

# ĐIỀU KHIỂN HỆ PENDUBOT SỬ DỤNG DẠNG TOÀN PHƯƠNG TUYẾN TÍNH DỰA TRÊN LOGIC MỜ

• Nguyễn Thành Nguyên<sup>(\*)</sup>, Nguyễn Phong Lưu<sup>(\*\*)</sup>, Nguyễn Văn Đông Hải<sup>(\*\*)</sup>

## Tóm tắt

Mô hình Pendubot là mô hình có ngõ vào điều khiển ít hơn số bậc tự do, có độ phi tuyến cao và rất khó để điều khiển. Pendubot với cấu trúc cơ khí không quá phức tạp nên được nhiều nhà nghiên cứu sử dụng để kiểm tra giải thuật điều khiển trong các phòng thí nghiệm. Trong bài báo này, tác giả sử dụng giải thuật toàn phương tuyến tính dựa trên logic mờ (Fuzzy Linear Quadratic Regulator) để giải quyết bài toán cân bằng tại vị trí TOP của mô hình Pendubot. Ý tưởng chính của phương pháp này là sử dụng giải thuật mờ kết hợp với giải thuật di truyền để lựa chọn thông số điều khiển cho bộ điều khiển tuyến tính bậc hai (Linear Quadratic Regulator). Thông qua mô phỏng trên Matlab/Simulink và thực nghiệm điều khiển tại vị trí TOP với thanh 1 và thanh 2 lần lượt tại 90 độ và 0 độ.

Từ khóa: Pendubot, mô hình SIMO, giải thuật LQR, giải thuật mờ, điều khiển cân bằng.

## 1. Giới thiệu

Pendubot được nghiên cứu trong đề tài bao gồm 2 thanh đồng chất, đầu 1 được gắn chặt vào động cơ DC. Đầu cuối thanh được gắn vào đầu thanh 2 sao cho thanh 2 quay tự do xung quanh thanh nối với thanh 1. Mô hình Pendubot được ứng dụng trong rất nhiều phòng thí nghiệm nhằm nghiên cứu các giải thuật điều khiển thông minh, điều khiển hiện đại.

Tác giả Cao Văn Kiên trong bài báo (năm 2016) [1] nghiên cứu giải thuật trượt để Swingup và cân bằng hai thanh của hệ thống Pendubot tại vị trí TOP. Tác giả Phan Việt Hùng trong [5] nghiên cứu bộ điều khiển Swingup và Balancing sử dụng phương pháp PID, LQR để ổn định hệ thống Pendubot tại vị trí TOP. Tác giả mong muốn thiết kế giải thuật Fuzzy-LQR lên hệ Pendubot để kiểm chứng bộ điều khiển.

Ở bài báo này, nhóm tác giả sẽ đề cập đến phương pháp điều khiển Fuzzy-LQR để điều khiển vị trí TOP cho hệ pendubot. Đầu tiên, tác giả thiết kế mô hình toán học của hệ thống lên Matlab/ Simulink kết hợp với bộ điều khiển LQR để tìm ra các thông số K để điều khiển Pendubot ổn định. Sau đó, lấy thông số K để tính toán các E (lỗi) và EC (tỉ lệ lỗi) và tiếp tục đưa vào mô phỏng Matlab/Simulink kết hợp giải thuật di truyền (GA) [4] và giải thuật Fuzzy để tìm ra các thông số của bộ tiền xử lý và hậu xử lý cho hệ thống.

## 2. Phương trình động lực học

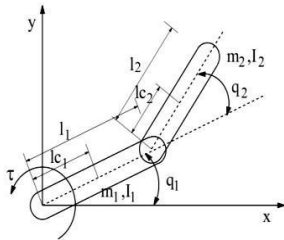
Theo [6], ta có cấu trúc của mô hình pendubot như Hình 1. Thông số của hệ thống được thể hiện ở Bảng 1.

Bảng 1. Thông số hệ thống

Ký hiệu	Đơn vị	Mô tả
I1	kg.m <sup>2</sup>	Mô-men quán tính thanh 1
I2	kg.m <sup>2</sup>	Mô-men quán tính thanh 2
m <sub>2</sub>	kg	Khối lượng thanh 2
l <sub>2</sub>	m	Chiều dài thanh 2
l <sub>c2</sub>	m	Khoảng cách từ trục encoder đến trọng tâm thanh 2
m <sub>1</sub>	kg	Khối lượng thanh 1
l <sub>1</sub>	m	Chiều dài thanh 1
l <sub>c1</sub>	m	Khoảng cách trục động cơ đến trọng tâm thanh 1
G	m/s <sup>2</sup>	Gia tốc trọng trường
K <sub>t</sub>	Nm/A	Hằng số mô-men
K <sub>b</sub>	V/(rad/s)	Hằng số phản điện
R <sub>m</sub>	Ohm	Điện trở động cơ
J <sub>m</sub>	Kg.m <sup>2</sup>	Mô-men quán tính rotor
C <sub>m</sub>	Nm/(rad/s)	Hệ số ma sát nhớt

<sup>(\*)</sup> Sinh viên, Trường Đại học Tôn Đức Thắng.

<sup>(\*\*)</sup> Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh.



**Hình 1. Cấu trúc của mô hình pendubot**

Dùng phương pháp Euler-Lagrange [5], ta có phương trình động lực học của hệ thống pendubot như sau:

$$M(q)\dot{q} + B(q, \dot{q})\dot{q} + G(q) = \tau \quad (1)$$

với  $q = [q_1 \quad q_2]^T$ ,  $q_1$  là góc giữa thanh 1 và trục hoành (Ox),  $q_2$  là góc giữa thanh 1 và thanh 2,  $\tau = [\tau_m \quad 0]^T$  là tín hiệu điều khiển, mô-men của động cơ đặt vào thanh 1;

$$M(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos(q_2) & \beta_2 + \beta_3 \cos(q_2) \\ \beta_2 + \beta_3 \cos(q_2) & \beta_2 \end{bmatrix} \quad (2)$$

$$B(q, \dot{q}) = \begin{bmatrix} -\beta_3 \sin(q_2)\dot{q}_2 & -\beta_3 \sin(q_2)\dot{q}_2 - \beta_3 \sin(q_2)\dot{q}_1 \\ \beta_3 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix} \quad (3)$$

$$G(q) = \begin{bmatrix} \beta_4 g \cos(q_1) + \beta_5 g \cos(q_1 + q_2) \\ \beta_5 g \cos(q_1 + q_2) \end{bmatrix} \quad (4)$$

$$\beta_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1; \beta_2 = m_2 l_{c2}^2 + I_2; \\ \beta_3 = m_2 l_1 l_{c2}; \beta_4 = m_1 l_{c1} + m_2 l_1; \beta_5 = m_2 l_{c2}.$$

Do tín hiệu điều khiển của hệ thống pendubot trong thực tế là điện áp cấp cho động cơ nên việc tìm mối liên quan giữa mô-men và điện áp là cần thiết. Theo [7] và qua một số phép biến đổi đơn giản, ta có được phương trình sau:

$$\tau_m = \frac{K_t}{R_m} u - \left( \frac{K_t K_b}{R_m} + C_m \right) \dot{q} - J_m \dot{q}_1 \quad (5)$$

Đặt  $K_1 = \frac{K_t}{R_m}; K_2 = \frac{K_t K_b}{R_m} + C_m; K_3 = J_m;$

$$x_1 = q_1; x_2 = \dot{q}_1; x_3 = q_2; x_4 = \dot{q}_2;$$

Kết hợp với (1), ta được:

$$\dot{x}_1 = x_2 \quad (6)$$

$$\dot{x}_2 = f_1(x) + b_1(x)u \quad (7)$$

$$\dot{x}_3 = x_4 \quad (8)$$

$$\dot{x}_4 = f_2(x) + b_2(x)u \quad (9)$$

với:

$$f_1(x) = \frac{1}{K_3 \beta_2 + \beta_1 \beta_2 - \beta_3^2 \cos^2(x_3)}. \quad (10)$$

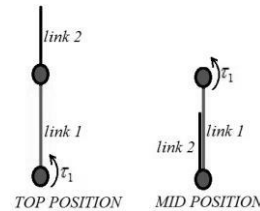
$$\begin{bmatrix} \beta_2 \beta_3 \sin(x_3)(x_2 + x_4)^2 + \beta_3^2 \sin(x_3) \cos(x_3)x_2^2 \\ -\beta_2 \beta_4 g \cos(x_1) + \beta_3 \beta_5 g \cos(x_3) \cos(x_1 + x_3) - K_2 \beta_2 x_2 \end{bmatrix}$$

$$b_1(x) = \frac{\beta_2 K_1}{K_3 \beta_2 + \beta_1 \beta_2 - \beta_3^2 \cos^2(x_3)} \quad (11)$$

$$f_1(x) = \frac{1}{K_3 \beta_2 + \beta_1 \beta_2 - \beta_3^2 \cos^2(x_3)}. \quad (12)$$

$$\begin{bmatrix} (-\beta_3 \sin(x_3)(\beta_2 + \beta_3 \sin(x_3))(x_2 + x_4))^2 \\ -\beta_3 \sin(x_3)(\beta_1 + \beta_3 \cos(x_3))x_2^2 \\ +\beta_2 \beta_4 g \cos(x_1)(\beta_2 + \beta_3 \cos(x_3)) \\ -\beta_5 g (K_3 + \beta_1 + \beta_3 \cos(x_3)) \cos(x_1 + x_3) \\ +K_2(\beta_2 + \beta_3 \cos(x_3))x_2 - K_3 \beta_3 x_2^2 \sin(x_3) \end{bmatrix}$$

$$b_2(x) = \frac{-\beta_2 - \beta_3 \cos(x_3)K_1}{K_3 \beta_2 + \beta_1 \beta_2 - \beta_3^2 \cos^2(x_3)} \quad (13)$$



**Hình 2. Vị trí cân bằng pendubot**

Hai vị trí cân bằng pendubot [6] phổ biến trong việc điều khiển hệ pendubot là vị trí TOP và vị trí MID như trong Hình 2. Trong bài báo này, nhóm tác giả điều khiển pendubot tại vị trí TOP. Ta đặt biến trạng thái mới  $X = [X_1 \quad X_2 \quad X_3 \quad X_4]^T$  để thay thế  $x = [x_1 \quad x_2 \quad x_3 \quad x_4]^T$ . Mục đích là để lái biến trạng thái X về 0 khi hệ thống ổn định. Theo hệ trục tọa độ như Hình 1, X sẽ được

$$\text{đặt như sau: } X_1 = x_1 - \frac{\pi}{2}; \quad X_2 = x_2; \quad X_3 = x_3;$$

$$X_4 = x_4. \text{ Thay } x_1 = X_1 + \frac{\pi}{2}; \quad x_2 = X_2; \quad x_3 = X_3;$$

$x_4 = X_4$  vào  $f_1(x)$ ,  $b_1(x)$ ,  $f_2(x)$ ,  $b_2(x)$  ta được  $f_1(X)$ ,  $b_1(X)$ ,  $f_2(X)$ .

### 3. Giải thuật điều khiển

#### 3.1. Giải thuật LQR

Phương trình trạng thái:

$$\dot{x} = Ax + Bu \quad (14)$$

$$J = \frac{1}{2} \int_0^{\infty} (x^T Qx + u^T Ru) dt \quad (15)$$

$$u(t) = -Kx(t) \quad (16)$$

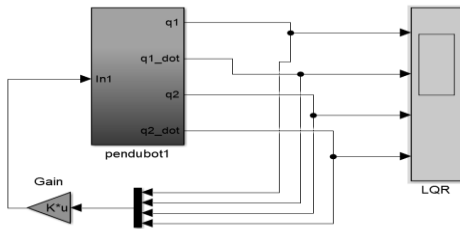
Với  $K$  là ma trận thu hồi thông tin trạng thái được xác định bởi công thức:

$$K = R^{-1}B^T S \quad (17)$$

Trong đó  $S$  là nghiệm của phương trình Riccati:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \quad (18)$$

Trong bài báo này, tác giả chọn  $Q = [1 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0; 0 \ 0 \ 1 \ 0; 0 \ 0 \ 0 \ 1]$ ,  $R = 1$ .

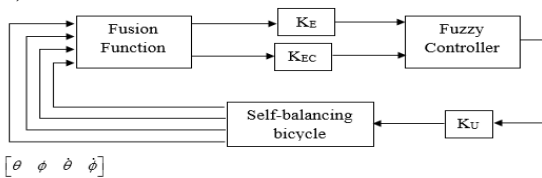


Hình 3. Sơ đồ bộ điều khiển LQR

### 3.2. Bộ điều khiển Fuzzy-LQR

#### 3.2.1. Kết hợp các biến

Ý chính là sử dụng kỹ thuật kết hợp để các biến của hệ thống thành sai số ( $E$ ) và tỉ lệ lỗi ( $EC$ )



Hình 4. Sơ đồ điều khiển Fuzzy

Mô tả cấu trúc điều khiển Fuzzy-LQR trong đó  $K_E$ ,  $K_{EC}$ ,  $K_U$  là các hệ số tiêu chuẩn hóa.

Hàm kết hợp được định nghĩa:

$$F(X) = \begin{bmatrix} \frac{K_\theta}{\|K\|_2} & \frac{K_\phi}{\|K\|_2} & 0 & 0 \\ 0 & 0 & \frac{K_{\dot{\theta}}}{\|K\|_2} & \frac{K_{\dot{\phi}}}{\|K\|_2} \end{bmatrix} \quad (19)$$

Trong đó,  $K = [K_\theta \ K_\phi \ K_{\dot{\theta}} \ K_{\dot{\phi}}]$  là ma trận phản hồi trạng thái được thu thập bởi vấn đề LQR ở trên.

$$\|K\|_2 = \sqrt{K_\theta^2 + K_\phi^2 + K_{\dot{\theta}}^2 + K_{\dot{\phi}}^2} \quad (20)$$

Thay thế  $K = [-133,2851 \ -25,8212 \ -136,9267 \ -15,5376]$  vào (21), ta được:

$$F(X) = \begin{bmatrix} -0,6890 & -0,1335 & 0 & 0 \\ 0 & 0 & -0,7078 & -0,0803 \end{bmatrix}$$

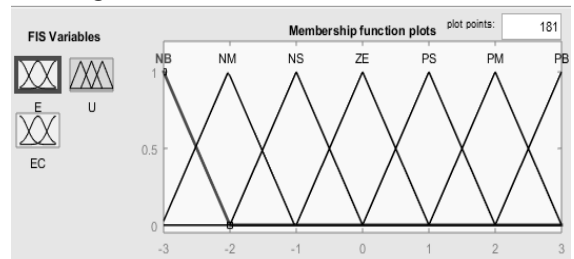
Lỗi  $E$  và tỉ lệ lỗi  $EC$  được tính:

$$\begin{bmatrix} E \\ EC \end{bmatrix} = F(X)X^T \quad (21)$$

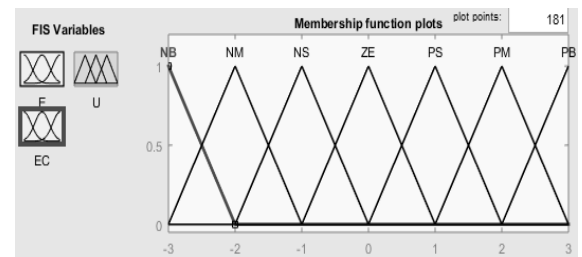
Trong trường hợp:  $\begin{cases} E = K_\theta \theta + K_\phi \phi \\ EC = K_{\dot{\theta}} \dot{\theta} + K_{\dot{\phi}} \dot{\phi} \end{cases}$

#### 3.2.2. Fuzzy-LQR

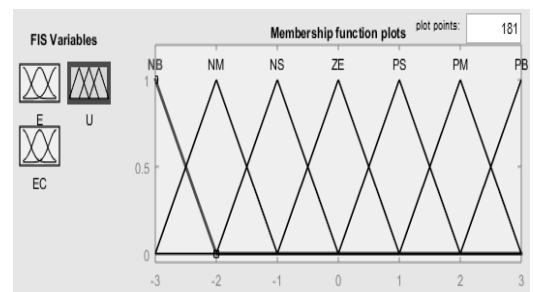
Bộ điều khiển mờ có 2 đầu vào  $E$  và  $EC$ , một đầu ra  $U$ . Các hàm thành viên đầu vào và đầu ra là hình tam giác và có 7 hàm liên thuộc như trong Hình 5, Hình 6 và Hình 7.



Hình 5. Môi quan hệ của hàm E



Hình 6. Môi quan hệ của hàm EC



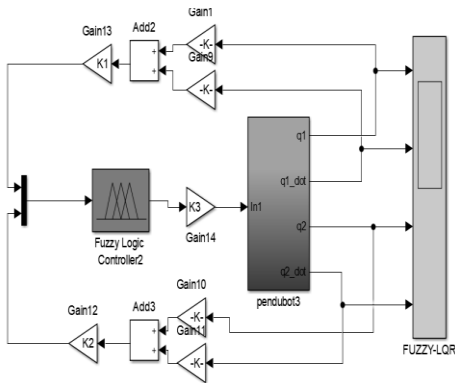
Hình 7. Môi quan hệ ngõ ra U

Bộ qui tắc luật mờ được chọn như Hình 8.

	EC	NB	NM	NS	ZE	PS	PM	PB
E								
NB		NB	NB	NB	NM	NM	NS	ZE
NS		NB	NB	NM	NM	NS	ZE	PS
NM		NB	NM	NM	NS	ZE	PS	PM
ZE		NM	NM	NS	ZE	PS	PM	PM
PS		NM	NS	ZE	PS	PM	PM	PB
PM		NS	ZE	PS	PM	PM	PB	PB
PB		ZE	PS	PM	PM	PB	PB	PB

Hình 8. Các qui tắc luật mờ

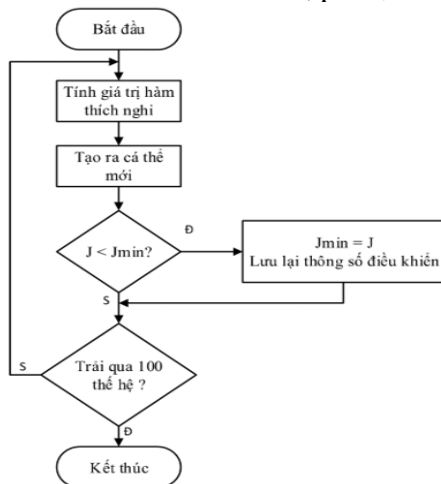
Sơ đồ bộ điều khiển Fuzzy-LQR được hiển thị trong Hình 9.



Hình 9. Sơ đồ điều khiển Fuzzy-LQR Simuthanh

Để tìm thông số tiền xử lý, hậu xử lý K1, K2, K3 trong Hình 9, ta sử dụng GA [4]. Các thông số GA được lựa chọn như sau:

- Số thế hệ tối đa: 20000;
- Số lượng cá thể: 1000;
- Hệ số lai ghép: 0,8;
- Hệ số đột biến: 0,2;
- Kiểu mã hóa: mã hóa nhị phân;



Hình 10. Lưu đồ giải thuật di truyền

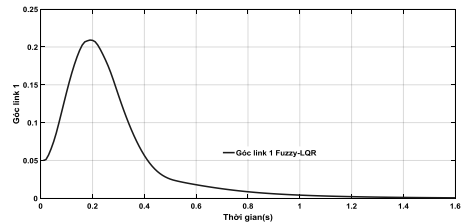
Sau khi kết quả 1000 thế hệ thu được:

- K1 = -9,0766;
- K2 = -8,0574;
- K3 = 32,3458;

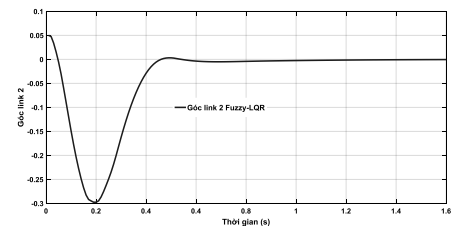
#### 4. Kết quả nghiên cứu

##### 4.1. Mô phỏng Matlab/Simuthanh

Mô phỏng được thực hiện trên Matlab/Simuthanh. Các giá trị ban đầu của góc thanh 1, vận tốc góc thanh 1, góc thanh 2, vận tốc góc thanh 2 lần lượt là 0 (rad); 0,05 (rad/s); 0 (rad); 0,05 (rad/s).



Hình 11. Kết quả mô phỏng thanh 1 trên Matlab/Simuthanh

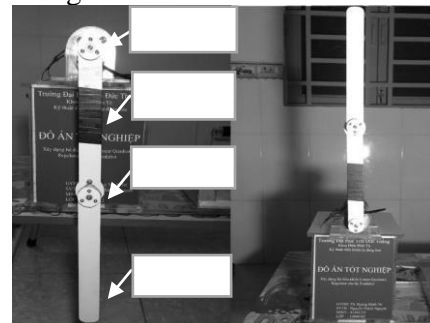


Hình 12. Kết quả mô phỏng thanh 2

Kết quả mô phỏng Hình 11, Hình 12 nhận thấy đáp ứng hệ thống khá tốt: thời gian xác lập của thanh 1 và thanh 2 lần lượt là 1,4s và 1,5s. Hệ thống có độ vọt lố nhỏ: độ vọt lố thanh 1 và thanh 2 lần lượt là 0,2 rad và 0,28 rad.

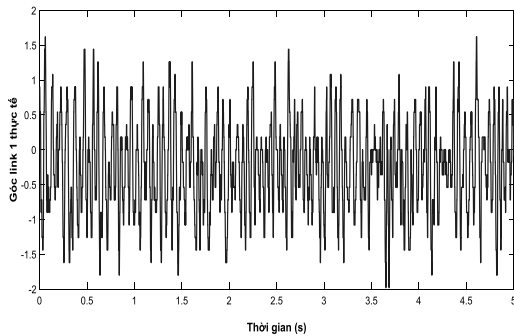
##### 4.2. Kết quả thực nghiệm

Mô hình pendubot thực tế được xây dựng theo Hình 13. Kết quả thực nghiệm được thể hiện trong Hình 14-16. Vị trí của pendubot trước khi được điều khiển thẳng đứng tại vị trí Top của khớp 1 và khớp 2 lần lượt là  $(-\frac{\pi}{2}; 0)$ . Sau đó, áp dụng giải thuật đề nghị để điều khiển pendubot dựng thẳng đứng với góc mong muốn của khớp 1 và khớp 2 lần lượt là  $(\frac{\pi}{2}; 0)$ . Đáp ứng thực tế cho thấy hệ thống ổn định với góc lệch tương đối nhỏ: góc thanh 1 trong khoảng  $\pm 2^\circ$ , góc thanh 2 trong khoảng  $\pm 3^\circ$  và điện áp cấp cho động cơ trong khoảng  $\pm 15V$ .

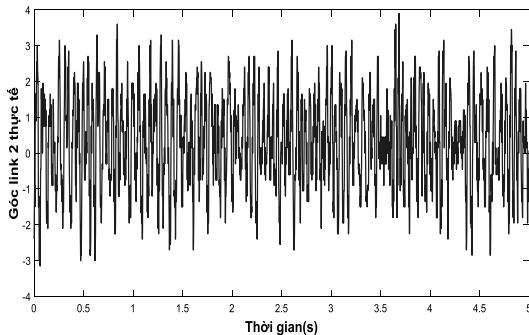


- a) Trạng thái nghỉ
- b) Trạng thái ổn định ở vị trí Top

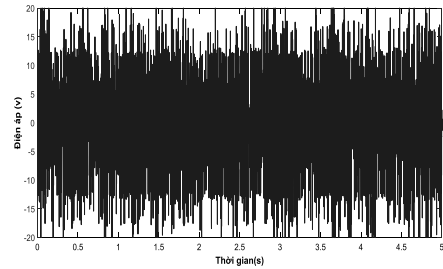
Hình 13. Mô hình thực tế pendubot



Hình 14. Góc thanh 1 thực tế (degree)



Hình 15. Góc thanh 2 thực tế (degree)



Hình 16. Điện áp cấp động cơ (V)

## 5. Kết luận

Sử dụng bộ điều khiển toàn phương tuyến tính sử dụng logic mờ thông qua mô phỏng hệ thống điều khiển cân bằng tại vị trí Top, theo thực nghiệm điều khiển bị lệch cụ thể thanh 1 lệch dao động trong khoảng  $\pm 2^\circ$  và thanh 2 lệch dao động trong khoảng  $\pm 3^\circ$  so với mong muốn tại vị trí Top. Nguyên nhân phần lớn bị lệch ở thực tế là do nhiều môi trường, phần cơ khí đặc biệt là encoder có độ sai số lớn để đảm bảo hệ thống điều khiển như mong muốn cần phải chọn encoder có độ chính xác cực kỳ cao khoảng 1000 xung và mô hình hoạt động trong môi trường lý tưởng./.

## Tài liệu tham khảo

- [1]. Van Kien Cao, Ngoc Son Nguyen, Pham Huy Anh Ho (2015), “Swing up and balancing implementation for the pendubot using Advanced Sliding Mode Control”, *International Conference of Electrical, Automation and Mechanical Engineering*, pp. 389-392.
- [2]. Vũ Đình Đạt, Huỳnh Xuân Dũng, Phan Văn Kiểm, Nguyễn Minh Tâm, Nguyễn Văn Đông Hải (2017), “Điều khiển mờ-trượt cho hệ pendubot”, *Tạp chí công nghệ Đà Nẵng*, (số 11), tr. 12-16.
- [3]. Nguyễn Thị Phương Hà (2016), *Lý thuyết điều khiển hiện đại*, NXB Đại học Quốc gia Thành phố Hồ Chí Minh.
- [4]. Huỳnh Thái Hoàng (2014), *Hệ thống điều khiển thông minh*, NXB Đại học Quốc gia Thành phố Hồ Chí Minh.
- [5]. Phan Việt Hùng (2013), *Nghiên cứu cánh tay Robot thiếu dẫn động hai bậc tự do- Pendubot*, Luận án Thạc Sĩ, Trường Đại học Đà Nẵng, Đà Nẵng.
- [6]. Xuan Dung Huynh, Duong Khanh Linh Huynh, Dinh Dat Vu, Thanh Phuong Nguyen, Minh Tam Nguyen, Van Dong Hai Nguyen (2017), “Application of fuzzy algorithm in optimizing hierarchical sliding mode control for pendubot system”, *International Journal of Robotica & Management*, No.2, pp. 8-12.
- [7]. Hoang Chinh Tran, Minh Tam Nguyen, Van Dong Hai Nguyen (2017), “Application of PID-FUZZY control for pendubot”, *Journal of Technical Education Science*, (44A), pp. 61-67.

## PENDUBOT CONTROLS USING LINEAR QUADRATIC REGULATOR COMBINED FUZZY

### Summary

Pendubot is a model with control input smaller than the number of free levels and a high degree of nonlinearity, which is difficult to control. Since Pendubot does not have a very complicated mechanical structure, many researchers use it to test control algorithms in laboratories. In this paper, the authors use linear algorithms based on Fuzzy Linear Quadratic Regulator to solve the equilibrium problem at the TOP position of Pendubot model. The main idea of this method is to use fuzzy algorithms associated with genetic algorithms to select control parameters for the Linear Quadratic Regulator. Simulations on Matlab Simulink and experiments in TOP position yield linking 1 and 2 at 90 degrees and 0 degrees, respectively.

Keywords: Pendubot; SIMO model; LQR algorithm; fuzzy algorithm; balance control.

Ngày nhận bài: 13/02/2019; Ngày nhận lại: 25/4/2019; Ngày duyệt đăng: 07/5/2019.