

XÁC ĐỊNH THÔNG SỐ BỘ ĐIỀU KHIỂN PID ĐIỀU KHIỂN VỊ TRÍ CON LẮC NGƯỢC BẰNG THUẬT TOÁN TỐI ƯU NGẪU NHIÊN

TRẦN MINH CHÍNH, NGUYỄN HOÀNG HIẾU

Khoa Công Nghệ Điện, Trường Đại học Công nghiệp Thành phố Hồ Chí Minh
Minhtranminhchinh@iuh.edu.vn

Tóm tắt. Hệ con lắc ngược và xe là một trong những hệ phi tuyến kinh điển thường được sử dụng để nghiên cứu trong điều khiển phi tuyến. Bộ điều khiển PID được sử dụng rộng rãi, cho kết quả tốt nhất trong điều kiện không biết mô hình toán học của hệ thống điều khiển. Tuy nhiên, để đạt được hiệu quả tốt nhất thì các thông số PID cần được tính toán và lựa chọn phù hợp cho từng hệ thống. Bài báo đề xuất phương pháp xác định thông số bộ điều khiển PID điều khiển con lắc ngược ở vị trí cân bằng thẳng đứng sử dụng thuật toán tối ưu Jaya. Bên cạnh đó, bài báo sẽ so sánh kết quả của thuật toán với các phương pháp điều khiển khác như LQR, tuyến tính hóa...

Từ khóa. Con lắc ngược và xe, Bộ điều khiển PID, điều khiển phi tuyến, thuật toán tối ưu ngẫu nhiên, thuật toán Jaya.

IDENTIFY PARAMETERS OF THE PID CONTROLLER FOR THE INVERTED PENDULUM SYSTEM USING METAHEURISTIC OPTIMIZATION ALGORITHM

Abstract. The inverted pendulum and cart system is one of the classic nonlinear systems commonly used in nonlinear control. The PID controller is widely used, giving the best results in the absence of the control system's mathematical model. However, to achieve the best performance, PID parameters need to be calculated and selected appropriately for each system. This paper proposes a method to determine parameters of the PID controller the inverted pendulum position using Jaya optimization algorithm. Besides, the paper will compare the results of the algorithm with other control methods such as LQR, linearization ...

Keywords (tiếng Anh). Inverted pendulum and cart system, PID controller, nonlinear control, metaheuristic optimization algorithm, Jaya algorithm.

1 GIỚI THIỆU

Hệ con lắc ngược đơn là một hệ thống phi tuyến điển hình một ngõ vào và nhiều ngõ ra (hệ SIMO), hệ có vô số điểm cân bằng không ổn định ở phía trên và vô số điểm cân bằng ổn định ở phía dưới. Tại mỗi vùng làm việc khác nhau hệ con lắc ngược đơn có đặc tính động học khác nhau, do đó việc điều khiển cân bằng của nó có nhiều khó khăn và thách thức. Do đặc tính phi tuyến tuyến cao nên hệ con lắc ngược đơn trở thành một mô hình kinh điển thường được dùng để kiểm chứng các giải thuật từ kinh điển đến hiện đại. Các bộ điều khiển cổ điển thường được sử dụng để điều khiển hệ con lắc ngược đơn như: PI, PD, PID, bộ điều khiển đặt cực, bộ điều khiển toàn phương tuyến tính LQR, ... cho đến các bộ điều khiển hiện đại: bộ điều khiển mờ, bộ điều khiển thích nghi hay sử dụng mạng thần kinh nhân tạo. Mỗi bộ điều khiển đã được nghiên cứu nêu trên đều có những ưu điểm và các hạn chế riêng. Trong bài báo [1] tác giả đã sử dụng bộ điều khiển đơn giản PD để điều khiển hệ con lắc ngược đơn. Bài báo [2] tác giả sử dụng hai bộ điều khiển PID để điều khiển vị trí xe và vị trí của con lắc. Cũng trong bài báo này, tác giả đã so sánh kết quả thực hiện với bộ điều khiển LQR. Tác giả Lal Bahadur Prasad trong [3] đã sử dụng bộ điều khiển PID kết hợp với bộ điều khiển LQR để nâng cao hiệu quả điều khiển. Trong [4] nhóm tác giả đề xuất hệ thống điều khiển hệ con lắc ngược dựa trên điều khiển PID kết hợp phương pháp tuyến tính hóa vào-ra. Các thông số của bộ điều khiển hay các loại bộ điều khiển sẽ ảnh hưởng rất lớn đến sự hoạt động cũng như sự ổn định của hệ thống. Việc lựa chọn các thông số của bộ điều khiển PID để hệ thống hoạt động tốt là một vấn đề nan giải. Một số phương pháp thông dụng để xác định thông số bộ điều khiển PID như việc thử sai dựa khá nhiều vào sự hiểu biết của người điều khiển vào hệ thống hay phương pháp giải tích chỉ áp dụng được khi biết được chính xác mô hình toán và thông số chính xác của hệ thống. Trong trường hợp đối tượng không thể mô tả được chính xác bằng mô hình toán, phương pháp dựa trên thuật toán tối ưu hàm mục tiêu điều

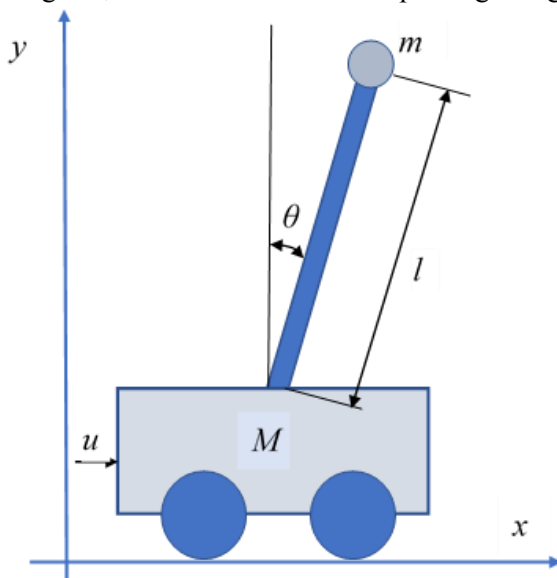
khiến theo các giá trị đo được ở ngõ vào và ngõ ra được đề xuất. Các thuật toán tối ưu hóa ngẫu nhiên thường được sử dụng để điều chỉnh các thông số của bộ điều khiển PID như: giải thuật di truyền (GA), thuật toán bầy đàn PSO, thuật toán đàn ong (BA). Trong các bài báo [7] – [10] các tác giả và cộng sự sử dụng giải thuật di truyền để tìm các thông số của bộ điều khiển PID. Trong khi đó [11], [12] sử dụng thuật toán đàn ong để xác định các tham số tốt hơn. Các tác giả trong [13], [14] sử dụng thuật toán tối ưu hóa bầy đàn (PSO). Bên cạnh các thuật toán cổ điển, các nhà nghiên cứu luôn nỗ lực áp dụng các thuật toán hiện đại để cải thiện chất lượng điều khiển. Trong các nghiên cứu [15], [16], các tác giả đã áp dụng lý thuyết mờ để điều khiển hệ con lắc ngược. Trong bài báo [17] tác giả xây dựng hai mạng neural nhân tạo để học hành vi lắc con lắc lên vị trí thẳng đứng và giữ cân bằng con lắc tại vị trí đó.

Mỗi bộ điều khiển hay mỗi phương pháp xây dựng các bộ điều khiển đều có các ưu và nhược điểm riêng. Thuật toán di truyền (GA) có khả năng tìm được các giá trị tối ưu toàn cục tuy nhiên thuật toán khá phức tạp, cần có nhiều kinh nghiệm trong việc lựa chọn các thông số cho GA như tỷ lệ đột biến, tỷ lệ lai ghép, tốc độ hội tụ chậm. Không giống như GA, thuật toán tối ưu bầy đàn (PSO) đơn giản hơn nhiều. PSO không có đột biến và lặp lại các phép tính, việc tính toán trong PSO cũng đơn giản hơn và tốc độ hội tụ nhanh hơn. Tuy nhiên, PSO vẫn tồn tại một vài nhược điểm: dễ rơi vào cực trị cục bộ, trong các bài toán không hệ tọa độ việc điều chỉnh tốc độ và hướng thường kém chính xác [18]. Trong bài báo này, tác giả đề xuất thuật toán tối ưu mới với khả năng tính toán mạnh mẽ hơn, đơn giản hơn, dễ hiểu hơn nhiều so với GA. Thuật toán tối ưu Jaya sẽ được giới thiệu trong phần 4 của bài báo.

Bố cục bài báo gồm có 6 phần: phần 2 giới thiệu mô hình động học của hệ con lắc ngược, phần 3 giới thiệu bộ điều khiển PID, phần 4 giới thiệu thuật toán tối ưu ngẫu nhiên Jaya, phần 5 trình bày kết quả xác định các thông số bộ điều khiển PID bằng thuật toán Jaya và so sánh kết quả với bộ điều khiển tối ưu LQR và bộ điều khiển tuyến tính hóa bằng phương pháp đặt cực, phần 6 đưa ra các kết luận.

2 MÔ HÌNH TOÁN CON LẮC NGƯỢC ĐƠN

Con lắc ngược đơn được mô tả như hình 1. Hệ thống gồm có xe có thể di chuyển sang trái và phải trên đường ray dưới tác động của lực u được cung cấp bởi động cơ gắn trên xe. Trên xe gắn con lắc là một thanh cứng có thể chuyển động quay quanh một trục cố định. Vị trí xe chính là độ dịch chuyển của xe theo phương ngang x , còn vị trí con lắc chính là góc lệch θ của con lắc so với phương thẳng đứng.



Hình 1: Mô hình hệ con lắc ngược đơn

Việc xây dựng mô hình toán học cho hệ con lắc ngược đơn có thể dựa trên các định luật Newton [2] hay dựa trên các phương trình Euler-Lagrange [3], [5]. Trong [6], nhóm tác giả đề xuất thuật toán phương pháp nội suy thực (RIM) để xác định mô hình con lắc ngược. Dù xây dựng theo cách tiếp cận nào thì cuối cùng mô hình động học của hệ con lắc ngược đơn cũng được mô tả bởi các phương trình (1) và (2):

$$\ddot{x} = \frac{u + ml(\sin \theta)\dot{\theta}^2 - mg \cos \theta \sin \theta}{M + m - m \cos^2 \theta} \quad (1)$$

$$\ddot{\theta} = \frac{u \cos \theta - (M + m)g(\sin \theta) + ml(\cos \theta \sin \theta)\dot{\theta}}{ml \cos^2 \theta - (M + m)l} \quad (2)$$

Trong đó các thông số vật lý của hệ con lắc ngược được cho trong bảng 1.

Bảng 1: Thông số vật lý của hệ con lắc ngược

Ký hiệu	Ý nghĩa vật lý	Giá trị
M	Trọng lượng của xe	1 kg
m	Trọng lượng của con lắc	0.1 kg
l	Chiều dài của con lắc	1 m
g	Gia tốc trọng trường	9.8 m/s^2
θ	Góc lệch của con lắc theo phương thẳng đứng	rad
x	Vị trí của xe	m
u	Lực tác động lên xe	N

Bằng cách đặt biến trạng thái $x_1 = \theta, x_2 = \dot{\theta}, x_3 = x, x_4 = \dot{x}$, phương trình động học của hệ con lắc ngược có thể được biểu diễn dưới dạng hệ phương trình trạng thái như sau:

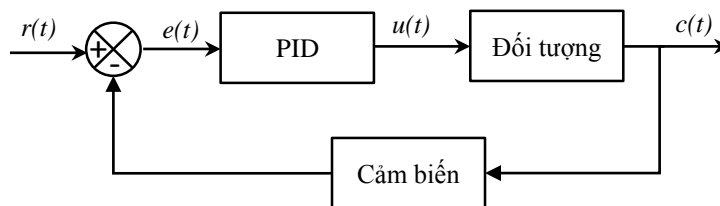
$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{u \cos x_1 - (M + m)g(\sin x_1) + ml(\cos x_1 \sin x_1)x_2}{ml \cos^2 x_1 - (M + m)l} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{u + ml(\sin x_1)x_2^2 - mg \cos x_1 \sin x_1}{M + m - m \cos^2 x_1} \end{cases} \quad (3)$$

Hệ phương trình (1), (2) hay hệ phương trình trạng thái (3) mô tả động học hệ con lắc ngược đơn và được dùng để mô phỏng kiểm chứng các thuật toán điều khiển. Bằng phương pháp tuyến tính hóa quanh điểm làm việc tĩnh tại vị trí con lắc thẳng đứng (góc lệch θ nhỏ hơn 10°), ta có phương trình tuyến tính hóa của con lắc ngược đơn như sau:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{M + m}{Ml}g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{m}{M}g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix} u \quad (4)$$

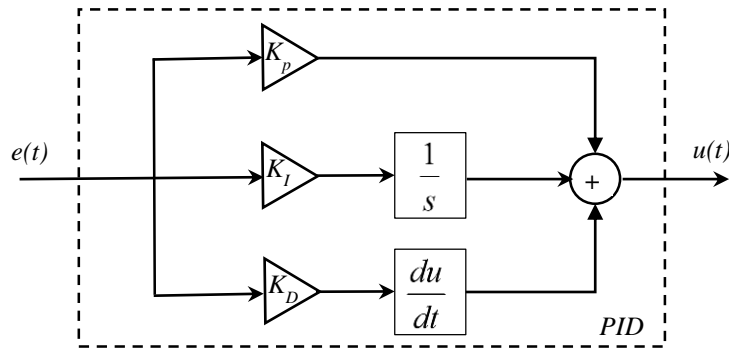
3 BỘ ĐIỀU KHIỂN PID

Bộ điều khiển PID được sử dụng rất rộng rãi trong công nghiệp để điều khiển nhiều đối tượng khác nhau mà không cần quan tâm đến mô hình toán học của hệ thống. Bộ điều khiển PID hoạt động dựa trên cơ chế cân bằng “sai số” $e(t)$ giữa tín hiệu phản hồi và tín hiệu đặt mong muốn. Hình 2 trình bày sơ đồ khối của một hệ thống điều khiển sử dụng bộ điều khiển PID.



Hình 2: Sơ đồ hệ thống điều khiển sử dụng PID

Bộ điều khiển PID bao gồm ba khâu được đặc trưng bởi ba hệ số riêng biệt (Hình 3).



Hình 3: Sơ đồ bộ điều khiển PID

Giải thuật tính toán bộ điều khiển PID được cho bởi công thức (5):

$$u(t) = K_p e(t) + K_I \int e(t) + K_D \frac{de(t)}{dt} \quad (5)$$

Trong đó các hệ số K_p, K_I, K_D lần lượt là các thông số đặc trưng của các khâu tỷ lệ, tích phân, vi phân của bộ điều khiển PID. Giá trị K_p tác động lên sai số hiện tại, giá trị K_I tác động lên sai số trong quá khứ, trong khi đó giá trị K_D ảnh hưởng lên tốc độ thay đổi của sai số. Ảnh hưởng của ba thông số K_p, K_I, K_D đến chất lượng của hệ thống được chỉ ra trong [3] và được tổng hợp trong bảng 2.

Bảng 2: Ảnh hưởng của các thông số bộ điều khiển PID đến hệ thống

Thông số	Thời gian lên t_s	Thời gian quá độ t_{qd}	Độ vọt lố	Sai số xác lập
K_p	Giảm	Thay đổi nhẹ	Tăng	Giảm
K_I	Giảm	Tăng	Tăng	Khử
K_D	Thay đổi nhẹ	Giảm	Giảm	Không thay đổi

Mặc dù bộ điều khiển PID có thể áp dụng rộng rãi cho nhiều vấn đề điều khiển ngay cả khi chúng ta không nắm được mô hình toán mô tả hệ thống. Tuy nhiên, khó khăn cơ bản của bộ điều khiển PID là nó là một hệ thống điều khiển phản hồi với các thông số không đổi do đó nó sẽ nhạy cảm với nhiễu và thường không đáp ứng được bài toán tối ưu. Để hệ thống hoạt động ổn định ở mức chấp nhận được, việc xác định các hệ số K_p, K_I, K_D của bộ điều khiển PID là cần thiết và không phải là một bài toán dễ dàng. Phần tiếp theo giới thiệu thuật toán tối ưu hóa ngẫu nhiên Jaya và việc áp dụng nó trong bài toán xác định các thông số cho bộ điều khiển PID để điều khiển hệ con lắc ngược đơn và xe.

4 THUẬT TOÁN TỐI ƯU HÓA NGẪU NHIÊN JAYA

Thuật toán Jaya là một giải thuật mới hiệu quả trong việc giải bài toán tối ưu có ràng buộc hoặc không có ràng buộc. Thuật toán này là một trong những thuật toán tối ưu hóa ngẫu nhiên được tác giả V. Rao và các cộng sự công bố vào năm 2016 [19]. So với các thuật toán tối ưu ngẫu nhiên đã biết như GA, DE, PSO, BA thì Jaya dễ hiểu hơn nhiều do việc tính toán đơn giản. Thuật toán chỉ yêu cầu hai thông số cơ bản là quy mô dân số (population size) và số các thế hệ (number of generations), ngoài ra không cần bất cứ thông số riêng nào. Hoạt động của thuật toán Jaya có thể được tóm tắt lại dưới đây.

- 1) Giống như các thuật toán tối ưu hóa ngẫu nhiên khác, Jaya cũng được bắt đầu bằng việc khởi tạo quy mô dân số ngẫu nhiên gồm NP cá thể, mỗi cá thể là một vector chứa D biến thiết kế $\mathbf{X} = (x_1, x_2, \dots, x_D)$. Tại mỗi bước lặp i , xác định giá trị tốt nhất của hàm mục tiêu (f_{best}) tương ứng với cá thể tốt nhất (x_{best}) và giá trị xấu nhất (f_{worst}) tương ứng với cá thể xấu nhất (x_{worst});
- 2) Các cá thể mới được cập nhật dựa trên các cá thể tốt nhất và xấu nhất theo công thức sau:

$$x'_{j,k} = x_{j,k} + r_{1,j}(x_{j,best} - |x_{j,k}|) - r_{2,j}(x_{j,worst} - |x_{j,k}|) \quad (6)$$

Trong đó $x_{j,best}, x_{j,worst}$ tương ứng là giá trị của biến tốt nhất và xấu nhất; $x'_{j,k}$ là giá trị cập nhật của $x_{j,k}$ và $r_{1,j}, r_{2,j}$ là các số ngẫu nhiên trong đoạn $[0,1]$. Biểu thức $r_{1,j}(x_{j,best} - |x_{j,k}|)$ giúp tiến gần hơn đến giải pháp tốt nhất, còn biểu thức $-r_{2,j}(x_{j,worst} - |x_{j,k}|)$ để loại bỏ giải pháp xấu nhất.

3) Giá trị cập nhật $x'_{j,k}$ sẽ được lựa chọn nếu giá trị hàm mục tiêu tại đó cho giá trị tốt hơn.

Hoạt động của thuật toán Jaya được giải thích cụ thể từng bước trong bài báo [19]. Thuật toán có thể biểu diễn dưới dạng mã giả như sau:

Thuật toán Jaya

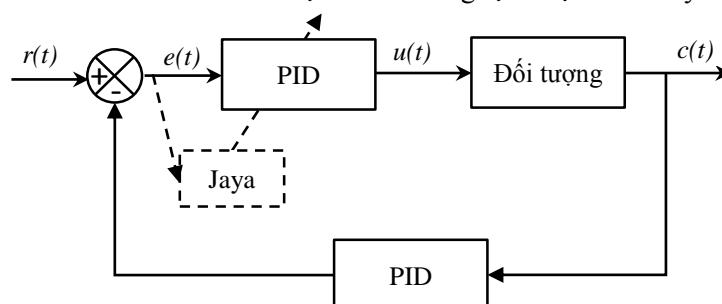
- 1: Khởi tạo quần thể
- 2: Tính giá trị hàm mục tiêu của mỗi cá thể trong quần thể
- 3: **While** (Delta > Epsilon and Iter < MaxIter) **do**
- 4: **for** $k=1$ **to** NP **do**
- 5: Xác định $x_{j,best}, x_{j,worst}$
- 6: **for** $j=1$ **to** D **do**
- 7: $x'_{j,k} = x_{j,k} + r_{1,j}(x_{j,best} - |x_{j,k}|) - r_{2,j}(x_{j,worst} - |x_{j,k}|)$
- 8: **end for**
- 9: Tính giá trị hàm mục tiêu tại $x'_{j,k}$
- 10: **if** (giá trị hàm mục tiêu của $x'_{j,k}$ tốt hơn giá trị hàm mục tiêu của $x_{j,k}$)
- 11: Cập nhật $x_{j,k} = x'_{j,k}$
- 12: **else** giữ nguyên giá trị cũ $x_{j,k} = x_{j,k}$
- 13: **end if**
- 14: **end for**
- 15: Tính Delta
- 16: Iter = Iter + 1
- 17: **End While**

Thuật toán Jaya đã được Rao và các cộng sự trong [19] kiểm chứng trên 24 hàm đối sánh với các đặc điểm khác nhau như: tuyến tính, phi tuyến, toàn phương, đơn trị, đa trị... Kết quả thực hiện của thuật toán Jaya được so sánh với kết quả của các thuật toán khác như: GA, PSO, DE, ABC, BBO, TLBO ... Thuật toán Jaya không những cho kết quả khả quan mà còn có thể dễ dàng áp dụng cho các bài toán có ràng buộc.

5 KẾT QUẢ MÔ PHỎNG VÀ SO SÁNH

5.1 Thiết kế bộ điều khiển PID

Sơ đồ thiết kế bộ điều khiển PID điều khiển vị trí con lắc ngược được trình bày như trong Hình 4.

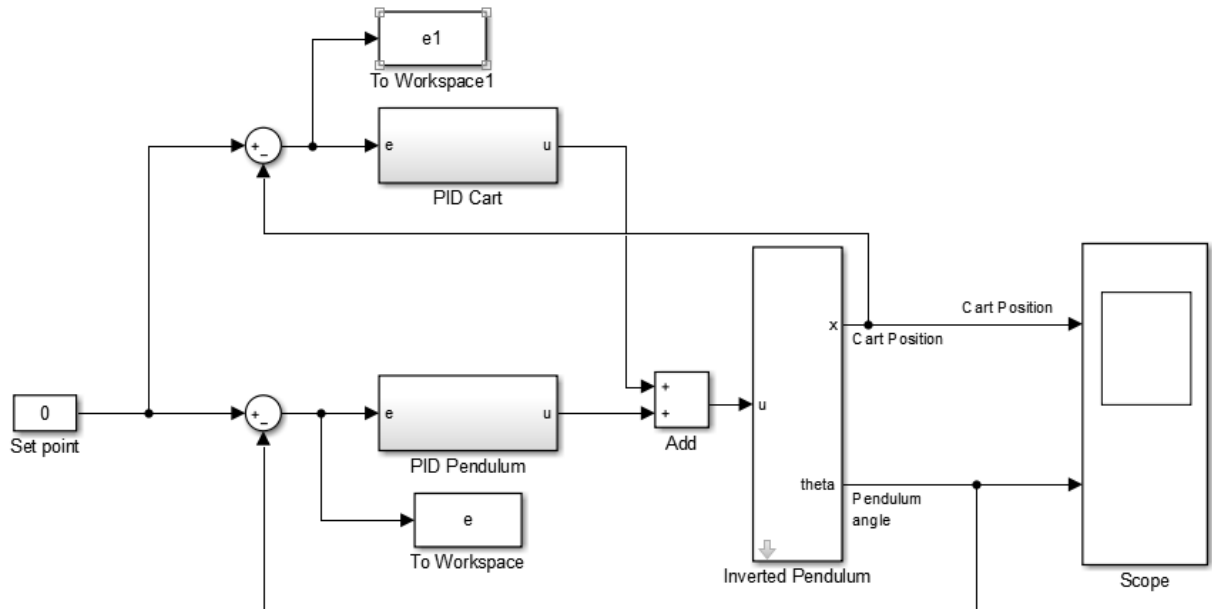


Hình 4: Sơ đồ thiết kế bộ điều khiển PID sử dụng thuật toán Jaya

Thuật toán tối ưu hóa ngẫu nhiên Jaya để xác các thông số K_p, K_I, K_D của bộ điều khiển PID được lập trình dưới dạng m-file trong phần mềm Matlab để tối thiểu hóa hàm mục tiêu. Hàm mục tiêu được chọn

để cung cấp cho thuật toán Jaya là hàm bình phương của sai số $e(t)$ giữa tín hiệu đặt $r(t)$ và tín hiệu ra $c(t)$.

Sơ đồ mô phỏng điều khiển hệ con lắc ngược đơn được thực hiện trên phần mềm Matlab (Hình 5.). Hệ thống sử dụng 2 bộ điều khiển PID: 1) Bộ điều khiển PID Cart dùng để điều khiển vị trí x của xe tại điểm đặt cho trước; 2) Bộ điều khiển PID Pendulum dùng để điều khiển con lắc ở vị trí thẳng đứng. Luật điều khiển u để điều khiển hệ thống được thỏa hiệp bởi hai luật điều khiển cho bởi hai bộ điều khiển nêu trên.



Hình 5: Sơ đồ mô phỏng điều khiển tổng quát.

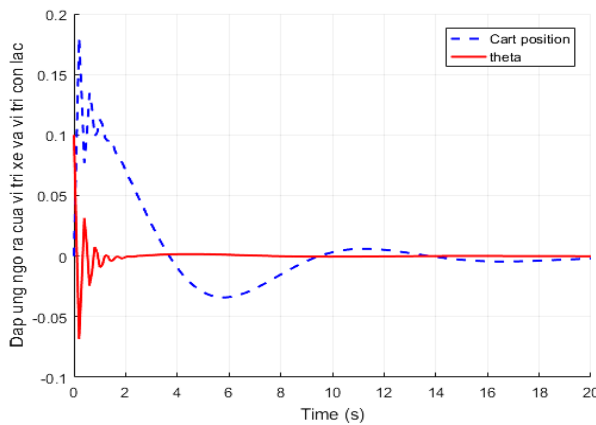
Tiến hành mô phỏng quá trình xác định thông số bộ điều khiển PID điều khiển hệ con lắc ngược và xe bằng thuật toán tối ưu Jaya với các thông số:

- Số lần lặp: RUNS = 10;
- Quy mô dân số (Population size) pop_size = 50;
- Số lượng các thế hệ (generations): gen = 100;

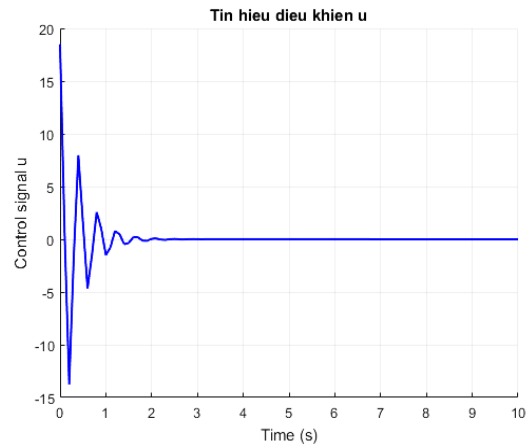
Kết quả thực hiện mô phỏng xác định thông số bộ điều khiển PID bằng thuật toán Jaya sau 10 lần lặp được cho bởi bảng 3. Các hệ số tốt nhất của bộ điều khiển PID và giá trị hàm mục tiêu được lưu trữ sau mỗi bước chạy (Run). Kết quả bước chạy thứ 7 cho giá trị hàm mục tiêu nhỏ nhất được chọn là kết quả tối ưu của thuật toán Jaya. Đáp ứng ngõ ra của vị trí xe và vị trí con lắc được thể hiện trên hình 6.

Bảng 3: Kết quả thực hiện thuật toán Jaya

Run	PID Cart			PID Pendulum			J
	K_p	K_D	K_I	K_p	K_D	K_I	
1	-187.7395	-12.0978	-6.6943	-6.6366	-9.2948	-0.029	2.6881
2	-190.6398	-10.6086	-2.5199	-5.8232	-8.7579	-0.0304	2.6865
3	-189.8906	-11.3273	14.5644	-7.6063	-8.2732	0.1746	2.5421
4	-188.5954	-11.7355	-23.5756	-5.9936	-8.4882	-0.0461	2.6710
5	-193.1875	-11.9365	-10.5004	-7.43	-9.1073	-0.1403	2.5769
6	-190.1543	-11.3655	-23.1158	-5.9292	-7.3003	-0.0839	2.6330
7	-187.7367	-11.3650	-0.8693	-6.6723	-8.3425	-0.1261	2.4989
8	-202.8649	-11.0641	20.8756	-7.0645	-8.8181	0.2177	2.5116
9	-190.9676	-11.2903	12.0828	-6.0168	-6.9172	0.7053	2.6871
10	-187.7395	-12.0978	-6.6943	-6.8435	-8.2908	-0.0302	2.6881



Hình 6: Đáp ứng ngõ ra của vị trí xe x và vị trí con lắc.



Hình 7: Tín hiệu điều khiển u.

Bộ điều khiển PID thiết kế dựa trên thuật toán tối ưu Jaya thỏa mãn chất lượng điều khiển, con lắc ổn định tại vị trí cân bằng thẳng đứng, vị trí xe bám theo vị trí đặt trước, độ vọt lố nhỏ, dao động nhanh tắt. Thời gian xác lập của con lắc khoảng 1.5s trong khi thời gian xác lập để vị trí xe ổn định tại vị trí cho trước khoảng 8s.

5.2 Điều khiển hồi tiếp trạng thái và điều khiển tối ưu LQR

Đối tượng tuyến tính mô tả bởi phương trình trạng thái:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ c(t) = Cx(t) \end{cases} \quad (7)$$

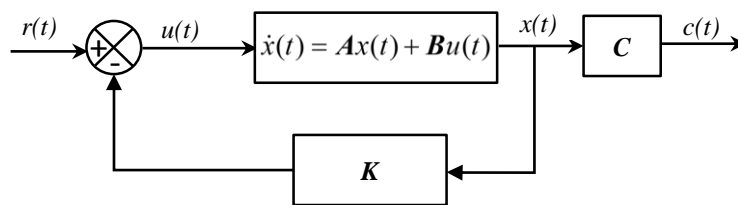
Hệ con lắc ngược và xe với các thông số trong bảng 1 được tuyến tính hóa quanh điểm làm việc tĩnh tại vị trí con lắc thẳng đứng (góc lệch θ nhỏ hơn 10°), ta có:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 10.78 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.98 & 0 & 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Hệ thống điều khiển hồi tiếp trạng thái được mô tả trong **Hình 8**, trong đó tín hiệu điều khiển được xác định bởi biểu thức:

$$u(t) = r(t) - Kx(t) \quad (8)$$



Hình 8: Hệ thống điều khiển hồi tiếp trạng thái.

Trong đó K là vecto hồi tiếp trạng thái cần được thiết kế sao cho hệ thống thỏa mãn yêu cầu chất lượng mong muốn. Phương pháp chọn K để hệ thống có các cực ở vị trí mong muốn được gọi là phương pháp phân bố cực.

Để hệ thống kín có các cực ở vị trí mong muốn $p = [-1 \ -2 \ -3 \ -4]$ cần chọn vecto $K = [-48.2290 \ -15.1020 \ -2.4490 \ -5.1020]$.

Ngoài ra \mathbf{K} còn được xác định bằng phương pháp điều khiển tối ưu LQR. Mục tiêu chính của phương pháp LQR là tìm ra tín hiệu điều khiển tốt nhất đáp ứng các yêu cầu ràng buộc bằng cách tối đa hóa hay tối thiểu hóa hàm chỉ tiêu chất lượng đặt ra. Hàm chỉ tiêu chất lượng có dạng:

$$J = \int_{t_0}^{t_f} (x^T \mathbf{Q} x + u^T \mathbf{R} u) dt \quad (9)$$

trong đó \mathbf{Q} là ma trận xác định dương hoặc bán xác định dương và \mathbf{R} là ma trận hằng số đối xứng xác định dương. Lúc này vecto \mathbf{K} được xác định bằng:

$$\mathbf{Q} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (10)$$

\mathbf{P} là ma trận hằng số đối xứng xác định dương và được xác định bằng việc giải phương trình đại số Riccati (ARE):

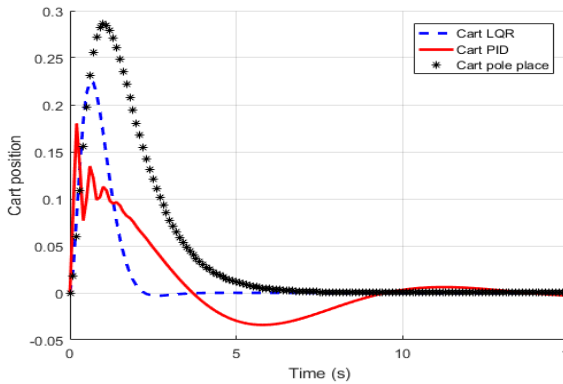
$$\mathbf{P} \mathbf{A} + \mathbf{A}^T \mathbf{P} + \mathbf{Q} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} = 0 \quad (11)$$

Thiết kế bộ điều khiển tối ưu LQR cho hệ con lắc ngược đơn bằng cách chọn $\mathbf{R} = 1$ và ma trận \mathbf{Q} như sau:

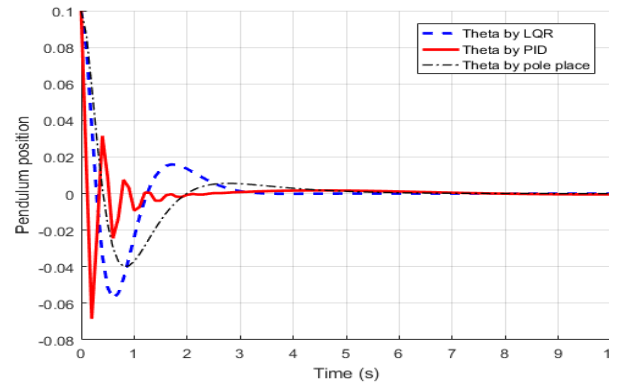
$$\mathbf{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

thì vecto cần thiết kế là $\mathbf{K} = [-71.7655 \quad -22.4050 \quad -10.0000 \quad -11.2009]$.

Kết quả mô phỏng điều khiển hệ thống con lắc ngược và xe bằng phương pháp phân bố cực và điều khiển tối ưu LQR được so sánh với phương pháp điều khiển bằng PID xác định bằng thuật toán tối ưu Jaya được trình bày trên các hình 9 và 10.



Hình 9: Đáp ứng vị trí xe.



Hình 10: Đáp ứng vị trí con lắc.

Kết quả so sánh đáp ứng của hệ thống con lắc ngược và xe tương ứng với các phương pháp điều khiển đã nêu trên được thể hiện trong bảng 4.

Bảng 4: Chất lượng điều khiển của các phương pháp điều khiển

	POT_x	POT_{theta}	t_{qd_x}	$t_{qd_{theta}}$
Phân bố cực	0.2858	0.0399	3.5	1.3
LQR	0.2249	0.0573	1.5	1
PID-Jaya	0.1802	0.0685	6	0.4

Thông qua kết quả trong bảng 4, đồ thị trong hình 9 và hình 10 dễ thấy rằng các phương pháp điều khiển được áp dụng trên đều thỏa mãn yêu cầu điều khiển vị trí con lắc thẳng đứng và vị trí xe tại điểm xác định trước. Bộ điều khiển PID sử dụng thuật toán Jaya cho kết quả tốt ở các thông số: vọt lố vị trí x của xe (POT_x) và thời gian đáp ứng của con lắc ($t_{qd_{theta}}$), tuy nhiên độ dao động của con lắc vẫn còn tồn tại.

6 KẾT LUẬN

Trong trường hợp biết được mô hình toán và thông số vật lý của hệ thống, ta có thể thiết kế bộ điều khiển tối ưu LQR hay hồi tiếp trạng thái để cho chất lượng đáp ứng tốt. Tuy nhiên, việc xác định mô hình toán

cho các hệ thống thực tế cũng như thông số vật lý của hệ không phải dễ dàng. Bài báo đã đề xuất thuật toán tối ưu hóa ngẫu nhiên Jaya để xác định thông số của bộ điều khiển PID, trong đó chỉ cần quan tâm đến tín hiệu ngõ ra và sai số mà không cần thiết phải xác định được mô hình toán học của hệ thống. Bộ điều khiển PID đề xuất đã thỏa mãn chất lượng điều khiển: con lắc cân bằng ổn định ở vị trí thẳng đứng và xe cũng cân bằng ở vị trí đặt cho trước.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Fu Wei, Giang Liangzhong, Yang Jin Bian Qingqing, Inverted pendulum control system based on GA optimization, *IEEE Comput Soc* (2007).
- [2] A. Alpher and J. P. N. Fotheringham-Smythe, Frobnication revisited, *Journal of Foo*, vol. 13, no. 1, pp. 234-778, 2003.
- [3] L. B. Prasad, B. Tyagi and H. O. Gupta, "Modelling and Simulation for Optimal Control of Nonlinear Inverted Pendulum Dynamical System Using PID Controller and LQR," *2012 Sixth Asia Modelling Symposium*, Bali, 2012, pp. 138-143, doi: 10.1109/AMS.2012.21.
- [4] Huỳnh Minh Ngọc, Trần Minh Chính. Điều khiển hệ con lắc ngược dùng PID kết hợp tuyến tính vào – ra. Vol 45 No 03 (2020): *Journal Of Science And Technology - Volume 45A (03-2020)*.
- [5] Murari, Krishna & Ahmad, Faraz & Kumar, Pushpendra. (2019). MODELING AND ANALYSIS OF AN INVERTED PENDULUM.
- [6] Tin, Phu & Trang, Tran & Nguyen, Dũng & Minh, Tran. (2019). Using real interpolation method for adaptive identification of nonlinear inverted pendulum system. *International Journal of Electrical and Computer Engineering*. 9. 1078-1089. 10.11591/ijece.v9i2.pp.1078-1089.
- [7] F. Wei, G. Liangzhong, Y. Jin and B. Qingqing, "Inverted Pendulum Control System Based on GA Optimization," *Workshop on Intelligent Information Technology Application (IITA 2007)*, Zhang Jiajie, 2007, pp. 347-350, doi: 10.1109/IITA.2007.85.
- [8] M.R. Dastranj, Design of optimal PID controller using genetic algorithm, *Aust J Basic Appl Sci*, 5 (10) (2011), pp. 996-1001.
- [9] M. Moghaddas, "Design of Optimal PID Controller for Inverted Pendulum Using Genetic Algorithm," *Int. J. Innov. Manag. Technol.*, vol. 3, no. 4, pp. 440–442, 2012, doi: 10.7763/ijimt.2012.v3.271.
- [10] L. A. Yusuf and N. Magaji, "GA-PID controller for position control of inverted pendulum," *IEEE Int. Conf. Adapt. Sci. Technol. ICASST*, vol. 2015-Janua, 2015, doi: 10.1109/ICASTECH.2014.7068099.
- [11] M.A. Sen, M. Kalyoncu, Optimisation of a PID controller for an inverted pendulum using the bees algorithm, *Appl Mech Mater*, 789–790 (2015), pp. 1039-1044.
- [12] D.T. Pham, E. Koc, M. Kalyoncu, M. Tinkır, Hierarchical PID controller design for a flexible link robot manipulator using the bees algorithm, *Proceedings of 6th international symposium on intelligent manufacturing systems*, Sakarya, Turkey (2008), pp. 757-765.
- [13] M. R. Dastranj, M. Moghaddas, S. S. Afghu, and M. Rouhani, "PID control of inverted pendulum using particle swarm optimization (PSO) algorithm," *2011 IEEE 3rd Int. Conf. Commun. Softw. Networks, ICCSN 2011*, no. 2, pp. 575–578, 2011, doi: 10.1109/ICCSN.2011.6013972.

- [14] K. Singh, S. Nema, and P. K. Padhy, "Modified PSO based PID sliding mode control for inverted pendulum," *2014 Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT 2014*, pp. 722–727, 2014, doi: 10.1109/ICCICCT.2014.6993054.
- [15] A. Wu and P.K.S. Tam, A fuzzy neural network based on fuzzy hierarchy error approach, *IEEE Trans. Fuzzy Syst.*, vol. 8, pp.808-816, 2000.
- [16] Ahmad Ilyas Roose, Samer Yahya, Hussain Al-Rizzo, Fuzzy-logic control of an inverted pendulum on a cart, *Computers & Electrical Engineering*, Volume 61, 2017, Pages 31-47, ISSN 0045-7906, <https://doi.org/10.1016/j.compeleceng.2017.05.016>.
- [17] Mladenov, Valeri & Tsenov, Georgi & Ekonomou, Lambros & Harkiolakis, Nicholas & Karampelas, Panagiotis. (2009). Neural network control of an inverted pendulum on a cart. 8.
- [18] Qinghai B," The Analysis of Particle Swarm Optimization Algorithm", in *CCSE*, February 2010, vol.3.
- [19] Rao, R.V. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 7(1), 19-34.

Ngày nhận bài: 17/12/2020

Ngày chấp nhận đăng: 26/03/2021