TS. NGUYỄN NHƯ HIỀN & TS. LẠI KHẮC LÃI

HỆ MỜ & NƠRON

TRONG KỸ THUẬT ĐIỀU KHIỂN

Sách Chuyên khảo dùng cho đào tạo Sau đại học ngành Điều khiển & Tự động hoá

NHÀ XUẤT BẢN KHOA HỌC TỰ NHIÊN VÀ CÔNG NGHỆ

HÀ NỘI – 2007

MỤC LỤC

LỜI NÓI ĐẦU 6

Chương 1: LÔGIC MỜ 1
1.1. TỔNG QUAN VỀ LÔGIC MỜ 1
1.1.1. Quá trình phát triển của 1ôgic mờ 1
1.1.2. Cơ sở toán học của 1ôgic mờ 1
1.1.3. Lôgic mờ là 1ôgic của con người 2
1.2. KHÁI NIỆM VỀ TẬP MỜ 3
1.2.1. Tập kinh điển 3
1.2.3. Các thông số đặc trưng cho tập mờ 4
1.2.4. Các dạng hàm liên thuộc của tập mờ 5
1.3. CÁC PHÉP TOÁN TRÊN TẬP MỜ 5
1.3.1. Phép hợp hai tập mờ 5
1.3.2. Phép giao của hai tập mờ 6
1.3.3. Phép bù của một tập mờ 8
1.4. BIẾN NGÔN NGỮ VÀ GIÁ TRỊ CỦA BIẾN NGÔN NGỮ 8
1.5. LUẬT HỢP THÀNH MỜ 9
1.5.1. Mệnh đề hợp thành 9
1.5.2. Mô tả mệnh đề hợp thành 9
1.5.3. Luật hợp thành mờ 10
1.5.4. Các cấu trúc cơ bản của luật hợp thành 11
1.5.5. Luật hợp thành đơn có cấu trúc SISO 12
1.5.7. Luật của nhiều mệnh đề hợp thành 19
1.5.7. Luật hợp thành SUM-MIN và SUM-PROD 22
1.6. GIẢI MỜ 23
2.6.1. Phương pháp cực đại 24
Chương 2: ĐIỀU KHIỂN MỜ 29
2.1. CẤU TRÚC CỦA BỘ ĐIỀU KHIỂN MỜ 29
2.1.1. Sơ đồ khối bộ điều khiển mờ 29
2.1.2. Phân loại bộ điều khiển mở 30
2.1.3. Các bước tổng hợp bộ điều khiển mờ 31
2.2. BỘ ĐIỀU KHIỂN MỞ TĨNH 32

2.2.1. Khái niệm 32
2.2.2. Thuật toán tổng hợp một bộ điều khiển mờ tĩnh 32
2.2.3. Tổng hợp bộ điều khiển mờ tuyến tính từng đoạn 33
2.3. BỘ ĐIỀU KHIỂN MỜ ĐỘNG 35
2.4. THIẾT KẾ HỆ ĐIỀU KHIỂN MỜ BẰNG PIIẦN MỀM MATLAB . 37
2.4.1. Giới thiệu hộp công cụ lôgic mờ 37
2.3.2. Ví dụ thiết kế hệ mờ 41
2.5. HỆ ĐIỀU KHIỂN MỜ LAI (F-PID) 45
2.6. HỆ ĐIỀU KHIỂN THÍCH NGHI MỜ 46
2.6.1. Khái niệm 46
2.6.2. Tổng hợp bộ điều khiển thích nghi mờ ổn định 48
2.7. TỔNG HỢP BỘ ĐIỀU KHIỂN MỜ THÍCH NGHI TRÊN CƠ SỞ LÝ THUYẾT THÍCH NGHI KINH ĐIỂN 58
2.7.1. Đặt vấn đề 58
2.7.2. Mô hình toán học của bộ điều khiển mờ 60
2.7.3. Xây dựng cơ cấu thích nghi cho bộ điều khiển mờ 66
2.7.4. Một số ứng dụng điều khiển các đối tượng công nghiệp 70
Chương 3: TỔNG QUAN VỀ MẠNG NƠRON 75
3.1. NƠRON SINH HỌC 75
3.1.1. Chức năng, tổ chức và hoạt động của bộ não con người 75
3.1.2. Mạng nơron sinh học 76
3.2. MẠNG NƠRON NHÂN TẠO 77
3.2.1. Khái niệm 77
3.2.2. Mô hình nơron 80
3.3. CẤU TRÚC MẠNG 83
3.3.1. Mạng một lớp 83
3.3.2. Mạng nhiều lớp 84
3.4. CẤU TRÚC DỮ LIỆU VÀO MẠNG 87
3.4.1. Mô tả véctơ vào đối với mạng tĩnh 88
3.4.2. Mô tả véctơ vào liên tiếp trong mạng động 89
3.5. HUẤN LUYỆN MẠNG 92
3.5.1. Huấn luyện gia tăng 92
3.5.2 Huấn luyện mạng theo gói 94
Chương 4: MẠNG PERCEPTRONS 98
4.1. MỞ ĐẦU 98

4.1.1. Mô hình nơron perceptron 98
4.1.2. Kiến trúc mạng perceptron 100
4.2. THIẾT LẬP VÀ MÔ PHỎNG PERCEPTRON TRONG MATLAB 100
4.2.1 Thiết lập 100
4.2.2. Mô phỏng (sim) 102
4.2.3. Khởi tạo 103
4.3. CÁC LUẬT HỌC 104
4.3.1. Khái niệm 104
4.3.2. Luật học Perceptron (learnp) 105
4.3.3. Huấn luyện mạng (train) 107
4.4. CÁC HẠN CHẾ CỦA PERCEPTRON 111
4.5. SỬ DỤNG GIAO DIỆN ĐỒ HỌA ĐỂ KHẢO SÁT MẠNG NƠRON 112
4.5.1. Giới thiệu về GUI 112
4.5.2. Thiết lập mạng Perceptron (nntool) 113
4.5.3. Huấn luyện mạng 115
4.5.4. Xuất kết quả Perceptron ra vùng làm việc 116
4.5.5. Xoá cửa sổ dữ liệu mạng (Network/Data Window) 117
4.5.6 Nhập từ dòng lệnh 117
4.5.7. Cất biến vào file và nạp lại nó 118
Chương 5: MẠNG TUYẾN TÍNH 119
5.1. MỞ ĐẦU 119
5.1.1. Khái niệm 119
5.1.2. Mô hình nơron 119
5.2. CẤU TRÚC MẠNG 120
5.2.1. Cấu trúc 120
5.2.2. Khởi tạo nơron tuyến tính (Newlin) 121
5.3. THUẬT TOÁN CỰC TIỂU TRUNG BÌNH BÌNH PHƯƠNG SAI LỆCH 122
5.4. THIẾT KẾ HỆ TUYẾN TÍNH 123
5.5. MẠNG TUYẾN TÍNH CÓ TRỄ 123
5.5.1 Mắt trễ 123
5.5.2. Thuật toán LMS (learnwh) 123
5.5.3. Sự phân loại tuyến tính (train) 125
5.6. MỘT SỐ HẠN CHẾ CỦA MẠNG TUYẾN TÍNH 126

Chương 6: HỆ MỜ - NƠRON (FUZZY-NEURAL) 128
6.1 SỰ KẾT HỢP GIỮA LOGIC MỜ VÀ MẠNG NƠRON 128
6.1.1 Khái niệm 128
6.1.2. Kết hợp điều khiển mờ và mạng nơron 129
6.2. NƠRON MỜ 133
6.3. HUẤN LUYỆN MẠNG NƠRON-MỜ 135
6.4. SỬ DỤNG CÔNG CỤ ANFIS TRONG MATLAB ĐỂ THIẾT KẾ HỆ MỜ - NƠRON (ANFIS and the ANFIS Editor GUI) 139
6.4.1. Khái niệm 139
6.4.2. Mô hình học và suy diễn mờ thông qua ANFIS (Model Learning and Inferencc Through ANFIS) 140
6.4.3. Xác nhận dữ liệu huấn luyện (Familiarity Brecds Validation) ... 141
6.5. SỬ DỤNG BỘ SOẠN THẢO ANFIS GUI 143
6.5.1. Các chức năng của ANFIS GUI 143
6.5.2. Khuôn dạng dữ liệu và bộ soạn thảo ANFIS GUI: kiểm tra và huấn luyện (Data Formalities and the ANFIS Editor GUI: Checking and Training) 144
6.5.3. Một số ví dụ 145
6.6. SOẠN THẢO ANFIS TỪ DÒNG LỆNH 153
6.7. THÔNG TIN THÊM VỀ ANFIS VÀ BỘ SOẠN THẢO ANFIS EDITOR GUI 157
6.7.1. Dữ liệu huấn luyện (Training Data) 158
6.7.2. Cấu trúc đầu vào FIS (Input FIS Structure) 158
6.7.3. Các tùy chọn huấn luyện (Training Options) 159
6.7.4 Tuỳ chọn hiển thị Display Options 159
6.7.5. Phương pháp huấn luyện (Method) 160
6.7.6. Cấu trúc đầu ra FIS cho dữ liệu huấn 1uyện 160
6.7.7. Sai số huấn luyện 160
6.7.8. Bước tính (Step-size) 160
6.7.9. Dữ liệu kiểm tra (Checking Data) 161
6.7.10. Cấu trúc đầu ra FIS cho dữ liệu kiểm tra (Output FIS Structure for Checking Data) 162
6.7.11. Sai số kiểm tra (Checking Error) 162

TÀI LIỆU THAM KHẢO 163

LỜI NÓI ĐẦU

Ngày nay, các hệ thống mờ và mạng nơ ron ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống xã hội. Đặc biệt, trong lĩnh vực điều khiển và tự động hoá, hệ mờ và mạng nơ ron ngày càng chiếm ưu thế và đã mang lại nhiều lợi ích to lớn. Với ưu điểm cơ bản là có thể xứ lý với độ chính xác cao những thông tin "không chính xác" hệ mờ và mạng nơron là cơ sở của hệ "điều khiển thông minh" và "trí tuệ nhân tạo".

Để đáp ứng nhu cầu tìm hiểu và ứng dụng lôgic mờ và mạng nơ ron của đông đảo bạn đọc, được sự cổ vũ và động viên của BGH trường Đại học Kỹ thuật Công nghiệp, chúng tôi đã mạnh dạn viết cuốn sách "Hệ mờ và nơron trong kỹ thuật điều khiển".

Cuốn sách được viết dựa trên các bài giảng về hệ thống điều khiển thông minh cho học viên cao học ngành Tự động hoá trường Đại học Kỹ thuật Công nghiệp. Cuốn sách không phân tích quá sâu những vấn đề lý thuyết phức tạp mà chỉ cung cấp cho bạn đọc những nội dung rất cơ bản về Hệ mờ, mạng nơ ron nhân tạo và hệ Mờ-nơron. Mục tiêu cao hơn là giúp bạn đọc biết cách khai thác những công cụ sẵn có của phần mềm MATLAB để phân tích, thiết kế các bộ điều khiển mờ, nơron nhằm điều khiển các đối tượng trong công nghiệp. Mỗi phần đều có các ví dụ cụ thể để hướng dẫn thiết kế.

Cuốn sách là tài liệu tham khảo cho học viên cao học, sinh viên ngành Điều khiển, các kỹ sư ngành Điện, Công nghệ thông tin và các nghiên cứu sinh quan tâm đến lĩnh vực điều khiển mờ và mạng nơron.

Trong quá trình biên soạn, không tránh khỏi còn nhiều sai sót. Chúng tôi mong nhận được sự đóng góp ý kiến các của đồng nghiệp và bạn đọc gần, xa.

Xin chân thành cảm ơn!

Thái Nguyên, ngày 01 tháng 12 năm 2006

Các tác giả

Chương 1

LÔGIC MỜ

1.1. TỔNG QUAN VỀ LÔGIC MỜ

1.1.1. Quá trình phát triển của 1ôgic mờ

Từ năm 1965 đã ra đời một lý thuyết mới đó là lý thuyết tập mờ (Fuzzy set theory) đo giáo sư Lofti A. Zadeh ở trường đại học Califonia - Mỹ đưa ra. Từ khi lý thuyết đó ra đời nó được phát triển mạnh mẽ qua các công trình khoa học của các nhà khoa học như: Năm 1972 GS Terano và Asai thiết lập ra cơ sở nghiên cứu hệ thống điều khiển mờ ở Nhật, năm 1980 hãng Smith Co. bắt đầu nghiên cứu điều khiển mờ cho lò hơi... Những năm đầu thập kỷ 90 cho đến nay hệ thống điều khiển mờ và mạng nơron (Fuzzy system and neural network) được các nhà khoa học, các kỹ sư và sinh viên trong mọi lĩnh vực khoa học kỹ thuật đặc biệt quan tâm và ứng dụng trong sản xuất và đời sống. Tập mờ và lôgic mờ đã dựa trên các thông tin "không đầy đủ, về đối tượng để điều khiển đầy đủ về đối tượng một cách chính xác.

Các công ty của Nhật bắt đầu dùng lôgic mờ vào kỹ thuật điều khiển từ năm 1980. Nhưng do các phần cứng chuẩn tính toán theo giải thuật 1ôgic mờ rất kém nên hầu hết các ứng dụng đều dùng các phần cứng chuyên về lôgic mờ. Một trong những ứng dụng dùng lôgic mờ đầu tiên tại đây là nhà máy xử lý nước của Fuji Electric vào năm 1983, hệ thống xe điện ngầm của Hitachi vào năm 1987.

1.1.2. Cơ sở toán học của 1ôgic mờ

Lôgic mờ và xác xuất thông kê đều nói về sự không chắn chắn. Tuy nhiên mỗi lĩnh vực định nghĩa một khái niệm khác nhau về đối tượng.

Trong xác suất thống kê sự không chắc chắn liên quan đến sự xuất hiện của một sự kiện chắc chắn" nào đó.

Ví dụ: *Xác suất viên đạn trúng đích là* 0,

Bản thân của sự kiện "trúng đích" đã được định nghĩa rõ ràng, sự không

chắc chắn ở đây là có trúng đích hay không và được định lượng bởi mức độ xác suất (trong trường hợp này là 0,8). Loại phát biểu này có thể được xử lý và kết hợp với các phát biểu khác bằng phương pháp thống kê, như là xác suất có điều kiện chẳng hạn.

Sự không chắc chắn trong ngữ nghĩa, liên quan đến ngôn ngữ của con người, đó là sự không chính xác trong các từ ngữ mà con người dùng để ước lượng vấn đề và rút ra kết luận. Ví dụ như các từ mô tả nhiệt độ "nóng", "lạnh", "ấm"sẽ không có một giá trị chính xác nào để gán cho các từ này, các khái niệm này cũng khác nhau đối với những người khác nhau (là lạnh đối với người này nhưng không lạnh đối với người khác). Mặc dù các khái niệm không được định nghĩa chính xác nhưng con người vẫn có thể sử dụng chúng cho các ước lượng và quyết định phức tạp. Bằng sự trừu tượng và óc suy nghĩ, con người có thể giải quyết câu nói mang ngữ cảnh phức tạp mà rất khó có thể mô hình bởi toán học chính xác.

Sự không chắc chắn theo ngữ vựng: Như đã nói trên, mặc dù dùng những phát biểu không mang tính định lượng nhưng con người vẫn có thể thành công trong các ước lượng phức tạp. Trong nhiều trường hợp, con người dùng sự không chắc chắn này để tăng thêm độ linh hoạt. Như trong hầu hết xã hội, hệ thống luật pháp bao gồm một số luật, mỗi luật mô tả một tình huống. Ví dụ một luật quy định tội trộm xe phải bị tù 2 năm, một luật khác lại giảm nhẹ trách nhiệm. Và trong một phiên tòa, chánh án phải quyết định số ngày phạt tù của tên trộm dựa trên mức độ rượu trong người, trước đây có tiền án hay tiền sự không,... từ đó kết hợp lại đưa ra một quyết định công bằng.

1.1.3. Lôgic mờ là 1ôgic của con người

Trong thực tế, ta không định nghĩa một luật cho một trường hợp mà định nghĩa một số luật cho các trường hợp nhất định. Khi đó những luật này là những điểm rời rạc của một tập các trường hợp liên tục và con người xấp xỉ chúng. Gặp một tình huống cụ thể, con người sẽ kết hợp những luật mô tả các tình huống tương tự. Sự xấp xỉ này dựa trên sự linh hoạt của các từ ngữ cấu tạo nên luật, cũng như sự trừu tượng và sự suy nghĩ dựa trên sự linh hoạt trong lôgic của con người.

Để thực thi lôgic của con người trong kỹ thuật cần phải có một mô hình toán học của nó. Từ đó lôgic mờ ra đời như một mô hình toán học cho phép

mô tả các quá trình quyết định và ước lượng của con người theo dạng giải thuật. Dĩ nhiên cũng có giới hạn, đó là lôgic mờ không thể bắt chước trí tưởng tượng và khả năng sáng tạo của con người. Tuy nhiên, lôgic mờ cho phép ta rút ra kết luận khi gặp những tình huống không có mô tả trong luật nhưng có sư tương đương. Vì vậy, nếu ta mô tả những mong muốn của mình đối với hệ thống trong những trường hợp cụ thể vào luật thì lôgic mờ sẽ tạo ra giải pháp dựa trên tất cả những mong muốn đó.

1.2. KHÁI NIỆM VỀ TẬP MỜ

1.2.1. Tập kinh điển

Khái niệm tập hợp được hình thành trên nền tảng lôgic và được định nghĩa như là sự sắp xếp chung các đối tượng có cùng tính chất, được gọi là phần tử của tập hợp đó.

Cho một tập hợp A, một phần tử x thuộc A được ký hiệu: $x \in A$. Thông thường ta dùng hai cách để biểu diễn tập hợp kinh điển, đó là:

Liệt kê các phần tử của tập họp, ví dụ tập A_1 = {xe đạp, xe máy, xe ca, xe tải};

- Biểu diễn tập hợp thông qua tính chất tổng quát của các phần tử, ví dụ: tập các số thực (R), Tập các số tự nhiên (N).

Để biểu diễn một tập hợp A trên tập nền X, ta dùng hàm thuộc $\mu_A(x)$, với:

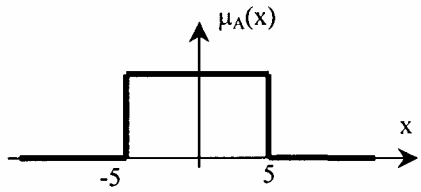
$$\mu A(x) = \begin{cases} 1 & \text{khi } x \in A \\ 0 & \text{khi } x \notin A \end{cases}$$

$\mu_A(x)$ chỉ nhận một trong 2 giá trị "1" hoặc "0"

ký hiệu = {$x \in X$| *x thoả mãn một số tính chất nào đó*}. Ta nói: Tập A được định nghĩa trên tập nền X.

Hình 1.1 mô tả hàm phụ thuộc $\mu_A(x)$ của tập các số thực từ -5 đến 5.

$A = \{x \in R | 5 \le x \le 5\}$



Hình 1.1. Hàm phụ thuộc $\mu_A(x)$của tập kinh điển A

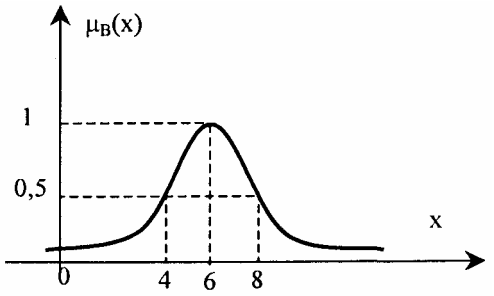
1.2.2. Định nghĩa tập mờ

Trong khái niệm tập hợp kinh điển hàm phụ thuộc $\mu_A(x)$ của tập A, chỉ có một trong hai giá trị là "1" nếu $x \in A$ hoặc "0" nếu $x \notin A$.

Cách biểu diễn hàm phụ thuộc như trên sẽ không phù hợp với những tập được mô tả "mờ" như tập B gồm các số thực gần bằng 5:

$$B = \{x \in R | \ x \approx 5\}.$$

Khi đó ta không thể khẳng định chắc chắn số 4 có thuộc B hay không? mà chỉ có thể nói nó thuộc B gao nhiêu phần trăm. Để trả lời được câu hỏi này, ta phải coi hàm phụ thuộc $\mu_B(x)$ có giá trị trong khoảng từ 0 đến 1 tức là: $0 \leq \mu_B(x) \leq 1$.



Hình 1.2. Hàm liên thuộc $\mu_B(x)$ của tập "mờ" B

Từ phân tích trên ta có định nghĩa: *Tập mờ B xác định trên tập kinh điển M là một tập mà một phần tử của nó được biểu diễn bởi một cặp giá trị* $(x,\mu_B(x))$. *Trong đó* $x \in M$ *và* $\mu_B(x)$ *là ánh xạ.*

Ánh xạ $\mu_B(x)$ được gọi là hàm liên thuộc của tập mờ B. Tập kinh điển M được gọi là cơ sở của tập mờ B.

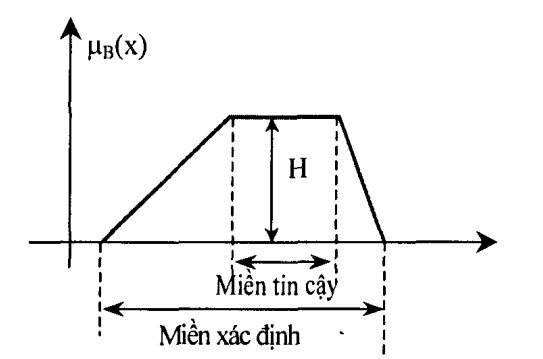
1.2.3. Các thông số đặc trưng cho tập mờ

Các thông số đặc trưng cho tập mờ là độ cao, miền xác định và miền tin cậy (hình 1.3)

+ **Độ cao** của một tập mờ B (Định nghĩa trên cơ sở M) là giá trị lớn nhất trong các giá trị của hàm liên thuộc:

$$H = \underset{x \in M}{SUP}\, \mu_B(x)$$

Một tập mờ có ít nhất một phần tử có độ phụ thuộc bằng 1 được gọi là tập mờ chính tắc (H = 1). Ngược lại, một tập mờ B với H < 1 gọi là tập mờ không chính tắc.



Hình 1.3. Độ cao, miền xác định, miền tin cậy của tập mờ

+ **Miền xác định** của tập mờ B (Định nghĩa trên cơ sở M) được ký hiệu bởi S là tập con của M có giá trị hàm liên thuộc khác không:

$$S = \{x \in M | \mu_B(x) > 0\}.$$

+ **Miền tin cậy** của tập mờ B (định nghĩa trên cơ sở M) được ký hiệu bởi T, là tập con của M có giá trị hàm liên thuộc bằng 1:

$$T = \{x \in M | \mu_B(X) = 1\}.$$

1.2.4. Các dạng hàm liên thuộc của tập mờ

Có rất nhiều cách khác nhau để biểu diễn hàm liên thuộc của tập mờ. Dưới đây là một số dạng hàm liên thuộc thông dụng:

+ Hàm liên thuộc hình tam giác (hình 1.4a);

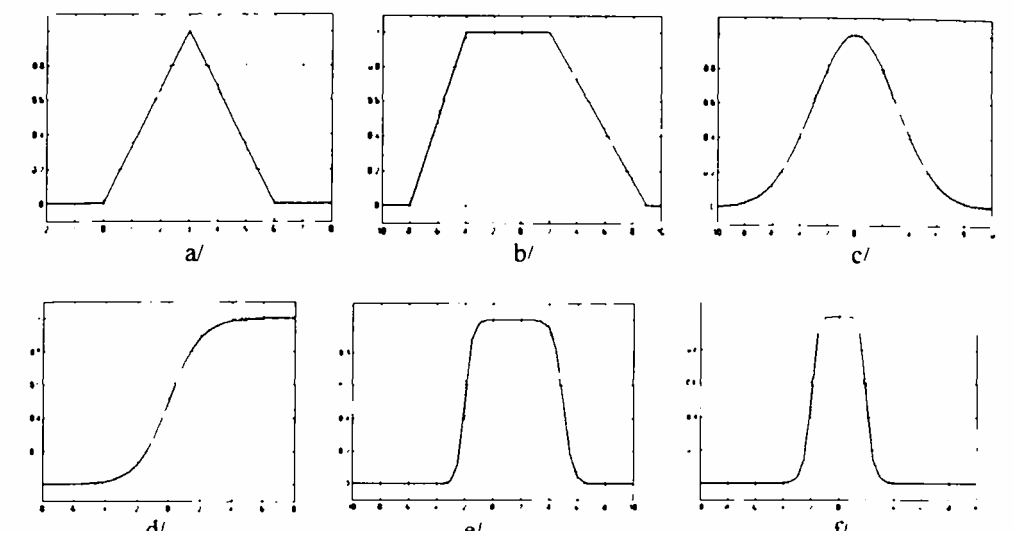
+ Hàm liên thuộc hình thang (hình 1.4b);

+ Hàm liên thuộc dạng Gauss (hình l.4c);

+ Hàm liên thuộc dạng Sign (hình 1.4d);

+ Hàm Sigmoidal (hình 1.4e);

+ Hàm hình chuông (hình 1.4f).



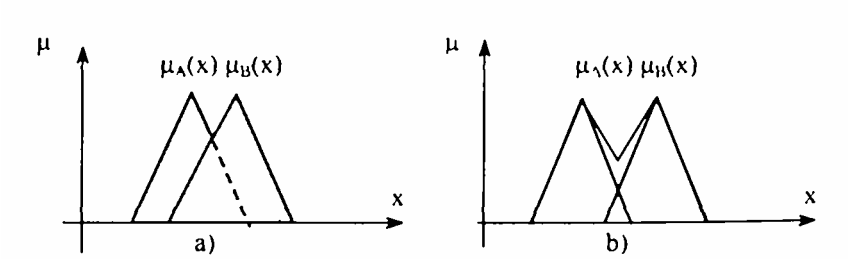
Hình 1.4. Các dạng hàm liên thuộc của tập mờ

1.3. CÁC PHÉP TOÁN TRÊN TẬP MỜ

Trên tập mờ có 3 phép toán cơ bản là phép hợp, phép giao, và phép bù.

1.3.1. Phép hợp hai tập mờ

a/ Hợp của hai lập mờ có cùng cơ sở



Hình 1.5. Hợp của hai tập mờ có cùng cơ sở

theo quy tắc Max (a), theo Lukasiewwiez (b)

Hợp của hai tập mờ A và B có cùng cơ sở M là một tập mờ cùng xác định trên cơ sở M với hàm liên thuộc được xác định theo một trong các công thức sau:

1. $\mu_{A \cup B}(x) = \text{Max}\{\mu_A(x), \mu_B(x)\}$;

2. $\mu_{A \cup B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\}$ phép hợp Lukasiewiez);

3. $\mu_{A \cup B}(x) = \begin{cases} \max\{\mu_A(x), \mu_B(x)\} \text{ khi } \min\{\mu_A(x), \mu_B(x)\}=0 \\ 1 \text{ khi } \min\{\mu_A(x), \mu_B(x)\} \neq 0 \end{cases}$

4. $\mu A \cup B(x) = \dfrac{\mu A(x) + \mu B(x)}{1 + \mu A(x) + \mu B(x)}$ (Tổng Einstein)

5. $\mu_{A \cup B}(x) = \mu_A(x) = \mu_B(x) - \mu_A(x)\mu_A(x)$ (tổng trực tiếp).

Chú ý: Có nhiều công thức khác nhau được dùng để tính hàm liên thuộc $\mu_{A \cup B}(x)$ của hai tập mờ. Song trong kỹ thuật điều khiển mờ ta chủ yếu dùng 2 công thức hợp, đó là lấy Max và phép hợp Lukasiewiez.

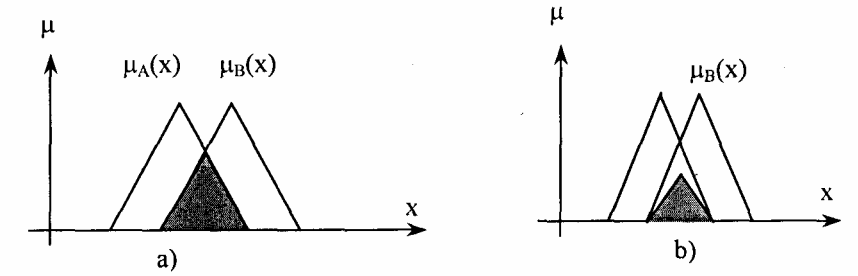
b/ Hợp hai tập mờ khác cơ sở

Để thực hiện phép hợp 2 tập mờ khác cơ sở, về nguyên tắc ta phải đưa chúng về cùng một cơ sở. Xét tập mờ A với hàm liên thuộc $\mu_A(x)$ được định nghĩa trên cơ sở M và B với hàm liên thuộc $\mu_B(x)$ được định nghĩa trên cơ sở N, hợp của 2 tập mờ A và B là một tập mờ xác định trên cơ sở MxN với hàm liên thuộc: $\mu_{A \cup B}(x, y) = \text{Max}\{\mu_A(x, y), \mu_B(x, y)\}$

Với $\mu_A(x, y) = \mu_A(x)$ với mọi $y \in N$ và $\mu_B(x, y) = \mu_B(y)$ với mọi $x \in M$.

1.3.2. Phép giao của hai tập mờ

a/ Giao hai tập mờ cùng cơ sở



Hình 1.6. Giao của hai tập mờ có cùng cơ sở
theo quy tắc Min (a) và theo tích đại số (b)

Giao của hai tập mờ A và B có cùng cơ sở M là một tập mờ cũng xác định trên cơ sở M với hàm liên thuộc $\mu_{A \cap B}(x)$ được tính:

1, $\mu_{A \cap B}(x) = \text{Min } \{\mu_A(x), \mu_B(x)\}$;

2. $\mu_{A \cap B}(x) = \mu_A(x).\mu_B(x)$ (tích đại số);

3. $\mu_{A \cap B}(x) = \begin{cases} \min\{\mu_A(x), \mu_B(x)\} \text{ khi } \min\{\mu_A(x), \mu_B(x)\} = 1 \\ 0 \text{ khi } \max\{\mu_A(x), \mu_B(x)\} \neq 1 \end{cases}$;

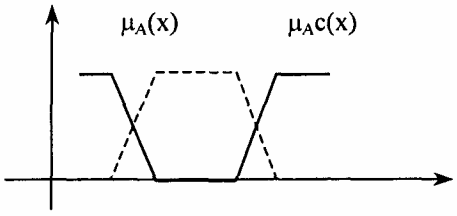
4. $\mu_{A \cap B}(x) = \max\{0, \mu_A(x)+\mu_B(x)-1\}$ (Phép giao Lukasiewiez);

5. $\mu_{A \cap B}(x) = \dfrac{\mu_A(x).\mu_B(x)}{2-(\mu_A(x)+\mu_B(x))-\mu_A(x)\mu_B(x)}$ (tích Einstein).

cũng giống như trong phép hợp, trong kỹ thuật điều khiển chủ yếu ta sử dụng công thức 1 và công thức 2 để thực hiện phép giao 2 tập mờ.

b/ Giao hai tập mờ khác cơ sở

Để thực hiện phép giao 2 tập mờ khác cơ sở, ta cần phải đưa về cùng cơ sở. Khi đó, giao của tập mờ A có hàm liên thuộc $\mu_A(x)$ định nghĩa trên cơ sở M với tập mờ B có hàm liên thuộc $\mu_B(x)$ định nghĩa trên cơ sở N là một tập mờ xác định trên cơ sở M *x* N có hàm liên thuộc được tính:



Hình 1.7. Bù của tập mờ

$\mu_{A \cap B}(x, y) = \text{MIN}\{\mu_A(x, y), \mu_B(x, y)\}$

Trong đó: $\mu_A(x, y) = \mu_A(x)$ với mọi $y \in N$ và $\mu_B(x, y) = \mu_B(x)$ với mọi $x \in M$.

1.3.3. Phép bù của một tập mờ

Bù của tập mờ A có cơ sở M và hàm liên thuộc $\mu_A(x)$ là một tập mờ A_C xác định trên cùng cơ sở M với hàm liên thuộc: $\mu_A(x) = 1 - \mu_A(x)$

1.4. BIẾN NGÔN NGỮ VÀ GIÁ TRỊ CỦA BIẾN NGÔN NGỮ

Thực tế hàng ngày chúng ta luôn dùng các từ ngữ, lời nói để mô tả các biến. Ví dụ khi ta nói: "Điện áp cao quá", "xe chạy nhanh quá",... như vậy biến "Điện áp", biến "Tốc độ xe",... nhận các giá trị từ "nhanh" đến "chậm", từ "cao" đến "thấp". Ở dạng tường minh, các biến này nhận các giá trị cụ thể (rõ) như điện áp bằng 200 V, 250 V...; tốc độ xe bằng 60 km/h, 90 km/h... Khi các biến nhận các giá trị không rõ ràng như "cao", "rất cao" "nhanh", "hơi nhanh"... ta không thể dùng các giá trị rõ để mô tả được mà phải sử dụng một số khái niệm mới để mô tả gọi là biến ngôn ngữ.

Mộ biến có thể gán bởi các từ trong ngôn ngữ tự nhiên làm giá trị của nó gọi là biến ngôn ngữ.

Một biến ngôn ngữ thường bao gồm 4 thông số: X, T, U, M. Với:

+ X: Tên của biến ngôn ngữ;

+ T: Tập của các giá trị ngôn ngữ;

+ U: Không gian nền mà trên đó biến ngôn ngữ X nhận các giá trị rõ;

+ M: Chỉ ra sự phân bố của T trên U.

Ví dụ: Biến ngôn ngữ "Tốc độ xe" có tập các giá trị ngôn ngữ là rất chậm, chậm, trung bình, nhanh, rất nhanh, không gian nền của biến là tập các số thực dương. Vậy biến tốc độ xe có 2 miền giá trị khác nhau:

- Miền các giá trị ngôn ngữ N = [rất chậm, chậm, trung bình, nhanh, rất nhanh].

- Miền các giá trị vật lý $V = \{x \in R\ (x \geq 0)\}$.

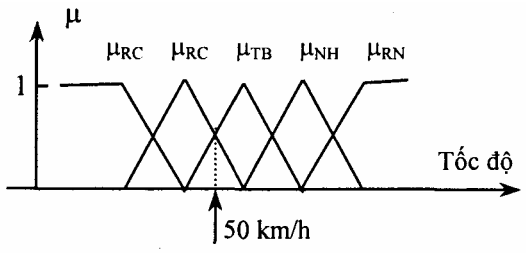
Mỗi giá trị ngôn ngữ (mỗi phần tử của Ni có tập nền là miền giá trị vật lý V. Từ một giá trị vật lý của biến ngôn ngữ ta có được một véctơ $\underline{\mu}$ gồm các độ phụ thuộc của x:

$X \rightarrow \underline{\mu}^T = [\mu_{\text{rất chậm}}\ \mu_{\text{chậm}}\ \mu_{\text{trung bình}}\ \mu_{\text{nhanh}}\ \mu_{\text{rất nhanh}}]$

ánh xạ trên được gọi là quá trình fuzzy hoá giá trị rõ x.

Ví dụ: ứng với tốc độ 50 km/h ta có

véc tơ: $\mu(50) = \begin{cases} 0 \\ 0,5 \\ 0,5 \\ 0 \\ 0 \end{cases}$



Hình 2.8. Mờ hoá biến "Tốc độ"

1.5. LUẬT HỢP THÀNH MỜ

1.5.1. Mệnh đề hợp thành

Xét hai biến ngôn ngữ χ và γ; Biến χ nhận giá trị (mờ) A có hàm liên thuộc $\mu_A(x)$ và γ nhận giá trị (mờ) B có hàm liên thuộc $\mu_B(x)$ thì hai biểu thức:

χ = A; γ = B được gọi là hai mệnh đề.

Luật Điều khiển: nếu **χ = A thì γ = B** được gọi là mệnh đề hợp thành.Trong đó χ = A gọi là mệnh đề điều kiện và γ = B gọi là mệnh đề kết luận. Một mệnh đề hợp

thành có thể có nhiều mệnh đề điều kiện và nhiều mệnh đề kết luận, các mệnh đề liên kết với nhau bằng toán tử "và". Dựa vào số mệnh đề điều kiện và số mệnh đề kết luận trong một mệnh đề hợp thành mà ta phân chúng thành các cấu trúc khác nhau:

- Cấu trúc SISO (một vào, một ra): Chỉ có một mệnh đề điều kiện và một mệnh đề kết luận. Ví dụ: nếu χ = A thì γ = B.

- Cấu trúc MISO (Nhiều vào, một ra): Có từ 2 mệnh đề điều kiện trở lên và một mệnh đề kết luận. Ví dụ: nếu $\chi_1 = A_1$ và $\chi_2 = A_2$ thì γ = B.

- Cấu trúc MIMO (Nhiều vào, nhiều ra): Có ít nhất 2 mệnh đề điều kiện và 2 mệnh đề kết luận. Ví dụ: nếu $\chi_1 = A_1$ và $\chi_2 = A_2$ thì $\gamma_1 = B_1$ và $\gamma_2 = B_2$

1.5.2. Mô tả mệnh đề hợp thành

Xét mệnh đề hợp thành: nếu **χ = A thì γ - B**; Từ một giá trị x_0 có độ phụ thuộc $\mu_A(x_0)$ đối với tập mờ A của mệnh đề điều kiện, ta xác định được độ thoả mãn mệnh đề kết luận. Biểu diễn độ thoả mãn của mệnh đề kết luận như một tập mờ B' cùng cơ sở với B thì mệnh đề hợp thành chính là ánh xạ: $\mu_A(x_0) \rightarrow \mu_B(y)$.

Ánh xạ này chỉ ra rằng mệnh đề hợp thành là một tập mà mỗi phần tử là một giá trị $(\mu_A(x_0), \mu_{B'}(y))$ tức là mỗi phần tử là một tập mờ. Mô tả mệnh đề hợp thành tức là mô tả ánh xạ trên. Ánh xạ $(\mu_A(x_0), \mu_{B'}(y))$ được gọi là hàm liên thuộc của luật hợp thành. Để xây dựng $\mu_{B'}(y)$ đã có rất nhiều ý kiến khác nhau. Trong kỹ thuật điều khiển ta thường sử dụng nguyên tắc của Mamdani ***"Độ phụ thuộc của kết luận không được lớn hơn độ phụ thuộc của điều kiện"?*** Từ nguyên tắc đó ta có hai công thức xác định hàm liên thuộc cho mệnh đề hợp thành A => B:

1. công thức MIN: $\mu_{A=>B}(x, y) = MIN\{\mu_A(x), \mu_B(y)\}$

2. công thức PROD: $\mu_{A=>B}(x, y) = \mu_A(x)\mu_B(xy$

1.5.3. Luật hợp thành mờ

Luật hợp thành là tên chung gọi mô hình R biểu diễn (một hay nhiều) hàm liên thuộc $\mu_{A=>B}(x, y)$ cho (một hay nhiều) mệnh đề hợp thành $A \Rightarrow B$.

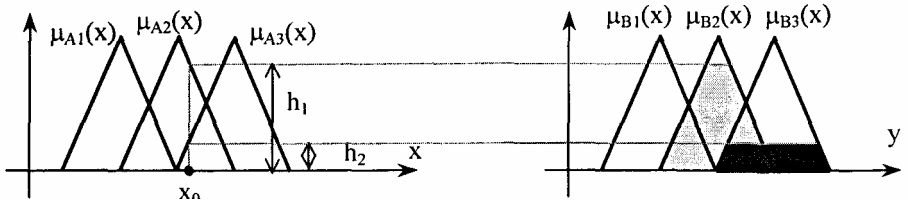
Một luật hợp thành chỉ có 1 mệnh đề hợp thành gọi là luật hợp thành đơn, có từ 2 mệnh đề hợp thành trở lên gọi là luật hợp thành phức.

Xét luật hợp thành R gồm 3 mệnh đề hợp thành:

R_1: Nếu $x = A_1$ Thì $y = B_1$ hoặc

R_2: Nếu $x = A_2$ Thì $y = B_2$ hoặc

R_3: Nếu $x = A_3$ Thì $y = B_3$ hoặc



Hình 1.9. Mô tả hàm liên thuộc của luật hợp thành

Với mỗi giá trị rõ x_0 của biến ngôn ngữ đầu vào, ta có 3 tập mờ ứng với 3 mệnh đề hợp thành R_1, R_2, R_3 của luật hợp thành R. Gọi hàm liên thuộc của các tập mờ đầu ra là: $\mu_{B_1'}(y)$; $\mu_{B_2'}(y)$; $\mu_{B_3'}(y)$ thì giá trị của luật hợp thành R ứng với x_0 là tập mờ B' thu được qua phép hợp 3 tập mờ: B' = B'$_1 \cup$ B'$_2 \cup$ B'$_3$.

Tuỳ theo cách thu nhận các hàm liên thuộc $\mu_{B_1'}(y)$; $\mu_{B_2'}(y)$; $\mu_{B_3'}(y)$ và phương pháp thực hiện phép hợp để nhận tập mờ B' mà ta có tên gọi các luật hợp thành khác nhau:

- Luật hợp thành MAX-MIN nếu $\mu_{B_1'}(y)$; $\mu_{B_2'}(y)$; $\mu_{B_3'}(y)$ thu được qua phép lấy Min còn phép hợp thực hiện theo luật Max;

- Luật hợp thành MAX-PROD nếu $\mu_{B_1'}(y)$; $\mu_{B_2'}(y)$; $\mu_{B_3'}(y)$ thu được qua phép PROD còn phép hợp thực hiện theo luật Max;

- Luật hợp thành SUM-MIN nếu $\mu_{B_1'}(y)$; $\mu_{B_2'}(y)$; $\mu_{B_3'}(y)$ thu được qua phép lấy Min còn phép hợp thực hiện theo luật SUM;

- Luật hợp thành SUM - PROD nếu $\mu_{B_1'}(y)$; $\mu_{B_2'}(y)$; $\mu_{B_3'}(y)$ thu được qua phép lấy PROD còn phép hợp thực hiện theo Lukasiewicz.

Vậy, để xác định hàm liên thuộc $\mu_{B'}(y)$ của giá trị đầu ra B' của luật hợp thành có n mệnh đề hợp thành R_1, R_2,… ta thực hiện theo các bước sau:

+ Xác định độ thoả mãn h_j.

+ Tính $\mu_{B_1'}(y)$; $\mu_{B_2'}(y)$; $\mu_{B_3'}(y)$ theo quy tắc min hoặc Prod

$$\mu_{B_j'}(y) = \text{Min}\{u_A(x_0), \mu_{B_i}(y)\} = \text{Min}\{h_j, \mu_{B_j}(y)\}$$

hoặc $\mu_{B_j}(y) = \mu_A(x_0).\mu_{B_j}(y) = h_j.\mu_{B_j}(y)$.

+ xác định $\mu_{B'}(y)$ bằng cách thực hiện phép hợp các $\mu_{B_j'}(y)$

1.5.4. Các cấu trúc cơ bản của luật hợp thành

Ta sẽ khảo sát hai cấu trúc cơ bản của luật hợp thành, đó là cấu trúc SISO và cấu trúc MISO.

+ **Cấu trúc SISO** là cấu trúc trong đó luật hợp thành có các mệnh đề điều kiện và mệnh đề kết luận là các mệnh đề đơn.

Ví dụ: R_1: nếu $\chi = A_1$ thì $\gamma = B_1$ hoặc

R_2: nếu $\chi = A_2$ thì $\gamma = B_2$.

+ **Cấu trúc MISO** là cấu trúc trong đó luật hợp thành có các mệnh đề điều kiện là mệnh đề phức và mệnh đề kết luận là mệnh đề đơn.

Ví dụ: R_1: nếu $\chi_1 = A_1$ và $\chi_2 = B_1$ thì $\gamma = C_1$ hoặc

R_2: nếu $\chi_1 = A_2$ và $\chi_2 = B_2$ thì $\gamma = C_2$.

1.5.5. Luật hợp thành đơn có cấu trúc SISO

a) Luật hợp thành MIN

Luật hợp thành MIN là tên gọi mô hình (ma trận) R của mệnh đề hợp thành $A \Rightarrow B$ khi hàm liên thuộc $\mu_{A=>B}(x, y)$ của nó được xây dựng theo quy tắc MIN.

Xét luật hợp thành chỉ có 1 mệnh đề: **Nếu $\chi = A$ thì $\gamma = B$**

Để xây dựng R, trước tiên hai hàm liên thuộc $\mu_A(x)$ và $\mu_B(y)$ được rời rạc hoá với tần số rời rạc đủ nhỏ để không bị mất thông tin.

Ví dụ: $\mu_A(x)$, $\mu_B(y)$ được rời rạc hoá tại các điểm:

$x \in \{10, 20, 30, 40, 50\}$

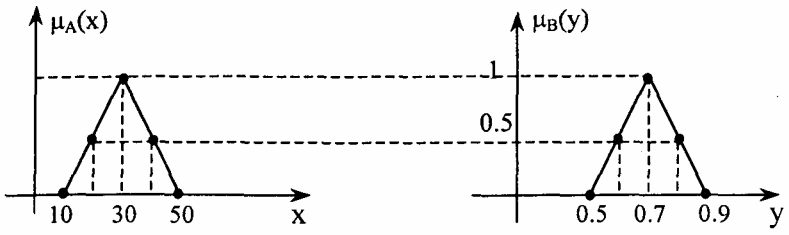
$y \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$.

Với các điểm rời rạc này thì theo

$\mu_{A=>B}(20; 0.7) = \mu_R(20; 0.7) = MIN\{\mu_A(20), \mu_b(0.7)\} = MIN\{0.5; 1\} = 0.5$

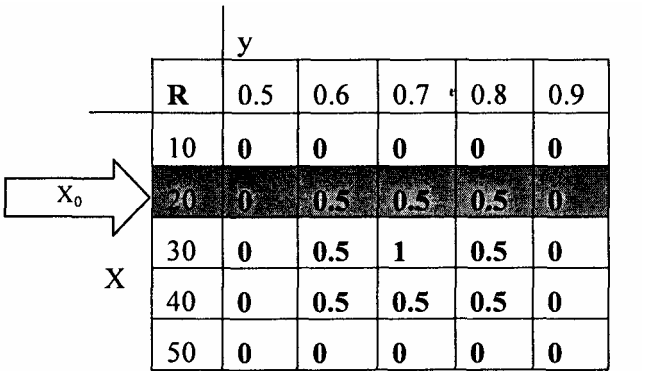
$\mu_{A=>B}(30; 0.7) = \mu_R(30; 0.7) = MIN\{\mu_A(30), \mu_b(0.7)\} = MIN\{1; 1\} = 1$

.....



Hình 1.10. Rời rạc hoá các hàm liên thuộc

Nhóm tất cả các giá trị $\mu_{A=>B}(x, y) = \mu_R(x,y)$ gồm 5 *x* 5= 25 giá trị, thành ma trận R (được gọi là ma trận hợp thành MIN) gồm 5 hàng 5 cột.



R	0.5	0.6	0.7	0.8	0.9
10	0	0	0	0	0
20	0	0.5	0.5	0.5	0
30	0	0.5	1	0.5	0
40	0	0.5	0.5	0.5	0
50	0	0	0	0	0

Khi tín hiệu đầu vào là một giá trị rõ $x_0 = 20$, tín hiệu đầu ra B' có hàm liên thuộc:

$\mu_{B'}(y) = \mu_R(20, y) = \{0; 0.5; 0.5; 0.5; 0\}$.

Để thuận tiện cho việc xác định hàm liên thuộc của tín hiệu ra dưới dạng nhân ma trận, ta định nghĩa một ma trận $T = \{a_1\ a_2 \ldots\}$ ma trận này chỉ có một phần tử bằng 1 còn các phần tử khác đều bằng 0. Ví dụ với tập 5 phần tử cho tín hiệu đầu vào xử {10; 20; 30; 40; 50} thì ứng với $x_0 = 20$ (phần tử thứ hai) ta có:

$$\underline{a} = (0\ 1\ 0\ 0\ 0)$$

Và khi đó

$$\mu_{B'}(y) = \mu_R(x_0, y) = \underline{a}^T.\ R = \{0\ 0.5\ 0.5\ 0.5\ 0\}.$$

Tổng quát cho một giá trị rõ x_0 bất kỳ

$$x_0 \in X = \{10\ 20\ 30\ 40\ 50\}$$

tại đầu vào véctơ chuyển vị có dạng:

$$\underline{a}^T = (a_1, a_2, a_3, a_4, a_5)$$

trong đó chỉ có một phần tử a; duy nhất có chỉ số i là chỉ số của x_0 trong X có giá trị bằng 1, các phần tử còn lại đều bằng 0. Hàm liên thuộc $m_{B'}(y)$ dưới dạng rời rạc được xác định:

$$\mu_{B'}(y) = \underline{a}^T.R = (a_1, a_2, a_3, a_4, a_5)\begin{pmatrix} r_{11}.....r_{15} \\ ..... \\ r_{51}.....r_{55} \end{pmatrix} =$$

$$= (l_1, l_2, l_3, l_4, l_5) \text{ với } l_k = \sum_{i=1}^{5} a_i r_{ik} . \qquad (1.1)$$

Chú ý: Trong biểu thức (1.1) để tính $\mu_{B'}(y)$ ta cần cài đặt thuật toán nhân ma trận của đại số tuyến tính, do đó tốc độ xử lý chậm. Để khắc phục nhược điểm này, phép nhân ma trận (1.1) được thay bởi luật MAX-MIN của Zadeh với MAX (phép lấy cực đại) thay vào vị trí phép cộng và MIN (phép lấy cực tiểu) thay vào vị trí phép nhân. Khi đó:

$$l_K = \max_{1 \le i \le 5} \min \{a_i\, r_{ki}\}$$

Kết quả hai phép tính (1.1) và (1.2) với đầu vào là một giá trị rõ hoàn toàn giống nhau. Cũng từ lý do trên mà luật hợp thành MIN còn có tên gọi là luật hợp thành MAX-MIN.

b/ Luật hợp thành PROD

Tương tự như đã làm với luật hợp thành MIN, ma trận R của luật hợp thành PROD được xây dựng gồm các hàng là m giá trị rời rạc của đầu ra $\mu_{B'}(y_1)$, $\mu_{B'}(y_2)$, $\mu_{B'}(y_m)$ cho n giá trị rõ đầu vào x_n, x_n,…., x_n Như Vậy ma trận R sẽ có n hàng và m cột. Xét ví dụ trên cho 5 giá trị đầu vào:

$$\{x_1, x_2, x_3, x_4, x_5\} = \{10\ 20\ 30\ 40\ 50\}$$

thì với từng giá trị x_i, 5 giá trị của hàm liên thuộc đầu ra tương ứng $\mu_{B'}(0.5)$, $\mu_{B'}(0.6)$, $\mu_{B'}(0.7)$, $\mu_{B'}(0.8)$, $\mu_{B'}(0.9)$ được liệt kê trong ma trận R được gọi là ma trận hợp thành PROD.

Từ ma trận R trên, hàm liên thuộc $\mu_{B'}(y)$ của giá trị đầu ra khi đầu vào là giá trị rõ x_4 cũng được xác định bằng công thức:

$$\underline{a}^T = (0, 0, 0, 1, 0)$$

$$\mu_{B'}(y) = \mu_R(x_4, y) = \underline{a}^T .R = \{0, 0.25, 0.5, 0.25, 0\}.$$

Đê rút ngắn thời gian tính và cũng để mở rộng công thức trên cho trường hợp đầu vào là giá trị mờ, phép nhân ma trận T.R cũng được thay bằng luật MAX- PROD của Zadeh như đã làm cho luật hợp thành MIN. Trong đó phép nhân được thực hiện bình thường còn phép lấy cực đại thay vào vị trí của phép cộng.

$$\mu_{B'}(y) = \underline{a}^T.R = (a_1, a_2, a_3, a_4, a_5)\begin{pmatrix} r_{11}.....r_{15} \\ ..... \\ r_{51}.....r_{55} \end{pmatrix} =$$

$$= (l_1, l_2, l_3, l_4, l_5)$$

với $l_k = \sum_{i=1}^{5} a_i r_{ik}$; $l_k = \max_{1\le i\le 5} \mathrm{Prod}\{a_i r_{ki}\}$.

	R	0.5	0.6	0.7	0.8	0.9
i = 1	10	**0**	**0**	**0**	**0**	**0**
i = 2	20	**0**	**0.25**	**0.5**	**0.25**	**0**
i = 3	30	**0**	**0.5**	**1**	**0.5**	**0**
i = 4	40	**0**	**0.25**	**0.5**	**0.25**	**0**
i = 5	50	**0**	**0**	**0**	**0**	**0**

c) Thuật toán xây dựng R

Từ các phân tích trên, ta rút ra thuật toán xây dựng R cho luật hợp thành đơn có cấu trúc SISO **(Nếu χ = A Thì γ = B)** như sau:

1- Rời rạc hoá $\mu_A(x)$ tại n điểm $x_1, x_2,\ldots,x_n$ tại m điểm $y_1, y_2,\ldots,y_n$ (n có thể khác m)

2- Xây dựng ma trận R gồm n hàng và m cột:

$$R = \begin{pmatrix} \mu_R(x_1, y_1).....\mu_R(x_1, y_m) \\ ..... \\ \mu_R(x_n, y_1).....\mu_R(x_n, y_m) \end{pmatrix} = \begin{pmatrix} r_{11}.....r_{1m} \\ ..... \\ r_{n1}.....r_{nm} \end{pmatrix}$$

3- Xác định hàm liên thuộc $\mu_{B'}(y)$ của đầu ra ứng với giá trị rõ dầu vào x_k theo biểu thức:

$$\mu_{B'}(y) = \underline{a}^T.R = (a_1, a_2, \ldots a_n) \begin{pmatrix} r_{11} \ldots\ldots\ldots\ldots r_{1m} \\ \ldots\ldots\ldots\ldots\ldots \\ r_{n1} \ldots\ldots\ldots\ldots r_{nm} \end{pmatrix} = (l_1, l_2, \ldots\ l_m)$$

với $l_k = \sum_{i=1}^{n} a_i r_{ik}$.

Với: $\underline{a}^T = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$

↑ vị trí thứ k

trong đó: $l_K = \max_{1 \le i \le n} \min \{a_i\ r_{ki}\}$, k = 1,2,.., m nếu sử dụng công thức MAX-MIN và $l_K = \max_{1 \le i \le n} \text{prod} \{a_i\ r_{ki}\}$, k = 1,2,.., m nếu sử dụng công thức MAX-PROD.

4- Xác định $\mu_{B'}(y)$ theo công thức: $\mu_{B'}(y) = (l_1, l_2, \ldots, l_m)$.

Chú ý:

☞Trong trường hợp đầu vào là giá trị mờ A' với hàm liên thuộc $\mu_{A'}(y)$ thì hàm liên thuộc $\mu_{B'}(y)$ của giá trị đầu ra B': $\mu_{B'}(y) = (l_1, l_2, \ldots, l_m)$ cũng được tính theo công thức (2.4) và

$$l_k = \max_{1 \le i \le n} \min \{a_i\ r_{ki}\},\ k = 1, 2, \ldots, m$$

trong đó $\underline{a}$ là véctơ gồm các giá trị rời rạc của hàm liên thuộc $\mu_{A'}(x)$ của A' tại các điểm:

$x \in X = \{x_1, x_2, \ldots, x_n\}$ tức là $\underline{a}^T = (\mu_{A'}(x_1), \mu_{A'}(x_2), \ldots, \mu_{A'}(x_n))$.

☞ Giả thiết có n điểm rời rạc $x_1, x_2, \ldots, x_n$ của cơ sở A và m điểm rời rạc $y_1, y_2, \ldots, y_m$ của cơ sở B ta có hai véctơ:

$\mu_A{}^T = \{\mu_A(x_1), \mu_A(x_2), \ldots, \mu_A(x_n)\}$ và $\mu_A{}^T = \{\mu_B(y_1), \mu_B(y_2), \ldots, \mu_B(x_m)\}$

theo Zadeh ta có thể xác đinh ngay được R thông qua tích dyadic, tức là tích của một véctơ với một véctơ chuyển vị:

$$R = \underline{\mu}_A \underline{\mu}_B{}^T$$

Trong đó nếu quy tắc áp dụng là MAX - MIN thì phép nhân phải được thay bằng phép tính lấy cực tiểu (min), với quy tắc MAX - PROD thì thực hiện phép nhân như bình thường.

Ví dụ: Luật điều khiển: **Nếu χ = A Thì γ = B**. Hãy xây dựng ma trận R của luật $\mu_{A \Rightarrow B}(x, y)$.

Với 5 điểm rời rạc của X (cơ sở của A) ta có:

$\{x_1, x_2, x_3, x_4, x_5\} = \{10, 20, 30, 40, 50\}$ tương ứng $\underline{\mu}_A{}^T = \{0; 0.5; 1; 0.5; 0\}$ Và Với 5 điểm rời rạc của Y (cơ sở của B)

$\{y_1, y_2, y_3, yx_4, y_5\} = \{0.5, 0.6, 0.7, 0.8, 0.9\}$ Tương ứng $\underline{\mu}_B{}^T = \{0; 0.5; l; 0.5; 0\}$.

Nếu sử dụng quy tắc MAX-MIN (phép nhân được thay bằng min) ma trận hợp thành R sẽ như sau:

$$R = \begin{pmatrix} 0 \\ 0.5 \\ 1 \\ 0.5 \\ 0 \end{pmatrix} \cdot (0\ 0.5\ 1\ 0.5\ 0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 1 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Nếu sử dụng quy tắc MAX-PROD (phép nhân thực hiện bình thường) ta có ma trận hợp thành R là:

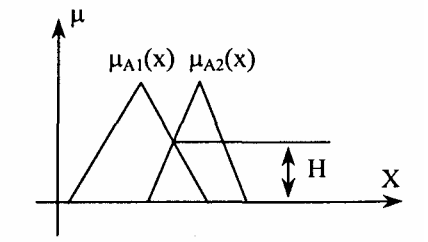
$$R = \begin{pmatrix} 0 \\ 0.5 \\ 1 \\ 0.5 \\ 0 \end{pmatrix} (0\ 0.5\ 1\ 0.5\ 0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.5 & 0.25 & 0 \\ 0 & 0.5 & 1 & 0.5 & 0 \\ 0 & 0.25 & 0.5 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

1.5.6. Luật hợp thành đơn có cấu trúc MISO

Xét một mệnh đề hợp thành với d mệnh đề điều kiện:

Nếu $\chi_1 = A_1$ và $\chi_2 = A_2$ và … và $\chi_d = A_d$ thì $\gamma = B$

Bao gồm d biến ngôn ngữ đầu vào χ_1, χ_2,…, χ_d và một biến đầu ra γ.



Hình 1.12. Xác định độ thoả mãn H

Việc mô hình hoá mệnh đề trên cũng được thực hiện tương tự như việc mô hình hoá mệnh đề hợp thành có một điều kiện, trong đó liên kết và giữa các mệnh đề (hay giá trị mờ) được thực hiện bằng phép giao các tập mờ A_1,

$A_2,\ldots,A_n$ Với nhau theo công thúc:

$$\mu_{A1 \cap A2}(x) = \min \{\mu_{A1}(x), \mu_{A2}(x)\}.$$

Kết quả của phép giao sẽ là độ thoả mãn H của luật (hình 1-12).

Các bước xây dựng luật hợp thành R như sau:

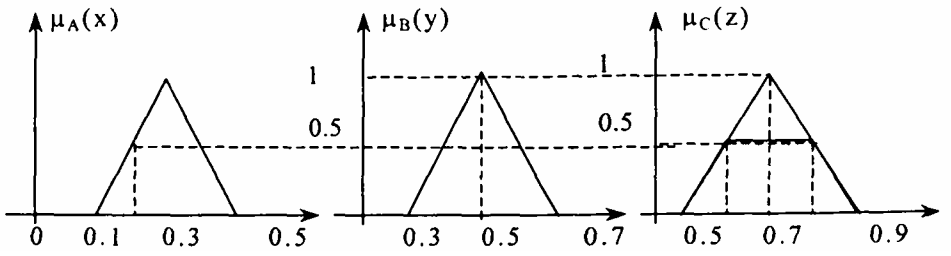
1- Rời rạc hoá miền xác định hàm liên thuộc $\mu_{A1}(x_1)$, $\mu_{A2}(x_2),\ldots,$ $\mu_{Ad}(x_d)$, $\mu_B(y)$ của các mệnh đề điều kiện và mệnh đề kết luận.

2- Xác định độ thoả mãn H cho tùng véctơ các giá trị rõ đầu vào là véctơ tổ hợp d điểm mẫu thuộc miền xác định của các hàm liên thuộc $\mu_A(x)$, (i = 1, 2,.., d).

Chẳng hạn với một véctơ các giá trị rõ đầu vào:

$\underline{x} = \begin{pmatrix} c_1 \\ \ldots\ldots \\ c_d \end{pmatrix}$ trong đó c_i (i= 1,2,...,d) là một trong các điểm mẫu trong miền xác định của $\mu_{Ai}(x)$ thì:

$H = \text{MIN}\{\mu_{A1}(c_1), \mu_{A2}(c_2),\ldots, \mu_{Ad}(c_d)\}$



Hình 1.13. Xây dựng R cho luật hợp thành hai mệnh đề điều kiện

3- Lập R gồm các hàm liên thuộc giá trị mờ đâu ra cho từng véctơ các giá trị đầu vào theo nguyên tắc:

$\mu_{B'}(y)$= MIN {H, $\mu_B(y)$} Nếu sử dụng quy tắc MAX-MIN

$\mu_{B'}(y)$= H, $\mu_B(y)$ Nếu sử dụng quy tắc MAX-PROD.

Chú ý: Đối với luật hợp thành R có d mệnh đề điều kiện không thể biểu diễn dưới dạng ma trận được nữa mà thành một lưới trong không gian d + 1 chiều.

Thật vậy, xét một mệnh đề hợp thành với hai mệnh đề điều kiện:

Nếu $\chi = A$ và $\gamma = B$ thì $\zeta = C$

Luật hợp thành R của nó có dạng như hình 2.12:

$$R: A \wedge B \Rightarrow C$$

Các bước xây dựng R như sau:

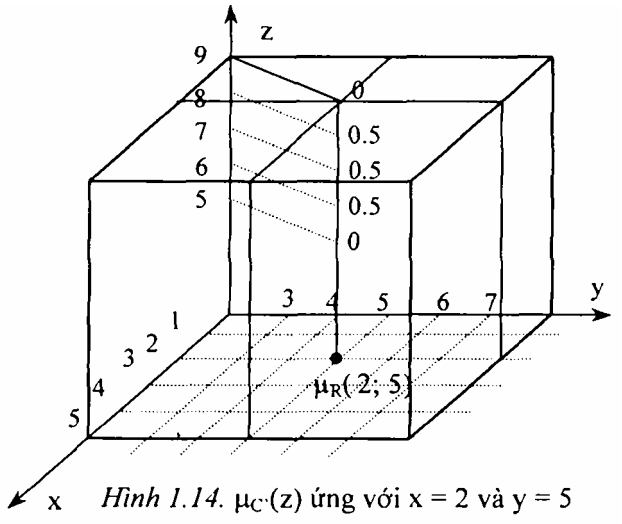
1. Rời rạc hoá các hàm liên thuộc:

- Hàm liên thuộc $\mu_A(x)$ được rời rạc hoá tại 5 điểm: $x \in \{1; 2; 3; 4; 5\}$.

- Hàm liên thuộc $\mu_B(y)$ được rời rạc hoá tạt 5 điểm: $y \in \{3; 4; 5; 6; 7\}$.

- Hàm liên thuộc $\mu_C(z)$ được rời rạc hoá tại 5 điểm: $z \in \{5; 6; 7; 8; 9\}$.

2. Lập R gồm các hàm liên thuộc cho từng vectơ giá trị đầu vào và ứng với từng cặp điểm đầu vào là một hàm liên thuộc $\mu_{C'}(z)$ của biến mờ đầu ra C' (hình 1.14).



Hình 1.14. $\mu_{C'}(z)$ ứng với x = 2 và y = 5

1.5.7. Luật của nhiều mệnh đề hợp thành

Trong thực tế hầu như không bộ Điều khiển mờ nào chỉ làm việc với một mệnh đề hợp thành mà thông thường với nhiều mệnh đề hợp thành? hay còn gọi là một tập các luật điều khiển R_k. sau đây ta sẽ trinh bày cách liên kết các luật điều khiển riêng rẽ R_k lại với nhau trong một bộ điều khiển chung và qua đó mà nêu bật được ý nghĩa của ký hiệu "MAX" sử dụng trong tên gọi luật hợp thành như MAX- MIN hay MAX-PROD.

a) Luật hợp thành của hai mệnh đề hợp thành

Xét luật điều khiển gồm hai mệnh đề hợp thành:

R1: Nếu $\chi = A_1$ thì $\gamma = B_1$ hoặc

R2: Nếu $\chi = A_2$ thì $\gamma = B_2$

Hàm liên thuộc của các tập mờ được mô tả trong hình 2.15.

Ký hiệu R là luật hợp thành chung của bộ điều khiển, ta có: $\mathbf{R = R_1 \cup R_2}$ Ký hiệu hàm liên thuộc của R_1 là $\mu_{R1}(x, y)$ và của R2 là $\mu_{R2}(x, y)$, thì theo công thức $\mu_{A \cup B}(x) = \max \{\mu_A(x), \mu_B(x)\}$.

Hàm liên thuộc của R sẽ được xác định: $\mu_R(x, y) = \max \{\mu_{R1}(x, y), \mu_{R2}(x, y)\}$. Với một giá trị rõ x_0 tại đầu vào, ta có độ thoả mãn của các mệnh đề điều kiện như sau:

Đối với luật điều khiển R_1:

- Độ thoả mãn: $H_1 = \mu_{A1}(x_0)$

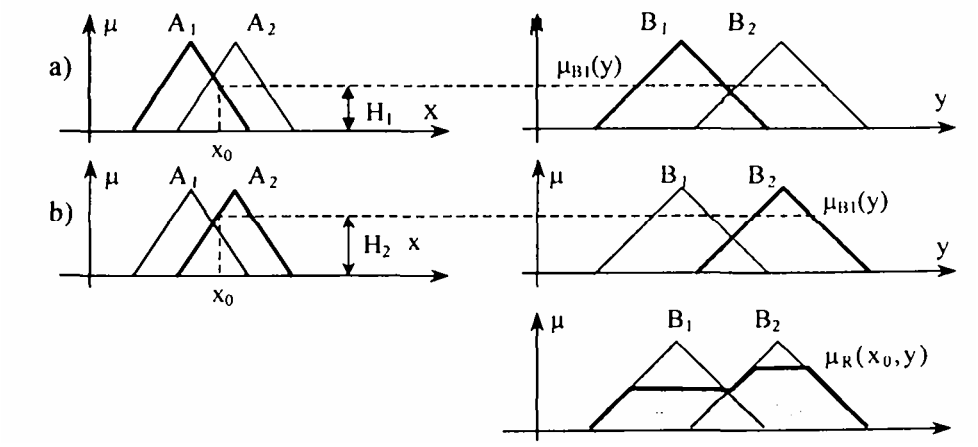
- Giá trị mờ đầu ra B_1: $\mu_{B1}(y) = \min\{H_1, \mu_{B1}(y)\}$(hình 2.15a).

Đối với luật điều khiển R_2:

- Độ thoả mãn: $H_2 = \mu_{A2}(x_0)$

- Giá trị mờ đầu ra B_2: $\mu_{B2}(y) = \min\{H_2, \mu_{B2}(y)\}$(hình 2.15b).

Từ đây ta có: $\mu_R(x_0, y) = MAX\{\mu_{B1}(y), \mu_{B2}(y)\}$



Hình 2.15. hàm liên thuộc của luật Điều khiển theo quy tắc MAX-MIN

a) Xác định hàm liên thuộc đầu ra của luật Điều khiển thứ nhất.

b) Xác định hàm liên thuộc đầu ra của luật điều khiển thứ hai.

c) Hàm liên thuộc đầu ra của luật hợp thành.

Đó chính là hàm liên thuộc của giá trị mờ đầu ra B' của bộ điều khiển gồm hai luật điều khiển $R = R_1 \cup R_2$ khi đầu vào là một giá trị rõ x_0 (hình 2.15c).

Để xác định luật hợp thành chung R, trước hết hai cơ sở X và Y của các giá trị A_1, A_2 và B_1, B_2 được rời rạc hoá, giả sử tại các điểm:

$X = \{x_1, x_2, x_3,\ldots,x_n\}$ (n điểm mẫu)

$Y = \{y_1, y_2, y_3,\ldots,y_m\}$ (m điểm mẫu).

Giá trị của các hàm liên thuộc $\mu_{A1}(x)$, $\mu_{A2}(x)$, $\mu_{B1}(y)$, $\mu_{B2}(y)$ sau khi rời rạc hoá là

$$\mu^T_{A1} = (\mu_{A1}(x_1), \mu_{A1}(x_2),..., \mu_{A1}(x_n))$$
$$\mu^T_{A2} = (\mu_{A2}(x_1), \mu_{A2}(x_2),..., \mu_{A2}(x_n))$$
$$\mu^T_{B1} = (\mu_{B1}(y_1), \mu_{B1}(y_2),..., \mu_{B1}(y_n))$$
$$\mu^T_{B2} = (\mu_{B2}(y_1), \mu_{B2}(y_2),..., \mu_{B2}(y_n)).$$

Từ đây suy ra:

$$R_1 = \underline{\mu}_{A1} \cdot \underline{\mu}^T_{B1} = \begin{pmatrix} r^1_{11} \ldots\ldots r^1_{1m} \\ \ldots\ldots\ldots \\ r^1_{n1} \ldots\ldots r^1_{nm} \end{pmatrix}$$

$$R_2 = \underline{\mu}_{A2} \cdot \underline{\mu}^T_{B2} = \begin{pmatrix} r^2_{11} \ldots\ldots r^2_{1m} \\ \ldots\ldots\ldots \\ r^2_{n1} \ldots\ldots r^2_{nm} \end{pmatrix}$$

và do đó luật hợp thành chung sẽ là:

$$R = R_1 \cup R_2 = \begin{pmatrix} \max\{r^1_{11}, r^2_{11}\} \ldots \max\{r^1_{1m}, r^1_{1m}\} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \max\{r^1_{n1}, r^2_{n1}\} \ldots \max\{r^1_{nm}, r^{\ 2}_{nm}\} \end{pmatrix}.$$

b) Luật hợp thành của nhiều mệnh đề hợp thành

Xét luật điều khiển R gồm p mệnh đề hợp thành:

R_1: Nếu $\chi = A_1$ Thì $\Upsilon = B_1$ hoặc

R_2: Nếu $\chi = A_2$ Thì $\Upsilon = B_2$ hoặc

.....

R_p: Nếu $\chi = Ap$ Thì $\Upsilon = B_p$

trong đó các giá trị mờ A_1, A_2,…, A_p có cùng cơ sở X và B_1, B_2,…, B_p có cùng cơ sở Y.

Gọi hàm liên thuộc của A_k và B_k là $\mu_{Ak}(x)$ và $\mu_{Bk}(y)$ với k = 1, 2,..., p.

Thuật toán triển khai: $R = R_1 \cup R_2 \cup \ldots \cup R_p$ được thực hiện theo các bước sau:

Bước 1: Rời rạc hoá X tại n điểm (x_1, x_2, x_3,…, x_n) Và Y tại m điểm (y_1, y_2, y_3,…, y_n)

Bước 2: Xác định các véctơ $\underline{\mu}_{Ak}$ và $\underline{\mu}_{Bk}$ (k = 1, 2,..,p) tại các điểm rời rạc theo biểu thức:

$$\underline{\mu}^T_{Ak} = \{\mu_{Ak}(x_1), \mu_{Ak}(x_2),\ldots, \mu_{Ak}(x_n)\}$$

$$\underline{\mu}^T_{Bk} = \{\mu_{Bk}(y_1), \mu_{Bk}(y_2),\ldots, \mu_{Bk}(y_n)\}$$

Bước 3: Xác định mô hình (ma trận) R_k cho mệnh đề thứ k

$$R_k = \mu_{Ak}.\mu^T_{Bk} = (r^k_{ij}),\ i = 1, 2,\ldots, n \text{ và } j = 1, 2,\ldots,m$$

trong đó phép (.) được thay bằng phép tính lấy cực tiểu min khi sử dụng nguyên tắc MAX-MIN và sử dụng phép nhân bình thường khi sử dụng nguyên tắc MAX- PROD.

Bước 4: Xác định luật hợp thành $R = \text{Max}\ (r^k_{ij})$ với k = 1, 2,..., p}.

1.5.7. Luật hợp thành SUM-MIN và SUM-PROD

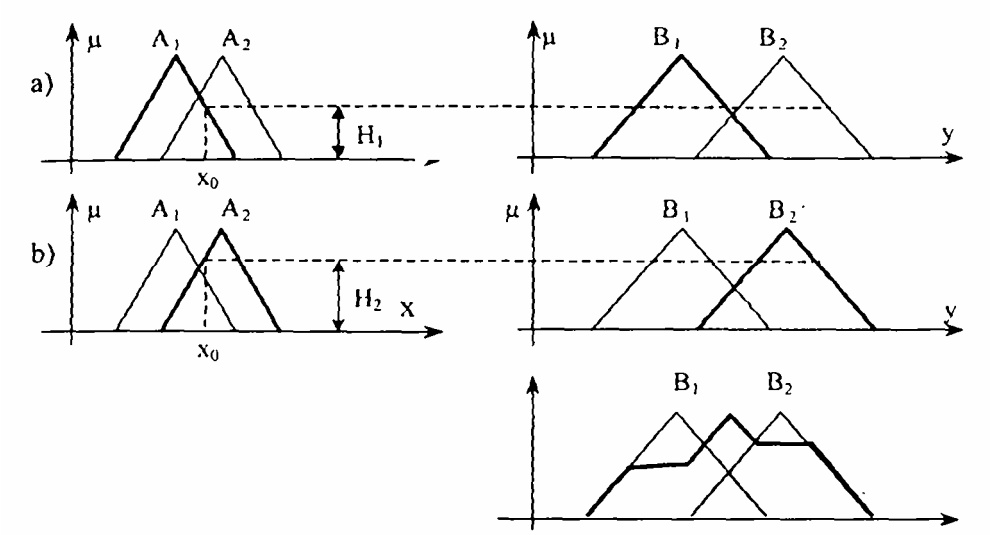
Ở phần trên, chúng ta đã tìm hiểu phương pháp xây dựng luật hợp thành chung R cho một tập gồm nhiều mệnh đề hợp thành R_k được liên kết với nhau bằng phép hợp theo biểu thức: $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$. Kiểu liên kết này không có tính thống kê. Ví dụ khi đa số các mệnh đề hợp thành R_k có cùng một giá trị đầu ra nhưng không phải là giá trị lớn nhất sẽ không được để ý tới và bị mất trong kết quả chung. Để khắc phục nhược điểm này phép hợp Lukasiewicz theo biểu:

$\mu_{A \cup B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\}$ thay cho $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$

để liên kết các luật điều khiển Rk lại với nhau thành luật hợp thành chung R

$$R = \min\left\{1, \sum_{k=1}^{p} R_k\right\}$$

trong đó phép lấy cực tiểu min được thực hiện giữa số 1 và từng phần tử của ma trận tổng. Ở công thức này, R được xác định bằng cách cộng các R_k Của các mệnh đề hợp thành nên luật hợp thành chung R theo liên kết Lukasiewicz sẽ có tên gọi là SUM-MIN hoặc SUM-PROD.



Hình 2.16. Hàm liên thuộc của hợp hai luật điều khiển theo quy tắc SUM-MIN

Thuật toán triển khai R theo quy tắc SUM-MIN hay SUM-PROD cũng bao gồm các bước như khi triển khai với quy tắc MAX-MIN hoặc MAX-PROD đã trình bày ở mục trên chỉ khác ở bước 4 ta sử dụng công thức: $R = \min\left\{1, \sum_{k=1}^{n} R_k\right\}$

Hình 1.16 là một ví dụ về mô hình hoá R gồm hai mệnh đề hợp thành theo quy tắc SUM-MIN.

1.6. GIẢI MỜ

Từ một giá trị rõ x_0 ở đầu vào, sau khi qua khối luật hợp thành ta có tập

mờ đầu ra B'. Vấn đề đặt ra là cần phải xác định giá trị rõ y_0 từ tập mờ đầu ra đó. Muốn vậy ta cần thực hiện việc giải mờ.

Giải mờ là quá trình xác định một giá trị rõ y_0 nào đó có thể chấp nhận được từ hàm liên thuộc $\mu_{B'}(y)$ của giá trị mờ B' (tập mờ B').

Có hai phương pháp giải mờ chính là phương pháp cực đại và phương pháp điểm trọng tâm.

2.6.1. Phương pháp cực đại

Để giải mờ theo phương pháp cực đại, ta cần thực hiện 2 bước:

- Xác định miền chứa giá trị rõ y_0 (miền G): Đó là miền mà tại đó hàm liên thuộc $\mu_{B'}(y)$ đạt giá trị cực đại (độ cao H của tập mờ B'), tức là miền:

$$G = \{y \in Y | \mu_{B'}(y) = H\}$$

- Xác định y_0 có thể chấp nhận được từ G.

Hình 1.17 là tập mờ đầu ra của một luật hợp thành gồm 2 mệnh đề hợp thành:

$$R_1: \text{Nếu } \chi = A_1 \text{ Thì } \gamma = B_1$$

$$R_2: \text{Nếu } \chi = A_2 \text{ Thì } \gamma = B_2$$

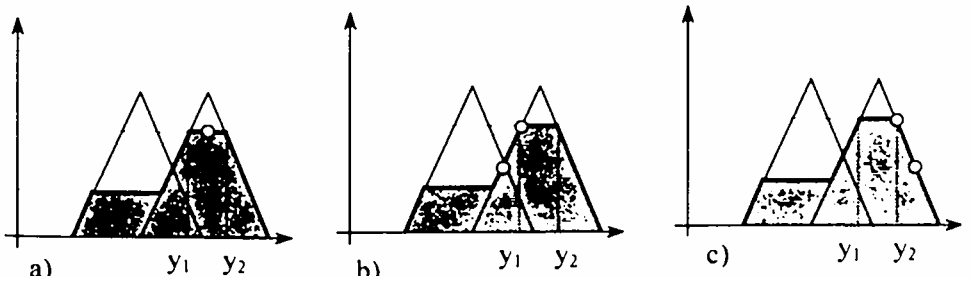
Miền chứa giá trị rõ G là khoảng $[y_1, y_2]$ của miền giá trị của tập mờ đầu ra B_2 của luật điều khiển:

$$R_2: \text{Nếu } \chi = A_2 \text{ Thì } \gamma = B_2$$

với y_1 là điểm cận trái của G $\left(y_1 = \inf_{y \in G}(y) \right)$ và y_2 là điểm cận phải của G $\left(y_1 = \sup_{y \in G}(y) \right)$. Khi đó, luật R_2 được gọi là luật Điều khiển quyết định.

Vậy luật điều khiển quyết định là luật R_k, $k \in \{1, 2, \ldots, p\}$ mà giá trị mở đầu ra của nó có độ cao lớn nhất (Bằng độ cao H của B').

Dê xác định y_0 trong khoảng $[y_1, y_2]$ ta có thể áp dụng theo một trong ba nguyên lý: Nguyên lý trung bình; nguyên lý cận trái và nguyên lý cận phải.



Hình 1.17a.b.c. Các nguyên lý giải mờ theo phương pháp cực dại

a) Nguyên lý trung binh

Giá trị rõ y_1 sẽ là trung bình cộng của y_1 và y_2

$$y_0 = \frac{y_1 + y_2}{2}$$

b) Nguyên lý cận trái

Giá trị rõ y_0 được lấy bằng cận trái y_1 của G

$$\left(y_1 = \inf_{y \in G}(y) \right).$$

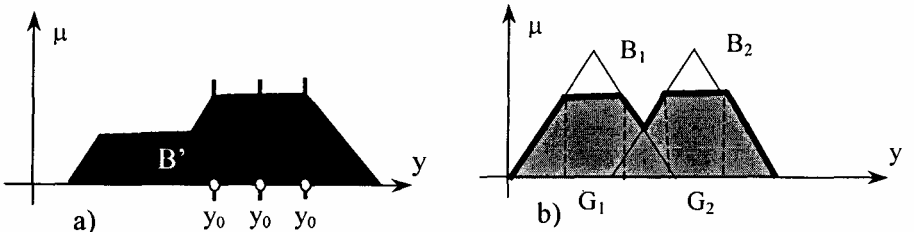
c) Nguyên lý cận phải

Giá trị rõ y_0 được lấy bằng cận phải y_2 của G

$$\left(y_2 = \sup_{y \in G}(y) \right).$$

Nhận xét:

+ Giá trị rõ y_0 lấy theo nguyên lý trung bình sẽ không phụ thuộc vào độ thoả mãn của luật điều khiển quyết định nếu tập mờ B' là tập đều (hình 1.17a), còn theo nguyên lý cận trái và cận phải, giá trị rõ y_0 Phụ thuộc tuyến tính vào độ thoả mãn của luật điều khiển quyết định (hình 1.17b,c).

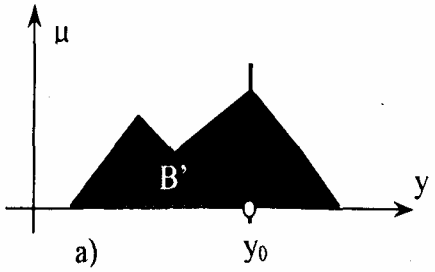


Hình 1.18. a) y_0 với các nguyên tắc chọn khác nhau

b) Hàm liên thuộc B' có miền G không liên thông

+ Sai lệch của ba giá trị rõ, xác định theo nguyên lý trung bình, cận trái hay cận phải sẽ càng lớn nếu độ thoả mãn H của luật điều khiển càng nhỏ (hình 1.18a).

+ Khi miền G là miền không liên thông sử dụng phương pháp cực đại sẽ không chính xác (hình 2.18b).



Hình 1.19. Hàm liên thuộc B' đối với luật hợp thành MAX-PROD

+ Đối với luật hợp thành MAX-PROD, miền G chỉ có một điểm duy nhất, do đó kết quả giải mờ theo cả 3 nguyên lý đề giống nhau (hình 1.19).

1.6.2. Phương pháp điểm trọng tâm

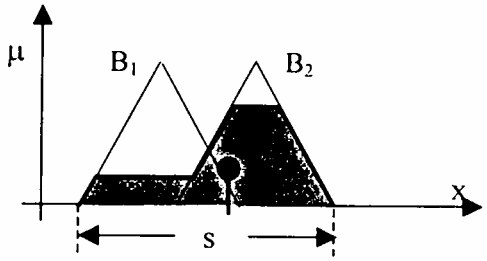
Giải mờ theo phương pháp điểm trọng tâm sẽ cho ra kết quả y' là hoành độ của điểm trọng tâm miền được bao bởi trục hoành và đường $\mu_{B'}(y)$ (hình 1.20). Công thức xác định y_0 theo phương pháp điểm trọng tâm như sau:

$$y' = \frac{\int_S y\mu_{B'}(y)dy}{\int_S \mu_{B'}(y)dy}$$

Với s là miền xác định của tập mờ B'.

a) Phương pháp điểm trọng tâm cho luật hợp thành SUM-MIN

Giả sử có q luật điều khiển được triển khai. Khi đó mỗi giá trị mờ B' tại đầu ra của bộ điều khiển sẽ là tổng của



Hình 2.20. Giá trị rõ y' là hoành độ của điểm trọng tâm

q giá trị mờ đầu ra của từng luật hợp thành. Ký hiệu giá trị mờ đầu ra của luật điều khiển thứ k là $\mu_{B'K}(y)$ với k = 1,2,...,q. Với quy tắc SUM- MIN, hàm liên thuộc $\mu_{B'}(x)$ sẽ là:

$$\mu_{B'}(y) = \sum_{k=1}^{q} \mu_{B'k}(y).$$

sau khi biên đổi, ta có:

$$y' = \frac{\int_S [y \sum_{k=1}^{q} \mu_{B'k}(y)]dy}{\int_S \sum_{k=1}^{q} \mu_{B'k}(y)dy} = \frac{\sum_{k=1}^{q} [\int_S y\mu_{B'k}(y)dy]}{\sum_{k=1}^{q} [\int_S \mu_{B'k}(y)dy]} = \frac{\sum_{k=1}^{q} M_k}{\sum_{k=1}^{q} A_k}$$

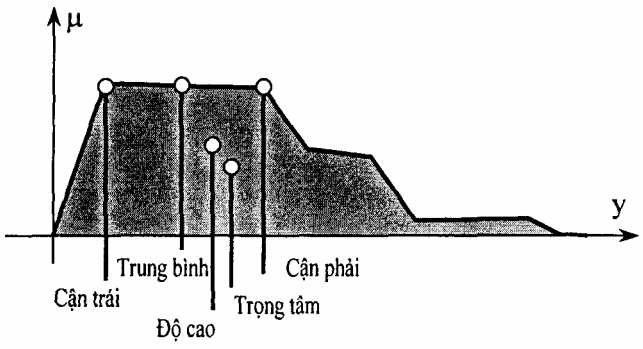
trong đó: $M_k = \int_S y\mu_{B'k}(y)dy$ và $A_k = \int_S \mu_{B'k}(y)dy$.

b) Phương pháp độ cao Sử dụng công thức:

$$y' = \frac{\int_S [y \sum_{k=1}^{q} \mu_{B'k}(y)]dy}{\int_S \sum_{k=1}^{q} \mu_{B'k}(y)dy} = \frac{\sum_{k=1}^{q} [\int_S y\mu_{B'k}(y)dy]}{\sum_{k=1}^{q} [\int_S \mu_{B'k}(y)dy]} = \frac{\sum_{k=1}^{q} M_k}{\sum_{k=1}^{q} A_k}$$

Cho cả hai luật hợp thành MAX-MIN và SUM-MIN với thêm một giả thiết là mỗi tập mờ $\mu_{B'K}(y)$ được xấp xỉ bằng một cặp giá trị (y_k, H_k) duy nhất (singleton), trong đó H_k là độ cao của $\mu_{B'K}(y)$ và y_k là một điểm mẫu trong miền giá trị của $\mu_{B'K}(y)$

Ta có: $\mu_{B'k}(y) = H_k$ và $y' = \frac{\sum_{k=1}^{q} y_k H_k}{\sum_{k=1}^{q} H_k}$.



Hình 1.21. So sánh các phương pháp giải mờ

Chú ý: Tuỳ hình dạng hàm liên thuộc B’ mà sai khác giữa các phương pháp giải mờ có khác nhau. Hình 1.21 cho biết kết quả các phương pháp giải mờ ứng với một hàm liên thuộc B’ cụ thể.

Chương 2

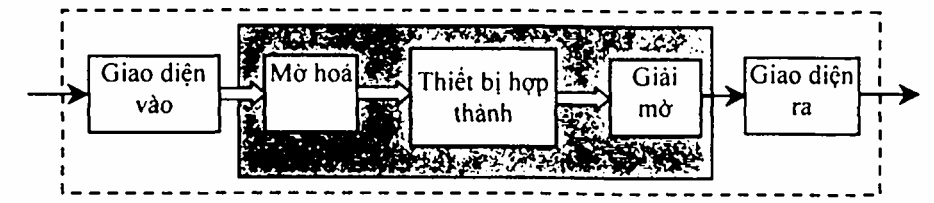
ĐIỀU KHIỂN MỜ

2.1. CẤU TRÚC CỦA BỘ ĐIỀU KHIỂN MỜ

2.1.1. Sơ đồ khối bộ điều khiển mờ

Hoạt động của một bộ điều khiển mờ phụ thuộc vào kinh nghiệm và phương pháp rút ra kết luận theo tư duy của con người sau đó được cài đặt vào máy tính trên cơ sở logic mờ.

Một bộ điều khiển mờ bao gồm 3 khối cơ bản: Khối mờ hoá, thiết bị hợp thành và khối giải mờ. Ngoài ra còn có khối giao diện vào và giao diện ra (hình 2.1).



Hình 2.1. Các khối chức năng của bộ Điều khiển mờ

- Khối mờ hoá có chức năng chuyển mỗi giá tri rõ của biến ngôn ngữ đầu vào thành véctơ μ có số phần tử bằng số tập mờ đầu vào.

-Thiết bị hợp thành mà bản chất của nó sự triển khai luật hợp thành R được xây dựng trên cơ sở luật điều khiển.

- Khối giải mờ có nhiệm vụ chuyển tập mờ đầu ra thành giá trị rõ y_0 (ứng với mỗi giá tri rõ x_0 đề điều khiển đối tượng.

- Giao diện đầu vào thực hiện việc tông hợp và chuyển đổi tin hiệu vào (từ tương tự sang số), ngoài ra còn có thể có thểm các khâu phụ trợ đê thực hiện bài toán động như tích phân, vi phân....

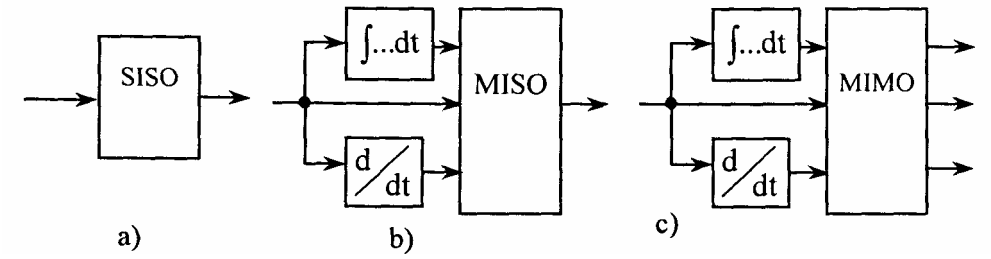
- Giao diện đầu ra thực hiện chuyển đổi tín hiệu ra (từ số sang tương tự) để điều khiển đối tượng.

Nguyên tắc tổng hợp một bộ điều khiển mờ hoàn toàn dựa vào những phương pháp toán học trên cơ sở định nghĩa các biến ngôn ngữ vào/ra và sự lựa chọn những luật điều khiển. Do các bộ điều khiển mờ có khả năng xử lý các giá trị vào/ra biểu diễn dưới dạng dấu phẩy động với độ chính xác cao nên chúng hoàn toàn đáp ứng được các yêu cầu của một bài toán điều khiển "rõ ràng" và "chính xác"

2.1.2. Phân loại bộ điều khiển mờ

Cũng giống như điều khiển kinh điển, bộ điều khiển mờ được phân loại dựa trên các quan điểm khác nhau:

Theo số lượng đầu vào và đầu ra ta phân ra bộ Điều khiển mờ "Một vào - một ra" (SISO); "Nhiều vào - một ra" (MISO); "Nhiều vào - nhiều ra" (MIMO) (hình 2.2a,b,c).



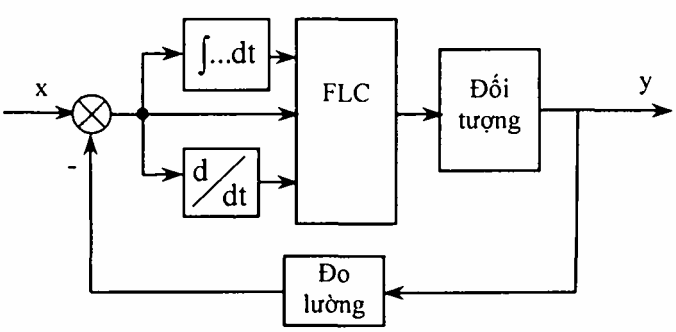
Hình 2.2a,b,c. Các bộ điều khiển mờ

Bộ điều khiển mờ MIMO rất khó cài đặt thiết bị hợp thành. Mặt khác, một bộ điều khiển mờ có m đầu ra dễ dàng cài đặt thành m bộ điều khiển mờ chỉ có một đầu ra vì vậy bộ điều khiển mờ MIMO chỉ có ý nghĩa về lý thuyết, trong thực tế không dùng.

- Theo bản chất của tín hiệu đưa vào bộ điều khiển ta phân ra bộ điều khiển mờ tĩnh và bộ điều khiển mờ động. Bộ điều khiển mờ tĩnh chỉ có khả năng xử lý các tín hiệu hiện thời, bộ điều khiển mờ động có sự tham gia của các giá trị đạo hàm hay tích phân của tín hiệu, chúng được ứng dụng cho các bài toán điều khiển động. Bộ điều khiển mờ tĩnh chỉ có khả năng xử lý các giá trị tín hiệu hiện thời. Để mở rộng miền ứng dụng của chúng vào các bài toán điều khiển động, các khâu động học cần thiết sẽ được nối thêm vào bộ điều khiển mờ tĩnh nhằm cung cấp cho bộ điều khiển các giá trị đạo hàm hay tích phân của tín hiệu. Cùng với những khâu động học bổ sung này, bộ điều khiển tĩnh sẽ trở thành bộ Điều khiển mờ động.

2.1.3. Các bước tổng hợp bộ điều khiển mờ

Cấu trúc tổng quát của một hệ điều khiển mờ được chỉ ra trên hình 2.3.



Hình 2.3. Cấu trúc tổng quát một hệ mờ

Với một miền compact $X \subset R_n$ (n là số đầu vào) các giá trị vật lý của biến ngôn ngữ đầu vào và một đường phi tuyến g(x) tuỳ ý nhưng liên tục cùng các đạo hàm của nó trên X thì bao giờ cũng tồn tại một bộ điều khiển mờ cơ bản có quan hệ:

$\underset{x \in X}{Sup} |y(x) - g(x)| < \varepsilon$ với ε là một số thực dương bất kỳ cho trước.

Điều đó cho thấy kỹ thuật điều khiển mờ có thể giải quyết được một bài toán tổng hợp điều khiển (tĩnh) phi tuyến bất kỳ.

Để tổng hợp được các bộ Điều khiển mờ và cho nó hoạt động một cách hoàn thiện ta cần thực hiện qua các bước sau:

1- Khảo sát đối tượng, từ đó định nghĩa tất cả các biến ngôn ngữ vào, ra và miền xác định của chúng. Trong bước này chúng ta cần chú ý một số đặc điểm cơ bản của đối tượng điều khiển như: Đối tượng biến đổi nhanh hay chậm? có trễ hay không? tính phi tuyến nhiều hay ít?,... Đây là những thông tin rất quan trọng để quyết định miền xác định của các biến ngôn ngữ đầu vào, nhất là các biến động học (vận tốc, gia tốc,...). Đối với tín hiệu biến thiên nhanh cần chọn miền xác định của vận tốc và gia tốc lớn và ngược lại.

2- Mờ hoá các biến ngôn ngữ vào/ra: Trong bước này chúng ta cần xác định số lượng tập mờ và hình dạng các hàm liên thuộc cho mỗi biến ngôn ngữ. Số lượng các tập mờ cho mỗi biến ngôn ngữ được chọn tuỳ ý. Tuy nhiên nếu chọn ít quá thì việc điều chỉnh sẽ không mịn, chọn nhiều quá sẽ khó khăn

khi cài đặt luật hợp thành, quá trình tính toán lâu, hệ thống dễ mất ổn định. Hình dạng các hàm liên thuộc có thể chọn hình tam giác, hình thang, hàm Gaus,...

3- Xây dựng các luật điều khiển (mệnh đề hợp thành): Đây là bước quan trọng nhất và khó khăn nhất trong quá trình thiết kế bộ điều khiển mờ. Việc xây dựng luật điều khiển phụ thuộc rất nhiều vào tri thức và kinh nghiệm vận hành hệ thống của các chuyên gia. Hiện nay ta thường sử dụng một vài nguyên tắc xây dựng luật hợp thành đủ để hệ thống làm việc, sau đó mô phỏng vả chỉnh định dần các luật hoặc áp dụng một số thuật toán tối ưu (được trình bày ở phần sau).

4- Chọn thiết bị hợp thành (MAX-MIN hoặc MAX-PROD hoặc SUM-MIN hoặc SUM-PRROD) và chọn nguyên tắc giải mờ (Trung bình, cận trái, cận phải, điểm trọng tâm, độ cao).

5- Tối ưu hệ thống: Sau khi thiết kế xong bộ điều khiển mờ, ta cần mô hình hoá và mô phỏng hệ thống để kiểm tra kết quả, đồng thời chỉnh định lại một số tham số để có chế độ làm việc tối ưu. Các tham số có thể điều chỉnh trong bước này là. Thêm, bớt luật điều khiển; Thay đổi trọng số các luật; Thay đổi hình dạng và miền xác định của các hàm liên thuộc.

2.2. BỘ ĐIỀU KHIỂN MỜ TĨNH

2.2.1. Khái niệm

Bộ điều khiển tĩnh là bộ điều khiển mờ có quan hệ vào/ra y(x), với x là đầu vào và y là đầu ra, theo dạng một phương trình đại số (tuyến tính hoặc phi tuyến). Bộ điều khiển mờ tĩnh không xét tới các yếu tố "động" của đối tượng (vận tốc, gia tốc,…). Các bộ điều khiển tĩnh điển hình là bộ khuếch đại P, bộ điều khiển re lay hai vị trí, ba vị trí,...

2.2.2. Thuật toán tổng hợp một bộ điều khiển mờ tĩnh

Các bước tổng hợp bộ điều khiển mờ tĩnh về cơ bản giống các bước chung để tổng hợp bộ điều khiển mờ như đã trình bày ở trên. Để hiểu kỹ hơn ta xét ví dụ cụ thể sau:

Ví dụ: Hãy thiết kế bộ điều khiển mờ tĩnh SISO có hàm truyền đạt y = f(x) trong khoảng x = $[a_1,a_2]$ tương ứng với y trong khoảng y $[\beta_1, \beta_2]$.

Bước 1: Định nghĩa các tập mờ vào, ra

- Định nghĩa N tập mờ đầu vào: A_1, A_2,…, A_n trên khoảng $[a_1,a_2]$ của x có hàm liên thuộc μ_{Ai} (x) (i = 1, 2,..., Ni dạng hình tam giác cân.

- Định nghĩa N tập mờ đầu ra: B_1, B_2,…, B_N trên khoảng $[\beta_1, \beta_2]$ của y có hàm liên thuộc $\mu_{Bj}(x)$ (j = 1, 2,..., N) dạng hình tam giác cân.

Bước 2: Xây dựng luật điều khiển

Với N hàm liên thuộc đầu vào ta sẽ xây dựng được N luật điều khiển theo cấu trúc:

$$\mathbf{R^i: nêu\ \chi = A_i;\ thì\ \gamma = B_i.}$$

Bước 3: Chọn thiết bị hợp thành

Giả thiết chọn nguyên tắc triển khai SUM-PROD cho mệnh đề hợp thành, và công thức Lukasiewicz cho phép hợp thì tập mờ đâu ra B' khi đầu vào là một giá trị rõ x_0 sẽ là:

$$\mu_{B'}(y) = \mathrm{MIN}\left\{1, \sum_{i=1}^{N} \mu_{Bi}(y)\mu_{Ai}(x_0)\right\}. \quad (2.1)$$

vì $\mu_{Bi}(y)$ là một hàm Kronecker $\mu_{Bi}(y)\mu_{Ai}(x_0) = \mu_{Ai}(x_0)$ khi đó:

$$\mu_{B'}(y) = \mathrm{MIN}\left\{1, \sum_{i=1}^{N} \mu_{Bi}(y)\mu_{Ai}(x_0)\right\}. \quad (2.2)$$

Bước 4: Chọn phương pháp giải mờ

Chọn phương pháp độ cao để giải mờ, ta có:

$$y(x_0) = \frac{\sum_{i=1}^{N} y_i H_i}{\sum_{i=1}^{N} H_i} = \frac{\sum_{i=1}^{N} y_i \mu_{Ai}(x_0)}{\sum_{i=1}^{N} \mu_{Ai}(x_0)}. \quad (2.3)$$

Quan hệ truyền đạt của bộ điều khiển mờ có dạng:

$$y(x) = \frac{\sum_{i=1}^{N} y_i \mu_{Ai}(x)}{\sum_{i=1}^{N} \mu_{Ai}(x)}. \quad (2.4)$$

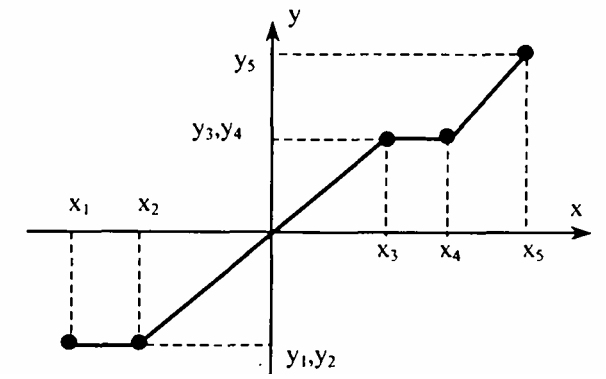
2.2.3. Tổng hợp bộ điều khiển mờ tuyến tính từng đoạn

Trong kỹ thuật nhiều khi ta cần phải thiết kế bộ điều khiển mờ với đặc tính vào - ra cho trước tuyến tính từng đoạn. Chẳng hạn, cần thiết kế bộ điều khiển mờ có đặc tính vào - ra như hình 2.4.

Thuật toán tổng hợp bộ điều khiển này giống như thuật toán tổng hợp bộ điều khiển mờ với hàm truyền đạt y(x) bất kỳ. Tuy nhiên, để các đoạn đặc tính thẳng và nối với nhau một cách liên tục tại các nút thì cần tuân thủ một số nguyên tắc sau:

+ Mỗi giá tri rõ đầu vào phải làm tích cực 2 luật điều khiển.

+ Các hàm liên thuộc đầu vào có dạng hình tam giác có đỉnh là một điểm ở nút k, có miền xác đinh là khoảng $[x_{k-1}, x_{k+1}]$ (hình 2.5a).



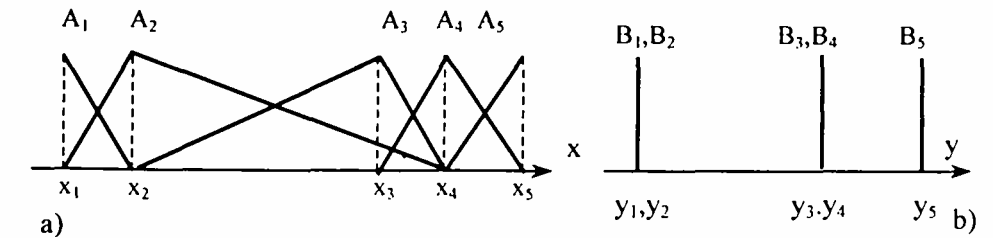
Hình 2.4. Đặc tính vào - ra cho trước

+ Các hàm liên thuộc đầu ra có dạng singleton tại các điểm nút y_k (hình 2.5b).

+ Cài đặt luật hợp thành Max-Min với luật điều khiển tổng quát:

R_k: nêu $\chi = A_k$; thì $\gamma = B_k$.

+ Giải mờ bằng phương pháp độ cao.



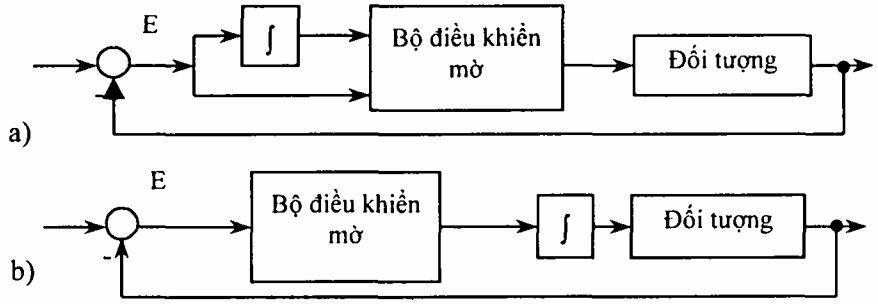
Hình 2.5 a.b. hàm liên thuộc của các biến ngôn ngữ vào, ra

2.3. BỘ ĐIỀU KHIỂN MỜ ĐỘNG

Bộ Điều khiển mờ động là bộ điều khiển mờ mà đầu vào có xét tới các trạng thái động của đối tượng như vận tốc, gia tốc, dạo hàm của gia tốc,.... Ví dụ đối với hệ điều khiển theo sai lệch thì đầu vào của bộ điều khiển mờ ngoài tín hiệu sai lệch e theo thời gian còn có các đạo hàm của sai lệch giúp cho bộ điều khiển phản ứng kịp thời với các biến động đột xuất của đối tượng.

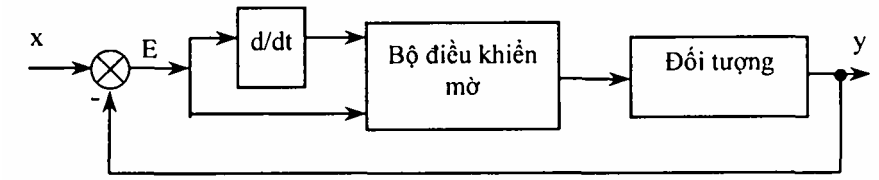
Các bộ điều khiển mờ động hay được dùng hiện nay là bộ điều khiển mờ theo luật tỉ lệ tích phân(PI), tỉ lệ vi phân (PD) và tỉ lệ vi tích phân (PID).

Một bộ điều khiển mờ theo luật I có thể thiết kế từ một bộ mờ theo luật P (bộ Điều khiển mờ tuyến tính) bằng cách mắc nối tiếp một khâu tích phân vào trước hoặc sau khối mờ đó. Do tính phi tuyến của hệ mờ, nên việc mắc khâu tích phân trước hay sau hệ mờ hoàn toàn khác nhau (hình 3.2 a,b).



Hình 2.6a,b. hệ điều khiển mờ theo luật PI

Khi mắc thêm một khâu vi phân ở đầu vào của một bộ điều khiển mờ theo luật tỉ lệ sẽ có được một bộ điều khiển mờ theo luật tỉ lệ vi phân PD (hình 2.4).



Hình 2.7. hệ điều khiển mờ theo luật PD

Các thành phần của bộ điều khiển này cũng giống như bộ điều khiển theo luật PD thông thường bao gồm sai lệch giữa tín hiệu chủ đạo và tín hiệu ra của hệ thống e và đạo hàm của sai lệch e'. Thành phần vi phân giúp cho hệ

thống phản ứng chính xác hơn với những biến đổi lớn của sai lệch theo thời gian.

Trong kỹ thuật Điều khiển kinh điển, bộ Điều khiển PID được biết đến như là một giải pháp đa năng và có miền ứng dụng rộng lớn. Đinh nghĩa về bộ điều khiển theo luật PID kinh điển trước đây vẫn có thể sử dụng cho một bộ điều khiển mờ theo luật PID. Bộ điều khiển mờ theo luật PID được thiết kế theo hai thuật toán:

- Thuật toán chỉnh định PID;

- Thuật toán PID tốc độ.

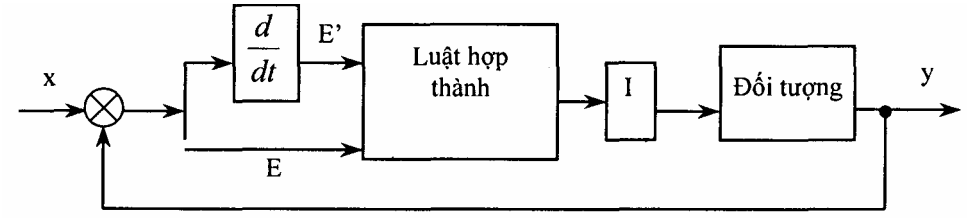
Bộ điều khiển mờ được thiết kế theo thuật toán chỉnh định PID có 3 đầu vào gồm sai lệch e giữa tín hiệu chủ đạo và tín hiệu ra, đạo hàm và tích phân của sai lệch. Đầu ra của bộ điều khiển mờ chính là tín hiệu điều khiển rút).

$$u(t) = K\left[e + \frac{1}{T_I}\int_0^t e.dt + T_D\frac{d}{dt}e\right]. \qquad (2.5)$$

Với thuật toán PID tốc độ, bộ điều khiển PID có 3 đầu vào: sai lệch e giữa tín hiệu đầu vào và tín hiệu chủ đạo, đạo hàm bậc nhất e' và đạo hàm bậc hai e" của sai lệch. Đầu ra của hệ mờ là đạo hàm $\frac{du}{dt}$ của tín hiệu điều khiển u(t).

$$\frac{du}{dt} = K\left[\frac{d}{dt}e + \frac{1}{T_I}e + \frac{d^2}{(dt)^2}e\right]. \qquad (2.6)$$

Do trong thực tế thường có một hoặc hai thành phần trong (3.6), (3.7) được bỏ qua, nên thay vì thiết kế một bộ điều khiển PID hoàn chỉnh người ta lại thường tổng hợp các bộ điều khiển PI hoặc PD.



Hình 2.8. Hệ điều khiển mờ theo luật PID

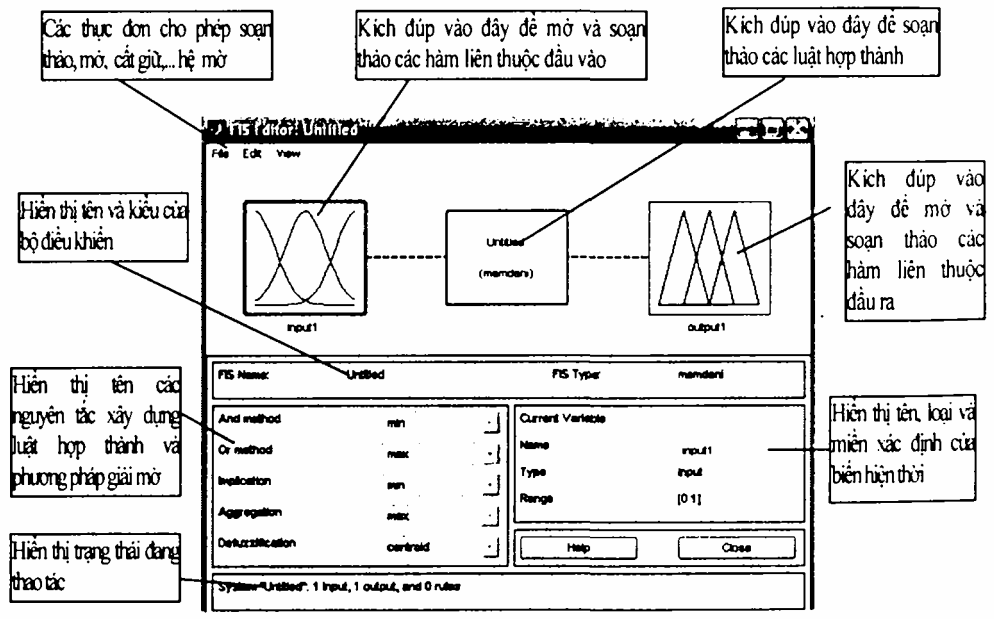
Bộ điều khiển PID mờ được thiết kế trên cơ sở của bộ điều khiển PD mờ bằng cách mắc nối tiếp ở đầu ra của bộ điều khiển PD mờ một khâu tích phân (hình 2.6).

Hiện nay đã có rất nhiều dạng cấu trúc khác nhau của PID mờ đã được nghiên cứu. Các dạng cấu trúc này thường được thiết lập trên cơ sở tách bộ điều chỉnh PID thành hai bộ điều chỉnh PD và PI (hoặc I). Việc phân chia này chỉ nhằm mục đích thiết lập các hệ luật cho PD và PI (hoặc I) gồm hai (hoặc 1) biến vào, một biên ra, thay vì phải thiết lập 3 biến vào. Hệ luật cho bộ điều chỉnh PID mờ kiểu này thường dựa trên ma trận do Mac Vicar-whelan đề xuất. Cấu trúc này không làm giảm số luật mà chỉ đơn giản cho việc tính toán.

2.4. THIẾT KẾ HỆ ĐIỀU KHIỂN MỜ BẰNG PIIẦN MỀM MATLAB

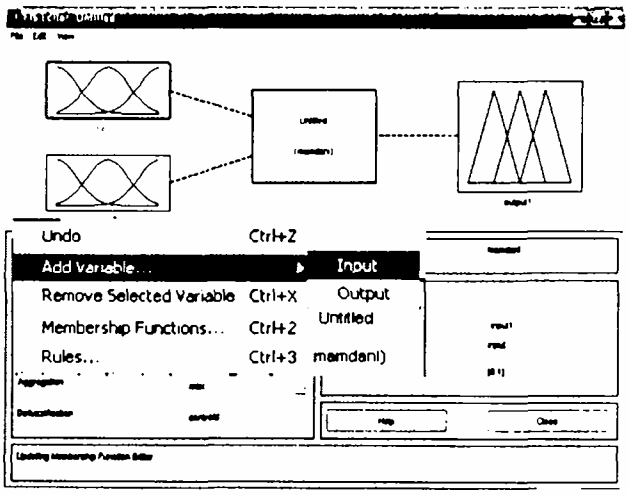
2.4.1. Giới thiệu hộp công cụ lôgic mờ

Hộp công cụ Lôgic mờ (The Fuzzy Logic Toolbox) là tổ hợp các hàm được xây dựng trên nền Matlab giúp cho việc thiết kế, mô phỏng, kiểm tra và hiệu chỉnh bộ điều khiển mờ một cách dễ dàng. Để thiết kế bộ điều khiển mờ trong hộp công cụ này, ta có thể thực hiện thông qua dòng lệnh hoặc thông qua giao diện đồ hoạ. Trong khuôn khổ cuốn sách này chỉ giới thiệu những thao tác cơ bản để thiết kế bộ điều khiển mờ thông qua giao diện đồ hoạ. Phần thiết kế thông qua dòng lệnh, ta có thề đọc trong phần **"Fuzzy Logic Toolbox"** của Malab.

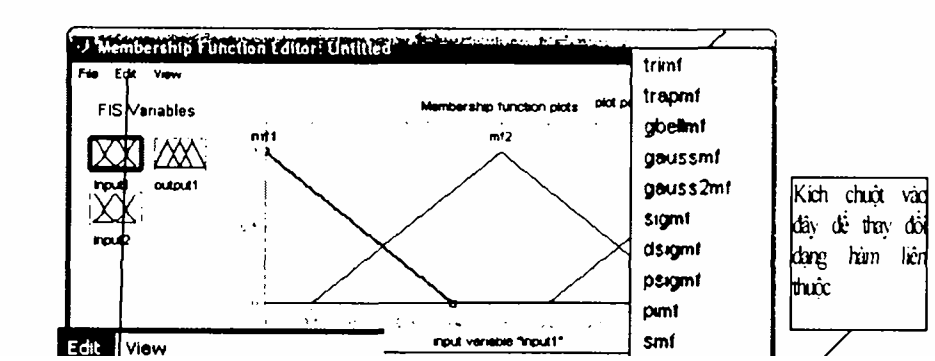


Hình 2. 9

Sau khi đã có cấu trúc của bộ Điều khiển mờ, ta tiến hành soạn thảo các hàm liên thuộc vào, hàm liên thuộc ra, các luật điều khiển.

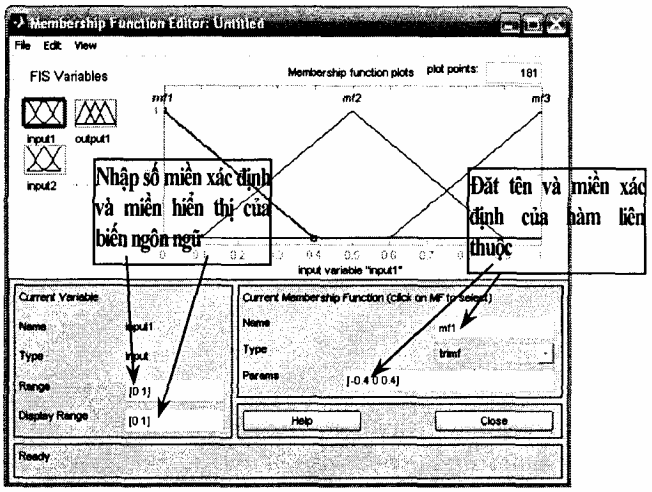


Hình 2.10

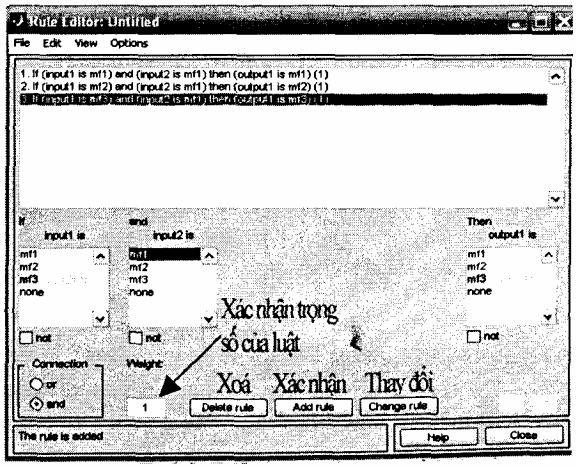


Hình 2.11

Kích đúp chuột vào biểu tượng Input (Hình 2.11)Chọn **Edit**, và chọn **Add MFs** hoặc **Add Custom MF** thêm hàm liên thuộc, chọn Remov Select MF để gỡ bỏ một hàm liên thuộc nào đó, nếu chọn Remov All MFs sẽ gỡ bỏ tất cả các hàm liên thuộc của biến đã chọn. Theo mặc định, số hàm liên thuộc là 3 có dạng tam giác, ta có thể thay đổi số lượng cũng như hình dạng hàm liên thuộc. Để thay đổi hình dạng một hàm liên thuộc nào đó, ta kích chuột vào hàm đó, nó sẽ chuyển sang mầu đỏ, sau đó kích chuột vào hộp thoại như chỉ ra ở hình 2.12 để chọn hàm liên thuộc mong muốn. Trên ô Range và Display Range ta có nhập các giá trị về miền xác định và miền hiển thị của biến ngôn ngữ, mặc định của các miền đó là từ 0 đến 1. Trên ô Name và ô Params (hình 2.12) ta có thể đặt tên và miền xác định cho từng tập mờ.



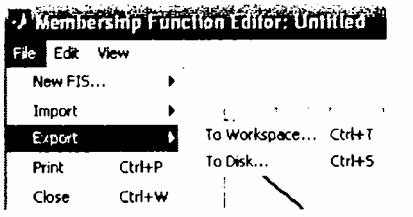
Hình 2.12



Hình 2.13. Cửa sổ soạn thảo luật

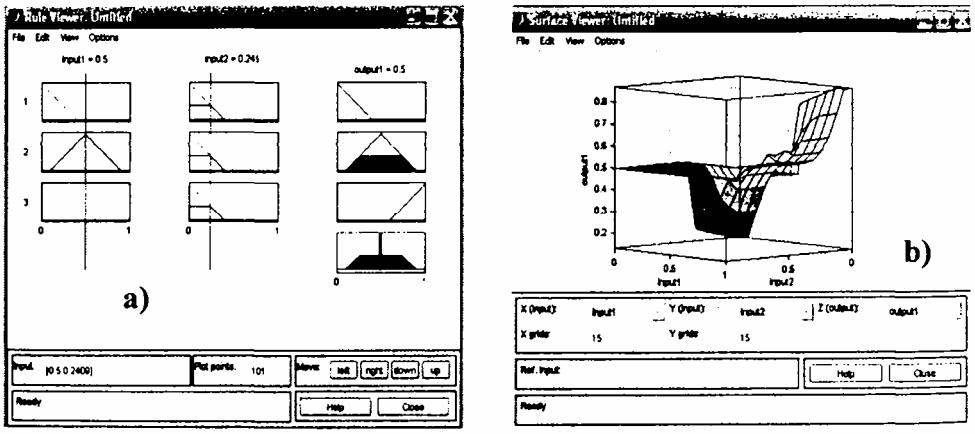
Để soạn thảo luật hợp thành, ta ấn **Edit, Rules** trên màn hình hiện ra cửa sổ hình 2.13. Sau mỗi lần soạn xong một luật ta ấn **Add rule** để xác nhận. Để thay đổi một luật hợp thành ta ấn **Change rule**. Để xoá một luật điều khiển ta ấn **Delete rules**. Muốn quan sát hoạt động của các luật ta ấn **View Rules**. Ấn **View Surface** để quan sát quan hệ vào – ra của bộ điều khiển (hình 2.14a, b).

Sau khi thiết kế xong bộ điều khiển, ta cần đặt tên và lưu chúng bằng cách ấn **File, Export To Disk** để cất vào đĩa hoặc to **Workspase** để lưu vào vùng làm việc của Matlab.



Muốn mở một bộ Điều khiển mờ đã lưu trên đĩa, ấn **File, Export To Disk** sau đó ấn **Import from disk**, chọn file cần mở.

Sau khi thiết kế xong bộ điều khiển mờ bằng cửa sổ Edit GUI, ta chuyển về cửa sổ mô phỏng SIMULINK, mở một file mới với đuôi '.mat', xây dựng mô hình mô phỏng cho hệ, tiến hành chạy mô phỏng và hiệu chỉnh hệ thống.



Hình 2. 14a.b. a) Quan sát hoạt động của các luật

b) Quan hệ vào-ra của bộ điều khiển

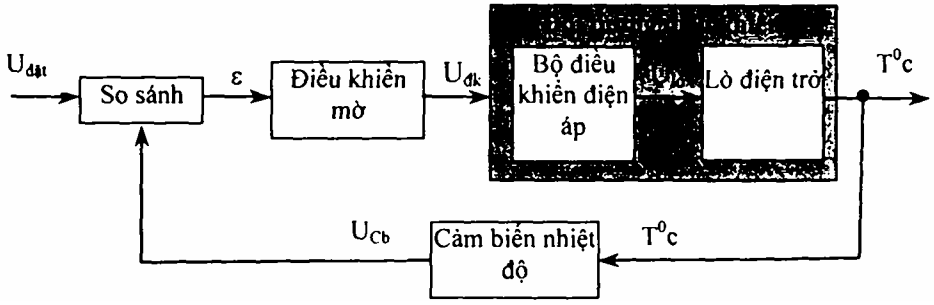
2.3.2. Ví dụ thiết kế hệ mờ

Để minh hoạ cho những vấn đề đã trình bày ở trên, sau đây chúng ta tiến hành phân tích, thiết kế bộ điều khiển mờ để điều khiển đối tượng nhiệt độ lò điện trở có hàm số truyền là:

$$W(s) = \frac{k}{Ts+1} e^{-\tau_1 S} = \frac{6,52}{150s+1} e^{-25S}. \qquad (2.7)$$

Biết điện áp cấp cho lò có giá trị định mức là 230 V.

Sơ đồ khối của hệ được chỉ ra trên hình 2.15.



Hình 2.15. Sơ đồ khối hệ điều khiển nhiệt độ lò điện trở

Bước 1: Tìm hiểu hệ thống

☞ Lò điện trở dùng để gia nhiệt chi tiết bằng kim loại cho các công đoạn như tôi, ram... Lò điện trở được nung nóng bằng dây điện trở, nguồn điện

cung cấp cho lò là nguồn áp có thể điều chỉnh được. Việc điều khiển nhiệt độ lò được thực hiện thông qua điều khiển điện áp cung cấp cho lò. Trong kỹ thuật điều khiển, người ta mô tả lò bằng một khâu quán tính bậc nhất có trễ có hàm số truyền:

$$W(s) = \frac{k}{Ts+1} e^{-\tau_1 s}$$

Trong đó, hằng số thời gian T và thời gian trễ T có giá tri tuỳ vào loại lò và công suất lò.

☞ Bộ Điều khiển điện áp có điện áp điều chỉnh được và biến thiên trong khoảng từ 100V: 230V, được mô tả gần đúng bằng một khâu có hàm số truyền:

w(s) = ke~2s với k = 23, ~ = O,05(s).

☞ Cảm biến nhiệt độ được coi là 1 khâu tỉ lệ với hệ số:

$$k = \frac{10V}{1500^0 C} = 0{,}0067\left(\frac{V}{^0C}\right).$$

☞ Điện áp đặt có giá trị lớn nhất là 10 V.

☞ Khâu so sánh làm nhiệm vụ so sánh điện áp đặt và điện áp phản hồi lấy từ đầu ra của khối cảm biến, đầu ra của khâu so sánh là sai lệch e = U – u_{cb}. Lò diện trở nói riêng, cũng như đối tượng nhiệt nói chung thường không cho phép có độ quá điều chỉnh, do đó e biến thiên trong khoảng từ 10 đến 0.

Bước 2: Chọn các biến ngôn ngữ vào, ra

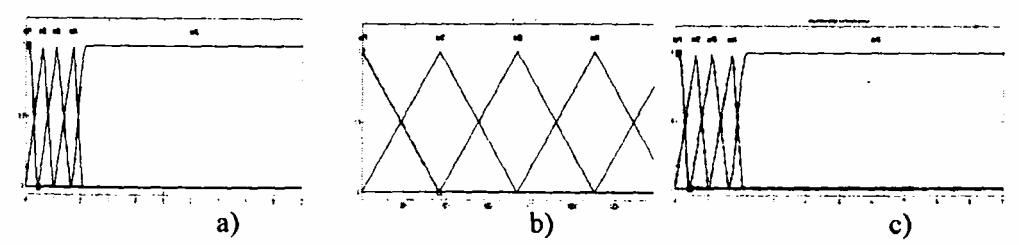
Giả thiết ta điều khiển lò điện trở theo quy luật PI, khi đó biến ngôn ngữ đầu vào bộ điều khiển mờ là sai lệch (ký hiệu là E) và tích phân sai lệch (ký hiệu là TE). Đầu ra bộ Điều khiển mờ là điện áp (ký hiệu là U). Miền giá trị của các biến ngôn ngữ được chọn như sau:

E = [0÷10]; TE = [0÷1500]; U = [0÷20]; hàm liên thuộc của các biến ngôn ngữ được chọn như hình 2.16a,b,c

$\underline{\mu}_E^T = [\mu_{E1}(x)\ \mu_{E2}(x)\ \mu_{E3}(x)\ \mu_{E4}(x)\ \mu_{E5}(x)]$ (hình 2.16a);

$\underline{\mu}_{TE}^T = [\mu_{TE1}(x)\ \mu_{TE2}(x)\ \mu_{TE3}(x)\ \mu_{TE4}(x)\ \mu_{TE5}(x)]$ (hình 2.16b);

$\underline{\mu}_U^T = [\mu_{U1}(x)\ \mu_{U2}(x)\ \mu_{U3}(x)\ \mu_{U4}(x)\ \mu_{U5}(x)]$ (hình 2.16a);



Hình 2.16a,b,c. Hình dạng các hàm liên thuộc đầu vào và đầu ra

Bước 3: Xây dựng luật hợp thành: Với 5 tập mờ của mỗi đầu vào, ta xây dựng được 5 x 5 = 25 luật điều khiển. Các luật điều khiển này được xây dựng theo 2 nguyên tắc sau:

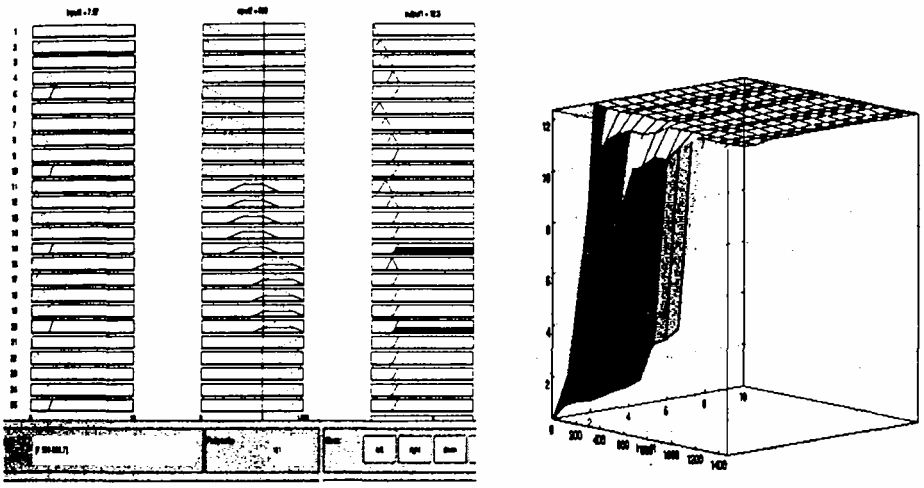
- Sai lệch càng lớn thì tác động điều khiển càng lớn.

- Tích phân sai lệch càng lớn thì tác động điều khiển càng lớn.

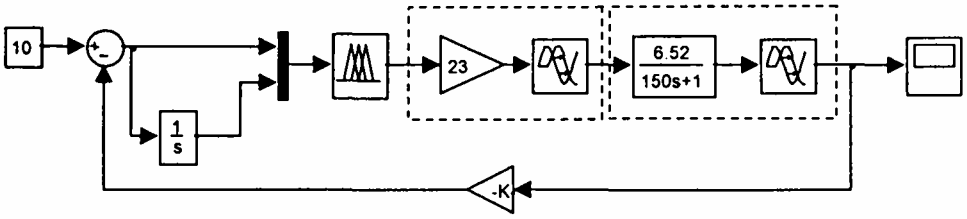
R_1:	Nếu	E	=	E_1	và	TE	=	TE_1	thì	U	=	U_1	hoặc
R_2:	Nếu	E	=	E_2	và	TE	=	TE_1	thì	U	=	U_2	hoặc
R_3:	Nếu	E	=	E_3	và	TE	=	TE_1	thì	U	=	U_3	hoặc
R_4:	Nếu	E	=	E_4	và	TE	=	TE_1	thì	U	=	U_4	hoặc
R_5:	Nếu	E	=	E_5	và	TE	=	TE_1	thì	U	=	U_5	hoặc
R_6:	Nếu	E	=	E_1	và	TE	=	TE_2	thì	U	=	U_2	hoặc
R_7:	Nếu	E	=	E_2	và	TE	=	TE_2	thì	U	=	U_3	hoặc
R_8:	Nếu	E	=	E_3	và	TE	=	TE_2	thì	U	=	U_4	hoặc
R_9:	Nếu	E	=	E_4	và	TE	=	TE_2	thì	U	=	U_5	hoặc
R_{10}:	Nếu	E	=	E_5	và	TE	=	TE_2	thì	U	=	U_5	hoặc
R_{11}:	Nếu	E	=	E_1	và	TE	=	TE_3	thì	U	=	U_3	hoặc
R_{12}:	Nếu	E	=	E_2	và	TE	=	TE_3	thì	U	=	U_4	hoặc
R_{13}:	Nếu	E	=	E_3	và	TE	=	TE_3	thì	U	=	U_5	hoặc
R_{14}:	Nếu	E	=	E_4	và	TE	=	TE_3	thì	U	=	U_5	hoặc
R_{15}:	Nếu	E	=	E_5	và	TE	=	TE_3	thì	U	=	U_5	hoặc
R_{16}:	Nếu	E	=	E_1	và	TE	=	TE_4	thì	U	=	U_4	hoặc
R_{17}:	Nếu	E	=	E_2	và	TE	=	TE_4	thì	U	=	U_5	hoặc
R_{18}:	Nếu	E	=	E_3	và	TE	=	TE_4	thì	U	=	U_5	hoặc
R_{19}:	Nếu	E	=	E_4	và	TE	=	TE_4	thì	U	=	U_5	hoặc
R_{20}:	Nếu	E	=	E_5	và	TE	=	TE_4	thì	U	=	U_5	hoặc
R_{21}:	Nếu	E	=	E_1	và	TE	=	TE_5	thì	U	=	U_5	hoặc
R_{22}:	Nếu	E	=	E_2	và	TE	=	TE_5	thì	U	=	U_5	hoặc
R_{23}:	Nếu	E	=	E_3	và	TE	=	TE_5	thì	U	=	U_5	hoặc
R_{24}:	Nếu	E	=	E_4	và	TE	=	TE_5	thì	U	=	U_5	hoặc
R_{25}:	Nếu	E	=	E_5	và	TE	=	TE_5	thì	U	=	U_5	

Bước 4: Chọn luật hợp thành Max-Min, giải mờ bằng phương pháp trọng tâm, ta quan sát được sự tác động của các luật và quan hệ vào - ra của bộ điều khiển như hình 2.17a,b.

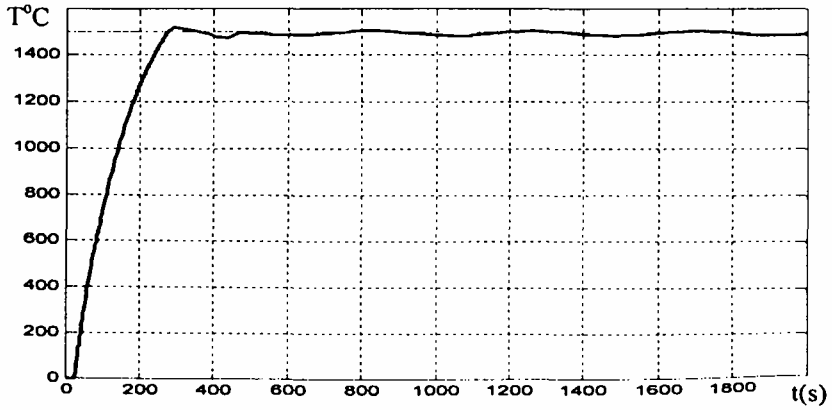
Bước 5: Mô phỏng hệ thống: Sơ đồ mô phỏng hệ thống được chỉ ra trên hình 2.18. Kết quả mô phỏng được chỉ ra trên hình 2.19.



H'nh 2.17a, b. Quan hệ vào - Ra của bộ Điều khiển

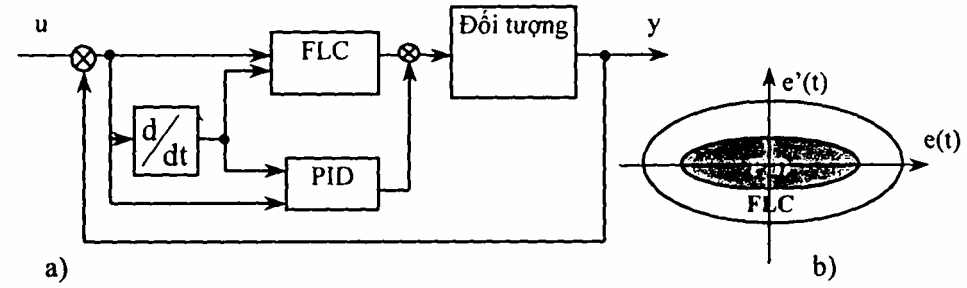


Hình 2.18. Sơ đồ mô phỏng hệ thống

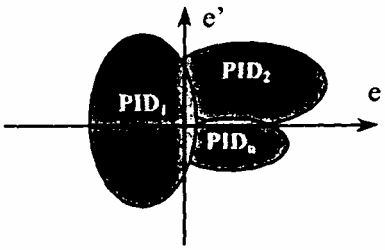


2.5. HỆ ĐIỀU KHIỂN MỜ LAI (F-PID)

Hệ mờ lai viết tắt là F-PID là hệ điều khiển trong đó thiết bị điều khiển gồm 2 thành phần: Thành phần điều khiển kinh điển và thành phần điều khiển mờ. Bộ Điều khiển F-PID có thể thiết lập dựa trên hai tín hiệu là sai lệch e(t) và đạo hàm của nó e'(t). Bộ Điều khiển mờ có đặc tính rất tốt ở vùng sai lệch lớn, ở đó với đặc tính phi tuyến của nó có thể tạo ra phản ứng động rất nhanh. Khi quá trình của hệ tiến gần đến điểm đặt (sai lệch e(t) và đạo hàm của nó e'(t) xấp xỉ bằng 0) vai trò của bộ điều khiển mờ (FLC) bị hạn chế nên bộ điều khiển sẽ làm việc như một bộ điều chỉnh PID bình thường. Trên hình 2.20 thể hiện ý tưởng thiết lập bộ điều khiển mờ lai F-PID và phân vùng tác động của chúng.



Hình 2.20. a) Nguyên lý điều khiển mờ lai;
b) Vùng tác động của các bộ điều khiển



Hình 2.21. Vùng tác động của các bộ điều khiển

Sự chuyển đổi giữa các vùng tác động của FLC và PID có thể thực hiện nhờ khoá mờ hoặc dùng chính FLC. Nếu sự chuyển đổi dùng FLC thì ngoài nhiệm vụ là bộ điều chỉnh FLC còn làm nhiệm vụ giám sát hành vi của hệ thống để thực hiện sự chuyển đổi. Việc chuyển đổi tác động giữa FLC và PID có thể thực hiện nhờ luật đơn giản sau:

if |e(t) dương lớn và $|\dot{e}(t)|$ dương lớn thì u là FLC **(2.8)**

if |e(t) dương nhỏ và $|\dot{e}(t)|$ dương nhỏ thì u là PID **(2.9)**

Để thực hiện chuyển đổi mờ giữa các mức FLC và bộ chuyển đổi PID, ta có thể thiết lập nhiều bộ điều chỉnh PIDi (i = 1,2... n) mà mỗi bộ được chọn để tối ưu chất lượng theo một nghĩa nào đó đề tạo ra đặc tính tốt trong 1 vùng giới hạn của biến vào (hình 2.21). Các bộ điều chỉnh này có chung thông tin ở đầu vào và sự tác động của chúng phụ thuộc vào giá trị đầu vào. Trong trường hợp này, luật chuyển đổi có thể viết theo hệ mờ như sau:

Nếu (trạng thái của hệ) là Ei thư (tín hiệu điều khiển) = u_i

Trong đó i = 1, 2,..., n; Ei là biến ngôn ngữ của tín hiệu vào, u_i là các hàm với các tham số của tác động điều khiển. Nếu tại mỗi vùng điều chỉnh, tác động điều khiển là do bộ điều chỉnh PIDi với:

$$u_i = K_{Pi}e + K_{Ii}\int_0^t e(t)dt + K_{Di}\frac{de}{dt} \quad (i = 1, 2, \dots n). \qquad (2.10)$$

Như vậy, các hệ số của bộ điều chỉnh PIDi mới phụ thuộc các tín hiệu đầu vào tổng quát hơn là phụ thuộc vào trạng thái của hệ. Nếu coi các hệ số K_{pi}, K_{Di} Và K_{Ii} chính là kết quả giải mờ theo phương pháp trung bình trọng tâm từ ba hệ mờ hàm:

☞ Hệ mờ hàm tính hệ số K_p với hệ luật:

Ru(i): **if** E **is** E_i **and** DE **is** DE_i **then** $K_p = K_{pi}$. (2. 11)

☞ Hệ mờ hàm tính hệ số K_D với hệ luật:

Ru(i): **if** E **is** E_i **and** DE **is** DE_i **then** $K_D = K_{Di}$. (2. 12)

☞ Hệ mờ hàm tính hệ số K_I với hệ luật:

Ru(i): **if** E **is** E_i **and** DE **is** DE_i **then** $K_I = K_{Ii}$. (2. 13)

2.6. HỆ ĐIỀU KHIỂN THÍCH NGHI MỜ

2.6.1. Khái niệm

a/ Định nghĩa: Hệ điều khiển thích nghi mờ là hệ điều khiển thích nghi được xây dựng trên cơ sở của hệ mờ

So với hệ điều khiển thích nghi kinh điển, hệ điều khiển thích nghi mờ có miền tham số chỉnh định rất lớn. Bên cạnh các tham số K_p, T_I, T_D giống như bộ điều khiển PID thông thường, ở bộ điều khiển mờ ta còn có thể chỉnh định các tham số khác như hàm liên thuộc, các luật hợp thành, các phép toán OR, AND, NOT, nguyên lý giải mờ v.v...

Trong thực tế, hệ điều khiển thích nghi được sử dụng ngày càng nhiều vì nó có các ưu điểm nổi bật so với hệ thông thường. Với khả năng tự chỉnh định lại các tham số của bộ điều chỉnh cho phù hợp với đối tượng chưa biết rõ đã đưa hệ thích nghi mờ trở thành một hệ điều khiển thông minh.

b/ Phân loại

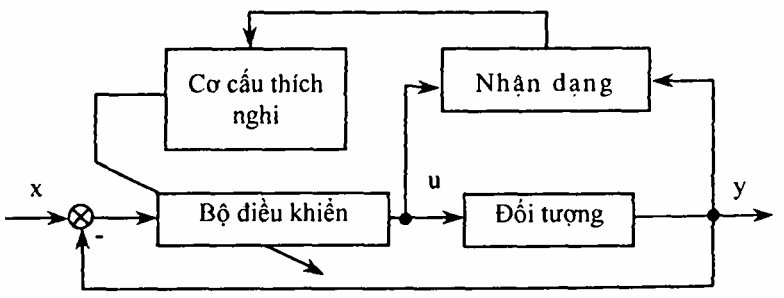
Một cách tông quát, hệ điều khiển thích nghi mờ có thể phân thành 2 loại:

- Bộ Điều khiển mờ tự chỉnh là bộ điều khiển mờ có khả năng chỉnh định các tham số của các tập mờ (các hàm liên thuộc);

- Bộ điều khiển mờ tự thay đổi cấu trúc là bộ điều khiển mờ có khả năng chỉnh định lại các luật điều khiển. Đối với loại này hệ thống có thể bắt đầu làm việc với một vài luật điều khiển đã được chỉnh định trước hoặc chưa đủ các luật.

c/ Các phương pháp điều khiển thích nghi mờ

Các bộ điều khiển thích nghi rõ và mờ đều có mạch vòng thích nghi được xây dựng trên cơ sở của 2 phương pháp:



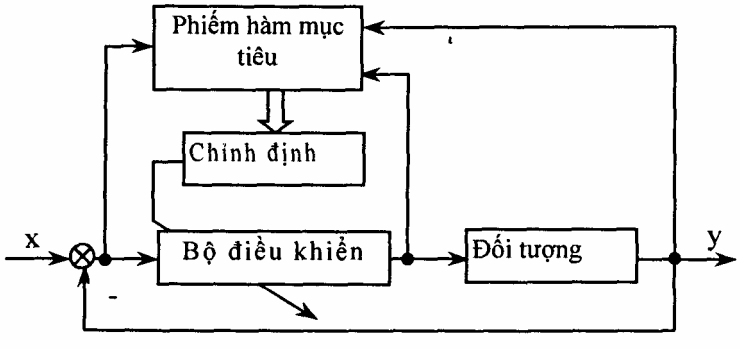
Hình 2.22. Cấu trúc phương pháp điều khiển thích nghi trực tiếp

☞ Phương pháp trực tiếp (hình 2.22) thực hiện thông qua việc nhận dạng thường xuyên các tham số của đối tượng trong hệ kín. Quá trình nhận dạng

thông số của đối tượng có thể thực hiện bằng cách thường xuyên đo trạng thái của các tín hiệu vào/ra của đối tượng và chọn 1 thuật toán nhận dạng hợp lý, trên cơ sở mô hình đối tượng đã biết trước hoặc mô hình mờ;

☞ Phương pháp gián tiếp (hình 2.23) thực hiện thông qua phiếm hàm mục tiêu của hệ kín xây dựng trên các chỉ tiêu chất lượng.

Phiếm hàm mục tiêu có thể được xây dựng trên cơ sở các chỉ tiêu chất lượng động của hệ thống như độ quá điều chỉnh, thời gian quá độ hay các chỉ tiêu tích phân sai lệch... Bộ điều khiển thích nghi mờ có thể chia thành 2 loại:



Hình 2.23. Cấu trúc phương pháp điều khiển thích nghi gián tiếp

2.6.2. Tổng hợp bộ điều khiển thích nghi mờ ổn định

a. Cơ sở lý thuyết

Xét 1 hệ phi tuyến SISO được mô tả bởi phương trình:

$y^{(n)} = f(y, y', \ldots, y^{(n-1)}) + bu$; $y = x$ là biến trạng thái.

$$y_{(n)} = f(y) + bu \qquad (2.14)$$

Trong đó u là đầu vào, y là đầu ra, hàm phi tuyến f(.) và hằng số b được giả thiết chưa biết, $y = [y, y', \ldots y^{(n-1)}]^T$. Mục tiêu là thiết kế bộ điều khiển mờ để tạo ra tín hiệu điều khiển u sao cho tín hiệu ra y(t) của hệ thống bám theo quĩ đạo y_d cho trước nào đó.

Nếu biết trước f(y) và b, ta có thể tổng hợp được bộ điều khiển theo các phương pháp kinh điển [9], [55], bộ điều khiển đó có tín hiệu đầu ra là:

$$u(t) = \frac{1}{b}\left[- f(\underline{y}) + \frac{d^n y_d}{dt^n} + K^T E \right] \qquad (2.15)$$

trong đó: $$K^T = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix}, \quad E = \begin{bmatrix} e \\ \frac{de}{dt} \\ \vdots \\ \frac{d^{n-1}e}{dt^{n-1}} \end{bmatrix}. \qquad (2.16)$$

Các hệ số $k_1, k_2, \ldots k_n$ được chọn sao cho tất cả các nghiệm của phương trình: $p^n + k_n p^{n-1} + \ldots + k_1 = 0$ nằm ở nửa trái mặt phẳng phức. Tức là các nghiệm p_k có phần thực âm:

$$Re(pk) < 0. \qquad (2.17)$$

Thay (2.15) vào (2.14) ta có:

$$\frac{d^n e}{dt^n} + k_n \frac{d^{n-1} e}{dt^{n-1}} + \ldots + k_1 e = 0. \qquad (2.18)$$

Do có điều kiện (2.17) nên nghiệm của e(t) chắc chắn thoả mãn điều kiện:

$$\lim_{t \to \infty} e(t) = 0.$$

Ta thấy rằng bài toán tổng hợp trên chỉ có ý nghĩa khi đã biết chính xác mô hình toán học của hệ thống, hay nói cách khác là trong (2.1) ta đã biết f(y) và b. Điều này không phù hợp với nhiều bài toán thực tế. Vì vậy mục tiêu điều khiển đề ra là phải xác định bộ điều khiển mờ $u = u(\underline{x}, \underline{\theta})$ và luật Điều khiển véctơ tham số $\underline{\theta}$ sao cho thoả mãn các điều kiện sau:

- Hệ kín phải ổn định toàn cục trong phạm vi của các biến $\underline{y}(t)$, $\underline{\theta}(t)$) và $u(x, \underline{\theta})$.

Tức là: $|\underline{x}(t)| \le M_x < \infty$; $|\underline{\theta}(t)| \le M_0 < \infty$; $|u(x, \underline{\theta})| \le M_u < \infty$ với mọi $t \ge 0$. Trong đó Mx, M_0, Mu là các tham số do người thiết kế đặt ra.

- Độ sai lệch $e = y_d - y$ càng nhỏ càng tốt.

Khi f(.) và b đã biết thì ta dễ dàng tổng hợp được bộ điều khiển:

$$u^* = \frac{1}{b}\left[-f(\underline{y}) + \frac{d^n y_d}{dt^n} + K^T \underline{e}\right] \qquad (2.19)$$

Trong đó, u* được coi là tối ưu. Nhưng vì f(.) và b chưa biết nên u* không thể thực hiện được, ta sẽ thiết kế bộ điều khiển mờ để xấp xỉ hoá điều khiển tối ưu này.

Giả thiết bộ điều khiển u là tổ hợp 2 bộ điều khiển: Bộ điều khiển mờ $u_f(\underline{x}, \underline{\theta})$ và bộ điều khiển giám sát $u_s(\underline{x})$:

$$u = u_f + u_s \qquad (2.20)$$

Trong đó $u_f(\underline{x}, \underline{\theta})$ là bộ điều khiển mờ được đề cập trong tổng kết 2.1.

Tổng kết 2.l: Xét một hệ logic mờ MISO có n đầu vào $\underline{x}$ và 1 đầu ra y ($\underline{x} = (x_1, x_2, \ldots, x_n)^T \in Rn$ và $y \in R$). Định nghĩa Nj tập mờ $A_{i_j}^j$ với các hàm liên thuộc $\mu_{A_{i_j}^j}$ bao phủ miền xác định của các biến ngôn ngữ đầu vào (j = 1,…, n là số đầu vào). Luật điều khiển $R_u^{i_1 \ldots i_n}$ có dạng:

if $e_1 = A_{i_1}^1$ and $e_2 = A_{i_2}^2$ and…and = $A_{i_n}^n$ then $u = B_{i_1 \ldots i_n}$ (2.21)

trong đó $i_1 = 1, 2..., N_1;... i_n = 1, 2,..., Nn$ là số hàm liên thuộc cho mỗi biến đầu vào, $B_{i_1 \ldots i_n}$ là tập mờ đầu ra.

Sử dụng luật hợp thành PROD, mờ hoá theo đường singleton và giải mờ bằng phương pháp trung bình trọng tâm, ta thu được bộ điều khiển mờ:

$$u = u(\underline{x}, \underline{\theta}) = \frac{\sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} y_{i_1 \cdots i_n} \left[\prod_{j=1}^{n} \mu_{A_{i_j}^j}(x_j)\right]}{\sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} \left[\prod_{j=1}^{n} \mu_{A_{i_j}^j}(x_j)\right]}, \qquad (2.22)$$

$$u = \theta^T \xi(\underline{x}) \qquad (2.23)$$

trong đó $\zeta(\underline{x})$ là véctơ hàm mờ cơ sở.

$$\xi(\underline{x}) = \frac{\prod_{j=1}^{n} \mu_{A_{i_j}^j}(x_j)}{\sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} \left[\prod_{j=1}^{n} \mu_{A_{i_j}^j}(x_j)\right]}. \qquad (2.24)$$

Thay *(2.20)* vào (2.14) ta được:

$$y^{(n)} = f(\underline{x}) + b[uf(\underline{x}, \underline{\theta}) + us(\underline{x})]. \qquad (2.25)$$

Từ (3.29) ta rút ra: $f(\underline{x}) = -bu^* + \frac{d^n y_d}{dt^n} + K^T\underline{e}$ thay vào (3.35)

$y^{(n)} = -bu^* + yd^{(n)} + KTe + b[uf(\underline{x}, \underline{\theta}) + uS(\underline{x})]$. Sau khi biến đổi ta được:

$$e^{(n)} = -KTe + b\,[u^* - uf(\underline{x}, \underline{\theta}) - us(\underline{x})]. \qquad (2.26)$$

Hoặc viết dưới dạng phương trình trạng thái:

$$\dot{\underline{e}} = A\underline{e} + B[u^* - u_f(\underline{x}, \underline{\theta}) - u_s(\underline{x})] \qquad (2.27)$$

Trong đó:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ & & & & \dots & \\ -k_n & -k_{n-1} & -k_{n-2} & & \dots & -k_1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \dots \\ b \end{bmatrix}. \qquad (2.28)$$

Chọn hàm Lyapunov $V = \frac{1}{2}\,\underline{e}^T p\underline{e}$. Trong đó P là ma trận dương đối xứng được xác định từ phương trình Lyapunov:

$$ATP + PA = -Q \ (Q > 0). \qquad (2.29)$$

Đạo hàm V ta được:

$$\dot{V} = \frac{1}{2}\dot{e}^T Pe + \frac{1}{2}e^T P\dot{e}. \qquad (2.30)$$

Thay (2.27), (2.29) vào (2.30) ta được:

$$\begin{aligned} \dot{V} &= -\frac{1}{2}\underline{e}^T Q\underline{e} + \underline{e}^T PB[u^* - u_f(\underline{x}, \underline{\theta}) - u_s(x)] \\ &\le -\frac{1}{2}\underline{e}^T Q\underline{e} + \left|\underline{e}^T PB\right|(|u^*| + |u_f|) - \underline{e}^T PBu_s \end{aligned} \qquad (2.31)$$

ta cần phải tìm hàm u_s sao cho $V \le 0$.

Giả thiết ta xác định được hàm $f^u(\underline{x})$ và hằng số b_L thoả mãn điều kiện: $|f(\underline{x})| \le f^u(\underline{x})$ và $0 < b_L < b$ thì hàm điều khiển giám sát $u_s(x)$ được xây dựng

như sau:

$$u_s(\underline{x}) = I_1^* \, \mathrm{sgn}(\underline{e}^T PB\left[|u_f| + \frac{1}{b_L}\left(f^U + |y_d^{(n)}| + |K^T \underline{e}|\right)\right]$$

Trong đó:

$$I_1^* = \begin{cases} 1 & \text{Khi } V > \overline{V} \\ 0 & \text{khi } V \le \overline{V} \end{cases}$$

($\overline{V}$ là 1 hằng số được chọn bởi người thiết kế).

Vì b > 0, sugn($\underline{e}^T \underline{PB}$) có thể xác định, hơn nữa tất cả các thành phần trong (2.32) có thể xác định được, vì vậy bộ điều khiển giám sát u_s là hoàn toàn xác định. Thay (2.32) và (2.19) vào (2.31) và xét cho trường hợp $I_1^* = 1$ ta có:

$$\begin{aligned} \dot{V} &\le -\frac{1}{2}\underline{e}^T Q \underline{e} + |\underline{e}^T PB|\left[\frac{1}{b}\left(|f| + |y_d^{(n)}| + |K^T \underline{e}|\right) + |u_f| - |u_f|\right. \\ &\quad \left. - \frac{1}{b_L}\left(f^U + |y_d^{(n)}| + |K^T \underline{e}|\right)\right] \\ &\le -\frac{1}{2}\underline{e}^T Q \le 0 \end{aligned}$$

vậy sử dụng u_s theo (2.32) ta luôn nhận được $V \le \overline{V}$.

Từ (2.32) ta thấy rằng u_s chỉ xuất hiện khi không thoả mãn điều kiện: $V \le \overline{V}$.

Do vậy trong khoảng sai số nhỏ (nghĩa là $V \le \overline{V}$) thì chỉ có bộ điều khiển mờ u_f làm việc còn bộ điều khiển giám sát không làm việc ($u_s = 0$). Khi hệ thống có khuynh hướng mất ổn định ($V > \overline{V}$) thì bộ điều khiển giám sát bắt đầu làm việc để hướng cho $V \le \overline{V}$.

Nếu chọn $I_1^* \equiv 1$ thì tử (2.33) ta cần phải đảm bảo không chỉ giới hạn của véctơ trạng thái mà còn phải đảm bảo cho e hội tụ về 0. Ta không chọn phương án này vì u_s thường rất lớn.

Thật vậy, từ (2.33) ta thấy u_s tỉ lệ với giới hạn trên của f_u mà giới hạn này thường rất lớn. Tín hiệu điều khiển lớn có thể gây phiền phức do có làm tăng

thêm chi phí phụ. Bởi vậy ta chọn u_s làm việc theo kiểu giám sát.

Để tìm luật điều khiển thích nghi véctơ tham số $\underline{\theta}$ ta thay $u_f(\underline{x}, \underline{\theta}) = \underline{\theta}\,\zeta(\underline{x})$. Đặt $\underline{\theta}^*$ là véctơ tham số tối ưu:

$$\underline{\theta}^* = \arg\min_{|\underline{\theta}|\le M_\theta}\left[\sup_{|\underline{x}|\le M_x}\left|u_c(x,\theta) - u^*\right|\right]$$

và đặt $w = uf(\underline{x}, \underline{\theta}^*) - u^*$ biểu thức (2.27) có thể viết:

$$\begin{aligned}\dot{\underline{e}} &= A\underline{e} + B\left[u^* - u_f(\underline{x},\underline{\theta}) - u_s(\underline{x})\right]\\ &= A\underline{e} + B[u_f(\underline{x},\underline{\theta}^*) - u_f(\underline{x},\underline{\theta})] - Bu_s(\underline{x}) - Bw\\ &= A\underline{e} + B\underline{\varphi}^T\underline{\xi}(\underline{x}) - Bu_s - Bw\end{aligned} \qquad (2.34)$$

trong đó $\underline{\varphi} = \underline{\theta}^* - \underline{\theta}$; $\xi(\underline{x})$ là hàm mờ cơ sở.

Chọn hàm Lyapunov dạng:

$$V = \frac{1}{2}\underline{e}^T P\underline{e} + \frac{b}{2\gamma}\underline{\phi}^T\underline{\phi}. \qquad (2.35)$$

Với γ là một hằng số dương, ta có:

$$\dot{V} = -\frac{1}{2}\underline{e}^T Q\underline{e} + \underline{e}^T PB\left[\underline{\varphi}^T\underline{\xi}(\underline{x}) - u_s - w\right] + \frac{b}{\gamma}\underline{\varphi}^T\dot{\underline{\varphi}}. \qquad (2.36)$$

Gọi Pn là cột cuối cùng của ma trận P, từ (2.28) ta có:

$$\underline{e}^T PB = \underline{e}^T P_n b. \qquad (2.37)$$

eTpB = To b. (2.37) Thay (2.37) vào(2.36) ta được:

$$\dot{V} = -\frac{1}{2}\underline{e}^T Q\underline{e} + \frac{b}{\gamma}\underline{\varphi}^T\left[\gamma\underline{e}^T P_n\underline{\xi}(\underline{x}) + \dot{\underline{\varphi}}\right] - \underline{e}^T PBu_s - \underline{e}^T PBw. \qquad (2.38)$$

Chọn luật thích nghi:

$$\dot{\underline{\theta}} = \gamma \underline{e}^T P_n \xi(\underline{x}) \qquad (2.39)$$

thì (2.38) trở thành:

$$\dot{V} \le -\frac{1}{2}\underline{e}^T Q \underline{e} - \underline{e}^T PBw \qquad (2.40)$$

trong đó: $\underline{e}^T PBu_s \ge 0$.

trong đó: $\underline{e}^T PBu_s \ge 0$

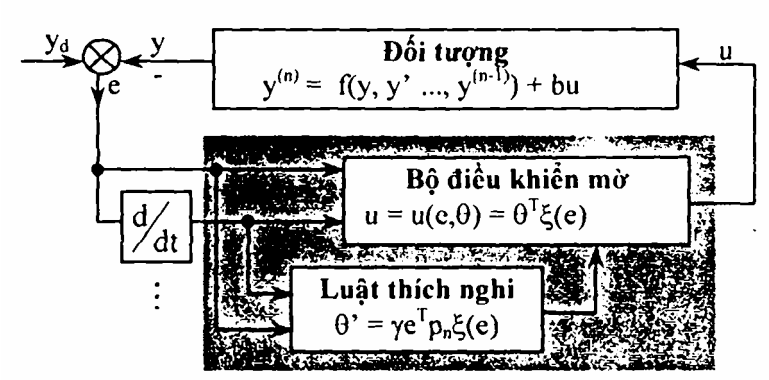
Đây là điều tốt nhất ta có thể đạt được.

b) Thuật toán tổng hợp bộ điều khiển mờ thích nghi

Để tổng hợp bộ điều khiển mờ thích nghi, ta có thể tiến hành theo 2 bước: Bước 1 là chọn cấu trúc của bộ điều khiển mờ, bước 2 là xác định thích nghi các véctơ tham số.

+ **Chọn cấu trúc của bộ điều khiển mờ**

Câu trúc của bộ điều khiển mờ thích nghi như hình 2.24. trong đó đối tượng điều khiển là 1 hệ phi tuyến bất kỳ được mô tả tổng quát bằng biểu thức (2.1). Bộ điều khiển mờ thích nghi có thể có nhiều đầu vào gồm sai lệch và các đạo hàm của chúng. Mục đích của việc thiết kế bộ điều khiển mờ là tạo ra tín hiệu điều khiển u, sao cho quĩ đạo đầu ra của đối tượng (y) bám theo quĩ đạo cho trước (y_d), cho dù có sự thay đổi thông số và cấu trúc của đối tượng.



Hình 2.24: Sơ đồ cấu trúc bộ điều khiển mờ thích nghi

+ Các bước thực hiện thuật toán

Trong trường hợp tổng quát, bộ điều khiển mờ có n đầu vào, thuật toán tông hợp được tóm tắt theo các bước sau:

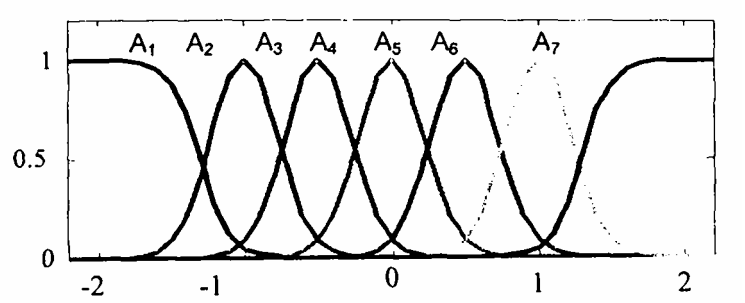
- **Bước 1.** Xác định hàm liên thuộc của các biến ngôn ngữ đầu vào.

Đặt $E = (e_1, e_2, \dots, e_n)^T = (y_d - y, \dot{y}_d - \dot{y} \cdots y_d^{(n-1)} - y^{(n-1)})^T$.

Định nghĩa miền xác định của các thành phần e_j là:

$\left[\alpha^j_{min}, \alpha^j_{max}\right]$ (j = 1, 2,..., n là số đầu vào).

Chú ý rằng, giá tri thức của e_j có thể ở bên ngoài khoảng $\left[\alpha\frac{j}{min}, \frac{j}{max}\right]$ đã chọn, ở đây $\left[\alpha\frac{j}{min}, \frac{j}{max}\right]$ là khoảng mà e_j rơi vào nhiều nhất.



Hình 2.25. Hàm liên thuộc với 7 tập mờ

Định nghĩa Nj tập mờ A_1j... A_nJ trên miền $\left[\alpha\frac{j}{min}, \frac{j}{max}\right]$, hàm liên thuộc của các tập mờ có thể chọn là hình tam giác, hình thang, hàm Gaus, hàm sigmoid v.v... Chọn hàm liên thuộc kiểu hình tam giác và hình thang có ưu điểm là đơn giản, song có nhược điểm là độ điều chỉnh không trơn. Hình 2.25 là ví dụ về hàm liên thuộc kiểu Gaus ở giữa và kiểu sigmoid ở 2 bên đối với 1 biến ngôn ngữ đầu vào.

$$\mu_{A_l^j}(e_j) = \mu\left(e_j; \delta_l^j, \alpha_l^j\right) = 1 - \frac{1}{1 + e^{-\delta_l^j\left(e_j + \alpha_l^j\right)}} \qquad (2.41)$$

$$\mu_{A_p^j}(e_j) = \mu\left(e_j; \delta_p^j, \alpha_p^j\right) = e^{-\delta_p^j\left(e_j - \alpha_p^j\right)^2} \qquad (2.42)$$

với p = 2, 3 ... , Nj -1 , còn

$$\mu_{A_N^j}(e_j) = \mu\left(e_j; \delta_{N_j}^j, \alpha_{N_j}^j\right) = \frac{1}{1 + e^{-\delta_{N_j}^j\left(e_j - \alpha_{N_j}^j\right)}} \qquad (2.43)$$

trong đó: $\alpha_{min}^j = \alpha_1^j < \alpha_2^j < \cdots < \alpha_{N-1}^j < \alpha_N^j = \alpha_{max}^j$.

- Bước 2. Xây dựng bộ điều khiển mờ u từ tích N_1... N_n luật sau đây: Luật $Ru^{i_1 \ldots i_n}$

if $e_1 = A_{i_1}^1$ and $e_2 = A_{i_2}^2$ and…and $e_n = A_{i_n}^n$ then $u = B_{i_1 \ldots i_n}$ (2.44)

Trong đó $i_1 = 1, 2..., N_1$;... $i_n = 1, 2,..., N_n$ là số hàm liên thuộc cho mỗi biến đầu vào

$B_{i_1 \ldots i_n}$ là tập mờ đầu ra sẽ được xác đinh.

Việc thiết kế bộ điều khiển mờ bây giờ chuyển sang việc xác định các thông số $B_{i_1 \ldots i_n}$

Sử dụng luật hợp thành PROD, mờ hoá theo đường singleton và giải mờ bằng phương pháp trung bình trọng tâm ta thu được bộ điều khiển mờ:

$$u = u(e, \theta) = \frac{\sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} y_{i_1 \cdots i_n} \left[\prod_{j=1}^{n} \mu_{A_{i_j}^j}(e_j)\right]}{\sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} \left[\prod_{j=1}^{n} \mu_{A_{i_j}^j}(e_j)\right]} \qquad (2.45)$$

$$u = \theta^T \xi(e) \qquad (2.46)$$

trong đó: ζ(e) là tập hợp hàm mờ cơ sở đã biết.

$$\xi(e) = \frac{\prod_{j=1}^{n} \mu_{A_{i_j}^j}(e_j)}{\sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} \left[\prod_{j=1}^{n} \mu_{A_{i_j}^j}(e_j)\right]} \qquad (2.47)$$

lưu đồ thuật toán tông hợp hàm mờ cơ sở xe) như hình 2.26.

$y_{i_1 \ldots i_n}$ là điểm trọng tâm của $B_{i_1 \ldots i_n}$ chúng sẽ được chỉnh định theo luật thích nghi cho phù hợp với đối tượng.

$\underline{\theta}$ là một véctơ gồm tập hợp các $y_{i_1 \cdots i_n}$ với $i_1 = 1 \ldots N_1; \ldots i_n = 1 \ldots N_n$

Đặt: $$\underline{\theta} = [y_{1\cdots1}, y_{1\cdots2}, \cdots y_{2\cdots1}, \cdots y_{N_1\cdots1}, \cdots y_{N_1\cdots N_n}]. \quad (2.48)$$

Các thông số e được chỉnh định nhờ sử dụng luật thích nghi sau:

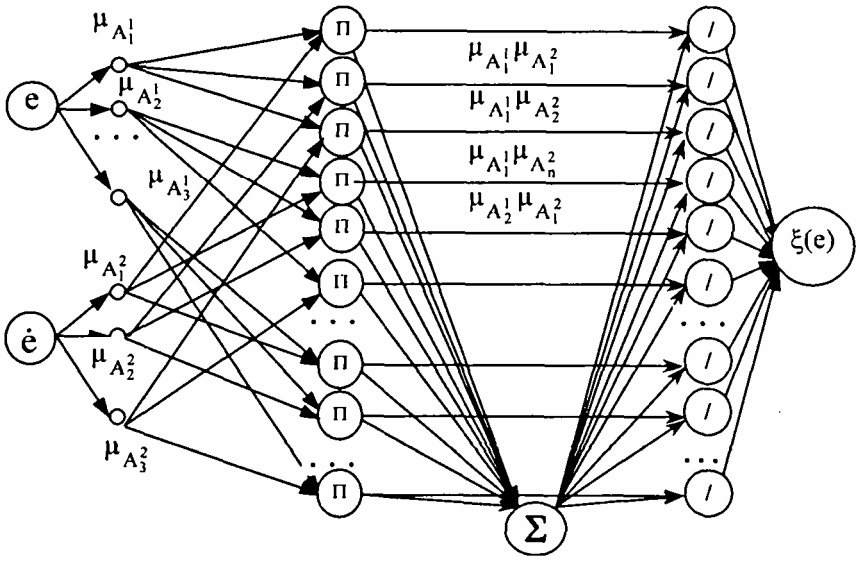
$$\dot{\underline{\theta}} = \gamma e^T P_n \xi(e) \quad (2.49)$$

Trong đó γ là 1 hằng số dương xác đinh tốc độ của thuật toán còn p_n là cột cuối cùng của ma trận P, với P là nghiệm của phương trình Lyapunov.

$$A^T P + pa = -Q \quad (2.50)$$

trong đó Q là ma trận dương xác định tuỳ ý, A là ma trận (n x n)

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ & & & & \ldots & \\ -k_n & -k_{n-1} & -k_{n-2} & & \ldots & -k_1 \end{bmatrix}. \quad (2.51)$$



Hình 2.26. Lưu đồ thuật toán tổng hợp hàm mờ cơ sở $\zeta(e)$

với các hằng số k_1, k_2… được chọn sao cho tất cả các nghiệm của phương trình: $Pn + knPn^{-1} + \ldots + k_1 = 0$ nằm bên nửa trái mặt phẳng phức. Với cách tông hợp như vậy hệ thống chắc chắn thoả mãn điều kiện $\lim_{t\to\infty} e(t) = 0$.

Từ các tập mờ đầu vào (2.41)... (2.43) và các thông số γ. Pn được xác định ở trên ta tiến hành xây dựng bộ điều khiển mờ theo trình tự sau:

- Định nghĩa các hàm liên thuộc (2.41)... (2.43).

- Xây dựng hàm mờ cơ sơ sở (2.47).

- Xác đinh luật thích nghi $\dot{\underline{\theta}} = \gamma e^T P_n \xi(e)$

- Xây dựng bộ điều khiển (2.46).

Chú ý:

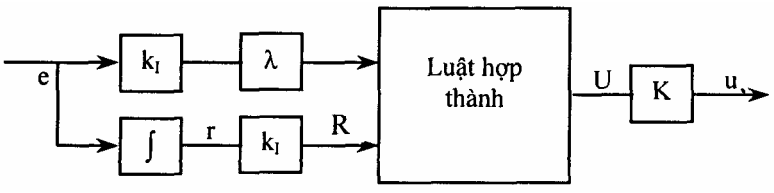
- Hệ số y trong (2.49) nói lên tốc độ hội tụ của thuật toán thích nghi. Nó được chọn và sau đó được kiểm nghiệm thông qua mô phỏng, nếu y chọn quá như thuật toán thích nghi hội tụ chậm, y chọn lớn, quá trình hội tụ nhanh nhưng nếu y chọn quá lớn hệ thống sẽ mất ổn định.

- Các giá trị P_1, P_2 được Xác định từ phương trình Lyapunov (2.40), Tuy nhiên độ lớn của nó cũng ảnh hưởng đáng kể đến chất lượng của hệ thống. Vì vậy sau khi thiết kế xong cần chỉnh định lại các giá trị của chúng sao cho đảm bảo chất lượng tốt trong toàn dải thay đổi của các thông số của đối tượng.

2.7. TỔNG HỢP BỘ ĐIỀU KHIỂN MỜ THÍCH NGHI TRÊN CƠ SỞ LÝ THUYẾT THÍCH NGHI KINH ĐIỂN

2.7.1. Đặt vấn đề

Một cấu trúc thông dụng nhất của hệ điều khiển logic mờ (FLC - Fuzzy Logic Control) là cấu trúc kiểu phản hồi sai lệch. Sơ đồ như hình 2.27. Trong đó k_1, λ là các hệ số khuếch đại đầu vào, K là hệ số khuếch đại đầu ra. Thực tiễn cho thấy việc chỉnh định FLC khó khăn hơn nhiều so với chỉnh định bộ điều khiển kinh điển, một trong những lý do chính là tính mềm dẻo của vùng nhận biết cơ bản của bộ điều khiển mờ và sự móc nối các thông số của chúng. Tuy nhiên không có một cách hệ thống hoá nào để đưa ra tất cả những thông số này.



Hình 2.27. Cấu trúc cơ bản của bộ điều khiển mờ 2 đầu vào

Hiện nay trong công nghiệp các bộ điều khiển logic mờ (FLC) thường được thiết kế theo kinh nghiệm và sự hiểu biết định tính đối tượng của các chuyên gia. Việc chỉnh định FLC được thực hiện thông qua chỉnh định các hàm liên thuộc đầu vào và đầu ra và mang nhiều tính chất "mò mẫm". Do đó không phù hợp cho việc chuẩn hoá chất lượng và khó trở thành một phương pháp luận có hệ thống. Trong mục này chúng ta sẽ tiếp cận kiểu thiết kế hỗn hợp theo hướng kết hợp cả hai cách tiếp cận định tính và tiếp cận định lượng. Đầu tiên ta xây dựng mô hình cơ bản của bộ điều khiển mờ bao gồm các hàm liên thuộc và các luật hợp thành, chúng có thể tạo ra một đáp ứng hợp lý ở một mức độ nào đó. Luật hợp thành cơ bản được chọn là một luật hợp thành tuyến tính, còn hàm liên thuộc có thể được xác định theo hình tam giác, hình thang hoặc hàm Gaus. Sau khi xác định được hàm liên thuộc và luật hợp thành cơ bản, ta sử dụng chúng để tìm ra hệ số khuếch đại tỷ lệ. Có thể sử dụng nhiều phương pháp định lượng khác nhau, việc xác định các hệ số khuếch đại tỷ lệ đúng rất quan trọng đối với sự hoạt động của FLC.

Trong điều khiển kinh điển, ta đã biết một Algorithm điều khiển thích nghi theo mô hình mẫu sử dụng phương pháp gradient hay phương pháp Lyapunov rất thích hợp cho việc điều khiển một quá trình không nhận biết được, đặc biệt đối với hệ phi tuyến. Một bộ điều khiển mờ với một luật hợp thành tuyến tính và các hàm liên hợp thuộc tam giác có thể xấp xỉ tuyến tính xung quanh trạng thái cân bằng. Do đó ta sử dụng ý tưởng của bộ điều khiển thích nghi kinh điển để áp dụng cho bộ điều khiển mờ thích nghi với một vài sự xấp xỉ nào đó. Mục tiêu chính của mục này là:

☛ Tìm ra cách tiếp cận định lượng để xác định mô hình toán học của bộ điều khiển mờ với một vài sự xấp xỉ nào đó.

☛ Xây dựng bộ điều khiển mờ thích nghi cho những hệ thống phi tuyến và hệ thống biến đổi theo thời gian trên cơ sở lý thuyết thích nghi kinh điển. Bộ điều khiển này có thể sử dụng để điều khiển đối tượng như là bộ thích

nghi trực tuyến, hoặc dùng làm cơ sở cho việc tổng hợp bộ điều khiển mờ thông thường.

Để đơn giản ta tiến hành xây dựng cơ chế thích nghi cho bộ điều khiển mờ hai đầu vào từ kết quả đó có thể dễ dàng mở rộng cho những bộ điều khiển mờ có nhiều đầu vào khác. Cấu trúc của các bộ điều khiển mờ thích nghi dựa trên cơ sở lý thuyết Lyapunov và phương pháp Gradien kinh điển.

2.7.2. Mô hình toán học của bộ điều khiển mờ

Xét bộ Điều khiển mờ hai đầu vào như hình 2.27. Để xây dựng mô hình toán học của nó ta thực hiện theo các bước sau:

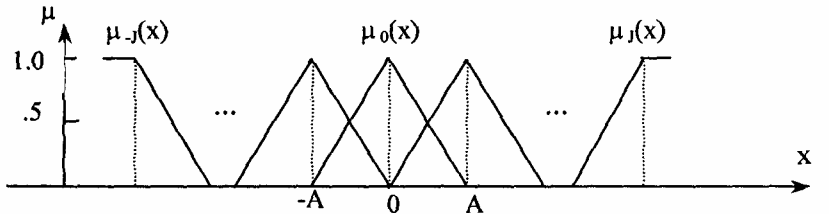
a/ Chọn các hàm liên thuộc

Các tập mờ đầu vào được chọn để mờ hoá là E và R. Ta chọn số lượng các tập mờ vào và ra bằng nhau và bằng N, các hàm liên thuộc sơ bộ chọn hình tam giác với mỗi hàm liên thuộc bao phủ không gian trạng thái 2A cho mỗi đầu vào và 2B cho đầu ra. Giả sử chọn j hàm liên thuộc âm cho E, R, U, chọn j hàm liên thuộc dương cho E, R, U và 1 hàm liên thuộc bằng zero cho E, R, U (hình 2.28). Như vậy số lượng các hàm liên thuộc của mỗi biến vào/ra là: $N = 2j + 1$.

Để đơn giản cho việc xây dựng luật hợp thành, thay vì sử dụng các ngôn ngữ như "âm nhiều", "dương nhiều" v.v... ta sử dụng các chỉ số là số, ví dụ

$$\mu_{-1}(x), \mu_{-2}(x), \mu_{-0}(x), \mu_1(x)$$

Ta thấy rằng, mặc dầu sử dụng các hàm liên thuộc giống nhau để mô tả 2 tập mờ đầu vào nhưng thông qua các hệ số k_1 và λ (hình 2.27) chúng thực sự là các hàm liên thuộc khác nhau.



Hình 2.28. Minh hoạ việc định nghĩa hàm liên thuộc cho các biến đầu vào và đầu ra

b/ Chọn luật điều khiển

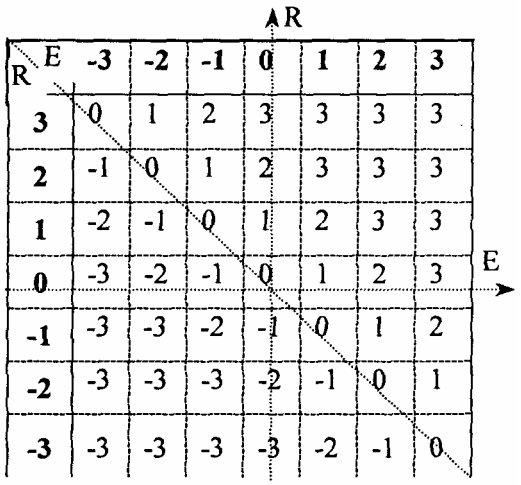
Với bộ điều khiển mờ 2 đầu vào, mỗi đầu vào có N tập mờ ta sẽ có N^2 luật điều khiển miêu tả tất cả các khả năng kết hợp của E_i và R_j Dạng tổng quát của luật hợp thành là:

Nếu E = E_i và R = R_j thì U = u_k Với k = f(i, j)

Định nghĩa 1: *Các luật điều khiển của một bộ điều khiển mờ được gọi là tuyến tính nếu f(i,j) là 1 hàm tuyến tính đối với i và j.*

Ví dụ: f = i + j; f = I + j + 1 vv

Trong đó f(i, j) là quy luật để sinh ra các luật điêu khiển. Với các f(i, j) khác nhau sẽ cho các luật điều khiển khác nhau. Việc chọn luật điều khiển có thể coi là một nghệ thuật và phụ thuộc rất nhiều vào kiến thức và kinh nghiệm của các chuyên gia. Trong mục này tác giả đề cập đến việc chuẩn hóa và đơn giản hóa việc chọn luật điều khiển nhằm tạo điều kiện thuận lợi cho người thiết kế hệ điều khiển mờ.



R \ E	-3	-2	-1	0	1	2	3
3	0	1	2	3	3	3	3
2	-1	0	1	2	3	3	3
1	-2	-1	0	1	2	3	3
0	-3	-2	-1	0	1	2	3
-1	-3	-3	-2	-1	0	1	2
-2	-3	-3	-3	-2	-1	0	1
-3	-3	-3	-3	-3	-2	-1	0

Hình 2.29. Luật hợp thành tuyến tính

Hình 2.29 minh hoạ luật điều khiển tuyến tính với f(i,j) = i +j cho bộ điều khiển mờ 2 đầu vào 1 đầu ra với 7 hàm liên thuộc cho mỗi biến vào và ra. Bảng 2.1 và Hình 2.30 là quan hệ vào-ra của luật hợp thành tuyến tính.

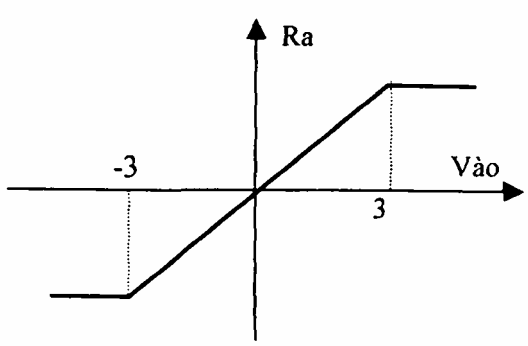
Bảng 2.1

I + j	-3	-2	1	0	1	2	3
U_{k-l}	3	2	-1	0	1	2	3

Đinh nghĩa 2: *Bộ điều khiển mờ cơ sở (Basis Fuzzy Controll - BFC) là bộ điều khiển mờ có 2 đầu vào và 1 đầu ra, số tập mờ của các đầu vào và đầu ra bằng nhau, luật hợp thành được sử dụng là luật hợp thành tuyến tính*

c/ Phân tích luật cơ sở thành ô suy luận

Các luật cơ sở chia vùng làm việc của bộ điều khiển mờ cơ bản thành nhiều ô vuông, với đầu ra của luật ở trên 4 góc như hình 2.29. Vì tất cả các thao tác mờ đều có thể được tính toán trên các ô này nên chúng được gọi là ô suy luận [33], [55].



Hình 2.28. Quan hệ Vào-ra của luật hợp thành tuyến tính

Một cách tổng quát ta có thể chọn ô suy luận IC(i, j) để phân tích. Ô này được tạo bởi các hàm liên thuộc $\mu_i(E)$, $\mu_{i+1}(E)$, $\mu_j(R)$ và $\mu_{j+1}(R)$ các đường chéo của ô chia chúng ra thành 4 vùng (ICI... IC4) (hình 2.3 l).

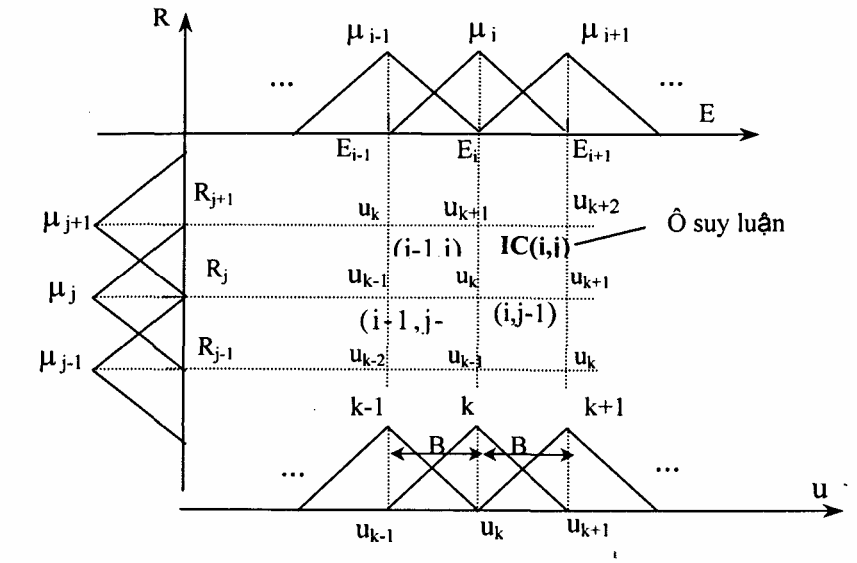
Vi trí tuyệt đối của 1 ô suy luận IC(i, j) trong luật cơ bản là từ [iA, jA] đến [(i+1)A,, (j + 1)A], vị trí tương đối của mỗi vùng trong ô IC(i,j) là từ [0, 0] đến [A, A].

Các dữ liệu vào (E, R) trong luật cơ bản luôn luôn được ánh xạ đến dữ liệu vào tương đối (e*, r* trong IC(i, j) theo công thức [22]:

$$E = iA + e^* \ (i = ..., -1, 0, 1, ...) \quad (2.52)$$

$$R = jA + r^* \ (j = ..., -1, 0, 1, ...). \quad (2.53)$$

Tất cả những thao tác mờ bao gồm "Mờ hoá", "suy diễn mờ" và "giải mờ" đều có thể được thực hiện trong ô suy luận IC.



Hình 2.29. Sự hình thành ô suy luận từ luật hợp thành

d/ Các thao tác mờ trong ô suy luận

Trong ô suy luận ta có thể thực hiện các thao tác mờ như: Mờ hoá, suy diễn mờ và giải mờ. Sử dụng phương pháp suy luận Max-Min của Mamdani, các thao tác đó được trình bày như sau:

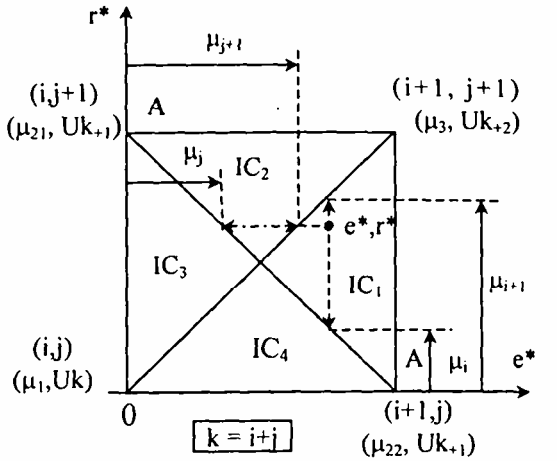
+ Mờ hoá: Từ các biểu thức (2.52) và (2.53) ta thấy trong một ô IC(i,j) các đầu vào (E, R) được xác định bởi (e*, r*) với các giá trị hàm liên thuộc của e* là μ_i và μ_{i+1}, các giá trị hàm liên thuộc của r* là μ_i và μ_{i+1}

Vì luôn tồn tại quan hệ: $\mu_i + \mu_{i+1} = 1$; μ_j và $\mu_{j+1} = 1$ do đó giá trị các hàm liên thuộc đầu vào trong ô suy luận:

$$\mu_i = 1 - \frac{e^*}{A}; \quad \mu_{i+1} = \frac{e^*}{A}$$
$$\mu_j = 1 - \frac{r^*}{A}; \quad \mu_{j+1} = \frac{r^*}{A} \qquad (2.54)$$

	μ_1	μ_2	μ_3
IC1	μ_i	μ_j	μ_{j+1}
IC2	μ_j	μ_j	μ_{i+1}
IC3	μ_j	μ_{j+1}	μ_{i+1}
IC4	μ_i	μ_{i+1}	μ_{j+1}

Bảng 2.2. Kết quả của phép lấy Max-min trong ô suy luận



Hình 2.30. Các vùng trong ô suy luận

+ Suy diễn mờ

Từ luật hợp thành cơ sở: **Nếu E = Ei và R = Rj thì U = u_k**

với $k = f(i,j) = i + j.$ (2.55)

Hàm liên thuộc của các tập mờ đầu ra được biểu diễn trong Hình 2.29 với giá trị đầu ra là:

$$u_k = k.B. \quad (2.56)$$

Tại mỗi vùng của ô suy luận ta thu được các giá trị μ_1, μ_2, μ_3 (bảng 2.2) thông qua phép lấy Max-min [21] với:

$$\left.\begin{aligned} &\mu_1 = \min(\mu_i,\mu_j) \text{ cho đầu ra } u_k \\ &\mu_{21} = \min(\mu_i,\mu_{j+1}) \text{ cho đầu ra } u_{k+1} \\ &\mu_{22} = \min(\mu_{i+1},\mu_j) \text{ cho đầu ra } u_{k+1} \\ &\mu_3 = \min(\mu_{i+1},\mu_{j+1}) \text{ cho đầu ra } u_{k+2} \\ &\mu_2 = \mathrm{Max}(\mu_{21},\mu_{22}) \end{aligned}\right\} \quad (2.57)$$

+ Giải mờ

Dùng phương pháp trung bình trọng tâm [20] ta được tín hiệu ra:

$$u = \frac{\sum_{l=1}^{3} \mu_l u_{k+l-1}}{\sum_{l=1}^{3} \mu_l} \qquad (2.58)$$

trong đó I = 1, 2, 3, 4 là các vùng tương ứng của ô suy luận.

e/ Xây dựng biểu thức toán học của bộ điều khiển mờ

Qua các phân tích trên ta thấy rằng các tín hiệu vào khác nhau (e*, r*) có thể rơi trên các vùng khác nhau của ô suy luận từ IC1 - IC4, đó là do kết quả của phép lấy Max- min.

+ Xét vùng IC1:

Từ (2.54) và bảng 2.2 ta có:

$$\sum_{l=1}^{3} \mu_l = \mu_i + \mu_j + \mu_{j+1} = \mu_{i+1} = 2 - \frac{e^*}{A} = \gamma_1^{-1} \qquad (2.59)$$

Từ bảng (2.2), (2.54) và (2.58) ta có:

$$\sum_{l=1}^{3} \mu_l u_{k+l-1} = \mu_i u_{k-1} + \mu_j u_k + \mu_{j+1} u_{k+1}$$

$$= (1 - e^*/A)(k - 1)B + (1 - r^*/A)kB + r^*/A(k + 1)B$$

$$= B[(k - 1)A + e^* + r^*]/A + kB(1 - e^*/A).$$

Với S = E + R = (k - 1)A + e* + r* ta có:

$$\sum_{l=1}^{3} \mu_l u_{k+l-2} = \frac{B}{A} S + kB(\gamma_1^{-1} - 1)$$

Từ đó ta rút ra:

$$u_1 = \frac{\sum_{l=1}^{3} \mu_l u_{k+l-2}}{\sum_{l=1}^{3} \mu_l} = \frac{\frac{B}{A} S + kB(\gamma_1^{-1} - 1)}{\gamma_1^{-1}}$$

$$u_1 = \frac{B}{A} \gamma_1 S + kB(1 - \gamma_1)$$

$$u_1 = kB + \frac{B}{A} \gamma_1 (S - kA).$$

Tương tự với các ô suy luận khác, cuối cùng ta thu được [10]:

$$u_1 = \frac{B}{A}\gamma_1 S + kB(1-\gamma_1) \quad (I = 1, 2, 3, 4) \tag{2.60}$$

hoặc

$$u_l = B/A\ \gamma_l(S-kA) + kB \tag{2.61}$$

$$\left.\begin{aligned} \gamma_l &= (1+\mu_i)^{-1} = (2-e^*/A)^{-1} \\ \gamma_2 &= (1+\mu_j)^{-1} = (2-e^*/A)^{-1} \\ \gamma_3 &= (1+\mu_{i+1})^{-1} = (1+e^*/A)^{-1} \\ \gamma_4 &= (1+\mu_{j+1})^{-1} = (1+e^*/A)^{-1} \\ S &= E + R = KI(\lambda e + r) \end{aligned}\right\} \tag{2.62}$$

$$= (k-1)A + e^* + r^*, \quad k = i + j$$

γ_1 (I = 1, 2, 3, 4) là tham số phi tuyến trong vùng IC1.

Ta thấy điều khiển mờ với luật hợp thành tuyến tính thực sự là điều khiển phi tuyến như biểu thức (2.61). Nó sẽ trở thành điều khiển tuyến tính ở trạng thái cân bằng. Trong biểu thức (2.61) ta cần phải xác định các hệ số khuếch đại tỷ lệ đầu vào k_1, λ và đầu ra K. Giá trị danh định của các hệ số khuếch đại đầu vào k_1 và λ có thể được xác định theo phương pháp của H.X. Li [10]. Thông thường việc xác định hệ số khuếch đại đầu ra K đúng là rất khó khăn.

2.7.3. Xây dựng cơ cấu thích nghi cho bộ điều khiển mờ

a/ Hệ điều khiển thích nghi theo mô hình mẫu (MRAS) dùng lý thuyết thích nghi kinh điển

Xét một đối tượng điều khiển được mô tả bởi phương trình:

$$\frac{dy}{dt} = -ay + bu. \tag{2.63}$$

Mô hình mẫu có phương trình:

$$\frac{dy_m}{dt} = -a_m y_m + b_m u_c. \tag{2.64}$$

Tín hiệu điều khiển: $u = \theta_1 u_c - \theta_2 y$ với sai số: $\varepsilon = y - y_m$

Biểu thức ε chứa tham số điều chỉnh. Ta cần tìm ra cơ cấu thích nghi để điều chỉnh các tham số θ_1 và θ_2 tới giá trị mong muốn sao cho sai số ε tiến tới 0. Để tìm ra cơ cấu thích nghi này ta có thể dùng lý thuyết ổn định Lyapunov

hoặc phương pháp Gradient theo các bổ đề sau:

Bổ đề 2.1: (luật thích nghi theo Lyapunov)

Giả thiết bη > 0 và và chọn hàm Lyapunov có dạng:

$$V(\varepsilon, \theta_1, \theta_2) = \frac{1}{2}\left[\varepsilon^2 + \frac{1}{b\eta}(b\theta_2 + a - a_m)^2 + \frac{1}{b\eta}(b\theta_1 - b_m)^2\right]$$

thì quy luật điều chỉnh các tham số θ_1, θ_2 để cho $\varepsilon \rightarrow 0$ là:

$$\frac{d\theta_1}{dt} = -\eta u_c \varepsilon; \qquad \frac{d\theta_2}{dt} = \eta y \varepsilon. \qquad (2.65)$$

Nếu chỉ có 1 tham số biến thiên, luật điều chỉnh thích nghi tham số trở thành:

$$\frac{d\theta_1}{dt} = -\eta u_c \varepsilon. \qquad (2.66)$$

Bổ đề 2.2: (Luật thích nghi theo Gradient)

Giả thiết $\underline{\theta}$ là một véctơ tham số cần được xác định, và phụ thuộc độ sai lệch giữa đầu ra của đối tượng (y) và đầu ra của mô hình (y_m). Tiêu chuẩn sai lệch đáp ứng của hệ được chọn:

$$J(\theta) = \frac{1}{2}\varepsilon^2 \rightarrow 0. \qquad (2.67)$$

thì quy luật điều chỉnh θ theo hướng của gradient của J là:

$$\frac{d\underline{\theta}}{dt} = -\eta \frac{\partial J}{\partial \underline{\theta}} = -\eta\varepsilon \frac{\partial \varepsilon}{\partial \underline{\theta}} = -\eta\varepsilon \frac{\partial y}{\partial \underline{\theta}}. \qquad (2.68)$$

Trong điều khiển thích nghi kinh điển, nói chung không cần một mô hình mẫu hoàn hảo, tuy nhiên sự sai khác giữa mô hình và đối tượng cũng như tính phi tuyến của nó chỉ nằm trong giới hạn nào đó, nếu quá giới hạn này bộ điều chỉnh sẽ không làm việc hiệu quả nữa. Để khắc phục nhược điểm đó, trong cuốn sách này các tác giả đề xuất sử dụng hệ điều khiển mờ thích nghi theo mô hình.

b/ Điều chỉnh thích nghi hệ số khuếch đại đầu ra của bộ điều khiển mờ

Tín hiệu đầu ra của bộ điều khiển mờ (2.60) được viết:

$$U_I = \frac{B}{A}\gamma_I S + kB(1-\gamma_I), \quad \text{với } \gamma \text{ là thông số phi tuyến.}$$

Thay $S = E + R = KI(\lambda+I)e$ (với $I = \int dt$) ta có:

$$U_I = \frac{B}{A}\gamma_I K_I(\lambda + I)e + kB(1-\gamma_I) = F.e + T$$

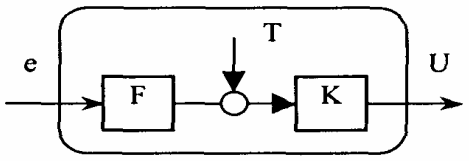
trong đó:

$$F = \frac{B}{A}\gamma_I S = \frac{B}{A}\gamma_I K_I(\lambda + I)$$

$$T = kB(1-\gamma_I).$$

Bộ điều khiển mờ 2 đầu vào trong biểu thức (2.60) với hệ số khuếch đại đầu ra K, có thể được biểu diễn như là F. e cộng thêm 1 giới hạn trễ T như biểu thức (2.69) (hình 2.31) giới hạn trễ T sẽ tiến tới zero khi hệ thống tiến đến điểm cân bằng [11], [12].

$$U = K(T + F.e) \qquad (2.69)$$



Hình 2.31. Sơ đồ khối bộ điều khiển mờ với hệ số khuếch đại đầu ra K

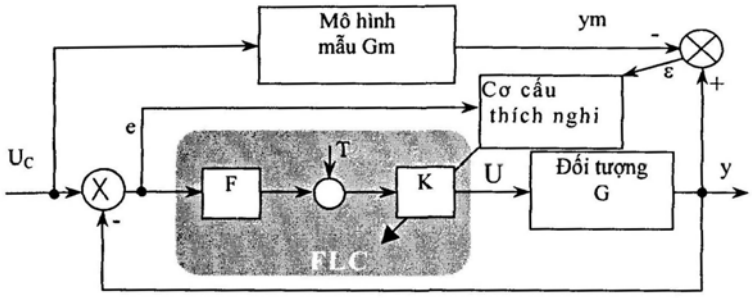
Ta sẽ áp dụng phương pháp Lyapunov và phương pháp Gradient để chỉnh định thích nghi hệ số khuếch đại đầu ra K của bộ điều khiển mờ. Quá trình điều chỉnh được thực hiện theo *2* câu trúc chính được gọi chung là điều khiển thích nghi mờ theo mô hình mẫu (MRAFC) (Model Reference Adaptive Fuzzy Controller). Ta tiến hành khảo sát 2 sơ đồ là sơ đồ phản hồi đầu ra và sơ đồ điều khiển thích nghi mờ theo mô hình hiệu chỉnh trước (FMRAFC) (Feedfonvard Model Reference Adaptive Fuzzy Controller).

c/ Sơ đo điều khiển thích nghi mờ theo mô hình mẫu (MRAFC)

Xét một cấu trúc điều khiển mờ thích nghi theo mô hình được biểu diễn trên hình 2.32 [19], [20].

Trong đó: Đối tượng Điều khiển có hàm số truyền G, mô hình mẫu có hàm truyền Gm, bộ điều khiển mờ bao gồm bộ điều khiển mờ cơ bản kết hợp với bộ khuếch đại K. Cần phải tìm ra quy luật chỉnh định hệ số K sao cho sai

lệch giữa mô hình và đối tượng tiến đến 0 (ε →0).



Hình 2.32. MRAFC điều chỉnh hệ số khuếch đại đầu ra

Xấp xỉ γ_1 trong (2.61) thành một hằng số, hệ thống vòng kín xung quanh trạng thái cân bằng trở thành tuyến tính với phương trình của vòng kín là:

$$y = \frac{KFG}{1+KFG}.U_C$$

Và

$$\frac{\partial \varepsilon}{\partial K} = \frac{\partial y}{\partial K} = \frac{KFG}{1+KFG}.\frac{e}{K} \approx G_m.\frac{e}{K}. \qquad (2.70)$$

Giả thiết y tiến đến y_m thì ta có thể xấp xỉ . $\frac{KFG}{1+KFG} \approx Gm$. Khi đó quy luật điều chỉnh thích nghi cho hệ số khuếch đại đầu ra của FLC có thể xác định từ (2.68) là:

$$\frac{dK}{dt} = -\eta G_m \frac{\varepsilon e}{K}. \qquad (2.71)$$

Để xét ổn định của sơ đồ trên, ta chọn hàm Lyapunov:

$$V = 1/2\ \varepsilon^2$$

$$\dot{V} = \varepsilon\left(\frac{\partial \varepsilon}{\partial K}\dot{K} + \frac{\partial \varepsilon}{\partial t}\right) = -\eta\left(\frac{e\varepsilon}{K}G_m\right)^2 + \varepsilon\frac{\partial \varepsilon}{\partial t}$$

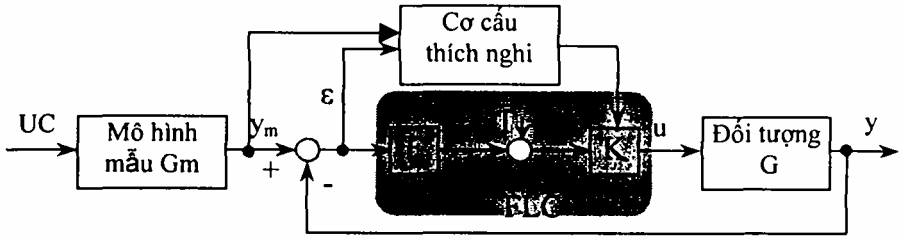
Hệ thống sẽ ổn định $\varepsilon\frac{\partial \varepsilon}{\partial t} < 0$.

d/ Sơ đồ điều khiển thích nghi mờ kiểu truyền thẳng (FMRAFC)

Một cấu trúc khác của bộ điều khiển thích nghi mờ được biểu diễn trên

hình 2.33 sơ đồ này gọi là sơ đổ thích nghi mờ truyền thẳng (Feedforward Model Reference Adaptive Fuzzy Controller - FMRAFC) [11].

Trong sơ đồ này sai lệch giữa tín hiệu đặt và tin hiệu đâu ra của đối tượng được thay thế bằng giá trị sai lệch giữa đối tượng và mô hình ε



Hình 2.33. Cấu trúc hệ FMRAFC

Hệ số khuếch đại thích nghi đầu ra: $y = \frac{KFG}{1+KFG} y_m$

$$\frac{\partial \varepsilon}{\partial K} = -\frac{\partial y}{\partial K} = -\frac{KFG}{(1+KFG)^2} \cdot \frac{y_m}{K} = -\frac{KFG}{1+KFG} \cdot \frac{\varepsilon}{K} \approx -\frac{\varepsilon}{K} \qquad (2.72)$$

trong đó: $\frac{y_m}{1+KFG} = \varepsilon$ và giả thiết rằng khi y tiến đến y_m thì KFG/(1 +KFG) ≈ 1.

Từ (2.68) ta rút ra quy luật thích nghi cho hệ số khuếch đại đầu ra là:

$$\frac{dK}{dt} = \eta \frac{\varepsilon^2}{k} \text{ (theo gradient)[21]} \qquad (2.73)$$

$$\frac{dK}{dt} = \eta y_m u_c \text{ (theo Lyapunov).} \qquad (2.74)$$

Ta thấy do hàm truyền của mô hình không có mặt ở luật thích nghi (2.73) và (2.74) nên cấu trúc thích nghi này chịu đựng tốt đối với giới hạn lớn độ sai lệch giữa mô hình và đối tượng. Trong thực tế nó chỉ cần một mô hình xấp xỉ gần đúng ví dụ mô hình mẫu bậc nhất: $G_m = \frac{b_m}{a_m + S}$ cũng có thể áp dụng cho phần lớn các đối tượng điều khiển.

2.7.4. Một số ứng dụng điều khiển các đối tượng công nghiệp

Mục đích của phần này là thông qua mô phỏng trình bày tính hiệu quả của bộ điều khiển thích nghi mờ được tổng hợp trên cơ sở lý thuyết điều khiển thích nghi kinh điển. Đồng thời thông qua đó (MRAFC) ta xác định được hệ số khuếch đại đầu ra cho bộ điều khiển mờ, làm cơ sở cho việc xây dựng thuật toán tổng hợp bộ điều khiển mờ. Các ứng dụng được xây dựng cho 3 lớp đối tượng điển hình trong công nghiệp:

☛ Đối tượng tuyến tính bậc hai trong đó có khâu tích phân được mô tả bởi:

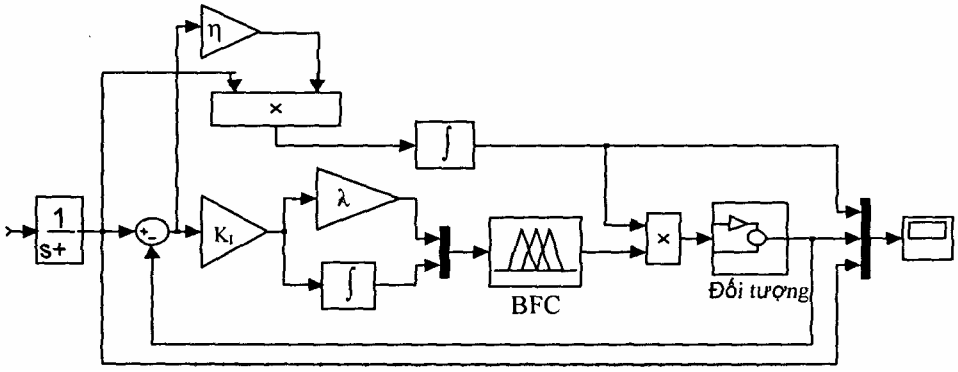
$$G_1(s) = \frac{1}{S}\frac{K}{TS+1}. \qquad (2.75)$$

☛ Đối tượng tuyến tính bậc 3 với những tham số không biết, được cho bởi cấu trúc gần đúng sau?

$$G_2(s) = \frac{1}{S+1}\left(\frac{K}{T_1S^2 + T_2S + 1}\right). \qquad (2.76)$$

☛ Một đối tượng phi tuyến với các thông số biến thiên theo thời gian được mô tả gần đúng bằng phương trình:

$$0.4\dot{y} + (1 + 0{,}2\sin(0{,}1t))y = 2u - \sin\dot{y}. \qquad (2.77)$$



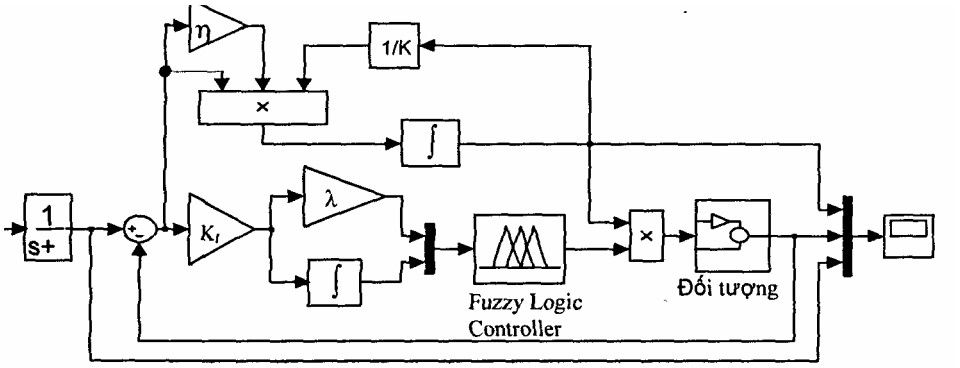
Hình 2.34. Sơ đồ cấu trúc hệ MRAFC với luật điều khiển theo Lyapunov

Mô hình mẫu là khâu quán tính bậc nhất có hàm truyền:

$$G_m = \frac{a_m}{b_m S + 1} = \frac{1}{S+1}$$

với $a_m = b_m = 1$. Tín hiệu đặt U_C là sóng hình vuông.

Sơ đồ cấu trúc hệ MRAFC với luật điều khiển theo Lyapunov được biểu diễn trên hình 2.34 và theo Gradient được biểu diễn trên hình 2.35.



Hình 2.35. Sơ đồ cấu trúc hệ MRAFC với luật điều khiển theo Gradient

a/ Kết quả mô phỏng

Các kết quả mô phỏng được chỉ ra trên các hình từ hình 2.36 đến hình 2.44.

Để tiện so sánh ta đưa ra đáp ứng tương ứng với *3* cấu trúc MRAC, FMRAFC theo Lyapunov và FMRAFC theo Phương pháp Gradient.

b/ Nhận xét

Từ kết quả mô phỏng ở trên ta rút ra một số nhận xét sau:

☛ Đáp ứng của hệ FMRAFC theo phương pháp Lyapunov và phương pháp Gradient gần giống nhau và được biểu diễn trên các hình từ hình 2.36 đến hình 2.41. Ta thấy:

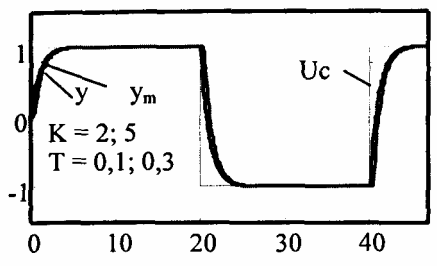
☛ Đối với đối tượng tuyến tính bậc hai có khâu tích phân đáp ứng của FMRAFC trong hình 2.36 và hình 2.37 đạt chất lượng động tốt, quá trình làm việc sẽ bám theo mô hình một cách nhanh chóng.

☛ Đối với đối tượng tuyến tính bậc 3 đáp ứng của FMRAFC trong hình 2.38 và hình 2.39 gần giống với đối tượng bậc nhất.

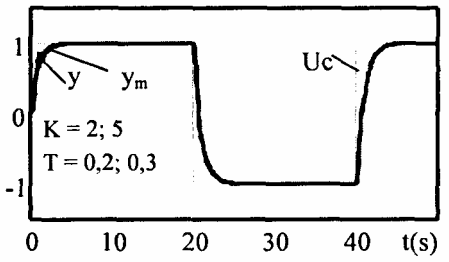
☛ Đối với đối tượng không tuyến tính biến đổi theo thời gian, đáp ứng của FMRAFC hình 2.40 và hình 2.41 không thay đổi nhiều so với đối tượng bậc 2.

Vậy hệ điều khiển thích nghi mờ (MRAFC) có thể đạt được đáp ứng tốt hơn rất nhiều so với hệ điều khiển thích nghi kinh điển (MRAC), đặc biệt cho

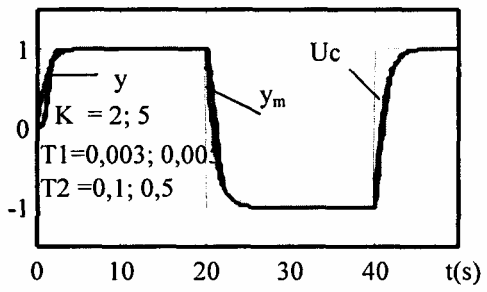
những đối tượng biến đổi theo thời gian và không mô hình hoá được. Bên cạnh đó chỉ ra khả năng to lớn của bộ điều khiển mờ thích nghi làm việc với các quá trình không nhận biết được. Từ những kết quả trên, ta có thể tiếp tục phát triển theo hướng này để xây dựng các bộ điều khiển mờ tự chỉnh trực tuyến mà có thể đạt được đáp ứng tối ưu một cách tự động cho một giới hạn rộng hơn các quá trình.



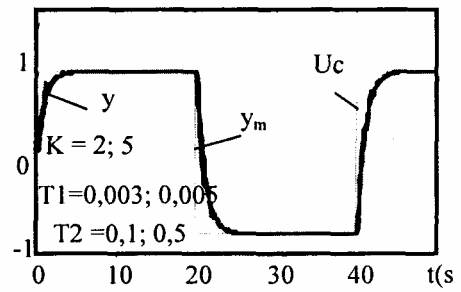
Hình 2.36: Đáp ứng của FMRAFC với lớp đối tượng bậc hai trong đó có khâu tích phân theo Liapunov ứng với 2 giá trị của K= 2; 5 và T = 0,1; 0,3



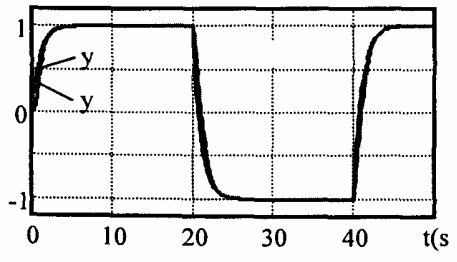
Hình 2.37: Đáp ứng của FMRAFC với lớp đối tượng bậc hai trong đó có khâu tích phân theo Gradient ứng với K=2; 5 và T=0,2; 0,3



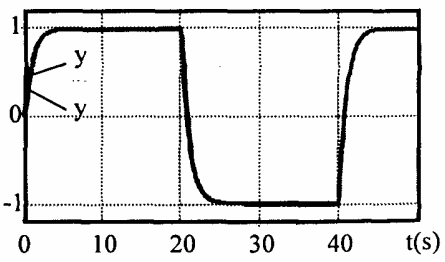
Hình 2.38: Đáp ứng của FMRAFC với lớp đối tượng bậc 3 theo Liapunov ứng với K= 2; 5; T_1=0,003; 0,005 và T_2 = 0,1; 0,5



Hình 2.39: Đáp ứng của FMRAFC với lớp đối tượng bậc 3 theo Gradient ứng với K=2; 5; T_1=0,003; 0,005 và T_2 = 0,1; 0,5



Hình 2.40: Đáp ứng hệ FMRAFC với đối tượng phi tuyến theo Liapunov



Hình 2.41: Đáp ứng của FMRAFC với đối tượng phi tuyến theo Gradient

Chương 3

TỔNG QUAN VỀ MẠNG NƠRON

3.1. NƠRON SINH HỌC

3.1.1. Chức năng, tổ chức và hoạt động của bộ não con người

Bộ não người có chức năng hết sức quan trọng trong đời sống của con người. Nó gần như kiểm soát hầu hết mọi hành vi của con người từ các hoạt động cơ bắp đơn giản đến những hoạt động phức tạp như học tập, nhớ, suy luận, tư duy, sáng tạo,...

Bộ não người được hình thành từ sự liên kết của khoảng 10^{11} phần tử (tế bào), trong đó có khoảng 10^{10} phần tử là nơron, số còn lại khoảng $9*10^{10}$ phần tử là các tế bào thần kinh đệm và chúng có nhiệm vụ phục vụ cũng như hỗ trợ cho các nơron. Thông thường một bộ não trung bình cân nặng khoảng 1,5 kg và có thể tích là 235 cm^3, cho đến nay người ta vẫn chưa thực sự biết rõ cấu tạo chi tiết của bộ não. Tuy vậy về đại thể thì cấu tạo não bộ được phân chia ra thành nhiều vùng khác nhau. Mỗi vùng có thể kiểm soát một hay nhiều hoạt động của con người.

Bộ não có cấu trúc nhiều lớp. Lớp bên ngoài thường thấy như là các nếp nhăn, là lớp có cấu tạo phức tạp nhất. Đây là nơi kiểm soát và phát sinh các hành động phức tạp như nghe, nhìn, tư duy,...

Hoạt động của bộ não nói riêng và của hệ thần kinh nói chung đã được con người quan tâm nghiên cứu từ lâu nhưng cho đến nay người ta vẫn chưa hiểu rõ thực sự về hoạt động của bộ não và hệ thần kinh. Đặc biệt là trong các hoạt động liên quan đến trí óc như suy nghĩ, nhớ, sáng tạo,... Tuy thế cho đến nay, người ta cũng có những hiểu biết căn bản về hoạt động cấp thấp của não.

Mỗi nơron liên kết với khoảng 10^4 nơron khác, cho nên khi hoạt động thì bộ não hoạt động một cách tổng lực và đạt hiệu quả cao. Nói một cách khác là các phần tử của não hoạt động một cách song song và tương tác hết sức tinh vi phức tạp, hiệu quả hoạt động thường rất cao, nhất là trong các vấn đề phức tạp, về tốc độ xử lý của bộ não người rất nhanh mặc dù tốc độ xử lý của

mỗi nơron (có thể xem như phần tử xử lý hay phần tử tính) là rất chậm so với xử lý của các cổng logic silicon trong các chíp vi xử lý (10^{-3} giây so với 1 0^{-10} giây).

Hoạt động của cả hệ thống thần kinh bao gồm não bộ và các giác quan như sau: Trước hết con người bị kích thích bởi giác quan từ bên ngoài hoặc trong cơ thể. Sự kích thích đó được biến thành các xung điện bởi chính các giác quan tiếp nhận kích thích. Những tín hiệu này được chuyển về trung ương thần kinh là não bộ để xử lý. Trong thực tế não bộ liên tục nhận thông tin xử lý, đánh giá và so sánh với thông tin lưu trữ để đưa ra các quyết định thích đáng.

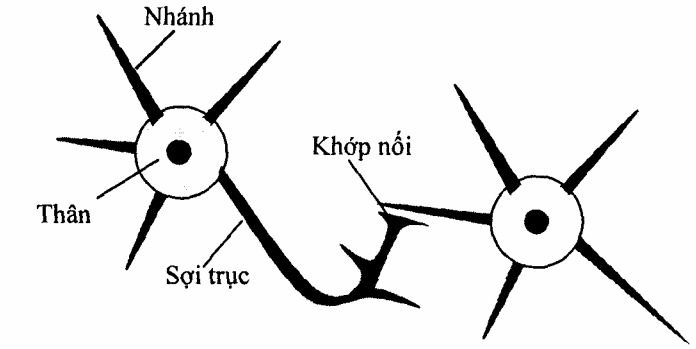
Những mệnh lệnh cần thiết được phát sinh và gửi đến những bộ phận thi hành thích hợp như các cơ tay, chân,... Những bộ phận thi hành biến những xung điện thành dữ liệu xuất của hệ thống.

Tóm lại: bộ não người có chức năng hết sức quan trọng đối với đời sống của con người. Cấu tạo của nó rất phức tạp, tinh vi bởi được tạo thành từ mạng nơron có hàng chục tỉ tế bào với mức độ liên kết giữa các nơron là rất cao. Hơn nữa, nó còn được chia thành các vùng và các lớp khác nhau. Bộ não hoạt động dựa trên cơ chế hoạt động song song của các nghìn tạo nên nó.

3.1.2. Mạng nơron sinh học

a/ Cấu tạo

Nơron là phần tử cơ bản tạo nên bộ não con người. Sơ đồ cấu tạo của một nơron sinh học được chỉ ra như trong hình 3.1. Một nơron điển hình có 3 phần chính:



Hình 3.1. Mô hình 2 nơron sinh học

- Thân nơron (so ma): Nhân của nơron được đặt ở đây.

- Các nhánh (dendrite): Đây chính là các mạng dạng cây của các dây thần kinh để nối các soma với nhau.

- Sợi trục (Axon): Đây là một nối kết, hình trụ dài và mang các tín hiệu từ đó ra ngoài. Phần cuối của axon được chia thành nhiều nhánh nhỏ (cả của dendrite và axon) kết thúc trong một cơ quan nhỏ hình củ hành được gọi là synapte mà tại đây các nơron đưa các tín hiệu của nó vào các nơron khác. Những điểm tiếp nhận với các synapte trên các nơron khác có thể ở các dendrite hay chính soma.

b/ Hoạt động

Các tín hiệu đưa ra bởi một synapte và được nhận bởi các dendrite là các kích thích điện tử. Việc truyền tín hiệu như trên liên quan đến một quá trình hóa học phức tạp mà trong đó các chất truyền đặc trưng được giải phóng từ phía gửi của nơi tiếp nối. Điều này làm tăng hay giảm điện thế bên trong thân của nơron nhận. Nơron nhận tín hiệu sẽ kích hoạt (fire) nếu điện thế vượt khỏi một ngưỡng nào đó và một xung (hoặc điện thế hoạt động) với độ mạnh (cường độ) và thời gian tồn tại cố định được gửi ra ngoài thông qua axon tới phần nhánh của nó rồi tới các chỗ nối synapte với các nơron khác. Sau khi kích hoạt, nơron sẽ chờ trong một khoảng thời gian được gọi là chu kỳ, trước khi nó có thể được kích hoạt lại. Synapses là Hưng phấn (excitatory) nếu chúng cho phép các kích thích truyền qua gây ra tình.trạng kích hoạt (fire) đối với nơron nhận. Ngược lại, chúng là ức chế (inhibitory) nếu các kích thích truyền qua làm ngăn trở trạng thái kích hoạt (fire) của nơron nhận.

3.2. MẠNG NƠRON NHÂN TẠO

3.2.1. Khái niệm

Nơron nhân tạo là sự sao chép nơron sinh học của não người, nó có những đặc tính sau:

- Mỗi nơron có một số đầu vào, những kết nối (Synaptic) và một đầu ra (axon)

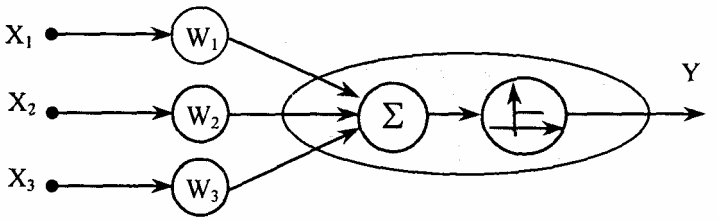
- Một nơron có thể hoạt động (+35 mV) hoặc không hoạt động (-0,75 mV)

- Chỉ có một đầu ra duy nhất của một nơron được nối với các đầu vào khác nhau của nơron khác. Điều kiện để nơron được kích hoạt hay không kích hoạt chỉ phụ thuộc những đầu vào hiện thời của chính nó.

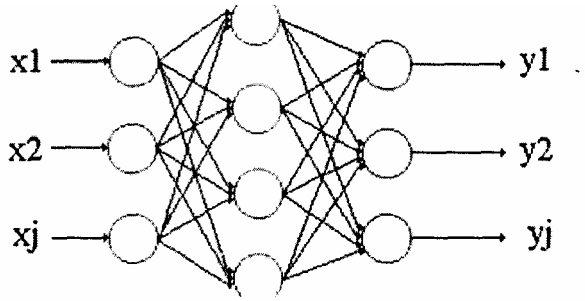
Một nơron trở nên tích cực nếu đầu vào của nó vượt qua ngưỡng ở một mức nhất định..

Có nhiều kiểu nơron nhân tạo khác nhau. Hình 3.2 biểu diễn một kiểu rất đơn giản.

Các đầu vào có hàm trọng W_j và bộ tổng. Đầu ra của bộ tổng được sử dụng để quyết định một giá trị của đầu ra thông qua hàm chuyển. Có nhiều kiểu hàm chuyển khác nhau (sẽ được đề cập ở phần sau). Tương tự nơron sinh học của con người, nơron sẽ được kích hoạt nếu tổng giá trị vào vượt quá ngưỡng và không được kích hoạt nếu tổng giá trị vào thấp hơn ngưỡng. Sự làm việc như vậy của nơron gọi là sự kích hoạt nhảy bậc.



Hình 3.2. Mô hình nơron đơn giản



Hình 3.3. Mạng nơron 3 lớp

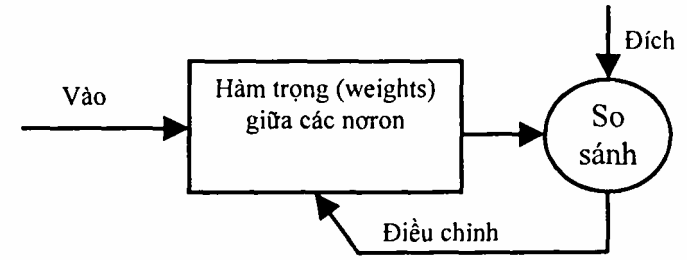
Kết nối một vài nơron ta được mạng nơron. Hình 3.3 là một mạng nơron gồm 3 lớp: lớp vào, lớp ẩn và lớp ra.

Các nơron lớp vào trực tiếp nhận tín hiệu ở đầu vào, ở đó mỗi nơron chỉ có một tín hiệu vào. Mỗi nơron ở lớp ẩn được nối với tất cả các nơron lớp vào và lớp ra. Các nơron ở lớp ra có đầu vào được nối với tất cả các nơron ở

lớp ẩn, chúng là đầu ra của mạng. Cần chú ý rằng một mạng nơron cũng có thể có nhiêu lớp ẩn. Các mạng nơron trong mỗi nơron chỉ được liên hệ với tất cả các nơron ở lớp kế tiếp và tất cả các mối liên kết chỉ được xây dựng từ trái sang phải được gọi là mạng nhiều lớp truyền thẳng (perceptrons).

Thông thường mạng nơron được điều chỉnh hoặc được huấn luyện để hướng các đầu vào riêng biệt đến đích ở đầu ra. Cấu trúc huấn luyện mạng được chỉ ra trên hình 3.4. Ở đây, hàm trọng của mạng được điều chỉnh trên cơ sở so sánh đầu ra với đích mong muốn (taget) cho tới khi đầu ra mạng phù hợp với đích. Những cặp vào/đích (input/taget) được dùng để giám sát cho sự huấn luyện mạng.

Để có được một số cặp vào/ra, ở đó mỗi giá trị vào được gửi đến mạng và giá trị ra tương ứng được thực hiện bằng mạng là sự xem xét và so sánh với giá trị mong muốn. Bình thường tồn tại một sai số bởi lẽ giá trị mong muốn không hoàn toàn phù hợp với giá trị thực. Sau một lần chạy, ta có tổng bình phương của tất cả các sai số. Sai số này được sử dựng để xác định các hàm trọng mới.



Hình 3.4. Cấu trúc huấn luyện mạng nơron

Sau mỗi lần chạy, hàm trọng của mạng được sửa đổi với đặc tính tốt hơn tương ứng với đặc tính mong muốn. Từng cặp giá trị vào/ra phải được kiểm tra và trọng lượng được điều chỉnh một vài lần. Sự thay đổi các hàm trọng của mạng được dừng lại nếu tổng các bình phương sai số nhỏ hơn một giá tri đặt trước hoặc đã chạy đủ một số lần chạy xác định (trong trường hợp này mạng có thể không thoả mãn yêu cầu đặt ra do sai lệch còn cao).

Có 2 phương pháp cơ bản đê huấn luyện mạng nơron: Huấn luyện gia tăng (tiến dần) và huấn luyện theo gói. Sự huấn luyện theo gói của mạng nhận được bằng việc thay đổi hàm trọng và độ dốc trong một tập (batch) của véctơ đầu vào. Huấn luyện tiến dần là thay đổi hàm trọng và độ dốc của

mạng sau mỗi lần xuất hiện của một phần tử véctơ đầu vào. Huấn luyện tiến dần đôi khi được xem như huấn luyện trực tuyến hay huấn luyện thích nghi.

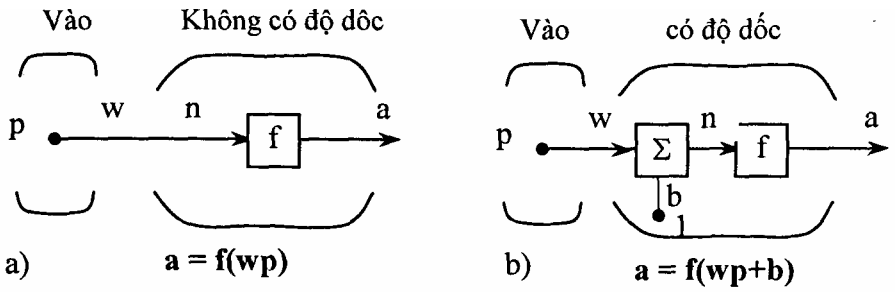
Mạng nơron đã được huấn luyện để thực hiện những hàm phức tạp trong nhiều lĩnh vực ứng dụng khác nhau như trong nhận dạng, phân loại sản phẩm, xử lý tiếng nói, chữ viết và điều khiển hệ thống.

Thông thường để huấn luyện mạng nơron, người ta sử dụng phương pháp huấn luyện có giám sát, nhưng cũng có mạng thu được từ sự huấn luyện không có giám sát. Mạng huấn luyện không giám sát có thể được sử dụng trong trường hợp riêng để xác đinh nhóm dữ liệu.

Mạng nơron bắt đầu xuất hiện từ 50 năm nhưng mới chi tìm thây các ứng dụng từ khoảng 10 năm trở lại đây và vẫn đang phát triển nhanh chóng. Như vậy, rõ ràng có sự khác biệt với những hệ thống điều khiển hoặc tối ưu hoá, nơi mà các thuật ngữ, cơ sở toán học và thủ tục thiết kế đã được thiết lập chắc chắn và được ứng dụng từ nhiều năm.

3.2.2. Mô hình nơron

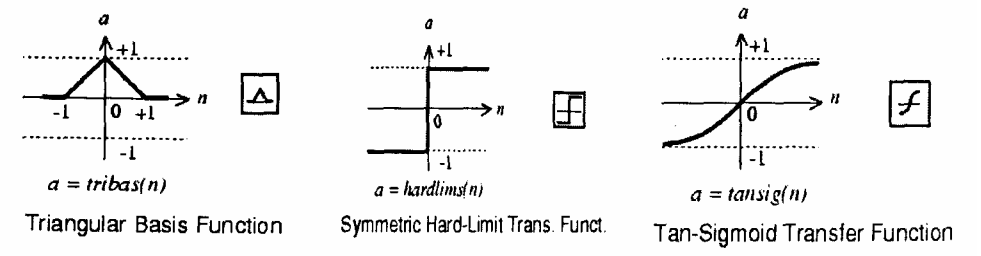
a/ Nơron đơn giản: một nơron với một đầu vào vô hướng và không có độ dốc được chỉ ra trên hình 1.5a,b.



Hình 3.5a,b. Mô hình nơron đơn giản

Tín hiệu vào vô hướng p thông qua trọng liên kết vô hướng w trở thành wp cũng là đại lượng vô hướng. Ở đây wp là đối số duy nhất của hàm truyền f, tín hiệu đầu ra là đại lượng vô hướng a. Hình l.5b là nơron có độ dốc b. Ta có thể hiểu b như là phép cộng đơn giản vào tích wp hoặc như là một sự thăng giáng của hàm f ở hình a đi một lượng b. Độ dốc được xem như một trọng lượng, chỉ có điều đầu vào là một hằng số bằng 1. Tín hiệu vào hàm truyền mạng là n là tổng của trọng đầu vào wp và độ đốc b, đáp ứng ra a

được coi là đối số của hàm chuyển f. Hàm chuyển f có thể là hàm bước nhảy, hàm sigmoid... Hình 3.6 dưới đây giới thiệu một số dạng hàm chuyển của nơron.

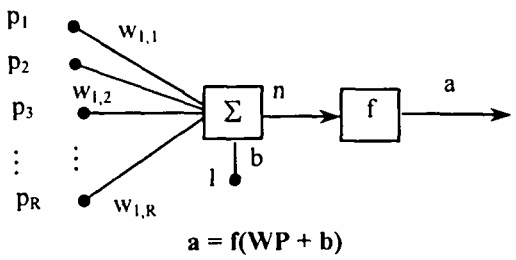


Hình 3.6. Một số dạng hàm chuyển của mạng nơron

Chú ý rằng w và b đều là các tham số điều chỉnh vô hướng của nơron. Ý tưởng cơ bản của mạng nơron điều chỉnh các tham số này như thế nào đó để mạng đạt được một đích mong muốn hay một hành vi nào đó. Như vậy ta có thể huấn luyện mạng làm một công việc nào đó bằng cách điều chỉnh các trọng liên kết và độ dốc, hoặc mạng có thể tự điều chỉnh các tham số này để đạt được các kết quả mong muốn.

Chú ý:

- Tất cả các nơron đều cho sẵn một độ dốc (b), tuy nhiên chúng ta có thể bỏ đi khi cần thiết.



Hình 3.7. Nơron với R đầu vào

- Độ dốc b là một tham số điều chỉnh vô hướng của nơron, nó không phải là một đầu vào, song hằng số 1 phải dược xem như đầu vào và nó cân được coi như vậy khi xem xét độ phụ thuộc tuyến tính của các véc lơ đầu vào.

b/ Nơron với nhiều đầu vào (véc tơ vào)

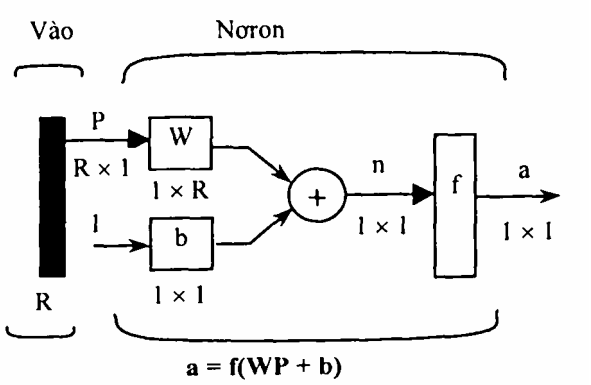
Nơron với véctơ vào gồm R phần tử được chi ra trên hình 3.7. Trong đó các đầu vào là p_1, p_2,…, p_R được nhân với các trọng liên kết $w_{1,1}$, $w_{1,2}$,… $w_{1,R}$ các trọng liên kết được biểu diễn bằng ma trận hàng, véctơ p là ma trận cột, khi đó ta có:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + w_{1,3}p_3 + ... + w_{1,R}p_R + b$$

$$n = W*P + b$$

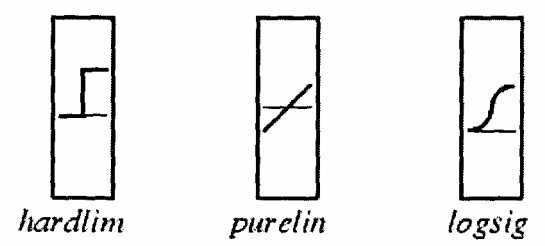
Trong đó W là ma trận trọng liên kết có kích thước 1 x R, P là véctơ vào gồm R phần tử.

Cách biểu diễn trên sẽ rất khó khăn khi mô tả mạng gồm nhiều nơron và có nhiều lớp. Để đơn giản ta sử dụng ký hiệu như hình 3.8.



Hình 3.8. Ký hiệu nơron với R đầu vào

Trong đó véctơ đầu vào được biểu diễn bởi thanh đậm bên trái. Kích thước của p được chỉ ra bên dưới ký hiệu p là R x 1.(ta sử dụng chữ viết hoa R để chỉ kích thước của một véctơ). Như vậy p là một véctơ gồm R phần tử vào, các đầu vào này nhân với ma trận W (1xR). Giống như phần trên, ở đây hằng số 1 đưa vào nơron như một đầu vào và được nhân với độ dốc b. Hàm chuyển của mạng là f. Đầu vào hàm chuyển là n bằng tổng của độ dốc b và tích Wp. Tổng này được đi qua hàm chuyển f để có đầu ra của nơron là a. Trong trường hợp này a là một đại lượng vô hướng. Chú ý rằng nếu có từ 2 nơron trở lên thì đầu ra sẽ là một véctơ.



Hình 3.9. một số hàm chuyển thông dụng

Một lớp mạng đã được định nghĩa như hình 3.8, đó là sự kết hợp giữa các trọng liên kết, phép nhân, phép cộng, độ dốc b và hàm chuyển f. Trong đó kích thước của ma trận được chỉ rõ ở bên dưới tên biển ma trận của chúng. Khi một hàm chuyển cụ thể được sử dụng thì trên hình vẽ biểu tượng của hàm chuyển đó sẽ thay thế f ở trên. Hình 3.9 là một vài ví dụ về các hàm

chuyển thông dụng.

3.3. CẤU TRÚC MẠNG

Nhiều nơron kết hợp với nhau tạo thành mạng nghìn, mạng nơron có thể có một lớp hoặc nhiều lớp.

3.3.1. Mạng một lớp

Một cấu trúc mạng 1 lớp với R đầu vào và S nơron được chỉ ra trên hình 3.10.

Trong đó:

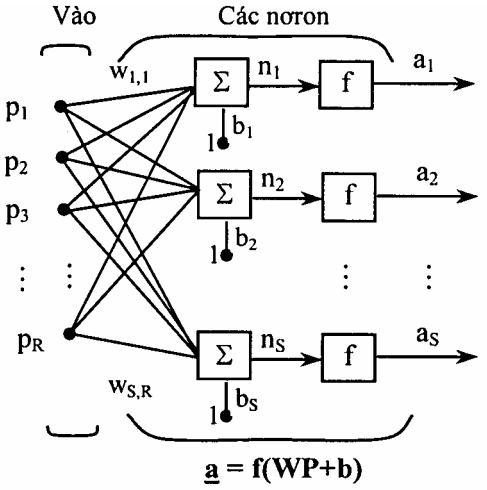
- Véc tơ vào **p** có R phần tử $\mathbf{p}^T = [p_1\ p_2 ...\ P_R]$.

- Véctơ vào **n** có s phần tử $\mathbf{n}^T = [n_1\ n_2 ...\ n_s]$.

- Véctơ vào **a** có s phần tử $\mathbf{a}^T = [a_1\ a_2 ...\ a_s]$.

Trong mạng này mỗi phần tử của véctơ vào **p** liên hệ với đầu vào mỗi nơron thông qua ma trận trọng liên kết **W**. Bộ cộng của nơron thứ i thu thập các trọng liên kết đầu vào và độ dốc để tạo thành một đầu ra vô hướng n;. Các n_i tập hợp với nhau tạo thành s phần tử của véctơ vào n. Cuối cùng ở lớp ra nơron ta thu được véctơ a gồm s phần tử.

Chú ý: Nhìn chung số đầu vào của một lớp khác với số nơron, tức là R ≠ S. Trong một lớp, không bắt buộc phải có số đầu vào bằng số nơron của nó.



Hình 3.10. Cấu trúc mạng nơron 1

Ta có thể thiết lập lớp đơn của các nơron có các hàm chuyển khác nhau một cách dễ dàng bởi lẽ hai mạng được đặt song song. Tất cả các mạng có thể có chung đầu vào và mỗi mạng có thể thiết lập một vài đầu ra.

Các phần tử của véctơ đầu vào được đưa vào mạng thông qua ma trận trọng **W**, với:

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1R} \\ w_{21} & w_{22} & \cdots & w_{2R} \\ \cdots & & & \\ w_{S1} & w_{S2} & \cdots & w_{SR} \end{bmatrix}$$

Trong đó: Chỉ số hàng trong các phần tử của ma trận **W** cho biết nơron nơi đến còn chỉ số cột cho biết nơi xuất phát của trọng liên kết. Ví dụ: w_{12} nói lên sự có mặt của tín hiệu vào từ phần tử thứ hai đến nơron thứ nhất với trọng liên kết là w_{12}.

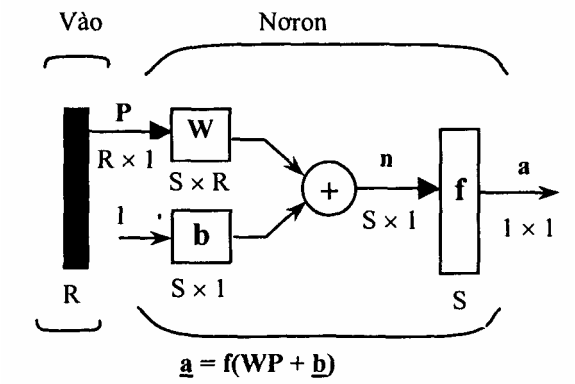
Tương tự như đã trình bày với 1 nơron, để đơn giản ta ký hiệu mạng một lớp gồm **S** nơron, **R** đầu vào như hình vẽ 3.11.Trong đó: véctơ vào **P** có kích thước R, ma trận trọng liên kết **W** có kích thước S *x* R còn a và b là các véctơ có kích thước S. Như chúng ta đã biết, một lớp mạng bao gồm ma trận trọng liên kết, toán tử nhân, véctơ độ dốc b, bộ tổng và hộp hàm truyền.

3.3.2. Mạng nhiều lớp

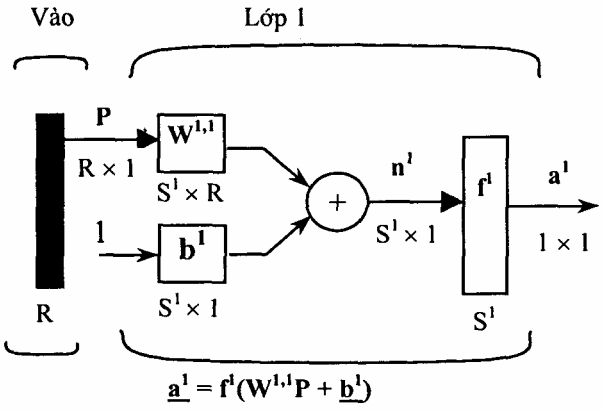
a/ Ký hiệu quy ước cho một lớp mạng

Để khảo sát mạng nhiều lớp trước hết chúng ta cần đưa ra các ký hiệu quy ước cho một lớp mạng. Đặc biệt ta cần phải phân biệt sự khác nhau giữa ma trận trọng liên kết ở đầu vào và các ma trận trọng liên kết giữa các lớp và nắm vững ký hiệu nguồn và đích của ma trận trọng liên kết.

Ta gọi ma trận trọng liên kết nối với đầu vào là các trọng vào (input weights) và các ma trận đến từ lớp ra là trọng liên kết lớp (layer weights). Ta sẽ dùng các chỉ số viết bên trên để phân biệt nguồn (chỉ số thứ hai) và đích (chỉ số thứ nhất) cho các trọng liên kết và các phần tử khác của mạng.



Hình 3.11. Ký hiệu mạng R đầu vào và S nơron



Hình 3.12. Ký hiệu một lớp mạng

Để minh hoạ, ta xét một lớp mạng có nhiều đầu vào như hình 3.12. Trong đó R là số phần tử lớp vào và S^1 là số nơron của lớp 1. Ta thấy ma trận trọng liên kết với véctơ vào **P** là ma trận trọng vào ($\mathbf{IW}^{1,1}$) có nguồn là 1 (chỉ số thứ 2) và đích là 1 (chỉ số thứ nhất). Đồng thời các phần tử của 1 lớp như độ dốc, tín hiệu vào hàm chuyển, đầu ra có chỉ số viết trên là 1 để nói rằng chúng được liên kết với lớp thứ nhất (b^1, n^1, a^1). Ở phần sau ta sẽ sử dụng ma trận trọng liên kết lớp (**LW**) giống như ma trận trọng vào (**IW**).

Với một mạng cụ thể có ma trận trọng $\mathbf{IW}^{1,1}$ được ký hiệu:

$$\mathbf{IW^{1,1}} \rightarrow \mathbf{net.IW\{1, 1\}}$$

Như vậy, ta có thể viết ký hiệu để thu được mạng nhập vào cho hàm chuyển như sau:

```
n{1} = net.IW{1, 1}*p + net.b{1};
```

Một mạng nơron có thể có một vài lớp. Mỗi lớp có ma trận trọng liên kết W, véctơ độ dốc b và đầu ra a. Để phân biệt các ma trận trọng liên kết véctơ vào cho mỗi lớp mạng trong sơ đồ, ta thêm con số chỉ lớp viết ở phía trên cho biến số quan tâm.

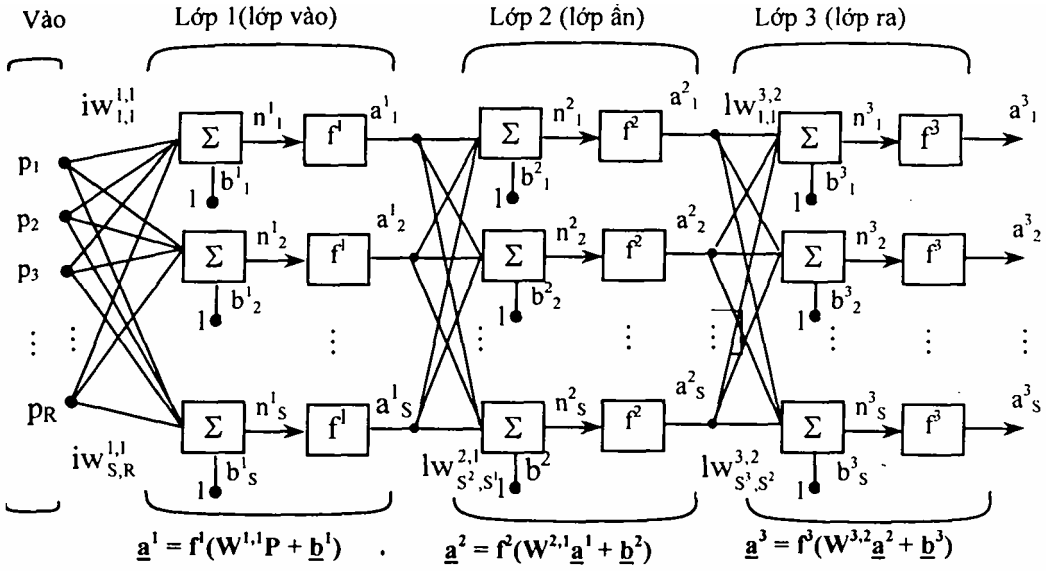
Hình 3.13 là ký hiệu sơ đồ mạng 3 lớp. Trong đó có R^1 đầu vào, S^1 nơron ở lớp 1, S^2 nơron ở lớp 2... Thông thường, các lớp khác nhau có số nơron khác nhau.

Chú ý rằng đầu ra của mỗi lớp trung gian là đầu vào của lớp tiếp theo. Như vậy lớp 2 có thể được xem như mạng 1 lớp với S^1 đầu vào, S^2 nơron và S^2 x S^1 trọng liên kết của ma trận W^2. Đầu vào của lớp 2 là véctơ $\mathbf{a^1}$, đầu ra là véctơ $\mathbf{a^2}$. Khi đã có ký hiệu của tất cả các véctơ và ma trận của lớp 2 ta có thể coi nó như là mạng 1 lớp. Cách tiếp cận này được dùng cho một lớp bất kỳ của mạng. Các lớp của mạng nhiều lớp đóng vai trò khác nhau. Lớp cuối cùng là kết quả ở đầu ra của mạng, được gọi là lớp ra. Tất cả các lớp khác được gọi là lớp ẩn. Mạng 3 lớp ở trên có 1 lớp ra (lớp 3) và 2 lớp ẩn (lớp 1 và lớp 2).

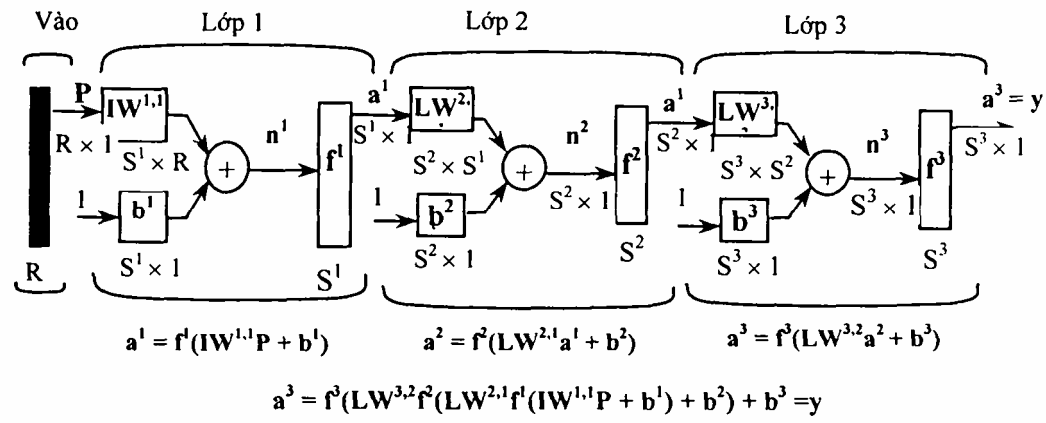
(Một vài tài liệu coi lớp vào như là lớp thứ tư ở đây ta không sử dụng quan điểm này).

Đối với mạng 3 lớp ta cũng có thể sử dụng ký hiệu tắt để biểu diễn (hình 3.14). Mạng nhiều lớp rất mạnh, ví dụ có mạng 2 lớp, trong đó lớp 1 có hàm chuyển sigmoid, lớp 2 có hàm chuyên linear có thể được huấn luyện để làm xấp xỉ một hàm bất kỳ (với số điểm gián đoạn có hạn chế). Loại mạng 2 lớp này sẽ được sử dụng rộng rãi ở chương 5 (mạng lan truyền ngược).

Trong đó a^3 là đầu ra của mạng, ta ký hiệu đầu ra này là y. Ta sẽ sử dụng ký hiệu này để định rõ đầu ra của mạng nhiều lớp.



Hình 3.13. Cấu trúc mạng nơron 3 lớp

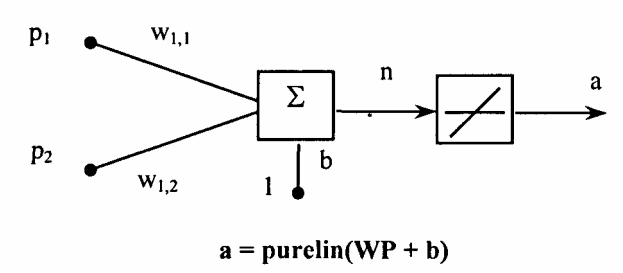


Hình 3.14. Ký hiệu tắt của mạng nơron 3 lớp

3.4. CẤU TRÚC DỮ LIỆU VÀO MẠNG

Để mô phỏng mạng nơron ta cần phải định rõ khuôn dạng của cấu trúc dữ liệu được dùng trong mạng. Dữ liệu đưa vào mạng được biểu diễn dưới 2 dạng cơ bản: một dạng xuất hiện đồng thời (tại cùng một thời điểm hoặc chuỗi thời điểm cụ thể) và một dạng xuất hiện liên tiếp theo thời gian. Đối với véctơ vào đồng thời, ta không cần quan tâm đến thứ tự của các phần tử, kiểu dữ liệu này được áp dụng cho mạng tĩnh. Đối với kiểu véctơ vào nối tiếp thì thứ tự xuất hiện của các phần tử véctơ rất quan trọng, nó được áp dụng

cho mạng động.



Hình 3.15. Một nơron với 2 đầu vào

3.4.1. Mô tả véctơ vào đối với mạng tĩnh

Đối với mạng tĩnh (không có phản hồi và trễ), ta không cần quan tâm tới việc có hay không véctơ vào xuất hiện trong một chuỗi thời điểm cụ thể, vì vậy ta có thể xem như các đầu vào là đồng thời. Trong phép cộng, ta giải quyết bài toán đơn giản bằng tổng của mạng chỉ có một véctơ vào:

$$n = W_{1,1}*p_1 + W_{1,2}*p_2+b.$$

Ví dụ: Mạng truyền thẳng có 2 đầu vào (hình 3.15) với các thông số: W = [1 2] và b = [0]; tập dữ liệu mô phỏng mạng có 4 véctơ vào đồng thời (Q = 4):

$$P_1 = \begin{bmatrix}1\\2\end{bmatrix}, \quad P_2 = \begin{bmatrix}2\\1\end{bmatrix}, \quad P_3 = \begin{bmatrix}2\\3\end{bmatrix}, \quad P_4 = \begin{bmatrix}3\\1\end{bmatrix},$$

Các véctơ vào đồng thời được trình bày trong mạng như một ma trận đơn giản:

P = [1 2 2 3; 2 1 3 1];

Sau khi chạy mô phỏng ta thu được các giá trị ở đầu ra

$$a_1 = W_{1,1}*p_1 + W_{1,2}*p_2+b = 1 * 1 + 2 * 2 + 0 = 5$$

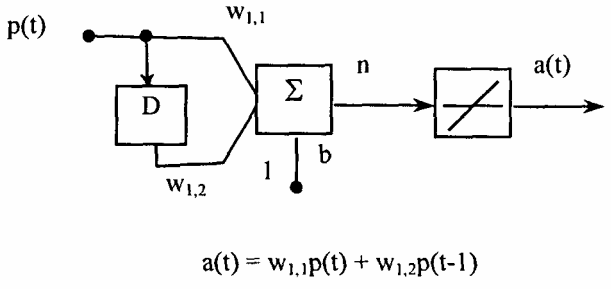
$$a_2 = W_{1,1}*p_1 + W_{1,2}*p_2+b = 1 * 2 + 2 * 1 + 0 = 4$$

$$a_3 = W_{1,1}*p_1 + W_{1,2}*p_2+b = 1 * 2 + 2 * 3 + 0 = 8$$

$$a_4 = W_{1,1}*p_1 + W_{1,2}*p_2+b = 1 * 3 + 2 * 1 + 0 = 5$$

Vậy véctơ véctơ đầu ra là: A = [5 4 8 5].

Một ma trận đơn của véctơ đồng thời được đưa tới mạng và mạng đưa ra một ma trận đơn của véctơ đồng thời ở đầu ra. Kết quả tương tự như 4 mạng làm việc song song, mỗi mạng có một véctơ vào và 1 véctơ ra. Thứ tự của các véctơ vào không quan trọng do chúng không ảnh hưởng lẫn nhau.



Hình 3.16. Nơron có chứa khâu trễ

3.4.2. Mô tả véctơ vào liên tiếp trong mạng động

Khi mạng có chứa khâu trễ, ở đầu vào mạng thường sẽ có một chuỗi các véctơ vào mà chúng xuất hiện theo thứ tự thời gian nào đó. Để minh hoạ cho trường hợp này ta sử dụng một mạng đơn bao gồm một khâu trễ (hình 3.16). Ta đưa vào mạng gồm dãy liên tiếp các dữ liệu vào thì mạng sinh ra một mảng bao gồm chuỗi liên tiếp các dữ liệu ra. Chú ý rằng thứ tự của dữ liệu vào rất quan trọng khi chúng được đưa vào như một sự nối tiếp. Trong trường hợp này dữ liệu ra thu được bằng cách nhân dữ liệu vào hiện thời với $w_{1,1}$, dữ liệu vào trước đó với $w_{1,2}$ rồi cộng kết quả lại nếu thay đổi thứ tự các dữ liệu vào nó có thể làm thay đổi những số thu được ở đầu ra.

Ví dụ: Mạng hình 3.16 có các thông số: W = [1 2]; b = 0; Chuỗi vào nối tiếp là:

p_1 = [1] p_2 = [2], p_3 = [3], p_4 = [4], được biểu diễn dưới dạng mảng:

P = {1 2 3 4}.

Sau khi chạy mô phỏng ta thu được một mảng dữ liệu ra với các phần tử có giá trị:

$a_1 = W_{1,1}*p_1+W_{1,2}*p_2=1*1+2*0+0=1$ (giá trị đầu vào 2 vẫn là 0)

$a_2 = W_{1,1}*p_1+W_{1,2}*p_2=1*2+2*1+0=4$ (giá trị đầu vào 2 là 1)

$a_3 = W_{1,1}*p_1+W_{1,2}*p_2=1*3+2*2+0=7$ (giá trị đầu vào 2 là 2)

$a_4 = W_{1,1}*p_1+W_{1,2}*p_2=1*4+2*3+0=10$ (giá trị đầu vào 2 là 3)

Vậy A = [1] [4] [7] [10].

3.4.3. Mô tả các dữ liệu vào đồng thời trong mạng động

Khi đưa vào mạng động đã xét ở trên một tập các dữ liệu đồng thời thay cho các dữ liệu liên tiếp, ta có thể thu được kết quả khác nhau hoàn toàn. Ví dụ có tập các dữ liệu vào đồng thời:

$P_1 = [1]$, $P_2= [2]$, $P_3 = [3]$, $P_4 = [4]$ được thiết lập theo mã sau:

P = [1 2 3 4];

Sau khi chạy mô phỏng với các dữ liệu vào đồng thời ta thu được:

A = [1 2 3 4].

Kết quả này giống như khi ta áp dụng đồng thời mỗi đầu vào tới một mạng riêng biệt và tính toán một đầu ra.

Chú ý: Một khi ta không ấn định bất kỳ điều kiện đầu nào cho mạng có trễ thì chúng được coi bằng zero. Trong trường hợp này đầu ra chỉ đơn giản là 1 nhân với đầu vào vì hàm trọng nhân với đầu vào hiện thời là 1.

Trong trường hợp đặc biệt, ta có thể cần phải mô phỏng đáp ứng của mạng với một vài chuỗi số khác nhau trong cùng một thời gian, ta cần đưa tới mạng với một tập đồng thời của chuỗi. Ví dụ ta cần đưa tới mạng hai dữ liệu liên tiếp sau:

$p_1(1) = [1]$, $p_1(2) = [2]$, $p_1(3) = [3]$, $p_1(4) = [4]$

$p_2(1) = [4]$, $p_2(2) = [3]$, $p_2(3) = [2]$, $p_2(4) = [1]$.

Đầu vào P cần phải là một mảng, trong đó mỗi phần tử của mảng bao gồm 2 phần tử liên tiếp mà chúng xuất hiện cùng một lúc.

P = {[1 4] [2 3] [3 2] [4 1]};

Chạy mô phỏng mạng:

A = sim(net,P);

Kết quả đầu ra của mạng sẽ là:

A = {[1 4] [4 11] [7 8] [10 5]} = {[a_{11} a_{21}] [a_{12} a_{22}] [a_{13} a_{23}] [a_{14} a_{24}] trong đó:

$$a_{11} = W_{1,1}.p_1 + W_{1,2}.p_2 + b = 1 * 1 + 2 * 0 + 0 = 1;$$

$$a_{21} = W_{1,1}.p_1 + W_{1,2}.p_2 + b = 1 * 4 + 2 * 0 + 0 = 4;$$

$$a_{12} = W_{1,1}.p_1 + W_{1,2}.p_2 + b = 1 * 2 + 2 * 1 + 0 = 4;$$

$$a_{22} = W_{1,1}.p_1 + W_{1,2}.p_2 + b = 1 * 3 + 2 * 4 + 0 = 11;$$

$$a_{13} = W_{1,1}.p_1 + W_{1,2}.p_2 + b = 1 * 3 + 2 * 2 + 0 = 7;$$

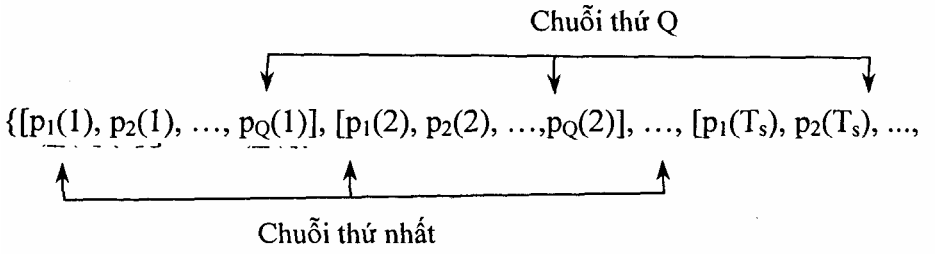
$$a_{23} = W_{1,1}.p_1 + W_{1,2}.p_2 + b = 1 * 2 + 2 * 3 + 0 = 8;$$

$$a_{14} = W_{1,1}.p_1 + W_{1,2}.p_2 + b = 1 * 4 + 2 * 3 + 0 = 7;$$

$$a_{24} = W_{1,1}.p_1 + W_{1,2}.p_2 + b = 1 * 1 + 2 * 2 + 0 = 8;$$

Ta có thể thấy cột đầu tiên của mỗi ma trận kết quả, chuỗi ra được tạo ra từ chuỗi vào đầu tiên mà chúng ta đã làm quen trong ví dụ trước. Cột thứ hai của mỗi ma trận kết quả chuỗi ra được tạo ra từ chuỗi vào thứ hai. Không có sự tương tác giữa hai chuỗi đồng thời. Nó giống như khi mỗi ứng dụng của các mạng riêng biệt được chạy song song.

Sơ đồ dưới đây chỉ ra khuôn dạng chung của đầu vào P khi ta có Q chuỗi vào đồng thời qua những bước thời gian Ts, nó bao hàm cả trường hợp khi có 1 véctơ vào. Mỗi phần tử của mảng là một ma trận của các véctơ đồng quy mà nó ứng với cùng một thời điểm cho mỗi chuỗi. Nếu có nhiều véctơ vào sẽ có nhiều hàng của ma trận trên mảng.



Trong mục này chúng ta đã áp dụng các nữ liệu vào liên tiếp và đồng thời cho mạng động.

Chú ý: ở mục 3.4.1 ta đã áp dụng dữ liệu vào đồng thời cho mạng tĩnh. Ta cũng có thể áp dụng dữ liệu vào liên tiếp cho mạng tĩnh, nó sẽ không làm thay đổi kết quả mô phỏng của mạng, nhưng nó có thể ảnh hưởng tới cách thức huấn luyện mạng.

3.5. HUẤN LUYỆN MẠNG

Trong phần này, chúng ta đề cập đến 2 kiểu huấn luyện mạng: Huấn luyện gia tăng (tiến dần) và huấn luyện theo gói. Đối với sự huấn luyện gia tăng, hàm trọng và độ dốc của mạng được cập nhật mỗi khi dữ liệu được đưa vào mạng. Đối với sự huấn luyện theo gói, hàm trọng và độ dốc chỉ được cập nhật sau khi tất cả các dữ liệu được đưa vào mạng.

3.5.1. Huấn luyện gia tăng

Sự huấn luyện gia tăng (huấn luyện tiến dần) có thể được áp dụng cho cả mạng tĩnh và mạng động. Tuy nhiên, trong thực tế nó được sử dụng nhiều hơn cho mạng động, *ví* dụ các bộ lọc thích nghi. Trong mục này, chúng ta sẽ giải thích sự huấn luyện gia tăng được thực hiện như thế nào trên mạng tĩnh và mạng động.

a/ Huấn luyện gia tăng đối với mạng tĩnh

Xét mạng tĩnh học, ta muốn huấn luyện nó gia tăng, sao cho hàm trọng và độ dốc của nó được cập nhật mỗi khi đầu vào có mặt. Trong trường hợp này chúng ta sử dụng hàm **"Adapt"** và ta coi các giá trị đầu vào và đích là các chuỗi nối tiếp.

Giả thiết ta muốn huấn luyện mạng để tạo ra hàm tuyến tính:

$$\mathbf{t = 2p_1 + P_2}$$

Các dữ liệu vào ban đầu được sử dụng:

$$p_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad p_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad p_3 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad p_4 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Đích của mạng là: $\mathbf{t_1 = [4]\ t_2 = [5]\ t_3 = [7]\ t_4 = [7]}$

Trước hết ta thiết lập mạng với những hàm trọng và độ dốc ban đầu bảng zero. Ta cũng đặt mức học xuất phát từ zero, để cho thấy hiệu ứng của sự huấn luyện gia tăng.

```
net = newlin([-1 1;-1 1], 1,0,0);
net.IW{1,1} = [0 0];
net.b{1} = 0;
```

Để huấn luyện gia tăng, ta cần coi các đầu vào và đích là các chuỗi:

P = {[1;2] [2;1] [2;3] [3;1]}

T = {4 5 7 7};

Như đã đề cập ở mục trước, đối với mạng tĩnh kết quả sự mô phỏng của mạng ở đầu ra liệu có giống như đầu vào đã được đưa ra như là một ma trận của véctơ đồng thời hay như là một mảng của các véctơ liên tiếp. Điều này là không đúng khi huấn luyện mạng. Tuy vậy khi sử dụng hàm **Adapt**, nếu một mảng các véctơ liên tục được đưa đến đầu vào thì hàm trọng được cập nhật như với mỗi đầu vào được đưa đến. Như chúng ta sẽ thấy ở phần sau, nêu một ma trận của véctơ đồng thời được đưa đến đầu vào thì hàm trọng chỉ được cập nhật sau khi tất cả các tín hiệu vào được đưa đến.

Để huấn luyện gia tăng ta sử dụng dòng lệnh:

[net,a,e,p,f] = adapt(net,P,T);

Đầu ra của mạng vẫn là zero bởi lẽ tốc độ học bằng zêro và hàm trọng không được cập nhật. Các giá trị sai lệch sẽ bằng các giá trị đích:

a = [0] [0] [0] [0]

e = [4] [5] [7] [7]

Nếu bây giờ ta đặt tốc độ học bằng 0, 1 ta có thể thấy mạng sẽ được điều chỉnh mỗi khi đầu vào có tín hiệu:

net,inputWeights{1,1}.learnParam.Ir=0.1;

net.biases{1,1}.learnParam.Ir=0.1;

[net,a,e,pf] = adapt(net,P,T);

a = [0] [2] [6.0] [5.8]

e = [4] [3] [1.0] [1.2]

Dữ liệu ra thứ nhất tương tự như dữ liệu ra với tốc độ học bằng 0, do không có sự cập nhật nào cho tới khi dữ liệu vào thứ nhất xuất hiện. Dữ liệu ra thứ hai là khác do hàm trọng đã được cập nhật. Các hàm trọng liên tục được sửa đổi theo mỗi sai lệch được tính toán. Nếu mạng có năng lực và tốc độ huấn luyện chọn hợp lý thì các sai lệch sẽ dần tiến tới zêro.

b/ Huấn luyện gia tăng đối với mạng động

Đối với mạng động, ta cũng có thể huấn luyện gia tăng (đây là kiểu huấn luyện chung nhất). Xét mạng tuyến tính với một trễ ở đầu vào mà ta đã đề cập ở phần trước. Ta cho giá trị ban đầu của hàm trọng bằng 0 và đặt tốc độ học là 0,1.

net = newlin([-l 1],1,[0 1], 0.1);

net.IW{1, 1} = [0 0];

net.biasconnect = 0;

Để huấn luyện gia tăng mạng này, ta biểu diễn dữ liệu vào và dữ liệu đích như là các phần tử của mảng.

Pi = {l};

P = {2 3 4};

T = {3 5 7};

Ở đây ta thử huấn luyện mạng thực hiện phép cộng dữ liệu vào hiện thời và dữ liệu vào trước để tạo ra dữ liệu ra hiện thời. Điều này giống như sự nối tiếp dữ liệu vào ta đã sử dụng ở ví dụ trước của sự sử dụng hàm **Sim**, Chỉ có điều chúng ta gán giới hạn đầu tiên trong sự nối tiếp như điều kiện ban đầu cho sự trì hoãn. Bây giờ ta có thể sử dụng hàm **Addapt** để huấn luyện mạng:

[net,a,e,pf] = adapt(net,P,T,Pi);

a = [0] [2.4] [7.98]

e = [3] [2.6] [-0.98]

Dữ liệu ra đầu tiên bằng 0 do hàm trọng chưa được cập nhật. Hàm trọng sẽ thay đổi tại mỗi bước thời gian kế tiếp.

3.5.2 Huấn luyện mạng theo gói

Huấn luyện theo gói trong đó các hàm trọng và độ dốc chỉ được cập nhật sau khi tất cả các dữ liệu vào và đích đã được đưa tới, có thể được áp dụng cho cả mạng tĩnh và mạng động. Trong mục này, chúng ta sẽ thảo luận kỹ cả hai loại mạng này.

a/ Huấn luyện theo gói đối với mạng tĩnh

Để huấn luyện theo gói, ta có thể sử dụng hàm adapt hoặc hàm train, song nói chung trai là tuỳ chọn tốt nhất, vì nó đặc trưng cho sự truy nhập có hiệu quả hơn của giải thuật huấn luyện. Như vậy, sự huấn luyện gia tăng chỉ có thể làm việc với hàm adapt, còn hàm train chỉ có thể thực hiện để huấn luyện theo gói. Trước hết ta hãy bắt đầu huấn luyện theo gói đối với mạng tĩnh đã đề cập trong ví dụ trước, tốc độ hoạc đặt bằng 0,1.

net = newlin([-1 1;-1 1],1,0,0.1);

net.IW{1,1} = [0 0];

net.b{1} = 0;

Để huấn luyện theo gói mạng tĩnh các véc tơ dữ liệu vào cần được đặt trong ma trận của các véc tơ đồng thời.

P = [1 2 2 3; 2 1 3 1];

T = [4 5 7 71;

Khi ta gọi lệnh **Adapt**, nó sẽ kéo theo **trains** (là các hàm thích nghi mặc định của mạng tuyến tính) và learnwh (là các hàm huấn luyện mặc định của hàm trọng và độ dốc).

[net,a,e,pf] = Adapt(net,P,T);

a = 0 0 0 0

e - 4 5 7 7.

Chú ý rằng tất cả các đầu ra của mạng đều bằng zero, bởi lẽ các hàm trọng chưa được cập nhật cho tới khi tất cả tập hợp huấn luyện được đưa tới. Nếu hiển thị trên màn hình ta thấy:

»net.IW**{1,l}**

ans = 4.9000 4.1000

»net.b{1}

ans =

2.3000.

Đây là sự khác nhau về kết quả ta nhận được sau một lần thực hiện hàm **Adapt** với sự cập nhật gia tăng. Bây giờ chúng ta hãy thực hiện việc huấn luyện theo gói sử dụng hàm **train**. Do luật Widrow-Hoff có thể sử dụng cho kiểu gia tăng và kiểu gói, nó có thể được gọi bằng **Adapt** hoặc **train**. Có một vài thuật toán huấn luyện chỉ có thể sử dụng trong kiểu gói (ví dụ Levenberg-Marquardt) và do đó các thuật toán này chỉ có thể gọi bằng lệnh **train**. Mạng

sẽ được cài đặt bằng cách tương tự.

net = newlin([-1 1;-1 1],0,0.1);

net.IW**{1,1} = [0 0];**

net.b{1} = 0;

Trong trường hợp này véc tơ dữ liệu vào có thể đặt dưới dạng ma trận của các véc tơ đồng thời (concurrent vectors) hoặc dưới dạng mảng của các véc tơ liên tiếp. Trong **Train**, mảng của các véc tơ liên tiếp bất kỳ được chuyển đổi thành ma trận của các véc tơ đồng thời. Đó là do mạng là tĩnh và do lệnh **train** luôn luôn hoạt động theo kiểu gói.

P = [1 2 2 3; 2 1 3 1];

T = [4 5 7 7];

Bây giờ ta sẵn sàng để huấn luyện mạng. Ta sẽ huấn luyện nó chỉ trong một kỳ vì ta chỉ sử dụng một lần hàm **Adapt**. Hàm huấn luyện mặc định cho mạng tuyến tính là **train** và hàm huấn luyện mặc định cho hàm trọng và độ dốc là learnwh, vì vậy ta có thể nhận được các kết quả tương tự kết quả sử dụng **Adapt** trong ví dụ trước, khi ta sử dụng hàm thích nghi mặc định là trains.

net.inputWeights{1,1}.learnParam.Ir = 0,1;

net.biases{l}.learnParam.Ir = 0,1;

net.trainparam.epochs : 1;

net = train(net,P,T);

Nếu cho hiển thị hàm trọng sau một kỳ huấn luyện ta thấy:

»net.IW**{1,1}**

ans = 4.9000 4.1000

»net.b{1}

ans =

2.3000.

Kết quả này tương tự với kết quả huấn luyện theo gói sử dụng **Adapt**. Đối với mạng tĩnh, hàm **Adapt** có thể thực hiện sự huấn luyện gia tăng hoặc theo gói tuỳ thuộc vào khuôn dạng dữ liệu vào. Nếu dữ liệu được đưa tới mạng dưới dạng ma trận của các véc tơ đồng thời thì huấn luyện theo gói sẽ xảy ra. Nếu dữ liệu được đưa tới dưới dạng chuỗi thì huấn luyện gia tăng sẽ xảy ra. Điều này không đúng vơi hàm **train**, nó luôn luôn huấn luyện theo gói mà không phụ thuộc vào khuôn dạng của dữ liệu vào.

b/ Huấn luyện theo gói đối với mạng động

Huấn luyện mạng tĩnh học tương đối dễ hiểu. Nếu ta sử dụng thun để huấn luyện mạng theo gói và dữ liệu vào được chuyển đổi thành véc tơ đồng thời (các cột của ma trận) cho dù khuôn dạng trước đây của chúng là chuỗi. Nếu ta sử dụng **Adapt** thì khuôn dạng dữ liệu vào quyết định phương pháp huấn luyện. Nếu khuôn dạng dữ liệu vào là chuỗi thì mạng được huấn luyện kiểu gia tăng, nếu khuôn dạng dữ liệu vào là véc tơ đồng thời thì mạng được huấn luyện kiểu gói.

Đối với mạng động, kiểu huấn luyện theo gói chỉ được thực hiện với hàm **train**. Để minh hoạ điều này ta lại xét mạng tuyến tính có trễ. Ta sử dụng tốc độ học là 0,02 để huấn luyện. Khi sử dụng giải thuật giảm độ dốc ta chọn tốc độ học cho kiểu huấn luyện gói nhỏ hơn kiểu huấn luyện gia tăng.

Ví dụ:

```
net = newlin([-1 1],1,[0 1],0.02);
net.IW{1,1}=[0 0];
net.biasConnect 0;
net.trainparam.epochs = 1;
Pi = {1};
P = {2 3 4};
T = {3 5 6};
```

Ta muốn huấn luyện mạng với chuỗi tương tự như đã sử dụng cho sự huấn luyện gia tăng trước đây thế nhưng thời điểm cần thiết để cập nhật các hàm trọng chỉ xảy ra sau khi tất cả dữ liệu vào được áp dụng (kiểu gói). Mạng được coi như tuần tự vì đầu vào là tuần tự, song các hàm trọng được cập nhật theo kiểu gói.

```
net=train(net,P,T,Pi);
»net.IW{1,1}
ans = 0.9000   0.6200.
```

Kết quả này khác với kết quả ta đã thu được bằng huấn luyện gia tăng, ở đó các hàm trọng được cập nhật 3 lần trong thời gian một tập huấn luyện. Đối với huấn luyện theo gói các hàm trọng chỉ được cập nhật một lần trong một khóa huấn luyện.