TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
KHOA ĐIỆN TỬ

BÀI GIẢNG:

ĐIỀU KHIỂN THÔNG MINH

BIÊN SOẠN:

NGUYỄN VIỆT HÙNG
NGUYỄN TẤN ĐỜI
TRƯƠNG NGỌC ANH
TẠ VĂN PHƯƠNG

TP HỒ CHÍ MINH, NĂM 2008

LỜI NÓI ĐẦU

Tài liệu được soạn dùng cho ngành sinh viên bậc Đại học, ngành Kỹ thuât Điện-Điện tử nhằm trang bị kiến thức ban đầu về Kỹ thuật điều khiển thông minh cho sinh viên các năm cuối.

Tài liệu được biên soạn theo hướng dễ hiểu, chú trọng đến các ý tưởng cốt lõi, trình bày các điểm tổng quát nhất, chưa đi sâu đến các phương pháp tính toán phức tạp.

TÀI LIỆU THAM KHẢO CHÍNH

FUZZY AND NEURAL CONTROL
DISC Course Lecture Notes (September 2004)
ROBERT BABUSKA
Delft Center for Systems and Control

Nhóm tác giả mong rằng tài liệu này sẽ giúp sinh viên tiếp cận nhanh và ứng dụng được các công nghệ điều khiển mới vào cuộc sống.

Nhóm các tác giả

MỤC LỤC

MỞ ĐẦU

Chương trình bày phần mở đầu ngắn về mục đích của sách và giới thiệu tóm tắt các chương. Đồng thời cung cấp thông tin về kiến thức cần trang bị cho người đọc. Cuối cùng, giới thiệu phần hỗ trợ từ các trang WEB và từ MATLAB.

1. Hệ thống điều khiển truyền thống

Lý thuyết điều khiển truyền thống dùng các mô hình toán học như phương trình vi phân và phương trình sai phân, theo đó các phương pháp và thủ tục thiết kế phân tích và kiểm nghiệm hệ thống điều khiển đã được phát triển. Tuy nhiên, các phương pháp này chỉ ứng dụng được trong một lớp nhỏ các mô hình (mô hình tuyến tính và một số dạng đặc biệt của mô hình phi tuyến) và thường không ứng dụng được nếu không tìm ra được mô hình cũa đối tượng hay quá trình điều khiển. Ngay khi có được mô hình chi tiết trên nguyên tắc thì vẫn chưa có được phương pháp thiết kế nhanh và luôn cần đến việc mô hình hóa tỉ mỉ, nên cần phát triển các hướng khác trong thiết kế.

2. Hệ điều khiển thông minh

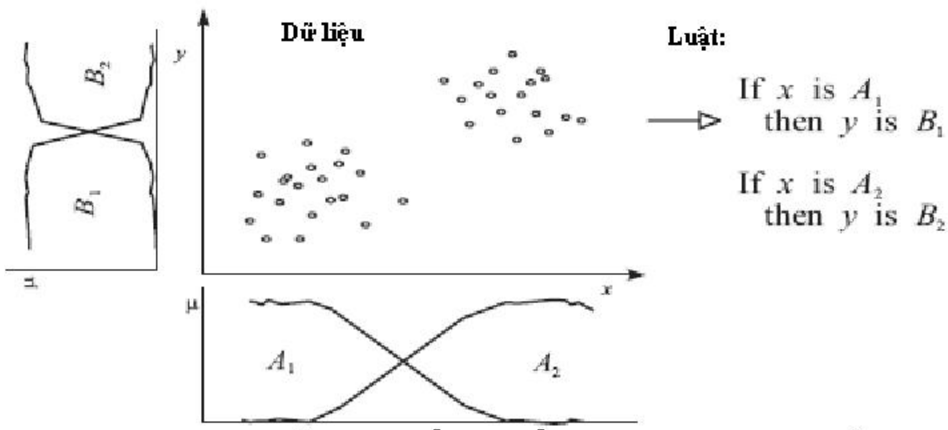
Thuật ngữ " Điều khiển thông minh" đã được giới thiệu trong khoảng ba thập niên với các phương pháp điều khiển có mục tiêu tham vọng hơn so với các hệ thống truyền thống. Trong khi hệ thống truyền thống thường cần các chi tiết dù nhiều dù ít về quá trình điều khiển thì hệ thống điều khiển thông minh có thể điều khiển một cách tự chủ các hệ thống phức tạp, các quá trình chưa được hiểu biết nhiều thí dụ như về mục tiêu điều khiển. Hệ thống này còn hoạt động được khi hệ thống có sự thay đổi về tham số hay môi trường điều khiển, thông qua quá trình học từ kinh nghiệm, tiếp thu và tổ chức kiến thức về môi trường xung quanh và hành vi sắp tới của hệ thống. Các mục tiêu đầy tham vọng này, xuất phát từ mong muốn bắt chước khả năng tuyệt vời của não bộ con người, mà thực ra cho đến giờ này thì chưa có hệ thống điều khiển thông minh nào là có thể đạt tới được. Hiện này, ý niệm "thông minh" thường được dùng cho để chỉ một số kỹ thuật có cội nguồn là lĩnh vực trí tuệ nhân tạo (artificial intelligence AI), có mục tiêu là bắt chước một số phần tử cơ bản của trí tuệ như lý luận (reasoning), học (learning), v.v,.. Trong đó phải kể đến mạng nơrôn nhân tạo, hệ chuyên gia, hệ logic mờ, mô hình định tính, thuật toán di truyền và nhiều tổ hợp từ các phương pháp này. Trong một số trường hợp, các kỹ thuật này đã thực sự đóng góp cho hệ thống một số khả năng thông minh, còn các trường hợp khác thì chỉ đơn thuần là phương tiện biểu diễn các luật điều khiển phi tuyến, mô hình của quá trình điều khiển hay các yếu tố bất định. Trường hợp sau tuy không đóng góp một cách rõ ràng vào mức độ thông minh của hệ thống, nhưng các phương pháp trên vẫn rất hữu ích. Chúng đã làm phong phú hóa lĩnh vực điều khiển thông qua các sơ đồ biểu diễn khác nhằm có được các thông tin đặc thù từ đối tượng điều khiển mà các phương pháp truyền thống không thể có được trên cơ sở của hệ phương trình vi phân và sai phân. Tài liệu này quan tâm đến hay công cụ quan trọng là hệ thống điều khiển mờ và mạng nơrôn. Điều khiển mờ là một thí dụ về các biểu diễn kiến thức con người qua các luật cùng quá trình diễn dịch tương ứng. Mạng nơrôn nhân tạo có thể thực hiện được tác động học phức tạp và nhiệm vụ thích ứng bằng cách bắt chước chức năng của hệ thống nơrôn sinh học.

Mục đích của phần này là giới thiệu ngắn về hai lĩnh vực này cùng với nguyên lý cơ bản của thuật toán di truyền.

3. Tổng quan về hệ thống điêu khiển

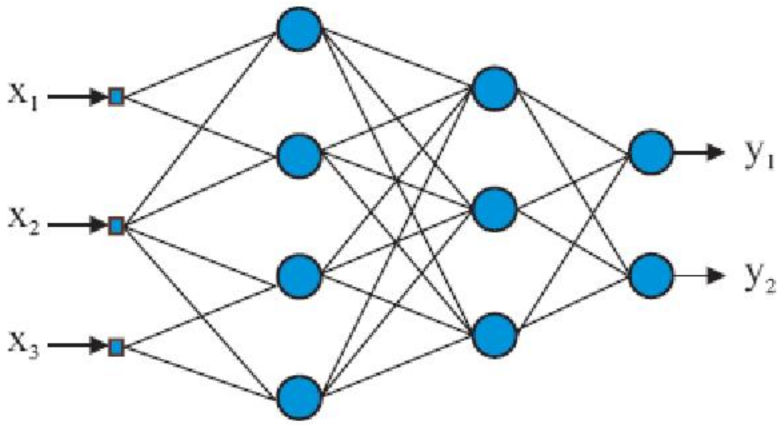
Hệ logic mờ (Fuzzy logic) mô tả quan hệ dựa trên luật nếu–thì (if–then rules), thí dụ như " *nếu* mở van nóng *thì* nhiệt độ tăng". Sự nhập nhằng (không xác định) trong định nghĩa của các thừa số ngôn ngữ (thí dụ, *nhiệt độ cao*) được biểu diễn thông qua *tập mờ*, là tập có các biên chồng khớp, xem hình 1.1. Theo ý nghĩa của tập mờ, thì một miền phần tử có thể đồng thời nằm trong nhiều tập (với các cấp độ tham gia khác nhau). Thí dụ $t = 20^\circ$C nằm trong tập nhiệt độ *Cao* có hàm thành viên là 0.4 và trong tập nhiệt độ *Trung bình* với hàm thành viên là 0.2. Sự thay đổi từ hàm thành viên sang không tham gia cho một kết quả suy diễn mịn dùng luật mờ nếu-thì; thực ra là một dạng nội suy.

Hệ logic mờ thích hợp để biêu diễn kiến thức định tính, có thể từ chuyên gia (trong hệ điều khiển mờ dùng nền tri thức) hay có thể lấy tự động từ dữ liệu (quy nạp, học). Trường hợp này thuật toán xâu chuỗi mờ thường được dùng để phân chia dữ liệu thành nhóm các đối tượng giống nhau. Từ đó, tìm được tập mờ và các luật nếu-thì cho các phân hoạch như mô tả ở hình 1.2. Phương pháp cho số lượng lớn các dữ liệu nhiều chiều được làm gọn, tạo ra các tóm tắt định tính. Nhằm gia tăng tính mềm dẽo cùng khả năng biểu diễn, có thể tìm được mô hình hồi qui từ phần hệ quả của luật (thường được gọi là hệ mờ Takagi–Sugeno).



Hình 1.1: Phép xâu chuỗi mờ có thể dùng để rút ra các luật định tính nếu -thì từ dữ liệu số học

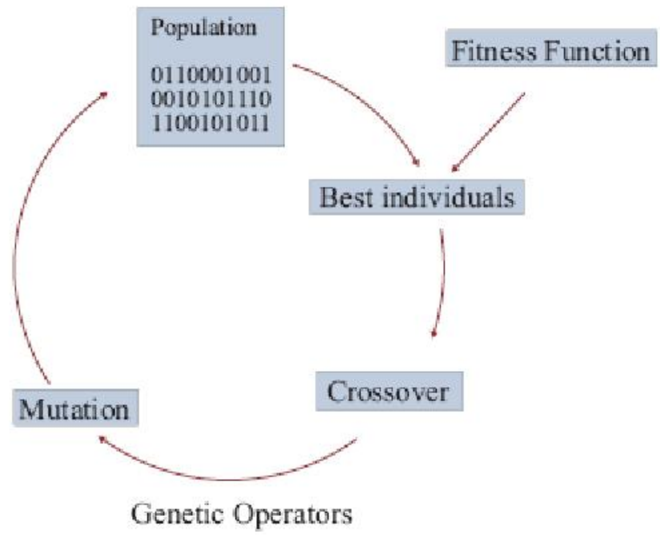
Mạng nơrôn nhân tạo (Artificial Neural Networks) là các mô hình đơn giản bắt chước chức năng của hệ nơrôn sinh học. Trong hệ logic mờ, thông tin được biểu diễn một cách tường minh theo dạng nếu-thì, còn trong mạng nơrôn, thông tin này được 'mã hóa' một cách không tường minh thành các thông số mạng. Khác với các kỹ thuật dùng nền tri thức (knowledge-based techniques), trong mạng không cần có kiến thức ẩn nào khi ứng dụng. Ưu điểm lớn nhất là khả năng học các quan hệ chức năng phức tạp bằng cách tổng quát hóa từ một lượng giới hạn của dữ liệu huấn luyện. Mạng nơrôn hiện có thể dùng làm mô hình (dạng hộp đen) cho hệ phi tuyến, đa biến tĩnh và động và có thể được huấn luyện dùng chính tập dữ liệu vào-ra quan sát được từ hệ thống.



Hình 1.3: Mạng nơrôn nhiều lớp

Hình 1.3 trình bày dạng mạng truyền thẳng thường gặp, gồm nhiều lớp chứa nhiều phần tử xử lý đơn giản được gọi là nơrôn, liên kết nối thông qua các trọng lượng chỉnh định được. Thông tin có được từ ánh xạ vào-ra của mạng được lưu trữ trong các trọng lượng này. Ngoài ra còn có các kiến trúc mạng khác, như dạng mạng nhiều lớp có phản hồi, mạng Hopfield và mạng tự tổ chức. Mạng nơrôn và hệ mờ thường có thể kết hợp trong hệ nơrôn-mờ (*neuro-fuzzy*) nhằm kết hợp một cách hiệu quả kỹ thuật dùng luật định cùng với thuật học từ dữ liệu.

Thuật toán di truyền (Genetic algorithms) là kỹ thuật tối ưu hóa ngẫu nhiên dựa trên thuyết tiến hóa và khả năng tồn tại của tự nhiên. Các nghiệm của bài toán được mã hóa thành chuỗi nhị phân hay thành các số thực. Tính khớp (fitness) về chất lượng, tính năng của các đáp số riêng biệt được ước lượng qua các hàm khớp (fitness function), được định nghĩa từ ngoài do người dùng hay từ các thuật toán cấp cao hơn. Cá thể khớp nhất trong trong nhóm (population) các nghiệm được sản sinh ra (reproduced) dùng các toán tử di truyền như trao đổi chéo (crossover) và đột biến (mutation). Theo hướng này thì có được một thế hệ mới các cá thể khớp nhất và toàn chu kỳ lại được khởi động lại (xem hình 1.4). Thuật toán di truyền đã được chứng tõ là hiệu quả trong quá trình tìm kiếm trong không gian nhiều chiều và được ứng dụng trong nhiều lĩnh vực, bao gồm việc tối ưu hóa cấu trúc bộ điều khiển, tinh chỉnh tham số trong hệ điều khiển phi tuyến, v.v,… Trong giáo trình này, ta chưa bàn đến thuật toán di truyền.



Hình 1.4: Thuật toán di truyền dựa trên quá trình mô phỏng giản đơn chu kỳ tiến hóa tự nhiên

4. Tổ chức của tài liệu

Tài liệu được tổ chức thành tám chương. Chương 2 trình bày nguyên lý cơ bản của lý thuyết tập mờ. Chương 3 giới thiệu các dạng hệ mờ khác nhau cùng ứng dụng trong mô hình hệ thống động. Kỹ thuật tập mờ rất hữu ích khi phân tích dữ liệu và nhận dạng mẫu. Tiếp đến, chương 4 giới thiệu các ý niệm cơ bản về phương pháp xâu chuỗi mờ (fuzzy clustering), được dùng trong kỹ thuật kiến tạo mô hình mờ từ dữ liệu. Các kỹ thuật kiến trúc dùng dữ liệu được đề cập trong chương 5. Bộ điều khiển có thể được thiết kế không cần mô hình đối tượng. Chương 6 đề cập đến các bộ điều khiển mờ không cần mô hình đối tượng trên cơ sở biến ngôn ngữ. Chương 7, giải thích các thuật ngữ cùng kiến trúc và việc huấn luyện mạng nơrôn nhân tạo. Các mô hình nơrôn và mờ có thể dùng trong thiết kế điều khiển hay dùng như một phần của các sơ đồ điều khiển có dùng mô hình như giới thiệu trong chương 8.

Mong muốn của tác giả là giới thiệu các thông tin mới (kỹ thuật mờ và mạng nơrôn) mà không cần có kiến thức tiên quyết để hiểu được giáo trình. Tuy nhiên, độc giả cần có kiến thức vè toán giải tích (hàm đơn và đa biến), đại số tuyến tính (hệ phương trình tuyến tính, nghiệm bình phương tối thiểu) và kiến thức về điều khiển và hệ thống (hệ động, phản hồi trạng thái, điều khiển PID, phương pháp tuyến tính hóa).

5. Các hỗ trợ từ WEB và Matlab

Tư liệu trong sách được cung cấp từ trang Web chứa các thông tin của bài giảng 'Knowledge-Based Control Systems' (SC4080) tại Delft University of Technology, cùng một số tư liệu download (MATLAB tools and demos, tóm lược bài giảng, các thí dụ). Địa chỉ (http://dcsc.tudelft.nl/˜sc4080). Sinh viên học lớp này được phép (và khuyến khích) mượn phần MATLAB Classroom Kit dùng cho máy tính tại nhà riêng trong thời gian theo học.

6. Tài liệu cần đọc

- Harris, C.J., C.G. Moore and M. Brown (1993). *Intelligent Control, Aspects of Fuzzy Logic and Neural Nets*. Singapore: World Scientific.
- Haykin, S. (1994). *Neural Networks*. New York: Macmillan Maxwell International.

- Jang, J.-S.R., C.-T. Sun and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing; a Computational Approach to Learning and Machine Intelligence*. Upper Saddle River: Prentice-Hall.
- Klir, G.J. and B. Yuan (1995). *Fuzzy sets and fuzzy logic; theory and applications*. Prentice Hall.
- Passino, K. M. and S. Yurkovich (1998). *Fuzzy Control*. Massachusetts, USA: Addison-Wesley.
- Zurada, Jacek M., Robert J. Marks II and Charles J. Robinson (Eds.) (1994). *Computational Intelligence: Imitating Life*. Piscataway, NJ: IEEE Press

7. Lời cảm tạ

Tác giả hết sức cảm ơn các đồng nghiệp đã đọc bản thảo và đóng góp ý kiến, cũng như ý kiến phản hồi của nhiều bạn sinh viên đã giúp cải thiện được tài liệu.

CHƯƠNG HAI: TẬP MỜ VÀ CÁC QUAN HỆ

Chương cung cấp phần mở đầu về tập mờ, quan hệ mờ, và các toán tử trong tập mờ. Để hiểu rõ thêm, tìm đọc (Klir and Folger, 1988; Zimmermann, 1996; Klir and Yuan, 1995).

Zadeh (1965) giới thiệu lý thuyết về tập mờ như một chuyên ngành toán học, cho dù các ý tưởng này đã được nhiều nhà luận lý và triết gia thừa nhận (Pierce, Russel, Łukasiewicz,v.v,..). Phần tổng quan dễ hiểu có thể tìm trong "Readings in Fuzzy Sets for Intelligent Systems", Prade và Yager (1993), nhà xuất bản Dubois. Các hướng nghiên cứu sâu về tập mờ bắt đầu từ thập niên bảy mươi của thế kỷ trước với nhiều ứng dụng trong điều khiển và các chuyên ngành kỹ thuật khác.

1. Tập mờ

Trong lý thuyết về tập bình thường, tập thực (không mờ), các phần tử có thể nằm hoàn toàn hay không nằm hoàn toàn trong tập này. Nhắc lại, hàm thành viên $\mu_A(x)$ của x trong tập truyền thống A, là tập con của vũ trụ X, thì được định nghĩa là:

$$\mu_A(x) = \begin{cases} 1, & x \in A, \\ 0, & x \notin A, \end{cases} \tag{2.1}$$

Điều này có nghĩa là phần tử có thể là thành viên của tập A ($\mu_A(x) = 1$) hay không ($\mu_A(x) = 0$). Việc phân lớp chặc chẽ này thường dùng trong toán học và các khoa học có dùng các định nghĩa chính xác. Lý thuyết về tập thực (tập thông thường) bổ sung thêm phần logic hai giá trị, nhằm trình bày vấn đề là đúng hay sai. Logic toán học thường nhấn mạnh đến việc giữ gìn giá trị chuẩn và đúng với mọi diễn đạt, trong khi trong cuộc sống thực và trong các bài toán kỹ thuật, thì lại có yêu cầu giữ gìn thông tin từ tình huống. Trong những trường hợp này, thì không nhất thiết là phải xác định rõ là phần tử phụ thuộc hay không phụ thuộc vào tập.

Thí dụ, nếu tập A biểu diễn số máy PC quá mắc so với sinh viên, thì tập này không có biên rõ ràng được. Dĩ nhiên, ta có thể nói giá PC là \$2500 là quá đắc, nhưng các giá PC là \$2495 hay \$2502 thì sao? Giá các PCs có là quá đặc hay không? Như thế, biên có thể được xác định là trên ngưỡng này thì là giá đắc cho các sinh viên trung bình, thí dụ \$2500, và dưới ngưỡng này là không đắc, thí dụ \$1000. Giữa các biên này, ta còn có giá khác không thể nói rõ ràng là quá đắc hay không. Trong ngưỡng này, có thể dùng thang điểm đánh giá các máy có giá quá đắc. Lúc này có thể dùng tập mờ, trong đó các hàm thành viên được cho điểm trong khoảng [0,1].

Môt tập mờ A là tập có các thành viên được cho điểm trong khoảng thực: $\mu_A(x) \in [0, 1]$.

Tức là các phần tử có thể thuộc vào tập mờ với một mức độ nào đó. Như thế, tập mờ có thể dùng làm biểu diễn toán học cho các ý niệm chưa rõ, thí dụ *nhiệt độ thấp, người hơi cao, xe hơi đắc tiền*, v.v,…

Định nghĩa 2.1 (Tập mờ -Fuzzy Set) *Một tập mờ A trong vũ trụ (miền) X là tập được định nghĩa bởi hàm thành viên $\mu_A(x)$ là ánh xạ từ vũ trụ X vào một khoảng đơn vị:*

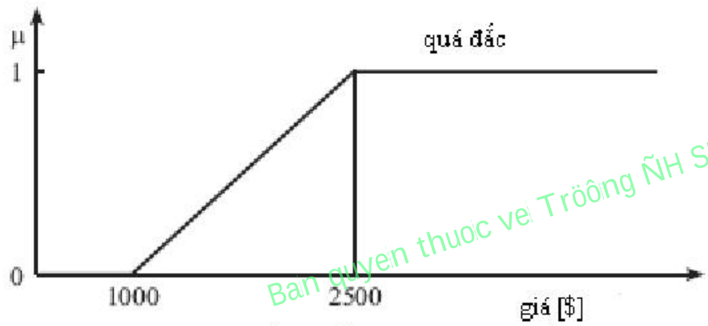
$$\mu_A(x): X \to [0, 1]. \tag{2.2}$$

F(*X*) *định nghĩa tất cả các tập mờ trong X.*

Nếu giá trị của hàm thành viên, được gọi là mức thành viên là bằng một, thì *x* phụ thuộc hoàn toàn vào tập mờ. Nếu giá trị này là không thì *x* không phụ thuộc vào tập. Nếu mức độ thành viên nằng giữa 0 và 1, thì *x* là thành phần của tập mờ:

$$\mu_A(x) \begin{cases} = 1 & x \text{ is a full member of } A \\ \in (0,1) & x \text{ is a partial member of } A \\ = 0 & x \text{ is not member of } A \end{cases} \qquad (2.3)$$

Trong các tài liệu về lý thuyết tập mờ, các tập bình thường (không mờ) thường được gọi là tập thực (*crisp)* hay tập cứng (*hard sets*). Có nhiều ký hiệu được dùng để chỉ hàm thành viên và mức tham gia như $\mu_A(x)$, $A(x)$ hay đôi khi chỉ là a.



Hình 2.1: Tập mờ biểu diễn giá PC quá đắc cho sinh viên

Thí dụ 2.1 (Tập mờ - Fuzzy Set) Hình 2.1 trình bày hàm thành viên có được từ tập mờ dùng biểu diễn giá PC *quá đắc* cho sinh viên.

Theo hàm thành viên này, nếu giá máy dươi $1000 thì rõ ràng là không quá đắc, và nếu giá máy là trên $2500 thì hoàn toàn là quá đắc. Ở giữa, có thể thấy được mức độ thành viên gia tăng của tập mờ quá đắc. Rõ ràng là không cần thành viên là phải tăng tuyến tính theo giá, hay là cần có việc chuyển giai đoạn không mịn từ $1000 sang $2500. Chú ý là trong các ứng dụng kỹ thuật, việc lựa chọn hàm thành viên cho tập mờ thường là tùy ý.

2. Đặc tính của tập mờ

Để thiết lập một khung sườn toán học cho tính toán dùng tập mờ, cần định nghĩa một số đặc tính của tập mờ. Phần này chỉ trình bày tổng quan về những gì cần cho tài liệu. Điều này gồm các định nghĩa về chiều cao (height), support, core, α-cut và cardinality của tập mờ. Ngoài ra, còn giới thiệu các đặc tính về normality và convexity. Cần tham khảo thêm (Klir and Yuan, 1995).

2.1 Tập mờ Normal và Subnormal

Ta biết là thành viên là yếu tố mức độ các phần tử của tập mờ. Chiều cao (*height)* của tập mờ là thành viên lớn nhất trong các phần tử của vũ trụ này. Tập mờ có chiều cao bằng một hay ít nhất có một phần tử x có trong miền X thì được gọi là tập mờ *normal*. Chiều cao của tập mờ *subnormal* thì bé hơn một với mọi phần tử trong miền. Khảo sát các định nghĩa sau:

Định nghĩa 2.2 (Chiều cao) *Chiều cao của tập mờ A là mức độ thành viên cao nhất của các phần tử trong A:*

$$hgt(A) = \sup_{x \in X} \mu_A(x) \quad (2.4)$$

Trong miền rời rạc *X*, phần lớn nhất (supremum) trở thành cực đại và do đó chiều cao là mức độ thành viên lớn nhất với mọi $x \in X$.

Định nghĩa 2.3 (Tập mờ Normal) *Tập mờ A là* normal *nếu* $\exists x \in X$ *sao cho* $\mu_A(x)=1$. *Tập mờ là không normal thì được gọi là* subnormal. *Toán tử* norm(*A*) cho thấy mức độ normal của tập mờ, *thí dụ* $A' = \text{norm}(A) \Leftrightarrow \mu'_A(x) = \mu A(x)/\text{hgt}(A)$, $\forall x$.

Support, core và α-cut là các tập *crisp* có được từ tập mờ thông qua cách chọn lựa các phần tử có mức thành viên thỏa một số điều kiện.

Định nghĩa 2.4 (Support) Support của tập mờ *A là tập con crisp của X, trong đó tất cả các phần tử đều có mức độ thành viên là không zero:*

$$\text{supp}(A) = \{x \mid \mu_A(x) > 0\}. \quad (2.5)$$

Định nghĩa 2.5 (Core) *Lõi (core) của tập mờ A là tập con của X bao gồm môi phần tử có mức độ thành vi6n đều bằng một:*

$$\text{core}(A) = \{x \mid \mu_A(x) = 1\}. \quad (2.6)$$

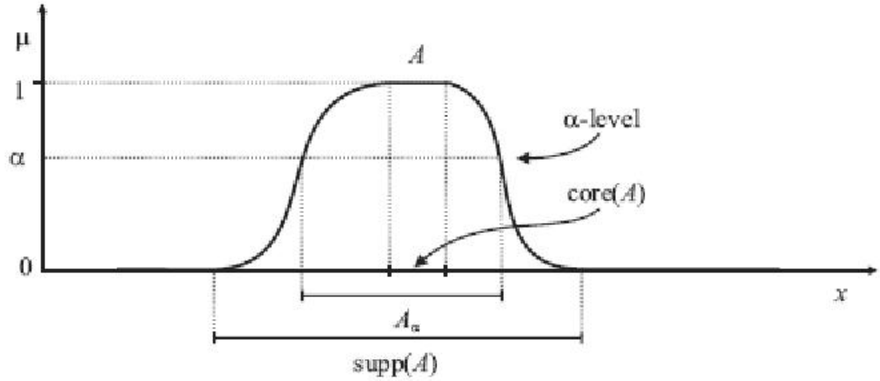
Trong một số tài liệu, đôi khi lõi (core) còn gọi là kernel, ker(*A*). Lõi của một tập mờ subnormal là trống.

Định nghĩa 2.6 (α-Cut) *Cắt* α*-cut* A_α *của tập mờ A là tập con crisp của vũ trụ X có tất cả các phần tử có mức độ thành viên lớn hơn hay bằng* α*:*

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\}, \qquad \alpha \in [0, 1]. \quad (2.7)$$

Toán tử α-cut *còn được gọi là* α-cut(*A*) *hay* α-cut(*A*, α). *Toán tử* α*-cut* A_α *là nghiêm ngặt nếu* $\mu_A(x) \neq \alpha$ *với mỗi* $x \in A_\alpha$. *Giá trị* α *được gọi là mức* α*-level.*

Hình 2.2 mô tả toán tử core, support và α-cut của tập mờ.



Hình 2.2 Lõi (core), support và α-cutcủa tập mờ.

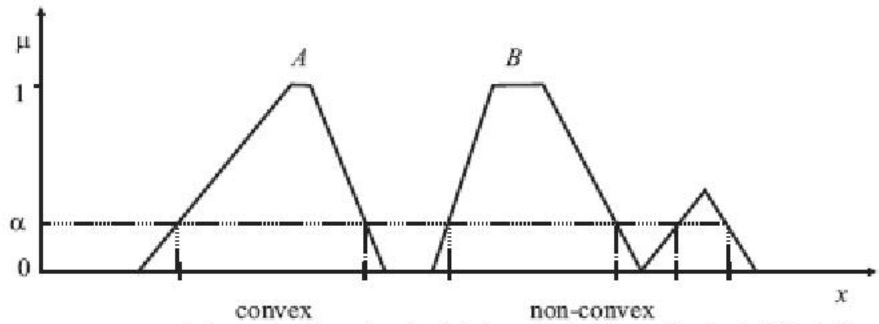
Lõi (core) và support của tập mờ còn có thể được định nghĩa từ α-cuts:

$$\text{core}(A) = 1\text{-cut}(A) \qquad (2.8)$$
$$\text{supp}(A) = 0\text{-cut}(A) \qquad (2.9)$$

Hàm thành viên có thể là unimodal (với một cực đại toàn cục) hay là multimodal (có nhiều maxima). Tập mờ unimodal được gọi là tập mờ lồi (convex fuzzy sets).
Tính lồi còn có thể được định nghĩa theo α-cuts:

Định nghĩa 2.7 (Tập mờ lồi) *Tậpmờ định nghĩa trong* R_n *là lồi (convex) nếu có từng tập* α*-cuts của mình là tập lồi.*

Hình 2.3 minh họa về tập mờ lồi và tập mờ không lồi.



Hình 2.3 Lõi (core)của các tập không lồi (tập crisp) và lồi (tập mờ)

Thí dụ 2.2 (Tập mờ không lồi) Hình 2.4 cho thí dụ về tập mờ không lồi biểu diễu "tuổi có rủi ro cao" trong chánh sách của công ty bảo hiểm xe. Các lái xe quá trẻ hay quá già đều có rủi ro cao hơn các lái xe trung niên.

Hình 2.4: Tập mờ định nghĩa "tuổi rủi ro cao" trong chính sách bảo hiểm xe là một thí dụ về tập mờ không lồi (non-convex)

Định nghĩa 2.8 (Cardinality) *Gọi* $A = \{\mu_A(x_i) \mid i = 1, 2, \ldots, n\}$ *là tập mờ rời rạc hữu hạn. Cardinality của tập mờ này được định nghĩa là tổng của các mức độ thành viên:*

$$|A| = \sum_{i=1}^{n} \mu_A(x_i) \quad (2.11)$$

Cardinality còn được định nghĩa là card(A).

3. Biểu diễn tập mờ

Có nhiều phương pháp định nghĩa tập (hay biểu diễn trên máy tính): thông qua mô tả giải tích các hàm thành viên $\mu_A(x) = f(x)$, thành danh mục miền thành phần cùng mức độ thành viên hay dùng toán tử α-cuts, như phân tích dưới đây.

3.1 Biểu diễn dùng nền tương đồng (Similarity-based)

Tập mờ thường được định nghĩa dùng tính tương đồng hay không tương đồng ((dis)similarity) của đối tượng x đang xét dùng prototype v của tập mờ

$$\mu(x) = \frac{1}{1 + d(x, v)} \quad (2.12)$$

Trường hợp này $d(x, v)$ định nghĩa đo lường về tính tương đồng trong không gian metric mà tiêu biểu là cự ly (thí dụ cự ly Euclide). Prototype là thành viên đầy đủ (phần tử tiêu biểu) của tập. Phần tử nào có cự ly đến prototype là không thì có mức độ thành viên gần một. Nếu cự ly tăng thì mức thành viên giảm. Thí dụ, xét hàm thành viên sau:

$\mu_A(x) = \dfrac{1}{1 + x^2}$, $\quad x \in$ R, *biểu diễn mức độ "gần zêrô" của số thực.*

3.2 Biểu diễn dùng tham số chức năng

Có nhiều dạng hàm thành viên tham số là:

Hàm thành viên dạng hình thang (trapezoidal):

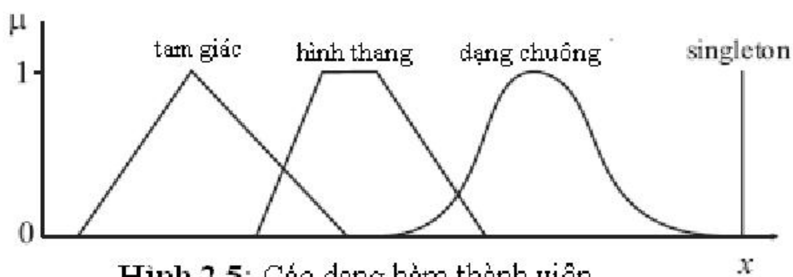
$$\mu(x,a,b,c,d) = \max\left(0, \min\left(\frac{x-a}{b-a}, \frac{d-x}{d-c}\right)\right), \tag{2.13}$$

Trong đó a, b, c và d là tọa độ các đỉnh của tam giác. Khi $b = c$, ta có hàm thành viên dạng tam giác.

Hàm thành viên dạng mũ từng đoạn:

$$\mu(x,c_l,c_r,w_l,w_r) = \begin{cases} \exp\left(-\left(\dfrac{x-c_l}{2w_l}\right)^2\right) & x < c_l \\ \exp\left(-\left(\dfrac{x-c_r}{2w_r}\right)^2\right) & x > c_r \\ 0 & otherwise \end{cases} \tag{2.14}$$

Trong đó c_l và c_r lần lượt là các vai trái và phải, và w_l, w_r lần lượt là bề rộng phải và trái. Khi $c_l = c_r$ và $w_l = w_r$ ta có hàm thành viên dạng Gauss.



Hình 2.5: Các dạng hàm thành viên

Hình 2.5 vẽ các dạng hàm thành viên tam giác, hình thang, dạng chuông (hàm mũ). Một tập mờ đặc biệt gọi là tập *singleton* (tập mờ biểu diễn bằng một số) đượcđịnh nghĩa là:

$$\mu_A(x) = \begin{cases} 1 & x = x_0 \\ 0 & otherwise \end{cases} \tag{2.15}$$

Một tập đặc biệt khác được gọi là tập vạn năng (*universal set)* với hàm thành viên bằng một trong mọi thành phần miền:

$$\mu A(x) = 1, \qquad \forall x. \tag{2.16}$$

Cuối cùng số mờ (*fuzzy number*) đôi khi được dùng chỉ tập mờ normal, convex được định nghĩa trên đường thẳng thực.

3.3 Biểu diễn theo điểm (Point-wise Representation)

Trong tập rời rạc $X = \{x_i \mid i = 1, 2, \ldots, n\}$, tập mờ A có thể được định nghĩa dùng bảng liệt kê các cặp có thứ tự: mức độ thành viên /phần tử của tập:

$$A = \{\mu_A(x_1)/x_1, \mu_A(x_2)/x_2, \ldots, \mu_A(x_n)/x_n\} = \{\mu_A(x)/x \mid x \in X\}, \qquad (2.17)$$

Thông thường, chỉ các phần tử $x \in X$ có mức độ thành viên khác không như đã liệt kê. Có thể gặp các trường hợp sau:

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \ldots + \mu_A(x_n)/x_n = \sum_{i=1}^{n} \mu_A(x_i)/x_i \qquad (2.18)$$

trong miền hữu hạn, và

$$A = \int_X \mu_A(x)/x \qquad (2.19)$$

trong miền liên tục. Chú ý, thay vì là tổng và tích phân, trong bài này, các ký hiệu $\sum$, + và $\int$ biểu diễn tập (union) các phần tử.

Cặp các vectơ (dãy trong các chương trình máy tính) có thể được dùng để lưu trữ các hàm thành viên rời rạc:

$$\mathrm{x} = [x_1, x_2, \ldots, x_n], \quad \mu = [\mu_A(x_1), \mu_A(x_2), \ldots, \mu_A(x_n)]. \qquad (2.20)$$

Có thể dùng phép nội suy để tìm các điểm trung gian. Biểu diễn này thường dùng trong các gói chương trình máy tính thương phẩm. Khi rời rạc hóa với các bước không đổi thì chỉ cần lưu trữ một mức độ thành viên μ.

3.4 Biểu diễn ở cấp tập hợp (Level Set Representation)

Tập mờ có thể được biểu diễn thành danh mục theo các mức α ($\alpha \in [0, 1]$) và các lát cắt (α-cuts) tương ứng:

$$A = \{\alpha_1/A_{\alpha 1}, \alpha_2/A_{\alpha 2}, \ldots, \alpha_n/A_{\alpha n}\} = \{\alpha/A_{\alpha n} \mid \alpha \in (0, 1)\}, \qquad (2.21)$$

Tầm của α cần được rời rạc hóa. Biểu diễn này có thể có ưu điểm là toán tử trong tập mờ con trong cùng vũ trụ, được định nghĩa như tập toán tử cổ điển trong các tập mức của chúng. Từ đó, thiết lập được đại số mờ (fuzzy arithmetic) dùng khoảng đại số (interval arithmetic), v.v,… Tuy nhiên, trong miền nhiều chiều, việc dùng biểu diễn theo mức tập hợp có thể làm gia tăng mức độ tính toán.

Thí dụ 2.3 (Đại số mờ: Fuzzy Arithmetic) Dùng phép biểu diễn trên mức tập hợp, có thể tìm kết quả của các toán tử đại số dùng số mờ (fuzzy numbers) dùng các phép toán tử đại số chuẩn trong cac phần cắt (α-cuts) của mình. Thí dụ xét phép cộng của hai số mờ A và B được định nghĩa trên đường thẳng thực:

$$A + B = \{\alpha/(A_{\alpha n} + B_{\alpha n}) \mid \alpha \in (0,\ 1)\}, \tag{2.22}$$

where $A_{\alpha n} + B_{\alpha n}$ là phép cộng của hai khoảng (intervals).

4. Các phép toán trên tập mờ

Định nghĩa các toán tử theo lý thuyết tập hợp (set-theoretic operations) như phép bù (complement), phép hội (union) và phép giao (intersection) có thể được mở rộng từ lý thuyết tập hợp truyền thống sang tập mờ. Do mức độ thành viên không còn bị giới hạn trong $\{0,\ 1\}$, nhưng có thể có giá trị nào đó trong khoảng $[0,\ 1]$, các toán tử này không thể được định nghĩa một cách độc nhất. Tuy nhiên, rõ ràng là các toán tử trong tập mờ phải cho kết quả đúng khi áp dụng vào tập truyền thống (trong đó tập truyền thống có thể xem là trường hợp đặc biệt của tập mờ).

Phần này giới thiệu các định nghĩa cơ bản của Zadeh về phép giao mờ (fuzzy intersection), phép hội (union) và phép bù (complement). Các toán tử giao và hội tổng quát, còn gọi là norms tam giác (t-norms) conorms tam giác (t-conorms) cũng được trình bày, ngoài ra toán tử ánh xạ (projection) và phép mở rộng trụ (cylindrical extension) có liên quan đến tập mờ nhiều chiều cũng được trình bày.

4.1 Phép bù (Complement), Hội (Union) và Giao (Intersection)

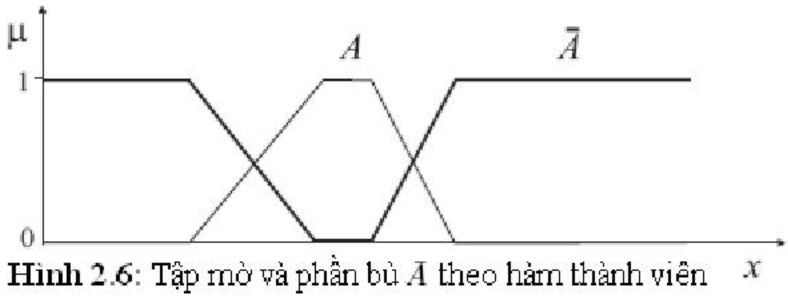
Định nghĩa 2.9 (phép bù của tập mờ) *Gọi A là tập mờ trong X. Phần phụ của A là tập mờ, gọi là tập mờ* $\bar{A}$*, sao cho với mỗi* $x \in X$*:*

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x). \tag{2.23}$$

Hình 2.6 trình bày thí dụ về phép bù mờ của hàm thành viên. Bên cạnh phép toán do Zadeh đề nghị, còn có thể dùng nhiều phép bù nữa. Thí dụ phép bù λ theo Sugeno (1977):

$$\mu_{\bar{A}}(x) = \frac{1 - \mu_A(x)}{1 + \lambda \mu_A(x)}. \tag{2.24}$$

Trong đó $\lambda > 0$ là tham số.

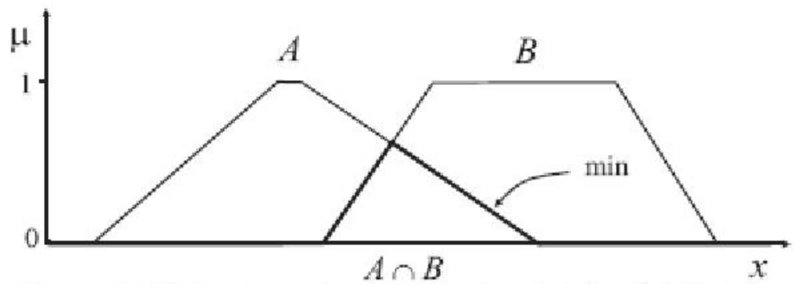


Hình 2.6: Tập mờ và phần bù $\bar{A}$ theo hàm thành viên

Định nghĩa 2.10 (phép giao của tập mờ) *Gọi A và B là hai tập mờ trong X. Phần giao(intersection) của A và B là tập mờ C, định nghĩa là $C = A \cap B$, sao cho với mỗi $x \in X$:*

$$\mu_C(x) = \min[\mu_A(x), \mu_B(x)]. \qquad (2.25)$$

Toán tử tối thiểu còn được gọi là '$\wedge$', thí dụ, $\mu_C(x) = \mu_A(x) \wedge \mu_B(x)$. Hình 2.7 cho thấy thí dụ về phần giao mờ của các hàm thành viên.

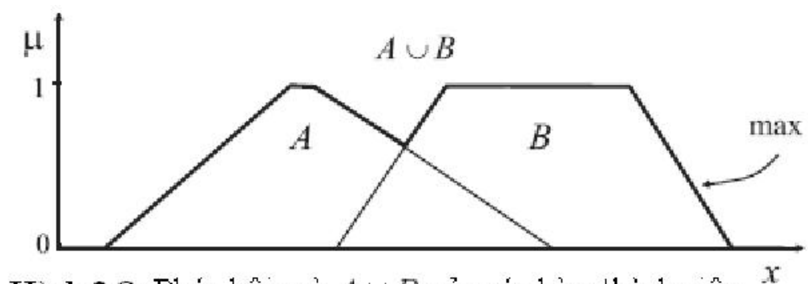


Hình 2.7: Phép giao mờ $A \cap B$ của các hàm thành viên

Định nghĩa 2.11: Hội của tập mờ (Union of Fuzzy Sets) *Gọi A và B là hai tập mờ trong X. Phép giao (union) của A và B là tập mờ C, định nghĩa là $C = A \cup B$, sao cho mỗi phần tử $x \in X$:*

$$\mu_C(x) = \max[\mu_A(x), \mu_B(x)]. \qquad (2.26)$$

Toán tử cực đại này còn được gọi là '$\vee$', thí dụ, $\mu_C(x) = \mu_A(x) \vee \mu_B(x)$. Hình 2.8 vẽ thí dụ về phép hội mờ của các hàm thành viên.



Hình 2.8: Phép hội mờ $A \cup B$ của các hàm thành viên

4.2 *T* -norms và *T* –conorms

Phép giao mờ của hai tập mờ có thể được xét một cách tổng quát dùng toán tử nhị phân trong khoảng đơn vị, thí dụ hàm có dạng:

$$T : [0, 1] \times [0, 1] \rightarrow [0, 1] \qquad (2.27)$$

Để có thể xem hàm T là hàm giao mờ, thì cần có một số đặc tính thích hợp. Hàm được gọi là t-norms (norms tam giác) có các đặc tính cần thiết cho phép giao. Tương tự, hàm gọi là t-conorms có thể dùng cho phép hội mờ.

Định nghĩa 2.12 (t-Norm/Phép giao mờ) *t-norm T là toán tử nhị phân trong khoảng đơn vị thỏa mãn ít nhất các tiên đề sau (axioms) với mọi a, b, c* $\in [0, 1]$ *(Klir and Yuan, 1995):*

$$
\begin{aligned}
&T(a, 1) = a \text{ (điều kiện biên)},\\
&b \le c \text{ dẫn đến } T(a, b) \le T(a, c) \text{ (tính đơn điệu)},\\
&T(a, b) = T(b, a) \text{ (tính giao hoán)},\\
&T(a, T(b, c)) = T(T(a, b), c) \text{ (tính phân bố)}.
\end{aligned}
\tag{2.28}
$$

Một số t-norms thường dùng là:

Phép giao chuẩn (Zadeh): $T(a, b) = \min(a, b)$

Tích đại số (phép giao xác suất): $T(a, b) = ab$

Phép giao Łukasiewicz (bold): $T(a, b) = \max(0, a + b - 1)$

Phép tối thiểu là phép t-norm lớn nhất (toán tử giao). Xem thí dụ trong hình 2.7 giới thiệu phần giao $A \cap B$ của các hàm thành viên có được từ các phép tính t-norm khác đều nằm dưới phần sậm màu của các hàm thành viên.

Định nghĩa 2.13 (t-Conorm/phép hội mờ) *t-conorm S là toán tử nhị phân trong khoảng đơn vị khi thỏa mãn ít nhất các tiên đề sau với mọi a, b, c* $\in [0, 1]$ *(Klir và Yuan, 1995):*

$$
\begin{aligned}
&S(a, 0) = a \text{ (điều kiện biên)},\\
&b \le c \text{ dẫn đến } S(a, b) \le S(a, c) \text{ (tính đơn điệu)},\\
&S(a, b) = S(b, a) \text{ (tính giao hoán)},\\
&S(a, S(b, c)) = S(S(a, b), c) \text{ (tính phân bố)}.
\end{aligned}
\tag{2.29}
$$

Một số t-conorms thường dùng là:

Phép hội chuẩn (Zadeh): $S(a, b) = \max(a, b)$,

Tổng đại số (phép hội xác suất): $S(a, b) = a + b - ab$,

Phép hội Łukasiewicz (bold): $S(a, b) = \min(1, a + b)$.

Phép tối đa là t-conorm bé nhất (toán tử hội). Trong thí dụ hình 2.8 tức là phép hội của $A \cup B$ có được từ các phép t-conorms khác đều nằm trên phần sậm màu của các hàm thành viên.

4.3 Ánh xạ và Mở rộng trụ (Projection and Cylindrical Extension)

Ánh xạ rút gọn tập mờ định nghĩa trong miền nhiều chiều (thí dụ R^2 của tập mờ sang miền có kích thước thấp hơn (như R). *Mở rộng trụ* là toán tử ngược lại, thí dụ phép mở rộng trụ định nghĩa từ miền có chiều thấp sang miền có nhiều chiều hơn, như sau:

Định nghĩa 2.14 (Ánh xạ của tập mờ) *Gọi* $U \subseteq U_1 \times U_2$ *là tập con trong không gian tích Cartesian, trong đó U_1 và U_2 tự thân đã là tích Cartesian trong các miền có chiều thấp hơn. Ánh xạ của tập mờ xác định U vào U_1 là phép chiếu* $\mathrm{proj}_{U1}: F(U) \to F(U_1)$ *định nghĩa bởi*

$$proj_{U1}(A) = \left\{ \sup_{U_2} \mu_A(u) / u_1 \in U_1 \right\}. \tag{2.30}$$

Cơ chế ánh xạ giảm chiều của không gian tích bằng cách lấy cực trị tối đa của hàm thành viên trong chiều cần phải giảm thiểu.

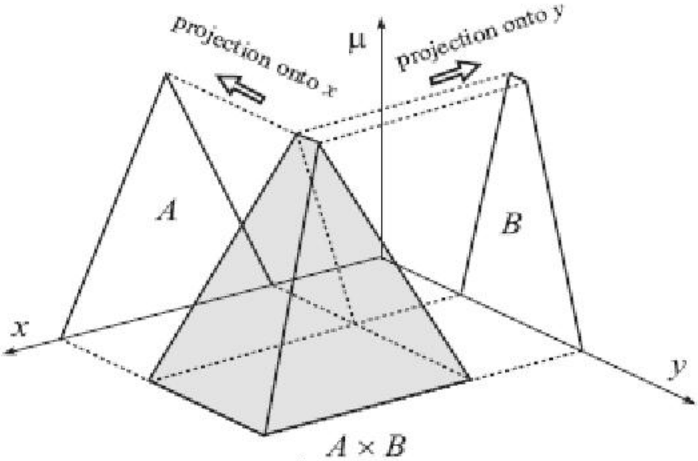
Thí dụ 2.4 (Ánh xạ) Giả sử tập mờ A định nghĩa trong $U \subset X \times Y \times Z$, với $X = \{x_1, x_2\}$, $Y = \{y_1, y_2\}$ và $Z = \{z_1, z_2\}$, như sau:

$$A = \{\mu_1/(x_1, y_1, z_1), \mu_2/(x_1, y_2, z_1), \mu_3/(x_2, y_1, z_1), \mu_4/(x_2, y_2, z_1), \mu_5/(x_2, y_2, z_2)\} \tag{2.31}$$

Tính ánh xạ của A vào X, Y và $X \times Y$:

$$\text{proj}_X(A) = \{\max(\mu_1, \mu_2)/x_1, \max(\mu_3, \mu_4, \mu_5)/x_2\}, \tag{2.33}$$
$$\text{proj}_Y(A) = \{\max(\mu_1, \mu_3)/y_1, \max(\mu_2, \mu_4, \mu_5)/y_2\}, \tag{2.34}$$
$$\text{proj}_{X\times Y}(A) = \{\mu_1/(x_1, y_1), \mu_2/(x_1, y_2), \mu_3/(x_2, y_1), \max(\mu_4, \mu_5)/(x_2, y_2)\}. \tag{2.35}$$

Có thể minh họa dễ dàng ánh xạ từ R^2 sang R như trong hình 2.9.



Hình 2.9: Thí dụ về ánh xạ từ $\mathbb{R}^2$ sang $\mathbb{R}$

Định nghĩa 2.15 (Mở rộng dạng trụ) *Xét $U \subseteq U_1 \times U_2$ là tập con của không gian tích Cartesian, trong đó U_1 và U_2 tự thân đã là tích Cartesian trong miền có chiều thấp hơn. Mở rộng trụ của tập mờ A định nghĩa U_1 vào U là phép áp* $\text{ext}_U: F(U_1) \rightarrow F(U)$ *định nghĩa bởi*

$$ext_U(A) = \{\mu_A(u_1 / u | u \in U\}. \tag{2.37}$$

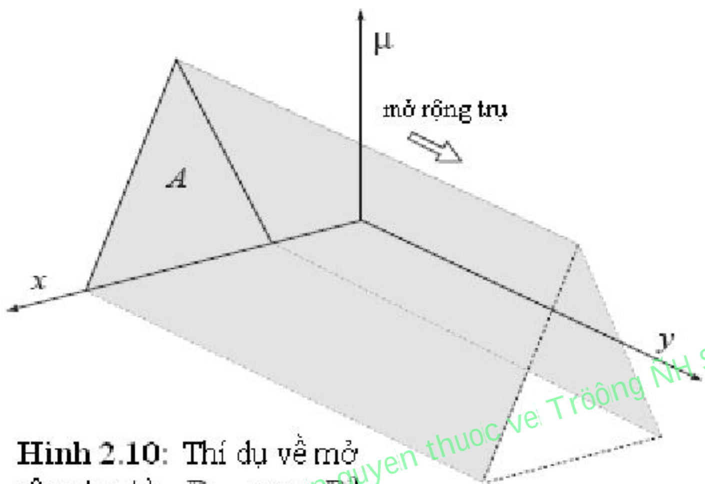
Mở rộng dạng trụ chỉ đơn giản là tạo bản sao mức độ thành viên từ miền hiện hữu sang các miền mới. Hình 2.10 mô tả phép mở rộng trụ từ R sang R^2.

Dễ dàng thấy được là phép ánh xạ dẫn đến mất thông tin, do A định nghĩa trong $X^n \subset X^m$ $(n < m)$ cho thấy là:

$$A = proj_{X^n}(ext_{X^m}(A)), \tag{2.38}$$

Nhưng
$$A \neq ext_{X^m}(proj_{X^n}(A)). \tag{2.39}$$

Chứng minh phần trong thí dụ 2.4 xem như là bài tập.



Hình 2.10: Thí dụ về mở rộng trụ từ $\mathbb{R}$ sang $\mathbb{R}^2$

4.4 Toán tử trong miền tích Cartesian

Các toán tử của lý thuyết tập hợp như phép hội và giao khi dùng trong tập mờ được định nghĩa trong các miền khác tạo tập mờ nhiều chiều trong tích Cartesian của các miền này. Toán tử được thực hiện đầu tiên là mở rộng tập mờ gốc vào trong miền tích Cartesian rồi tính toán tử trên các tập nhiều chiều này.

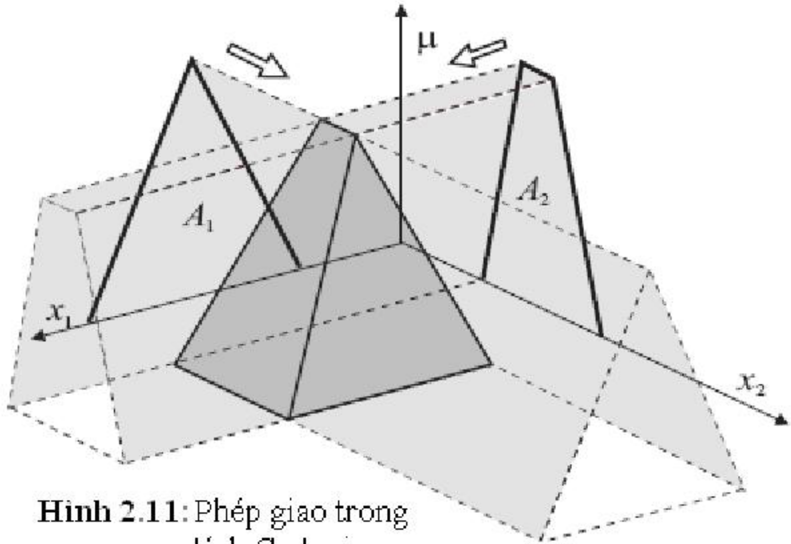
Thí dụ 2.5 (Phép giao trong tích Cartesian) Xét hai tập mờ A_1 và A_2 lần lượt định nghĩa trong các miền X_1 và X_2. Phép giao $A_1 \cap A_2$, còn được gọi là $A_1 \times A_2$ được cho bởi:

$$A_1 \times A_2 = \mathrm{ext}_{X2}(A_1) \cap \mathrm{ext}_{X1}(A_2). \tag{2.40}$$

Phép mở rộng trụ thường được xem là không tường minh và không định nghĩa:

$$\mu_{A1 \times A2}(x_1, x_2) = \mu_{A1}(x_1) \wedge \mu_{A2}(x_2). \tag{2.41}$$

Hình 2.11 minh họa phép toán này.



Hình 2.11: Phép giao trong tích Cartesian

4.5 Biên ngôn ngữ (Linguistic Hedges)

Các tập mờ có thể dùng biểu diễn thừa số ngôn ngữ định lượng (ý niệm: notions) tương tự như "ngắn", "dài", "đắc", v.v,.. thành hàm thành viên định nghĩa trong miền (cự ly, giá, v.v,..).

Khi dùng *linguistic hedges* (bộ bổ nghĩa: linguistic modifiers) thì ý nghĩa của các thừa số này có thể được thay đổi mà không cần định nghĩa lại các hàm thành viên. Thí dụ về các biên (hedges) là: *rất, hơi, nhiều hơn, ít hơn, thay vì, v.v,..* Thí dụ bổ nghĩa "rất" có thể dùng thay đổi từ "đắc" thành "rất đắc".

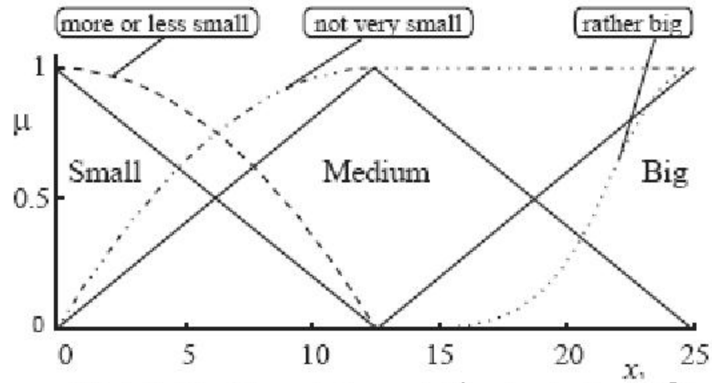
Có hai hướng chính dùng thực hiện (linguistic hedges) là *powered* hedges và *shifted* hedges. Powered hedges dùng hàm hoạt động trong mức độ thành viên của thừa số ngôn ngữ (Zimmermann, 1996). Thí dụ biên *rất* bình phương mức độ thành viên của thừa số có ý nghĩa cần thay đổi, thí dụ $\mu_{rất A}(x) = \mu^2{}_A(x)$. Shifted hedges (Lakoff, 1973), thì khác, dời hàm thành viên dọc theo miền hoạt động. Tổ hợp hai hướng này cũng đã được nghiên cứu (Novák, 1989; Novák, 1996).

Thí dụ2.6 Xét ba tập mờ *Small*, *Medium* và *Big* định nghĩa dùng hàm thành viên dạng tam giác. Hình2.12 vẽ các hàm thành viên này (đường sậm) dọc theo hàm thành viên đã bổ nghĩa "more or less small", "nor very small"và "rather big" có được khi áp dụng biên trong bảng 2.6.

linguistic hedge	operation	linguistic hedge	operation
very A	μ_A^2	more or less A	$\sqrt{\mu_A}$
not very A	$1-\mu_A^2$	rather A	$\text{int}\,(\mu_A)$

Trong bảng này, A là tập mờ và "int" là toán tử contrast intensification operator cho bởi:

$$\operatorname{int}(\mu_A) = \begin{cases} 2\mu_A^2 & \mu_A \le 0.5 \\ 1 - 2(1-\mu_A)^2 & otherwise \end{cases}$$



Hình 2.12: Tập mờ tham chiếu và các thay đổi dùng biến ngôn ngữ

5. Quan hệ mờ

Quan hệ mờ là tập mờ trong tích Cartesian $X_1 \times X_2 \times \cdots \times X_n$. Mức độ thành viên biểu diễn mức tương quan của các phần tử trong các miền X_i khác nhau.

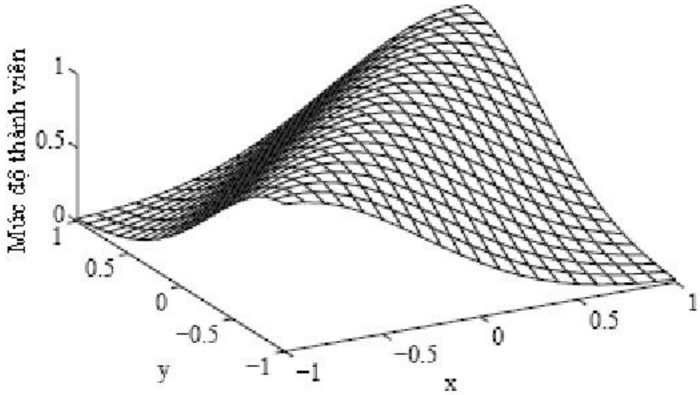
Định nghĩa 2.16 (Quan hệ mờ) *Quan hệ mờ bậc n là ánh xạ:*

$$R\text{:}\ \ X_1 \times X_2 \times \cdots \times X_n \rightarrow [0,\ 1], \qquad (2.42)$$

Qui định mức độ thành viên của mọi cặp $(x_1,\ x_2, \ldots,\ x_n)$ *của tích Cartesian* $X_1 \times X_2 \times \cdots \times X_n$.

Trên máy tính, R thường được biểu diễn dùng dãy n chiều: $R = [r_{i1, i2, \ldots, in}]$.

Thí dụ 2.7 (Quan hệ mờ) Xét quan hệ mờ R mô tả quan hệ $x \approx y$ ("*x là xấp xỉ bằng y*") dùng các hàm thành viên sau $\mu_R(x, y) = e^{-(x-y)^2}$.
Hình 2.13 minh họa quan hệ trong không gian ba chiều.



Hình 2.13: Quan hệ mờ $\mu_R(x, y) = e^{-(x-y)^2}$.

6. Tổ hợp quan hệ

Tổ hợp được định nghĩa (Zadeh, 1973) như sau: giả sử tồn tại quan hệ mờ R trong $X \times Y$ và A là tập mờ trong X. Thì tập con mờ B của Y có thể suy ra từ A thông qua tổ hợp A và R:

$$B = A \circ R. \tag{2.43}$$

Tổ hợp được định nghĩa là:

$$B = proj_Y\left(R \cap ext_{XxY}(A)\right). \tag{2.44}$$

Tổ hợp có thể xem như gồm hai pha: *tổ hợp* (phép giao) và *phép ánh xạ*. Zadeh đề nghị dùng tổ hợp *sup-min*. Giả sử A là tập mời có hàm thành viên $\mu_A(x)$ và R là quan hệ mờ có hàm thành viên là $\mu_R(x, y)$:

$$\mu_B(y) = \sup_x \min(\mu_A(x), \mu_R(x, y)). \tag{2.45}$$

Trong đó phép mở rộng trụ của A vào $X \times Y$ là không tường minh và *sup, min* lần lượt biểu diễn các pha ánh xạ và tổ hợp. Trường hợp tổng quát của tổ hợp, dùng *t*-norm T thay cho phép giao:

$$\mu_B(y) = \sup_x T\left(\mu_A(x), \mu_R(x, y)\right). \tag{2.46}$$

Thí dụ 2.8 (Quan hệ tổ hợp) Xét quan hệ mờ R biểu diễn quan hệ "*x là xấp xỉ bằng y*":

$$\mu_R(x, y) = \max(1 - 0.5 \cdot |x - y|, 0). \tag{2.47}$$

Hơn nữa, xét tập mờ A *"xấp xỉ 5"*:

$$\mu_A(x) = \max(1 - 0.5 \cdot |x - 5|, 0). \tag{2.48}$$

Giả sử R và A được rời rạc hóa với $x, y = 0, 1, 2, \ldots,$ vào $[0, 10]$. Như thế, tổ hợp là:

$$
\mu_B(y) = \overbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{2} \\ 1 \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}^{\mu_A(x)} \circ \overbrace{\begin{pmatrix}
1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 1
\end{pmatrix}}^{\mu_R(x,y)} =
$$

$$
= \max_x \overbrace{\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}}^{\min(\mu_A(x), \mu_R(x,y))} =
$$

$$
= \overbrace{\begin{pmatrix} 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \end{pmatrix}}^{\max_x \min(\mu_A(x), \mu_R(x,y))}
$$

Tập mờ có được này, định nghĩa trong Y có thể được diễn đạt thành *"xấp xỉ 5"*. Tuy nhiên, cần chú ý là điều này rộng hơn (ít chắc chắn hơn) so với tập được tìm ra. Điều này là do tính bất định của ngõ vào tập mờ đã được tổ hợp với yếu tố bất định trong quan hệ.

7. Tóm tắt và các vấn đề cần quan tâm

Tập mờ là tập không có biên rõ ràng: thành viên của tập mờ là số thực trong khoảng [0, 1]. Đã trình bày nhiều đặc tính khác nhau của tập mờ và các phép tính trên tập mờ. Quan hệ là tập mờ nhiều chiều có mức độ thành viên biểu diễn mức tương quan của các phần tử trong các miền khác nhau. Tổ hợp các quan hệ, dùng phép ánh xạ và phép mở rộng trụ là ý niệm quan trọng của logic mờ và suy luận xấp xỉ (approximate reasoning), sẽ được trình bày trong các chương tiếp.

8. Bài tập

1. Cho biết sự khác biệt giữa hàm thành viên của tập thường và của tập mờ?

2. Xét tập mờ C định nghĩa dùng hàm thành viên $\mu_C(x)$:R $\rightarrow [0, 1]$:
$\mu_C(x) = 1/(1 + |x|)$. Tính phép α-cut của C khi $\alpha = 0.5$.

3. Xét tập mờ A và B sao cho lõi core(A) $\cap$ core(B) $= \varnothing$. Tập mờ $C = A \cap B$ có là normal không? Cho biết điều kiện về supports của A và B sao cho card(C)>0 luôn luôn đúng?

4. Xét tập mờ A được định nghĩa trong $X \times Y$ với $X = \{x_1, x_2\}$, $Y = \{y_1, y_2\}$:

$$A = \{0.1/(x_1, y_1), 0.2/(x_1, y_2), 0.7/(x_2, y_1), 0.9/(x_2, y_2)\}$$

Tính ánh xạ của A vào X và Y.

5. Tìm mở rộng trụ của tập mờ $A = \{0.3/x_1, 0.4/x_2\}$ vào miền tích Cartesian $\{x_1, x_2\} \times \{y_1, y_2\}$.

6. Cho tập mờ $A = \{0.1/x_1, 0.6/x_2\}$ và $B = \{1/y_1, 0.7/y_2\}$, tìm phần hội $A \cup B$ và phần giao $A \cap B$. Dùng các toán tử của Zadeh (max, min).

7. Cho quan hệ mờ R: $X \times Y \rightarrow [0, 1]$:

$R =$

	y_1	y_2	y_3
x_1	0.7	0.3	0.1
x_2	0.4	0.8	0.2
x_3	0.1	0.2	0.9

Và tập mờ $A = \{0.1/x_1, 1/x_2, 0.4/x_3\}$. Tính tập mờ $B = A \circ R$, trong đó '$\circ$' là toán tử tổ hợp max-min.

8. Chứng minh định lý De Morgan $\overline{A \cup B} = \overline{A} \cap \overline{B}$ cũng đúng trong các tập mờ A và B, dùng các toán tử hội, giao, bù của Zadeh.

CHƯƠNG 3: HỆ MỜ

Hệ tĩnh và hệ động dùng tập mờ và khung sườn toán học tương ứng được gọi là *hệ mờ* (*fuzzy system*). Các tập mờ này có thể bao hàm trong hệ thống theo một số cách, thí dụ:

- *Trong mô tả hệ thống.* Thí dụ một hệ thống có thể được định nghĩa là một tập các luật nếu-thì dùng các thuộc tính mờ (fuzzy predicates), hay là quan hệ mờ. Thí dụ luật mờ mô tả quan hệ giữa công suất nhiệt và xu hướng nhiệt độ trong phòng như sau:

  Nếu công suất nhiệt là cao thì nhiệt độ sẽ tăng nhanh.

- *Trong đặc trưng các tham số của hệ thống.* Hệ thống có thể được định nghĩa bằng phương trình đại số hay phương trình vi phân, với các tham số là các số mờ (fuzzy numbers) thay vì là số thực (real numbers). Thí dụ, xét phương trình $y = \tilde{3}x_1 + \tilde{5}x_2$, trong đó $\tilde{3}$ và $\tilde{5}$ là các số mờ lần lượt là "vào khoảng ba" và "vào khoảng năm", do các hàm thành viên định nghĩa. Số mờ diễn tả tính không chắc chắn (uncertainty) trong giá trị tham số.
- *Ngõ vào, ngõ ra và các biến trạng thái của hệ thống có thể là tập mờ.* Các ngõ vào mờ có thể được đọc từ các cảm biến chưa đáng tin cậy (unreliable sensors) hay các dữ liệu có nhiễu ("noisy" data), hay các đại lượng có liên quan đến cảm nhận của con người, như tiện nghi, sắc đẹp, v.v,…Hệ mờ có thể xử lý các thông tin này, mà các hệ thống truyền thống (hệ crisp) không xử lý được

.

Một hệ mờ có thể có đồng thời nhiều thuộc tính trên. Hệ mờ có thể được xem như là tổng quát hóa của hệ thống có giá trị từng đoạn (interval-valued systems), chính là tổng quát của hệ crisp. Quan hệ này được mô tả trong hình 3.1 về thí dụ của hàm crisp và các khoảng giá trị cùng với phép tổng quát hóa mờ (fuzzy generalizations). Đồng thời cũng mô tả một cách hệ thống các ước lượng về hàm crisp, khoảng và dữ liệu mờ.

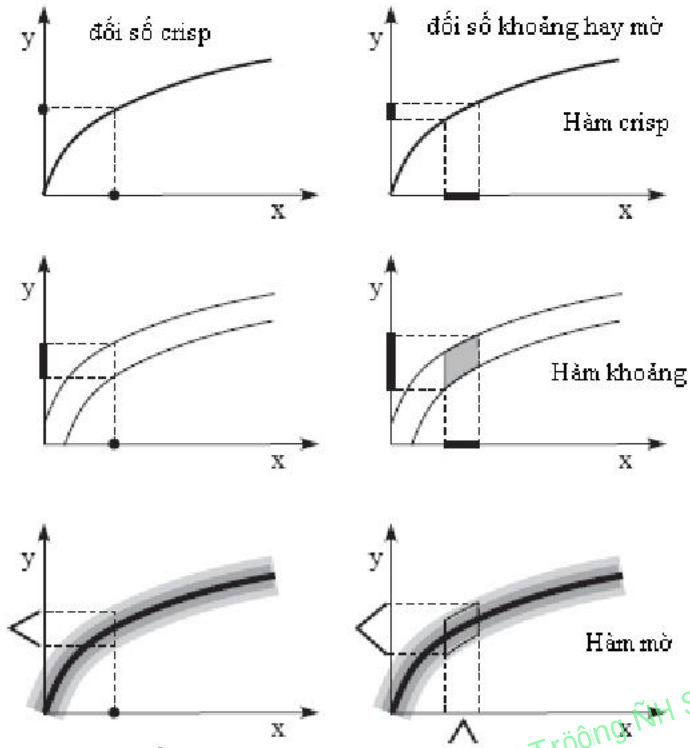
Một hàm $f: X \rightarrow Y$ có thể xem là tập con của tích Cartesian $X \times Y$, thí dụ theo *quan hệ (relation)*. Việc ước lượng hàm cho từng giá trị vào được thực hiện theo ba bước (hình 3.1):

1. Mở rộng ngõ vào cho trước vào không gian tích $X \times Y$ (đường dọc đứt nét).
2. Tìm phần giao của mở rộng này cới quan hệ (phần giao của đường đứt nét dọc với hàm).
3. Chiếu phần giao này vào Y (đường đứt nét ngang)

.

Thủ tục này dùng được cho tập crisp, khoảng và hàm mờ, dữ liệu mờ. Chú ý là hình vẽ trên giúp bạn hiểu được vai trò của quan hệ mờ trong suy diễn mờ (fuzzy inference).

Thông thường nhất thì hệ mờ được định nghĩa dùng luật nếu-thì: hệ mờ dùng luật nền (*rule-based fuzzy systems*). Trong phần tiếp sau đây chỉ chú ý đến các hệ thống dạng này. Hệ mờ có thể được dùng trong nhiều mục đích, như mô hình hóa, phân tích dữ liệu, dự báo và điều khiển. Để đơn giản, các hệ mờ dùng luật nền sẽ được gọi là hệ mờ, trừ khi có các ghi chú khác.



Hình.1: Ước lượng các hàm crisp, khoảng và mờ từ các đối số crisp, khoảng và mờ

1. Hệ mờ dùng luật nền

Trong hệ mờ dùng luật nền, quan hệ giữa các biến được biểu diễn dùng các luật nếu-thì theo dạng tổng quát sau:

Nếu tiền đề thì hệ quả.

Mệnh đề mờ được định nghĩa theo "*x* là lớn", trong đó "lớn" gọi là *nhãn ngôn ngữ* (*linguistic label)*, được định nghĩa dùng tập mờ trong vũ trụ của biến *x*. Các nhãn ngôn ngữ được xem là các hằng số mờ (fuzzy constants), thừa số mờ (fuzzy terms) hay các ý niệm mờ (fuzzy notions). Bổ nghĩa (linguistic modifiers: hedges) có thể dùng để thay đổi ý nghĩa của nhãn ngôn ngữ. Thí dụ, bổ nghĩa *rất* có thể dùng để thay đổi từ "*x* là lớn" sang "*x* là rất lớn ". Tiền đề thường là mệnh đề mờ có dạng "x là *A*" trong đó x là biến ngôn ngữ và *A* là hằng số ngôn ngữ (thừa số). Tùy theo cấu trúc đặc thù của mệnh đề hệ quả, có ba dạng mô hình chính sau đây:

- *Mô hình ngôn ngữ mờ (Linguistic fuzzy model)* (Zadeh, 1973; Mamdani, 1977), trong đó cả phần tiền đề và hệ quả đều là mệnh đề mờ. Mô hình mờ *Singleton* là dạng đặc biệt trong đó hệ quả nằm trong tập singleton (các hằng số thực).
- *Mô hình quan hệ mờ (Fuzzy relational model:* Pedrycz, 1984; Yi và Chung, 1993), có thể xem là trường hợp tổng quát của mô hình ngôn ngữ, cho phép một mệnh đề tiền đề đặc thù quan hệ với nhiều mệnh đề hệ quả khác nhau dùng quan hệ mờ (fuzzy relation).
- *Mô hình mờ Takagi–Sugeno (TS fuzzy mode)l* (Takagi and Sugeno, 1985), trong đó hệ quả là các hàm crisp của biến tiền đề thay vì là mệnh đề mờ.

Phần sau trình bày chi tiết các dạng mô hình mờ.

2. Mô hình dạng ngôn ngữ

Mô hình mờ dạng ngôn ngữ (Zadeh, 1973; Mamdani, 1977) được trình bày nhằm nắm được kiến thức định tính theo dạng luật nếu-thì:

$$\mathfrak{R}_i: \text{Nếu x là } A_i \text{ thì y là } B_i, \quad i = 1, 2, \ldots, K. \tag{3.1}$$

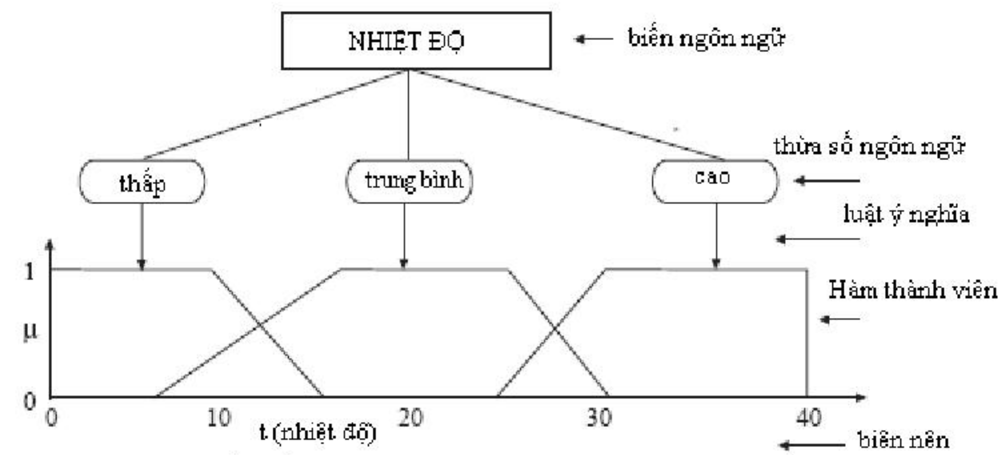
Biến vào x (tiền đề) gọi là biến ngôn ngữ (*linguistic variable)*, và hệ quả A_i là thừa số ngôn ngữ (nhãn) (*linguistic terms*-labels). Tương tự, hệ quả ngõ ra y là biến ngôn ngữ và B_i là thừa số hệ quả dạng ngôn ngữ. Các giá trị x(y) thường là tập mờ, ngoài ra do số thực là một trường hợp đặc biệt của tập mờ (tập singleton), nên các biến này có thể có giá trị thực (vector). Thừa số ngôn ngữ $A_i(B_i)$ luôn luôn là tập mờ.Thừa số ngôn ngữ có thể xem là các giá trị định tính (information granulae) được dùng để mô tả quan hệ đặc thù của các luật ngôn ngữ. Thường thì tập N các thừa số ngôn ngữ $A = \{A_1, A_2, \ldots, A_N\}$ được định nghĩa trong miền của biến x. Do biến này giả định các giá trị ngôn ngữ, nên được gọi là biến ngôn ngữ. Nhằm phân biệt giữa biến ngôn ngữ và biến gốc dạng số, nên biến sau được gọi là biến nền (*base variable)*.

Đinh nghĩa 3.1 (Biến ngôn ngữ) *Biến ngôn ngữ L được định nghĩa là tập gồm năm giá trị (quintuple: Klir and Yuan, 1995):*

$$L = (x, A, X, g, m), \tag{3.2}$$

Trong đó x *là biến nền (còn được gọi là biến ngôn ngữ),*
$A = \{A_1, A_2, \ldots, A_N\}$ *là tập* các thừa số ngôn ngữ*,* X *là* miền *(vũ trụ hoạt động) của* x*,* g là luật cú pháp (syntactic rule) *nhằm tạo ra các thừa số ngôn ngữ và* m *là luật ý nghĩa (*semantic rule) nhằm định nghĩa ý nghĩa của từng thừa số ngôn ngữ *(tập mờ trong* X*).*

Thí dụ 3.1 (Biến ngôn ngữ) Hình 3.2 trình bày thí dụ về biến ngôn ngữ "nhiệt độ" với ba thừa số ngôn ngữ "thấp", "trung bình" và "cao". Biến nền là nhiệt độ có giá trị là đơn vị vật lý phù hợp.



Hình 3.2: Thí dụ về biến ngôn ngữ "nhiệt độ"dùng ba thừa số ngôn ngữ

Các thừa số ngôn ngữ cần thỏa mãn các đặc tính về (*bao phủ*) *coverage* và *semantic soundness* (Pedrycz, 1995).

Bao phủ (Coverage). Coverage có nghĩa là từng miền của các phần tử phải được định nghĩa với ít nhất là một tập mờ có mức độ thành viên khác không, thí dụ:

$$\forall x \in X, \exists i, \mu_{Ai}(x) > 0; \qquad (3.3)$$

Mặt khác, một điều kiện mạnh hơn được gọi là ε-coverage phát biểu như sau::

$$\forall x \in X, \exists i, \mu_{Ai}(x) > \varepsilon, \quad \varepsilon \in (0,1). \qquad (3.4)$$

Thí dụ, các hàm thành viên trong hình 3.2 thỏa mãn ε-coverage với $\varepsilon = 0.5$. Thuật toán xâu chuỗi dùng tạo tự động mô hình mờ từ dữ liệu được trình bày trong chương 4 còn có yêu cầu về điều kiện mạnh hơn:

$$\sum_{i=1}^{N} \mu_{Ai}(x) = 1, \qquad \forall x \in X. \qquad (3.5)$$

cho thấy với từng x, thì tổng của mức độ thành viên phải bằng một. Tập các hàm thành viên này được gọi là partition mờ (*fuzzy partition*), được trình bày kỹ trong chương 4.

Semantic Soundness. Ý nghĩa đầy đủ (Semantic soundness) liên quan ý nghĩa ngôn ngữ của các tập mờ. Thông thường, A_i là tập lồi (convex) và tập mờ chuẩn (normal fuzzy sets) , thường là đủ phân cách (disjoint), và số tập con N các biến là ít (cao nhất là chín). Số thừa số ngôn ngữ và hình dáng đặc thù cùng phần chồng lắp (overlap) của các hàm thành viên có ảnh hưởng đến tính *tạo hạt* (granularity) của quá trình xử lý thông tin trên tập mờ, thì cũng ảnh hưởng đến mức chính xác cho hệ thống cần biểu diễn dùng tập mờ. Thí dụ, các hàm thành viên dạng tam giác như vẽ ở hình 3.2, cung cấp một số dạng về vấn đề ẩn thông tin "information hiding" của dữ liệu bên trong lõi (cores) của hàm thành viên (thí dụ, không thể phân biệt nhiệt độ trong khoãng từ 0 và 5 độ, do đều được xếp vào lớp *thấp* với độ 1). Ánh xạ tốt về hình dáng có thể biểu diễn chính xác dùng độ *tạo hạt* (granularity) rất thấp.

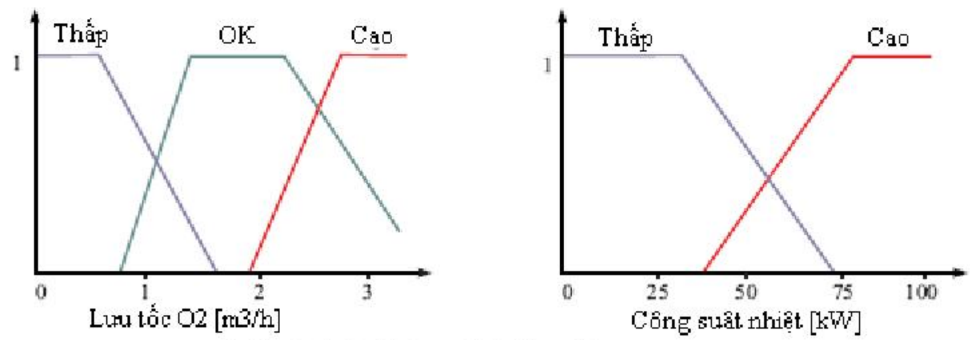
Hàm thành viên có thể được định nghĩa nhờ bộ phát triển mô hình (model developer: expert), dùng kiến thức đã có, như trong điều khiển mờ dùng nền tri thức (Driankov, et al., 1993). Trường hợp này thì các hàm thành viên được thiết kế để biểu diễn ý nghĩa của thừa số ngôn ngữ trong ngữ cảnh đã cho. Khi đã có được dữ liệu vào-ra của hệ thống đang khảo sát, thì áp dụng được các phương pháp cấu tạo hay thích ứng các hàm thành viên, xem chương 5.

Thí dụ 3.2 (Mô hình ngôn ngữ) Xét mô hình mờ đơn giản mô tả định tính công suất nhiệt của bộ đốt gas phụ thuộc vào lượng oxy cung cấp (giả sử lượng gas cung cấp là không đổi). Ngõ vào dạng vô hướng là lưu tốc của oxy (x), và ngõ ra vô hướng là công suất nhiệt (y). Định nghĩa tập thừa số tiền đề ngôn ngữ: A = {*Thấp, ,OK, Cao*}, và tập thừa số ngôn ngữ hệ quả: B = {*Thấp, Cao*}.

Quan hệ định tính giữa mô hình vào và ra có thể được biểu diễn dùng các luật sau:

$\mathfrak{R}_1$: Nếu lưu tốc O2 là *Thấp* thì công suất nhiệt là *Thấp*.

$\Re_2$: Nếu lưu tốc O2 là *OK* thì công suất nhiệt là *Cao*.
$\Re_3$: Nếu lưu tốc O2 là *Cao* thì công suất nhiệt là *Thấp*.



Hình 3.3: Hàm thành viên

Ý nghĩa của các thừa số ngôn ngữ được định nghĩa từ hàm thành viên, vẽ ở hình 3.3. Các giá trị số của các biến nền được chọn lựa một cách bất kỳ. Chú ý là không định nghĩa được ý nghĩa tổng quát của các biến ngôn ngữ. Trong thí dụ này, thì phụ thuộc vào dạng của lưu tốc, của hơi đốt, loại bộ đốt, v.v,... Tuy nhiên, quan hệ định tính do các luật diễn tả vẫn có giá trị.

2.1 Suy diễn từ mô hình ngôn ngữ

Suy diễn từ biến ngôn ngữ trong hệ dùng luật nền mờ là quá trình tìm tập mờ ngõ ra theo các luật và tập các tín hiệu vào. Cơ chế suy diễn trong mô hình ngôn ngữ dùng cơ sở luật suy diễn tổ hợp (*compositional rule of inference*: Zadeh, 1973).

Mỗi luật trong (3.1) có thể được xem là quan hệ mờ (các giới hạn mờ trên sự xuất hiện đồng thời các giá trị x và y): $R: (X \times Y) \rightarrow [0,\ 1]$ được tính từ:

$$\mu_R(\mathrm{x},\ \mathrm{y}) = \mathrm{I}(\mu_\mathrm{A}(\mathrm{x}),\ \mu_\mathrm{B}(\mathrm{y}))\,. \qquad (3.6)$$

Chỉ số i được bỏ qua cho ý niệm dễ dàng. Toán tử I có thể là hàm ý mờ (fuzzy implication) hay là toán tử kết thợp (conjunction) (dạng *t*-norm). Chú ý là $I(\cdot,\ \cdot)$ được tính trong không gian tích Cartesian $X \times Y$, với mọi cặp có thể có của x và y.

Hàm ý mờ (Fuzzy implications) được dùng khi luật (3.1) được xem là hàm ý: $A_i \rightarrow B_i$, thí dụ "A_i hàm ý B_i". Trong phép logic cổ điển thì điều này có nghĩa là nếu A đúng, thì B phải đúng cũng như phép hàm ý là đúng. Không thể nói gì về B khi A không đúng, và quan hệ cũng không thể đảo ngược được. Khi dùng phép kết nối, $A \wedge B$, thì diễn dịch thành luật nếu-thì là "sẽ là đúng nếu A và B cùng đúng". Quan hệ này là đối xứng (không có chiều) và có thể đảo được.

Thí dụ về hàm ý mờ là hàm ý Łukasiewicz cho bởi:

$$\mathrm{I}(\mu_\mathrm{A}(\mathrm{x}),\ \mu_\mathrm{B}(\mathrm{y})) = \min(1,\ 1 - \mu_\mathrm{A}(\mathrm{x}) + \mu_\mathrm{B}(\mathrm{y})), \qquad (3.7)$$

Hay hàm ý Kleene–Diene:

$$\mathrm{I}(\mu_\mathrm{A}(\mathrm{x}),\ \mu_\mathrm{B}(\mathrm{y})) = \max(1 - \mu_\mathrm{A}(\mathrm{x}),\ \mu_\mathrm{B}(\mathrm{y})). \qquad (3.8)$$

Thí dụ về t-norms là tối thiểu, tuy không phải lúc nào cũng đúng, được gọi là *hàm ý* Mamdani,

$$I(\mu_A(x), \mu_B(y)) = \min(\mu_A(x), \mu_B(y)), \qquad (3.9)$$

Hay trường hợp tích, còn được gọi là hàm ý Larsen,

$$I(\mu_A(x), \mu_B(y)) = \mu_A(x) \cdot \mu_B(y). \qquad (3.10)$$

Chi tiết về hàm ý mờ có thể tham khảo từ (Klir and Yuan, 1995; Lee, 1990a; Lee, 1990b; Jager, 1995).

Cơ chế suy diễn được dựa trên luật *modus ponens* tổng quát:

Nếu x là *A* thì y là *B*

x là *A'*

y là *B'*

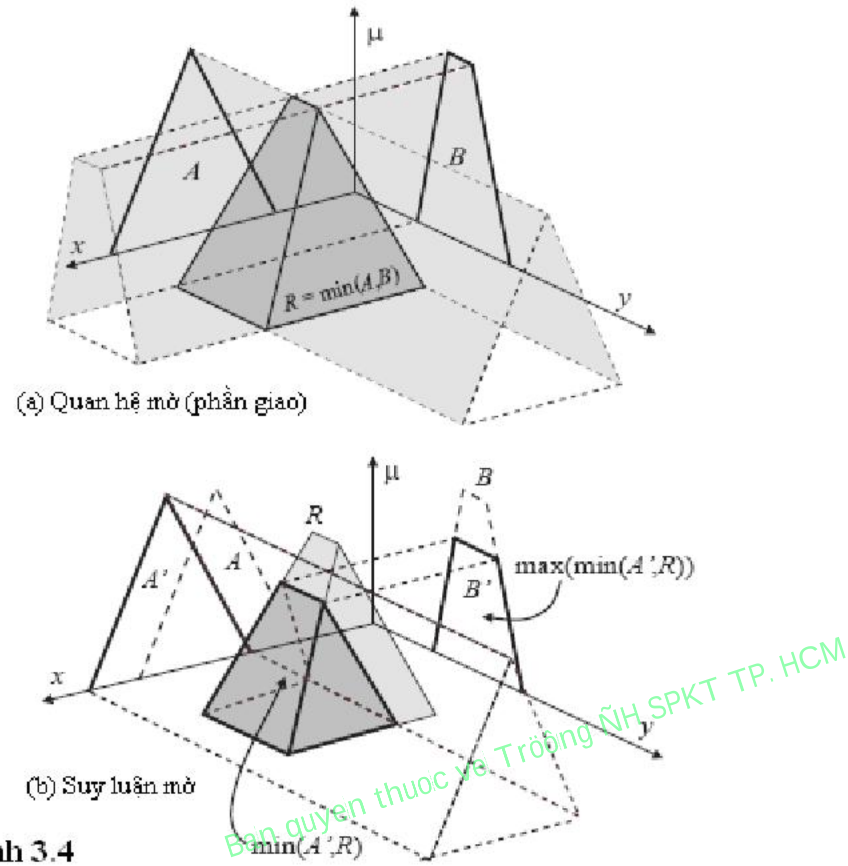
Luật nếu-thì vừa cho và thực tế là "x là *A'* ", tập mờ ta *B'* tìm được từ tổ hợp quan hệ max-t (Klir và Yuan, 1995):

$$B' = A' \circ R\,. \qquad (3.11)$$

Trường hợp t-norm tối thiểu, có được tổ hợp max-min:

$$\mu_{B'}(y) = \max_{X} \min_{X,Y}\left(\mu_{A'}(x), \mu_R(x,y)\right) \qquad (3.12)$$

Hình 3.4a minh họa thí dụ về quan hệ mờ R được tính từ (3.9). Hình 3.4b cho thấy kết luận của B', cho quan hệ R và ngõ ra A', dùng tổ hợp max-min (3.12). Có thể thấy B' là subnormal, biểu diễn yếu tố bất định (uncertainty) của ngõ vào ($A' \neq A$). Quan hệ tính toán phải được thiết lập trong miền rời rạc, hảy xem thí dụ.



Hình 3.4

(a) Quan hệ mờ biểu diễn luật "Nếu **x** là **A** thì **y** là **B**"

(b) Luật suy diễn tổ hợp

Thí dụ 3.3 (Luật suy diễn tổ hợp) Xét luật mờ

Nếu x là A thì y là B

Cùng tập mờ:

$$A = \{0/1,\ 0.1/2,\ 0.4/3,\ 0.8/4,\ 1/5\},$$
$$B = \{0/-2,\ 0.6/-1,\ 1/0,\ 0.6/1,\ 0/2\}.$$

Dùng phép t-norm tối thiểu (hàm ý Mamdani), quan hệ R_M biểu diễn luật mờ được tính dùng (3.9):

$$R_M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0.1 & 0.1 & 0 \\ 0 & 0.4 & 0.4 & 0.4 & 0 \\ 0 & 0.6 & 0.8 & 0.6 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \end{bmatrix}. \qquad (3.14)$$

Các hàng trong ma trận quan hệ tương ứng với miền các phần tử của A và cột là miền các phần tử của B. Xét tập mờ ngõ vào của luật:

$$A' = \{0/1,\ 0.2/2,\ 0.8/3,\ 1/4,\ 0.1/5\}. \qquad (3.15)$$

Ứng dụng tổ hợp max-min (3.12), $B'_M = A' \circ R_M$, có tập mờ ngõ ra:

$$B'_M = \{0/-2,\ 0.6/-1,\ 0.8/0,\ 0.6/1,\ 0/2\}. \qquad (3.16)$$

Dùng hàm ý mờ Łukasiewicz (3.7), có các quan hệ sau:

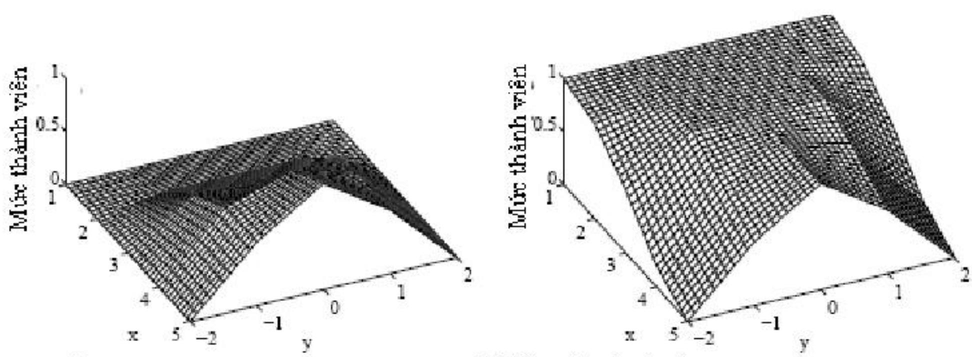
$$R_L = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0.9 & 1 & 1 & 1 & 0.9 \\ 0.6 & 1 & 1 & 1 & 0.6 \\ 0.2 & 0.8 & 1 & 0.8 & 0.2 \\ 0 & 0.6 & 1 & 0.6 & 0 \end{bmatrix}$$

Dùng tổ hợp max-t, trong đó *t*-norm là phần giao Łukasiewicz (bold) (xem định nghĩa 2.12), tập suy luận mờ $B'_L = A' \circ R_L$ bằng:

$$B'_L = \{0.4/-2,\ 0.8/-1,\ 1/0,\ 0.8/1,\ 0.4/2\}. \qquad (3.18)$$

Chú ý là sai biệt giữa các quan hệ R_M và R_L, được vẽ ở hình 3.5. Hàm ý chỉ sai (nhập zero trong quan hệ) khi *A* đúng và *B* thì không. Khi *A* không đúng, giá trị thực của hàm ý là 1 bất chấp *B*. Tuy nhiên, *t*-norm là sai khi *A* hay *B* hay cả hai đều sai, và như thế biểu diễn một quan hệ hai chiều (tương hỗ).

Sai biệt này ảnh hưởng một cách tự nhiên lên kết quả của quá trình suy diễn. Do tập mờ vào *A'* khác biệt với tập tiền đề *A*, kết luận có được *B'* trong tất cả các trường hợp đều "không chắc chắn" so với *B*. Sai biệt cùng với hàm ý mờ được phản ánh trong tập giá trị thành viên gia tăng của miền các phần tử có mức thành viên thấp hay zêrô trong *B*, điều này có nghĩa là các giá trị ngõ ra có khả năng có mức độ cao hơn. Tuy nhiên, phép *t*-norm làm giảm mức độ thành viên của các phần tử có mức thành viên cao trong B, làm cho kết quả này càng ít có khả năng. Điều này ảnh hưởng lên đặc tính của hai cơ chế suy diễn và việc chọn lựa phương pháp giải mờ thích hợp, sẽ được thảo luận sau.



(a) toán tử t-norm (minimum) (b) Hàm ý Lukasiewicz

Hình 3.5 Quan hệ mờ có được dùng toán tử t-norm (minimum) và phép hàm ý mờ

Toàn bộ luật nền (3.1) được biểu diễn bằng cách gộp các quan hệ R_i của từng luật vào một quan hệ mờ. Nếu R_i biểu diễn các hàm ý, thì R tìm được từ toán tử giao:

$$R = \bigcap_{i=1}^{K} R_i \quad \text{tức là} \quad \mu_R(x, y) = \min_{1 \le i \le K} \mu_{Ri}(x, y) . \tag{3.19}$$

Nếu I là t-norm, thì quan hệ gộp R được tính từ phép hội của từng luật trong quan hệ mờ R_i:

$$R = \bigcup_{i=1}^{K} R_i \quad \text{tức là} \quad \mu_R(x, y) = \max_{1 \le i \le K} \mu_{Ri}(x, y) . \tag{3.20}$$

Tập mờ ra *B'* được suy luận cùng phương pháp với trường hợp một luật, dùng tổ hợp luật suy diễn (3.11).

Phần biểu diễn nói trên của hệ dùng quan hệ mờ được gọi là graph mờ (*fuzzy graph)*, và tổ hợp luật suy diễn có thể xem là phép ước lượng hàm tổng quát hóa dùng graph này (xem hình 3.1). Quan hệ mờ R, định nghĩa trong không gian tích Cartesian của các biến hệ thống $X_1 \times X_2 \times \cdot \cdot \cdot X_p \times Y$ là khả năng phân bố (giới hạn) của sai biệt vào-ra $(x_1, x_2, \ldots, x_p, y)$. Phép α-cut của R có thể được biểu diễn dùng tập các tổ hợp vào-ra có thể có với mức độ lớn hơn hay bằng α.

Thí dụ 3.4 Tính quan hệ mờ cho mô hình ngôn ngữ của thí dụ 3.2. Đầu tiên ta rời rạc hóa các miền vào và ra, thí dụ: $X = \{0, 1, 2, 3\}$ và $Y = \{0, 25, 50, 75, 100\}$. Các hàm thành viên rời rạc hóa được cho trong bảng 3.1 về các thừa số ngôn ngữ tiền đề và ghi các thừa số hệ quả trong bảng 3.2.

Bảng 3.1: Hàm thành viên hệ quả

	miền các phần tử			
thừa số ngôn ngữ	0	1	2	3
Low	1.0	0.6	0.0	0.0
OK	0.0	0.4	1.0	0.4
High	0.0	0.0	0.1	1.0

Bảng 3.2: Hàm thành viên hệ quả

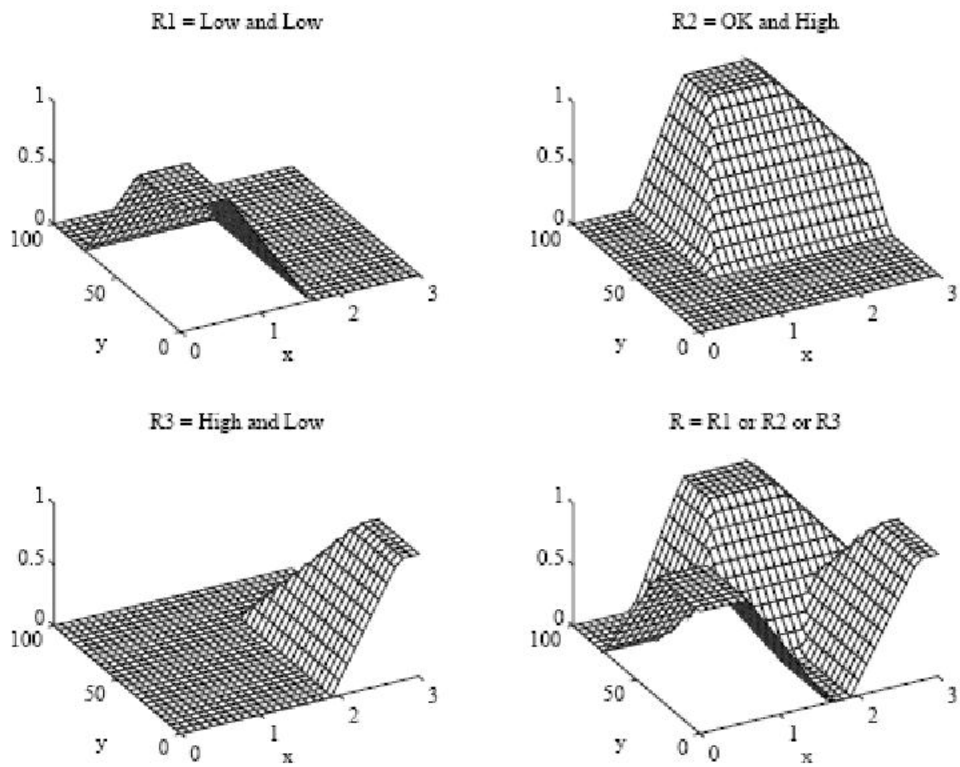
	miền các phần tử				
thừa số ngôn ngữ	0	25	50	75	100
Low	1.0	1.0	0.6	0.0	0.0
High	0.0	0.0	0.3	0.9	1.0

Quan hệ mờ R_i tương ứng cho từng luật, có thể được tính dùng (3.9). Trường hợp luật $R_1 = Low \times Low$, trường hợp R_2, ta có $R_2 = OK \times High$, và cuối cùng cho luật R_3, $R_3 = High \times Low$. Quan hệ mờ R, biểu diễn toàn thể luật nền, là phép hội (element-wise maximum) của các quan hệ R_i.

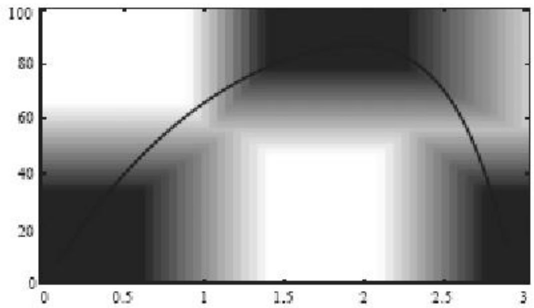
$$
\left.
\begin{aligned}
R_1 &= \begin{bmatrix} 1.0 & 1.0 & 0.6 & 0 & 0 \\ 0.6 & 0.6 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
R_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0.4 & 0.4 \\ 0 & 0 & 0.3 & 0.9 & 1.0 \\ 0 & 0 & 0.3 & 0.4 & 0.4 \end{bmatrix} \\
R_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0 & 0 \\ 1.0 & 1.0 & 0.6 & 0 & 0 \end{bmatrix}
\end{aligned}
\right\}
\quad R = \begin{bmatrix} 1.0 & 1.0 & 0.6 & 0 & 0 \\ 0.6 & 0.6 & 0.6 & 0.4 & 0.4 \\ 0.1 & 0.1 & 0.3 & 0.9 & 1.0 \\ 1.0 & 1.0 & 0.6 & 0.4 & 0.4 \end{bmatrix}
\tag{3.21}
$$

Các bước này được minh họa trong hình 3.6. Để thấy rõ hơn, cần tính quan hệ với bước rời rạc hóa mịn hơn trường hợp hàm thành viên của hình 3.3.
Thí dụ này có thể chạy trong MATLAB bằng cách gọi hàm script ling.

Hảy xét tập mờ vào của mô hình *A'* = [1, 0.6, 0.3, 0], có thể được xem là lưu tốc *Somewhat Low*, do gần với *Low* nhưng không bằng *Low*. Kết quả của tổ hợp max-min composition là tập mờ *B'* = [1, 1, 0.6, 0.4, 0.4], cho các kết quả mong muốn xấp xỉ *Low* của công suất nhiệt. Với *A'* = [0, 0.2, 1, 0.2] (approximately *OK*), ta có *B'* = [0.2, 0.2, 0.3, 0.9, 1], tức là, công suất approximately *High*. Xem phần kiểm tra các kết quả này xem như bài tập. Hình 3.7 vẽ graph mờ của thí dụ (vẽ contours của *R*, trong đó miền đánh bóng tương ứng với mức thành viên).



Hình 3.6: Quan hệ mờ R_1, R_2, R_3 tương ứng với từng luật, và quan hệ tính gộp R tương ứng của luật nền.



Hình 3.7 Một graph mờ của mô hình ngôn ngữ trong thí dụ 3.4 Vùng sậm màu tương ứng với mức độ thành viên cao. Đường sậm màu là hàm crisp có thể nhằm biểu diễn quan hệ thương tự là mô hình mờ

2.2 Suy diễn Max-min

Ta đã thấy là luật nền có thể được biểu diễn như quan hệ mờ. Ngõ ra của luật nền được tính từ tổ hợp quan hệ max-min. Chứng minh được là khi dùng fuzzy implications với các ngõ vào crisp, và dùng *t*-norms khi có ngõ vào là crisp và mờ, thì sơ đồ suy diễn có thể đơn giản hóa, dùng phép toán quan hệ (Jager, 1995). Điều này rất có lợi, do tránh được việc rời rạc hóa miền và việc lưu trữ quan hệ *R*. Trường hợp *t*-norm, việc đơn giản hóa đưa đến dạng sơ đồ nổi tiếng, được gọi là max-min hay phép suy diễn Mamdani, như phần trình bày dưới đây.

Giả sử giá trị mờ vào x = *A'*, và ngõ ra *B'* được cho bởi tổ hợp quan hệ:

$$\mu_{B'}(y) = \max_X \left[\mu_{A'}(x) \wedge \mu_R(x,y)\right]. \tag{3.22}$$

Sau khi thế μ_R(x, y) từ (3.20), có được:

$$\mu_{B'}(y) = \max_X \left\{\mu_{A'}(x) \wedge \max_{1\le i\le K}\left[\mu_{Ai}(x) \wedge \mu_{Bi}(y)\right]\right\}. \tag{3.23}$$

Toán tử max và min được thực hiện trong nhiều miền khác nhau, nên thay đổi được thứ tự như sau::

$$\mu_{B'}(y) = \max_{1\le i\le K} \left\{\max_X\left[\mu_{A'}(x) \wedge \mu_{Ai}(x)\right] \wedge \mu_{Bi}(y)\right\}. \tag{3.24}$$

Gọi $\beta_i = \max_X[\mu_{A'}(x) \wedge \mu_{Ai}(x)]$ là mức hoàn thành (*degree of fulfillment*) của luật tiền đề thứ *i*. Tập ra mờ của mô hình ngôn ngữ là:

$$\mu_{B'}(y) = \max_{1\le i\le K}\left[\beta_i \wedge \mu_{Bi}(y)\right], \qquad y \in Y. \tag{3.25}$$

Thuật toán *max-min* (*Mamdani*), tóm tắt trong Algorithm 3.1 và vẽ tại hình 3.8.

Algorithm 3.1 *Suy diễn Mamdani (max-min)*

1. Tính mức hoàn thành của từng luật dùng : $\beta_i = \max_X[\mu_{A'}(x) \wedge \mu_{Ai}(x)]$, $1 \le i \le K$. Chú ý trong tập singleton ($\mu_{A'}$ (x) = 1 với x = x_0 and $\mu_{A'}$ (x) = 0 trong các trường hợp khác) thì β_i được đơn giản thành $\beta_i = \mu_{Ai}(x_0)$.

2. Tìm tập ra mờ $B_i^{'}$: $\mu_{B'i}(y) = \beta_i \wedge \mu_{Bi}(y)$, $y \in Y$, $1 \le i \le K$.

3. Tính gộp các tập ra mờ B'_i : $\mu_{B'}(y) = \max\limits_{1 \le i \le K} \mu_{B'i}(y)$, $y \in Y$.

Thí dụ 3.5 Lấy tập mờ vào A' = [1, 0.6, 0.3, 0] từ bảng 3.4 và tính tập ra mờ tương ứng dùng phương pháp suy diễn Mamdani.

Bước 1 tìm được các mức hoàn thành sau:

$$\beta_1 = \max_X[\mu_{A'}(x) \wedge \mu_{A1}(x)] = \max([1, 0.6, 0.3, 0] \wedge [1, 0.6, 0, 0]) = 1.0,$$

$$\beta_2 = \max_X[\mu_{A'}(x) \wedge \mu_{A2}(x)] = \max([1, 0.6, 0.3, 0] \wedge [0, 0.4, 1, 0.4]) = 0.4$$

$$\beta_3 = \max_X[\mu_{A'}(x) \wedge \mu_{A3}(x)] \beta 3 = \max([1, 0.6, 0.3, 0] \wedge [0, 0, 0.1, 1]) = 0.1 .$$

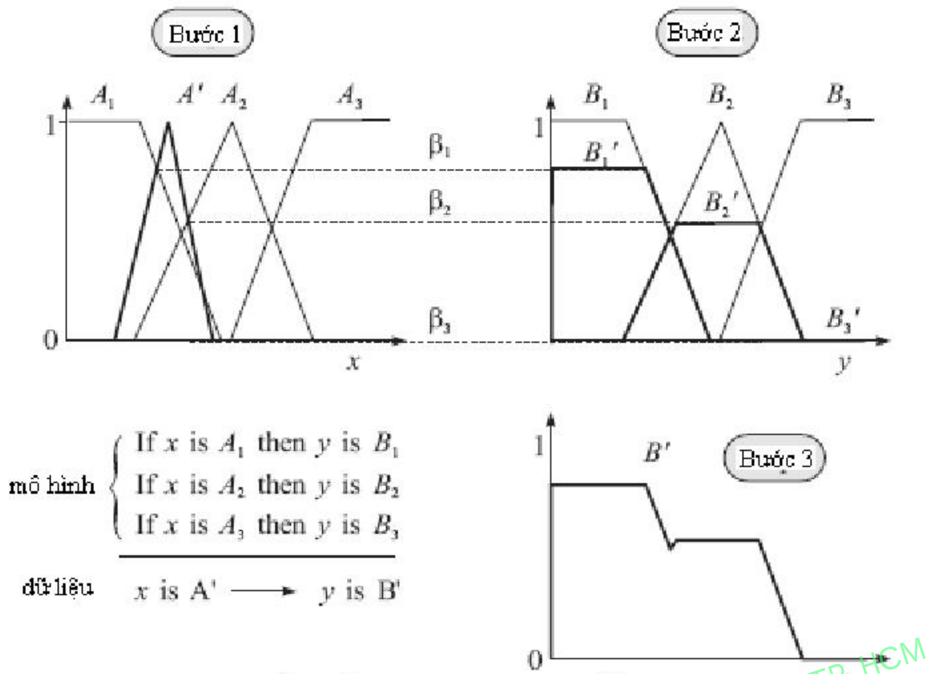
Trong bước 2, từng tập mờ hệ quả được tính:

$B'_1 = \beta_1 \wedge B_1$ = 1.0 $\wedge$ [1, 1, 0.6, 0, 0] = [1, 1, 0.6, 0, 0],
$B'_2 = \beta_2 \wedge B_2$ = 0.4 $\wedge$ [0, 0, 0.3, 0.9, 1] = [0, 0, 0.3, 0.4, 0.4],
$B'_3 = \beta_3 \wedge B_3$ = 0.1 $\wedge$ [1, 1, 0.6, 0, 0] = [0.1, 0.1, 0.1, 0, 0] .

Cuối cùng, bước 3 cho tập mờ ngõ ra:

$$B' = \max_{1 \le i \le K} \mu_{B'i} = [1, 1, 0.6, 0.4, 0.4]$$

Tương tự như kết quả từ thí dụ 3.4. Bài tập xem ngõ vào thứ hai của tập mờ trong thí dụ 3.4.

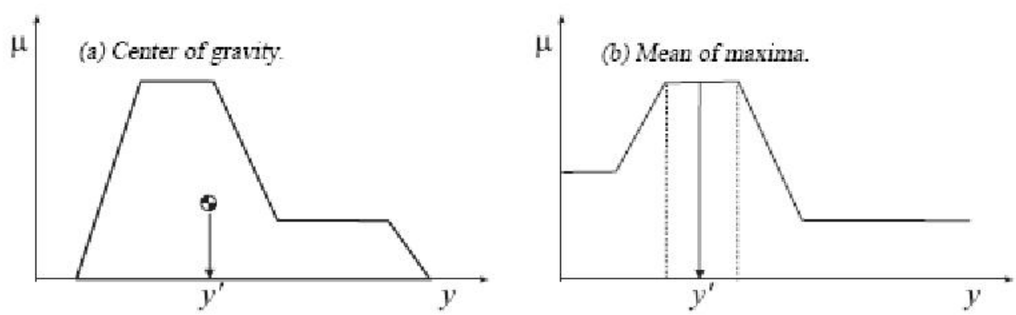


Hình 3.8 Biểu diễn thuật toán suy diễn Mamdani

Từ so sánh số lượng phép toán trong thí dụ 3.4 và 3.5, ta thấy tác động giảm phép tính của suy diễn Mamdani so với tổ hợp quan hệ là không đáng kể. Tuy nhiên, điều này chỉ đúng khi rời rạc hóa thô (rough discretization) như trường hợp của thí dụ 3.4 và trường hợp số ngõ vào là ít (trường hợp này là một). Chú ý là phương pháp suy diễn Mamdani không cần có bất kỳ phép rời rạc hóa nào nên có thể hoạt động được với các hàm thành viên dạng giải tích. Ngoài ra, phương pháp này còn cho phép dùng các luật học, như trình bày trong chương 5.

2.3 Giải mờ

Kết quả của suy diễn mờ là tập mờ *B'*. Nếu giá trị ra là dạng crisp (dạng số học), cần có giá trị ngõ ra, tập mờ ra cần được giải mờ (*defuzzified*). Giải mờ là biến đổi nhằm thay thế tập mờ bằng một giá trị số học biểu diễn tập này. Hình 3.9 vẽ hai phương pháp giải mờ thường dùng là: trọng tâm (center of gravity: COG) và trung bình cực đại (mean of maxima:MOM).



Hình 3.9 Các phương pháp giải mờ: trọng tâm và trung bình -cực đại

Phương pháp COG tính toán số học tọa độ y của trọng tâm tập mờ *B'*:

$$y' = cog(B') = \frac{\sum_{j=1}^{F} \mu_{B'}(y_j) y_j}{\sum_{j=1}^{F} \mu_{B'}(y_j)} \tag{3.28}$$

Trong đó F là số phần tử y_j trong Y. Miền liên tục Y cần được rời rạc hóa để tính được trọng tâm.

Phương pháp MOM tính giá trị trung bình của khoảng dùng mức thành viên lớn nhất:

$$mom(B') = cog\left\{ y \,\middle|\, \mu_{B'}(y) = \max_{y \in Y} \mu_{B'}(y) \right\} \tag{3.29}$$

Phương pháp COG được dùng cho phép suy diễn max-min Mamdani, cung cấp phép nội suy giữa các hệ quả, tỉ lệ theo chiều của từng hệ quả. Điều này là cần thiết do tự thân phương pháp suy diễn Mamdani không nội suy, và việc dùng phương pháp MOM trong trường hợp này có thể tạo ra các ngõ ra dạng bước (step-wise). Phương pháp MOM được dùng với phép suy diễn có nền dùng hàm ý mờ (fuzzy implications), nhằm chọn được ngõ ra "tốt nhất có thể". Suy diễn dùng nội suy hàm ý, cung cấp các tập hệ quả đủ trùng lắp (Jager, 1995). Không dùng được trực tiếp phương pháp COG trong trường hợp này, do yếu tố bất định trong ngõ ra làm gia tăng mức thành viên, như thí dụ 3.3. Phương pháp COG sẽ cho kết quả không thích hợp.

Để tránh tích phân số trong phương pháp COG, thường dùng phương pháp cải tiến gọi là giải mờ dùng phương pháp trung bình mờ (*fuzzy-mean*). Tập hệ quả mờ được giải mờ đầu tiên, nhằm tìm đươc các giá trị crisp biểu diễn tập mờ, thí dụ dùng pương pháp trung bình-cực đại b_j = mom(B_j). Giá trị ra crisp được tính từ trung bình trọng lượng của b_j:

$$y' = \frac{\sum_{j=1}^{M} \omega_i b_j}{\sum_{j=1}^{M} \omega_i} \tag{3.30}$$

Trong đó M là số tập mờ B_j và ω_j là cực đại của mức hoàn thành β_i trong mọi luật có hệ quả B_j . Để có thể tính gộp tập mờ B', có thể tính ω_j dùng $\omega_j = \mu_{B'}(b_j)$. Phương pháp này bảo đãm tính nội suy tuyến tính giữa các b_j, với các hàm thành viên tiền đề được tuyến tính hóa từng đoạn. Điều này không giống như trường hợp của phương pháp COG, có tạo yếu tố phi tuyến, tùy theo dạng của hàm hệ quả (Jager, et al., 1992).

Từ việc giải mờ riêng lẽ được thực hiện ngoại tuyến (off line), yếu tố hình dáng và trùng lắp của tập mờ hệ quả không tạo ra ảnh hưởng, nên có thể được thay thế trực tiếp bằng các giá trị giải mờ (singletons), xem phần 3.3. Để có thể tính từng phần sai biệt giữa tập mờ hệ quả, dùng phương pháp giải mờ trung bình- mờ (fuzzy-mean defuzzification):

$$y' = \frac{\sum_{j=1}^{M} \gamma_i S_i b_j}{\sum_{j=1}^{M} \gamma_i S_i} \qquad (3.31)$$

Trong đó S_j là phần diện tích nằm dưới hàm thành viên B_j. Ưu điểm của phương pháp trung bình mờ (fuzzy-mean) (3.30) và (3.31) là các tham số b_j có thể được ước lượng dùng kỹ thuật ước lượng tuyến tính trình bày trong chương 5.

Thí dụ 3.6 Xét tập ra mờ $B' = [0.2,\ 0.2,\ 0.3,\ 0.9,\ 1]$ của thí dụ 3.4, trong đó miền ra là $Y = [0,\ 25,\ 50,\ 75,\ 100]$. Ngõ ra giải mờ có được từ công thức (3.28):

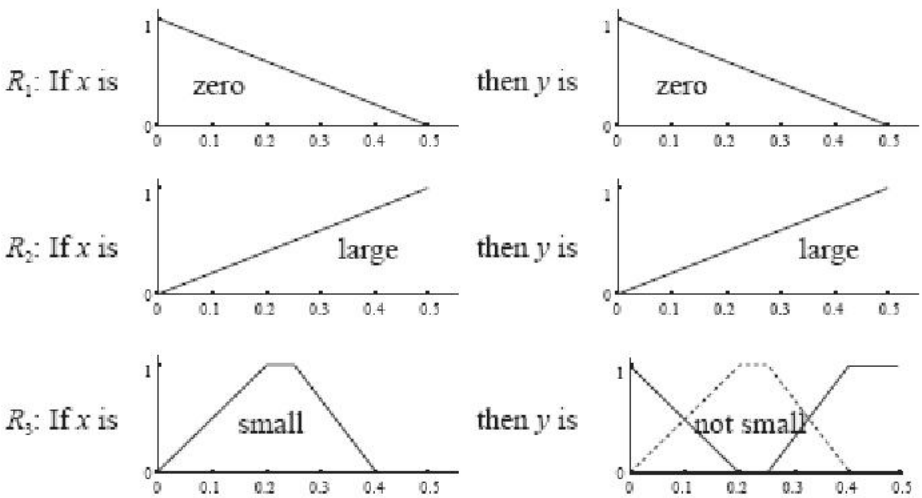
$$y' = \frac{0{,}2.0 + 0{,}2.25 + 0{,}3.50 + 0{,}9.75 + 1.100}{0{,}2 + 0{,}2 + 0{,}3 + 0{,}9 + 1} = 72{,}12$$

Công suất nhiệt của bộ đốt, tính từ mô hình mờ là 72.12W.

2.4 Hàm ý mờ và suy diễn Mamdani

Câu hỏi đặt ra là: Phương pháp suy diễn nào tốt hơn, hay trong trường hợp nào thì một phương pháp nào thích hợp hơn phương pháp khác? Để tìm đáp số, cần có một phân tích chi tiết về các phương pháp đã trình bày, điều này ngoài mục tiêu của tài liệu này. Tuy nhiên, ta có thể dùng các thí dụ minh họa sau.

Thí dụ 3.7 (Ưu điểm của hàm ý mờ) Xét luật nền vẽ ở hình 3.10. Các luật R_1 và R_2 biểu diễn quan hệ đơn điệu giản đơn (monotonic) (xấp xỉ tuyến tính) của hai biến.



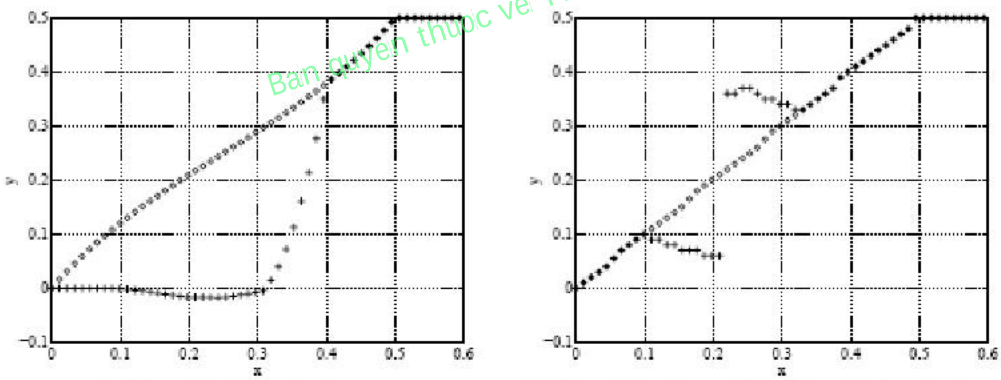
Hình 3.10 Luật nền được xem xét

Thí dụ việc thiết lập luật nền của luật điều khiển tỉ lệ. Luật R_3, "Nếu x là small thì y là not small", biểu diễn một dạng "ngoại lệ" từ quan hệ đơn giản của phép nội suy từ hai luật trước đó. Trong điều khiển, luật này có thể gặp các hiện tượng không mong muốn, như ma sát tĩnh. Thí dụ, khi điều khiển một động cơ điện có lực ma sát Coulomb lớn, khi đưa vào dòng điện bé vào động cơ, không quay được do không vượt qua lực ma sát được, mà chỉ tiêu tốn năng lượng. Ba luật này có thể xem là trường hợp đơn giản của kiến thức tổng quát cơ bản, các thông tin sâu hơn thì được dùng bổ sungtrong các thừa số ngoại lệ (terms of exceptions).

Hình 3.11a minh họa kết quả của phương pháp suy diễn Mamdani, dùng phép giải mờ COG. Ta thấy là phương pháp Mamdani chưa hoạt động tốt. Lý do là phép nội suy là do giải mờ tạo ra mà không phải từ tự thân cơ chế suy diễn. Sự hiện diện của luật thứ ba làm méo dạng đáng kể phần gốc, hầu như có đặc tính tuyến tính, cùng với vùng trong đó R_1 có mức thành viên lớn nhất. Mục tiêu là tránh các giá trị bé cho y đã không thực hiện được.

Hình 3.11b vẽ kết quả của suy diễn luận lý dùng phép hàm ý Łukasiewicz và phương pháp giải mờ MOM. Có thể thấy là luật thứ ba hoàn thành nhiệm vụ của mình, tức là làm cho hệ mờ thoát khỏi vùng có ngõ ra giá trị thấp (xung quanh 0.25) khi các giá trị vào thấp (xung quanh 0.25). Dạng chính xác của ánh xạ vào-ra tùy thuộc vào việc lựa chọn các toán tử suy diễn đặc thù (hàm ý, tổ hợp), nhưng đáp ứng chung vẫn giữa không đổi.

Cần thấy rằng, suy luận dùng phép hàm ý có một số yêu cầu về việc trùng lắp (overlap) của các hàm thành viên hệ quả, có thể rất khó thực hiện khi dùng nhiều ngõ vào (Jager, 1995). Hơn nữa, phương pháp này thường cần được thiết lập dùng các quan hệ mờ và luật suy diễn tổ hợp, làm tăng thêm yêu cầu về tính toán.



(a) Suy diễn Mamdani (b) Suy diễn dùng phép hàm ý Lukasiewicz

Hình 3.11 Ánh xạ vào-ra dùng luật nền hình 3.10 với hai phương pháp suy diễn khác nhau. Dấu hiệu 'o' chỉ vẽ ngõ giải mờ ra của các luật R_1 và R_2, dấu '+' vẽ ngõ ra giải mờ của luật chung

2.5 Các luật có nhiều ngõ vào, Liên kết logic

Trước đây, chỉ giới thiệu mô hình ngôn ngữ theo cách thông thường gồm các trường hợp SISO và MIMO. Trường hợp MIMO, tất cả các tập mờ trong mô hình được định nghĩa trong miền vectơ dùng hàm thành viên nhiều biến (multivariate membership functions). Tuy nhiên, để tiện thì nên viết các mệnh đề tiền đề và hệ quả thành tổ hợp của các mệnh đề mờ có các hàm thành viên đơn biến (univariate membership functions). Toán tử logic mờ (liên kết: connectives), như là conjunction, disjunction và negation (phép bù), có thể được dùng tổ hợp các mệnh đề này.

Kết nối *and* và *or* được thiết lập dùng lần lượt phép *t*-norms và *t*-conorms. Có vô số phép *t*-norms và *t*-conorms, nhưng thực tế thì chỉ có một số toán tử là được dùng nhiều. Bảng 3.3 liệt kê ba dạng thông dụng nhất.

Việc lựa chọn *t*-norms và *t*-conorms cho logic kết nối phụ thuộc vào ý nghĩa và ngữ cảnh của các mệnh đề. Các toán tử *max* và *min* do Zadeh đề nghị thì bỏ qua yếu tố dư thừa (redundancy), thí dụ trong phép tổ hợp (conjunction or disjunction) thì dùng hai mệnh đề mờ giống nhau để giới thiệu cùng một mệnh đề:

$$\mu_{A\cap A}(x) = \mu_A(x) \wedge \mu_A(x) = \mu_A(x), \tag{3.32}$$

$$\mu_{A\cup A}(x) = \mu_A(x) \vee \mu A(x) = \mu A(x). \tag{3.33}$$

Bảng 3.3 Các toán tử thường dùng mô tả phép kết nối **and** và **or**

and	or	name
$\min(a, b)$	$\max(a, b)$	Zadeh
$\max(a + b - 1, 0)$	$\min(a + b, 1)$	Łukasiewicz
ab	$a + b - ab$	probability

Điều này không đúng với các *t*-norms và *t*-conorms khác. Tuy nhiên, khi các mệnh đề mờ không bằng nhau, nhưng chúng tương quan hay tương tác với nhau, thì có thể dùng các toán tử khác như *min* và *max*.

Nếu các mệnh đề liên quan đến các vũ trụ khác nhau, thì kết nối logic tạo ra tập mờ nhiều biến. Xét mệnh đề sau:

$$P : x_1 \text{ là } A_1 \text{ và } x_2 \text{ là } A_2$$

Trong đó A_1 và A_2 có hàm thành viên $\mu_{A1}(x_1)$ và $\mu_{A2}(x_2)$. Mệnh đề *p* có thể được biểu diễn dùng tập mờ *P* có hàm thành viên:

$$\mu_P(x_1, x_2) = \mathrm{T}(\mu_{A1}(x_1), \mu_{A2}(x_2)), \tag{3.35}$$

trong đó *T* là *t*-norm nhằm mô hình kết nối *and*. Tổ hợp các mệnh đề lại là một mệnh đề.

Phủ định trong mệnh đề mờ có liên quan đến phép bù của tập mờ.
Với mệnh đề

$$P : x \text{ là not } A$$

Phép bù chuẩn (standard complement) cho kết quả:

$$\mu_P(x) = 1 - \mu_A(x)$$

Thường gặp nhất là dạng *conjunctive form* của tiền đề, được cho từ:

$$\mathfrak{R}_i\text{: Nếu } x_1 \text{ là } A_{i1} \text{ và } x_2 \text{ là } A_{i2} \text{ và} \ldots \text{và } x_p \text{ là } A_{ip} \text{ thì } y \text{ là } B_i, \quad i = 1, 2, \ldots, K. \tag{3.36}$$

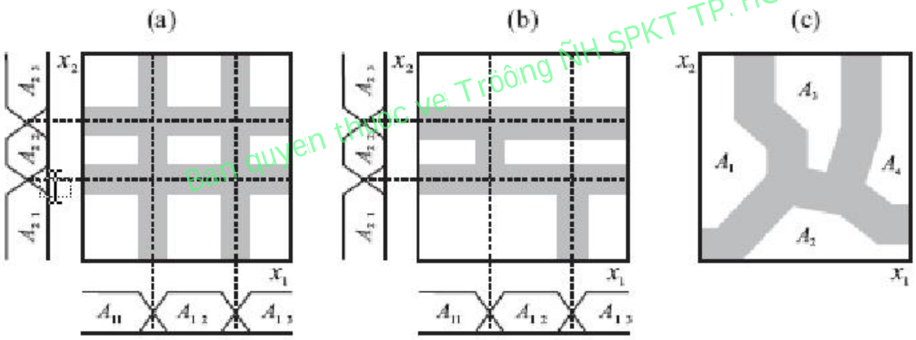
Chú ý là mô hình trên là trường hợp đặc biệt của (3.1), với tập mờ A_i trong (3.1) có được từ tích conjunction Cartesian của tập mờ A_{ij}: $A_i = A_{i1} \times A_{i2} \times \cdots \times A_{ip}$. Như thế, khi ngõ vào là crisp thì mức độ hoàn thành (bước 1 trong Algorithm 3.1) được cho bởi:

$$\beta_i = \mu_{Ai1}(x_1) \wedge \mu_{Ai2}(x_2) \wedge \cdots \wedge \mu_{Aip}(x_p), \quad 1 \le i \le K. \tag{3.38}$$

Tập các luật trong dạng tiền đề conjunctive chia miền ngõ thành mắt lưới (lattice) của hyperboxes mờ, song song với các trục. Từng hyperboxes là không gian tích Cartesian giao (intersection) với tập mờ univariate tương ứng. Điều này được vẽ ở hình 3.12a. Số luật trong dạng conjunctive, cần được phủ hết miền, cho bởi:

$$K = \prod_{i=1}^{p} N_i$$

Trong đó p là miền của không gian vào và N_i là số thừa số ngôn ngữ của biến tiền đề thứ i.



Hình 3.12 Các partition khác nhau của không gian tiền đề. Miền có màu xám định nghĩa vùng trùng lắp (overlapping)của tập mờ

Bằng cách kết hợp các phép conjunctions, disjunctions và negations, có thể tìm được nhiều partitions khác nhau của không gian tiền đề, tuy nhiên, các đường biên bị giới hạn trong các lưới vuông được định nghĩa từ các tập mờ của từng biến, như trong hình 3.12b. Thí dụ, xét luật tiền đề phủ góc trái phía dưới của không gian tiền đề trong hình này:

Nếu x_1 là not A_{13} và x_2 là A_{21} thì . . .

Mức hoàn thành của luật này được tính dùng phép bù và phép giao:

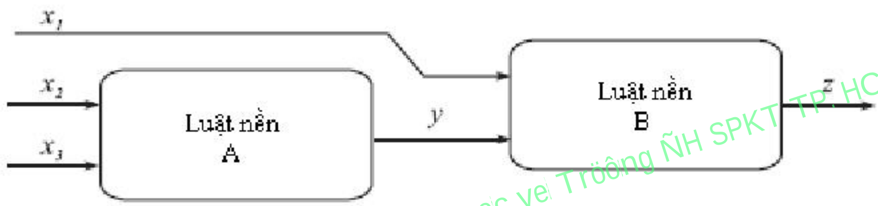
$$\beta = [1 - \mu_{A13}(x_1)] \wedge \mu_{A21}(x_2). \tag{3.39}$$

Dạng tiền đề có các hàm thành viên multivariate (3.1) là một dạng tổng quát nhất, do không có hạn chế về hình dạng của vùng mờ. Các biên giới giữa các vùng này có thể là đường cong bất kỳ và nằm xiên so với các trục, như vẽ trong hình 3.12c. Ngoài ra, một số các tập mờ cần thiết để phủ không gian tiền đề có thể nhỏ hơn rất nhiều so với trường hợp trước đó. Như thế, trong hệ multivariable phức tạp thì phương pháp biểu diễn dùng partition có lẽ là phương pháp hiệu quả nhất. Chú ý là các tập mờ từ A_1 đến

A_4 trong hình 3.12c vẫn có thể chiếu vào trong X_1 và X_2 để có diễn đạt ngôn ngữ của vùng cần mô tả.

2.6 Xâu chuỗi luật (Rule Chaining)

Cho đến nay, chỉ mới khảo sát cấu trúc một lớp của mô hình mờ. Tuy nhiên, trong thực tế thì ngõ ra của một luật có thể dùng làm ngõ vào của luật nền khác. Điều này tạo ra cấu trúc nhiều lớp và luật xâu chuỗi (chained rules). Thí dụ trường hợp này xuất hiện trong các mô hình dạng phân cấp hay bộ điều khiển bao gồm nhiều luật nền. Tổ chức phân cấp về tri thức thường được dùng như hướng tự nhiên trong rút gọn độ phức tạp. Luật nền lớn có thể phân chia nhiều biến vào thành nhiều luật liên kết nối có số ngõ vào ít hơn. Thí dụ, giả sử luật nền có ba ngõ vào, mỗi ngõ vào gồm năm thừa số ngôn ngữ. Dùng dạng conjunctive (3.36), định nghĩa được 125 luật nhằm phủ tất cả các tình trạng ngõ vào. Chia các luật nền thành hai luật nền nhỏ hơn, như vẽ ở hình 3.13, ta có tổng số là 50 luật.



Hình 3.13 Kết nối chồng (cascade) hai luật nền

Một thí dụ khác về rule chaining là mô phỏng hệ thống mờ động, trong đó kết nối đuôi luật nền tạo ra thực tế giá trị dự báo bằng mô hình tạo thời điểm k được dùng làm ngõ vào tại thời gian $k + 1$. Xét mô hình hệ rời rạc phi tuyến.

$$\hat{x}(k+1) = f\left(\hat{x}(k), u(k)\right) \tag{3.40}$$

Trong đó f là ánh xạ thực hiện từ luật nền, $\hat{x}(k)$, là trạng thái dự báo của qua trình tại thời gian k (tại cùng thời gian với trạng thái của mô hình), và $u(k)$ là ngõ vào. Tại bước thời gian kế tiếp, ta có:

$$\hat{x}(k+2) = f\left(\hat{x}(k+1), u(k+1)\right) = f\left(f\left(\hat{x}(k), u(k)\right), u(k+1)\right) \tag{3.41}$$

Là dạng xâu chuỗi luật (cascade chain of rules).

Cấu trúc phân cấp của luật nền vẽ ở hình 3.13 đòi hỏi phải có thông tin suy ra từ Luật nền A để chuyển sang Luật nền B. Điều này thực hiện được bằng cách giải mờ tại ngõ ra của luật nền thứ nhất và phép giải mờ hệ quả tại ngõ vào của luật nền thứ hai. Yếu điểm của phương pháp này là hàm thành viên cần được định nghĩa tại biến trung gian và cần chọn lựa phương pháp giải mờ thích hợp. Nếu kiểm tra được giá trị biến trung gian bằng dữ liệu, thì chưa có phương pháp trực tiếp để kiểm tra xem lựa chọn đã thích hợp chưa. Đồng thời, mức mờ hóa tại ngõ ra của tầng thứ nhất được gở bỏ bằng phép giải mờ và phép giải mờ kế tiếp. Phương pháp này được dùng chủ yếu trong mô phỏng hệ thống động, như (3.41), khi biến trung gian được cùng lúc dùng làm ngõ ra crisp của hệ thống.

Một khả năng khác là đưa trực tiếp tập mờ tại ngõ ra của luật nền thứ nhất (không cần giải mờ) vào luật nền thứ hai. Ưu điểm của phương pháp này là không cần thêm bất kỳ thông tin nào từ người dùng. Tuy nhiên, trong trường hợp tổng quát, thì tổ hợp quan hệ cần thực hiện phép rời rạc hóa các miền và các thiết lập thường phức tạp. Trong trường hợp phép suy diễn max-min Mamdani, thì phép suy luận có thể được đơn giản hóa, do mức thành viên của tập ra mờ trực tiếp trở thành mức thành viên của mệnh đề tiền đề trong đó xuất hiện các thừa số ngôn ngữ đặc thù. Thí du, giả sử suy luận trong Luật nền A tạo mức hoàn thành tính gộp của thừa số ngôn ngữ B_1 đến B_5:

$$\omega = [0/B_1,\ 0.7/B_2,\ 0.1/B_3,\ 0/B_4,\ 0/B_5].$$

Mức thành viên của mệnh đề “Nếu y là B_2” trong luật nền B là 0.7, mức thành viên của mệnh đề “Nếu y là B_3” là 0.1, và các mệnh đề với các thừa số ngôn ngữ còn lại có mức thành viên là zero.

3. Mô hình Singleton

Một trường hợp đặc biệt của mô hình ngôn ngữ mờ khi tập mờ hệ quả B_i là tập singleton. Các tập này có thể được biểu diễn dùng các số thực b_i, có được từ các luật sau:

$$\Re_i: \text{Nếu } x \text{ là } A_i \text{ thì } y = b_i, \quad i = 1, 2, \ldots, K. \tag{3.42}$$

Mô hình này được gọi là mô hình *singleton*. Khác với mô hình ngôn ngữ, số lượng các singletons phân biệt trong luật nền thường không bị giới hạn, tức là mỗi luật có thể có các singleton hệ quả riêng. Trong mô hình, phương pháp giải mờ COG tạo ra trong phương pháp trung bình-mờ (*fuzzy-mean method)*:

$$y = \frac{\sum_{i=1}^{K} \beta_i b_i}{\sum_{i=1}^{K} \beta_i}. \tag{3.43}$$

Chú ý rằng tất cả K luật đều đóng góp cho việc giải mờ, khác với phương pháp ở (3.30). Điều này có nghĩa là nếu có hai luật có cùng hệ quả singleton đều tích cực, thì singleton được tính hai lần trong trung bình trọng lượng (3.43).
Khi dùng (3.30), mỗi hệ quả sẽ chỉ tính một lần khi trọng lượng bằng hay lớn hơn hai mức độ hoàn thành. Chú ý là mô hình singleton có thể được xem là trường hợp đặc biệt của mô hình Takagi–Sugeno, giới thiệu trong phần 3.5.

Ưu điểm của mô hình singleton so với mô hình ngôn ngữ là các tham số hệ quả b_i có thể được tính dễ dàng từ dữ liệu, dùng kỹ thuật bình phương tối thiểu.
Mô hình mờ singleton thuộc vào nhóm chung các hàm xấp xỉ tổng quát, được gọi là *khai triển hàm cơ sở*, (Friedman, 1991), có dạng:

$$y = \sum_{i=1}^{K} \phi_i(x) b_i. \tag{3.44}$$

Đa số cấu trúc dùng trong hệ nhận dạng phi tuyến, như mạng nơrôn nhân tạo, mạng hàm radial basis, hay splines, tùy theo dạng hệ thống. Trong mô hình singleton, hàm cơ sở $\phi_i(\mathrm{x})$ được cho bởi mức hoàn thành chuẩn hóa của luật tiền đề, và các hệ số b_i là hệ quả. Nội suy đa tuyến tính giữa các luật hệ quả có được nếu:

- hàm thành viên tiền đề có dạng tam giác, các cặp trùng lắp (pairwise overlapping) và mức thành viên tính tổng đến một cho mỗi miền thành phần.
- toán tử tích được dùng biểu diễn phép and kết nối trong luật tiền đề.

Thí dụ về (univariate) được vẽ ở hình 3.14a.

Rõ ràng, mô hình singleton có thể dùng biểu diễn có thể biểu diễn ánh xạ tuyến tính có dạng:

$$y = k^T x + q = \sum_{i=1}^{p} k_i x_i + q \tag{3.45}$$

Trong trường hợp này, hàm thành viên tiền đề phải là dạng tam giác. Hệ quả singletons có thể tính dùng cách ước lượng ánh xạ mong muốn (3.45) cho lõi cores a_{ij} của tập mờ tiền đề A_{ij} (xem hình 3.14b):

$$b_i = \sum_{j=1}^{p} k_j a_{ij} + q \tag{3.46}$$

Đặc tính này là hữu ích, do mô hình mờ có thể được khởi tạo sao cho bắt chước được mô hình tuyến tính hay bộ điều khiển cho trước (có thể là không chính xác) và có thể được tối ưu hóa sau.

4. Mô hình quan hệ

Mô hình quan hệ mờ (Pedrycz, 1985; Pedrycz, 1993) mã hóa tương quan giữa các thừa số ngôn ngữ được định nghĩa trong ngõ vào của hệ và miền ra dùng quan hệ mờ. Các thành phần riêng của quan hệ biểu diễn cường độ tương quan (strength of association) giữa các tập mờ. Trước hết, hảy xem xét mô hình mờ dạng ngôn ngữ gồm các luật sau:

$\mathfrak{R}_i$: Nếu x_1 là $A_{i,1}$ và ... và x_n là $A_{i,n}$ thì y là B_i, $i = 1, 2, \ldots, K$. (3.47)

Gọi A_j là tập các thừa số ngôn ngữ định nghĩa tập các biến tiền đề x_j :

$$A_j = \{A_{j,l} \mid l = 1, 2, \ldots, N_j\}, \quad j = 1, 2, \ldots, n,$$

Trong đó $\mu_{A_{j,l}}(xj): X_j \to [0, 1]$. Tương tự, tập các thừa số ngôn ngữ định nghĩa tập các biến ra y:

$$B = \{B_l \mid l = 1, 2, \ldots, M\},$$

Với $\mu_{Bl}(y)$: $Y \rightarrow [0, 1]$. Điểm chủ yếu để hiểu được nguyên lý về các mô hình mờ dạng quan hệ là thực hiện luật nền (3.47) được biểu diễn thành quan hệ *crisp* S giữa tập các thừa số A_j và thừa số hệ quả B:

$$S: A_1 \times A_2 \times \cdots \times A_n \times B \rightarrow \{0, 1\}. \qquad (3.48)$$

Gọi $A = A_1 \times A_2 \times \cdots \times A_n$ là không gian Cartesian các thừa số ngôn ngữ, (3.48) đơn giản thành S: $A \times B \rightarrow \{0, 1\}$. Chú ý là nếu luật đã định nghĩa mọi khả năng tổ hợp của các thừa số tiền đề, K = card(A). Thì S có thể được biểu diễn thành ma trận $K \times M$, chỉ có ràng buộc là có phần tử khác không trong từng hàng.

Thí dụ 3.8 (Biểu diễn theo quan hệ của luật nền) Xét mô hình hệ mờ có hai ngõ vào x_1, x_2, và một ngõ ra y. Định nghĩa hai thừa số ngôn ngữ cho từng ngõ vào:
$A_1 = \{Low, High\}$, $A_2 = \{Low, High\}$, và ba thừa số cho ngõ ra:
$B = \{Slow, Moderate, Fast\}$. Từ mọi khả năng tổ hợp các thừa số tiền đề, có được bốn luật (các hệ quả được chọn bất kỳ):

Nếu x_1 là *Low* và x_2 là *Low* thì y là *Slow*
Nếu x_1 là *Low* và x_2 là *High* thì y là *Moderate*
Nếu x_1 là *High* và x_2 là *Low* thì y là *Moderate*
Nếu x_1 là *High* và x_2 là *High* thì y là *Fast*.

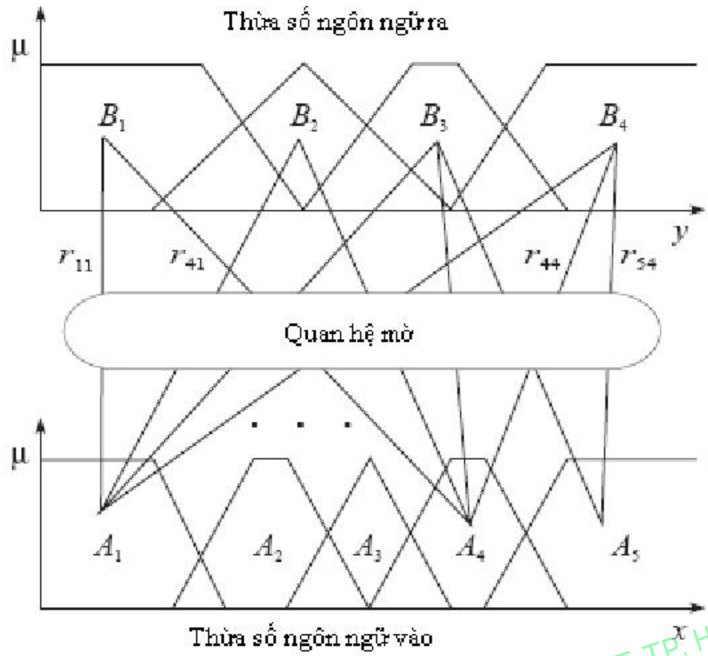
Trong thí dụ này thì $A = \{(Low, Low), (Low, High), (High, Low), (High, High)\}$. Các luật trên có thể được biểu diễn dùng ma trận quan hệ S:

			y	
x_1	x_2	*Slow*	*Moderate*	*Fast*
Low	*Low*	1	0	0
Low	*High*	0	1	0
High	*Low*	0	1	0
High	*High*	0	0	1

Mô hình quan hệ mờ là mở rộng của quan hệ crisp S thành quan hệ mờ $R = [r_{i,j}]$

$$R: A \times B \rightarrow [0, 1]. \qquad (3.49)$$

Mỗi luật đang chứa đựng tất cả các khả năng của thừa số hệ quả, với từng thừa số trọng lượng riêng, lần lượt được cho bởi các thành phần r_{ij} trong quan hệ mờ (Hình 3.15). Các trọng lượng này cho phép mô hình được tinh chỉnh tốt hơn nhằm khớp được với dữ liệu.



Hình.15: Xem quan hệ mờ là ánh xạ giữa các thừa số ngôn ngữ vào và ra

Điều cần nhấn mạnh là quan hệ R trong (3.49) khác với quan hệ trong quan hệ mã hóa ngôn ngữ luật nếu-thì (3.19). Quan hệ sau là hàm thành viên nhiều chiều định nghĩa trong không gian tích của các miền ngõ vào và ngõ ra, trong đó các phần tử biểu diễn mức tương quan giữa từng phần tử crisp riêng trong miền tiền đề và hệ quả. Tuy nhiên trong mô hình quan hệ mờ thì quan hệ biểu diễn tương quan giữa từng thừa số ngôn ngữ riêng lẽ.

Thí dụ 3.9 Mô hình quan hệ. Dùng thừa số ngôn ngữ của thí dụ 3.8, định nghĩa mô hình quan hệ mờ dùng quan hệ R như sau:

x_1	x_2	*y* *Slow*	 *Moderate*	 *Fast*
Low	*Low*	0.9	0.2	0.0
Low	*High*	0.0	1.0	0.0
High	*Low*	0.0	0.8	0.2
High	*High*	0.0	0.1	0.8

Các phần tử $r_{i,j}$ mô tả quan hệ giữa tổ hợp các thừa số ngôn ngữ tiền đề và thừa số ngôn ngữ hệ quả. Điều này hàm ý là hệ quả không phải chính xác là bằng các thừa số ngôn ngữ đã được định nghĩa trước, nhưng cho bởi các tổ hợp trọng lượng. Chú ý là tổng các trọng lượng không bắt buộc phải bằng một. Quan hệ này được biểu diễn theo dạng luật như sau:

Nếu x_1 là *Low* và x_2 là *Low* thì y là *Slow* (0.9), y là *Mod.* (0.2), y là *Fast* (0.0)
Nếu x_1 là *Low* và x_2 là *High* thì y là *Slow* (0.0), y là *Mod.* (1.0), y là *Fast* (0.0)
Nếu x_1 là *High* và x_2 là *Low* thì y là *Slow* (0.0), y là *Mod.* (0.8), y là *Fast* (0.2)
Nếu x_1 là *High* và x_2 là *High* thì y là *Slow* (0.0), y là *Mod.* (0.1), y là *Fast* (0.8)

Các số trong dấu ngoặt lần lượt là các phần tử $r_{i,j}$ của R.

Phép duy diễn dùng cơ sở là tổ hợp quan hệ (2.45) của tập mờ biểu diễn mức hoàn thành β_i và quan hệ R, và được cho trong thuật toán sau.

Algorithm 3.2 *Suy diễn trong mô hình quan hệ mờ.*

1. Tính mức độ hoàn thành:

$$\beta_i = \mu_{Ai1}(x_1) \wedge \cdots \wedge \mu_{Aip}(x_p), \quad i = 1, 2, \ldots, K. \tag{3.50}$$

(Có thể dùng các toán tử giao khác, như phép tích.)

2. Dùng tổ hợp quan hệ $\omega = \beta \circ R$, cho bởi:

$$\omega_j = \max_{1 \le i \le K}(\beta_i \wedge r_{ij}), \qquad j = 1, 2, \ldots, M. \tag{3.51}$$

3. Giải mờ tập hệ quả dùng:

$$y = \frac{\sum_{j=1}^{M} \omega_l . b_l}{\sum_{j=1}^{M} \omega_l} \tag{3.52}$$

Trong đó b_l là trọng tâm của tập mờ hệ quả B_l tính được dùng các phương pháp giải mờ như trọng tâm (3.28) hay dùng phương pháp trung bình-cực đại (3.29) của từng tập mờ riêng lẽ B_l.

Chú ý là nếu R là crisp, thì dùng suy diễn Mamdani với phương pháp giải mờ trung bình –mờ (3.30).

Thí dụ 3.10 (Suy diễn) Giả sử khi dùng luật nền trong thí dụ 3.8, ta có mức thành viên như sau:

$$\mu_{\text{Low}}(x_1) = 0.9, \quad \mu_{\text{High}}(x_1) = 0.2, \quad \mu_{\text{Low}}(x_2) = 0.6, \quad \mu_{\text{High}}(x_2) = 0.3,$$

với các ngõ vào cho trước x_1 và x_2. Để suy ra y, đầu tiên dùng phương trình (3.50) để tìm β. Dùng tích t-norm, có được các giá trị sau:

$$\beta_1 = \mu_{\text{Low}}(x_1) \cdot \mu_{\text{Low}}(x_2) = 0.54 \qquad \beta_2 = \mu_{\text{Low}}(x_1) \cdot \mu_{\text{High}}(x_2) = 0.27$$
$$\beta_3 = \mu_{\text{High}}(x_1) \cdot \mu_{\text{Low}}(x_2) = 0.12 \qquad \beta_4 = \mu_{\text{High}}(x_1) \cdot \mu_{\text{High}}(x_2) = 0.06$$

Như thế, mức hoàn thành: $\beta = [0.54, 0.27, 0.12, 0.06]$. Dùng phương trình (3.51) để tìm tập mờ ra ω:

$$\omega = \beta \circ R = \begin{bmatrix} 0.54 & 0.27 & 0.12 & 0.06 \end{bmatrix} \circ \begin{bmatrix} 0.9 & 0.2 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.8 & 0.2 \\ 0.0 & 0.1 & 0.8 \end{bmatrix}. \qquad (3.55)$$

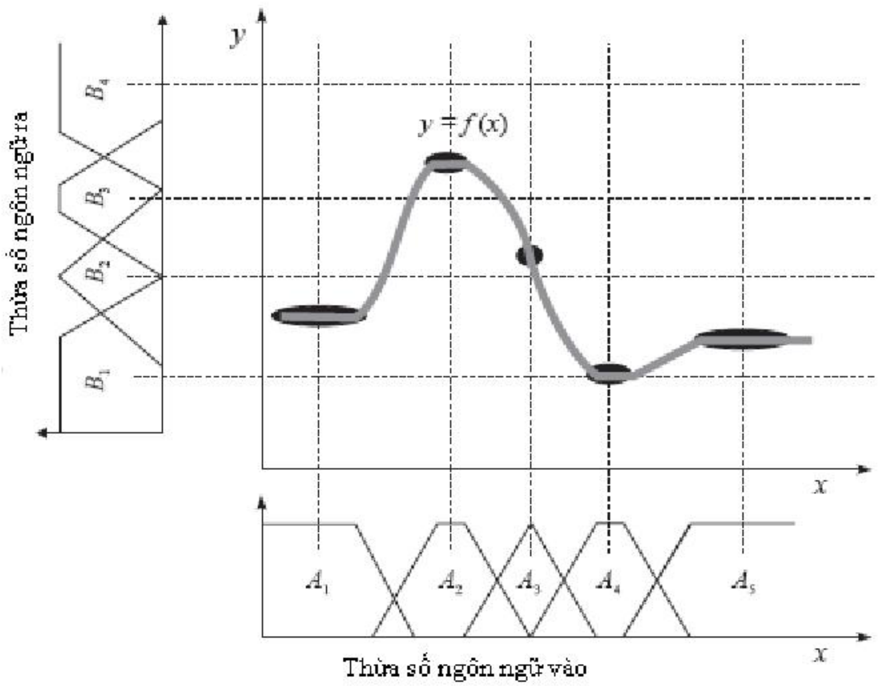
Sau cùng, dùng phương trình (3.52), tính ngõ ra giải mờ là:

$$y = \frac{0.54cog(Slow) + 0.27\cot(Moderate) + 0.12\cot(Fast)}{0.54 + 0.27 + 0.12}. \qquad (3.56)$$

Ưu điểm chủ yếu của mô hình quan hệ là tinh chỉnh được ánh xạ vào-ra mà không cần thay đổi các tập mờ hệ quả (thừa số ngôn ngữ). Trong mô hình ngôn ngữ, từ kết quả của từng luật riêng, giới hạn trong lưới các trọng tâm của các tập mờ ra, không phải là trường hợp của mô hình quan hệ, xem hình 3.16.

Đối với các bậc tự do cộng thêm này, tạo ra thêm nhiều tham số tự do (các phần tử trong quan hệ). Nếu không có ràng buộc của các tham số này, nhiều phần tử trong hàng của R có thể khác không, điều này có thể ngăn trở quá trình diễn đạt của mô hình. Hơn nữa, hình dáng của tập mờ ngõ ra không có ảnh hưởng lên giá trị giải mờ ngõ ra, do khi giải mờ thì chỉ quan tâm đến trọng tâm của các tập này.

Dễ dàng nhận thấy là nếu các tập mờ tiền đề có tổng lên đến một và dùng tổ hợp sum–product có chặn, thì có thể thay thế bằng mô hình tính toán hợp lý hơn dùng các hệ quả singleton (Voisin, et al., 1995).



Hình 3.16 Các khả năng ánh xạ của mô hình quan hệ mờ

Thí dụ 3.11 (Mô hình quan hệ và mô hình Singleton) Mô hình quan hệ mờ:

Nếu x là A_1 thì y là B_1 (0.8), y là B_2 (0.1), y là $B3$ (0.0).
Nếu x là A_2 thì y là B_1 (0.6), y là B_2 (0.2), y là $B3$ (0.0).

Nếu x là A_3 thì y là B_1 (0.5), y là B_2 (0.7), y là $B3$ (0.0).
Nếu x là A_4 thì y là B_1 (0.0), y là B_2 (0.1), y là $B3$ (0.9).

Có thể được thay thế dùng mô hình singleton:

Nếu x is A_1 thì $y = (0.8b_1 + 0.1b_2)/(0.8 + 0.1)$,
Nếu x is A_2 thì $y = (0.6b_1 + 0.2b_2)/(0.6 + 0.2)$,
Nếu x is A_3 thì $y = (0.5b_1 + 0.7b_2)/(0.5 + 0.7)$
Nếu x is A_4 thì $y = (0.1b_2 + 0.9b_3)/(0.1 + 0.9)$

Nếu các hàm thành viên hệ quả còn tạo thành partition, thì mô hình singleton có thể đảo ngược bằng cách diễn tả thành mô hình quan hệ tương đương thông qua phép tính mức thành viên của singletons trong tập mờ hệ quả B_j. Các mức thành viên này trở thành các phần tử của quan hệ mờ:

$$R = \begin{bmatrix} \mu_{B1}(b_1) & \mu_{B2}(b_1) & \dots & \mu_{BM}(b_1) \\ \mu_{B1}(b_2) & \mu_{B2}(b_2) & \dots & \mu_{BM}(b2) \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{B1}(b_K) & \mu_{B2}(b_K) & \dots & \mu_{BM}(b_K) \end{bmatrix}, \qquad (3.59)$$

Rõ ràng là mô hình ngôn ngữ là trường hợp đặc biệt của mô hình quan hệ mờ, khi đó R trở thành quan hệ crisp bị ràng buộc sao cho chỉ có một phần tử khác không được phép tồn tại trong từng hàng của R (mỗi luật chỉ có một hệ quả).

5. Mô hình Takagi–Sugeno

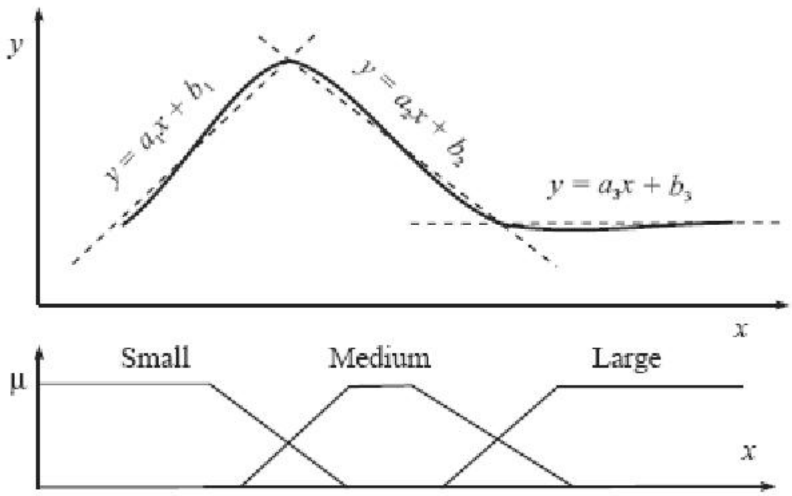
Mô hình mờ Takagi–Sugeno (Takagi and Sugeno, 1985), thì khác, dùng các hàm crisp trong hệ quả. Như thế, có thể xem là tổ hợp của phương pháp mô hình hóa dạng ngôn ngữ và dạng hồi qui toán học (mathematical regression) theo nghĩa là tiền đề mô tả vùng mờ trong không gian vào, trong đó các hàm hệ quả có giá trị. Các luật TS có dạng sau:

$\mathfrak{R}_i$: Nếu x là A_i thì $y_i = f_i(x)$, $\qquad i = 1, 2, \dots, K$. $\qquad$ (3.60)

Khác với mô hình ngôn ngữ, ngõ vào x là biến crisp (về nguyên tắc có thể có biến ngôn ngữ vào, nhưng cần dùng nguyên lý mở rộng (Zadeh, 1975) để tính giá trị mờ của y_i). Hàm f_i thì thường có cùng dạng cấu trúc, chỉ khác các tham số trong mỗi luật. Thông thường, f_i là hàm vectơ, nhưng để dễ ta xem f_i là hàm vô hướng. Một phương pháp dễ dàng và thực tế khi tham số hóa là phép affine (tuyến tính về tham số) form, có luật:

$\mathfrak{R}_i$: Nếu x là A_i thì $y_i = a^T_i x + b_i$, $\qquad i = 1, 2, \dots, K$, $\qquad$ (3.61)

Trong đó a_i là vectơ tham số và b_i là phần offset vô hướng. Mô hình này còn được gọi là mô hình *affine TS*. Chú ý là nếu $a_i = 0$ với mỗi i, ta có được mô hình singleton (3.42).



Hình 3.17 Mô hình mờ Tagagi-Sugeno làm bộ xấp xỉ tuyến tính từng đoạn mịn của hàm phi tuyến.

5.1 Phép suy diễn trong mô hình TS

Công thức suy diễn trong mô hình TS là mở rộng phương pháp suy diễn của mô hình singleton (3.43):

$$y = \frac{\sum_{i=1}^{K} \beta_i y_i}{\sum_{i=1}^{K} \beta_i} = \frac{\sum_{i=1}^{K} \beta_i (a_i^T x + b_i)}{\sum_{i=1}^{K} \beta_i} \quad (3.62)$$

Khi các tập mờ tiền đề được định nghĩa riêng biệt nhưng các vùng trùng lắp (overlapping) trong không gian tiền đề và các tham số a_i và b_i tương ứng với tuyến tính hóa cục bộ của hàm phi tuyến, mô hình TS có thể xem là phép xấp xỉ từng đoạn mịn (smoothed piece-wise approximation) của hàm nay, xem hình 3.17.

5.2 Mô hình TS làm hệ thống giả tuyến tính (Quasi-Linear System)

Mô hình affine TS có thể xem là hệ giả-tuyến tính (tức là hệ tuyến tính có ngõ vào phụ thuộc tham số). Để minh họa, định nghĩa mức hoàn thành chuẩn hóa là:

$$\gamma_i(x) = \frac{\beta_i(x)}{\sum_{j=1}^{K} \beta_j(x)} \quad (3.63)$$

Ta viết β_i(x) một cách tường minh là hàm của x nhằm nhấn mạnh là mô hình TS là dạng mô hình giả tuyến tính có dạng sau:

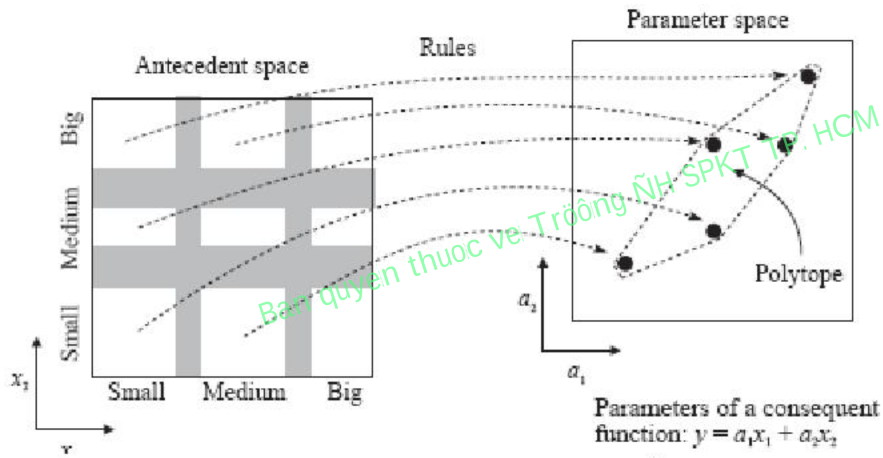
$$y = \left(\sum_{i=1}^{K} \gamma_i(x) a_i^T \right) x + \sum_{i=1}^{K} \gamma_i(x) b_i = a^T(x) x + b(x) \quad (3.64)$$

‘Tham số’ a(x), b(x) là tổ hợp tuyến tính lồi của các tham số hệ quả a_i và b_i, thí dụ:

$$a(x) = \sum_{i=1}^{K} \gamma_i(x) a_i, \quad b(x) = \sum_{i=1}^{K} \gamma_i(x) b_i \tag{3.65}$$

Theo nghĩa này thì mô hình TS có thể xem là ánh xa từ không gian tiền đề (vào) đến vùng lồi (polytope) trong không gian tham số của hệ giả tuyến tính, như mô tả ở hình 3.18.

Tính chất này làm dễ dàng khi phân tích mô hình TS trong khung sườn tương tự như hệ thống tuyến tính. Nhiều phương pháp đã được phát triển nhằm thiết kế bộ điều khiển bộ điều khiển với các đặc tính vòng kín mong muốn (Filev, 1996) và để phân tích tính ổn định (Tanaka và Sugeno, 1992; Zhao, 1995; Tanaka, et al., 1996).



Hình 3.18: Mô hình TS dùng hệ quả affine có thể được xem là ánh xa từ không gian tiền đề sang tham số hệ quả.

6. Hệ mờ động

Trong mô hình hóa và nhận dạng hệ thống, thường gặp bài toán xấp xỉ hệ thống động. Hệ động thay đổi theo thời gian thường được mô hình hóa thành hàm tĩnh dùng ý niệm trạng thái của hệ thống. Cho trạng thái của hệ thống với ngõ vào cho trước, xác định được trạng thái kế tiếp. Trong thiết lập rời rạc theo thời gian, viết:

$$x(k+1) = f(x(k), u(k)), \tag{3.66}$$

trong đó x(k) và u(k) lần lượt là trạng thái và ngõ vào tại thời gian k, và f là hàm tĩnh, còn gọi là *hàm chuyển trạng thái*. Nhiều dạng mô hình mờ có thể dùng làm xấp xỉ hàm chuyển trạng thái. Do không đo được trạng thái của quá trình, nên thường dùng phương pháp mô hình hóa vào-ra. Thông dụng nhất là dạng mô hình NARX (Nonlinear AutoRegressive with eXogenous input):

$$y(k+1) = f(y(k), y(k-1), \ldots, y(k-n_y+1), u(k), u(k-1), \ldots, u(k-n_u+1)). \tag{3.67}$$

Trường hợp này thì $y(k), \ldots, y(k-n_y+1)$ và $u(k), \ldots, u(k-n_u+1)$ lần lượt là các ngõ ra và các ngõ vào quá khứ và n_y, n_u là các số nguyên có liên quan đến bậc của hệ động. Thí dụ, mô hình mờ singleton của hệ thống động có thể gồm các luật theo dạng sau:

$\mathcal{R}_i$: Nếu $y(k)$ là A_{i1} và $y(k-1)$ là A_{i2} và . . .$y(k-n+1)$ là A_{in}
và $u(k)$ là B_{i1} và $u(k-1)$ là B_{i2} và . . .$u(k-m+1)$ là B_{im}
thì $y(k+1)$ là c_i. (3.68)

Theo nghĩa này thì đáp ứng động cần được chăm sóc (taken care) nhờ các bộ lọc động thêm vào hệ mờ, xem hình 3.19. Trong (3.68), bộ lọc động ngõ vào đơn giản là các khâu trễ cho ngõ vào và ngõ ra, và không có bộ lọc tại ngõ ra. Mô hình TS động là một dạng hệ thống hoạch định tham số (parameter-scheduling) và có hệ quả là các mô hình tuyến tính với tham số thường khác nhau trong từng luật:

$\mathcal{R}_i$: Nếu $y(k)$ là A_{i1} và $y(k-1)$ là A_{i2} và . . . $y(k-n_y+1)$ là A_{iny}
và $u(k)$ là B_{i1} và $u(k-1)$ là B_{i2} và . . . $u(k-n_u+1)$ là B_{inu}
thì $$y(k+1)=\sum_{j=1}^{ny} a_{ij}\,y(k-j+1)+\sum_{j=1}^{nu} b_{ij}\,u(k-j+1)+c_i \qquad (3.70)$$

Bên cạnh các dạng hệ vào-ra thường dùng, các mô hình mờ còn có thể biểu diễn hệ phi tuyến trong không gian trạng thái:

$$\begin{aligned} \mathrm{x}(k+1) &= g(\mathrm{x}(k), \mathrm{u}(k)) \\ \mathrm{y}(k) &= h(\mathrm{x}(k)) \end{aligned}$$

trong đó hàm chuyển trạng thái g ánh xạ trạng thái hiện tại $\mathrm{x}(k)$ và ngõ vào $\mathrm{u}(k)$ sang trạng thái mới $\mathrm{x}(k+1)$. Hàm ra h ánh xạ trạng thái $\mathrm{x}(k)$ sang ngõ ra $\mathrm{y}(k)$.
Thí dụ về biểu diễn mô hình không gian trạng thái của mô hình Takagi–Sugeno:

Nếu $\mathrm{x}(k)$ là A_i và $\mathrm{u}(k)$ là B_i thì $$\begin{cases} x(k+1)=A_i x(k)+B_i u(k) \\ y(k)=C_i x(k)+c \end{cases} \qquad (3.73)$$

với $i = 1, \ldots, K$. Trường hợp này thì A_i, B_i, C_i, a_i và c_i là các ma trận và vectơ có chiều thích hợp, liên quan với luật thứ i.

Biểu diễn không gian-trạng thái hữu ích khi kiến thức ban đầu cho phép thiết lập mô hình hệ thống từ nguyên lý thứ nhất như cân bằng khối lượng và năng lượng. Trong các tài liệu thì hướng này được gọi là mô hình hóa trong không gian trạng thái dạng hộp trắng (Ljung, 1987). Nếu trạng thái được đo trực tiếp trên hệ thống, hay có thể cấu trúc lại từ việc đo lường các biến khác, thì có thể xấp xỉ cả hàm g và h dùng kỹ thuật hồi qui phi tuyến (nonlinear regression techniques). Ưu điểm của hướng mô hình hóa dùng không gian trạng thái là cấu trúc của mô hình có liên quan đến cấu trúc của hệ thống thực, nên các tham số mô hình thường có ý nghĩa thích đáng. Điều này, không giống như trường hợp của các mô hình vào-ra. Hơn nữa, chiều của bài toán hồi qui trong mô hình không gian trạng thái thường nhỏ hơn trường hợp của mô hình vào-ra, do trạng thái của hệ thống có thể được biểu diễn dùng vectơ với chiều thấp hơn phương pháp hồi qui (3.67).
Do mô hình mờ có khả năng xấp xỉ hàm mịn (smooth function) đến độ chính xác mong muốn, (Wang, 1992) các mô hình dạng (3.68), (3.70) và (3.73) có thể xấp xỉ bất

kỳ chế độ quan sát được và điều khiển được của nhiều lớp các hệ thống phi tuyến rời rạc theo thời gian (Leonaritis and Billings, 1985)

7. Tóm tắt và các vấn đề cần quan tâm

Chương này trình bày phần tổng quan về bốn dạng mô hình mờ dùng luật nền: mô hình dạng ngôn ngữ (Mamdani-type), mô hình quan hệ mờ (fuzzy relational), mô hình singleton và mô hìnhTakagi–Sugeno. Khác biệt chủ yếu là giữa các mô hình ngôn ngữ, có tập mờ trong cả luật tiền đề và hệ quả, và mô hình, trong đó hệ quả có dạng hàm thực (crisp) theo các biến vào. Mô hình quan hệ mờ có thể được xem là dạng mở rộng của mô hình ngôn ngữ, trong đó cho phép các mức độ tương quan khác nhau giữa các thừa số ngôn ngữ tiền đề và hệ quả.

8. Bài tập

1. Định nghĩa biến ngôn ngữ. Cho biết khác biệt giữa biến ngôn ngữ và thừa số ngôn ngữ?
2. Phép minimum *t*-norm có thể dùng biểu diễn các luật nếu-thì tương tự như phép hàm ý mờ (fuzzy implications). Tuy nhiên, đây lại không phải là hàm hàm ý (implication function). Giải thích tại sao? Cho ít nhất một thí dụ về hàm có tính hàm ý mờ.

3. Xét luật Nếu x là A thì y là B với các tập mờ $A = \{0.1/x_1, 0.4/x_2, 1/x_3\}$ và $B = \{0/y_1, 1/y_2, 0.2/y_3\}$. Tìm quan hệ mờ R biểu diễn bảng sự thật của luật mờ này. Đầu tiên dùng phép minimum *t*-norm rồi dùng tiếp hàm ý Łukasiewicz implication. Thảo luận về sai biệt giữa các kết quả.
4. Giải thích các bước của thuật toán suy diễn Mamdani (max-min) cho một hệ mờ với một ngõ vào (crisp) và một ngõ ramờ. Áp dụng vào luật nền sau:

   1) Nếu x là A_1 thì y là B_1,
   2) Nếu x là A_2 thì y là B_2

Và

$A_1 = \{0.1/1,\ 0.6/2,\ 1/3\}$, $A_2 = \{0.9/1,\ 0.4/2,\ 0/3\}$,
$B_1 = \{1/4,\ 1/5,\ 0.3/6\}$, $B_2 = \{0.1/4,\ 0.9/5,\ 1/6\}$,

Tìm suy diễn. Tính tập mờ ra B' khi $x = 2$.

5. Định nghĩa các phương pháp giải mờ trọng tâm và trung bình-cực đại. Áp dụng vào tập mờ $B = \{0.1/1,\ 0.2/2,\ 0.7/3,\ 1/4\}$ và so sánh với kết quả số:
6. Xét các luật Takagi–Sugenosau:

   1) Nếu x là A_1 và y là B_1 thì $z_1 = x + y + 1$
   2) Nếu x là A_2 và y là B_1 thì $z_2 = 2x + y + 1$
   3) Nếu x là A_1 và y là B_2 thì $z_3 = 2x + 3y$
   4) Nếu x là A_2 và y là B_2 thì $z_4 = 2x + 5$

Cho công thức tính ngõ ra z và tìm z khi $x = 1, y = 4$ và tập mờ tiền đề là

$A_1 = \{0.1/1,\ 0.6/2,\ 1/3\}$, $A_2 = \{0.9/1,\ 0.4/2,\ 0/3\}$,
$B_1 = \{1/4,\ 1/5,\ 0.3/6\}$, $B_2 = \{0.1/4,\ 0.9/5,\ 1/6\}$.

7. Xét hệ động ẩn $y(k+1) = f(y(k),\ u(k))$. Cho thí dụ về mô hình mờ singleton dùng được để xấp xỉ hệ thống này. Cho biết tham số tự do của mô hình?

CHƯƠNG BỐN: XÂU CHUỖI FUZZY (FUZZY CLUSTERING)

Kỹ thuật xâu chuỗi là phương pháp không giám sát (unsupervised methods) được dùng khi tổ chức dữ liệu thành nhóm dùng tính giống nhau của từng mục dữ liệu riêng. Hầu hết các thuật toán xâu chuỗi đều dùng các phương pháp thống kê truyền thống, như phương pháp phân bố dữ liệu thống kê cơ sở, nên rất hữu ích trong trường hợp biết rất it thông tin ban đầu. Khả năng của các thuật toán xâu chuỗi trong nhằm phát hiện cấu trúc cơ bản (underlying structures) trong dữ liệu, và được khái thác trong rất nhiều ứng dụng như xếp lớp, xử lý ảnh, phân loại mẫu, mô hình và nhận dạng.

Chương này trình bày tổng quan về thuật toán xâu chuỗi mờ trên nền *c-means*. Độc giả có thể tham khảo thêm về phép xâu chuỗi mờ trong tài liệu cổ điển của Duda và Hart (1973), Bezdek (1981) và Jain và Dubes (1988). Gần đây có thêm phần tổng quan về các thuật toán xâu chuỗi của (Bezdek and Pal, 1992).

1. Các ý niệm cơ bản

Trình bày các ý niệm cơ bản về dữ liệu, chuỗi cluster, và chuỗi prototypes cùng tổng quan về nhiều hướng xâu chuỗi khác.

1.1 Tập dữ liệu

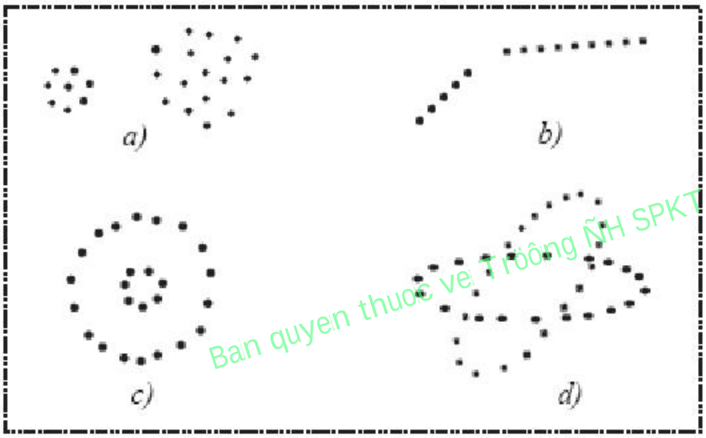
Kỹ thuật xâu chuỗi có thể áp dụng cho dữ liệu định lượng (dạng số), dữ liệu định tính (khẳng định), hay hỗn hợp cả hai. Chương này xem xét việc xâu chuỗi các dữ liệu định lượng. Dữ liệu là quan sát tiêu biểu của các quá trình vật lý nào đó. Mỗi quan sát n biến đo được, nhóm thành vectơ cột n-chiều $\mathbf{z}_k = [z_{1k}, \ldots, z_{nk}]^T$, $\mathbf{z}_k \in \mathrm{R}^n$. Tập của N quan sát được gọi là $\mathbf{Z} = \{\mathbf{z}_k \mid k = 1, 2, \ldots, N\}$, và được biểu diễn thành ma trận $n \times N$:

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1N} \\ z_{21} & z_{22} & \cdots & z_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nN} \end{bmatrix} \qquad (4.1)$$

Trong thuật ngữ về nhận dạng mẫu, các cột của ma trận này được gọi là mẫu (*patterns*) hay đối tượng (objects), các hàng được gọi là đặc trưng (*features*) hay hay thuộc tính (attributes), và **Z** được gọi là mẫu hình (*pattern*) hay ma trận dữ liệu. Ý nghĩa của các hàng và các cột trong **Z** tùy thuộc vào ngữ cảnh. Thí dụ, trong chẩn đoán y khoa, các cột này có thể là bệnh nhân, và các hàng là các hiện tượng, hay các xác nghiệm của các bệnh nhân này. Khi dùng phương pháp xâu chuỗi trong mô hình hóa và nhận dạng hệ thống động, các cột trong **Z** có thể chứa các mẫu tín hiệu thời gian, và các cột là các biến vật lý quan sát được của hệ thống (vị trí, áp suất, nhiệt độ, v.v,..). Để biểu diễn được các đăc tính động của hệ thống, cũng cần có thêm các trị quá khứ của các biến này trong **Z**.

1.2 Clusters và Prototypes

Có nhiều định nghĩa về cluster, tùy theo mục tiêu xâu chuỗi. Thông thường, xem quan điểm rằng cluster là nhóm các đối tượng giống nhau nhiều hơn so với các thành viên của nhóm các clusters khác (Bezdek, 1981; Jain và Dubes, 1988). Thừa số "tương tự" cần được hiểu theo nghĩa tương tự toán học theo nghĩa chính xác. Trong không gian mêtric, tương tự thường được định nghĩa thông qua ý nghĩa norm cự ly (*distance norm)*. Cự ly có thể đo theo tự thân vectơ dữ liệu, hay là cự ly từ vectơ dữ liệu đến một số (prototype) của cluster. Các prototypes thì thường không biết được trước, và được thuật toán xâu chuỗi tìm kiếm cùng lúc với việc tạo các partition dữ liệu. Các prototypes có thể là vectơ cùng chiều với các đối tượng dữ liệu, nhưng cũng có thể được định nghĩa như là đối tượng hình học "cấp cao", như hàm hay không gian con phi tuyến.



Hình 4.1 Cluster của nhiều hình dáng và kích thước khác nhau trong $\mathbb{R}^2$. Theo (Jain và Dubes, 1988)

Dữ liệu có thể phát hiện các cluster với nhiều dạng hình học khác nhau, về kích thước và mật độ như mô tả trong hình 4.1. Do clusters (a) có dạng cầu, các cluster từ (b) đến (d) có thể được đặc trưng là không gian con tuyến tính hay phi tuyến trong không gian dữ liệu. Hiệu năng của hầu hết các thuật toán xâu chuỗi thường không chỉ bị ảnh hưởng từ dạng hình học và mật độ của từng cluster riêng lẽ, mà còn từ quan hệ không gian và cự ly bên trong cluster. Các cluster có thể được phân cách nhau rất tốt, kết nối liên tục, hay trùng lắp với nhau.

1.3 Tổng quan về phương pháp xâu chuỗi

Trong nhiều tài liệu đã giới thiệu về nhiều thuật toán xâu chuỗi. Do có thể xem các cluster là không gian con của tập dữ liệu, nên có một khả năng xếp lớp các phương pháp xâu chuỗi thành tập con mờ (*fuzzy*) hay *crisp* (cứng).

Phương pháp xâu chuỗi cứng (*Hard clustering*) dùng lý thuyết tập hợp cổ điển, có yêu cầu là đối tượng có thể thuộc hay không thuộc về một cluster. Phép xâu chuỗi cứng tức là tạo các partition dữ liệu thành con số đặc thù hay các tập con loại trừ nhau.

Phương pháp xâu chuỗi mờ (*Fuzzy clustering*) thì trái lại, cho phép các đối tượng đồng thời thuộc về nhiều cluster, với các mức thành viên khác nhau. Trong nhiều trường hợp, xâu chuỗi mờ còn tự nhiên hơn phương pháp xâu chuỗi cứng. Các đối tượng trên biên giữa nhiều lớp thì không bắt buộc phải thuộc hoàn toàn trong một lớp, nhưng có thể được định nghĩa mức thành viên nằm giữa 0 và 1, chỉ thị mức tham gia của mình. Bản chất rời rạc của phép tạo partition cứng còn tạo khó khăn cho các thuật toán dùng giải tích hàm, do các hàm này không khả vi.

Các phương pháp xếp lớp khác có thể liên quan đến các hướng thuật toán dùng nhiều kỹ thuật khác nhau (Bezdek, 1981).

- *Các phương pháp phân cấp dùng tính gộp (Agglomerative hierarchical methods)* và các phương pháp phân cấp dùng tính chia (*splitting hierarchical methods*) tao các cluster mới bằng cách định vị lại mức thành viên tại một thời điểm, dùng một số phương pháp đo lường tính tương đồng thích hợp.

- Khi dùng *phương pháp graph* (*graph-theoretic methods*), thì **Z** được xem là tập các nút. Trọng lượng biên giữa các cặp nút được tính từ đo lường tính tương đồng giữa các nút này.

- Thuật toán xâu chuỗi có thể dùng *hàm đối tượng* (*objective function*) để đo mức khát khao của các partitions. Các thuật toán tối ưu hóa phi tuyến được dùng tìm kiếm cực tiểu cục bộ của hàm đối tượng.

Phần còn lại của chương tập trung vào phương pháp xâu chuỗi mờ dùng hàm đối tượng. Các phương pháp này tương đối dễ hiểu, và có minh chứng toán học về đặc tính hội tụ và phương pháp đánh giá cluster.

2. Phân chia cứng và phân chia mờ

Ý niệm về phân chia mờ chủ yếu dùng trong phân tích cluster, nên được dùng trong kỹ thuật nhận dạng dùng phép xâu chuỗi mờ. Phương pháp phân chia mờ và phân chia possibilistic có thể được xem là tổng quát hóa của phương pháp phân chia cứng đã được tạo dùng các tập con cổ điển..

2.1 Phân chia cứng

Mục tiêu của xâu chuỗi là phân chia (tạo partition cho) tập dữ liệu **Z** thành c clusters (nhóm, lớp). Thí dụ giả sử là đã biết c dùng kiến thức đã có. Một tập cổ điển, một partition cứng (*hard partition*) của **Z** có thể được định nghĩa là họ các tập con $\{Ai \mid 1 \le i \le c\} \subset P(\mathbf{Z})$[1] dùng các đặc tính sau (Bezdek, 1981):

$$\bigcup_{i=1}^{C} A_i = Z, \qquad (4.2a)$$

$$A_i \cap A_j = \varnothing, \qquad 1 \le i \ne j \le c, \qquad (4.2b)$$

$$\varnothing \subset Ai \subset \mathbf{Z}, \qquad 1 \le i \le c. \qquad (4.2c)$$

Phương trình (4.2a) có nghĩa là tập hội A_i chứa mọi dữ liệu. Các tập con này cần tháo rời được, như định nghĩa ở (4.2b), và không có tập con nào là trống hay chứa mọi dữ liệu trong **Z** (4.2c). Dùng hàm thành viên (đặc tính), partition có thể được biểu diễn một cách thuận tiện qua ma trận partition **U** = $[\mu_{ik}]_{c\times N}$. Hàng thứ i trong ma trận này chứa các giá trị của hàm thành viên μ_i của tập con thứ i là A_i của **Z**. Theo (4.2), phần tử của **U** phải thỏa mãn các điều kiện sau:

$$\mu_{ik} \in \{0,1\}, \qquad 1 \le i \le c, \qquad 1 \le k \le N, \tag{4.3a}$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \qquad 1 \le k \le N, \tag{4.3b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N_{i,} \qquad 1 \le i \le c. \tag{4.3c}$$

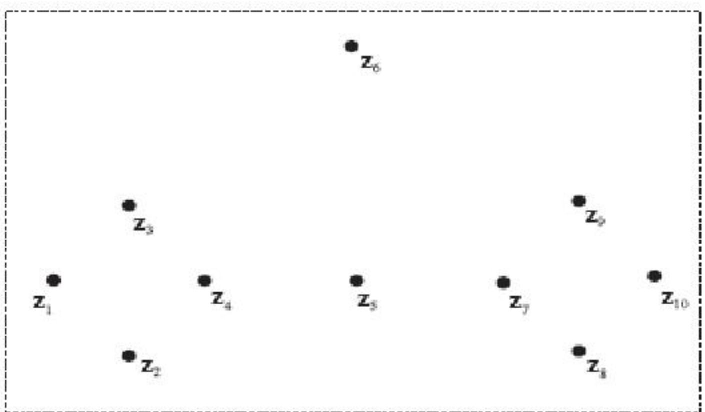
Không gian của mọi ma trận partition cứng có thể có của **Z**, được gọi là không gian partition phân chia cứng (Bezdek, 1981), được định nghĩa là:

$$M_{hc} = \left\{ U \in R^{cXN} \middle| \mu_{ik} \in \{0,1\}, \forall i,k; \sum_{i=1}^{c} \mu_{ik}, \forall k; 0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i \right\}.$$

Example 4.1 Hard partition. Minh họa ý niệm partition cứng bằng một thí dụ đơn giản. Xét tập dữ liệu **Z** = $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{10}\}$, vẽ ở hình 4.2.

Kiểm tra bằng mắt dữ liệu A này, cho đề xuất hai cluster phân biệt nhau (các điểm dữ liệu lần lượt từ $\mathbf{z}_1$ đến $\mathbf{z}_4$ và $\mathbf{z}_7$ đến $\mathbf{z}_{10}$), một điểm giữa hai cluster ($\mathbf{z}_5$), và một điểm nằm ngoài "outlier" $\mathbf{z}_6$. Một partition đặc biệt **U** $\in M_{hc}$ của dữ liệu trong hai tập con (vượt quá 2^{10} khả năng tạo partitions cứng) là:

$$U = \begin{Bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{Bmatrix}$$



Hình 4.2: Tập dữ liệu trong $\mathbb{R}^2$.

Cột thứ nhất của **U** định nghĩa hàm đặc tính theo điểm của tập con thứ nhất A_1 của **Z**, và cột thứ hai định nghĩa hàm đặc tính của tập con A_2 của **Z**. Mỗi mẫu phải được định nghĩa trong một tập con (cluster) của partition. Trường hợp này, cả điểm trên biên $\mathbf{z}_5$

và điểm nằm ngoài $\mathbf{z}_6$ đã được định nghĩa trong A_1. Rõ ràng là phương pháp chia partition cứng không cho được một hình ảnh hiện thực về dữ liệu cơ bản (underlying data). Các điểm dữ liệu trên biên có thể biểu diễn các mẫu (patterns) dùng tính chất hỗn hợp của dữ liệu trong A_1 và A_2, và như thế không thể được hoàn toàn chỉ định là trong lớp này hay lớp khác. Yếu điểm này có thể được giảm nhẹ khi dùng phương pháp partition mờ và partition possibilistic như trình bày trong các phần dưới đây.

2.2 Phân chia mờ (*Fuzzy Partition*)

Tổng quát hóa các partition cứng sang trường hợp mờ được thực hiện bằng cách cho phép μ_{ik} đạt các giá trị thực trong khoảng [0, 1]. Các điều kiện về ma trận partition mờ, tương tự như trong (4.3), được cho bởi (Ruspini, 1970):

$$\mu_{ik} \in [0,1], \qquad 1 \le i \le c, \qquad 1 \le k \le N, \tag{4.4a}$$

$$\sum_{i=1}^{c} \mu_{ik} = 1, \qquad 1 \le k \le N, \tag{4.4b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N, \qquad 1 \le i \le c. \tag{4.4c}$$

Hàng thứ i trong ma trận partition **U** chứa các giá trị của hànm thành viên thứ i của tập mờ con A_i trong **Z**. Phương trình (4.4b) ràng buộc tổng của mỗi cột với 1, như thế thì tổng thành viên của mỗi $\mathbf{z}_k$ trong **Z** thì bằng một. Không gian partition mờ của **Z** là tập

$$M_{fc} = \left\{ U \in R^{cXN} \middle| \mu_{ik} \in [0,1], \forall i,k; \sum_{i=1}^{c} \mu_{ik}, \forall k; 0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i \right\}$$

Thí dụ 4.2 Partition mờ. Xét tập dữ liệu trong thí dụ 4.1. Một trong vô số các partition mờ trong **Z** là:

$$U = \begin{Bmatrix} 1.0 & 1.0 & 1.0 & 0.8 & 0.5 & 0.5 & 0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.5 & 0.5 & 1 & 1.0 & 1.0 & 1.0 \end{Bmatrix}$$

Điểm nằm trên biên $\mathbf{z}_5$ bây giờ có mức thành viên là 0.5 trong tất cả các lớp, phản ảnh đúng đắn vị trí nằm giữa hai clusters. Tuy nhiên, cần chú ý là điểm nằm ngoài $\mathbf{z}_6$ có cùng mức thành viên, cho dù nằm xa hơn so với hai clusters, như thế có thể xem là ít tiêu biểu hơn cho cả A_1 và A_2 so với $\mathbf{z}_5$. Đây là vì điều kiện (4.4b) yêu cầu là tổng các thành viên của mỗi điểm là bằng một. Dĩ nhiên, có thể cho rằng ba clusters thì thích hợp trong thí dụ này hơn so với hai cluster. Tổng quát, rất khó để phát hiện các điểm ngoài và chỉ định cho một cluster ngoại lệ. Việc dùng partition possibilistic, được giới thiệu trong phần sau, giải quyết được yếu điểm của phép partition mờ.

2.3 Phân chia Possibilistic

Một dạng tổng quát hơn của phép partition mờ là partition *possibilistic*, có thể có được thông qua việc bỏ ràng buộc (4.4b). Tuy nhiên, ràng buộc này không bị gở bỏ hoàn toàn nhằm bảo đãm là từng điểm được chỉ định ít nhất trong một tập mờ con có mức thành viên lớn hơn zero. Phương trình (4.4b) có thể được thay thế bằng ràng buộc ít nghiêm ngặt hơn $\forall k, \exists i, \mu_{ik} > 0$. Điều kiện tạo ma trận partition possibilistic là:

$$\mu_{ik} \in [0,1], \qquad 1 \le i \le c, \qquad 1 \le k \le N, \tag{4.5a}$$

$$\exists i,\ \mu_{ik} > 0, \forall k, \tag{4.5b}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N_{i,} \qquad 1 \le i \le c. \tag{4.5c}$$

Tương tự trường hợp trước đây, không gian partition possibilistic **Z** là tập

$$M_{pc} = \left\{ U \in R^{cXN} \middle| \mu_{ik} \in [0,1], \forall i,k; \sum_{i=1}^{c} \mu_{ik}, \forall k; 0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i \right\}$$

Thí dụ 4.3 Partition possibilistic. Một thí dụ về ma trận partition possibilistic của dữ liệu là:

$$U = \begin{Bmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 0.5 & 0.2 & 0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.2 & 1.0 & 1.0 & 1.0 & 1.0 \end{Bmatrix}$$

Do tổng các phần tử trong mỗi cột của $\mathbf{U} \in M_{fc}$ là không còn bị ràng buộc, nên điểm nằm ngoài có thành viên là 0.2 trong tất cả clusters, giá trị này bé hơn thành viên của điểm biên $\mathbf{z}_5$, phản ảnh thực tế là điểm mày ít tiêu biểu hơn cho hai cluster so với $\mathbf{z}_5$.

3. Chức năng Fuzzy *c*-Means

Hầu hết các thuật toán xâu chuỗi mờ (cũng như các thuật toán được trình bày trong chương này) đều dựa trên phép tối ưu hóa hàm mục tiêu *c*-means cơ bản, hay có một số hiệu chỉnh trên đó. Như thế, ta bắt đầu thảo luận về chức năng *c*-means

3.1 Chức năng Fuzzy *c*-Means

Một số lớn họ các thuật toán xâu chuoỗi mờ đều dùng phép tối thiểu hóa chức năng *fuzzy c-means* được đề nghị từ (Dunn, 1974; Bezdek, 1981):

$$J(Z;U,V) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m \|z_k - v_i\|_A^2 \tag{4.6a}$$

Trong đó

$$\mathbf{U} = [\mu ik] \in M_{fc} \tag{4.6b}$$

Là ma trận partition mờ của **Z**,

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c], \qquad \mathbf{v}i \in R^n \tag{4.6c}$$

là vectơ *cluster prototypes* (trung tâm), được định nghĩa theo,

$$D^2_{ikA} = \|z_k - v_i\|^2_A = (z_k - v_i)^T A(z_k - v_i) \tag{4.6d}$$

Là norm cự ly của tích trong bình phương (squared inner-product distance norm), và

$$m \in [1,\infty) \tag{4.6e}$$

là tham số định nghĩa độ mờ (fuzziness) của các clusters kết quả. Giá trị của hàm chi phí (4.6a) có thể được xem là đo lường của phương sai tổng của $\mathbf{z}_k$ từ $\mathbf{v}_i$.

3.2 Thuật toán Fuzzy *c*-Means

Tối thiểu hóa chức năng *c*-means trong (4.6a) biểu diễn bài toán tối ưu hóa phi tuyến có thể được giải dùng nhiều phương pháp khác nhau, bao gồm từ phương pháp tối thiểu hóa dùng bước lặp (iterative minimization), tôi mô phỏng (simulated annealing) hay thuật toán di truyền. Phương pháp thường dùng nhất là phép lặp đơn giản Picard dùng điều kiện bậc nhất của điểm dừng của (4.6a), được gọi là thuật toán FCM (fuzzy *c*-means).

Các điểm dừng của hàm mục tiêu (4.6a) có thể tìm được bằng các ghép ràng buộc (4.4b) vào J bằng nhân tử Lagrange:

$$\bar{J}(Z;U,V,\lambda) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m D^2_{ikA} + \sum_{k=1}^{N}\lambda_k\left[\sum_{k=1}^{c}\mu_{ik} - 1\right], \tag{4.7}$$

Cho gradient của $\bar{J}$ theo **U**, **V** và $\boldsymbol{\lambda}$ về zero. Có thể thấy là nếu cho $D^2_{ikA} > 0, \forall i,k$ và m>0, thì $(\mathbf{U},\mathbf{V}) \in M_{fc} \times R_n \times c$ chỉ tối thiểu hóa (4.6a) được nếu:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}\left(D_{ikA}/D_{jkA}\right)^{2/(m-1)}}, \qquad 1 \le i \le c, \qquad 1 \le k \le N, \tag{4.8a}$$

và

$$v_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^m z_k}{\sum_{k=1}^{N}(\mu_{ik})^m}. \tag{4.8b}$$

Nghiệm này cũng thỏa mãn các ràng buộc còn lại (4.4a) và (4.4c). Phương trình (4.8) là điều kiện cần bậc nhất để điểm dừng của hàm (4.6a). Thuật toán FCM (Algorithm

4.1) tính lặp từ (4.8a) và (4.8b). Điều kiện đủ của (4.8) và hội tụ của thuật toán FCM đã được chứng minh (Bezdek, 1980). Chú ý là (4.8b) cho $\mathbf{v}_i$ là trung bình trọng số của mục dữ liệu phụ thuộc vào cluster, trong đó trọng lượng là mức thành viên. Điều này giải thích tại sao thuật toán được gọi là "*c*-means".
Cần chú ý một số điểm sau:

1. Mục tiêu của nhánh "if...otherwise" trong bước 3 là nhằm giải quyết tính singularity xuất hiện trong FCM khi $D_{is\mathbf{A}} = 0$ với một số $\mathbf{z}_k$ và một hay nhiều cluster prototypes $\mathbf{v}_s$, $s \in S \subset \{1, 2, \ldots, c\}$. Trường hợp này thì không thể tính được mức thành viên trong (4.8a). Khi xuất hiện điều này thì chỉ định 0 cho mỗi μ_{ik}, $i \in \bar{S}$ và thành viên được phân phối bất kỳ trong μ_{sj} chịu ràng buộc $\sum_{s \in S} \mu_{sj} = 1$, $\forall k$.

2. Thuật toán FCM hội tụ đến cực tiểu cục bộ của chức năng *c*-means (4.6a). Như thế, khởi tạo khác nhau có thể dẫn đến các kết quả khác nhau.

3. Bước 1 và 2 thực hiện dễ, nhưng bước 3 thì khó hơn, do xuất hiện singularity trong FCM khi $D_{ik\mathbf{A}} = 0$ với một số $\mathbf{z}_k$ và một hay nhiều $\mathbf{v}_i$. Khi xuất hiện điều này (ít khi xảy ra), thì cho các cluster có mức thành viên là zero.
Với $D_{ik\mathbf{A}} > 0$ và thành viên được phân bố bất kỳ dọc theo clusters có $D_{ik\mathbf{A}} = 0$, sao cho thỏa mãn ràng buộc trong (4.4b).

4. Một dạng sơ đồ tối ưu khác dùng vòng FCM với ước lượng $\mathbf{U}^{(l-1)} \rightarrow \mathbf{V}^{(l)} \rightarrow \mathbf{U}^{(l)}$ rồi chấm dứt ngay khi $\left\| U^{(l)} - U^{(l-1)} \right\| < \varepsilon$. Nói cách khác thì thuật toán có thể được khởi tạo dùng $\mathbf{V}^{(0)}$, lặp vòng qua $\mathbf{V}^{(l-1)} \rightarrow \mathbf{U}^{(l)} \rightarrow \mathbf{V}^{(l)}$, và chấm dứt khi $\left\| U^{(l)} - U^{(l-1)} \right\| < \varepsilon$. Norm của sai số trong tiêu chuẩn chấm dứt thường được chọn là $\max_{ik}(|\mu^{(l)}_{ik} - \mu^{(l-1)}_{ik}|)$. Có thể có nhiều kết quả với cùng giá trị của của ε, do tiêu chuẩn dừng dùng trong thuật toán 4.1 yêu cầu càng nhiều tham số lân cận nhau.

Algorithm 4.1 Fuzzy *c*-means (FCM).

Cho tập dữ liệu $\mathbf{Z}$, chọn số clusters $1 < c < N$, số mủ trọng lượng $m > 1$, dung sai chấp nhận là $\epsilon > 0$ và norm-inducing matrix $\mathbf{A}$.
Khởi tạo ma trận partition một cách ngẫu nhiên, như $\mathbf{U}^{(0)} \in M_{fc}$.

Repeat for $l = 1, 2, \ldots$

Step 1: Tính cluster prototypes (trung bình):

$$v_i^{(l)} = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m z_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \qquad 1 \le i \le c.$$

Step 2: Tính khoảng cách (cự ly):

$$D_{ikA}^2 = (z_k - v_i^{(l)})^T A(z_k - v_i^{(l)}), \quad 1 \le i \le c, \qquad 1 \le k \le N.$$

Step 3: Cập nhật ma trận partition:
for $1 \le k \le N$
if $D_{ik\mathbf{A}} > 0$ for all $i = 1, 2, \ldots, c$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left(D_{ikA} / D_{jkA}\right)^{2/(m-1)}},$$

Otherwise

$\mu_{ik}^{(l)} = 0$ if $D_{ik\mathbf{A}} > 0$, and $\mu_{ik}^{(l)} \in [0.1]$ with $\sum_{i=1}^{c} \mu_{ik}^{(l)} = 1$.

until $\left\| U^{(l)} - U^{(l-1)} \right\| < \varepsilon$.

3.3 Tham số của thuật toán FCM

Trước khi dùng thuật toán FCM, cần đặc trưng các tham số sau: số lượng clusters, c, thừa số mũ 'fuzziness', m, dung sai chấm dứt, ε, là norm-inducing matrix, **A**. Hơn nữa, còn phải khởi tạo ma trận partition **U**. Việc lựa chọn các tham số này được mô tả như sau:

Số lượng các clusters. Số lượng c các clusters là tham số quan trọng nhất, theo nghĩa là các tham số còn lại ít gây ảnh hưởng lên partition tìm được. Khi xâu chuỗi dữliệu thực không có một chút thông tin ban đầu về cấu trúc dữ liệu, thường dùng giả định về số các cluster cơ bản. Việc chọn lựa các thuật toán xâu chuỗi tiếp tục với việc tìm kiếm cho c clusters, bất chấp là chúng có thực sự hiện diện trong dữ liệu hay không. Có hai hướng quan trọng dùng định nghĩa số lương thích hợp các cluster cần được phân biệt:

1. *Đo lường đánh giá (Validity measures).* Chỉ số vô hướng dùng chỉ thị partition tìm được có tốt không. Thuật toán xâu chuỗi thường quan tậm đến vị trí của các cluster compac hay phân biệt rõ. Khi số cluster được chọn là bằng với nhóm đang hiện hữu trong dữ liệu, có thể hy cọng là thuật toán xâu chuỗi sẽ nhận dạng đúng ra chúng. Nếu không, việc nhận dạng sai xuất hiện. Như thế, hầu hết các đo lường đánh giá được thiết kế để định lượng yếu tố phân biệt cùng tính compac của các cluster. Tuy nhiên, theo Bezdek (1981) thì ý niệm về đo lường đánh giá các cluster hiện còn mở và có thể được tạo lập theo nhiều phương cách khác nhau.
Như thế, có nhiều phương pháp đo lường đánh giá đã được trình bày, xem

(Bezdek, 1981; Gath và Geva, 1989; Pal và Bezdek, 1995), trong đó, có trình bày chỉ số Xie-Beni dùng cho thuật toán FCM (Xie and Beni, 1991)

$$\chi(Z;U,V) = \frac{\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m}\left\|z_k - v_i\right\|^2}{c.\min_{i\neq j}\left(\left\|v_i - v_j\right\|^2\right)} \qquad (4.9)$$

đã được tìm ra và chứng tõ là hoạt động tốt trong thực tế. Chỉ số này có thể xem là tỉ số của tổng phương sai trong nhóm và tính phân biệt của các cluster trung tâm. Partition tốt nhất tốithiểu hóa được giá trị của $\chi(\mathbf{Z};\mathbf{U},\mathbf{V})$.

2. *Iterative merging or insertion of clusters.* Ý tưởng cơ bản của việc sáp nhập cluster (cluster merging) là bắt đầu với số lượng lớn các cluster, rồi giảm liên tiếp số lượng này bằng cách sát nhập các cluster tương tự (tương thích) theo một số tiêu chuẩn được định nghĩa rõ ràng (Krishnapuram and Freg, 1992; Kaymak và Babuska, 1995). Ngoài ra còn có thể chấp nhận một xu hướng ngược lại, tức là bắt đầu với một số lượng ít các cluster rồi dùng bước lặp chèn thêm cluster vào vùng mà các điểm dữ liệu có mức thành viên thấp trong các cluster hiện hữu (Gath and Geva, 1989).

Tham số mờ hóa (Fuzziness Parameter). Trọng số mũ *m* cũng là tham số quan trọng , do có ảnh hưởng lớn lên độ mờ của partition kết quả. Khi *m* tiến đến một, thì partition trở thành cứng ($\mu_{ik} \in \{0, 1\}$) và $\mathbf{v}_i$ thành các trung bình thông thường của cluster. Khi $m \to \infty$, thì partition trở thành hoàn toàn mờ ($\mu_{ik} = 1/c$) và các trung bình của cluster thì bằng trung bình của $\mathbf{Z}$. Các đặc tính giới hạn của (4.6) thì độc lập với phương pháp tối ưu được dùng (Pal and Bezdek, 1995). Thông thường, bước đầu thường chọn $m = 2$.

Tiêu chuẩn dừng (Termination Criterion). Thuật toán FCM dừng tính lặp khi norm của sai biệt giữa **U** trong hai bước lặp kế tiếp nhỏ hơn tham số dừng ε. Khi có norm tối đa ($|\mu^{(l)}{}_{ik} - \mu^{(l-1)\backslash}{}_{ik}|$), thường chọn $\varepsilon = 0.001$, ngay khi dùng $\varepsilon = 0.01$ có hoạt động tốt trong một số trường hợp, do giảm thiểu được thời gian tính của máy.

Norm-Inducing Matrix. Hình dáng của các clusters được xác định bằng việc lựa chọn ma trận **A** trong đo lường cự ly (4.6d). Thường chọn $\mathbf{A} = \mathbf{I}$, cho norm Euclide chuẩn:

$$D^2{}_{ik} = (\mathbf{z}_k - \mathbf{v}_i)^T(\mathbf{z}_k - \mathbf{v}_i). \qquad (4.10)$$

Một chọn lựa nữa là của **A** là ma trận đường chéo (diagonal matrix) được tính theo nhiều phương sai trong các chiều của hệ trục theo **Z**:

$$A = \begin{bmatrix} (1/\sigma_1)^2 & 0 & \cdots & 0 \\ 0 & (1/\sigma_2)^2 & \cdots & 0 \\ \vdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & (1/\sigma n)^2 \end{bmatrix}. \qquad (4.11)$$

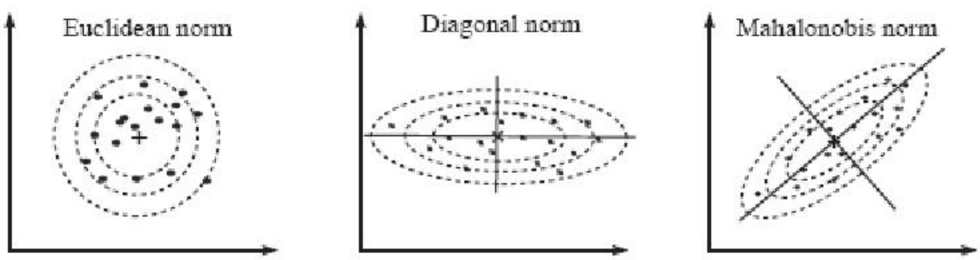
Ma trận này dẫn đến chuẩn đường chéo (diagonal norm) trong R^n. Cuối cùng, **A** còn có thể được định nghĩa là phần nghịch của ma trận đồng phương sai của **Z**: **A** = $\mathbf{R}^{-1}$, trong đó:

$$R = \frac{1}{N}\sum_{k=1}^{N}(z_k - \bar{z})(z_k - \bar{z})^T . \qquad (4.12)$$

Với $\bar{z}$ là trung bình của dữ liệu. Trong trường hợp này, **A** dẫn đến chuẩn Mahalanobis trên R^n.

Norm ảnh hưởng lên các tiêu chuẩn xâu chuỗi bằng cách thay đổi đo lường mức không tương đồng. Norm Euclide dẫn đến các cluster hyperspherical (các mặt và các thành viên hằng số là hyperspheres). Cả hai norm về diagonal và Mahalanobis đều tạo ra các cluster hyperellipsoidal. Khi dùng norm diagonal, thì trục của siêu ellip là song song với hệ trục, còn norm Mahalanobis thì hướng của siêu ellip là bất kỳ, như vẽ trong hình 4.3.

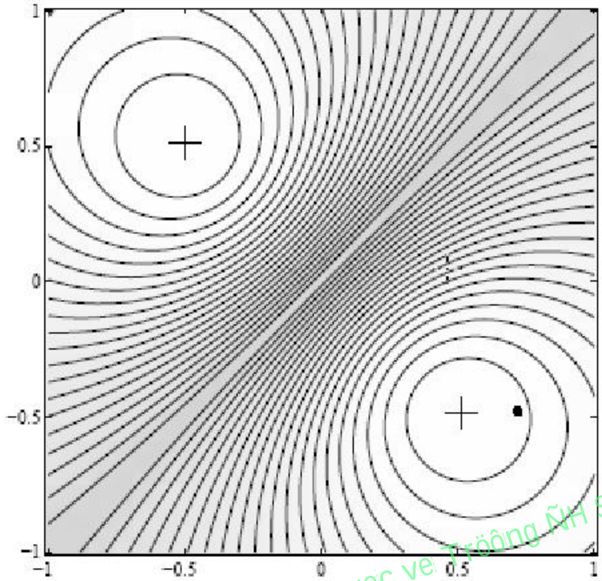
Hạn chế thường gặp của thuật toán xâu chuỗi dùng cự ly cố định là norm cưỡng bức hàm mục tiêu đến cluster prefer của hình dạng nào đó ngay cả khi chúng không hiện diện trong dữ liệu. Thí dụ sau đây minh họa đều trên.



Hình 4.3: Các norm cự ly khác nhau dùng trong xâu chuỗi mờ

Thí dụ 4.4 Xâu chuỗi dùng Fuzzy c-means. Xét tập dữ liệu tổng hợp trong R^2, bao gồm hai cluster phân biệt rõ của nhiều dạng khác nhau, như mô tả trong hình 4.4. Các mẫu của cả các cluster được vẽ từ phân bố chuẩn. Độ lệch chuẩn cho cluster phía trên là 0.2 cho cả hệ trục, trong khi cluster dưới là 0.2 cho trục ngang và 0.05 cho trục dọc. Thuật toán FCM được dùng cho tập dữ liệu này. Ma trận norm-inducing được thiết lập với **A** = **I** cho cả hai clusters, thừa số trọng lượng dạng mủ là m = 2, và tiêu chuẩn dừng $\varepsilon = 0.01$. Thuật toán được khởi tạo dùng ma trận partition ngẫu nhiên và hội tụ được sau 4 bước lặp. Từ đường cong mô tả mức thành viên trong hình 4.4, ta thấy là thuật toán FCM tạo hình dáng tròn cho hai cluster, ngay cả khi hình dạng của cluster thứ hai bị kép dãn ra.

Chú ý là hiện không có trợ giúp nào khi chọn giá trị **A** khác, do hai cluster có các hình dáng khác nhau. Thông thường thì cần nhiều ma trận $\mathbf{A}_i$ nhưng lại không có được hướng dẫn bước đầu để chọn chúng. Trong phần 4.4, ta sẽ thấy là các ma trận này có thể được cập nhật dùng ước lượng đồng phương sai (data covariance) của dữ liệu. Thí dụ 4.5 trình bày partition có được dùng thuật toán dùng norm cự ly thích nghi, thuật toán Gustafson–Kessel.



Hình 4.4 Thuật toán FCM tạo ra hình cầu trong các cluster, bất chấp phân bố dữ liệu hiện tại. Các chấm biểu diễn điểm dữ liệu, '+' là trung bình của cluster. Đồng thời cũng trình bày đường đẳng mức của các cluster. Vùng sậm màu tương ứng với mức thành viên vào khoảng 0.5

Ma trận partition ban đầu. Ma trận partition thường được khởi tạo ngẫu nhiên, sao cho $\mathbf{U} \in M_{fc}$. Một hướng đơn giản để có điều này là khởi tạo các trung tâm cluster $\mathbf{v}_i$ ngẫu nhiên và tính toán giá trị $\mathbf{U}$ tương ứng dùng (4.8a) (tức là dùng bước thứ b của thuật toán FCM)

3.4 Mở rộng của thuật toán FCM

Có nhiều mở rộng nổi tiếng về thuật toán từ FCM:

- Các thuật toán dùng các đo lường cự ly thích nghi, như thuật toán Gustafson–Kessel (Gustafson and Kessel, 1979) và thuật toán ước lượng (fuzzy maximum likelihood) (Gath and Geva, 1989).
- Các thuật toán dùng siêu phẳng (hyperplanar) hay prototypes chức năng, hay các prototypes được hàm định nghĩa. Đó là fuzzy c-varieties (Bezdek, 1981), fuzzy c-elliptotypes (Bezdek, et al., 1981), và các mô hình hồi qui mờ (Hathaway and Bezdek, 1993).
- Các thuật toán tìm kiếm các partition possibilistic trong dữ liệu, tức là các partition trong đó các ràng buộc (4.4b) được giải tỏa.

Phần tiếp theo, ta chú trọng đến thuật toán Gustafson–Kessel.

4. Thuật toán Gustafson–Kessel

Gustafson và Kessel (Gustafson and Kessel, 1979) đã mở rộng thuật toán FCM chuẩn thành thuật toán dùng norm cự ly thích nghi, nhằm phát hiện các cluster có các dạng hình học khác nhau trong tập dữ liệu. Mỗi cluster có ma trận norm-inducing matrix $\mathbf{A}_i$, có được từ các norm dùng tích trong:

$$D^2_{ik\mathbf{A}i} = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A}^i (\mathbf{z}^k - \mathbf{v}^i). \tag{4.13}$$

Ma trận $\mathbf{A}_i$ thường dùng làm biến tối ưu trong c-means functional, nên cho phép mỗi cluster cập nhật norm cự ly từ cấu trúc tôpô cục bộ của dữ liệu.. hàm mục tiêu của thuật toán GK được định nghĩa là:

$$J(Z;U,V,\{A_i\}) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m D^2_{ikAi} \tag{4.14}$$

Hàm mục tiêu không thể tốithiểu hóa một cách trực tiếp theo $\mathbf{A}_i$, do là tuyến tính theo $\mathbf{A}_i$. Để giải quyết, cần giới hạn $\mathbf{A}_i$ theo một số cách. Phương pháp thường dùng là ràng buộc định thức của $\mathbf{A}_i$:

$$|\mathbf{A}_i| = \rho_i, \rho_i > 0, \forall i. \tag{4.15}$$

Điều này cho phép ma trận $\mathbf{A}_i$ thay đổi khi định thức không đổi tương ứng với hình dạng của cluster cần tối ưu hóa trong khi khối lượng được giữa không đổi. Dùng phương pháp nhân tử Lagrange, có được các biểu thức sau cho $\mathbf{A}_i$ (Gustafson and Kessel, 1979):

$$\mathbf{A}_i = [\rho_i \det(\mathbf{F}_i)]^{1/n}\mathbf{F}^{-1}_i, \tag{4.16}$$

Trong đó $\mathbf{F}_i$ là ma trận đồng phương sai của cluster thứ i được cho bởi:

$$F_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^m (z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^{N}(\mu_{ik})^m}. \tag{4.17}$$

Chú ý là việc thay thế các phương trình (4.16) và (4.17) vào (4.13) cho norm cự ly bình phương tổn quát Mahalanobis, trong đó đồng phương sai được lượng hóa dùng mức thành viên của **U**. Thuật toán GK được minh họa trong Algorithm 4.2 và trong thiết lập MATLAB tìm trong phần phụ lục. Thuật toán GK đươc tính toán phức tạp hơn so với trường hợp FCM, do phần đảo và định thức của ma trận cluster đồng phương sai phải được tính trong từng bước lặp.

Algorithm 4.2 Gustafson–Kessel (GK) algorithm.

Cho tập dữ liệu **Z**, chọn số các cluster $1 < c < N$, thừa số mũ trọng lượng $m > 1$ và dung sai dừng $\varepsilon > 0$ và khối lượng cluster là ρ_i. Khởi tạo ma trận partition một cách ngẫu nhiên, sao cho $\mathbf{U}^{(0)} \in M_{fc}$.

Repeat for $l = 1, 2, \ldots$

Step 1: Compute cluster prototypes (means):

$$v_i^{(l)} = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m z_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \qquad 1 \le i \le c .$$

Step 2: Tính các ma trận cluster đồng phương sai:

$$F_i = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m \left(z_k - v_i^{(l)}\right)\left(z_k - v_i^{(l)}\right)^T}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \qquad 1 \le i \le c .$$

Step 3: Tính toán cự ly:

$$D_{ikAi}^2 = \left(z_k - v_i^{(l)}\right)^T \left[\rho_i \det(F_i)^{1/n} F_i^{-1}\right]\left(z_k - v_i^{(l)}\right), \qquad 1 \le i \le c, \qquad 1 \le k \le N .$$

Step 4: Cập nhật ma trận partition:

for $1 \le k \le N$

if $D_{ik\mathbf{A}i} > 0$ for all $i = 1, 2, \ldots, c$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left(D_{ikA} / D_{jkA}\right)^{2/(m-1)}},$$

otherwise

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ik\mathbf{A}} > 0, \text{ and } \mu_{ik}^{(l)} \in [0.1] \text{ with } \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1.$$

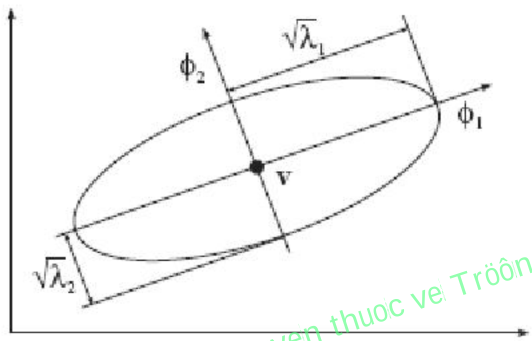
until $\left\| U^{(l)} - U^{(l-1)} \right\| < \varepsilon$.

4.1 Tham số của Thuật toán Gustafson–Kessel

Các tham số phải được đặc trưng tương tự như trong thuật toán FCM algorithm (trừ ma trận norm inducing **A**, được cập nhật tự động): số lượng các cluster c, thừa số mủ 'fuzziness' m, dung sai dừng ε. Các tham số còn lại là khối lượng cluster ρ_i. Khong có kiến thứa ban đầu nào, ρ_i chỉ đơn giản là 1 cho từng cluster. Nhược điểm của thiết lập này là do ràng buộc (4.15), nên thuật toán GK chỉ có thể tìm được các cluster có xấp xỉ cùng khối lượng.

4.2 Diễn đạt của ma trận cluster đồng phương sai

Cấu trúc riêng của ma trận các cluster đồng phương sai $\mathbf{F}_i$ cung cấp thông tin về hình dáng và hướng của cluster. Tỉ lệ giữa chiều dài của trục siêu ellip của cluster được cho từ tỉ số căn bình phương của các trị riêng của $\mathbf{F}_i$. Chiều của các trục được cho bởi các vectơ riêng $\mathbf{F}_i$, như vẽ trong hình 4.5.
Thuật toán GK có thể được dùng để phát hiện các cluster dọc theo không gian con tuyến tính của không gian dữ liệu. Các clusters được biểu diễn dùng nmột siêu ellip phẳng, có thể được xem là hyperplanes. Các vectơ riêng tương ứng với các trị riêng bé nhất xác định tính trực giao với hyperplane, và có thể đươc dùng tín htoán mô hình tuyến tính cục bộ tối ưu từ ma trận đồng phương sai.

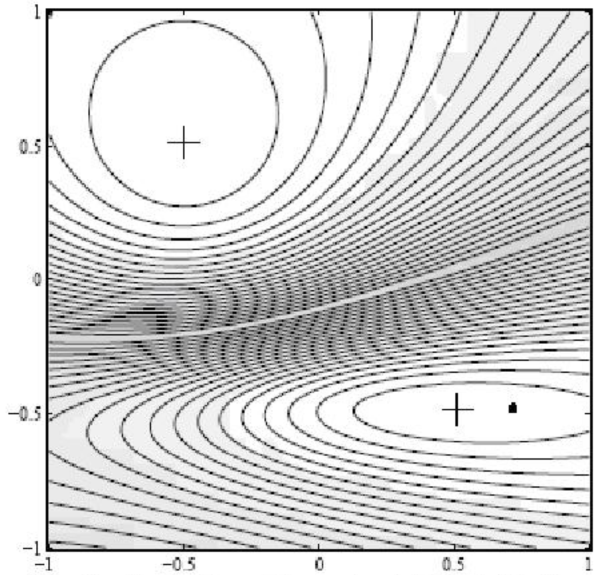


Hình 4.5: Phương trình $(\mathbf{z}-\mathbf{v})^T\mathbf{F}^{-1}(\mathbf{x}-\mathbf{v}) = 1$ định nghĩa siêu ellip. Chiều dài thứ j của siêu ellip cho bởi $\sqrt{\lambda_j}$ và các chiều được ϕ_j mở rộng, trong đó λ_j và ϕ_j lần lượt là trị riêng và vectơ riêng của $\mathbf{F}$.

Thí dụ 4.5 Thuật toán Gustafson–Kessel. Thuật toán GK được ứng dụng cho tập dữ liệu lấy từ thí dụ 4.4, dùng cùng các thiếp lập như thuật toán FCM. Hình 4.4 cho thấy thuật toán GK có thể cập nhật norm cự ly thành phân bố cơ bản (underlying distribution) của dữ liệu. Có được một cluster dạng tròn và một có dạng ellip kéo dài. Hình dáng của của các cluster có thể đượv xác định từ cấu trúc riêng (eigenstructure) của ma trận tạo đồng phương sai $\mathbf{F}i$. Các trị riêng của các cluster được cho ở bảng Table 4.1.
Ta thấy rằng tỉ số cho trong cột cuối phản ảnh gần chính xác tỉ số của độ lệch chuẩn (standard deviations) trong từng nhóm dữ liệu (lần lượt từ 1 đến 4). Đối với các cluster thấp hơn, vectơ đơn vị riêng tương ứng với $\lambda 2$, $\boldsymbol{\varphi}2 = [0.0134,\ 0.9999]T$, có thể xem là trực giao với đường thẳng biểu diễn chiều của cluster thứ hai, và như thế thì gần như là song song với trục dọc (vertical axis).

Bảng 4.1: Các trị riêng của ma trận cluster đồng phương sai của các cluster trong thí dụ 4.5

cluster	λ_1	λ_2	$\sqrt{\lambda_1}/\sqrt{\lambda_2}$
upper	0.0352	0.0310	1.0666
lower	0.0482	0.0028	4.1490



Hình 4.6: Thuật toán Gustafson-Kessel có thể phát hiện cluster có hình dáng và hướng khác nhau. Các điểm '+'biểu diễn dữ liệu. Hình còn ghi các mức của cluster.Phần bóng sậm màu tương ứng với mức thành viên vào khoảng 0.5

5. Tóm tắt và các vấn đề cần quan tâm

Phương pháp xâu chuỗi mờ là phương pháp không giám sát rất mạnh dùng phân tích dữ liệu và kiến tạo các mô hình. Chương này trình bày tổng quát về các thuật toán xâu chuỗi mờ thường dùng nhất. Chương cho thấy là sơ đồ tính lặp dạng *c*-means có thể được dùng kết hợp với phương pháp đo lường cự ly thích nghi để phát hiện các clusters với nhiều hình dáng khác nhau. Đồng thời cũng trình bày việc lựa chọn các tham số quan trọng do người dùng định nghĩa, như số clusters và các tham số mờ hóa.

6. Bài tập

1. Tìm định nghĩa và thảo luận về khác biệt giữa các partition mờ và không mờ (cứng). Cho thí dụ về ma trận partition mờ và không mờ. Cho biết ưu điểm của phương pháp xâu chuỗi mờ so với phương pháp xâu chuỗi cứng?

2. Định nghĩa toán học của ít nhất hai norms cự ly khác nhau trong xâu chuỗi mờ. Giải thích về sự khác biệt này.

3. Trình bày hai thuật toán xâu chuỗi mờ và giải thích sự khác biệt giữa chúng với nhau.

4. Định nghĩa chức năng *c*-mean mờ và giải thích mọi ký hiệu.

5. Liệt kê các bước cần có để khởi tạo và thực hiện thuật toán fuzzy *c*-means. Cho biết vai trò và ảnh hưởng của các tham số do người dùng định nghĩa trong thuật toán?

CHƯƠNG NĂM: CÁC KỸ THUẬT KIẾN TẠO HỆ MỜ

Hai nguồn thông tin quan trọng dùng xây dựng hệ mờ là *kiến thức ban đầu* (*prior knowledge)* và *dữ liệu* (*data*) đo lường được. Kiến thức ban đầu có được từ bản chất xấp xỉ (kiến thức định tính, heuristics), thường có được từ các "chuyên gia"., nhà thiết kế quá trình, người vận hành, v.v,... Theo nghĩa này thì mô hình mờ có thể được xem như là hệ chuyên gia mờ đơn giản (Zimmermann, 1987).

Trong một số quá trình, dữ liệu có được ghi nhận từ hoạt động của quá trình hay từ các kinh nghiệm nhận dạng đặc biệt được thiết kế để có dữ liệu này. Xây dựng mô hình mờ từ dữ liệu cần các phương pháp dùng logic mờ và phép suy luận xấp xỉ (approximate reasoning), nhưng cũng cần các ý tưởng từ lĩnh vực mạng nơrôn, phân tích dữ liệu và các phương pháp nhận dạng hệ thống truyền thống. Thu thập dữ liệu hay tinh chỉnh hệ mờ dùng dữ liệu được gọi là *nhận dạng hệ mờ* (*fuzzy systems identification*).

Hai hướng quan trọng để tích hợp kiến thức và dữ liệu trong mô hình mờ là:

1. Kiến thức của chuyên gia biểu diễn dạng ngôn ngữ được chuyển thành tập các luật nếu-thì. Từ đó, tạo ra được một số cấu trúc mô hình. Các tham số của cấu trúc này (hàm thành viên, hệ quả dạng singletons hay các tham số của hệ quả TS) có thể được tinh chỉnh dùng các dữ liệu vào-ra. Các thuật toán tinh chỉnh đặc biệt khai thác thực tế là tại cấp độ tính toán, mô hình mờ có thể được xem là cấu trúc lớp (trong mạng) tương tự như mạng nơrôn nhân tạo, nên có thể dùng các thuật toán học dạng chuẩn. Đây là hướng mô hình mờ-nơrôn (*neuro-fuzzy modeling*).

2. Khi không có kiến thức ban đầu về hệ thống đang khảo sát để tạo luật, thì thường dùng dữ liệu để tạo mô hình mờ. Điều mong muốn là các luật tìm được cùng các hàm thành viên có thể cung cấp một diễn đạt tiếp theo về hành vi của hệ thống (posteriori interpretation). Một chuyên gia khi đương đầu với thông tin này có thể thay đổi luật, hay cung cấp một luật mới, và có thể thiết kế thêm các kinh nghiệm nhằm tạo nhiều dữ liệu thông tin mới. Xu hướng này được gọi là rút ra luật (*rule extraction*). Phương pháp xâu chuỗi mờ là một trong những kỹ thuật thường được áp dụng (Yoshinari, et al., 1993; Nakamori and Ryoke, 1994; Babuska and Verbruggen, 1997).

Dĩ nhiên là có thể kết hợp các kỹ thuật này lại, tùy theo ứng dụng đặc thù. Phần này mô tả các bước và các lựa chọn cơ bản để tạo các mô hình mờ trên nền tri thức, và các kỹ thuật chủ yếu để rút tìm luật hay tinh chỉnh các mô hình hệ mờ từ dữ liệu.

1. Cấu trúc và tham số

Khi thiết kế mô hình mờ (và các dạng khác), cần phân biệt hai vấn đề sau: cấu trúc và tham số của mô hình. Cấu trúc xác định tính mềm dẽo của mô hình khi xấp xỉ các ánh xạ ẩn. Tham số sau đó được chỉnh định (ước lượng) để khớp với dữ liệu có được. Một mô hình với cấu trúc phong phú thì có thể xấp xỉ nhiều hàm phức tạp, nhưng như thế thì sẽ kém về tính tổng quát hóa (*generalization*). Tổng quát hóa tốt tức là mô hình khớp được với một tập dữ liệu thì cũng khớp tốt với tập dữ liệu khác trong cùng quá trình. Trong mô hình mờ, việc chọn lựa cấu trúc bao gồm các chọn lựa sau:

- *Các biến vào và ra.* Trong các hệ thống phức tạp thì rất khó để xác định được biến nào sẽ đươc dùng làm ngõ vào cho mô hình. Trường hợp hệ thống động, còn phải ước lượng bậc của hệ thống. Đối với mô hình vào-ra NARX (nonlinear autoregressive with exogenous input) (3.67) thì cần phải lần lượt định nghĩa số ngõ vào và ngõ ra trễ n_y and n_u. Các kiến thức ban đầu, tìm được qua đáp ứng của quá trình cho mục tiêu mô hình hóa là nguồn thông tin tiêu biểu dùng cho các chọn lựa vừa nêu. Đôi khi có thể dùng phương pháp lựa chọn tự động từ dữ liệu, để so sánh các chọn lựa khác nhau, nhằm thỏa mãn một số tiêu chí về tính năng.
- *Cấu trúc các luật.* Lựa chọn này bao gồm dạng mô hình (dạng ngôn ngữ, singleton, quan hệ, Takagi-Sugeno) và dạng các tiền đề (antecedent form) (xem phần 3.2.6). Yếu tố quan trọng là mục tiêu mô hình hóa và kiến thức có được.
- *Số lượng và dạng các hàm thành viên cho mỗi biến.* Chọn lựa này nhằm xác định tầm mức của chi tiết tạo hạt (granularity) của mô hình. Một lần nữa, mục tiêu mô hình hóa và chi tiết về kiến thức có được sẽ gây ảnh hưởng lên chọn lựa. Các phương pháp tự động, dùng nguồn dữ liệu có thể được dùng để thêm, bớt các hàm thành viên trong mô hình.
- *Cơ chế suy diễn, các toán tử kết nối (connective operators), phương pháp giải mờ.* Các chọn lựa này bị giới hạn từ dạng của mô hình mờ (Mamdani, TS). Trong các hạn chế này thì tồn tại một số yếu tố tự do, thí dụ như chọn lựa các toán tử liên kết (conjunction operators), v.v,.. Nhằm tạo điều kiện dễ dàng cho việc tối ưu hóa dữ liệu của mô hình mờ (quá trình học), thường dùng nhiều toán tử khác nhau (tích, tổng) thay vì chỉ dùng các toán tử chuẩn là min và max.

Sau khi đã chọn xong mô hình, tính năng của mô hình mờ được tinh chỉnh bằng cách chỉnh định các tham số. Các tham số chỉnh định được của mô hình dạng ngôn ngữ là các tham số của các hàm thành viên tiền đề và hệ quả (xác định được từ vị trí và hình dạng các hàm này) và các luật (xác định được từ ánh xạ giữa vùng mờ tiền đề và vùng mờ hệ quả). Trong dạng mô hình mờ dạng quan hệ (fuzzy relational models), ánh xạ này được mã hóa trong quan hệ mờ. Mô hình Takagi-Sugeno có tham số trong hàm thành viên tiền đề và trong hàm hệ quả (**a** và **b** cho trường hợp mô hình affine TS).

2. Thiết kế tri thức nền

Thiết kế mô hình mờ dạng ngôn ngữ dùng nền tri thức chuyên gia, cần có các bước sau:

1. Chọn các biến vào và ra, cấu trúc các luật, phương pháp suy diễn và phương pháp giải mờ.

2. Quyết định số lượng thừa số ngôn ngữ cho từng biến và định nghĩa các hàm thành viên tương ứng.

3. Tạo lập kiến thức dùng các luật mờ nếu -thì.

4. Đánh giá mô hình (thường dùng tập dữ liệu). Nếu mô hình không khớp được với tính năng mong muốn, lặp lại các bước thiết kế.

Phương pháp này tương tự như phương pháp thiết kế bộ điều khiển mờ dùng heuristic (chương 6.3.4). Chú ý là thiết kế dùng nền tri thức thì phụ thuộc rất lớn vào bài toán phải làm, cũng như việc mở rộng và chất lượng của kiến thức đang có. Trong một số bài toán, có thể tìm ra nhanh được mô hình, trong một số trường hợp thì quá trình này lại tiêu tốn nhiều thời gian và không hiệu quả (đặc biệt khi phải tinh chỉnh thủ công các tham số mô hình). Như thế, nên kết hợp phương thức thiết kế trên nền tri thức với phương pháp chỉnh định tham số dùng dữ liệu. Phần tiếp sau điểm lại một số phương pháp dùng tinh chỉnh tham số mô hình mờ dùng tập dữ liệu.

3. Thu thập dữ liệu và tinh chỉnh các mô hình mờ

Tiềm năng lớn nhất của mô hình mờ là khả năng kết hợp kiến thức heuristic biểu diễn thành dạng các luật với thông tin có được từ đo lường dữ liệu. Trong phần này trình bày các phương pháp khác nhau dùng ước lượng và tối ưu hóa các tham số của mô hình mờ.

Giả sử có tập N các cặp dữ liệu vào-ra $\{(\mathbf{x}_i, y_i) \mid i = 1, 2, \ldots, N\}$ dùng cho cấu trúc hệ mờ. Nhắc lại, $\mathbf{x}_i \in R^p$ là các vectơ vào và y_i là ngõ ra vô hướng. Đặt $\mathbf{X} \in R^{N\times p}$ là ma trận có các cột là vectơ $\mathbf{x}^T_k$, và $\mathbf{y} \in R^N$ là vectơ chứa các ngõ ra y^k:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T, \ \mathbf{y} = [y_1, \ldots, y_N]^T. \qquad (5.1)$$

Phần tiếp theo trình bày cách ước lượng các tham số tiên đề và hệ quả.

3.1 Phương pháp ước lượng bình phương tối thiểu và hệ quả

Công thức giải mờ cho mô hình singleton và mô hình TS có dạng tuyến tính theo các tham số hệ quả, $\mathbf{a}_i$, b_i (xem lần lượt các phương trình (3.43) và (3.62)). Như thế, các tham số này có thể được ước lượng từ dữ liệu có được dùng kỹ thuật bình phương tối thiểu. Gọi $\mathbf{\Gamma}_i \in R^{N\times N}$ là ma trận đường chéo có mức độ hàm thành viên chuẩn hóa là $\gamma_i(\mathbf{x}_k)$ là các thành phần đường chéo thứ k. Dùng phép gắn thêm (append) một cột cho $\mathbf{X}$, tạo ma trận mở rộng $\mathbf{X}_e = [\mathbf{X}, \mathbf{1}]$. Hơn nữa, gọi $\mathbf{X'}$ là ma trận $R^{N\times K(p+1)}$ hợp thành từ các tích của ma trận $\mathbf{\Gamma}_i$ and $\mathbf{X}_e$

$$\mathbf{X'} = [\mathbf{\Gamma}_1\mathbf{X}_e, \mathbf{\Gamma}_2\mathbf{X}_e, \ldots, \mathbf{\Gamma}_K\mathbf{X}_e]. \qquad (5.2)$$

Các tham số hệ quả $\mathbf{a}_i$ và b_i được tính gộp vào một vectơ tham số $\boldsymbol{\theta} \in R^{K(p+1)}$:

$$\boldsymbol{\theta} = [\mathbf{a}^T_1, b_1, \mathbf{a}^T_2, b_2, \ldots, \mathbf{a}^T_K, b_K]^T. \qquad (5.3)$$

Cho dữ liệu $\mathbf{X}$, $\mathbf{y}$, phương trình eq. (3.62) viết thành dạng ma trận như sau, $\mathbf{y} = \mathbf{X'}\boldsymbol{\theta} + \varepsilon$. Từ đó, có thể giải tìm tham số $\boldsymbol{\theta}$:

$$\theta = \left[(X')^T X'\right]^{-1} (X')^T y. \tag{5.4}$$

Đây là nghiệm bình phương tối thiểu tối ưu cho sai số dự báo bé nhất, và thích hợp cho các mô hình dự báo (prediction models). Tuy nhiên, điều này có thể làm phân cực (bias) lên việc ước lượng tham số hệ quả thành tham số của mô hình cục bộ. Khi cần có ước lượng chính xác về tham số mô hình cục bộ, có thể dùng phương pháp bình phương tối thiểu (weighted least-squares) cho từng luật:

$$[\mathbf{a}^T_i\,,\, b_i]^T = [\mathbf{X}^t_e\ \mathbf{\Gamma}_i \mathbf{X}_e\}^{-1} \mathbf{X}^t_e\ \ \mathbf{\Gamma}_i \mathbf{y}\,. \tag{5.5}$$

Trường hợp này thì tham số hệ quả cho từng luật được ước lượng một cách độc lập, do đó không phân cực "biased" tương tác lên các luật khác. Khi bỏ qua $\mathbf{a}_i$ với mọi $1 \le i \le K$, và đặt $\mathbf{X}_e = \mathbf{1}$, thì các phương trình (5.4) và (5.5) áp dụng trực tiếp vào mô hình singleton (3.42).

3.2 Mô hình hóa dùng bảng mẫu (template)

Trong phương pháp này thì miền của các biến tiền đề được chia thành đơn giản vào các số đặc thì của các hàm thành viên có dạng và được phân bố giống nhau. Luật nền được thiết lập để bao gồm mọi tổ hợp của các thừa số tiền đề. Các tham số hệ quả được ước lượng dùng phương pháp bình phương tối thiểu.

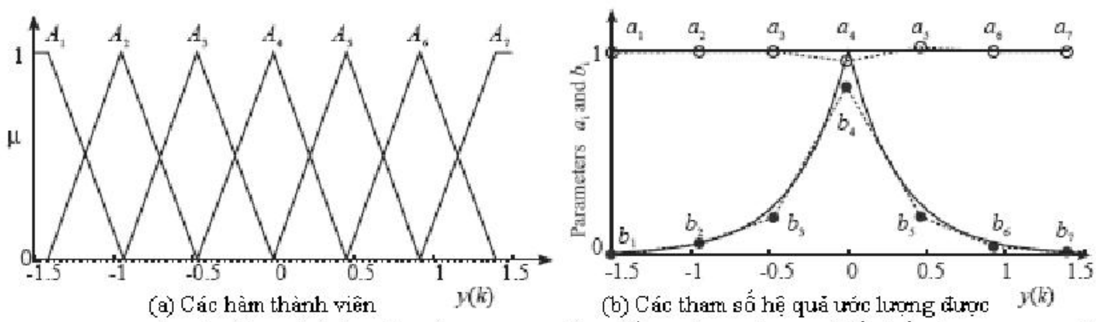
Thí dụ 5.1 Xét hệ động phi tuyến được mô tả dạng phương trình sai phân bậc nhất:

$$y(k+1) = y(k) + u(k)e^{-3|y(k)|}\,. \tag{5.6}$$

Dùng các tín hiệu vào dạng bậc thang (stepwise) để tạo ra cùng phương trình này một tập gồm 300 cặp vào-ra (xem hình 5.2a). Giả sử biết được hệ thống là bậc nhất và tính phi tuyến chỉ xuất phát từ y, chọn cấu trúc luật TS như sau:

$$\textbf{Nếu } y(k) \text{ là } Ai \textbf{ thì } y(k+1) = a_i y(k) + b_i u(k). \tag{5.7}$$

Giả sử, chưa có được các kiến thức trước đó, dùng bảy hàm thành viên dạng tam giác cách đều nhau, từ A_1 đến A_7, được định nghĩa trong miền của $y(k)$, và vẽ ở hình 5.1a.



Hình 5.1: a) Các hàm thành viên dạng tam giác giống nhau được thiết kế cho ngõ ra y(k); b) So sánh giữa tính phi tuyến thực của hệ thống (đường sậm) và thừa số xấp xỉ theo tham số hệ quả ước lượng được (đường đứt nét)

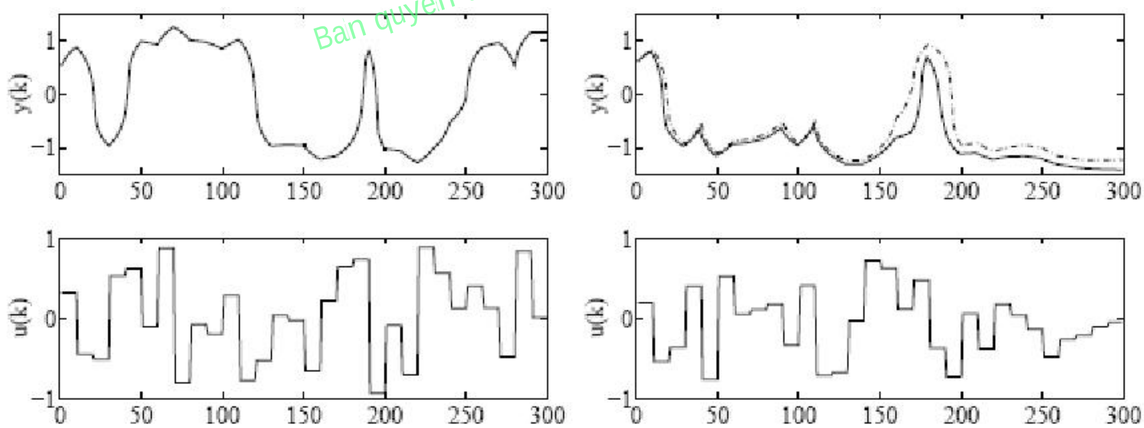
Tham số hệ quả được ước lương dùng phương pháp bình phương tối thiểu. Hình 5.1b minh họa các tham số a_i, b_i theo cores của tập mờ tiền đề A_i. Trong đó cũng vẽ phép nội suy tuyến tính giữa các tham số (đường đứt nét) và tính phi tuyến thực của hệ (đường sậm). Phép nội suy giữa a_i và b_i là tuyến tính, do các hàm thành viên có dạng tuyến tính hóa từng đoạn (tam giác). Có thể thấy là phụ thuộc giữa tham số hệ quả và tham số tiền đề xấp xỉ gần đúng được tính phi tuyến của hệ thống, điều này làm cho mô hình trở nên minh bạch. Các giá trị,

$\mathbf{a}^T = [1.00,\ 1.00,\ 1.00,\ 0.97,\ 1.01,\ 1.00,\ 1.00]$

và $\mathbf{b}^T = [0.01,\ 0.05,\ 0.20,\ 0.81,\ 0.20,\ 0.05,\ 0.01]^T$, cho thấy mức phi tuyến cao của ngõ vào và đặc tính động tuyến tính trong (5.6). Đánh giá mô hình qua mô phỏng vơi nhiều tập dữ liệu cho ở hình 5.2b.

Cấu trúc cục bộ trong suốt của mô hình TS cho phép tổ hợp dễ dàng các mô hình cục bộ đã có từ phép ước lượng tham số và phép tuyến tính hóa các mô hình mechanistic đã biết (mô hình hộp trắng). Nếu đo lường được trong một tầm của vùng hoạt động của quá trình, thì các tham số của phần còn lại có thể tìm được bằng cách tuyến tính hóa (dùng giá trị cục bộ) của mô hình mechanistic của quá trình. Giả sử mô hình được cho bởi hàm $y = f(\mathbf{x})$. Phép tuyến tính hóa xung quanh tâm $\mathbf{c}_i$ của luật thứ i của hàm thành viên tiền đề cho các tham số sau của mô hình affine TS (3.61):

$$a_i = \frac{df}{dx}\Big|_{x=ci} \qquad\qquad b_i = f(c_i) \qquad (5.8)$$



(a) Dữ liệu nhận dạng (b) Đánh giá

Hình 5.2 Tập dữ liệu nhận dạng (a) và tính năng của mô hình so với tập đánh giá (b). Đường sậm: qúa trình, đường đứt nét: mô hình

Một yếu diểm của phương pháp bảng mẫu (template-based approach) là số luật trong mô hình có thể phát triển rất nhanh. Nếu không có hiểu biết về biến tạo ra tính phi tuyến, thì nên phân các biến tiền đề thành các partition đồng đều (partitioned uniformly). Tuy nhiên, độ phức tạp của đáp ứng thường là đồng đều. Một số vùng hoạt động có thể được xấp xỉ tốt dùng một mô hình đơn, trong khi trong một số vùng khác thì đòi hỏi có sự phân chia nhỏ hơn. Để biểu diễn hiệu quả hơn mà dùng ít luật, thì thay đổi các hàm thành viên sao cho chúng nắm bắt được đáp ứng không đồng đều của hệ thống. Điều này cần có đo lường về hệ thống để tạo hàm thành viên, như thảo luận ở phần dưới đây.

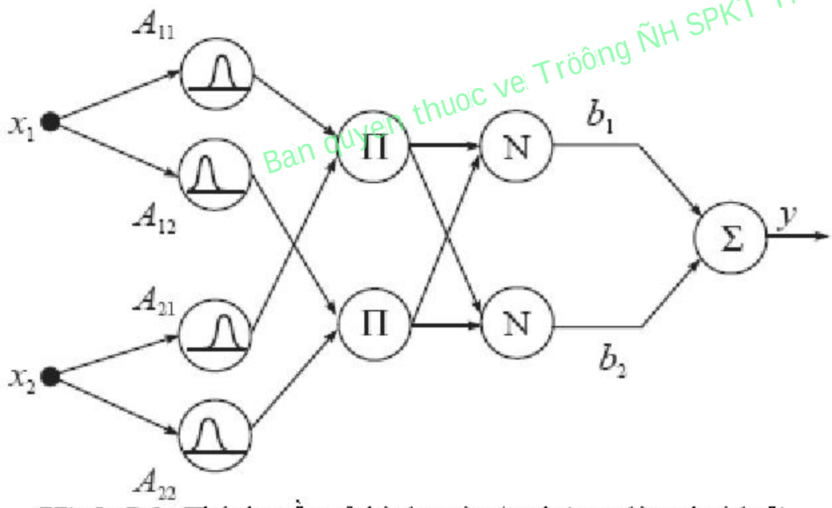
3.3 Mô hình neural-fuzzy

Ta đã thấy là các tham số tuyến tính có liên quan đến ngõ ra có thể được ước lượng (một cách tối ưu) dùng phương pháp bình phương tối thiểu. Để tối ưu hóa các tham số có liên quan đến ngõ ra theo hướng phi tuyến, có thể dùng các thuật toán huấn luyện của mạng nơrôn hay phương pháp tối ưu hóa phi tuyến. Các kỹ thuật này khi thác ở cấp độ tính toán, thì mô hình mờ có thể được xem là (mạng) có cấu trúc thành lớp, tương tự như mạng nơrôn nhân tạo.. Như thế, xu hướng này đưa đến phương pháp mờ-nơrôn (Jang và Sun, 1993; Brown và Harris, 1994; Jang, 1993).

Hình 5.3 cho thấy một thí dụ về mô hình mờ singleton có hai luật được biểu diễn theo dạng mạng. Các luật đó là:

Nếu x_1 là A_{11} **và** x_2 là A_{21} **thì** $y = b_1$.
Nếu x_1 là A_{12} **và** x_2 là A_{22} **thì** $y = b_2$.

Các nút trong lớp thứ nhất tính mức độ hàm thành viên của các ngõ vào trong tập mờ tiền đề. Nút tích Π trong lớp thứ hai biểu diễn toán tử kết nối hệ quả. Nút chuẩn hóa N và nút tổng Σ thực hiện toán tử fuzzy-mean (3.43).



Hình 5.3: Thí dụ về mô hình mờ singleton dùng hai luật biểu diễn mạng mờ-nơrôn

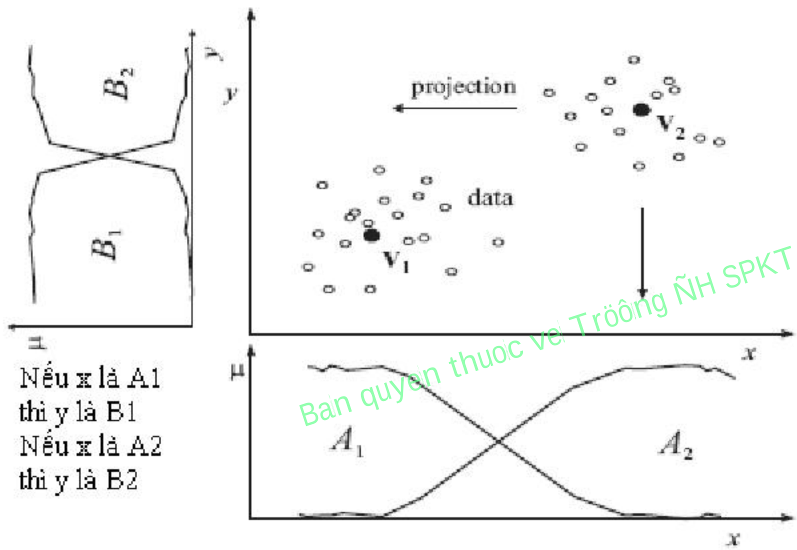
Khi dùng các hàm thành viên tiền đề có dạng mịn (thí dụ hàm Gauss)

$$\mu_{Aij}\left(x_j; c_{ij}, \sigma_{ij}\right) = \exp\left(-\left(\frac{x_j - c_{ij}}{2\sigma_{ij}}\right)^2\right) \tag{5.10}$$

Trong đó các tham số c_{ij} và σ_{ij} có thể được chỉnh định theo thuật toán học dùng phương pháp giảm theo gradien, như phép lan truyền ngược (xem phần 7.6.3).

3.4 Mô hình hóa dùng phương pháp xâu chuỗi mờ

Phương pháp nhận dạng dùng xâu chuỗi mờ có cội nguồn là phân tích nhận dạng mẫu, trong đó ý niệm về thành viên xếp hạng (graded membership) được dùng để biểu diễn mức độ của đối tượng cho trước, được biểu diễn như vectơ đặc trưng (vector of features) tương tự như trong một dố đối tượng nguyên mẫu (prototypical object). Mức độ tương đồng (degree of similarity) có thể được tính toán dùng phép đo lường cự ly thích hợp. Trên cơ sở tính tương đồng, vectơ đặc trưng có thể được xâu chuỗi (clustered) sao cho các vectơ trong chuỗi là càng tương đồng càng tốt, và các vectơ trong các chuỗi khác nhau càng không tương đồng càng tốt (xem chương 4).



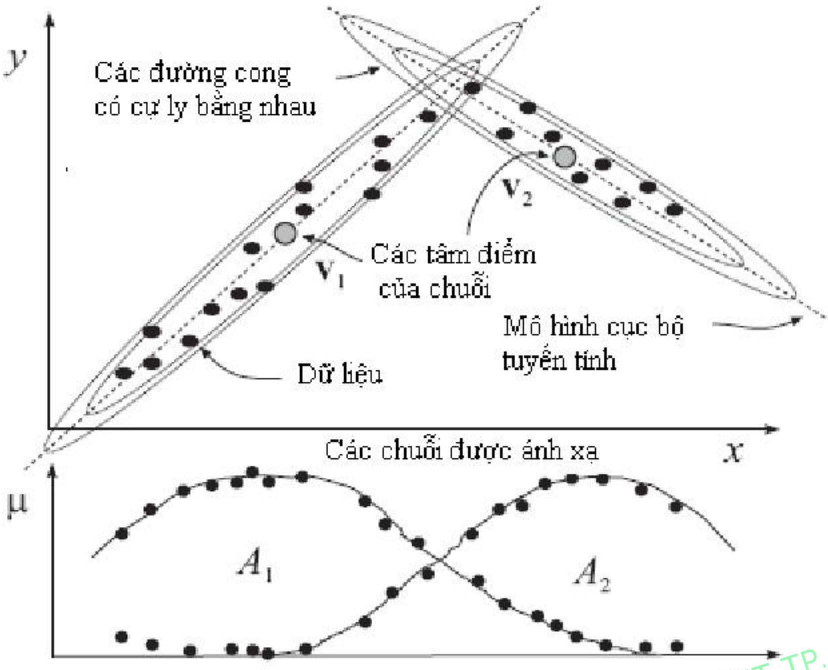
Hình 5.4: Diễn đạt luật nền từ phép xâu chuỗi mờ

Hình 5.4 cho thí dụ về tập dữ liệu trong R^2 được xâu chuỗi thành hai nhóm với các prototypes $\mathbf{v}_1$ và $\mathbf{v}_2$, dùng đo lường cự ly dạng Eucle. Rút ra được luật nếu-thì mờ bằng cách chiếu các chuỗi lên các trục.

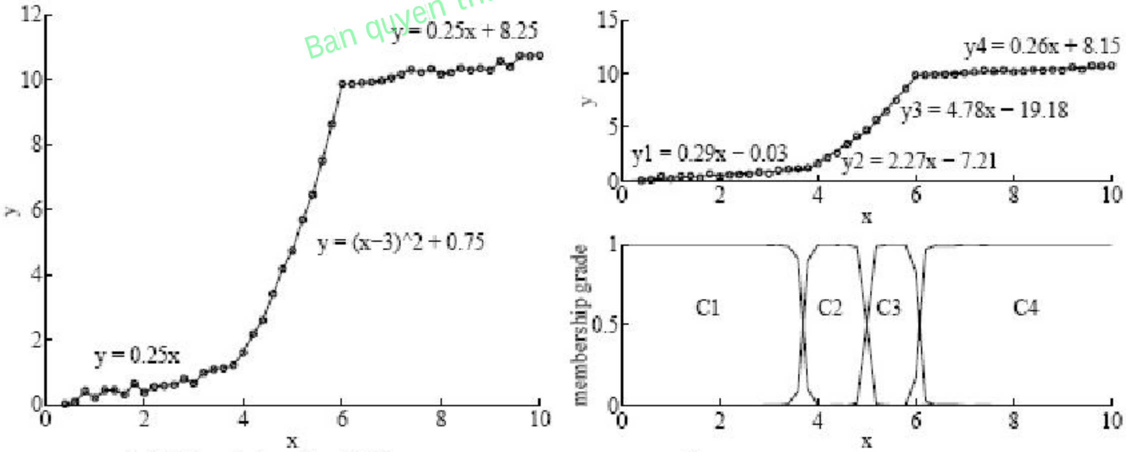
Các prototypes còn có thể được định nghĩa là các không gian con tuyến tính, (Bezdek, 1981) hay là các chuỗi có thể có dạng ellip dùng dạng ellip được xác định thích nghi (thuật toán Gustafson–Kessel, xem phần 4.4). Từ các chuỗi này, rút ra được các hàm thành viên tiền đề và các tham số hệ quả của mô hình Takagi–Sugeno (Hình 5.5):

Nếu x là A_1 **thì** $y = a_1x + b_1$,
Nếu x là A_2 **thì** $y = a_2x + b_2$,

Mỗi chuỗi tìm được biểu diễn thành một luật trong mô hình Takagi–Sugeno. Hàm thành viên của tập mờ A_1 và A_2 được tạo ra từ ánh xạ điểm-điểm (point-wise projection) của các ma trận partition lên các biến tiền đề. Tiếp đến, các điểm định nghĩa tập mờ này được xấp xỉ dùng các hàm tham số thích hợp. Các tham số hệ quả của từng luật được xác định dùng phương pháp ước lượng bình phương tối thiểu (5.4) hay (5.5).



Hình 5.5: Xâu chuỗi mờ dạng Hyper ellip



(a) Hàm phi tuyến (5.12) Chuỗi prototype và các tập mờ tương ứng

Hình 5.6: Xấp xỉ một hàm phi tuyến tĩnh dùng mô hình mờ Takagi-Sugeno

Thí dụ 5.2 Xét hàm phi tuyến $y = f(x)$ được định nghĩa theo điểm như sau:

$$\begin{aligned} &y = 0.25x, \ \text{for } x \leq 3 \\ &y = (x-3)^2 + 0.75, \ \text{for } 3 < x \leq 6 \\ &y = 0.25x + 8.25, \ \text{for } x > 6 \end{aligned} \tag{5.12}$$

Hình 5.6a vẽ đồ thị của hàm này ước lượng với 50 mẫu phân bố đều trong tầm $x \in [0, 10]$. Nhiễu trung bình là Zero, phân bố đều có biên độ 0.1 được cộng vào y.

Tập dữ liệu $\{(x_i, y_i) | i = 1,2,\ldots,50\}$ được xâu chuỗi thành bốn chuỗi hyperellipsoidal. Hình phía trên trong 5.6b vẽ mô hình tuyến tính cục bộ có được từ phép xâu chuỗi, đồ thị phía dưới vẽ các partition mờ. Dùng luật dạng TS, có mô hình mờ là:

X_1: **Nếu** x là C_1 **thì** $y = 0.29x - 0.03$
X_2: **Nếu** x là C_2 **thì** $y = 2.27x - 7.21$
X_3: **Nếu** x là C_3 **thì** $y = 4.78x - 19.18$
X_4: **Nếu** x là C_4 **thì** $y = 0.26x + 8.15$

Chú ý là các hệ quả X_1 và X_4 thì hầu như tương ứng chính xác với phương trình thứ nhất và thứ ba của (5.12). Các hệ quả X_2 và X_3 xấp xỉ vuông góc với parabol định nghĩa bởi phương trình thứ hai của (5.12) lần lượt theo các tâm của chuỗi.

Nguyên lý nhận dạng trong không gian tích mở rộng thẳng được đến hệ vào-ra động. Trường hợp này, thì không gian tích được tạo bởi các bộ hồi qui (regressors) từ các ngõ vào, các ngõ ra có trễ và regressand (là ngõ ra cần dự báo). Thí dụ, xét mô hình NARX bậc hai $y(k + 1) = f\,(y(k),\ y(k - 1),\ u(k),\ u(k - 1))$. Tập giá trị đo lường được là, $S = \{(u(j),\ y(j)) \mid j = 1, 2, \ldots, Nd\}$, ma trận hồi qui (regressor matrix) và vectơ regressand (regressand vector) là:

$$X = \begin{bmatrix} y(2) & y(1) & u(2) & u(1) \\ y(3) & y(2) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ y(N_d - 1) & y(N_d - 2) & u(N_d - 1) & u(N_d - 2) \end{bmatrix}, \qquad y = \begin{bmatrix} y(3) \\ y(4) \\ \vdots \\ y(N_d) \end{bmatrix},$$

Trong thí dụ này thì $N = N_d - 2$ hàm phi tuyến ẩn $y = f(\mathbf{x})$ biểu diễn mặt (siêu) phẳng phi tuyến trong không gian tích: $(X \times Y) \subset \mathrm{R}^{p+1}$. Mặt phẳng này được gọi là *mặt phẳng hồi qui*. Các dữ liệu có được biểu diễn một mẫu từ mặt phẳng hồi qui. Khi xâu chuỗi dữ liệu, tìm được các mô hình tuyến tính cục bộ bằng cách xấp xỉ mặt phẳng hồi qui.

Thí dụ 5.3 Có thể quan sát được mặt phẳng hồi qui trong các hệ thống có bậc thấp. Thí dụ, xét hệ nối tiếp gồm khâu phi tuyến chết/bảo hòa tĩnh có hệ động tuyến tính bậc một:

$$y(k + 1) = 0.6y(k) + w(k), \qquad (5.13a)$$

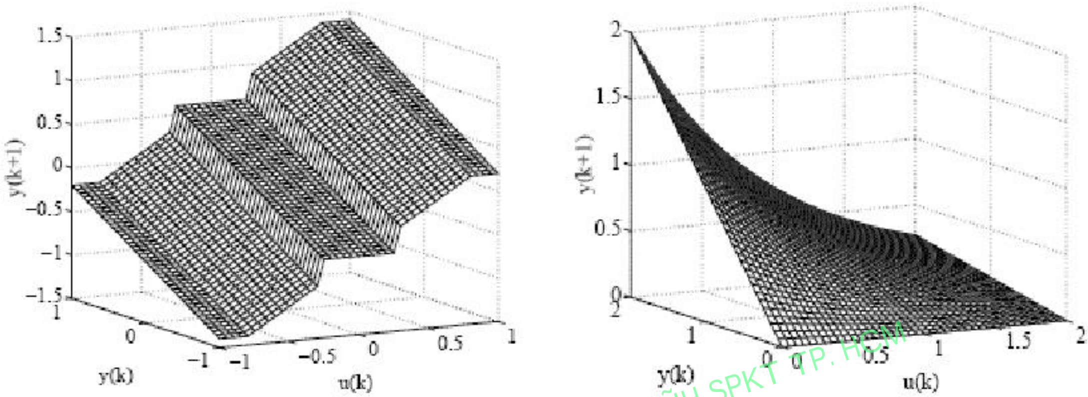
trong đó $w = f(u)$ cho bởi:

$$w = \begin{cases} 0 & -0{,}3 \le u \le 0{,}3 \\ u, & 0{,}3 \le |u| \le 0{,}8 \\ 0{,}8 sign(u) & 0{,}8 \le |u| \end{cases} \qquad (5.13b)$$

Mô tả vào-ra của hệ thống dùng mô hình NARX (3.67) có thể được xem là mặt phẳng trong không gian $(U \times Y \times Y) \subset \mathrm{R}^3$, vẽ ở hình 5.7a. Thí dụ khác, xem hệ không gian-trạng thái (Chen và Billings, 1989):

$$x(k+1) = x(k) + u(k),$$
$$y(k) = \exp(-x(k)). \qquad (5.14)$$

Trong thí dụ này, tìm được mô hình hồi qui vào-ra $y(k+1) = y(k)\exp(-u(k))$. Mặt phẳng hồi qui tương ứng vẽ ở hình 5.7b. Chú ý là nếu có đươc các đo lường trạng thái, thì có thể xấp xỉ riêng lẽ các ánh xạ trạng thái và ngõ ra của hệ, để có được bài toán tuyến tính (two-variate) và bài toán phi tuyến (univariate) và d=gải được dễ dàng.



(a) Hệ có khâu trễ và bão hòa (b) Hệ y(k+1)=y(k)exp(-u(k)).

Hình 5.7: Mặt phẳng hồi qui của hai hệ thống phi tuyến động

Thí dụ 5.4 (Nhận dạng hệ tự hồi qui (Autoregressive System**))** Xét chuỗi thời gian sinh ra từ hệ tự hồi qui phi tuyến được định nghĩa (Ikoma và Hirota, 1993):

$$y(k+1) = f\big(y(k)\big) + u(k), \qquad f(y) = \begin{cases} 2y-2 & 0{,}5 \le y \\ -2y & -0{,}5 < y < 0{,}5 \\ 2y+2 & y \le -0{,}5 \end{cases} \qquad (5.16)$$

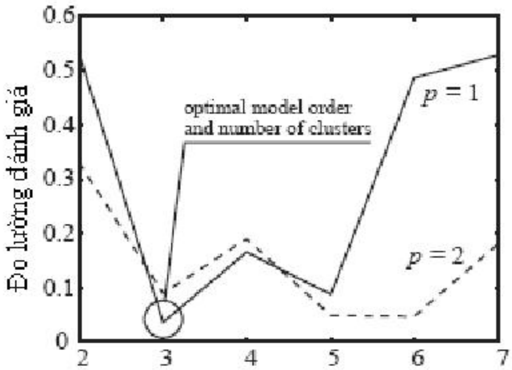
Trường hợp này, $\in(k)$ là biến ngẫu nhiên độc lập của $N(0, \sigma^2)$ với $\sigma = 0.3$. Từ dữ liệu có được $x(k)$, $k = 0, \ldots, 200$, và điều kiện đầu $x(0) = 0.1$, thì 100 điểm đầu tiên được dùng cho nhận dạng và các điểm còn lại dùng cho đánh giá mô hình. Dùng phép xâu chuỗi mờ, tìm được mô hình affine TS với ba tập mờ tham chiếu. Giả sử chỉ có được một kiến thức ban đầu được hệ tự hồi qui phi tuyến tạo ra theo:

$$y(k+1) = f(y(k), y(k-1), \ldots, y(k-p+1)) = f(\mathbf{x}(k)), \qquad (5.17)$$

với p là bậc của hệ thống. Trường hợp này $\mathbf{x}(k) = [y(k), y(k-1), \ldots, y(k-p+1)]^T$ là veto hồi quy và $y(k+1)$ là biến đáp ứng. Ma trận **Z** được tạo ra từ dữ liệu nhận dạng:

$$Z = \begin{bmatrix} y(p) & y(p+1) & \cdots & y(N-1) \\ \cdots & \cdots & \cdots & \cdots \\ y(1) & y(2) & \cdots & y(N-p) \\ y(p+1) & y(p+2) & \cdots & y(N) \end{bmatrix} \qquad (5.18)$$

Để nhận dạng được hệ thống thì cần tìm bậc p và dùng phép affine TS để xấp xỉ hàm f. Bậc của hệ thống và số chuỗi (clusters) có thể được xác định dùng phép đo lường đánh giá chuỗi, và đạt giá trị thấp khi nằm ở các partitions "tốt" (Babuska, 1998). Đo lường đánh giá này được tính toán trong tầm của các bậc mô hình $p = 1, 2 \ldots, 5$ và số lượng chuỗi $c = 2, 3 \ldots, 7$. Kết quả cho ở dạng ma trận trong hình 5.8b. Trị tối (gía trị in đậm) có được khi $p = 1$ và $c = 3$ tương ứng với (5.16). Trong hình 5.8a thì đo lường đánh giá được vẽ là hàm theo c với các bậc $p = 1, 2$. Chú ý là trong hàm này có thể có nhiều cực tiểu cục bộ, và thường được chọn nhằm có được mô hình đơn giản với ít luật.



(a) Số xâu chuỗi

Số xâu chuỗi \ p	1	2	3	4	5
2	0.53	0.33	0.50	4.64	1.27
3	**0.03**	0.08	0.07	0.21	2.45
4	0.16	0.19	5.62	0.36	1.60
5	0.08	0.04	0.06	0.18	0.27
6	0.48	0.04	0.43	0.22	0.51
7	0.52	0.18	2.07	0.12	0.13

(b)

Hình 5.8: Đo lường đánh giá với nhiều bậc mô hình và số chuỗi khác nhau

Hình 5.9a vẽ ánh xạ chuỗi vào biến $y(k)$ với hệ đúng bậc $p = 1$ và số chuỗi $c = 3$. Hình 5.9b vẽ các chuỗi (cluster prototypes):

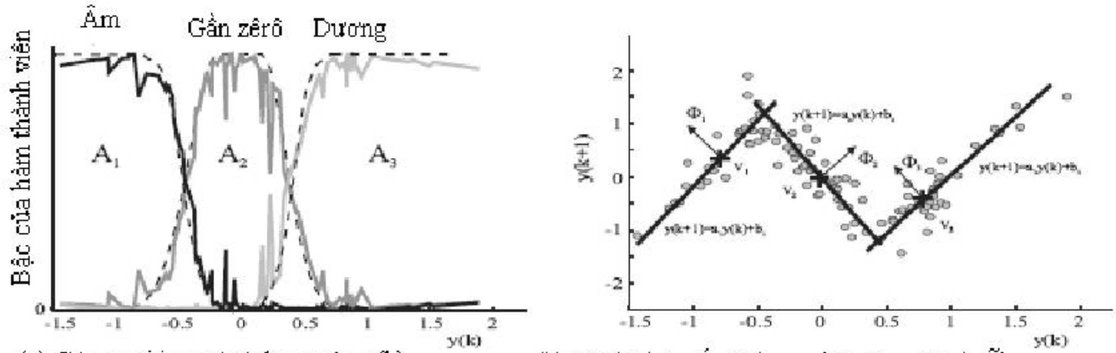
$$V = \begin{bmatrix} -0{,}772 & -0{,}019 & 0{,}751 \\ 0{,}405 & 0{,}098 & -0{,}410 \end{bmatrix}$$

Từ ma trận chuỗi đồng phương sai (covariance) cho dưới đây ta thấy phương sai (variance) trong một chiều thì cao hơn theo chiều khác, như thế hyperellipsoids là phẳng và mô hình có thể được biểu diễn theo quan hệ hàm giữa các biến khi xâu chuỗi:

$$F_1 = \begin{bmatrix} 0.057 & 0{,}099 \\ 0{,}099 & 0{,}249 \end{bmatrix} \qquad F_2 = \begin{bmatrix} 0.063 & -0{,}099 \\ -0{,}099 & 0{,}224 \end{bmatrix} \qquad F_3 = \begin{bmatrix} 0.065 & 0{,}107 \\ 0{,}107 & 0{,}261 \end{bmatrix}$$

Điều này được xác nhận bằng cách xem xét các trị riêng (eigenvalues) của ma trận (covariance matrices):

$$\lambda_{1,1} = 0.015, \quad \lambda_{1,2} = 0.291,$$
$$\lambda_{2,1} = 0.017, \quad \lambda_{2,2} = 0.271,$$
$$\lambda_{3,1} = 0.018, \quad \lambda_{3,2} = 0.308.$$



(a) Các partition mờ ánh xạ vào y(k) (b) Mô hình tuyến tính cục bộ rút ra từ chuỗi

Hình 5.9: Kết quả từ phép xâu chuỗi mờ khi p = 1 và c = 3. Phần (a) vẽ các hàm thành viên có được từ ánh xạ các ma trận partition và y(k). Phần (b) vẽ các chuỗi prototype v_i , hướng của các vectơ riêng Φ_i và chiều của mô hình hệ quả (đường thẳng)

Có thể thấy là với mỗi chuỗi, thì bậc của trị riêng (eigenvalues) nhỏ hơn một bậc. Dùng ước lượng bình phương tối thiểu, tìm được các tham số a_i và b_i của mô hình affine TS. Hàm thành viên dạng mũ tuyến tính hóa (2.14) được dùng để định nghĩa các tập mờ tiền đề (antecedent fuzzy sets). Các hàm này khớp với các chuỗi ánh xạ A_1 đến A_3 bằng cách tối ưu hóa số các tham số c_l, c_r, w_l và w_r. kết quả được vẽ thành đường đứt nét trong hình 5.9a. Sau khi đặt tên các tập mờ này lần lượt là ÂM, GẦN ZÊRO và DƯƠNG, thì mô hình TS được viết theo:

Nếu $y(k)$ là ÂM thÌ $y(k+1) = 2.371y(k) + 1.237$
Nếu $y(k)$ là GẦN ZÊRÔ thì $y(k+1) = -2.109y(k) + 0.057$
Nếu $y(k)$ là DƯƠNG thì $y(k+1) = 2.267y(k) - 2.112$

Các tham số hệ quả ước lượng được tương ứng một cách xấp xỉ với định nghĩa của đoạn đường trong phần xác định (5.16). Đồng thời, partition của miền tiền đề được chấp nhận trong định nghĩa của hệ thống.

4. Mô hình Semi-Mechanistic

Cùng với thấu hiểu vật lý về hệ thống, thì có thể tạo biến đổi phi tuyến từ các tín hiệu đo lường được. Thí dụ, khi mô hình hóa quan hệ giữa nhiệt độ phòng và điện áp đặt vào lò điện, thì tín hiệu công suất được tính từ bình phương điện áp, do chính công suất chứ không phải là điện áp đã tạo nên sự thay đổi của nhiệt độ (Lindskog and Ljung, 1994). Biến mới này lại được dùng trong mô hình hộp đen tuyến tính thay cho điện áp. Nhu cầu về một bộ hồi qui phi tuyến (nonlinear regressors) không phải là vô ích (luật, tham số, v.v,…) để ước lượng sự kiện mà ta thực sự đã biết.

Một xu hướng nữa là kết hợp giữa mô hình hộp đen và mô hình hộp trắng. Trong một số hệ thống, thí dụ như trong các quá trình sinh-hóa học, nhiệm vụ mô hình hóa có thể chia thành hai nhiệm vụ con: mô hình hóa một cơ chế đã biết rõ trên cơ sở cân bằng giữa khối lượng và năng lượng (nguyên tắc thứ nhất trong mô hình hóa: first-principle modeling), và việc xấp xỉ phần quan hệ chưa biết rõ thí dụ như tốc độ phản ứng đặc thù. Một số xu hướng mô hình hóa hỗn hợp đã đề nghị việc kết hợp nguyên lý thứ nhất với mô hình hóa hộp đen phi tuyến, thí dụ dùng mạng nơrôn (Psichogios and Ungar, 1992; Thompson and Kramer, 1994) hay dùng mô hình mờ (Babuska, et al., 1999). Mạng nơrôn hay mô hình mờ thường được dùng làm bộ xấp xỉ phi tuyến vạn năng nhằm “học” quan hệ chưa biết (ẩn) từ dữ liệu và dùng như bộ dự báo (predictor) cho đại lượng chưa đo được của quá trình, là công việc rất khó khăn khi ta dùng nguyên lý thứ nhất.

Thí dụ, mô hình hóa bộ fed-batch stirred bioreactor được mô tả từ phương trình lấy từ cân bằng khối lượng (Psichogios and Ungar, 1992):

$$\frac{dX}{dt} = \eta(.)X - \frac{F}{V}X \tag{5.20a}$$

$$\frac{dS}{dt} = -k_1\eta(.)X + \frac{F}{V}[S_i - S] \tag{5.20b}$$

$$\frac{dV}{dt} = F \tag{5.20c}$$

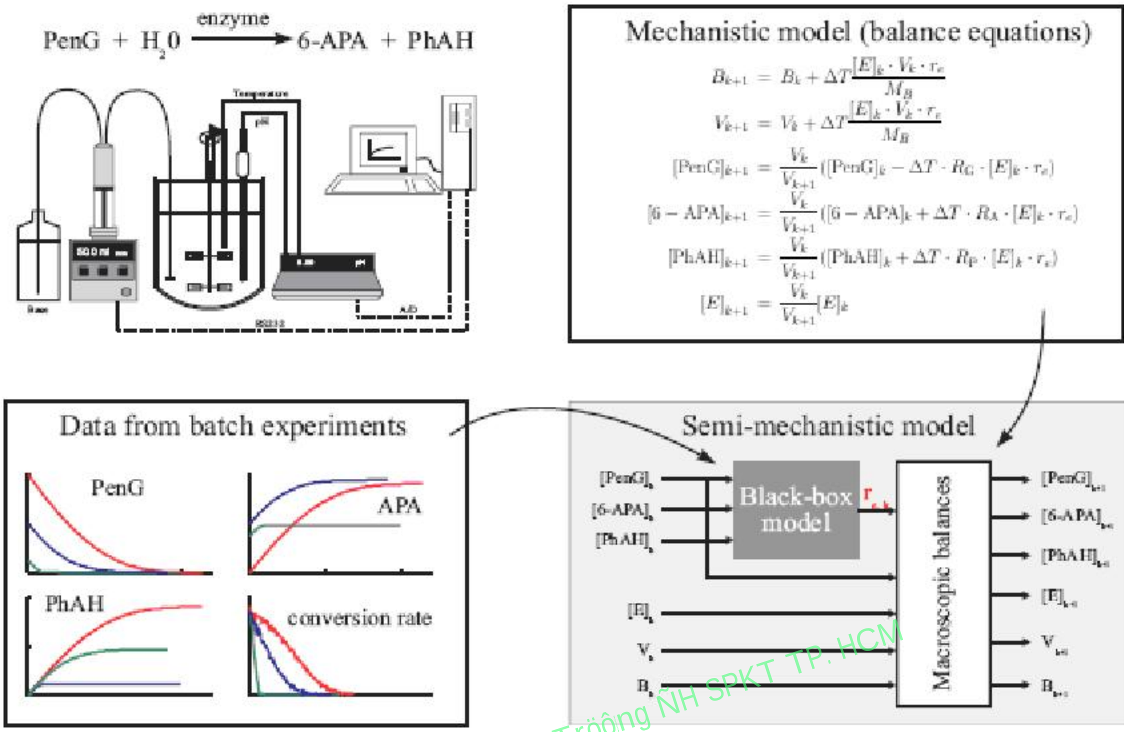
Trong đó X là mật độ sinh khối, S là mật độ chất nền, V là thể tích lò phản ứng, F là lưu tốc vào, k_1 là hệ số chuyển đổi từ substrate đến cell, và S_i là mật độ tại ngõ vào. Cân bằng khối lượng cung cấp mộ hình từng phần. Động lực của quá trình được biểu diễn từ tốc độ tăng trưởng đặc thù $\eta(\cdot)$ nhằm tính toán chuyển đổi từ nền đến sinh khối, và thường là hàm phi tuyến phức tạp theo biến quá trình. Tuy có nhiều mô hình khác nhau đã được đề cập, nhưng việc lựa chọn mô hình đúng cho quá trình là không dễ dàng. Hướng hỗn hợp dùng phương pháp xấp xỉ $\eta(\cdot)$ bằng một mô hình phi tuyến (hộp đen) từ đo lường của quá trình và kết hợp quan hệ phi tuyến nhận dạng được trong mô hình hộp trắng. Dữ liệu có thể tìm từ thực nghiệm (batch experiments) với $F = 0$, và phương trình (5.20a) rút thành biểu thức:

$$\frac{dX}{dt} = \eta(.)X \tag{5.21}$$

Trong đó $\eta(\cdot)$ là đã biết. Mô hình này được dùng trong mô hình hộp trắng cho bởi phương trình (5.20) cho hai chế độ batch và fed-batch. Một thí dụ về ứng dụng của hướng semi-mechanistic là việc mô hình hóa (enzymatic Penicillin G conversion) (Babuska, et al., 1999), xem hình 5.10.

5. Tóm tắt và các vấn đề cần quan tâm

Mô hình hóa mờ là khung sườn cho việc kết hợp nhiều phương pháp mô hình hóa và nhận dạng khác nhau, và cung cấp một giao diện trong suốt với nhà thiết kế hay người vận hành. Ngoài ra, đây còn là một công cụ mềm dẽo để mô hình hóa và điều khiển hệ thống phi tuyến. Khả năng dùng luật của mô hình mờ cho phép mô hình diễn đạt tương tự như phương thức con người mô tả thế giới thực. Các phương pháp truyền thống dùng phương thức đánh giá thống kê trên cơ sở dữ liệu số học được bổ sung từ kinh nghiệm của người chuyên gia, thường bao hàm tri thức heuristic cùng yếu tố trực giác.



Hình 5.10 Ứng dụng hướng mô hình hóa semi-mechanistic trong quá trình chuyển đổi Penicillin G

6. Bài tập

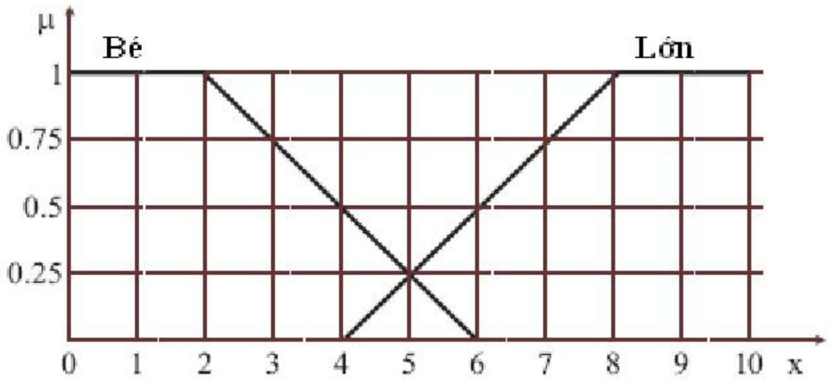
1. Cho biết các bước cần thực hiện khi thiết kế mô hình mờ trên nền trị thức (knowledge-based fuzzy model). Một trong những thế mạnh của hệ mờ là khả năng tích hợp các kiến thức đã có và dữ liệu. Giải thích về phương thức thực hiện điều này trong hệ mờ.

2. Xét mô hình mờ singleton fuzzy $y = f(x)$ có hai luật sau::
i) **Nếu** x là *Bé* **thì** $y = b1$, ii) **Nếu** x là *Lớn* **thì** $y = b2$.
và hàm thành viên cho trong bảng 5.11.
Ngoài ra, còn có tập dữ liệu:

$x1 = 1$, $y1 = 3$

$x2 = 5$, $y2 = 4.5$

Tính toán các tham số hệ quả (consequent parameters) $b1$ và $b2$ để mô hình có tổng sai số bình phương là bé nhất. Cho biết giá trị này?



Hình 5.11: Các hàm thành viên

3. Xét các luật mờ sau cùng các hệ quả singleton (singleton consequents):
i) **Nếu** x là $A1$ **và** y là $B1$ **thì** $z = c1$, iii) **Nếu** x là $A1$ **và** y là $B2$ **thì** $z = c3$,
ii) **Nếu** x là $A2$ **và** y là $B1$ **thì** $z = c2$, iv) **Nếu** x là $A2$ **và** y *là* $B2$ **thì** $z = c4$.
Vẽ sơ đồ mạng nơrôn-mờ tương ứng. Cho biết về các tham số tự do (chỉnh định được) trong mạng này? Cho biết có thể dùng phương pháp nào để tối ưu hóa tham số từ tập dữ liệu vào-ra?

4. Viết phương trình tổng quát của mô hình NARX (nonlinear autoregressive with exogenous input). Giải thích các ký hiệu, và cho thí dụ về một số mô hình NARX.

5. Giải thích thuật ngữ mô hình hóa semi-mechanistic (hỗn hợp). Bạn hiểu gì về thuật ngữ "*chọn lọc cấu trúc*" (structure selection) và "*ước lượng tham số*" (parameter estimation) dùng trong mô hình này?

CHƯƠNG SÁU: ĐIỀU KHIỂN MỜ DÙNG PHƯƠNG PHÁP TRI THỨC

Phần này trình bày nguyên tắc điều khiển mờ trên nền tri thức dùng các sơ đồ điều khiển mờ cơ bản. Trong đó, chú trọng đến quá trình thiết kế heuristic cho bộ điều khiển mờ. Các thiết kế dùng mô hình được thảo luận trong chương 8.
Điều khiển tự động là một trong những ứng dụng quan trọng nhất của lý thuyết tập mờ. Năm 1974, có báo cáo về ứng dụng thành công logic mờ trong điều khiển (Mamdani, 1974). Quá trình điều khiển mờ lò nung xi măng được ứng dụng trong công nghiệp sớm nhất (Holmblad and Østergaard, 1982). Từ sản phẩm dùng logic mờ đầu tiên được đưa vào thị trường năm 1987, ứng dụng điều khiển mờ đã gia tăng dần. Một số môi trường CAD dùng cho thiết kế hệ điều khiển mờ đã hòa quyện cùng phần cứng VLSI trong các ứng dụng tốc độ cao. Điều khiển mờ đã được ứng dụng trong công nghiệp chế biến (Froese, 1993; Santhanam and Langari, 1994; Tani, et al., 1994), điện tử tiêu dùng (Hirota, 1993; Bonissone, 1994), vận hành tự động xe lửa (Yasunobu and Miyamoto, 1985) và điều khiển giao thông (Hellendoorn, 1993), cùng nhiều lĩnh vực khác (Hirota, 1993; Terano, et al., 1994).
Chương này trình bày các yếu tố đầu tiên đã thúc đẩy hệ điều khiển mờ. Tiếp đến, giải thích các ý niệm khác nhau về điều khiển mờ: Mamdani, Takagi–Sugeno và hệ thống điều khiển mờ có giám sát. Sau cùng, trình bày các công cụ phần cứng và phần mềm giúp thiết kế và thiết lập các bộ điều khiển mờ.

1. Yếu tố thúc đẩy hệ điều khiển mờ

Lý thuyết điều khiển truyền thống dùng mô hình toán học của đối tượng điều khiển và các đặc tính vòng kín của ngõ ra để thiết kế bộ điều khiển. Vấn đề là tìm ra mô hình là rất khó khăn, đặc biệt khi hệ thống có một phần đặc tính ẩn hay có độ phi tuyến cao. Việc thiết kế bộ điều khiển cho các công việc hằng ngày như lái xe, hay cầm nắm một vật dễ vỡ tuy rất đơn giản với con người nhưng lại là vấn đề khó khăn cho một rôbôt. Trong khi con người chưa cần dùng đến mô hình toán học hay phải tìm ra quĩ đạo chính xác khi thực hiện các thao tác điều khiển này.

Nhiều quá trình do người điều khiển trong công nghiệp không thể được tự động hóa từ các kỹ thuật điều khiển truyền thống, do khả năng của các bộ điều khiển thường thấp hơn rất nhiều so với người vận hành. Một lý do nữa là các hệ thống tuyến tính thường được dùng trong hệ điều khiển truyền thống thì lại không thích hợp được với các bộ điều khiển phi tuyến. Hơn nữa, con người thường tích lủy nhiều dạng thông tin khác nhau rồi kết hợp trong chiến lược điều khiển, điều này lại không tích hợp được trong bộ điều khiển với luật điều khiển đơn nhất dạng giải tích. Ý tưởng về *điều khiển trên nền tri thức* là nhằm nắm bắt và thiết lập kinh nghiệm và kiến thức cần thiết từ chuyên gia (thí dụ người vận hành).

Một dạng đặc biệt của phương pháp điều khiển trên nền tri thức là hệ điều khiển dùng các luật mờ, theo đó tác động điều khiển tương ứng với các điều kiện đặc thù của hệ thống và được mô tả theo luật mờ nếu-thì (fuzzy if-then rules). Tập mờ dược dùng định nghĩa ý nghĩa của các giá trị định tính của ngõ vào và ngõ ra của bộ điều khiển như sai số *bé*, tác động điêu khiển *lớn*.

Ban đầu, điều khiển được dùng với mong muốn

- Nhằm bắt chước tác động điều khiển của người vận hành đầy kinh nghiệm (phần nền tri thức)
- Nhằm đạt được yếu tố nội suy mịn (smooth interpolation) giữa các ngõ ra rời rạc thực có (phần logic mờ)

Từ đó, ứng dụng của điều khiển mờ ngày càng rộng rãi. Tuy nhiên, có hai yếu tố thúc đẩy quan trọng sau. Bản chất ngôn ngữ của điều khiển mờ cho phép hệ thống diễn đạt được kiến thức của quá trình gồm cả phương thức điều khiển và cả phương thức đáp ứng của hệ thống. Phương thức nội suy của điều khiển mờ dẫn đến quan điểm là có thể xem hệ mờ là sơ đồ xấp xỉ mịn hàm.

Trong hầu hết các trường hợp thì bộ điều khiển mờ được dùng làm bộ điều khiển phản hồi trực tiếp. Tuy nhiên, hệ còn được dùng ở cấp độ điều khiển giám sát (supervisory level) như hệ thống tự-chỉnh (self-tuning device) trong các bộ điều khiển PID truyền thống. Ngoài ra, điều khiển mờ còn được dùng để diễn đạt trực tiếp các kiến thức đã có từ quá trình. Thí dụ, bộ điều khiển mờ có thể được dùng từ phương pháp nhận dạng đối tượng điều khiển. Như thế cần có một định nghĩa tổng quát về hệ điều khiển mờ.

Định nghĩa 6.1 (Điều khiển mờ) Bộ điều khiển mờ là bộ điều khiển có *chứa các ánh xạ (phi tuyến) được định nghĩa từ các luật mờ nếu-thì.*

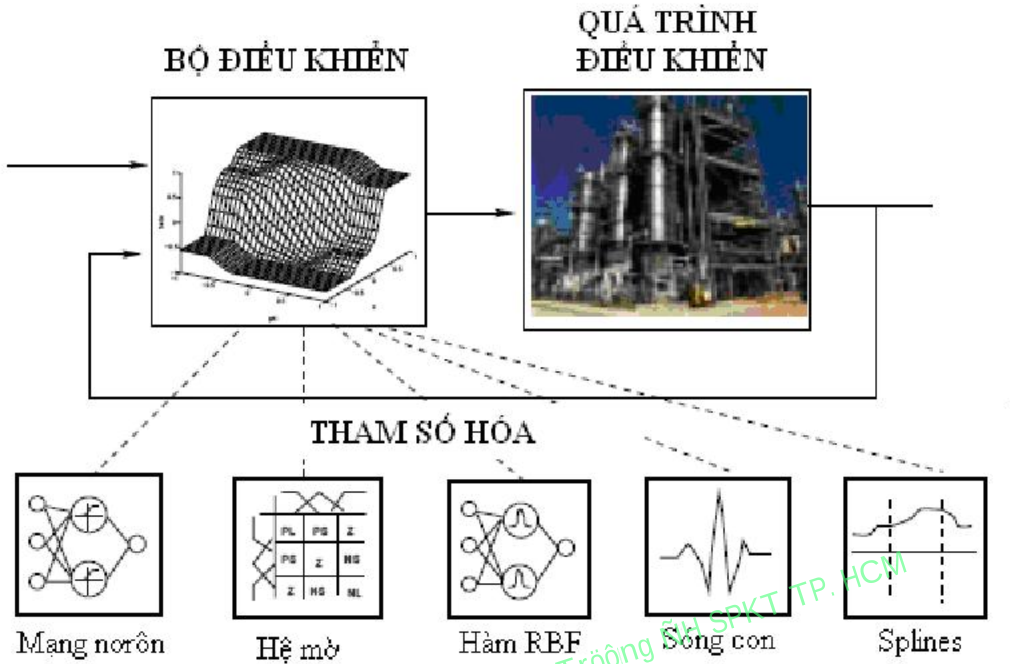
2. Điều khiển mờ để tham số hóa yếu tố phi tuyến của bộ điều khiển

Kết quả chủ yếu rút từ định nghĩa trên là là *ánh xạ phi tuyến* và *luật mờ nếu-thì*. Nhu cầu ngày càng lớn của công nghiệp về chất lượng và tính năng của hệ thống trong một tầm hoạt động rộng đã dẫn đến mối quan tâm càng cao về các phương pháp điều khiển phi tuyến trong những năm gần đây. Sự xuất hiện của các kỹ thuật “mới” như điều khiển mờ, mạng nơrôn, mạng sóng con (wavelets), và các hệ thống lai càng làm tăng thêm mối quan tâm này.

Hệ thống phi tuyến được xem xét thí dụ như khi quá trình được điều khiển là phi tuyến hay khi các đặc tính về tính năng là phi tuyến. Về cơ bản thì mọi quá trình trong thực tế đều là phi tuyến, từ yếu tố đặc tính động là phi tuyến hay do các yếu tố ràng buộc về trạng thái, các biến vào hay các biến khác. Có hai xu hướng cần theo là:

- *Thiết kế dùng mô hình phi tuyến.* Kỹ thuật phi tuyến có thể dùng trong mô hình hóa quá trình điều khiển. Mô hình tìm được có thể dùng làm cơ sở cho các thiết kế điều khiển dùng mô hình. Mô hình có thể dùng không trực tuyến khi thiết kế hay dùng trực tuyến như là một khâu của bộ điều khiển (xem chương 8).
- *Bộ điều khiển phi tuyến không dùng mô hình (model-free nonlinear contro)l.* Kỹ thuật phi tuyến còn được dùng để thiết kế trực tiếp bộ điều khiển, mà không cần có mô hình. Khâu phi tuyến có thể dùng trong vòng phản hồi hay đường truyền thẳng. Trong thực tế thì các khâu phi tuyến thường dùng kết hợp với các bộ lọc tuyến tính.

Có nhiều phương pháp khác nhau khi định nghĩa tính phi tuyến. Bao gồm các phương trình giải tích, mạng nơrôn dùng hàm sigmoid, splines, hàm radial basis functions, sóng con, các khâu mô hình/điều khiển tuyến tính hóa từng phần. Các phương pháp này biểu diễn nhiều phương thức khác nhau khi tham số hóa tính phi tuyến như trình bày ở hình 6.1.

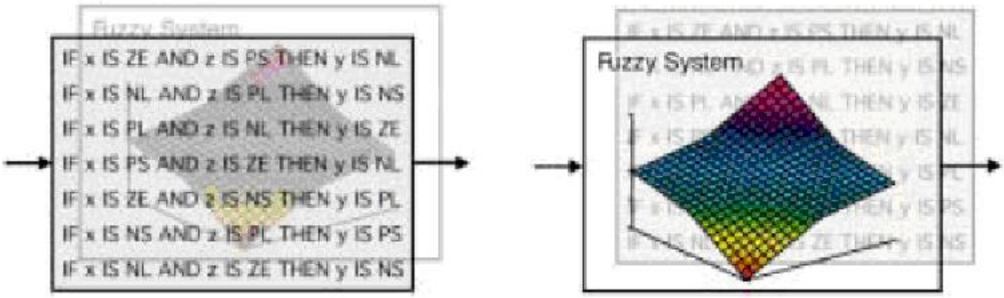


Hình 6.1 Các phương pháp tham số hóa hệ điều khiển phi tuyến

Các phương pháp này đã được chứng minh là những bộ xấp xỉ vạn năng (universal function approximators) cho một số dạng hàm. Điều này tức là chúng có thể xấp xỉ nhiều dạng hàm có tính phi tuyến. Tuy nhiên, chưa thể nói là phương pháp nào là tốt nhất nếu chỉ xem xét đến khả năng điều khiển vòng kín. Theo quan điểm của quá trình thì không quan tâm đến tính phi tuyến mà chỉ quan tâm đến phương thức tham số hóa tính phi tuyến này.

Tuy nhiên, bên cạnh khả năng xấp xỉ thì còn có nhiều yếu tố khác cũng cần được quan tâm. Một trong những yếu tố đó là tính hiệu quả (*efficiency)* của phép xấp xỉ theo tham số cần xấp xỉ của hàm đang khảo sát. Ngoài ra còn có yếu tố thực tiễn quan trọng là phương pháp là cục bộ hay toàn cục. Phương pháp cục bộ cho phép tinh chỉnh cục bộ. Thí dụ phương pháp hàm RBF (radial basis functions), splines, và hệ mờ. Các phương pháp này tốt khi hỗ trợ cho việc xử lý tính phi tuyến từ dữ liệu vào/ra, thí dụ nhận dạng/học/huấn luyện. Một yếu tố quan trọng nữa là tính sẵn sàng cho phương pháp phân tích và thiết kế, tức là khả năng diễn đạt các hiểu biết trước đó về hệ thống, hiệu quả của phương pháp tính toán, tính sẵn sàng của các công cụ máy tính, và cuối cùng là sự thoải mái của người thiết kế/người vận hành khi dùng phương pháp, với mức huấn luyện nào cần thiết khi dùng và hiểu về phương pháp.

Hệ logic mờ có vẽ là thỏa được hầu hết các tiêu chí này. Chúng là bộ xấp xỉ vạn năng, và với một số lựa chọn thiết kế, thì phép xấp xỉ là tương đối có hiệu quả. Tùy thuộc vào số lượng hàm liên thuộc (hàm thành viên) mà phương pháp sẽ là cục bộ hay toàn cục. Hệ mờ có các đặc tính xấp xỉ tương tự như mạng nơrôn dùng hàm sigmoid. Hệ logic mờ có thể là rất minh bạch (transparent) nên cho phép diễn tả tốt các kiến thức trước đó về quá trình. Một số công cụ máy tính đã sẳng sàng cho các thiết lập điều khiển mờ. Điều khiển mờ có thể được nhìn từ hai quan điểm. Thứ nhất chú ý đến luật nếu-thì mờ nên được dùng định nghĩa cục bộ ánh xạ phi tuyến và có thể được xem là phần giao diện với người dùng của hệ mờ. Quan điểm thứ hai gồm ánh xạ phi tuyến được tạo ra từ các luật và quá trình suy diễn. (xem hình 6.2).



Hình 6.2. Các quan điểm về hệ mờ. Các luật mờ (bên trái) được dùng làm giao diện với người dùng của hệ mờ. Chúng định nghĩa ánh xạ phi tuyến (bên phải) nhằm biểu diễn quan hệ vào-ra của hệ thống.

Các luật và cơ chế suy diễn tương ứng của bộ điều khiển mờ có thể có nhiều dạng như mô tả trong chương 3. Thường dùng nhất là:

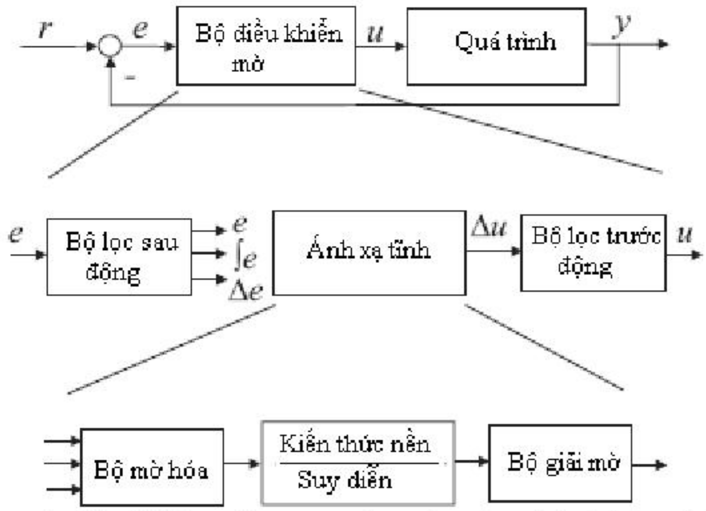
- Bộ điều khiển *Mamdani (ngôn ngữ)* có thể dùng các hệ quả mờ hay singleton. Dạng điều khiển này thường được dùng trong bộ điều khiển vòng kín dạng trực tiếp.

Phần tiếp theo mô tả hay dạng điều khiển này

3. Bộ điều khiển Mamdani

Bộ điều khiển Mamdani thường dùng trong điều khiển phản hồi, do luật nền biểu diễn ánh xạ tĩnh giữa tiền đề và hệ quả, nên cần có thêm bộ lọc động bên ngoài để có được đáp ứng ra động của bộ điều khiển (hình 6.3).

Giao thức điều khiển được lưu trữ dạng các luật nếu-thì là một phần của nền tri thức (*knowledge base*). Do các luật có cơ sở là kiến thức định tính, các hàm thành viên định nghĩa các thừa số ngôn ngữ cung cấp giao diện mịn cho các biến quá trình dạng số và các điểm đặt (set-points). Bộ mờ hóa (*fuzzifier*) xác định mức thành viên của giá trị biến vào bộ điều khiển trong các tập tiền đề mờ. Cơ chế suy diễn tổ hợp thông tin này với thông tin chứa trong các luật và xác định ngõ ra nào của luật sẽ được dùng. Thông thường thì ngõ ra này lại cũng là tập mờ. Để có thể điều khiển thì cần có tín hiệu điều khiển dạng rõ (crisp control signal). Bộ giải mờ (*defuzzifier*) tính toán giá trị trong tín hiệu thực từ các ngõ ra của bộ điều khiển mờ. Từ hình 6.3, ta thấy là ánh xạ mờ là một phần của bộ điều khiển mờ. Cần xử lý tín hiệu trước và sau khi thực hiện ánh xạ mờ.



Hình 6.3: Bộ điều khiển mờ trong cấu trúc vòng kín (phía trên), gồm bộ lọc động và ánh xạ tĩnh (phần giữa). Ánh xạ tĩnh được tạo nên từ nền tri thức, cơ chế suy diễn mờ và giao diện giải mờ

3.1 Bộ lọc động trước

Quá trình lọc trước (*pre-filter*) các tín hiệu vào của bộ điều khiển nhằm có được các ngõ vào của hệ mờ tĩnh. Thực hiện các tác động sau lên các ngõ vào:

Tỉ lệ tín hiệu (Signal Scaling**).** Thường nên làm việc với các tín hiệu trong vùng chuẩn hóa, thí dụ $[-1, 1]$. Điều này được thực hiện dùng độ lợi chuẩn hóa để chuẩn hóa tín hiệu vào trong tầm $[-1, 1]$.

Lọc động (Dynamic Filtering**).** Trong bộ điều khiển mờ PID, bộ lọc tuyến tính được dùng để tìm các giá trị đạo hàm, tích phân của sai số điều khiển *e*. Bộ lọc phi tuyến có trong bộ quan sát phi tuyến, và trong bộ điều khiển thích nghi mờ trong đó chúng được dùng để tìm các tham số ước lượng của hệ mờ.
Trích xuất tính năng (Feature Extraction**).** Qua việc trích nhiều tính năng khác nhau mà thực hiện phép biến đổi số lên các ngõ vào của bộ điều khiển. Các biến đổi này có thể là biến đổi Fourier hay biến đổi sóng con (wavelet transforms) chuyển đổi trục hay các phép toán cơ bản thực hiện trên các ngõ vào của bộ điều khiển mờ.

3.2 Bộ lọc động sau

Bộ lọc sau (*post-filter*) biểu diễn phép xử lý tín hiệu thực hiện tại ngõ ra của bộ điều khiển mờ để tạo tín hiệu điều khiển. Các hoạt động của bộ lọc sau thường là:

Tỉ lệ tín hiệu (Signal Scaling**).** Khôi phục lại tín hiệu đã chuẩn hóa tại ngõ ra của hệ mờ về vùng hoạt động thực của tín hiệu này.

Lọc động (Dynamic Filtering**).** Trong một số trường hợp, ngõ ra của hệ mờ là gia số của tác động điều khiển. Từ đó có được tín hiệu điều khiển khi tích phân gia số của điều khiển. Rõ ràng, còn có thể dùng những dạng khác như thiết bị mịn hay bộ lọc phi tuyến.

Việc phân tích bộ điều khiển thành ánh xạ tĩnh và bộ lọc động có thể có trong các cấu trúc điều khiển cổ điển. Thử xét trường hợp bộ PID (Proportional-Integral-Differential) được mô tả bởi phương trình:

$$u(t) = Pe(t) + I\int_0^t e(\tau)d\tau + D\frac{de(t)}{dt}, \qquad (6.1)$$

Trong đó $u(t)$ là tín hiệu điều khiển vào quá trình điều khiển và $e(t) = r(t) - y(t)$ là tín hiệu sai biệt giữa tín hiệu tham chiếu và tín hiệu ra đo tại quá trình. Việc thiết lập bộ điều khiển PID trên máy tính có thể biểu diễn dùng phương trình sai phân:

$$u_{\mathrm{PID}}[k] = u_{\mathrm{PID}}[k-1] + k_I e[k] + k_P \Delta e[k] + k_D \Delta^2 e[k] \qquad (6.2)$$

trong đó:

$$\Delta e[k] = e[k] - e[k-1], \qquad \Delta^2 e[k] = \Delta e[k] - \Delta e[k-1]$$

Độ lợi rời rạc theo thời gian k_P, k_I và k_D được rút ra từ phép rời rạc các độ lợi liên tục trong miền thời gian P, I và D. Phương trình (6.1) là hàm tuyến tính (geometrically a hyperplane):

$$u = \sum_{i=1}^{3} a_i x_i \qquad (6.4)$$

Trong đó $x_1 = e(t)$, $x_2 = \int_0^t e(\tau)d\tau$, $x_3 = \frac{de}{dt}$ và các tham số của a_i là các độ lợi P, I và D. Dạng tuyến tính (6.4) có thể được tổng quát hóa thành hàm phi tuyến:

$$\mathbf{u} = f(\mathbf{x}) \qquad (6.5)$$

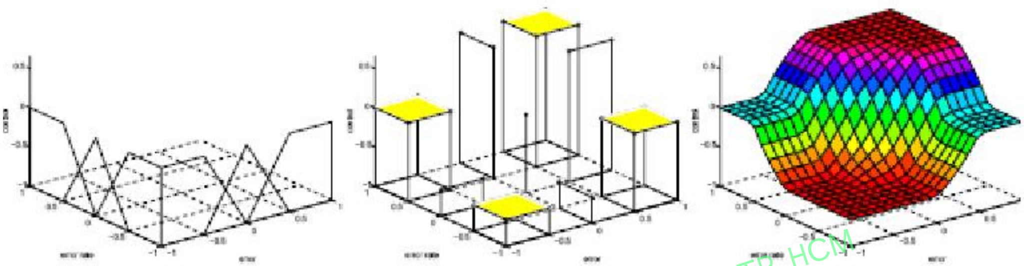
Trường hợp bộ điều khiển logic mờ, thì hàm phi tuyến f được biểu diễn bởi ánh xạ mờ. Rõ ràng hơn thì bộ điều khiển mờ tương đồng với các bộ điều khiển tuyến tính P, PI, PD hay PID có thể được thiết kế dùng các bộ lọc động thích hợp như bộ vi phân và tích phân.

3.3 Luật nền

Hệ mờ Mamdani thì rất gần với điều khiển tự nhiên hay điều khiển dùng nhân công. Bộ điều khiển được định nghĩa là chọn ngõ ra nào trong tổ hợp của nhiều tín hiệu vào. Mỗi tổ hợp tín hiệu vào được biểu diễn thành luật như sau:

R_i: **Nếu** x_1 là A_{i1} . . . **và** x_n là A_{in} **thì** u là B_i, $i = 1, 2, \ldots, K$. (6.6)

Ngoài ra còn có thể dùng các kết nối logic hay các toán tử khác, thí dụ *or* hay *not*. Trong hệ mờ Mamdani thì tập mờ tiền đề và hệ quả thường được chọn là tam giác hay dạng hàm Gauss. Thường các hàm thành viên tại ngõ vào chồng lắp (overlap) sao cho các giá trị thành viên của luật tiền đề luôn có tổng là một. Trong trường hợp này, và nếu luật nền có dạng conjunctive, thì có thể diễn dịch từng luật bằng cách định nghĩa giá trị ra tại một điểm trong không gian vào. Điểm không gian vào là điểm có được bằng cách lấy trọng tâm của tập mờ ngõ vào và giá trị ra là trọng tâm của tập mờ ngõ ra. Phép suy diễn mờ kết luận dùng phép nội suy mịn giữa các điểm trong không gian vào, xem hình 6.4.



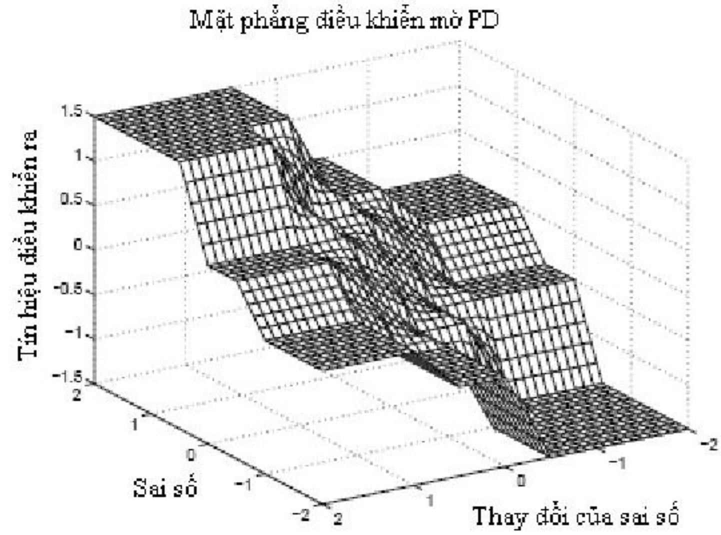
Hình 6.4: Bên trái. partition của các hàm thành viên trong không gian vào
Bên phải, phép nội suy logic mờ giữa các giá trị hằng

Từ cách diễn đạt này thì có thể xem hệ Mamdani là hàm hằng từng đoạn (piecewise constant function) với rất nhiều phép nội suy. Tùy phương pháp suy diễn mà có các phép nội suy khác nhau. Khi chọn đúng thì cũng có thể có phép nội suy tuyến tính hay nội suy đa tuyến tính (multilinear). Điều này thường thực hiện bằng cách thay các tập hệ quả mờ bằng singletons. Như thế thể phép suy diễn và phép giải mờ đưoc kết hợp lại thành một bước, xem phương trình (3.43), phần 3.3.

Thí dụ 6.1 (Điều khiển PD mờ) Xét hệ mờ đối ngẫu với bộ điều khiển PD tuyến tính (proportional-derivative) controller. Luật nền có hai ngõ vào– sai số e, và đạo hàm của sai số $\dot{e}$, và một ngõ ra – tác động điều khiển u. Một thí dụ về luật nền này là:

Năm thừa số ngôn ngữ được dùng cho mỗi biến (NB–*Negative big*, NS – *Negative small*, ZE – *Zero*, PS – *Positive small* và PB – *Positive big*). Mỗi mục của bảng định nghĩa một luật, thí dụ *R*23: "**Nếu** e là NS và $\dot{e}$ là ZE **thì** u là NS". Hình 6.5 minh họa mặt phẳng điều khiển kết quả có được từ phép vẽ các tác động điều khiển tìm được u với các giá trị rời rạc hóa của e và $\dot{e}$.

				$\dot{e}$		
		NB	NS	ZE	PS	PB
	NB	NB	NB	NS	NS	ZE
	NS	NB	NS	NS	ZE	PS
e	ZE	NS	NS	ZE	PS	PS
	PS	NS	ZE	PS	PS	PB
.	PB	ZE	PS	PS	PB	PB



Hình 6.5 Mặt phẳng điều khiển PD mờ

Trong điều khiển PD mờ, sai biệt giản đơn $\Delta e = e(k) - e(k-1)$ thường được dùng như phép xấp xỉ (xấu) cho phép đạo hàm.

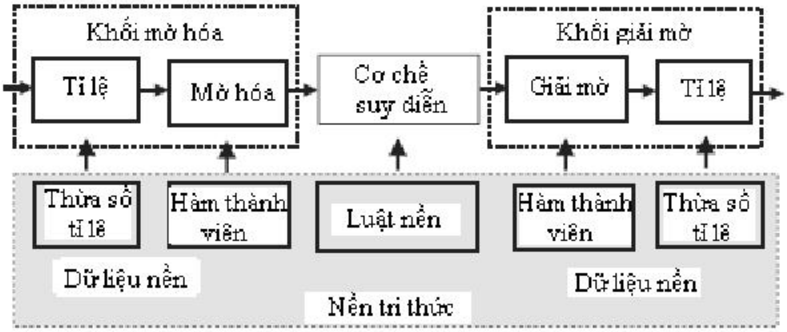
Xác định các ngõ vào và các ngõ ra. Trong bước này, cần có kiến thức cơ bản về đặc tính động của quá trình (ổn định, không ổn định, dừng, thay đổi theo thời gian, v.v,..), đặc tính về tính phi tuyến, mục tiêu điều khiển và các ràng buộc. Các đặc tính động của đối tượng cùng với mục tiêu điều khiển nhằm xác định các đặc tính động học của bộ điều khiển, thí dụ bộ điều khiển mờ dạng PI, PD hay PID.

Để bổ chính các tính phi tuyến, thay đổi theo thời gian, hay các hiện tượng không mong muốn của đối tượng điều khiển, cần có thêm các biến khác ngoài sai số. Các giá trị đạo hàm hay tích phân của sai số có thể được dùng làm ngõ vào cho bộ điều khiển. Thông thường thì đó có thể là ngõ ra, các trạng thái đo lường được hay cấu trúc lại được, các nhiễu đo được hay các biến ngoài khác. Tuy nhiên, điều cần thấy là khi tăng số lượng ngõ vào của bộ điều khiển mờ, thì mức độ phức tạp của bộ điều khiển mờ cũng gia tăng thêm nhiều lần.

Trong thực tế, cần nhìn nhận ảnh hưởng của các biến khác nhau và phân giải bộ điều khiển mờ với nhiều ngõ vào thành nhiều bộ điều khiển đơn giản có số ngõ vào ít hơn, hoạt động song song hay trong câu trúc phân cấp (hierarchical structure) (xem phần 3.2.7).

Điều quan trọng cần thực hiện là không như hệ điều khiển tuyến tính, có sự khác biệt giữa dạng tăng trưởng và dạng tuyêt đối của bộ điều khiển mờ. Dạng tuyệt đối của bộ điều khiển mờ, thực hiện ánh xạ $u = f(e,\dot{e})$, trong khi dạng tăng trưởng (incremental form) thì thực hiện ánh xạ $\dot{u} = f(\dot{e},\ddot{e})$. Trong dạng tăng trưởng, chiến lược điều khiển phi tuyến có thể thì lại liên quan đến tốc độ thay đổi của tác động điều khiển trong khi ở dạng tuyết đối thì chỉ phụ thuộc vào tự thân tác động điều khiển. Đây là mối liên quan trực tiếp với thiết kế dùng luật nền và đồng thời cũng là các đặc tính tổng quát của bộ điều khiển. Thí dụ, ngõ ra của bộ điều khiển mờ trong dạng tuyệt đối thì bị giới hạn từ định nghĩa, điều này là không đúng với dạng tăng trưởng.

Một vấn đề khác cũng cần được xem xét là bộ điều khiển mờ sẽ là bộ điều khiển tự động đầu tiên cho ứng dụng, hay nó sẽ bổ sung hay thay thế bộ điều khiển hiện hữu. Nếu thay thế, thì việc chọn lựa cấu trúc bộ điều khiển mờ tùy thuộc vào cấu hình của bộ điều khiển hiện tại.Tóm lại, cần nhấn mạnh rằng đây là bước quan trọng, do nếu chọn lựa không phù hợp cấu trúc thì có thể làm xáo trộn toàn bộ thiết kế, bất chấp các luật và các hàm thành viên.



Hình 6.6 Các môđun khác nhau trong bộ điều khiển mờ và các khâu tương ứng trong nền tri thức

Định nghĩa hàm thành viên và các thừa số tỉ lệ. Như vẽ trong hình 6.6, các thừa số ngôn ngữ, các hàm thành viên của chúng, và các thừa số tỉ lệ miền là một phần của bộ điều khiển mờ dùng nền tri thức.

Đầu tiên, người thiết kế phải quyết định chọn bao nhiêu thừa số ngôn ngữ cho từng biến vào. Để có thể duy trì được các luật vào, số lượng các thừa số cho mỗi biến nên bé. Ngoài ra, khi có ít thừa số, thì tính mềm dẽo của của luật nền bị giới hạn do phần phi tuyến giải quyết được trong ánh xạ điều khiển.

Số lượng các thừa số này cần được chọn lựa cẩn thận, xem xét nhiều thiết lập khác nhau cho các biến khác nhau tùy theo ảnh hưởng mong muốn của chúng lên chiến lược điều khiển. Tốt nhất thì nên bắt đầu với số ít thừa số (thí dụ 2 hay 3 cho các ngõ vào và 5 cho các ngõ ra) rồi tăng dần số này khi cần. Thừa số ngôn ngữ thường có một số ý nghĩa, thí dụ, chúng diễn tả biên độ của một số biến vật lý, thí dụ *Small*, *Medium*, *Large*, v.v,... Trong các miền đối xứng xung quanh zero, biên độ này được kết hợp với dấu, thí dụ. *Positive small* or *Negative medium*.

Các hàm thành viên có thể là một phần của kiến thức từ chuyên gia, tức là chuyên gia biết một cách xấp xỉ về ý nghĩa “nhiệt độ cao” (trong từng ứng dụng cụ thể). Khi thiếu các kiến thức này, thì có thể dùng hàm thành viên với cùng dạng (shape) phân phối đồng đều. làm thiết lập ban đầu và sẻ được tinh chỉnh sau. Để tính toán thích hợp, nên dùng hàm thành viên dạng tam giác, và hình thang thay vì dùng các hàm có dạng hình chuông.

Thông thường thì các biến vào và ra được định nghĩa trong một tầm giới hạn của đường thực. Để đơn giản thiết kế điều khiển, thiết lập và tinh chỉnh, nên hoạt động trong vùng chuẩn hóa, thí dụ $[-1, 1]$. Các thừa số tỉ lệ thường được dùng để chuyển đổi các giá trị từ tầm hoạt động sang vùng chuẩn hóa. Các thừa số tỉ lệ có thể được dùng để tinh chỉnh các độ lợi của bộ điều khiển mờ, tương tự như trong trường hợp hệ PID.

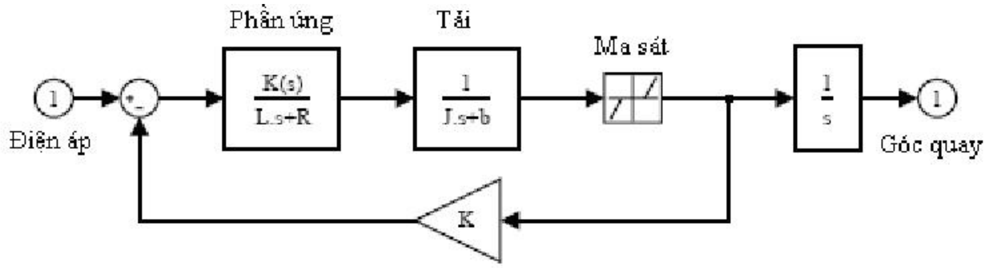
Thiết kế luật nền. Cấu trúc luật nền là yếu tố then chốt trong thiết kế, do luật nền mã hóa giao thức điều khiển của bộ điều khiển mờ. Cần phân biệt về nhiều phương pháp thiết kế luật nền. Một số dựa hoàn toàn vào kiến thức trực giác từ người vận hành (operators) thành dạng thích hợp để cấu trúc nên luật nền, phương pháp này thường kết hợp với các nguyên lý của lý thuyết điều khiển và hiểu biết tốt về đặc tính động của hệ thống. Xu hướng khác là dùng mô hình mờ của quá trình từ đó tìm ra luật nền điều khiển. Thường thì có dạng luật nền “chuẩn” được dùng bảng biểu mẫu (template). Dạng luật nền này bắt chước hoạt động của bộ điều khiển tuyến tính có dạng thích hợp (thí dụ bộ điều khiển PD sẽ có dạng như trong thí dụ 6.1. Chú ý là luật nền thì đối xứng chung quanh đường chéo và tương ứng với dạng tuyến tính $u = Pe + D\dot{e}$. Các độ lợi P và D có thể được định nghĩa từ việc lựa chọn thích hợp các thừa số tỉ lệ.

Tinh chỉnh bộ điều khiển. Việc tinh chỉnh bộ điều khiển mờ thường được so sánh với việc tinh chỉnh bộ PID, nhấn mạnh đến số rất lớn các tham số điều khiển mờ, so sánh với 3 độ lợi trong bộ PID. Có hai chú ý thích hợp ở đây. Đầu tiên, bộ điều khiển mờ có dạng điều khiển tổng quát so với PID, có khả năng điều khiển hệ thống phi tuyến mà bộ điều khiển tuyến tính không thể thực hiện một cách trực tiếp, hay là cải thiện điều khiển cho (hầu hết) các hệ thống tuyến tính khi vượt khỏi khả năng điều khiển của các bộ điều khiển tuyến tính. Từ đó, điều phải trả giá là việc định nghĩa và tinh chỉnh càng nhiều tham số điều khiển. Thứ hai, trong trường hợp các hệ thống phức tạp, có yếu tố ghép nối quan trọng về ảnh hưởng của ba độ lợi bộ PID, và như thế thì việc tinh chỉnh bộ PID se có thể là một công việc rất phức tạp. Mặt khác, trong điều khiển mờ, các luật và hàm thành viên có những ảnh hưởng cục bộ và là yếu tố lợi điểm trong đều khiển các hệ thống phi tuyến. Thí dụ, các luật điều khiển không đối xứng có thể được thiết kế cho các hệ thống phải chịu yếu tố động không đối xứng, thí dụ như trong các hệ thống nhiệt.

Mục tiêu ảnh hưởng lên các tham số riêng lẽ của hệ mờ thì khác hẳn. Các thừa số tỉ lệ, xác định độ lợi chung của bộ điều khiển mờ và cùng các độ lợi tương đối của các ngõ vào điều khiển riêng biệt, thì có ảnh hưởng toàn cục lớn nhất.Chú ý là việc thay đổi thừa số tỉ lệ cũng tỉ lệ được tính phi tuyến được các luật nền định nghĩa, là điều không mong muốn. Ảnh hưởng lên hàm thành viên thì còn cục bộ hơn nữa. Việc thay đổi hàm thành viên, gọi là *Small*, cho một biến cụ thể, thì chỉ ảnh hưởng lên các luật này, khi đã dùng thừa số này. Cục bộ lớn nhất là ảnh hưởng lên các hệ quả của từng luật riêng biệt. Việc thay đổi luật hệ quả chỉ ảnh hưởng lên vùng xác định của luật hệ quả.

Như đã biết thì hệ thống suy diễn mờ là bộ xấp xỉ hàm tổng quát, tức là có thể xấp xỉ bất kỳ hàm mịn với độ chính xác mong muốn. Điều này có nghĩa là bộ điều khiển tuyến tính có thể xem là một trường hợp đặc biệt của bộ điều khiển mờ, khi xem xét theo quan điểm chức năng vào-ra. Như thế, bộ điều khiển mờ có thể được khởi tạo dùng các luật điều khiển tuyến tính, có thể được xem là bước tinh chỉnh đơn giản ban đầu, mà bảo đãm được tính năng “tối thiểu ” của bộ điều khiển mờ. Luật nền hay hàm thành viên từ đó có thể được thay đổi nhằm cải thiện tính năng của hệ thống hay giảm bớt ảnh hưởng của một số hiện tượng cục bộ không mong muốn, thí dụ như ma sát v.v,…Xem thí dụ minh họa dưới đây

Thí dụ 6.2 (Bổ chính ma sát dùng bộ điều khiển mờ**)** Trong thí dụ này ta sẽ phát triển bộ điều khiển mờ dùng mô phỏng một động cơ DC bao gồm mô hình đơn giản hóa của ma sát tĩnh. Thí dụ này được thiết lập trong file MATLAB/Simulink (fricdemo.m). Hình 6.7 vẽ sơ đồ khối của động cơ DC.



Hình 6.7 Động cơ DC có khâu ma sát

Đầu tiên, bộ điều khiển tỉ lệ tuyến tính được thiết kế dùng phương pháp chuẩn (thí dụ phương pháp qui đạo nghiệm). Tiếp đến, bộ điều khiển tỉ lệ mờ được phát triển nhằm bắt chước chính xác bộ điều khiển tuyến tính. Hai bộ điều khiển này có đáp ứng y hệt nhau và đều chịu sai số xác lập do ma sát. Các luật đặc biệt được cộng vào luật nền nhằm giảm sai số này. Bộ điều khiển tuyến tính và bộ điều khiển mờ được so sánh dùng sơ đồ khối trong hình 6.8.

Các luật điều khiển mờ nhằm bắt chước bộ điều khiển tuyến tính là:
Nếu sai số là Zero

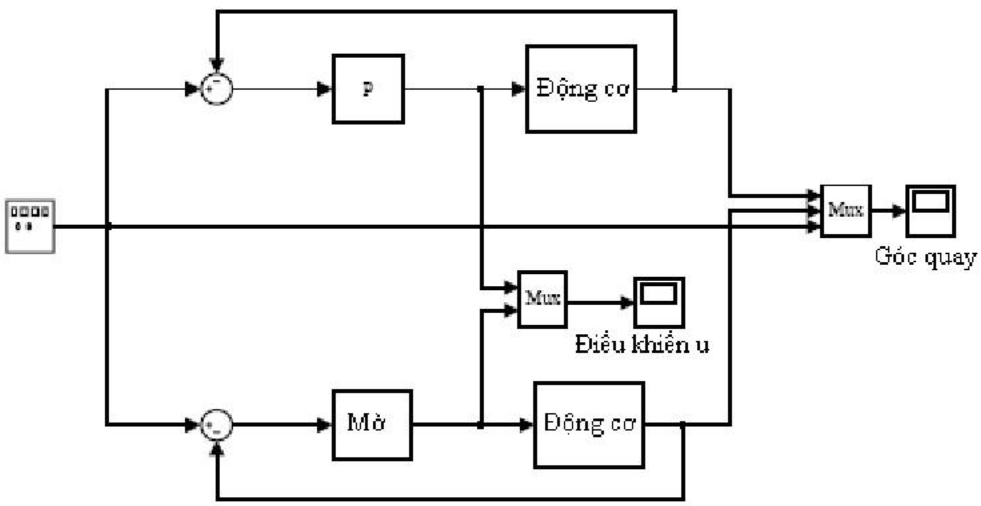
Thì điều khiển vào là Zero;

Nếu sai số là Positive Big

Thì điều khiển vào là Positive Big;

Nếu sai số là Negative Big

Thì điều khiển vào là Negative Big;



Hình 6.8 Sơ đồ khối nhằm so sánh giữa điều khiển tuyến tính và điều khiển mờ

Kết quả điều khiển được vẽ ở hình 6.9.

Hai luật phụ thêm vào nhằm ngăn ngừa bộ điều khiển tạo ra tác động điều khiển *small* bao giờ tác động điều khiển là *small*. Các tác động điều khiển này rõ ràng không có ảnh hưởng lên động cơ, do chúng không vượt qua được ma sát

.

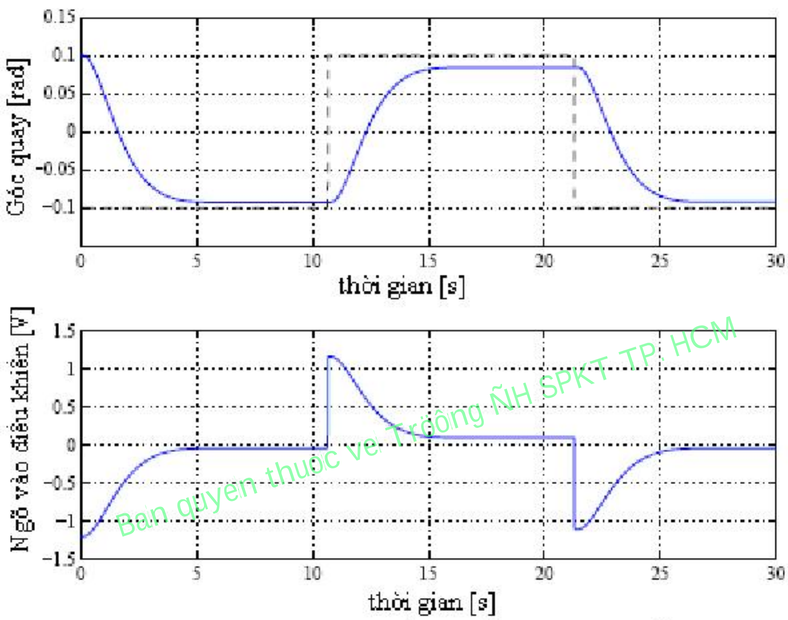
Nếu sai số là Negative Small

Thì điều khiển vào là NOT Negative Small;

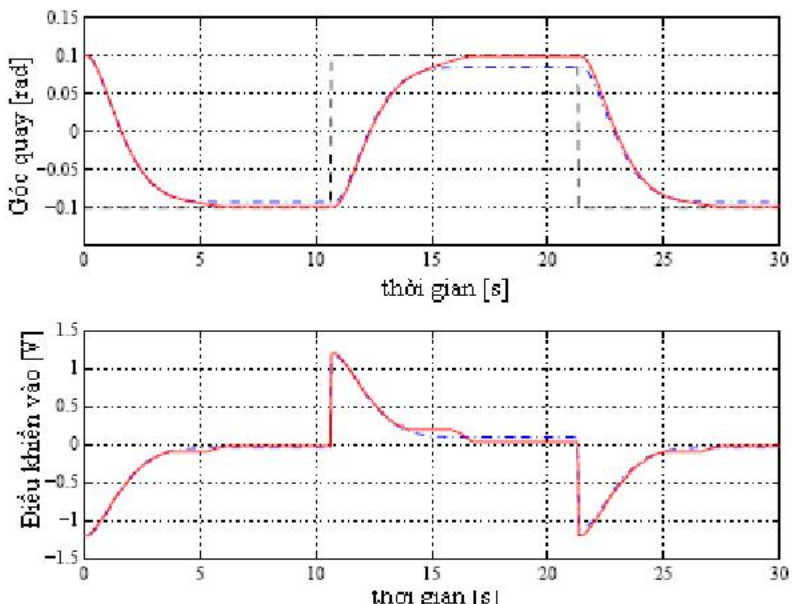
Nếu sai số là Positive Small

Thì điều khiển vào là NOT Positive Small;

Hàm thành viên của hai thừa số ngôn ngữ “Negative Small” và “Positive Small” đã tìm ra từ kế tquả ở hình 6.9. Łukasiewicz implication được dùng nhằm xử lý toán tử *not* (xem thêm thí dụ 3.7). Kết quả điều khiển có được vẽ ở hình 6.10. Chú ý là sai số xác lập hầu như bị loại bỏ.



Hình 6.9 Đáp ứng của bộ điều khiển khi thay đổi theo tín hiệu dạng bước của góc mong muốn

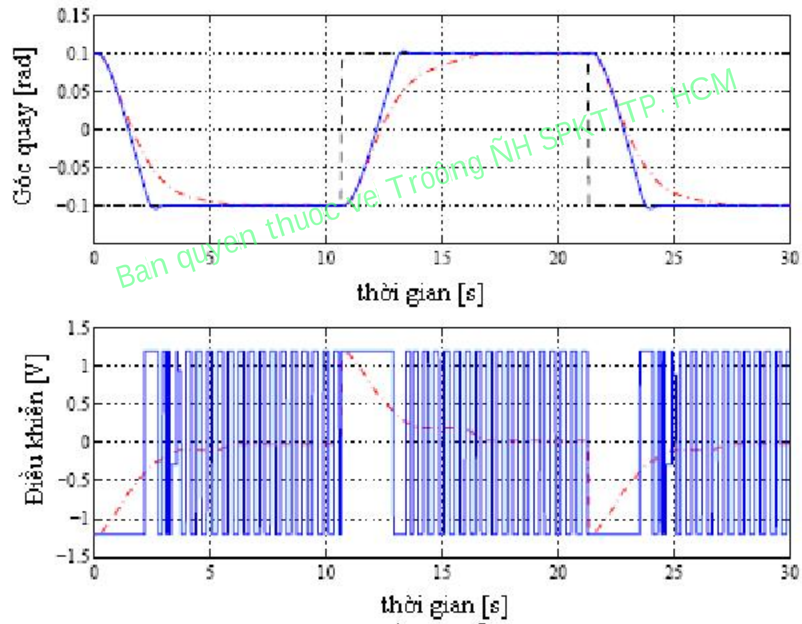


Hình 6.10 So sánh giữa bộ điều khiển tuyến tính (đường đứt nét) và điều khiển mờ (đường sậm)

Một phương pháp dùng điều khiển mờ khác cho ma sát là điều khiển PI và điều khiển theo chế độ trượt (sliding mode control). Tác động tích phân của bộ điều khiển PI sẽ tạo ra dao động trong vòng và làm hỏng hiệu năng điều khiển. Lý do là yếu tố phi tuyến từ khâu ma sát tạo khâu không liên tục trong vòng. Chế độ trượt là bền vững đối với tính phi tuyến của quá trình, và tác động nhanh hơn hệ điều khiển mờ, nhưng cần có tác động điều khiển mãnh liệt (violent control actions) (xem hình 6.11).

4. Bộ điều khiển Takagi-Sugeno

Bộ điều khiển mờ Takagi–Sugeno (TS) rất gần với hướng điều khiển chỉnh định độ lợi (gain scheduling). Nhiều bộ điều khiển tuyến tính đã định nghĩa là mỗi bộ điều khiển thì phù hợp với một vùng không gian ngõ vào khác nhau. Ngõ ra tổng của bộ điều khiển có được từ cách lựa chọn một bộ điều khiển dựa trên giá trị các ngõ vào (phương pháp chỉnh định độ lợi truyền thống), hay dùng phương pháp nội suy giữa nhiều bộ điều khiển tuyến tính (chỉnh định độ lợi mờ, điều khiển TS), xem hình 6.12.



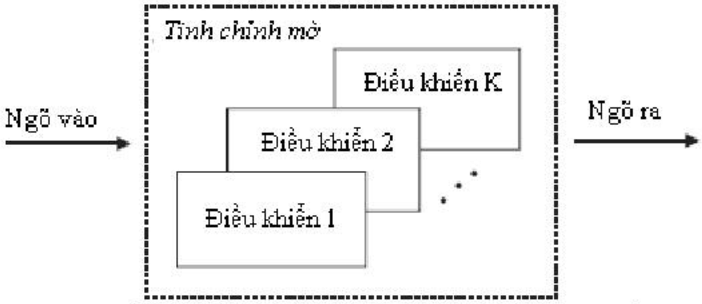
Hình 6.11 So sánh giữa điều khiển mờ (đường đứt nét) và điều khiển theo chế độ trượt (đường sậm)

Khi dùng hệ mờ TS thì thường dùng các tập mờ ngõ vào có dạng tam giác (trapezoidal). Mỗi tập mờ xác định không gian vào, trong đó ở trường hợp tuyến tính thì ngõ ra được xác định bằng hàm tuyến tính theo ngõ vào. Logic mờ chỉ được dùng để nội suy trong trường hợp các vùng trong không gian vào bị trùng lắp (overlap). Hệ TS lúc này có thể xem là hàm affine tuyến tính hóa từng đoạn với phép nội suy giới hạn.

Một thí dụ về luật điều khiển TS là:

$R1$: **Nếu** r là *Low* **thì** $u_1 = P_{\text{Low}}e + D_{\text{Low}}\dot{e}$

$R2$: **Nếu** r là *High* **thì** $u_2 = P_{\text{High}}e + D_{\text{High}}\dot{e}$ (6.7)



Hình 6.12 Bộ điều khiển TS có thể xem là tập của nhiều bộ điều khiển cục bộ được tổ hợp dùng cơ chế scheduling mờ.

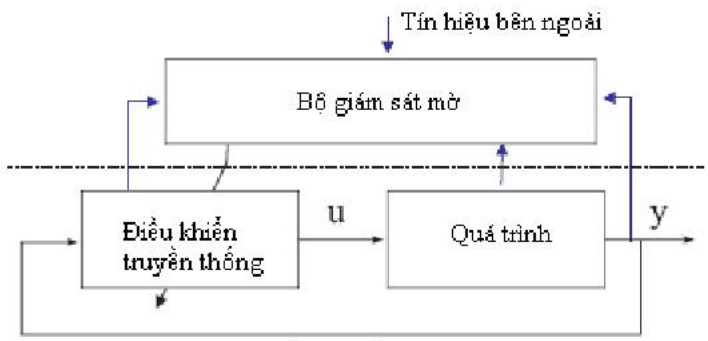
Ở đây cần chú ý là biến tiền đề là tín hiệu tham chiếu r trong khi các biến hệ quả là sai số e và đạo hàm $\dot{e}$ của sai số. Như thế bộ điều khiển là tuyến tính theo e và $\dot{e}$, nhưng các tham số của ánh xạ tuyến tính thì tùy thuộc vào tham chiếu:

$$
\begin{aligned}
u &= \frac{\mu_{Low}(r)u_1 + \mu_{High}(r)u_2}{\mu_{Low}(r) + \mu_{High}(r)} \\
&= \frac{\mu_{Low}(r)(P_{Low}e + D_{Low}\dot{e}) + \mu_{High}(r)(P_{High}e + D_{High}\dot{e})}{\mu_{Low}(r) + \mu_{High}(r)}
\end{aligned}
\qquad (6.9)
$$

.Nếu các bộ điều khiển cục bộ chỉ khác nhau về tham số. thì bộ điều khiển TS có dạng luật nền của cơ chế gain-scheduling. Mặt khác, bộ điều khiển hỗn tạp (*heterogeneous control*) (Kuipers and Astr¨om, 1994) có thể dùng các luật điều khiển khác nhau trong các vùng hoạt động khác nhau. Trong trường hợp sau, thí dụ bộ điều khiển tối ưu về thời gian cho quá trình chuyển giai đoạn động có thể được kết hợp với bộ điều khiển PI(D) trong vùng lân cận của các điểm thiết lập (setpoints). Như thế, bộ điều khiển TS có thể được xem là dạng đơn giản của bộ điều khiển giám sát.

5. Bộ điều khiển giám sát mờ

Hệ thống suy diễn mờ còn có thể được dùng trong cấp độ cao hơn, cấp độ giám sát trong phương pháp điều khiển phân cấp (control hierarchy). Bộ điều khiển giám sát là bộ điều khiển thứ cấp nhằm giúp bộ điều khiển hiện hữu đạt mục tiêu điều khiển, mà bộ điều khiển hiện hữu này không thể thực hiện được nếu không có giám sát. Bộ điều khiển giám sát tinh chỉnh tham số của bộ điều khiển cấp thấp hơn tùy theo thông tin có từ quá trình (Hình 6.13).



Hình 6.13 Bộ điều khiển giám sát mờ

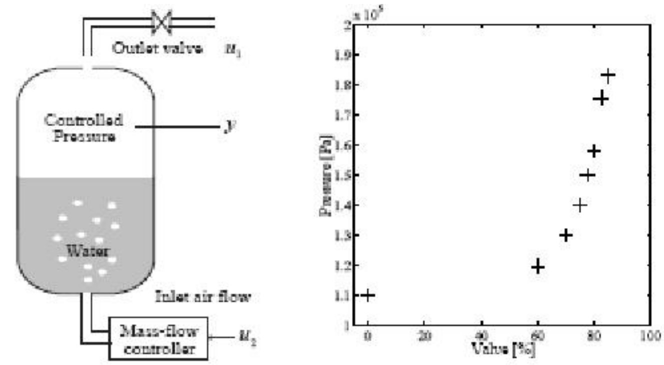
Trong phương pháp này thì các đặc tính ngõ ra động hay tĩnh của bộ điều khiển cấp thấp hơn có thể được thay đổi nhằm khớp với tính phi tuyến của quá trình hay sự thay đổi của điểm làm việc hay môi trường. Một ưu điểm của cấu trúc giám sát là có thể cộng thêm vào các hệ thống điều khiển hiện có. Như thế, bộ điều khiển nguyên thủy có thể được dùng như bộ điều khiển ban đầu (initial controllers), từ đó bộ điều khiển giám sát có thể tinh chỉnh để cải thiện hiệu năng của hệ thống. Kiến trúc giám sát có thê dùng cho nhiều chiến lược điều khiển khác nhau trong một bộ điều khiển đơn. Thí dụ như trong bộ điều khiển tỉ lệ, thì khi hệ thống còn chưa khớp được với tín hiệu tham chiếu và cần chuyển sang điều khiển dạng điều khiển PI ở lân cận tín hiệu tham chiếu. Do các tham số thay đổi trong đáp ứng động, nên các bộ điều khiển giám sát thường là điều khiển phi tuyến.

Nhiều quá trình trong công nghiệp dùng các bộ điều khiển PID. Bên cạnh các ưu điểm thì các bộ điều khiển PID truyền thống phải chịu thực tế là phải được chỉnh định lại khi điều kiện làm việc thay đổi. Yếu điểm này có thể được giải quyết dùng bộ giám sát mờ để tinh chỉnh lại các tham số của bộ điều khiển cấp thấp. Dùng chuyên gia để xác định tập các luật điều chỉnh độ lợi *P* và *D* của bộ điều khiển PD, thí dụ dùng điểm thiết lập hiện hữu *r*. Các luật này có dạng sau:

Nếu ngõ ra của quá trình là *High*
thì giảm độlợi bộ tỉ lệ *Slightly* **và**
tăng độ lợi bộ vi phân *Moderately*.

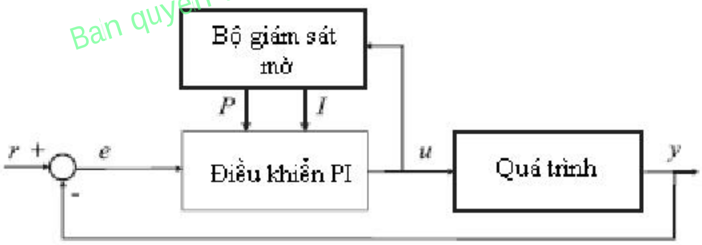
Bộ điều khiển TS có thể đươc diễn đạt như một phiên bản đơn giản của điều khiển giám sát. Thí dụ các luật TS (6.7) có thể được viết theo các luật Mamdani hay singleton có ngõ ra là các tham số *P* và *D*. Từ đó, đã chuyển dạng điều khiển PD truyền thống thành dạng điều khiển ở cấp thấp hơn.

Thí dụ 6.3: Bộ điều khiển giám sát mờ được dùng trong điều khiển áp suất của bộ lên men trong phòng thí nghiệm, như hình 6.14



Hình 6.14 Trái: thiết lập thử nghiệm; phải: đặc tính xác lập phi tuyến

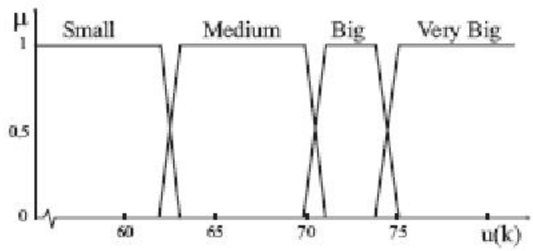
Thể tích của bồn lên men là 40 l, và thường được đổ vào 25 l nước. Phần trên của bồn là không khí được đưa vào nước với lưu lượng không đổi, thông qua bộ điều khiển lưu lượng tại chỗ. Không khí nén phía trên mức nước được khống chế bằng vòi ra đặt tại phần trên bồn. Khi lưu tốc vào không đổi, hệ thống có một ngõ vào là vi trí của van và một ngõ ra là áp suất không khí. Do cơ chế vật lý này, và từ đặc tính phi tuyến của van điều khiển, quá trình có đặc tính xác lập phi tuyến, như vẽ ở hình 6.14, cùng với đặc tính phi tuyến động tại ngõ ra.



Hình 6.15: Sơ đồ bộ điều khiển giám sát mờ

Một bộ giám sát gồm một ngõ vào, hai ngõ ra giám sát được vẽ ở hình 6.15 was designed. Ngõ vào của bộ giám sát là vị trí van $u(k)$ và các ngõ ra là độ lợi bộ tỉ lệ và độ lợp bộ tích phân của bộ điều khiển PI truyền thống. Bộ giám sát cập nhật giá trị độ lợi PI tại mỗi bước lấy mẫu của vòng điều khiển của bộ điều khiển cấp thấp (5 s).

Miền của vị trí van (0–100%) được chia ra thành bốn tập mờ ('Small', 'Medium', 'Big' vá 'Very Big'), xem thêm các hàm thành viên trong hình 6.16.



Hình 6.16 Các hàm thành viên u(k)

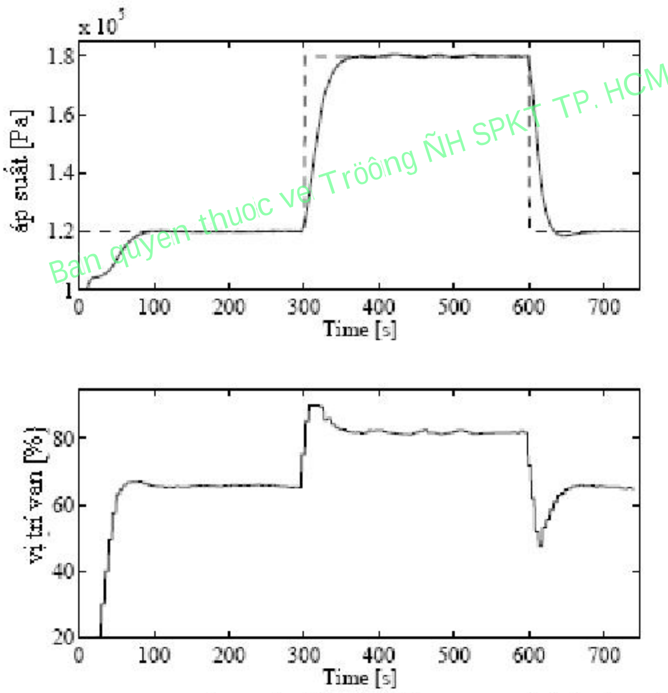
Độ lợi của bộ PI là *P* và *I* được cho trong bảng sau:

Gains \ $u(k)$	Small	Medium	Big	Very big
P	190	170	155	140
I	150	90	70	50

Các giá trị của P và I tìm được qua mô phỏng lần lượt tại các vùng của vị trí van. Ngõ ra chung lấy từ mô phỏng được tính toán như trung bình trọng lượng của các độ lợi cục bộ.

Bộ điều khiển giám sát mờ, được thử nghiệm và tinh chỉnh qua phép mô phỏng, được áp dụng trực tiếp vào quá trình (mà không cần phải chỉnh định lại), trong điều kiện danh định.

Các kết quả điều khiển trong thời gian thực được vẽ ở hình 6.17.



Hình 6.17: Kết quả điều khiển trong thời gian thực của bộ điều khiển giám sát mờ

6. Hỗ trợ từ người vận hành

Cho dù lý thuyết điều khiển đã có nhiều tiến bộ đi nữa, mức độ tự động hóa trong một số công nghiệp (như hóa chất, sinh hóa hay công nghiệp thực phẩm) vẫn còn rất chậm. Dù đã thiết lập những vòng điều khiển tự động cơ bản, người vận hành luôn cần có để giám sát và phối hợp các chức năng điều khiển này, thiết lập hay tinh chỉnh các tham số và đồng thời điều khiển thủ công quá trình kh mới khởi động, dừng máy hay trong các giai đoạn chuyển tiếp. Các dạng chiến lược điều khiển không thể được biểu diễn dùng dạng giải tích hay dễ dàng dùng các luật nếu-thì. Khi có kinh nghiệm từ người vận hành, các bộ điều khiển mờ có thể được dùng như phần hỗ trợ quyết định, nhằm giảm thiểu sự lệ thuộc vào kính nghiệm của người vận hành (dùng khả năng biểu diễn minh bạch kiến thức của bộ điều khiển mờ). Từ đó, giảm thiểu sự khác biệt khi dùng nhiều người vận hành khác nhau, cho phép giảm thiểu chi phí về năng lượng và vật tư, v.v,… Hệ mờ có thể làm đơn giản nhiệm vụ của người vận hành bằng cách rút ra các thông tin có giá trị từ khối lượng lớn các đo lường và dữ liệu. Cần có một giao diện thích hợp với người dùng để thông tin với người vận hành. Việc dùng các biến ngôn ngữ và khả năng diễn đạt các biến này có thể cải thiện giao diện người-máy.

7. Các công cụ phần cứng và phần mềm

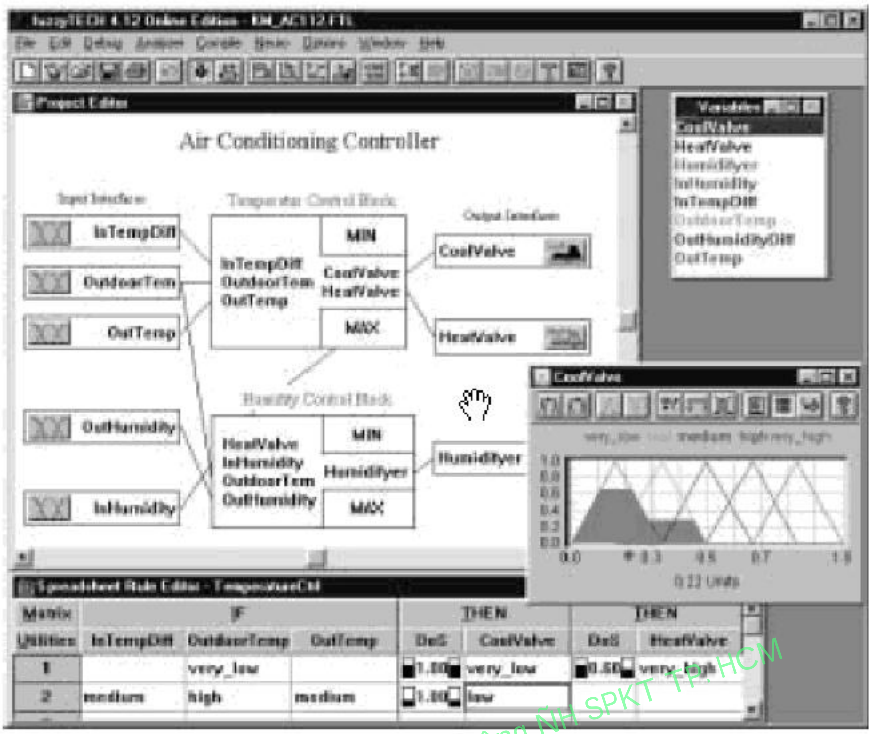
Sự phát triển các bộ điều khiển mờ tất cần có sự tương tác với người dùng, nên nhiều công cụ phần mềm đặc biệt đã được nhiều nhà cung cấp phần cứng và phần mềm giới thiệu như Omron, Siemens, Aptronix, Inform, National Semiconductors, v.v,... Hầu hết các chương trình chạy trên máy tính, trong môi trường Windows, một số còn có thể vận hành cả trong hệ UNIX. Tham khảo thêm http://www.isis.ecs.soton.ac.uk/resources/nfinfo/ . Hê mờ đã dần trở thành chuẩn cho các hệ thống điều khiển tại các nhà máy, thí dụ như hệ của Honeywell. Hầu hết các công cụ phần mềm thường gồm các khối sau.

7.1 Bộ soạn thảo dự án

Tâm điểm của giao diện với người dùng có bộ soạn thảo dự án dạng đồ họa, cho phép người dùng xây dựng hệ điều khiển mờ từ các khối cơ bản. Các biến vào và biến ra có thể được định nghĩa và kết nối với đơn vị suy diễn mờ một các trực tiếp hay thông qua các khâu xử lý trước (pre-processing) hay xử lý sau (post-processing) như các khâu lọc động (dynamic filters) bộ tích phân, bộ vi phân, v.v,…Nhiệm vụ các khâu này được người dùng định nghĩa, dùng ngôn ngữ C hay các biến thể của ngôn ngữ này. Có thể tổ hợp nhiều đơn vị suy diễn mờ để tạo ra các sơ đồ điều khiển mờ phức tạp hơn (thí dụ dạng phân cấp hay dạng phân bố).

7.2 Luật nền và các hàm thành viên

Luật nền và các hàm thành viên (các tập mờ có liên quan) được định nghĩa dùng luật nền và bộ soạn thảo hàm thành viên (membership function editors). Bộ soạn thảo luật nền (*rule base editor*) là bảng hay bảng tính có thể nhập hay thay đổi các luật. Bộ biên soạn hàm thành viên (*membership functions editor*) là môi trường đồ họa (graphical environment) dùng định nghĩa dạng và vị trí của các hàm thành viên. Hình 6.18 cho thí dụ về nhiều dạng màn hình giao diện của FuzzyTech.



Hình 6.18 Màn hình giao diện của Fuzzy Tech

7.3 Các công cụ phân tích và mô phỏng

Sau khi đã thiết kế xong các luật và hàm thành viên, có thể thử nghiệm các chức năng của bộ điều khiển mờ dùng công cụ *phân tích tĩnh* và phương pháp *mô phỏng động*. Các giá trị vào được nhập từ bàn phím hay dạng thực nhằm kiểm tra bộ điều khiển có tạo được ngõ ra như mong muốn không. Mức độ hoàn thành của từng luật, việc tinh chỉnh tập các ngõ ra mờ, kết quả của việc kết hợp các luật, quá trình giải mờ có thể được hiển thị thành từng dòng hay thành các tệp tin. Đối với từng cặp vào/ra thì có thể khảo sát mặt phẳng điều khiển với hai hay ba chiều. Một số gói (packages) còn cung cấp phương tiện kiểm tra tự động về tính trọn vẹn (completeness) và tính dư thừa (redundancy) của các luật trong luật nền. Đặc tính động của hệ vòng kín còn có thể phân tích dùng phép mô phỏng trực tiếp trong môi trường thiết kế hay thông qua việc tạo mã trong các chương trình điều khiển mô phỏng (thí dụ Simulink).

7.4 Bộ tạo mã nguồn (code) và kết nối thông tin

Sau khi đã kiểm tra bộ điều khiển mờ dùng các công cụ phần mềm phân tích, có thể dùng điều khiển trực tiếp tại hiện trường (dùng cổng ra của máy tính hay các ngõ vào/ra dạng analog), hay dùng các mã chương trình. Các bộ tạo chương trình quan trọng nhất là dùng C-code đồng thời cũng là mã máy cho một số phần cứng, thí dụ như vi điều khiển hay điều khiển lập trình (PLCs). Từ phương tiện này, các phần cứng hiện hữu đều có thể dùng cho điều khiển mờ. Bên cạnh đó, hiện đã có các phần cứng dạng mờ trên thị trường, thí dụ như các chip dùng cho điều khiển mờ (có cả analog và số, xem hình 6.19) hay dùng các bộ đồng xử lý mờ (fuzzy coprocessors) dùng cho PLC.

Hình 6.19 Chip dùng suy diễn mờ (Siemens)

8. Tóm tắt và các vấn đề cần quan tâm

Bộ điều khiển dùng logic mờ có thể được xem là hệ chuyên gia nhỏ trong thời gian thực, thiết lập từ kinh nghiệm của người vận hành hay các kỹ sư. Theo quan điểm điều khiển, bộ điều khiển mờ là một bộ điều khiển phi tuyến. Trong một số thiết lập điều khiển dạng PID, ngõ ra của bộ điều khiển là hàm của tín hiệu sai số và đạo hàm của sai số. Ứng dụng trong công nghiệp ngày càng tăng và càng có nhiều nhà sản xuất hàng tiêu dùng ứng dụng hệ điều khiển mờ trong các thiết bị điện tử dân dụng như máy rửa chén, máy giặt, các hệ truyền động tự động trên xe hơi.

Điều khiển mờ là một kỹ thuật điều khiển mới có thể xem như là mở rộng của các phương thức điều khiển truyền thống nhưng không thay thế phương pháp này. Điều khiển mờ cung cấp thêm công cụ đặc biệt cho các kỹ sư điều khiển để học hỏi về phương thức sử dụng khi cần thiết. Các hệ thống phi tuyến hay hệ chỉ được hiểu biết một phần đã làm cho phương thức đều khiển truyền thống phải dùng phương thức điều khiển mờ. Như thế công nghệ điều khiển dần bước đến tầm vóc cao hơn trong lĩnh vực điều khiển mà trước đó chưa thực hiện được.

Trong giới hàn lâm thì ngày càng nhiều nghiên cứu quan tâm đến điều khiển mờ. Trọng tâm là các phương pháp phân tích và tổng hợp hệ thống. Trong một số dạng điều khiển mờ, thí dụ như hệ tuyến tính Takagi-Sugeno, đã có nhiều ý niệm nghiên cứu đã được phát triển mạnh.

9. Bài tập

1. Có nhiều phương thức để tham số hóa mô hình và bộ điều khiển phi tuyến. Liệt kê ít nhất ba phương pháp khác nhau và giải thích các điểm khác biệt giữa các phương pháp này.

2. Vẽ sơ đồ điều khiển mờ PD (proportional-derivative), bao gồm cả đối tượng điều khiển. Giải thích cấu trúc bên trong của bộ điều khiển mờ PD, bao gồm các bộ lọc động, các luật nền, v.v,..

3. Cho thí dụ về luật nền và các hàm thành viên có liên quan trong bộ điều khiển mờ PI (proportional-integral). Cho biết phương pháp thiết kế các tham số và phương pháp xác định các tham số này?

4. Hảy tự định nghĩa về bộ điều khiển mờ. Cho biết khác biệt giữa bộ điều khiển mờ và các bộ điều khiển tuyến tính, như bộ PID hay điều khiển phản hồi trạng thái? Cho biết trường hợp nào thì bộ điều khiển mờ chứng tõ được khả năng vượt trội so với các bộ điều khiển tuyến tính?

5. Cho thí dụ về nhiều luật trong bộ điều khiển mờ Takagi–Sugeno. Cho biết các tham số của bộ điều khiển này? Cho biết bộ điều khiển này dùng được cho các đối tượng điều khiển nào?

6. Phần cứng dùng logic mờ có nhất thiết là phải dùng cho thiết lập các bộ điều khiển mờ không? Giải thích?

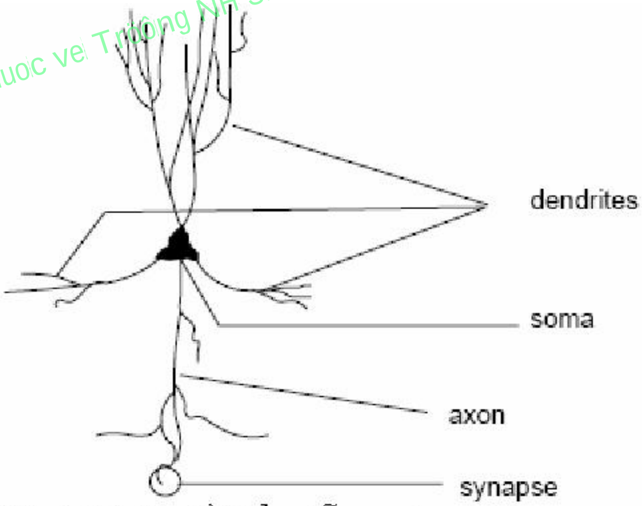
CHƯƠNG BẢY: MẠNG NƠRÔN NHÂN TẠO

1. Mở đầu

Mạng nơrôn và hệ mờ đều xuất phát từ mong muốn bắt chước lý luận của con người. Trong hệ mờ, quan hệ được biểu diễn một cách tường minh với dạng luật nếu-thì (if-then). Trong mạng nơrôn thì quan hệ này không được cho một cách tường minh, nhưng được "mã hóa" trong mạng và qua các tham số của mạng. Khác với trường hợp của các kỹ thuật dùng nền tri thức (knowledge-based techniques), mạng nơrôn không cần các kiến thức ẩn trong ứng dụng của mình.

Mạng nơrôn nhân tạo (ANN: Artificial neural nets) có thể xem như chức năng của mạng nơrôn sinh học nên thừa hưởng được một số ưu điểm của hệ thống sinh học so với các hệ thống tính toán thông thường. Mạng ANN có khả năng học được các quan hệ phức tạp thông qua việc khái quát hóa từ một lượng dữ liệu huấn luyện có giới hạn. Như thế mạng có thể sùng mô hình hóa dạng hộp đen các hệ thống phi tuyến, đa biến tĩnh và động, đồng thời có thể được huấn luyện từ tập dữ liệu vào-ra quan sát được từ hệ thống.

Ban đầu, các nghiên cứu về mạng ANN là nhằm mô hình chức năng sinh lý của não bộ, tạo ra mô hình có khả năng bắt chước con người thông qua quá trình tính toán hay ngay trong mức thực hiện phần cứng. Con người có khả năng thực hiện các tác vụ phức tạp như tri giác (perception), nhận dạng mẫu tốt hơn nhiều so với các máy tính hiện đại nhất. Con người còn có khả năng học từ các thí dụ và hệ não bộ của con người còn có khả năng chấp nhận lỗi. Các đặc tính này làm cho mạng ANN thích hợp với nhiều ứng dụng kỹ thuật như nhận dạng mẫu (pattern recognition), phân lớp, xấp xỉ hàm, nhận dạng hệ thống, v.v,…



Hình 7.1 Sơ đồ biểu diễn nơrôn sinh học

Mạng ANN thường là có dạng nhiều lớp gồm các phần tử xử lý đơn giản được gọi là nơrôn, liên kết nối với nhau thông qua các giá trị trọng lượng liên quan đến kết nối. Các thông tin có được từ ánh xạ vào – ra của mạng được lưu trữ trong các trọng lượng mạng.

2. Nơrôn sinh học

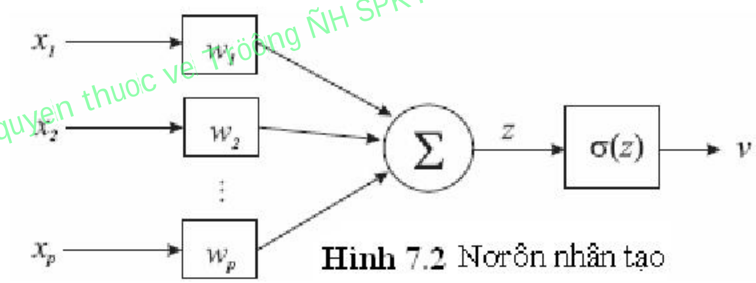
Nơrôn sinh học gồm có thân (hay *soma*), sợi trục thần kinh (*axon*) và nhiều *dendrites* (như vẽ ở hình 7.1). Dendrite là các ngõ vào của nơrôn, còn axon là ngõ ra.. Axon từ một nơrôn tạo thành các kết nối (synaptic) với các mơrôn khác. Axon là một ống dài, mỏng được chia thành nhiều nhánh tận cùng là những bầu với dendrite của các nơrôn khác. Khoảng hở giữa bầu này và dendrite của tế bào khác thì được gọi là *synapse*.

Các xung lan truyền xuống đến axon của nơrôn và tác động đến các synapses, gởi đi tín hiệu với nhiều cường độ khác nhau đến dendrites của các nơrôn khác. Cường độ của các tin hiệu này xác định hiệu suất (*efficiency*) của quá trình truyền synaptic. Tín hiệu đến dendrite có thể là cấm (*inhibitory*) hay kích thích (*excitatory*). Một nơrôn sinh học kích tức là gởi các tín hiệu xuống đến các axon, nếu mức kích thích vượt qua ngưỡng cấm một lượng tới hạn, tức là ngưỡng của nơrôn.

Các nghiên cứu về mô hình não bộ người đã được hình thành từ thế kỷ 19 (James, 1890). Cho đến năm 1943 trước khi McCulloch và Pitts (1943) tạo lập ý tưởng đầu tiên về mô hình toán học có tên gọi là nơrôn McCulloch-Pitts. Đến năm 1957, ý niệm về mạng nơrôn nhiều lớp mới được đưa ra. Tuy nhiên, tiến bộ đáng kể nhất trong các nghiên cứu về mạng nơrôn phải kể đến phương pháp lan truyền ngược dùng huấn luyện mạng nhiều lớp (Rumelhart, et al., 1986).

3. Nơrôn nhân tạo

Mô hình toán học của nơrôn sinh học (còn được gọi là nơrôn nhân tạo) bắt chước hoạt động của nơrôn sinh học theo nhiều cấp độ khác nhau. Ở đây, ta chỉ xem xét một nơrôn đơn giản, là một hàm tĩnh với nhiều ngõ vào (biểu diễn các dendrites) và một ngõ ra (biểu diễn axon). Mỗi ngõ vào liên quan đến thừa số trọng lượng (synaptic strength). Các ngõ vào có trọng lượng được thêm vào và đi qua một hàm phi tuyến, được gọi là hàm kích hoạt (*activation function*). Giá trị của hàm này là ngõ ra của nơrôn (xem hình 7.2).



Hình 7.2 Nơrôn nhân tạo

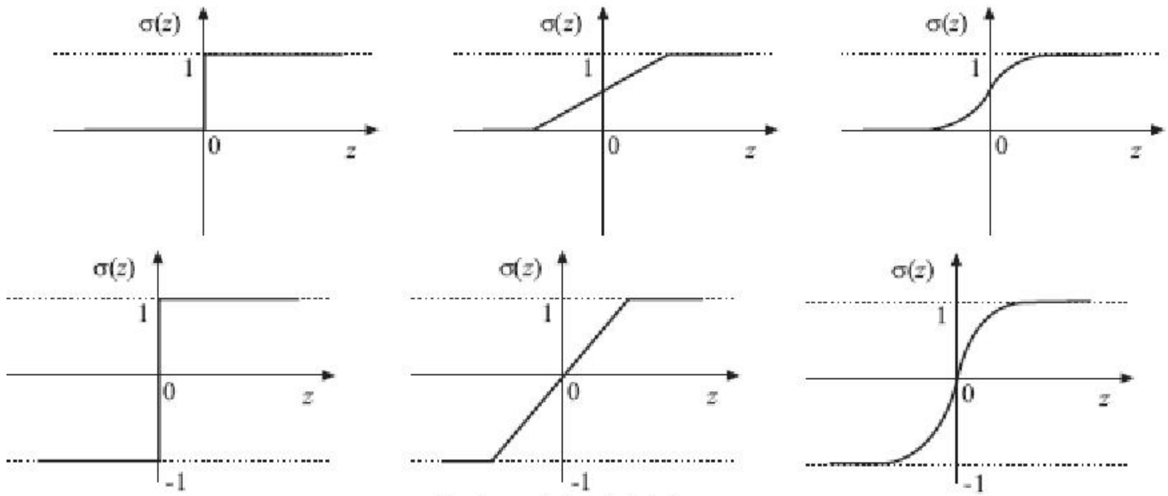
Tổng trọng lượng của các ngõ vào cho bởi:

$$z = \sum_{i=1}^{p} w_i x_i = w^T x \qquad (7.1)$$

Đôi khi, cần cộng thêm phân cực khi tính hàm kích hoạt của nơrôn:

$$z = \sum_{i=1}^{p} w_i x_i + b = \left[w^T b\right]\begin{bmatrix} x \\ 1 \end{bmatrix}$$

Phân cực bias được xem như là trọng lượng từ một ngõ vào có giá trị (đơn vị) không đổi, trường hợp trong công thức (7.1) ta đã bỏ qua giá trị này. Nhằm làm cho ý niệm đơn giản, ta tiếp tục dùng nguyên công thức (7.1). Hàm kích hoạt tạo ánh xạ kích hoạt của nơrôn z vào một khoảng nào đó, thí dụ [0, 1] hay [−1, 1]. Hàm kích hoạt thường có dạng hàm ngưỡng (threshold), hàm sigmoidal và hàm tanh (Hình 7.3).



Hình 7.3: Các dạng hàm kích hoạt

- Hàm ngưỡng (hard limiter)

$$\sigma(z) = \begin{cases} 0 & khi \quad z < 0 \\ 1 & khi \quad z \geq 0 \end{cases}$$

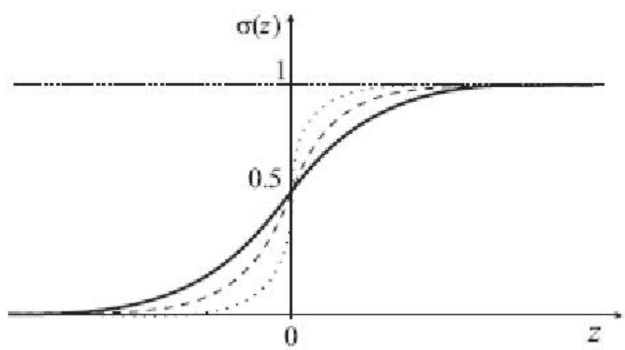
- Hàm tuyến tính từng phần (bảo hòa)

$$\begin{cases} 0 & khi & z < -\alpha \\ \frac{1}{2\alpha}(z+\alpha) & khi & -\alpha \leq z \leq \alpha \\ 1 & khi & z > \alpha \end{cases}$$

- Hàm Sigmoid

$$\sigma(z) = \frac{1}{1+\exp(-sz)}$$

Trong đó, s là hằng số xác định độ dốc đường cong sigmoidal. Khi $s \to 0$ thì hàm sigmoid là rất phẳng và khi $s \to \infty$ thì đường cong tiếp cận hàm ngưỡng. Hình 7.4 vẽ ba dạng đường cong với các giá trị s khác nhau. Thường dùng giá trị $s = 1$ (đường cong sậm màu trong hình 7.4).



Hình 7.4: Hàm kích hoạt dạng sigmoid

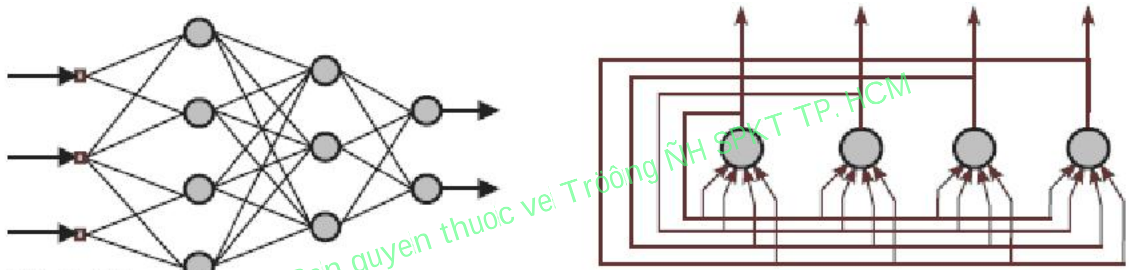
- Hàm tanh

$$\sigma(z) = \frac{1 - \exp(-2z)}{1 + \exp(-2p)}$$

4. Kiến trúc mạng nơrôn

Mạng nơrôn nhân tạo gồm nhiều nơrôn liên kết nối với nhau. Các nơrôn thường được sắp xếp trong nhiều *lớp*, được gọi là *kiến trúc* mạng. Mạng có thể có nhiều lớp hay một lớp, trong đó các nơrôn có thể được kết nối theo theo hay dạng cơ bản sau:

- *Mạng truyền thẳng*: các nơrôn được sắp xếp trong nhiều lớp. Thông tin chỉ truyền theo một hướng, từ lớp vào đến lớp ra.
- *Mạng hồi qui*: các nơrôn được sắp xếp trong một hay nhiều lớp và phản hồi được thực hiện trong nội tại các nơrôn hay với các nơrôn khác trong cùng mạng hay với các nơrôn của các lớp trước.



Hình 7.5:
Mạng ANN truyền thẳng nhiều lớp (trái) và mạng phản hồi một lớp (Hopfield) (trái)

Hình 7.5 trình bày mạng truyền thẳng nhiều lớp ANN (perceptron) và mạng hồi qui một lớp (mạng Hopfield).

Để đơn giản, ta sẽ tập trung vào mạng nơrôn nhiều lớp và một dạng mạng một lớp đặc biệt có tên là mạng RBF (radial basis function).

5. Học

Quá trình học trong mạng nơrôn sinh học có cơ sở là việc thay đổi cường độ liên kết nối giữa các nơrôn. Kết nối synaptic giữa các nơrôn đồng thời biểu lộ các tác động được cường điệu.

Trong mạng nơrôn nhân tạo, nhiều ý niệm học đã được đề cập. Một xấp xỉ toán của quá trình học, gọi là phương pháp học của Hebb và dùng trong dạng mạng Hopfield. Mạng nhiều lớp, thường dùng một số phương pháp tối ưu hóa nhằm tối thiểu sai biệt giữa đáp ứng thực và đáp ứng ra mong muốn của mạng.

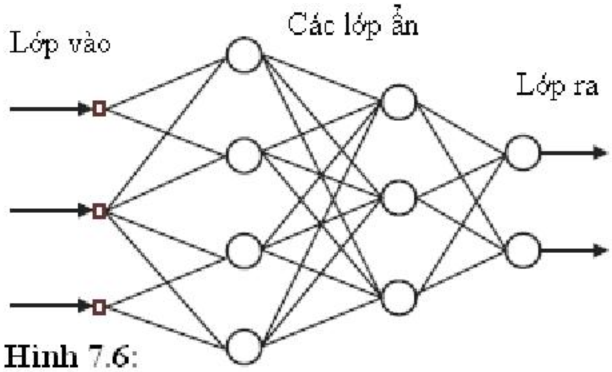
Có hai phương pháp học đã được ghi nhận: học có giám sát và học không giám sát:

- *Học có giám sát (supervised learning)*: mạng được cung cấp đồng thời các giá trị vào và giá trị ra đúng, và trọng lượng mạng được chỉnh định từ sai biệt với ngõ ra tính được. Phương pháp này được trình bày trong phần 7.6.3.
- *Học không giám sát (unsupervised learning)*: mạng chỉ được cấp các giá trị vào và trọng lượng mạng được chỉnh định dùng các giá trị của ngõ vào và giá trị ngõ ra hiện tại. Quá trình học không giám sát tương tự như xu hướng xâu chuỗi (clustering) đã trình bày ở chương 4.

Để đơn giản, ta chỉ khảo sát quá trình học có giám sát

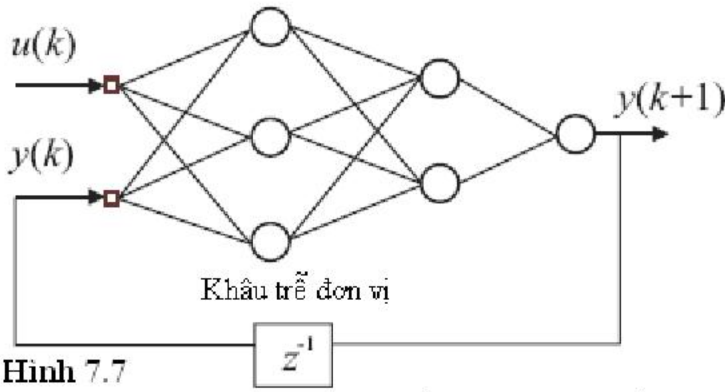
6. Mạng nơrôn nhiều lớp

Mạng nơrôn nhiều lớp (MNN) có một lớp vào, một lớp ra và môt số lớp ẩn (xem hình 7.6).



Hình 7.6:
Mạng nhiều lớp tiêu biểu với một lớp vào, một hay nhiều lớp ẩn và một lớp ra

Có thể thực hiện một mạng động dùng mạng truyền thẳng tĩnh kết hợp với khâu phản hồi. Ngõ ra của mạng được phản hồi về ngõ vào thông qua khâu trễ đơn vị z^{-1}. Hình 7.7 nêu thí dụ về mạng nơrôn được dùng biểu diễn hệ bậc nhất $y(k+1) = f_{nn}(y(k), u(k))$. Trong đó, vấn đề nhận dạng được tạo lập như bài toán xấp xỉ hàm tĩnh (xem thảo luận ở phần 3.6).



Hình 7.7
Mô hình mạng nơrôn của hệ thống động bậc nhất

Trong mạng nhiều lớp, cần phân biệt hai pha tính toán sau:

1. *Tính bước thuận (Feedforward computation).* Từ các ngõ vào x_i, $i = 1, \ldots, N$, tính được các ngõ ra của lớp ẩn, rồi dùng các giá trị này như là ngõ vào của lớp kế để tiếp tục tính, v.v,.. để cuối cùng tìm được ngõ ra của mạng.

2. *Chỉnh định trọng lượng mạng (Weight adaptation).* Ngõ ra của mạng được so sánh với ngõ ra đích. Sai biệt giữa hai giá trị này được gọi là sai số, và được dùng để chỉnh định trọng lượng mạng lớp ra, rồi đến lớp phía sau, v.v,.. cho đến khi sai số giảm. Phương pháp tính toán này được gọi là lan truyền ngược sai số (error backpropagation).

Thuật toán lan truyền ngược được Werbos (1974) và nhóm Rumelhart (1986) trình bày. Phần tiếp sẽ khai triển thuật toán.

6.1 Tính toán bước thuận

Xét trường hợp đơn giản là mạng MNN có một lớp ẩn (hình 7.8). Lớp nơrôn vào không thực hiện phép tính nào, mà chỉ đơn thuần phân bố các ngõ vào x_i đến các trọng lượng mạng w_{ij}^h của lớp ẩn. Các nơrôn trong lớp ẩn chứa hàm kích hoạt dạng *tanh*, còn hàm của nơrôn ra thường là tuyến tính. Trọng lượng lớp ra là w_{ij}^o.

Bước tính thuận được thực hiện trong ba bước:

1. Tính hàm kích hoạt của mơrôn lớp ẩn z_j:

$$z_j = \sum_{i=1}^{p} w_{ij}^h x_i + b_j^h \qquad j = 1, 2, \ldots, \mathrm{h}.$$

Trong đó w_{ij}^h và b_j^h lần lượt là trọng lượng và phân cực:

2. Tính các ngõ ra v_j của lớp ẩn:

$$v_j = \sigma(z_j) \qquad j = 1, 2, \ldots, \mathrm{n}.$$

3. Tính ngõ ra y_l của nơrôn lớp ra (và là ngõ ra toàn mạng):

$$y_l = \sum_{j=1}^{h} w_{jl}^o v_j + b_l^o \qquad l = 1, 2, \ldots, n.$$

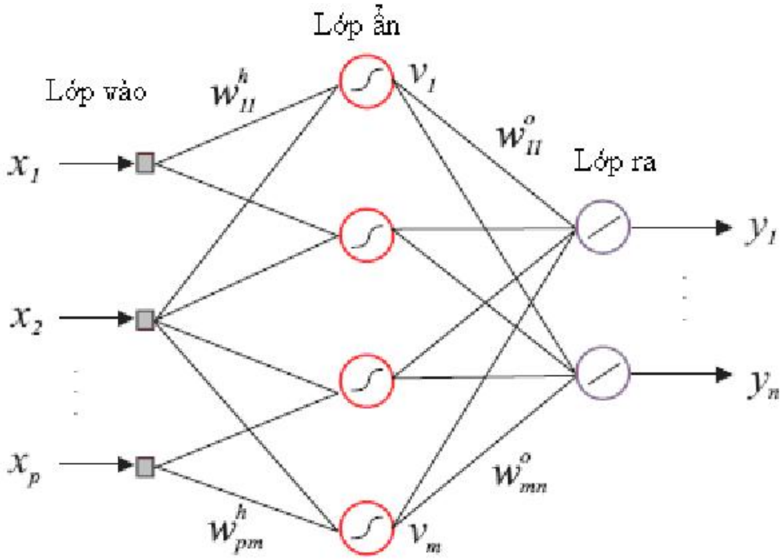
Trong đó w_{ij}^o và b_j^0 lần lượt là trọng lượng và phân cực.

Ba bước này có thể được viết gọn lại theo dạng ma trận:

$$\mathbf{Z} = \mathbf{X}_b \mathbf{W}^h$$
$$\mathbf{V} = \sigma(\mathbf{Z})$$
$$\mathbf{Y} = \mathbf{V}_b \mathbf{W}^o$$

Trong đó $\mathbf{X}_b = [\mathbf{X}\ \mathbf{1}]$ và $\mathbf{V}_b = [\mathbf{V}\ \mathbf{1}]$ và

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix}$$



Hình 7..8: Mạng nhiều lớp MNN với một lớp ẩn dùng hàm kích học là nơrôn tanh, và nơrôn ra tuyến tính

6.2 Khả năng xấp xỉ

Mạng nơrôn nhiều lớp có thể xấp xỉ nhiều dạng hàm với mức chính xác mong muốn. Nói rõ hơn, nhờ vào khả năng xếp chồng (superposition) các trọng lượng có dạng hàm sigmoid có khả năng tạo ánh xạ phức tạp. Thí dụ, xét mạng đơn giản MNN có một ngõ vào, một ngõ ra và một lớp ẩn có hai nơrôn tanh

Ngõ ra của mạng là:

$$y = w_1^o \tanh(w_1^h x + b_1^h) + w_2^h \tanh(w_2^h x + b_2^h)$$

Xem hình 7.9, trong đó mô tả ba bước tính thuận

Chú ý với hai lớp nơrôn đã có đủ khả năng biểu diễn các hàm không đơn điệu (nonmonotonic) tương đối phức tạp. Khả năng này của mạng nhiều lớp đã được Cybenko (1989) khẳng định như sau:

Mạng nơrôn nhiều lớp có ít nhất một lớp ẩn dung hàm kích hoạt sigmoid thì có thể xấp xỉ tốt bất kỳ hàm phi tuyến liên tục nào Rp → Rn *trong tập compac, nếu có đủ nơrôn lớp ẩn.*

Điểm yếu là trong phát biểu này chưa đề cập đến số lượng nơrôn cần có, cũng như phương thức xác định trọng lượng mạng, v.v,… Tuy còn có nhiều phương thức xấp xỉ hàm dạng khác như khai triển đa thức, chuỗi Fourier, mạng sóng con (wavelet), v.v,.. nhưng mạng nơrôn tỏ ra hiêu quả hơn khi thực hiện mức chính xác xấp xỉ với một số nơrôn cho trước. Điêu này được Barron (1993) phát biểu:

Mạng nơrôn với một lớp ẩn dùng hàm kích hoạt sigmoid có thể thực hiện sai số bình phương tích phân (integrated squared error (trường hợp hàm mịn) có bậc

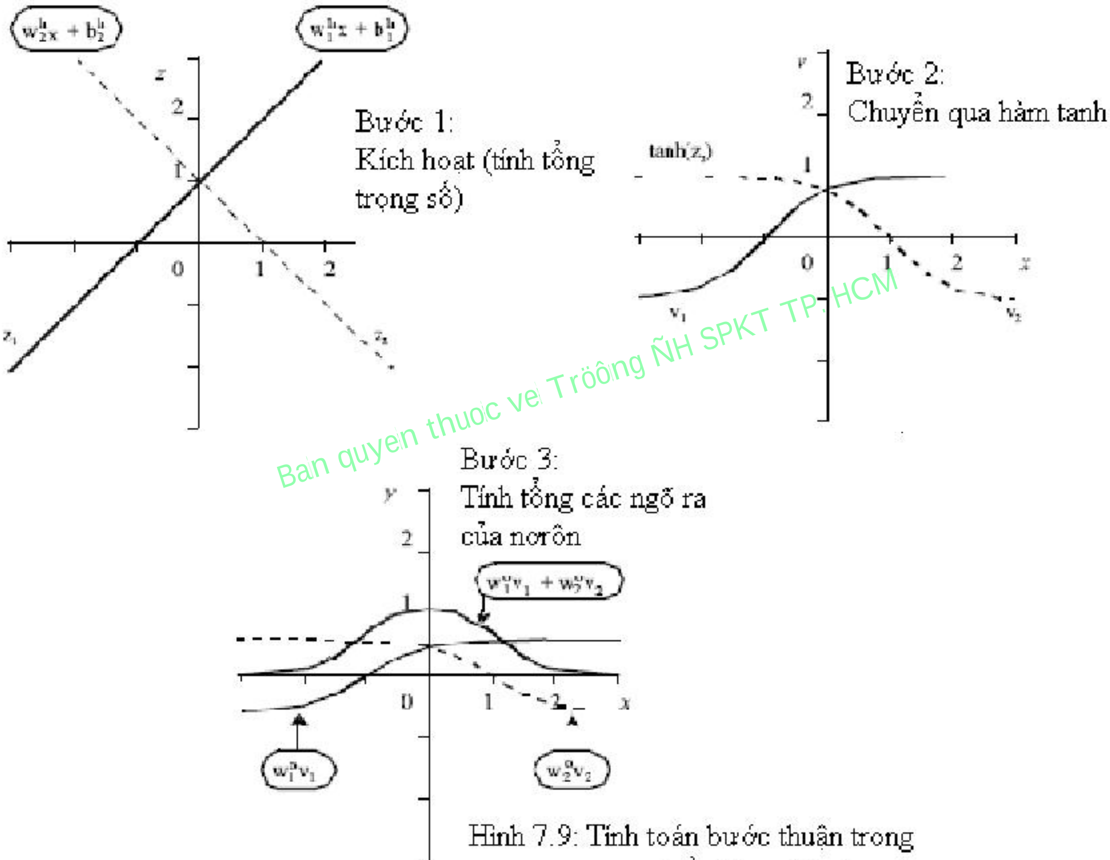
$$J = \mathrm{O}\left(\frac{1}{h}\right)$$

độc lập với chiều của không gian vào p, trong đó h là số nơrôn ẩn.

Đối với mô hình mở rộng (dạng đa thức, khai triển lượng giác, mô hình mờ singleton, v.v,…) có h thừa số, trong đó chỉ có một tổ hợp tuyến tính được chỉnh định, thì

$$J = O\left(\frac{1}{h^{2/p}}\right)$$

với p là chiều của ngõ vào.



Hình 7.9: Tính toán bước thuận trong mạng nhiều lớp với hai nơrôn

Thí dụ 7.1 (Độ chính xác xấp xỉ) Nhằm minh họa khác biệt giữa khả năng xấp xỉ của mạng sigmoid và các phương pháp khai triển hàm (thí dụ phương pháp đa thức), xét hai ngõ vào có chiều p:

i) $p = 2$ (hàm có hai biến):

đa thức $\quad J = O\left(\frac{1}{h^{2/2}}\right) = O\left(\frac{1}{h}\right)$

mạng nơrôn $\quad J = O\left(\frac{1}{h}\right)$

Như thế, khi $p = 2$, thì không có sự khác biệt trong quan hệ về độ phức tạp và tính chính xác giữa mạng sigmoid và các khai triển hàm.

ii) $p = 10$ (hàm có mười biến) và $h = 21$:

đa thức $J = O\left(\frac{1}{21^{2/10}}\right) = 0{,}54$

mạng nơrôn $J = O\left(\frac{1}{21}\right) = 0{,}048$

Mức chính xác xấp xỉ của mạng sigmoid cho thấy tốt hơn về mặt biên độ. Ta xét tiếp xem cần có bao nhiêu thừa số trong khai triển hàm (thí dụ trường hợp khai triển đa thức) nhằm có cùng mức chính xác của mạng nơrôn:

$$O\left(\frac{1}{h_n}\right) = O\left(\frac{1}{h_b}\right)$$

$$h_n = h_b^{1/p} \Rightarrow h_b = \sqrt{h_n^p} = \sqrt{21^{10}} \approx 4.10^6$$

Như thế thì mạng nơrôn, ít nhất về mặt lý thuyết, có thể xấp xỉ được các hàm khác. Câu hỏi đặt ra là phương thức xác định cấu trúc thích hợp (số lớp ẩn, số nơrôn) và tham số trọng lượng mạng. Mạng có một lớp ẩn thường là đủ (về mặt lý thuyết thì luôn luôn đủ). Càng nhiều lớp thì cho phép khớp tốt hơn, nhưng cần thời gian huấn luyện lâu hơn.. Chọn đúng số nơrôn trong lớp ẩn điều điều cốt yếu để có được kết quả tốt. Quá ít nơrôn thì không khớp tốt được, nhưng khi quá nhiều nơrôn thì đưa đến quá khớp (không có tính khái quát dữ liệu). Thường cần có yếu tố thõa hiệp từ các phương pháp thử và sai.

6.3 Huấn luyện, Lan truyền ngược sai số

Huấn luyện là quá trình cập nhật trong lượng mạng nhiều lớp sao cho sai số giữa ngõ ra đích và ngõ ra của mạng được tối thiểu. Giả sử có tập N dữ liệu:

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}, \qquad D = \begin{bmatrix} d_1^T \\ d_2^T \\ \vdots \\ d_N^T \end{bmatrix}$$

Trong đó, x là ngõ vào của mạng và d là ngõ ra đích. Quá trình huấn luyện gồm hai bước:

1. *Bước tính thuận.* Từ ngõ vào x_i, $i = 1, \ldots, N$, hàm kích hoạt lớp ẩn, tính được ngõ ra mạng như sau:

$$\mathbf{Z} = \mathbf{X}_b\mathbf{W}^h, \qquad \mathbf{X}_b = [\mathbf{X} \quad \mathbf{1}]$$
$$\mathbf{V} = \sigma(\mathbf{Z})$$
$$\mathbf{Y} = \mathbf{V}_b\mathbf{W}^o, \qquad \mathbf{V}_b = [\mathbf{V} \quad \mathbf{1}]$$

2. *Cập nhật trọng lượng mạng.* Ngõ ra mạng được so sánh với ngõ ra đích. Sai biệt giữa hai giá trị này được gọi là sai số:

$$\mathbf{E} = \mathbf{D} - \mathbf{Y}$$

Sai số này được dùng để chỉnh định trọng lượng mạng thông qua phép tối thiểu hóa hàm chi phí (tổng bình phương sai số):

$$J(w) = \frac{1}{2}\sum_{k=1}^{N}\sum_{j=1}^{l} e_{kk}^{2} = trace(EE^{T})$$

$$\mathbf{w} = [\mathbf{W}^{h} \quad \mathbf{W}^{o}]$$

Quá trình huấn luyện mạng MNN tạo lập thành bài toán *tối ưu hóa phi tuyến* theo trọng lượng mạng.

Có nhiều phương pháp có thể dùng:

- Lan truyền ngược sai số (gradien bậc nhất).
- Phương pháp Newton, Levenberg-Marquardt (gradien bậc hai).
- Gradien liên hợp.
- Variable projection.
- Thuật toán di truyền, và các phương pháp khác . . .

Phương pháp gradien bậc nhất dùng luật cập nhật trọng lượng sau:

$$w(n+1) = w(n) - \alpha(n)\nabla J\big(w(n)\big) \qquad (7.5)$$

Trong đó $w(n)$ là vectơ với các trọng lượng tại các bước tính lặp *n*, $\alpha(n)$ là biến tốc độ học và $\nabla J(w)$ là Jacobi của mạng.

$$\nabla J(w) = \left[\frac{\partial J(w)}{\partial w_1}, \quad \frac{\partial J(w)}{\partial w_2}, \quad \cdots, \quad \frac{\partial J(w)}{\partial w_M}\right]^{T}$$

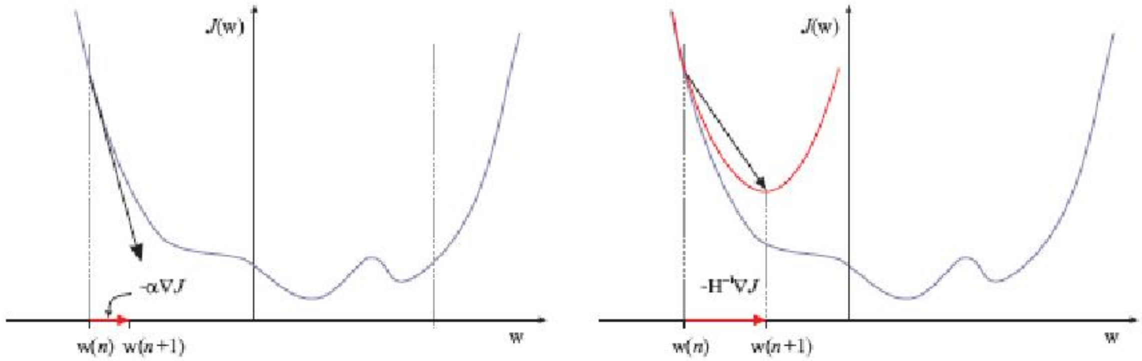
Bài toán tối ưu hóa phi tuyến còn lại là giải dùng thừa số đầu tiên của chuỗi khai triển Taylor. Phương pháp gradien bậc hai thì dùng thừa số thứ hai (đoạn có độ cong) như là:

$$J(w) \approx J(w_0) + \nabla J(w_0)^{T}(w - w_0) + \frac{1}{2}(w - w_0)^{T} H(w_0)(w - w_0)$$

Trong đó $\mathbf{H}(\mathbf{w}_0)$ là ma trận Hess tại điểm cho trước $\mathbf{w}_0$ trong không gian trọng lượng. Khi khi sắp xếp lại, luật cập nhật trọng lượng có dạng sau:

$$w(n+1) = w(n) - H^{-1}\big(w(n)\big)\nabla J\big(w(n)\big) \qquad (7.6)$$

Điều khác biệt cơ bản giữa hai công thức (7.5) và (7.6) ilà kích thước của bước giảm gradien, theo mô tả trong hình 7.10.



(a) gradien bậc nhất (b) gradien bậc hai

Hình 7.10: Phương pháp tối ưu hóa dùng gradien bậc nhất và giadien bậc hai

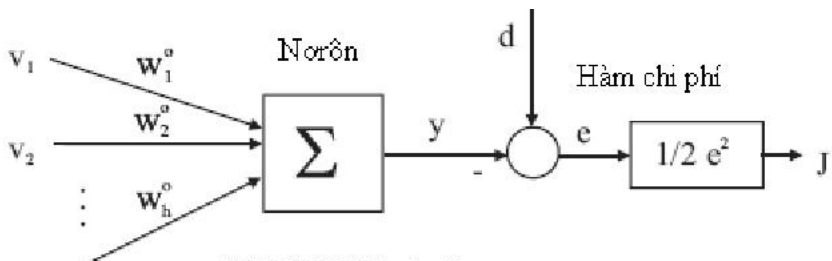
Phương pháp bậc hai thường có hiệu quả hơn phương pháp bậc nhất. Tuy nhiên, ở đây trình bày phương pháp lan *truyền ngược sai số* (phương pháp gradien bậc nhất) là phương pháp dễ nắm bắt rồi mới tiếp tục một ít đến phương pháp bậc hai.

Ý tưởng chính của phương pháp lan truyền ngược (backpropagation BP) có thể được phát biểu như sau:

- Tính toán sai số tại ngõ ra
- Chỉnh định trọng lượng tại các ngõ ra
- Lan truyền ngược sai số qua mạng rồi chỉnh định trọng lượng lớp ẩn

Phát triển phương pháp lan truyền ngược để xử lý tập dữ liệu, rất thích hợp với các trường hợp trực tuyến và không trực tuyến. Đầu tiên, xét trọng lượng lớp ra và trọng lượng lớp ẩn.

Trọng lượng lớp ra. Xét nơrôn trong lớp ra vẽ ở hình 7.11



Hình 7.11 Nơrôn lớp ra

Hàm cho phí cho bởi:

$$J = \frac{1}{2}\sum_l e_l^2 \quad \text{trong đó} \quad e_l = d_l - y_l \qquad \text{và} \quad y_l = \sum_j w_j^0 v_j$$

Ma trận Jacobi là:

$$\frac{\partial J}{\partial w_{jl}^{0}} = \frac{\partial J}{\partial e_l}\frac{\partial e_l}{\partial y_l}\frac{\partial y_l}{\partial w_{jl}^{0}} \quad (7.8)$$

Và đạo hàm riêng phần

$$\frac{\partial J}{\partial e_l} = e_l, \qquad \frac{\partial e_l}{\partial y_l} = -1, \qquad \frac{\partial y_l}{\partial w_{jl}^{0}} = v_j \quad (7.9)$$

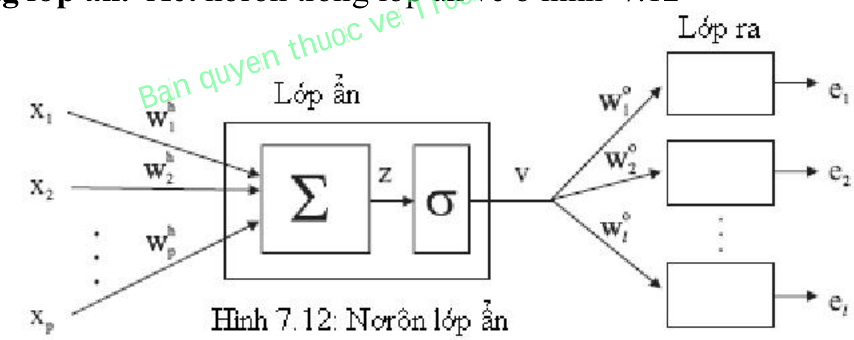
Như thế, tại lớp ra thì Jacobi là:

$$\frac{\partial J}{\partial w_{jl}^{0}} = -v_j e_l \quad (7.10)$$

Từ (7.5) thì luật cập nhật trọng lượng lớp ra là:

$$w_{jl}^{0}(n+1) = w_{jl}^{0}(n) + \alpha(n) v_j e_l \quad (7.11)$$

Trọng lượng lớp ẩn. Xét nơrôn trong lớp ẩn vẽ ở hình 7.12



Hình 7.12: Nơrôn lớp ẩn

Jacobi là:

$$\frac{\partial J}{\partial w_{ij}^{h}} = \frac{\partial J}{\partial v_j}\frac{\partial v_j}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}^{h}} \quad (7.12)$$

Và đạo hàm riêng phần (sau một số bước tính toán:

$$\frac{\partial J}{\partial v_j} = \sum_j -e_l w_{jl}^{0}, \qquad \frac{\partial v_j}{\partial z_j} = \sigma_j^{'}(z_j), \qquad \frac{\partial z_j}{\partial w_{ij}^{h}} = x_i \quad (7.13)$$

Việc tính tìm ra công thức trên có thể xem như bài tập cho độc giả, thế (7.12) thì có Jacobi:

$$\frac{\partial J}{\partial w_{ij}^{h}} = -x_i.\sigma_j^{'}(z_j).\sum_l e_l w_{jl}^{0} \quad (7.14)$$

Từ (7.5), tìm ra phương trình cập nhật trọng lượng mạng trong óp ẩn là:

$$w_{ij}^{h}(n+1) = w_{ij}^{h}(n) + \alpha(n) x_i . \sigma_j^{'}(z_j) . \sum_l e_l w_{jl}^{0} \qquad (7.15)$$

Từ phương trình này, ta thấy sai số được lan truyền từ lớp ra về lớp ẩn, nên có tên gọi là “lan truyền ngược”. Thuật toán được tóm tắt trong Algorithm 7.1.

Algorithm 7.1 Lan truyền ngược

Khởi tạo ngẫu nhiên trọng lượng mạng.

Bước 1: Giới thiệu các ngõ vào và các ngõ ra đích

Bước 2: Tính toán ngõ ra hiện tại và sai số.

Bước 3: Tìm gradien và cập nhật trọng lượng mạng theo:

$$w_{ji}^{0} := w_{ji}^{0} + \alpha v_l e_l$$
$$w_{ij}^{h} := w_{ij}^{h} + \alpha x_i . \sigma_j^{'}(z_j) . \sum_l e_l w_{jl}^{0}$$

Làm lại bước 1

Theo phương pháp này, các điểm dữ liệu lần lượt được học nên thích hợp cho quá trình học trực tuyến. Tuy vậy, phương pháp cũng có thể ứng dụng cho phương pháp học không trực tuyến nếu tìm được toàn dữ liệu (whole batch of data)
Việc đưa ra toàn tập dữ liệu được gọi là một chu kỳ học (*epoch)*. Thường thì, cần dùng nhiều epochs để khớp được dữ liệu. Về mặt tính toán, thì nên đưa tập dữ liệu như whole batch. Công thức theo lan truyền ngược được áp dụng vào trong các vectơ dữ liệu thay vì cho từng mậu riêng lẽ.

7. Mạng RBF

Mạng RBF (radial basis function network RBFN) là mạng hai lớp. Có hai khác biệt cơ bản với mạng nhiều lớp dạng sigmoid.

- Hàm kích hoạt trong lớp ẩn có dạng hàm radial basis so với hàm sigmoid. Hàm radial được giải thích như sau.
- Trọng lượng chỉ được hiệu chỉnh tai lớp ra. Kết nối từ lớp vào đến lớp ẩn là không đổi (trọng lượng đơn vị). Tuy nhiên, các tham số của hàm radial được chỉnh định.

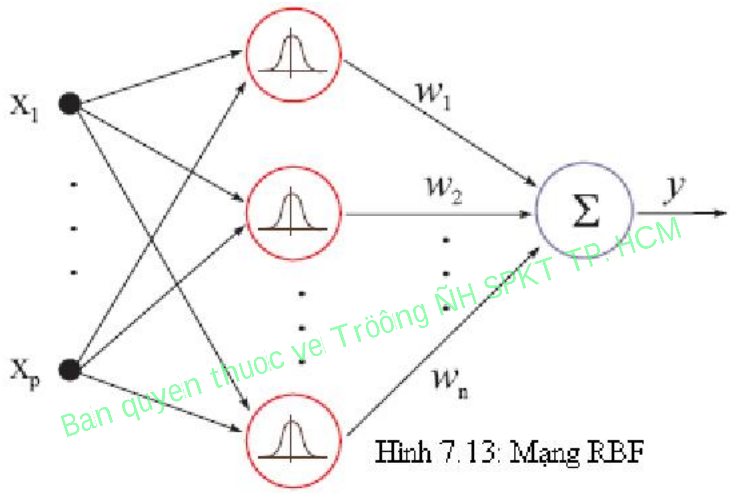
Các nơrôn lớp ra là tuyến tính, do đó mạng RBFN thuộc nhóm các mô hình dạng khai triển hàm, tương tự như mô hình singleton trong phần 3.3 và thực hiện ánh xạ f: → Rp → R

$$y = f(x) = \sum_{i=1}^{n} w_i \phi_i(x, c_i) \qquad (7.17)$$

Các dạng hàm cơ sở $\phi_i(x,c_i) = \phi_i(\|x - c_i\|) = \phi_i(r)$ thường được chọn là:

- $\phi(r) = \exp(-r^2/\rho^2)$, là dạng hàm Gauss
- $\phi(r) = r^2 \log(r)$, là dạng hàm thin-plate-spline
- $\phi(r) = r^2$, dạng hàm quân phương
- $\phi(r) = (r^2 + \rho^2)^{\frac{1}{2}}$, dạng hàm multiquadratic

Hình 7.13 minh họa kiến trúc của mạng RBF



Hình 7.13: Mạng RBF

Ba tham số tự do của mạng RBF là các trọng lượng ra w_i và các tham số của hàm cơ sở and (trọng tâm $\mathbf{c}_i$ và bán kính ρ_i).

Ngõ ra của mạng (7.17) tuyến tính theo trọng lương w_i, nên có thể được ước lượng dùng phương pháp bình phương tối thiểu. Với từng điểm dữ liệu $\mathbf{x}_k$, tính các ngõ ra của nơrôn là

$$v_{ki} = \varphi_i(\mathbf{x},\ \mathbf{c}_i)\ .$$

Do ngõ ra tuyến tính theo trong lượng w_i, viết được phương trình ma trận sau cho toàn tập dữ liệu:

$$\mathbf{d} = \mathbf{V}\mathbf{w},$$

trong đó $\mathbf{V} = [v_{ki}]$ là ma trận các ngõ ra của nơrôn tại từng điểm dữ liệu và $\mathbf{d}$ là vectơ các ngõ ra đích của mạng RBFN. Phép bình phương tối thiểu ước lượng được trong lượng $\mathbf{w}$ là:

$$\mathbf{w} = [\mathbf{V}^T\mathbf{V}]^{-1}\mathbf{V}^T\,\mathbf{y}$$

Việc huấn luyện các tham số mạng RBF $\mathbf{c}_i$ và ρ_i là bài toán tối ưu hóa phi tuyến có thể được giải từ các phương pháp cho ở phần 7.6.3. Vị trí trọng tâm ban đ62u thường được xác định từ phương pháp xâu chuỗi (clustering) (xem Chương 4).

8. Tóm tắt và các điều cần chú ý

Mạng nơrôn nhân tạo, có cội nguồn từ chức năng của nơrôn mạng sinh học là có thể học được các quan hệ phức tạp thông qua qua trình tổng quat hóa từ một lượng dữ liệu huấn luyện giới hạn. Từ đó, mạng nơrôn có thể được dùng làm mô hình (dạng hộp đen) cho các hệ thống phi tuyến, đa biến tĩnh và động và có thể được huấn luyện dùng tập dự liệu vào-ra quan sát được từ hệ thống. Tuy có nhiều dạng cấu trúc mạng đã được đưa ra nhưng trong điều khiển và nhận dạng thì dạng mạng nhiều lớp và mạng RBF được dùng nhiều nhất. Từ cấu trúc này, đã xuất hiện nhiều thuật toán huấn luyện rất hiệu quả.

9. Bài tập

1. Cho biết yếu tố ban đầu nào thúc đẩy sự phát triển của mạng nơrôn nhân tạo? Cho ít nhất hai thí dụ về ứng dụng của mạng nơrôn nhân tạo trong kỹ thuật?
2. Vẽ sơ đồ khối và trình bày các công thức của mạng nơrôn nhân tạo, giải thích các thuật ngữ và ký hiệu này?
3. Cho ít nhất ba thí dụ về hàm kích hoạt?
4. Giải thích thuật ngữ " huấn luyện" mạng?
5. Trình bày các bước trong thuật toán lan truyền ngược? và cho biết thuật toán này dùng với cấu trúc mạng dạng nào?
6. Giải thích sự khác biệt giữa phương pháp tối ưu hóa bậc một và bậc hai của gradien?
7. Tìm luật lan truyền ngược của ngõ ra nơrôn có hàm kích hoạt dạng sigmoid?
8. Cho biết sự khác biệt giữa mạng truyền thẳng nhiều lớp và mạng RBF?
9. Xét hệ thống động $y(k+1) = f(y(k), y(k-1), u(k), u(k-1))$, trong đó f hàm ẩn. Nếu ta muốn xấp xỉ hàm f bằng mạng nơrôn dùng chuỗi dữ liệu vào-ra N đo từ hệ thống ẩn $\{(u(k), y(k)) | k = 0, 1, \ldots, N\}$.
   a) Chọn kiến trúc mạng, vẽ sơ đồ mạng và định nghĩa các ngõ vào và các ngõ ra.
   b) Tham số tự do nào cần được huấn luyện (tối ưu hóa) nhằm giúp mạng khớp được với dữ liệu?
   c) Định nghĩa hàm chi phí dùng huấn luyện mạng (viết công thức) và kể ra thí dụ hai phương pháp có thể dùng để huấn luyện tham số mạng.

CHƯƠNG 8:
HỆ THỐNG ĐIỀU KHIỂN MỜ VÀ ĐIỀU KHIỂN DÙNG MẠNG NƠRÔN

Chương này trình này việc thiết kế bộ điều khiển phi tuyến dùng các mô hình fuzzy và mạng nơrôn thích hợp dùng cho hệ cần điều khiển. Một số kỹ thuật dùng được cho cả hệ mờ và mạng nơrôn (điều khiển dùng mô hình dự báo, điều khiển dùng phương pháp tuyến tính hóa phản hồi), một số kỹ thuật thì chỉ thích hợp cho mô hình mờ (gain scheduling, analytic inverse).

1. Điều khiển nghịch

Phương pháp đơn giản nhất trong thiết kế dùng mô hình của bộ điều khiển phi tuyến là điều khiển nghịch (*inverse control*). Phương pháp này có thể dùng được cho các hệ thống ổn định vòng hở (hay đã được ổn định dùng phản hồi) và có đặc tính nghịch ổn định, tức là các hệ thống không có đáp ứng pha không tối thiểu.
Để đơn giản, ứng dụng phương pháp đối với mô hình SISO không có khâu trễ từ ngõ vào đến ngõ ra. Từ đó. Có thể viết mô hình phi tuyến tổng quát cho hệ mờ và mạng nơrôn là:

$$y(k+1) = f\big(x(k), u(k)\big) \tag{8.1}$$

Mô hình có các ngõ vào là các trạng thái hiện tại là:

$$x(k) = [y(k), \ldots, y(k-n_y+1), u(k-1), \ldots, u(k-n_u+1)]^T \tag{8.2}$$

Và ngõ vào hiện tại $u(k)$. Mô hình dự báo ngõ ra của hệ thống trong bước thời gian kế tiếp, $y(k+1)$. Hàm f biểu diễn ánh xạ phi tuyến của hệ mờ hay mạng nơrôn.

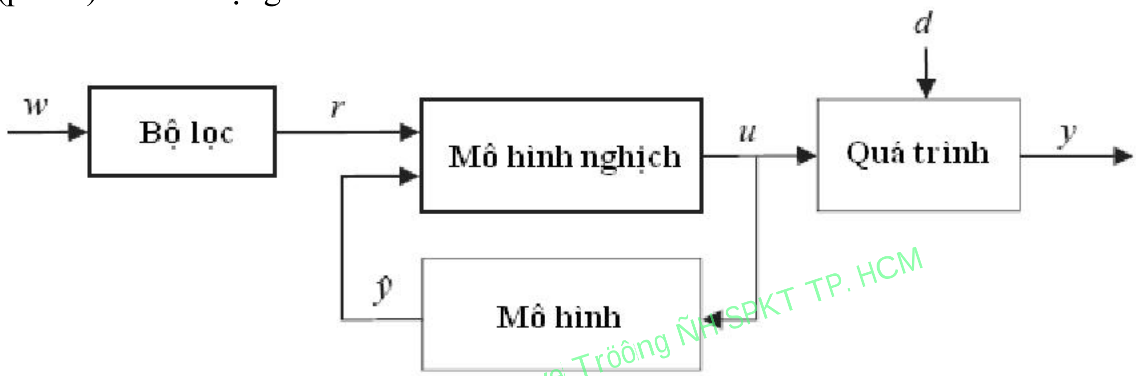
Mục tiêu của điều khiển nghịch là tính toán với trạng thái hiện tại $x(k)$, ngõ vào hiện tại $u(k)$, thì ngõ ra của hệ thống tại bước thời gian kế có giá trị bằng ngõ ra tham chiếu $r(k+1)$. Điều này có thể thực hiện được nếu từ (8.1) có thể tìm được:

$$u(k) = f^{-1}\big(x(k), r(k+1)\big) \tag{8.3}$$

Trường hợp này thì tín hiệu tham chiếu $r(k+1)$ đã được ngõ ra $y(k+1)$ thay thế. Mô hình nghịch có thể dùng làm bộ điều khiển tiếp tới vòng hở (open-loop feedforward controller) hay như bộ điều khiển vòng hở dùng phản hồi từ ngõ ra (còn được gọi là bộ điều khiển phản hồi vòng hở). Khác biệt cơ bản giữa hai sơ đồ điều khiển này nằm ở phương thức cập nhật $x(k)$.

1.1 Điều khiển tiếp tới vòng hở

Trạng thái **x**(k) của mô hình nghịch (8.3) được cập nhật dùng ngõ ra của mô hình (8.1), xem hình 8.1. Do không có phản hồi từ ngõ ra hệ, nên bộ điều khiển được ổn định nhờ độ ổn định vòng hở, của hệ có pha tối thiểu. Tuy nhiên, khi mô hình không khớp (mismatch) hay có tồn tại yếu tố nhiễu *d* tạo sai số xác lập tại ngõ ra của hệ thống. Sai số này có thể được bù (compensated) dùng một số dạng phản hồi, thí dụ như trường hợp sơ đồ điều khiển dùng mô hình nội tại (IMC) sẽ mô tả trong phần 8.1.5.
Bên cạnh mô hình và bộ điều khiển, thì sơ đồ còn có thêm bộ lọc sửa dạng tín hiệu tham chiếu (reference-shaping filter). Bộ lọc này thường là mô hình tham chiếu bậc một hay bậc hai, có nhiệm vụ tại các đặc tính động cần có và nhằm tránh yếu tố định (peaks) của tác động điều khiển.

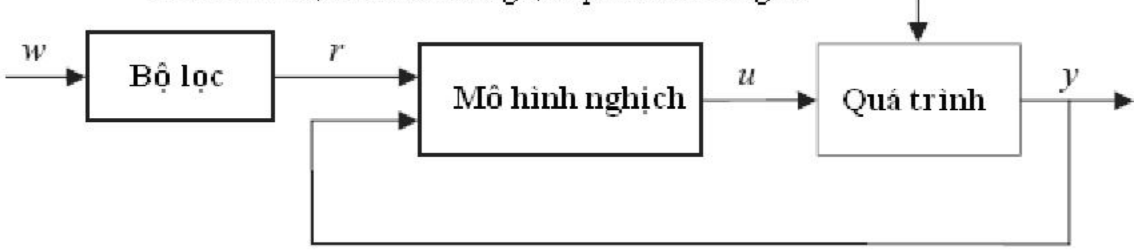


Hình 8.1: Bộ điều khiển nghịch tiếp tới vòng hở

1.2 Điều khiển phản hồi vòng hở

Ngõ vào **x**(k) của mô hình nghịch (8.3) được cập nhật dùng ngõ ra của tự thân hệ, xem hình 8.2. Bộ điều khiển thì thực tế hoạt động như hệ vòng hở (không dùng sai số giữa tín hiệu tham chiếu và ngõ ra), tuy nhiên ngõ ra hiện tại $y(k)$ lại được dùng để cập nhật trạng thái trong **x**(k) trong từng bước thời gian của bộ điều khiển. Điều này cải thiện tình chính xác của dự báo và giảm thiểu yếu tố offsets. Tuy nhiên, trong lúc này thì hệ thống có thể bị dao động hay không ổn định khi có sự hiện diện của nhiễu hay có yếu tố không khớp mô hình. Trong sơ đồ cũng cần có bộ lọc sửa dạng tín hiệu tham chiếu (reference-shaping filter)

Hình 8.2: Bộ điều khiển nghịch phản hồi vòng hở



1.3 Tính toán bộ nghịch

Thông thường thì rất khó tìm hàm ngược f^{-1} theo dạng giải tích. Tuy nhiên, có thể tìm được từ phương pháp tìm kiếm tối ưu dạng số. Định nghĩa hàm mục tiêu:

$$J(u(k)) = (r(k+1) - f(x(k), u(k)))^2 \quad (8.5)$$

Tối thiểu hóa J theo $u(k)$ cho tín hiệu điều khiển tương ứng với hàm ngược (8.3), nếu tồn tại, hay là xấp xỉ tốt nhất có thể. Có thể dùng nhiều phương pháp tối ưu khác nhau (như Newton hay Levenberg- Marquardt). Xu hướng này mở rộng trực tiếp được cho hệ MIMO. Yếu điểm lớn nhất là độ tính toán phức tạp do phải thực hiện trực tuyến phép tối ưu hóa số.

Một số dạng đặc biệt của (8.1) có thể được tính trực tiếp phần nghịch bằng pháp giải tích. Thí dụ phép ánh xạ ngõ vào của mô hình Takagi–Sugeno (TS) và mô hình singleton model dùng hàm thành viên $u(k)$ dạng tam giác.

Affine TS Model. Xét mô hình hệ mờ dạng vào-ra Takagi–Sugeno (TS):

R_i : **Nếu** $y(k)$ là A_{i1} **và** . . . **và** $y(k - n_y + 1)$ là A_{iny} **và** $u(k - 1)$ là B_{i2} **và** . . . **và** $u(k - n_u + 1)$ là B_{inu} **thì**

$$y_i(k+1) = \sum_{j=1}^{ny} a_{ij} y(k-j+1) + \sum_{j=1}^{nu} b_{ij} u(k-j+1) + c_i \quad (8.6)$$

Trong đó $i = 1, \ldots, K$ là các luật, A_{il}, B_{il} là các tập mờ, và a_{ij} , b_{ij} , c_i là tham số hệ quả (then-part). Gọi các biến quá khứ (bao gồm cả $u(k)$), là:

$$x(k) = [y(k), y(k-1), \ldots, y(k-n_y+1), u(k-1), \ldots, u(k-n_u+1)] \quad (8.8)$$

Dùng công thức trung bình trọng lượng (weighted mean) để tính $y(k+1)$:

$$y(k+1) = \frac{\sum_{i=1}^{K} \beta_i(x(k)) y_i(k+1)}{\sum_{i=1}^{K} \beta_i(x(k))} \quad (8.9)$$

Trong đó β_i là mức độ hoàn thành (fulfillment) của tiền đề sau:

$$\beta_i(x(k)) = \mu_{Ai1}(y(k)) \wedge \cdots \wedge \mu_{Ainy}(y(k-n_y+1)) \wedge \\ \mu_{Bi2}(u(k-1)) \wedge \cdots \wedge \mu_{Binu}(u(k-n_u+1)). \quad (8.10)$$

Do các tiền đề trong (8.6) không bao hàm thừa số vào $u(k)$, nên ngõ ra của mô hình $y(k + 1)$ là phép affine của ngõ vào $u(k)$. Để minh họa, định nghĩa mức hoàn thành chuẩn

$$\lambda_i(x(k)) = \frac{\beta_i(x(k))}{\sum_{j=1}^{K} \beta_j(x(k))} \quad (8.12)$$

Và thay hệ quả (8.6) và giá trị λ_i từ (8.12) vào (8.9):

$$y(k+1) = \sum_{i=1}^{K} \lambda_i(x(k)) \left[\sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=2}^{n_u} b_{ij} u(k-j+1) + c_i \right] + \\ + \sum_{i=1}^{K} \lambda_i(x(k)) b_{i1} u(k) \quad (8.13)$$

Đây là hệ affine-vào phi tuyến có thể được với theo thừa số tổng quát:

$$y(k+1) = g\big(x(k)\big) + h\big(x(k)\big)u(k) \tag{8.15}$$

Mục tiêu là ngõ ra của mô hình tại bước thời gian (k + 1) phải bằng với ngõ ra tham chiếu $y(k + 1) = r(k + 1)$, thì ngõ vào tương ứng $u(k)$, được tính toán từ phép tính đại số đơn giản:

$$u(k) = \frac{r(k+1) - g\big(x(k)\big)}{h\big(x(k)\big)} \tag{8.17}$$

Từ (8.13) ta tìm được luật điều khiển mô hình nghịch:

$$u(k) = \frac{r(k+1) - \sum_{i=1}^{K} \lambda_i\big(x(k)\big)\Big[\sum_{j=1}^{n_y} a_{ij}\, y(k-j+1) + \sum_{j=2}^{n_u} b_{ij}\, u(k-j+1) + c_i\Big]}{\sum_{i=1}^{K} \lambda_1\big(x(k)\big) b_{ij}} \tag{8.18}$$

Mô hình Singleton. Xét mô hình mờ singleton SISO. Trong chương này, để đơn giản ta không ghi chỉ số của luật. Luật mờ cho bởi biểu thức sau:

Nếu $y(k)$ là A_1 **và** $y(k-1)$ là A_2 **và** . . . **và** $y(k-n_y+1)$ là A_{ny}
và $u(k)$ là $B1$ **và** . . . **và** $u(k-n_u+1)$ là B_{nu} (8.19)
thì $y(k+1)$ là c,

Trong đó $A_1, \ldots, A_{ny}$ and $B_1, \ldots, B_{nu}$ là tập mờ và c là singleton, xem (3.42). Dùng vectơ trạng thái $\mathbf{x}(k)$ trong (8.8), có chứa các giá trị ngõ vào quá khứ $n_u - 1$, $n_y - 1$ giá trị ngõ ra quá khứ và ngõ ra hiện tại, tức là các biến trạng thái trước đó trong (8.19). Tập mờ tương ứng được tổ hợp vào một tập mờ trạng thái nhiều chiều X, dùng toán tử t-norm trên không gian tích Cartesian của biến trạng thái: $X = A_1 \times \cdots \times A_{ny} \times B_2 \times \cdots \times B_{nu}$. Để đơn giản, viết B thay cho B_1. Luật (8.19) viết lại thành:

Nếu $\mathbf{x}(k)$ là X **và** $u(k)$ là B **thì** $y(k+1)$ là c . (8.21)

Chú ý là biến đổi từ (8.19) sang (8.21) chỉ là dạng đơn giản chính thức của luật nền mà không làm thay đổi bậc của mô hình động, do $\mathbf{x}(k)$ là vectơ và X là tập mờ nhiều chiều. Gọi M là số tập mờ X_i xác định trạng thái $\mathbf{x}(k)$ và N là số tập mờ B_j định nghĩa ngõ vào $u(k)$. Giả sửlà luật nền gồm tất cả các khả năng tổ hợp của các tập X_i và B_j, thì số tổng các luật là $K = MN$. Toàn thể các luật có thể được biểu diễn thành bảng sau:

$\mathbf{x}(k)$	$u(k)$: B_1	B_2	$\ldots$	B_N
X_1	c_{11}	c_{12}	$\ldots$	c_{1N}
X_2	c_{21}	c_{22}	$\ldots$	c_{2N}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_M	c_{M1}	c_{M2}	$\ldots$	c_{MN}

(8.22)

Khi dùng toán tử t-norm, mức độ hoàn thành của luật tiền đề $\beta_{ij}(k)$ được tính theo:

$$\beta_{ij}(k) = \mu_{Xi}(\mathbf{x}(k)) \cdot \mu_{Bj}(u(k)) \tag{8.23}$$

Ngõ ra của mô hình $y(k+1)$ được tính theo trung bình của các hệ quả c_{ij} lượng hóa theo mức hoàn thành chuẩn hóa β_{ij} :

$$\begin{aligned} y(k+1) &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\beta_{ij}(k).c_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\beta_{ij}(k)} \\ &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\mu_{Xi}(x(k)).\mu_{Bj}(u(k)).c_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\mu_{Xi}(x(k)).\mu_{Bj}(u(k))} \end{aligned} \tag{8.25}$$

Thí dụ 8.1 Xét hệ mờ có dạng $y(k+1) = f(y(k), y(k-1), u(k))$
Trong đó dùng hai thừa số biến ngôn ngữ *{thấp, cao}* được dùng thay cho $y(k)$ và $y(k-1)$ và dùng ba thừa số *{bé, trung bình, lớn}* cho $u(k)$. Toàn bộ luật nền gồm $2 \times 2 \times 3 = 12$ luật:

Nếu $y(k)$ là *thấp* và $y(k-1)$ là *thấp* và $u(k)$ là *bé* **thì** $y(k+1)$ là c_{11}
Nếu $y(k)$ là *thấp* và $y(k-1)$ là *thấp* và $u(k)$ là *trung bình* **thì** $y(k+1)$ là c_{12}
. . .
Nếu $y(k)$ là *cao* và $y(k-1)$ là *cao* và $u(k)$ là *lớn* **thì** $y(k+1)$ là c_{43}

Trong thí dụ này $\mathbf{x}(k) = [y(k), y(k-1)]$, $Xi \in$ *{(thấp × thấp), (thấp × cao), (cao× thấp), (cao× cao) }*, $M = 4$ và $N = 3$. Luật nền được biểu diễn trong bảng sau:

$\mathbf{x}(k)$	$u(k)$: *small*	*medium*	*large*
X_1 *(low × low)*	c_{11}	c_{12}	c_{13}
X_2 *(low × high)*	c_{21}	c_{22}	c_{23}
X_3 *(high × low)*	c_{31}	c_{32}	c_{33}
X_4 *(high × high)*	c_{41}	c_{42}	c_{43}

(8.28)

Phương pháp chuyển ngược (inversion) đòi hỏi là hàm thành viên tiền đề μ_{Bj} $(u(k))$ có dạng tam giác và tạo một partition, tức là, hoàn thành (fulfill):

$$\sum_{j=1}^{N} \mu_{B_j}(u(k)) = 1 \tag{8.29}$$

Ý tưởng cơ bản là. Trong từng biến trạng thái $\mathbf{x}(k)$, thì pháp ánh xạ (multivariate mapping) (8.1) được rút lại thành (univariate mapping)

$$y(k + 1) = f_x(u(k)), \tag{8.30}$$

trong đó chỉ số dưới x cho thấy là f_x là cho trường hợp trạng thái đặc thù $\mathbf{x}$. Từ phép ánh xạ này, là dạng tuyến tính hóa từng phần, thì có thể dễ dàng tìm được phép ánh xạ ngược $u(k) = f^{-1}x\ (r(k + 1))$, cho thấy là mô hình có tính ngịch chuyển. Có thể kiểm tra tính nghịch (invertibility) cho trường hợp hàm (univariate functions). Đầu tiên, dùng (8.29), thì hàm ra của mô hình (8.25) đơn giản thành:

$$\begin{aligned} y(k+1) &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \mu_{Xi_i}(x(k)).\mu_{B_j}(u(k)).c_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} \mu_{Xi_i}(x(k)).\mu_{B_j}(u(k))} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N} \lambda_i(x(k)).\mu_{B_j}(u(k)).c_{ij} \\ &= \sum_{j=1}^{N} \mu_{B_j}(u(k)) \sum_{i=1}^{M} \lambda_i(x(k)).c_{ij} \end{aligned} \tag{8.31}$$

Trong đó $\lambda_i(\mathbf{x}(k))$ là mức độ hoàn thành chuẩn hóa của phần trạng thái trong tiền đề:

$$\lambda(x(k)) = \frac{\mu_{Xi}(x(k))}{\sum_{j=1}^{K} \mu_{Xi}(x(k))} \tag{8.33}$$

Khi có được trạng thái $\mathbf{x}(k)$, tính được tổng trong (8.31), ta có:

$$y(k+1) = \sum_{j=1}^{N} \mu_{B_j}(u(k)) c_j, \tag{8.34}$$

Trong đó:

$$c_j = \sum_{i=1}^{M} \lambda_i(x(k)).c_{ij}, \tag{8.36}$$

Đây là phương trình của mô hình singleton có ngõ vào $u(k)$ và ngõ ra $y(k + 1)$:

Nếu $u(k)$ là Bj **thì** $y(k + 1)$ là $c_j(k)$, $\quad j = 1, \ldots, N$. (8.37)

Từng luật trong các luật trên được nghịch chuyển bằng các chuyển đổi các tiền đề và hệ quả, từ đó có các luật sau:

Nếu $r(k+1)$ là $c_j(k)$ **thì** $u(k)$ là B_j $\quad j = 1, \ldots, N$. (8.38)

Trong đó tín hiệu tham chiếu $r(k+1)$ đã thay chỗ cho $y(k+1)$. Do $c_j(k)$ là singletons, nên cần có phép nội suy giữa các hệ quả $c_j(k)$ để tìm $u(k)$. Phép nội suy này được thực hiện dùng tập mờ C_j dùng hàm thành viên dạng tam giác:

$$\mu_{C1}(r) = \max\left(0, \min\left(1, \frac{c_2 - r}{c_2 - c_1}\right)\right) \tag{8.39a}$$

$$\mu_{Cj}(r) = \max\left(0, \min\left(\frac{r - c_{j-1}}{c_j - c_{j-1}}, \frac{c_{j+1} - r}{c_{j+1} - c_j}\right)\right) \tag{8.39b}$$

$$\mu_{CN}(r) = \max\left(0, \min\left(\frac{r - c_{N-1}}{c_N - c_{N-1}}, 1\right)\right) \tag{8.39c}$$

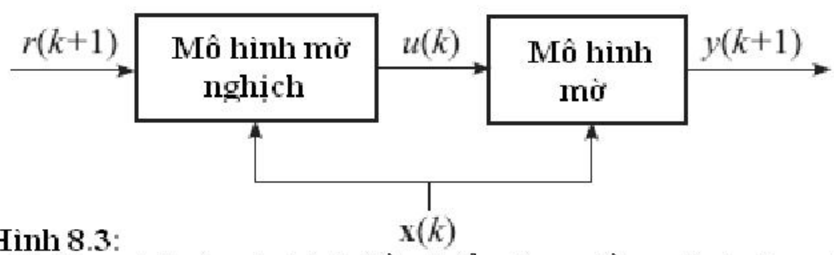
Ngõ ra của bộ điều khiển nghịch là:

$$u(k) = \sum_{j=1}^{N} \mu_{Cj}\big(r(k+1)\big) b_j, \tag{8.40}$$

Trong đó b_j là lõi (cores) của B_j . Phép nghịch cho bởi các phương trình (8.33), (8.39) và (8.40). Có thể kiểm nghiệm lại là kết nối nối tiếp giữa bộ điều khiển và mô hình nghịch, được vẽ ở hình 8.3, cho phép ánh xạ đơn vị (identity mapping) (điều khiển hoàn hảo)

$$y(k+1) = f_x\big(u(k)\big) = f_x\Big(f^{-1}\big(r(k+1)\big)\Big) = r(k+1), \tag{8.41}$$

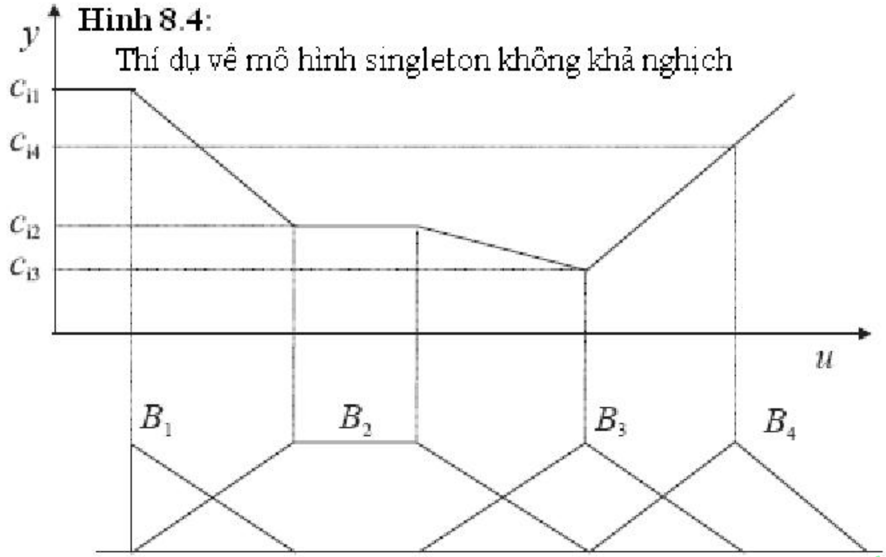
Nếu tồn tại $u(k)$ sao cho $r(k+1) = f(\mathbf{x}(k), u(k))$. Khi không tồn tại $u(k)$, thì sai biệt $\left| r(k+1) - f_x\Big(f_x^{-1}\big(r(k+1)\big)\Big) \right|$ phải càng bé càng tốt. Phần chứng minh xem như là bài tập cho độc giả.



Hình 8.3:
Nối tiếp mô hình mờ và bộ điều khiển dùng phần nghịch của mô hình

Bên cạnh việc tính toán mức độ hàm thành viên, cả mô hình và bộ điều khiển có thể được thiết lập dùng các phép tính toán ma trận và phép nội suy tuyến tính, làm cho thuật toán thích hợp cho các thiết lập trong thời gian thực.

Trong luật nền không khả nghịch (noninvertible rule base) (xem hình 8.4), có thể tìm được tập tín hiệu điều khiển bằng cách phân chia luật nền thành hai hay nhiều phần khả nghịch. Trong từng phần, tìm tác động điều khiển dùng phép nghịch đảo. Trong số các tác động điều khiển này, chỉ chọn được một, bằng cách đưa thêm vào một số tiêu chuẩn phụ, như điều kiện là tác động điều khiển là bé nhất. (thí dụ tối thiểu $u(k)$ hay $|u(k) - u(k-1)|$).



Hình 8.4:
Thí dụ về mô hình singleton không khả nghịch

Tính khả nghịch của mô hình mờ có thể được kiểm tra khi chạy, bằng cách kiểm tra tính đơn điệu của các hệ quả gộp chung c_j theo cores của tập mờ ngõ vào b_j, xem (8.36). Điều này là hữu ích do các mô hình phi tuyến có thể chỉ là không khả nghịch cục bộ, đưa đến một dạng ngoại lệ của thuật toán nghịch. Hơn nữa, trong các mô hình trực tuyến thì phép kiểm tra này là cần thiết.

Example 8.2 Xét mô hình mờ từ thí dụ 8.1, được lặp lại như sau:

$\mathbf{x}(k)$	$u(k)$		
	small	*medium*	*large*
$X_1(low \times low)$	c_{11}	c_{12}	c_{13}
$X_2(low \times high)$	c_{21}	c_{22}	c_{23}
$X_3(high \times low)$	c_{31}	c_{32}	c_{33}
$X_4(high \times high)$	c_{41}	c_{42}	c_{43}

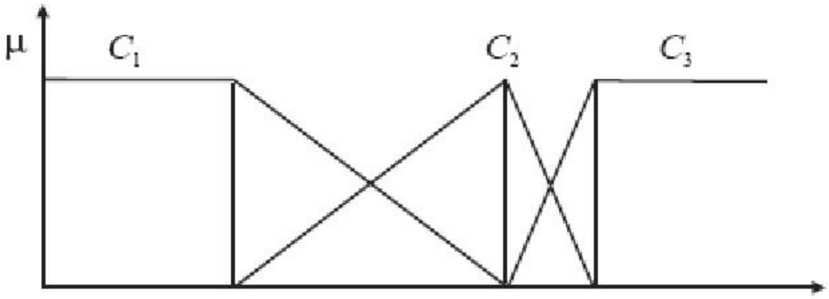
Cho trạng thái $\mathbf{x}(k) = [y(k), y(k-1)]$, mức độ hoàn thành của tiền đề đầu tiên "$\mathbf{x}(k)$ 'là X_i", được tính như là $\mu_{Xi}(\mathbf{x}(k))$. Trường hợp X_2, thì $\mu_{X2}(\mathbf{x}(k)) = \mu_{\text{low}}(y(k)) \cdot \mu_{\text{high}}(y(k-1))$. Dùng (8.36), có được cores $c_j(k)$:

$$c_j(k) = \sum_{i=1}^{4} \mu_{Xi}(x(k)) c_{ij}, \qquad j = 1, 2, 3 . \tag{8.42}$$

Thí dụ, hàm thành viên của tập mờ C_j , lấy từ (8.39), được cho ở hình 8.5:
Giả sử là $b_1 < b_2 < b_3$, mô hình là khả nghịch (cục bộ) nếu $c_1 < c_2 < c_3$ hay nếu $c_1 > c_2 > c_3$. Trường hợp này, có được càc luật sau:

1) **Nếu** $r(k+1)$ là $C_1(k)$ **thì** $u(k)$ là B_1
2) **Nếu** $r(k+1)$ là $C_2(k)$ **thì** $u(k)$ là B_2
3) **Nếu** $r(k+1)$ là $C_3(k)$ **thì** $u(k)$ là B_3

Nói cách khác, nếu mô hình không khả nghịch, tức là, $c_1 > c_2 < c_3$, thì các luật trên phải được chia ra thành hai luật nền. Luật đầu chứ luật 1 và 2, và luật hai chứa luật 2 và 3.



Hình 8.5: $c_1(k)$ $c_2(k)$ $c_3(k)$
Các partition mờ tạo ra từ $c_1(k)$, $c_2(k)$ và $c_3(k)$.

1.4 Mô hình nghịch dùng các khâu trễ

Khi mô hình có các khâu trễ tại ngõ vào $y(k+1) = f(\mathbf{x}(k), u(k-n_d))$, không dùng được phép nghịch một cách trực tiếp, mà cần làm trễ các tác động điều khiển $u(k)$ đi n_d bước thời gian. Để có thể tạo ra được tín hiệu $u(k)$ thích hợp, thì cần chuyển mô hình đi trước $n_d - 1$, thí dụ $u(k) = f^{-1}(r(k+n_d+1), \mathbf{x}(k+n_d))$, trong đó

$$\mathbf{x}(k+n_d) = [y(k+n_d), \ldots, y(k+1), \ldots \\ y(k-n_y+n_d+1), u(k-1), \ldots, u(k-n_u+1)]^T. \quad (8.44)$$

Các giá trị ẩn, $y(k+1), \ldots, y(k+n_d)$, được dự báo hồi quy dùng mô hình:

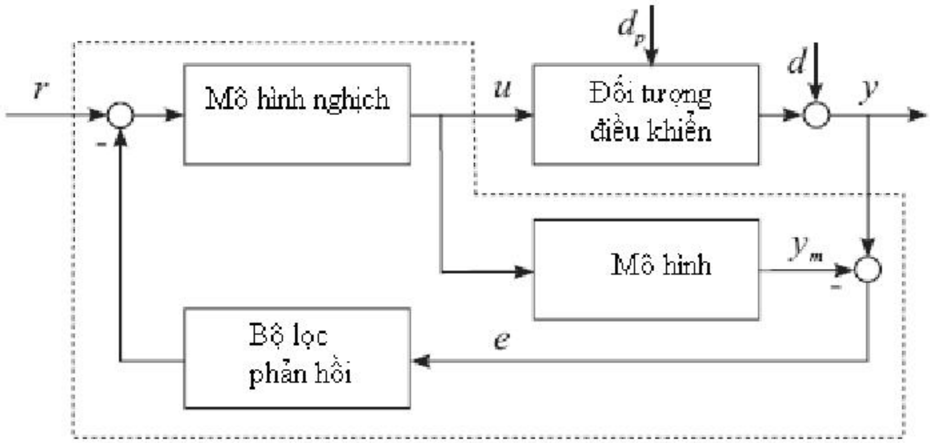
$$y(k+i) = f(\mathbf{x}(k+i-1), u(k-nd+i-1)), \\ \mathbf{x}(k+i) = [y(k+i), \ldots, y(k-n_y+i+1), u(k-n_d+i-1), \ldots \\ u(k-n_u-n_d+i+1)]^T \quad (8.46)$$

với $i = 1, \ldots, n_d$.

1.5 Điều khiển dùng mô hình nội tại

Nhiễu tác động lên quá trình, nên nhiễu đo được và mô hình không còn khớp với đối tượng, tạo sai lệch giữa ngõ ra mô hình và đối tượng. Trong điều khiển vòng hở, điều này làm sai số giữa tín hiệu tham chiếu và ngõ ra của quá trình. Sơ đồ *điều khiển dùng mô hình nội tại* IMC (Economou, et al., 1986) là một phương thức để bổ chính sai số này.

Hình 8.6 minh họa sơ đồ IMC, gồm ba khâu: khâu điều khiển lấy từ mô hình ngược của đối tượng, và bản thân mô hình, cùng với khâu lọc phản hồi. Khâu điều khiển (đường vạch) có hai ngõ vào, tín hiệu tham chiếu và đo lường tại ngõ ra của quá trình và một ngõ ra là tín hiệu điều khiển.



Hình 8.5: Sơ đồ điều khiển dùng mô hình nội tại

Mục đích của mô hình mắc song song với đối tượng điều khiển là nhằm trừ bớt ảnh hưởng của tác động điều khiển từ ngõ ra của quá trình. Nếu ngõ ra dư báo và ngõ ra của quá trình bằng nhau, thì sai số *e* bằng không và bộ điều khiển hoạt động theo cấu hình vòng hở. Nếu nhiễu *d* tác động lên ngõ ra của quá trình, tín hiệu phản hồi *e* là bằng với ảnh hưởng của nhiễu và không ảnh hưởng lên tác động điều khiển. Tín hiệu này bị trừ với tín hiệu tham chiếu. Khi có mô hình đối tượng hoàn hảo, thì sơ đồ IMC có khả năng tiệt tiêu ảnh hưởng của nhiễu cộng tại ngõ vào chưa đo được.

Bộ lọc phản hồi được đưa vào nhằm lượt bỏ nhiễu đo lường được và ổn định vòng thông qua việc giảm bớt độ lợi vòng tại vùng tần số cao. Trong các hệ thống phi tuyến và mô hình, bộ lọc này phải được thiết kế dùng kinh nghiệm.

2. Điều khiển dùng mô hình dự báo

Điều khiển dùng mô hình dự báo (Model-based predictive control: MBPC) là phương pháp tổng quát nhằm giải quyết các bài toán điều khiển trong miền thời gian, và dựa trên ba ý niệm cơ bản:

1. Mô hình được dùng để dự báo các ngõ ra của quá trình tại các bước thời gian rời rạc trong tương lai, trong vùng *chân trời dự báo* (*prediction horizon*).

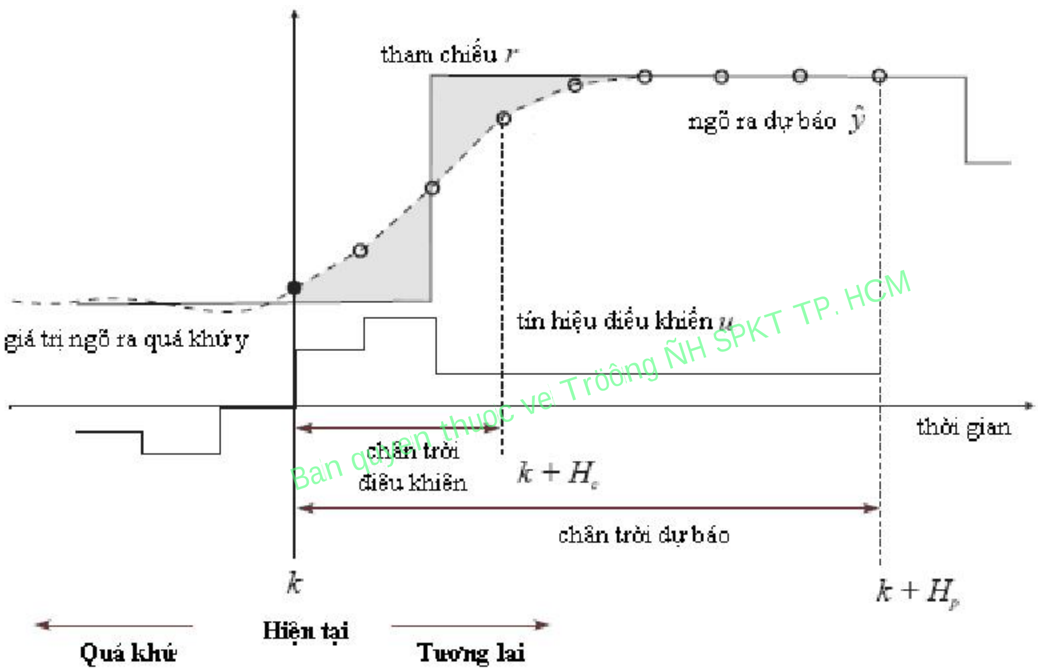
2. Chuỗi các tín hiệu điều khiển tương lai được tính toán trong *chân trời điều khiển* (*control horizon*) bằng cách tối thiểu hóa hàm mục tiêu cho trước.

3. Chỉ đưa tín hiệu điều khiển đầu tiên của chuỗi, thì chân trời được di chuyển về hướng tương lai và quá trình tối ưu hóa đươc lặp lại, điều này được gọi là nguyên tắc chân trời lùi dần (*receding horizon*).

Nhờ hướng tối ưu hóa và sử dụng mô hình tường minh của đối tượng, nên MBPC có thể dùng trong điều khiển tối ưu nhiều biến, giải quyết các quá trình phi tuyến, và có thể giải quyết hiệu quả các yếu tố ràng buộc.

2.1 Dự báo và chân trời điều khiển

Ngõ ra tương lai của quá trình được dự báo trong suốt chân trời dự báo (*prediction horizon*) H_p dùng mô hình của quá trình. Giá trị ngõ ra dự báo, gọi là $\hat{y}(k+1)$ cho các $i = 1, \ldots, H_p$, phụ thuộc vào trạng thái của quá trình tại thời gian hiện tại k và tín hiệu điều khiển sắp tới $u(k+i)$ với $i = 0, \ldots, H_c - 1$, với $H_c \le H_p$ là chân trời điều khiển (*control horizon*). Tín hiệu điều khiển chỉ được tính toán trong chân trời điều khiển và giữa không đổi sau đó, tức là $u(k+i) = u(k + H_c - 1)$ với $i = H_c, \ldots, H_p - 1$, xem hình 8.7.



Hình 8.7. Nguyên lý cơ bản của điều khiển dùng mô hình dự báo

2.2 Hàm mục tiêu

Chuỗi các tín hiệu điều khiển $\mathbf{u}(k+i)$ với $i = 0, 1, \ldots, H_c - 1$ thường được tính bằng phương pháp tối ưu hóa hàm chi phí quân phương (Clarke, et al., 1987):

$$J = \sum_{i=1}^{Hp} \left\| r(k+i) - \hat{y}(k+i \right\|_{Pi}^{2} + \sum_{i=1}^{Hc} \left\| (\Delta u(k+i-1)) \right\|_{Qi}^{2} \qquad (8.48)$$

Thừa số đầu tiên được dùng để tối thiểu hóa phương sai (variance) của ngõ ra quá trình với tín hiệu tham chiếu, thừa số thứ hai biểu diễn hàm phạt cho tự thân u. $\mathbf{P}_i$ và $\mathbf{Q}_i$ là ma trận trọng số được định nghĩa là dương nhằm miêu tả tầm quan trọng của từng thừa số lẫn nhau trong các bước dự báo của (8.48). Các thừa số phụ có thể được thêm vào trong hàm chi phí để tính toán với các tiêu chí điều khiển khác.

Đối với các hệ thống có vùng chết n_d mẫu, chỉ có ngõ ra tại các thời điểm từ $k + n_d$ là được xem xét trong hàm mục tiêu, do các ngõ ra trước các thời gian này không chịu ảnh hưởng của tín hiệu điều khiển $u(k)$. Lý luận tương tự cho trường hợp các hệ có pha không tối thiểu.

Các ràng buộc "cứng" ("Hard") thí dụ mức và tốc độ của ràng buộc của tín hiệu điều khiển, ngõ ra quá trình, hay các biến khác có thể xem là một phần của bài toán tối ưu:

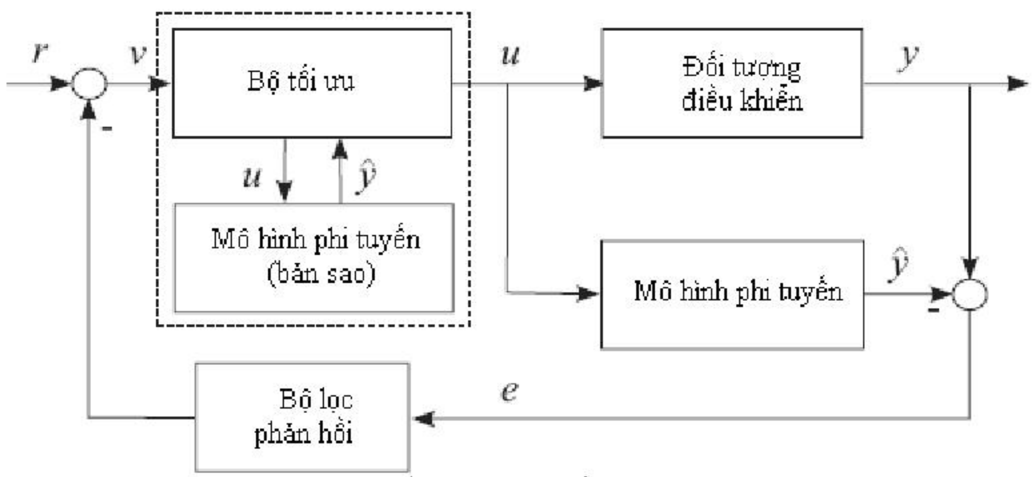
$$\begin{aligned} &u^{\min} \le u \le u^{\max} \\ &\Delta u^{\min} \le \Delta u \le \Delta u^{\max} \\ &y^{\min} \le y \le y^{\max} \\ &\Delta y^{\min} \le \Delta y \le \Delta y^{\max} \end{aligned} \tag{8.50}$$

Các biến có chỉ số trên min và max lần lượt là biên dưới và biên trên của tín hiệu.

2.3 Nguyên lý chân trời lùi dần

Chỉ có tín hiệu điều khiển $u(k)$ được đưa vào quá trình. Trong bước thời gian kế tiếp, tồn tại ngõ ra của quá trình $y(k + 1)$ và có thể lặp lại các dự báo và phép tối ưu hóa với các giá trị cập nhật được. Điều này được gọi là nguyên lý chân trời lùi dần (receding horizon principle). Tín hiệu điều khiển $u(k + 1)$ được tính tại bước thời gian $k + 1$ thường sẽ khác với tín hiệu tín tại bước thời gian k, do có thên nhiều thông tin hơn về quá trình. Ý niệm này tương tự như chiến lược điều khiển vòng hở đã thảo luận trong phần 8.1. Đồng thời mô hình có thể dùng độc lập với quá trình, như trong trường hợp điều khiển vòng hở đúng nghĩa.

Mạng nơrôn hay hệ mờ hoạt động như bộ dự báo số học của ngõ ra quá trình và có thể được tích hợp trực tiếp vào trong sơ đồ MBPC như vẽ ở 8.8. Sơ đồ IMC thường được dùng để bổ chính yếu tố nhiễu và sai số mô hình hóa, xem thêm phần 8.1.5.



Hình 8.8. Mô hình phi tuyến trong sơ đồ MBPC dùng mô hình nội tại và phản hồi để bổ chính yếu tố nhiễu và sai số mô hình hóa.

2.4 Tối ưu hóa trong phương pháp MBPC

Tối ưu hóa (8.48) thường cần có phương pháp tối ưu hóa phi tuyến không lồi (non-convex). Cần phân biệt một số xu hướng chính sau.

Thuật toán tối ưu hóa theo bước lặp Xu hướng này bao gồm các phương pháp như phương pháp Nelder-Mead hay phương pháp lập trình quân phương tuần tự (sequential quadratic programming SQP). Đối với các chương trình điều khiển dài hơn (*Hc*), các thuật toán này thường hội tụ về cực tiểu cục bộ. Điều này làm xấu kết quả của bài toán tối ưu hóa và hệ quả là làm xấu hiệu năng của bộ điều khiển dự báo. Một phương thức sửa chữa từng phần là tìm tốt được nghiệm ban đầu, thì dụ dùng phương pháp tìm kiếm lưới (grid search) (Fischer and Isermann, 1998). Tuy nhiên, phương pháp này chỉ hiệu quả trong các bài toán có kích thước bé.

Kỹ thuật tuyến tính hóa. Một hướng có thể thực hiện được trong xu hướng NPC là tuyến tính hóa mô hình phi tuyến tại mỗi bước lấy mẫu và dùng mô hình tuyến tính hóa này trong các sơ đồ điều khiển dự báo chuẩn (Mutha, et al., 1997; Roubos, et al., 1999). Tùy thuộc vào các phương pháp tuyến tính hóa đặc thù, mà có thể dùng nhiều hướng khác nhau như sau:

Tuyến tính hóa dùng bước đơn Mô hình phi tuyến được tuyến tính hóa trong bước thời gian hiện tại *k* và có được mô hình tuyến tính dùng trong suốt chân trời dự báo. Phương pháp này cho thiết lập dễ và nhanh. Tuy nhiên, trong các quá trình có tính phi tuyến cao cùng với chân trời dự báo dài, phương pháp tuyến tính hóa đơn bước thường cho kết quả không tốt. Yếu điểm này được giải quyết dùng phương pháp tuyến tính hóa theo nhiều bước.

Tuyến tính hóa theo nhiều bước Mô hình phi tuyến được tuyến tính hóa lần đầu tại bước thời gian *k*. Tín hiệu điều khiển có được là $\mathbf{u}(k)$ được dùng trong dự báo cho $\hat{y}(k+1)$ và mô hình phi tuyến được tuyến tính hóa ,ần nữa xung quanh điểm làm việc sắp tới. Lặp lại thủ tục này nhiều lần co đến $k + H_p$. Theo phương pháp này thì mức xấp xỉ mo hình phi tuyến càng chính xác, đặc biệt trong trường hợp chân trời dài. Chi phí quan trọng là khối lượng tính toán lớn.

Cả trường hợp tuyến tính hóa đơn bước và đa bước, thì cần có bước hiệu chỉnh (correction step) dùng một vectơ nhiễu (Peterson, et al., 1992). Đối với mô hình tuyến tính hóa, thì tìm được nghiệm tối ưu (8.48) dùng chương trình sau:

$$\min_{\Delta u}\left\{\frac{1}{2}\Delta u^T H \Delta u + c^T \Delta u\right\} \tag{8.51}$$

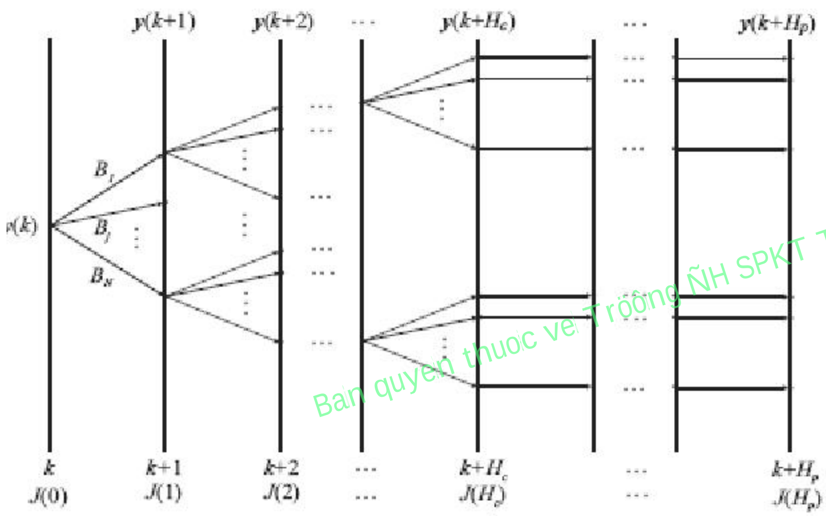
Trong đó:

$$\begin{cases} H = 2\left(R_u^T P R_u + Q\right) \\ c = 2\left[R_u^T P^T (R_X A_X(k) - r + d)\right]^T \end{cases} \tag{8.52}$$

Các ma trận $\mathbf{R}_u$, $\mathbf{R}_x$ và $\mathbf{P}$ được cấu trúc từ ma trận của hệ thống tuyến tính hóa và từ mô tả của các ràng buộc. Nhiễu $\mathbf{d}$ có thể được tính cho sai số tuyến tính hóa khi có sai biệt giữa ngõ ra của mô hình phi tuyến và mô hình tuyến tính hóa.

Tuyến tính hóa phản hồi Kỹ thuật tuyến tính hóa phản hồi (chính xác và xấp xỉ) cũng dùng được cho hệ NPC. Có hai khác biệt cơ bản giữa tuyến tính hóa phản hồi phương pháp tuyến tính hóa dùng hai điểm làm việc (two operating-point linearization) như sau:

- Quá trình tuyến tính hóa phản hồi có đặc tính động không đổi theo thời gian. Đây không phải là trường hợp quá trình được tuyến tính hóa tại điểm làm việc, Như thế, thì việc tinh chỉnh bộ điều khiển dự báo về sau này sẽ gặp khó khăn.
- Tuyến tính hóa phản hồi biến đổi ràng buộc ngõ vào theo phương thức phi tuyến. Đây rõ ràng là một khuyết điểm, do chương trình quadratic program (8.51) cần có các ràng buộc tuyến tính. Một số nghiệm của bài toán này đã được đề nghị (Oliveira, et al., 1995; Botto, et al., 1996).



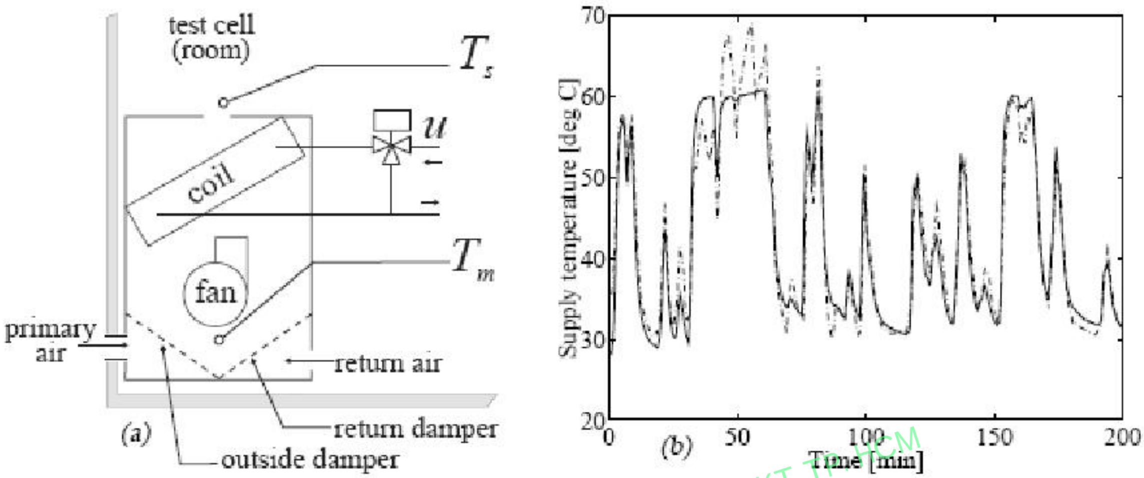
Hình 8.9 Phương pháp tìm kiếm dạng cây dùng trong điều khiển dư báo

Kỹ thuật tìm kiếm rời rạc Một hướng khác được dùng trong tối ưu hóa NPC trên cơ sở kỹ thuật tìm kiếm rời rạc như lập trình động (dynamic programming: DP), branch-and-bound (B&B) methods (Lawler and Wood, 1966; Sousa, et al., 1997), thuật toán di truyền (GAs) (Onnen, et al., 1997),v.v,... Ý tưởng cơ bản là rời rạc hóa không gian của tín hiệu điều khiển và dùng phương pháp tìm kiên thông minh đề tìm nghiệm cận tối ưu toàn cục trong không gian này. Hình 8.9 minh họa ý tưởng cơ bản này trong không gian rời rạc N (*N* alternatives):

$$\mathbf{u}(k+i-1) \in \{\omega j \mid j = 1, 2, \ldots, N\}.$$

Rõ ràng là số nghiệm có thể có tăng theo dạng hàm mủ với H_c và nhiều mánh lới đã được dùng trong các phương pháp khác nhau. Phương pháp lập trình động dựa trên yếu tố lưu trữ các nghiệm tối ưu trung gian trong bộ nhớ. Phương pháp B&B dùng các biên trên và dưới của nghiệm nhằm cắt các nhánh không dẫn đến nghiệm tối ưu. Thuật toán di truyền tìm kiếm trong không gian với phương thức ngẫu nhiên.

Thí dụ 8.3 (Điều khiển một đơn vị máy điều hòa không khí) Điều khiển dự báo nhiệt độ phi tuyến trong hệ máy điều hòa không khí (Sousa, et al., 1997) được minh họa như một thí dụ. Bộ điều khiển dự báo phi tuyến được phát triển để điều khiển nhiệt độ của cuộn dây quạt, là một phần trong hệ thống điều khòa nhiệt độ. Nước nóng hay lạnh được cấp vào cuộn day qua một van. Trong đơn vị, không khí bên ngoài được trộn lại và tạo không khí đưa về phòng. Không khí hỗn hợp này được quạt thổi qua cuộn dây và nóng lên hay nguội xuống (hình 8.10a).

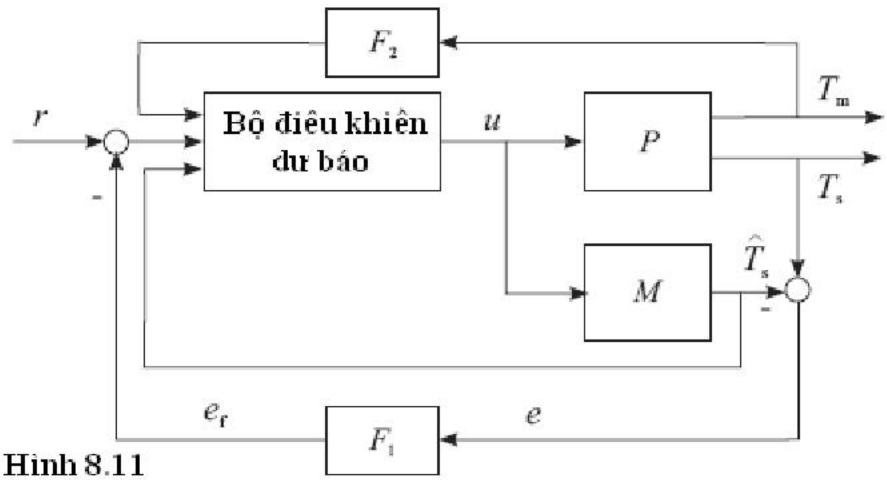


Hình 8.10 Đơn vị điều hòa nhiệt độ (a) và đánh giá của mô hình TS (ngõ ra đo được - đường sậm, ngõ ra của mô hình - đường đứt nét)

Quá trình này có tính phi tuyến cao (do đặc tính của van) và rất khó để mô hình hóa theo phương pháp mechanistic. Dùng phương pháp nhận dạng phi tuyến, ta có thể có được mô hình chính xác trong một thời gian ngắn. Trong nghiên cứu được báo cáo (Sousa, et al., 1997), xây dựng một mô hình mờ TS từ đo lường ngõ ra dùng phương pháp xâu chuỗi mờ (fuzzy clustering). Mô hình này dự báo nhiệt độ cung cấp T dùng các luật có dạng:

Nếu $\hat{T}_S(k)$ là A_{i1} **và** $T_m(k)$ là A_{i2} **và** u(k) là A_{13} **và** u(k − 1) là A_{14}
thì $\hat{T}_S(k+1) = a_i^T\left[\hat{T}_S(k)T_m(k)u(k)u(k-1)\right]^T + b_i$

Dữ liệu nhận dạng chứa 800 mẫu, lấy được từ hai thời điểm khác nhau trong ngày (buổi sáng và buổi trưa). Thời gian lấy mẫu là 30 giây. Tín hiệu kích thích gồm có nhiều tín hiệu sin với năm tần số và biên độ khác nhau, và xung với biên độ và độ rộng ngẫu nhiên. Tập dữ liệu riêng biệt, được đo trong một ngày khác được dùng để đánh giá mô hình. Hình 8.10b so sánh nhiệt độ cung cấp đo được và nhiệt độ dự báo để qui từ mô hình.



Hình 8.11
Thiết lập bộ điều khiển dự báo mờ cho cuộn dây quạt dùng cấu trúc IMC

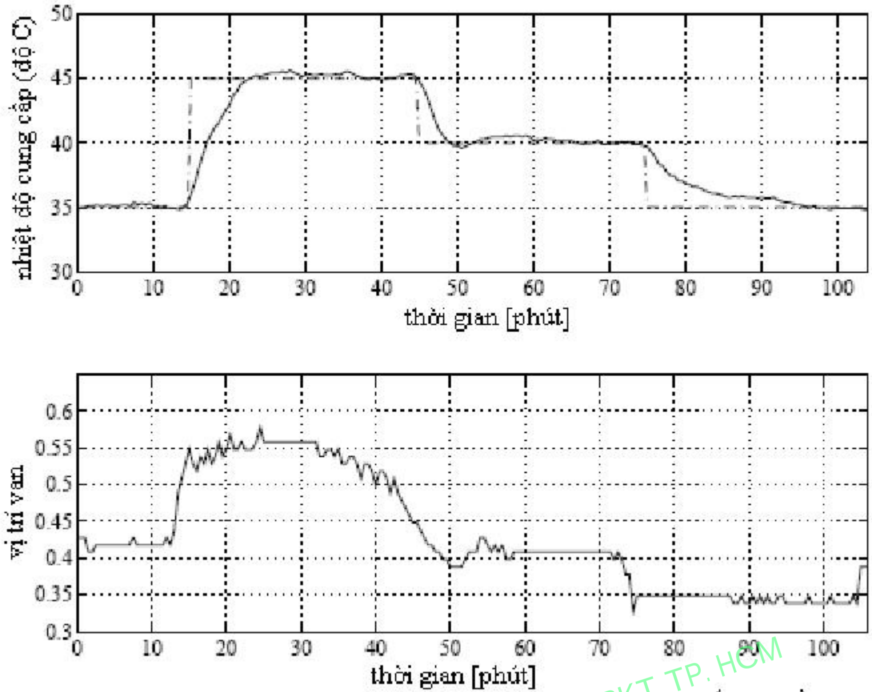
Một bộ điều khiển dùng mô hình dự báo được thiết kế theo phương pháp B&B. Bộ điều khiển dùng mô hình IMC ở hình 8.11 được dùng bổ chính cho sai số mô hình và nhiễu. Các ngõ vào bộ điều khiển là điểm thiết lập (setpoint), nhiệt độ cung cấp dự báo $\hat{T}_S$, và nhiệt độ hỗn hợp đã lọc *Tm*. Tín hiệu sai số, $e(k) = T_S(k) - \hat{T}_S(k)$, được đưa qua bộ lọc thông thấp số bậc nhất F_1. Một bộ lọc tương tự F_2 được dùng lọc T_m. Các bộ lọc này đều thiết kế theo dạng lọc Butterworth, có tần số cắt được chỉnh định theo kinh nghiệm, lấy từ mô phỏng, nhằm có được bộ lọc đáng tin cậy lọc được nhiễu, và cho đáp ứng nhanh. Hình 8.12 vẽ một một kết quả có được trong thời gian thực với H_c = 2 và H_p = 4.

3. Điều khiển thích nghi

Các quá trình có đáp ứng thay đổi theo thời gian không thể điều khiển tốt dùng các bộ điều khiển có tham số cố định. *Điều khiển thích nghi* (*Adaptive control*) là phương pháp điều khiển mà tham số được tinh chỉnh trực tuyến để duy trì các tính năng của hệ thống khi có sự thay đổi trong quá trình. Có nhiều phương pháp thiết kế bộ điều khiển thích nghi, và có thể được chia thành hai nhóm chính:

- *Điều khiển thích nghi gián tiếp (Indirect adaptive control)*. Mô hình điều khiển được thích ứng trực tuyến và các tham số điều khiển được rút ra từ tham số của mô hình.
- *Điều khiển thích nghi trực tiếp (Direct adaptive control)*. Không dùng mô hình, tham số điều khiển được cập nhật trực tiếp

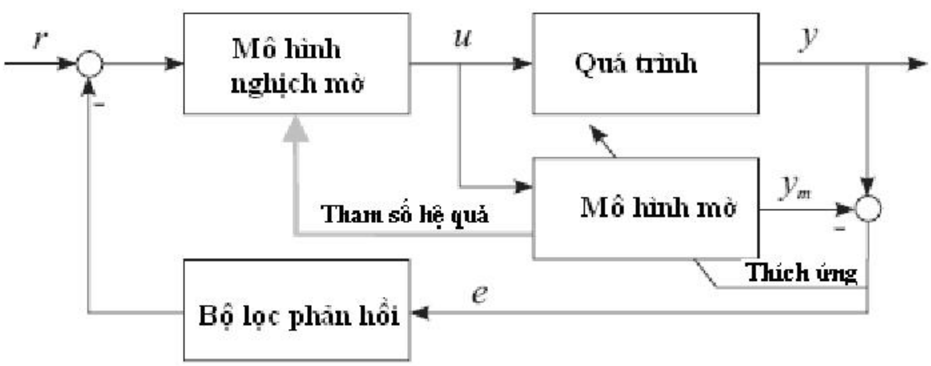
Phần tiếp sẽ trình bày các thí dụ vụ về các phương pháp điều khiển vừa nêu.



Hình 8.12 Đáp ứng trong thời gian thực của hệ thống điều hòa nhiệt độ, Ngõ ra - đường sậm; tín hiệu tham chiếu -đường đứt nét

3.1 Điều khiển thích nghi gián tiếp

Có thể dùng phương pháp chỉnh định trực tuyến (on-line adaptation) để giải quyết yếu tố chưa khớp giữa đối tượng và mô hình. Trong nhiều trường hợp, yếu tố không khớp xuất hiện như là hệ quả của các thay đổi (tạm thời). Chỉnh định trực tuyến còn dùng được để giải quyết yếu tố không khớp giữa quá trình và các tham số quá trình. Để giải quyết các hiện tượng này, đặc biệt nếu có ảnh hưởng của yếu tố thay đổi theo thời gian, có thể chỉnh định mô hình ngay trong vòng điều khiển. Do tác động điều khiển được suy ra từ việc làm nghịch mô hình một cách trực tuyến, nên bộ điều khiển được chỉnh định một cách tự động. Hình 8.13 minh họa sơ đồ IMC với phép thích ứng trực tuyến các tham số hệ quả trong bộ điều khiển mờ.



Hình 8.13: Sơ đồ điều khiển thích nghi dùng mô hình

Do ngõ ra của mô hình từ (8.25) có dạng tuyến tính theo các tham sô hệ quả, nên có thể dùng thuật toán bình phương tối thiểu đệ qui (*recursive least-squares algorithms*) để ước lượng các tham số hệ quả từ dữ liệu. Giả sử là các luật của mô hình mờ cho bởi

(8.19) và các tham số hệ quả được đánh theo chỉ số tuân tự theo luật số. Vectơ cột các hệ quả được cho bởi $\mathbf{c}(k) = [c_1(k), c_2(k), \ldots, c_K(k)]^T$,
trong đó K là số luật. Mức độ hoàn thành chuẩn hóa được cho bởi:

$$\gamma_i(k) = \frac{\beta_i(k)}{\sum_{j=1}^{K} \beta_j(k)}, \quad i = 1,2,\ldots,K. \tag{8.54}$$

Sắp xếp vectơ cột $\boldsymbol{\gamma}(k) = [\gamma_1(k), \gamma_2(k), \ldots, \gamma_K(k)]^T$. Vectơ hệ quả $\mathbf{c}(k)$ được cập nhật đệ qui từ:

$$c(k) = c(k-1) + \frac{P(k-1)\gamma(k)}{\lambda + \gamma^T(k)P(k-1)\gamma(k)}[y(k) - \gamma^T(k)c(k-1)], \tag{8.55}$$

Trong đó λ là thừa số quên không đổi (constant forgetting factor) gây ảnh hưởng lên khả năng bám theo của thuật toán thích ứng. Khi λ càng bém thì cập nhật các tham số hệ quả càng nhanh, tuy nhiên thuêt toán lại nhạy cảm với nhiễu. Như thế, việc chọn lựa λ là bài toán phụ thuộc. Ma trận đồng phương sai (covariance matrix) $\mathbf{P}(k)$ được cập nhật theo:

$$P(k) = \frac{1}{\lambda}\left[P(k-1) - \frac{P(k-1)\gamma(k)\gamma^T(k)P(k-1)}{\lambda + \gamma^T(k)P(k-1)\beta(k)}\right]. \tag{8.56}$$

Đồng phương sai đầu tiên thường được chọn là $\mathbf{P}(0) = \alpha \cdot \mathbf{I}$, trong đó $\mathbf{I}$ là ma trân đơn vị $K \times K$ và α là hằng số dương có giá trị lớn.

3.2 Học tăng cường

Học tăng cường (reinforcement learning: RL) xuất phát từ nguyên lý học của người và sinh vật. Khi ứng dụng vào điều khiển, RL không cần mô hình tường minh về đối tượng điều khiển. Hơn nữa, việc ước lượng các tính năng điều khiển, yếu tố tăng cường (*the reinforcement*, có thể hơn thô bạo (crude) (thí dụ như tín hiệu nhị phân cho thấy là thành công hay thất bại) và có thể liên quan đến toàn chuỗi tác động điều khiển. Điều này khác với phương thức học có giám sát (*supervised learning*) theo đó tín hiệu sai biệt cho hoàn toàn thông tin về biên độ và dấu của sai biệt giữa ngõ ra thực và ngõ ra tham chiếu.

Thí dụ 8.4 Con người có khả năng tối ưu hành vi trong từng môi trường cụ thể. Nhiều nhiệm vụ học bao gồm các bước thử lặp lại nhiều lần qua các yếu tố thưởng hay phạt. Mỗi lần thử có thể là một chuỗi động các hành động trong khi qua 1 trị đánh giá (reinforcement) chỉ nhận được vào phút cuối.

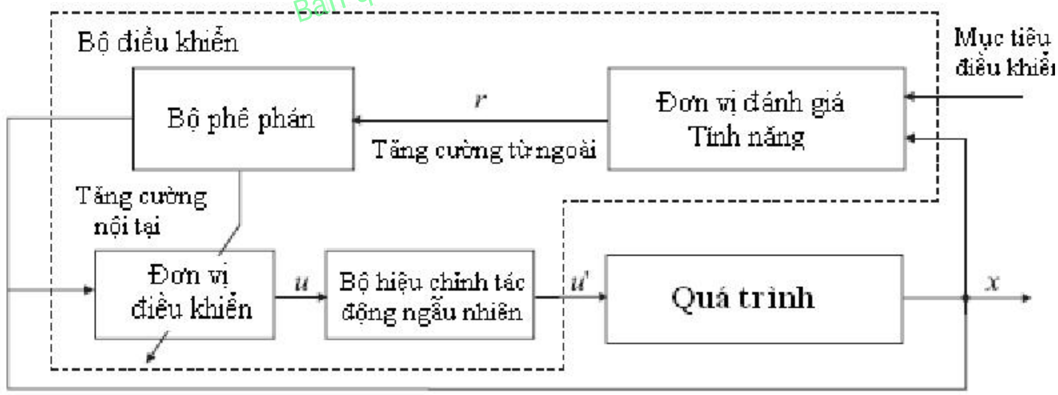
Thí dụ, bạn muốn học đánh tennis. Thử nghiệm điều khiển là bạn muốn đánh đúng vào banh. Trong trường hợp học có giám sát bạn sẽ cần đến giáo viên nhằm đánh giá khả năng của bạn trong các thời gian và cho bạn biết là bạn cần thay đổi chiến lược để tự cải thiện mình. Huấn luyện viên có thể giải thích chi tiết về phương thức thay đổi cách đánh, phương thức tiếp cận với banh, v.v,..

Trong phương pháp học tăng cường (reinforcement learning) thì khác, nhiệm vụ của giáo viên là chỉ cho bạn biết là cú đánh là OK (thưởng) hay không (phạt), và cho bạn khả năng xác định phương thức sửa chữa phù hợp nhất cho chiến lược của mình.

Điều quan trọng là sau mỗi phép thử là một chuỗi động các tác động (hướng banh, chuẩn bị và đánh banh) trong khi tác động tăng cường thực tế chỉ nhận được vào phút cuối. Như thế, một số lượng lớn các phép thử có thể là cần thiết để tìm ra được tác động nào là đúng và tác động nào phải hiệu chỉnh lại.

Mục tiêu của học tăng cường RL là nhằm phát hiện ra chiến lươc điều khiển nhằm tối đa hóa tác động tăng cường (thưởng) nhận được. Do không có giáo viên hay người giám sát từ ngoài để đ1nh giá tác động điều khiển, RL dùng bộ đánh giá nội tại được gọi là phê phán (*critic*). Vai trò của phê phán là dự báo kết quả của từng tác động điều khiển trong từng trạng thái của quá trình.

Chiến lược điều khiển là chỉnh định dùng phương pháp khám phá, tức là cân nhắc về thay đổi của tác động điều khiển do bộ điều khiển tính toán được và thông qua so sánh với yếu tố tăng cường nhận được với từng dự báo do bộ phê phán tạo ra. Sơ đồ khối một bộ RL cổ điển được vẽ ở hình 8.14 (Barto, et al., 1983; Anderson, 1987), gồm có đơn vị đánh giá tính năng, bộ phê phán, đơn vị điều khiển và bộ bổ chính tác động ngẫu nhiên.



Hình 8.14 Sơ đồ học tăng cường

Quá trình học trong sơ đồ RL thực hiện trong thời gian rời rạc. Gọi k là thời gian hiện tại, hệ thống được điều khiển dùng phương trình chuyển trạng thái sau:

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), u(k)), \qquad (8.57)$$

trong đó f là hàm ẩn. Để đơn giản ta chỉ xét hệ một ngõ vào, một ngõ ra.

Đơn vị đánh giá tính năng. Khối cung cấp tín hiệu học tăng cường từ ngoài (*external reinforcement)* $r(k)$ thường được giả sử là có hai giá trị:

$$r(k) = \begin{cases} 0 & safistied \\ -1 & failure \end{cases} \qquad (5.58)$$

Khối phê phán. Nhiệm vụ của phê phán là dự báo tín hiệu tăng cường sắp đến r mà quá trình nhận trong trang thái hiện tại tùy theo chiến lược điều khiển hiện tại.
Dự báo này được dùng để có được nhiều tín hiệu mang thông tin, được gọi là tăng cường nội tại (*internal reinforcement)*, có liên quan đến quá trình thích ứng bộ phê phán và bộ điều khiển.

Trong nhiệm vụ học động, tác động điều khiển không thể xét đoán riêng lẽ do từ các đặc tính động của quá trình. Không biết được là tín hiệu điều khiển đặc thù nào tạo ra được trạng thái đặc thù nào. Điều này đưa đến bài toán gọi là credit assignment problem (Barto, et al., 1983). Mục tiêu là tối đa hóa yếu tố tăng cường tổng trong suốt thời gian, và có thể được biểu diễn theo tổng của các tín hiệu tăng cường bên ngoài (tức thời).

$$V(k)=\sum_{i=k}^{\infty}\gamma^{i-k}r(i) \tag{8.59}$$

where $\gamma \in [0,\ 1)$ là thừa số discounting dạng mủ, r là tín hiệu tăng cường từ ngoài, k là thời gian rời rạc, và $V(k)$ là tổng (discounted sum) của các tín hiệu tăng cường sắp tới thường được gọi là hàm giá trị (*value function)*.

Bộ phê phán được huấn luyện để dự báo hàm giá trị tương lai $V(k+1)$ của trạng thái hiện tại của quá trình $\mathbf{x}(k)$ và tín hiệu điều khiển $u(k)$. Gọi $\hat{V}(k)$ là dự báo của $V(k)$. Để tìm luật phê phán, viết lại phương trình (8.59):

$$V(k)=\sum_{i=k}^{\infty}\gamma^{i-k}r(i)=r(k)+\gamma V(k+1) \tag{8.60}$$

Để huấn luyện bộ phê phán, cần tính sai số dự báo $\Delta(k)=V(k)-\hat{V}(k)$. Giá trị thực của hàm giá trị $V(k)$ là chưa biết, nhưng có thể xấp xỉ được bằng cách thay thế sai số dự báo:

$$\Delta(k)=V(k)-\hat{V}(k)=r(k)+\gamma\hat{V}(k+1)-\hat{V}(k) \tag{8.61}$$

Do $\Delta(k)$ được tính toán dùng hai giá trị liên tiếp nhau $\hat{V}(k)$ và $\hat{V}(k+1)$, nên được gọi là sai biệt tạm thời (*temporal difference)* (Sutton, 1988). Chú ý là cả $\hat{V}(k)$ và $\hat{V}(k+1)$ đều được biết tại thời điểm k, và do $\hat{V}(k+1)$ là dự báo có được từ trạng thái hiện tại của quá trình. Sai biệt tạm thời dùng làm tín hiệu tăng cường nội tại, xem hình 8.14. Có thể dùng sai biệt tạm thời để huấn luyện bộ phê phán. Xét bộ phê phán được biểu diễn thông qua mạng nơrôn hay hệ mờ:

$$\hat{V}(k+1)=h\big(x(k),u(k);\theta(k)\big) \tag{8.62}$$

Trong đó $\boldsymbol{\theta}(k)$ lá vectơ của tham số chỉnh định. Để cập nhật $\boldsymbol{\theta}(k)$, dùng luật học giảm theo gradien:

$$\theta(k+1) = \theta(k) + a_h \frac{\partial h}{\partial \theta}(k)\Delta(k), \tag{8.63}$$

Trong đó $a_h > 0$ là tốc độ của bộ phê phán.

Đơn vị điều khiển, Bộ hiệu chỉnh tác động ngẫu nhiên. Khi huấn luyện bộ phê phán để dự báo tính năng sắp tới của hệ thống (hàm giá trị), thì đơn vị điều khiển có thể được cập nhật nhằm thiết lập ánh xạ tối ưu giữa các trạng thái hệ thống và tác động điều khiển. Sai biệt tạm thời được dùng để cập nhật đơn vị điều khiển như sau.

Cho một trạng thái nào đó, tác động điều khiển u được tính dùng bộ điều khiển hiện tại. Tác động này không được áp dụng vào quá trình, nhưng lại bị thay đổi một cách ngẫu nhiên để *u'* bằng cách cộng thêm giá trị ngẫu nhiên từ N(0, σ) vào u. Sau khi hiệu chỉnh, tác động *u'* được gởi đến quá trình tính được giá trị sai biệt tạm thời. Nếu tính năng hiện tại tốt hơn dự báo, thì bộ điều khiển được cập nhật theo hướng tác động hiệu chỉnh *u'*.

Xét bộ điều khiển được biểu diễn dùng mạng nơrôn hay hệ mờ

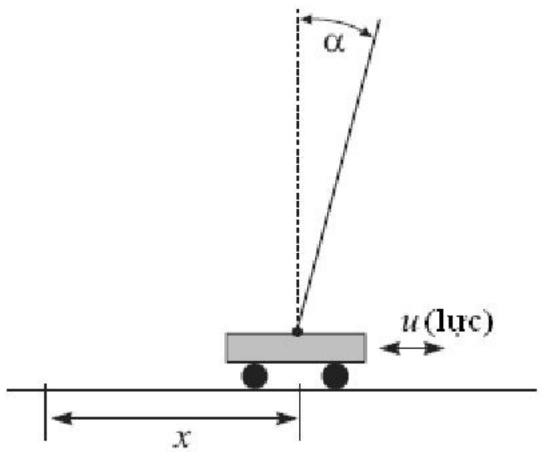
$$u(k) = g(\mathbf{x}(k); \boldsymbol{\varphi}(k)) \tag{8.64}$$

trong đó $\boldsymbol{\varphi}(k)$ là vectơ tham số hiệu chỉnh. Để cập nhật $\boldsymbol{\varphi}(k)$, dùng luật huấn luyện sau:

$$\varphi(k+1) = \varphi(k) + a_g \frac{\partial g}{\partial \varphi}(k)\left[u'(k) - u(k)\right]\Delta(k), \tag{8.65}$$

Trong đó $a_g > 0$ là tốc độ học của bộ điều khiển.

Thí dụ 8.5 (Con lắc ngược) Trong thí dụ này, học tăng cường được dùng để huấn luyện bộ điều khiển con lắc ngược, là một bài toán kiểm nghiệm nổi tiếng. Mục tiêu là huấn luyện để bộ điều khiển cân bằng con lắc thẳng đứng khi xe chạy tới lui như hình 8.15.



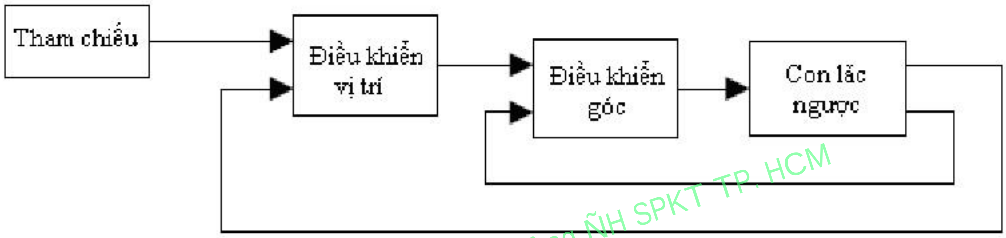
Hình 8.15. Con lắc ngược

Hệ thống có một ngõ vào u, gia tốc của xe (cart), và hai ngõ ra, vị trí xe x và góc lệch của con lắc α. Khi có được mô hình toán học hay mô phỏng của hệ thống, thì không khó khăn lắm để thiết kế bộ điều khiển. Hình 8.16 vẽ sơ đồ khối của các bộ điều khiển

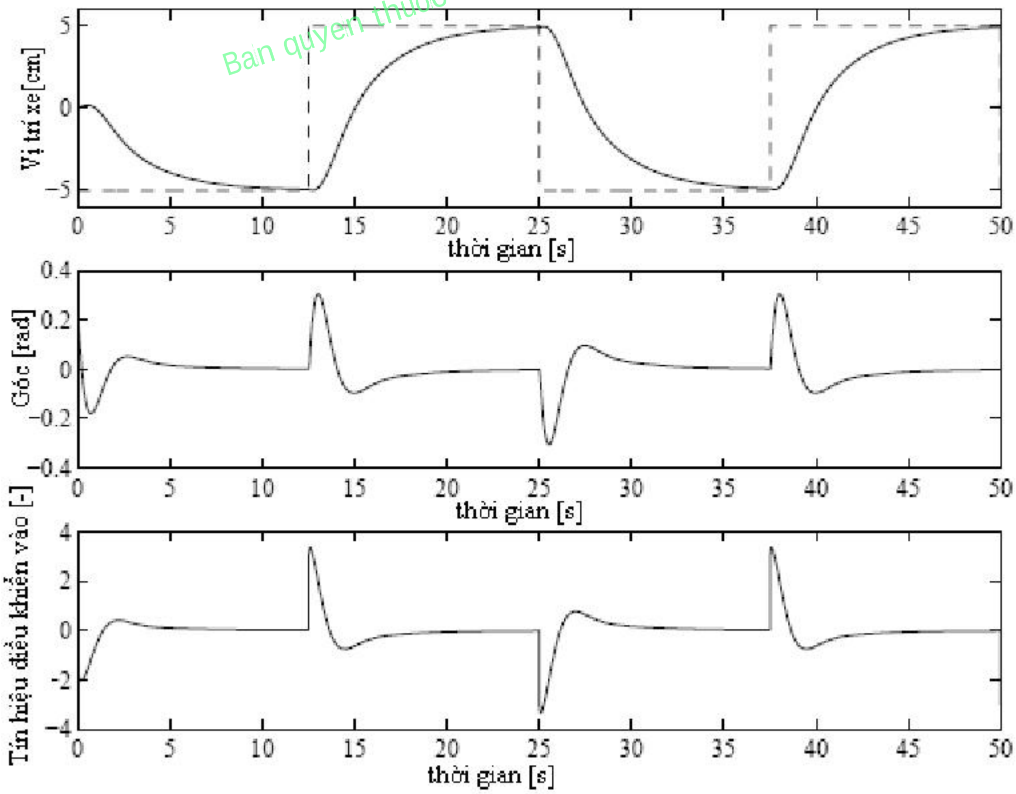
PD nối đuôi và được tinh chỉnh từ phép thử và sai dùng mô hình Simulink của hệ thống (invpend.mdl). Hình 8.17 vẽ đáp ứng của bộ điều khiển PD theo vị trí tham chiếu.

Khi dùng thực nghiệm phép học RL, thì bộ điều khiển bên trong được tạo thích nghi, trong khi bộ điều khiển vị trí PD vẫn được giữa nguyên. Mục đích là ổ định con lắc, hoàn toàn không phụ thuộc các chiến lược điều khiển ban đầu (tác động ngẫu nhiên).

Bộ phê phán được biểu diễn dùng mô hình mờ singleton dùng hai ngõ vào, giá trị góc hiện tại $\alpha(k)$ và tín hiệu điều khiển hiện tại $u(k)$. Dùng bảy hàm thành viên tam giác cho mỗi ngõ vào. Hàm thành viên là không đổi và tham số hệ quả là thích nghi. Các giá trị đầu là -1 cho từng tham số hệ quả.



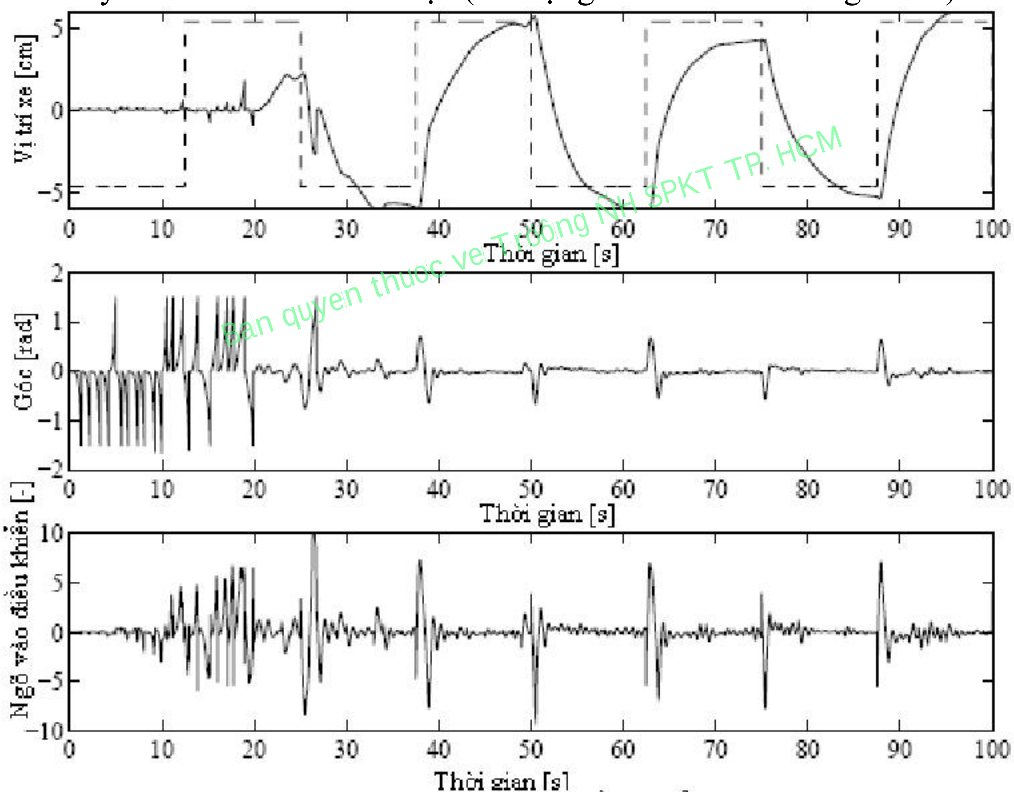
Hình 8.16 Bộ điều khiển PD nối tiếp cho con lắc



Hình 8.17. Hiệu năng của bộ điều khiển PD

Bộ điều khiển được biểu diễn dùng mô hình mờ singleton có hai ngõ vào, góc hiện tại $\alpha(k)$ và giá trị đạo hàm $\frac{\partial \alpha}{\partial t}(k)$. Năm hàm thành viên tam giác được dùng cho từng ngõ vào. Các hàm thành viên là không đổi và các tham số hệ quả là thích nghi. Giá trị đầu là 0 cho từng tham số hệ quả. Chiến lược điều khiển ban đầu được xác định hoàn toàn dùng bộ hiệu chỉnh tác động ngẫu nhiên (tự thân đã là ngẫu nhiên). Như thế chắc chắn là bộ điều khiển không ổn định. Sau khi thử với nhiều tác động điều khiển, (con lắc ngược được reset về hướng thẳng đứng sau mỗi thất bại), sơ đồ RL học phương thức điều khiển hệ thống (Hình 8.18).

Chú ý là trong khoãng gần 20 giây, bộ điều khiển không ổn định được hệ thống. Sau khoảng 20 đến 30 lần thất bại, hiệu năng được cải thiện nhanh và tiến dần đến hiệu năng của bộ điều khiển PD đã được chỉnh định tốt (hình 8.19). Để tạo kết quả này, tham số sau cùng của bộ điều khiển được cố định lại và nhiễu bị loại hoàn toàn. Hình 8.20 vẽ mặt phẳng phê phán và điều khiển sau cùng (final). Chú ý là phê phán ở trạng thái thưởng nhiều khi $\alpha = 0$ và $u = 0$. Trạng thái khi cả α và u đều là âm là phạt, do chúng tạo ra hỏng hóc (tác động điều khiển có chiều sai). Trạng thái khi α là âm nhưng u là dương (và ngược lại) thì được ước lượng giữa hai cực trị này. Các tác động điều khiển này có thể dẫn đến cải thiện (tác động điều khiển đi đúng chiều).



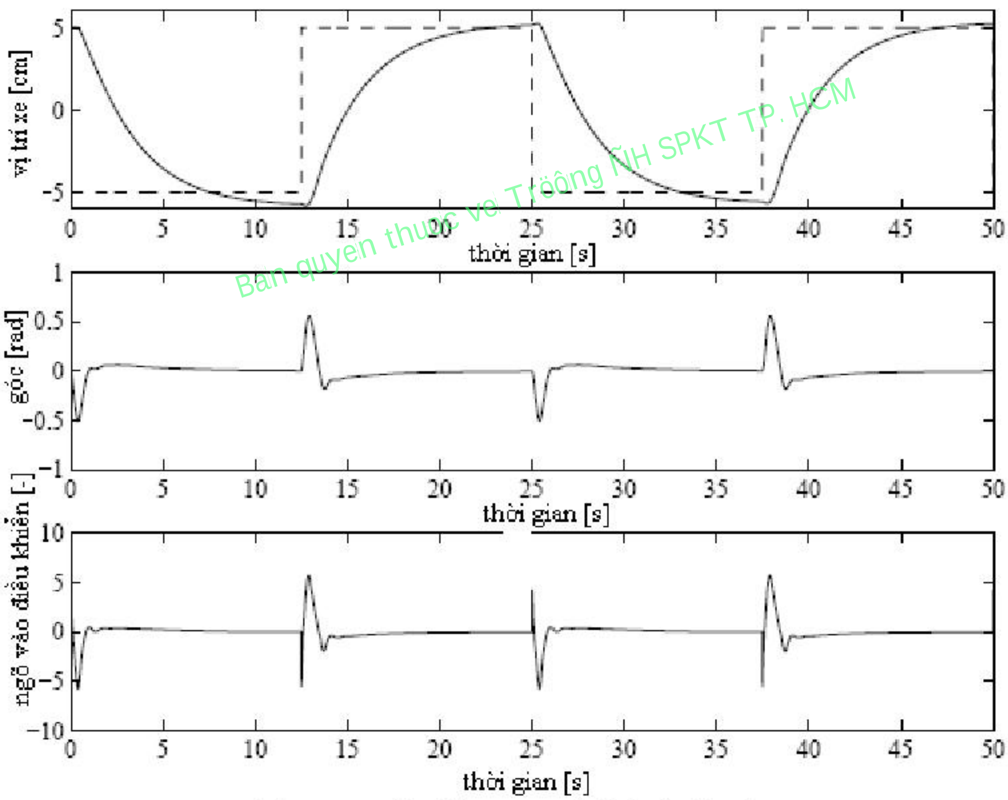
Hình 8.18. Quá trình học của bộ điều khiển RL

4. Tóm tắt và các điểm cần quan tâm

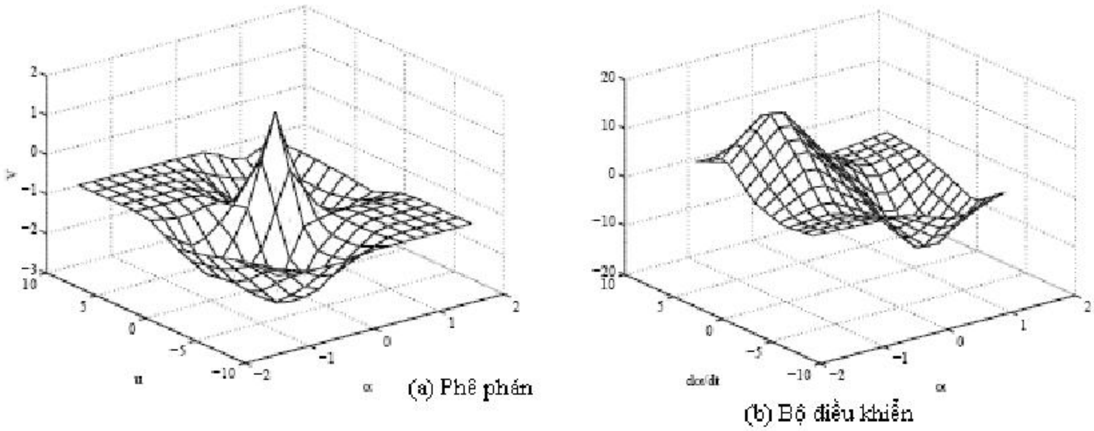
Chương đã giới thiệu nhiều phương pháp phát triển các bộ điều khiển phi tuyến dùng mô hình mờ hay mạng nơrôn quá trình điều khiển. Đó là các bộ điều khiển nghịch, điều khiển dự báo, và hai kỹ thuật điều khiển thích nghi. Mô hình nội tại có thể dùng trong phương pháp tổng quát để loại nhiễu cộng tại ngõ vào và các sai số bé khi mô hình hóa trong điều khiển nghịch hay mô hình dự báo.

5. Bài tập

1. Vẽ sơ đồ tổng quát của hệ điều khiển truyền thẳng trong đó bộ điều khiển dùng mô hình nghịch của đặc tính động của quá trình điều khiển. Mô tả các khối và tín hiệu trong sơ đồ.
2. Xét hệ mô hình hình Takagi–Sugeno dang affine bậc một:
3. Giải thích ý niệm của phương pháp điều khiển dự báo. Cho biết công thúc tìm hàm chi phí và giải thích các ký hiệu.
4. Nguyên tắc điều khiển thích nghi gián tiếp là gì? Vẽ sơ đồ khối của sơ đồ điều khiển gián tiếp và giải thích chức năng các khối.
5. Giải thích ý tưởng của phương pháp điều khiển dùng mô hình nội tại (IMC: internal model control).
6. Cho biết phương trình dùng cho hàm giá trị (*value function*) được dùng trong luật học tăng cường (reinforcement learning).



Hình 8.19. Hiệu năng của bộ RL sau khi huấn luyện



Hình 8.20. Mặt phẳng sau cùng của phê pháp (trái) và điều khiển(phải)

TÀI LIỆU THAM KHẢO

[1] Chin-Teng Lin, C.S George Lee, **NEURAL FUZZY SYSTEMS**, Prentice Hall 1996

[2] Michael Negnevitsky, **ARTIFICIAL INTELLIGENCE**, Addison-Wesley 2002

[3] Robert Babuska, **FUZZY AND NEURAL CONTROL,** DISC Course Lecture Notes (September 2004)

[4] Bùi Công Cường và Nguyễn Doãn Phước, Hệ mờ – Mạng nơron và Ứng dụng, NXB Khoa Học và Kỹ Thuật, 2001.

[5] Nguyễn Hoàng Hải, Công cụ phân tích Wavelets và ứng dụng trong Matlab, NXB Khoa học kỹ thuật, 2005.

[6] Phan Xuân Minh và Nguyễn Doãn Phước, Lý thuyết Điều khiển Mờ, NXB Khoa Học và Kỹ Thuật, 2004.

[7] Nguyễn Đình Thúc, Mạng Nơron Phương pháp và Ứng dụng, NXB Giáo Dục, 2000.

[8] Đỗ Trung Tuấn, Hệ Chuyên gia, NXB Giáo Dục, 1999.

[9] Nguyễn Thiện Thành, Mạng Nơron: Nhận Dạng Dự Báo và Điều Khiển, ĐH Bách Khoa TPHCM, 2001.