Tạp chí tin học và Điều khiển học, T.14, Số 2 (1998), (8-12)

ĐIỀU KHIỂN MỜ THÍCH NGHI CÁC HỆ THỐNG PHI TUYẾN THỜI GIAN RỜI RẠC

CHU VĂN HỶ[1]

Abstract: This paper present a method of designing an adaptive fuzzy controller for discrete - time nonlinear systems. The unknown nonlinear fuctions f(x), g(x) are modeled by fuzzy system $\hat{f}(x, \theta_f)$, $\hat{g}(x, \theta_g)$ with singleton fuzzifier, product inference engine, center average defuzzifier. The adaptation laws for the adjustable parameters θ_f, θ_g are obtained by using gradient descent algorithm or recusive least squares method.

I. MỞ ĐẦU

Trong khoảng 40 năm tồn tại, điều khiển thích nghi chủ yếu nghiên cứu các hệ thống tuyến tính [1]. Cho các hệ thống phi tuyến, một số kết quả quan trọng chỉ mới đạt được trong vòng 10 năm lại đây [2]. Đấy cũng là thời kỳ lí thuyết mờ bắt đầu được áp dụng vào điều khiển các hệ thống kĩ thuật [3]. Cũng như mạng nơron, hệ thống mờ được sử dụng cho điều khiển phi tuyến chính là nhờ khả năng xấp xỉ các hàm phi tuyến của chúng.

Trong bài này, chúng ta xét một lớp các đối tượng phi tuyến rời rạc 1 đầu vào 1 đầu ra, bậc tương đối (relative degree) bằng 1. Để xấp xỉ các hàm phi tuyến không biết f(x), g(x), ta xây dựng hệ thống mờ $\hat{f}(x, \theta_f)$, $\hat{g}(x, \theta_g)$. Bằng cách hiệu chỉnh các véctơ thông số θ_f, θ_g, có thể cực tiểu hoá sai số xấp xỉ và bộ điều khiển thích nghi với sự thay đổi động học của hệ thống. Ở đây rất khó áp dụng phương pháp hàm Liapunop như trong trường hợp hệ thống liên tục [3]. Cách tính của chúng tôi dựa vào nhận xét: nếu ta chọn θ_f, θ_g là véc tơ các tâm của tập mờ trong phần thì của hệ luật, thì có thể biểu diễn sai số xấp xỉ bằng một hàm tuyến tính đối với θ_f, θ_g. Do đó hoàn toàn có thể ước lượng θ_f, θ_g theo các phương pháp truyền thống, như: bình phương cực tiểu truy hồi, gọi gradient. Với $\hat{f}(x, \theta_f)$, $\hat{g}(x, \theta_g)$ đã tính được, ta thiết kế bộ điều khiển - gọi là tương đương vững (certainly equivalent controller) - như trong điều khiển tuyến tính.

II. THÀNH LẬP BÀI TOÁN

Xét đối tượng phi tuyến rời rạc 1 đầu vào 1 đầu ra:

$$y(k+1) = f(x) + g(x)u(k) \tag{1}$$

Trong đó: u(k) là điều khiển, y(k) là đầu ra, x kí hiệu véc tơ hồi qui:

$$x_{n+1} = [x_1, x_2, \ldots, x_n; x_{n+1}, \ldots, x_{n+m}]^T$$

$$:= [y(k-n+1), y(k-n+2), \ldots, y(k), u(k-m+1), u(k-m+2), \ldots, u(k-1)]^T;\ m \le n \tag{2}$$

Cho mô tả các hàm phi tuyến f(x), g(x) không biết, ta có một số luật mờ như sau:

$$L_f^r\text{: NẾU } x_1 \text{ là } A_1^r \text{ và } \ldots\ x_{n+m} \text{ là } A_{n+m}^r\text{, THÌ } f(x) \text{ là } E^r,\ r = 1, 2 \ldots N_f \tag{3}$$

N_g^s: NẾU x_1 là B_1^s và ... x_{n+m} là B_{n+m}^s, THÌ $g(x)$ là H^s, $s = 1, 2 \dots N_g$ (4)

Để đơn giản, ta xét trường hợp: mờ hoá đơn điệu, động cơ suy diễn tích, rõ hoá trung bình tâm (singleton fuzzifier, product inference engine, center average defuzzifier). Cần tìm điều khiển thícn nghi u(k) sao cho đầu ra của hệ thống y(k) bám sát đầu ra của mô hình mẫu $y_m(k)$.

III. PHƯƠNG PHÁP GIẢI

Nếu biết các hàm số f(x), g(x) ta có thể chọn điều khiển u(k) để khử tính phi tuyến - tức là để hệ thống kín được mô tả bằng một phương trình tuyến tính - như sau:

$$u^*(k) = [y_m(k+1) + c_1 e_t(k) - f(x)]/\,g(x) \qquad (5)$$

Trong đó: $e_t(k)$ là sai số bám

$$e_t(k) = y_m(k) - y(k) \qquad (6)$$

Thật vậy, thay (5) vào (1) ta nhận được phương trình tuyến tính đặc trưng cho hệ thống kín.

$$e_t(k+1) + c_1 e_t(k) = 0 \qquad (7)$$

Do đó có thể áp dụng các phương pháp điều khiển tuyến tính quen biết. Ví dụ: theo phương pháp đặc cực, để hệ thống bám tiệm cận cần chọn $/c_1/ < 1$.

Nhưng, trong thực tế f(x), g(x) là không biết. Ta có thể thay thế bằng các hệ thống mờ $\hat{f}(x)$, $\hat{g}(x)$ dựa trên cơ sở hệ luật (3), (4). Thông thường số luật này tương đối ít, nên $\hat{f}(x)$ và $\hat{g}(x)$ chỉ là xấp xỉ thô của f(x), g(x). Để tăng độ chính xác, ta sẽ xây dựng 2 hệ thống mờ $\hat{f}(x, \theta_f)$, $\hat{g}(x, \theta_g)$ với những bổ sung sau [3]:

- Tăng số tập mờ mô tả các biến số x_i, $i = 1, 2 \dots n + m$
- Chọn các vét tơ thông số ... θ_f, θ_g thích hợp để dễ dàng hiệu chỉnh theo động học của hệ thống.

Cho biến số x_i, $i = 1, 2 \dots n + m$, ta định nghĩa p_i tập mờ C_i^{ji}, $ji = 1, 2 \dots p_i$, bao gồm A_i^r, $r = 1, 2, \dots N_f$ trong (3); và định nghĩa q_i tập mờ D_i^{ki}, $ki = 1, 2 \dots q_i$, bao gồm B_i^s, $s = 1, 2 \dots N_g$ trong (4). Như vậy ta có $\prod_{i=1} p_i$ luật cho $\hat{f}(x, \theta_f)$ và $\prod_{i=1} q_i$ luật cho $\hat{g}(x, \theta_g)$:

NẾU x_1 là C_1^{j1} và ... x_{n+m} là C_{n+m}^{jn+m}, THÌ $\hat{f}(x, \theta_f)$ là $P^{j1, \dots jn+m}$, $ji = 1, 2 \dots p_i$, $i = 1, 2, \dots n + m$ (8)

NẾU x_1 là D_1^{k1} và ... x_{n+m} là D_{n+m}^{kn+m}, THÌ $\hat{g}(x, \theta_g)$ là $Q^{k1, \dots kn+m}$, $ki = 1, 2, \dots q_i$, $i = 1, 2 \dots n + m$ (9)

Với phương pháp mờ hoá đơn điệu, động cơ suy diễn tích, rõ hoá trung bình tâm, ta nhận được:

$$\hat{f}(x, \theta_f) = [\sum_{j1=1} \dots \sum_{jn+m=1} y_f \quad (\prod_{i=1} \mu C_i\ (x_i))]/[\sum_{j1=1} \dots \sum_{jn+m=1} (\prod_{i=1} \mu C_i\ (x_i))] \tag{10}$$

$$\hat{g}(x, \theta_g) = [\sum_{k1=1} \dots \sum_{kn+m=1} y_f^{k1 \dots kn+m} (\prod_{i=1} \mu D_i^{ki} (x_i))]/[\sum_{k1=1} \dots \sum_{kn+m=1} (\prod_{i=1} \mu D_i^{ki} (x_i))] \tag{11}$$

Ta thấy: thích hợp nhất, nếu chọn θ_f (và θ_g) là vét tơ với phần tử $y_f^{j1 \dots jn+m}$ (và $y_g^{k1, \dots kn+m}$) - cô động là tâm của các tập mờ $P^{ji, \dots jn+m}$ (và $Q^{k1, \dots kn+m}$). Sau đó có thể viết (10), (11) dưới dạng cô động:

$$\hat{f}(x, \theta_f) = \xi^T(x)\theta_f \tag{12}$$

$$\hat{g}(x, \theta_g) = \eta^T(x)\theta_g \tag{13}$$

Trong đó: $\xi(x)$ là véc tơ gồm $\prod_{i=1} p_i$ phần tử:

$$\xi_{ji, \dots jn+m}(x) = [\prod_{i=1} \mu C_i^{ji} (x_i)]/[\sum_{j1=1} \dots \sum_{jn+m=1} (\prod_{i=1} \mu C_i^{ji} (x_i))] \tag{14}$$

và $\eta(x)$ là véc tơ bao gồm $\prod_{i=1} q_i$ phần tử:

$$\eta_{ki, \dots kn+m}(x) = [\prod_{i=1} \mu D_i^{ki} (x_i))]/[\sum_{k1=1} \dots \sum_{kn+m=1} (\prod_{i=1} \mu D_i^{ki} (x_i))] \tag{15}$$

Véc tơ giá trị ban đầu $\theta_f(0)$ (và $\theta_g(0)$) được xác định như sau: nếu phần NẾU của (8) (và (9)) phù hợp với phần NẾU của (3) (và (4)), thì ta lấy $P^{j1, \dots jn+m} = E^r$ (và $Q^{ki, \dots kn+m} = H^s$); từ đó xác định được các tâm $y_f^{j1, \dots jn+m}(0)$ (và $y_g^{k1, \dots kn+m}(0)$) tương ứng. Những tâm còn lại được chọn tuỳ ý.

Các hệ thống mờ ở đây là xấp xỉ vạn năng (universal approximator): có khả năng xấp xỉ các hàm phi tuyến bất kỳ, và bằng cách tăng số luật mờ có thể đạt độ chính xác tùy ý. Ta có thể thay $f(x) = \hat{f}(x, \theta_f)$ và $g(x) = \hat{g}(x, \theta_g)$ vào (5) và nhận được bộ điều khiển mờ - gọi là tương đương vững (certainty equivalent controller):

$$u(k) = [y_m(k+1) + c_1 e_r(k) - \hat{f}(x, \theta_f)] / \hat{g}(x, \theta_g) \tag{16}$$

Bây giờ ta tìm luật chỉnh các véctơ θ_f, θ_g để xấp xỉ là tốt nhất theo một số tiêu chuẩn. Ta thấy rất khó áp dụng phương pháp hàm Liapunov như trong trường hợp hệ thống liên tục [3]. Ta thành lập đầu ra xấp xỉ:

$$\hat{y}(k+1) = \hat{f}(x, \theta_f(k)) + \hat{g}(x, \theta_g(k))u(k) \tag{17}$$

Theo (12), (13), có thể biểu diễn $\hat{y}(k+1)$ dưới dạng:

$$\hat{y}(k+1) = \xi^T(x)\theta_f(k) + \eta^T(x)u(k)\theta_g(k) = \varphi^T(k)\theta(k) \tag{18}$$

Trong đó:

$$\varphi^T(k) = [\xi^T(x), \eta^T(x)u(k)]^T \tag{19}$$

$$\theta(k) = (\theta_f^T(k), \theta_g^T(k)]^T \tag{20}$$

Ta nhận thấy: ở thời điểm cắt mẫu thứ k + 1, các véc tơ x, $\xi(x)$, $\eta(x)$ và điều khiển u(k) đã biết, nêu $\varphi(k)$ đã xác định $\hat{y}(k+1)$ là hàm tuyến tính đối với các véc tơ $\theta(k)$. Do đó, có thể ước lượng $\theta(k)$ theo các phương pháp trong điều khiển thích nghi tuyến tính.

Phương pháp gradient:

Ta kí hiệu xấp xỉ sai số của đầu ra:

$$e_a(k+1) = y(k+1) - \hat{y}(k+1) \qquad (21)$$

Trong đó $\hat{y}(k+1)$ được tạo ra nhờ các hệ thống mờ theo (18), còn $y(k+1)$ có thể tính từ đầu ra yêu cầu và sai số bám theo (6), (7)

$$y(k+1) = y_m(k+1) + c_1e_t(k) \qquad (22)$$

Định nghĩa tiêu chuẩn:

$$J_1 = e_a^2(k+1)/2 \qquad (23)$$

Ta có luật hiệu chỉnh các véc tơ $\theta_f(k)$, $\theta_g(k)$ như sau:

$$\theta_f(k+1) = \theta_f(k) - \alpha_1(\partial J_1/\partial\theta_f(k)) \qquad (24)$$

$$\theta_g(k+1) = \theta_g(k) - \alpha_2(\partial J_1/\partial\theta_g(k)) \qquad (25)$$

Thay (23), (22), (21), (18), vào (24), (25), sau một vài biến đổi ta nhận được kết quả:

$$\theta_f(k+1) = \theta_f(k) + \alpha_1(y_m(k+1) + c_1e_t(k) - \hat{y}(k+1))\,\xi(x) \qquad (26)$$

$$\theta_g(k+1) = \theta_g(k) + \alpha_2(y_m(k+1) + c_1e_t(k) - \hat{y}(k+1))u(k)\eta(x) \qquad (27)$$

Ta thấy: nếu $\theta_f(k)$ và $\theta_g(k)$ hội tụ đến θ_f^* và θ_g^* khi $k \to \infty$, thì $\partial J_1/\partial\theta_f = 0$ và $\partial J_1/\partial\theta_g = 0$. Điều đó có nghĩa: θ_f^*, θ_g^* là cực tiểu cục bộ của J_1. Vậy, theo phương pháp gradient ta có thể hiệu chỉnh các véc tơ θ_f, θ_g để cực tiểu hoá bình phương sai số xấp xỉ đầu ra tại từng thời điểm cắt mẫu. Ưu điểm của phương pháp này là: khá đơn giản cho lập trình và có hiệu quả trong nhiều trường hợp. Nhược điểm chính là: không đảm bảo đạt được cực tiểu toàn cục như phương pháp bình phương cực tiểu; rất nhậy cảm với việc chọn các giá trị ban đầu $\theta_f(0)$, $\theta_g(0)$ và độ dài bước α - trong một số trường hợp thậm chí có thể ngăn cản sự hội tụ của thuật toán. Để khắc phục, người ta đã phát triển kỹ thuật xác định độ dài bước tự động, tăng tốc độ hội tụ [4]....
Phương pháp bình phương cực tiểu truy hồi

Giúp ta ước lượng véctơ thông sô $\theta(k)$ để cực tiểu hoá bình phương sai số xấp xỉ đầu ra:

$$J_2 = \prod_{i=1} e_a^2(i) \qquad (28)$$

Ta có thể dẫn ra công thức truy hồi:

$$\theta(k+1) = \theta(k) + P(k)\varphi(k)[1 + \varphi^T(k)P(k)]^{-1}[y(k+1) - \varphi^T(k)\theta(k)] \qquad (29)$$

$$P(k+1) = P(k) - P(k)\varphi(k)[1 + \varphi^T(k)P(k)\varphi(k)]^{-1}\varphi^T(k)P(k) \qquad (30)$$

Trong đó: giá trị ban đầu $\theta(0)$ được xác định như trên, và có thể chọn: $P(0) = \beta I$, β là một số rất lớn. Bên cạnh lời giải cơ bản (29), (30), ta còn thấy một số lời giải tương đương khác (như phương trình ổn định hoá, thừa số hoá....), và những cải tiến nhằm tăng tính hội tụ của thuật toán, hoặc tính bền vững của hệ thống khi một khi có sự thay đổi thông số hoặc chịu tác động của nhiễu.

KẾT LUẬN

Trên đây đã trình bày phương pháp cơ bản cho thiết kế bộ điều khiển mờ thích nghi phi tuyến - một lĩnh vực còn rất ít kết quả và các phương pháp truyền thống đang bộc lộ nhiều yếu điểm. Bộ điều khiển ở đây có 2 tính chất quan trọng của hệ thống thông minh là: khả năng thích nghi và suy luận (kiểu Nếu - Thì). Phương pháp ở trên có thể áp dụng cho các hệ thống phi tuyến tổng quát có bậc tương đối lớn hơn 1. Một số vấn đề cần nghiên cứu tiếp là [3]:

- Xác định miền giới hạn cho các vectơ θ_f, θ_g: bởi vì θ_f rất lớn - nhất là khi thuật toán không hội tụ $\theta_f \to \infty$, hoặc θ_g rất nhỏ ($\theta_g \approx 0$), dẫn đến điều khiển u(k) quá lớn, không thể thực hiện được. Tình hình xảy ra tương tự như trong các hệ thống tuyến tính, và ta có thể giải quyết bằng áp dụng các kĩ thuật: chiếu thông số, miền chết, σ - biến dạng, ε - biến dạng.
- Đảm bảo ổn định cho hệ thống: ngoài phương pháp thông thường (xác định cấu trúc và thông số của bộ điều khiển), ta có thể sử dụng các bộ điều khiển giám sát để can thiệp vào hệ thống khi có xu hướng mất ổn định.

Công trình này được thực hiện trong khuôn khổ nghiên cứu của đề tài cấp nhà nước KHCN - 04-09 thuộc chương trình tự động hoá.

TÀI LIỆU THAM KHẢO:

1. **Krstic M, Kanellakopoulos, I, Kokotovic P**. *Nonlinear and Adaptive Control Design*, John Wiley, New York, 1995.

2. **Chen F. C, Knalil H. K**: *Adaptive control of a class of nonliear discrete-time systems using neural networks*, IEEE Trans, Automat, Contr, 40, No. 5(1995), 791 - 801.

3. **Wang L. X**: *A Course in Fuzzy Systems and Control*, Prentice - Hall International 1997.

4. **Legras J**: *Algorithms et programmes d optinisation nonlinéaire avec contraintes*, Masson, Paris, 1980.