ĐIỀU KHIỂN ROBOT VỚI MẠNG HÀM BÁN KÍNH CƠ SỞ CÓ HỆ SỐ HỌC ĐƯỢC TỐI ƯU BẰNG THUẬT TOÁN DƠI - BAT

TS. Nguyễn Trần Hiệp

Đại học Thành Đông

Email: hiepnt@thanhdong.edu.vn

TÓM TẮT

Bài báo này điều tra việc sử dụng Thuật toán Dơi (BAT) để tối ưu hóa các tham số điều khiển trong bộ điều khiển robot sử dụng Mạng Chức năng Cơ sở Radial (RBFN). Các nghiên cứu trước đây thường chọn các hệ số học tập dựa trên kinh nghiệm, điều này thường yêu cầu thời gian đáng kể để tìm các hệ số đáp ứng yêu cầu của vấn đề điều khiển và không đảm bảo tính tối ưu. Thuật toán Dơi (BAT), lấy cảm hứng từ việc dơi sử dụng định vị bằng tiếng vang để điều hướng, đã được áp dụng để tối ưu hóa toàn cầu các hệ số học tập cho RBFN. Kết quả mô phỏng cho thấy sự cải thiện đáng kể về tốc độ hội tụ và chất lượng điều khiển so với các phương pháp truyền thống.

Từ khóa: *Thuật toán Dơi (BAT), RBFN, Tối ưu hóa Hệ số học tập, Điều khiển Robot.*

ABSTRACT

This paper investigates the use of the Bat Algorithm (BAT) to optimize control parameters in a robot controller utilizing Radial Basis Function Networks (RBFN). Previous studies often selected learning coefficients based on experience, which typically required significant time to find coefficients that meet the control problem's requirements and did not guarantee optimality. The Bat Algorithm (BAT), inspired by bats' use of echolocation for navigation, has been applied to globally optimize the learning coefficients for the RBFN. Simulation results show significant improvements in convergence speed and control quality compared to traditional methods.

Keywords: *Bat Algorithm (BAT), RBFN, Learning Coefficient Optimization, Robot Control.*

1. ĐẶT VẤN ĐỀ

Thuật toán dơi - BAT, được phát triển bởi Xin-She Yang vào năm 2010, dựa trên đặc tính điều hướng âm thanh hoặc sóng âm sinh học của dơi nhỏ (Yang, 2010). Dơi nhỏ thường sử dụng một loại sonar gọi là echolocation để phát hiện mồi, tránh các chướng ngại vật và định vị các khe nghỉ của chúng trong bóng tối. Chúng có thể phát ra một xung âm rất to và lắng nghe âm thanh phản xạ từ các đối tượng xung quanh (Richardson, 2008). Các xung của chúng thay đổi về tính chất và có thể được liên kết với chiến lược săn mồi của chúng, phụ thuộc vào loài. Hầu hết các dơi sử dụng tín hiệu ngắn, tần số biến đổi để quét qua khoảng một tám, và mỗi xung kéo dài trong vài phần nghìn của một giây (lên đến khoảng 8 đến 10 ms) trong khoảng tần số từ 25kHz đến 150 kHz. Thông thường, dơi nhỏ có thể phát ra khoảng 10 đến 20 xung âm như vậy mỗi giây, và tốc độ phát xung có thể tăng lên đến khoảng 200 xung mỗi giây khi đang di chuyển về mục tiêu săn mồi của chúng.

Vì tốc độ âm thanh trong không khí khoảng v = 340 m/s, bước sóng λ của các xung âm siêu âm với tần số f không đổi được cho bởi λ = v/f, nằm trong khoảng từ 2mm đến 14mm cho khoảng tần số điển hình từ 25kHz đến 150kHz. Thú vị là, những bước sóng này cùng kích thước với mồi của chúng.

Mặc dù trong thực tế, dơi nhỏ cũng có thể sử dụng độ trễ thời gian giữa tai và biến đổi độ ồn để cảm nhận môi trường ba chiều, chúng ta chủ yếu quan tâm đến một số đặc điểm của điều hướng âm thanh để liên kết chúng với hàm mục tiêu của một vấn đề tối ưu hóa, điều này làm cho việc đặc tả một thuật toán thông minh, thuật toán dơi trở nên khả thi.

Ưu đểm lớn nhất của BAT là cho phép tìm kiếm tối ưu trên toàn bộ không gian của các biến trạng thái. Do đó đã có nhiều công trình, bài báo sử dụng BAT để tối ưu các tham số của bộ điều khiển, PD hay PID [1]. Khi mạng nơron được sử dụng rộng rãi để xây dựng các bộ điều khiển mạng hàm bán kính cơ sở - RBFN. Đây là một mạng nơron có cấu trúc đơn giản nhưng cho phép xấp xỉ hóa được các hàm phi tuyến. Giống như BAT, thuật học hàm bán kính cơ sở cũng cho phép tối ưu hóa trọng số của mạng nơron trên toàn bộ không gian của biến trạng thái nhưng đảm bảo quá trình học online. Đối với những bài toán cần tối ưu các tham số cố định một cách trực tiếp BAT vẫn thể hiện được ưu điểm của nó. Trong các nghiên cứu gần đây về quá trình học của mạng nơron dù bằng phương pháp hàm bán kính cơ sở hay bằng phương pháp lan truyền ngược theo sai số (EBP) thì hệ số học η của mạng nơron vẫn được chọn theo kinh nghiệm của người lập trình. Bài báo này, tác giả đề cập đến việc sử dụng công cụ BAT để tối ưu hóa hệ số học η của RBFN thay vì lựa chọn bằng kinh nghiệm như đã nêu ra ở trên. Thứ tự của quá trình học và tối ưu hóa hệ số học η của mạng nơron được thực hiện như sau: Đầu tiên chọn một hệ số học η theo kinh nghiệm và RBFN sẽ học online để tìm ra được tập hợp trọng số tối ưu của RBFN. Bước tiếp theo, chúng ta sử dụng BAT để tìm hệ số học η tối ưu tương ứng với tập hợp trọng số của RBFN đã tìm được trước đó. Bài báo được chia thành năm phần. Sau phần mở đầu, phần hai trình bày bài toán điều khiển robot với rất nhiều thành phần bất định và phương pháp điều khiển trượt sử dụng RBFN với hệ số học η được chọn theo kinh nghiệm để xây dựng bộ điều khiển. Phần ba trình bày các bước tối ưu hóa hệ số học η bằng BAT trong bài toán điều khiển robot sử dụng RBFN. Kết quả mô phỏng BAT tối ưu hệ số học cho RBFN điều khiển robot hai bậc tự do được trình bày trong phần bốn. Phần năm là những nhận xét và đánh giá về chất lượng của hệ điều khiển robot sử dụng RBFN có hệ số học được tối ưu hóa bằng BAT.

2. XÂY DỰNG MÔ HÌNH BỘ ĐIỀU KHIỂN ROBOT SỬ DỤNG MẠNG RBFN

2.1. Bài toán điều khiển robot với nhiều thành phần phi tuyến bất định

Phương trình động lực học của một hệ robot *n* khớp cứng có thể được mô tả bằng phương trình vi phân phi tuyến dạng vector [2, 3, 4] như sau:

$$\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}+\hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\hat{\mathbf{g}}(\mathbf{q})+\mathbf{d}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (1)$$

Trong đó: $\mathbf{q}=[q_1,q_2,......q_n]^T$, $\dot{\mathbf{q}}=[\dot{q}_1,\dot{q}_2,......\dot{q}_n]^T$, $\ddot{\mathbf{q}}=\left[\ddot{q}_1,\ddot{q}_2,.....\ddot{q}_n\right]^T$ là các vector biểu diễn vị trí, vận tốc và gia tốc góc của các khớp tương ứng;

$\boldsymbol{\tau} = [\tau_1, \tau_2, ...\tau_n]^T$ là vector biểu diễn momen tác động lên các khớp;

$\hat{\mathbf{M}}(\mathbf{q}) \in \mathbf{R}^{n*n}$ là ma trận quán tính - đối xứng và xác định dương;

$\hat{\mathbf{B}}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbf{R}^{n*n}$ là ma trận hệ số Coriolis và lực hướng tâm;

$\hat{\mathbf{g}}(\mathbf{q}) \in \mathbf{R}^{n*1}$ là thành phần gia tốc trọng trường;

$\mathbf{d}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbf{R}^{n*1}$ là thành phần lực ma sát và nhiễu tác động lên các khớp của robot. Trong phương trình (1) có các thành phần không được biết chính xác do tính bất định của mô hình robot. Các tham số $\hat{\mathbf{M}}(\mathbf{q})$, $\hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}})$, $\hat{\mathbf{g}}(\mathbf{q})$ có thể được mô tả như sau:

$$\hat{\mathbf{M}}(\mathbf{q}) = \mathbf{M}(\mathbf{q}) + \Delta\mathbf{M}(\mathbf{q}) \quad (2a)$$

$$\hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{B}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\mathbf{B}(\mathbf{q},\dot{\mathbf{q}}) \quad (2b)$$

$$\hat{\mathbf{g}}(\mathbf{q}) = \mathbf{g}(\mathbf{q}) + \Delta\mathbf{g}(\mathbf{q}) \quad (2c)$$

$\mathbf{M}(\mathbf{q})$, $\mathbf{B}(\mathbf{q},\dot{\mathbf{q}})$, $\mathbf{g}(\mathbf{q})$ là các thành phần được xác định chính xác.

$\Delta\mathbf{M}(\mathbf{q}), \Delta\mathbf{B}(\mathbf{q},\dot{\mathbf{q}}), \Delta\mathbf{g}(\mathbf{q})$ biểu diễn sai lệch do tính bất định của mô hình robot.

Giả thiết

$$\|\Delta\mathbf{M}(\mathbf{q})\| < \mathbf{m_0}, \|\Delta\mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\| < \mathbf{b_0}, \|\Delta\mathbf{g}(\mathbf{q})\| < \mathbf{g_0}$$

(m_0, b_0, g_0 là các giá trị được biết). Phương trình (1) có thể được biểu diễn dưới dạng:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (3a)$$

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) \quad (3b)$$

$$\boldsymbol{\tau}_0 = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (3c)$$

$$\mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) = \Delta\mathbf{M}(\mathbf{q}) + \Delta\mathbf{B}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q}) + \mathbf{d}(\mathbf{q},\dot{\mathbf{q}}) \quad (3d)$$

$\mathbf{f}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbf{R}^{n*1}$ trong (3d) là tổng hợp các thành phần không xác định của hệ động lực, ma sát, nhiễu loạn tác động lên robot và bị chặn bởi $\|\mathbf{f}\| \le f_0$ với f_0 có thể xác định được .

Mục đích của điều khiển là xác định tín hiệu điều khiển $\boldsymbol{\tau}$ sao cho robot bám được quỹ đạo $\mathbf{q_d}$ mong muốn, nghĩa là sai lệch $\mathbf{e} = (\mathbf{q} - \mathbf{q_d}) \to 0$;

$\dot{\mathbf{e}} = (\dot{\mathbf{q}} - \dot{\mathbf{q}}_\mathbf{d}) \to 0$ và $\ddot{\mathbf{e}} = (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_\mathbf{d}) \to 0$. Ở đây e, $\dot{\mathbf{e}}, \ddot{\mathbf{e}}$ là các giá trị sai số về vị trí, vận tốc và gia tốc góc.

2.2. Xây dựng thuật điều khiển robot theo nguyên lý trượt sử dụng mạng nơron RBFN

Các hệ điều khiển trượt có khả năng điều khiển tốt cho các hệ phi tuyến rất phù hợp với đối tượng là các hệ robot. Quá trình điều khiển trượt gồm hai bước: *bước một* là đưa hệ thống vào quỹ đạo của mặt trượt. *Bước tiếp theo* là duy trì hệ thống trên mặt trượt.

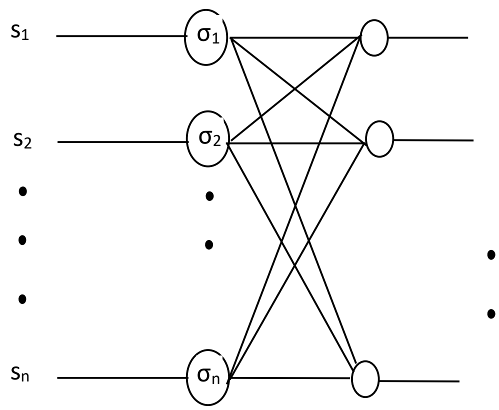
Trong trường hợp này, mặt trượt được bổ sung dưới dạng tích phân (PID):

$$\mathbf{s}(t) = \dot{\mathbf{e}} + \mathbf{C_1}\,\mathbf{e} + \mathbf{C_2}\int_0^t \mathbf{e}\,\mathbf{dt} \quad (4)$$

$\mathbf{C_1}$, $\mathbf{C_2}$ là các ma trận đường chéo xác định dương và $\mathbf{s} = [s_1, s_2, ...s_n]^T$.

Phương trình (4) cho thấy quan hệ nhất quán giữa $(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{e}, \dot{\mathbf{e}})$ và s. Do đó phương trình (3b) có thể viết dưới dạng:

$$\boldsymbol{\tau} = \boldsymbol{\tau_0} + \mathrm{f}(\mathrm{s}) \quad (5)$$



Hình 1. Mạng nơron RBF dùng để xấp xỉ các thành phần phi tuyến bất định của robot

Sự bất định của f(s) là nguyên nhân chính làm giảm chất lượng của quá trình điều khiển. Nếu chúng ta thực hiện bù thành phần phi tuyến bất định này thì chất lượng điều khiển có thể được cải thiện. Theo định lý Stone-Weiertrass [2, 3, 4] ta có thể chọn được một mạng nơron nhân tạo (ANN) với số nút nơron hữu hạn có khả năng xấp xỉ một hàm phi tuyến với độ chính xác cho trước. Để xấp xỉ hàm $\mathbf{f(s)}$ ta chọn cấu trúc mạng như sau:

$$\mathbf{f(s)} = \mathbf{W\sigma} + \boldsymbol{\varepsilon} \qquad (6a)$$

Hay:

$$\mathbf{f(s)} = \hat{\mathbf{f}} + \boldsymbol{\varepsilon} \qquad (6b)$$

trong đó: $\hat{\mathbf{f}} = \left[\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \ldots\ldots\hat{\mathbf{f}}_n\right]^{\mathbf{T}} = \mathbf{W\sigma}$ là thành phần xấp xỉ của f(s), $\mathbf{W}$ là ma trận trọng số của mạng được cập nhật online, ε là sai số bị chặn $\varepsilon \le \varepsilon_0$

Với $\|\mathbf{f(s)}\| < \mathbf{f}_0$ ta có thể xác định được giới hạn $\boldsymbol{\varepsilon}_0$ của $\boldsymbol{\varepsilon}$: $\|\boldsymbol{\varepsilon}\| \le \boldsymbol{\varepsilon}_0$.

Đặt w_i là vector cột thứ i của ma trận W ta có:

$$\hat{\mathbf{f}} = \mathbf{W\sigma} = [\mathbf{w}_1, \mathbf{w}_2, \ldots\ldots\mathbf{w}_n]\boldsymbol{\sigma} \qquad (7)$$

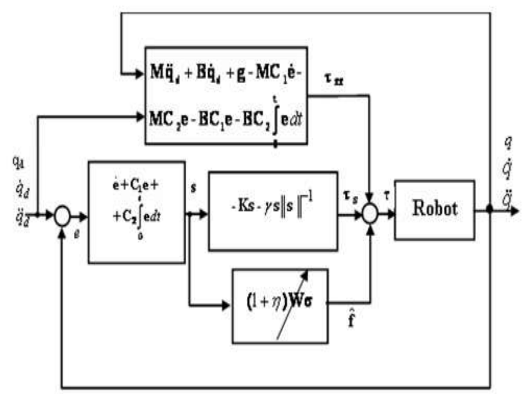
Các tác giả đã chứng minh [2]: Hệ động lực học robot *n* bậc tự do (1) với mạng nơron RBF (7) và mặt trượt (4) sẽ bám theo quỹ đạo mong muốn q_d với sai số $\mathbf{e} = (\mathbf{q} - \mathbf{q_d}) \to \mathbf{0}$ nếu ta chọn thuật điều khiển moment $\boldsymbol{\tau}$ và thuật học của mạng nơron như sau:

$$\begin{aligned}\boldsymbol{\tau} = {} & \mathbf{M\ddot{q}_d} + \mathbf{B\dot{q}_d} + \mathbf{g} - \mathbf{MC_1\dot{e}} - \mathbf{MC_2e} - \mathbf{BC_1e} \\ & - \mathbf{BC_2}\int_0^t \mathbf{e}\,dt - \mathbf{Ks} - \gamma\mathbf{s}\|\mathbf{s}\|^{-1} + (1+\eta)\mathbf{W\sigma}\end{aligned} \qquad (8)$$

$$\dot{w}_i = -\eta s \sigma_j \qquad (9)$$

trong đó các tham số tự chọn $\mathbf{K} = \mathbf{K}^{\mathbf{T}} > \mathbf{0}$ là ma trận đối xứng xác định dương, $\eta, \gamma > 0$.

Cấu trúc của hệ điều khiển có thể mô tả theo sơ đồ trên hình 2:



Hình 2: Sơ đồ hệ điều khiển trượt, sử dụng RBFN bù các thành phần bất định của robot

Moment $\boldsymbol{\tau}$ gồm ba thành phần chính: thành phần bù phi tuyến

$$\boldsymbol{\tau}_{ff} = \mathbf{M\ddot{q}_d} + \mathbf{B\dot{q}_d} + \mathbf{g} - \mathbf{MC_1\dot{e}} - \mathbf{MC_2e} - \mathbf{BC_1e} - \mathbf{BC_2}\int_0^t \mathbf{e}\,dt,$$

thành phần trượt $\boldsymbol{\tau}_s = -\mathbf{Ks} - \gamma\,\mathbf{s}\|\mathbf{s}\|^{-1}$ và thành phần xấp xỉ $\hat{\mathbf{f}}$ là mạng RBF có thuật học online.

Tốc độ hội tụ của mạng nơron phụ thuộc vào thuật học (8), trong đó có sự tham gia của mặt trượt s và hệ số học η. Để cho quá trình học hội tụ nhanh, đảm bảo các chỉ tiêu chất lượng điều chỉnh như độ quá chỉnh, số lần dao động .v.v. chúng ta có thể lựa chọn một hệ số học η tối ưu bằng cách sử dụng thuật toán BAT.

3. ĐIỀU KHIỂN ROBOT VỚI RBFN CÓ HỆ SỐ HỌC DƯỢC TỐI ƯU BẰNG THUẬT TOAN DƠI – BAT

3.1. Thuật toán dơi – BAT

Theo thuật toán BAT, ba quy tắc hoặc ý tưởng sau đây [5, 6, 7] được sử dụng:

- Tất cả các con dơi sử dụng phát âm thanh để phân biệt khoảng cách, đồng thời phân biệt được thức ăn, con mồi và các vật cản;

- Các con dơi bay ngẫu nhiên với vận tốc v_i tại vị trí x_i. Chúng có thể tự điều chỉnh tần số Q (hoặc bước sóng) của các xung phát ra và điều chỉnh độ rộng của xung $r\in[0,1]$, phụ thuộc vào khoảng cách đến mục tiêu;

- Giả định rằng độ lớn của tiếng vọng thay đổi từ giá trị tối đa A_0 đến giá trị tối thiểu A_{min}.

Tại một vị trí x_i^t ở vòng lặp thứ t, trong một không gian tìm kiếm d chiều mỗi con dơi sẽ có một vận tốc v_i^t. Trong tổng số các con dơi, sẽ có một con có vị trí tốt nhất x_*. Do đó, ba quy tắc trên có thể được chuyển thành các phương trình cập nhật cho vị trí x_i^t và vận tốc v_i^t như sau:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \qquad (10)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*) f_i \qquad (11)$$

$$x_i^t = x_i^{t-1} + v_i^{t-1} \qquad (12)$$

trong đó β ∈ [0, 1] là một vector ngẫu nhiên từ một phân phối đồng nhất.

Có thể sử dụng bước sóng hoặc tần số để triển khai thuật toán BAT, giả sử dải tần số nằm trong khoảng $f_{\min} = 0$ và $f_{\max} = O(1)$. Đầu tiên, mỗi con dơi được gán ngẫu nhiên một tần số theo phân phối đều trong khoảng $[f_{\min}, f_{\max}]$. Vì lý do này, thuật toán dơi có thể được coi là một thuật toán điều chỉnh tần số để cung cấp một sự kết hợp cân bằng giữa việc phát xung và thu nhận tiếng vọng. Tiếng vọng thu được và tần số phát xung cho phép tập trung vào khu vực có con mồi [7].

Để có được một cơ chế hiệu quả giữa việc phát xung và thu nhận tiếng vọng và khi nào thì chuyển sang giai đoạn phát hiện được con mồi. Cần phải biến đổi tiếng vọng A_i và tỉ lệ phát xung r_i trong các vòng lặp. Âm lượng của tiếng vọng được chọn trong dải từ A_{min} đến A_{max} sẽ giảm đi khi con dơi tìm thấy mồi, khi đó tỉ lệ phát xung tăng lên, khi $A_{min} = 0$ nghĩa là con dơi đã tìm thấy mồi và tạm dừng phát xung. Với những giả định này, ta có:

$$A_i^{t+1} = \alpha A_i^t \qquad (13)$$

Và

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \qquad (14)$$

Ở đây α và γ là hằng số, α tương tự như hệ số làm mát của một quá trình nung chảy, với bất kỳ $0 < \alpha < 1$ và $\gamma > 0$ ta có:

$A_i^t \to 0$ và $r_i^t = r_i^0$ khi $t \to \infty$ trong trường hợp đơn nhất có thể chọn $\alpha = \gamma$, khi mô phỏng thường chọn $\alpha = 0.9$ và $\gamma = 0.98$.

Thuật toán dơi tiêu chuẩn đảm bảo tối ưu hóa liên tục, có thể mở rộng thuật toán dơi tiêu chuẩn thành thuật toán dơi nhị phân (BBA). Chuyển đổi các vị trí giá trị liên tục của các con dơi thành giá trị nhị phân bằng cách sử dụng hàm sigmoid như sau:

$$S(x_i^t) = \frac{1}{1+\exp[-x_i^t]} \qquad (15)$$

Dẫn đến:

$$x_i^t = \begin{cases} 1 & if \quad S(x_i^t) > \sigma \\ 0 & otherwise \end{cases} \qquad (16)$$

σ là một biến ngẫu nhiên được lấy từ một phân bố đều U(0, 1).

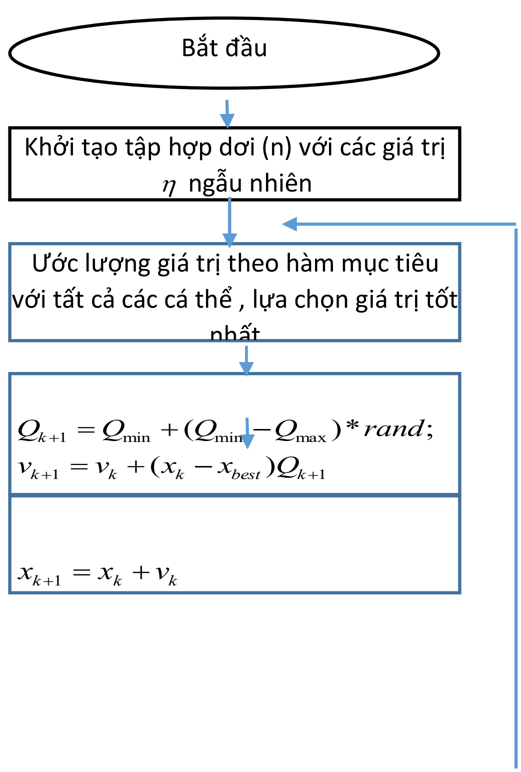
3.2. Sử dụng BAT tối ưu hệ số học RBFN trong bài toán điều khiển robot

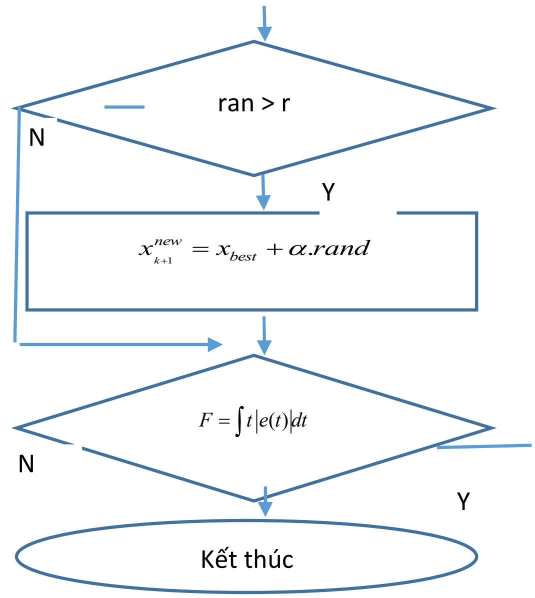
Sử dụng BAT để xác định hệ số học η tối ưu, theo các chỉ tiêu chất lượng quá trình quá độ của hệ điều khiển robot như sau:

Bảng 1: Chỉ tiêu của quá trình quá độ

Thời gian thiết lập (T_C)	≤ 3 sec
Độ quá chỉnh (O_C)	$\leq 20\%$ giá trị thiết lập (Q_c)
Số lần dao động (N)	≤ 4

Các bước sử dụng BAT để tối ưu hệ số học η như sau:





Hình 3: Thuật toán BAT

1. Khởi tạo quần thể dơi (n) chúng có vị trí (x_i) và vận tốc (v_i) ngẫu nhiên tham số η;

2. Tính giá trị hàm mục tiêu của tất các cá thể dơi trong quần thể vừa khởi tạo.

3. So sánh các giá trị của hàm mục tiêu vừa tìm được để tìm ra con dơi (x_{best}) cho giá trị hàm mục tiêu tốt nhất.

4. Cập nhật lại tần số xung Q và vận tốc v của tất cả các con dơi theo phương trình sau:

$$Q_{k+1} = Q_{\min} + (Q_{\min} - Q_{\max}) * rand;$$
$$v_{k+1} = v_k + (x_k - x_{best})Q_{k+1}$$

5. Cập nhật lại vị trí của tất cả các con dơi theo phương trình sau:

$$x_{k+1} = x_k + v_k$$

6. Cập nhật lại vị trí của cá thể dơi nếu độ rộng xung (r) nhỏ hơn độ rộng xung tín hiệu nhỏ hơn độ rộng xung tín hiệu tạo ra ngẫu nhiên (*rand*).

$$\text{if}(rand > r)$$
$$x_{k+1}^{new} = x_{best} + \alpha.rand$$

7. Kiểm tra điều kiện $(rand < A \,\&\, f(x_{k+1}^{new}) < f(x_{k+1}))$ chấp nhận quần thể mới và tăng độ rộng xung (r) và giảm tiếng vọng (A) .

8. Kiểm tra nếu giá trị tốt nhất của vị trí mới trong quần thể dơi nhỏ hơn giá trị yêu cầu thì kết thúc thuật toán, còn không lặp lại bước thứ 4.

Hàm mục tiêu của thuật toán BAT được lựa chọn là hàm ITAE :

$$ITAE: \quad F = \int t\left|e(t)\right|dt$$

Lưu đồ thuật toán tối ưu tham số bộ điều khiển robot dựa trên thuật toán BAT được thể hiện như sau:

4. Mô phỏng BAT tối ưu hệ số học cho RBFN điều khiển robot hai bậc tự do

Với mô hình robot có phương trình động lực học như sau:

$$\hat{\mathbf{M}}(\mathbf{q}) = \begin{bmatrix} m_1 l_{g1}^2 + I_1 + I_2 + \\ +m_2(l_1^2 + l_{g2}^2 + 2l_1 l_{g2}\cos q_2) & m_2(l_{g2}^2 + l_1 l_{g2}\cos q_2) + I_2 \\ m_2(l_{g2}^2 + l_1 l_{g2}\cos q_2) + I_2 & m_2 l_{g2}^2 + I_2 \end{bmatrix}$$

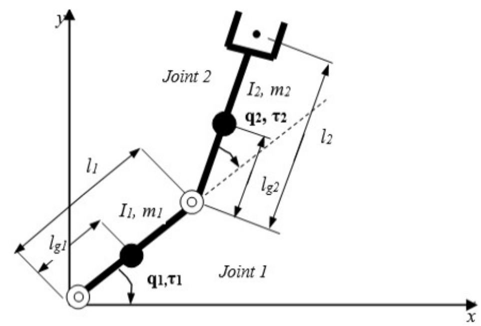
$$\hat{\mathbf{g}}(\mathbf{q}) = \begin{pmatrix} (m_1 l_{g1} + m_2 l_1) g\cos(q_1) + m_2 l_{g2} g\cos(q_1 + q_2) \\ m_2 l_{g2} g\cos(q_1 + q_2) \end{pmatrix}$$

$$\hat{\mathbf{B}}(\mathbf{q},\dot{\mathbf{q}}) = m_2 l_1 l_{g2}\begin{pmatrix} -2\dot{q}_2 & -\dot{q}_2 \\ \dot{q}_1 & 0 \end{pmatrix}\sin q_2$$

Với $m_i = m_{mi} + m_{li}$;

Bảng 2. Các tham số của robot phẳng hai bậc tự do

	Khớp thứ nhất	Khớp thứ hai
Khối lượng khớp m_{li} [kg]	50.0	50.0
Khối lượng của động cơ m_{mi} [kg]	5.0	5.0
Momen quán tính của khớp I_i [kg.m^2]	10.0	10.0
Độ dài của khớp l_i [m]	1	1
Khoảng cách đến trọng tâm của khớp l_{gi} [m]	0.5	0.5



Hình 4. Mô hình robot hai bậc tự do

Với robot khớp cứng hai bậc tự do, mạng RBFN có cấu trúc như hình 5:

Đầu tiên đặt $\eta = 200$, ta chọn các tham số điều khiển cho moment $\boldsymbol{\tau}$ và thuật chỉnh trọng mạng nơron $\dot{\mathbf{w}}_i$ cho robot như (8), (9). Ma trận $\mathbf{K}$ và hệ số γ trong phương trình (8) được chọn như sau: $\mathbf{K} = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$ $\gamma = 20$; ;

Trạng thái ban đầu của robot là: $q_{10} = -0.4;\ q_{20} = 1.85$.

Với các tham số hàm Gauss của mạng nơron: $\lambda_1 = 1;\ \lambda_2 = 2;\quad c_1 = 0.01$; $c_2 = 0.02$;

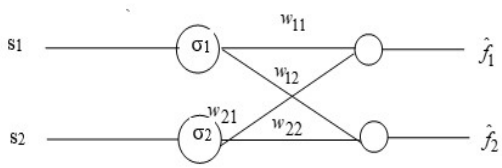
Độ bất định của robot được chọn tới 30% giá trị thật:

$$\Delta\mathbf{M} = 30\%\mathbf{M}; \Delta\mathbf{B} = 30\%\mathbf{B}; \Delta\mathbf{g} = 30\%\mathbf{g}$$

.

Thành phần ma sát và nhiễu loạn được giả thiết:

$$\mathbf{d(q,\dot{q})} = \mathbf{d(t)} = \begin{pmatrix} 3\sin(20t) + 1 + 5\dot{q}_1 \\ \cos(20t) + 3\dot{q}_2 \end{pmatrix}$$



Hình 5. Cấu trúc mạng RBF để bù các thành phần bất định của robot

Ma trận $\mathbf{C_1}$, $\mathbf{C_2}$ của mặt trượt được chọn là: $\mathbf{C_1} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}; \mathbf{C_2} = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix}$.

Các momen của các khớp có giá trị chặn trên là +1500(N.m) và chặn dưới là: -1500(N.m).

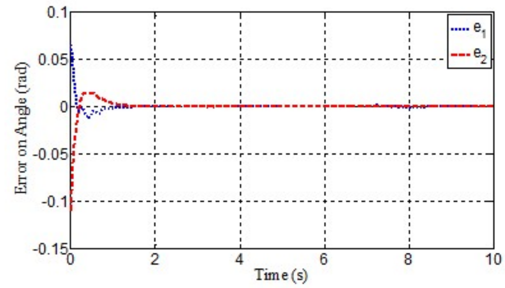
Tốc độ biến thiên lớn nhất và nhỏ nhất của các momen là: ± 800 (N.m/s).

Cho quỹ đạo mong muốn $q_{1d}(t)$ và $q_{2d}(t)$ khi tay nắm của robot di chuyển từ điểm đầu A(1.5; 0.8) đến điểm cuối B(0.8; 0.1) theo đường tròn tâm C(0.8; 0.8) ngược chiều kim đồng hồ trong thời gian T theo quy luật tốc độ góc hình thang với gia tốc góc β cho trước.

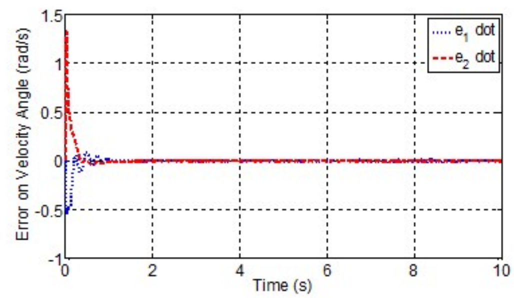
α(t) là góc trên cung AB. α(0) = 0. α(T) = 3π/2

$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0.8 \end{pmatrix} + 0.7\begin{pmatrix} \cos\alpha \\ \sin\varepsilon \end{pmatrix}$ là quỹ đạo chuyển động mong muốn của robot.

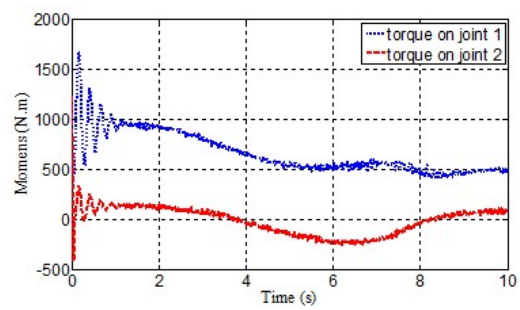
Kết quả mô phỏng như sau:



Hình 6a. Sai lệnh vị trí góc của khớp 1 và khớp 2 trong không gian trục



Hình 6b. Sai lệnh vận tốc góc của khớp 1 và khớp 2 trong không gian trục

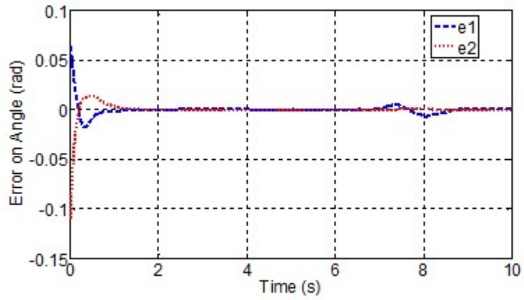


Hình 6c. Biểu diễn của moomen tác động lên khớp 1 và khớp 2

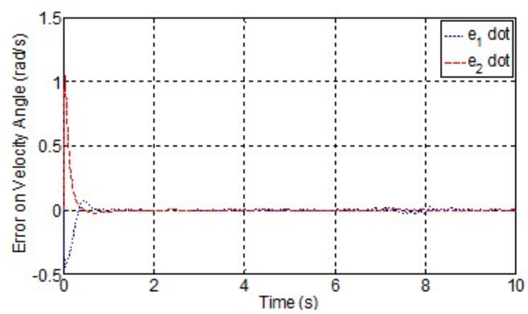
Để nâng cao chất lượng điều khiển, chúng ta có thể tối ưu hệ số học η bằng BAT.

Hàm mục tiêu của thuật toán BAT được lựa chọn là hàm ITAE :

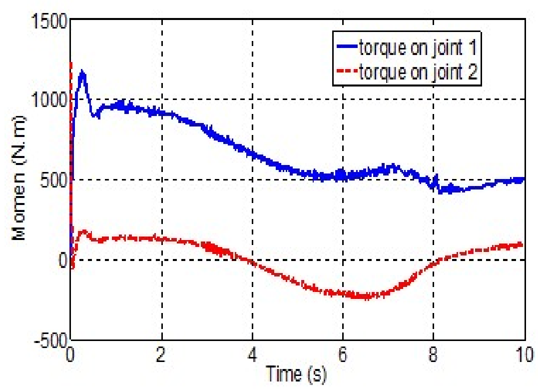
Kết quả mô phỏng như sau:



Hình 7a. Sai lệnh vị trí góc của khớp 1 và khớp 2 trong không gian trục



Hình 7b. Sai lệnh vận tốc góc của khớp 1 và khớp 2 trong không gian trục



Hình 7c. Biểu diễn của moomen tác động lên khớp 1 và khớp 2

Nhận xét: So sánh kết quả thu được trên hình 6 và hình 7 ta thấy sai lệch về vị trí và vận tốc góc trong 2 trường hợp không có sự thay đổi lớn, nhưng biểu diễn của momen tác động lên khớp thì có sự thay đổi. Khi hệ số học η chưa được tối ưu, yêu cầu của momen là vượt ngưỡng (τ_1 >1500 N.m). Sử dụng BAT xác định được hệ số học η tối ưu thì (τ_1 <1500 N.m) nằm trong dải cho phép. Rõ ràng chất lượng điều khiển đã được cải thiện.

5. KẾT LUẬN

Mô hình điều khiển robot theo nguyên lý trượt sử dụng RBFN đảm bảo quá trình học online trên toàn bộ không gian trạng thái. Kết quả cho thấy hệ thống đảm bảo hội tụ và sai lệch tiến đến không.

Tiếp theo, sử dụng BAT để tìm hệ số học tối ưu cho phép chúng ta khảo sát, đánh giá được ảnh hưởng của hệ số học đến quá trình hội tụ của thuật học đồng thời nâng cao hơn nữa chất lượng của quá trình điều khiển. Đây là một bước tiến trong quá trình nghiên cứu ứng dụng mạng nơron để xây dựng các bộ điều khiển robot vừa đảm bảo tính hội tụ và ổn định đồng thời nâng cao hơn chất lượng điều khiển.

TÀI LIỆU TRÍCH DẪN

[1] Nguyen Tran Hiep, Nguyen X Chiem (2023) - Optimization of Sliding Mode Control Parameters for Manipulator Robots Using the BAT Algorithm, The International Conference on Power, Energy and Cybernetics 2023, December 16-17, 2023, Hai Duong, Vietnam.

[2] Nguyen Tran Hiep, Pham Thuong Cat - *Robust Neural Sliding Mode Control of Robot Manipulator*s – Proceeding of 2nd Mediterrannean Conference on Intelligent Systems and Automation, March 23-25 2009, Zarzis, Tunisia. Page 210 -215 (API Conference Proceedings 1107).

[3] Nguyen Tran Hiep, Pham Thuong Cat - *Robot control nDOF by integrated sliding surface and approximating neural networks* – Proceeding of European Control Conreference 2009, 23 - 26 August 2009 Budapest Hungary. Page 2187-2192.

[4] Yun Li and Alexander HauBler - University of Glasgow, UK- *Artificial evolution of neural networks and its application to feedback control*. Artifical Intelligence In Engineering, 10(2), pp 143 – 152, February.

[5] Xin-She Yang, "Nature-Inspired Optimization Algorithms", Elsevier First edition, 2014, 258 page.

[6] Xin-She Yang, "Bat Algorithm: Literature Review and Applications", Int. J. Bio-Inspired Computation, Vol. 5, No. 3, 2013, page: 141-149.

[7] Nguyen, X. C., Hoang, D. L., Pham, X. T., Nguyen, T. H., Le, M. K., Nguyen, V. X. (2021), Synthesis of feedback linearization controller with parameters optimization based on bat algorithm for a magnetic levitation system, Journal of Science and Technique, 16(03), 123 - 139. https://doi.org/10.56651/lqdtu.jst.v16.n03.283.