

THIẾT KẾ BỘ ĐIỀU KHIỂN CHO MÔ HÌNH XE MÁY TỰ CÂN BẰNG PHỤC VỤ HỌC TẬP VÀ GIẢNG DẠY

Đường Khánh Sơn¹, Huỳnh Minh Vũ¹, Trần Sĩ Lâm¹, Nguyễn Quốc Anh²
và Lê Vĩ Khang²

¹Trường Đại học Kỹ thuật - Công nghệ Cần Thơ
Email: dkson@ctu.edu.vn

²Sinh viên Khoa Kỹ thuật Cơ khí, Trường Đại học Kỹ thuật - Công nghệ Cần Thơ

TÓM TẮT

Thông tin chung:

Ngày nhận bài:

05/7/2024

Ngày nhận bài sửa:

06/11/2024

Ngày duyệt đăng:

08/11/2024

Từ khóa:

Bánh đà, bộ điều khiển, xe máy tự cân bằng.

Bài báo này trình bày phương pháp thiết kế bộ điều khiển cho mô hình xe máy tự cân bằng với mục tiêu phục vụ cho nghiên cứu, học tập và giảng dạy. Mô hình động lực học của xe được mô phỏng và tính toán trên phần mềm MATLAB. Sau đó, mô hình 3D được thiết kế trên phần mềm Autodesk Inventor, công nghệ in 3D được sử dụng để chế tạo mô hình hoàn chỉnh. Cuối cùng hai bộ điều khiển PID và LQR được áp dụng vào để cân bằng cho xe. Kết quả thực nghiệm cho thấy bộ điều khiển LQR cho đáp ứng hệ thống tốt hơn, xe tự cân bằng ổn định ở góc nghiêng $\pm 8^\circ$ và dốc cao tới đa 40° . Nghiên cứu này không chỉ mang lại những kết quả thiết thực mà còn mở ra nhiều triển vọng ứng dụng và cải tiến công nghệ trong tương lai.

1. ĐẶT VẤN ĐỀ

Mô hình con lắc ngược là một hệ thống động lực học phi tuyến, thường được sử dụng trong giảng dạy ở bậc đại học về điều khiển tự động và kỹ thuật cơ khí. Tương tự, mô hình xe máy tự cân bằng có mô hình toán học tương đương với mô hình con lắc ngược, được biểu diễn qua các phương trình vi phân phi tuyến. Những phương trình này thể hiện sự phức tạp và tương tác giữa các lực tác động lên hệ thống. Xe máy tự cân bằng là công cụ lý tưởng để minh họa cho các nguyên lý cơ bản của điều khiển tự động, bao gồm việc ổn định hệ thống và điều khiển phi tuyến. Thiết kế và điều khiển hệ thống này không chỉ đòi hỏi sự áp dụng kiến thức lý thuyết vào thực tiễn mà còn giúp sinh viên phát triển các kỹ năng kỹ thuật và lập trình, tạo điều kiện tiếp cận việc triển khai các bộ điều khiển vào hệ

thống thực tế, từ đó nâng cao khả năng giải quyết vấn đề.

Theo Nikolaos (2022), bài toán cân bằng xe máy được xem như bài toán điều khiển một con lắc ngược, một bài toán kinh điển trong lý thuyết điều khiển. Trong bài toán này, xe máy đóng vai trò như một con lắc với trọng tâm nằm trên một trục quay, và nhiệm vụ của hệ thống điều khiển là giữ cho con lắc luôn thăng bằng bằng cách điều chỉnh các lực tác động lên nó. Việc giải quyết bài toán này đòi hỏi sự phối hợp chính xác giữa các cảm biến để đo lường trạng thái hiện tại của xe và các bộ điều khiển để thực hiện các hiệu chỉnh cần thiết.

Việc xây dựng mô hình xe máy điện tự cân bằng cung cấp môi trường thực tế để đánh giá hiệu quả và tối ưu hóa các thuật toán điều khiển khác nhau. Đề tài “Điều

hiển xe đạp tự cân bằng dùng giải thuật Fuzzy-LQR” của Hoàng (2018) đã nghiên cứu thành công mô hình toán mô tả hệ xe đạp cân bằng, xây dựng được thuật toán LQR và Fuzzy-LQR, xây dựng được bộ lọc Kalman để ước lượng tối ưu góc nghiêng của xe. Tuy nhiên, mô hình thực tế chưa thiết kế phần điều khiển góc lái và các thông số cho thuật toán Fuzzy-LQR chưa tối ưu do sử dụng phương pháp thử sai.

Nghiên cứu về xe hai bánh tự cân bằng có ứng dụng rộng rãi trong phát triển robot di động và các hệ thống cơ điện tử, cải thiện sự ổn định và khả năng điều khiển trong môi trường thực tế. Đề tài “Xe điện tự cân bằng” của Tuấn và Vinh (2018) cho thấy dự án tạo ra một giải pháp vận chuyển cá nhân thực tế, mô hình thực tế có phần điều khiển góc lái. Tuy nhiên, xe cân bằng chưa được ổn định, phạm vi góc cân bằng của xe nhỏ và chưa thể vừa di chuyển vừa cân bằng.

Sau khi phân tích và so sánh các đề tài trên dựa theo các tiêu chí như độ ổn định, độ chính xác, khả năng thích ứng và tính khả thi trong thực tế, bài báo này xây dựng mô hình xe máy tự cân bằng và thiết kế hai bộ điều khiển PID và LQR. Mục tiêu là đánh giá hiệu quả của từng bộ điều khiển khi áp dụng vào mô hình xe máy tự cân bằng, từ đó đưa ra lựa chọn giải pháp điều khiển phù hợp nhất. Mô hình hoàn thiện được điều khiển qua giao diện trên điện thoại sử dụng công cụ MIT App Inventor, giúp xe dễ dàng thực hiện các thao tác tiến, lùi, và bẻ lái.

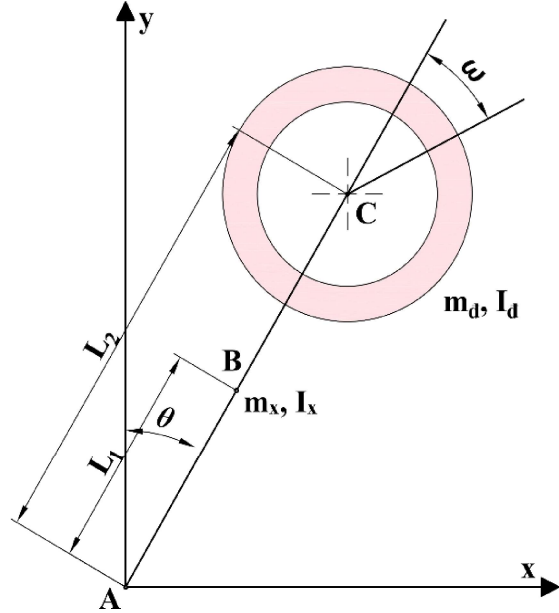
2. PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Mô hình hóa

2.1.1. Mô hình hóa xe máy hai bánh tự cân bằng

Mô hình xe được xây dựng theo Nikolaos Marios Patsouras (2022). Dựa trên nguyên lý

con lắc ngược có bánh đà quán tính, nếu bỏ qua các lực tạo ra khi di chuyển và đánh lái ta có thể rút ra mô hình động học bằng phương pháp Euler-Lagrange.



Hình 1. Mô hình toán của xe

Nguồn: Công bố của tác giả, (2024).

A là trục khớp quay giữa thanh con lắc với mặt đất, B là trọng tâm của xe và C là trọng tâm của bánh đà quán tính.

L1 và L2 lần lượt là khoảng cách từ trọng tâm xe đến A và khoảng cách từ trọng tâm bánh đà đến A.

Góc theta và omega là góc quay của xe và bánh đà, mx và md là khối lượng của xe và khối lượng của bánh đà. Ix và Id lần lượt là moment quán tính của xe và bánh đà.

Phương trình Lagrange cho hệ thống:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = Q \text{ và } L = T - V \quad (1)$$

Tổng động năng của xe:

$$T_x = \frac{1}{2} m_x |V_B|^2 + \frac{1}{2} I_x \dot{\theta}^2 \quad (2)$$

Tổng động năng của bánh đà:

$$T_d = \frac{1}{2} m_d |V_C|^2 + \frac{1}{2} I_d (\dot{\theta} + \dot{\omega})^2 \quad (3)$$

Tổng động năng của hệ thống:

$$T = T_x + T_d = \frac{1}{2}I_d\dot{\omega}^2 + I_d\dot{\theta}\dot{\omega} + \frac{1}{2}(m_xL_1^2 + m_dL_2^2 + I_x + I_d)\dot{\theta}^2 \quad (4)$$

Tổng thế năng của hệ thống:

$$V = m \cdot g \cdot \cos(\theta) = (m_xL_1 + m_dL_2) \cdot g \cdot \cos(\theta) \quad (5)$$

Hàm Lagrange thu được là:

$$L = T - V = \frac{1}{2}(m_xL_1^2 + m_dL_2^2 + I_x + I_d)\dot{\theta}^2 + \frac{1}{2}I_d\dot{\omega}^2 + I_d\dot{\theta}\dot{\omega} - (m_xL_1 + m_dL_2) \cdot g \cdot \cos(\theta) \quad (6)$$

Với $q = \theta$ thay vào (6), ta có:

$$(m_xL_1^2 + m_dL_2^2 + I_x + I_d)\ddot{\theta} + I_d\ddot{\omega} - (m_xL_1 + m_dL_2) \cdot g \cdot \sin(\theta) = 0 \quad (7)$$

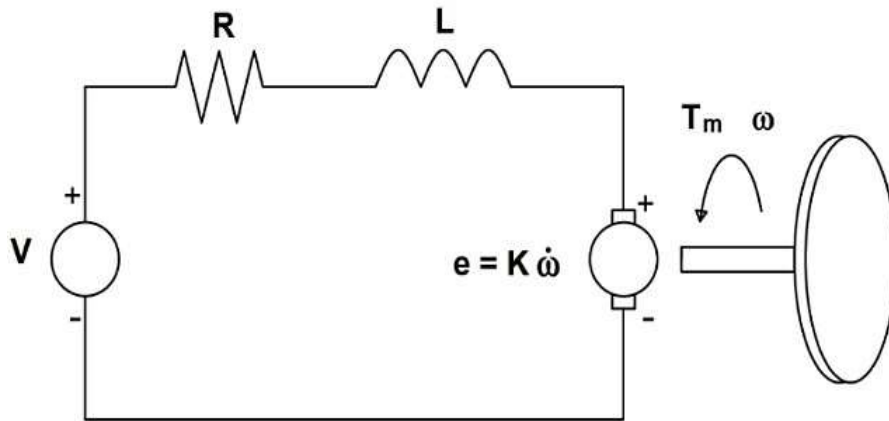
Với $q = \omega$, gọi T_m là moment xoắn tạo ra bởi bánh đà, thay vào (6), ta có:

$$I_d\ddot{\theta} + I_d\ddot{\omega} = T_m \quad (8)$$

Trong đó T_m được xác định từ mô hình toán của động cơ DC.

2.1.2. Mô hình động cơ DC

Mô hình động cơ DC đóng vai trò quan trọng trong việc hình thành các phương trình động học của hệ thống.



Hình 2. Mô hình toán động cơ

Nguồn: Công bố của tác giả, (2024).

Áp dụng định luật Kirchhoff:

$$R_m i + L_m \frac{di}{dt} + e = V_m \quad (9)$$

Với V_m là điện áp đầu vào, R_m là điện trở cuộn dây phần ứng, L_m là độ tự cảm của cuộn dây phần ứng, e là suất điện động tác động lên phần ứng động cơ.

$$e = K_e \dot{\omega} \quad (10)$$

Do moment quay của động cơ tỉ lệ với dòng điện I , ta có:

$$T_m = K_t i \quad (11)$$

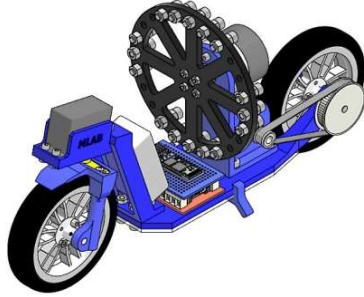
K_t là hằng số moment xoắn động cơ. Hệ số tự cảm được xem là không đáng kể ($L_m = 0$) do độ tự cảm của động cơ nhỏ hơn so với điện trở động cơ ($L_m \ll R_m$) suy ra:

$$T_m = \frac{K_t}{R_m} (V_m - K_e \dot{\omega}) \quad (12)$$

Từ (8) và (12), ta có:

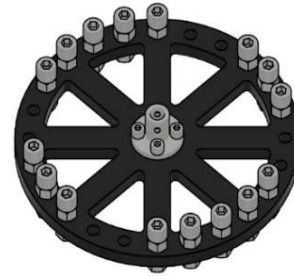
$$I_d \ddot{\theta} + I_d \dot{\omega} = \frac{K_t}{R_m} (V_m - K_e \dot{\omega}) \quad (13)$$

2.2. Thiết kế và chế tạo mô hình



Hình 3. Mô hình xe máy tự cân bằng

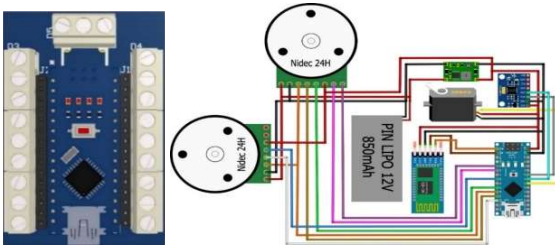
Mô hình xe máy tự cân bằng được thiết kế bằng phần mềm Inventor trình bày ở Hình 3 và 4.



Hình 4. Cụm bánh đà

Nguồn: Công bố của tác giả, (2024).

Mạch điện điều khiển thiết kế trên phần mềm Altium Designer được trình bày như Hình 5.



Hình 5. PCB và sơ đồ mạch điện

Nguồn: Công bố của tác giả, (2024).

Nguyên lý hoạt động:

Khối nguồn cung cấp nguồn điện cho hoạt động của toàn bộ hệ thống, vi điều khiển Arduino Nano xử lý dữ liệu góc đọc từ cảm biến MPU-6050 (độ phân giải ± 250 °/s, thời gian lấy mẫu 10 ms, góc đo nhỏ nhất của cảm biến là 0,0000763 độ) và tính toán xung cấp cho động cơ bánh đà, đồng thời nhận dữ liệu từ điện thoại qua Module HC05 điều khiển góc lái bằng Servo MG996R và động cơ dẫn động bánh sau.

2.3. Thiết kế bộ điều khiển

Mô hình sau khi được tính toán, gia công và lắp ráp có thông số như Bảng 1.

Giao diện điều khiển được thiết kế bằng MIT App Inventor như Hình 6.



Hình 6. Giao diện điều khiển

Nguồn: Công bố của tác giả, (2024).

Bảng 1. Thông số hệ thống của xe

Kí hiệu	Giá trị	Kí hiệu	Giá trị
L_1	0,0545 m	I_d	0,000431 Kg.m ²
L_2	0,109 m	g	9,81 m/s ²
m_x	0,784 Kg	K_t	0,0987 N.m/A
m_d	0,136 Kg	K_e	0,0987 V.m/rad
I_x	0,0013454 Kg.m ²	R_m	1,3629 Omh

Nguồn: Công bố của tác giả, (2024).

2.3.1. Tuyến tính hóa mô hình

Mô hình của xe là phương trình vi phân phi tuyến nên cần tuyến tính hóa xung quanh các vị trí cân bằng.

Khai triển chuỗi Taylor cho hàm số $\sin(\theta)$, ta có:

$$f(\theta) = \sin(\theta) = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \frac{\theta^9}{9!} - \dots \quad (14)$$

Vì góc θ rất nhỏ ($\theta \approx 0$) nên:

$$f(\theta) = \sin(\theta) \approx \theta \quad (15)$$

Từ (7) và (15), ta có phương trình tuyến tính hóa Taylor:

$$(m_x L_1^2 + m_d L_2^2 + I_x + I_d)\ddot{\theta} + I_d \dot{\omega} - (m_x L_1 + m_d L_2) \cdot g \cdot \theta = 0 \quad (16)$$

2.3.2. Thiết kế bộ điều khiển PID

Khi xe ở vị trí cân bằng, bánh đà sẽ ở hai trạng thái là đứng yên hoặc quay đều, khi động cơ quay đều ở vận tốc tối đa thì lúc này động cơ sẽ không sinh ra moment để tác dụng lên bánh đà. Vì vậy, vấn đề là cần bộ điều khiển PID cho góc nghiêng θ của xe và P cho vận tốc góc bánh đà $\dot{\omega}$. Tổng ngõ ra hai bộ điều khiển là điện áp cấp cho động cơ.

Tín hiệu điều khiển góc nghiêng θ của xe:

$$u_\theta = -(K_{P,\theta} \cdot \theta + K_{I,\theta} \cdot \int \theta + K_{D,\theta} \cdot \dot{\theta}) \quad (17)$$

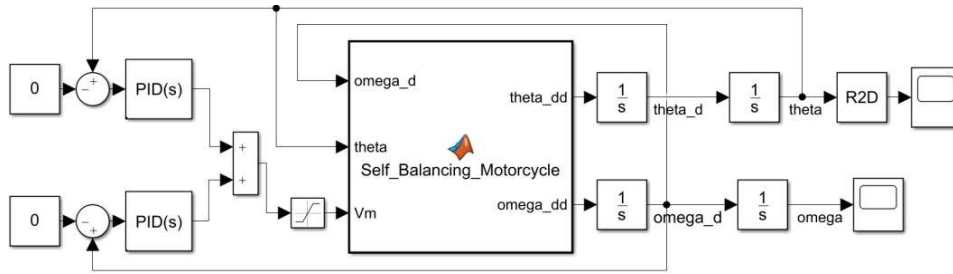
Tín hiệu điều khiển vận tốc góc bánh đà $\dot{\omega}$:

$$u_{\dot{\omega}} = -(K_{P,v} \cdot \dot{\omega}) \quad (18)$$

Tín hiệu điều khiển tạo ra:

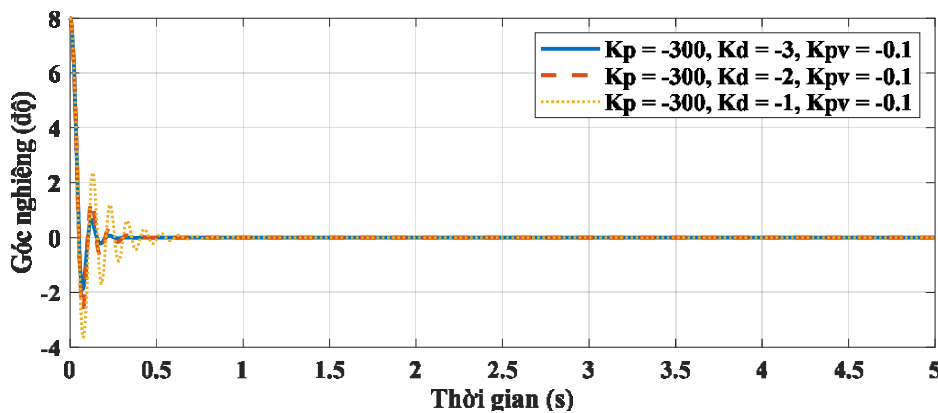
$$u = u_\theta + u_{\dot{\omega}} \quad (19)$$

Các thông số của bộ điều khiển PID và hệ số $K_{p,v}$ được tìm bằng phương pháp thử sai mà mô phỏng trên Simulink. Sơ đồ mô phỏng bộ điều khiển PID được thể hiện ở Hình 7 và đáp ứng góc nghiêng θ của xe với các hệ số PID được thể hiện ở Hình 8.



Hình 7. Sơ đồ mô phỏng bộ điều khiển PID

Nguồn: Công bố của tác giả, (2024).



Hình 8. Đáp ứng góc nghiêng của xe

Nguồn: Công bố của tác giả, (2024).

2.3.3. Thiết kế bộ điều khiển LQR

Điều khiển LQR là bài toán điều khiển tối ưu tuyến tính, bộ điều khiển chỉ áp dụng với các đối tượng tuyến tính. Đối tượng tuyến tính mô tả bởi phương trình trạng thái sau đây:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \mathbf{x}(0) = \mathbf{x}_0 \quad (20)$$

Điều khiển LQR là bài toán xác định ma trận \mathbf{K} của luật điều khiển $u(t) = -\mathbf{K}\mathbf{x}(t)$ thỏa mãn chỉ tiêu J đạt cực tiểu. J được tính theo công thức:

$$J = \frac{1}{2} \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + u^T \mathbf{R} u) dt \quad (21)$$

Với \mathbf{Q} và \mathbf{R} là các ma trận xác định dương. Điều khiển LQR thuộc điều khiển tối ưu khi đó ma trận \mathbf{K} được xác định theo công thức.

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (22)$$

Trong đó: \mathbf{P} là nghiệm của phương trình đại số Riccati:

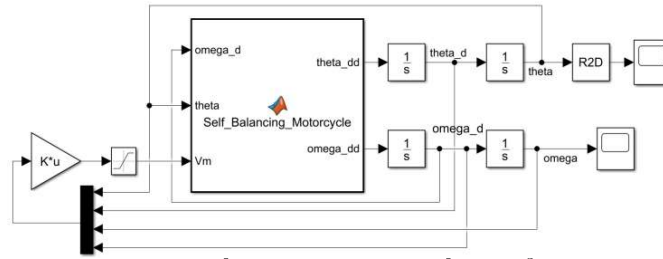
$$\mathbf{P}\mathbf{A} + \mathbf{A}^T \mathbf{P} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P} = 0 \quad (23)$$

Ma trận \mathbf{Q} có dạng:

$$\mathbf{Q} = \begin{bmatrix} Q_{11} & 0 & 0 & 0 \\ 0 & Q_{22} & 0 & 0 \\ 0 & 0 & Q_{33} & 0 \\ 0 & 0 & 0 & Q_{44} \end{bmatrix}$$

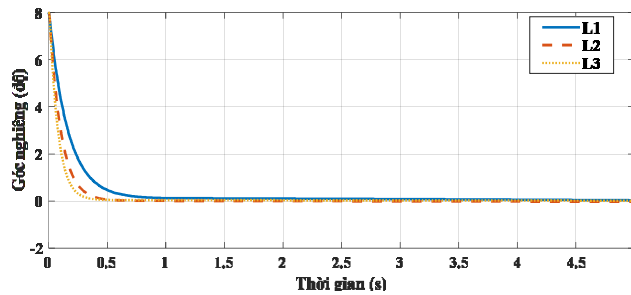
Trong đó: Q_{11} , Q_{22} , Q_{33} , Q_{44} lần lượt là trọng số của góc nghiêng θ , vận tốc góc của xe $\dot{\theta}$, góc quay ω và vận tốc góc $\dot{\omega}$ của bánh đà.

Các trọng số của bộ điều khiển LQR được lựa chọn bằng phương pháp thử sai trên MATLAB để tìm ra bộ thông số thích hợp cho hệ thống. Sơ đồ mô phỏng bộ điều khiển LQR được thể hiện ở Hình 9 và đáp ứng góc nghiêng θ của xe với các hệ số LQR được thể hiện ở Hình 10.



Hình 9. Sơ đồ mô phỏng bộ điều khiển LQR

Nguồn: Công bố của tác giả, (2024).



Hình 10. Đáp ứng góc nghiêng của xe với bộ điều khiển LQR

Nguồn: Công bố của tác giả, (2024).

3. KẾT QUẢ VÀ THẢO LUẬN

3.1. Kết quả

3.1.1. Kết quả chế tạo mô hình



Hình 11. Mô hình xe và cụm bánh đà sau khi lắp ráp hoàn chỉnh

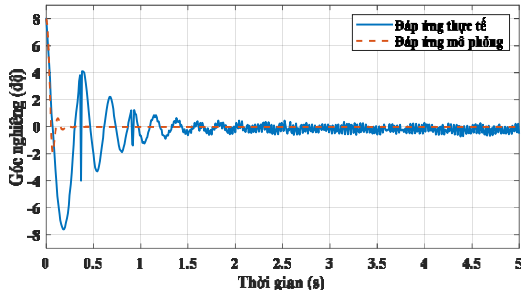
Nguồn: Công bố của tác giả, (2024).

3.1.2. Kết quả thực nghiệm bộ điều khiển

Đối với bộ điều khiển PID, mô hình thực tế được thực nghiệm với góc nghiêng ban đầu của xe là 8 độ và các thông số là: $K_{P,\theta} = 138$, $K_{D,\theta} = 2$, $K_{P,v} = 0,0006$. Khi tiến hành mô phỏng, tác giả sử dụng bộ hồi tiếp âm, trong khi trong điều kiện vận hành thực tế lại áp dụng hồi tiếp dương. Do đó, các thông số của bộ điều khiển sẽ khác nhau về dấu. Tuy nhiên, đáp ứng của

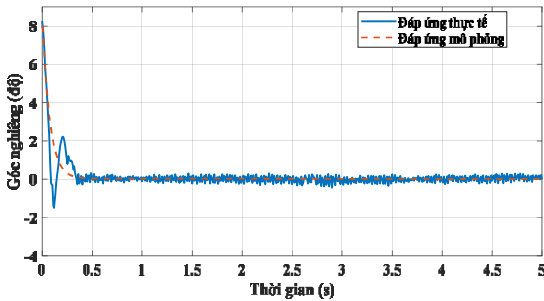
Hình 11 trình bày mô hình xe và cụm bánh đà sau khi chế tạo và lắp ráp hoàn chỉnh.

hai bộ điều khiển nêu trên vẫn tương đồng. Ngoài ra, do sai số trong chế tạo, trọng tâm của xe trong mô hình thực tế có xu hướng lệch sang một bên, nên các thông số của bộ điều khiển cũng có sự sai lệch nhất định. Đồ thị ở Hình 12 thể hiện đáp ứng thực tế của xe. Kết quả cho thấy thời gian xác lập của hệ thống là 1,5 giây, sai số xác lập khoảng 0,5 độ, độ vọt lố cao 7,8 độ, trong thời gian từ 0 đến 1,5 giây hệ thống còn dao động khá nhiều.



Hình 12. Đáp ứng thực tế của xe với bộ điều khiển PID

Nguồn: Công bố của tác giả, (2024).



Hình 13. Đáp ứng thực tế của xe với bộ điều khiển LQR

Nguồn: Công bố của tác giả, (2024).

Đối với bộ điều khiển LQR, đáp ứng thực tế của xe được thực nghiệm với góc nghiêng ban đầu của xe là 8 độ và hệ số ma trận $K = [100 \ 12 \ 22 \ 0,0006]$. Đồ thị trên Hình 13 thể hiện đáp ứng của thực tế của xe, kết quả cho thấy, thời gian xác lập của hệ thống 0,3 giây, sai số xác lập khoảng 0,2 độ, xe ổn định trong khoảng $\pm 0,5$ độ, độ vọt lố 2,1 độ.

Ổn định là điều kiện cần đối với một hệ điều khiển tự động, song chưa phải là đủ để hệ thống được sử dụng thực tế. Ngoài ra, nhiều yêu cầu đòi hỏi hệ thống phải thỏa mãn được cùng một lúc các tiêu chuẩn khác nhau như độ chính xác, độ ổn định, đáp ứng quá độ, độ nhạy... Theo tiêu chuẩn IAE, ta có công thức (24) tính tích phân trị tuyệt đối biên độ sai số như sau:

$$I_{IAE} = \int_0^T |e(t)| dt \quad (24)$$

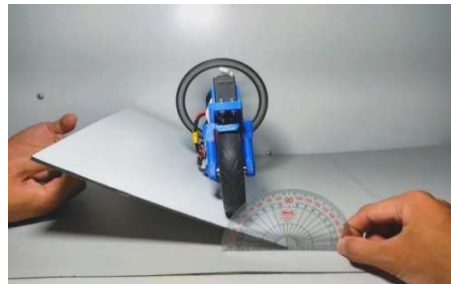
Bảng 2: Giá trị J_{IAE}

Bộ điều khiển	Giá trị J_{IAE}
PID	380,69
LQR	136

Bảng 2 cho thấy giá trị J_{IAE} của bộ điều khiển LQR nhỏ hơn PID. Với các thông số được lựa chọn như trên, chất lượng của bộ điều khiển LQR đáp ứng tốt hơn trong hệ thống xe máy hai bánh tự cân bằng.

3.1.3. Kết quả các chế độ cân bằng

Hình 14 thực nghiệm chế độ cân bằng trên mặt phẳng nghiêng. Thử nghiệm được tiến hành bằng cách đặt xe đang ở trạng thái cân bằng lên một tấm bảng, sau đó từ từ nâng tấm bảng lên đến khi xe ngã. Kết quả thử nghiệm thực tế cho thấy xe có thể cân bằng ở mặt phẳng nghiêng tối đa 20°. Hình 15 thực nghiệm cân bằng của xe ở dốc cao. Thử nghiệm được tiến hành tương tự như thử nghiệm trước. Kết quả thử nghiệm thực tế cho thấy xe có thể cân bằng ở dốc cao lên đến 40°.



Hình 14. Chế độ cân bằng trên mặt nghiêng 20°

Nguồn: Công bố của tác giả, (2024).



Hình 15. Chế độ cân bằng dốc cao

Nguồn: Công bố của tác giả, (2024).

3.2. Thảo luận

Các kết quả mô phỏng và thực nghiệm cho thấy bộ điều khiển LQR cho hiệu quả vượt trội so với PID, giảm thiểu được dao động và độ vọt lố, đồng thời rút ngắn thời gian xác lập. Kết quả thực nghiệm được so sánh với kết quả nghiên cứu của đề tài “Điều khiển xe đạp tự cân bằng dùng giải thuật Fuzzy-LQR” của tác giả Nguyễn Huy Hoàng (2018). So với đề tài này nghiên cứu cho thời gian xác lập nhanh hơn (0,3 giây so với 2 giây) nhưng ngược lại độ vọt lố cao hơn (2,1 độ so với 1 độ). Tuy nhiên, hệ thống chỉ tiến hành thực nghiệm ở góc cân bằng $\pm 8^\circ$ và chưa có bộ điều khiển Swing up để xử lý các góc nghiêng lớn hơn. Vì vậy, vấn đề này tác giả sẽ nghiên cứu áp dụng thêm các giải thuật điều khiển mới như AI, Noron để tìm ra giải thuật có kết quả điều khiển tốt nhất.

4. KẾT LUẬN

Đề tài chế tạo thành công xe máy tự cân bằng, áp dụng được bộ điều khiển PID và LQR để so sánh chất lượng của hai bộ điều khiển. Nghiên cứu này không chỉ là sự kết hợp chặt chẽ giữa lý thuyết điều khiển, thiết kế cơ khí, lập trình nhúng và cảm biến, mà còn mang lại những kết quả thực tiễn và ý nghĩa cho học tập và nghiên cứu khoa học. Nhìn chung, nghiên cứu đã góp phần quan trọng vào việc phát triển công nghệ xe máy tự cân bằng, mở ra nhiều triển vọng ứng dụng thực tiễn và cải tiến trong tương lai.

Tài liệu tham khảo

Kiattisin Kanjanawanishkul, (2015), “LQR and MPC controller design and comparison for a stationary Self-Balancing Bicycle Robot with a reaction wheel”. *Kybernetika*, Volume 51, Number 1, pp. 173 – 191.

Nguyễn Huy Hoàng, (2018), “Điều khiển xe đạp tự cân bằng dùng giải thuật Fuzzy-LQR”, Luận văn thạc sĩ, Trường Đại học Bách Khoa, Tp. Hồ Chí Minh.

Nikolaos Marios Patsouras, (2022), “Modelling and control of a self-balancing motorcycle”. Diploma Thesis, University of Thessaly.

Nguyễn Thị Phương Hà - Huỳnh Thái Hoàng, (2005), Lý thuyết điều khiển tự động. Nhà xuất bản Đại học Quốc gia TP. Hồ Chí Minh.

Nguyễn Anh Tuấn và Nguyễn Quang Vinh (2018), “Xe điện tự cân bằng”, Đồ án tốt nghiệp, Trường Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh.

Q. K. Ho and C. B. Pham, (2018), "Study on Inertia Wheel Pendulum Applied to Self-Balancing Electric Motorcycle", 4th International Conference on Green Technology and Sustainable Development (GTSD), Ho Chi Minh City, Vietnam, pp. 687-692.

CONTROLLER DESIGN FOR A SELF-BALANCING MOTORCYCLE MODEL FOR LEARNING AND TEACHING PURPOSES

ABSTRACT

This paper presents a method for designing a controller for a self-balancing motorcycle model aimed at supporting research, learning, and teaching purposes. The motorcycle's dynamic model is simulated and calculated using MATLAB software. Subsequently, a 3D model is designed with Autodesk Inventor software, and 3D printing technology is employed to fabricate the complete model. Finally, two controllers, PID and LQR, are implemented to balance the motorcycle. Experimental results show that the LQR controller provides better system response, allowing the motorcycle to balance stably at a tilt angle of $\pm 8^\circ$ and on a maximum slope of 40° . This research not only yields practical outcomes but also opens promising prospects for future technological applications and advancements.

Keywords: *Controler, flywheel, self-balancing motorcycle*