

Tối ưu điều khiển trượt với hàm trượt dạng PID sử dụng giải thuật bầy đàn

Huỳnh Minh Vũ* và Nguyễn Đình Tứ

Trường Đại học Kỹ thuật - Công nghệ Cần Thơ

TÓM TẮT

Bộ điều khiển trượt với hàm trượt dạng PID (Proportional Integral Derivative) có ưu điểm là có tính ổn định, bền vững bằng cách giảm thiểu hiện tượng dao động quanh mặt trượt. Tuy nhiên, để thiết kế được bộ điều khiển trượt với hàm trượt dạng PID người thiết kế cần chọn các tham số bằng phương pháp thử-sai. Vì vậy, việc cải thiện một giải pháp giúp tìm kiếm tham số tối ưu của bộ điều khiển trượt với hàm trượt dạng PID là cần thiết. Bài báo trình bày một phương pháp để tối ưu tham số của bộ điều khiển trượt với hàm trượt dạng PID sử dụng giải thuật bầy đàn cho mô hình tay máy một bậc tự do. Kết quả mô phỏng dựa trên phần mềm MATLAB cho thấy bộ điều khiển ổn định và bền vững dưới tác động của nhiễu và thành phần dead-zone, đáp ứng của hệ thống không xuất hiện vọt lố và sai số xác lập bị triệt tiêu.

Từ khóa: Điều khiển trượt với hàm trượt dạng PID, Giải thuật bầy đàn, Tay máy một bậc tự do, dead-zone, nhiễu

1. GIỚI THIỆU

Điều khiển trượt là phương pháp điều khiển có tính ổn định cao và bền vững trong điều kiện hệ thống có tham số thay đổi hay có nhiễu tác động từ bên ngoài [1].

Tuy nhiên, cách tiếp cận của bộ điều khiển trượt thường xuất hiện hiện tượng dao động quanh mặt trượt (chattering). Hiện tượng này thường ảnh hưởng tới chất lượng điều khiển của hệ thống. Để khắc phục hiện tượng này, bộ điều khiển trượt (SMC - Sliding Mode Control) với hàm trượt có dạng phương trình của bộ điều khiển PID (PID-SMC) đã được đề xuất [2], nghiên cứu của tác giả Nguyễn Hoàng Dũng [3] xây dựng bộ điều khiển trượt với hàm trượt dạng PID sao cho thỏa mãn tiêu chuẩn Hurwitz cũng mang lại hiệu quả làm giảm hiện tượng dao động quanh mặt trượt. Phương pháp điều khiển PID-SMC loại bỏ đáng kể hiện tượng dao động quanh mặt trượt xuất hiện ở bộ điều khiển trượt truyền thống. Hơn nữa, đặc tính dead-zone thường xuất hiện trong các hệ thống thực tế như động cơ servo, van điều khiển thủy lực.v.v [4 - 5]. Dead-zone có thể ảnh hưởng đáng kể tới hiệu suất của hệ thống [5]. Hơn nữa, đặc tính này thường ít được xem xét đến khi thiết kế bộ điều khiển do chúng thay đổi theo thời gian. Do đó, việc xem xét một thành phần phi tuyến như dead-zone trong việc thiết kế bộ điều khiển là cần thiết.

Hiện nay, các giải thuật tối ưu đã được sử dụng rộng rãi cho mục đích tối ưu các tham số của bộ điều khiển. Fatima Loucif và Sihem Kechida đề xuất việc tối ưu bộ điều khiển PID-SMC sử dụng giải thuật ALO [6] cho tay máy. Nghiên cứu chỉ ra sự hiệu quả của ALO so với GWO [7] trong việc tối ưu các thông số của PID-SMC [8]. Ngoài ra, việc tối ưu cho bộ điều khiển PID-SMC còn được đề xuất trong nghiên cứu của Mokeddem và Draïdi [9], nghiên cứu chỉ ra hiệu quả của ALO so với giải thuật di truyền (GA) [10] và giải thuật bầy đàn (PSO) [11] trong quá trình tối ưu PID-SMC để điều khiển hệ thống phi tuyến. Tuy nhiên, việc khảo sát việc tối ưu bộ điều khiển PID-SMC trong trường hợp hệ thống có nhiễu và sự can thiệp của dead-zone vẫn chưa được nhiều sự quan tâm.

Dựa trên những kết quả nghiên cứu trên, bài báo này đề xuất một phương pháp tối ưu bộ điều khiển PID-SMC sử dụng giải thuật bầy đàn (PSO). Các tiếp cận này làm giảm đáng kể sự dao động quanh mặt trượt khi có sự can thiệp của dead-zone và nhiễu trong hệ thống. Đồng thời, việc đánh giá chất lượng điều khiển trong nghiên cứu này được thực hiện trên mô phỏng điều khiển tay máy một bậc tự do với tải trọng thay đổi.

Bài báo được tổ chức gồm bốn phần: Giới thiệu là phần 1, mô hình toán đối tượng điều khiển được trình bày ở phần 2, phần 3 trình bày phương pháp

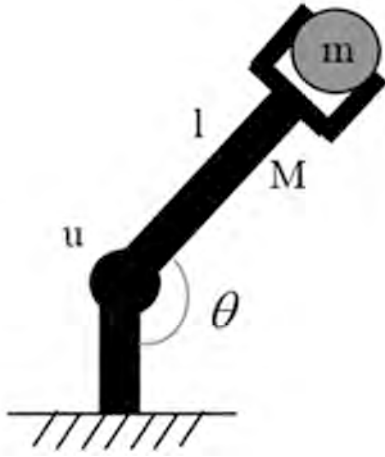
Tác giả liên hệ: Huỳnh Minh Vũ

Email: hmvu@ctu.edu.vn

thiết kế bộ điều khiển được đề xuất và kết luận là nội dung của phần 4.

2. MÔ HÌNH TOÁN HỌC TAY MÁY

Khảo sát mô hình máy một bậc tự do, được trình bày như Hình 1.



Hình 1. Mô hình tay máy một bậc tự do

Dựa trên việc phân tích phương trình Euler-Lagrange, phương trình động lực học của tay máy một bậc tự do được thiết lập như (1) [12].

$$(J + ml^2)\ddot{\theta} + B\dot{\theta} + (ml + Ml_c)\sin(\theta) = u \quad (1)$$

Trong đó: $J = Ml_c^2$ là moment quán tính của cánh tay, các thông số còn lại của mô hình được mô tả chi tiết như trong Bảng 1.

Tay máy sẽ quay quanh trục nhờ moment u , vị trí θ của tay máy được xác định bằng góc hợp bởi phương thẳng đứng và phương của tay máy. Như vậy, u sẽ là ngõ vào điều khiển hệ tay máy và θ là ngõ ra của hệ thống. Giả sử, thời điểm ban đầu tay máy chưa gấp vật, sau khoảng thời gian t tay máy sẽ gấp một vật nặng có khối lượng m .

Bảng 1. Thông số mô phỏng cho mô hình tay máy

Kí hiệu	Ý nghĩa	Giá trị
M	Khối lượng cánh tay	1 kg
m	Khối lượng vật nặng	0.1 kg
l	Chiều dài tay máy	0.4 m
l_c	Khoảng cách từ trọng tâm đến trục quay	0.15 m
B	Hệ số ma sát nhớt	0.2 kg.m ² /s
g	Gia tốc trọng trường	9.81 m ² /s

Đặt $x_1 = \theta(t)$ và $x_2 = \dot{\theta}(t)$

Phương trình (1) có thể biểu diễn dưới dạng phương trình trạng thái như (2).

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{u(t) - Bx_2(t) - (ml + Ml_c)g \sin(x_1(t))}{J + ml^2} \end{cases} \quad (2)$$

Và ngõ ra $y(t) = \theta(t) = x_1(t)$

3. ĐIỀU KHIỂN MÔ HÌNH TAY MÁY

3.1. Thiết kế bộ điều khiển trượt với hàm trượt kiểu PID

Để loại bỏ hiện tượng dao động quanh mặt trượt khi biên độ của luật điều khiển trượt thay đổi lớn, trong bài báo này tác giả chọn hàm trượt theo kiểu PID [2], được trình bày như sau:

$$s(t) = \dot{e}(t) + \lambda_1 e(t) + \lambda_2 \int_0^t e(\tau) d(\tau) \quad (3)$$

Trong đó, $e(t) = y(t) - y_d(t)$ là sai số giữa đáp ứng ngõ ra $y(t)$ và tín hiệu đặt $y_d(t)$; λ_1 và λ_2 là hai hằng số dương được chọn sao cho phương trình đặc tính $S^2 + \lambda_1 S + \lambda_2 = 0$ phải Hurwitz. Trong đó S là biến phức và nghiệm của phương trình $S^2 + \lambda_1 S + \lambda_2 = 0$ phải nằm ở nửa bờ trái của mặt phẳng phức. Khi đó, phương trình (3) được viết lại như (4).

$$\begin{aligned} s(t) &= \dot{y}(t) - \dot{y}_d(t) + \lambda_1 (y(t) - y_d(t)) \\ &+ \lambda_2 \int_0^t (y(\tau) - y_d(\tau)) d(\tau) \end{aligned} \quad (4)$$

Theo lý thuyết ổn định Lyapunov, chọn một hàm xác định dương:

$$V(t) = \frac{1}{2} s^2(t) \rightarrow \dot{V}(t) = s(t) \cdot \dot{s}(t) \quad (5)$$

Để $\dot{V}(t)$ xác định âm, chọn:

$$\dot{s}(t) = -K \cdot \text{sign}(s(t)) \quad (6)$$

Thay (6) vào (5), suy ra (7). Với K là hằng số dương chọn trước.

$$\dot{V}(t) = -s(t) \cdot K \cdot \text{sign}(s(t)) \quad (7)$$

Khi đó, luật điều khiển trượt với hàm trượt kiểu PID được trình bày như phương trình (8).

$$\begin{aligned} u(t) &= B \cdot x_2(t) + (ml + Ml_c)g \sin(x_1(t)) + \\ &+ (J + ml^2) \left[\ddot{y}_d(t) + \lambda_1 (\dot{y}_d(t) - x_2(t)) + \right. \\ &\left. + \lambda_2 (y_d(t) - x_1(t)) + \right. \\ &\left. + K \cdot \text{sign}(s(t)) \right] \end{aligned} \quad (8)$$

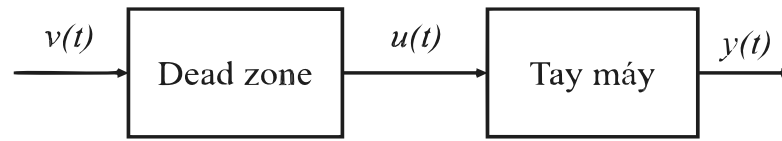
Từ luật điều khiển ở phương trình (8), thấy rõ rằng có ba tham số cần chọn là: λ_1 , λ_2 và K . Thông thường, các tham số này được chọn dựa trên kinh nghiệm và kiểm tra bằng cách thử-sai. Do đó, trong bài báo này tác giả đề xuất sử dụng giải thuật bầy đàn (PSO) để tối ưu các tham số này.

3.2. Hiện tượng Dead-zone

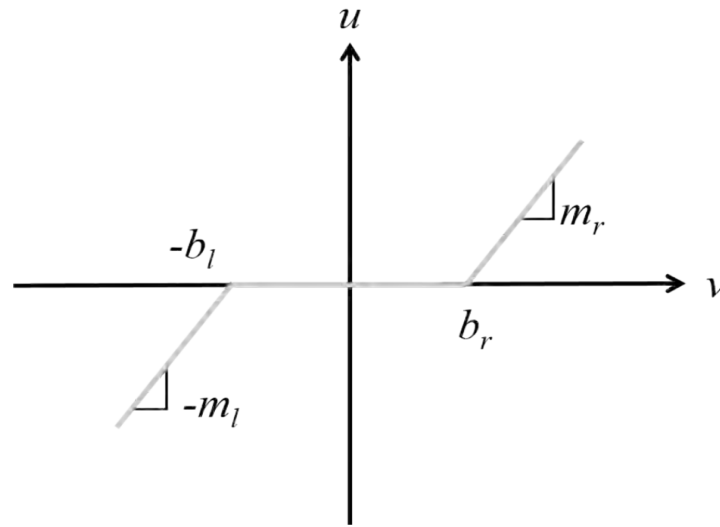
Hiện tượng dead-zone được mô tả là mối liên hệ

giữa đầu vào và đầu ra, đầu ra được cho là bằng không khi đầu vào vượt qua một giá trị nào đó. Hiện tượng này xảy ra trong nhiều bộ phận khác nhau của hệ thống điều khiển như ở các cảm biến,

bộ khuếch đại và cơ cấu truyền động. Ngoài ra, hiện tượng này còn gây ra một số ảnh hưởng đến các hệ điều khiển, làm giảm độ chính xác của điều khiển, mất ổn định của hệ thống.



Hình 2. Hệ thống có dead zone [13]



Hình 3. Đồ thị biểu diễn dead zone [13]

Trên Hình 2, $u(t)$ và $y(t)$ là đầu vào và đầu ra của đối tượng tay máy, $u(t)$ đồng thời là đầu ra của dead zone, đầu vào của dead-zone là $v(t)$.

$$u(t) = \begin{cases} m_r v(t) - m_r b_r, & v(t) > b_r \text{ or } u(t) > 0 \\ 0, & -b_l \leq v(t) \leq b_r \text{ or } u(t) = 0 \\ m_l v(t) + m_l b_l, & v(t) < -b_l \text{ or } u(t) < 0 \end{cases} \quad (9)$$

Dead-zone được biểu diễn cụ thể như Hình 3. Trong đó b_l, b_r, m_r, m_l là các hằng số thực dương.

3.3. Giải thuật bầy đàn (PSO)

Giải thuật bầy đàn (PSO) là một phương pháp tối ưu hóa ngẫu nhiên dựa trên một quần thể và sau đó tìm nghiệm tối ưu bằng cách cập nhật các thể hệ, được phát triển bởi Eberhart và Kennedy (1995) [11]. Giải thuật này phỏng theo hành vi của các bầy chim hay các đàn cá trong quá trình tìm kiếm thức ăn. Mỗi cá thể trong quần thể cập nhật vị trí của nó theo vị trí tốt nhất của nó và của cá thể trong quần thể tính tới thời điểm hiện tại. Quá trình cập nhật các phần tử dựa trên các công thức sau:

$$V_{i,j}^{k+1} = w \times V_{i,j}^k + c_1 r_1 (Pbest_{i,j}^k - X_{i,j}^k) + c_2 r_2 (Gbest_j^k - X_{i,j}^k) \quad (10)$$

$$X_{i,j}^{k+1} = X_{i,j}^k + V_{i,j}^{k+1} \quad (11)$$

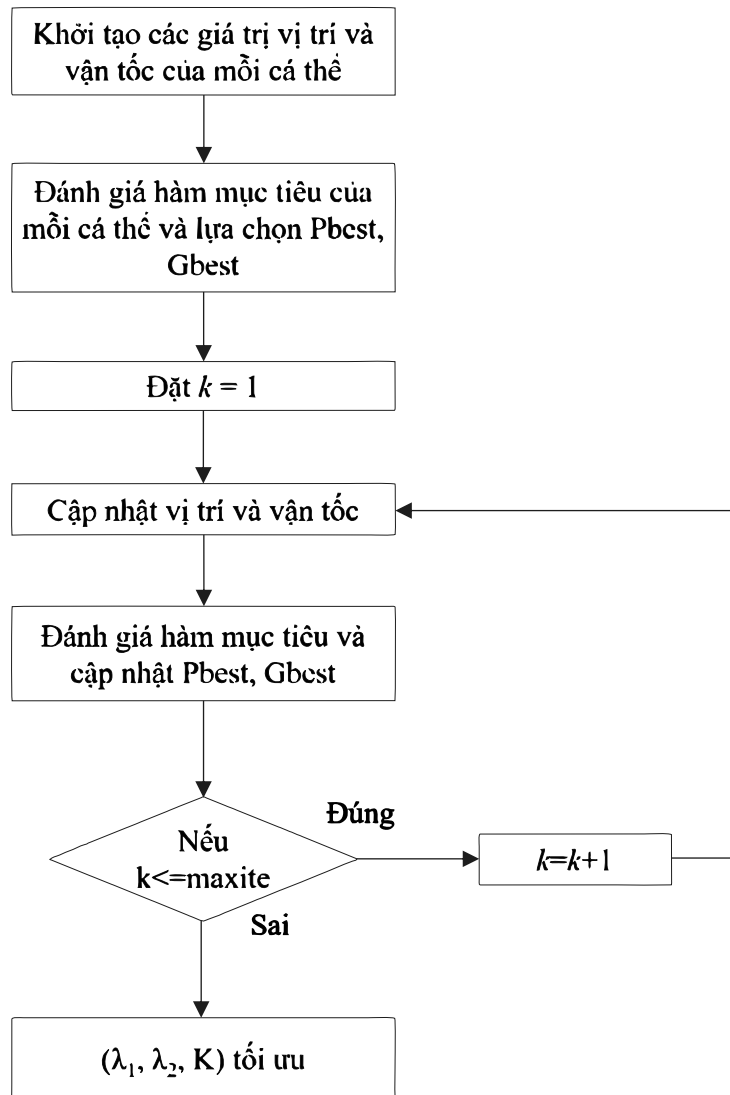
Trong đó, $X_{i,j}^k$ và $V_{i,j}^k$ là vị trí và vận tốc của phần tử ở lần lặp thứ k , w là hệ số trọng lượng quán tính, c_1, c_2 là hệ số gia tốc, r_1 và r_2 là số ngẫu nhiên.

3.4. Điều chỉnh bộ điều khiển trượt-PID dựa trên giải thuật bầy đàn

Trong giải thuật bầy đàn thì mỗi phần tử sẽ chứa 3 tham số λ_1, λ_2 và K Điều đó có nghĩa là không gian tìm kiếm là 3 thông số này. Từ đó, lưu đồ giải thuật của hệ điều khiển PSO PID-SMC được trình bày như Hình 4. Hàm mục tiêu được chọn như sau:

$$J = \int_0^\infty |e(t)| dt \quad (12)$$

Mục tiêu của phương pháp điều chỉnh PID-SMC sử dụng giải thuật bầy đàn là cực tiểu hóa hàm mục tiêu và tìm được đáp ứng tối ưu cho hệ thống trong điều kiện đối tượng có thành phần phi tuyến dead-zone và nhiễu.



Hình 4. Lưu đồ giải thuật bầy đàn tối ưu thông số bộ điều khiển PID-SMC

3.5. Kết quả mô phỏng

Để thấy rõ khả năng điều khiển của bộ điều khiển SMC (mặt trượt kiểu PID) được tối ưu bằng giải thuật bầy đàn, kết quả mô phỏng sẽ được khảo sát với dead-zone trong khoảng [0, 0.5] thông số hệ thống biến thiên và dưới tác động của nhiễu.

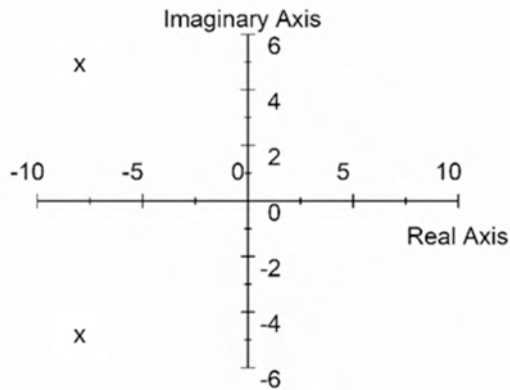
Các tham số đầu vào của giải thuật bầy đàn (PSO) được thể hiện trong Bảng 2. Kích thước quần thể không được nhỏ hơn số cá thể khởi tạo, nếu nhỏ hơn sẽ gây ra lỗi do không đủ không gian tìm kiếm. Số lần lặp dựa trên kinh nghiệm của mỗi lần tối ưu, nếu lặp quá lớn sẽ mất nhiều thời gian tìm kiếm và ngược lại độ hội tụ sẽ thấp dẫn đến giải thuật chưa tối ưu. Trọng số khởi tạo w được chọn là 0.5 và các thông số c_1, c_2 được chọn bằng 2, hệ số r_1 và r_2 được chọn ngẫu nhiên trong khoảng [0, 1].

Bảng 2. Thông số đầu vào cho giải thuật PSO

Thông số thuật toán PSO	Giá trị
Số lần lặp	60
Kích thước quần thể	50
Trọng số quán tính w	0.5
Hệ số (c_1, c_2)	(2, 2)
Hệ số ngẫu nhiên r_1, r_2	[0, 1]
Giá trị ngưỡng dưới của $(\lambda_1, \lambda_2, K)$	(1, 1, 1)
Giá trị ngưỡng dưới của $(\lambda_1, \lambda_2, K)$	(100, 100, 100)

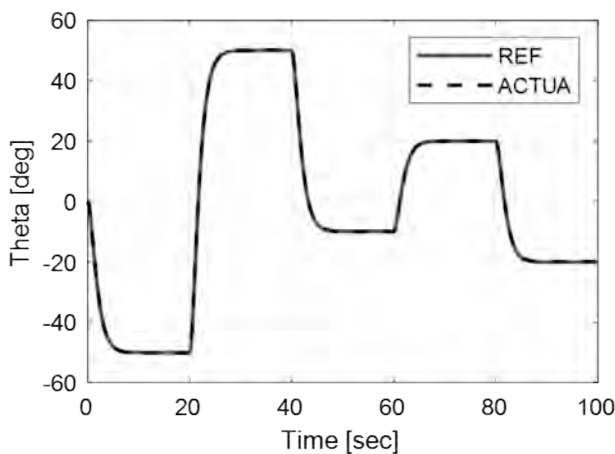
Tiến hành kiểm chứng đáp ứng của bộ điều khiển được đề xuất, tác giả khai báo các thông số mô hình điều khiển như Bảng 1 và các giá trị được tối ưu bởi giải thuật bầy đàn là $\lambda_1 = 16, \lambda_2 = 88$ và $K = 13$.

Dựa theo mặt phẳng phức, ta có thể thấy, việc tìm kiếm hai thông số $\lambda_1 = 16, \lambda_2 = 88$ thỏa điều kiện để hệ thống ổn định.

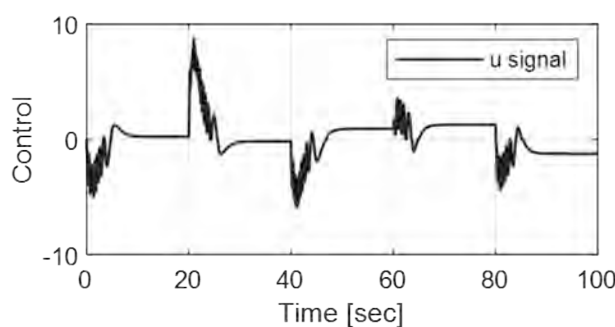


Hình 5. Mặt phẳng phức

Đáp ứng của hệ thống và tín hiệu điều khiển khi tín hiệu đặt là hình nấc được thể hiện như Hình 6 và Hình 7.

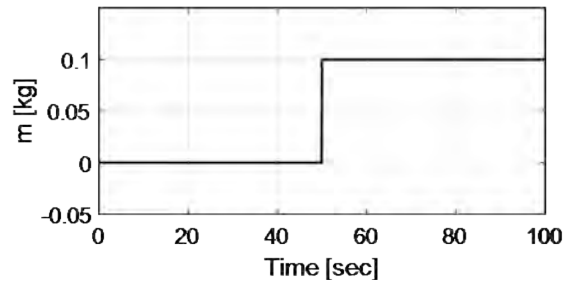


Hình 6. Đáp ứng hệ thống khi tín hiệu đặt hình nấc

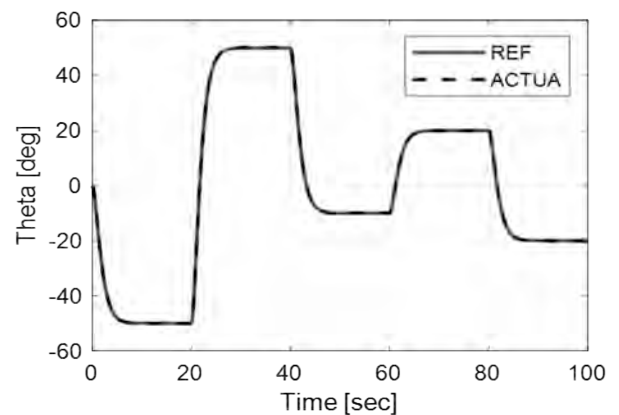


Hình 7. Tín hiệu điều khiển khi tín hiệu đặt hình nấc

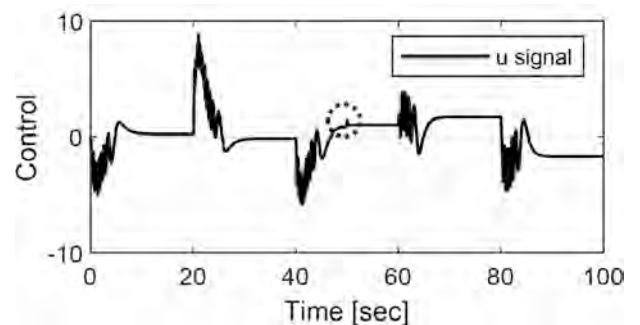
Để khảo sát tính bền vững của bộ điều khiển, tác giả đã giả sử dụng khối lượng vật gấp của tay máy thay đổi theo thời gian (được trình bày như Hình 8), cụ thể là ban đầu tay máy sẽ hoạt động không tải, sau đó sẽ gấp một vật nặng có khối lượng $m = 0.1\text{kg}$. Kết quả mô phỏng đáp ứng và tín hiệu điều khiển khi khối lượng vật gấp thay đổi được trình bày như Hình 9 và Hình 10.



Hình 8. Khối lượng của vật gấp thay đổi theo thời gian

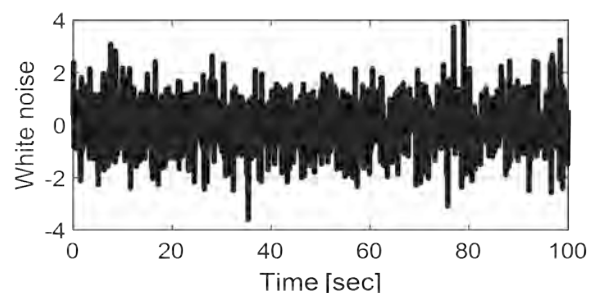


Hình 9. Đáp ứng hệ thống khi tải trọng thay đổi theo thời gian

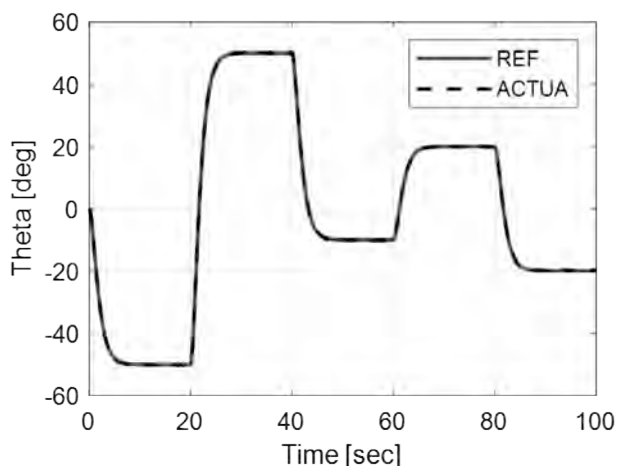


Hình 10. Tín hiệu điều khiển khi hệ thống tải trọng thay đổi

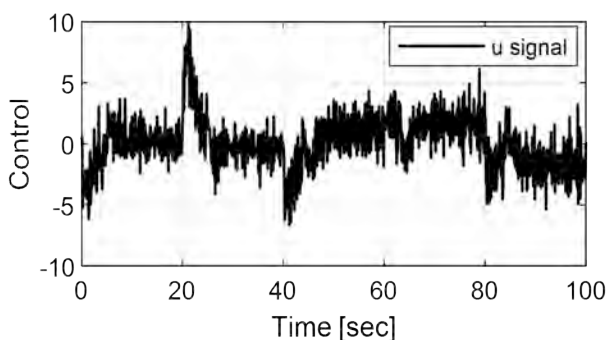
Để kiểm tra sự ổn định của bộ điều khiển dưới tác động của nhiễu và thành phần dead-zone, tác giả đã giả sử hệ thống chịu tác động bởi nhiễu trắng có biên độ dao động được trình bày như Hình 11 và thành phần dead-zone. Kết quả mô phỏng đáp ứng và tín hiệu điều khiển khi dưới tác động của nhiễu và thành phần dead-zone tác động được trình bày như Hình 12 và Hình 13.



Hình 11. Tín hiệu nhiễu trắng

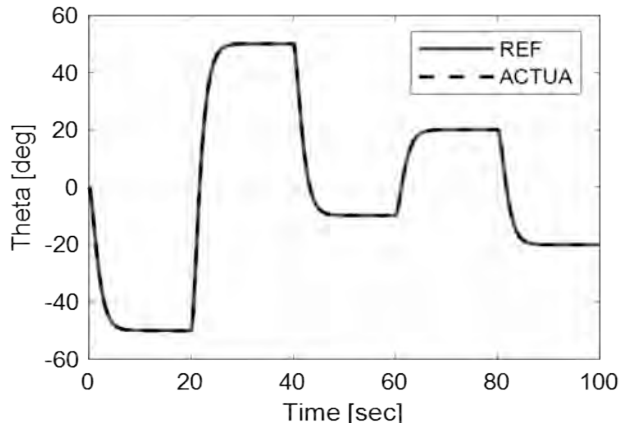


Hình 12. Đáp ứng hệ thống khi có nhiễu và thành phần dead-zone tác động

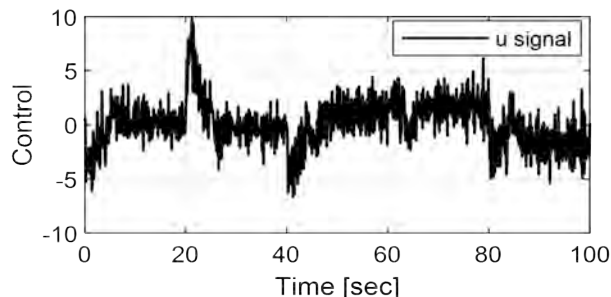


Hình 13. Tín hiệu điều khiển khi hệ thống dưới tác động của nhiễu trắng và thành phần dead-zone

Nhằm khẳng định chất lượng của bộ điều khiển được đề xuất tác giả đã giả sử hệ thống gần với thực tế là tải trọng sẽ thay đổi, dưới tác động của nhiễu và có thành phần dead-zone. Kết quả mô phỏng được trình bày như Hình 14 và Hình 15.



Hình 14. Đáp ứng của hệ thống khi tải trọng thay đổi, nhiễu trắng và thành phần dead-zone tác động



Hình 15. Tín hiệu điều khiển khi hệ thống tải trọng thay đổi nhiễu trắng, và thành phần dead-zone tác động

Từ các Hình 6, 9, 12, và 14 cho thấy, khi tay máy chuyển từ trạng thái không tải sang có tải hay chịu tác động của nhiễu, thành phần dead-zone hay bị ảnh hưởng của cả ba yếu tố trên thì tín hiệu điều khiển sẽ bị ảnh hưởng, tuy nhiên tín hiệu thực vẫn bám theo tín hiệu tham khảo. Vẫn đảm bảo các tiêu chí chất lượng hệ thống như độ vọt lố, thời gian tăng, thời gian xác lập và sai số xác lập. Riêng đối với bộ điều khiển trượt với mặt trượt dạng PID thì tín hiệu đáp ứng không còn bám so với tín hiệu đặt khi có thành phần nhiễu và dead-zone.

Bảng 3. Kết quả so sánh giữa bộ điều khiển PID-SMC-PSO với bộ điều khiển PID-SMC

Loại điều khiển	Độ vọt lố (%)	Thời gian xác lập (s)	Thời gian tăng (s)
PID-SMC	0.02	3.2	1.7
PID-SMC-PSO	0	1.2	0.5
PID-SMC-PSO khi có nhiễu và dead-zone	0	1.5	0.9

Kết quả được so sánh cụ thể trên Bảng 3 cho thấy bộ điều khiển PID-SMC-PSO hoạt động tốt, đáp ứng cải thiện tốt hơn PID-SMC với độ vọt lố là 0%, đạt trạng thái xác lập sau 1.2s và thời gian tăng là 0.5s. Hơn nữa, khi có tác động của nhiễu và dead-zone, bộ điều khiển PID-SMC-PSO vẫn hoạt động ổn định với độ vọt lố là 0%, thời gian xác lập là 1.5s, và thời gian tăng là 0.9s.

4. KẾT LUẬN

Trong nghiên cứu này, tác giả đã đề xuất một giải pháp là sử dụng giải thuật bầy đàn để tối ưu các tham số của bộ điều khiển trượt với hàm trượt dạng PID.

Các kết quả mô phỏng cho thấy tín hiệu thực tế bám tốt so với tín hiệu đặt, không có độ vọt lố và sai số xác lập gần như được triệt tiêu. Song, hiệu

quả của bộ điều khiển được đề xuất còn được kiểm chứng khi điều khiển góc khớp của mô hình tay máy một bậc tự do bám theo tín hiệu đặt trước ngay cả trong khi tải trọng thay đổi và dưới sự tác động của nhiễu và thành phần dead-zone. Ngoài

ra, những kết quả mô phỏng đạt được trong nghiên cứu này còn là cơ sở để phát triển bộ điều khiển PID-SMC cho các mô hình phức tạp như robot ba bánh đa hướng, tay máy nhiều bậc tự do trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] K.D. Young, V.I. Utkin, and U. Ozguner, "A control engineer's guide to sliding mode control," *IEEE Transactions on Control Systems Technology*, 7, pp.328-342, 1999.
- [2] L. Yangming, "Adaptive sliding mode control with perturbation estimation and PID sliding surface for motion tracking of a piezo-driven micromanipulator," *IEEE Transactions on Control Systems Technology*, 18, pp.798-810, 2010.
- [3] Nguyễn Hoàng Dũng, "Điều khiển trượt dựa trên hàm trượt kiểu PID," *Tạp chí Khoa học Trường Đại học Cần Thơ*, 21a 30-36, 2012.
- [4] S.Y. Oh and D.J. Park, "Design of new adaptive fuzzy logic controller for nonlinear plants with unknown or time-varying dead zones," *IEEE Transactions on Fuzzy Systems*, 6, pp.482-491, 1998.
- [5] J.E. Normey-Rico and E.F. Camacho, *Control of dead-time processes*, Place of publication: Springer, 2007.
- [6] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, 83, pp.80-98, 2015.
- [7] S. Mirjalili, S.M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, 69, pp.46-61, 2014.
- [8] L. Fatima and K. Sihem, *Recent advances in engineering mathematics*, Place of publication: Springer, 1st ed., 2020, pp.19-32.
- [9] D. Mokeddem and H. Draïdi, "Optimization of PID sliding surface using Antlion optimizer," *International Symposium on Modeling and Implementation of Complex Systems (MISC)*, pp.133-145, 2018.
- [10] L. Tsoukalas and R. Uhrig, *Fuzzy and neural approaches in engineering*, Place of publication: Wiley, 1997.
- [11] R. Eberhart and J. Kennedy, "Particle swarm optimization," *Proceeding of the IEEE International Conference on Neural Networks*, pp.1942-1948, 1995.
- [12] Huỳnh Thái Hoàng, *Hệ thống điều khiển thông minh*. Thành phố Hồ Chí Minh: Nhà xuất bản Đại học Quốc Gia TP. Hồ Chí Minh, 2006.
- [13] T. Gang, Adaptive control of nonsmooth dynamic systems, Place of publication: Springer, 2001.

An optimized sliding mode control with pid sliding surface using partical swam optimization algorithm

Huynh Minh Vu and Nguyen Dinh Tu

ABSTRACT

The sliding mode controller with PID (Proportional Integral Derivative) sliding surface provides the stability and robustness by decreasing chattering phenomenon. However, the traditional procedure generally involves trial and error methods in designing the sliding mode controller with PID sliding surface. Therefore, an improved selection procedure of parameters for sliding mode controller with PID sliding surface is necessary to improve the system performance. This study proposes an approach to optimize the sliding mode controller with PID sliding surface using the particle swarm optimization for the one degree of freedom manipulator. Simulations are performed using MATLAB to demonstrate

the stability and robustness of the proposed controller by exhibiting no overshoot/steady-state error in dealing with dead-zone and unexpected disturbance.

Keywords: *Sliding mode control with PID sliding surface, Particle swarm optimization, one degree of freedom manipulator, Dead-zone, Disturbance*

Received: 16/10/2023

Revised: 13/11/2023

Accepted for publication: 14/11/2023