

MẠNG NƠ RON TRUYỀN THĂNG CHO ĐIỀU KHIỂN THÍCH NGHI CÁC HỆ THỐNG PHI TUYẾN

CHU VĂN HỠ

Abstract. Multilayer feedforward artificial neural networks are used in nonlinear adaptive control problems. In case of a feedback-linearizable plant, two networks are needed to model the unknown system and generate the feedback control. In general case, an inverse model adaptive control strategy is developed and analysed.

1. MỞ ĐẦU

Điều khiển thích nghi các hệ thống phi tuyến là một vấn đề khó; song nếu giải quyết được sẽ có ý nghĩa khoa học và thực tiễn rất lớn. Bởi vì hầu như các hệ thống thực đều là phi tuyến. Bấy nay, do thiếu các phương pháp có hiệu quả, đặc biệt chưa có những phương tiện kỹ thuật thích hợp để thực hiện các luật điều khiển phức tạp, nên ta đã đơn giản hóa xét chúng như những hệ thống tuyến tính. Trong điều khiển thích nghi tuyến tính hiện tồn tại một số vấn đề sau:

- Vấn đề cấu trúc của mô hình đối tượng: lý thuyết điều khiển bền vững đã cơ bản giải quyết được sự bất định thông số của hệ thống, nhưng bất định về cấu trúc, nhất là trường hợp đối tượng có cấu trúc biến đổi hoặc hoàn toàn không biết vẫn còn nan giải.

- Vấn đề thời gian thực.

Các nhà điều khiển học tìm thấy ở mạng nơ ron nhân tạo một phương tiện mới để khắc phục các trở ngại trên. Người ta đã chứng minh được rằng (Cybenko, 1988 [1]; Funahashi, 1989 [2]; Hornik, 1989 [3]; Hecht-Nielson, 1989 [4]...): mạng nơ ron truyền thăng có một hoặc nhiều lớp ẩn với hàm hoạt động $a(\cdot)$ (activation function, non-dynamic nonlinear function) có dạng bẹp (squashing), và hàm tổng trọng (weighted summer, integration function) tuyến tính hoặc đa thức - là công cụ xấp xỉ vạn năng: có thể xấp xỉ các hàm phi tuyến bất kỳ với tốc độ chính xác tùy ý. Do đó, ngay trong trường hợp khó nhất - khi không biết cấu trúc của mô hình đối tượng - ta vẫn có thể tìm được điều khiển thích nghi nhờ phương pháp rất độc đáo: sử dụng mô hình ngược của đối tượng (inverse model, inverse system). Một ưu điểm nữa của mạng nơ ron là: khả năng xử lý phân tán song song. Nhờ những vi mạch thông minh chứa số lượng lớn các nơ ron nhân tạo ở rải rác trong hệ thống, đồng thời xử lý tín hiệu, thì vấn đề thời gian thực sẽ được khắc phục. Với sự ra đời của mạng nơ ron mờ, vấn đề bất định thông số hệ thống cũng được giải quyết khá dễ dàng.

Trong bài này, chúng tôi trình bày các phương pháp điều khiển thích nghi phi tuyến nhờ mạng nơ ron truyền thăng. Trong trường hợp mô hình đối tượng có dạng: $y(k+d) = F(z(k)) + G(z(k))u(k)$ ta vẫn sử dụng phương pháp điều khiển phản hồi tuyến tính hóa truyền thống [5] nhưng dùng 2 mạng nơ ron để xấp xỉ các hàm phi tuyến không biết $F(z(k))$, $G(z(k))$. Trong trường hợp tổng quát: $y(k+1) = f(x(k), u(k))$, nhờ mô hình ngược

của đối tượng xây dựng bằng mạng nơ ron ta có thể tìm được điều khiển thích nghi $u(k)$ [6].

2. ĐIỀU KHIỂN PHẢN HỒI TUYẾN TÍNH HÓA

Xét đối tượng điều khiển 1 đầu vào 1 đầu ra có trễ:

$$y(k+1) = f_0(x(k)) + g_0(x(k)) u(k-d+1). \quad (1)$$

Trong đó: u là điều khiển; y là đầu ra; $f_0(\cdot)$, $g_0(\cdot)$ là các hàm phi tuyến không biết; $x(k)$ kí hiệu véctor hồi qui:

$$\begin{aligned} x(k) &= [x_1(k), \dots, x_n(k), x_{n+1}(k), \dots, x_{n+m}(k)]^T \\ &= [y(k-n+1), \dots, y(k), u(k-m-d+1), \dots, u(k-d)]^T, \quad m \leq n, \end{aligned} \quad (2)$$

d là trễ tổng quát:

$$d = r + L; \quad r \geq 1, \quad L \geq 0. \quad (3)$$

r là bậc tương đối, L là bội số của trễ điều khiển so với chu kỳ cắt mẫu. Nhớ rằng $u(k-d+1)$ trong (1) là điều khiển trễ đã được sinh ra ở thời điểm cắt mẫu thứ $(k-d+1)$ trong quá khứ. Để biểu diễn quan hệ giữa đầu ra với điều khiển $u(k)$ ở thời điểm k đang xét, ta biến đổi như sau:

$$\begin{aligned} y(k+2) &= f_0(y(k-n+2), \dots, y(k+1), u(k-m-d+2), \dots, u(k-d+1)) + \\ &g_0(y(k-n+2), \dots, y(k+1), u(k-m-d+2), \dots, u(k-d+1)) u(k-d+2). \end{aligned}$$

Thay $y(k+1)$ theo (1), ta có

$$\begin{aligned} y(k+2) &= f_1(y(k-n+1), \dots, y(k), u(k-m-d+1), \dots, u(k-d+1)) + \\ &g_1(y(k-n+1), \dots, y(k), u(k-m-d+1), \dots, u(k-d+1)) u(k-d+2). \end{aligned}$$

Thực hiện $d-1$ lần, ta nhận được

$$y(k+d) = F(z(k)) + G(z(k)) u(k). \quad (4)$$

Trong đó: $F(\cdot)$, $G(\cdot)$ là các hàm phi tuyến phức tạp; $z(k)$ là véctor gồm $N = n + m + d - 1$ phần tử

$$\begin{aligned} z(k) &= [z_1(k), \dots, z_n(k), z_{n+1}(k), \dots, z_N(k)]^T \\ &= [y(k-n+1), \dots, y(k), u(k-m-d+1), \dots, u(k-1)]^T, \end{aligned} \quad (5)$$

Bộ điều khiển được xây dựng theo nguyên lí tương đương vững (Certainty Equivalence Principle) như sau. Ở bước nhận dạng, các hàm phi tuyến không biết $F(z(k))$, $G(z(k))$ được xấp xỉ bằng 2 mạng nơ ron $\hat{F}(z(k), w(k))$, $\hat{G}(z(k), v(k))$. Các véctor trọng $w(k)$, $v(k)$

được tính theo phương pháp xây dựng trên cơ sở thuật toán lan truyền ngược. Ở bước điều khiển, để khử tính phi tuyến của hệ thống có thể tính

$$u(k) = \frac{y_m(k+d) - \hat{F}(z(k), w(k))}{\hat{G}(z(k), v(k))}, \quad (6a)$$

trong đó $y_m(k+d)$ là đầu ra yêu cầu ở thời điểm cắt mẫu thứ $k+d$ trong tương lai. Cho các hệ thống thông thường không trễ $L=0$, có bậc tương đối $r=1$, ta có $d=1$ và có thể thiết kế hệ thống bám như sau

$$u(k) = \frac{y_m(k+1) + [c_1 e_t(k) + c_2 e_t(k-1) + \dots + c_p e_t(k-p+1)] - \hat{F}(z(k), w(k))}{\hat{G}(z(k), v(k))}, \quad (6b)$$

trong đó biểu thức trong dấu [] chính là sai số bám $e_t(k+1)$ với dấu ngược lại:

$$e_t(k+1) = y_m(k+1) - y(k+1). \quad (7)$$

Phương trình đặc trưng của hệ thống kín sẽ là

$$e_t(k+1) + c_1 e_t(k) + c_2 e_t(k-1) + \dots + c_p e_t(k-p+1) = 0, \quad p \geq r. \quad (8)$$

Bậc và các hệ số được xác định để hệ thống bám có các tính chất mong muốn, ví dụ theo phương pháp đặt cực.

Bây giờ ta xây dựng các mạng nơ ron $\hat{F}(z(k), w(k))$, $\hat{G}(z(k), v(k))$ (với lớp vào gồm $N = n + m + d - 1$ nơ ron, lớp ra 1 nơ ron). Các công trình [1] ÷ [4] mới chủ yếu chứng minh sự tồn tại, còn về cấu trúc của mạng: số lớp, số nơ ron trong mỗi lớp ẩn - thì chưa đưa ra được phương pháp xác định cụ thể. Mặc dù lí thuyết đã chỉ ra: chỉ cần mạng 3 lớp là đủ, nhưng trong thực tế vẫn phải dùng 4, 5 lớp hoặc nhiều hơn - tùy theo độ phi tuyến của hệ thống [7], bởi vì xấp xỉ bằng mạng 3 lớp thường đòi hỏi số lượng nơ ron quá lớn; mặt khác là không đủ cho ổn định hóa, nhất là trong xấp xỉ không liên tục. Các kết quả thực nghiệm (Chester, 1990...) cho thấy: mạng với 2 lớp ẩn cho độ chính xác cao hơn, tính tổng quát hóa tốt hơn và cần số nơ ron ít hơn so với mạng 1 lớp ẩn. Có thể nói (Sontag, 1992; Chen, 1994...): mạng 4 lớp với 2 lớp ẩn phi tuyến là thích hợp cho nhiều ứng dụng. Phân tích chính xác số nơ ron cần thiết trong từng lớp ẩn là khá khó do tính phức tạp của mạng, nên thường xác định bằng thực nghiệm.

Đầu ra $\hat{F}(z(k), w(k))$, $\hat{G}(z(k), v(k))$ của các mạng trên cần cho tính điều khiển $u(k)$ theo công thức (6), đồng thời dùng để thành lập đầu ra xấp xỉ dự báo:

$$\hat{y}(k+d) = \hat{F}(z(k), w(k)) + \hat{G}(z(k), v(k)) u(k). \quad (9)$$

Ta có sai số xấp xỉ - là hàm phi tuyến đối với các véctơ trọng $w(k)$, $v(k)$:

$$e_a(k+d) = y(k+d) - \hat{y}(k+d). \quad (10)$$

Ta định nghĩa hàm mục tiêu

$$J = e_a^2(k+d)/2. \quad (11)$$

Các trọng của mạng được cập nhật theo phương pháp gradient nhằm cực tiểu hóa hàm mục tiêu J :

$$w(k+1) = w(k) - \alpha_1 \frac{\partial J}{\partial w(k)}, \quad (12a)$$

$$v(k+1) = v(k) - \alpha_2 \frac{\partial J}{\partial v(k)}. \quad (12b)$$

Trong đó α_1, α_2 là các hằng học (learning constant) được chọn để đáp ứng các yêu cầu về tính hội tụ cũng như thời gian học. Theo (11), (10), (9) ta tính các đạo hàm riêng trong (12), và coi đầu ra trong tương lai $y(k+d)$ gần bằng đầu ra yêu cầu $y_m(k+d)$ có sẵn, ta nhận được

$$w(k+1) \approx w(k) + \alpha_1 [y_m(k+d) - \hat{y}(k+d)] \frac{\partial \hat{F}(z(k), w(k))}{\partial w(k)}, \quad (13a)$$

$$v(k+1) \approx v(k) + \alpha_2 [y_m(k+d) - \hat{y}(k+d)] u(k) \frac{\partial \hat{G}(z(k), v(k))}{\partial v(k)}. \quad (13b)$$

Cho trường hợp $d=1$, thay $y(k+1)$ theo (7), (8) ta có công thức chính xác

$$w(k+1) = w(k) + \alpha_1 [y_m(k+1) + c_1 e_t(k) + c_2 e_t(k-1) + \dots + c_p e_t(k-p+1) - \hat{y}(k+1)] \frac{\partial \hat{F}(z(k), w(k))}{\partial w(k)}, \quad (14a)$$

$$v(k+1) = v(k) + \alpha_2 [y_m(k+1) + c_1 e_t(k) + c_2 e_t(k-1) + \dots + c_p e_t(k-p+1) - \hat{y}(k+1)] u(k) \frac{\partial \hat{G}(z(k), v(k))}{\partial v(k)}. \quad (14b)$$

Việc còn lại là: tính các đạo hàm riêng đầu ra của mạng theo các trọng trong (13), (14) sẽ được thực hiện theo phương pháp của thuật toán lan truyền ngược [8]. Để đơn giản, ta lấy ví dụ mạng có lớp vào gồm N nơ ron, 1 lớp ẩn gồm M nơ ron, lớp ra 1 nơ ron, và bỏ qua biến số thời gian trong các kí hiệu dưới đây. Hàm tổng trọng của các nơ ron trong lớp ẩn là

$${}^h S_j = \sum_{i=1}^N {}^h w_{ji} z_i, \quad j = 1, 2, \dots, M. \quad (15)$$

Đầu ra của các nơ ron ẩn sẽ là

$${}^h O_j = a({}^h S_j). \quad (16)$$

Tổng trọng của nơ ron lớp ra

$${}^0 S_1 = \sum_{j=1}^M {}^0 w_{1j} {}^h O_j. \quad (17)$$

Ta có đầu ra của mạng

$$\hat{F}(z, w) = a({}^0 S_1) = a\left(\sum_{j=1}^M {}^0 w_{1j} a\left(\sum_{i=1}^N {}^h w_{ji} z_i\right)\right). \quad (18)$$

Từ đó tính được: đạo hàm đầu ra của mạng theo các trọng của lớp ra

$$\frac{\partial \hat{F}(z, w)}{\partial w_{1j}} = \dot{a}({}^0S_1)^h O_j, \quad j = 1, 2, \dots, M \quad (19a)$$

và theo các trọng của lớp ẩn

$$\frac{\partial \hat{F}(z, w)}{\partial w_{ji}} = \dot{a}({}^0S_1)^0 w_{1j} \dot{a}({}^hS_j) z_i; \quad j = 1, 2, \dots, M, \quad i = 1, 2, \dots, N. \quad (19b)$$

Cho hàm hoạt động là một hàm sigmoid điển hình

$$a(S) = \tanh(S) = \frac{e^S - e^{-S}}{e^S + e^{-S}}, \quad (20)$$

ta có đạo hàm

$$\dot{a}(S) = 1 - S^2. \quad (21)$$

Các công thức trên đây dễ dàng tổng quát hóa cho trường hợp mạng có nhiều lớp ẩn hoặc nhiều đầu ra. Thuật toán học lan truyền ngược là một tron những phát triển quan trọng nhất trong lịch sử mạng nơ ron nhân tạo (Brayson, Ho, 1969; Werbos, 1974; Le Cun, 1985; Parker, 1985; Rumelhart, ..., 1986). Nhờ nó mạng truyền thẳng nhiều lớp đã trở thành cấu trúc rất được ưa chuộng cho các ứng dụng trong nhiều lĩnh vực như: nhận dạng, điều khiển, xử lí tín hiệu, phân lớp các mẫu.... Mạng truyền thẳng cùng với thuật toán học lan truyền ngược gọi là mạng lan truyền ngược, đã được chế tạo thành các vi mạch rất thuận tiện cho ứng dụng.

3. ĐIỀU KHIỂN SỬ DỤNG MÔ HÌNH NGƯỢC

Nếu mô hình (1) không phù hợp, ta cần mô tả đối tượng dưới dạng tổng quát - cho trường hợp không trễ, bậc tương đối bằng 1:

$$y(k+1) = f(x(k), u(k)), \quad (22)$$

trong đó $f(\cdot)$ là hàm phi tuyến không biết. Các phương pháp điều khiển thích nghi phi tuyến truyền thống đều chưa đề cập đến trường hợp tổng quát này. Ý tưởng ở đây cũng thực đơn giản, như sau [6]. Chắc chắn giữa $u(k)$ và $y(k+1)$, $x(k)$ tồn tại quan hệ hàm số

$$u(k) = f^{-1}(x(k), y(k+1)). \quad (23)$$

Cho các đối tượng trong thực tế, có thể giả thiết: $f(\cdot)$, $f^{-1}(\cdot)$ là các hàm đơn trị. Vậy hoàn toàn có thể xấp xỉ hàm phi tuyến $f^{-1}(\cdot)$ bằng một mạng nơ ron truyền thẳng N1. Lúc đó, nếu đầu vào của mạng là $x(k)$, $y(k+1)$, thì đầu ra sẽ là $u(k)$ phải tìm:

$$N1 \quad u(k) = \hat{f}^{-1}(x(k), y(k+1), w(k)), \quad (24)$$

trong đó $w(k)$ là véc tơ trọng. Để lập hàm sai số xấp xỉ $\epsilon_a(k)$, khác với trường hợp trên kia - ở đây không tồn tại giá trị yêu cầu cho đầu ra của mạng $u(k)$. Ngược lại, bài toán

đặt ra là: phải tìm điều khiển $u(k)$ để đầu ra của hệ thống $y(k+1)$ bám sát giá trị yêu cầu $y_m(k+1)$. Nên, có thể viết

$$e_a(k) = u_m(k) - u(k). \quad (25)$$

Trong đó $u_m(k)$ là đầu ra của mạng nơ ron N2 - giống hệt mạng N1 nhưng có đầu vào là $x(k)$, $y_m(k+1)$

$$N2 \quad u_m(k) = \hat{f}^{-1}(x(k), y_m(k+1), w(k)). \quad (26)$$

Ta đã có thể căn cứ vào $e_a(k)$ để cập nhật $w(k)$. Ta định nghĩa hàm mục tiêu

$$J = e_a^2(k)/2. \quad (27)$$

Trong thực tế ta sẽ dùng mạng N1 để ước lượng véctor trọng $w(k)$ và dùng mạng N2 làm bộ điều khiển [9]. Ta có thể cập nhật theo phương pháp gradient nhằm cực tiểu hóa hàm J :

$$\begin{aligned} w(k+1) &= w(k) - \alpha \frac{\partial J}{\partial w(k)} \\ &= w(k) + \alpha e_a(k) \frac{\partial \hat{f}^{-1}(x(k), y(k+1), w(k))}{\partial w(k)}. \end{aligned} \quad (28)$$

Vấn đề xác định cấu trúc của mạng và cách tính công thức (28) đã được trình bày ở trên. Quá trình điều khiển xảy ra như sau. Ở thời điểm bắt đầu $k=0$: sau khi được luyện N1, N2 có véctor trọng $w(0)$; bộ điều khiển N2 nhận các đầu vào $x(0)$, $y_m(1)$ và sinh ra điều khiển $u_m(0)$ - tác động vào đối tượng. Ở thời điểm cắt mẫu thứ $k=1$: đầu ra của hệ thống $y(1)$ được đo; từ các đầu vào $x(0)$, $y(1)$ mạng N1 tính được $u(0)$; so sánh với $u_m(0)$ để tính sai số $e_a(0)$; tính véctor trọng mới $w(1)$; mạng N2 sao lại véctor trọng $w(1)$ và từ đầu vào $x(1)$, $y_m(2)$ sinh ra điều khiển $u_m(1)$ - tác động vào đối tượng v.v...

Một số sơ đồ điều khiển khác có thể xem [6, 7].

4. KẾT LUẬN

Với phương tiện mới là mạng nơ ron truyền thẳng nhiều lớp, ta có thể giải bài toán điều khiển thích nghi cho tất cả các hệ thống phi tuyến, trong khi các phương pháp truyền thống chỉ có hiệu quả cho một số lớp đối tượng nhất định. Nhờ tính chất xấp xỉ vạn năng của mạng nơ ron truyền thẳng, vấn đề không phù hợp về cấu trúc giữa mô hình và đối tượng thực - do nhận dạng không chính xác hoặc bất định của hệ thống, từng gây rất nhiều trở ngại trong điều khiển tuyến tính - đã được khắc phục. Vấn đề bất định thông số sẽ được giải quyết nhờ mạng nơ ron mờ, hoặc sử dụng kỹ thuật miền chết (Dead-Zone) và các kỹ thuật khác [9].... Tuy nhiên, phương pháp gradient ở đây có một số nhược điểm như: vấn đề điểm cực tiểu cục bộ, hội tụ tương đối chậm, dao động phân kì khi chọn hằng học α lớn.... Nên gần đây người ta chú ý đến mạng nơ ron RBF (Radial Basis Function) là một loại mạng truyền thẳng đặc biệt với 1 lớp ẩn, do có ưu điểm nổi bật là: sai số xấp xỉ là hàm tuyến tính đối với véctor trọng, do đó có thể sử dụng luật cập nhật theo phương pháp bình phương cực tiểu truy hồi với nhiều tính chất ưu việt.

TÀI LIỆU THAM KHẢO

- [1] Cybenko G., Approximation by supositions of a sigmoidal function, Dept. Elec. Computer Eng., Univ. Illinois at Urbana-Champaign, Tech. Rep. 856, 1988.
- [2] Hornik K., Stinchcombe M., White H., Multilayer feedforward networks are universal approximators, *Neural Networks* 2 (1989) 359-366.
- [3] Funahashi K., On the approximate realization of continuous mappings by neural networks, *Neural Networks* 2 (1989) 183-192.
- [4] Hecht-Nielson R., Theory of the back-propagation neural network, Proc. Int. Joint Conf. Neural Networks, June 1989, 593-605.
- [5] Krstic M., Kanellakopoulos I., Kokotovic P., *Nonlinear and Adaptive Control Design*, John Wiley, New York, 1995.
- [6] Hunt K. J., Sbarbabo D., Zbikowski R., Gawthrop P. J., Neural networks for control systems, *Automatica* 28 (1992) 1083-1112.
- [7] Chin-Teng Lin, George Lee C. S., *Neural Fuzzy System*, Prentice-Hall International, 1996.
- [8] Rumelhart D., Hinton G. E., Williams R. J., *Learning Internal Representation by Error Propagation, Parallel Distributed Processing*, Ed. Cambridge MA, MIT Press, 1986.
- [9] Chen X., Feng G., Jixin Q., Adaptive control based on RBF networks, Proc. of 35th Conf. Decis. Contr., Kobe, Japan, 1996, 3810-3815.

Received: June 19, 1997

Viện Công nghệ thông tin, Trung tâm KHTN và CNQG.