

Điều Khiển Swing-up và Cân Bằng Hệ Con Lắc Ngược Quay Áp Dụng Bộ Điều Khiển LQR-based ANFIS

Swing-up and Stabilization Control of Rotary Inverted Pendulum by using LQR-based ANFIS

Trần Minh Đức¹, Võ Minh Tài¹, Phạm Quang Minh², Nguyễn Đức Trung², Võ Hữu Thống², Trần Nho Thê², Lê Đình Đạt², Nguyễn Lê Vương², Trần An Khang², Phan Cao Hoàng²

¹Khoa Điện-điện tử, Trường Đại học Bách Khoa TP HCM (BKU), Đại học Quốc gia TP HCM, Tp. Hồ Chí Minh

²Khoa Đào tạo Chất Lượng Cao, Trường Đại học Sư Phạm Kỹ Thuật TP HCM (HCMUTE), Tp. Hồ Chí Minh

Tác giả liên hệ: Võ Minh Tài, E-mail: vmtai.sdh212@hcmut.edu.vn

Tóm tắt: Con lắc ngược quay (rotary inverted pendulum – RIP) là một hệ thống phi tuyến một vào-nhiều ra (single input-multi output –SIMO) có cấu trúc cơ khí đơn giản nhưng có độ phi tuyến cao. Đây là hệ thường được sử dụng trong các phòng thí nghiệm (PTN) điều khiển tự động (ĐKTD). Trong bài báo này, chúng tôi trình bày áp dụng mạng thần kinh suy diễn mờ (Adaptive Neuro Fuzzy Inference System - ANFIS) để mô tả bộ điều khiển (BĐK) linear quadratic regulator (LQR) trong cân bằng ổn định hệ thống tại điểm làm việc thẳng đứng hướng lên. Bên cạnh đó, một BĐK swing-up bằng phương pháp năng lượng cũng được trình bày trong nghiên cứu này. Các kết quả mô phỏng và thử nghiệm trong bài báo cho thấy sự khả thi của các phương pháp này trong việc cân bằng đối tượng RIP.

Từ khóa: ANFIS; con lắc ngược quay; điều khiển cân bằng; phương pháp LQR; swing-up

Abstract: RIP is a SIMO nonlinear system which has simple mechanical structure but high nonlinear level. This is a popular model in laboratories of control automation. In this paper, we present an ANFIS network which is developed from a former LQR controller in stabilizing system at equilibrium point– upright position. In addition, an energy swing-up controller is designed and tested in this paper. Simulation and experimental results in this paper prove the abilities of these methods in balancing RIP model.

Keywords: ANFIS; balancing control; LQR method; rotary inverted pendulum; swing-up

1. Giới thiệu

RIP là một mô hình kinh điển và phức tạp có độ phi tuyến cao trong lĩnh vực ĐKTD. Qua đó, nhiều nghiên cứu đã được thực hiện ở mô hình trên. Ở tài liệu [1], việc áp dụng các giải thuật điều khiển bền vững trượt thích nghi với nhiễu. Tương tự, ở nghiên cứu [2], giải thuật quasi kết hợp điều khiển để điều khiển đối tượng cân bằng và ổn định tại vị trí thẳng đứng hướng lên. Tuy nhiên, việc điều khiển trong các nghiên cứu giải thuật sâu như vậy chỉ được áp dụng thành công trên mô phỏng chứ chưa được kiểm nghiệm trên mô hình thực nghiệm.

Ở nghiên cứu [3], giải thuật điều khiển LQR đã được thử nghiệm thành công trên cả mô phỏng và thực nghiệm. Tuy nhiên, giải thuật LQR đã trở nên kinh điển và nhu cầu về một BDK thông minh dựa vào kinh nghiệm chuyên gia để có thể bắt chước một hoạt động điều khiển lặp đi lặp lại đối với RIP đang là xu hướng mới [4]. Ở nghiên cứu [5], mạng thần kinh (neuron network -NN) đã được thực hiện trên cả mô phỏng và thực nghiệm để cải thiện đáp ứng một BDK PID khi cân bằng hệ RIP. Tuy nhiên, mạng thần kinh trong nghiên cứu này thực chất chỉ là một neuron duy nhất với trọng số neuron là các thông số K_p , K_i , K_d . Cấu trúc neuron này chỉ giúp PID tự hiệu chỉnh thông số tốt hơn chứ bản thân neuron này điều khiển trực tiếp được hệ thống. Việc điều khiển toàn bộ hệ phi tuyến RIP thì chỉ một neuron chưa thực hiện nổi.

Như vậy, một khối neuron gồm nhiều neuron điều khiển thành công hệ RIP mô phỏng và thực nghiệm là một nghiên cứu mới. Đồng thời, nghiên cứu này tạo

nền tảng cho việc phát triển các giải thuật thông minh trên đối tượng này. Trong bài báo này, chúng tôi xây dựng một NN để bắt chước hoạt động của một BDK LQR để mô tả việc một giải thuật thông minh bắt chước hoạt động của một chuyên gia. Bên cạnh đó, giải thuật swing-up cũng được đề nghị để hệ RIP được tự di chuyển đến vị trí cân bằng thay vì cần sự tác động của người điều khiển.

2. Xây dựng BDK

2.1. BDK LQR

Xét một hệ thống phi tuyến có dạng sau

$$\dot{x} = f(x) + g(x)u \quad (1)$$

Trong đó, $x = [x_1 \ x_2 \ \dots \ x_n]^T$ là ma trận biến trạng thái của hệ thống; u là tín hiệu điều khiển của hệ thống.

Khi điều khiển hệ thống quanh một điểm làm việc có dạng

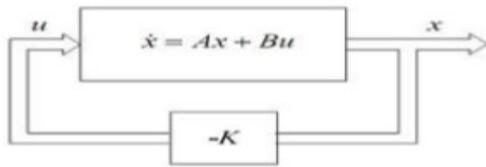
$$x_1 = x_2 = \dots = 0 \quad (x=x_0) \quad (2)$$

mà tại đó nếu $u=0$ thì hệ cân bằng, ta có thể xấp xỉ hệ thống ở (1) về dạng tuyến tính

$$\dot{x} = Ax + Bu \quad (3)$$

Trong đó, $A = \left. \frac{\partial f}{\partial x} \right|_{\substack{x=x_0 \\ u=0}}$; $B = \left. \frac{\partial g}{\partial x} \right|_{\substack{x=x_0 \\ u=0}}$

Khi hệ làm việc quanh vị trí cân bằng này, ta có thể xem như hệ xấp xỉ 1 hệ tuyến tính để việc thiết kế giải thuật điều khiển tuyến tính là khả thi. Cấu trúc điều khiển LQR tại điểm làm việc tĩnh của một hệ thống tuyến tính có dạng như ở (3) được thể hiện như ở Hình 1 sau



Hình 1. Cấu trúc BDK LQR

Tín hiệu điều khiển được chọn là

$$u = -Kx \quad (4)$$

Việc tính toán ma trận điều khiển K phải thông qua việc giải phương trình Ricatti rất phức tạp. Tuy nhiên, phần mềm Matlab đã hỗ trợ công cụ lệnh lqr() để việc tính toán ma trận K trở nên dễ dàng hơn. Lệnh tính toán ra K được thực hiện như sau:

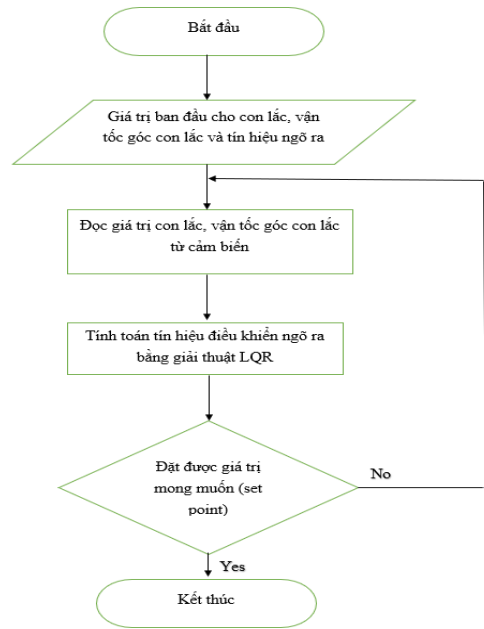
$$K = \text{lqr}(A, B, Q, R) \quad (5)$$

Trong đó, A, B được tính toán ở (3), Q và R là các ma trận trọng số được lựa chọn như sau

$$Q = \begin{bmatrix} Q_1 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_n \end{bmatrix}; R = R_1 \quad (5)$$

với Q_i ($i=1, 2, \dots, n$) và R_1 đều là các hằng số dương

Lưu đồ giải thuật LQR được thể hiện như ở Hình 2 sau:



Hình 2. Lưu đồ BDK LQR

2.2. BDK ANFIS [6]

Thuật toán lai huấn mạng ANFIS:

Chọn tập dữ liệu vào ra dùng để huấn luyện gồm K mẫu:

$$\begin{aligned} &(x(1), d(1)), (x(2), d(2)), \dots, \\ &(x(K), d(K)) \end{aligned} \quad (6)$$

Bước 1: Chọn tốc độ học $\eta > 0$, chọn sai số cực đại E_{\max} .

Bước 2: Khởi động:

Gán sai số $E=0$

Gán các giá trị đầu cho thông số phi tuyến $\theta_N(0)$

Bước 3: Ước lượng thông số tuyến tính dùng thuật toán bình phương tối thiểu:

$$\sum_{k=1}^K \left[\begin{matrix} d(k) - \\ -H(x(k), \theta_N(0))\theta_L \end{matrix} \right]^2 \rightarrow \min \quad (7)$$

$$\Rightarrow \theta_L = \left(\sum_{k=1}^K \begin{bmatrix} H^T(x(k), \theta_N(0)) \\ H(x(k), \theta_N(0)) \end{bmatrix}^{-1} \right) \left(\sum_{k=1}^K \begin{bmatrix} H^T(x(k), \theta_N(0)) \\ d(k) \end{bmatrix} \right) \quad (8)$$

Bước 4: Cập nhật trọng số phi tuyến dùng thuật toán suy giảm độ dốc: Với $k=1:K$

Tính sai số:

$$E(k) = \frac{1}{2} (d(k) - H(x(k), \theta_N) \theta_L)^2 \quad (9)$$

Cập nhật sai số:

$$\theta_N(k) = \theta_N(k-1) + \eta \frac{\partial E(k)}{\partial \theta_N} \quad (10)$$

Tính sai số tích lũy:

$$E = E + E(k) \quad (11)$$

Bước 5: Kết thúc một chu kỳ huấn luyện

Nếu $E < E_{max}$ thì kết thúc quá trình học.

Nếu $E \geq E_{max}$ thì gán $E=0$, và trở lại bước 3 bắt đầu một chu kỳ huấn luyện mới.

2.3. BDK LQR-based ANFIS

Phương pháp thiết kế của LQR-based ANFIS được giải thích được trình bày dựa theo thuật toán đã trình bày ở phần 2.1 và 2.2. BDK ANFIS sẽ bắt chước hoạt động của một BDK LQR đã hoạt động tốt với ma trận điều khiển.

$$K = [118.8867 \quad 12.8087 \quad -9.9470 \quad -12.3743] \quad (12)$$

Ma trận K này được tính theo (5) tương ứng với ma trận trọng số Q, R được tìm và tối ưu hóa từ giải thuật di truyền (genetic algorithm –GA) đối hệ

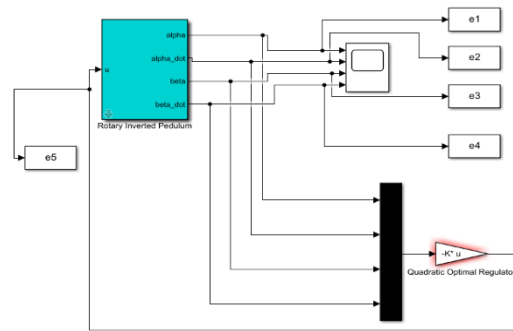
RIP có thông số ở Bảng 1. Khi đó, các ma trận Q và R có các giá trị sau

$$Q = \begin{bmatrix} 5.109 & 0 & 0 & 0 \\ 0 & 3.104 & 0 & 0 \\ 0 & 0 & 98.969 & 0 \\ 0 & 0 & 0 & 40.116 \end{bmatrix}; \quad (13)$$

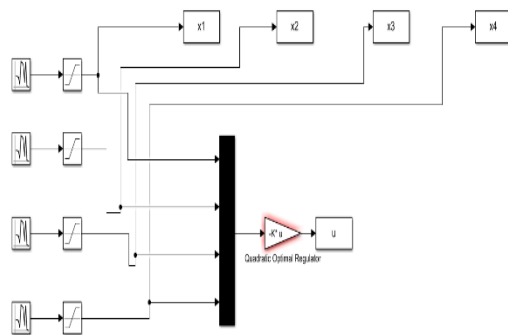
$$R = 0.1$$

2.3.1. Xây dựng tập dữ liệu

Việc xây dựng tập dữ liệu, cung cấp cho ANFIS, được thực hiện dưới dạng nhiều đầu vào và đầu ra. Để hình thành tập dữ liệu, BDK LQR tối ưu được thiết kế, tùy thuộc vào thông số hệ thống. Việc tối thiểu hóa hàm chi phí, được coi là một bài toán tối ưu hóa.



Hình 3. Chương trình mô phỏng hệ thống tạo tập dữ liệu vào ra



Hình 4. Chương trình thu thập dữ liệu trong Matlab/Simulink

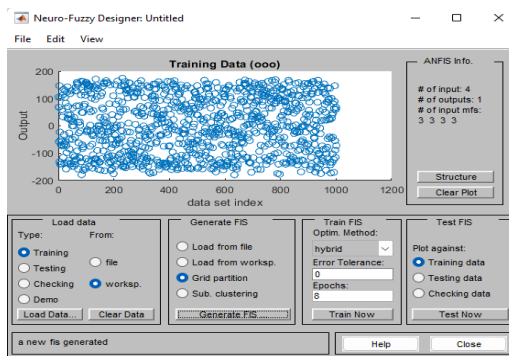
2.3.2. Xác định các thông số điều khiển của ANFIS

Các tham số điều khiển của ANFIS như số lượng và loại hàm liên thuộc, khả năng chịu lỗi (error tolerance), số lượng các epoch và phương pháp học được nêu như sau:

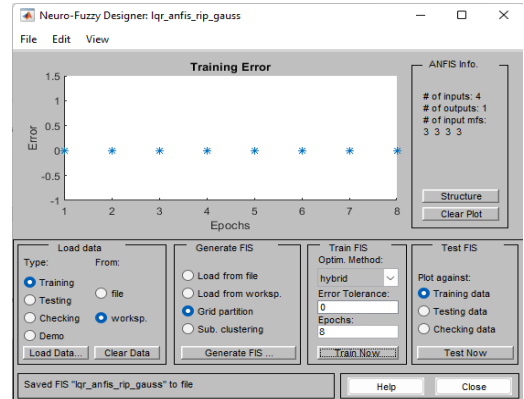
- Số hàm liên thuộc: 3
- Loại hàm liên thuộc: Gauss
- Error tolerance: 0
- Số epoch: 8
- Hình thức học: Hybrid

2.3.3. Huấn luyện ANFIS

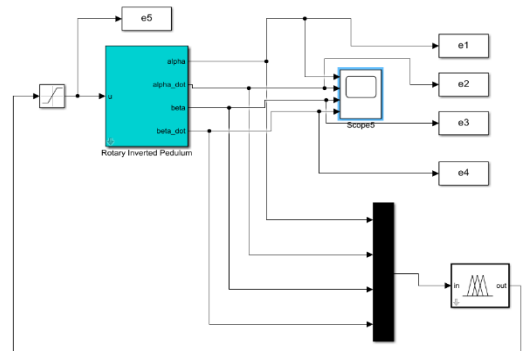
Một khối cấu trúc được ANFIS huấn luyện bằng cách kết hợp các tham số cho trong Phần 2.3.2 để phù hợp với tập dữ liệu thu thập trong Phần 2.3.1, trong đó giá trị u của tập dữ liệu được coi là đầu ra của suy luận mờ hệ thống trong khi các cột khác được coi là đầu vào của hệ thống suy luận mờ. Việc huấn luyện ANFIS có thể được thực hiện dễ dàng trong Matlab bằng cách sử dụng một lệnh `anfisedit`. Tập `.fis` được tạo ra hoạt động như một BDK phản hồi trạng thái cho hệ thống được thiết kế.



Hình 5. Giao diện tool ANFIS khi dữ liệu vào ra được cung cấp



Hình 6. Giá trị sai số sau khi huấn luyện 8 epoch



Hình 7. Chương trình mô phỏng hệ thống RIP với BDK LQR based ANFIS

2.4. Swing-up bằng phương pháp năng lượng

Bên cạnh giải thuật ổn định cân bằng – vốn chỉ hoạt động khi hệ thống đã ở lân cận vị trí cân bằng - thì cần có giải thuật swing-up để hệ RIP tự đưa con lắc đến vị trí thẳng đứng hướng lên. Phương pháp swing-up bằng năng lượng được áp dụng vào hệ thống để giải quyết yêu cầu điều khiển này. Theo tài liệu [8], tín hiệu điều khiển swing-up được áp dụng cho nghiên cứu này của chúng tôi như sau:

$$e = -e_{\max} \operatorname{sgn}((E - E_0)\dot{\alpha} \cos(\alpha)) \quad (14)$$

$$e = \begin{cases} e_{\max} & \text{if } (E - E_0)\dot{\alpha} \cos \alpha < 0 \\ -e_{\max} & \text{if } (E - E_0)\dot{\alpha} \cos \alpha > 0 \end{cases} \quad (15)$$

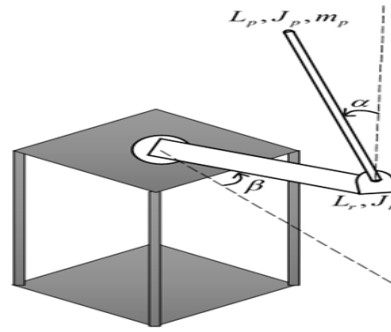
Trong đó, ta có $E_0 = mgl$,

$$E = ml^2 \dot{\alpha}^2 / 2 + mgl \cos(\alpha)$$

3. Mô hình RIP

3.1. Phương trình động lực học

Cấu trúc toán hệ RIP được thể hiện ở Hình 8



Hình 8. Cấu trúc hệ RIP

Theo [7], phương trình động lực học hệ thống dưới dạng ma trận có dạng sau:

$$D(\alpha, \beta) \cdot (\ddot{\alpha}, \ddot{\beta}) + C(\alpha, \dot{\alpha}, \beta, \dot{\beta}) \cdot (\dot{\alpha}, \dot{\beta}) + G(\alpha, \beta) = u \quad (16)$$

Trong đó, ta có

$$D(\alpha, \beta) = \begin{bmatrix} -\frac{1}{2} m_p L_p L_r \cos(\alpha) & m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos(\alpha)^2 + J_r + k_3 \\ \left(J_p + \frac{1}{4} m_p L_p^2 \right) & -\frac{1}{2} m_p L_p L_r \cos(\alpha) \end{bmatrix}$$

$$C(\alpha, \dot{\alpha}, \beta, \dot{\beta}) = \begin{bmatrix} \left(\frac{1}{2} m_p L_p L_r \sin(\alpha) \right) \dot{\alpha} & \left(\frac{1}{2} m_p L_p^2 \sin(\alpha) \cos(\alpha) \right) \dot{\alpha} + B_r + k_2 \\ B_p & -\frac{1}{4} m_p L_p^2 \cos(\alpha) \sin(\alpha) \dot{\beta}^2 \end{bmatrix}$$

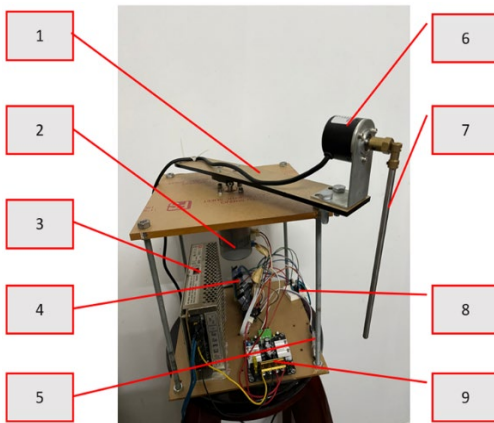
Bảng 1. Thông số hệ RIP (lấy từ mô hình thực nghiệm ở Hình 6)

α	Góc của thanh con lắc so với phương thẳng đứng	na	rad
β	Góc của thanh cánh tay so với vị trí gốc	na	rad
$\dot{\alpha}$	Vận tốc góc của thanh con lắc	na	rad / s
$\dot{\beta}$	Vận tốc góc của thanh cánh tay	na	rad / s
m_p	Khối lượng của thanh con lắc	0.15	kg
L_p	Chiều dài thanh con lắc	0.11	m
J_p	Moment quán tính của thanh con lắc	6.0500e-04	kgm ²
m_r	Khối lượng của thanh cánh tay	0.38	kg
L_r	Chiều dài thanh cánh tay	0.23	m
J_r	Moment quán tính thanh cánh tay	0.0067	kgm ²

g	Gia tốc trọng trường	9.80665	m / s^2
B_r	Hệ số ma sát cánh tay	0.1	n/a
B_p	Hệ số ma sát con lắc	6.6e-04	n/a
Thông số động cơ:			
$L_m = 0.250868; K_b = 0.09; R_m = 2.3; T_f = 0.0014; K_f = 0.1; J_m = 4e-05; C_m = 0.000048$			

3.2. Mô hình thực nghiệm

Mô hình thực nghiệm RIP được chúng tôi chế tạo có dạng như sau:



Hình 8. Mô hình phần cứng hệ RIP

Các thành phần trong Hình 6 bao gồm:

1. Thanh cánh tay
2. Servo Nisca NF5475 motor with encoder 200ppr

3. Nguồn xung

4. Kit STM32F407 Discovery

5. Kit CP2102 USB 2.0 to TTL UART

6. Encoder cho thanh con lắc

7. Thanh con lắc

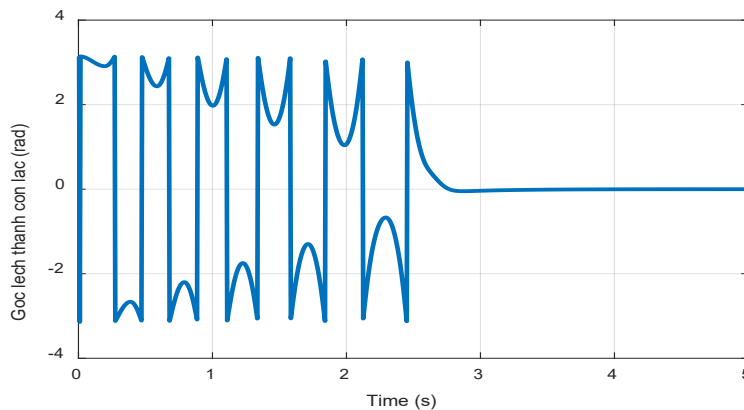
8. Điện trở kéo lên cho thanh cánh tay và thanh con lắc

9. H-BRIDGE-IR2184

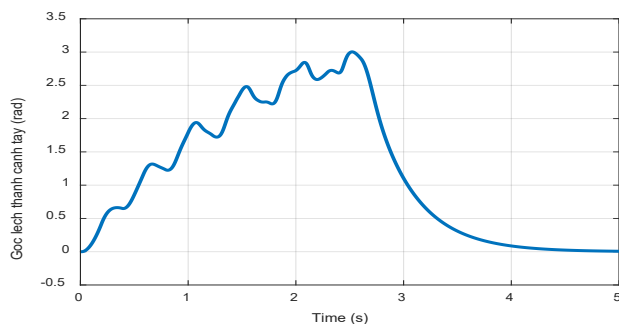
4. Kết quả điều khiển

Kết quả đáp ứng ngõ ra của hệ RIP được thể hiện từ Hình 7 tới Hình 11. Trong các hình này, giải thuật swing-up bằng phương pháp năng lượng và điều khiển bằng BDK ANFIS với các giá trị ban đầu cụ thể như sau:

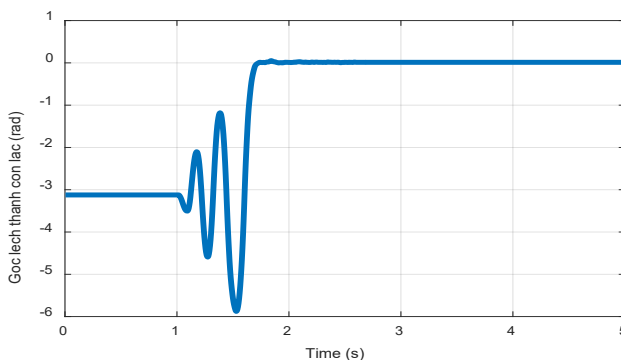
$$[\alpha(0), \dot{\alpha}(0)]^T = [\pi, 0]^T$$



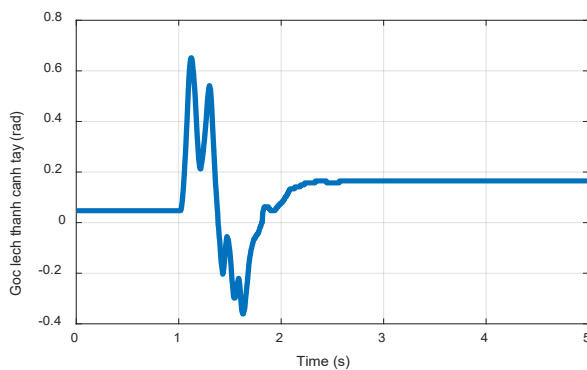
Hình 9. Kết quả mô phỏng góc lệch thanh con lắc (rad)



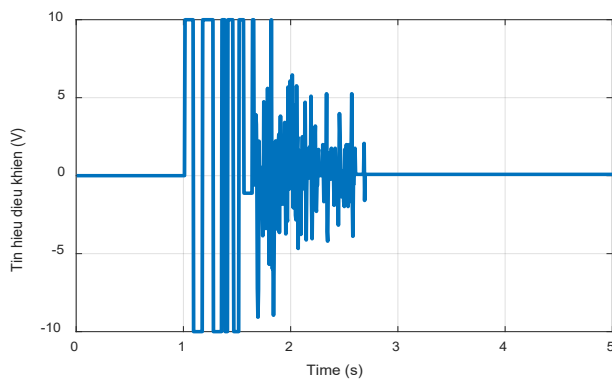
Hình 10. Kết quả mô phỏng đáp ứng góc lệch thanh cánh tay (rad)



Hình 11. Kết quả thực nghiệm đáp ứng góc lệch thanh con lắc (rad)



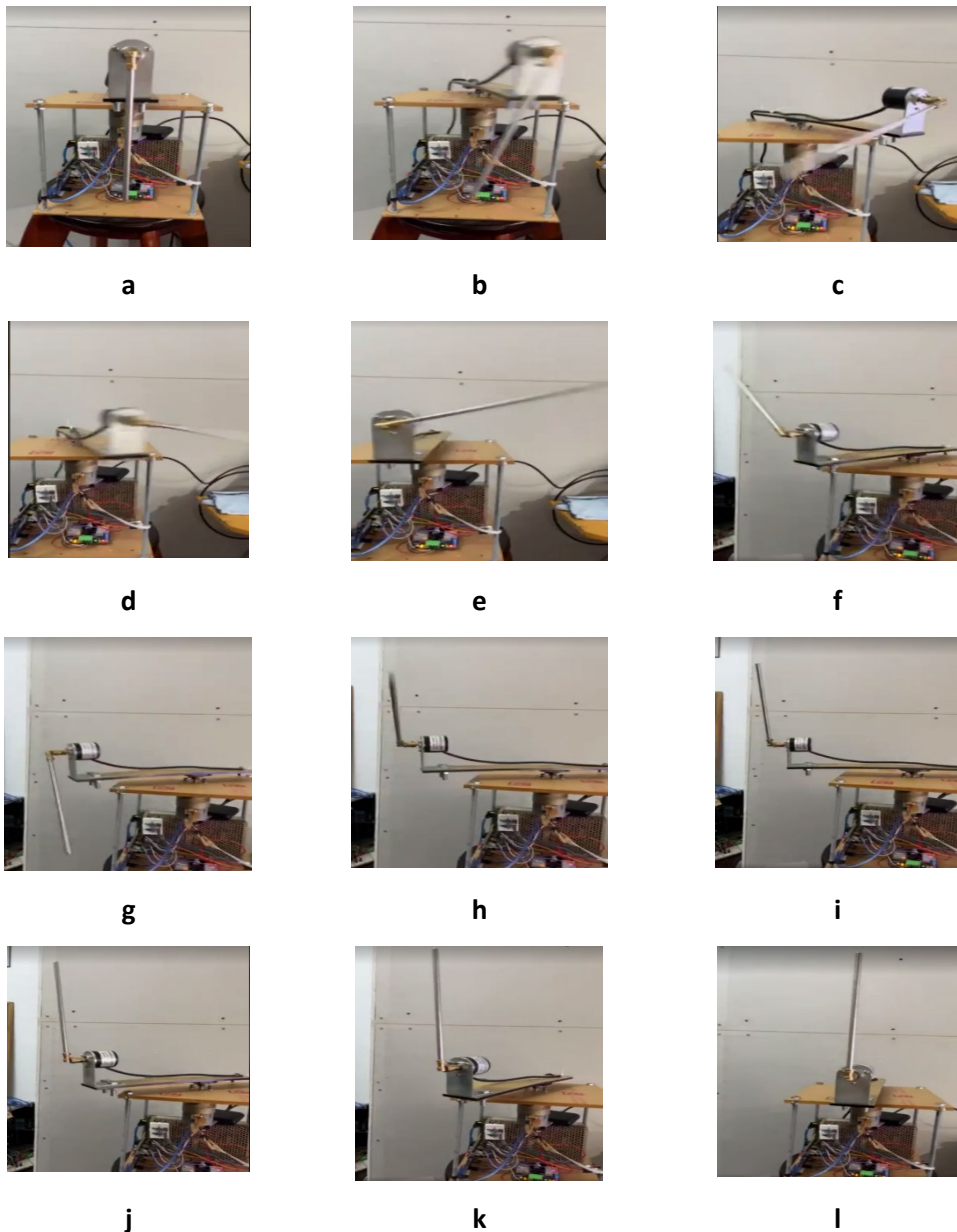
Hình 12. Kết quả thực nghiệm đáp ứng góc lệch thanh cánh tay (rad)



Hình 13. Kết quả thực nghiệm tín hiệu điện áp cấp cho động cơ (V)

Kết quả mô phỏng và thí nghiệm cho thấy khi swing-up thanh con lắc vào vùng bất ổn định $[-150;150]$ thì con lắc sẽ được giữ ổn định bằng BDK LQR-based ANFIS. Ở Hình 7, khi áp dụng phương pháp năng lượng để swing up hệ RIP và giữ ổn định bằng giải thuật LQR-based ANFIS, thì thời gian chuyển tiếp giữa điều khiển swing up và LQR-based ANFIS là gần 2 giây. Trong khoảng thời

gian đó, cánh tay có dịch chuyển nhưng sau 2 giây thì quay về vị trí ổn định 0 ban đầu (Hình 8). Kết quả thực nghiệm ở Hình 9 và Hình 10 cũng cho kết quả tương tự. Khi hệ cân bằng thì điện áp cấp cho động cơ cũng về giá trị 0 (không cần điều khiển nữa) (ở Hình 11). Quá trình hoạt động của hệ RIP thực nghiệm ở Hình 9, Hình 10 được thể hiện thông qua các hình ảnh được liệt kê ở Hình 12.



Hình 14. Quá trình hoạt động của mô hình thực nghiệm

5. Kết luận

Bộ BDK LQR-based ANFIS được đề xuất trong bài báo này đã điều khiển tốt hệ RIP. Như vậy, công cụ ANFIS mà Matlab cung cấp đã tạo ra một BDK thông minh có khả năng bắt chước hoạt động của một chuyên gia (ở đây là BDK LQR). BDK thông minh tạo ra từ công cụ này điều khiển tốt hệ thống ở cả thực nghiệm và mô phỏng. Bên cạnh đó, cũng thông qua mô phỏng và thực nghiệm, nhóm tác giả cũng đã kiểm chứng thành công giải thuật swing-up cân bằng năng lượng để kết hợp với BDK ANFIS để con lắc tự đưa đến vị trí cân bằng và tự cân bằng.

Lời cảm ơn

Nhóm nghiên cứu xin chân thành cảm ơn TS. Nguyễn Văn Đông Hải (PTN – C205, Khoa Điện-Điện tử, trường Đại học Sư phạm Kỹ thuật TPHCM) đã cung cấp nơi làm việc và hỗ trợ chúng tôi chỉnh sửa bài báo này.

Tài liệu tham khảo

- [1] F. F. M. El-Sousy et al, "Robust Adaptive Super-Twisting Sliding Mode Stability Control of Underactuated Rotational Inverted Pendulum With Experimental Validation," IEEE Access, Vol. 10, pp. 100857-100866, 2022.
- [2] Gupta, Neha, and Lillie Dewan, "Trajectory tracking and balancing control of rotary inverted pendulum system using quasi-sliding mode control," Mechatronic Systems and Control, Vol. 50, No. 1, 2022.

[3] Vo Anh Khoa et al, "Model and control algorithm construction for rotary inverted pendulum in laboratory," Journal of Technical Education Science, Vol. 49, pp. 32-40, 2018.

[4] Pedro Poce et al, "A review of intelligent control systems applied to the inverted-pendulum problem", American Journal of Engineering and Applied Sciences, Vol. 7, No. 2, pp. 161-207, 2014.

[5] Nguyễn Văn Đông Hải, Ngô Văn Thuyên, "Xây dựng BDK PID-neuron cho hệ con lắc ngược quay," Tạp chí Khoa học Giáo dục Kỹ thuật, Số 23, pp. 37-45, 2012.

[6] Ahmed Al-Hmouz, Jun Shen, Modelling and Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Mobile Learning, IEEE Transactions on Learning Technologies, Vol. 5, No. 3, pp. 226-237, 2012.

[7] Vo, M. T. et al, "Design of Input-Output Feedback Linearization Control for Rotary Inverted Pendulum System," Journal of Technical Education Science, Vol. 69, pp. 26-35, 2022.

[8] Hong , G. B. et al., "Comparison between two swing-up algorithms: partial feedback linear and energy based method," Journal of Technical Education Science, Vol. 55, pp. 1-6, 2019.

Ngày nhận bài: 14/1/2023

Ngày hoàn thành sửa bài: 24/3/2023

Ngày chấp nhận đăng: 27/3/2023