

Chương 1

CÔNG NGHỆ TÍNH TOÁN MỀM (Soft Computing technology)

Vài nét về lịch sử phát triển lý thuyết điều khiển .

- Phương pháp biến phân cổ điển Euler_Lagrange 1766 .
- Tiêu chuẩn ổn định Lyapunov 1892 .
- Trí tuệ nhân tạo 1950 .
- Hệ thống điều khiển máy bay siêu nhẹ 1955 .
- Nguyên lý cực tiểu Pontryagin 1956 .
- Phương pháp quy hoạch động Belman 1957 .
- Điều khiển tối ưu tuyến tính dạng toàn phương LQR (LQR : Linear Quadratic Regulator) .
- Điều khiển kép Feldbaum 1960 .
- Thuật toán di truyền 1960 .
- Nhận dạng hệ thống 1965 .
- Logic mờ 1965 .
- Luật điều khiển hệ thống thích nghi mô hình tham chiếu MRAS và bộ tự chỉnh định STR 1970 (MRAS : Model-Reference Adaptive System , STR : Self-Tuning Regulator) .
- Hệ tự học Tsytkin 1971 .
- Sản phẩm công nghiệp 1982 .
- Lý thuyết bền vững 1985 .
- Công nghệ tính toán mềm và điều khiển tích hợp 1985 .



1.1. Giới thiệu về công nghệ tính toán mềm

Trong thực tế cuộc sống, các bài toán liên quan đến hoạt động nhận thức, trí tuệ của con người đều hàm chứa những đại lượng, thông tin mà bản chất là không chính xác, không chắc chắn, không đầy đủ. Ví dụ: sẽ chẳng bao giờ có các thông tin, dữ liệu cũng như các mô hình toán đầy đủ và chính xác cho các bài toán dự báo thời tiết. Nhìn chung con người luôn ở trong bối cảnh là không có thông tin đầy đủ và chính xác cho các hoạt động ra quyết định của bản thân mình.

Trong lĩnh vực khoa học kỹ thuật cũng vậy, các hệ thống phức tạp trên thực tế thường không thể mô tả đầy đủ và chính xác bởi các phương trình toán học truyền thống. Kết quả là những cách tiếp cận kinh điển dựa trên kỹ thuật phân tích và các phương trình toán học nhanh chóng tỏ ra không còn phù hợp. Vì thế, công nghệ tính toán mềm chính là một giải pháp trong lĩnh vực này.

Một số đặc điểm của công nghệ tính toán mềm:

- Tính toán mềm căn cứ trên các đặc điểm, hành vi của con người và tự nhiên để đưa ra các quyết định hợp lý trong điều kiện không chính xác và không chắc chắn.
- Các thành phần của tính toán mềm có sự bổ sung, hỗ trợ lẫn nhau.
- Tính toán mềm là một hướng nghiên cứu mở, bất kỳ một kỹ thuật mới nào được tạo ra từ việc bắt chước trí thông minh của con người đều có thể trở thành một thành phần mới của tính toán mềm.

Công nghệ tính toán mềm bao gồm 3 thành phần chính:

- Điều khiển mờ
- Mạng nơ-ron nhân tạo
- Lập luận xác suất (thuật giải di truyền và lý thuyết hỗn mang..).

Ta sẽ đi vào phân tích từng thành phần của công nghệ tính toán mềm.

1.2. Điều khiển mờ

Trong công nghệ tính toán mềm, thành phần phát triển vượt bậc nhất và được ứng dụng rộng rãi nhất đó là **logic mờ**.

Khái niệm về logic mờ được giáo sư L.A Zadeh đưa ra lần đầu tiên năm 1965, tại trường Đại học Berkeley, bang California - Mỹ. Từ đó lý thuyết mờ đã được phát triển và ứng dụng rộng rãi.

Năm 1970 tại trường Mary Queen, London – Anh, Ebrahim Mamdani đã dùng logic mờ để điều khiển một máy hơi nước mà ông không thể điều khiển được bằng kỹ thuật cổ điển. Tại Đức Hann Zimmermann đã dùng logic mờ

cho các hệ ra quyết định. Tại Nhật logic mờ được ứng dụng vào nhà máy xử lý nước của Fuji Electronic vào 1983, hệ thống xe điện ngầm của Hitachi vào 1987.

Lý thuyết mờ ra đời ở Mỹ, ứng dụng đầu tiên ở Anh nhưng phát triển mạnh mẽ nhất là ở Nhật. Trong lĩnh vực Tự động hoá logic mờ ngày càng được ứng dụng rộng rãi. Nó thực sự hữu dụng với các đối tượng phức tạp mà ta chưa biết rõ hàm truyền, logic mờ có thể giải quyết các vấn đề mà điều khiển kinh điển không làm được.

1.2.1 Khái niệm cơ bản

Một cách tổng quát, một hệ thống mờ là một tập hợp các qui tắc dưới dạng If ... Then ... để tái tạo hành vi của con người được tích hợp vào cấu trúc điều khiển của hệ thống.

Việc thiết kế một hệ thống mờ mang rất nhiều tính chất chủ quan, nó tùy thuộc vào kinh nghiệm và kiến thức của người thiết kế. Ngày nay, tuy kỹ thuật mờ đã phát triển vượt bậc nhưng vẫn chưa có một cách thức chính quy và hiệu quả để thiết kế một hệ thống mờ. Việc thiết kế vẫn phải dựa trên một kỹ thuật rất cổ điển là thử - sai và đòi hỏi phải đầu tư nhiều thời gian để có thể đi tới một kết quả chấp nhận được.

Để hiểu rõ khái niệm “MỜ” là gì ta hãy thực hiện phép so sánh sau :

Trong toán học phổ thông ta đã học khá nhiều về tập hợp, ví dụ như tập các số thực R , tập các số nguyên tố $P=\{2,3,5,\dots\}$... Những tập hợp như vậy được gọi là tập hợp kinh điển hay tập rõ, tính “RÕ” ở đây được hiểu là với một tập xác định S chứa n phần tử thì ứng với phần tử x ta xác định được một giá trị $y=S(x)$.

Giờ ta xét phát biểu thông thường về tốc độ một chiếc xe mô tô : chậm, trung bình, hơi nhanh, rất nhanh. Phát biểu “CHẠM” ở đây không được chỉ rõ là bao nhiêu km/h , như vậy từ “CHẠM” có miền giá trị là một khoảng nào đó, ví dụ $5km/h - 20km/h$ chẳng hạn. Tập hợp $L=\{chậm, trung bình, hơi nhanh, rất nhanh\}$ như vậy được gọi là một tập các biến ngôn ngữ. Với mỗi thành phần ngôn ngữ x_k của phát biểu trên nếu nó nhận được một khả năng $\mu(x_k)$ thì tập hợp F gồm các cặp $(x, \mu(x_k))$ được gọi là tập mờ.

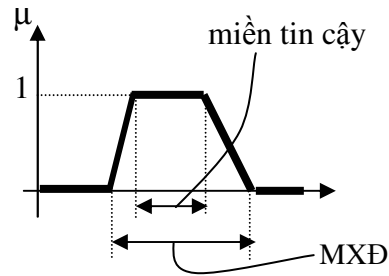
1. Định nghĩa tập mờ

Tập mờ F xác định trên tập kinh điển B là một tập mà mỗi phần tử của nó là một cặp giá trị $(x, \mu_F(x))$, với $x \in X$ và $\mu_F(x)$ là một ánh xạ :

$$\mu_F(x) : B \rightarrow [0 \ 1]$$

trong đó : μ_F gọi là hàm thuộc , B gọi là tập nền.

2. Các thuật ngữ trong logic mờ

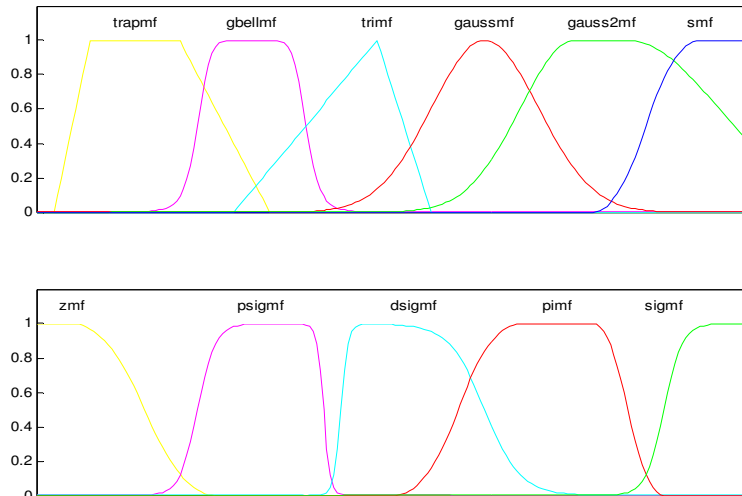


Hình 1.1:

- Độ cao tập mờ F là giá trị $h = \text{Sup} \mu_F(x)$, trong đó $\text{sup} \mu_F(x)$ chỉ giá trị nhỏ nhất trong tất cả các chặn trên của hàm $\mu_F(x)$.
- Miền xác định của tập mờ F , ký hiệu là S là tập con thoả mãn :

$$S = \text{Supp} \mu_F(x) = \{ x \in B \mid \mu_F(x) > 0 \}$$
- Miền tin cậy của tập mờ F , ký hiệu là T là tập con thoả mãn :

$$T = \{ x \in B \mid \mu_F(x) = 1 \}$$
- Các dạng hàm thuộc (membership function) trong logic mờ
 Có rất nhiều dạng hàm thuộc như : Gaussian, PI-shape, S-shape, Sigmoidal, Z-shape ...



3. Biến ngôn ngữ

Biến ngôn ngữ là phần tử chủ đạo trong các hệ thống dùng logic mờ. Ở đây các thành phần ngôn ngữ của cùng một ngữ cảnh được kết hợp lại với nhau.

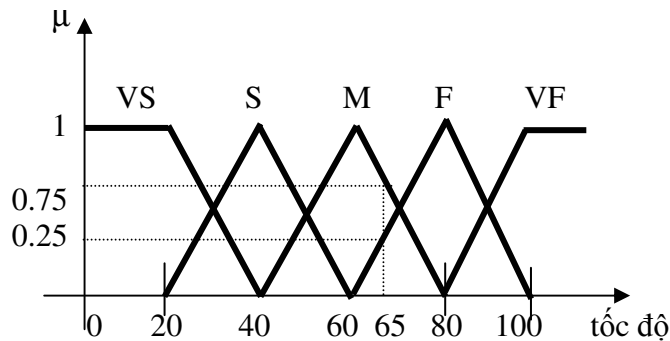
Để minh họa về hàm thuộc và biến ngôn ngữ ta xét ví dụ sau :

Xét tốc độ của một chiếc xe mô tô, ta có thể phát biểu xe đang chạy:

- Rất chậm (VS)
- Chậm (S)
- Trung bình (M)
- Nhanh (F)
- Rất nhanh (VF)

Những phát biểu như vậy gọi là biến ngôn ngữ của tập mờ. Gọi x là giá trị của biến tốc độ, ví dụ $x = 10\text{km/h}$, $x = 60\text{km/h}$... Hàm thuộc tương ứng của các biến ngôn ngữ trên được ký hiệu là :

$$\mu_{VS}(x), \mu_S(x), \mu_M(x), \mu_F(x), \mu_{VF}(x)$$



Hình 1.2:

Như vậy biến tốc độ có hai miền giá trị :

- Miền các giá trị ngôn ngữ :
 $N = \{ \text{rất chậm, chậm, trung bình, nhanh, rất nhanh} \}$
- Miền các giá trị vật lý :
 $V = \{ x \in B \mid x \geq 0 \}$

Biến tốc độ được xác định trên miền ngôn ngữ N được gọi là biến ngôn ngữ.

Với mỗi $x \in B$ ta có hàm thuộc :

$$x \rightarrow \mu_x = \{ \mu_{VS}(x), \mu_S(x), \mu_M(x), \mu_F(x), \mu_{VF}(x) \}$$

Ví dụ hàm thuộc tại giá trị rõ $x = 65\text{km/h}$ là :

$$\mu_x(65) = \{ 0; 0; 0.75; 0.25; 0 \}$$

4. Các phép toán trên tập mờ

Cho X, Y là hai tập mờ trên không gian nền B , có các hàm thuộc tương ứng là μ_X, μ_Y , khi đó :

- Phép hợp hai tập mờ : $X \cup Y$
 - + Theo luật Max $\mu_{X \cup Y}(b) = \text{Max}\{ \mu_X(b), \mu_Y(b) \}$
 - + Theo luật Sum $\mu_{X \cup Y}(b) = \text{Min}\{ 1, \mu_X(b) + \mu_Y(b) \}$
 - + Tổng trực tiếp $\mu_{X \cup Y}(b) = \mu_X(b) + \mu_Y(b) - \mu_X(b) \cdot \mu_Y(b)$
- Phép giao hai tập mờ : $X \cap Y$
 - + Theo luật Min $\mu_{X \cap Y}(b) = \text{Min}\{ \mu_X(b), \mu_Y(b) \}$
 - + Theo luật Lukasiewicz $\mu_{X \cap Y}(b) = \text{Max}\{0, \mu_X(b) + \mu_Y(b) - 1\}$
 - + Theo luật Prod $\mu_{X \cap Y}(b) = \mu_X(b) \cdot \mu_Y(b)$
- Phép bù tập mờ : $\mu_{X^c}(b) = 1 - \mu_X(b)$

5. Luật hợp thành

A. Mệnh đề hợp thành

Ví dụ điều khiển mực nước trong bồn chứa, ta quan tâm đến 2 yếu tố :

- + Mực nước trong bồn $L = \{ \text{rất thấp, thấp, vừa} \}$
- + Góc mở van ống dẫn $G = \{ \text{đóng, nhỏ, lớn} \}$

Ta có thể suy diễn cách thức điều khiển như thế này :

- Nếu *mực nước = rất thấp* Thì *góc mở van = lớn*
- Nếu *mực nước = thấp* Thì *góc mở van = nhỏ*
- Nếu *mực nước = vừa* Thì *góc mở van = đóng*

Trong ví dụ trên ta thấy có cấu trúc chung là “*Nếu A thì B*”. Cấu trúc này gọi là mệnh đề hợp thành, A là mệnh đề điều kiện, $C = A \Rightarrow B$ là mệnh đề kết luận.

Định lý Mamdani :

“*Độ phụ thuộc của kết luận không được lớn hơn độ phụ thuộc điều kiện*”

Nếu hệ thống có nhiều đầu vào và nhiều đầu ra thì mệnh đề suy diễn có dạng tổng quát như sau :

$$\text{If } N = n_i \text{ and } M = m_i \text{ and } \dots \text{ Then } R = r_i \text{ and } K = k_i \text{ and } \dots$$

B. Luật hợp thành mờ

Luật hợp thành là tên gọi chung của mô hình biểu diễn một hay nhiều hàm thuộc cho một hay nhiều mệnh đề hợp thành.

Các luật hợp thành cơ bản

- + Luật Max – Min
- + Luật Max – Prod
- + Luật Sum – Min
- + Luật Sum – Prod

a. Thuật toán xây dựng mệnh đề hợp thành cho hệ SISO

Luật mờ cho hệ SISO có dạng “**If A Then B**”

Chia hàm thuộc $\mu_A(x)$ thành n điểm x_i , $i = 1, 2, \dots, n$

Chia hàm thuộc $\mu_B(y)$ thành m điểm y_j , $j = 1, 2, \dots, m$

Xây dựng ma trận quan hệ mờ R

$$R = \begin{bmatrix} \mu_R(x_1, y_1) & \dots & \dots & \mu_R(x_1, y_m) \\ \mu_R(x_2, y_1) & \dots & \dots & \mu_R(x_2, y_m) \\ \dots & \dots & \dots & \dots \\ \mu_R(x_n, y_1) & \dots & \dots & \mu_R(x_n, y_m) \end{bmatrix} = \begin{bmatrix} r_{11} & \dots & \dots & r_{1m} \\ r_{21} & \dots & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{n1} & \dots & \dots & r_{nm} \end{bmatrix}$$

Hàm thuộc $\mu_{B'}(y)$ đầu ra ứng với giá trị rõ đầu vào x_k có giá trị

$\mu_{B'}(y) = a^T \cdot R$, với $a^T = \{ 0, 0, 0, \dots, 0, 1, 0, \dots, 0, 0 \}$. Số 1 ứng với vị trí thứ k .

Trong trường hợp đầu vào là giá trị mờ A' thì $\mu_{B'}(y)$ là :

$\mu_{B'}(y) = \{ l_1, l_2, l_3, \dots, l_m \}$ với $l_k = \max \min \{ a_i, r_{ik} \}$.

b. Thuật toán xây dựng mệnh đề hợp thành cho hệ MISO

Luật mờ cho hệ MISO có dạng :

“If $cd_1 = A_1$ and $cd_2 = A_2$ and ... Then $rs = B$ ”

Các bước xây dựng luật hợp thành R :

- Rời rạc các hàm thuộc $\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n), \mu_B(y)$
- Xác định độ thỏa mãn H cho từng vectơ giá trị rõ đầu vào $x = \{ c_1, c_2, \dots, c_n \}$ trong đó c_i là một trong các điểm mẫu của $\mu_{A_i}(x_i)$. Từ đó suy ra

$$H = \text{Min} \{ \mu_{A_1}(c_1), \mu_{A_2}(c_2), \dots, \mu_{A_n}(c_n) \}$$

- Lập ma trận R gồm các hàm thuộc giá trị mờ đầu ra cho từng vectơ giá trị mờ đầu vào: $\mu_{B'}(y) = \text{Min} \{ H, \mu_B(y) \}$ hoặc $\mu_{B'}(y) = H \cdot \mu_B(y)$

6. Giải mờ

Giải mờ là quá trình xác định giá trị rõ ở đầu ra từ hàm thuộc $\mu_{B'}(y)$ của tập mờ B' . Có 2 phương pháp giải mờ :

1. Phương pháp cực đại

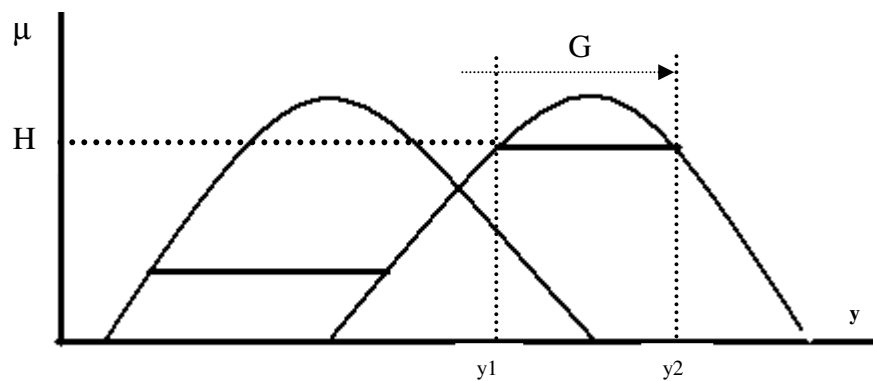
Các bước thực hiện :

- Xác định miền chứa giá trị y' , y' là giá trị mà tại đó $\mu_{B'}(y)$ đạt Max

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

- Xác định y' theo một trong 3 cách sau :

- + Nguyên lý trung bình
- + Nguyên lý cận trái
- + Nguyên lý cận phải



Hình 1.3:

- Nguyên lý trung bình : $y' = \frac{y1 + y2}{2}$
- Nguyên lý cận trái : chọn $y' = y1$
- Nguyên lý cận phải : chọn $y' = y2$

2. Phương pháp trọng tâm

Điểm y' được xác định là hoành độ của điểm trọng tâm miền được bao bởi trục hoành và đường $\mu_{B'}(y)$.

Công thức xác định :

$$y' = \frac{\int_S y \mu(y) dy}{\int_S \mu(y) dy} \quad \text{trong đó } S \text{ là miền xác định của tập mờ } B'$$

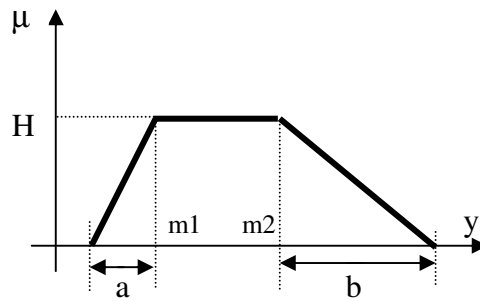
♦ Phương pháp trọng tâm cho luật Sum-Min

Giả sử có m luật điều khiển được triển khai, ký hiệu các giá trị mờ đầu ra của luật điều khiển thứ k là $\mu_{B^k}(y)$ thì với quy tắc Sum-Min hàm thuộc sẽ là

$$\mu_B(y) = \sum_{k=1}^m \mu_{B^k}(y) , \text{ và } y' \text{ được xác định :}$$

$$y' = \frac{\int_S \left(y \sum_{k=1}^m \mu_{B^k}(y) \right) dy}{\int_S \sum_{k=1}^m \mu_{B^k}(y) dy} = \frac{\sum_{k=1}^m (y \mu_{B^k}(y) dy)}{\sum_{k=1}^m \left(\int_S \mu_{B^k}(y) dy \right)} = \frac{\sum_{k=1}^m M_k}{\sum_{k=1}^m A_k} \quad (1.1)$$

trong đó $M_i = \int_S y \mu_{B^k}(y) dy$ và $A_i = \int_S \mu_{B^k}(y) dy \quad i=1,2,\dots,m$



Xét riêng cho trường hợp các hàm thuộc dạng hình thang như hình trên :

$$M_k = \frac{H}{6} (3m_2^2 - 3m_1^2 + b^2 - a^2 + 3m_2b + 3m_1a)$$

$$A_k = \frac{H}{2} (2m_2 - 2m_1 + a + b)$$

Chú ý hai công thức trên có thể áp dụng cả cho luật Max-Min

♦ Phương pháp độ cao

Từ công thức (1.1), nếu các hàm thuộc có dạng Singleton thì ta được:

$$y' = \frac{\sum_{k=1}^m y_k H_k}{\sum_{k=1}^m H_k} \quad \text{với } H_k = \mu_{B^k}(y_k)$$

Đây là công thức giải mờ theo phương pháp độ cao.

7. Mô hình mờ Tagaki-Sugeno

Mô hình mờ mà ta nói đến trong các phần trước là mô hình Mamdani. Ưu điểm của mô hình Mamdani là đơn giản, dễ thực hiện nhưng khả năng mô tả hệ thống không tốt. Trong kỹ thuật điều khiển người ta thường sử dụng mô hình mờ Tagaki-Sugeno (TS).

Tagaki-Sugeno đưa ra mô hình mờ sử dụng cả không gian trạng thái mờ lẫn mô tả linh hoạt hệ thống. Theo Tagaki/Sugeno thì một vùng mờ LX^k được mô tả bởi luật :

$$R_{sk} : \text{If } x = LX^k \text{ Then } \dot{x} = A(x^k)x + B(x^k)u \quad (1.2)$$

Luật này có nghĩa là: nếu vectơ trạng thái x nằm trong vùng LX^k thì hệ thống được mô tả bởi phương trình vi phân cục bộ $\dot{x} = A(x^k)x + B(x^k)u$. Nếu toàn bộ các luật của hệ thống được xây dựng thì có thể mô tả toàn bộ trạng thái của hệ trong toàn cục. Trong (1.2) ma trận $A(x^k)$ và $B(x^k)$ là những ma trận hằng của hệ thống ở trọng tâm của miền LX^k được xác định từ các chương trình nhận dạng. Từ đó rút ra được :

$$\dot{x} = \sum w_k (A(x^k)x + B(x^k)u) \quad (1.3)$$

với $w_k(x) \in [0, 1]$ là độ thỏa mãn đã chuẩn hoá của x^* đối với vùng mờ LX^k

Luật điều khiển tương ứng với (1.2) sẽ là :

$$R_{ck} : \text{If } x = LX^k \text{ Then } u = K(x^k)x$$

Và luật điều khiển cho toàn bộ không gian trạng thái có dạng:

$$u = \sum_{k=1}^N w_k K(x^k)x \quad (1.4)$$

Từ (1.2) và (1.3) ta có phương trình động học cho hệ kín:

$$\dot{x} = \sum w_k (x)w_l (x)(A(x^k) + B(x^k)K(x^l))x$$

Ví dụ 1.1:

Một hệ TS gồm hai luật điều khiển với hai đầu vào x_1, x_2 và đầu ra y .

$$R_1 : \text{If } x_1 = \text{BIG} \quad \text{and } x_2 = \text{MEDIUM} \text{ Then } y_1 = x_1 - 3x_2$$

$$R_2 : \text{If } x_1 = \text{SMALL} \text{ and } x_2 = \text{BIG} \quad \text{Then } y_2 = 4 + 2x_1$$

Đầu vào rõ đo được là $x_1^* = 4$ và $x_2^* = 60$. Từ hình bên dưới ta xác định được :

$$LX_{BIG}(x_1^*) = 0.3 \quad \text{và} \quad LX_{BIG}(x_2^*) = 0.35$$

$$LX_{SMALL}(x_1^*) = 0.7 \quad \text{và} \quad LX_{MEDIUM}(x_2^*) = 0.75$$

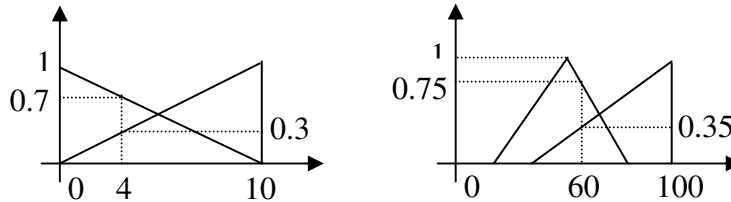
Từ đó xác định được :

$$\text{Min}(0.3 ; 0.75)=0.3 \text{ và } \text{Min}(0.35 ; 0.7)=0.35$$

$$y_1 = 4-3 \times 60 = -176 \text{ và } y_2 = 4+2 \times 4 = 12$$

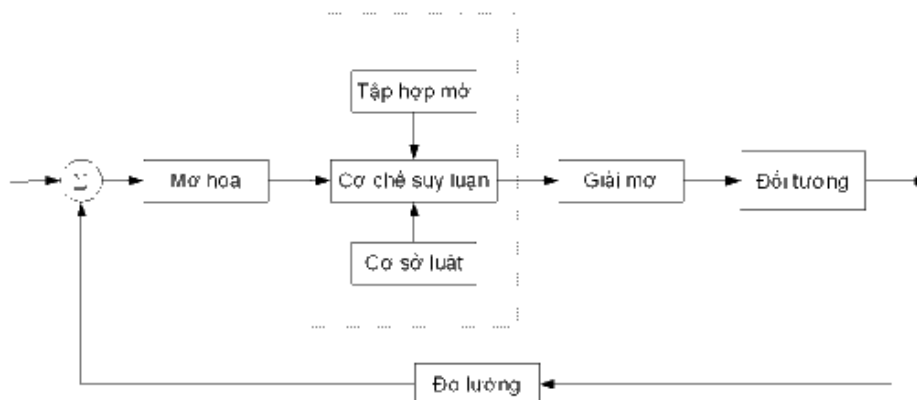
Như vậy hai thành phần R_1 và R_2 là $(0.3 ; -176)$ và $(0.35 ; 12)$. Theo phương pháp tổng trọng số trung bình ta có:

$$y = \frac{0.3 \times (-176) + 0.35 \times 12}{0.3 + 0.35} = -74.77$$



1.2.2. Bộ điều khiển mờ

1. Cấu trúc một bộ điều khiển mờ

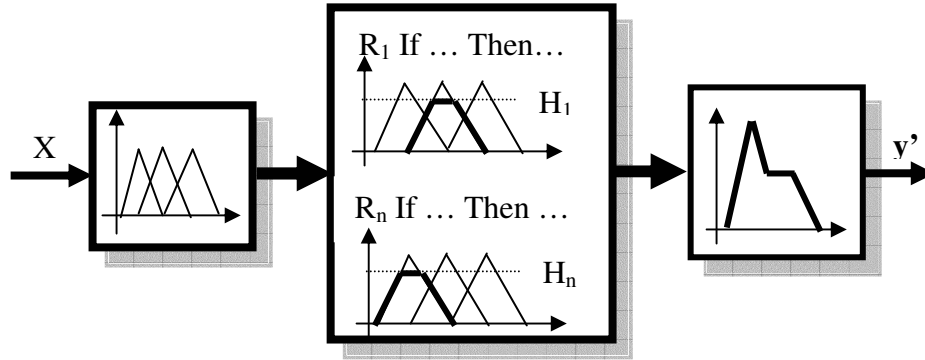


Sơ đồ khối bộ điều khiển mờ

Một bộ điều khiển mờ gồm 3 khâu cơ bản:

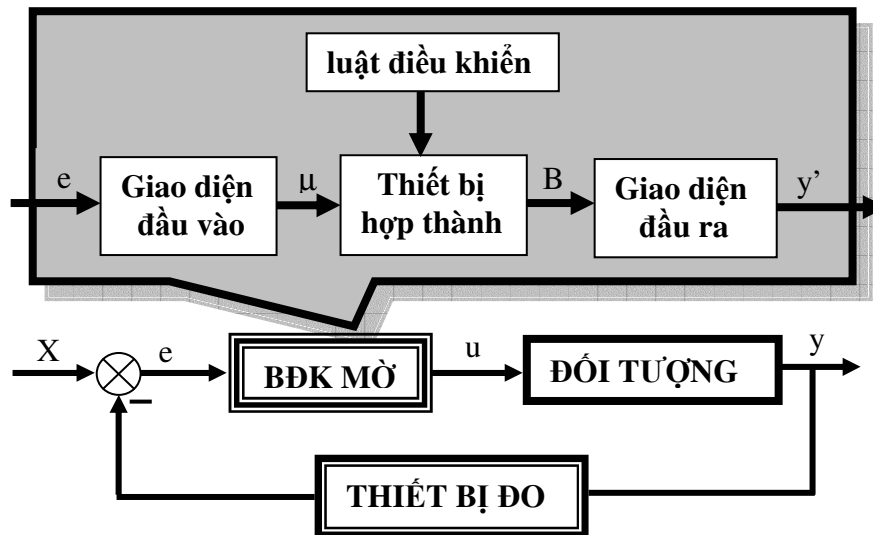
- + Khâu mờ hoá
- + Thực hiện luật hợp thành
- + Khâu giải mờ

Xét bộ điều khiển mờ MISO sau, với véctơ đầu vào $X = [u_1 \ u_2 \ \dots \ u_n]^T$



Hình 1.4:

2. Nguyên lý điều khiển mờ



Hình 1.5:

- ◆ Các bước thiết kế hệ thống điều khiển mờ.
 - + Giao diện đầu vào gồm các khâu: mờ hóa và các khâu hiệu chỉnh như tỷ lệ, tích phân, vi phân ...
 - + Thiết bị hợp thành : sự triển khai luật hợp thành R
 - + Giao diện đầu ra gồm : khâu giải mờ và các khâu giao diện trực tiếp với đối tượng.

3. Thiết kế bộ điều khiển mờ

- Các bước thiết kế:

B1 : Định nghĩa tất cả các biến ngôn ngữ vào/ra.

B2 : Xác định các tập mờ cho từng biến vào/ra (mờ hoá).

+ Miền giá trị vật lý của các biến ngôn ngữ.

+ Số lượng tập mờ.

+ Xác định hàm thuộc.

+ Rời rạc hoá tập mờ.

B3 : Xây dựng luật hợp thành.

B4 : Chọn thiết bị hợp thành.

B5 : Giải mờ và tối ưu hoá.

- Những lưu ý khi thiết kế BDK mờ

- Không bao giờ dùng điều khiển mờ để giải quyết bài toán mà có thể dễ dàng thực hiện bằng bộ điều khiển kinh điển.

- Không nên dùng BDK mờ cho các hệ thống cần độ an toàn cao.

- Thiết kế BDK mờ phải được thực hiện qua thực nghiệm.

- Phân loại các BDK mờ

i. Điều khiển Mamdani (MCFC)

ii. Điều khiển mờ trượt (SMFC)

iii. Điều khiển tra bảng (CMFC)

iv. Điều khiển Tagaki/Sugeno (TSFC)

4. Ví dụ minh họa:

Ví dụ 1.2:

Để điều khiển tự động máy điều hoà nhiệt độ bằng kỹ thuật logic mờ, người ta dùng hai cảm biến: Trong phòng là cảm biến nhiệt T_i , bên ngoài là cảm biến nhiệt T_o . Việc điều hoà nhiệt độ thông qua điều khiển tốc độ quạt làm lạnh máy điều hoà. Biết rằng:

- Tầm nhiệt độ quan tâm là $[0^\circ\text{C} - 50^\circ\text{C}]$

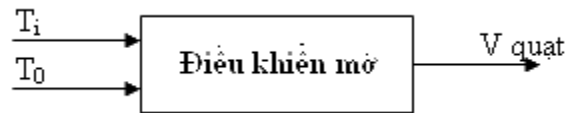
- Tốc độ quạt là $v \in [0 - 600 \text{ vòng/ phút}]$

Hãy tính tốc độ quạt trong trường hợp sau:

$$T_i = 27^\circ\text{C} \quad T_o = 32^\circ\text{C}$$

Giải bài toán theo đúng trình tự:

Bước 1: Xác định các biến ngôn ngữ vào – ra



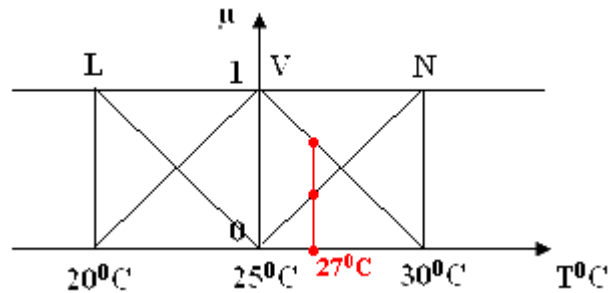
Bước 2: Xác định tập mờ cho từng biến vào/ra

T_i, T_o : {Lạnh ,Vừa ,Nóng) tương ứng với {20, 25, 30°C}

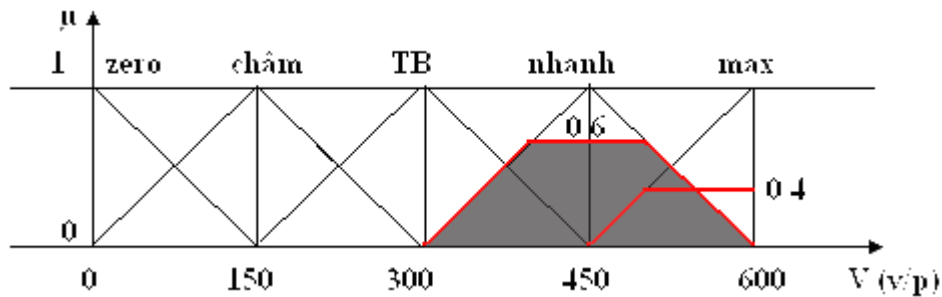
V : {Zero, Chậm, Trung bình ,Nhanh, Max) tương ứng với {0, 150, 300, 450, 600 vòng/phút}

Hàm thuộc: ta chọn hàm thuộc là hàm tam giác

Ngõ vào:



Ngõ ra:



Xét trường hợp: $T_i = 27^{\circ}\text{C}$ $T_o = 32^{\circ}\text{C}$

Ta có:

$$\mu(27^{\circ}\text{C}) = \{0 ; 0.6 ; 0.4\}$$

$$\mu(32^{\circ}\text{C}) = \{0 ; 0 ; 1 \}$$

Bước 3: Xây dựng luật hợp thành mờ

Ti \ To	Lạnh	Vừa	Nóng
Lạnh	Zero	Chậm	Trung bình
Vừa	Chậm	Trung bình	Nhanh
Nóng	Trung bình	Nhanh	Max

Bước 4: Giải mờ và tối ưu hóa

- **Chọn thiết bị hợp thành Max – Min:**

$$\mu(27^{\circ}\text{C}) = \{0 ; 0.6 ; 0.4\}$$

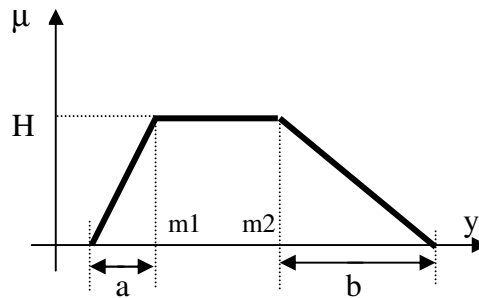
$$\mu(32^{\circ}\text{C}) = \{0 ; 0 ; 1\}$$

Luật max – min cho ta: Nhanh : 0.6

Max : 0.4

- **Giải mờ:**

- ♦ Phương pháp trọng tâm:



$$M_k = \frac{H}{6} (3m_2^2 - 3m_1^2 + b^2 - a^2 + 3m_2b + 3m_1a)$$

$$A_k = \frac{H}{2} (2m_2 - 2m_1 + a + b)$$

$$y' = \frac{\sum_{k=1}^m M_k}{\sum_{k=1}^m A_k} = 530 \text{ vòng/phút}$$

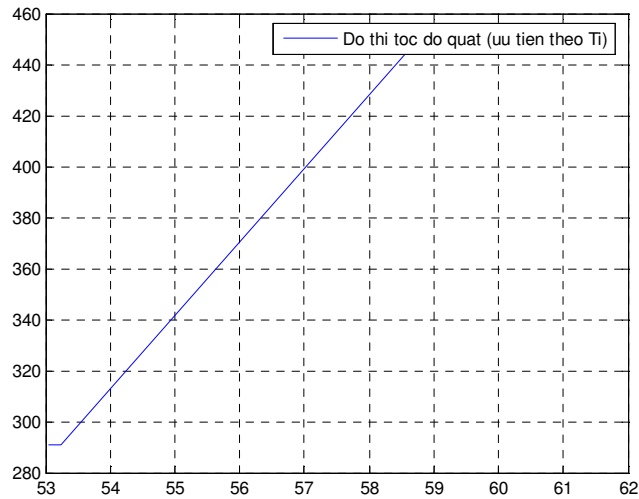
◆ Phương pháp độ cao:

$$y' = \frac{\sum_{k=1}^m y_k H_k}{\sum_{k=1}^m H_k} \quad \text{với } H_k = \mu_{B'k}(y_k)$$

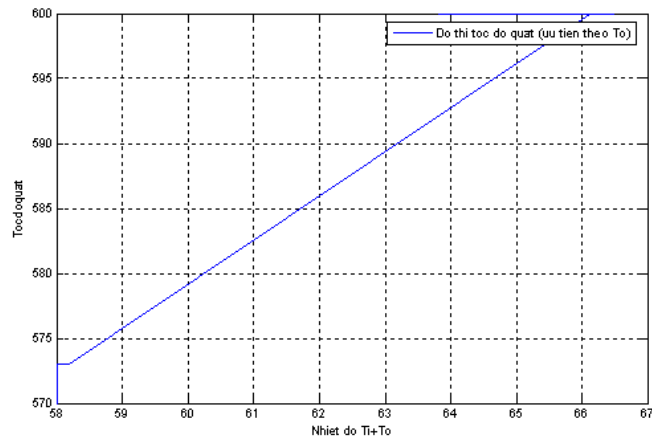
$$y' = \frac{0.6 \times 450 + 0.4 \times 600}{0.6 + 0.4} = 510 \text{ vòng/phút}$$

Bước 5: Vẽ đồ thị và nhận xét

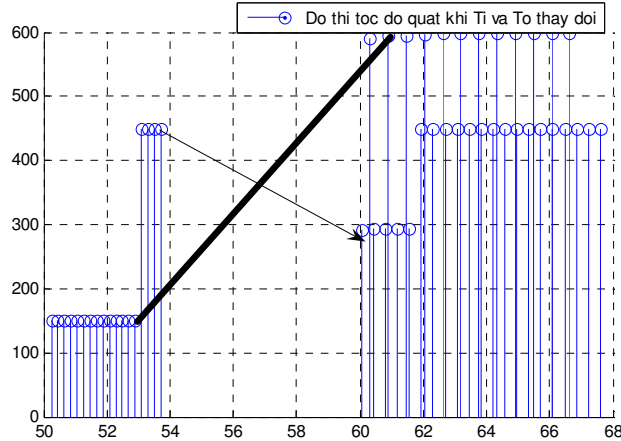
Đồ thị tốc độ quạt theo T_i



Đồ thị tốc độ quạt theo T_o



Đồ thị tốc độ quạt khi Ti và To thay đổi



Nhận xét:

- Đồ thị tốc độ quạt tăng tuyến tính khi vẽ theo Ti hoặc To
- Nếu cả 2 Ti và To thay đổi bất kì thì đồ thị $(Ti + To, V)$ cũng tăng tuyến tính và các điểm khác nằm đối xứng 2 bên của đường thẳng đó.
- Kết quả điều khiển chấp nhận được.

Nếu vẫn chưa đáp ứng được chất lượng đề ra (sai số, độ vọt lố ..), ta có thể tăng số phân cấp của các biến ngôn ngữ. Tuy nhiên, nếu tăng quá mức sẽ dẫn đến tình trạng quá khớp. Ví dụ:

Chọn các biến ngôn ngữ: $Ti, To: \{Rất lạnh, Lạnh, Vừa, Nóng, Rất nóng\}$
 tương ứng với các nhiệt độ $\{15, 20, 25, 30, 35\}$

Tốc độ quạt vẫn là: $\{Zero, Chậm, Trung bình, Nhanh, Max\}$

To \ Ti	Rất lạnh	Lạnh	Vừa	Nóng	Rất nóng
Rất lạnh	Zero	Zero	Zero	Chậm	TB
Lạnh	Zero	Chậm	Chậm	TB	Nhanh
Vừa	Zero	Chậm	TB	Nhanh	Max
Nóng	Chậm	TB	Nhanh	Nhanh	Max
Rất nóng	TB	Nhanh	Max	Max	Max

Khi đó, ta cũng sẽ thu được kết quả tương tự như trên.

Ví dụ 1.3:

Thiết kế bộ mờ điều khiển nhiệt độ. Bộ mờ có 2 ngõ vào là sai lệch $e(t)$ [ET] và đạo hàm sai lệch $de(t)$ [DET], một ngõ ra là đạo hàm công suất [DP].

Biết rằng:

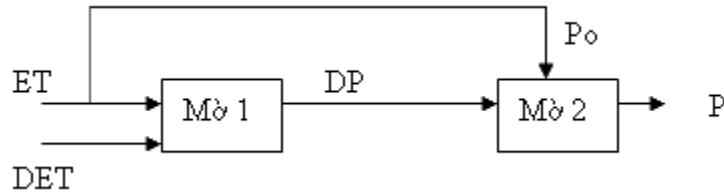
- _ Lò nhiệt có công suất là 5KW, tầm đo max là 200 °C, sai số là ± 5%.
- _ Tầm thay đổi của DET là -10 °C / s → 10 °C / s
- _ Tầm thay đổi của DP là -100W/s → 100W/s

Hãy tính công suất cung cấp cho lò trong trường hợp sau:

$$ET = 8^{\circ}\text{C} \quad DET = 9^{\circ}\text{C/s}$$

Lời giải:

Bước 1: Xác định các biến ngôn ngữ vào – ra



$$ET = T_{\text{đặt}} - T_{\text{đo}}$$

$$DET(i+1) = (ET(i+1) - ET(i))/T$$

$$DP(i+1) = (P(i+1) - P(i))/T$$

$$\Rightarrow P(i+1) = P(i) + DP(i+1) * T \leq P_{\text{max}} = 5\text{KW}$$

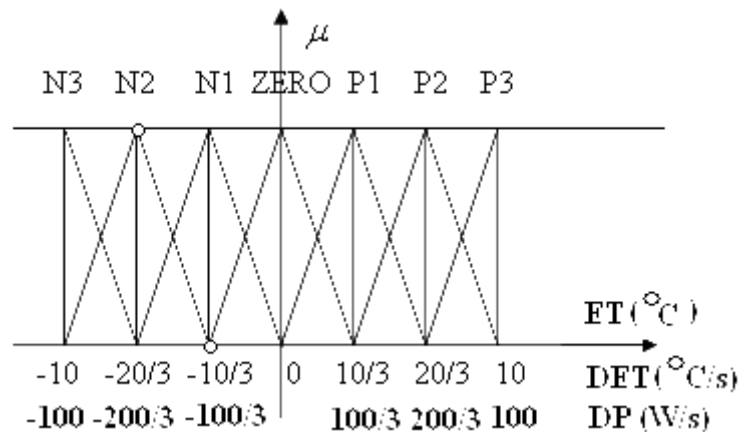
Bước 2: Xác định tập mờ cho từng biến vào/ra

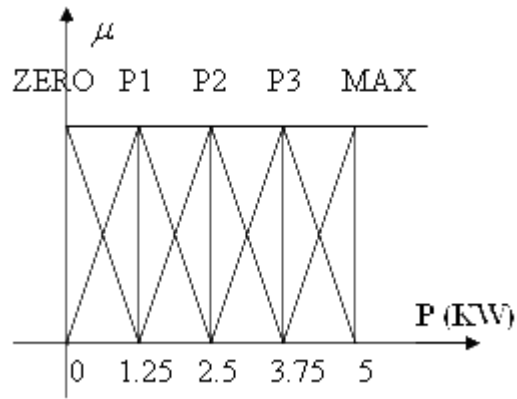
$$ET = \{ N3, N2, N1, \text{ZERO}, P1, P2, P3 \}$$

$$DET = \{ N3, N2, N1, \text{ZERO}, P1, P2, P3 \}$$

$$DP = \{ N3, N2, N1, \text{ZERO}, P1, P2, P3 \}$$

$$P = \{ \text{ZERO}, P1, P2, P3, \text{MAX} \}$$





Bước 3: Xây dựng luật hợp thành mờ

DP		DET						
		N3	N2	N1	Z	P1	P2	P3
ET	N3	N3	N3	N3	N3	N2	N1	Z
	N2	N3	N2	N2	N2	N1	Z	P1
	N1	N3	N2	N1	N1	Z	P1	P2
	Z	N3	N2	N1	Z	P1	P2	P3
	P1	N2	N1	Z	P1	P1	P2	P3
	P2	N1	Z	P1	P2	P2	P2	P3
	P3	Z	P1	P2	P3	P3	P3	P3

Bước 4: Giải mờ và tối ưu hóa

- **Chọn thiết bị hợp thành Max – Min:**

Xét trường hợp: $ET = 8^{\circ}C$; $DET = 9^{\circ}C/s$

$$\mu(8^{\circ}C) = \{0; 0; 0; 0; 0; 0.6; 0.4\}$$

$$\mu(9^{\circ}C/s) = \{0; 0; 0; 0; 0; 0; 0.3; 0.7\}$$

Luật max – min cho ta:

$$\begin{array}{l}
 ET: \mu = 0.6 \\
 DET: \mu = 0.3
 \end{array}
 \left. \begin{array}{l} \\ \\ \end{array} \right\} \xrightarrow{\min} 0.3 P2$$

$$\begin{array}{l}
 ET: \mu = 0.4 \\
 DET: \mu = 0.7
 \end{array}
 \left. \begin{array}{l} \\ \\ \end{array} \right\} \xrightarrow{\min} 0.4 P3$$

$$\begin{array}{l}
 ET: \mu = 0.4 \\
 DET: \mu = 0.3
 \end{array}
 \left. \begin{array}{l} \\ \\ \end{array} \right\} \xrightarrow{\min} 0.3 P3$$

$$\begin{array}{l}
 ET: \mu = 0.6 \\
 DET: \mu = 0.7
 \end{array}
 \left. \begin{array}{l} \\ \\ \end{array} \right\} \xrightarrow{\min} 0.6 P3$$

$$\left. \begin{array}{l} 0.3 P2 \\ 0.4 P3 \\ 0.3 P3 \\ 0.6 P3 \end{array} \right\} \xrightarrow{\max} 0.6 P3$$

Vậy
$$DP = \frac{0.3P_2 + 0.6P_3}{0.6 + 0.3} = \frac{0.6 * 100 + 0.3 * 200 / 3}{0.6 + 0.3} = 88.89 \text{ W/s}$$

Theo lý thuyết:

$$ET = 10^{\circ}\text{C} \longrightarrow P = 5 \text{ KW}$$

$$ET = 8^{\circ}\text{C} \longrightarrow P = 4 \text{ KW}$$

Do DP có 4 phân cấp dương nhưng P lại có 5 phân cấp dương nên sẽ có 2 trường hợp. Theo luật điều khiển mờ ta có:

- Chọn DP = { ZERO, P1, P2, P3 } ≡ { ZERO, P1, P2, P3 } = P

$$P = \frac{0.6 * 5 / 2 + 0.4 * 15 / 4}{1} = 3 \text{ KW}$$

- Chọn DP = { ZERO, P1, P2, P3 } ≡ { P1, P2, P3, MAX } = P

$$P = \frac{0.6 * 15 / 4 + 0.4 * 5}{1} = 4.25 \text{ KW}$$

Ta chọn P = 4.25 KW vì nó gần với giá trị lý thuyết hơn.

$$P = 4.25 + DP.T.10^{-3} < 5 \text{ KW}$$

$$\Rightarrow T < 8.4\text{s}$$

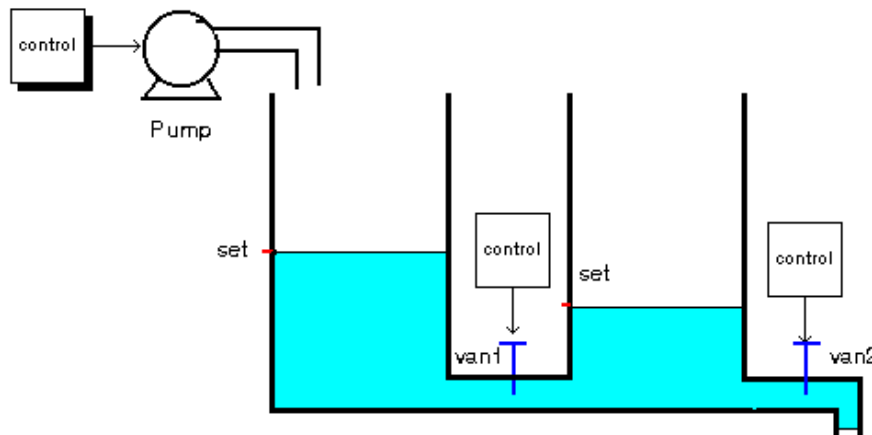
Ta chọn T = 8s $\Rightarrow P = 4.96 \text{ KW}$

Ta tính giá trị T trong trường hợp xấu nhất (ET_{\max}) và giữ nguyên giá trị này trong suốt quá trình tính toán để so sánh kết quả.

Ví dụ 1.4:

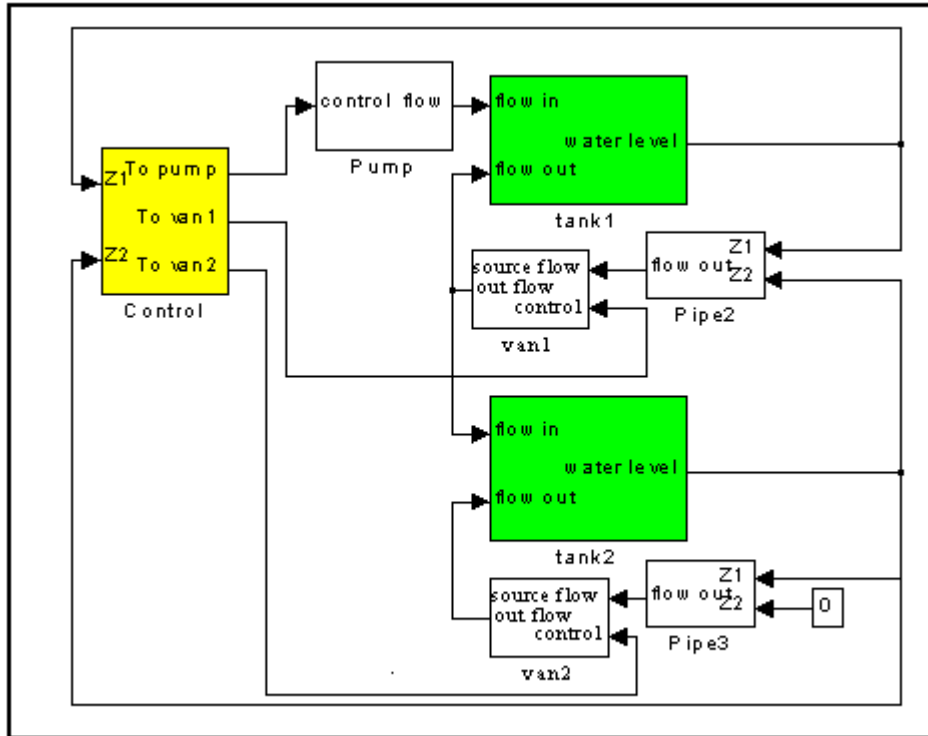
Dùng điều khiển mờ để điều khiển hệ thống bơm xả nước tự động. Hệ thống sẽ duy trì độ cao bồn nước ở một giá trị đặt trước như mô hình bên dưới.

♦ **Mô hình :**

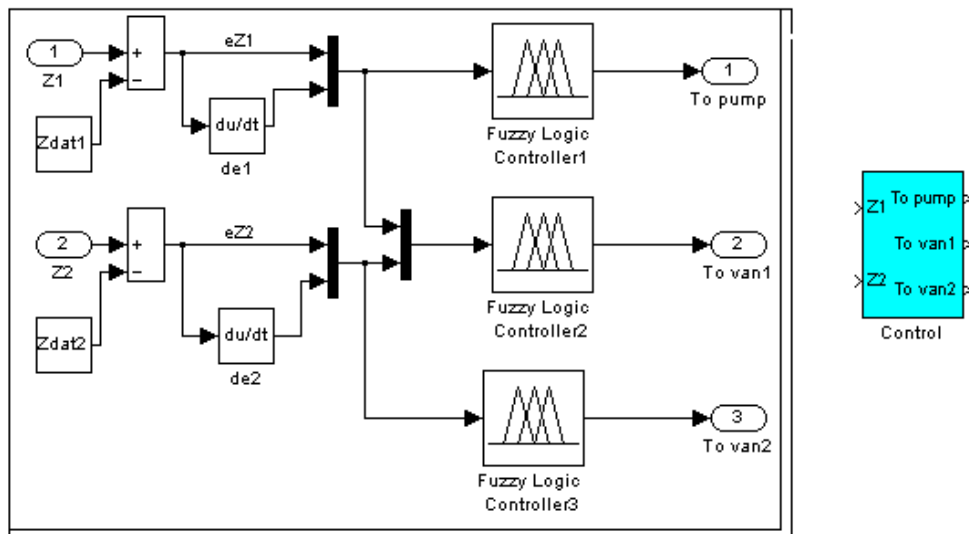


Ba bộ điều khiển mờ (*control*) sẽ điều khiển : bơm, van1, van2 sao cho mực nước ở 2 bồn đạt giá trị đặt trước (*set*).

◆ Sơ đồ simulink:



◆ Sơ đồ khối điều khiển:



◆**Thiết lập hệ thống điều khiển mờ :**

•Xác định các ngõ vào/ra :

- + Có 4 ngõ vào gồm : sai lệch $e1, e2$; đạo hàm sai lệch $de1, de2$
- + Có 3 ngõ ra gồm : $control1, control2, control3$

•Xác định biến ngôn ngữ :

Sai lệch $E = \{âm\ lớn, âm\ nhỏ, bằng\ không, dương\ nhỏ, dương\ lớn\}$
 $E = \{NB, NM, ZR, PM, PB\}$

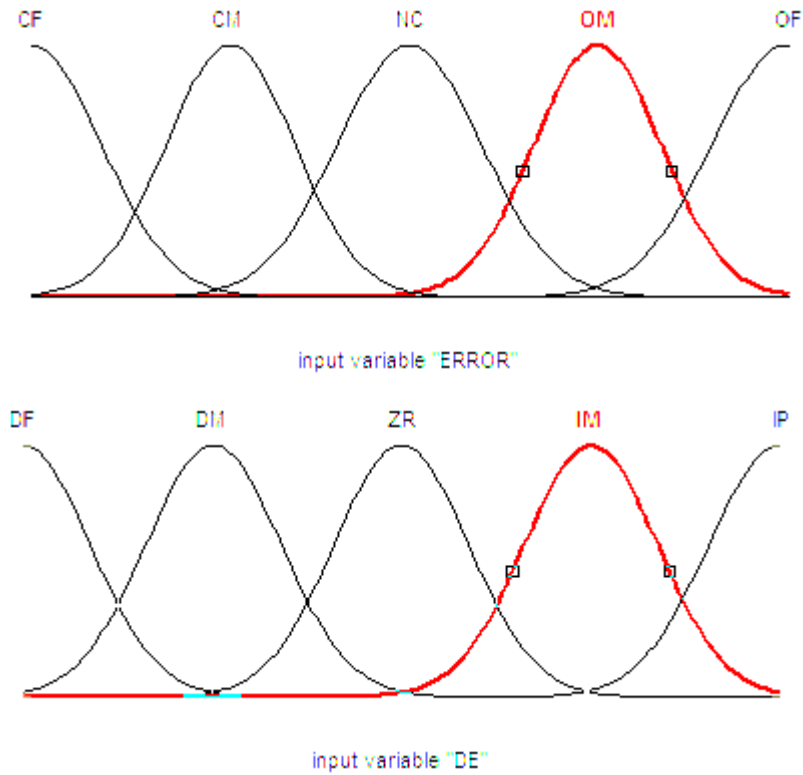
Đạo hàm $D = \{giảm\ nhanh, giảm\ vừa, không\ đổi, tăng\ vừa, tăng\ nhanh\}$
 $D = \{DF, DM, ZR, IM, IP\}$

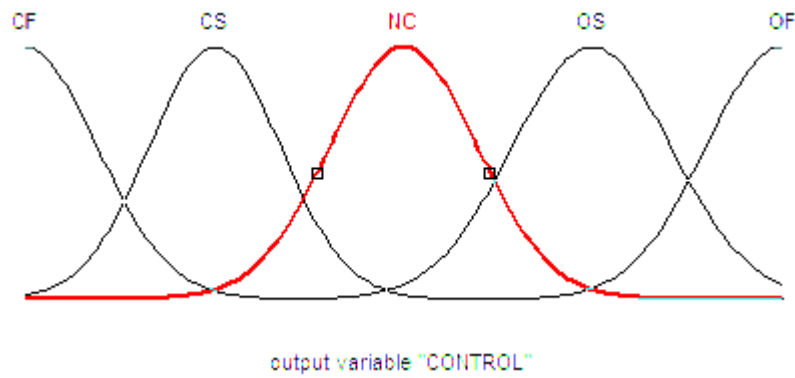
Điều khiển $C = \{đóng\ nhanh, đóng\ chậm, không\ đổi, mở\ chậm, mở\ nhanh\}$
 $C = \{CF, CS, NC, OS, OF\}$

•Luật điều khiển :

+ Khối “*controller1*” và “*controller2*” :

(Hai khối này chỉ khác nhau ở luật hợp thành)





Luật hợp thành mờ Max – Min

DE	<i>Khối controller1</i>				
ERROR	DB	DM	ZR	IM	IB
NB	OF		OF		NC
NM			OS		
ZR	OF	OS	NC	CS	CF
PM			CS		
PB	NC		CF		CF
DE	<i>Khối controller2</i>				
ERROR	DB	DM	ZR	IM	IB
NB	CF		CF		NC
NM			CS		
ZR	CF	CS	NC	OS	OF
PM			OS		
PB	NC		OF		OF

+ Khối “control3”

Đây là khối điều tiết lưu lượng cho bồn 2, ta đưa ra mức ưu tiên như sau :
 Khi sai lệch bồn 1 lớn thì van2 sẽ điều tiết để sai lệch này nhỏ rồi mới đến bồn 2.

If $error1=NB$ and $del=DB$ Then $control=CF$

If $error1=NB$ and $del=DM$ Then $control=CS$

If $error1=NB$ and $del=ZR$ Then $control=CS$

If $error1=NM$ and $del=DB$ Then $control=CS$

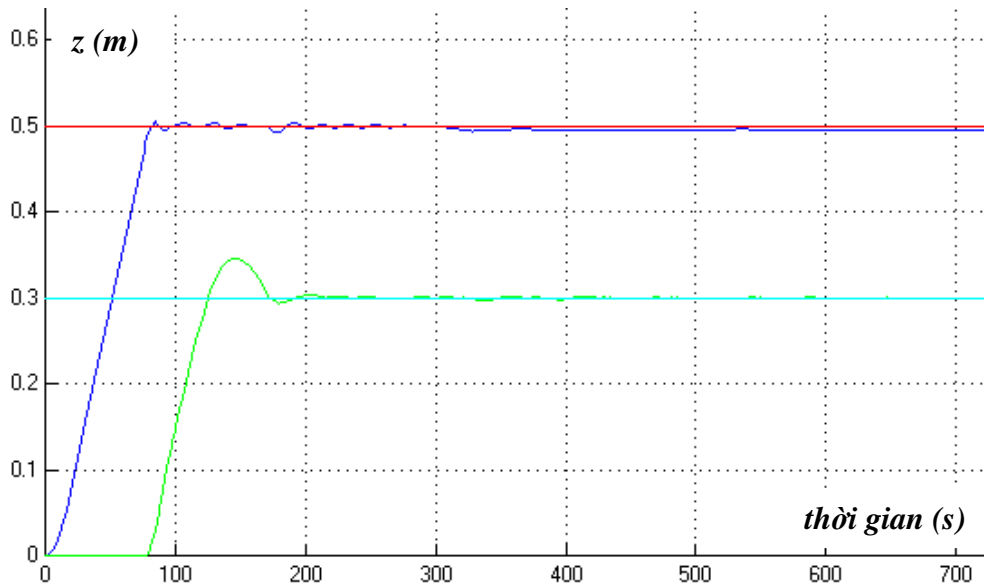
If $error1=PB$ and $de1=IB$ Then $control=OF$
If $error1=PB$ and $de1=IM$ Then $control=OF$
If $error1=PB$ and $de1=ZR$ Then $control=OF$
If $error1=PM$ and $de1=IB$ Then $control=OF$
If $error1 \neq NB$ and $error2=NB$ and $de1 \neq DB$ and $de2=DB$ Then $control=OF$
If $error1 \neq NB$ and $error2=NB$ and $de1 \neq DB$ and $de2=DM$ Then $control=OF$
If $error1 \neq NB$ and $error2=NB$ and $de1 \neq DB$ and $de2=ZR$ Then $control=OF$
If $error1 \neq NB$ and $error2=NM$ and $de1 \neq DB$ and $de2=DB$ Then $control=OS$
If $error1 \neq NB$ and $error2=NM$ and $de1 \neq DB$ and $de2=DM$ Then $control=OS$
If $error1 \neq PB$ and $error2=PB$ and $de1 \neq IB$ and $de2=IB$ Then $control=CF$
If $error1 \neq PB$ and $error2=PB$ and $de1 \neq IM$ and $de2=IB$ Then $control=CS$

• Kết quả đáp ứng với các thông số hệ thống :

- Chiều cao bồn $height=1m$
- Diện tích đáy $area = 0.125m^2$
- Lưu lượng max $pump\ maxflow = 1lit/s$
- Diện tích ống dẫn $pipe\ area = 0.001m^2$

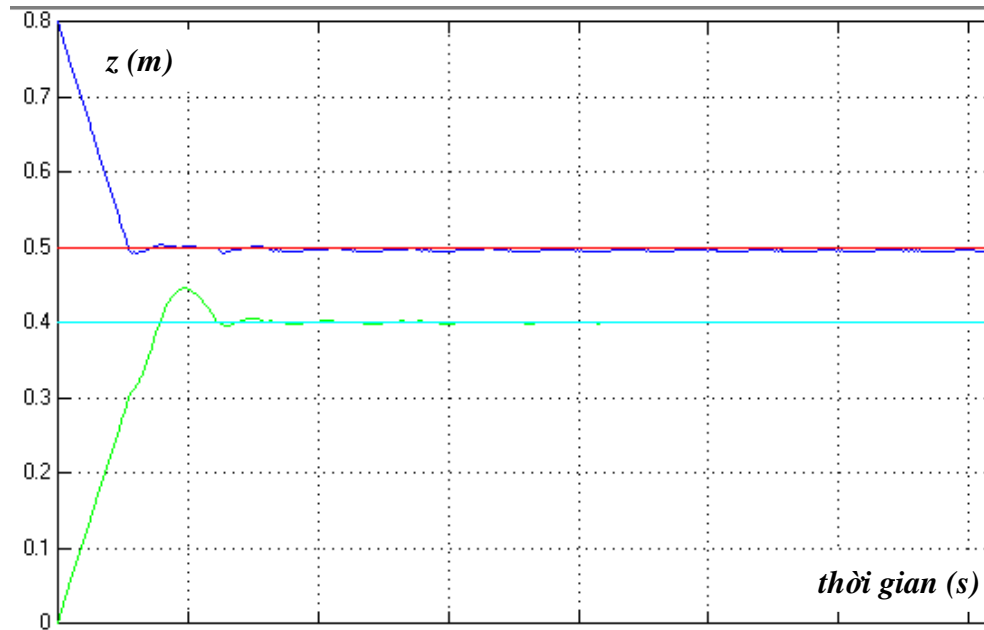
mức nước đặt $Z_{dat}=[0.5 \quad 0.3]$

mức nước ban đầu $Z_{init}=[0 \quad 0]$



mức nước đặt $Z_{dat}=[0.5 \quad 0.4]$

mức nước ban đầu $Z_{init}=[0.8 \quad 0]$

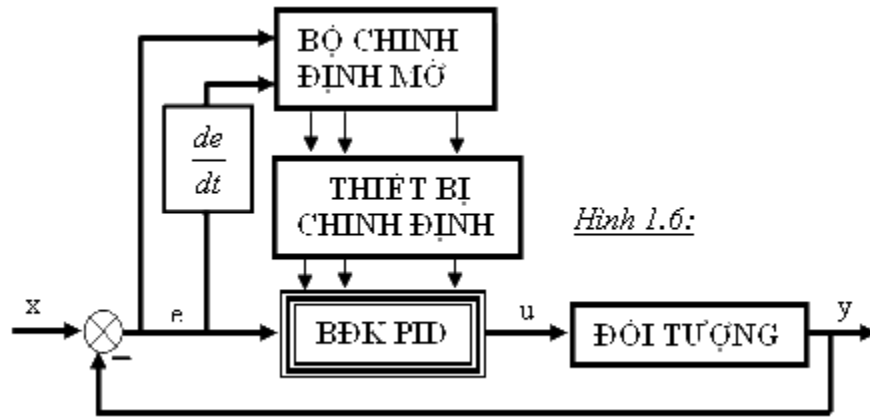


1.2.3. Thiết kế PID mờ

Có thể nói trong lĩnh vực điều khiển, bộ PID được xem như một giải pháp đa năng cho các ứng dụng điều khiển Analog cũng như Digital. Theo một nghiên cứu cho thấy có khoảng hơn 90% các bộ điều khiển được sử dụng hiện nay là bộ điều khiển PID. Bộ điều khiển PID nếu được thiết kế tốt có khả năng điều khiển hệ thống với chất lượng quá độ tốt (đáp ứng nhanh, độ vọt lố thấp) và triệt tiêu được sai số xác lập.

Việc thiết kế bộ PID kinh điển thường dựa trên phương pháp Zeigler-Nichols, Offerein, Reinish ... Tuy nhiên nếu đối tượng điều khiển là phi tuyến thì bộ điều khiển PID kinh điển không thể đảm bảo chất lượng điều khiển tại mọi điểm làm việc. Do đó để điều khiển các đối tượng phi tuyến ngày nay người ta thường dùng kỹ thuật hiệu chỉnh PID mờ (dựa trên phạm mờ), đây chính là cơ sở của thiết kế PID mờ hay PID thích nghi.

1. Sơ đồ điều khiển sử dụng PID mờ :



Hình 1.6:

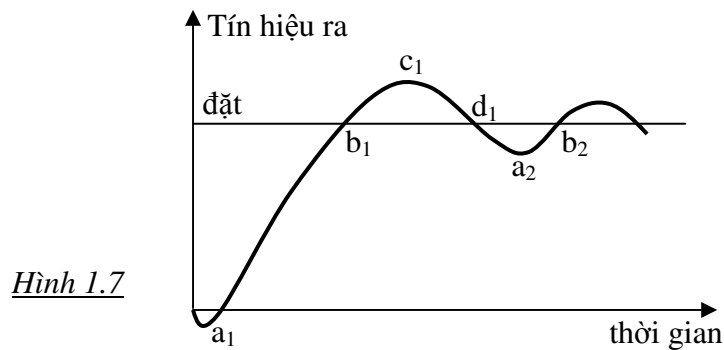
Mô hình toán của bộ PID:

$$u(t) = K_p e(t) + K_I \int_0^t e(x) dx + K_D \frac{de(t)}{dt}$$

$$G_{PID}(s) = K_p + \frac{K_I}{s} + K_D s$$

Các tham số K_p , K_I , K_D được chỉnh định theo từng bộ điều khiển mờ riêng biệt dựa trên sai lệch $e(t)$ và đạo hàm $de(t)$. Có nhiều phương pháp khác nhau để chỉnh định bộ PID (xem các phần sau) như là dựa trên phiếm hàm mục tiêu, chỉnh định trực tiếp, chỉnh định theo Zhao, Tomizuka và Isaka ... Nguyên tắc chung là bắt đầu với các trị K_p , K_I , K_D theo Zeigler-Nichols, sau đó dựa vào đáp ứng và thay đổi dần để tìm ra hướng chỉnh định thích hợp.

2. Luật chỉnh định PID:



Hình 1.7

+ Lân cận a_1 ta cần luật ĐK mạnh để rút ngắn thời gian lên, do vậy chọn: K_p lớn, K_D nhỏ và K_I nhỏ.

- + Lân cận b_1 ta tránh vọt lố lớn nên chọn: K_P nhỏ, K_D lớn, K_I nhỏ.
- + Lân cận c_1 và d_1 giống như lân cận a_1 và b_1 .

Ví dụ 1.5: Ví dụ ứng dụng Matlab

Xây dựng bộ PID mờ để điều khiển lò nhiệt. Hàm truyền lò nhiệt theo

Zeigler-Nichols : $G(s) = \frac{Ke^{-Ts}}{Ls + 1}$, tuyến tính hoá $G(s) = \frac{K}{(Ts + 1)(Ls + 1)}$.

Các bước thiết kế :

1. Xác định biến ngôn ngữ:

- Đầu vào : 2 biến

+ Sai lệch: $ET = \text{Đo} - \text{Đặt} / \text{Đặt} - \text{Đo}$

+ Tốc độ tăng $DET = \frac{ET(i+1) - E(i)}{T}$, với T là chu kỳ lấy mẫu.

- Đầu ra : 3 biến

+ K_P hệ số tỷ lệ $(K_p^0 = \frac{K_p \text{ max} - K_p}{K_p - K_p \text{ min}})$

+ K_I hệ số tích phân

+ K_D hệ số vi phân

- Số lượng biến ngôn ngữ

ET = { âm nhiều, âm vừa, âm ít, zero, dương ít, dương vừa, dương nhiều }

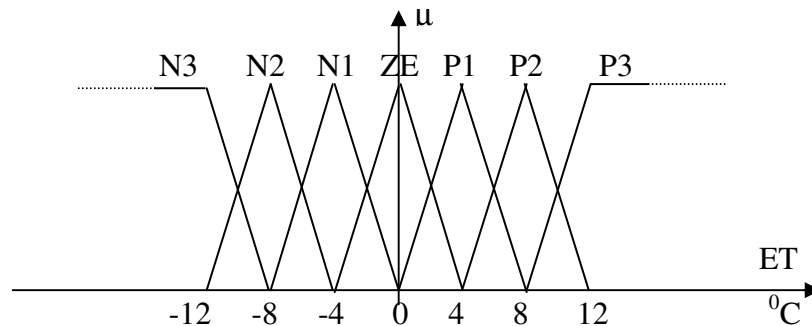
ET = { N3, N2, N1, ZE, P1, P2, P3 }

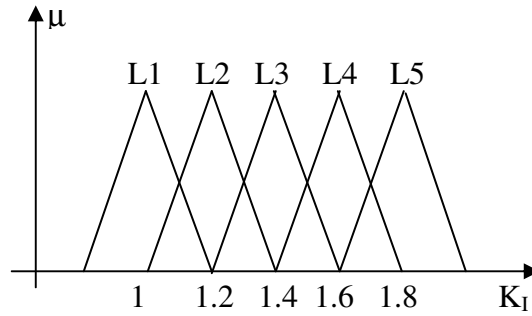
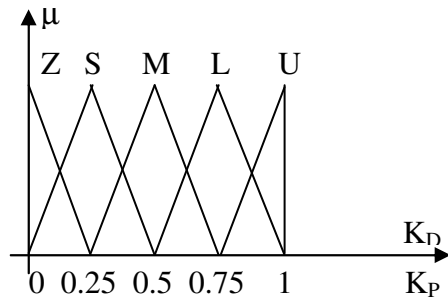
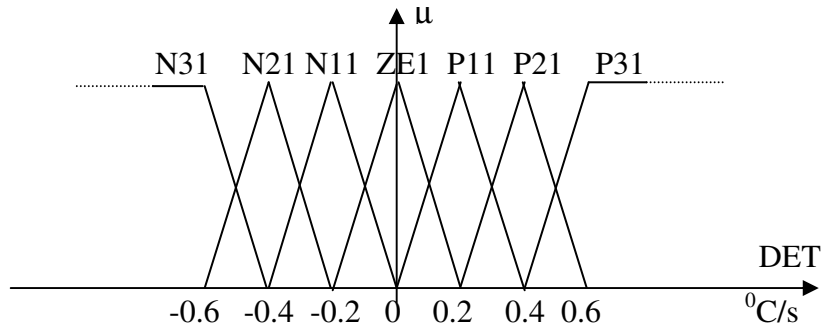
DET = { âm nhiều, âm vừa, âm ít, zero, dương ít, dương vừa, dương nhiều }

DET = { N31, N21, N11, ZE1, P11, P21, P31 }

K_P/K_D = { zero, nhỏ, trung bình, lớn, rất lớn } = { Z, S, M, L, U }

K_I = { mức 1, mức 2, mức 3, mức 4, mức 5 } = { L1, L2, L3, L4, L5 }





2. Luật hợp thành:

Có tổng cộng là $7 \times 7 \times 3 = 147$ luật IF ... THEN

Luật chỉnh định K_P :

K_P		DET						
		N31	N21	N11	ZE1	P11	P21	P31
ET	N3	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>
	N2	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>
	N1	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>
	ZE	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>
	P1	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>
	P2	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>
	P3	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>

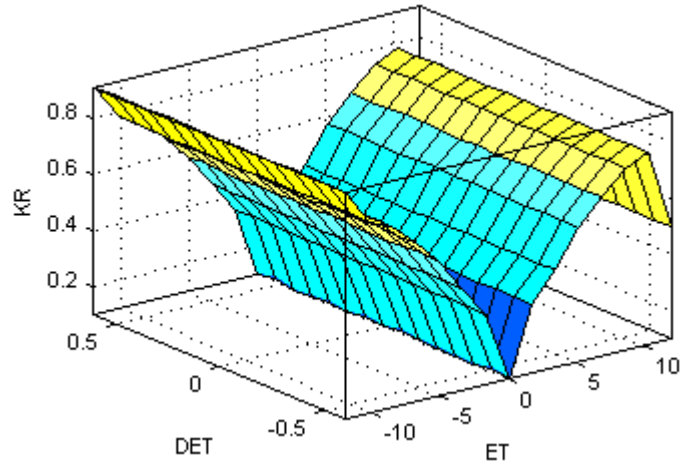
Luật chỉnh định K_D :

K_D		DET						
		N31	N21	N11	ZE1	P11	P21	P31
ET	N3	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>
	N2	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>
	N1	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>
	ZE	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>	<i>U</i>
	P1	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>	<i>L</i>
	P2	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>
	P3	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>

Luật chỉnh định K_I :

K_I		DET						
		N31	N21	N11	ZE1	P11	P21	P31
ET	N3	<i>L1</i>	<i>L1</i>	<i>L1</i>	<i>L1</i>	<i>L1</i>	<i>L1</i>	<i>L1</i>
	N2	<i>L3</i>	<i>L2</i>	<i>L2</i>	<i>L1</i>	<i>L2</i>	<i>L2</i>	<i>L3</i>
	N1	<i>L4</i>	<i>L3</i>	<i>L2</i>	<i>L1</i>	<i>L2</i>	<i>L3</i>	<i>L4</i>
	ZE	<i>L5</i>	<i>L4</i>	<i>L3</i>	<i>L2</i>	<i>L3</i>	<i>L4</i>	<i>L5</i>
	P1	<i>L4</i>	<i>L3</i>	<i>L2</i>	<i>L1</i>	<i>L2</i>	<i>L3</i>	<i>L4</i>
	P2	<i>L3</i>	<i>L2</i>	<i>L2</i>	<i>L1</i>	<i>L2</i>	<i>L2</i>	<i>L3</i>
	P3	<i>L1</i>	<i>L1</i>	<i>L1</i>	<i>L1</i>	<i>L1</i>	<i>L1</i>	<i>L1</i>

Biểu diễn luật chỉnh định K_P trong không gian



3. Chọn luật và giải mờ

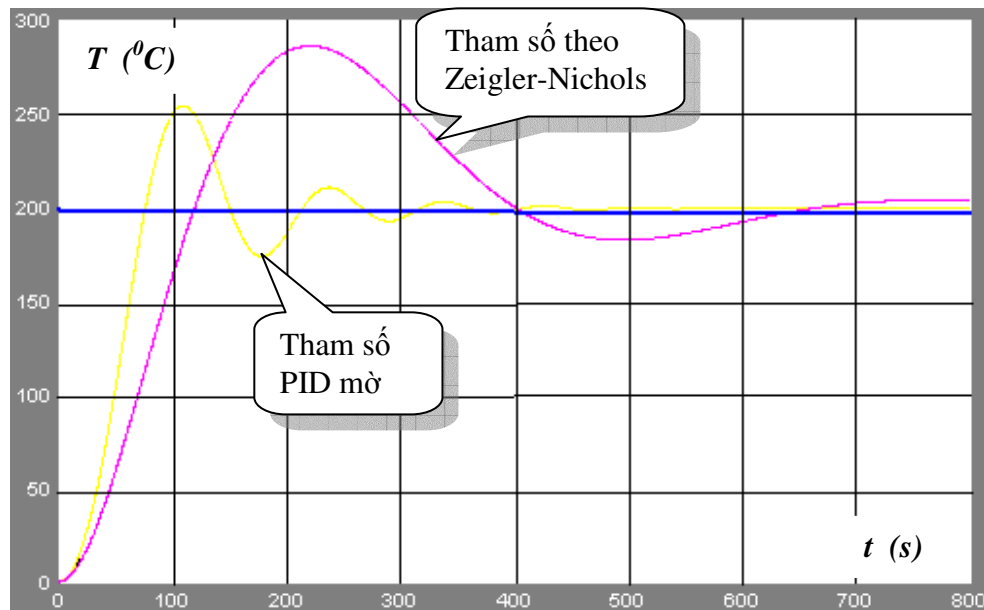
- + Chọn luật hợp thành theo quy tắc Max-Min
- + Giải mờ theo phương pháp trọng tâm.

4. Kết quả mô phỏng

Với các thông số : $K=1$; $T=60$; $L=720$

Từ đây theo Zeigler-Nichols ta tìm ra được bộ ba thông số $\{K_P, K_I, K_D\}$

Đồ thị dưới đây sẽ cho ta thấy sự khác biệt của điều khiển mờ so với điều khiển kinh điển.



1.2.4. Hệ mờ lai

Hệ mờ lai (Fuzzy Hybrid) là một hệ thống điều khiển tự động trong đó thiết bị điều khiển bao gồm: phần điều khiển kinh điển và phần hệ mờ

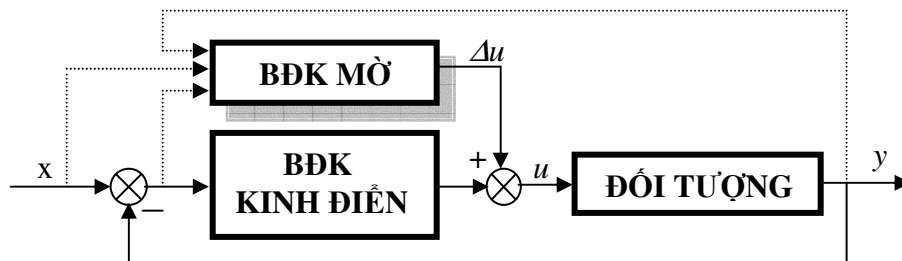
Các dạng hệ mờ lai phổ biến:

1. Hệ mờ lai không thích nghi



Hình 1.8

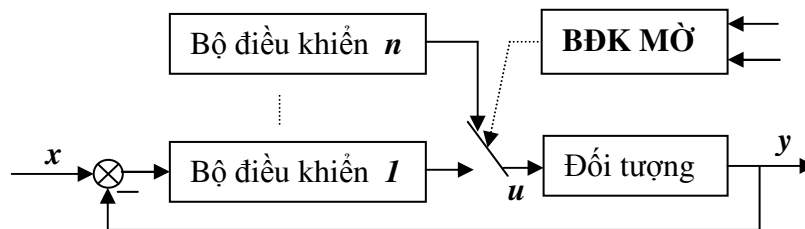
2. Hệ mờ lai cascade



Hình 1.9

3. Công tắc mờ

Điều khiển hệ thống theo kiểu chuyển đổi khâu điều khiển có tham số đòi hỏi thiết bị điều khiển phải chứa đựng tất cả các cấu trúc và tham số khác nhau cho từng trường hợp. Hệ thống sẽ tự chọn khâu điều khiển có tham số phù hợp với đối tượng.

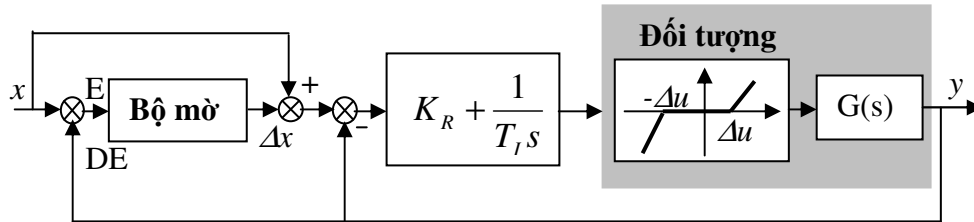


Hình 1.10

Ví dụ 1.6:

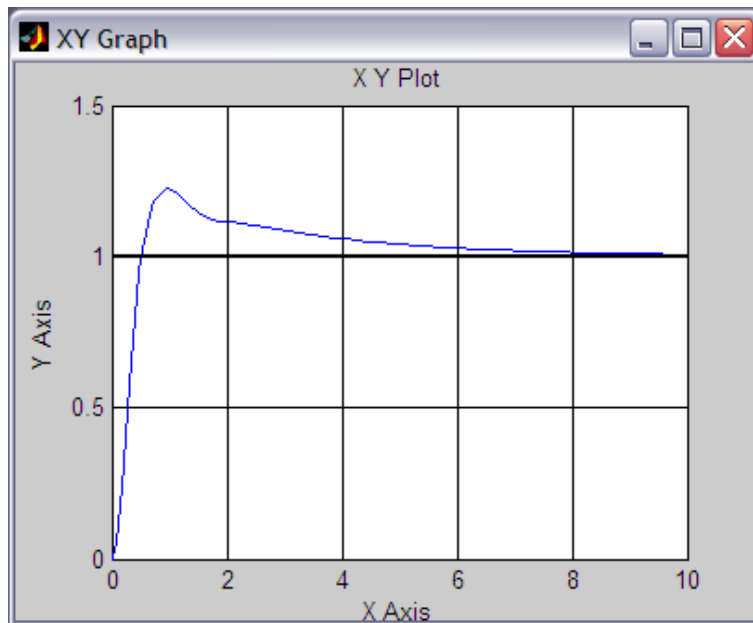
Hãy xét sự khác biệt khi sử dụng bộ tiền xử lý mờ để điều khiển đối tượng gồm khâu có vùng không nhạy là Δu nối tiếp với khâu $G(s) = \frac{K}{s(1+0.2s)}$.

Chọn BDK **PI** với tham số $K_P = 10$, $T_I = 0.3\text{sec}$.



Sử dụng **Simulink** kết hợp với **toolbox FIS Editor** của Matlab để mô phỏng hệ thống trên.

Đáp ứng hệ thống khi không có bộ mờ:



Thử với các giá trị Δu và K khác nhau cho thấy đặc tính động của hệ sẽ xấu đi khi vùng chết rộng hoặc hệ số khuếch đại lớn. Để hiệu chỉnh đặc tính động của hệ thống ta đưa vào bộ lọc mờ như hình vẽ ở trên.

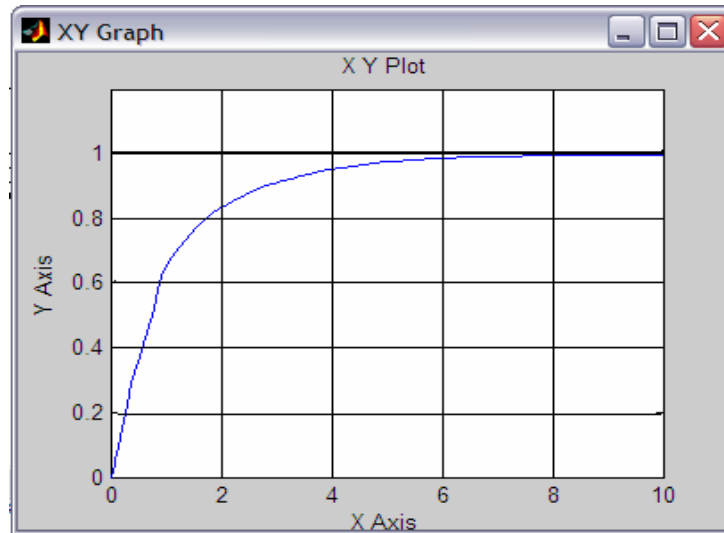
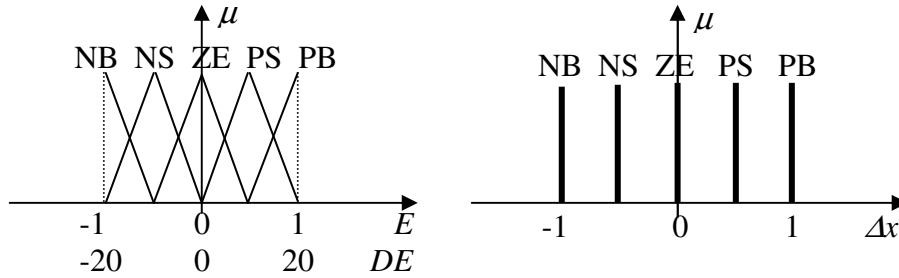
Xây dựng luật điều khiển với 2 đầu vào và một đầu ra như sau:

		DE					
		Δx	NB	NS	ZE	PS	PB
E	NB				NB	NS	
	NS			NS	NS	NS	ZE
	ZE	NB	NS	ZE	PS	PB	
	PS		PS	PS	PS	PS	
	PB		PB	PS	PB	PB	

Tất cả 18 luật có khuôn dạng như sau:

Nếu $E = x1$ và $DE = x2$ Thì $\Delta x = x3$

Trong đó $x1, x2, x3 \in \{NB, NS, ZE, PS, PB\}$



Từ hai đồ thị trên ta thấy được bộ mờ đã cải thiện rất tốt đặc tính động của hệ thống. Thử với nhiều Δu khác nhau ta sẽ thấy đáp ứng hầu như không phụ thuộc vào Δu .

1.2.5. Ứng dụng điều khiển mờ trong thiết kế hệ thống

- Điều khiển mờ không thích nghi (Nonadaptive Fuzzy Control)
- Điều khiển mờ thích nghi (Adaptive Fuzzy Control)

Điều khiển mờ thích nghi sẽ được xem xét ở chương 3 (Điều khiển thích nghi). Ở đây, chúng ta chỉ nghiên cứu về Điều khiển mờ không thích nghi.

1. Bộ điều khiển mờ tuyến tính ổn định SISO

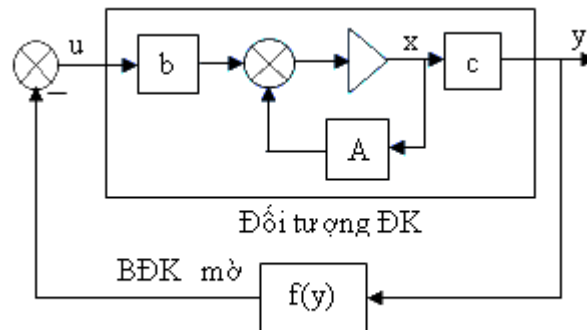
Phương trình biến trạng thái của hệ SISO

$$\dot{x}(t) = Ax(t) + bu(t)$$

$$y(t) = cx(t)$$

$$u(t) = -f[y(t)]$$

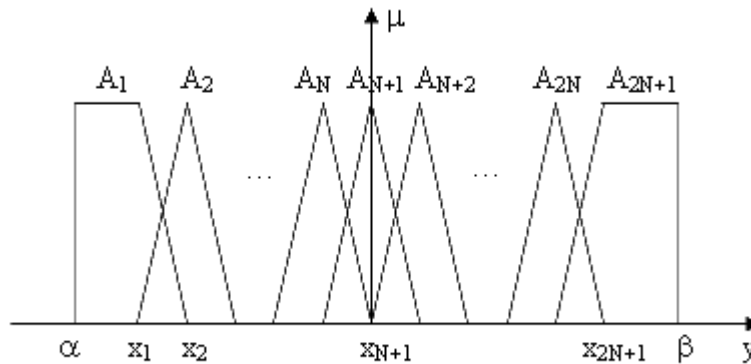
Thay phương trình cuối vào hai phương trình trên ta được hệ mờ vòng kín:



Hình 1.11: Cấu trúc hệ SISO

Thiết kế BDK mờ ổn định SISO

- Bước 1: Giả sử $y(t)$ có miền giá trị là khoảng $U=[\alpha \ \beta]$, chia U ra $2N+1$ khoảng A_k như hình vẽ bên dưới:



Hình 1.12: Hàm thuộc của BDK

- Bước 2: Thành lập $2N+1$ luật mờ IF – THEN có khuôn dạng

$$IF \ y = A_k \ THEN \ u = B_k$$

trong đó $k = 1, 2, \dots, 2N+1$ và trọng tâm \bar{y} của khoảng mờ B_k là:

$$\bar{y} \begin{cases} \leq 0 \rightarrow k = 1, \dots, N \\ = 0 \rightarrow k = N + 1 \\ \geq 0 \rightarrow k = N + 2, \dots, 2N + 1 \end{cases} \quad (1.5)$$

- Bước 3: Chọn luật hợp thành tích, giải mờ theo phương pháp trung bình trọng số, ta có luật điều khiển như sau:

$$u = -f(y) = -\frac{\sum_{k=1}^{2N+1} \bar{y} \mu_{A_k}(y)}{\sum_{k=1}^{2N+1} \mu_{A_k}(y)}$$

với \bar{y} thỏa (1.5) và $\mu_{A_k}(y)$ được nêu trong Hình 1.12.

2. Bộ ĐK mờ tuyến tính ổn định MIMO

Phương trình biến trạng thái của hệ MIMO:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (1.6)$$

Giả sử hệ có m đầu vào và m đầu ra thì $u(t) = (u_1(t), \dots, u_m(t))^T$ có dạng :

$$u_k(t) = -f_k[y(t)] \quad (1.7)$$

với $k=1, 2, \dots, m$ và $f_k[y(t)]$ là hệ mờ m đầu vào 1 đầu ra.

Mô hình hệ thống có cấu trúc như Hình 1.11, nhưng thay cho các số b, c bởi các ma trận B, C , hàm vô hướng f bởi vectơ $f = (f_1, f_2, \dots, f_m)^T$.

Thiết kế BDK mờ ổn định MIMO

- Bước 1: Giả sử đầu ra $y_k(t)$ có miền giá trị là $U_k = [\alpha_k \ \beta_k]$, với $k=1, \dots, m$. Chia U_k ta $2N_k+1$ khoảng $A_k^{l_i}$ và thiết lập hàm thuộc như Hình F.2

- Bước 2: Thành Lập m nhóm luật mờ IF – THEN, nhóm thứ k chứa $\prod_{i=1}^m (2N_k + 1)$ luật dạng:

$$IF \ y_1 = A_1^{l_1} \ And \ \dots \ And \ y_m = A_m^{l_m}, \ THEN \ u = B_k^{l_1 \dots l_m}$$

Trong đó $l_i=1, 2, \dots, 2N_k+1$; $k=1, 2, \dots, m$ và trọng số $\bar{y}_k^{l_1 \dots l_m}$ của tập mờ $B_k^{l_1 \dots l_m}$ được chọn như sau:

$$\bar{y}_k^{l_1 \dots l_m} \begin{cases} \leq 0 \rightarrow l_k = 1, 2, \dots, N_k \\ = 0 \rightarrow l_k = N_k + 1 \\ \geq 0 \rightarrow l_k = N_k + 2, \dots, 2N_k + 1 \end{cases} \quad (1.8)$$

• Bước 3: Chọn luật hợp thành tích, giải mờ theo phương pháp trung bình trọng số, ta được luật điều khiển:

$$u_k = -f_k(y) = -\frac{\sum_{l_1=1}^{2N_1+1} \dots \sum_{l_m=1}^{2N_m+1} \bar{y}_k^{l_1 \dots l_m} (\prod_{i=1}^m \mu_{A_i^{l_i}}(y_i))}{\sum_{l_1=1}^{2N_1+1} \dots \sum_{l_m=1}^{2N_m+1} (\prod_{i=1}^m \mu_{A_i^{l_i}}(y_i))} \quad (1.9)$$

với $k=1, 2, \dots, m$.

3. Bộ điều khiển mờ tối ưu

Phương trình trạng thái

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ x(0) &= x_0 \end{aligned} \quad (1.10)$$

với $x \in \mathbb{R}^n$ và $u \in \mathbb{R}^m$, và chỉ tiêu chất lượng dạng toàn phương:

$$J = x^T(T)Mx(T) + \int_0^T [x^T(t)Qx(t) + u^T(t)Ru(t)] dt \quad (1.11)$$

với $M \in \mathbb{R}^{n \times n}$, $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$ là các ma trận xác định dương.

Ta xác định $u(t)$ dạng như (1.9), với $u(t) = (u_1, u_2, \dots, u_m)^T$

$$u_k = -f_k(x) = -\frac{\sum_{l_1=1}^{2N_1+1} \dots \sum_{l_n=1}^{2N_n+1} \bar{y}_k^{l_1 \dots l_n} (\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i))}{\sum_{l_1=1}^{2N_1+1} \dots \sum_{l_n=1}^{2N_n+1} (\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i))} \quad (1.12)$$

Chúng ta cần xác định thông số $\bar{y}_k^{l_1 \dots l_n}$ để cực tiểu J.

Ta định nghĩa hàm mờ cơ sở $b(x) = (b_1(x), \dots, b_N(x))^T$ với:

$$b_l(x) = \frac{\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{2N_1+1} \dots \sum_{l_n=1}^{2N_n+1} (\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i))} \quad (1.13)$$

với $l_i = 1, 2, \dots, 2N_i+1$; $l = 1, 2, \dots, N$ và $N = \prod_{i=1}^n (2N_i + 1)$. Ta định nghĩa ma trận thông số $\Theta \in \mathbb{R}^{m \times N}$ như sau :

$$\Theta = [-\Theta_1^T, -\Theta_2^T, \dots, -\Theta_m^T]^T \quad (1.14)$$

với $\Theta_k^T \in R^{1 \times N}$ chứa N thông số $\bar{y}_k^{1 \dots n}$, có bậc giống như $b_l(x)$. Ta viết lại tín hiệu điều khiển mờ dạng $u = (u_1, u_2, \dots, u_m)^T = (-f_1(x), \dots, -f_m(x))^T$ như sau:

$$u = \Theta b(x) \quad (1.15)$$

Giờ ta giả sử $\Theta = \Theta(t)$. Thay (1.15) vào (1.10) và (1.11) ta được :

$$\dot{x}(t) = Ax(t) + B\Theta(t)b[x(t)] \quad (1.16)$$

và hàm chỉ tiêu chất lượng là :

$$J = x^T(T)Mx(T) + \int_0^T [x^T(t)Qx(t) + b^T(x(t))\Theta^T(t)R\Theta(t)b(x(t))]dt \quad (1.17)$$

Vì vậy vấn đề cần giải quyết bây giờ là xác định $\Theta(t)$ tối ưu để cực tiểu hoá J.

Xét hàm Hamilton:

$$H(x, \Theta, p) = x^T Qx + b^T(x)\Theta^T R\Theta b(x) + p^T [Ax + B\Theta b(x)] \quad (1.18)$$

Ta có:
$$\frac{\partial H}{\partial \Theta} = 2R\Theta b(x)b^T(x) + B^T p b^T(x) = 0$$

Suy ra :
$$\Theta = -\frac{1}{2} R^{-1} B^T p b^T(x) [b(x)b^T(x)]^{-1} \quad (1.19)$$

Thay (1.19) vào (1.18) ta được:

$$H^*(x, p) = x^T Qx + p^T Ax + [\alpha^2(x) - \alpha(x)] p^T B R^{-1} B^T p \quad (1.20)$$

trong đó:
$$\alpha(x) = \frac{1}{2} b^T(x) [b(x)b^T(x)]^{-1} b(x) \quad (1.21)$$

Áp dụng nguyên lý cực tiểu Pontryagin ta được:

$$\dot{x} = \frac{\partial H^*}{\partial p} = Ax + 2[\alpha^2(x) - \alpha(x)] B R^{-1} B^T p \quad (1.22)$$

$$\dot{p} = -\frac{\partial H^*}{\partial x} = -2Qx - A^T p - [2\alpha(x) - 1] \frac{\partial \alpha(x)}{\partial x} p^T B R^{-1} B^T p \quad (1.23)$$

Giải hai phương trình vi phân (1.21) và (1.22) ta sẽ được $x^*(t)$ và $p^*(t)$, từ đó ta xác định được:

$$\Theta^*(t) = -\frac{1}{2} R^{-1} B^T p^*(t) b^T(x^*(t)) [b(x^*(t))b^T(x^*(t))]^{-1} \quad (1.24)$$

Và bộ mờ tối ưu sẽ là:

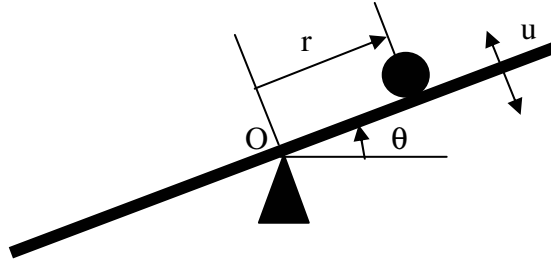
$$u^* = \Theta^*(t)b(x) \quad (1.25)$$

Các bước để thiết kế BDK mờ tối ưu:

- Bước 1: Xác định hàm thuộc $\mu_{A_i^{l_i}}(x_i)$, với $l_i = 1, 2, \dots, 2N_i+1$ và $I = 1, \dots, n$. Chọn dạng hàm thuộc là Gaussian.
- Bước 2: Tính hàm mờ cơ sở $b_i(x)$ theo (1.13) và tính $\alpha(x)$ theo (1.21), xác định trị đạo hàm : $\frac{\partial \alpha(x)}{\partial x}$.
- Bước 3: Giải (1.22) và (1.23) để được $x^*(t)$ và $p^*(t)$, tính $\Theta^*(t)$ theo (1.24) với $t \in [0 \ T]$.
- Bước 4: Xác định BDK mờ tối ưu từ (1.25)

Ví dụ 1.7:

Hãy thiết kế và mô phỏng hệ thống “Quả bóng và đòn bẩy” như hình vẽ sau:



Hình 1.13

Thiết kế BDK mờ để điều khiển quả bóng di chuyển từ điểm gốc O đến mục tiêu (vị trí đặt) cách O khoảng r . Chọn biên trạng thái như sau:

$$x = (r, \dot{r}, \theta, \dot{\theta})^T = (x_1, x_2, x_3, x_4)^T \text{ và } y = r = x_1$$

Phương trình biên trạng thái được chọn là:

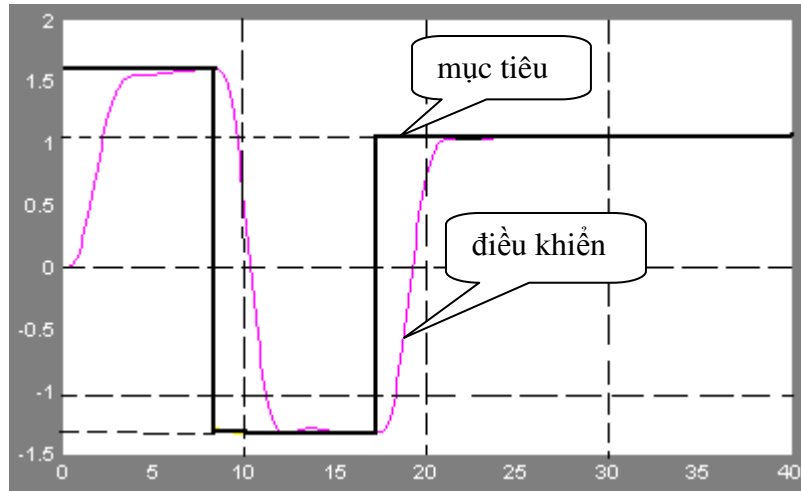
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \alpha(x_1 x_4^2 - \beta \sin x_3) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

Chọn $M=0$, $Q=I$, $R=I$, $N_i=2$ với $i=1,2,3,4$. Chọn hàm thuộc dạng:

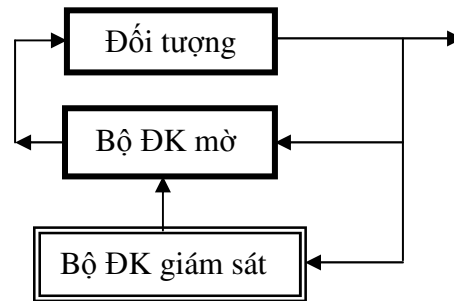
$$\mu_{A_i^{l_i}}(x_i) = \exp[-2(x_i - \bar{x}_i^{l_i})^2]$$

Trong đó $i=1, 2, 3, 4$; $l_i=1, 2, 3, 4, 5$ và $\bar{x}_i^{l_i} = a_i + b_i(l_i - 1)$ với $a_1 = a_2 = -2$, $a_3 = a_4 = -1$, $b_1 = b_2 = 1$, $b_3 = b_4 = 0.5$.

Chọn $\alpha = 0.7143$, $\beta = 9.81$. Kết quả mô phỏng với 3 mục tiêu khác nhau:



4. Điều khiển mờ có hệ thống giám sát



Hình 1.14

- Thiết kế bộ giám sát

Xét hệ thống phi tuyến được cho bởi phương trình vi phân:

$$\dot{x}^{(n)} = f(x, \dot{x}, \dots, x^{(n-1)}) + g(x, \dot{x}, \dots, x^{(n-1)})u \quad (1.26)$$

trong đó $x = (x, \dot{x}, \dots, x^{(n-1)})^T$ là vectơ trạng thái ra, $u \in R$ là tín hiệu điều khiển, f và g là các hàm chưa biết, giả thiết $g > 0$. Giả sử ta đã có BDK mờ:

$$u = u_{fuz}(x)$$

Giả sử $|x(t)| \leq M_x, \forall x$ với $M_x = \text{const}$. Khi thêm bộ giám sát thì tín hiệu điều khiển hệ thống sẽ là:

$$u = u_{fuzz}(x) + I^*u_s(x) \quad (1.27)$$

trong đó $I^* = I$ nếu $|x(t)| \geq M_x, I^* = 0$ nếu $|x(t)| < M_x$. Ta cần thiết kế bộ giám sát $u_s(t)$.

Thay (1.27) vào (1.26) ta được:

$$x^{(n)} = f(x) + g(x)u_{fuzz}(x) + g(x)I^*u_s(x) \quad (1.28)$$

Giả sử ta luôn xác định được hai hàm $f^U(x)$ và $g_L(x)$ sao cho $|f(x)| \leq f^U(x)$ và $0 < g_L(x) \leq g(x)$.

Đặt :
$$u^* = \frac{1}{g(x)} [-f(x) - k^T x] \quad (1.29)$$

Trong đó $k = (k_n, k_{n-1}, \dots, k_1)^T \in R$. Ta viết lại (1.28) như sau:

$$x^{(n)} = -k^T x + g[u_{fuzz} - u^* + I^*u_s] \quad (1.30)$$

Đặt
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ -k_n & -k_{n-1} & \dots & \dots & \dots & -k_2 & -k_1 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ \dots \\ 0 \\ g \end{bmatrix}$$

Viết (1.30) dạng vectơ :

$$\dot{x} = Ax + b[u_{fuzz} - u^* + I^*u_s] \quad (1.31)$$

Xét hàm Lyapunov :
$$V = \frac{1}{2} x^T P x \quad (1.32)$$

Trong đó P là ma trận đối xứng xác định dương thoả phương trình Lyapunov :

$$A^T P + PA = -Q \quad (1.33)$$

Từ (1.31), (1.33) và xét trường hợp $|x| \geq M_x$, ta có:

$$\dot{V} = -\frac{1}{2}x^T Qx + x^T Pb[u_{fuzz} - u^* + u_s] \leq |x^T Pb|(|u_{fuzz}| + |u^*|) + x^T Pbu_s \quad (1.34)$$

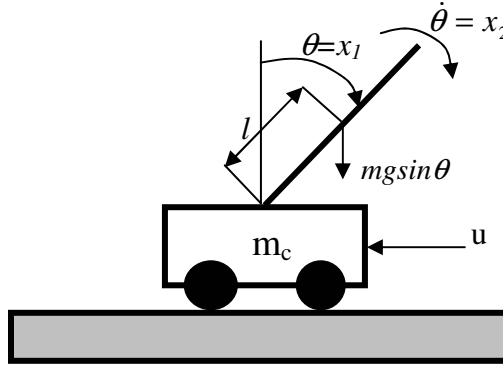
Ta cần tìm u_s để $\dot{V} \leq 0$, kết hợp phương trình trên với (4.6.25) ta được:

$$u_s = -\text{sign}(x^T Pb) \left[\frac{1}{g_L} (f^U + |k^T x|) + |u_{fuzz}| \right] \quad (1.35)$$

Thay (1.35) vào (1.34) ta sẽ được $\dot{V} \leq 0$.

• Ví dụ 1.8:

Thiết kế hệ thống có bộ giám sát để giữ cân bằng cho con lắc ngược.



Hình 1.15

Phương trình trạng thái:

$$\dot{x}_1 = x_2 \quad (1.36)$$

$$\dot{x}_2 = \frac{g \sin x_1 - \frac{mlx_2^2 \cos x_1 \sin x_1}{m_c + m}}{l\left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m}\right)} + \frac{\frac{\cos x_1}{m_c + m}}{l\left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m}\right)} u \quad (1.37)$$

Thiết kế bộ giám sát

Đầu tiên ta tìm f^U và g_L , ta có

$$|f(x_1, x_2)| = \left| \frac{g \sin x_1 - \frac{mlx_2^2 \cos x_1 \sin x_1}{m_c + m}}{l\left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m}\right)} \right| \leq \frac{9.8 + \frac{0.25}{1.1}x_2^2}{\frac{2}{3} - \frac{0.05}{1.1}} = 15.78 + 0.0366x_2^2$$

chọn $f^u(x_1, x_2) = 15.78 + 0.0366x_2^2$

Để con lắc ổn định thì góc $x_1 = \theta \leq 20^\circ$. Suy ra $M_x = 20^\circ$.

$$|g(x_1, x_2)| \geq \frac{\cos 20^\circ}{1.1\left(\frac{2}{3} + \frac{0.05}{1.1} \cos^2 20^\circ\right)} = 1.1$$

chọn $g_L(x_1, x_2) = 1.1$

Chọn các thông số thiết kế như sau:

$$a = \pi/18, k_1 = 2, k_2 = 1, Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

Giải phương trình Lyapunov (1.33) ta được : $P = \begin{bmatrix} 15 & 5 \\ 5 & 15 \end{bmatrix}$

Thiết kế BDK mờ để được $u_{fuzz}(x)$.

Từ (1.35) ta sẽ được BDK có giám sát hệ con lắc ngược.

Dùng simulink của matlab chạy mô phỏng ta sẽ thấy được tính ưu việt khi có và không sử dụng bộ giám sát.

5. Điều khiển mờ trượt

1. Nguyên lý điều khiển trượt

Xét hệ thống phi tuyến

$$\dot{x}^{(n)} = f(X) + g(X)u \quad (1.38)$$

$$y(t) = x(t)$$

trong đó u là tín hiệu điều khiển, x là tín hiệu ra, $X = (x, \dot{x}, \dots, x^{(n-1)})^T$ là vectơ trạng thái. Trong (1.38) $f(X)$ là hàm chưa biết và bị chặn bởi một hàm đã biết:

$$f(X) = \hat{f}(X) + \Delta f(X) \quad (1.39)$$

và $|\Delta f(X)| \leq F(X) \quad (1.40)$

$$0 < g_0 < g(X) < g_1 \quad (1.41)$$

trong đó $\hat{f}(X), F(X)$ đã biết, g_0, g_1 là các hằng số dương.

Đối với mục tiêu điều khiển ổn định hệ thống thì chúng ta cần xác định luật điều khiển hồi tiếp $u = u(X)$ sao cho ngõ ra của hệ thống $x \rightarrow 0$ khi $t \rightarrow \infty$.

Để làm được điều này ta đưa ra hàm trượt sau:

$$S = \frac{d^{n-1}x}{dt^{n-1}} + a_{n-2} \frac{d^{n-2}x}{dt^{n-2}} + \dots + a_1 \frac{dx}{dt} + a_0 x \quad (1.42)$$

trong đó n là bậc của đối tượng.

Các hệ số a_0, a_1, \dots, a_{n-2} phải được chọn sao cho đa thức đặc trưng của phương trình vi phân $S=0$ là đa thức Hurwitz.

Phương trình $S=0$ mô tả một mặt trong không gian trạng thái n chiều gọi là mặt trượt (*Sliding surface*).

Ta cần xác định luật điều khiển u sao cho $S \rightarrow 0$ để có $x \rightarrow 0$.

Đối với điều khiển bám mục tiêu, ta cần xác định luật điều khiển $u = u(X)$ sao cho trạng thái của hệ thống vòng kín sẽ bám theo trạng thái mong muốn

$$X_d = (x_d, \dot{x}_d, \dots, x_d^{(n-1)})^T$$

Gọi e là sai lệch giữa tín hiệu ra và tín hiệu đặt:

$$e = X - X_d = (e, \dot{e}, \dots, e^{(n-1)})^T$$

Mục tiêu điều khiển là triệt tiêu e khi $t \rightarrow \infty$.

Định nghĩa hàm trượt :

$$S(e) = \frac{d^{n-1}e}{dt^{n-1}} + a_{n-2} \frac{d^{n-2}e}{dt^{n-2}} + \dots + a_1 \frac{de}{dt} + a_0 e \quad (1.43)$$

trong đó n là bậc của đối tượng điều khiển, các hệ số a_0, a_1, \dots, a_{n-2} được chọn sao cho đa thức đặc trưng của $S(e)=0$ là đa thức Hurwitz.

Sử dụng phương pháp Lyapunov, chọn hàm V xác định dương như sau:

$$V = \frac{1}{2} S^2 \quad (1.44)$$

$$\Rightarrow \dot{V} = S\dot{S} \quad (1.45)$$

Để \dot{V} xác định âm ta chọn luật điều khiển u sao cho:

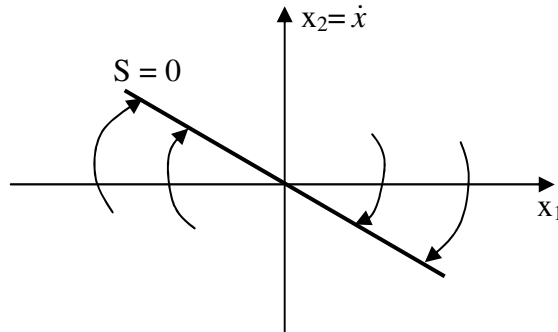
$$\text{Khi } S > 0 \text{ thì } \dot{S} < 0$$

$$\text{Khi } S < 0 \text{ thì } \dot{S} > 0$$

Do vậy với hàm trượt $S(e)$ ta xác định luật điều khiển u thỏa:

$$\frac{dS}{dt} \text{sign}(S) < 0 \quad (1.46)$$

Với luật điều khiển như vậy, hệ thống sẽ ổn định theo tiêu chuẩn Lyapunov, lúc này mọi quỹ đạo trạng thái của hệ thống bên ngoài mặt trượt sẽ được đưa về mặt trượt và duy trì một cách bền vững.



Hình 1.16 Mặt trượt bậc hai

2. Hệ thống điều khiển trượt mờ

Xét hệ thống (1.38), ta cần xác định luật điều khiển u để đưa ngõ ra của hệ thống bám theo giá trị mong muốn cho trước $y(t) \rightarrow y_d(t)$ hay nói cách khác là $e^{(i)} = (y^{(i)} - y_d^{(i)}) \rightarrow 0, i = 0, 1, \dots, n-1$

Dựa vào đặt tính của bộ điều khiển trượt ta cần thực hiện hai bước sau:

Bước 1: Chọn mặt trượt S

Bước 2: Thiết kế luật điều khiển cho hệ thống rơi vào mặt trượt $S = 0$ và duy trì ở chế độ này mãi mãi.

Gọi $e(t) = (e_1(t), e_2(t), \dots, e_n(t))^T = (e(t), \dot{e}(t), \dots, e^{(n-1)}(t))^T$

Chọn hàm trượt:

$$S(e) = \frac{d^{n-1}e}{dt^{n-1}} + b_{n-2} \frac{d^{n-2}e}{dt^{n-2}} + \dots + b_1 \frac{de}{dt} + b_0 e \quad (1.47)$$

Trong đó b_0, b_1, \dots, b_{n-2} được chọn sao cho nghiệm của đa thức đặc trưng $p^{n-1} + b_{n-2}p^{n-2} + \dots + b_1p + b_0 = 0$ đều nằm bên trái mặt phẳng phức.

Mặt trượt S được cho bởi phương trình $S(e) = 0$, luật điều khiển u được chọn sao cho $\frac{dS}{dt} \text{sign}(S) < 0$.

3. Thiết kế bộ điều khiển mờ trượt bậc hai

Xét hệ thống phi tuyến bậc hai sau:

$$\ddot{x} = f(X) + g(X)u \quad (1.48)$$

$$y = x \quad (1.49)$$

trong đó $X = (x, \dot{x})^T$ là vectơ trạng thái, u là ngõ vào điều khiển $y(t)$ là ngõ ra của hệ thống.

Mục tiêu của điều khiển là xác định luật điều khiển u để ngõ ra của hệ thống bám theo quỹ đạo mong muốn $y_d(t)$ với sai số nhỏ nhất.

Luật điều khiển u gồm 2 thành phần:

$$u = u_{eq} + u_s \quad (1.50)$$

Thành phần u_{eq} được thiết kế như sau:

$$u_{eq}(t) = \frac{1}{\hat{g}} \left[-\hat{f}(X, t) + \ddot{y}_d(t) - \lambda e \right], \quad (\lambda > 0) \quad (1.51)$$

Thành phần u_s được chọn là:

$$u_s(t) \geq \frac{1}{\hat{g}} \left[\alpha(F(X, t) + \eta) + (\alpha - 1)|u_{eq}(t)| \right] \quad (1.52)$$

Trong đó $\hat{f}(X, t)$ là giá trị ước lượng của $f(X, t)$

$F(X, t)$ là cận trên của sai số ước lượng

$$0 < g_0 < g(X) < g_1$$

$$\hat{g} = \sqrt{g_0 g_1}$$

$$\alpha = \sqrt{\frac{g_1}{g_0}}$$

Luật điều khiển mờ được thiết kế như sau:

$$u(t) = \begin{cases} u^-(t), \rightarrow S < 0 \\ u^+(t), \rightarrow S > 0 \end{cases} \quad (1.53)$$

Trong đó:

$$\begin{aligned} u^-(t) &= u_{eq}(t) + k\hat{g}^{-1} \left[\alpha(F(X, t) + \eta) + (\alpha - 1)|u_{eq}(t)| \right] \\ u^+(t) &= u_{eq}(t) - k\hat{g}^{-1} \left[\alpha(F(X, t) + \eta) + (\alpha - 1)|u_{eq}(t)| \right] \end{aligned} \quad (1.54)$$

$(k \geq 1)$

Hệ qui tắc mờ có khuôn dạng như sau:

$$\begin{aligned} R^1 : \text{Nếu } S < 0 \text{ Thì } u^1(t) &= u^-(t) \\ R^2 : \text{Nếu } S > 0 \text{ Thì } u^2(t) &= u^+(t) \end{aligned} \quad (1.55)$$

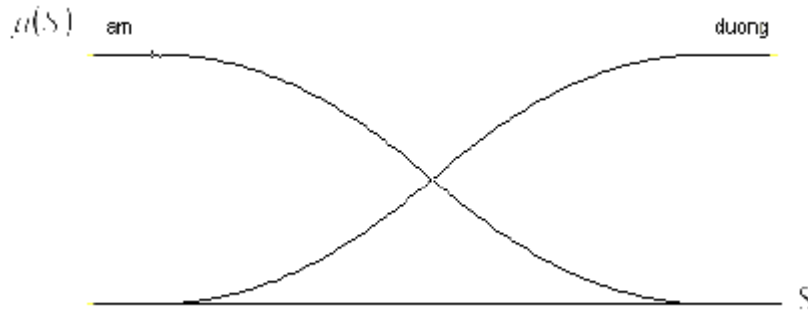
Chọn luật hợp thành tích, giải mờ theo phương pháp trọng tâm, luật điều khiển u được xác định như sau:

$$u(t) = \frac{\sum_{i=1}^r \beta_i(S) u^i(t)}{\sum_{i=1}^r \beta_i(S)} \quad (1.56)$$

Với r : số luật mờ

$$\beta_i(S) = \prod_{j=1}^n \mu_{A_j^i}(S)$$

$\mu_{A_j^i}(S)$ là hàm thuộc có dạng Gaussian như sau:

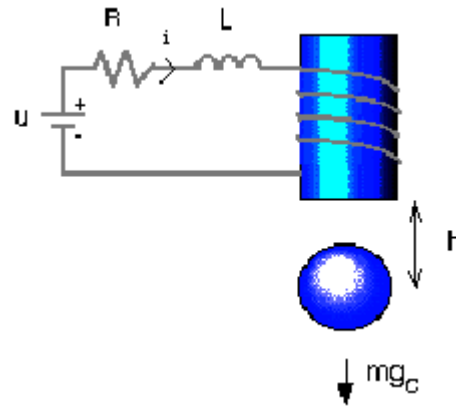


Hình 1.17 : Dạng hàm thuộc để mờ hóa

4. Thiết kế BDK mờ trượt cho hệ thống nâng vật trong từ trường

Mô hình:

Hình 1.18 minh họa một hệ thống nâng vật bằng từ trường, từ trường được tạo ra từ cuộn dây quấn quanh lõi thép, cuộn dây nhận áp điều khiển u .



Hình 1.18 : Hệ thống nâng vật trong từ trường

Phương trình toán mô tả hệ thống

$$\begin{cases} \frac{dh}{dt} = v \\ u = Ri + \frac{d(L(h)i)}{dt} \\ m \frac{dv}{dt} = mg - C \left(\frac{i}{h} \right)^2 \end{cases} \quad (1.57)$$

Trong đó:

- h : vị trí hòn bi (m)
- v : vận tốc hòn bi (m/s)
- i : dòng điện qua cuộn dây (A)
- u : điện áp cung cấp cho cuộn dây (V)
- R, L : điện trở và điện cảm cuộn dây (Ω , H)
- C : hằng số lực từ (Nm^2/A^2)
- m : khối lượng hòn bi (Kg)
- g : gia tốc trọng trường. (m/s^2)

Điện cảm của cuộn dây là một hàm phi tuyến phụ thuộc vào vị trí của hòn bi

$$L(h) = L_1 + \frac{2C}{h} \quad (1.58)$$

L_1 là điện cảm của cuộn dây khi hòn bi ở rất xa.

Chọn biến trạng thái như sau:

$$x_1 = h, x_2 = v, x_3 = i \quad (1.59)$$

Véc tơ trạng thái của hệ thống $X = (x_1, x_2, x_3)^T$

Từ (1.57), (1.58) và (1.59) ta được phương trình trạng thái:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = g - \frac{C}{m} \left(\frac{x_3}{x_1} \right)^2 \\ \dot{x}_3 = \frac{R}{L} x_3 + \frac{2C}{L} \left(\frac{x_2 x_3}{x_1^2} \right) + \frac{1}{L} u \end{cases} \quad (1.60)$$

Điểm cân bằng của hệ thống là nghiệm của hệ $(\dot{x}_1 = 0, \dot{x}_2 = 0, \dot{x}_3 = 0)$

Giải ra được $X_b = [x_{1b}, 0, x_{3b}]^T$, với $x_{3b} = x_{1b} \sqrt{\frac{gm}{C}}$

Gọi $X_d = [x_{1d}, x_{2d}, x_{3d}]^T$ là véc tơ trạng thái mong muốn.

Mục tiêu của hệ thống là đưa X tiến về X_d với sai số nhỏ nhất.

Thiết kế BDK trượt

Thực hiện phép đổi trục như sau:

$$\begin{cases} z_1 = x_1 - x_{1d} \\ z_2 = x_2 \\ z_3 = g - \frac{C}{m} \left(\frac{x_3}{x_1} \right)^2 \end{cases} \quad (1.61)$$

Lúc này ta cần xác định luật điều khiển u sao cho $Z = (z_1, z_2, z_3)^T$ tiến về $(0,0,0)^T$ khi $t \rightarrow \infty$, khi ấy $X \rightarrow X_d$.

Kết hợp (1.60), (1.61) và một số phép biến đổi ta được:

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = 2(g - z_3) \left[\frac{z_2}{z_1 + x_{1d}} \left(1 - \frac{2C}{L(z_1 + x_{1d})} \right) + \frac{R}{L} \right] - \frac{2}{L(z_1 + x_{1d})} \sqrt{\frac{C}{m}} (g - z_3) u \end{cases} \quad (1.62)$$

Đặt

$$\begin{cases} g(z) = -\frac{2}{L(z_1 + x_{1d})} \sqrt{\frac{C}{m}} (g - z_3)u \\ f(z) = 2(g - z_3) \left[\frac{z_2}{z_1 + x_{1d}} \left(1 - \frac{2C}{L(z_1 + x_{1d})} \right) + \frac{R}{L} \right] \end{cases} \quad (1.63)$$

Từ (1.62) và (1.63) ta được mô hình động học của hệ thống trong hệ tọa độ mới như sau:

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = f(z) + g(z)u \end{cases} \quad (1.64)$$

Ngõ ra của hệ thống trong hệ tọa độ mới là:

$$e = z_1 = x_1 - x_{1d} \quad (1.65)$$

Mối quan hệ ngõ vào và ngõ ra:

$$e^{(3)} = f(z) + g(z)u \quad (1.66)$$

Hai hàm $f(z)$, $g(z)$ tương ứng trong hệ tọa độ ban đầu là $f_1(x)$, $g_1(x)$:

$$\begin{cases} f_1(x) = \frac{2C}{m} \left(\frac{x_2 x_3^2}{x_1^3} \left(1 - \frac{2C}{Lx_1} \right) + \frac{R}{L} \left(\frac{x_3}{x_1} \right)^2 \right) \\ g_1(x) = -\frac{2Cx_3}{Lmx_1^2} \end{cases} \quad (1.67)$$

Ta viết lại (1.66) trong hệ tọa độ ban đầu:

$$e^{(3)} = f_1(x) + g_1(x)u \quad (1.68)$$

Chọn mặt trượt như sau:

$$S = \ddot{e} + a_1 \dot{e} + a_0 e \quad (1.69)$$

Với a_1 , a_0 được chọn sao cho đa thức đặt trung của phương trình $S = 0$ là Hurwitz.

Từ (1.69) và (1.64) ta được:

$$S = z_3 + a_1 z_2 + a_0 z_1 \quad (1.70)$$

Lấy đạo hàm của S theo thời gian ta được:

$$\dot{S} = \dot{z}_3 + a_1 \dot{z}_2 + a_0 \dot{z}_1 = f(z) + g(z)u + a_1 z_3 + a_0 z_2 \quad (1.71)$$

Chọn luật điều khiển u như sau:

$$u = \frac{1}{g(z)} [-f(z) - a_1 z_3 - a_0 z_2 - W \text{sign}(z_3 + a_1 z_2 + a_0 z_1)] \quad (1.72)$$

Thay (1.72) vào (1.71) ta được:

$$\dot{S} = -W \text{sign}(z_3 + a_1 z_2 + a_0 z_1) = -W \text{sign}(S) \quad (1.73)$$

Nếu chọn W là hằng số dương thì ta sẽ được $\dot{S} < 0$. Do vậy biến trạng thái Z sẽ hội tụ về $zero$ khi $t \rightarrow \infty$ thỏa yêu cầu đề ra.

Ta có thể viết lại mặt trượt S dưới dạng hàm của x_1, x_2, x_3 như sau:

$$S = g - \frac{C}{m} \left(\frac{x_3}{x_1} \right)^2 + a_1 x_2 + a_0 (x_1 - x_{1d}) \quad (1.74)$$

Và luật điều khiển u là:

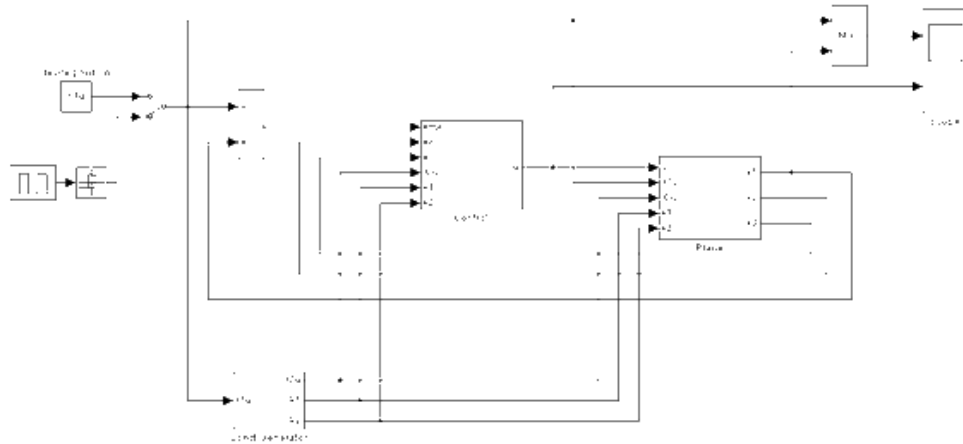
$$u = \frac{1}{g_1} \left[-f_1 - a_1 \left(g - \frac{C}{m} \left(\frac{x_3}{x_1} \right)^2 \right) - a_0 x_2 - W \text{sign} \left(g - \frac{C}{m} \left(\frac{x_3}{x_1} \right)^2 + a_1 x_2 + a_0 (x_1 - x_{1d}) \right) \right] \quad (1.75)$$

Mô phỏng Matlab:

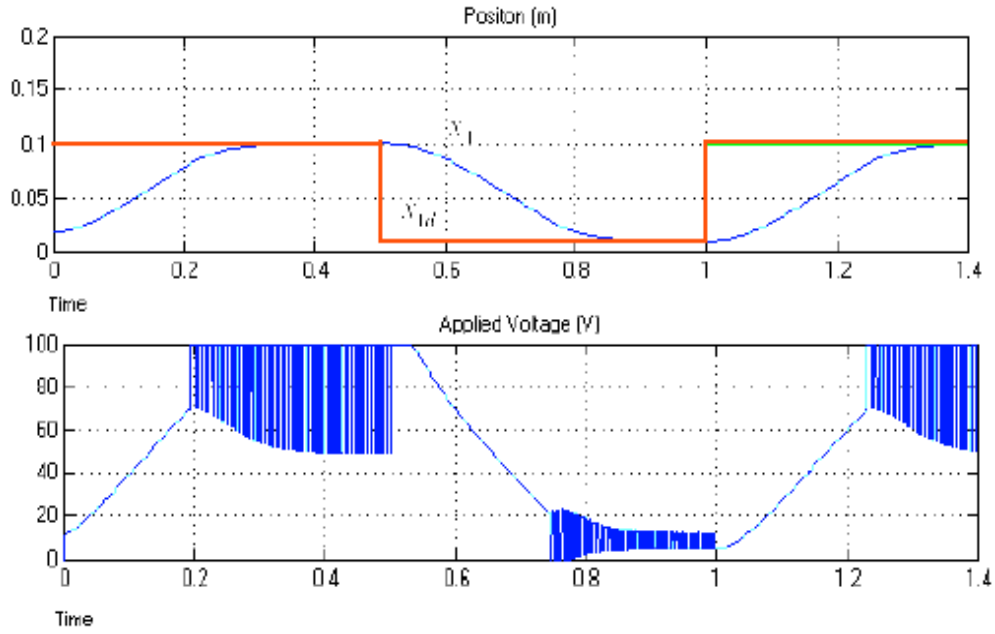
Các thông số mô phỏng của hệ thống

Khối lượng hòn bi $m = 11.87g$, bán kính $R = 7.14mm$, một nam châm điện, điện trở cuộn dây $R = 28.7\Omega$, điện kháng $L_1 = 0.65H$, hằng số lực từ $C = 1.4 \times 10^{-4} Nm^2 A^2$.

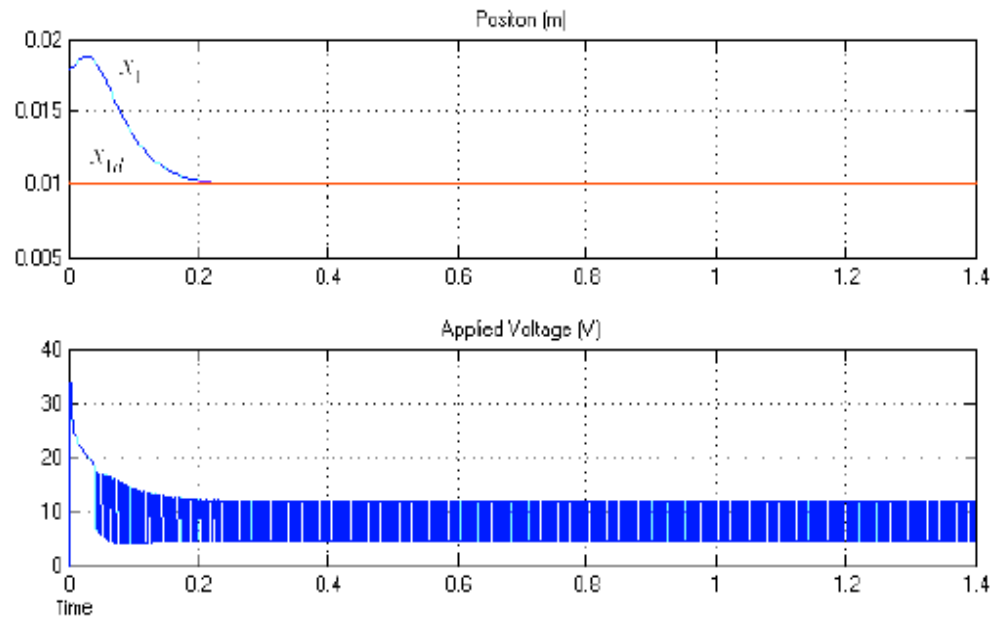
Sơ đồ Simulink



Kết quả mô phỏng:



Hình 1.19: Vị trí và áp điều khiển khi tín hiệu đặt biến thiên



Hình 1.20: Vị trí và áp điều khiển khi tín hiệu đặt là hằng số

Thiết kế BDK trượt mờ cho hệ thống nâng vật trong từ trường

Trong phần thiết kế BDK trượt ta đã biết luật điều khiển u như sau:

$$u = \frac{1}{g_1} \left[-f_1 - a_1 \left(g - \frac{C}{m} \left(\frac{x_3}{x_1} \right)^2 \right) - a_0 x_2 - W \text{sign}(S) \right]$$

với S được xác định từ (1.74), f_1 và g_1 được xác định từ (1.67).

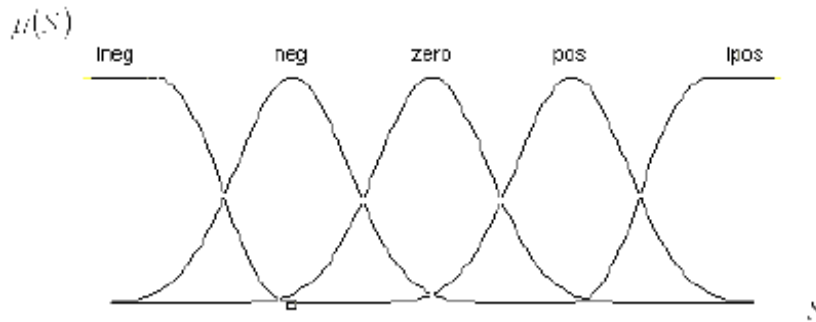
Do trong luật điều khiển có hàm sign nên gây ra hiện tượng dao động, để khắc phục nhược điểm này ta thêm khâu xử lý mờ trong bộ điều khiển để thay thế cho hàm sign .

Chọn luật điều khiển $u = u_{eq} + u_s$, với:

$$u_{eq} = \frac{1}{g_1} \left[-f_1 - a_1 \left(g - \frac{C}{m} \left(\frac{x_3}{x_1} \right)^2 \right) - a_0 x_2 \right] \tag{1.76}$$

Các bước xây dựng bộ mờ:

Bước 1: Mờ hoá mặt trượt S



Hình 1.21: Hàm thuộc với 5 tập mờ

Bước 2: Xây dựng hệ qui tắc mờ:

- R^1 : If S is zero Then $u^1 = u_{eq}$
 - R^2 : If S is pos Then $u^2 = u_{eq} + C_0$
 - R^3 : If S is lpos Then $u^3 = u_{eq} + C_1$
 - R^4 : If S is neg Then $u^4 = u_{eq} - C_0$
 - R^5 : If S is lneg Then $u^5 = u_{eq} - C_1$
- C_0, C_1 là các hằng số dương $C_0 > C_1$

Bước 3: Giải mờ

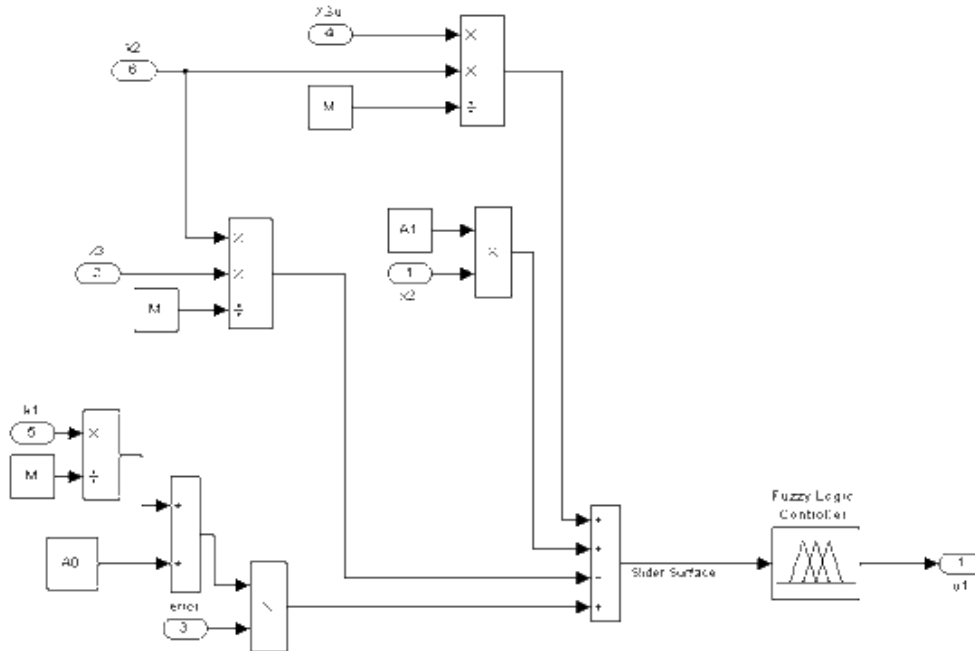
Bằng phương pháp giải mờ trọng tâm, luật điều khiển u được xác định:

$$u = \frac{\sum_{i=1}^5 \beta_i u^i}{\sum_{i=1}^5 \beta_i} \quad (1.77)$$

Trong đó β_i là độ đúng của qui tắc thứ i :

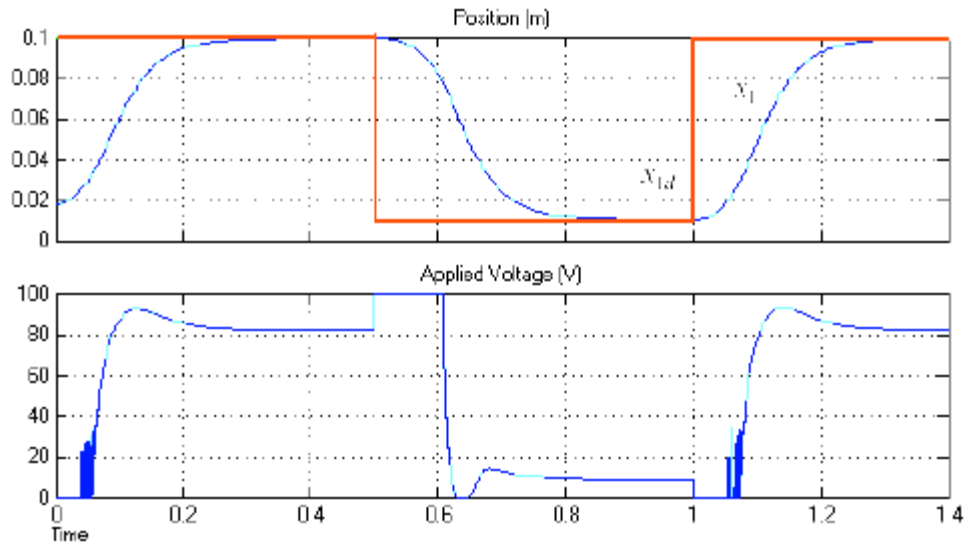
$$\begin{aligned} \beta_1 &= \mu_{zero}(S) \\ \beta_2 &= \mu_{pos}(S) \\ \beta_3 &= \mu_{lpos}(S) \\ \beta_4 &= \mu_{neg}(S) \\ \beta_5 &= \mu_{lneg}(S) \end{aligned} \quad (1.78)$$

Sơ đồ Simulink

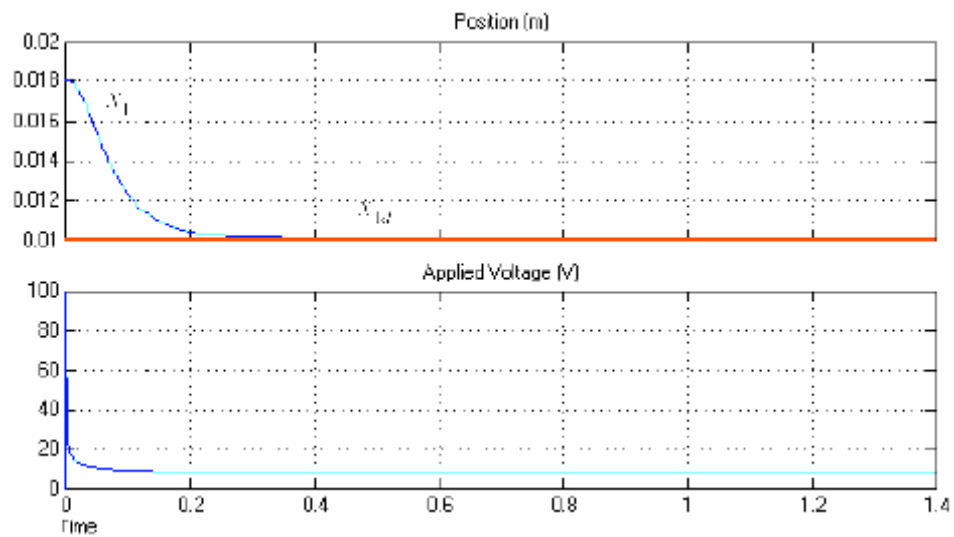


Kết quả mô phỏng

- Sử dụng 3 tập mờ, chọn $C_0 = 350$.

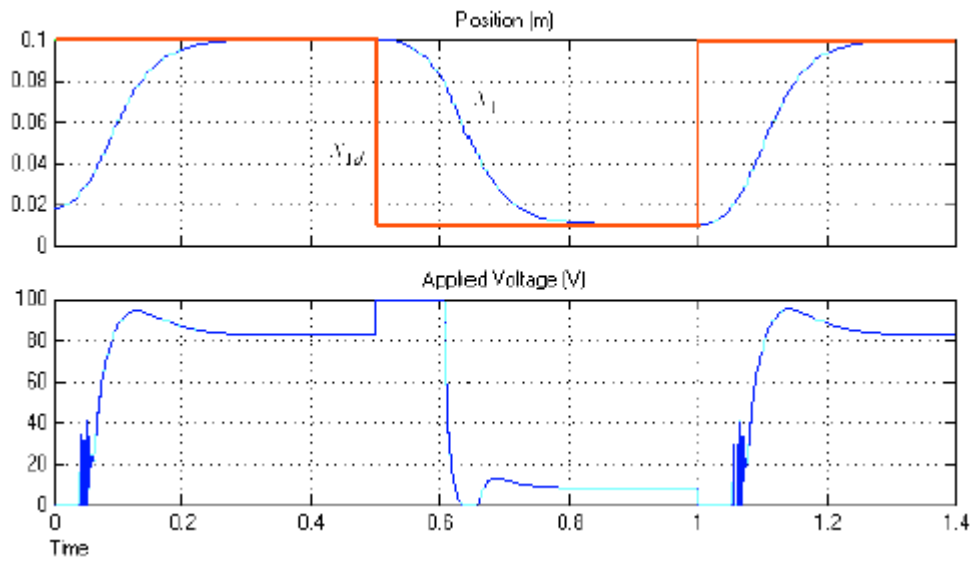


Hình 1.22: Vị trí và áp ĐK khi tín hiệu đặt là xung vuông

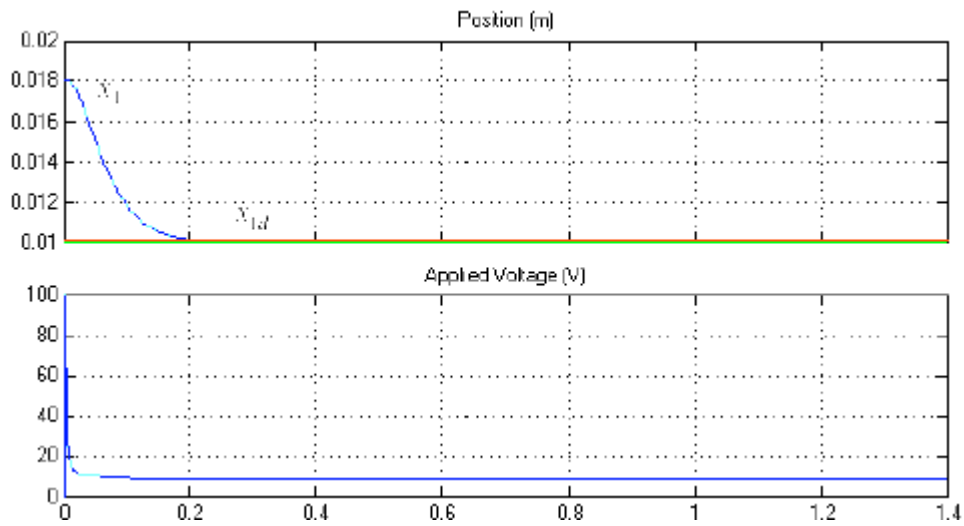


Hình 1.23 Vị trí và áp ĐK khi tín hiệu đặt là hằng số

- Sử dụng 5 tập mờ, chọn $C_0 = 100$ và $C_1 = 350$.

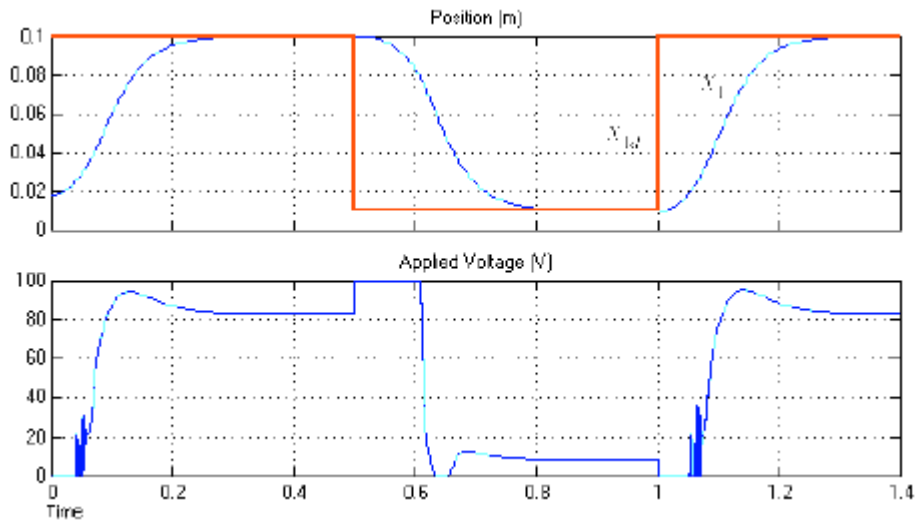


Hình 1.24 Vị trí và áp ĐK khi tín hiệu đặt là xung vuông

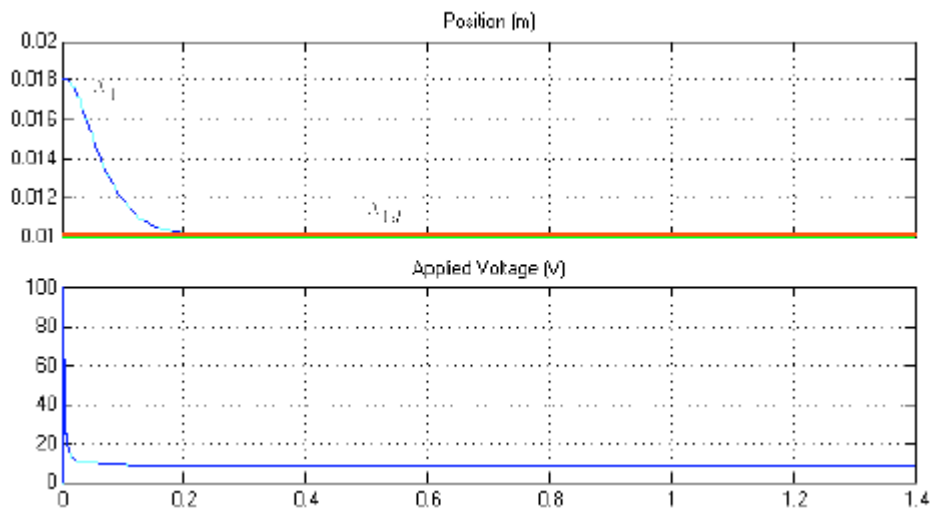


Hình 1.25: Vị trí và áp ĐK khi tín hiệu đặt là hằng số

- Sử dụng 7 tập mờ, chọn $C_0 = 100$, $C_1 = 200$ và $C_2 = 350$.



Hình 1.26: Vị trí và áp ĐK khi tín hiệu đặt là xung vuông



Hình 1.27: Vị trí và áp ĐK khi tín hiệu đặt là hằng số

Kết luận

- Việc thêm BDK mờ đã triệt tiêu hiện tượng dao động.
- Đáp ứng hệ thống tốt hơn.
- Chọn 5 tập mờ là thích hợp nhất khi xây dựng BDK mờ.

1.3. Mạng nơ-ron nhân tạo

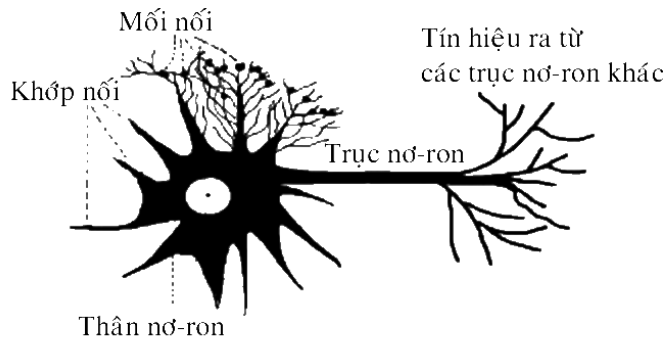
Những năm 40 của thế kỷ 20, với mong muốn chế tạo những cỗ máy thông minh có chức năng như bộ não con người, các nhà nghiên cứu đã phát triển những mô hình phân cứng (sau đó là phần mềm) đơn giản về các nơ-ron sinh học và cơ chế tương tác của chúng. McCulloch và Pitts là những người đầu tiên đã xuất bản một công trình nghiên cứu có tính hệ thống về mạng nơ-ron nhân tạo.

Đến những năm 50 và 60, một nhóm các nhà nghiên cứu đã kết hợp những kiến thức về sinh học và tâm thần học để tạo ra một mạng nơ-ron nhân tạo hoàn chỉnh đầu tiên. Ban đầu được chế tạo dưới dạng một mạch điện tử, sau đó các mạng nơ-ron được chuyển thành các chương trình mô phỏng trên máy tính có mức độ linh hoạt cao hơn.

Trong thời gian gần đây mạng nơ-ron phát triển mạnh mẽ với các nghiên cứu của Rosenblatt, Widrow và Hoff về các mạng nơ-ron nhiều lớp phức tạp. Mặt khác, do khả năng của phần cứng máy tính tăng lên rất nhiều cũng góp phần vào sự phát triển của lĩnh vực nghiên cứu này. Mạng nơ-ron nhân tạo được ứng dụng trong nhiều lĩnh vực như: nhận dạng, phân loại, xử lý ảnh, các hệ thống điều khiển, dự báo....

1.3.1. Cấu trúc mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN):

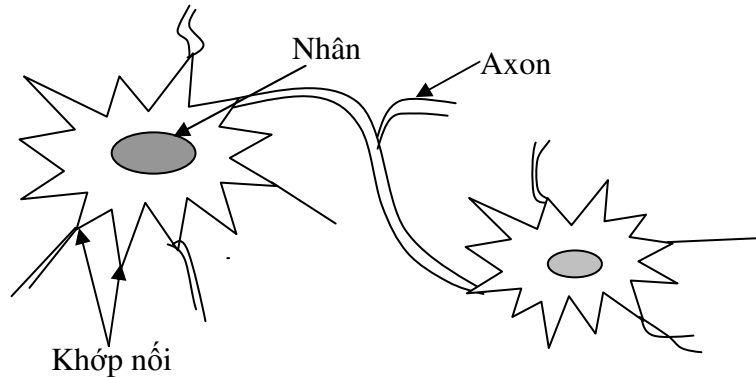
Bộ não con người có khoảng $10^{11} \div 10^{12}$ nơ-ron. Mỗi nơ-ron có thể liên kết với 10^4 nơ-ron khác thông qua các khớp nối (dendrite). Các nơ-ron nhận tín hiệu điện từ các khớp nối và khi sự tổng hợp của các tín hiệu này vượt quá một ngưỡng cho phép thì nơ-ron sẽ kích hoạt một tín hiệu điện ở ngõ ra để truyền tới trục nơ-ron (axon) và dẫn đến các nơ-ron khác.



Cấu trúc một nơ-ron sinh học

Mạng nơ-ron nhân tạo cũng hoạt động dựa theo cách thức của bộ não con người nhưng ở cấp độ đơn giản hơn.

Mạng nơron là sự tái tạo bằng kỹ thuật những chức năng của hệ thần kinh con người. Mạng nơron gồm vô số các nơron liên kết với nhau như hình sau

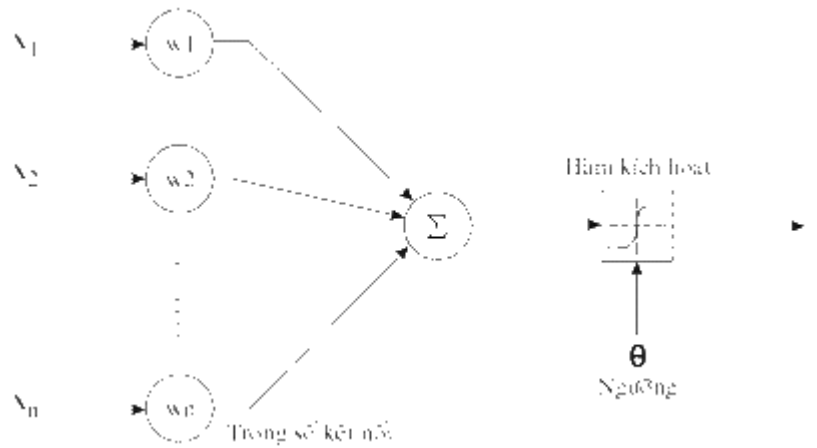


Hai đặc tính cơ bản của mạng nơron là:

- + Quá trình tính toán được tiến hành song song và phân tán trên nhiều nơron gần như đồng thời.
- + Tính toán thực chất là quá trình học, chứ không phải theo sơ đồ định sẵn từ trước.

Mô hình toán của mạng nơron nhân tạo :

Một mạng nơron nhân tạo gồm 3 thuộc tính: trọng số kết nối, ngưỡng và hàm kích hoạt.



Ta đặt: $X = [x_1, x_2, \dots, x_n]^T$ là cường độ của vector ngõ vào.

Ngõ ra của mạng được tính theo công thức:

$$y = f\left(\sum_{j=1}^n w_j x_j - \theta\right) = f(W^T X - \theta)$$

Trọng số kết nối:

$$W = [w_1, w_2, \dots, w_n]^T$$

Ngưỡng (ngưỡng phân cực – bias): Kí hiệu: θ

Ngưỡng là giá trị biên độ độ lệch, nó ảnh hưởng đến sự kích hoạt ngõ ra của nơ-ron theo công thức:

$$u = X^T W - \theta$$

Hàm kích hoạt

Quá trình xử lý thông tin gồm: xử lý tín hiệu đầu vào và tín hiệu đầu ra

- **Hàm tổng (u):** dùng để kết hợp và xử lý các thông tin ở đầu vào

Một số dạng hàm tổng thường gặp:

Hàm tuyến tính (linear function):

$$u = \left(\sum_{j=1}^n w_j x_j \right) - \theta = W^T X - \theta$$

Hàm toàn phương (quadratic function):

$$u = \left(\sum_{j=1}^n w_j x_j^2 \right) - \theta$$

Hàm cầu (spherical function):

$$u = \left(\frac{1}{\rho^2} \sum_{j=1}^n (x_j - w_j)^2 \right) - \theta = \frac{1}{\rho^2} (X - W)^T (X - W) - \theta$$

Thông thường, hàm tổng được sử dụng nhiều nhất là hàm tuyến tính.

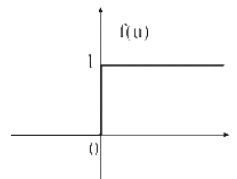
- **Hàm kích hoạt** (activation function): là hàm xử lý tín hiệu ngõ ra.

Kí hiệu: $f(\cdot)$

Một số kiểu hàm kích hoạt thông dụng :

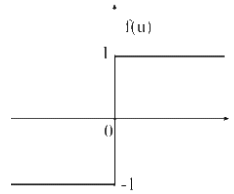
Hàm nấc:

$$y = f(u) = \begin{cases} 0, & u < 0 \\ 1, & u \geq 0 \end{cases}$$



Hàm dấu:

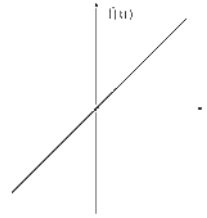
$$y = f(u) = \begin{cases} -1, & u < 0 \\ 1, & u \geq 0 \end{cases}$$



Hàm tuyến tính (linear function):

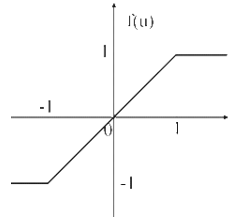
$$y = f(u) = \alpha \cdot u$$

Với α là hệ số góc của hàm.



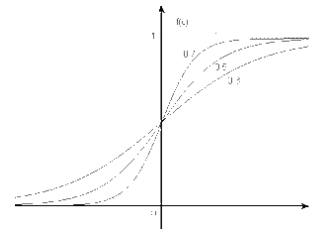
Hàm tuyến tính bão hòa:

$$y = f(u) = \begin{cases} -1, & u < -1 \\ u, & -1 \leq u < 1 \\ 1, & u \geq 1 \end{cases}$$



Hàm dạng S (Sigmoid) đơn cực:

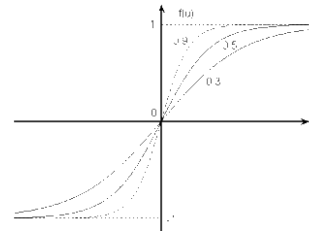
$$y = f(u) = \frac{1}{1 + e^{-\lambda u}}$$



Với: λ là hệ số dạng của hàm sigmoid

Hàm dạng S lưỡng cực:

$$y = f(u) = \frac{2}{1 + e^{-\lambda u}} - 1$$

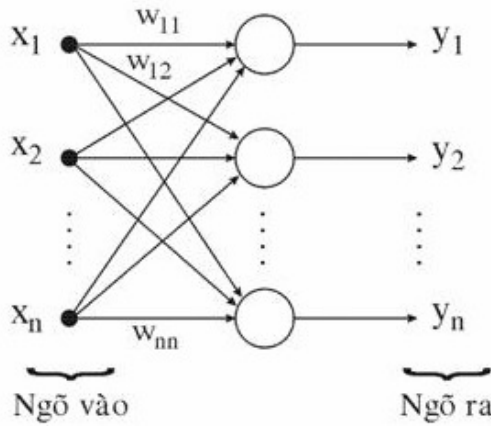


1.3.2. Phân loại mạng nơ-ron nhân tạo:

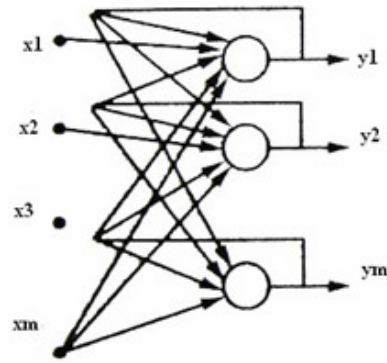
Các loại mạng có thể được phân loại dựa trên các tính chất của nó:

- Theo số lớp:
 - Mạng một lớp (mạng đơn nơ-ron): mạng chỉ gồm 1 nơ-ron
 - Mạng nhiều lớp (mạng đa nơ-ron): mạng gồm nhiều nơ-ron

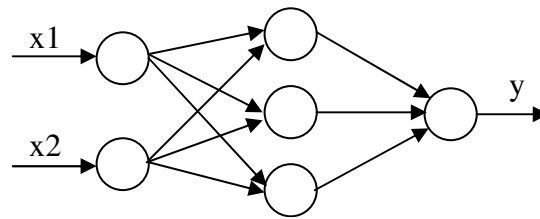
- Mạng truyền thẳng (mạng nuôi tiến): các tín hiệu trong mạng chỉ truyền theo một chiều từ ngõ vào đến ngõ ra.
- Mạng hồi quy (mạng nuôi lùi): các tín hiệu hồi tiếp từ ngõ ra trở về ngõ vào.



Mạng truyền thẳng 1 lớp

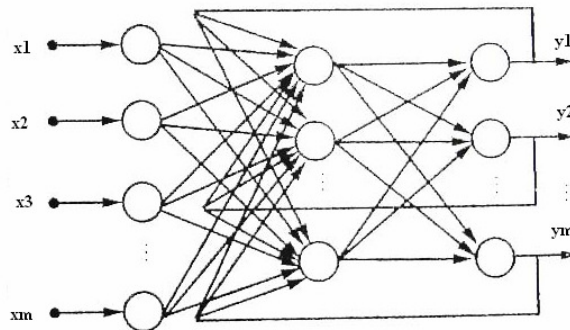


Mạng hồi quy 1 lớp



Lớp vào Lớp bị che Lớp ra

Mạng truyền thẳng nhiều lớp



Lớp vào Lớp bị che Lớp ra

Mạng hồi quy nhiều lớp

1.3.3. Các phương pháp huấn luyện mạng nơ-ron:

Có hai kiểu học:

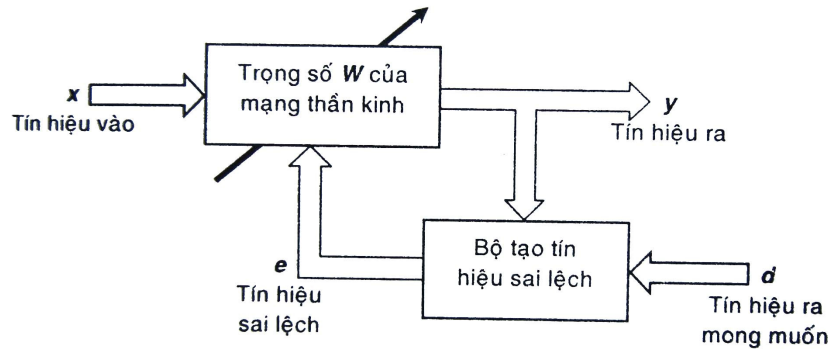
- Học thông số (Parameter Learning): dùng để cập nhật các trọng số liên kết giữa các tế bào nơ-ron và ngưỡng phân cực trong mạng.
- Học cấu trúc (Structure Learning): dùng để thay đổi cấu trúc mạng bao gồm cả tế bào nơ-ron và cách liên kết giữa chúng.

Ta có thể sử dụng riêng rẽ hay đồng thời cả hai kiểu học trên. Tuy nhiên, ở đây ta chỉ đề cập đến kiểu học thông số. Có 3 luật học thông số:

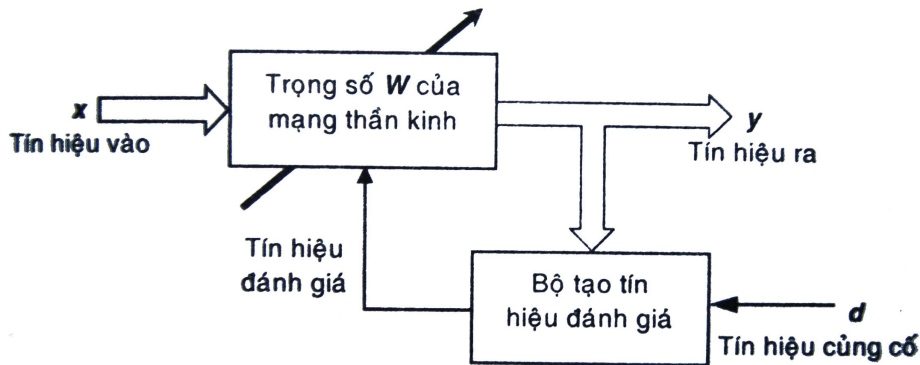
- Luật học có giám sát:

Cho tập các dữ liệu vào – ra: $\{x_1, d_1\}, \{x_2, d_2\}, \dots, \{x_Q, d_Q\}$

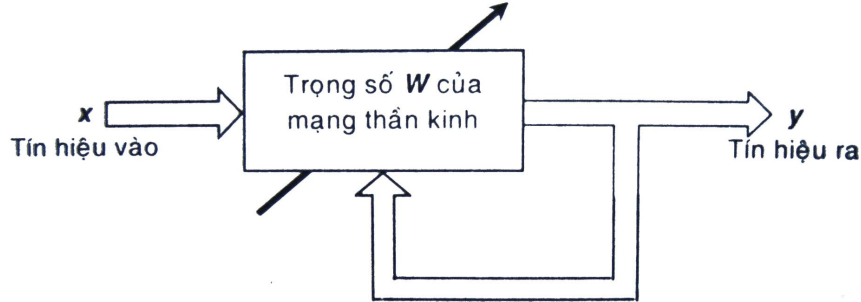
Ban đầu các trọng số của mạng được chọn bất kỳ. Khi có ngõ vào mạng x_q , ngõ ra của mạng y_q được so sánh với ngõ ra mục tiêu d_q . Luật học dựa vào sai số $e_q = d_q - y_q$ hiệu chỉnh trọng số và ngưỡng phân cực của mạng để đưa ngõ ra về gần với mục tiêu.



- Luật học củng cố: không như học có giám sát được cung cấp mục tiêu trước (mỗi tín hiệu ngõ vào sẽ có một tín hiệu ra tương ứng), luật học củng cố chỉ được cung cấp theo mức (Ví dụ: “đúng” hoặc “sai”). Bởi vì trong thực tế không phải lúc nào ta cũng có đầy đủ thông tin về đối tượng. Vì vậy, kiểu học củng cố sẽ thích hợp nhất cho những ứng dụng điều khiển hệ thống.



▪ *Luật học không có giám sát*: trọng số và ngưỡng phân cực chỉ được hiệu chỉnh đối với ngõ vào. Không có mục tiêu ngõ ra được đặt trước. Có vẻ như điều này thiếu thực tế, vì làm sao huấn luyện được mạng nếu không biết nó sẽ làm gì? Hầu hết những thuật toán này sẽ tự phát hiện các đặc điểm, các mối tương quan giữa các mẫu dữ liệu vào và mã hóa thành dữ liệu ra. Điều này rất hữu ích trong những ứng dụng như lượng tử hoá vector.



1.3.4. Mạng truyền thẳng một lớp (mạng perceptron đơn giản):

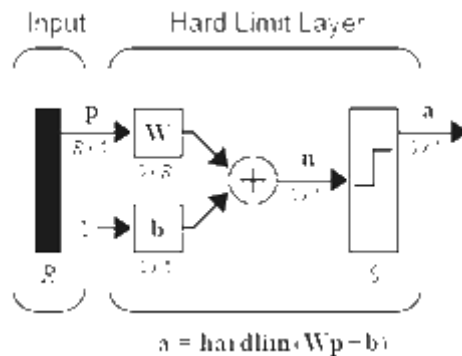
Thuật toán học perceptron:

Vào cuối những năm 1950, Frank Rosenblatt và một số nhà nghiên cứu khác đã phát triển một loại mạng neuron gọi là perceptron. Rosenblatt đã giới thiệu một luật học để huấn luyện mạng perceptron giải quyết một số vấn đề nhận dạng kiểu mẫu. Mạng perceptron sẽ học từ những đáp ứng mà ta đưa vào. Nó cũng có thể học khi được cho những giá trị trọng số và phân cực ngẫu nhiên.

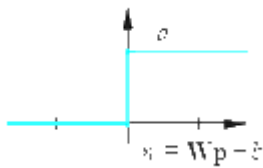
Ngày nay, mạng perceptron vẫn được xem là một mạng quan trọng vì tính nhanh và chính xác đối với những vấn đề mà nó có thể giải quyết.

1. Cấu trúc mạng perceptron

Mạng perceptron tổng quát có cấu trúc như hình :



Perceptron có hàm kích hoạt là hàm *hardlim* (hàm giới hạn cứng):

$$a = \text{hardlim}(n) = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$


Ma trận trọng số:

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \dots & w_{m,n} \end{bmatrix} = \begin{bmatrix} \mathbf{w}^T \\ \mathbf{w}^T \\ \vdots \\ \mathbf{w}^T \end{bmatrix} \quad \text{với} \quad \mathbf{w} = \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix}$$

Ngõ ra của mạng:

$$a = \text{hardlim}(Wp - b)$$

$$a_j = \text{hardlim}(n_j) = \text{hardlim}(\mathbf{w}_j^T \mathbf{p} - b_j)$$

Nhận xét:

Mỗi nơ-ron trong mạng sẽ chia không gian ngõ vào thành 2 miền: một miền bao gồm các ngõ vào sao cho ngõ ra bằng 1 và một miền bao gồm các ngõ vào sao cho ngõ ra bằng 0. Vì vậy, ta cần khảo sát đường phân chia giữa những miền này.

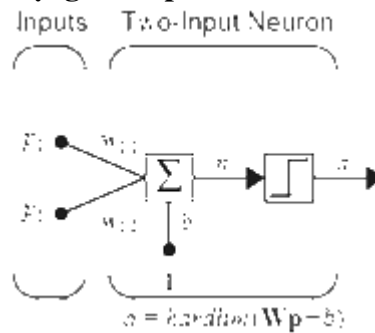
2. Cách xác định đường phân chia ranh giới:

Có 2 cách xác định :

- **Cách 1:** Dựa vào phương trình đường phân chia: $\mathbf{w}^T \mathbf{p} + b_i = 0$

Ví dụ 1.9:

Khảo sát mạng **Mạng Perceptron đơn neutron** như hình



Ngõ ra của mạng là:

$$a = \text{hardlim}(n) = \text{hardlim}(Wp - b)$$

$$= \text{hardlim}(\mathbf{w}^T \mathbf{p} - b) = \text{hardlim}(w_{1,1}p_1 - w_{1,2}p_2 - b)$$

Phương trình đường phân chia :

$$a = \mathbf{w}^T \mathbf{p} - b = w_{11} p_1 + w_{12} p_2 - b = 0.$$

Ta chọn trọng số và ngưỡng phân cực:

$$w_{11} = 1, w_{12} = 1, b = -1$$

Đường phân chia lúc này là:

$$a = \mathbf{w}^T \mathbf{p} - b = w_{11} p_1 + w_{12} p_2 - b = p_1 + p_2 - 1 = 0$$

Để tìm đường thẳng này, có thể tìm giao điểm của nó với trục p_1 và p_2 .

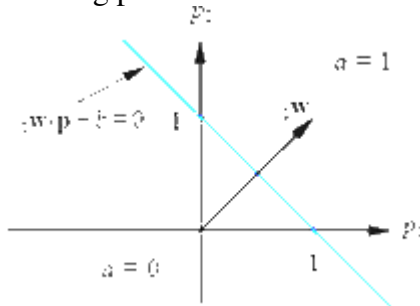
Tìm giao điểm với trục p_2 , đặt $p_1=0$:

$$p_2 = -\frac{b}{w_{12}} = -\frac{-1}{1} = 1 \quad \text{if } p_1 = 0$$

Tìm giao điểm với trục p_1 , đặt $p_2=0$:

$$p_1 = -\frac{b}{w_{11}} = -\frac{-1}{1} = 1 \quad \text{if } p_2 = 0$$

Khi đó, ta tìm được đường phân chia như hình sau:

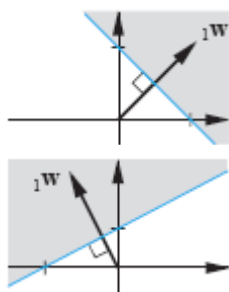


Miền tương ứng với ngõ ra bằng 1 được xác định bằng cách kiểm tra thử một điểm. Với $\mathbf{p}=[2 \ 0]^T$, ngõ ra sẽ là:

$$a = \text{hardlim}(\mathbf{w}^T \mathbf{p} - b) = \text{hardlim}\left(\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} - 1\right) = 1$$

Vậy ngõ ra của mạng sẽ bằng 1 ở miền bên trên và bên phải của đường phân chia. Miền này được tô đậm ở hình trên

➤ **Cách 2:** Dựa vào đồ thị:



Ta chú ý rằng đường phân chia luôn trực giao với \mathbf{w} như hình bên. Đường phân chia được định nghĩa là:

$$\mathbf{w}^T \mathbf{p} - b = 0$$

Đối với những điểm trên đường phân chia, tích $\mathbf{w}^T \mathbf{p}$ luôn không đổi. Có nghĩa là những vector ngõ vào \mathbf{p} này phải có cùng hình chiếu lên vector trọng số. Vậy chúng phải nằm trên cùng một đường thẳng trực giao với

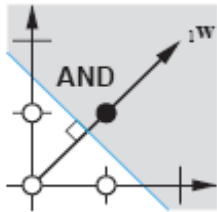
vector trọng số. Hơn nữa, những vector ở miền tô đậm của hình trên sẽ có tích $\mathbf{w}^T \mathbf{p}$ lớn hơn $-b$, còn ở miền còn lại thì tích $\mathbf{w}^T \mathbf{p}$ sẽ nhỏ hơn. Vì vậy vector trọng số \mathbf{w} sẽ luôn hướng về miền có ngõ ra bằng 1.

Sau khi đã chọn vector trọng số đúng hướng, ngưỡng phân cực được tính bằng cách chọn một điểm trên đường biên và cho thoả phương trình đường phân chia.

Ví dụ 1.10:

Thiết kế mạng perceptron thực hiện hàm chức năng cổng AND. Các cặp ngõ vào và mục tiêu của cổng AND là:

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, t_1 = 0 \right\} \left\{ \mathbf{p}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, t_2 = 0 \right\} \left\{ \mathbf{p}_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, t_3 = 0 \right\} \left\{ \mathbf{p}_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, t_4 = 1 \right\}.$$



Ta chọn một đường thẳng phân vùng trắng và vùng đen ra 2 miền như hình bên.

Tiếp theo ta chọn vector trọng số trực giao với đường phân chia:

$$\mathbf{w} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Cuối cùng, chọn một điểm trên đường phân chia $\mathbf{p} = [1.5 \ 0]^T$ và tính ngưỡng phân cực b bằng cách thế vào phương trình:

$$\mathbf{w}^T \mathbf{p} - b = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 0 \end{bmatrix} - b = 1.5 - b = 0 \quad \Rightarrow \quad b = 1.5$$

Có thể kiểm tra lại đáp ứng của mạng với một cặp ngõ vào/mục tiêu. Nếu thử với \mathbf{p}_2 , ngõ ra sẽ là:

$$a = \text{hardlim}(\mathbf{w}^T \mathbf{p}_2 - b) = \text{hardlim} \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} - 1.5 \right)$$

$$a = \text{hardlim}(-1) = 0.$$

Ta thấy ngõ ra trùng với mục tiêu t_2 . Kiểm tra cho tất cả các ngõ vào còn lại, ta thấy chúng đều phân loại đúng.

Chú ý : đối với mạng perceptron đa nơ-ron, mỗi nơ-ron có một đường phân chia. Đường phân chia cho nơ-ron thứ i là:

$$\mathbf{w}_i^T \mathbf{p} - b_i = 0.$$

Một perceptron đơn neuron có thể phân vector ngõ vào thành 2 loại vì ngõ ra của nó có thể là 0 hoặc 1. Mạng perceptron đa neuron có thể phân ngõ vào thành nhiều loại, mỗi loại đại diện bởi một ngõ ra khác nhau. Vì mỗi

phần tử của vector ngõ ra có thể bằng 0 hoặc 1, nên có tổng cộng 2^S loại ngõ vào với S là số neutron.

3. Luật học perceptron

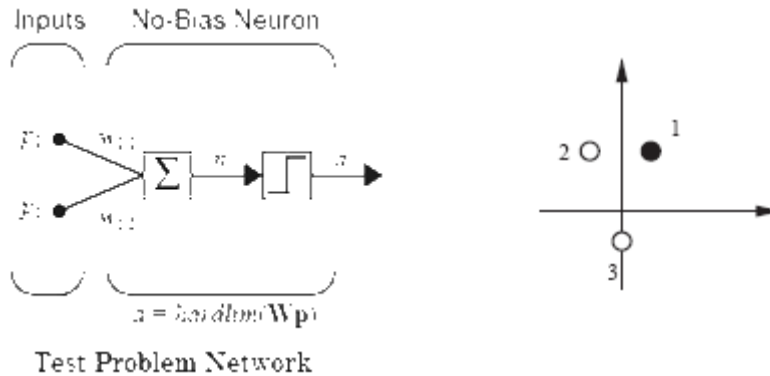
Luật học perceptron thuộc loại luật học có giám sát.

Bài toán thử nghiệm:

Cho các cặp ngõ vào và mục tiêu của bài toán là:

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, t_1 = 1 \right\} \left\{ \mathbf{p}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, t_2 = 0 \right\} \left\{ \mathbf{p}_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, t_3 = 0 \right\}.$$

Mạng perceptron dùng để giải bài toán này cần có 2 ngõ vào và 1 ngõ ra.



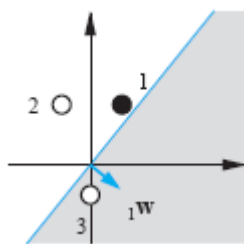
Xây dựng luật học

Trước hết ta gán các giá trị khởi đầu cho thông số của mạng.

Chọn $b = 0$ và ${}_1w$ bất kỳ:

$${}_1w^T = [1 \ 0 \ -0.5]$$

Đưa vector ngõ vào vào mạng perceptron, bắt đầu với \mathbf{p}_1 :

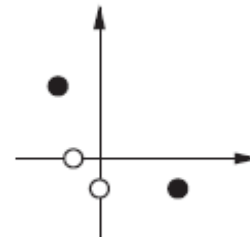


$$o = \text{hardlim}({}_1w^T \mathbf{p}_1) = \text{hardlim} \left([1 \ 0 \ -0.5] \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \right)$$

$$o = \text{hardlim}(-0.6) = 0.$$

Ngõ ra chưa đúng với giá trị mong muốn ($t_1 = 1$), nghĩa là đường phân chia không chính xác. Ta cần thay vector trọng số để nó hướng về \mathbf{p}_1 nhiều hơn.

Ta có thể đặt ${}_1w = \mathbf{p}_1$. Cách này đơn giản và đảm bảo \mathbf{p}_1 sẽ được phân loại đúng. Tuy nhiên nó không thể giải quyết những bài toán có dạng như hình bên. Bởi vì mỗi lần vector này không được phân loại đúng thì trọng số sẽ hoán đổi qua lại và không thể giải bài toán.



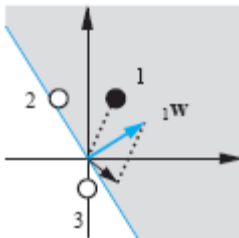
Một cách khác là cộng p_1 vào ${}_1w$. Việc cộng thêm sẽ làm cho ${}_1w$ hướng về p_1 nhiều hơn. Luật này có thể được viết như sau:

$$\text{If } \tau = 1 \text{ and } \sigma = 0 \text{ then } {}_1w^{new} = {}_1w^{old} + p.$$

Áp dụng tính giá trị mới cho ${}_1w$:

$${}_1w^{new} = {}_1w^{old} + p = \begin{bmatrix} 1 & 0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}$$

Ta thấy p_1 đã được phân loại đúng. Tương tự cho các ngõ vào còn lại



Với ngõ vào p_2 , ngõ ra tương ứng là:

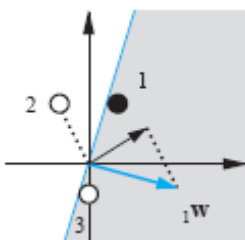
$$\begin{aligned} \sigma &= \text{hardlim}({}_1w^T p_2) = \text{hardlim} \left(\begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right) \\ &= \text{hardlim}(0.4) = 1 \end{aligned}$$

Ngõ ra chưa đúng với giá trị mong muốn ($t_2 = 0$). Ta cần đưa ${}_1w$ ra xa p_1 hơn, bằng cách đổi luật:

$$\text{If } \tau = 0 \text{ and } \sigma = 1, \text{ then } {}_1w^{new} = {}_1w^{old} - p$$

Áp dụng vào bài toán ta được:

$${}_1w^{new} = {}_1w^{old} - p_1 = \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ -0.8 \end{bmatrix}$$

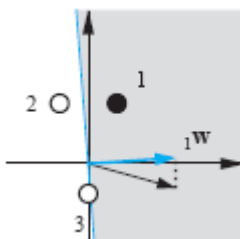


Kiểm tra tiếp vector p_3 :

$$\begin{aligned} \sigma &= \text{hardlim}({}_1w^T p_3) = \text{hardlim} \left(\begin{bmatrix} 3 & 0 \\ -0.8 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right) \\ &= \text{hardlim}(0.8) = 1 \end{aligned}$$

Đường biên không phân loại đúng p_3 . Cần phải tiếp tục hiệu chỉnh ${}_1w$:

$${}_1w^{new} = {}_1w^{old} - p_3 = \begin{bmatrix} 3 & 0 \\ -0.8 \end{bmatrix} - \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$$



Hình bên cho thấy mạng perceptron đã học được cách phân loại đúng với 3 vector ngõ vào trên. Nếu đưa vào mạng 1 ngõ vào, nó sẽ cho ra ngõ ra đúng với phân loại. Từ đây ta có luật thứ 3 và cũng là cuối cùng: nếu mạng phân loại đúng thì giữ nguyên ${}_1w$.

Tóm lại ta có 3 luật cho tất cả các trường hợp:

$$\text{If } t = 1 \text{ and } a = 0, \text{ then } {}_i w^{new} = {}_i w^{old} - p$$

$$\text{If } t = 0 \text{ and } a = 1 \text{ then } {}_i w^{new} = {}_i w^{old} - p$$

$$\text{If } t = a, \text{ then } {}_i w^{new} = {}_i w^{old}$$

Thống nhất luật học:

Ta đặt e là sai số của mạng perceptron: $e = t - a$

Ta có thể viết lại 3 luật trên như sau:

$$\text{If } e = 1, \text{ then } {}_i w^{new} = {}_i w^{old} + p$$

$$\text{If } e = -1 \text{ then } {}_i w^{new} = {}_i w^{old} - p$$

$$\text{If } e = 0, \text{ then } {}_i w^{new} = {}_i w^{old}$$

Ta cũng có thể hợp nhất 3 luật thành một biểu thức đơn giản:

$${}_i w^{new} = {}_i w^{old} - e p = {}_i w^{old} - (t - a) p$$

Có thể áp dụng luật này để huấn luyện ngưỡng phân cực (thay $p=1$)

$${}_i w^{new} = {}_i w^{old} - e$$

Huấn luyện mạng Perceptron đa neutron:

Chúng ta có thể tổng quát hóa qui luật cập nhật vector trọng số cho mạng perceptron đa neutron.

Công thức để cập nhật hàng thứ i của ma trận trọng số:

$${}_i w^{new} = {}_i w^{old} + e_i p$$

Công thức để cập nhật hàng thứ i vector phân cực:

$$b_i^{new} = b_i^{old} + e_i$$

Luật perceptron được viết gọn lại thành hệ thống ma trận kí hiệu như sau:

$$W^{new} = W^{old} + e p^T$$

$$b^{new} = b^{old} + e$$

Ví dụ 1.10: Ví dụ về luật học perceptron

Bài toán nhận biết quả táo/quả cam. Các vector mẫu ngõ vào/ra là:

$$\left\{ p_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, t_1 = [0] \right\} \quad \left\{ p_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, t_2 = [1] \right\}$$

Với qui ước rằng: ngõ ra yêu cầu sẽ bằng 0 khi mẫu vào là “quả cam”
ngõ ra yêu cầu sẽ bằng 1 khi mẫu vào là “quả táo”

Cảm biến hình dạng: 1 : tròn
-1 : không tròn

Cảm biến bề mặt: 1 : nhẵn
-1 : xù xì

Cảm biến trọng lượng: 1 : nặng hơn 1 pound
-1 : nhẹ hơn 1 pound

Đầu tiên, chúng ta sẽ khởi động ma trận trọng số và ngưỡng phân cực:

$$\mathbf{W} = [0.5 \ -1 \ -0.5], \quad b = 0.5.$$

➤ Ta đưa vectơ ngõ vào \mathbf{p}_1 đến hệ thống mạng:

$$\begin{aligned} a &= \text{hardlim}(\mathbf{W}\mathbf{p}_1 - b) = \text{hardlim}\left([0.5 \ -1 \ -0.5] \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - 0.5\right) \\ &= \text{hardlim}(2.5) = 1 \end{aligned}$$

Sau đó, chúng ta tính toán sai số:

$$e = t_1 - a = 0 - 1 = -1.$$

Trọng số được cập nhật là:

$$\begin{aligned} \mathbf{W}^{new} &= \mathbf{W}^{old} + e\mathbf{p}^T = [0.5 \ -1 \ -0.5] + (-1)[1 \ -1 \ -1] \\ &= [-0.5 \ 0 \ 0.5]. \end{aligned}$$

Ngưỡng phân cực được cập nhật là:

$$b^{new} = b^{old} + e = 0.5 + (-1) = -0.5.$$

Kết thúc lần lặp đầu tiên.

➤ Lần lặp thứ 2 của luật perceptron sẽ là:

$$a = \text{hardlim}(\mathbf{W}\mathbf{p}_2 + b) = \text{hardlim}\left([-0.5 \ 0 \ 0.5] \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} + (-0.5)\right)$$

$$= \text{hardlim}(-0.5) = 0$$

$$e = t_2 - a = 1 - 0 = 1$$

$$\mathbf{W}^{new} = \mathbf{W}^{old} + e\mathbf{p}^T = [-0.5 \ 0 \ 0.5] + 1[1 \ 1 \ -1] = [0.5 \ 1 \ -0.5]$$

$$b^{new} = b^{old} + e = -0.5 + 1 = 0.5$$

- Lần lặp thứ 3 sẽ bắt đầu với vector ngõ vào của lần lặp đầu tiên:

$$a = \text{hardlim}(\mathbf{W}\mathbf{p}_1 + b) = \text{hardlim}\left(\begin{bmatrix} 0.5 & 1 & -0.5 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + 0.5\right)$$

$$= \text{hardlim}(0.5) = 1$$

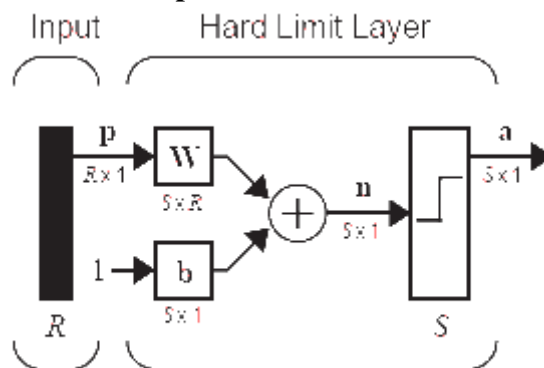
$$e = t_1 - a = 0 - 1 = -1$$

$$\begin{aligned} \mathbf{W}^{new} &= \mathbf{W}^{old} + e\mathbf{p}^T = \begin{bmatrix} 0.5 & 1 & -0.5 \end{bmatrix} + (-1)\begin{bmatrix} 1 & -1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} -0.5 & 2 & 0.5 \end{bmatrix} \end{aligned}$$

Nhận xét: Nếu tiếp tục với các lần lặp kế, ta sẽ nhận ra rằng: giờ đây, cả hai vector ngõ vào đều đã được phân loại chính xác. Thuật toán (các phép lặp) trên đã hội tụ về cùng một đáp án.

Tóm tắt thuật toán:

Cấu trúc mạng tế bào-Perceptron



$$a = \text{hardlim}(\mathbf{W}\mathbf{p} - b)$$

$$a_i = \text{hardlim}(n_i) = \text{hardlim}({}_i\mathbf{W}^T\mathbf{p} + b_i)$$

Ma trận trọng số:

$$\mathbf{W} = \begin{bmatrix} {}_1\mathbf{w}^T \\ {}_2\mathbf{w}^T \\ \vdots \\ {}_s\mathbf{w}^T \end{bmatrix}$$

Đường biên ranh giới: ${}_i\mathbf{w}^T\mathbf{p} + b_i = 0$

Đường ranh giới luôn vuông góc (trực giao) với vector trọng số.

Luật học tế bào - Perceptron:

$$W^{new} = W^{old} + ep^T$$

$$b^{new} = b^{old} + e$$

với: $e = t - a$.

4. Hạn chế của luật học Perceptron:

Mạng perceptron phân loại chính xác các vectơ ngõ vào bằng một đường thẳng tuyến tính. Do đó, mạng perceptron sẽ gặp khó khăn khi giải các bài toán không thể phân chia tuyến tính được. Vì thế, ngoài mạng perceptron, còn có nhiều phương pháp thích nghi trọng số khác cũng được sử dụng.

Ví dụ: Phương pháp Widrow-Hoff tính toán sai lệch giữa ngõ ra mong muốn với ngõ ra thật sự của mạng:

$$\Delta = d(t) - y(t)$$

Sau đó cập nhật trọng số của mạng theo luật:

$$w_i(t+1) = w_i(t) - \alpha \Delta x_i(t)$$

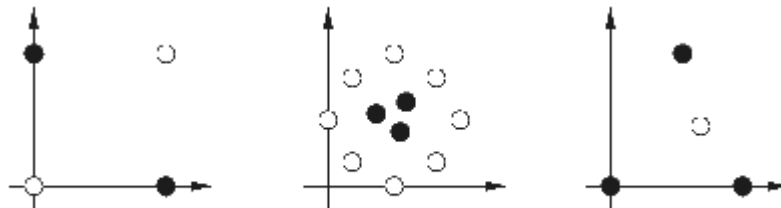
Trong đó:

$d(t)$ là ngõ ra mong muốn và $y(t)$ là ngõ ra của mạng
 α là hệ số thích nghi ($0 < \alpha < 1$)

Widrow gọi những mạng sử dụng thuật toán trên là mạng thích nghi tuyến tính (Adaptive Linear Neural, ADALINE). Việc kết nối nhiều mạng ADALINE lại với nhau tạo nên một cấu trúc gọi là mạng MADALINE (many ADALINE). Thuật toán huấn luyện các mạng này xuất phát từ việc cố gắng tìm một đường thẳng phân chia 2 vùng không gian của bài toán. Tuy nhiên, có rất nhiều trường hợp các vùng không gian không thể phân chia tuyến tính được.

Một ví dụ kinh điển cho loại bài toán này cổng logic XOR. Các ngõ vào và ra của cổng XOR được thể hiện như sau:

$$\left\{ p_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, r_1 = 0 \right\} \left\{ p_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, r_2 = 1 \right\} \left\{ p_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, r_3 = 1 \right\} \left\{ p_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, r_4 = 0 \right\}$$



Một số bài toán không thể phân chia tuyến tính được.

Để giải quyết các bài toán trên ta phải sử dụng mạng truyền thẳng nhiều lớp.

1.3.5. Mạng truyền thẳng nhiều lớp (*Multilayer perceptron _ MLP*) :

Để khắc phục những khó khăn đối với những bài toán có mẫu phân chia không tuyến tính, mạng nơ-ron nhiều lớp được sử dụng. Có rất nhiều công trình nghiên cứu về mạng MLP và đã cho thấy nhiều ưu điểm của mạng này. Mạng MLP là một giải pháp hữu hiệu cho việc mô hình hoá, đặc biệt với quá trình phức tạp hoặc cơ chế chưa rõ ràng. Nó không đòi hỏi phải biết trước dạng hoặc tham số. Mạng MLP là cơ sở cho thuật toán lan truyền ngược và khả năng xấp xỉ liên tục.

1. Thuật toán lan truyền ngược (Back Propagation _BP):

Giải thuật lan truyền ngược được đặt ra bởi Rumelhart, McClelland và Williams vào những năm 80 của thế kỷ 20 là một trong những giải thuật phổ biến để huấn luyện mạng nơ-ron nhiều lớp. Đây cũng là một kiểu học có giám sát. Trọng số của mỗi nơ-ron được chỉnh định theo hướng tỉ lệ với sai số của nơ-ron mà nó kết nối. Việc lan truyền ngược những sai số này từ ngõ ra đến ngõ vào của mạng có thể chỉnh định tất cả các trọng số của mạng.

Giải thuật lan truyền ngược huấn luyện mạng nơ-ron nhiều lớp:

Tập dữ liệu đã cho có n mẫu (x_n, d_n) , với mỗi n , x_n là tín hiệu đầu vào, d_n là đầu ra mong muốn. Quá trình học là việc thực hiện cực tiểu hoá hàm G sau:

$$G = \sum_{n=1}^N G_n, \text{ với } G_n = \frac{1}{N} \sum_{q=1}^N (yq(x_n) - dq(x_n))^2$$

Q là số nút tại lớp ra của mạng.

Trọng số liên kết mạng được điều chỉnh theo phép lặp sau :

$$w(k+1) = w(k) - \eta \frac{\partial G}{\partial w}, \text{ trong đó } \eta > 0 \text{ là hằng số tốc độ học.}$$

Thuật toán lan truyền ngược đề cập ở các phần trên đã được ứng dụng rất rộng rãi trong việc huấn luyện các mạng nơ-ron. Tuy nhiên, nó vẫn mắc phải 2 khuyết điểm cố hữu là: **tốc độ hội tụ chậm và dễ bị rơi vào các vùng cực trị cục bộ**. Hai khuyết điểm này có thể hạn chế bằng việc lựa chọn tối ưu tốc độ học (hệ số η) và sử dụng bổ sung hệ số momentum vào luật học:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) - \alpha \Delta w_{ij}(t-1)$$

Trong đó: α là hệ số momentum

Tuy có thể được cải thiện phần nào chất lượng học nhưng rõ ràng điều này vẫn chưa đủ để có thể ứng dụng giải thuật lan truyền ngược vào thực tế do tốc độ hội tụ chậm của nó. Và những hạn chế này có thể được khắc phục khi sử dụng mạng hàm cơ sở xuyên tâm.

2. Mạng hàm cơ sở xuyên tâm (Radial basis functions _ RBF)

Mạng RBF có thể được huấn luyện nhanh chóng và do có cấu trúc mạng đặc biệt nên mạng RBF rất ít khi rơi vào các vùng cực trị cục bộ.

Cấu trúc mạng RBF: Mạng RBF cũng là một mạng truyền thẳng ba lớp

➤ Lớp ẩn sử dụng:

- Hàm tổng ngõ vào có dạng hàm cầu.

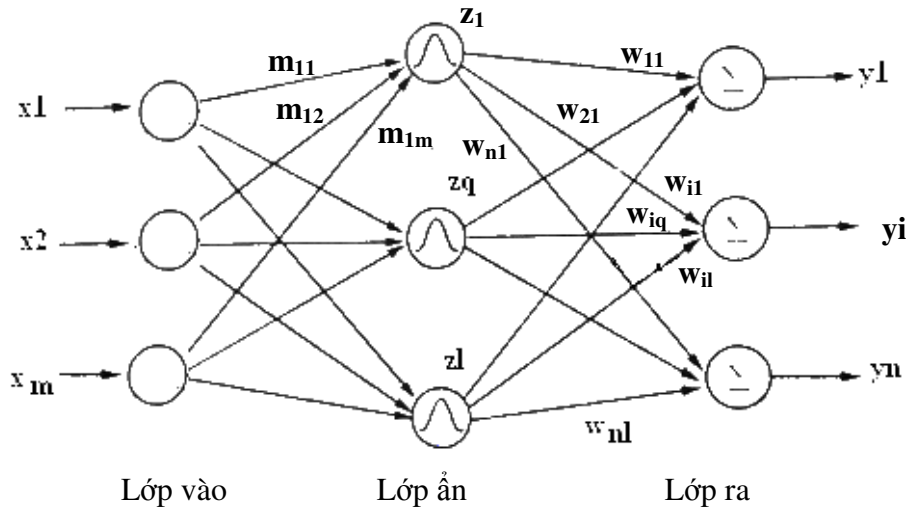
- Hàm kích hoạt là hàm Gauss có dạng: $\varphi(x) = e^{\frac{-x^2}{2\sigma_q^2}}$

➤ Lớp ra sử dụng:

- Hàm tổng ngõ vào là hàm tuyến tính.

- Hàm kích hoạt là hàm kích hoạt đơn vị.

Trong mạng RBF thì ngưỡng của các nơ-ron là bằng 0.



Các phương trình toán truyền tín hiệu từ lớp vào đến lớp ra của mạng:

➤ Ngõ ra của nơ-ron thứ q (thuộc lớp ẩn):

$$z_q = e^{\frac{-\|x-m_q\|^2}{2\sigma_q^2}}$$

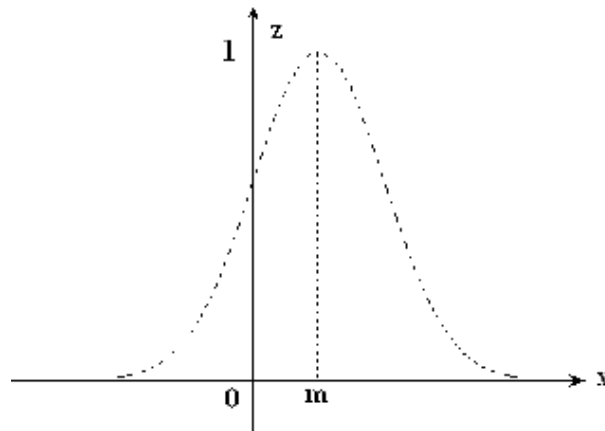
Trong đó:

X là vector ngõ vào

m_q là tâm hàm RBF (trọng số của nơ-ron lớp ẩn thứ q)

σ_q là bề rộng hàm RBF của nơ-ron ẩn thứ q

$$\|X - m_q\| = \sqrt{(x - m_q)^T (x - m_q)} \text{ khoảng cách Euclide}$$



Ngõ ra của nơ-ron lớp ẩn

➤ Ngõ ra của nơ-ron thứ i (thuộc lớp ra):

$$y_i = \sum_{q=1}^l w_{iq} z_q \quad (i = \overline{1, n}; \quad q = \overline{1, l})$$

➤ Hàm năng lượng sai số:

$$E = \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2$$

Khi sử dụng mạng RBF, ta cần chú ý các điểm sau:

- Lựa chọn dạng hàm kích hoạt φ phù hợp với mô hình.
- Tính toán số lượng tâm tối ưu. Nếu nhiều tâm quá sẽ không đủ dữ liệu để huấn luyện mạng, còn nếu tâm ít quá sẽ cho mô hình sai lệch.
- Tìm vị trí tâm và số lượng dữ liệu thích hợp để huấn luyện mạng.

Thuật toán huấn luyện mạng RBF, bao gồm các bước sau:

Bước 1: Chọn tốc độ học $\eta > 0$, chọn sai số cực đại E_{\max}

Bước 2: Đặt giá trị đầu: $E = 0$; $k = 1$

Gán giá trị ngẫu nhiên cho các trọng số $w_{iq}(k)$

Bước 3: Tính ngõ ra của mạng với tín hiệu vào là $x(k)$:

$$z_q(k) = e^{-\frac{\|x(k) - m_q\|^2}{2\sigma_q}}$$

$$y_i(k) = \sum_{q=1}^l w_{iq}(k) z_q(k) \quad (i = \overline{1, n}; \quad q = \overline{1, l})$$

Cập nhật trọng số cho lớp ra của mạng:

$$w_{iq}(k+1) = w_{iq}(k) + \eta(d_i(k) - y_i(k))z_q(k)$$

Tính sai số tích lũy:

$$E = E + \frac{1}{2} \sum_{i=1}^n (d_i(k) - y_i(k))^2$$

Thực hiện vòng lặp ở bước 3 với k chạy từ 1 đến K.

Bước 4: Kết thúc một chu trình huấn luyện

Nếu $E < E_{\max}$ thì kết thúc quá trình học

Nếu $E > E_{\max}$ thì gán $E = 0$; $K = 1$ và trở lại bước 3.

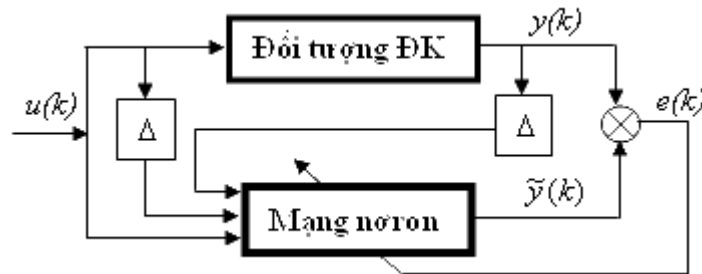
Nhận xét:

Mạng RBF không bị ảnh hưởng bởi vấn đề cực trị cục bộ. Tuy nhiên, RBF thường đáp ứng chậm trong giai đoạn nhó do có một lượng lớn nơron ở lớp ẩn. Nhưng ta có thể cải thiện vấn đề này bằng việc chọn tâm và độ phân tán phù hợp.

Nhờ khả năng xấp xỉ các hàm phi tuyến bất kỳ với độ chính xác tùy ý, mạng nơron, đặc biệt là mạng RBF là công cụ quan trọng cho mô hình hoá hệ thống và cho điều khiển thích nghi các hệ thống phi tuyến.

1.3.6. Nhận dạng mô hình và điều khiển sử dụng mạng nơron

1. Nhận dạng thông số mô hình

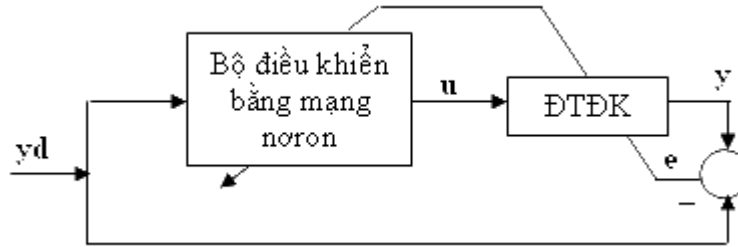


Nhận dạng thông số chính là quá trình luyện mạng. Tín hiệu sai số $e = y - \tilde{y}$ là cơ sở cho luyện mạng, Δ là thời gian trễ.

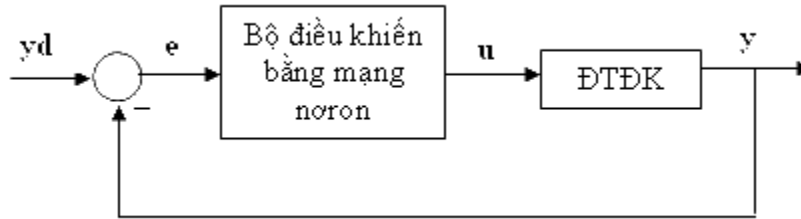
2. Điều khiển sử dụng mạng nơron

Ta có nhiều cấu trúc điều khiển sử dụng mạng nơron khác nhau. Có thể kể đến một số cấu trúc điều khiển như sau:

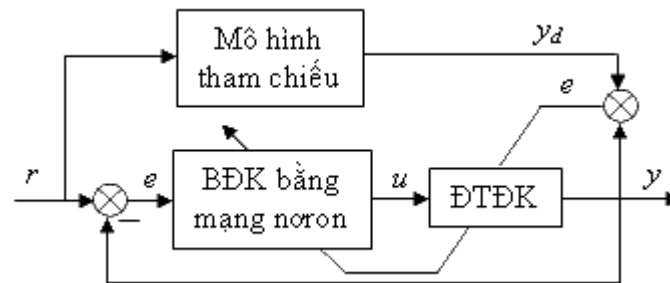
+ Điều khiển theo vòng hở



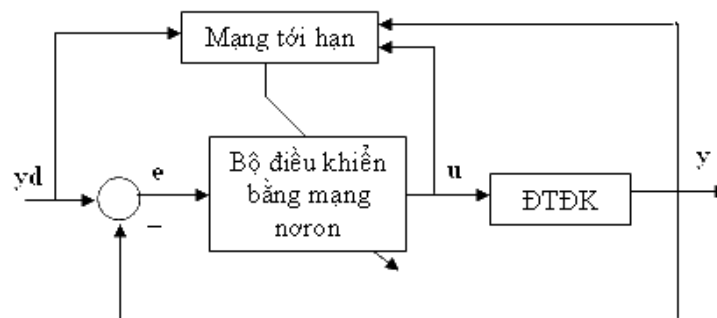
+ Điều khiển theo vòng kín



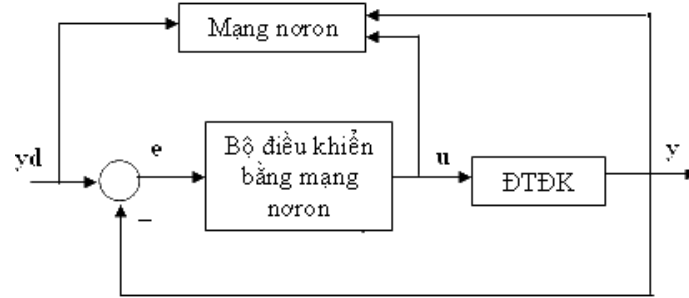
+ Điều khiển với mô hình tham chiếu và sai số lan truyền qua ĐTĐK



+ Điều khiển theo thời gian vượt quá (over time)



+ Bộ điều khiển với quyết định hỗ trợ của mạng nơron



3. Ứng dụng mạng RBF để nhận dạng hệ động lực học phi tuyến

Xét hệ động lực học phi tuyến của ĐTĐK

$$\dot{x} = f(x) + g(x)u \quad (1.79)$$

Giả sử ĐTĐK là ổn định vòng hở, vectơ trạng thái x là quan sát được. Cần tìm mô hình xấp xỉ (1.79). Chọn $A \in \mathbb{R}^{n \times n}$ là ma trận ổn định, ta viết lại (1.79) dạng :

$$\dot{x} = Ax + (f(x) - Ax) + g(x)u$$

Theo tính chất xấp xỉ của mạng RBF cho hàm phi tuyến: Nếu số lượng các nút trong lớp ẩn là đủ lớn thì $f(x) - Ax$ và $g(x)$ có thể xấp xỉ bằng các mạng RBF sau:

$$f(x) - Ax = W^*S(x) \quad \text{và} \quad g(x) = V^*S(x)$$

trong đó $W^* \in \mathbb{R}^{n \times N}$ và $V^* \in \mathbb{R}^{n \times N}$ là các ma trận trọng số của các tổ hợp tuyến tính trên. N xác định số lượng nút trong một lớp RBF của mạng.

$$S(x) = [S_1, S_2, \dots, S_N]^T, \text{ vectơ các hàm cơ sở sau:}$$

$$S_k = \left(\|x - C_k\|^2 + \rho_k^2 \right)^{-\frac{1}{2}}, \text{ với } k = 1, 2, 3, \dots, N$$

Tâm $C_k \in \mathbb{R}^n$ và độ rộng $\rho_k \in \mathbb{R}^n$ được biết trước. Ta viết lại (1.79) như sau:

$$\dot{x} = Ax + W^*S(x) + V^*S(x)u$$

Vậy mô hình của đối tượng có thể được mô tả bằng phương trình:

$$\dot{\tilde{x}} = A\tilde{x} + WS(x) + VS(x)u$$

trong đó $W \in \mathbb{R}^{n \times N}$, $V \in \mathbb{R}^{n \times N}$ là các ma trận ước lượng của W^* , V^* , $\tilde{x} \in \mathbb{R}^n$ là ước lượng trạng thái của x . Gọi

$$x_e = \tilde{x} - x, \quad W_e = W^* - W, \quad V_e = V^* - V$$

Phương trình sai số ước lượng sẽ là :

$$\dot{x}_e = Ax_e + W_e S(x) + V_e S(x)u \quad (1.80)$$

Thuật toán nhận dạng sử dụng hàm Lyapunov:

$$L(x_e, W_e, V_e) = \frac{1}{2} x_e^T P x_e + \frac{1}{2} Tr(W_e^T W_e) + \frac{1}{2} Tr(V_e^T V_e) \quad (1.81)$$

với P là ma trận đối xứng xác định dương. Có thể xác định ma trận Q đối xứng xác định dương thỏa phương trình Lyapunov sau:

$$PA + A^T P = -Q.$$

Thay (1.80) vào (1.81) và lấy đạo hàm ta được:

$$\dot{L} = \frac{1}{2} x_e^T (PA + A^T P) x_e + S^T(x) W_e^T P x_e + S^T(x) V_e^T P x_e u + \frac{1}{2} Tr(\dot{W}_e^T W_e) + \frac{1}{2} Tr(\dot{V}_e^T V_e)$$

$$\text{Chọn :} \quad Tr(\dot{W}_e^T W_e) = -S^T(x) W_e^T P x_e \quad (1.82)$$

$$Tr(\dot{V}_e^T V_e) = -S^T(x) V_e^T P x_e u \quad (1.83)$$

$$\text{thì :} \quad \dot{L}(x_e, W_e, V_e) = -\frac{1}{2} x_e^T Q x_e \quad (1.84)$$

Do các ma trận W^* và V^* là ma trận hằng nên từ (1.82), (1.83) ta suy ra thuật nhận dạng mô hình như sau:

$$\dot{W}_{ij} = -S_j \sum_{k=1}^n P_{ik} x_{ek}$$

$$\dot{V}_{ij} = -S_j \sum_{k=1}^n P_{ik} x_{ek} u$$

với $i = 1, 2, \dots, N$ và $j = 1, 2, \dots, N$, P_{ij} là phần tử của ma trận Lyapunov P.

Từ (1.81) ta thấy rằng $L(x_e, W_e, V_e) \geq 0$

Từ (1.84) nhận được $\dot{L}(x_e, W_e, V_e) \leq 0$

Vì vậy $x_e(t) \rightarrow 0$, $W_e \rightarrow 0$, $V_e \rightarrow 0$, hoặc $\dot{x} \rightarrow x$, $W \rightarrow W^*$, $V \rightarrow V^*$ khi $t \rightarrow \infty$.

Để tính toán đơn giản có thể chọn :

$$A = aI, Q = qI, P = pI, \text{ với } a > 0, q > 0 \text{ và } I \text{ là ma trận đơn vị}$$

Khi đó thuật toán nhận dạng mô hình đơn giản như sau:

$$\dot{W}_{ij} = -p S_j x_{ei}$$

$$\dot{V}_{ij} = -p S_j x_{ei} u$$

Từ phương trình Lyapunov rút ra : $p = \frac{q}{2a} > 0$.

Để hội tụ đến trọng số thực, hệ động lực phải có đủ giàu thông tin ở đầu vào. Vì thế đa số đầu vào được chọn ngẫu nhiên.

1.3.7. Thuật toán học trực tuyến cấu trúc, thông số mạng nơ-ron mờ động

Hầu hết các thuật toán chỉnh định cấu trúc và thông số cho mạng nơ-ron mờ đều là học off-line. Quá trình học off-line (có giám sát) đòi hỏi phải cung cấp trước mẫu dữ liệu vào ra và việc học thông qua các vòng lặp huấn luyện. Do được cung cấp trước toàn bộ thông tin về hệ thống, cộng với ưu thế về mặt thời gian (do off-line nên các vấn đề về **tốc độ học và thời gian học** là không quan trọng) và tài nguyên tính toán nên hầu hết các thuật toán học off-line đều cho chất lượng mạng tốt (tính đặc trưng cao, sai số thấp, cấu trúc đơn giản...).

Vậy thì tại sao phải là on-line khi off-line đã quá đầy đủ? Trong lĩnh vực điều khiển tự động, bài toán điều khiển đối tượng khi đặc tính động học của đối tượng thay đổi đã dành được sự quan tâm của nhiều người. Thực tế đã chứng minh không phải bất kỳ một đối tượng nào cũng mô hình hóa hoàn toàn được (do đặc tính phức tạp của nó) và khi đã mô hình hóa được cũng chưa chắc sẽ sử dụng được (vì đặc tính đối tượng không phải là bất biến theo thời gian). Để giải quyết bài toán này, người ta nghĩ ngay đến các phương pháp điều khiển thích nghi. Mạng nơ-ron mờ với khả năng tự học của một mạng nơ-ron và tính đặc trưng cao của một hệ mờ là lời giải thích hợp nhất cho bài toán trên.

1. Cấu trúc mạng nơ-ron mờ động:

Mạng nơ-ron mờ động (Dynamic Fuzzy Neural Network – DFNN) có cấu trúc dựa trên nền tảng mạng nơ-ron sử dụng hàm cơ sở xuyên tâm (mạng RBF). Điểm khác nhau cơ bản giữa DFNN và mạng RBF là ở số lớp nơ-ron trong mạng: DFNN gồm 5 lớp (mạng RBF chỉ có 3 lớp). Tuy nhiên, DFNN và mạng RBF có sự tương đồng về mặt phương trình toán học truyền tín hiệu trong mạng. Đặc tính “động” (dynamic) của DFNN thể hiện ở cấu trúc mạng – nó được tinh gọn gần như tối đa (số lượng nơ-ron ở lớp 3 của mạng bằng với số ngõ vào) và nhờ vậy nó có thể dễ dàng thay đổi cấu trúc của mình trong khi vẫn đảm bảo được khả năng đặc trưng vốn có của một mạng nơ-ron mờ mà vẫn hạn chế đến mức thấp nhất sự xáo trộn trong số lượng thông số của mạng.

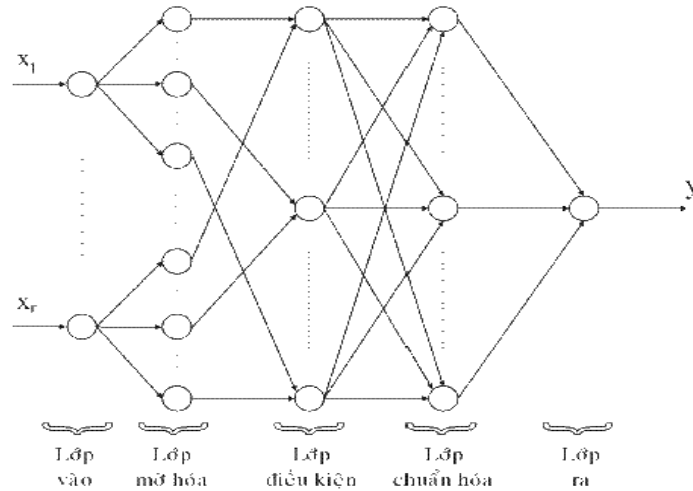
Về mặt chức năng, mạng DFNN tương đương với một hệ thống suy luận mờ Takagi – Sugeno – Kang (TSK). Để đơn giản nhưng vẫn không mất tính tổng quát, ta chỉ xét mạng DFNN nhiều ngõ vào – một ngõ ra. Các kết quả nhận được hoàn toàn có thể ứng dụng đối với mạng nhiều ngõ vào – nhiều ngõ ra. Cấu trúc một mạng DFNN gồm 5 lớp:

➤ Lớp 1 (lớp vào)

Lớp vào gồm r nút, mỗi nút tương ứng với 1 ngõ vào x_i ($i = 1, 2, \dots, r$). Những nút này thực hiện việc chuyển tín hiệu trực tiếp từ ngõ vào đến lớp 2.

Tập hợp r ngõ vào này trong một thời điểm tạo thành một vector ngõ vào:

$$X = [x_1 \ \dots \ x_r]^T \quad (r: \text{số ngõ vào})$$



➤ **Lớp 2 (lớp mờ hóa)**

Mỗi biến ngôn ngữ ngõ vào x_i được đặc trưng bởi u giá trị ngôn ngữ

$$A_i = [a_{i1}, a_{i2}, \dots, a_{iu}] \quad (i = 1, 2, \dots, r)$$

Trong đó : A_i : tập các giá trị ngôn ngữ của ngõ vào x_i ,

a_{ij} : số mờ thứ j của ngõ vào thứ i với hàm liên thuộc dạng Gauss.

Ngõ ra của một nút thuộc lớp 2 sau khi mờ hóa:

$$\mu_{ij}(x_i) = \exp \left[-\frac{(x_i - c_{ij})^2}{\sigma_j^2} \right] \quad (j = 1, 2, \dots, u)$$

với c_{ij} và σ_j là tâm và bề rộng hàm liên thuộc Gauss thứ j của ngõ vào x_i .

➤ **Lớp 3 (lớp điều kiện/lớp luật mờ)**

Lớp 3 có u luật mờ (nơron), mỗi luật mờ là sự kết hợp của r ngõ ra từ lớp 2:

$$R_j: \text{NẾU } (x_1 \text{ là } a_{1j} \text{ và } x_2 \text{ là } a_{2j} \text{ và } \dots \text{ và } x_r \text{ là } a_{rj}) \text{ THÌ } \dots$$

Ngõ ra của mỗi nút trong lớp 3:

$$\phi_j = \prod_{i=1}^r \mu_{ij}(x_i) = \exp \left[-\frac{\sum_{i=1}^r (x_i - c_{ij})^2}{\sigma_j^2} \right] \quad \begin{matrix} (i = 1, 2, \dots, r) \\ (j = 1, 2, \dots, u) \end{matrix}$$

Nếu đặt: $C_j = [c_{1j} \ \dots \ c_{rj}]^T$ là vector chứa tâm hàm liên thuộc của luật thứ j

$$\Rightarrow \phi_j = \exp \left[-\frac{\|X - C_j\|^2}{\sigma_j^2} \right]$$

Ta có thể thấy mỗi nút trong lớp này cũng là một hàm RBF của vector ngõ vào X với tâm là vector C_j và bề rộng là σ_j

➤ **Lớp 4 (lớp chuẩn hóa)**

Lớp 4 gồm u nút, mỗi nút trong lớp này làm nhiệm vụ chuẩn hóa ngõ ra của 1 nút ở lớp 3 tương ứng theo công thức:

$$\psi_j = \frac{\phi_j}{\sum_{k=1}^u \phi_k} \quad (j = 1, 2, \dots, u)$$

➤ **Lớp 5 (lớp ra)**

Lớp này là tổng có trọng số của các ngõ ra ở lớp 4:

$$y = \sum_{j=1}^u w_j \psi_j$$

Với w_j là giá trị của mệnh đề THÌ của một luật mờ NẾU-THÌ:

R_j : NẾU (x_1 là a_{1j} và x_2 là a_{2j} và ... và x_r là a_{rj}) THÌ (y là w_j)

2. Thuật toán học cấu trúc và thông số on-line cho mạng DFNN

Khái niệm về học on-line:

Đây là một quá trình học có giám sát. Ban đầu, cấu trúc mạng là hoàn toàn “rỗng”, chỉ có các nơ-ron lớp vào và lớp ra (đây là những nơ-ron cố định trong suốt quá trình học). Học on-line, có nghĩa là mạng sẽ không được cung cấp trước một tập mẫu dữ liệu vào- ra, các mẫu dữ liệu vào ra này sẽ đến với mạng ngay trong quá trình học, từng mẫu từng mẫu một. Do đó, mạng buộc phải hoàn tất một quá trình tính toán (chính định cấu trúc và thông số) trước khi lại tiếp tục nhận được một mẫu dữ liệu vào ra mới. Quá trình học on-line là quá trình tiếp diễn không ngừng, từ mẫu dữ liệu vào ra đầu tiên cho đến khi hệ thống ngừng hoạt động hoặc mạng được lệnh ngừng học. Vì vậy, thuật toán học on-line phải thỏa mãn các yêu cầu sau: tốc độ hội tụ nhanh, cấu trúc mạng tinh gọn (số thông số ít) và có thể vận hành trong điều kiện tài nguyên tính toán hạn chế.

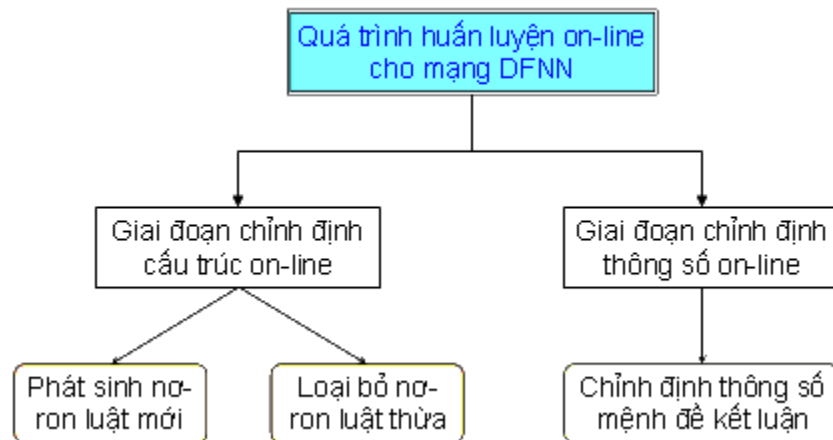
Thuật toán học on-line cho mạng DFNN:

Khi mẫu dữ liệu vào ra đầu tiên vào mạng sẽ xác lập một cấu trúc ban đầu cho mạng – mạng có 1 nơ-ron luật.

Khi mạng nhận được mẫu dữ liệu tiếp theo, quá trình chỉnh định thông số sẽ được tiến hành trong mỗi chu kỳ lấy mẫu thông qua thuật toán bình phương tối thiểu đệ quy. Quá trình chỉnh định cấu trúc chỉ xảy ra khi:

- sai số ngõ ra của mạng tương đối lớn
- khả năng đặc trưng dữ liệu đầu vào của mạng kém – mạng sẽ được bổ sung thêm nơ-ron luật.

Sau một thời gian, các nơ-ron luật mới thêm vào do khả năng đặc trưng tốt hơn sẽ vô hiệu hóa sự ảnh hưởng của các nơ-ron luật cũ trước đó. Các nơ-ron luật cũ này sẽ bị xóa bỏ khỏi mạng bằng kỹ thuật giảm lược để giữ cho cấu trúc mạng được tinh gọn. Quá trình này cứ tiếp tục cho đến mẫu dữ liệu vào ra cuối cùng mà mạng nhận được.



Thuật toán học cho mạng DFNN bao gồm 2 quá trình chính: quá trình chỉnh định cấu trúc và quá trình chỉnh định thông số

A. Giai đoạn chỉnh định cấu trúc on-line:

- Tiêu chuẩn phát sinh luật

Để một luật mới (một neural ở lớp 3) được phát sinh thì hệ thống phải thỏa mãn 2 điều kiện:

- Sai số hệ thống: với mỗi mẫu dữ liệu vào ra thu thập được tại thời điểm t (X_t, d_t), tính sai số của hệ thống:

$$\|e_t\| = \|d_t - y_t\| \quad (1.85)$$

với y_t là ngõ ra của mạng DFNN

Nếu như $\|e_t\| > k_e$ thì thỏa mãn điều kiện thứ nhất để phát sinh luật mới. Ở đây, k_e là hệ số không cố định mà giảm dần theo thời gian:

$$k_e = \max [e_{\max} \times \beta^t, e_{\min}] \quad (1.86)$$

với $0 < \beta < 1$ được gọi là hằng số suy giảm, e_{\max} là sai số tối đa và e_{\min} là độ chính xác mong muốn.

- **Khoảng cách tối thiểu:** với mỗi mẫu dữ liệu vào ra thu thập được tại thời điểm t (X_t, d_t), tính các khoảng cách tương đối giữa vector ngõ vào X và tâm các hàm liên thuộc C_j :

$$md_j = \|X_t - C_j\| \quad (j = 1, 2, \dots, u) \quad (1.87)$$

$$\text{Tìm } J = \arg \min_j (md_j)$$

Nếu như $md_{\min} = md_J > k_d$, có nghĩa là ngõ vào hiện tại không thể được đặt trung bởi các luật mờ của hệ thống hiện tại, thì thỏa mãn điều kiện thứ hai để phát sinh luật mới. Ở đây, k_d là hệ số không cố định mà giảm dần theo thời gian:

$$k_d = \max [d_{\max} \times \gamma^t, d_{\min}] \quad (1.88)$$

với $0 < \gamma < 1$ được gọi là hằng số suy giảm, d_{\max} và d_{\min} là khoảng cách tối đa và tối thiểu tùy thuộc vào từng đối tượng.

Có 4 trường hợp có thể xảy ra:

- Trường hợp 1: $\|e_t\| \leq k_e, md_{\min} \leq k_d$. Điều này thể hiện mạng đã đặc trưng rất tốt cho mẫu dữ liệu hiện tại (X_t, d_t). Do đó không thực hiện bất cứ thay đổi nào trong mạng hoặc chỉ thực hiện việc chỉnh định thông số mệnh đề kết luận.
- Trường hợp 2: $\|e_t\| \leq k_e, md_{\min} > k_d$. Mặc dù mẫu dữ liệu hiện tại (X_t, d_t) không được đặt trung tốt nhưng mạng vẫn mô tả đúng đối tượng. Điều này xảy ra khi mẫu dữ liệu (X_t, d_t) không mang nhiều thông tin mô tả đối tượng. Do đó không thực hiện bất cứ thay đổi nào trong mạng hoặc chỉ thực hiện việc chỉnh định thông số mệnh đề kết luận.

- Trường hợp 3: $\|e_t\| > k_e, md_{\min} \leq k_d$. Mặc dù mẫu dữ liệu hiện tại (X_t, d_t) được đặt trung tốt bởi hàm RBF ở lớp 3 gần nhất nhưng mạng không thể mô tả đúng đối tượng, có nghĩa là thông số các hàm RBF hiện tại chưa đúng. Do đó, quá trình chỉnh định thông số mệnh đề mệnh đề kết luận phải được thực hiện.
- Trường hợp 4: $\|e_t\| > k_e, md_{\min} > k_d$. Cấu trúc mạng DFNN hiện tại không đủ khả năng mô hình hóa đối tượng, do đó phải thực hiện việc phát sinh một luật mới.

- *Thông số ban đầu cho luật mới*

Sau khi một luật mới đã được phát sinh, vấn đề tiếp theo là làm thế nào để xác định các thông số ban đầu cho luật mới này. Các thông số của một luật mới thứ $u+1$ bao gồm vector vị trí $C_{u+1} = [c_{1(u+1)}, c_{2(u+1)}, \dots, c_{r(u+1)}]^T$ và bề rộng σ_{u+1} của các hàm RBF mới ở lớp 3. Trong mạng DFNN, bề rộng của một hàm RBF ảnh hưởng rất lớn đến tính đặc trưng của mạng. Nếu 2 hàm RBF kế cận nhau không chồng lấp hợp lý lên nhau thì một ngõ vào có vị trí ở giữa 2 hàm này sẽ không được đặc trưng tốt và ngõ ra của mạng sẽ chứa đựng rất ít thông tin về ngõ vào này. Trong khi đó, nếu bề rộng của hàm RBF quá lớn thì các ngõ vào xung quanh hàm RBF này luôn cho giá trị mờ hóa là 1 và ngõ ra của mạng cũng sẽ bị ảnh hưởng rất ít bởi ngõ vào này. Cả 2 trường hợp trên đều cho kết quả là mô hình nhận dạng được sẽ không mô tả tốt hệ thống thực. Trên sơ sở đó, các thông số cho luật mới có thể được chọn như sau:

$$C_{u+1} = X_t \quad (1.89)$$

$$\sigma_{u+1} = knr \times md_{\min} \quad (i = 1, 2, \dots, r) \quad (1.90)$$

với knr là hệ số chồng lấp (thường được chọn trong khoảng 1.05 đến 1.2). Với cách cấp phát thông số ban đầu như trên sẽ đảm bảo tính chồng lấp hợp lý của hàm RBF mới được tạo.

Ban đầu mạng chưa tồn tại bất cứ hàm RBF nào, khi mẫu dữ liệu đầu tiên (X_1, d_1) được thu thập, mạng sẽ phát sinh luật đầu tiên. Và các thông số ban đầu cho luật này cũng có thể được cấp phát theo (1.89) và (1.90) như sau:

$$C_1 = X_1$$

$$\sigma_1 = \sigma_0 \quad (i = 1, 2, \dots, r)$$

với σ_0 là hằng số xác định tùy thuộc vào miền xác định của X .

- Kỹ thuật giảm lược

Đôi khi, một luật mới có thể được tạo ra khi thỏa mãn 2 điều kiện phát sinh luật nhưng sau đó lại đóng góp không đáng kể đến khả năng nhận dạng của mạng. Những luật dư thừa như vậy không những làm tăng khối lượng tính toán của hệ thống mà còn có thể ảnh hưởng đến chất lượng nhận dạng của mạng DFNN. Do đó, một đòi hỏi đặt ra là làm thế nào để giảm lược những luật thừa này để tối giản đến mức thấp nhất có thể bộ luật của hệ thống (từ đó tối thiểu số thông số của mạng) mà vẫn giữ nguyên chất lượng nhận dạng của mạng. Trong quá trình học on-line của mạng DFNN, một kỹ thuật giảm lược dựa trên thuật toán Error Reduction Ratio (ERR) được sử dụng để tự động loại bỏ những luật thừa như sau:

Cho n cặp mẫu dữ liệu vào ra (X_k, d_k) , $k = 1, 2, \dots, n$. Đặt:

$$PSI = \begin{bmatrix} \psi_{11} & \psi_{11}x_{11} & \cdots & \psi_{11}x_{r1} & \psi_{u1} & \psi_{u1}x_{11} & \cdots & \psi_{u1}x_{r1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi_{1n} & \psi_{1n}x_{1n} & \cdots & \psi_{1n}x_{rn} & \psi_{un} & \psi_{un}x_{1n} & \cdots & \psi_{un}x_{rn} \end{bmatrix} \quad (1.91)$$

$$W = [\alpha_{10} \dots \alpha_{1r} \dots \alpha_{u0} \dots \alpha_{ur}] \in \mathfrak{R}^{u(r+1) \times 1} \quad (1.92)$$

$$Y = [y(1) \quad \cdots \quad y(n)]^T \in \mathfrak{R}^{n \times 1} \quad (1.93)$$

Trong đó:

$$PSI \in \mathfrak{R}^{u(r+1) \times 1} : \text{là ma trận chứa các phần tử hồi quy} \quad (1.94)$$

$$W \in \mathfrak{R}^{u(r+1) \times 1} : \text{là vector thông số mệnh đề kết luận}$$

Có thể xem ngõ ra mong muốn của mạng như một mô hình hồi quy tuyến tính như sau:

$$D = Y + E = PSI \times W + E \quad (1.95)$$

Với:

$$E = [e(1) \quad \cdots \quad e(n)]^T \text{ là sai số ngõ ra với giả thiết là nhiễu trắng trung bình bằng 0}$$

$$D = [d(1) \quad \cdots \quad d(n)]^T \in \mathfrak{R}^{n \times 1} \text{ là ngõ ra mong muốn của hệ thống.}$$

Với ma trận PSI bất kỳ có số hàng lớn hơn số cột ($n > u(r+1)$), có thể dùng phép phân ly nhanh QR để chuyển ma trận PSI sang ma trận chứa các vector cơ sở trực giao:

$$PSI = QA \quad (1.96)$$

Trong đó:

$$Q = [q_1 \quad q_2 \quad \cdots \quad q_{u(r+1)}] \in \mathfrak{R}^{n \times u(r+1)} \text{ có các cột là các vector trực giao nhau } (Q^T Q = I)$$

$A \in \mathfrak{R}^{u(r+1) \times u(r+1)}$ là ma trận tam giác trên

Phép biến đổi trên giúp dễ dàng tính mức độ đóng góp của mỗi vector cơ sở trực giao q_i tới ngõ ra mong muốn của hệ thống.

Từ (1.96) và (1.95):

$$D = QAW + E = QG + E \quad (1.97)$$

với $G = AW$.

Dùng phương pháp bình phương tối thiểu tuyến tính để giải phương trình (1.97) suy ra $G = (Q^T Q)^{-1} Q^T D \in \mathfrak{R}^{u(r+1) \times 1}$

hay:

$$g_i = \frac{q_i^T D}{q_i^T q_i} \quad (i = 1, 2, \dots, u(r+1)) \quad (1.98)$$

Do các vector q_i và q_j trực giao nhau khi $i \neq j$, tổng bình phương ngõ ra mong muốn cho bởi công thức:

$$D^T D = \sum_{i=1}^{u(r+1)} g_i^2 q_i^T q_i + E^T E \quad (1.99)$$

Theo phương trình (1.99) có thể thấy thành phần $g_i^2 q_i^T q_i$ đặc trưng cho sự đóng góp của vector cơ sở q_i đến bình phương ngõ ra mong muốn. Do đó, có thể định nghĩa các tỉ số err_i để đặc trưng cho sự đóng góp này như sau:

$$err_i = \frac{g_i^2 q_i^T q_i}{D^T D} \quad (i = 1, 2, \dots, u(r+1)) \quad (1.100)$$

Thay (1.98) vào (1.100). Suy ra:

$$err_i = \frac{(q_i^T D)^2}{q_i^T q_i D^T D} = \frac{(q_i^T D)^2}{D^T D} \quad (i = 1, 2, \dots, u(r+1)) \quad (1.101)$$

Phương trình (1.101) cung cấp một cách thức đơn giản nhưng hiệu quả để tính mức độ đóng góp vào ngõ ra của các phần tử hồi quy. Ý nghĩa của (1.101) là err_i tượng trưng cho sự tương quan giữa q_i và D . Nếu err_i lớn có nghĩa q_i có đóng góp đáng kể đến ngõ ra hệ thống.

Chuyển vector $ERR_i = [err_1 \quad err_2 \quad \dots \quad err_{u(r+1)}] \in \mathfrak{R}^{1 \times u(r+1)}$ sang ma trận $\Delta = [\delta_1 \quad \delta_2 \quad \dots \quad \delta_u]^T \in \mathfrak{R}^{u \times (r+1)}$ với hàng thứ j của Δ tương ứng với luật thứ j của mạng DFNN. Để tính mức độ đóng góp của luật thứ j vào ngõ ra của hệ thống, định nghĩa:

$$\eta_j = \sqrt{\frac{\delta_j \delta_j^T}{r+1}} \quad (1.102)$$

Nếu $\eta_j < k_{err}$ với k_{err} là một hằng số xác định, có nghĩa là luật thứ j của mạng đóng góp không đáng kể đến ngõ ra của hệ thống trong n tập mẫu dữ liệu. Do đó, luật thứ j của mạng có thể được xóa bỏ.

B. Chính định thông số mệnh đề kết luận

Mục tiêu của việc chính định thông số mệnh đề kết luận là tối thiểu bình phương sai số nhận dạng của mạng:

$$E = \frac{1}{2} [d(t) - y(t)]^2 \quad (1.103)$$

Các thông số mệnh đề kết luận có thể chính định thông qua phương pháp bình phương tối thiểu đệ quy (Recursive Least Square RLS) như sau:

- Bước 1: Tính sai số nhận dạng

$$e(t) = d(t) - y(t) \quad (1.104)$$

- Bước 2: cập nhật hệ số quên λ

$$\lambda(t) = \lambda(t-1) + [1 - \lambda(t-1)] \Delta \lambda \quad (1.105)$$

với $\Delta \lambda$ là một hằng số dương xác định

- Bước 3: tính ma trận tạm

$$L(t) = \frac{P(t-1)PSI_n^T}{\lambda(t) + PSI_n P(t-1)PSI_n^T} \quad (1.106)$$

với PSI_n là hàng thứ n của ma trận PSI trong phương trình (1.91)

- Bước 4: cập nhật ma trận hiệp phương sai $P(t)$

$$P(t) = \frac{P(t-1) - L(t)PSI_n P(t-1)}{\lambda(t)} \quad (1.107)$$

trong đó $P(0) = \mu I$ với μ là số dương lớn và I là ma trận đơn vị $u(r+1) \times u(r+1)$

- Bước 5: cập nhật vector thông số mệnh đề kết luận $W(t)$

$$W(t) = W(t-1) + L(t)e(t) \quad \text{với } W(0) = 0 \quad (1.108)$$

1.3.8. Kết luận

Sau khi đã nghiên cứu về mạng nơ-ron nhân tạo, ta thấy mặc dù có rất nhiều cấu trúc mạng cũng như thuật toán huấn luyện mạng khác nhau nhưng giữa chúng vẫn có một vài điểm chung quan trọng như:

- Khả năng của mạng nơ-ron nhân tạo không do lập trình sẵn mà đạt được từ quá trình huấn luyện mạng. Huấn luyện mạng là quá trình hiệu chỉnh các trọng số kết nối giữa các nơ-ron trong mạng sao cho mạng có thể đáp ứng chính xác với các mẫu dữ liệu ngõ vào.

- Có hai kỹ thuật học được sử dụng chủ yếu là học giám sát và học không giám sát. Kỹ thuật học giám sát yêu cầu cần phải được cung cấp các cặp mẫu dữ liệu vào-ra.

- Một mạng sau khi được huấn luyện thành công có khả năng tổng quát hóa (đặc trưng hóa) cao: nó có thể đáp ứng chính xác với ngõ vào mà nó chưa từng gặp trước đó.

- Thời gian huấn luyện mạng còn tùy thuộc vào kiến trúc, thuật toán học và chức năng của mạng. Thông thường, việc huấn luyện mạng nơ-ron đòi hỏi nhiều thời gian.

- Khả năng của mạng tùy thuộc vào số lượng nơ-ron dùng trong mạng. Một cách tổng quát, mạng có càng nhiều nơ-ron thì khả năng của mạng càng cao (nhưng thời gian huấn luyện mạng là rất lớn) và ngược lại.

1.4. Thuật toán di truyền (GA)

1.4.1 Giới thiệu

Thuật toán di truyền là thuật toán tối ưu ngẫu nhiên dựa trên cơ chế chọn lọc tự nhiên và tiến hóa di truyền. Nguyên lý cơ bản của thuật toán di truyền đã được Holland giới thiệu vào năm 1962. Cơ sở toán học đã được phát triển từ cuối những năm 1960 và đã được giới thiệu trong quyển sách đầu tiên của Holland, *Adaptive in Natural and Artificial Systems*. Thuật toán di truyền được ứng dụng đầu tiên trong hai lĩnh vực chính: **tối ưu hóa và học tập của máy**. Trong lĩnh vực tối ưu hóa thuật toán di truyền được phát triển nhanh chóng và ứng dụng trong nhiều lĩnh vực khác nhau như tối ưu hàm, xử lý ảnh, bài toán hành trình người bán hàng, nhận dạng hệ thống và điều khiển.

Thuật toán di truyền cũng như các thuật toán tiến hóa nói chung, hình thành dựa trên quan niệm cho rằng, quá trình tiến hóa tự nhiên là quá trình hoàn hảo nhất, hợp lý nhất và tự nó đã mang tính tối ưu. Quan niệm này có thể xem như một tiên đề đúng, không chứng minh được, nhưng phù hợp với thực tế khách quan. Quá trình tiến hóa thể hiện tính tối ưu ở chỗ, thế hệ sau bao giờ cũng tốt hơn (phát triển hơn, hoàn thiện hơn) thế hệ trước bởi tính kế thừa và đấu tranh sinh tồn.

• Các tính chất đặc thù của thuật toán di truyền

- GA lập luận mang tính chất ngẫu nhiên (stochastic) thay vì xác định (deterministic) như toán học giải tích.
- GA xét duyệt toàn bộ các giải pháp, sau đó lựa chọn giải pháp tốt nhất dựa trên hệ số thích nghi.
- GA chỉ tập trung vào giải pháp (dãy số tượng trưng cho giải pháp) mà không cần quan tâm đến chi tiết vấn đề.
- GA thích hợp cho việc tìm điều kiện tối ưu cho việc điều hành và phân nhóm những giải pháp có được.

1.4.2 Các phép toán của thuật toán di truyền

1. Tái sinh (Reproduction)

Tái sinh là quá trình chọn quần thể mới thỏa phân bố xác suất dựa trên độ thích nghi. Độ thích nghi là một hàm gán một giá trị thực cho cá thể trong quần thể. Các cá thể có độ thích nghi lớn sẽ có nhiều bản sao trong thế hệ mới. Hàm thích nghi có thể không tuyến tính, không đạo hàm, không liên tục bởi vì thuật toán di truyền chỉ cần liên kết hàm thích nghi với các chuỗi số.

Quá trình này được thực hiện dựa trên bánh xe quay roulette (bánh xe số xổ) với các rãnh được định kích thước theo độ thích nghi. Kỹ thuật này được gọi là lựa chọn cha mẹ theo bánh xe roulette. Bánh xe roulette được xây dựng như sau (giả định rằng, các độ thích nghi đều dương, trong trường hợp ngược lại thì ta có thể dùng một vài phép biến đổi tương ứng để định lại tỷ lệ sao cho các độ thích nghi đều dương).

- Tính độ thích nghi f_i , $i=1 \div n$ của mỗi nhiễm sắc thể trong quần thể hiện hành, với n là kích thước của quần thể (số nhiễm sắc thể trong quần thể).
- Tìm tổng giá trị thích nghi toàn quần thể: $F = \sum_{i=1}^n f_i$
- Tính xác suất chọn p_i cho mỗi nhiễm sắc thể: $p_i = \frac{f_i}{F}$
- Tính vị trí xác suất q_i của mỗi nhiễm sắc thể: $q_i = \sum_{j=1}^i p_j$

Tiến trình chọn lọc được thực hiện bằng cách quay bánh xe roulette n lần, mỗi lần chọn một nhiễm sắc thể từ quần thể hiện hành vào quần thể mới theo cách sau:

- Phát sinh ngẫu nhiên một số r trong khoảng $[0 \div 1]$ khi quay bánh xe.
- Nếu $r < q_1$ thì chọn nhiễm sắc thể đầu tiên; ngược lại thì chọn nhiễm sắc thể thứ i sao cho $q_{i-1} < r \leq q_i$

Ví dụ 1.11: Xét bài toán dân số có 6 nhiễm sắc thể với chuỗi mã hóa và hệ số thích nghi tương ứng cho trong bảng 1.

Bảng 1: Các nhiễm sắc thể và các giá trị thích nghi

<i>Nhiễm sắc thể</i>	Chuỗi mã hóa	Trị thích nghi $f(i)$
1	01110	8
2	11000	15
3	00100	2
4	10010	5
5	01100	12
6	00011	8

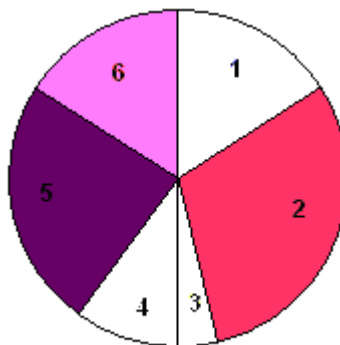
- Giá trị tổng thích nghi toàn quần thể là : $F = \sum_{i=1}^6 f_i = 50$
- Tính xác suất chọn p_i cho mỗi nhiễm sắc thể: $p_i = \frac{f_i}{F}$
- Tính vị trí xác suất q_i của mỗi nhiễm sắc thể: $q_i = \sum_{j=1}^i p_j$

Kết quả thu được điền vào bảng 2:

Bảng 2: Kết quả tính toán cho các nhiễm sắc thể

<i>Nhiễm sắc thể</i>	Chuỗi mã hóa	Trị thích nghi $f(i)$	Xác suất chọn p_i	Vị trí xác suất q_i
1	01110	8	0.16	0.16
2	11000	15	0.3	0.46
3	00100	2	0.04	0.5
4	10010	5	0.1	0.6
5	01100	12	0.24	0.84
6	00011	8	0.16	1

Bây giờ ta quay bánh xe roulette 6 lần, mỗi lần chọn một nhiễm sắc thể cho quần thể mới. Giá trị ngẫu nhiên của 6 số trong khoảng $[0;1]$ và các nhiễm sắc thể tương ứng được chọn được cho trong bảng 2.



Bánh xe roulette

Bảng 2: Quần thể mới

Số lần quay	1	2	3	4	5	6
Số ngẫu nhiên	0.55	0.1	0.95	0.4	0.8	0.7
Nhiễm sắc thể	4	1	6	2	5	5

Ta thấy:

Lần 1: $r_1 = 0.55 > q_1 = 0.16$ và $q_3 < r < q_4$ nên ta chọn NST 4

Lần 2 : $r_2 = 0.1 < q_1 = 0.16$ nên ta chọn NST 1

Tương tự cho các lần quay còn lại, ta có sẽ được một quần thể mới.

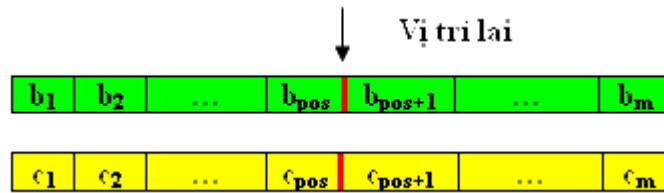
Ta thấy, có thể sẽ có một số nhiễm sắc thể được chọn nhiều lần, các nhiễm sắc thể có độ thích nghi cao hơn sẽ có nhiều bản sao hơn, các nhiễm sắc thể có độ thích nghi kém nhất thì dần dần chết đi.

Sau khi lựa chọn được quần thể mới, bước tiếp theo trong thuật toán di truyền là thực hiện các phép toán lai ghép và đột biến.

2. Lai ghép (Crossover)

Phép lai là quá trình hình thành nhiễm sắc thể mới trên cơ sở các nhiễm sắc thể cha - mẹ, bằng cách ghép một hay nhiều đoạn gen của hai (hay nhiều) nhiễm sắc thể cha - mẹ với nhau. Phép lai xảy ra với xác suất p_c , được thực hiện như sau:

- Đối với mỗi nhiễm sắc thể trong quần thể mới, phát sinh ngẫu nhiên một số r trong khoảng $[0 \div 1]$, nếu $r < p_c$ thì nhiễm sắc thể đó được chọn để lai ghép.
- Ghép đôi các nhiễm sắc thể đã chọn được một cách ngẫu nhiên, đối với mỗi cặp nhiễm sắc thể được ghép đôi, ta phát sinh ngẫu nhiên một số nguyên pos trong khoảng $[0 \div m-1]$ (m là tổng chiều dài của một nhiễm sắc thể - tổng số gen). Số pos cho biết vị trí của điểm lai. Điều này được minh họa như sau:



- Chuyển đổi các gen nằm sau vị trí lai.



Như vậy phép lai này tạo ra hai chuỗi mới, mỗi chuỗi đều được thừa hưởng những đặc tính lấy từ cha và mẹ của chúng. Mặc dù phép lai ghép sử dụng lựa chọn ngẫu nhiên, nhưng nó không được xem như là một lối đi ngẫu nhiên qua không gian tìm kiếm. Sự kết hợp giữa tái sinh và lai ghép làm cho thuật toán di truyền hướng việc tìm kiếm đến những vùng tốt hơn.

3. Đột biến (Mutation)

Đột biến là hiện tượng cá thể con mang một (số) tính trạng không có trong mã di truyền của cha mẹ. Phép đột biến xảy ra với xác suất p_m , nhỏ hơn rất nhiều so với xác suất lai p_c .

Theo nghiên cứu của Kenneth De Jong thì:

- ✓ Tỷ lệ lai ghép trung bình : 0.6
- ✓ Tỷ lệ đột biến : 0.001
- ✓ Tỷ lệ tạo sinh : 0.399

Mỗi gen trong tất cả các nhiễm sắc thể có cơ hội bị đột biến như nhau, nghĩa là đối với mỗi nhiễm sắc thể trong quần thể hiện hành (sau khi lai) và đối với mỗi gen trong nhiễm sắc thể, quá trình đột biến được thực hiện như sau:

- Phát sinh ngẫu nhiên một số r trong khoảng $[0 \div 1]$
- Nếu $r < p_m$, thì đột biến gen đó.

Đột biến làm tăng khả năng tìm được lời giải gần tối ưu của thuật toán di truyền. Đột biến không được sử dụng thường xuyên vì nó là phép toán tìm

kiếm ngẫu nhiên, với tỷ lệ đột biến cao, thuật toán di truyền sẽ còn xấu hơn phương pháp tìm kiếm ngẫu nhiên.

Sau quá trình tái sinh, lai và đột biến, quần thể mới tiếp tục được tính toán các giá trị thích nghi, sự tính toán này được dùng để xây dựng phân bố xác suất (cho tiến trình tái sinh tiếp theo), nghĩa là, để xây dựng lại bánh xe roulette với các rãnh được định kích thước theo các giá trị thích nghi hiện hành. Phần còn lại của thuật toán di truyền chỉ là sự lặp lại chu trình của những bước trên.

• **Hàm thích nghi (Fitness)**

Ta xây dựng hàm thích nghi $f(x)$ nhận giá trị không âm. Có 2 trường hợp:

- Đối với bài toán tìm cực tiểu hàm $g(x)$

$$f(x) = \begin{cases} C_{\max} - g(x) & \text{khi } g(x) < C_{\max} \\ 0 & \text{khi } g(x) \geq C_{\max} \end{cases}$$

Có thể lấy C_{\max} là giá trị g lớn nhất trong quần thể hiện tại.

- Đối với bài toán tìm cực đại hàm $g(x)$

$$f(x) = \begin{cases} u(x) + C_{\min} & \text{khi } u(x) + C_{\min} > 0 \\ 0 & \text{khi } u(x) + C_{\min} \leq 0 \end{cases}$$

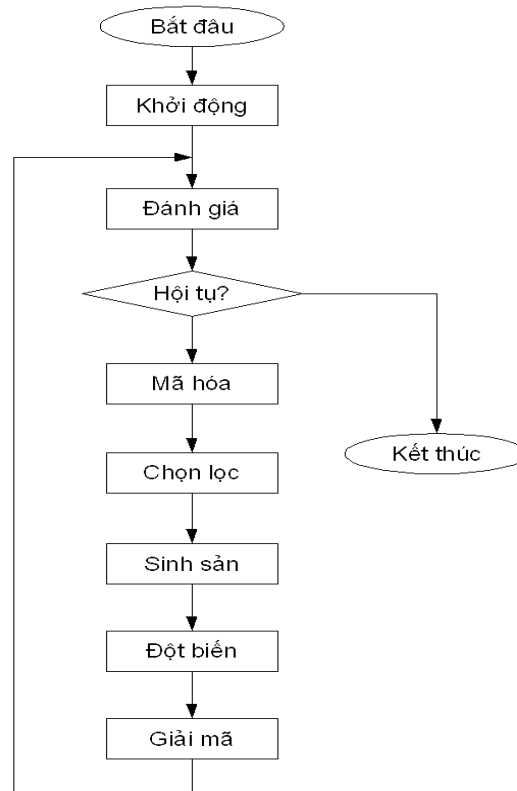
Có thể lấy C_{\min} là trị tuyệt đối của u bé nhất trong quần thể hiện tại.

1.4.3 Cấu trúc của thuật toán di truyền tổng quát

Thuật toán di truyền bao gồm các bước sau:

- Bước 1: Khởi tạo quần thể các nhiễm sắc thể. Chọn mô hình cho giải pháp của vấn đề. Chỉ định cho mỗi giải pháp một ký hiệu.
- Bước 2: Tìm hàm thích nghi và xác định giá trị thích nghi của từng nhiễm sắc thể.
- Bước 3: Sao chép lại các nhiễm sắc thể dựa vào giá trị thích nghi của chúng (tạo sinh) và tạo ra những nhiễm sắc thể mới bằng các phép toán di truyền (lai ghép hay đột biến).
- Bước 4: Tính hệ số thích nghi cho các thành viên mới để loại bỏ những thành viên không phù hợp trong quần thể.

Bước 5: Nếu chưa tìm được giải pháp tối ưu thì trở lại bước 3. Nếu mục tiêu tìm kiếm đã đạt được thì dừng lại. Báo cáo kết quả.



1.4.4 Ví dụ minh họa:

Ví dụ 1.12: Tìm đáp số cho phương trình $X^2 = 64$. Đây là một bài toán đơn giản để giúp ta có thể hiểu rõ hơn các bước của thuật toán di truyền.

Giải bài toán di truyền theo các bước sau:

- **Bước 1:** Chúng ta sử dụng hệ nhị phân để xây dựng mô hình bài toán. Ta dùng 4 bit nhị phân để mã hóa cho các đáp số của bài toán. Giả sử ta không biết đáp số của bài toán, ta sẽ chọn 4 số trong các đáp số có thể có và ký hiệu cho các đáp số đó.

Bảng chọn lựa:

Thứ tự	Nhi phân	Thập phân
1	0 0100	4
2	1 0101	21
3	0 1010	10
4	1 1000	24

➤ **Bước 2:** Tìm hàm số thích nghi và tính hệ số thích nghi cho từng đáp số. Ta chọn hàm số thích nghi sau: $f(X) = 1000 - (X^2 - 64)$
 Vậy, đáp số nào có hệ số thích nghi f gần bằng 1000 nhất thì đó là đáp số.

Khảo sát kết quả tính được:

Thứ tự	Nhi phân	Thập phân (X)	$X^2 - 64$	Hệ số thích nghi $f(x)$
1	0 0100	4	- 48	952
2	1 0101	21	377	623
3	0 1010	10	36	964
4	1 1000	24	512	488

➤ **Bước 3:** ta thấy, hệ số thích nghi của các đáp số vẫn còn cách xa 1000. Do đó, cần tạo ra các đáp số mới bằng cách biến hóa các đáp số cũ. Ta thấy số 4 và 10 có hệ số thích nghi cao hơn nên được chọn để tạo sinh và biến hóa. Đồng thời số 21 và 24 có hệ số thích nghi thấp sẽ bị loại. Giả sử ta lai ghép hai số 4 và 10 theo hình sau:

$$\begin{array}{ccc} 001 | 00 (4) & \begin{array}{c} \nearrow \\ \searrow \end{array} & 010 | 00 (8) \\ 010 | 10 (10) & \begin{array}{c} \searrow \\ \nearrow \end{array} & 001 | 10 (6) \end{array}$$

➤ **Bước 4:** Tính hệ số thích nghi cho quần thể mới

Thứ tự	Nhi phân	Thập phân (X)	$X^2 - 64$	Hệ số thích nghi $f(x)$
1	0 0100	4	- 48	952
2	0 1010	10	36	964
3	0 1000	8	0	1000
4	0 0110	6	28	968

- **Bước 5:** May mắn chúng ta đã tìm được kết quả là $X = 8$ với hệ số thích nghi cao nhất là 1000.

Vậy kết quả của bài toán là $X = 8$.

Ví dụ 1.13:

Tìm giá trị cực đại của hàm sau:

$$f(x_1, x_2) = 21.5 + x_1 \times \sin(4\pi x_1) + x_2 \times \sin(20\pi x_2)$$

$$\text{Với } -3.0 \leq x_1 \leq 12.1 \text{ và } 4.1 \leq x_2 \leq 5.8$$

Giả sử kích thước quần thể là 20, các xác suất di truyền tương ứng là $p_c = 0.25$ và $p_m = 0.01$

Giải:

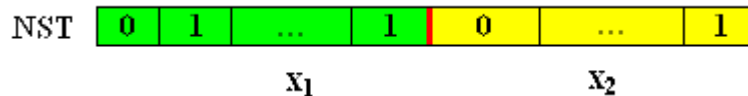
- **Bước 1:** Chúng ta sử dụng hệ nhị phân để xây dựng mô hình bài toán

Giả sử ta cần chính xác đến 4 số lẻ thì số khoảng mà mỗi biến cần là:

$$\begin{aligned} \text{Biến } x_1: & \quad \{12.1 - (-3.0)\} \times 10000 = 151000 \text{ khoảng} \\ & \Rightarrow \text{số bit cần dùng để mã hóa cho biến } x_1 \text{ là 18 bit} \end{aligned}$$

$$\begin{aligned} \text{Biến } x_2: & \quad \{5.8 - (4.1)\} \times 10000 = 17000 \text{ khoảng} \\ & \Rightarrow \text{số bit cần dùng để mã hóa cho biến } x_1 \text{ là 15 bit} \end{aligned}$$

Vậy chiều dài của một nhiễm sắc thể là $m = 15 + 18 = 33$ bit



Ký hiệu NST thứ i là $v_i = (x_1, x_2)$ với $i = \overline{1,20}$

Tiếp theo ta tiến hành khởi tạo quần thể một cách ngẫu nhiên:

$$\begin{aligned} v_1 &= 100110100000001111 \mathbf{111010011011111} \\ v_2 &= 111000100100110111 \mathbf{001010100011010} \\ v_3 &= 000010000011001000 \mathbf{001010111011101} \\ v_4 &= 100011000101101001 \mathbf{111000001110010} \\ v_5 &= 000111011001010011 \mathbf{01011111000101} \\ v_6 &= 000101000010010101 \mathbf{001010111111011} \\ v_7 &= 001000100000110101 \mathbf{111011011111011} \end{aligned}$$

$v_8 = 100001100001110100 \mathbf{010110101100111}$
 $v_9 = 100000000101100010 \mathbf{110000001111100}$
 $v_{10} = 000001111000110000 \mathbf{011010000111011}$
 $v_{11} = 011001111110110101 \mathbf{100001101111000}$
 $v_{12} = 110100010111101101 \mathbf{000101010000000}$
 $v_{13} = 111011111010001000 \mathbf{110000001000110}$
 $v_{14} = 010010011000001010 \mathbf{100111100101001}$
 $v_{15} = 111011101101110000 \mathbf{100011111011110}$
 $v_{16} = 110011110000011111 \mathbf{100001101001011}$
 $v_{17} = 011010111111001111 \mathbf{010001101111101}$
 $v_{18} = 011101000000001110 \mathbf{100111110101101}$
 $v_{19} = 000101010011111111 \mathbf{110000110001100}$
 $v_{20} = 101110010110011110 \mathbf{011000101111110}$

➤ **Bước 2:** Giải mã và tính hệ số thích nghi (lấy NST thứ 1 làm ví dụ)

Mã hóa NST:

$$v_1 = 100110100000001111 \mathbf{111010011011111}$$

$$x_1 = -3.0 + \text{dec}(100110100000001111) \times \frac{12.1 - (-3.0)}{2^{18} - 1} = 6.0845$$

$$x_2 = 4.1 + \text{dec}(\mathbf{111010011011111}) \times \frac{5.8 - 4.1}{2^{15} - 1} = 5.6522$$

Tính trị thích nghi:

$$f(x_1, x_2) = f(6.0845 ; 5.6522) = 26.0196$$

Ta lần lượt tính giá trị thích nghi của các NST còn lại.

Giá trị tổng thích nghi toàn quần thể: $F = \sum_{i=1}^{20} f_i = 387.7768$

Xác suất chọn p_1 : $p_1 = \frac{f_1}{F} = \frac{26.0196}{387.7768} = 0.067$

Vị trí xác suất q_1 : $q_i = \sum_{j=1}^i p_j \Rightarrow q_1 = \sum_{j=1}^1 p_j = p_1 = 0.067$

Ta lần lượt tính giá trị của các NST còn lại. Ghi kết quả vào bảng.

Bảng kết quả thu được:

NST	X_1	X_2	Trị thích nghi $f(x_1, x_2)$	Xác suất chọn p_i	Vị trí xác suất q_i
1	6.0845	5.6522	26.0196	0.067	0.067
2	10.3484	4.3802	7.5800	0.019	0.086
3	-2.5166	4.3904	19.6263	0.050	0.137
4	5.2786	5.5934	17.4067	0.044	0.181
5	-1.2551	4.7345	25.3411	0.065	0.247
6	-1.8117	4.3919	18.1004	0.046	0.293
7	-0.9915	5.6803	16.0208	0.041	0.335
8	4.9106	4.7030	17.9597	0.046	0.381
9	0.7954	5.3814	16.1278	0.041	0.423
10	-2.5548	4.7937	21.2784	0.054	0.478
11	3.1300	4.9961	23.4107	0.060	0.538
12	9.3561	4.2395	15.0116	0.038	0.577
13	11.1346	5.3786	27.3167	0.070	0.647
14	1.3359	5.1513	19.8762	0.051	0.698
15	11.0890	5.0545	30.0602	0.077	0.776
16	9.2116	4.9937	23.9672	0.061	0.837
17	3.3675	4.5713	13.6961	0.035	0.873
18	3.8430	5.1582	15.4141	0.039	0.812
19	-1.7466	5.3955	20.0959	0.051	0.964
20	7.9359	4.7573	13.6669	0.035	1.000

➤ **Tạo sinh: Xây dựng hệ thống bánh xe rulét cho tiến trình chọn lọc.**

Quay bánh xe rulét 20 lần, phát sinh ngẫu nhiên 20 số (r) trong khoảng [0 1]
 Mỗi lần quay chọn 1 NST mới (xem lại cách chọn NST ở phần trên)

Lần quay	1	2	3	4	5	6
Số ngẫu nhiên	0.514	0.175	0.308	0.534	0.947	0.171
NST được chọn	v_{11}	v_4	v_7	v_{11}	v_{19}	v_4

Lần	7	8	9	10	11	12	13
r	0.702	0.226	0.494	0.424	0.704	0.389	0.277
NST	v_{15}	v_5	v_{11}	v_3	v_{15}	v_9	v_6

Lần	14	15	16	17	18	19	20
r	0.368	0.983	0.005	0.765	0.646	0.767	0.780
NST	v_8	v_{20}	v_1	v_{10}	v_{13}	v_{15}	v_{16}

Như vậy, quần thể mới được chọn là:

$v'_1 = 011001111110110101$ **100001101111000**
 $v'_2 = 100011000101101001$ **111000001110010**
 $v'_3 = 001000100000110101$ **111011011111011**
 $v'_4 = 011001111110110101$ **100001101111000**
 $v'_5 = 000101010011111111$ **110000110001100**
 $v'_6 = 100011000101101001$ **111000001110010**
 $v'_7 = 111011101101110000$ **100011111011110**
 $v'_8 = 000111011001010011$ **010111111000101**
 $v'_9 = 011001111110110101$ **100001101111000**
 $v'_{10} = 000010000011001000$ **001010111011101**
 $v'_{11} = 111011101101110000$ **100011111011110**
 $v'_{12} = 100000000101100010$ **110000001111100**
 $v'_{13} = 000101000010010101$ **001010111111011**
 $v'_{14} = 100001100001110100$ **010110101100111**
 $v'_{15} = 101110010110011110$ **011000101111110**
 $v'_{16} = 100110100000001111$ **111010011011111**
 $v'_{17} = 000001111000110000$ **011010000111011**
 $v'_{18} = 111011111010001000$ **110000001000110**
 $v'_{19} = 111011101101110000$ **100011111011110**
 $v'_{20} = 110011110000011111$ **100001101001011**

➤ **Quá trình lai ghép:** xác suất lai $p_c = 0.25$

Đối với mỗi NST trong quần thể mới phát sinh ngẫu nhiên một số r trong khoảng $[0, 1]$. Nếu $r < 0.25$ thì ta chọn một NST cho trước để lai tạo.

Giả sử thứ tự của các số ngẫu nhiên là:

0.822	0.151	0.625	0.314	0.346
0.917	0.519	0.401	0.606	0.785
0.031	0.869	0.166	0.574	0.758
0.581	0.389	0.200	0.355	0.826

Ta thấy các NST $v'_2, v'_{11}, v'_{13}, v'_{18}$ được chọn để lai tạo. Ở đây, chúng ta có số NST cần được lai tạo là một số chẵn nên ta có thể bắt cặp dễ dàng.

Trường hợp tổng số NST cần lai tạo là số lẻ thì ta có thể cộng thêm một NST ngoại hoặc lấy bớt đi một NST một cách ngẫu nhiên.

Ta giả sử kết hợp (v'_2, v'_{11}) và (v'_{13}, v'_{18}) . Đối với mỗi cặp NST sẽ phát sinh một số nguyên ngẫu nhiên pos trong khoảng $[1, 32]$.

Số pos cho biết vị trí của điểm lai tạo.

- Cặp NST (v'_2, v'_{11}): giả sử pos = 9

$$\begin{aligned} v'_2 &= 100011000 \mathbf{1} 101101001 \mathbf{111000001110010} \\ v'_{11} &= 111011101 \mathbf{1} 101110000 \mathbf{100011111011110} \end{aligned}$$

$$\Rightarrow \begin{aligned} v''_2 &= 100011000 \ 101110000 \ \mathbf{100011111011110} \\ v''_{11} &= 111011101 \ 101101001 \ \mathbf{111000001110010} \end{aligned}$$

- Cặp NST (v'_{13}, v'_{18}): giả sử pos = 20

$$\begin{aligned} v'_{13} &= 000101000010010101 \mathbf{00} \mathbf{1010111111011} \\ v'_{18} &= 111011111010001000 \mathbf{11} \mathbf{0000001000110} \end{aligned}$$

$$\Rightarrow \begin{aligned} v''_{13} &= 000101000010010101 \ \mathbf{11} \ \mathbf{0000001000110} \\ v''_{18} &= 111011111010001000 \ \mathbf{00} \ \mathbf{1010111111011} \end{aligned}$$

- **Phép toán đột biến:** xác suất đột biến $p_m = 0.01$

Toàn bộ quần thể có: $33 \times 20 = 660$ bit

Như vậy, trung bình có 6.6 đột biến trong mỗi thế hệ. Vì mỗi bit đều có cơ hội đột biến ngang nhau nên ta phát sinh ngẫu nhiên số r trong khoảng $[0, 1]$. Nếu bit nào $r < 0.01$ thì xảy ra đột biến ở bit đó. Giả sử có 5 vị trí đột biến:

Vị trí bit	Số ngẫu nhiên	Nhiễm sắc thể thứ	Vị trí bit trong NST
112	0.0002	4	13
349	0.0099	11	19
418	0.0088	13	22
429	0.0054	13	33
603	0.0028	19	9

Ta thấy có 4 NST bị đột biến. Trong đó, NST thứ 13 bị đột biến ở 2 vị trí. Sau khi đột biến, quần thể hiện hành là:

$$\begin{aligned} v_1 &= 011001111110110101 \ \mathbf{100001101111000} \\ v_2 &= 100011000101110000 \ \mathbf{100011111011110} \\ v_3 &= 001000100000110101 \ \mathbf{111011011111011} \\ v_4 &= 011001111110010101 \ \mathbf{100001101111000} \\ v_5 &= 000101010011111111 \ \mathbf{110000110001100} \\ v_6 &= 100011000101101001 \ \mathbf{111000001110010} \\ v_7 &= 111011101101110000 \ \mathbf{100011111011110} \\ v_8 &= 000111011001010011 \ \mathbf{010111111000101} \\ v_9 &= 011001111110110101 \ \mathbf{100001101111000} \end{aligned}$$

$v_{10} = 000010000011001000 \mathbf{001010111011101}$
 $v_{11} = 111011101101101001 \mathbf{011000001110010}$
 $v_{12} = 100000000101100010 \mathbf{110000001111100}$
 $v_{13} = 000101000010010101 \mathbf{110100001000111}$
 $v_{14} = 100001100001110100 \mathbf{010110101100111}$
 $v_{15} = 101110010110011110 \mathbf{011000101111110}$
 $v_{16} = 100110100000001111 \mathbf{111010011011111}$
 $v_{17} = 000001111000110000 \mathbf{011010000111011}$
 $v_{18} = 111011111010001000 \mathbf{001010111111011}$
 $v_{19} = 111011100101110000 \mathbf{100011111011110}$
 $v_{20} = 110011110000011111 \mathbf{100001101001011}$

Như vậy ta đã hoàn thành xong một vòng lặp (nghĩa là một thế hệ).

Ta tính toán tương tự cho vòng lặp kế. Kết quả thu được từ quần thể mới là:

Tổng độ thích nghi: $F = 447.049$ cao hơn so với quần thể trước (387.77).

NST tốt nhất v_{11} có độ thích nghi là 33.35 tốt hơn NST tốt nhất v_{11} của quần thể trước là 30.06.

➤ **Nhận xét:**

Cứ tiếp tục như thế, cá thể “trội hơn cả” sẽ được lưu trữ tại một vị trí riêng biệt. bằng cách này, thuật giải có thể duy trì được cá thể tốt nhất tìm được trong suốt quá trình (không có nghĩa là tốt nhất trong quần thể cuối cùng).

1.4.5. Kết luận:

Thuật toán di truyền đã chứng tỏ tính hữu ích của nó khi được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau của cuộc sống.

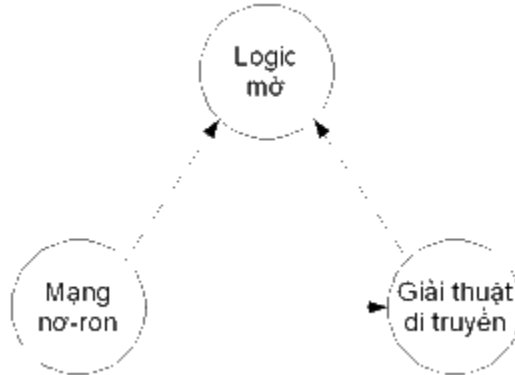
Trong lĩnh vực điều khiển tự động, thuật toán di truyền có thể được sử dụng để xác định thông số tối ưu cho các bộ điều khiển. Thông số bộ điều khiển được mã hóa thành các nhiễm sắc thể, thông qua mô phỏng, các nhiễm sắc thể này được đánh giá và lựa chọn thông qua mức độ thích nghi của chúng (cũng chính là các chỉ tiêu chất lượng của hệ thống). Kết quả của thuật toán sẽ cho một bộ điều khiển có thông số tốt nhất.

Trong y học, cấu trúc của các chất hóa học được mã hóa thành các nhiễm sắc thể hoặc đồ thị. Thuật toán di truyền sẽ lai ghép, lựa chọn để tạo ra các nhiễm sắc thể mới (các chất hóa học mới). Và trong thực tế đã có rất nhiều loại thuốc mới được tạo ra như vậy.

1.5. Sự kết hợp các thành phần của tính toán mềm

Ba thành phần chính của tính toán mềm có thể sử dụng hoàn toàn độc lập với nhau, tuy nhiên thực tế đã cho thấy việc kết hợp các thành phần này với nhau sẽ làm tăng đáng kể chất lượng của thuật toán.

Có 3 kiểu kết hợp tương đối phổ biến :



Sự kết hợp các thành phần của tính toán mềm

1.5.1 Kết hợp mạng nơ-ron nhân tạo và logic mờ (neuro-fuzzy)

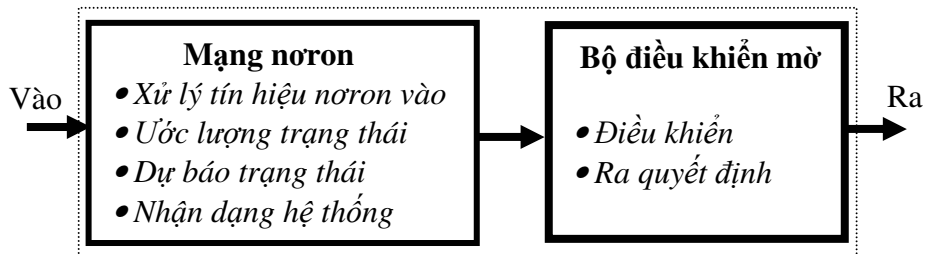
Những ưu nhược điểm của mạng nơ-ron và điều khiển mờ:

Tính chất	Mạng Nơ-ron	Bộ điều khiển mờ
Thể hiện tri thức	Thông qua trọng số được thể hiện ẩn trong mạng	Được thể hiện ngay tại luật hợp thành
Nguồn của tri thức	Từ các mẫu học	Từ kinh nghiệm chuyên gia
Xử lý thông tin không chắc chắn	Định lượng	Định lượng và định tính
Lưu giữ tri thức	Trong nơ-ron và trọng số của từng đường ghép nối nơ-ron	Trong luật hợp thành và hàm thuộc
Khả năng cập nhật và nâng cao kiến thức	Thông qua quá trình học	Không có
Tính nhạy cảm với những thay đổi của mô hình	Thấp	Cao

Mạng nơ-ron có khuyết điểm lớn là tính chất “hộp đen” của nó. Dữ liệu được đưa vào mạng, quá trình tính toán xảy ra và cho kết quả ở ngõ ra. Tuy vậy, ta không biết một cách chính xác chuyện gì đã xảy ra.

Logic mờ thì ngược lại, các tập luật của chúng rất gần gũi với tri thức con người. Tuy nhiên, logic mờ lại không có được khả năng tự học của mạng nơ-ron. Các luật mờ và hàm liên thuộc chỉ có thể được điều chỉnh bằng tay.

Vì thế, sự kết hợp mạng nơron và điều khiển mờ đã vô hình chung xóa bỏ những khuyết điểm của chúng, tạo ra bộ điều khiển mờ - nơron có ưu điểm vượt trội.



Kiến trúc kiểu mẫu của một hệ mờ-nơron

Hệ thống nơ-ron mờ sử dụng thuật toán học của mạng nơ-ron nhiều lớp để chỉnh định thông số và cấu trúc của bộ mờ. Điều này có nghĩa là bộ mờ có thể tự xây dựng chính nó (không cần kiến thức chuyên gia về đối tượng mà nó mô hình hóa) khi được cung cấp những tập mẫu dữ liệu giàu thông tin. Cấu trúc xây dựng xong có thể dễ dàng “biên dịch” sang ngôn ngữ luật gần gũi với con người.

Đây là sự kết hợp đầy hiệu quả trong kỹ thuật tính toán mềm

1.5.2. Kết hợp giải thuật di truyền và logic mờ (genetic fuzzy)

Khi hệ thống quá phức tạp (hoặc giá trị ngõ vào hoặc ngõ ra không đo lường được), ta không thể xây dựng cấu trúc bộ mờ bằng kiến thức chuyên gia cũng như không thể huấn luyện nó bằng mạng nơ-ron. Nếu như, việc đánh giá chất lượng của hệ thống là không quá khó và có thể xem như một hàm thích nghi thì ta có thể sử dụng giải thuật di truyền để chỉnh định bộ mờ.

Các hệ quy tắc và hàm liên thuộc của bộ mờ được mã hóa thành các chuỗi nhiễm sắc thể. Giải thuật di truyền được dùng để tạo ra lời giải tối ưu. Đó chính là các thông số và tập luật làm cho chất lượng của hệ thống tối ưu.

Một trong những ứng dụng thành công là việc phân tích nguyên nhân làm xuất hiện các vết nứt trong các công trình xây dựng bằng đá. Đây là một lĩnh vực đòi hỏi rất nhiều kiến thức chuyên gia và phụ thuộc vào nhiều đặc tính khác nhau. Hệ thống mờ được dùng có ngõ vào là 42 đặc tính đo lường được để xác định lý do chính từ 6 khả năng. Do hệ thống có số lượng thông số quá lớn và cũng quá phức tạp, việc xây dựng bộ luật và xác định các hàm liên thuộc cho hệ thống bằng tay là một điều gần không có tính khả thi. Giải thuật di truyền đã được vận dụng và đem lại kết quả tương đối tốt với tỉ lệ sai số nhỏ hơn 4%.

1.5.3. Kết hợp mạng nơ-ron và giải thuật di truyền

➤ Kết hợp mạng nơ-ron với giải thuật di truyền

Giải thuật di truyền đòi hỏi ta phải xác định các hàm thích nghi. Tuy nhiên, đôi khi việc xác định hàm thích nghi hợp lý là một điều không thể. Ví dụ khi dùng giải thuật di truyền để chỉnh định thông số cho bộ điều khiển của một hệ thống phức tạp. Khi đó, hàm thích nghi chính là chất lượng điều khiển của toàn hệ thống. Thông số này chỉ có thể đạt được bằng cách thử nghiệm từng bộ điều khiển (từng cá thể) trên hệ thống thực. Điều này gây tốn kém về mặt thời gian và kinh tế.

Trong những trường hợp như vậy, mạng nơ-ron có thể được dùng để xây dựng hàm thích nghi cho giải thuật di truyền. Thế mạnh của mạng nơ-ron là khả năng nhận dạng do nó có thể xấp xỉ bất kỳ hàm phi tuyến nào. Sau khi nhận dạng được mô hình hệ thống, mạng nơ-ron có thể thay thế việc thử nghiệm trên hệ thống thực bằng việc mô phỏng trên máy tính. Điều này khiến giải thuật di truyền thực thi nhanh hơn do việc mô phỏng trên máy tính được thực hiện hoàn toàn tự động và cho kết quả nhanh chóng.

Một ứng dụng của kỹ thuật này là trong ngành công nghiệp thép. Việc dự báo đặc tính cơ của hợp kim thép khi được cung cấp các thông số về thành phần kim loại và nhiệt độ tôi là một việc không dễ dàng. Các nhà nghiên cứu tại đại học Sheffield đã huấn luyện một mạng nơ-ron dựa trên những thông tin này và đặc tính cơ đo đạt được để xây dựng mô hình dự báo đặc tính hợp kim. Mạng nơ-ron này sau đó được dùng như một hàm thích nghi trong giải thuật di truyền để phát triển các thiết kế hợp kim mới với những đặc tính mong muốn.

➤ **Kết hợp giải thuật di truyền với mạng nơ-ron**

Ví dụ: Ta sẽ xem xét việc ứng dụng GA để huấn luyện mạng nơron truyền thẳng đa lớp nhằm giải quyết bài toán XOR quen thuộc.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

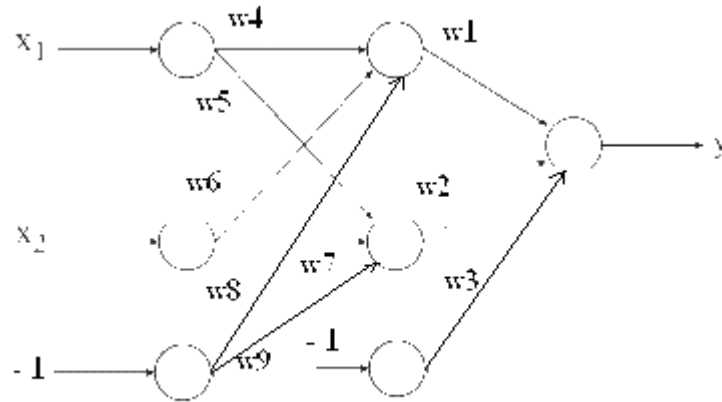
Ở đây, ta sử dụng hàm kích hoạt trong mạng là hàm sigmoid vô cực. Chuỗi nhiễm sắc thể (NST) được mã hóa dưới dạng (w_1, w_2, \dots, w_9) . Trong đó, mỗi phần tử là một trọng số có giá trị thực. Phần tử nhỏ nhất mà quá trình lai ghép và đột biến điều khiển chính là các trọng số. Giả sử, toán tử lai ghép được dùng là lai ghép đều. Ta thực hiện quá trình theo các bước sau:

- Trước tiên, ta chọn ngẫu nhiên n_c vị trí của phần tử trong NST.
- Sau đó thực hiện việc trao đổi lẫn nhau giữa các phần tử tại những vị trí đã chọn ở trên.

Thông thường số lượng các phần tử được trao đổi n_c được chọn bằng một phần hai tổng số các phần tử trong NST.

Toán tử đột biến xảy ra với xác suất bằng 1.0 trên các cá thể chọn lựa khi các cá thể đó không tham gia vào hoạt động lai ghép. Như vậy, nếu xác suất lai ghép của một cặp cá thể là p_c thì xác suất đột biến sẽ là $(1 - p_c)$.

Toán tử đột biến sẽ làm tăng các phần tử thêm một giá trị bằng Δw , giá trị này sẽ được chọn ngẫu nhiên trong khoảng xác định cho trước.



Cấu trúc mạng cho bài toán XOR

Có hai kiểu chọn lọc có thể được sử dụng:

- Chọn ra các cặp để ứng dụng những toán tử di truyền (giống như giải thuật chuẩn hóa.
- Chọn ra những cá thể tồn tại cho lần tạo sinh kế tiếp. Sau khi các toán tử di truyền tạo ra những cá thể mới, những cá thể tồn tại sẽ được chọn từ những cá thể mới được tạo ra và những cá thể hiện tại. Giá trị thích nghi sẽ được tính toán và gán đến tất cả các cá thể. Sau đó, các cá thể dùng cho lần tạo sinh kế sẽ được chọn với xác suất tương ứng với giá trị thích nghi của chúng. Kích cỡ quần thể của những lần tạo sinh sau đó cũng được giữ cố định (bằng một phần hai quần thể thử nghiệm).

Các kỹ thuật điều chỉnh và phân chia cũng được dùng:

$$f' = af + b$$

Trong đó:

f là giá trị thích nghi gốc; f' là giá trị thích nghi đã điều chỉnh

Hệ số a và b được chọn sao cho:

$$f'_{avg} = f_{avg}$$

$$\text{và } f'_{max} = C_{mult} f_{avg}$$

Ở đây, C_{mult} là số lượng con cháu dự tính tạo ra từ cá thể thích nghi nhất (giả sử ta chọn bằng 2).

Bây giờ, ta sẽ ứng dụng GA được mô tả như trên vào mạng nơron (như hình) để giải quyết bài toán XOR.

Giả sử rằng:

- x_1, x_2 và y chỉ nhận hai giá trị là 0.1 hoặc 0.9 tương ứng với TRUE hoặc FALSE. Kích cỡ quần thể là 20.
- Giá trị khởi tạo ban đầu là các số nguyên trong khoảng $[-10, 10]$.
- Xác suất lai ghép $p_c = 0.7$ và số vị trí lai ghép $n_c = 4$.
- Giá trị đột biến Δw là một số nguyên ngẫu nhiên trong miền $[-3, 3]$.

Toán tử đột biến được điều chỉnh sao cho từng trong số không vượt quá miền giới hạn $[-10, 10]$ và bảo quản được kích cỡ của không gian tìm kiếm. Giá trị thích nghi được tính dựa vào hàm lỗi, chính là trị tuyệt đối của tổng các sai lệch ngõ ra trên 4 mẫu XOR.

Sau 30 lần tạo sinh, ta sẽ có một mạng nơron hợp lý với giá trị hàm lỗi nhỏ hơn 0.05 (giá trị thích nghi lớn hơn 20).

1.6. Hệ thống điều khiển tích hợp

Ngành điều khiển học đã ra đời và phát triển từ rất sớm, đặc biệt là trong 2 thập niên gần đây việc ứng dụng Lý thuyết mờ và Mạng nơron đã tạo ra nhiều phương pháp điều khiển mới với đặc tính “*linh hoạt*” và “*thông minh*” hơn. Công nghệ mờ và công nghệ mạng nơron là hai trụ cột chính để tạo nên công nghệ tích hợp mới, công nghệ tính toán mềm (*Soft computing*).

1.6.1. Khái niệm

Một số phương pháp được sử dụng trong ngành điều khiển học:

ĐK Kinh điển & Hiện đại	ĐK Thông minh
PID	GA
Tối ưu	Nơron
Thích nghi	Mờ
Bền vững	...

Mỗi phương pháp đều có những điểm mạnh và hạn chế nhất định, vì vậy người ta thường có xu hướng kết hợp chúng lại với nhau để tạo ra một mô hình điều khiển có khả năng đáp ứng cao với các đòi hỏi thực tế. Việc kết hợp này đã cho ra một phương pháp điều khiển mới đó là điều khiển tích hợp.

Điều khiển tích hợp : *Điều khiển kết hợp phương pháp kinh điển hoặc hiện đại với phương pháp điều khiển thông minh.*

1.6.2. Một số hệ thống tích hợp

- Điều khiển sử dụng PID mờ
- Điều khiển mờ - thích nghi, mờ - tối ưu.
- Sử dụng hệ mờ - nơron để nhận dạng & tối ưu hệ thống.
- Ứng dụng thuật toán GA trong thiết kế hệ thống điều khiển.

.....

Ở phần 1.2.3 ta đã trình bày về cách thiết kế bộ PID mờ, phần 1.2.5 đã nói về việc tích hợp công nghệ mờ trong điều khiển. Ở phần sau chúng ta sẽ thiết kế hệ thống điều khiển neuron mờ hệ con lắc ngược dựa trên phương pháp tiếp cận sai số phân cấp mờ.

Thiết kế hệ thống điều khiển neuron mờ hệ con lắc ngược dựa trên phương pháp tiếp cận sai số phân cấp mờ.

Chúng ta xét mô hình con lắc ngược như ở ví dụ 1.8:

Với giả thiết:

- Khối lượng xe $m_c = 1$ kg
- Khối lượng vật $m = 0.1$ kg
- Chiều dài của vật $2l = 1$ m

Phương trình động của hệ con lắc ngược được cho như (1.36) và (1.37):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (f(x_1, x_2) + g(x_1, x_2)u + d)$$

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Trong đó:

- u là lực tác dụng (tín hiệu điều khiển)
- d là nhiễu ngoài tiền định (giả sử là tín hiệu sóng vuông với biên độ +/- 0.1 và chu kì 2π).
- Ngõ ra của hệ là y và tín hiệu ngõ ra mong muốn là $y_m = (\pi/30) \cdot \sin(t)$

Nhiệm vụ điều khiển của ta là điều khiển trạng thái $y = x$ của hệ thống bám theo quỹ đạo của tín hiệu mong muốn y_m .

1. Bài toán điều khiển thích nghi:

Xem xét hệ phi tuyến một ngõ vào - một ngõ ra (SISO) bậc n có dạng sau:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B(f(\mathbf{x}) + g(\mathbf{x})u + d) \\ y &= C^T \mathbf{x} \end{aligned} \tag{1}$$

Trong đó: A là ma trận dạng thường, $B^T = [0 \ 0 \ \dots \ 0 \ 1]$, $C^T = [1 \ 0 \ \dots \ 0 \ 0]$

$f(\cdot)$ và $g(\cdot)$ là hàm phi tuyến chưa biết của hệ thống.

$u \in \mathbb{R}$ và $y \in \mathbb{R}$ là hàm ngõ vào và ngõ ra của hệ tương ứng

Cho $\mathbf{x} = (x, x^{(1)}, \dots, x^{(n-1)})^T = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ là vectơ trạng thái của hệ và giả thiết là không đo được hoàn toàn, và cho d là nhiễu giới hạn bên ngoài. Để hệ (1) là điều khiển được, chúng ta yêu cầu $g(\mathbf{x}) \neq 0$ và không mất tính tổng quát, giả thiết rằng $|g(\mathbf{x})| > 0$.

Xét một ngõ ra hệ thống y là đo được và tín hiệu mong muốn cho trước y_m . Giả thiết $y_m, y_m^{(1)}, \dots, y_m^{(n-1)}$ là ràng buộc và đo được tương ứng. Mục tiêu điều khiển là tìm kiếm luật điều khiển $u(t)$ để ngõ ra thực $y(t)$ tiệm cận đến

ngõ ra mong muốn $y_m(t)$. Vector sai số bám theo e và vector sai số ước lượng được định nghĩa như sau:

$$\begin{aligned} \mathbf{e} &= \mathbf{y}_m - \mathbf{x}, \\ \hat{\mathbf{e}} &= \mathbf{y}_m - \hat{\mathbf{x}} \end{aligned} \quad (2)$$

Trong đó $y_m = y_m, y_m^{(1)}, \dots, y_m^{(n-1)}$, $\mathbf{e} = (e, e^{(1)}, \dots, e^{(n-1)})$,

$\hat{\mathbf{x}}$ và $\hat{\mathbf{e}}$ ký hiệu ước lượng của \mathbf{x} và \mathbf{e} tương ứng.

Với $\forall \mathbf{x} \in \mathbb{R}^n$, $|g(\mathbf{x})| > 0$, nhiệm vụ của điều khiển là sử dụng ngõ vào và ngõ ra của hệ thống để thiết kế luật cập nhật của bộ điều khiển neuron mờ $u = u(\mathbf{y} | \theta_f, \theta_g)$ và vector thông số θ_f và θ_g . Điều này phải thỏa mãn rằng tất cả các biến có trong hệ thống điều khiển vòng kín là bị chặn và sai số bám theo phải càng nhỏ càng tốt thỏa chỉ tiêu ràng buộc. Theo tiêu chuẩn ổn định Lyapunov và dựa trên tiếp cận tương đương chắc chắn, ta có luật điều khiển là:

$$u = \frac{1}{\hat{g}(\hat{\mathbf{x}})} [-\hat{f}(\hat{\mathbf{x}}) + y_m^{(n)} + \mathbf{K}_c^T \hat{\mathbf{e}} - u_c] \quad (3)$$

Trong đó: hàm $\hat{f}(\hat{\mathbf{x}})$ và $\hat{g}(\hat{\mathbf{x}})$ là ước lượng của hàm phi tuyến $f(x)$ và $g(x)$, $\mathbf{K}^T = (k_n, k_{n-1}, \dots, k_1)$ là vector độ lợi hồi tiếp và u_c là tín hiệu điều khiển bù nhiễu bên ngoài và sai số ước lượng. Ta chọn \mathbf{K}^T để đa thức đặc tính $(A - \mathbf{BK}^T)$ là Hurwitz, nghĩa là cực đặc tính nằm bên trái mặt phẳng phức.

Để đạt được luật thích nghi cần xác định nguyên tắc chỉnh θ_f , và θ_g , ta xét hàm Lyapunov :

$$V = \frac{1}{2} \tilde{\mathbf{e}}^T P \tilde{\mathbf{e}} + \frac{1}{2\gamma_1} \tilde{\theta}_f^T \tilde{\theta}_f + \frac{1}{2\gamma_2} \tilde{\theta}_g^T \tilde{\theta}_g$$

Trong đó: $\tilde{\theta}_f = \theta_f - \theta_f^*$, $\tilde{\theta}_g = \theta_g - \theta_g^*$, γ_1 và γ_2 là các giá trị vô hướng dương và $P = P^T \geq 0$ là ma trận xác định dương thỏa phương trình Lyapunov Đạo hàm theo thời gian của V , ta được:

$$\dot{V} = \frac{1}{2} \dot{\tilde{\mathbf{e}}}^T P \tilde{\mathbf{e}} + \frac{1}{2} \tilde{\mathbf{e}}^T P \dot{\tilde{\mathbf{e}}} + \frac{1}{\gamma_1} \dot{\tilde{\theta}}_f^T \tilde{\theta}_f + \frac{1}{\gamma_2} \dot{\tilde{\theta}}_g^T \tilde{\theta}_g$$

Theo tiêu chuẩn ổn định Lyapunov, $\dot{V} \leq 0$, ta có luật thích nghi $\dot{\theta}_f, \dot{\theta}_g$.

$$\dot{\theta}_f = -\gamma_1 e^T P b \xi(x) \theta$$

$$\dot{\theta}_g = -\gamma_2 e^T P b \eta(x) u_1$$

Chúng ta sẽ nghiên cứu vấn đề này rõ hơn trong phần Điều khiển mờ thích nghi ở chương 3.

2. Cơ sở lý thuyết mạng Neuron mờ:

Cấu hình xấp xỉ hàm neuron mờ được đề nghị như hình 1.26. Mạng gồm có hai phần: mạng tiền đề và mạng kết quả. Mạng tiền đề là mạng neuron truyền thẳng 4 lớp để đáp ứng yêu cầu của luật mờ và tạo ra độ mạnh cháy chuẩn hóa như là ngõ ra để thể hiện cấp độ quan trọng của luật mờ. Mạng neuron kết quả thực hiện việc học và giải mờ ở ngõ ra.

A. Mạng tiền đề:

Mỗi nút lớp 2 (lớp mờ hóa) thực hiện tính toán của hàm thành viên:

$$\mu_{F_i^l}(x_i) = \exp\left(-\left(\frac{x_i - \bar{x}_{fi}^l}{\sigma_{fi}^l}\right)^2\right) \quad \mu_{G_i^l}(x_i) = \exp\left(-\left(\frac{x_i - \bar{x}_{gi}^l}{\sigma_{gi}^l}\right)^2\right)$$

Trong lớp 3 (lớp luật) độ mạnh cháy của luật được cho bởi:

$$\psi_{F_i^l}(x_i) = \prod_{i=1}^n \mu_{F_i^l}(x_i) \quad \psi_{G_i^l}(x_i) = \prod_{i=1}^n \mu_{G_i^l}(x_i)$$

Lớp 4, lớp ngõ ra của mạng tiền đề thực hiện tính toán chuẩn hóa của độ mạnh cháy bởi:

$$\xi_f^l(\mathbf{x}) = \frac{\prod_{i=1}^n \mu_{F_i^l}(x_i)}{\sum_{l=1}^M \left(\prod_{i=1}^n \mu_{F_i^l}(x_i)\right)} = \frac{\psi_{F_i^l}(x_i)}{\sum_{l=1}^M \psi_{F_i^l}(x_i)} \quad \xi_g^l(\mathbf{x}) = \frac{\prod_{i=1}^n \mu_{G_i^l}(x_i)}{\sum_{l=1}^M \left(\prod_{i=1}^n \mu_{G_i^l}(x_i)\right)} = \frac{\psi_{G_i^l}(x_i)}{\sum_{l=1}^M \psi_{G_i^l}(x_i)}$$

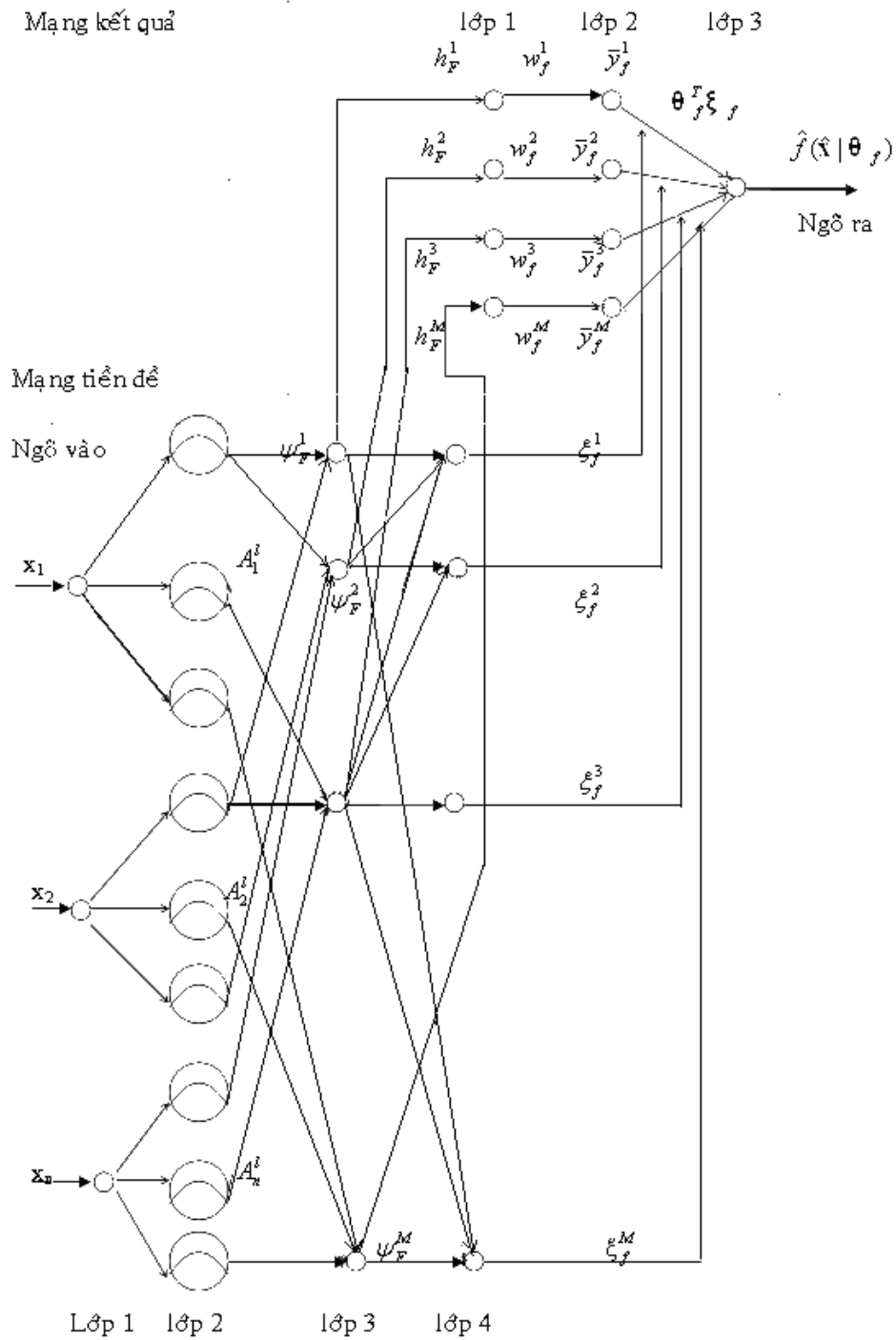
B. Mạng kết quả:

Lớp vào của mạng kết quả lấy hệ số trọng lượng w_f^l để nối với lớp ẩn. Ngõ ra mạng y được tính như là tổng trọng số trung bình của y_f^{-l} thông qua hiệu chỉnh hệ số trọng lượng bởi giải thuật tiếp cận sai số phân cấp mờ (fuzzy hierarchy error approach algorithm - FHEA). Độ mạnh cháy của luật thứ l $\psi_{F_i^l}(x)$ được chuyển như là biến ngõ ra $h_{F_i}^l$ của lớp 1 của mạng kết quả bởi phép tính đơn giản. Lớp 2 thực hiện tính toán giá trị trọng số luật w_f^l , \bar{y}_f^l trong đó kết quả của luật mờ thứ l được định nghĩa là hàm tuyến tính của biến lớp vào $h_{F_i}^l$ của mạng kết quả, nghĩa là:

Luật l : If x_1 is A_1^l and x_2 is A_2^l and .. and x_n is A_n^l Then $\bar{y}_f^l = w_f^l \cdot h_{F_i}^l$

Ta chỉ cần tính giá trị trọng số w_f^l của độ mạnh cháy thứ l cho mỗi luật được tác động bởi giải thuật FHEA. Lớp 3 là lớp ngõ ra của mạng kết quả cũng là ngõ ra của toàn thể mạng. Ngõ ra của mạng được cho bởi:

$$y = \sum_{l=1}^M \xi_f \bar{y}_f^l = \sum_{l=1}^M \xi_f (w_f^l h_{F_i}^l)$$



Hình 1.26 Cấu trúc của mạng neuron mờ dựa trên giải thuật FHEA

Giải thuật cập nhật trọng số w_j được đề xuất:

Vì mạng neuron mờ được cho ở hình 1.26 chủ yếu là mạng truyền thẳng nhiều lớp, nên có thể thiết kế giải thuật FHEA với giải thuật suy giảm độ dốc để học thông số dùng tiếp cận lan truyền sai số.

Giả sử hàm tổn thất sai số E được cho bởi:

$$E = \frac{1}{2}(y_m - y)^2$$

Ta dùng giải thuật lan truyền ngược sai số và giải thuật tối ưu độ dốc bậc 1 để chỉnh thông số w_j .

Đầu tiên, thực hiện các bước để xác định sai số học:

$$\delta^{(3)} = -\frac{\partial E}{\partial net^{(3)}} = -\frac{\partial E}{\partial y} = y_m - y$$

Tính $\frac{\partial E}{\partial w_j}$:

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial net^{(2)}} \cdot \frac{\partial net^{(2)}}{\partial w_j} = -\delta^{(2)} h_j$$

Trong đó:

$$\delta_j^{(2)} = -\frac{\partial E}{\partial net_j^{(2)}} = -\frac{\partial E}{\partial net^{(3)}} \cdot \frac{\partial net^{(3)}}{\partial f(net_j^{(2)})} \cdot \frac{\partial f(net_j^{(2)})}{\partial net_j^{(2)}} = \delta^{(3)} \xi_j$$

Suy ra: $\frac{\partial E}{\partial w_j} = (y - y_m) \cdot \xi_j \cdot h_j$

Giải thuật học để chỉnh trọng số là:

$$w_j(k+1) = w_j(k) - \eta \frac{\partial E}{\partial w_j}, j = 1, 2, \dots, M$$

Trong đó: tốc độ học η là hằng số dương ($0 < \eta < 1$).

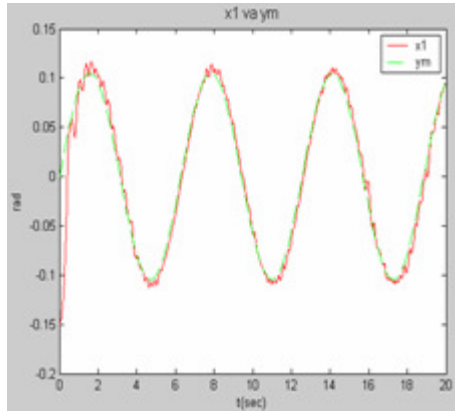
Giải thuật:

- Bước 1: Khởi động thông số con lắc.
Xây dựng tập mẫu từ sơ đồ điều khiển thích nghi.
- Bước 2: Trọng số được cập nhật trong quá trình huấn luyện mạng với tập mẫu nhận được từ bước 1.
- Bước 3: Xây dựng hàm S-function để mô tả đặc tính phi tuyến, bất ổn định của đối tượng.
- Bước 4: Kết nối sơ đồ Simulink điều khiển neuron mờ với S-function với các thông số đối tượng thay đổi (m và l), có nhiễu tiền định và nhiễu ngẫu nhiên.
- Bước 5: So sánh kết quả nhận được với các kết quả đã công bố.

4. Kết quả mô phỏng:

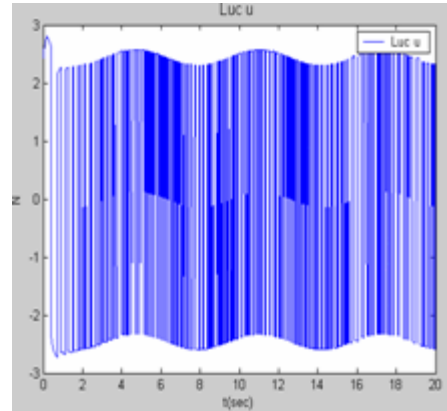
Giả sử với điều kiện ban đầu là $x(0)=(-0.15, 0)^T$.

Điều khiển neuron mờ thích nghi hệ con lắc ngược



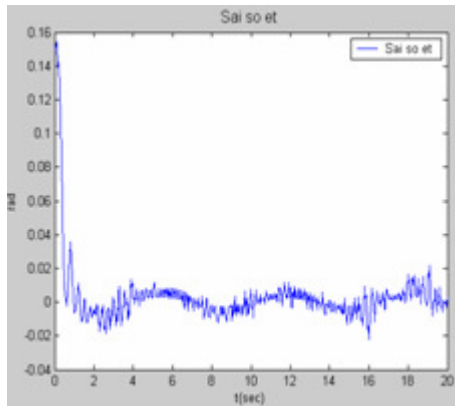
Hình 1.27

Quỹ đạo trạng thái $x_1(t)$



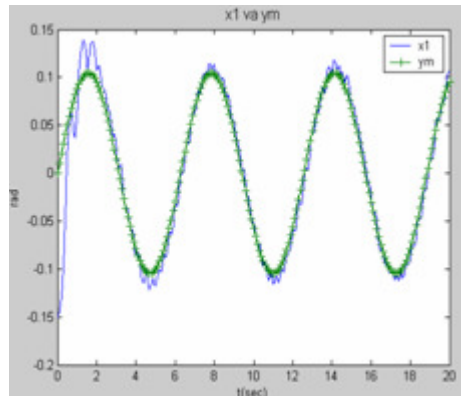
Hình 1.28

Quỹ đạo tín hiệu điều khiển $u(t)$



Hình 1.29

Sai số ngõ ra bám E_{out} cho điều khiển bám theo



Hình 1.30

Quỹ đạo trạng thái $x_1(t)$ khi dùng giải thuật Singleton

Nhận xét:

Quỹ đạo trạng thái $x_1(t)$ bám theo y_m khá tốt, tuy nhiên luật điều khiển $u(t)$ có tần số dao động cao. Sai số ngõ ra E_{out} tiến đến 0 sau 20 giây.

So sánh với mạng neuron mờ singleton:

Trong phần này hàm S-function thực hiện mô phỏng mạng neuron mờ singleton thay cho giải thuật FHEA.

Kết quả mô phỏng thu được ở hình 1.30, ta thấy x_1 bám theo y_m cũng khá tốt khi dùng mạng neuron mờ singleton thay cho giải thuật FHEA. Tuy nhiên so sánh 2 kết quả thu được từ hình 1.27 và hình 1.30 ta thấy dùng giải thuật FEHA tốt hơn dùng mạng neuron mờ Singleton.

5. Kết luận:

Tóm tắt kết quả điều khiển neuron mờ với giải thuật FHEA cho hệ con lắc ngược được đề nghị.

1. Trong sơ đồ điều khiển neuron mờ hệ con lắc ngược , hàm phi tuyến f và g được xấp xỉ bằng mạng neuron mờ dựa trên tiếp cận sai số ngõ ra mờ phân cấp . Mạng neuron mờ FHEA được học và huấn luyện trước dựa vào tập mẫu vào ra trong cách thức không trực tuyến.

2. Giá trị trọng số là phi tuyến vì hệ con lắc ngược là phi tuyến , f và g phi tuyến theo.

3. Kết quả mô phỏng đã chứng minh rằng giải thuật FHEA có thể cải thiện tính hiệu quả của xấp xỉ bám theo cho hệ phi tuyến chưa biết. Với nhiều tiền định, ngõ ra còn bám theo tín hiệu mong muốn. Nếu đưa vào nhiễu ngẫu nhiên thì điều khiển bám theo mất ổn định.

4. Bộ điều khiển neuron mờ cho kết quả mô phỏng tốt trong giới hạn trọng lượng con lắc m và chiều dài con lắc $L=2l$: m thay đổi từ 0.1kg đến 0.3kg, và L thay đổi từ 1m đến 5m.

5. Kết quả dùng giải thuật FHEA cho kết quả mô phỏng tốt hơn và nhanh hơn dùng mạng neuron mờ singleton.

CÂU HỎI ÔN TẬP VÀ BÀI TẬP

1. Nêu rõ mô hình điều khiển mờ theo Mamdani và theo Tagaki/Sugeno. So sánh ưu nhược điểm hai mô hình đó.
2. Các bước thiết kế một bộ điều khiển mờ. Vì sao nói điều khiển mờ là điều khiển dựa trên kinh nghiệm ?
3. Nguyên lý chỉnh định thông số của bộ PID mờ. Ứng dụng trong điều khiển nhiệt độ.
4. Thế nào là mạng nơron nhân tạo ? Nêu nội dung của thuật toán lan truyền ngược.
5. Vì sao phải kết hợp mạng nơron và hệ mờ ? Nêu sơ đồ kiểu mẫu của một hệ mờ - nơron. Ứng dụng mạng RBF trong nhận dạng hệ thống phi tuyến.
6. Để điều khiển tự động máy điều hoà nhiệt độ bằng kỹ thuật logic mờ, người ta dùng hai cảm biến: Trong phòng là cảm biến nhiệt T_i , bên ngoài là cảm biến nhiệt T_o . Việc điều hoà nhiệt độ thông qua điều khiển tốc độ quạt làm lạnh máy điều hoà. Biết rằng:

- Tâm nhiệt độ quan tâm là $[0^{\circ}\text{C} - 50^{\circ}\text{C}]$
- Tốc độ quạt là $v \in [0 - 600 \text{ vòng/ phút}]$

Hãy tính tốc độ quạt trong các trường hợp sau:

1. $T_i = 21^{\circ}\text{C}$ $T_o = 28^{\circ}\text{C}$
2. $T_i = 30^{\circ}\text{C}$ $T_o = 35^{\circ}\text{C}$
3. $T_i = 26^{\circ}\text{C}$ $T_o = 33^{\circ}\text{C}$

Nhận xét kết quả của các trường hợp trên.

7. Thiết kế bộ mờ điều khiển nhiệt độ. Bộ mờ có 2 ngõ vào là sai lệch $e(t)$ [ET] và đạo hàm sai lệch $de(t)$ [DET], một ngõ ra là đạo hàm công suất [DP]. Biết rằng:

- _ Lò nhiệt có công suất là 5KW, tầm đo max là 200°C , sai số là $\pm 5\%$.
- _ Tầm thay đổi của DET là $-10^{\circ}\text{C}/s \rightarrow 10^{\circ}\text{C}/s$
- _ Tầm thay đổi của DP là $-100\text{W}/s \rightarrow 100\text{W}/s$

Hãy tính công suất cung cấp cho lò trong các trường hợp sau:

1. $ET = 8^{\circ}\text{C}$ $DET = 9^{\circ}\text{C}/s$
2. $ET = 2^{\circ}\text{C}$ $DET = 9^{\circ}\text{C}/s$
3. $ET = 7^{\circ}\text{C}$ $DET = -7^{\circ}\text{C}/s$

8. Cho một đối tượng lò nhiệt có hàm truyền :

$$G(S) = \frac{4.228}{(S + 0.5)(S^2 + 1.64S + 8.456)}$$

a. Tính thông số bộ PID theo Zeigler-Nichols, tính POT, t_s .

b. Thiết kế bộ PID mờ thoả mãn các điều kiện sau: POT < 10% và $t_s < 5$

Biết rằng:

- Nhiệt độ đặt $T_s = 200^0\text{C}$
- Sai số $e_{\max} = \pm 5\%$
- DET $\in [-10 \ +10]$ ($^0\text{C/s}$)
- Công suất của lò nhiệt $P = 5\text{KW}$

Tìm K_P , K_I , K_D và tính công suất trong các trường hợp sau:

1. ET = 8^0C DET = 9^0C/s
2. ET = 2^0C DET = 2^0C/s
3. ET = 8^0C DET = $- 9^0\text{C/s}$

Nhận xét kết quả đạt được.

9. Xét hệ thống phi tuyến bậc hai như sau:

$$\ddot{x} + \alpha(t)\dot{x}^2 \cos 3x = u$$

trong đó $\alpha(t)$ chưa biết và $1 \leq \alpha(t) \leq 2$.

Thiết kế BDK trượt u để x bám theo quỹ đạo mong muốn x_d .

10. Cho hệ thống :

$$\begin{aligned} \dot{x}_1 &= \sin x_2 + x_2 \sqrt{t+1} \\ \dot{x}_2 &= \alpha_1(t)x_1^4 \cos x_2 + \alpha_2(t)u \end{aligned}$$

Trong đó $\alpha_1(t)$ và $\alpha_2(t)$ là hai hàm chưa biết và : $|\alpha_1(t)| \leq 10, 1 \leq \alpha_2(t) \leq 2$.

Thiết kế BDK mờ trượt với trạng thái vòng kín $x_I(t)$ được cho bởi trạng thái mong muốn $x_d(t)$.

11. Cho bài toán phân loại với các cặp vào/ra như sau:

$$\left\{ p_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, r_1 = : \right\} \left\{ p_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, r_2 = 1 \right\} \left\{ p_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, r_3 = 1 \right\}$$

$$\left\{ p_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, r_4 = 0 \right\} \quad \left\{ p_5 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, r_5 = 0 \right\}$$

1. Vẽ sơ đồ mạng perceptron cho bài toán. Có bao nhiêu ngõ vào yêu cầu
2. Biểu diễn các cặp ngõ vào/ra trên đồ thị. Bài này có thể giải bằng sơ đồ mạng vừa thiết kế ở trên không ? Tại sao?

12. Cho bài toán phân loại với các cặp vào/ra như sau:

$$\left\{ p_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, r_1 = 1 \right\} \left\{ p_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, r_2 = 1 \right\} \left\{ p_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, r_3 = 0 \right\} \left\{ p_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, r_4 = 0 \right\}.$$

1. Thiết kế mạng single-neuron perceptron (SNP) để giải bài toán này. Vẽ đồ thị các vector vào/ra, từ đó chọn vector trọng số W.
2. Kiểm tra lại mạng vừa thiết kế với 4 vector ngõ vào trên.
3. Dùng mạng trên để phân loại 4 vector sau:

$$p_5 = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \quad p_6 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad p_7 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad p_8 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

4. Trong 4 vector trên, vector nào cho kết quả phân loại giống nhau (không phụ thuộc vào W và b). Tại sao?

13. Giải lại bài toán 12 bằng bất phương trình

14. Giải lại bài toán 12 bằng cách sử dụng luật perceptron với giá trị đầu là:

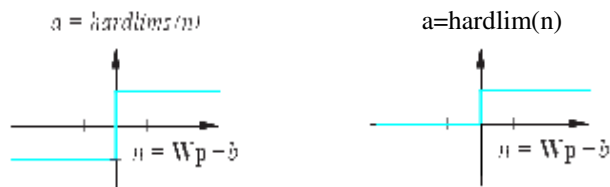
$$W(0) = [0 \ 0] \text{ và } b(0) = 0$$

15. Chứng minh bằng toán học rằng không thể giải bài toán sau bằng mạng single-neuron perceptron hai ngõ vào:

$$\left\{ p_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, r_1 = 1 \right\} \left\{ p_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, r_2 = 0 \right\} \left\{ p_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, r_3 = 1 \right\} \left\{ p_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, r_4 = 0 \right\}$$

(Gợi ý: dùng bất đẳng thức).

16. Sử dụng hàm hardlim() bằng hàm hardlims(), tức là giá trị ngõ ra sẽ là [-1 1] thay vì [0 1].



1. Viết biểu thức chuyển đổi từ hàm hardlim() sang hàm hardlims()
2. Hai mạng neuron có cấu trúc W và b như nhau: mạng thứ nhất sử dụng hàm nấc thường hardlim(), mạng thứ hai sử dụng hàm nấc đối xứng hardlims(). Với cùng giá trị đầu vào p và cùng sử dụng luật học perceptron thì vector W có giá trị như nhau cho cả hai mạng không?
3. Nếu sự biến đổi của W là khác nhau thì chúng khác nhau như thế nào. Tại sao?

4. Giữ nguyên giá trị ban đầu W và b (giá trị đầu của mạng perceptron sử dụng hàm `hardlim()`). Hãy tạo một thuật toán cho mạng perceptron sử dụng hàm `hardlims()` để hai mạng trên luôn đáp ứng như nhau khi huấn luyện trên cùng một dữ liệu.

17. Khi đo trọng lượng và chiều dài của tai thỏ và gấu ta thu được các vector:

$$\left\{ p_1 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}, r_1 = 0 \right\} \left\{ p_2 = \begin{bmatrix} 1 \\ 5 \end{bmatrix}, r_2 = 0 \right\} \left\{ p_3 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, r_3 = 0 \right\} \left\{ p_4 = \begin{bmatrix} 2 \\ 5 \end{bmatrix}, r_4 = 0 \right\}$$

$$\left\{ p_5 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, r_5 = 1 \right\} \left\{ p_6 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, r_6 = 1 \right\} \left\{ p_7 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, r_7 = 1 \right\} \left\{ p_8 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, r_8 = 1 \right\}$$

Trong đó:

Phần tử thứ nhất của vector ngõ vào là trọng lượng

Phần tử thứ hai của vector ngõ vào là chiều dài của tai

Chuẩn ngõ ra 0 ứng với thỏ

Chuẩn ngõ ra 1 ứng với gấu

1. Dùng Matlab để đặt giá trị đầu và huấn luyện mạng giải bài toán.
2. Dùng Matlab để kiểm tra giá trị W và b đạt được dựa vào các vector ngõ vào.
3. Thay đổi giá trị các ngõ vào sao cho bài toán có lời giải (tức là tồn tại đường phân chia tuyến tính). Sau đó huấn luyện lại mạng này.

18. Sau đây là một dạng khác của luật học perceptron:

$$W^{neu} = W^{old} + \alpha e p^T$$

$$b^{neu} = b^{old} + \alpha e$$

Trong đó: α được gọi là tốc độ học

Chứng minh giải thuật này hội tụ. Kết quả có đòi hỏi gì về giới hạn của α không? Giải thích.

19. Dùng Simulink để mô phỏng các hệ thống “Quả bóng và đòn bẩy”, “Con lắc ngược”, hệ thống điều khiển nhiệt độ dùng PID mờ.

20. Tham khảo các ví dụ trong phần Help/Fuzzy Control Toolbox của Matlab.

21. Xây dựng M-file, s-function trong Simulink cho hệ thống.

Chương 3

ĐIỀU KHIỂN THÍCH NGHI

3.1 Khái niệm

3.1.1 Định nghĩa

“Thích nghi là quá trình thay đổi thông số và cấu trúc hay tác động điều khiển trên cơ sở lượng thông tin có được trong quá trình làm việc với mục đích đạt được một trạng thái nhất định, thường là tối ưu khi thiếu lượng thông tin ban đầu cũng như khi điều kiện làm việc thay đổi” hay :

“Điều khiển thích nghi là tổng hợp các kỹ thuật nhằm tự động chỉnh định các bộ điều chỉnh trong mạch điều khiển nhằm thực hiện hay duy trì ở một mức độ nhất định chất lượng của hệ khi thông số của quá trình được điều khiển không biết trước hay thay đổi theo thời gian”.

Hệ thống được mô tả trong hình dưới đây gồm 2 vòng:

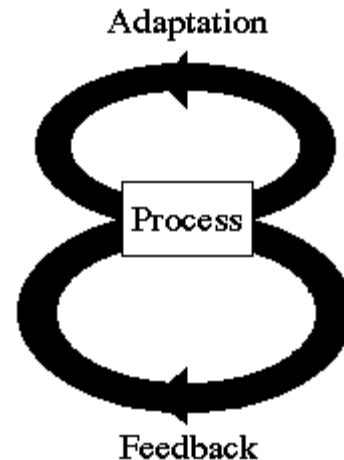
- Vòng hồi tiếp thông thường
- Vòng hồi tiếp điều khiển thích nghi

Kết luận

1. Điều khiển thích nghi liên quan đến:
 - Sự thay đổi của quá trình động học
 - Sự thay đổi của các nhiễu lên hệ thống
2. Các hệ thống thích nghi là phi tuyến

3.1.2 Nhận dạng hệ thống

- Làm thế nào để có được mô hình?



- Vật lí (hộp trắng)
- Kinh nghiệm (hộp đen)
- Kết hợp (hộp xám)
- Kế hoạch hoá thực nghiệm
- Chọn lựa cấu trúc mô hình
 - Các hàm chuyển đổi
 - Đáp ứng xung
 - Các mô hình trạng thái
- Tham số thích nghi
 - Thống kê
 - Các vấn đề nghịch đảo (Inverse Problems)
- Sự hợp lí

3.1.3 Ước lượng tham số thích nghi thời gian thực

1. Giới thiệu
2. Bình phương cực tiểu và hồi qui
3. Hệ thống động
4. Các điều kiện thực nghiệm
5. Các ví dụ
6. Các kết luận

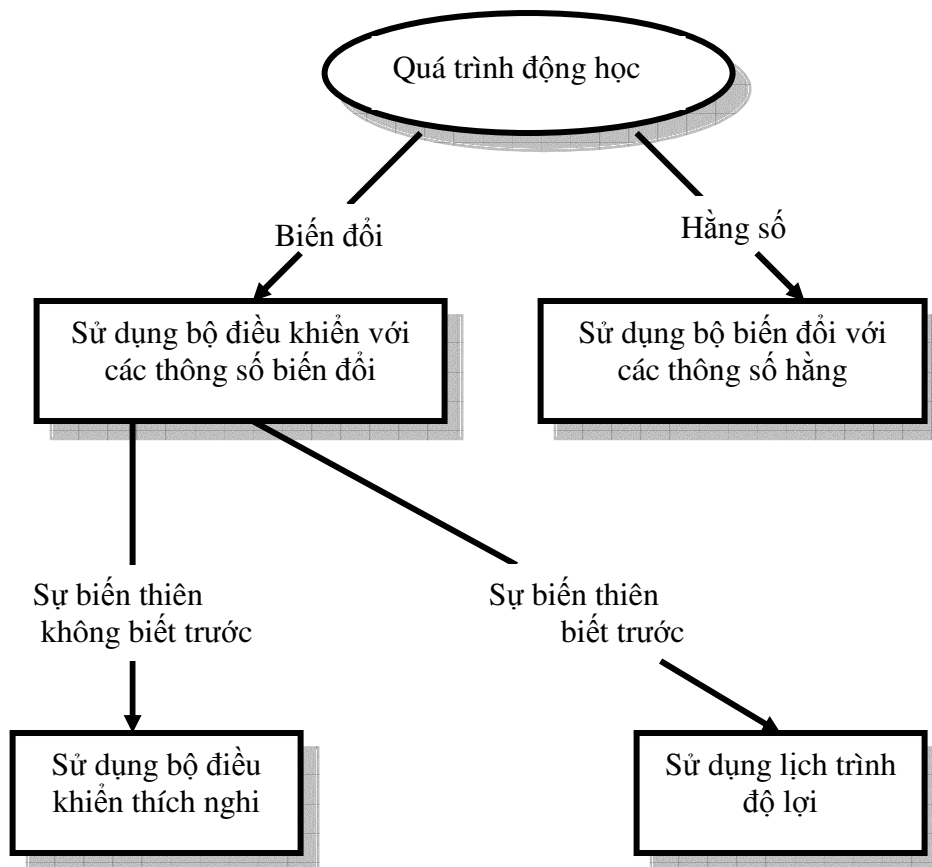
3.1.4 Phân loại

Có thể phân loại các hệ thích nghi theo các tiêu chuẩn sau :

1. Hệ thích nghi mô hình tham chiếu (MRAS)
2. Bộ tự chỉnh định (STR)
3. Lịch trình độ lợi
4. Hệ tự học
5. Hệ tự tổ chức

3.1.5 Ứng dụng

- Tự chỉnh định
- Lịch trình độ lợi
- Thích nghi liên tục



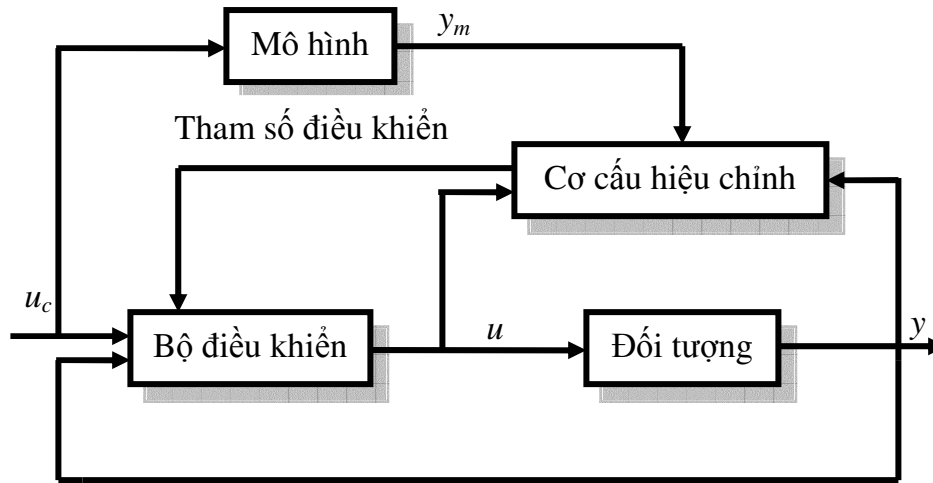
Hình 3.1 Sơ đồ các ứng dụng

3.2 Hệ thích nghi mô hình tham chiếu – MRAS

(Model Reference Adaptive Systems)

3.2.1 Sơ đồ chức năng

Hệ thống thích nghi sử dụng mô hình chuẩn là một trong những phương pháp chính của điều khiển thích nghi. Nguyên lí cơ bản được trình bày ở hình 3.2



Hình 3.2 Sơ đồ khối của một hệ thống thích nghi mô hình tham chiếu

Mô hình chuẩn sẽ cho đáp ứng ngõ ra mong muốn đối với tín hiệu đặt (yêu cầu). Hệ thống có một vòng hồi tiếp thông thường bao gồm đối tượng và bộ điều khiển. Sai số e là sai lệch giữa ngõ ra của hệ thống và của mô hình chuẩn $e = y - y_m$. Bộ điều khiển có thông số thay đổi dựa vào sai số này. Hệ thống có hai vòng hồi tiếp: hồi tiếp trong là vòng hồi tiếp thông thường và vòng hồi tiếp bên ngoài hiệu chỉnh tham số cho vòng hồi tiếp bên trong. Vòng hồi tiếp bên trong được giả sử là nhanh hơn vòng hồi tiếp bên ngoài.

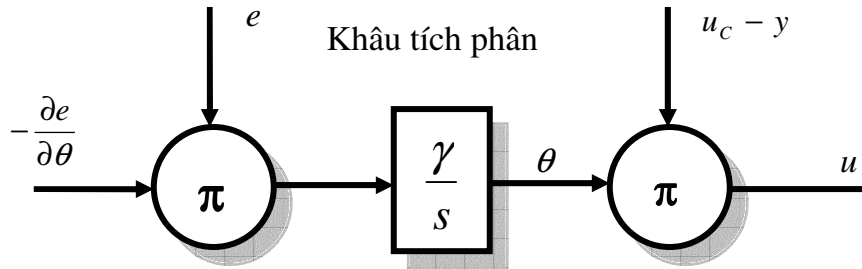
Hình 3.2 là mô hình MRAS đầu tiên được đề nghị bởi Whitaker vào năm 1958 với hai ý tưởng mới được đưa ra: Trước hết sự thực hiện của hệ thống được xác định bởi một mô hình, thứ hai là sai số của bộ điều khiển được chỉnh bởi sai số giữa mô hình chuẩn và hệ thống. Mô hình chuẩn sử dụng

trong hệ thích nghi bắt nguồn từ hệ liên tục sau đó được mở rộng sang hệ rời rạc có nhiễu ngẫu nhiên.

Chương này tập trung vào ý tưởng cơ bản. Để vấn đề được trình bày một cách rõ ràng, ta chỉ tập trung vào cấu hình trong hình 3.2 được gọi là hệ MRAS song song. Đây là một trong nhiều cách có thể xây dựng mô hình chuẩn. Chương này đề cập chính đến hệ liên tục theo phương pháp trực tiếp có nghĩa là tham số được cập nhật một cách trực tiếp.

3.2.2 Luật MIT (Massachusetts Institute Technology)

(MIT = Massachusetts Institute Technology : Viện công nghệ Massachusetts)



Hình 3.3 Mô hình sai số

Hệ thống thích nghi mô hình tham chiếu đầu tiên được đưa ra để giải quyết vấn đề: các đặc điểm của một mô hình tham chiếu yêu cầu ngõ ra là quá trình lý tưởng cần có đáp ứng đối với tín hiệu điều khiển như thế nào. Đồ thị minh họa trong hình 3.2. Trong trường hợp này, mô hình tham chiếu mang tính song song hơn là nối tiếp, giống như cho SOAS (Self Oscillating Adaptive Systems). Bộ điều khiển có thể được xem như bao gồm hai vòng: một vòng phía trong gọi là vòng hồi tiếp thông thường có quá trình và bộ điều khiển. Các thông số của bộ điều khiển được chỉnh định bởi vòng ngoài sao cho sai số e giữa ngõ ra y và ngõ ra mô hình y_m là nhỏ nhất. Vì vậy vòng ngoài còn được gọi là vòng chỉnh định. Vấn đề là xác định cơ cấu chỉnh định cho hệ thống ổn định, nghĩa là sai số bằng zero. Điều này không thể thực hiện được. Cơ cấu chỉnh định với thông số sau được gọi là luật MIT, được sử dụng cho hệ MRAS đầu tiên:

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}$$

Trong phương trình này e là sai số của mô hình $e = y - y_m$. Các thành phần của vector $\partial e / \partial \theta$ là đạo hàm độ nhạy của sai số đối với các thông số chỉnh định θ . Thông số γ xác định tốc độ thích nghi. Luật MIT có thể được giải thích như sau. Giả sử rằng các thông số θ thay đổi chậm hơn nhiều so với các biến khác của hệ thống. Để bình phương sai số là bé nhất, cần thay đổi các thông số theo hướng gradient âm của bình phương sai số e^2 .

Giả sử muốn thay đổi thông số của bộ điều khiển sao cho sai số giữa ngõ ra của đối tượng và của mô hình chuẩn tiến tới zero. Đặt e là sai số và θ là thông số hiệu chỉnh. Chỉ tiêu chất lượng :

$$J(\theta) = \frac{1}{2} e^2 \quad (3.1)$$

để làm cho $J(\theta)$ MIN thì cần phải thay đổi các thông số theo hướng âm của gradient J , có nghĩa là :

$$\frac{\partial \theta}{\partial t} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (3.2)$$

Giả sử rằng các thông số cần thay đổi θ thay đổi chậm hơn nhiều so với các biến khác của hệ thống. Vì vậy đạo hàm $\frac{\partial e}{\partial \theta}$ được tính với giả thiết θ là hằng số. Biểu thức đạo hàm $\frac{\partial e}{\partial \theta}$ gọi là hàm độ nhạy của hệ thống. Luật điều chỉnh theo phương trình (3.2) với $\frac{\partial e}{\partial \theta}$ là độ nhạy thì có liên hệ giống như luật MIT. Cách chọn hàm tổn thất theo phương trình (3.1) có thể là tùy ý. Nếu chọn

$$J(\theta) = |e| \quad (3.3)$$

Khi đó luật hiệu chỉnh sẽ là :

$$\frac{d\theta}{dt} = -\gamma \frac{\partial e}{\partial \theta} \text{sign}(e) \quad (3.4)$$

Hoặc

$$\frac{d\theta}{dt} = -\gamma \text{sign}\left(\frac{\partial e}{\partial \theta}\right) \text{sign}(e)$$

Đây gọi là giải thuật dấu - dấu. Hệ rời rạc sử dụng giải thuật này được ứng dụng trong viễn thông nơi đòi hỏi tính toán nhanh và thực hiện đơn giản.

Phương trình (3.2) còn được áp dụng trong trường hợp có nhiều thông số hiệu chỉnh, khi đó θ trở thành một vector và $\frac{\partial e}{\partial \theta}$ là gradient của sai số đối với các thông số tương ứng. Ứng dụng của luật MIT được biểu diễn bằng hai ví dụ sau :

Ví dụ 3.1 - Hiệu chỉnh độ lợi nuôi tiến

Xét vấn đề hiệu chỉnh độ lợi nuôi tiến với mô hình và đối tượng đều có hàm truyền là $G(S)$. Sai số là:

$$e = y - y_m = G(p)\theta u_c - G(p)\theta^o u_c$$

với u_c là tín hiệu đặt, y_m là ngõ ra mô hình, y là ngõ ra đối tượng, θ là thông số hiệu chỉnh, và $p = d/dt$ là toán tử vi phân. Độ nhạy khi ấy bằng :

$$\frac{\partial e}{\partial \theta} = G(p)u_c = y_m/\theta^o$$

Luật MIT được cho :

$$\frac{d\theta}{dt} = -\gamma y_m e / \theta^o$$

Nếu dấu của θ^o được biết, khi ấy đưa ra $\gamma = \gamma / \theta^o$

Sự thay đổi của tham số θ tỉ lệ với tích sai số e và ngõ ra của mô hình y_m .

Ví dụ trên không dùng việc xấp xỉ : Khi luật MIT được áp dụng vào những vấn đề phức tạp hơn thì cần phải có xấp xỉ để tính được độ nhạy.

Ví dụ 3.2 MRAS cho hệ bậc nhất

Xét hệ thống được mô tả bởi phương trình:

$$\frac{dy}{dt} = -ay + bu \quad (3.5)$$

với u là biến điều khiển, y là ngõ ra được đo lường.

Giả sử mong muốn có được hệ vòng kín được mô tả bởi:

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

Mô hình kèm theo hoàn hảo có thể đạt được với bộ điều khiển :

$$u(t) = t_0 u_c(t) - s_0 y(t) \quad (3.6)$$

với tham số $t_0 = b_m/b$; $s_0 = (a_m - a)/b$

Chú ý hồi tiếp sẽ là dương nếu $a_m < a$, nghĩa là mô hình mong muốn thì chậm hơn quá trình. Để áp dụng luật MIT, sử dụng sai số $e = y - y_m$, với y là ngõ ra hệ kín.

Theo phương trình (3.5) và (3.6) thì:

$$y = \frac{bt_0}{p + a + bs_0} u_c$$

với p là toán tử vi phân. Độ nhạy có thể tính được bằng cách lấy đạo hàm riêng phần theo tham số của bộ điều khiển s_0 và t_0 :

$$\frac{\partial e}{\partial t_0} = \frac{b}{p + a + bs_0} u_c$$

$$\frac{\partial e}{\partial s_0} = -\frac{b^2 t_0}{(p + a + bs_0)^2} u_c = -\frac{b}{p + a + bs_0} y$$

Các công thức này không thể dùng vì thông số đối tượng a và b chưa biết. Vì vậy cần phải làm xấp xỉ để có được luật hiệu chỉnh tham số thực tế. Để thực hiện điều này, đầu tiên cần quan sát với giá trị tối ưu của tham số bộ điều khiển, ta có :

$$p + a + bs_0 = p + a_m$$

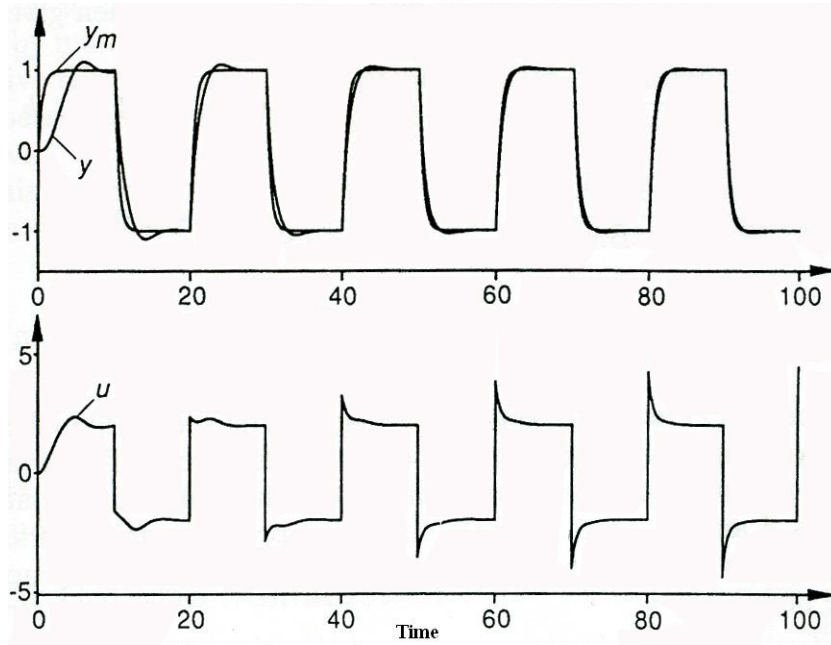
Hơn nữa cần chú ý là b có thể được bao gồm trong hệ số tốc độ thích nghi γ . Bởi vì nó xuất hiện trong tích γb , điều này đòi hỏi dấu của b phải được biết. Sau khi xấp xỉ, luật cập nhật các tham số điều khiển có được là:

$$\begin{aligned} \frac{dt_0}{dt} &= -\gamma \left(\frac{1}{p + a_m} u_c \right) e \\ \frac{ds_0}{dt} &= \gamma \left(\frac{1}{p + a_m} y \right) e \end{aligned} \quad (3.7)$$

Ví dụ trên chỉ cách sử dụng luật MIT để tạo được luật hiệu chỉnh thông số.

Kết quả mô phỏng hệ MRAS trong ví dụ 3.2 các với thông số như sau:

- $a = 1$, $b = 0.5$, $a_m = 2$ và $b_m = 2$.
- Tín hiệu vào là sóng vuông với biên độ bằng 1 và $\gamma = 2$.



Đáp ứng của ngõ ra y , ngõ ra tham chiếu y_m và tín hiệu điều khiển u .

Nhận xét:

- Hệ thống vòng kín đã đạt đến đáp ứng mong muốn chỉ sau một thời gian ngắn.
- Tốc độ hội tụ phụ thuộc vào hai thông số là γ và b

Điều đáng quan tâm nhất qua ví dụ trên là cách mà luật MIT được sử dụng để hiệu chỉnh các thông số.

- Nó không nhất thiết đòi hỏi phải có một mô hình kèm theo hoàn hảo. Và quá trình này có thể áp dụng cho hệ phi tuyến.
- Ví dụ này đã sử dụng lại cấu trúc như hình 3.3. Có 2 bộ nhân được sử dụng. Trong đó: bộ nhân thứ nhất là của e và $\frac{\partial e}{\partial \theta}$, và phương trình 3.7 cung cấp thông số cho bộ nhân thứ hai.
- Việc xấp xỉ là rất quan trọng bởi vì nếu xấp xỉ tốt ta sẽ có được luật hiệu chỉnh thông số đáng tin cậy.

Luật MIT sẽ đạt hiệu quả cao nếu như ta chọn độ thích nghi γ nhỏ. Tuy nhiên, giới hạn này còn tùy thuộc vào biên độ của tín hiệu chuẩn cũng như là độ lợi của hệ thống. Trong một số trường hợp, luật MIT có thể làm mất tính ổn định của hệ thống. Do đó, khi sử dụng luật hiệu chỉnh ta cũng cần phải quan tâm đến tính ổn định của hệ thống.

Vài tính chất sau cần chú ý:

1. Không cần thiết đòi hỏi một mô hình kèm theo hoàn hảo. Các thủ tục có thể được áp dụng cho hệ phi tuyến. Phương pháp này cũng có thể được dùng để điều khiển cho hệ biết trước một phần.

2. Cấu trúc như hình 3.3 có một phép nhân giữa e và $\frac{\partial e}{\partial \theta}$.

Lấy tích phân phương trình (3.7) sẽ cho ra các tham số và được truyền đến bộ điều khiển sử dụng phép nhân thứ hai.

3. Sự xấp xỉ là cần thiết để có được luật điều khiển hiệu chỉnh tham số thực tế.

Luật MIT có thể thực hiện tốt nếu độ lợi thích nghi γ là nhỏ. Độ lớn γ tùy thuộc vào biên độ của tín hiệu chuẩn và độ lợi của đối tượng. Vì vậy không thể có một giới hạn cố định đảm bảo an toàn do đó luật MIT có thể cho một hệ vòng kín không an toàn. Luật hiệu chỉnh bổ sung có thể được dùng bằng lý thuyết ổn định. Những luật này tương tự luật MIT nhưng các hàm độ nhạy thì đương nhiên là khác. Ý này được trình bày nhiều hơn trong mục 3.2.4

3.2.3 Nội dung, phương pháp thiết kế MRAS

Có ba phương pháp cơ bản để phân tích và thiết kế hệ MRAS :

- Phương pháp tiếp cận Gradient
- Hàm Lyapunov
- Lý thuyết bị động

Phương pháp gradient được dùng bởi Whitaker đầu tiên cho hệ MRAS. Phương pháp này dựa vào giả sử tham số của bộ hiệu chỉnh thay đổi chậm hơn các biến khác của hệ thống. Giả sử này thừa nhận có sự ổn định giả cần thiết cho việc tính toán độ nhạy và cho cơ cấu hiệu chỉnh thích nghi. Phương pháp tiếp cận gradient không cho kết quả cần thiết cho hệ thống kín ổn định. Bộ quan sát được đưa ra để áp dụng lý thuyết ổn định Lyapunov và lý thuyết bị động được dùng để bổ sung cho cơ cấu thích nghi.

Đối với hệ thống có tham số điều chỉnh được như trong hình 3.2, phương pháp thích nghi sử dụng mô hình chuẩn cho một cách hiệu chỉnh tham số tổng quát để có được hàm truyền hệ thống vòng kín gần với mô hình. Đây gọi là vấn đề mô hình kèm theo. Một câu hỏi đặt ra là chúng ta làm cho sai

lệch nhỏ như thế nào, điều này phụ thuộc bởi mô hình, hệ thống và tín hiệu đặt. Nếu có thể làm cho sai số bằng 0 đối với mọi tín hiệu yêu cầu thì gọi là mô hình kèm theo hoàn hảo.

Mô hình kèm theo

Vấn đề mô hình kèm theo có thể được giải quyết bằng thiết kế phân số cực (miêu tả ngắn gọn về thiết kế phân cực được cho trong phụ lục A (TLTK[1])). Mô hình kèm theo là cách đơn giản để thiết lập hay giải một vấn đề điều khiển tự động. Mô hình sử dụng có thể là tuyến tính hay phi tuyến. Các tham số trong hệ thống được hiệu chỉnh để có được y càng gần với y_m càng tốt đối với một tập các tín hiệu vào. Phương pháp thích nghi là một công cụ thiết kế hệ MRAS, vấn đề này được trình bày trong mục 3.2.4. Mặc dù mô hình kèm theo hoàn hảo chỉ có thể đạt được trong điều kiện lý tưởng nhưng phân tích trường hợp này sẽ cho hiểu biết sâu sắc vào vấn đề thiết kế.

Xét hệ 1 đầu vào, 1 đầu ra có thể là liên tục hay rời rạc có phương trình:

$$y(t) = \frac{B}{A}u(t) \quad (3.8)$$

với u là tín hiệu điều khiển, y là ngõ ra. Kí hiệu A, B là những đa thức theo biến S hay Z . Giả sử bậc của $A \geq$ bậc của B nghĩa là hệ thống là hợp thức (đối với hệ liên tục) và nhân quả đối với hệ rời rạc. Giả sử hệ số bậc cao nhất của A là 1. Tìm bộ điều khiển sao cho quan hệ giữa tín hiệu đặt u_c và tín hiệu ra mong muốn y_m được cho bởi :

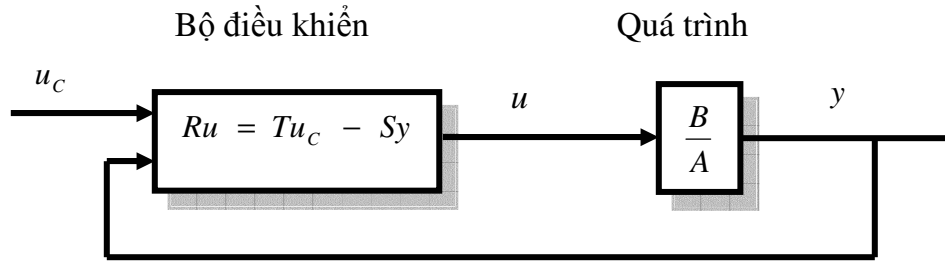
$$y_m = \frac{B_m}{A_m}u_c(t) \quad (3.9)$$

với A_m, B_m cũng là những đa thức theo biến S hoặc Z .

Luật điều khiển tổng quát được cho bởi :

$$Ru = Tu_c - Sy \quad (3.10)$$

với R, S, T là các đa thức. Luật điều khiển này được xem như vừa có thành phần hồi tiếp âm với hàm truyền $-S/R$ và thành phần nuôi tiến với hàm truyền T/R . Xem hình 3.4



Hình 3.4 Hệ vòng kín với bộ điều khiển tuyến tính tổng quát

Khử u ở 2 phương trình (3.8) và (3.10) được phương trình sau cho hệ thống vòng kín :

$$(AR + BS)y = BTu_c \quad (3.11)$$

Để đạt được đáp ứng vòng kín mong muốn, thì $AR + BS$ phải chia hết cho A_m , các zero của đối tượng, khi cho $B = 0$, sẽ là zero của hệ kín nếu không bị khử bởi cực vòng kín.

Bởi vì các điểm zero không ổn định không thể bị khử nên có thể phân tích thành $B = B^+B^-$, trong đó B^+ chứa những thành phần có thể khử đi, B^- là thành phần còn lại.

Theo phương trình (3.11) $AR + BS$ là đa thức đặc trưng của hệ thống được phân tích thành ba thành phần : khử zero của đối tượng: B^+ ; cực mong muốn của mô hình được cho bởi A_m ; các cực của bộ quan sát A_0 . Vì thế :

$$AR + BS = B^+A_0A_m \quad (3.12)$$

gọi là *phương trình Diophantine* (hay là *phương trình nhận dạng Benzout*). Vì B^+ có thể khử nên :

$$R = B^+R_1 \quad (3.13)$$

Chia phương trình (3.12) cho B^+ sẽ được:

$$A \cdot R_1 + B^- \cdot S = A_0A_m \quad (3.14)$$

Vì yêu cầu là phải giống đáp ứng mong muốn nên tử số (3.11) phải chia hết cho B_m , nếu không thì sẽ không có lời giải cho bài toán thiết kế. Vì vậy :

$$B_m = B^- \cdot B'_m \quad (3.15)$$

$$T = A_0B'_m$$

Điều kiện để đảm bảo tồn tại lời giải là :

$$\text{bậc}(A_0) \geq 2 \text{ bậc}(A) - \text{bậc}(A_m) - \text{bậc}(B^+) - 1$$

$$\text{bậc}(A_m) - \text{bậc}(B_m) \geq \text{bậc}(A) - \text{bậc}(B)$$

Những điều kiện này được cho trong phụ lục A (TLTK[1]).

Giả sử tất cả các zero đều bị khử, khi đó có thể viết (3.14) lại như sau :

$$A_0 A_m = AR_I + b_0 S$$

Nhân 2 vế cho y và dùng thêm phương trình (3.8) ta được :

$$\begin{aligned} A_0 A_m y &= BR_I u + b_0 S y \\ &= b_0 (Ru + Sy) \end{aligned} \quad (3.16)$$

Các thông số ở vế trái đã biết, vế phải chưa biết. Đa thức T có được trực tiếp từ phương trình (3.15). Các tham số mô hình của phương trình (3.16) bây giờ có thể được dùng để ước lượng các tham số chưa biết của bộ điều khiển (chương 3 TLTK[1]). Điều này dẫn đến hệ MRAS trực tiếp. Lời giải tổng quát được trình bày trong chương 4 TLTK[1].

Hệ tuyến tính tổng quát

Hệ SISO được mô tả bởi phương trình sau:

$$Ay = Bu$$

Với đặc tính hệ thống mong muốn đạt được là:

$$A_m y_m = B_m u_c$$

Bộ điều khiển:

$$Ru = Tu_c - Sy \quad (*)$$

Hệ vòng kín được mô tả:

$$y = \frac{BT}{AR + BS} u_c$$

Thay y vào (*) ta tính được:

$$u = \frac{AT}{AR + BS} u_c$$

Sai số là:

$$e = y - y_m$$

Bây giờ cần phải xác định các đạo hàm riêng của sai số đối với từng tham số hiệu chỉnh để tìm luật chỉnh định thông số các hàm độ nhạy.

Đặt r_i, s_i, t_i là các hệ số của đa thức R, S, T . Các hàm độ nhạy được cho bởi:

$$e = \frac{BT}{AR + BS} u_C - \frac{B_m u_C}{A_m}$$

$$\rightarrow \frac{\partial e}{\partial r_i} = -\frac{BTAp^{k-i}}{(AR + BS)^2} u_C = -\frac{Bp^{k-i}}{AR + BS} u \quad i = 1, \dots, k$$

$$\frac{\partial e}{\partial s_i} = -\frac{BTBp^{l-i}}{(AR + BS)^2} u_C = -\frac{Bp^{l-i}}{AR + BS} y \quad i = 0, \dots, l$$

$$\frac{\partial e}{\partial t_i} = \frac{Bp^{m-i}}{AR + BS} u_C \quad i = 0, \dots, m$$

Trong đó $k = \text{bậc}(R)$, $l = \text{bậc}(S)$, $m = \text{bậc}(T)$.

Về phía các phương trình trên còn chứa A, B là các thông số chưa biết nên không tính được các hàm độ nhạy. Một cách xấp xỉ để có được luật cập nhật có thực tế là:

$$AR + BS \approx A_0 A_m B^+$$

Suy ra các hàm độ nhạy:

$$\frac{\partial e}{\partial r_i} \approx -\frac{B^- p^{k-i}}{A_0 A_m} u$$

Tương tự cho s_i và t_i

Tuy nhiên về phía vẫn còn B^- là chưa biết. Nếu tất cả các zero đều được khử, khi đó ta có $B^- = b_0$. Nếu dấu của b_0 biết được thì có thể thực hiện được luật cập nhật thông số. Thành phần b_0 có thể được bao gồm trong cả γ . Nên có thể suy ra luật cập nhật hiệu chỉnh các thông số như sau:

$$\frac{dr_i}{dt} = \gamma e \frac{p^{k-i}}{A_0 A_m} u \quad i = 1, \dots, k = \text{bậc}(R)$$

$$\frac{ds_i}{dt} = \gamma e \frac{p^{l-i}}{A_0 A_m} y \quad i = 0, \dots, l = \text{bậc}(S)$$

$$\frac{dt_i}{dt} = -\gamma e \frac{p^{m-i}}{A_0 A_m} u_C \quad i = 0, \dots, m = \text{bậc}(T)$$

Nhận xét:

- Cần phải xây dựng 3 trạng thái của bộ lọc $\frac{1}{A_0 A_m}$ cho luật hiệu chỉnh trên.
- Sự thay đổi các tham số này tỉ lệ với tích sai số e và tín hiệu bộ lọc $\frac{1}{A_0 A_m}$
- Để có được luật điều chỉnh các tham số trên cần phải giả sử các zero phải ổn định và dấu của b_0 phải được biết.
- Có thể tránh được giả sử này bằng cách sử dụng các thuật toán phức tạp hơn như ước lượng trạng thái...

• **Tiêu chuẩn cực tiểu hoá**

- Luật MIT có thể được sử dụng cho các hàm tổn thất khác.
- Luật hiệu chỉnh các tham số có thể đạt được bằng cách tính gradient hàm tổn thất đối với các tham số và sự thay đổi các tham số phải ngược dấu với gradient.
- Phương pháp này cần biết các tham số của mô hình đối tượng để tính toán độ nhạy. Tuy nhiên điều này là không có thực và do đó có thể sử dụng phương pháp xấp xỉ hay bằng các bộ ước lượng thông số.

Sai số và sự hội tụ tham số

Hệ thống thích nghi sử dụng mô hình chuẩn dựa vào ý tưởng là làm cho sai số $e = y - y_m$ tiến tới zero. Điều này không có nghĩa là các tham số điều khiển tiến tới giá trị đúng của nó (ví dụ như trường hợp tín hiệu = 0).

Ví dụ 3.3 Hội tụ sai số

Giả sử hệ thống có sơ đồ như hình 3.5:

Ngõ ra: $y = u$

Luật điều khiển: $u = \theta u_c$

Mô hình: $y_m = \theta^0 u_c$

Sai số: $e = y - y_m = \theta u_c - \theta^0 u_c = (\theta - \theta^0) u_c$

Luật hiệu chỉnh tham số theo phương pháp gradient:

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta} = -\gamma u_c^2 (\theta - \theta^0)$$

Lời giải cho phương trình vi phân ở trên là:

$$\theta(t) = \theta^0 + [\theta(0) - \theta^0] e^{-\gamma t} \quad (*)$$

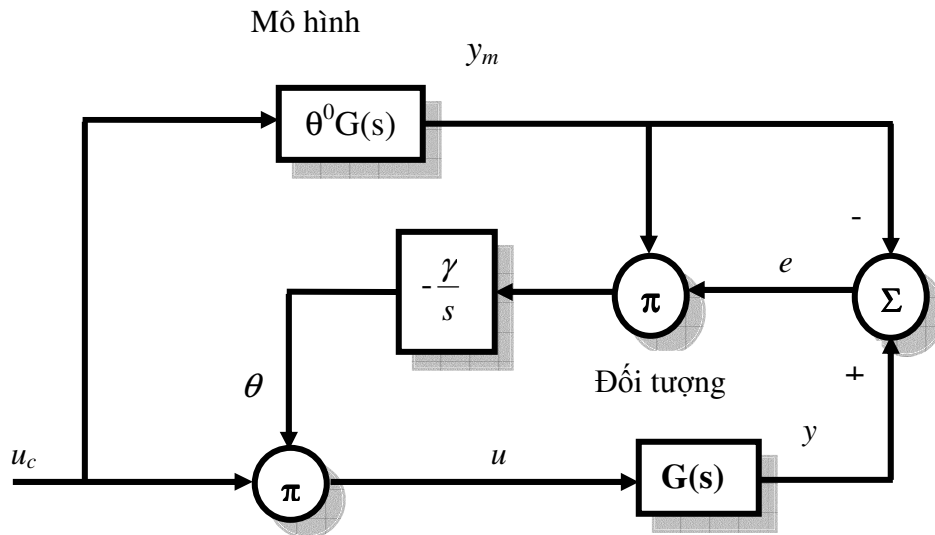
Trong đó:
$$I_t = \int_0^t u_c^2(\tau) d\tau$$

$\theta(0)$ là giá trị ban đầu của θ .

Và vì vậy sai số e trở thành:

$$e(t) = u_c(t) [\theta(0) - \theta^0] e^{-\gamma t}$$

Do $I_t > 0$ nên khi $t \rightarrow \infty$ thì $e(t) \rightarrow 0$ ngay cả khi tín hiệu điều khiển $u_c(t) \rightarrow 0$.



Hình 3.5 Mô hình hội tụ sai số

Giá trị giới hạn của θ phụ thuộc vào tính chất của $u_c(\tau)$ (hội tụ hoặc phân kì) (do $\theta(t)$ tính theo biểu thức (*)).

Ví dụ trên cho biết được sai số $e \rightarrow 0$ tuy nhiên tham số θ không tiến đến giá trị đúng của nó. Đây là tính chất của hệ thống thích nghi sử dụng mô hình chuẩn. Điều kiện chính xác để hội tụ tham số là tín hiệu kích thích phải luôn tồn tại.

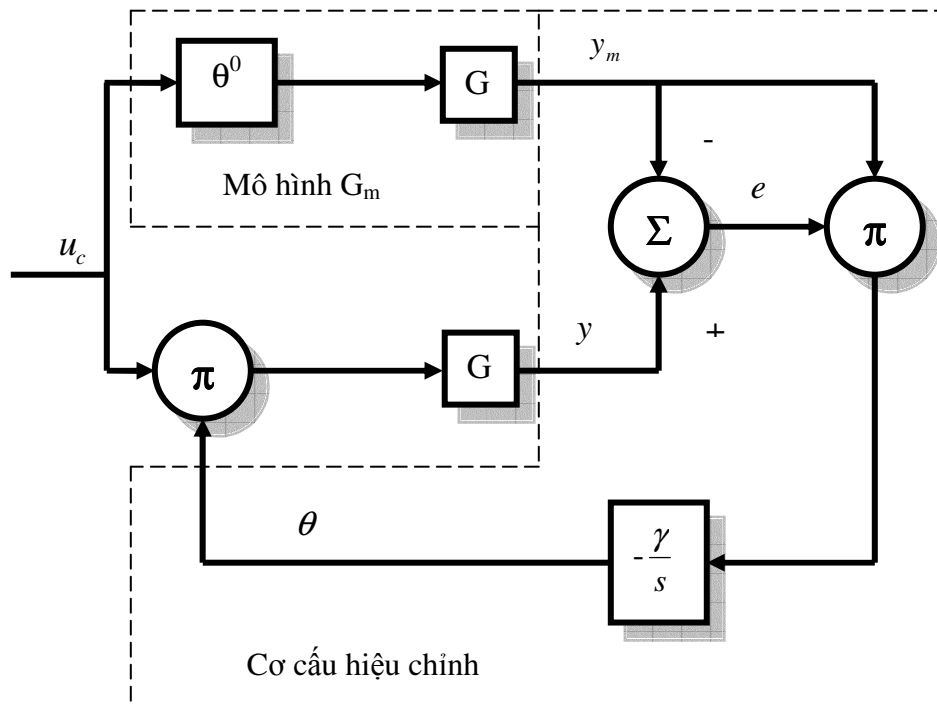
Ổn định của vòng điều khiển thích nghi

Ở ví dụ trên độ biến thiên tham số θ tỉ lệ với bình phương tín hiệu điều khiển u_c . Điều này hợp lí trong một số trường hợp là khi tín hiệu điều khiển u_c càng lớn thì càng dễ phát hiện giá trị bị sai của θ .

Tuy nhiên độ thay đổi của tham số điều chỉnh phụ thuộc vào biên độ của tín hiệu điều khiển có thể dẫn đến không ổn định. Ví dụ sau đây cho luật điều khiển không phụ thuộc vào u_c :

Ví dụ 3.4

Giả sử hệ thống có mô hình ở hình 3.6:



Hình 3.6 Hệ thống thích nghi mô hình tham chiếu cho việc chỉnh định độ lợi nuôi tiến

Vấn đề là điều chỉnh $\theta \rightarrow \theta^0$. Giả sử hàm truyền được cho bởi:

$$G(s) = \frac{1}{s^2 + a_1s + a_2}$$

Sai số $e = G(p)(\theta - \theta^0)u_c$

Trong đó p biểu thị cho phép lấy đạo hàm. Vì vậy:

$$\frac{\partial e}{\partial \theta} = G(p)u_c = \frac{y_m}{\theta^0}$$

Điều chỉnh tham số theo luật MIT:

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta} = -\gamma e \frac{y_m}{\theta^0} = -\gamma e y_m \quad \text{với } \gamma = \frac{\gamma'}{\theta^0}$$

Hệ thống điều khiển thích nghi vì vậy biểu diễn được bởi các phương trình vi phân sau:

$$\frac{d^2 y_m}{dt^2} + a_1 \frac{dy_m}{dt} + a_2 y_m = \theta^0 u_c \quad \text{(I)}$$

$$\frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_2 y = \theta u_c \quad \text{(II)}$$

$$\frac{d\theta}{dt} = -\gamma e y_m = -\gamma(y - y_m)y_m \quad \text{(III)}$$

Phương trình (I) có thể giải được nếu cho sẵn hàm u_c , xem như biến y_m biết trước

Đạo hàm (II) ta được:

$$\frac{d^3 y}{dt^3} + a_1 \frac{d^2 y}{dt^2} + a_2 \frac{dy}{dt} = \frac{d\theta}{dt} u_c + \theta(t) \frac{du_c}{dt}$$

Thay (III) vào ta được:

$$\begin{aligned} \frac{d^3 y}{dt^3} + a_1 \frac{d^2 y}{dt^2} + a_2 \frac{dy}{dt} &= -\gamma(y - y_m)y_m u_c + \theta(t) \frac{du_c}{dt} \\ &= -\gamma y_m(t)u_c(t)y(t) + \gamma y_m^2(t)u_c + \theta(t) \frac{du_c}{dt} \end{aligned}$$

Suy ra:

$$\frac{d^3 y}{dt^3} + a_1 \frac{d^2 y}{dt^2} + a_2 \frac{dy}{dt} + \gamma u_c(t)y_m(t)y(t) = \theta(t) \frac{du_c}{dt} + \gamma u_c(t)y_m^2(t)$$

Đây là phương trình vi phân tuyến tính biến thiên theo thời gian.

Để hiểu được hệ thống, ta thực hiện cách thử như sau:

- Đầu tiên giả sử u_c là hằng số u_c^0
- Ngõ ra mô hình khi đó sẽ có giá trị cân bằng là y_m^0 .

Giả sử cơ cấu hiệu chỉnh thích nghi được nối vào khi đạt đến điểm cân bằng (trạng thái cân bằng). Khi đó phương trình (II) ở trên sẽ có các hệ số hằng và có lời giải trạng thái cân bằng là:

$$y(t) = y_m^0 = \theta^0 u_c^0 / a_2$$

ổn định nếu $a_1 a_2 > \gamma u_c^0 y_m^0 = \frac{\gamma'}{a_2} (u_c^0)^2$

Luật hiệu chỉnh bổ sung

Những hiểu biết có được từ việc tính toán trong ví dụ 3.3 chỉ ra rằng cần phải bổ sung cho luật MIT. Luật MIT là phương pháp gradient cơ bản. Độ giảm có được bằng luật MIT được quyết định bởi tham số γ , số này là do người dùng chọn.

Có thể đạt được phương pháp gradient bổ sung mà tỉ lệ hiệu chỉnh không phụ thuộc vào biên độ của tín hiệu (đặt) yêu cầu. Một khả năng là làm chuẩn hoá và thay thế luật MIT bởi:

$$\frac{d\theta}{dt} = -\gamma \frac{e \frac{\partial e}{\partial \theta}}{\alpha + \left(\frac{\partial e}{\partial \theta}\right)^T \left(\frac{\partial e}{\partial \theta}\right)}$$

Tham số $\alpha > 0$ được đưa vào để tránh trường hợp chia cho 0.

Có thể nhận thấy rằng tỉ lệ hiệu chỉnh tham số phụ thuộc vào biên độ của tín hiệu yêu cầu một lượng nhỏ bởi vì do nhiễu đo lường.

Trình tự giải quyết bài toán điều khiển thích nghi:

- Đặt vấn đề
- Giải thuật
- Thiết kế
- Kết quả mô phỏng
- Luật hiệu chỉnh bổ sung
- Điều kiện hoạt động ổn định
- Kết luận

Một số ví dụ minh họa:

Ví dụ 3.5:

1. Đặt vấn đề:

Xét một quá trình có hàm truyền:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b}{s(s+1)}$$

Trong đó: b : thông số thay đổi theo thời gian.

$Y(s)$: đầu ra quá trình

$U(s)$: đầu vào quá trình

Cần thiết kế bộ điều khiển sao cho hàm truyền đạt của đáp ứng vòng kín hệ thống thể hiện hàm truyền đạt mong muốn:

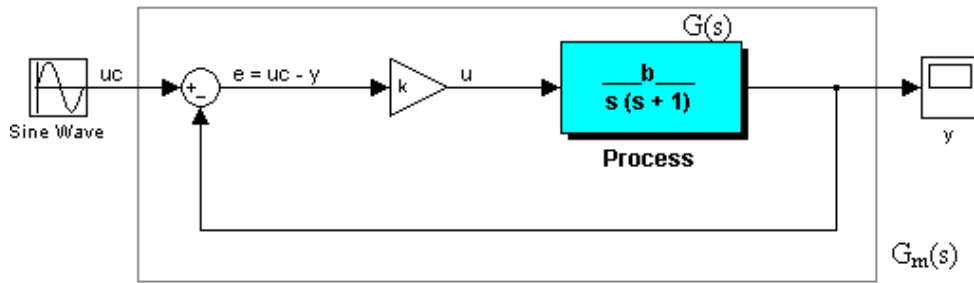
$$G_m(s) = \frac{Y_m(s)}{U_c(s)} = \frac{1}{s^2 + s + 1}$$

Trong đó: $Y_m(s)$: đầu ra mong muốn

$U_c(s)$: đầu vào hệ thống

• **Nếu dùng bộ điều khiển kinh điển:**

Giả sử ta dùng bộ điều khiển P kinh điển để thực hiện yêu cầu trên



Hàm truyền đạt vòng kín của hệ thống:

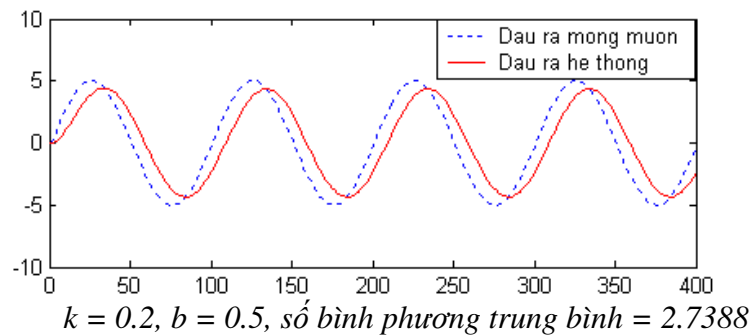
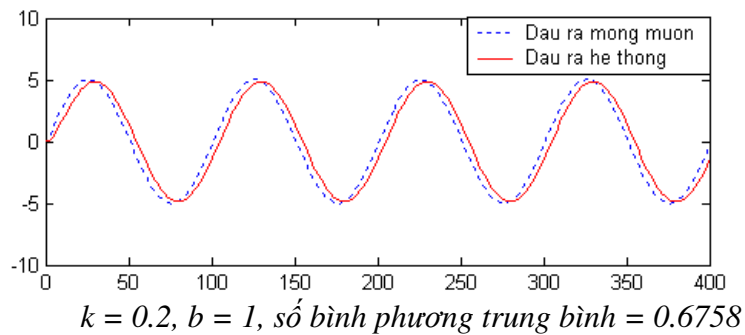
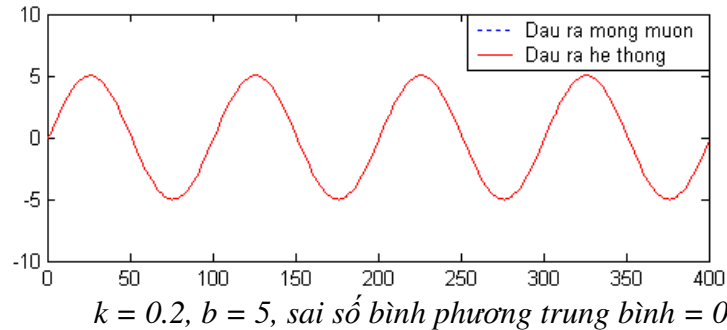
$$G_c(s) = \frac{Y(s)}{U_c(s)} = \frac{kG(s)}{1+kG(s)} = \frac{kb}{s^2 + s + kb}$$

Ta chọn hệ số tỉ lệ k để $G_c(s)$ tiến đến hàm truyền đạt mong muốn $G_m(s)$:

$$G_c(s) = \frac{kb}{s^2 + s + kb} \equiv G_m(s) = \frac{1}{s^2 + s + 1}$$

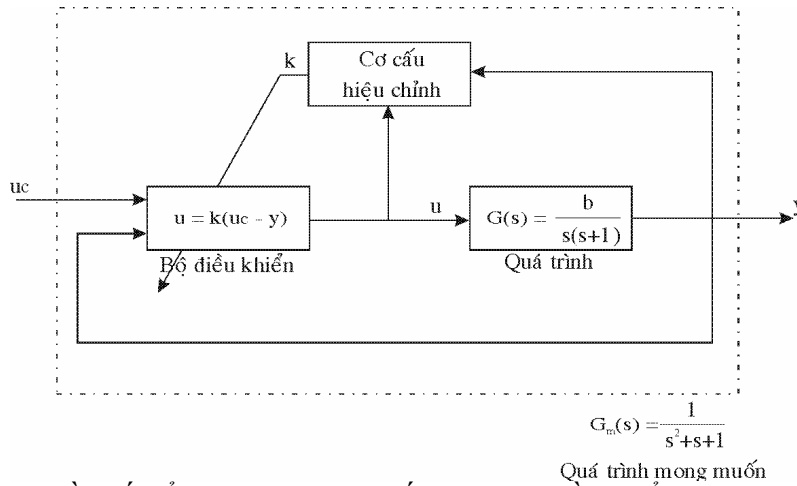
$$\text{Suy ra: } kb = 1 \text{ hay } k = \frac{1}{b}$$

Biểu thức trên chỉ ra rằng để hệ thống đạt được hàm truyền đạt mong muốn thì hệ số k của bộ điều khiển phải liên tục được cập nhật theo thông số b của quá trình. Nếu điều này không thỏa mãn, bộ điều khiển sẽ không làm tốt chức năng điều khiển của nó. Điều này được minh họa bằng kết quả mô phỏng bộ điều khiển P trên với hệ số tỉ lệ $k = 0.2$:



- Như vậy, bộ điều khiển kinh điển không thể tự cập nhật hệ số tỉ lệ k theo thông số b của quá trình. Do đó không thể sử dụng để điều khiển quá trình được. Vấn đề đặt ra là thiết kế một bộ điều khiển sao cho nó có thể thích nghi với quá trình khi thông số b thay đổi theo thời gian trong một khoảng tương đối rộng. Bộ điều khiển thích nghi mô hình tham chiếu (MRAS) có thể giải quyết bài toán này.

Phương pháp thiết kế được chọn là **phương pháp tiếp cận Gradient**.



Sơ đồ khối tổng quát của hệ thống với bộ điều khiển thích nghi

2. Giải thuật:

Hàm truyền đạt vòng hở của quá trình:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b}{s(s+1)} \quad (2.1)$$

$$\Leftrightarrow (s^2 + s)Y(s) = bU(s)$$

$$\Rightarrow (p^2 + p)y = bu \quad (2.2)$$

Trong đó p là toán tử vi phân $\frac{d}{dt}$

Luật điều khiển:

$$u = k(u_c - y) \quad (2.3)$$

Thay (2.3) vào (2.2) ta được:

$$\Rightarrow (p^2 + p)y = bk(u_c - y)$$

$$\Leftrightarrow (p^2 + p + bk)y = bku_c \quad (2.4)$$

$$\Leftrightarrow y = \frac{bk}{p^2 + p + bk} u_c \quad (2.5)$$

Sai số của hệ thống vòng kín:

$$e = y - y_m \quad (2.6)$$

Từ phương trình (2.5) suy ra độ nhạy của sai số theo hệ số tỉ lệ k:

$$\frac{\partial e}{\partial k} = \frac{\partial y}{\partial k} = \frac{b(p^2 + p + bk) - b^2k}{(p^2 + p + bk)^2} u_c$$

$$\Rightarrow \frac{\partial e}{\partial k} = \frac{b(p^2 + p)}{(p^2 + p + bk)^2} u_c \quad (2.7)$$

Do đó, theo MIT, luật cập nhật hệ số tỉ lệ k có dạng:

$$\frac{dk}{dt} = -\gamma' e \frac{\partial e}{\partial k} = -\gamma'(y - y_m) \left[\frac{b(p^2 + p)}{(p^2 + p + bk)^2} u_c \right] \quad (2.8)$$

Phương trình (2.8) không thể sử dụng trực tiếp để cập nhật hệ số tỉ lệ k của bộ điều khiển được do thông số b của quá trình là không biết được. Do đó phải sử dụng phép xấp xỉ để loại bỏ đi thông số chưa biết này.

Hàm truyền đạt mong muốn của hệ thống vòng kín:

$$G_m(s) = \frac{Y_m(s)}{U_c(s)} = \frac{1}{s^2 + s + 1}$$

$$\Leftrightarrow (s^2 + s + 1)Y_m(s) = U_c(s)$$

$$\Leftrightarrow (p^2 + p + 1)y_m = u_c \quad (2.9)$$

$$\Leftrightarrow y_m = \frac{u_c}{p^2 + p + 1} \quad (2.10)$$

Khi hàm truyền đạt của hệ thống đạt tới hàm truyền đạt mong muốn thì phương trình (2.4) sẽ đạt tới phương trình mong muốn (2.9):

$$(p^2 + p + bk)y = bku_c \quad \equiv \quad (p^2 + p + 1)y_m = u_c$$

Hay: $bk = 1$. Do đó phương trình (2.8) có thể xấp xỉ:

$$\frac{dk}{dt} = -\gamma'(y - y_m) \left[\frac{b(p^2 + p)}{(p^2 + p + 1)^2} u_c \right] \quad (2.11)$$

Hơn nữa, đặt $\gamma = \gamma'b$, một lần nữa phương trình (2.11) có thể xấp xỉ:

$$\boxed{\frac{dk}{dt} = -\gamma(y - y_m) \left[\frac{p^2 + p}{(p^2 + p + 1)^2} u_c \right]} \quad (2.12)$$

Phương trình (2.12) đã loại bỏ thông số b qua 2 phép xấp xỉ do đó có thể sử dụng để cập nhật hệ số tỉ lệ k của bộ điều khiển.

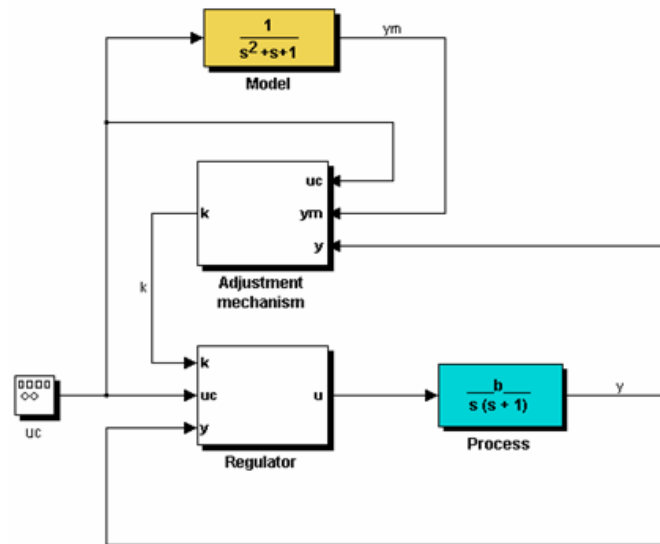
Một câu hỏi đặt ra là 2 phép xấp xỉ này ảnh hưởng như thế nào tới chất lượng điều khiển. Như đã biết, hàm truyền đạt vòng kín của hệ thống chỉ có thể hội tụ về hàm truyền đạt vòng kín mong muốn khi thông số γ được chọn đủ nhỏ. Phép xấp xỉ $\gamma = \gamma'b$ đã gộp thông số b thay đổi theo thời gian vào γ , hay nói cách khác γ cũng trở thành thay đổi theo thời gian.

γ tượng trưng cho tốc độ hội tụ về hàm truyền đạt vòng kín mong muốn của hệ thống (tốc độ thích nghi). Do đó, nếu γ nhỏ (khi thông số b nhỏ), hệ thống sẽ hội tụ chậm. Nếu γ lớn (khi thông số b lớn), tính ổn định của hệ thống sẽ không được bảo đảm và hệ thống sẽ không điều khiển được. Như vậy, bộ điều khiển chỉ có thể thích nghi khi thông số b của quá trình thay đổi trong một giới hạn cho phép.

Phép xấp xỉ $bk = 1$ không ảnh hưởng đáng kể đến chất lượng điều khiển vì một khi hệ thống vòng kín tiến đến hàm truyền đạt mong muốn thì phép xấp xỉ này cũng tiến đến một phép toán chính xác.

Nói tóm lại, tính ổn định của hệ thống phụ thuộc nhiều vào thông số b của quá trình. Việc lựa chọn thông số γ' của bộ điều khiển thích nghi phải căn cứ vào tầm thay đổi của thông số b khi hệ thống hoạt động.

3. Thiết kế:



Các khối chính của bộ điều khiển

Trong đó:

- **Khối Process:** quá trình cần điều khiển

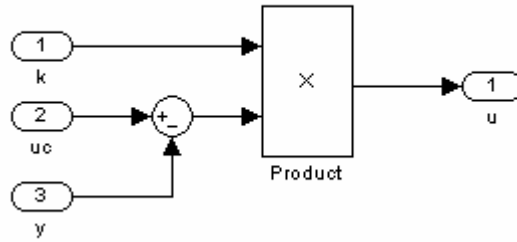
$$G(s) = \frac{Y(s)}{U(s)} = \frac{b}{s(s+1)}$$

- **Khối Model:** hàm truyền đạt mong muốn của hệ thống vòng kín

$$G_m(s) = \frac{Y_m(s)}{U_c(s)} = \frac{1}{s^2 + s + 1}$$

- **Khối Regulator:** luật điều khiển

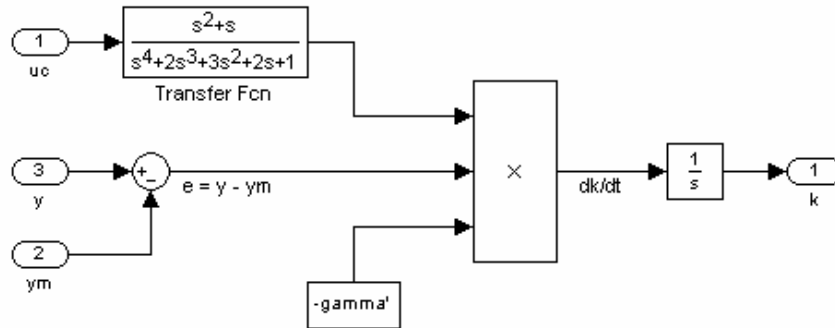
$$u = k(u_c - y)$$



- **Khối Adjustment mechanism:** là khối quan trọng nhất của bộ điều khiển thích nghi, có chức năng hiệu chỉnh hệ số tỉ lệ k của khối **Regulator** theo luật cập nhật thông số MIT

$$\frac{dk}{dt} = -\gamma(y - y_m) \left[\frac{p^2 + p}{(p^2 + p + 1)^2} u_c \right]$$

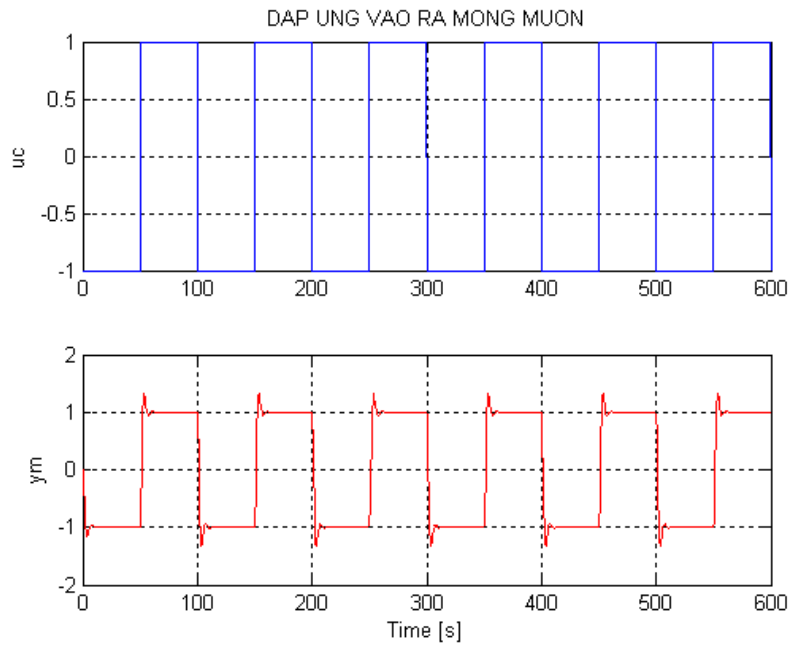
Suy ra:
$$\frac{dk}{dt} = -\gamma(y - y_m) \left[\frac{p^2 + p}{p^4 + 2p^3 + 3p^2 + 2p + 1} u_c \right]$$



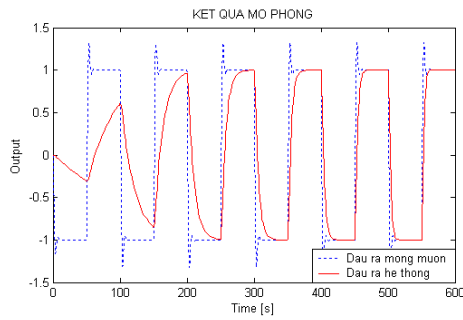
4. Kết quả mô phỏng:

Giả sử tầm thay đổi của b là $[0.1 \div 10]$, tốc độ hội tụ $\gamma' = 0.1$, tín hiệu kích thích u_c là xung vuông lưỡng cực có biên độ ± 1 và tần số 0.01Hz:

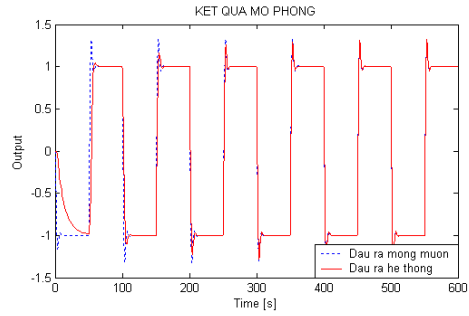
Tín hiệu kích thích đầu vào và đáp ứng đầu ra mong muốn



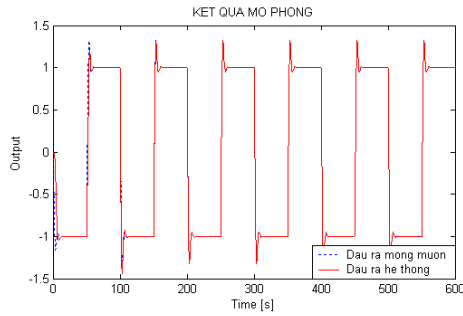
Đáp ứng đầu ra hệ thống khi thông số b thay đổi:



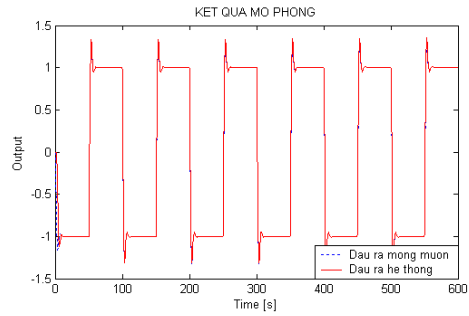
$b = 0.1$



$b = 1$



$b = 5$



$b = 10$

Nhận xét:

Khi b nhỏ thì hệ thống chậm hội tụ về mô hình mong muốn, khi b càng lớn thì tốc độ hội tụ càng tăng lên (Như đã phân tích ở mục 2).

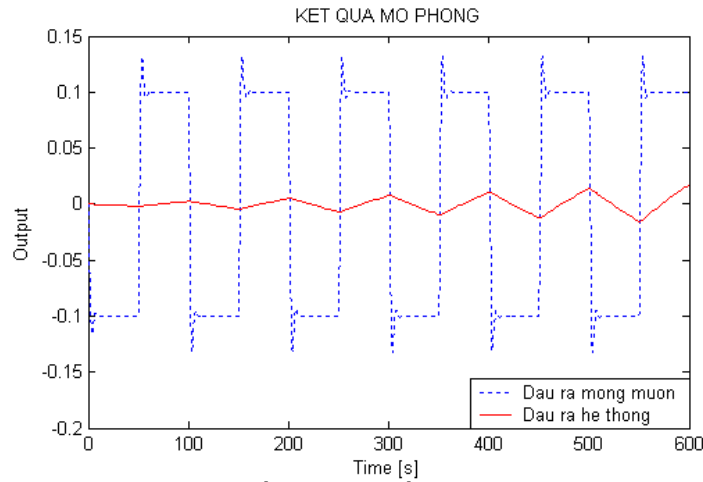
Tuy nhiên khi b tăng đến một mức nào đó hệ thống sẽ bị mất tính ổn định. Điều này sẽ được đề cập ở mục 6.

5. Luật hiệu chỉnh bổ sung:

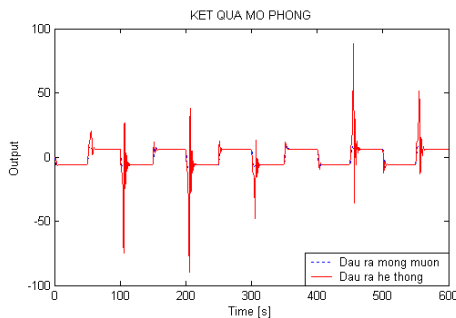
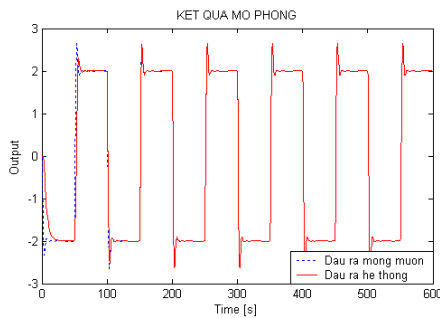
Thay (2.5) và (2.10) vào luật hiệu chỉnh thông số (2.12) ở mục 1 ta được:

$$\frac{dk}{dt} = -\gamma \left(\frac{bk}{p^2 + p + bk} - \frac{1}{p^2 + p + 1} \right) \left[\frac{p^2 + p}{(p^2 + p + 1)^2} \right] u_c^2 \quad (5.1)$$

Phương trình (2.13) cho thấy tốc độ hội tụ của k phụ thuộc vào bình phương biên độ của tín hiệu vào u_c . Đây là hiệu ứng không mong muốn. Điều này có thể minh họa qua việc điều khiển hệ thống với tín hiệu kích thích đầu vào u_c là xung vuông lưỡng cực có biên độ thay đổi trong khoảng $[0.1 \div 6]$, tần số $\gamma' = 0.01\text{Hz}$ và thông số của quá trình không đổi $b = 5$:



Đáp ứng đầu ra hệ thống khi $u_c = \pm 0.1$



Đáp ứng đầu ra hệ thống khi $u_c = \pm 2$ và $u_c = \pm 6$

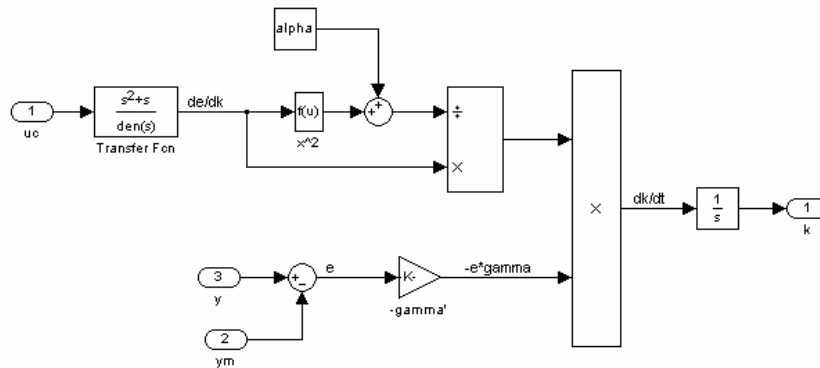
Nhận xét: Khi tín hiệu vào có biên độ nhỏ, tốc độ hội tụ của hệ thống rất chậm, khi biên độ vào tăng lên, tốc độ hội tụ cũng tăng lên (theo tỉ lệ bình phương) tương ứng. Khi biên độ vào đạt đến 6 thì hệ thống gần như không ổn định nữa và xuất hiện các xung vọt lố không khống chế được. Rõ ràng điều này đã hạn chế rất lớn tầm điều khiển của hệ thống. Nhược điểm này có thể khắc phục được bằng luật hiệu chỉnh thông số bổ sung:

$$\frac{dk}{dt} = -\gamma \frac{e \frac{\partial e}{\partial k}}{\alpha + \left(\frac{\partial e}{\partial k}\right)^T \left(\frac{\partial e}{\partial k}\right)} = -\gamma e \frac{\frac{\partial e}{\partial k}}{\alpha + \left(\frac{\partial e}{\partial k}\right)^2} \quad (5.2)$$

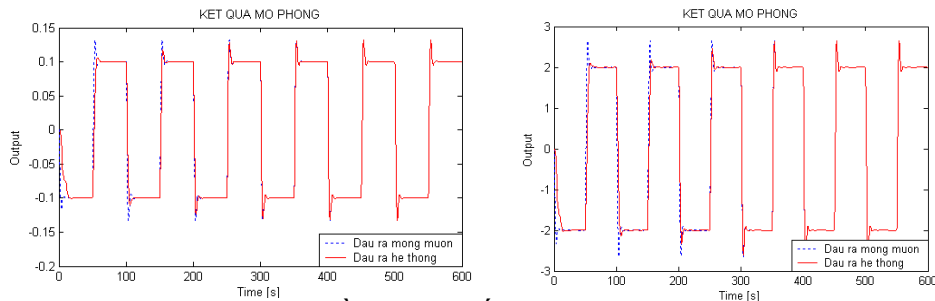
Trong đó: α là một số dương nhỏ tùy ý (chọn $\alpha = 0.00001$) đưa vào để tránh cho mẫu số của phương trình (5.2) bằng 0 và:

$$\frac{\partial e}{\partial k} \approx \frac{p^2 + p}{(p^2 + p + 1)^2} u_c = \frac{p^2 + p}{p^4 + 2p^3 + 3p^2 + 2p + 1} u_c$$

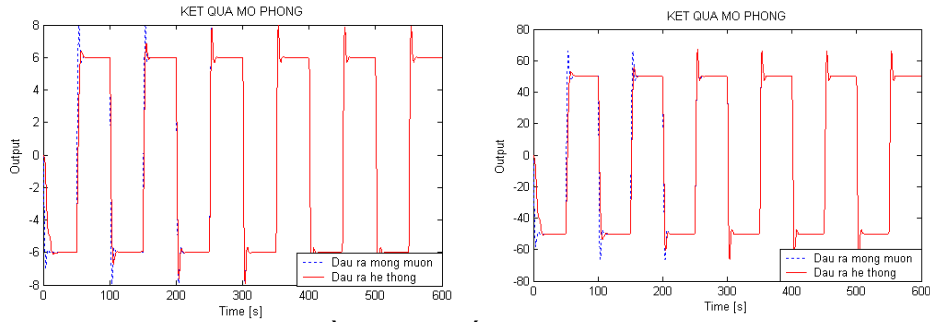
Từ phương trình (5.2), khối **Adjustment mechanism** thiết kế lại như sau:



Kết quả mô phỏng khi u_c thay đổi với luật hiệu chỉnh thông số bổ sung:



Đáp ứng đầu ra hệ thống khi $u_c = \pm 0.1$ và $u_c = \pm 2$



Đáp ứng đầu ra hệ thống khi $u_c = \pm 6$ và $u_c = \pm 50$

Nhận xét:

Rõ ràng khi có luật hiệu chỉnh thông số bổ sung đã cho kết quả tốt và ổn định hơn hẳn. Tốc độ hội tụ về mô hình mong muốn của hệ thống hầu như không còn phụ thuộc vào biên độ tín hiệu vào. Kết quả là về mặt lý thuyết tầm điều khiển của hệ thống có thể mở rộng đến bất kỳ giá trị u_c nào.

6. Điều kiện hoạt động ổn định:

Có 3 yếu tố ảnh hưởng đến tính ổn định của hệ thống đó là:

- Biên độ tín hiệu u_c (khắc phục bằng luật hiệu chỉnh thông số)
- Tốc độ hội tụ thật sự γ
- Nhiễu tác động vào hệ thống

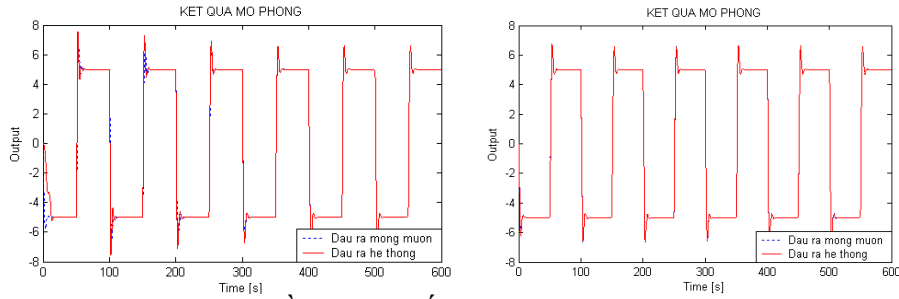
a. Giới hạn của tích số $\gamma = \gamma' b$:

Nếu γ quá lớn có thể làm hệ thống mất ổn định. Do đó, việc lựa chọn γ' tùy thuộc vào tầm thay đổi của b sao cho $\gamma = \gamma' b$ vẫn còn đủ nhỏ khi b đạt đến giá trị cực đại. Giả sử hệ thống không chịu sự tác động của nhiễu, chọn $\gamma' = 1$ và cho thông số b tăng dần, đến khi b đạt đến giá trị $b_{\max} = 2.5$ thì hệ thống bắt đầu mất ổn định hay nói cách khác, khi:

$$\gamma = \gamma' b \leq 2.5 * 0.8 = 2 = \gamma_{\max} \quad (0.8 \text{ là hệ số an toàn}) \quad (6.1)$$

thì hệ thống vẫn hoạt động ổn định trong trường hợp không có nhiễu.

Dựa vào (6.1) ta có thể chọn tốc độ hội tụ γ' thích hợp với sự thay đổi của thông số quá trình. Chẳng hạn nếu biết được khi vận hành, thông số b của quá trình đạt giá trị cực đại $b_{\max} = 50$ có thể chọn $\gamma' = \frac{2}{b_{\max}} = 0.04$ để đảm bảo tính ổn định của việc điều khiển.



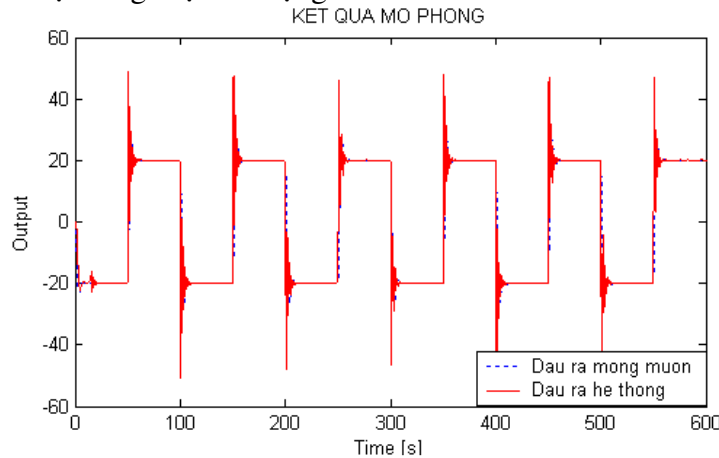
Đáp ứng đầu ra hệ thống khi $b = 1$ và $b_{max} = 50$ $\gamma' = 0.04$

b. Hệ thống chịu tác động của nhiễu:

Thật khó để đưa ra một ước lượng chính xác để đánh giá tác động của nhiễu đối với tính ổn định của hệ thống. Một cách tương đối, khi tốc độ hội tụ γ' càng nhỏ thì hệ thống càng bền vững với nhiễu (và yếu tố đánh đổi là hệ thống chậm hội tụ về hàm truyền mong muốn). Do đó, khi có nhiễu tác động vào hệ thống, chỉ có thể chọn γ' bằng cách thử - sai: cho thông số b đạt giá trị b_{max} và tăng dần γ' đến giá trị γ'_{max} (khi hệ thống đạt đến biên giới ổn định). Nếu chọn $\gamma' < 0.8\gamma'_{max}$ sẽ đảm bảo tính ổn định của hệ thống.

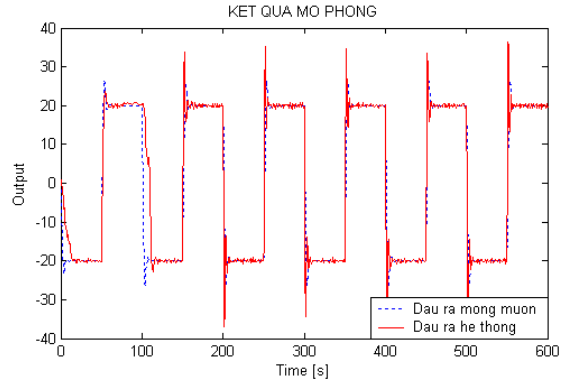
Giả sử nhiễu tác động vào hệ thống có kỳ vọng = 0.5 và phương sai = 0.05, hệ thống khi hoạt động có $b_{max} = 10$, tín hiệu kích thích đầu vào u_c là xung vuông lưỡng cực biên độ ± 20 , tần số 0.01Hz. Cần xác định giá trị γ'_{max} của bộ điều khiển thích nghi sao cho hệ thống vẫn ổn định.

- Hệ thống chịu tác động của nhiễu đầu vào:



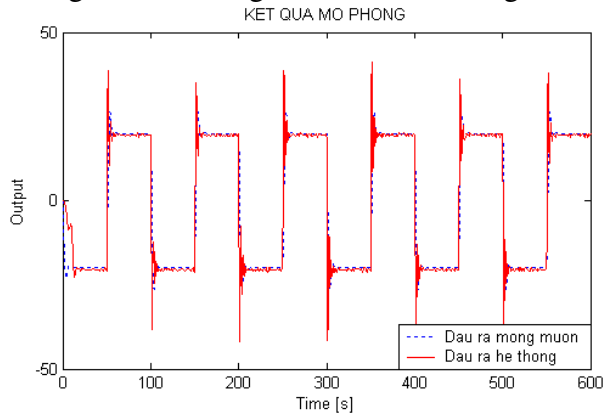
$\gamma'_{max} = 0.005$

- Hệ thống chịu tác động của nhiễu đầu ra:



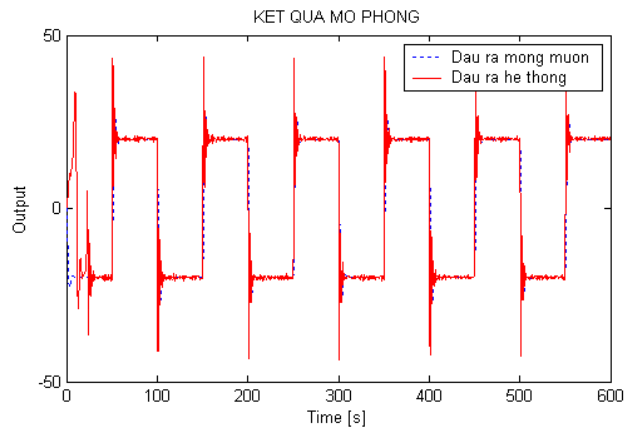
$$\gamma'_{\max} = 0.002$$

- Hệ thống chịu tác động của nhiễu đo lường:



$$\gamma'_{\max} = 0.0025$$

Hệ thống chịu tác động của nhiễu đầu vào, nhiễu đầu ra và nhiễu đo lường:



$$\gamma'_{\max} = 0.0008$$

7. Kết luận:

Bộ điều khiển thích nghi mô hình tham chiếu (MRAS) với phương pháp tiếp cận gradient có thể giải quyết tốt vấn đề điều khiển hệ thống khi thông số hệ thống thay đổi. Hơn nữa, với luật hiệu chỉnh thông số bổ sung thay thế cho luật hiệu chỉnh thông số MIT có thể nâng cao hơn nữa tính ổn định và tầm điều khiển của hệ thống. Tuy bộ điều khiển là thích nghi nhưng tính thích nghi chỉ có thể thỏa mãn trong một điều kiện làm việc giới hạn, phải chấp nhận đánh đổi giữa tính ổn định của hệ thống với tốc độ hội tụ về giá trị đúng của thông số cần điều khiển.

Ví dụ 3.6:

1. Đặt vấn đề:

Cho hệ thống:

$$G(s) = \frac{1}{s(s+a)} = \frac{B}{A} \quad \text{Với } a \text{ là thông số chưa biết.} \quad (1)$$

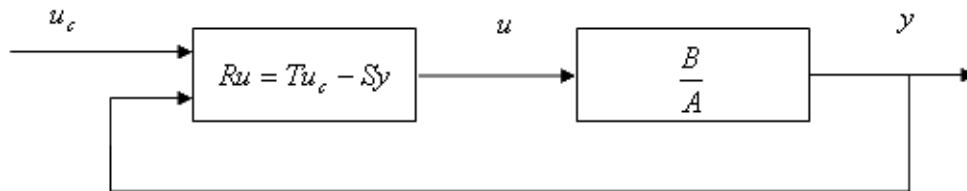
Xác định bộ điều khiển có thể cho hệ thống vòng kín sau:

$$G_m(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} = \frac{B_m}{A_m} \quad (2)$$

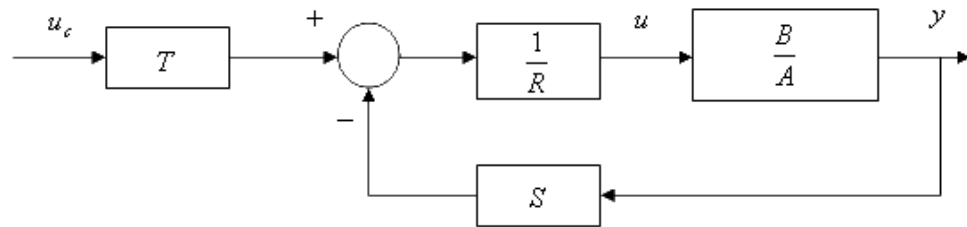
Xác định các bộ điều kiện thích nghi mô hình mẫu dựa trên phương pháp gradient và lý thuyết ổn định.

2. Giải thuật:

Luật điều khiển tuyến tính có dạng: $Ru = Tu_c - Sy$ (3)



với R, S, T là các đa thức. Có thể vẽ lại như sau:



Ta có: $y = u \frac{B}{A} \Rightarrow u = y \frac{A}{B}$ Thay u vào (3), ta được:

$$(AR + BS)y = BTu_c \quad (4)$$

Ta có: $B = B^+ B^- = 1 \Rightarrow B^+ = B^- = 1$

Để có hệ thống vòng kín như mong muốn thì $(AR + BS)$ phải chia hết cho A_m , đa thức này phải chứa $A_m B^+$ và có bậc lớn hơn bằng bậc của $A_m B^+$.

Phương trình Diophantine:

$$AR + BS = B^+ A_0 A_m \quad (5)$$

Với $R = B^+ R_1$. Khử B^+ trong (5) ta được:

$$AR_1 + B^- S = A_0 A_m \quad (6)$$

$$s(s+a)R_1 + S = A_0(s^2 + 2\zeta\omega s + \omega^2)$$

Ta chọn:

$$\begin{cases} R_1 = 1 \Rightarrow R = B^+ R_1 = 1 \\ A_0 = 1 \\ S = s_0 s + \omega^2 \end{cases}$$

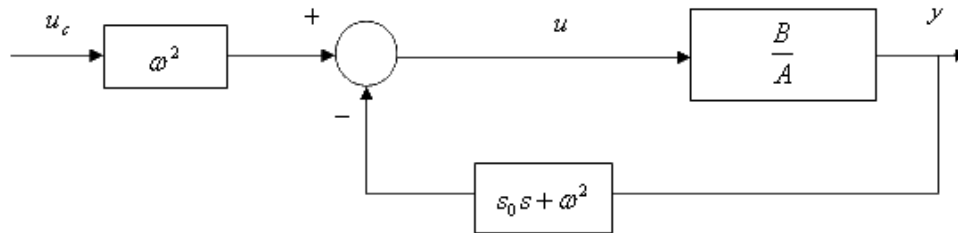
$$\Rightarrow s^2 + as + s_0 s + \omega^2 = s^2 + 2\zeta\omega s + \omega^2 \Rightarrow a + s_0 = 2\zeta\omega$$

Đề (4) có dạng (2) thì:

$$\begin{cases} B_m = B^- B'_m = \omega^2 \\ T = A_0 B'_m \end{cases}$$

Với: $B^- = 1 \Rightarrow B'_m = \omega^2$ và $T = A_0 B'_m = \omega^2$

Như vậy, ta có bộ điều khiển tuyến tính như sau:



Luật điều khiển có dạng:

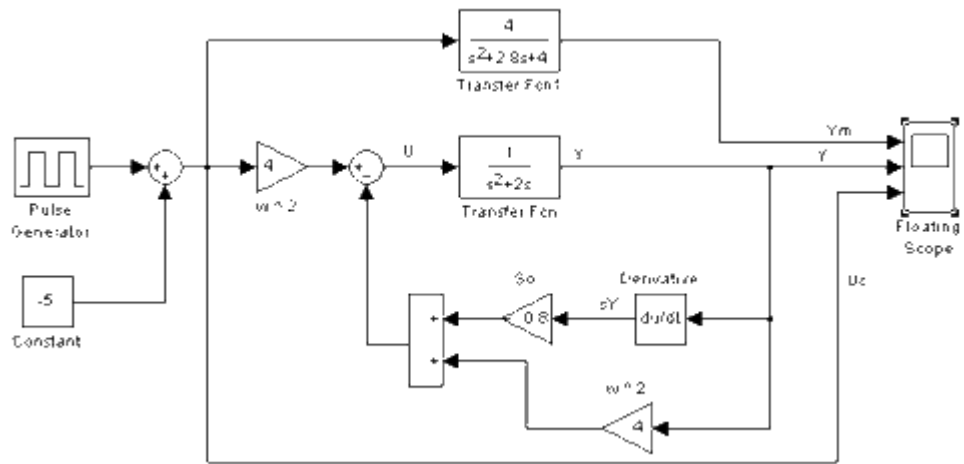
$$u = \omega^2 u_c - (s_0 s + \omega^2) y$$

Phương trình của hệ kín:

$$(AR + BS)y = BTu_c \Rightarrow (s^2 + as + s_0 s + \omega^2)y = \omega^2 u_c$$

$$\Rightarrow y = \frac{\omega^2}{s^2 + (a + s_0)s + \omega^2} u_c$$

Thiết kế theo mô hình cổ điển:

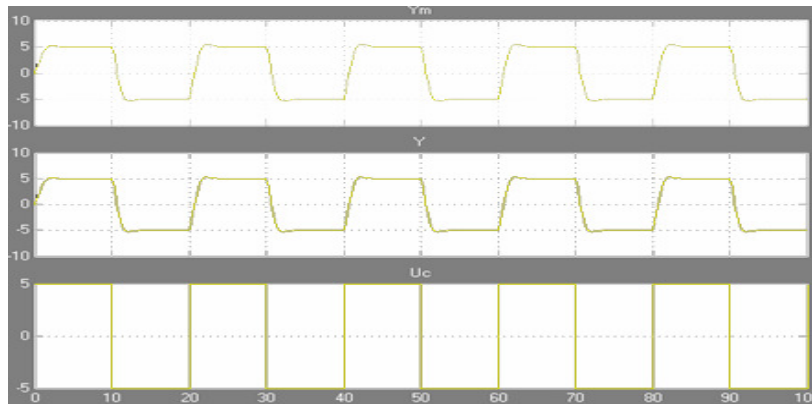


Nếu tham số a của hệ thống được biết trước, ta có thể tính được s_0 theo công thức sau: $s_0 = 2\zeta\omega - a$.

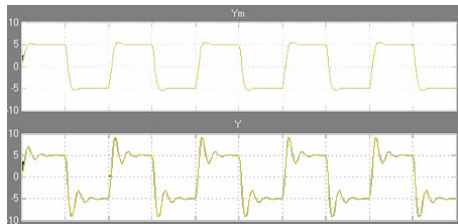
Ví dụ $a = 2$, và giả sử hệ thống mong muốn có $\zeta = 0,7$ và $\omega = 2$

$$\Rightarrow s_0 = 2\zeta\omega - a = 0.8$$

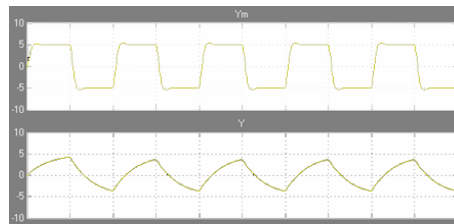
Với $a = 2$



Với $a = 0.2$



Với $a = 20$



Nhận xét:

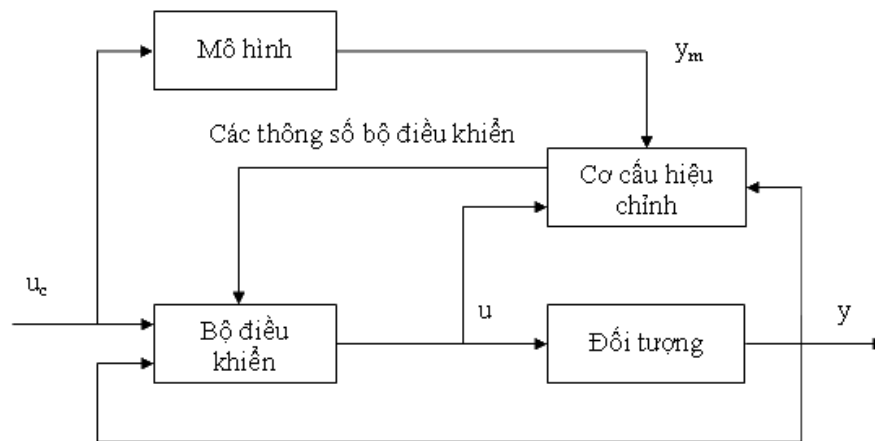
Khi $a = 2$, $s_0 = 0.8$: bộ điều khiển cho đáp ứng hệ thống đúng như yêu cầu. Nhưng trong thực tế thông số a thay đổi theo thời gian, làm cho bộ điều khiển không cho kết quả như mong muốn (với $a = 0.2$, hệ thống bị dao động và với $a = 20$, hệ thống không đáp ứng kịp).

Như vậy, bộ điều khiển tuyến tính cổ điển không thể tự cập nhật hệ số s_0 theo thông số a của hệ thống nên không thể sử dụng để điều khiển quá trình có thông số thay đổi theo thời gian.

Bộ điều khiển thích nghi sẽ giải quyết được bài toán này. Ở đây ta xem xét bộ điều khiển thích nghi mô hình tham chiếu (MRAS).

3. Thiết kế:

Hệ thống thích nghi mô hình tham chiếu



Trong hệ thống, **cơ cấu hiệu chỉnh** sẽ thay đổi các thông số của **bộ điều khiển** sao cho ngõ ra y của **đối tượng** giống với ngõ ra y_m của **mô hình**. Ta sử dụng **phương pháp gradient** để xây dựng luật cập nhật thông số cho cơ cấu hiệu chỉnh.

Định nghĩa sai số:

$$e = y - y_m$$

Ta phải thay đổi thông số của bộ điều khiển sao cho sai số e này về 0.

Xét chuẩn:

$$J(\theta) = \frac{1}{2} e^2$$

Để làm cho J nhỏ thì chiều thay đổi của thông số là chiều âm giá số của J , như sau:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta}$$

Với hệ đã cho, ta có:

$$\frac{\partial e}{\partial s_0} = \frac{-\omega^2 s}{[s^2 + (a + s_0)s + \omega^2]^2} u_c = \frac{-s}{s^2 + (a + s_0)s + \omega^2} y$$

Khi hàm truyền đạt của hệ thống đạt tới hàm truyền đạt mong muốn thì:

$$s^2 + (a + s_0)s + \omega^2 \approx s^2 + 2\zeta\omega s + \omega^2$$

$$\text{hay } a + s_0 \approx 2\zeta\omega \Rightarrow s_0 \approx 2\zeta\omega - a$$

Do đó:
$$\frac{\partial e}{\partial s_0} = \frac{-s}{s^2 + 2\zeta\omega s + \omega^2} y$$

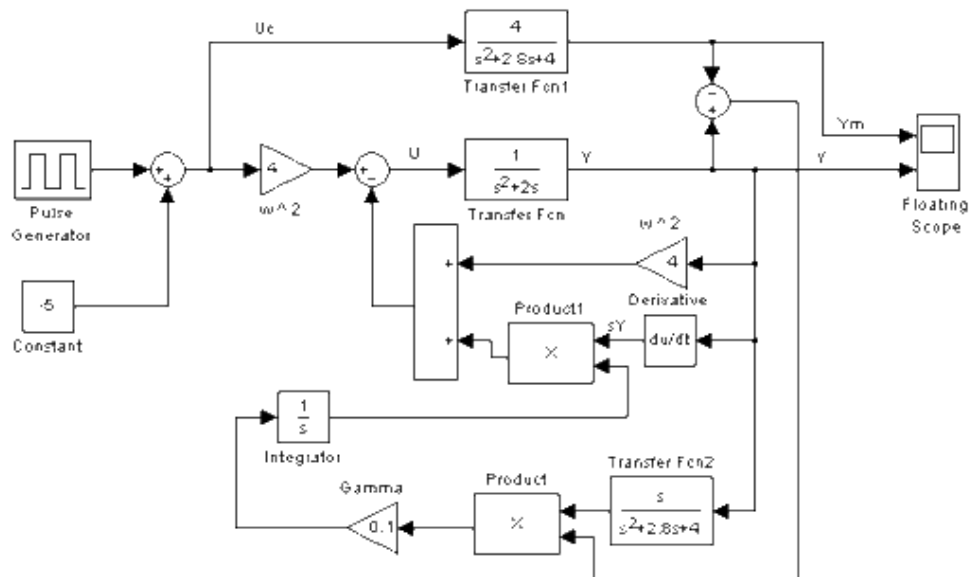
Suy ra:

$$\frac{ds_0}{dt} = -\gamma e \frac{\partial e}{\partial s_0} = \gamma e \frac{s}{s^2 + 2\zeta\omega s + \omega^2} y$$

Tương tự ví dụ 3.5, tính ổn định hay không của hệ thống phụ thuộc nhiều vào thông số a của quá trình. Việc lựa chọn thông số γ của bộ điều khiển thích nghi phải căn cứ vào tầm thay đổi của thông số a khi hệ thống hoạt động.

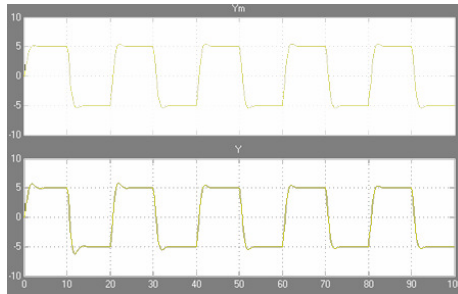
4. Kết quả mô phỏng:

Thiết kế theo mô hình thích nghi

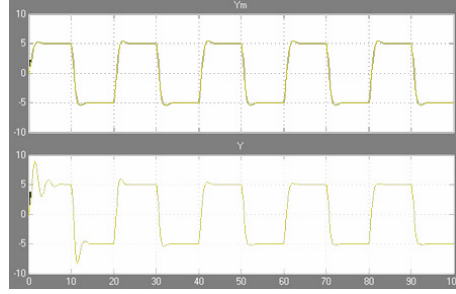


Kết quả mô phỏng thu được:

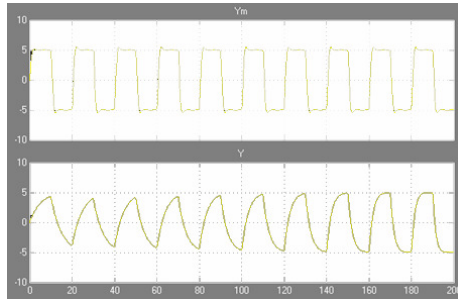
Với $\gamma = 0.1, a = 2$:



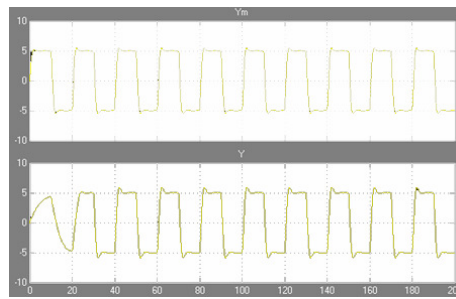
Với $\gamma = 0.1, a = 0.2$:



Với $\gamma = 0.1, a = 20$:



Với $\gamma = 1, a = 20$:



5. Kết luận:

Bộ điều khiển thích nghi chỉ cho kết quả tốt trong một điều kiện làm việc giới hạn, phải chấp nhận đánh đổi giữa tính ổn định của hệ thống với tốc độ hội tụ về giá trị đúng của thông số cần điều khiển.

Ví dụ 3.7:

1. Đặt vấn đề:

Hệ thống đã cho có dạng:

$$y = \frac{B}{A} .u = \frac{b.q}{(s + a)(s + p)} .u \quad (1)$$

Thiết kế bộ điều khiển sao cho hệ thống mong muốn :

$$y_m = \frac{B_m}{A_m} .u_c = \frac{\omega^2}{s^2 + 2.\xi.\omega.s + \omega^2} .u_c \quad (2)$$

2. Giải thuật:

Ta có thỏa điều kiện :

- bậc (A) = bậc (A_m)
- Bậc (B) = bậc (B_m)
- Bậc (A₀) = bậc (A) - bậc (B⁺) - 1 = 2-0-1=1

Vậy ta chọn: $A_0(s) = s + a_0$

Cấu trúc bộ điều khiển theo phương pháp đặt cực có dạng:

$$\mathbf{R.u} = \mathbf{T.u}_c - \mathbf{S.y} \quad (3)$$

Khi đó phương trình PT Diophantine có dạng:

$$A.R + B.S = A_m.A_0 \quad (4)$$

Trong đó ta có: $A = (s+a)(s+p)$ $B = b.q$

$$A_m = s^2 + 2.\xi.\omega.s + \omega^2 \quad B_m = \omega^2$$

Và chọn các đa thức: $R(s) = s + r_1$ $A_0(s) = s + a_0$

$$S(s) = s_0.s + s_1 \quad T(s) = t_0$$

Khi đó (4) sẽ là:

$$(s+a)(s+p)(s+r_1) + b.q.(s_0.s + s_1) = (s^2 + 2.\xi.\omega.s + \omega^2)(s+a_0) \quad (5)$$

$$\Leftrightarrow s^3 + (r_1 + p + a).s^2 + (p.r_1 + a.r_1 + a.p + b.q.s_0).s + a.p.r_1 + b.q.s_1 =$$

$$s^3 + (a_0 + 2.\xi.\omega).s^2 + (2.\xi.\omega.a_0 + \omega^2).s + \omega^2.a_0$$

Đồng nhất thức 2 vế theo s^3, s^2, s^1, s^0 ta được:

$$\Leftrightarrow \begin{cases} (r_1 + p + a) = (a_0 + 2.\xi.\omega) \\ (p.r_1 + a.r_1 + a.p + b.q.s_0) = (2.\xi.\omega.a_0 + \omega^2) \\ a.p.r_1 + b.q.s_1 = \omega^2.a_0 \end{cases}$$

$$\Rightarrow \begin{cases} r_1 = 2.\xi.\omega + a_0 - a - p \\ s_0 = \frac{2.\xi.\omega^2 - (a_0 - p - a) + \omega^2 + a^2 + a.p + p^2 - (p+a).a_0}{b.q} \\ s_1 = \frac{a_0.\omega^2 - 2.\xi.\omega.a.p - a.a_0.p + a^2.p + a.p^2}{b.q} \end{cases}$$

Mặt khác ta có đa thức:

$$T = \frac{A_0.B_m}{B} \Rightarrow T(s) = \frac{\omega^2.(s+a_0)}{b.q} = \frac{\omega^2}{b.q}.s + \frac{\omega^2.a_0}{b.q}$$

Mà: $T(s) = t_0.s + t_1$

Đồng nhất thức ta có: $t_0 = \frac{\omega^2}{b.q}$; $t_1 = \frac{\omega^2.a_0}{b.q}$

Thay R, T và S vào (3) ta được bộ điều khiển của hệ thống có dạng:

$$\begin{aligned} (s + r_1).u(s) &= t_0.u_c(s) - (s_0.s + s_1).y(s) \\ \Leftrightarrow s.u(s) + r_1.u(s) &= t_0.u_c(s) - s_0.s.y(s) - s_1.y(s) \\ \Leftrightarrow \frac{du}{dt} + r_1.u &= t_0.uc - s_1.y - s_0.\frac{dy}{dt} \end{aligned} \quad (6)$$

Có 4 thông số mà ta cần xác định r_1, t_0, s_0, s_1 mà ta cần xác định

Vì tham số p đã biết nên ta chọn đa thức:

$$A_0(s) = s + p$$

Và thay $s = -p$ vào phương trình (5) ta sẽ có

$$\begin{aligned} -s_0.p + s_1 &= 0 \\ \Rightarrow s_1 &= p.s_0 \end{aligned}$$

Trong trường hợp này ta có tính s_1 khi biết $t_0, s_0,$ và r_1 . tuy nhiên tham số q đã biết thường xuất hiện với tham số không biết b

Với $a_0 = p$ từ phương trình (5) ta có

$$(s+a)(s+r_1) + b.q.s_0 = s^2 + 2.\xi.\omega.s + \omega^2$$

Ta có :

$$A'(s) = s+a$$

$$S'(s) = s_0$$

$$T' = t_0$$

Ta có hệ thống vòng kín được mô tả bởi

$$y = \frac{B.T'}{A'.R + B.S'} . u_c$$

Thay y vào (3) ta tính được:

$$u = \frac{\left(T - \frac{S.B.T'}{A.R + B.S'} \right) u_c}{R} = \frac{A.T'}{A.R + B.S'} . u_c$$

3. Thiết kế:

Sai số là : $e = y - y_m$

Bây giờ cần phải xác định các đạo hàm riêng của sai số đối với từng tham số hiệu chỉnh để tìm luật chỉnh định thông số các hàm độ nhạy.

Thay y và y_m vào công thức $e = y - y_m$ và tính đạo hàm riêng theo từng biến t_0, s_0, r_0 ta sẽ có.

$$\frac{\partial e}{\partial t_0} = \frac{B}{A'.R + B.S'} . u_c \approx \frac{B}{A_m} . u_c$$

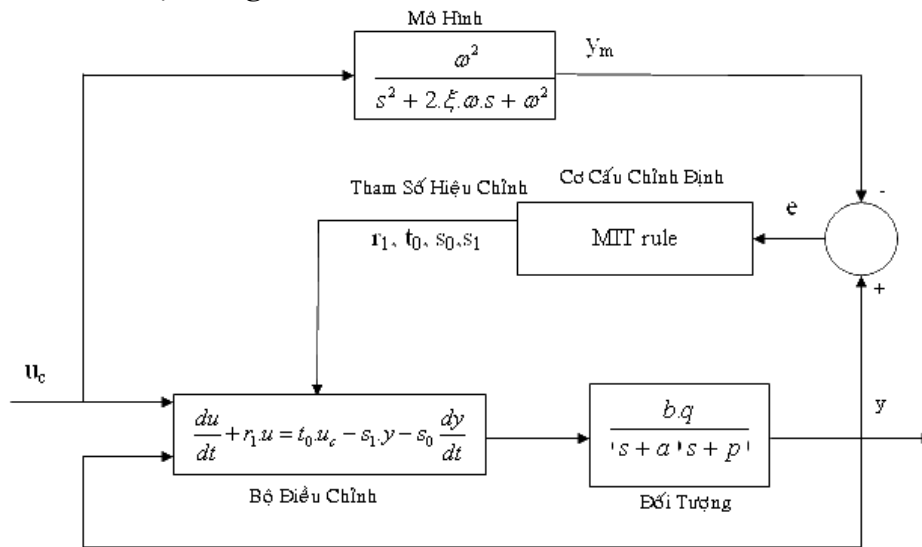
$$\frac{\partial e}{\partial s_0} = -\frac{B.T'.B}{(A'.R + B.S')^2} . u_c = -\frac{B}{A'.R + B.S'} . y \approx -\frac{B}{A_m} . y$$

$$\frac{\partial e}{\partial r_1} = -\frac{A'.B.T'}{(A'.R + B.S')^2} . u_c = -\frac{A'}{A'.R + B.S'} . y \approx -\frac{A'}{A_m} . y = -\frac{A'}{A_m} . \frac{B}{A} . u = -\frac{B}{A_m(s+p)} . u$$

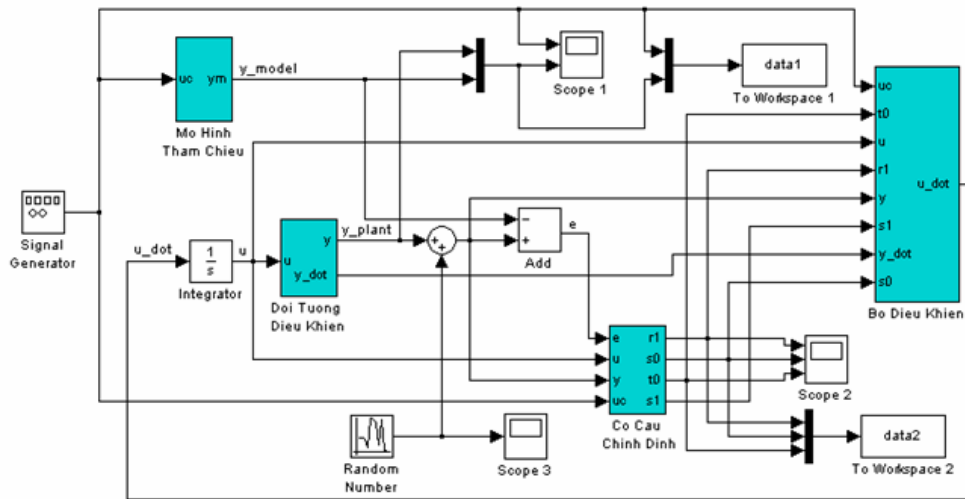
Vậy ta có luật hiệu chỉnh MIT rule như sau :

$$\begin{aligned} \frac{dr_1}{dt} &= \gamma \cdot \left(\frac{1}{(s+p).A_m} \cdot u \right) \cdot e \\ \frac{ds_0}{dt} &= \gamma \cdot \left(\frac{1}{A_m} \cdot y \right) \cdot e \\ \frac{dt_0}{dt} &= -\gamma \cdot \left(\frac{1}{A_m} \cdot u_c \right) \cdot e \end{aligned} \quad (7)$$

Sơ đồ khối hệ thống

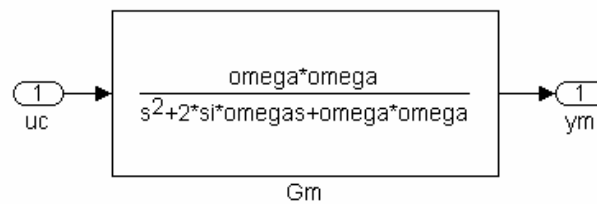


4. Kết quả mô phỏng:

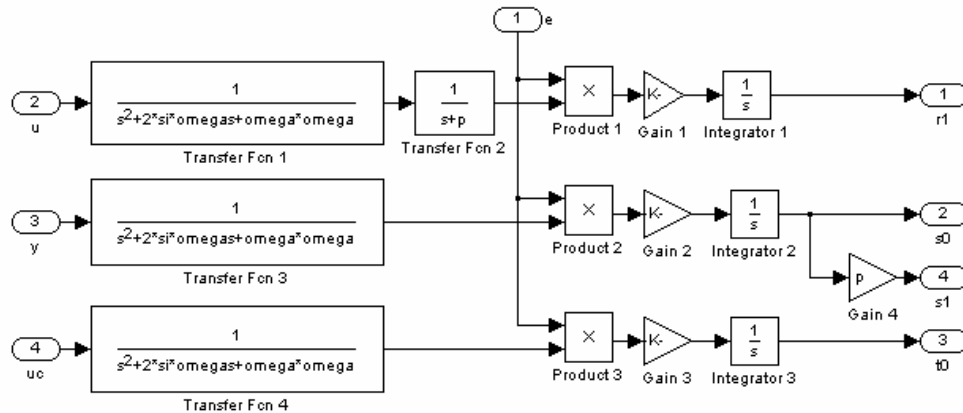


Sơ đồ khối mô phỏng hệ thống theo luật MIT

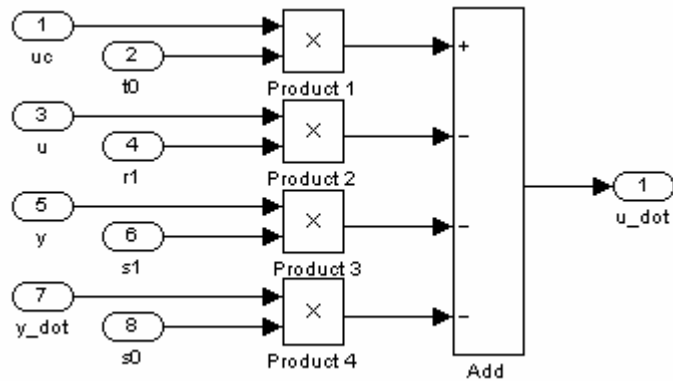
Trong đó khối mô hình tham chiếu có dạng sau:



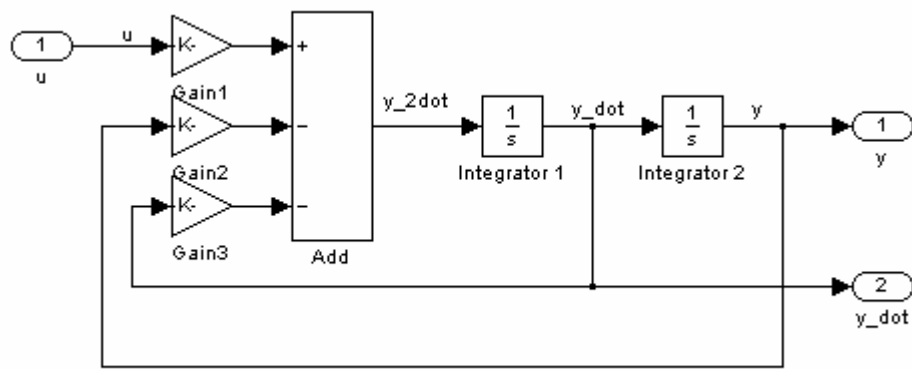
Khối cơ cấu hiệu chỉnh có dạng sau:



Khối bộ điều khiển có dạng sau:



Khối đối tượng điều khiển có dạng sau:

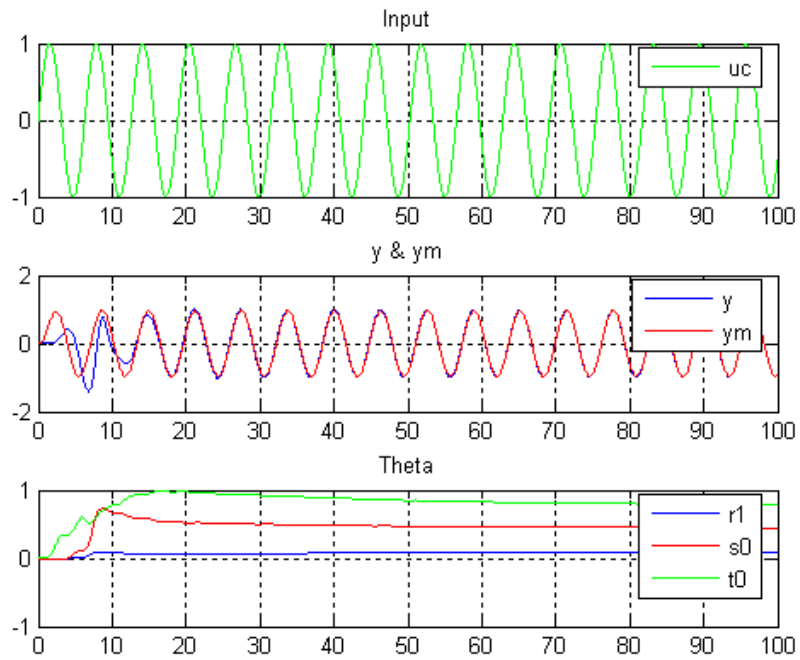


***Trường hợp không có nhiễu:**

Chọn các tham số điều chỉnh như sau:

- Chọn $T_s = 3s$, $\xi = 0.707$, $\gamma = 1$, $\omega = \frac{4}{(T_s * \xi)}$
- Chọn $p = 2$ và $q = (3/2) * p$
- Chọn a,b thay đổi theo tỷ lệ $b = 3 * a$ với $a = 1$

Khi đó ta có kết quả mô phỏng như sau:



Mit Rule for $\gamma=1$ and Amplitude = ± 1

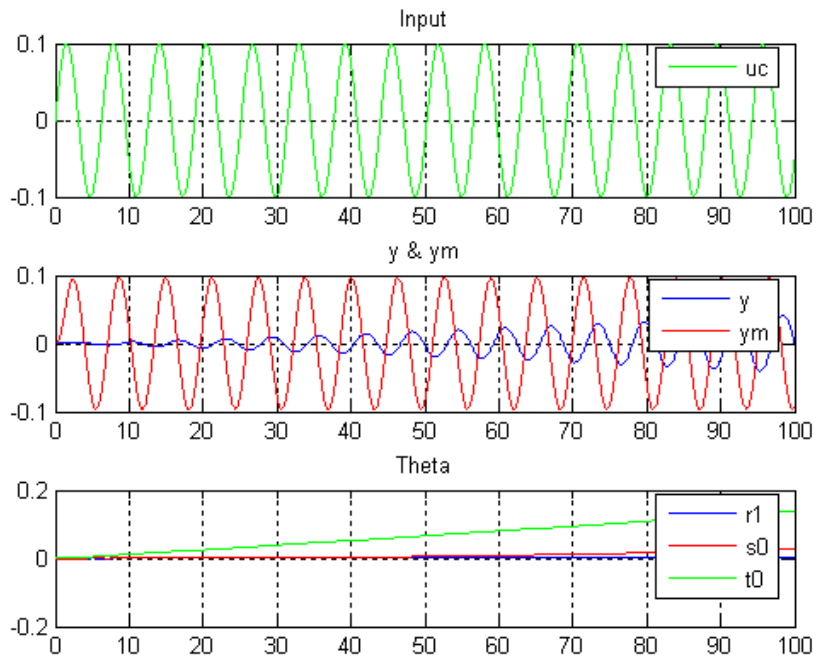
Nhận Xét:

+Trong trường hợp biên độ bằng ± 1 thì luật MIT cho kết quả tương đối tốt .Sai số giữa ngõ ra của đối tượng và của mô hình chuẩn tiến đến Zero.

Chọn các tham số điều chỉnh như sau:

- Chọn $T_s = 3s$, $\xi = 0.707$, $\gamma=1$, $\omega = \frac{4}{(T_s * \xi)}$
- Chọn $p = 2$ và $q = (3/2)*p$
- Chọn a,b thay đổi theo tỷ lệ $b=3*a$ với $a=1$

Khi đó ta có kết quả mô phỏng như sau:



Mit Rule for $\gamma=1$ and Amplitude = ± 0.1

Nhân Xét:

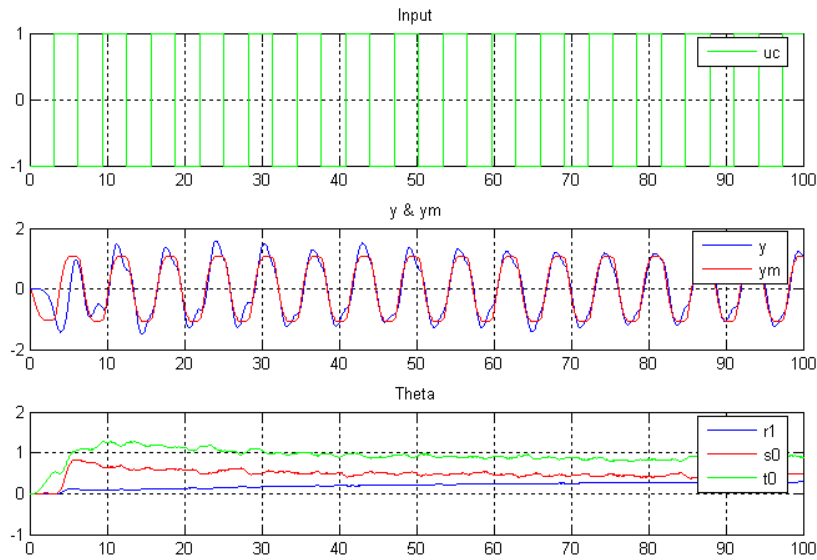
Trong trường hợp biên độ bằng ± 0.1 thì luật MIT cho kết quả không tốt, phụ thuộc vào biên độ của tín hiệu (đặt) yêu cầu. Sai số giữa ngõ ra của đối tượng và của mô hình chuẩn không tiến đến Zero.

***Trường hợp có nhiễu:**

Chọn các tham số điều chỉnh như sau:

- Chọn Mean = 0.01, Var = 0.5
- Chọn $T_s = 3s$, $\xi = 0.707$, $\gamma=1$, $\omega = \frac{4}{(T_s * \xi)}$
- Chọn $p = 2$ và $q = (3/2)*p$
- Chọn a,b thay đổi theo tỷ lệ $b=3*a$ với $a=1$

Khi đó ta có kết quả mô phỏng như sau:



Mit Rule for $\gamma=1$ and Amplitude = ± 1

Nhận Xét:

+Trong trường hợp có nhiễu thì luật MIT bị ảnh hưởng của nhiễu và nó không thể loại được nhiễu.

+Thông số γ xác định tốc độ thích.

+Luật MIT có thể thực hiện tốt nếu độ lợi thích nghi γ là nhỏ. Độ lớn γ tùy thuộc vào biên độ của tín hiệu chuẩn và độ lợi của đối tượng.

+Độ giảm có được bằng luật MIT được quyết định bởi tham số γ , số này do người dùng chọn.

5. Luật hiệu chỉnh bổ sung:

Ta có:

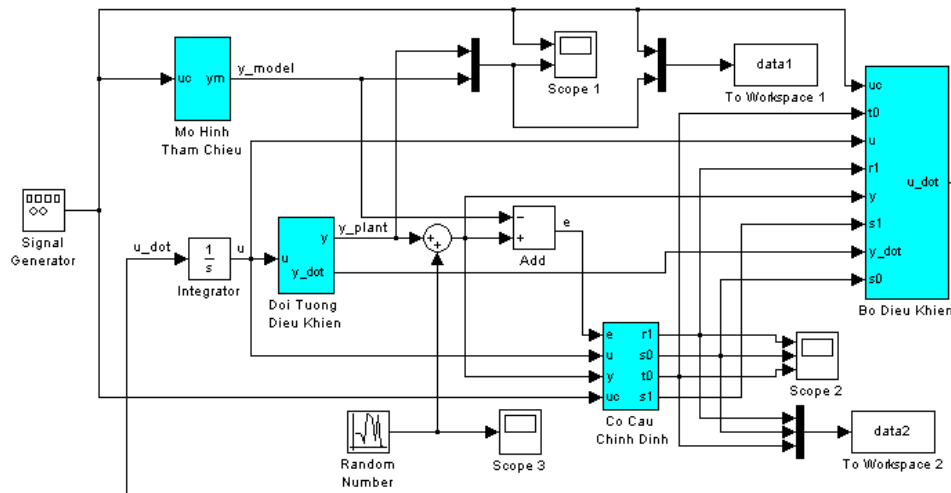
$$\frac{d\theta}{dt} = -\gamma.e. \frac{\frac{\partial e}{\partial \theta}}{\alpha + \left(\frac{\partial e}{\partial \theta}\right)^T \cdot \left(\frac{\partial e}{\partial \theta}\right)}$$

Với các thông số chỉnh định là r_1 , s_0 và t_0 ta sẽ có MIT bổ sung như sau:

$$\frac{dr_1}{dt} = -\gamma.e. \frac{\frac{\partial e}{\partial r_1}}{\alpha + \left(\frac{\partial e}{\partial r_1}\right)^T \cdot \left(\frac{\partial e}{\partial r_1}\right)} = \gamma.e. \frac{\frac{B}{A_m \cdot (s+p)} \cdot u}{\alpha + \left(\frac{A}{A_m \cdot (s+p)} \cdot u\right)^T \cdot \left(\frac{A}{A_m \cdot (s+p)} \cdot u\right)}$$

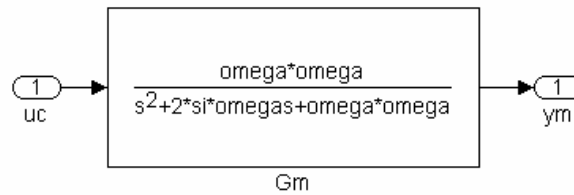
$$\frac{ds_0}{dt} = -\gamma.e. \frac{\frac{\partial e}{\partial s_0}}{\alpha + \left(\frac{\partial e}{\partial s_0}\right)^T \cdot \left(\frac{\partial e}{\partial s_0}\right)} = \gamma.e. \frac{\frac{B}{A_m} \cdot y}{\alpha + \left(\frac{B}{A_m} \cdot y\right)^T \cdot \left(\frac{B}{A_m} \cdot y\right)} \quad (8)$$

$$\frac{dt_0}{dt} = -\gamma.e. \frac{\frac{\partial e}{\partial t_0}}{\alpha + \left(\frac{\partial e}{\partial t_0}\right)^T \cdot \left(\frac{\partial e}{\partial t_0}\right)} = -\gamma.e. \frac{\frac{B}{A_m} \cdot u_c}{\alpha + \left(\frac{B}{A_m} \cdot u_c\right)^T \cdot \left(\frac{B}{A_m} \cdot u_c\right)}$$

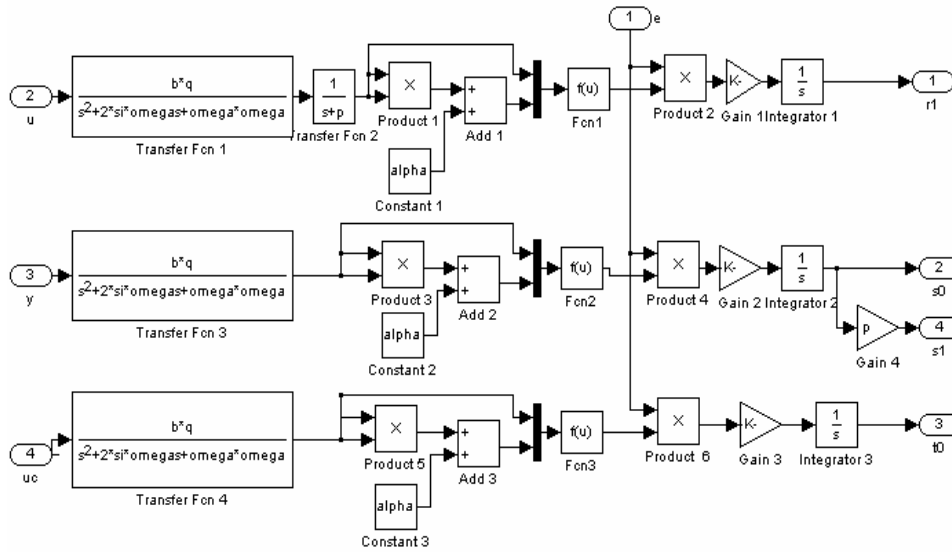


Sơ đồ khối mô phỏng hệ thống theo luật MIT sửa đổi

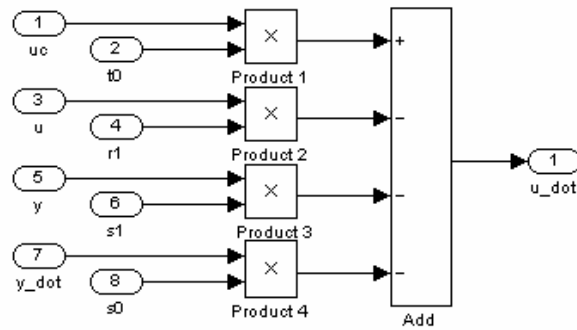
Trong đó khối mô hình tham chiếu có dạng sau:



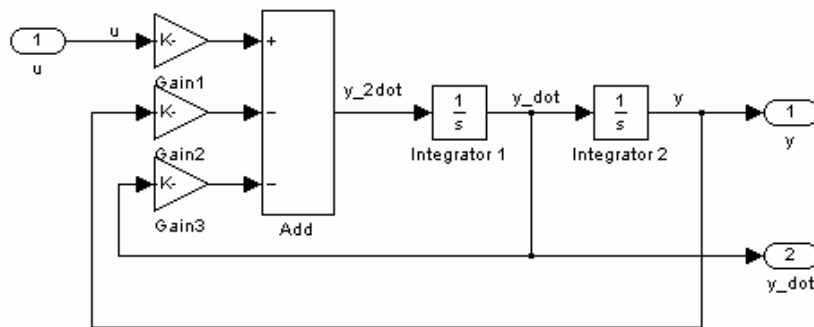
Khối cơ cấu hiệu chỉnh có dạng sau:



Khối bộ điều khiển có dạng sau:



Khối đối tượng điều khiển có dạng sau:



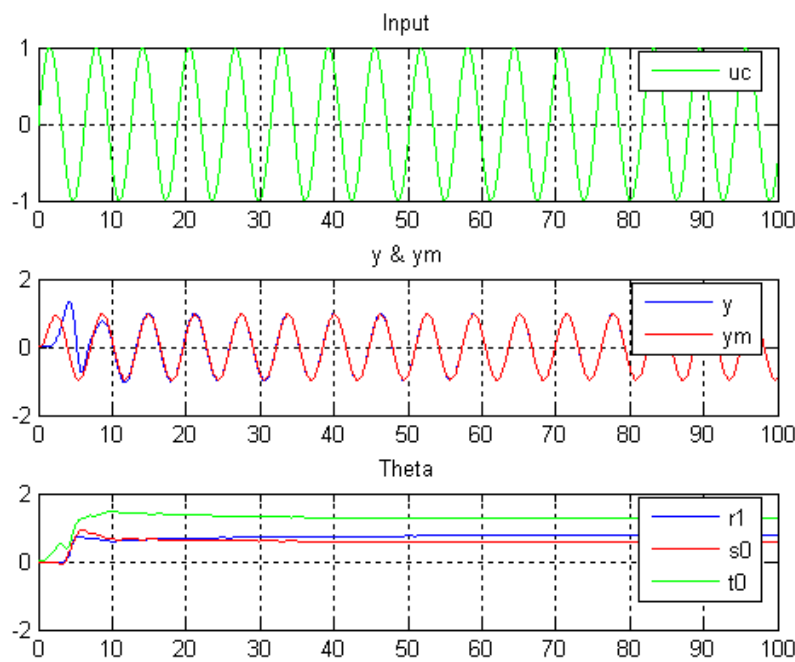
Kết quả mô phỏng theo luật MIT sửa đổi

*Trường hợp không có nhiễu:

Chọn các tham số điều chỉnh như sau:

- Chọn $T_s = 3s$, $\xi = 0.707$, $\gamma = 1$, $\omega = \frac{4}{(T_s * \xi)}$, $\alpha = 1$
- Chọn $p = 2$, $q = (3/2)*p$
- Chọn a, b thay đổi theo tỷ lệ $b = 3*a$ với $a = 1$

Khi đó ta có kết quả mô phỏng như sau:



Modified Mit Rule for $\gamma = 1$ and Amplitude = ± 1

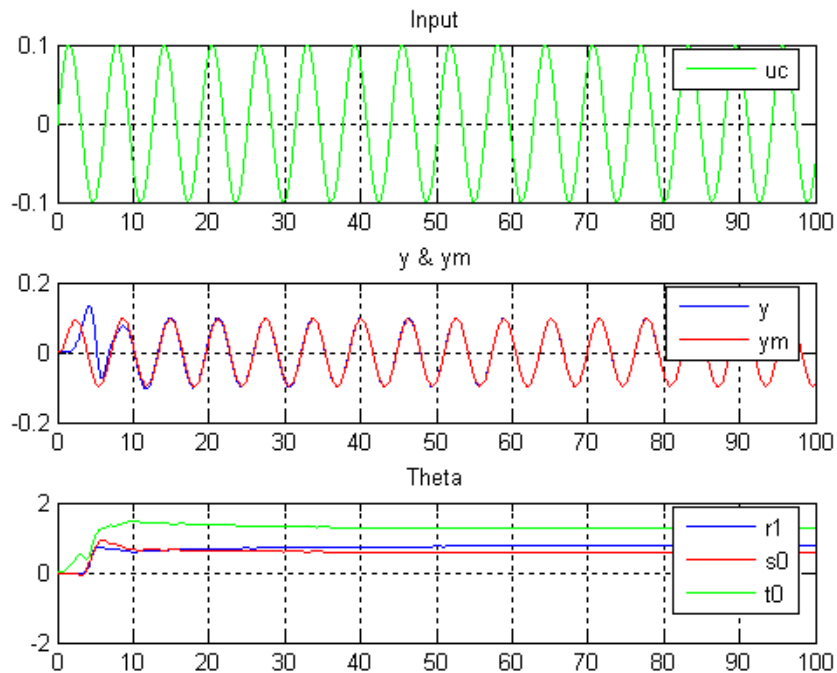
Nhận Xét:

Trong trường hợp biên độ bằng ± 1 thì luật MIT bổ sung cho kết quả tương đối tốt. Sai số giữa ngõ ra của đối tượng và của mô hình chuẩn tiến đến Zero.

Chọn các tham số điều chỉnh như sau:

- Chọn $T_s = 3s$, $\xi = 0.707$, $\gamma = 1$, $\omega = \frac{4}{(T_s * \xi)}$, $\alpha = 0.01$
- Chọn $p = 2$, $q = (3/2)*p$
- Chọn a, b thay đổi theo tỷ lệ $b = 3*a$ với $a = 1$

Khi đó ta có kết quả mô phỏng như sau:



Modified Mit Rule for $\gamma = 1$ and Amplitude = ± 0.1

Nhận Xét:

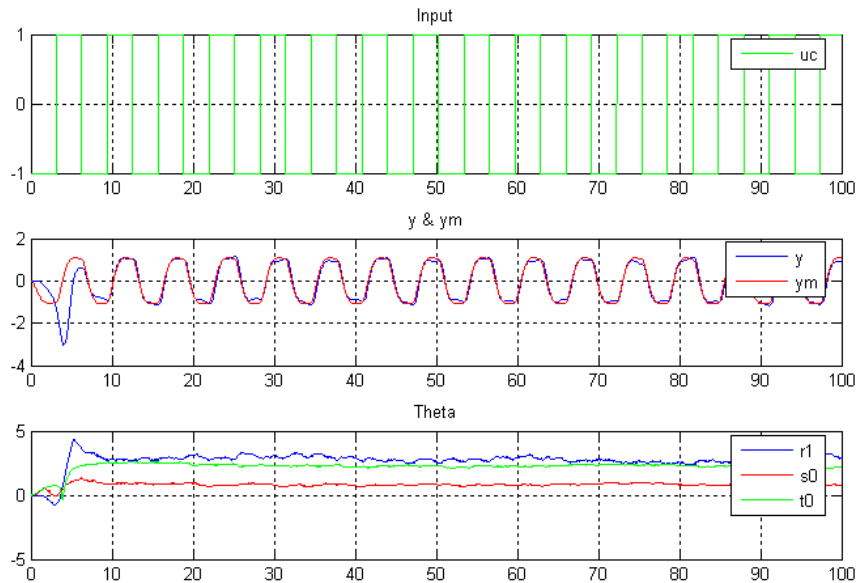
Trong trường hợp biên độ bằng ± 1 thì luật MIT bổ sung cho kết quả tương đối tốt, không phụ thuộc vào biên độ của tín hiệu (đặt) yêu cầu. Sai số giữa ngõ ra của đối tượng và của mô hình chuẩn tiến đến Zero.

***Trường hợp có nhiễu:**

- Chọn Mean = 0.01, Var = 0.5

- Chọn $T_s = 3s$, $\xi = 0.707$, $\gamma = 1$, $\omega = \frac{4}{(T_s * \xi)}$, $\alpha = 0.1$
- Chọn $p = 2$ và $q = (3/2)*p$
- Chọn a, b thay đổi theo tỷ lệ $b = 3*a$ với $a = 1$

Khi đó ta có kết quả mô phỏng như sau:



Modified Mit Rule for $\gamma = 1$ and Amplitude = ± 1

Nhận Xét:

- + Tham số α ảnh hưởng rất lớn đối với luật hiệu chỉnh và việc chọn tham số α cũng rất khó. Tham số $\alpha > 0$ để tránh trường hợp chia hết cho 0.
- + Có thể đạt được Luật MIT bổ sung mà tỉ lệ hiệu chỉnh không phụ thuộc vào biên độ của tín hiệu (đặt) yêu cầu.
- + Có thể nhận thấy rằng tỉ lệ hiệu chỉnh tham số phụ thuộc vào biên độ của tín hiệu yêu cầu một lượng nhỏ bởi vì do nhiễu đo lường.

6. Kết luận:

Khi thiết kế hệ MRAS bậc 2 bằng luật MIT ta thấy kết quả khá tốt với tín hiệu vào là bất kỳ và trong trường hợp có nhiễu đầu vào, nhiễu đầu ra hay nhiễu đo lường thì mô hình cho kết quả tương đối tốt. Tuy nhiên với mức

nhiều quá lớn thì ta cũng không thể cho kết quả tốt nhất được nhưng có thể loại nhiễu bằng cách chỉnh α, γ để có kết quả tương đối.

Ưu điểm

- Loại được nhiễu
- Thiết kế cho hệ thống có các thông số thay đổi không biết trước
- Thích nghi với sự thay đổi của đối tượng

Một số điểm cần chú ý:

- Trong quá trình làm việc để hệ thống vẫn đạt các chỉ tiêu như thiết kế ban đầu thì phải giả thuyết rằng đối tượng không tự thay đổi tức là độ chính xác mô hình vẫn giữ nguyên (điều này thực tế không đạt được), do phần lớn các mô hình đều chứa trong nó một sai lệch nhất định so với đối tượng trong quá trình làm việc.
- Hằng số γ giữ vai trò quyết định đến tốc độ hội tụ hay phân kỳ của thuật toán chỉnh định, dẫn đến phụ thuộc nhiều vào γ và chọn γ rất khó.
- Phương pháp tiệm cận Gradient không cho kết quả cần thiết cho hệ thống kín ổn định.
- Phương pháp này cần thiết cho các tham số của mô hình đối tượng để tính toán độ nhạy. Tuy nhiên điều này không có thực và do đó có thể sử dụng phương pháp xấp xỉ hay bằng các bộ ước lượng thông số.

3.2.4 Thiết kế MRAS dùng lý thuyết ổn định của Lyapunov

Với luật hiệu chỉnh tham số có được từ phương pháp Gradient được trình bày trong mục 3.2.3 lấy gần đúng là để có được luật hiệu chỉnh tham số dựa vào kinh nghiệm có vẻ hợp lí rồi chúng ta thử chỉ ra rằng sai số mô hình sẽ tiến đến 0. Một khả năng khác để có được vòng ngoài của hệ thống thích nghi sử dụng mô hình chuẩn là tìm ra luật hiệu chỉnh mà đảm bảo sai số tiến về 0. Những nghiên cứu cho luật hiệu chỉnh như vậy đã được thực hiện trong một khoảng thời gian dài. Ý tưởng cơ bản để thiết kế luật hiệu chỉnh dựa vào lý thuyết ổn định được trình bày trong mục này và được thể hiện theo lịch sử phát triển.

Để tập trung vào vấn đề chính tránh những chi tiết không cần thiết, tự hiệu chỉnh độ lợi nuôi tiến của hệ thống được biết trước được dùng trong mục này. Hệ thống dùng ở đây giống như ở hình 3.6 nhưng cơ cấu thích nghi thì khác. Vấn đề là tìm luật hồi tiếp để bảo đảm sai số $e = y - y_m$ trong hình 3.6 tiến đến 0, cần biết rằng vấn đề điều khiển hệ thống với đặc tính động học biết trước và hệ số độ lợi chưa biết thì không quá khó. Vấn đề riêng biệt được chọn để trình bày ý tưởng hơn là trình bày một vấn đề thực tế. Một khi ý tưởng cơ bản được phát triển, sự mở rộng đến những cấu hình tổng quát thì tương đối dễ hiểu hơn, chi tiết được trình bày trong TLTK[1].

Phương pháp thứ hai của Lyapunov

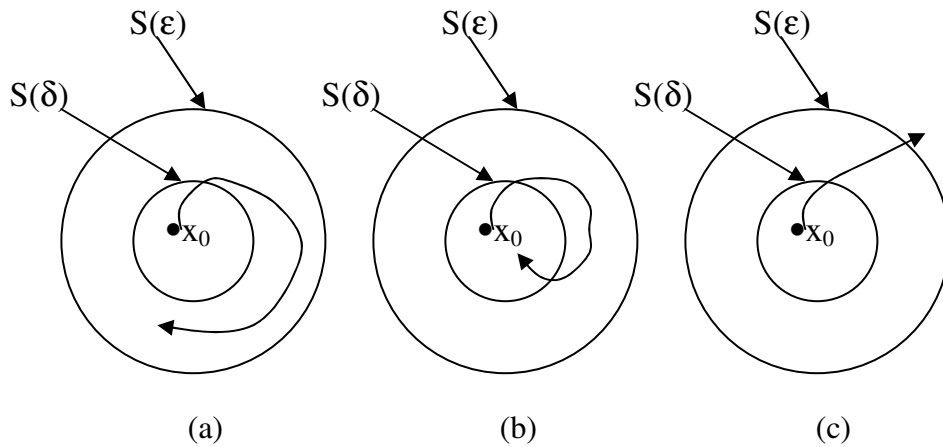
Minh họa bằng đồ thị phương pháp Lyapunov

Hình 3.7 (a), (b) và (c) biểu diễn các trạng thái cân bằng và những đường cong tiêu biểu tương ứng đối với hệ thống ổn định, ổn định tiệm cận và không ổn định. Trong hình 3.7 (a), (b) hoặc (c), vùng $S(\delta)$ giới hạn cho trạng thái ban đầu x_0 , và vùng $S(\epsilon)$ tương ứng với giới hạn cho quỹ đạo xuất phát tại x_0 .

Chú ý rằng những định nghĩa đã được đề cập trước đây không chỉ ra chính xác vùng của điều kiện cho phép ban đầu. Vì vậy các định nghĩa áp dụng cho vùng lân cận của trạng thái cân bằng (là trạng thái tại đó mọi đạo hàm đều triệt tiêu), trừ khi $S(\epsilon)$ tương ứng với trạng thái ban đầu của đối tượng.

Chú ý là trong hình 3.7 (c), đường cong rời vùng $S(\epsilon)$ và dẫn đến trạng thái cân bằng không ổn định. Tuy nhiên, chúng ta không thể nói rằng đường cong sẽ đi đến vô tận bởi vì nó có thể đến gần một vòng tròn giới hạn phía ngoài vùng $S(\epsilon)$. (Nếu một hệ thống tuyến tính bất biến theo thời gian là không ổn định, các đường cong bắt đầu gần với trạng thái cân bằng không ổn định đi đến vô cực. Nhưng trong trường hợp của hệ thống phi tuyến, điều này thật sự không cần thiết).

Sự hiểu biết về các định nghĩa đã nói ở trên là yêu cầu tối thiểu để hiểu việc phân tích ổn định của các hệ thống tuyến tính và phi tuyến có mặt trong phần này. Chú ý rằng những định nghĩa này không chỉ hạn chế ở các khái niệm về sự ổn định của một trạng thái cân bằng. Thực ra, những cách định nghĩa khác cũng được sử dụng. Chẳng hạn, trong các lý thuyết điều khiển thông thường hoặc kinh điển, chỉ có các hệ thống ổn định tiệm cận mới được gọi là hệ thống ổn định, còn các hệ thống khác ổn định theo Lyapunov, nhưng không ổn định tiệm cận, được gọi là không ổn định.



Hình 3.7 (a) Trạng thái cân bằng ổn định
 (b) Trạng thái cân bằng tiệm cận
 (c) Trạng thái cân bằng không ổn định

Ví dụ 3.8 Xét hệ thống được mô tả bởi phương trình trạng thái sau:

$$\dot{x}_1 = x_2 - x_1(x_1^2 + x_2^2)$$

$$\dot{x}_2 = -x_1 - x_2(x_1^2 + x_2^2)$$

Trạng thái cân bằng (đạo hàm = 0) tại gốc tọa độ ($x_1 = 0, x_2 = 0$). Nếu chúng ta định nghĩa một hàm vô hướng $V(x)$ như sau:

$$V(x) = x_1^2 + x_2^2$$

là hàm xác định dương, sao cho đạo hàm theo thời gian hàm $V(x)$ theo một đường cong bất kì

$$\begin{aligned} \dot{V}(x) &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= -2(x_1^2 + x_2^2)^2 \end{aligned}$$

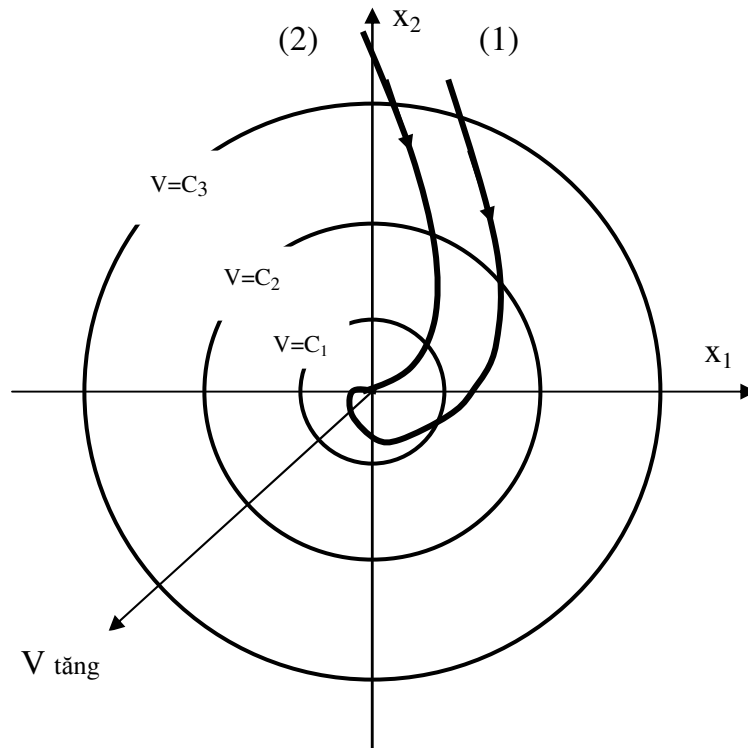
là hàm xác định âm. Điều này cho thấy rằng $V(x)$ tăng liên tục theo đường cong bất kì; vì vậy $V(x)$ là hàm Lyapunov. Hàm $V(x)$ trở thành vô hạn với độ lệch vô hạn từ trạng thái cân bằng, trạng thái cân bằng ở gốc của hệ thống là ổn định tiệm cận trong vùng rộng.

Chú ý rằng nếu chúng ta để $V(x)$ nhận giá trị hằng số 0, C_1, C_2, \dots ($0 < C_1 < C_2 < \dots$), thì $V(x) = 0$ tương ứng với gốc của trạng thái đối tượng và $V(x) = C_1, V(x) = C_2, \dots$ mô tả những vòng tròn không so sánh kèm theo gốc của trạng thái đối tượng, như minh họa ở hình 3.8. Cũng cần chú ý rằng $V(x)$ là bán kính vô tận, hoặc $V(x) \rightarrow \infty$ khi $\|x\| \rightarrow \infty$.

Khi vòng tròn $V(x) = C_k$ nằm hoàn toàn trong vòng tròn $V(x) = C_{k+1}$, một đường cong đại diện đi qua vùng biên giới của các đường viền V từ ngoài vào trong. Từ đây, biểu diễn hình học của hàm Lyapunov có thể được phát biểu như sau: $V(x)$ là thước đo khoảng cách của biến trạng thái x từ gốc tọa độ của trạng thái trung gian. Nếu khoảng cách giữa gốc và biến trạng thái tức thời $x(t)$ tăng liên tục khi t tăng $\{V[x(t)] < 0\}$ thì $x(t) \rightarrow 0$.

Quỹ đạo (1) trên hình 3.8 là chuyển động ổn định tiệm cận về gốc tọa độ, song không thỏa tiêu chuẩn ổn định thứ 2 của Lyapunov: hàm $\dot{V}(x)$ không phải là hàm xác định âm với mọi biến trạng thái x . Tiêu chuẩn ổn định thứ 2 của Lyapunov là điều kiện đủ, không phải là điều kiện cần để đánh giá tính ổn định của nghiệm phương trình vi phân phi tuyến. Nếu thỏa tiêu chuẩn thì hệ ổn định. Nếu không thỏa, vấn đề kết luận về tính ổn định còn bỏ ngỏ, phụ thuộc vào:

1. Chọn hàm $V(x)$
2. Chọn biến trạng thái x



Hình 3.8 Các vòng tròn hằng số V và hai quỹ đạo ổn định

Ví dụ thiết kế MRAS dùng Lyapunov

Giả sử tất cả các biến trạng thái của hệ thống đều đo lường được, định lý về ổn định của Lyapunov có thể dùng để thiết kế luật điều khiển thích nghi đảm bảo sự ổn định cho hệ thống vòng kín, ví dụ sau trình bày ý tưởng này.

Ví dụ 3.9 Hệ MRAS bậc nhất dựa vào lý thuyết ổn định.

Xét bài toán như trong ví dụ 3.2. Khi tham số của đối tượng được biết luật điều khiển theo phương trình 3.6 cho kết quả mong muốn. Một hệ thích nghi sử dụng mô hình chuẩn mà có thể tìm ra các hệ số t_0 và s_0 khi tham số a, b không được biết có thể đạt được như sau :

Sai số :
$$e = y - y_m$$

Lấy đạo hàm và sử dụng phương trình (3.5), (3.14) và mô hình mong muốn để khử đạo hàm y và y_m , ta được :

$$\frac{de}{dt} = -a_m e + (a_m - a - b s_0) y + (b t_0 - b_m) u_c$$

Chú ý rằng sai số e sẽ tiến đến 0 nếu các tham số này bằng với giá trị mong muốn. Bây giờ ta cần cố gắng xây dựng một cơ cấu hiệu chỉnh tham số sao cho các thông số t_0 và s_0 tiến đến giá trị mong muốn. Sử dụng cho mục đích này, hàm Lyapunov có dạng :

$$V(e, t_0, s_0) = \frac{1}{2} [e^2 + \frac{1}{b\gamma} (b s_0 + a - a_m)^2 + \frac{1}{b\gamma} (b t_0 - b_m)^2]$$

Hàm này sẽ bằng 0 khi $e = 0$ và các tham số bộ điều khiển bằng với giá trị tối ưu. Đạo hàm của V là :

$$\begin{aligned} \frac{dV}{dt} &= e \frac{de}{dt} + \frac{1}{\gamma} (b s_0 + a - a_m) \frac{ds_0}{dt} + \frac{1}{\gamma} (b t_0 - b_m) \frac{dt_0}{dt} \\ &= -a_m e^2 + \frac{1}{\gamma} (b s_0 + a - a_m) \left(\frac{ds_0}{dt} - \gamma y e \right) + \frac{1}{\gamma} (b t_0 - b_m) \left(\frac{dt_0}{dt} + \gamma_c e \right) \end{aligned}$$

Nếu các tham số được cập nhập bởi:

$$\frac{dt_0}{dt} = -\gamma_c e \tag{3.17}$$

$$\frac{ds_0}{dt} = \gamma e$$

ta được

$$\frac{dV}{dt} = -a_m e^2$$

Như vậy:

Hàm V sẽ giảm khi e khác 0. Vì vậy có thể kết luận là sai số e sẽ tiến về 0. Tuy nhiên cần chú ý là các tham số t_0 và s_0 sẽ hội tụ đến giá trị cân bằng nếu không có các điều kiện khác tác động vào. Vì vậy luật này tương tự như luật MIT nhưng độ nhạy được thay đổi bởi tín hiệu khác.

Luật hiệu chỉnh các thông số làm ổn định cho hệ thống mà các biến trạng thái có thể đo lường được xây dựng bằng sự tổng quát hoá trực tiếp của kỹ thuật dùng trong ví dụ sau .

Luật hiệu chỉnh theo phương trình 3.17 đạt được bằng cách áp dụng lý thuyết ổn định tương tự như bằng luật MIT (so sánh với ví dụ 3.2) trong cả hai trường hợp, luật hiệu chỉnh có thể viết như sau :

$$\frac{d\theta}{dt} = \gamma \varphi e$$

với θ là vector các tham số , $\varphi = [-u_c \quad y]^T$ khi sử dụng luật theo Lyapunov và

$$\varphi = [-u_c \quad y]^T / (p + a_m)$$

nếu sử dụng luật MIT vector φ có thể được giải thích như là giá trị âm của gradient hàm tổn thất.

Phương pháp Lyapunov bây giờ được áp dụng cho việc hiệu chỉnh hệ độ lợi nuôi tiến.

Ví dụ 3.10

Ở đây chỉ xét vấn đề hiệu chỉnh độ lợi nuôi tiến. Sai số được cho bởi

$$e = G(p)(\theta - \theta^0)u_c$$

giới thiệu một không gian trạng thái biểu thị cho hàm truyền G. Quan hệ giữa tham số θ và sai số e được viết bởi:

$$\frac{dx}{dt} = Ax + B(\theta - \theta^0)u_c \quad (3.18)$$

$$e = Cx$$

Nếu hệ đồng nhất $\dot{x} = Ax$ là ổn định tiệm cận và có tồn tại 2 ma trận P và Q xác định dương sao cho:

$$A^T P + PA = -Q \quad (3.19)$$

Chọn hàm Lyapunov như sau :

$$V = \frac{1}{2} [\gamma x^T P x + (\theta - \theta^0)^2]$$

Đạo hàm V và sử dụng phương trình sai phân 3.18 được :

$$\frac{dV}{dt} = \frac{\gamma}{2} \left(\frac{dx^T}{dt} P x + x^T P \frac{dx}{dt} \right) + (\theta - \theta^0) \frac{d\theta}{dt}$$

Sử dụng phương trình 3.18 ta được :

$$\begin{aligned} \frac{dV}{dt} &= \frac{\gamma}{2} \{ [Ax + B u_c (\theta - \theta^0)]^T P x + x^T P [Ax + B u_c (\theta - \theta^0)] \} \\ &\quad + (\theta - \theta^0) \frac{d\theta}{dt} \\ &= -\frac{\gamma}{2} x^T Q x + (\theta - \theta^0) \left(\frac{d\theta}{dt} + \gamma u_c B^T P x \right) \end{aligned}$$

Nếu luật hiệu chỉnh tham số được chọn là :

$$\frac{d\theta}{dt} = -\gamma u_c B^T P x \quad (3.20)$$

thì đạo hàm của hàm Lyapunov sẽ âm khi $x \neq 0$. Với luật hiệu chỉnh theo phương trình 3.20 vector trạng thái x và cả sai số $e = Cx$ vì vậy sẽ tiến đến không. Tuy nhiên chú ý là sai số tham số $\theta - \theta^0$ không cần thiết là phải tiến đến không.

Ví dụ hệ bậc hai MRAS

Ví dụ 3.11 Xét $G(s) = \frac{K}{s(s+a)}$

và mô hình là $G_m(s) = \frac{Bm}{Am} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$

đa thức A_0 , R , S và T được chọn bởi :

$$A_0(s) = s + a_0$$

$$R(s) = s + r_1$$

$$S(s) = s_0 s + s_1$$

$$T(s) = t_0 s + t_1$$

Phương trình Diophantine 3.7 cho lời giải sau :

$$r_1 = 2\zeta\omega + a_0 - a$$

$$s_0 = (2\zeta\omega a_0 + \omega^2 - a r_1) / K$$

$$s_1 = a_0 \omega^2 / K$$

$$t_0 = \omega^2 / K$$

$$t_1 = a_0 \omega^2 / K$$

để đơn giản hóa, ta chọn : $Q(s) = A_0(s) \cdot A_m(s)$

$$P_1(s) = A_m(s)$$

$$P_2(s) = A_0(s)$$

Mô phỏng hệ bậc hai MRAS trong ví dụ 3.11 với các thông số như sau:

$$\gamma = 1, \zeta = 0.7, \omega = 1, a_0 = 2, a = 1 \text{ và } K = 2.$$

Giả sử rằng $\hat{b}_0 = b_0$.

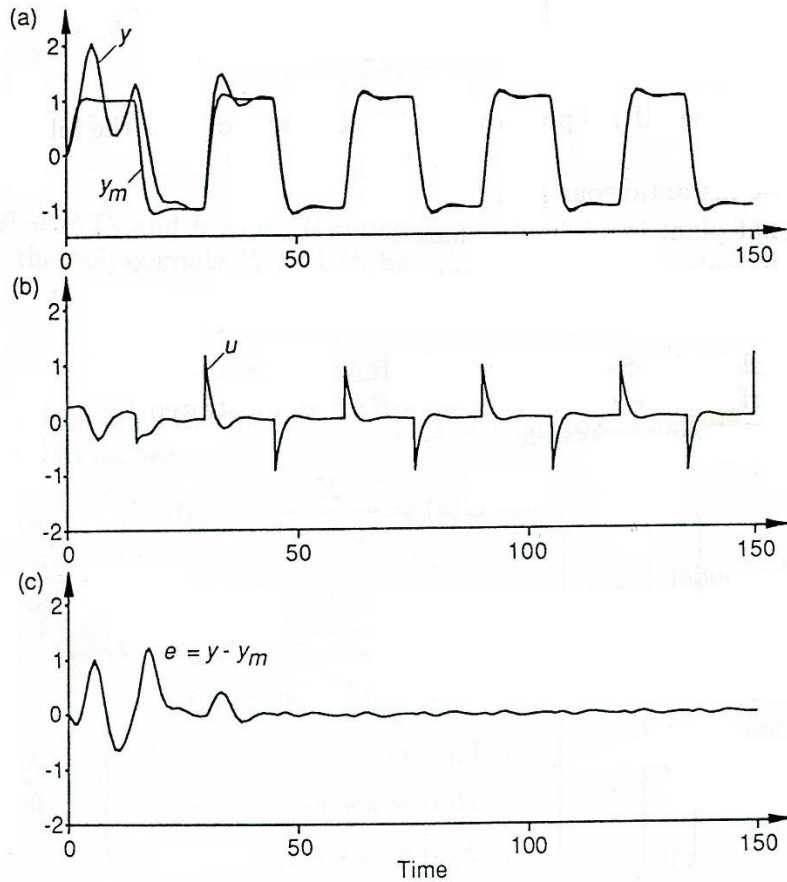
Kết quả mô phỏng của hệ thống thu được như hình 3.9. Trong đó:

- Hình (a): Đáp ứng ngõ ra của hệ thống và ngõ ra của mô hình chuẩn.
- Hình (b): Tín hiệu điều khiển.
- Hình (c): Sai số $e = y - y_m$

Nhận xét:

Các đa thức P_1 , P_2 , Q và A_0 được lựa chọn sao cho tốc độ hội tụ của y về y_m là nhanh nhất. Mặc dù ngõ ra của hệ thống ở cuối quá trình mô phỏng vẫn chưa tiến đến giá trị tối ưu nhưng sai số là rất bé (hình (c)).

Ứng với luật điều khiển trên thì tại thời điểm $t = 150s$ hệ thống vòng kín có một cực thực là -1.95 và hai cực ảo tương ứng với $\omega = 0.84$ và $\zeta = 0.78$, so với đa thức $A_0 A_m$ có một cực thực là -2 và hai cực ảo tương ứng với $\omega = 1$ và $\zeta = 0.7$,



Hình 3.9 Kết quả mô phỏng của hệ thống trong ví dụ 3.11

Hệ thống MRAS rời rạc

Hệ MRAS đã được thực hiện cho hệ liên tục không có nhiễu, nhưng có thể thực hiện được MRAS cho hệ rời rạc. Thuật giải ở trên có thể được dùng cho trường hợp hệ rời rạc. Bộ ước lượng có thể dựa vào chuẩn bình phương tối thiểu. Phần này để dành trình bày trong bộ điều khiển tự chỉnh định trong phần 3.3

MRAS cho hệ thống chỉ biết được từng phần

Trong phần trước ta đã giả sử tất cả mô hình của đối tượng là chưa biết. Trong một số trường hợp đặc tính động học của hệ thống được biết một phần, còn lại là không biết. Sự biết trước này có thể được kết hợp vào hệ MRAS. Điều này có thể thực hiện tùy thuộc chủ yếu vào tham số và cấu trúc của mô hình đối tượng. Phương pháp này được minh họa bằng ví dụ .

Điều khiển thích nghi cho tay máy

Giả sử các biến trạng thái được đo lường đầy đủ, có thể tìm được một biến sai số tuyến tính đối với các tham số, điều này làm dễ dàng trong việc xây dựng hệ thích nghi sử dụng mô hình chuẩn ổn định. Điều này được minh họa bằng việc điều khiển tay máy khi mà đặc tính động học là phi tuyến.

Một thao tác trực tiếp được mô tả bởi mô hình :

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = T \quad (3.21)$$

với q là vector tọa độ tổng quát. H là ma trận quán tính, C là ma trận tắt, G là vector trọng trường. Biến điều khiển là moment đặt vào cơ cấu chấp hành. Phương trình mô tả tay máy có tính chất :

$$\frac{1}{2} \frac{d}{dt} (\dot{q}^T H \dot{q}) = \dot{q}^T H(q) \ddot{q} + \dot{q}^T C(q, \dot{q}) \dot{q} = \dot{q}^T (T - G) \quad (3.21a)$$

Điều này được giải thích là đạo hàm của động năng $\dot{q}^T H \dot{q}$ bằng với công suất được cung cấp bởi cơ cấu chấp hành và moment trọng lực.

Ví dụ 3.12: Tay máy hai khớp nối

Xét tay máy hai khớp nối với tải chưa biết trong hình dưới đây. Khớp nối thứ hai với tải chưa biết được xem như là có thêm một khớp nối với 4 tham số chưa biết: khối lượng m_e , moment quán tính I_e , khoảng cách từ trọng tâm đến khớp nối thứ hai l_{ce} , góc δ_e so với khâu liên kết thứ hai. Hệ thống được mô tả bởi phương trình (3.21) với

$$H = \begin{pmatrix} \theta_1 + 2\theta_3 \cos q_2 + 2\theta_4 \sin q_2 & \theta_2 + \theta_3 \cos q_2 + \theta_4 \sin q_2 \\ \theta_2 + \theta_3 \cos q_2 + \theta_4 \sin q_2 & \theta_2 \end{pmatrix}$$

$$G = \begin{pmatrix} \theta_3 Y_1 + \theta_4 Y_2 + (\theta_1 - \theta_2 + e_1) e_2 \cos q_1 \\ \theta_3 Y_3 + \theta_4 Y_4 \end{pmatrix}$$

với:

$$Y_1 = -2 \sin(q_2) \dot{q}_1 \dot{q}_2 - \sin(q_2) \dot{q}_2^2$$

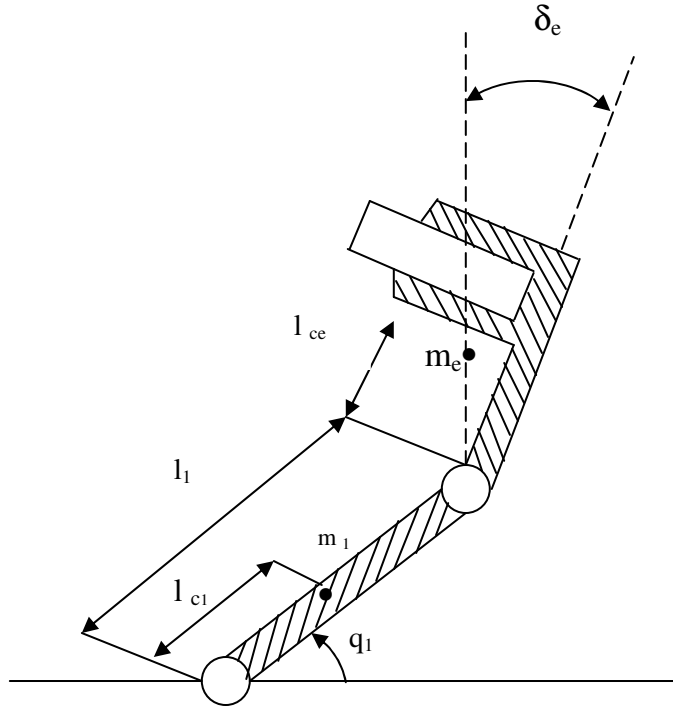
$$Y_2 = 2 \cos(q_2) \dot{q}_1 \dot{q}_2 + \cos(q_2^2)$$

$$Y_3 = \sin(q_2) \dot{q}_1^2 + e_2 \cos(q_1 + q_2)$$

$$Y_4 = -\cos(q_2) \dot{q}_1^2 + e_2 \sin(q_1 + q_2)$$

$$e_1 = m_1 l_1 l_{c1} - m_1 l_{c1}^2$$

$$e_2 = g / l_1$$



Tay máy hai khớp nối với tải chưa biết

với g là gia tốc thường và bốn tham số chưa biết $\theta_1, \dots, \theta_4$ là những hàm có các tham số vật lý chưa biết.

$$\theta_1 = I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2$$

$$\theta_2 = I_1 + m_e l_{ce}^2$$

$$\theta_3 = m_e l_1 l_{ce} \cos \delta_e$$

$$\theta_4 = m_e l_1 l_{ce} \sin \delta_e$$

Bốn tham số chưa biết m_e, I_e, l_{ce} và δ_e được xác định duy nhất bởi $\theta_1, \dots, \theta_4$. Hệ thống có thể được viết lại:

$$\varphi^T(q, \dot{q}, \ddot{q})\theta = T$$

với φ^T được cho bởi:

$$\begin{pmatrix} \ddot{q}_1 + e' \ddot{q}_2 - e' 2 \cos(q_2) \dot{q}_1 + \cos(q_2) \ddot{q}_2 + Y_1 & 2 \sin(q_2) \dot{q}_1 + \sin(q_2) \ddot{q}_2 + Y_2 \\ 0 & \ddot{q}_1 + \ddot{q}_2 & \cos(q_2) \dot{q}_1 + Y_3 & \sin(q_2) \dot{q}_1 + Y_4 \end{pmatrix}$$

$$\theta = (\theta_1, \theta_2, \theta_3, \theta_4)^T$$

$$T = \begin{pmatrix} \tau_1 - e_1 e_2 \cos(q_1) \\ \tau_2 \end{pmatrix}$$

với $e' = e_2 \cos(q_1)$ và τ_1, τ_2 là các moment tác dụng vào. Đặc tính động học có thể được viết dưới dạng tuyến tính theo các tham số với giả sử là tất cả các trạng thái và gia tốc có thể đo lường được.

Ví dụ có thể được tổng quát hoá và phương trình (3.21) có thể được viết thành:

$$T - H'(q)\ddot{q} - C'(q, \dot{q})\dot{q} - G'(q) = \varphi^T(q, \dot{q}, \ddot{q})\theta^0$$

với H', C', G' và φ là biết trước hay có thể đo lường được. Dù là mô hình không tuyến tính, nó vẫn tuyến tính theo các tham số có thể thay đổi. Một điều quan trọng là kiến thức biết trước được dùng và hệ thống đó không xem như là mô hình hộp đen với tham số thay đổi theo thời gian. Mô hình thì vẫn còn chưa thoả mãn bởi vì gia tốc phải được đo cùng với vị trí và vận tốc.

Đặt quỹ đạo tham khảo cho vị trí và vận tốc là q_m và \dot{q}_m . Đưa ra phương trình Lyapunov như sau:

$$V = \frac{1}{2}(\dot{\tilde{q}}^T H(q)\dot{\tilde{q}} + \tilde{q}^T K_p \tilde{q} + \tilde{\theta}^T \Gamma \tilde{\theta})$$

Trong đó: $\tilde{q} = q - q_m, \dot{\tilde{q}} = \dot{q} - \dot{q}_m, \tilde{\theta} = \theta - \theta^0$; K_p và Γ là những ma trận xác định dương. Lấy vi phân V , sử dụng phương trình (3.21a) cho ta:

$$\begin{aligned} \dot{V} &= \dot{\tilde{q}}^T H \ddot{\tilde{q}} + \frac{1}{2} \dot{\tilde{q}}^T \dot{H} \dot{\tilde{q}} + \dot{\tilde{q}}^T K_p \tilde{q} + \tilde{\theta}^T \Gamma \dot{\tilde{\theta}} \\ &= \dot{\tilde{q}}^T (H \ddot{\tilde{q}} - H \ddot{q}_m + C \dot{\tilde{q}} + K_p \tilde{q}) + \tilde{\theta}^T \Gamma \dot{\tilde{\theta}} \\ &= \dot{\tilde{q}}^T (T - G - H \ddot{q}_m - C \dot{q}_m + K_p \tilde{q}) + \tilde{\theta}^T \Gamma \dot{\tilde{\theta}} \end{aligned}$$

Đưa ra luật điều khiển:

$$T = H' \ddot{q}_m + C' \dot{q}_m + G' - K_p \tilde{q} - K_d \dot{\tilde{q}} \quad (3.21b)$$

Luật điều khiển bao gồm thành phần nuôi tiến từ thành phần đã biết của mô hình và thành phần tỉ lệ và hồi tiếp vận tốc, nghĩa là:

$$\dot{V} = \tilde{q}^T (\tilde{H} \ddot{q}_m - \tilde{C} \dot{q}_m + \tilde{G} - K_d \dot{\tilde{q}}) + \tilde{\theta}^T \Gamma \dot{\tilde{\theta}}$$

Trong đó:

$$\begin{aligned}\tilde{H}(q) &= H'(q) - H(q) \\ \tilde{C}(q, \dot{q}) &= C'(q, \dot{q}) - C(q, \dot{q}) \\ \tilde{G}(q) &= G'(q) - G(q)\end{aligned}$$

Đặt:

$$\tilde{H} \ddot{q}_m - \tilde{C} \dot{q}_m + \tilde{G} = \varphi_m^T \tilde{\theta}$$

Đặt được như vậy là do mô hình tuyến tính đối với các tham số. Hơn nữa $\varphi_m = \varphi_m(q, \dot{q}, \ddot{q}_m, \dot{q}_m)$ có nghĩa là chỉ có gia tốc của mô hình phải được biết, không phải là gia tốc thực. Dẫn đến:

$$\dot{V} = -\dot{\tilde{q}}^T K_d \dot{\tilde{q}} + \tilde{\theta}^T (\Gamma \dot{\tilde{\theta}} + \varphi_m^T \dot{\tilde{q}})$$

đề nghị cập nhật thông số:

$$\dot{\tilde{\theta}} = \dot{\tilde{\theta}} = -\Gamma^{-1} \varphi_m^T \dot{\tilde{q}} = -\Gamma^{-1} \varphi_m^T (\dot{q} - \dot{q}_m) \quad (3.21c)$$

Hàm V thoả tính chất của hàm Lyapunov là xác định dương và đạo hàm:

$$\dot{V} = -\dot{\tilde{q}}^T K_d \dot{\tilde{q}}$$

là *bán xác định âm*. Điều này có nghĩa là hệ vòng kín ổn định và vận tốc khi xác lập bằng không. Bộ điều khiển cũng có thể được bổ sung để đảm bảo là sai số vị trí bằng 0.

Luật điều khiển theo phương trình (3.21b) và tham số được cập nhật theo phương trình (3.21c) là các hàm của biến $q, \dot{q}, q_m, \dot{q}_m$ và \ddot{q}_m , nhưng gia tốc của khớp nối không cần thiết phải đo được. Để ý rằng luật điều khiển là trường hợp đặc biệt của hệ MRA tổng quát với $e = \dot{q} - \dot{q}_m$.

3.2.5 Kết luận

Các ý tưởng cơ bản dựa trên MRAS đã được trình bày trong phần này bao gồm :

1. Phương pháp gradient
2. Thiết kế theo Lyapunov và siêu ổn định
3. Số gia sai số

Trong mọi trường hợp luật cập nhật tham số cho dưới dạng :

$$\frac{d\theta}{dt} = \gamma\varphi\varepsilon \quad (3.22)$$

hay dưới dạng chuẩn hoá :

$$\frac{d\theta}{dt} = \gamma \frac{\varphi\varepsilon}{\alpha + \varphi^T \varphi} \quad (3.23)$$

Trong phương pháp gradient, vector φ là giá trị âm của gradient sai số theo các tham số. Ước lượng thông số hay xấp xỉ có thể được dùng trong phương pháp gradient. Trong những trường hợp khác φ là vector lùi có được bằng cách lọc ngõ vào, ra và tín hiệu đặt. Số hạng ε là số gia sai số được giải thích là sai số dự báo của vấn đề ước lượng. Thường dùng số gia sai số tuyến tính theo các thông số.

Phương pháp gradient linh hoạt và đơn giản để áp dụng vào mọi cấu trúc hệ thống. Cách tính toán đòi hỏi phải xác định được hàm độ nhạy bởi vì luật hiệu chỉnh dựa vào việc tính gradient, có thể khẳng định là phương pháp sẽ hội tụ, được cho bởi độ lợi thích nghi γ được chọn là đủ nhỏ. Hơn nữa, giá trị ban đầu của tham số phải chọn để hệ thống vòng kín là ổn định. Phương pháp này sẽ gây không ổn định nếu hệ số độ lợi thích nghi lớn. Vấn đề là khó tìm được giới hạn ổn định trước.

Hệ MRAS tổng quát được đưa ra dựa vào việc thiết kế mô hình kèm theo. Thuật giải này bao gồm những trường hợp đặc biệt của việc thiết kế MRAS đã được trình bày trong các phần trên. Việc ước lượng tham số có thể được thực hiện với nhiều cách khác so với phương trình 3.22 và 3.23.

3.3 Bộ tự chỉnh định (STR – Self Tuning Regulator)

3.3.1 Đặt vấn đề

Sự tương đương chắc chắn

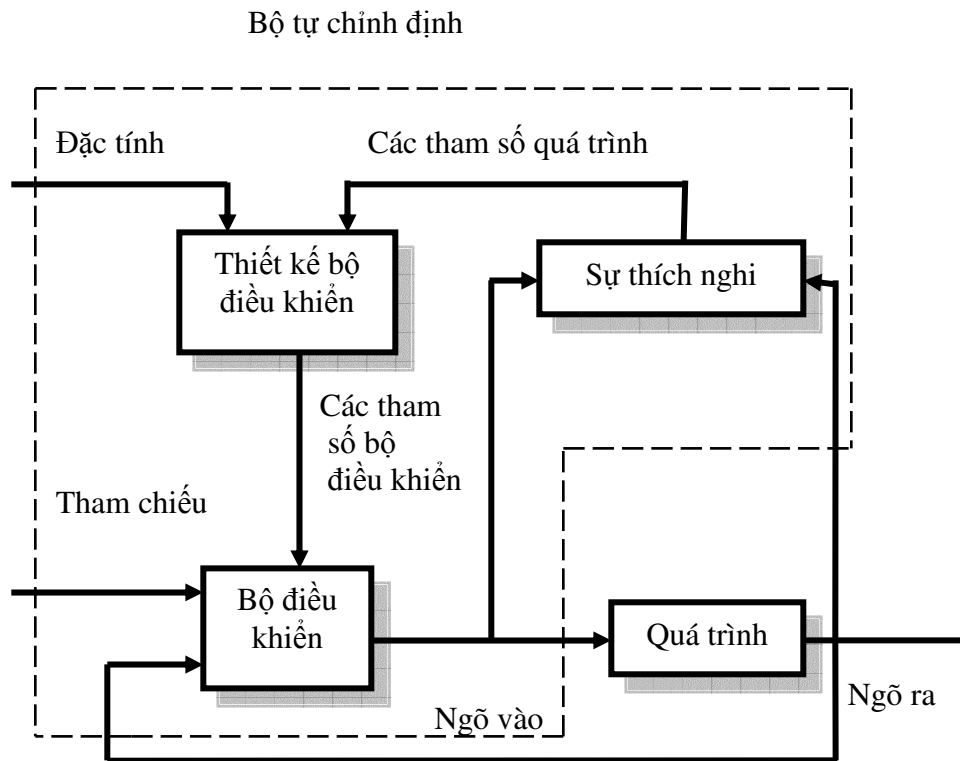
Thông số ước lượng

- Phương pháp gradient
- Bình phương cực tiểu

Các phương pháp thiết kế bộ điều khiển

- PID
- Vị trí cực
- LQG (Linear Quadratic Gaussian)

Bộ tự chỉnh định (STR) dựa trên quan điểm phân tích, đánh giá các thông số chưa biết. Ý tưởng cơ bản được minh họa trong hình 3.10. Các thông số chưa biết được đánh giá trực tuyến (on-line) bằng cách dùng phương pháp ước lượng đệ quy. Các thông số ước lượng được xem như là thông số thực, độ không tin cậy của các ước lượng là bỏ qua. Đây gọi là qui tắc tương đồng nhất định (certainty equivalence principle).



Hình 3.10 Mô hình tự chỉnh định

Nhiều phương pháp ước lượng khác nhau có thể được vận dụng như xấp xỉ ước đoán, bình phương tối thiểu..... Khối 'design' ở hình 3.10 tượng trưng

cho bài giải trực tuyến các bài toán thiết kế hệ thống với các thông số chưa biết trước. Đây là bài toán thiết kế cơ bản. Diễn hình cho phương pháp này là phương pháp khác biệt cực tiểu, bình phương tuyến tính, đặt cực, *model – following*. Phương pháp thiết kế được lựa chọn phụ thuộc vào đặc tính của hệ thống vòng kín. Mục tiêu của mục này là đưa ra quan điểm cơ bản và tính chất của các bộ tự chỉnh định. Bộ tự chỉnh định ban đầu chỉ áp dụng cho các hệ thống lấy mẫu dữ liệu, nhưng các thuật toán liên tục và hỗn hợp (hybrid) cũng được phát triển.

Trong mục này, giả sử hệ thống là SISO :

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (3.24)$$

y : đầu ra

u : đầu vào

$\{e(t)\}$: chuỗi phân bố Gausse

A, B, C : các đa thức theo q (toán tử sai phân tới).

Giả thiết $\text{bậc}A = \text{bậc}C = n$ và $\text{bậc}A - \text{bậc}C = d_0$. Quá trình điều khiển thường được mô tả ở dạng toán tử q^{-1} . Đa thức đặc tính có dạng:

$$A^*(z) = z^n A(z^{-1})$$

$n = \text{bậc}A$. Khi đó mô hình (3.24) được mô tả như sau:

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t - d_0) + C^*(q^{-1})e(t)$$

Bộ tự chỉnh định dựa trên quan điểm ước lượng các thông số của quá trình. Phương pháp dễ hiểu là ước lượng các thông số của hàm truyền của quá trình và nhiễu (thuật toán thích nghi gián tiếp). Các thông số của bộ chỉnh định sẽ không được cập nhật trực tiếp mà là gián tiếp thông qua ước lượng mô hình của hệ thống. Bộ điều khiển thích nghi loại này dựa trên phương pháp bình phương tối thiểu và điều khiển bám theo (Kalman 1958). Phương pháp này không dựa vào đặc tính vòng kín của hệ thống.

Các thông số của bộ chỉnh định cũng có thể ước lượng trực tiếp gọi là thuật toán thích nghi trực tiếp. Cả 2 phương pháp trực tiếp và gián tiếp đều gọi là điều khiển tự chỉnh định.

3.3.2 Bộ tự chỉnh định gián tiếp

Trong phần này, giả sử mô hình của hệ thống có phương trình 3.24. Cách dễ dàng nhất là tạo bộ tự chỉnh định theo như phần 3.3.1 để ước lượng các thông số của đa thức A, B, C .

Xét trường hợp xác định ($e(t) = 0$). Nhiều phương pháp đệ qui đã đề cập có thể được sử dụng để ước lượng các thông số của A, B.

$$\theta^T = [b_0 \ b_1 \dots \ b_m \ a_1 \dots \ a_n]$$

$$\varphi^T(t-1) = [u(t-d_0) \dots u(t-d_0-m) - y(t-1) \dots - y(t-n)]$$

trong đó $n - m = d_0$. Khi đó bộ ước lượng bình phương cực tiểu được cho bởi:

$$\theta(t) = \theta(t-1) + K(t)\varepsilon(t) \quad (3.25)$$

$$\varepsilon(t) = y(t) - \varphi^T(t-1)\theta(t-1) \quad (3.26)$$

$$K(t) = P(t-1)\varphi(t-1) \left[\lambda + \varphi^T(t-1)P(t-1)\varphi(t-1) \right]^{-1} \quad (3.27)$$

$$P(t) = \left[I - K(t)\varphi^T(t-1) \right] P(t-1) / \lambda \quad (3.28)$$

Trong trường hợp nhiễu là ngẫu nhiên, phương pháp bình phương tối thiểu cho ra các ước lượng sai lệch nếu $C(q) \neq q^n$. Lúc này, chúng ta phải dùng các phương pháp như cực đại đệ qui, bình phương cực tiểu tổng quát.

Tính hội tụ

Nếu tín hiệu đầu vào được kích thích đầy đủ và cấu trúc của mô hình cần ước lượng thích hợp thì các ước lượng sẽ hội tụ đến một giá trị thực nếu hệ thống vòng kín ổn định. Điều kiện hội tụ cho các phương pháp khác nhau là khác nhau.

Trong cả 2 trường hợp nhiễu xác định ($e(t) = 0$) và nhiễu ngẫu nhiên ($e(t) \neq 0$) thì điều kiện hội tụ phụ thuộc tín hiệu đầu vào, quá trình và nhiễu của hệ thống. Tín hiệu điều khiển $u(t)$ được phát đi qua khâu hồi tiếp. Điều này làm phức tạp việc phân tích nhưng nó cần thiết để yêu cầu hệ thống vòng kín phải ổn định. Trong MRAS việc phân biệt tính hội tụ sẽ được đề cập rõ hơn ở chương 6 (TLTK[1]).

Bài toán thiết kế nền tảng cho những hệ thống biết trước

Nhiều phương pháp thiết kế được sử dụng trong các bộ tự chỉnh định phụ thuộc vào đặc tính của hệ thống vòng kín. Phương pháp thiết kế thường sử dụng là đặt cực (pole placement). Phương pháp dựa theo mô hình mẫu (mode – following) và phương pháp đặt cực đã được đề cập ở phần 3.2 và phụ lục A (TLTK[1]).

Xét mô hình của hệ thống có phương trình (3.24) và đáp ứng của hệ thống vòng kín mong muốn là :

$$A_m(q).y(t) = B_m(q).u_c(t) \quad (3.29)$$

Bộ điều khiển là:

$$R(q)u(t) = T(q)u_c(t) - S(q)y(t) \quad (3.30)$$

R_1 và S là giải pháp cho phương trình Diophantine

$$AR_1 + B^-S = A_0A_m \quad (3.31)$$

trong đó

$$B = B^+B^- \quad (3.32)$$

$$B_m = B^-B'_m \quad (3.33)$$

$$T = A_0B'_m \quad (3.34)$$

$$R = B^+R_1 \quad (3.35)$$

Một vài điều kiện phải thoả mãn để chắc rằng bộ điều khiển là nhân quả (causal) (xem phụ lục A TLTK[1]). Các phương trình ở trên là cơ bản cho nhiều bài toán thiết kế khác nhau.

*** Một kiểu mẫu cho một bộ tự chỉnh định gián tiếp**

Bộ tự chỉnh định gián tiếp dựa trên thiết kế đặt cực có thể biểu diễn trong thuật toán sau:

Thuật toán 3.1 - Bộ tự chỉnh định gián tiếp

Dữ liệu : Hàm truyền đáp ứng xung vòng kín mong muốn B_m/A_m và đa thức quan sát mong muốn A_0 được cho trước.

Bước 1: Ước lượng các hệ số của đa thức A, B, C trong phương trình (3.24) dùng phương pháp bình phương tối thiểu từ các phương trình (3.25) – (3.28)

Bước 2: Thay A, B, C bằng các ước lượng đạt được ở bước 1 và giải phương trình (3.31) để tìm R_1, S . Tính R bằng phương trình (3.35) và T bằng phương trình (3.34).

Bước 3 : Tính tín hiệu điều khiển từ phương trình (3.30)

Lặp lại bước 1, 2, 3 ở mỗi chu kỳ lấy mẫu.

Một số vấn đề cần chú ý với thuật toán này :

+ Bậc của các đa thức ở phương trình 3.24 hoặc giới hạn bậc cao nhất phải biết trước.

+ Thừa số chung của các ước lượng A, B có khả năng giải được phương trình 3.31

+ Phải đảm bảo hệ thống vòng kín là ổn định.

+ Các tín hiệu nên kích thích liên tục để đảm bảo sự hội tụ các thông số.

Ví dụ 3.13 Bộ tự chỉnh định gián tiếp với nhiễu xác định

Xét hệ thống có hàm truyền :

$$G(s) = \frac{1}{s(s+1)}$$

Hàm truyền này được xem như là mô hình cơ bản của động cơ. Hàm truyền đáp ứng xung với chu kỳ lấy mẫu $h = 0.5$ là :

$$H(q) = \frac{B}{A} = \frac{0.107 q + 0.090}{q^2 - 1.61 q + 0.61} = \frac{0.107(q+0.84)}{(q-1)(q-0.61)}$$

Hệ thống được lấy mẫu có 1 zero = -0.84 bên trong vòng tròn đơn vị với hệ số tắt nhỏ. Giả sử hệ thống vòng kín mong muốn là :

$$\frac{B_m}{A_m} = \frac{0.18}{q^2 - 1.32q + 0.50}$$

Điều này tương ứng với hệ thống có tần số dao động tự nhiên 1 rad/sec và hệ số tắt $\zeta = 0.7$

Giả sử đa thức quan sát là :

$$A_0 = (q - 0.5)^2$$

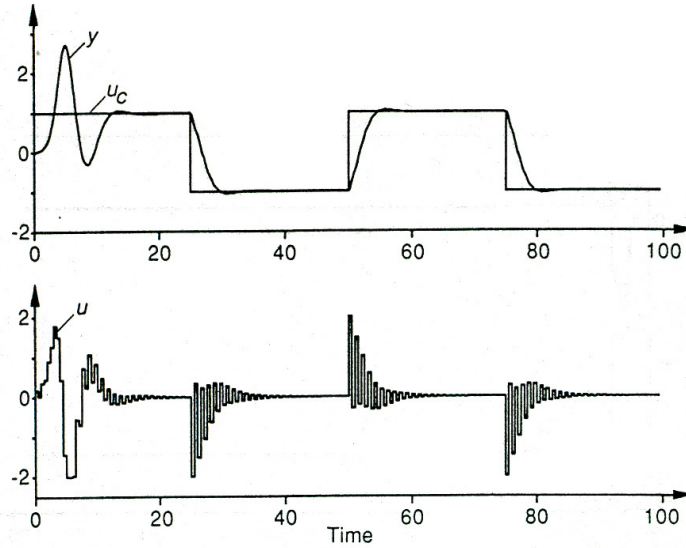
Mô phỏng hệ thống trong ví dụ 3.13.

Kết quả mô phỏng được mô tả ở hình (3.11), (3.12) và (3.13).

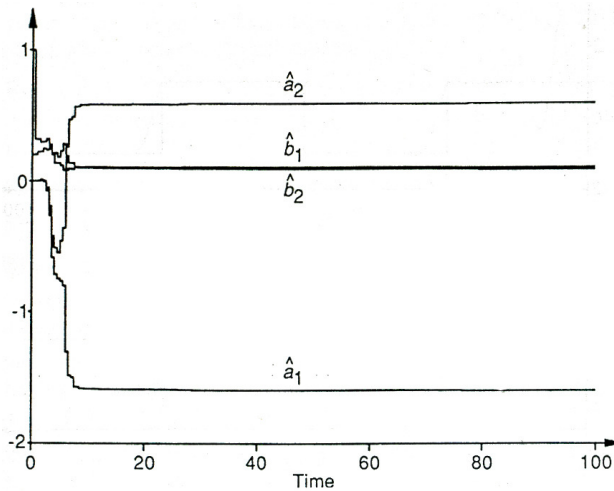
- Hình 3.11 biểu diễn tín hiệu đầu ra và tín hiệu điều khiển của hệ thống thực khi một bộ tự chỉnh định gián tiếp được sử dụng với phương pháp bình phương cực tiểu và zero $z = -0.84$ của hệ thống thực bị khử.
- Hình 3.12 chỉ ra việc ước lượng các thông số của hệ thống hội tụ nhanh đến các thông số của mô hình thực. Có sự dao động lớn của

tín hiệu điều khiển do việc khử zero. Dao động này là kết quả của sự chọn lựa kém trong bài toán thiết kế cơ bản chứ không phải phụ thuộc vào bộ tự chỉnh định. Dao động này có thể tránh được bằng cách thay đổi thiết kế mà không khử zero của hệ thống thực (chẳng hạn $B_m = B$).

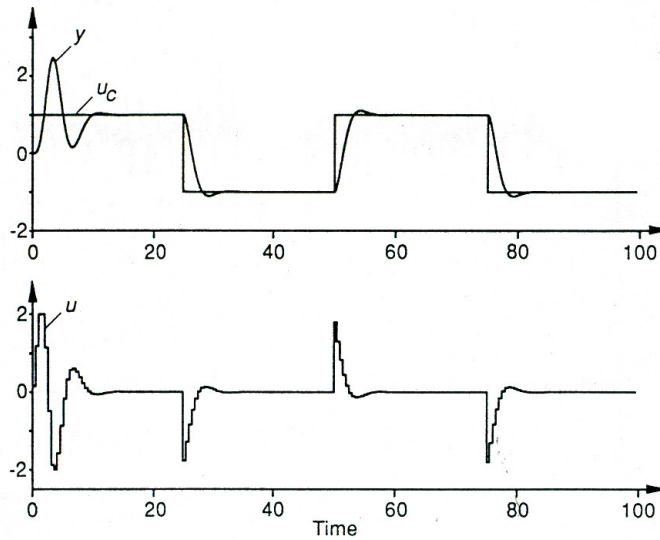
- Hình 3.13 chỉ ra kết quả khi thay đổi thiết kế không có zero nào bị khử. Đáp ứng của hệ thống vòng kín bây giờ đã được thoả mãn.



Hình 3.11 Tín hiệu đầu ra và tín hiệu điều khiển của hệ thống thực khi sử dụng bộ tự chỉnh định gián tiếp.



Hình 3.12 Kết quả mô phỏng việc ước lượng thông số.



Hình 3.13 Tương tự hình 3.11 nhưng với hệ thống không khử điểm zero.

Ví dụ 3.14 Bộ tự chỉnh định với nhiễu ngẫu nhiên :

Xét hệ thống được mô tả như sau :

$$y(t) + ay(t-1) = bu(t-1) + e(t) + c e(t-1)$$

với $a = -0.9$, $b = 3$, $c = -0.3$. Bài toán thiết kế cơ bản được sử dụng là điều khiển sai lệch cực tiểu. Bộ điều khiển sai lệch cực tiểu được cho như sau :

$$u(t) = -\frac{c-a}{b}y(t) = -0.2y(t)$$

Điều này dẫn đến hệ thống vòng kín : $y(t) = e(t)$

Phương pháp cực đại đệ qui được sử dụng để ước lượng các thông số chưa biết a , b và c . Các ước lượng đạt được từ phương trình 3.25 – 3.28 với :

$$\theta^T = [b \ a \ c]$$

$$\varphi^T(t-1) = [u(t-1) \ -y(t-1)\varepsilon(t-1)]$$

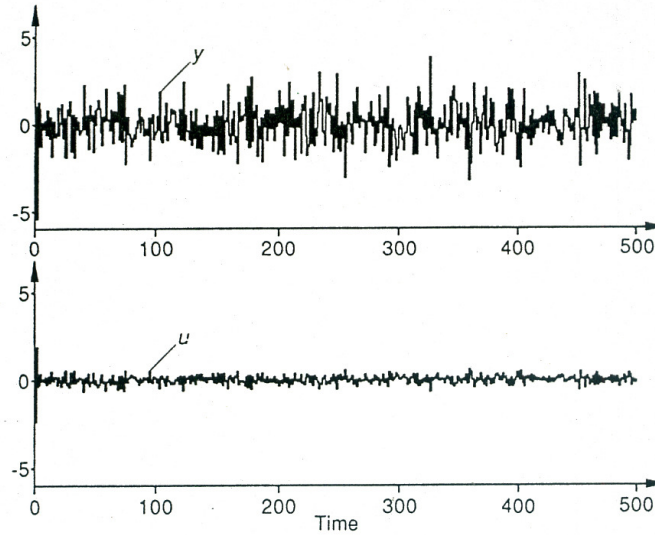
$$\varepsilon(t) = y(t) - \varphi^T(t-1)\hat{\theta}(t-1)$$

Bộ điều khiển là:

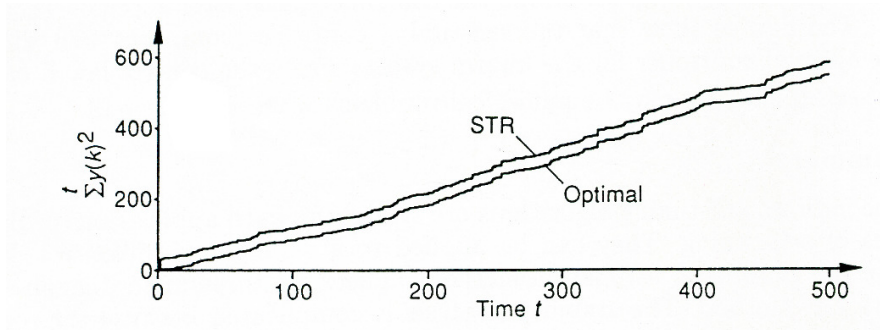
$$u(t) = -\hat{s}_0(t)y(t)$$

$$\hat{s}_0(t) = \frac{\hat{c}(t) - \hat{a}(t)}{\hat{b}(t)}$$

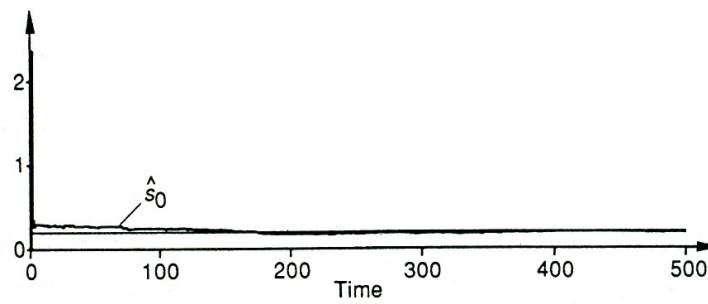
Mô phỏng bộ tự chỉnh định trong ví dụ 3.14.



Hình 3.14 Ngõ ra và tín hiệu điều khiển của hệ thống khi sử dụng bộ tự chỉnh định gián tiếp dựa trên điều khiển sai lệch cực tiểu



Hình 3.15 Hàm chi phí của hệ thống khi sử dụng bộ tự chỉnh định gián tiếp và điều khiển sai lệch cực tiểu tối ưu



Hình 3.16 Biểu diễn thông số $\hat{s}_0(t)$ của bộ điều khiển.

Hình 3.15 biểu diễn hàm chi phí :

$$V(t) = \sum_{i=1}^t y^2(i)$$

Khi sử dụng bộ điều khiển sai lệch cực tiểu tối ưu và bộ tự chỉnh định gián tiếp. Đường cong cho tổn hao tích lũy của STR gần với đường cong tối ưu. Điều này có nghĩa bộ tự chỉnh định gần như tối ưu ngoại trừ khoảng t quá độ khi khởi động.

Tóm tắt : Thuật toán tự chỉnh định gián tiếp là những ứng dụng đơn giản của ý tưởng tự chỉnh định. Chúng có thể được áp dụng tới nhiều phương pháp thiết kế bộ điều khiển và ước lượng thông số. Có 3 khó khăn chính với phương pháp này. Phân tích tính ổn định là phức tạp bởi vì các thông số chỉnh định phụ thuộc vào các thông số đã ước lượng. Thường thì cần phải giải các phương trình tuyến tính trong các thông số bộ điều khiển. Lộ trình từ các thông số quá trình đến các thông số tự chỉnh có thể có các điểm kỳ dị. Điều này xảy ra trong các phương pháp thiết kế dựa vào phương pháp đặt cực, chẳng hạn, nếu mô hình đã ước lượng có chung cực và zero. Các cực và zero chung cần phải loại bỏ trước khi tiến hành phương pháp đặt cực. Do đó việc phân tích tính ổn định chỉ thực hiện trong một số ít trường hợp. Để đảm bảo các thông số hội tụ đến các giá trị chính xác thì cấu trúc của mô hình phải chính xác và tín hiệu đầu vào phải kích thích liên tục.

3.2.3 Bộ tự chỉnh định trực tiếp

Khối lượng tính toán cho các thuật toán ở phần trước tốn nhiều thời gian và tính ổn định rất khó để phân tích. Nhiều thuật toán khác được đề xuất để việc tính toán thiết kế đơn giản hơn. Ý tưởng là dùng các đặc tính, các cực và zero mong muốn để viết lại mô hình hệ thống sao cho các bước thiết kế là không đáng kể. Điều này dẫn tới việc thông số hoá lại mô hình.

Nhân phương trình Diophantine (3.31) với $y(t)$ và dùng mô hình có phương trình 3.24 thì :

$$\begin{aligned} A_0 A_m y(t) &= R_1 A y(t) + B^- S y(t) \\ &= R_1 B u(t) + B^- S y(t) + R_1 C e(t) \\ &= B^- [R u(t) + S y(t)] + R_1 C e(t) \end{aligned} \quad (3.36)$$

Chú ý rằng phương trình 3.36 có thể được xem như là một mô hình của hệ thống được thông số hoá trong B^- , R và S . Việc ước lượng các thông số này tạo ra các đa thức R và S của bộ chỉnh định một cách trực tiếp. Kết hợp

phương trình 3.34, tín hiệu điều khiển được tính từ phương trình 3.30. Lưu ý mô hình ở phương trình 3.36 là phi tuyến trừ phi B^- là hằng số.

Cách khác để thông số hoá là viết mô hình ở phương trình 3.36 như:

$$A_0 A_m y = \bar{R}u + \bar{S}y + R_1 C e \quad (3.37)$$

Trong đó

$$\bar{R} = B^- R \quad \text{và}$$

$$\bar{S} = B^- S$$

Chú ý đa thức R ở phương trình (3.36) là monic (đa thức có hệ số ở bậc cao nhất bằng 1) nhưng \bar{R} ở phương trình (3.37) thì không phải monic. Các đa thức \bar{R} và \bar{S} có một thừa số chung tương ứng cho các zero tắt kém. Thừa số chung này nên khử bỏ trước khi tính toán luật điều khiển.

• **Thuật toán 3.2 - Bộ tự chỉnh định trực tiếp :**

Bước 1: Ước lượng các hệ số của đa thức \bar{R} và \bar{S} ở mô hình phương trình (3.37).

Bước 2: Khử các thừa số chung trong \bar{R} và \bar{S} để đạt được R và S .

Bước 3: Tính tín hiệu điều khiển từ phương trình 3.30 mà R và S có được ở bước 2.

Lặp lại bước 1, 2, 3 ở mỗi chu kỳ lấy mẫu.

Thuật toán này tránh việc ước lượng phi tuyến nhưng cần phải ước lượng nhiều thông số hơn khi dùng phương trình 3.36 vì các thông số của đa thức B^- được ước lượng 2 lần. Bước 2 do đó rất khó thực hiện.

Vì việc ước lượng các thông số ở phương trình 3.36 tương đối khó nên ta xét trường hợp đặc biệt B^- là hằng số. Giả sử tất cả các zero có thể bị khử ($B^- = b_0$)

$$A_0 A_m y(t) = b_0 [Ru(t) + Sy(t)] + R_1 C e(t) \quad (3.38)$$

Đáp ứng mong muốn như sau:

$$A_m y_m(t) = b_0 T u_c(t)$$

Trong đó:

bậc(A) = n và A_0 chia hết cho T. Sai số $e(t) = y(t) - y_m$ được cho bởi:

$$\varepsilon(t) = \frac{b_0}{A_0 A_m} [R u(t) + S y(t) - T u_c(t)] + \frac{R_1 C}{A_0 A_m} e(t)$$

Bây giờ ta xem xét các trường hợp khác nhau. Đầu tiên giả sử $e = 0$. Đa thức quan sát có thể được chọn tự do, khi dùng mô hình liên tục theo thời gian thì điều cần thiết phải giả sử $b_0/(A_0 A_m)$ là SPR (Strictly Positive Real = Thực dương chặt) để đạt được một MRAS ổn định. Ta cũng cần lưu ý rằng hàm truyền có các hệ số là số thực dương thoả điều kiện cần để ổn định được gọi là PR (Positive Real). Hàm là SPR nếu nó ổn định với độ dự trữ ε dương nhỏ tùy ý. Một điều kiện tương tự cũng là cần thiết cho các mô hình rời rạc theo thời gian. Viết lại mô hình như sau:

$$\begin{aligned} \varepsilon(t) &= b_0 \left[R \frac{u(t)}{A_0 A_m} + S \frac{y(t)}{A_0 A_m} - T \frac{u_c(t)}{A_0 A_m} \right] \\ &= b_0 [R^* u_f(t-d_0) + S^* y_f(t-d_0) - T^* u_{cf}(t-d_0)] \end{aligned}$$

trong đó

$$\begin{aligned} u_f(t) &= \frac{1}{A_0^*(q^{-1})A_m^*(q^{-1})} u(t) \\ y_f(t) &= \frac{1}{A_0^*(q^{-1})A_m^*(q^{-1})} y(t) \\ u_{cf}(t) &= \frac{1}{A_0^*(q^{-1})A_m^*(q^{-1})} u_c(t) \end{aligned}$$

Điều này tương ứng với trường hợp $P = Q = A_0 A_m$ ở phần 3.2. Tính hội tụ bây giờ sẽ phụ thuộc vào dấu của b_0 . Điều này chỉ ra mối liên hệ giữa MRAS và STR.

• **Thuật toán 3.3 - Bộ tự chỉnh trực tiếp với nhiễu xác định**

Dữ liệu : Cho trước giới hạn thấp nhất của thời gian trễ d_0 và dấu của b_0 , đáp ứng xung hàm truyền vòng kín mong muốn b_0/A_m^* và đa thức quan sát mong muốn A_0 .

Bước 1 : Ước lượng các hệ số của đa thức R^* , S^* , và T^* ở phương trình 3.38 dùng phương pháp ước lượng đệ qui.

Bước 2 : Tính tín hiệu điều khiển từ :

$$R^* u(t) = -S^* y(t) + T^* u_c(t)$$

Lặp lại các bước 1, 2 ở mỗi chu kỳ lấy mẫu.

Thuật toán này tương ứng với bộ điều khiển thích nghi dùng mô hình chuẩn ở phần 3.2. Chú ý thuật toán yêu cầu b_0 phải biết trước. Nếu không biết trước b_0 thì cũng có thể ước lượng được bằng cách thay phương trình 3.38 bằng :

$$A_0 A_m y(t) = R u(t) + S y(t) + R_1 C e(t)$$

mà R bây giờ không phải là monic.

- **Các bộ điều khiển thay đổi cực tiểu và mức trung bình di chuyển (Minimum – Variance and Moving – average)**

Các thuật toán điều khiển trong trường hợp nhiễu ngẫu nhiên cho hệ thống được mô tả bởi phương trình 3.24 sẽ được xem xét. Đầu tiên giả sử mô hình biết trước, e là một nhiễu ngẫu nhiên và $u_c = 0$. Đa thức của bộ quan sát tối ưu cho mô hình ở phương trình 3.24 là $A_0 = C$. Tiêu chuẩn thiết kế là thay đổi cực tiểu hoặc trung bình di chuyển.

Nếu quá trình là cực tiểu pha, bộ chỉnh định thay đổi cực tiểu được cho bởi:

$$R^*(q^{-1})u(t) = -S^*(q^{-1})y(t) \quad (3.39)$$

Trong đó R^* và S^* là nghiệm có bậc cực tiểu của phương trình Diophantine

$$A^*(q^{-1})R^*(q^{-1}) + q^{-d_0}B^*(q^{-1})S^*(q^{-1}) = B^*(q^{-1})C^*(q^{-1}) \quad (3.40)$$

với $d_0 = \text{Bậc}(A) - \text{Bậc}(B)$. Bộ điều khiển thay đổi cực tiểu tương ứng với mô hình mong muốn với một khoảng trễ d_0 bước, $A_m^* = I$. Từ phương trình 3.40 thì R^* phải chia hết cho B^* :

$$R^* = R_1^* B^*$$

Trong đó : $\text{Bậc}(R_1^*) = d_0 - 1$. Phương trình 3.40 được viết lại :

$$A^* R_1^* + q^{-d_0} S^* = C^*$$

$$\begin{aligned} C^* y(t) &= A^* R_1^* y(t) + S^* y(t - d_0) \\ &= B^* R_1^* u(t - d_0) + S^* y(t - d_0) + R_1^* C^* e(t) \\ &= R^* u(t - d_0) + S^* y(t - d_0) + R_1^* C^* e(t) \end{aligned}$$

phương trình này có thể được viết lại :

$$y(t + d_0) = \frac{1}{C^*} [R^* u(t) + S^* y(t)] + R_1^* e(t + d_0) \quad (3.41)$$

với bộ điều khiển ở phương trình 3.39 thì đầu ra của hệ thống vòng kín trở thành :

$$y(t) = R^*_1(q^{-1}).e(t)$$

Ngõ ra vì vậy là một trung bình di chuyển với bậc $(d_0 - 1)$. Trong Åström (1970) chỉ ra rằng bộ chỉnh định sẽ cực tiểu sự thay đổi ngõ ra. Một đặc điểm quan trọng là ngõ ra trở thành một trung bình di chuyển bậc $(d_0 - 1)$. Chú ý số tự nhiên d_0 được diễn tả như là số mẫu trôi qua để đầu ra thay đổi khi đầu vào thay đổi.

Bộ điều khiển thay đổi cực tiểu có hạn chế là tất cả các zero của quá trình đều bị khử. Điều này có nghĩa sẽ là khó khăn nếu B có các zero bên ngoài vòng tròn đơn vị. Các khó khăn này sẽ tránh được ở bộ điều khiển trung bình di chuyển. Bộ điều khiển này làm cho ngõ ra có bậc lớn hơn $(d_0 - 1)$. Bộ điều khiển được đề xuất như sau: thừa số B^+ và B^- trong B với B^+ có các zero tắt nhanh (*zero well-damped*). Xác định R^* và S^* từ :

$$A^*R^* + q^{-d_0}B^*S^* = B^+C^*$$

Phương trình 3.41 cho ta:

$$y(t+d) = \frac{1}{C^*}[R^*u(t) + S^*y(t)] + R^*_1e(t+d) \quad (3.42)$$

Trong đó:

$$R^* = R^+_1B^+$$

Vì ngõ ra được điều khiển là một quá trình trung bình di chuyển với bậc $(d - 1)$ nên chúng ta gọi là điều khiển trung bình di chuyển. Chú ý không có zero nào bị khử nếu $B^+ = 1$, có nghĩa $d = \text{bậc}(A) = n$.

Cả 2 luật điều khiển thay đổi cực tiểu và trung bình di chuyển dẫn đến mô hình tương đương của phương trình 3.41 và 3.42. Sự khác nhau duy nhất là ở giá trị của d mà sẽ điều khiển số zero của quá trình bị khử. Với $d = d_0 = \text{Bậc}(A) - \text{Bậc}(B)$: tất cả zero bị khử. Với $d = \text{Bậc}(A)$: không có zero nào bị khử.

Lọc với A^*_0 trong phương trình 3.38 cũng có thể tạo ra mô hình của phương trình 3.42 :

$$y(t+d) = \frac{A^*_0}{C^*}[R^*u_f(t) + S^*y_f(t)] + R^*_1e(t+d) \quad (3.43)$$

Nếu B^+ chứa tất cả các zero ổn định của hệ thống thì nó sẽ tương ứng như bộ điều khiển thay đổi cực tiểu cận tối ưu trong Åström (1970)

Bộ tự chỉnh định thay đổi cực tiểu và trung bình di chuyển

Thuật toán 3.4 - Thuật toán tự chỉnh định trực tiếp cơ bản

Dữ liệu : Cho trước khoảng dự báo d . Gọi k và l tương ứng là số thông số trong R^* và S^* .

Bước 1: Ước lượng các hệ số của đa thức R^* và S^*

$$y(t+d) = R^*(q^{-1})u_f(t) + S^*(q^{-1})y_f(t) + \varepsilon(t+d) \quad (3.44)$$

trong đó : $R^*(q^{-1}) = r_0 + r_1q^{-1} + \dots + r_kq^{-k}$

$$S^*(q^{-1}) = s_0 + s_1q^{-1} + \dots + s_lq^{-l}$$

Và

$$u_f(t) = \frac{1}{A_0^*(q^{-1})} u(t)$$

$$y_f(t) = \frac{1}{A_0^*(q^{-1})} y(t)$$

sử dụng các phương trình 3.25 – 3.28 với

$$\varepsilon(t) = y(t) - R^*u_f(t-d) - S^*y_f(t-d) = y(t) - \varphi^T(t-d)\hat{\theta}(t-1)$$

$$\varphi^T = \frac{1}{A_0^*(q^{-1})} [u(t) \dots u(t-k) \quad y(t) \dots y(t-l)]$$

$$\theta^T = [r_0 \dots r_k \quad s_0 \dots s_l]$$

Bước 2: Tính luật điều khiển

$$R^*(q^{-1})u(t) = -S^*(q^{-1})y(t) \quad (3.45)$$

Với R^* và S^* được thay bằng các ước lượng tương ứng trong bước 1.

Lặp lại các bước 1 và 2 ở mỗi chu kỳ lấy mẫu.

Chú ý: Thông số r_0 có thể ước lượng hoặc giả sử biết trước. Ở các trường hợp sau để thuận lợi ta viết R^* như sau:

$$R^*(q^{-1}) = r_0(1 + r_1'q^{-1} + \dots + r_k'q^{-k})$$

Và sử dụng

$$\varepsilon(t) = y(t) - r_0u_f(t-d) - \varphi^T(t-d)\hat{\theta}(t-1)$$

$$\varphi^T = \frac{1}{A_0^*(q^{-1})} [r_0 u(t-1) \cdots r_0 u(t-k) \quad y(t) \cdots y(t-l)]$$

$$\theta^T = [r_1' \cdots r_k' \quad s_0 \cdots s_l]$$

Tính chất tiệm cận

Mô hình ở phương trình 3.41 và 3.42 được diễn tả như là việc thông số hoá lại mô hình ở phương trình 3.24. Chúng tương đồng với mô hình ở phương trình 3.44 trong thuật toán 3.4 nếu A_0 được chọn bằng C. Vector hồi qui không tương quan với sai số và phương pháp ước lượng bình phương tối thiểu sẽ hội tụ tới thông số thật. Một kết quả đáng kinh ngạc là cũng tự chỉnh định chính xác khi $A_0 \neq C$. Kết quả sau chỉ ra các thông số tự chỉnh định chính xác có giá trị tương đồng với thuật toán 3.4 khi $A_0 \neq C$.

Định lý 3.1 – Tính chất tiệm cận

Xét thuật toán 3.4 với $A_0^* = 1$ dùng phương pháp ước lượng bình phương cực tiểu. Thông số $b_0 = r_0$ có thể cố định hoặc được ước lượng. Giả sử vector hồi qui có giới hạn, và các ước lượng là hội tụ. Hệ thống vòng kín đạt được trong điều kiện giới hạn có đặc điểm

$$\begin{aligned} \overline{y(t+\tau)y(t)} &= 0 & \tau &= d, d+1, \dots, d+l \\ \overline{y(t+\tau)u(t)} &= 0 & \tau &= d, d+1, \dots, d+k \end{aligned} \quad (3.46)$$

trong đó dấu gạch chỉ giá trị trung bình theo thời gian; k, l là số các thông số ước lượng trong R^* và S^* .

Chứng minh: Mô hình của phương trình 3.44 có thể được viết lại:

$$y(k+d) = \varphi^T(k)\theta + \varepsilon(k+d)$$

và luật điều khiển trở thành:

$$\varphi^T(k)\hat{\theta}(k+d) = 0$$

Tại một trạng thái cân bằng, các thông số ước lượng $\hat{\theta}$ là những hằng số. Hơn nữa, chúng thoả mãn các phương trình chuẩn, trong trường hợp này được viết lại như sau:

$$\frac{1}{t} \sum_{k=1}^t \varphi(k)y(k+d) = \frac{1}{t} \sum_{k=1}^t \varphi(k)\varphi^T(k)\hat{\theta}(t+d)$$

Sử dụng luật điều khiển

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=1}^t \varphi(k) y(k+d) = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=1}^t \varphi(k) \varphi^T(k) [\hat{\theta}(t+d) - \hat{\theta}(k+d)]$$

Nếu thông số ước lượng $\hat{\theta}(t)$ hội tụ khi $t \rightarrow \infty$, và các vector hồi qui bị giới hạn thì vế phải sẽ tiến tới zero. Phương trình 3.46 bây giờ kéo theo $A_0^* = 1$ và xác định về sự hội tụ vector trong thuật toán 3.4

Định lí 3.2 – Tính chất tiệm cận 2

Giả sử thuật toán 3.4 với phương pháp ước lượng bình phương cực tiểu được áp dụng cho phương trình 3.24 và:

$$\min(k, l) \geq n - 1 \quad (3.47)$$

Có nghĩa tín hiệu ra là quá trình có mức trung bình di chuyển bậc (d-1).

Nếu các ước lượng tiệm cận của R và S liên quan với nhau, nghiệm trạng thái cân bằng là:

$$\overline{y(t+\tau)y(t)} = 0 \quad \tau = d, d+1, \dots \quad (3.48)$$

Chứng minh: Hệ thống vòng kín được mô tả như sau:

$$R^* u(t) = -S y(t)$$

$$A^* y(t) = B^* u(t-d_0) + C^* e(t)$$

Vì vậy

$$(A^* R^* + q^{-d_0} B^* S^*) y = R^* C^* e$$

$$(A^* R^* + q^{-d_0} B^* S^*) u = -S^* C^* e$$

Tín hiệu ω được định nghĩa

$$(A^* R^* + q^{-d_0} B^* S^*) \omega = C^* e \quad (3.49)$$

Vì vậy:

$$y = R^* \omega \quad \text{và} \quad u = -S^* \omega$$

Điều kiện của phương trình 3.46 đưa đến

$$\overline{R\omega(t+\tau)y(t)} = 0 \quad \tau = d, d+1, \dots, d+l$$

$$\overline{S\omega(t+\tau)y(t)} = 0 \quad \tau = d, d+1, \dots, d+k$$

Đặt

$$C_{\omega_y}(\tau) = \overline{\omega(t + \tau)y(t)}$$

các phương trình trên có thể được viết lại:

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_k & 0 & \cdots & 0 \\ 0 & r_0 & r_1 & r_2 & \cdots & r_k & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \ddots & \\ 0 & \cdots & 0 & r_0 & r_1 & r_2 & \cdots & r_k \\ s_0 & s_1 & s_2 & \cdots & s_l & 0 & \cdots & 0 \\ 0 & s_0 & s_1 & s_2 & \cdots & s_l & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \ddots & \\ 0 & \cdots & 0 & s_0 & s_1 & s_2 & \cdots & s_l \end{bmatrix} \begin{bmatrix} C_{\omega_y}(d+k+l) \\ \vdots \\ C_{\omega_y}(d) \end{bmatrix} = 0$$

$$C_{\omega_y}(\tau) = 0 \quad \tau = d, d+1, \dots, d+k+l$$

Hàm tương quan thoả mãn phương trình:

$$F^*(q^{-1})C_{\omega_y}(\tau) = 0 \quad \tau \geq 0$$

Hệ thống phương trình 3.49 có bậc

$$n+k = n + \max(k, l)$$

Nếu

$$k+l+1 \geq n + \max(k, l)$$

hoặc tương đương với

$$\min(k, l) \geq n-1$$

dẫn đến

$$C_{\omega_y}(\tau) = 0 \quad \tau = d, d+1, \dots$$

là điều cần chứng minh.

3.3.4 Kết nối giữa MRAS và STR

Các hệ thống thích nghi dùng mô hình chuẩn trực tiếp đã được đề cập trong phần 3.2. Trong phụ lục A (TLTK[1]) cũng chỉ ra mô hình kèm theo và đặt cực là liên quan với nhau. Bây giờ chúng ta sẽ chứng tỏ bộ chỉnh định trực tiếp dùng phương pháp đặt cực ở thuật toán 3.2 là tương đương với một MRAS. Trong trường hợp nhiều xác định, khi B^- là hằng số, mô hình của quá trình được viết lại như sau:

$$y(t) = \varphi_f^T(t - d_0)\theta$$

Trong thuật toán gián tiếp, các thông số được ước lượng bằng các thông số của bộ chỉnh định. Phương pháp bình phương cực tiểu được sử dụng cho việc ước lượng và $\varepsilon(t)$ được viết lại:

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi_f^T(t - d_0)\hat{\theta} \quad (3.50)$$

Thông số cập nhật có thể được viết lại:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi_f^T(t - d_0)\varepsilon(t) \quad (3.51)$$

Chú ý rằng theo phương trình 3.50 thì

$$\varphi_f^T(t - d_0) = -\text{grad}_{\theta}\varepsilon(t)$$

Vector $\varphi_f^T(t - d_0)$ diễn tả như là đạo hàm của độ nhảy. Việc cập nhật thông số ở phương trình 3.51 là một phiên bản rời rạc theo thời gian của luật MIT. Sự khác biệt chính là sai số mô hình $e(t) = y(t) - y_m(t)$ được thay bằng giá trị thặng dư $\varepsilon(t)$ và độ lợi γ ở MRAS được thay bằng ma trận $P(t)$ cho ở phương trình 3.28. P làm thay đổi hướng của gradient và tạo ra một chiều dài bước thích hợp. Ngược lại, luật MIT cũng có thể xem như là một thuật toán gradient để cực tiểu e^2 , phương trình 3.51 được xem như là một phương pháp Newton để cực tiểu $\varepsilon^2(t)$. Giá trị thặng dư ε được xem như số gia sai số.

Chú ý rằng trong các kỹ thuật nhận dạng như các bộ tự chỉnh định chúng ta thường cố gắng đạt được một kiểu mẫu tương tự với

$$y(t) = \varphi_f^T\theta$$

Với phương pháp mô hình chuẩn thì thường xuyên chỉ có thể đạt một mô hình kiểu $y(t) = G(p)(\varphi_f^T\theta)$

Với $G(p)$ là SPR.

Ví dụ 3.15 - Bộ tự chỉnh định trực tiếp với thay đổi cực tiểu

Mô hình của quá trình ở phương trình 3.44 là :

$$y(t+1) = r_0u(t) + s_0y(t) + \varepsilon(t+1)$$

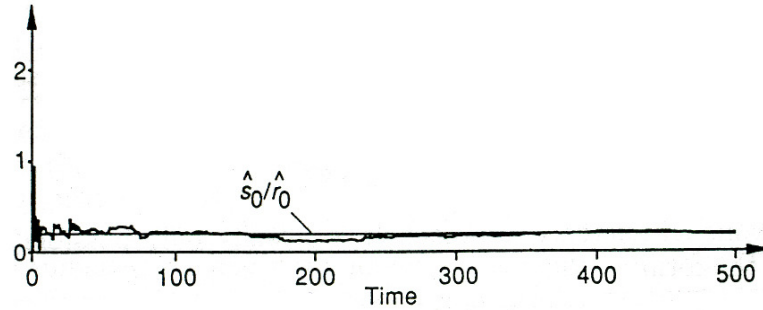
Giả sử r_0 cố định tới giá trị $\hat{r}_0 = 1$. Chú ý điều này khác với giá trị thật là bằng 3. Thông số s_0 được ước lượng dùng phương pháp bình phương cực tiểu.

Luật điều khiển trở thành:

$$u(t) = -\frac{\hat{s}_0}{\hat{r}_0} y(t)$$

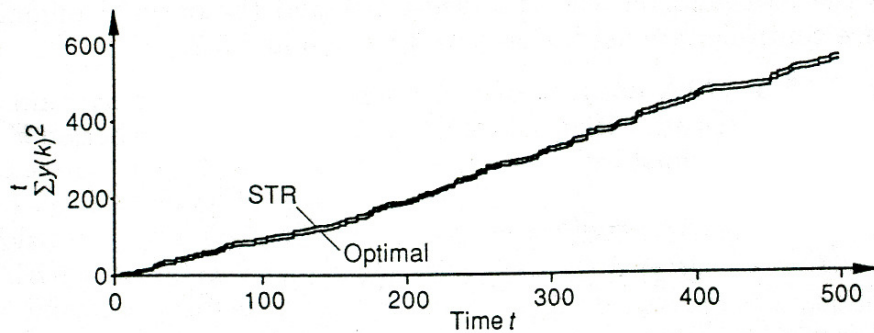
Mô phỏng cho ví dụ 3.15

Kết quả mô phỏng được biểu diễn ở hình 3.17 và hình 3.18



Hình 3.17

Hình 3.17 biểu diễn tỉ số \hat{s}_0 / \hat{r}_0 , nó nhanh chóng hội tụ đến một giá trị của bộ điều khiển thay đổi cực tiểu tối ưu thậm chí \hat{r}_0 không bằng giá trị thật của nó.



Hình 3.18

Hình 3.18 biểu diễn hàm tổn hao khi dùng bộ tự chỉnh định và bộ điều khiển thay đổi cực tiểu tối ưu.

3.3.5 Điều khiển dự báo thích nghi

Thuật toán 3.4 là cách để thực hiện một bộ điều khiển với tầm dự báo thay đổi. Bài toán điều khiển cơ bản là bộ điều khiển trung bình di chuyển. Bộ điều khiển trung bình di chuyển cũng có thể áp dụng được cho các hệ thống

không cực tiểu pha như được minh họa ở phần “Bộ chỉnh định trực tiếp”. Nhiều cách khác để có điều khiển dự báo sẽ được đề cập trong tài liệu, một vài trong số này sẽ được thảo luận và phân tích. Cũng như đối với các thuật toán trước, xác định bài toán điều khiển cơ bản là rất quan trọng để hiểu rõ các tính chất tiệm cận của thuật toán. Trước tiên ta sẽ phân tích trường hợp các thông số là biết trước.

Thuật toán điều khiển dự báo dựa trên một mô hình của quá trình giả thuyết và các tín hiệu điều khiển ở tương lai. Điều này tạo ra một chuỗi các tín hiệu điều khiển. Chỉ có một tín hiệu đầu tiên là được áp dụng cho quá trình và một chuỗi các tín hiệu điều khiển mới được tính toán khi thực hiện phép đo đặc mới. Đây gọi là bộ điều khiển lùi tầm (receding – horizon controller).

Dự báo ngõ ra

Ý tưởng cơ bản trong các thuật toán điều khiển dự báo là viết lại mô hình quá trình để có được một biểu thức rõ ràng cho ngõ ra ở một thời điểm tương lai. Xét mô hình :

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t-d_0) \quad (3.52)$$

$$1 = A^*(q^{-1})F^*(q^{-1}) + q^{-d}G^*(q^{-1}) \quad (3.53)$$

Trong đó

$$\text{bậc}(F_d^*) = d - 1$$

$$\text{bậc}(G_d^*) = n - 1$$

Chỉ số d là tầm dự báo với d bước. Giả sử $d \geq d_0$. Việc đồng nhất đa thức ở phương trình 3.52 được dùng để dự báo ngõ ra ở d bước phía trước. Vì vậy :

$$y(t+d) = A^*F_d^*y(t+d) + G_d^*y(t) = B^*F_d^*u(t+d-d_0) + G_d^*y(t)$$

$$B^*(q^{-1})F_d^*(q^{-1}) = R_d^*(q^{-1}) + q^{-(d-d_0+1)}\bar{R}_d^*(q^{-1})$$

$$\text{Bậc}(R_d^*) = d - d_0$$

$$\text{Bậc}(\bar{R}_d^*) = n - 2$$

Các hệ số của R_d^* là những giới hạn $d - d_0 + 1$ đầu tiên của đáp ứng xung của hệ thống vòng hở. Điều này có thể thấy như sau:

$$q^{-d_0}B^*/A^* = q^{-d_0}B^*(F_d^* + q^{-d}\frac{G_d^*}{A^*})$$

$$= q^{-d_0} R_d^*(q^{-1}) + q^{-(d+1)} \bar{R}_d^*(q^{-1}) + \frac{B^*(q^{-1})G_d^*(q^{-1})}{A^*(q^{-1})} q^{-(d+d_0)} \quad (3.54)$$

$$\begin{aligned} y(t+d) &= R_d^*(q^{-1}) u(t+d-d_0) + \bar{R}_d^*(q^{-1}) u(t-1) + G_d^*(q^{-1}) y(t) \\ &= R_d^*(q^{-1}) u(t+d-d_0) + \bar{y}_d(t) \end{aligned} \quad (3.55)$$

$R_d^*(q^{-1}) u(t+d-d_0)$ phụ thuộc vào $u(t), \dots, u(t+d-d_0)$, $\bar{y}_d(t)$ là hàm của $u(t-1), u(t-2), \dots$ và $y(t), y(t-1), \dots$. Biến $\bar{y}_d(t)$ được hiểu như là điều kiện dự báo của $y(t+d)$ với giả sử $u(t)$ và các tín hiệu điều khiển tương lai là zero. Ngõ ra ở thời điểm $(t+d)$ vì vậy phụ thuộc vào các tín hiệu điều khiển tương lai (nếu $d > d_0$), tín hiệu điều khiển, các ngõ vào và ngõ ra ở thời điểm trước. Cũng có thể giả sử tín hiệu điều khiển duy trì hằng số:

$$u(t) = u(t+d) = \dots = u(t+d-d_0) \quad (3.56)$$

Cách khác để xác định luật điều khiển là mang $y(t+d)$ đến một giá trị mong muốn trong khi cực tiểu mục tiêu điều khiển theo tầm dự báo:

$$\sum_{k=t}^{t+d} u(k)^2 \quad (3.57)$$

Điều khiển không thay đổi theo thời gian:

Chọn ngõ ra được dự báo bằng với ngõ ra mong muốn y_m và giả sử vẫn giữ phương trình 3.56 :

$$[R_d^*(1) + q^{-1} \bar{R}_d^*(q^{-1})] u(t) + G_d^*(q^{-1}) y(t) = y_m(t+d)$$

Luật điều khiển là:

$$u(t) = \frac{y_m(t+d) - G_d^*(q^{-1}) y(t)}{R_d^*(1) + R_d^*(q^{-1}) q^{-1}} \quad (3.58)$$

Tín hiệu điều khiển này sẽ được sử dụng cho quá trình. Ở lần lấy mẫu kế tiếp, một phép đo mới đạt được và luật điều khiển ở 3.58 được sử dụng tiếp. Chú ý giá trị của tín hiệu điều khiển thay đổi theo thời gian chứ không phải cố định. Ở đây ta sử dụng qui tắc điều khiển lùi tầm. Chú ý luật điều khiển là không đổi ngược với bộ điều khiển LQ cố định tầm.

Bây giờ chúng ta sẽ phân tích hệ thống vòng kín khi sử dụng phương trình 3.58 để điều khiển quá trình 3.52. Việc thực hiện các phép tính ở toán tử sai

phân tởi là cần thiết để có thể quan sát các cực ban đầu. Phương trình 3.53 được viết lại theo toán tử sai phân tởi như sau:

$$q^{n+d-1} = A(q)F_d(q) + G_d(q) \quad (3.59)$$

Đa thức đặc tính của hệ thống vòng kín là:

$$P(q) = A(q)[q^{n-1}R_d(1) + \bar{R}_d(q)] + G_d(q)B(q)$$

$$Bậc(P) = Bậc(A) + n - 1 = 2n - 1$$

Phương trình thiết kế 3.59 có thể được sử dụng để viết lại hàm P(q):

$$\begin{aligned} B(q)q^{n+d-1} &= A(q)B(q)F_d(q) + G_d(q)B(q) \\ &= A(q)[q^{n-1}R_d(q) + \bar{R}_d(q)] + G_d(q)B(q) \end{aligned}$$

Vì vậy :

$$A(q)\bar{R}_d(q) + G_d(q)B(q) = B(q)q^{n+d-1} - A(q)q^{n-1}R_d(q)$$

Cho ta :

$$P(q) = q^{n-1}A(q)R_d(1) + q^{n-1}[q^d B(q) - A(q)R_d(q)]$$

Nếu hệ thống ổn định thì các số hạng phía sau của 3.54 sẽ biến mất khi $d \rightarrow \infty$.

Do đó: $\lim_{d \rightarrow \infty} P(q) = q^{n-1}A(q)R_d(1)$ nếu A(z) là một đa thức ổn định.

Ví dụ 3.16 - Điều khiển dự báo

Xét quá trình :

$$y(t+1) = ay(t) + bu(t)$$

Phương trình 3.59 cho ta :

$$q^d = (q-a)(q^{d-1} + f_1q^{d-2} + \dots + f_{d-1}) + g_0$$

Vì vậy:

$$F(q) = q^{d-1} + aq^{d-2} + a^2q^{d-3} + \dots + a^{d-1}$$

$$G(q) = a^d$$

$$R_d(q) = bF(q)$$

$$\bar{R}_d(q) = 0$$

và khi $y_m = 0$, luật điều khiển trở thành:

$$u(t) = -\frac{a^d}{b(1+a+\dots+a^{d-1})}y(t) = -\frac{a^d(a-1)}{b(a^d-1)}y(t)$$

Phương trình đặc tính của hệ thống vòng kín là:

$$P(q) = q - a + \frac{a^d(a-1)}{a^d-1}$$

có cực:

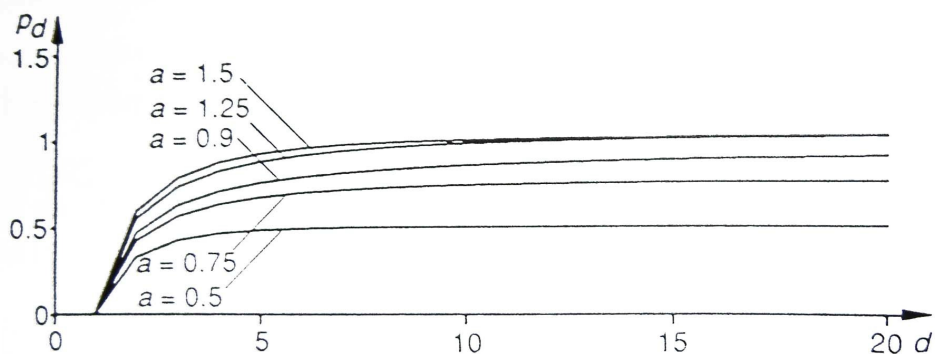
$$p_d = \frac{a^d - a}{a^d - 1}$$

Vị trí của cực được cho bởi:

$$0 \leq p_d < a \quad |a| \leq 1 \text{ (hệ thống ổn định)}$$

$$0 \leq p_d < 1 \quad |a| > 1 \text{ (hệ thống không ổn định)}$$

Cực vòng kín với các giá trị khác nhau của a và b được chỉ ở hình sau:



Ví dụ cũng cho thấy để việc quan sát là đầy đủ thì tầm dự báo phải từ 5 – 10 mẫu.

Cũng có thể tổng quát hoá kết quả ở ví dụ 3.16 cho các hệ thống bậc cao hơn. Đối với các hệ thống thay đổi chậm hoặc không ổn định thì đáp ứng vòng kín của nó sẽ rất chậm khi tăng tầm dự báo. Vì vậy giới hạn ở (3.56) khi đó sẽ là không hữu ích.

Nỗ lực điều khiển cực tiểu

Thuật toán điều khiển là sẽ mang $y(t+d)$ tới $y_m(t+d)$ trong khi cực tiểu phương trình 3.57. Phương trình 3.55 được viết lại:

$$\begin{aligned} y(t+d) &= R_d^*(q^{-1})u(t+d-d_0) + \bar{y}_d(t) \\ &= r_{d_0}u(t+v) + \dots + r_{dv}u(t) + \bar{y}_d(t) \end{aligned}$$

$v = d - d_0$. Giới thiệu hàm Lagrange:

$$2J = u(t)^2 + \dots + u(t+v)^2 + 2\lambda[y_m(t+d) - \bar{y}_d(t) - R_d^*(q^{-1})u(t+v)]$$

Cho đạo hàm riêng đối với các biến $u(t), \dots, u(t+v)$ và λ bằng 0 ta được:

$$u(t) = \lambda r_{dv}$$

.

.

.

$$u(t+v) = \lambda r_{d_0}$$

$$y_m(t+d) - \bar{y}_d(t) = r_{d_0}u(t+v) + \dots + r_{dv}u(t)$$

Các phương trình này cho ta:

$$u(t) = \frac{y_m(t+d) - \bar{y}_d(t)}{\mu}$$

trong đó:

$$\mu = \frac{\sum_{i=0}^v r_{di}^2}{r_{dv}}$$

Sử dụng định nghĩa $\bar{y}_d(t)$ cho ta:

$$\mu u(t) = y_m(t+d) - \bar{R}_d^* u(t-1) - G_d^* y(t)$$

hoặc

$$u(t) = \frac{y_m(t+d) - G_d^* y(t)}{\mu + q^{-1} \bar{R}_d^*} = \frac{y_m(t+d+n-1) - G_d(q)y(t)}{\mu q^{n-1} + \bar{R}_d(q)} \quad (3.60)$$

Sử dụng phương trình 3.60 và mô hình của phương trình 3.52 cho đa thức đặc tính vòng kín:

$$P(q) = A(q) [q^{n-1} \mu + \bar{R}_d(q)] + G_d(q)B(q)$$

Phương trình này có dạng như 3.58 với $R_d(1)$ được thay bằng μ . Điều này có nghĩa các cực vòng kín tiến gần tới zero của $q^{n-1}A(q)$ khi $A(q)$ là ổn định

và khi $d \rightarrow \infty$. Điều gì sẽ xảy ra khi hệ thống không ổn định. Hãy xét ví dụ sau đây:

Ví dụ 3.17 - Điều khiển nỗ lực cực tiểu

Xét hệ thống tương tự như ví dụ 3.16. Bộ điều khiển nỗ lực cực tiểu trong trường hợp này được cho bởi:

$$\mu = b \frac{1 + a^2 + \dots + a^{2(d-1)}}{a^{d-1}} = \frac{b(a^{2d} - 1)}{a^{d-1}(a^2 - 1)}$$

cho ta (khi $y_m = 0$)

$$u(t) = -\frac{a^d}{\mu} y(t) = -\frac{a^{2d-1}(a^2 - 1)}{b(a^{2d} - 1)} y(t)$$

Cực của hệ thống vòng kín là:

$$p_d = a - \frac{a^{2d-1}(a^2 - 1)}{a^{2d} - 1} = \frac{a^{2d-1} - a}{a^{2d} - 1}$$

cho ta:

$$\lim_{d \rightarrow \infty} p_d = a \quad |a| \leq 1 \text{ (hệ thống ổn định)}$$

$$\lim_{d \rightarrow \infty} p_d = 1/a \quad |a| > 1 \text{ (hệ thống không ổn định)}$$

Ở ví dụ này, bộ điều khiển nỗ lực cực tiểu sẽ tạo ra một hệ thống vòng kín tốt hơn nếu điều khiển tương lai được giả sử là hằng số.

Điều khiển dự báo tổng quát:

Các bộ điều khiển dự báo đề cập từ trước chỉ xem xét giá trị ngõ ra chỉ ở một thời điểm ở tương lai. Nhiều tổng quát hoá khác nhau của điều khiển dự báo được đề xuất mà trong đó hàm tổn hao là cực tiểu:

$$J(N_1, N_2, N_u) = E \left\{ \sum_{k=N_1}^{N_2} [y(t+k) - y_m(t+k)]^2 + \sum_{k=1}^{N_u} \rho \Delta u(t+k-1)^2 \right\} \quad (3.61)$$

Trong đó $\Delta = 1 - q^{-1}$ là toán tử vi phân. Sự lựa chọn các giá trị khác nhau của N_1, N_2, N_u sẽ đưa ra các phương pháp khác nhau.

Phương pháp điều khiển dự báo tổng quát được minh hoạ bằng cách dùng hàm tổn hao 3.60 và mô hình quá trình:

$$A^*(q)y(t) = B^*(q^{-1})u(t-d_0) + e(t) / \Delta \quad (3.62)$$

Mô hình này được gọi là CARIMA (Controlled AutoRegressive Intergrating Moving Average). Nó có thuận lợi là bộ điều khiển bản thân sẽ chứa một khâu tích phân. Giống như phương trình 3.53 ta có đồng nhất:

$$1 = A^*(q)F_d^*(q^{-1})(1 - q^{-1}) + q^{-d}G_d^*(q^{-1}) \quad (3.63)$$

Công thức này được sử dụng để xác định ngõ ra ở d bước kế tiếp:

$$y(t + d) = F_d^*B^* \Delta u(t + d - d_0) + G_d^*y(t) + F_d^*e(t + d)$$

F_d^* có bậc $d - 1$. Bộ dự báo với sai số quân phương tối ưu với ngõ ra được đo đặc đến thời điểm t và chuỗi ngõ vào bất kì là:

$$\hat{y}(t + d) = F_d^*B^* \Delta u(t + d - d_0) + G_d^*y(t) \quad (3.64)$$

Giả sử đầu ra mong muốn $y_m(t + k)$, $k = 1, 2, \dots$ là có sẵn. Hàm tổn hao ở 3.61 sẽ được cực tiểu để cho ra một chuỗi các tín hiệu điều khiển ở tương lai. Chú ý giá trị mong đợi ở 3.61 sẽ có được tương ứng với dữ liệu có được tới thời điểm t với giả sử các đo đặc ở tương lai không có sẵn. Điều này có nghĩa chỉ có thừa số đầu tiên của chuỗi điều khiển là được sử dụng. Các phép toán sẽ lặp lại khi có được một đo đặc mới. Bộ điều khiển với kết quả như thế gọi là điều khiển hồi tiếp tối ưu vòng hở. Như tên của nó, giả sử sử dụng hồi tiếp nhưng nó chỉ được tính toán chỉ dựa vào thông tin có sẵn ở thời điểm hiện tại. Dùng phương trình 3.55 :

$$y(t + 1) = R_1^*(q^{-1}) \Delta u(t + 1 - d_0) + \bar{y}_1(t) + F_1^* e(t + 1)$$

$$y(t + 2) = R_2^*(q^{-1}) \Delta u(t + 2 - d_0) + \bar{y}_2(t) + F_2^* e(t + 2)$$

.
.
.

$$y(t + N) = R_N^*(q^{-1}) \Delta u(t + N - d_0) + \bar{y}_N(t) + F_N^* e(t + N)$$

Mỗi giá trị ngõ ra bao gồm các tín hiệu điều khiển ở tương lai (nếu $d > d_0$), ngõ vào đo được và tín hiệu nhiễu ở tương lai. Các phương trình ở trên có thể được viết lại:

$$y = R\Delta u + \bar{y} + e$$

trong đó:

$$y = [y(t + 1) \dots y(t + N)]^T$$

$$\Delta u = [\Delta u(t + 1 - d_0) \dots \Delta u(t + N - d_0)]^T$$

$$\bar{y} = [\bar{y}_1(t) \dots \bar{y}_N(t)]^T$$

$$e = [F_1^* e(t+1) \dots F_N^* e(t+N)]^T$$

Từ phương trình 3.54 ta thấy các hệ số của R_d^* chính là $(d - d_0 + 1)$ số hạng đầu của đáp ứng xung $q^{-d_0} B^* / (A^* \Delta)$ và cũng giống như $(d - d_0 + 1)$ số hạng đầu của đáp ứng bước $q^{-d_0} B^* / A^*$. Do đó ma trận R là ma trận tam giác dưới:

$$R = \begin{bmatrix} r_0 & 0 & \dots & 0 \\ r_1 & r_0 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ r_{N-1} & r_{N-2} & \dots & r_0 \end{bmatrix}$$

Nếu hệ thống có thời gian trễ ($d_0 > 1$) thì $(d_0 - 1)$ hàng đầu của R sẽ là zero. Gọi:

$$y_m = [y_m(t+1) \dots y_m(t+N)]^T$$

Giá trị mong đợi của hàm tổn hao được viết lại:

$$\begin{aligned} J(I, N, N) &= E\{(y - y_m)^T (y - y_m) + \rho \Delta u^T \Delta u\} \\ &= (R \Delta u + \bar{y} - y_m)^T (R \Delta u + \bar{y} - y_m) + \rho \Delta u^T \Delta u \end{aligned}$$

Cực tiểu hoá biểu thức này theo Δu ta được:

$$\Delta u = (R^T R + \rho I)^{-1} R^T (y_m - \bar{y}) \quad (3.65)$$

Thành phần đầu trong Δu là $\Delta u(t)$ là tín hiệu điều khiển ứng dụng cho hệ thống. Chú ý bộ điều khiển tự động có một khâu tích phân. Điều này là cần thiết để bù cho số hạng nhiễu sai lệch ở phương trình 3.62

Việc tính toán phương trình 3.65 liên quan tới ma trận nghịch đảo $N \times N$, mà N là tầm dự báo của hàm tổn hao. Để giảm khối lượng tính toán thì ta có thể giới hạn các tín hiệu điều khiển ở tương lai. Chẳng hạn, ta giả sử việc tăng tín hiệu điều khiển là bằng zero sau N_u bước ($N_u < N$):

$$\Delta u(t+k-1) = 0 \text{ với } k > N_u$$

Điều này có nghĩa tín hiệu điều khiển sau N_u bước sẽ là hằng số. So sánh với điều kiện không chế ở phương trình 3.57. Luật điều khiển (phương trình 3.65) sẽ thay đổi:

$$\Delta u = (R_1^T R_1 + \rho I)^{-1} R_1^T (y_m - \bar{y}) \quad (3.66)$$

R_1 là ma trận $N \times N_u$

$$R_1 = \begin{bmatrix} r_0 & 0 & \dots & 0 \\ r_1 & r_0 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ \vdots & & \dots & r_0 \\ r_{N-1} & r_{N-2} & \dots & r_{N-N_u} \end{bmatrix}$$

Ma trận lấy nghịch đảo bây giờ có bậc $N_u \times N_u$. Ngõ ra và các tầm điều khiển được chọn như sau:

N_1 : Nếu thời gian trễ biết trước thì $N_1 = d_0$, ngược lại chọn $N_1 = 1$.

N_2 : Tầm ngõ ra cực đại N_2 được chọn sao cho $N_2 h$ có giá trị bằng với thời gian lên của hệ thống, trong đó h là thời gian lấy mẫu của bộ điều khiển.

N_u : Thường $N_u = 1$ sẽ có được kết quả tốt đối với những hệ thống đơn giản. Đối với các hệ thống phức tạp, N_u ít nhất phải bằng với số cực không ổn định hoặc số cực gây dao động tắt yếu.

Để bộ điều khiển dự báo tổng quát có khả năng thích nghi thì điều cần thiết là phải ước lượng A^* và B^* ở mỗi bước thời gian. Các giá trị dự báo ứng với các tầm dự báo khác nhau sẽ được tính toán và tính tín hiệu điều khiển ở phương trình 3.66. Bộ điều khiển dự báo thích nghi vì vậy sẽ là một thuật toán điều khiển gián tiếp. Phương trình 3.64 được tính bằng cách đệ quy để đơn giản khối lượng tính toán. Cuối cùng, N_u thường có giá trị nhỏ để ma trận nghịch đảo có bậc thấp.

Tín hiệu điều khiển $\Delta u(t)$ từ phương trình 3.66 là:

$$\begin{aligned} \Delta u &= [1 \ 0 \ \dots \ 0] [R_1^T R_1 + \rho I]^{-1} R_1^T [y_m - \bar{y}] \\ &= [\alpha_1 \ \dots \ \alpha_N] [y_m - \bar{y}] \end{aligned}$$

Hơn nữa, từ phương trình 3.62, sử dụng phương trình 3.54

$$\bar{y} = \begin{bmatrix} \bar{R}_1^* \Delta u(t-1) + G_1^* y(t) \\ \vdots \\ \bar{R}_N^* \Delta u(t-1) + G_N^* y(t) \end{bmatrix} = \begin{bmatrix} \frac{\bar{R}_1^* A^* \Delta}{B^*} q^{d_0-1} + G_1^* \\ \vdots \\ \frac{\bar{R}_N^* A^* \Delta}{B^*} q^{d_0-1} + G_N^* \end{bmatrix} y(t)$$

Hệ thống vòng kín có phương trình đặc tính:

$$A^* \Delta + [\alpha_1 \dots \alpha_N] \begin{bmatrix} \bar{R}_1^* A^* \Delta q^{d_0-1} + B^* G_1^* \\ \vdots \\ \bar{R}_N^* A^* \Delta q^{d_0-1} + B^* G_N^* \end{bmatrix}$$

Đồng nhất phương trình 3.63 cho ta:

$$\begin{aligned} B^* &= A^* \Delta B^* F_d + q^{-d} G_d^* B^* \\ &= A^* \Delta [R_d^* + q^{-(d-d_0+1)} \bar{R}_d^*] + q^{-d} G_d^* B^* \end{aligned}$$

Điều này cho ta phương trình đặc tính:

$$\begin{aligned} A^* \Delta + [\alpha_1 \dots \alpha_N] \begin{bmatrix} (B^* - A^* \Delta R_1^*) q \\ \vdots \\ (B^* - A^* \Delta R_N^*) q^N \end{bmatrix} \\ = A^* \Delta + \sum_{i=1}^N \alpha_i q^i (B^* - A^* \Delta R_i^*) \end{aligned} \quad (3.67)$$

Phương trình 3.67 cho ra một biểu thức của phương trình đặc tính vòng kín nhưng vẫn còn khó khăn để đưa ra một kết luận tổng quát về tính chất của hệ thống vòng kín ngay cả khi quá trình đã biết trước.

Nếu $N_u = 1$ thì:

$$\alpha_i = \frac{r_i}{\rho + \sum_{j=1}^N r_j^2}$$

Nếu ρ đủ lớn, hệ thống vòng kín sẽ không ổn định khi hệ thống vòng hở không ổn định. Tuy nhiên nếu cả 2 tầm điều khiển và tầm dự báo đều tăng thì bài toán sẽ tương tự như bài toán điều khiển LQ với tầm cố định và do đó nó sẽ có đặc tính ổn định tốt hơn.

3.3.6 Kết luận

Trong phần này chúng ta đã xem xét nhiều bộ tự chỉnh định khác nhau. Ý tưởng cơ bản là ước lượng các thông số chưa biết của hệ thống và thiết kế bộ điều khiển. Các thông số ước lượng giả sử bằng với thông số thực khi thiết kế bộ điều khiển. Tình huống cũng bao gồm các ước lượng chưa chắc chắn vào trong thiết kế. Bằng cách kết hợp các phương pháp ước lượng khác

nhau và các phương pháp thiết kế khác nhau ta sẽ có được các bộ tự chỉnh với các tính chất khác nhau. Trong phần này ta chỉ đề cập ý tưởng cơ bản và các tính chất tiệm cận. Tính hội tụ của ước lượng và tính ổn định của hệ thống sẽ được thảo luận trong chương 6 (TLTK[1]).

Khía cạnh quan trọng nhất của các bộ tự chỉnh định là đưa ra các thông số hoá. Một thông số hoá lại có thể đạt được bằng cách sử dụng mô hình hệ thống và đáp ứng vòng kín mong muốn. Mục tiêu của việc thông số hoá lại là để thực hiện ước lượng trực tiếp các thông số của bộ điều khiển sao cho mô hình mới tuyến tính với các thông số.

Chỉ có vài thuật toán tự chỉnh định được đề cập và giải quyết trong phần này. Việc kết hợp các phương pháp ước lượng khác nhau và vấn đề thiết kế cơ bản sẽ tạo ra các thuật toán với các tính chất khác nhau. Mục tiêu của phần này là đưa ra một cảm nhận cách phát triển và phân tích các thuật toán. Khi thực hiện một bộ tự chỉnh thì việc lựa chọn bài toán thiết kế cơ bản là rất quan trọng. Một phương pháp thiết kế mà không phù hợp cho hệ thống biết trước thì cũng sẽ không tốt hơn khi hệ thống chưa biết trước.

Bộ tự chỉnh định cũng có khả năng áp dụng cho các hệ thống MIMO. Trường hợp MIMO là rất khó để phân tích. Khó khăn chính là xác định được kiến thức đầu tiên cần thiết trong hệ MIMO là gì.

Cũng tương đối đơn giản khi đưa ra một thuật toán tự chỉnh tương ứng với bộ tự chỉnh định trực tiếp tổng quát ở các trường hợp hạn chế khi các ma trận tương tác của hệ thống đã biết trước.

3.4 Chỉnh định tự động và lịch trình độ lợi

3.4.1 Giới thiệu

1. Chỉnh định và thích nghi
2. Kiến thức đầu tiên
3. Giá trị ban đầu của bộ điều khiển thích nghi
4. Điều khiển PID
5. Các vấn đề vận hành
6. Giao tiếp điều khiển

Một loại đặc biệt của thích nghi vòng hở hay sự thay đổi các tham số bộ điều chỉnh được đề cập trong phần này. Trong nhiều trường hợp, có thể biết được sự thay đổi động học của quá trình theo các điều kiện vận hành. Nguồn gốc của sự thay đổi động học có thể là tính phi tuyến. Có thể thay đổi tham số của bộ điều khiển bằng cách giám sát các điều kiện vận hành

của quá trình. Khái niệm này gọi là lịch trình độ lợi, vì mô hình đầu tiên được sử dụng chỉ để điều chỉnh độ lợi của quá trình.

3.4.2 Kỹ thuật chỉnh định

1. Phương pháp Zeigler – Nichols

Luật điều khiển PID:

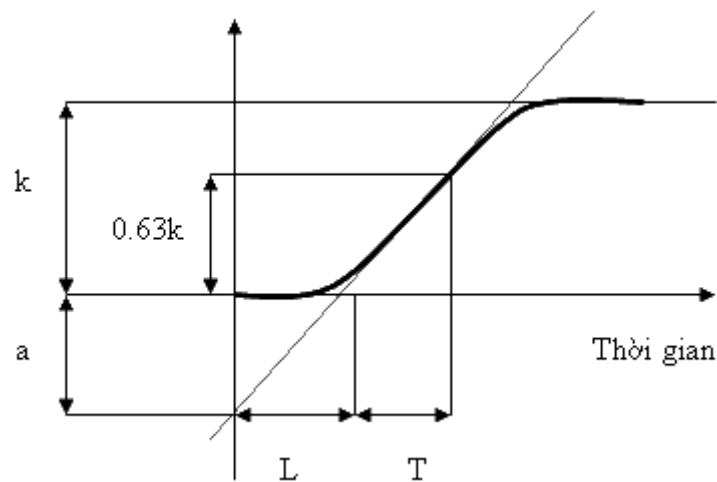
$$u(t) = K_c \left[e(t) + \frac{1}{T_i} \int_0^t e(s) ds + T_d \frac{de}{dt} \right]$$

2. Phương pháp đáp ứng quá độ

Mô hình 3 thông số:

$$G(s) = \frac{k}{1 + sT} e^{-sL}$$

Phương pháp đáp ứng nấc:



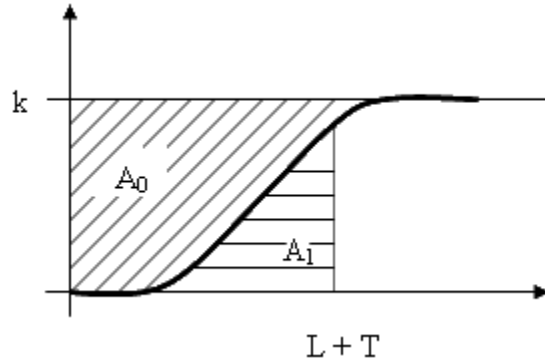
Phương pháp Zeigler – Nichols:

Bộ điều khiển	aK_c	T_i / L	T_d / L	T_p / L
P	1			4
PI	0.9	3		5.7
PID	1.2	2	0.5	3.4

Những khó khăn đối với phương pháp Zeigler – Nichols:

- Khó xác định các thông số
- Tất quá chậm
- Hai thông số thì không đủ

Phương pháp diện tích:

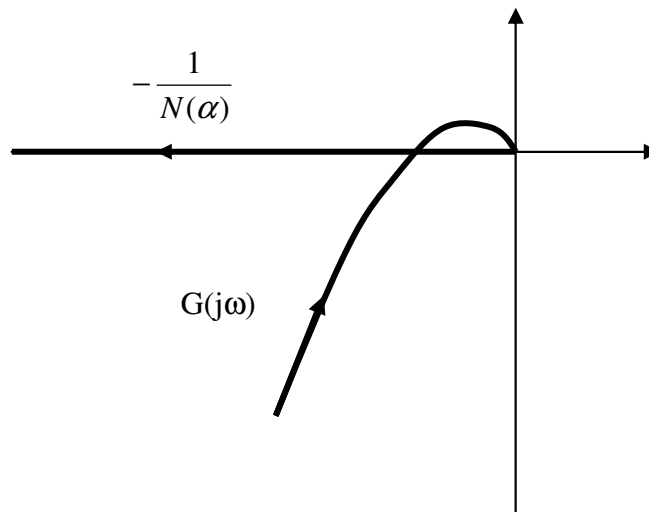


$$T + L = \frac{A_0}{k}$$
$$T = \frac{eA_1}{k}$$

3. Phương pháp đáp ứng tần số

Ý tưởng: Cho chạy bộ điều khiển tỉ lệ, tăng độ lợi cho đến khi hệ thống bắt đầu dao động. Quan sát “Độ lợi K_u giới hạn” và “Chu kỳ giới hạn T_u ”.

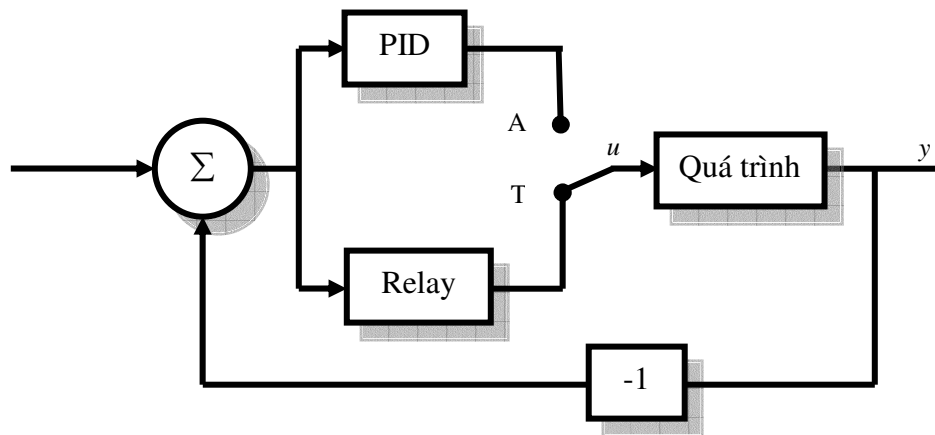
Lặp lại: Xác định đặc tính đáp ứng tần số.



Các thông số bộ điều khiển:

Bộ điều khiển	K_c / K_u	T_i / T_u	T_d / T_u	T_p / T_u
P	0.5			1
PI	0.4	0.8		1.4
PID	0.6	0.5	0.12	0.85

Thực nghiệm:



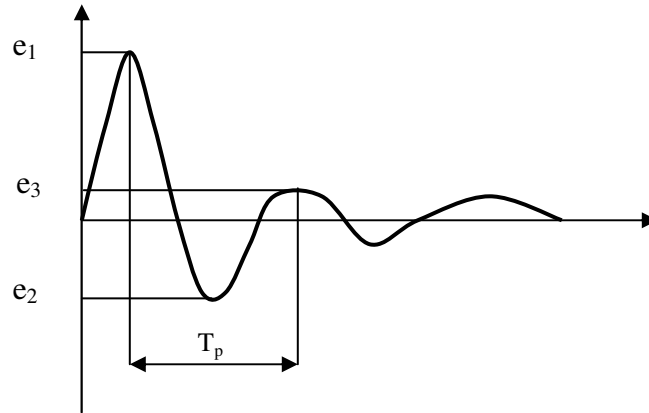
Kết quả thực tế

- Thông tin biết trước?
- Bắt đầu thực nghiệm như thế nào?
- Hồi tiếp đến biên độ giới hạn của dao động.
- Hiệu chỉnh luật Zeigler – Nichols:
 - Thay đổi các giá trị trong bảng.
 - Sử dụng 3 thông số: K_u , T_u và K_p .
- Làm sao để đưng đầu với nhiễu được
 - Nhiễu tải
 - Nhiễu đo
 - Trì trễ

Sự lặp lại trực tuyến

Ý tưởng: Tìm các nét đặc trưng của đáp ứng trực tuyến đối với điểm đặt hoặc các nhiễu tải.

Hiệu chỉnh bộ điều khiển dựa trên các đặc tính quan sát được.



Đặc tính: hệ số tắt ξ và độ vọt lố σ

$$\xi = \frac{e_3 - e_2}{e_1 - e_2} \qquad \sigma = -\frac{e_2}{e_1}$$

Bộ điều khiển hiệu chỉnh dựa trên luật thử và sai.

Dễ dàng đối với PI và khó khăn hơn đối với PID.

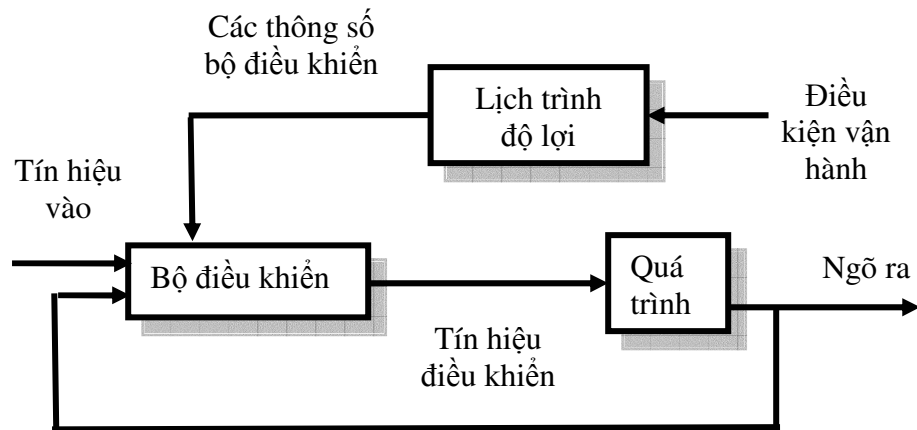
- Thông tin biết trước
- Tiền chỉnh định

3.4.3 Lịch trình độ lợi

Ví dụ các biến lịch trình

- Tốc độ sản xuất
- Tốc độ máy
- Số tỉ lệ và áp lực động

Thỉnh thoảng có thể tìm thấy những biến đổi phụ có tương quan tốt với những thay đổi của quá trình động học. Vì thế có thể làm giảm ảnh hưởng của tham số biến động chỉ đơn giản bằng việc thay đổi tham số của bộ điều chỉnh như các hàm của các biến phụ (xem hình 3.10)



Hình 3.20 Mô hình lịch trình độ lợi

Lịch trình độ lợi có thể được xem như hệ thống điều khiển hồi tiếp mà độ lợi hồi tiếp được chỉnh bởi bộ bù được cung cấp trước.

Ưu, khuyết điểm của lịch trình độ lợi

Mặt hạn chế của lịch trình độ lợi là bù vòng hở. Không có hồi tiếp để bù cho sai số lịch trình. Hạn chế khác của lịch trình độ lợi là việc thiết kế tốn nhiều thời gian. Tham số bộ điều chỉnh phải được chọn cho nhiều điều kiện vận hành và đặc tính kỹ thuật phải được kiểm tra bằng nhiều quá trình mô phỏng. Những khó khăn này tránh được nếu lịch trình dựa vào các phép chuyển đổi phi tuyến.

Lịch trình độ lợi có ưu điểm là các tham số bộ điều chỉnh có thể đáp ứng rất nhanh với sự thay đổi của quá trình. Khi không có ước lượng tham số, nhân tố giới hạn phụ thuộc vào tốc độ đáp ứng các phép đo phụ với sự thay đổi của quá trình.

3.4.4 Xây dựng lịch trình

- Lựa chọn các biến lịch trình
- Hoàn thiện việc thiết kế điều khiển cho những điều kiện vận hành khác nhau.
- Sử dụng việc chỉnh định tự động.
- Sự biến đổi.

Thật khó để tìm luật chung cho việc thiết kế bộ điều chỉnh theo lịch trình độ lợi. Vấn đề chính là việc quyết định các biến sử dụng làm biến lịch trình. Rõ ràng các tín hiệu phụ phải phản ánh điều kiện vận hành của đối tượng. Sẽ có những trình bày lí tưởng đơn giản cho các tham số bộ điều chỉnh liên quan đến các biến lịch trình. Vì thế cần có kiến thức tốt về hệ động học của quá trình nếu lịch trình độ lợi được sử dụng. Các khái niệm tổng quát sau có thể phục vụ cho mục đích này.

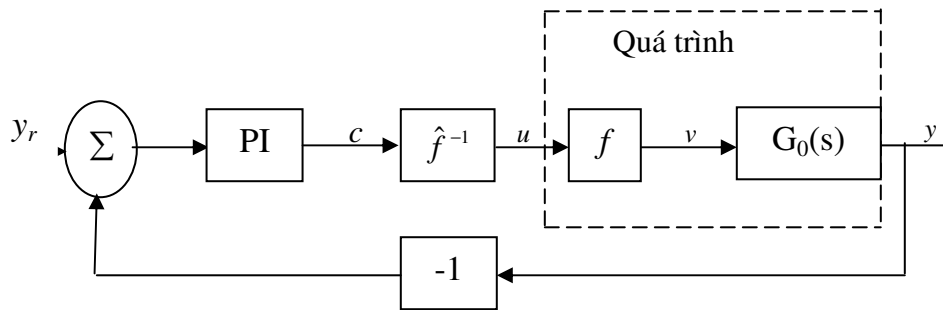
- Tuyến tính hoá cơ cấu dẫn động phi tuyến.
- Lập trình độ lợi dựa vào đo đạc các biến phụ
- Vận hành dựa vào hiệu suất
- Các phép biến đổi phi tuyến.

Các khái niệm này được minh hoạ trong các ví dụ sau.

Ví dụ 3.18

Xem hệ thống với 1 valse phi tuyến. Tính phi tuyến được giả sử là:

$$v = f(u) = u^4, \quad u \geq 0$$

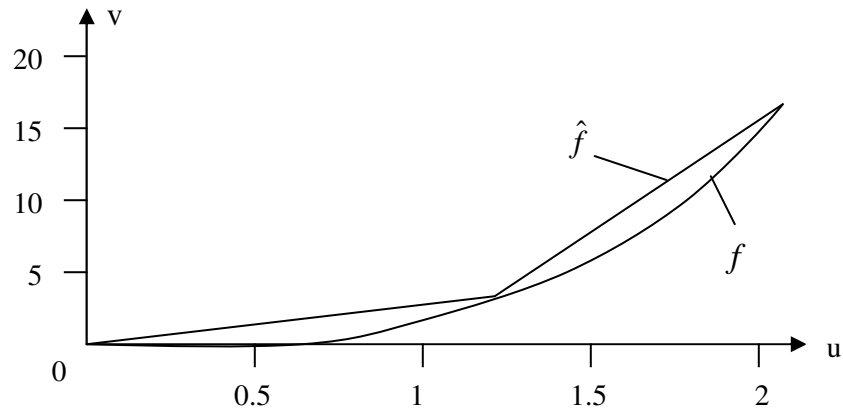


Hình (a)

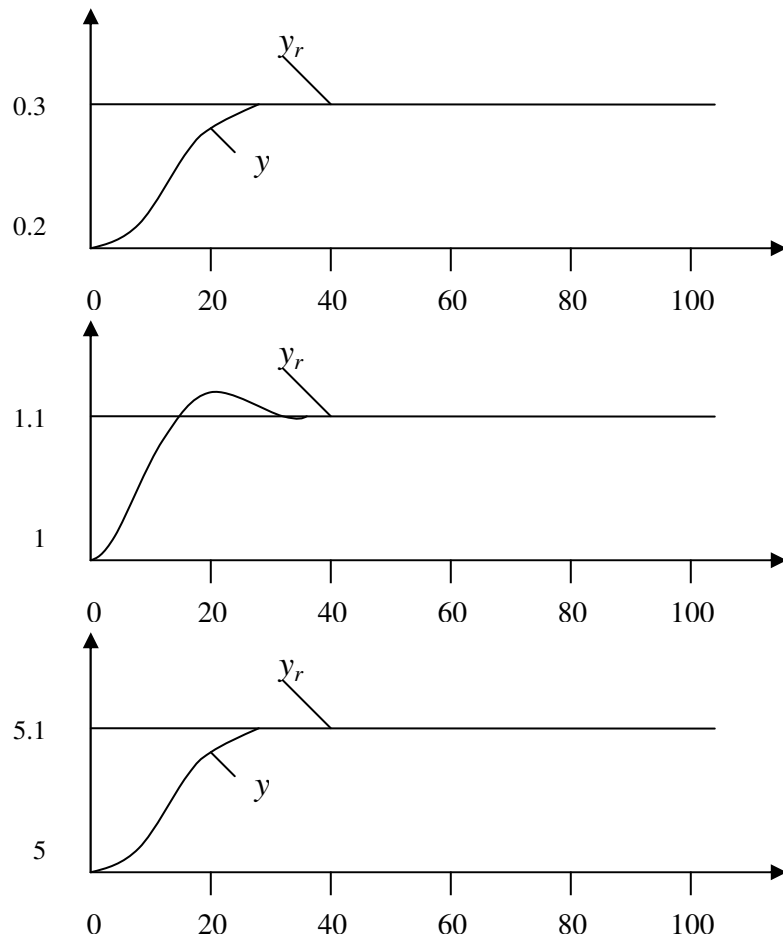
Đặt \hat{f}^{-1} là hàm ngược xấp xỉ của đặc tính valse. Để bù cho tính phi tuyến, ngõ ra của bộ điều chỉnh được cung cấp thông qua hàm này trước khi nó được áp vào valse (xem hình (a)). ta có quan hệ : $v = f(u) = f[\hat{f}^{-1}(c)]$

Với c là ngõ ra của bộ điều chỉnh PI. Hàm $f[\hat{f}^{-1}(c)]$ có độ lợi ít thay đổi hơn hàm f.

Nếu \hat{f}^{-1} chính xác là hàm ngược của f thì : $v = c$. Giả sử $f(u) = u^4$ được xấp xỉ bởi 2 đường thẳng: một đường nối từ điểm (0, 0) đến điểm (1.3, 3) và đường thẳng thứ hai nối giữa 2 điểm (1.3, 3) và (2, 16), được vẽ trong hình (b) .



Hình (b)



Hình (c)

Khi đó:

$$\hat{f}^{-1}(c) = \begin{cases} 0.433c & , 0 \leq c \leq 3 \\ 0.0538c + 1.139 & , 3 \leq c \leq 16 \end{cases}$$

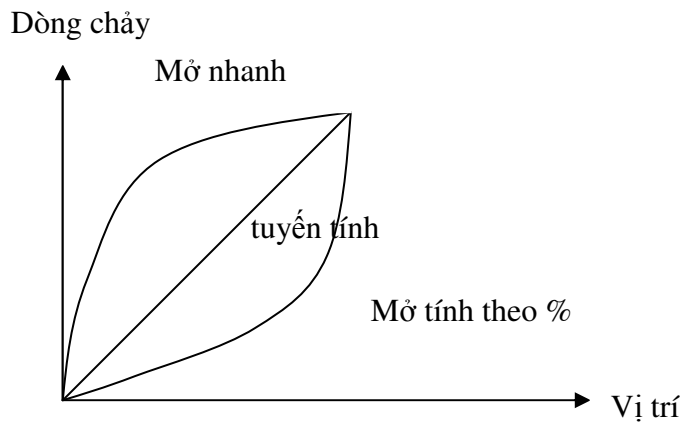
Hình (c) cho thấy sự thay đổi của đáp ứng quá độ tại 3 điều kiện vận hành khác nhau khi sử dụng hàm \hat{f}^{-1} như hình (a). Ta thấy đặc tuyến của hệ thống vòng kín là rất tốt. Dùng hàm ngược \hat{f}^{-1} trong hệ thống sẽ cho đáp ứng bằng phẳng hơn trong các bài toán điều khiển valse phi tuyến.

Ví dụ trên đã cho thấy tính đơn giản và tiện dụng trong việc bù cho hệ thống phi tuyến tĩnh biết trước. Trong thực tế thường xấp xỉ hệ phi tuyến bằng một vài đoạn thẳng (nhiều hơn 2). Có nhiều bộ điều khiển vòng đơn thương mại sử dụng phương pháp bù này. Trong ví dụ trên không có sự đo đạc nào của điều kiện vận hành ngoài trừ việc điều chỉnh ngõ ra. Trong các trường hợp khác, tính phi tuyến được xác định từ sự đo đạc một vài biến số.

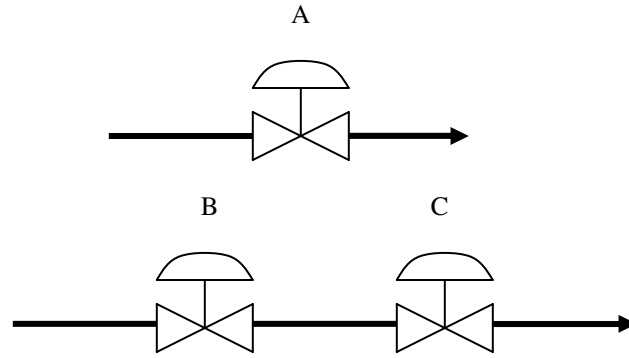
3.4.5 Ứng dụng

Lịch trình độ lợi là phương pháp rất hữu dụng. Nó yêu cầu phải có kiến thức tốt về quá trình và các biến phụ có thể được đo đạc. Một thuận lợi lớn của phương pháp này là bộ điều chỉnh thích nghi (đáp ứng) nhanh khi các điều kiện thay đổi. Một số ứng dụng như: định hướng cho tàu, kiểm soát nồng độ pH, kiểm soát khí đốt, điều khiển động cơ và điều khiển bay.

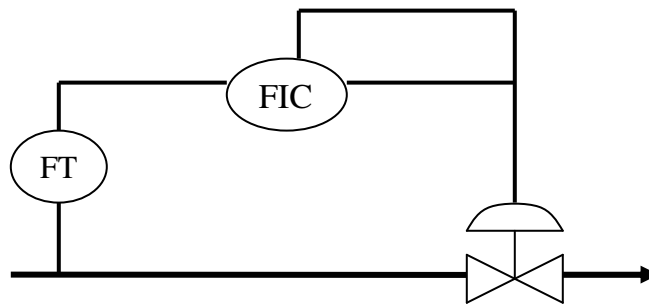
Đặc điểm của van



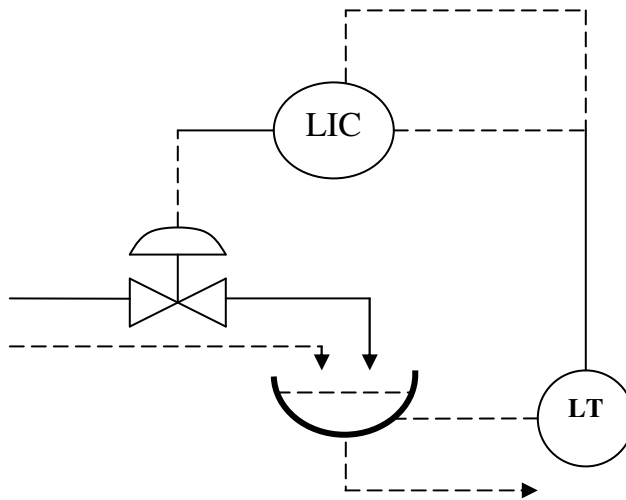
Đặc tính van phụ thuộc vào việc cài đặt.



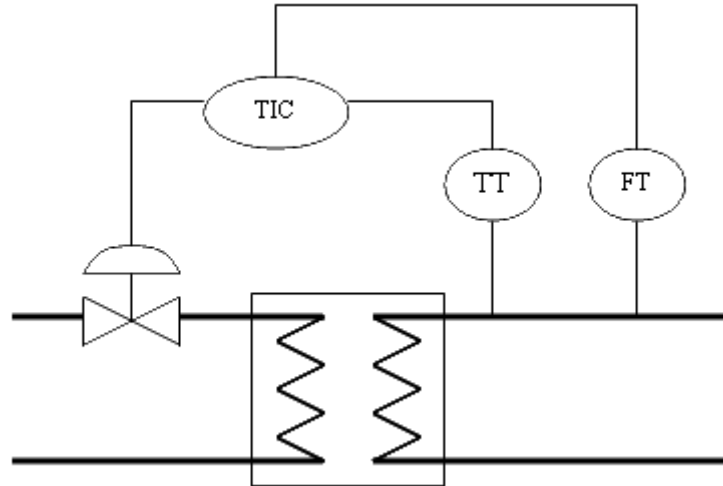
Lịch trình cho ngõ ra bộ điều khiển



Lịch trình cho biến quá trình



Lịch trình cho biến ngoài



3.4.6 Kết luận

Lịch trình độ lợi là cách tốt để bù cho đặc tính phi tuyến biết trước. Bộ điều chỉnh có thể phản ứng nhanh với sự thay đổi của các điều kiện. Mặt hạn chế của kĩ thuật này là thiết kế tốn nhiều thời gian nếu không dùng phép chuyển đổi phi tuyến và tự động chỉnh định. Mặt hạn chế khác là các tham số điều khiển được thay đổi trong vòng hở, không có hồi tiếp từ đặc tính làm việc của hệ thống. Phương pháp này không thể dùng được nếu đặc tính động học của quá trình hoặc nhiễu không được biết trước đầy đủ, chính xác.

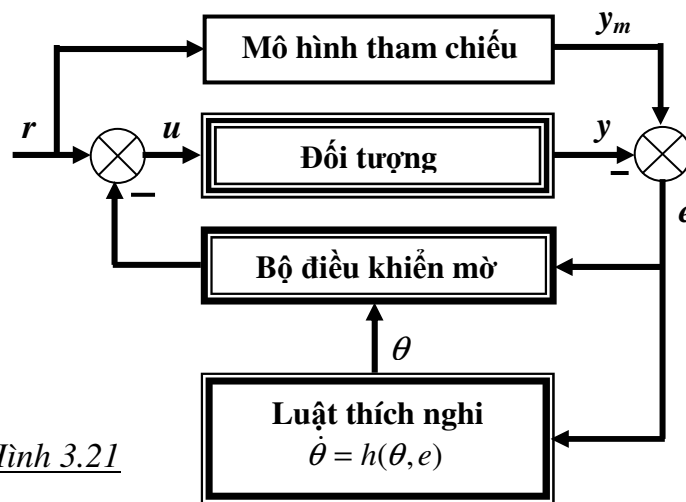
3.5. Điều khiển mờ thích nghi (Adaptive Fuzzy Control)

Trong thực tế, hầu như các đối tượng đều có thông số thay đổi không dự đoán trước được do ảnh hưởng của điều kiện làm việc. Nếu ta chỉ dùng các bộ điều khiển mờ không thích nghi (đã đề cập trong phần 1.2.5_chương 1) để điều khiển hệ thống thì sẽ không khống chế tốt đối tượng. Vì vậy, điều khiển mờ cần phải có tính thích nghi để tạo nên một hệ thống điều khiển trong đó thông số và cấu trúc của bộ điều khiển thay đổi trong quá trình vận hành nhằm giữ vững chất lượng điều khiển của hệ thống khi có sự hiện diện của các yếu tố bất định cũng như sự thay đổi thông số trong hệ thống.

So với điều khiển mờ không thích nghi thì điều khiển mờ thích nghi có những ưu điểm và khuyết điểm sau:

- Ưu điểm:
 - Khả năng thích nghi tốt do nó có thể tự điều chỉnh để thích nghi với sự thay đổi của môi trường.
 - Không cần thông tin đầy đủ về đối tượng do luật thích nghi sẽ tự học các thuộc tính động của đối tượng trong quá trình vận hành.
- Khuyết điểm:
 - Rất khó trong việc phân tích kết quả điều khiển vì tính phi tuyến và sự thay đổi liên tục theo thời gian của hệ thống.
 - Chi phí lắp đặt cao hơn.

• Mô hình cơ bản của BDK mờ thích nghi:



Hình 3.21

Trong đó:

- Mô hình tham chiếu: xác định đáp ứng mong đợi của hệ thống.
- Đối tượng: chứa các thành phần thông số và cấu trúc chưa biết.
- Bộ điều khiển mờ: là hệ thống mờ với thông số θ điều chỉnh được.
- Luật thích nghi: dùng để điều chỉnh thông số θ sao cho ngõ ra của đối tượng $y(t)$ tiến đến ngõ ra $y_m(t)$ của mô hình tham chiếu.

3.5.1. Điều khiển hồi tiếp tuyến tính hoá:

Xét bài toán điều khiển hệ phi tuyến SISO bậc n có quan hệ vào/ra được mô tả bởi phương trình vi phân sau:

$$x^{(n)} = f(x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}) + g(x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}) \cdot u(x) \quad (3.68)$$

$$y = x \quad (3.69)$$

Trong đó:

$f(x)$ và $g(x)$ là hai hàm mô tả đối tượng chưa được xác định rõ ràng

$u \in \mathbb{R}$ là đầu vào, $y \in \mathbb{R}$ là đầu ra,

$x = (x_1, x_2, \dots, x_n)^T$ là vectơ trạng thái (giả sử là có thể đo được)

Để hệ thống ở (3.68) điều khiển được, chúng ta cần có $g(x) \neq 0$.

Giả thiết $g(x) > 0$ và ta biết được chặn trên, chặn dưới của $g(x)$:

$$0 < \underline{g}(x) \leq g(x) \leq \bar{g}(x) < \infty \quad (3.70)$$

Giả thiết tốc độ biến thiên của $g(x)$ bị chặn, tức là tồn tại hàm $D_g(x)$ liên tục sao cho $|\dot{g}(x)| \leq D_g(x)$. Bài toán đặt ra là thiết kế bộ điều khiển hồi tiếp trạng thái để lái ngõ ra y của đối tượng bám theo quỹ đạo mong muốn y_m . Hay nói cách khác là phải giảm tối đa sai số giữa ngõ ra đối tượng và ngõ ra của mô hình tham chiếu.

Do đó ta đặt :

$$\bullet \quad e = y_m - y = y_m - x \quad \text{với } e = (e, \dot{e}, \dots, e^{(n-1)})^T$$

$$\bullet \quad k = (k_n, k_{n-1}, \dots, k_1)^T$$

Các hệ số k_i được chọn như sau:

$$\Delta(s) = s^{(n-1)} + k_1 s^{(n-2)} + \dots + k_{n-2} s + k_{n-1} \quad (3.71)$$

Đa thức trên được gọi là **đa thức Hurwitz**, nghĩa là tất cả nghiệm của phương trình $\Delta(s) = 0$ đều nằm bên trái mặt phẳng phức.

Ta chọn luật điều khiển như sau:

$$u^* = \frac{1}{g(x)} [-f(x) + y_m^{(n)} + k^T e] \quad (3.72)$$

Thay (3.72) vào (3.68) ta có được :

$$e^{(n)} + k_1 e^{(n-1)} + \dots + k_n e = 0$$

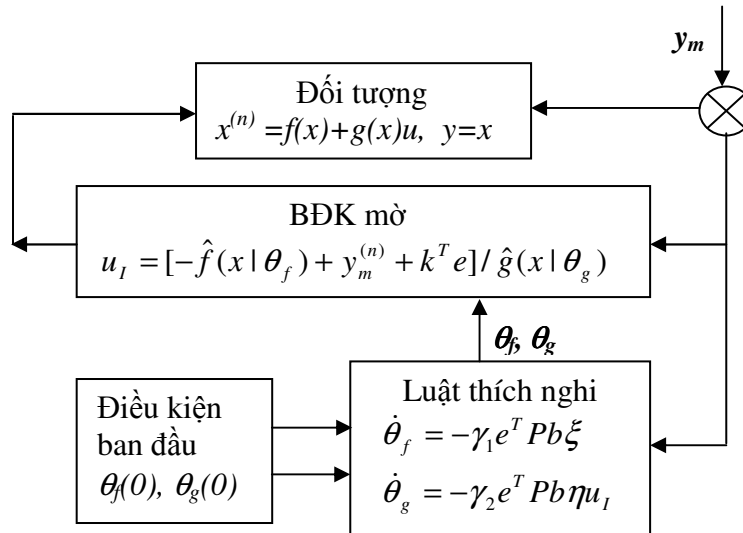
Ta sẽ chọn k sao cho $e(t) \rightarrow 0$ khi $t \rightarrow \infty$. Và khi ấy $y(t) \rightarrow y$, tức là tín hiệu $y(t)$ tiệm cận bám theo tín hiệu chuẩn $y_m(t)$ với sai số bằng 0.

Tuy nhiên do $f(x)$ và $g(x)$ là hàm chưa biết nên bộ điều khiển lý tưởng (3.72) không thể thực hiện được. Bài toán đặt ra là nhận dạng luật điều khiển (3.72). Có hai cách để giải quyết bài toán trên:

- Điều khiển thích nghi mờ gián tiếp: nhận dạng trực tiếp $f(x)$ và $g(x)$ và dựa vào kết quả nhận dạng đó để tính $u^*(t)$.
- Điều khiển thích nghi mờ trực tiếp: nhận dạng trực tiếp luật điều khiển lý tưởng $u^*(t)$.

Ngoài ra, ta cũng có thể sử dụng phối hợp cả 2 cách trên. Đây gọi là phương pháp điều khiển thích nghi mờ hỗn hợp.

3.5.2. Thiết kế BDK mờ thích nghi gián tiếp



Hình 3.22: Hệ thống ĐK mờ thích nghi gián tiếp

• Đặt vấn đề:

Cho một đối tượng phi tuyến có phương trình trạng thái như (3.68)

Mục tiêu của việc điều khiển là thiết kế **một tín hiệu điều khiển hồi tiếp $u = u(x|\theta)$** và **luật thích nghi** dùng để điều chỉnh thông số θ sao cho ngõ ra của hệ thống $y(t) \rightarrow y_m(t)$ của mô hình tham chiếu.

Với lưu ý: $f(x)$ và $g(x)$ là hai hàm mô tả đối tượng chưa được xác định rõ ràng. Vì thế ta sẽ xây dựng một luật mờ IF-THEN để có thể mô tả đáp ứng vào ra của $f(x)$ và $g(x)$.

Giả sử $f(x)$ và $g(x)$ được biểu diễn tả qua luật mờ:

$$\text{If } x_1 = F_1^r \text{ and } \dots \text{ and } x_n = F_n^r \text{ Then } f(x) = C^r \quad (3.73)$$

$$\text{If } x_1 = G_1^s \text{ and } \dots \text{ and } x_n = G_n^s \text{ Then } g(x) = D^s \quad (3.74)$$

Trong đó: F_i^r , C^r , G_i^s và D^s là các tập mờ

$$r = 1, 2, \dots, L_f \quad s = 1, 2, \dots, L_g$$

Kế đến ta sẽ đi thiết kế bộ điều khiển mờ.

• **Thiết kế BDK mờ**

Mặc dù $f(x)$ và $g(x)$ là các hàm chưa biết nhưng nó được mô tả bởi luật mờ IF-THEN ở (3.73) và (3.74). Vì thế, ta có thể thay $f(x)$ và $g(x)$ bởi hệ mờ $\hat{f}(x)$ và $\hat{g}(x)$ tuân theo luật mờ trên.

Tuy nhiên, $\hat{f}(x)$ và $\hat{g}(x)$ chỉ là phép xấp xỉ của $f(x)$ và $g(x)$. Do đó để nâng cao độ chính xác, ta phải đề một số thông số của $\hat{f}(x)$ và $\hat{g}(x)$ tự do. Giả sử ta chọn hai thông số $\theta_f \in R^{M_f}$ và $\theta_g \in R^{M_g}$ là tự do, ta ký hiệu như sau : $\hat{f}(x) = \hat{f}(x | \theta_f)$ và $\hat{g}(x) = \hat{g}(x | \theta_g)$, thay vào (3.80) ta được:

$$u = u_l = \frac{1}{\hat{g}(x | \theta_g)} \left[-\hat{f}(x | \theta_f) + y_m^{(n)} + k^T e \right] \quad (3.75)$$

u_l được gọi là tín hiệu điều khiển tương đương chắc chắn.(do $\hat{f}(x)$ và $\hat{g}(x)$ là các thông số tính toán được).

Để xây dựng BDK (3.75) ta phải xác định $\hat{f}(x | \theta_f)$ và $\hat{g}(x | \theta_g)$, điều này được thực hiện qua 2 bước sau:

Bước 1: Với mỗi biến x_i ($i=1,2,\dots,n$), định nghĩa p_i tập mờ $A_i^{l_i}$ ($l_i=1,\dots,p_i$) và q_i tập mờ $B_i^{l_i}$ ($l_i=1,\dots,q_i$).

Trong đó: F_i^r ($r = 1, 2, \dots, L_f$) và G_i^s và D^s ($s = 1, 2, \dots, L_g$) lần lượt là các trường hợp đặc biệt của các tập mờ $A_i^{l_i}$, $B_i^{l_i}$.

Bước 2:

Xác định hệ mờ $\hat{f}(x | \theta_f)$ từ $\prod_{i=1}^n p_i$; luật mờ có dạng:

$$\text{If } x_1 = A_1^{l_1} \text{ and } \dots \text{ and } x_n = A_n^{l_n}, \text{ Then } \hat{f} = E^{l_1 \dots l_n}$$

Xác định hệ mờ $\hat{g}(x | \theta_g)$ từ $\prod_{i=1}^n q_i$; luật mờ có dạng:

$$\text{If } x_1 = B_1^{l_1} \text{ and } \dots \text{ and } x_n = B_n^{l_n}, \text{ Then } \hat{g} = H^{l_1 \dots l_n}$$

Chọn thiết bị hợp thành tích, hàm mờ dạng singleton, giải mờ theo phương pháp trung bình trọng số, ta được:

$$\hat{f}(x | \theta_f) = \frac{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} \bar{y}_f^{l_1 \dots l_n} (\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i))}{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} (\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i))} \quad (3.76)$$

$$\hat{g}(x | \theta_g) = \frac{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} \bar{y}_g^{l_1 \dots l_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)}{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)} \quad (3.77)$$

Cho $\bar{y}_f^{l_1 \dots l_n}$ và $\bar{y}_g^{l_1 \dots l_n}$ là các thông số tự do. Vì thế ta có thể dồn vào θ_f và θ_g , ta viết lại (3.76) và (3.77) như sau:

$$\hat{f}(x | \theta_f) = \theta_f^T \xi(x) \quad (3.78)$$

$$\hat{g}(x | \theta_g) = \theta_g^T \eta(x) \quad (3.79)$$

Trong đó $\xi(x)$ là vectơ $\prod_{i=1}^n p_i$ chiều và $\eta(x)$ là vectơ $\prod_{i=1}^n q_i$ chiều, với thành phần $l_1 \dots l_n$ được cho bởi:

$$\xi_{l_1 \dots l_n}(x) = \frac{\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)} \quad (3.80)$$

$$\eta_{l_1 \dots l_n}(x) = \frac{\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i)}{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)} \quad (3.81)$$

Ta thấy các thông số θ_f và θ_g được chọn dựa theo luật mờ (3.73) và (3.74). Các thông số này sẽ được đặt các giá trị ban đầu và sau đó thay đổi liên tục. Trong phần kế, chúng ta sẽ thiết kế luật thích nghi cho θ_f và θ_g để đạt được cực tiểu hóa sai số e .

• **Thiết kế luật thích nghi**

Thay (3.75) vào (3.68) và sau một vài biến đổi ta được:

$$e^{(n)} = -k^T e + [\hat{f}(x | \theta_f) - f(x)] + [\hat{g}(x | \theta_g) - g(x)] \mu_I \quad (3.82)$$

Đặt :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ -k_n & -k_{n-1} & \dots & \dots & \dots & \dots & -k_1 \end{bmatrix}, b = \begin{bmatrix} 0 \\ \dots \\ 0 \\ 1 \end{bmatrix} \quad (3.83)$$

Ta viết lại (3.83) dạng vectơ:

$$\dot{e} = Ae + b \{ [\hat{f}(x | \theta_f) - f(x)] + [\hat{g}(x | \theta_g) - g(x)] \mu_I \} \quad (3.84)$$

Định nghĩa các thông số tối ưu như sau:

$$\theta_f^* = \arg \min_{\theta_f \in R^{\prod_{i=1}^n p_i}} \left[\sup_{x \in R^n} |\hat{f}(x | \theta_f) - f(x)| \right] \quad (3.85)$$

$$\theta_g^* = \arg \min_{\theta_g \in R^{\prod_{i=1}^n q_i}} \left[\sup_{x \in R^n} |\hat{g}(x | \theta_g) - g(x)| \right] \quad (3.86)$$

Như vậy, $\hat{f}(x | \theta_f^*)$ và $\hat{g}(x | \theta_g^*)$ sẽ là các xấp xỉ tốt nhất (min-max) của $f(x)$ và $g(x)$ trong tất cả các giá trị cho ở công thức (3.76) và (3.77).

Định nghĩa sai số xấp xỉ cực tiểu:

$$w = [\hat{f}(x | \theta_f^*) - f(x)] + [\hat{g}(x | \theta_g^*) - g(x)] u_I \quad (3.87)$$

Sử dụng w , ta viết lại (3.84) như sau:

$$\dot{e} = Ae + b \left\{ [\hat{f}(x | \theta_f) - \hat{f}(x | \theta_f^*)] + [\hat{g}(x | \theta_g) - \hat{g}(x | \theta_g^*)] u_I + w \right\} \quad (3.88)$$

Thay (3.78) và (3.79) vào (3.87) ta được phương trình động học vòng kín diễn tả mối liên hệ giữa sai số e và thông số θ_f và θ_g .

$$\dot{e} = Ae + b \left[(\theta_f - \theta_f^*)^T \xi(x) + (\theta_g - \theta_g^*)^T \eta(x) u_I + w \right] \quad (3.89)$$

Ta cần tìm luật thích nghi để chỉnh định θ_f và θ_g sao cho sai số e , và các sai số $\theta_f - \theta_f^*$, $\theta_g - \theta_g^*$ đạt giá trị nhỏ nhất. Xét phương trình lyapunov:

$$V = \frac{1}{2} e^T P e + \frac{1}{2\gamma_1} (\theta_f - \theta_f^*)^T (\theta_f - \theta_f^*) + \frac{1}{2\gamma_2} (\theta_g - \theta_g^*)^T (\theta_g - \theta_g^*) \quad (3.90)$$

Trong đó: γ_1 và γ_2 là các hằng số dương,

P là ma trận xác định dương thoả phương trình:

$$A^T P + P A = -Q$$

với Q là ma trận ($n \times n$) xác định dương và ma trận A được cho bởi (3.83)

Lấy đạo hàm V dọc theo quỹ đạo hệ thống ta được:

$$\begin{aligned} \dot{V} = & -\frac{1}{2} e^T P e + e^T P b w + \frac{1}{\gamma_1} (\theta_f - \theta_f^*)^T \left[\dot{\theta}_f + \gamma_1 e^T P b \xi(x) \right] \\ & + \frac{1}{\gamma_2} (\theta_g - \theta_g^*)^T \left[\dot{\theta}_g + \gamma_2 e^T P b \eta(x) u_I \right] \end{aligned} \quad (3.91)$$

Việc cực tiểu hoá e , $\theta_f - \theta_f^*$, $\theta_g - \theta_g^*$, tương đương với cực tiểu V , ta sẽ chọn luật thích nghi sao cho $\dot{V} < 0$. Do $-\frac{1}{2}e^T P e < 0$ nên ta chọn hệ mờ sao cho w là rất nhỏ. Tốt nhất ta chọn luật thích nghi để hai số hạng cuối ở công thức (3.91) bằng 0.

$$\dot{\theta}_f = -\gamma_1 e^T P b \xi(x) \quad (3.92)$$

$$\dot{\theta}_g = -\gamma_2 e^T P b \eta(x) u_i \quad (3.93)$$

Phương pháp này gọi là phương pháp tổng hợp Lyapunov.

Hai phương trình (3.92) và (3.93) chính là luật thích nghi cần tìm.

Ví dụ 3.19

Làm lại **ví dụ 1.8** (trang 45) điều khiển con lắc ngược có sử dụng phương pháp mờ thích nghi gián tiếp và so sánh kết quả đạt được. Trong đó:

- Mô hình con lắc ngược như hình 1.15
- Phương trình trạng thái cho bởi (1.36) và (1.37)
- Chọn $k_1 = 2$ và $k_2 = 1$ để $(s^2 + k_1 s + k_2)$ là ổn định
- Chọn $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$
- Giải phương trình Lyapunov ta được $P = \begin{bmatrix} 15 & 5 \\ 5 & 15 \end{bmatrix}$

Ta chọn: $p_1 = p_2 = q_1 = q_2 = 5$ và $A_1^l = A_2^l = B_1^l = B_2^l$ ($l = 1, 2, \dots, 5$) với

$$\mu_{A_1^1}(x_1) = \exp\left[-\left(\frac{x_1 + \pi/6}{\pi/24}\right)^2\right]; \quad \mu_{A_2^1}(x_1) = \exp\left[-\left(\frac{x_1 + \pi/12}{\pi/24}\right)^2\right]$$

$$\mu_{A_3^1}(x_1) = \exp\left[-\left(\frac{x_1}{\pi/24}\right)^2\right]; \quad \mu_{A_4^1}(x_1) = \exp\left[-\left(\frac{x_1 - \pi/12}{\pi/24}\right)^2\right]$$

$$\mu_{A_5^1}(x_1) = \exp\left[-\left(\frac{x_1 - \pi/6}{\pi/24}\right)^2\right]; \quad x_1 \in [-\pi/6, \pi/6]$$

Do tầm của $f(x_1, x_2)$ quá lớn so với $g(x_1, x_2)$ nên ta chọn: $\gamma_1 = 50$; $\gamma_2 = 1$

Nhận xét : Khi không có tín hiệu điều khiển, tức $u = 0$ thì gia tốc của góc $\theta = x_1$ tương đương $f(x_1, x_2)$. Vậy ta có nhận xét:

x_1 càng lớn thì $f(x_1, x_2)$ càng lớn

Bây giờ, ta sẽ đi xây dựng luật mờ cho $f(x_1, x_2)$. Do $(x_1, x_2) = (0, 0)$ là điểm cân bằng của hệ thống nên ta có luật đầu tiên:

$$R_f^1 : \text{Nếu } x_1 = A_1^3 \text{ và } x_2 = A_2^3 \text{ Thì } f(x_1, x_2) \approx 0$$

Từ hình vẽ mô hình con lắc ngược ta thấy gia tốc của x_1 tỷ lệ với trọng lực $mg \sin(x_1)$, ta có thể chọn $f(x_1, x_2) = \alpha \sin(x_1)$. Ta thấy: $f(x_1, x_2)$ đạt max khi

$x_1 = \pi/2$ nên có thể chọn $\alpha = 16$. Ta được luật mờ cho $f(x_1, x_2)$ như sau:

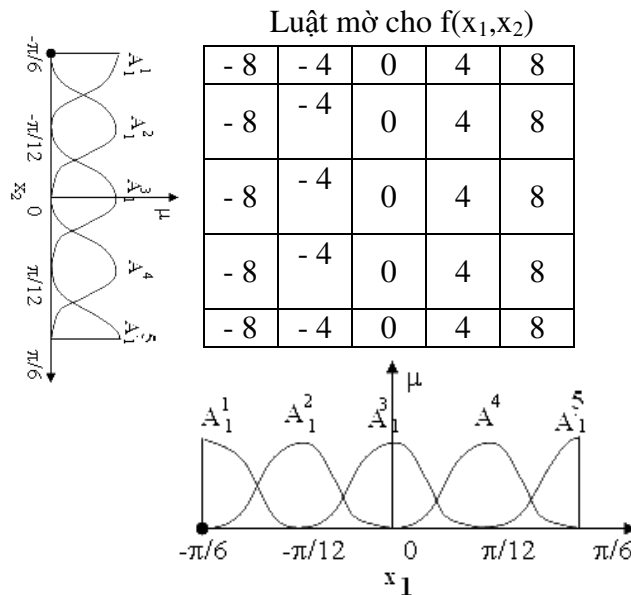
$$R_f^2 : \text{Nếu } x_1 = A_1^1 \text{ và } x_2 = A_2^3 \text{ Thì } f(x_1, x_2) \approx -8$$

$$R_f^3 : \text{Nếu } x_1 = A_1^2 \text{ và } x_2 = A_2^3 \text{ Thì } f(x_1, x_2) \approx -4$$

$$R_f^4 : \text{Nếu } x_1 = A_1^4 \text{ và } x_2 = A_2^3 \text{ Thì } f(x_1, x_2) \approx 4$$

$$R_f^5 : \text{Nếu } x_1 = A_1^5 \text{ và } x_2 = A_2^3 \text{ Thì } f(x_1, x_2) \approx 8$$

Những luật này sẽ được dùng để xác định giá trị đầu $\theta_f(0)$



Tiếp theo ta xác định luật mờ cho hàm $g(x_1, x_2)$, hàm g xác định độ mạnh của luật điều khiển u , ta có nhận xét sau:

x_1 càng nhỏ thì $g(x_1, x_2)$ càng lớn

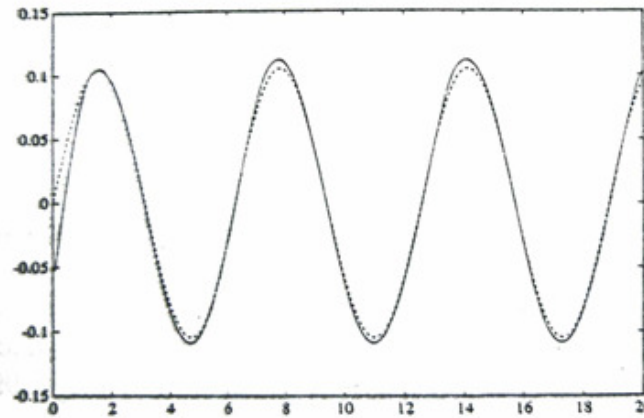
Tương tự như hàm $f(x_1, x_2)$ ta có luật mờ cho hàm $g(x_1, x_2)$:

1.26	1.36	1.46	1.36	1.26
1.26	1.36	1.46	1.36	1.26
1.26	1.36	1.46	1.36	1.26
1.26	1.36	1.46	1.36	1.26
1.26	1.36	1.46	1.36	1.26

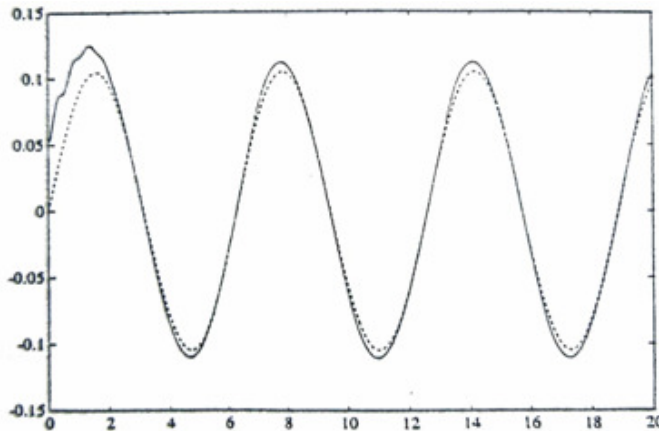
Mô phỏng kết quả ví dụ 3.19

Hình 3.23 và hình 3.24 biểu diễn tín hiệu $x_1(t)$ so với ngõ ra lý tưởng $y_m(t)$ của hệ thống trong 2 điều kiện đầu khác nhau:

Trong đó: $x_1(t)$ được biểu diễn bằng đường đậm
 $y_m(t)$ được biểu diễn bằng đường đứt nét

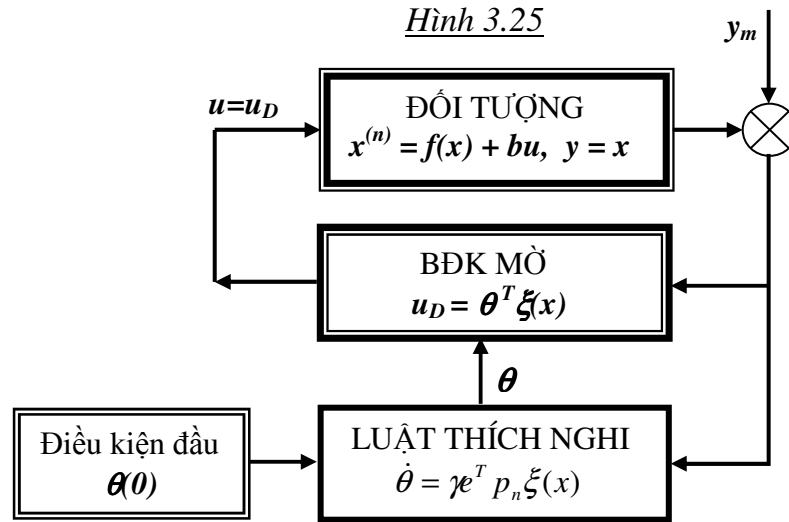


Hình 3.23 Giá trị đầu: $x(0) = (-\pi/60, 0)$



Hình 3.24 Giá trị đầu: $x(0) = (\pi/60, 0)$

3.5.2. Thiết kế BDK mờ thích nghi trực tiếp



• **Đặt vấn đề:**

Cho một đối tượng phi tuyến có phương trình trạng thái như sau:

$$x^{(n)} = f(x, \dot{x}, \dots, x^{(n-1)}) + bu \tag{3.94}$$

$$y = x \tag{3.95}$$

Trong đó: f là hàm đã biết b là hằng số dương chưa biết. Ta cần thiết kế BDK $u = u(x|\theta)$ dựa trên hệ mờ và luật thích nghi để chỉnh định thông số θ . Luật mờ có dạng như sau:

$$\text{IF } x_1 = P_1^r \text{ và } \dots \text{ và } x_n = P_n^r, \text{ THEN } u = Q^r \tag{3.96}$$

Trong đó: P_i^r và Q^r là các tập mờ, $r = 1, 2, \dots, L_u$.

• **Thiết kế BDK mờ**

Hệ mờ $u = u_D(x|\theta)$ được thiết kế theo 2 bước sau:

+ **Bước 1:** Với mỗi biến x_i ($i=1, 2, \dots, n$) ta định nghĩa m_i tập mờ $A_i^{l_i}$ ($l_i=1, 2, \dots, m_i$) . và P_i^r ($r = 1, 2, \dots, L_u$) là một trường hợp đặc biệt.

+ **Bước 2:** Xây dựng hệ mờ $u_D(x|\theta)$ từ $\prod_{i=1}^n m_i$, luật mờ có dạng:

$$\text{IF } x_1 = A_1^{l_1} \text{ and } \dots \text{ and } x_n = A_n^{l_n}, \text{ THEN } u_D = S^{l_1 \dots l_n} \tag{3.97}$$

Trong đó $l_i = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$ và $S^{l_1 \dots l_n}$ bằng với Q^r trong (3.96)

Sử dụng luật hợp thành tích, mờ hoá singleton, giải mờ theo phương pháp trung bình trọng số, ta được:

$$u_D(x|\theta) = \frac{\sum_{l_1=1}^{m_1} \dots \sum_{l_n=1}^{m_n} \bar{y}_u^{l_1 \dots l_n} [\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i)]}{\sum_{l_1=1}^{m_1} \dots \sum_{l_n=1}^{m_n} [\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i)]} \quad (3.98)$$

Chọn $\bar{y}_u^{l_1 \dots l_n}$ như thông số có thể chỉnh định và ta đưa $\bar{y}_u^{l_1 \dots l_n}$ vào thành phần của vectơ thông số θ , từ đó luật điều khiển được xác định:

$$u_D(x|\theta) = \theta^T \xi(x) \quad (3.99)$$

• **Thiết kế luật thích nghi**

Xem u^* như là BDK lý tưởng như (3.72), với $g(x) = b$, ta được:

$$e^{(n)} = -k^T e + b[u^* - u_D(x|\theta)] \quad (3.100)$$

Ma trận A được định nghĩa như (3.83), $b = (0, \dots, 0, b)^T$,

Ta viết lại (3.100) dạng vectơ như sau:

$$\dot{e} = Ae + b[u^* - u_D(x|\theta)] \quad (3.101)$$

Định nghĩa thông số tối ưu θ^* :

$$\theta^* = \arg \min_{\theta \in R^{\chi}} \left[\sup_{x \in R^n} |u_D(x|\theta) - u^*| \right] \quad (3.102)$$

$$\text{với } \chi = \prod_{i=1}^n m_i$$

Sai số xấp xỉ cực tiểu là:

$$w = u_D(x|\theta^*) - u^* \quad (3.103)$$

Từ (3.99) và (3.103) ta viết lại (3.107) như sau:

$$\dot{e} = Ae + b(\theta^* - \theta)^T \xi(x) - bw \quad (3.104)$$

Xét phương trình Lyapunov

$$V = \frac{1}{2} e^T P e + \frac{b}{2\gamma} (\theta^* - \theta)^T (\theta^* - \theta) \quad (3.105)$$

trong đó P là ma trận xác định dương thoả:

$$A^T P + P A = -Q \quad (3.106)$$

Đạo hàm (3.105) và sử dụng các biểu thức (3.104) và (3.106) ta được:

$$\dot{V} = -\frac{1}{2}e^T Qe + e^T P b [(\theta^* - \theta)^T \xi(x) - w] - \frac{b}{\gamma} (\theta^* - \theta)^T \dot{\theta} \quad (3.107)$$

Xem p_n là cột cuối của ma trận P, từ $b = (0, \dots, 0, b)^T$, ta có $e^T P b = e^T p_n b$.

Ta viết lại (3.107) như sau:

$$\dot{V} = -\frac{1}{2}e^T Qe + \frac{b}{\gamma} (\theta^* - \theta)^T [\gamma e^T p_n \xi(x) - \dot{\theta}] - e^T p_n b w \quad (3.108)$$

Từ (3.108) để thoả mãn $\dot{V} < 0$ ta chọn luật thích nghi như sau:

$$\dot{\theta} = \gamma e^T p_n \xi(x) \quad (3.109)$$

Ví dụ 3.20:

Cho hệ thống phi tuyến bậc nhất:

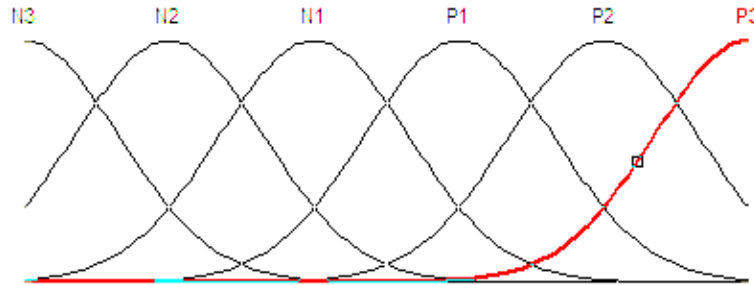
$$\dot{x}(t) = \frac{1 - e^{-x(t)}}{1 + e^{-x(t)}} + u(t) \quad (*)$$

Thiết kế BDK mờ thích nghi trực tiếp để đưa $x(t)$ hội tụ về zero.

Khi $u(t) \equiv 0$ thì: $\dot{x}(t) = \frac{1 - e^{-x(t)}}{1 + e^{-x(t)}} < 0$ khi $x < 0$ và

$\dot{x}(t) = \frac{1 - e^{-x(t)}}{1 + e^{-x(t)}} > 0$ khi $x > 0$ nên hệ (*) là không ổn định.

Chọn $\gamma = 1$ và định nghĩa các tập mờ như sau:



Xây dựng 2 luật mờ như sau:

NẾU $x=N_2$, THÌ $u(x) = PB$

NẾU $x=P_2$, THÌ $u(x) = NB$

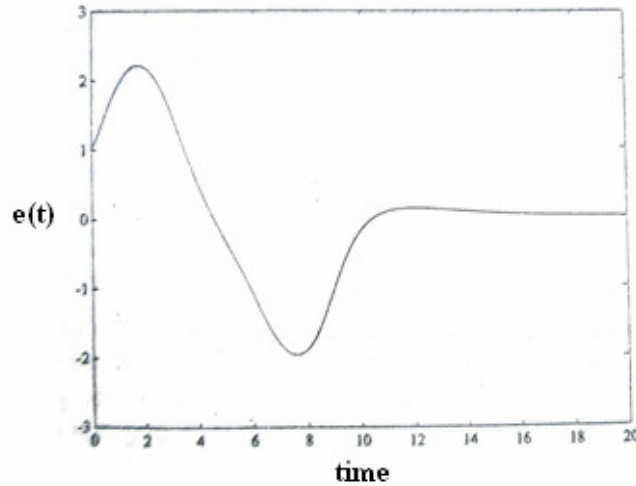
Trong đó: $\mu_{PB}(u) = \exp(-(u-2)^2)$ và $\mu_{NB}(u) = \exp(-(u+2)^2)$.

Nếu x âm thì tín hiệu điều khiển $u(x)$ sẽ càng dương để đảm bảo $\dot{x} > 0$.

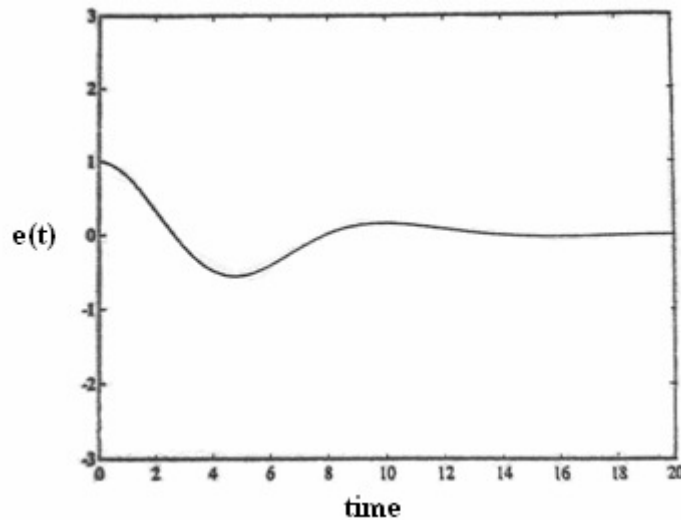
Nếu x dương thì tín hiệu điều khiển $u(x)$ sẽ càng âm để đảm bảo $\dot{x} < 0$.

Mô phỏng kết quả:

Ta tiến hành mô phỏng để thấy được đáp ứng trong hai trường hợp có và không có luật mờ. Giả sử điều kiện ban đầu là $x(0) = 1$.



Hình 3.26 Điều khiển mờ thích nghi trực tiếp khi không có luật mờ



Hình 3.27 Điều khiển mờ thích nghi trực tiếp khi có luật mờ

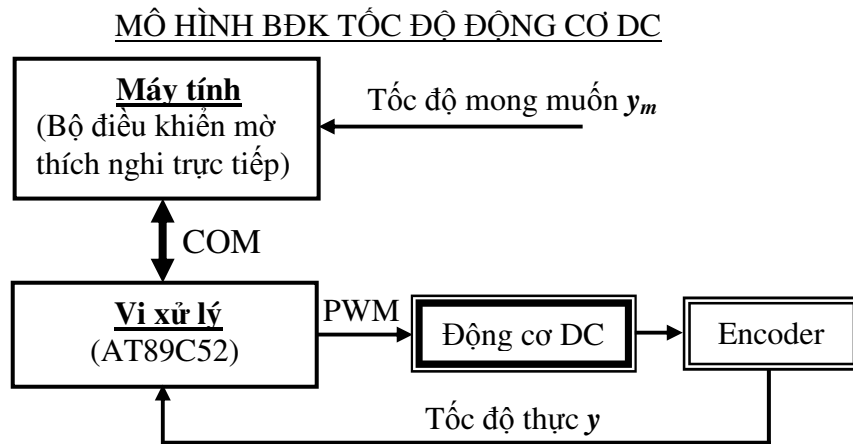
Nhận xét:

Theo kết quả mô phỏng thu được từ hai hình 3.26 và hình 3.27 ta thấy

- Bộ điều khiển mờ thích nghi trực tiếp có thể dùng để điều chỉnh đôi tượng trở về zero mà không cần sử dụng đến luật mờ.
- Nếu bộ điều khiển mờ thích nghi trực tiếp có sử dụng kèm theo luật điều khiển mờ thì sẽ cho tốc độ hội tụ nhanh hơn nhiều.

3.5.4. Một số ứng dụng

Ví dụ 3.21: Xây dựng BDK tốc độ động cơ DC



Hình 3.28

Mô hình gồm có :

1. Động cơ DC 14V, tốc độ Max 2100vòng/phút, làm việc không tải.
2. Cảm biến tốc độ Incremental 200xung/vòng.
3. Vi xử lý AT89C52, tần số xung clock 11.059MHz, chu kỳ máy $T_{VXL} = \frac{12}{11.059.000} (s)$, có nhiệm vụ đo tốc độ động cơ gửi về máy tính điều khiển áp cấp cho động cơ bằng phương pháp PWM.
4. Chu kỳ PWM = $1024 \times T_{VXL} (\approx 1.1ms)$, chu kỳ lấy mẫu $46.080 \times T_{VXL} (\approx 50ms)$, tốc độ port nối tiếp 19200Kbps.

5. Hệ số thích nghi γ thay đổi tùy thuộc vào sai lệch $\varepsilon = \frac{y_m - y}{y_m}$. Khi sai lệch $\varepsilon \geq 1\%$ thì $\gamma = \gamma_0$, khi $\varepsilon < 1\%$ thì $\gamma = \gamma_0/10$, với γ_0 đã chọn trước.

Xây dựng BDK mờ thích nghi trực tiếp

1. Xác định biến ngôn ngữ

•Hai ngõ vào:

Tốc độ x_1 (vòng/phút), có tầm giá trị từ 0...2000 vòng/phút, được chuẩn hoá về [0...1].

Hàm thuộc $\mu_1^i(x_1)$ dạng Gaussian, với $i=1...m_1$, m_1 là số lượng tập mờ

Gia tốc x_2 (vòng/phút/giây), có tầm giá trị từ - 4500...4500 (v/p/g), được chuẩn hoá về [-1...1].

Hàm thuộc $\mu_2^j(x_2)$ dạng Gaussian, với $j=1...m_2$, m_2 là số lượng tập mờ

•Một ngõ ra:

Độ rộng xung PWM (%), ký hiệu là u , có tầm giá trị 0...100%.

Hàm thuộc dạng Singleton $\theta_{i,j}$, với $i=1...m_1$, $j=1...m_2$.

Bảng luật hợp thành:

		BIẾN NGÔN NGỮ GIA TỐC				
		$\mu_2^1(x_2)$	$\mu_2^2(x_2)$	$\mu_2^3(x_2)$...	$\mu_2^{m_2}(x_2)$
BIẾN NGÔN NGỮ TỐC ĐỘ	$\mu_1^1(x_1)$	$\theta_{1,1}$	$\theta_{1,2}$	$\theta_{1,3}$...	θ_{1,m_2}
	$\mu_1^2(x_1)$	$\theta_{2,1}$	$\theta_{2,2}$	$\theta_{2,3}$...	θ_{2,m_2}
	$\mu_1^3(x_1)$	$\theta_{3,1}$	$\theta_{3,2}$	$\theta_{3,3}$...	θ_{3,m_2}

	$\mu_1^{m_1}(x_1)$	$\theta_{m_1,j}$	$\theta_{m_1,2}$	$\theta_{m_1,3}$...	$\theta_{m_1,j}$

2. Luật hợp thành:

Xét luật hợp thành thứ (i,j), với $i = 1...m_1$, $j = 1...m_2$

IF $x_1 \in \mu_1^i(x_1)$ AND $x_2 \in \mu_2^j(x_2)$ THEN $u = \theta_{i,j}$

3. Giải mờ:

Chọn thiết bị hợp thành Max – Product , phương pháp giải mờ độ cao.
 Giá trị rõ đầu ra PWM điều khiển động cơ:

$$u = \frac{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \theta_{i,j} \cdot \mu_1^i(x_1) \cdot \mu_2^j(x_2)}{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \mu_1^i(x_1) \cdot \mu_2^j(x_2)} \quad (3.110)$$

4. Luật cập nhật thông số:

$$\begin{cases} \Delta \theta_{i,j} = \gamma \cdot E^T \cdot p_2 \cdot \xi_{i,j}(x_1, x_2) \\ \theta_{i,j}(k+1) = \theta_{i,j}(k) + \Delta \theta_{i,j} \end{cases} \quad (3.111)$$

Trong đó:

$\theta_{i,j}$: Thông số cần cập nhật ở luật hợp thành thứ (i,j).

$E = (e, \dot{e})$: Vectơ sai số, với sai số $e = y_m - y$, với y_m là vận tốc đặt.

p_2 : là cột thứ 2 của ma trận P có được từ phương trình Ricatti (3.106).

Với $A = \begin{bmatrix} 0 & 1 \\ k_1 & k_2 \end{bmatrix}$, k_1, k_2 được chọn sao cho phương trình $s^2 + k_1 s + k_2 = 0$ có nghiệm nằm bên trái mặt phẳng phức.

Các thí nghiệm trong bài được chọn với $s = -0.1 \pm j20$.

$\gamma > 0$ là hệ số cập nhật

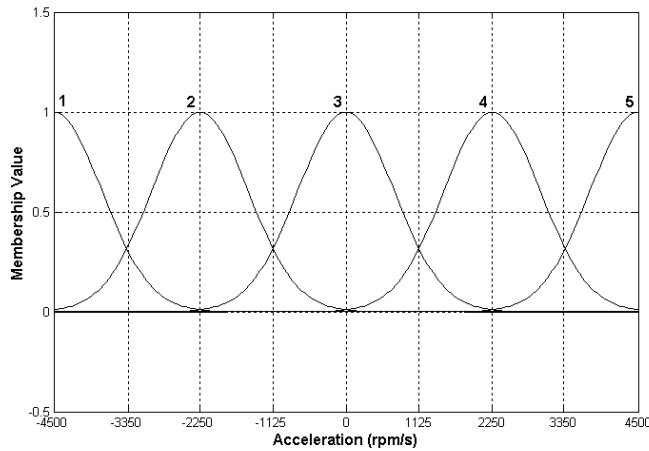
$$\xi_{i,j}(x_1, x_2) = \frac{\mu_1^i(x_1) \cdot \mu_2^j(x_2)}{\sum_{k=1}^{m_1} \sum_{l=1}^{m_2} \mu_1^k(x_1) \cdot \mu_2^l(x_2)}$$

thành thứ (i,j).

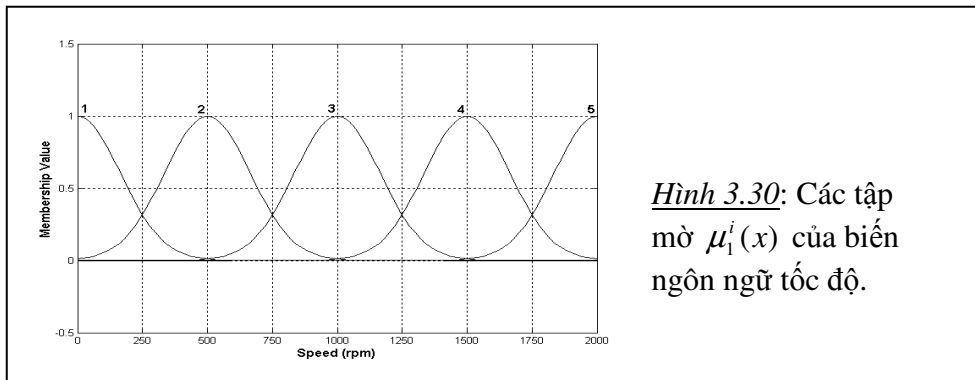
Kết quả mô phỏng và nhận xét:

Ghi chú : Trong các đồ thị bên dưới, đường liền nét là tốc độ mong muốn y_m đường còn lại là tốc độ thực.

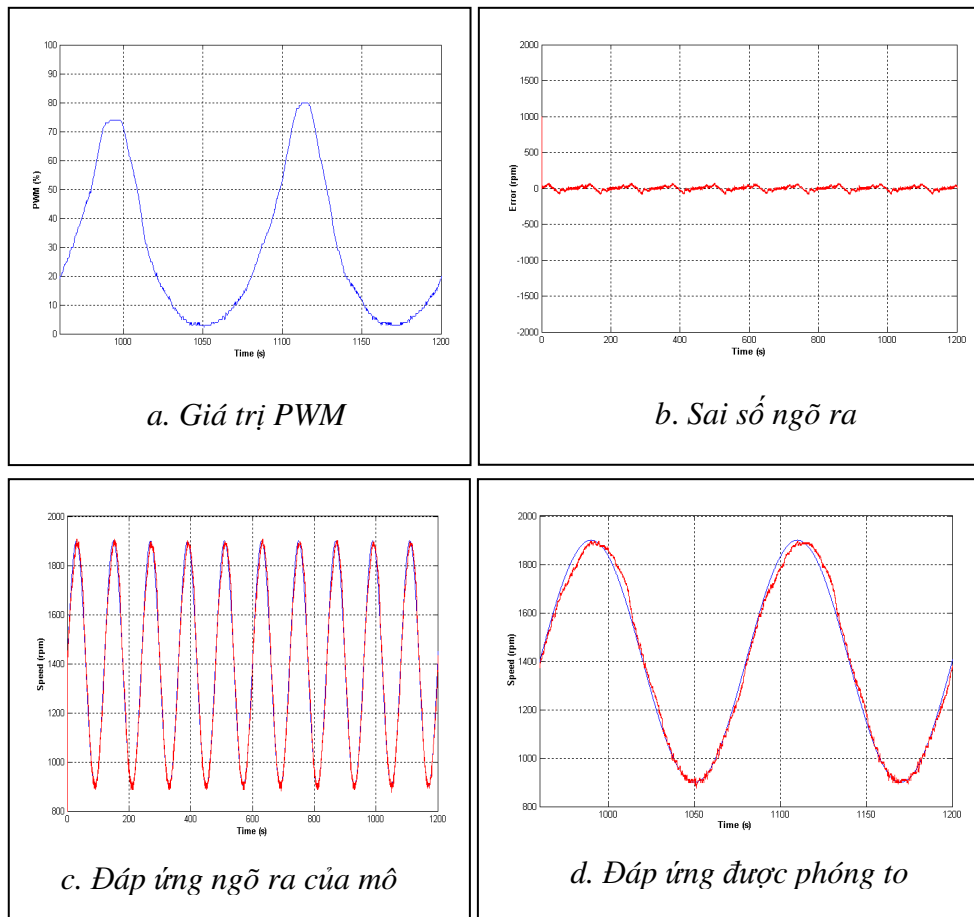
Trường hợp 1: Chọn $\gamma_0 = 0.5$; $y_m = 1400 + 500 \sin(\frac{2\pi}{120}t)$, $\theta_{i,j} = 20$, với $i = 1 \dots m_1, j = 1 \dots m_2$. Các tập mờ cho bởi Hình 3.29 và Hình 3.30



Hình 3.29: Các tập mờ $\mu_2^j(x)$ của biến ngôn ngữ gia tốc.



Hình 3.30: Các tập mờ $\mu_1^i(x)$ của biến ngôn ngữ tốc độ.

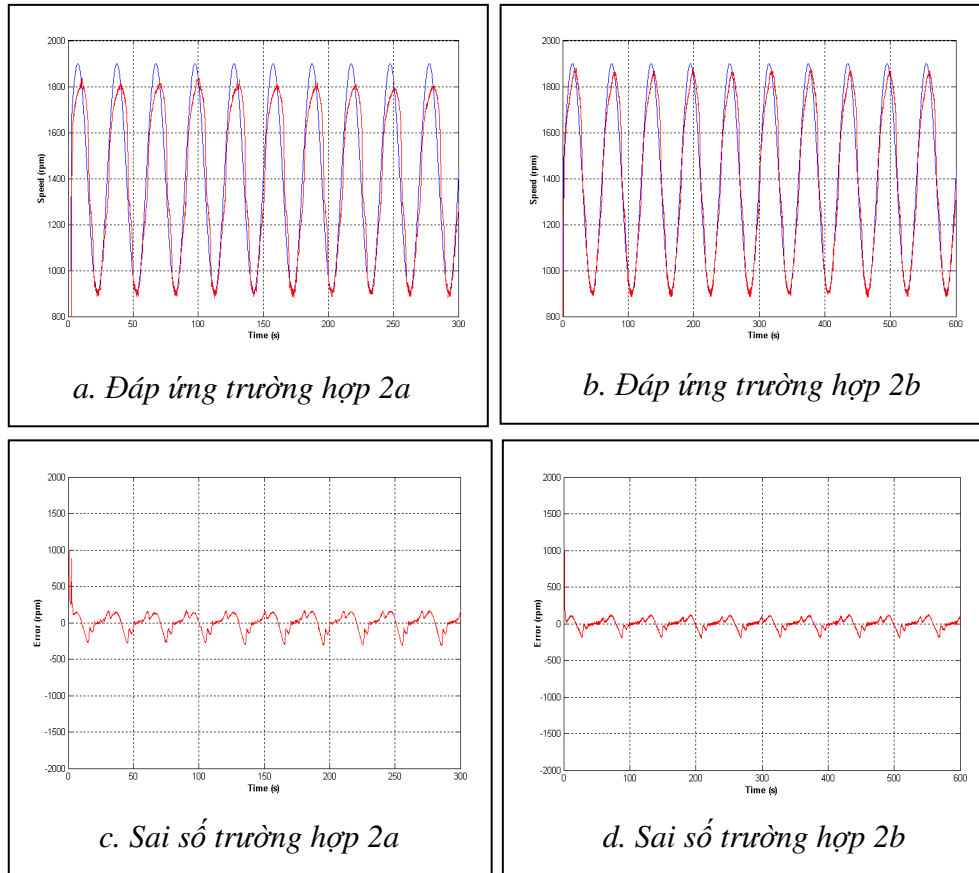


Hình 3.31: Kết quả điều khiển của Trường hợp 1.

Nhận xét: Từ các đồ thị ở Hình 3.31 ta thấy rằng:

Ở tốc độ thấp, giá trị PWM thay đổi ít nhưng tốc độ thay đổi nhiều; ở tốc độ cao giá trị PWM thay đổi nhiều nhưng tốc độ thay đổi ít.

Bộ điều khiển mờ ban đầu được thiết kế mà không dựa trên nhiều thông tin về đối tượng, nhưng chất lượng điều khiển là khá tốt dù đối tượng là phi tuyến.



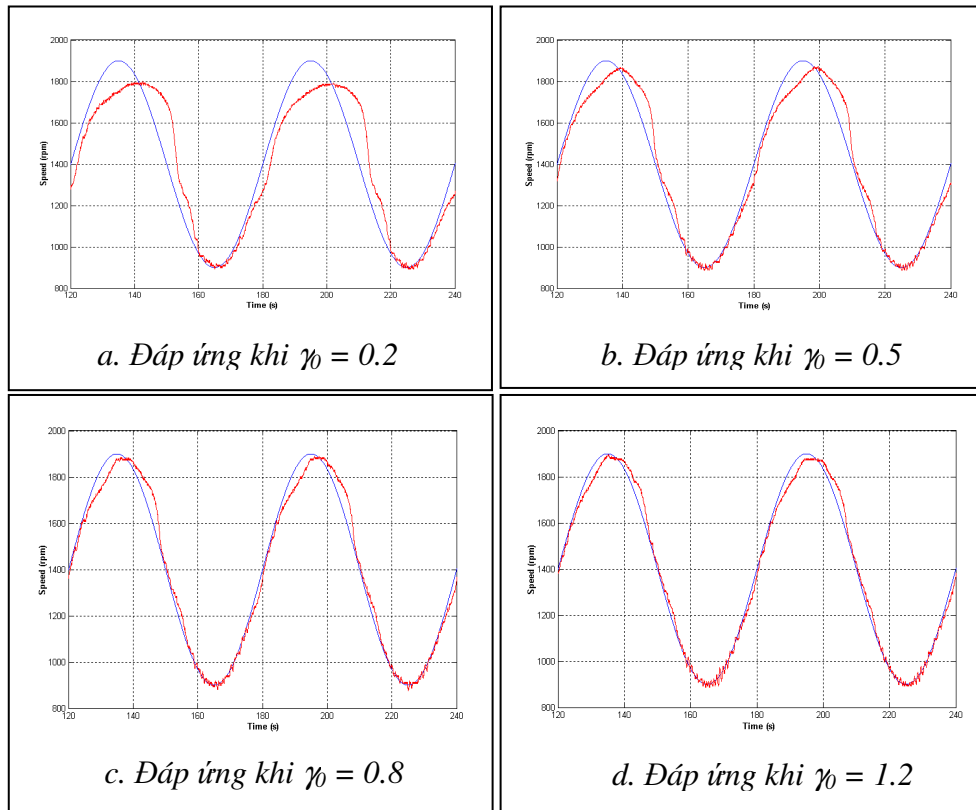
Hình 3.32: Kết quả điều khiển của Trường hợp 2

Trường hợp 2: $\gamma_0 = 0.5$; $y_m = 1400 + 500 \sin(\frac{2\pi}{30}t)$ (trường hợp 2a) và

$y_m = 1400 + 500 \sin(\frac{2\pi}{60}t)$ (trường hợp 2b); $\theta_{i,j} = 20$, với $i = 1..5$, $j = 1..5$.

Các tập mờ vẫn như trường hợp 1. (Xem kết quả ở Hình 3.32)

Nhận xét: Với cùng hệ số cập nhật và các giá trị ban đầu $\theta_{i,j}$, khi tốc độ mong muốn y_m biến thiên nhanh hơn thì tốc độ thức y không bám theo kịp dẫn đến sai số lớn. Do luật cập nhật phụ thuộc vào y_m nên ta cần hiệu chỉnh lại thông số γ_0 cho phù hợp.



Hình 3.33: Kết quả điều khiển Trường hợp 3

Trường hợp 3: $\theta_{i,j} = 20$; với $i = 1..5$, $j = 1..5$; $y_m = 1400 + 500 \sin\left(\frac{2\pi}{60}t\right)$, các tập mờ như Trường hợp 1, γ_0 lần lượt là 0.2, 0.5, 0.8, 1.2.

Nhận xét:

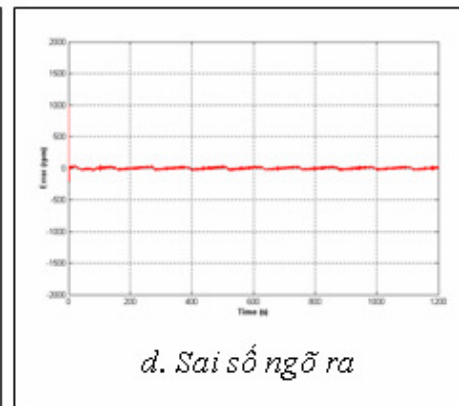
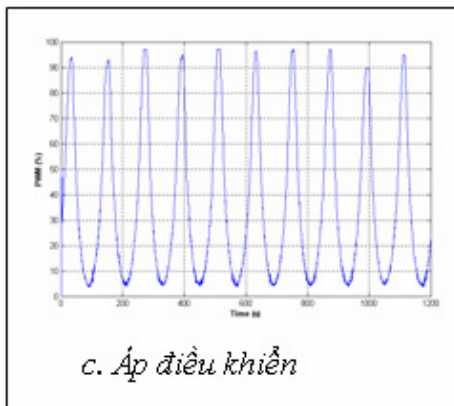
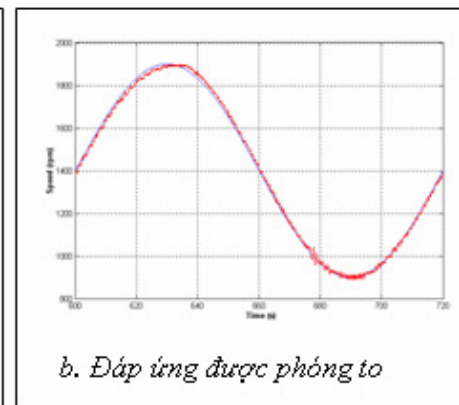
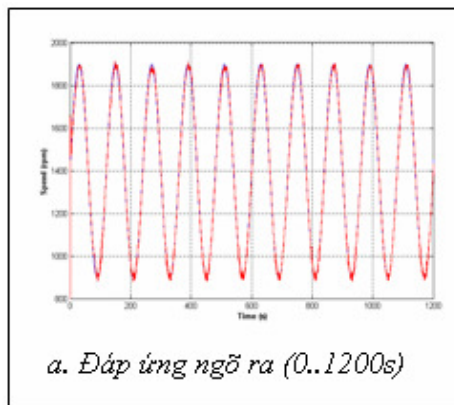
Việc tăng γ_0 sẽ làm cho luật cập nhật nhạy hơn với sai số, do vậy đáp ứng hệ thống sẽ tốt hơn.

Tuy vậy ở tốc độ thấp, khi γ_0 tăng sẽ làm cho tốc độ động cơ bị dao động lớn hơn. Sự dao động tỷ lệ thuận với việc tăng γ_0 .

Bằng kinh nghiệm qua các trường hợp đã xét ta thấy rằng đáp ứng tốc độ phụ thuộc vào nhiều yếu tố như: số lượng tập mờ, hệ số γ_0 , $\theta_{i,j}$, tốc độ biến thiên của tốc độ mong muốn... Từ đó ta đưa ra việc lựa chọn các thông số cho phù hợp để tối ưu đáp ứng của hệ thống.

Trường hợp 4: $\gamma_0 = 2.5$; $y_m = 1400 + 500 \sin(\frac{2\pi}{60}t)$; $\theta_{i,j}$ được chọn như bảng bên dưới, sử dụng 7 tập mờ cho biến tốc độ và 5 tập mờ cho biến gia tốc.

		BIẾN NGÔN NGỮ GIA TỐC				
		$\mu_1^1(x_2)$	$\mu_2^2(x_2)$	$\mu_3^3(x_2)$	$\mu_4^4(x_1)$	$\mu_5^5(x_2)$
BIẾN NGÔN NGỮ TỐC ĐỘ	$\mu_1^1(x_1)$	10	10	10	10	10
	$\mu_1^2(x_1)$	10	10	10	10	10
	$\mu_1^3(x_1)$	20	20	20	20	20
	$\mu_1^4(x_1)$	30	30	30	30	30
	$\mu_1^5(x_1)$	45	45	45	45	45
	$\mu_1^6(x_1)$	60	60	60	60	60
	$\mu_1^7(x_1)$	80	80	80	80	80



Hình 3.34: Kết quả điều khiển trường hợp 4

Kết luận chung

Đối tượng động cơ DC được điều khiển bằng phương pháp PWM là đối tượng phi tuyến. Một BDK mờ thích nghi được thiết kế hợp lý sẽ điều khiển tốc độ của động cơ bám theo nhiều dạng tốc độ mong muốn khác nhau.

Những kinh nghiệm, thông tin đã biết về đối tượng sẽ rất hữu ích trong việc tìm ra BDK thích nghi tối ưu.

Các thông số quyết định chất lượng hệ thống là : hệ số γ_0 , giá trị ban đầu $\theta_{i,j}$, tín hiệu mong muốn $y_m \dots$ Với mỗi thông số có một tác dụng riêng, việc tìm ra bộ thông số tối ưu cần dựa vào kinh nghiệm và kiến thức về hệ thống điều khiển.

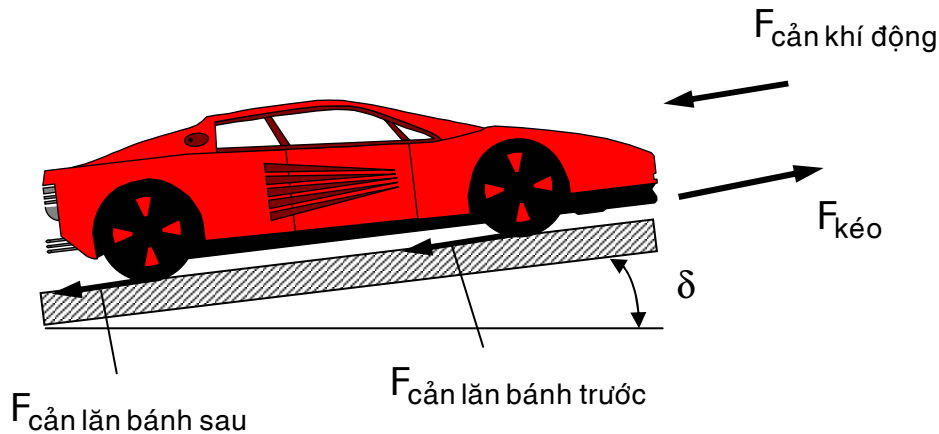
Ví dụ 3.22: Mô hình Hệ thống ga tự động trên ô tô

1. Động lực học của ô tô trên đường:

Ô tô vận hành trên đường nhờ moment sinh ra từ động cơ, thông qua hệ thống truyền động, chuyển thành lực kéo tiếp tuyến tại các bánh xe chủ động đẩy ô tô dịch chuyển lên phía trước. Lực kéo tiếp tuyến này luôn cân bằng với các lực cản tác động vào ô tô theo định luật I Newton:

$$F_{kéo} = F_{cản lăn} + F_{cản khí động} + F_{cản leo dốc} + F_{cản quán tính}$$

Tổng các lực cản đối với ô tô không phụ thuộc tuyến tính vào vận tốc của ô tô và các thành phần lực cản này có những hệ số phụ thuộc vào điều kiện làm việc của ô tô như loại đường, độ mấp mô, độ nghiêng của mặt đường, loại lốp xe, nhiệt độ môi trường, gió, tải trọng của xe, tình trạng của động cơ, của hệ thống truyền động, độ mòn của lốp... Các điều kiện làm việc này không cố định mà thay đổi mỗi khi ô tô vận hành và trong lúc ô tô vận hành.



Hình 3.35 Động lực học của ô tô trên đường

2. Phương trình trạng thái:

Đối tượng vận hành trên đường là một đối tượng phi tuyến chỉ bao gồm một tín hiệu điều khiển vào là độ mở cánh bướm ga của động cơ (hay vị trí bàn đạp ga trên ô tô), và một đầu ra là vận tốc ô tô. Các trạng thái của đối tượng là vận tốc và gia tốc. Dựa theo phương trình trạng thái (3.94) và (3.95), đặc tính của đối tượng này có thể được biểu diễn bởi hệ phương trình vi phân tuyến tính bậc 2 ở dạng chính tắc:

$$\begin{cases} x_1 = v_{xe} \\ x_2 = \dot{x}_1 \\ \dot{x}_2 = f(x_1, x_2) + b.\alpha, (b > 0) \\ y = x_1 \end{cases}$$

với $f(x_1, x_2)$ là hàm phi tuyến chưa biết và $b > 0$ là giá trị chưa biết, α là độ mở bướm ga của động cơ. Cả $f(x_1, x_2)$ và b phụ thuộc vào điều kiện làm việc, chế độ vận hành, tình trạng của động cơ, hệ thống truyền động của ô tô.

3. Luật điều khiển đối tượng:

Luật điều khiển đối tượng dựa trên tuyến tính hoá hồi tiếp để ngõ ra y của đối tượng bám sát ngõ ra mong muốn y_m có dạng như (3.72):

$$u^* = \frac{1}{b} [-f(x) + y_m^{(n)} + K^T E]$$

với: $e = y_m - y = y_m - x = (e, \dot{e}, \dots, e^{(n-1)})^T$ và $k = (k_n, k_{n-1}, \dots, k_1)^T$

$s^n + k_1 s^{n-1} + \dots + k_n = 0$ có tất cả các nghiệm nằm bên trái trục ảo của mặt phẳng phức.

Với luật điều khiển u^* , các thành phần phi tuyến của đối tượng bị triệt tiêu. Thành phần $K^T E$ được đưa vào để đảm bảo sai số ngõ ra vẫn hội tụ về 0 trong trường hợp trạng thái ban đầu của đối tượng không làm cho ngõ ra y bám ngay ngõ ra mong muốn y_m .

Do thành phần $f(x)$ và b chưa xác định nên luật điều khiển u^ cho đối tượng được xem là chưa biết. Trong bộ điều khiển mờ thích nghi trực tiếp, một hệ thống mờ được sử dụng để tìm ra hay xấp xỉ luật điều khiển mong muốn u^* chưa biết này.*

4. Thiết kế luật thích nghi (luật cập nhật, chỉnh định thông số)

Với bộ điều khiển mờ thích nghi trực tiếp, luật chỉnh định thông số để vector thông số θ hội tụ về vector thông số lí tưởng θ^* (nghĩa là $u_D(X, \theta)$ hội tụ về $u_D(X, \theta^*)$ là xấp xỉ của u^* với sai số xấp xỉ nhỏ nhất ϵ), được xác định theo tiêu chuẩn ổn định Lyapunov.

Xét biểu thức Lyapunov (3.104) cho hệ thống mờ thích nghi trực tiếp $u_D(X, \theta)$ dùng cho đối tượng được mô tả trong phương trình sau:

$$V(E) = \frac{1}{2} E^T P E + \frac{b}{2\gamma} (\theta^* - \theta)^T (\theta^* - \theta)$$

với:

- $\gamma > 0$ là một hằng số, được gọi là hệ số cập nhật hay hằng số hội tụ.
- $P \in R^{n \times n}$ là ma trận thực, xác định dương thoả mãn phương trình:

$$\Lambda^T P + P \Lambda = -Q$$

Trong đó, $Q \in R^{n \times n}$ là ma trận thực, dương được chọn trước.

Theo tiêu chuẩn ổn định Lyapunov, với $V(E)$ xác định dương và $\dot{V}(E)$ xác định âm thì sai số E sẽ tiến về 0, hay giá trị ngõ ra y sẽ bám theo giá trị ngõ ra y_m mong muốn, khi đó xác định được luật thích nghi (3.109):

$$\dot{\theta} = \gamma E^T p_n \xi(X)$$

Mô hình động lực học ô tô trên đường

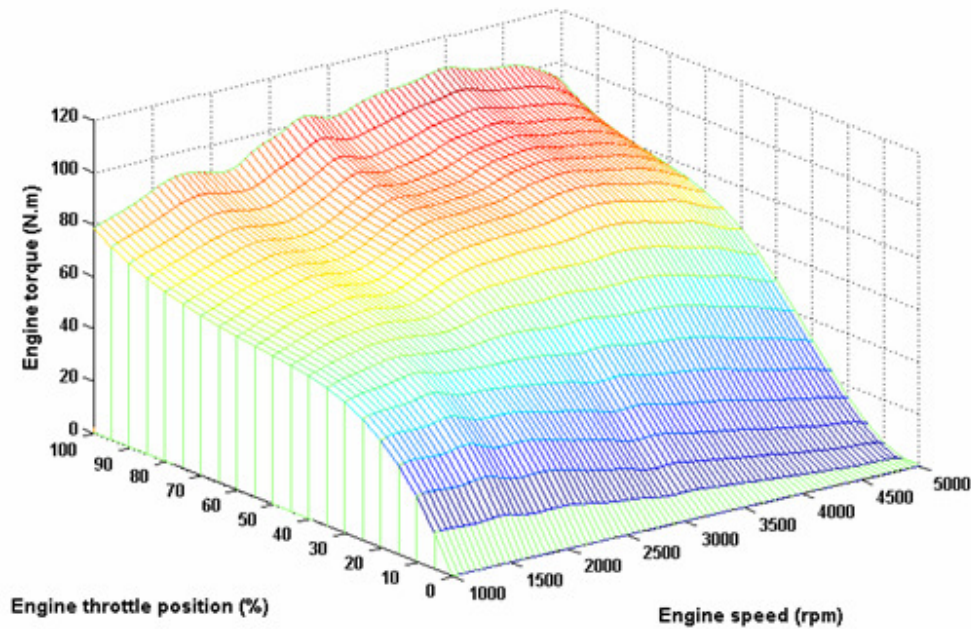
Đặc tính động học của ô tô trên đường được mô tả qua phương trình sau:

$$\left(\frac{m_{xe}}{g} + J_e \right) a_{xe} = \frac{i_e}{r_w} M_e - \sum F_{load}(V_{xe})$$

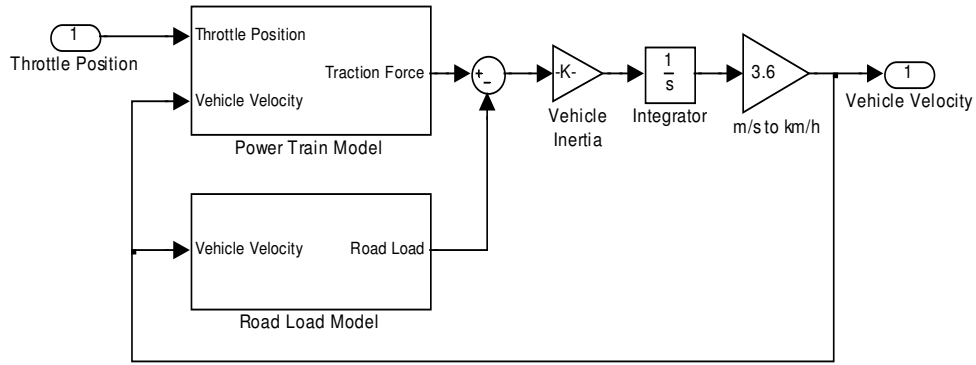
Trong đó:

- $m_{xe} = G_0 + G_t$ là trọng lượng toàn bộ của ô tô, [N].
- a_{xe} là gia tốc ô tô, [m/s²].
- V_{xe} là vận tốc ô tô, [m/s].
- n_e là tốc độ động cơ, [rpm] (vòng/phút).

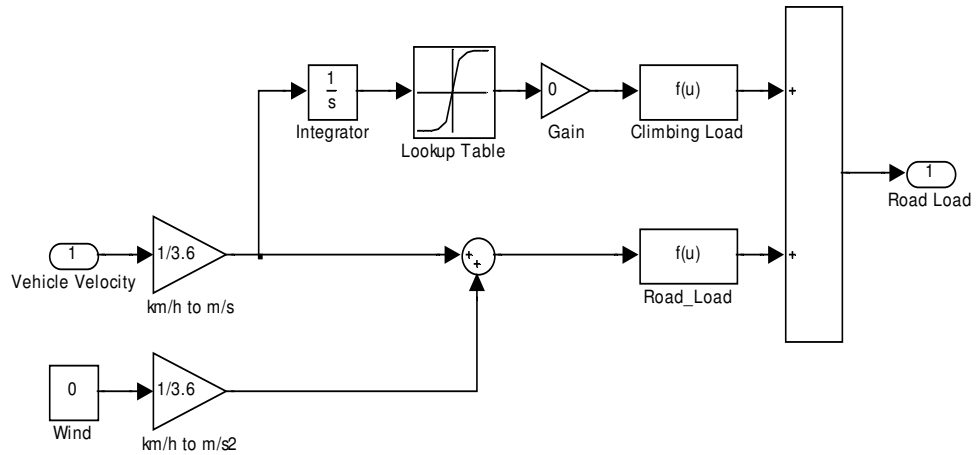
- α là vị trí (độ mở) cánh bướm ga, [%].
- $i_e = i_4 i_0$ là tỉ số truyền lực của hệ thống truyền động.
- M_e là moment xoắn có ích do động cơ sinh ra, [N.m].
- $\sum F_{load} = F_{roadload} + F_{loss} + F_{slope}$ là tổng các lực cản đối với hệ thống truyền động của ô tô, [N].
- $F_{roadload} = F_2 V_v^2 + F_1 V_v + F_0$ là tổng các lực do chuyển động trên đường, bao gồm lực cản lăn giữa lốp và mặt đường, lực cản khí động, [N].
- $F_{loss} = F_{l_3} V_v^3 + F_{l_2} V_v^2 + F_{l_1} V_v + F_{l_0}$ là lực cản do tổn thất cơ giới trong hệ thống truyền động, [N].
- $F_{slope} = m_{xe} \cdot g \cdot \sin(Grade)$ là lực cản leo dốc, [N].



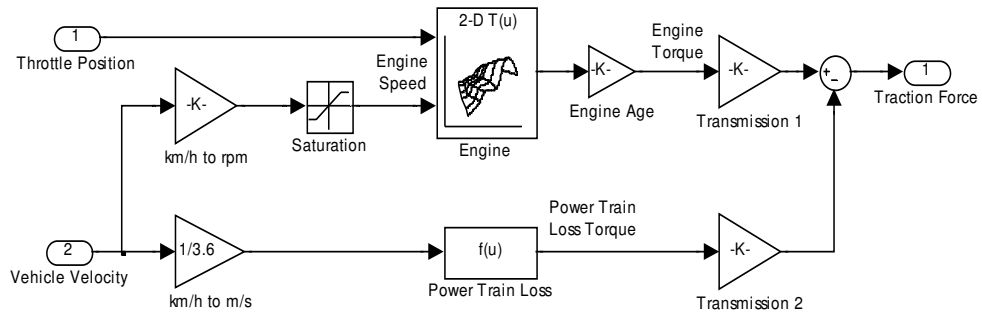
Đặc tính moment theo tốc độ và độ mở bướm ga của động cơ



Hình 3.36 Mô hình động lực học ô tô trên đường



Hình 3.37 Mô hình lực cản trên đường



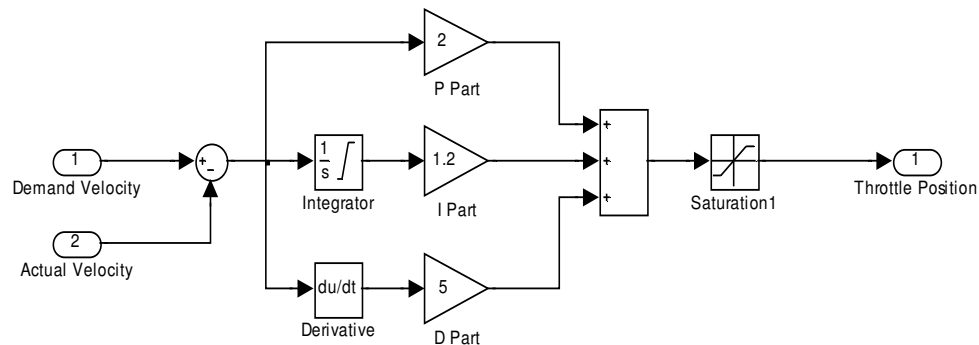
Hình 3.38 Mô hình hệ thống truyền lực của ô tô

3. Hai bộ điều khiển hệ thống ga tự động trên ô tô

Vận tốc ban đầu của ô tô là 40km/h, ô tô sẽ được điều khiển đạt vận tốc ổn định 60km/h sau 20s bằng bộ điều khiển PID và bộ điều khiển mờ thích nghi trực tiếp (DAF – Direct Adaptive Fuzzy).

Cả 2 bộ điều khiển được xây dựng với giả thiết đã có bộ điều khiển độ mở cánh bướm ga lí tưởng, điều khiển chính xác độ mở cánh bướm ga với thời gian quá độ rất bé.

A. Bộ điều khiển PID: $K_P = 2$; $K_I = 1.2$; $K_D = 5$



Hình 3.39 Bộ điều khiển PID

B. Bộ điều khiển mờ thích nghi trực tiếp

Bộ điều khiển mờ thích nghi trực tiếp có những đặc điểm sau:

1. Ngõ vào: 2 ngõ vào

a. Actual Velocity

- Tầm giá trị: 0...120km/h
- 5 tập mờ như hình 2.18

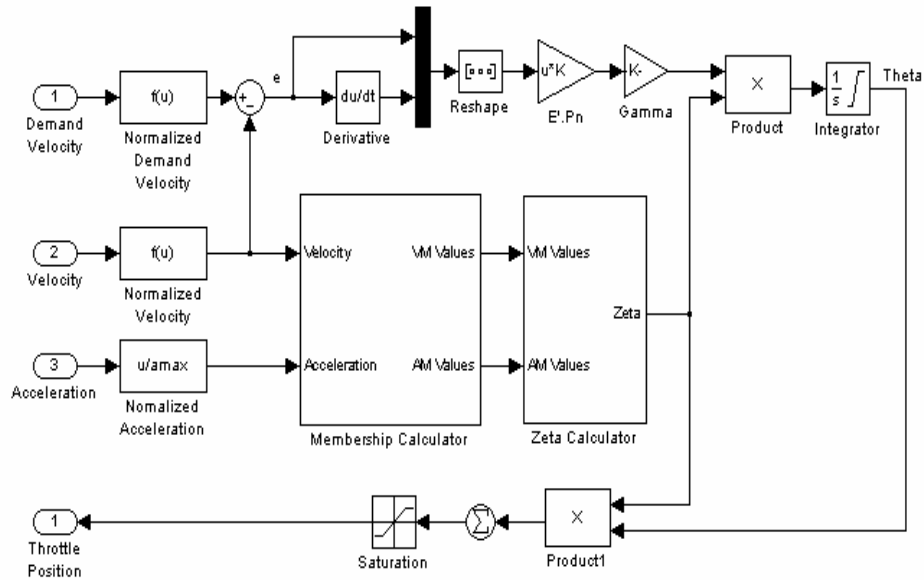
b. Acceleration

- Tầm giá trị: -4...4m/s².
- 5 tập mờ như hình 2.19

2. Ngõ ra: 1 ngõ ra

- Tên biến ngôn ngữ: Trottle Position

- Tầm giá trị: 0...100%
- 25 tập mờ dạng singleton, là các thông số được điều chỉnh của hệ thích nghi.



Hình 3.40 Bộ điều khiển mờ thích nghi trực tiếp (DAF)

3. Bảng luật hợp thành với giá trị ban đầu của các thông số

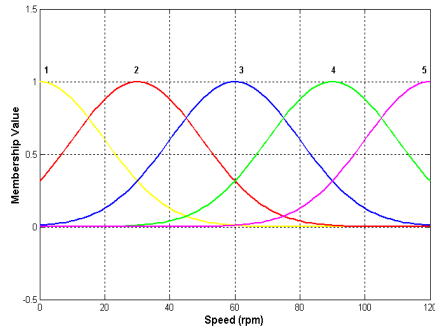
		<i>Biến ngôn ngữ ACCELERATION</i>				
		$\mu_2^1(x_2)$	$\mu_2^2(x_2)$	$\mu_2^3(x_2)$	$\mu_2^4(x_2)$	$\mu_2^5(x_2)$
<i>Biến ngôn ngữ ACTUAL VELOCITY</i>	$\mu_1^1(x_1)$	1	1	1	1	1
	$\mu_1^2(x_1)$	15	15	15	15	15
	$\mu_1^3(x_1)$	30	30	30	30	30
	$\mu_1^4(x_1)$	48	48	48	48	48
	$\mu_1^5(x_1)$	100	100	100	100	100

Bảng 3.1 Bảng luật hợp thành của bộ điều khiển mờ thích nghi trực tiếp

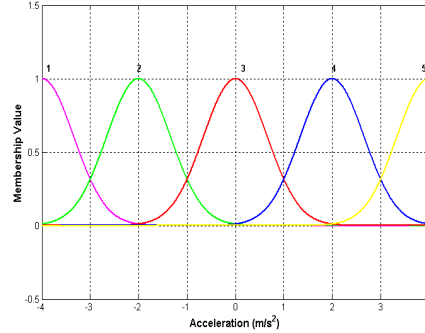
4. Chọn ma trận $\Lambda = \begin{bmatrix} 0 & 1 \\ -0.01 & -0.1 \end{bmatrix}$

5. Chọn ma trận $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$

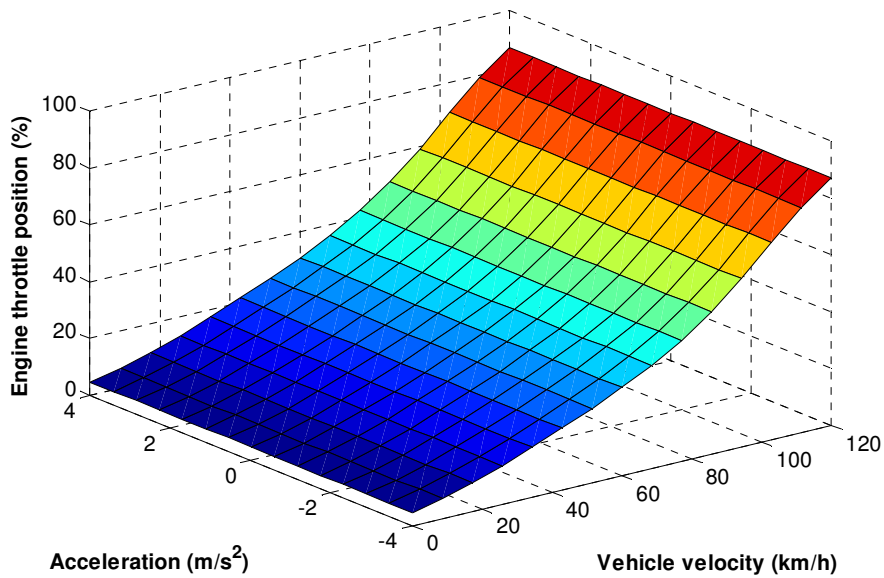
6. Chọn hệ số $\gamma = 1$.



Hình 3.41 5 tập mờ $\mu_1^i(x)$, $i = 1..5$, của ngõ vào Actual Velocity



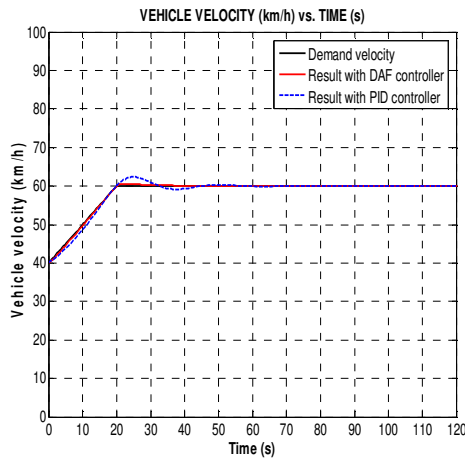
Hình 3.42 5 tập mờ $\mu_2^i(x)$, $i = 1..5$, của ngõ vào Acceleration



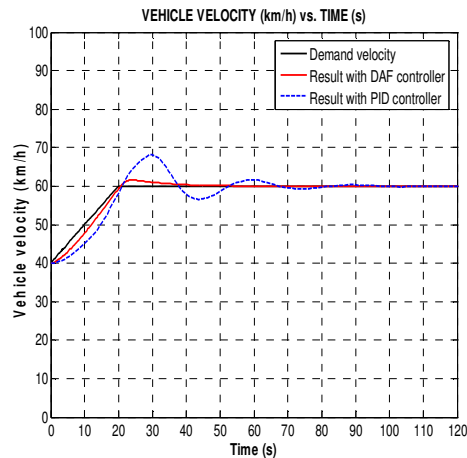
Hình 3.43 Đặc tính làm việc của bộ điều khiển mờ thích nghi trực tiếp khi mới được khởi tạo

4. So sánh kết quả điều khiển

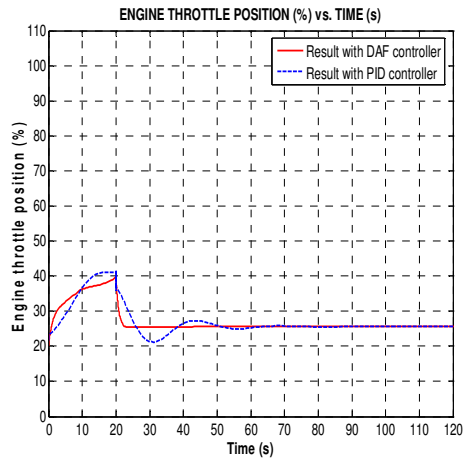
1. Trường hợp 1: Age = 100%, $G_t = 100\text{kg}$, $v_{\text{wind}} = 0 \text{ km/h}$, ô tô đi trên đường bằng Grade = 0° (xem hình 2.21).
2. Trường hợp 2: Age = 100%, $G_t = 500\text{kg}$, $v_{\text{wind}} = 0 \text{ km/h}$, ô tô đi trên đường bằng Grade = 0° (xem hình 2.22).
3. Trường hợp 3: Age = 100%, $G_t = 100\text{kg}$, $v_{\text{wind}} = 30 \text{ km/h}$, ô tô đi trên đường bằng Grade = 0° .
4. Trường hợp 4: Age = 100%, $G_t = 500\text{kg}$, $v_{\text{wind}} = 30 \text{ km/h}$, ô tô đi trên đường bằng Grade = 0° .
5. Trường hợp 5: Age = 85%, $G_t = 100\text{kg}$, $v_{\text{wind}} = 0 \text{ km/h}$, ô tô đi trên đường bằng Grade = 0° .
6. Trường hợp 6: Age = 85%, $G_t = 500\text{kg}$, $v_{\text{wind}} = 0 \text{ km/h}$, ô tô đi trên đường bằng Grade = 0° .
7. Trường hợp 7: Age = 85%, $G_t = 100\text{kg}$, $v_{\text{wind}} = 30 \text{ km/h}$, ô tô đi trên đường bằng Grade = 0° .
8. Trường hợp 8: Age = 85%, $G_t = 500\text{kg}$, $v_{\text{wind}} = 30 \text{ km/h}$, ô tô đi trên đường bằng Grade = 0° .
9. Trường hợp 9: Age = 100%, $G_t = 300\text{kg}$, $v_{\text{wind}} = 30 \text{ km/h}$, ô tô lên và xuống dốc Grade = 5° .
10. Trường hợp 10: Age = 85%, $G_t = 300\text{kg}$, $v_{\text{wind}} = 30 \text{ km/h}$, ô tô lên và xuống dốc Grade = 5° .



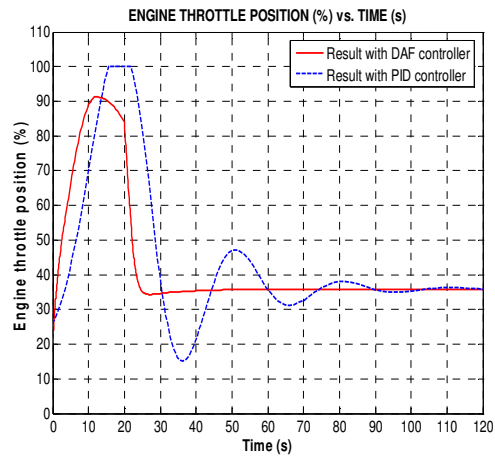
Hình 3.44.a



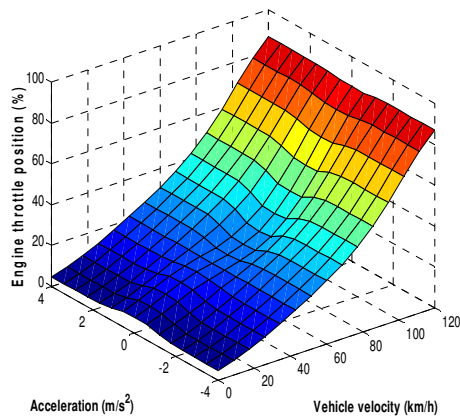
Hình 3.45.a



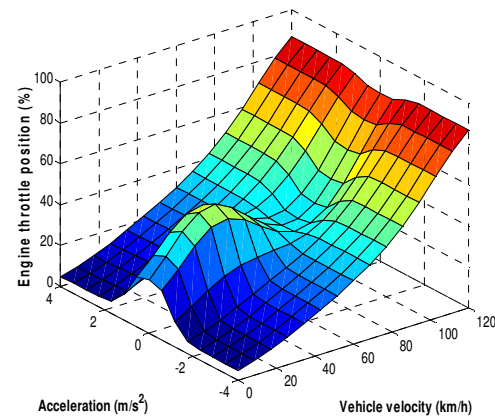
Hình 3.44.b



Hình 3.45.b



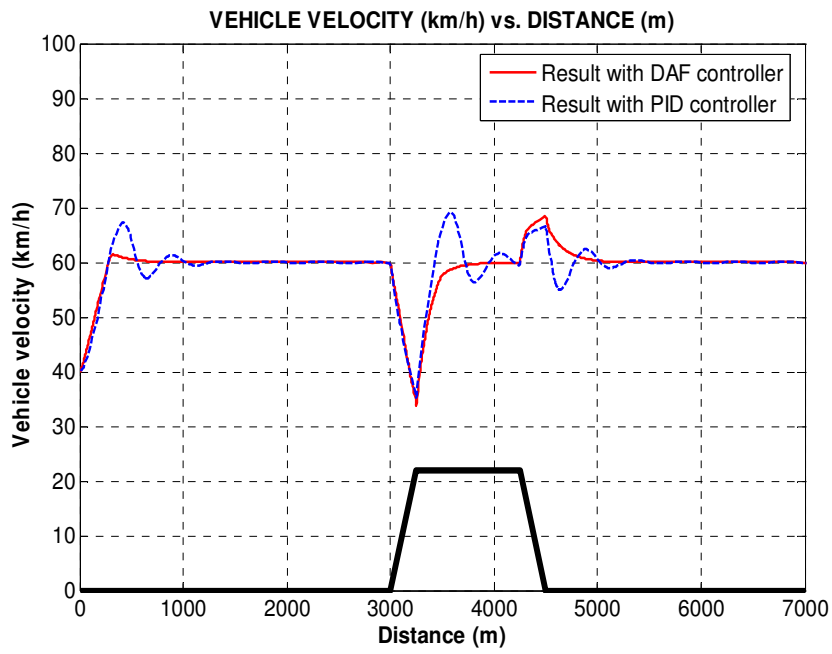
Hình 3.44.c



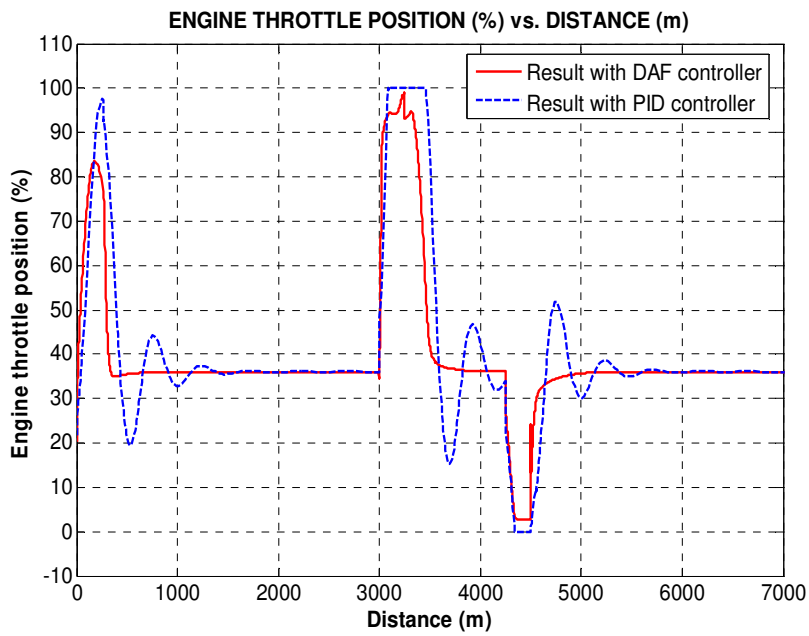
Hình 3.45.c

Trường hợp 1 (hình 3.44) và **2** (hình 3.45), vận tốc ô tô (a), độ mở bướm ga (b) và đặc tính làm việc của bộ điều khiển sau khi xác lập

Hình 3.46: Vận tốc (a) và độ mở bướm ga (b) khi ô tô lên và xuống dốc 5° ở trường hợp 9



Hình 3.46.a



Hình 3.46.b

Nhận xét:

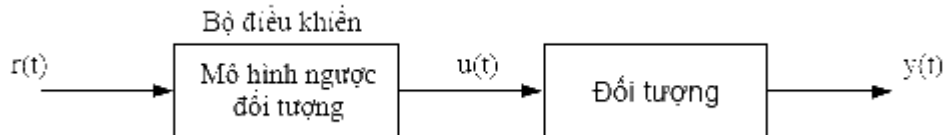
- Bộ điều khiển DAF với cấu trúc và những thông số được chọn thích hợp đã cho kết quả điều khiển tốt trong các trường hợp được khảo sát. Vận tốc ô tô được điều khiển với sai số xác lập bằng 0, ít vọt lố, không bị dao động, hệ thống không bị mất ổn định trong quá trình điều khiển.
- Với cùng một bộ giá trị ban đầu của các thông số, dù có sự thay đổi về tải trọng tác dụng xe, hao mòn trong hệ thống truyền động hay động cơ, ảnh hưởng của gió, sự thay đổi về độ dốc của mặt đường, bộ điều khiển DAF vẫn tự chỉnh định được các thông số và đảm bảo chất lượng điều khiển tốt.
- Do luật chỉnh định thông số được xây dựng trên tiêu chuẩn ổn định Lyapunov và vận tốc ô tô được điều chỉnh bám theo hàm dốc, mục tiêu đặt ra trong các trường hợp khảo sát này chỉ là luôn duy trì vận tốc ô tô không đổi ở giá trị 60 km/h; và đảm bảo hệ thống làm việc không bị mất ổn định hay bị trải qua những giai đoạn mất ổn định. Những mục tiêu khác như động cơ tiêu hao ít nhiên liệu nhất, hay gia tốc ô tô phải nằm trong giới hạn cho phép để đem lại cảm giác thoải mái, êm dịu cho người đi xe không thực hiện được.

3.6. Điều khiển thích nghi hệ động mạng DFNN:

Trên thực tế, hầu hết các đối tượng cần điều khiển là những hệ thống phi tuyến phức tạp và đặc tính động học của chúng luôn biến đổi theo thời gian. Việc điều khiển chúng chỉ có thể dựa trên một kỹ thuật duy nhất là điều khiển thích nghi. Trong thời gian gần đây, các bộ điều khiển nơ-ron mờ thích nghi với sự kết hợp các ưu điểm của logic mờ và mạng nơ-ron được xem là giải pháp hiệu quả nhất để giải quyết những vấn đề trên.

3.6.1. Điều khiển thích nghi dựa trên mô hình ngược DFNN

Một đối tượng bất kỳ có thể được điều khiển để bám theo một tín hiệu đặt bất kỳ bằng việc cung cấp tín hiệu đặt này tới một bộ điều khiển mà đặc tính động học của nó là nghịch đảo của đặc tính động học đối tượng.



Phương pháp điều khiển dựa trên mô hình ngược

Giả sử đối tượng cần điều khiển được mô tả bằng phương trình sai phân:

$$y(t-1) = g \{ y(t), \dots, y(t-n-1), u(t), \dots, u(t-m) \}$$

với $(.) g$ là 1 hàm phi tuyến bất kỳ

Bộ điều khiển được huấn luyện để học mô hình ngược của đối tượng:

$$\hat{u}(t) = \hat{g}^{-1} \{ y(t-1), y(t), \dots, y(t-n+1), u(t-1), \dots, u(t-m) \}$$

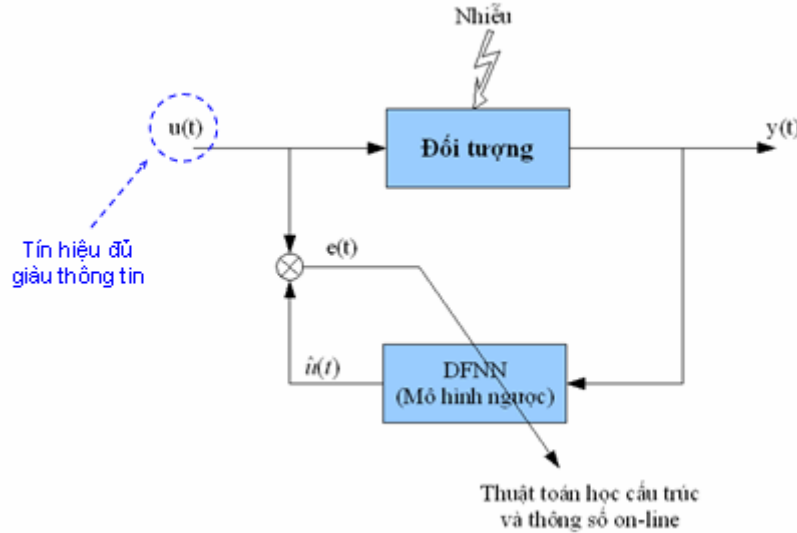
3.6.2. Quy trình điều khiển thích nghi với mạng DFNN

Quy trình này bao gồm 2 bước độc lập: nhận dạng on-line mô hình ngược đối tượng và điều khiển thích nghi đối tượng.

- **Bước 1:** Nhận dạng on-line mô hình ngược đối tượng

Đây là quá trình chỉnh định on-line cấu trúc và thông số cho mạng DFNN (dựa trên thuật toán đã đề cập trong chương 1) sao cho mạng có khả năng mô tả được đặc tính động học ngược của đối tượng cần điều khiển.

Muốn vậy, tín hiệu ngõ vào cung cấp cho đối tượng phải đủ giàu thông tin để tạo tín hiệu đáp ứng ngõ ra của đối tượng cũng đủ giàu thông tin. Các tập mẫu vào – ra như vậy sẽ giúp cho quá trình nhận dạng on-line mau hội tụ về mô hình ngược gần đúng nhất của đối tượng.



Nhận dạng on-line mô hình ngược đối tượng

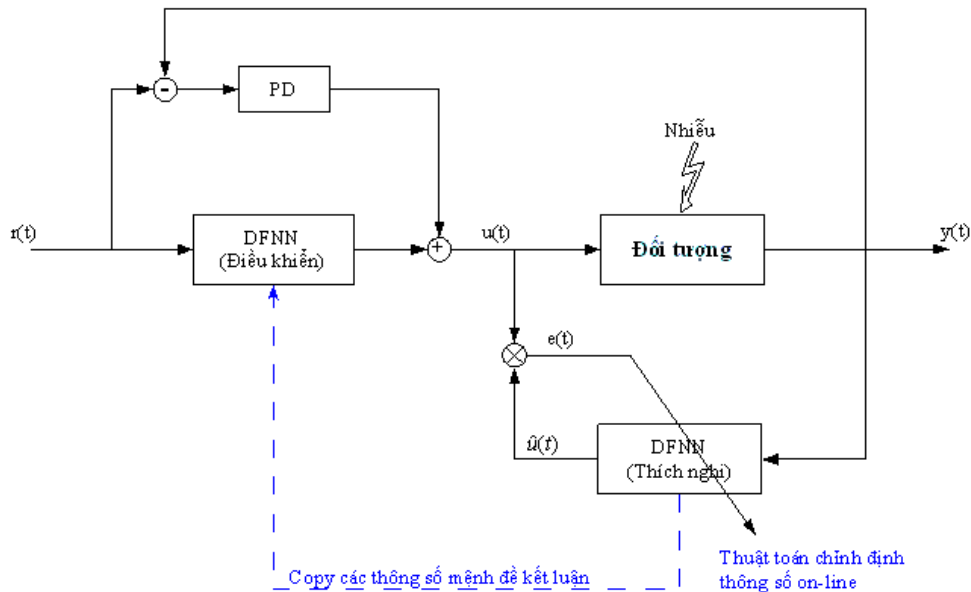
- **Bước 2:** Điều khiển thích nghi hệ động

Sau khi đã hoàn tất quá trình nhận dạng on-line mô hình ngược. Để có thể ứng dụng được mô hình ngược trên vào điều khiển thì bộ điều khiển phải có

khả năng thích nghi với những thay đổi trong đặc tính động học ngược của đối tượng và với nhiễu. Do đó, bên cạnh quá trình điều khiển đối tượng thì mạng DFNN vẫn phải tiếp tục tiến hành song song quá trình tự học on-line mô hình ngược của đối tượng.

Về bản chất, quá trình học on-line này hoàn toàn tương tự như ở bước 1. Tuy nhiên, do được thừa hưởng một cấu trúc mạng gần như hoàn chỉnh cho việc mô tả đặc tính động học ngược của đối tượng từ bước 1 nên trên thực tế, quá trình *chỉnh định cấu trúc* mạng DFNN rất khó xảy ra ở bước 2 (trừ khi có sự thay đổi rất lớn trong đặc tính động học của đối tượng dẫn đến đối tượng gần như trở thành một đối tượng mới thì quá trình chỉnh định cấu trúc mới có khả năng xảy ra). Vì vậy, trong bước này chủ yếu chỉ xảy ra quá trình *chỉnh định thông số* mệnh đề kết luận của mạng.

Với quá trình chỉnh định thông số, mạng DFNN vẫn đảm bảo được khả năng thích nghi với những thay đổi không quá lớn trong đặc tính động học ngược của đối tượng. Trong trường hợp đối tượng thay đổi quá lớn có thể dẫn tới cần phải thay đổi cấu trúc mạng thì nên tiến hành lại bước 1 để nhận dạng một mô hình ngược mới hoàn toàn về đối tượng.



Điều khiển thích nghi hệ động

Quá trình điều khiển thích nghi hệ động bao gồm 2 giai đoạn:

- Giai đoạn điều khiển: quá trình tính toán được lan truyền từ tín hiệu đặt, qua mạng DFNN, đến tín hiệu ngõ ra của mạng (tín hiệu điều khiển).

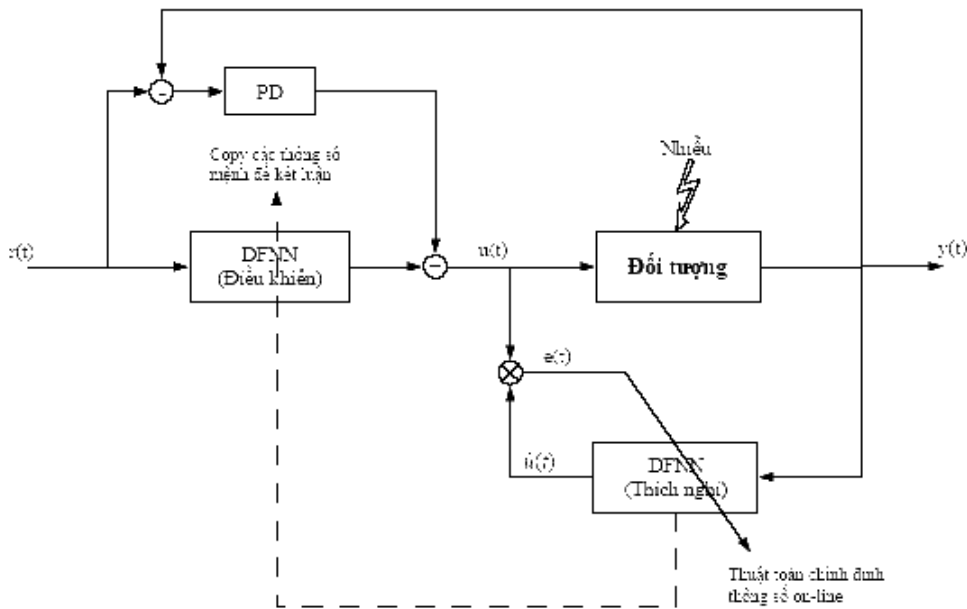
- Giai đoạn thích nghi: các dữ liệu ngõ vào (tín hiệu điều khiển) và ngõ ra của đối tượng được sử dụng để chỉnh định các thông số mệnh đề kết luận của mạng.

Chú ý:

Theo mô hình trên, mặc dù hai giai đoạn trên được biểu diễn bằng hai mạng DFNN, mỗi mạng thực hiện một chức năng riêng biệt, tuy nhiên trên thực tế thì chỉ cần sử dụng một cấu trúc mạng DFNN nhưng tích hợp cả hai giai đoạn trên vào quá trình thực thi của thuật toán mạng (do trong giai đoạn thích nghi đã bỏ qua quá trình chỉnh định cấu trúc mạng).

3.6.3. Điều khiển thích nghi hệ động với khâu PD bù sai số

Trên thực tế, đối với một số đối tượng điều khiển phức tạp, không phải bất cứ khâu phi tuyến nào trong đặc tính động học ngược của đối tượng cũng có thể nhận dạng và mô hình hóa được. Điều này dẫn đến sai số cấu trúc trong mô hình.



Điều khiển thích nghi hệ động với khâu PD bù sai số

Mặc khác, sai số còn xuất hiện trong mô hình do các nhiễu. Nhiễu làm thông tin thu thập cho quá trình nhận dạng (các mẫu dữ liệu vào – ra) bị sai lệch dẫn đến sự không chính xác trong mô hình nhận dạng. Ngoài ra, tương tự như bất kỳ một thuật toán chỉnh định thông số nào khác, do đặc tính

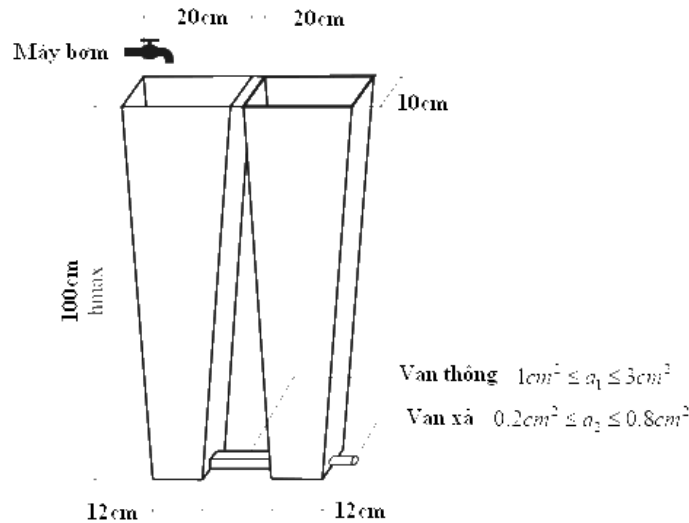
chỉnh định on-line nên các thông số chỉnh định trong mỗi chu kỳ lấy mẫu có thể dao động với một biên độ nhỏ xung quanh giá trị chính xác của nó. Sự dao động này cũng có thể dẫn đến sai số ngõ ra mô hình. Để khắc phục các sai số trên, lưu đồ điều khiển thích nghi cần phải được bổ sung một bộ điều khiển phụ PD kinh điển (bộ điều khiển tỉ lệ, vi phân):

Tùy theo sai số ngõ ra mà bộ điều khiển phụ này sẽ bù vào tín hiệu điều khiển thu được từ ngõ ra mạng DFNN một lượng thích hợp. Bộ điều khiển phụ PD này còn có thể hỗ trợ tích cực cho quá trình điều khiển tại thời điểm đặc tính động học của đối tượng vừa mới thay đổi và bộ điều khiển mô hình ngược chưa kịp thích nghi với đặc tính động học mới.

3.6.4. Ví dụ minh họa:

Ví dụ 3.23:

Sử dụng công cụ Simulink của MATLAB 7.0 để xây dựng mô hình và thuật toán điều khiển thích nghi hệ bồn chứa nước nối tiếp có thông số các van thay đổi



Mô hình thực của hệ bồn chứa nước nối tiếp

Đặt vấn đề:

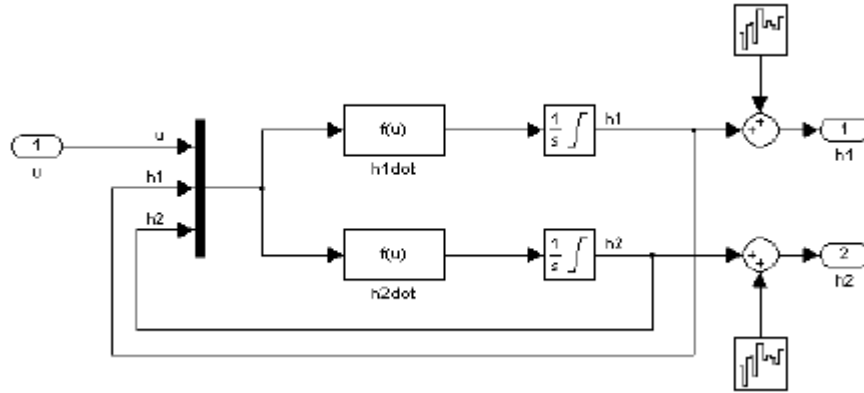
Mục tiêu của thuật toán là điều khiển độ cao mực nước ở bồn 2 theo tín hiệu đặt. Về bản chất, hệ mô hình bồn chứa nước nối tiếp là hệ SISO:

- **Ngõ vào:** điện áp điều khiển máy bơm $[u(t)]$
- **Ngõ ra:** độ cao mực nước bồn 2 $[h_2(t)]$
- **Thông số thay đổi:** tiết diện 2 van $[a_1, a_2]$

Giải pháp:

Sử dụng 2 mạng DFNN để nhận dạng on-line mô hình ngược từng bồn nước

- Mạng DFNN nhận dạng mô hình ngược bồn 1: mạng điều khiển
- Mạng DFNN nhận dạng mô hình ngược bồn 2: mạng dự báo



Mô hình Simulink của hệ bồn nước nối tiếp với các nhiễu đo lường ở ngõ ra

Hệ phương trình toán liên tục của hệ bồn nước nối tiếp:

$$\begin{cases} \dot{h}_1(t) = \frac{ku(t) - a_1 \operatorname{sgn}\{h_1(t) - h_2(t)\} \sqrt{2g|h_1(t) - h_2(t)|}}{\left[(L_{1\max} - L_{1\min}) \frac{h_1(t)}{h_{\max}} + L_{1\min} \right] W} \\ \dot{h}_2(t) = \frac{a_1 \operatorname{sgn}\{h_1(t) - h_2(t)\} \sqrt{2g|h_1(t) - h_2(t)|} - a_2 \sqrt{2gh_2(t)}}{\left[(L_{2\max} - L_{2\min}) \frac{h_2(t)}{h_{\max}} + L_{2\min} \right] W} \end{cases}$$

Quy trình nhận dạng và điều khiển

- **Bước 1: Nhận dạng on-line mô hình ngược**
Rời rạc hóa phương trình toán liên tục của hệ bồn nước:

$$\begin{cases} \dot{h}_1(k+1) = \dot{h}_1(k) + \frac{T \left[ku(k) - a_1 \operatorname{sgn} \{h_1(k) - h_2(k)\} \sqrt{2g|h_1(k) - h_2(k)|} \right]}{\left[(L_{1\max} - L_{1\min}) \frac{h_1(k)}{h_{\max}} + L_{1\min} \right] W} \\ \dot{h}_2(k-1) = \dot{h}_2(k) + \frac{T \left[a_1 \operatorname{sgn} \{h_1(k) - h_2(k)\} \sqrt{2g|h_1(k) - h_2(k)|} - a_2 \sqrt{2gh_2(k)} \right]}{\left[(L_{2\max} - L_{2\min}) \frac{h_2(k)}{h_{\max}} + L_{2\max} \right] W} \end{cases}$$

Trong đó: T là chu kỳ lấy mẫu dữ liệu.

Tổng quát và đơn giản hóa hệ trên ta có:

$$\dot{h}_1(k-1) = f_1 \{h_1(k), h_2(k), u(k)\} \quad (1)$$

$$\dot{h}_2(k-1) = f_2 \{h_1(k), h_2(k)\} \quad (2)$$

Từ phương trình (2) ta có:

$$\dot{h}_2(k-2) = f_2 \{h_1(k+1), h_2(k+1)\}$$

$$\Rightarrow \dot{h}_2(k-2) = f_2 \{f_1 \{h_1(k), h_2(k), u(k)\}, h_2(k-1)\}$$

Phương trình trên có thể viết lại bằng cách sử dụng một hàm phi tuyến $g\{\cdot\}$:

$$\dot{h}_2(k-2) = g \{h_2(k+1), h_2(k), h_1(k), u(k)\}$$

Ta thấy quan hệ giữa ngõ vào u và ngõ ra h_2 là một **quan hệ có trễ** (trễ 1 chu kỳ lấy mẫu). **Mô hình có trễ là một mô hình không tồn tại mô hình ngược** (mô hình *bất khả đảo* – noninvertible model).

Vì thế, ta sẽ khắc phục bằng cách tách đối tượng ra làm 2 bồn riêng biệt.

Khi đó, quan hệ vào ra của mỗi bồn là một quan hệ không có trễ:

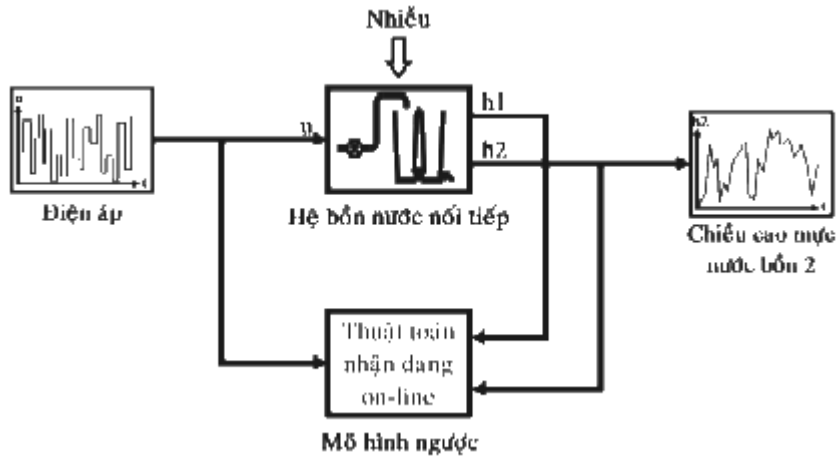
$$\begin{cases} u(k) = f_1^{-1} \{h_1(k-1), h_1(k), h_2(k)\} \\ h_1(k) = f_2^{-1} \{h_2(k+1), h_2(k)\} \end{cases}$$

Để nhận dạng mô hình ngược của mỗi bồn nước thì phương trình viết lại là:

$$\begin{cases} u(k-1) = f_1^{-1} \{h_1(k), h_1(k-1), h_2(k-1)\} \\ h_1(k-1) = f_2^{-1} \{h_2(k), h_2(k-1)\} \end{cases}$$

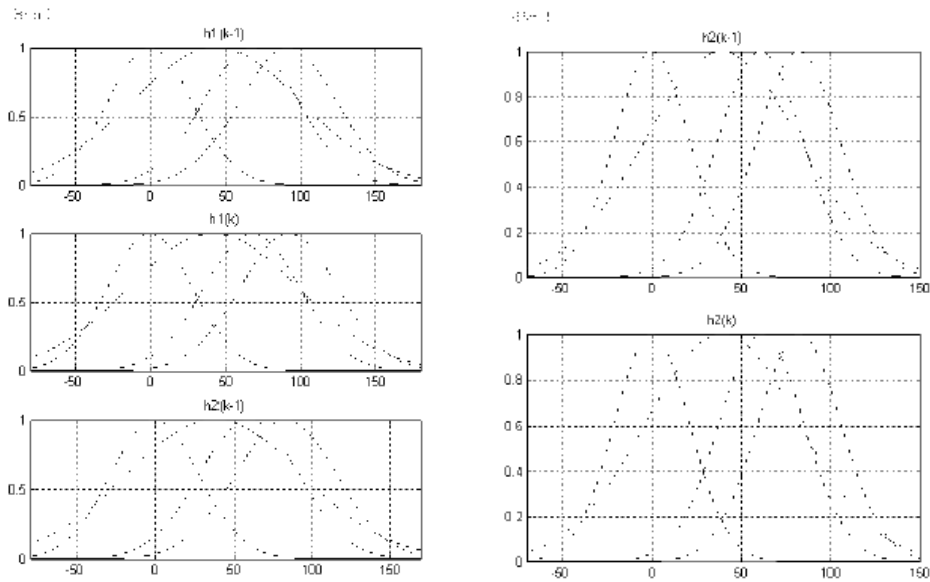
Trong đó: $f_1^{-1}(\cdot)$ và $f_2^{-1}(\cdot)$ là cấu trúc mạng DFNN nhận dạng mô hình ngược của bồn 1 và bồn 2 tương ứng.

Do hệ bồn chứa nước là hệ đáp ứng tương đối chậm nên u có thể được chọn là dãy xung vuông biên độ ngẫu nhiên với chu kỳ đủ lớn $0V \leq u \leq 12V$

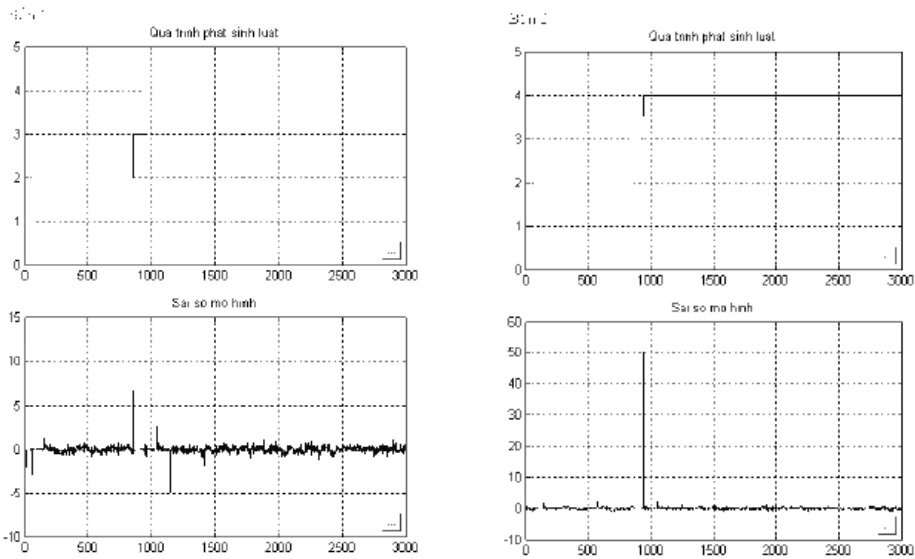


Lưu đồ nhận dạng mô hình ngược hệ bồn chứa nước nối tiếp

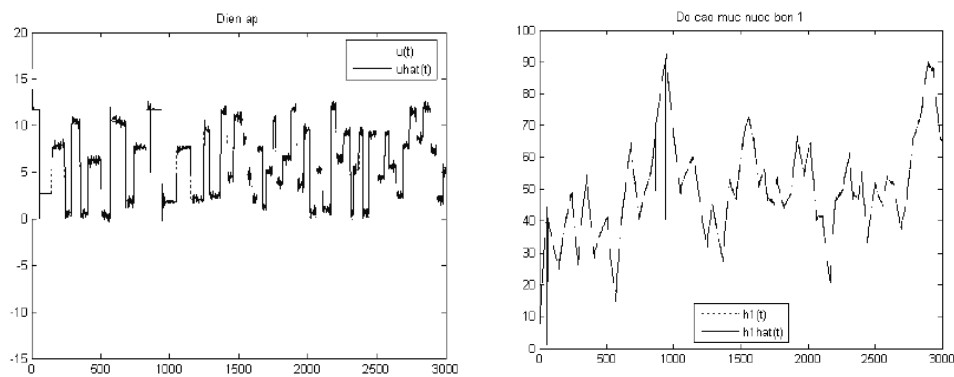
Mô hình ngược 2 bồn được tiến hành nhận dạng trong điều kiện có nhiễu đo lường ở cả h_1 và h_2 (là sự dao động của mặt nước khi lấy mẫu). Chu kỳ lấy mẫu nhận dạng là 2s, tương ứng với 1500 mẫu dữ liệu. Tiết diện các van khi nhận dạng là $a_1 = 2 \text{ cm}^2$ và $a_2 = 0.5 \text{ cm}^2$. Kết quả của quá trình nhận dạng:



Các hàm liên thuộc lớp 2 cấu trúc mạng DFNN



Sự phát sinh nơ-ron luật và sai số ra quá trình nhận dạng



Ngõ ra mạng DFNN bồn 1 (\hat{u}) và bồn 2 (\hat{h}_1) trong quá trình nhận dạng

Các thông số của thuật toán nhận dạng mô hình ngược bồn 1 được chọn như sau: $d_{\max} = 100$, $d_{\min} = 40$, $\gamma = 0.977$, $e_{\max} = 2$, $e_{\min} = 0.01$, $\beta = 0.977$, $\sigma_c = 40$, $k_{\text{tr}} = 1.1$, $\gamma_p = 10^2$, $\lambda = 0.99$, $-\lambda = 0.1$, $k_{\text{qtr}} = 0.0002$

Các thông số của thuật toán nhận dạng mô hình ngược bồn 2 được chọn như sau: $d_{\max} = 100$, $d_{\min} = 30$, $\gamma = 0.977$, $e_{\max} = 2$, $e_{\min} = 0.01$, $\beta = 0.977$, $\sigma_c = 30$, $k_{\text{tr}} = 1.1$, $\gamma_p = 10^2$, $\lambda = 0.99$, $-\lambda = 0.1$, $k_{\text{qtr}} = 0.0002$

Bảng Kết quả nhận dạng mô hình ngược bồn 1

Số nơ-ron luật	4
Số thông số	32
Vị trí các hàm liên thuộc ứng với các ngõ vào	$h_1(k-1): \{0 \ 44.16 \ 66.99 \ 91.48\}$ $h_1(k): \{0.03 \ 42.28 \ 67.68 \ 92.02\}$ $h_2(k-1): \{0 \ 38.42 \ 60.21 \ 83.72\}$
Bề rộng các hàm liên thuộc ứng với các nơ-ron luật ngõ vào	$\{40 \ 79.41 \ 44.56 \ 45.94\}$
Thông số các mệnh đề kết luận	$\{-0.75 \ -0.75 \ 1.72 \ -1.71 \ 15.38 \ -1.96$ $2.86 \ -1.28 \ 15.01 \ -2.08 \ 3.04 \ -1.04$ $7.27 \ -3.21 \ 3.91 \ -0.68\}$

Bảng Kết quả nhận dạng mô hình ngược bồn 2

Số nơ-ron luật	4
Số thông số	24
Vị trí các hàm liên thuộc ứng với các ngõ vào	$h_2(k-1): \{0 \ 39.39 \ 60.87 \ 82.14\}$ $h_2(k): \{0.03 \ 39.48 \ 61.44 \ 82.7\}$
Bề rộng các hàm liên thuộc ứng với các nơ-ron luật ngõ vào	$\{30 \ 61.32 \ 33.8 \ 33.08\}$
Thông số các mệnh đề kết luận	$\{-0.92 \ -0.92 \ 1.89 \ 7.32 \ -1.4 \ 2.21$ $10.37 \ -2.19 \ 3.16 \ 6.85 \ -4.52 \ 5.55\}$

➤ **Bước 2: Điều khiển thích nghi hệ động**

Đối với các đối tượng có khâu trễ, tuy không đo lường trực tiếp được nhưng ngõ vào có thể dự báo được. Mô hình ngược của bồn được viết lại như sau:

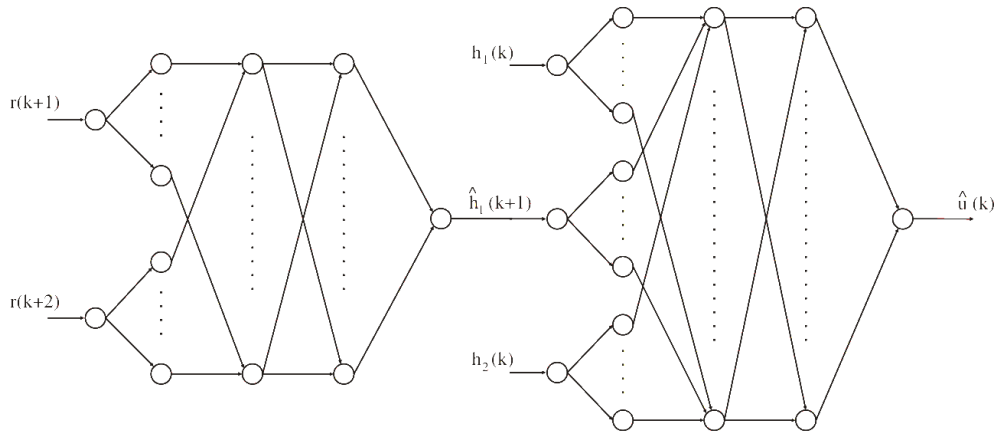
$$\begin{cases} u(k) = f_1^{-1} \{h_1(k-1), h_1(k), h_2(k)\} \\ h_1(k-1) = f_2^{-1} \{h_2(k-2), h_2(k-1)\} \end{cases}$$

Khi thay các tín hiệu ngõ ra tương lai $h_2(k+2)$ và $h_2(k+1)$ bằng các tín hiệu ngõ ra mong muốn (tín hiệu đặt) $r(k+2)$ và $r(k+1)$, có thể dự báo được độ cao mực nước bồn 2 tại thời điểm tiếp theo trong tương lai là:

$$\hat{h}_1(k-1) = f_2^{-1} \{r(k-2), r(k+1)\}$$

Tín hiệu điều khiển tại thời điểm hiện tại:

$$\hat{u}(k) = f_1^{-1} \left\{ \hat{h}_1(k+1), h_1(k), h_2(k) \right\}$$



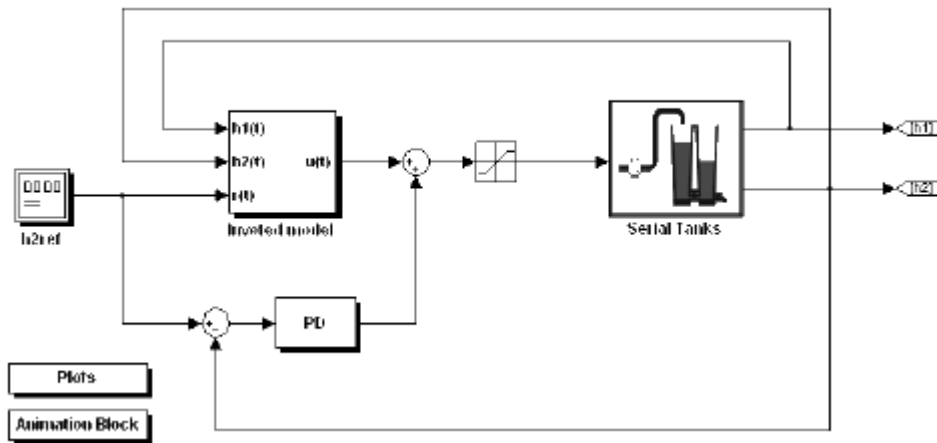
Mạng DFNN mô hình ngược bậc 2
(Mạng dự báo)

Mạng DFNN mô hình ngược bậc 1
(Mạng điều khiển)

Bên cạnh quá trình điều khiển (sử dụng mô hình ngược), quá trình nhận dạng thông số on-line của 2 mạng vẫn phải được tiếp tục tiến hành song song để tạo khả năng thích nghi cho quá trình điều khiển.

Ta tiến hành khảo sát các trường hợp điều khiển khi thông số van thay đổi:

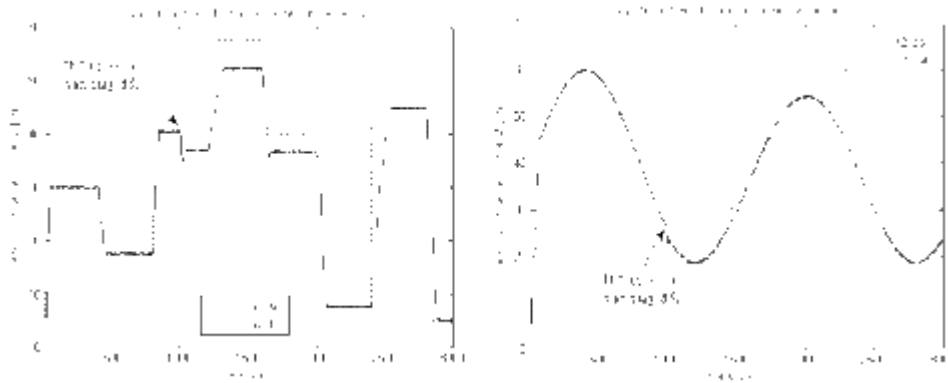
➤ Điều khiển không thích nghi khi thông số các van thay đổi



Lưu đồ điều khiển không thích nghi

Khi điều khiển, 1000s đầu tiên thông số 2 van vẫn giữ nguyên như bước 1 ($a_1 = 2 \text{ cm}^2$ và $a_2 = 0.5 \text{ cm}^2$). Tiết diện van thông và van xả chỉ thay đổi đến

giá trị mới là $a_1 = 1 \text{ cm}^2$ và $a_2 = 0.8 \text{ cm}^2$ từ giây thứ 1001 đến khi kết thúc quá trình điều khiển (3000s)..

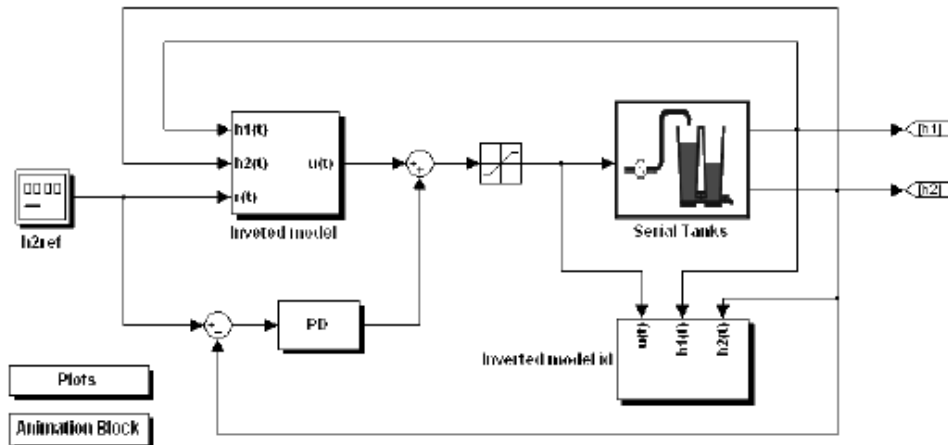


Quá trình điều khiển không thích nghi với tín hiệu đặt có dạng vuông, sin

Nhận xét:

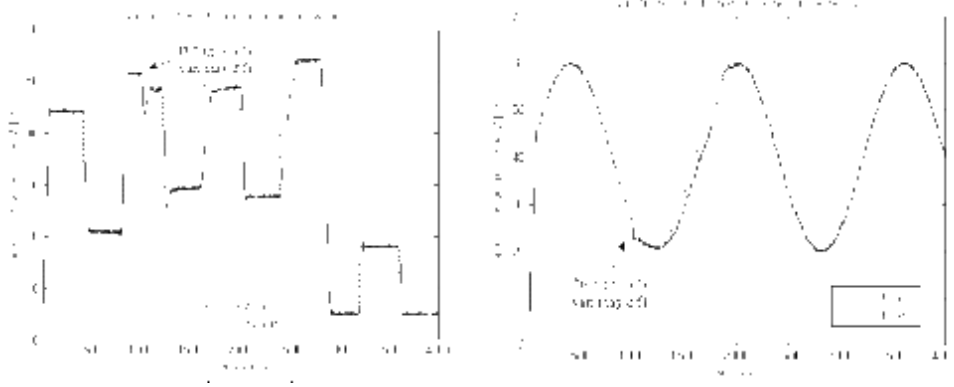
Ban đầu, khi đặc tính động học đối tượng không thay đổi, đối tượng được điều khiển rất chính xác. Tuy nhiên, khi đặc tính động học đối tượng bắt đầu thay đổi, do không có sự thích nghi, bộ điều khiển được huấn luyện ở bước 1 tỏ ra không còn phù hợp với đặc tính động học ngược mới của đối tượng và quá trình điều khiển xuất hiện sai số điều khiển tương đối lớn.

➤ **Điều khiển thích nghi khi thông số các van thay đổi**



Lưu đồ điều khiển thích nghi hệ bồn nước nối tiếp trong Simulink

Quá trình điều khiển tương tự như trường hợp trên. Tuy nhiên, thuật toán điều khiển thực thi luôn cả giai đoạn điều khiển và giai đoạn thích nghi.



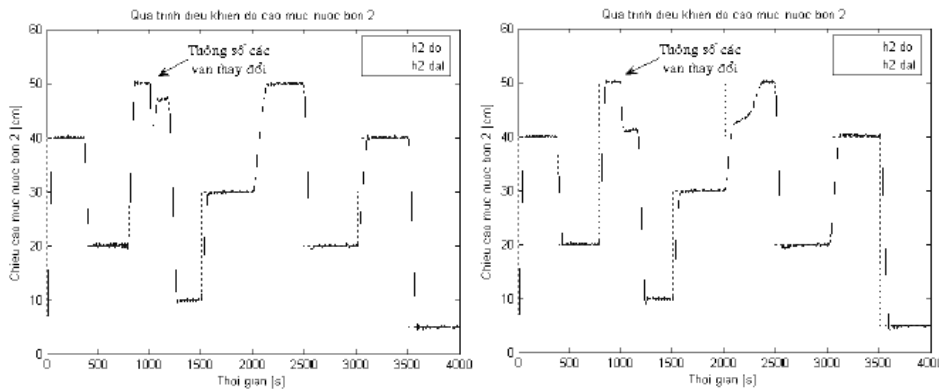
Quá trình điều khiển thích nghi với tín hiệu đặt có dạng vuông, sin

Nhận xét:

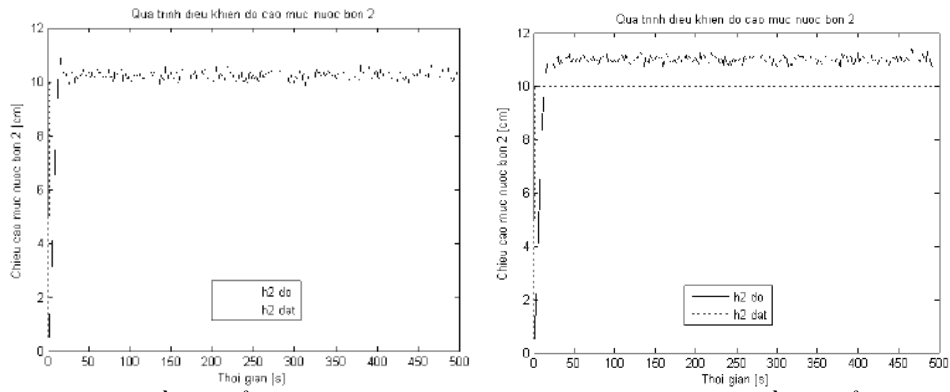
Bộ điều khiển thích nghi mô hình ngược có thể điều khiển chính xác đối tượng ngay cả khi thông số đối tượng thay đổi và với mọi dạng tín hiệu đặt. Tại thời điểm đặc tính động học đối tượng thay đổi, sai số điều khiển có thể tương đối lớn do quá trình huấn luyện chưa kịp thích nghi. Tuy nhiên, chỉ sau một thời gian ngắn huấn luyện on-line, bộ điều khiển mô hình ngược đã có thể thích nghi được với đặc tính động học ngược mới của đối tượng.

➤ **Vai trò của bộ điều khiển phụ PD kinh điển:**

Ta sẽ xem xét sự ảnh hưởng của bộ điều khiển phụ PD trong cả 2 quá trình: **điều khiển thích nghi và điều khiển không thích nghi.**



Quá trình điều khiển thích nghi có và không có bộ điều khiển phụ PD



Quá trình điều khiển không thích nghi có và không có bộ điều khiển phụ PD

Nhận xét:

Trong cả 2 trường hợp, ta thấy khi không có bộ điều khiển phụ PD hỗ trợ, quá trình thích nghi tỏ ra chậm hơn. Thật ra, tốc độ thích nghi là như nhau. Tuy nhiên, khi bộ điều khiển mô hình ngược chưa kịp thích nghi với đặc tính động học mới thì bộ điều khiển phụ PD đã hỗ trợ rất nhiều trong việc điều khiển đối tượng, hạn chế phần lớn sai số điều khiển. Như vậy, bộ điều khiển phụ PD đã giúp cho việc ổn định hóa và nâng cao chất lượng điều khiển của hệ thống.

CÂU HỎI ÔN TẬP VÀ BÀI TẬP

1. Thế nào là điều khiển thích nghi ?
2. Tại sao phải điều khiển thích nghi ?
3. Lưu đồ giải thuật thiết kế luật MIT?
4. Xét tính ổn định của vòng điều khiển thích nghi trong ví dụ 3.4, ứng dụng Matlab mô phỏng kết quả thu được trong hai trường hợp:
 1. Sử dụng luật MIT.
 2. Sử dụng luật hiệu chỉnh bổ sung.
5. Hệ thích nghi mô hình tham chiếu MRAS ?
 - Sơ đồ nguyên lý.
 - Nội dung phương pháp gradient và lưu đồ giải thuật.
6. Ứng dụng Matlab mô phỏng ví dụ 3.2: MRAS cho hệ bậc nhất.
7. Thiết kế MRAS dùng lý thuyết ổn định Lyapunov.
8. Thế nào là thiết kế bộ điều khiển hồi tiếp tuyến tính ngõ ra?
9. Xét hệ thống bậc 2 không dao động có hàm truyền đạt như sau:

$$G(s) = \frac{k}{s(a_0 + a_1s + s^2)}$$

Trong đó:

a_0, a_1, k là những tham số (dương) chưa biết và phụ thuộc vào t . Ta điều khiển đối tượng này bằng bộ PID tự chỉnh gián tiếp và một bộ tiền xử lý để làm giảm độ vọt lố của hệ kín.

1. Thiết kế cơ cấu chỉnh định các tham số cho hai bộ điều khiển trên.
 2. Xác định các giả thiết cần phải có về tốc độ thay đổi các tham số a_0, a_1, k (nhanh/chậm như thế nào) để hệ thống thích nghi trên làm việc đạt chất lượng yêu cầu?
10. Xét đối tượng bất định có hàm truyền:

$$G(s) = \frac{k}{1 + Ts} = \frac{Y(s)}{U(s)}$$

Trong đó: k, T là hai hằng số dương chưa biết.

Thiết kế bộ điều khiển thích nghi ngõ ra $u = -kx$

Với x là biến trạng thái

$$x_1 = y$$

$$x_2 = \dot{x}_1 = \dot{y}$$

Sao cho hệ kín có hàm truyền là:

$$G_k(s) = \frac{1}{1+3s}$$

11. Ứng dụng Matlab mô phỏng ví dụ 3.9 và ví dụ 3.10 dùng lý thuyết ổn định Lyapunov.
12. Bộ tự chỉnh định STR gián tiếp.
13. Ứng dụng Matlab mô phỏng ví dụ 3.13: Bộ tự chỉnh định gián tiếp với nhiễu xác định.
14. Dùng Matlab mô phỏng ví dụ 3.14: Bộ tự chỉnh định gián tiếp với nhiễu ngẫu nhiên.
15. Bộ tự chỉnh định STR trực tiếp.
16. Điều khiển dự báo thích nghi.
17. So sánh giữa MRAS và STR.
28. Chỉnh định tự động.
19. Lịch trình độ lợi.
20. Dùng Matlab mô phỏng ví dụ 3.11 để kiểm định lại kết quả hình 3.19.
21. Ứng dụng Matlab mô phỏng ví dụ 3.12: Tay máy hai khớp nối
22. Dùng Matlab mô phỏng ví dụ 3.19: Bộ điều khiển mờ thích nghi gián tiếp cho hệ con lắc ngược.
23. Dùng Matlab mô phỏng ví dụ 3.20: Bộ điều khiển mờ thích nghi trực tiếp.
24. Cho quá trình được mô tả bởi:

$$G(s) = \frac{50}{s(s+4)}$$

Và có chỉ tiêu chất lượng như sau:

$$\int_0^{\infty} \left((y - u_c)^2 + \rho u^2 \right) dt$$

Biết rằng, luật điều khiển có dạng:

$$u(t) = -s_0 (y - u_c)$$

Hay

$$u(t) = -\frac{s_0 p + s_1}{p + r_1} (y - u_c)$$

Sử dụng phương pháp gradients tối ưu hóa một cách chính xác các thông số của hệ thống từ mô hình ước lượng quá trình.

25. Cho quá trình được mô tả bởi:

$$\dot{x} = A(\theta)x + B(\theta)u_c$$

$$y = C(\theta)x$$

Và mô hình kèm theo là:

$$\dot{x}_m = A_m x_m + B_m u_c$$

$$y_m = C_m x_m$$

Giả sử rằng: bậc của quá trình và bậc của mô hình kèm theo là giống nhau. Và tất cả các trạng thái đều ổn định.

(a) Tìm điều kiện để có thể đạt được các trạng thái trong mô hình hoàn hảo kèm theo.

$$(\text{đặt } e = x - x_m)$$

(b) Sử dụng lý thuyết Lyapunov để đạt được cơ cấu chỉnh định ổn định

$$(\text{xác định } \dot{A}(\theta) \text{ và } \dot{B}(\theta))$$

(Gợi ý:

Sử dụng hàm Lyapunov sau:

$$V = e^T P e + \text{tr}(A - A_m)^T Q_a (A - A_m) + \text{tr}(B_m - B)^T Q_b (B_m - B)$$

Trong đó:

$$e = x - x_m).$$

GIỚI THIỆU LÝ THUYẾT ĐIỀU KHIỂN TỰ ĐỘNG

Điều khiển tự động đóng vai trò quan trọng trong sự phát triển của khoa học và kỹ thuật. Lĩnh vực này hữu hiệu khắp nơi từ hệ thống phi thuyền không gian, hệ thống điều khiển tên lửa, máy bay không người lái, người máy, tay máy trong các quy trình sản xuất hiện đại, và ngay cả trong đời sống hàng ngày: điều khiển nhiệt độ, độ ẩm...

Phát minh đầu tiên khởi đầu cho việc phát triển của lĩnh vực điều khiển tự động là bộ điều tốc ly tâm để điều chỉnh nhiệt độ máy hơi nước của Jame Watt năm 1874. Các công trình đáng chú ý trong bước đầu phát triển lý thuyết điều khiển là của các nhà khoa học Minorsky, Hazen, Nyquist... năm 1922. Minorsky thực hiện hệ thống điều khiển tự động các con tàu và chứng minh tính ổn định của hệ thống có thể được xác định từ phương trình vi phân mô tả hệ thống. Năm 1932, Nyquist đã đưa ra một nguyên tắc tương đối đơn giản để xác định tính ổn định của hệ thống vòng kín dựa trên cơ sở đáp ứng vòng hở đối với các tín hiệu vào hình sin ở trạng thái xác lập. Năm 1934, Hazen đã giới thiệu thuật ngữ điều chỉnh cơ tự động (servo mechanism) cho những hệ thống điều khiển định vị và thảo luận đến việc thiết kế hệ thống relay điều chỉnh động cơ với ngõ vào tín hiệu thay đổi.

Trong suốt thập niên 40 của thế kỷ 20 phương pháp đáp ứng tần số đã giúp cho các kỹ sư thiết kế các hệ thống vòng kín tuyến tính thỏa các yêu cầu chất lượng điều khiển. Từ cuối thập niên 40 cho đến đầu thập niên 50 phương pháp quỹ đạo nghiệm của Evan được phát triển khá toàn vẹn.

Phương pháp quỹ đạo nghiệm và đáp ứng tần số được xem là cốt lõi của lý thuyết điều khiển cổ điển cho phép ta thiết kế được những hệ thống ổn định và thỏa các chỉ tiêu chất lượng điều khiển. Những hệ thống này được chấp nhận nhưng chưa phải là tối ưu, hoàn thiện nhất. Cho tới cuối thập niên 50 của thế kỷ 20 việc thiết kế một hay nhiều hệ thống dần dần được chuyển qua việc thiết kế một hệ thống tối ưu với ý nghĩa đầy đủ hơn.

Khi các máy móc hiện đại ngày càng phức tạp hơn với nhiều tín hiệu vào và ra thì việc mô tả hệ thống điều khiển hiện đại này đòi hỏi một lượng rất lớn các phương trình. Lý thuyết điều khiển cổ điển liên quan các hệ thống một ngõ vào và một ngõ ra trở nên bất lực để phân tích các hệ thống nhiều đầu vào, nhiều đầu ra. Kể từ khoảng năm 1960 trở đi nhờ máy tính số cho phép ta phân tích các hệ thống phức tạp trong miền thời gian, lý thuyết điều khiển hiện đại phát triển để đối phó với sự phức tạp của các hệ thống hiện đại. Lý thuyết điều khiển hiện đại dựa trên phân tích trong miền thời gian và tổng hợp dùng các biến trạng thái, cho phép giải các bài toán điều khiển có các yêu cầu chặt chẽ về độ chính xác, trọng lượng và giá thành của các hệ thống trong lĩnh vực kỹ nghệ không gian và quân sự.

Sự phát triển gần đây của lý thuyết điều khiển hiện đại là trong nhiều lĩnh vực điều khiển tối ưu của các hệ thống ngẫu nhiên và tiền định. Hiện nay máy vi tính ngày càng rẻ, gọn nhưng khả năng xử lý lại rất mạnh nên nó được dùng như là một phần tử trong các hệ thống điều khiển. Những áp dụng gần đây của lý thuyết điều khiển hiện đại vào ngay cả những ngành kỹ thuật như: sinh học, y học, kinh tế, kinh tế xã hội.

I. NHỮNG KHÁI NIỆM CƠ BẢN

1. Điều khiển học (Cybernetics):

Là khoa học nghiên cứu những quá trình điều khiển và truyền thông máy móc, sinh vật và kinh tế. Điều khiển học mang đặc trưng tổng quát và được phân chia thành nhiều lĩnh vực khác nhau như: toán điều khiển, điều khiển học kỹ thuật, điều khiển học sinh vật (phỏng sinh vật: bionics), điều khiển học kinh tế.

2. Lý thuyết điều khiển tự động:

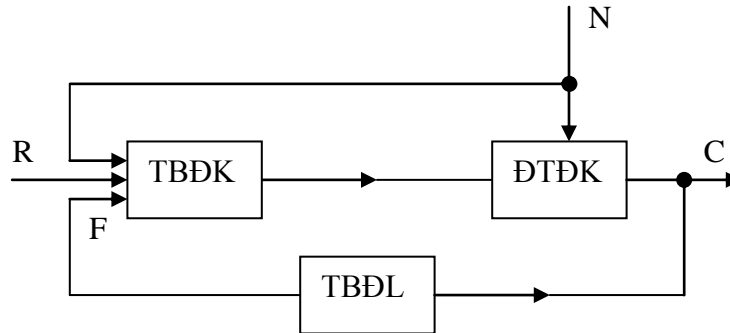
Là cơ sở lý thuyết của điều khiển học kỹ thuật. Điều khiển tự động là thuật ngữ chỉ quá trình điều khiển một đối tượng trong kỹ thuật mà không có sự tham gia của con người (automatic) nó ngược lại với quá trình điều khiển bằng tay (manual).

3. Hệ thống điều khiển tự động:

Một hệ thống điều khiển tự động bao gồm 3 phần chủ yếu:
Thiết bị điều khiển (TBĐK).

- Đối tượng điều khiển (ĐTĐK).
- Thiết bị đo lường.

Hình 1.1 là sơ đồ khối của hệ thống điều khiển tự động.



Hình 1.1

Trong đó:

C: tín hiệu cần điều khiển, thường gọi là tín hiệu ra (output).

U: tín hiệu điều khiển.

R: tín hiệu chủ đạo, chuẩn, tham chiếu (reference) thường gọi là tín hiệu vào (input).

N: tín hiệu nhiễu tác động từ bên ngoài vào hệ thống.

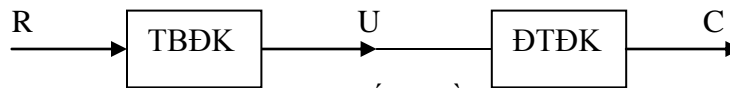
F: tín hiệu hồi tiếp, phản hồi (feedback).

4. Hệ thống điều khiển kín (closed loop control system):

Là hệ thống điều khiển có phản hồi (feedback) nghĩa là tín hiệu ra được đo lường và đưa về thiết bị điều khiển. Tín hiệu hồi tiếp phối hợp với tín hiệu vào để tạo ra tín hiệu điều khiển. Hình 1.1 chính là sơ đồ của hệ thống kín. Cơ sở lý thuyết để nghiên cứu hệ thống kín chính là lý thuyết điều khiển tự động.

5. Hệ thống điều khiển hở:

Đối với hệ thống hở, khâu đo lường không được dùng đến. Mọi sự thay đổi của tín hiệu ra không được phản hồi về thiết bị điều khiển. Sơ đồ hình 1.2 là hệ thống điều khiển hở.



Hình 1.2: Hệ thống điều khiển hở

Cơ sở lý thuyết để nghiên cứu hệ thống hở là lý thuyết về relay và lý thuyết ô-tô-mát hữu hạn.

II. PHÂN LOẠI HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG

Hệ thống điều khiển có thể phân loại bằng nhiều cách khác nhau. Sau đây là một số phương pháp phân loại:

1. Hệ tuyến tính và phi tuyến:

Có thể nói hầu hết các hệ thống vật lý đều là hệ phi tuyến, có nghĩa là trong hệ thống có ít nhất một phần tử là phần tử phi tuyến (quan hệ vào ra là quan hệ phi tuyến). Tuy nhiên, nếu phạm vi thay đổi của các biến hệ thống không lớn, hệ thống có thể được tuyến tính hóa trong phạm vi biến thiên của các biến tương đối nhỏ. Đối với hệ tuyến tính, phương pháp xếp chồng có thể được áp dụng.

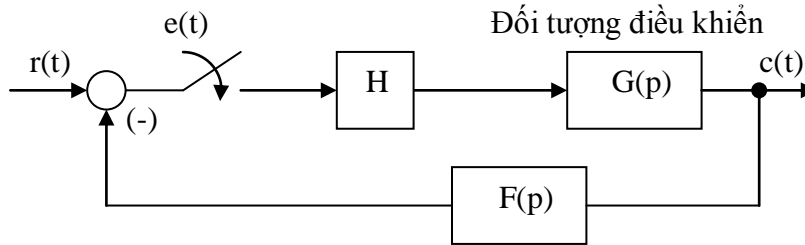
2. Hệ bất biến và biến thiên theo thời gian:

Hệ bất biến theo thời gian (hệ dừng) là hệ thống có các tham số không đổi (theo thời gian). Đáp ứng của các hệ này không phụ thuộc vào thời điểm mà tín hiệu vào được đặt vào hệ thống điều khiển phi tuyến không gian, với khối lượng giảm theo thời gian do tiêu thụ năng lượng trong khi bay.

3. Hệ liên tục và gián đoạn theo thời gian:

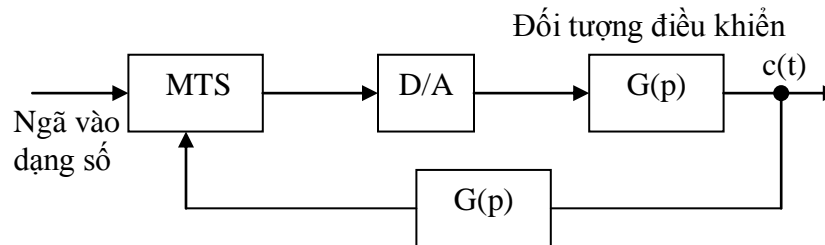
Trong hệ liên tục theo thời gian, tất cả các biến là hàm liên tục theo thời gian. Công cụ phân tích hệ thống liên tục là phép biến đổi Laplace hay Fourier. Trong khi đó, hệ gián đoạn là hệ thống có ít nhất một tín hiệu là hàm gián đoạn theo thời gian. Người ta phân biệt hệ thống gián đoạn gồm:

- Hệ thống xung: là hệ thống mà trong đó có một phần tử xung (khóa đóng ngắt) hay là tín hiệu được lấy mẫu (sample) và giữ (hold). (Hình 1.3)



Hình 1.3: Hệ thống điều khiển xung.

- Hệ thống số: là hệ thống gián đoạn trong đó tín hiệu được mã hóa thành logic 1, 0. Đó là các hệ thống có các khâu biến đổi tương tự / số (A/D), số/ tương tự (D/A) và để kết nối kết nối tín hiệu với máy tính số. (Hình 1.4)



Hình 1.4: Hệ thống điều khiển số

Công cụ để phân tích hệ thống gián đoạn là phép biến đổi Laplace, Fourier gián đoạn hay phép biến đổi Z.

4. Hệ đơn biến và đa biến:

Hệ đơn biến là hệ chỉ có một ngõ vào và một ngõ ra. Công cụ để phân tích và tổng hợp hệ đơn biến là lý thuyết điều khiển cổ điển. Ví dụ: hệ điều khiển định vị (vị trí).

Hệ đa biến là hệ có nhiều ngõ vào và nhiều ngõ ra. Công cụ để phân tích và tổng hợp hệ đa biến là lý thuyết điều khiển hiện đại dựa trên cơ sở biểu diễn hệ trong không gian trạng thái. Ví dụ: hệ điều khiển quá trình (Process Control System) có thể gồm có điều khiển nhiệt độ và áp suất.

5. Hệ thống thích nghi và hệ thống không thích nghi:

Hệ thống thích nghi là hệ thống hoạt động theo nguyên tắc tự chỉnh định, trong đó hệ thống tự phát hiện những thay đổi của các tham số do ảnh hưởng của môi trường bên ngoài và thực hiện việc điều chỉnh tham số để đạt được chỉ tiêu tối ưu được đề ra.

6. Hệ xác định (deterministic) và hệ ngẫu nhiên (stochastic):

Một hệ thống điều khiển là xác định khi đáp ứng đối với một ngõ vào nhất định có thể được biết trước (predictable) và có thể lặp lại được (repeatable). Nếu không thỏa mãn 2 điều kiện trên, hệ thống điều khiển là ngẫu nhiên.

III. NHIỆM VỤ CỦA LÝ THUYẾT ĐIỀU KHIỂN TỰ ĐỘNG

Để khảo sát và thiết kế một hệ thống điều khiển tự động người ta thực hiện các bước sau:

- a) Dựa trên các yêu cầu thực tiễn, các mô hình vật lý ta xây dựng mô hình toán học dựa trên các quy luật, hiện tượng, quan hệ của các đối tượng vật lý. Mô hình toán học của hệ thống được xây dựng từ các mô hình toán học của các phần tử riêng lẻ.
- b) Dựa trên lý thuyết ổn định, ta khảo sát tính ổn định của hệ thống. Nếu hệ thống không ổn định ta thay đổi đặc tính của hệ thống bằng cách đưa vào một khâu bù chính (compensation) hay thay đổi tham số của hệ để hệ thành ổn định.
- c) Khảo sát chất lượng của hệ theo các chỉ tiêu đề ra ban đầu. Nếu hệ không đạt chỉ tiêu chất lượng ban đầu, ta thực hiện bổ chính hệ thống.
- d) Mô phỏng hệ thống trên máy tính để kiểm tra lại thiết kế.
- e) Thực hiện mô hình mẫu (prototype) và kiểm tra thiết kế bằng thực nghiệm.
- f) Tinh chỉnh lại thiết kế để tối ưu hóa chỉ tiêu chất lượng và hạ thấp giá thành nếu có yêu cầu.
- g) Xây dựng hệ thống thực tế.