



# Xecanbang - điều khiển xe hai bánh tự cân bằng với thuật toán PID

Kỹ thuật đo lường (Trường Đại học Bách Khoa - Đại học Đà Nẵng)

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN**



**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ ĐIỆN TỰ ĐỘNG**

**ĐỀ TÀI:**

**MÔ HÌNH ROBOT HAI BÁNH**  
**TỰ CÂN BẰNG**

**GVHD : TS. NGUYỄN MINH TÂM**

**SVTH:** Cù Minh Phước  
Nguyễn Khắc Mẫn

**MSSV:** 09118042  
09118032

Tp. Hồ Chí Minh 07/2013

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

Cán bộ hướng dẫn: TS. Nguyễn Minh Tâm

Sinh viên thực hiện: Cù Minh Phước  
Nguyễn Khắc Mẫn

Cán bộ chấm phản biện: .....

Đồ án tốt nghiệp được bảo vệ tại HỘI ĐỒNG CHẤM BẢO VỆ ĐỒ ÁN TỐT NGHIỆP TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH.

Ngày ... tháng 07 năm 2013

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT

TP. HỒ CHÍ MINH

**KHOA ĐIỆN - ĐIỆN TỬ**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**ĐỘC LẬP - TỰ DO - HẠNH PHÚC**

Tp. HCM, ngày ... tháng 7 năm 2013

## **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Cù Minh Phước

MSSV: 09118042

Nguyễn Khắc Mẫn

MSSV: 09118032

Chuyên ngành: Điện tự động

Mã ngành: 18

Hệ đào tạo: Đại học chính quy

Mã hệ: 1

Khóa: 2009

Lớp: 091180

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

**II. NHIỆM VỤ VÀ NỘI DUNG:**

.....

.....

.....

.....

.....  
.....  
.....  
III. NGÀY GIAO NHIỆM VỤ: 10/04/2013

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 15/7/2013

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: TS. Nguyễn Minh Tâm

CÁN BỘ HƯỚNG DẪN

BM. TỰ ĐỘNG ĐIỀU KHIỂN

TS. NGUYỄN MINH TÂM

TS. NGUYỄN MINH TÂM

## 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 tốt nghiệp đã hoàn thành đúng tiến độ. Nhóm thực hiện xin chân thành cảm ơn thầy **Nguyễn Minh Tâm** 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.

Đồng thời Nhóm thực hiện cũng xin chân thành cảm ơn các Thầy Cô trong **Khoa Điện - Điện Tử** đã tạo điều kiện, cung cấp cho nhóm những kiến thức cơ bản, cần thiết để chúng em có điều kiện và đủ kiến thức để thực hiện quá trình nghiên cứu.

Bên cạnh đó, nhóm cũng xin cảm ơn các thành viên trong lớp 091180 đã có những ý kiến đóng góp, bổ sung, cũng như động viên khích lệ giúp nhóm hoàn thành tốt đề tài.

Ngoài ra, nhóm cũng đã nhận được sự chỉ bảo của các anh (chị) đi trước. Các anh (chị) cũng đã hướng dẫn và giới thiệu tài liệu tham khảo thêm trong việc thực hiện nghiên cứu.

Mặc dù nhóm thực hiện đề tài đã cố gắng hoàn thiện được quyền 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**

Nhóm thực hiện đồ án

Cù Minh Phước

Nguyễn Khắc Mẫn

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

# Giáo viên phản biện **NHẬN XÉT CỦA GIÁO VIÊN** **HƯỚNG DẪN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Tp. HCM, ngày ... tháng ... năm 2013*

**Giáo viên hướng dẫn**

TS. Nguyễn Minh Tâm



## PHẦN B

# NỘI DUNG

### **Danh sách các hình ảnh sử dụng trong đồ án:**

- Hình 1.1: Robot dạng 3 bánh xe di chuyển trên địa hình bằng phẳng
- Hình 1.2: Robot dạng 3 bánh xe khi xuống dốc
- Hình 1.3: Robot dạng 3 bánh xe khi lên dốc
- Hình 1.4: Robot 2 bánh di chuyển trên các địa hình khác nhau theo hướng bảo toàn sự thăng bằng
- Hình 1.5: nBot
- Hình 1.6: JOE
- Hình 1.7: NXTway-GS của LEGO MINDSTORMS
- Hình 1.8: Xe Segway I2, I2 cargo, X2 Adventure
- Hình 1.9: Xe Winglet
- Hình 1.10: Xe Iswing
- 
- Hình 2.1: Mô hình robot 2 bánh tự cân bằng trên mặt phẳng
- Hình 2.2: Mô hình phi tuyến của robot hai bánh tự cân bằng trong Matlab Simulink
- Hình 2.3: Bên trong khối “Two Wheeled Balancing Robot (Non-Linear Model)”
- Hình 2.4: Bên trong khối “DeCoupling”
- Hình 2.5: Quy trình thực hiện của bộ lọc Kalman
- Hình 2.6: Tổng quan chu trình thực hiện bộ lọc Kalman hoàn chỉnh
- Hình 2.7: Cấu trúc bộ điều khiển PID cho hệ robot hai bánh tự cân bằng
- Hình 2.8: Cấu trúc bên trong bộ điều khiển PID rời rạc với thông số cố định.
- Hình 2.9: Robot hai bánh tự cân bằng sử dụng 3 bộ điều khiển PID cố định.
- Hình 2.10: Đáp ứng ngõ ra vị trí với tín hiệu đặt vị trí là xung vuông

- Hình 2.11: Đáp ứng ngõ ra góc nghiêng với tín hiệu đặt góc nghiêng là xung vuông
- Hình 2.12: Đáp ứng ngõ ra góc xoay với tín hiệu đặt góc xoay là xung vuông
- Hình 2.13: Đáp ứng ngõ ra vị trí với tín hiệu đặt vị trí là sóng sine
- Hình 2.14: Đáp ứng ngõ ra góc nghiêng với tín hiệu đặt góc nghiêng là sóng sine
- Hình 2.15: Đáp ứng ngõ ra góc xoay với tín hiệu đặt góc xoay là sóng sine
- Hình 2.16: đáp ứng ngõ ra vị trí trong trường hợp tín hiệu đặt vị trí là xung vuông
- Hình 2.17: đáp ứng ngõ ra góc nghiêng trong trường hợp tín hiệu đặt góc nghiêng là xung vuông
- Hình 2.18: đáp ứng ngõ ra vị trí trong trường hợp tín hiệu đặt vị trí là xung vuông
- Hình 2.19: đáp ứng ngõ ra góc nghiêng trong trường hợp tín hiệu đặt góc nghiêng là xung vuông
- Hình 2.20: Pin Li-Po 11.1V/2200mA
- Hình 2.21: DSP TMS320F28335
- Hình 2.22: Razor IMU 9DOF
- Hình 2.23: Sơ đồ khối cảm biến gia tốc ADXL345
- Hình 2.24: Phương và chiều các trục gia tốc tương ứng
- Hình 2.25: Đáp ứng giá trị gia tốc ngõ ra tương ứng đối với hướng của trọng lực
- Hình 2.26: Sơ đồ chân ngõ ra trên cảm biến gia tốc
- Hình 2.27: Sơ đồ khối cảm biến ITG3200
- Hình 2.28: Cảm biến HMC5883L
- Hình 2.29: Sơ đồ nguyên lý mạch cầu H
- Hình 2.30: Mạch cầu H thực tế
- Hình 2.31: Encoder
- Hình 2.32: Module Bluetooth HC06 slave
- Hình 2.33: Động cơ
- 
- Hình 3.1: Bản vẽ 3D mô hình robot 2 bánh tự cân bằng

- Hình 3.2: Mô hình robot thực tế
- Hình 3.3: Cấu trúc điều khiển phần cứng của robot 2 bánh tự cân bằng
- Hình 3.4: Sơ đồ kết nối phần cứng
- Hình 3.5: Mô hình bộ lọc Kalman với 3 biến trạng thái.
- Hình 3.6: Mô hình bộ lọc Kalman trong Matlab Simulink
- Hình 3.7: Tín hiệu vào-ra của bộ lọc Kalman
- Hình 3.8: So sánh khi cho tín hiệu góc nghiêng ngõ vào và ra của bộ lọc Kalman
- Hình 3.9: Lưu đồ giải thuật cho mô hình robot 2 bánh tự cân bằng
- Hình 3.10: Thư viện Target Support Package cho DSP F28335
- Hình 3.11: Sơ đồ tổng quát
- Hình 3.12: Khối đọc các nút nhấn
- Hình 3.13: Bên trong khối calibre
- Hình 3.14: Khối đọc cảm biến IMU
- Hình 3.15: Khối lọc Kalman
- Hình 3.16: Tổng quát Khối điều khiển trung tâm
- Hình 3.17: Bên trong Khối đọc xung encoder Hình 3.18: Bên trong Khối xuất tín hiệu điều khiển
- Hình 3.19: Khối gửi dữ liệu
- Hình 3.20: Ngõ ra góc nghiêng và điện áp trên 2 động cơ khi chạy mô hình thực tế
- Hình 3.21: Giao diện của chương trình giao tiếp máy tính

### **Danh sách các bảng sử dụng trong đồ án :**

- Bảng 2.1: Ký hiệu và ý nghĩa của các đại lượng
- Bảng 3.1: Bảng giá trị của các đại lượng trong robot 2 bánh tự cân bằng
- Bảng 3.2: Các khối chức năng sử dụng trong chương trình

## Mục lục

<b>Chương 1. Tổng quan về đề tài.....</b>	<b>13</b>
1.1 Đặt vấn đề.....	13
1.2 Tình hình nghiên cứu robot 2 bánh tự cân bằng hiện nay:.....	15
1.2.1 Các mô hình robot 2 bánh tự cân bằng.....	15
1.2.2 Một số sản phẩm thực tế dựa trên mô hình robot 2 bánh tự cân bằng.....	17
1.3 Mục tiêu đề tài.....	19
1.4 Giới hạn đề tài.....	20
1.5 Phương pháp nghiên cứu.....	20
<b>Chương 2. Các cơ sở lý thuyết.....</b>	<b>21</b>
2.1 Đặc tính động lực học.....	21
2.1.1 Mô hình hóa robot 2 bánh tự cân bằng trên địa hình phẳng.....	21
2.1.2 Mô hình hóa robot hai bánh tự cân bằng trên địa hình phẳng trong Matlab Simulink.....	26
2.2 Bộ lọc Kalman.....	28
2.2.1 Giới thiệu về bộ lọc Kalman.....	28
2.2.2 Quá trình ước lượng:.....	29
2.2.3 Bản chất xác suất của bộ lọc.....	31
2.2.4 Thuật toán Kalman rời rạc.....	31
2.3 Giải thuật điều khiển.....	34
2.3.1 Cấu trúc bộ điều khiển PID cho robot hai bánh tự cân bằng.....	34
2.3.2 Bộ điều khiển PID với thông số cố định.....	35
2.3.3 Các thành phần chính của mô hình:.....	42
<b>Chương 3. Thiết kế hệ thống.....</b>	<b>49</b>
3.1 Thiết kế phần cứng mô hình Robot 2 bánh tự cân bằng.....	49
3.1.1 Thiết kế cơ khí.....	49
3.1.2 Cấu trúc điều khiển phần cứng.....	51
3.2 Thiết kế phần mềm.....	52
3.2.1 Bộ lọc Kalman cho thành phần IMU.....	52
3.2.2 Lưu đồ giải thuật.....	58
3.2.3 Bộ điều khiển nhúng cho robot 2 bánh tự cân bằng.....	59

3.2.4	Giao diện trên máy tính.....	64
<b>Chương 4.</b>	<b>Kết luận và hướng phát triển của đề tài.....</b>	<b>66</b>
4.1	Những kết quả đã đạt được.....	66
4.2	Hạn chế của đề tài.....	66
4.3	Kết luận.....	66
4.4	Hướng phát triển.....	66

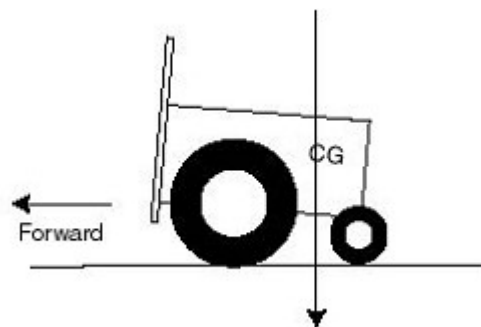
# Chương 1. Tổng quan về đề tài

## 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.

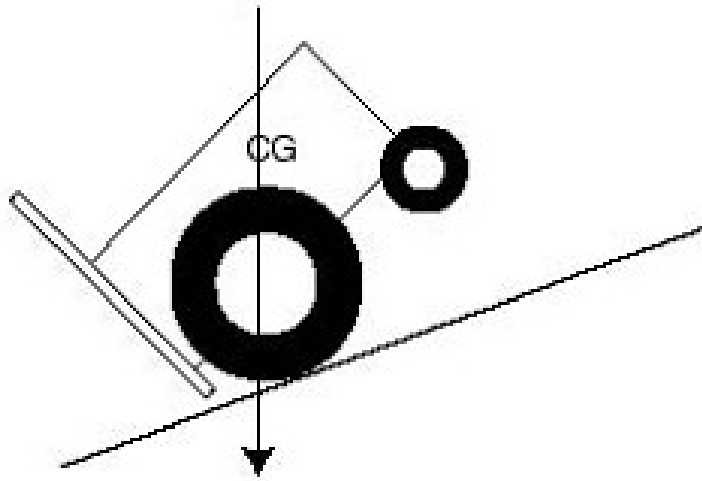
Kết quả nghiên cứu mô hình con lắc ngược cơ bản, ví dụ như mô hình xe-con lắc, con lắc ngược quay... có thể ứng dụng và kế thừa sang các mô hình tương tự khác nhưng có tính ứng dụng thực tiễn hơn, chẳng hạn như mô hình tên lửa, mô hình xe hai bánh tự cân bằng... do đó khắc phục được những nhược điểm 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.



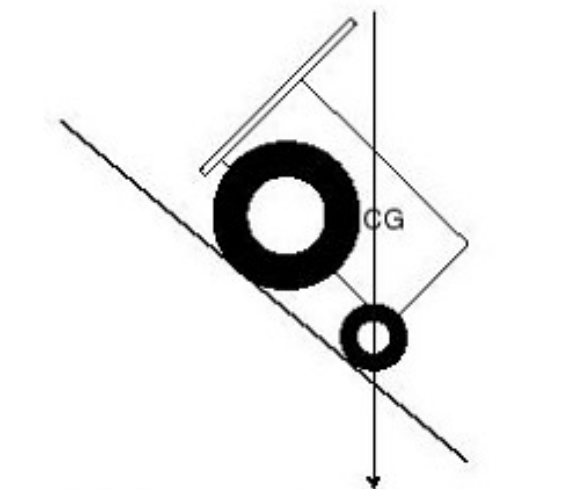
Hình 1.1 - Robot dạng 3 bánh xe di chuyển trên địa hình bằng phẳng

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



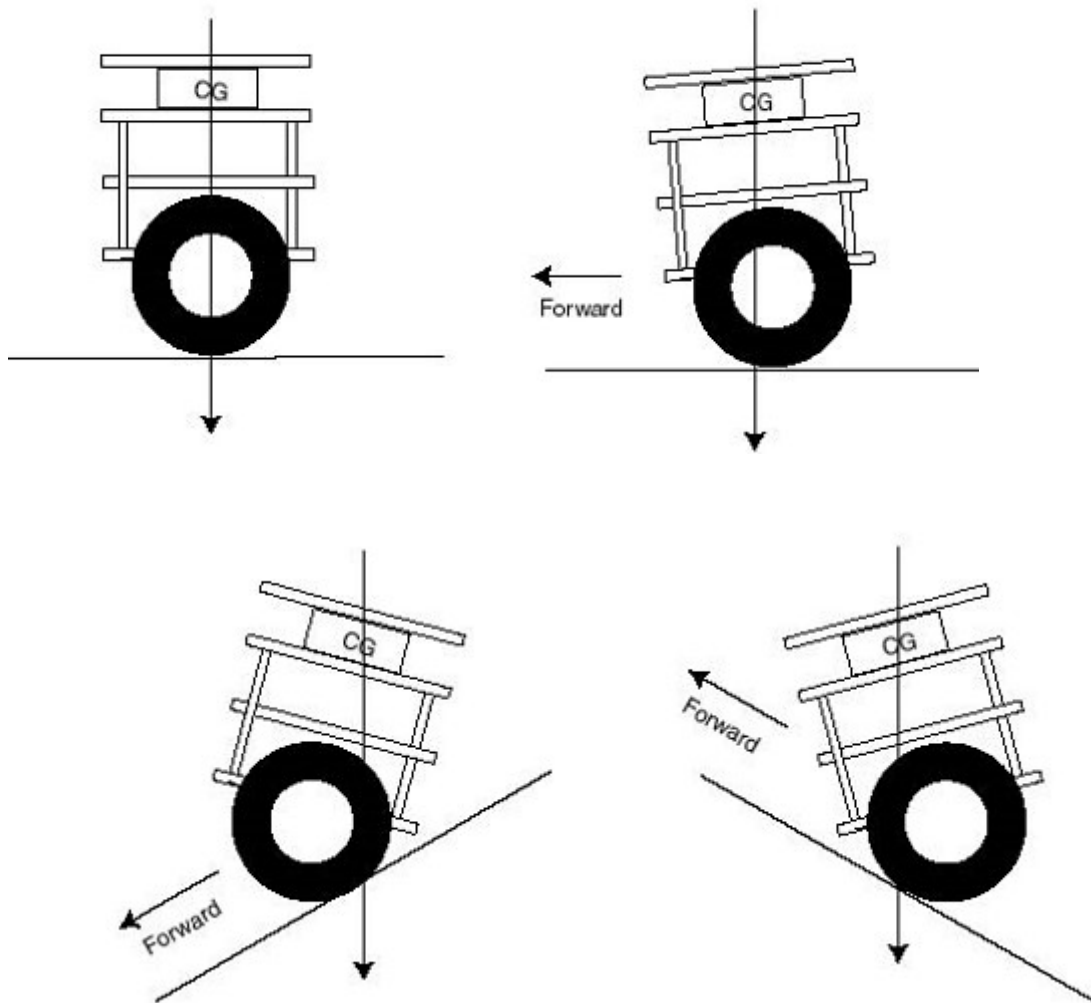
Hình 1.2 - Robot dạng 3 bánh xe khi xuống dốc

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.



Hình 1.3 - 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.



Hình 1.4 - Robot 2 bánh di chuyển trên các địa hình khác nhau theo hướng bảo toàn sự thăng bằng

## 1.2 Tình hình nghiên cứu robot 2 bánh tự cân bằng hiện nay:

### 1.2.1 Các mô hình robot 2 bánh tự cân bằng

#### 1.2.1.1 nBot

Robot nBot do David P.Anderson chế tạo. Nguyên tắc điều khiển nBot như sau: các bánh xe sẽ chạy theo hướng mà phần trên robot sắp ngã, nếu bánh xe có thể được lái theo cách giữ vững trọng tâm robot thì robot sẽ được giữ cân bằng.

Quá trình điều khiển sử dụng tín hiệu từ hai cảm biến: cảm biến góc nghiêng của thân robot so với phương của trọng lực và encoder gắn ở bánh xe để đo vị trí robot. Tín hiệu này hình thành nên 4 biến: góc nghiêng thân robot, vận tốc góc nghiêng,



vị trí robot và vận tốc robot; 4 biến này được tính toán thành điện áp điều khiển động cơ cho robot.



Hình 1.5: nBot

#### 1.2.1.2 JOE

JOE do phòng thí nghiệm điện tử công nghiệp của viện Công nghệ Liên bang Lausanne, Thụy Sĩ tạo ra vào năm 2002. Hình dạng của nó gồm hai bánh xe đồng trục, mỗi bánh gắn với một động cơ DC, robot này có thể chuyển động xoay theo hình chữ U. Hệ thống điều khiển gồm hai bộ điều khiển “không gian trạng thái” (state space) tách rời nhau, kiểm soát động cơ để giữ cân bằng cho hệ thống. Thông tin trạng thái được cung cấp bởi hai encoder quang và hai cảm biến: gia tốc góc và con quay hồi chuyển (gyro). JOE được điều khiển bởi một bộ điều khiển từ xa RC . Bộ điều khiển trung tâm và xử lý tín hiệu là một board xử lý tín hiệu số (DSP) phát triển bởi chính nhóm và của viện Federal, kết hợp với FPGA của XILINC.



Hình 1.6: JOE

### 1.2.1.3 NXTway-GS



Hình 1.7: NXTway-GS của LEGO MINDSTORMS

## 1.2.2 Một số sản phẩm thực tế dựa trên mô hình robot 2 bánh tự cân bằng

### 1.2.2.1 Xe Segway

Segway PT (viết tắt của Segway Personal Transporter - Xe cá nhân Segway), thường được gọi tắt là Segway là một phương tiện giao thông cá nhân có hai bánh, hoạt động trên cơ chế tự cân bằng do Dean Kamen phát minh. Loại xe này được sản xuất bởi công ty Segway Inc. ở bang New Hampshire, Hoa Kỳ. Từ "Segway" phát âm gần giống với "segue" (một từ gốc tiếng Ý có nghĩa "di chuyển nhẹ nhàng"). Đặc điểm nổi bật của Segway là cơ chế tự cân bằng nhờ hệ thống máy tính, động cơ và con quay hồi chuyển đặt bên trong xe, nó giúp cho xe dù chỉ có một trục chuyển động với hai bánh nhưng luôn ở trạng thái cân bằng, người sử dụng chỉ việc ngả về đằng trước hoặc đằng sau để điều khiển xe đi tiến hoặc đi lùi.



Hình 1.8: Xe Segway I2, I2 cargo, X2 Adventure

#### 1.2.2.2 Xe Winglet

Xe Winglet do hãng Toyota phát triển được giới thiệu năm 2008 có kích thước phần đế chỉ bằng tờ giấy A3. Xe dựa trên mô hình 2 bánh tự cân bằng và có cảm biến để nhận biết cử động của người điều khiển từ đó đưa ra lệnh để xe hoạt động theo ý muốn của người điều khiển.



Hình 1.9: Xe Winglet

#### 1.2.2.3 Iswing Toyota:

I-swing được mệnh danh là một trong những ý tưởng táo bạo nhất của ngành công nghiệp ô tô trong thời gian gần đây. Xuất hiện lần đầu tiên tại triển lãm Tokyo 2005, I-swing là biểu tượng cho phương tiện cá nhân trong tương lai bởi nó khác xa so với ô tô thông thường. Sự di chuyển, điều khiển, giao tiếp giữa người và xe đều mang

tính nhân bản. Chẳng hạn như khi đặt ghế ở chế độ "thân xe mở tối thiểu", tầm mắt tài xế sẽ ngang với người đang đứng nên rất dễ trò chuyện và tạo cảm giác thân thiện. I-swing được điều khiển thông qua tay nắm và các nút, giống với cách con người sử dụng các thiết bị điện tử hàng ngày.



Hình 1.10: Xe Iswing

### 1.3 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.
- Tính toán các thông số động lực học, xây dựng các hàm không gian-trạng thái (state-space) của mô hình.
- Tìm hiểu, lựa chọn các loại cảm biến và bộ điều khiển trung tâm. Trong đề tài này sẽ sử dụng cảm biến IMU 9 DOF và bộ điều khiển DSP F28335.
- Mô phỏng mô hình trên Matlab Simulink.
- 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.

## 1.4 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)

## 1.5 Phương pháp nghiên cứu

Xây dựng mô hình lý thuyết robot 2 bánh tự cân bằng và mô phỏng mô hình trên Matlab Simulink.

Xây dựng mô hình thực:

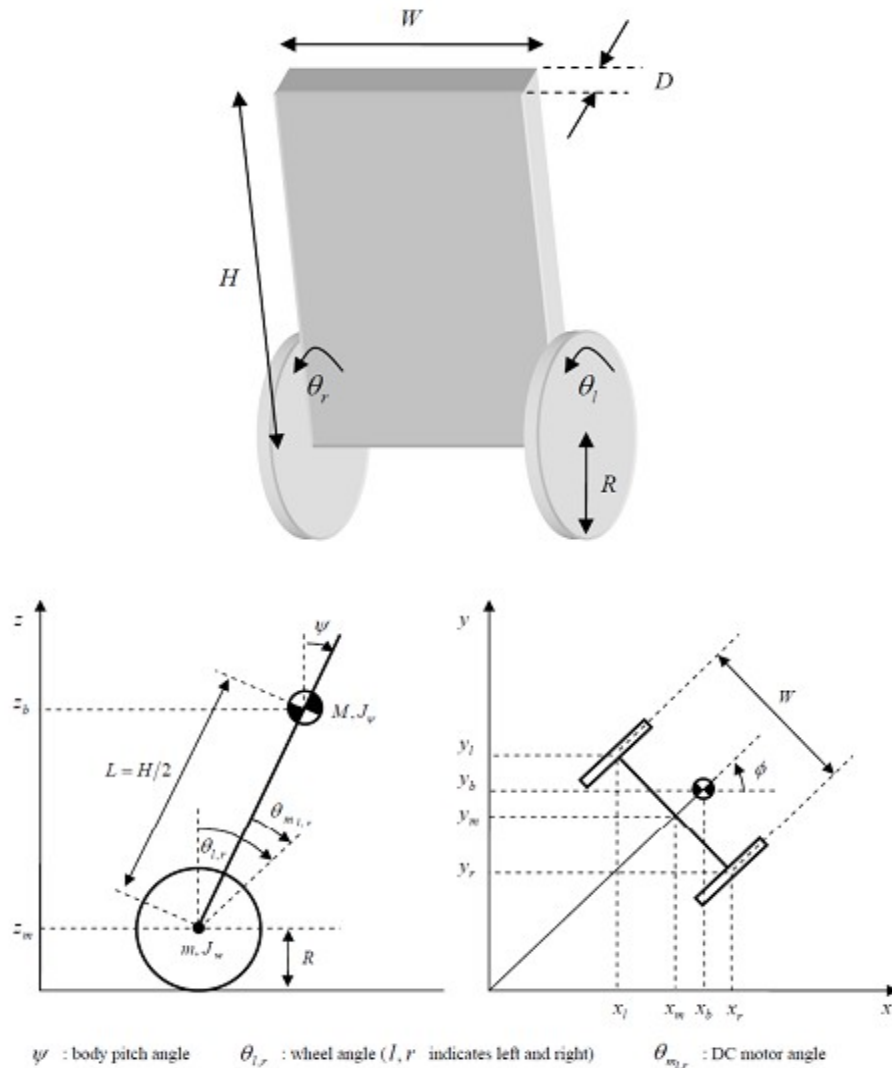
- Thiết kế cơ khí: khung sườn của mô hình.
- Ứng dụng các mạch điện tử, mạch công suất, cảm biến, vi xử lý.
- Mạch cảm biến và bù trừ giá trị cảm biến.

## 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.



Hình 2.1: 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
$J_m$	$kg \cdot m^2$	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
$\theta$	rad	Góc trung bình của bánh trái và phải
$\theta_{l,r}$	rad	Góc của bánh trái và phải
$\psi$	rad	Góc nghiêng của phần thân robot
$\phi$	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
$F_\theta, F_\psi, F_\phi$	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

Bảng 2.1: Kí 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:

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

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 :

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

Với

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

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

Phương trình thế năng:

$$U = mg z_l + mg z_r + mg z_b \quad [2.10]$$

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:

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

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

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

Momen động lực do động cơ DC sinh ra:

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

Và:

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

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

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

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:

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

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

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

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

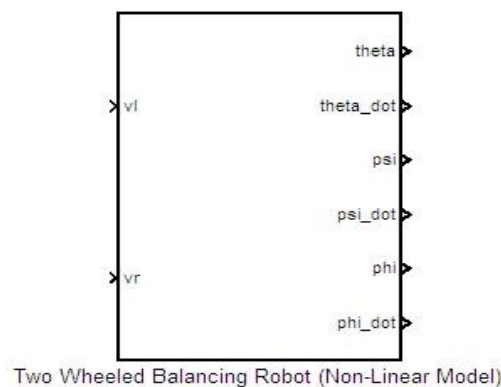
Thu được phương trình động lực học mô tả chuyển động của robot như sau:\

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

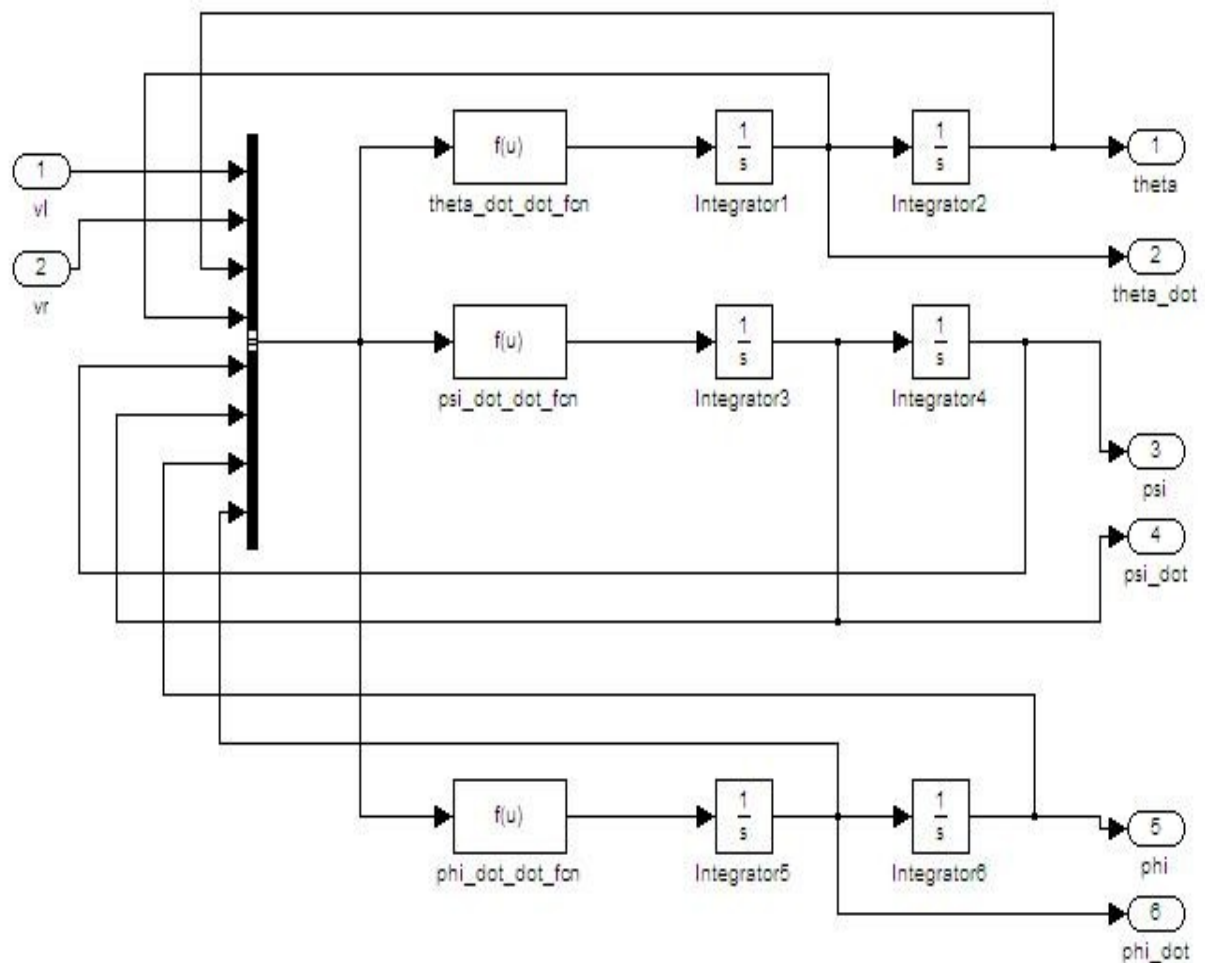
$$(MLR \cos \psi - 2n^2 J_m) \dot{\theta} + (ML^2 + J_\psi + 2n^2 J_m) \dot{\psi} - MgL \sin \psi - ML^2 \dot{\phi}^2 \sin \psi \cos \psi = -\alpha (v_l + v_r) + 2\beta \dot{\theta} - 2\beta \dot{\psi} \quad [2.29]$$

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

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



Hình 2.2 : Mô hình phi tuyến của robot hai bánh tự cân bằng trong Matlab Simulink

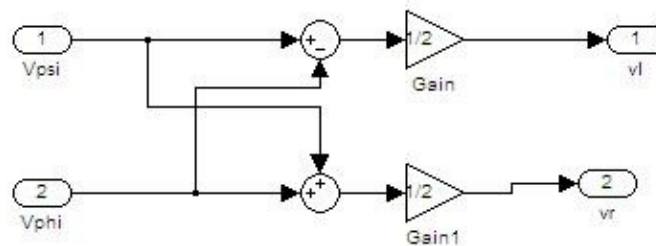
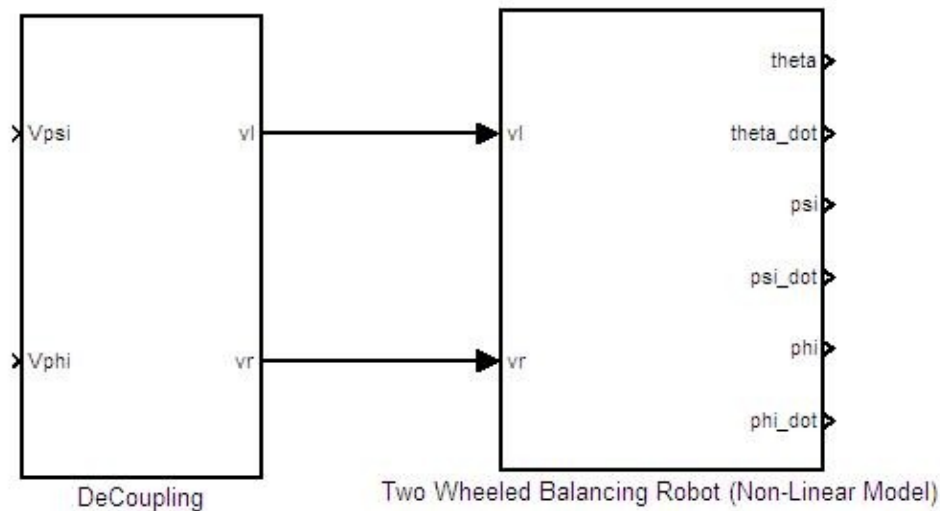


Hình 2.3: Bên trong khối “Two Wheeled Balancing Robot (Non-Linear Model)”

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  $V_\psi, V_\phi$

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

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



Hình 2.4: 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  $x \in \mathbb{R}^n$  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:

$$x_k = A x_{k-1} + B u_{k-1} + w_{k-1} \quad [2.32]$$

Với giá trị  $z \in \mathbb{R}^m$  là:

$$z_k = H x_k + v_k \quad [2.33]$$

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  $u \in \mathbb{R}^L$  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:

$$\hat{x}_k^- = E \{ x_k \mid y_1, y_2, \dots, y_{k-1} \} \quad \text{là giá trị ước lượng của } x_k \text{ trước khi ta xử lý giá trị đo}$$

tại thời điểm  $k$ .

$$\hat{x}_k^+ = E \{ x_k \mid y_1, y_2, \dots, y_k \} \quad \text{là giá trị ước lượng của } x_k \text{ sau khi ta xử lý giá trị đo tại}$$

thời điểm  $k$ .

$$\hat{x}_k \in \mathbb{R}^n \quad \text{là giá trị ước lượng trạng thái sau tại bước } k \text{ có được sau khi so sánh với giá}$$

trị đo  $z_k$ . Và chúng ta có sai số ước lượng trạng thái trước và sau:

$$e_k = x_k - \hat{x}_k \quad [2.36]$$

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

$$H \hat{x}_k : \quad [2.39]$$

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:

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



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  $P_k^{-\hat{\epsilon}}$  tiến đến 0, khi đó:

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

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  $Hx_k^{-\hat{\epsilon}}$  sẽ có độ tin cậy càng thấp. Mặt khác, nếu tương quan sai

số ước lượng priori  $P_k^{-\hat{\epsilon}}$  tiến tới 0 thì  $z_k$  sẽ không đáng tin mà giá trị ước lượng

$Hx_k^{-\hat{\epsilon}}$  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  $\hat{x}_k^{-\hat{\epsilon}}$

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

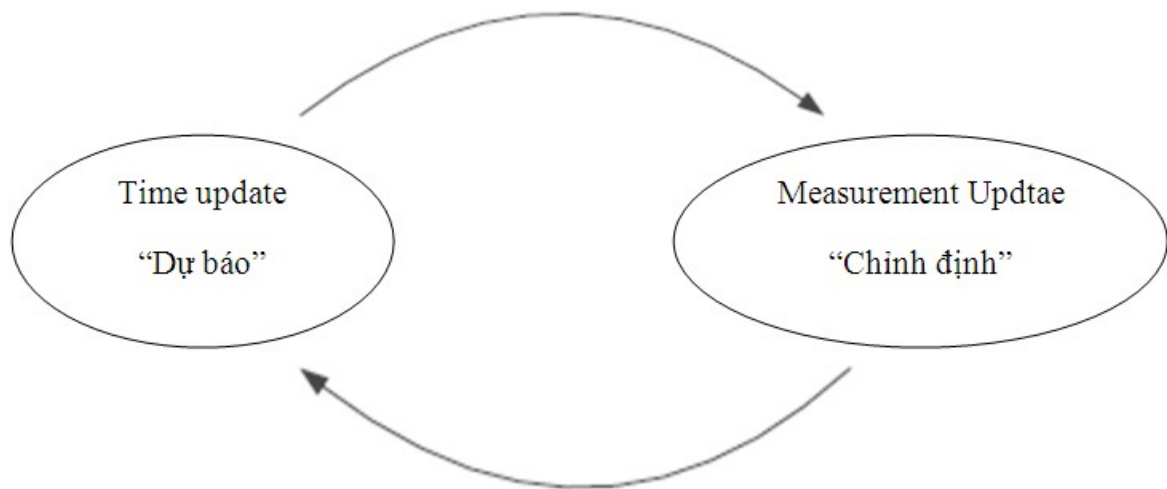
bố trạng thái. Tương quan sai số ước lượng trạng thái posteriori phản ánh sự thay đổi của phân bố trạng thái. Ngoài ra ta còn có:

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

#### 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:



Hình 2.5: Quy trình thực hiện của bộ lọc Kalman

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

$$\begin{aligned} \hat{x}_k &= A \hat{x}_{k-1} + B u_{k-1} \\ \hat{x}_k &= A \hat{x}_{k-1} \end{aligned} \quad [2.46]$$

$$\begin{aligned} P_k &= A P_{k-1} A^T + Q \\ P_k &= A P_{k-1} A^T + Q \end{aligned} \quad [2.47]$$

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

$$\begin{aligned} & -\hat{z}_k - H^T P_k^{-1} (z_k - H \hat{x}_k) \\ & K_k = P_k^{-1} H^T (H^T P_k^{-1} H + R)^{-1} \end{aligned} \quad [2.48]$$

$$\begin{aligned} & \hat{x}_k = \hat{x}_k + K_k (z_k - H \hat{x}_k) \\ & \hat{x}_k = \hat{x}_k \end{aligned} \quad [2.49]$$

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

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.

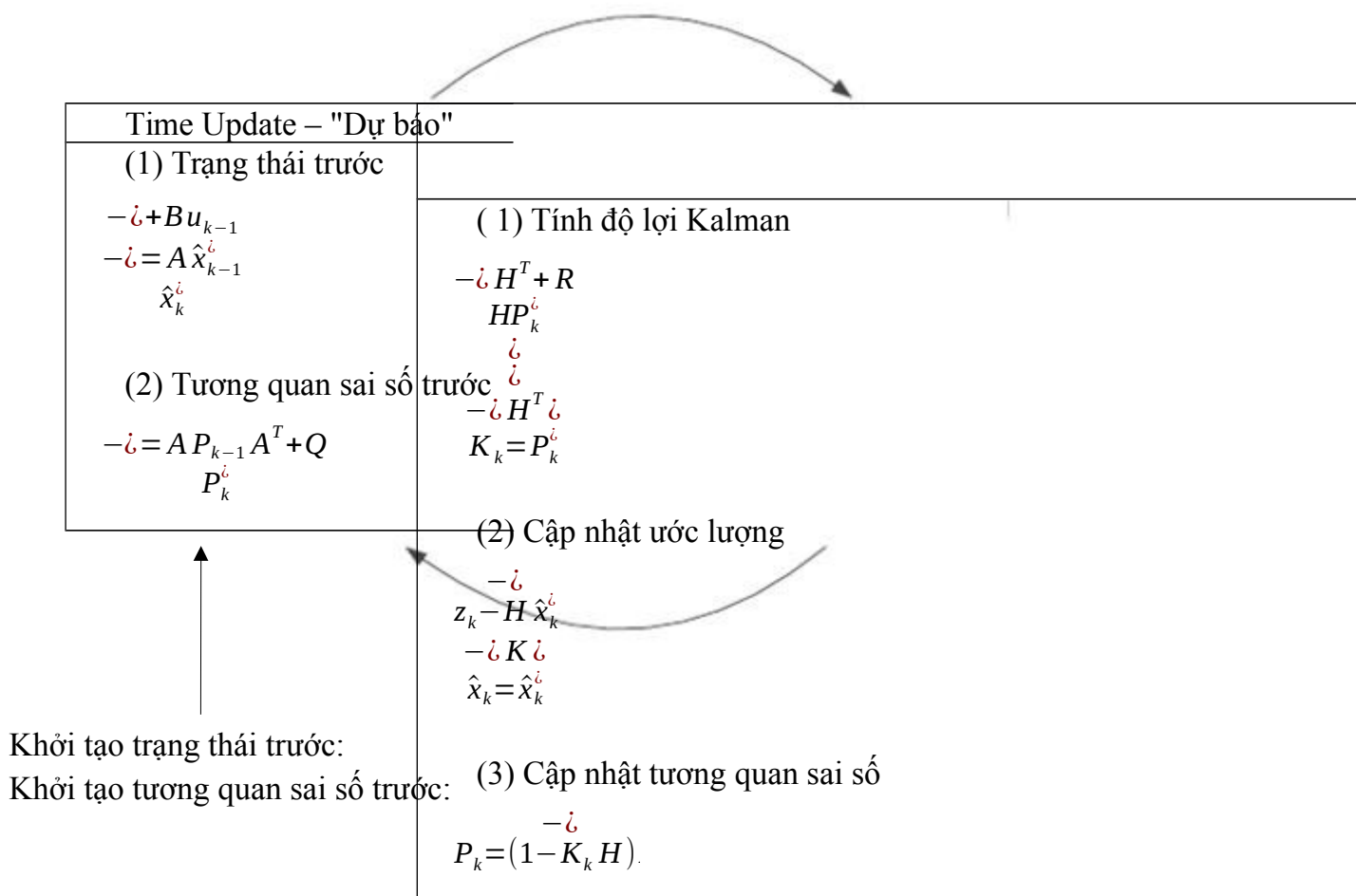
Trong điều kiện thực hiện thực tế của bộ lọc, giá trị nhiễu tương quan  $R$  thường được dùng làm giá trị ưu tiên để tính toán cho bộ lọc. Trên thực tế, việc đo các giá trị ma trận  $R$  là rất phổ biến bởi vì chúng ta có thể đo quy trình theo nhiều cách vì vậy mà thường lấy mẫu giá trị để đưa ra khuynh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



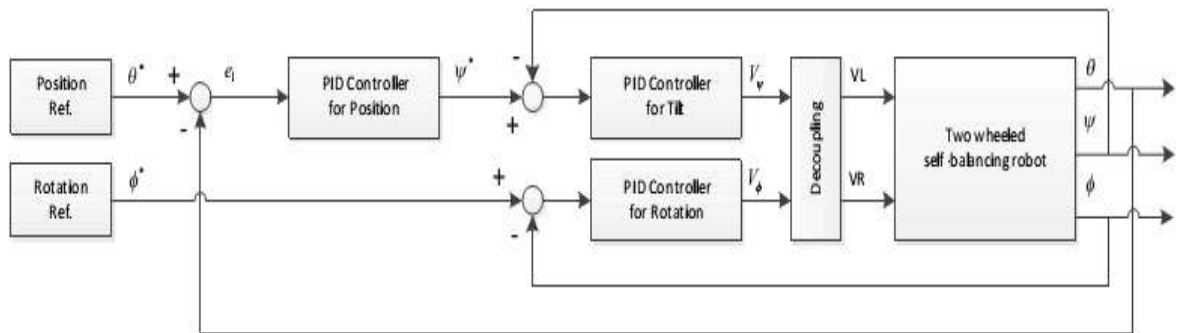
Hình 2.6: Tổng quan chu trình thực hiện bộ lọc Kalman hoàn chỉnh

## 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 ( $\psi$ )
- Bộ PID điều khiển vị trí ( $\theta$ )
- Bộ PID điều khiển góc xoay ( $\phi$ )



Hình 2.7: 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)

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

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

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

Do đó

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

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

Và

$$\begin{aligned} (1-z) &= \frac{T_s}{2} F[z](1+z^{-1}) \\ y(k) &= y(k-1) + \frac{f(k) - f(k-1)}{2} \cdot T_s ZY[z] \end{aligned}$$

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

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

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

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

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

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

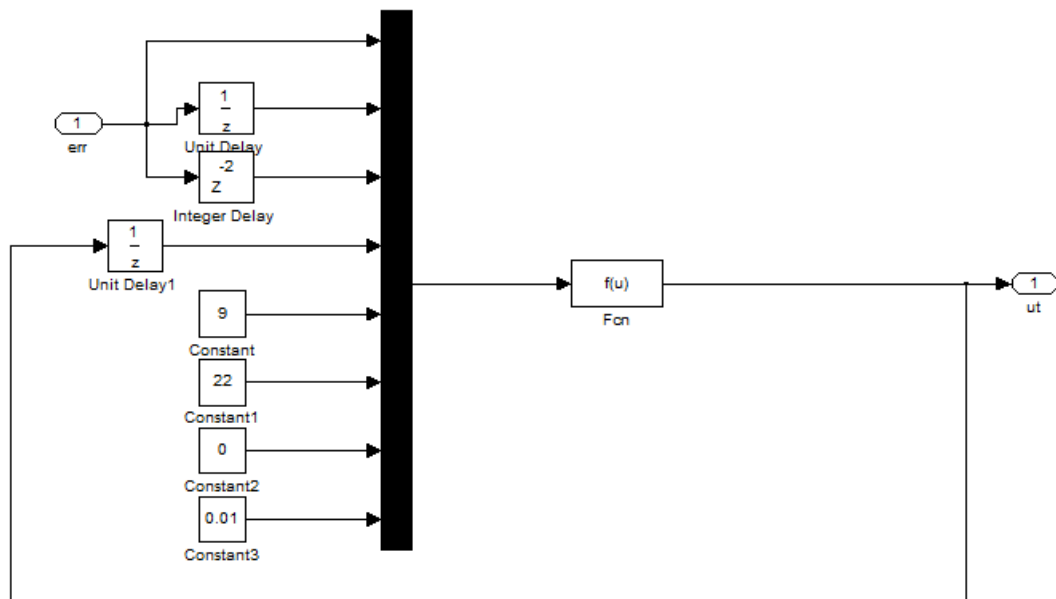
$$\bar{u}[k] = u[k-1] + ae[k] + be[k-1] + ce[k-2] \quad [2.61]$$

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

### 2.3.2 Bộ điều khiển PID với thông số cố định

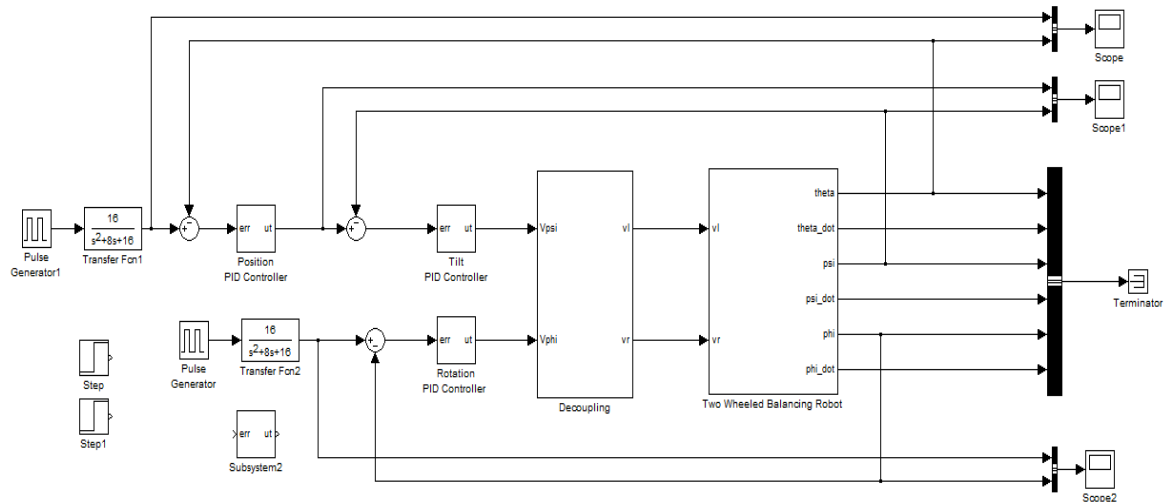
Bộ điều khiển PID với thông số KP, KI, KD cố định như trên, hệ thống chỉ làm việc tốt trong điều kiện hệ số KP, KI, KD đã được chỉnh định tối ưu và trong quá trình làm việc, các thông số trong mô hình không đổi.

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.



Hình 2.8: 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:



Hình 2.9: 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( $f_w$ ) thông qua file “Two\_wheelRobot.m”

### 2.3.2.1 Trường hợp thông số mô hình được chọn cố định

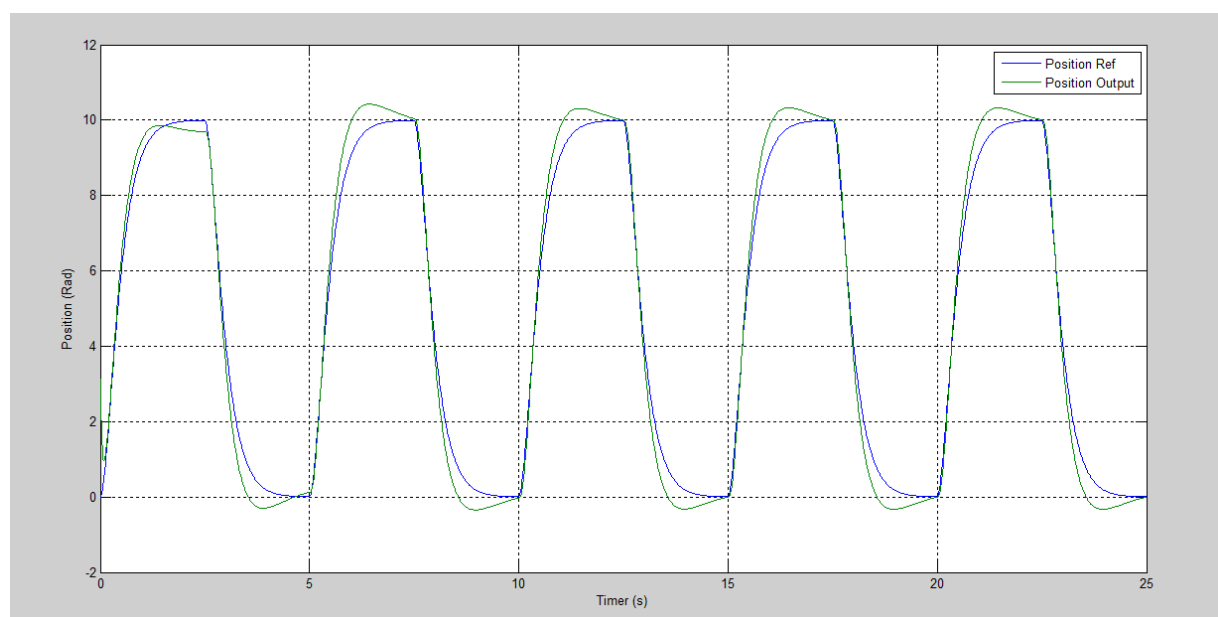
Với  $M=8,2$  kg,  $f_w=0,001$ , thông số của các bộ điều khiển lần lượt là:

-Position PID controller [ $K_p=12$   $K_i=16$   $K_d=0$ ]

-Tilt PID controller [ $K_p=120$   $K_i=110$   $K_d=90$ ]

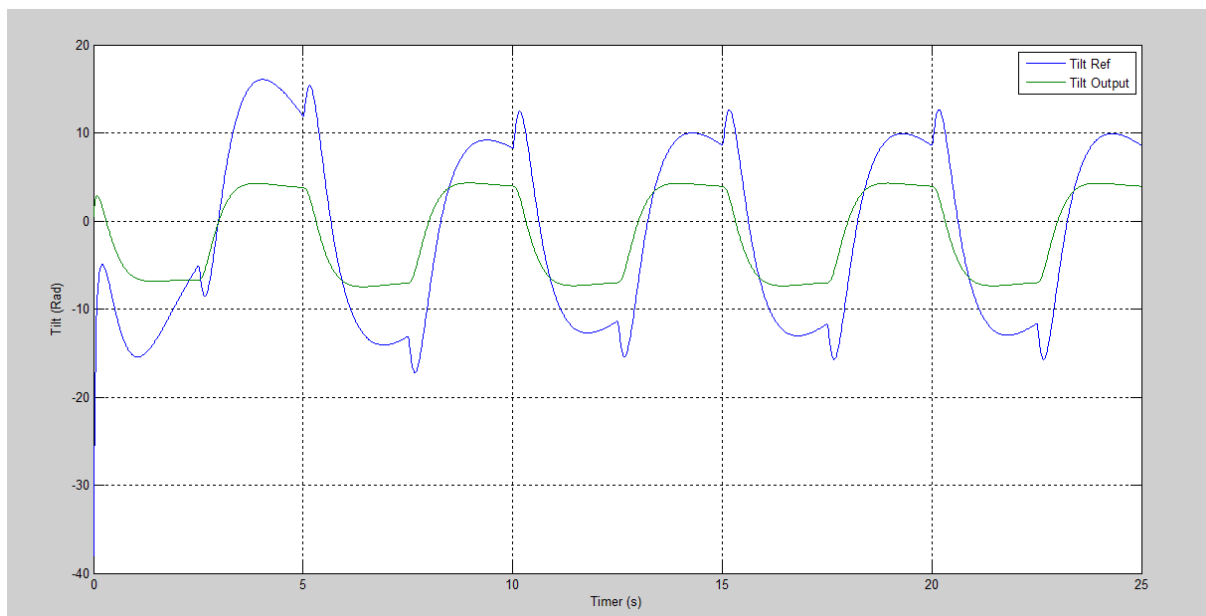
-Rotation PID Controller [ $K_p=334$   $K_i=229$   $K_d=432$ ]

Bộ PID cố định sẽ điều khiển hệ thống hoạt động tốt ứng với các tín hiệu đầu vào là xung vuông hoặc sóng sine như sau:

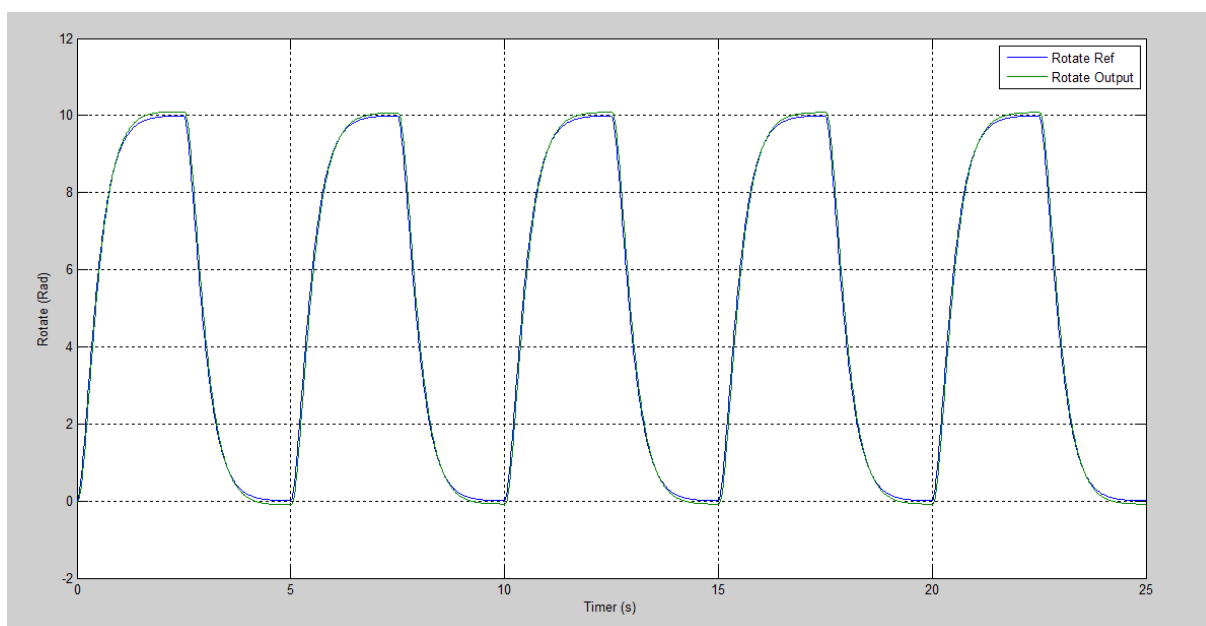




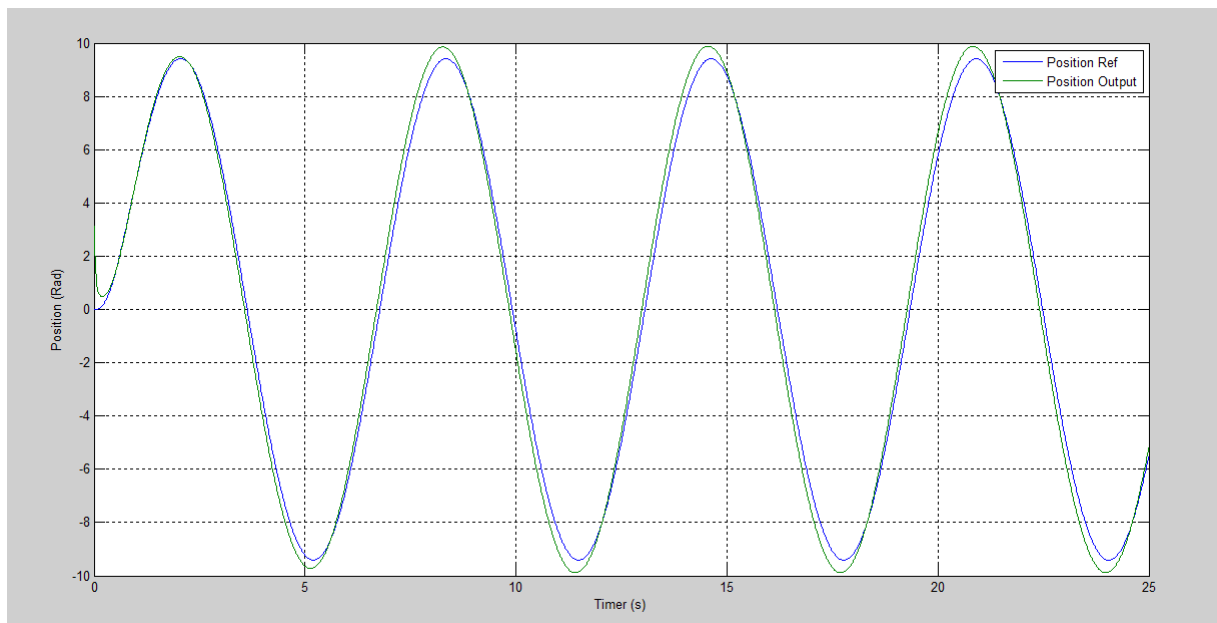
Hình 2.10: Đáp ứng ngõ ra vị trí với tín hiệu đặt vị trí là xung vuông



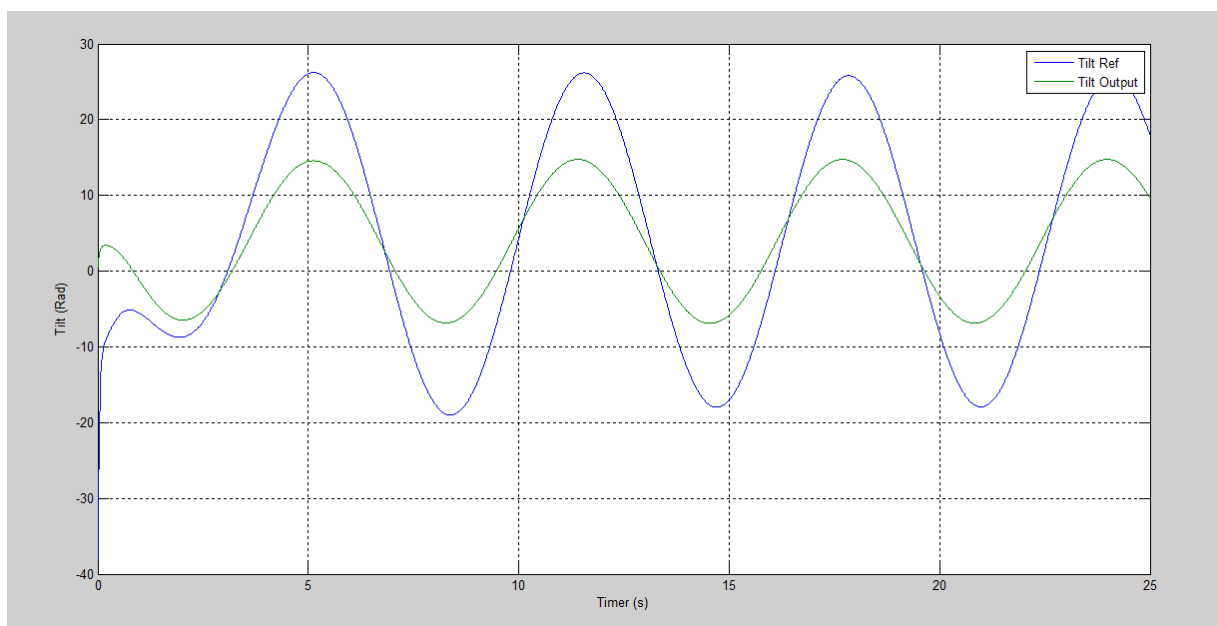
Hình 2.11: Đáp ứng ngõ ra góc nghiêng với tín hiệu đặt góc nghiêng là xung vuông



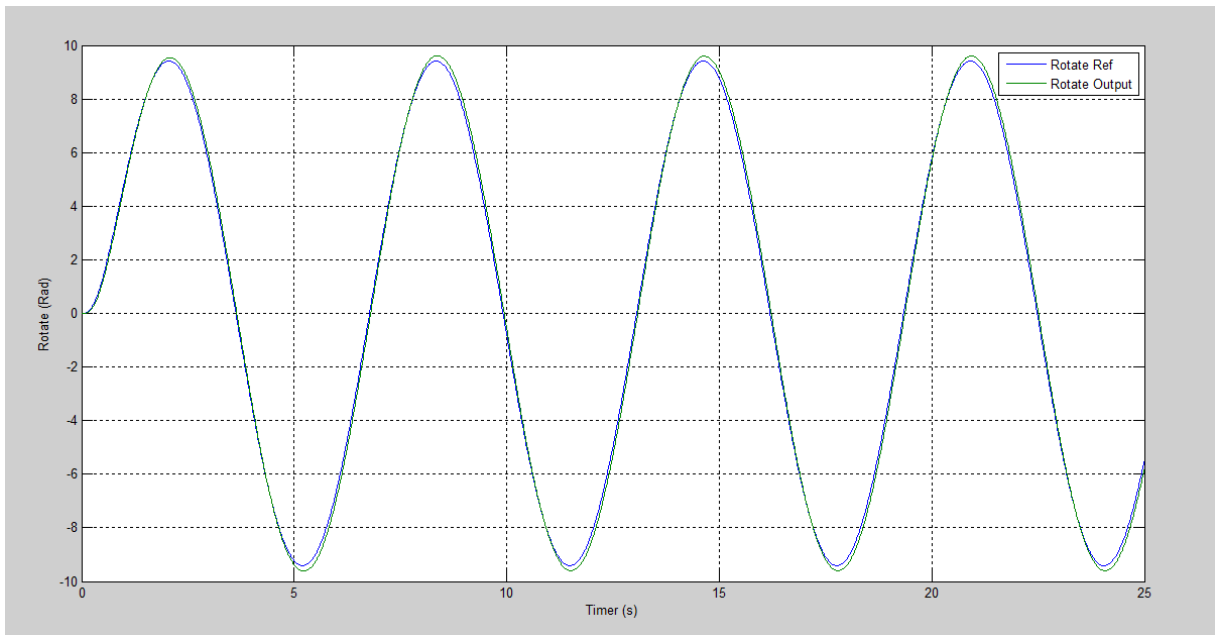
Hình 2.12: Đáp ứng ngõ ra góc xoay với tín hiệu đặt góc xoay là xung vuông



Hình 2.13: Đáp ứng ngõ ra vị trí với tín hiệu đặt vị trí là sóng sine



Hình 2.14: Đáp ứng ngõ ra góc nghiêng với tín hiệu đặt góc nghiêng là sóng sine

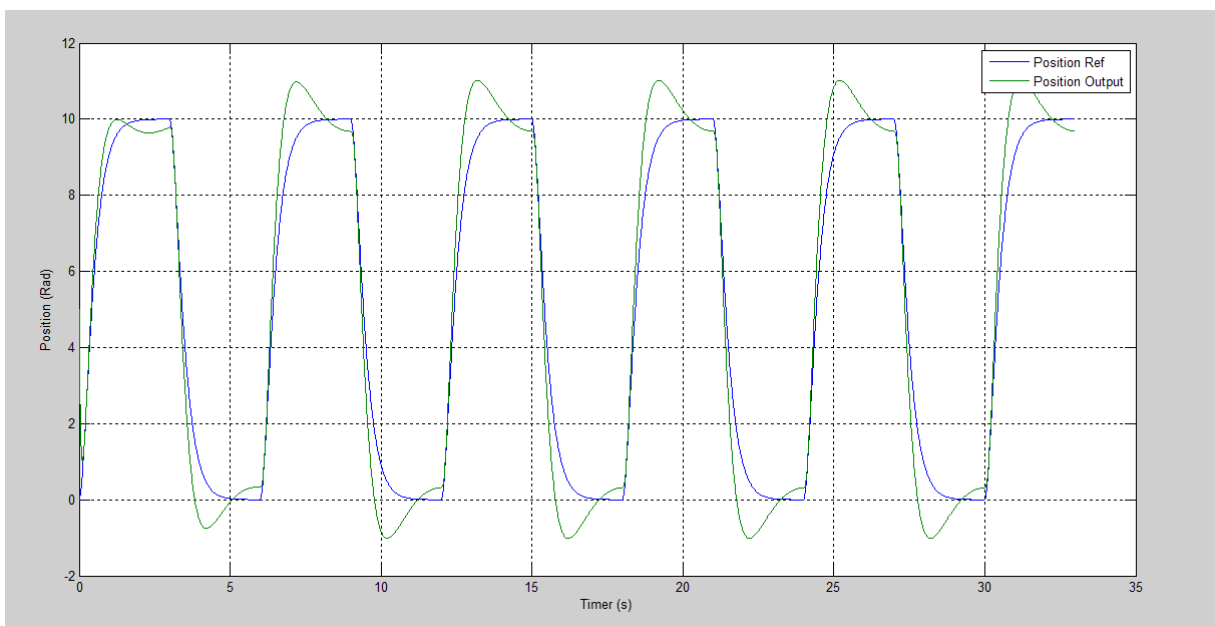


Hình 2.15: Đáp ứng ngõ ra góc xoay với tín hiệu đặt góc xoay là sóng sine

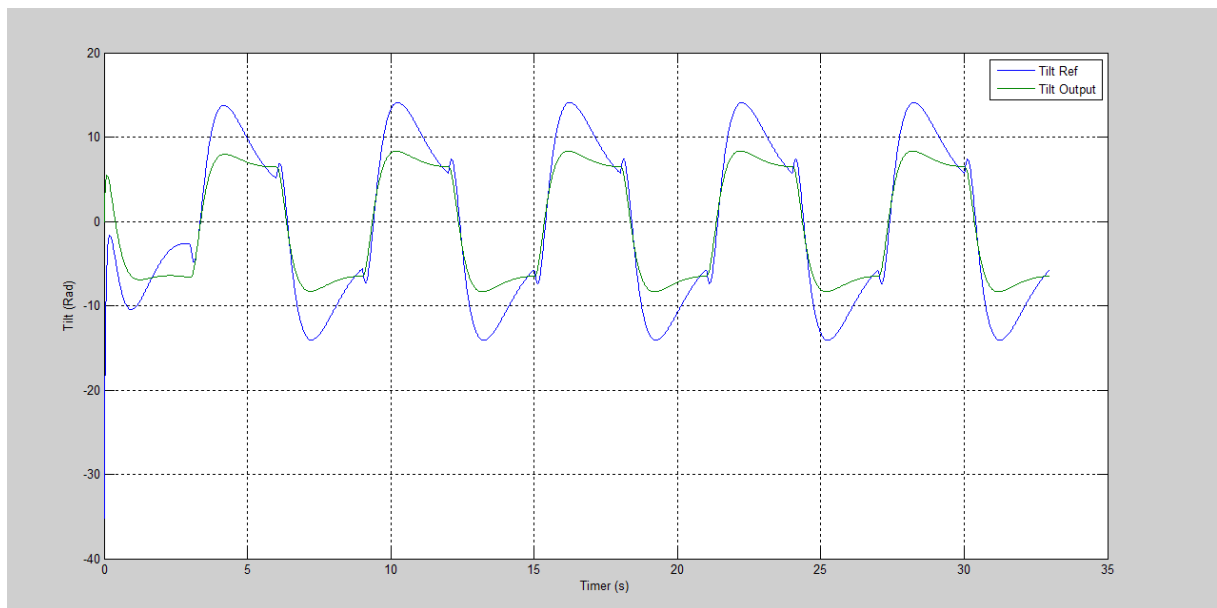
### 2.3.2.2 Trường hợp có sự thay đổi thông số mô hình:

- Trường hợp fw thay đổi:

Giá trị  $f_w$  thay đổi từ 0.001N đến 0.008N còn khối lượng xe M vẫn giữ cố định. Đáp ứng ngõ ra của hệ thống có sự thay đổi như sau:



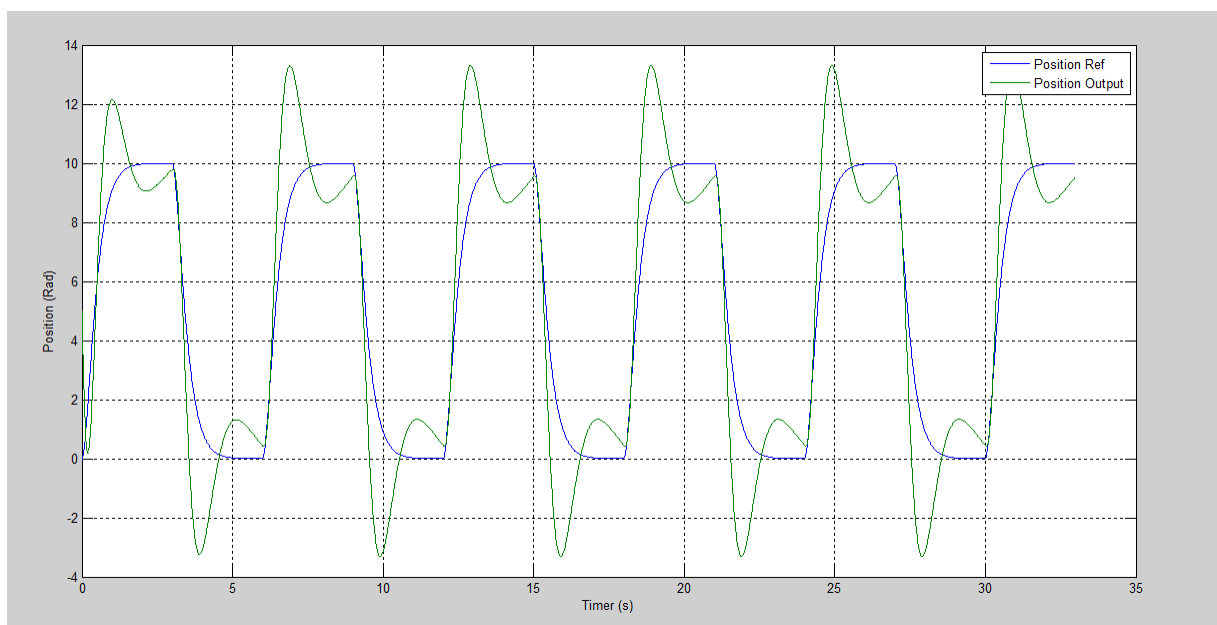
Hình 2.16: đáp ứng ngõ ra vị trí trong trường hợp tín hiệu đặt vị trí là xung vuông



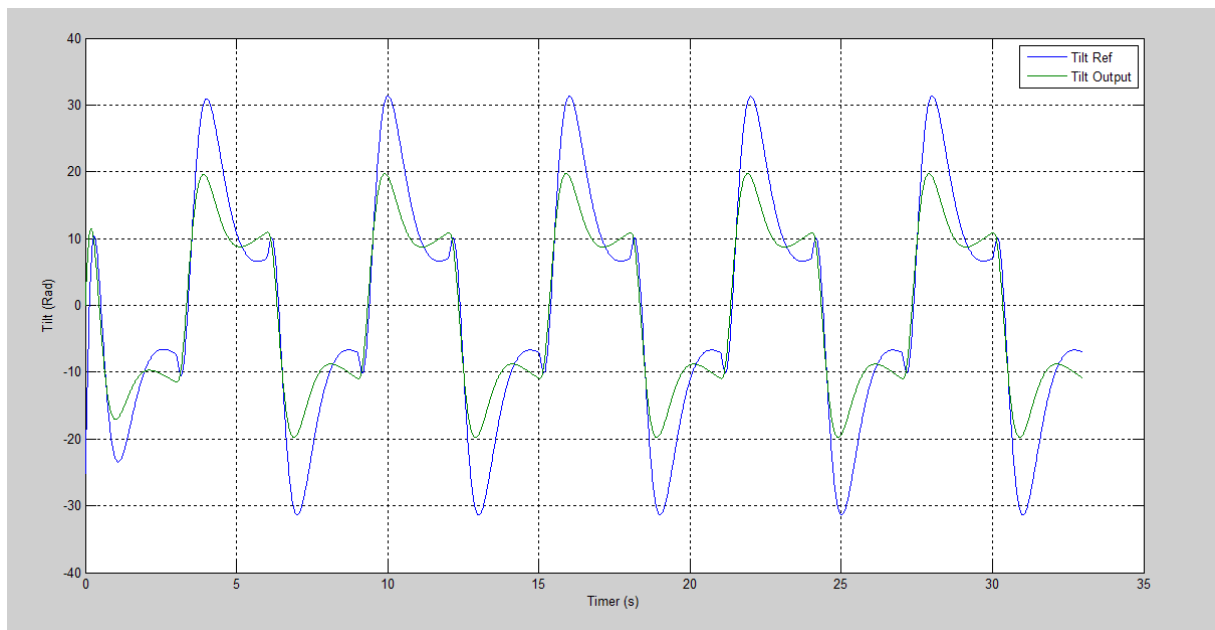
Hình 2.17: đáp ứng ngõ ra góc nghiêng trong trường hợp tín hiệu đặt góc nghiêng là xung vuông

- Trường hợp thay đổi khối lượng xe (M)

Giá trị  $M$  thay đổi từ 5 Kg đến 6Kg còn  $f_w=0.001$  vẫn giữ cố định. Đáp ứng ngõ ra của hệ thống có sự thay đổi như sau:



Hình 2.18: đáp ứng ngõ ra vị trí trong trường hợp tín hiệu đặt vị trí là xung vuông



Hình 2.19: đáp ứng ngõ ra góc nghiêng trong trường hợp tín hiệu đặt góc nghiêng là xung vuông

### 2.3.2.3 Kết luận

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  $f_w$  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  $f_w$  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

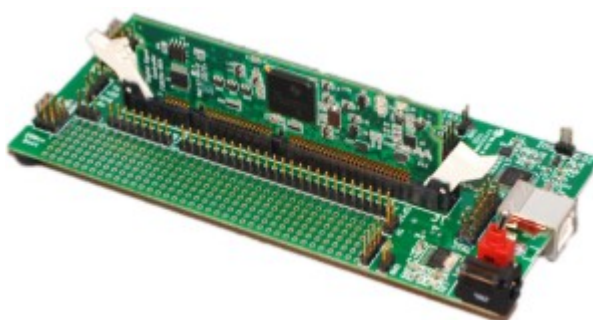
Nguồn cung cấp cho robot sử dụng pin Lithium-polimer 11.1V/2200mA (x2)



Hình 2.20: Pin Li-Po 11.1V/2200mA

### 2.3.3.2 *Vi xử lý*

Robot sử dụng DSP (Digital signal processing) TMS320F28335 của hãng Texas Instrument dưới dạng KIT thí nghiệm Delfino C28335 làm trung tâm thu thập dữ liệu và điều khiển mọi hoạt động của robot.



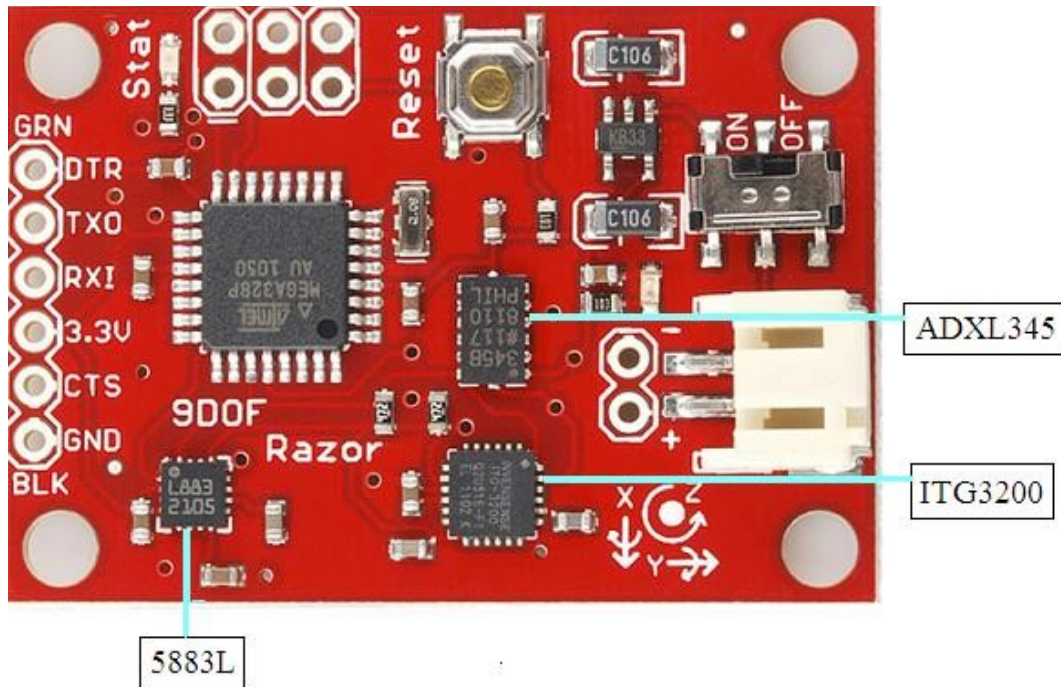
Hình 2.21: DSP TMS320F28335

- Hoạt động ở tần số 150Mhz ( 6.67ns/chu kỳ máy).
- Hỗ trợ hoạt động tính toán số thực single precision 32bit FPU.
- 16 x 16 Dual MAC, 16 x 16 và 32 x 32 MAC.
- Bus có kiến trúc Harvard.
- Kênh điều khiển DMA cho: ADC, McBSP, ePWM, XINTF và SARAM.
- Bộ nhớ: 256 x 16KB Flash, 34KB x 16 SARAM.
- Boot ROM 8K x 16.
- 18 kênh PWM trong đó có 6 kênh PWM có độ phân giải cao 150ps.
- 8 Timer 32 bits.
- Hỗ trợ các chuẩn giao tiếp: 2xCAN, 3xSCI(UART), 2xMcBSP, 1xSPI, 1xI2C.
- 2 bộ giao tiếp ENCODER 32 bit.

### 2.3.3.3 *Cảm biến IMU*

Robot sử dụng cảm biến IMU (Inertial Measurement Unit-cảm biến đo lường quán tính) để tính toán, ước lượng giá trị góc nghiêng và vận tốc góc nghiêng

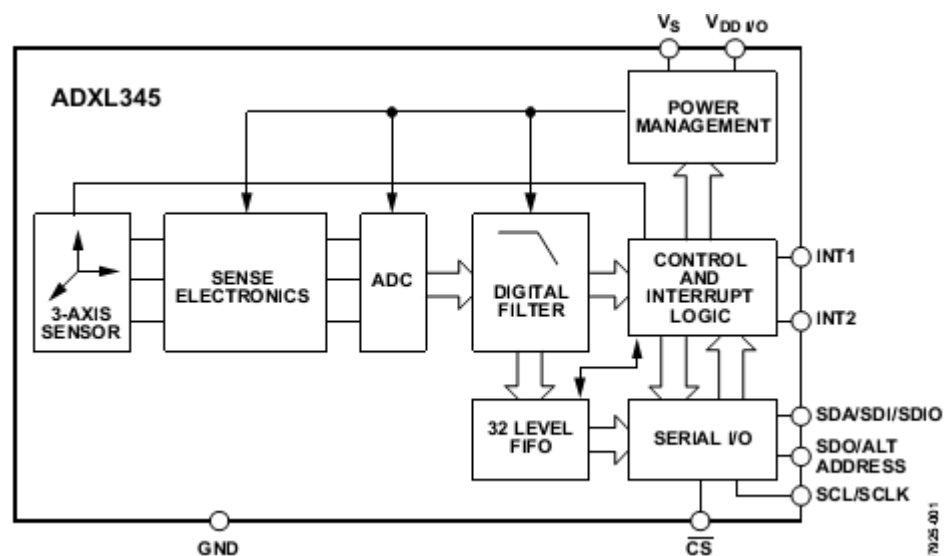
thânrobot, tên đầy đủ là Razor IMU 9 Degrees Of Freedom, là sự tích hợp của cảm biến gia tốc ADXL 345 (triple-axis accelerometer), cảm biến con quay hồi chuyển ITG3200 (MEMS triple-axis gyro) và cảm biến từ trường HMC5883L (triple-axis magnetometer) trên cùng 1 bo mạch.



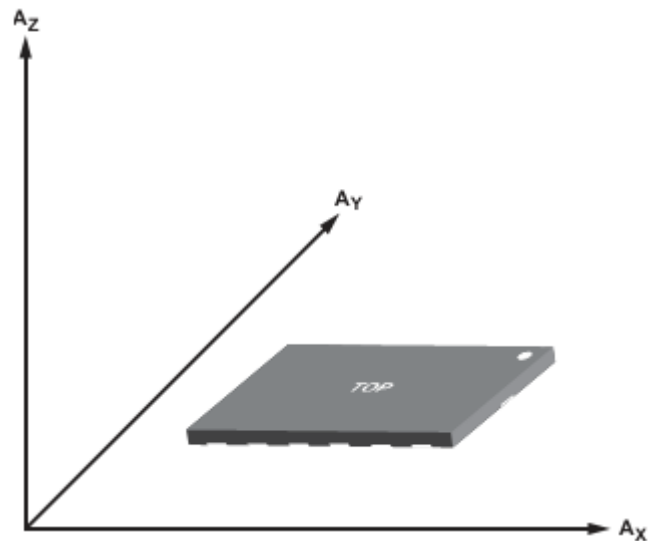
Hình 2.22: Razor IMU 9DOF

### Cảm biến gia tốc ADXL 345 (triple-axis accelerometer)

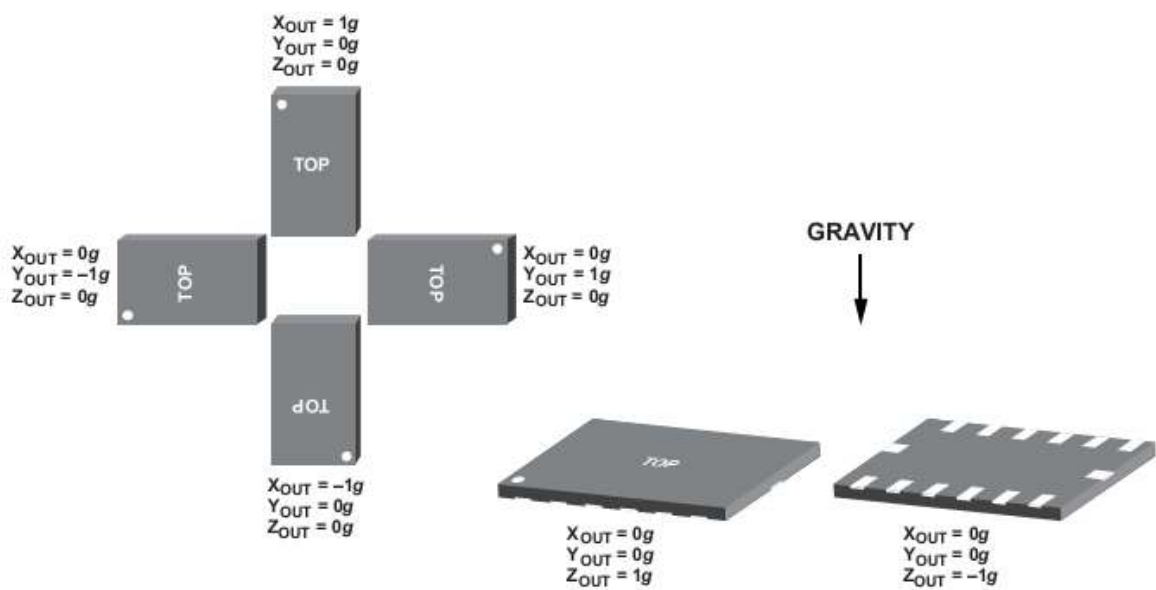
ADXL345 là cảm biến gia tốc 3 trục công suất thấp của hãng Analog Devices thích hợp cho các ứng dụng di động, độ phân giải cao (13bit), với tầm đo  $\pm 16g$  và có ngõ ra theo chuẩn truyền dữ liệu I2C hay SPI tùy chỉnh



Hình 2.23: Sơ đồ khối cảm biến gia tốc ADXL345

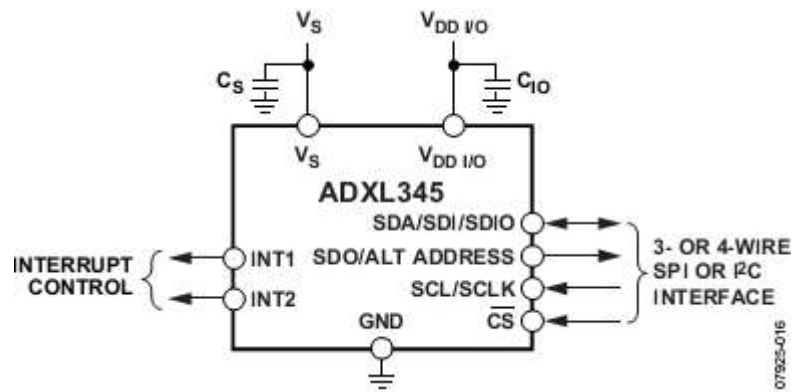


Hình 2.24: Phương và chiều các trục gia tốc tương ứng



Hình 2.25: Đáp ứng giá trị gia tốc ngõ ra tương ứng đối với hướng của trọng lực





Hình 2.26: Sơ đồ chân ngõ ra trên cảm biến gia tốc

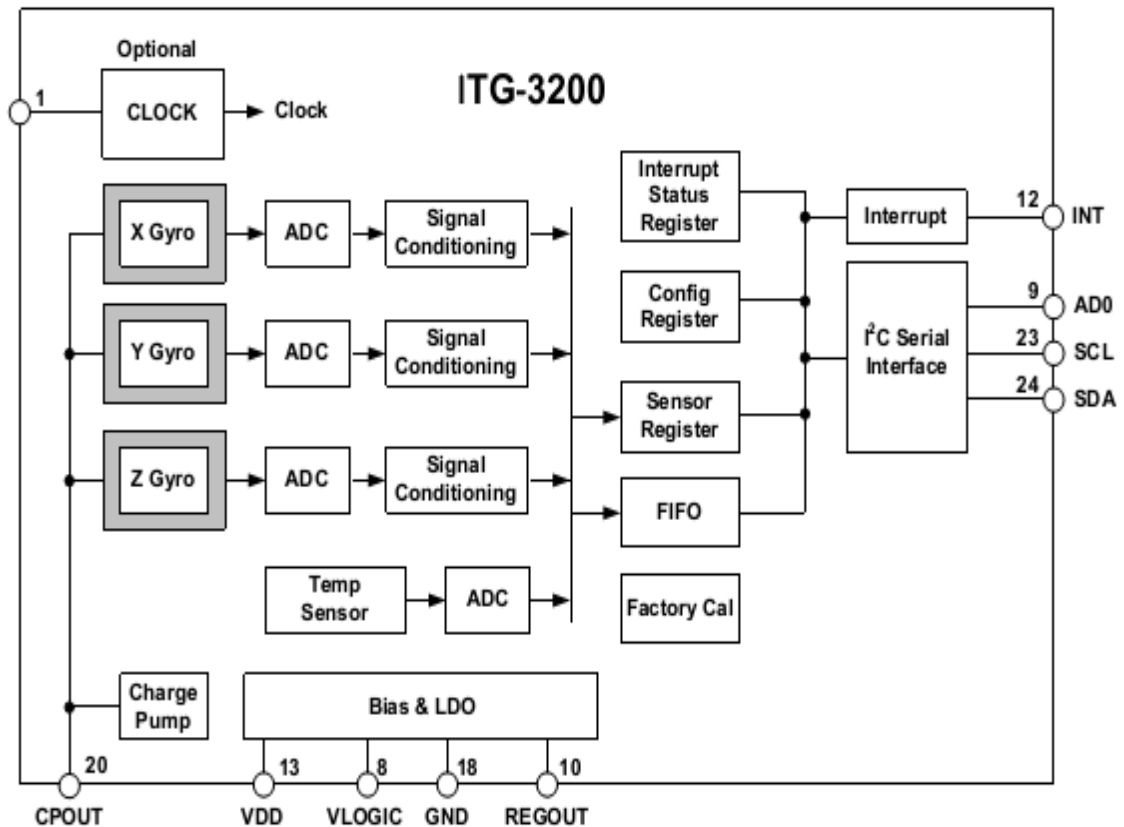
Dựa vào cảm biến gia tốc ADXL345, có thể đo được gia tốc của robot theo 3 phương  $A_x$ ,  $A_y$ ,  $A_z$  và dựa vào gia tốc có thể xác định được độ nghiêng theo công thức sau:

$$\psi = \frac{A_x}{A_y} \quad [2.63]$$

### Cảm biến con quay hồi chuyển ITG3200 (MEMS triple-axis gyro)

ITG3200 là cảm biến con quay hồi chuyển của hãng InvenSense tích hợp 3 bộ ADC 16bit để chuyển đổi ngõ ra analog, tích hợp sẵn bộ khuếch đại và bộ lọc thông thấp, có độ phân giải  $\pm 2000^\circ/s$ . Cảm biến được tích hợp chuẩn truyền dữ liệu I2C.

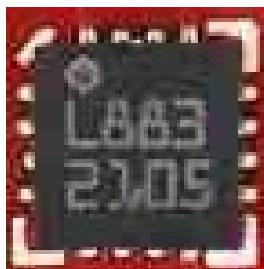
Con quay hồi chuyển được sử dụng để đo vận tốc góc nghiêng của robot. Sự kết hợp 2 cảm biến: gia tốc và con quay hồi chuyển để đo 2 thông số: góc nghiêng tĩnh và vận tốc góc nghiêng của robot.



Hình 2.27: Sơ đồ khối cảm biến ITG3200

### Cảm biến từ trường HMC5883L (triple-axis magnetometer)

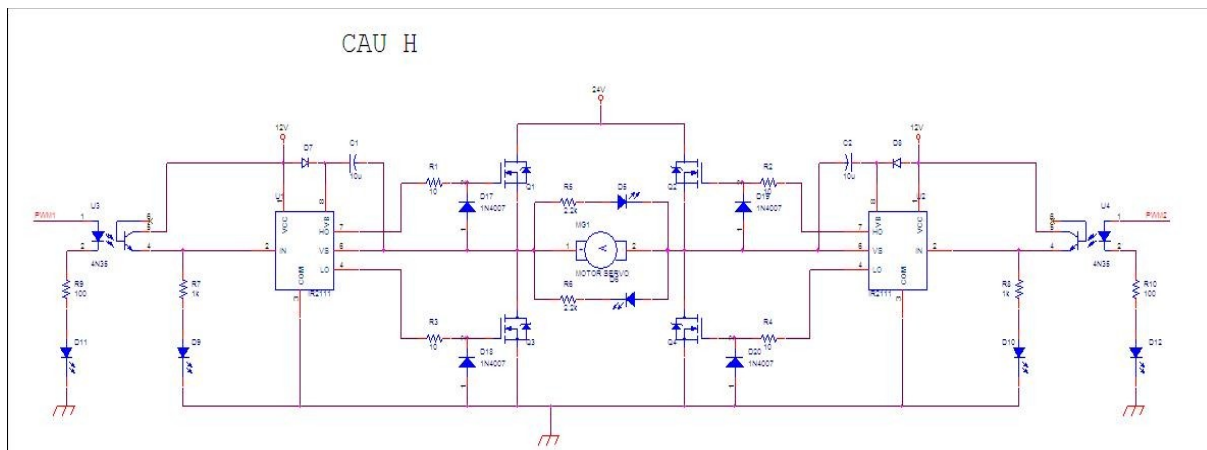
HMC5883L là cảm biến từ trường có độ phân giải  $\pm 80e$  tích hợp truyền dữ liệu I2C.



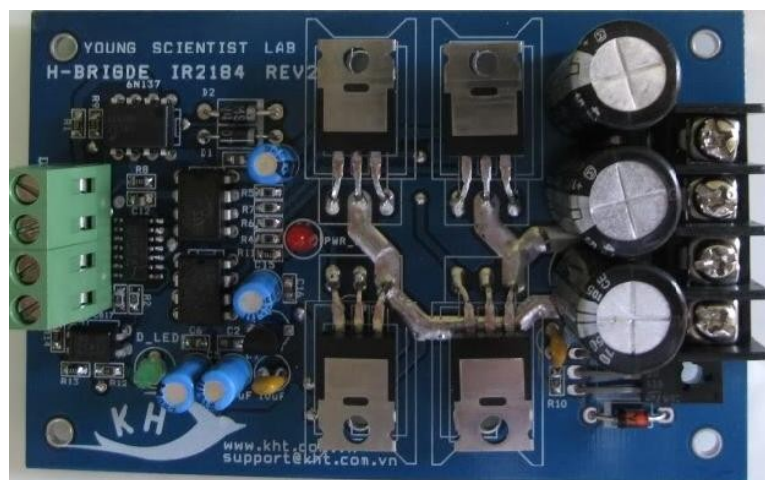
Hình 2.28: Cảm biến HMC5883L

#### 2.3.3.4 Mạch cầu H

Gồm 4 Mosfet có tác dụng đảo chiều động cơ



Hình 2.29: Sơ đồ nguyên lý mạch cầu H



Hình 2.30: Mạch cầu H thực tế

#### 2.3.3.5 Encoder

Encoder hiệu sharp100 xung/vòng gắn trên thân robot, tiếp xúc không trượt với bánh xe robot, tỉ lệ số vòng bánh xe encoder và số vòng bánh xerobot là 2:1 để đo vị trí và vận tốc di chuyển của robot.



Hình 2.31: Encoder

#### 2.3.3.6 Module thu phát, truyền nhận dữ liệu

1 module RF để điều khiển từ xa mọi hoạt động của robot, 1 module bluetooth truyền tín hiệu uart về máy tính để theo dõi và giám sát robot.



Hình 2.32: Module Bluetooth HC06 slave

#### 2.3.3.7 Động cơ

Mô hình sử dụng 2 động cơ có hộp số, 12VDC, tỉ số truyền 30, moment xoắn 2Nm



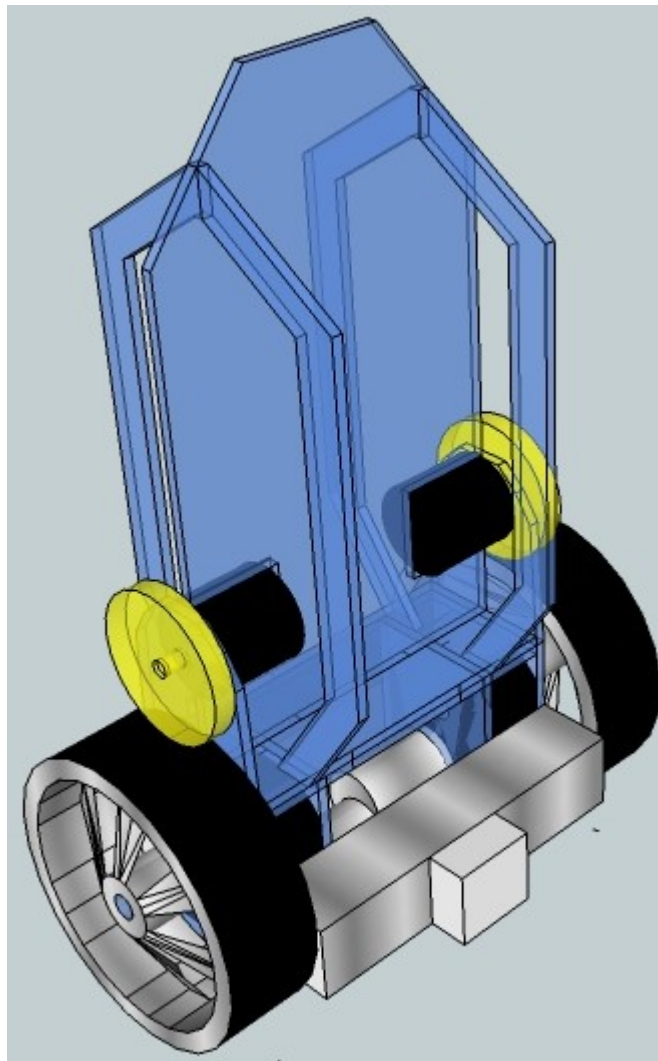
Hình 2.33: Động cơ

## 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 4,5mm, được cố định theo cơ cấu ghép rãnh. Cơ cấu chuyển động encoder áp bánh và bố trí mạch theo chiều dựng đứng.



Hình 3.1: Bản vẽ 3D mô hình robot 2 bánh tự cân bằng

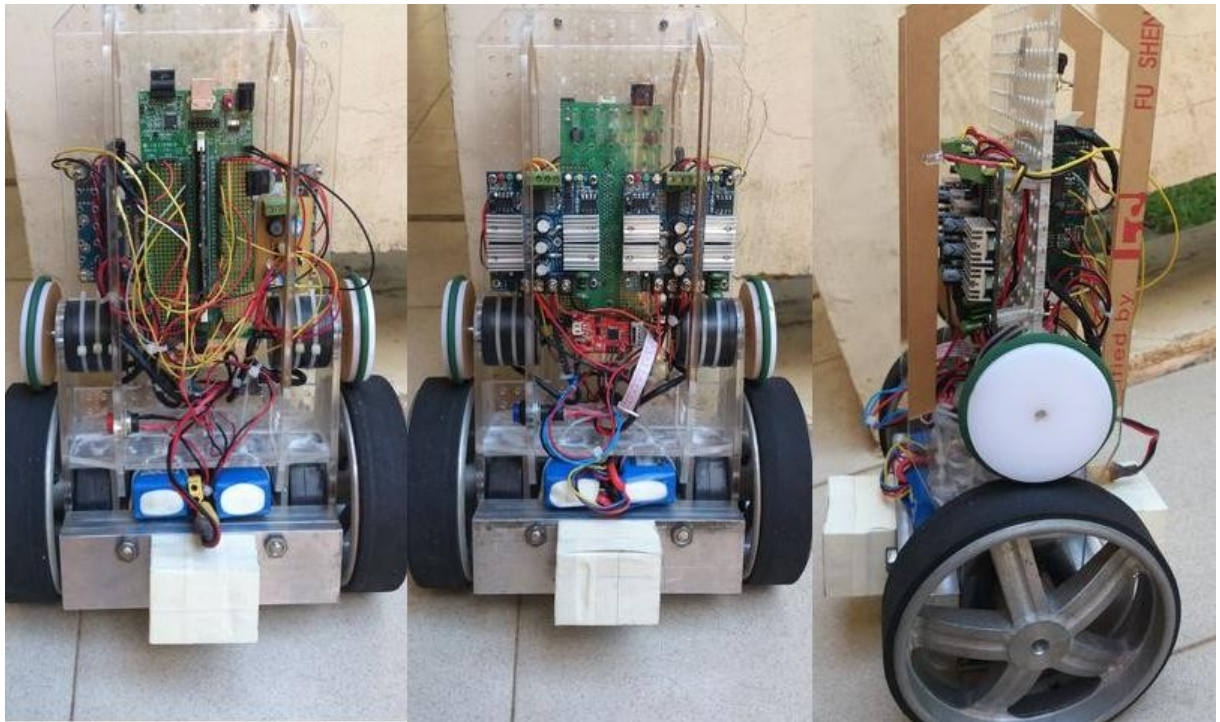
Các giá trị thực tế của mô hình cơ khí được ghi trong bảng sau:

Kí hiệu	Đơn vị	Ý nghĩa	Giá trị
M	kg	Khối lượng của bánh xe	0,4
M	kg	Khối lượng của robot	5
R	m	Bán kính bánh xe	0,0725
W	m	Chiều rộng của robot	0.17
D	m	Chiều ngang của robot	0.08
H	m	Chiều cao của robot	0.3
L	m	Khoảng cách từ trọng tâm robot đến trục bánh xe	0.12
$f_w$		Hệ số ma sát giữa bánh xe và mặt phẳng di chuyển	0.001
$f_m$		Hệ số ma sát giữa robot và động cơ DC	0.001
$J_m$	$kg.m^2$	Moment quán tính của động cơ DC	0.5462
$R_m$	ohm	Điện trở động cơ DC	20
Kb	$\frac{V}{sec/rad}$	Hệ số EMF của động cơ DC	0.002
Kt	Nm/A	Moment xoắn của động cơ DC	4.2
N		Tỉ số giảm tốc	30
G	$m/s^2$	Gia tốc trọng trường	9.81

Bảng 3.1: Bảng giá trị của các đại lượng trong robot 2 bánh tự cân bằng

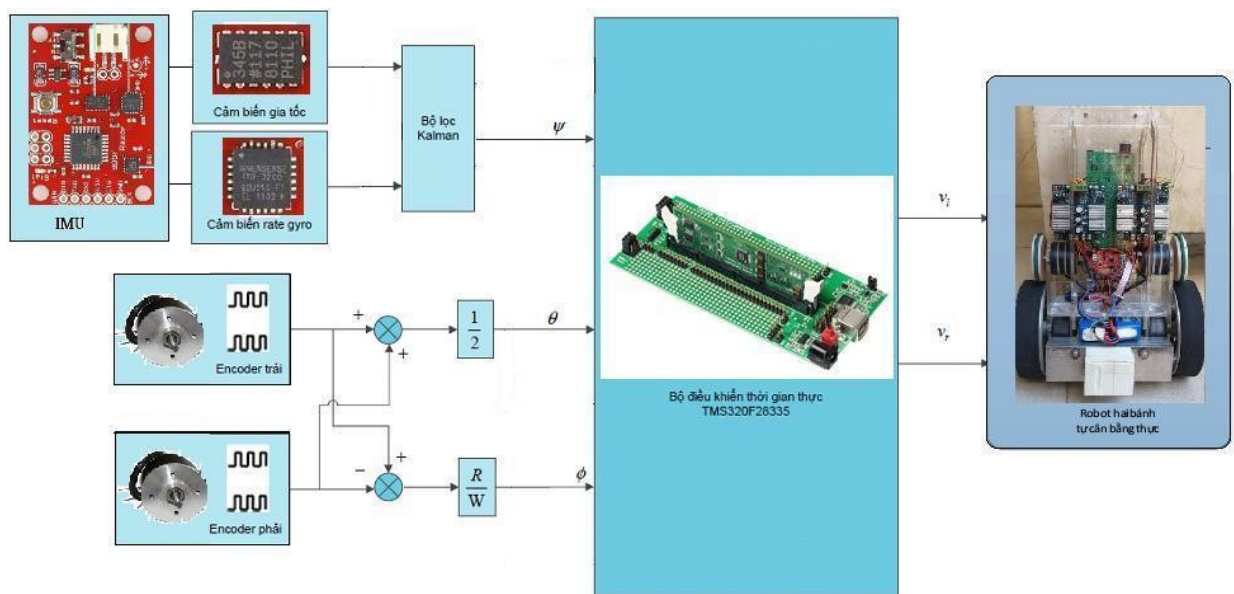
Hình ảnh thực tế của mô hình robot 2 bánh tự cân bằng





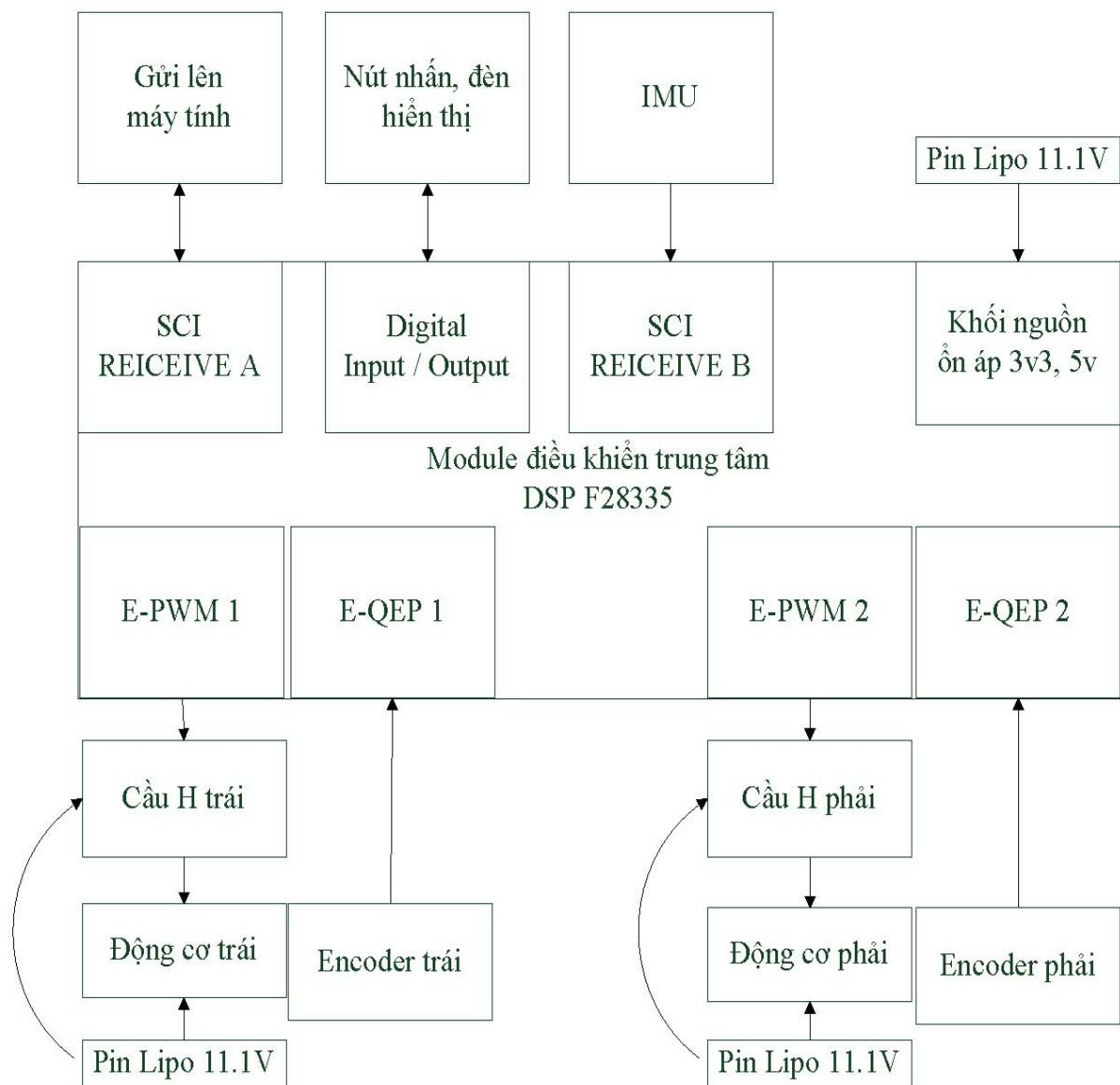
Hình 3.2: Mô hình robot thực tế

### 3.1.2 Cấu trúc điều khiển phản cứng



Hình 3.3: Cấu trúc điều khiển phản cứng của robot 2 bánh tự cân bằng

Sơ đồ kết nối phần cứng của mô hình robot 2 bánh tự cân bằng:



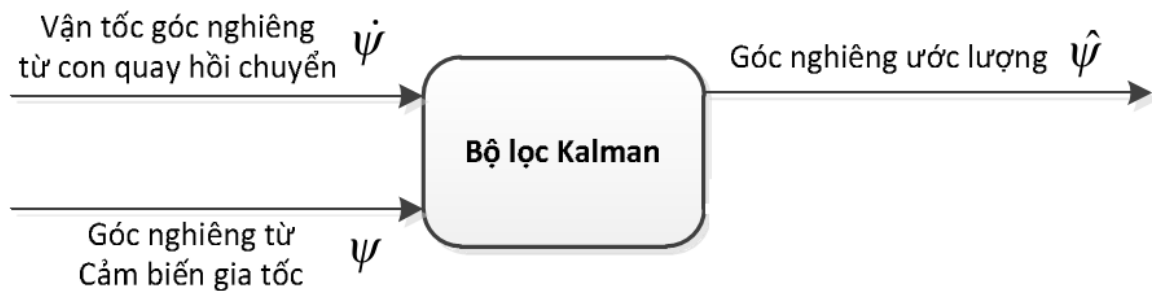
Hình 3.4: Sơ đồ kết nối phần cứng

## 3.2 Thiết kế phần mềm

### 3.2.1 Bộ lọc Kalman cho thành phần IMU

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

### 3.2.1.1 Lập trình bộ lọc Kalman

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:

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

Và cập nhật ma trận covariance qua hàm :

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

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

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

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

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

$$\dot{P} = \begin{bmatrix} Q_{angle} - P[0][1] - P[1][0] & -P[1][1] \\ -P[1][1] & Q_{gyro} \end{bmatrix} \quad [3.9]$$

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

$$\dot{P} = \begin{bmatrix} -P[1][0] & -P[1][1] \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -P[0][1] & 0 \\ -P[1][1] & 0 \end{bmatrix} + \begin{bmatrix} Q_{angle} & 0 \\ 0 & Q_{gyro} \end{bmatrix} \quad [3.10]$$

Lưu trữ giá trị ước lượng chưa bias của gyro :

$$\text{rate} = q = q_m - q_{bias} \quad [3.11]$$

(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 :

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

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

- “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 :

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

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 giá trị tốc độ góc và góc ước lượng.

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

Tính sai số ước lượng. Từ bộ lọc Kalman :

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

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

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

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 :

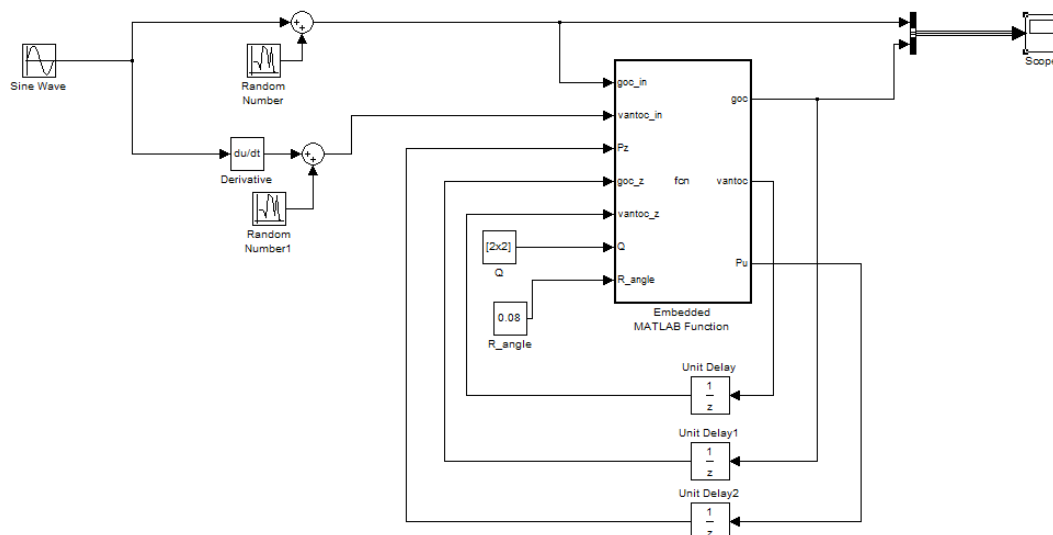
$$X = \begin{bmatrix} angle \\ gyro_{bias} \end{bmatrix} = X + K (Z_{measure} - H X_{estimate}) = X + K \begin{bmatrix} angle_{err} \\ angle_{err} \end{bmatrix} \quad [3.23]$$

$$\begin{matrix} err \\ angle_{err} = q_i \end{matrix} \quad [3.24]$$

### 3.2.1.2 Mô phỏng bộ lọc Kalman đã lập trình

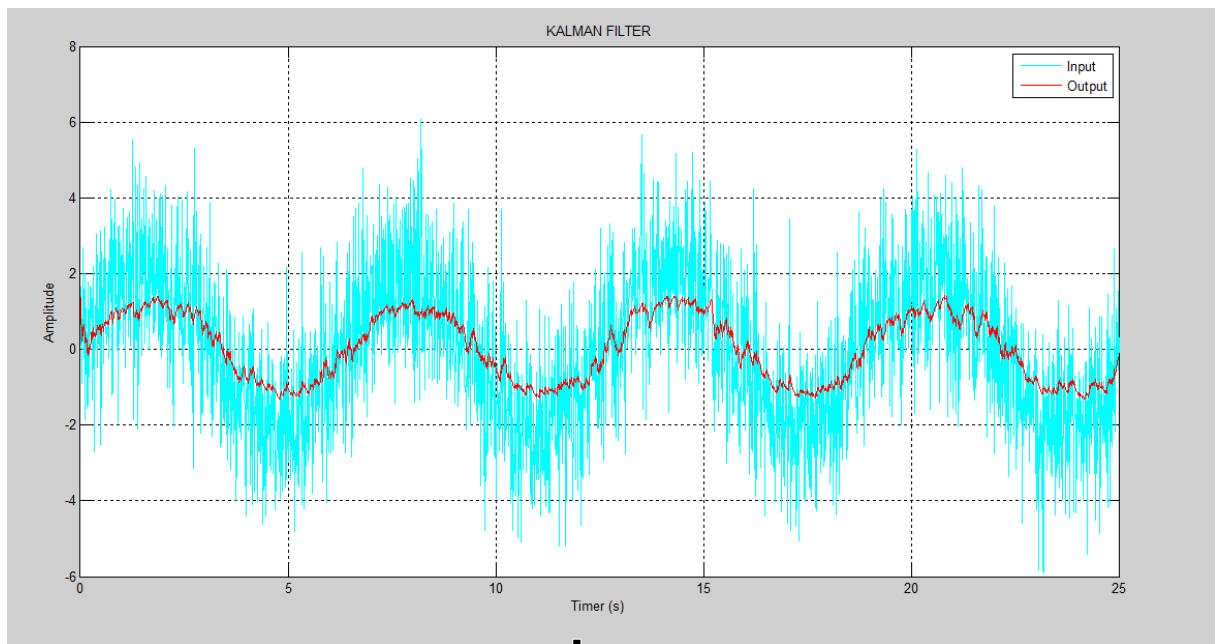
Sử dụng Matlab Simulink để kiểm tra tính đúng đắn của bộ lọc Kalman đã lập trình ở phần 3.2.1.1

Mô hình bộ lọc Kalman trong Matlab Simulink được xây dựng như sau :



Hình 3.6: Mô hình bộ lọc Kalman trong Matlab Simulink

Khi cho nhiều tác động, ngõ ra của bộ lọc Kalman đã lọc nhiễu tốt:



Hình 3.7: Tín hiệu vào-ra của bộ lọc Kalman

### 3.2.1.3 Ứng dụng bộ lọc Kalman vào mô hình thật

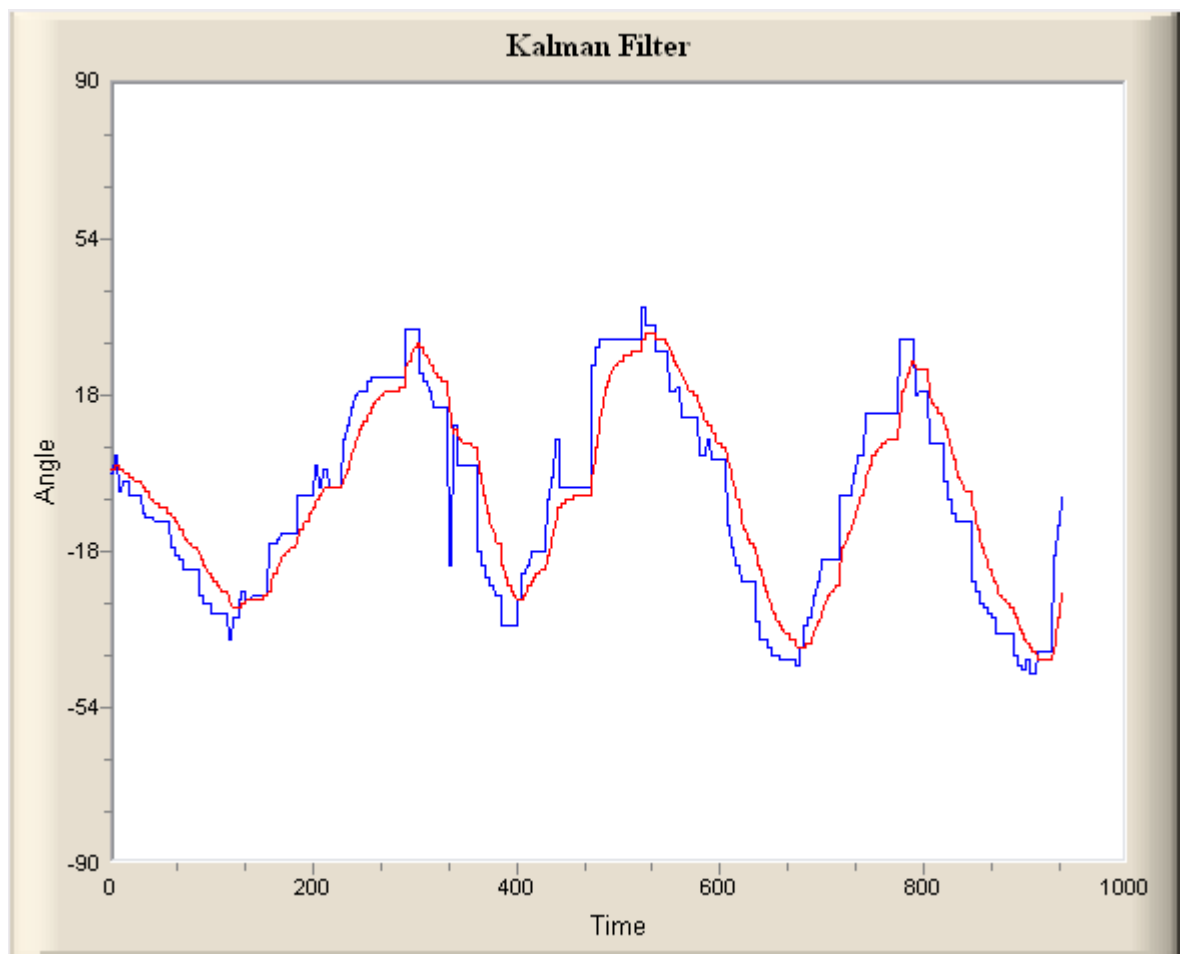
Áp dụng bộ lọc Kalman để ước lượng góc nghiêng so với điểm 0 của mô hình robot 2 bánh tự cân bằng trong thực tế, chu kì lấy mẫu 1ms, ma trận các trọng số được chọn thử sai trong quá trình thực nghiệm để cho kết quả tốt nhất như sau:

$$Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.003 \end{bmatrix};$$

$$R = 0.08$$

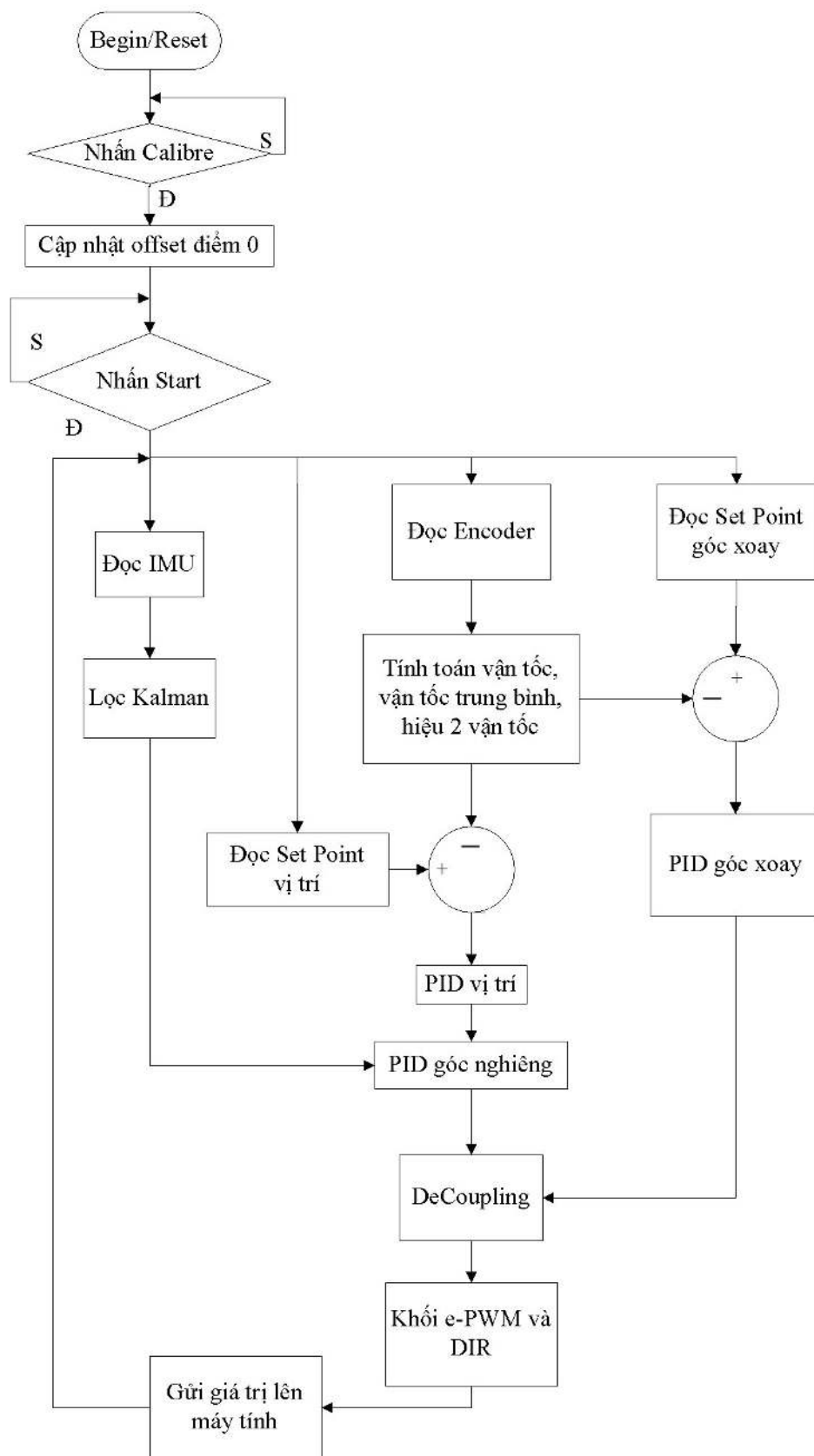
$$\Delta t = 0.064 \text{ s}$$

Kết quả quan sát hệ thống thực của giá trị góc nghiêng đo được từ cảm biến IMU sử dụng trong mô hình



Hình 3.8: So sánh khi cho tín hiệu góc nghiêng ngõ vào và ra của bộ lọc Kalman (đường màu xanh là góc nghiêng chưa qua bộ lọc, đường màu đỏ là tín hiệu góc nghiêng sau khi qua bộ lọc)

### 3.2.2 Lưu đồ giải thuật

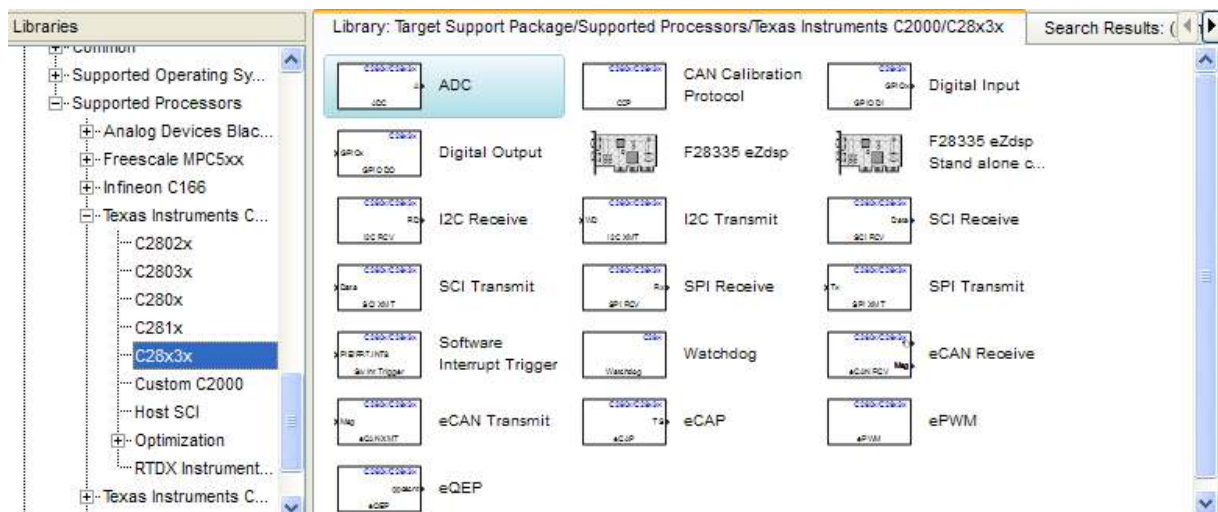


Hình 3.9: Lưu đồ giải thuật cho mô hình robot 2 bánh tự cân bằng

### 3.2.3 Bộ điều khiển nhúng cho robot 2 bánh tự cân bằng

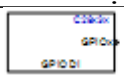
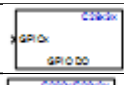
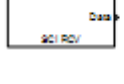
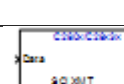

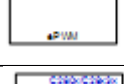
#### 3.2.3.1 Giới thiệu

Quá trình điều khiển robot được thực hiện trên vi điều khiển DSP TMS320F28335 với sự hỗ trợ của thư viện Target Support Package của Matlab –hỗ trợ cho các vi điều khiển họ TI C2000. Bộ điều khiển được thiết kế trên Simulink trong miền rời rạc, sau đó các ứng dụng được Embedded IDE Link biên dịch sang mã ngôn ngữ C tương thích với thư viện hỗ trợ bởi Code Composer, sau đó tiếp tục được Code Composer biên dịch thành mã máy nhúng vào DSP TMS320F28335.



Hình 3.10 : Thư viện Target Support Package cho DSP F28335

#### 3.2.3.2 Các khối chức năng sử dụng trong chương trình

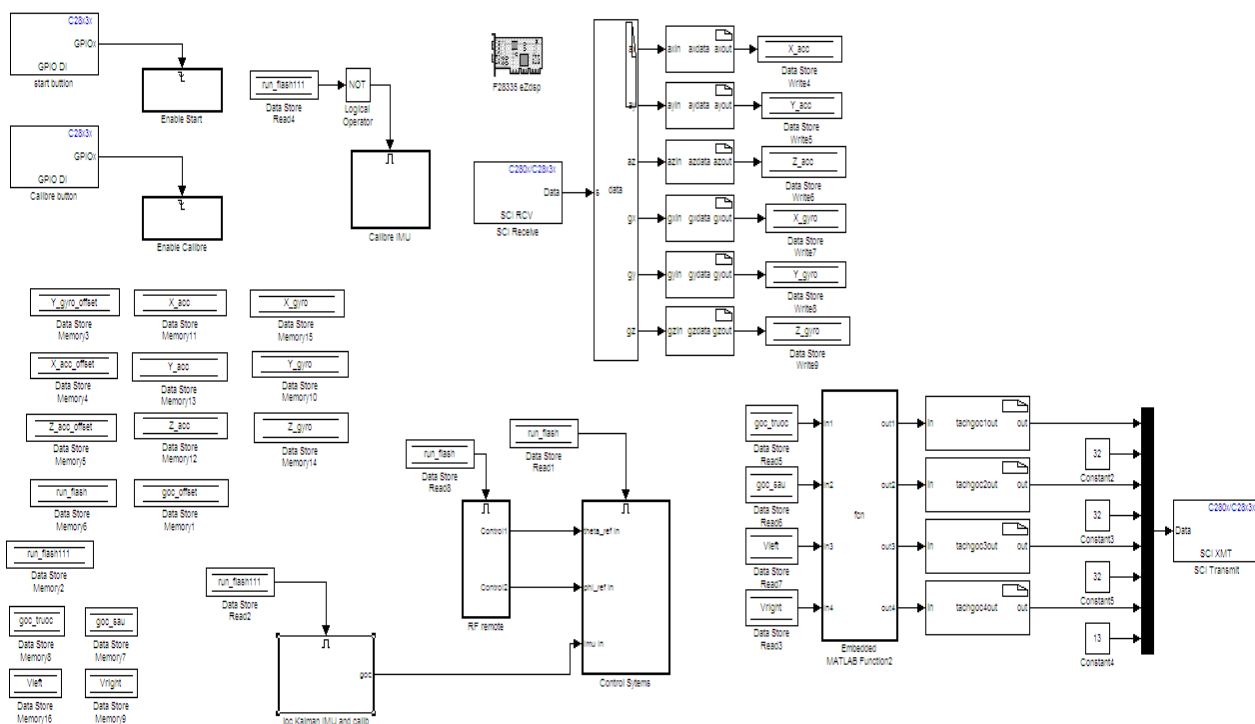
Biểu tượng	Chức năng
 Digital Input	Nhận tín hiệu digital vào: nút start, stop, calibrate
 Digital Output	Xuất tín hiệu digital: các đèn hiển thị
 SCI Receive	Nhận tín hiệu truyền thông UART: đọc cảm biến IMU, nhận lệnh từ module bluetooth HC06
 SCI Transmit	Xuất tín hiệu truyền thông UART: gửi các giá trị về máy tính để theo dõi, giám sát
 ePWM	Điều chế độ rộng xung, đảo chiều động cơ
 eQEP	Đọc encoder

Bảng 3.2: Các khối chức năng sử dụng trong chương trình



### 3.2.3.3 Chương trình chính trên Matlab Simulink

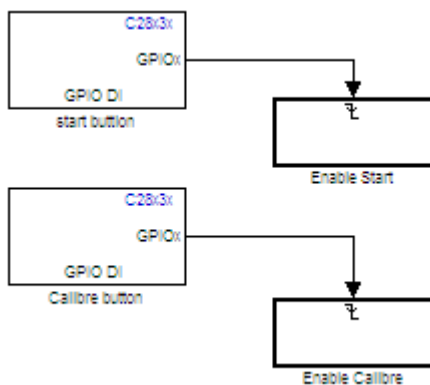
## Tổng quát bộ điều khiển của mô hình robot 2 bánh tự cân bằng trên Matlab Simulink



Hình 3.11: Sơ đồ tổng quát

Bộ điều khiển bao gồm các khối sau:

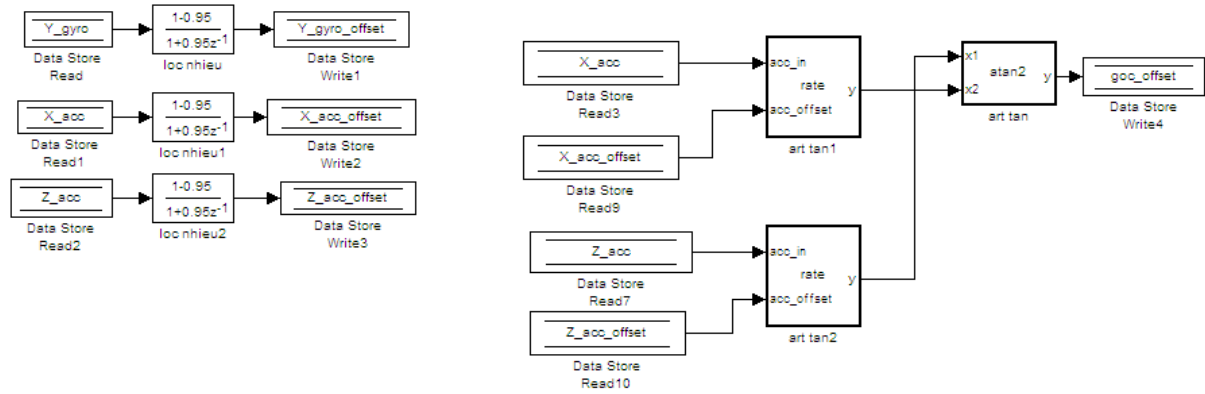
Khởi đọc các nút nhấn: Start, Calibre



Hình 3.12: Khối đọc các nút nhấn

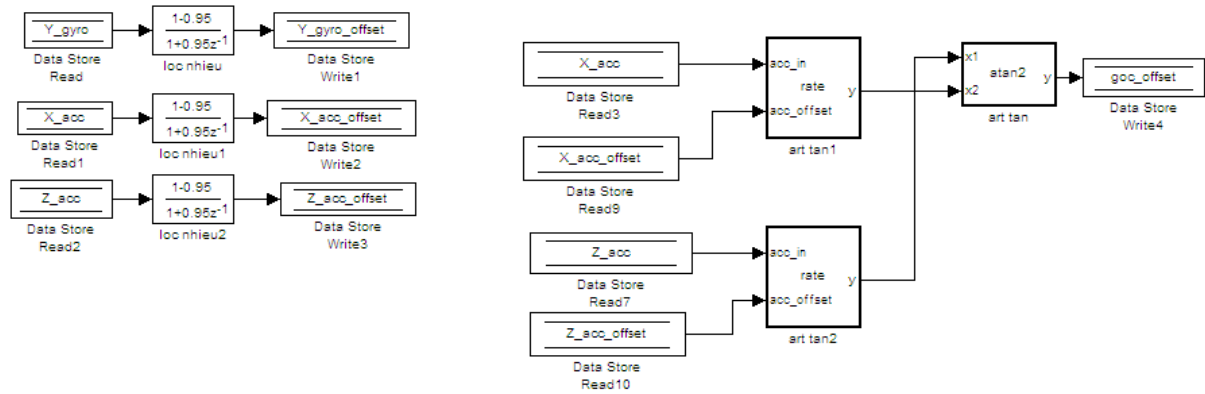
Khối Calibre: khi được cấp điện, robot chưa hoạt động thì sẽ liên tục cập nhật giá

trị offset



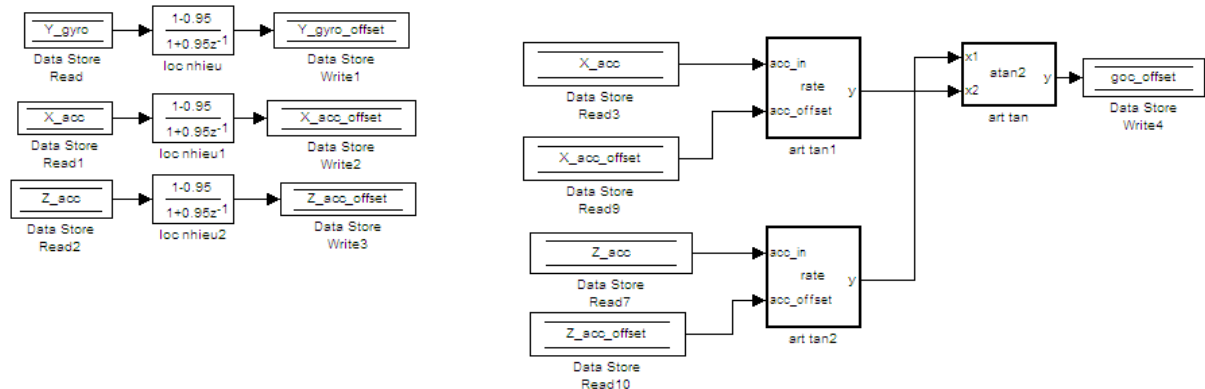
Hình 3.13: Bên trong khối calibre

Khối đọc cảm biến IMU: đọc và phân tách các giá trị của cảm biến IMU



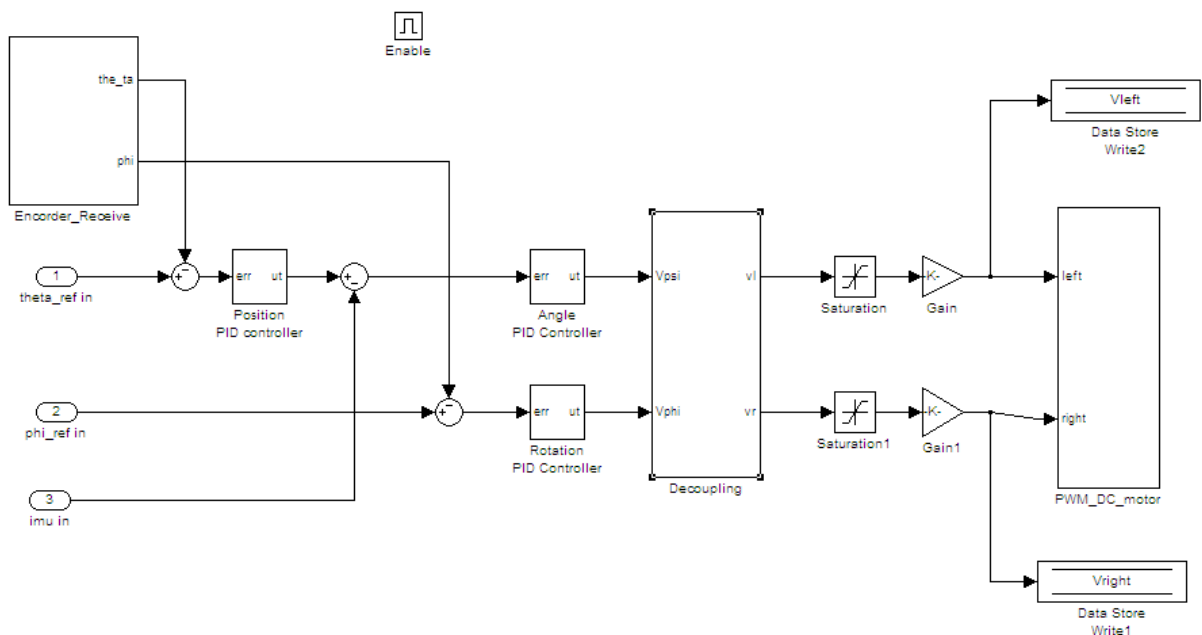
Hình 3.14: Khối đọc cảm biến IMU

Khối lọc Kalman: lấy thành phần gia tốc từ cảm biến gia tốc và tính góc xoay alpha kết hợp với vận tốc góc từ 1 trục đưa vào bộ lọc Kalman để cho ra góc nghiêng đã được loại bỏ nhiễu.



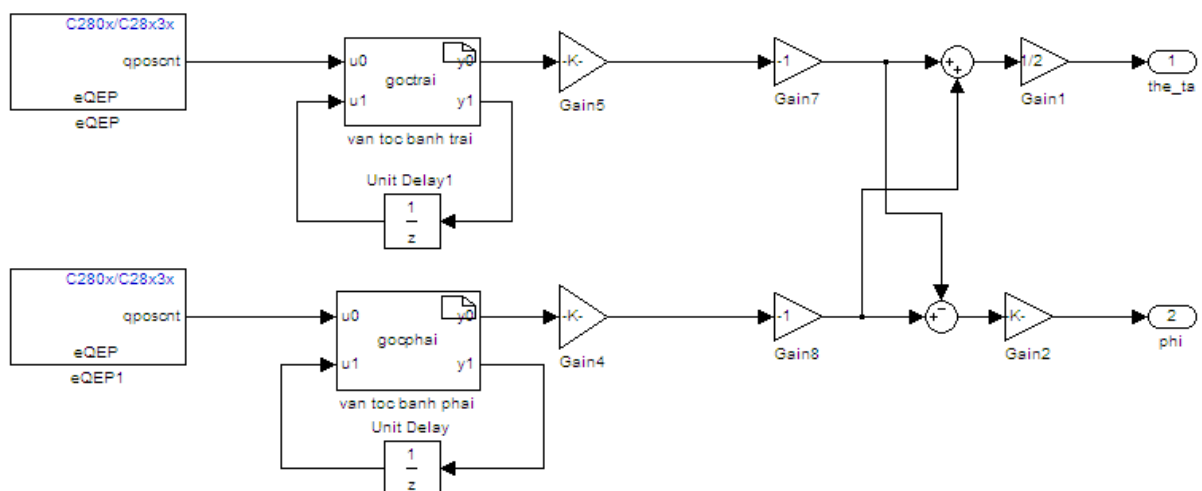
Hình 3.15: Khối lọc Kalman

Khối điều khiển trung tâm : Điều khiển các khâu vị trí, khâu góc xoay, khâu góc nghiêng.



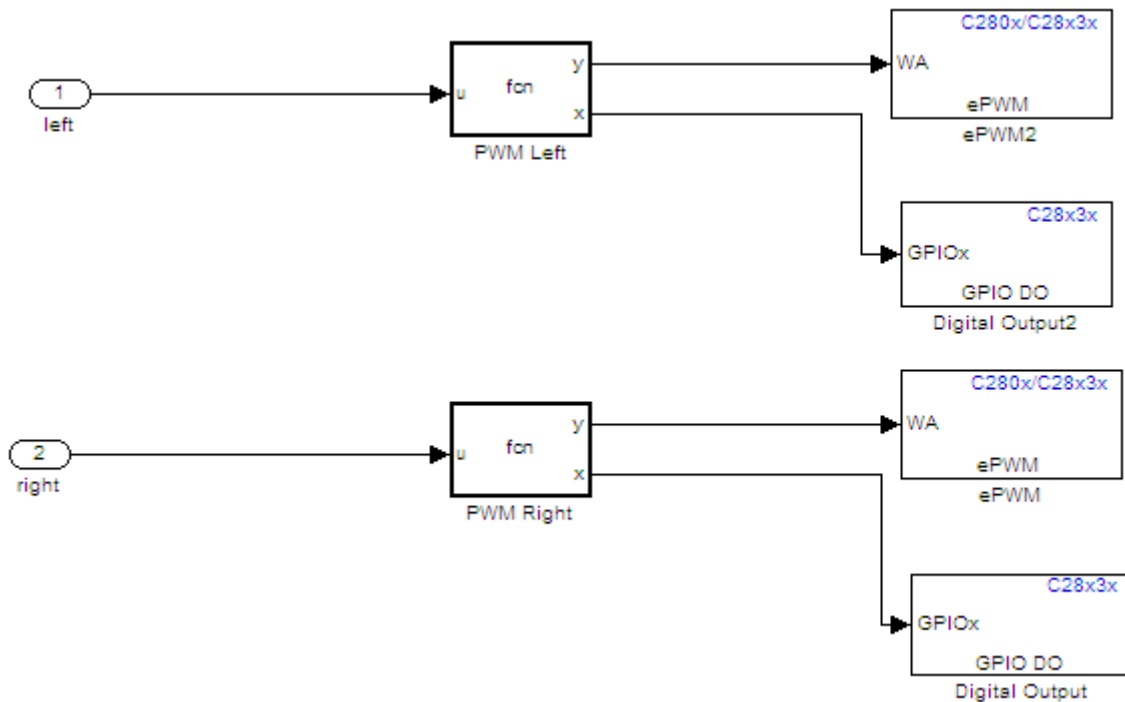
Hình 3.16: Tổng quát Khối điều khiển trung tâm

Khối đọc xung encoder: đọc xung từ encoder và tính toán các giá trị cần thiết



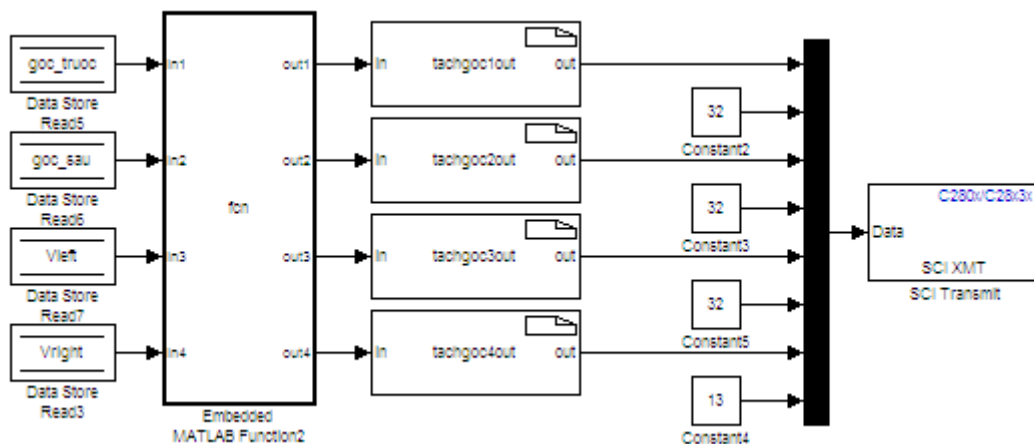
Hình 3.17: Bên trong Khối đọc xung encoder

Khối xuất tín hiệu điều khiển: gồm xuất xung PWM và DIR cho động cơ trái phải



Hình 3.18: Bên trong Khối xuất tín hiệu điều khiển

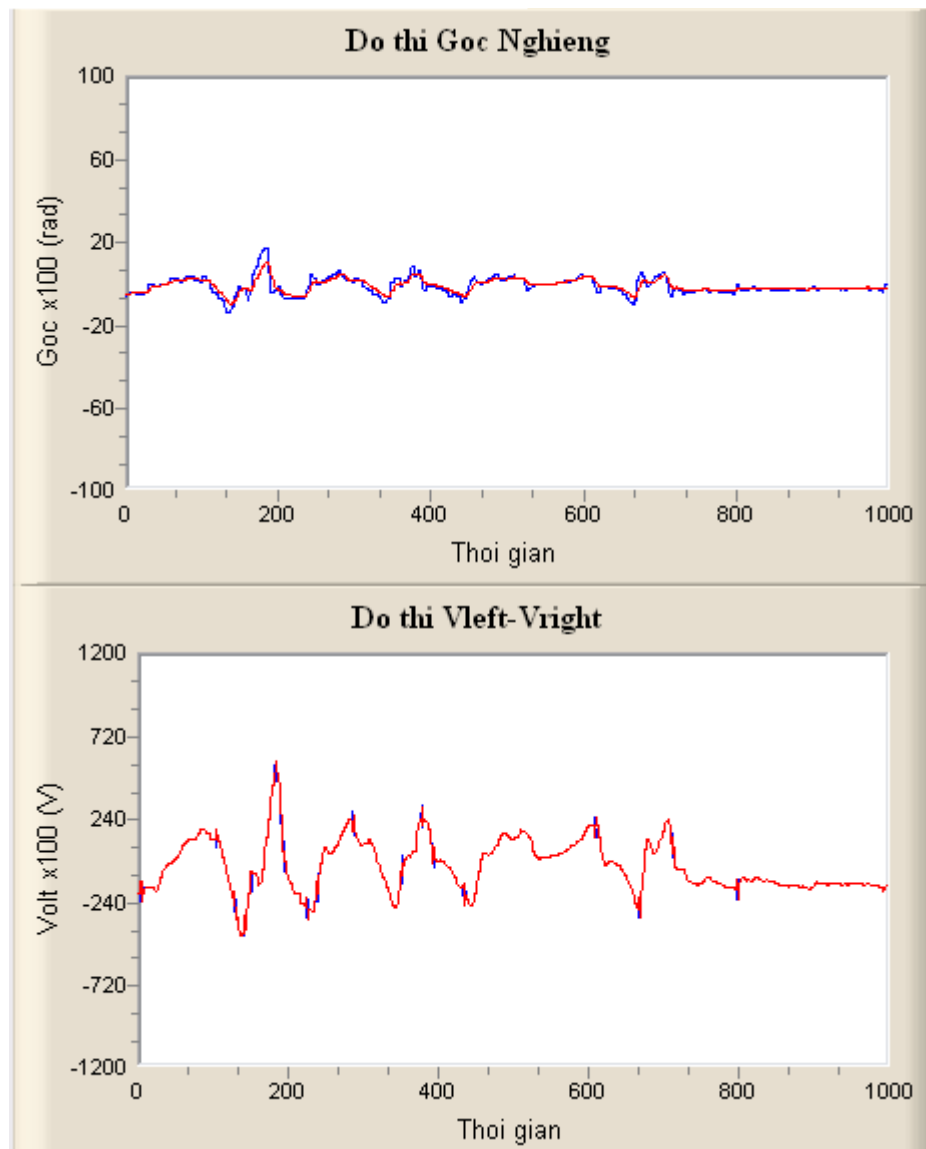
Khối gửi dữ liệu về máy tính: gửi các thông số từ bộ điều khiển lên máy tính phục vụ giao diện và giám sát, canh chỉnh PID



Hình 3.19: Khối gửi dữ liệu

### 3.2.3.4 Kết quả thu được trên mô hình thực tế:

Với bộ điều khiển PID thông số cố định, robot đã giữ thăng bằng tại chỗ, di chuyển tiến lùi với tốc độ chậm trên địa hình bằng phẳng với khả năng đáp ứng tác động của ngoại lực biên độ nhỏ.



Hình 3.20: Ngõ ra góc nghiêng và điện áp trên 2 động cơ khi chạy mô hình thực tế

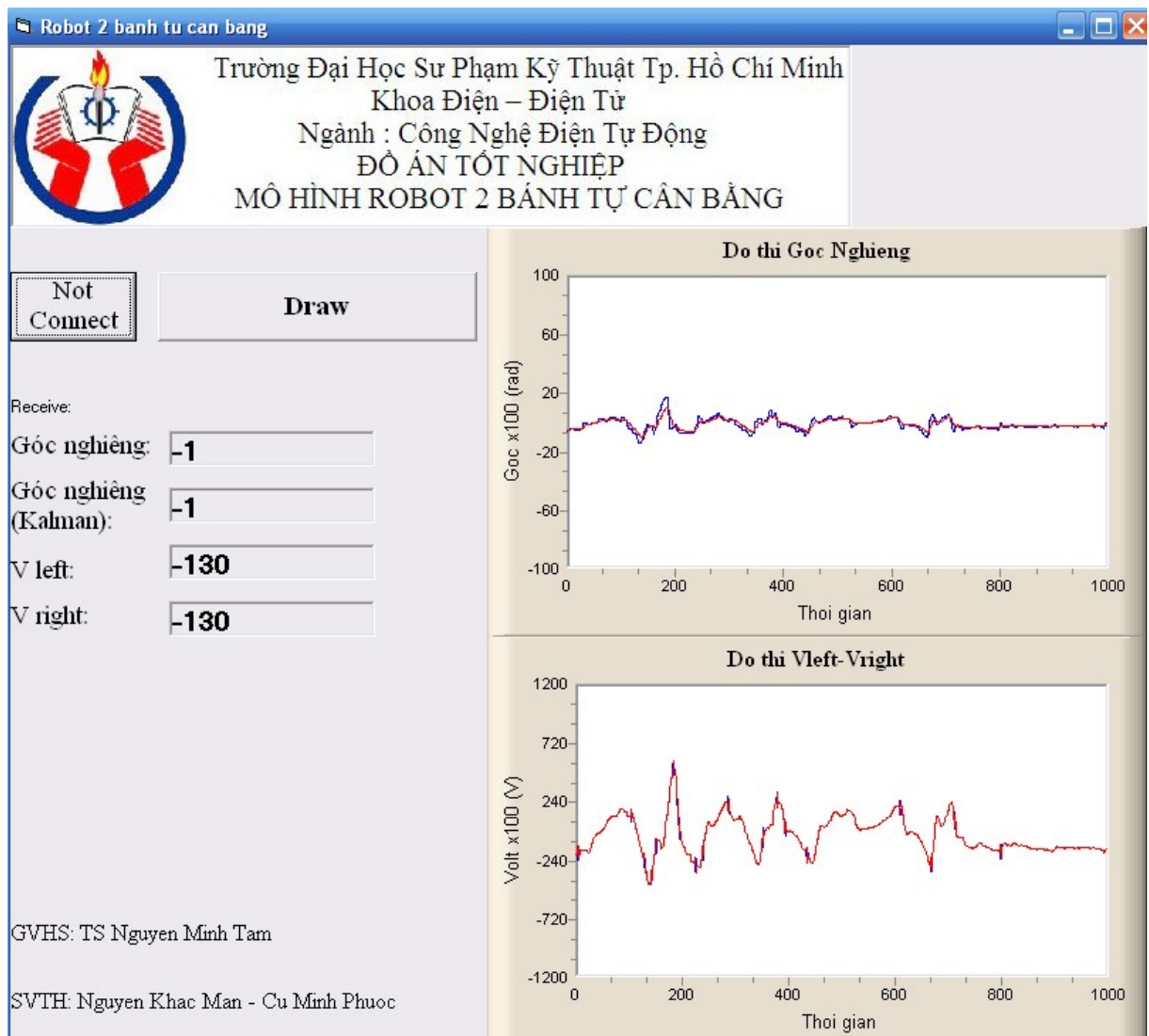
### 3.2.4 Giao diện trên máy tính

Trên mô hình có sử dụng module bluetooth để truyền dữ liệu không dây các giá trị gửi về từ robot, từ đó vẽ biểu đồ so sánh nhằm đối chiếu, đánh giá kết quả thực hiện.

Giao diện trên máy tính được xây dựng bằng phần mềm Visual Basic 6.0, có sử dụng thêm thư viện MSCOMM32.OCX để giao tiếp với cổng COM và thư viện NTGRAPH.OCX để vẽ biểu đồ.

Phần mềm giao diện đơn giản dễ sử dụng, có 2 nút để điều khiển phần mềm:

- Connect: Thiết lập kết nối đến cổng com, sẵn sàng để nhận giá trị từ robot gửi về
- Draw: Vẽ biểu đồ theo thời gian



Hình 3.21: Giao diện của chương trình giao tiếp máy tính

## 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
- Giao tiếp với máy tính bằng UART và vẽ biểu đồ hoạt động của robot
- 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.

## PHỤ LỤC

### 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.

Nguy ễn 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/>