

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

NGUYỄN ĐỊNH

CAO PHƯƠNG HOÀNG

**ĐỒ ÁN TỐT NGHIỆP**

**THIẾT KẾ VÀ ĐIỀU KHIỂN MÔ HÌNH ROBOT NHẸN 4 CHÂN**

**CỦ NHÂN NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**TP. HỒ CHÍ MINH, 2024**

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

NGUYỄN ĐỊNH – 2011092

CAO PHƯƠNG HOÀNG – 2011220

ĐỒ ÁN TỐT NGHIỆP

THIẾT KẾ VÀ ĐIỀU KHIỂN MÔ HÌNH ROBOT NHỆN  
4 CHÂN

(DESIGN AND CONTROL A QUADRUPED SPIDERBOT)

CỦ NHÂN NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

GIẢNG VIÊN HƯỚNG DẪN

BÙI THANH HUYỀN

TP. HỒ CHÍ MINH, 2024

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI  
TRƯỜNG ĐẠI HỌC BÁCH KHOA –ĐHQG -HCM**

Cán bộ hướng dẫn Khóa luận tốt nghiệp : .....  
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 1 : .....  
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 2 : .....  
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Khóa luận tốt nghiệp được bảo vệ tại Trường Đại học Bách Khoa, ĐHQG  
Tp.HCM ngày .... tháng .... năm ....

Thành phần Hội đồng đánh giá khóa luận tốt nghiệp gồm:  
(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ khóa luận tốt nghiệp)

1. .....
2. .....
3. .....
4. .....
5. .....

Xác nhận của Chủ tịch Hội đồng đánh giá khóa luận tốt nghiệp và Chủ nhiệm Bộ  
môn sau khi đồ án đã được sửa chữa (nếu có).

**CHỦ TỊCH HỘI ĐỒNG**

**CHỦ NHIỆM BỘ MÔN.....**

TP. HCM, ngày .... tháng ..... năm .....

NHẬN XÉT ĐÔ ÁN TỐT NGHIỆP

CỦA CÁN BỘ HƯỚNG DẪN

Tên Đồ án:

**THIẾT KẾ VÀ ĐIỀU KHIỂN MÔ HÌNH ROBOT NHỆN 4 CHÂN**

**(DESIGN AND CONTROL A QUADRUPED SPIDERBOT)**

Nhóm Sinh viên thực hiện:

Nguyễn Định

Cao Phượng Hoàng

Cán bộ hướng dẫn:

2011092 Bùi Thanh Huyền

2011220 Bùi Thanh Huyền

Đánh giá Đồ án

1. Về cuốn báo cáo:

Số trang

\_\_\_\_\_

Số chương

\_\_\_\_\_

Số bảng số liệu

\_\_\_\_\_

Số hình vẽ

\_\_\_\_\_

Số tài liệu tham khảo

\_\_\_\_\_

Sản phẩm

\_\_\_\_\_

Một số nhận xét về hình thức cuốn báo cáo:

<nhận xét về định dạng, cách thức viết báo cáo, phân bố nội dung, chương mục có hợp lý không, ...>

2. Về nội dung đồ án:

<nhận xét về kiến thức, phương pháp mà sinh viên đã tìm hiểu, nghiên cứu, nhận xét ưu điểm và hạn chế>

3. Về tính ứng dụng:

*<nhận xét về việc xây dựng ứng dụng demo, nhận xét ưu điểm và hạn chế>*

4. Về thái độ làm việc của sinh viên:

*<nhận xét về thái độ, ưu khuyết điểm của từng sinh viên tham gia>*

**Đánh giá chung:** Đồ án đạt/không đạt yêu cầu của một đồ án tốt nghiệp cử nhân, xếp loại Giỏi/Khá/ Trung bình

**Điểm từng sinh viên:**

Nguyễn Định:...../10

Cao Phượng Hoàng:...../10

**Cán bộ hướng dẫn**  
(Ký tên và ghi rõ họ tên)

TP. HCM, ngày.....tháng.....năm.....

**NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP**

**CỦA CÁN BỘ PHẢN BIỆN**

**Tên Đồ án**

**THIẾT KẾ VÀ ĐIỀU KHIỂN MÔ HÌNH ROBOT NHỆN 4 CHÂN**

**(DESIGN AND CONTROL A QUADRUPED SPIDERBOT)**

**Nhóm Sinh viên thực hiện:**

Nguyễn Định

2011092 <Họ tên>

Cao Phượng Hoàng

2011220 <Họ tên>

**Cán bộ phản biện:**

**Đánh giá Đồ án**

5. Về cuốn báo cáo:

Số trang

\_\_\_\_\_

Số chương

\_\_\_\_\_

Số bảng số liệu

\_\_\_\_\_

Số hình vẽ

\_\_\_\_\_

Số tài liệu tham khảo

\_\_\_\_\_

Sản phẩm

\_\_\_\_\_

Một số nhận xét về hình thức cuốn báo cáo:

<nhận xét về định dạng, cách thức viết báo cáo, phân bố nội dung, chương mục có hợp lý không, ...>

6. Về nội dung đồ án

<nhận xét về kiến thức, phương pháp mà sinh viên đã tìm hiểu, nghiên cứu, nhận xét ưu điểm và hạn chế>

7. Về tính ứng dụng:

*<nhận xét về việc xây dựng ứng dụng demo, nhận xét ưu điểm và hạn chế>*

8. Về thái độ làm việc của sinh viên:

*<nhận xét về thái độ, ưu khuyết điểm của từng sinh viên tham gia>*

**Đánh giá chung:** Đồ án đạt/không đạt yêu cầu của một đồ án tốt nghiệp cử nhân, xếp loại Giỏi/Khá/ Trung bình

**Điểm từng sinh viên:**

Nguyễn Định:...../10

Cao Phượng Hoàng:...../10

**Người nhận xét**  
(Ký tên và ghi rõ họ tên)

TP. HCM, ngày .... tháng ..... năm .....

**ĐỀ CƯƠNG CHI TIẾT**

**TÊN ĐỒ ÁN: THIẾT KẾ VÀ ĐIỀU KHIỂN MÔ HÌNH ROBOT NHỆN 4 CHÂN**

**Cán bộ hướng dẫn:** Bùi Thanh Huyền

**Thời gian thực hiện:** Từ ngày 08/01/2024 đến ngày 10/05/2024

**Sinh viên thực hiện:**

Nguyễn Định – 2011092

Cao Phượng Hoàng – 2011220

**Nội dung đề tài:** (Mô tả chi tiết mục tiêu, phạm vi, đối tượng, phương pháp thực hiện, kết quả mong đợi của đề tài)

**Mục tiêu:**

Thiết kế một mô hình robot nhện 4 chân hoàn chỉnh. Điều khiển robot thực hiện được những chức năng cơ bản. Có thể vận chuyển và thu thập hình ảnh từ xa. Xây dựng bộ điều khiển PID cho việc giữ thăng bằng robot.

Đề xuất việc kết hợp cảm biến IMU và máy ảnh để ước lượng góc hướng tốt hơn.

Tối ưu tốc độ xử lý, giảm thiểu sai số của vị trí và hướng.

Xây dựng một giao diện người dùng thân thiện, dễ sử dụng.

Phân tích kết quả mô phỏng và thực nghiệm

**Đối tượng:**

Đối tượng của Đồ án Tốt nghiệp là một Robot Nhện 4 chân Di động với 3 khớp truyền động mỗi chân, được trang bị một IMU và một camera để xác định hướng, góc và thu thập hình ảnh. Robot Nhện 4 chân được điều khiển bởi GUI máy tính thông qua truyền nhận không dây RF.

### **Phương pháp thực hiện:**

- Tìm hiểu về thiết kế cơ học của mô hình robot, lựa chọn công cụ để thiết kế.
- Lựa chọn các linh kiện liên quan: động cơ, mạch điều khiển, module giao tiếp, ...
- Tính toán công suất tiêu thụ, lựa chọn pin.
- Thiết kế mạch điện kết nối các linh kiện và pin cho robot có thể hoạt động.
- Lắp ráp, hoàn thiện ô hình phần cứng, hiệu chỉnh động cơ, module phù hợp với mục tiêu điều khiển.
- Tìm hiểu lý thuyết điều khiển robot: động học thuận động học ngược. Áp dụng và xây dựng bộ điều khiển robot
- Tìm hiểu về dáng đi của một robot 4 chân, áp dụng dáng đi này để robot có thể di chuyển được. Hiệu chỉnh và tối ưu hóa các thông số để robot có thể di chuyển một cách trơn tru nhất.
- Dựa và những cơ sở trên và việc đọc dữ liệu cảm biến, điều khiển vòng kín cho robot thực hiện các chức năng như chân bằng thân, di chuyển theo hướng chỉ định

### **Kết quả:**

- Xây dựng thành công mô hình phần cứng và điều khiển được robot hoạt động
- Điều khiển vòng kín cho robot cân bằng và di chuyển đúng khoảng cách đã chọn
- Lập trình giao diện giao tiếp và điều khiển robot từ xa.

**Kế hoạch thực hiện:** (*Mô tả kế hoạch làm việc và phân công công việc cho từng sinh viên tham gia*)

Cao Phượng Hoàng	Nguyễn Định
<b>Nghiên cứu về robot nhện bốn chân</b> <ul style="list-style-type: none"><li>• Tìm hiểu dáng đi trong tự nhiên, mô phỏng</li><li>• Tìm và chọn mô hình thiết kế phù hợp</li></ul>	<b>Nghiên cứu về IMU</b> <ul style="list-style-type: none"><li>• Tìm hiểu về MPU9250 bao gồm các cảm biến (gia tốc, con quay, từ kế ) có trong nó</li></ul>

<ul style="list-style-type: none"> <li>Xác định các chức năng robot có thể thực hiện</li> </ul> <p><b>Tính toán động học Robot</b></p> <ul style="list-style-type: none"> <li>Tính toán động học thuận</li> <li>Tính toán động học nghịch</li> <li>Xây dựng chương trình điều khiển từng chân robot</li> </ul> <p><b>Triển khai điều khiển Robot</b></p> <ul style="list-style-type: none"> <li>Hiệu chỉnh đồng bộ servo</li> <li>Viết chương trình điều khiển cho robot từ mô hình động học</li> </ul> <p><b>Tìm hiểu module Remote Control</b></p> <ul style="list-style-type: none"> <li>Tìm hiểu về NRF24L01</li> <li>Viết chương trình truyền nhận dữ liệu giữa hai NRF24L01</li> </ul> <p><b>Tìm hiểu về RTOS</b></p> <ul style="list-style-type: none"> <li>Tìm hiểu về hệ điều hành thời gian thực và ứng dụng vào lập trình multi threads cho STM32</li> </ul>	<ul style="list-style-type: none"> <li>Tìm hiểu các xác định góc và hướng từ IMU</li> </ul> <p><b>Triển khai kỹ thuật để thu góc và hướng từ IMU</b></p> <ul style="list-style-type: none"> <li>Tìm hiểu về AHRS (Attitude and heading reference system)</li> <li>Tìm hiểu về hệ tọa độ Quaternion. Mối liên hệ giữa hệ tọa độ Euler và Quaternion</li> <li>Cách chuyển đổi từ Quaternion sang ma trận các góc xoay</li> <li>Bộ lọc Madgwick</li> <li>Viết chương trình đọc IMU</li> </ul> <p><b>Thiết kế mạch điện cho robot</b></p> <ul style="list-style-type: none"> <li>Thiết kế sơ đồ kết nối</li> <li>Tính toán công suất sử dụng</li> <li>Lựa chọn linh kiện phù hợp</li> <li>Thiết kế mạch Schematic, PCB bằng Altium</li> </ul> <p><b>Xây dựng giao diện điều khiển robot (GUI)</b></p> <ul style="list-style-type: none"> <li>Tìm hiểu và làm quen với PythonQt và Qt Designer</li> <li>Thiết kế giao diện GUI bằng PythonQt</li> <li>Thiết kế luồng và Frame truyền dữ liệu</li> </ul>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Công việc chung:**

- Chỉnh sửa chi tiết, lắp ráp phần cứng Robot
- Thông nhất dáng đi Robot
- Định hình và thiết kế Frame truyền giao tiếp giữa robot và GUI
- Thiết kế bộ điều khiển PID, và hiệu chỉnh thông số cho robot
- Tiến hành các thí nghiệm và đánh giá hiệu suất tổng thể của hệ thống

**Xác nhận của Cán bộ hướng dẫn**

(Ký tên và ghi rõ họ tên)

TP. HCM, ngày.....tháng .....năm.....

**Sinh viên**

(Ký tên và ghi rõ họ tên)

## **DANH SÁCH HỘI ĐỒNG BẢO VỆ ĐỒ ÁN**

Hội đồng chấm đồ án tốt nghiệp, thành lập theo Quyết định số .....  
ngày ..... của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

1. ..... – Chủ tịch.
2. ..... – Thư ký.
3. ..... – Ủy viên.
4. ..... – Ủy viên.
5. ..... – Ủy viên.

## MỤC LỤC

Chương 1. GIỚI THIỆU VỀ ĐỀ TÀI .....	4
1.1. Lý do chọn đề tài .....	4
1.2. Mục tiêu của đồ án .....	5
1.3. Đối tượng và phạm vi nghiên cứu .....	5
1.3.1. Đối tượng nghiên cứu .....	5
1.3.2. Phạm vi nghiên cứu .....	5
Chương 2. CƠ SỞ LÝ THUYẾT .....	6
2.1. Tổng quan về robot có chân .....	6
2.1.1. Ưu điểm và nhược điểm .....	7
2.2. Cấu trúc của robot có chân .....	8
2.2.1. Cấu trúc thiết kế .....	8
2.2.2. Dáng đi robot .....	9
2.3. Động học Robot (Robot Kinematics) .....	11
2.3.1. Động học thuận (Forward Kinematics) .....	13
2.3.2. Động học nghịch (Inverse Kinematics) .....	16
2.4. Bộ điều khiển PID .....	18
2.5. Sensor Fusion .....	21
2.5.1. IMU (Inertial Measurement Unit) và Magnetometer .....	21
2.6. Bộ lọc Madgwick .....	23
2.7. Giao thức truyền thông .....	28
2.7.1. Chuẩn giao tiếp UART .....	28
2.7.2. Chuẩn giao tiếp I2C .....	29
2.7.3. Chuẩn giao tiếp SPI .....	29
2.7.4. DMA (truy cập bộ nhớ trực tiếp) .....	31
Chương 3. THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG .....	33
3.6. Mô hình robot .....	33
3.6.1. Thiết kế mô hình 3D SolidWorks .....	33
3.1.2. Hoàn thiện mô hình robot thực tế .....	44

3.2. Sơ đồ kết nối .....	49
3.2.1. Sơ đồ nguồn điện .....	49
3.2.2. Sơ đồ hệ thống .....	50
3.3. Thiết kế Schematic và PCB .....	50
3.3.1. Thiết kế mạch Schematic .....	50
3.3.2. Thiết kế mạch PCB .....	51
3.3.3. Hoàn thiện mạch thực tế .....	52
3.4. Lựa chọn linh kiện .....	53
3.4.1. Board điều khiển (STM32F407 Discovery).....	53
3.4.2. Động cơ (MG996R).....	54
3.4.3. Nguồn điện (Pin 7.4V 2800mAh).....	56
3.4.4. Module chuyển đổi điện áp (Buck DC-DC LM2596).....	56
3.4.5. Mạch điều khiển PWM (PCA9685).....	57
3.4.6. Cảm biến 9DOF IMU MPU9250.....	58
3.4.7. Module giao tiếp không dây (NRF24L01) .....	60
3.4.8. Camera ESP32-CAM.....	61
Chương 4. THIẾT KẾ GIAO DIỆN VÀ GIẢI THUẬT ĐIỀU KHIỂN .....	63
4.1. Thiết kế giải thuật cho vi điều khiển trên robot.....	63
4.1.1. Luồng điều khiển hoạt động của robot .....	65
4.1.2. Luồng truyền nhận dữ liệu với máy tính.....	74
4.2. Xác định hướng và vị trí robot trên giao diện .....	78
4.2.1. Xác định hướng bằng IMU .....	78
4.2.2. Xác định vị trí tương đối qua Camera.....	79
4.3. Thiết kế giao diện điều khiển trên máy tính .....	80
Chương 5. KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ .....	82
5.1. Di chuyển tiến lùi.....	82
5.2. Xoay trái xoay phải .....	85
5.3. Cân bằng thân robot bằng PID .....	87
5.3.1. Nghiêng theo góc chỉ định.....	87

5.3.2.    Tạo góc nghiêng để robot tự cân bằng.....	99
5.4.    Di chuyển trên bề mặt dốc .....	110
<b>Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>113</b>
6.1.    Kết luận.....	113
6.1.1.    Mục tiêu đạt được.....	113
6.1.2.    Hạn chế cần khắc phục .....	113
6.2.    Định hướng phát triển đề tài .....	114

## DANH MỤC HÌNH VẼ

Hình 2.1 Hình ảnh robot có chân .....	6
Hình 2.2 Sơ đồ dáng đi 2 pha không liên tục, (a) pha 1, (b) pha 2. ....	10
Hình 2.3 Sơ đồ dáng đi xoay tại chỗ, (a) xoay trái, (b) xoay phải.....	11
Hình 2.4 Mối liên hệ giữa các khớp.....	12
Hình 2.5 Đặt hệ trục tọa độ cho chân phải trước của Robot.....	14
Hình 2.6 Góc theta 2 và theta 3 của robot trong không gian 3D.....	17
Hình 2.7 Biểu diễn góc theta 2 và theta 3 của robot trong mặt phẳng 2D.....	17
Hình 2.8 Sensor Fusion IMU và Magnetometer.....	23
Hình 2.9 Sơ đồ bộ lọc Madgwick .....	27
Hình 2.10 Giao tiếp UART giữa hai thiết bị .....	28
Hình 2.11 UART packet.....	28
Hình 2.12 Giao tiếp I2C giữa các thiết bị .....	29
Hình 2.13 Giao tiếp SPI giữa các thiết bị.....	30
Hình 2.14 DMA1 Request mapping .....	32
Hình 2.15 DMA2 Request mapping .....	32
Hình 3.1 Thân trên Robot .....	34
Hình 3.2 Thân dưới Robot.....	35
Hình 3.3 Trụ đỡ cho thân Robot.....	36
Hình 3.4 Thân Robot hoàn chỉnh .....	36
Hình 3.5 Động cơ Servo MG996R.....	37
Hình 3.6 Mặt trên khớp vai Robot.....	38
Hình 3.7 Mặt dưới khớp vai Robot .....	38
Hình 3.8 Khớp đùi Robot.....	39
Hình 3.9 Khớp cẳng Robot.....	39
Hình 3.10 Chân phải Robot sau khi hoàn thiện .....	40
Hình 3.11 Chân phải Robot-mặt trên.....	40
Hình 3.12 Chân trái Robot-mặt trước .....	41
Hình 3.13 Chân trái Robot-mặt trên.....	41
Hình 3.14 Mô hình robot hoàn chỉnh.....	42
Hình 3.15 Mô hình robot hoàn chỉnh-mặt trên.....	42
Hình 3.16 Mô hình robot hoàn chỉnh-mặt dưới .....	43
Hình 3.17 Mô hình robot hoàn chỉnh-mặt trước .....	43
Hình 3.18 Mô hình robot sau khi gắn động cơ.....	44
Hình 3.19 Thứ tự động cơ của 1 chân.....	45
Hình 3.20 Mô hình robot với module PWM và module nguồn .....	46
Hình 3.21 Mô hình robot với mạch PCB .....	47

Hình 3.22 Dáng đứng của robot khi khởi động.....	48
Hình 3.23 Mô hình robot hoàn chỉnh.....	48
Hình 3.24 Sơ đồ mô tả nguồn điện qua các thiết bị.....	49
Hình 3.25 Sơ đồ kết nối hệ thống .....	50
Hình 3.26 Khối Vi xử lí STM32F407 .....	
	Hình 3.27 Các khối chức năng 51
Hình 3.28 2D hình vẽ PCB mạch điện .....	52
Hình 3.29 3D hình vẽ PCB mạch điện.....	52
Hình 3.30 Mạch PCB hoàn thiện .....	53
Hình 3.31 Kit STM32F407 Discovery .....	53
Hình 3.32 Động cơ MG996R .....	54
Hình 3.33 Chu kì xung PWM của MG996R.....	55
Hình 3.34 Pin 7.4V 2800mAh.....	56
Hình 3.35 Mạch giảm áp DC-DC Buck LM2596 3A.....	56
Hình 3.36 Mạch điều khiển 16 Chanel PWM PCA9685 .....	57
Hình 3.37 9DOF IMU MPU9250 .....	58
Hình 3.38 Module thu phát RF NRF24L01 2.4Ghz .....	60
Hình 3.39 Kit phát triển Wifi ESP32-CAM.....	61
Hình 4.1 Lưu đồ giải thuật tổng quát của robot.....	64
Hình 4.2 Luồng điều khiển hoạt động của robot.....	65
Hình 4.3 Robot trạng thái đứng-mặt trên .....	69
Hình 4.4 Robot trạng thái đứng-mặt trước .....	70
Hình 4.5 Điều chỉnh chiều cao của từng chân để cân bằng thân robot .....	70
Hình 4.6 Mặt phẳng nghiêng theo góc Pitch.....	71
Hình 4.7 Mặt phẳng nghiêng theo góc Roll .....	71
Hình 4.8 Luồng truyền nhận dữ liệu với máy tính.....	74
Hình 4.9 Frame của mode đứng, ngồi, stop và emergency .....	75
Hình 4.10 Frame của mode di chuyển tiến lùi .....	75
Hình 4.11 Frame của mode xoay trái, xoay phải.....	76
Hình 4.12 Frame của mode điều khiển từng chân robot .....	76
Hình 4.13 Frame của mode điều khiển từng động cơ robot.....	77
Hình 4.14 Frame của mode cân bằng thân robot .....	77
Hình 4.15 Frame dữ liệu cảm biến gửi đến máy tính .....	78
Hình 4.16 Xác định hướng bằng IMU .....	79
Hình 4.17 Xác định vị trí tương đối bằng camera .....	80
Hình 4.18 Giao diện điều khiển trên máy tính.....	80
Hình 5.1 Sơ đồ dáng đi tới, lui hai pha không liên tục .....	82

Hình 5.2 Sơ đồ dáng đi tới, lui thực tế - Pha 1 .....	82
Hình 5.3 Sơ đồ dáng đi tới, lui thực tế - Pha 2 .....	82
Hình 5.4 Góc và hướng thân robot trong khi đi tới liên tục lần 1 .....	83
Hình 5.5 Góc và hướng thân robot trong khi đi tới liên tục lần 2 .....	84
Hình 5.6 Góc nghiêng thân robot trong khi đi lùi liên tục .....	84
Hình 5.7 Sơ đồ dáng xoay trái (a), xoay phải (b) .....	85
Hình 5.8 Sơ đồ dáng xoay trái của robot .....	85
Hình 5.9 Góc nghiêng thân robot trong khi xoay trái lần 1 .....	86
Hình 5.10 Góc nghiêng thân robot trong khi xoay trái lần 2 .....	86
Hình 5.11 Góc nghiêng thân robot trong khi xoay trái lần 3 .....	86
Hình 5.12 Nghiêng robot theo góc pitch TH1 .....	87
Hình 5.13 Nghiêng robot theo góc pitch TH2 .....	88
Hình 5.14 Nghiêng robot theo góc pitch TH3 .....	88
Hình 5.15 Nghiêng robot theo góc pitch TH4 .....	89
Hình 5.16 Nghiêng robot theo góc pitch TH5 .....	89
Hình 5.17 Nghiêng robot theo góc roll TH1 .....	93
Hình 5.18 Nghiêng robot theo góc roll TH2 .....	94
Hình 5.19 Nghiêng robot theo góc roll TH3 .....	94
Hình 5.20 Nghiêng robot theo góc roll TH4 .....	95
Hình 5.21 Nghiêng robot theo góc roll TH5 .....	95
Hình 5.22 Cân bằng thân robot bằng PID .....	99
Hình 5.23 Robot tự cân bằng trên góc pitch TH1 .....	100
Hình 5.24 Robot tự cân bằng trên góc pitch TH2 .....	100
Hình 5.25 Robot tự cân bằng trên góc pitch TH3 .....	101
Hình 5.26 Robot tự cân bằng trên góc pitch TH4 .....	101
Hình 5.27 Robot tự cân bằng trên góc roll TH1 .....	105
Hình 5.28 Robot tự cân bằng trên góc roll TH2 .....	106
Hình 5.29 Robot tự cân bằng trên góc roll TH3 .....	106
Hình 5.30 Robot tự cân bằng trên góc roll TH4 .....	106
Hình 5.31 Bề mặt dốc .....	110
Hình 5.32 Robot di chuyển trên bề mặt dốc .....	110
Hình 5.33 Đồ thị góc nghiêng robot khi di chuyển tiến trên bề mặt dốc lần 1 ..	111
Hình 5.34 Đồ thị góc nghiêng robot khi di chuyển tiến trên bề mặt dốc lần 2 ..	111
Hình 5.35 Đồ thị góc nghiêng robot khi di chuyển lùi trên bề mặt dốc lần 1 ....	112
Hình 5.36 Đồ thị góc nghiêng robot khi di chuyển lùi trên bề mặt dốc lần 2 ....	112

## **DANH MỤC BẢNG**

Bảng 2.1 So sánh robot bánh xe và robot có chân.....	7
Bảng 2.2 Bảng DH.....	15
Bảng 2.3 Đặc điểm của các khâu trong điều khiển vòng kín.....	21
Bảng 3.1 Kích thước các khớp của chân Robot.....	39
Bảng 3.2 Bảng hiệu chỉnh góc quay ban đầu của 12 động cơ.....	45
Bảng 3.3 Tầm quay giới hạn của động cơ.....	46
Bảng 3.4 Thông số kỹ thuật cơ bản STM32F4 đáp ứng yêu cầu robot.....	54
Bảng 3.5 Thông số kỹ thuật MG996R .....	55
Bảng 3.6 Thông số kỹ thuật Mạch giảm áp DC-DC Buck .....	57
Bảng 3.7 Thông số kỹ thuật PCA9685 .....	58
Bảng 3.8 Thông số kỹ thuật IMU MPU9250 .....	59
Bảng 3.9 Thông số kỹ thuật NRF24L01 .....	61
Bảng 3.10 Thông số kỹ thuật ESP32-S .....	62
Bảng 3.11 Thông số kỹ thuật Camera OV2640 .....	62
Bảng 5.1 Khoảng cách di chuyển sau mỗi chu kỳ tiến lên.....	83
Bảng 5.2 Đáp ứng điều khiển PID theo góc pitch .....	92
Bảng 5.3 Đáp ứng điều khiển PID theo góc roll.....	98
Bảng 5.4 Đáp ứng điều khiển PID theo góc pitch .....	104

## **DANH MỤC TỪ VIẾT TẮT**

DoF: Degree of Freedom

DH: Denavit Hartenberg

FK: Forward Kinematics

IK: Inverse Kinematics

PID: Proportional-Integral-Derivative

UART: Universal Asynchronous Receiver-Transmitter

SPI: Serial Peripheral Interface

I2C: Inter-Integrated Circuit

PWM: Pulse Width Modulation

GUI: Graphical User Interface

GPIO: General Purpose Input Output

IMU: Inertial measurement unit

## TÓM TẮT ĐỒ ÁN BẰNG TIẾNG VIỆT

Đồ án này trình bày quá trình phát triển và điều khiển một mô hình robot nhện bốn chân linh hoạt và ổn định, có khả năng thực hiện các động tác cơ bản như tiến, lùi, xoay trái, xoay phải và cân bằng. Trong quá trình nghiên cứu, đã phân tích và tham khảo các mô hình robot nhện hiện có để xác định các yếu tố quan trọng cần tích hợp. Mục tiêu là tối ưu hóa hiệu suất của robot trên các địa hình phức tạp.

Tiếp theo, thiết kế mô hình robot nhện bốn chân, hoàn thiện mạch điều khiển cố định chắc chắn cho robot bao gồm cơ học và cơ điện học chi tiết để đảm bảo tính ổn định và khả năng di chuyển linh hoạt.

Hệ thống điều khiển của robot được phát triển dựa trên các thuật toán điều khiển hiện đại, cho phép robot thực hiện các động tác một cách chính xác và mượt mà. Sự tích hợp của các cảm biến và hệ thống phản hồi tức thì giúp robot tự cân bằng và thích ứng tự động với môi trường xung quanh.

Tiến hành thử nghiệm thực tế để đánh giá hiệu suất và tính năng của mô hình robot nhện bốn chân. Kết quả thử nghiệm cho thấy robot có khả năng di chuyển một cách linh hoạt và ổn định trên nhiều loại địa hình khác nhau. Đồng thời, robot thể hiện khả năng thực hiện đa dạng động tác một cách chính xác và đáng tin cậy.

Tổng kết, đồ án này đề xuất và thực hiện một mô hình robot nhện bốn chân có khả năng điều khiển linh hoạt và đáng tin cậy. Nghiên cứu này góp phần nâng cao hiểu biết về robot học và điều khiển tự động, đồng thời cung cấp tiềm năng ứng dụng trong các lĩnh vực như nghiên cứu cứu hộ, khám phá môi trường nguy hiểm và các ứng dụng công nghiệp có tính phức tạp.

## **ABSTRACT**

This thesis presents the development and control process of a flexible and stable four-legged spider robot model capable of performing basic actions such as forward, backward, left turn, right turn, and balancing. During the research phase, existing spider robot models were analyzed and referenced to identify key factors for integration. The goal was to optimize the robot's performance on complex terrains.

Subsequently, the design of the four-legged spider robot model was carried out, along with the meticulous incorporation of detailed mechanical and electromechanical control circuits to ensure stability and flexible movement. The robot's control system was developed based on modern control algorithms, enabling precise and smooth execution of actions. Integration of sensors and real-time feedback systems helped the robot maintain balance and adapt automatically to its surrounding environment.

Real-world testing was conducted to evaluate the performance and features of the four-legged spider robot model. Test results demonstrated the robot's ability to maneuver flexibly and stably across various types of terrain, while also showcasing its precise and reliable execution of diverse actions.

In conclusion, this project proposes and implements a four-legged spider robot model with flexible and reliable control capabilities. This research contributes to advancing understanding in robotics and autonomous control, while also offering potential applications in fields such as search and rescue, hazardous environment exploration, and complex industrial applications.

## MỞ ĐẦU

Ngày nay, việc thiết kế, phát triển và lập kế hoạch chuyển động của một robot di động khám phá các lĩnh vực nghiên cứu trong lĩnh vực robot. Robot di động có phạm vi ứng dụng rộng rãi trong các lĩnh vực khác nhau như khám phá không gian, ứng dụng quân sự, sử dụng trong công nghiệp và nhiều lĩnh vực khác. Do đó, việc thiết kế và phát triển một robot di động là một phần quan trọng của ứng dụng nói trên.

Tiềm năng của robot có nhiều chân trong việc vận hành trên địa hình khắc nghiệt vượt trội so với các loại robot di động sử dụng bánh xe hoặc bánh xích. Chúng thể hiện tính linh hoạt và khả năng thích ứng với địa hình phức tạp ở tốc độ thấp. Tuy nhiên, việc điều khiển và tính toán cho robot nhiều chân đặt ra nhiều thách thức hơn so với các loại robot khác. Cấu trúc động học và đặc tính động học của robot này đóng vai trò quan trọng trong việc đảm bảo khả năng vận hành và kiểm soát chuyển động của chúng. Để robot có thể hoạt động hiệu quả, việc lập kế hoạch cho các chuyển động, tính toán động học và xây dựng mô hình động lực học là cần thiết. Việc tiêu thụ ít năng lượng, ổn định và di chuyển tốt trên địa hình không bằng phẳng và gồ ghề, hoạt động trong những môi trường nguy hiểm... là những ưu điểm chính của robot bốn chân. Số lượng nghiên cứu về robot bốn chân đã tăng lên trong những năm gần đây do hoạt động tốt hơn trong các điều kiện địa hình khó khăn. Ví dụ về các nghiên cứu hiện tại về robot bốn chân; BigDog được phát triển bởi Boston Dynamics, HyQ2Max được phát triển bởi Semini và cộng sự, Jinpong được phát triển bởi Cho và cộng sự.

Trong số tất cả các loại robot di động, robot có chân, vượt trội hơn so với robot có bánh xe và bánh xích do tiềm năng khám phá mọi địa hình như người và động vật, có thể hoạt động thay thế con người trong những môi trường độc hại... Do đó việc đầu tư nghiên cứu và phát triển các loại robot di động phục vụ cho công việc của con người là rất cần thiết cho hiện tại và tương lai.

## **Chương 1. GIỚI THIỆU VỀ ĐỀ TÀI**

### **1.1. Lý do chọn đề tài**

Các loại robot nhiều chân, đặc biệt là robot có bốn chân, đang trở nên ngày càng quan trọng do tính linh hoạt và khả năng thích ứng của chúng với môi trường khác nhau. Tuy nhiên, vấn đề về khả năng di chuyển vẫn là một thách thức lớn trong quá trình phát triển của các loại robot này. Việc xây dựng một robot có sự di chuyển mềm mại và ổn định mang lại nhiều ứng dụng rộng rãi trong cả ngành công nghiệp và gia đình. Đề án này tập trung vào việc đề xuất một phương pháp để phát triển một robot bốn chân dạng nhện đảm bảo sự mượt mà khi di chuyển và ổn định.

Một trong những lý do đáng chú ý cho việc nghiên cứu và phát triển robot nhện 4 chân là khả năng thích ứng với môi trường khắc nghiệt và khó tiếp cận. Với khả năng di chuyển linh hoạt qua các bề mặt không đồng nhất như bã cỏ, đá, cát hay cấu trúc phức tạp, robot nhện có tiềm năng ứng dụng rộng rãi trong nhiều ngành công nghiệp như khảo sát môi trường, tìm kiếm cứu hộ, khám phá không gian hạn chế, và thậm chí trong ngành y tế.

Điều đó đã dẫn tới việc nhóm quyết định chọn mô hình robot nhện bốn chân để phát triển. Việc thực hiện một mô hình robot như vậy không chỉ là việc nghiên cứu về kỹ thuật, mà còn là cơ hội để thể hiện sự sáng tạo và tích hợp những kiến thức đa dạng. Bên cạnh đó, việc thiết kế và điều khiển robot nhện còn đòi hỏi sự kết hợp giữa nhiều kiến thức đa dạng như cơ học, điện tử, lập trình. Nghiên cứu về mô hình này sẽ giúp chúng ta khám phá cách tối ưu hóa cấu trúc cơ học, phát triển các thuật toán điều khiển thông minh để đảm bảo chính xác và hiệu quả trong việc di chuyển và thực hiện nhiệm vụ.

Tóm lại, lý do chọn đề tài "Thiết kế và Điều khiển Mô hình Robot Nhện bốn Chân" không chỉ nằm trong sự hấp dẫn của việc nghiên cứu và phát triển về một dạng robot độc đáo, mà còn giúp sinh viên học hỏi những kỹ năng mới cũng như cải thiện những kỹ năng hiện tại.

## **1.2. Mục tiêu của đồ án**

Nhiệm vụ chính của đồ án là thiết kế một mô hình robot nhện bốn chân hoàn chỉnh có thể hoạt động được, đồng thời lập trình để điều khiển mô hình.

Tìm hiểu sâu hơn về robot, đặc biệt là những loại robot có chân. Những yếu tố giúp một mô hình robot có chân có thể hoạt động được: kết cấu cơ khí, cách điều khiển, thuật toán điều khiển,...

Thiết kế cơ khí một mô hình robot hoàn chỉnh có thể hoạt động được. Tìm hiểu lập trình đa luồng để điều khiển robot thực hiện những động tác từ xa đồng thời quan sát được trạng thái của robot.

Thiết kế giải thuật điều khiển nhằm điều khiển robot hoạt động một cách hiệu quả.

## **1.3. Đối tượng và phạm vi nghiên cứu**

### **1.3.1. Đối tượng nghiên cứu**

Đồ án này sẽ tập trung nghiên cứu và phát triển mô hình robot nhện 4 chân dựa trên nguyên tắc cơ học và điều khiển. Đối tượng nghiên cứu của đồ án bao gồm:

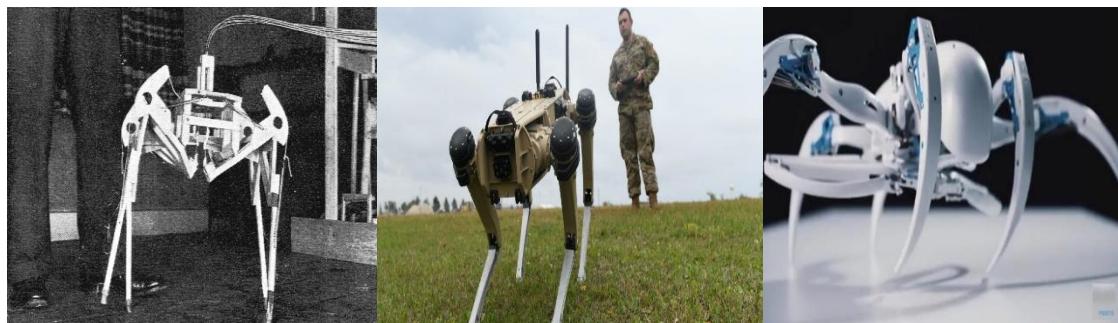
- Mô hình Robot Nhện 4 Chân: Nghiên cứu sẽ tập trung vào thiết kế và xây dựng một mô hình robot nhện có 4 chân với cấu trúc cơ học tối ưu để đạt được khả năng di chuyển linh hoạt và ổn định.
- Thuật toán Điều khiển: Đồ án sẽ khám phá và phát triển thuật toán điều khiển để robot nhện có thể hoạt động, thực hiện những thao tác cơ bản và điều khiển robot theo ý muốn.

### **1.3.2. Phạm vi nghiên cứu**

Đồ án nghiên cứu trong lĩnh vực cơ khí, robot và điều khiển. Các lý thuyết robot cũng như những vấn đề để một robot có thể hoạt động được như: kích thước, dáng đi, giải thuật điều khiển, trọng tâm,...

## Chương 2. CƠ SỞ LÝ THUYẾT

### 2.1. Tổng quan về robot có chân



Hình 2.1 Hình ảnh robot có chân

Trong ba thập kỷ qua, robot di động đã thu hút nhiều sự chú ý vì khả năng khám phá trong môi trường phức tạp, không gian, hoạt động cứu hộ, hoàn thành nhiệm vụ mà không cần sự can thiệp của con người, v.v. Robot di động có thể được phân loại rộng rãi thành ba loại: robot bánh xe, robot bánh xích và robot có chân. Trong đó, hệ thống di chuyển của robot là đặc điểm quan trọng của thiết kế di động, và nó phụ thuộc không chỉ vào không gian làm việc mà còn vào các yếu tố kỹ thuật như khả năng di chuyển, khả năng kiểm soát, điều kiện địa hình, hiệu suất và ổn định. Mỗi hệ thống robot có những ưu và nhược điểm riêng. So sánh chi tiết về các tiêu chí kỹ thuật của robot chân so với robot bánh xe ta thấy được sự phát triển của hệ thống di chuyển của robot có chân qua vài thập kỷ qua do có nhiều ưu điểm hơn so với các phương tiện robot bánh xe [3], điều này được quan sát từ Bảng 2.1

Tiêu chí kỹ thuật	Robot Bánh xe	Robot có chân
Khả năng vận động	X	✓
Khả năng vượt địa hình	X	✓
Khả năng kiểm soát	✓	X

Điều hướng Vượt Chướng ngại vật	X	✓
Hiệu quả chi phí	✓	X
Sự linh hoạt	X	✓

Bảng 2.1 So sánh robot bánh xe và robot có chân

### 2.1.1. Ưu điểm và nhược điểm

#### 2.1.1.1. Ưu điểm

Ưu điểm của việc di chuyển bằng chân phụ thuộc vào tư thế, số lượng chân và chức năng của chân. Mặc dù robot có bánh xe và bánh xích có thể hoạt động trên địa hình bằng phẳng, nhưng hầu hết chúng không thể hoạt động ở địa hình lõn xộn, môi trường phức tạp và nguy hiểm. Robot có chân có nhiều khả năng hơn để đi lang thang trên hầu hết các bề mặt trái đất ở các địa hình khác nhau, giống như con người và động vật. Động vật sử dụng đôi chân của mình để di chuyển nhanh chóng và đáng tin cậy trên các địa hình khác nhau với khả năng vận động và sự nhanh nhẹn tuyệt vời. Thông thường, chúng trang bị với tốc độ và hiệu quả cao tùy thuộc vào điều kiện môi trường. Quan điểm về đáng đi ổn định và hiệu quả, thậm chí số lượng chân phù hợp hơn ở phổ thông. Robot bốn chân là sự lựa chọn tốt nhất trong số tất cả các robot có chân liên quan đến khả năng di chuyển và ổn định khi vận động. Bốn chân của robot được điều khiển, thiết kế và bảo trì dễ dàng so với hai hoặc sáu chân. Sự vận động lấy cảm hứng từ mặt sinh học của dáng chạy là khả năng chịu tải trọng cao và khả năng giữ thăng bằng của robot bốn chân đã được nhà nghiên cứu bắt đầu trước đó. Để đạt được tốc độ thời gian thực và chuyển động tự nhiên như bò, chó, báo săn, cần phải có hệ thống điều khiển phát triển và tạo dáng đi động của robot bốn chân.

Một ưu điểm khác mà gần đây đang được nghiên cứu, liên quan đến khả năng chịu lỗi trong quá trình chuyển động ổn định tĩnh. Hậu quả của việc hỏng một trong các bánh của phương tiện có bánh là mất khả năng di chuyển nghiêm trọng, vì tất cả các bánh của các loại phương tiện này phải tiếp xúc liên tục với mặt đất trong quá trình chuyển động. Tuy nhiên, trong trường hợp một hoặc một số chân của robot bị hỏng, chúng vẫn có thể duy trì cân bằng tĩnh và tiếp tục di chuyển, giảm tác động của sự cố lên khả năng làm nhiệm vụ của robot.

### **2.1.1.2. Nhược điểm**

Mặc dù những điểm đã được đề cập cho thấy rằng robot chân có ưu thế so với robot bánh xe truyền thống, nhưng cần lưu ý rằng, trong tình trạng phát triển hiện nay, những phương tiện này vẫn có những hạn chế đáng kể. Chúng có tốc độ thấp, khó xây dựng và yêu cầu các thuật toán điều khiển phức tạp. Ngoài ra, các cơ chế hiện nay cồng kềnh vì cần một số lượng lớn hệ thống truyền động để di chuyển nhiều độ tự do trong chân, mỗi hệ thống này tiêu thụ nhiều năng lượng hơn.

Ngoài các hạn chế đã đề cập, robot có chân cũng gặp phải nhiều thách thức khác. Do tỷ lệ sự cố hoặc hỏng hóc cao trong quá trình vận hành. Bảo dưỡng định kỳ là điều cần thiết để đảm bảo hoạt động ổn định và hiệu quả của các robot này, do cấu trúc cơ khí phức tạp của chúng. Ngoài ra, robot có chân có thể gặp khó khăn khi hoạt động trên các môi trường đặc biệt như địa hình đầm lầy, tuyết, hoặc các bề mặt không đồng nhất.

## **2.2. Cấu trúc của robot có chân**

### **2.2.1. Cấu trúc thiết kế**

Nhìn lại thành tựu của robot bốn chân đã được phát triển trước đó, có thể thấy rằng các bộ kích hoạt thông thường được sử dụng là thủy lực và điện. Trong số các loại bộ kích hoạt thủy lực và điện, loại thủy lực có tỷ lệ công suất đầu ra trên trọng lượng cao, nhưng đồng thời, trọng lượng của robot

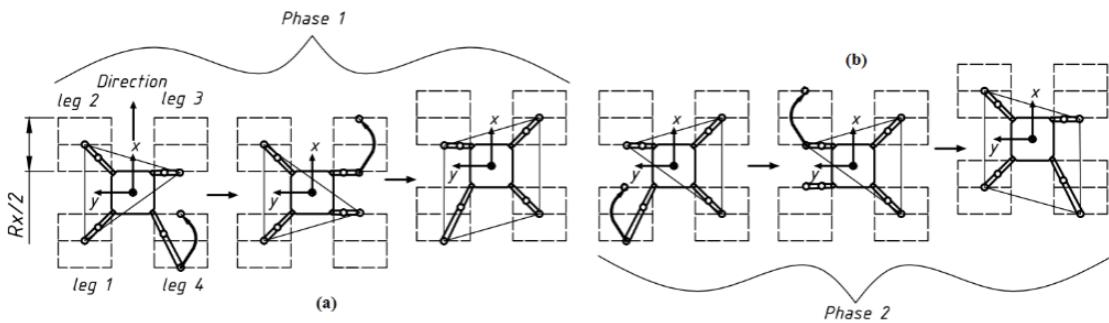
cũng tăng lên tương ứng. Thông số vật lý và kích thước đóng vai trò quan trọng trong việc thiết kế cấu trúc cơ khí. Ban đầu, việc xác định kích thước và trọng lượng chính xác của robot là rất khó khăn. Kích thước của robot phụ thuộc vào sự có sẵn của các thành phần trên thị trường. Trọng lượng của robot phụ thuộc vào hệ thống kích hoạt (Thủy lực hoặc Điện) và tất cả thiết bị điện tử trên board. Mặt khác, việc lựa chọn chiều dài chân là một tham số quan trọng trong thiết kế và phát triển robot. Hầu hết chiều dài chân của robot ít hơn kích thước của robot. Khi chân quá dài so với cơ thể, chúng thường va chạm với nhau và cấu trúc cơ thể tỷ lệ cao yêu cầu cân bằng hơn với độ linh hoạt ít hơn.

Sự phức tạp phụ thuộc vào số lượng khớp nối hoạt động có mặt trong mỗi chân của robot. Mỗi khớp đều làm tăng trọng lượng của robot. Cấu trúc chân hai khớp dễ kiểm soát và thiết kế, thường được sử dụng trong các robot bốn chân như Spot Mini, LittleDog, ANYmal, v.v. Tuy nhiên, từ góc độ sinh học, động vật như hổ, báo, sư tử và sói, có cấu trúc chân ba khớp, mang lại những ưu điểm xuất sắc về tốc độ chạy và hiệu suất năng lượng. Động vật 4 chân sử dụng các bước chạy đối xứng (thường là bước đi và bước chạy) ở tốc độ thấp, trong khi sử dụng các bước chạy không đối xứng ở tốc độ cao. Trong bước chạy đối xứng, các chân có cùng pha giữa, sao cho chân trái và chân phải được đặt xuống sau một khoảng thời gian bằng nhau. Trong các bước chạy không đối xứng, chân sau được đặt xuống ngay trước khoảng thời gian ngắn, và chân trước được đặt xuống ngay sau khoảng thời gian đó. Chân sau nên tạo ra một lực tác động phanh nhỏ, tiếp theo là một lực đẩy lớn hơn.

### 2.2.2. Dáng đi robot

Robot nhiều chân sở hữu tiềm năng to lớn về khả năng cơ động trên địa hình gồ ghề, đặc biệt là so với thiết bị di động có bánh xe hoặc bánh xích thông thường. Nó có tính linh hoạt hơn và khả năng thích ứng địa hình ở tốc độ thấp. Tuy nhiên, các yêu cầu phối hợp giữa chuyển động của chân, điều khiển và tính toán robot bốn chân gấp nhiều khó khăn hơn so với các loại khác. Động

học và đặc điểm động của robot bốn chân tác động toàn diện đến khả năng để đáp ứng điều khiển chuyển động của robot. Để robot có thể di chuyển theo ý muốn của chúng ta, chúng ta phải lập kế hoạch cho bước đi, tính toán bài toán động học. Do đó, cần xây dựng một mô hình động học hoàn chỉnh, lập kế hoạch dáng đi, đảm bảo hình dạng được di chuyển, tính ổn định cho robot, từ đó xây dựng mô hình động, đây là cơ sở để xây dựng các bộ điều khiển chuyển động, nâng cao khả năng chuyển động, thực hiện nhiệm vụ, và tối ưu hóa việc cung cấp năng lượng.



Hình 2.2 Sơ đồ dáng đi 2 pha không liên tục, (a) pha 1, (b) pha 2.

Dáng đi của robot được hiện là hai pha không liên tục (two-phase discontinuous). Với các mô hình như vậy, tại mỗi điểm trên cơ thể robot đều được duy trì cân bằng bởi ba chân đặt trên mặt đất, khiến cho robot rất ổn định khi di chuyển. Chuyển động của robot với cách bước đi này là hai giai đoạn không liên tục. Cơ thể được đẩy về phía trước hoặc phía sau với tất cả các chân đặt chắc chắn trên mặt đất và một chân được chuyển đổi cùng với ba chân và cơ thể dừng lại. Đối với mỗi giai đoạn, cơ thể robot sẽ đẩy về phía trước một lần.

Trong đề tài này, dáng đi của robot sẽ được bắt đầu từ bước thứ 2 như để đảm bảo vấn đề cân bằng của robot. Sau đây là các bước cụ thể để điều khiển robot thực hiện dáng đi hai pha không liên tục (two-phase discontinuous):

- Bước 1: di chuyển chân 3 cùng chiều trực y một đoạn là  $2y\_step$ .
- Bước 2: di chuyển cả 4 chân robot ngược chiều trực y một đoạn  $y\_step$ .

Lúc này thân robot nhích lên một đoạn  $y\_step$ .

- Bước 3: di chuyển chân 1 cùng chiều trục y một đoạn  $y_{\text{step}}$
- Bước 4: di chuyển chân 2 cùng chiều trục y một đoạn là  $2y_{\text{step}}$ .
- Bước 5: di chuyển cả 4 chân robot ngược chiều trục y một đoạn  $y_{\text{step}}$ .

Lúc này thân robot nhích lên một đoạn  $y_{\text{step}}$ .

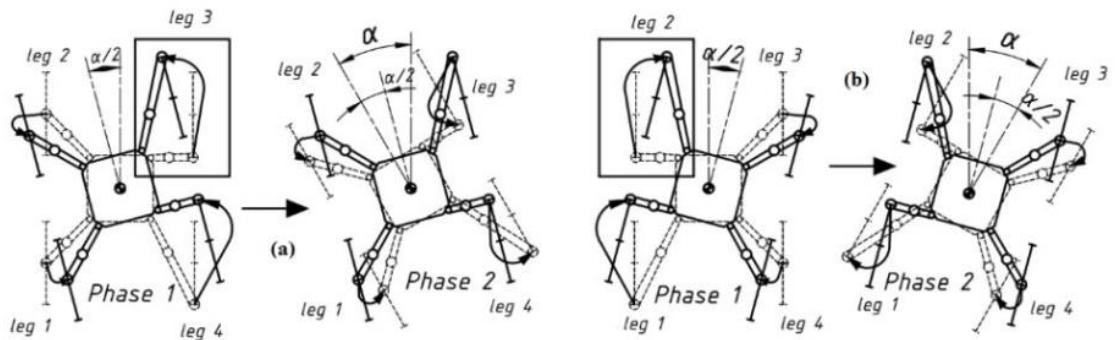
- Bước 6: di chuyển chân 4 cùng chiều trục y một đoạn  $y_{\text{step}}$

Thực hiện xong 6 bước trên thì tư thế của robot sẽ trở về vị trí ban đầu, robot thực hiện được một chu kỳ di chuyển.

Tuy nhiên trong quá trình lập trình để điều khiển robot, nhận thấy rằng không nhất thiết một bước đi của robot phải thực hiện cả 6 bước như trên mà có thể chia nhỏ thành 2 phần:

- Phần 1: 3 bước đầu tiên trong quá trình 6 bước như trên.
- Phần 2: 3 bước sau.

Hai phần này cách di chuyển của robot hoàn toàn tương tự nhau, đối xứng nhau qua trục y. Do đó, bằng việc thêm một số câu lệnh điều kiện trong phần lập trình, có thể chia nhỏ một bước của robot thành hai bước khi điều khiển. Điều này giúp tăng tính linh hoạt của robot khi hoạt động.

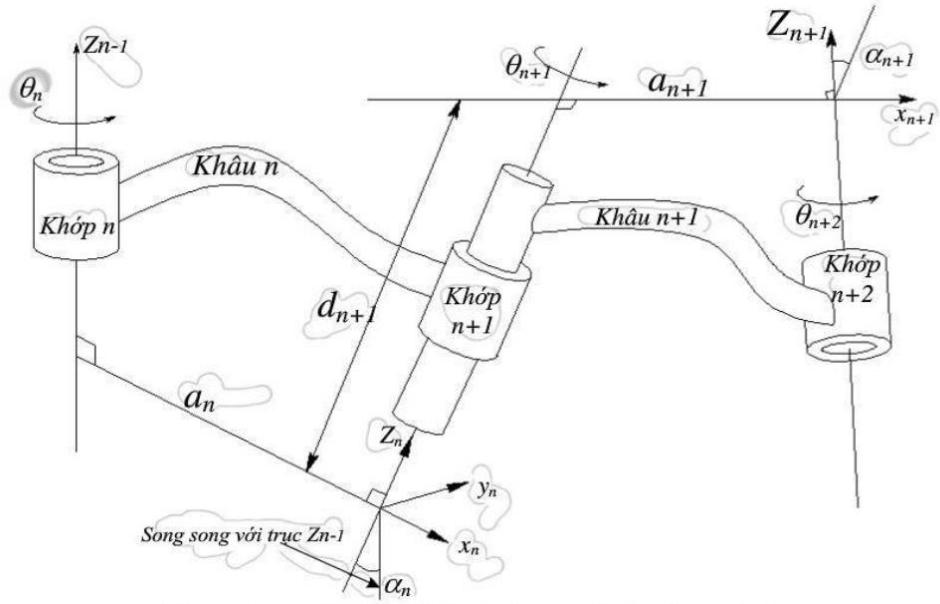


Hình 2.3 Sơ đồ dáng đi xoay tại chỗ, (a) xoay trái, (b) xoay phải

### 2.3. Động học Robot (Robot Kinematics)

Đối với robot nối tiếp (chuỗi hở), các khâu được liên kết với nhau thông qua các khớp động (quay/tịnh tiến). Nếu mỗi khâu gắn với một hệ trục tọa độ, 2 khâu liên tiếp được thể hiện qua ma trận thuần nhất T.

Để xây dựng được phương trình động học cho robot, ta sử dụng quy tắc Denavit-Hartenberg (DH).



Hình 2.4 Mối liên hệ giữa các khớp

#### Bộ thông số D-H bao gồm 4 thông số:

- Độ dài đường pháp tuyến chung  $a_n$ . (đường vuông góc chung 2 trục  $z_{n-1}$  và  $z_n$ ).
- Góc giữa trực khớp trong mặt phẳng vuông góc với pháp tuyến  $\alpha_n$ .
- Góc quay quanh trực z 1 góc  $\theta$ .
- Độ dài tiếp tuyến chung  $d$  (đường vuông góc chung 2 trục Ox).

Tóm lại:

- $\alpha_n$  : góc quay quanh trực x.(trục z đổi hướng).
- $a_n$  : khoảng cách 2 trực z
- $d$ : khoảng cách 2 trực x.
- $\theta_n$ : góc quay quanh trực z.

Ý nghĩa của bộ thông số DH: Giúp ta xác định vị trí và hướng của 1 khâu so với khâu trước nó và so với hệ tọa độ gốc. Nếu khớp nối hai khâu là khớp quay thì  $\theta_n$  là biến khớp (3 thông số còn lại là hằng số. Nếu khớp nối là tịnh tiến thì  $d_n$  là biến khớp ( $\theta_n = 0, a_n = 0, \alpha_n = const$ )).

#### Quy tắc chọn gốc tọa độ cho robot (Quy tắc DH):

- Góc của khâu thứ n nằm trên đường tâm của trục khớp thứ (n+1) và giao điểm đường pháp tuyến chung an.
- Nếu 2 trục cắt nhau thì gốc tại điểm cắt đó, nếu 2 trục song song thì On nằm ở vị trí nào để thuận tiện cho quá trình tính toán.
- Chọn trục Zn : trục Zn nằm dọc theo trục khớp thứ n+1.
- Trục Xn : nằm pháp tuyến chung (đường vuông góc chung) hướng từ trục khớp n đến trục khớp n+1
- Nếu 2 trục khớp cắt nhau thì  $\vec{x}n = \vec{z}n - 1 \cdot \vec{z}n$
- Góc quay cùng chiều kim đồng hồ: dương; ngược chiều kim đồng hồ: âm.

Ma trận  $T_i$  được gọi là ma trận chuyển đổi thuần nhất, nó có dạng:

$${}_{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Từ đó ta có ma trận vector cuối:

Với:  
+  $a^{\rightarrow}$ : Vector có hướng mà theo đó bàn tay sẽ tiếp cận đến tối tượng  
+  $o^{\rightarrow}$  Vector có hướng theo đó các ngón tay cầm nắm hay thả tối tượng  
+  $n^{\leftrightarrow}$ : Vector pháp tuyến của  $o^{\rightarrow}$  và  $a^{\rightarrow}$ :  $n^{\leftrightarrow} = o^{\rightarrow} \cdot a^{\rightarrow}$

### 2.3.1. Động học thuận (Forward Kinematics)

Trong đại đa số các trường hợp, tay máy là một chuỗi động hở, được cấu tạo bởi một số khâu (Links), được nối với nhau nhờ các khớp. Một đầu của chuỗi nối với giá (Base), còn đầu kia nối với phần công tác. Mỗi khâu hình thành cùng với khớp phía trước nó một cặp khâu – khớp. Tuỳ theo kết cấu của mình mà mỗi loại khớp đảm bảo cho khâu nối sau nó các khả năng chuyển động nhất định.

Mỗi khớp (thực chất là cặp khâu – khớp) được đặc trưng bởi 2 loại thông số:

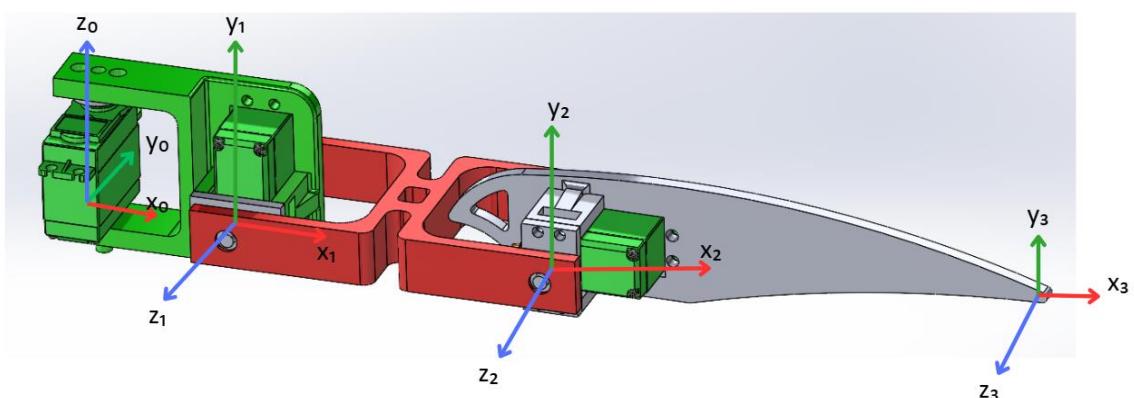
- Các thông số không thay đổi giá trị trong quá trình làm việc của tay máy được gọi là tham số.

- Các thông số thay đổi khi tay máy làm việc được gọi là các biến khớp. Hai loại khớp thông dụng nhất trong kỹ thuật tay máy là khớp trượt và khớp quay. Chúng đều là loại khớp có một bậc tự do.

Mục đích của bài toán động học thuận là xác định vị trí của khâu tác động cuối của tay máy khi biết các biến khớp của tay máy. Các bước thực hiện bài toán động học thuận cho tay máy:

- Bước 1: Đặt các hệ trục tọa độ lên tay máy theo quy tắc DH (Mục 2.3)
- Bước 2: Lập bảng tham số động học DH
- Bước 3: Tính các ma trận biến đổi thuần nhất giữa các hệ trục tọa độ bằng cách thay các tham số động học ở mỗi dòng bảng DH vào ma trận (2.)
- Bước 4: Rút các phương trình động học thuận tay máy từ ma trận  $n \times T$  được tính theo công thức (2.)

Áp dụng vào đồ án:



Hình 2.5 Đặt hệ trục tọa độ cho chân phải trước của Robot

Do chân phải và chân trái của Robot là đối xứng, ta chỉ cần tính toán động học thuận, động học nghịch cho một chân của Robot và dùng kết quả này áp dụng cho ba chân còn lại.

Với thông số của các khớp đã biết trong quá trình thiết kế, độ dài của các khớp vai, đùi, cẳng của robot lần lượt là:

$$a_1 = 56.84 \text{ (mm)}$$

$$a_2 = 114 \text{ (mm)}$$

$$a_3 = 170.74 \text{ (mm)}$$

Ta thành lập được bảng DH:

Link	a	$\alpha$	d	$\theta$
1	$a_1$	$\pi/2$	0	$\theta_1$
2	$a_2$	0	0	$\theta_2$
3	$a_2$	0	0	$\theta_3$

Bảng 2.2 Bảng DH

Xác định các ma trận chuyển đổi thuần nhất:

$${}^0T_1 = \begin{bmatrix} c_1 & 0 & s_1 & a_1 c_1 \\ s_1 & 0 & -c_1 & a_1 s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Kết quả ma trận vecto cuối:

$${}^0T_3 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & c_1(a_1 + a_2 c_2 + a_3 c_{23}) \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & s_1(a_1 + a_2 c_2 + a_3 c_{23}) \\ s_{23} & c_{23} & 0 & a_2 s_2 + a_3 s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Với:  $c_1 = \cos(\theta_1)$ ;  $s_1 = \sin(\theta_1)$ ;  $c_{12} = \cos(\theta_1 + \theta_2)$ ;

$$s_{12} = \sin(\theta_1 + \theta_2); \dots$$

Ta được vị trí khâu cuối:

$$\begin{cases} p_x = c_1(a_1 + a_2c_2 + a_3c_{23}) \\ p_y = s_1(a_1 + a_2c_2 + a_3c_{23}) \\ p_z = a_2s_2 + a_3s_{23} \end{cases}$$

### 2.3.2. Động học nghịch (Inverse Kinematics)

Động học là việc nghiên cứu về chuyển động mà không xem xét đến nguyên nhân của chuyển động, chẳng hạn như lực và mô-men xoắn. Động học nghịch là việc sử dụng các phương trình động học để xác định chuyển động của robot để đạt được vị trí mong muốn. Ví dụ, để thực hiện việc lấy vật tự động từ các ngăn, một tay robot được sử dụng trong dây chuyền sản xuất cần phải thực hiện chuyển động chính xác từ vị trí ban đầu đến vị trí mong muốn giữa các ngăn và máy sản xuất. Đầu bám của cánh tay robot được gọi là bộ kết thúc. Cấu hình của robot là danh sách các vị trí khớp nằm trong giới hạn vị trí của mô hình robot và không vi phạm bất kỳ ràng buộc nào mà robot đang có.

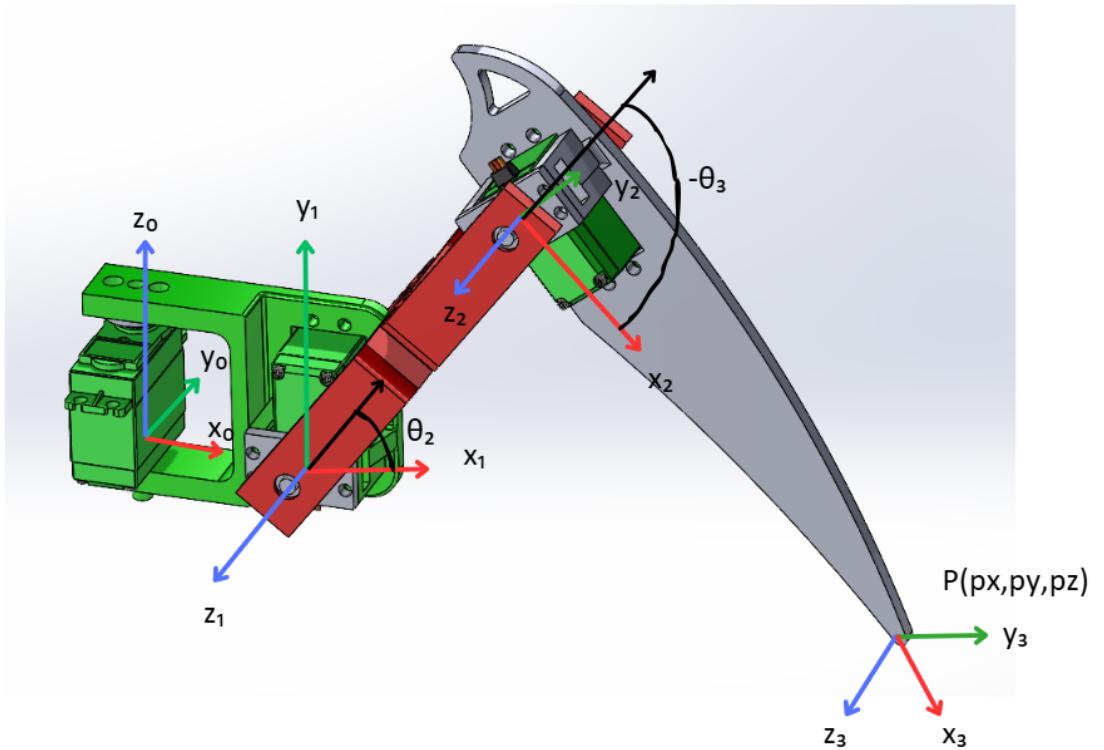
Với vị trí bộ kết thúc mong muốn của robot, động học nghịch (IK) có thể xác định một cấu hình khớp thích hợp để bộ kết thúc di chuyển đến vị trí mục tiêu.

Giả bài toán động học nghịch dựa vào ma trận vector cuối vừa tính được ở mục “Động học thuận”.

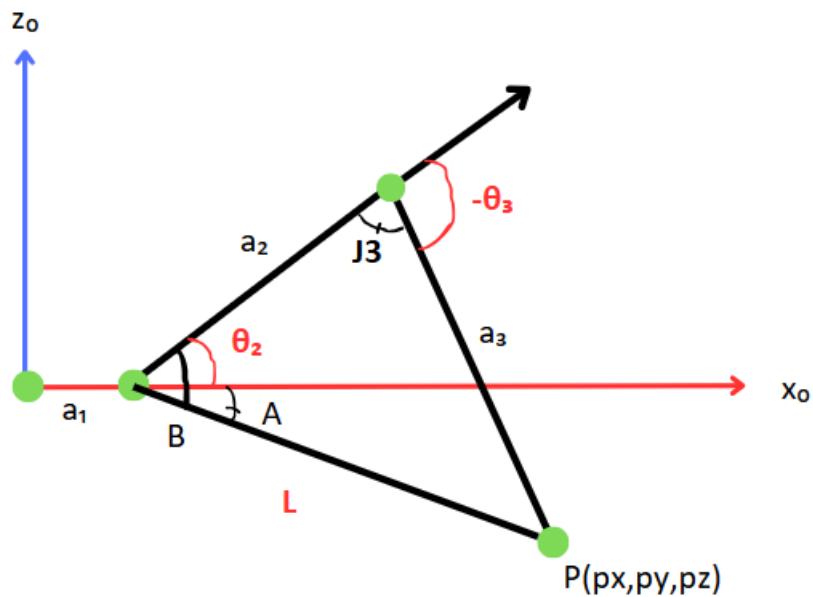
$$\begin{cases} p_x = c_1(a_1 + a_2c_2 + a_3c_{23}) \\ p_y = s_1(a_1 + a_2c_2 + a_3c_{23}) \\ p_z = a_2s_2 + a_3s_{23} \end{cases}$$

Ta có:

$$\begin{aligned} \frac{p_y}{p_x} &= \frac{s_1(a_1 + a_2c_2 + a_3c_{23})}{c_1(a_1 + a_2c_2 + a_3c_{23})} = \frac{s_1}{c_1} = \tan_1 \\ \Rightarrow \theta_1 &= \tan^{-1}\left(\frac{p_y}{p_x}\right) \end{aligned}$$



Hình 2.6 Góc theta 2 và theta 3 của robot trong không gian 3D



Hình 2.7 Biểu diễn góc theta 2 và theta 3 của robot trong mặt phẳng 2D

Ta có:

$$L = \sqrt{(px - a_1 c_1)^2 + (py - a_1 s_1)^2 + pz^2}$$

$$J3 = \cos^{-1} \left( \frac{a_2^2 + a_3^2 - L^2}{2a_2 a_3} \right)$$

$$\Rightarrow \theta_3 = J3 - \pi$$

$$B = \cos^{-1} \left( \frac{L^2 + a_2^2 - a_3^2}{2a_2 L} \right)$$

$$A = \tan^{-1} \left( \frac{-pz}{px - a_1 c_1} \right)$$

$$\Rightarrow \theta_2 = B - A$$

## 2.4. Bộ điều khiển PID

Một trong những nguyên tắc cơ bản của kỹ thuật điều khiển hệ thống là phương pháp điều khiển PID (Proportional-Integral-Derivative). Bộ điều khiển PID sẽ tính toán giá trị "sai số" là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào.

Bộ điều khiển PID rời rạc được sử dụng trong nhiều ứng dụng điều khiển hệ thống. Cụ thể trong các hệ thống robot, dữ liệu được lấy mẫu theo chu kỳ nhất định và bộ điều khiển PID tính toán tính hiệu điều khiển và tính toán sai số tại các khoảng thời gian đều nhau được xác định bởi chu kỳ lấy mẫu. Chúng ta có hàm chuyển đổi của bộ điều khiển PID rời rạc dưới dạng sau đây

$$G(z) = \frac{U(z)}{E(z)} = K_p + \frac{K_I T}{2} \frac{z+1}{z-1} + \frac{K_D}{T} \frac{z-1}{z} \quad (2.1)$$

$$G(z) = \frac{\left( K_p + \frac{K_I T}{2} + \frac{K_D}{T} \right) + \left( -K_p + \frac{K_I T}{2} - \frac{2K_D}{T} \right) z^{-1} + \frac{K_D}{T} z^{-2}}{1 - z^{-1}} \quad (2.2)$$

Đặt:

$$a_0 = K_p + \frac{K_I T}{2} + \frac{K_D}{T}$$

$$a_1 = -K_p + \frac{K_I T}{2} - \frac{2K_D}{T}$$

$$a_2 = \frac{K_D}{T}$$

Ta có:

$$G(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - z^{-1}} \quad (2.3)$$

Từ đó, chúng ta có thể tính toán giá trị đầu ra của bộ điều khiển PID  $u(k)$  khi giá trị đầu vào là  $e(k)$  như sau:

$$u(k) = G(z)e(k) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - z^{-1}} e(k) \quad (2.4)$$

Thay các giá trị của  $a_0$ ,  $a_1$  và  $a_2$  vào phương trình (2.8) và áp dụng tính chất dịch chuyển thời gian qua biến đổi Z, chúng ta có:

$$u(k) = u(k-1) + \left( K_p + \frac{K_I T}{2} + \frac{K_D}{T} \right) e(k) + \left( -K_p + \frac{K_I T}{2} - \frac{2K_D}{T} \right) e(k-1) + \frac{K_D}{T} e(k-2) \quad (2.5)$$

Chúng ta có thể viết lại phương trình (2.5) như sau:

$$u(k) = u(k-1) + P_{part} + I_{part} + D_{part}$$

Với:

$$P_{part} = K_P(e(k) - e(k-1))$$

$$I_{part} = \frac{K_I T}{2}(e(k) + e(k-1))$$

$$D_{part} = \frac{K_D}{T}(e(k) - 2e(k-1) + e(k-2))$$

$P_{part}$ : Khâu tỉ lệ (đôi khi còn được gọi là *độ lợi*) làm thay đổi giá trị đầu ra, tỉ lệ với giá trị sai số hiện tại. Đáp ứng tỉ lệ có thể được điều chỉnh bằng cách nhân sai số đó với một hằng số  $K_p$ , được gọi là hệ số tỉ lệ. Hệ số của khâu tỉ lệ lớn là do thay đổi lớn ở đầu ra mà sai số thay đổi nhỏ. Nếu hệ số của khâu tỉ lệ quá cao, hệ thống sẽ không ổn định. Ngược lại, hệ số nhỏ là do đáp ứng đầu ra nhỏ trong khi sai số đầu vào lớn, và làm cho bộ điều khiển kém nhạy, hoặc đáp ứng chậm. Nếu Hệ số của khâu tỉ lệ quá thấp, tác động điều khiển có thể sẽ quá bé khi đáp ứng với các nhiễu của hệ thống.

$I_{part}$ : Phân phối của khâu tích phân (đôi khi còn gọi là *reset*) tỉ lệ thuận với cả biên độ sai số lẫn quãng thời gian xảy ra sai số. Tổng sai số tức thời theo thời gian (tích phân sai số) cho ta tích lũy bù đũa được hiệu chỉnh trước đó. Khâu tích phân (khi cộng thêm khâu tỉ lệ) sẽ tăng tốc chuyển động của quá trình tới điểm đặt và khử số dư sai số ổn định với một tỉ lệ chỉ phụ thuộc vào bộ điều khiển. Tuy nhiên, vì khâu tích phân là đáp ứng của sai số tích lũy trong quá khứ, nó có thể khiến giá trị hiện tại vọt lố qua giá trị đặt (ngang qua điểm đặt và tạo ra một độ lệch với các hướng khác).

$D_{part}$ : Khâu vi phân làm chậm tốc độ thay đổi của đầu ra bộ điều khiển và đặc tính này là đáng chú ý nhất để đạt tới điểm đặt của bộ điều khiển. Từ đó, điều khiển vi phân được sử dụng để làm giảm biên độ vọt lố được tạo ra bởi thành phần tích phân và tăng cường độ ổn định của bộ điều khiển hỗn hợp.

Tuy nhiên, phép vi phân của một tín hiệu sẽ khuếch đại nhiễu và do đó khâu này sẽ nhạy hơn đối với nhiễu trong sai số, và có thể khiến quá trình trở nên không ổn định nếu nhiễu và độ lợi vi phân đủ lớn.

Một yếu tố quan trọng trong việc đạt được hiệu suất tốt nhất từ một bộ điều khiển PID là điều chỉnh tinh chỉnh các hệ số PID ( $K_p$ ,  $K_i$  và  $K_d$ ). Lấy vào xét các biến như thời gian lên, độ vọt lô, thời gian xác lập và sai số xác lập, việc điều chỉnh là quá trình xác định các giá trị phù hợp cho các hệ số để cung cấp phản ứng hệ thống mong muốn. Cho đến khi đạt được một mức độ hiệu suất phù hợp, các hệ số được điều chỉnh lại lặp đi lặp lại trong quá trình điều chỉnh

<b>Closed-loop response</b>	<b>Thời gian lên</b>	<b>Vọt lô</b>	<b>Thời gian xác lập</b>	<b>Sai số xác lập</b>
<b><math>K_p</math></b>	Giảm	Tăng	Ít thay đổi	Giảm
<b><math>K_i</math></b>	Giảm	Tăng	Tăng	Loại bỏ
<b><math>K_d</math></b>	Thay đổi ít	Giảm	Giảm	Thay đổi ít

Bảng 2.3 Đặc điểm của các khâu trong điều khiển vòng kín

## 2.5. Sensor Fusion

### 2.5.1. IMU (Inertial Measurement Unit) và Magnetometer

IMU là viết tắt của Inertial Measurement Unit. Đây là thiết bị kết hợp giữa cảm biến Gyroscope còn gọi là cảm biến con quay hồi chuyển. Và cảm biến Accelerometer còn gọi là cảm biến gia tốc. IMU có chức năng là cung cấp thông tin về độ nghiêng và góc quay. Để giúp hệ thống cân bằng một cách tự động.

Accelerometer: Cảm biến gia tốc là một cảm biến đo gia tốc của bản thân module và thường sẽ có 3 trục xyz ứng với 3 chiều không gian. Ở trạng thái tĩnh, cảm biến gia tốc đo lực của trọng lực.

Gyroscope: cảm biến góc quay đo vận tốc góc quay quanh ba trục của nó. gyro chỉ đo tốc độ quay chứ không đo trực tiếp góc quay, nên khi quay module một góc nào đó rồi dừng, giá trị của gyro sẽ tăng lên rồi hạ xuống về 0.

Magnetometer: một cảm biến từ, đơn giản là đo lường các trường từ. Với việc hiệu chuẩn đúng trong một trường từ tĩnh, nó có thể phát hiện sự biến đổi trong trường từ của Trái Đất. Thông qua những biến đổi đó, nó tìm ra vector hướng về phía Bắc từ trường từ của Trái Đất, cung cấp cho nó một hướng tuyệt đối.

Giống như những cảm biến khác, IMU và từ kế cũng có những lỗi nhất định mỗi khi sử dụng. Cụ thể:

Accelerometer: Acc luôn có offset ở trên mỗi trục. Chính điều này đã khiến giá trị đo được thường bị lệch đi so với thực tế. Ngoài ra, giá trị này thường khá nhiều dẫn đến tình trạng đọc trở nên khó khăn hơn.

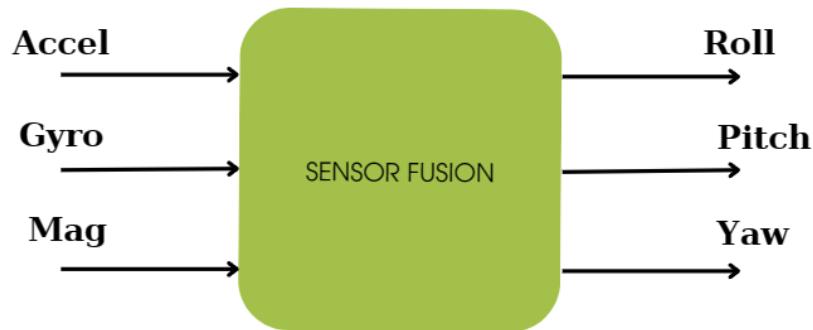
Gyroscope: Tuy ít nhiều hơn Acc nhưng Gyro cũng có offset. Thêm một vấn đề nữa ở Gyro đó chính là drift. Nghĩa là bias có thể thay đổi chậm theo thời gian.

Magnetometer có thể dễ bị ảnh hưởng bởi các yếu tố như soft iron và hard iron, dẫn đến sai số trong dữ liệu đo lường. Dữ liệu từ cảm biến từ sẽ bị biến đổi so với trạng thái thực tế của từ trường, làm giảm độ chính xác của dữ liệu và ảnh hưởng đến tính đáng tin cậy của hệ thống.

Việc chỉ sử dụng một loại cảm biến có thể gây ra các hạn chế và sai sót trong việc đo lường và đánh giá trạng thái chuyển động của một hệ thống. Điều này có thể bao gồm hạn chế dữ liệu, dễ bị ảnh hưởng bởi nhiều, thiếu thông tin cân đối và khả năng phục hồi và bảo mật yếu. Vì vậy cần kết hợp các thông tin của cảm biến để tạo ra đầu ra tin cậy được gọi là Sensor Fusion.

Sensor Fusion là quá trình kết hợp thông tin từ nhiều cảm biến khác nhau để tạo ra một hình ảnh toàn diện và chính xác về tình trạng chuyển động, hướng và vị trí của một hệ thống. Trong Sensor Fusion, dữ liệu từ các cảm biến như accelerometer, gyroscope và magnetometer được tổng hợp và phân tích để tạo

ra một hình ảnh thực sự đại diện cho trạng thái chuyển động của hệ thống. Cảm biến tốc độ đo lực tác động lên hệ thống, cảm biến góc đo vận tốc quay quanh các trục và cảm biến từ đo lường các trường từ. Khi kết hợp thông tin từ các cảm biến này, Sensor Fusion có thể loại bỏ nhiễu, hiệu chỉnh sai số và tạo ra một dự đoán hoặc ước tính chính xác về trạng thái chuyển động của hệ thống.



Hình 2.8 Sensor Fusion IMU và Magnetometer

## 2.6. Bộ lọc Madgwick

Phương pháp Madgwick AHRS (Attitude and Heading Reference System) là một phương pháp được phát triển bởi Sebastian Madgwick để ước tính và theo dõi hướng và vị trí của một vật thể trong không gian ba chiều (3D) dựa trên dữ liệu từ các cảm biến như gyroscope, accelerometer và magnetometer.

Được sử dụng trong khối Sensor Fusion, thuật toán này cung cấp một cách hiệu quả và chính xác để tính toán quaternions, biểu diễn hướng xoay trong không gian ba chiều, từ dữ liệu từ cảm biến.

### 2.6.1. Quaternion và phép quay không gian

Bộ lọc Madgwick thực hiện tính toán trên không gian quaternion cho phép biểu diễn hướng xoay trong không gian ba chiều. Thay vì sử dụng 3 góc Euler để biểu diễn quan hệ tương đối của hai hệ quy chiếu (Frames). Quaternion là

một vectơ  $4 \times 1$  trong đó các phần tử bao gồm phần vô hướng  $s$  và phần vectơ  $v$ .

$$q = \begin{bmatrix} s \\ \vec{v} \end{bmatrix} = \begin{bmatrix} s \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

$s$ : là phần vô hướng

$v$ : là phần vector

Như vậy Quaternion có thể biểu diễn một góc quay duy nhất trong không gian mà không gặp vấn đề trùng lặp như gimbal lock trong Euler angles. Gimbal lock xảy ra hai trong ba gimbal trùng lặp hoặc sắp xếp gần nhau đến mức làm mất đi một bậc tự do. hai trong ba gimbal trùng lặp hoặc sắp xếp gần nhau đến mức làm mất đi một bậc tự do làm một trong ba góc quay trở thành tương đương với góc của một gimbal khác. Điều này làm giảm độ phức tạp của hệ thống và gây ra sự không xác định trong biểu diễn góc. Một trong những lợi ích lớn nhất của việc sử dụng quaternion là khả năng biểu diễn mọi góc quay một cách liền mạch và không gặp phải các vấn đề liên quan đến trùng lặp hoặc không xác định.

Một số công thức liên quan đến Quaternion:

Chuẩn hóa vector  $q$  theo Norm:

$$|q| = \sqrt{q_s^2 + q_x^2 + q_y^2 + q_z^2} \quad (2.x)$$

$$\|q\| = \frac{q}{|q|} \quad (2.x)$$

Phép nhân:

$$q_1 \otimes q_2 = q = \begin{bmatrix} s \\ \vec{v} \end{bmatrix} = \begin{bmatrix} s_1 \cdot s_2 - \vec{v}_1 \circ \vec{v}_2 \\ s_1 \cdot \vec{v}_2 + s_2 \cdot \vec{v}_1 + \vec{v}_1 \times \vec{v}_2 \end{bmatrix} \quad (2.6)$$

Từ vector Quaternion ta có thể chuyển đổi sang ma trận xoay Euler Angles:

$$T = \begin{bmatrix} q_s^2 + q_x^2 - q_y^2 - q_z^2 & 2 \cdot (q_x \cdot q_y - q_z \cdot q_s) & 2 \cdot (q_x \cdot q_z + q_y \cdot q_s) \\ 2 \cdot (q_x \cdot q_y + q_z \cdot q_s) & q_s^2 - q_x^2 + q_y^2 - q_z^2 & 2 \cdot (q_y \cdot q_z - q_x \cdot q_s) \\ 2 \cdot (q_x \cdot q_z - q_y \cdot q_s) & 2 \cdot (q_y \cdot q_z + q_x \cdot q_s) & q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \quad (2.7)$$

Từ ma trận xoay Euler Angles (còn được gọi là DCM) (2.x). Tính toán các giá trị roll, pitch, yaw.

$$\begin{aligned} {}^A R_{ZX'Z''}(\phi, \theta, \psi) &= R_Z(\phi)R_{X'}(\theta)R_{Z''}(\psi) \\ &= \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c\phi c\psi - s\phi c\theta s\psi & -c\phi s\psi - s\phi c\theta c\psi & s\phi s\theta \\ s\phi c\psi + c\phi c\theta s\psi & -s\phi s\psi + c\phi c\theta c\psi & -c\phi s\theta \\ s\theta s\psi & s\theta c\psi & c\theta \end{bmatrix} \quad (2.8) \end{aligned}$$

Từ (2.x) Ta có :

$$\begin{aligned} {}^A R_{ZX'Z''}(\phi, \theta, \psi) &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c\phi c\psi - s\phi c\theta s\psi & -c\phi s\psi - s\phi c\theta c\psi & s\phi s\theta \\ s\phi c\psi + c\phi c\theta s\psi & -s\phi s\psi + c\phi c\theta c\psi & -c\phi s\theta \\ s\theta s\psi & s\theta c\psi & c\theta \end{bmatrix} \\ &\left. \begin{array}{l} r_{13}^2 + r_{23}^2 = (s\theta)^2 \\ r_{33} = c\theta \end{array} \right\} \Rightarrow \theta = ATAN2(\pm\sqrt{r_{11}^2 + r_{21}^2}, r_{33}) \\ &\left. \begin{array}{l} \text{If } r_{13}^2 + r_{23}^2 \neq 0 \text{ (i.e., } s\theta \neq 0) \\ r_{31}/s\theta = s\psi, r_{32}/s\theta = c\psi \end{array} \right\} \Rightarrow \psi = ATAN2(r_{31}/s\theta, r_{32}/s\theta) \\ &\phi = ATAN2(r_{13}/s\theta, -r_{23}/s\theta) \quad (2.9) \end{aligned}$$

Như vậy từ vector quaternion ta có thể tính toán được chính xác góc roll, pitch, yaw mà không gặp phải tình trạng gimbal lock ở hệ Euler.

## 2.6.2. Tính toán Bộ lọc Madgwick

Bộ lọc Madgwick hoạt động trong hai hệ tọa độ chính:

Hệ tọa độ chính cảm biến: Là hệ tọa độ gắn liền với vật thể, với các trục x, y và z được căn chỉnh với các trục của cảm biến.

Hệ tọa độ Trái đất: Là hệ tọa độ tham chiếu cố định, thường được căn chỉnh với các vector trọng lực và từ trường của Trái đất.

### 2.6.2.1. Tính toán với Gyroscope

Ta xây dựng một phương pháp tính toán Quaternion để liên hệ tốc độ quay của vật thể (được đo bằng con quay hồi chuyển) với trạng thái của đối tượng. Tính toán tốc độ thay đổi hướng của vật :

$$\overset{S}{E}\dot{\hat{\mathbf{q}}} = \frac{1}{2} \overset{S}{\omega} \hat{\mathbf{q}} \otimes \overset{S}{\omega}$$

Với:

$\overset{S}{E}\dot{\hat{\mathbf{q}}}$  là tốc độ thay đổi hướng của khung trái đất so với khung cảm biến

$$S_{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$
 là tốc độ góc đọc từ Gyroscope

Để cập nhật tốc độ thay đổi sau mỗi khoảng thời gian  $\Delta t$ :

$$\overset{S}{E}\mathbf{q}_{\omega,t} = \overset{S}{E}\hat{\mathbf{q}}_{est,t-1} + \overset{S}{E}\dot{\hat{\mathbf{q}}}_{\omega,t} \Delta t$$

### 2.6.2.2. Tính toán với Accelerometer và Magnetometer

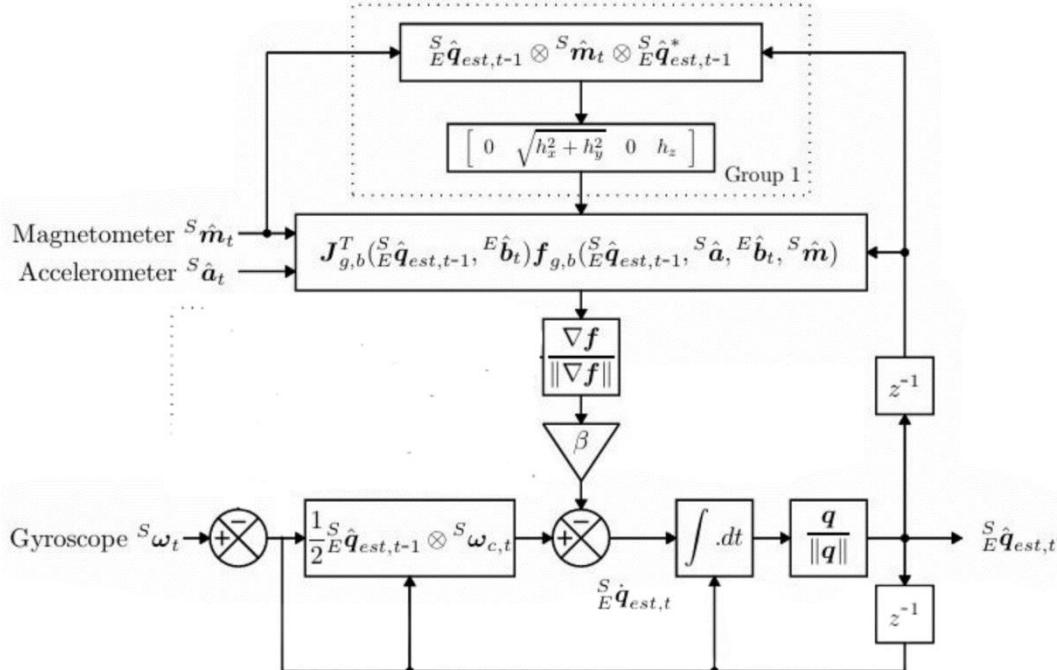
Trong trường hợp này, mục tiêu là giảm thiểu sai số giữa hướng của trường từ được đo trong không gian sensor ( $\hat{\mathbf{d}}^S$ ) và hướng của trường từ đã biết trong không gian trái đất ( $\hat{\mathbf{d}}^E$ ). Mục đích cuối cùng là ước tính một hướng chính xác cho khung tọa độ của cảm biến (Sensor frame) so với khung tọa độ của Trái đất (Earth frame). Điều này đồng nghĩa với việc tối thiểu hóa hàm mất mát:

$$\min_{\substack{S \\ E}} \hat{\mathbf{q}} \in \mathbb{R}^4 \mathbf{f}(\hat{\mathbf{q}}, \hat{\mathbf{d}}, \hat{\mathbf{s}})$$

Gradient descent là một trong những phương pháp tối ưu hóa đơn giản và hiệu quả được sử dụng rộng rãi trong các bài toán tối ưu hóa. Phương pháp này hoạt động bằng cách lặp đi lặp lại việc cập nhật giá trị của  $\hat{\mathbf{q}}$  dựa trên gradient của hàm mất mát. Cụ thể, nó di chuyển  $\hat{\mathbf{q}}$  theo hướng ngược với gradient của hàm mất mát để tìm ra điểm cực tiểu.

$$\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}_k - \mu \frac{\nabla \mathbf{f}(\hat{\mathbf{q}}_k, \hat{\mathbf{d}}, \hat{\mathbf{s}})}{\|\nabla \mathbf{f}(\hat{\mathbf{q}}_k, \hat{\mathbf{d}}, \hat{\mathbf{s}})\|}, \quad k = 0, 1, 2, \dots, n$$

Như vậy ta có sơ đồ hoàn chỉnh của bộ lọc để ước lượng thông số q tiếp theo sau khoảng thời gian  $\Delta t$ .



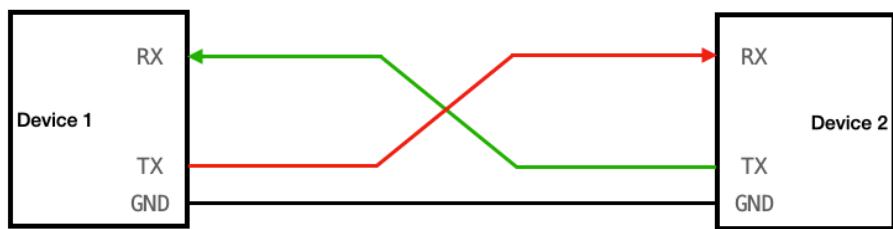
Hình 2.9 Sơ đồ bộ lọc Madgwick

## 2.7. Giao thức truyền thông

### 2.7.1. Chuẩn giao tiếp UART

UART (Universal Asynchronous Receiver-Transmitter – Bộ truyền nhận dữ liệu không đồng bộ) là một giao thức truyền thông phần cứng dùng giao tiếp nối tiếp không đồng bộ và có thể cấu hình được tốc độ

Dữ liệu truyền đến và đi từ UART song song với thiết bị điều khiển. Khi tín hiệu gửi trên chân Tx (truyền), bộ giao tiếp Uart đầu tiên sẽ dịch thông tin song song này thành dạng nối tiếp và sau đó truyền tới thiết bị nhận. Chân Rx (nhận) của UART thứ 2 sẽ biến đổi nó trở lại thành dạng song song để giao tiếp với các thiết bị điều khiển.



Hình 2.10 Giao tiếp UART giữa hai thiết bị

Dữ liệu truyền qua UART sẽ đóng thành các gói (packet) được truyền và nhận qua các đường truyền này dưới dạng các khung dữ liệu (data frame) có cấu trúc chuẩn, với một bit bắt đầu (start bit), 5 – 9 bit dữ liệu (tùy thuộc vào bộ UART), một bit kiểm tra chẵn lẻ (parity bit) và một hoặc nhiều bit dừng (stop bit).

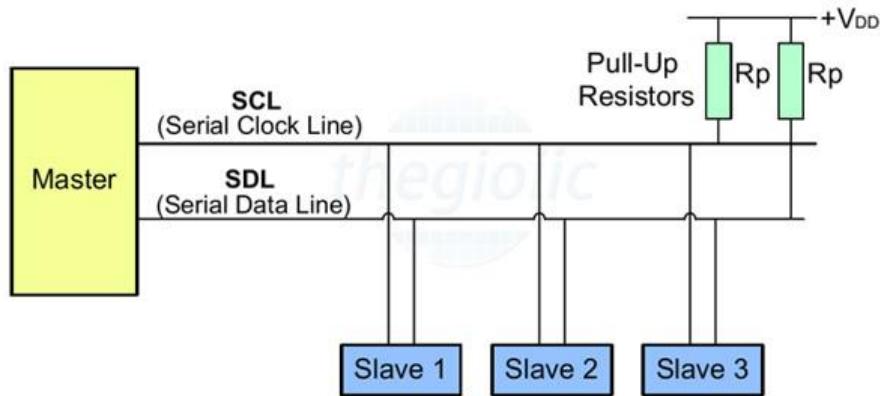
Start Bit ( 1 bit )	Data Frame ( 5 to 9 Data Bits )	Parity Bits ( 0 to 1 bit )	Stop Bits ( 1 to 2 bits )
------------------------	------------------------------------	-------------------------------	------------------------------

Hình 2.11 UART packet

Giao tiếp UART được sử dụng rộng rãi trong nhiều ứng dụng khác nhau, bao gồm giao tiếp nối tiếp giữa các vi điều khiển, giao tiếp với các thiết bị ngoại vi, và trao đổi dữ liệu giữa máy tính và các hệ thống nhúng. Nó được biết đến với tính đơn giản, chi phí thấp và sự hỗ trợ rộng rãi trên các nền tảng và thiết bị khác nhau.

### 2.7.2. Chuẩn giao tiếp I2C

I2C hay IIC (Inter – Integrated Circuit) là 1 giao thức giao tiếp nối tiếp đồng bộ, sử dụng để truyền nhận dữ liệu giữa các IC với nhau chỉ sử dụng hai đường truyền tín hiệu.



Hình 2.12 Giao tiếp I2C giữa các thiết bị

I2C chỉ sử dụng hai dây để truyền dữ liệu giữa các thiết bị:

- SDA (Serial Data) - đường truyền cho master và slave để gửi và nhận dữ liệu.
- SCL (Serial Clock) - đường mang tín hiệu xung nhịp.

Các bit dữ liệu sẽ được truyền từng bit một theo một đường duy nhất (SDA) theo các khoảng thời gian đều đặn được thiết lập bởi 1 tín hiệu đồng hồ (SCL).

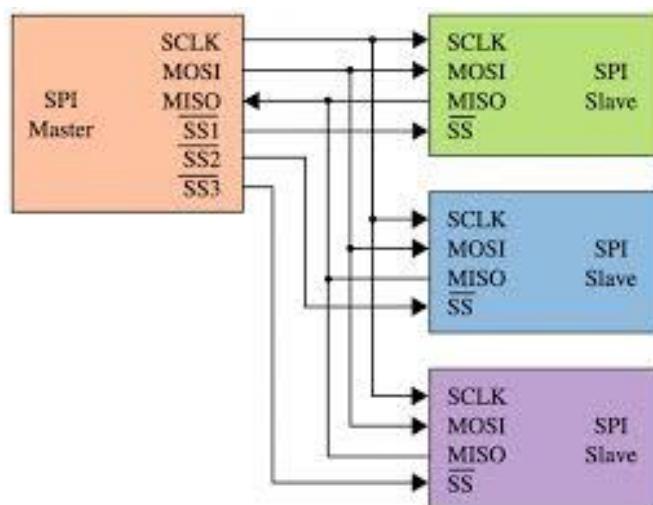
Với I2C, dữ liệu được truyền trong các tin nhắn. Tin nhắn được chia thành các khung dữ liệu. Mỗi tin nhắn có một khung địa chỉ chứa địa chỉ nhị phân của địa chỉ slave và một hoặc nhiều khung dữ liệu chứa dữ liệu đang được truyền. Thông điệp cũng bao gồm điều kiện khởi động và điều kiện dừng, các bit đọc / ghi và các bit ACK / NACK giữa mỗi khung dữ liệu

### 2.7.3. Chuẩn giao tiếp SPI

SPI (Serial Peripheral Interface) là một giao thức truyền thông đồng bộ được sử dụng phổ biến trong các hệ thống nhúng để kết nối các vi điều khiển, vi xử lý

và các thiết bị ngoại vi. Giao thức này cho phép truyền dữ liệu giữa các thiết bị theo cách đồng bộ, nhanh chóng và hiệu quả.

Cấu trúc cơ bản của SPI bao gồm một thiết bị chủ (master) và một hoặc nhiều thiết bị nô lệ (slave). Thiết bị chủ điều khiển quá trình truyền thông bằng cách phát tín hiệu đồng hồ và lựa chọn thiết bị muốn truyền dữ liệu tới. Mỗi thiết bị nô lệ được gắn với một dây MOSI (Master Out Slave In) để nhận dữ liệu từ thiết bị chủ, một dây MISO (Master In Slave Out) để truyền dữ liệu đến thiết bị chủ, và một dây đồng hồ (CLK) để đồng bộ hóa truyền thông.



Hình 2.13 Giao tiếp SPI giữa các thiết bị

Mỗi thiết bị Master và Slave đều có một thanh ghi dịch 8 bits (Shift Register), một bộ tạo xung nhịp (Clock Generator).

Khi Master truyền dữ liệu, Master truyền đi 8 bits dữ liệu vào thanh ghi dịch của nó, sau đó 8 bits dữ liệu được truyền theo đường tín hiệu MOSI sang thiết bị Slave. Tương tự khi Slave truyền dữ liệu, các bits trên thanh ghi dịch của Slave truyền đến Master thông qua đường tín hiệu MISO.

Bằng cách này, dữ liệu của hai thanh ghi sẽ được trao đổi với nhau. Việc đọc và ghi dữ liệu vào Slave diễn ra cùng một lúc nên tốc độ trao đổi dữ liệu diễn ra rất nhanh. Do đó, giao thức SPI là một giao thức rất có hiệu quả.

## 2.7.4. DMA (truy cập bộ nhớ trực tiếp)

Direct memory access viết tắt là DMA. Đây là một cơ chế truyền dữ liệu với tốc độ cao từ ngoại vi tới bộ nhớ và từ bộ nhớ tới bộ nhớ. Hay Direct memory access là một phương cho phép các thiết bị gửi hoặc nhận (Input/Output) dữ liệu một cách trực tiếp tới bộ nhớ và từ bộ nhớ mà không cần qua CPU.

STM32F401 hỗ trợ 2 DMA Controller có 16 streams (luồng) với 8 streams cho mỗi Controller, mỗi stream dành riêng để quản lý các yêu cầu truy cập bộ nhớ từ một hoặc nhiều thiết bị ngoại vi. Mỗi Stream có thể có tối đa 8 Channel và có một cơ chế phân xử để xử lý mức độ ưu tiên giữa các Channel.

Bộ điều khiển DMA thực hiện truyền/nhận bộ nhớ một cách trực tiếp: Với vai trò AHB Master, DMA có thể nắm quyền điều khiển bus matrix AHB để bắt đầu các quá trình truyền nhận:

- Peripheral to Memory
- Memory to Peripheral
- Memory to Memory

Vì vậy, muốn di chuyển dữ liệu từ thiết bị ngoại vi sang bộ điều khiển DMA, sang bộ nhớ. Chọn stream (1 - 8) - Sử dụng "DMA Request Mapping" bảng (dành riêng cho MCU) để tìm hiểu kênh hợp lệ cho luồng đó và thiết bị ngoại vi được chọn là gì.

**Table 20. DMA1 request mapping**

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	SPI3_RX		SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX		SPI3_TX
Channel 1	I2C1_RX		TIM7_UP		TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Channel 2	TIM4_CH1		I2S2_EXT_RX	TIM4_CH2	I2S2_EXT_TX	I2S3_EXT_RX	TIM4_UP	TIM4_CH3
Channel 3	I2S3_EXT_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_RX	I2C3_TX	TIM2_CH1	TIM2_UP TIM2_CH4	TIM2_UP TIM2_CH4
Channel 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
Channel 5			TIM3_CH4 TIM3_UP		TIM3_CH1 TIM3_TRIG	TIM3_CH2		TIM3_CH3
Channel 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2		TIM5_UP	
Channel 7		TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

Hình 2.14 DMA1 Request mapping

Table 21. DMA2 request mapping

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	ADC1		TIM8_CH1 TIM8_CH2 TIM8_CH3		ADC1		TIM1_CH1 TIM1_CH2 TIM1_CH3	
Channel 1		DCMI	ADC2	ADC2				DCMI
Channel 2	ADC3	ADC3				CRYP_OUT	CRYP_IN	HASH_IN
Channel 3	SPI1_RX		SPI1_RX	SPI1_TX		SPI1_TX		
Channel 4			USART1_RX	SDIO		USART1_RX	SDIO	USART1_TX
Channel 5		USART6_RX	USART6_RX				USART6_TX	USART6_TX
Channel 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
Channel 7		TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3			TIM8_CH4 TIM8_TRIG TIM8_COM

Hình 2.15 DMA2 Request mapping

## **Chương 3. THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG**

### **3.6. Mô hình robot**

#### **3.6.1. Thiết kế mô hình 3D SolidWorks**

Trong phần này, chúng ta sẽ tập trung vào việc tạo ra một mô hình robot nhện 4 chân có độ chính xác cao và có khả năng hoạt động theo kế hoạch đã đề ra. Bằng cách sử dụng phần mềm SolidWorks, chúng ta đã có khả năng tạo ra một bản thiết kế chi tiết của robot, từ cấu trúc cơ học của các chân đến việc gắn kết các linh kiện và các bộ phận quan trọng khác.

SolidWorks không chỉ giúp chúng ta tạo ra một mô hình 3D đẹp mắt mà còn cho phép chúng ta quan sát một cách trực quan cách từng khớp chuyển động và chuyển động tổng thể của cả 4 chân. Điều này rất quan trọng để đảm bảo rằng robot sẽ hoạt động một cách mượt lắc và hiệu quả khi được triển khai trong thực tế.

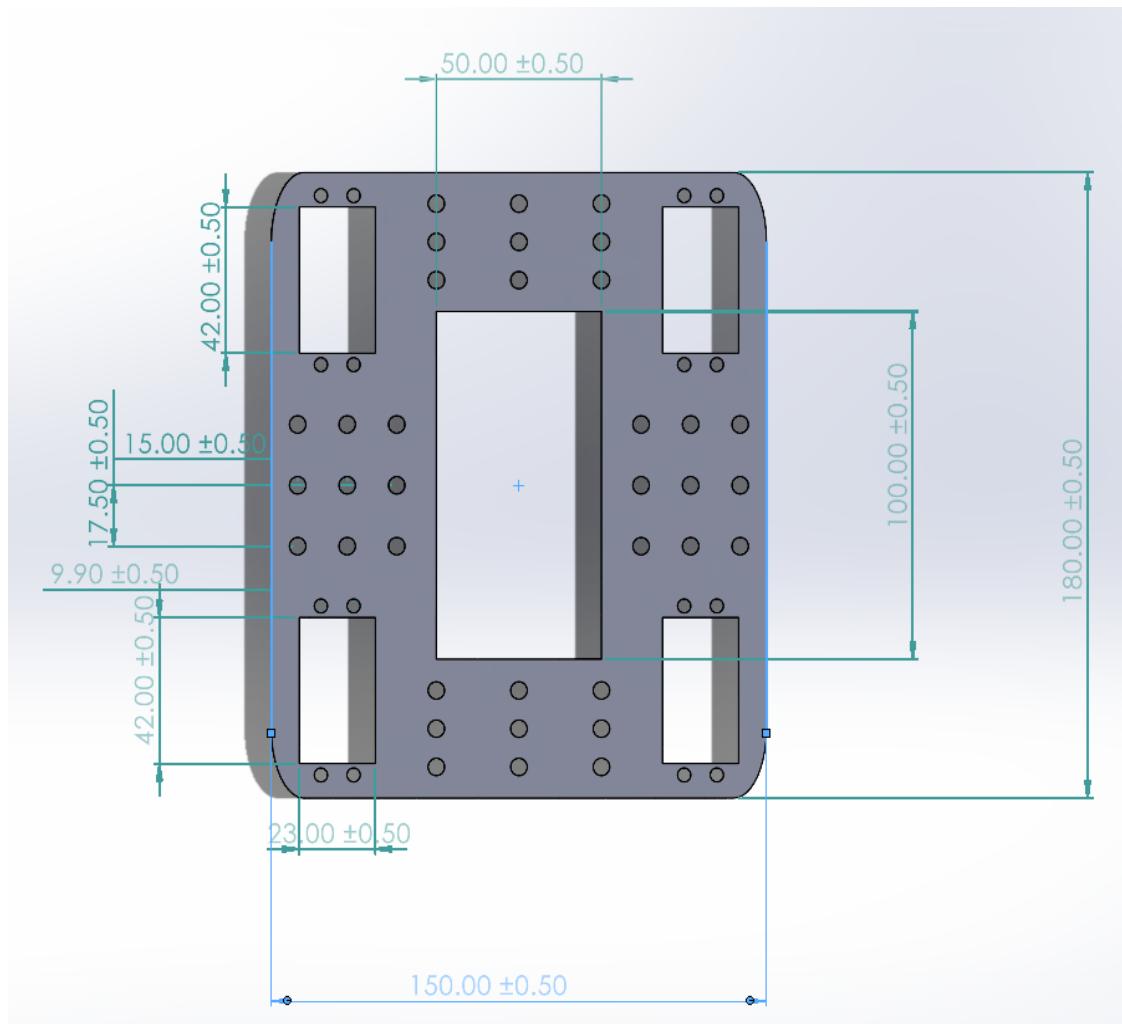
Bằng cách sử dụng SolidWorks, chúng ta có thể kiểm tra và điều chỉnh mọi khía cạnh của robot trước khi tiến hành sản xuất. Điều này giúp giảm thiểu rủi ro và chi phí do sửa chữa sau này. Ngoài ra, việc quan sát mô hình 3D trong SolidWorks cũng giúp chúng ta dễ dàng hiểu và giải quyết bất kỳ vấn đề nào mà có thể phát sinh trong quá trình phát triển.

Tóm lại, việc sử dụng SolidWorks đã giúp chúng ta không chỉ tạo ra một mô hình robot nhện 4 chân chính xác và hoạt động theo kế hoạch mà còn cung cấp cho chúng ta các công cụ quan trọng để kiểm tra, điều chỉnh và hiểu rõ hơn về robot trước khi đưa nó vào sản xuất và triển khai.

##### **3.6.1.1. Thân robot**

Trong mô hình robot, phần thân là bộ phận liên kết các chân robot và là nơi đặt các linh kiện, cảm biến, pin... nên được thiết kế một cách chắc chắn nhằm đảm bảo độ cứng cáp cho robot. Thân trên và thân dưới đều được thiết kế nhiều lỗ tròn với đường kính 4 (mm). Điều này nhằm mục đích bắt vít cho động cơ, đặt các trụ đồng nâng đỡ cho mạch PCB, dễ

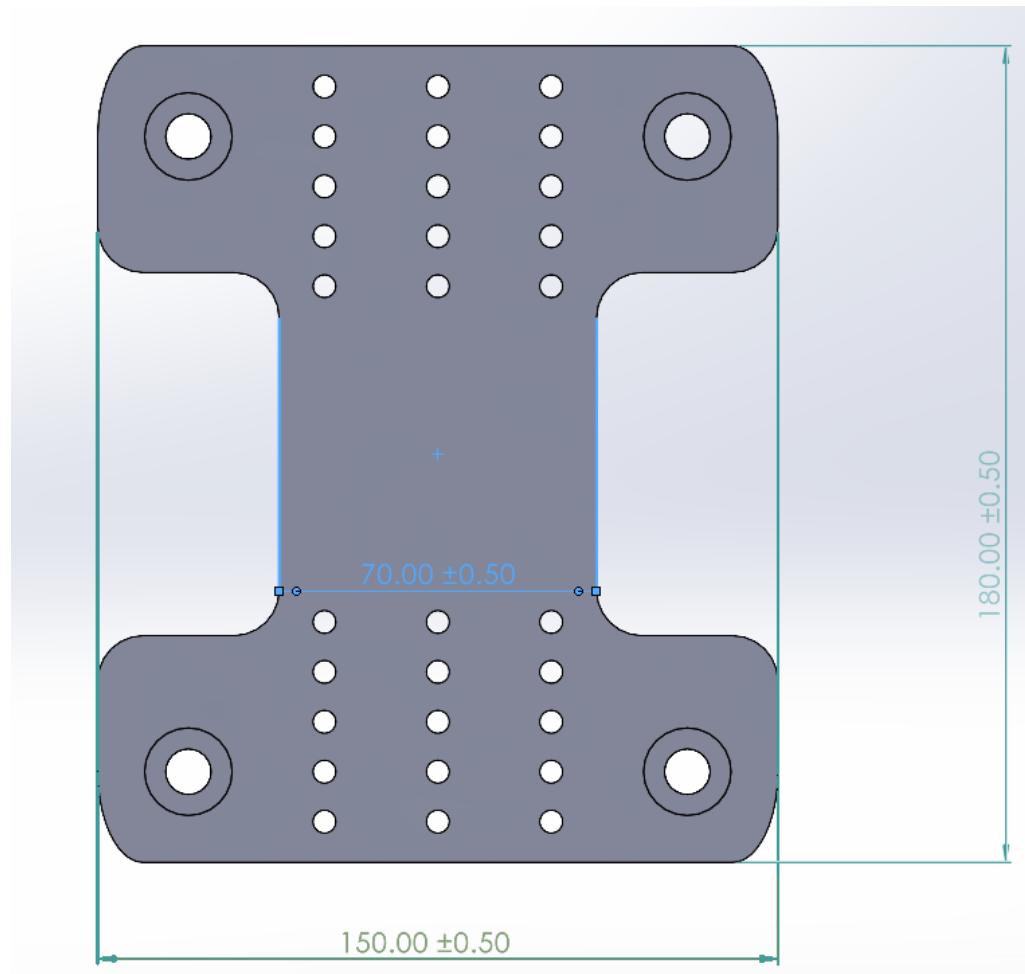
dùng gá cứng các linh kiện và phần nào giúp giảm bớt trọng lượng của robot. Phần thân của robot có kích thước 180 x 150 (mm):



Hình 3.1 Thân trên Robot

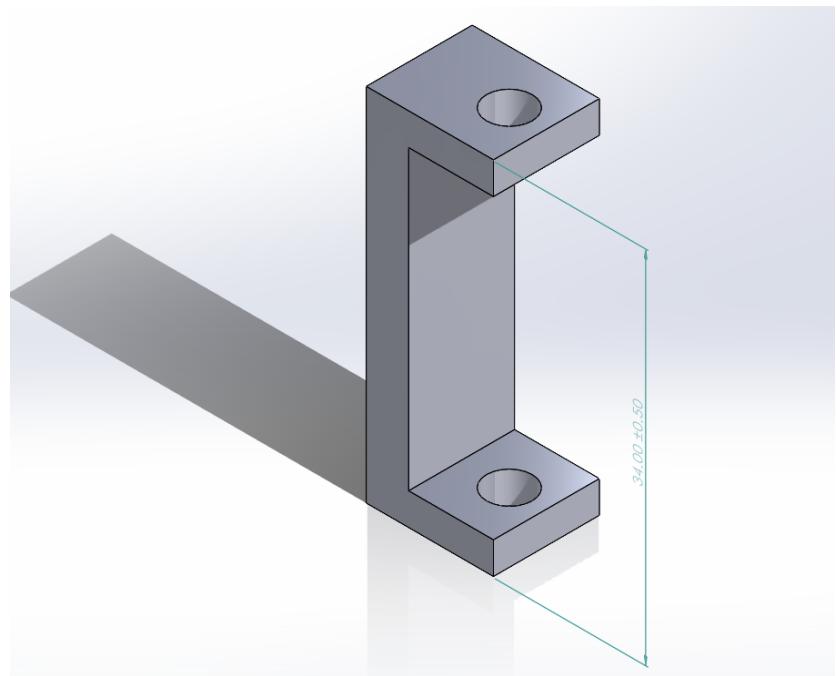
Thân trên Robot trống phần giữa để giảm trọng lượng và là nơi để cho các sợi dây kết nối luồn qua. Các lỗ tròn được sắp đặt hợp lí, cân đối phù hợp cho việc phát triển thêm các module phần cứng sau này.

Bốn động cơ servo được đặt đối xứng tại bốn góc của thân Robot để Robot có thể di chuyển một cách linh hoạt, cân đối.



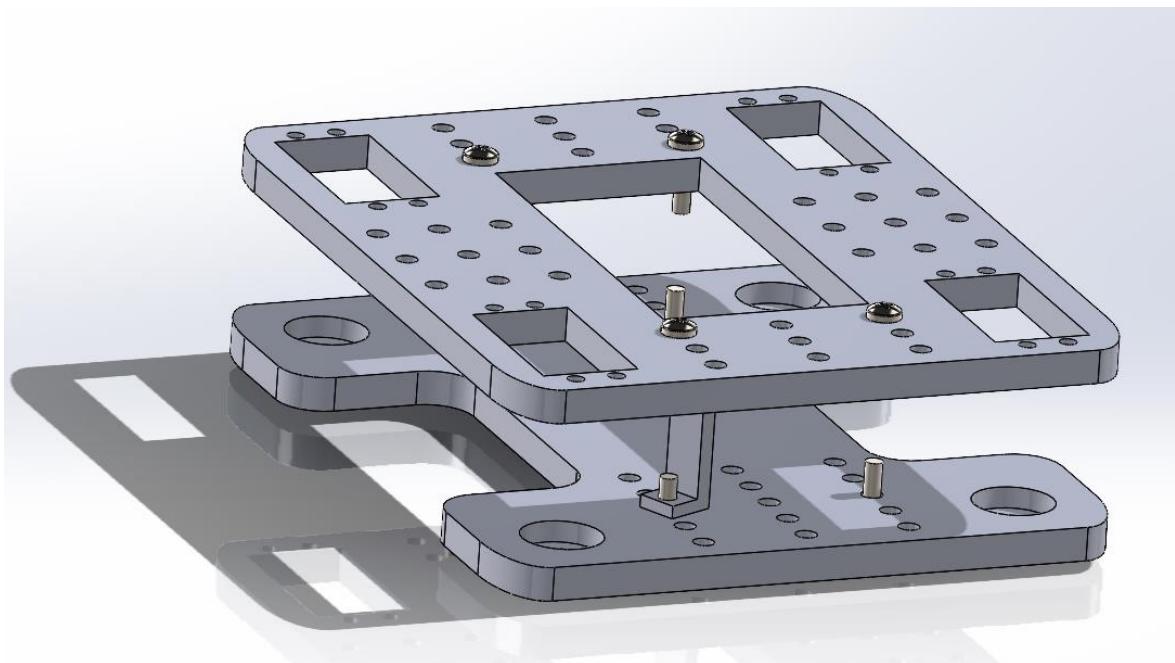
Hình 3.2 Thân dưới Robot

Thân dưới Robot có kích thước tương tự thân trên: 180 x 150 (mm). Tại bốn góc của Robot sẽ được đặt bốn bạc đạn giúp cho khớp của Robot xoay một cách trơn tru. Diện tích phần giữa được tiêu giảm tối thiểu, đủ để đặt module PWM, Buck DC-DC và pin nhằm giảm bớt trọng lượng Robot. Các lỗ tròn cũng được khoét đối xứng với thân trên để bắt vít cố định các module. Tăng độ chắc chắn và đảm bảo tính ổn định khi di chuyển của Robot.



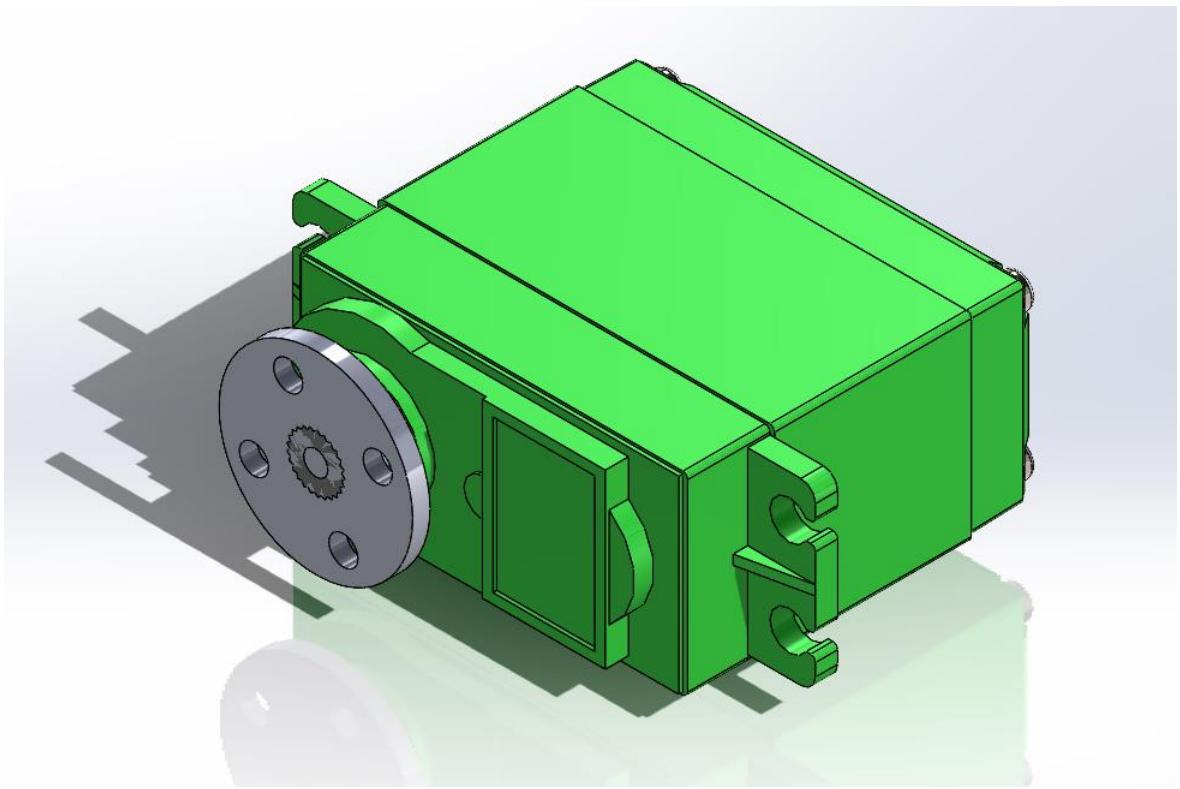
Hình 3.3 Trụ đỡ cho thân Robot

Robot sử dụng bốn trụ đỡ với chiều cao 34 (mm) để đặt động cơ và các module trong thân Robot.



Hình 3.4 Thân Robot hoàn chỉnh

### 3.6.1.2. Động cơ Servo

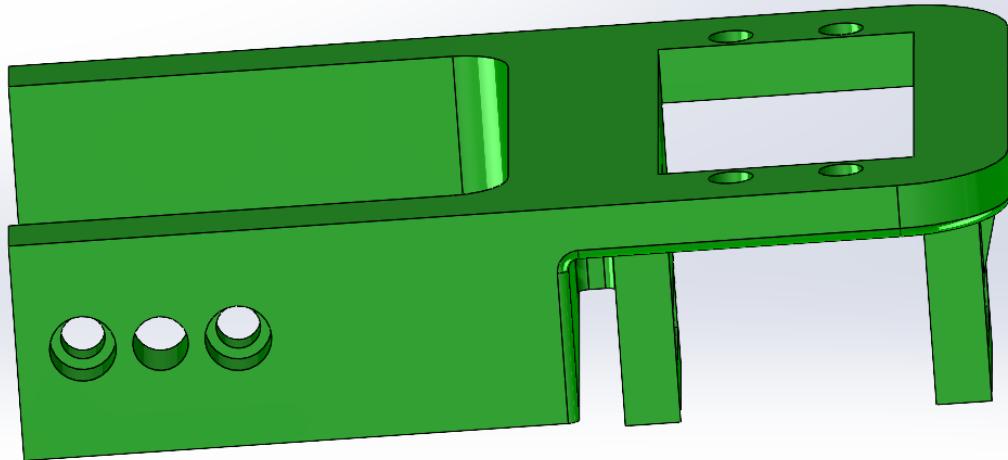


Hình 3.5 Động cơ Servo MG996R

Động cơ Servo MG996R được nhóm lựa chọn nhờ khả năng cung cấp mô-men xoắn lớn giúp điều khiển các chân robot và di chuyển một cách dễ dàng. Động cơ này có độ chính xác cao, dễ dàng điều chỉnh thông qua tín hiệu PWM từ các bo mạch điều khiển như STM32. Ngoài ra, kích thước và trọng lượng của MG996R phù hợp với yêu cầu của robot, không làm tăng quá nhiều tải trọng.

### 3.1.1.3. Chân robot

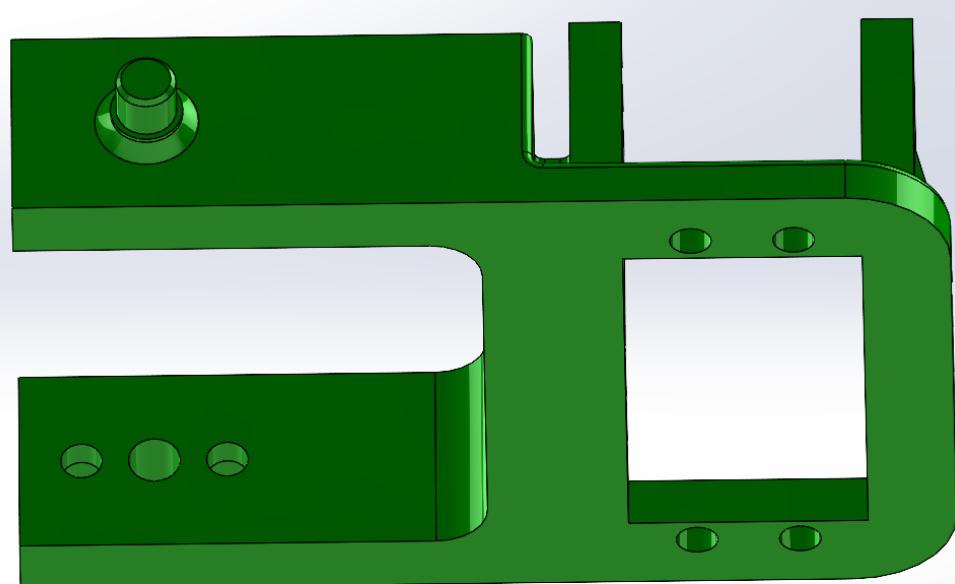
Chân của Robot được cấu tạo bởi 3 khâu chuyển động, để dễ hình dung ta sẽ gọi 3 khớp lần lượt từ thân Robot ra ngoài là: vai, đùi cẳng.



Hình 3.6 Mặt trên khớp vai Robot

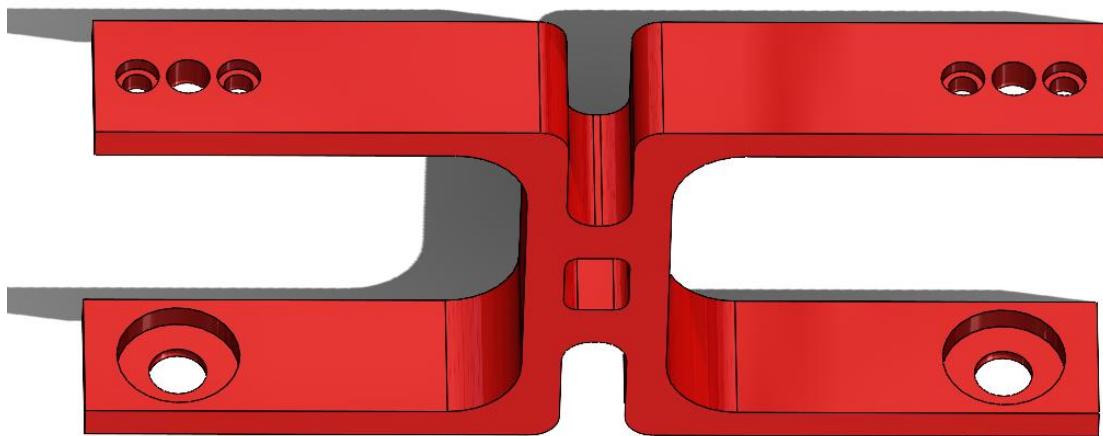
Mặt trên khớp vai Robot được khoét 3 lỗ để cố định với động cơ servo.

Khâu này sẽ liên kết 2 động cơ servo thứ nhất và thứ hai.



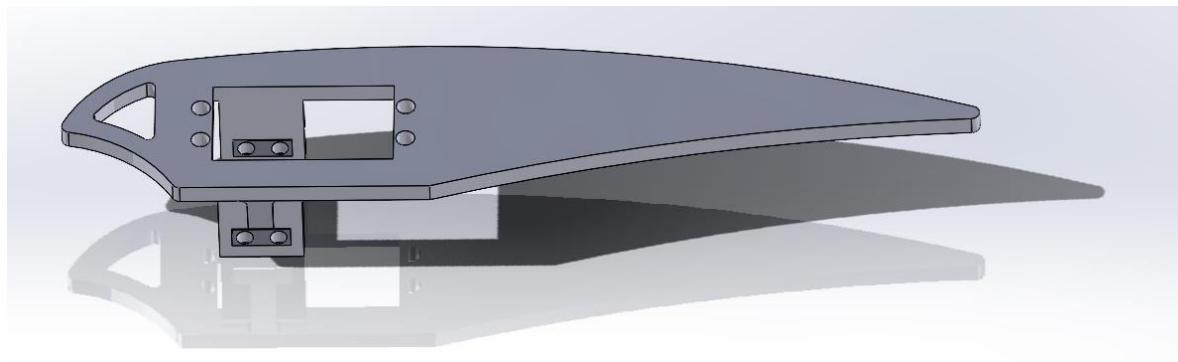
Hình 3.7 Mặt dưới khớp vai Robot

Mặt dưới khớp vai Robot thiết kế thêm khói trụ để cố định với thân Robot. Phần thân dưới sẽ sử dụng bạc đạn để khớp vai này có thể xoay tròn tru và ổn định.



Hình 3.8 Khớp đùi Robot

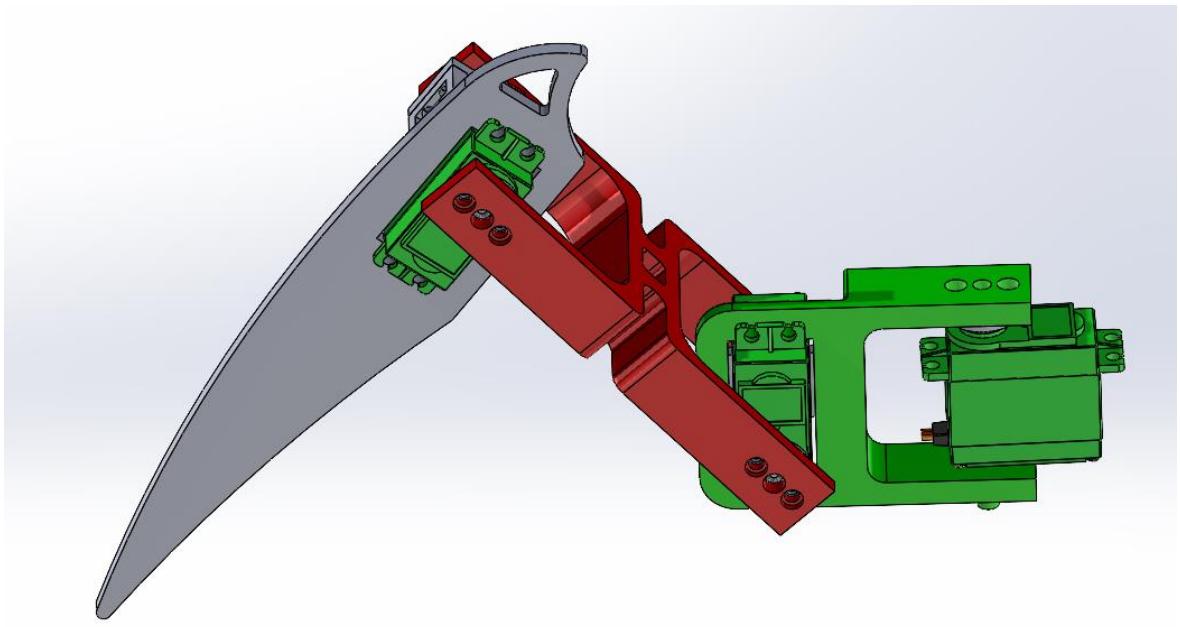
Khớp đùi được thiết kế đối xứng, bên trên khoét 3 lỗ để cố định động cơ servo, bên dưới sẽ sử dụng 2 bạc đạn để động cơ có thể xoay tròn tru và ổn định trong quá trình di chuyển.



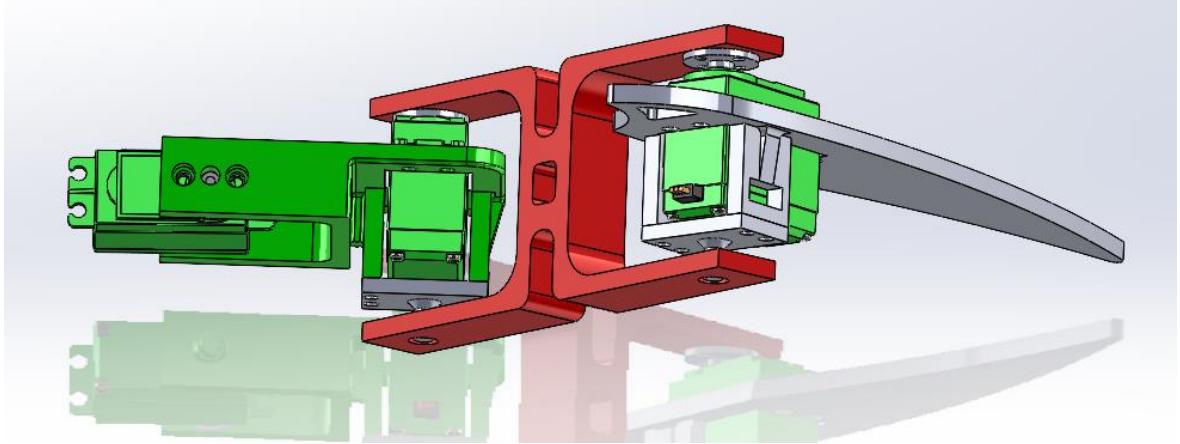
Hình 3.9 Khớp cẳng Robot

Bộ phận	Dài (mm)	Rộng (mm)	Cao (mm)
Vai	90	62	20
Đùi	140	65	20
Cẳng	214	40	29

Bảng 3.1 Kích thước các khớp của chân Robot

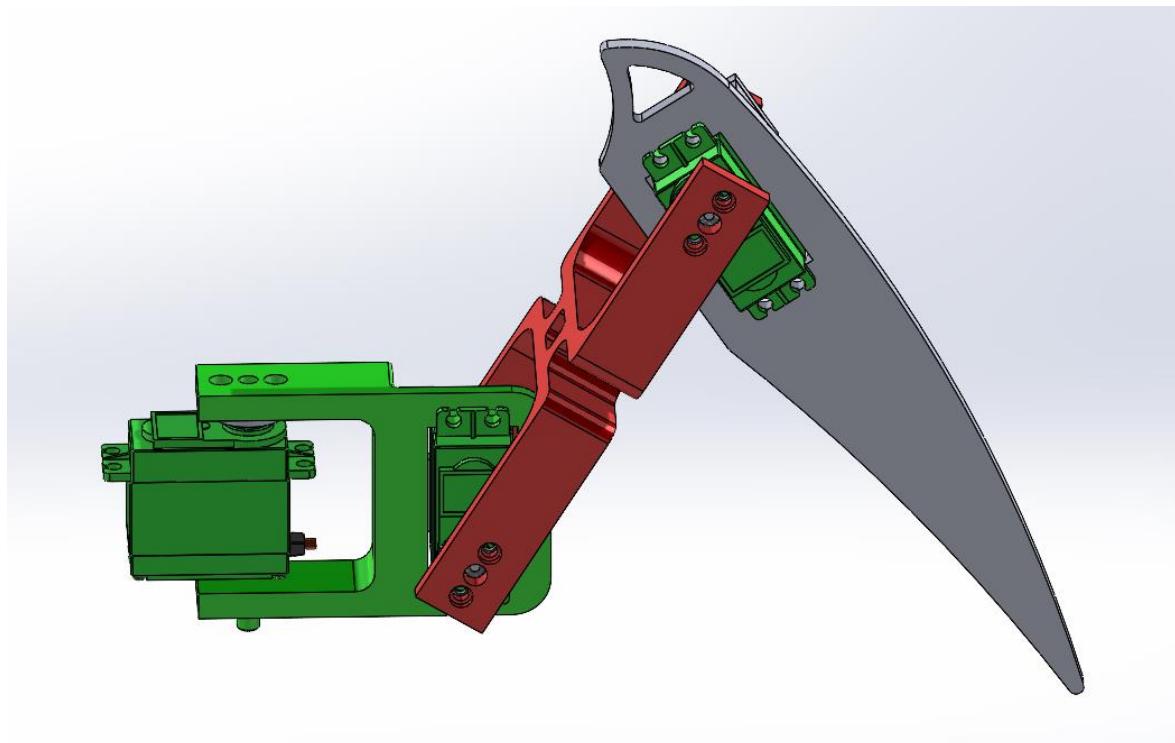


Hình 3.10 Chân phải Robot sau khi hoàn thiện

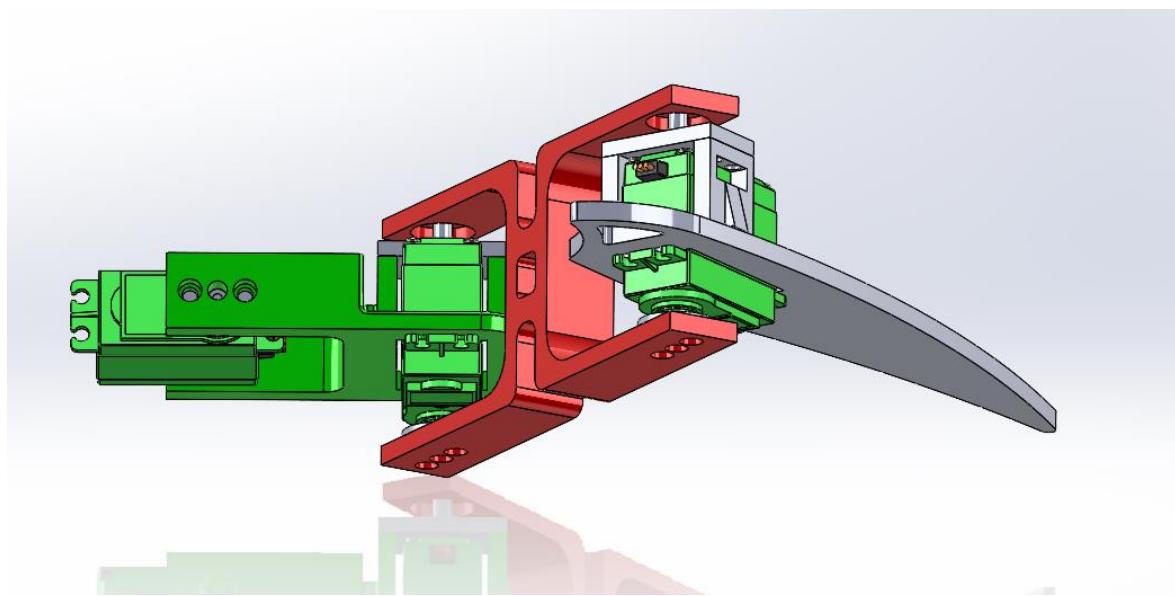


Hình 3.11 Chân phải Robot-mặt trên

Chân trái của Robot được thiết kế đối xứng với chân phải, giúp Robot cân đối trong việc di chuyển.



Hình 3.12 Chân trái Robot-mặt trước

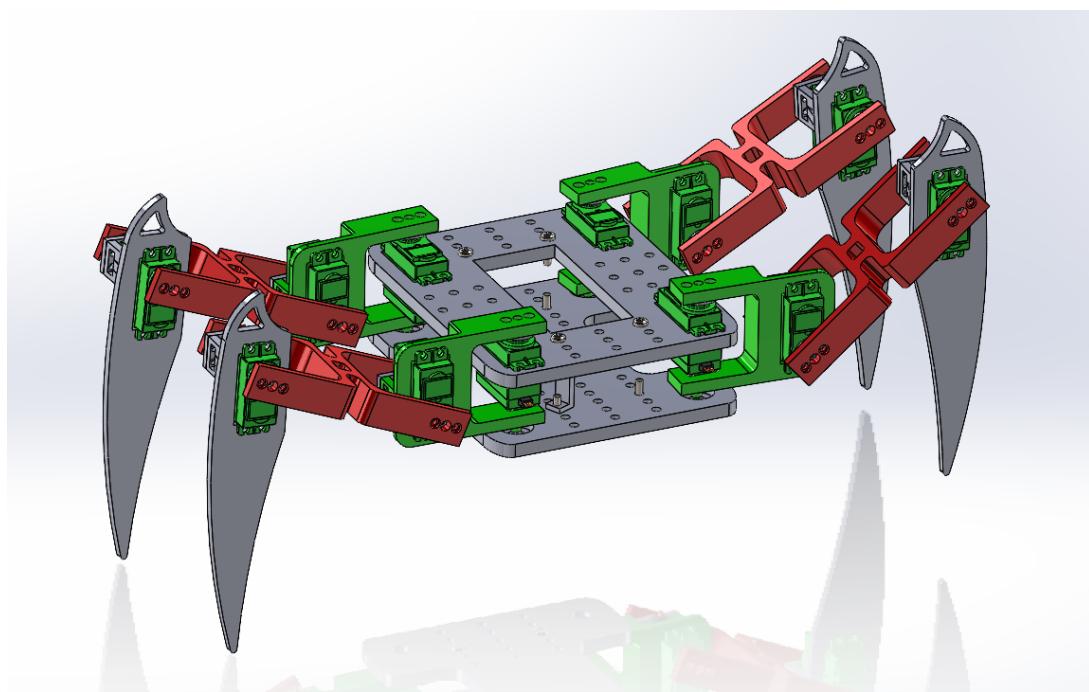


Hình 3.13 Chân trái Robot-mặt trên

#### 3.1.1.4. Hoàn thiện mô hình 3D robot

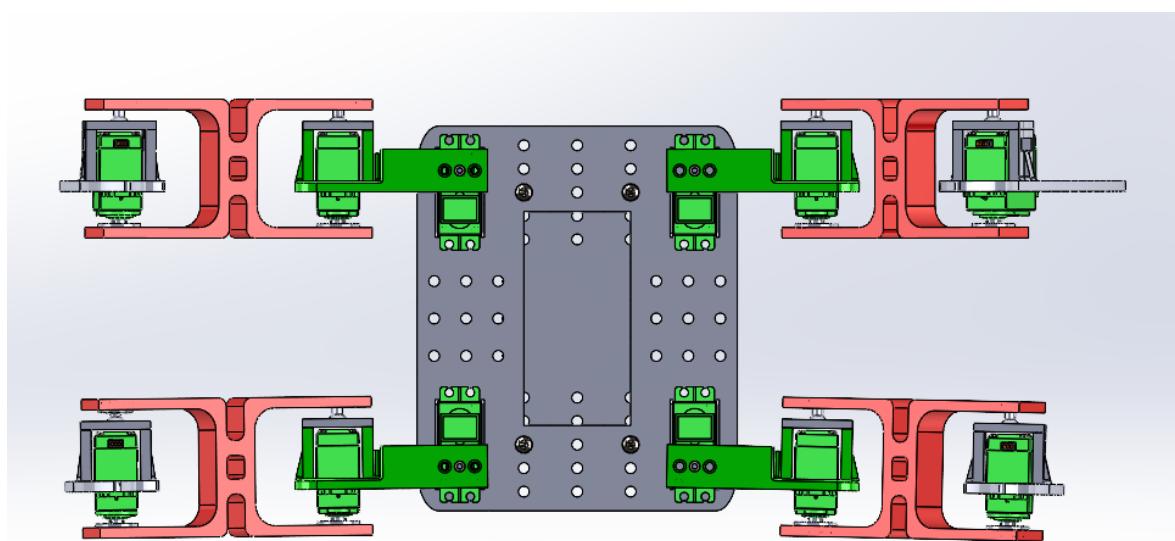
Sau khi thiết kế đủ tất cả các bộ phận của robot, sử dụng chức năng “Mate” của Solidworks để lắp ghép các bộ phận thành mô hình robot hoàn chỉnh. Sau khi có mô

hình hoàn chỉnh, ta thực hiện rà soát lại toàn bộ các chi tiết của robot như kích thước các chi tiết, dung sai lỗ, sự đồng trực giữa trực quay động cơ và lỗ khớp.

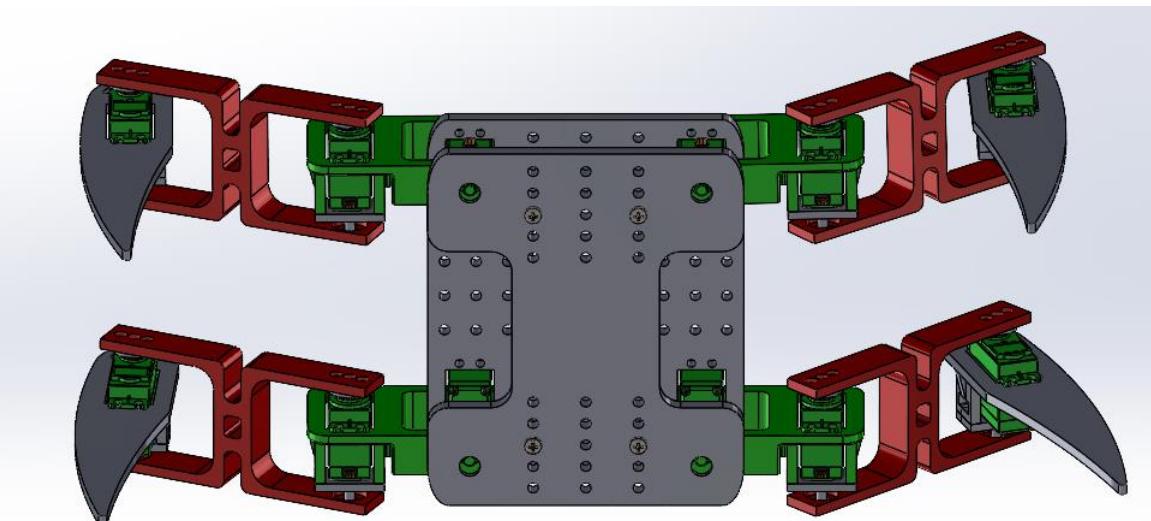


Hình 3.14 Mô hình robot hoàn chỉnh

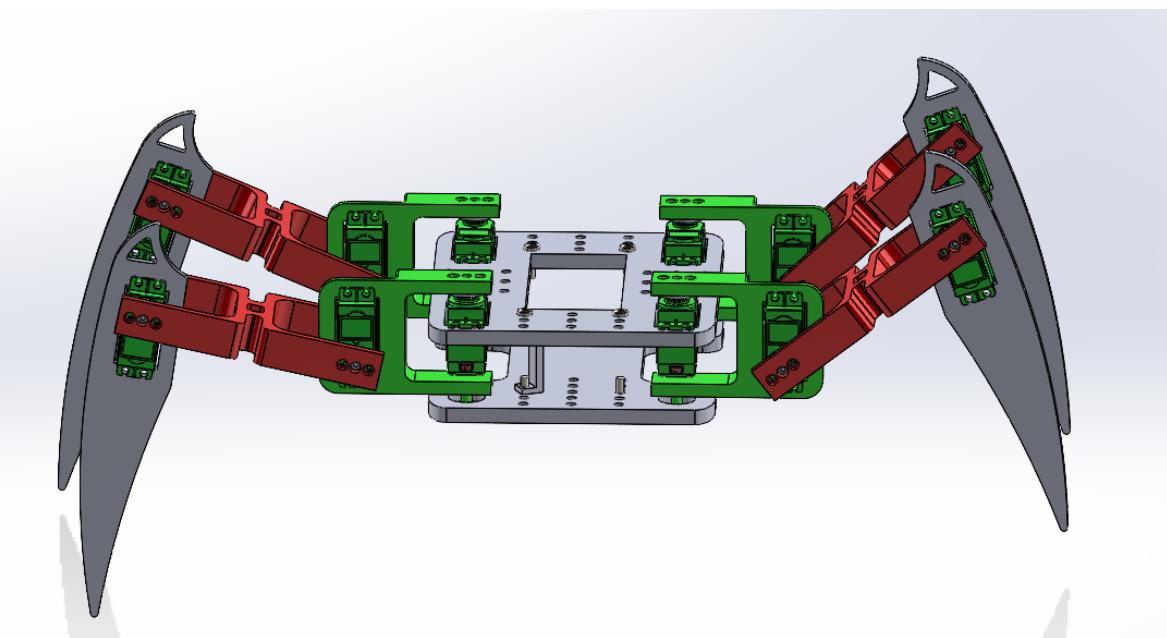
Sau khi có mô hình robot hoàn chỉnh, ta có thể dễ dàng mô phỏng chuyển động của các khớp quay, từ đó xác định được tầm hoạt động của các servo và sự phối hợp giữa các khớp khi robot di chuyển.



Hình 3.15 Mô hình robot hoàn chỉnh-mặt trên

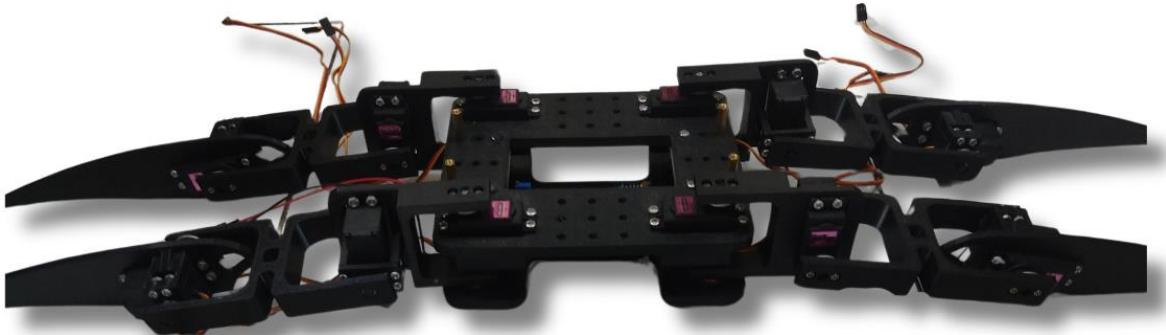


Hình 3.16 Mô hình robot hoàn chỉnh-mặt dưới



Hình 3.17 Mô hình robot hoàn chỉnh-mặt trước

### 3.1.2. Hoàn thiện mô hình robot thực tế



Hình 3.18 Mô hình robot sau khi gắn động cơ

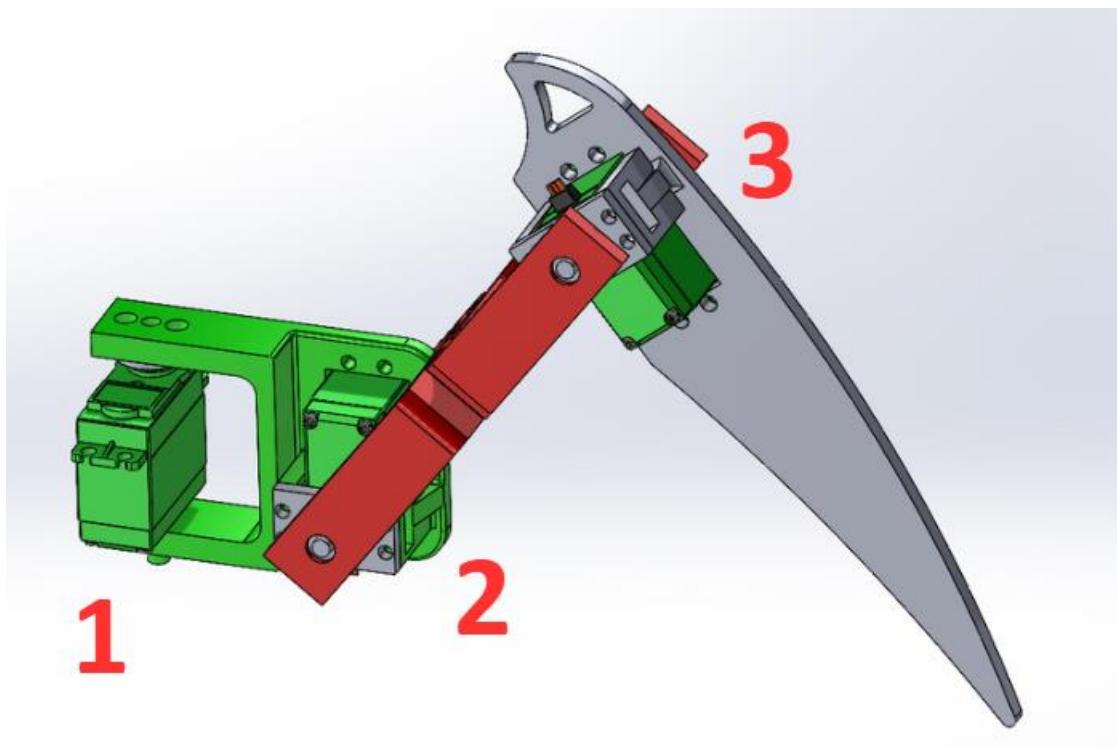
Trước khi cố định động cơ, nhóm đã calib và căn chỉnh động cơ quay sẵn ở một góc cố định, mục đích để giảm thiểu tối đa khả năng va chạm của robot khi điều khiển và để tối ưu tầm quay của động cơ servo. Do tầm hoạt động của servo chỉ trong khoảng 0 – 180 độ.

Ta có bảng hiệu chỉnh góc quay ban đầu của 12 động cơ:

Chân	Thứ tự động cơ	Góc ban đầu (độ)
Trái trước	1	45
	2	70
	3	138
Trái sau	1	138
	2	76
	3	132
Phải sau	1	134
	2	77
	3	132

Phải trước	1	43
	2	74
	3	138

Bảng 3.2 Bảng hiệu chỉnh góc quay ban đầu của 12 động cơ



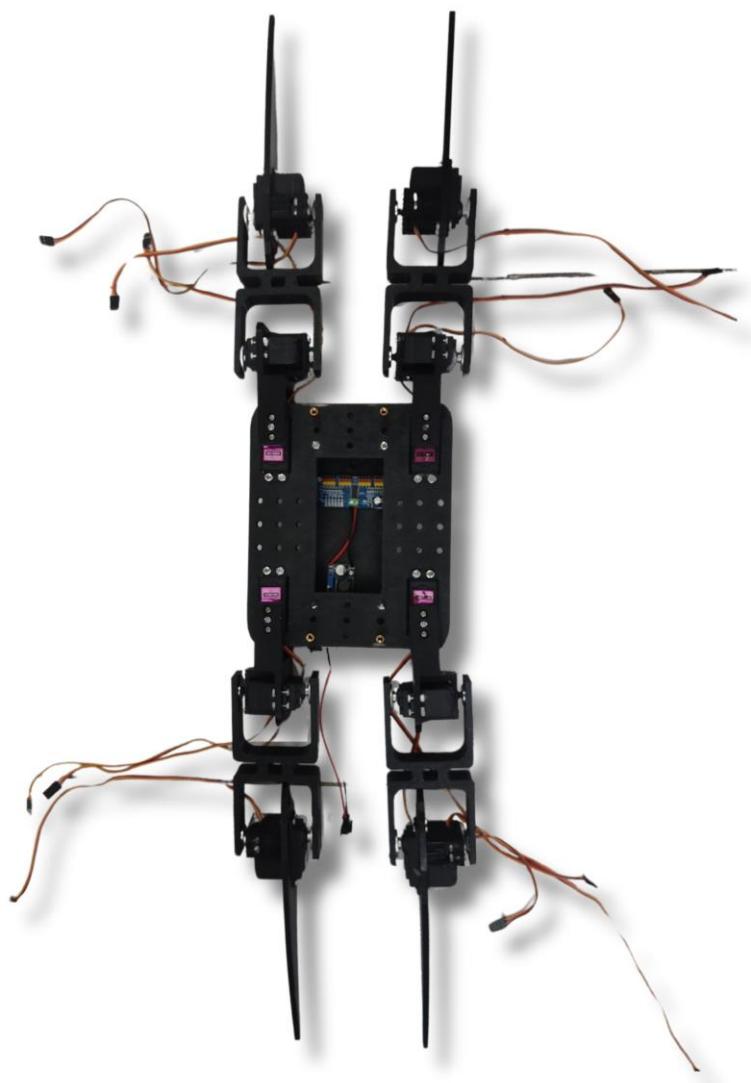
Hình 3.19 Thứ tự động cơ của 1 chân

Sau khi đã hiệu chỉnh góc quay ban đầu của 12 động cơ, ta xác định tầm quay giới hạn của từng động cơ để trong quá trình điều khiển, không xảy ra va chạm giữa các chi tiết. Do 4 chân của robot tương tự nhau nên ta chỉ cần xác định giới hạn của 1 chân và áp dụng cho 3 chân còn lại.

Thứ tự động cơ	Tầm quay giới hạn (độ)
1	25 – 155

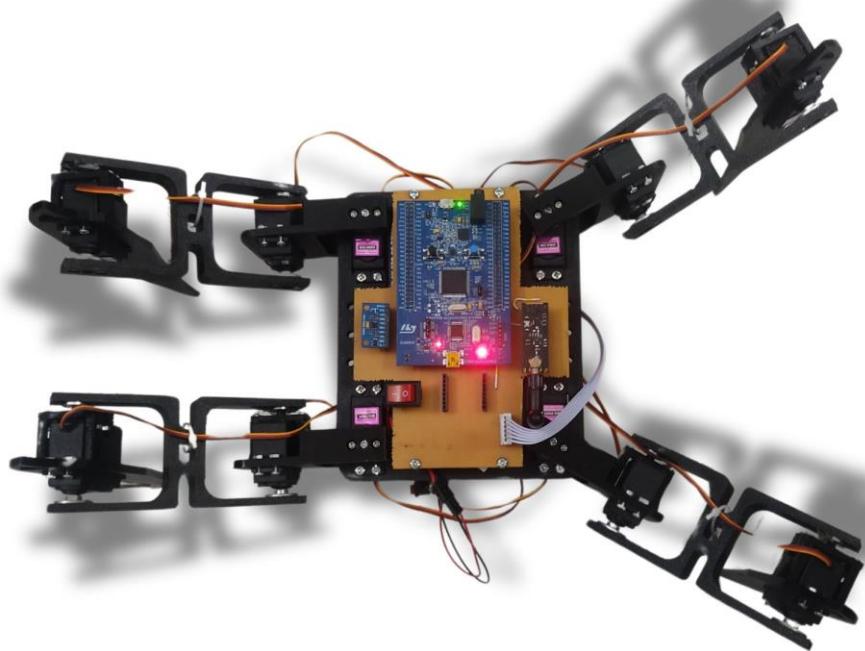
2	0 – 180
3	0 – 180

Bảng 3.3 Tầm quay giới hạn của động cơ



Hình 3.20 Mô hình robot với module PWM và module nguồn

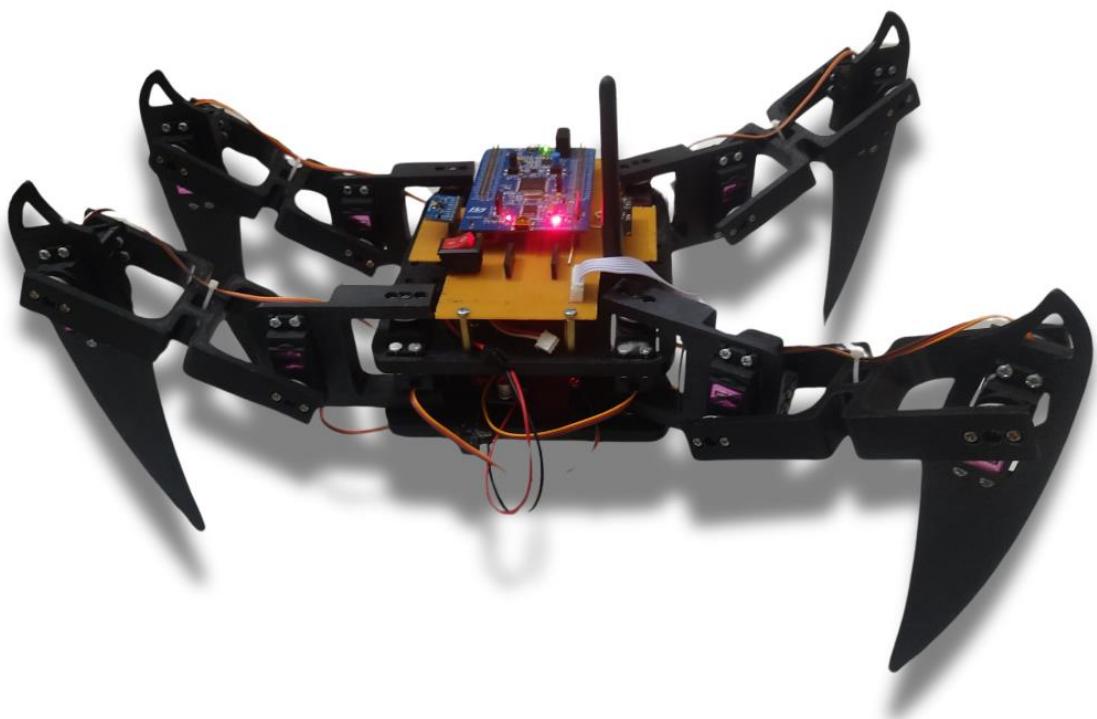
Module PWM và module nguồn được gắn cố định vào thân dưới của robot bằng ốc và bu lông giúp robot ổn định khi di chuyển.



Hình 3.21 Mô hình robot với mạch PCB

Mạch PCB được cố định 4 góc nhờ trụ đồng, tạo khoảng trống bên dưới mạch PCB và thân robot, các dây tín hiệu của servo cũng được cố định vào chân robot để không bị vướng khi robot thực hiện các động tác tiến, lùi.

Mạch PCB được thiết kế dạng hình chữ thập để tránh va chạm với khớp vai của robot khi robot di chuyển.



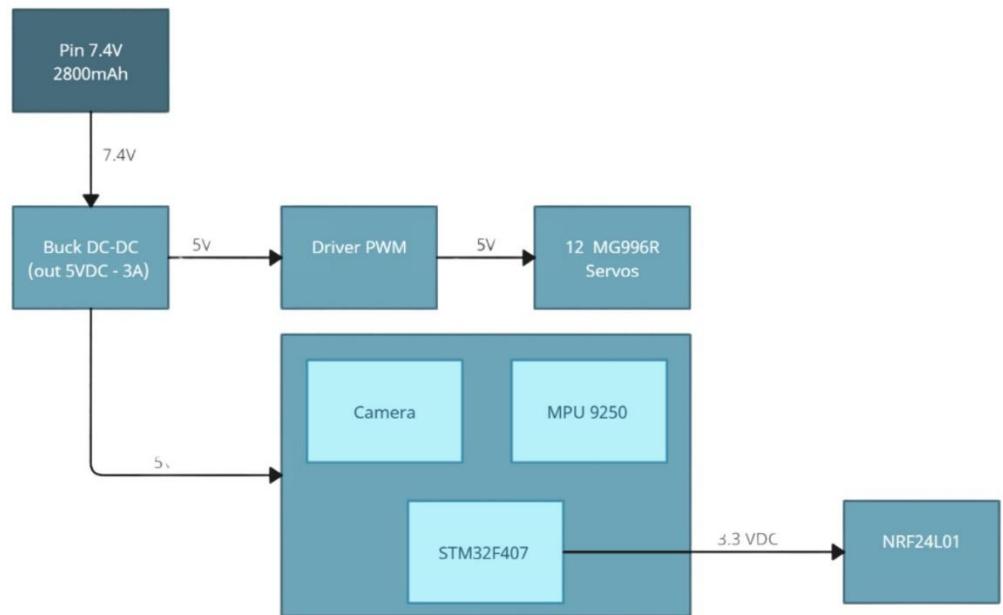
Hình 3.22 Dáng đứng của robot khi khởi động



Hình 3.23 Mô hình robot hoàn chỉnh

### 3.2. Sơ đồ kết nối

#### 3.2.1. Sơ đồ nguồn điện

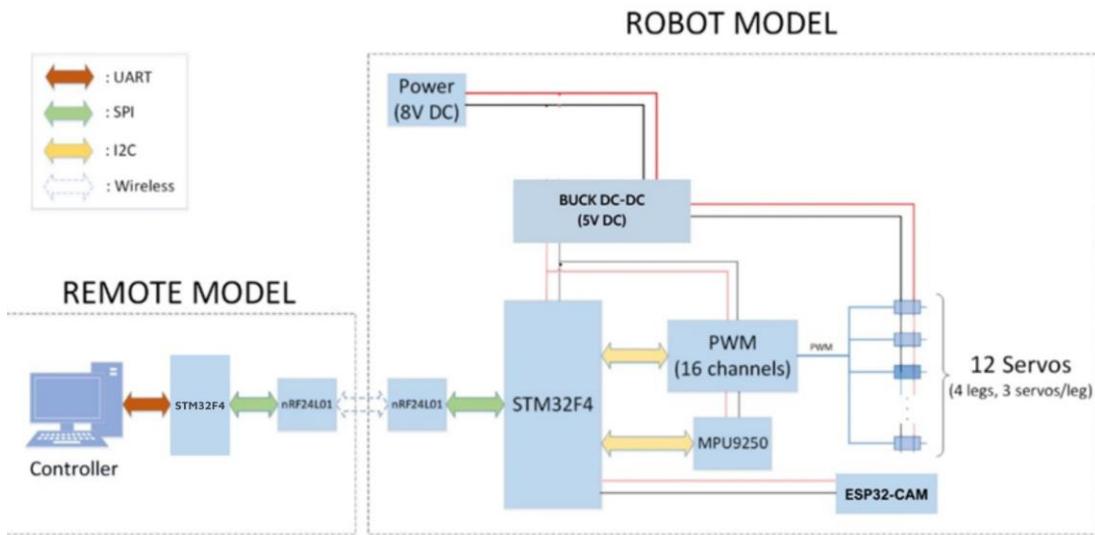


Hình 3.24 Sơ đồ mô tả nguồn điện qua các thiết bị

Toàn bộ nguồn điện cho Robot Nhện được cung cấp từ một pin LiPo 2800mAh 7.4V. Nguồn từ pin được điều chỉnh qua mạch giảm áp DC-DC buck LM2596 để giảm xuống điện áp ổn định là 5V. Điện áp 5V này được sử dụng để cung cấp nguồn cho động cơ thông qua mạch PCA9685. Mạch giảm áp cũng có khả năng chịu dòng cao, đáp ứng được cho nhu cầu tối đa để điều khiển tất cả 12 động cơ.

Sau khi đi qua module giảm áp DC-DC Buck, điện áp ổn định 5V được sử dụng để cung cấp nguồn cho Board và các ngoại vi khác như STM32F4, ESP32-CAM, IMU MPU9250. Ngõ ra 3V của STM32F4 được sử dụng làm nguồn hoạt động cho NRF24L01. Mặc dù việc cấp nguồn từ vi xử lí cho ngoại vi không được khuyến khích, nhưng với việc giảm bớt module và số lượng dây, điều này giúp giảm diện tích và cân nặng toàn bộ mạch điện. Do đó, việc cấp nguồn được cân nhắc và sử dụng.

### 3.2.2. Sơ đồ hệ thống



Hình 3.25 Sơ đồ kết nối hệ thống

Phần cứng bao gồm 2 nhóm chính:

- ROBOT MODEL: Mô hình robot nhện, vi điều khiển, động cơ, pin và các linh kiện điện tử cần thiết.
- REMOTE MODEL: máy tính (giao diện điều khiển), vi xử lý, module giao tiếp không dây.

REMOTE MODEL và ROBOT MODEL sẽ giao tiếp không dây với nhau thông qua module nRF24L01.

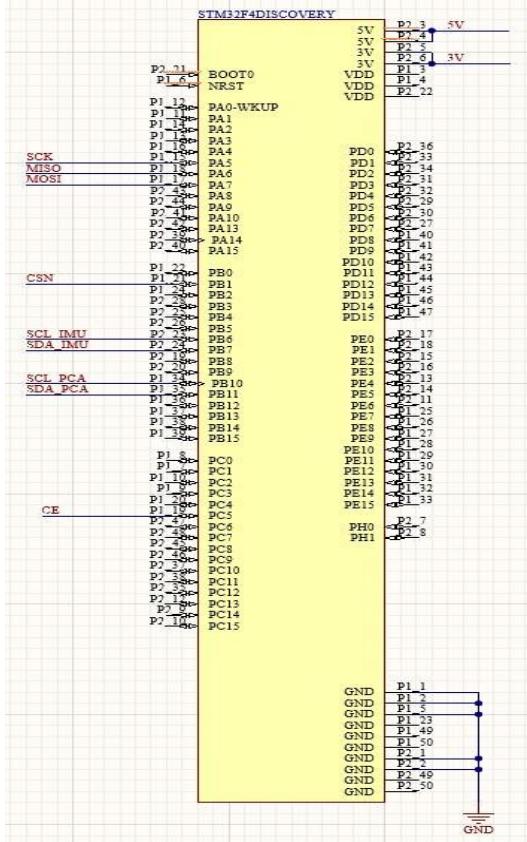
## 3.3. Thiết kế Schematic và PCB

Altium Designer - một phần mềm thiết kế PCB (Printed Circuit Board) mạnh mẽ và phổ biến được sử dụng để thiết kế mạch điện cho mô hình robot nhện 4 chân. Altium Designer cho phép người dùng thiết kế các mạch điện từ đơn giản đến phức tạp một cách hiệu quả và linh hoạt.

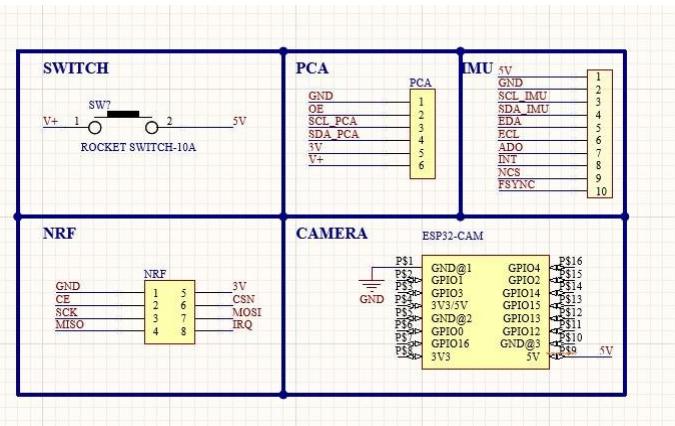
### 3.3.1. Thiết kế mạch Schematic

Để liên kết các khối chức năng và giảm bớt các dây liên kết làm cho robot trở nên gọn hơn và ổn định hơn ta cần thực hiện các mạch để kết nối các khối

chức năng. Schematic sẽ mô tả cách các thành phần trong mạch được kết nối với nhau và cách chúng tương tác để hoạt động như mong đợi. Điều này sẽ là cơ sở cho việc thiết kế mạch PCB sau này.



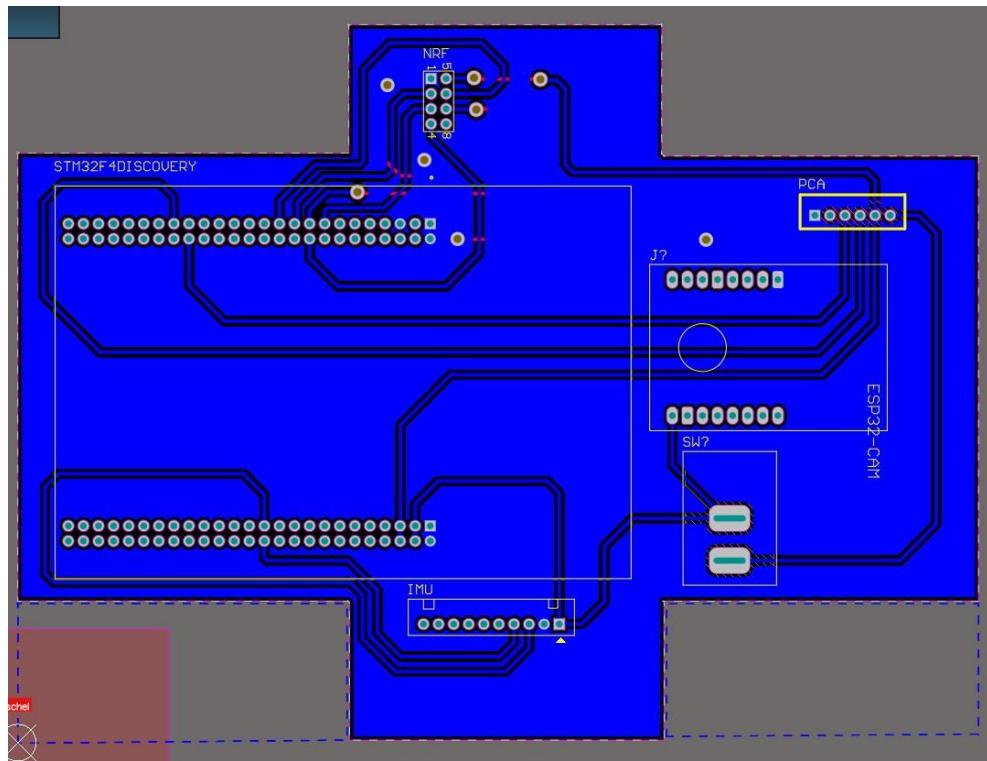
Hình 3.26 Khối Vi xử lí STM32F407



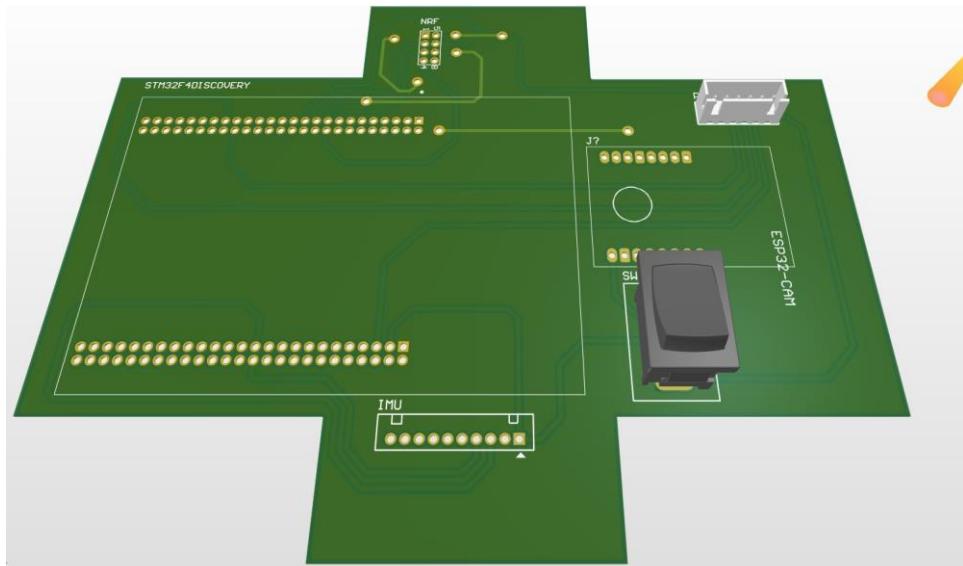
Hình 3.27 Các khối chức năng

### 3.3.2. Thiết kế mạch PCB

Sau khi hoàn thiện bản vẽ schematic, sẽ tiến hành thiết kế mạch in PCB (Printed Circuit Board). Mục tiêu là chuyển các kết nối và linh kiện từ bản vẽ schematic sang một bảng mạch in cụ thể. Điều này đảm bảo rằng các linh kiện được đặt đúng vị trí và kết nối mạch được tối ưu hóa.



Hình 3.28 2D hình vẽ PCB mạch điện



Hình 3.29 3D hình vẽ PCB mạch điện

### 3.3.3. Hoàn thiện mạch thực tế

Xuất bảng mạch in từ thiết kế PCB, lắp ráp các linh kiện lên bảng mạch, và kiểm tra mạch để đảm bảo rằng nó hoạt động đúng như mong đợi.



Hình 3.30 Mạch PCB hoàn thiện

### 3.4. Lựa chọn linh kiện

#### 3.4.1. Board điều khiển (STM32F407 Discovery)

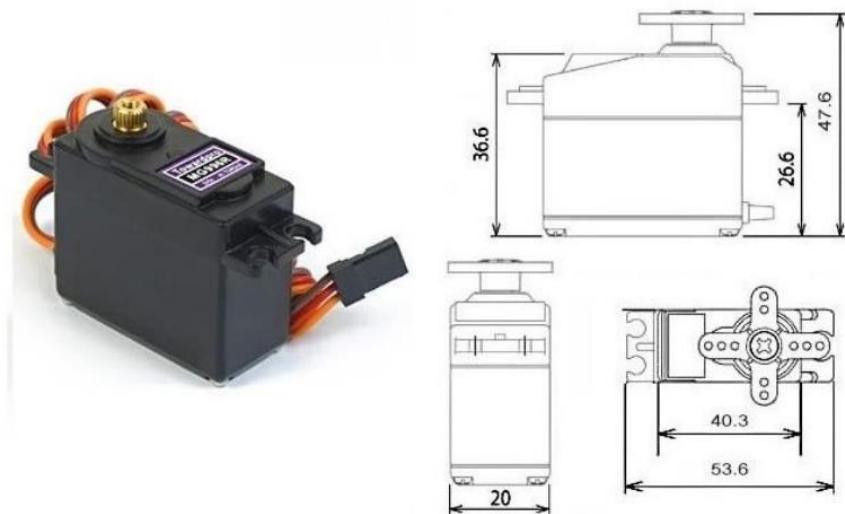


Hình 3.31 Kit STM32F407 Discovery

<b>STM32F407 Discovery</b>	<b>Đặc điểm</b>
Tốc độ Clock	Lên đến 168MHz
Nguồn cung cấp	5V USB, nguồn ngoài từ 2.0V- 3.6V
Flash Memory	1MB
SRAM	192KB
UART	4 USARTs, 4 UARTs
I2C	3 I2C interfaces
SPI	3 SPI interfaces
I/O Pins	100 chân GPIO
Timers	17 timers
DMA	Hỗ trợ 2 DMA, 7 kênh mỗi DMA

Bảng 3.4 Thông số kỹ thuật cơ bản STM32F4 đáp ứng yêu cầu robot

### 3.4.2. Động cơ (MG996R)

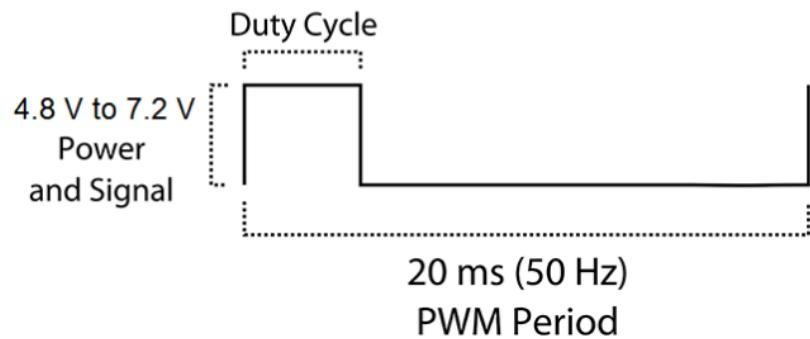


Hình 3.32 Động cơ MG996R

<b>MG996R</b>	<b>Đặc điểm</b>
Kích thước	40.7 x 19.7 x 42.9 mm
Moment xoắn	9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
Tốc độ vận hành	0.17 giây/60° (4.8 V), 0.14 giây/60° (6 V)
Điện áp vận hành	Từ 4.8 V đến 7.2 V
Dòng vận hành	500mA-900mA(6V)
Dòng xoắn dừng (khi động cơ bị kẹt)	2.5A (6V)
Phạm vi quay	180°
Trọng lượng	55g

Bảng 3.5 Thông số kỹ thuật MG996R

Phương pháp điều khiển PWM:



Hình 3.33 Chu kì xung PWM của MG996R

- Độ rộng xung 0.5ms ~ 2.5ms tương ứng 0-180 độ
- Tần số 50Hz, chu kỳ 20ms

### 3.4.3. Nguồn điện (Pin 7.4V 2800mAh)



Hình 3.34 Pin 7.4V 2800mAh

Tổng dòng điện tiêu thụ của 12 động cơ sẽ là:

$$12 \text{ động cơ} \times 700 \text{ mA} = 8400 \text{ mA.}$$

Thời lượng sử dụng của pin:

Dung lượng pin: 2800 mAh Thời lượng sử dụng (giờ) = Dung lượng pin (mAh) / Tổng dòng điện tiêu thụ (mA)

$$\text{Thời lượng sử dụng (giờ)} = 2800 \text{ mAh} / 8400 \text{ mA} = 0.333 \text{ giờ}$$

Vậy thời lượng sử dụng dự kiến sẽ là khoảng 0.333 giờ hoặc 20 phút khi kéo 12 động cơ MG996R ở chế độ làm việc bình thường.

### 3.4.4. Module chuyển đổi điện áp (Buck DC-DC LM2596)



Hình 3.35 Mạch giảm áp DC-DC Buck LM2596 3A

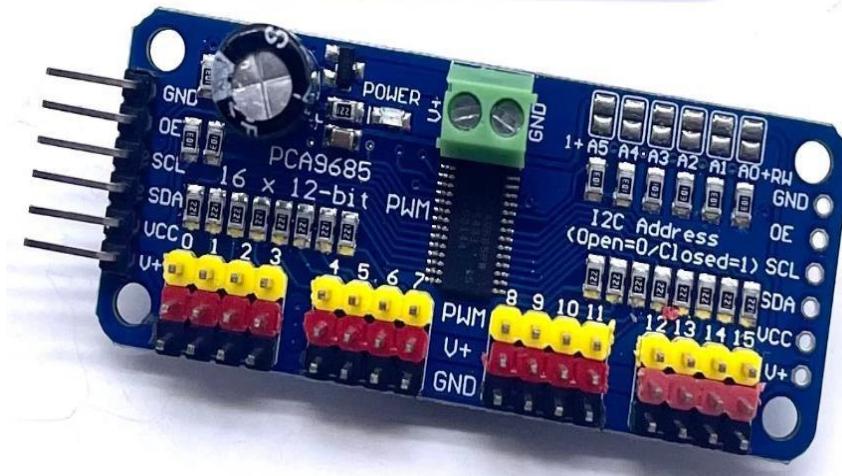
Mạch giảm áp DC-DC Buck LM2596 3A có kích thước nhỏ gọn có khả năng giảm áp từ 30VDC xuống 1.5VDC mà vẫn đạt hiệu suất cao (92%), thích hợp cho các ứng dụng chia nguồn, hạ áp, cấp cho các thiết bị như camera, robot,...

Đối với mạch của robot nhện. Mạch giảm áp DC-DC Buck đóng vai trò là module giảm áp từ nguồn pin 7.4V xuống 5V-3A để cấp cho động cơ, vi xử lý và các thiết bị ngoại vi khác.

Buck DC-DC	Đặc điểm
Điện áp đầu vào	3V-30V
Điện áp đầu ra	Điều chỉnh được: 1.5V – 30V
Dòng đáp ứng tối đa	3A
Hiệu suất	92%
Công suất	15W
Kích thước	45x20x14mm

Bảng 3.6 Thông số kỹ thuật Mạch giảm áp DC-DC Buck

#### 3.4.5. Mạch điều khiển PWM (PCA9685)



Hình 3.36 Mạch điều khiển 16 Chanel PWM PCA9685

Mạch điều khiển 16 Chanel PWM PCA9685 được sử dụng để có thể xuất ra đồng thời 16 xung PWM từ 16 cổng khác nhau thông qua giao tiếp I2C sử

dụng IC PCA9685, có thể điều khiển đồng thời 16 RC Servo hoặc Dimmer 16 thiết bị đồng thời,..

Mạch điều khiển 16 Chanel PWM PCA9685 có chất lượng phần cứng, gia công tốt, độ bền cao, phù hợp cho các nhu cầu cần điều khiển nhiều xung PWM như Robot cánh tay máy, Robot nhện,..

PCA9685	Đặc điểm
IC chính	PCA9685
Điện áp sử dụng	2.3VDC – 5.5VDC
Số kênh PWM	16 kênh, tần số: 40-1000Hz
Độ phân giải PWM	12bit
Giao tiếp I2C	TTL 3-5VDC
Kích thước	62.5mmx25.4mmx3mm

Bảng 3.7 Thông số kỹ thuật PCA9685

### 3.4.6. Cảm biến 9DOF IMU MPU9250



Hình 3.37 9DOF IMU MPU9250

IMU (Inertial Measurement Unit) là một hệ thống tích hợp cảm biến gia tốc và cảm biến góc quay để đo và theo dõi chuyển động của một đối tượng trong không gian ba chiều.

IMU MPU9250 bao gồm ba cảm biến chính sau:

Cảm biến gia tốc (Accelerometer): Cảm biến này đo lường gia tốc tuyến tính của một đối tượng và cung cấp thông tin về sự thay đổi vận tốc của đối tượng theo các trục x, y và z. Nó có thể đo cả gia tốc tĩnh và gia tốc động.

Cảm biến góc quay (Gyroscope): Cảm biến này đo lường tốc độ góc của một đối tượng xung quanh các trục x, y và z. Nó cung cấp thông tin về tốc độ quay của đối tượng.

Cảm biến từ (Magnetometer): Cảm biến này đo lường trường từ xung quanh một đối tượng và cho phép xác định hướng của đối tượng trong không gian.

<b>9DoF IMU MPU 9250</b>	<b>Đặc điểm</b>
Accelerometer	Measurement range: $\pm 2$ g, $\pm 4$ g, $\pm 8$ g, $\pm 16$ g Resolution: 16-bit
Gyroscope	$\pm 250$ , $\pm 500$ , $\pm 1000$ , and $\pm 2000^{\circ}/sec$ (dps) Resolution: 16-bit
Magnetometer	Measurement range: $\pm 4800\mu T$ Resolution: 16-bit
Chuẩn giao tiếp	I2C (Default address: 0x28) UART
Điện áp sử dụng	3~5VDC
Kích thước	15x25mm

Bảng 3.8 Thông số kỹ thuật IMU MPU9250

### 3.4.7. Module giao tiếp không dây (NRF24L01)



Hình 3.38 Module thu phát RF NRF24L01 2.4Ghz

Module RF 2.4GHz NRF24L01+ sử dụng chip truyền sóng NRF24L01+ mới nhất từ hãng Nordic với nhiều cải tiến so với chip NRF24L01 cũ về tốc độ truyền, khoảng cách, độ nhạy, bổ sung thêm pipelines, buffers, và tính năng auto-retransmit nhưng vẫn tương thích ngược với phiên bản cũ về cách sử dụng ..., NRF24L01+ hoạt động trên dải tần 2.4GHz và sử dụng giao tiếp SPI, khoảng cách tối đa trong điều khiển không vật cản lên đến 100m.

NRF24L01	Đặc điểm
Điện thế hoạt động	1.9 ~ 3.6VDC
Tần số thu phát	2.4GHz
Phạm vi truyền dẫn	100m trong môi trường mở baud 250kbps
Tốc độ truyền dữ liệu	250kbps đến 2Mbps
Tính năng đặc biệt	Tự động bắt tay ( Auto Acknowledge), tự động truyền lại khi bị lỗi(Auto Re-Transmit), Multiceiver -6 Data pipes

Bộ đệm dữ liệu	32 Byte riêng biệt cho TX và RX
Kênh truyền song	Chọn được 125 kênh
Kích thước	15x29mm

Bảng 3.9 Thông số kỹ thuật NRF24L01

### 3.4.8. Camera ESP32-CAM



Hình 3.39 Kit phát triển Wifi ESP32-CAM

Việc sử dụng camera trong mô hình robot nhện 4 chân có thể mang lại một số lợi ích quan trọng:

Dò tìm môi trường: Camera có thể được sử dụng để quét và phân tích môi trường xung quanh robot nhện, giúp nó phát hiện và tránh các vật cản trong quá trình di chuyển.

Hệ thống phát hiện vị trí: Bằng cách sử dụng xử lý hình ảnh, robot nhện có thể xác định vị trí của nó trong môi trường. Điều này có thể hữu ích để xác định vị trí mục tiêu hoặc để theo dõi vị trí của robot trong thời gian thực.

Giao tiếp với người điều khiển: Camera cũng có thể được sử dụng để truyền hình ảnh hoặc dữ liệu video trực tiếp về lại người điều khiển, giúp họ quan sát môi trường và điều khiển robot một cách chính xác hơn.

<b>Ai-Thinker ESP32-S</b>	<b>Đặc điểm</b>
Nguồn cấp	5VDC(cần nguồn 2A trở lên)
Điện áp giao tiếp GPIO	3.3VDC
UART Baudrate	Mặc định 115200 bps
Phạm vi tần số	2412~2484MHz
Kích thước	40.5 x 27 x 4.5 mm

Bảng 3.10 Thông số kỹ thuật ESP32-S

<b>Camera OV2640</b>	<b>Đặc điểm</b>
Độ phân giải	2 Megapixel (1600 x 1200 pixels)
Tỷ lệ khung hình	15fps (1600x1200), 30fps (UXGA), 60fps (VGA)
Độ nhạy sáng tối thiểu	0.5Lux
Điện áp hoạt động	3.3VDC
Kích thước	6 x 6mm

Bảng 3.11 Thông số kỹ thuật Camera OV2640

## **Chương 4. THIẾT KẾ GIAO DIỆN VÀ GIẢI THUẬT ĐIỀU KHIỂN**

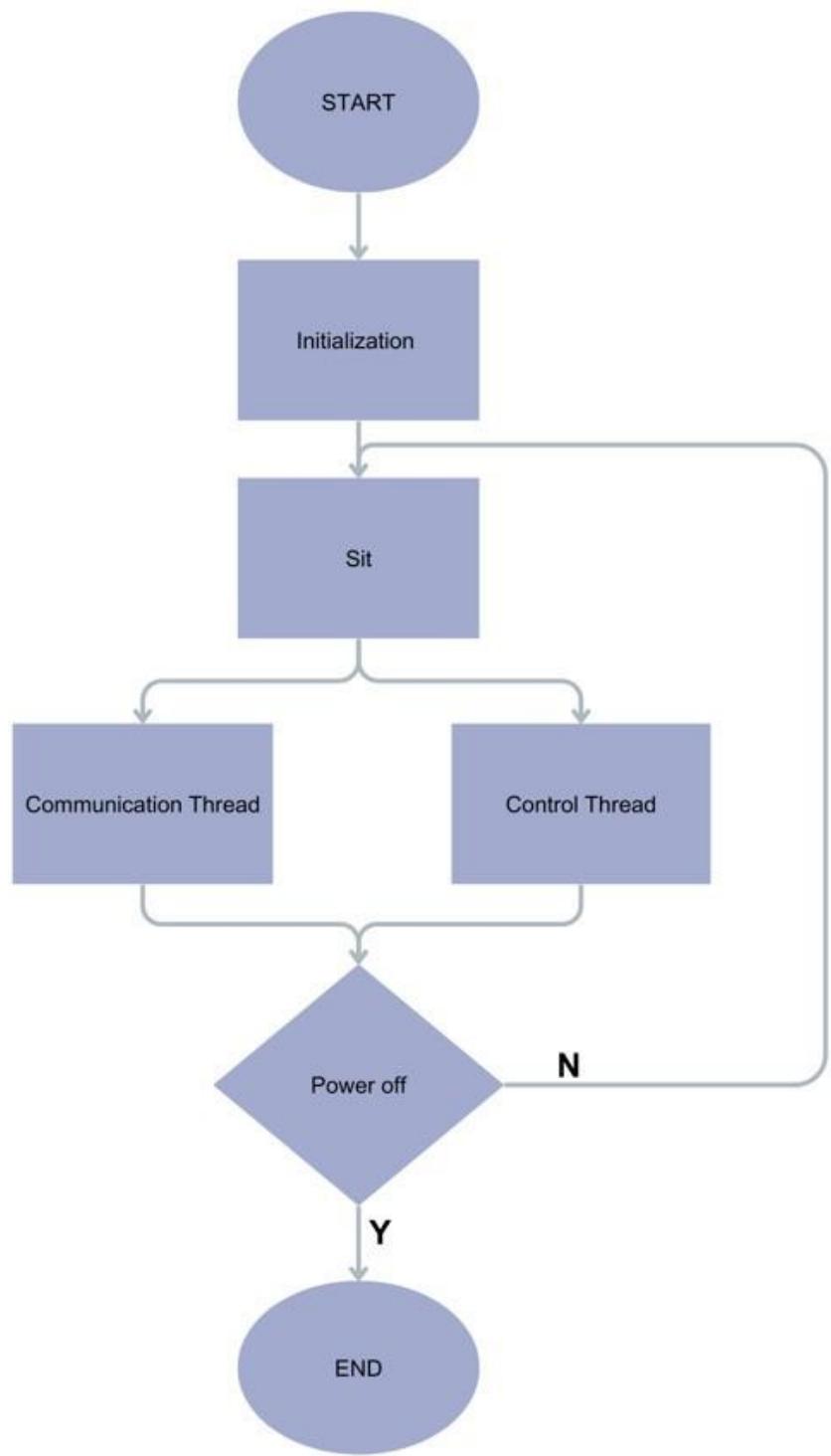
Sau khi đã hoàn thành phần cứng của robot, ta bắt đầu thiết kế giao diện và giải thuật điều khiển cho robot từ xa. Trong đồ án này, phải lập trình cho 3 phần: lập trình vi điều khiển để điều khiển robot, lập trình vi điều khiển cho module điều khiển từ xa và lập trình giao diện điều khiển trên máy tính. Bởi vì module điều khiển từ xa giao tiếp với máy tính thông qua UART và vai trò chính của module này là truyền nhận dữ liệu không dây giữa máy tính với robot nên ta có thể xem khối giao diện và module điều khiển là một, coi như dữ liệu được truyền nhận trực tiếp từ robot đến giao diện điều khiển.

### **4.1. Thiết kế giải thuật cho vi điều khiển trên robot**

Khi điều khiển robot từ xa, dữ liệu robot cần nhận được là lệnh điều khiển và dữ liệu robot gửi đi sẽ là giá trị của góc roll, pitch, yaw đo được từ cảm biến. Bởi vì việc đọc dữ liệu cảm biến và điều khiển cho robot di chuyển phải được thực hiện liên tục và đồng thời trong khi vi điều khiển nhóm sử dụng chỉ có 1 nhân xử lý. Để giải quyết, nhóm quyết định sử dụng hệ điều hành thời gian thực (Real Time Operating System – RTOS) được cung cấp từ hãng ST để có thể chạy 2 tác vụ đồng thời nhờ việc phân chia khe thời gian được đảm nhiệm bởi. Việc này giúp ta có thể gửi dữ liệu cảm biến lên giao diện trong khi vẫn điều khiển cho robot di chuyển được. Hai luồng được khởi tạo và chạy song song là:

- Control Thread: luồng điều khiển hoạt động cho robot.
- Communicate Thread: luồng truyền nhận không dây giữa vi điều khiển trên robot với máy tính.

Hai luồng này sẽ chạy song song và liên tục cho đến khi ngắt nguồn. Điều này giúp cho việc đọc dữ liệu cảm biến được liên tục và chính xác.

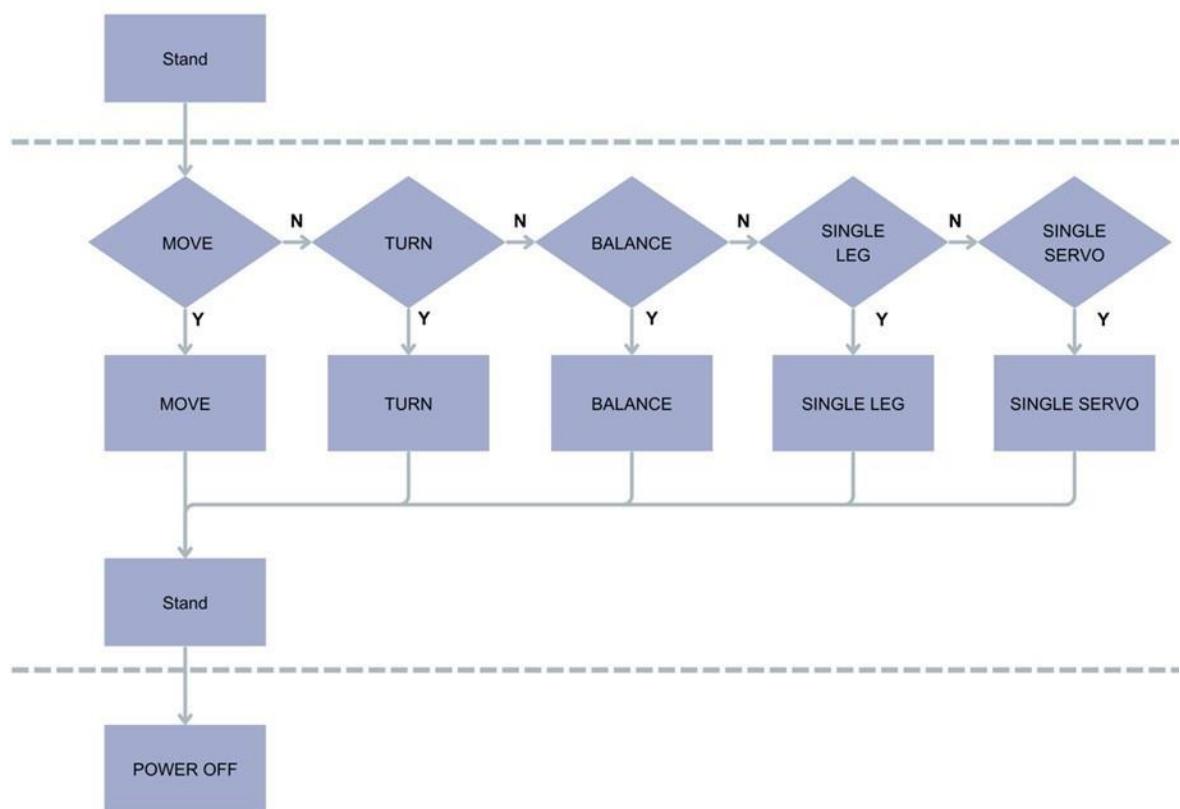


Hình 4.1 Lưu đồ giải thuật tổng quát của robot

Khi bật nguồn, robot cần khoảng 30 giây để khởi tạo và calib cho cảm biến  
gia tốc, giúp giá trị đọc được ít nhiễu và chính xác. Sau đó robot sẽ chuyển

sang dáng ngồi và sẵn sàng nhận lệnh điều khiển. Lúc này giá trị cảm biến được gửi liên tục lên máy tính với chu kỳ 0.1 giây. Sau khi điều khiển cho robot chuyển sang dáng đứng, các chức năng như di chuyển tiến lùi, xoay trái xoay phải và cân bằng thân sẽ được thực hiện khi nhận lệnh từ máy tính. Nếu robot vẫn đang ở trạng thái ngồi, các chức năng trên sẽ bị vô hiệu trên giao diện để đảm bảo an toàn trong việc điều khiển robot.

#### 4.1.1. Luồng điều khiển hoạt động của robot



Hình 4.2 Luồng điều khiển hoạt động của robot

Bắt đầu luồng điều khiển, chương trình sẽ xét mode điều khiển tương ứng với từng hành động của robot. Nếu không có lệnh điều khiển từ chương trình,

robot sẽ duy trì trạng thái đứng và chờ lệnh. Có 5 mode điều khiển tương ứng cho từng chức năng của robot:

- MOVE: điều khiển robot bước tiến hoặc bước lùi theo khoảng cách được nhập trên giao diện.
- TURN: điều khiển robot thực hiện xoay trái hoặc xoay phải
- BALANCE: điều khiển thân robot cân bằng hoặc theo giá trị đặt góc roll và pitch. Sử dụng bộ điều khiển PID.
- SINGLE LEG: điều khiển từng chân của robot với tọa độ end effector được nhập từ giao diện. Nếu tọa độ này vượt quá giới hạn quay của động cơ, lệnh sẽ không được thực hiện và giao diện thông báo cho người dùng cần thay đổi tọa độ khác.
- SINGLE SERVO: điều khiển từng động cơ servo đến góc được nhập trên giao diện, góc này là góc so với góc tọa độ  $(x_0, y_0, z_0)$  của từng chân robot.

Khi xác định được mode điều khiển, chương trình sẽ thực hiện điều khiển robot thực hiện hành động tương ứng. Sau khi thực hiện xong, chương trình sẽ xóa mode điều khiển quay lại chờ lệnh điều khiển tiếp theo, robot quay lại trạng thái đứng.

#### 4.1.1.1. Mode điều khiển tiến lùi

Chương trình sẽ gửi mode lệnh và khoảng cách di chuyển xuống cho robot (đơn vị mm). Robot sẽ tính toán số bước di chuyển và khoảng cách mỗi bước để di chuyển được đúng khoảng cách mong muốn. Robot thực hiện di chuyển theo dáng đi 2 pha không liên tục như đã phân tích bên trên, mỗi pha robot sẽ tiến lên theo khoảng cách là  $y$ . Sau 1 chu kỳ (2 pha) robot sẽ tiến lên 1 khoảng có giá trị  $2y$ . Giá trị  $y$  lý tưởng là 76 mm (là khoảng cách cân đối so với kích thước robot). Giải thuật bước tới của robot như sau:

- Bước 1: Xét tham số Distance bằng 0 hay khác 0. Nếu bằng 0, lệnh sẽ không được thực hiện và chương trình sẽ tiếp tục quay lại chờ đến khi nhận được lệnh mới. Nếu khác 0, tiếp tục thực thi chương trình.

- Bước 2: Chương trình sẽ tính số chu kỳ bước tới của robot dựa vào tham số Distance được nhập trên giao diện điều khiển.

$$\text{Số chu kỳ} = \text{Distance}/152$$

Nếu phần dư lớn hơn 76, số chu kỳ bước sẽ tăng thêm một đơn vị.

$$\text{Khoảng cách mỗi bước} = \text{Distance}/(\text{Số chu kỳ})$$

Sau đó chương trình sẽ điều khiển robot bước tới với số chu kỳ và khoảng cách mỗi bước tiến của robot đã tính được bên trên.

Vậy nên nếu khoảng cách nhập trên giao diện quá nhỏ (nhỏ hơn 76mm) robot cũng sẽ không di chuyển. Khoảng cách di chuyển lý tưởng của robot là bội của 152 mm.

- Bước 3: Từ tọa độ cuối của 4 chân. Tính toán góc quay cho từng động cơ servo bằng công thức động học ngược đã được nêu bên trên.
- Bước 4: Điều khiển động cơ của robot di chuyển đến góc chỉ định. Bước 3 và 4 sẽ được lặp lại đến khi xong một chu kỳ bước của robot. Nếu có lệnh dừng khẩn cấp thì sẽ dừng điều khiển động cơ ngay lập tức và xóa mode.
- Bước 5: Xét cờ STOP, nếu cờ được bật thì xóa mode và đợi đến khi robot hoàn thành xong chu kỳ bước.

Đối với lệnh Bước lùi, giải thuật tương tự với lệnh bước tiến.

#### **4.1.1.2. Mode xoay trái xoay phải**

Chương trình sẽ gửi mode lệnh và số lần xoay xuống cho robot. Robot sẽ thực hiện xoay đúng theo số lần được nhập từ giao diện. Trong dáng xoay của robot, việc quan trọng nhất là phải giữ cho trọng tâm của robot ổn định để robot không bị ngã. Robot thực hiện dáng xoay dạng 2 pha không liên tục. Mỗi pha robot sẽ quay một góc khoảng 16 độ. Sau khi thực

hiện xong 2 pha robot sẽ quay về dáng đứng ban đầu. Giải thuật xoay của robot như sau:

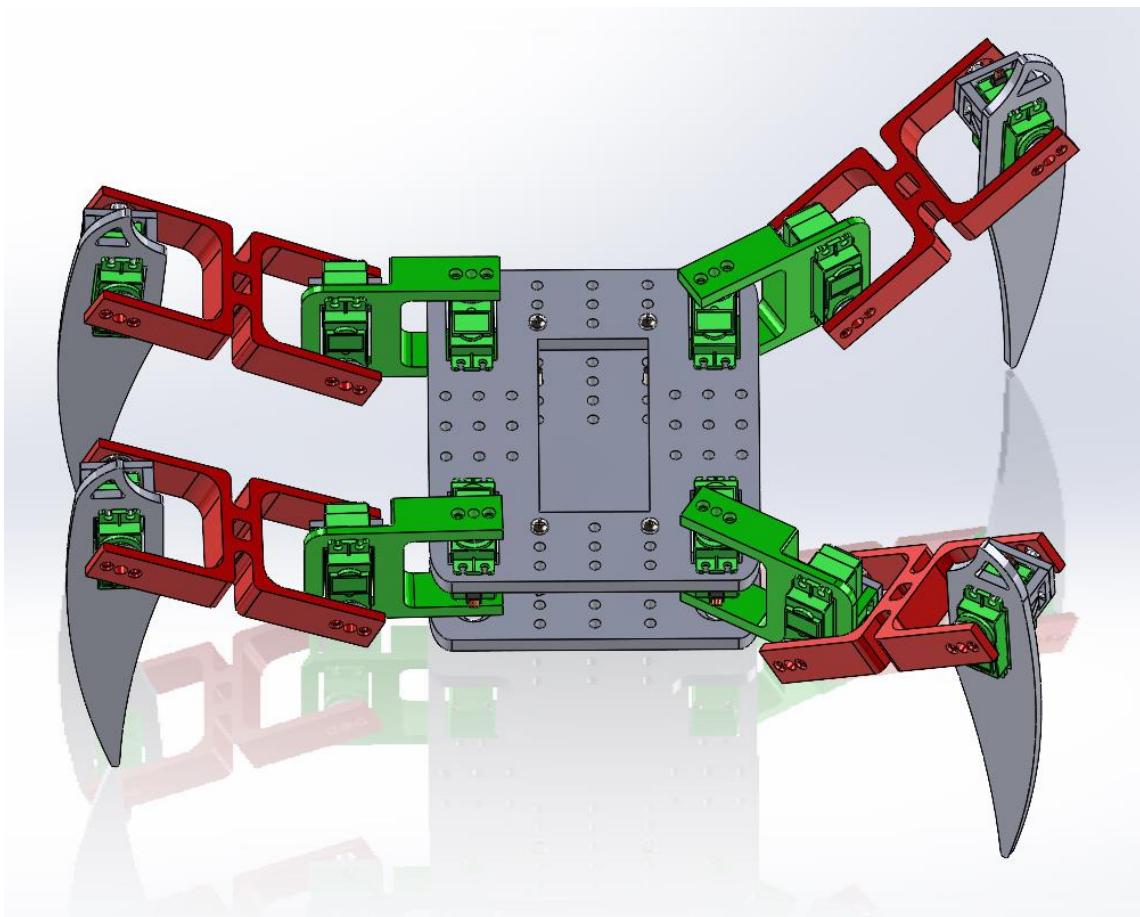
- Bước 1: Xét tham số Steps bằng 0 hay khác 0. Nếu bằng 0, lệnh sẽ không được thực hiện và chương trình sẽ tiếp tục quay lại chờ đến khi nhận được lệnh mới. Nếu khác 0, tiếp tục thực thi chương trình.
- Bước 2: Từ tọa độ cuối của 4 chân. Tính toán góc quay cho từng động cơ servo bằng công thức động học ngược đã được nêu bên trên.
- Bước 3: Điều khiển động cơ để robot di chuyển chân đến vị trí chỉ định. Nếu quay trái, thứ tự di chuyển của từng chân là: Phải trước – trái trước – trái sau – phải sau. Khi quay phải thứ tự di chuyển của từng chân ngược lại với khi quay trái. Nếu có lệnh dừng khẩn cấp thì sẽ dừng điều khiển động cơ ngay lập tức và xóa mode.
- Bước 4: Xét cờ STOP, nếu cờ được bật thì xóa mode và đợi đến khi robot hoàn thành xong chu kỳ xoay.

Đối với lệnh quay phải, giải thuật tương tự với lệnh quay trái.

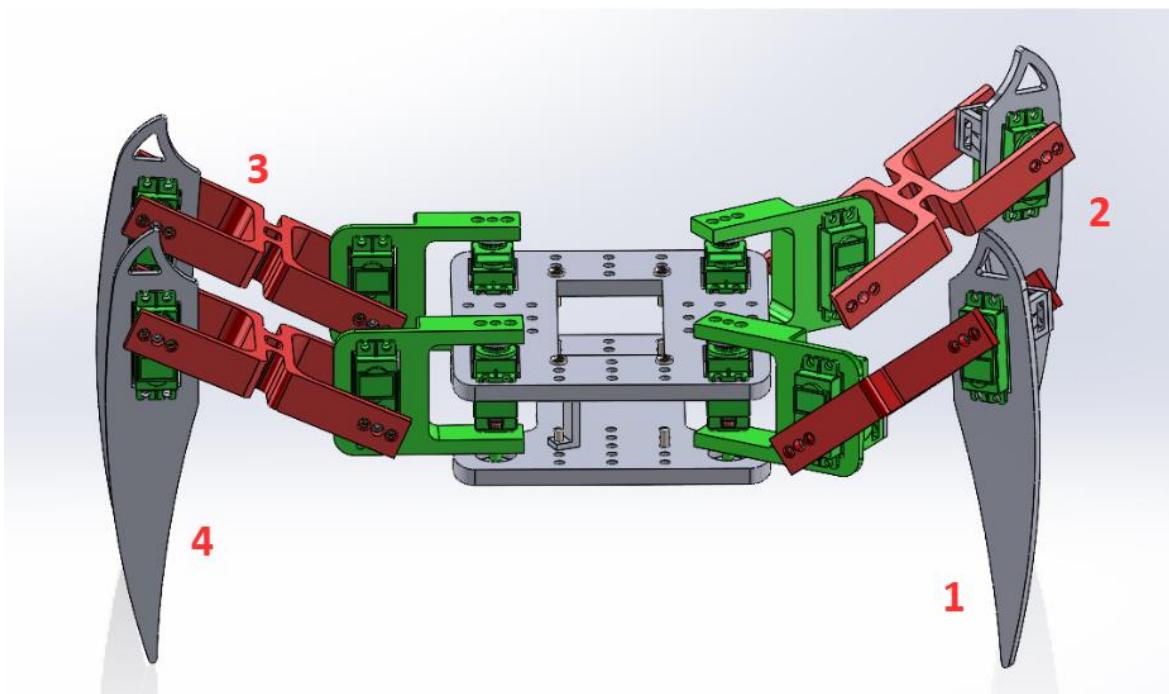
#### **4.1.1.3. Mode cân bằng thân robot**

Chương trình sẽ gửi mode lệnh và các tham số để điều khiển cân bằng thân robot, gồm giá trị Kp, Ki, Kd, giá trị tham chiếu góc roll và pitch. Robot sẽ tính toán và xuất ra tín hiệu điều khiển là  $u_{roll}$  và  $u_{pitch}$  theo thuật toán của bộ điều khiển PID, sau đó tín hiệu điều khiển được đưa qua khâu khuếch đại bão hòa với hệ số khuếch đại là 1 và giá trị tối đa của nó là 80. Việc này giúp robot không bị ngã khi tín hiệu điều khiển quá lớn, bảo vệ robot khỏi những va chạm không đáng có. Sau đó robot sẽ điều chỉnh chiều cao để đáp ứng cả góc roll và góc pitch một cách đồng thời.

Để điều khiển chân của robot ta cần 3 tham số đó là tọa độ của chân trong không gian (x, y, z). Để điều khiển góc nghiêng của robot ta cần điều khiển vị trí của cả 4 chân để robot có thể giữ thăng bằng.

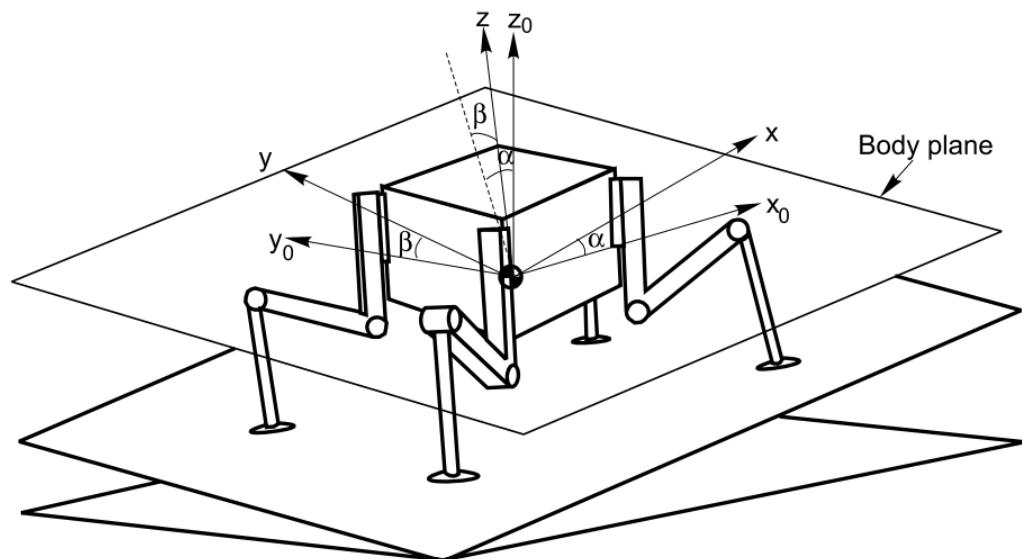


Hình 4.3 Robot trạng thái đứng-mặt trên

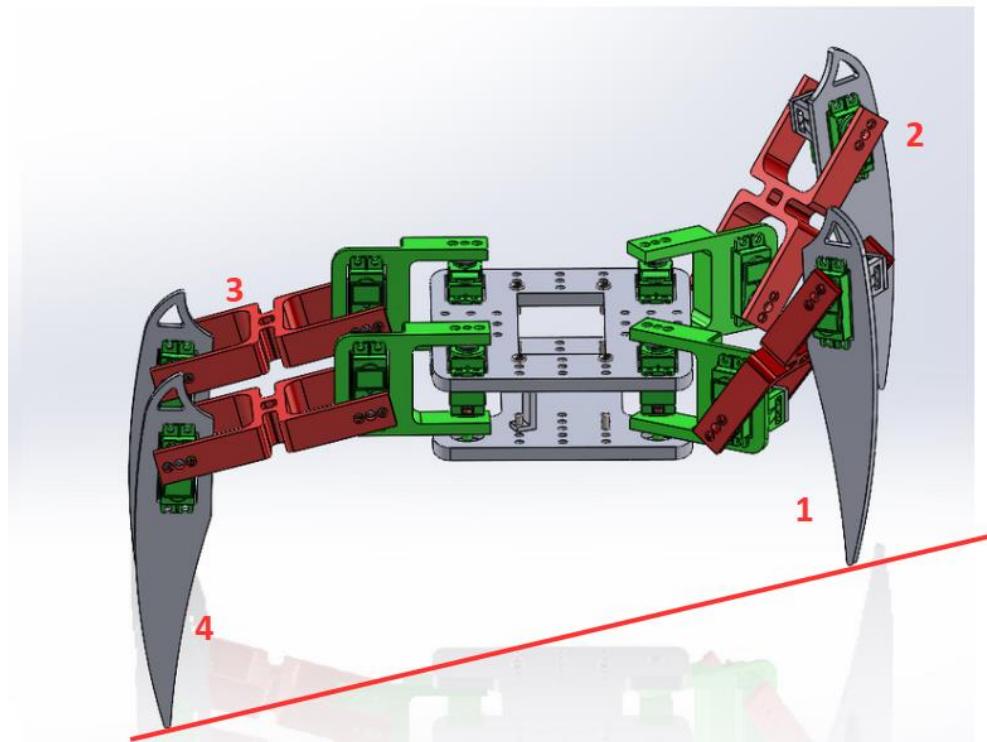


Hình 4.4 Robot trạng thái đứng-mặt trước

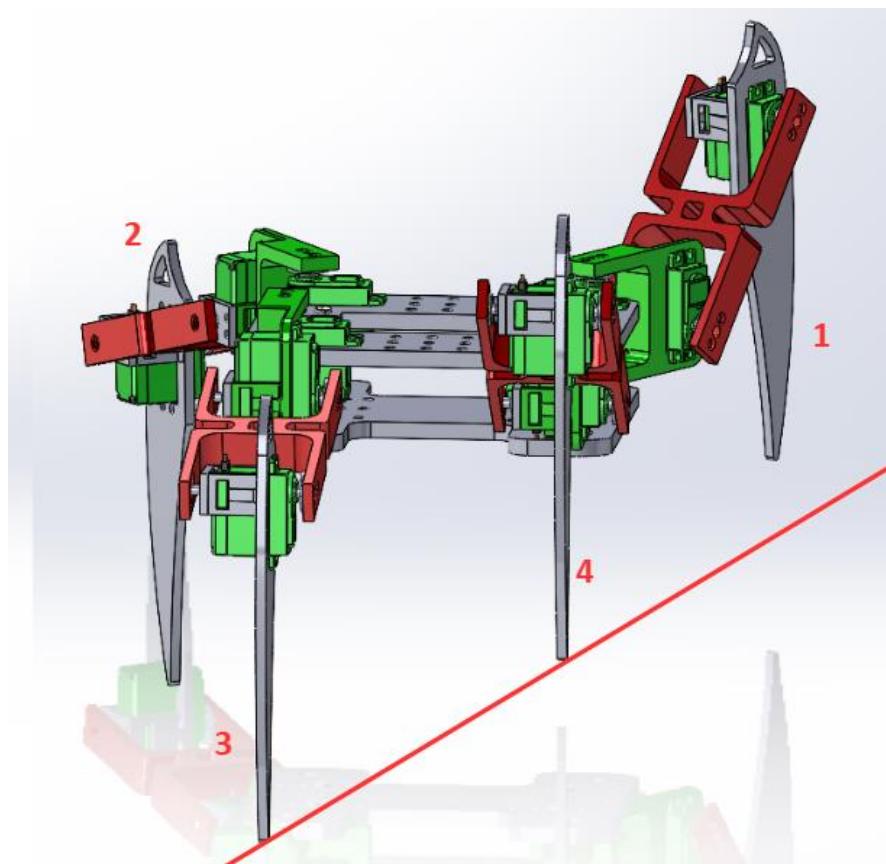
Đây là trạng thái đứng của robot, để dễ hình dung ta sẽ đánh số thứ tự cho các chân của robot. Lần lượt từ chân trái-trước của robot đến chân phải-trước của robot ngược chiều kim đồng hồ.



Hình 4.5 Điều chỉnh chiều cao của từng chân để cân bằng thân robot



Hình 4.6 Măt phẳng nghiêng theo góc Pitch



Hình 4.7 Măt phẳng nghiêng theo góc Roll

Để thân robot cân bằng, ta phải hiệu chỉnh chiều cao của mỗi chân theo công thức sau:

$$z\_1 = z + u\_pitch - u\_roll$$

$$z\_2 = z + u\_pitch + u\_roll$$

$$z\_3 = z - u\_pitch + u\_roll$$

$$z\_4 = z - u\_pitch - u\_roll$$

Sau khi có vị trí của 4 chân, chương trình sẽ tính toán ra góc quay cần thiết của mỗi servo và điều khiển 4 chân của robot để thân robot cân bằng.

Việc cân bằng thân robot chỉ được thực hiện khi robot ở trạng thái đứng vì khi robot di chuyển thân robot dao động với biên độ lớn và tần số khá cao. Bộ điều khiển PID không điều khiển được tốt trong trường hợp này. Robot sẽ luôn điều chỉnh vị trí các chân đến khi người dùng nhấn nút dừng hoặc dừng khẩn cấp trên giao diện.

#### 4.1.1.4. Mode điều khiển từng chân

Chương trình sẽ gửi mode lệnh, thứ tự chân và vị trí xuống cho robot.

Robot sẽ tính toán góc của 3 khớp theo công thức động học ngược được trình bày bên trên và điều khiển chân đến vị trí chỉ định. Giải thuật điều khiển từng chân của robot như sau:

- Bước 1: Xét thứ tự chân và vị trí chỉ định.
- Bước 2: Từ tọa độ chỉ định. Tính toán góc quay cho từng động cơ servo bằng công thức động học ngược.
- Bước 3: Điều khiển động cơ để robot di chuyển chân đến vị trí chỉ định. Nếu có lệnh dừng khẩn cấp thì sẽ dừng điều khiển động cơ ngay lập tức và xóa mode.
- Bước 4: Sau khi di chuyển chân robot đến vị trí chỉ định, xóa mode.

#### **4.1.1.5. Mode điều khiển từng động cơ**

Chương trình sẽ gửi mode lệnh, thứ tự động cơ và góc quay xuống cho robot. Robot sẽ điều khiển động cơ quay đến góc chỉ định. Giải thuật điều khiển từng động cơ của robot như sau:

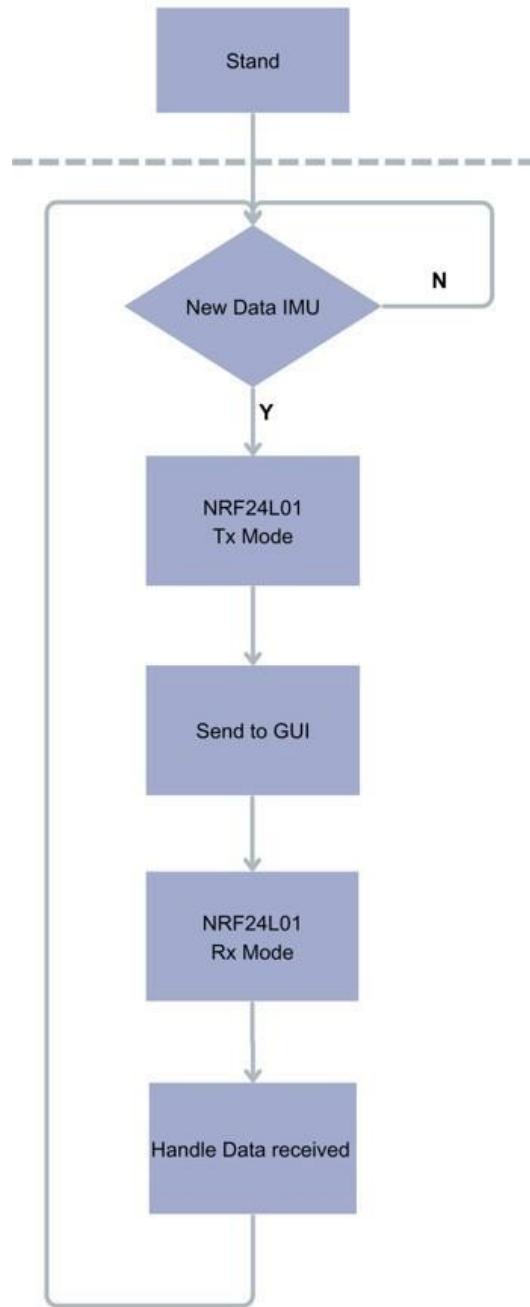
- Bước 1: Xét thứ tự động cơ và góc quay chỉ định.

- Bước 2: Điều khiển động cơ để robot di chuyển chân đến vị trí chỉ định. Nếu có lệnh dừng khẩn cấp thì sẽ dừng điều khiển động cơ ngay lập tức và xóa mode.

- Bước 3: Sau khi di chuyển động cơ robot đến góc chỉ định, xóa mode.

Khi điều khiển góc của từng động cơ, động cơ sẽ không quay được quá góc giới hạn đã trình bày bên trên để tránh va chạm trong quá trình điều khiển.

#### 4.1.2. Luồng truyền nhận dữ liệu với máy tính

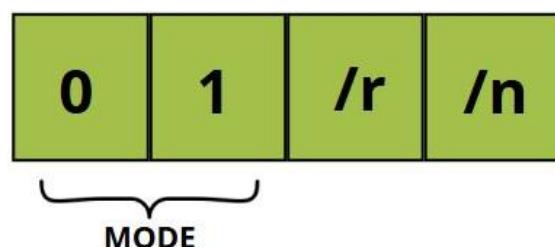


Hình 4.8 Luồng truyền nhận dữ liệu với máy tính

Bởi vì việc giao tiếp không dây chỉ có thể truyền hoặc nhận dữ liệu tại từng thời điểm nên ta sẽ thiết lập cho module mặc định là ở trạng thái nhận dữ liệu và chỉ chuyển sang trạng thái truyền dữ liệu nếu có sự kiện xảy ra. Vậy nên,

mỗi khi dữ liệu cảm biến được cập nhật, module giao tiếp không dây sẽ được chuyển sang trạng thái Tx (trạng thái truyền dữ liệu) và gửi dữ liệu đến module điều khiển. Sau khi dữ liệu được gửi xong, module giao tiếp không dây sẽ được chuyển về trạng thái Rx (trạng thái nhận dữ liệu) để có thể sẵn sàng nhận lệnh điều khiển mới.

#### 4.1.2.1. Mode đứng, ngồi, stop và emergency

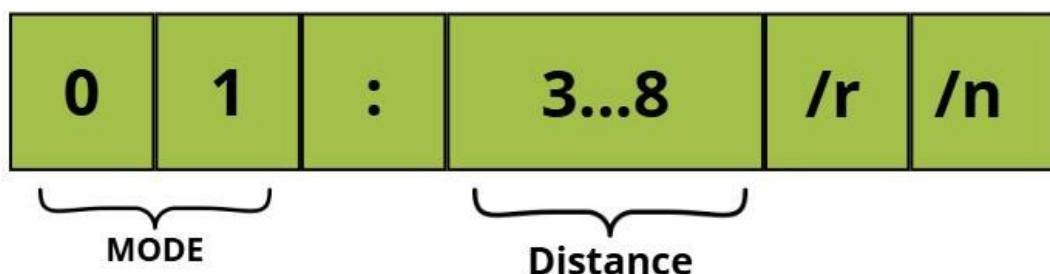


Hình 4.9 Frame của mode đứng, ngồi, stop và emergency

Frame truyền sẽ sử dụng dạng ASCII, điều này tuy tăng số byte cần truyền nhưng dễ dàng cho việc tách dữ liệu. Đồng thời các byte mode sẽ đóng vai trò là các byte kiểm tra để đảm bảo dữ liệu nhận được là chính xác, 2 byte cuối có giá trị “/r”, “/n” giúp xác định chuỗi dữ liệu đã được nhận đầy đủ hay chưa.

Để điều khiển robot đứng, ngồi, dừng hay dừng khẩn cấp chỉ cần truyền giá trị của mode để robot có thể thực hiện các hành động tương ứng.

#### 4.1.1.6. Mode di chuyển tiến, lùi

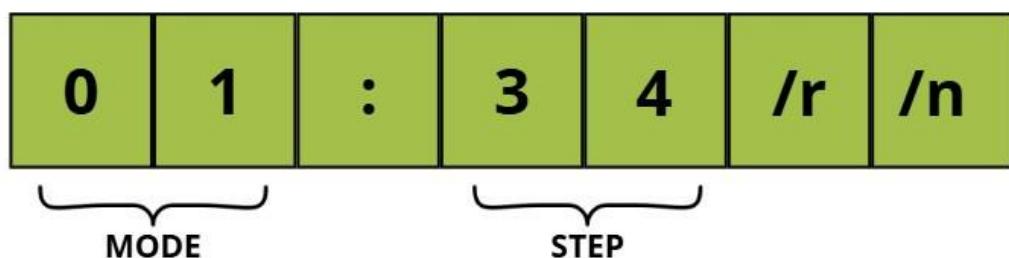


Hình 4.10 Frame của mode di chuyển tiến lùi

Trong đó:

- 2 byte đầu là mode điều khiển.
- Byte 2: phân tách dữ liệu với mode
- Byte 3 – 8: khoảng cách cần di chuyển, dữ liệu dạng số nguyên và đơn vị là mm
- 2 byte cuối: kết thúc frame truyền

#### 4.1.2.2. Mode xoay trái, xoay phải



Hình 4.11 Frame của mode xoay trái, xoay phải

Trong đó:

- 2 byte đầu là mode điều khiển.
- Byte 2: phân tách dữ liệu với mode
- Byte 3 – 4: số chu kỳ thực hiện
- 2 byte cuối: kết thúc frame truyền

#### 4.1.2.3. Mode điều khiển từng chân robot

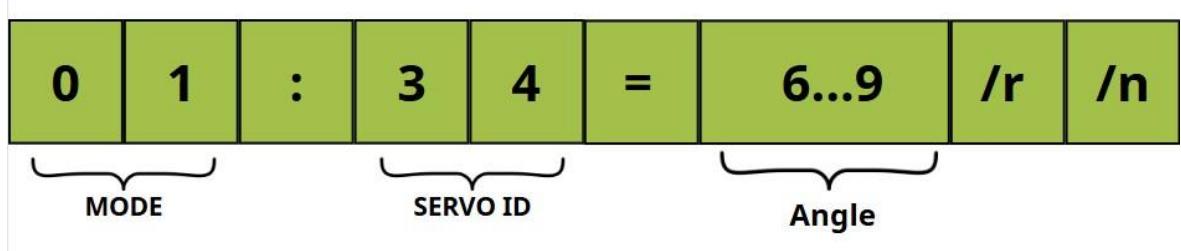


Hình 4.12 Frame của mode điều khiển từng chân robot

Trong đó:

- 2 byte đầu là mode điều khiển.
- Byte 2: phân tách dữ liệu với mode
- Byte 3 – 27: vị trí chỉ định, dạng số thực có hai chữ số thập phân và có thể mang giá trị âm.
- Byte 28: thứ tự chân cần điều khiển
- 2 byte cuối: kết thúc frame truyền

#### 4.1.2.4. Mode điều khiển từng động cơ robot

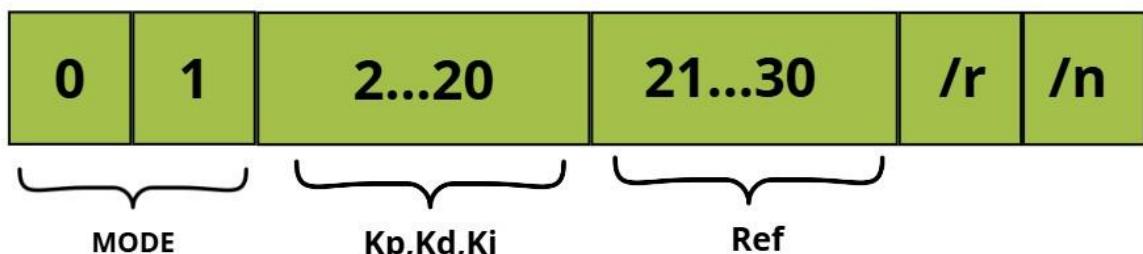


Hình 4.13 Frame của mode điều khiển từng động cơ robot

Trong đó:

- 2 byte đầu là mode điều khiển.
- Byte 2: phân tách dữ liệu với mode
- Byte 3 – 4: thứ tự động cơ cần điều khiển
- Byte 5: phân tách id và giá trị góc chỉ định
- Byte 6 – 9: Góc chỉ định, dạng số nguyên và có thể mang giá trị âm
- 2 byte cuối: kết thúc frame truyền

#### 4.1.2.5. Mode cân bằng thân robot

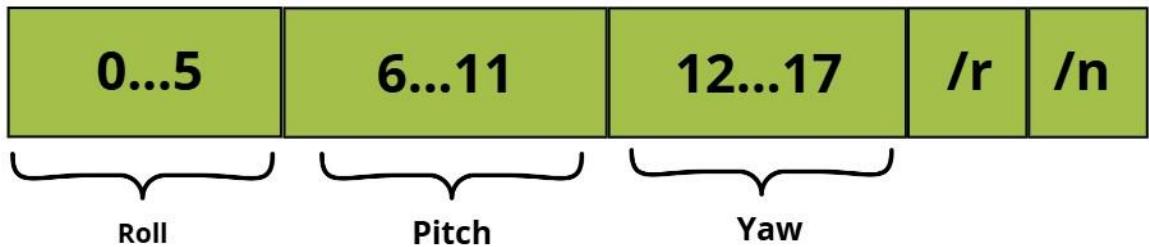


Hình 4.14 Frame của mode cân bằng thân robot

Trong đó:

- 2 byte đầu là mode điều khiển.
- Byte 2 – 20: giá trị Kp, Ki, Kd dạng số thực có 3 chữ số thập phân và chỉ nhận giá trị dương
- Byte 21 – 30: giá trị tham chiếu góc roll, pitch, yaw dạng số nguyên và có thể mang giá trị âm
- 2 byte cuối: kết thúc frame truyền

#### 4.1.2.6. Gửi dữ liệu cảm biến từ robot đến máy tính



Hình 4.15 Frame dữ liệu cảm biến gửi đến máy tính

Trong đó:

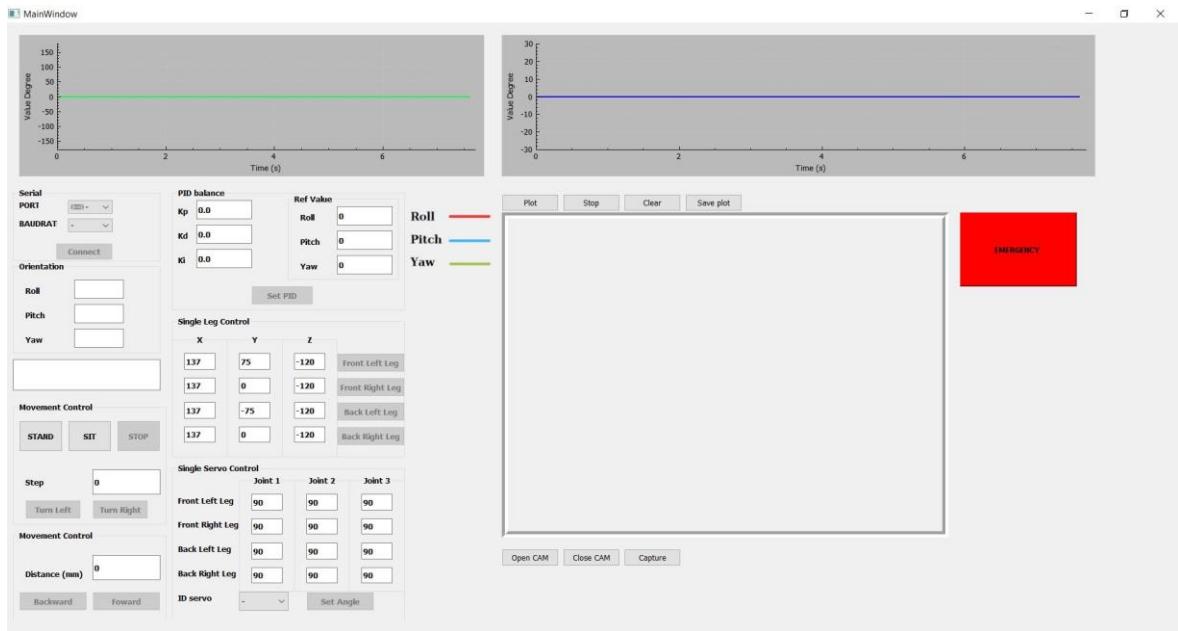
- Byte 0 – 5: Giá trị góc Roll dạng số thực và có thể mang giá trị âm
- Byte 6 – 11: Giá trị góc Pitch dạng số thực và có thể mang giá trị âm
- Byte 12 – 17: Giá trị góc Yaw dạng số thực và có thể mang giá trị âm
- 2 byte cuối: kết thúc frame truyền

## 4.2. Xác định hướng và vị trí robot trên giao diện

### 4.2.1. Xác định hướng bằng IMU

Giá trị đọc được của cảm biến gia tốc gồm 3 góc: Roll, Pitch và Yaw. Góc Roll và Pitch thể hiện độ nghiêng của thân robot và góc yaw thể hiện

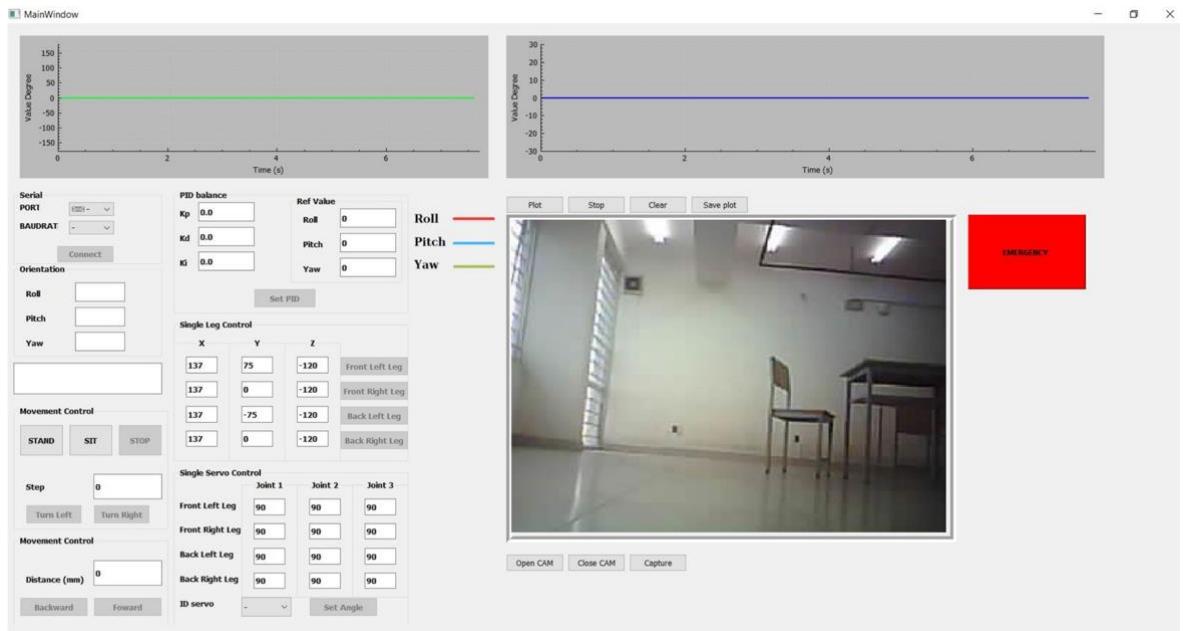
hướng hiện tại của robot. Từ đó thị và ô giá trị các góc trên giao diện ta có thể xác định hướng của robot và độ nghiêng của thân.



Hình 4.16 Xác định hướng bằng IMU

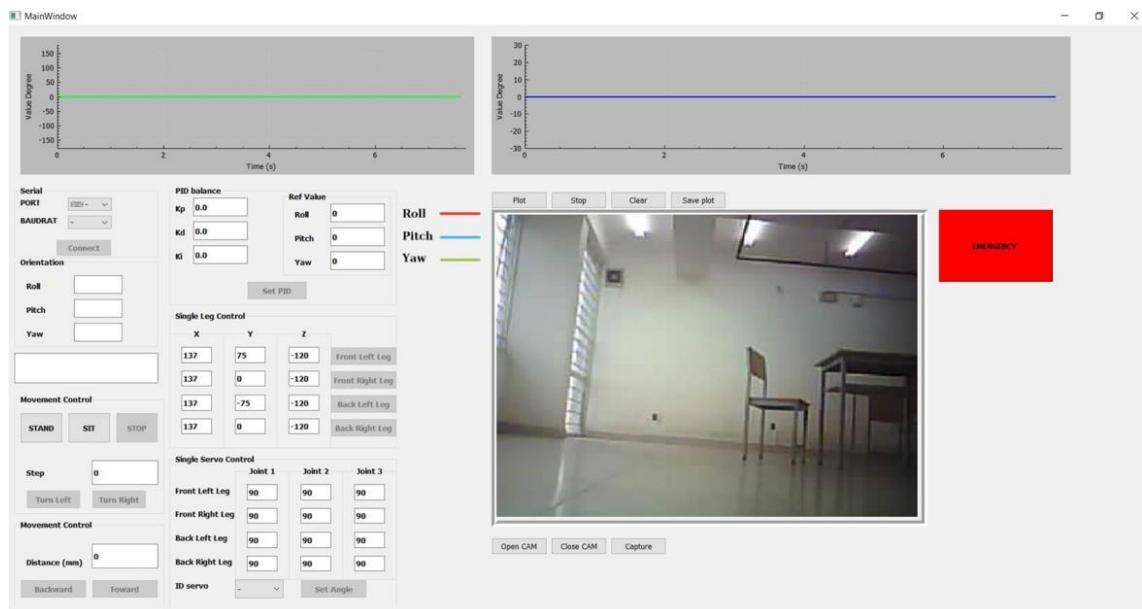
#### 4.2.2. Xác định vị trí tương đối qua Camera

Robot được trang bị camera là ESP Cam, giao tiếp bằng wifi với giao diện trên máy tính. Nhờ vậy người dùng sẽ quan sát được hình ảnh của camera thu được và xác định được vị trí tương đối của robot hiện tại. Camera chỉ có tác dụng thu thập hình ảnh và gửi lên giao diện, không có tác dụng trong việc thực hiện các chức năng của robot vậy nên trong quá trình điều khiển có thể không bật camera để giảm năng lượng tiêu thụ và giảm băng thông trong quá trình truyền nhận dữ liệu giữa robot với máy tính.



Hình 4.17 Xác định vị trí tương đối bằng camera

### 4.3. Thiết kế giao diện điều khiển trên máy tính



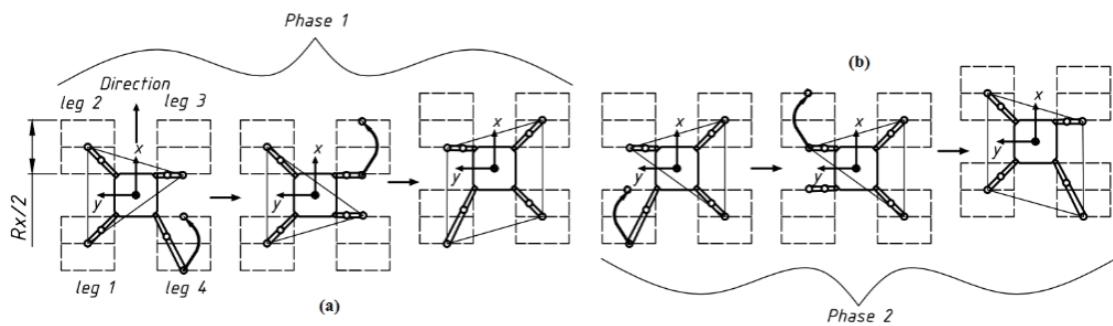
Hình 4.18 Giao diện điều khiển trên máy tính

Nhóm sử dụng phần mềm PyQt5 để thiết kế giao diện điều khiển cho robot. Giao diện điều khiển gồm hai phần chính: bảng điều khiển, đồ thị góc roll, pitch, yaw đọc được từ cảm biến và camera được đặt trên robot. Bảng điều khiển gồm:

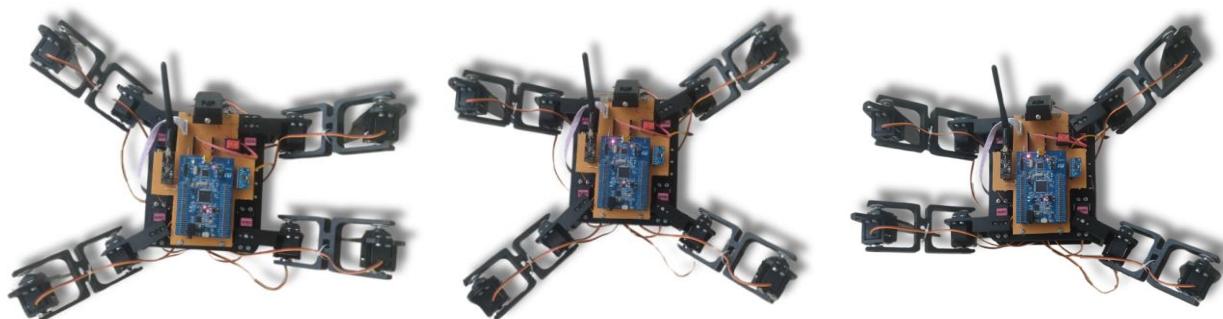
- Khối giao tiếp UART giao tiếp với vi điều khiển thông qua cổng COM
- Khối hiệu chỉnh thông số điều khiển PID
- Khối hiển thị giá trị chính xác góc roll, pitch, yaw đọc được từ cảm biến
- Khối điều khiển từng chân của robot
- Khối điều khiển từng động cơ servo
- Khối ra lệnh di chuyển tiến, lùi, xoay trái, xoay phải và dừng khẩn cấp

## Chương 5. KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ

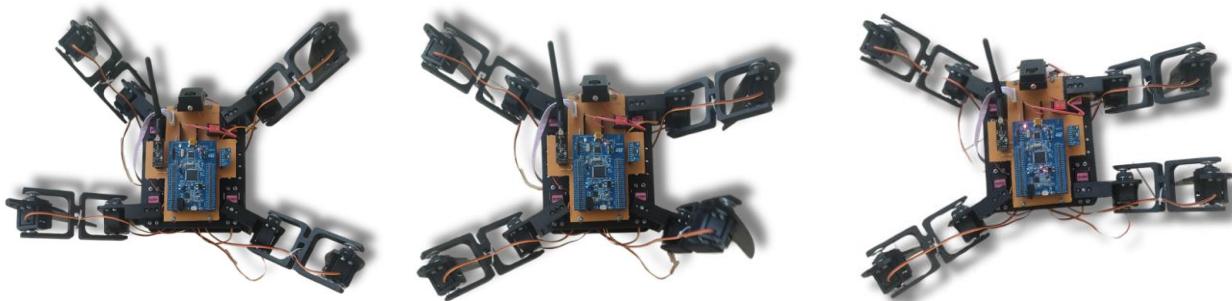
### 5.1. Di chuyển tiến lùi



Hình 5.1 Sơ đồ dáng đi tới, lui hai pha không liên tục



Hình 5.2 Sơ đồ dáng đi tới, lui thực tế - Pha 1



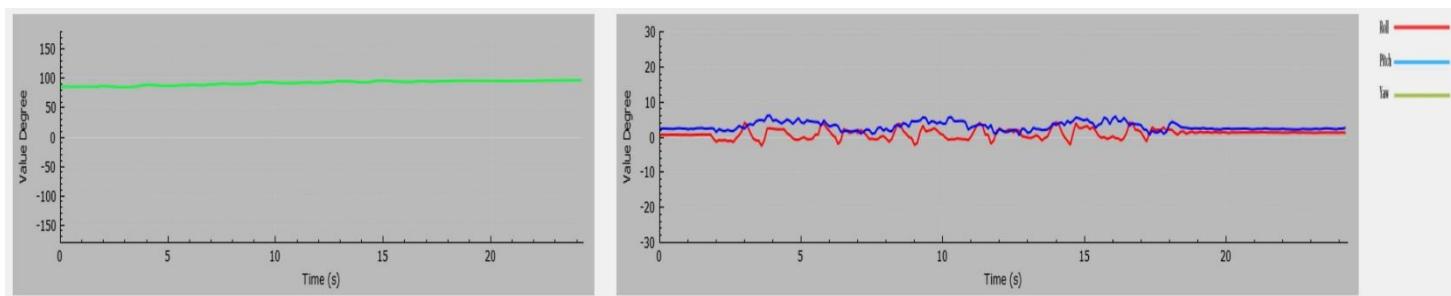
Hình 5.3 Sơ đồ dáng đi tới, lui thực tế - Pha 2

Sau khi hoàn thành một pha, cơ thể robot sẽ tiến lên/lùi xuống 1 khoảng x (mm). Sau khi hoàn thành 1 chu kỳ bước, cơ thể robot tiến lên/lùi xuống 1 đoạn khoảng 2x (mm) và quay lại đúng ban đầu.

Thực hiện đi tới 10 lần với khoảng cách là 400mm ta có:

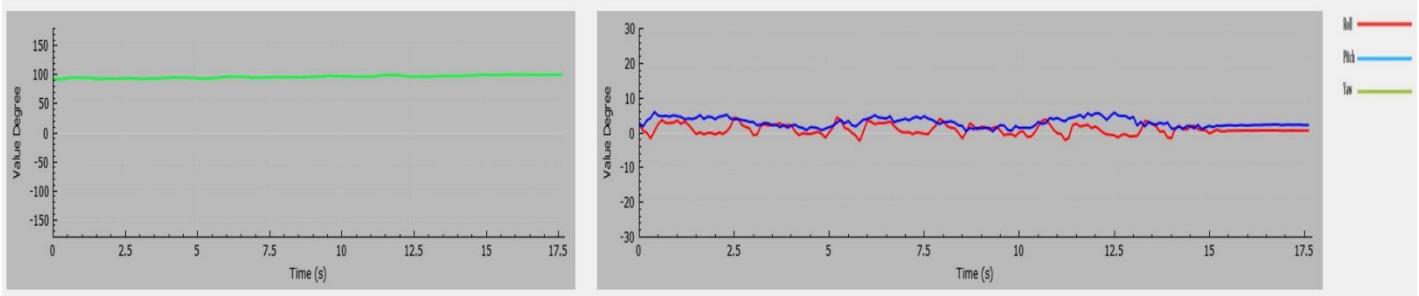
Lần	Khoảng cách (mm)	Sai số
1	402	2
2	406	6
3	405	5
4	390	-10
5	393	-7
6	393	-7
7	401	1
8	405	5
9	396	-4
10	407	7
Trung bình	399.8	5.4

Bảng 5.1 Khoảng cách di chuyển sau mỗi chu kỳ tiến lên



Hình 5.4 Góc và hướng thân robot trong khi đi tới liên tục lần 1

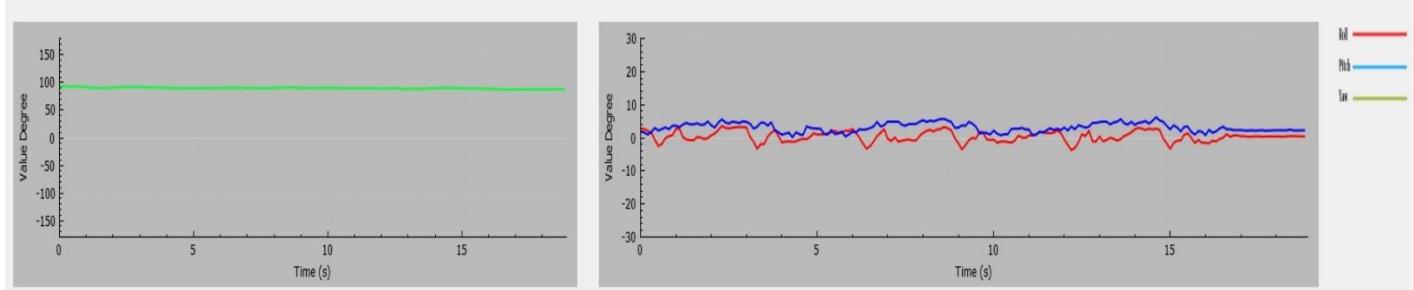
Di chuyển 400mm lần thứ 2:



Hình 5.5 Góc và hướng thân robot trong khi đi tới liên tục lần 2

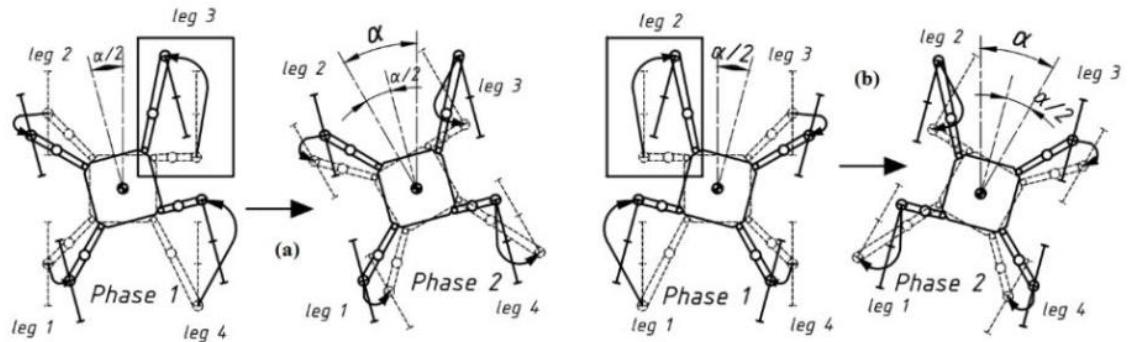
Khi robot đi tới liên tục, góc nghiêng của thân được biểu diễn như đồ thị hình 5.x. Góc Roll dao động trong khoảng từ -4 độ đến 4 độ so với giá trị đặt. Góc Pitch dao động từ -2 độ đến 4 độ. Góc yaw vẫn giữ nguyên giá trị trong suốt quá trình di chuyển. Kết luận, trong khi di chuyển tới lui, thân robot bị nghiêng không quá 5 độ. Robot di chuyển một cách ổn định. Tuy nhiên, khi di chuyển càng lâu, sai số khoảng cách của robot càng lớn. Lý do dẫn đến điều này là do động cơ hoạt động không tốt, khi đã qua sử dụng một thời gian động cơ sẽ xuống cấp. Lý do thứ hai, nguồn điện không đủ đáp ứng cho 12 động cơ servo nên chúng bị yếu, giảm hiệu quả hoạt động của robot.

Tương tự cho đi lùi với khoảng cách là 400mm:

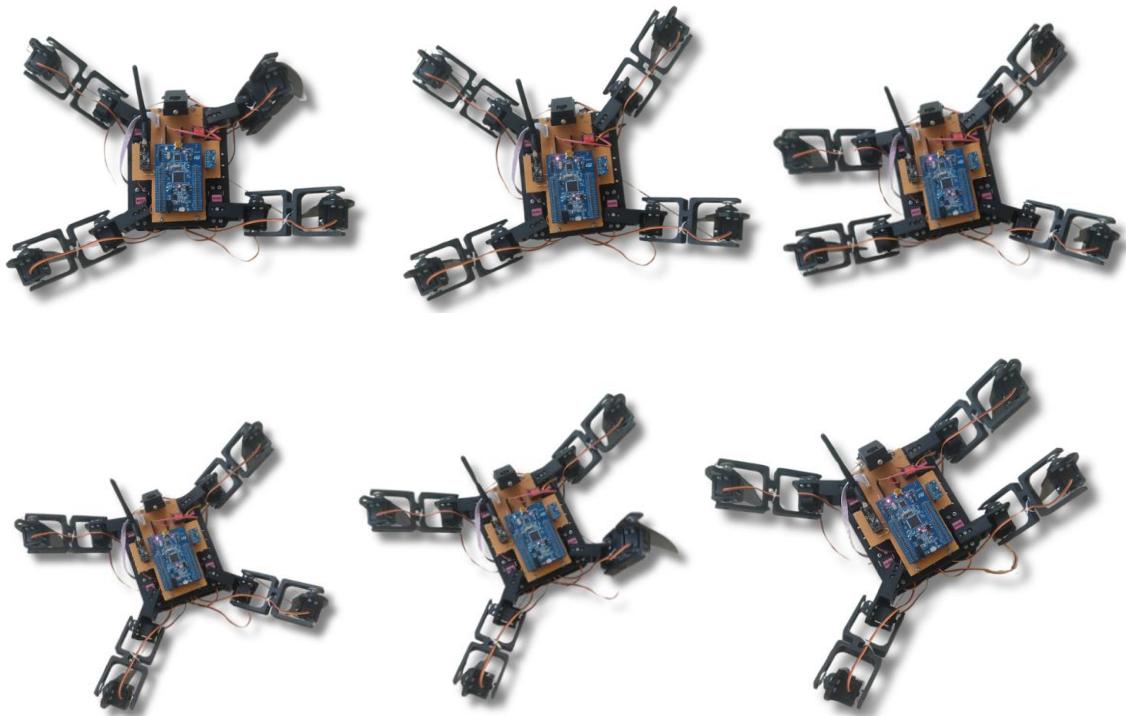


Hình 5.6 Góc nghiêng thân robot trong khi đi lùi liên tục

## 5.2. Xoay trái xoay phải



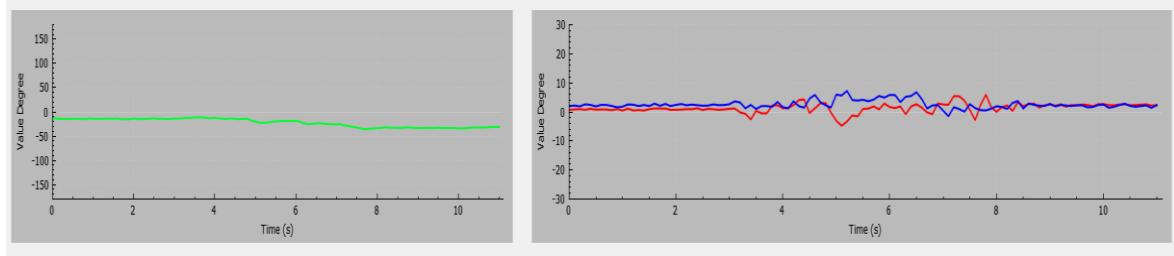
Hình 5.7 Sơ đồ dáng xoay trái (a), xoay phải (b)



Hình 5.8 Sơ đồ dáng xoay trái của robot

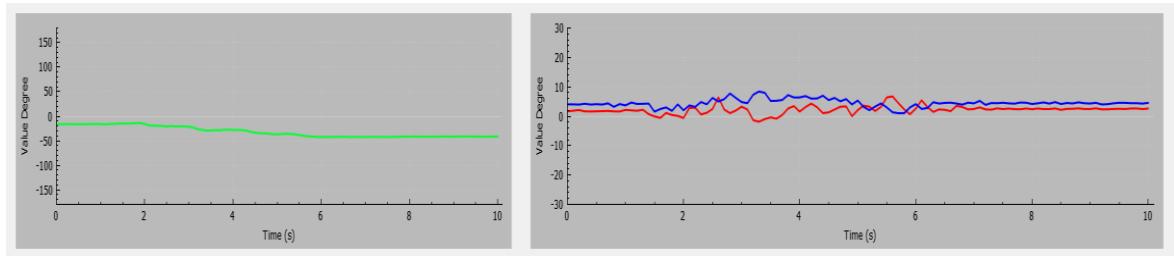
Dáng đi xoay phải tương tự với dáng đi xoay trái. Sau khi hoàn thành chu kỳ xoay trái, thân robot xoay khoảng 30 độ. Giúp cho robot có thể linh hoạt thay đổi hướng đi để tránh vật cản khi di chuyển.

Lần 1:



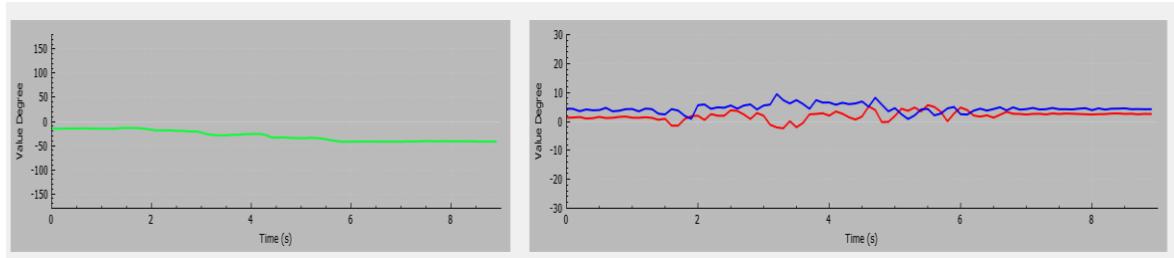
Hình 5.9 Góc nghiêng thân robot trong khi xoay trái lần 1

Lần 2:



Hình 5.10 Góc nghiêng thân robot trong khi xoay trái lần 2

Lần 3:



Hình 5.11 Góc nghiêng thân robot trong khi xoay trái lần 3

Kết luận: Khi thực hiện xoay trái/ phải các góc roll pitch của robot nghiêng không quá 8 độ. Robot tương đối cân bằng và không có hiện tượng đổ về một phía. Tuy nhiên vẫn có sai số giữa các lần quay của robot là khoảng 2 độ. Điều này có thể do tiếp xúc chân của robot với mặt sàn không tốt dẫn đến bị trượt.

### 5.3. Cân bằng thân robot bằng PID

#### 5.3.1. Nghiêng theo góc chỉ định

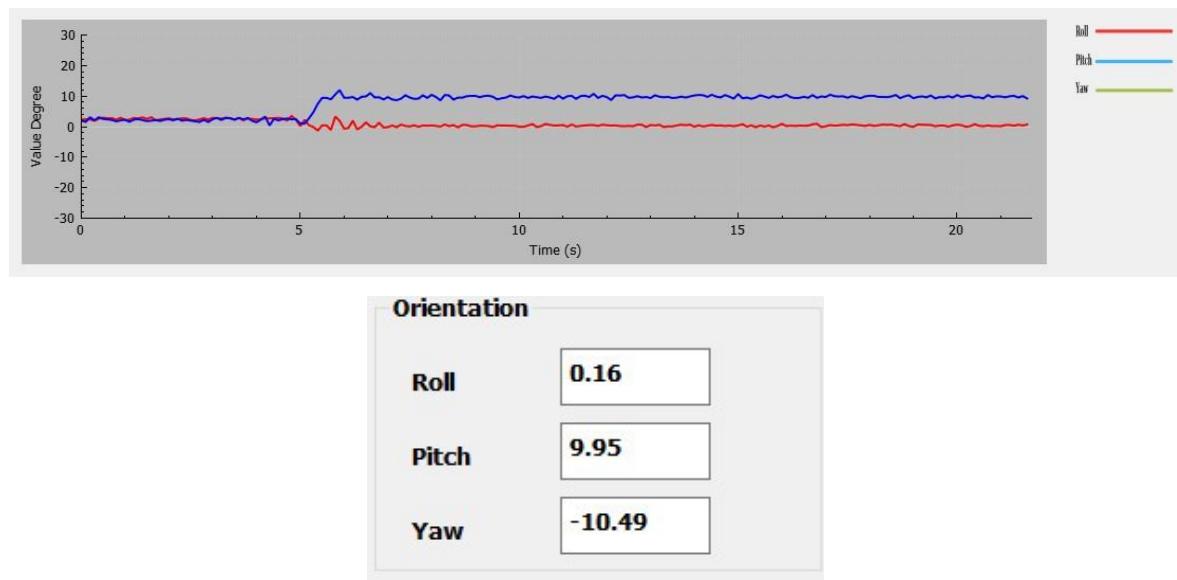
Đặt góc nghiêng chỉ định cho robot:

$$\text{Roll} = 0^0$$

$$\text{Pitch} = 10^0$$

#### Hiệu chỉnh K<sub>p</sub>:

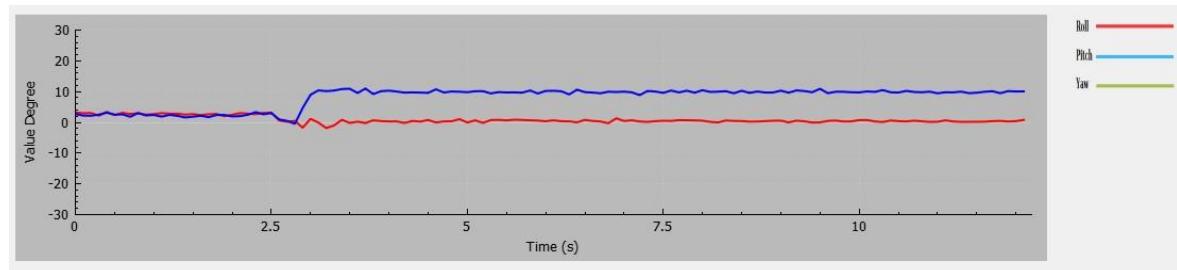
TH1: K<sub>p</sub> = 0.5, K<sub>i</sub> = 0, K<sub>d</sub> = 0:



Hình 5.12 Nghiêng robot theo góc pitch TH1

Thời gian xác lập 0.7s, gần như không có vọt lố mà chỉ có nhiễu đo lường từ cảm biến.

TH2: K<sub>p</sub> = 1, K<sub>i</sub> = 0, K<sub>d</sub> = 0:

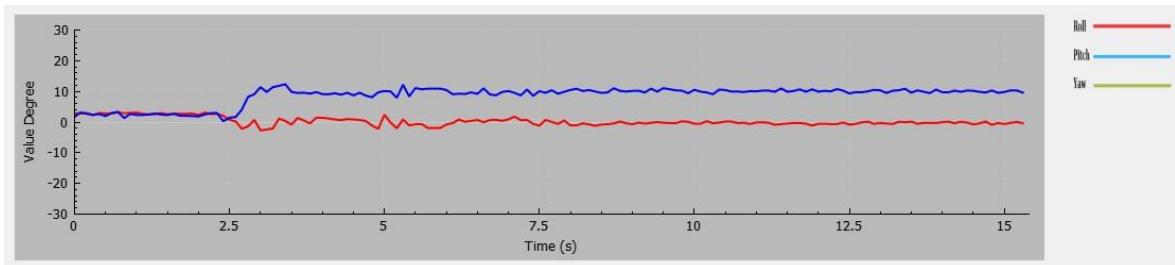


Orientation	
Roll	0.33
Pitch	9.87
Yaw	-9.1

Hình 5.13 Nghiêng robot theo góc pitch TH2

Thời gian xác lập khoảng 0.5s, đáp ứng bám tốt với giá trị đặt.

TH3:  $K_p = 2$ ,  $K_i = 0$ ,  $K_d = 0$ :

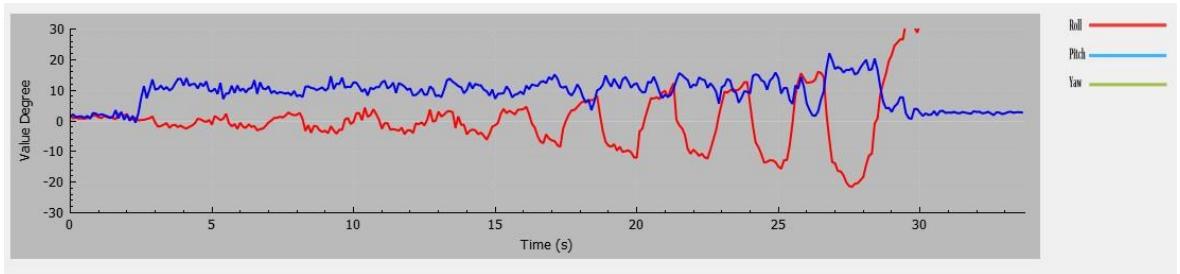


Orientation	
Roll	-0.52
Pitch	9.93
Yaw	-10.73

Hình 5.14 Nghiêng robot theo góc pitch TH3

$K_p$  lớn nên đáp ứng có vọt lố khoảng 2 độ, sau đó đáp ứng bám tốt với giá trị đặt, sai số  $< 0.7$  độ. Do giá trị đọc được của cảm biến vẫn còn nhiều và chưa có  $K_i$  nên khó để có thể giảm sai số xác lập.

TH4:  $K_p = 3$ ,  $K_i = 0$ ,  $K_d = 0$ :

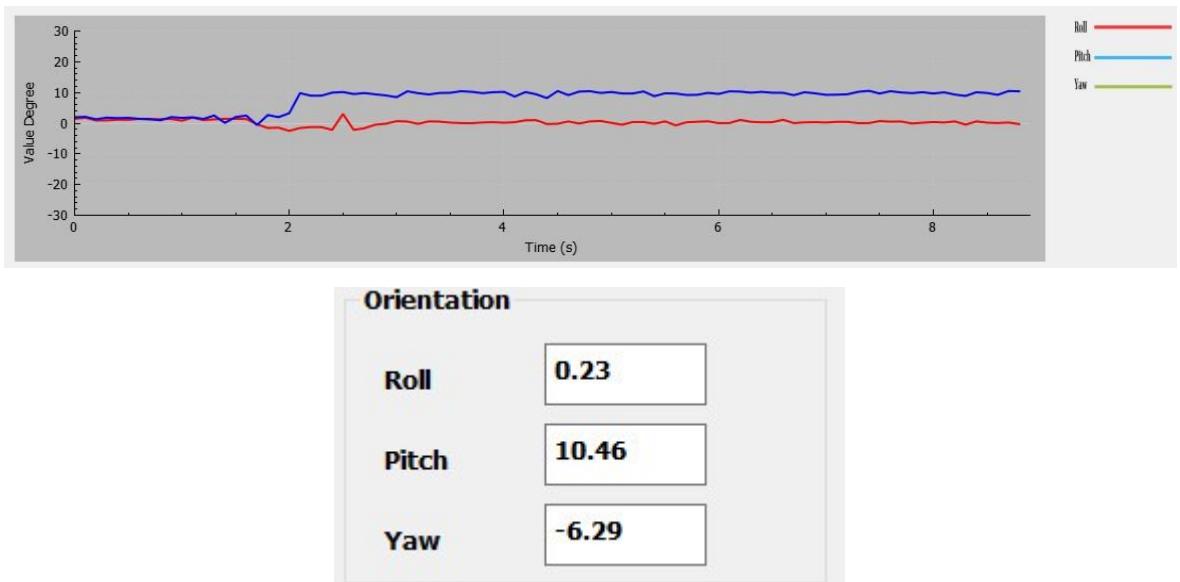


Hình 5.15 Nghiêng robot theo góc pitch TH4

Khi  $K_p$  quá lớn, hệ bị mất ổn định và bắt đầu dao động.

### Hiệu chỉnh PI:

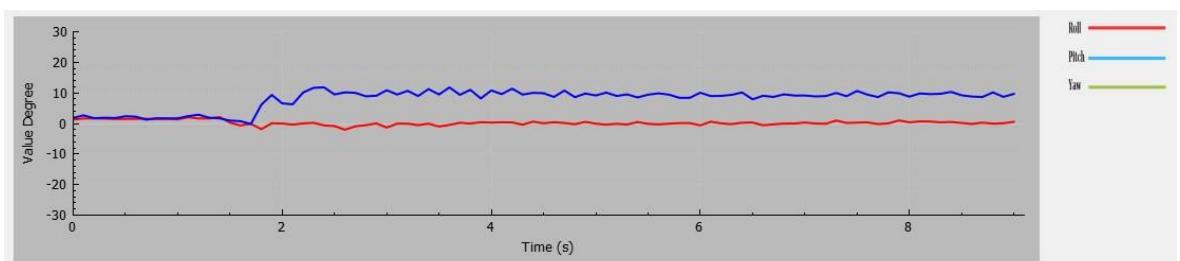
TH5:  $K_p = 0.5$ ,  $K_i = 0.5$ ,  $K_d = 0$ :



Hình 5.16 Nghiêng robot theo góc pitch TH5

Thời gian xác lập khoảng 0.8s, sai số xác lập giảm còn < 0.5 độ.

TH6:  $K_p = 0.5$ ,  $K_i = 0.7$ ,  $K_d = 0$ :

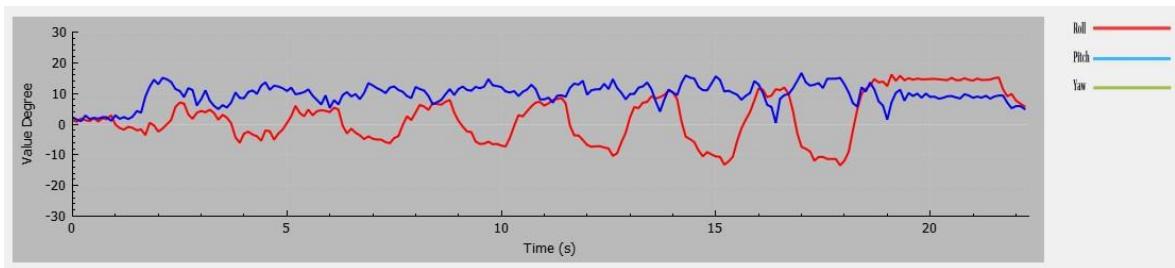


Orientation	
Roll	0.14
Pitch	9.87
Yaw	-7.36

Hình 5.12 Nghiêng robot theo góc pitch TH6

Thời gian xác lập khoảng 0.6s, sai số xác lập khoảng  $< 0.6$  độ

TH7:  $K_p = 1.2$ ,  $K_i = 1.2$ ,  $K_d = 0$ :

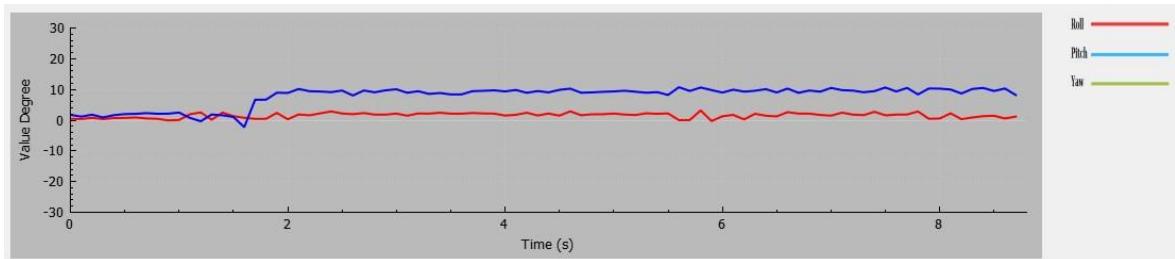


Hình 5.13 Nghiêng robot theo góc pitch TH7

$K_i$  lớn dẫn đến hệ môt ổn định và dao động.

### Hiệu chỉnh PID:

TH8:  $K_p = 0.5$ ,  $K_i = 0.5$ ,  $K_d = 0.3$ :

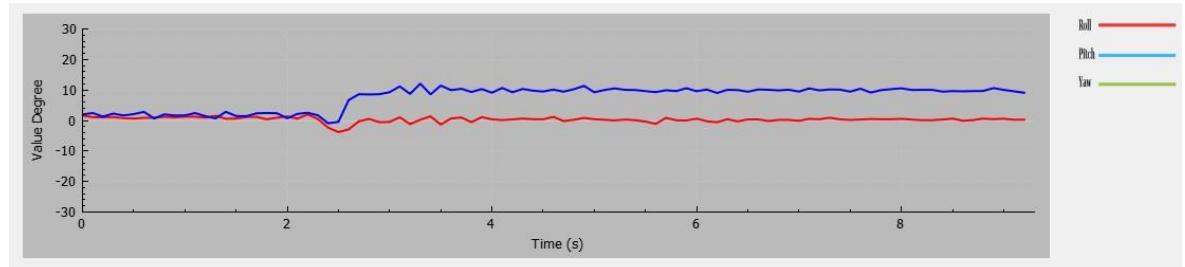


Orientation	
Roll	<b>0.76</b>
Pitch	<b>10.5</b>
Yaw	<b>-4.09</b>

Hình 5.14 Nghiêng robot theo góc pitch TH8

Thời gian xác lập khoảng 0.7s, sai số xác lập  $< 0.9$  độ.

TH9:  $K_p = 0.7$ ,  $K_i = 0.5$ ,  $K_d = 0.3$ :

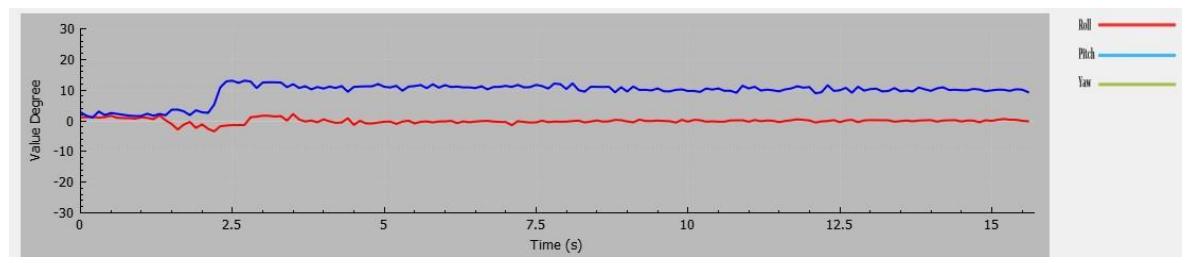


Orientation	
Roll	<b>-0.38</b>
Pitch	<b>10.1</b>
Yaw	<b>-4.56</b>

Hình 5.15 Nghiêng robot theo góc pitch TH9

Thời gian xác lập giảm còn 0.3s, sai số xác lập  $< 0.5$  độ và không có vọt lố.

TH10:  $K_p = 0.5$ ,  $K_i = 0.5$ ,  $K_d = 0.5$ :





Hình 5.16 Nghiêng robot theo góc pitch TH10

Kd hơi lớn dẫn đến vọt lố khoảng 3 độ, sau đó đám ứng bám dần về giá trị đặt. Thời gian xác lập khoảng 0.5s và sai số xác lập  $< 0.5$  độ.

Kết luận: Bộ điều khiển PID điều khiển tốt và ổn định cho việc cân bằng thân robot nghiêng theo góc pitch.

Kp	Ki	Kd	Đáp ứng
0.5	0	0	Không vọt lố, $T_{x1} = 0.7s$ , $e_{x1} = 0.5$
1	0	0	Không vọt lố, $T_{x1} = 0.5s$ , $e_{x1} = 0.6$
2	0	0	Vọt lố 2 độ, $T_{x1} = 0.5s$ , $e_{x1} = 0.5$
3	0	0	Hệ mất ổn định
0.5	0.5	0	Không vọt lố, $T_{x1} = 0.8s$ , $e_{x1} = 0.46$
0.5	0.7	0	Không vọt lố, $T_{x1} = 0.6s$ , $e_{x1} = 0.6$
1.2	1.2	0	Hệ mất ổn định
0.5	0.5	0.3	Không vọt lố, $T_{x1} = 0.7s$ , $e_{x1} = 0.9$
0.7	0.5	0.3	Không vọt lố, $T_{x1} = 0.3s$ , $e_{x1} = 0.5$
0.5	0.5	0.5	Vọt lố 3 độ, $T_{x1} = 0.5s$ , $e_{x1} = 0.5$

Bảng 5.2 Đáp ứng điều khiển PID theo góc pitch

Tương tự với nghiêng góc pitch. Trường hợp đặt góc nghiêng theo roll

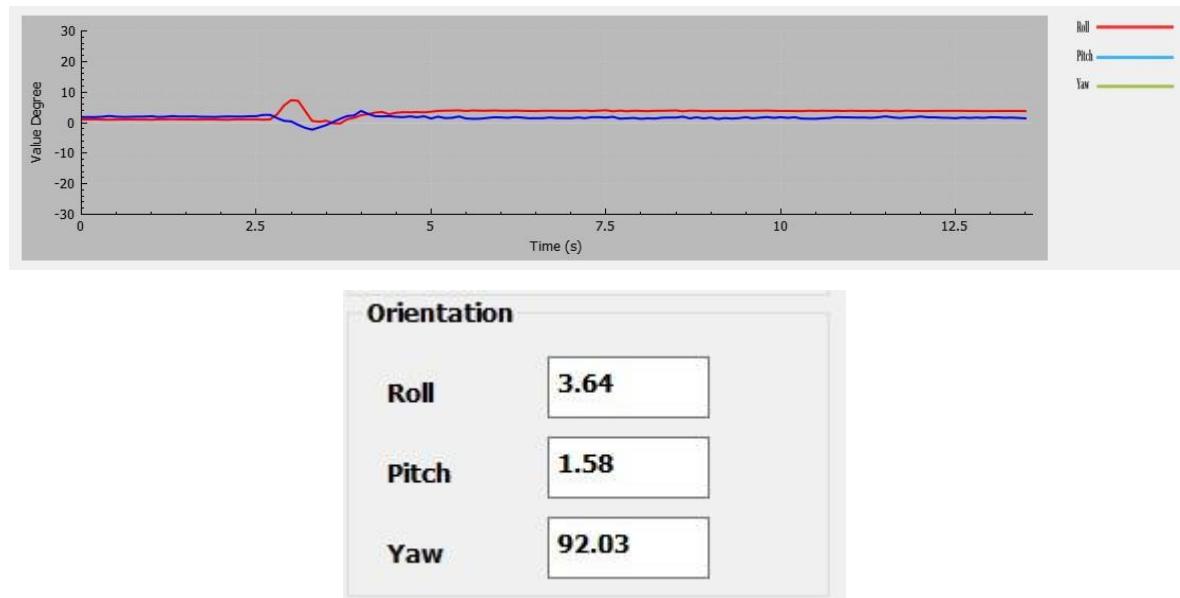
$$\text{Roll} = 10^0$$

$$\text{Pitch} = 0^0$$

Việc điều khiển robot nghiêng theo góc roll sẽ khó hơn do dáng đứng của robot có chân trái và phải không đối xứng.

### **Hiệu chỉnh Kp:**

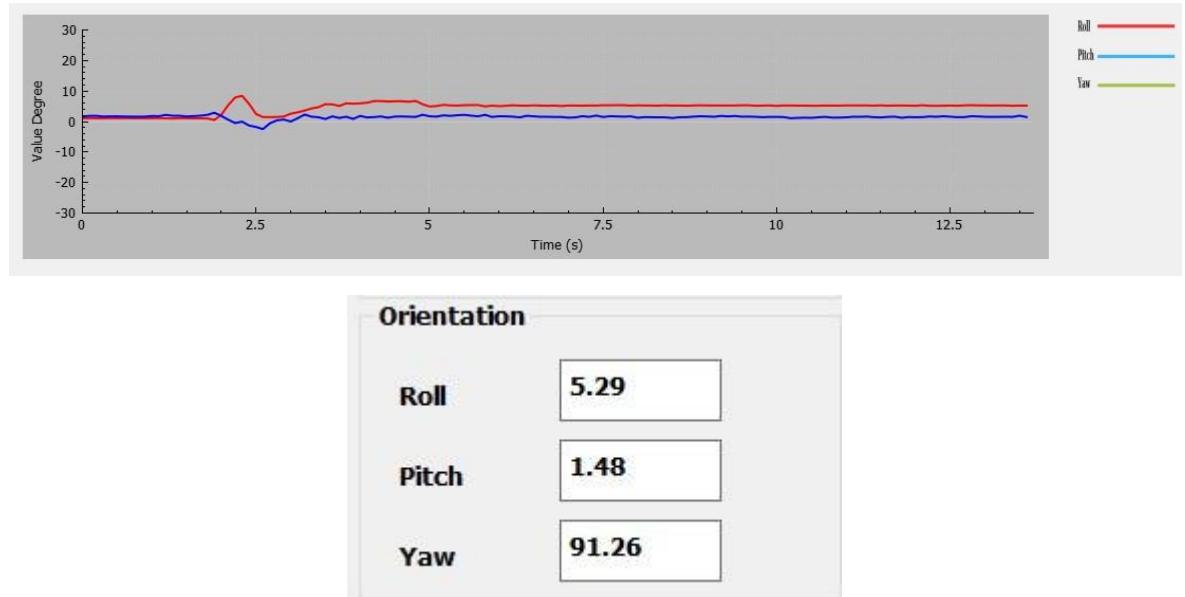
TH1:  $K_p = 0.5$ ,  $K_i = 0$ ,  $K_d = 0$ :



Hình 5.17 Nghiêng robot theo góc roll TH1

$K_p$  nhỏ, giá trị xác lập chỉ lên được khoảng 3.7 độ, sai số xác lập 6.3 độ. Hệ có vọt lố khoảng 5 độ.

TH2:  $K_p = 1$ ,  $K_i = 0$ ,  $K_d = 0$ :



Hình 5.18 Nghiêng robot theo góc roll TH2

Giá trị xác lập vẫn chưa nâng lên được gần với giá trị đặt, sai số xác lập 4.7 độ.

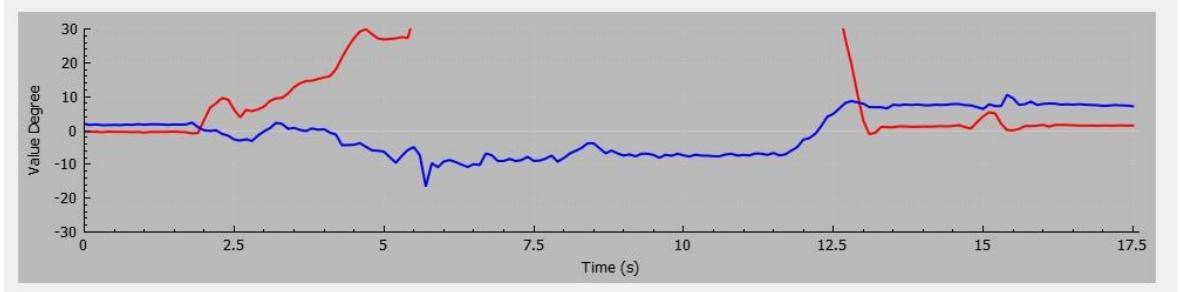
TH3:  $K_p = 2$ ,  $K_i = 0$ ,  $K_d = 0$ :



Hình 5.19 Nghiêng robot theo góc roll TH3

$K_p$  đủ lớn đáp ứng đã có thể bám với giá trị đặt, sai số xác lập khoảng 0.8 độ, thời gian xác lập khoảng 2s.

TH4:  $K_p = 3.5$ ,  $K_i = 0$ ,  $K_d = 0$ :

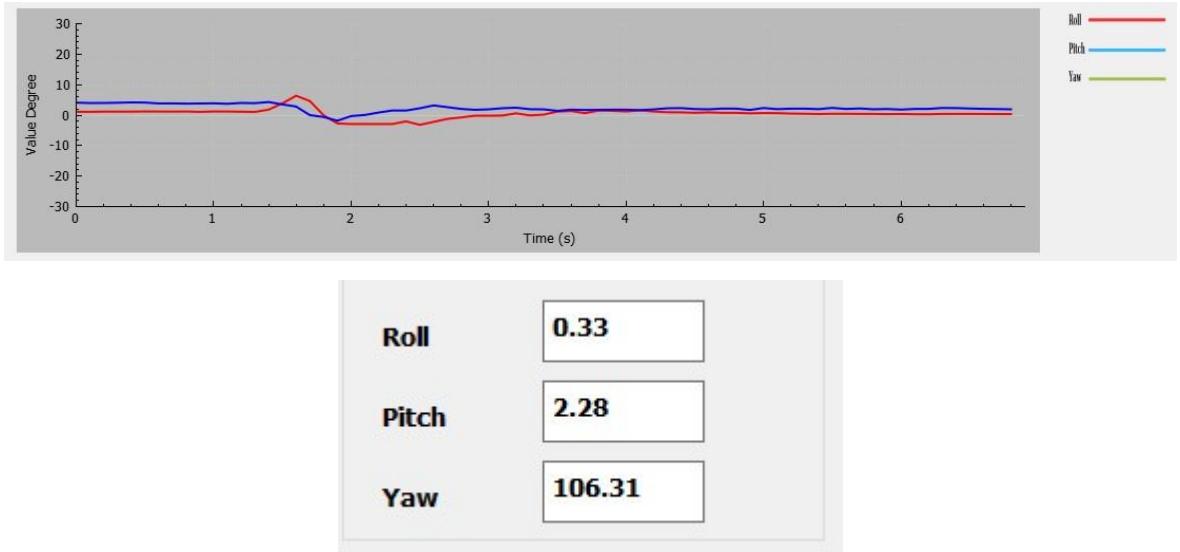


Hình 5.20 Nghiêng robot theo góc roll TH4

$K_p$  quá lớn dẫn đến hệ mất ổn định.

### Hiệu chỉnh PI:

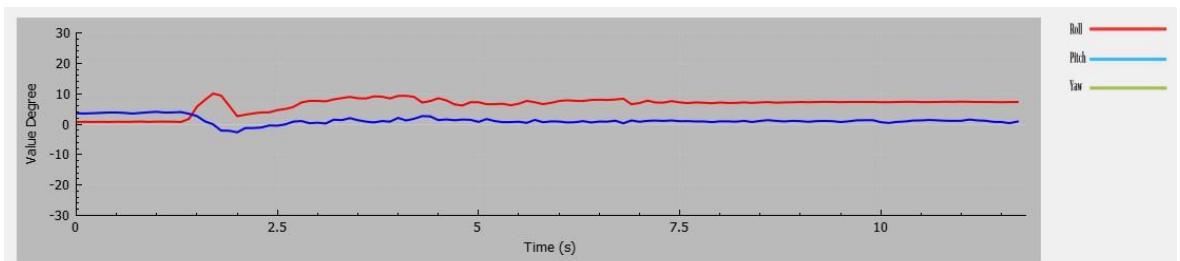
TH5:  $K_p = 0.5$ ,  $K_i = 0.2$ ,  $K_d = 0$ :

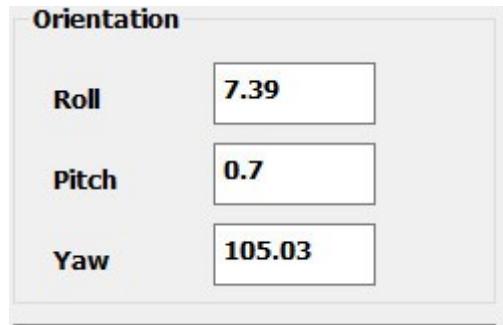


Hình 5.21 Nghiêng robot theo góc roll TH5

Đáp ứng không bám được với giá trị đặt cho  $K_p$  và  $K_i$  quá nhỏ.

TH6:  $K_p = 1$ ,  $K_i = 0.7$ ,  $K_d = 0$ :

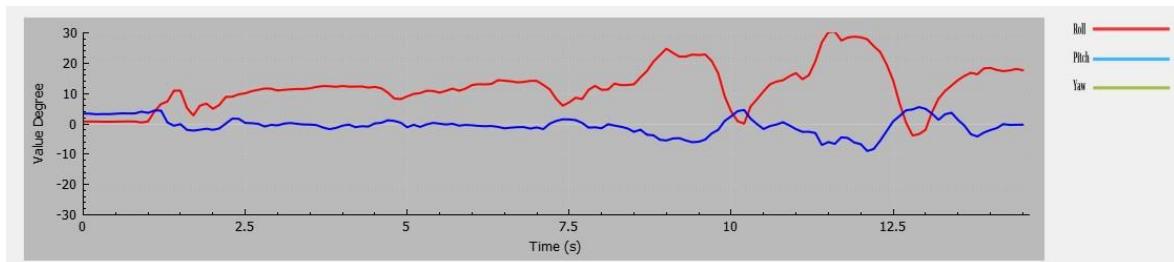




Hình 5.13 Nghiêng robot theo góc roll TH6

Sai số xác lập được cải thiện nhưng vẫn còn lớn, khoảng 2.6 độ

TH7:  $K_p = 1$ ,  $K_i = 1.2$ ,  $K_d = 0$ :

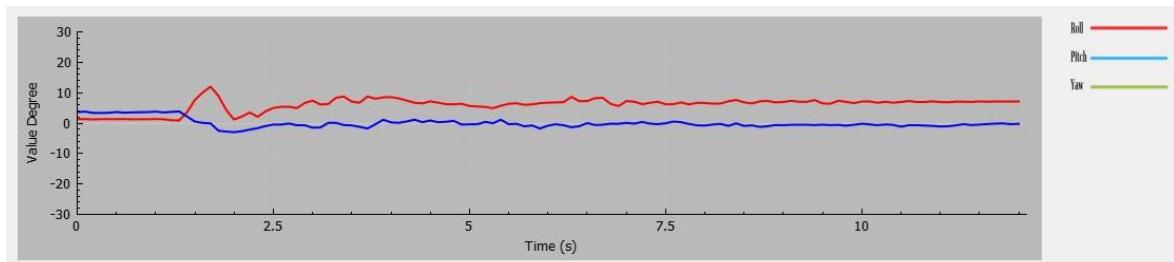


Hình 5.14 Nghiêng robot theo góc roll TH7

Hệ bị mất ổn định và không điều khiển được. Ta thấy đáp ứng khi điều khiển theo góc roll xấu hơn rất nhiều so với điều khiển theo góc pitch.

### Hiệu chỉnh PID:

TH8:  $K_p = 0.7$ ,  $K_i = 0.7$ ,  $K_d = 0.1$ :

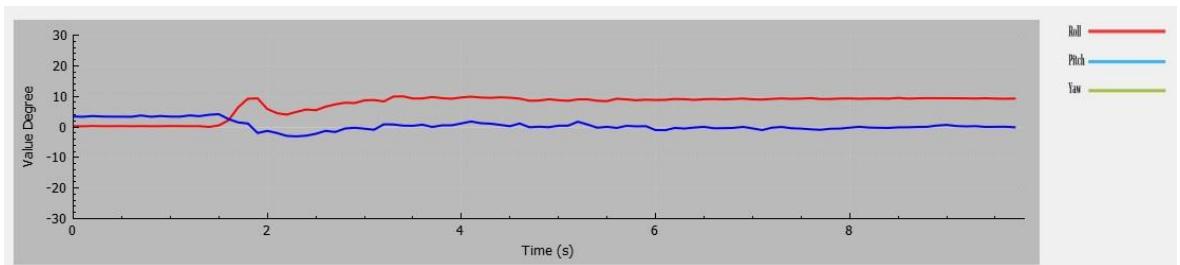


Orientation	
Roll	7.29
Pitch	0.0
Yaw	114.63

Hình 5.14 Nghiêng robot theo góc pitch TH8

Sai số xác lập vẫn còn cao, khoảng 2.7 độ. Độ vọt lô khoảng 6 độ.

TH9:  $K_p = 0.7$ ,  $K_i = 0.7$ ,  $K_d = 0.3$ :

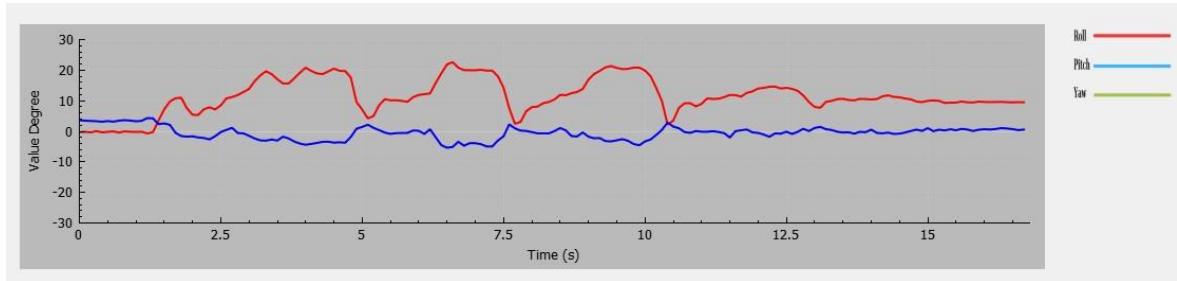


Orientation	
Roll	9.26
Pitch	-0.06
Yaw	112.48

Hình 5.15 Nghiêng robot theo góc roll TH9

Đáp ứng đã tốt hơn trước, thời gian xác lập khoảng 2s và sai số xác lập khoảng 0.7 độ.

TH10:  $K_p = 0.7$ ,  $K_i = 0.7$ ,  $K_d = 0.5$ :



Hình 5.16 Nghiêng robot theo góc roll TH10

Hệ bị mất ổn định.

Kết luận: Bộ điều khiển PID có thể điều khiển cho việc cân bằng thân robot nghiêng theo góc roll. Tuy vậy hệ thống khó đáp ứng và rất dễ mất ổn định vì dáng đứng cố định của robot (bên chân song song và bên chân chéo góc hình chữ 'Y') không tối ưu cho việc nghiêng thân robot về trước hoặc sau.

$K_p$	$K_i$	$K_d$	Đáp ứng
0.5	0	0	Vọt lố 5 độ, sai số xác lập lớn, $e_{xl} = 6.3$
1	0	0	Sai số xác lập cải thiện, $e_{xl} = 4.7$
2	0	0	Bám được với giá trị đặt, $T_{xl} = 2s$ , $e_{xl} = 0.8$
3.5	0	0	Hệ mất ổn định
0.5	0.2	0	Không bám được đến giá trị đặt, sai số xác lập lớn, $e_{xl} = 6.7$
0.5	0.7	0	Sai số xác lập cải thiện, $e_{xl} = 2.6$
1.2	1.2	0	Hệ mất ổn định
0.5	0.5	0.3	Sai số xác lập vẫn còn cao, $e_{xl} = 2.7$ , có vọt lố 6 độ
0.7	0.5	0.3	Đáp ứng tốt với sai số xác lập nhỏ 0.7 độ
0.7	0.7	0.5	Hệ bị mất ổn định

Bảng 5.3 Đáp ứng điều khiển PID theo góc roll

### 5.3.2. Tạo góc nghiêng để robot tự cân bằng



Hình 5.22 Cân bằng thân robot bằng PID

Đặt góc nghiêng chỉ định cho robot:

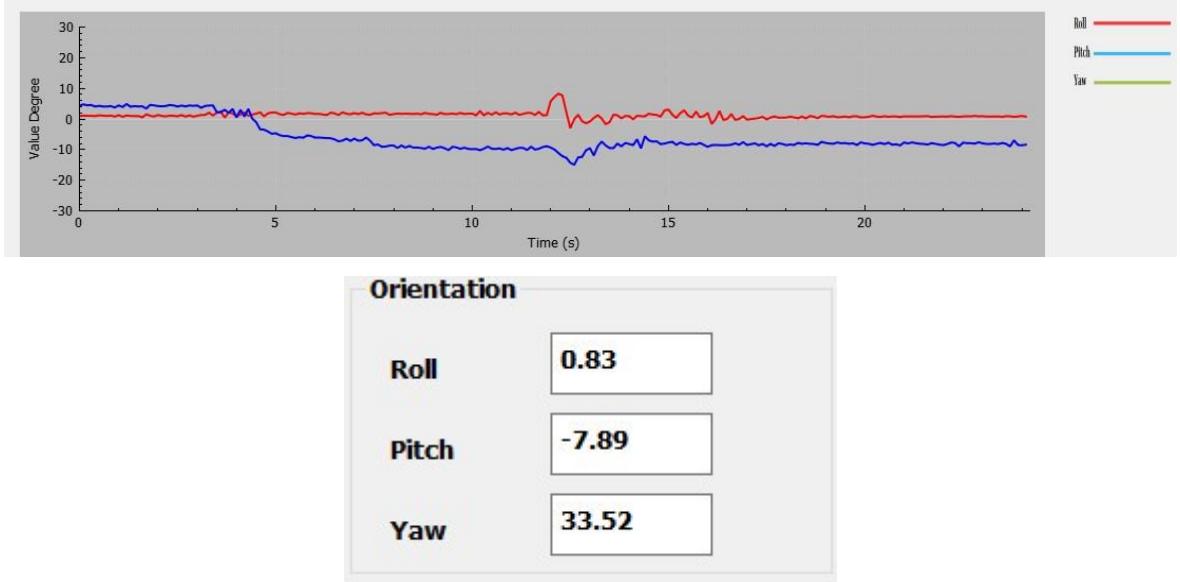
$$\text{Roll} = 0^0$$

$$\text{Pitch} = 0^0$$

Cho robot đứng trên mặt phẳng nghiêng Pitch =  $-10^0$  và hiệu chỉnh thông số PID để robot tự cân bằng:

**Hiệu chỉnh Kp:**

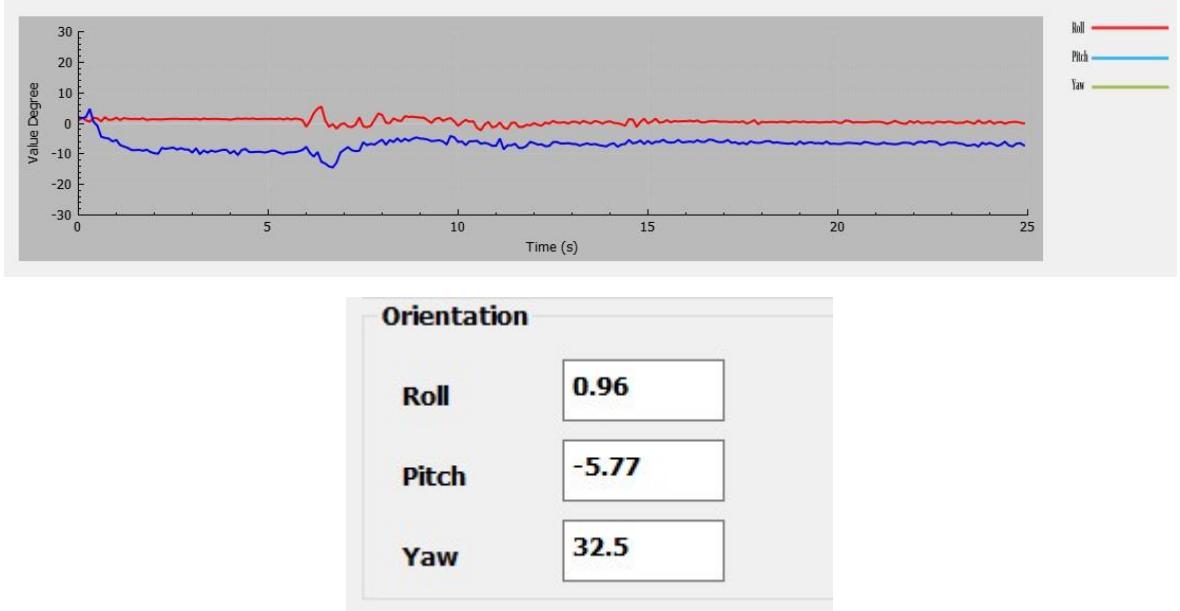
TH1: Kp = 1, Ki = 0, Kd = 0:



Hình 5.23 Robot tự cân bằng trên góc pitch TH1

Do Kp nhỏ, Pitch giá trị xác lập chỉ lên được khoảng -7.89 độ xa so với giá trị đặt, sai số xác lập 7.89 độ. Roll đạt xác lập, vọt lố 10 độ.

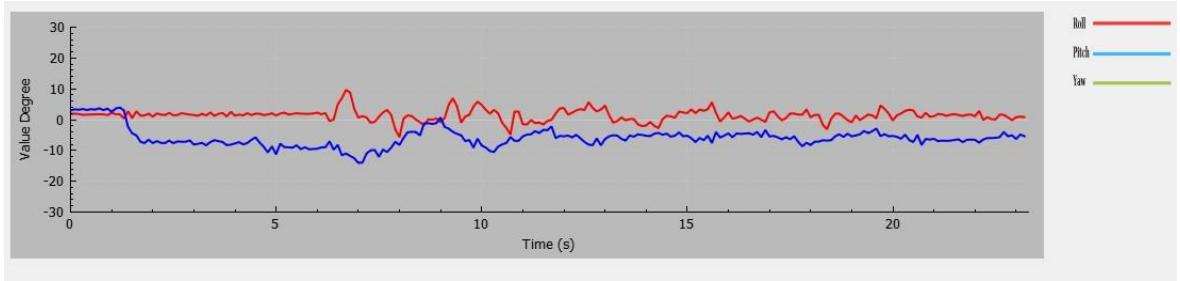
TH2: Kp = 2, Ki = 0, Kd = 0:



Hình 5.24 Robot tự cân bằng trên góc pitch TH2

Giá trị xác lập được cải thiện nhưng vẫn chưa nâng lên được gần với giá trị đặt, sai số xác lập 5.8 độ.

TH3:  $K_p = 3$ ,  $K_i = 0$ ,  $K_d = 0$ :

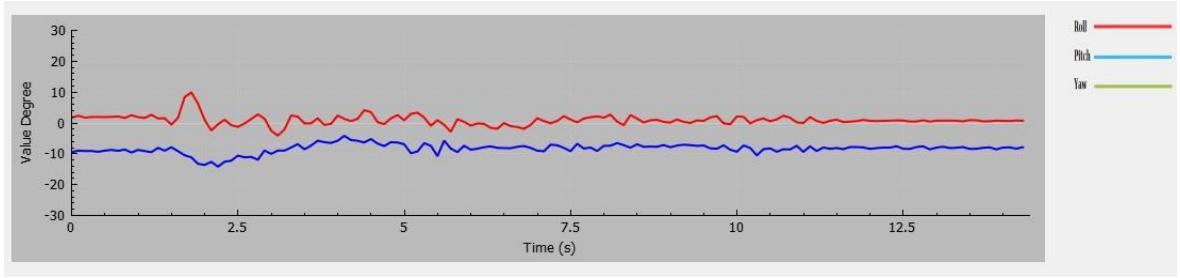


Hình 5.25 Robot tự cân bằng trên góc pitch TH3

Khi  $K_p$  lớn, hệ dẫn dao động mất ổn định.

### Hiệu chỉnh PI:

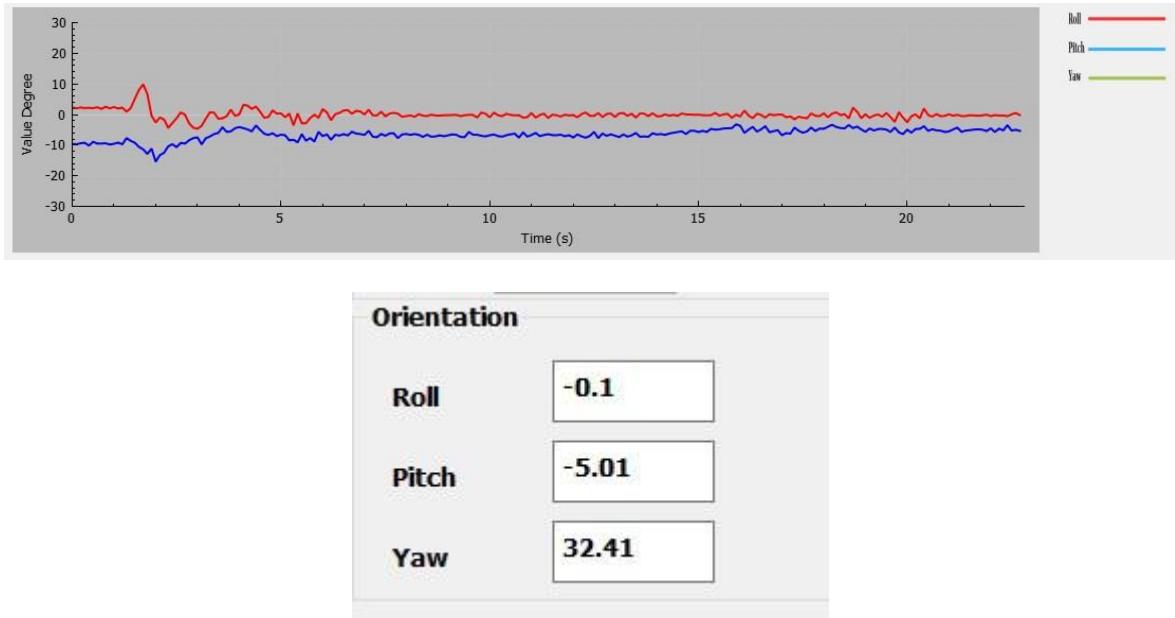
TH4:  $K_p = 1.5$ ,  $K_i = 0.5$ ,  $K_d = 0$ :



Hình 5.26 Robot tự cân bằng trên góc pitch TH4

Góc Pitch vẫn chưa bám được tới giá trị xác lập cho  $K_p$  và  $K_i$  còn nhỏ. Sai số xác lập góc Pitch 8.5 độ. Roll dao động nhiều quanh giá trị đặt.

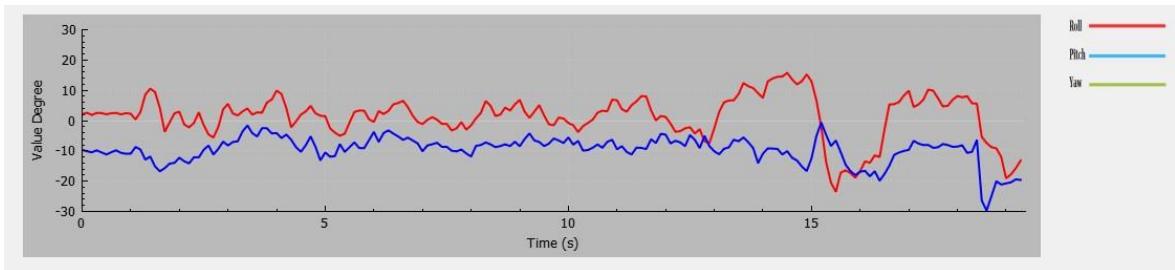
TH5:  $K_p = 1.5$ ,  $K_i = 0.7$ ,  $K_d = 0$ :



Hình 5.12 Robot tự cân bằng trên góc pitch TH5

Sai số xác lập được cải thiện nhưng vẫn còn lớn khoảng 5 độ.

TH6:  $K_p = 2$ ,  $K_i = 1.5$ ,  $K_d = 0$ :

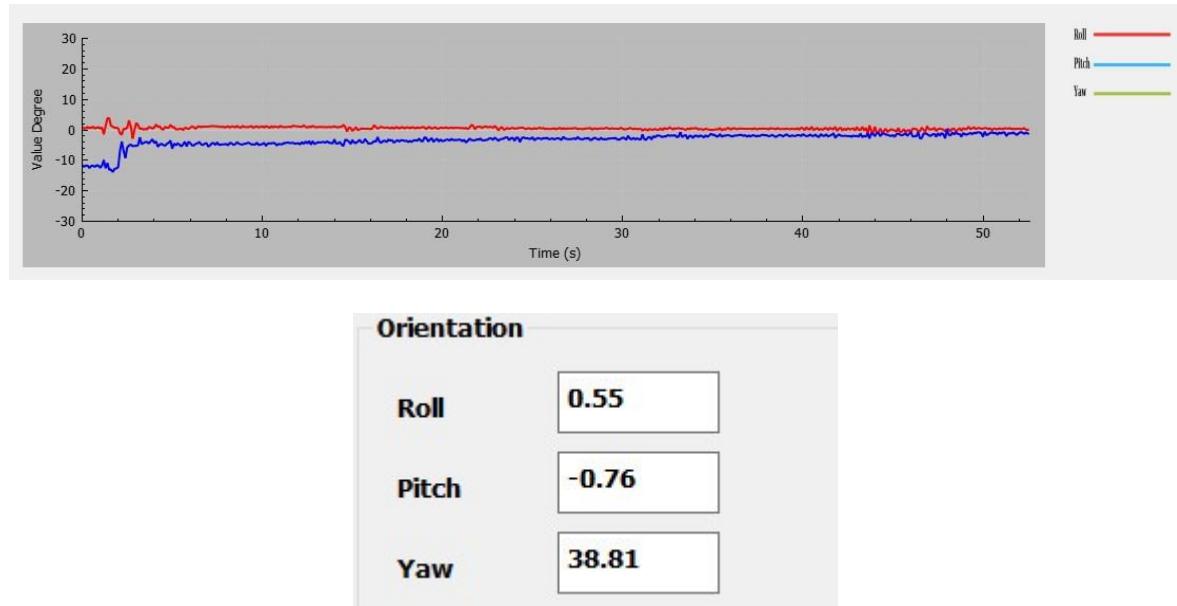


Hình 5.13 Robot tự cân bằng trên góc pitch TH6

Với  $K_p$  và  $K_i$  quá lớn hệ thống định và dao động.

### **Hiệu chỉnh PID:**

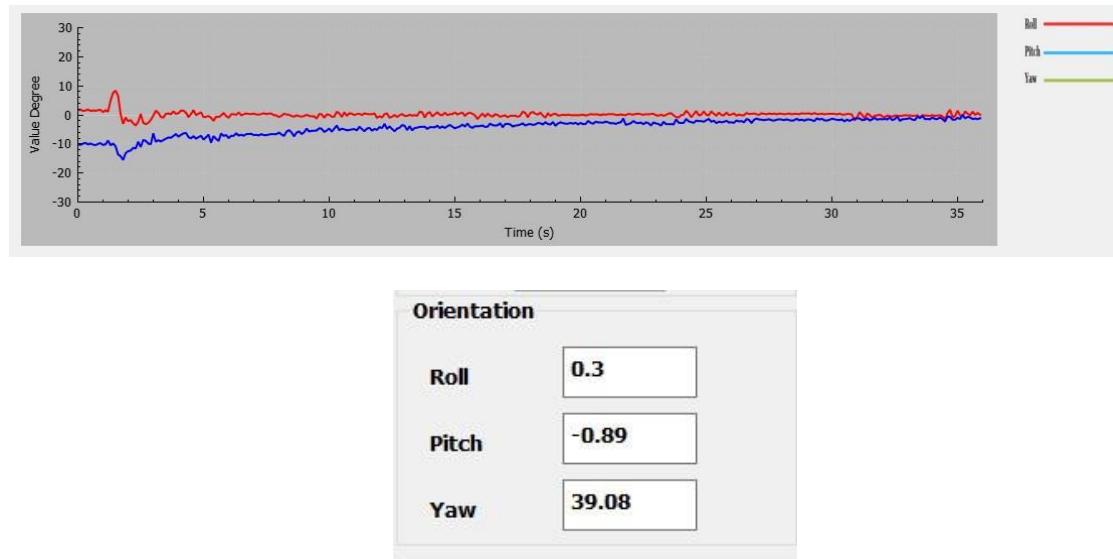
TH7:  $K_p = 0.7$ ,  $K_i = 0.5$ ,  $K_d = 0.1$ :



Hình 5.14 Robot tự cân bằng trên góc pitch TH7

Hệ thống ổn định. Do các hệ số  $K_p$ ,  $K_i$ ,  $K_d$  thời gian đáp ứng khá lâu xấp xỉ 25s nhưng vẫn bám được đến giá trị đặt.

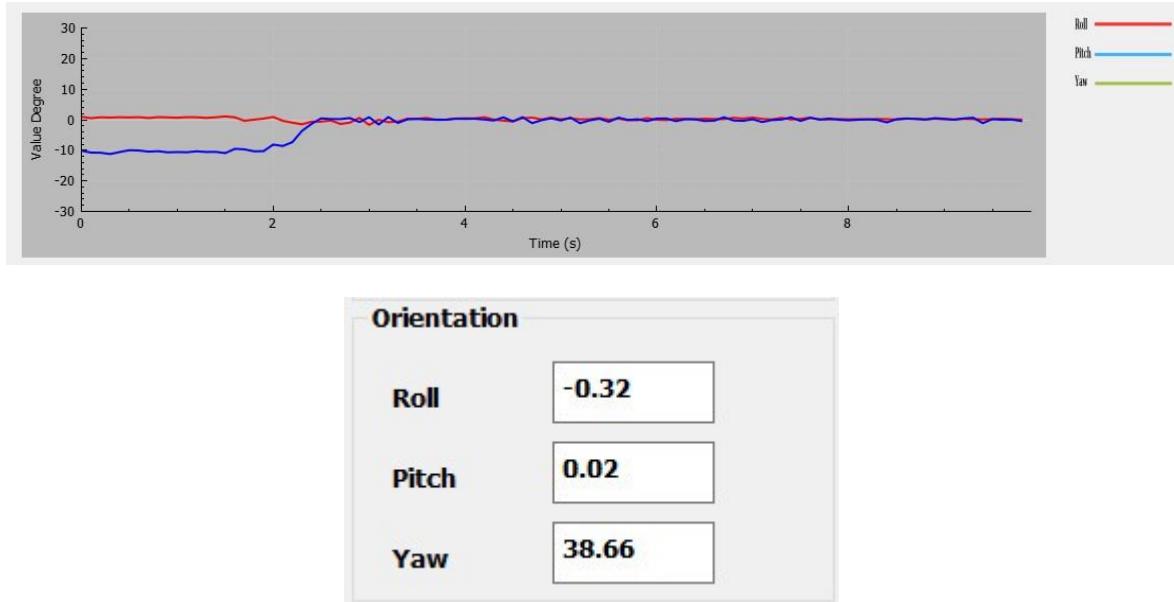
TH8:  $K_p = 0.7$ ,  $K_i = 0.5$ ,  $K_d = 0.3$ :



Hình 5.15 Robot tự cân bằng trên góc pitch TH8

Việc tăng dần các hệ số khiến thời gian đáp ứng được cải thiện. Có xuất hiện vọt lố Roll, Pitch khoảng 9 độ.

TH9:  $K_p = 0.7$ ,  $K_i = 0.5$ ,  $K_d = 0.5$ :



Hình 5.16 Robot tự cân bằng trên góc pitch TH9

Thời gian đáp ứng nhanh, bám sát giá trị đặt và không gây vọt lố

Kết luận: Khi nghiêng góc Pitch, robot tự cân bằng bằng cách điều chỉnh góc nghiêng thân theo hai bên. Dáng đứng robot phù hợp để điều khiển tốt và ổn định việc cân bằng thân robot nghiêng theo góc pitch với biên độ dao động lớn.

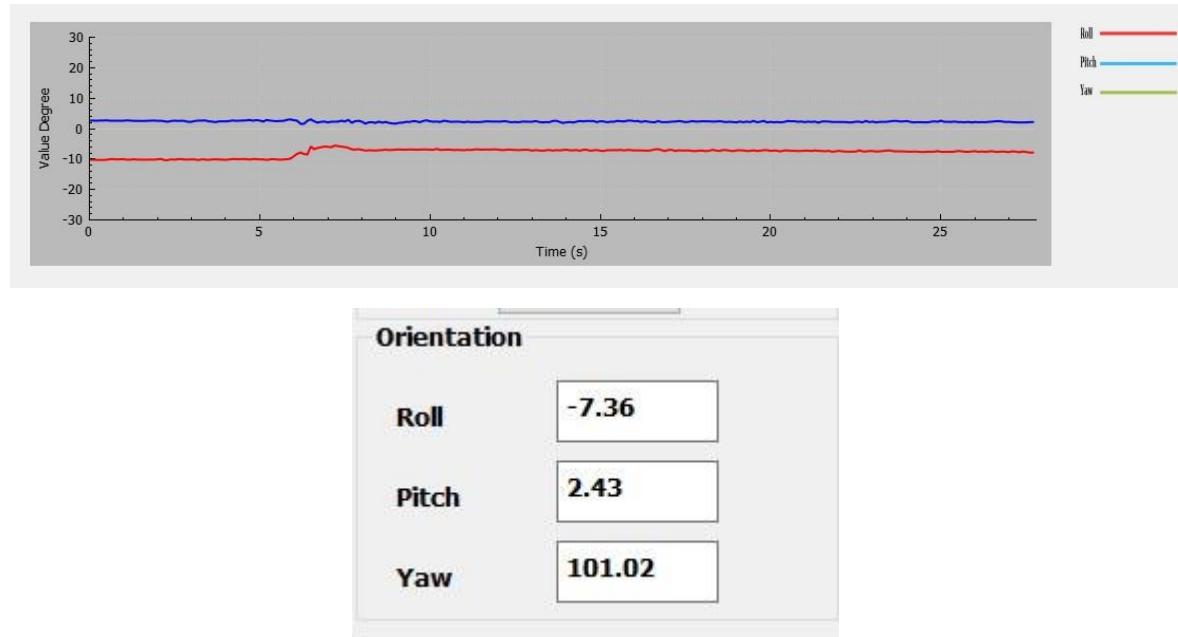
$K_p$	$K_i$	$K_d$	Đáp ứng
1	0	0	Giá trị xác lập xa so với giá trị đặt
2	0	0	Giá trị xác lập cải thiện, vẫn xa so với giá trị đặt
3	0	0	Hệ mát ổn định
1.5	0.5	0	Không bám được đến giá trị đặt
1.5	0.7	0	Giá trị xác lập cải thiện, vẫn xa so với giá trị đặt
2	1.5	0	Hệ mát ổn định
0.7	0.5	0.1	Đáp ứng tốt nhưng thời gian xác lập cao
0.7	0.5	0.3	Đáp ứng tốt cải thiện sai số xác lập, có vọt lố
0.7	0.7	0.5	Hệ thống ổn định, đáp ứng nhanh

Bảng 5.4 Đáp ứng tự cân bằng robot khi nghiêng góc pitch

Cho robot đứng trên mặt phẳng nghiêng  $\text{Roll} = -10^\circ$  và Tuning thông số PID tự cân bằng:

### Hiệu chỉnh Kp:

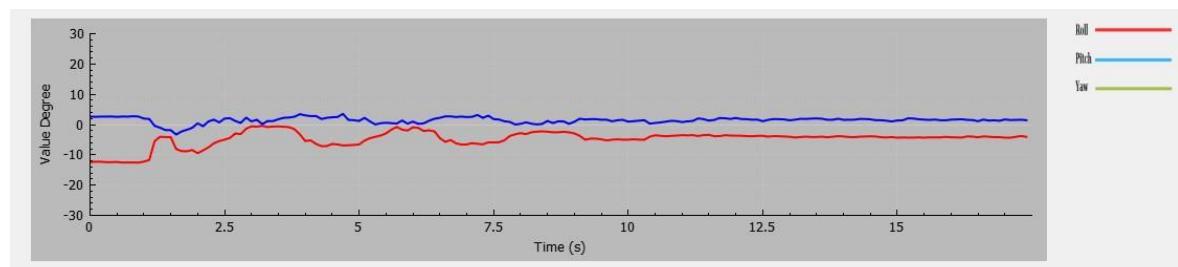
TH1:  $K_p = 1$ ,  $K_i = 0$ ,  $K_d = 0$ :



Hình 5.27 Robot tự cân bằng trên góc roll TH1

$K_p$  nhỏ, đáp ứng không bám tới giá trị đặt, sai số xác lập góc Pitch khoảng  $7.36^\circ$  và góc Roll khoảng  $2.43^\circ$ .

TH2:  $K_p = 2$ ,  $K_i = 0$ ,  $K_d = 0$ :

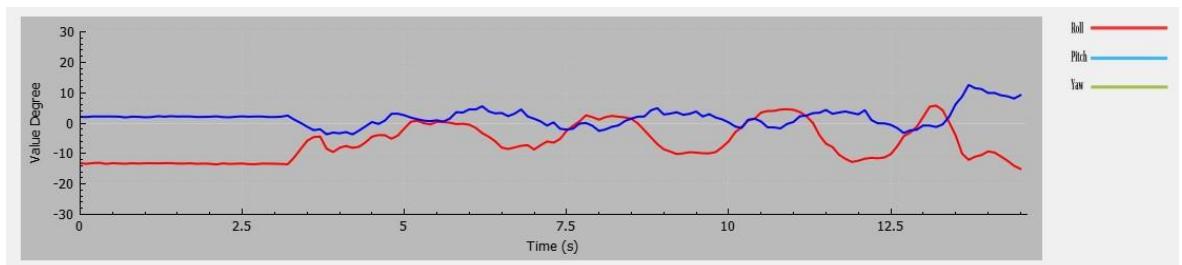


Orientation	
Roll	-4.23
Pitch	1.68
Yaw	101.04

Hình 5.28 Robot tự cân bằng trên góc roll TH2

Cải thiện đáp ứng nhưng vẫn chưa bám được đến giá trị đặt, sai số xác lập 4.3 độ.

TH3:  $K_p = 3$ ,  $K_i = 0$ ,  $K_d = 0$ :

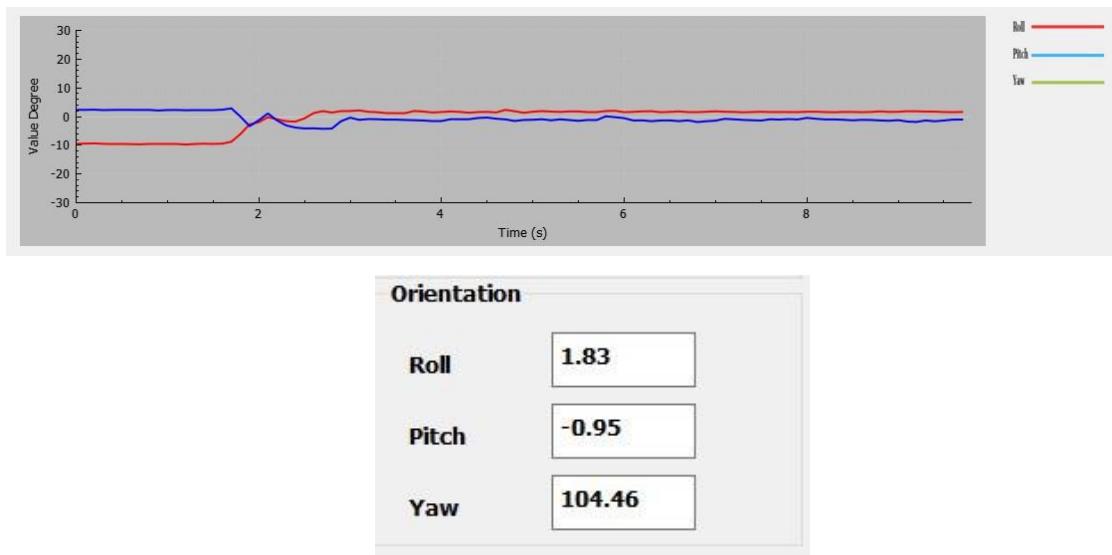


Hình 5.29 Robot tự cân bằng trên góc roll TH3

$K_p$  quá lớn khiến hệ thống mất ổn định.

#### Hiệu chỉnh PI:

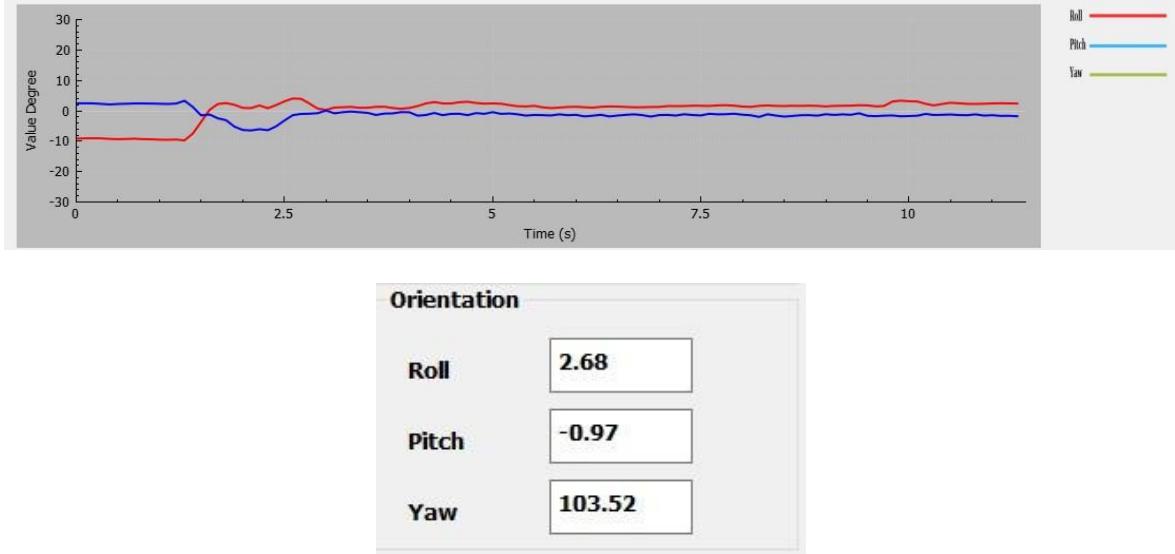
TH4:  $K_p = 1$ ,  $K_i = 0.3$ ,  $K_d = 0$ :



Hình 5.30 Robot tự cân bằng trên góc roll TH4

Chất lượng đáp ứng đã được cải thiện, góc Pitch bám với giá trị đặt, Roll có sai số xác lập thấp khoảng 1.8 độ.

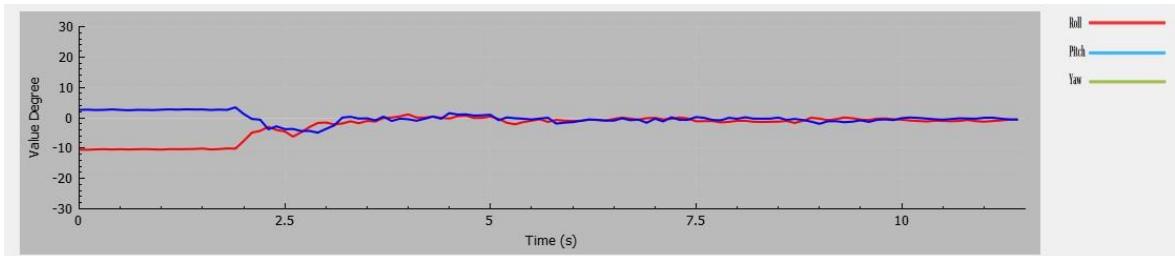
TH5:  $K_p = 1$ ,  $K_i = 0.5$ ,  $K_d = 0$ :



Hình 5.12 Robot tự cân bằng trên góc roll TH5

Sai số xác lập vẫn còn lớn, khoảng 2.7 độ và thời gian xác lập khoảng 2s.

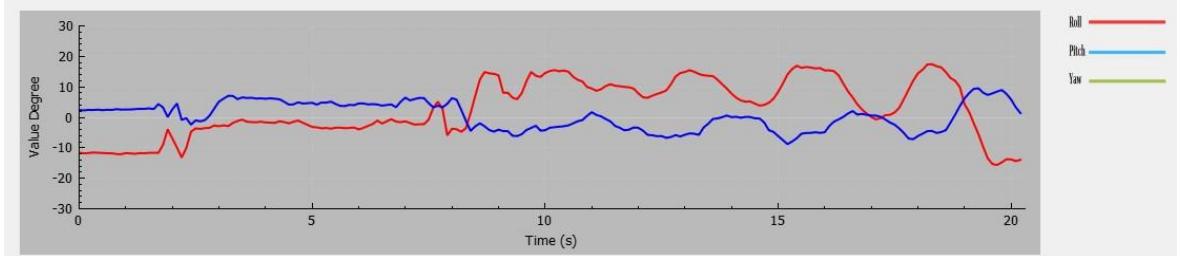
TH6:  $K_p = 1.5$ ,  $K_i = 1$ ,  $K_d = 0$ :



Hình 5.13 Robot tự cân bằng trên góc roll TH6

$K_p$ ,  $K_i$  đủ lớn, hệ thống bám sát với giá trị đặt và thời gian xác lập khoảng 1.8s.

TH7:  $K_p = 2$ ,  $K_i = 1.5$ ,  $K_d = 0$ :

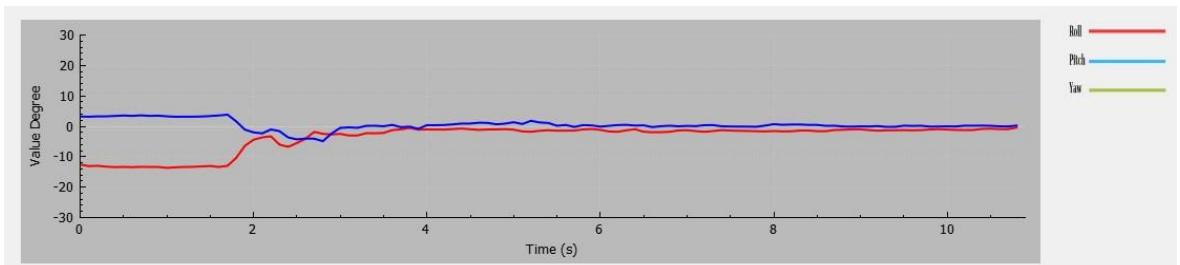


Hình 5.14 Robot tự cân bằng trên góc roll TH7

Khi Kp, Ki quá lớn hệ thống mất ổn định và dao động.

### Hiệu chỉnh PID:

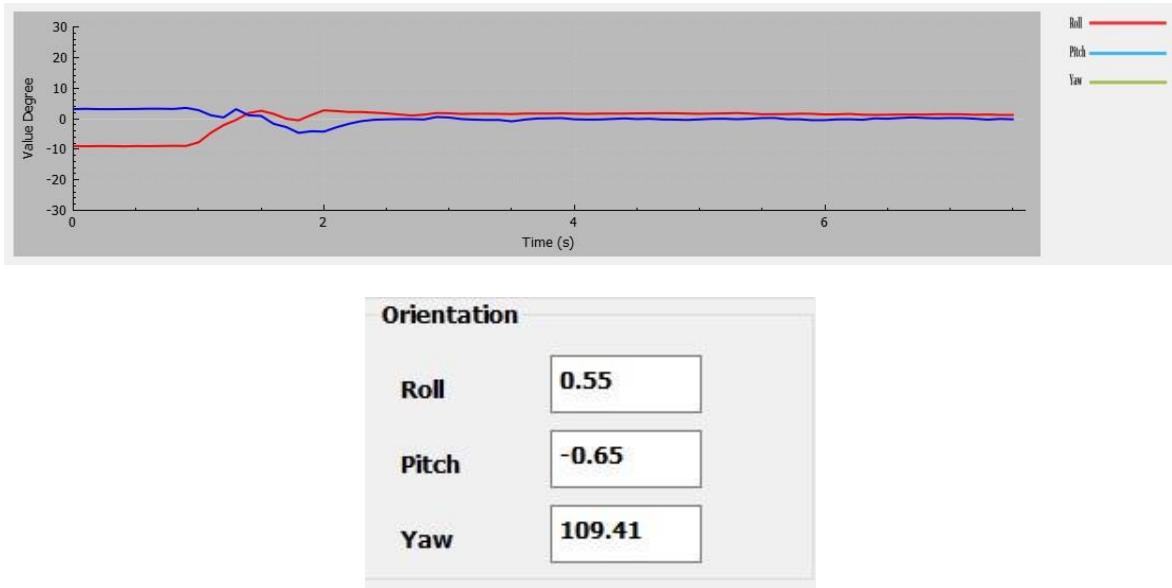
TH8: Kp = 0.7, Ki = 0.3, Kd = 0.3:



Hình 5.14 Robot tự cân bằng trên góc roll TH8

Hệ thống ổn định, Roll Pitch đều bám tốt với giá trị đặt, sai số xác lập khoảng 0.7 và không có vọt lố.

TH9:  $K_p = 0.7$ ,  $K_i = 0.3$ ,  $K_d = 0.5$ :



Hình 5.15 Robot tự cân bằng trên góc roll TH9

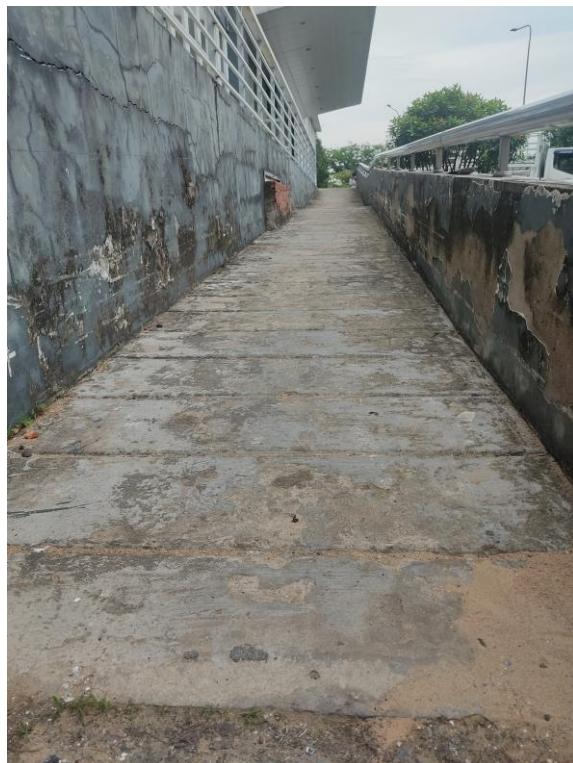
Hệ bám tốt với giá trị đặt, sai số xác lập khoảng 0.6, thời gian xác lập khoảng 2s và không có vọt lố.

Kết luận: Khi nghiêng góc Roll, robot tự cân bằng bằng cách điều chỉnh góc nghiêng thân theo hướng trước sau. Tuy nhiên biên độ dao động của góc Roll lớn hơn góc Pitch vì robot dễ mất thăng bằng trong quá trình tự điều chỉnh theo góc Roll.

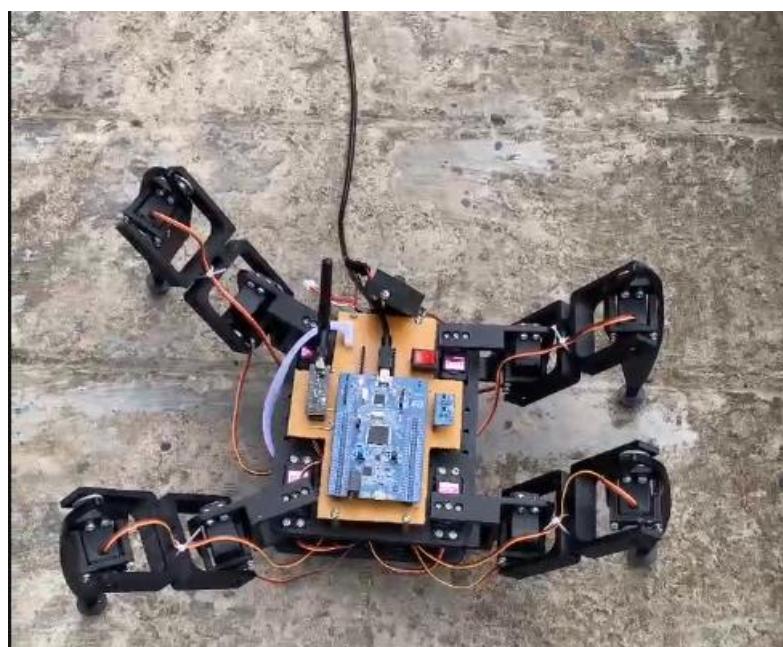
Kp	Ki	Kd	Đáp ứng
1	0	0	Giá trị xác lập xa so với giá trị đặt
2	0	0	Giá trị xác lập cải thiện, mất ổn định giai đoạn đầu
3	0	0	Hệ mất ổn định
1	0.3	0	Bám sát đến giá trị đặt, sai số xác lập thấp
1.5	0.5	0	Sai số xác lập được cải thiện
1.5	1	0	Hệ thống ổn định
2	1.5	0.1	Hệ thống mất ổn định
0.7	0.3	0.3	Hệ thống ổn định
0.7	0.7	0.5	Hệ thống ổn định, đáp ứng nhanh

Bảng 5.5 Đáp ứng tự cân bằng robot khi nghiêng góc roll

#### 5.4. Di chuyển trên bờ mặt dốc

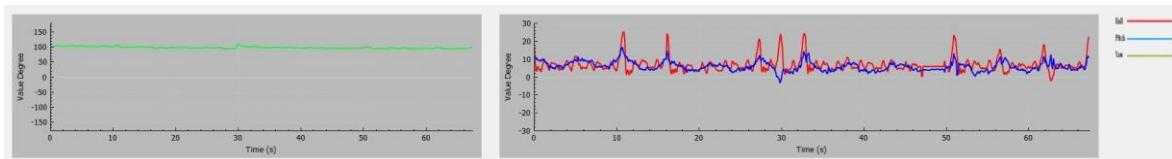


Hình 5.31 Bờ mặt dốc

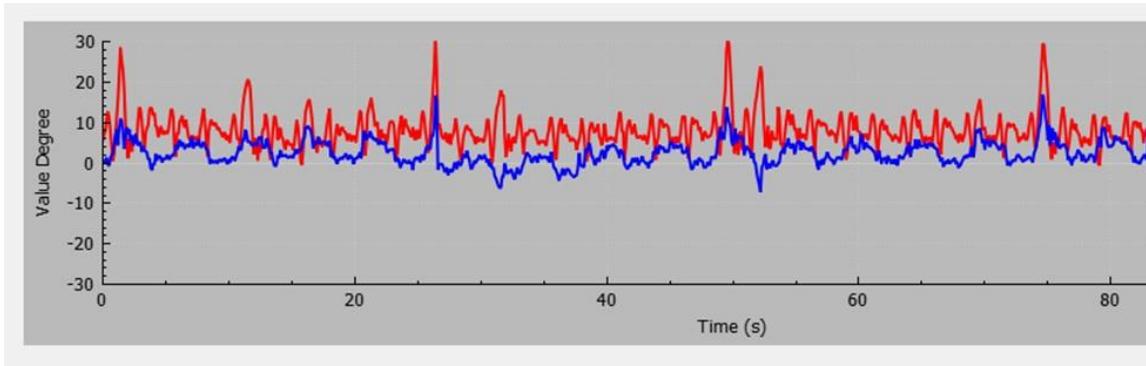


Hình 5.32 Robot di chuyển trên bờ mặt dốc

### TH1: Robot đi lên dốc



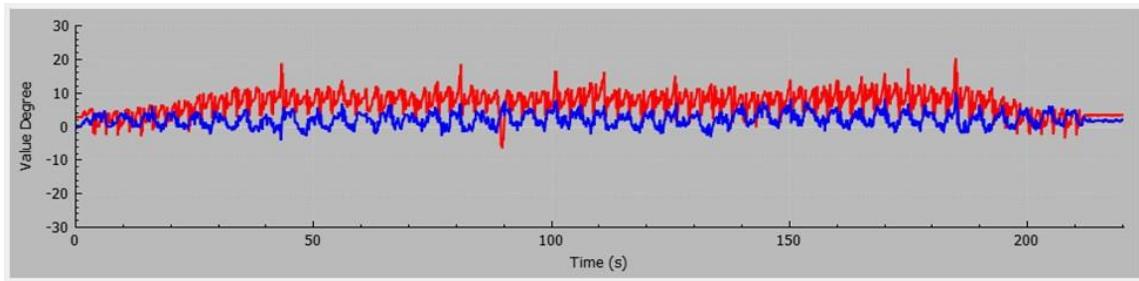
Hình 5.33 Đồ thị góc nghiêng robot khi di chuyển tiến trên bề mặt dốc lần 1



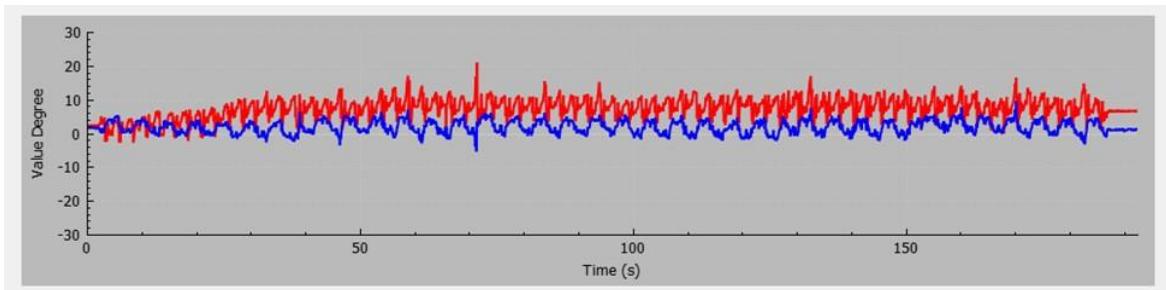
Hình 5.34 Đồ thị góc nghiêng robot khi di chuyển tiến trên bề mặt dốc lần 2

Bề mặt dốc được khảo sát có độ dốc khoảng 10 độ và không bằng phẳng. Khi di chuyển nếu chân robot bước vào những rãnh lõm sẽ bị mất cân bằng và có hiện tượng hơi đổ người về sau, khi này góc roll tăng cao. Tuy nhiên robot vẫn di chuyển được và tiến lên trên bề mặt dốc này. Bởi vì khối lượng thân của robot khá nặng và chưa áp dụng dáng đi phù hợp cho việc di chuyển trên góc nghiêng nên trong quá trình di chuyển, thân robot dao động với biên độ khá lớn, khoảng 10 độ. Việc khảo sát này giúp kết luận được rằng robot vẫn chưa thích nghi tốt với các môi trường di chuyển phức tạp.

## TH2: Robot đi xuống dốc



Hình 5.35 Đồ thị góc nghiêng robot khi di chuyển lùi trên bề mặt dốc lần 1



Hình 5.36 Đồ thị góc nghiêng robot khi di chuyển lùi trên bề mặt dốc lần 2

Khi di chuyển xuống dốc, robot di chuyển ổn định hơn so với khi lên dốc vì trọng tâm robot được dời về sau do góc nghiêng. Khi robot di chuyển không có hiện tượng bị đổ người về trước hay về sau. Thân robot dao động với biên độ thấp, khoảng 6 – 8 độ và khi robot di chuyển từ mặt phẳng nằm ngang sang mặt phẳng dốc, robot di chuyển ổn định, không có hiện tượng bị đổ người về sau khi độ dốc mặt phẳng thay đổi.

## **Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **6.1. Kết luận**

#### **6.1.1. Mục tiêu đạt được**

- Nhóm đã thiết kế lại và phát triển được mô hình robot nhện 4 chân từ mô hình đơn giản có trước. Học thêm được những kiến thức, kỹ năng mới về mảng cơ khí và áp dụng vào đồ án tạo nên mô hình robot hoàn chỉnh bao gồm bộ khung robot kết hợp với board PCB gồm những linh kiện cần thiết để robot có thể hoạt động được.
- Điều khiển Robot có thể thực hiện những chức năng cơ bản như di chuyển tiến lùi, xoay trái, xoay phải và cân bằng thân robot.
- Đảm bảo ước lượng góc và hướng robot bằng IMU với sai số không quá cao, có hệ thống định vị triển khai bằng cách tích hợp dữ liệu từ một camera và cảm biến IMU. Nâng cao độ chính xác và đáng tin cậy của việc định vị.
- Đã phát triển một Giao diện điều khiển không dây trực quan bằng PyQt5 và Qt Designer dùng để điều khiển robot từ xa và đánh giá hiệu suất của robot trong thời gian thực.

#### **6.1.2. Hạn chế cần khắc phục**

- Độ chính xác và độ tin cậy của hệ thống định vị phụ thuộc nhiều vào độ chính xác của cảm biến IMU. Nhóm vẫn chưa ước lượng góc và hướng ổn định hoàn toàn (giá trị đo được vẫn còn nhiều) do những cảm biến này có giá thành thấp. Ngoài ra còn có sai số về cơ khí có thể ảnh hưởng đến độ chính xác tổng thể của hệ thống. IMU dễ bị trôi lệch và sai số bias theo thời gian. Các đo lường gia tốc kế và gyroscope từ IMU có thể tích lũy sai số, dẫn đến sự trôi lệch trong vị trí và hướng. Những lỗi này có thể tăng không giới hạn nếu không được bù đắp đúng cách, ảnh hưởng đến sự ổn định và độ chính xác của hệ thống định vị trong dài hạn.

- Ngoài ra, Camera dễ bị ảnh hưởng bởi các điều kiện ánh sáng thay đổi, che khuất và các loại nhiễu điểm ảnh trong thị giác. Trong môi trường ánh sáng yếu hoặc có sự lóa, khả năng của camera phát hiện thu thập hình ảnh một cách chính xác có thể bị ảnh hưởng.
- Trong Đồ án Tốt nghiệp đã bao gồm các thí nghiệm trên thực tế của robot, nhưng các trường hợp thực nghiệm có thể chưa bao gồm toàn bộ các điều kiện thực tế mà robot có thể gặp phải. Môi trường thực tế thường mang lại những thách thức như địa hình không đồng đều, bề mặt trơn trượt làm robot không thể bám và các vật cản động, có thể ảnh hưởng đáng kể đến hiệu suất của robot. Những yếu tố này có thể mang lại các nhiễu thêm, động lực học chuyển động không đoán trước được mà chưa được giải quyết một cách toàn diện trong Đồ án Tốt nghiệp.

## **6.2. Định hướng phát triển đề tài**

- Cải thiện dáng đi của robot để robot có thể di chuyển mượt mà, linh hoạt. Bằng việc tối ưu hóa cấu trúc và thiết kế của chân hoặc sử dụng bộ điều khiển cho hệ phi tuyến, chúng ta có thể cải thiện khả năng di chuyển và điều khiển của robot thích nghi trên nhiều loại địa hình phức tạp như bề mặt dốc hoặc cầu thang. Tối ưu hóa cơ cấu chân, sử dụng vật liệu và cơ điện tử hiệu quả có thể tăng cường khả năng di chuyển và làm cho robot hoạt động tốt hơn trong môi trường khắc nghiệt.
- Việc tích hợp Camera với chức năng xử lý ảnh và thị giác máy cũng sẽ mở ra cơ hội phát triển những tính năng mới cho robot. Bằng cách này, robot có thể nhận diện và phản ứng với môi trường xung quanh một cách thông minh hơn. Nâng cao trí tuệ nhân tạo trong robot nhện là một phương tiện mạnh mẽ để tự động hóa và học máy. Việc tích hợp các thuật toán trí tuệ nhân tạo giúp robot nhện trở nên linh hoạt và tự động hơn trong các nhiệm vụ.

- Đồng thời, việc nghiên cứu về cách robot nhện tương tác và hợp tác với con người và các robot khác cũng là một hướng đi quan trọng. Bằng nhiều cách như tạo ra giao diện App mobile hoặc tay cầm để điều khiển, ...Khả năng này có thể mở ra nhiều ứng dụng mới trong công việc và cuộc sống hàng ngày.
- Nâng cao hệ thống cảm biến và xây dựng bản đồ môi trường chính xác cũng là một phần quan trọng của việc phát triển robot nhện. Bằng cách này, robot có thể nhận biết và đảm bảo an toàn trong quá trình di chuyển trên môi trường đa dạng.
- Cuối cùng, tối ưu tiêu thụ năng lượng của robot nhện là một yếu tố quan trọng để tăng khả năng triển khai của chúng trong môi trường hạn chế về không gian và nguồn năng lượng.

## TÀI LIỆU THAM KHẢO

- [1] Arash Kalantari and Matthew Spenko. Modeling and performance assessment of the HyTAQ, a hybrid terrestrial/aerial quadrotor. *IEEE Transactions on Robotics*, Vol. 30, No. 5, pp. 1278–1285, 2014.
- [2] T. Khanh Dat, T. Thien Phuc, "A Study on Locomotions of Quadruped Robot", Journal ([www.researchgate.net](http://www.researchgate.net)), pp. 2-6, 2014
- [3] Prases Kumar Mohanty and Priyaranjan Biswal, "Development of quadruped walking robots: A review," *Ain Shams Engineering Journal*, p. <https://doi.org/10.1016/j.asej.2020.11.005>, December 2020.
- [4] Joachim Knälmann, Marcus Saläng, "A study on selfbalancing for a quadruped robot," 2023. , p. 86, 2023-05-10.
- [5] İ. Özdemir, "Quaternion Based AHRS Estimation Using MPU9250 and STM32G431," 8 January 2022. [Online]. Available: <https://ibrahimcahitozdemir.com/2022/01/08/quaternion-based-ahrs-estimation-using-mpu9250-and-stm32g431/>.
- [6] K. Winner, "MPU9250," 2018. [Online]. Available: <https://github.com/kriswiner/MPU9250/tree/master/Documents>.
- [7] Noordin, A, Basri, MAM, and Mohamed, Z %J TELKOMNIKA (2018), "Sensor fusion algorithm by complementary filter for attitude estimation of quadrotor with low-cost IMU". 16(2), pp. 868-875.
- [8] Dipl.-Ing. Karsten Grojekatthöfer, Dr.-Ing. Zizung Yoon, "Introduction into quaternions for spacecraft attitude," Berlin, Germany, May 31, 2012.

- [9] C. Lum, "rotation kinematics quaternions," 12 2023. [Online]. Available: [https://github.com/clum/YouTube/blob/main/FlightMech09/lecture03b\\_rotation\\_kinematics\\_quaternions.pdf](https://github.com/clum/YouTube/blob/main/FlightMech09/lecture03b_rotation_kinematics_quaternions.pdf).
- [10] H. N. Nghia, "Design and Control the quadruped spider robot model," Ho Chi Minh, 2023.
- [11] Radhika Kotecha, Rakesh Singh , "Quadruped Robot Gait Planning for Enhanced," *SSRN Electronic Journal*, 2020.
- [12] A. K. Mishra, "Design, Simulation, Fabrication and Planning of Bio-Spired Quadruped Robot," Anand Kumar , 2014.
- [13] Hutter, Marco, "ANYmal: A Unique Quadruped Robot Conquering Harsh Environments," *Research Features*, vol. (126), pp. 54 - 57, 2018-05-03.
- [14] ALGOT LINDESTAM, DAVID LORANG, "Design and Stability of a Quadruped Robot," STOCKHOLM, SWEDEN, 2021.
- [15] Pablo Gonzalez de Santos, Elena Garcia and Joaquin Estremera, Quadrupedal Locomotion AnIntroduction to theControl of Four-legged Robots, Madrid, Spain, 2006.