# MÔ PHỎNG BỘ ĐIỀU KHIỂN NEURON VỚI LUẬT HỌC HỆ SỐ HỌC THÍCH NGHI VÀ PHƯƠNG PHÁP XUNG LƯỢNG

**Từ Diệp Công Thành**
Trường Đại học Bách Khoa, ĐHQG-HCM
*(Bài nhận ngày01 tháng 11 năm 2007, hoàn chỉnh sửa chữa ngày 03 tháng 03 năm 2008)*

***TÓM TẮT:*** *Trong những năm gần đây, mạng thần kinh nhân tạo đã được áp dụng thành công trong nhiều lĩnh vực kỹ thuật như xử lí tín hiệu, nhận dạng hình ảnh, giao thông, y học, điều khiển… Nhiều sơ đồ điều khiển dùng mạng thần kinh với thuật toán lan truyền ngược được ứng dụng để giải các bài toán điều khiển các hệ phi tuyến phức tạp và bất ổn định. Thuật toán suy giảm độ dốc là một trong những thuật toán đơn giản và thường dùng nhất để huấn luyện mạng thần kinh. Để đảm bảo thuật toán luôn hội tụ và huấn luyện mạng nhanh, có hai phương pháp nhằm nâng cao chất lượng mạng là dùng hệ số học thích nghi và phương pháp xung lượng. Trong bài viết này, chúng ta sẽ tiến hành lập trình mô phỏng, kiểm chứng và so sánh các phương pháp trên bằng chương trình MATLAB.*

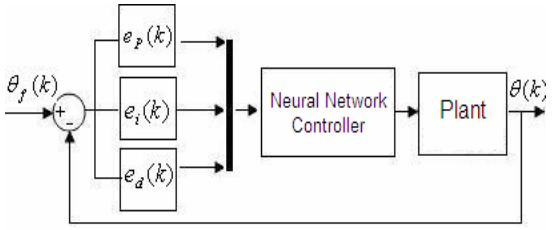
## 1.GIỚI THIỆU

Sự phát triển không ngừng của khoa học công nghệ làm xuất hiện các đối tượng điều khiển có độ phức tạp ngày càng tăng. Yêu cầu thực tiễn đặt ra là phải điều khiển các hệ thống động ngày càng phức tạp, trong điều kiện các yếu tố bất định ngày càng gia tăng với yêu cầu chất lượng điều khiển ngày càng cao. Các yêu cầu trên không thể được đáp ứng một cách trọn vẹn, đồng thời nếu chỉ dùng các lí thuyết điều khiển thông thường sẵn có. Đây chính là động lực cho sự ra đời của hàng loạt các lí thuyết điều khiển hiện đại. hứa hẹn một hướng giải quyết triệt để các bài toán điều khiển phi tuyến phức tạp.

Trong những năm gần đây, mạng thần kinh nhân tạo đã được áp dụng thành công vào những lĩnh vực kỹ thuật như giao thông[1], robot[2], thị giác máy tính[3], tay máy[4]… Nhiều sơ đồ điều khiển dùng mạng thần kinh với thuật toán lan truyền ngược được ứng dụng để giải các bài toán điều khiển các hệ phi tuyến phức tạp và bất ổn định[5][6]. Tính thích nghi cho phép mạng thần kinh vẫn thực hiện tốt chức năng của nó khi môi trường và đối tượng điều khiển thay đổi theo thời gian bằng cách cập nhật cấu trúc  mạng cũng như các trọng số của mình. Có rất nhiều thuật toán đã được phát triển để huấn luyện mạng thần kinh với nhưng ưu và khuyết điểm riêng[7][8]. Thuật toán suy giảm độ dốc là một trong những thuật toán đơn giản và thường dùng nhất để cập nhật các trọng số của mạng thần kinh. Nhằm nâng cao chất lượng mạng, hai phương pháp huấn luyện mạng nhằm đảm bảo thuật toán luôn hội tụ và huấn luyện nhanh là hệ số học thích nghi và phương pháp xung lượng. Trong bài viết này, chúng ta sẽ tiến hành lập trình mô phỏng, kiểm chứng và so sánh các phương pháp huấn luyện mạng trên bằng chương trình MATLAB.

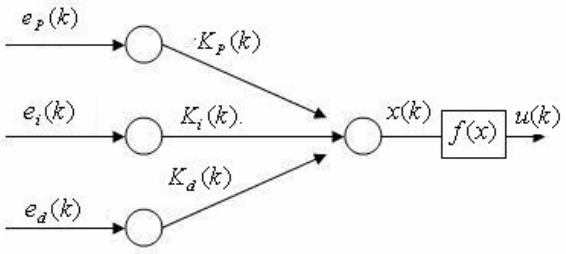
## 2.THUẬT TOÁN SUY GIẢM ĐỘ DỐC THÔNG THƯỜNG

Bộ điều khiển sử dụng mạng thần kinh có thể dùng trong các sơ đồ điều khiển khác nhau. Điều khiển trực tiếp là một trong những sơ đồ thường gặp nhất. Tín hiệu ra của đối tượng được so sánh với tính hiệu đặt, nếu có sai lệch thì bộ điều khiển sẽ xuất tín hiệu tác động vào đối tượng nhằm mục đích làm sai lệch giảm về 0. Sơ đồ điều khiển được thể hiện như hình 1.




**Hình 1.** Sơ đồ điều khiển

Sơ đồ chi tiết của bộ điều khiển sử dụng mạng thần kinh như hình 2.Mạng thần kinh được sử dụng bao gồm 2 lớp: lớp vào và lớp ra được huấn luyện bằng giải thuật lan truyền ngược để tối thiểu hóa sai số giữa tín hiệu đặt và tín hiệu ra của hệ thống.




**Hình 2.** Sơ đồ chi tiết bộ điều khiển sử dụng mạng thần kinh

Trong đó $K_P, K_i, K_d$ lần lượt là các khâu tỉ lệ, tích phân, vi phân; $e_P$, $e_i$, $e_d$ lần lượt là sai số hệ thống giữa tín hiệu đặt $\theta_f$ và tín hiệu ra của hệ thống $\theta$, tích phân của sai số và sự sai lệch của sai số.

Tính hiệu điều khiển $u(k)$ được xác định bằng công thức sau:

$$u(k) = f(x)$$

Trong đó $f(.)$ là hàm tác động dạng sigmoid

$$f(x) = \frac{xg.(1 - e^{-4x/xg})}{2.(1 + e^{-4x/xg})} \quad (1)$$

Trong đó $x$ là đối số đầu vào, $xg$ là tham số xác định hình dạng của hàm.
Ta có:

$$x(k) = K_P(k)e_P(k) + K_i(k)e_i(k) + K_d(k)e_d(k) \quad (2)$$

Trong đó:

$$e_P(k) = \theta_f(k) - \theta(k)$$

$$e_i(k) = \sum_{n=1}^{k} e_p(n).\Delta T$$

$$e_d(k) = \frac{e_p(k)(1 - z^{-1})}{.\Delta T} \tag{3}$$

$\Delta T$ là thời gian lấy mẫu

K discrete sequence

z: operator of Z – transform

Để hiệu chỉnh các thông số của bộ điều khiển, ta sử dụng phương pháp suy giảm độ dốc có công thức như sau:

$$K_P(k+1) = K_P(k) - \eta_P \frac{\partial E(k)}{\partial K_P}$$

$$K_i(k+1) = K_i(k) - \eta_i \frac{\partial E(k)}{\partial K_i}$$

$$K_d(k+1) = K_d(k) - \eta_d \frac{\partial E(k)}{\partial K_d} \tag{4}$$

Trong đó $\eta_P, \eta_i, \eta_d$ là các hệ số học

Ở đây tiêu chuẩn huấn luyện mạng được sử dụng là chuẩn toàn phương :

$$E(k) = \frac{1}{2}\left(\theta_f(k) - \theta(k)\right)^2 \tag{5}$$

Mặt khác ta có :

$$\frac{\partial E(k)}{\partial K_P} = \frac{\partial E(k)}{\partial \theta}\frac{\partial \theta(k)}{\partial u}\frac{\partial u(k)}{\partial x}\frac{\partial x(k)}{\partial K_P}$$

$$\frac{\partial E(k)}{\partial K_i} = \frac{\partial E(k)}{\partial \theta}\frac{\partial \theta(k)}{\partial u}\frac{\partial u(k)}{\partial x}\frac{\partial x(k)}{\partial K_i}$$

$$\frac{\partial E(k)}{\partial K_d} = \frac{\partial E(k)}{\partial \theta}\frac{\partial \theta(k)}{\partial u}\frac{\partial u(k)}{\partial x}\frac{\partial x(k)}{\partial K_d} \tag{6}$$

Vậy:

$$\frac{\partial E(k)}{\partial \theta} = -(\theta_f(k) - \theta(k)) = -e_P(k)$$
$$\frac{\partial u(k)}{\partial x} = f'(x(k)); \frac{\partial x(k)}{\partial K_P} = e_P(k)$$
$$\frac{\partial x(k)}{\partial K_i} = e_i(k); \frac{\partial x(k)}{\partial K_d} = e_d(k) \quad (7)$$

Từ (3) (5) (6) ta có:

$$\frac{\partial E(k)}{\partial K_P} = -e_P(k)\frac{\partial \theta(k)}{\partial u} f'(x(k)).e_P(k)$$
$$\frac{\partial E(k)}{\partial K_i} = -e_i(k)\frac{\partial \theta(k)}{\partial u} f'(x(k)).e_i(k)$$
$$\frac{\partial x(k)}{\partial K_d} = -e_d(k)\frac{\partial \theta(k)}{\partial u} f'(x(k)).e_d(k) \quad (8)$$

Để cho đơn giản, theo Yamada và Yubuta [9], ta có thể giả sử $\frac{\partial \theta(k)}{\partial u} = 1$, cuối cùng ta được

$$K_P(k+1) = K_P(k) - \eta_P e_P(k).e_P(k) f'(x)$$
$$K_i(k+1) = K_i(k) - \eta_i e_P(k) e_i(k) f'(x)$$
$$K_d(k+1) = K_d(k) - \eta_d e_P(k) e_d(k) f'(x) \quad (9)$$

Mô hình của đối tượng được sử dụng trong thí nghiệm là:

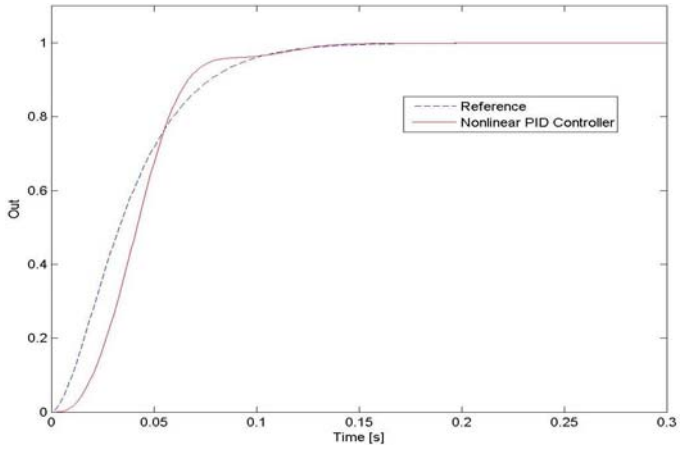
$$G(s) = \frac{740}{s^2 + 13s} \quad (10)$$

Tín hiệu đầu vào của hệ thống được sử dụng là hàm step được làm trơn:

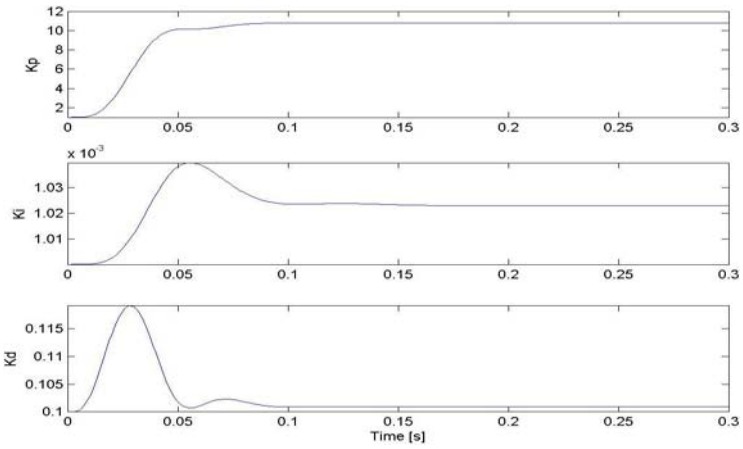
$$F(s) = \frac{2500}{s^2 + 100s + 2500} \quad (11)$$

Hình 3 thể hiện đáp ứng ngõ ra của hệ thống sử dụng bộ điều khiển thần kinh với thuật toán suy giảm độ dốc thông thường. Trong bài thí nghiệm này, các hệ số $K_P$, $K_i$ và $K_d$ được gán giá trị ban đầu lần lượt là 1; 0,001 và 0,1. Các giá trị này được lựa chọn ngẫu nhiên để kiểm tra khả năng thích nghi, tối ưu hóa các trọng số của bộ điều khiển. Các hệ số học $\eta_P, \eta_i, \eta_d$ trong công thức (4) được gán các giá trị lần lượt là 10; 0,001; 0,005. Các hệ số này được lựa chọn theo phương pháp thử sai trong suốt quá trình thí nghiệm để có được đáp ứng hệ thống tương đối tốt. Từ hình 3, ta có thể thấy được tính thích nghi của bộ điều khiển, đáp ứng ngõ ra của hệ thống nhanh chóng đạt được giá trị mong muốn cần điều khiển.

Hình 4 thể hiện các giá trị trọng số $K_P$, $K_i$ và $K_d$ của bộ điều khiển trong quá trình thí nghiệm.Trong hình vẽ 4, ta có thể thấy $K_P$ tăng nhanh để tăng đáp ứng hệ thống và đạt được giá trị tối ưu khi hệ thống đạt được giá trị xác lập, $K_d$ tăng rất nhanh trong khoảng thời gian ban đầu để tăng đáp ứng của hệ thống và nhanh chóng giảm ở cuối thời gian xác lập.




**Hình 3.** Đáp ứng ngõ ra của bộ điều khiển thần kinh




**Hình 4.** Kết quả thí nghiệm sử dụng bộ điều khiển dùng mạng thần kinh với phương pháp suy giảm độ dốc thông thường

## 3.THUẬT TOÁN SUY GIẢM ĐỘ DỐC VỚI HỆ SỐ HỌC THÍCH NGHI

Một trong những yếu tố quan trọng ảnh hưởng mạnh đến tốc độ học và tính hội tụ của thuật toán lan truyền ngược là hệ số học $\eta$. Giá trị $\eta$ lớn làm tăng tốc độ học, nhưng nếu lớn quá thì có thể làm cho thuật toán không hội tụ, ngược lại giá trị $\eta$ nhỏ bảo đảm thuật toán hội tụ nhưng tốc độ học lại rất chậm.

Phương pháp hiệu quả nhất để đảm bảo thuật toán lan truyền ngược vừa hội tụ vừa huấn luyện mạng nhanh là dùng hệ số học thích nghi như sau: ở mỗi bước lặp ta kiểm tra xem trọng số vừa được cập nhật có làm giảm tiêu chuẩn huấn luyện mạng không, nếu không có nghĩa là đã xảy ra vọt lố, trong trường hợp này nên giảm $\eta$; ngược lại nếu trong vài bước lặp liên tiếp tiêu chuẩn huấn luyện mạng đều giảm thì $\eta$ quá nhỏ, trong trường hợp này nên tăng $\eta$.

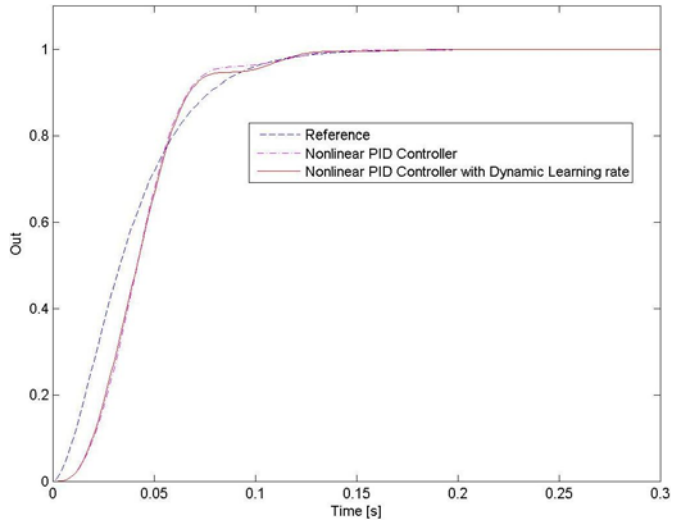
Cụ thể thuật toán học thích nghi được mô tả bởi các biểu thức:

$$\eta(k+1) = \eta(k) + \Delta\eta$$

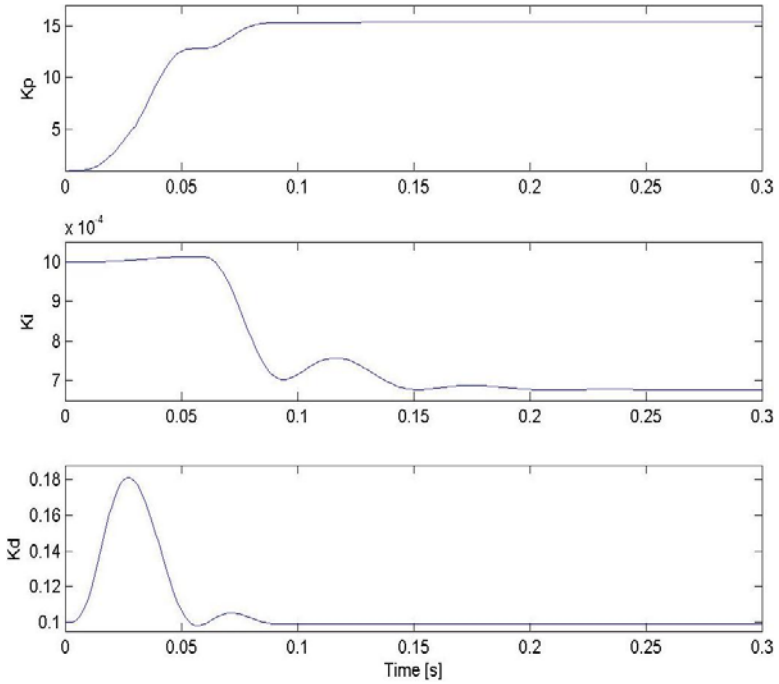
$$\Delta\eta = \begin{cases} +a \\ -b.\Delta\eta \\ 0 \end{cases} \tag{12}$$

Hình 5 thể hiện đáp ứng ngõ ra của hệ thống sử dụng bộ điều khiển thần kinh có dùng hệ số học thích nghi. Trong hình 5, ta có thể thấy hệ thống cũng nhanh chóng đạt được giá trị xác lập, tuy nhiên khả năng đeo bám của nó tốt hơn so với thuật toán suy giảm độ dốc thông thường. Chất lượng hệ thống được cải thiện so với phương pháp suy giảm độ dốc thông thường.

Hình 6 thể hiện các giá trị trọng số $K_P$, $K_i$ và $K_d$ của bộ điều khiển trong quá trình thí nghiệm.




**Hình 5**. So sánh đáp ứng ngõ ra của hệ thống sử dụng bộ điều khiển mạng thần kinh sử dụng phương pháp suy giảm độ dốc có dùng và không dùng hệ số học thích nghi




**Hình 6.** Kết quả thí nghiệm sử dụng bộ điều khiển dùng mạng thần kinh với phương pháp suy giảm độ dốc thông thường

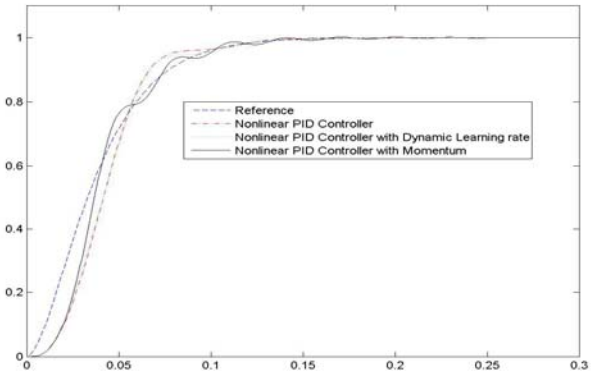
## 4.THUẬT TOÁN SUY GIẢM ĐỘ DỐC VỚI PHƯƠNG PHÁP XUNG LƯỢNG

Phương pháp suy giảm độ dốc thông thường có thể rất chậm nếu hệ số học quá nhỏ và có thể dao động mạnh nếu hệ số học quá lớn. Hiện tượng này thường xảy ra nếu điểm cực tiểu có dạng thung lũng có độ dốc lớn ở sườn và độ dốc nhỏ ở đáy. Một phương pháp hiệu quả thường dùng cho phép hệ số học lớn mà không xảy ra dao động phân kỳ là cộng thêm một xung lượng vào phương pháp suy giảm độ dốc thông thường.

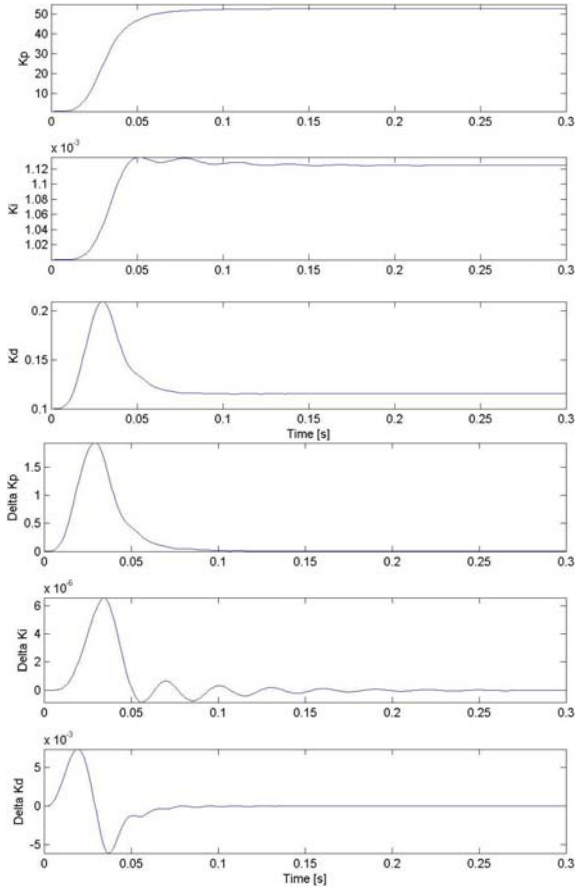
$$K_i(k+1) = K_i(k) + \Delta K_i(k)$$
$$K_i(k) = -\eta_i . \frac{\partial E(k)}{\partial K_i} + \alpha . \Delta K_i(k) \quad (13)$$

Trong công thức trên, α gọi là hệ số xung lượng ($0 \le \alpha \le 1$), giá trị thường dùng là $\alpha = 0{,}9$. Nhờ xung lượng cộng vào mà mỗi trọng số sẽ thay đổi theo hướng độ dốc trung bình, do đó thuật toán không bị dao động.

Hình 7 thể hiện đáp ứng ngõ ra của hệ thống sử dụng bộ điều khiển thần kinh có dùng hệ số học thích nghi. Trong hình 5, ta có thể thấy hệ thống cũng nhanh chóng đạt được giá trị xác lập, tuy nhiên khả năng đeo bám của nó tốt hơn hẳn so với hai thuật toán trên.




**Hình 7.** So sánh đáp ứng ngõ ra của hệ thống sử dụng bộ điều khiển mạng thần kinh sử dụng phương pháp suy giảm độ dốc thông thường, dùng hệ số học thích nghi và phương pháp xung lượng




**Hình 8.** Kết quả thí nghiệm sử dụng bộ điều khiển dùng mạng thần kinh với phương pháp suy giảm độ dốc thông thường với phương pháp xung lượng

Hình 8 thể hiện các giá trị trọng số $K_P, K_i$ và $K_d$ cũng như $\Delta K_P, \Delta K_i$ và $\Delta K_d$ của bộ điều khiển trong quá trình thí nghiệm.

## 5.KẾT LUẬN

Bài báo đã cung cấp cho chúng ta một bộ điều khiển tiên tiến sử dụng mạng thần kinh với các giải thuật huấn luyện mạng và các phương pháp nâng chất lượng mạng, báo đảm tính hội tụ và huấn luyện mạng nhanh. Kết quả mô phỏng đã cho thấy bộ điều khiển rất thích hợp cho các bài toán điều khiển đeo bám mục tiêu, các hệ thống phi tuyến phức tạp trong thực tế do nó có tính thích nghi, tối ưu hóa các trọng số trong quá trình điều khiển. Trong tương lai, các giải thuật điều khiển trên hứa hẹn một hướng giải quyết triệt để cho các hệ thống phi tuyến phức tạp và bất ổn định trong thực tế như điều khiển tay máy, các hệ MIMO như Scara Robot và Omni Mobile Robot.

## BACK PROPAGATION ALGORITHM WITH ADAPTATION LEARNING RATE AND MOMENTUM METHOD IN NEURAL NETWORK CONTROLLER

**Tu Diep Cong Thanh**
University of Technology, VNU-HCM

***ABSTRACT:*** *In recent years, Artificial Neural Network has been successfully used in many industrial applications such as signal processing, image identification, transport, medicine, control… Many neural network control schemes using Back Propagation algorithm have been used for a kind of plant with nonlinearity uncertainties and disturbances. And Gradient Descent is one of popular and simple algorithms for training of neural network. In order to ensure algorithm always converge and fast network training, two methods are used to improve network's performance - Adaptation Learning Rate and Momentum method. In this study, we will simulate, verify and compare those theories using MATLAB package.*

## TÀI LIỆU THAM KHẢO

[1]. Jeongdai Jo, Dong-Soo Kim, Kwang-Young Kim, *Design and Implementation of Autopilot for Vehicle Control*, ISEE (2005)

[2]. A.M.S. Zalzala and A.S.Morris. Braintain, *Neural Network for Robotic Control. Theory and Application,*(1996).

[3]. Nguyễn Đức Minh. *Điều khiển Robot Scorbot dùng thị giác máy tính,* Luận văn thạc sĩ. Mã số ĐKKT-K13-099. Đại học Bách Khoa TP.HCM. (2004)

[4]. Hesselroth, T., Sarkar, K., Patrick van der Smagt, P., and Schulten, K., *Neural network control of a pneumatic robot arm,* IEEE Trans Syst., Man., Cybernetics, Vol. 24, No 1, pp. 28-38, (1994).

[5]. Aders Forsgren, Robert Kling, *An Implementation of Recurrent Neural Network for Prediction and Control of Nonlinear Dynamic Systems*, Tech Report, Monash Univ., Melbourne, Autralia, March 27, (2003).

[6]. Asriel U. Levin, Kumpati S. Nadrendra, *Control of Nonliear Systems Using Neural Networks*, IEEE Trans. on N.Networks, Jan.(1996).

[7]. Syed Muhammand Aqil Burney, Tahseen Ahmed Jilani, Cemal Ardil., *A comparison of First and Second Order Training Algorithms for Artificial Neural Networks*. International Journal of Computational Intelligence. Volume 1 number 3, ISSN: 1304-4508, (2004).

[8]. Bogdan M. Wilamowski, M. Önder Efe. *An Algorithm for Fast Convergence in Training Neural Networks*. IEEE (2001)

[9]. Yamada, T., Yabuta, T., *Neural network controller using autotuning method for nonlinear functions,* in IEEE Trans., Neural Networks, Vol. 3, pp. 595-601, (1992).