**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐIỆN – ĐIỆN TỬ**

****

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI: THIẾT KẾ MÔ HÌNH ROBOT HAI BÁNH**

**TỰ CÂN BẰNG**

**GVHD:** TS. TRẦN ĐỨC THIỆN

**MÔN HỌC**: ĐỒ ÁN 2 ĐIỀU KHIỂN TỰ ĐỘNG

**SINH VIÊN THỰC HIỆN:**

NGUYỄN HOÀNG MINH TRIẾT – 21151176

HUỲNH THỊ KIM TRÂM – 21151486

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 1 – 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT  TP HỒ CHÍ MINH  KHOA ĐIỆN – ĐIỆN TỬ | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập – Tự do – Hạnh phúc |
|  | *Tp. Hồ Chí Minh, Ngày Tháng 1 Năm 2024* |

**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

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

|  |  |  |
| --- | --- | --- |
| **Họ và tên sinh viên:** | Nguyễn Hoàng Minh Triết  Huỳnh Thị Kim Trâm | **MSSV**: 21151176  **MSSV**: 21151486 |
| **Chuyên ngành:** | CNKT Điều khiển và Tự động hóa | **Mã ngành:** 151 |
| **Hệ đào tạo:** | Đại học chính quy | **Mã hệ:** |
| **Khóa:** | 2021 | **Lớp:** 21151CL3A |

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

II. NHIỆM VỤ VÀ NỘI DUNG:

|  |
| --- |
| **GIẢNG VIÊN HƯỚNG DẪN** |
| **TS. TRẦN ĐỨC THIỆN** |

**LỜI CẢM ƠN**

Đầu tiên, nhóm thực hiện xin gửi lời cảm ơn chân thành đến Trường Đại học Sư phạm Kỹ thuật TP. HCM và khoa Điện – Điện tử đã cho chúng em có cơ hội vận dụng kiến thức đã học khi đưa môn học *Đồ án 2 Điều khiển tự động* vào chương trình giảng dạy. Đặc biệt, chúng em vô cùng tri ân sự hướng dẫn tận tình và theo dõi sát sao đầy tinh thần trách nhiệm và sự yêu thương trong suốt thời gian thực hiện bài báo cáo đến giảng viên hướng dẫn – Thầy Trần Đức Thiện. Nhóm thực hiện xin chân thành cảm ơn sự hỗ trợ tận tình hỗ trợ trong việc cung cấp kiến thức để hoàn thành tốt bài báo cáo, sự nhiệt tình trong quá trình giảng dạy của thầy đã giúp cho chúng em đạt những kiến thức bổ ích trong suốt thời gian vừa qua. Sau khi tham gia môn học *Đồ án 2 Điều khiển tự động* của thầy, chúng em đã tích lũy cho bản thân nhiều kiến thức bổ ích, tinh thần học tập nghiêm túc. Đây chắc chắn sẽ là hành trang quý báu để nhóm có thể vững bước sau này.

*Đồ án 2 Điều khiển tự động* là môn học thú vị và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Trong quá trình thực hiện bài báo cáo, nhận thấy mình đã cố gắng hết sức nhưng vì kiến thức vẫn còn hạn hẹp nên bài báo cáo vẫn còn nhiều điều thiết sót, chưa chính xác, kính mong thầy xem xét và góp ý để bài báo cáo của chúng em được hoàn thiện hơn.

Sau cùng nhóm thực hiện xin chúc Thầy sức khoẻ, thành công và tiếp tục đào tạo những sinh viên giỏi đóng góp cho đất nước.

Nhóm thực hiện xin chân thành cảm ơn!

|  |
| --- |
| **Trân trọng**  **Nhóm thực hiện báo cáo** |
| Nguyễn Hoàng Minh Triết  Huỳnh Thị Kim Trâm |

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

|  |
| --- |
| *TP. Hồ Chí Minh, Ngày Tháng 1 Năm 2024* |
| **GIẢNG VIÊN HƯỚNG DẪN** |
| **TS. TRẦN ĐỨC THIỆN** |

**MỤC LỤC**

NỘI DUNG

[CHƯƠNG 1 TỔNG QUAN 9](#_Toc155215037)

[1.1 Đặt vấn đề 9](#_Toc155215038)

[1.1.1 Thế nào là xe hai bánh tự cân bằng? 9](#_Toc155215039)

[1.1.2 Tại sao phải thiết kế xe hai bánh tự cân bằng? 9](#_Toc155215040)

[1.2 Mô tả đề tài 11](#_Toc155215041)

[1.3 Mục tiêu nghiên cứu 11](#_Toc155215042)

[1.4 Phần cứng sử dụng 12](#_Toc155215043)

[1.5 Nội dung nghiên cứu 12](#_Toc155215044)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT CỦA XE HAI BÁNH TỰ CÂN BẰNG 13](#_Toc155215045)

[2.1 Nguyên lý cân bằng robot 13](#_Toc155215046)

[2.2 Mô hình con lắc ngược 14](#_Toc155215047)

[2.3 Đặc tính động lực học 16](#_Toc155215048)

[2.4 Điều khiển toàn phương tuyến tính 20](#_Toc155215049)

[2.4.1 Bài toán điều khiển toàn phương tuyến tính liên tục 20](#_Toc155215050)

[2.4.2 Thuật toán LQR 21](#_Toc155215051)

[2.4.3 Tuyến tính hệ thống 22](#_Toc155215052)

[2.5 Bộ lọc Kalman 24](#_Toc155215053)

[2.5.1 Phương trình trạng thái 25](#_Toc155215054)

[2.5.2 Phương trình đo 25](#_Toc155215055)

[2.5.3 Phương trình bộ lọc Kalman 26](#_Toc155215056)

[2.6 Sơ đồ mô phỏng hệ thống trên Matlab Simulink 26](#_Toc155215057)

[CHƯƠNG 3 THIẾT KẾ PHẦN CỨNG VÀ THI CÔNG HỆ THỐNG 31](#_Toc155215058)

[3.1 Yêu cầu thiết kế 31](#_Toc155215059)

[3.2 Sơ đồ khối hệ thống 31](#_Toc155215060)

[3.3 Lựa chọn thiết bị phần cứng 32](#_Toc155215061)

[3.3.1 Khối cấp nguồn 32](#_Toc155215062)

[3.3.2 Khối động cơ và cảm biến đọc xung của động cơ 33](#_Toc155215063)

[3.3.3 Khối điều khiển động cơ 34](#_Toc155215064)

[3.3.4 Khối điều khiển trung tâm 36](#_Toc155215065)

[3.3.5 Khối cảm biến đo góc nghiêng 37](#_Toc155215066)

[3.3.6 Khối thu phát tín hiệu 39](#_Toc155215067)

[3.4 Sơ đồ nối dây 39](#_Toc155215068)

[3.5 Xây dựng mô hình 3D trên môi trường Inventor 40](#_Toc155215069)

[CHƯƠNG 4 THIẾT KẾ PHẦN MỀM 42](#_Toc155215070)

[4.1 Yêu cầu thiết kế 42](#_Toc155215071)

[4.2 Xây dựng thuật toán điều khiển 42](#_Toc155215072)

[4.2.1 Thuật toán chương trình chính 42](#_Toc155215073)

[4.2.2 Thuật toán xử lý xung trả về từ Encoder và tính toán giá trị góc tới, góc xoay 44](#_Toc155215074)

[4.2.3 Thuật toán đọc và xử lí giá trị từ cảm biếng góc nghiêng 46](#_Toc155215075)

[4.2.4 Thuật toán tính xung băm PWM mỗi bánh với giải thuật LQR và điều khiển động cơ 48](#_Toc155215076)

[4.3 Thiết kế GUI 50](#_Toc155215077)

[4.3.1 Thiết kế giao diện giám sát 50](#_Toc155215078)

[4.3.2 Kết quả giao diện 50](#_Toc155215079)

[CHƯƠNG 5 KẾT QUẢ THỰC HIỆN 51](#_Toc155215080)

[5.1 Kết quả thực tế 51](#_Toc155215081)

[5.2 Kết luận 54](#_Toc155215082)

[5.3 Hướng phát triển của đề tài 54](#_Toc155215083)

[TÀI LIỆU THAM KHẢO 56](#_Toc155215084)

**DANH SÁCH HÌNH ẢNH**

[Hình 1 Nguyên lý cân bằng robot 9](#_Toc155225149)

[Hình 2 Mô hình robot 3 bánh trên các dạng địa hình 10](#_Toc155225150)

[Hình 3 Mô hình robot 2 bánh trên địa hình nghiêng 11](#_Toc155225151)

[Hình 4 Nguyên lý hoạt động của hệ xe cân bằng 13](#_Toc155225152)

[Hình 5 Sơ đồ khối tín hiệu của hệ con lắc ngược 15](#_Toc155225153)

[Hình 6 Mô hình động học hệ xe cân bằng 16](#_Toc155225154)

[Hình 7 Sơ đồ khối bộ điều khiển toàn phương tuyến tính LQR 21](#_Toc155225155)

[Hình 8 Mô tả hoạt động của bộ lọc Kalman 25](#_Toc155225156)

[Hình 9 Mô hình xe cân bằng mô phỏng trong Matlab Simulink 28](#_Toc155225157)

[Hình 10 Đáp ứng ngõ ra của hệ thống 30](#_Toc155225158)

[Hình 11 Pin Lithium-ion (Li-Ion) 18650 32](#_Toc155225159)

[Hình 12 Mạch hạ áp DC-DC LM2596 33](#_Toc155225160)

[Hình 13 Động cơ hộp số JGB37-520 12V333RPM 34](#_Toc155225161)

[Hình 14 Mạch cầu H điều khiển động cơ DC L298N 35](#_Toc155225162)

[Hình 15 Vi điều khiển Arduino Mega2560 37](#_Toc155225163)

[Hình 16 Cảm biến góc nghiêng MPU6050 38](#_Toc155225164)

[Hình 17 Module Bluetooth HC06 39](#_Toc155225165)

[Hình 18 Sơ đồ nối dây của mô hình 40](#_Toc155225166)

[Hình 19 Thông số mô hình 40](#_Toc155225167)

[Hình 20 Mô hình 3D robot 2 bánh tự cân bằng 41](#_Toc155225168)

[Hình 21 Lưu đồ thuật toán chương trình chính 43](#_Toc155225169)

[Hình 22 Lưu đồ chương trình ngắt xử lý xung trả về từ Encoder 45](#_Toc155225170)

[Hình 23 Lưu đồ tính toán giá trị góc tới và góc xoay 45](#_Toc155225171)

[Hình 24 Lưu đồ thuật toán đọc và xử lý giá trị từ cảm biến góc nghiêng 47](#_Toc155225172)

[Hình 25 Lưu đồ thuật toán tính xung băm PWM mỗi bánh với giải thuật LQR 48](#_Toc155225173)

[Hình 26 Lưu đồ thuật toán điều khiển động cơ 49](#_Toc155225174)

[Hình 27 Giao diện máy tính giao tiếp với mô hình 50](#_Toc155225175)

[Hình 28 Mô hình phần cứng thi công thực tế hướng trực diện 51](#_Toc155225176)

[Hình 29 Mô hình phần cứng thi công thực tế hướng nằm ngang 52](#_Toc155225177)

[Hình 30 Mô hình phần cứng thi công thực tế hướng nhìn từ trên xuống 53](#_Toc155225178)

[Hình 31 Kết quả thu được từ đồ thị 54](#_Toc155225179)

**LỜI MỞ ĐẦU**

Được nghiên cứu khá nhiều trong các phòng thí nghiệm liên quan đến lĩnh vực điều khiển tự động. Hệ mô hình robot hai bánh tự động cân bằng là một hệ MIMO (nhiều tín hiệu vào và nhiều tín hiệu ra) mang tính chất lý thuyết và thực tiễn cao cũng như đã được ứng dụng nhiều vào thực tiễn.

Robot hai bánh tự cân bằng đã cho thấy nhiều ứng dụng trong cuộc sống và trở thành đối tượng của các nhà nghiên cứu, các kỹ sư và những người đam mê robot. Robot hai bánh tự cân bằng được mô hình như một hệ con lắc ngược di động. Robot được cấu tạo gồm phần thân robot và hai bánh xe đồng trục truyền động bằng hai động cơ DC riêng lẻ. Để giữ thăng bằng cho robot, một bộ xử lý đọc các tín hiệu cảm biến, thông qua quá trình tính toán, tín hiệu điều khiển được gởi đến hai động cơ để điều khiển robot di chuyển về sau hay trước sao cho trọng lượng của robot luôn đặt trên trọng tâm của hai bánh xe. Vấn đề cần quan tâm là bài toán điều khiển cân bằng cho robot khi di chuyển trên các địa hình khác nhau. Đã có nhiều giải thuật điều khiển đã được công bố và tiến hành thực nghiệm như: Điều khiển PID, điều khiển bằng phương pháp Fuzzy Logic, mạng nơron nhân tạo,...

Qua so sánh và phân tích các ưu nhược điểm của một số bộ điều khiển có thể sử dụng cho hệ xe hai bánh tự cân bằng, nhóm đã thực hiện nghiên cứu đề tài “Hệ thống xe hai bánh tự cân bằng” thông qua sử dụng giải thuật LQR (Linear Quadratic Regulator) và mô phỏng hệ thống trên phần mềm hỗ trợ Matlab để có thể đưa ra các bộ trọng số làm tối ưu hóa hoạt động của mô hình.

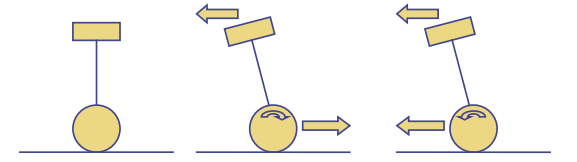
**CHƯƠNG 1 TỔNG QUAN**

* 1. **Đặt vấn đề**
     1. ***Thế nào là xe hai bánh tự cân bằng?***

Đối với các loại xe ba, bốn bánh,… việc giữ thăng bằng và ổn định xe là nhờ vào trọng tâm của chúng nằm trong bề mặt chân đế do bánh xe tạo ra. Đối với xe 2 bánh việc thăng bằng trở nên tương đối khó khi di chuyển chậm và rất khó khi dứng yên do tính chất của việc thăng bằng của xe dựa trên tính chất con quay hồi chuyển ở bánh xe khi đang quay. Đối với xe hai bánh tự cân bằng trọng tâm của xe cần được giữ nằm ngay giữa các bánh xe giống như các vận động viên xiếc giữ yên cây gậy thẳng đứng trong lòng bàn tay.

Trên thực tế, trọng tâm của toàn bộ xe cân bằng không được biết nằm ở vị trí nào, không có cách để tìm ra và có thể không có khả năng di chuyển xe đủ nhanh để giữ nó luôn ở dưới toàn bộ trọng tâm.

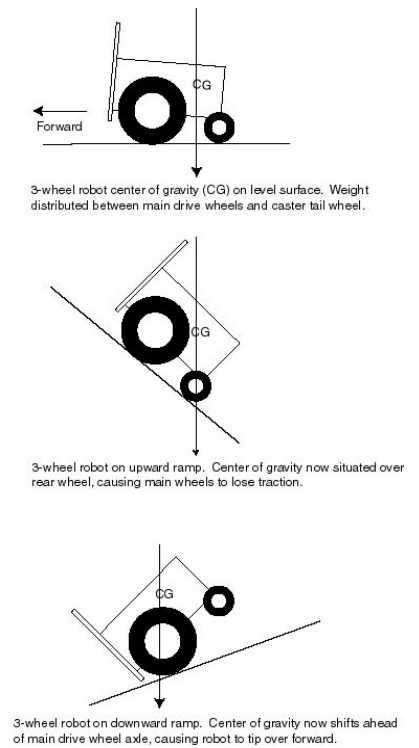
Xét về góc độ kỹ thuật, góc giữ sàn và chiều trọng lược có thể tính toán đuộc. Vì thế thay vì xác định trọng tâm của xe điều vốn dĩ khó thực hiện, chỉ cần giữ xe thẳng dứng, vông góc với sàn xe (góc cân bằng là zero).



Hình 1 Nguyên lý cân bằng robot

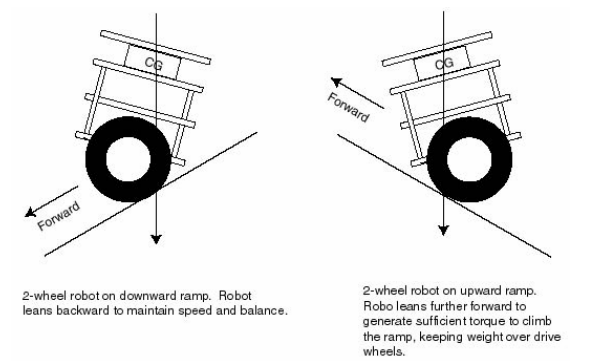
* + 1. ***Tại sao phải thiết kế xe hai bánh tự cân bằng?***

Những mobile robot xây dựng hầu hết là robot di chuyển bằng bánh xe. Với 2 bánh láy được lắp đồng trục và một bánh đuôi nhỏ. Còn đối vói xe 4 bánh thường 1 đầu xe có 2 bánh truyền động và 2 bánh còn lại là bánh lái. Việc thiết kế 3 hay 4 bánh làm cho xe/mobile được thăng bằng ổn định nhờ trọng lượng. Nếu trọng lượng đặt nhiều vào bánh lái thì xe không ổn định và dễ bị ngã; nếu đặt vào bánh đuôi thì 2 bánh lái sẽ mất khả năng bám. Thiết kế 3 hay 4 bánh xe có thể di chuyển tốt trên địa hình phẳng nhưng khó di chuyển trên địa hình lồi lõm.



Hình 2 Mô hình robot 3 bánh trên các dạng địa hình

Ngược lại, xe 2 bánh đồng trục lại thăng bằng linh động trên nhiều địa hình phức tạp dù bản chất của nó là một hệ thống không ổn định. Khi lên dốc, xe tự động nghiêng ra trước và giữ cho trọng lượng dồn về bánh xe. Tương tự như vậy khi xuống dốc, xe nghiêng ra sau và giữ trọng tâm rơi vào bánh xe. Chính vì vậy không có hiện tượng trọng tâm rơi ra khỏi xe và làm cho nó bị lật.



Hình 3 Mô hình robot 2 bánh trên địa hình nghiêng

* 1. **Mô tả đề tài**

Thiết kế mô hình và bộ điều khiển từ xa cho xe hai bánh tự cân bằng. Mô hình sử dụng vi điều khiển chính là Arduino, dùng pin cell để cấp nguồn. Khung xe được thiết kế phù hợp cho việc sắp xếp linh kiện. Sử dụng cảm biến góc nghiên để thu thập dữ liệu góc nghiên của xe. Nhóm sử dụng động cơ DC có encoder và dùng mạch cầu H điều khiển động cơ. Động cơ hoạt động dựa trên động học được tính toán từ cơ sở lý thuyết. Để giao tiếp với máy tính nhóm sử dụng kết nối bluetooth qua HC06 để lấy dữ liệu. Bộ điều khiển được thiết kế phù hợp để hệ thống hoạt động tối ưu.

* 1. **Mục tiêu nghiên cứu**

Nhóm thực hiện đề tài với mục tiêu chính là sử dụng giải thuật điều khiển LQR để mô hình xe hai bánh có thể tự cân bằng ngay cả khi bị tác động bởi những yếu tố môi trường bên ngoài. Ngoài ra còn những mục tiêu khác của đề tài đặt ra như sau:

* Tìm hiểu các mô hình xe hai bánh tự cân bằng và các nguyên lý cơ bản của việc giữ thăng bằng cho xe
* Tính toán các thông số động lực học, xây dựng các hàm không gian – trạng thái (state – space) của mô hình
* Xây dựng thuật toán điều khiển động cơ duy trì trạng thái cân bằng robot
* Nghiên cứu lí thuyết điều khiển giải thuật PID cho điều khiển hệ xe tự cân bằng
* Thực hiện mô phỏng Matlab Simulink để khiểm nghiệm và đi tìm bộ thông số phù hợp, tối ưu cho hệ xe
* Thi công thực tế và thực hiện lập trình cho hệ xe cân bằng
  1. **Phần cứng sử dụng**
* Cảm biến gia tốc MPU6050
* Mạch cầu H L298N
* Vi điều khiển Arduino Mega 2560
* Động cơ giảm tốc JGB37-520 12V 333rpm có encoder kèm gá đỡ và 2 bánh xe V2 65mm
* Module thu phát Bluetooth HC-06
  1. **Nội dung nghiên cứu**

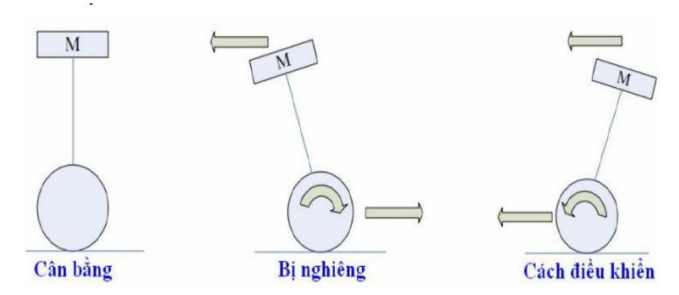
Các nội dung chính sẽ được thực hiện trong đề tài:

* Chương 1: Tổng quan
* Chương 2: Cơ sở lý thuyết của xe hai bánh tự cân bằng
* Chương 3: Thiết kế phần cứng và thi công hệ thống
* Chương 4: Thiết kế phần mềm
* Chương 5: Kết quả thực hiện

**CHƯƠNG 2 CƠ SỞ LÝ THUYẾT CỦA XE HAI BÁNH TỰ CÂN BẰNG**

**2.1 Nguyên lý cân bằng robot**

Một Robot có khả năng tự giữ thăng bằng khi di chuyển trên hai bánh xe của nó được gọi là robot hai bánh tự cân bằng. Quá trình cân bằng là quá trình điều khiển hai động cơ bánh xe sao cho đưa hệ xe về đúng vị trí cân bằng mà ta thiết lập, hay có thể nói cách khác nó duy trì trạng thái ổn định cho hệ thống. Hai bánh xe là đối tượng điều khiển chính giúp hệ xe chuyển động qua lại xung quanh ngưỡng cân bằng và ổn định.



Hình 4 Nguyên lý hoạt động của hệ xe cân bằng

Khi xe đứng thăng bằng thẳng đứng thì góc nghiêng giữa thân xe và trục thẳng đứng (lực trong trường) 𝜓 = 0.

Khi hệ xe có xu hướng nghiêng về phía trước 𝜓>0, nếu không có điều khiển thì theo quán tính, hai bánh xe sẽ tự động di chuyển ngược về phía sau (phía ngược lại của thân robot đang nghiêng) điều đó sẽ làm xe bị đổ. Trong tình huống này, chúng ta sẽ điều khiển làm sao cho 2 bánh xe chạy về phía trước (phía xe đang nghiêng) nhằm giảm góc lệch 𝜓=0, xe sẽ thăng bằng trở lại.

Tương tự như vậy, khi xe nghiêng về phía sau 𝜓>0, nếu không có điều khiển thì xe sẽ di chuyển thẳng về phía trước (phía ngược lại của thân xe đang nghiêng) làm xe sẽ bị đổ. Vì vậy, chúng ta sẽ điều khiển sao cho 2 bánh xe chạy về phía sau (phía xe đang nghiêng) nhằm giảm góc lệch 𝜓 =0, xe sẽ thăng bằng trở lại

Loại robot này đặt ra nhiều thử thách cho những người nghiên cứu trong điều kiện làm việc không ổn định đòi hỏi người thực hiện phải lựa chọn giải thuật và đưa ra các 4 thông số phù hợp. Lý thuyết để duy trì sự ổn định này dựa trên lý thuyết điều khiển con lắc ngược là một đối tượng điều khiển phi tuyến tính kinh điển, mô hình xe cân bằng cũng dựa trên điều đó mà phát triển lên, có nhiều điểm tương đồng giữa hai mô hình nhưng cả hai đều nhằm mục đích duy trì sự cân bằng và ổn định của hệ thống.

Mục đích của nguyên lý con lắc ngược là giữ cho các bánh xe ở ngay dưới trọng tâm của khung gầm robot. Khi khung xe có xung hướng nghiêng về phía trước, hai bánh xe sẽ tự động chuyển động về phía trước sao cho góc nghiêng bằng không thì robot sẽ được duy trì sự cân bằng. Mặt khác, nếu khung nghiêng về sau hai bánh xe sẽ di chuyển ngược về phía sau

**2.2 Mô hình con lắc ngược**

***Mô hình toán học của con lắc ngược với các tham số***:

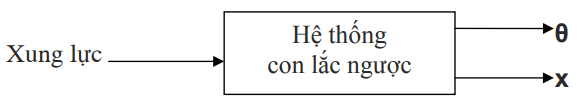
|  |  |  |
| --- | --- | --- |
| **Kí hiệu** | **Đơn vị** | **Ý nghĩa** |
| M | Kg | Khối lượng xe |
| m | Kg | Khối lượng con lắc |
| b | N | Ma sát của xe |
| L | M | Chiều dài ½ con lắc |
|  | Nm | Momen quán tính của con lắc |
| F | N | Lực tác động vào xe |
| *X* | M | Vị trí của xe |
|  | rad | Góc của con lắc so với phương thẳng đứng |

Mô hình nghiên cứu gồm:

- Một con lắc gắn bởi 1 lớp bản lề với xe

- Tác động lực F vào xe

Sơ đồ khối tín hiệu ngõ vào và ra của con lắc:



Hình 5 Sơ đồ khối tín hiệu của hệ con lắc ngược

Phân tích lực tác động ta có:

- Tổng lực tác động lên xe theo phương ngang:

- Tổng lực tác động lên con lắc theo phương ngang:

* Tổng lực tác dụng ở phương vuông góc với con lắc:

- Tổng momen tại khối tâm con lắc:

Từ (3) và (6) xấp xỉ tuyến tính hóa tại :

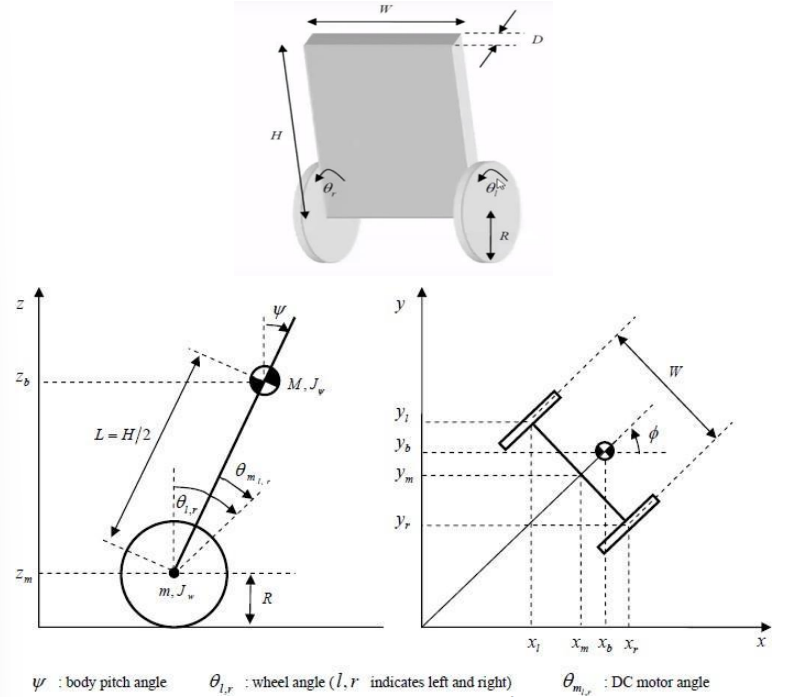
* Hàm trạng thái (state-space):

Nếu xét ta được hàm trạng thái đơn giản:

**2.3 Đặc tính động lực học**

Theo Jian Fang (2014). The LQR Controller Design of Two-Wheeled Self-Balancing Robot Based on the Particle Swarm Optimization Algorithm. Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014.

Ta có mô hình xe hai bánh tự cân bằng như sau:



Hình 6 Mô hình động học hệ xe cân bằng

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Kí hiệu*** | ***Đơn vị*** | ***Ý nghĩa*** |  | ***Kí hiệu*** | ***Đơn vị*** | ***Ý nghĩa*** |
| m | Kg | Khối lượng bánh xe | N |  | Tỷ số giảm tốc |
| M | Kg | Khối lượng robot |  | rad | Góc trung bình của 2 bánh xe |
| R | m | Bán kính bánh xe | G |  | Gia tốc trọng trường |
| W | m | Chiều rộng Robot |  | rad | Góc bánh trái, phải |
| D | m | Chiều dài robot |  | rad | Góc nghiêng robot |
| H | m | Chiều cao robot |  | rad | Góc xoay robot |
| L | m | Khoảng cách từ trọng tâm đến trục bánh xe |  | m | Tọa độ bánh trái |
|  |  | Hệ số ma sát giữa robot với mặt sàn |  | m | Tọa độ bánh phải |
|  |  | Hệ số ma sát giữa robot với động cơ |  | m | Tọa độ trung bình |
|  |  | Momen quán tính của động cơ |  | Nm | Momen phát động theo các phương khác nhau |
|  |  | Trở kháng động cơ |  | Nm | Momen phát động của bánh trái và bánh phải |
|  | Vs/rad | Hệ số EFM của động cơ |  | A | Dòng điện của động cơ |
|  | Nm/A | Momen xoắn của động cơ |  | V | Điện áp động cơ |

Sử dụng phương pháp Euler-Lagrange để xây dựng mô hình động học. Giả sử tại thời điểm t=0 robot di chuyển về phía trước (x>0) ta có:

Góc tịnh tiến trung bình của 2 bánh xe và góc xoay của robot:

Tọa độ trung bình robot trong hệ quy chiếu:

Và

Tọa độ của bánh trái trong hệ quy chiếu:

Tọa độ của bánh phải trong hệ quy chiếu:

Tọa độ tâm đối xứng giữa 2 động cơ trong hệ quy chiếu:

Phương trình động năng của chuyển động tịnh tiến:

Phương trình động năng của chuyển động xoay:

Trong đó:

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

Phương trình thế năng:

Phương trình Lagrange:

Lấy đạo hàm L theo các biến ta được:

Momen động lực do động cơ DC sinh ra:

Và

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

Giả sử điện áp phần ứng rất nhỏ, xem như bằng 0, ta có:

Các moment lực sinh ra:

Trong đó:

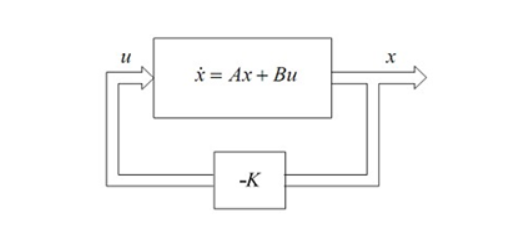
Phương trình động học mô tả chuyển động của robot:

**2.4 Điều khiển toàn phương tuyến tính**

***2.4.1 Bài toán điều khiển toàn phương tuyến tính liên tục***

Lý thuyết điều khiển tối ưu là một phương pháp tối ưu hóa cho các lý thuyết điều khiển phát sinh. Điều khiển tối ưu có thể được xem như là một chiến lược điều khiển trong lý thuyết điều khiển tự động. Phương pháp giải quyết bài toán tìm kiếm một quy luật điều khiển cho một hệ thống cho trước như là một tiêu chuẩn tối ưu ổn định đã đạt được. Hệ thống có thể được ổn định bằng cách sử dụng thông tin phản hồi trạng thái đầy đủ. Một điều khiển tối ưu là một tập hợp các phương trình vi phân mô tả đường đi của các biến điều khiển.

Sơ đồ của loại hệ thống điều khiển này được thể hiện ở hình dưới đây:



Hình 7 Sơ đồ khối bộ điều khiển toàn phương tuyến tính LQR

***2.4.2 Thuật toán LQR***

LQR (Linear Quadratic Regulator) là thuật toán điều khiển được xây dựng dựa trên cơ sở nguyên lí phản hồi trạng thái, còn gọi là phương pháp tuyến tính hóa dạng toàn phương. Bộ điều khiển LQR thường được áp dụng trên các hệ phi tuyến với nhiều ngõ vào ra. Bộ điều khiển nhận tín hiệu vào là trạng thái của hệ thống và tín hiệu mẫu, sau đó tính toán và chuyển thành tín hiệu điều khiển cho hệ thống.

Một hệ điều khiển được thiết kế ở chế độ làm việc tốt nhất là hệ thống luôn ở trạng thái tối ưu theo một tiêu chuẩn chất lượng nào đó (đạt được giá trị cực trị). Trạng thái tối ưu có đạt được hay không tùy thuộc vào yêu cầu chất lượng đặt ra, sự hiểu biết về đối tượng và các tác động lên đối tượng, điều kiện làm việc của hệ,...

Xét hệ thống có tác động ngoài (), đối tượng tuyến tính mô tả bởi phương trình trạng thái:

Với

Ta cần tìm ma trận K của vecto điều khiển tối ưu: thỏa mãn chỉ tiêu chất lượng J đạt giá trị cực tiểu:

Trong đó:

* Q là ma trận trọng số bán xác định dương của tín hiệu đầu ra
* R là ma trận trọng số xác định dương của tính hiệu điều khiển

Ma trận K tối ưu được xác định từ phương trình Riccati có dạng:

Như vậy, luật điều khiển tối ưu cho bài toán điều khiển tối ưu dạng toàn phương với chỉ tiêu chất lượng là phương trình tuyến tính có dạng:

Ma trận P khi đó phải thỏa mãn phương trình Riccati:

Khi P không thay đổi theo thời gian , ta có phương trình đại số Riccati như sau:

Từ vector K thu được xác định thông số điều khiển đưa trở lại hệ thống. Với điều kiện các phần tử trong ma trận Q và R mang các giá trị dương, ma trận Q và R làm cân bằng tính ổn định của hệ thống. Số phần tử trong ma trận Q phụ thuộc vào số phần tử của ma trận trạng thái và số phần tử của ma trận R phục thuộc vào số lượng đầu vào điề khiển hệ thống.

***2.4.3 Tuyến tính hệ thống***

Từ cơ sở động lực học hệ thống ta có:

Hạ bậc hệ thống về dạng:

Đặt:

Đặt:

Tuyến tính hóa hệ thống tại vị trí cân bằng:

Với và

Ma trận A tại vị trí cân bằng được tính:

Với các thông số để tinh chỉnh cho bộ điều khiển LQR. Trong đó các tham số được coi là trọng số tối ưu tương ứng cho 6 biến trạng thái . Việc lựa chọn trọng số Q và R tùy thuộc tiêu chuẩn đặt ra của người điều khiển và thông qua thực nghiệm để đưa ra trọng số Q và R một cách tối ưu. Để ưu tiên biến trạng thái nào, hãy tăng tham số trọng số tương ứng của nó so với các thông số trọng số khác, đồng nghĩa với việc những biến trạng thái khác sẽ không còn ổn định như ban đầu. Với mô hình hệ xe hai bánh ta có ma trận Q là ma trận 6x6 (tương ứng với 6 biến trạng thái) và R là 2x2 (tương ứng với 2 biến ngõ vào). Sau khi chọn được các tham số điều khiển tương ứng, ta có thể xây dựng được tham số phản hồi K với tín hiệu điều khiển

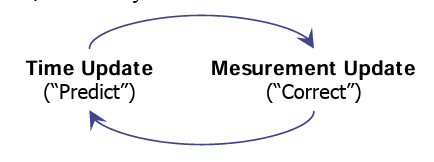
Thông số K được tính toán dựa vào phương trình Riccati. Tuy nhiên Matlab đã hỗ trợ việc tính toán thủ công bằng hàm lqr(A,B,Q,R) khi các ma trận A,B,Q,R đã được tìm ra

**2.5 Bộ lọc Kalman**

Từ năm 1960 giáo sư Kalman để thu thập và kết hợp các thông tin từ cảm biến thành phần, khi phương trình định hướng và mẫu thống kê nhiễu trên cảm biến được xác định, bộ lọc Kalman sẽ cho ước lượng giá trị tối ưu như đang sử dụng một tín hiệu tinh khiết và độ phân bố không đổi. Trong hệ thống xe hai bánh tự cân bằng, tín hiệu ngõ ra của bộ lọc là tín hiệu từ cảm biến góc và cảm biến vận tốc góc đã được loại bỏ nhiễu nhờ nguồn tín hiệu hỗ trợ và xử lý trong bộ lọc thông qua quan hệ vận tốc góc bằng vi phân giá trị góc.

Bộ lọc Kalman ước tính quá trình bằng cách ử dụng một hình thức kiểm soát phản hồi. Các trạng thái của quá trình được ước tính bở bộ lọc tại một thời điểm nhất định và sau đó có được một phản hồi về các phép đo nhiễu. Do đó phương trình bộ lọc gồm có phương trình cập nhật thời gian và phương trình cập nhật đo lường.

Hoạt động của bộ lọc được mô tả:



Hình 8 Mô tả hoạt động của bộ lọc Kalman

***2.5.1 Phương trình trạng thái***

Bộ lọc Kalman có thể ước lượng các trạng thái khi hệ thống được mô hình đầy đủ dưới dạng phương trình vi phân tuyến tính. Khi một hệ thống được mô hình hóa trong không gian trạng thái và phương trình đại diện cho quá trình trạng thái của hệ thống trong ký hiệu nhiễu trắng.

F(k): mô hình chuyển đổi trạng thái.

B(k): mô hình tín hiệu điều khiển

W(k): nhiễu quá trình có phân bố Gausian với trung bình 0 và ma trận hiệp phương sai Q tại thời điểm k

Bộ lọc này không giả định các nhiễu là Gausian nhưng bộ lọc tạo ra ước tính xác xuất có điều khiện chính xác trong trường hợp đặt biệt là tất cả các nhiễu đều phân bố Gausian.

***2.5.2 Phương trình đo***

Các phép đo hệ thống đuộc thực hiện tại các mốc thời gian rời rạc được bao gồm trong vector đo lường và có thể được mô hình hóa theo quan hệ sau:

H: mô hình quan sát

v(k): nhiễu quá trình có phân bố Gausian với trung bình 0 và ma trận hiệp phương sai R tại thời điểm k

Các phép đo phục thuộc vào trạng thái hệ thống, có liên hệ với ma trận đo kết hợp với nhiễu trong phép đo.

***2.5.3 Phương trình bộ lọc Kalman***

Phương trình cập nhật thời gian bộ lọc Kalman gián đoạn

Phương trình cập nhật giá trị đo bộ lọc Kalman gián đoạn:

Phương trình ước tính lỗi:

**2.6 Sơ đồ mô phỏng hệ thống trên Matlab Simulink**

***Mô phỏng hệ thống bằng Matlab Simulink với các thông số thực tế của mô hình:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| m | 0.04 |  | H | 0.113 |
| M | 0.72 | L | 0.565 |
| R | 0.0325 |  | 0.0218 |
| W | 0.16 |  | 0.4312 |
| D | 0.08 |  |  |
|  | 40 |  |  |
|  | 0.002 |  |  |
|  | 0.18 |  |  |
| N | 45 |  |  |
| G | 9.81 |  |  |
| T | 0.0197 |  |  |

***Xét hệ xe hai bánh tự cân bằng được mô tả như sau:***

function [teta\_dd,psi\_dd,phi\_dd] = fcn(vl,vr,teta,teta\_d,psi,psi\_d,phi,phi\_d)

%% Thông số hệ thống xe 2 bánh tự cân bằng dùng LQR

m = 0.04; %Khoi luong banh xe

M = 0.72; %Khoi luong robot

R = 0.0325; %ban kinh ban xe

W = 0.16; %Chieu rong robot

D = 0.08; %Chieu sau robot

H = 0.113; %Chieu cao robot

L = 0.565; %khoang cach tu trong tam den truc banh xe

fw = 0.18; %He so ma sat giua banh xe voi mat phang

fm = 0.002; %he so ma sat giua dong co va robot

Jm = 1.4762\*10^(-5); %moment quan tinh cua dong co

Jw = m\*R^2/2;

J\_psi = M\*L^2/3;

J\_phi = M\*(W^2+D^2)/12;

Rm = 40; %Dien tro dong co DC

Kb = 0.0218; %he so emf cua dong co

Kt = 0.4312; %Momen xoan cua dong co DC

n = 45; %Ty so giam toc

g = 9.81; %Gia toc trong truong

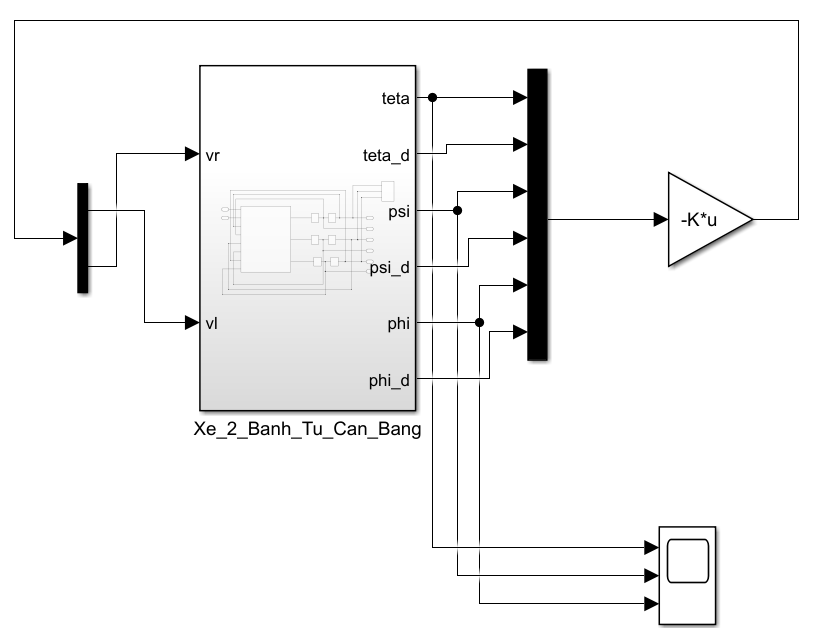
alpha = n\*Kt/Rm; beta=n\*Kt\*Kb/Rm+fm; a =alpha;

T=0.0197;

teta\_dd = (J\_psi\*a\*vl + J\_psi\*a\*vr - 2\*J\_psi\*beta\*teta\_d + 2\*J\_psi\*beta\*psi\_d - 2\*J\_psi\*fw\*teta\_d + L^2\*M\*a\*vl + L^2\*M\*a\*vr - 2\*L^2\*M\*beta\*teta\_d + 2\*L^2\*M\*beta\*psi\_d - 2\*L^2\*M\*fw\*teta\_d - 4\*Jm\*fw\*n^2\*teta\_d + L^3\*M^2\*R\*psi\_d^2\*sin(psi) - 2\*L\*M\*R\*beta\*teta\_d\*cos(psi) + 2\*L\*M\*R\*beta\*psi\_d\*cos(psi) - L^3\*M^2\*R\*phi\_d^2\*cos(psi)^2\*sin(psi) - L^2\*M^2\*R\*g\*cos(psi)\*sin(psi) + J\_psi\*L\*M\*R\*psi\_d^2\*sin(psi) + 2\*Jm\*L\*M\*g\*n^2\*sin(psi) + L\*M\*R\*a\*vl\*cos(psi) + L\*M\*R\*a\*vr\*cos(psi) + 2\*Jm\*L^2\*M\*n^2\*phi\_d^2\*cos(psi)\*sin(psi) + 2\*Jm\*L\*M\*R\*n^2\*psi\_d^2\*sin(psi))/(2\*J\_psi\*Jw + L^2\*M^2\*R^2 + 2\*Jw\*L^2\*M + J\_psi\*M\*R^2 + 2\*J\_psi\*Jm\*n^2 + 4\*Jm\*Jw\*n^2 + 2\*J\_psi\*R^2\*m + 2\*Jm\*L^2\*M\*n^2 + 2\*Jm\*M\*R^2\*n^2 + 2\*L^2\*M\*R^2\*m + 4\*Jm\*R^2\*m\*n^2 - L^2\*M^2\*R^2\*cos(psi)^2 + 4\*Jm\*L\*M\*R\*n^2\*cos(psi));

psi\_dd = -(2\*Jw\*a\*vl + 2\*Jw\*a\*vr - 4\*Jw\*beta\*teta\_d + 4\*Jw\*beta\*psi\_d + M\*R^2\*a\*vl + M\*R^2\*a\*vr - 2\*M\*R^2\*beta\*teta\_d + 2\*M\*R^2\*beta\*psi\_d + 2\*R^2\*a\*m\*vl + 2\*R^2\*a\*m\*vr + 4\*Jm\*fw\*n^2\*teta\_d - 4\*R^2\*beta\*m\*teta\_d + 4\*R^2\*beta\*m\*psi\_d - L\*M^2\*R^2\*g\*sin(psi) - 2\*Jw\*L\*M\*g\*sin(psi) - 2\*L\*M\*R\*beta\*teta\_d\*cos(psi) + 2\*L\*M\*R\*beta\*psi\_d\*cos(psi) - 2\*L\*M\*R\*fw\*teta\_d\*cos(psi) + L^2\*M^2\*R^2\*psi\_d^2\*cos(psi)\*sin(psi) - L^2\*M^2\*R^2\*phi\_d^2\*cos(psi)\*sin(psi) - 2\*Jw\*L^2\*M\*phi\_d^2\*cos(psi)\*sin(psi) - 2\*Jm\*L\*M\*g\*n^2\*sin(psi) - 2\*L\*M\*R^2\*g\*m\*sin(psi) + L\*M\*R\*a\*vl\*cos(psi) + L\*M\*R\*a\*vr\*cos(psi) - 2\*Jm\*L^2\*M\*n^2\*phi\_d^2\*cos(psi)\*sin(psi) - 2\*L^2\*M\*R^2\*m\*phi\_d^2\*cos(psi)\*sin(psi) - 2\*Jm\*L\*M\*R\*n^2\*psi\_d^2\*sin(psi))/(2\*J\_psi\*Jw + L^2\*M^2\*R^2 + 2\*Jw\*L^2\*M + J\_psi\*M\*R^2 + 2\*J\_psi\*Jm\*n^2 + 4\*Jm\*Jw\*n^2 + 2\*J\_psi\*R^2\*m + 2\*Jm\*L^2\*M\*n^2 + 2\*Jm\*M\*R^2\*n^2 + 2\*L^2\*M\*R^2\*m + 4\*Jm\*R^2\*m\*n^2 - L^2\*M^2\*R^2\*cos(psi)^2 + 4\*Jm\*L\*M\*R\*n^2\*cos(psi));

phi\_dd = -(W^2\*beta\*phi\_d + W^2\*fw\*phi\_d + R\*W\*a\*vl - R\*W\*a\*vr + 4\*L^2\*M\*R^2\*psi\_d\*phi\_d\*cos(psi)\*sin(psi))/(2\*J\_phi\*R^2 + Jm\*W^2 + Jm\*W^2\*n^2 + R^2\*W^2\*m + 2\*L^2\*M\*R^2\*sin(psi)^2);

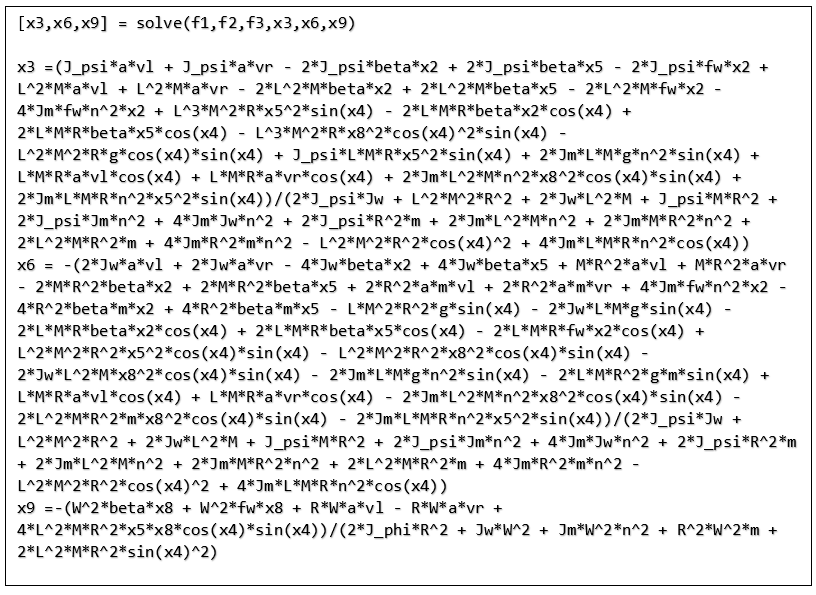


Hình 9 Mô hình xe cân bằng mô phỏng trong Matlab Simulink

Từ phương trạng thái của hệ thống được xây dựng từ chương trước:

Với và

Thông qua phần mềm Matlab với lệnh solve:



Sau đó chúng ta tìm vector điều khiển tối ưu K cho tín hiệu điều khiển

bằng cách giải phương trình Riccati hoặc sử dụng lệnh được hỗ trợ từ phần mềm Matlab: K = lqr(A,B,Q,R)

*Kết quả thu được vector K và Q:*

K = [ -70.7107 -2.0518 -467.9601 -132.4575 -2.2361 -0.4851

-70.7107 -2.0518 -467.9601 -132.4575 2.2361 0.4851]

*Đáp ứng ngõ ra:*



Hình 10 Đáp ứng ngõ ra của hệ thống

**CHƯƠNG 3 THIẾT KẾ PHẦN CỨNG VÀ THI CÔNG HỆ THỐNG**

**3.1 Yêu cầu thiết kế**

Để thiết kế phần cứng đảm bảo cho xe hoạt động, cần lựa chọn các thiết bị phù hợp với các yêu cầu sau:

- Đảm bảo tổng khổi lượng của xe để động cơ có thể kéo cân bằng được.

- Nguồn cung cấp đủ cho hệ thống hoạt động.

- Khung xe chắc chắn.

- Cảm biến trả tín hiệu về tránh tối đa nhiễu và sai số.

- Động cơ moment đủ để kéo tải.

- Vi điều khiển và các cảm biến, đảm bảo xử lý tín hiệu nhanh

**3.2 Sơ đồ khối hệ thống**

*Kích thước*: 10 x 18 x 20 2cm

*Sơ đồ khối mô tả hệ thống được chia thành các thành phần sau*:

- Bộ khung xe.

- Khối động cơ và cảm biến đọc xung của động cơ, cảm biến đo góc nghiêng.

- Khối điều khiển trung tâm, khối điều khiển động cơ.

- Khối nguồn

- Khối thu phát tín hiệu

*Chức năng của từng khối từng khối*:

- ***Bộ khung xe***: Cố định từng khối thành phần của xe với nhau, lựa chọn kích thước chiều dài chiều rộng phù hợp để có thể chứa vừa các khối thành phần đã được nêu và dễ dàng cân bằng được.

- ***Khối cảm biến đo góc nghiêng***: Có nhiệm vụ đọc dữ liệu trả về và gửi tín hiệu đó về *khối điều khiển trung tâm* xử lý ra giá trị góc nghiêng của xe.

- ***Khối điều khiển trung tâm*** là một vi điều khiển, có nhiệm vụ chính là xử lý tất cả dữ liệu đọc về từ *khối đo góc nghiêng* và *khối cảm biến đọc xung của động cơ*, xử lý các thuật toán điều khiển đưa ra tín hiệu để điều khiển *khối điều khiển động cơ*.

- ***Khối điều khiển động cơ***: Nhiệm vụ là nhận tín hiệu từ *khối điều khiểu trung tâm* để cấp điện áp ra phù hợp để điều khiển *khối động cơ*

- ***Khối động cơ và cảm biến đọc xung của động cơ***: Nhận điện áp cấp từ *khối điều khiển động cơ* điều khiển xe di chuyển qua lại, nhanh, chậm, dừng để cho xe có thể đạt được trạng thái cân bằng. Ngoài ra còn có một bộ mã hoá quay encoder để trả về tín hiệu số xung đọc được, từ đó qua *khối điều khiển trung tâm* xử lý có thể xác định chính xác được vị trí, tốc độ quay và chiều quay động cơ.

- ***Khối nguồn***: Cấp nguồn ổn định cho các linh kiện và động cơ

***- Khối thu phát tín hiệu***: Dùng để truyền nhận dữ liệu, cung cấp tín hiệu cho vi điều khiển

**3.3 Lựa chọn thiết bị phần cứng**

***3.3.1 Khối cấp nguồn***

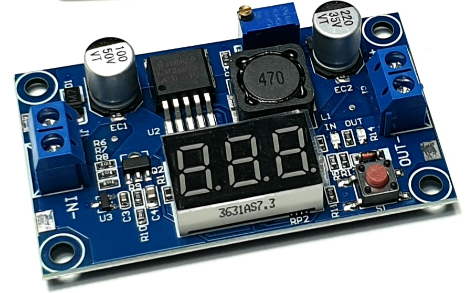
Cung cấp nguồn ổn định cho các khối giúp hệ thống hoạt động tốt. Có thể cấp nguồn cho hệ thống bằng nhiều cách như pin, nguồn xung tổ ong hoặc adapter chuyển đổi AC-DC, …

Nguồn cấp được lựa chọn là Pin Lithium-ion (Li-Ion) 18650 4.2V 4200mA. Mô hình sử dụng bốn viên cấp nguồn cho hệ thống hoạt động bình thường.



Hình 11 Pin Lithium-ion (Li-Ion) 18650

Ngoài ra, nguồn cấp còn có mạch hạ áp DC-DC LM2596 dùng để giảm điện áp nguồn từ 16V xuống còn 12V cấp cho động cơ DC



Hình 12 Mạch hạ áp DC-DC LM2596

*Thông số kỹ thuật:*

* Dùng IC LM2596 với tần số lên đến 150Khz.
* Có nút nhấn chuyển chế độ hiển thị ngõ ra/vào.
* Điện áp đầu vào: Từ 4~30VDC
* Điện áp đầu ra: Điều chỉnh được trong khoảng 1.5~29VDC
* Dòng ngõ ra tối đa là 3A.
* Công suất : 15W
* Kích thước: 66 x 35mm.

***3.3.2 Khối động cơ và cảm biến đọc xung của động cơ***

Khối động cơ và cảm biến đọc xung của động cơ gồm:

- Hai động cơ bên phải và bên trái là cơ cấu chấp hành chính của xe để điều khiển xe cân bằng.

- Hai bộ mã hoá quay encoder gắn liền với mỗi động cơ giúp hồi tiếp xung về *khối điều khiển trung tâm* từ đó xử lý tín hiệu giúp xác định chính xác vị trí, tốc độ quay và chiều quay của từng động cơ.

Nhóm đã lựa chọn động cơ hộp số JGB37-520 12V 333rpm. Động cơ hộp số JGB37-520 12V333RPM có encoder là loại động cơ có hộp số và Encoder, thường được dùng trong các hệ cơ khí cần độ chính xác cao và các mô hình như xe dò line, xe chạy mê cung, xe 2 bánh tự cân bằng,...



Hình 13 Động cơ hộp số JGB37-520 12V333RPM

*Thông số kỹ thuật:*

- Tỉ số truyền 30:1 (động cơ quay 30 vòng trục chính hộp giảm tốc quay 1 vòng).

- Dòng không tải: 120mA

- Dòng chịu đựng tối đa khi có tải: 1A

- Tốc độ không tải: 333rpm (333 vòng 1 phút)

- Tốc độ chịu đựng tối đa khi có tải: 250rpm (250 vòng 1 phút)

- Lực kéo Momen định mức: 3.5 kg.cm

- Lực léo Moment tối đa: 5 kg.cm

- Chiều dài hộp số L: 22mm

- Số xung Encoder mỗi kênh trên 1 vòng quay trục chính: 11 x 30 = 330 xung.

***Bảng kết nối chân của động cơ và encoder:***

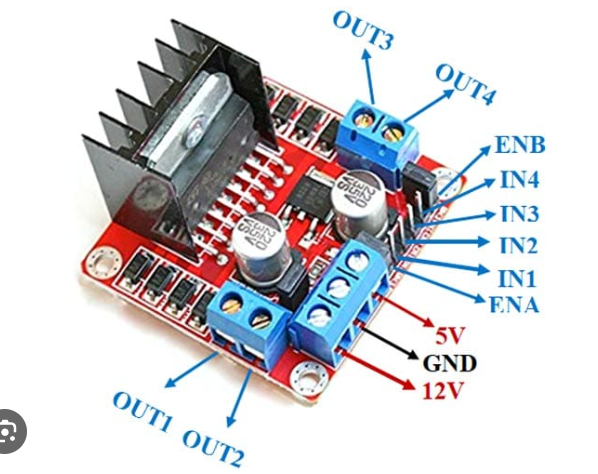
|  |  |
| --- | --- |
| **Chân** | **Kết nối và chức năng** |
| (đỏ) | Dây cấp nguồn dương cho động cơ |
| GND (đen) | Dây cấp nguồn âm cho Encoder |
| /A (vàng) | Tín hiệu trả xung kênh A |
| /B (xanh lá) | Tín hiệu trả xung kênh B |
| (xanh dương) | Dây cấp nguồn dương cho Encoder |
| (trắng) | Dây cấp nguồn âm cho động cơ |

***3.3.3 Khối điều khiển động cơ***

Khối điều khiển động cơ nhận tín hiệu PWM (điều chế độ rộng xung) từ *khối điều khiển trung tâm* để dễ dàng điều khiển hai động cơ bánh trái và bánh phải quay thuận, quay nghịch, dừng hoặc theo tốc độ nhanh, chậm để xe dễ dàng đạt trạng thái cân bằng.

Giá trị trung bình của điện áp (và dòng điện) cung cấp cho tải được kiểm soát bằng cách thay đổi việc đóng cắt giữa nguồn và tải tắt với tốc độ rất nhanh. Thời gian đóng càng lâu so với thời gian cắt, thì tổng công suất cung cấp cho tải càng cao.

Nhóm lựa chọn mạch điều khiển động cơ DC L298N (Dual Full Bridge Driver).



Hình 14 Mạch cầu H điều khiển động cơ DC L298N

*Thông số kĩ thuật*:

* IC chính: L298 – Dual Full Bridge Driver
* Điện áp hoạt động: 5~30VDC
* Công suất tối đa: 25W 1 cầu (lưu ý công suất = dòng điện x điện áp nên điện áp cấp vào càng cao, dòng càng nhỏ, công suất cố định 25W).
* Dòng tối đa cho mỗi cầu H là: 2A
* Mức điện áp logic: Low: -0.3V~1.5V; High: 2.3V~Vss
* Kích thước: 43x43x27mm

***Bảng kết nối chân của mạch cầu H L298N:***

|  |  |
| --- | --- |
| **Chân** | **Kết nối và chức năng** |
| 12V | Chân cấp nguồn dương (9 - 12) |
| GND | Chân nối GND (nối chung với GND Arduino) |
| 5V | Chân nguồn ra 5V |
| ENA | Chân điều khiển mạch cầu H động cơ 1 (cấp xung PWM cho động cơ 1) |
| IN1 | Chân nhận tín hiệu điều khiển động cơ 1 |
| IN2 | Chân nhận tín hiệu điều khiển động cơ 1 |
| ENB | Chân điều khiển mạch cầu H động cơ 2 (cấp xung PWM cho động cơ 2) |
| IN3 | Chân nhận tín hiệu điều khiển động cơ 2 |
| IN4 | Chân nhận tín hiệu điều khiển động cơ 2 |
| OUT1 | Chân nối với chân dương của động cơ 1 |
| OUT2 | Chân nối với chân âm của động cơ 1 |
| OUT3 | Chân nối với chân dương của động cơ 2 |
| OUT4 | Chân nối với chân âm của động cơ 2 |

***3.3.4 Khối điều khiển trung tâm***

Khối điều khiển trung tâm là một vi điều khiển có nhiệm vụ xử lý dữ liệu đọc về từ cảm biến đo góc nghiêng và bộ mã hoá quay encoder từ đó xử lý các thuật toán về điều khiển xuất ra tín hiệu đến *khối điều khiển động cơ*.

Khối điều khiển trung tâm phải có tốc độ xử lý nhanh với hiệu suất làm việc cao. Một số Board vi điều khiển có thể xử dụng như Arduino Uno R3, STM32F4, Rasberry, họ vi điều khiển PIC, …

Nhóm đã chọn sử dụng Board Arduino Mega 2560 để nhận dữ liệu từ cảm biến góc nghiêng, cấp tín hiệu điều khiển động cơ và truyền nhận dữ liệu với Module Bluetooth



Hình 15 Vi điều khiển Arduino Mega2560

*Thông số kĩ thuật:*

* Vi xử lý chính: ATmega2560
* IC nạp và giao tiếp UART: CH340.
* Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ giắc tròn DC
* Số chân Digital I/O: 54 (trong đó 15 chân có khả năng xuất xung PWM)
* Số chân Analog Input: 16
* Dòng điện DC Current trên mỗi chân I/O: 20mA
* Dòng điện DC Current chân 3.3V: 50mA
* Flash Memory: 256 KB trong đó 8 KB sử dụng cho bootloader.
* SRAM: 8 KB
* EEPROM: 4 KB
* Clock Speed: 16 MHz
* LED\_BUILTIN: 13
* Kích thước: 101.52 x 53.3 mm

***3.3.5 Khối cảm biến đo góc nghiêng***

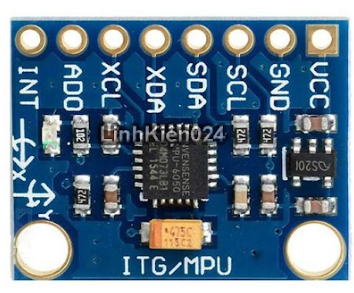
Khối cảm biến đo góc nghiêng là một module gồm hai cảm biến đó là:

- Accelerometer: cảm biến đo gia tốc của module (bao gồm cả gia tốc trọng lực), có ba trục xyz tương ứng với ba chiều không gian.

- Gyroscope: cảm biến đo tốc độ quay của module quanh một trục, có ba trục xyz tương ứng với ba chiều không gian.

Khối điều khiển trung tâm sẽ nhận tín hiệu trả về từ hai cảm biến này và sẽ xử lý tính toán ra được giá trị góc nghiêng của xe.

Đối với yêu cầu hệ thống, chỉ để xác định giá trị góc nghiêng của xe thì nhóm lựa chọn ra được loại cảm biến GY-521 MPU6050



Hình 16 Cảm biến góc nghiêng MPU6050

*Thông số kỹ thuật*:

* Gồm 6 trục cảm biến (6DOF): Cảm biến gia tốc (3 trục) và cảm biến con quay hồi chuyển (3 trục)
* Bộ chuyển đổi tương tự sang số (ADC): 16 bit
* Điện áp sử dụng: 3~5VDC
* Điện áp giao tiếp: 3~5VDC
* Chuẩn giao tiếp: I2C
* Giá trị Gyroscopes trong khoảng: 250, 500, 1000, 2000/sec
* Giá trị Acceleration trong khoảng: 2g, 4g, 8g, 16g

***Bảng kết nối chân của cảm biến MPU6050:***

|  |  |
| --- | --- |
| **Chân** | **Kết nối và chức năng** |
|  | Chân cấp nguồn dương |
| GND | Chân cấp GND |
| SCL | Serial Clock Line |
| SDA | Serial Data Line |

***3.3.6 Khối thu phát tín hiệu***

Khối thu phát tín hiệu dùng để giao tiếp giữa máy tính và vi điều khiển, cung cấp tín hiệu điều khiển từ máy tính và trả về tín hiệu các góc ngõ ra. Để thực hiện đề tài này, nhóm quyết định chọn Module Bluetooth HC06 dùng làm khối thu phát tín hiệu.

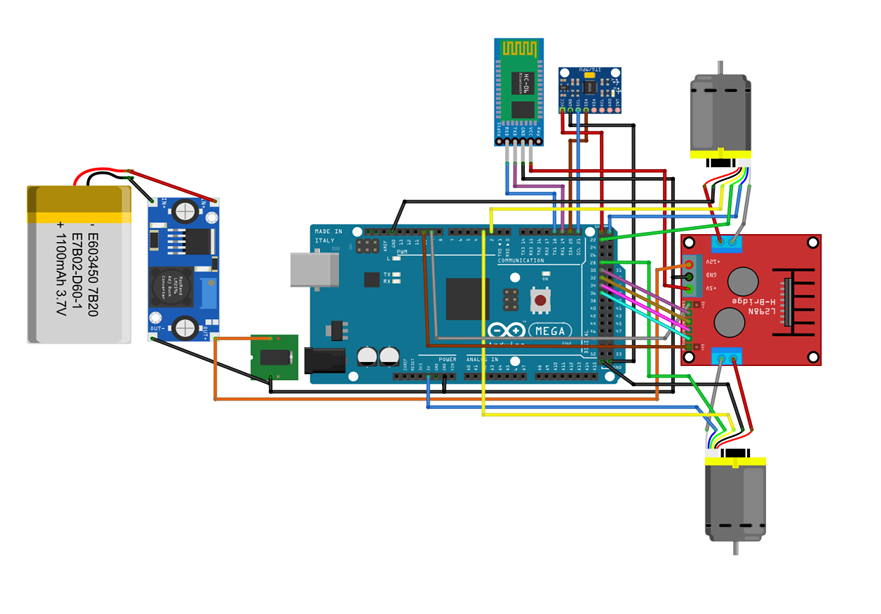


Hình 17 Module Bluetooth HC06

*Thông số kĩ thuật*:

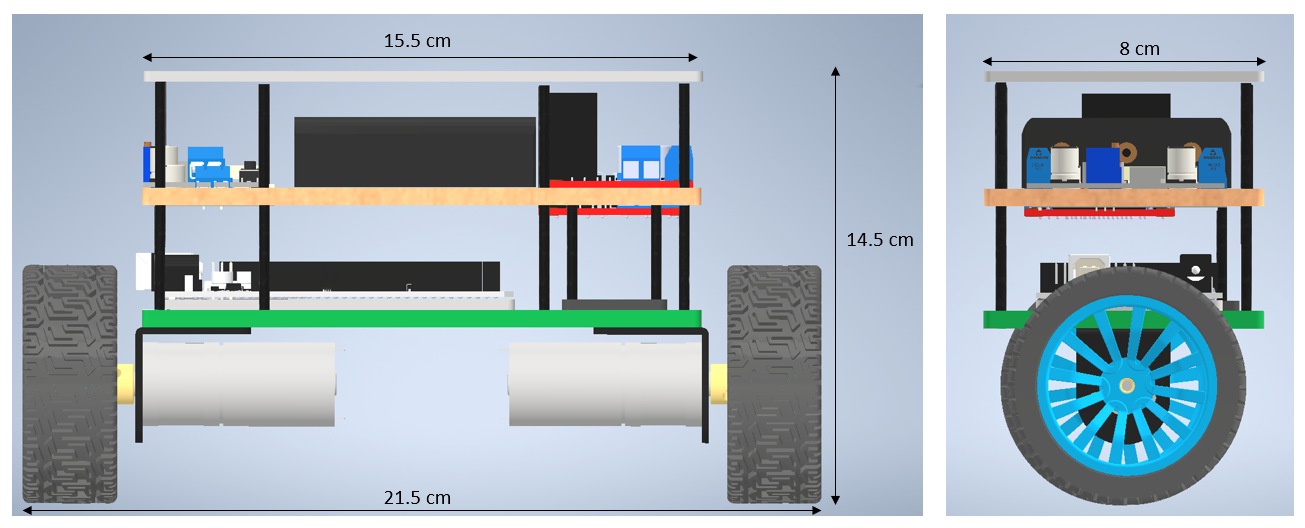
* Điện áp hoạt động: 3.3 ~ 5VDC.
* Điện áp giao tiếp: TTL tương thích 3.3VDC và 5VDC.
* Baudrate UART có thể chọn được: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
* Dải tần sóng hoạt động: Bluetooth 2.4GHz
* Sử dụng CSR mainstream bluetooth chip, bluetooth V2.0 protocol standards.
* Dòng điện khi hoạt động: khi Pairing 30 mA, sau khi pairing hoạt động truyền nhận bình thường 8 mA
* Kích thước của module chính: 28 mm x 15 mm x 2.35 mm

**3.4 Sơ đồ nối dây**

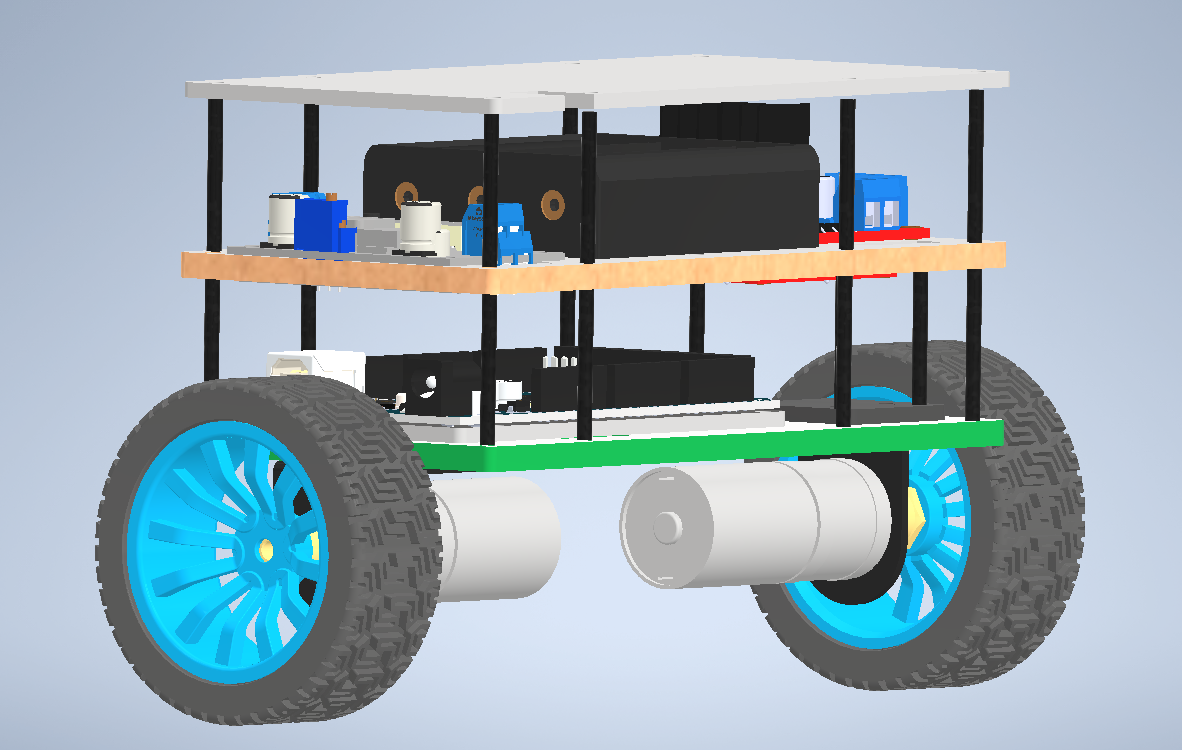


Hình 18 Sơ đồ nối dây của mô hình

**3.5 Xây dựng mô hình 3D trên môi trường Inventor**



Hình 19 Thông số mô hình



Hình 20 Mô hình 3D robot 2 bánh tự cân bằng

**CHƯƠNG 4 THIẾT KẾ PHẦN MỀM**

**4.1 Yêu cầu thiết kế**

Về phần mềm điều khiển hệ thống các yêu cầu thiết kế để xe đảm bảo cân bằng như sau:

- Đọc tín hiệu cảm biến MPU6050 thông qua truyền thông I2C.

- Đọc giá trị xung encoder trả về.

- Xuất xung PWM điều khiển mạch cầu H điều khiển động cơ theo tốc độ, chạy thuận, chạy nghịch, dừng động cơ.

- Giới hạn góc nghiêng để cân bằng xe từ -20 đến 20

**4.2 Xây dựng thuật toán điều khiển**

Để sử dụng và lập trình Arduino Mega 2560 thì phần mềm sử dụng để lập trình là Arduino IDE 1.8.13, thuận tiện trong việc kết nối Arduino với máy tính bằng dây cáp USB.

Ngôn ngữ lập trình của Arduino là C/C++, lập trình đơn giản hơn nhiều so với các loại vi điều khiển khác nhờ cộng đồng người dùng lớn và sẵn các thư viện hỗ trợ lập trình đa dạng.

***4.2.1 Thuật toán chương trình chính***

Bắt đầu

Khai báo thư viện

Khai báo Ports

Khai báo biến

Khởi tạo Serial

Khởi tạo Timer2

Khởi tạo ngắt

Khởi tạo I2C

Thời gian lấy mẫu 6ms

Đọc và xử lí giá trị từ MPU6050

Ngắt nhận và xử lý tín hiệu từ Encoder

Cập nhật các giá trị mới sau khoảng thời gian lấy mẫu

Tính xung băm PWM mỗi bánh

Điều khiển động cơ

Hình 21 Lưu đồ thuật toán chương trình chính

***Phần khai báo và khởi tạo chương trình***:

- Khai báo các thư viện sử dụng: Wire.h để giao tiếp I2C với cảm biến góc nghiêng MPU6050 và Kalman.h để sử dụng bộ lọc Kalman lọc nhiễu cho cảm biến góc nghiêng MPU6050.

- Định nghĩa các chân kết nối.

- Khai báo các biến và giá trị ban đầu.

- Khởi tạo Serial giao tiếp máy tính với baud rate là 115200.

- Khởi tạo sử dụng timer2 để xuất xung PWM ở chân 9 và chân 5.

- Khởi tạo các chân xuất tín hiệu ngõ ra là chân xuất PWM của mỗi bánh, chân điều chỉnh chiều quay của mỗi bánh.

- Khởi tạo các chân nhận tín hiệu ngõ vào là chân nhận tín hiệu từ encoder kênh A và kênh B của mỗi bánh.

- Khởi tạo ngắt 0 sử dụng để đọc tín hiệu trả về từ kênh A bánh trái tác động cạnh lên, ngắt 1 sử dụng để đọc tín hiệu trả về từ kênh A bánh phải tác động cạnh lên.

- Khởi tạo hàm đọc giá trị cảm biến góc nghiêng từ giao tiếp I2C.

***Phần vòng lặp:***

- Đọc và xử lý giá trị góc nghiêng trả về từ MPU6050.

- Thời gian lấy mẫu là 6ms.

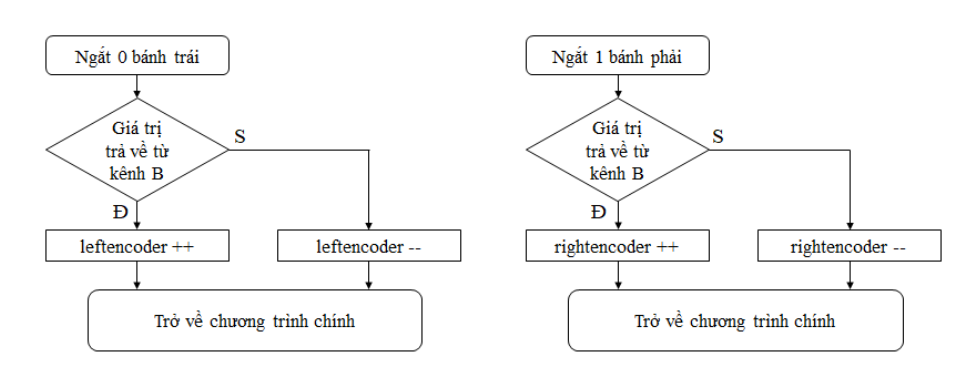
- Cập nhập các giá trị theta, psi, phi và tính toán giá trị thetadot, psidot, phidot.

- Tính toán điện áp cung cấp ở bánh trái và bánh phải.

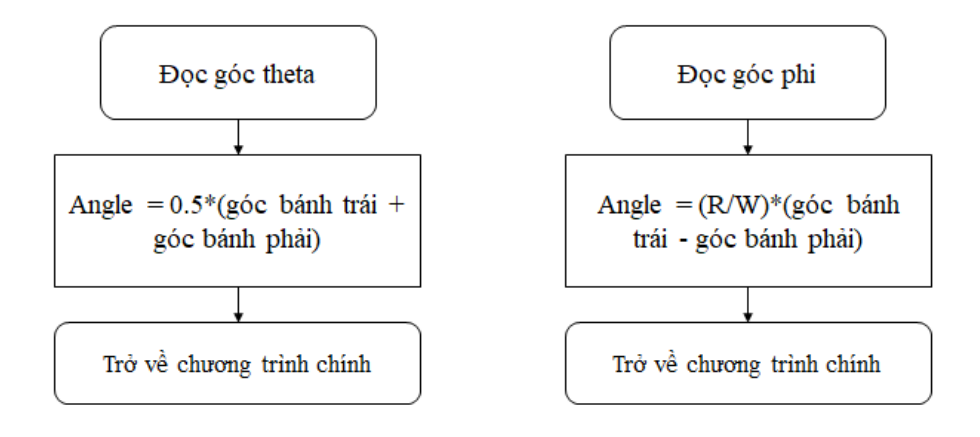
- Quy đổi sang PWM để xuất xung điều khiển cho mạch cầu.

***4.2.2 Thuật toán xử lý xung trả về từ Encoder và tính toán giá trị góc tới, góc xoay***

Sử dụng chức năng ngắt ngoại của vi điều khiển để đọc giá trị xung encoder mà không làm ảnh hưởng đến chương trình chính



Hình 22 Lưu đồ chương trình ngắt xử lý xung trả về từ Encoder



Hình 23 Lưu đồ tính toán giá trị góc tới và góc xoay

***Phần khởi tạo***:

- PIN 2 nối với encoder kênh A động cơ trái (sử dụng ngắt 0).

- PIN 22 nối với encoder kênh B động cơ trái (đọc tín hiệu trả về).

- PIN 3 nối với encoder kênh A động cơ phải (sử dụng ngắt 1).

- PIN 26 nối với encoder kênh B động cơ trái (đọc tín hiệu trả về).

***Thuật toán đọc giá trị động cơ trái***:

- Sử dụng ngắt 0 cho kênh A.

- Khi xung cạnh lên tác động thì chương trình ngắt thực hiện.

- Lúc này đọc giá trị trả về từ kênh B: Nếu kênh B đang ở mức cao thì động cơ đang quay thuận, *leftencoder++* và ngược lại kênh B đang ở mức thấp thì động cơ đang quay theo chiều nghịch, *leftencoder - -*

***Thuật toán đọc giá trị động cơ phải***:

- Sử dụng ngắt 1 cho kênh A.

- Khi xung cạnh lên tác động thì chương trình ngắt thực hiện.

- Lúc này đọc giá trị trả về từ kênh B: nếu kênh B đang ở mức cao thì động cơ đang quay thuận, *rightencoder++* và ngược lại kênh B đang ở mức thấp thì động cơ đang quay theo chiều nghịch, *rightencoder - -*

Sử dụng phương pháp này sẽ tối ưu được tốc độ xử lý của vi điều khiển thay vì phương pháp sử dụng ngắt cho cả hai kênh A và B.

***Thuật toán đọc giá trị góc tới và góc xoay :***

- Từ phần xây dựng mô hình toán đối tượng trong chương 2 ta có được công thức tính góc tới trung bình của hai bánh xe.

- Động cơ sử dụng hộp số với tỉ số truyền là 1:45, đĩa Encoder trả ra 11 xung/ vòng. Nên số xung của Encoder trên 1 vòng quay trục chính động cơ là 45 x 11 = 495 xung/vòng.

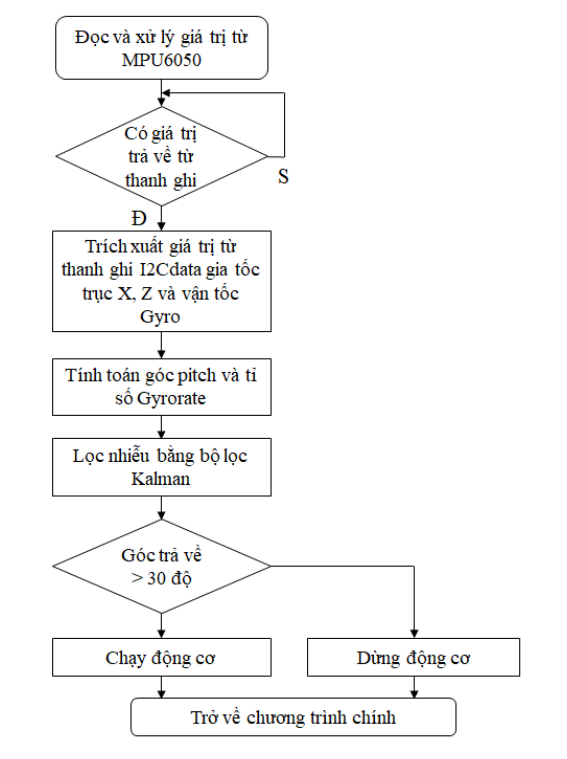
***Góc Encoder được xác định theo công thức:***

***Suy ra:***

Với:

* là xung encoder đọc được ở bánh trái
* là xung encoder đọc được ở bánh phải

***4.2.3 Thuật toán đọc và xử lí giá trị từ cảm biếng góc nghiêng***



Hình 24 Lưu đồ thuật toán đọc và xử lý giá trị từ cảm biến góc nghiêng

***Phần khởi tạo***:

- Thiết lập địa chỉ I2C là 0x68, 0x69 địa chỉ này giúp ta kết nối nhiều module, cảm biến khác nhau trên cùng 1 đường dữ liệu.

- Thiết lập các giá trị và mảng lưu giá trị thanh ghi từ module I2C.

- Thiết lập tần số I2C đến 400 kHz TWBR = ((F\_CPU / 400000UL) - 16) / 2.

***Thuật toán đọc giá trị từ cảm biến***:

- Kiểm tra khoảng thời gian Arduino hoạt động đến thời gian đặt thì lấy mẫu cảm biến 1 lần.

- Kiểm tra có giá trị trả về từ module I2C về không

- Cập nhập giá trị trả về vào mảng i2cData và trích xuất các giá trị gia tốc theo phương X biến AcX, gia tốc theo phương Y biến AcY, vận tốc con xoay hồi chuyển trục Y biến Gyro. Ở đây chỉ cần quan tâm các giá trị cần thiết để tính góc pitch, đó chính là góc nghiêng của xe.

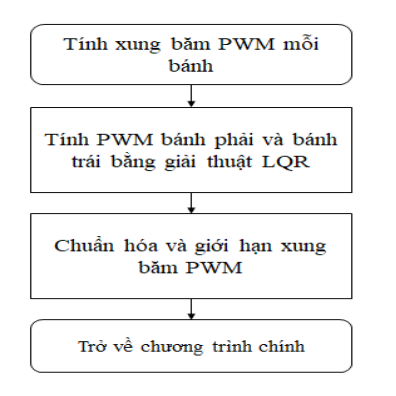
- Cập nhập thời gian chu kì đọc dt là thời gian từ lúc đọc tới lúc nhận được kết quả (quy ra giây).

- Tính toán giá trị góc pitch và tỉ lệ vận tốc con xoay hồi chuyển Gyrorate .

- Sử dụng hàm kalman.getAngle(pitch, Gyrorate,dt) trong thư viện Kalman.h để trả về giá trị góc nghiêng. - Giới hạn vùng hoạt động của xe: Nếu góc nghiêng trả về trong khoảng từ -20 đến thì xe sẽ điều khiển cân bằng, ngoài khoảng đó thì ngắt động cơ dừng điều khiển xe.

Sử dụng hàm Kalman lọc nhiễu ước lượng giá trị góc nghiêng trả về phẳng, ít gai hơn. Từ đó ổn định được tín hiệu điều khiển cho hệ thống, giúp hệ thống hoạt động mượt mà hơn

***4.2.4 Thuật toán tính xung băm PWM mỗi bánh với giải thuật LQR và điều khiển động cơ***



Hình 25 Lưu đồ thuật toán tính xung băm PWM mỗi bánh với giải thuật LQR

***Thuật toán tính xung băm PWM mỗi bánh***:

Từ ma trận K tính toán được ở chương cơ sở lý thuyết, ma trận K sẽ có dạng:

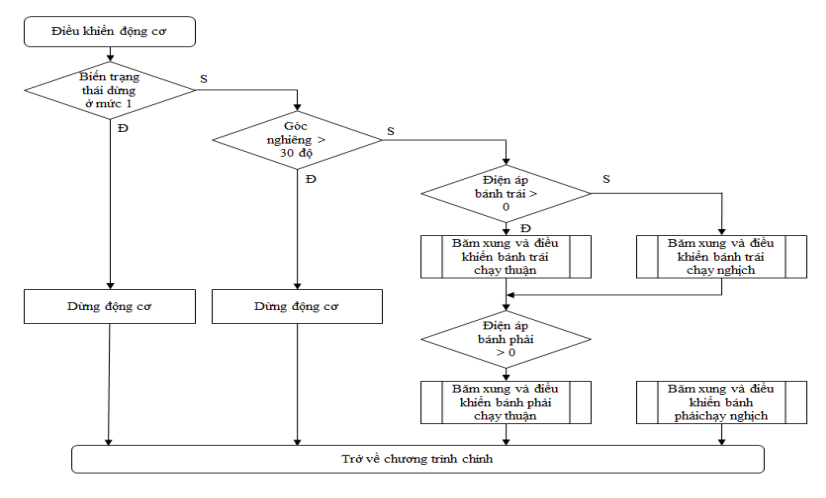
Và tín hiệu điều khiển:

Theo giải thuật LQR thì tín hiệu điều khiển cấp cho động cơ là:

Suy ra tín hiệu điều khiển cho bánh trái và bánh phải lần lượt là:

Từ tín hiệu điều khiển, chuẩn hoá và giới hạn giá trị xung băm PWM: Khi động cơ nghiêng 1 góc là 20 độ thì sẽ băm xung tối đa (tuỳ theo tần số băm xung cầu H sẽ có một mức giới hạn riêng).

***Thuật toán điều khiển động cơ:***



Hình 26 Lưu đồ thuật toán điều khiển động cơ

**4.3 Thiết kế GUI**

***4.3.1 Thiết kế giao diện giám sát***

- Chuẩn bị: Phần mềm Visual Studio 2022, thư viện ZedGraph (dùng để vẽ) và System.IO.Ports (dùng để kết nối các ports và các com)

- Các phần trong chương trình:

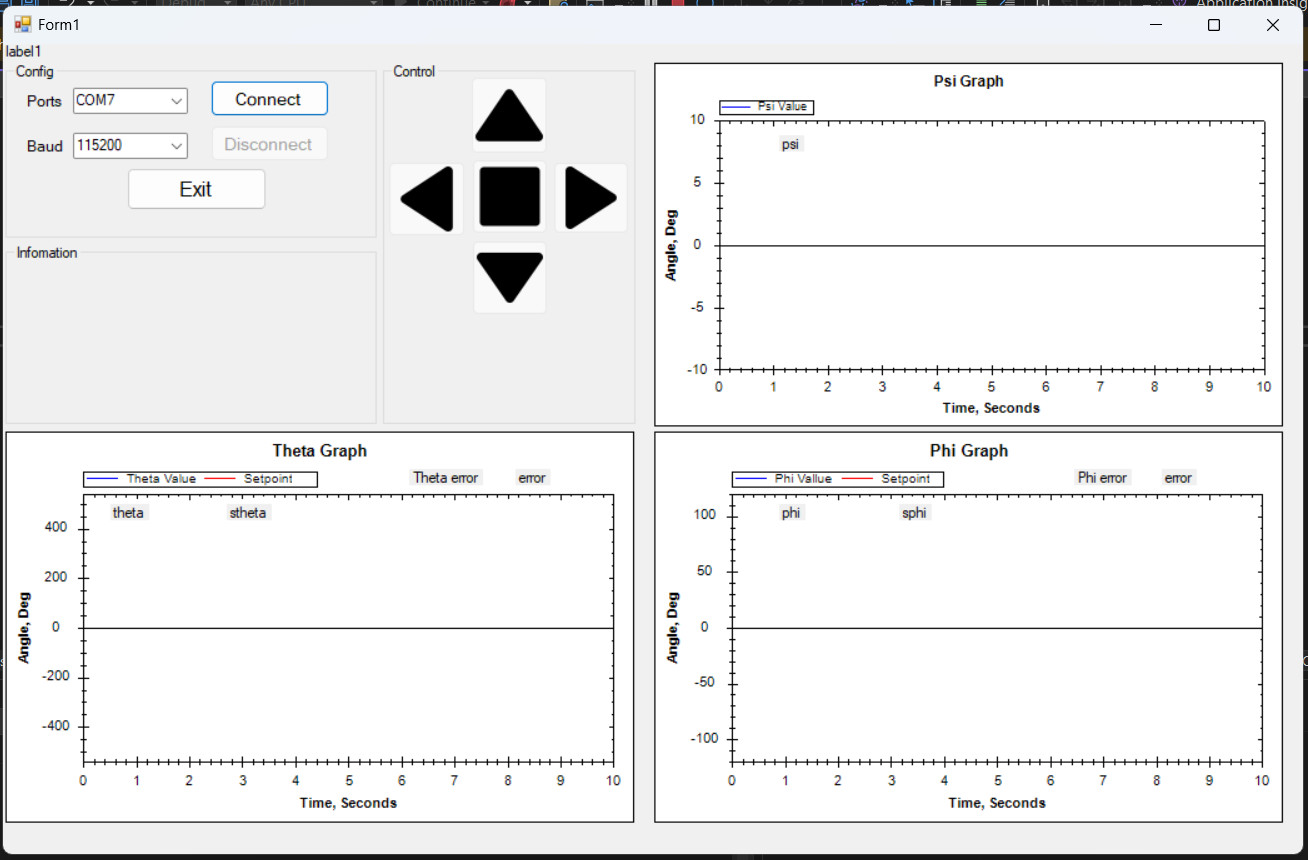
+ Thiết lập giao diện form1 với nút nhận kết nối từ port và com

+ Thiết lập timer1 để đếm thời gian chính xác cho đồ thị

+ Thiết lập các giá trị hiển thị bao gồm: Baud (tốc độ truyền), port (kết nối với các com), các giá trị đặt ban đầu cho baud, 3 đồ thị cho psi (góc nghiêng), theta (góc xoay bánh xe) và phi (góc xoay thân xe)

+ Thiết lập các hàm để vẽ từ những thông số được truyền từ xe đến máy tính

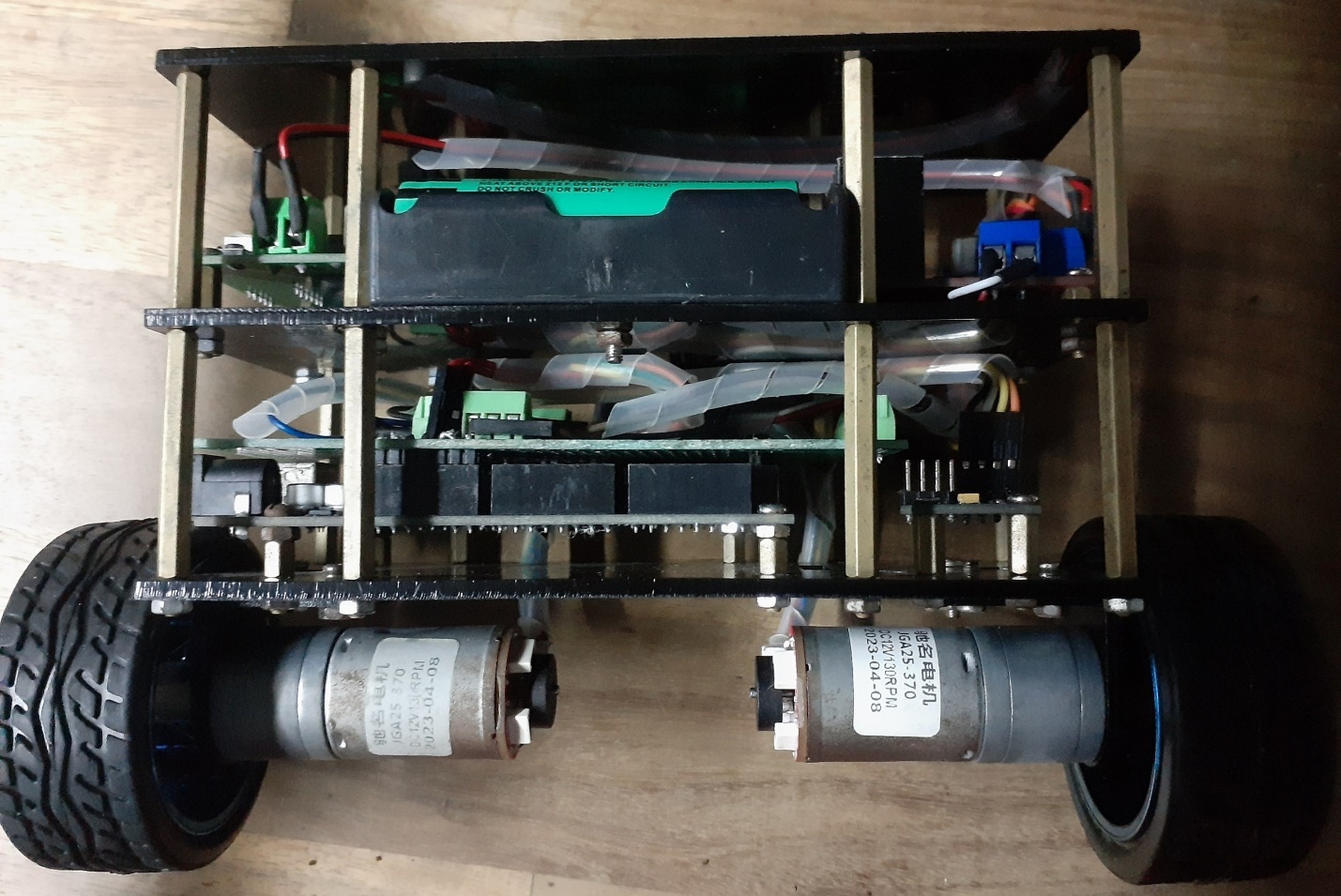
***4.3.2 Kết quả giao diện***



Hình 27 Giao diện máy tính giao tiếp với mô hình

**CHƯƠNG 5 KẾT QUẢ THỰC HIỆN**

**5.1 Kết quả thực tế**



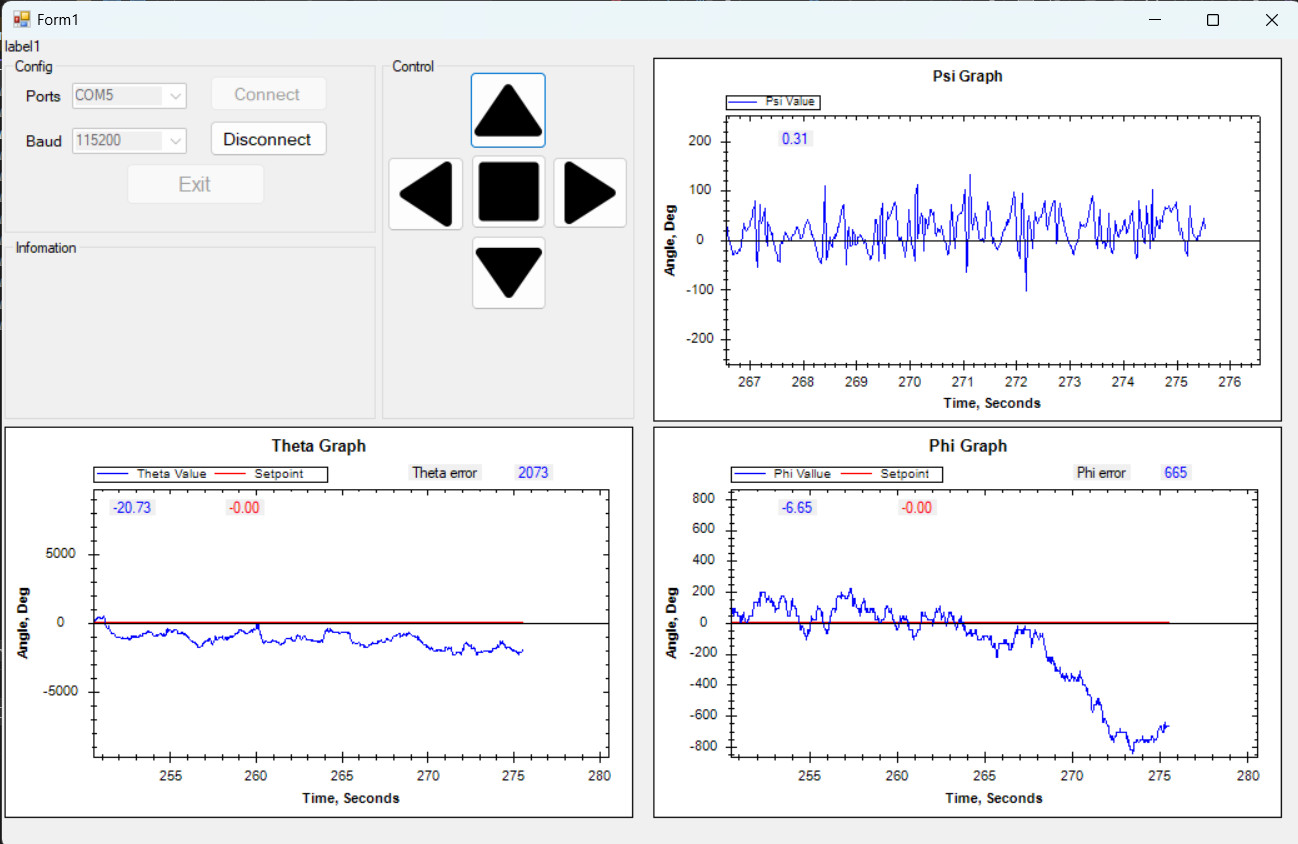
Hình 28 Mô hình phần cứng thi công thực tế hướng trực diện

**

Hình 29 Mô hình phần cứng thi công thực tế hướng nằm ngang

**

Hình 30 Mô hình phần cứng thi công thực tế hướng nhìn từ trên xuống



Hình 31 Kết quả thu được từ đồ thị

**5.2 Kết luận**

Qua quá trình nghiên cứu và thực hiện đề tài nhóm xin tổng hợp lại những thành quả đã đạt được:

- Thiết kế hoàn chỉnh và hoàn thiện mô hình xe hai bánh tự cân bằng.

- Tính toán mô hình toán học, hàm trạng thái cho hệ thống.

- Thực hiện mô phỏng hệ thống thành công trên MATLAB Simulink.

- Các thành phần điện và vi điều khiển hoạt động tốt:

+ Giá trị trả về từ cảm biến góc nghiêng tương đối chính xác.

+ Điều chế độ rộng xung cho mạch cầu H tốt, ổn định không bị quá tải, quá nhiệt trong quá trình hoạt động.

+ Đọc bộ mã hoá xoay encoder tương đối chính xác.

+ Vi điều khiển hoạt động ổn định, xử lý tín hiệu nhanh.

- Mô hình có thể cân bằng tại chỗ và di chuyển theo tín hiệu điều khiển.

**5.3 Hướng phát triển của đề tài**

- Thiết kế bộ điều khiển với bộ trọng số tối ưu nhất để hệ thống vận hành mượt mà, hạn chế được nhiễu tối đa.

- Khảo sát và thiết kế cho hệ xe hai bánh tự cân bằng trên các bề mặt môi trường như dốc nghiêng, gồ ghề, trơn trượt, ...

- Phát triển thêm nhiều chức năng cho hệ như mang vác vật, bám đối tượng, bám line,...

**TÀI LIỆU THAM KHẢO**

[1 ] N. T. Hiếu, “Điều Khiển Robot Xe Hai Bánh Cân Bằng Trên Địa Hình Phẳng” Đại Học Bách Khoa TP. HCM.

[2 ] Nguyễn Trị Phương Hà; Huỳnh Thái Hoàng, “Lý Thuyết Điều Khiển Tự Động” in Lý Thuyết Điều Khiển Tự Động, 1st ed., Nguyễn Trị Phương Hà, Ed. Hồ Chí Minh City: Nhà Xuất Bản Đại Học Quốc Gia TP. HCM, 2005, pp. 311–314.

[3 ] Huỳnh Thái Hoàng, “Giải Thuật Di Truyền” in Điều Khiển Thông Minh, 1st ed., Huỳnh Thái Hoàng, Ed. Hồ Chí Minh City: Nhà Xuất Bản Đại Học Quốc Gia TP. HCM, 2006, pp. 126–129.

[4 ] Huỳnh Thái Hoàng, “Bộ Điều Khiển Mờ” in Hệ Thống Điều Khiển Thông Minh, Huỳnh Thái Hoàng, Ed. Hồ Chí Minh City: Đại Học Quốc Gia TP. Hồ Chí Minh, 2006, pp. 25–60.

[5 ] Nguyễn Trị Phương Hà, “Phương Pháp Toàn Phương Tuyến Tính LQR” in Lý Thuyết Điều Khiển Hiện Đại, Nguyễn Trị Phương Hà, Ed. Hồ Chí Minh City: Đại Học Quốc Gia TP.Hồ Chí Minh, 2009, pp. 154–170.

[6 ] R. S. A. A.-W. Murtadha Awoda, “Parameter Estimation of a Permanent Magnetic DC Motor” Iraqi Journal for Electrical And Electronic Engineering, Iraq, pp. 28–36, Jun. 2019.

[7 ] HELLMAN, “Two-Wheeled Self-Balancing Robot Design and control based on the concept of an inverted pendulum” Bachelor’s Thesis in Mechatronics, pp.1- 14, 2015-05-20

[8] Nguyễn Thanh Tần (2017). Ứng dụng giải thuật di truyền chỉnh định ma trận tối ưu LQR cho hệ con lắc ngược phi tuyến. *Tạp chí khoa học Trường Đại học Trà Vinh*, 28, 50 - 55

[9] Hồ Đắc Lộc, Nguyễn Hùng, Trương Ngọc Sơn (2012). Điều khiển ổn định cho robot hai bánh tự cân bằng. *Tạp Chí Khoa Học Giáo Dục Kỹ Thuật*, 22, 23 - 27