

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM

KHOA CÔNG NGHỆ KỸ THUẬT ĐIỆN – ĐIỆN TỬ



ĐỒ ÁN HỌC PHẦN 2

ĐỀ TÀI: MÔ HÌNH ROBOT HAI BÁNH TỰ CÂN BẰNG

GV Hướng Dẫn: LÊ MINH THANH

SV Thực hiện: HUỲNH BÁ MÃN

MSSV: 2002140327

Lớp: 05DHDT3

TP HCM. Tháng 6 Năm 2017

NHIỆM VỤ ĐỒ ÁN HỌC PHẦN 2

Họ tên sinh viên: Huỳnh Bá Mẫn

MSSV: 2002140327

Khóa: 2014

Lớp: 05DHDT3

I. TÊN ĐỀ TÀI: MÔ HÌNH ROBOT HAI BÁNH TỰ CÂN BẰNG

II. NHIỆM VỤ VÀ NỘI DUNG:

.....

.....

.....

.....

.....

.....

.....

III. NGÀY GIAO NHIỆM VỤ: 10/04/2017

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 16/6/2017

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: ThS. LÊ MINH THANH

LỜI CẢM ƠN



Trong suốt quá trình thực hiện đề tài, mặc dù gặp phải nhiều khó khăn nhưng được sự giúp đỡ, hỗ trợ kịp thời từ quý Thầy Cô và các bạn nên Đồ án đã hoàn thành đúng tiến độ. Em xin chân thành cảm ơn thầy **Lê Minh Thanh** đã tận tình hướng dẫn, chỉ bảo kinh nghiệm quý báu cũng như hỗ trợ phương tiện thí nghiệm trong suốt quá trình tìm hiểu, nghiên cứu đề tài.

Em cũng xin cảm ơn các thành viên trong lớp 05DHDT3 đã có những ý kiến đóng góp, bổ sung, cũng như động viên khích lệ giúp em hoàn thành tốt đề tài.

Mặc dù nhóm thực hiện đề tài đã cố gắng hoàn thiện được đồ án, nhưng trong quá trình soạn thảo cũng như kiến thức còn hạn chế nên có thể còn nhiều thiếu sót. Nhóm thực hiện đề tài mong nhận được sự đóng góp ý kiến của quý thầy cô cùng các bạn sinh viên.

Sau cùng nhóm thực hiện xin chúc Thầy cô sức khỏe, thành công và tiếp tục đào tạo những sinh viên giỏi đóng góp cho đất nước. Chúc các anh (chị), các bạn sức khỏe, học tập thật tốt để không phụ công lao các Thầy Cô đã giảng dạy. Nhóm thực hiện xin chân thành cảm ơn.

Trân trọng

Người thực hiện

Huỳnh Bá Mẫn

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. HCM, ngày ... tháng ... năm 2017

Giáo viên hướng dẫn

ThS. Lê Minh Thanh

Mục lục

Chương 1. Tổng quan về đề tài.....	5
1.1 Đặt vấn đề	5
1.2 Mục tiêu đề tài	7
1.3 Giới hạn đề tài.....	8
1.4 Phương pháp nghiên cứu	8
Chương 2. Các cơ sở lý thuyết.....	9
2.1 Đặc tính động lực học	9
2.1.1 Mô hình hóa robot 2 bánh tự cân bằng trên địa hình phẳng	9
2.1.2 Mô hình hóa robot hai bánh tự cân bằng trên địa hình phẳng trong Matlab Simulink	14
2.2 Bộ lọc Kalman	15
2.2.1 Giới thiệu về bộ lọc Kalman	15
2.2.2 Quá trình ước lượng:.....	17
2.2.3 Bản chất xác suất của bộ lọc	18
2.2.4 Thuật toán Kalman rời rạc	19
2.3 Giải thuật điều khiển	21
2.3.1 Cấu trúc bộ điều khiển PID cho robot hai bánh tự cân bằng	21
2.3.2 Bộ điều khiển PID với thông số cố định	23
2.3.3 Các thành phần chính của mô hình	25
Chương 3. Thiết kế hệ thống	28
3.1 Thiết kế phần cứng mô hình Robot 2 bánh tự cân bằng	28
3.1.1 Thiết kế cơ khí.....	28
3.1.2 Kết nối phần cứng	28
3.2 Thiết kế phần mềm.....	29
3.2.1 Bộ lọc Kalman.....	29
3.2.2 Lưu đồ giải thuật điều khiển	32
3.2.3 Code arduino và Simulink.....	33
Chương 4. Kết luận và hướng phát triển của đề tài	38
4.1 Những kết quả đã đạt được	38
4.2 Hạn chế của đề tài	38
4.3 Kết luận	38

4.4	Hướng phát triển	38
4.5	Tài liệu tham khảo	39

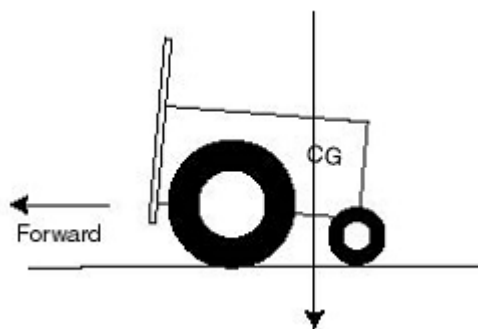
Chương 1. Tổng quan về đề tài

1.1 Đặt vấn đề

Trong ngành tự động hóa - điều khiển tự động nói chung và điều khiển học nói riêng, mô hình con lắc ngược là một trong những đối tượng nghiên cứu điển hình và đặc thù bởi đặc tính động không ổn định của mô hình nên việc điều khiển được đối tượng này trên thực tế đặt ra như một thử thách.

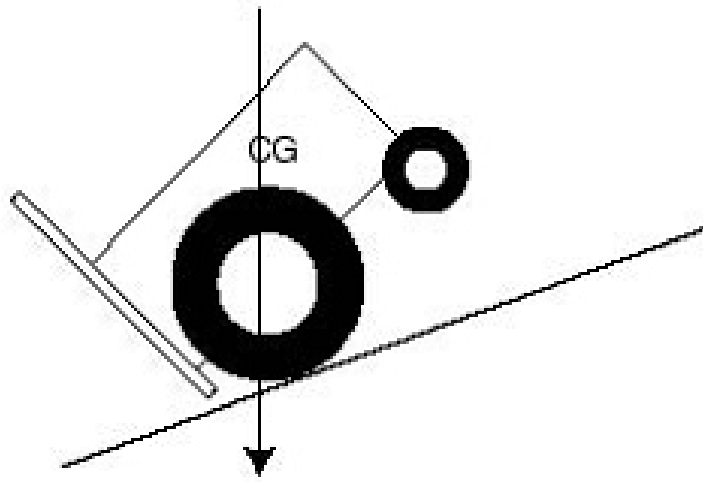
Kết quả nghiên cứu mô hình con lắc ngược cơ bản, ví dụ như mô hình xe-con lắc, con lắc ngược quay... có thể ứng dụng và kế thừa sang các mô hình tương tự khác nhưng có tính ứng dụng thực tiễn hơn, chẳng hạn như mô hình tên lửa, mô hình xe hai bánh tự cân bằng... do đó khắc phục được những nhược điểm vốn có của các robot hai hoặc ba bánh kinh điển. Các robot hai hoặc ba bánh kinh điển, theo đó có cấu tạo gồm bánh dẫn động và bánh tự do (hay bất kì cái gì khác) để đỡ trọng lượng robot. Nếu trọng lượng được đặt nhiều vào bánh lái thì robot sẽ không ổn định và dễ bị ngã, còn nếu đặt vào nhiều bánh đuôi thì hai bánh chính sẽ mất khả năng bám. Nhiều thiết kế robot có thể di chuyển tốt trên địa hình phẳng nhưng không thể di chuyển lên xuống trên địa hình lồi lõm hoặc mặt phẳng nghiêng. Khi di chuyển lên đồi, trọng lượng robot dồn vào đuôi xe làm mất khả năng bám và trượt ngã.

Robot dạng 3 bánh xe di chuyển trên địa hình bằng phẳng trọng lượng được chia đều cho bánh lái và bánh dẫn nhỏ.



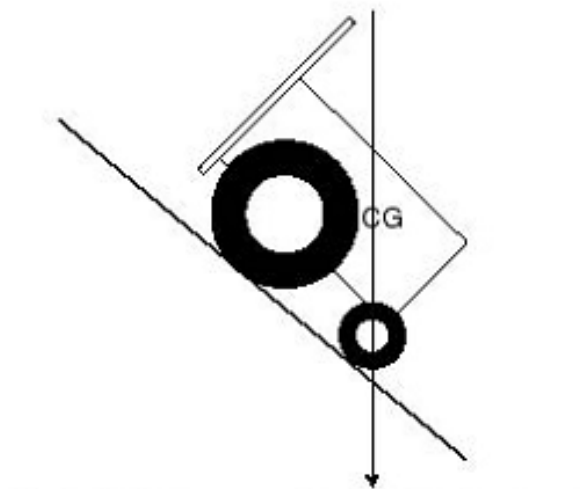
Robot dạng 3 bánh xe di chuyển trên địa hình bằng phẳng

Robot dạng 3 bánh xe khi xuống dốc, trọng lực dồn vào bánh sau khiến xe có thể bị lật úp.



Robot dạng 3 bánh xe khi xuống dốc

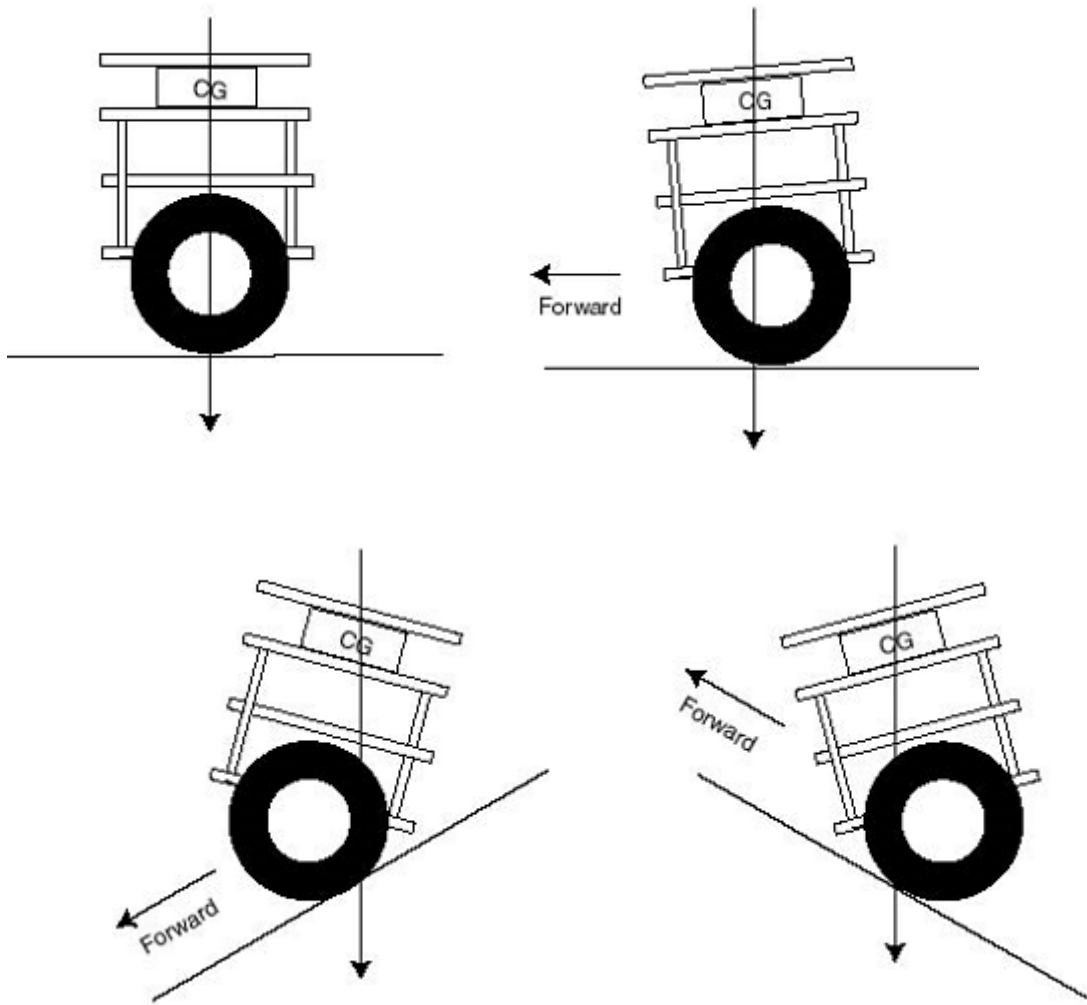
Robot dạng 3 bánh xe khi lên dốc, trọng lượng dồn vào bánh trước khiến lực ma sát giúp xe bám trên mặt đường không được đảm bảo.



Robot dạng 3 bánh xe khi lên dốc

Ngược lại, các robot dạng hai bánh đồng trục lại thăng bằng rất linh động khi di chuyển trên địa hình phức tạp, mặc dù bản thân robot là một hệ thống không ổn định. Khi robot di chuyển trên địa hình dốc, nó tự động nghiêng ra trước và giữ cho trọng lượng dồn về hai bánh chính. Tương tự, khi di chuyển xuống dốc, nó nghiêng ra sau

và giữ trọng tâm rơi vào bánh chính. Vì vậy, không bao giờ có hiện tượng trọng tâm xe rơi ngoài vùng đỡ bánh xe để có thể gây ra lật úp.



Robot 2 bánh di chuyển trên các địa hình khác nhau theo hướng bảo toàn sự thăng bằng

1.2 Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng mô hình robot 2 bánh tự cân bằng dựa trên nền tảng lý thuyết mô hình con lắc ngược. Trong thời gian làm đề tài, những mục tiêu của đề tài được đặt ra như sau:

- Tìm hiểu các mô hình xe, robot 2 bánh tự cân bằng và các nguyên lý cơ bản về cân bằng.
- Tìm hiểu và áp dụng Bộ lọc Kalman để lọc nhiễu cho cảm biến, xây dựng các thuật toán bù trừ để có giá trị góc chính xác.
- Xây dựng thuật toán điều khiển động cơ, giữ thăng bằng cho robot.

1.3 Giới hạn đề tài

Trong khuôn khổ của đề tài này, mô hình robot 2 bánh tự cân bằng chỉ đứng hay chạy tiến lùi mà chưa thể xoay vòng hay đi lên dốc được.

Phương pháp điều khiển: chỉ sử dụng phương pháp điều khiển PID với thông số cố định, chưa ứng dụng phương pháp mạng neuron thích nghi, phương pháp mạng toàn phương tuyến tính (LQR).

1.4 Phương pháp nghiên cứu

Xây dựng mô hình lý thuyết robot 2 bánh tự cân bằng và tính toán các thông số PID.

Xây dựng mô hình thực:

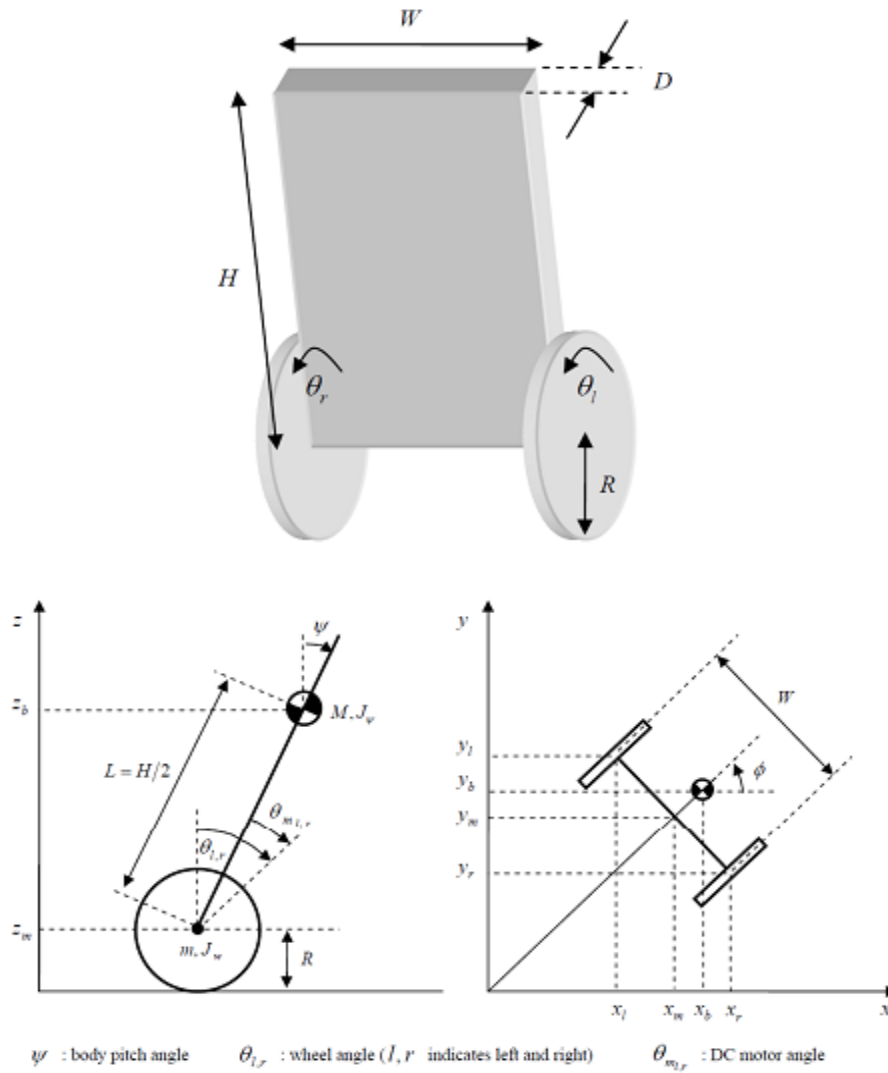
- Thiết kế cơ khí: khung sườn của mô hình.
- Ứng dụng các mạch điện tử, mạch công suất, cảm biến, vi xử lý.
- Mạch cảm biến và bù trừ giá trị cảm biến.

Chương 2. Các cơ sở lý thuyết

2.1 Đặc tính động lực học

2.1.1 Mô hình hóa robot 2 bánh tự cân bằng trên địa hình phẳng

Xây dựng hệ phương trình trạng thái mô tả hệ thống robot hai bánh tự cân bằng.



Mô hình robot 2 bánh tự cân bằng trên mặt phẳng

Trong đề tài này sẽ sử dụng các kí hiệu, đơn vị như sau:

Kí hiệu	Đơn vị	Ý nghĩa
M	kg	Khối lượng của bánh xe
M	kg	Khối lượng của robot
R	m	Bán kính bánh xe
W	m	Chiều rộng của robot
D	m	Chiều ngang của robot
H	m	Chiều cao của robot
L	m	Khoảng cách từ trọng tâm robot đến trục bánh xe
f_w		Hệ số ma sát giữa bánh xe và mặt phẳng di chuyển
f_m		Hệ số ma sát giữa robot và động cơ DC
J_m	$kg.m^2$	Moment quán tính của động cơ DC
R_m	ohm	Điện trở động cơ DC
Kb	V sec/rad	Hệ số EMF của động cơ DC
Kt	Nm/A	Moment xoắn của động cơ DC
N		Tỉ số giảm tốc
G	m/s^2	Gia tốc trọng trường
θ	rad	Góc trung bình của bánh trái và phải
$\theta_{l,r}$	rad	Góc của bánh trái và phải
ψ	rad	Góc nghiêng của phần thân robot
ϕ	rad	Góc xoay của robot
x_l, y_l, z_l	m	Tọa độ bánh trái
x_r, y_r, z_r	m	Tọa độ bánh phải
x_m, y_m, z_m	m	Tọa độ trung bình
F_θ, F_ψ, F_ϕ	Nm	Moment phát động theo các phương khác nhau
$F_{l,r}$	Nm	Moment phát động của động cơ bánh trái, phải
i_l, i_r	A	Dòng điện động cơ bánh trái, phải
v_l, v_r	V	Điện áp động cơ bánh trái, phải

Ký hiệu và ý nghĩa của các đại lượng

Sử dụng phương pháp Euler-Lagrange để xây dựng mô hình động học. Giả sử tại thời điểm $t = 0$, robot di chuyển theo chiều dương trục x , ta có các phương trình sau:

Góc tịnh tiến trung bình của hai bánh xe và góc xoay của robot được xác định như sau:

$$\begin{bmatrix} \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\theta_l + \theta_r) \\ \frac{R}{W}(\theta_l - \theta_r) \end{bmatrix} \quad [2.1]$$

Trong đó tọa độ trung bình của Robot trong hệ qui chiếu:

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} \int \dot{x}_m \\ \int \dot{y}_m \\ R \end{bmatrix} \quad [2.2]$$

$$\text{Và } \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \end{bmatrix} = \begin{bmatrix} R\dot{\theta}\cos\phi \\ R\dot{\theta}\sin\phi \end{bmatrix} \quad [2.3]$$

Tọa độ bánh trái trong hệ qui chiếu :

$$\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \begin{pmatrix} x_m - \frac{W}{2}\sin\phi \\ y_m + \frac{W}{2}\cos\phi \\ z_m \end{pmatrix} \quad [2.4]$$

Tọa độ bánh phải trong hệ qui chiếu :

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{pmatrix} x_m + \frac{W}{2}\sin\phi \\ y_m - \frac{W}{2}\cos\phi \\ z_m \end{pmatrix} \quad [2.5]$$

Tọa độ tâm đối xứng giữa hai động cơ trong hệ qui chiếu :

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} x_m + L\sin\psi\cos\phi \\ y_m L\sin\psi\sin\phi \\ z_m + L\cos\psi \end{bmatrix} \quad [2.6]$$

Phương trình động năng của chuyển động tịnh tiến:

$$T_1 = \frac{1}{2}m(\dot{x}_l^2 + \dot{y}_l^2 + \dot{z}_l^2) + \frac{1}{2}m(\dot{x}_r^2 + \dot{y}_r^2 + \dot{z}_r^2) + \frac{1}{2}m(\dot{x}_b^2 + \dot{y}_b^2 + \dot{z}_b^2) \quad [2.7]$$

Phương trình động năng của chuyển động quay :

$$T_2 = \frac{1}{2}J_w\dot{\theta}_l^2 + \frac{1}{2}J_w\dot{\theta}_r^2 + \frac{1}{2}J_\psi\dot{\psi}^2 + \frac{1}{2}J_\phi\dot{\phi}^2 + \frac{1}{2}n^2J_m(\dot{\theta}_l - \dot{\psi})^2 + \frac{1}{2}n^2J_m(\dot{\theta}_r - \dot{\psi})^2 \quad [2.8]$$

Với

$$\frac{1}{2}n^2J_m(\dot{\theta}_l - \dot{\psi})^2 ; \frac{1}{2}n^2J_m(\dot{\theta}_r - \dot{\psi})^2 \quad [2.9]$$

là động năng quay của phần ứng động cơ trái và phải.

Phương trình thế năng:

$$U = mgz_l + mgz_r + mgz_b \quad [2.10]$$

Phương trình Lagrange :

$$L = T_1 + T_2 - U \quad [2.11]$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = F_\theta \quad [2.12]$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) - \frac{\partial L}{\partial \psi} = F_\psi \quad [2.13]$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \phi} = F_\phi \quad [2.14]$$

Lấy đạo hàm L theo các biến ta được:

$$[(2m + M)R^2 + 2J_w + 2n^2J_m]\ddot{\theta} + (MLR\cos\psi - 2n^2J_m)\ddot{\psi} - MLR\dot{\psi}^2\sin\psi = F_\theta \quad [2.15]$$

$$(MLR\cos\psi - 2n^2J_m)\ddot{\theta} + (ML^2 + J_\psi + 2n^2J_m)\ddot{\psi} - MgL\sin\psi - ML^2\dot{\phi}^2\sin\psi\cos\psi = F_\psi \quad [2.16]$$

$$\left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2}(J_w + n^2J_m) + ML^2\sin^2\psi \right] \ddot{\phi}^2 + 2ML^2\dot{\psi}\dot{\phi}\sin\psi\cos\psi = F_\phi \quad [2.17]$$

Momen động lực do động cơ DC sinh ra:

$$\begin{bmatrix} F_\theta \\ F_\psi \\ F_\phi \end{bmatrix} = \begin{bmatrix} F_l + F_r \\ F_\psi \\ \frac{W}{2R}(F_l - F_r) \end{bmatrix} \quad [2.18]$$

Và:

$$F_l = nK_t i_l + f_m(\dot{\psi} - \dot{\theta}_l) - f_w \dot{\theta}_l \quad [2.19]$$

$$F_r = nK_t i_r + f_m(\dot{\psi} - \dot{\theta}_r) - f_w \dot{\theta}_r \quad [2.20]$$

$$F_\psi = -nK_t i_l - nK_t i_r - f_m(\dot{\psi} - \dot{\theta}_l) - f_m(\dot{\psi} - \dot{\theta}_r) \quad [2.21]$$

Sử dụng phương pháp PWM để điều khiển động cơ nên chuyển từ dòng điện sang điện áp động cơ:

$$L_m \dot{i}_{l,r} = v_{l,r} + K_b(\dot{\psi} - \dot{\theta}_{l,r}) - R_m i_{l,r} \quad [2.22]$$

Xem điện cảm phản ứng tương đối nhỏ (gần bằng 0), có thể bỏ qua, suy ra:

$$i_{l,r} = \frac{v_{l,r} + K_b(\dot{\psi} - \dot{\theta}_{l,r})}{R_m} \quad [2.23]$$

Từ đó, các moment lực sinh ra:

$$F_\theta = \alpha(v_l + v_r) - 2(\beta + f_w)\dot{\theta} + 2\beta\dot{\psi} \quad [2.24]$$

$$F_\psi = -\alpha(v_l + v_r) + 2\beta\dot{\theta} - 2\beta\dot{\psi} \quad [2.25]$$

$$\text{Với } \alpha = \frac{nK_t}{R_m} \text{ và } \beta = \frac{nK_t K_b}{R_m} + f_m \quad [2.26]$$

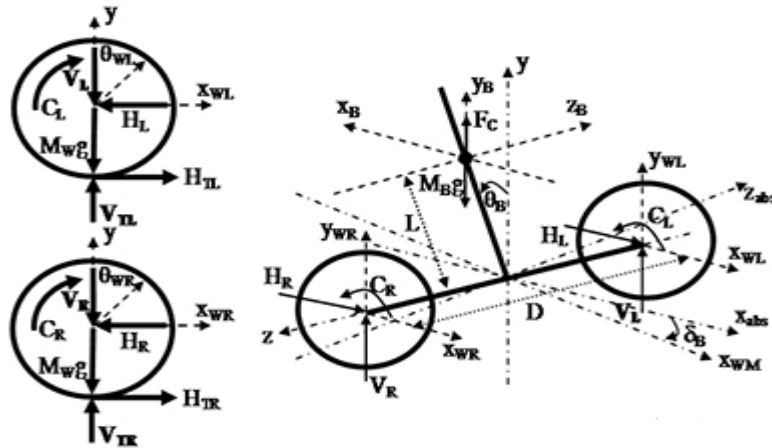
$$F_\phi = \frac{W}{2R} \alpha(v_r - v_l) - \frac{W^2}{2R^2} (\beta + f_w) \dot{\phi} \quad [2.27]$$

Thu được phương trình động lực học mô tả chuyển động của robot như sau:\

$$[(2m + M)R^2 + 2J_w + 2n^2 J_m] \ddot{\theta} + (MLR \cos \psi - 2n^2 J_m) \ddot{\psi} - MLR \dot{\psi}^2 \sin \psi = \alpha(v_l + v_r) - 2(\beta + f_w) \dot{\theta} + 2\beta \dot{\psi} \quad [2.28]$$

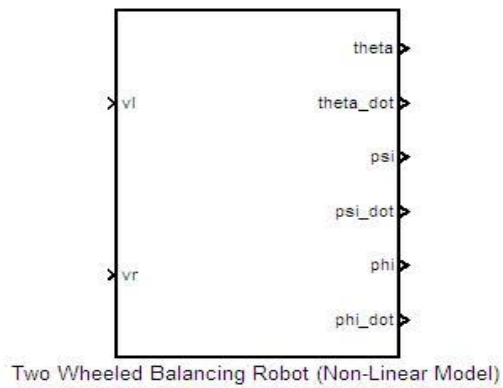
$$(MLR \cos \psi - 2n^2 J_m) \ddot{\theta} + (ML^2 + J_\psi + 2n^2 J_m) \ddot{\psi} - MgL \sin \psi - ML^2 \dot{\phi}^2 \sin \psi \cos \psi = -\alpha(v_l + v_r) + 2\beta \dot{\theta} - 2\beta \dot{\psi} \quad [2.29]$$

$$\left[\frac{1}{2} m W^2 + J_\phi + \frac{W^2}{2R^2} (J_w + n^2 J_m) + ML^2 \sin^2 \psi \right] \ddot{\phi} + 2ML^2 \dot{\psi} \dot{\phi} \sin \psi \cos \psi = \frac{W}{2R} \alpha(v_r - v_l) - \frac{W^2}{2R^2} (\beta + f_w) \dot{\phi} \quad [2.30]$$

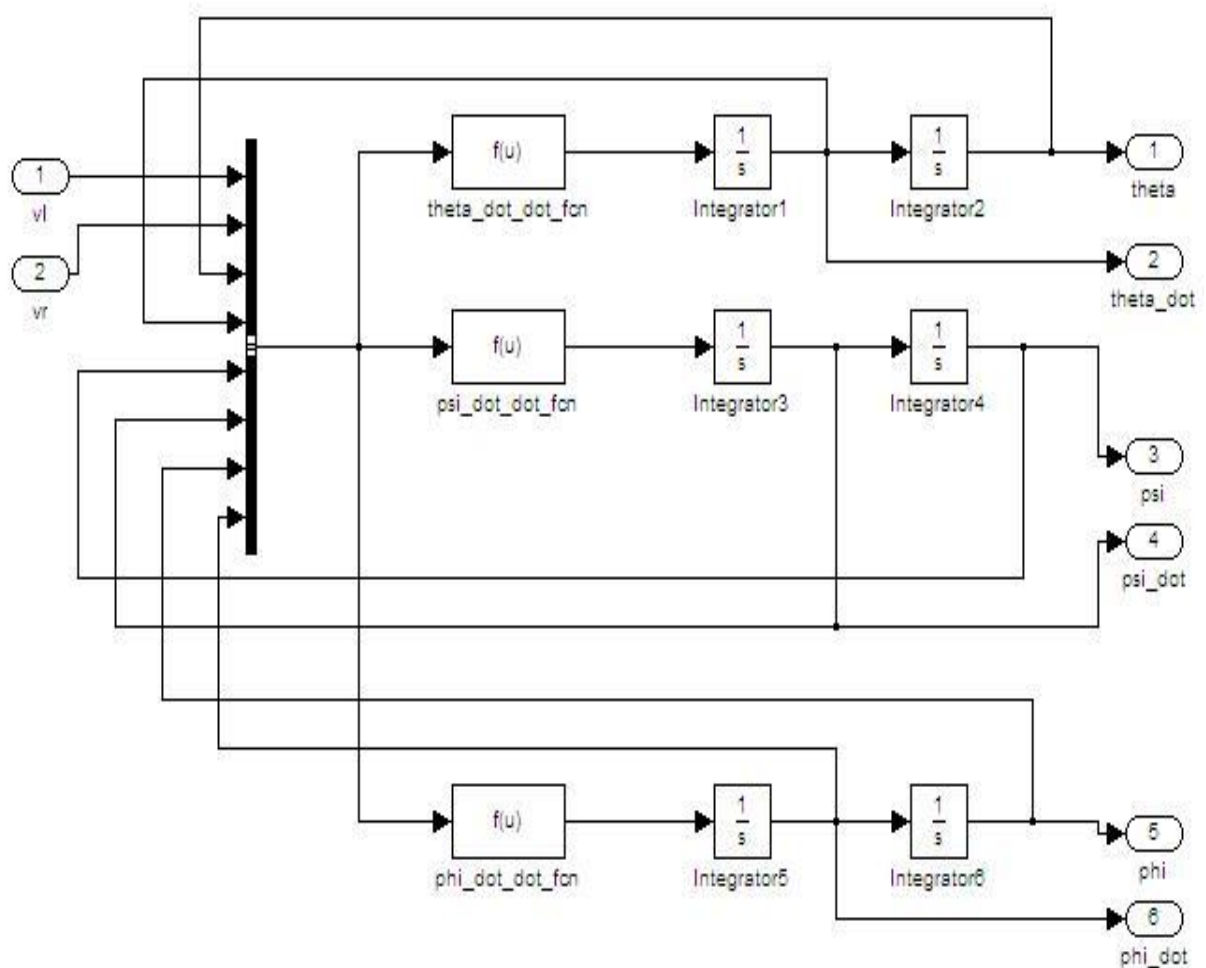


2.1.2 Mô hình hóa robot hai bánh tự cân bằng trên địa hình phẳng trong

Matlab Simulink



Mô hình phi tuyến của robot hai bánh tự cân bằng trong Matlab Simulink

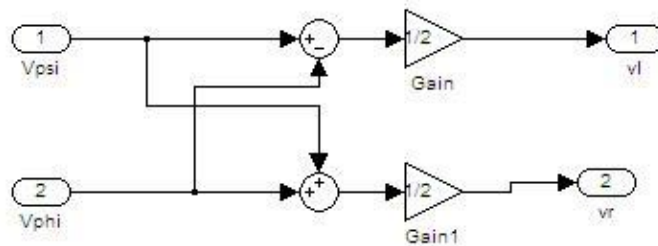
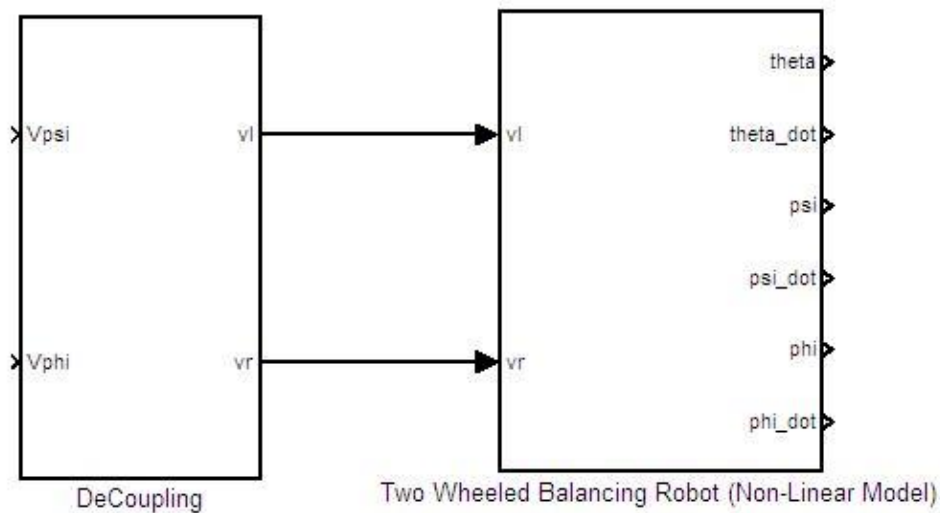


Bên trong khối “Two Wheeled Balancing Robot (Non-Linear Model)”

Phương trình động lực học của robot như trên thể hiện mối quan hệ giữa giá trị điện áp điều khiển hai động cơ, với độ nghiêng, vị trí, vận tốc của hệ robot, giá trị điện áp hai động cơ v_l, v_r tác động lên các thông số đó dưới dạng tổng $v_l + v_r$ còn với góc xoay, giá trị điện áp hai động cơ v_l, v_r tác động lên thông số này dưới dạng hiệu $v_r - v_l$. Khi đó, tách bài toán hệ robot thành hai bài toán nhỏ hơn với hai tín hiệu điều khiển V_ψ, V_ϕ

$$\begin{cases} V_\psi = v_l + v_r \\ V_\phi = v_r - v_l \end{cases} \Rightarrow \begin{cases} v_l = \frac{1}{2}(V_\psi - V_\phi) \\ v_r = \frac{1}{2}(V_\psi + V_\phi) \end{cases} \quad [2.31]$$

Khối thực hiện chức năng này gọi là khối phân tách (decoupling)



Bên trong khối “DeCoupling”

2.2 Bộ lọc Kalman

2.2.1 Giới thiệu về bộ lọc Kalman

Vào năm 1960, R.E.Kalman cho xuất bản nghiên cứu của mình, đưa ra giải pháp đệ quy để rời rạc hóa dữ liệu trong bộ lọc tuyến tính. Kể từ đó, việc giải quyết các bài toán kỹ thuật số, một lĩnh vực rất rộng lớn, đã trở nên dễ dàng hơn rất nhiều. Bộ lọc

Kalman được mở rộng ra nghiên cứu và ứng dụng, đặc biệt là trong lĩnh vực tự động hay hỗ trợ việc tự định vị.

Bộ lọc Kalman thu thập và kết hợp linh động các tín hiệu từ các cảm biến thành phần. Mỗi khi mẫu thống kê nhiều trên các cảm biến được xác nhận, bộ lọc Kalman sẽ cho ước lượng giá trị tối ưu (do đã loại được nhiễu), và có độ phân bố ổn định. Trong đề tài này tín hiệu đưa vào bộ lọc được lấy từ hai cảm biến: cảm biến gia tốc (accelerometers) sẽ cho ra giá trị góc đo và cảm biến vận tốc góc (rate gyro), và tín hiệu ngõ ra của bộ lọc là tín hiệu đã được xử lý lẫn nhau trong bộ lọc; dựa vào mối quan hệ: vận tốc góc = đạo hàm của góc.

Bộ lọc Kalman là thuật toán xử lý dữ liệu hồi quy tối ưu. Nó tối ưu đối với chi tiết cụ thể trong bất kỳ tiêu chuẩn có nghĩa nào. Bộ lọc Kalman tập hợp tất cả thông tin được cung cấp tới nó, xử lý các giá trị sẵn có, ước lượng các giá trị quan tâm, sử dụng các hiểu biết động học, thiết bị giá trị và hệ thống, để mô tả số liệu thống kê của hệ thống nhiều và những thông tin bất kỳ về điều kiện ban đầu của các giá trị cần ước lượng.

Đối với bộ lọc Kalman, thuật ngữ “lọc” không có ý nghĩa như các bộ lọc khác. Đây là một giải thuật tính toán và ước lượng thống kê tối ưu tất cả các thông tin ngõ vào được cung cấp tới nó để có được một giá trị ra đáng tin cậy nhất cho việc xử lý tiếp theo. Do vậy lọc Kalman có thể sử dụng để loại bỏ các tín hiệu nhiễu mà được mô hình là những tín hiệu nhiễu trắng trên tất cả dải thông mà nó nhận được từ ngõ vào, dựa trên các thống kê trước đó và chuẩn trực lại giá trị ước lượng bằng các giá trị đo hiện tại với độ lệch pha gần như không tồn tại và có độ lợi tối thiểu xấp xỉ là 0 đối với những tín hiệu ngõ vào không đáng tin cậy. Mặc dù phải tốn khá nhiều thời gian xử lý lệnh, nhưng với tốc độ hiện tại của các vi điều khiển thời gian thực làm việc tính toán ước lượng tối ưu của bộ lọc này trở nên đơn giản và đáng tin cậy rất nhiều. Nhờ có cơ chế tự cập nhật các giá trị cơ sở (bias) tại mỗi thời điểm tính toán, cũng như xác định sai lệch của kết quả đo trước với kết quả đo sau nên giá trị đo luôn được ổn định, chính xác, gần như không bị sai số về độ lợi và độ lệch pha của các tín hiệu. Hơn thế, được xây dựng bởi hàm trạng thái, do vậy bộ lọc Kalman có thể kết hợp không chỉ hai tín hiệu từ hai cảm biến, mà có thể kết hợp được nhiều cảm biến đo ở những dải tần khác nhau của cùng một giá trị đại lượng vật lý. Chính vì điều này, làm bộ lọc Kalman trở nên phổ dụng hơn tất cả những bộ lọc khác trong việc xử lý tín hiệu chính xác của các cảm biến tọa độ, như cảm biến la bàn, GPS, góc, gyro...

2.2.2 Quá trình ước lượng:

Vấn đề chung của bộ lọc Kalman nhằm ước lượng biến trạng thái $x \in R^n$ của quá trình điều khiển rời rạc được điều chỉnh bởi các phương trình tuyến tính ngẫu nhiên khác nhau. Phương trình không gian trạng thái của bộ lọc:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad [2.32]$$

Với giá trị $z \in R^m$ là:

$$z_k = Hx_k + v_k \quad [2.33]$$

Biến ngẫu nhiên w_k, v_k đặc trưng cho nhiễu quá trình và nhiễu đo của hệ. Chúng độc lập với nhau, tần suất phân bố thông thường:

$$p(w) \sim N(0, Q) \quad [2.34]$$

$$p(v) \sim N(0, R) \quad [2.35]$$

Trên thực tế, ma trận tương quan nhiễu quá trình Q và ma trận tương quan nhiễu đo R có thể thay đổi sau mỗi bước thời gian hay giá trị, tuy nhiên để đơn giản, trong hầu hết các trường hợp Q và R được xem là hằng số.

Ma trận vuông A trong phương trình [2.32] thể hiện mối quan hệ của các biến trạng thái ở thời điểm $k-1$ với thời điểm hiện tại k . Thực ra trên thực tế ma trận A thay đổi sau mỗi bước thời gian, nhưng ở đây ma trận A xem như hằng số. Ma trận B thể hiện mối liên hệ tín hiệu điều khiển $u \in R^L$ đối với biến trạng thái x . Ma trận H trong phương trình [2.33] thể hiện mối liên hệ giữa biến trạng thái với tín hiệu ra z , H cũng được xem là hằng số.

Những tính toán căn bản của bộ lọc:

Định nghĩa:

$\hat{x}_k^- = E\{x_k | y_1, y_2 \dots y_{k-1}\}$ là giá trị ước lượng của x_k trước khi ta xử lý giá trị đo tại thời điểm k .

$\hat{x}_k^+ = E\{x_k | y_1, y_2 \dots y_k\}$ là giá trị ước lượng của x_k sau khi ta xử lý giá trị đo tại thời điểm k .

$\hat{x}_k \in R^n$ là giá trị ước lượng trạng thái sau tại bước k có được sau khi so sánh với giá trị đo z_k . Và chúng ta có sai số ước lượng trạng thái trước và sau:

$$\begin{cases} e_k^- \equiv x_k - \hat{x}_k^- \\ e_k \equiv x_k - \hat{x}_k \end{cases} \quad [2.36]$$

Tương quan sai số ước lượng trước “priori”:

$$P_k^- = E\{e_k^- e_k^{-T}\} \quad [2.37]$$

Tương quan sai số ước lượng sau “posteriori”:

$$P_k = E\{e_k e_k^T\} \quad [2.38]$$

Khi lấy đạo hàm phương trình bộ lọc Kalman, với mục đích tìm một phương trình để tính toán trạng thái ước lượng posteriori \hat{x}_k thể hiện sự tương quan giữa giá trị ước lượng priori \hat{x}_k^- và độ sai lệch giữa giá trị đo thực z_k và giá trị đo ước lượng $H\hat{x}_k^-$:

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad [2.39]$$

Ma trận K trong [3.8] là ma trận độ lợi hay hệ số trộn để tối thiểu hóa phương trình tương quan sai số posteriori. Biểu thức tính K để tối thiểu hóa phương trình [3.8] như sau:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad [2.40]$$

Từ đó thấy rằng tương quan sai số giá trị đo lường R tiến tới 0, khi đó:

$$\lim_{R_k \rightarrow 0} K_k = H^{-1} \quad [2.41]$$

Mặt khác, tương quan sai số ước lượng priori của P_k^- tiến đến 0, khi đó:

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad [3.42]$$

Một cách nghĩ khác về giá trị hiệu chỉnh bù bởi K là nếu ma trận tương quan sai số giá trị đo lường R tiến tới 0 thì giá trị đo được z_k sẽ có độ tin cậy càng cao, trong khi giá trị ước lượng $H\hat{x}_k^-$ sẽ có độ tin cậy càng thấp. Mặt khác, nếu tương quan sai số ước lượng priori P_k^- tiến tới 0 thì z_k sẽ không đáng tin mà giá trị ước lượng $H\hat{x}_k^-$ sẽ càng đáng tin.

2.2.3 Bản chất xác suất của bộ lọc

Sự điều chỉnh cho x_k trong [3.8] đã xác định bản chất ước lượng priori \hat{x}_k^- với điều kiện tất cả các giá trị đo z_k đều có nghĩa (Luật phân bố Bayer). Điều đó cho thấy bộ lọc Kalman duy trì hai thời điểm đầu tiên của sự phân bố trạng thái:

$$E[x_k] = \hat{x}_k \quad [2.43]$$

$$E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = P_k \quad [2.44]$$

Phương trình ước lượng trạng thái posteriori phản ánh giá trị trung bình của phân

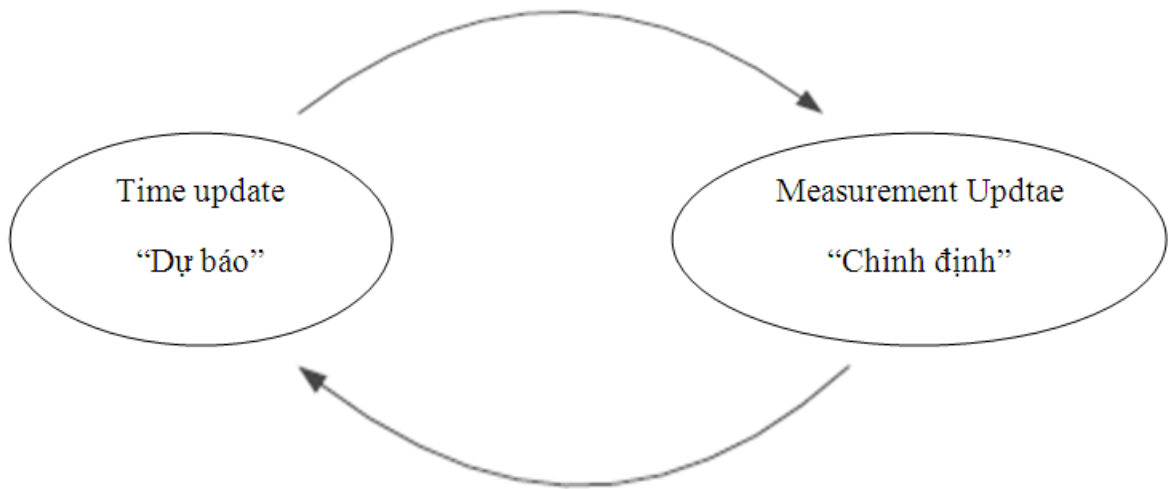
bố trạng thái. Tương quan sai số ước lượng trạng thái posteriori phản ánh sự thay đổi của phân bố trạng thái. Ngoài ra ta còn có:

$$p(x_k | z_k) \sim N(E[x_k], E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]) = N(\hat{x}_k, P_k) \quad [2.45]$$

2.2.4 Thuật toán Kalman rời rạc

Bộ lọc Kalman ước lượng tiến trình bằng cách sử dụng dạng điều khiển hồi tiếp: bộ lọc ước lượng các trạng thái của quá trình tại một vài thời điểm và sau đó chứa tín hiệu hồi tiếp trong các dạng của giá trị đo lường. Do đó, phương trình bộ lọc Kalman chia làm hai nhóm: phương trình cập nhật thời gian và phương trình cập nhật giá trị đo lường. Phương trình cập nhật thời gian chịu trách nhiệm cho việc dự báo trước (về mặt thời gian) của trạng thái hiện tại và ước lượng sai số tương quan để chứa vào bộ ước lượng trước priori cho bước thời gian tiếp theo. Phương trình cập nhật giá trị đo lường chịu trách nhiệm cập nhật cho tín hiệu hồi tiếp, nghĩa là cập nhật giá trị mới vào giá trị ước lượng trước priori để tạo tín hiệu ước lượng sau posteriori tốt hơn.

Phương trình cập nhật thời gian cũng có thể được coi là phương trình dự đoán. Trong khi đó phương trình cập nhật giá trị đo lường thì được xem như là phương trình hiệu chỉnh. Vì vậy, thuật toán ước lượng cuối cùng đều giống nhau ở thuật toán dự đoán và hiệu chỉnh để giải quyết vấn đề số học như hình vẽ dưới đây:



Quy trình thực hiện của bộ lọc Kalman

Phương trình cập nhật thời gian cho bộ lọc Kalman rời rạc:

$$\hat{x}_k^- = A\hat{x}_{k-1}^- + Bu_{k-1} \quad [2.46]$$

$$P_k^- = AP_{k-1}A^T + Q \quad [2.47]$$

Phương trình cập nhật giá trị đo lường cho bộ lọc Kalman rời rạc:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad [2.48]$$

$$\hat{x}_k = \hat{x}_k^- K(z_k - H\hat{x}_k^-) \quad [2.49]$$

$$P_k = (1 - K_k H)P_k^- \quad [2.50]$$

Nhiệm vụ đầu tiên trong suốt quá trình cập nhật giá trị đo lường là tính toán độ lợi Kalman K_k . Bước tiếp theo là xử lý giá trị đo thực được chứa trong z_k . Sau đó, tính trạng thái ước lượng sau posteriori bằng cách kết hợp giá trị đo được theo công thức \hat{x}_k ở trên. Bước cuối cùng là tính giá trị sai số ước lượng tương quan posteriori vào P_k . Sau mỗi chu trình tính toán của bộ lọc Kalman, các giá trị được cập nhật theo cặp, tiến trình được lặp lại với ước lượng posteriori của trạng thái trước dùng để dự đoán ước lượng priori mới. Trạng thái đệ quy tự nhiên là một trong những điểm đặc trưng của bộ lọc Kalman, nó thay thế điều kiện đệ quy ước lượng hiện tại cho giá trị đã qua.

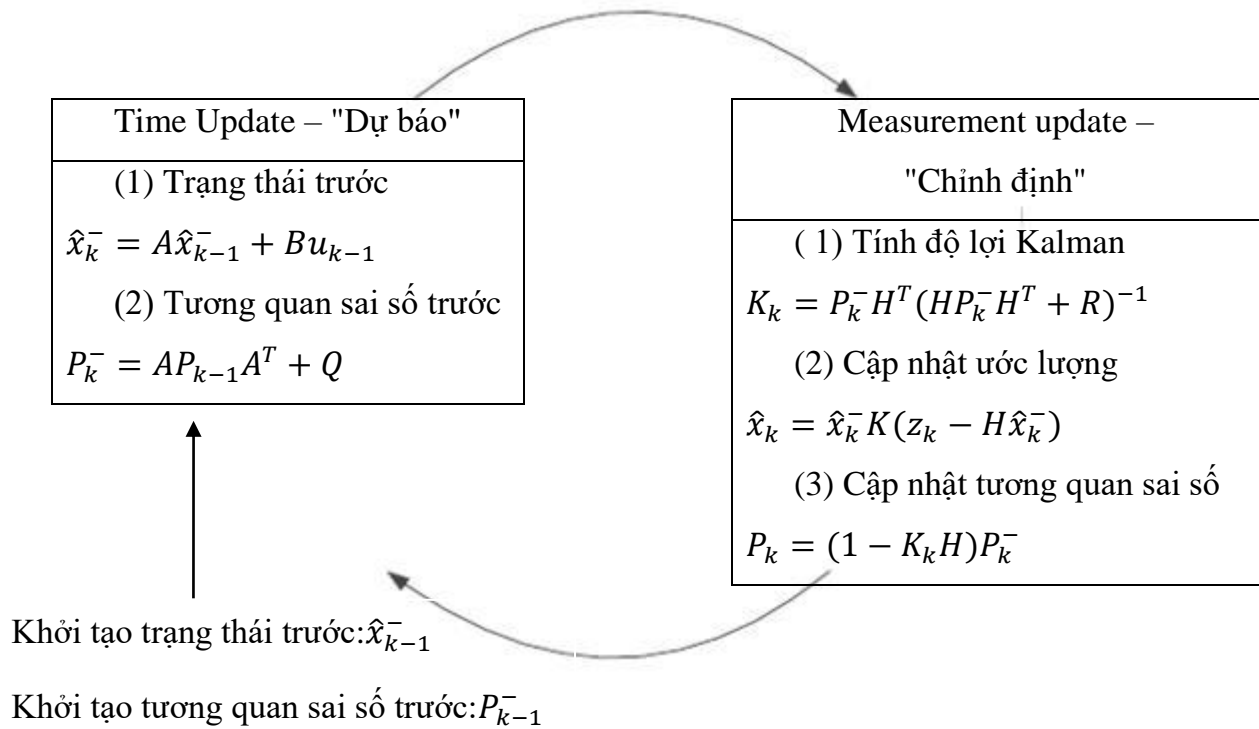
Trong điều kiện thực hiện thực tế của bộ lọc, giá trị nhiễu tương quan R thường được dùng làm giá trị ưu tiên để tính toán cho bộ lọc. Trên thực tế, việc đo các giá trị ma trận R là rất phổ biến bởi vì chúng ta có thể đo quy trình theo nhiều cách vì vậy mà thường lấy mẫu giá trị để đưa ra khuynh hướng thay đổi của giá trị nhiễu.

Sự xác định rõ tương quan nhiễu quá trình Q thường rất khó bởi vì điều điển hình là chúng ta không có khả năng quan sát trực tiếp tiến trình mà chúng ta đang ước lượng. Đôi khi sự liên hệ tới những quy trình mẫu đơn giản có thể đưa ra những giá trị chấp nhận được nếu một mẫu xen vào không chắc chắn đủ với tiến trình thông qua sự lựa chọn Q . Chắc chắn trong trường hợp này, mẫu đó sẽ hi vọng rằng giá trị tiến trình là đáng tin cậy.

Trong những trường hợp khác, dù muốn hay không chúng ta đều có cái chuẩn hợp lý cho việc lựa các thông số, thường thì chất lượng bộ lọc sẽ tốt hơn nhiều lần khi có chứa sự hiệu chỉnh các tham số Q và R . Sự hiệu chỉnh thường được thực hiện gián tiếp, thường thì với sự giúp đỡ của một bộ lọc Kalman khác trong quy trình chung, liên hệ như một hệ thống đồng nhất.

Với điều kiện Q và R là các hằng số thực, cả hai cho phép ước lượng sai số tương quan P_k và độ lợi Kalman K_k sẽ ổn định nhanh chóng và sau đó trở thành hằng số.

Trong điều kiện luận văn, thông số Q và R được hiệu chỉnh dựa vào quá trình thử sai để dự đoán khuynh hướng hiệu chỉnh của hệ thống và tìm ra bộ thông số phù hợp nhất.



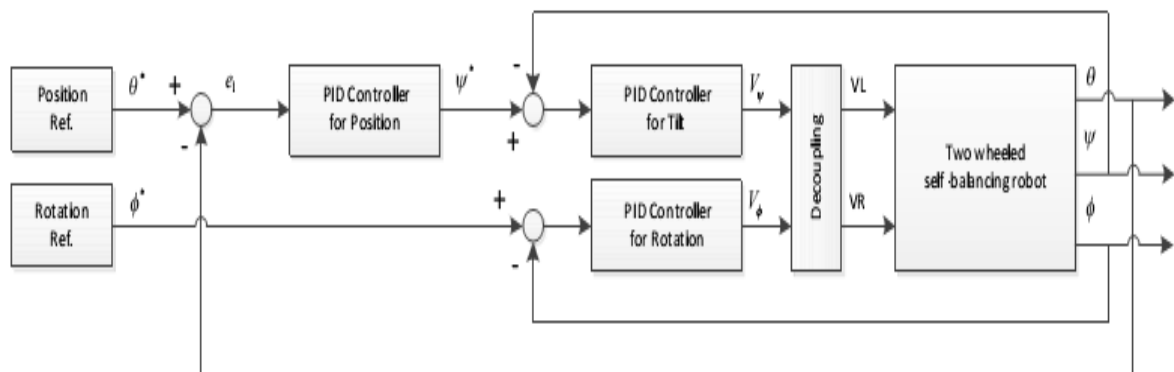
Tổng quan chu trình thực hiện bộ lọc Kalman hoàn chỉnh

2.3 Giải thuật điều khiển

2.3.1 Cấu trúc bộ điều khiển PID cho robot hai bánh tự cân bằng

Ba bộ PID được sử dụng để điều khiển robot hai bánh tự cân bằng, bao gồm:

- Bộ PID điều khiển góc nghiêng (ψ)
- Bộ PID điều khiển vị trí (θ)
- Bộ PID điều khiển góc xoay (ϕ)



Cấu trúc bộ điều khiển PID cho hệ robot hai bánh tự cân bằng

Hàm truyền đạt bộ điều khiển PID liên tục:

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D \left(\frac{s}{1+\tau s} \right) \quad [2.51]$$

Rời rạc hóa đạo hàm theo thời gian:

$$y(k) \approx \frac{f(k) - f(k-1)}{T_s} \quad [2.52]$$

Rời rạc hóa tích phân theo thời gian:

$$\int_0^{k.T_s} y(t)dt = y(k) \approx y(k-1) + \frac{f(k) - f(k-1)}{2} \cdot T_s \quad [2.53]$$

Phép biến đổi rời rạc (z-Transform)

$$X[z] = \mathcal{Z}\{x[k]\} = \sum_{k=0}^{\infty} x[k]z^{-k} \quad [2.54]$$

$$z = Ae^{j\theta} = A(\cos\theta + j\sin\theta) \quad [2.55]$$

$$\text{Ta có } \mathcal{Z}\{x[k-n]\} = z^{-n}X[z] \text{ và } \mathcal{Z}\{x[k]\} = X[z] \quad [2.56]$$

Do đó

$$y(k) = \frac{f(k) - f(k-1)}{T_s} \xrightarrow{z} Y[z] = \frac{F[z] - z^{-1}F[z]}{T_s}$$

$$\Rightarrow \frac{Y[z]}{F[z]} = \frac{z-1}{zT_s} \quad [2.57]$$

Và

$$y(k) = y(k-1) + \frac{f(k) - f(k-1)}{2} \cdot T_s \xrightarrow{z} Y[z](1 - z^{-1}) = \frac{T_s}{2} F[z](1 + z^{-1})$$

$$\Rightarrow \frac{Y[z]}{F[z]} = \frac{T_s}{2} \frac{z+1}{z-1} \quad [2.58]$$

$$\frac{U[z]}{E[z]} = K_p + K_i \frac{T_s}{2} \frac{z+1}{z-1} + K_d \frac{z-1}{zT_s} \quad [2.59]$$

$$\Leftrightarrow \frac{U[z]}{E[z]} = \frac{K_p(z^2 - z) + K_i \frac{T_s}{2}(z^2 + z) + \frac{K_d}{T_s}(z^2 - 2z + 1)}{z^2 - z}$$

$$\Leftrightarrow \frac{U[z]}{E[z]} = \frac{\left(K_p + K_i \frac{T_s}{2} + \frac{K_d}{T_s}\right) z^2 + (-K_p + K_i \frac{T_s}{2} + \frac{2K_d}{T_s}) z + \frac{K_d}{T_s}}{z^2 - z}$$

$$\Leftrightarrow \frac{U[z]}{E[z]} = \frac{\left(K_p + K_i \frac{T_s}{2} + \frac{K_d}{T_s}\right) + (-K_p + K_i \frac{T_s}{2} + \frac{2K_d}{T_s}) z^{-1} + \frac{K_d}{T_s} z^{-2}}{1 - z^{-1}}$$

$$\Leftrightarrow U[z] = z^{-1}U[z] + aE[z] + bz^{-1}E[z] + cz^{-2}E[z] \quad [2.60]$$

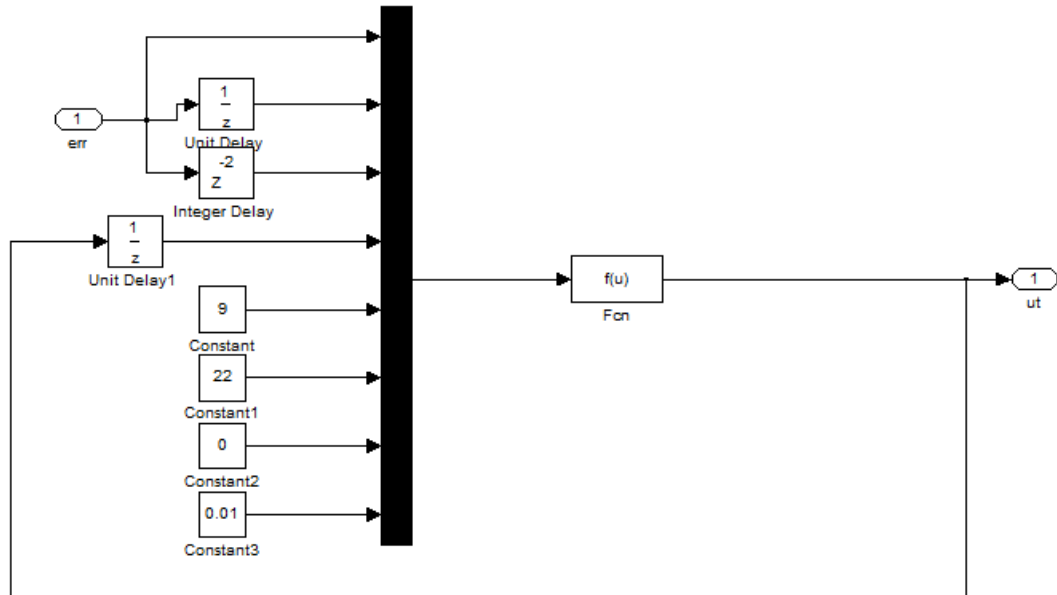
$$\Leftrightarrow u[k] = u[k-1] + ae[k] + be[k-1] + ce[k-2] \quad [2.61]$$

$$\text{Trong đó: } a = K_p + K_i \frac{T_s}{2} + \frac{K_d}{T_s} ; b = -K_p + K_i \frac{T_s}{2} + \frac{2K_d}{T_s} ; c = \frac{K_d}{T_s} \quad [2.62]$$

2.3.2 Bộ điều khiển PID với thông số cố định

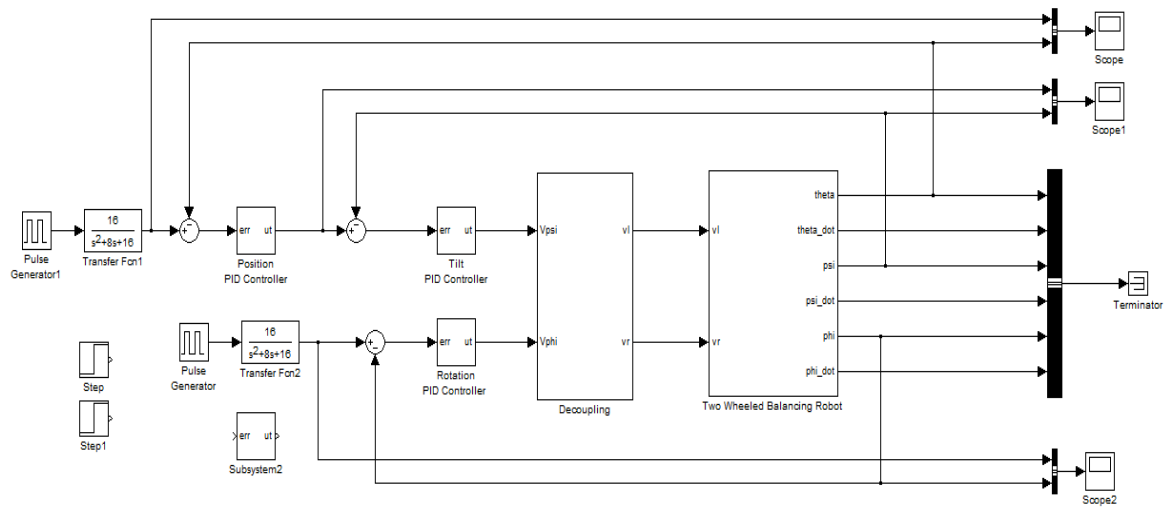
Bộ điều khiển PID với thông số KP, KI, KD cố định như trên, hệ thống chỉ làm việc tốt trong điều kiện hệ số KP, KI, KD đã được chỉnh định tối ưu và trong quá trình làm việc, các thông số trong mô hình không đổi.

Cấu trúc bộ điều khiển PID rời rạc với hệ số cố định sử dụng để điều khiển góc nghiêng, vị trí và góc xoay là như nhau và được hiện thực như sau trong Matlab Simulink.



Cấu trúc bên trong bộ điều khiển PID rời rạc với thông số cố định.

Hệ thống điều khiển robot hai bánh tự cân bằng sử dụng bộ PID rời rạc với thông số cố định hiện thực như sau:



Robot hai bánh tự cân bằng sử dụng 3 bộ điều khiển PID cố định.

Trong đó, mô hình Robot 2 bánh tự cân bằng trong trường hợp này sẽ được thay đổi các thông số như khối lượng thân Robot(M) và hệ số ma sát giữa bánh xe với bề mặt di chuyển(fw) thông qua file “Two_wheelRobot.m”

Có thể thấy, đáp ứng ngõ ra vị trí của hệ thống bị ảnh hưởng một cách rõ rệt, ngõ ra vị trí của hệ thống không thể tiếp tục đáp ứng theo tín hiệu đặt như ban đầu khi hệ số fw thay đổi. Còn các bộ PID điều khiển góc nghiêng thì vẫn đảm bảo ngõ ra hệ thống bám tốt theo tín hiệu đặt.

Đặc biệt khi có sự thay đổi về khối lượng Robot (M), ngõ ra vị trí của hệ thống bị vọt lố khá lớn. Tuy nhiên khả năng điều khiển góc nghiêng của bộ PID vẫn duy trì và bám khá tốt với tín hiệu đặt.

Như vậy trong trường hợp có sự thay đổi hệ số ma sát fw và khối lượng Robot M trong mô hình thì bộ PID cố định điều khiển vị trí không thể đảm bảo được chất lượng điều khiển. Tuy nhiên khâu góc nghiêng và góc xoay đáp ứng khá tốt.

2.3.3 Các thành phần chính của mô hình

2.3.3.1 Nguồn

Nguồn cung cấp cho robot sử dụng pin Lithium-polimer 11.1V/1100mA



Pin Li-Po 11.1V/1100mA

2.3.3.2 Arduino Mega 2560



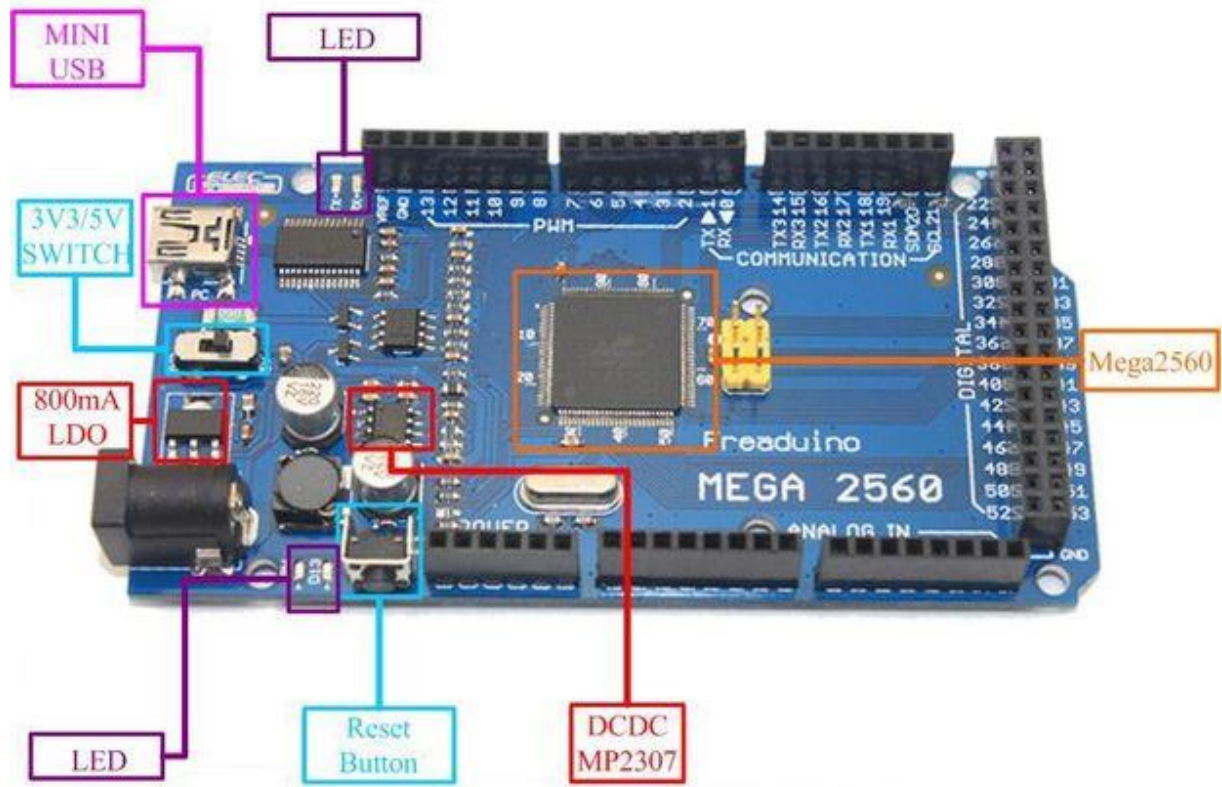
Arduino Mega2560 là một vi điều khiển bằng cách sử dụng ATmega2560.

Bao gồm:

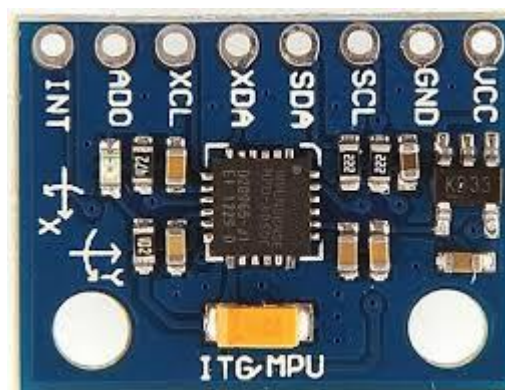
- 54 chân digital (15 có thể được sử dụng như các chân PWM)
- 16 đầu vào analog,
- 4 UARTs (cổng nối tiếp phân cứng),
- 1 thạch anh 16 MHz,
- 1 cổng kết nối USB,
- 1 jack cắm điện,
- 1 đầu ICSP,
- 1 nút reset.

Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển.

Arduino Mega2560 khác với tất cả các vi xử lý trước giờ vì không sử dụng FTDI chip điều khiển chuyển tín hiệu từ USB để xử lý. Thay vào đó, nó sử dụng ATmega16U2 lập trình như là một công cụ chuyển đổi tín hiệu từ USB. Ngoài ra, Arduino Mega2560 cơ bản vẫn giống Arduino Uno R3, chỉ khác số lượng chân và nhiều tính năng mạnh mẽ hơn, nên các bạn vẫn có thể lập trình cho con vi điều khiển này bằng chương trình lập trình cho Arduino Uno R3.



2.3.3.3 Cảm biến gia tốc góc nghiêng MPU6050 GY521



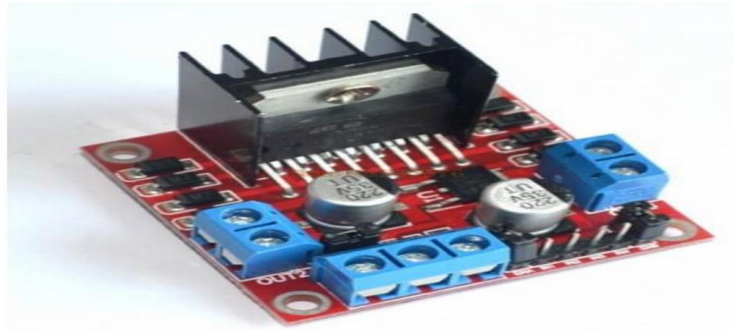
Thông số kỹ thuật:

- MPU-6050 module (3 trục góc + 3 trục gia tốc)
- Chip : MPU-6050
- Nguồn cấp : 3-5V
- Chuẩn giao tiếp: I2C

- Chip 16bit AD converter, 16-bit data Output
- Độ phân giải góc : $\pm 250 \ 500 \ 1000 \ 2000 \text{ }^\circ/\text{s}$
- Độ phân giải gia tốc : $\pm 2 \pm 4 \pm 8 \pm 16\text{g}$
- Chuẩn giắc cắm 2.54mm
- Chuẩn giao tiếp I2C.

2.3.3.4 Module điều khiển động cơ L298N

Mạch điều khiển động cơ L298 giúp bạn có thể điều khiển tốc độ và chiều quay của động cơ DC một cách dễ dàng, ngoài ra module L298 còn điều khiển được 1 động cơ bước lưỡng cực. Mạch cầu H L298 động cơ có điện áp từ 5V đến 35V. Trên mạch cầu H L298 có tích hợp một IC nguồn 7805 để tạo ra nguồn 5V để cung cấp cho các thiết bị khác.



2.3.3.5 Motor 25GA370

Động cơ: 25GA370

Ứng dụng: Thuyền, Xe hơi, Xe đạp điện, Quạt, Đồ gia dụng, khác...

Điện áp: 12V

Tốc độ không tải: 320 rpm

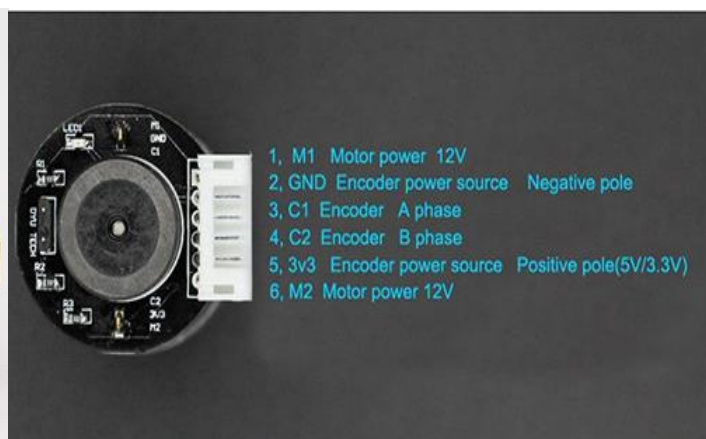
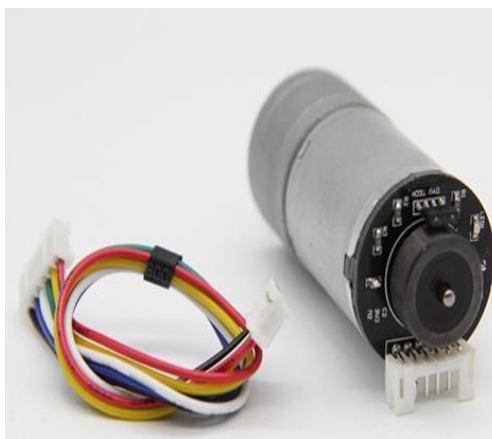
Dòng : 0.08A

Công suất: 5.53W

Tốc độ: 284 rpm

Dòng: 0.6A

Mô-men xoắn cực đại: 1.88kg. Cm

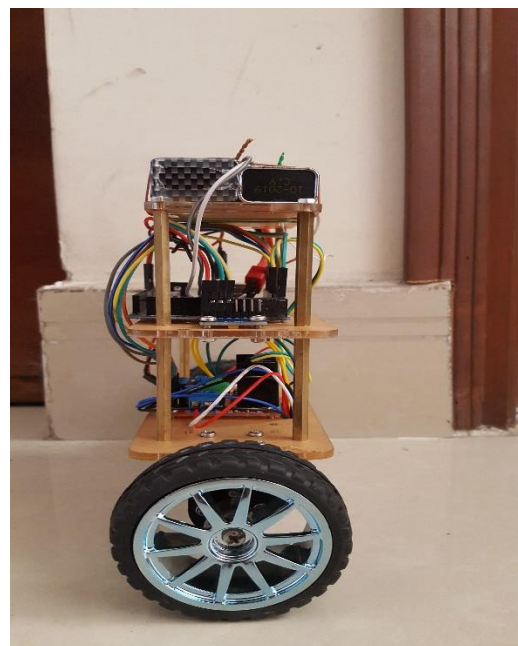
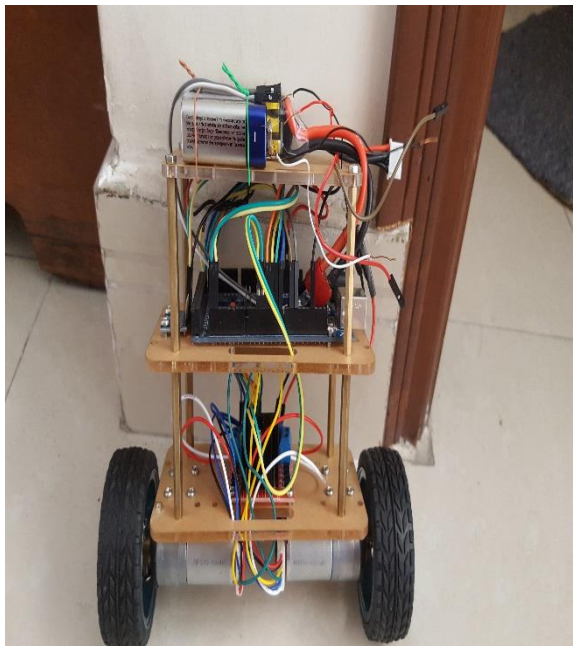


Chương 3. Thiết kế hệ thống

3.1 Thiết kế phần cứng mô hình Robot 2 bánh tự cân bằng

3.1.1 Thiết kế cơ khí

Khung sườn sử dụng chất liệu mica trong và mica màu 4mm, được cố định theo cơ cấu ghép rãnh. Cơ cấu chuyển động encoder áp bánh và bố trí mạch theo chiều dựng đứng.



Mô hình Robot thực tế

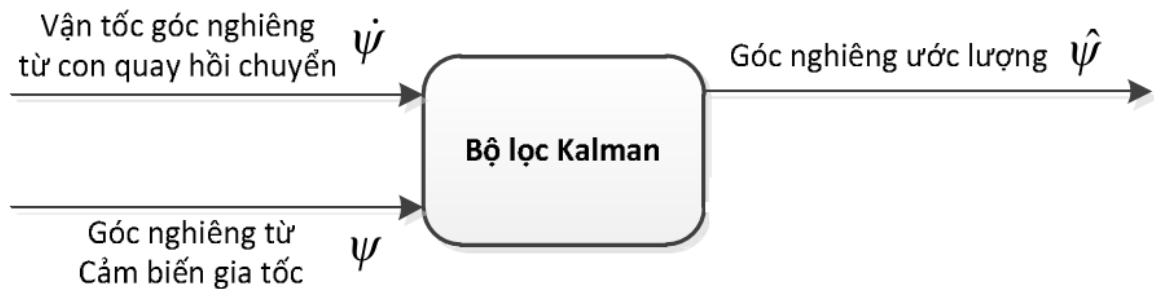
3.1.2 Kết nối phần cứng

Board Arduino			
Mega 2560	Chức năng	Kết nối	Ghi chú
Chân 2	Input	Encode motor	
Chân 3	Input	Encode motor	
Chân 4	Output	Chân input L298N – IN1	
Chân 5	Output	Chân input L298N – IN2	
Chân 6	Output	Chân input L298N – IN3	Output L298N nối motor
Chân 7	Output	Chân input L298N – IN4	
Chân 9	Output	Chân EA - L298N	
Chân 10	Output	Chân EB - L298N	
Chân 20	Input	Chân SDA cảm biến Gyro	Giao tiếp chuẩn I2C
Chân 21	Input	Chân SCL cảm biến Gyro	

3.2 Thiết kế phần mềm

3.2.1 Bộ lọc Kalman

Robot sử dụng hai cảm biến: cảm biến gia tốc (accelecometer) và cảm biến con quay hồi chuyển(gyroscope) để đo góc nghiêng và vận tốc góc nghiêng. Tuy nhiên, vấn đề đặt ra là cần phải kết hợp thông tin từ hai cảm biến để xác định chính xác góc nghiêng thực của hệ robot loại bỏ được ảnh hưởng của nhiễu đo và nhiễu quá trình. Để giải quyết vấn đề này, giải thuật lọc Kalman được sử dụng, với mục đích ước lượng giá trị góc nghiêng của hệ robot từ hai loại cảm biến trên và loại bỏ được ảnh hưởng của nhiễu. Bộ lọc Kalman được khảo sát với mô hình 3 biến trạng thái như sau:



Hình 3.5: Mô hình bộ lọc Kalman với 3 biến trạng thái.

Với mô hình này, bộ lọc sử dụng 2 biến ngõ vào là vận tốc góc nghiêng từ cảm biến con quay hồi chuyển và góc nghiêng từ cảm biến gia tốc; 1 biến ngõ ra là góc nghiêng ước lượng

Ma trận $P_{khoitao}$ đặc trưng cho tương quan sai số:

$$\text{Với } P_{khoitao} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad [3.1]$$

angle, q_bias lần lượt là góc, vận tốc sử dụng trong tính toán của bộ lọc.

R tượng trưng cho giá trị nhiễu covariance. Trong trường hợp này, nó là ma trận 1x1 được mong đợi có giá trị 0.08 rad $\approx 4,5$ độ từ cảm biến gia tốc:

$$R_angle=0.08 \quad [3.2]$$

Q là ma trận 2x2 tượng trưng cho tiến trình nhiễu covariance. Trong trường hợp này, nó chỉ mức độ tin cậy của cảm biến gia tốc quan hệ với cảm biến gyro:

$$Q = \begin{bmatrix} q_angle & 0 \\ 0 & q_gyro \end{bmatrix} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.003 \end{bmatrix} \quad [3.3]$$

- “Giai đoạn dự báo”

Trong mỗi khoảng chu kỳ quét dt với giá trị cơ sở gyro được canh chỉnh tùy theo nhu cầu sử dụng và cách lắp cơ khí bởi người dùng module IMU. Giai đoạn sẽ cập nhật góc hiện thời và vận tốc ước lượng.

$$\text{Vecto giá trị: } X = \begin{bmatrix} angle \\ gyro_bias \end{bmatrix} \quad [3.4]$$

Nó chạy trên sự ước lượng giá trị qua hàm giá trị:

$$\dot{X} = \begin{bmatrix} \dot{angle} \\ \dot{gyro_bias} \end{bmatrix} = \begin{bmatrix} gyro - gyro_bias \\ 0 \end{bmatrix} \quad [3.5]$$

Và cập nhật ma trận covariance qua hàm :

$$\dot{P} = AP + PA' + Q \quad [3.6]$$

$$\dot{P} = A.P + P.A' + Q \text{ lý thuyết là } (P = A.P.A^T + Q) \quad [3.7]$$

A là Jacobian của \dot{X} với giá trị mong đợi :

$$A = \begin{bmatrix} \frac{d(angle)}{d(angle)} & \frac{d(angle)}{d(gyro_bias)} \\ \frac{d(gyro_bias)}{d(angle)} & \frac{d(gyro_bias)}{d(gyro_bias)} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \quad [3.8]$$

Để dễ dàng trong việc lập trình, ma trận P sẽ được khai triển đến mức tối thiểu :

$$\dot{P} = \begin{bmatrix} Q_angle - P[0][1] - P[1][0] & -P[1][1] \\ -P[1][1] & Q_gyro \end{bmatrix} \quad [3.9]$$

$$= \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} + \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} + \begin{bmatrix} Q_angle & 0 \\ 0 & Q_gyro \end{bmatrix}$$

$$= \begin{bmatrix} -P[1][0] & -P[1][1] \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -P[0][1] & 0 \\ -P[1][1] & 0 \end{bmatrix} + \begin{bmatrix} Q_angle & 0 \\ 0 & Q_gyro \end{bmatrix} \quad [3.10]$$

Lưu trữ giá trị ước lượng chưa bias của gyro :

$$rate = q = q_m - q_bias \quad [3.11]$$

(Với rate là sai số góc tiên đoán)

Cập nhật ước lượng góc :

$$angle = angle + \dot{angle} dt = angle + rate dt \quad [3.12]$$

Cập nhật ma trận covariance :

$$P = P + \dot{P}dt$$

$$= \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} + \begin{bmatrix} Q_{angle} - P[0][1] - P[1][0] & -P[1][1] \\ -P[1][1] & Q_{gyro} \end{bmatrix} dt \quad [3.13]$$

- “Giai đoạn cập nhật giá trị bộ lọc Kalman”

Ma trận C là ma trận 1x2 (giá trị x trạng thái), đó là ma trận Jacobian của giá trị đo lường với giá trị mong đợi. Trong trường hợp này C là :

$$C = \begin{bmatrix} \frac{d(angle_m)}{d(angle)} & \frac{d(angle_m)}{d(gyro_bias)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \text{ (C chính là H)} \quad [3.14]$$

Vì giá trị góc đáp ứng trực tiếp với góc ước lượng và giá trị góc không quan hệ với giá trị gyro bias nên C_0 cho thấy giá trị trạng thái quan hệ trực tiếp với trạng thái ước lượng như thế nào, C_1 cho thấy giá trị trạng thái không quan hệ với giá trị cơ sở gyro ước lượng

Error là giá trị khác nhau trong giá trị đo lường và giá trị ước lượng. Trong trường hợp này, nó khác nhau giữa hai gia tốc kế đo góc và góc ước lượng.

$$angle_{error} = angle_m - angle \quad [3.15]$$

Tính sai số ước lượng. Từ bộ lọc Kalman :

$$E = CPC^T + R = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R = P[0][0] + R \quad [3.16]$$

$$PCt_0 = C[0] \times P[0][0] \quad [3.17]$$

$$PCt_1 = C[0] \times P[1][0] \quad [3.18]$$

Ước tính bộ lọc Kalman đạt được. từ lý thuyết bộ lọc Kalman :

$$\begin{bmatrix} K0 \\ K1 \end{bmatrix} = \begin{bmatrix} PCt_0/E \\ PCt_1/E \end{bmatrix} \quad [3.19]$$

$$P = (1 - KH)P = P - KHP = P - KCP \quad [3.20]$$

Ta có phép nhân điểm trôi (floating point)

$$CP = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} = \begin{bmatrix} P[0][0] & P[0][1] \end{bmatrix} = \begin{bmatrix} t_0 & t_1 \end{bmatrix} \quad [3.21]$$

$$P = \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} - \begin{bmatrix} K0 \\ K1 \end{bmatrix} \begin{bmatrix} t_0 & t_1 \end{bmatrix} = P - \begin{bmatrix} K0.t_0 & K0.t_1 \\ K1.t_0 & K1.t_1 \end{bmatrix} \quad [3.22]$$

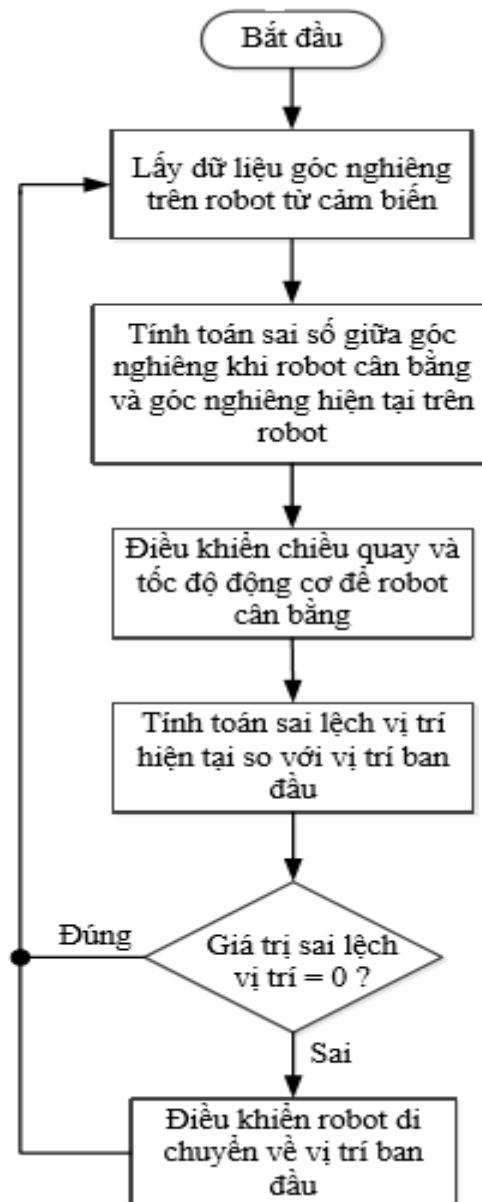
Cập nhật giá trị ước lượng. Lần nữa, từ Kalman :

$$X = \begin{bmatrix} angle \\ gyro_bias \end{bmatrix} = X + K(Z_{measure} - HX_{estimate}) = X + K \begin{bmatrix} angle_err \\ angle_err \end{bmatrix} \quad [3.23]$$

$$angle_err = q_bias_err \quad [3.24]$$

3.2.2 Lưu đồ giải thuật điều khiển

Dựa trên nền tảng của thuật toán điều khiển PID, giải thuật cân bằng và điều khiển bám theo vị trí ban đầu trên mô hình robot được xây dựng theo lưu đồ sau:



3.2.3 Code arduino và Simulink

3.2.3.1 Code arduino

```
#include <Kalman.h>
#include<Servo.h>
#include<Wire.h>

Kalman kalmanX;
//IMU 6050=====
int16_t accX, accY, accZ;
int16_t tempRaw;
int16_t gyroX, gyroY, gyroZ;
float accXangle;
float gyroXangel;
float kalAngelX;
unsigned long timer;
uint8_t i2cData[14];
float CurrentAngle;
// MOTOR=====
int AIN1 = 4;
int AIN2 = 5;
int BIN1 = 6;
int BIN2 = 7;
int CIN1 = 9;
int CIN2 = 10;
int speed;
// PID=====
const float Kp = 10;
const float Ki = 0;
const float Kd = 2;
float pTerm, iTerm, dTerm, integrated_error, last_error, error;
const float K = 1.9*1.12;
#define GUARD_GAIN 10.0
#define runEvery(t) for (static typeof(t) _lasttime;(typeof(t))((typeof(t))millis() - _lasttime) > (t);_lasttime += (t))

void setup()
{
  pinMode(AIN1, OUTPUT);
  pinMode(AIN2, OUTPUT);
  pinMode(BIN1, OUTPUT);
  pinMode(BIN2, OUTPUT);
  Serial.begin(9600);
  Wire.begin();
```

```

i2cData[0] = 7; // Set the sample rate to 1000Hz - 8kHz/(7+1) =
1000Hz
i2cData[1] = 0x00; // Disable FSYNC and set 260 Hz Acc
filtering, 256 Hz Gyro filtering, 8 KHz sampling
i2cData[2] = 0x00;
i2cData[3] = 0x00;
while(i2cWrite(0x19,i2cData,4,false));
while(i2cWrite(0x6B,0x01,true));
while(i2cRead(0x75,i2cData,1));
if(i2cData[0] != 0x68) { // Read "WHO_AM_I" register
Serial.print(F("Error reading sensor"));
while(1);
}
delay(100);

//Kalman=====
while(i2cRead(0x3B,i2cData,6));
accX = ((i2cData[0] << 8) | i2cData[1]);
accY = ((i2cData[2] << 8) | i2cData[3]);
accZ = ((i2cData[4] << 8) | i2cData[5]);
accXangle = (atan2(accY,accZ)+PI)*RAD_TO_DEG;
kalmanX.setAngle(accXangle);
gyroXangel = accXangle;
timer = micros();
}

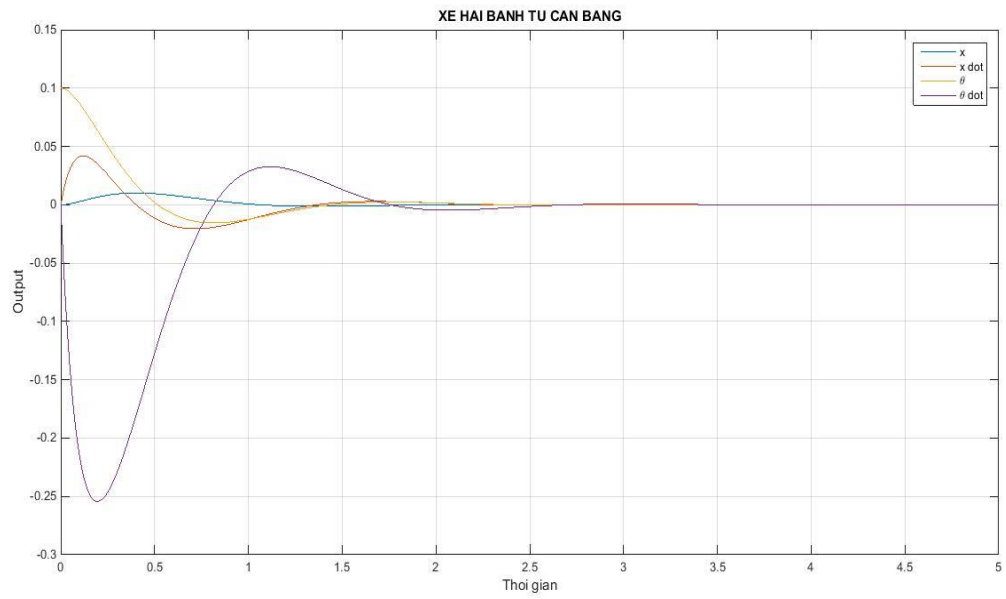
void loop()
{
Serial.println(accX);
// Serial.println(accY);
// Serial.println(accZ);
Serial.println(accXangle);
Serial.println(CurrentAngle);
runEvery(25)
{
dof();
if(CurrentAngle <=179 && CurrentAngle >=178.5)
{
stop();
}
else
{
if(CurrentAngle < 230 && CurrentAngle >130)
{
Pid();
Motors();
}
}
}
}

```

```

else
    {
stop();
    }
    }
}
void Motors()
{
if(speed > 0)
{
analogWrite(CIN1, speed);
digitalWrite(AIN1, LOW);
digitalWrite(AIN2, HIGH);
analogWrite(CIN2, speed);
digitalWrite(BIN1, LOW);
digitalWrite(BIN2, HIGH);
}
else
{
speed = map(speed,0,-255,0,255);
analogWrite(CIN1, speed);
digitalWrite(AIN1, HIGH);
digitalWrite(AIN2, LOW);
analogWrite(CIN2, speed);
digitalWrite(BIN1, HIGH);
digitalWrite(BIN2, LOW);
}
}
void stop()
{
speed = map(speed,0,-150,0,150);
analogWrite(CIN1, speed);
digitalWrite(AIN1, LOW);
digitalWrite(AIN2, HIGH);
analogWrite(CIN2, speed);
digitalWrite(BIN1, LOW);
digitalWrite(BIN2, HIGH);
}
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);

```

Tín hiệu đầu ra

Chương 4. **Kết luận và hướng phát triển của đề tài**

4.1 Những kết quả đã đạt được

- Thiết kế hoàn chỉnh, hoàn thiện mô hình robot 2 bánh tự cân bằng
- Thiết lập mô hình toán học, hàm trạng thái của mô hình và mô phỏng thành công mô hình trên Matlab Simulink.
- Các thành phần mạch điện và vi điều khiển hoạt động tốt:
 - Giá trị cảm biến đọc về chính xác, không bị trễ, sai sót, mất frame truyền.
 - Mạch cầu H hoạt động ổn định không bị quá tải, quá nhiệt.
 - Đọc encoder chính xác, không bị sai lệch
 - Vi điều khiển hoạt động ổn định, không tự reset hoặc ngắt kết nối
- Mô hình có thể giữ thăng bằng tại chỗ.

4.2 Hạn chế của đề tài

Giữ cân bằng chưa tốt, dễ bị vọt lố.

Chịu tác động của ngoại lực còn yếu.

Hệ thống chưa đáp ứng được khi hệ số ma sát và khối lượng M của robot thay đổi.

Chỉ có thể giữ thăng bằng tại chỗ mà chưa di chuyển được.

4.3 Kết luận

Mô hình robot 2 bánh tự cân bằng là một đề tài không mới nhưng đặt ra nhiều thách thức trong nghiên cứu và chế tạo. Để có thể thiết kế một mô hình robot 2 bánh tự cân bằng hoàn chỉnh, linh hoạt hơn và làm cơ sở để chế tạo xe 2 bánh tự cân bằng chở được người thì cần phải tính toán thật chính xác về kết cấu cơ khí, cũng như phải có được các thông số thật chính xác về động cơ, encoder, cảm biến và 1 bộ điều khiển có thể thích nghi được như mạng neuron...

4.4 Hướng phát triển

Trong tương lai mô hình robot 2 bánh tự cân bằng sẽ được phát triển như sau:

- Mô hình sẽ được điều khiển thông qua máy tính nhờ kết nối bluetooth và giao diện trên máy tính sẽ thân thiện trực quan hơn.
- Tối ưu hóa khối lượng, thuật toán để mô hình có thể hoạt động trơn tru, linh hoạt, dễ điều khiển, tiết kiệm năng lượng.
- Có thể gắn thêm camera và gps để robot có thể định vị, ghi hình, xử lý ảnh và tự hoạt động trong không gian lớn.
- Mô hình robot 2 bánh tự cân bằng là tiền đề để phát triển thành xe 2 bánh tự cân bằng có thể chở được người và di chuyển linh hoạt hơn

4.5 Tài liệu tham khảo

- Tài liệu trong nước:

Huỳnh Thái Hoàng, Lý thuyết điều khiển tự động, Nhà xuất bản Đại học Quốc Gia, 2005

Huỳnh Thái Hoàng; “ Mô Hình Hóa Và Nhận Dạng Hệ Thống”; Đại học bách khoa Tp. HCM.

Nguyễn Phùng Quang; “ Matlab và Simulink”; NXB khoa học và kỹ thuật.

Dương Hoài Nghĩa; “Điều khiển hệ thống đa biến”; Nhà xuất bản Đại học Quốc Gia TP.HCM, 2007.

Huỳnh Thái Hoàng; “ Mô Hình Hóa Và Nhận Dạng Hệ Thống”; Đại học bách khoa Tp. HCM.

- Tài liệu nước ngoài:

Greg WELCH and Gary BISHOP , *An Introduction to the Kalman Filter*, University of North Carolina at Chapel Hill, 2004

- Tài liệu website:

<http://www.segway.com>

<http://sourceforge.net>

<http://www.ti.com>

<http://www.mathworks.com/matlabcentral/answers/>