

Tối ưu hóa tham số bộ điều khiển trượt cho robot song song phẳng

Optimization of sliding mode controller parameters for planar parallel robot

Thi-Van-Anh Nguyen¹, Thanh-Long Nguyen¹ and Manh-Linh Nguyen^{1*}

¹Trường Điện-Điện tử, Đại học Bách khoa Hà Nội

*Corresponding author E-mail: linh.nguyenmanh@hust.edu.vn

Abstract

In this paper, we present a method for optimizing the parameters of a Sliding Mode Controller (SMC) using the Particle Swarm Optimization (PSO) algorithm. The SMC is well-known for its robust performance in handling system uncertainties and external disturbances. However, selecting the optimal parameters for the SMC plays a crucial role in achieving high performance in trajectory tracking tasks. Manual tuning often leads to suboptimal solutions, especially for complex nonlinear systems. To address this issue, we apply the PSO algorithm, a powerful parameter optimization technique inspired by the swarm behavior of birds or fish, to automatically search for the optimal control parameters. The optimized SMC parameters are then applied to the trajectory tracking control of a planar parallel robot. Simulation results, verified using Matlab Simulink, demonstrate high performance in trajectory tracking for the planar parallel robot.

Keywords: Điều khiển trượt; tối ưu hóa bầy đàn; robot song song phẳng; điều khiển bám quỹ đạo.

Tóm tắt

Trong bài báo này, chúng tôi đề xuất một phương pháp mới để tối ưu hóa các tham số của bộ điều khiển trượt bằng cách sử dụng thuật toán tối ưu hóa bầy đàn. Bộ điều khiển trượt được biết đến rộng rãi nhờ khả năng bền vững với những yếu tố bất định hệ thống, do đó đặc biệt hiệu quả đối với các đối tượng phi tuyến. Để đạt được hiệu quả cần thiết, việc lựa chọn các tham số bộ điều khiển sao cho tối ưu là rất quan trọng. Các phương pháp điều chỉnh thủ công thường khá mất thời gian và khó có thể đạt được kết quả tối ưu, đặc biệt đối với các hệ thống phi tuyến phức tạp. Để giải quyết vấn đề này, chúng tôi áp dụng thuật toán PSO, một công cụ mạnh mẽ dựa trên hành vi bầy đàn, để tự động xác định các tham số của bộ điều khiển trượt thỏa mãn một hàm mục tiêu cho trước. Phương pháp tiếp cận này được áp dụng cho mô hình robot song song phẳng nhằm đạt được độ chính xác cao trong bám quỹ đạo. Tính đúng đắn và hiệu quả của phương pháp đề xuất được kiểm chứng thông qua mô phỏng.

1. Giới thiệu

Robot công nghiệp là một hệ chuyển động phức hợp nhiều bậc tự do có khả năng lập trình được, do đó được sử dụng rộng rãi trong nhiều ứng dụng khác nhau [1]. Việc sử dụng hệ thống robot công nghiệp đã trở nên phổ biến trong các doanh nghiệp sản xuất trên toàn cầu. Nó được coi là giải pháp tự động hóa hiệu quả nhất hiện nay, mang lại lợi thế cạnh tranh cho các doanh nghiệp trong lĩnh vực sản xuất. Có một sự nhấn mạnh rõ ràng vào việc nâng cao độ chính xác vị trí trong quá trình phát triển của robot công nghiệp [2] [3].

Trong thực tế, robot được chế tạo theo rất nhiều cấu hình khác nhau để phù hợp với từng tác vụ nhất định. Các robot nối tiếp với tầm hoạt động rộng, được sử dụng rộng rãi trong công

nh nghiệp sản xuất ô tô, lắp ráp, phân loại sản phẩm, công nghệ hàn... Trong khi đó, robot song song phẳng với những ưu điểm như độ cứng cao, quán tính nhỏ và tỷ lệ tải trọng trên trọng lượng lớn, phù hợp với các ứng dụng đòi hỏi độ chính xác và tốc độ cao, đã thu hút sự chú ý của nhiều nghiên cứu gần đây [4] [5]. Tuy nhiên, loại robot này lại có cấu trúc động học và động lực học tương đối phức tạp, không gian làm việc nhỏ và ẩn chứa nhiều ràng buộc gây khó khăn cho việc xây dựng hệ điều khiển [6].

Để giải quyết bài toán điều khiển chuyển động robot song song phẳng, phương án điều khiển kinh điển dựa trên mô hình như điều khiển động học ngược [17] thường gặp khó khăn do ảnh hưởng của các yếu tố bất định như sai lệch mô hình và sự thay đổi của tải trọng. Các phương pháp điều khiển thông minh như mạng nơ-ron [18] và logic mờ [19] cũng đã được nghiên cứu, nhưng yêu cầu năng lực tính toán cao và phụ thuộc vào dữ liệu huấn luyện. Điều khiển trượt (SMC) là một phương pháp hiệu quả trong việc xử lý các đối tượng phi tuyến ẩn chứa các yếu tố bất định. Và trong mọi trường hợp, việc tối ưu hóa tham số của bộ điều khiển luôn đóng vai trò quyết định đối với chất lượng của hệ điều khiển. Do đó, các nghiên cứu liên quan đến tối ưu hóa tham số bộ điều khiển luôn thu hút số lượng lớn các nhà nghiên cứu.

Việc tối ưu hóa tham số bộ điều khiển thường được thực hiện thông qua việc giải bài toán tối thiểu hóa một hàm chi phí cụ thể. Các nghiên cứu trước đây đã áp dụng các phương pháp tối ưu hóa cổ điển như Gradient Descent hay Newton-Raphson, tuy nhiên, các phương pháp này gặp khó khăn với đối tượng phi tuyến. Gần đây, các thuật toán tiến hóa đã được sử dụng rộng rãi để khắc phục những hạn chế này, nhờ khả năng tìm kiếm trong không gian phức tạp và tránh rơi vào cực trị địa phương. Thuật toán di truyền (Genetic Algorithm - GA) [20] là

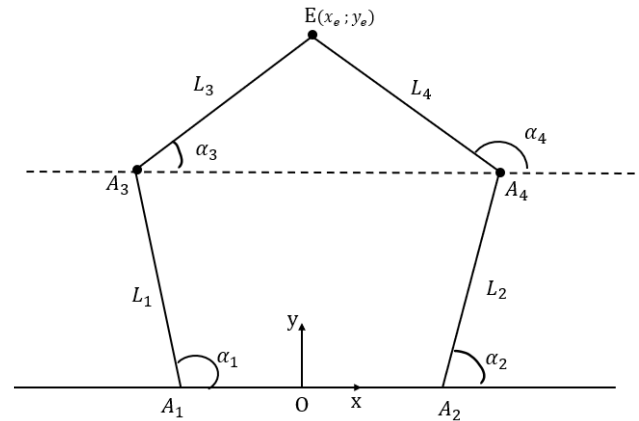
một trong những phương pháp đầu tiên được áp dụng cho bài toán tối ưu tham số điều khiển. GA đã chứng minh được hiệu quả trong việc tìm kiếm các tham số cho SMC, giúp giảm hiện tượng chattering và nâng cao độ chính xác [21]. Tuy nhiên, GA thường yêu cầu thời gian tính toán lớn, đặc biệt với các hệ thống phức tạp như robot song song. Ngoài GA, các thuật toán tương tự khác như tối ưu hóa kiểu đàn kiến (Ant Colony Optimization - ACO) [22], tiến hóa vi phân (Differential Evolution - DE) [23], tối ưu hóa đàn dơi (Bat Algorithm) [24] cũng đã được áp dụng để tối ưu hóa tham số điều khiển. Với hàm chi phí ẩn chứa nhiều mục tiêu điều khiển như chất lượng động học [7] [8], mức độ tiêu thụ năng lượng [9], độ cứng và không gian làm việc [10], sai lệch quỹ đạo chuyển động và mô men xoắn [11], cũng như đồng thời cả động học và động lực học [12] [13]. Các thuật toán tiến hóa đa mục tiêu (MOEAs) thường được sử dụng theo cách tạo ra một tập hợp các giải pháp mang tính chất thỏa hiệp, trong đó việc cải thiện bất kỳ tiêu chí nào trong hàm chi phí sẽ dẫn đến sự đánh đổi với ít nhất một tiêu chí khác. Các kỹ thuật như tối ưu MOEAs [14], thuật giải di truyền đa mục tiêu (MOGA) [15], và các thuật toán tiến hóa vi phân (DE) [16] đã được áp dụng trong thiết kế tối ưu của robot song song.

Một trong những phương pháp nổi bật nhất trong những năm gần đây là tối ưu hóa bầy đàn (Particle Swarm Optimization - PSO), với cơ chế mô phỏng hành vi xã hội của bầy đàn, cho phép tìm kiếm giải pháp hiệu quả trong không gian tham số phức tạp. Với nhiều lợi thế như tính đơn giản, dễ triển khai, PSO là một công cụ đa năng để giải quyết bài toán tối ưu trong nhiều lĩnh vực [25] [26]. Trong nghiên cứu này, PSO đã được áp dụng để tối ưu hóa các tham số điều khiển của bộ SMC cho robot song song phẳng 2 bậc tự do, từ đó cải thiện đáng kể chất lượng của hệ thống điều khiển chuyển động. Cụ thể, các đóng góp chính của bài báo bao gồm:

- Xây dựng và phân tích đầy đủ mô hình động lực học của hệ robot, bao gồm các phương trình chuyển động, ma trận Jacobian và các yếu tố ảnh hưởng đến động lực học như momen quán tính và lực tác động. Mô hình này tạo nền tảng cho việc thiết kế và kiểm chứng các thuật toán điều khiển, đảm bảo khả năng ứng dụng linh hoạt trong các bài toán thực tế.
- Áp dụng PSO để tối ưu hóa các tham số của bộ SMC trong hệ thống robot song song phẳng 2 bậc tự do
- Cung cấp một giải pháp điều khiển hiệu quả có thể áp dụng rộng rãi không chỉ cho các hệ thống robot mà cả các hệ thống điều khiển khác

2. Mô hình Robot

Robot song song phẳng 2 bậc tự do (2DOF-PPR) với cấu trúc đối xứng, là đối tượng chính của nghiên cứu này, gần đây đã được phát triển thành công dưới dạng nguyên mẫu [27], [28]. Cấu trúc chi tiết của 2DOF-PPR được minh họa trong Hình 1, bao gồm hai chuỗi động học giống hệt nhau, mỗi chuỗi gồm khớp chủ động (A_1 và A_2), khớp thụ động (A_3 và A_4), cùng với các thanh liên kết cứng có độ dài L_1 , L_2 , L_3 , và L_4 . Để phân tích động học của 2DOF-PPR, một hệ tọa độ quán tính Oxy được đặt tại trung điểm của hai khớp chủ động A_1 và A_2 . Khi đó, tọa độ của cơ cấu tác động cuối E được xác định tại điểm (x_e, y_e) .



Hình 1: Mô hình của robot song song hai bậc tự do

2.1. Mô hình động học của robot song song phẳng 2 bậc tự do

Giả sử $\mathbf{E} = [x_e \ y_e]^T$ là vị trí vectơ tác động cuối của robot trên hệ tọa độ quán tính Oxy trong đó $OA_1 = OA_2$. Vị trí của các khớp thụ động A_i (với $i = 3, 4$) là:

$$\begin{aligned} \mathbf{A}_3 &= [L_1 \cos(\alpha_1) - L_3 \quad L_1 \sin(\alpha_1)]^T \\ \mathbf{A}_4 &= [L_2 \cos(\alpha_2) + L_3 \quad L_2 \sin(\alpha_2)]^T \end{aligned} \quad (1)$$

Trong đó, α_1 và α_2 là các góc quay của các khớp chủ động. Phép toán động học ngược được thiết lập bằng cách xem xét các ràng buộc động học sau:

$$|\mathbf{A}_3\mathbf{E}| = L_3, |\mathbf{A}_4\mathbf{E}| = L_4; \quad (2)$$

Từ (1) và (2) dẫn tới:

$$(x_e - L_1 \cos(\alpha_1) + OA_1)^2 + (y_e - L_1 \sin(\alpha_1))^2 = L_3^2 \quad (3)$$

$$(x_e - L_2 \cos(\alpha_2) - OA_1)^2 + (y_e - L_2 \sin(\alpha_2))^2 = L_4^2 \quad (4)$$

Hai phương trình này cho phép chúng ta xác định các góc α_1 và α_2 cần thiết để đạt được vị trí mong muốn của cơ cấu tác động cuối.

2.2. Ma trận Jacobian

Ma trận Jacobian của robot được tìm ra bằng cách lấy đạo hàm theo thời gian của các phương trình (3) và (4), dẫn đến:

$$\mathbf{G}\dot{\alpha} = \dot{\mathbf{H}}\mathbf{E} \quad (5)$$

Giải phương trình (5) dễ dàng thu được:

$$\mathbf{J} = \mathbf{G}^{-1}\mathbf{H} \quad (6)$$

Với:

$$\begin{aligned} \dot{\mathbf{E}} &= [\dot{x}_e \ \dot{y}_e]^T, \dot{\alpha} = [\dot{\alpha}_1 \ \dot{\alpha}_2]^T \\ \mathbf{G} &= \begin{bmatrix} y_e c_{\alpha_1} - (x_e + OA_1) s_{\alpha_1} & 0 \\ 0 & y_e c_{\alpha_2} + (OA_1 - x_e) s_{\alpha_2} \end{bmatrix} \\ \mathbf{H} &= \begin{bmatrix} x_e + OA_1 - L_1 c_{\alpha_1} & y_e - L_1 s_{\alpha_1} \\ x_e - OA_1 - L_2 c_{\alpha_2} & y_e - L_2 s_{\alpha_2} \end{bmatrix} \end{aligned} \quad (7)$$

Trong đó $\cos(\alpha_i) = c_{\alpha_i}$ và $\sin(\alpha_i) = s_{\alpha_i}$, $i = (1, 2)$

2.3. Không gian làm việc

Không gian làm việc lý thuyết của 2DOF-PPR được định nghĩa là khu vực mà khâu tác động cuối có thể tiếp cận khi các khớp động xoay góc $0 < \alpha_i < 2\pi$. Trong định nghĩa về không gian làm việc lý thuyết này, các va chạm tiềm tàng giữa các liên kết và các vị trí kỳ dị, nơi ma trận Jacobian trở nên kỳ dị, không được tính đến. Do đó, không gian làm việc lý thuyết được giới hạn bởi hai vòng tròn giới hạn, được xác định dựa trên chiều dài của chuỗi động học đầu tiên:

$$I_{1o} : (x + OA_1)^2 + y^2 = (L_1 + L_3)^2 \quad (8)$$

$$I_{1i} : (x + OA_1)^2 + y^2 = (L_1 - L_3)^2 \quad (9)$$

Trong đó I_{1o} và I_{1i} được định nghĩa là các vòng tròn giới hạn ngoài và trong.

Ngoài ra, khi xem xét chiều dài của chuỗi động học thứ hai, không gian làm việc lý thuyết được bao quanh bởi các vòng tròn giới hạn sau:

$$I_{2o} : (x - OA_1)^2 + y^2 = (L_4 + L_2)^2 \quad (10)$$

$$I_{2i} : (x - OA_1)^2 + y^2 = (L_4 - L_2)^2 \quad (11)$$

Khu vực được bao quanh bởi các vòng tròn giới hạn được xác định bởi các phương trình (8), (9), (10), và (11) đại diện cho không gian làm việc lý thuyết.

Ngược lại, không gian làm việc sử dụng được định nghĩa là vùng liên tục lớn nhất trong không gian làm việc lý thuyết mà không có bất kỳ vị trí kỳ dị nào. Do đó, không gian làm việc sử dụng được giới hạn bởi các vị trí kỳ dị bên ngoài, như đã được mô tả trước đó bởi [29]. Để đánh giá thêm về không gian làm việc sử dụng, vòng tròn tối đa bên trong (MIC) được coi là một chỉ số hữu ích. MIC là vòng tròn lớn nhất có thể được nội suy trong không gian làm việc sử dụng và tiếp xúc với các vị trí kỳ dị được mô tả bởi I_{1s} [28]. Không gian làm việc tối đa bên trong (MIW) được định nghĩa là khu vực bao quanh vòng tròn MIC này. Vòng tròn MIC, được biểu diễn như một hàm của các chiều dài liên kết, được cho bởi công thức sau:

$$x^2 + (y - y_{MIC})^2 = L_{MIC}^2 \quad (12)$$

Trong đó L_{MIC} là bán kính và (x_{MIC}, y_{MIC}) là tâm của vòng tròn MIC. Đối với các trường hợp mà $L_i + L_j < OA_1$ với $i = 1; 2$ và $j = 3; 4$, vòng tròn MIC được định nghĩa bởi:

$$L_{MIC} = \frac{L_1 + L_2 - |L_1 - L_2|}{2}$$

$$x_{MIC} = 0 \quad (13)$$

$$y_{MIC} = \sqrt{\frac{(L_1 + L_2 + |L_1 - L_2|)^2}{4} - L_3^2}$$

Đối với các trường hợp mà $L_i + L_j > OA_1$ với $i = 1; 2$ và $j = 3; 4$, bán kính và tâm của vòng tròn MIC được định nghĩa bởi:

$$L_{MIC} = |y_{MIC}| - y_{col}$$

$$x_{MIC} = 0 \quad (14)$$

$$y_{MIC} = \frac{(L_1 + L_2 + y_{col})^2 - L_3^2}{2(L_1 + L_2 + y_{col})}$$

Trong đó:

$$y_{col} = \sqrt{L_i^2 - (L_j - OA_1)^2} \quad (15)$$

và $i = 1; 2$ và $j = 3; 4$.

2.4. Mô hình động lực học của 2DOF-PPR

Bước đầu tiên trong việc xây dựng các phương trình chuyển động là lựa chọn "hệ thống tự do." Trong kịch bản này, robot được tưởng tượng là bị cắt rời tại điểm tác động cuối, tạo ra hai robot nối tiếp, mỗi robot có hai bậc tự do như minh họa trong Hình 2. Đặt m_i là khối lượng, d_i biểu thị khoảng cách đến trung tâm của liên kết, và L_i cho biết chiều dài của liên kết i . Momen quán tính của liên kết i đối với đường thẳng đi qua tâm khối lượng song song với trục quay được biểu thị bằng C_i . Mặc dù các tham số cho liên kết 1 và liên kết 3 được định nghĩa tương tự như đối với liên kết 2 và liên kết 4, nhưng chúng không được minh họa trong hình. Do đó, các phương trình ràng buộc phát sinh từ yêu cầu rằng điểm C trùng với điểm D , điều này có thể được diễn đạt như sau:

$$\varphi(\alpha') = \begin{bmatrix} \varphi(1) \\ \varphi(2) \end{bmatrix} = \mathbf{0} \quad (16)$$

Trong đó:

$$\begin{aligned} \varphi(1) &= L_1 \cos(\alpha_1) + L_3 \cos(\alpha_1 + \alpha_3) \\ &\quad - L_2 \cos(\alpha_2) - L_4 \cos(\alpha_2 + \alpha_4) \\ \varphi(2) &= L_1 \sin(\alpha_1) + L_3 \sin(\alpha_1 + \alpha_3) \\ &\quad - L_2 \sin(\alpha_2) - L_4 \sin(\alpha_2 + \alpha_4) \end{aligned} \quad (17)$$

Đặt $\alpha' = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4]^T$ là vectơ tọa độ tổng quát của hệ thống tự do. Lưu ý rằng vì các khớp được kích hoạt là khớp 1 và 2 cho 2DOF-PPR, nên vectơ tọa độ tổng quát của hệ thống bị ràng buộc được chọn là:

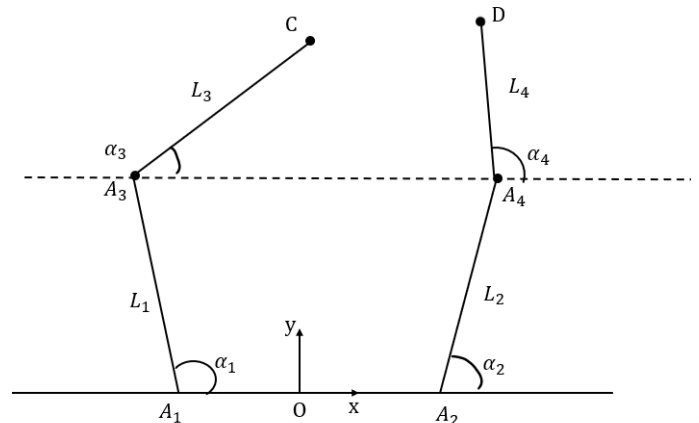
$$\alpha = [\alpha_1 \ \alpha_2]^T \quad (18)$$

Mục tiêu tiếp theo là phát triển các biểu thức cho từng thuật ngữ xuất hiện trong (16) cho 2DOF-PPR để thu được các phương trình chuyển động. Chúng tôi bắt đầu với tham số hóa

$$\alpha(\alpha') = \alpha \quad (19)$$

Dẫn đến:

$$\alpha(\alpha') = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \alpha' = \alpha \quad (20)$$



Hình 2: Hệ thống tự do

Bây giờ, bằng cách kết hợp các phương trình (16) và (20) và vi phân theo α' , chúng tôi thu được biểu thức sau cho $\eta_{\alpha'}(\alpha')$:

$$\eta_{\alpha'}(\alpha') = \begin{bmatrix} \eta_{\alpha'}(1,1) & \eta_{\alpha'}(1,2) & \eta_{\alpha'}(1,3) & \eta_{\alpha'}(1,4) \\ \eta_{\alpha'}(2,1) & \eta_{\alpha'}(2,2) & \eta_{\alpha'}(2,3) & \eta_{\alpha'}(2,4) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (21)$$

Trong đó,

$$\begin{aligned} \eta_{\alpha'}(1,1) &= -L_1 \sin(\alpha_1) - L_3 \sin(\alpha_1 + \alpha_3) \\ \eta_{\alpha'}(1,2) &= L_2 \sin(\alpha_2) + L_4 \sin(\alpha_2 + \alpha_4) \\ \eta_{\alpha'}(1,3) &= -L_3 \sin(\alpha_1 + \alpha_3) \\ \eta_{\alpha'}(1,4) &= L_4 \sin(\alpha_2 + \alpha_4) \\ \eta_{\alpha'}(2,1) &= L_1 \cos(\alpha_1) + L_3 \cos(\alpha_1 + \alpha_3) \\ \eta_{\alpha'}(2,2) &= -L_2 \cos(\alpha_2) - L_4 \cos(\alpha_2 + \alpha_4) \\ \eta_{\alpha'}(2,3) &= L_3 \cos(\alpha_1 + \alpha_3) \\ \eta_{\alpha'}(2,4) &= -L_4 \cos(\alpha_2 + \alpha_4) \end{aligned} \quad (22)$$

Chúng ta có biểu thức sau cho $\tau(\alpha')$:

$$\tau(\alpha') = \eta_{\alpha'}^{-1}(\alpha') \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (23)$$

Vì $\tau(\alpha')$ được biểu diễn dưới dạng ma trận nghịch đảo, nên việc lấy đạo hàm theo thời gian của nó không đơn giản. Do đó, chúng ta sử dụng biểu thức sau cho $\dot{\tau}(\alpha', \dot{\alpha}')$, có thể dễ dàng thu được bằng cách nhân trước (23) với $\eta_{\alpha'}(\alpha')$ và sau đó lấy đạo hàm theo thời gian:

$$\dot{\tau}(\alpha', \dot{\alpha}') = -\eta_{\alpha'}^{-1}(\alpha') \dot{\eta}_{\alpha'}(\alpha', \dot{\alpha}') \tau(\alpha') \quad (24)$$

trong đó $\dot{\eta}_{\alpha'}(\alpha', \dot{\alpha}')$ Có thể thu được bằng cách lấy đạo hàm của phương trình (21) theo thời gian. Tham khảo [30], chúng ta sẽ phát triển các biểu thức cho: $N'(\alpha')$, $M'(\alpha', \dot{\alpha}')$, và $\mathbf{h}'(\alpha')$. Các biểu thức sau đây được suy ra bằng phương pháp Lagrangian:

$$N'(\alpha') = \begin{bmatrix} n_{1,1} & 0 & n_{1,3} & 0 \\ 0 & n_{2,2} & 0 & n_{2,4} \\ n_{3,1} & 0 & n_{3,3} & 0 \\ 0 & n_{4,2} & 0 & n_{4,4} \end{bmatrix}, \quad (25)$$

Với:

$$\begin{aligned} n_{1,1} &= m_{l1} d_1^2 + m_{l3} (L_1^2 + d_3^2 + 2L_1 d_3 \cos(\alpha_3)) + C_1 + C_3 \\ n_{1,3} &= m_{l3} (d_3^2 + L_1 d_3 \cos(\alpha_3)) + C_3 \\ n_{2,2} &= m_{l2} d_2^2 + m_{l4} (L_2^2 + l_{c4}^2 + 2L_2 d_4 \cos(\alpha_4)) + C_2 + C_4 \\ n_{2,4} &= m_{l4} (d_4^2 + L_2 d_4 \cos(\alpha_4)) + C \\ n_{3,1} &= n_{1,3}, n_{3,3} = m_{l3} l_c^2 + \\ n_{4,2} &= n_{2,4}, n_{4,4} = m_{l4} d_4^2 + C_4 \end{aligned} \quad (26)$$

$M'(\alpha', \dot{\alpha}') =$

$$\begin{bmatrix} \rho_1 \dot{\alpha}_3 & 0 & \rho_1 (\dot{\alpha}_1 + \dot{\alpha}_3) & 0 \\ 0 & \rho_2 \dot{\alpha}_4 & 0 & \rho_2 (\dot{\alpha}_2 + \dot{\alpha}_4) \\ -\rho_1 \dot{\alpha}_1 & 0 & 0 & 0 \\ 0 & -\rho_2 \dot{\alpha}_2 & 0 & 0 \end{bmatrix} \quad (27)$$

Với

$$\begin{aligned} \rho_1 &= -m_{l3} L_1 d_3 \sin(\alpha_3) \\ \rho_2 &= -m_{l4} L_2 d_4 \sin(\alpha_4) \end{aligned} \quad (28)$$

Và:

$$\mathbf{h}'(\alpha') = \begin{bmatrix} (m_{l1} d_1 + m_{l3} L_1) \cos(\alpha_1) + m_3 l_{c3} \cos(\alpha_1 + \alpha_3) \\ (m_{l2} d_2 + m_{l4} L_2) \cos(\alpha_2) + m_4 l_{c4} \cos(\alpha_2 + \alpha_4) \\ m_{l3} d_3 \cos(\alpha_1 + \alpha_3) \\ m_{l4} d_4 \cos(\alpha_2 + \alpha_4) \end{bmatrix} \quad (29)$$

trong đó $g = 9.81 \text{ m/s}^2$ là gia tốc trọng trường. Ở giai đoạn này, chúng tôi đã thành công trong việc thiết lập các phương trình chuyển động cho robot song song hai bậc tự do theo α' . Nhiệm vụ duy nhất còn lại là lấy tham số hóa $\alpha' = \sigma(\alpha)$. Nói chung, việc thu được một biểu thức phân tích cho $\sigma(\alpha)$, và nó phải được tính toán bằng các phương pháp số. Tuy nhiên, đối với 2DOF-PPR, có thể giải quyết (16) để tạo ra kết quả sau:

$$\begin{aligned} \alpha_4 &= \tan^{-1} \left[\frac{\pm \sqrt{\omega(\alpha_1, \alpha_2)^2 + \varepsilon(\alpha_1, \alpha_2)^2 - \chi(\alpha_1, \alpha_2)^2}}{\chi(\alpha_1, \alpha_2)} \right] \\ &+ \tan^{-1} \left[\frac{\varepsilon(\alpha_1, \alpha_2)}{\omega(\alpha_1, \alpha_2)} \right] - \alpha_2 \end{aligned} \quad (30)$$

Trong đó

$$\begin{aligned} \omega(\alpha_1, \alpha_2) &= 2L_4 \phi(\alpha_1, \alpha_2) \\ \varepsilon(\alpha_1, \alpha_2) &= 2L_4 \eta(\alpha_1, \alpha_2) \\ \chi(\alpha_1, \alpha_2) &= L_3^2 - L_4^2 - \phi(\alpha_1, \alpha_2)^2 - \eta(\alpha_1, \alpha_2)^2 \end{aligned} \quad (31)$$

Với

$$\begin{aligned} \phi(\alpha_1, \alpha_2) &= L_2 \cos(\alpha_2) - L_1 \cos(\alpha_1) + c \\ \eta(\alpha_1, \alpha_2) &= L_2 \sin(\alpha_2) - L_1 \sin(\alpha_1) \end{aligned} \quad (32)$$

Cuối cùng ta có:

$$\alpha_3 = \tan^{-1} \left[\frac{\eta(\alpha_1, \alpha_2) + L_4 \sin(\alpha_2 + \alpha_4)}{\phi(\alpha_1, \alpha_2) + L_4 \cos(\alpha_2 + \alpha_4)} \right] - \alpha_1 \quad (33)$$

Vì vậy, (30) và (33) kết hợp lại đại diện cho tham số hóa $\alpha' = \sigma(\alpha)$. Điều này kết thúc việc xây dựng các phương trình chuyển động của 2DOF-PPR. Tóm lại, các phương trình chuyển động của robot song song hai bậc tự do, được thể hiện như sau:

$$N(\alpha') \ddot{\alpha} + M(\dot{\alpha}', \alpha') \dot{\alpha} + \mathbf{h}(\alpha') = \mathbf{u} \quad (34)$$

Trong đó

$$\begin{aligned} N(\alpha') &= \tau(\alpha')^T N'(\alpha') \tau(\alpha') \\ M(\alpha', \dot{\alpha}') &= \tau(\alpha')^T M'(\alpha', \dot{\alpha}') \tau(\alpha') \\ &+ \tau(\alpha')^T N'(\alpha') \dot{\tau}(\alpha', \dot{\alpha}') \\ \mathbf{h}(\alpha') &= \tau(\alpha')^T \mathbf{h}'(\alpha') \\ \alpha &= [\alpha_1 \quad \alpha_2] \end{aligned} \quad (35)$$

3. Tối ưu tham số bộ điều khiển trượt cho 2DOF-PPR

3.1. Thiết kế bộ điều khiển trượt

Giả sử rằng $E_d = [x_d \ y_d]^T$ là quỹ đạo mong muốn của cơ cấu tác động cuối và α_d là quỹ đạo mong muốn của hai góc quay. Từ đó, sai lệch bám e và mặt trượt s được định nghĩa như sau:

$$e = \alpha - \alpha_d \quad (36)$$

$$s = \dot{e} + ce \quad (37)$$

Trong đó $\alpha, \alpha_d \in \mathbb{R}^2$. Do đó, e cũng là một vector cột 2×1 : $e = \begin{bmatrix} e_{\alpha_1} \\ e_{\alpha_2} \end{bmatrix}$, $e \in \mathbb{R}^2$ và, $c = \begin{bmatrix} c_{11} & 0 \\ 0 & c_{22} \end{bmatrix}$ là một ma trận đường chéo xác định dương 2×2 và $s \in \mathbb{R}^2$.

Một hàm Lyapunov $V(s)$ điển hình cho SMC có thể được định nghĩa như sau:

$$V(s) = \frac{1}{2} s^T s \quad (38)$$

Đạo hàm theo thời gian của hàm Lyapunov ta thu được:

$$\dot{V}(s) = s^T \dot{s} \quad (39)$$

Để đảm bảo tính ổn định, điều kiện sau cần được thỏa mãn:

$$\dot{V}(s) < 0 \quad \forall s \neq 0 \quad (40)$$

Để thỏa mãn (40), chúng ta cần thiết kế luật điều khiển phù hợp. Trong bộ SMC, một luật điều khiển điển hình có dạng:

$$u = u_{eq} - K \cdot \text{sign}(s) \quad (41)$$

Trong đó:

- u_{eq} là điều khiển tương đương, đại diện cho phần điều khiển liên tục giúp hệ thống duy trì trên bề mặt trượt.
- $K \cdot \text{sign}(s)$ là điều khiển không liên tục để đẩy hệ thống về bề mặt trượt.
- K là một hằng số dương đủ lớn để đảm bảo hệ thống hội tụ.

Sự xuất hiện của điều khiển không liên tục với hàm $\text{sign}(s)$ đảm bảo rằng:

$$\dot{V}(s) = s^T \dot{s} = s^T (-K \cdot \text{sign}(s)) = -K s^T \text{sign}(s) = -K \sum_{i=1}^2 |s_i| \quad (42)$$

Vì $K > 0$, ta có $\dot{V}(s) < 0$, từ đó hệ thống sẽ ổn định theo tiêu chuẩn Lyapunov.

Để tính u_{eq} , ta đạo hàm phương trình (37) để có \dot{s} và cho $\dot{s} = 0$

$$\dot{s} = \ddot{e} - c\dot{e} = 0 \quad (43)$$

Từ phương trình (34), (36) và (43) có:

$$\dot{s} = -\ddot{\alpha}_d - c\dot{e} + N(\alpha')^{-1} [u_{eq} - M(\alpha', \alpha') \dot{\alpha} - h(\alpha')] = 0$$

$$(44)$$

Suy ra :

$$u_{eq} = N(\alpha') (\ddot{\alpha}_d + c\dot{e}) + M(\alpha', \alpha') \dot{\alpha} + h(\alpha') \quad (45)$$

Kết hợp phương trình (41) và (45) ta được bộ điều khiển cho robot song song 2 bậc tự do như sau:

$$u = N(\alpha') (\ddot{\alpha}_d + c\dot{e}) + M(\alpha', \alpha') \dot{\alpha} + h(\alpha') - K \cdot \text{sign}(s) \quad (46)$$

Từ phương trình (46), có thể kết luận rằng với tín hiệu điều khiển u như trong (45), sai số bám e sẽ tiến về 0 tiệm cận, đồng thời hệ thống điều khiển được đảm bảo ổn định.

Trong thực tế, thành phần phi tuyến $u_{sw} = \text{sign}(S)$ là một vấn đề chính của SMC cổ điển do tính chất tần số cao của nó. Một giải pháp phổ biến là thay thế u_{sw} bằng một tín hiệu điều khiển tương đương liên tục, ví dụ như $\text{sat}(s)$. Kết quả là tín hiệu điều khiển cuối cùng là:

$$u = N(\alpha') (\ddot{\alpha}_d + c\dot{e}) + M(\alpha', \alpha') \dot{\alpha} + h(\alpha') - K \cdot \text{sat}(s) \quad (47)$$

Trong đó, $\text{sat}(s)$ được biểu diễn bởi:

$$\text{sat}(s_i) = \begin{cases} \frac{s_i}{\delta}, & \text{nếu } |s_i| \leq \delta \\ \text{sign}(s_i), & \text{nếu } |s_i| > \delta \end{cases} \quad (48)$$

với δ là số dương nhỏ.

3.2. Tối ưu hóa bầy đàn

3.2.1. Cấu trúc và nguyên lý hoạt động của PSO

Thuật toán PSO là một phương pháp tối ưu hóa dựa trên mô phỏng hành vi xã hội của các loài động vật, đặc biệt là các bầy chim. Mô hình PSO bắt đầu bằng cách khởi tạo một quần thể gồm N cá thể, trong đó mỗi cá thể đại diện cho một giải pháp tiềm năng trong không gian tìm kiếm có D chiều. Các cá thể được khởi tạo ngẫu nhiên với vị trí X_i và vận tốc V_i , được xác định như sau:

$$X_i = lb + (ub - lb) \cdot \text{rand}(1, D) \quad (49)$$

$$V_i = 0$$

Trong đó, lb và ub lần lượt là giới hạn dưới và giới hạn trên của các tham số cần tối ưu hóa. Mỗi cá thể sẽ tính toán giá trị hàm mục tiêu f cho vị trí hiện tại của nó. Giá trị này sau đó được so sánh với giá trị tốt nhất trước đó mà cá thể đã tìm thấy (Pbest) và giá trị tốt nhất toàn cục (Gbest) trong quần thể. Mỗi cá thể sẽ cập nhật vận tốc và vị trí của mình theo công thức:

$$V_i(t+1) = w \cdot V_i(t) + b_1 \cdot r_1 \cdot (Pbest_i - X_i(t)) + b_2 \cdot r_2 \cdot (Gbest - X_i(t)) \quad (50)$$

Trong đó:

- w là trọng số quán tính, điều chỉnh ảnh hưởng của vận tốc trước đó.
- b_1 và b_2 là các hệ số học tập cá nhân và xã hội, xác định mức độ ảnh hưởng của Pbest và Gbest.
- r_1 và r_2 là các số ngẫu nhiên trong khoảng $[0, 1]$.

Sau khi cập nhật vận tốc, vị trí mới của mỗi cá thể sẽ được kiểm tra và giới hạn trong khoảng $[lb, ub]$, nhằm đảm bảo rằng các giải pháp vẫn nằm trong không gian tìm kiếm hợp lệ.

3.2.2. Quy trình cập nhật và tiêu chí dừng

Trong mỗi vòng lặp, giá trị hàm mục tiêu được tính toán cho vị trí mới của mỗi cá thể. Nếu giá trị này tốt hơn giá trị Pbest trước đó của cá thể, Pbest sẽ được cập nhật. Tương tự, nếu giá trị hàm mục tiêu của một cá thể tốt hơn giá trị Gbest, thì Gbest cũng sẽ được cập nhật. Quá trình này sẽ được lặp đi lặp lại cho đến khi đạt được một trong các tiêu chí dừng đã định sẵn, chẳng hạn như số vòng lặp tối đa hoặc khi không có sự cải thiện nào trong giá trị Gbest trong một số vòng lặp liên tiếp. Kết quả cuối cùng của thuật toán là giá trị tốt nhất của hàm mục tiêu và vị trí tương ứng, từ đó cung cấp các tham số tối ưu cho mô hình SMC (SMC).

Thuật toán PSO dưới đây được thiết kế để giảm thiểu một hàm mục tiêu tùy chỉnh, nhằm tối ưu hóa các tham số của hệ thống điều khiển. Những thông số chính của thuật toán bao gồm:

- Kích thước quần thể: 30 cá thể, cho phép tìm kiếm giải pháp hiệu quả trong không gian tham số.
- Số lượng tham số cần tối ưu hóa là 4 (tương ứng với các tham số điều khiển của hệ thống: $\delta, K, c_{11}, c_{22}$).
- Số vòng lặp tối đa: 25, nhằm đảm bảo hội tụ trong thời gian tính toán hợp lý.
- Biên dưới và biên trên (lb, ub): Các tham số được khởi tạo trong khoảng [0.01, 100] dựa trên các giới hạn vật lý và kỹ thuật của hệ thống điều khiển.
- Hệ số quán tính (w): 0.7, giúp cân bằng giữa khả năng thăm dò và khai thác trong không gian tìm kiếm.
- Hệ số tăng tốc (b_1, b_2): 1.5, đảm bảo ảnh hưởng đồng đều giữa vị trí tốt nhất của cá thể (Pbest) và vị trí tốt nhất toàn cục (Gbest).

Điều kiện khởi tạo :

- Vị trí ban đầu (X): Các cá thể được khởi tạo ngẫu nhiên trong phạm vi $[lb, ub]$ cho từng tham số điều khiển.
- Vận tốc ban đầu (V): Được khởi tạo bằng 0, đảm bảo các cá thể không có chuyển động ngay từ đầu.

Hàm mục tiêu được theo tiêu chuẩn tích phân tối ưu IAE của hệ thống điều khiển. Quá trình tính toán như sau:

- Sử dụng mô hình Simulink bằng các công thức ở phần 2 để mô phỏng hệ thống với các tham số điều khiển hiện tại.
- f_{ex}, f_{ey} là sai lệch điều khiển theo hai phương và IAE_{fex}, IAE_{fey} là tích phân giá trị tuyệt đối của sai lệch điều khiển theo thời gian.
- Hàm mục tiêu tổng quát được định nghĩa bởi:

$$f = IAE_{fex} + IAE_{fey}. \quad (51)$$

Đoạn chương trình sau đây được sử dụng để dò tìm tham số theo phương pháp PSO trên Matlab

```
1 clear; clc;
2 f = @(x) object(x);
3 num_particles = 30;
4 num_dimensions = 4;
5 max_iter = 25;
6 w = 0.7;
7 b1 = 1.5;
8 b2 = 1.5;
9 lb = 0.01;
10 ub = 100;
11 X = lb + (ub - lb) * rand(num_particles,
    num_dimensions);
```

```
12 V = zeros(num_particles, num_dimensions);
13 Pbest = X;
14 Pbest_fitness = zeros(num_particles, 1);
15 for i = 1:num_particles
16     Pbest_fitness(i) = f(X(i, :));
17 end
18 [Gbest_fitness, idx] = min(Pbest_fitness);
19 Gbest = Pbest(idx, :);
20 for iter = 1:max_iter
21     for i = 1:num_particles
22         x = X(i, :);
23         r1 = rand(1, num_dimensions);
24         r2 = rand(1, num_dimensions);
25         V(i, :) = w * V(i, :) + b1 * r1 .* (
26             Pbest(i, :) - x) + b2 * r2 .* (Gbest - x);
27         x = x + V(i, :);
28         x = max(min(x, ub), lb);
29         X(i, :) = x;
30         fitness = f(x);
31         if fitness < Pbest_fitness(i)
32             Pbest(i, :) = x;
33             Pbest_fitness(i) = fitness;
34         end
35         if Pbest_fitness(i) < Gbest_fitness
36             Gbest_fitness = Pbest_fitness(i);
37             Gbest = Pbest(i, :);
38         end
39     end
40     fprintf('Iteration %d: Best fitness = %.6f\n',
41         iter, Gbest_fitness);
42     fprintf('Best solution found: %.6f at
43         position [%.4f, %.4f, %.4f, %.4f]\n',
44         Gbest_fitness, Gbest(1), Gbest(2), Gbest(3),
45         Gbest(4));
46 end
47 fprintf('Best solution found: %.6f at position
48     [%.4f, %.4f, %.4f, %.4f]\n', Gbest_fitness,
49     Gbest(1), Gbest(2), Gbest(3), Gbest(4));
50
51 function F = object(k)
52     n = size(k, 1);
53     A = zeros(n, 1);
54
55     for i = 1:n
56
57         set_param('toiuu4bac/delta', 'Value',
58             num2str(k(i, 1)));
59         set_param('toiuu4bac/K', 'Value',
60             num2str(k(i, 2)));
61         set_param('toiuu4bac/c11', 'Value',
62             num2str(k(i, 3)));
63         set_param('toiuu4bac/c22', 'Value',
64             num2str(k(i, 4)));
65         simOut = sim('toiuu4bac.slx', '
66             SrcWorkspace', 'current');
67
68         fex = simOut.get('ex');
69         fey = simOut.get('ey');
70         time = simOut.get('time');
71
72         IAE_fex = trapz(time, abs(fex));
73         IAE_fey = trapz(time, abs(fey));
74         f = IAE_fex + IAE_fey;
75         A(i) = abs(f);
76     end
77     F = A;
```

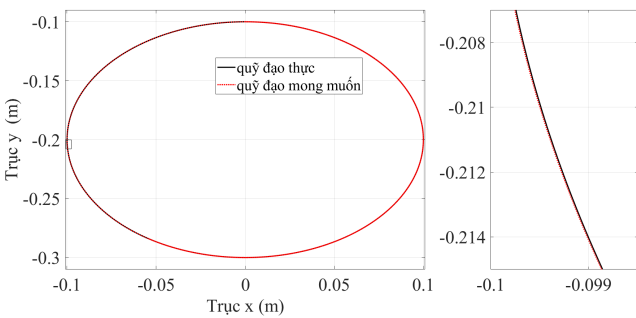
4. Kết quả mô phỏng

Để kiểm chứng giải pháp điều khiển được đề xuất, chúng tôi đã tiến hành mô phỏng cho khâu tác động cuối của 2DOF-PPR với quỹ đạo hình tròn sử dụng bộ điều khiển được đề xuất ở trên. Sau khi tối ưu tham số bằng bộ PSO, chúng tôi thấy các

giá trị $c = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$, $K = 3.8503, \delta = 0.01$ sẽ cho robot chạy với kết quả tốt nhất. Các thông số của robot cũng như bộ điều khiển tối ưu SMC được sử dụng trong các mô phỏng được cung cấp trong bảng 1. Có thể thấy từ hình 3, hình 4 và hình 5 rằng khâu tác động cuối bám theo quỹ đạo tham chiếu một cách chính xác.

Bảng 1: Thông số của robot

| Thông số của robot | Giá trị | Đơn vị |
|--------------------|----------|-------------------|
| m_1 | 0.2451 | kg |
| m_2 | 0.2352 | kg |
| m_3 | 0.2611 | kg |
| m_4 | 0.2611 | kg |
| L_1 | 0.2794 | m |
| L_2 | 0.2794 | m |
| L_3 | 0.3048 | m |
| L_4 | 0.3048 | m |
| C_1 | 0.002779 | kg m ² |
| C_2 | 0.002607 | kg m ² |
| C_3 | 0.003476 | kg m ² |
| C_4 | 0.003476 | kg m ² |



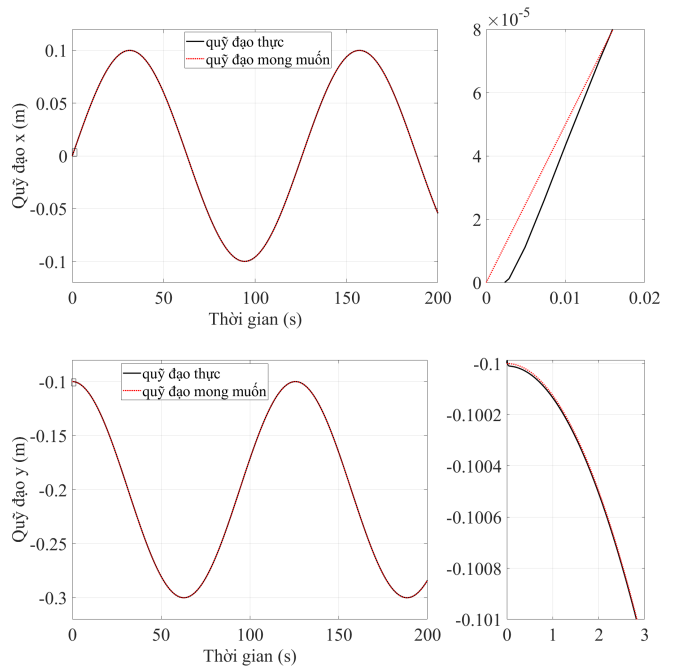
Hình 3: Quỹ đạo vị trí của khâu tác động cuối

Hình 4 và Hình 5 lần lượt cho thấy quỹ đạo vị trí của cơ cấu tác động cuối và các khớp. Có thể thấy các cơ cấu này được điều khiển bám theo quỹ đạo tham chiếu một cách hoàn hảo. Đặc biệt là quá trình quá độ diễn ra nhanh, không có hiện tượng dao động, cho thấy tính hiệu quả của bộ tham số tối ưu được tìm kiếm bởi thuật toán PSO.

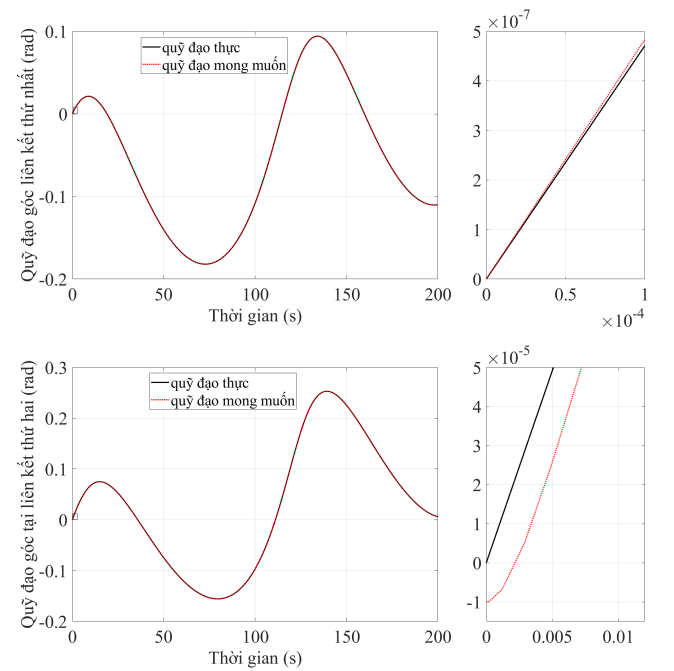
Hình 6 cho thấy sai lệch bám của hệ điều khiển chuyển động nhanh chóng tiến về 0. Điều này khẳng định sự thành công của hệ thống điều khiển trong việc bám quỹ đạo một cách chính xác. Hình 7 cung cấp thông tin về tín hiệu điều khiển, trong thực tế chính là mô men đặt vào các khớp chủ động. Có thể thấy tín hiệu điều khiển này có giá trị hợp lý và không có sự thay đổi đột ngột trong tín hiệu. Điều này giúp giảm thiểu căng thẳng cho các bộ truyền động và các thành phần cơ khí của hệ thống. Hiện tượng "chattering" đã được loại bỏ hoặc giảm thiểu đáng kể nhờ việc sử dụng hàm bão hòa thay thế cho hàm đóng/cắt.

5. Kết Luận

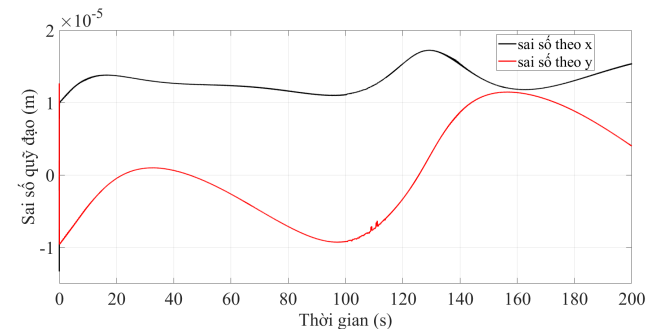
Kết quả mô phỏng cho thấy hiệu quả rõ rệt của phương pháp đề xuất, với sự cải thiện đáng kể trong việc kiểm soát quỹ đạo của robot song song phẳng 2 bậc tự do. Phương pháp này không



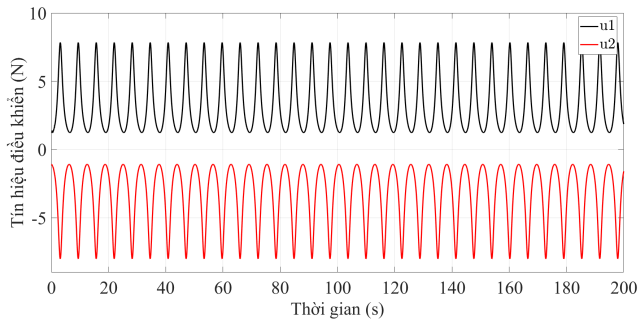
Hình 4: Quỹ đạo vị trí của khâu tác động cuối theo hệ trục tọa độ



Hình 5: Quỹ đạo góc của khớp điều khiển robot



Hình 6: Sai số quỹ đạo của khâu tác động cuối



Hình 7: Tín hiệu điều khiển

chỉ giúp tối ưu hóa tham số cho bộ SMC, mà còn giảm thiểu thời gian tinh chỉnh thủ công, mang lại sự tiện lợi và hiệu suất cao trong quá trình điều khiển. Bộ điều khiển tối ưu đã được kiểm chứng qua các mô phỏng trên hệ thống robot, khẳng định tiềm năng ứng dụng của phương pháp này trong điều khiển thời gian thực các hệ thống động học phức tạp.

Mặc dù đạt được các kết quả ấn tượng, nghiên cứu vẫn còn một số hạn chế. Đầu tiên, phương pháp PSO được áp dụng trong môi trường mô phỏng, và việc triển khai thực tế trên các hệ thống robot có thể gặp phải các yếu tố không chắc chắn, như nhiễu động từ môi trường hoặc sai số trong các mô hình động học. Thêm vào đó, PSO có thể yêu cầu thời gian tính toán lớn khi xử lý các bài toán có không gian tham số rộng và phức tạp hơn.

Một hướng mở rộng tiềm năng cho nghiên cứu này là áp dụng phương pháp tối ưu hóa cho các hệ thống robot có số bậc tự do cao hơn hoặc trong các môi trường thực tế có điều kiện làm việc thay đổi, như trong các ứng dụng công nghiệp với độ phức tạp và biến động lớn hơn. Cũng có thể kết hợp phương pháp tối ưu hóa với các kỹ thuật điều khiển thích nghi để nâng cao khả năng thích ứng với các điều kiện không chắc chắn trong thời gian thực, từ đó mở rộng khả năng ứng dụng của phương pháp này trong các hệ thống robot phức tạp hơn.

Tài liệu

- [1] Ji, W., Wang, L. (2019). Industrial robotic machining: a review. *The International Journal of Advanced Manufacturing Technology*, 103, 1239-1255.
- [2] Jia-Qing, X., Sun-Han, X. (2014). Review on kinematics calibration technology of serial robots [j]. *International Journal of Precision Engineering and Manufacturing*, 15(8), 1759-1774.
- [3] Sun, T., Zhai, Y., Song, Y., Zhang, J. (2016). Kinematic calibration of a 3-DoF rotational parallel manipulator using laser tracker. *Robotics and Computer-Integrated Manufacturing*, 41, 78-91.
- [4] Li, Z., Li, S., Luo, X. (2021). An overview of calibration technology of industrial robots. *IEEE/CAA Journal of Automatica Sinica*, 8(1), 23-36.
- [5] Penteridis, L., D'Onofrio, G., Sancarlo, D., Giuliani, F., Ricciardi, F., Cavallo, F., ... Gkiokas, A. (2017). Robotic and sensor technologies for mobility in older people. *Rejuvenation research*, 20(5), 401-410.
- [6] De Luca, A., Book, W. J. (2016). Robots with flexible elements. *Springer handbook of robotics*, 243-282.
- [7] Mo, J., Shao, Z. F., Guan, L., Xie, F., Tang, X. (2017). Dynamic performance analysis of the X4 high-speed pick-and-place parallel robot. *Robotics and Computer-Integrated Manufacturing*, 46, 48-57.
- [8] Wu, K., Krewet, C., Kuhlenkötter, B. (2018). Dynamic performance of industrial robot in corner path with CNC controller. *Robotics and Computer-Integrated Manufacturing*, 54, 156-161.
- [9] Soori, M., Arezoo, B., Dastres, R. (2023). Optimization of energy consumption in industrial robots, a review. *Cognitive Robotics*, 3, 142-157.
- [10] Zhang, D., Wei, B. (2017). Interactions and optimizations analysis between stiffness and workspace of 3-UPU robotic mechanism. *Measurement Science Review*, 17(2), 83-92.
- [11] Tajdari, F., Ebrahimi Toulkani, N. (2022). Implementation and intelligent gain tuning feedback-based optimal torque control of a rotary parallel robot. *Journal of Vibration and Control*, 28(19-20), 2678-2695.
- [12] Qin, L., Peng, H., Huang, X., Liu, M., Huang, W. (2024). Modeling and simulation of dynamics in soft robotics: A review of numerical approaches. *Current Robotics Reports*, 5(1), 1-13.
- [13] Shen, S., Zhang, D., Han, X., Jiang, C., Li, Q. (2024). A robust optimization framework for design of robotic system with kinematic and dynamic criteria. *Computer Methods in Applied Mechanics and Engineering*, 423, 116866.
- [14] Mazloomi, N., Zandinejad, Z., Zaretalab, A., Gholipour, M. (2024). Multi-objective QoS optimization in swarm robotics. *Robotics and Autonomous Systems*, 182, 104796.
- [15] Hosseini, S. M., Shirvani, M. H., Motameni, H. (2024). Multi-objective discrete Cuckoo search algorithm for optimization of bag-of-tasks scheduling in fog computing environment. *Computers and Electrical Engineering*, 119, 109480.
- [16] Salgotra, R., Gandomi, A. H. (2024). A novel multi-hybrid differential evolution algorithm for optimization of frame structures. *Scientific Reports*, 14(1), 4877.
- [17] Chen, X., Xin, C., Zhang, Z., Guo, Y., Yin, A., Chen, Z. (2024). Design principles and kinematic analysis of a novel spherical 2-degree-of-freedom (DOF) parallel mechanism. *Mechanical Sciences*, 15(2), 473-486.
- [18] Aminiazar, W., Najafi, F., Nekoui, M. A. (2013). Optimized intelligent control of a 2-degree of freedom robot for rehabilitation of lower limbs using neural network and genetic algorithm. *Journal of neuroengineering and rehabilitation*, 10, 1-11.
- [19] Mayetin, U., Küçük, S. (2022). Comparative Study of Control Algorithms for 2-Degree of Freedom RR Planar Robot Manipulator Using Particle Swarm Optimization. *Uluslararası Sivas Bilim ve Teknoloji Üniversitesi Dergisi*, 1(2), 35-62.
- [20] Sheng, L., Li, W. (2018). Optimization design by genetic algorithm controller for trajectory control of a 3-RRR parallel robot. *Algorithms*, 11(1), 7.
- [21] Lopez, E. G., Yu, W., Li, X. (2021). Optimum design of a parallel robot using neuro-genetic algorithm. *Journal of Mechanical Science and Technology*, 35, 293-305.
- [22] Kashyap, A. K., Parhi, D. R. (2021). Optimization of stability of humanoid robot NAO using ant colony optimization tuned MPC controller for uneven path. *Soft Computing*, 25(7), 5131-5150.
- [23] Wang, X., Yang, D., Dolly, D. R. J., Chen, S., Alassafi, M. O., Alsaadi, F. E. (2024). Opposition-based differential evolution for synchronized control of multi-agent systems with uncertain nonlinear dynamics. *Applied Soft Computing*, 150, 111044.
- [24] Dao, T. K., Nguyen, T. T. (2024). A review of the bat algorithm and its varieties for industrial applications. *Journal of Intelligent Manufacturing*, 1-23.
- [25] Yin, L., Zhang, Y. (2024). Particle swarm optimization based on data driven for EV charging station siting. *Energy*, 310, 133197.
- [26] Aljaidi, M., Samara, G., Singla, M. K., Alsarhan, A., Hassan, M., Safaraliev, M., ... Tavlintsev, A. (2024). A particle swarm optimizer-based optimization approach for locating electric vehicles charging stations in smart cities. *International Journal of Hydrogen Energy*, 87, 1047-1055.
- [27] Zhang, Y. X., Cong, S., Shang, W. W., Li, Z. X., Jiang, S. L. (2007). Modeling, identification and control of a redundant planar 2-DOF parallel manipulator. *International Journal of Control, Automation, and Systems*, 5(5), 559-569.
- [28] Liu, X. J., Wang, J., Pritschow, G. (2006). Performance atlases and optimum design of planar 5R symmetrical parallel mechanisms. *Mechanism and machine theory*, 41(2), 119-144.
- [29] Liu, X. J., Wang, J., Pritschow, G. (2006). Kinematics, singularity and workspace of planar 5R symmetrical parallel mechanisms. *Mechanism and machine theory*, 41(2), 145-169.
- [30] Ghorbel, F., Gunawardana, R. (1997, December). A validation study of PD control of a closed-chain mechanical system. In *Proceedings of the 36th IEEE Conference on Decision and Control (Vol. 2, pp. 1998-2004)*. IEEE.