

Nghiên cứu mô phỏng và thiết kế robot xe cân bằng

Mai Nhật Thiên^a, Nguyễn Hoàng Phúc^{b*}

Tóm tắt:

Bài báo trình bày cách thiết kế và cách thức xây dựng robot xe cân bằng hai bánh, thông qua sự kết hợp của con quay hồi chuyển và gia tốc MPU6050 thiết lập thiết bị phát hiện vị trí của xe. Trong đó, nhiệm vụ của bộ điều khiển Arduino UNO là tiếp nhận và xử lý thông tin để duy trì trọng tâm của xe đứng yên với sai số nhỏ về góc nghiêng của thân xe với nền khi xe chuyển động. Đặc biệt, hệ thống điều khiển sử dụng bộ điều khiển PID giúp xe có thể nhanh chóng đạt được trạng thái ổn định khi có tác động của một lượng giao thoa nhiễu từ môi trường.

Từ khóa: *xe cân bằng hai bánh, MPU6050, Arduino UNO, PID, mạch cầu L298N*

^a Khoa Điện - Điện tử, Trường Đại học Sư phạm Kỹ thuật Vĩnh Long; 73 Nguyễn Huệ, phường 2, thành phố Vĩnh Long, tỉnh Vĩnh Long. e-mail: thienmn@vlute.edu.vn

^b Khoa Điện - Điện tử, Trường Đại học Sư phạm Kỹ thuật Vĩnh Long; 73 Nguyễn Huệ, phường 2, thành phố Vĩnh Long, tỉnh Vĩnh Long. e-mail: 21007051@st.vlute.edu.vn

* Tác giả chịu trách nhiệm chính.

Research and Design of Balanced Robot.

Mai Nhat Thien^a, Nguyen Hoang Phuc^{b*}

Abstract:

The paper presents the design and construction of a two-wheeled self balancing robot, through the combination of gyroscope and accelerometer MPU6050 to set up the vehicle's detected position. In which, the task of the Arduino UNO control board is to receive and process information for maintaining the vehicle's center of gravity stationary as well as to make small errors in the tilt angle of the vehicle body with the ground that the vehicle is moving. In particular, the PID controller is used to quickly help the vehicle by reaching a stable state that affects a certain amount of the interferenced environment.

Keywords: *two-wheeled balance vehicle, MPU6050, Arduino UNO, PID, L298N bridge circuit*

Received: 6.5.2024; Accepted: 15.12.2024; Published: 30.12.2024

10.59907/daujs.3.4.2024.317

^a Faculty of Electrical and Electronics, Vinh Long University of Technology; 73 Nguyen Hue Street, Ward 2, Vinh Long City, Vinh Long Province, Vietnam. e-mail: thienmn@vlute.edu.vn

^b Faculty of Electrical and Electronics, Vinh Long University of Technology; 73 Nguyen Hue Street, Ward 2, Vinh Long City, Vinh Long Province, Vietnam. e-mail: 21007051@st.vlute.edu.vn

* *Corresponding author.*

Đặt vấn đề

Xe cân bằng hai bánh hoạt động theo nguyên lý con lắc ngược. Robot có thể tự giữ cân bằng. Nghiên cứu về phương pháp điều khiển và thực nghiệm mô hình robot tự cân bằng hai bánh được thực hiện bởi nhóm nghiên cứu M.R.M. Romlay và cộng sự (Shekhawat & Rohilla, 2020). Ngoài ra, nhóm tác giả Ian D. Loram Martin Lakie đã nghiên cứu điều khiển vị trí bằng cách chuyển động ngắn cho con lắc ngược tự cân bằng thân người (Loram & Lakie, 2002).

Mô hình xe tự cân bằng hai bánh được nghiên cứu xây dựng mô hình toán học và mô phỏng trên các phần mềm Matlab/Simulink dùng trong giảng dạy tại các trường (Hellman & Sunnerman). Ngoài ra mô hình động lực học trên con lắc ngược đã được nghiên cứu trước đó (Glad & Ljung, 2006; Ooi, 2003). Ngày nay cũng có nhiều nhóm nghiên cứu về robot xe cân bằng hai bánh như nhóm M.R.M. Romlay và các cộng sự (Romlay et al., 2019).

Trong bài báo này, chúng tôi nghiên cứu về bộ điều khiển PID cho xe cân bằng, trọng tâm của bài báo trình bày nói về mô hình toán học và mô phỏng thực nghiệm bằng phần mềm Matlab/Simulink trên đối tượng xe cân bằng hai bánh. Bài báo này được tóm tắt bao gồm mô tả phương trình toán học, kết nối mạch điện của mô hình thực nghiệm và trình bày kết quả đạt được và một số nhận xét được đưa ra.

Đối tượng và phương pháp nghiên cứu

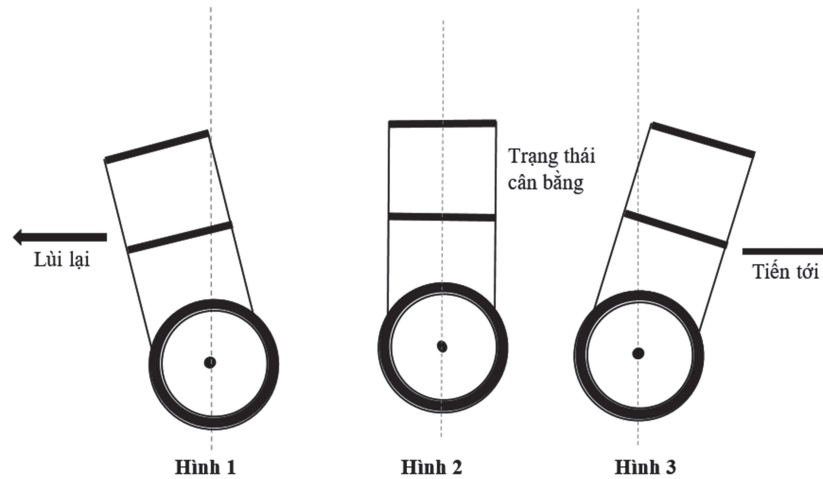
Đối tượng

Có thể cho rằng ví dụ phổ biến nhất về con lắc ngược ổn định là con người. Một người đứng thẳng hoạt động như một con lắc ngược với chân làm trục và nếu không điều chỉnh cơ bắp nhỏ liên tục thì sẽ bị ngã. Hệ thống thần kinh của con người chứa một hệ thống kiểm soát phản hồi vô thức, cảm giác thăng bằng hoặc phản xạ giữ thăng bằng, sử dụng đầu vào cảm nhận từ mắt, cơ và khớp cũng như đầu vào định hướng từ hệ thống tiền đình bao gồm ba kênh bán khuyên ở tai trong, và hai cơ quan sỏi tai, để thực hiện những điều chỉnh nhỏ liên tục cho các cơ xương để giữ cho chúng ta đứng thẳng.

Khi hệ xe có xu hướng ngã về phía trước thì lúc đó hệ thống sẽ tác động vào hai động cơ, đưa cho hệ xe trở về vị trí cân bằng. Hình 1 xe có xu hướng ngã về phía bên tay trái, lúc này hệ thống sẽ cấp xung hợp lí vào động cơ để điều khiển động cơ chạy về hướng phía bị nghiêng và cân bằng. Tương tự như cho hình 3.

Để ổn định con lắc ở vị trí đảo ngược này, có thể sử dụng hệ thống điều khiển phản hồi, theo dõi góc của con lắc và di chuyển vị trí của điểm xoay sang một bên khi con lắc bắt đầu rơi xuống, để giữ cho nó cân bằng như hình 2. Con lắc ngược là một bài toán kinh điển trong lý thuyết động lực học và điều khiển và được sử dụng rộng rãi làm chuẩn mực

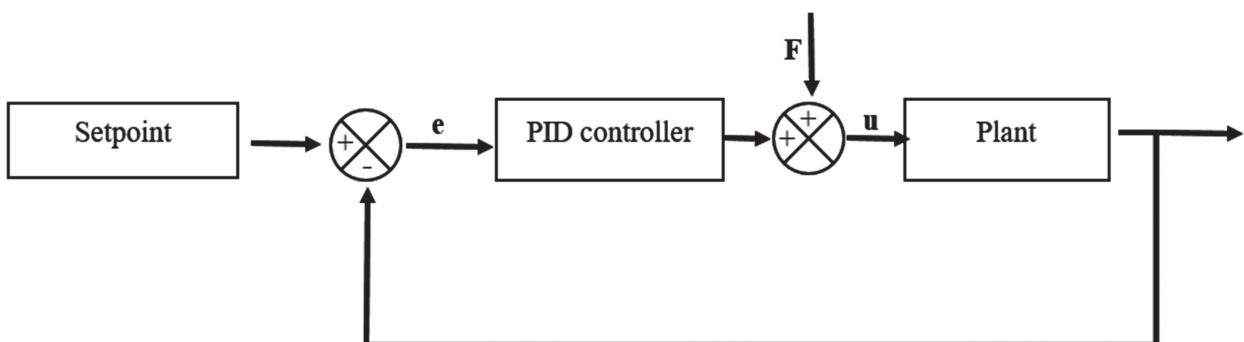
để thử nghiệm các thuật toán điều khiển (bộ điều khiển PID, mạng nơ-ron, điều khiển mờ, thuật toán di truyền...).



Phương pháp nghiên cứu

Robot cân bằng hai bánh hoạt động sử dụng bộ điều khiển Arduino UNO với bộ cảm biến MPU (cảm biến kết hợp gia tốc kế và con quay hồi chuyển) dựa vào phương pháp điều khiển PID (Tỷ lệ-Tích phân-Đạo hàm). Cảm biến MPU, bao gồm con quay hồi chuyển và gia tốc kế, liên tục theo dõi hướng và chuyển động của robot. Con quay hồi chuyển phát hiện vận tốc góc, cung cấp thông tin chi tiết về góc nghiêng của rô-bốt và tốc độ nghiêng về phía trước hoặc phía sau, trong khi gia tốc kế đo gia tốc tuyến tính, cho biết liệu rô-bốt có nghiêng sang một bên hay không. Phương pháp điều khiển PID, đóng vai trò là bộ não của hệ thống, liên tục so sánh góc thẳng đứng mong muốn với góc thực tế được phát hiện bởi cảm biến MPU. Bộ điều khiển PID điều chỉnh tốc độ động cơ của bánh xe robot qua mạch cầu L298N. Quá trình lặp đi lặp lại này đảm bảo rô-bốt tự động điều chỉnh tốc độ bánh xe để chống lại những sai lệch so với vị trí cân bằng mong muốn, cho phép rô-bốt duy trì ổn định và thẳng đứng trong nhiều điều kiện khác nhau.

Ta có sơ đồ bộ điều khiển PID như sau:



Hình 4. Sơ đồ bộ điều khiển PID

- e : tín hiệu sai số.
- F : lực nhiễu môi trường bên ngoài.
- u : điện áp điều khiển.

Khi đó ta có được ngõ ra của bộ điều khiển PID là:

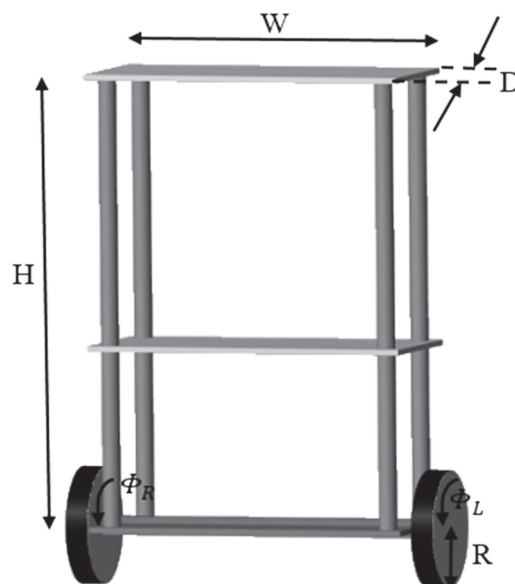
$$PID_{out} = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (1)$$

Trong đó

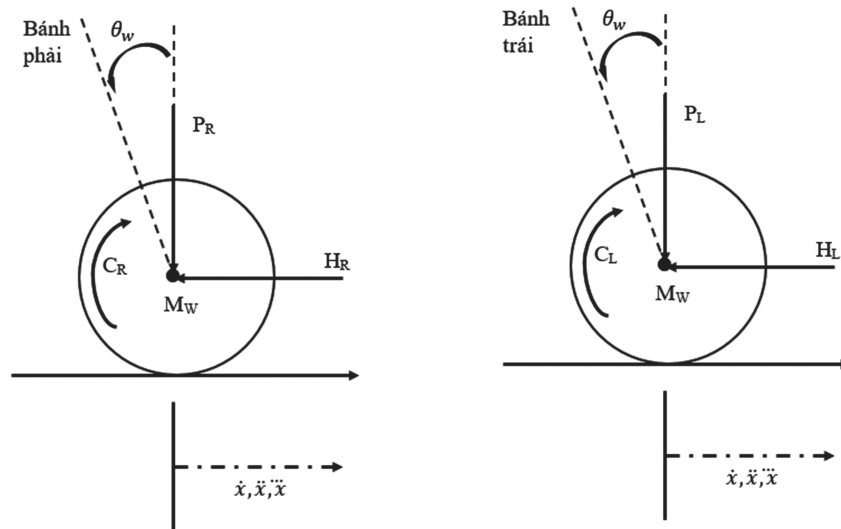
- K_p , K_i , K_d lần lượt là hằng số tỷ lệ, tích phân và đạo hàm tương ứng với các bộ điều khiển tỷ lệ, tích phân và đạo hàm.
- $e(t)$ là tín hiệu lỗi thể hiện góc lệch trục X mong muốn và thực tế của robot.

Giá trị tỉ lệ xác định tác động của sai số hiện tại, giá trị tích phân xác định tác động của tổng các sai số quá khứ, và giá trị vi phân xác định tác động của tốc độ biến đổi sai số. Tổng chập của ba tác động này dùng để điều chỉnh quá trình thông qua một phần tử điều khiển như vị trí của van điều khiển hay bộ nguồn của phần tử gia nhiệt. Nhờ vậy, những giá trị này có thể làm sáng tỏ về quan hệ thời gian: P phụ thuộc vào sai số hiện tại, I phụ thuộc vào tích lũy các sai số quá khứ, và D dự đoán các sai số tương lai, dựa vào tốc độ thay đổi hiện tại.

Sơ đồ và lực tác động lên hệ xe hai bánh tự cân bằng được thể hiện như ở hình 5, 6.



Hình 5. Hình ảnh mô tả robot xe hai bánh cân bằng



Hình 6. Phân tích lực tác động lên 2 bánh xe

Trong đó, các thành phần của hệ được liệt kê ở bảng 1:

Bảng 1. Ký hiệu và ý nghĩa của các đại lượng

Ký hiệu	Đơn vị	Ý nghĩa
M_p	Kg	Khối lượng robot
M_w	Kg	Khối lượng bánh xe
r	m	Bán kính bánh xe
θ_P	Rad	Góc nghiêng của thân robot
θ_w	Rad	Góc xoay của bánh xe
$\dot{\theta}_w$	Rad/s	Vận tốc góc của bánh xe
R	Ohms	Điện trở của động cơ
L	H	Độ tự cảm của dây quấn động cơ
k_f	Nms/rad	Hệ số ma sát
k_m	Nm/A	Hệ số moment
K_e	Vs/rad	Hệ số Back emf của động cơ
α	rad/s ²	Gia tốc góc của động cơ
V_a	V	Điện áp cấp cho động cơ
V_e	V	Điện áp phản ứng
i	A	Dòng điện phản ứng
I_R	Kgm ²	Moment quán tính của động cơ
τ_m	Nm	Moment của động cơ

Ký hiệu	Đơn vị	Ý nghĩa
I_w	Kg.m ²	Moment quán tính của bánh xe
I_p	Kg.m ²	Moment quán tính của thân xe
l	m	Khoảng cách từ tâm bánh xe đến tâm trọng trường của thân xe
H_L, H_R, P_L, P_R	N	Lực tác động giữa bánh xe và thân robot
C_L, C_R	Nm	Moment động cơ tác động lên bánh xe
H_{fL}, H_{fR}	N	Lực ma sát giữa bánh xe và mặt phẳng
β		Hệ số hiệu chỉnh

Lý thuyết điều khiển

Mô hình động học của xe động cơ DC

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-k_m k_e}{I_{RR}} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{I_{RR}} & \frac{-1}{I_R} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \quad (2)$$

Áp dụng định luật Newton tính được tổng lực tác động lên bánh xe theo phương ngang:

$$\sum F_x = Ma \quad (3)$$

$$M_w \ddot{x} = H_{fR} - H_R \quad (4)$$

Tổng lực tác động vào trục bánh xe:

$$\sum M_o = I\alpha \quad (5)$$

$$I_w \ddot{\theta}_w = C_R - H_{fR} r \quad (6)$$

Từ phương trình động học của động cơ DC, ta có moment của động cơ:

$$\tau_m = I_R \frac{d\omega}{dt} + \tau_a \quad (7)$$

Từ phương trình động học và phương trình vi phân của động cơ DC, ta có moment ngõ ra tổng quát của động cơ của động cơ:

$$C = I_R \frac{d\omega}{dt} = \frac{-k_m k_e}{R} \dot{\theta}_w + \frac{k_m}{R} V_a \quad (8)$$

Thay phương trình (8) vào phương trình (6) ta được:

$$I_w \ddot{\theta}_w = \frac{-k_m k_e}{R} \dot{\theta}_w + \frac{k_m}{R} V_a - H_{fR} r \quad (9)$$

Suy ra:

$$H_{fR} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w \quad (10)$$

Phương trình cho bánh trái:

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w - H_L \quad (11)$$

Phương trình cho bánh phải:

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w - H_R \quad (12)$$

Vì chuyển động tuyến tính tác dụng lực lên trục động cơ, vận tốc góc có thể chuyển thành vận tốc tuyến tính theo phương trình sau:

$$\ddot{\theta} \dots r = \ddot{x} \Rightarrow \ddot{\theta} \dots = \frac{\ddot{x}}{r} \quad (13)$$

$$\dot{\theta}_w r = \dot{x} \Rightarrow \dot{\theta}_w = \frac{\dot{x}}{r} \quad (14)$$

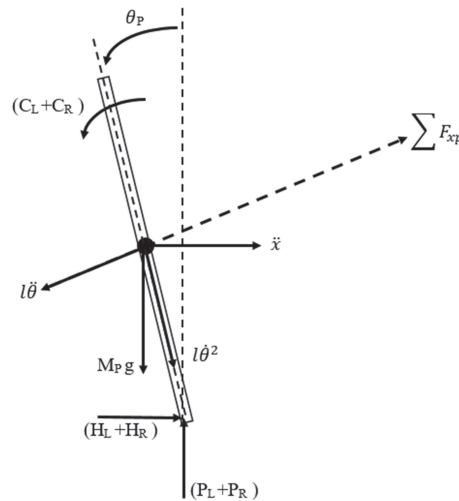
Từ phương trình (13) và (14) ta thay vào phương trình (11) và (12) ta được:

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr^2} \dot{x} + \frac{k_m}{Rr} V_a - \frac{I_w}{r^2} \ddot{x} - H_L \quad (15)$$

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr^2} \dot{x} + \frac{k_m}{Rr} V_a - \frac{I_w}{r^2} \ddot{x} - H_R \quad (16)$$

Tính tổng hai phương trình của hai bánh xe:

$$2(M_w + \frac{I_w}{r^2}) \ddot{x} = \frac{-2k_m k_e}{Rr^2} \dot{x} + \frac{2k_m}{Rr} V_a - (H_L + H_R) \quad (17)$$



Hình 7. Phân tích lực tác động lên thân bánh xe

Áp dụng định luật Newton tính được tổng lực tác động lên bánh xe theo phương ngang:

$$\sum F_x = M_p \ddot{x} \quad (18)$$

$$(H_L + H_R) - M_p l \ddot{\theta}_p \cos \theta_p + M_p l \dot{\theta}_p^2 \sin \theta_p = M_p \ddot{x} \quad (19)$$

Vì vậy,

$$(H_L + H_R) = M_p \ddot{x} + M_p l \ddot{\theta}_p \cos \theta_p - M_p l \dot{\theta}_p^2 \sin \theta_p \quad (20)$$

Lực vuông góc tác động lên thân xe:

$$\sum F_{xp} = M_p \ddot{x} \cos \theta_p \quad (21)$$

$$(H_L + H_R) \cos \theta_p + (P_L + P_R) \sin \theta_p - M_p g \sin \theta_p - M_p l \ddot{\theta}_p = M_p \ddot{x} \cos \theta_p \quad (22)$$

Tổng moment tác động lên trọng tâm của thân xe:

$$\sum M_o = I \alpha \quad (23)$$

$$-(H_L + H_R) l \cos \theta_p - (P_L + P_R) \sin \theta_p - (C_L + C_R) = I_p \ddot{\theta}_p \quad (24)$$

Ta có:

$$C_L + C_R = \frac{-2k_m k_e \dot{x}}{R} + \frac{2k_m}{R} V_a \quad (25)$$

Thay phương trình (25) vào phương trình (24) ta có:

$$-(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin(\theta_p) - \left(-\frac{2k_m k_e}{Rr} \dot{x} + \frac{2k_m}{R} V_a \right) = I_p \ddot{\theta}_p \quad (26)$$

Nhân -1 vào phương trình (22) và thế phương trình (26) vào đó ta có:

$$I_p \ddot{\theta}_p - \frac{2k_m k_e}{Rr} \dot{x} + \frac{2k_m}{R} V_a + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta}_p = -M_p l \ddot{x} \cos \theta_p \quad (27)$$

Thế phương trình (20) vào (17) ta có:

$$2(M_w + \frac{I_w}{r^2}) \ddot{x} = \frac{-2k_m k_e}{Rr^2} \dot{x} + \frac{2k_m}{Rr} V_a - M_p \ddot{x} - M_p l \ddot{\theta}_p \cos \theta_p + M_p l \dot{\theta}_p^2 \sin \theta_p \quad (28)$$

Hai phương trình (27) và (28) có thể tuyến tính hóa bằng cách giả sử ta cho $\theta_p = \pi + \phi$ trong đó ϕ là một góc nhỏ theo phương thẳng đứng lên trên. Việc đơn giản hóa đã được sử dụng để thu được mô hình tuyến tính có thể triển khai các bộ điều khiển không gian trạng thái tuyến tính.

Do đó, ta có:

$$\begin{aligned} \cos \theta_p &= -1 \\ \sin \theta_p &= -\phi \end{aligned} \quad (29)$$

$$\left(\frac{d\theta_p}{dt} \right)^2 = 0$$

Phương trình tuyến tính hóa của chuyển động là:

$$(I_p + M_p l^2) \ddot{\phi} - \frac{2k_m k_e}{Rr} \dot{x} + \frac{2k_m}{R} V_a - M_p g l \phi = M_p l \ddot{x} \quad (30)$$

$$\frac{2k_m}{Rr} V_a = \left(2M_w + \frac{2I_w}{r^2} + M_p \right) \ddot{x} + \frac{2k_m k_e}{Rr^2} \dot{x} - M_p l \ddot{\phi} \quad (31)$$

Để có được mô hình biến trạng thái của hệ thống ta rút các biến trạng thái từ phương trình (30) và (31), ta có:

$$\ddot{\phi} = \frac{M_p l}{(I_p + M_p l^2)} \ddot{x} + \frac{2k_m k_e}{Rr(I_p + M_p l^2)} \dot{x} - \frac{2k_m}{R(I_p + M_p l^2)} V_a + \frac{M_p g l}{(I_p + M_p l^2)} \phi \quad (32)$$

$$\ddot{x} = \frac{2k_m}{Rr(2M_w + \frac{2I_w}{r^2} + M_p)} V_a - \frac{2k_m k_e}{Rr^2(2M_w + \frac{2I_w}{r^2} + M_p)} \dot{x} + \frac{M_p l}{(2M_w + \frac{2I_w}{r^2} + M_p)} \ddot{\phi} \quad (33)$$

- Bằng cách thay hai phương trình (32) và (33) vào lần lượt các phương trình (30) và (31) ta được hệ không gian trạng thái như sau:

$$\begin{bmatrix} \dot{x} \\ \dot{x} \\ \dot{\phi} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_e (M_p l r - I_p - M_p l^2)}{Rr^2 \alpha} & \frac{M_p^2 g l^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e (r\beta - M_p l)}{Rr^2 \alpha} & \frac{M_p g l \beta}{\alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m (I_p + M_p l^2 - M_p l r)}{Rr \alpha} \\ 0 \\ \frac{2k_m (M_p l - r\beta)}{Rr \alpha} \end{bmatrix} V_a \quad (34)$$

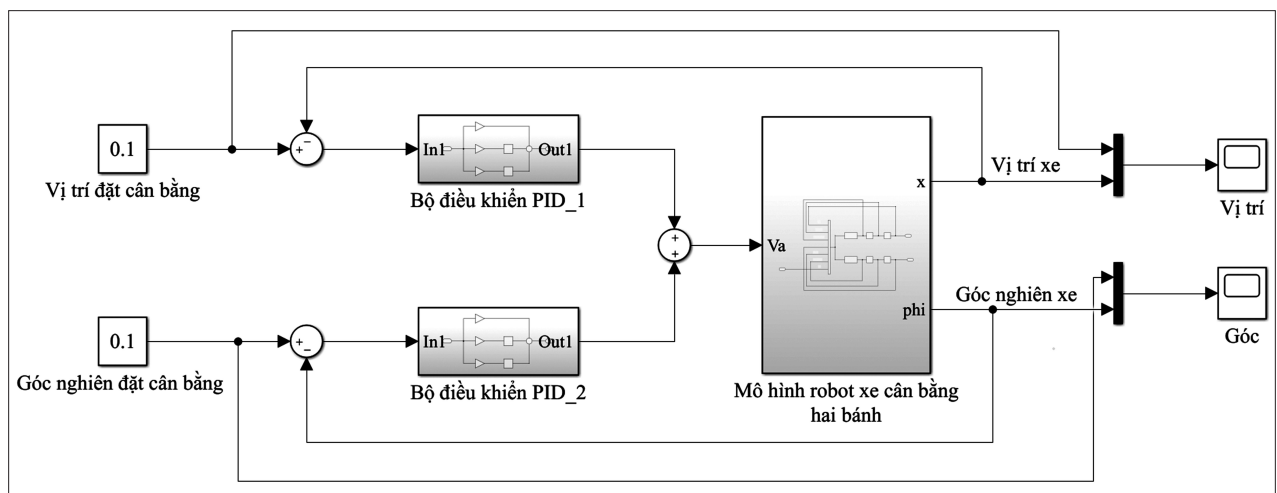
Với:

$$\beta = \left(2M_w + \frac{2I_w}{r} + M_n \right) \quad (35)$$

$$\alpha = \left[I_p \beta + 2M_p l^2 \left(M_w + \frac{I_w}{r^2} \right) \right] \quad (36)$$

Thiết kế ổn định PID

Từ phương trình (32) và (33), ta thiết kế mô hình điều khiển trên Matlab Simulink. Ở hệ thống điều khiển robot xe hai bánh tự cân bằng, ta phải điều khiển ổn định hai tham số là góc nghiêng và vị trí của hệ thống sử dụng bộ điều khiển PID như hình 8.

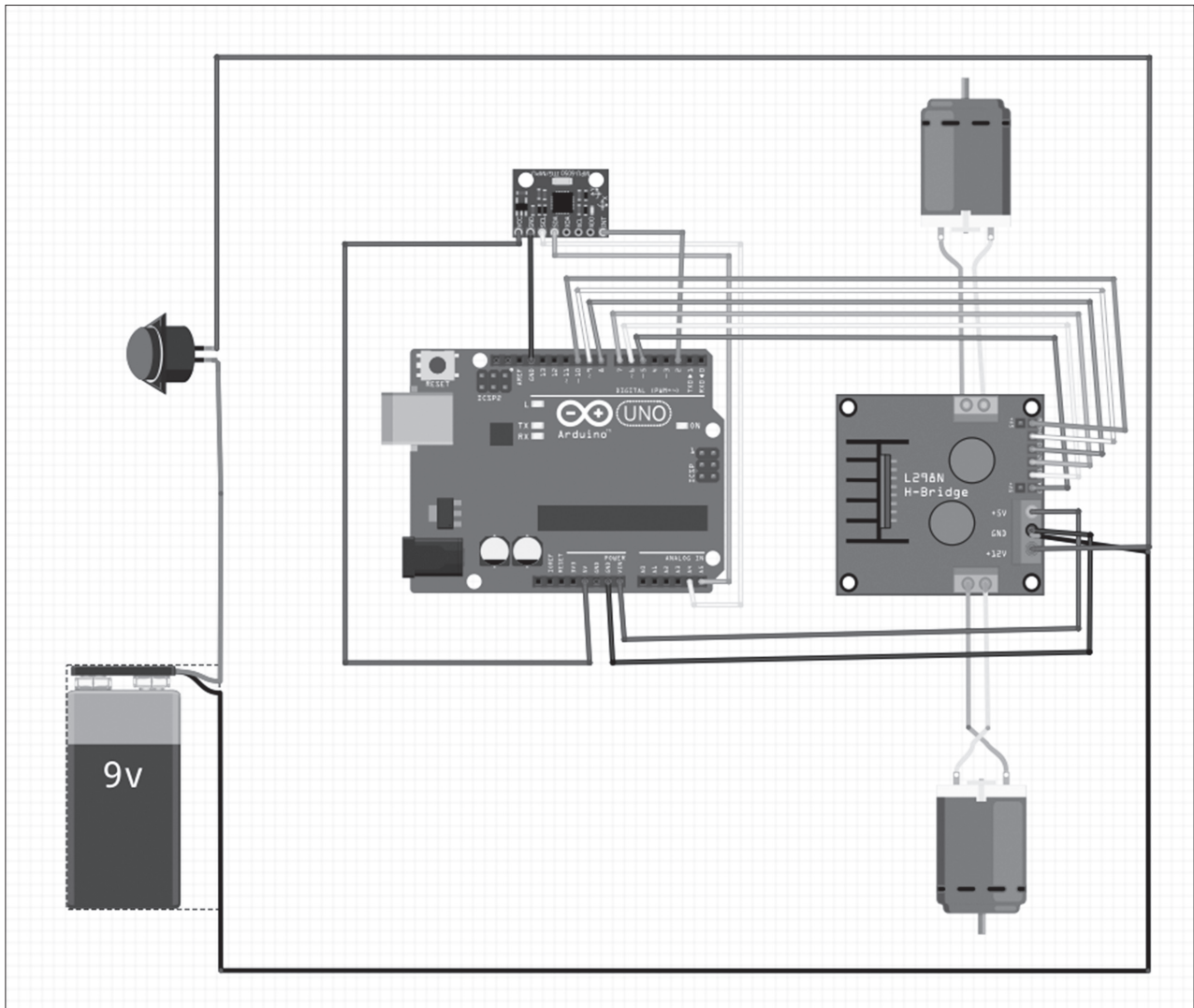


Hình 8. Sơ đồ khối bộ điều khiển vòng kín robot xe cân bằng hai bánh.

Qua thời gian nghiên cứu nhóm tác giả đã sử dụng qua các bộ điều khiển P,PI và PID. Tuy nhiên, bộ điều khiển PID được chọn vì vị trí đặt, thời gian sát lập tương đối tốt. Độ vọt lố không vượt quá 5%.

Sơ đồ phân cứng hệ thống

Sơ đồ mạch điện phân cứng sử dụng bo điều khiển Arduino Uno, mạch cầu H L298N, cảm biến MPU 6050 và động cơ DC JGA25.



Hình 9. Mô hình phân cứng thực nghiệm.

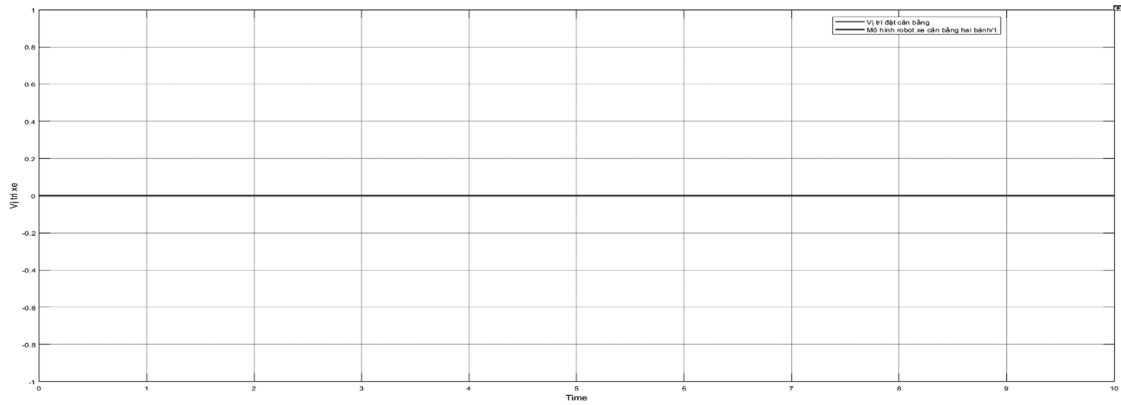
Kết quả và thảo luận

Kết quả

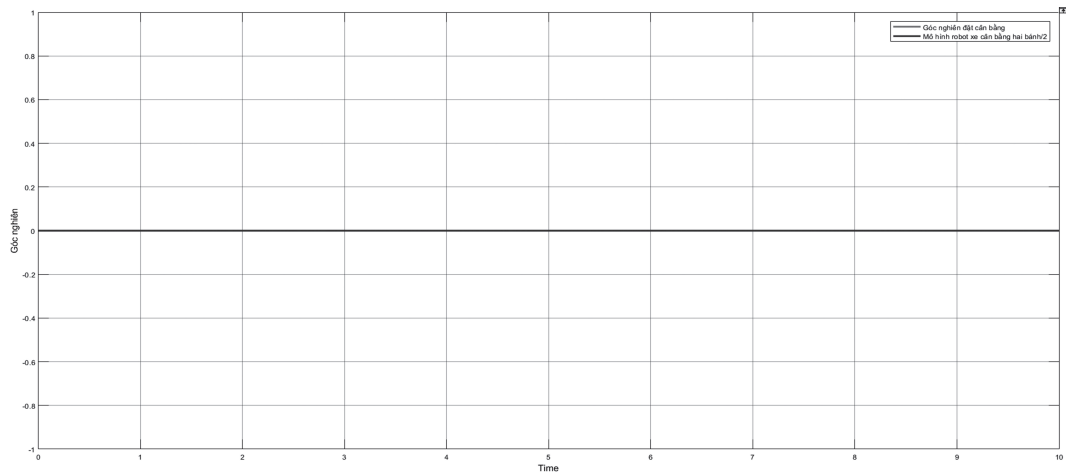
Thông số mô hình được chọn như sau:

$M_p=1$; $M_w=0,03$; $I_p=0,0012$; $I_w=0,000016$; $l=0,05$; $g=9,81$; $km=0,022$; $ke=0,4$; $R=1,7$; $r=0,0325$

Giá trị đặt bằng 0



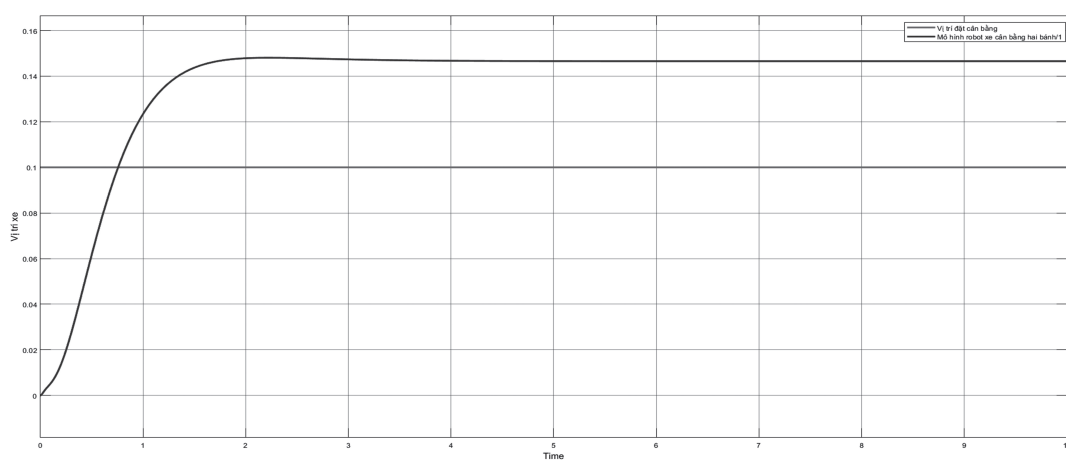
Hình 10. Vị trí xe



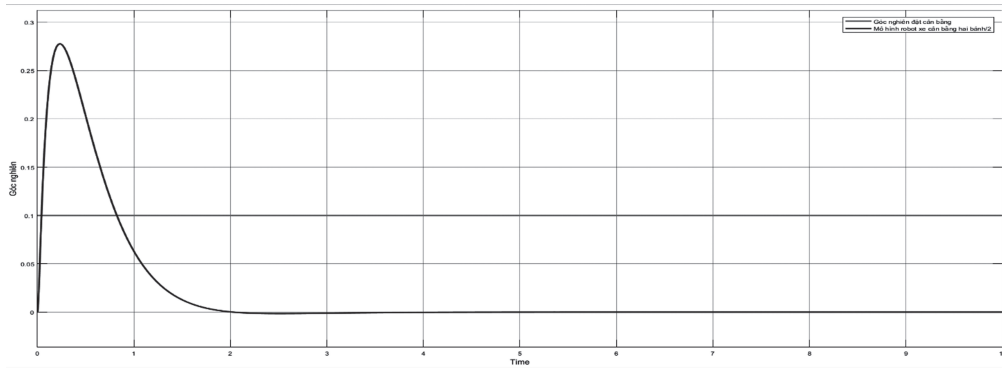
Hình 11. Góc nghiêng xe

Giá trị đặt khác 0

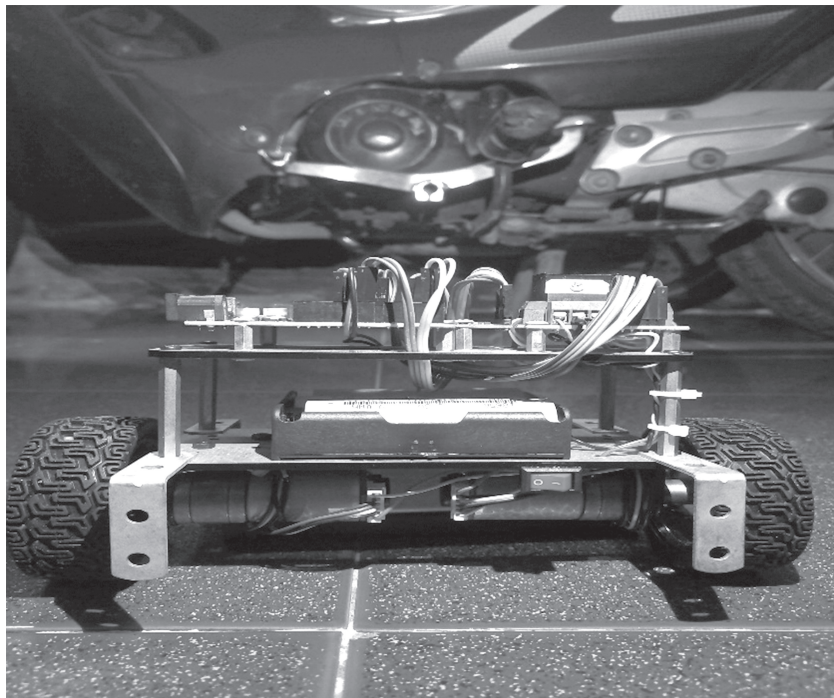
Ta chọn giá trị $\phi_{\text{đặt}} = 0.1$, $x_{\text{đặt}} = 0.1$



Hình 12. Vị trí xe (khi đặt giá trị khác 0)



Hình 13. Góc nghiêng xe (khi đặt giá trị khác 0)



Hình 14. Mô hình thực tế xe hai bánh tự cân bằng

Sau thời gian nghiên cứu nhóm thiết kế được mô hình và xây dựng bộ điều khiển PID vòng kín như hình 8. Với bộ thông số như sau:

$$K_{p1}=200; K_{i1}=290; K_{d1}=12$$

$$K_{p2}=90; K_{i2}=135; K_{d2}=8$$

Như kết quả mô phỏng ở các hình trên hình 13 và hình 14, ta thấy trạng thái xe luôn bám sát tín hiệu đặt, bộ điều khiển PID đáp ứng tốt.

Từ kết quả mô phỏng trên nhóm tác giả sử dụng board Arduino Uno, cảm biến MPU-6050, bộ điều khiển động cơ L298N và các thành phần khác để tạo nên mô hình xe cân bằng. Khi có tác động kích thích nhỏ vào thân xe, xe có thể di chuyển và tự cân bằng. Bộ điều khiển PID được lập trình trong đoạn code như sau:

```

void Pid()
{
error = 180 - CurrentAngle; // 180 = level
pTerm = Kp * error;
    integrated_error += error;
iTerm = Ki*constrain(integrated_error, -GUARD_GAIN, GUARD_GAIN);
dTerm = Kd*(error - last_error);
    last_error = error;
speed = constrain(K*(pTerm + iTerm + dTerm), -255, 255);
}

```

Hình 15. Đoạn code bộ điều khiển PID

Tuy nhiên, khi có lực tác động vào xe khoảng ± 5 độ xe có thể tự cân bằng. Khuyết điểm của mô hình này không thể tự giữ cân bằng khi lực tác động vào làm cho góc nghiêng lớn hơn ± 10 độ. Ngoài ra, hệ thống không thể giữ xe cân bằng ở địa hình gồ ghề và đường dốc có góc lớn hơn ± 10 độ.

Kết luận

Từ kết quả trình bày, nhóm cần tìm hiểu về các bộ điều khiển Fuzzy, bộ điều khiển PID kết hợp với LQR để so sánh các đáp ứng vị trí và góc của xe hai bánh tự cân bằng. Trong tương lai, nhóm tập trung vào nghiên cứu điều khiển thông qua Bluetooth, Wifi,.. Ngoài ra, một hướng phát triển khác là dùng các phương pháp nhận dạng để xác định thông số mô hình động cơ. Từ đó, tính toán giá trị LQR phù hợp nhất để có thể điều khiển xe chạy tới, lùi, quẹo trái, quẹo phải trong thực tế.

Tài liệu tham khảo

- Glad, T., & Ljung, L. (2006). Reglerteknik, grundläggande teori, fjärde upplagan. *Studentlitteratur AB*.
- Hellman, H., & Sunnerman, H. *Two-Wheeled Self-Balancing Robot*.
- Loram, I. D., & Lakie, M. (2002). Human balancing of an inverted pendulum: position control by small, ballistic-like, throw and catch movements. *The Journal of physiology*, 540(3), 1111-1124.
- Ooi, R. C. (2003). Balancing a two-wheeled autonomous robot. *University of Western Australia*, 3.
- Romlay, M., Azhar, M., Toha, S., & Rashid, M. (2019). Two-wheel balancing robot; review on control methods and experiment. *International Journal of Recent Technology and Engineering (IJRTE)*, 7(68), 106-112.
- Shekhawat, A. S., & Rohilla, Y. (2020). *Design and control of two-wheeled self-balancing robot using Arduino*. 2020 International Conference on Smart Electronics and Communication (ICOSEC).