

# KẾT HỢP MẠNG NƠ-RON RBF VÀ GIẢI THUẬT DI TRUYỀN TRONG NHẬN DẠNG VÀ XỬ LÝ TÍN HIỆU PHI TUYẾN

Học viên: NGUYỄN NGỌC MINH THÔNG

Hướng dẫn khoa học: PGS.TS NGUYỄN THANH PHƯƠNG

## TÓM TẮT:

Hệ phi tuyến tồn tại ở hầu hết các hệ thống điều khiển, để đạt được giá trị đầu ra mong muốn đòi hỏi phải có một bộ điều khiển được thiết kế phù hợp với từng hệ thống khác nhau. Với đòi hỏi ngày càng cao của hệ thống điều khiển chính xác cần có những giải pháp điều khiển ngày càng tối ưu hơn. Ứng dụng trí tuệ nhân tạo để giải quyết bài toán này là một trong những hướng nghiên cứu hiện đại và thiết thực nhất. Mạng Nơ-ron dùng hàm Xuyên tâm cơ sở (Neural network Radial Basic Function – RBFNN) là công cụ mạnh trong nhận dạng và xấp xỉ hệ phi tuyến. Tuy nhiên RBFNN vẫn có nhược điểm đòi hỏi người sử dụng phải có kinh nghiệm trong việc chọn các thông số học. Để giải quyết vấn đề này tác giả đề xuất giải pháp dùng Giải thuật di truyền (Genetic Algorithm – GA) giúp huấn luyện RBFNN trở nên dễ dàng hơn không cần phải có nhiều kinh nghiệm lựa chọn các tham số học cho RBF.

**Từ khóa:** Mạng nơ-ron Xuyên tâm cơ sở, Giải thuật di truyền, Điều khiển phi tuyến, Phân tích hệ thống phi tuyến.

## ABSTRACT:

Nonlinear systems exist on the most of the control systems, to achieve the desired output values require a controller is designed to suit each different systems. With requiring the high technology of the precise control system needs many control solutions is more optimal. Applying the artificial intelligence to solve this problem is the one of the morden research and practically. Neural network Radial Basic Function (RBFNN) is a powerful tool in the identification and approximate nonlinear systems. However, RBFNN has a limitation to require to has experience in the selection of training parameters of them. To resolve this issue, I propose Genetic Algorithm (GA) to training parameters of RBFNN and do not need to have any experience to select of them.

**Keywords:** Neural network Radial Basic Function, Genetic algorithm, Nonlinear control, Nonlinear systems analysis.

## 1. ĐẶT VẤN ĐỀ

Điều khiển phi tuyến đã được nhiều nhà vật lý, nhà toán học, nhà khoa học, nhà thiết kế hệ thống điều khiển,... quan tâm từ các hệ thống, thiết bị đơn giản phục vụ sinh hoạt hằng ngày, đến các robot, thiết bị không gian vũ trụ. Trong đó các

thay đổi đơn giản trong một phần của hệ thống hoặc có nhiều loạn tác động đến hệ thống sẽ tạo ra các hiệu ứng phức tạp và có thể dẫn đến tan rã cả hệ thống điều khiển. [1]

Các phương pháp cổ điển trong điều khiển phi tuyến nổi bật là các phương

pháp phân tích mặt phẳng pha; phương pháp phân tích và điều khiển hệ Hammerstein; hệ Wiener, phương pháp cân bằng điều hòa; lý thuyết Lyapunov hay phương pháp điều khiển trượt (*Sliding Mode Control - SMC*),... giúp khai phá về hệ phi tuyến và điều khiển phi tuyến. [2]

Với các hệ điều khiển phi tuyến thực theo tác giả Nguyễn Doãn Đức [3] nhờ sự ra đời của Lý thuyết tập mờ (*Fuzzy logic – Klir 1997*), mạng nơ-ron nhân tạo (*Artificial Neural network*), Giải thuật di truyền (*Genetic algorithm*), kỹ thuật vi điều khiển giúp cho khoảng cách giữa lý thuyết và thực tế của hệ phi tuyến đã được thu hẹp đáng kể. Trong đó RBFNN với khả năng tính toán mạnh mẽ bằng cấu trúc đơn giản, loại bỏ nhiều yếu tố không cần thiết đã cho thấy khả năng đa dạng của việc xấp xỉ có thể thực hiện trên nhiều hàm phi tuyến một cách đơn giản và có độ chính xác cao.

Tuy nhiên RBFNN có nhược điểm trong xác định các tham số học đòi hỏi phải có nhiều kinh nghiệm dẫn đến khó tiếp cận cho người không am tường hệ thống và các hàm trong RBF. Để hạn chế điều đó tác giả đề xuất dùng GA để tìm chọn lại các tham số sao cho việc chọn lựa các tham số RBF trở nên không cần có nhiều kinh nghiệm giúp quá trình sử dụng RBFNN trở nên dễ dàng hơn mà vẫn đáp ứng được khả năng nhận dạng và xấp xỉ hàm phi tuyến một cách tối ưu hơn so với các cách cổ điển trước đây.

## 2. HUẤN LUYỆN RBFNN

### 2.1 Xấp xỉ mạng nơ-ron RBF trên cơ sở phương pháp giảm Gradient: [4]

Xét mạng nơ-ron RBF có vector ngõ vào  $x = [x_1 \ x_2 \ \dots \ x_n]^T$  và chọn hàm Gaussian  $h_j$  cho nút mạng thứ  $j$ , ta có :

$$h_j(t) = \exp\left(-\frac{\|x(t) - c_j(t)\|^2}{2b_j^2}\right)$$

Ngõ ra của mạng nơ-ron RBF được tính là:

$$y(t) = w^T h = w_1 h_1 + w_2 h_2 + \dots + w_m h_m$$

Trong đó:

$i = 1, 2, \dots, n$ : Số ngõ vào.

$j = 1, 2, \dots, m$  : Số nút ẩn.

$w = [w_1, \dots, w_m]^T$ : Trọng số RBF.

$c = [c_{ij}] = \begin{bmatrix} c_{11} & \dots & c_{1m} \\ \dots & \dots & \dots \\ c_{n1} & \dots & c_{nm} \end{bmatrix}$  là giá trị

tương ứng của tâm hàm Gaussian của mạng  $j$  tại ngõ vào thứ  $i$ .

$b = [b_1, \dots, b_m]^T$ : Trị độ rộng của hàm Gaussian tại mạng nơ-ron  $j$ .

Có 2 cách để tối ưu độ rộng mạng neural RBF:

▪ *Cách 1*: Trong thuật toán huấn luyện nhanh (Quick training) độ rộng được tính theo đề xuất của Looney hoặc Haykin [5]:

Theo Looney:  $b_k = \frac{1}{\sqrt{2(2N)^{k/n}}}$  hoặc

theo Haykin:  $b = \frac{d_{max}}{\sqrt{2N}}$  (2.1)

Trong đó: khoảng cách lớn nhất của các

tâm  $d_{max} = \max\{d(x^i, x^k); i, k \in \{1, \dots, N\}\}$ ;  $N$  là số tâm.

▪ *Cách 2*: Mỗi giá trị độ rộng của hàm Gaussian được ước lượng độc lập. Điều này có thể được tính độ lệch tiêu chuẩn giữa khoảng cách dữ liệu và tâm tương ứng của chúng. Theo Moody and Darken [6] dùng nguyên tắc tiêu chuẩn giá trị lân cận nhất  $r$  để tính độ rộng:

$$b_j = \sqrt{\frac{1}{r} \sum_{i=1}^r \|c_i - c_j\|^2} \quad (2.2)$$

Trong đó  $c_i$  (có  $i=1, \dots, r$ ) là  $r$  tâm lân cận của tâm  $c_j$ , thông thường  $r = 2$  hoặc  $3$ .

## 2.2 Huấn luyện RBFNN

a. Huấn luyện một pha mạng nơ-ron RBF, gồm 2 bước:

- Chọn hệ số độ rộng  $b$  theo (2.1) hoặc (2.2).
- Cập nhật trọng số  $w$  theo phương pháp giảm Gradient

$$\Delta w_j(t) = -\eta \frac{\partial E}{\partial w_j} = \eta(y(t) - y_m(t))h_j$$

$$w_j(t) = w_j(t-1) + \Delta w_j(t) + \alpha(w_j(t-1) - w_j(t-2))$$

Trong đó:  $\eta \in (0,1)$  là hệ số học và  $\alpha \in (0,1)$  là hệ số khuếch đại.

b. Huấn luyện hai pha mạng nơ-ron RBF, gồm 2 bước:

- Cập nhật độ rộng  $b$  theo phương pháp giảm Gradient [4]

$$\Delta b_j(t) = -\eta \frac{\partial E}{\partial b_j} = \eta(y(t) - y_m(t))h_j \frac{\|x - c_j\|^2}{b_j^3}$$

$$b_j(t) = b_j(t-1) + \Delta b_j(t) + \alpha(b_j(t-1) - b_j(t-2))$$

- Cập nhật trọng số  $w$  theo phương pháp giảm Gradient (như cách a)

c. Huấn luyện ba pha mạng nơ-ron RBF, gồm 3 bước:

- Cập nhật độ rộng  $b$  (như cách b)
- Cập nhật trọng số  $w$  (như cách a)
- Cập nhật tâm  $c$  theo phương pháp giảm Gradient:

$$\begin{aligned} \Delta c_{ji}(t) &= -\eta \frac{\partial E}{\partial c_{ji}} \\ &= \eta(y(t) - y_m(t))w_j h_j \frac{x_j - c_{ji}}{b_j^2} \\ c_{ji}(t) &= c_{ji}(t-1) + \Delta c_{ji}(t) + \alpha(c_{ji}(t-1) - w_j(t-2)) \end{aligned}$$

## 2.3 Xấp xỉ thực nghiệm khi lựa chọn các tham số RBFNN theo kinh nghiệm:

Xét đối tượng phi tuyến rời rạc như sau:  $y(k) = u(k)^3 + \frac{y(k-1)}{1+y(k-1)^2}$

Xét cấu trúc mạng neural RBF có 2 ngõ vào, 5 nút ẩn, 1 ngõ ra.

Có:  $x(1) = u(k)$ ;  $x(2) = y(k)$ ;  $\alpha = 0.05$ ;  $\eta = 0.15$ ; Các trọng số ngõ vào được chọn ngẫu nhiên trong khoảng  $[0,1]$ .

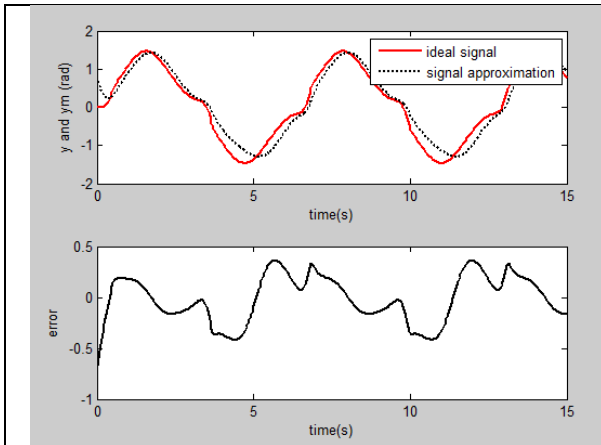
Các tham số đầu vào của hàm Gaussian được chọn như sau:

$$c_j = \begin{bmatrix} -1 & -0.5 & 0 & 0.5 & 1 \end{bmatrix}^T; b_j = 3.0; j = 1,2,3,4,5$$

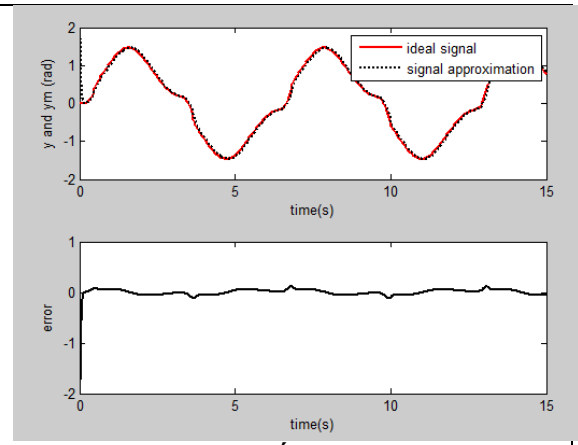
Chọn ngõ vào là hàm  $u(k) = \sin t$ ;

$$t = k \times T; T = 0.03.$$

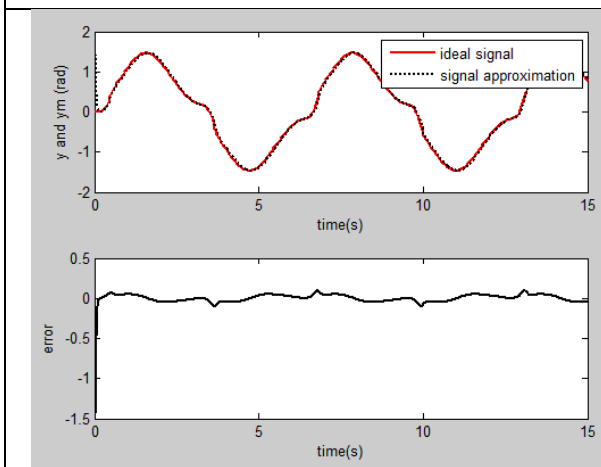
**Mô phỏng trên Matlab Simulink 2012b:**



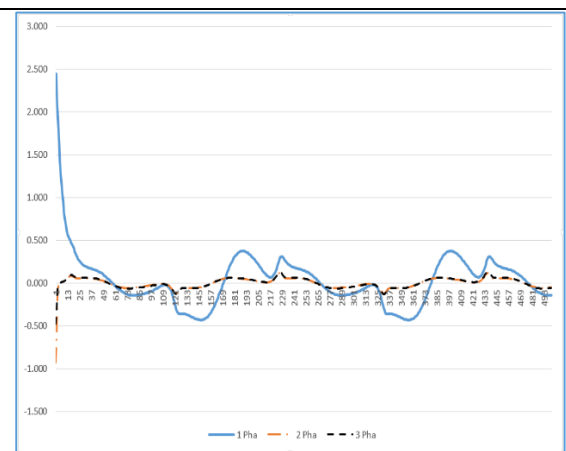
Hình 2.1: Huấn luyện 1 pha



Hình 2.2: Huấn luyện 2 pha



Hình 2.3: Huấn luyện 3 pha



Hình 2.4: Biểu đồ hiệu suất huấn luyện 1 pha, 2 pha, 3 pha của RBFNN

**Bảng 2.1** Hiệu suất huấn luyện một pha, hai pha, ba pha của RBFNN

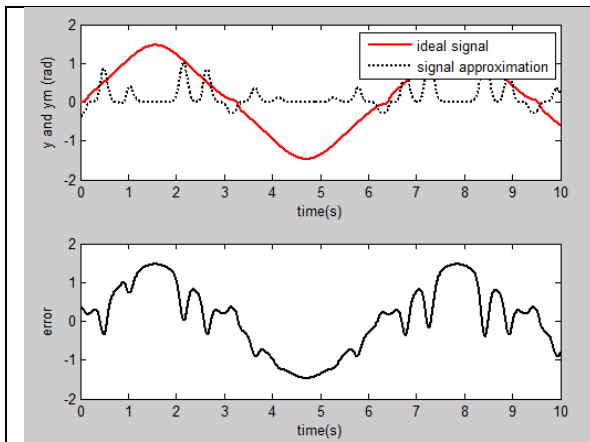
Thời điểm (s)	0.03	2.1	4.2	6.3	8.4	10.5	12.6	15
1 pha	2,444	-0,119	-0,409	0,131	-0,123	-0,412	0,123	-0,144
2 pha	-0,928	-0,056	-0,059	0,009	-0,056	-0,059	0,008	-0,050
3 pha	-0,481	-0,063	-0,061	0,012	-0,063	-0,060	0,011	-0,058

Theo số liệu so sánh tại *Bảng 2.1* và *Hình 2.4*, thực nghiệm trên cho thấy việc huấn luyện mạng nơ-ron RBF có hiệu suất cao khi có thể huấn luyện nhiều pha hơn (ở đây là 3 pha với việc cập nhật huấn luyện ba tham số  $b$ ,  $w$  và  $c$ ). Tuy nhiên đây là việc xấp xỉ theo các tham số lý tưởng và có nhiều kinh nghiệm

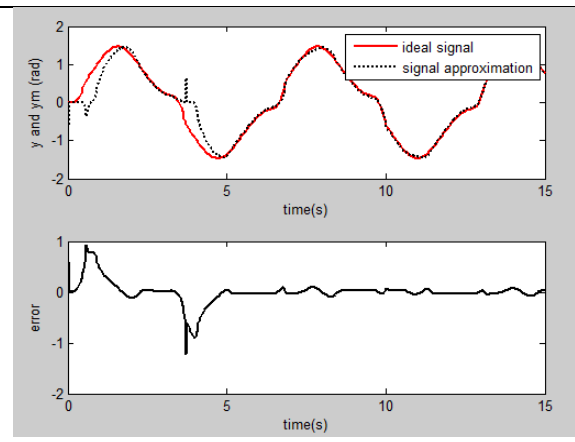
chọn lựa tham số mô phỏng. Vấn đề tiếp tục được đặt ra là với các bài toán việc xác định các tham số trên trở nên khó khăn, hay nói khác hơn chỉ có thể cập nhật một trong các tham số độ rộng ( $b$ ) hay trọng số ( $w$ ) hay tâm ( $c$ ) thì hiệu suất huấn luyện có đạt được yêu cầu hay không.

### 2.4 Xấp xỉ thực nghiệm khi lựa chọn các tham số RBFNN ngẫu nhiên:

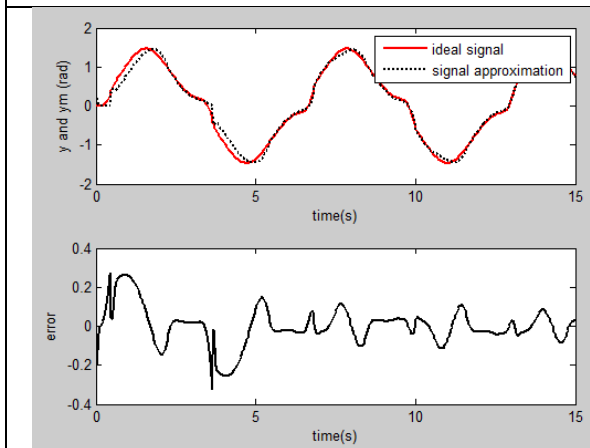
Xét đối tượng như mục 2.3 như trên, nhưng thực nghiệm với việc chọn ngẫu nhiên tham số cho RBF (ở đây chọn ngẫu nhiên  $b = 0.1$ ), kết quả xấp xỉ như sau:



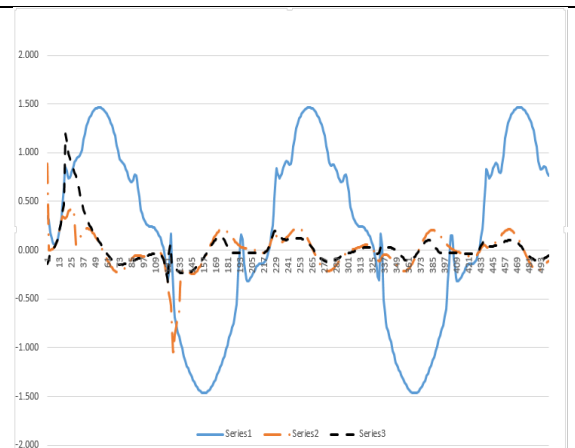
Hình 2.5: Huấn luyện 1 pha



Hình 2.6: Huấn luyện 2 pha



Hình 2.7: Huấn luyện 3 pha



Hình 2.8: Biểu đồ hiệu suất huấn luyện 1 pha, 2 pha, 3 pha của RBFNN

**Bảng 2.2** Hiệu suất huấn luyện một pha, hai pha, ba pha của RBFNN với  $b = 0.1$

Thời điểm (s)	0.03	2.1	4.2	6.3	8.4	10.5	12.6	15
1 pha	0.350	1.070	-1.191	-0,155	0,996	-1,204	-0,148	0,763
2 pha	0.888	-0,224	-0,208	-0,011	-0,216	-0,170	-0,058	-0,107
3 pha	-0,138	-0,137	-0,235	-0,032	-0,121	-0,019	-0,041	-0,056

Từ các số liệu của *Bảng 3.2* và quan sát trên biểu đồ so sánh *Hình 3.8* cho thấy sai số huấn luyện rất cao (sai số lý tưởng là bằng 0) nếu chỉ luyện mạng 1 pha hay

2 pha khi chọn tham số độ rộng  $b$  không phù hợp ( $b = 0.1$ ). Để có thể xấp xỉ tốt hơn chỉ có thể huấn luyện 3 pha điều này sẽ khó thực hiện ở những hệ phi tuyến

phức tạp, do đó cần tìm các giải pháp tối ưu khác để hệ thống có thể đạt được xấp xỉ tốt hơn và áp dụng được các hệ phi tuyến phức tạp hơn.

### 3. GIẢI THUẬT DI TRUYỀN

Giải thuật di truyền (*Genetic Algorithm* – viết tắt *GA*) là một kỹ thuật của khoa học máy tính giúp tìm giải pháp tối ưu cho các vấn đề tối ưu tổ hợp. Trong kỹ thuật điều khiển, *GA* có thể giúp tìm kiếm tối ưu trên toàn bộ không gian của các biến trạng thái. Giống như trong tự nhiên, bằng quá trình chọn lọc, thuật gen sẽ tìm ra những nhiễm sắc thể tốt nhờ hiệu chỉnh thông tin của chúng theo yêu cầu của bài toán, cũng như tối ưu các tham số của bộ điều khiển. [7]

Cấu trúc của giải thuật di truyền đơn giản bao gồm 3 toán tử sau:

- Tái tạo (*Reproduction*).
- Lai ghép (*Crossover*).
- Đột biến (*Mutation*).

Giải thuật di truyền bao gồm các bước sau:

1. Khởi tạo quần thể ban đầu (các đáp án ban đầu của bài toán).
2. Xác định giá trị hàm mục tiêu (*Fitness*) cho cá thể trong quần thể.
3. Tạo ra quần thể mới bằng cách lai ghép chéo (*Crossover*) từ các cá thể hiện tại có chọn lọc (*Selection*), đồng thời tạo ra các đột biến (*Mutation*) trong quần thể mới theo một xác suất nhất định.

4. Xác định hàm mục tiêu cho các chuỗi nhiễm sắc thể mới và đưa nó vào trong một quần thể mới.

**Program:** Huấn luyện mạng nơ-ron RBF có kết hợp GA

**Begin**

**Pha 1:** Dùng GA tìm tham số độ rộng  $h_q$  tối ưu

*Begin*

- Cung cấp tham số đầu vào: Tín hiệu ra mong muốn ( $y_{out}$ ) và tín hiệu ra thực tế ( $y_m$ )

- Khởi tạo quần thể  $h_q$ .

- Giải mã nhiễm sắc thể (mã hóa nhị phân  $h_q$ ).

- Cập nhật giải pháp tốt nhất  $h_q(i)$  sau cho  $(y_{out} - y_m) \rightarrow 0$

*End*

**Pha 2:** Xác định trọng số  $w$  của mạng nơ-ron RBF: Như II.2

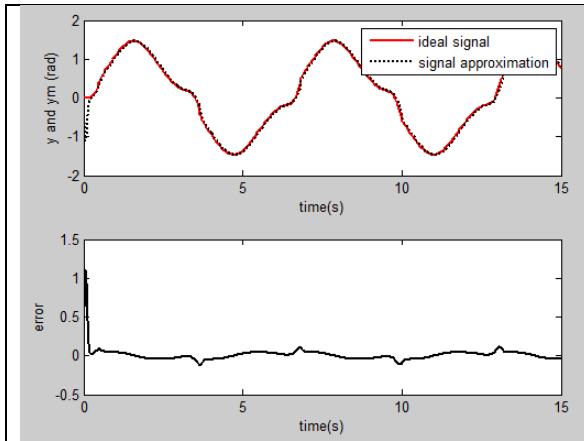
**End**

5. Nếu điều kiện dừng đã thỏa mãn thì dừng lại và trả về chuỗi nhiễm sắc thể tốt nhất cùng với giá trị hàm mục tiêu của nó, nếu không thì quay về bước 3.

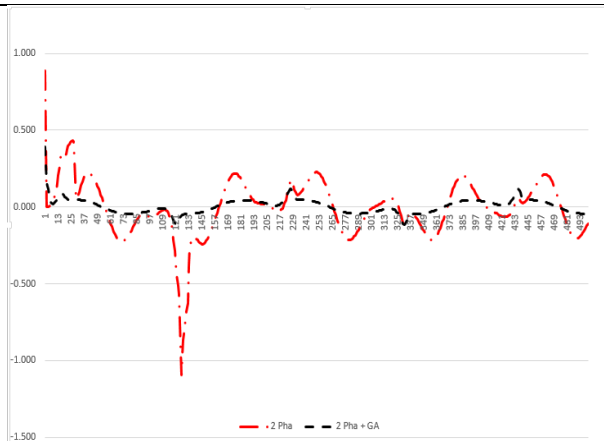
### 3.1 Dùng GA trong huấn luyện hai pha mạng nơ-ron RBF

Thuật toán đề xuất huấn luyện hai pha mạng nơ-ron RBF có sử dụng GA bằng cách dùng GA tối ưu một pha độ rộng  $b$ , pha còn lại cập nhật trọng số  $w$  theo phương pháp giảm Gradient

### 3.2 Kết quả thực nghiệm kết hợp GA-RBF trong bài toán 2.4



Hình 3.1: Hiệu suất huấn luyện 2 pha mạng nơ-ron RBF có sử dụng GA



Hình 3.2: So sánh hiệu suất xấp xỉ hai phương pháp có và không có kết hợp GA

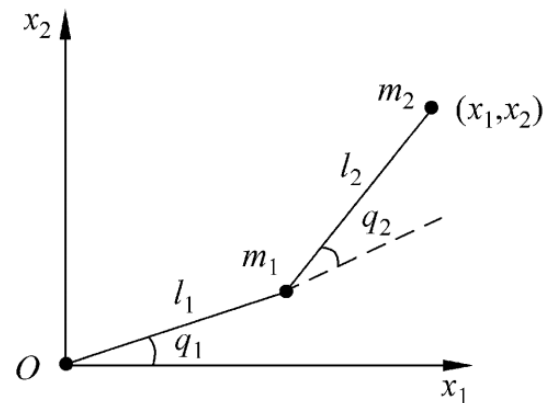
Kết quả mô phỏng trên cho thấy xấp xỉ mạng nơ-ron RBF có kết hợp với Giải thuật di truyền trong tối ưu độ rộng đạt được hiệu suất tốt, có thể áp dụng cho các dạng toán phức tạp khi không cần chú trọng việc chọn độ rộng hay tâm của mạng nơ-ron RBF.

Thuật toán huấn luyện mạng Nơ-ron RBF theo phương pháp giảm Gradient đã có thể xấp xỉ hệ phi tuyến tương đối tốt. Tuy nhiên với kết quả thực nghiệm này cho thấy lợi thế của việc kết hợp với Giải thuật di truyền trên khía cạnh tối ưu độ rộng của mạng nơ-ron RBF cho kết quả tốt hơn trong khi thời gian xử lý là như nhau. Với nghiên cứu này đã đóng góp vào các công trình nghiên cứu về xấp xỉ đối tượng phi tuyến bằng mạng nơ-ron RBF vốn là giải thuật ưu thế nhất đối với hệ phi tuyến.

#### 4. KẾT HỢP RBFNN VÀ GA ĐIỀU KHIỂN CÁNH TAY MÁY HAI BẬC TỰ DO

Với việc kết hợp thành công 2 giải thuật GA và RBF ở trên, tác giả thực hiện

nghiên cứu áp dụng vào cánh tay máy 2 bậc tự do. Mô tả toán học cánh tay máy 2 bậc tự do [4]



Hình 4.1: Mô tả cánh tay máy hai bậc tự do trong không gian Descartes

Phương trình cân bằng động của tay máy hai bậc tự do có dạng:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \quad (4.1)$$

Trong đó:  $\mathbf{M}(\mathbf{q})$ : Ma trận quán tính  $n \times n$ ;  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ : Ma trận lực hướng tâm và hệ số Coriolis  $n \times n$ ;  $\mathbf{G}(\mathbf{q})$ : Ma trận lực hướng tâm và momen  $n \times 1$

Đặt:  $x_1 = \mathbf{q}$ ,  $x_2 = \dot{\mathbf{q}}$ , thay vào phương trình (4.1) ta được:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M^{-1}(x_1)(\tau - C(x_1, x_2)x_2 - G(x_1)) \end{cases} \quad (4.2)$$

**Luật điều khiển sử dụng RBF [18]:**

$$\tau = \tau_m + K_p r + K_i \int r dt + \tau_r \quad (4.3)$$

**Luật đáp ứng của hệ:**

$$\begin{aligned} M(q)\dot{r} + C(q, \dot{q})r + K_i \int_0^t r dt = \\ -K_p r - K_r \text{sgn}(r) + [\{\hat{W}_M\}^T \cdot \{\varepsilon_M(q)\}] \ddot{q}_r + \\ [\{\hat{W}_C\}^T \cdot \{\varepsilon_C(z)\}] \dot{q}_r + [\{\hat{W}_G\}^T \cdot \{\varepsilon_G(q)\}] + E \end{aligned} \quad (4.4)$$

**Điều khiển cánh tay máy hai khớp bằng RBF theo phương pháp Runge-Kutta-Merson [4] [8]**

Xét đối tượng điều khiển tay máy 2 bậc theo phương trình (4.2)

- Thời gian lấy mẫu  $T = 0.001s$ .
- Quỹ đạo mong muốn  $q_{d1} = q_{d2} = \sin(2\pi kT)$ ; thời điểm ban đầu  $q_0 = [0 \ 0]^T$ ,  $\dot{q}_0 = [0 \ 0]^T$
- Luật điều khiển sử dụng RBF theo (5.3), luật đáp ứng theo (5.6).

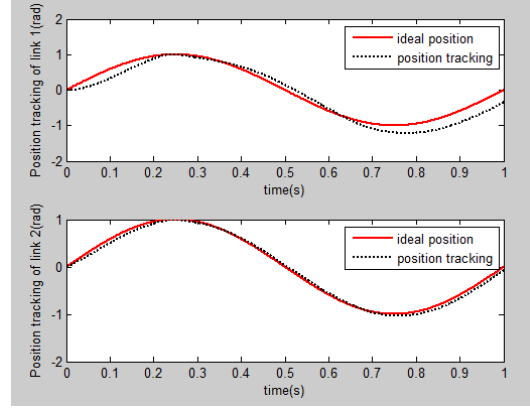
- Các tham số được chọn là:

$$\begin{aligned} K_p &= \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, & K_i &= \\ & & & \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, & K_r &= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \\ \Lambda &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \end{aligned}$$

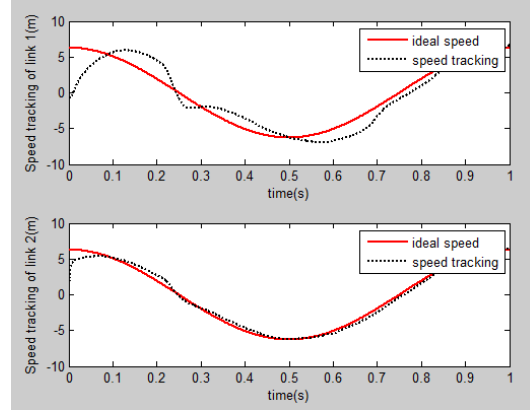
- Chọn cấu trúc mạng RBF 2-5-1, các tham số được chọn theo hàm Gaussian  $c_i = \begin{bmatrix} -1 & 0.5 & 0 & 0.5 & 1 \\ -1 & 0.5 & 0 & 0.5 & 1 \end{bmatrix}$ , tham số  $b = 0.1$  (chọn ngẫu nhiên), thời điểm ban đầu trọng số bằng 0.

Theo luật đáp ứng (4.4) ta chọn:

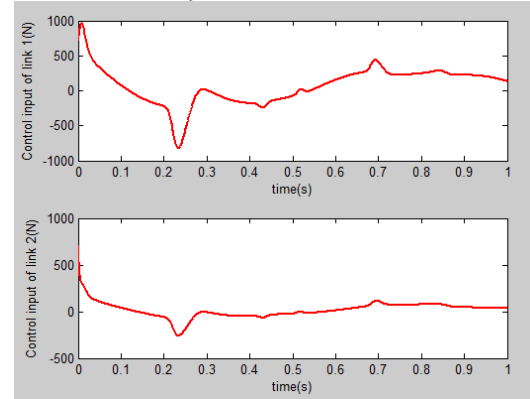
$$\begin{aligned} \Gamma_{Mk}(i, i) &= 5.0, \Gamma_{Ck}(i, i) = 10, \\ \Gamma_{Gk}(i, i) &= 5.0, i = 1, 2, 3, 4, 5. \end{aligned}$$



Hình 4.2: Đáp ứng vị trí của tay máy hai bậc tự do



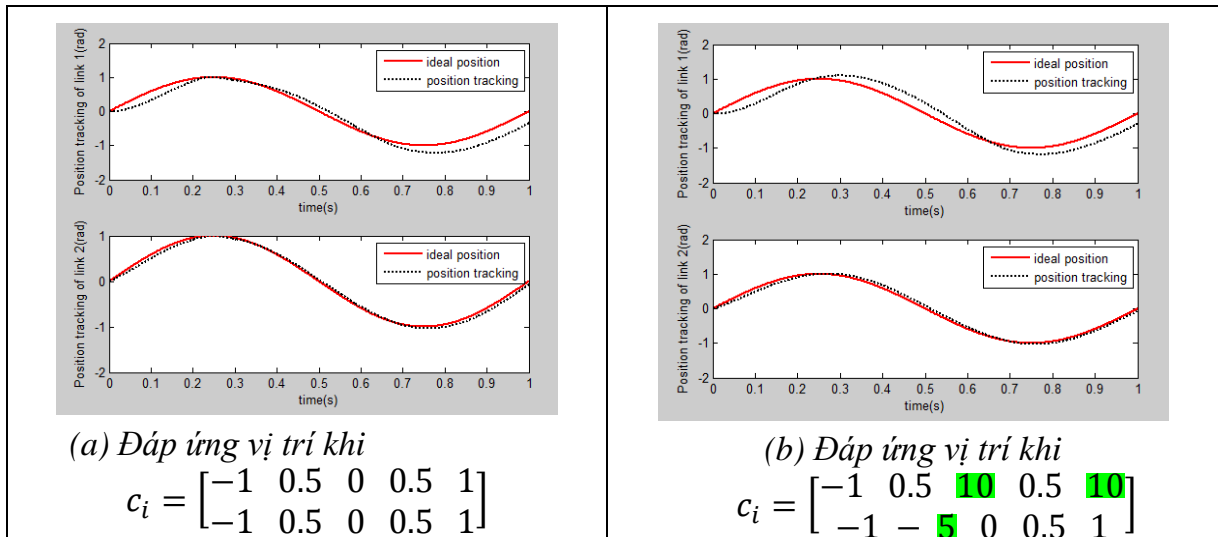
Hình 4.3: Đáp ứng tốc độ của cánh tay máy hai bậc tự do



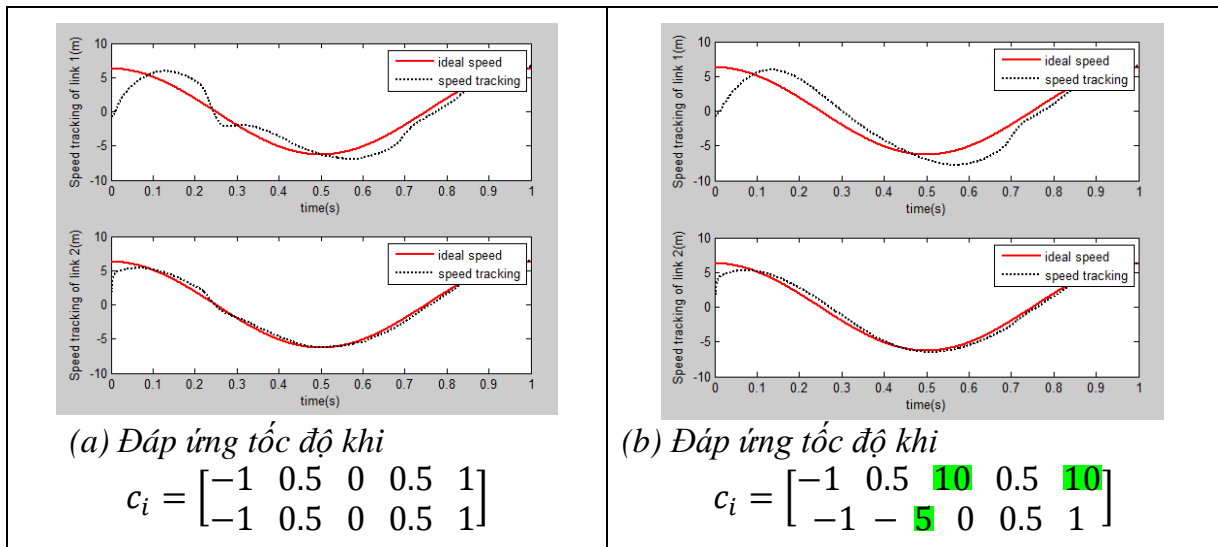
Hình 4.4: Tín hiệu điều khiển cánh tay máy hai bậc tự do



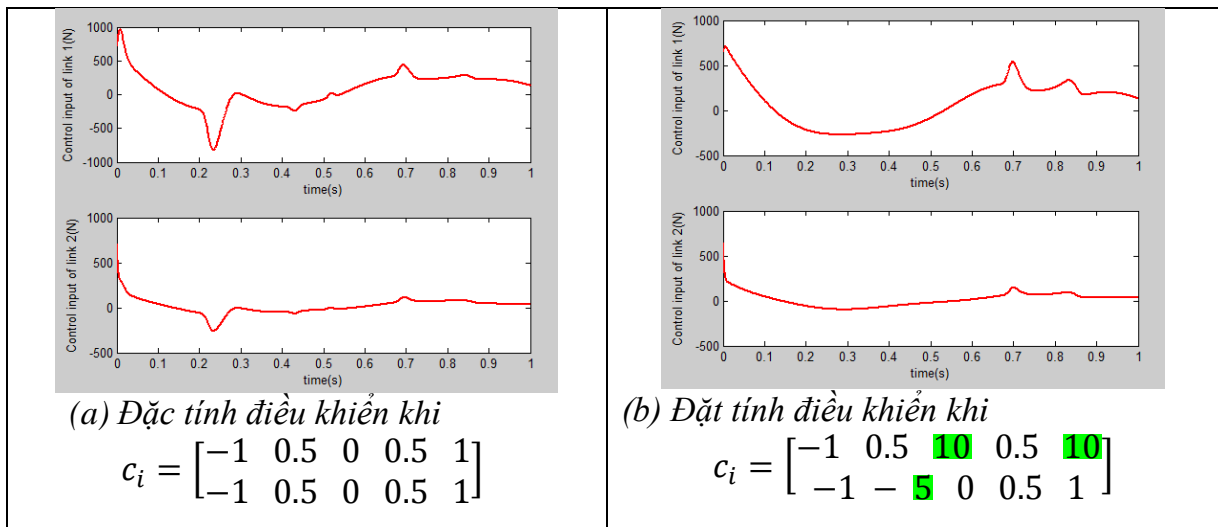
**Mô phỏng so sánh giữa việc chọn tâm:**



Hình 4.5: So sánh đáp ứng vị trí khi chọn tâm  $c_i$  khác nhau



Hình 4.6: So sánh đáp ứng tốc độ khi chọn tâm  $c_i$  khác nhau



Hình 4.7: So sánh đặc tính điều khiển khi chọn tâm  $c_i$  khác nhau

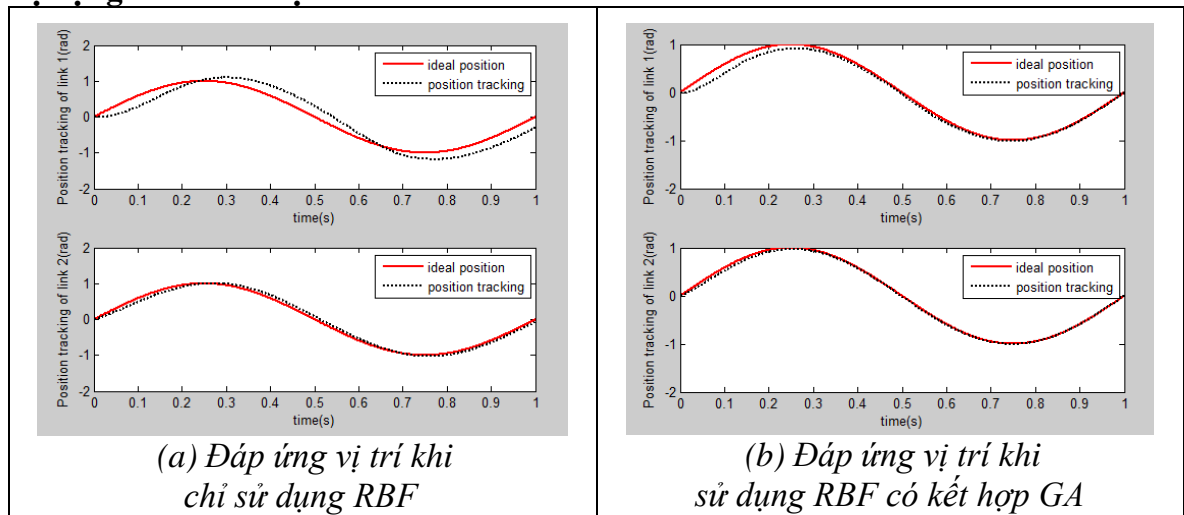
Từ mô phỏng trên ta thấy đáp ứng tốc độ mong muốn của việc chọn tâm  $c_i$  không hợp lý dẫn đến hệ bám chậm hơn. Để ít chịu ảnh hưởng bởi việc chọn tâm  $c_i$  cũng như độ rộng  $b$ , tác giả đề xuất tối ưu việc chọn hệ số độ rộng ( $hq$ ) RBF bằng giải thuật di truyền cho hệ cánh tay máy 2 bậc tự do mà không cần phải dựa vào kinh nghiệm chọn tâm hay độ rộng RBF.

### Kết hợp giải thuật di truyền để tối ưu độ rộng RBF cho hệ điều khiển cánh

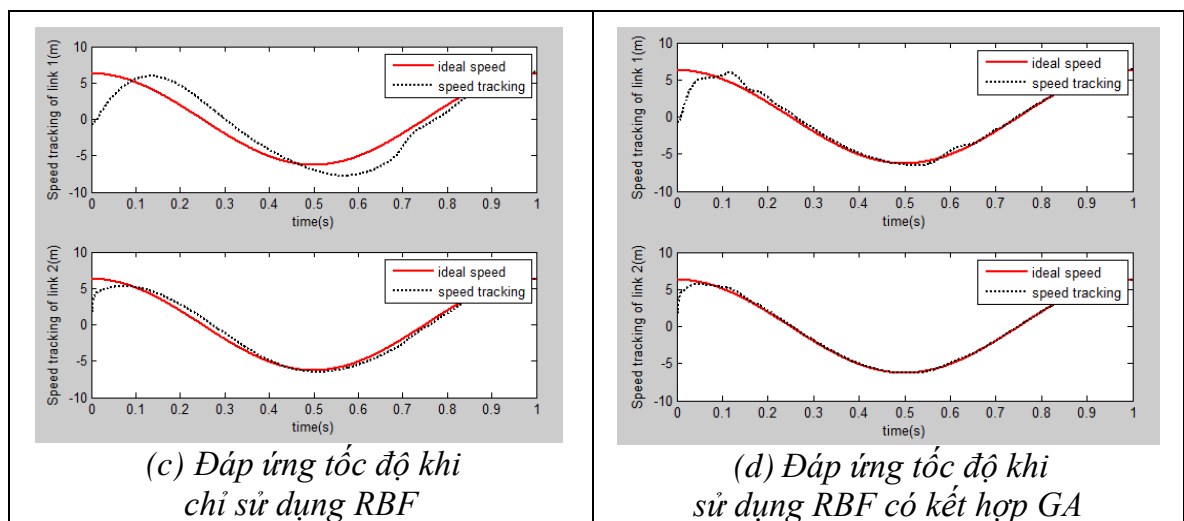
### tay máy 2 bậc tự do sử dụng phương pháp điều khiển Runge-Kutta-Merson

Tối ưu độ rộng RBF bằng giải thuật di truyền dựa trên sai số tốc độ trong luyện mạng 2 pha.

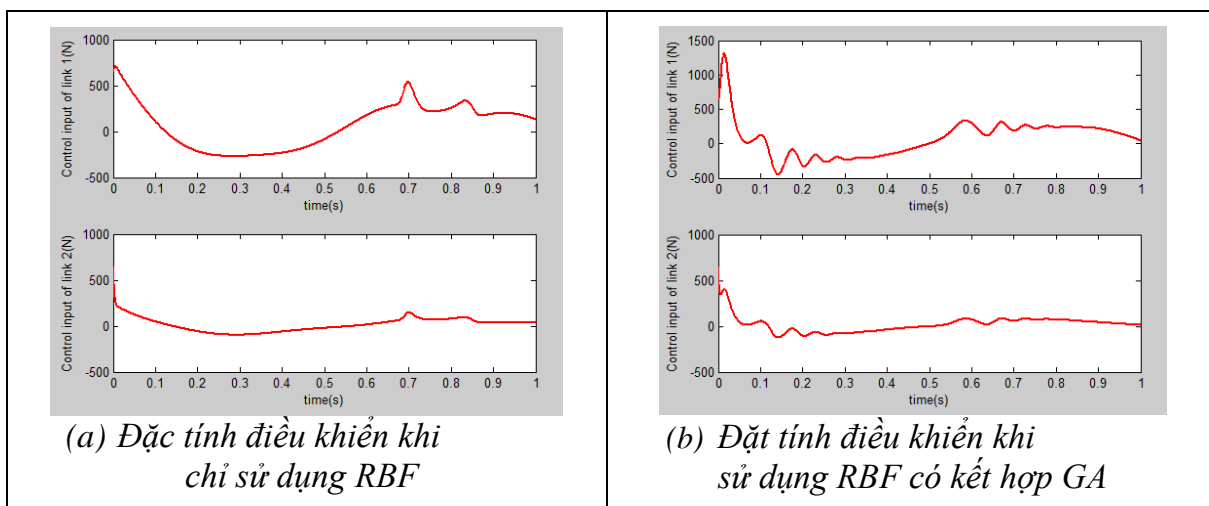
- *Pha 1:* Lựa chọn tham số độ rộng  $hq$  bằng GA, áp dụng như đề xuất 4.1 của luận văn này.
- *Pha 2:* Cập nhật trọng số  $w$  theo phương pháp giảm Gradient.



Hình 4.8: So sánh đáp ứng vị trí khi chưa có và có kết hợp GA



Hình 4.9: So sánh đáp ứng tốc độ khi chưa có và có kết hợp GA



Hình 4.10: So sánh đặc tính điều khiển khi chưa có và có kết hợp GA

Bằng mô phỏng thực nghiệm cho thấy thuật toán của RBFNN kết hợp với điều khiển tối ưu vòng kín  $K_p$ ,  $K_d$  theo phương pháp Runge-Kutta-Merson điều khiển đối tượng phi tuyến (ở đây là cánh tay máy hai bậc tự do) là đạt được yêu cầu. Tuy nhiên việc kết hợp thêm GA làm cho hệ thống phi tuyến này đã tối ưu càng được tối ưu thêm, đồng thời có thể bỏ qua kinh nghiệm lựa chọn các tham số như tâm  $c_i$ , độ rộng  $b$ , hệ số học  $alpha$  ( $\alpha$ ) cho RBFNN.

## 5. KẾT LUẬN

Việc kết hợp RBFNN với GA trong phương pháp Runge-Kutta-Merson điều khiển đối tượng phi tuyến bỏ qua yếu tố kinh nghiệm trong việc chọn các tham số cho RBFNN là một bước ngoặt quan trọng mở ra phương pháp mới giúp dễ dàng nhận dạng và xử lý tín hiệu phi tuyến. Đây là tiền đề có thể phát triển hướng nghiên cứu xấp xỉ nhiều dạng đối tượng phi tuyến khác nhau chỉ sử dụng hai công cụ mạnh của trí tuệ nhân tạo là RBFNN và GA.

## 1. TÀI LIỆU THAM KHẢO

- [1] “<https://vi.wikipedia.org>,” [Trực tuyến]. Available: [https://vi.wikipedia.org/wiki/Hệ\\_thống\\_phi\\_tuyến](https://vi.wikipedia.org/wiki/Hệ_thống_phi_tuyến). [Đã truy cập 20 7 2016].
- [2] W. Jean-Jacques.Slotine, Applied Nonlinear Control, Englewood Cliffs, New Jersey 07632: PrenticeHall.
- [3] N. D. Đức, Phân tích và điều khiển hệ phi tuyến, NXB Bách Khoa Hà Nội, 2012.
- [4] J. Liu, Radial Basis Function (RBF) Neural Network Control for Mechanical Systems, New York Dordrecht London: Springer Heidelberg, 2013.
- [5] S.Haykin, Neural networks: a comprehensive foundation, New Jersey: Prentice-Hall, 1999.

[6] J. Moody, C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, no. 1, p. 281–294, 1989.

[7] J. Carr, "An Introduction to Genetic Algorithms,," May 30, 2014..

[8] "<https://en.wikipedia.org/>," [Online]. Available: [https://en.wikipedia.org/wiki/Runge-Kutta\\_methods](https://en.wikipedia.org/wiki/Runge-Kutta_methods). [Accessed 1 8 2016].

**Thông tin liên hệ tác giả chính (người chịu trách nhiệm bài viết):**

Họ tên: NGUYỄN NGỌC MINH THÔNG

Đơn vị: Trường Đại học Xây dựng Miền Tây

Điện thoại: 0939712127

Email: [thongmtu@gmail.com](mailto:thongmtu@gmail.com)

# BÀI BÁO KHOA HỌC

THỰC HIỆN CÔNG BỐ THEO QUY CHẾ ĐÀO TẠO THẠC SĨ

Bài báo khoa học của học viên

có xác nhận và đề xuất cho đăng của Giảng viên hướng dẫn

---



Bản tiếng Việt ©, TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH và TÁC GIẢ

Bản quyền tác phẩm đã được bảo hộ bởi Luật xuất bản và Luật Sở hữu trí tuệ Việt Nam. Nghiêm cấm mọi hình thức xuất bản, sao chụp, phát tán nội dung khi chưa có sự đồng ý của tác giả và Trường Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh.

**ĐỂ CÓ BÀI BÁO KHOA HỌC TỐT, CẦN CHUNG TAY BẢO VỆ TÁC QUYỀN!**

---

Thực hiện theo MTCL & KHTHMTCL Năm học 2017-2018 của Thư viện Trường Đại học Sư phạm Kỹ thuật Tp. Hồ Chí Minh.