống dẫn) thường được trang bị thêm cho các hệ thống ống dẫn có sẵn.

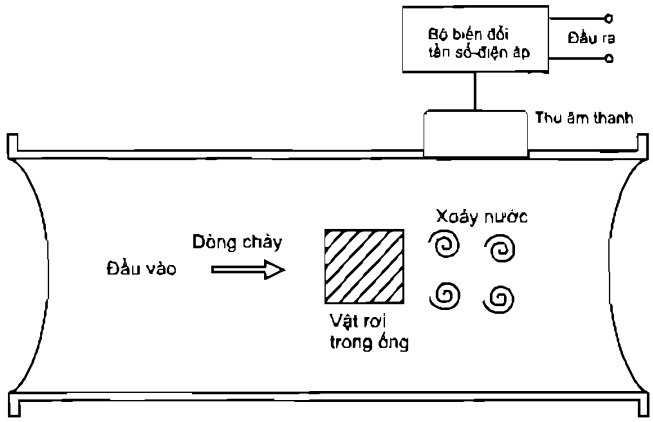


HÌNH 45.10 Lưu tốc kế áp suất chất lỏng: (a) Lưu tốc kế tua bin, (b) Lưu tốc kế guồng quay, (c) Lưu tốc kế hộp số

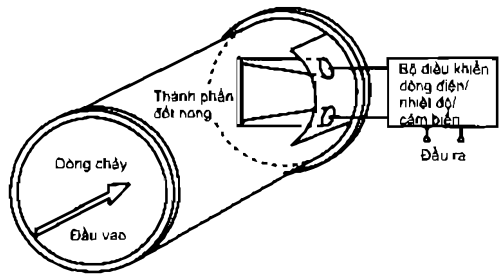


HÌNH 45.11 Lưu tốc kế siêu âm: (a) Lưu lượng kế siêu âm dạng truyền, (b) Lưu lượng kế siêu âm doppler

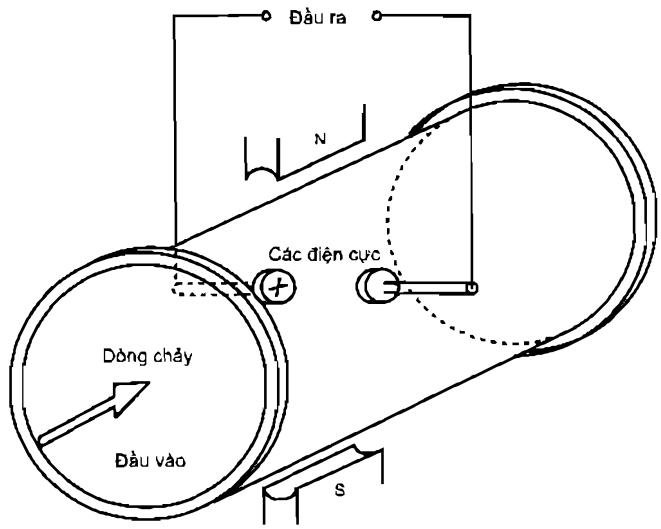
Lưu tốc kế dạng thả xoáy (hình 45.12) đưa vật vào dòng chảy để sinh ra *dòng chảy rối*. Âm thanh kèm với sự sinh ra và mất đi của dòng chảy rối được giám sát và phân tích. Tần số âm thanh trội biểu thị tốc độ sinh ra và mất đi của dòng chảy rối, và do đó nó cũng sẽ biểu thị cho tốc độ dòng chảy. Lưu tốc kế dạng thả xoáy rất hữu ích đối với những dòng chảy tốc độ thấp và không hỗn loạn.



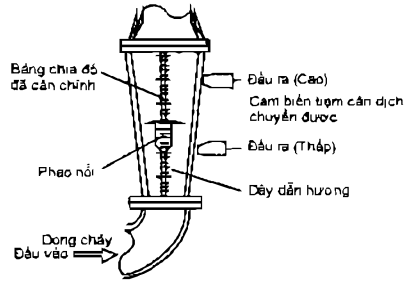
HÌNH 45.12 Lưu lượng kế thả xoáy



HÌNH 45.13 Phong tốc kế nhiệt



HÌNH 45.14 Lưu tốc kế điện từ



HÌNH 45.15 Lưu tốc kế có tiết diện thay đổi trên đường đi (lưu lượng kế kiểu phao)

Phong tốc kế nhiệt (hình 45.13) được dùng trong các dòng khí tốc độ thấp với tiết diện lớn như trong các ống dẫn nhiệt, ống thông hơi, và điều hoà không khí. Việc đo tốc độ dòng chảy được thực hiện dựa trên dòng điện cần thiết để duy trì nhiệt độ ổn định trong phần tử nhiệt hoặc dựa trên sự thay đổi nhiệt độ khi dòng điện được duy trì không đổi.

Lưu tốc kế điện từ (hình 45.14) hữu ích đối với những dòng chảy chậm như chất lỏng, bùn đặc hoặc huyền phù. Chất liệu dòng chảy phải hỗ trợ cho việc dẫn điện giữa các điện cực và do đó trong một số trường hợp để sử dụng được lưu tốc kế điện từ thì nhất thiết phải ion hoá dòng chảy từ điểm đo ngược trở lại.

Lưu tốc kế có tiết diện thay đổi trên đường đi (hình 45.15) hay còn gọi là *lưu lượng kế kiểu phao* đôi khi được coi như các thước ngắm do chúng cung cấp chỉ số có thể quan sát được của tốc độ dòng chảy. Những thiết bị này nếu được gắn thêm cảm biến tiệm cận (ví dụ như cảm biến tiệm cận điện dung) có chức năng nhận biết sự hiện diện của phao thì có thể được dùng trong các ứng dụng điều khiển đóng-ngắt.

Bộ chuyển đổi mức chất lỏng

Việc đo mức chất lỏng là khá dễ dàng và bộ chuyển đổi được phân thành dạng tiếp xúc hoặc không tiếp xúc. Các đo lường có thể liên tục, trong đó mức chất lỏng được giám sát liên tục trong phạm vi hoạt động của nó, hoặc theo từng điểm, trong đó mức chất lỏng được xác định nằm trên hoặc dưới một số mức đã định trước.

Bộ chuyển đổi tiếp xúc thường gặp là:

- Dạng phao
- Áp suất thủy tĩnh
- Điện dung
- Siêu âm

Bộ chuyển đổi không tiếp xúc thường gặp là:

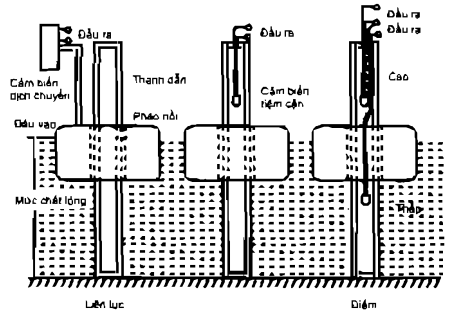
- Cảm biến tiệm cận dạng điện dung
- Siêu âm
- Tần số radiô
- Quang điện

Bộ chuyển đổi mức chất lỏng dạng phao có rất nhiều cấu hình được dùng trong các đo lường liên tục hoặc theo từng điểm. Dạng được mô tả trong hình 45.16 dùng cho đo lường liên tục và cho cả đo lường đơn điểm lẫn lưỡng điểm.

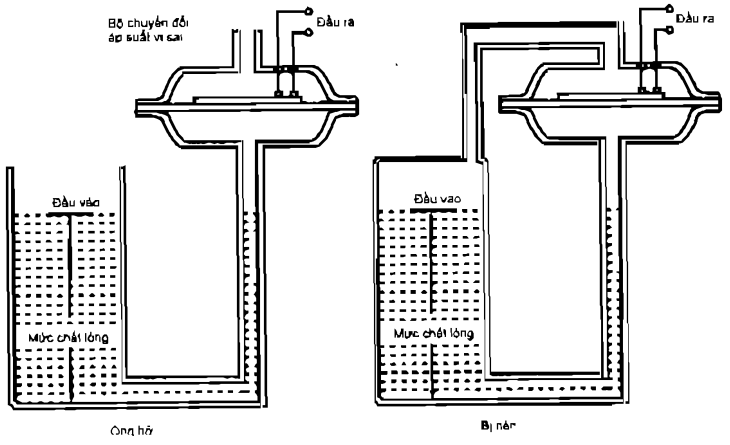
Bộ chuyển đổi mức chất lỏng áp suất thuỷ tĩnh có thể được dùng trong các ứng dụng thông hơi hoặc tạo áp suất (hình 45.17). Trong cả hai trường hợp này, áp suất vi phân tỷ lệ thuận với trọng lượng cột chất lỏng.

Các đầu dò điện dung [hình 45.18(a)] được sử dụng rộng rãi trong ứng dụng đo mức chất lỏng. Khi vách của thùng chứa là kim loại thì có thể sử dụng một thanh kim loại cách điện hoặc không

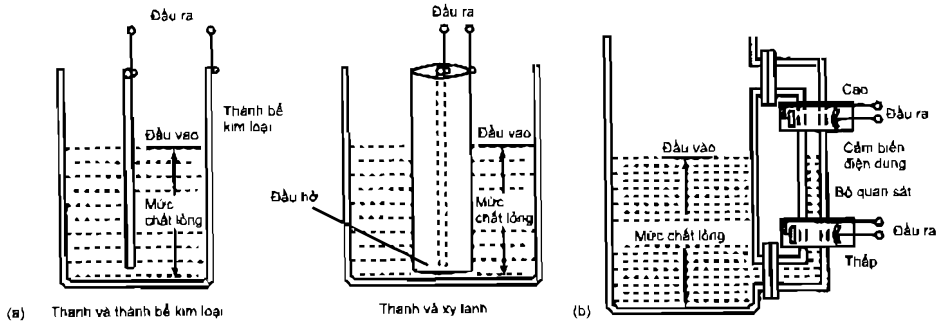
cách điện làm một bản mặt tụ điện, vách của thùng chứa sẽ là bản mặt khác của tụ điện. Nhìn chung, đầu dò điện dung thường có một thanh kim loại nằm trong một hình trụ đồng tâm hở hai đầu giúp cho bộ chuyển đổi không phụ thuộc vào cấu trúc thùng chứa. Một ứng dụng khá thú vị của đầu dò điện dung là đồng hồ chỉ lượng nhiên liệu trong máy bay. Bộ chuyển mạch điện dung có thể được sử dụng như mô tả trong hình 45.18(b) để đo mức chất lỏng theo phương pháp điểm không tiếp xúc.



HÌNH 45.16 Bộ chuyển đổi mức chất lỏng dạng phao



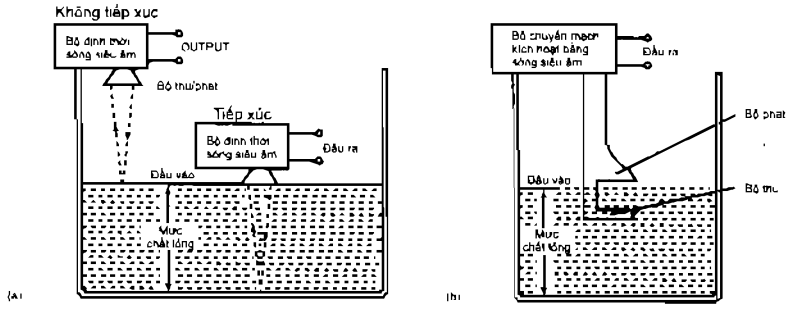
HÌNH 45.17 Bộ chuyển đổi mức chất lỏng áp suất thuỷ tĩnh



HÌNH 45.18 Bộ chuyển đổi mức chất lỏng dạng điện dung: (a) Đầu dò điện dung, (b) Bộ chuyển mạch điện dung

Bộ chuyển đổi siêu âm phản hồi có thể được dùng dưới dạng *nhúng ướt* (tiếp xúc) hoặc *không nhúng ướt* (không tiếp xúc) để thực hiện liên tục phép đo mức chất lỏng [hình 45.19(a)]. Một ứng

dụng khá thú vị của bộ chuyển đổi dạng nhúng ướt là máy dò độ sâu và máy dò cá cho tàu thuyền. Bộ chuyển đổi không nhúng ướt cũng có thể được sử dụng cho các vật liệu rời như hạt ngũ cốc và bột mì. Bộ chuyển đổi mức chất lỏng dạng tần số radiô và quang điện thường là những thiết bị dạng phản hồi không tiếp xúc có nguyên lý và ứng dụng tương tự với bộ chuyển đổi siêu âm không nhúng ướt.



HÌNH 45.19 Bộ chuyển đổi mức chất lỏng siêu âm: (a) Bộ chuyển đổi mức chất lỏng dạng phạm vi phản hồi. (b) Bộ chuyển mạch siêu âm

Bộ chuyển đổi siêu âm có thể thích hợp cho các phép đo theo từng điểm bằng cách đặt bộ truyền và bộ nhận đối diện nhau và cách nhau một khoảng nhất định [hình 45.19(b)]. Khi chất lỏng lấp đầy khoảng này thì sự suy giảm năng lượng siêu âm rõ ràng sẽ thấp hơn so với khi không khí lấp đầy khoảng này. Mạch chuẩn hóa tín hiệu sử dụng sự tăng mạnh mức năng lượng siêu âm này, được nhận biết nhờ bộ thu, để kích hoạt bộ chuyển mạch.

Bộ chuyển đổi nhiệt độ

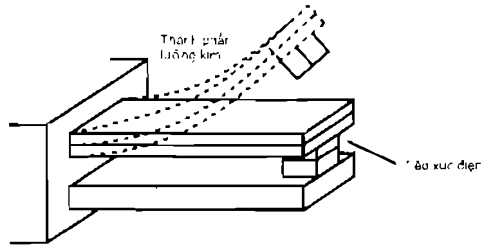
Đo nhiệt độ dựa trên một trong các nguyên lý vật lý sau:

- Giãn nở nhiệt
- Hiện tượng nhiệt điện
- Hiệu ứng nhiệt điện trở
- Hiệu ứng nhiệt trên các mối nối bán dẫn
- Bức xạ nhiệt

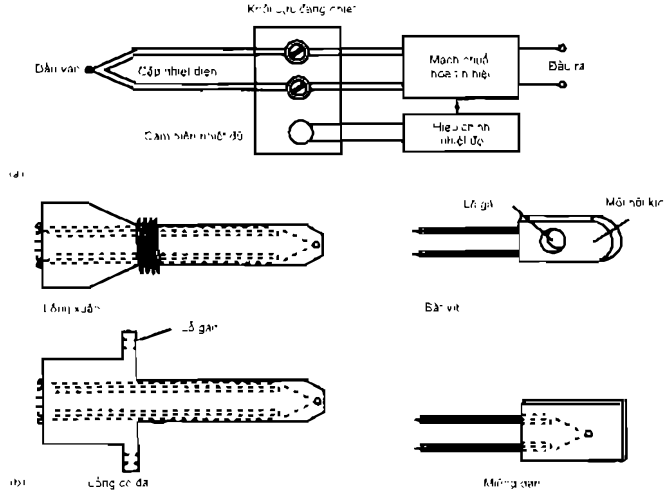
(Nói đúng ra, bất kỳ thiết bị nào được dùng để đo nhiệt độ đều có thể gọi là nhiệt kế, tuy nhiên người ta thường dùng những thuật ngữ mang tính mô tả hơn khi nói đến những thiết bị dùng để điều khiển nhiệt độ.)

Bộ chuyển mạch lưỡng kim (hình 45.20) được sử dụng rộng rãi trong các hệ thống điều khiển nhiệt độ dạng đóng-mở. Nếu hai tấm kim loại có *hệ số nở vì nhiệt* khác nhau được nối lại với nhau ở cùng nhiệt độ, thì cấu trúc lưỡng kim này sẽ bị uốn cong khi nhiệt độ thay đổi. Mặc dù những thiết bị này thường được gọi là *cầu chì nhiệt*, ngụ ý là chúng được sử dụng trong bộ chuyển mạch kín thông thường, nhưng chúng có thể được sản xuất theo dạng đóng hoặc mở. Các phần tử lưỡng kim cũng có thể được chế tạo ở dạng cuộn hoặc xoắn ốc để mở rộng phạm vi dịch chuyển do nở vì nhiệt.

Cặp nhiệt độ là những cảm biến nhiệt độ khỏe và linh hoạt thường thấy trong các hệ thống điều khiển công nghiệp. Một cặp nhiệt độ gồm một cặp dây kim loại khác nhau được xoắn lại với nhau hoặc được nối với nhau tại điểm cuối. *Hiệu ứng Seebeck* là hiện tượng vật lý giải thích hoạt động của cặp nhiệt độ, do đó cặp nhiệt độ có thể được gọi là *mối nối Seebeck*. Hiệu điện thế (*điện áp Seebeck*) giữa các đầu tự do của dây tỷ lệ với sự chênh lệch giữa nhiệt độ tại mối nối và nhiệt độ tại các đầu tự do. Cặp nhiệt độ có thể đo nhiệt độ từ -270°C đến 2300°C mặc dù không một cặp nhiệt độ nào có khả năng đo toàn bộ phạm vi này. Tùy thuộc vào kim loại dùng trong dây dẫn, cặp nhiệt độ được phân loại thành dạng B, C, D, E, G, J, K, N, R, S, hoặc T .



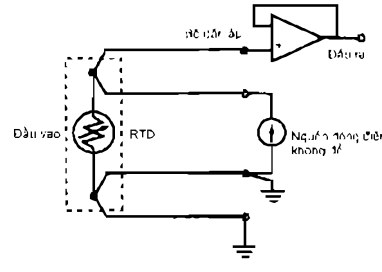
HÌNH 45.20 Bộ chuyển mạch nhiệt lưỡng kim



HÌNH 45.21 Cặp nhiệt độ: (a) Sơ đồ cặp nhiệt độ điển hình (b) Một số phụ tùng và hình dạng của cặp nhiệt độ

Điện áp Seebeck là khá nhỏ, do đó cần phải chuẩn hóa và khuyếch đại tín hiệu, tức là phải nối các dây dẫn cặp nhiệt độ với mạch chuẩn hóa tín hiệu. Những liên kết này tạo ra thêm hai mối nối Seebeck, mỗi mối nối này lại tạo ra một điện áp Seebeck cần loại bỏ. Để thực hiện việc loại bỏ cần các hiệu chỉnh sau [hình 45.21(a)]:

- Các cực đầu vào của mạch chuẩn hóa tín hiệu phải được làm từ cùng một loại kim loại.
- Hai cực phải nằm trên một *khối cực đẳng nhiệt* sao cho các mối nối Seebeck được tạo ra từ liên kết này có nhiệt độ giống nhau.
- Nhiệt độ của khối cực phải biết trước.



HÌNH 45.22 Một ứng dụng bộ dò nhiệt điện trở (RTD) tiêu biểu

Hai yêu cầu đầu tiên có thể được thoả mãn bằng cách xây dựng hợp lý mạch chuẩn hóa tín hiệu. Yêu cầu thứ ba có thể được đáp ứng bằng cách sử dụng một cảm biến nhiệt độ tham chiếu, có thể là loại bộ chuyển đổi nhiệt IC được trình bày dưới đây.

Cặp nhiệt độ và các phụ kiện của nó được chế tạo cho rất nhiều ứng dụng [hình 45.21(b)]. Tấm chắn bảo vệ (lồng nhiệt) được dùng để bảo vệ các mối nối của cặp nhiệt độ trong môi trường ăn mòn hoặc ở nơi chất lỏng dẫn điện có thể làm ngắn mạch điện áp của cặp nhiệt độ; tuy nhiên các mối nối trần (không cách ly) có thể được sử dụng ở bất cứ đâu, đặc biệt khi cần có đáp ứng nhanh.

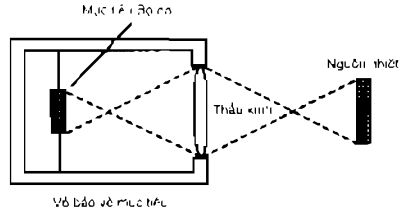
Bộ đo nhiệt độ điện trở (RTD) dựa trên nguyên lý *suất điện trở* của các kim loại tăng lên theo nhiệt độ. Đối với RTD, bạch kim là loại kim loại được ưa chuộng hơn cả, mặc dù nhiều kim loại khác loại rẻ tiền hơn vẫn được dùng trong một số ứng dụng. Suất điện trở của bạch kim là một trong những suất điện trở tiêu chuẩn dùng để đo nhiệt độ. Độ tuyến tính tương đối tốt của suất điện trở bạch kim trong một phạm vi nhiệt độ rộng (từ -200 đến 800°C) làm cho RTD bạch kim phù hợp với các bộ chuyển đổi nhiệt độ chính xác, ổn định, có thể dễ dàng thích nghi với các ứng dụng hệ thống điều khiển.

Nhược điểm của RTD là phần tử cảm nhận nhiệt độ là một sợi kim loại khá mong quấn quanh một ống gốm hoặc một lá kim loại mỏng gắn trên nền gốm. Các phần tử RTD thường được che kín và ít khi được sử dụng như những phần tử không cách điện. Các phụ kiện và gói ứng dụng dùng với RTD tương tự như những gì được sử dụng với cặp nhiệt độ [hình 45.21(b)].

Hầu hết các RTD bạch kim đều được chế tạo sao cho chúng có điện trở danh nghĩa là 100 W ở 0°C. *Hệ số nhiệt điện trở* của bạch kim xấp xỉ 3-4 mW/W °C, do đó độ phân giải nhiệt độ trong vòng 1°C đối với một phần tử RTD 100-W danh nghĩa đòi hỏi độ phân giải của điện trở tuyệt đối trong khoảng 0.3 0.4 W. Những yêu cầu về độ phân giải nhiệt độ này đòi hỏi phải sử dụng các kỹ thuật chuẩn hóa tín hiệu đặc biệt để loại bỏ điện trở dây dẫn và điện trở tiếp xúc của phần tử RTD (hình 45.22). Mạch mô tả trong hình 45.22 là dạng biến thể của *ôm kế 4 dây*. Hầu hết các RTD được sản xuất với 4 dây dẫn để có thể tương thích với những mạch như vậy.

Nhiệt điện trở là chất bán dẫn oxit kim loại đặc biệt có hệ số nhiệt độ âm mạnh, hoàn toàn đối lập với hệ số nhiệt độ dương yếu của RTDs. Điện trở danh nghĩa của nhiệt điện trở, thông thường ở 25°C, nằm trong khoảng từ dưới 1000 W đến hơn 1 MW, với độ nhạy lớn hơn 100 W/°C. Vì vậy, nhiệt điện trở là cơ sở cho các cảm biến nhiệt độ nhạy hơn và ít phải chuẩn hóa tín hiệu so với cặp nhiệt kế hoặc RTD. Tuy nhiên yếu tố cần quan tâm là tính phi tuyến rõ rệt của đặc tính điện trở-nhiệt độ. Để giải quyết vấn đề này, nhà sản xuất cung cấp các gói trong đó điện trở nhiệt được nối thành mạng điện trở để cung cấp đặc tính điện trở -nhiệt độ tuyến tính tương đối trong khoảng nhiệt độ danh nghĩa.

Sự phát triển của công nghệ điện trở nhiệt đã cho ra đời cảm biến nhiệt độ IC trong đó các mối nối nhạy nhiệt và mạch chuẩn hóa tín hiệu được cung cấp trong một gói nguyên khối. Yêu cầu đặt ra đối với người sử dụng là phải cung cấp điện áp nguồn (thông thường 5V DC) cho IC nhằm có được điện áp tương tự ở đầu ra tỷ lệ với nhiệt độ. Nhiệt điện trở và cảm biến nhiệt độ IC có thể được sản xuất với hình dáng nhỏ gọn cho phép thực hiện các phép đo nhiệt độ theo vùng. Một số nhiệt điện trở được thiết kế cho mục đích nghiên cứu sinh học (gắn trên đầu kim tiêm). Nhược điểm của cả nhiệt điện trở và cảm biến nhiệt độ IC là chúng không bền, không thể sử dụng trong môi trường hóa chất độc hại và bị giới hạn ở dưới mức nhiệt độ xấp xỉ 200°C.



HÌNH 45.23 Trình bày lược đồ nhiệt điện kế bức xạ

Nhiệt điện kế bức xạ được sử dụng để nhận biết nhiệt độ từ xa (không tiếp xúc) trong những tình huống cảm biến tiếp xúc không thể dùng được. Hoạt động của nó dựa trên nguyên lý truyền nhiệt thông qua bức xạ nhiệt. Nhiệt điện kế bức xạ tập trung năng lượng *hồng ngoại* từ một nguồn

nhiệt vào một *vật đen* (mục tiêu) trong phạm vi của nhiệt điện kế bức xạ (hình 45.23). Một trong những cảm biến nhiệt độ tiếp xúc mô tả ở trên được gắn vào mục tiêu để đo nhiệt độ của mục tiêu. Sự tăng nhiệt độ tại mục tiêu có liên quan đến nhiệt độ nguồn. Nhiệt điện kế bức xạ điển hình có phạm vi cân bằng (độ dài tiêu cự) trong khoảng 0.5-1.5m, nhưng thông thường các thiết bị đo hiện nay có độ dài tiêu cự từ 1 cm đến 10 m. Nhiệt điện kế bức xạ có phạm vi rộng, đơn sắc, hoặc nhiệt biểu học hai mầu.

45.4 Khả năng hoạt động của bộ chuyển đổi

Hoạt động của bộ chuyển đổi trong một hệ thống điều khiển được miêu tả dưới dạng *đặc tính tĩnh* và *đặc tính động* của nó. Đặc tính tĩnh được quan tâm nhiều nhất là:

- Hệ số tỷ lệ (hoặc độ nhạy)
- Độ chính xác, khoảng không rõ, và sai số hệ thống (hoặc độ lệch)
- Ngưỡng, độ phân giải, dải chết và hiện tượng trễ.
- Sự tuyến tính
- Độ trôi tương tự

Đặc tính động được quan tâm nhất là:

- Hằng số thời gian, thời gian đáp ứng.
- Quá mức, thời gian dừng.
- Đáp ứng tần số

Đặc tính tĩnh được tài liệu hoá thông qua *căn chỉnh*, tức là cung cấp một đầu vào đã biết (số lượng hoặc hiện tượng có thể đo được) sau đó quan sát và ghi lại giá trị đầu ra của bộ chuyển đổi. Trong một quy trình căn chỉnh điển hình, đầu vào được tăng lên dưới dạng số gia từ giới hạn dưới đến giới hạn trên của bộ chuyển đổi, sau đó lại giảm xuống giới hạn dưới. Phạm vi của một thiết bị bao gồm tất cả các giá trị đầu vào cho phép. Sự chênh lệch giữa giới hạn trên và giới hạn dưới là *khoảng đầu vào* của thiết bị; sự chênh lệch giữa đầu ra ở giới hạn trên với đầu ra ở giới hạn dưới là *khoảng đầu ra*.

Đặc tính động được tài liệu hoá bằng cách cung cấp một thay đổi đã biết (thường là một bước nhảy) ở đầu vào, rồi quan sát và ghi lại đầu ra của bộ chuyển đổi, thường với một máy ghi biểu đồ hoặc một máy hiện sóng có lưu trữ.

45.5 Sự phù hợp giữa tải và bộ chuyển đổi

Một yêu cầu thiết yếu đối với một bộ chuyển đổi thích hợp là nó phải *phù hợp* với đầu vào của nó. Sự phù hợp đó trong trường hợp này có nghĩa là năng lượng đầu vào cần thiết cho quá trình hoạt động bình thường của bộ chuyển đổi không làm thay đổi đáng kể giá trị cần đo. Ví dụ, một vôn kế phải có đầu vào trở kháng cao sao cho việc đo hiệu điện thế không thay đổi đáng kể dòng điện trong mạch và làm thay đổi hiệu điện thế đang được đo.

Tài liệu tham khảo

[1] Bateson, R.N. 1993. *Introduction to Control System Technology*, 4th ed. Merrill, Columbus, OH. Berlin, H.M. and Getz, F.C., Jr. 1988. *Principles of Electronic Instrumentation and Measurement*. Merrill, Columbus, OH.

[2] Buchla, D. and McLachlan, W. 1992. *Applied Electronic Instrumentation and Measurement*. Macmillan, New York.

[3] Chaplin, J.W. 1992. *Instrumentation and Automation for Manufacturing*. Delmar, Albany, NY.

[4] Doeblin, E.O. 1990. *Measurement Systems Application and Design*, 4th ed. McGraw-Hill, New York.

[5] Dorf, R.C. and Bishop, R.H. 1995. *Modern Control Systems*, 7th ed. Addison-Wesley, Reading, MA.

[6] O'Dell, T.H. 1991. *Circuits for Electronic Instrumentation*. Cambridge Univ. Press, Cambridge, England, UK.

[7] Seippel, R.G. 1983. *Transducers, Sensors, and Detectors*. Reston Pub., Reston, VA.

[8] Webb, J. and Greshock, K. 1993. *Industrial Control Electronics*, 2nd ed. Macmillan, New York.

46

Biến đổi A/D và D/A

46.1 Giới thiệu 46-1
46.2 Lấy mẫu 46-1
46.3 Đặc điểm kỹ thuật bộ ADC 46-2
46.4 Đặc điểm kỹ thuật bộ DAC 46-7

Mike Tyler
National Instruments, Inc.

46.1 Giới thiệu

Khi máy tính trở nên phổ biến, các kỹ sư và các nhà khoa học nhận ra rằng máy tính có thể trở thành một công cụ rất hữu ích. Tuy nhiên, hầu hết các hiện tượng tự nhiên (như ánh sáng, áp suất, vận tốc, nhiệt độ,...) là các tín hiệu tương tự, trong khi đó máy tính lại phụ thuộc vào các tín hiệu số. Vì vậy, rất nhiều công ty đã bắt tay vào việc cải tiến bộ chuyển đổi tương tự-số và số-tương tự (ADC và DCA). Những thiết bị này đã trở thành bộ phận chủ chốt trong tất cả các thiết bị đo lường. Chương này sẽ nghiên cứu các chức năng của ADC và DAC cũng như bàn về các đặc điểm kỹ thuật chính của chúng.

46.2 Lấy mẫu

Để chuyển đổi một tín hiệu tương tự thành tín hiệu số trước hết tín hiệu tương tự phải được lấy mẫu. Lấy mẫu là việc biến đổi giá trị của một tín hiệu ở một thời điểm nhất định. Nhìn chung, các biến đổi diễn ra đồng nhất theo thời gian. Ví dụ, một hệ thống số hóa có thể biến đổi một tín hiệu theo chu kỳ 5 μs hoặc lấy mẫu với tốc độ 200 kS/s. Mặc dù không nhất thiết phải lấy mẫu tín hiệu một cách đồng nhất như vậy nhưng việc này sẽ mang lại một số lợi ích nhất định mà sẽ được đề cập tới trong phần sau.

Một mạch lấy mẫu điển hình gồm có hai thành phần chính: Mạch track-and-hold (T/H) và bộ ADC. Do quá trình biến đổi thực tế trong bộ ADC diễn ra trong một khoảng thời gian nhất định nên cần giữ nguyên giá trị của tín hiệu đang được biến đổi. Tại thời điểm lấy mẫu, mạch T/H sẽ lưu giá trị lấy mẫu ngay cả khi tín hiệu vẫn đang thay đổi. Khi quá trình chuyển đổi kết thúc, mạch T/H sẽ giải phóng các giá trị hiện đang lưu và sẵn sàng cho việc theo dõi các giá trị tiếp theo.

Một vấn đề không tránh khỏi khi lấy mẫu là một số thông tin sẽ bị bỏ qua. Điều này có nghĩa là một dạng sóng tương tự có vô số mẫu mà không có cách nào lưu lại tất cả. Khó khăn lớn nhất trong vấn đề này là việc lấy mẫu diễn ra quá chậm. Nếu một sóng hình sin 10 kHz được thu nhận và việc lấy mẫu chỉ diễn ra với tốc độ 5 kS/s thì dạng sóng thực sự sẽ không được duy trì. Trên thực tế sẽ sinh ra một dạng sóng có tần số khác. Hậu quả của việc lấy mẫu chậm là tạo ra sóng

răng cưa. Theo lý thuyết Nyquist về lấy mẫu, quá trình lấy mẫu nên diễn ra với tốc độ gấp hai lần thành phần tần số cao nhất của tín hiệu. Nhìn chung, học thuyết này chỉ duy trì tần số của tín hiệu, do đó nếu muốn hình dạng của sóng theo yêu cầu thì có lẽ việc lấy mẫu ít nhất phải nhanh gấp 10 lần tín hiệu.

46.3 Đặc điểm kỹ thuật bộ ADC

Giới hạn

Giới hạn đầu vào của bộ ADC là khoảng điện áp mà bộ ADC có thể thực hiện chức năng chuyển đổi. Ví dụ, giới hạn đầu vào thông thường cho một bộ ADC là 0-5 V. Điều này có nghĩa là bộ ADC có thể chuyển đổi một đầu vào trong khoảng 0-5 V. Nếu giới hạn dưới bằng 0 V thì giới hạn được gọi là đơn cực và nếu giới hạn trên và giới hạn dưới có cùng độ lớn, ví dụ như -5 V đến +5 V, thì giới hạn của bộ ADC được gọi là lưỡng cực. Nếu điện áp đầu vào nằm ngoài giới hạn thì bộ ADC được gọi là bị vượt quá giới hạn. Trong trường hợp này, hầu hết các ADC sẽ trả về một giá trị gần nhất với điện áp cần biến đổi.

Độ phân giải

Độ phân giải của một bộ số hóa là sự thay đổi điện áp nhỏ nhất có thể phát hiện ra được. Tuy nhiên, độ phân giải của một ADC thường liên quan đến số lượng bit nhị phân nó sinh ra. Ví dụ, bộ ADC 12-bit biểu diễn một giá trị tương tự được biến đổi bằng cách sử dụng 12 bit. Bộ biến đổi 12 bit này có thể biến đổi một giá trị sang một trong 4096 ($=2^{12}$) mức khác nhau. Một cách thông dụng khác để xác định độ phân giải là thông qua các số thập phân. Một vôn kế 6 bit đo trên thang 1 V có thể đo các bước có giá trị 0.000001 V từ -0.999999 V đến 0.999999 V.

Quy ước mã hoá

Các định dạng khác nhau mà một bộ ADC có thể sử dụng để biểu diễn đầu ra được gọi là quy ước mã hoá. Một bộ ADC sử dụng mã hóa nhị phân sinh ra toàn giá trị 0 ở giới hạn dưới và toàn giá trị 1 ở giới hạn trên (ví dụ như bộ chuyển đổi 3 bit sẽ sinh ra các giá trị từ 000 đến 111).

Sai số tuyến tính

Sai số tuyến tính là sai số lớn nhất và phổ biến nhất trong bộ ADC nhưng cũng có thể dễ dàng sửa được thông qua căn chỉnh đơn giản hoặc bằng cách cộng và nhân với các hằng số hiệu chỉnh. Mặc dù sai số tuyến tính không bóp méo chức năng truyền dữ liệu của ADC, nhưng chúng có thể làm thay đổi giới hạn làm việc chính xác của ADC.

Sai số phi tuyến

Khác với sai số tuyến tính, việc bù sai số phi tuyến dưới dạng tín hiệu số hoặc tương tự có vẻ khó khăn hơn. Cách tốt nhất để giảm bớt sai số phi tuyến là lựa chọn một bộ ADC có thiết kế và cấu tạo tốt. Sai số phi tuyến được đặc trưng bởi hai thông số kỹ thuật khác nhau: độ phi tuyến vi phân (DNL – differential nonlinearity) và độ phi tuyến tích phân (INL – integral nonlinearity).

DNL đo tính không đều trong độ rộng mã (thay đổi nhỏ nhất có thể phát hiện được) bằng cách so sánh sự thay đổi giá trị thực tế với giá trị lý tưởng của một độ rộng mã (hoặc 1 LSB). INL đo độ lệch so với đường truyền lý tưởng của chuyển tiếp mã.

Một đặc tính kỹ thuật quan trọng khác của bộ ADC có liên quan đến độ phi tuyến vi phân là nếu bất kỳ mã nào bị mất đi thì kích thước bước tại điểm đó trong hàm truyền được nhân đôi, khiến độ phân giải địa phương của bộ ADC bị giảm xuống còn một nửa. Vì thế, tài liệu kỹ thuật của bộ ADC sẽ ghi rõ bộ ADC có bị mất mã hay không.

Một cách khác để lưu giữ thông tin như vậy là phép đo độ chính xác tương đối. Độ chính xác tương đối không những chỉ ra phép biến đổi mã hoá sai lệch bao nhiêu so với mã lý tưởng mà còn cho thấy bất kỳ thành phần nào của hàm truyền, bao gồm cả sai số "bậc thang" lượng tử hoá, lệch bao xa so với mã lý tưởng. Trong một ADC lý tưởng không nhiễu, độ chính xác tương đối trong trường hợp xấu thường lớn hơn INL. Tuy nhiên, nếu bộ ADC có nhiễu nội tại và có nhiễu (liên quan đến dao động) được thêm vào đầu vào thì độ chính xác tương đối sẽ tăng lên. Việc thêm nhiễu vào bộ lượng tử hoá có xu hướng làm mịn hàm truyền trung bình. Hàm truyền này thường gây ra ít hiệu ứng "bậc thang" hơn. Sự cải thiện độ tuyến tính của hàm truyền có được là vì sai số biến đổi do nhiễu đã được giảm bớt.

Lỗi khe hở

Tất cả các ADC đều cần một tín hiệu điều khiển để báo cho ADC biết khi nào bắt đầu thực hiện chuyển đổi. Cần một khoảng thời gian kể từ khi bộ ADC nhận được tín hiệu điều khiển cho đến khi việc chuyển đổi được thực hiện xong. Khoảng thời gian này được gọi là trễ "khe hở". Hầu hết các ADC đều có trễ khe hở khoảng một vài nano giây. Tuy nhiên, hầu hết các thiết bị đo đều có mạch điện khác ở phía trước ADC như bộ khuyếch đại chẳng hạn. Mạch điện này có chức năng loại bỏ trễ khe hở do bộ ADC gây ra. Chẳng hạn, nếu bộ ADC có trễ 10 ns và bộ khuyếch đại có trễ 160 ns, thì trễ khe hở thực tế của hệ thống là -150 ns.

Một thông số thời gian quan trọng khác là "rung". Rung (hay còn gọi là rung khe hở) đo chênh lệch thời gian giữa các lần lấy mẫu. Nếu một tín hiệu được lấy mẫu với tốc độ 1 triệu mẫu trên giây (1 MS/s), thì khoảng thời gian mong đợi giữa mỗi lần lấy mẫu sẽ là 1μs. Thời gian thực tế giữa các mẫu có thể thay đổi từ 1 μs đến một vài nano giây từ chu kỳ này đến chu kỳ khác. Rung có thể do nguồn xung nhịp, mạch tạo xung hoặc mạch S/H gây ra. Hiệu ứng phổ biến nhất của rung là tạo ra nhiễu ở các mức tần số rất gần với tín hiệu cần lấy mẫu.

Nhiễu

Nhiễu làm hạn chế độ phân giải của bộ ADC do có một dạng sóng xen vào trong tín hiệu đầu vào khi nó đang được chuyển đổi. Nguồn nhiễu phổ biến nhất trong tín hiệu là nhiễu nhiệt. Nhiễu nhiệt do các thuộc tính ngẫu nhiên của các thành phần điện tử gây ra. Khi nhiệt độ và điện trở trong các thành phần tăng lên thì nhiễu nhiệt cũng tăng lên. Các nguồn nhiễu phổ biến khác là nhiễu điện do các mạch điện liền kề với bộ ADC gây ra, ví dụ như các mạch logic và mạch tạo xung nhịp.

Sai số lượng tử hoá, được nói tới trong phần trước, đôi khi còn được gọi là nhiễu lượng tử hoá. Mặc dù sai số lượng tử hóa hoàn toàn có thể dự đoán được đối với tín hiệu đầu vào nhưng khi một tín hiệu khá "bận" (tức là các chuyển đổi liên tiếp không sinh ra các bit dữ liệu chung) thì sai số lượng tử hoá sẽ trở nên xáo trộn. Khi vấn đề này xảy ra, lỗi lượng tử hóa có thể được coi là một nguồn nhiễu ngẫu nhiên khác có phân bố thống kê đồng nhất từ -0.5 đến 0.5 LSB và có độ lệch chuẩn là $1/\sqrt{12}$ LSB. Trong phân tích phổ, sai số này đôi khi còn được gọi là nguồn nhiễu chủ yếu.

Khi ADC bị nhiễu, có nhiều cách để loại bỏ nhiễu khỏi tín hiệu, với điều kiện nhiễu là một tín hiệu độc lập. Một trong những cách thông thường nhất để giảm nhiễu trong một phép đo DC là thu thập các điểm và tính trung bình các giá trị. Nếu nhiễu là loại nhiễu trắng ngẫu nhiên có mật độ năng lượng bằng nhau ở tất cả các tần số thì việc tính bình quân sẽ giúp làm giảm lượng nhiễu bằng cách lấy căn bậc hai số mẫu được tính bình quân. Nếu nhiễu giao thoa với một dạng sóng lặp (repetitive waveform) thì có thể giảm nhiễu bằng cách đo các sóng và tính trung bình các. Hầu hết các máy hiện sóng số đều có chức năng bình quân các dạng sóng.

Hầu hết các đặc tính nhiễu của bộ ADC đều được xây dựng với các tín hiệu tĩnh có trở kháng thấp. Để duy trì đặc tính nhiễu của bộ ADC, người dùng phải nối các tín hiệu với đầu vào bằng dây được bảo vệ nhằm giữ cho tín hiệu không bị nhiễu điện từ.

Khoảng động lực

Khoảng động lực là tỷ số giữa tín hiệu lớn nhất và tín hiệu nhỏ nhất mà bộ ADC có thể biểu diễn. Khoảng động lực được xác định bằng cách lấy giá trị tín hiệu giới hạn và đem so sánh nó với mức nhiễu nhỏ nhất có thể phát hiện được của bộ ADC. Khoảng động lực thường được biểu thị dưới dạng đêxiben (dB) và có thể được xác định bởi công thức sau:

$$\text{Khoảng động lực} = 20 \log (S/N)$$

Trong đó S là mức tín hiệu lớn còn N là mức nhiễu. Mức nhiễu bao gồm nhiễu lượng tử hóa của ADC. Đối với một bộ ADC lý tưởng thì nhiễu này tương đương với $1/\sqrt{12}$ LSB rms. Một sóng hình sin giới hạn có biên độ là 2^{n-1} LSB hoặc $2^{n-1}/\sqrt{2}$ (n = số lượng bit của ADC). Vì vậy, một bộ ADC lý tưởng có khoảng động lực bằng:

$$\text{Khoảng động lực} = 6.0206n + 1.7609$$

Do không có bộ ADC nào là lý tưởng nên số lượng bit thực tế (ENOB – effective number of bits) của một ADC có thể tính được nhờ công thức trên. ENOB biểu diễn độ phân giải thực tế của bộ ADC và có thể tính được thông qua khoảng động lực nhờ công thức sau:

$$\text{NOB} = (\text{khoảng động lực} - 1.7609) / 6.0206$$

Ví dụ, một bộ ADC 12 bit với khoảng động lực là 69 dB có ENOB là 11.17 bit.

Các loại ADC

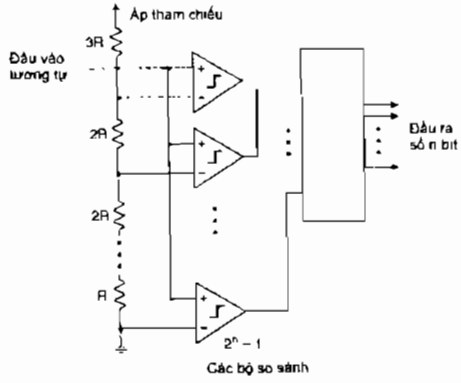
Tất cả các loại ADC đều thực hiện nhiệm vụ giống nhau là biến đổi một tín hiệu tương tự thành một tín hiệu số. Hai đặc trưng cơ bản của một bộ ADC là tốc độ chuyển đổi và độ phân giải. Trong hầu hết các trường hợp đều có một sự thoả hiệp giữa tốc độ và độ phân giải. Chẳng hạn, hiện nay chưa có bộ chuyển đổi 24 bit với tốc độ 100 MS/s, nhưng lại có bộ chuyển đổi (có thể là 8 bit) với tốc độ 100 MS/s và bộ chuyển đổi 24 bit (với tốc độ 1 kS/s). Điều quan trọng là phải hiểu được bộ ADC sẽ được sử dụng như thế nào để lựa chọn áp dụng những bộ chuyển đổi phù hợp.

Mặc dù có rất nhiều bộ ADC khác nhau, nhưng chúng đều có những đặc điểm chung. Phần trọng tâm của bất kỳ bộ ADC nào là bộ so sánh. Bộ so sánh là một bộ ADC 1 bit đơn giản, có hai đầu vào tương tự và một đầu ra số. Một trong hai tín hiệu vào tương tự là điện áp so sánh có giá trị biết trước. Đầu vào còn lại hoặc là lớn hơn hoặc là nhỏ hơn giá trị đầu vào so sánh và giá trị này sẽ chuyển thành giá trị số 0 hoặc 1. Trên thực tế một bộ ADC có nhiều bộ so sánh nhưng nguyên lý cơ bản của chúng đều giống nhau.

Flash

ADC kiểu Flash là bộ ADC có tốc độ nhanh nhất. Tuy nhiên, nếu có sự thoả hiệp giữa tốc độ và độ phân giải như đã nói ở trên thì các bộ chuyển đổi Flash nhìn chung có độ phân giải là 10 bit hoặc thấp hơn. Một bộ chuyển đổi Flash với độ phân giải n bit gồm 2^{n-1} bộ so sánh tốc độ cao làm việc song song, xem hình 46.1. Một chuỗi 2^{n-1} điện trở giữa hai tham chiếu điện áp sẽ sinh ra một tập hợp các mức điện áp cách đều nhau. Những điện áp này được nối với đầu vào của mỗi bộ so sánh. Tín hiệu điện áp đầu vào sau đó được đem so sánh với các mức điện áp trên một cách đồng thời. Bộ so sánh sẽ trả về giá trị 1 cho tất cả các điện áp dưới mức điện áp đầu vào, và giá trị 0 cho tất cả điện áp trên mức điện áp đầu vào. Những giá trị số kết quả này sau đó được chuyển sang dạng logic để cho ra một giá trị n-bit.

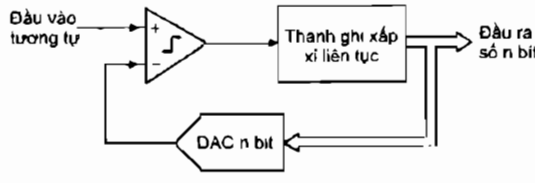
Do được thiết kế đơn giản, các bộ chuyển đổi Flash rất nhanh, nhưng khi độ phân giải của bộ chuyển đổi tăng lên thì số lượng các bộ so sánh và điện trở cần thiết cũng tăng lên theo hàm mũ. Cả kích thước và công suất cần thiết để vận hành bộ chuyển đổi cũng tăng lên theo hàm mũ khi độ phân giải tăng lên, do đó độ phân giải của các bộ chuyển đổi Flash thường bị giới hạn. Tuy nhiên, do các giá trị của điện trở chỉ khác nhau một vài phần trăm nên đặc tính tuyến tính vi phân của bộ ADC Flash tương đối tốt.



HÌNH 46.1 ADC Flash có 2^{n-1} bộ so sánh làm việc đồng thời

ADC thanh ghi xấp xỉ liên tục

Bộ ADC có thanh ghi xấp xỉ liên tục (SAR – successive-approximation register) là bộ ADC phổ biến nhất với độ phân giải 8-16 bit và tốc độ đạt tới 10 MS/s. Những bộ ADC này có giá phải chăng và nhìn chung chúng đều có đặc tính tuyến tính tích phân khá tốt. Bộ ADC SAR có một bộ DAC tốc độ cao nằm trong một vòng lặp phản hồi, xem hình 46.2. SAR lặp lại DAC thông qua nhiều mức, các mức này sau đó được đem so sánh với điện áp đầu vào. Khi quá trình chuyển đổi diễn ra, SAR sẽ tạo ra đầu ra số *n*-bit, là kết quả của những so sánh này. Khi SAR hoàn thành, đầu ra của DAC càng gần với tín hiệu đầu vào càng tốt và đầu vào số của DAC sẽ trở thành đầu ra của bộ ADC SAR.

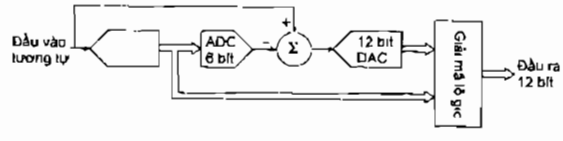


HÌNH 46.2 Bộ chuyển đổi xấp xỉ liên tụ ở một bộ so sánh

Tốc độ của bộ DAC SAR bị giới hạn bởi khả năng làm việc của bộ DAC bên trong vòng lặp phản hồi. Để đạt được tốc độ nhanh hơn, cấu tạo SAR có thể được dùng làm cơ sở cho một bộ ADC khác, ADC đa tầng.

ADC đa tầng

Để đạt được tốc độ cao hơn SAR, ADC đa tầng dùng phương pháp lặp của SAR nhưng làm giảm số lượng bộ so sánh cần thiết để hoàn thành quá trình chuyển đổi. Bên cạnh bộ so sánh, ADC đa tầng sử dụng các bộ chuyển đổi Flash có độ phân giải thấp, xem hình 46.3. Trong hình này, Flash 6 bit được dùng để chuyển đổi các sai số dư từ Flash 8 bit. Hai đầu ra này của ADC sau đó được kết hợp nhờ logic số để sinh ra một đầu ra 12 bit.



HÌNH 46.3 ADC đa tầng - Bộ chuyển đổi đa tầng là sự kết hợp giữa bộ chuyển đổi SAR và bộ chuyển đổi Flash để có thể lấy mẫu với tốc độ cao hơn bộ chuyển đổi SAR và có độ phân giải cao hơn bộ chuyển đổi Flash

Trên thực tế, hầu hết các ADC đa tầng đều là ADC ống dẫn (Pipelined). ADC ống dẫn có cấu tạo giống như một ADC đa tầng nhưng mỗi bộ chuyển đổi Flash lại gồm có một T/H tại đầu vào. Nó cho phép từng tầng chuyển đổi sai số dư trong khi tầng trước đó đã chuyển sang mẫu tiếp theo. Theo cách này, cả bộ chuyển đổi có thể vận hành ở tốc độ của tầng thấp nhất, đối lập với ADC đa tầng vận hành ở tốc độ bằng tổng của tất cả các tầng.

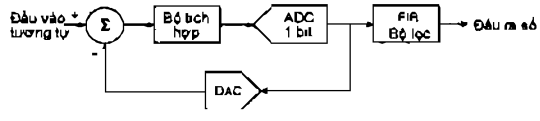
ADC tích hợp

ADC tích hợp hoàn toàn đối lập với bộ chuyển đổi Flash. Chúng được thiết kế để có độ phân giải cao nhưng vận hành với tốc độ chậm hơn. Chúng có thiết kế rất đơn giản. ADC tích hợp gồm một bộ khuyếch đại tích hợp, một bộ so sánh, một bộ đếm số và một tụ điện rất ổn định. ADC tích hợp phổ biến nhất là độ dốc kép (dual slope). Với cấu tạo này, ban đầu tụ được phóng điện để không có điện thế trên nó. Trong một khoảng thời gian định sẵn, đầu vào được đưa vào tụ điện và nó bắt đầu nạp điện trong khoảng thời gian định sẵn T_1 Do đặc tính của tụ điện nên tốc độ nạp điện tỷ lệ với điện áp đầu vào. Sau khoảng thời gian T_1, tụ điện được chuyển sang điện áp so sánh âm và bắt đầu phóng điện với tốc độ tỷ lệ với điện áp đầu vào so sánh. Bộ đếm số đo khoảng thời gian T_2 mà tụ điện phóng điện hoàn toàn. Do khoảng thời gian T_1 và tốc độ phóng điện của tụ điện là những giá trị đã biết nên điện áp đầu vào có thể tính được bằng một phép tỷ lệ đơn giản.

Vấn đề quan trọng cần lưu ý là bộ biến đổi thực tế không đo điện áp đầu vào. Thay vào đó, bộ ADC có được giá trị điện áp bằng cách đo thời gian và sử dụng logic số để tính điện áp đầu vào. Phương pháp này có ưu điểm là loại bỏ được nhiễu mà các loại ADC khác rất dễ mắc phải như nhiễu chu kỳ chẳng hạn. Thêm vào đó, hầu hết các ADC tích hợp đều vận hành trên nhiều chu kỳ xoay chiều (1/60 hoặc 1/50 s) do đó các nhiễu điện từ do hệ thống điện năng gây ra được loại bỏ.

ADC Sigma-Delta.

ADC Sigma-Delta (SD) là một trong những loại ADC phổ biến nhất do sự phù hợp của mối quan hệ tốc độ và độ phân giải. ADC SD có thể cung cấp độ phân giải từ 16 đến 24 bit với tốc độ lấy mẫu lên tới hàng trăm nghìn mẫu trên giây. Với tốc độ và độ phân giải như vậy, ADC này trở nên khá lý tưởng trong một số ứng dụng nhất định như phân tích dao động và âm thanh. Tuy nhiên quá trình tích phân khiến cho ADC SD có độ chính xác điện một chiều không cao. Hình 46.4 cho thấy thiết kế của một ADC SD. Phần trung tâm của một ADC SD chính là ADC 1-bit với tốc độ lấy mẫu rất cao. Ngoài ADC tốc độ cao, cấu tạo của ADC SD còn có một bộ lọc thông thấp tương tự và một bộ DAC cùng nằm trong vòng phản hồi. Kết quả sẽ buộc nhiễu lượng tử hóa chuyển sang các dải tần số cao hơn. Quang phổ do nhiễu sinh ra là một phần trong quá trình tạo dạng nhiễu. Đầu ra của vòng phản hồi này sau đó được đưa tới một bộ lọc số. Bộ lọc số sau đó sẽ tăng độ phân giải, giảm tốc độ truyền dữ liệu và dùng một bộ lọc số thông thấp cho những dữ liệu đi ra khỏi vòng phản hồi. Sau quá trình này, bộ ADC SD sẽ có một đầu ra với độ phân giải cao, tín hiệu nằm trong dải tần số mong muốn và loại bỏ phần lớn các nhiễu điện từ nội tại.



HÌNH 46.4 Bộ chuyển đổi sigma-delta dùng bộ so sánh 1-bit để xác định giá trị tín hiệu

Bộ chuyển đổi số - tương tự (DAC)

Trong khi bộ ADC nhận tín hiệu giá trị tương tự và tạo ra giá trị số thì ngược lại bộ DAC lại nhận giá trị số và tạo ra giá trị tương tự. Bộ chuyển đổi DAC là một thiết bị, với sự biểu diễn số của một tín hiệu cho trước, có thể tạo ra một tín hiệu tương tự tại một mức điện áp nhất định. Mặc dù nhiều lý thuyết về ADC đã được áp dụng với DAC, nhưng DAC vẫn có một tập hợp các thuật ngữ và hiện tượng riêng.

Cập nhật

Cập nhật có thể được coi tương tự như lấy mẫu. Nếu một DAC phát ra sóng hình sin từ tập hợp các giá trị số, chúng ta cần có cách nào đó để xác định dạng sóng này được sinh ra như thế nào. Nói một cách đơn giản, tốc độ cập nhật có nghĩa là bộ DAC có thể sinh ra một giá trị tương tự với bao nhiêu điểm trên giây, và nhìn chung nó được biểu thị dưới dạng mẫu trên giây, kilo mẫu trên giây hoặc triệu mẫu trên giây.

46.4 Đặc điểm kỹ thuật bộ DAC

Giới hạn

Giới hạn của DAC giống như định nghĩa về giới hạn của ADC. Nó biểu diễn phạm vi giá trị điện áp mà DAC có thể tạo ra.

Độ phân giải

Độ phân giải của một DAC được định nghĩa giống như bộ ADC. Tuy nhiên, thực tế lại hoàn toàn ngược lại. Trong bộ DAC, độ phân giải cho biết có bao nhiêu bit số biểu diễn cho một giá trị tương tự. Độ phân giải còn cho chúng ta biết có bao nhiêu bit số cần cung cấp cho DAC để tạo ra tín hiệu mong muốn.

Tính đơn điệu

Một trong những đặc tính kỹ thuật hữu ích nhất của DAC là tính đơn điệu. Nếu bộ DAC là đơn điệu thì khi giá trị số sẽ tăng lên, giá trị đầu ra tương tự cũng tăng lên hoặc ít nhất là vẫn giữ nguyên. Ngược lại, một thiết bị được coi là không đơn điệu. Nhiều ứng dụng rất nhạy cảm với những thay đổi của giá trị đầu ra vì thế bất kỳ DAC được ứng dụng nào cũng cần đơn điệu trên tất cả các bit.

Thời gian biến đổi và tốc độ thay đổi (Settling Time and Slew Rate)

Thời gian biến đổi và tốc độ thay đổi quyết định một DAC có thể thay đổi giá trị tương tự nó đang tạo ra nhanh như thế nào. Thời gian biến đổi là lượng thời gian cần thiết để đầu ra của bộ DAC đạt tới độ chính xác mong muốn. Hầu hết các bộ DAC đều quy định thời gian biến đổi là sự thay đổi điện áp trong cả dải thang đo, từ giá trị đầu ra nhỏ nhất đến giá trị đầu ra lớn nhất. Tốc độ thay đổi, được biểu thị bằng vôn trên giây, là tốc độ thay đổi đầu ra lớn nhất của DAC. Vì vậy, một DAC với tốc độ thay đổi nhanh và thời gian biến đổi nhỏ có thể sinh ra các tín hiệu tần số cao do một mức điện áp chính xác có thể đạt được trong một khoảng thời gian ngắn.

Sai số dịch và sai số khuyếch đại

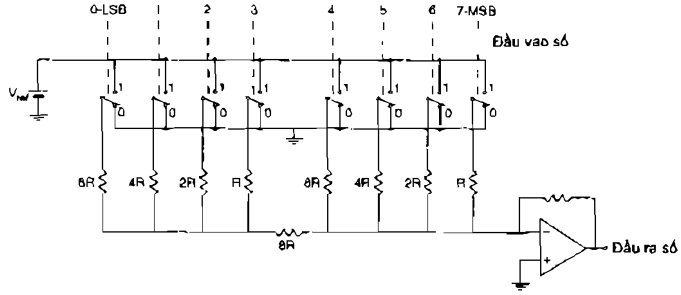
Sai số dịch ám chỉ đặc tính truyền của bộ DAC không sinh ra giá trị tương tự 0 khi giá trị số đầu vào là 0. Sai số dịch có thể được coi như sự tịnh tiến đường truyền lên hoặc xuống từ đường truyền lý tưởng. Sai số khuyếch đại cho biết độ lệch tuyến tính từ đường truyền lý tưởng của DAC. Nó có thể được tạo ra bởi nhiều yếu tố dẫn đến sự thay đổi độ dốc so với đường truyền lý tưởng.

Cấu tạo của DAC

Không giống với ADC, DAC không thực hiện nhiều các phương pháp để chuyển mã đầu vào số sang giá trị tương tự. Thay vào đó, hầu hết các DAC sử dụng sự tổ hợp của mạng chuyển mạch, mạng điện trở và bộ khuyếch đại cộng. Đó là chưa kể đến tất cả DAC đều có chung thiết kế nhưng chúng đều dựa trên nguyên tắc chuyển mạch.

Mạng chuyển mạch

Hệ thống chuyển mạch của DAC có thể được coi là phần trung tâm của quá trình chuyển đổi. Do các bit số có thể bật hoặc tắt nên chúng có thể được dùng để điều khiển các công tắc một cực. Những công tắc này sau đó được dùng để điều khiển một dạng mạch tương tự nào đó nhằm sinh ra một giá trị tương tự. Ví dụ, một DAC 3 bit sẽ bao gồm 3 bộ chuyển mạch, mỗi bộ cho một bit dữ liệu đầu vào. Tùy theo mã định sẵn, những bộ chuyển mạch này sẽ đóng lại theo cách có thể sinh ra một giá trị tương tự từ một nguồn tham chiếu, xem hình 46.5. Căn cứ trên thiết kế của sơ đồ mạch điện tương tự trong DAC, các bộ chuyển mạch có thể kết nối các tham chiếu dòng điện hoặc điện áp với mạng điện trở.



HÌNH 46.5 Cấu tạo DAC - Hầu hết các bộ chuyển đổi số-tương tự (DAC) đều theo một cấu tạo chuẩn của mạng chuyển mạch, mạng điện trở và bộ khuyếch đại

Mạng điện trở

Mạng điện trở trong DAC cung cấp điện áp đầu ra tương tự ở các mức khác nhau giúp hình thành nên đầu ra của DAC. Mặc dù có rất nhiều mạch điện trở được ứng dụng trong thiết kế DAC nhưng chúng đều có chung nguyên tắc cơ bản. Trong hình 46.5, bit 0-3 và 4-7 được chia cắt bởi một điện trở. Trong hai nhóm độc lập này, mỗi tứ giác điện trở có các giá trị điện trở là 1*R*-2*R*-4*R*-8*R* trong đó *R* bằng 10 kWs. Nếu bất kỳ bộ chuyển mạch nào ở vị trí số 1 thì một dòng điện sẽ được chạy qua điện trở có giá trị tỷ lệ với giá trị điện trở. Do đó, nếu bit 0 tích cự thì dòng điện tỷ lệ với 1/1*R* sẽ được sinh ra, nếu bộ chuyển mạch 2 tích cực thù dòng điện tỷ lệ với 1/4R sẽ được sinh ra. Điện trở giữa 2 bộ điện trở có hiệu ứng suy giảm dòng điện 16:1, do đó ngay cả bit 4 có thể sinh ra một dòng điện tỷ lệ với 1/1*R*, thì khi nó đến bộ khuyếch đại, nó sẽ có một dòng điện tỷ lệ với 1/16R. Trong trường hợp này, 0 bit sẽ là MSB và 7 bit sẽ là LSB.

Bộ khuyếch đại cộng

Bộ khuyếch đại thuật toán (op-amp) được dùng trong mạch điện DAC như hình 46.5 đóng vai trò như một bộ khuyếch đại cộng. Khi các bit khác nhau sinh ra một dòng điện nhất định thì op-amp sẽ thu thập toàn bộ dòng điện và sinh ra một điện áp đầu ra . Điện áp đầu ra này của op-amp là biểu diễn dưới dạng tương tự của đầu vào số cung cấp cho DAC.

47

Chuẩn hóa tín hiệu

Stephen A. Dyer
Kansas State University

47.1 Tác động tuyến tính 47-1
47.2 Tác động phi tuyến 47-5

Nguyên tắc đầu tiên của Kelvin về thiết bị đo lường cho rằng, về cơ bản, thiết bị đo không làm thay đổi đối tượng đo. Với những mục đích hiện tại, ta có thể coi thiết bị đo bao gồm một bộ chuyển đổi đầu vào, tiếp đến là phần chuẩn hóa tín hiệu và phần hiển thị (phần còn lại của thiết bị đo). Ta đang sử dụng thuật ngữ *thiết bị đo* theo nghĩa rộng, tức là nó có thể thực sự là một hệ thống con trong bất kỳ hệ thống nào.

Bộ chuyển đổi dùng để mô phỏng chân thực một sự kiện phải thoả mãn những yêu cầu nhất định sau: tuyến tính biên độ, tuyến tính pha và đặc tuyến tần số tương xứng. Nhiệm vụ của bộ chuẩn hóa tín hiệu là nhận các tín hiệu đầu ra từ bộ chuyển đổi và từ đó sinh ra một dạng tín hiệu phù hợp với phần còn lại của thiết bị đo.

Chuẩn hóa tín hiệu tương tự có thể bao gồm những tác động hoàn toàn *tuyến tính*, những tác động hoàn toàn *phi tuyến* hoặc sự kết hợp của cả hai. Ngoài ra, bộ chuẩn hóa tín hiệu có thể được dùng để cung cấp các chức năng phụ trợ như tạo ra sự cách ly về mặt điện, cung cấp một dạng tham chiếu cho bộ chuyển đổi, hoặc sinh ra một tín hiệu kích thích cho bộ chuyển đổi.

Những ví dụ quan trọng về các tác động tuyến tính bao gồm *biến đổi biên độ, biến đổi trở kháng, lọc tuyến tính và điều biến.*

Một vài ví dụ về tác động phi tuyến bao gồm tính toán *giá trị bình phương trung bình (root mean square - rms), căn bậc hai, giá trị tuyệt đối,* hoặc lấy *logarit* tín hiệu đầu vào.

Có rất nhiều các khối được xây dựng dưới dạng môđun hoặc mạch tích hợp (IC) để thực hiện quá trình chuẩn hóa tín hiệu tương tự. Các khối như vậy bao gồm bộ khuếch đại thuật toán, bộ khuếch đại đo, bộ khuếch đại cách ly và các mạch xử lý phi tuyến như bộ so sánh, bộ nhân/chia tương tự, bộ khuếch đại loga/đối loga, bộ biến đổi rms-thành-DC và bộ tạo hàm lượng giác.

Ngoài ra còn có các hệ thống con chuẩn hóa tín hiệu hoàn chỉnh bao gồm các môđun vào/ra theo kiểu plug-in có khả năng liên kết với nhau thông qua các tấm đế đa năng (backplane).

47.1 Tác động tuyến tính

Ba loại tác động tuyến tính quan trọng để chuẩn hóa tín hiệu là thay đổi biên độ, biến đổi trở kháng và lọc tuyến tính.

Biến đổi biên độ

Biên độ của tín hiệu ở đầu ra bộ chuyển đổi thường phải được khuếch đại hoặc làm suy giảm trước khi được xử lý.

Khuếch đại

Nhìn chung, khuếch đại được thực hiện thông qua *bộ khuếch đại thuật toán*, *bộ khuếch đại đo* hoặc *bộ khuếch đại cách ly*.

Bộ khuếch đại thuật toán

Một bộ khuếch đại thuật toán thông thường có một đầu vào vi phân và một đầu ra. Bộ khuếch đại thuật toán *lý tưởng* có hệ số khuếch đại bằng vô cùng, dải thông vô hạn, trở kháng đầu vào vi phân vô cùng lớn, tốc độ vô cùng lớn và tỷ lệ loại tín hiệu cùng pha (CMRR – Common Mode Rejection Ratio) vô hạn. Ngoài ra, nó còn có trở kháng đầu ra bằng không, nhiễu bằng không, dòng thiên áp lệch bằng không và điện áp lệch đầu vào bằng không. Tất nhiên, các bộ khuếch đại thuật toán thực không thể so sánh được với các khuếch đại thuật toán lý tưởng.

Khi lựa chọn một khuếch đại thuật toán cần xem xét các thông số quan trọng sau:

1. Hệ số khuếch đại điện áp một chiều K_0.
2. Tích số dải thông-hệ số khuếch đại tín hiệu nhỏ f_T (GBWP - Gain Brandwidth Product). Đối với hầu hết các bộ khuếch đại thuật toán thì $f_T \approx K_0 f_1$, trong đó f_1 là tần số ngắt bé hơn trong hàm truyền của bộ khuếch đại thuật toán. GBWP đặc trưng cho vòng lặp kín, đáp ứng tần số cao của mạch khuếch đại thuật toán.
3. Tốc độ đáp ứng chỉ phối ứng xử tín hiệu của bộ khuếch đại thuật toán. Tốc độ đáp ứng nằm trong khoảng từ nhỏ hơn 1 V/μs đến vài nghìn V/μs.

Ngoài ra, cần xem xét các thông số khác như trở kháng đầu vào/ra, điện áp lệch một chiều, dòng điện lệch, điện áp và dòng điện trôi, các đặc tính nhiễu, v. v..., khi lựa chọn bộ khuếch đại thuật toán để ứng dụng trong một trường hợp nào đó.

Có nhiều loại bộ khuếch đại thuật toán. Bên cạnh khuếch đại thuật toán "thông dụng" còn có một vài bộ khuếch đại thuật toán có các tính năng được tối ưu hóa để dùng trong một vài ứng dụng. Các loại khuếch đại thuật toán bao gồm:

1. Khuếch đại thuật toán có *nhiễu thấp*, sử dụng trong một số bộ phận của bộ chuẩn hóa tín hiệu nhằm khuếch đại các tín hiệu rất thấp.
2. Khuếch đại thuật toán *ngắt ổn định*, sử dụng trong các ứng dụng yêu cầu độ ổn định một chiều cao.
3. Khuếch đại thuật toán *nhanh*, sử dụng khi cần tốc độ đáp ứng và GBWP lớn.
4. Khuếch đại thuật toán *công suất*, sử dụng khi cần cung cấp các dòng điện lớn hơn một vài mA cho tải của khuếch đại thuật toán.
5. Khuếch đại thuật toán *tĩnh điện kế* được sử dụng khi cần điện trở đầu vào rất lớn ($>10^{13}\ \Omega$) và dòng điện thiên lệch đầu vào rất nhỏ (< 1 pA).

Các bộ khuếch đại thuật toán và cấu hình mạch điện cơ bản thường tìm thấy trong các sách giáo khoa về lý thuyết mạch điện hoặc điện tử học. Bạn đọc có thể tìm thấy những tài liệu mang tính lý thuyết khá chi tiết và rất nhiều cấu hình và ứng dụng hữu ích trong Roberge (1975), Graeme và cộng sự (1971), Graeme (1973, 1977), Horowitz và Hill (1989), Stout và Kaufman (1976).

Bộ khuếch đại đo

Bộ khuếch đại đo là các khối khuếch đại được tối ưu hoá để cung cấp trở kháng đầu vào cao, trở kháng đầu ra thấp, độ khuếch đại ổn định, tỷ lệ loại tín hiệu cùng pha (CMR – Common Mode Rejection) khá cao, độ lệch (offset) tương đối thấp. Chúng rất phù hợp để khuếch đại tín hiệu đầu ra của các bộ chuyển đổi như biến dạng kế, hoặc khuếch đại các tín hiệu thấp có kèm theo các điện áp cao cùng pha và trong các trường hợp cần sự cách ly về điện giữa bộ chuyển đổi và phần còn lại của thiết bị đo.

Mặc dù các bộ khuếch đại đo có thể được xây dựng từ các bộ khuếch đại thuật toán truyền thống [Thông thường người ta hay nhắc tới cấu hình 3 khuếch đại thuật toán, ví dụ: xem Stout và Kaufman (1976)]. Chúng sẵn có với giá cả phải chăng dưới dạng IC. Một số bộ khuếch đại đo có hệ số khuếch đại khả trình, trong khi đó những bộ khuếch đại đo khác lại được lập trình thông bằng cách kết nối các điện trở trong với IA thông qua các chân (pin) bên ngoài.

Bộ khuếch đại cách ly

Bộ khuếch đại cách ly được dùng trong các ứng dụng mà ở đó điện áp hoặc dòng điện có điện áp cao cùng pha kèm theo phải được đo một cách an toàn, chính xác và với CMR cao. Chúng cũng rất hữu ích để đảm bảo an toàn trong khi đo các dòng điện rò một chiều và có tần số, ví dụ như trong thiết bị đo y sinh học.

Bộ khuếch đại cách ly có thể được coi như sự kết hợp của ba phần: tầng ra, tầng vào và mạch công suất. Tất cả những bộ khuếch đại cách ly đều có các tầng vào cách điện với các tầng ra. Quá trình trao đổi thông tin giữa các tầng vào và tầng ra được thực hiện thông qua quá trình điều biến/giải điều biến.

Một bộ khuếch đại cách ly cho phép cách điện hai cổng nếu có kết nối một chiều (DC) giữa mạch công suất và tầng ra của nó. Nếu mạch công suất của nó được cách điện với tầng ra cũng như với tầng vào thì bộ khuếch đại sẽ cho phép cách điện ba cổng.

Các bộ khuếch đại cách ly có thể tồn tại dưới dạng môđun với sự cách ly hai cổng hoặc ba cổng. Hiện nay có cả môđun một kênh và nhiều kênh.

Sự suy giảm

Mặc dù phần lớn các bộ chuyển đổi là những thiết bị cấp thấp như cặp nhiệt điện, nhiệt điện trở, RTD (Resistance Temparature Detector) và biến dạng kế có các đầu ra cần khuếch đại, nhưng cũng có rất nhiều tình huống đo lường trong đó tín hiệu đầu ra phải được làm suy giảm trước khi đưa vào phần còn lại của hệ thống.

Biến đổi điện áp

Về cơ bản, các tín hiệu được suy giảm phải có dạng điện áp. Nói rộng ra là việc làm suy giảm được thực hiện thông qua *bộ chia điện áp* hoặc *máy biến thế*.

Bộ chia điện áp

Trong nhiều trường hợp, một bộ chia chuỗi đơn giản được coi là thích hợp. Hàm truyền của một chuỗi hai thành phần của trở kháng Z_1 (s) và Z_2 (s) là:

$$\frac{V_0(s)}{V_{in}(s)} = \frac{Z_1(s)}{Z_1(s) + Z_2(s)}$$

Trong đó điện áp đầu ra $V_0(s)$ là điện áp trên với $Z_1(s)$ và điện áp đầu vào V_{in} là điện áp tổng của $Z_1(s)$ và $Z_2(s)$. Đương nhiên, trở kháng của bộ chuyển đổi và tải (phần còn lại của hệ thống) phải được xem xét khi thiết kế hệ thống bộ chia.

Bộ phân chia dùng điện trở

Nếu các phần tử trong chuỗi là điện trở thì bộ chia sẽ phát huy tác dụng từ vùng DC lên tới các tần số mà không có thành phần điện trở nào. Nếu $Z_1(s) = R_1$ và $Z_2(s) = R_2$ thì

$$\frac{V_0(s)}{V_{in}(s)} = \frac{R_1}{R_1 + R_2}$$

Các bộ chia dùng điện trở có thể có các cấu hình khác. Ví dụ như bộ chia Kelvin-Varley, với một vài lợi thế giúp nó trở nên hữu ích trong những tình huống cần có độ chính xác cao. Xem mô tả chi tiết trong Gregory (1973).

Bộ chia dùng điện dung

Nếu các phần tử trong bộ chia dạng chuỗi là tụ điện thì bộ chia có hàm truyền là

$$\frac{V_0(s)}{V_{in}(s)} = \frac{C_2}{C_1 + C_2}$$

Loại bộ chia này được sử dụng trong khoảng từ tần số thấp đến tần số vài MHz. Một ứng dụng phổ biến là biến đổi những điện áp lớn.

Bộ chia điện cảm

Nếu các phần tử trong chuỗi bộ chia là các cuộn cảm thì có thể dùng một bộ tự biến thế. Các bộ chia điện cảm rất hữu ích trong khoảng tần số từ một vài Hz đến một vài trăm kHz. Có thể đạt được sai số trong khoảng vài phần tỷ.

Máy biến thế

Máy biến thế là một trong những thiết bị phổ biến nhất thực hiện việc biến đổi điện áp ở các tần số trên đường truyền. Cấu hình dây đôi tiêu chuẩn rất hiệu quả nếu mức điện áp trên 200 kV được giám sát. Các loại cấu hình khác như *biến áp tụ điện* và *biến áp ghép tầng* được sử dụng trong trường hợp mức điện áp quá cao (Gregory, 1973).

Biến đổi dòng điện

Việc biến đổi dòng điện thường được thực hiện thông qua bộ phân dòng (Shunt) hoặc bộ biến dòng (current transformer).

Một *bộ phân dòng* về bản chất là một điện trở đã biết có dòng điện cần đo chạy qua. Điện áp sinh ra do dòng điện chạy qua điện trở là đại lượng cần đo. Bộ phân dòng được dùng trong các dòng điện một chiều và các tần số vùng âm thanh. Hai nhược điểm là: (1) bộ phân dòng tiêu hao năng lượng và (2) mạch điện đo phải có cùng mức điện thể với bộ phân dòng.

Bộ biến dòng khắc phục được những nhược điểm đã nêu của bộ phân dòng. Thông thường, bộ biến dòng bao gồm một lõi hình xuyến có cấu trúc đặc biệt trong đó cuộn thứ cấp được che kín và cuộn sơ cấp đi qua nó. Mặc dù cuộn sơ cấp đa vòng rất phổ biến nhưng người ta thường dùng cuộn sơ cấp đơn vòng.

Các bộ suy giảm khác

Ngoài các thiết bị biến đổi điện áp và biến đổi dòng điện đã nêu còn có các mạng hai cửa. Không những làm giảm điện áp hoặc công suất, chúng còn có khả năng làm hòa hợp trở kháng giữa mạch nguồn và mạch tải. Các mạng hai cửa thông thường bao gồm dạng T, dạng L và dạng Π, có thể là đối xứng hoặc không đối xứng. Các mạng hai cửa có điện trở được trình bày trong hầu hết các cuốn sách giáo khoa về thiết kế mạch điện (ví dụ như Cuthbert, 1983). Chúng rất hữu ích từ vùng một chiều đến vài trăm MHz.

Biến đổi trở kháng

Thông thường trở kháng của bộ chuyển đổi phải được biến đổi sang một giá trị thích hợp với phần còn lại của hệ thống đo lường. Trong nhiều trường hợp, công suất tối đa phải được chuyển từ tín hiệu đầu ra của bộ chuyển đổi đến mạch điện còn lại. Trong các trường hợp khác, chỉ cần cung cấp vùng đệm có trở kháng cao đối với bộ chuyển đổi và trở kháng thấp đối với phần còn lại của hệ thống và hệ số khuếch đại điện áp bằng một là đủ.

Các bộ chuyển đổi phù hợp, các hệ thống thụ động như các mạng hai cửa và bộ đệm hệ số khuếch đại bằng 1 là những thiết bị tiêu chuẩn thực hiện quá trình biến đổi trở kháng. Các bộ đệm hệ số khuếch đại bằng 1 có dạng IC.

Lọc tuyến tính

Mặc dù quá trình xử lý tín hiệu số có rất nhiều ưu điểm so với kỹ thuật tương tự trong việc lọc tín hiệu nhưng có nhiều ứng dụng đơn giản rất thích hợp với *lọc tương tự lựa chọn tần số*.

Các bộ lọc được sử dụng trong những bộ chuẩn hóa tín hiệu nhằm làm giảm ảnh hưởng của nhiễu đến tín hiệu đầu vào, đóng vai trò như một bộ phận của bộ giải điều biến, giới hạn dải thông tín hiệu và hạn chế dải thông tín hiệu nhằm ngăn chặn hiện tượng tín hiệu răng cưa khi lấy mẫu tín hiệu. Những bộ lọc này có thể được xây dựng hoàn toàn dựa trên các bộ phận bị động hoặc các thiết bị chủ động như bộ khuếch đại thuật toán.

Nhiều tài liệu tham khảo trình bày các phương pháp mô tả, xác định và xây dựng các bộ lọc tương tự tần số chọn. Xem Van Valkenburg (1960) để biết thiết kế của các bộ lọc bị động, xem Sedra và Brackett (1978) và Stephenson (1985) để biết thiết kế của các bộ lọc chủ động dùng RC.

47.2 Tác động phi tuyến

Nhiều tác động phi tuyến rất hữu ích đối với các nhiệm vụ chuẩn hóa tín hiệu. Dưới đây là mô tả tóm tắt một vài khối phi tuyến điển hình . Hầu hết các khối là các IC.

1. *Bộ so sánh.* Bộ so sánh là một thiết bị hai đầu vào có điện áp đầu ra V_o nhận hai giá trị ổn định, V_{o0} và V_{o1}, như sau:

$$V_o = \begin{cases} V_{o0}, \text{nếu } V_2 < V_1 \\ V_{o1}, \text{ngược lại} \end{cases}$$

Trong đó V_1 và V_2 là những điện áp ở hai đầu vào.

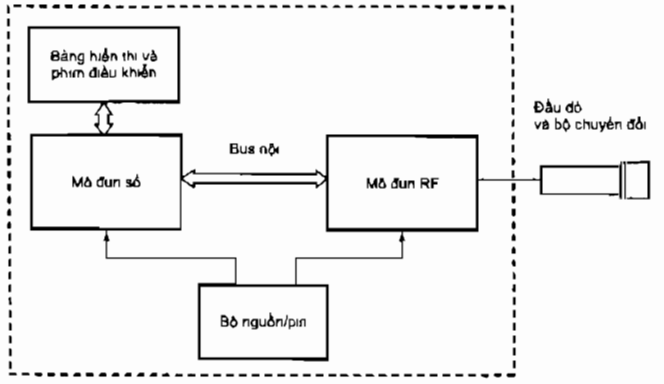
2. *Trigger Schmitt.* Là một bộ so sánh có trễ. Nó có thể được xây dựng từ một bộ so sánh bằng cách sử dụng hồi tiếp dương.
3. *Bộ nhân.* Một bộ nhân hai đầu vào cung cấp một điện áp đầu ra tỷ lệ với tích của các điện áp đầu.
4. *Bộ chia.* Một bộ chia hai đầu vào có điện áp đầu ra tỷ lệ với tỷ số các điện áp đầu vào. Các chức năng nhân hoặc chia thông thường được kết hợp trong cùng một thiết bị.
5. *Bộ bình phương.* Một bộ bình phương có điện áp đầu ra tỷ lệ với bình phương đầu vào. Các bộ bình phương có thể được xây dựng dựa trên: các bộ nhân, các mạng điôt-điện trở và các tranzito FET...
6. *Bộ căn bậc hai.* Một bộ căn bậc hai có điện áp đầu ra tỷ lệ với căn bậc hai của đầu vào. Một bộ căn bậc hai có thể được xây dựng khá dễ dàng từ một bộ chia hoặc từ một bộ khuếch đại lôga/đối lôga.
7. *Bộ khuyếch đại lôga/đối lôga.* Một bộ khuếch đại lôga/đối lôga có điện áp đầu ra tỷ lệ với lôga hoặc đối lôga điện áp đầu vào.
8. *Bộ biến đổi RMS - DC.* Một bộ biến đổi RMS-DC tính căn bậc hai trung bình của bình phương tức thời tín hiệu đầu vào trong một khoảng thời gian nào đó. Phép toán trung bình thường được thực hiện thông qua một bộ lọc thông thấp đơn giản có tụ điện để tạo ra những khoảng ngắt mong muốn.
9. *Bộ tạo hàm lượng giác.* Các bộ tạo hàm dưới dạng IC có đầu ra là các hàm lượng giác chuẩn. Chúng là hàm của điện áp sai lệch đầu vào.
10. *Bộ khuếch đại sample-and-hold và track-and-hold.* Bộ khuếch đại *sample-and-hold* (SHA) là một thiết bị lấy mẫu tín hiệu ở đầu vào và lưu các giá trị tức thời có yêu cầu của tín hiệu điều khiển lôgic. Bộ khuếch đại *track-and-hold* cũng có đặc tính như bộ SHA nhưng nó được dùng trong các ứng dụng mà ở đó hầu hết thời gian được dành cho việc theo dõi các tín hiệu đầu vào. Điều này hoàn toàn đối lập với SHA khi hầu hết thời gian được sử dụng ở chế độ "giữ".

11. *Mạch dựa trên điốt chính xác.* Các mạch điện như bộ chỉnh lưu nửa sóng chính xác, mạch điện giá trị tuyệt đối, bộ dò đỉnh chính xác và bộ giới hạn chính xác được thiết kế và xây dựng khá dễ dàng dựa trên các điôt và khuếch đại thuật toán. Xem Horowitz và Hill (1989), Stout và Kaufman (1976) và Graeme (1977).

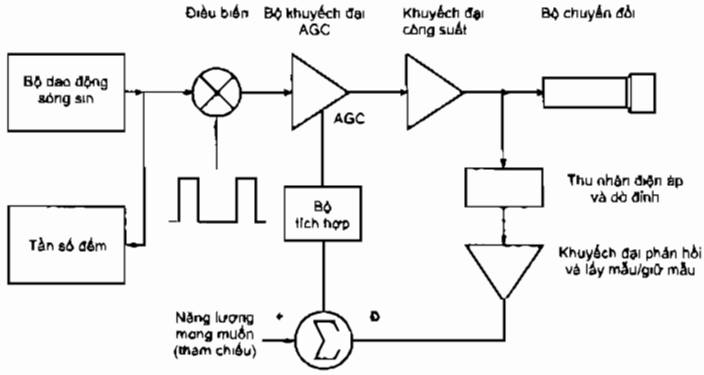
Mô tả chi tiết về những thiết bị này và các khối mạch điện phi tuyến khác có thể được tìm thấy trong Sheingold (1976).

Ví dụ

Xin dẫn ra một ví dụ về một thiết bị có các mạch điện chuẩn hóa tín hiệu được nhúng bên trong. Hình 47.1 chỉ ra sơ đồ khối cơ bản của một thiết bị siêu âm, được ứng dụng rộng rãi trong y học.



HÌNH 47.1 Sơ đồ khối cơ bản của thiết bị siêu âm dùng trong y học



HÌNH 47.2 Sơ đồ khối đơn giản của môđun tần số radiô dùng trong thiết bị siêu thanh mô tả ở hình 47.1

Thiết bị này gồm 5 hệ thống con chủ yếu:

1. Một đầu dò ứng dụng và một bộ chuyển đổi siêu âm truyền năng lượng siêu âm đến tế bào đang được chữa trị. *Cần lưu ý rằng bộ chuyển đổi này không phải là một bộ chuyển đổi đầu vào như đã giới thiệu trong phần bộ chuẩn hóa tín hiệu.*
2. Một môđun tần số radiô (RF) cung cấp tín hiệu kích thích điện đến bộ chuyển đổi siêu âm.
3. Bộ hiển thị và các công tắc chuyển mạch cho phép giao tiếp giữa thiết bị và người điều khiển.
4. Một môđun số dựa trên bộ vi xử lý đóng vai trò điều khiển toàn bộ thiết bị siêu âm.

5. Nguồn nuôi/ắc quy cung cấp năng lượng cho thiết bị hoạt động.

Bây giờ chúng ta tập trung vào môđun RF với sơ đồ khối cơ bản được minh họa trong hình 47.2. Môđun bao gồm một bộ tạo sóng hình sin phát ra tín hiệu ở tần số cộng hưởng của bộ chuyển đổi, một bộ điều biến cho phép tín hiệu được điều biến thành các xung và một bộ khuếch đại với phản hồi điện áp RF. Bên trong môđun RF là một bộ khuếch đại công suất có khả năng vận hành bộ chuyển đổi và một mạch điện điều khiển hệ số khuếch đại tự động (AGC – Automatic-gain-control) nhằm điều chỉnh công suất đầu ra cho phù hợp với yêu cầu của người. AGC sử dụng một vòng điều khiển phản hồi chuẩn để duy trì một điện áp không đổi trên đầu ra tín hiệu RF từ bộ khuếch đại công suất.

Một số bộ chuẩn hóa tín hiệu được áp dụng trong môđun RF bao gồm:

1. Bộ lọc điện áp RF ở đầu ra của bộ khuếch đại công suất. Bộ lọc sử dụng một bộ chỉnh lưu nửa sóng, sau đó là một bộ chia điện dung dạng chuỗi đơn giản để biến đổi điện áp.
2. Một bộ dò đỉnh chính xác để lấy giá trị đỉnh của đầu ra bộ chia điện áp trong chu trình điều biến và đưa giá trị đó tới vòng phản hồi.
3. Một bộ khuếch đại có hệ số thay đổi được để khuếch đại đầu ra của bộ dò đỉnh.
4. Bộ khuếch đại sample-and-hold dùng để lưu đầu ra được khuếch đại từ bộ dò đỉnh trong thời gian giữa hai chu kỳ của bộ điều biến. Sử dụng SHA là cần thiết do hằng số thời gian của bộ dò đỉnh không đủ để ngăn chặn hiện tượng "droop" trong thời gian giữa hai chu kỳ của bộ điều biến.
5. Một bộ tích phân tạo ra điện áp điều khiển cho vòng AGC từ đầu ra của bộ vi phân.
6. Một bộ phân dòng điện, không được biểu diễn trong hình 47.2, được dùng để giám sát dòng điện DC cung cấp cho bộ khuếch đại công suất.

Có thể thấy trong ví dụ này nhiều hàm chuẩn hóa tín hiệu được áp dụng trong cùng một hệ thống đơn giản và có thể chính hệ thống này lại không phải là một thiết bị!

Định nghĩa các thuật ngữ

Loại tín hiệu cùng pha (Common Mode Rejection - CMR): CMRR được biểu thị dưới dạng đexiben. CMR = 20 $\log|CMRR|$. CMR là một hàm phi tuyến của điện áp cùng pha và phụ thuộc vào các yếu tố khác như nhiệt độ.

Tỷ lệ loại tín hiệu cùng pha (Common Mode Rejection Ratio - CMRR): Tỷ lệ giữa hệ số khuếch đại vi phân với hệ số khuếch đại cùng pha của bộ khuếch đại.

Tích số dải thông-hệ số khuếch đại (Gain Bandwith Product - GBWP): Tích số giữa hệ số khuếch đại lớn nhất của bộ khuếch đại và dải thông tương ứng của nó.

Tốc độ đáp ứng: Tốc độ đáp ứng tối đa có thể đạt được của bộ khuếch đại tương ứng với sự thay đổi của điện áp đầu vào.

Tài liệu tham khảo

[1] Cuthbert, T. R. 1983. *Circuit Design Using Personal Computers*. John Wiley & Sons, New York.

[2] Graeme, J. G. 1973. *Applications of Operational Amplifiers*. McGraw-Hill, New York.

[3] Graeme, J. G. 1977. *Designing with Operational Amplifiers*. McGraw Hill, New York.

[4] Graeme, J. G., Tobey, G. E., and Huelsman, L. P. (Ed.) 1971. *Operational Amplifiers*. McGraw-Hill, New York.

[5] Gregory, B. A. 1973. *An Introduction to Electrical Instrumentation*. Macmillan, London.

[6] Horowitz, P. and Hill, W. 1989. *The Art of Electronics*, 2nd ed. Cambridge University Press, New York.

[7] Roberge, J. K. 1975. *Operational Amplifiers*. John Wiley & Sons, New York.

[8] Sedra, A. S. and Brackett, P. O. 1978. *Filter Theory and Design: Active and Passive*. *Matrix*, Beaverton, OR.

[9] Sheingold, D. H. (Ed.) 1976. *Nonlinear Circuits Handbook*. Analog Devices, Norwood, MA.

[10] Stephenson, F. W. 1985. *RC Active Filter Design Handbook*. John Wiley & Sons, New York.

[11] Stout, D. F. and Kaufman, M. (Ed.) 1976. *Handbook of Operational Amplifier Circuit Design*. McGraw-

[12] Hill, New York.

[13] Van Valkenburg, M. E. 1960. *Introduction to Modern Network Synthesis*. John Wiley & Sons, New York.

48

Các hệ thống đo dựa trên máy tính

48.1 Sức mạnh của phần mềm 48-2
48.2 Số hóa thế giới tương tự 48-3
48.3 Triển vọng 48-4

Kris Fuller
National Instruments, Inc

Ngày nay, các hệ thống tự động hóa, hệ thống đo lường nối mạng và dựa trên máy tính đều có các phần mềm mang lại hiệu quả cao trong một môi trường quen thuộc. Với những hệ thống này, các kỹ sư không những giảm được chi phí mà còn tăng năng suất và tìm ra nhiều giải pháp phục vụ trực tiếp cho nhu cầu của khách hàng.

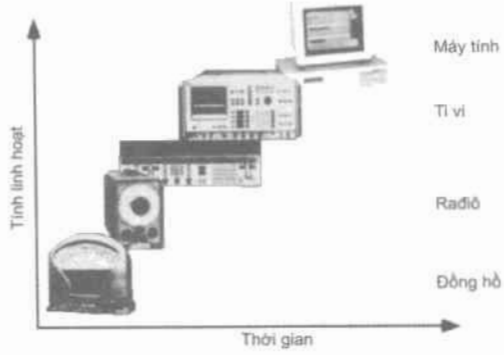
Các thiết bị thử nghiệm điện và điện tử luôn ứng dụng những công nghệ hiện đại đã được sử dụng rộng rãi. Bộ phận chuyển động bằng đá quý của đồng hồ ở thế kỷ 19 đã được áp dụng trong việc chế tạo các dụng cụ đo tương tự. Vào những năm 1930, khi tụ có điện dung biến đổi, biến trở, đèn chân không bắt đầu được sử dụng rộng rãi trong radio, các thiết bị đo lường điện tử đầu tiên sử dụng những bộ phận đó đã ra đời. Khi công nghệ hiển thị được cải tiến để ứng dụng trong những chiếc tivi đầu tiên thì máy hiện sóng và bộ phân tích bắt đầu áp dụng công nghệ tương tự để biểu diễn thông số đo lường cho người sử dụng (xem hình 48.1). Những bước đầu tiên này của hệ thống đo lường dựa trên máy tính đã gặp phải những thách thức đáng kể. Hệ thống đo lường dựa trên máy tính của những năm 1960 cần có các giao diện phần cứng và các hợp ngữ bậc thấp theo yêu cầu khách hàng. Sự phát triển của các chuẩn, chẳng hạn như sự ra đời của bus giao diện đa năng kết nối giữa thiết bị đo và máy tính, đã tạo nền tảng cho những cải tiến mang tính cách mạng trong sự phát triển và ứng dụng các thiết bị đo dựa trên máy tính.

Bằng việc sử dụng bus giao diện đa năng, các kỹ sư đã bắt đầu viết các chương trình, trước tiên dưới dạng ngôn ngữ BASIC, sau đó là ngôn ngữ dựa trên C và cuối cùng là môi trường phát triển giao diện đồ họa. Các chương trình này đã biến máy tính của họ thành những bộ điều khiển đo lường hiệu quả có khả năng lưu trữ dữ liệu. Vào những năm 1980, bộ số hoá và bo mạch thu thập dữ liệu đã trở thành các thiết bị thay thế được công nhận rộng rãi cho những thiết bị đo lường riêng lẻ đắt đỏ. Với sự kết hợp giữa phần cứng và phần mềm, các kỹ sư đã bắt đầu tạo ra "các thiết bị đo lường ảo".

Trong suốt những năm 1980 và 1990, ý tưởng về những thiết bị đo lường ảo đã được hưởng ứng rộng rãi hơn khi sức mạnh của máy tính để bàn được tăng lên nhanh chóng. Nhu cầu của người tiêu dùng và các doanh nghiệp về những CPU nhanh hơn, hiệu quả hơn, các ASIC có tính năng cao hơn và gọn nhẹ hơn, những ổ đĩa cứng có dung lượng lớn hơn và nhanh hơn, bus giao diện có tính năng cao hơn đã tạo điều kiện thuận lợi cho những người thiết kế các hệ thống đo lường dựa trên máy tính.

Những hệ thống đo lường ngày nay phần lớn bị ảnh hưởng bởi các cuộc cách mạng Internet và máy tính cá nhân. Hiện nay, máy tính cá nhân được trang bị những công cụ tính toán hữu hiệu có thể kết hợp với phần mềm để tạo ra một thiết bị đo lường khá phức tạp. Dữ liệu có thể dễ dàng

chuyển tải đến bất kỳ ai ở bất kỳ nơi nào trên thế giới có kết nối với thiết bị đo lường thông qua mạng Internet.



HÌNH 48.1 Lịch sử hệ thống đo lường

Với những phần mềm tiên tiến, tốc độ chuyển tải dữ liệu cũng như tốc độ xử lý nhanh, bằng cách sử dụng các thiết bị thu thập dữ liệu (DAQ) và những thiết bị đo lường khác máy tính có thể tái tạo đầu vào/đầu ra, chuẩn hoá tín hiệu, lưu trữ dữ liệu, số hoá và khả năng phân tích của những thiết bị đo truyền thống. Hơn nữa, những thiết bị dựa trên máy tính này liên tục có những cải tiến ở mức chi phí thấp hơn nhiều so với những thiết bị đo lường riêng lẻ truyền thống do toàn bộ công nghệ trong ngành công nghiệp máy tính liên tục phát triển. Với những công cụ dựa trên máy tính, các kỹ sư có thể cập nhật những hệ thống đo lường của họ chỉ đơn giản bằng cách mua một chiếc máy tính mới.

Để hiểu được tính linh hoạt của thiết bị đo lường dựa trên máy tính, cần xem xét kỹ hơn từng bộ phận cấu thành của hệ thống, trước tiên là các phần mềm.

48.1 Sức mạnh của phần mềm

Thiết bị đo lường dựa trên máy tính hiện đại đòi hỏi sự kết hợp phức tạp của phần mềm, từ driver của thiết bị đến phần mềm phát triển và phần mềm quản lý thử nghiệm tạo nên một tích hợp liền mạch nhằm đưa ra một giải pháp hoàn thiện cho các doanh nghiệp. Phần mềm là trung tâm của các hệ thống đo lường dựa trên máy tính. Chúng thể hiện tính linh hoạt và tính liên kết hoạt động nhằm cung cấp các giải pháp khác nhau cho mỗi ứng dụng. Mỗi nhà khoa học hay kỹ sư có thể sử dụng phần mềm cho từng ứng dụng thử nghiệm cụ thể để thỏa mãn các mục đích khác nhau. Khi sử dụng thiết bị đo truyền thống, các nhà cung cấp chỉ định chức năng, giao diện và các thông số của mỗi thiết bị đo. Những hạn chế này đã không còn phổ biến. Các nhà khoa học và các kỹ sư giờ đây có thể xác định và biến đổi tính năng của thiết bị đo bằng cách cập nhật phần mềm hoặc thay đổi các bộ phận phần cứng đơn lẻ. Giao diện và các thông số mà bộ điều khiển tương tác với chúng cũng có thể được thiết kế theo yêu cầu của khách hàng nhằm đáp ứng các nhu cầu cá nhân.

Phần mềm được sử dụng trong các hệ thống đo lường dựa trên máy tính ngày nay có ba cấp. Driver của thiết bị đo nằm ở cấp đầu tiên của cấu trúc phần mềm. Nó tương tác với phần cứng giúp truyền dữ liệu đo về máy tính. Chúng cũng bao gồm mã lệnh giúp điều khiển việc tạo lệnh và các hàm truyền thông cần thiết để trao đổi thông tin với các thiết bị đo lường đơn lẻ qua GPIB. Driver giúp các hệ thống thử nghiệm có khả năng duy trì hơn do chúng có thể dễ dàng được nâng cấp hoặc thay đổi mỗi khi có những thay đổi trong phần cứng.

Cấp thứ hai của cấu trúc phần mềm là phần mềm phát triển giúp xây dựng giao diện người dùng đồ họa (GUI – Graphical User Interface) cho một thiết bị đo lường cụ thể. Tính năng riêng biệt của thiết bị đo dựa trên máy tính được xác định tại lớp phần mềm ứng dụng. Việc thiết kế theo yêu cầu khách hàng ở cấp này phụ thuộc vào công cụ phần mềm được sử dụng. Một số phần mềm được viết cho một phần cứng nhất định, chức năng và giao diện được xác định ngay từ đầu. Hiện

nay có rất nhiều môi trường phát triển mở, chúng cho phép người sử dụng biến đổi hệ thống đo theo ý muốn của mình. National Instruments LabVIEW™, Measurement Studio™, Microsoft Visual Basic và Visual C++ là ví dụ cho những môi trường phát triển như vậy. Sử dụng chương trình lập trình dựa trên đồ họa hoặc ngôn ngữ dựa trên C truyền thống các kỹ sư có thể thiết kế những giao diện sao cho chúng chỉ bao gồm các nút bấm và đồ thị dùng cho việc phân tích ở cấp cao và hiển thị dữ liệu.

Do có rất nhiều sự lựa chọn khác nhau nên khi xem xét phần mềm ứng dụng điều quan trọng là phải nắm bắt được công dụng chính của phần mềm. Hãy lựa chọn một môi trường phát triển có thể làm tăng năng suất và đáp ứng đầy đủ các yêu cầu của ứng dụng đo lường. National Instruments LabVIEW™, Microsoft Visual Basic và Microsoft Visual C là những phần mềm được sử dụng nhiều nhất trong các ứng dụng đo lường. National Instruments LabVIEW là một môi trường phát triển đồ họa có những chức năng đặc biệt phục vụ cho nhiều ứng dụng đo lường và cho phép phát triển giao diện người dùng một cách nhanh chóng. Microsoft Visual Basic và Visual C là những ngôn ngữ dựa trên ký tự khá phổ biến trong tất cả các ứng dụng phần mềm.

Cấp thứ ba của cấu trúc phần mềm điển hình được cấu thành bởi chương trình quản lý các thử nghiệm. Chúng hỗ trợ nhiều mã thử nghiệm phổ biến và có thể tạo ra các báo cáo thông qua các chương trình phần mềm chủ đạo như Microsoft Excel. Những chương trình thử nghiệm hữu ích này có thể làm giảm chi phí thử nghiệm sản phẩm và cho phép giải phóng các kỹ sư khỏi những công việc đó. Họ không phải sử dụng thời gian quý báu của mình để xây dựng phần mềm có nhiệm vụ thực hiện các chức năng quản lý thử nghiệm.

48.2 Số hóa thế giới tương tự

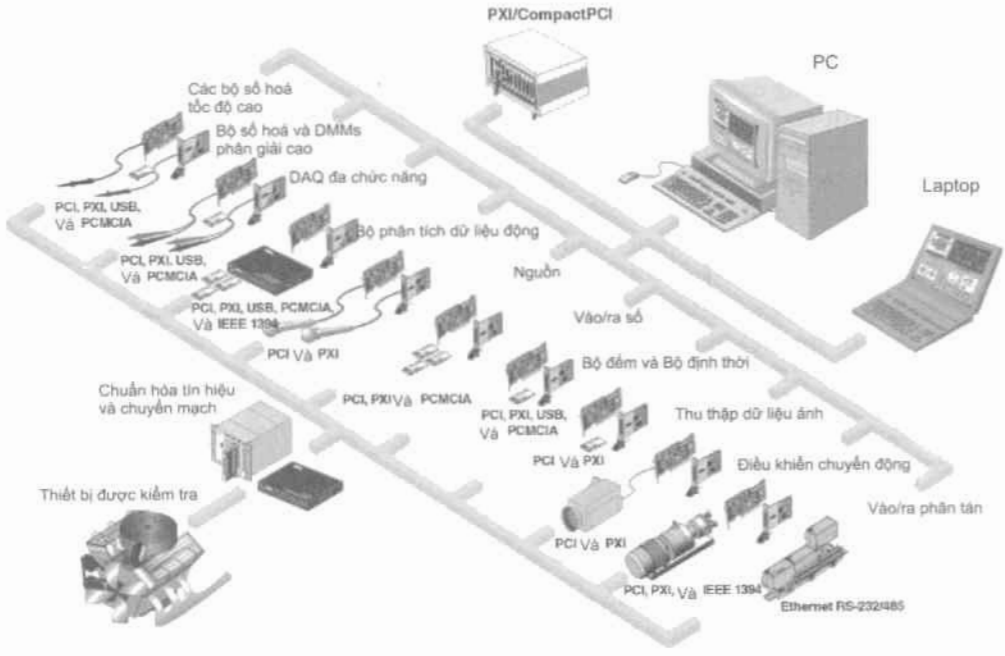
Trước khi phần mềm có thể bắt đầu phân tích các dữ liệu đo lường, nó cần phải được chuyển đổi từ thế giới tương tự sang tư duy máy tính. Để làm được điều này cần có nhiều loại phần cứng có khả năng số hoá các tín hiệu tương tự. Như đã giải thích ở trên, khi thu nhận những tín hiệu từ phần cứng đo lường riêng lẻ, driver của thiết bị đo sắp xếp các dữ liệu sao cho phần mềm phát triển có thể thực hiện thêm cấp phân tích và biểu diễn dữ liệu. Tuy nhiên, các loại phần cứng khác như thiết bị đo lường dạng plug–in, bộ điều khiển logic khả trình (PLC – Programmable Logic Controller), hệ thống vào/ra phân tán, những thiết bị hỗ trợ USB và truyền thông nối tiếp cũng có thể thu nhận tín hiệu bằng cách sử dụng phần mềm driver của chúng (xem Hình 48.2).

Như đã nêu ở trên, với công nghệ ngày nay, phần cứng dựa trên máy tính có thể thu nhận tín hiệu ở tốc độ bằng hoặc nhanh hơn tốc độ của những thiết bị đo lường riêng lẻ. Các thiết bị đo lường dựa trên máy tính có thể số hoá những tín hiệu này với tốc độ và độ phân giải có thể cạnh tranh được với những thiết bị đo lường truyền thống. Tuy nhiên, với công nghệ dựa trên máy tính, việc tích hợp phần cứng với phần mềm tạo ra một số khác biệt. Nhờ sự kết hợp chặt chẽ này, các kỹ sư có thể tạo ra thiết bị đo lường đáp ứng được những yêu cầu của mình. Họ định nghĩa những tính năng của thiết bị thay cho các nhà cung cấp. Hơn nữa, với những thiết bị dựa trên máy tính này, các kỹ sư có thể dễ dàng truyền dữ liệu từ những thiết bị đo thông qua mạng Internet bằng cách sử dụng Ethernet và các công nghệ khác.

Tính linh hoạt này sẽ mang lại nhiều sự lựa chọn. Khi lựa chọn giữa những loại phần cứng đo lường khác nhau, người sử dụng phải lưu ý mục đích và khả năng ứng dụng của chúng. Việc thu nhận các tín hiệu tốc độ cao, lên tới 100 MHz chẳng hạn, hoặc các tín hiệu thay đổi nhanh như phân tích âm thanh hoặc dao động, có thể được thực hiện thông qua bộ số hoá và bộ phân tích tín hiệu động học được nối với máy tính thông qua bus PCI hoặc PXI/Compact PCI.

Các bo mạch thu thập dữ liệu kiểu plug-in có lợi thế thu nhận dữ liệu trực tiếp vào bộ nhớ máy tính với tốc độ rất cao. Một nhược điểm của các bo mạch trên là bản thân chúng lại không cung cấp một giao diện công nghiệp cho tất cả các bộ chuyển đổi và tín hiệu. Ví dụ, các cặp nhiệt điện sinh ra những tín hiệu rất nhỏ, do đó tín hiệu của chúng cần được tuyến tính hoá và cần bù mối nối lạnh. Các ứng dụng khác có thể yêu cầu tính năng chuẩn hoá tín hiệu, ví dụ như sự dồn kênh để tăng số lượng kênh, lọc để loại nhiễu, cách ly để bảo vệ hệ thống khỏi các tín hiệu điện áp cao,

hoặc khuyếch đại những tín hiệu điện áp nhỏ. Một thiết bị chuẩn hoá tín hiệu chuyên dụng thông thường được kết hợp với một bo mạch thu thập dữ liệu để đáp ứng nhu cầu kết nối máy tính với thế giới bên ngoài.



HÌNH 48.2

Những ứng dụng mà ở đó máy tính được đặt ngay sát thiết bị đo lường là không hợp lý. Các thiết bị vào/ra phân tán có thể được dùng cho những loại ứng dụng này. Phần cứng số hoá hoặc chuẩn hoá tín hiệu được nối với máy tính thông qua truyền thông nối tiếp, RF, hoặc Ethernet.

Rất nhiều phép đo có thể được đưa vào máy tính và được gửi từ máy tính trong các ứng dụng đo lường và tự động. Các thiết bị thu thập hình ảnh và bộ điều khiển chuyển động cũng có thể được tích hợp vào trong hầu hết các ứng dụng đo lường.

48.3 Triển vọng

Ngày nay, các phần mềm kết hợp với sức mạnh của Internet và các giao thức truyền thông tiêu chuẩn cho phép phát triển các ứng dụng đo lường nối mạng đơn giản. Việc thu thập dữ liệu được thực hiện qua mạng hoặc qua một trình duyệt Web.

Việc tích hợp này mở ra những cấp mới của cấu trúc phần mềm. Các kỹ sư và các nhà khoa học ngày càng hiểu rằng thu nhận và phân tích dữ liệu chỉ là một phần của quá trình. Dữ liệu của những thử nghiệm này có thể được lưu giữ ở cơ sở dữ liệu trung tâm có thể được chia sẻ với các đồng nghiệp. Việc chia sẻ và quản lý dữ liệu đem đến hiệu quả không ngờ. Phân tích, so sánh các thử nghiệm và kết hợp dữ liệu có thể đem lại kiến thức mới và những quyết định đúng đắn.

Thiết lập những hệ thống quản lý này đòi hỏi việc thu thập, lưu trữ dữ liệu, báo cáo và phân tích hệ thống. Tất cả các công việc có thể được thực hiện trên máy tính.

Những thay đổi tương tự cũng diễn ra với phần cứng dựa trên máy tính. Bộ vi xử lý nhúng giờ đây có thể cho chạy các hệ thống điều hành thời gian thực có chức năng thực hiện những nhiệm vụ của CPU máy tính một cách độc lập.

Các ứng dụng thời gian thực có thể được thực hiện trên máy tính để bàn và trong môi trường Microsoft Windows. Nó mở ra một loạt những ứng dụng cho các kỹ sư, giúp họ tránh được rắc rối của việc lập trình thời gian thực truyền thống.

Ngành công nghiệp đo lường luôn thúc đẩy các công nghệ sẵn có để tạo ra những thiết bị thử nghiệm hiệu quả. Các công nghệ dựa trên máy tính và Internet cho phép tạo ra nhiều giải pháp phù hợp với khách hàng hơn, có liên quan mật thiết đến cuộc sống hàng ngày hơn so với trước kia.

49

Thiết kế và phát triển phần mềm

49.1 Khái niệm phần mềm 49-2
49.2 Bản chất của kỹ thuật phần mềm 49-5
49.3 Sự phát triển trước thực tế. 49-9
49.4 Thực nghiệm với công nghệ DBTF 49-16
49.5 Kết luận 49-17

Margaret H. Hamilton
Hamilton Technologies, Inc.

Một hệ thống dựa trên phần mềm có thể được so sánh với một thực thể sinh học mà người ta thường gọi là một xã hội hữu cơ. Bao gồm phần mềm, phần cứng, phần con người, khả năng nối kết (ví dụ như Internet) của chúng, và yêu cầu tất cả phải tồn tại, xã hội hữu cơ silicon bản thân nó là một phần của một xã hội hữu cơ lớn hơn – ví dụ một hệ thống y tế bao gồm bệnh nhân, thuốc men, các công ty dược phẩm, bác sĩ, và các trung tâm y tế; một cuộc thám hiểm vũ trụ bao gồm tàu vũ trụ, các định luật vũ trụ, điều khiển nhiệm vụ, và các phi hành gia; một hệ thống nghiên cứu gien bao gồm các tổ chức tài trợ, các quỹ tài trợ, các nhà nghiên cứu, các đề tài nghiên cứu, và gien; một hệ thống tài chính bao gồm các nhà đầu tư, vốn, các chính sách, các tổ chức tài chính, thị trường chứng khoán và tình trạng nền kinh thế thế giới; hoặc nó có thể chỉ là bản thân hoạt động thực tiễn.

Cho dù hoạt động thực tiễn mang tính chất chính phủ, khoa học hay thương mại, hệ thống dựa trên phần mềm, giống như một hệ sinh học, phải phát triển và thích nghi để đáp ứng một cách nhanh chóng những nhu cầu đang biến đổi. Và giống như các tổ chức khác, hoạt động thực tiễn có cả cơ sở vật chất và các chính sách vận hành giúp hướng dẫn và đôi khi ép buộc hướng đi và tốc độ phát triển của nó để nó có thể tồn tại được mà không bị chệch hướng.

So với một xã hội hữu cơ phải trải qua rất nhiều thế hệ mới có thể đạt được một thay đổi thậm chí khả nhỏ bé mang tính di truyền, thì phần mềm có thể được biến đổi ngay lập tức. Điều này khiến nó trở nên ưu việt hơn rất nhiều so với thực thể sinh học về khả năng thích nghi tiến hoá của nó. Tính liên tục của quy luật kinh doanh và/hoặc cơ sở vật chất đã làm nảy sinh một vấn đề khá căng thẳng mang tính tự nhiên giữa việc "Phần mềm có thể biến đổi nhanh như thế nào" với việc "toàn bộ hệ thống có thể chấp nhận sự thay đổi đó nhanh ra sao". Phần mềm, bộ não của xã hội hữu cơ silicon, điều khiển các hoạt động của toàn bộ thực thể. Tuy nhiên, cũng phải ghi nhớ rằng chính bản thân con người đã tạo ra phần mềm.

Trong chương này, chúng ta sẽ bàn về các nguyên lý của phần mềm, nó là gì và nó được phát triển như thế nào, cũng như những nhận thức về kỹ thuật phần mềm, vốn được coi là những phương pháp biến các ý tưởng thành sản phẩm phần mềm cụ thể.

49.1 Khái niệm phần mềm

Phần mềm là hiện thân của các quá trình lôgic bất kể nhằm phục vụ mục đích kinh doanh hay điều khiển các thiết bị vật lý. Bản chất của phần mềm giống như một quá trình có thể được ứng dụng rất rộng rãi, khi mô hình hoá các tổ chức phức tạp hoặc hẹp hơn khi thực hiện một thuật toán số rời rạc. Phần mềm có khả năng được ứng dụng khá rộng rãi và được thiết kế tốt để có thể được sử dụng trong thời gian dài.

Trong khi một số người định nghĩa phần mềm chỉ là mã do một ngôn ngữ lập trình tạo ra trong quá trình biên dịch, thì một định nghĩa rộng và chính xác hơn thể hiện các yêu cầu đặt ra, các thông số kỹ thuật, thiết kế, các bản liệt kê chương trình, tư liệu, quy trình, nguyên tắc, các thông số đo lường và dữ liệu cũng như các công cụ được dùng để tạo ra, kiểm tra, tối ưu hoá và thực thi phần mềm.

Việc có nhiều định nghĩa về phần mềm là kết quả trực tiếp của bản thân quá trình phát triển khá phức tạp của phần mềm. Một nghiên cứu vào năm 1991 do Viện kỹ thuật phần mềm (SEI – Software Engineering Institude) [1] tiến hành đã khuếch đại điều này hơn cả một vấn đề đáng kinh ngạc. SEI đã phát triển một phương pháp để phân loại "khả năng thuần thục phần mềm" của một tổ chức thành một trong năm cấp độ, từ cấp độ thứ 1 (không có kiến thức căn bản về quy trình phần mềm), đến cấp độ 5 (cấp độ tối ưu). Ở cấp độ 5, các phương pháp, quy trình song hành với việc tập trung cải tiến liên tục độ tin cậy của phần mềm. Kết quả của nghiên cứu này đã chỉ ra rằng 86% tổ chức được điều tra ở Mỹ nằm ở cấp độ 1. Ở cấp độ này, các thuật ngữ như "phi thể thức", "phụ thuộc vào nhân vật chính", và "lộn xộn" thường được sử dụng. Và với tính phức tạp của các ứng dụng dựa trên Internet ngày nay, sẽ chẳng có gì đáng ngạc nhiên khi phần trăm các tổ chức thuộc mức 1 ngày càng gia tăng.

Để lập lại trật tự từ những lộn xộn này cần phải có một sự am hiểu tường tận về các bộ phận cấu thành phần mềm cũng như quá trình phát triển của nó. Vay mượn từ thế giới khoa học tự nhiên, tự mục đích là một cái gì đó khá phức tạp xuất hiện khi một số lượng lớn các vật thể đơn giản được tập hợp lại với nhau. Ví dụ, một phân tử nước sẽ không có gì đáng chú ý khi nó thiếu đi sự vận động. Nhưng nếu đổ một tập hợp những phân tử nước này vào trong một cái cốc thì sẽ có một vòng tròn gợn sóng trên bề mặt của nước. Nếu nhiều phân tử nước này được hợp lại thì biển sẽ ra đời. Phần mềm cũng tương tự như vậy. Dòng mã bản thân nó chỉ là một phần khá đơn giản. Nhưng nếu kết hợp nhiều dòng mã lại với nhau sẽ cho ra đời một chương trình phần mềm khá tinh vi. Nếu tập hợp nhiều chương trình phần mềm lại với nhau sẽ cho ra đời một hệ thống có thể giúp đưa con người lên mặt trăng.

Mặc dù toàn thể thì bao giờ cũng lớn hơn tổng các bộ phận của nó, nhưng người ta vẫn phải hiểu được những bộ phận của nó nếu toàn thể vận hành theo một trật tự và có điều khiển. Giống như một thực thể vật chất, phần mềm cũng có thể bị "hao mòn" do kết quả của sự duy trì bảo dưỡng, những thay đổi trong hệ thống cơ sở và những cải tiến nhằm đáp ứng nhu cầu của người sử dụng. Entropi là một hiện tượng quan trọng trong phần mềm đặc biệt thường xảy ra đối với các tổ chức nằm ở cấp 1.

Phần mềm ở cấp độ lập trình thấp nhất được gọi là mã nguồn. Nó hoàn toàn khác với mã có thể thực thi được (tức là, một loại mã có thể được chạy bởi phần cứng để thực hiện một hay nhiều chức năng được định rõ) ở chỗ phần mềm được viết bằng một hoặc nhiều ngôn ngữ lập trình và bản thân nó không thể được chạy bởi phần cứng. Ngôn ngữ lập trình là một tập hợp các từ, chữ, số và các thuật nhớ viết tắt được quy định bởi một cú pháp rõ ràng, được dùng để mô tả chương trình cho máy tính. Có rất nhiều ngôn ngữ lập trình, rất nhiều trong số đó được sửa đổi sao cho phù hợp với một ứng dụng cụ thể nào đó. C là một trong những ngôn ngữ lập trình phổ biến ngày nay, được ứng dụng trong môi trường thương mại cũng như kỹ thuật. Trong khi đó những ngôn ngữ hướng đối tượng như C++ [2] và Smalltalk đã thích nghi được trong cả hai môi trường này. Gần đây nhất, Java [3] cũng đã có được khả năng thích nghi đó. Trên thực tế, nó đã trở thành ngôn ngữ được lựa chọn để dùng trong các ứng dụng dựa trên Internet. Mới đây, trong kỹ thuật, người ta thường sử dụng các ngôn ngữ lập trình như FORTRAN, HAL (hoặc HAL/s) cho các ứng dụng không gian của NASA, và Ada cho các ứng dụng của chính phủ. Đối với các

ứng dụng kinh tế thương mại, người ta lại có xu hướng sử dụng COBOL (COmmon Business Oriented Language). Nhìn chung, không tổ chức nào có những quy định cụ thể ngoại trừ những quy định liên quan đến những gì phổ biến nhất. Những quy định này quyết định loại ngôn ngữ nào được sử dụng.

Ngôn ngữ lập trình, có thể là C++, Java, Visual BASIC, C, FORTRAN, HAL/s, COBOL, hoặc một vài ngôn ngữ khác, cho phép khả năng mã hoá những cấu trúc logic như:

- *Giao diện người dùng.* Cung cấp một cơ chế mà nhờ đó người sử dụng có thể nhập, xem, quản lý và truy cập thông tin từ các hệ thống máy tính của một tổ chức. Các nghiên cứu đã chỉ ra rằng hiệu quả sẽ tăng một cách đáng kể khi có các giao diện người dùng trực quan. Cũng giống như các giao diện người dùng đồ họa (GUIs – Graphical User Interfaces), mỗi hệ thống vận hành cho thấy một sự biến đổi của riêng nó. Một số tiêu chuẩn đồ họa phổ biến là Motif dùng trong các hệ thống UNIX và Microsoft Windows áp dụng cho các hệ thống dựa trên PC.
- *Tính toán mô hình.* Thực hiện các phép tính hoặc thuật toán (các quy trình từng bước để giải quyết một vấn đề) của một chương trình chẳng hạn như điều khiển quá trình, tính bảng lương, hoặc bộ lọc Kalman.
- *Điều khiển chương trình.* Thực hiện việc điều khiển dưới dạng so sánh, phân nhánh, gọi các chương trình khác và lặp lại để thực hiện logic của chương trình.
- *Xử lý thông tin.* Có một vài loại xử lý thông tin. Xử lý thông báo trợ giúp là quá trình chương trình phản hồi lại những yêu cầu trợ giúp của người dùng. Xử lý thông điệp lỗi là khả năng tự động của chương trình trong việc thông báo và sau đó là khôi phục lại từ một lỗi nào đó xảy ra trong quá trình nhập dữ liệu, lấy dữ liệu, tính toán và trao đổi thông tin... Trong những môi trường phát triển hướng dữ liệu, việc xử lý thông tin ám chỉ khả năng chuyển tải thông tin của các đối tượng chương trình sang các đối tượng chương trình khác.
- *Dữ liệu hoạt động.* Các chương trình thường lưu trữ dữ liệu trong một cấu trúc dữ liệu. Dữ liệu có thể được truyền giữa các cấu trúc dữ liệu trong cùng một chương trình, tức là truyền từ file hoặc một cơ sở dữ liệu bên ngoài đến một cấu trúc dữ liệu bên trong hoặc từ người dùng đến cấu trúc dữ liệu bên trong của một chương trình. Hoặc, dữ liệu có thể được truyền từ một cấu trúc dữ liệu bên trong đến một cơ sở dữ liệu hoặc thậm chí đến giao diện người dùng. Phân loại và định dạng là các thao tác truyền dữ liệu được dùng để chuẩn bị dữ liệu cho các thao tác tiếp theo.
- *Cơ sở dữ liệu.* Là một tập hợp dữ liệu (đối tượng) hoặc thông tin về một chủ đề hoặc các chủ đề liên quan, hoặc về một hệ thống (ví dụ, động cơ của một chiếc xe tải hoặc phòng nhân sự trong một tổ chức). Một cơ sở dữ liệu có thể bao gồm các đối tượng, ví dụ như các mẫu đơn, các báo cáo hoặc tập hợp các tài liệu về một hệ thống (ví dụ như thông tin của phòng nhân sự về các nhân viên trong một công ty). Một cơ sở dữ liệu được tổ chức sao cho người sử dụng có thể dễ dàng truy cập. Dữ liệu của nó đại diện cho các sự kiện, khái niệm, chỉ dẫn sao cho phù hợp với việc xử lý bằng máy tính. Nó có thể được trình bày, cập nhật, truy cập hoặc in ấn, và các báo cáo có thể ra đời từ đó. Một cơ sở dữ liệu có thể tổ chức dữ liệu theo vài cách bao gồm dưới dạng quan hệ, phân cấp, mạng hoặc hướng đối tượng.
- *Khai báo dữ liệu.* Mô tả dữ liệu và cấu trúc dữ liệu cho một chương trình. Ví dụ như một cấu trúc dữ liệu xác định với kiểu của nó (như dữ liệu về một nhân viên nhất định có thể thuộc kiểu người).
- *Đối tượng.* Một người, vật hoặc nơi chốn có thể là cụ thể hoặc trừu tượng. Một đối tượng bao gồm nhiều đối tượng (hoặc dữ liệu) cơ bản khác và một tập hợp các thao tác để điều khiển các đối tượng (hoặc dữ liệu). Khi được đưa vào cuộc sống, nó nhận biết được các sự vật (được gọi là các thuộc tính) và có thể tác động lên các sự vật (nhằm thay đổi bản thân nó hoặc tương tác với các dữ liệu khác). Chẳng hạn, trong một hệ

rôbốt, đối tượng robot có thể bao gồm các chức năng để có thể di chuyển đến vị trí ổn định, trong khi nó phối hợp với một robot khác để truyền đối tượng khác cho nhau. Các đối tượng có thể giao tiếp với nhau thông qua các phương tiện truyền thông (ví dụ như truyền thông điệp, sóng radio, Internet).

- *Thời gian thực.* Là một hệ thống phần mềm có thể đáp ứng được các yêu cầu chặt chẽ về thời gian. Độ chính xác của phần mềm phụ thuộc vào các kết quả tính toán, cũng như thời gian có được các kết quả đó. Các hệ thống thời gian thực có thể có các yêu cầu khác nhau như thực hiện một nhiệm vụ trong một thời gian cụ thể và xử lý dữ liệu liên quan đến một quá trình khác bên ngoài máy tính. Các ứng dụng như xử lý các giao dịch, khoa học điện tử, quản lý văn phòng, các hệ thống ô tô, và các trò chơi video là những ví dụ về các hệ thống thời gian thực.
- *Phân tán.* Trong bất kỳ hệ thống nào cũng có các quá trình độc lập có thể phối hợp với nhau. Mô hình khách/chủ là một trong những dạng điển hình nhất của phân tán đang được sử dụng ngày nay. Trong mô hình này, máy khách sẽ khởi tạo hoạt động phân tán và máy chủ sẽ thực hiện hoạt động đó.
- *Mô phỏng.* Là việc trình bày các đặc trưng có lựa chọn về ứng xử của một hệ thống vật chất hoặc trừu tượng bởi một hệ thống khác. Ví dụ, một chương trình phần mềm có thể mô phỏng máy bay, một tổ chức hoặc một chương trình phần mềm khác.
- *Tài liệu.* Bao gồm mô tả các yêu cầu, đặc điểm kỹ thuật và thiết kế cũng như các tài liệu được viết hoặc được tạo ra. Chúng mô tả cách thức mỗi chương trình trong một hệ thống lớn hơn được vận hành và được sử dụng như thế nào.
- *Các công cụ.* Các chương trình phần mềm thường được dùng để thiết kế, phát triển, thử nghiệm, phân tích hoặc bảo trì các thiết kế hệ thống hay một chương trình phần mềm khác và tài liệu của nó. Chúng bao gồm các bộ tạo mã, trình biên dịch, trình soạn thảo, các hệ thống quản trị cơ sở dữ liệu (DBMS – Database Management Systems), trình xây dựng GUI, trình gỡ rối, các hệ điều hành và các công cụ phát triển phần mềm và kỹ thuật hệ thống. Vào những năm 1990, người ta thường nhắc tới chúng như những công cụ kỹ thuật phần mềm có sự trợ giúp của máy tính (CASE – Computer-Aided Software Engineering Tools) và ngày nay chúng được nhắc tới như môi trường phát triển vòng đời, bao gồm một tập hợp các công cụ, một vài trong số đó đã được nhắc tới ở trên.

Giống như một đơn vị cơ sở, một dòng mã có thể được kết hợp với các dòng mã khác để hình thành nên nhiều thứ. Trong một môi trường phần mềm truyền thống, rất nhiều dòng mã hình thành nên một chương trình, đôi khi được gọi là một trình ứng dụng hoặc chỉ là một ứng dụng đơn giản. Tuy nhiên, chính bản thân các dòng mã nguồn lại không thể thực thi được. Trước hết, mã nguồn phải được chạy thông qua một trình biên dịch để tạo ra một mã đối tượng. Sau đó, mã đối tượng được cho chạy qua một cầu nối được dùng để xây dựng mã có thể thực thi. Trình biên dịch chính là các chương trình. Chức năng của chúng có hiệu quả gấp đôi. Trước hết, trình biên dịch sẽ kiểm tra mã nguồn để tìm các lỗi cú pháp và sau đó nếu không phát hiện ra lỗi thì sẽ tạo ra các mã đối tượng cho một hệ điều hành nhất định. UNIX, Linux (một sản phẩm của UNIX), và NT là những ví dụ về những hệ điều hành. Một hệ điều hành có thể được coi như một chương trình giám sát điều khiển các chương trình ứng dụng nằm trong sự kiểm soát của nó. Do các hệ điều hành (cũng như các cấu trúc máy tính) có thể khác nhau nên mã đối tượng được tạo từ mã nguồn biên dịch cho một hệ điều hành không thể được thực thi dưới dạng hệ điều hành khác mà không có sự biên dịch lại.

Việc giải quyết một vấn đề kinh doanh hay kỹ thuật phức tạp thường đòi hỏi phải có nhiều hơn một chương trình. Một hoặc nhiều chương trình cùng chạy để giải quyết một vấn đề chung được gọi là một hệ thống. Kỹ thuật phát triển hướng đối tượng hiện đại hơn đã bỏ qua khái niệm về chương trình và thay nó bằng khái niệm hướng phân loại của một đối tượng.

Khi một chương trình được coi như một tập hợp của nhiều mã lệnh, thực hiện nhiều chức năng nhằm giải quyết một vấn đề mà không cần cân nhắc đến ranh giới đối tượng, thì một đối tượng được liên kết với mã để giải quyết một tập hợp chức năng nhất định. Bằng cách kết hợp các đối

tượng, giống như các phân tử, có thể tạo ra những hệ thống có tổ chức hơn những hệ thống được tạo ra bởi những phương pháp truyền thống. Phần mềm được phát triển nhanh hơn và ít bị lỗi hơn. Do các đối tượng có thể được tái sử dụng, nên một khi được kiểm tra và sử dụng, chúng có thể được đưa vào thư viện cho những chuyên viên thiết kế phần mềm khác tái sử dụng. Càng có nhiều đối tượng trong thư viện, việc phát triển những hệ thống mới càng trở nên nhanh chóng và dễ dàng hơn. Về mặt lý thuyết, do các đối tượng đang được tái sử dụng đã được bảo đảm (chẳng hạn chúng được kiểm tra và không bị lỗi), nên ít có khả năng những hệ thống hướng đối tượng có những kiếm khuyết lớn.

Quá trình viết phần mềm và/hoặc đối tượng được coi là quá trình phát triển phần mềm, hoặc kỹ thuật phần mềm. Nó gồm có một loạt các bước hay còn gọi là giai đoạn, gộp chung lại người ta gọi là vòng đời phát triển. Các giai đoạn bao gồm (ở mức tối thiểu cơ bản): giai đoạn phân tích hoặc yêu cầu, ở giai đoạn này các vấn đề thực tiễn được phân tích và tìm ra nguyên nhân; giai đoạn mô tả kỹ thuật, ở đó các quyết định được đưa ra liên quan đến việc các yêu cầu được đáp ứng như thế nào (ví dụ quyết định chức năng nào là của phần mềm, chức năng nào là của phần cứng); giai đoạn thiết kế, đây là giai đoạn mà mọi thứ từ GUI đến cơ sở dữ liệu đầu ra đều được thiết kế hoặc lựa chọn như một phần của bản thiết kế; giai đoạn thực thi và lập trình, với một hoặc nhiều công cụ được sử dụng để viết và/hoặc tạo mã; giai đoạn kiểm tra (gỡ lỗi), ở đó mã được kiểm tra và lỗi trong chương trình được tìm ra và sửa chữa; giai đoạn cài đặt là giai đoạn các hệ thống được đưa vào sản xuất; và giai đoạn bảo dưỡng là giai đoạn sửa chữa hệ thống. Tuy nhiên, những người khác nhau lại phát triển các hệ thống theo các cách khác nhau. Những mô hình khác nhau này làm phát sinh các quan điểm trái ngược nhau về kỹ thuật phần mềm.

49.2 Bản chất của kỹ thuật phần mềm

Các kỹ sư thường sử dụng thuật ngữ "kỹ thuật hệ thống" để nói đến nhiệm vụ xác định, thiết kế và mô phỏng các hệ thống phi phần mềm, ví dụ như một cây cầu hoặc một thiết bị điện tử. Mặc dù phần mềm có thể được dùng cho mục đích mô phỏng, nhưng nó lại là một phần của quá trình kỹ thuật hệ thống. Mặt khác, kỹ thuật phần mềm có liên quan đến việc sản xuất phần mềm.

Trong ngành công nghiệp của những năm 1970, các chuyên gia đã bắt đầu nhận thấy rằng chi phí sản xuất các hệ thống quy mô lớn đang gia tăng với tốc độ cao và nhiều dự án đã thất bại hoặc cho ra đời những sản phẩm không đáng tin cậy. Trong cuộc khủng hoảng phần mềm, có rất nhiều vấn đề nảy sinh, trong đó quan trọng nhất là:

- *Hiệu quả của nhà lập trình.* Trong chính phủ vào những năm 1980, một chuyên viên thiết kế sử dụng C, trung bình có thể tạo ra được 10 dòng mã lệnh một ngày (một chuyên viên thiết kế trong một tổ chức thương mại trung bình có thể tạo ra được 30 dòng mã mỗi tháng); ngày nay, trong chính phủ con số đó chỉ còn khoảng từ 2 đến 5 dòng mã mỗi ngày trong khi đó nhu cầu thực tế lại cao hơn rất nhiều, khoảng độ vài đơn đặt hàng khối lượng lớn mỗi tháng, và kết quả là các đơn đặt hàng bị dồn ứ lại. Hiệu quả của nhà lập trình phụ thuộc vào nhiều yếu tố - từ sự thuần thục đến sự phức tạp của một vấn đề được mã hoá và độ lớn của chương trình được tạo ra. Do bản thân kỹ thuật phần mềm có rất nhiều mô hình khác nhau, nên việc đo lường phần mềm cũng có nhiều mô hình. Ngày nay, các công thức đo lường rất phức tạp và cần được cân nhắc, xem xét đến các yếu tố: chi phí, thời gian thương mại hoá sản phẩm, hiệu quả của những dự án trước đó, truyền thông dữ liệu, chức năng phân tán, khả năng hoạt động, cấu hình được dùng nhiều, tốc độ truyền, nhập dữ liệu trực tuyến, hiệu quả sử dụng, cập nhật trực tuyến, quá trình xử lý phức tạp, khả năng tái sử dụng, lắp đặt, vận hành và các điểm vận hành.
- *Chi phí loại bỏ lỗi.* Những biến số giống nhau, làm ảnh hưởng đến hiệu quả của nhà lập trình, cũng sẽ gây ảnh hưởng đến chi phí "gỡ lỗi" chương trình và/hoặc đối tượng được tạo ra bởi các nhà lập trình. Người ta quan sát thấy rằng việc kiểm tra và sửa chữa các chương trình tiêu tốn phần lớn sức lực.

- *Môi trường phát triển.* Các công cụ phát triển và thực tiễn phát triển có ảnh hưởng lớn đến số lượng cũng như chất lượng phần mềm. Hầu hết những môi trường thiết kế và lập trình ngày nay chỉ là một phần nhỏ trong việc phát triển một hệ thống hoàn thiện. Những môi trường phát triển vòng đời là những ví dụ rõ nét cho hiện tượng này. Hầu hết những công cụ này có thể được coi hoặc là như phần trên của vòng đời (ví dụ chúng thực hiện việc phân tích và thiết kế) hoặc như là như phần dưới của vòng đời (chẳng hạn chúng thực hiện việc tạo mã). Trên thị trường có rất ít công cụ tích hợp (nghĩa là chúng có thể thực hiện cả chức năng bên trên cũng như dưới). Thậm chí rất ít công cụ có chức năng mô phỏng, kiểm tra rất hiếm khi các công cụ tích hợp thiết kế hệ thống với phát triển phần mềm.
- *Phát triển GUI.* Phát triển GUI là một quá trình khó khăn và khá tốn kém trừ phi có sự hỗ trợ của các công cụ thích hợp. Sự phát triển của hệ thống từ một môi trường dựa trên máy chủ đến máy trạm và/hoặc PC đã cho thấy sự gia nhập thị trường của vô số các chương trình phát triển GUI. Tuy nhiên, đa phần những công cụ dựa trên GUI này không có khả năng phát triển toàn bộ hệ thống (ví dụ, thành phần xử lý chứ không chỉ đơn thuần là ngoại vi). Kết quả là sẽ cho ra đời những hệ thống dở dang và có xu hướng bị lỗi. Để mang lại hiệu quả, những chuyên viên thiết kế GUI cần phải được tiếp cận với môi trường phát triển phần mềm.

Do những vấn đề nêu trên mà hầu hết những hệ thống ngày nay đều yêu cầu tập trung nhiều nguồn lực cho việc duy trì bảo dưỡng hơn là cho việc phát triển từ đầu chính hệ thống đó. Lientz và Swanson [4] đã chứng minh rằng trên thực tế vấn đề này còn nghiêm trọng hơn so với những năm 1970. Quá trình phát triển phần mềm thực sự rất phức tạp và Jones [5] đã tái hiện lại một cách đầy đủ những hạn chế về những gì một nhóm các kỹ sư phần mềm với lượng thời gian, kinh phí và khả năng có hạn có thể sản xuất được.

Về cơ bản, đã có nhiều mô hình kỹ thuật phần mềm cố gắng giải quyết những nguyên nhân làm giảm hiệu suất và chất lượng. Đáng tiếc là những mô hình hiện nay chỉ giải quyết những vấn đề mang tính triệu chứng chứ không đi vào tận gốc rễ của vấn đề. Trên thực tế, bản thân kỹ thuật phần mềm lại bị phụ thuộc rất nhiều vào cả phần cứng lẫn phần mềm cũng như môi trường thực tế của chúng [6].

Mạng lưới thuần thục quá trình của SEI đã chỉ rõ nguồn gốc của hầu hết những vấn đề phát sinh trong quá trình phát triển phần mềm hiện nay. Việc có tới 86% các tổ chức được điều tra vẫn còn nằm ở mức phi thể thức hoặc chưa ổn định cho thấy chỉ có một số ít các tổ chức (14%) có thể thích nghi được với bất kỳ giai đoạn phát triển nào của kỹ thuật phần mềm. Nói một cách đơn giản, 86% các tổ chức đối phó với một vấn đề thực tiễn chỉ đơn giản bằng cách viết mã. Nếu họ chỉ áp dụng một nguyên tắc của kỹ thuật phần mềm, thì rất có khả năng nó sẽ không còn phù hợp với những yêu cầu của môi trường thực tiễn luôn luôn vận động và phát triển.

Vào những năm 1970, "hệ phương pháp cấu trúc - structured methodology" trở nên khá phổ biến. Mặc dù có những biến đổi về chủ đề (ví dụ, các phiên bản khác nhau của kỹ thuật cấu trúc bao gồm phương pháp nổi tiếng của Gane-Sarson và Yourdon) nhưng chủ yếu nó cung cấp cho chúng ta một phương pháp xây dựng phần mềm nhằm phát triển các hệ thống khả dụng trong kỷ nguyên máy tính xử lý song song. Ngày nay, các hệ thống trực tuyến lại chiếm vị trí chủ đạo còn GUI lại trở thành điều không tưởng khi bức tường Berlin sụp đổ.

Mặc dù đã trải qua một thời gian dài phát triển và ngày nay phần cứng đã mạnh hơn gấp nghìn lần so với khi kỹ thuật cấu trúc ra đời, nhưng công nghệ này vẫn tồn tại. Và nó đã tồn tại bất chấp thực tế là những người phát triển nó đã chuyển sang nghiên cứu những mô hình có khả năng thích ứng cao hơn, và những môi trường kỹ thuật hệ thống cùng với những phần mềm hiện đại hơn đã có mặt trên thị trường.

Vào năm 1981, Finkelstein và Martin đã phát triển "kỹ thuật thông tin" [7] cho những người sử dụng có mục đích thương mại (ví dụ, những người có các vấn đề cần giải quyết có xu hướng tập trung hơn vào cơ sở dữ liệu). Cho đến ngày nay, công nghệ này vẫn khá phổ biến đối với những người phát triển máy tính lớn với sự đầu tư vào chiến lược CASE của những năm 1990. Về cơ

bản, kỹ thuật thông tin là sự cải tiến phương pháp tiếp cận cấu trúc. Tuy nhiên, thay vì tập trung vào những dữ liệu theo cách tiếp cận cấu trúc, kỹ thuật thông tin lại tập trung vào nhu cầu thông tin của toàn bộ doanh nghiệp. Theo đó, các chuyên gia kinh tế sẽ xác định các mô hình thông tin cấp độ cao cũng như các mô hình dữ liệu chi tiết. Kết quả là hệ thống được thiết kế dựa trên những mô hình này.

Cả phương pháp cấu trúc và kỹ thuật thông tin đều ra đời từ nhu cầu ứng dụng máy tính lớn trong thương mại. Ngày nay, sự ra đời của công nghệ chủ/khách (ở đó dữ liệu của các tổ chức có thể được truyền qua các máy chủ được phân theo khu vực địa lý trong khi người tiêu dùng sử dụng GUI theo lựa chọn của mình để thực hiện quá trình xử lý địa phương) đã vô hiệu hoá phần lớn những phương pháp này. Trên thực tế, hiện nay có rất nhiều vấn đề đã nảy sinh trong các ứng dụng thương mại giống với những vấn đề cần được dự báo sớm trong những ngành kỹ thuật như viễn thông, hàng không.

Các môi trường chủ/khách được đặc trưng bởi tính đa dạng của chúng. Một tổ chức có thể lưu trữ dữ liệu trên nhiều cơ sở dữ liệu, lập trình bằng một số ngôn ngữ lập trình và sử dụng nhiều hơn một hệ điều hành, và vì thế họ cũng sử dụng những GUI khác nhau. Do quá trình phát triển phần mềm trở nên phức tạp hơn nên nhu cầu hiện nay là cần có một phương pháp phát triển phần mềm mới tốt hơn. Những kỹ thuật hướng đối tượng ngày nay giải quyết được một vài trong số những vấn đề này. Với tính phức tạp của môi trường chủ/khách, mã được gài trong các chương trình sẽ không đủ linh hoạt để đáp ứng nhu cầu của kiểu môi trường này. Tuy nhiên, việc phát triển phương pháp hướng đối tượng giống như một con dao hai lưỡi.

Trong khi rõ ràng việc làm chủ kỹ thuật này sẽ giúp tăng hiệu suất đáng kể, nhưng có một thực tế đáng buồn là việc phát triển nó, nếu không được thực hiện theo đúng cách, có thể sẽ gây ra những vấn đề phức tạp hơn rất nhiều so với những vấn đề được tạo ra từ các kỹ thuật cấu trúc. Lý do cho vấn đề này là rất đơn giản. Kinh phí thực hiện cao hơn. Những môi trường hướng đối tượng phức tạp hơn bất kỳ môi trường nào, những vấn đề thực tiễn được chọn để giải quyết bằng phương pháp hướng đối tượng phức tạp hơn rất nhiều so với bất kỳ loại vấn đề nào, và có rất ít phương pháp hướng đối tượng truyền thống cũng như các công cụ kèm theo giúp đội ngũ thiết kế được những hệ thống tốt. Có rất nhiều lựa chọn cho phương pháp hướng đối tượng. Do sự đa dạng này nên thường xảy ra một số rủi ro. Kết quả là những vấn đề sau cần phải được xem xét trước khi bật máy tính.

- Việc tích hợp là một vấn đề mang tính thách thức và cần được xem xét ngay từ đầu. Với những hệ thống truyền thống, các nhà phát triển dựa vào các phương pháp lập mô hình bất cân xứng để nắm bắt được các khía cạnh của vấn đề. Cho dù tích hợp đối tượng với đối tượng, môđun với môđun, giai đoạn với giai đoạn, kiểu ứng dụng với kiểu ứng dụng, thì đây là một quá trình khá phức tạp. Việc ghép không cân xứng các sản phẩm dùng trong thiết kế và phát triển sẽ càng làm cho vấn đề trở nên phức tạp hơn. Việc tích hợp thường tạo điều kiện tốt cho sự phát triển các thiết bị của nhiều nhà thiết kế. Đôi khi các hệ thống ra đời rất khó nắm bắt và các đối tượng rất khó theo rõi. Điều nguy hiểm lớn nhất là sự ít phù hợp với thế giới bên ngoài. Các giao diện thường không tương thích và lỗi thường diễn ra trong suốt quá trình phát triển. Kết quả là các hệ thống được xây dựng theo phương pháp này thường mơ hồ và hay bị lỗi.
- Lỗi cần được tối thiểu hoá. Các phương pháp truyền thống bao gồm những phương pháp hướng đối tượng thường làm gia tăng lỗi ví dụ như gia tăng lỗi thông qua việc tái sử dụng đối tượng có lỗi nội tại và nhúng trong suốt quá trình phát triển. Lỗi phải được giảm thiểu ngay từ đầu quá trình phát triển trước khi chúng bắt đầu vòng đời của mình.
- Ngôn ngữ cần được chuẩn hoá hơn. Mặc dù một số ngôn ngữ đã được chuẩn hóa và một số khác khá quen thuộc nhưng rất khó tìm những ngôn ngữ vừa *chuẩn hóa* vừa *quen thuộc*. Trong môi trường có nhiều ngôn ngữ không chuẩn hoá được sử dụng thì việc thiếu khả năng tìm kiếm và quá nhiều lỗi giao diện là một hiện tượng phổ biến. Gần đây, nhiều ngôn ngữ phần mềm hiện đại hơn đã ra đời (ví dụ Ngôn ngữ mô hình hợp nhất – Unified Modeling Language, UML [18]), hầu hết trong số đó là không chuẩn hóa

(hoặc bán chuẩn hóa); một số ngôn ngữ này được tạo ra bằng cách "tích hợp" một số ngôn ngữ lại với nhau.

- Hội chứng thiết kế cứng cần phải loại bỏ. Thông thường, các nhà thiết kế bị buộc phải phát triển trên cơ sở công nghệ ứng dụng không có cấu trúc mở ví dụ như một giản đồ cơ sở dữ liệu đặc biệt hoặc một GUI. Quả là tồi tệ nếu cứ cố gắng phát triển một hệ thống như vậy. Tuy nhiên còn tồi tệ hơn nếu sử dụng các bộ phận cấu thành nó như các bộ phận được tái sử dụng cho một hệ thống không dựa trên những công nghệ này. Những kế hoạch được chuẩn bị tốt, những thực tiễn kinh doanh chính thống và sự thi hành chúng sẽ giúp giảm thiểu được vấn đề này trong một tổ chức.
- Khả năng thay đổi và đối phó với những bất ngờ cần được xem xét trước tiên. Thông thường, người ta hay quên rằng khi xây dựng một ứng dụng cần phải xem xét quá trình phát triển của nó. Người sử dụng thay đổi thị hiếu, môi trường phát triển phần mềm thay đổi và các công nghệ cũng thay đổi. Những định nghĩa về yêu cầu trong bối cảnh phát triển thường tập trung vào nhu cầu ứng dụng của người sử dụng, mà không cân nhắc đến sự thay đổi nhu cầu của người sử dụng hoặc môi trường trong tương lai. Việc chuyển tới một môi trường mới đã trở thành một hướng phát triển mới mẻ cho mỗi cấu trúc, hệ điều hành, cơ sở dữ liệu, môi trường đồ họa hoặc ngôn ngữ mới. Do đó, người ta thường tránh phát triển những chức năng tới hạn do sợ những vấn đề không lường trước được và bảo dưỡng (phần gây ra nhiều rủi ro và tốn kém chi phí nhất của một vòng đời hệ thống) thường bị bỏ qua trong quá trình phát triển. Để giải quyết vấn đề này, cần phải sử dụng các công cụ và kỹ thuật để cho phép công nghệ giao thoa và công nghệ thay đổi, cũng như tạo ra sự thay đổi và phát triển các cấu trúc.
- Các nhà thiết kế phải chuẩn bị sẵn sàng cho môi trường song song và môi trường phân tán. Thông thường, khi một hệ thống được thiết kế cho môi trường phân tán, trước hết nó được xác định và phát triển sao cho thích ứng được với môi trường đơn xử lý và sau đó được tái phát triển lại cho phù hợp với môi trường phân tán - một cách tái sử dụng nguồn lực. Môi trường song song và phân tán phải được xem xét ngay từ đầu của quá trình phát triển.
- Việc phân bổ các nguồn lực cần phải được thể hiện rõ với người sử dụng. Cho dù một hệ thống có được phân bổ cho các bộ xử lý phân tán, thiếu đồng bộ hoặc đồng bộ hay không và cho dù hai hay mười bộ xử lý được lựa chọn theo phương pháp truyền thống, thì các nhà thiết kế và phát triển sản phẩm vẫn phải quan tâm đến việc tích hợp những chi tiết như vậy vào ứng dụng. Không có một sự khác biệt nào giữa mô tả kỹ thuật về sự vận hành của hệ thống với cách hệ thống thực thi nó. Điều này dẫn tới việc có quá nhiều chi tiết thực thi được đưa vào hệ thống trong giai đoạn thiết kế. Một khi cấu trúc nguồn này trở nên lỗi thời, điều cần thiết là phải thiết kế lại và phát triển lại những hệ thống ứng dụng có những thiết kế quá lỗi thời này.
- Kỹ thuật tự động hoá giúp giảm thiểu công việc thủ công. Trên thực tế, bản thân tự động hoá vốn là một quá trình có thể tái sử dụng được. Nếu một hệ thống không thể tái sử dụng được thì tất nhiên nó không phải là một hệ thống tự động hoá. Tuy nhiên, hầu hết quá trình phát triển ngày nay không nhất thiết phải thực hiện bằng tay. Các hệ thống ngày nay được xây dựng với độ thông minh không đủ cho các công cụ tự động hoá có thể sử dụng chúng như đầu vào. Trên thực tế, các công cụ tự động hoá tập trung vào việc hỗ trợ cho quá trình lao động thủ công thay vì thực hiện các công việc cụ thể. Điển hình là các chuyên viên thiết kế đã chuyển các định nghĩa sang dạng mã một cách thủ công. Quá trình được cơ khí hóa trước khi tái sử dụng thường được thực hiện bằng tay lặp đi lặp lại. Vì những lý do nêu trên nên thậm chí khi kỹ thuật tự động cố gắng thực hiện các công việc cụ thể, nó thường gây ra sự thiếu hoàn thiện giữa các miền ứng dụng hoặc thậm chí trong nội bộ một miền ứng dụng dẫn đến bộ mã thiếu hoàn chỉnh như mã shell chẳng hạn. Mã được tạo ra thường không hiệu quả và khó kiểm soát đối với một loại thuật toán, một loại cấu trúc, ngôn ngữ hoặc thậm chí một phiên bản ngôn ngữ nhất định. Thông thường việc tự động hoá từng phần cần đi kèm với sự tự động hoá từng

phần đối lập với nó hoặc quá trình thủ công nào đó. Các quá trình thủ công là rất cần thiết nhằm hoàn thiện nốt các công việc mà kỹ thuật tự động hoá chưa làm được.

- Việc phân tích hiệu suất làm việc (các quyết định về thuật toán và cấu trúc) nên dựa vào các định nghĩa chuẩn. Các định nghĩa hệ thống truyền thống chứa thông tin không đầy đủ về hoạt động của một hệ thống, bao gồm các quyết định liên quan đến thuật toán hoặc cấu trúc. Các định nghĩa hệ thống cần xem xét cả phương thức tách hệ thống ra khỏi môi trường mục tiêu của nó. Do không cân nhắc đến phương thức này nên các quyết định liên quan đến thiết kế phải dựa vào việc phân tích các kết quả của quá trình thực nghiệm có liên quan.
- Việc tạo ra các định nghĩa tin cậy có thể tái sử dụng cần phải được phát huy đặc biệt là những định nghĩa đã được cung cấp sẵn. Các định nghĩa yêu cầu truyền thống thiếu công cụ giúp tìm kiếm, sử dụng, và đảm bảo sự tương đồng trong hệ thống. Các nhà thiết kế buộc phải sử dụng các phương pháp thủ công và không chuẩn hóa để tìm cách phân chia hệ thống ra thành các bộ phận cho mục đích tái sử dụng. Do những bộ phận này không thể tự tích hợp lại với nhau nên kết quả là chúng có xu hướng bị lỗi. Mặt khác, những hệ thống này lại không thể vận chuyển hay sửa chữa được nên chúng ít có khả năng được tái sử dụng. Trong các phương pháp truyền thống, sự dư thừa đã trở thành một kiểu kinh doanh. Thậm chí khi các phương pháp được xây dựng nhằm hướng vào đối tượng, các chuyên viên thiết kế thường để mặc cho các thiết bị của họ thực hiện các ứng dụng hướng đối tượng. Sở dĩ như vậy là do những phương pháp này không trợ giúp tất cả những gì gắn với quá trình hướng đối tượng.
- Tính toàn vẹn thiết kế là bước đầu tiên khi xây dựng một hệ thống khả dụng. Khi sử dụng các phương pháp truyền thống, người ta không biết được liệu một thiết kế có tốt hay không cho tới khi việc ứng dụng nó bị thất bại hoặc thành công. Việc thiết kế một hệ thống thông thường dựa vào những cân nhắc mang tính ngắn hạn bởi vì kiến thức không được sử dụng lại từ những bài học trước đó. Cuối cùng, sự phát triển dẫn đến thất bại. Giải pháp được đưa ra ở đây là cần phải có phương pháp để xây dựng những định nghĩa tin cậy và có thể tái sử dụng được.

Khi những vấn đề nêu trên được giải quyết, thì phần mềm sẽ trở nên rẻ hơn và người ta không cần phải mất nhiều thời gian để phát triển chúng. Tuy nhiên, thời gian lại là một vấn đề khá cơ bản. Những vấn đề này đang trở nên cấp thiết khi các chuyên viên thiết kế chuẩn bị cho các môi trường phân tán cùng đồng hành với những ưu thế ngày càng gia tăng của các ứng dụng Internet.

Với những thách thức nêu trên, một tổ chức thường có nhiều lựa chọn. Những lựa chọn này bao gồm: (1) duy trì mọi thứ ở trạng thái ban đầu; (2) thêm các công cụ và kỹ thuật để hỗ trợ công việc, ngoài ra cung cấp thêm các trợ giúp cho những vùng được lựa chọn; (3) áp dụng những công cụ và kỹ thuật hiện đại hơn để thay thế những công cụ và kỹ thuật hiện có; (4) sử dụng một mô hình mới với những công cụ và kỹ thuật tiên tiến nhất nhằm chuẩn hoá quá trình phát triển phần mềm đồng thời tận dụng những phần mềm đã được phát triển trước đó; (5) bắt đầu với một mô hình hoàn toàn mới nhằm chuẩn hoá quá trình phát triển phần mềm và ứng dụng những công cụ và kỹ thuật tiên tiến nhất.

49.3 Sự phát triển trước thực tế

Chương này đã giải thích nguồn gốc của phần mềm và cố gắng cho chúng ta thấy nó đã phát triển như thế nào theo thời gian để trở thành "bộ não" của bất kỳ một hệ thống tự động nào. Tuy nhiên, cũng giống như bộ não con người, bộ não phần mềm phải được thiết kế cẩn thận nhằm tăng hiệu quả, chất lượng và đảm bảo khả năng kiểm soát và tái sử dụng.

Những mô hình kỹ thuật phần mềm trước kia không nhìn nhận quá trình phát triển phần mềm ở một góc độ lớn hơn như một xã hội hữu cơ đã được trình bày ở đầu chương này. Chỉ khi chúng ta coi quá trình phát triển phần mềm như việc chế tạo những bộ phận riêng biệt nhưng có khả năng tích hợp cao thì chúng ta mới có thể bắt đầu phát triển một phương pháp phát triển phần mềm mang lại những lợi ích như đã được hứa hẹn bởi sự ra đời của các thập niên phần mềm trước đó.

Từ góc độ này, kỹ thuật phần mềm bao gồm một phương pháp cùng một loạt các công cụ giúp thực thi các giải pháp cho các vấn đề thực tiễn trước mắt. Tuy nhiên, thậm chí trước khi công cụ đầu tiên có thể được áp dụng, phương pháp kỹ thuật phần mềm phải được xây dựng để hỗ trợ việc xác định những yêu cầu đặt ra của vấn đề này. Làm thế nào để có thể đạt được điều này một cách thành công khi phải đối mặt với những vấn đề cần giải quyết đã nêu ở phần trước? Làm thế nào để giải quyết được vấn đề này khi các tổ chức phải phát triển những hệ thống có thể chạy qua các nền phần cứng, cơ sở dữ liệu, ngôn ngữ lập trình và GUI đa dạng và phân tán trong khi những hệ phương pháp truyền thống không tạo ra tính đa dạng này? Và làm thế nào để phát triển phần mềm mà không cần giải quyết vô số những vấn đề gây ra "sự đã rồi" này của việc phát triển phần mềm?

Vấn đề đặt ra ở đây là chúng ta cần phải xem xét lại một cách nghiêm túc cách thức xây dựng phần mềm để hiểu được làm thế nào có thể xây dựng được các hệ thống vận dụng đúng kỹ thuật, đúng thời điểm. Trước hết, đây làm một phương pháp tiếp cận mang tính phòng ngừa, tức là nó tạo cho chúng ta một khuôn khổ làm việc chính xác ngay từ đầu. Những vấn đề tồn tại gắn liền với phương pháp thiết kế và phát triển phần mềm truyền thống đã được ngăn chặn trước khi chúng xảy ra chỉ bằng cách định hình hệ thống. Phương pháp tiếp cận như vậy sẽ giúp ngăn chặn xảy ra những vấn đề trong khi phát triển phần mềm thay vì để chúng xảy ra "sau khi sự đã rồi", và sau đó lại giải quyết chúng trong thời điểm đắt đỏ và không thích hợp nhất.

Chúng ta cần cân nhắc cách tiếp cận này khi ứng dụng nó với các hệ thống thân thiện với con người. Hàn răng trước khi sâu tấn công đến ống răng là việc chữa răng khỏi bị sâu răng, nhưng chính là việc ngăn ngừa nó khỏi bị rỗng ống răng. Ngăn ngừa sâu răng bằng cách ăn uống hợp lý không chỉ ngăn ngừa ống răng mà còn ngăn ngừa sâu răng nữa. Việc chữa răng khi nó bị rỗng ống răng là khá tốn kém, việc chữa sâu răng lại còn tốn kém hơn, và việc ngăn chặn sâu răng ngay từ đầu là liệu pháp ít tốn kém nhất.

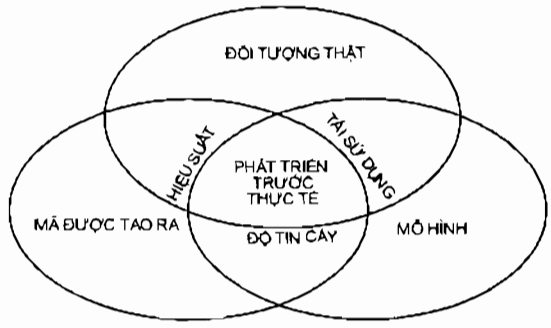
Phòng ngừa là một khái niệm mang tính tương đối. Đối với một hệ thống, dù là con người hay phần mềm thì mục tiêu lớn nhất là phải ngăn chặn càng sớm càng tốt những gì có thể xảy ra trong vòng đời của nó.

Với quan điểm phòng ngừa, các hệ thống cần phải được phát triển một cách cẩn thận để ngay từ đầu giảm thiểu những rắc rối có thể xảy ra. Một hệ thống có thể được phát triển với những tính năng cho phép kiểm soát thiết kế và sự phát triển của riêng nó. Kết quả là cho ra đời những hệ thống có thể tái sử dụng được góp phần thúc đẩy sự phát triển của kỹ thuật tự động hoá. Mỗi định nghĩa hệ thống phải mô hình hoá được cả ứng dụng và vòng đời của nó với những ràng buộc gắn liền, những ràng buộc này giúp bảo vệ các chuyên viên thiết kế mà không làm mất tính linh hoạt của họ.

Thực chất của quan điểm phòng ngừa là cần phải xây dựng những hệ thống đáng tin cậy. Chỉ những hệ thống tin cậy mới được coi là "một khối kết cấu", và chỉ những hệ thống đáng tin cậy mới có khả năng tích hợp những khối này lại thành một hệ thống mới. Hệ thống mới này lại được tái sử dụng khi xây dựng những hệ thống khác.

Việc tái sử dụng hiệu quả chính là một nội dung của quan điểm phòng ngừa, tức là, khi tái sử dụng một bộ phận nào đó (ví dụ như các yêu cầu hay mã lệnh) mà không bị lỗi và cho phép thực hiện được những chức năng như mong muốn thì có thể tránh được các sai sót và giảm được chi phí phát triển một hệ thống mới. Nó giúp giải quyết vấn đề một cách sớm nhất có thể. Tuy nhiên để làm cho hệ thống có khả năng tái sử dụng một cách thực sự thì chúng ta cần phải tiến hành ngay từ đầu chứ không phải từ khi kết thúc một vòng đời hệ thống, trong giai đoạn vận hành hoặc bảo dưỡng.

Những hệ thống được xây dựng theo quan điểm phòng ngừa là những hệ thống thực sự đáng tin cậy, ở đó những phần tử phần mềm tự kết hợp lại với nhau để hình thành nên một tổng thể lớn hơn rất nhiều so với tổng các bộ phận của nó. Hoặc chúng ta có thể nghĩ đến việc xây dựng những hệ thống từ những món đồ chơi được chắp vá lại của thuở niên thiếu. Chúng ta có thể hồi tưởng lại rằng khi xưa còn là một đứa trẻ, chúng ta không bị mắc sai lầm khi xây dựng những cấu trúc tuyệt đẹp từ những món đồ chơi chắp vá. Thực vậy, những món đồ chơi chắp vá được làm từ những vật liệu có thể tái sử dụng được, có khả năng gắn kết một cách hoàn hảo và hoàn toàn thân thiện với người sử dụng.



HÌNH 49.1 Mô hình phát triển trước thực tế

Một cách tiếp cận theo quan điểm phòng ngừa là sự phát triển trước thực tế (DBTF – development before the fact), như thấy ở hình 49.1. Hiện nay quan điểm này đang được các tổ chức nghiên cứu tiên phong sử dụng thành công và hiện đang được vận dụng cho mục đích thương mại. Quan điểm này được mô tả nhằm minh họa cho những khả năng có thể xảy ra theo quan điểm phòng ngừa.

Trong khi các quan điểm tiếp cận truyền thống bắt đầu quá trình phát triển phần mềm trong giai đoạn "sự đã rồi" thì mô hình DBTF mới đang ở giai đoạn sơ khai. Nó bắt nguồn từ việc kết hợp các bước giải quyết những vấn đề của hệ thống kỹ thuật truyền thống và của việc phát triển phần mềm. DBTF bao gồm một công nghệ, một ngôn ngữ và một quá trình (phương pháp luận) dựa trên một học thuyết căn bản.

Ngôn ngữ

Các đặc tính của một thiết kế tốt có thể được tái sử dụng bằng cách kết hợp chúng thành một dạng ngôn ngữ để xác định bất kỳ hệ thống nào (như hệ thống phần mềm chẳng hạn). Ngôn ngữ dựa trên DBTF là một hình thức thể hiện mô hình toán học của hệ thống. Một hệ thống được xác định dựa trên ngôn ngữ này có các đặc tính có thể tự điều khiển được chính bản thân hệ thống. Dựa trên quan điểm (DBTF) với sự mở rộng mô hình toán học truyền thống của hệ thống thông qua khái niệm duy nhất về điều khiển, ngôn ngữ hình thức nhưng quen thuộc này tự bản thân nó đã bao gồm sự biểu diễn vật lý tự nhiên của thời gian và không gian. Nhờ có dạng ngôn ngữ này, mọi đối tượng đều là đối tượng hướng hệ thống (SOO – system-oriented object). Loại đối tượng này là sự tích hợp các vấn đề hướng chức năng và hướng đối tượng. Thay vì các hệ thống hướng đối tượng, các đối tượng đều hướng hệ thống. Tất cả các hệ thống là những đối tượng và tất cả các đối tượng là những hệ thống.

Vì lý do đó nên nhiều đối tượng trước đây không khả thi theo phương pháp truyền thống thì nay đã trở nên khả thi. Một hệ thống DBTF bản thân nó bao gồm tất cả các đối tượng của chính nó (và các khía cạnh, mối quan hệ cũng như quan điểm của những đối tượng này) và sự kết hợp các tính năng bao gồm sự định giờ; tối đa hoá độ tin cậy và tính linh hoạt đối với những thay đổi về yêu cầu đích, cấu trúc động lực học, tĩnh học, quá trình cũng như sự tái cấu hình trong thời gian thực; tận dụng tính song song và khả năng lần theo vết của chính nó; hỗ trợ phân tích hiệu suất làm việc; tối đa hoá khả năng tái sử dụng chính nó (cung cấp sự phân bổ và tái sử dụng nguồn lực nội tại mà không cần sự can thiệp của các chuyên viên thiết kế); và cuối cùng là cung cấp khả năng tự động hoá giai đoạn thiết kế và quá trình phát triển bất cứ lúc nào và bất cứ nơi đâu có thể. Mỗi hệ thống DBTF được xây dựng với chất lượng, công suất và khả năng điều khiển nội tại.

Ngôn ngữ - hệ thống lệnh được mã hóa dùng trong lập trình, thực sự là một nhân tố quan trọng trong hệ thống DBTF. Vai trò chính của nó là giúp các chuyên viên lập trình giảm thiểu sự phức tạp và mang lại sự rõ ràng, mạch lạc trong quá trình nghiên cứu của họ để biến nó thành loại ngôn ngữ có khả năng tái sử dụng tối ưu nhất. Nó có thể được dùng để xác định bất kỳ bộ phận nào của hệ thống và tích hợp bộ phận đó với bất kỳ bộ phận nào khác của hệ thống. Vấn đề quan trọng là

những bộ phận này có mối quan hệ trực tiếp với thế giới thực và do đó cùng một loại ngôn ngữ có thể được sử dụng để xác định các yêu cầu hệ thống, đặc điểm kỹ thuật, thiết kế và thiết kế chi tiết cho những cấu trúc chức năng, nguồn lực, sự phân bổ nguồn lực thông qua tất cả các cấp độ và các lớp định nghĩa của hệ thống bao gồm phần cứng, phần mềm và phần con người.

Ngôn ngữ dựa trên hệ thống DBTF này có thể được sử dụng để xác định các tổ chức của con người, các hệ thống ngân hàng, các hệ thống nhận thức cũng như môi trường thời gian thực hoặc môi trường cơ sở dữ liệu và do đó nó rất phù hợp cho việc ứng dụng trong các ngành kinh tế, trường học và chính phủ.

Công nghệ

Kinh nghiệm từ cuộc sống đã biến công nghệ DBTF thành hiện thực. Qua hơn ba thập kỷ phát triển, công nghệ này được bắt nguồn từ một loạt các lý thuyết hệ thống, các phương pháp chuẩn hóa và công nghệ hướng đối tượng. Công nghệ DBTF bao gồm lý thuyết, ngôn ngữ hỗ trợ sự hiển thị của nó và kỹ thuật tự động hoá hỗ trợ tính ứng dụng của nó. Trên thực tế, mỗi bộ phận cấu thành của công nghệ này được phát triển theo hướng kinh nghiệm thực tiễn giúp hình thành nên lý thuyết và lý thuyết giúp xây dựng ngôn ngữ, còn ngôn ngữ thì hỗ trợ kỹ thuật tự động hóa. Tất cả những bộ phận cấu thành nên công nghệ này được sử dụng kết hợp để thiết kế hệ thống và phát triển phần mềm.

Công nghệ DBTF bắt đầu được nghiên cứu phát triển vào năm 1968, khi người ta tiến hành phân tích nhiệm vụ của tàu Apollo. Một phương pháp tốt hơn là rất cần thiết để xác định và phát triển các hệ thống thay cho những công nghệ hiện tại bởi vì cũng giống như những công nghệ truyền thống chúng đã không giải quyết được những vấn đề cấp bách hiện nay. Nghiên cứu phát triển phần mềm nhằm phục vụ nhu cầu của con người giúp phát hiện ra rằng các lỗi giao diện chiếm tới gần 75% tất cả các lỗi được tìm trong các phần mềm điều khiển máy bay trong lần thử nghiệm cuối cùng (trước kia con số này là 90%). Những lỗi như vậy bao gồm các lỗi luồng dữ liệu, ưu tiên, và lỗi thời gian từ mức cao nhất của hệ thống đến các mức thấp nhất của các chi tiết. Loại lỗi này được phân loại dựa trên tiêu chí nó có thể được ngăn chặn như thế nào thông qua cách hệ thống được định nghĩa. Việc này dẫn đến sự ra đời của một học thuyết và hệ phương pháp nhằm xác định hệ thống có khả năng giảm thiểu tất cả các lỗi giao diện.

Công nghệ đầu tiên bắt nguồn từ học thuyết này tập trung vào việc xác định và xây dựng các hệ thống tin cậy. Do nhận thức được lợi ích của việc định rõ một vấn đề cơ bản như độ tin cậy chẳng hạn, nên người ta đã tiếp tục tập trung nghiên cứu các vấn đề cơ bản khác với cùng một cách tiếp cận giống nhau là dựa trên cách xác định hệ thống [9-11].

Công nghệ DBTF là công nghệ tập trung vào chức năng và hướng đối tượng dựa trên quan niệm duy nhất về điều khiển, mà người ta chưa từng ứng dụng nó trong bất kỳ một mô hình kỹ thuật phần mềm nào khác. Cơ sở lý luận của công nghệ này được dựa trên một loạt các tiên đề và giả định về một tập hợp phổ biến các đối tượng. Mỗi tiên đề xác định một mối quan hệ chỉ phối trực tiếp. Tập hợp các mối quan hệ được xác định bởi các tiên đề chính là điều khiển. Trong các trường hợp khác, tiên đề xác lập các mối quan hệ của một đối tượng về đầu vào/đầu ra, quyền truy nhập đầu vào/đầu ra, phát hiện và khắc phục lỗi và đưa ra quyết định trong quá trình vận hành và phát triển của nó. Bảng 49.1 Tóm tắt một số đặc điểm của các đối tượng trong hệ thống DBTF.

Quy trình

Kỹ thuật phần mềm khiếm khuyết ở chỗ nó không có khả năng nhận thức được rằng trong số nhiều mô hình thì phải lựa chọn được một mô hình phù hợp và mô hình đó phải là một phần của môi trường giúp cung cấp các phương pháp tự động tích hợp để giải quyết một vấn đề ngay lập tức. Điều đó có nghĩa là mô hình này phải đi kèm với một hệ thống các công cụ tích hợp để thực thi các kết quả của việc sử dụng mô hình đó để phát triển mô hình hệ thống.

Về cơ bản, mô hình này giúp tạo ra mô hình hệ thống và cần phải có một loạt các công cụ để tạo ra hệ thống. Công nghệ DBTF sẽ cung cấp những khả năng ưu việt này.

Công nghệ DBTF được ứng dụng xuyên suốt vòng đời hệ thống bắt đầu từ các yêu cầu, tiếp theo là phân tích chức năng, mô phỏng, mô tả kỹ thuật, phân tích, thiết kế, thiết kế cấu trúc hệ thống, phát triển thuật toán, thực thi, quản lý cấu hình, thử nghiệm, bảo dưỡng và duy trì. Người ứng dụng công nghệ này bao gồm người tiêu dùng, nhà quản lý, kỹ sư hệ thống, kỹ sư phần mềm, và kỹ sư thử nghiệm.

Quy trình công nghệ DBTF bao gồm việc hoàn thiện các thuật toán với sự chính xác về kỹ thuật. Mục đích của nó là làm đơn giản hóa hướng phát triển theo kiểu "làm mọi thứ chính xác ngay từ đầu", tránh phương pháp tiếp cận truyền thống theo kiểu "chạy theo sửa chữa những thứ bị hư hỏng". Kỹ thuật tự động hóa theo phương pháp này được phát triển với các lưu ý sau: ngăn chặn lỗi phát sinh ngay từ giai đoạn đầu khi xác định hệ thống, điều khiển vòng đời của hệ thống và tái sử dụng những hệ thống có độ tin cậy cao. Vòng đời phát triển được chia thành một chuỗi các giai đoạn bao gồm các yêu cầu và xây dựng mô hình thiết kế dựa trên các phân tích và mô tả kỹ thuật chuẩn hóa, tạo mã tự động dựa trên các mô hình hoàn thiện một cách nhất quán và logic, thử nghiệm, vận hành và mô phỏng.

BẢNG 49.1 Các đặc trưng hệ thống hướng đối tượng của công nghệ DBTF

Chất lượng (tốt hơn, nhanh hơn, rẻ hơn)
- Tin cậy
- Giá cả hợp lý

Tin cậy (tốt hơn)
- Trong quá trình điều khiển
- Dựa trên các tiên đề
  - Nhận dạng miền (có chủ đích, không chủ đích)
  - Thứ tự (ưu tiên và thời gian)
  - Quyền truy nhập: đối tượng nhập, đối tượng xuất
  - Thay thế
- Hình thức
  - Nhất quán, hoàn thiện một cách lôgic
  - Cần thiết và đầy đủ
  - Cơ sở ngữ nghĩa thông dụng
  - Nhận dạng trạng thái duy nhất
- Lỗi tự do (Dựa trên định nghĩa chính thống về "lỗi")
  - luôn luôn nhận được câu trả lời chính xác đúng lúc, đúng chỗ
  - Thoả mãn nhu cầu của người sử dụng và ý đồ của lập trình viên
- Giải quyết những vấn đề không lường trước được
- Lường trước được

Giá cả hợp lý (nhanh hơn, rẻ hơn)
- Khả năng tái sử dụng
- Tối ưu hóa nguyên liệu trong quá trình vận

Đối phó với những vấn đề không lường trước được
- Xuyên suốt quá trình phát triển của hệ thống
- Không làm ảnh hưởng đến khu vực phi can thiệp
- Nhận dạng và khắc phục lỗi
- Tiếp xúc, biến đổi và tái cấu hình trong môi trường thời gian thực, phân tán, đồng nhất

Linh hoạt
- Có khả năng thay đổi mà không bị phản ứng phụ
- Có khả năng phát triển
- Bền
- Tin cậy
- Có khả năng mở rộng
- Có khả năng tách rời và hợp nhất
  - Một đối tượng với nhiều đối tượng: điều chỉnh, phân ly, thuyết minh
  - Nhiều đối tượng với một đối tượng: Hợp nhất, tích hợp, trừu tượng
- Có thể sách tay
  - Giải thoát
  - Những khai triển lớp đa dạng và biến đổi
  - Cấu trúc mờ (sự bổ sung, sự phân bổ nguồn nguyên liệu và sự thực thi độc lập)
  - Cắm vào hoặc cấu hình tự thích nghi của của các môđun khác nhau
  - Khả năng thích nghi đối với các tổ chức, các ứng dụng, chức năng, con người và sản phẩm khác nhau.

Tự động hóa
- Hình thức tái sử dụng tối ưu

hành và phát triển
- Thời gian và không gian tối thiểu
- Hiệu quả sử dụng nguyên liệu cao

Khả năng tái sử dụng

- Có thể nắm bắt được, tích hợp được và duy trì được
- Linh hoạt
- Đáp ứng tiêu chuẩn
- Tự động hóa
- Định nghĩa thông dụng
  - Tự điều chỉnh
    + Tự tách rời (vd tách cấu trúc chức năng khỏi cơ cấu nguyên liệu)
    + Mô hình cảm
    + Một đối tượng được tích hợp với cấu trúc, phương thức vận hành và đặc điểm điều khiển
  - Tích hợp về mặt cấu trúc và phương thức vận hành
  - Loại cơ cấu
    + Bản đồ chức năng (gắn kết chức năng này với chức những chức năng khác của đối tượng)
    + Sơ đồ loại đối tượng (gắn kết đối tượng với đối tượng)
    + Loại và cấu trúc chức năng
  - Phân loại
    + Tương đối
      ° Thuyết minh
      ° Sự đa hình thái
      ° Mẹ/con
      ° Trạng thái/hoạt động
      ° Có/không
    + Trừu tượng
      ° Tóm tắt
      ° Thay thế
      ° Mối quan hệ chức năng
      ° Typing bao gồm phân loại
      ° Hình thức bao gồm cả cấu trúc và cách hoạt động (đối với các loại hình và chức năng đối tượng)
    + Nguồn gốc
      ° Giảm thiểu
      ° Kết luận
      ° Di truyền
- Trang trọng hóa, cơ khí hóa, tiếp theo là tự động hóa
  - Nó
  - Sự phát triển của nó
  - Các nhân tố giúp tự động hóa quá trình phát triển của nó

Có thể nắm bắt được, tích hợp được và duy trì được

- Tin cậy
- Sự việc có thể đo được
- Đáp ứng với thế giới thực
  - Tiếp tục tồn tại, tạo ra và xóa đi
  - Xuất hiện và biến mất
  - Khả năng truy nhập
  - Sự tham khảo
  - Giải thiết về sự tồn tại của đối tượng
  - Thời gian thực và hạn chế về không gian
  - Sự miêu tả
  - Tính tương đối, sự trừu tượng, nguồn gốc
- Cung cấp cho người sử dụng những định nghĩa quen thuộc
  - Nhận ra rằng sự quen thuộc của người sử dụng này là khó khăn của người sử dụng khác
  - Che dấu các chi tiết không cần thiết (trừu tượng)
  - Có thể thay đổi, cú pháp do người sử dụng lực chọn
  - Tự đào tạo
  - Bắt nguồn từ một cơ sở ngữ nghĩa thông dụng
  - Cơ cấu định nghĩa thông dụng
- Liên kết các ngữ nghĩa phổ thông với tất cả các thực thể
- Được xác định càng đơn giản càng tốt nhưng không quá đơn giản
- Được xác định với sự tích hợp tất cả các đối tượng của nó (và tất cả các khía cạnh của những đối tượng này)
- Đưa ra nguồn gốc của phương thức hoạt động, cấu trúc cũng như những thay đổi của chúng (bảo dưỡng) xuyên suốt vòng đời
- Nhận biết và có khả năng đạt được mục đích
  - Định nghĩa
  - Sự phát triển của chính nó và của các nhân tố giúp thúc đẩy quá trình phát triển của nó
    + Phân tích
    + Thiết kế
    + Thực thi
    + Thuyết minh
    + Thử nghiệm
    + Bảo dưỡng

Ghi chú: Tất cả những từ gạch chân ám chỉ khả năng tái sử dụng.

Nguồn: Hamilton, M., "Software Design and Development," *The Electronics Handbook*, CRC Press, Boca Raton, FL, 1996. Đã được cấp phép.

Bước đầu tiên để xây dựng một hệ thống DBTF là dùng ngôn ngữ định nghĩa mô hình. Quy trình này có thể nằm ở bất kỳ giai đoạn nào của vòng đời phát triển, bao gồm phân tích các vấn đề, vận hành, và thiết kế. Mô hình này được tự động phân tích để đảm bảo nó được xác định một cách hợp lý. Việc này bao gồm phân tích tĩnh cho các tính chất phòng ngừa và phân tích động theo mục đích của người sử dụng.

Trong giai đoạn tiếp theo, bộ tạo mã nguồn sẽ tự động tạo ra một phần mềm được tích hợp đầy đủ cho bất kỳ một ứng dụng nào phù hợp với mô hình này và cho môi trường mục tiêu xác định theo ngôn ngữ và cấu trúc lựa chọn. Nếu môi trường được lựa chọn đã được định hình thì bộ tạo mã sẽ lựa chọn môi trường đó một cách trực tiếp, nếu không bộ tạo mã sẽ được định hình trước cho một ngôn ngữ và cấu trúc mới.

Nhờ có cấu trúc mở mà bộ tạo mã có thể được định hình để tồn tại dưới bất kỳ cấu trúc nào (hoặc tiếp xúc với bất kỳ môi trường bên ngoài nào), ví dụ, với một ngôn ngữ, gói truyền thông, một giao diện Internet, một gói cơ sở dữ liệu hoặc một hệ điều hành; hay nó có thể được định hình để giao tiếp với mã kế thừa của người sử dụng. Khi được định hình trong một môi trường mới, một hệ thống có thể tự động phục hồi để có thể tồn tại trong môi trường mới đó. Cấu trúc mở này tự thích ứng với quá trình phát triển dựa trên bộ phận cấu thành thực, tạo cho người sử dụng tính linh hoạt khi thay đổi các yêu cầu hoặc cấu trúc hay khi chuyển từ một công nghệ cũ sang một công nghệ mới.

BẢNG 49.2 Sự so sánh

Truyền thống	DBTF
Lỗi giao diện (trên 75% của tất cả các lỗi)	*Không có lỗi giao diện*
Phần lớn tìm thấy sau khi phát triển	Tất cả tìm thấy trước quá trình vận hành
Một số được tìm thấy bằng cách thủ công	Tất cả được tìm thấy tự động và qua phân tích
Một số được tìm thấy qua phân tích quá trình vận hành	thống kê.
Một số không tìm thấy	Luôn luôn tìm thấy
Những yêu cầu không rõ ràng	*Yêu cầu rõ ràng*
Ngôn ngữ không chính thống và bán chính thống	Ngôn ngữ chính thức, nhưng thân thiện
Các giai đoạn, ngôn ngữ và công cụ khác nhau	Tất cả các giai đoạn, cùng ngôn ngữ và các công cụ
Ngôn ngữ khác nhau cho hệ thống khác thay vì cho phần mềm	Cùng ngôn ngữ cho phần mềm, phần cứng và hệ thống khác
Tự động hóa hỗ trợ cho quá trình thủ công	*Tự động hóa thực hiện các công việc*
Tài liệu, lập trình, kiểm tra, theo rõi một cách thủ công	Tài liệu, chương trình, kiểm tra, theo rõi một cách tự động
	100% mã được tự động sinh ra
Không đảm bảo sự toàn vẹn chức năng sau quá trình vận hành	*Đảm bảo sự toàn vẹn chức năng sau quá trình vận hành*
Hệ thống không thể theo rõi hoặc phát triển	*Hệ thống có thể theo rõi hoặc phát triển*
Sản phẩm, cấu trúc, v.v... đóng	Cấu trúc mở
Khó kế thừa	Dễ kế thừa
Tái sử dụng phi nội tại	*Tái sử dụng nội tại*
Tái sử dụng lựa chọn	Mọi đối tượng được tái sử dụng
Sự tùy biến và tái sử dụng không phụ thuộc nhau	Sự tùy biến làm tăng khả năng tái sử dụng
Các đối tượng, giai đoạn, sản phẩm, cấu trúc và môi trường không tương xứng	*Các đối tượng, giai đoạn, sản phẩm, cấu trúc và môi trường tích hợp*
Hệ thống không được tích hợp với phần mềm	Hệ thống được tích hợp với phần mềm

Hướng chức năng hoặc hướng đối tượng	Các đối tượng hướng hệ thống: tích hợp chức năng, thời gian và hướng đối tượng.
GUI không đi liền với các ứng dụng	GUI không đi liền với các ứng dụng
Mô phỏng không tích hợp với mã phần mềm	Mô phỏng tích hợp với mã phần mềm
Không tự xác định và phát triển tự động	*Tự xác định và phát triển tự động*
Tốn kém, hệ thống thường bị lỗi	*Hệ thống tốt hơn, nhanh hơn, rẻ hơn*
Không hiệu quả	Tiết kiệm 10, 20, 50... USD
Khó hoàn thành kế hoạch	Hoàn thành trong thời gian ngắn
Nhiều thứ cần thì không có, không cần lại có	Những thứ bạn cần không thừa, không thiếu

Tiếp theo, có thể vận hành hệ thống thu được. Nhờ có phần mềm mà hệ thống có thể chạy thử nghiệm để kiểm tra những lỗi do người sử dụng tạo ra. Nó sẽ sẵn sàng vận hành sau quá trình thử nghiệm này. Những thay đổi ứng dụng thường là những thay đổi về yêu cầu/định nghĩa mô tả kỹ thuật chứ không phải thay đổi về mã lệnh (các chuyên viên thiết kế thậm chí không cần thay đổi mã). Những thay đổi cấu trúc đích thường là những thay đổi về cấu hình của môi trường bộ tạo mã chứ không phải là sự thay đổi về mã. Nếu hệ thống thực là phần cứng hoặc phần con người, hệ thống phần mềm sẽ đóng vai trò là công cụ mô phỏng mà hệ thống thực có thể dựa vào. Khi một hệ thống được phát triển, hệ thống này và quá trình phát triển nó được phân tích để tìm hiểu phương pháp cải tiến vòng phát triển tiếp theo của hệ thống.

Sự tích hợp hữu cơ được thực hiện từ hệ thống tới phần mềm, yêu cầu, thiết kế, mã, kiểm tra và đến các yêu cầu khác rồi lặp lại; từ cấp độ đến cấp độ và từ lớp đến lớp. Các chuyên viên thiết kế có thể lần theo các yêu cầu đến mã và quay trở lại.

Giả sử một kỹ thuật tự động có những khả năng này, thì không có gì đáng ngạc nhiên khi kỹ thuật tự động của DBTF tự xác định và tiếp tục tự động tạo ra nó, khi nó phát triển với những thay đổi về cấu trúc và công nghệ. Bảng 49.2 bao gồm bảng tóm tắt một số khác biệt giữa mô hình phòng ngừa hiện đại và mô hình truyền thống.

49.4 Thực nghiệm với công nghệ DBTF

Sự tối ưu của quan điểm phòng ngừa đã được chứng minh qua một vài thực nghiệm. Công nghệ DBTF đã được các học giả hàng đầu, các cơ quan chính phủ cũng như các tổ chức thương mại nhiều lần tiến hành đánh giá, so sánh và nó luôn chiếm vị trí hàng đầu. Công nghệ này khác biệt với các công nghệ khác ở chỗ nó giúp tạo nên một môi trường phát triển và thiết kế hệ thống tích hợp tổng thể, trong khi đó các công nghệ truyền thống lại tạo ra một vòng đời hệ thống phi chính thống, khó tích hợp (bao gồm các môđun ứng dụng cũng như các sản phẩm để thực thi chúng), thiếu tính hoàn thiện, mang tính thủ công và phát triển sau khi các vấn đề đã nảy sinh.

Cơ quan kiểm tra quốc gia của Bộ Quốc Phòng Hoa Kỳ đã đứng ra tiến hành một cuộc thí nghiệm, trong đó họ cung cấp cho ba đội nhà thầu được chọn lựa từ nhiều nhà thầu một vấn đề và môi trường phát triển dựa trên một tập hợp các yêu cầu được xác định rõ ràng. Nhiệm vụ của các đội này là phải cho ra đời một hệ thống phục vụ khách hàng, đa người sử dụng, phân tán và thời gian thực.

Tất cả các đội đều có khả năng hoàn thành phần công việc đầu tiên là xác định các yêu cầu sơ bộ, ban đầu. Có hai đội hoàn thành được phần thiết kế chi tiết nhưng chỉ có một đội có khả năng tạo ra bằng kỹ thuật tự động một loại mã mang tính hoàn chỉnh, được tích hợp và hoàn toàn sẵn sàng cho việc sản xuất. Phần chính của loại mã này được cho chạy trong C và Ada vào cuối cuộc thử nghiệm [12]. Đội đã tạo ra được mã hoàn chỉnh, sẵn sàng cho việc sản xuất đó đã ứng dụng 001 Tool Suite, một môi trường phát triển dựa trên công nghệ DBTF.

49.5 Kết luận

Những doanh nghiệp mong muốn đạt được năng suất cao từ việc đầu tư vào công nghệ, trong nhiều trường hợp vẫn chỉ biết ngồi chờ đợi kết quả đầu tư của mình. Phần lớn vấn đề bắt nguồn ở cách thức các tổ chức xây dựng các hệ thống tự động hóa của họ. Trong khi các tính năng của phần cứng đã tăng lên đáng kể, thì các tổ chức lại vẫn đang gặp nhiều khó khăn với các phương pháp lạc hậu gắn liền với sự ra đời của những bộ máy tính cồng kềnh. Các công nghệ cũ này không thể giúp xây dựng nên những hệ thống mới.

Bên cạnh đó, còn có rất nhiều những đổi thay. Người sử dụng thường mong muốn những hệ thống của họ phải có nhiều tính năng hơn và linh hoạt hơn. Và do những công nghệ mới giúp giải quyết rất nhiều vấn đề nên những hệ thống này cũng không được mắc lỗi.

Trong khi xã hội hữu cơ sinh học có các cơ chế điều khiển nội tại giúp tăng năng suất và chất lượng, thì cho đến nay xã hội hữu cơ silicon vẫn không có được cơ chế ấy. Và do đó năng suất không tăng.

Thông thường, cách duy nhất để giải quyết những vấn đề chính này là phải thông qua các phương pháp phi truyền thống hoặc các phương pháp cách tân. Người ta phải tạo ra những phương pháp mới hoặc những môi trường mới để sử dụng những phương pháp mới này.

Đổi mới để có được thành công thường khởi đầu bằng việc xem xét lại những khiếm khuyết của các hệ thống trước đây. Trước hết phải nhận dạng được cốt lõi thực của các vấn đề, sau đó phân loại chúng theo cách thức chúng được giải quyết. Bước tiếp theo là tìm các cách giải quyết thực tiễn. Việc thu được lời giải thực tế là bước logic tiếp theo. Sự lặp đi lặp lại của quá trình này dẫn đến việc tìm ra những vấn đề mới này sinh trong môi trường giải quyết vấn đề. Đó chính là cách thức hoạt động của công nghệ DBTF.

Nhờ có công nghệ DBTF, tất cả các khía cạnh của việc phát triển và thiết hệ thống được tích hợp với một ngôn ngữ hệ thống và kỹ thuật tự động của nó. Việc tái sử dụng sẽ diễn ra một cách tự nhiên xuyên suốt vòng đời hệ thống. Các đối tượng đều có thể được tái sử dụng và tích hợp bất chấp độ phức tạp của chúng. Các cấu hình môi trường cho các loại cấu trúc khác nhau cũng có thể được tái sử dụng. Một hệ thống mới được phát triển có thể an toàn cho việc tái sử dụng nhằm thúc đẩy hơn nữa hiệu suất của những hệ thống được phát triển dựa vào nó.

Mô hình sẽ được thay đổi khi các chuyên viên thiết kế nhận ra rằng rất nhiều trong số các công cụ trước đây không còn cần thiết cho việc thiết kế vào phát triển hệ thống nữa. Ví dụ, nhờ có ngôn ngữ nghĩa học chính thống giúp xác định và tích hợp tất cả các bộ phận của một hệ thống nên nhiều ngôn ngữ mô hình (và phương pháp luận) có nhiệm vụ xác định một phần hệ thống không còn cần thiết nữa. Cũng không cần thiết phải làm tương thích nhiều kỹ thuật với ngữ nghĩa học để tạo ra sự tương tác giữa chúng.

Công nghệ DBTF có thể hỗ trợ người sử dụng trong việc giải quyết nhiều vấn đề thách thức hiện nay trong các môi trường phát triển phần mềm. Tuy nhiên, vẫn còn rất nhiều việc phải làm để có thể ứng dụng một cách tối ưu công nghệ này. Do nó được dựa trên một số tiền đề và giả định nên nhờ nó mà một loạt các vấn đề có thể và sẽ được thay đổi. Nhờ có ngôn ngữ mà một số vấn đề nay đã được giải quyết. Như chúng ta đã biết, việc phát triển phần mềm sẽ không bao giờ đứng yên tại chỗ. Nhiều thứ sẽ không còn cần thiết nữa, trên thực tế, chúng sẽ dần mất đi cũng giống như các hiện tượng xảy ra cùng với quá trình sàng lọc tự nhiên trong hệ thống sinh học. Các kỹ thuật giúp nối liền khoảng cách từ giai đoạn này đến giai đoạn khác cũng trở lên lỗi thời. Quá trình kiểm tra và các công cụ tìm kiếm lỗi cũng không còn cần thiết nữa bởi vì những lỗi này không còn tồn tại. Người ta cũng không còn cần đến các công cụ hỗ trợ việc lập trình thủ công.

So với việc phát triển ứng dụng các công nghệ truyền thống, hiệu suất của các hệ thống phát triển dựa trên công nghệ DBTF cao hơn rất nhiều. Phân tích sâu hơn chút nữa, người ta khám phá ra rằng một hệ thống càng lớn, càng phức tạp, thì hiệu quả càng cao. Điều này hoàn toàn ngược lại những gì chúng ta thường thấy ở các hệ thống ứng dụng những công nghệ truyền thống. Điều này, một phần là do tỷ lệ tái sử dụng cao của các hệ thống ứng dụng công nghệ DBTF. Hệ thống càng

lớn thì nó càng có nhiều khả năng tái sử dụng. Do khả năng tái sử dụng cao nên hiệu suất tiếp tục tăng lên. Việc đo hiệu suất là một quá trình mang tính tương đối, tức là nó được so sánh tương quan với hệ thống được phát triển trước đó.

Việc tận dụng khả năng tái sử dụng trong môi trường DBTF là một lĩnh vực nghiên cứu luôn tiếp diễn khá thú vị. Điều này dẫn đến ý tưởng là các bộ phận được tái sử dụng có thể được phân loại theo nhiều cách. Một cách trong số đó là phân loại theo tính tiết kiệm thời gian (tức là nó ảnh hưởng như thế nào đến chi phí và kế hoạch). Do đó, nhiều thoả hiệp tối ưu hơn được thực hiện. Chúng ta càng biết nhiều về cách thức tái sử dụng các bộ phận thì chúng ta càng có nhiều thông tin để ước lượng chi phí cho toàn bộ hệ thống. Tuy nhiên, cũng cần ghi nhớ rằng, các phương pháp truyền thống trong việc ước lượng thời gian và chi phí phát triển phần mềm không còn hữu ích trong việc đánh giá những hệ thống được phát triển theo các kỹ thuật có tính phòng ngừa.

Ngoài ra, còn có rất nhiều lý do khác lý giải cho hiệu suất cao hơn này của hệ thống, ví dụ như thời gian và chi phí được tiết kiệm do giảm bớt được nhiều quy trình và nhiệm vụ nhờ có việc ứng dụng quan điểm phòng ngừa . Có ít thứ phải học hơn, ít việc phải làm hơn, ít phải phân tích hơn, ít phải thử nghiệm hơn, ít phải quản lý hơn, ít phải tài liệu hóa hơn, ít phải bảo dưỡng hơn và ít phải tích hợp hơn. Điều này là do phần lớn những công việc này đã được tự động hoá hoặc do chúng được thực hiện nội tại nhờ có ngôn ngữ hệ thống chính thống của công nghệ DBTF.

Tóm lại, chính sự kết hợp của công nghệ này với phương pháp thực thi nó đã tạo nền tảng cho sự ra đời của các phần mềm hoàn chỉnh. Phần mềm đóng vai trò quan trọng trong xã hội của chúng ta đến nỗi sự thành bại của nó có ảnh hưởng sâu sắc đến cả sự tồn tại và thành công của một tổ chức. Vì lý do đó mà những quyết định về việc phát triển kỹ thuật hệ thống và phần mềm có ảnh hưởng rất sâu rộng.

Phần mềm là một lĩnh vực còn khá mới mẻ và cần sự đổi mới liên tục. Việc chuyển đổi từ một môi trường phần mềm truyền thống sang một môi trường có tính phòng ngừa cũng tương tự như việc chuyển từ máy đánh chữ sang máy xử lý văn bản. Mỗi khi có bất kỳ một thay đổi lớn nào thì cũng cần phải học hỏi phương thức làm việc mới. Tuy nhiên, nhờ có sự ra đời của máy xử lý văn bản mà sự phát triển lại nối tiếp sự phát triển.

Những kinh nghiệm thu được đã khẳng định chắc chắn rằng chất lượng và hiệu quả sẽ tăng lên cùng với việc gia tăng sử dụng các đặc tính của các hệ thống có tính phòng ngừa. Đối lập với luận điểm "muộn còn hơn không" của quan điểm chạy theo giải quyết các vấn đề đã xảy ra, thì quan điểm có tính phòng ngừa với công nghệ DBTF là giải quyết, hoặc nếu có thể ngăn chặn một vấn đề nào đó càng sớm càng tốt. Việc tìm ra một vấn đề bằng phương pháp tĩnh học tốt hơn việc tìm ra nó bằng phương pháp động học. Ngăn chặn nó theo cách hệ thống được xác định sẽ tốt hơn rất nhiều. Tuy nhiên, ở đây tốt hơn không có nghĩa là không phải xác định hay xây dựng hệ thống.

Việc tái sử dụng một hệ thống tin cậy tốt hơn việc tái sử dụng một hệ thống không tin cậy. Khả năng tái sử dụng tự động hoá tốt hơn khả năng tái sử dụng thủ công. Nhưng khả năng tái sử dụng nội tại lại tốt hơn khả năng tái sử dụng tự động hoá. Khả năng tái sử dụng có thể phát triển tốt hơn khả năng tái sử dụng không có khả năng phát triển. Tuy nhiên, tốt hơn hết vẫn là khả năng tái sử dụng có thể tận dụng tối đa ưu điểm của nó.

Câu trả lời vẫn tiếp tục được giải đáp thông qua những thành công nhận được cũng giống như trong hệ thống sinh học và mục tiêu là những hệ thống của ngày mai sẽ kế thừa được những đặc tính tốt nhất của ngày hôm nay.

Tài liệu tham khảo

[1] Software Engineering Institute. *Capability Maturity Model*, Pittsburgh, PA: Carnegie, Mellon University, 1991.

[2] Stroustrup, B., *The C++ Programming Language*, Reading, MA: Addison-Wesley, 1997.

[3] Gosling, J., Joy, B., and Steele, G., *The Java Language Specification*, Reading, MA: Addison-Wesley, 1996.

[4] Lientz, B.P., and Swanson, E.B., *Software Maintenance Management*, Reading, MA: Addison-Wesley, 1980.

[5] Jones, T.C., *Program Quality and Programmer Productivity*, IBM Tech. Report TR02.764 January: 80, San Jose, CA: Santa Teresa Labs, 1977.

[6] Keyes, J., *Handbook of E-Business*, Chapter F5, Hamilton, M., Defining e...com for e-Profits, RIA, 2000.

[7] Martin, J., and Finkelstein, C.B., *Information Engineering*, Carnforth, Lancs, U.K.: Savant Institute, 1981.

[8] Booch, G., Rumbaugh, J., and Jacobson, I., *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.

[9] Hamilton, M., *"Inside Development Before the Fact,"* Electronic Design, April 4, 1994, ES.

[10] Hamilton, M., *"Development Before the Fact in Action,"* Electronic Design, June 13, 1994, ES.

[11] Keyes, J., *The Ultimate Internet Developers Sourcebook*, AMACOM, to be published Fall 2001.

[12] Software Engineering Tools Experiment-Final Report, Vol. 1, Experiment Summary, Table 1, Page 9, Department of Defense, Strategic Defense Initiative, Washington, D.C., 20301-7100, October 1992.

Định nghĩa các thuật ngữ

Hệ quản lý cơ sở dữ liệu (Data Base Management System - DBMS): Chương trình máy tính được dùng để kiểm soát và truy cập nhanh cơ cở dữ liệu. Một ngôn ngữ được ứng dụng trong hệ thống DBMS để nhằm kiểm soát các chức năng do hệ thống DBMS cung cấp. Ví dụ, SQL là một loại ngôn ngữ được dùng để kiểm soát tất cả các chức năng mà một hệ thống DBMS dựa trên cấu trúc quan hệ cung cấp cho người sử dụng bao gồm chức năng xác định cơ sở dữ liệu, khôi phục, xử lý cơ sở dữ liệu, kiểm soát việc truy cập, chia sẻ dữ liệu và bảo toàn cơ sở dữ liệu.

Giao diện người dùng đồ họa (Graphical User Interface - GUI): Giao diện đồ họa tối ưu nhờ đó một hệ thống triển khai có thể giao diện với máy tính một cách hiệu quả nhất bằng việc ứng dụng các biện pháp trực giác. Giao diện người dùng đồ hoạ cung cấp một loạt những kỹ thuật đồ hoạ, nhiều màu sắc và trực giác giúp người sử dụng có thể xem xét, cập nhật và xử lý thông tin.

Giao diện (Interface): Một điểm truy cập nằm trong ranh giới giữa các đối tượng, chương trình hoặc hệ thống. Ở chính điểm nối này có rất nhiều lỗi xảy ra. Phần mềm có thể giao diện với phần cứng, con người và các phần mềm khác.

Phương pháp luận (Methodology): Một tập hợp các quy trình, quy tắc và sự hình thành việc xây dựng phần mềm.

Metrics: Một loạt các công thức toán học giúp đánh giá hiệu quả và chất lượng.

Cấu trúc phần mềm (Software Architecture): Cấu trúc và mối quan hệ giữa các bộ phận cấu thành phần mềm.

Chuẩn hóa (Formal): Một hệ thống được xác định dựa trên một tập hợp các tiên đề (giả định) đã biết; do đó, nó có cơ sở toán học (ví dụ, một hệ thống DBTF được dựa trên một tập hợp các tiên đề điều khiển). Một số đặc trưng của nó là nó mang tính nhất quán và hoàn thiện về mặt logic. Một hệ thống được coi là nhất quán nếu giả định này không mâu thuẫn với giả định khác của hệ thống. Một hệ thống được coi là hoàn thiện về mặt logic nếu các giả định xác định một cách hoàn toàn được các đặc trưng hệ thống. Điều này đảm bảo rằng một mô hình phương pháp có được tập hợp các đặc trưng đó. Các đặc trưng khác của mô hình được xác định theo phương pháp này có thể không được chứng minh từ những giả định của phương

pháp này. Một hệ thống hoàn toàn logic là một hệ thống có một cơ sở ngữ nghĩa học (ví dụ, cách thức diễn đạt ý nghĩa của các đối tượng nằm trong hệ thống đó). Về mặt ngữ nghĩa học của một hệ thống, điều này có nghĩa là nó không có lỗi giao diện và nó rất rõ ràng, bao gồm những gì cần thiết và đầy đủ, và có sự nhận dạng trạng thái duy nhất.

50

Thu thập và ghi dữ liệu

50.1 Tổng quan 50-1
50.2 Bối cảnh lịch sử 50-1
50.3 Yêu cầu chức năng của thu thập dữ liệu 50-3
50.4 Hệ thống thu thập dữ liệu 50-10
50.5 Kết luận 50-13

Tom Magruder
National Instruments, Inc.

50.1 Tổng quan

Thu thập và ghi dữ liệu là ứng dụng đo lường rất phổ biến. Thu thập dữ liệu cơ bản là đo lường và lưu trữ các thông số vật lý hoặc điện trong một khoảng thời gian nhất định. Các thông số này có thể là nhiệt độ, ứng suất, dịch chuyển, dòng chảy, áp suất, điện áp, dòng điện, điện trở, công suất, hoặc bất kỳ một dạng thông số đo lường nào khác. Các ứng dụng thu thập dữ liệu trong thực tế thường phức tạp hơn so với việc thu thập và ghi tín hiệu đơn thuần. Thông thường, nó liên quan đến sự kết hợp của phân tích trực tuyến, phân tích ngoại tuyến, hiển thị, lập báo cáo và chia sẻ dữ liệu. Bên cạnh đó, nhiều ứng dụng thu thập dữ liệu đã bắt đầu đòi hỏi thu thập và lưu trữ các kiểu dữ liệu khác nhau. Ví dụ như việc ghi lại âm thanh và hình ảnh kết hợp với đo các thông số khác trong thử nghiệm va chạm xe.

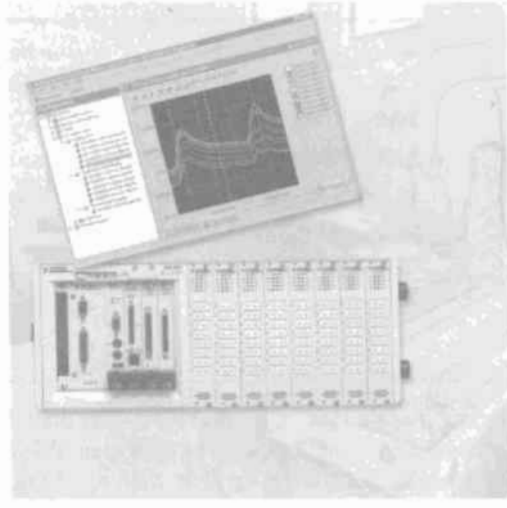
Thu thập dữ liệu được ứng dụng trong rất nhiều lĩnh vực. Các nhà hóa học ghi lại các dữ liệu như nhiệt độ, độ pH và áp suất khi tiến hành các thực nghiệm trong phòng thí nghiệm. Các kỹ sư thiết kế ghi lại các thông số hoạt động của máy móc như độ dao động, nhiệt độ, mức năng lượng để đánh giá bản thiết kế sản phẩm. Các kỹ sư xây dựng thường ghi lại các thông số như ứng suất và tải trọng của những cây cầu theo thời gian để đánh giá độ an toàn của chúng. Nhà địa chất thường dựa trên các dữ liệu thu thập được để phán đoán sự hình thành dầu thô khi khoan dầu. Các nhà máy bia rượu thường ghi lại tình trạng của các kho lưu trữ và ủ rượu bia để nhằm duy trì chất lượng sản phẩm.

Ứng dụng của việc thu thập dữ liệu còn rất nhiều, tuy nhiên tất cả các ứng dụng này đều có những yêu cầu chung giống nhau. Mục đích của chương này là cung cấp một cái nhìn tổng quan về thu thập dữ liệu, thảo luận về các yêu cầu chức năng phổ biến trong hầu hết các ứng dụng thu thập dữ liệu và xem xét một số phần mềm và phần cứng hiện đại cho phép các nhà khoa học và các kỹ sư vận hành những hệ thống thu thập dữ liệu dựa trên PC mạnh.

50.2 Bối cảnh lịch sử

Dạng đầu tiên của thu thập dữ liệu là đo đạc thủ công với các thiết bị đo tương tự như nhiệt kế và áp kế. Các kết quả đo này được ghi theo thời gian thành một bản ghi. Để biết được các xu hướng thay đổi tín hiệu theo thời gian, người ta vẽ lại lên giấy các kết quả đo dưới dạng đồ thị. Vào cuối thế kỷ mười chín, các quá trình này đã bắt đầu được tự động hóa bằng máy móc và các máy ghi biểu đồ lên băng giấy. Máy ghi biểu đồ băng giấy này là những thiết bị tương tự có chức

năng chuyển các xung điện từ cảm biến sang dạng chuyển động cơ khí của một cánh tay. Một chiếc bút được gắn với cánh tay và những cuộn giấy dài được chuyển động ở một tốc độ không đổi bên dưới chiếc bút. Kết quả nhận được trên giấy là một biểu đồ hiển thị các thông số đo được trong một khoảng thời gian nhất định. Máy ghi biểu đồ băng giấy này là một bước nhảy vĩ đại so với ban ghi bằng tay, tuy nhiên nó vẫn còn nhiều nhược điểm. Ví dụ, việc chuyển đổi các đường vẽ trên giấy thành các thông số kỹ thuật là một việc làm rất buồn tẻ và các dữ liệu được ghi lại tốn rất nhiều giấy.



HÌNH 50.1 Ứng dụng thu thập dữ liệu dựa trên PC

Cùng với sự phát triển của máy tính cá nhân trong các thập niên 70 và 80, người ta đã bắt đầu sử dụng máy tính phục vụ cho việc phân tích, lưu trữ dữ liệu và lập báo cáo. Nhu cầu đưa dữ liệu vào PC tạo ra một kiểu thiết bị thu thập dữ liệu mới, đó là bộ thu thập dữ liệu. Bộ thu thập dữ liệu là một thiết bị dạng hộp đơn lẻ (stand alone box) dùng để đo các tín hiệu, chuyển chúng sang dạng dữ liệu số và lưu trữ. Dữ liệu này phải được đưa vào PC để phân tích, lưu trữ lâu dài và lập báo cáo. Dữ liệu thường được đưa vào PC một cách thủ công bằng thiết bị lưu trữ như đĩa mềm từ bộ thu thập dữ liệu đến PC hay bằng cách nối bộ thu thập dữ liệu với PC thông qua một số kết nối truyền thông, như nối tiếp hoặc mạng cục bộ.

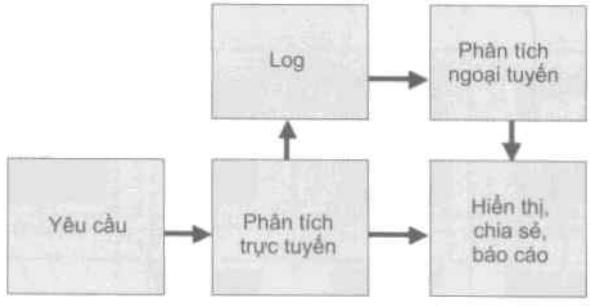
Thập niên 1990 đã chứng kiến sự phát triển mang tính cách mạng của thu thập dữ liệu khi người ta bắt đầu tạo ra những hệ thống thu thập dữ liệu dựa trên PC. Những hệ thống này kết hợp khả năng thu thập và lưu trữ của một bộ thu thập dữ liệu đơn lẻ với khả năng lưu trữ, phân tích, báo cáo và hiển thị của các PC hiện đại. Những hệ thống thu thập dữ liệu dựa trên PC đã tự động hóa toàn bộ quá trình thu thập dữ liệu. Việc phát triển hệ thống thu thập dữ liệu dựa trên PC được thúc đẩy bởi ba cải tiến công nghệ sau:

1. Tăng độ tin cậy của PC
2. Giảm đáng kể chi phí dung lượng ổ cứng của PC
3. Phần cứng đo lường dựa trên PC có thể bằng hoặc vượt khả năng đo lường của các bộ thu thập dữ liệu đơn lẻ

Ngày nay, các hệ thống thu thập dữ liệu dựa trên PC cung cấp rất nhiều kiểu đo, khả năng phân tích và công cụ lập báo cáo. Phần còn lại của chương này sẽ tập trung vào chức năng cần thiết để thực hiện một hệ thống thu thập dữ liệu dựa trên PC.

50.3 Yêu cầu chức năng của thu thập dữ liệu

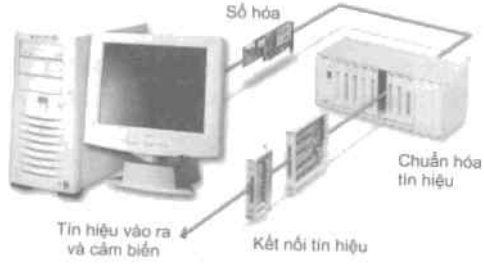
Các ứng dụng thu thập dữ liệu, ngay từ thế kỷ 15 khi các thầy tu ghi lại thủ công các dạng thời tiết cho đến thế kỷ 21 khi các nhà vật lý ghi lại các thông số thực nghiệm của một thí nghiệm lò phản ứng nóng chảy, có thể được chia thành một tập hợp năm yêu cầu chức năng thông thường, như được minh họa trong hình 50.2. Thu thập dữ liệu là quá trình đo lường thực tế các thông số vật lý và đưa chúng vào hệ thống lưu trữ. Phân tích trực tuyến là quá trình xử lý dữ liệu ngay khi thu thập chúng, bao gồm cảnh báo, chuẩn hóa dữ liệu và đôi khi là điều khiển. Thu thập hoặc lưu trữ dữ liệu là một yêu cầu hiển nhiên của mọi hệ thống thu thập dữ liệu. Phân tích ngoại tuyến là tất cả hoạt động xử lý dữ liệu sau khi thu thập chúng nhằm tách các thông tin hữu ích ra khỏi chúng. Khối chức năng cuối cùng bao gồm việc hiển thị, báo cáo và chia sẻ dữ liệu. Đó là tất cả những yêu cầu "phức tạp" hình thành nên chức năng của một hệ thống thu thập dữ liệu. Hãy tìm hiểu xem mỗi chức năng này được thực hiện như thế nào trong hệ thống thu thập dữ liệu dựa trên PC hiện đại.



HÌNH 50.2 Phần tử cơ bản của hệ thống thu thập dữ liệu

Thu thập dữ liệu

Chức năng thu thập dữ liệu là một trong những thành phần quan trọng nhất của mọi hệ thống thu thập dữ liệu. Trong một hệ thống dựa trên PC, việc thu thập dữ liệu được thực hiện bằng phần cứng đo lường. Phần cứng này có thể được chia thành cảm biến, kết nối tín hiệu, chuẩn hóa tín hiệu và biến đổi tương tự số (A/D), như thấy ở hình 50.3. Các chủ đề này được trình bày chi tiết hơn trong các chương khác của cuốn sách này, do đó chương này chỉ trình bày các vấn đề mang tính tổng quan nhất.



HÌNH 50.3 Phần cứng đo lường của hệ thống thu thập dữ liệu dựa trên PC

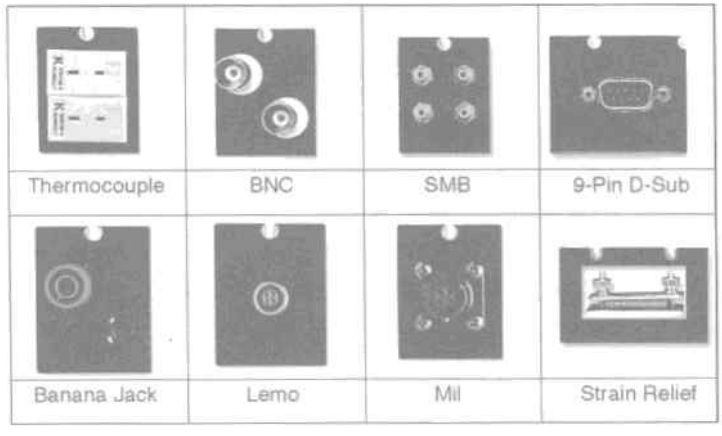
Cảm biến

Rất nhiều cảm biến được dùng để chuyển đổi các thông số vật lý thành các tín hiệu điện. Cảm biến nhiệt độ như là cặp nhiệt, RTD hay nhiệt kế là một vài trong số các cảm biến thông dụng nhất được dùng trong các ứng dụng thu thập dữ liệu. Các cảm biến phổ biến khác là lưu lượng kế, bộ

chuyển đổi áp suất, máy đo ứng suất, gia tốc kế, và micrô. Phần lựa chọn và cài đặt của các cảm biến không được trình bày trong chương này.

Kết nối tín hiệu

Sau khi các cảm biến được cài đặt, chúng phải được kết nối với hệ thống thu thập dữ liệu. Kết nối tín hiệu là một bộ phận trong phần cứng đo lường, cho phép kết nối các cảm biến với hệ thống thu thập dữ liệu. Đầu nối ren cho phép kết nối trực tiếp các dây trần từ cảm biến đến hệ thống thu thập, đây là dạng kết nối cơ bản nhất. Đầu nối ren là sự lựa chọn hợp lý cho các ứng dụng đa năng, đặc biệt khi cần kết nối một số lượng lớn các tín hiệu vào một không gian nhỏ. Nhược điểm của đầu nối ren là mất nhiều thời gian trong việc kết nối và khó tái cấu hình. Hình 50.4 giới thiệu một số liên kết chuẩn khác được thiết kế để nối và ngắt các cảm biến. Bộ kết nối cặp nhiệt mini là một lựa chọn được sử dụng rộng rãi cho cặp nhiệt độ. Bộ kết nối BNC và SMB được dùng phổ biến khi cần cách điện để loại trừ âm thanh. Banana Jack thường được dùng khi đo dòng điện, điện trở hoặc điện áp cao hơn. Các nhà cung cấp cảm biến thường định rõ những loại kết nối phù hợp với mỗi cảm biến và người dùng chỉ phải chọn phần cứng đo lường có thể chấp nhận kết nối đó.



HÌNH 50.4 Các loại kết nối tín hiệu

Chuẩn hóa tín hiệu

Chuẩn hóa tín hiệu là một trong những thành phần quan trọng nhất và cũng rất dễ bị bỏ qua nhất của hệ thống thu thập dựa trên PC. Rất nhiều cảm biến cần công nghệ chuẩn hóa tín hiệu đặc biệt. Ví dụ như tín hiệu thu được từ cặp nhiệt có điện áp rất thấp và yêu cầu có bộ khuếch đại, bộ lọc và sự tuyến tính hóa. Các cảm biến khác như là máy đo lực căng và gia tốc kế cần năng lượng bổ sung để khuếch đại và lọc, trong khi các tín hiệu khác có thể yêu cầu sự cách ly để bảo vệ hệ thống khỏi điện áp cao. Không một bộ thu thập dữ liệu đơn lẻ nào có được tính linh hoạt cần thiết để có thể thực hiện tất cả các phép đo này. Tuy vậy, với chuẩn hóa tín hiệu ngoại vi, ta có thể kết hợp các công nghệ cần thiết để đưa các dạng tín hiệu khác nhau này vào một hệ thống thu thập dữ liệu dựa trên PC.

Phần lớn các tín hiệu đều cần một số sự chuẩn bị trước khi được số hóa. Như đã trình bày ở trên, các tín hiệu từ cặp nhiệt độ có mức điện áp rất thấp, vì vậy chúng phải được khuếch đại trước khi số hóa. Các cảm biến khác, như RTD, nhiệt điện trở, đo lực căng, hoặc gia tốc kế yêu cầu năng lượng điện để vận hành. Thậm chí những tín hiệu điện áp thuần túy cũng cần những công nghệ đặc biệt để ngăn chặn những tín hiệu lớn hoặc giúp an toàn trong việc đo lường tín hiệu điện áp cao. Tất cả những công nghệ chuẩn bị này là các dạng của chuẩn hóa tín hiệu. Do có nhiều công nghệ chuẩn hóa tín hiệu khác nhau nên vai trò và sự cần thiết của mỗi công nghệ này có thể gây nên sự nhầm lẫn. Dưới đây là danh mục các kiểu chuẩn hóa tín hiệu phổ biến, chức năng và các ví dụ.

- *Khuếch đại.* Khi mức điện áp đo là rất thấp, khuếch đại được sử dụng để tăng tối đa khả năng của bộ số hóa. Bằng cách khuếch đại tín hiệu đầu vào, tín hiệu được chuẩn hóa sử dụng hiệu quả hơn phạm vi cho phép của bộ biến đổi A/D. Điều đó cho phép làm tăng độ chính xác và độ phân giải của phép đo. Các cảm biến thông thường yêu cầu có bộ khuếch đại là cặp nhiệt độ và cảm biến đo độ căng.
- *Suy giảm.* Sự suy giảm đối lập với sự khuếch đại. Nó cần thiết khi điện áp để số hóa nằm ngoài phạm vi đầu vào của bộ số hóa. Dạng chuẩn hóa tín hiệu này ngăn tín hiệu đầu vào sao cho tín hiệu được chuẩn hóa nằm trong phạm vi của bộ biến đổi A/D. Sự suy giảm là cần thiết khi đo lường điện áp cao.
- *Cách ly.* Các tín hiệu điện áp nằm ngoài phạm vi của bộ số hóa có thể phá hủy hệ thống đo lường và gây tổn hại cho bộ điều khiển. Vì lý do này, cách ly được yêu cầu kết hợp với suy giảm nhằm bảo vệ hệ thống và người sử dụng khỏi điện áp nguy hiểm hoặc các xung điện áp. Việc cách ly cũng rất cần thiết khi các cảm biến ở mức điện áp nổi đất khác với các cảm biến đo lường (giống như việc lắp đặt cặp nhiệt độ trên một động cơ).
- *Dồn kênh.* Thông thường, bộ số hóa là phần đắt nhất của một hệ thống thu thập dữ liệu. Dồn kênh cho phép tự động đưa nhiều tín hiệu vào một bộ số hóa đơn. Đây là một giải pháp mang tính kinh tế giúp mở rộng số lượng tín hiệu trên hệ thống. Dồn kênh là cần thiết cho các ứng dụng có số lượng kênh lớn.
- *Lọc.* Lọc có chức năng loại bỏ các thành phần tần số không mong muốn ra khỏi một tín hiệu. Điều này giúp ngăn chặn chồng phổ tín hiệu và giảm bớt tín hiệu nhiễu. Các cặp nhiệt độ thường yêu cầu phải có một bộ lọc thông thấp để loại bỏ nhiễu nguồn ra khỏi các tín hiệu thu được. Các đo lường dao động thường yêu cầu phải có một bộ lọc thông thấp tần số cao hơn để loại bỏ các thành phần tín hiệu tần số cao nằm ngoài phạm vi của hệ thống thu thập dữ liệu.
- *Kích thích.* Nhiều loại cảm biến, bao gồm RTD, cảm biến đo độ căng và gia tốc kế yêu cầu có một số dạng năng lượng để thực hiện phép đo. Kích thích là công nghệ chuẩn hóa tín hiệu được sử dụng để cung cấp loại năng lượng này. Kích thích này có thể là một nguồn điện áp hoặc dòng điện, tùy vào từng loại cảm biến.
- *Tuyến tính hóa.* Một số dạng cảm biến sinh ra các tín hiệu điện áp không có quan hệ tuyến tính với đại lượng vật lý mà chúng đo. Tuyến tính hóa là quá trình chuyển đổi tín hiệu từ cảm biến thành thông số vật lý. Việc này có thể được thực hiện bằng phương pháp chuẩn hóa tín hiệu hoặc phần mềm. Cặp nhiệt độ là một ví dụ điển hình về loại cảm biến yêu cầu phải có tuyến tính hóa.
- *Bù nhiệt.* Một công nghệ khác rất cần cho quá trình đo của cặp nhiệt độ là bù nhiệt ở mối nối "lạnh" (CJC). Khi cặp nhiệt được nối với hệ thống thu thập dữ liệu, nhiệt độ của kết nối đó phải được biết để tính toán nhiệt độ thực mà cặp nhiệt độ đang đo. Một cảm biến CJC phải được đặt tại vị trí của các kết nối.
- *Lấy mẫu đồng thời.* Khi cần phải đo nhiều tín hiệu vào cùng một thời điểm, việc lấy mẫu đồng thời là cần thiết. Chuẩn hóa tín hiệu đầu vào có thể cung cấp một giải pháp lấy mẫu đồng thời hiệu quả về giá thành hơn việc đầu tư một bộ số hóa có khả năng này. Các ứng dụng tiêu biểu có thể yêu cầu lấy mẫu đồng thời bao gồm đo dao động và đo độ lệch pha.

Hầu hết các cảm biến yêu cầu sự kết hợp của các công nghệ chuẩn hóa tín hiệu ở trên. Một lần nữa, cặp nhiệt độ là một ví dụ điển hình vì nó cần khuếch đại, tuyến tính hóa, bù nhiệt, lọc và đôi khi cả cách ly. Về lý tưởng, một hệ thống thu thập dữ liệu dựa trên PC tốt cung cấp khả năng lựa chọn dạng chuẩn hóa tín hiệu cần thiết cho các ứng dụng. Trong một số hệ thống, chuẩn hóa tín hiệu đầu vào là một sự lựa chọn, nhưng trong các hệ thống khác, chuẩn hóa tín hiệu đầu vào là cần thiết để thực hiện các đo lường được yêu cầu. Về cơ bản, hệ thống đo lường nên bao gồm phần chuẩn hóa tín hiệu đầu vào nếu các loại thiết bị sau được sử dụng: cặp nhiệt độ, RTD, nhiệt điện trở, cảm biến đo độ căng, LVDT, gia tốc kế, công tắc, dồn kênh, các tín hiệu điện áp cao/thấp hỗn hợp, đầu vào dòng điện, hoặc đầu vào điện trở.

Chuyển đổi

Sau khi các thông số vật lý được chuyển đổi thành các tín hiệu điện và được chuẩn hóa, các tín hiệu điện tương tự sẽ được chuyển đổi thành các giá trị số và đưa các giá trị đó trở lại máy tính. Quá trình chuyển đổi A/D có thể được thực hiện với một bo mạch thu thập dữ liệu gắn liền hoặc nó có thể được tích hợp bên trong một thiết bị đơn lẻ có khả năng chuẩn hóa và kết nối.

Sự kết hợp các cảm biến, khả năng kết nối tín hiệu, chuẩn hóa tín hiệu, và chuyển đổi A/D tạo nên phần cứng đo lường của hệ thống thu thập dữ liệu. Trong một hệ thống dựa trên PC, phần cứng đo lường được cấu hình và điều khiển thông qua phần mềm và đó chính là điều kiện sử dụng phần mềm được thiết kế để tích hợp hoàn hảo tất cả các bộ phận cấu thành trong hệ thống thu thập dữ liệu.

Phân tích trực tuyến

Thành phần chức năng tiếp theo trong hệ thống thu thập dữ liệu điển hình là phân tích trực tuyến. Trong hệ thống dựa trên PC, phân tích trực tuyến được thực hiện thông qua phần mềm. Có nhiều dạng phân tích trực tuyến khác nhau trong các ứng dụng thu thập dữ liệu. Ở đây chúng ta sẽ trình bày một số dạng phổ biến nhất.

Chuyển đổi thang đo (scaling) là việc biến đổi giá trị nhị phân đầu ra của hệ thống thu thập dữ liệu thành các giá trị đo có tỷ lệ hợp lý với thiết bị kỹ thuật phù hợp. Một ví dụ là việc tính toán nhiệt độ từ một đầu đo nhiệt độ. Bộ số hóa trả lại các kết quả đo dạng nhị phân của điện áp cặp nhiệt độ và điện áp bù nhiệt. Phần mềm chuyển đổi giá trị đo dạng nhị phân thành điện áp và sau đó sử dụng công thức chuyển đổi để tính toán nhiệt độ. Tương tự các phương pháp chuyển đổi thang đo cũng được dùng cho cảm biến đo độ căng, RTD, gia tốc kế và các loại khác. Các phần mềm đo lường dựa trên PC tiên tiến thực hiện hầu hết các chức năng chọn thang đo một cách tự động.

Một chức năng phân tích trực tuyến quan trọng khác là cảnh báo và quản lý sự kiện. Nó bao gồm việc giám sát kênh và tạo ra một số thông báo nếu các giới hạn định sẵn bị vượt quá. Thông báo này có thể đơn giản như bật đèn cảnh báo hoặc phức tạp như nhắn cho ai đó thông tin về vấn đề xảy ra. Cảnh báo cũng bao gồm một đáp ứng tự động đối với các sự kiện nào đó. Ví dụ, hệ thống thu thập dữ liệu có thể tắt một máy đang được giám sát nếu nhiệt độ dầu bôi trơn vượt quá giá trị giới hạn nào đó.

Các chức năng phân tích trực tuyến có thể được sử dụng trong các ứng dụng thu thập dữ liệu khác nhau. Chức năng này có thể bao gồm các hệ thống điều khiển phản hồi hoặc phân tích tín hiệu cao cấp. Chỉ có các hệ thống thu thập dữ liệu dựa trên PC mới có được khả năng linh hoạt để đáp ứng được các yêu cầu khác nhau này.

Thu thập và lưu trữ dữ liệu

Theo định nghĩa, khối chức năng thu thập và lưu trữ dữ liệu là cần thiết đối với mọi hệ thống thu thập dữ liệu. Các phương pháp lưu trữ dữ liệu cũng khác nhau trong các hệ thống khác nhau. Máy ghi biểu đồ dạng băng sử dụng giấy, bộ thu thập dữ liệu truyền thống có thể sử dụng bộ nhớ ổn định (nonvolatile memory), đĩa mềm hoặc một loạt các thiết bị trung gian khác. Hệ thống thu thập dữ liệu dựa trên PC thường xuyên sử dụng đĩa cứng của PC, mặc dù chúng cũng có thể dùng ổ băng từ, ổ đĩa mạng, ổ đĩa RAID và nhiều lựa chọn khác.

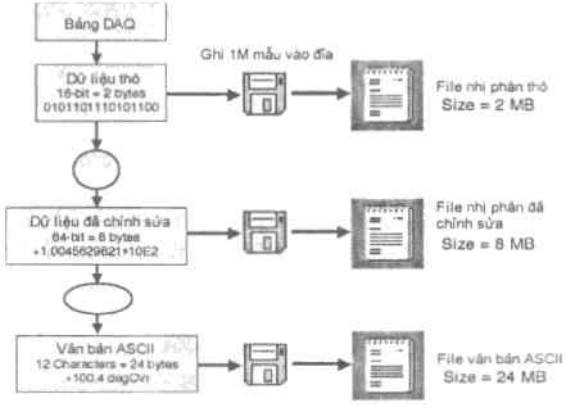
Phần mềm đóng vai trò rất quan trọng trong hệ thống thu thập dữ liệu dựa trên PC do phần mềm thu thập dữ liệu được xây dựng tốt sẽ quyết định việc lưu trữ dữ liệu như thế nào, tốc độ thu thập dữ liệu lên đĩa ra sao và hiệu quả sử dụng không gian lưu trữ như thế nào. Phần mềm thu thập dữ liệu cũng cung cấp khả năng quản lý dữ liệu, ví dụ như là việc thay đổi định dạng dữ liệu, cấu trúc dữ liệu và liên kết với các cơ sở dữ liệu.

Định dạng lưu trữ dữ liệu có mối liên hệ chặt chẽ với sự hoạt động và tính dễ dàng sử dụng của hệ thống thu thập dữ liệu. Có 3 loại định dạng phổ biến hay được sử dụng để lưu trữ trong hệ thống thu thập dữ liệu là: tệp văn bản dạng ASCII, tệp nhị phân, và cơ sở dữ liệu.

Tệp văn bản dạng ASCII là dạng lưu trữ dữ liệu phổ biến và linh hoạt nhất. Các tệp văn bản trong các ứng dụng thu thập dữ liệu thường được tạo thành bởi một đoạn tiêu đề và các cột dữ liệu. Phần tiêu đề cung cấp các thông tin như tên kênh, đơn vị, thiết bị kiểm tra và ghi chú của người sử dụng. Cột dữ liệu đầu tiên thường là cột thời gian của mỗi mẫu và tiếp theo là các cột khác cho mỗi kênh đang được thu thập. Tệp văn bản được sử dụng rộng rãi bởi vì chúng có thể được mở hoặc nhập vào hầu hết các gói phần mềm và chúng cũng dễ dàng được truyền giữa các hệ điều hành với nhau. Một số nhược điểm của tệp dạng văn bản là chúng sử dụng không gian đĩa kém hiệu quả và chúng cũng cần xử lý bổ sung trước khi ghi và đọc từ các tệp. Tệp văn bản dạng ASCII được dùng rộng rãi khi tốc độ thu thập dữ liệu thấp (<1000 mẫu trên 1 giây), tổng số dữ liệu được thu thập không quá lớn và người sử dụng cần dễ dàng chia sẻ dữ liệu giữa các ứng dụng phần mềm khác nhau.

Tệp nhị phân là một phương pháp lưu trữ dữ liệu hiệu quả nhất. Với tệp nhị phân, các byte mà máy tính sử dụng để lưu trữ dữ liệu trong bộ nhớ được ghi trực tiếp lên tệp. Dữ liệu này chiếm ít không gian hơn các thông tin được ghi dưới dạng văn bản ASCII và nó ít phải xử lý hơn so với định dạng thành văn bản. Tệp nhị phân không thể xem được bằng các phần mềm ứng dụng phổ thông như là MS Excel. Thay vào đó, chúng phải được chuyển đổi bằng bằng một chương trình phần mềm thành dạng dữ liệu có nghĩa. Với hệ thống thu thập dữ liệu dựa trên PC, ta có thể thu thập dữ liệu chuẩn vừa được xử lý theo đơn vị kỹ thuật chính xác hoặc cũng có thể thu thập các giá trị nhị phân thô được lấy từ bộ số hóa. Những giá trị nhị phân thể hiện việc chuyển đổi A/D của mỗi mẫu được lấy từ thiết bị thu thập dữ liệu (DAQ) 16 bit chiếm 16 bit, hoặc 2 byte của bộ nhớ. Chương trình chọn thang đo trong phần mềm thu thập sẽ tự động chuyển đổi dữ liệu thô này thành một số thực biểu diễn giá trị vật lý đo được. Dữ liệu sau đó được lưu trữ bên trong phần mềm thu thập dữ liệu như là một giá trị dấu phẩy động, nó là kiểu dữ liệu chiếm 8 byte của bộ nhớ trong hầu hết các hệ thống máy tính.

Do mục đích thực hiện, một số hệ thống thu thập dữ liệu tốc độ cao có thể thu thập các giá trị nhị phân ban đầu lên đĩa, cùng với hệ số tỷ lệ cần thiết để sau đó thực hiện chuyển đổi thang đo. Hình 50.5 cho thấy mối liên hệ giữa dữ liệu nhị phân ban đầu, nhị phân đã được chuyển thang đo và văn bản ASCII. Tệp nhị phân chiếm ít không gian và cho phép cải thiện tốc độ ghi lên đĩa. Tệp nhị phân ban đầu có thể chiếm ít hơn 1/10 kích thước của tệp văn bản chứa cùng một thông tin. Nhược điểm của tệp nhị phân là chúng thường phải chuyển đổi sang định dạng khác trước khi được chia sẻ giữa các ứng dụng khác nhau.



HÌNH 50.5 Ví dụ về kích thước tệp lưu trữ dữ liệu

Nhiều phần mềm thu thập dữ liệu ghi dữ liệu vào cơ sở dữ liệu. Cơ sở dữ liệu thường là các tệp nhị phân cung cấp một định dạng cấu trúc để chèn và lấy dữ liệu. Chúng được tối ưu hóa để quản lý có hiệu quả một lượng lớn dữ liệu và để tìm kiếm các thông tin trong cơ sở dữ liệu mà không cần phải nạp tất cả mọi thứ vào bộ nhớ. Cơ sở dữ liệu cũng thường được thiết kế để cho phép sao chép dự phòng và lưu trữ dữ liệu một cách dễ dàng và cho phép nhiều người sử dụng cùng một

lúc. Chúng thường có các giải pháp phần mềm cho phép dễ dàng nhập dữ liệu vào các gói phần mềm khác nhau để phân tích và lập báo cáo. Trong nhiều trường hợp, cơ sở dữ liệu là định dạng lưu trữ dữ liệu lý tưởng cho hệ thống thu thập dữ liệu dựa trên PC. Hai nhược điểm của việc sử dụng cơ sở dữ liệu để lưu trữ là chúng làm tăng độ phức tạp và khó khăn trong thực thi nếu bắt đầu từ vạch xuất phát.

Nhiều loại phương tiện lưu trữ khác nhau được sử dụng để thu thập dữ liệu. Bộ thu thập dữ liệu đơn lẻ có thể sử dụng đĩa mềm, thẻ nhớ PCMCIA, băng từ hoặc bất kỳ một lựa chọn nào khác. Hệ thống thu thập dữ liệu dựa trên PC thường phụ thuộc vào ổ cứng trong máy tính. Điều đó là khả thi do xu thế phát triển của ổ cứng ngày càng tin cậy hơn và khả năng lưu trữ cũng cao hơn. Ngày nay, các ổ cứng có dung lượng lên đến 20 GB (và lớn hơn) đã làm nó trở thành một trong những thiết bị lưu trữ kinh tế nhất. Chúng vẫn thích hợp cho việc sao lưu định kỳ hoặc lưu trữ dữ liệu.

Các ứng dụng thu thập dữ liệu tốc độ cao có thể vượt quá tốc độ ghi lên đĩa của các ổ cứng PC thông thường. Một trong những ưu điểm của hệ thống thu thập dựa trên PC là có thể chuyển đến các thiết bị lưu trữ có hiệu suất cao hơn và các máy tính có khả năng hoạt động tốt hơn, với một chút hoặc không phải điều chỉnh phần mềm thu thập hoặc phần cứng đo lường. Một dạng của thiết bị lưu trữ hiệu suất cao là bộ điều khiển RAID (Hệ thống đĩa dự phòng - redundant array of independent disks). Bộ điều khiển RAID sử dụng nhiều ổ cứng được phối hợp để tăng tốc độ ghi lên đĩa tổ hợp và cung cấp dữ liệu một cách nguyên vẹn. Thiết bị nghe nhìn AV (Audio-visual) là một dạng khác của thiết bị lưu trữ được sử dụng cho việc thu thập dữ liệu tốc độ cao. Thiết bị AV được tối ưu cho việc truyền một lượng lớn thông tin về âm thanh và hình ảnh lên đĩa và sự tối ưu đó cũng làm cho chúng thích hợp hơn trong các ứng dụng ghi dữ liệu hiệu suất cao. Cuối cùng, một số công ty tạo ra phần cứng cho phép các thiết bị DAQ truyền dữ liệu qua bus PCI của máy tính, trực tiếp lên thiết bị lưu trữ. Tốc độ truyền của những thiết bị này bị giới hạn bởi dải thông hiện có của bus PCI, với tốc độ lớn nhất theo lý thuyết là 132 MB/s trên phần lớn các máy tính.

Phân tích ngoại tuyến

Phân tích ngoại tuyến là thực hiện các hàm toán học trên dữ liệu sau khi chúng được thu thập nhằm lấy ra các thông tin quan trọng. Các dạng phân tích ngoại tuyến có thể bao gồm thống kê cơ bản về các thông số đo lường trên cơ sở của máy tính cũng như nhiều chức năng nâng cao như danh mục tần số của tín hiệu và phân tích thứ tự. Phân tích ngoại tuyến có thể được tích hợp với phần còn lại của ứng dụng thu thập dữ liệu hoặc được thực hiện riêng thông qua gói phần mềm phân tích đơn lẻ. Trong nhiều trường hợp, phân tích ngoại tuyến được kết hợp với việc lập báo cáo, hiển thị dữ liệu và các chức năng chia sẻ dữ liệu.

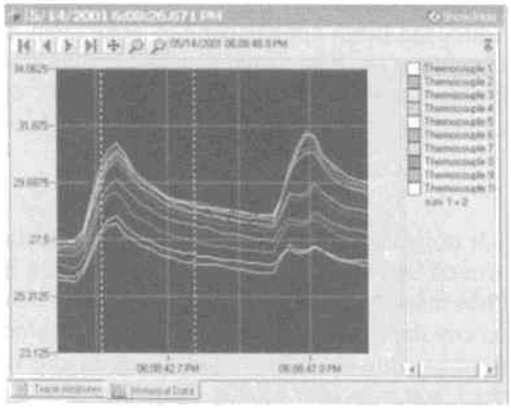
Hiển thị

Phần lớn các ứng dụng thu thập dữ liệu yêu cầu một số dạng hiển thị để quan sát các giá trị đo lường đang được lưu trữ. Chức năng hiển thị có thể được chia thành quan sát dữ liệu trực tiếp và dữ liệu trong quá khứ. Hiển thị dữ liệu trực tiếp là rất cần thiết nếu cần quan sát dữ liệu trong khi nó đang được thu thập. Nhiều bộ thu thập dữ liệu đơn lẻ có bộ hiển thị dữ liệu trực tiếp được tích hợp trong cùng một hộp. Hiển thị dữ liệu trong quá khứ cho phép quan sát các dữ liệu đã được thu thập trước đây. Phần lớn các bộ thu thập dữ liệu đơn lẻ yêu cầu phải chuyển dữ liệu tới PC để quan sát dữ liệu quá khứ. Các ứng dụng thu thập dữ liệu dựa trên PC cho phép kết hợp cả hiển thị dữ liệu trực tiếp và hiển thị dữ liệu quá khứ vào trong cùng một giao diện người dùng. Các tiện ích quan sát dữ liệu cần cung cấp giao diện người dùng trực quan, khả năng cuộn và phóng to, con trỏ và các đặc tính tùy chọn chung. Hình 50.6 là một ví dụ hiển thị dữ liệu trong quá khứ tiêu biểu.

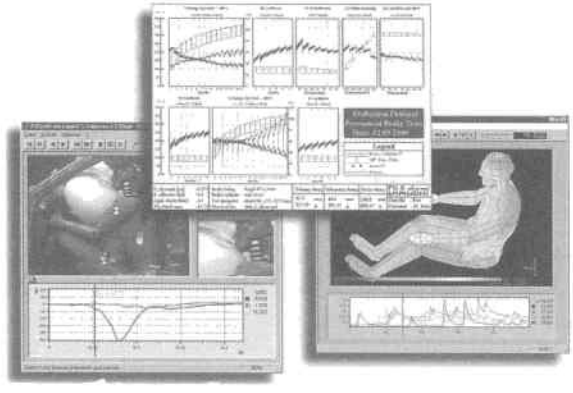
Lập báo cáo

Lập báo cáo là một chức năng thường không được coi là một phần trong các ứng dụng thu thập dữ liệu. Trong thực tế, phần lớn ứng dụng thu thập dữ liệu cần một số dạng khả năng lập báo cáo, với lý do đơn giản là nếu đang thu thập dữ liệu, một số người cần xem nó trong một định dạng có

thể đọc được thì cần phải có một báo cáo. Chức năng lập báo cáo có thể được tích hợp vào trong các ứng dụng thu thập dữ liệu dựa trên PC nhằm tăng hiệu suất. Ứng dụng thu thập có thể được thiết lập để lập các báo cáo theo định kỳ và phân phối chúng đến đối tượng thích hợp.



HÌNH 50.6 Ví dụ hiển thị dữ liệu lịch sử của phần mềm National Instruments VI Logger



HÌNH 50.7 Khả năng lập báo cáo của gói phần mềm National Instruments DIAdem

Hiện nay có những phần mềm thương mại rất mạnh, chúng được thiết kế để cung cấp khả năng phân tích dữ liệu và lập báo cáo từ các thông số đo. Hình 50.7 đưa ra ví dụ về khả năng lập báo cáo có thể có của gói phần mềm thương mại này. Khi lựa chọn phần mềm để lập báo cáo, điều quan trọng là nó phải tích hợp một cách hoàn hảo với phần còn lại của phần mềm thu thập dữ liệu. Trong trường hợp lý tưởng, phần mềm thu thập dữ liệu nên có khả năng truyền dữ liệu trực tiếp đến ứng dụng lập báo cáo và tự động lập báo cáo.

Chia sẻ và xuất bản dữ liệu

Để dữ liệu được ghi trở nên hữu ích, thì nó phải được đưa đến đúng người cần. Với những khả năng nối mạng của các phần mềm thu thập dữ liệu hiện đại, việc chia sẻ dữ liệu và đưa nó lên mạng không yêu cầu nhiều kiến thức về khoa học máy tính nữa. Ứng dụng thu thập có thể được thiết lập để xuất bản dữ liệu trực tiếp lên mạng khi nó nhận được dữ liệu, gửi e-mail định kỳ dữ liệu thô và kết quả phân tích tới người nhận hoặc tự động đưa các báo cáo lên trang Web.

Trong các ứng dụng thu thập dữ liệu được phân tán rộng rãi, mỗi điểm thu thập có thể xuất bản các kết quả đo của nó lên mạng và máy chủ có thể đóng vai trò là trung tâm thu nhận. Máy tính trung tâm nhận các thông số đo lường từ mỗi điểm, kết hợp chúng để phân tích, lưu trữ kết quả lâu dài và lập báo cáo phân tích dữ liệu theo định kỳ.

50.4 Hệ thống thu thập dữ liệu

Chúng ta đã đề cập đến các thành phần chức năng của hệ thống thu thập dữ liệu, hãy xem xét xem những bộ phận này được thực hiện như thế nào trong các hệ thống thực. Tất cả các hệ thống thu thập dữ liệu dựa trên PC được tạo thành từ phần cứng và phần mềm. Phần cứng đo lường quản lý phần thu thập dữ liệu của ứng dụng thu thập và phần cứng xác định số kênh, dạng cảm biến, tốc độ thu thập và độ chính xác phép đo. Phần mềm đo lường ngoài việc điều khiển phần cứng còn quản lý phân tích trực tuyến, thu thập, phân tích ngoại tuyến, hiển thị, báo cáo và chia sẻ dữ liệu.

Tùy chọn phần mềm

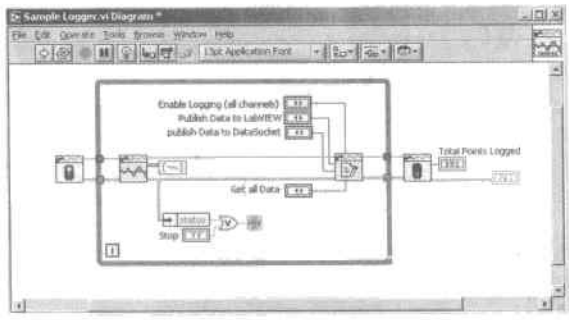
Lựa chọn phần mềm là một trong các bước quan trọng nhất khi xác định một hệ thống thu thập dữ liệu dựa trên PC. Phần mềm có tính chất quyết định hệ thống lưu trữ nhằm cung cấp nhiều tiện ích và giải pháp mềm dẻo. Phần mềm đo lường phải được thiết kế để tích hợp liền mạch với phần cứng. Ngoài nhiệm vụ cơ bản của thu thập dữ liệu và ghi nó lên đĩa, phần mềm nên cung cấp các công cụ để quản lý cấu hình phần cứng đo lường, chọn thang đo cho các kênh và căn chỉnh hệ thống. Phần mềm nên cho phép thực hiện toàn bộ ứng dụng, bao gồm lập báo cáo, phân tích, lưu trữ và chia sẻ. Có 2 loại phần mềm có thể được dùng cho các ứng dụng thu thập dữ liệu dựa trên PC - phần mềm trọn gói (turnkey software), còn được gọi là phần mềm dựa trên cấu hình và các môi trường phát triển ứng dụng.

Các gói phần mềm trọn gói sẵn sàng chạy trên các ứng dụng phần mềm thu thập dữ liệu, nó kết nối với phần cứng đo lường để thu thập dữ liệu. Những ứng dụng này cung cấp môi trường làm việc thân thiện nhằm thực hiện nhanh hơn các công việc cấu hình hệ thống, khởi động và thực thi. Một gói phần mềm thu thập dữ liệu dựa trên cấu hình tốt sẽ cung cấp:

- *Giao diện người dùng trực giác.* Cấu hình phần mềm nên thực hiện qua một giao diện dựa trên Windows, điều khiển bằng menu với các chức năng và hướng dẫn trợ giúp truy cập dễ dàng.
- *Tự động thu thập và lưu trữ dữ liệu.* Một trong các chức năng cơ bản của bất kỳ gói phần mềm thu thập dữ liệu nào là quản lý việc lưu trữ dữ liệu. Nó nên tự động lưu trữ dữ liệu một cách có hiệu quả, và phần mềm nên cung cấp một phương pháp để sao lưu và cất giữ dữ liệu.
- *Khả năng xuất dữ liệu.* Ở mức tối thiểu, một gói phần mềm nên cho phép xuất dữ liệu tới các tệp văn bản dạng ASCII để có thể nhập nó vào các gói phần mềm khác. Các gói phần mềm thu thập dữ liệu cao cấp hơn sẽ cho phép dữ liệu được chuyển đổi tự động thành các cơ sở dữ liệu và chương trình phân tích thông thường.
- *Quản lý sự kiện và cảnh báo.* Phần mềm thu thập dữ liệu phải cung cấp khả năng quản lý sự kiện và cảnh báo. Đó là khả năng phát hiện xem một tín hiệu có vượt quá hoặc dưới mức giới hạn hay nằm ngoài hoặc nằm trong một phạm vi hay không. Nếu có cảnh báo, phần mềm nên cho phép tiến hành một loạt hoạt động như gửi báo cáo hoặc thực hiện một số dạng đầu ra số hoặc tương tự.
- *Các công cụ hiển thị và đồ thị.* Các gói phần mềm lưu trữ cần có một giao diện tốt để quan sát cả dữ liệu hiện tại và dữ liệu trong quá khứ. Giao diện này phải cho phép cuộn dữ liệu, phóng to tại các khu vực quan tâm, và quan sát các đồ thị dài hạn của dữ liệu.

Một nhược điểm của các ứng dụng dựa trên cấu hình là chúng sẽ bị bó hẹp trong các tính năng do nhà sản suất cung cấp trừ khi nhà sản xuất chế tạo theo yêu cầu đặt hàng. Nếu phép đo cần thay đổi và cần thêm một dạng tín hiệu khác, ta có thể không gặp may nếu gói phần mềm không hỗ trợ phép đo đó. Ngoài ra, nếu muốn tích hợp các chức năng phân tích ngoại tuyến, lập báo cáo và kết nối mạng vào trong ứng dụng thu thập dữ liệu, thì một ứng dụng phần mềm trọn gói đóng khó có thể thực hiện điều này. Mặt khác, có những ứng dụng phần mềm trọn gói cung cấp các phương pháp lựa chọn theo yêu cầu của khách hàng cùng với các công cụ phát triển ứng dụng phổ biến. Các gói phần mềm thu thập dữ liệu có thể tùy biến này cung cấp khả năng tốt nhất cho phép nhanh

chóng thích nghi với các ứng dụng thu thập và đem lại phương pháp tích hợp nhiều chức năng nâng cao sau này.



HÌNH 50.8 Ví dụ lập trình đồ họa của ứng dụng thu thập dữ liệu với National Instruments LabVIEW

Các công cụ phát triển ứng dụng là một lựa chọn khác cho việc phát triển các hệ thống thu thập dữ liệu dựa trên PC. Các công cụ phát triển là những ngôn ngữ lập trình dựa trên văn bản cho đến các môi trường lập trình đồ họa. Hình 50.8 là một ví dụ mã phần mềm cho một ứng dụng thu thập dữ liệu được phát triển trong môi trường lập trình đồ họa. Các công cụ phát triển cho phép xây dựng ứng dụng thu thập dữ liệu tùy biến của riêng người dùng nhằm thực hiện chính xác những gì mong muốn. Các công cụ phát triển ứng dụng cho phép sửa đổi ứng dụng theo yêu cầu, tích hợp các khả năng phân tích và báo cáo và tự động hoàn toàn hệ thống thu thập dữ liệu.

Khi phát triển các ứng dụng thu thập dữ liệu, nên lựa chọn một môi trường phát triển với các đặc trưng về khả năng làm việc cho phép tạo ra những hệ thống thu thập dữ liệu mạnh dựa trên PC. Một số đặc trưng cần xem xét khi đánh giá các công cụ phát triển ứng dụng là:

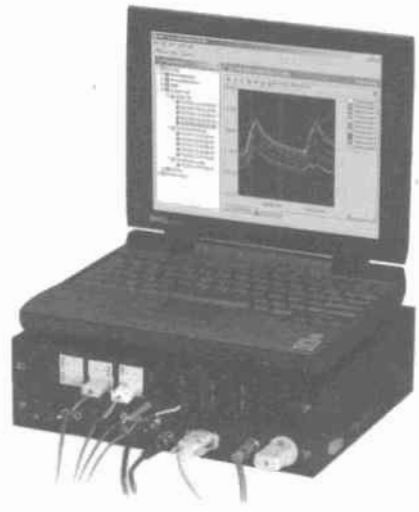
- *Phạm vi của các thành phần giao diện người dùng đồ họa.* Việc phát triển các thành phần giao diện người dùng như các đồ thị, hiển thị và điều khiển tốn khá nhiều thời gian. Nên chọn một môi trường phát triển có các thành phần giao diện người dùng chất lượng cao.
- *Tích hợp chặt chẽ với phần cứng đo lường.* Cần sử dụng phần mềm được thiết kế để làm việc với phần cứng đo lường. Tích hợp phần mềm hợp lý không chỉ giúp rút ngắn thời gian phát triển một cách đáng kể mà nó còn đảm bảo có được các thông số đo lường tin cậy.
- *Chức năng phân tích.* Một trong các lý do cơ bản của việc phát triển phần mềm ứng dụng thu thập dữ liệu là tích hợp các chức năng phân tích cao cấp. Một môi trường phát triển ứng dụng tốt sẽ cung cấp một loạt các chức năng phân tích để quản lý hầu hết những gì mong muốn.
- *Khả năng kết nối mạng.* Trong môi trường mạng ngày nay, khả năng kết nối ứng dụng thu thập dữ liệu tới Web có thể rất quan trọng. Phần mềm phát triển ứng dụng nên cung cấp công cụ tin cậy để đưa các kết quả lên mạng.
- *Báo cáo.* Môi trường phát triển ứng dụng nên cho phép lập báo cáo một cách tự động hoặc cho phép điều khiển theo chương trình các gói báo cáo lập từ bên ngoài.

Sự lựa chọn giữa phần mềm đóng gói và các công cụ phát triển phụ thuộc vào độ phức tạp của các ứng dụng thu thập dữ liệu và các yêu cầu của khách hàng. Với mỗi sự lựa chọn, quan trọng phải có một nhà cung cấp phần mềm chuyên kết nối hệ thống đo với máy tính và cung cấp dịch vụ cũng như hỗ trợ chất lượng cao.

Lựa chọn phần cứng

Có nhiều nền tảng phần cứng khác nhau cho hệ thống thu thập dữ liệu. Việc lựa chọn nền tảng nào phụ thuộc vào các yêu cầu về kích thước, môi trường hoạt động và lắp đặt. Mặc dù sự kết hợp

này rất mật thiết nhưng các nền tảng của thu thập dữ liệu dựa trên PC có thể chia thành bốn loại chính sau: di động, để bàn, lắp trên khung/công nghiệp và phân tán. Một trong những lợi ích chính của hệ thống thu thập dữ liệu dựa trên PC là nhiều phần mềm thu thập dữ liệu giống nhau được dùng với những nền tảng này.



HÌNH 50.9 Hệ thống thu thập dữ liệu dựa trên PC di động

Giải pháp thu thập dữ liệu di động cần thiết trong nhiều ứng dụng khác nhau, ví dụ như thiết bị thu thập dữ liệu trong ô tô hoặc thiết bị thực nghiệm. Giải pháp dựa trên PC di động sử dụng máy tính xách tay thay cho các máy tính và phần cứng đo lường được thiết kế phù hợp cho việc di động. Hình 50.9 giới thiệu một hệ thống thu thập dữ liệu dựa trên PC di động của National Instruments. Bộ số hóa là một card thu thập dữ liệu PCMCIA cắm liền nối với những hộp nhỏ cỡ máy tính xách tay để chuẩn hóa và kết nối tín hiệu. Hệ thống di động thường bị giới hạn tối đa đến 40 kênh tùy thuộc vào kích thước.

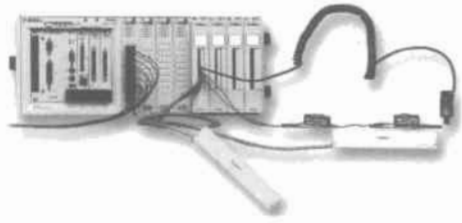
Hệ thống để bàn, như thấy ở Hình 50.3, sử dụng phần cứng đo lường được thiết kế để làm việc với các PC để bàn chuẩn. Hệ thống để bàn thích hợp cho các ứng dụng thu thập dự liệu trong phòng thí nghiệm, ví dụ như kiểm tra chất lượng của sản phẩm mới. Do hệ thống để bàn cố định không bị giới hạn bởi kích thước, các chức năng kết nối và chuẩn hóa tín hiệu thường được thực hiện bởi một hệ chuẩn hóa tín hiệu ngoại vi dạng môđun cung cấp khả năng đo rất nhiều loại cảm biến và tín hiệu cũng như dễ dàng mở rộng để thu thập hàng trăm kênh.

Nhiều khi các hệ thống để bàn chiếm quá nhiều diện tích hoặc nó không phù hợp với mỹ quan của môi trường như phòng thí nghiệm lớn hoặc các phương tiện sản xuất. Trong những trường hợp như vậy, giải pháp thu thập dữ liệu phù hợp và gọn nhẹ hơn là sử dụng PC công nghiệp có tính môđun, dựa trên chuẩn PXI hoặc Compact PCI. Hình 50.10 là một ví dụ về hệ thống thu thập dựa trên PXI. Một hệ thống môđun bao gồm PC, bo mạch DAQ, bộ chuẩn hóa tín hiệu và thiết bị kết nối. Những hệ thống như vậy được thiết kế để lắp trên khung do đó chúng có thể dễ dàng được cài đặt trong môi trường công nghiệp hoặc phòng thí nghiệm.

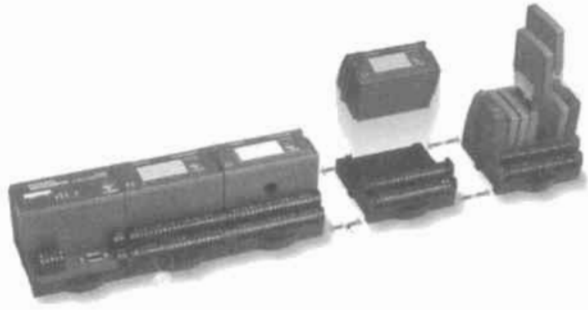
Cuối cùng, một số hệ thống thu thập dữ liệu cần phân tán khỏi PC. Đây là trường hợp khi cần thu thập dữ liệu từ nhiều vị trí xung quanh một thiết bị như khi thu thập các thông số hoạt động của xưởng hóa chất. Hệ thống thu thập phân tán nên gọn nhẹ để chúng có thể được lắp đặt ở nơi kín đáo, và chúng phải hoạt động trong phạm vi nhiệt độ rộng hơn. Với hệ thống thu thập phân tán, thường có nhiều điểm đo liên lạc với máy tính trung tâm thông qua một đường truyền như RS 485 hoặc Ethernet. Hình 50.11 là một ví dụ về hệ thống thu thập dữ liệu phân tán.

Sự lựa chọn nền tảng thu thập dữ liệu phụ thuộc vào yêu cầu của hệ thống thu thập dữ liệu và một số hệ thống có thể yêu cầu sử dụng nhiều nền tảng khác nhau. Khi phần cứng và phần mềm

được thiết kế hợp lý, các hệ thống thu thập dữ liệu có thể phát triển từ các hệ thống thí nghiệm đơn giản, số kênh thấp đến các hệ thống thu thập công nghiệp phân tán, số lượng kênh rất lớn.



HÌNH 50.10 Hệ thống thu thập dữ liệu dựa trên PC công nghiệp lắp trên khung



HÌNH 50.11 Hệ thống thu thập dữ liệu phân tán

50.5 Kết luận

Thu thập dữ liệu cho phép các nhà khoa học và các kỹ sư đánh giá các hiện tượng khác nhau từ thời tiết đến hoạt động của nhà máy. Hệ thống thu thập dữ liệu dựa trên PC mang lại sự linh hoạt, tùy biến và tích hợp. Để xác định một hệ thống thu thập dữ liệu, cần phải đánh giá được các yêu cầu về việc thu thập, phân tích trực tuyến, thu thập dữ liệu, phân tích ngoại tuyến, hiển thị, lập báo cáo và chia sẻ dữ liệu. Dựa trên các yêu cầu đó, có thể chọn phần cứng và phần mềm thu thập dữ liệu theo nhu cầu.

NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI
16 Hàng Chuối - Hai Bà Trưng - Hà Nội
Điện thoại: (04) 39714896; (04) 39724770. Fax: (04) 39714899

Chịu trách nhiệm xuất bản:

Giám đốc : PHÙNG QUỐC BẢO
Tổng biên tập : NGUYỄN BÁ THÀNH

Biên tập: VÂN NGỌC

Trình bày bìa: PHAN MẠNH QUÂN

Đối tác liên kết xuất bản:
VIỆN CÔNG NGHỆ VŨ TRỤ

SÁCH LIÊN KẾT

CƠ ĐIỆN TỬ - TẬP 2

Mã số: 1L - 13 ĐH2009
In 500 cuốn, khổ 18,2 × 25,7 cm tại Công ty TNHH Thiên Ấn
Số xuất bản: 62 - 2009/CXB/07 - 04/ĐHQGHN, ngày 20/1/2009
Quyết định xuất bản số: 13 LK-TN/XB
In xong và nộp lưu chiểu quý I năm 2009.

CƠ ĐIỆN TỬ

Cơ điện tử hiện nay đã mở ra một phương pháp mới trong ứng dụng kỹ thuật và đã thực sự bao trùm lên mọi diện mạo của thế giới hiện đại. Cơ điện tử chính là sự tích hợp hữu cơ của các hệ cơ khí, điện tử và máy tính. Để xây dựng thành công các hệ cơ điện tử đòi hỏi trí tuệ tổng hợp của các chuyên gia trong những lĩnh vực trên.

Với mục đích trở thành tài liệu tham khảo chuẩn cho các chuyên gia kỹ thuật trên khắp thế giới, Cuốn ***CƠ ĐIỆN TỬ*** đã đưa ra một cách nhìn toàn diện, thống nhất, chi tiết và đầy đủ về lĩnh vực năng động, ấn tượng này, và nêu ra xu hướng phát triển của các hệ cơ điện tử hiện đại. Hơn 60 bài viết của các nhà khoa học và công nghệ hàng đầu thế giới trình bày mọi khía cạnh của cơ điện tử, từ cái nhìn tổng quát về lịch sử cơ điện tử qua những lý thuyết cơ bản, các hệ thống, các quá trình và thực tiễn sản xuất đến viễn cảnh tương lai của cơ điện tử.

Cơ điện tử đang rộng mở, đầy những tiềm năng và cơ hội xán lạn. Cuốn ***CƠ ĐIỆN TỬ*** là một dấu ấn, đồng thời là cánh cửa và con đường dẫn bạn đến với thế giới cơ điện tử.

CÁC ĐIỂM NỔI BẬT CỦA CUỐN SÁCH:

- Giới thiệu những nghiên cứu toàn diện và mới nhất hiện nay của cơ điện tử.
- Tác giả là các nhà khoa học và công nghệ hàng đầu thế giới.
- Gồm đầy đủ công thức, dữ liệu, minh họa và tài liệu tham khảo.
- Các giải thích rõ ràng của mỗi lĩnh vực kỹ thuật đảm bảo để người đọc mặc dù không phải là chuyên gia cũng có thể hiểu và sử dụng các thông tin trong cuốn sách.
- Các tài liệu tham khảo từ nhiều nguồn khác nhau.

Chủ biên: **Robert H. Bishop** là giáo sư hạng Joe J. King ngành kỹ thuật của Khoa Công nghệ vũ trụ và Cơ kỹ thuật, trường Đại học Texas tại Austin (Mỹ). Ông là tác giả, nhà giáo, và nhà biên tập đã đạt được nhiều danh hiệu lớn. Các nghiên cứu hiện nay của ông tập trung chủ yếu vào lĩnh vực hàng không vũ trụ.

NHÀ XUẤT BẢN CRC

Giá: 180 000 VNĐ