

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM**



TRẦN ANH TỨ

**NGHIÊN CỨU, THIẾT KẾ, THỬ NGHIỆM XE
HAI BÁNH TỰ CÂN BẰNG**

LUẬN VĂN THẠC SĨ

Chuyên ngành: Kỹ thuật Cơ điện tử

Mã số ngành: 60520114

CÁN BỘ HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN DUY ANH

TP. HỒ CHÍ MINH, tháng 01 năm 2014

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM**

Cán bộ hướng dẫn khoa học: TS. NGUYỄN DUY ANH
Giảng viên trường Đại học Bách Khoa TP HCM

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP.HCM
ngày 10 tháng 5 năm 2014

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

TT	Họ và tên	Chức danh Hội đồng
1	PGS.TS. Nguyễn Tấn Tiến	Chủ tịch
2	TS. Nguyễn Thanh Phương	Phản biện 1
3	TS. Nguyễn Hùng	Phản biện 2
4	TS. Võ Hoàng Duy	Ủy viên
5	TS. Võ Đình Tùng	Ủy viên, Thư ký

Xác nhận của Chủ tịch Hội đồng đánh giá Luận sau khi Luận văn đã được
sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

PGS.TS. Nguyễn Tấn Tiến

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM
PHÒNG QLKH – ĐTSĐH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

TP. HCM, ngày..... tháng..... năm 20.....

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: .Trần Anh Tứ

Giới tính: Nam

Ngày, tháng, năm sinh: 28/9/1971

Nơi sinh: Bến Tre

Chuyên ngành: Kỹ thuật cơ điện tử

MSHV:1241840021

I-Tên đề tài:

NGHIÊN CỨU, THIẾT KẾ, THỬ NGHIỆM XE HAI BÁNH TỰ CÂN BẰNG

II-Nhiệm vụ và nội dung:

- Nghiên cứu thiết kế phần cơ khí và mạch điện phù hợp với xe hai bánh tự cân bằng
- Nghiên cứu cài đặt, lập trình CCS và Matlab/Simulink cho DSP C2000 để thực hiện việc thu thập dữ liệu và điều khiển hệ thống
- Nghiên cứu giải thuật điều khiển phù hợp để xe tự giữ cân bằng, có thể chạy tới, chạy lui, quẹo trái, quẹo phải.

III-Ngày giao nhiệm vụ: ngày 12 tháng 6 năm 2013

IV-Ngày hoàn thành nhiệm vụ: ngày.....tháng.....năm.....

V-Cán bộ hướng dẫn: Tiến sĩ NGUYỄN DUY ANH

Giảng viên Trường Đại học Bách Khoa TP HCM

CÁN BỘ HƯỚNG DẪN

KHOA QUẢN LÝ CHUYÊN NGÀNH

TS. NGUYỄN DUY ANH

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

Trần Anh Tứ

LỜI CẢM ƠN



Trong suốt quá trình thực hiện đề tài, mặc dù gặp phải nhiều khó khăn nhưng được sự giúp đỡ, hướng dẫn từ quý Thầy,Cô và các bạn nên Luận văn tốt nghiệp Thạc sĩ đã hoàn thành đúng tiến độ. Tôi xin chân thành cảm ơn thầy **TS.Nguyễn Duy Anh** đã tận tình hướng dẫn, chỉ bảo kinh nghiệm quý báu cũng như tạo mọi điều kiện thuận lợi trong suốt quá trình tìm hiểu, nghiên cứu đề tài.

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

Bên cạnh đó, Tôi cũng xin cảm ơn các bạn trong lớp cao học đã có những ý kiến đóng góp, bổ sung, động viên giúp đỡ Tôi hoàn thành tốt đề tài.

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

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

Sau cùng Tôi xin chúc quý Thầy,Cô sức khỏe, thành công và tiếp tục đào tạo những sinh viên giỏi đóng góp cho đất nước. Chúc các bạn sức khỏe, học tập thật tốt để không phụ công lao các Thầy Cô đã giảng dạy. Tôi xin chân thành cảm ơn.

Trân trọng!

Trần Anh Tú

TÓM TẮT

Luận văn trình bày cách thức chế tạo phần cứng một mô hình xe hai bánh tự cân bằng, bao gồm cả phần mạch điện và cơ khí. Ngoài ra, luận văn còn trình bày cách thức xây dựng bộ điều khiển PID và bộ điều khiển LQR cho hệ xe hai bánh tự cân bằng trên. Kết quả điều khiển PID và LQR được mô phỏng chạy tốt trong môi trường Matlab/Simulink.

Mặt khác, mô hình thực cũng được kiểm chứng hoạt động tốt với giải thuật PID. Phần cứng được lập trình điều khiển bằng chương trình Matlab/Simulink liên kết CCS, ứng dụng cho chip TMS320F28335. Từ các kết quả có được từ mô phỏng và thực nghiệm, tác giả đưa ra các nhận xét về ưu khuyết điểm của mỗi loại giải thuật điều khiển: PID và LQR.

ABSTRACT

This thesis represents the methods of creating a hardware of a two-wheeled self-balancing cart, including electronic and mechanical elements. Moreover, this thesis also represents methods of building a PID controller and LQR controller for a two-wheeled self-balancing cart. The controlling results were simulated well in Matlab/Simulink.

Moreover, empirical model is also proved to work well with PID controller. Software is Matlab/Simulink linking to CCS. The chip TMS320F28335 was used. From experiments and simulations, I give some comments about advantages and disadvantages of each PID and LQR controller.

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN	ii
TÓM TẮT	iii
MỤC LỤC	v
DANH MỤC CÁC TỪ VIẾT TẮT	vii
DANH MỤC CÁC BẢNG	viii
DANH MỤC HÌNH ẢNH	ix
MỞ ĐẦU	1
CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI	
1.1 ĐẶT VẤN ĐỀ	4
1.2 TÌNH HÌNH NGHIÊN CỨU ROBOT 2 BÁNH TỰ CÂN BẰNG HIỆN NAY ..	6
1.2.1 Các mô hình robot 2 bánh tự cân bằng trong phòng thí nghiệm	6
1.2.2 Một số sản phẩm thực tế dựa trên mô hình robot 2 bánh tự cân bằng	8
1.2.3 Tình hình nghiên cứu robot hai bánh tự cân bằng trong nước	11
1.3 GIỚI HẠN ĐỀ TÀI	11
CHƯƠNG 2 CÁC CƠ SỞ LÝ THUYẾT	
2.1 ĐẶC TÍNH ĐỘNG LỰC HỌC	13
2.1.1 Mô hình hóa robot 2 bánh tự cân bằng trên địa hình phẳng	13
2.1.2 Mô hình hóa robot trong Simulink	18
2.2 GIỚI THIỆU VỀ BỘ LỌC KALMAN ĐỌC CẢM BIẾN ĐỘ NGHIÊNG	19
2.3 GIẢI THUẬT ĐIỀU KHIỂN	20
2.3.1 Cấu trúc bộ điều khiển PID cho robot hai bánh tự cân bằng	20
2.3.2 Bộ điều khiển LQR	21
2.3.3 Các thành phần chính của mô hình	26
CHƯƠNG 3 THIẾT KẾ HỆ THỐNG	
3.1 THIẾT KẾ PHẦN CỨNG MÔ HÌNH ROBOT HAI BÁNH TỰ CÂN BẰNG .	32
3.1.1 Thiết kế cơ khí	32
3.1.2 Cấu trúc điều khiển phần cứng	34

3.2 THIẾT KẾ PHẦN MỀM.....	35
3.2.1 Lưu đồ giải thuật.....	35
3.2.2 Bộ điều khiển nhúng cho robot 2 bánh tự cân bằng	39
CHƯƠNG 4 KẾT QUẢ MÔ PHỎNG	
4.1 GIẢI THUẬT PID	41
4.1.1 Chương trình mô phỏng.....	41
4.1.2 Kết quả mô phỏng.....	41
4.1.3 Nhận xét.....	43
4.2 GIẢI THUẬT LQR	44
4.2.1 Chương trình mô phỏng.....	44
4.2.2 Kết quả mô phỏng.....	44
4.2.3 Nhận xét.....	50
CHƯƠNG 5 KẾT QUẢ THỰC NGHIỆM	
5.1 ĐIỀU KHIỂN PID.....	ERROR! BOOKMARK NOT DEFINED.
5.1.1 Điều khiển đứng yên tại vị trí cân bằng.....	52
5.1.2 Nhận xét.....	53
5.2 ĐIỀU KHIỂN LQR	ERROR! BOOKMARK NOT DEFINED.
5.2.1 Điều khiển đứng yên tại vị trí cân bằng.....	Error! Bookmark not defined.
5.2.2 Nhận xét	55
5.2.3 Điều khiển vị trí đặt khác 0	56
5.2.4 Nhận xét.....	Error! Bookmark not defined.
5.3 CHƯƠNG TRÌNH THU NHẬP DỮ LIỆU.....	64
CHƯƠNG 6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	
6.1 KẾT LUẬN.....	67
6.2 HƯỚNG PHÁT TRIỂN	68
TÀI LIỆU THAM KHẢO.....	67

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Tiếng Việt
LQR	Linear Quadratic Regulator	Điều khiển tối ưu
PID	Propotional Integral Derivative	Vị tích phân tỉ lệ
IMU	Inertial Magnetic Unit	Cảm biến từ trường
PWM	Pulse Width Modulation	Điều rộng xung

DANH MỤC BẢNG BIỂU

Bảng 2.1: Ký hiệu và ý nghĩa của các đại lượng	14
Bảng 3.1: Các khối chức năng sử dụng trong chương trình	40

DANH MỤC HÌNH ẢNH

Hình 1.1: Robot dạng 3 bánh xe di chuyển trên địa hình bằng phẳng.....	4
Hình 1.2: Robot dạng 3 bánh xe khi xuống dốc	5
Hình 1.3: Robot dạng 3 bánh xe khi lên dốc	5
Hình 1.4: Robot 2 bánh di chuyển trên các địa hình khác nhau theo hướng bảo toàn sự thăng bằng.....	6
Hình 1.5: nBot	7
Hình 1.6: JOE	8
Hình 1.7: NXTway-GS của LEGO MINDSTORMS.....	8
Hình 1.8: Xe Segway I2, I2 cargo, X2 Adventure	9
Hình 1.9: Xe Winglet.....	10
Hình 1.10: Xe Iswing.....	9
Hình 2.1: Mô hình robot 2 bánh tự cân bằng trên mặt phẳng.....	13
Hình 2.2: Mô hình phi tuyến của robot hai bánh tự cân bằng trong Matlab Simulink.....	18
Hình 2.3: Bên trong khối Two Wheeled Balancing Robot (Non-Linear Model)....	18
Hình 2.4: Bên trong khối “DeCoupling”	19
Hình 2.5: Cấu trúc bộ điều khiển PID cho hệ robot hai bánh tự cân bằng.....	20
Hình 2.6: Sơ đồ bộ điều khiển LQR.	22
Hình 2.7: Sơ đồ chi tiết bộ điều khiển LQR cho hệ xe hai bánh tự cân bằng	26
Hình 2.8: Nguồn cấp.....	Error! Bookmark not defined.
Hình 2.9: DSP TMS320F28335	27
Hình 2.10: IMU MPU6050.....	28
Hình 2.11: Sơ đồ nguyên lý mạch cầu H.....	29
Hình 2.12: Mạch cầu H thực tế.....	30
Hình 2.13: Nguyên lý hoạt động bộ đọc nhân 4.....	31
Hình 2.14: Động cơ	31
Hình 3.1: Mô hình robot thực tế	32

Hình 3.2: Sơ đồ kết nối phần cứng	Error! Bookmark not defined.
Hình 3.3: Lưu đồ giải thuật điều khiển PID	35
Hình 3.4: Sơ đồ khối bộ điều khiển PID	36
Hình 3.5: Cấu trúc bộ điều khiển PID cho hệ robot hai bánh tự cân bằng.....	37
Hình 3.6: Sơ đồ khối bộ điều khiển LQR.....	38
Hình 3.7: Thư viện Target Support Package cho DSP F28335.....	39
Hình 4.1: Sơ đồ lập trình thực cho bộ điều khiển PID	41
Hình 4.2: Sơ đồ khối điều khiển PID	41
Hình 4.3: Tín hiệu vị trí góc quay của bánh xe θ (rad)	42
Hình 4.4: Tín hiệu góc nghiêng của bánh xe ψ (rad)	42
Hình 4.5: Tín hiệu điện áp điều khiển (Volt)	43
Hình 4.6: Chương trình lập trình LQR	44
Hình 4.7: Góc bánh xe θ (rad).....	45
Hình 4.8: Góc nghiêng ψ (rad).....	45
Hình 4.9: Góc xoay ϕ (rad).....	46
Hình 4.10: Điện áp cấp cho động cơ phải (Volt).....	47
Hình 4.11: Điện áp cấp cho động cơ trái (Volt)	47
Hình 4.12: Góc bánh xe θ (rad)	48
Hình 4.13: Góc nghiêng ψ (rad)	49
Hình 4.14: Góc xoay ϕ (rad).....	49
Hình 4.15: Tín hiệu điện áp cấp cho động cơ trái (volt)	50
Hình 4.16: Tín hiệu điện áp cấp cho động cơ bánh phải (volt)	50
Hình 5.1: Chương trình thực tế điều khiển PID hệ thống	52
Hình 5.2: Khối điều khiển PID.....	52
Hình 5.3: Khối phân tích các thành phần psi, teta, phi.....	53
Hình 5.4: Khối phân tích để cho ra các thành phần phi và teta	53
Hình 5.5: Góc teta (degree)	54
Hình 5.6: Góc psi (degree).....	54
Hình 5.7: Góc phi (degree)	54

Hình 5.10: Điện áp cấp cho hai động cơ (volt).....	54
Hình 5.11: Chương trình điều khiển quẹo	Error! Bookmark not defined.
Hình 5.12: Góc tới teta (độ).....	Error! Bookmark not defined.
Hình 5.13: Góc lệch psi (độ)	Error! Bookmark not defined.
Hình 5.14: Góc xoay phi (độ)	Error! Bookmark not defined.
Hình 5.15: Điện áp cấp cho động cơ trái	Error! Bookmark not defined.
Hình 5.16: Điện áp cấp cho động cơ phải	Error! Bookmark not defined.
Hình 5.17: Góc tới teta (độ).....	Error! Bookmark not defined.
Hình 5.18: Góc nghiêng psi (độ)	Error! Bookmark not defined.
Hình 5.19: Góc xoay phi (độ)	Error! Bookmark not defined.
Hình 5.20: Điện áp cấp cho động cơ trái	Error! Bookmark not defined.
Hình 5.21: Điện áp cấp cho động cơ phải	Error! Bookmark not defined.
Hình 5.22: Chương trình thu thập dữ liệu thông qua cổng USB ...	Error! Bookmark not defined.
Hình 5.23: Chương trình giao diện điều khiển	68

MỞ ĐẦU

1. Lý do chọn đề tài

Hệ thống xe hai bánh tự cân bằng là một đối tượng phi tuyến MIMO thường được dùng trong các phòng thí nghiệm để kiểm chứng các giải thuật điều khiển. Ngoài ra, hệ thống trên cũng đã được ứng dụng thành công trong thực tế, phát triển thương mại ở các nước đang phát triển. Tuy nhiên, ở Việt Nam, đề tài trên còn khá mới mẻ, nhiều thách thức. và tính chất MIMO của hệ xe hai bánh tự cân bằng là khó điều khiển vì chỉ với hai động cơ, ta phải thực hiện điều khiển thỏa mãn cả vị trí xe đi tới, quẹo, đứng yên mà không ngã, tức hệ thống của ta là hệ hai vào, ba ra.

2. Mục tiêu nghiên cứu của đề tài

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

- Tìm hiểu các mô hình xe, robot 2 bánh tự cân bằng và các nguyên lý cơ bản về cân bằng.
- Tính toán các thông số động lực học, xây dựng các hàm không gian-trạng thái (state-space) của mô hình.
- Tìm hiểu, lựa chọn các loại cảm biến và bộ điều khiển trung tâm. Trong đề tài này sẽ sử dụng cảm biến IMU 9 DOF và bộ điều khiển DSP F28335.
- Mô phỏng mô hình trên Matlab Simulink cho các giải thuật điều khiển PID và LQR.
- Tìm hiểu và ứng dụng bộ lọc Kalman để đọc cảm biến góc nghiêng, xây dựng các thuật toán bù trừ để có giá trị góc chính xác.
- Xây dựng thuật toán điều khiển động cơ, giữ thăng bằng cho robot.

3. Đối tượng và phạm vi nghiên cứu

Đảm bảo đối tượng là xe 02 bánh tự cân bằng và di chuyển trên mặt phẳng và có các tính chất điều khiển như ta mong muốn.

Phạm vi nghiên cứu của đề tài dùng các giải thuật điều khiển là rất quan trọng. Có nhiều giải thuật điều khiển được đề cập tới với nhiều đối tượng phi tuyến khác

nhau. Các giải thuật điều khiển thường được sử dụng là điều khiển tuyến tính, điều khiển phi tuyến và điều khiển thông minh... Đa số các bộ điều khiển trong thực tế đều là điều khiển tuyến tính (PID, LQR...) và đều cho đáp ứng tốt. Một số bộ điều khiển tuyến tính đòi hỏi phương trình toán học hệ thống (LQR, đặt cực...), một số bộ điều khiển tuyến tính thì không yêu cầu (PID...). Như vậy, đối với đối tượng điều khiển phi tuyến như hệ xe hai bánh tự cân bằng thì giải thuật điều khiển tuyến tính có đáp ứng tốt không? Đáp ứng tốt ở mức độ như thế nào? Các luật điều khiển tuyến tính có ưu khuyết điểm so với nhau như thế nào? Đó là những câu hỏi mà luận văn muốn hướng đến để giải quyết các vấn đề thắc mắc trên.

Trong khuôn khổ luận văn này, mục tiêu điều khiển là thực hiện các giải thuật PID và LQR trên đối tượng hệ thống xe hai bánh tự cân bằng. Các kết quả kiểm chứng được trình bày trên môi trường mô phỏng Matlab/Simulink và có kết quả thực tế để kiểm chứng.

4. Phương pháp nghiên cứu

4.1 Phương pháp lý thuyết

Thông qua tìm hiểu từ sách vở, các nguồn tài liệu trên internet, học viên nghiên cứu về phương pháp PID và LQR trong điều khiển các đối tượng phi tuyến, đặc biệt là hệ thống xe hai bánh tự cân bằng. Việc nắm rõ phương pháp bao gồm hiểu về lý thuyết hình thành, xây dựng bộ điều khiển, cách thức tinh chỉnh thông số bộ điều khiển, dạng đáp ứng ngõ ra tương ứng...

Kết quả xây dựng được thử nghiệm trên phần mềm mô phỏng. Trong khuôn khổ luận văn, học viên sử dụng phần mềm Matlab/Simulink.

4.2 Phương pháp thực nghiệm

Thực hiện chế tạo cơ khí, mạch điện để kiểm chứng trên mô hình thực. Qua đáp ứng thực của hệ thống, học viên đưa ra các nhận xét, kết luận về mỗi phương pháp.

5. Kết cấu luận văn

Chương 1: “ **Tổng quan về đề tài** ” Trình bày khái quát tình hình nghiên cứu Robot 02 bánh tự cân bằng hiện nay, các mô hình Robot 02 bánh tự cân bằng trong

phòng thí nghiệm và một số sản phẩm thực tế và tình hình nghiên cứu Robot xe 02 bánh tự cân bằng trong nước. Nêu lên giới hạn của đề tài.

Chương 2 : “**Cơ sở lý thuyết**” Tìm hiểu về đặc tính động lực học như mô hình hóa robot 2 bánh tự cân bằng trên địa hình phẳng, mô hình hóa trên Simulink, từ đó tìm ra mối quan hệ giữa điện áp điều khiển hai động cơ qua phương trình động lực học, giới thiệu về bộ lọc Kalman ứng dụng trong đọc cảm biến góc nghiêng cho IMU, với giải thuật lọc Kalman đã được tích hợp bên trong chương trình đọc cảm biến độ nghiêng giao tiếp giữa Arduino và IMU MPU 6050. Trình bày giải thuật điều khiển PID và LQR và giới thiệu các thành phần chính của mô hình như: Nguồn, DSP TMS320F28335, cảm biến IMU, mạch cầu H, động cơ.

Chương 3: “**Thiết kế hệ thống**” Trình bày thiết kế mô hình cứng xe 02 bánh tự cân bằng như thiết kế cơ khí, cấu trúc điều khiển phần cứng. Thiết kế phần mềm các lưu đồ giải thuật PID, LQR. Giới thiệu các khối chức năng trong chương trình Matlab của bộ điều khiển nhúng cho xe hai bánh tự cân bằng.

Chương 4: “**Kết quả mô phỏng**” dùng giải thuật điều khiển PID và mô phỏng theo lý thuyết, để có những nhận xét qua mô phỏng về góc nghiêng (ψ), góc tới (θ), góc xoay (ϕ), mà trong đó ta không điều khiển được góc xoay ϕ . Dùng giải thuật LQR và mô phỏng như PID cho các giá trị đặt, qua mô phỏng có nhận xét sự ổn định các thành teta, phi, psi, teta-dot, phi-dot, psi-dot và điện áp cấp cho từng thành phần động cơ.

Chương 5: “**Kết quả thực nghiệm**” chương trình điều khiển PID cho xe đứng yên vị trí cân bằng, điều khiển quẹo trái và phải. Nhận thấy ưu điểm của PID là quá trình thử sai các thông số điều khiển mà không cần biết phương trình toán học của hệ thống. Cũng như PID chương trình điều khiển LQR cho ta thấy là điều khiển thỏa hiệp và hệ thống phụ thuộc hoàn toàn khi ta tính toán chọn các thông số chưa được tối ưu, nên vẫn tồn tại sai số xác lập.

Chương 6: “**Kết luận và hướng phát triển**” trình bày các nhiệm vụ hoàn thành và nêu các hạn chế của đề tài, từ đó đưa ra hướng phát triển đề tài.

CHƯƠNG 1

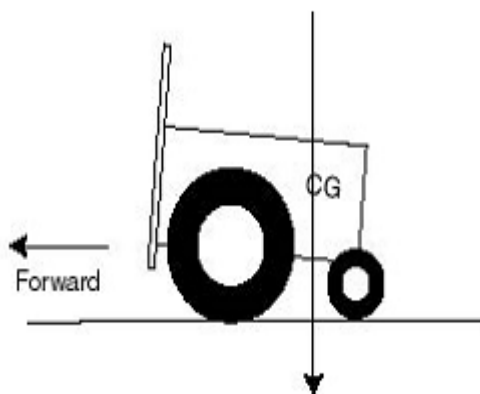
TỔNG QUAN VỀ ĐỀ TÀI

1.1 Đặt vấn đề

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

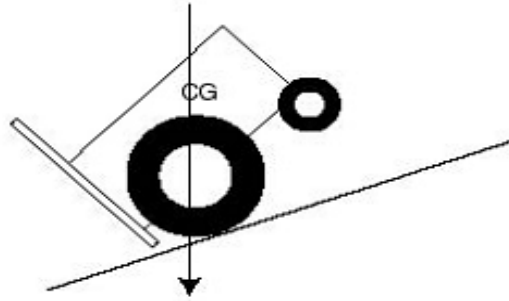
Kết quả nghiên cứu mô hình con lắc ngược cơ bản, ví dụ như mô hình xe-con lắc, con lắc ngược quay... có thể ứng dụng và kế thừa sang các mô hình tương tự khác nhưng có tính ứng dụng thực tiễn hơn, chẳng hạn như mô hình tên lửa, mô hình xe hai bánh tự cân bằng... do đó khắc phục được những nhược điểm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

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



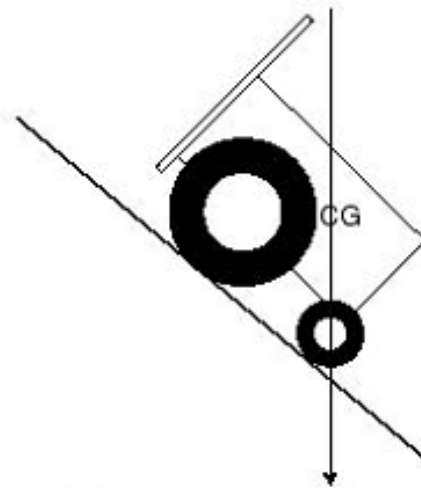
Hình 1.1: Robot dạng 3 bánh xe di chuyển trên địa hình bằng phẳng

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



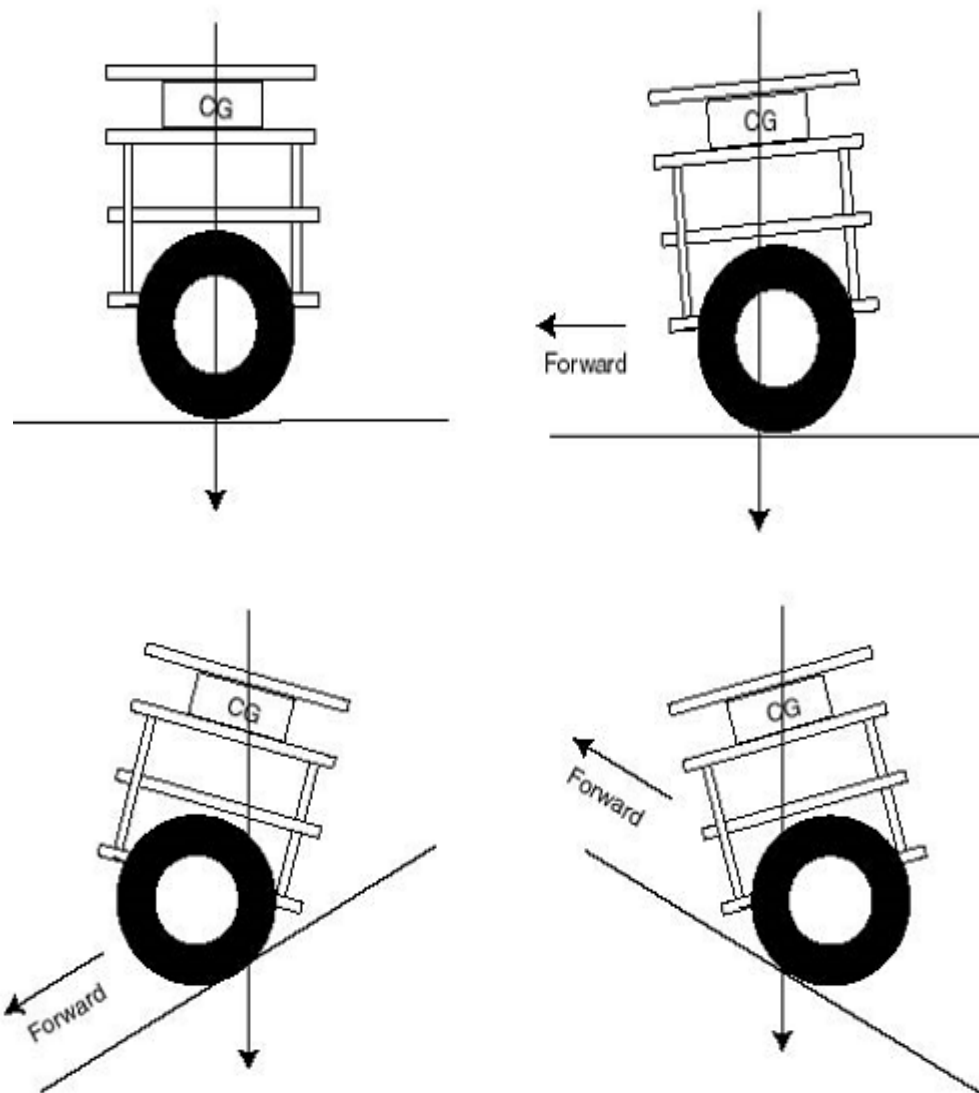
Hình 1.2: Robot dạng 3 bánh xe khi xuống dốc

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



Hình 1.3: Robot dạng 3 bánh xe khi lên dốc

Ngược lại, các robot dạng hai bánh đồng trục lại thăng bằng rất linh động khi di chuyển trên địa hình phức tạp, mặc dù bản thân robot là một hệ thống không ổn định. Khi robot di chuyển trên địa hình dốc, nó tự động nghiêng ra trước và giữ cho trọng lượng dồn về hai bánh chính. Tương tự, khi di chuyển xuống dốc, nó nghiêng ra sau và giữ trọng tâm rơi vào bánh chính. Vì vậy, không bao giờ có hiện tượng trọng tâm xe rơi ngoài vùng đỡ bánh xe để có thể gây ra lật úp.



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

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

1.2.1 Các mô hình robot 2 bánh tự cân bằng trong phòng thí nghiệm

1.2.1.1 nBot

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

Quá trình điều khiển sử dụng tín hiệu từ hai cảm biến: cảm biến góc nghiêng của thân robot so với phương của trọng lực và encoder gắn ở bánh xe để đo vị trí

robot. Tín hiệu này hình thành nên 4 biến: góc nghiêng thân robot, vận tốc góc nghiêng, vị trí robot và vận tốc robot; 4 biến này được tính toán thành điện áp điều khiển động cơ cho robot.



Hình 1.5: nBot theo [7]

1.2.1.2 JOE

JOE do phòng thí nghiệm điện tử công nghiệp của viện Công nghệ Liên bang Lausanne, Thụy Sĩ tạo ra vào năm 2002. Hình dạng của nó gồm hai bánh xe đồng trục, mỗi bánh gắn với một động cơ DC, robot này có thể chuyển động xoay theo hình chữ U.

Hệ thống điều khiển gồm hai bộ điều khiển “không gian trạng thái” (state space) tách rời nhau, kiểm soát động cơ để giữ cân bằng cho hệ thống. Thông tin trạng thái được cung cấp bởi hai encoder quang và hai cảm biến: gia tốc góc và con quay hồi chuyển (gyro).

JOE được điều khiển bởi một bộ điều khiển từ xa RC . Bộ điều khiển trung tâm và xử lý tín hiệu là một board xử lý tín hiệu số (DSP) phát triển bởi chính nhóm và của viện Federal, kết hợp với FPGA của XILINC.



Hình 1.6: JOE

1.2.1.3 NXTway-GS



Hình 1.7: NXTway-GS của LEGO MINDSTORMS

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

Xe Segway

Segway PT (viết tắt của Segway Personal Transporter - Xe cá nhân Segway), thường được gọi tắt là Segway là một phương tiện giao thông cá nhân có hai bánh,

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



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

Xe Winglet

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



Hình 1.9: Xe Winglet

Iswing Toyota:

Iswing được mệnh danh là một trong những ý tưởng táo bạo nhất của ngành công nghiệp ô tô trong thời gian gần đây. Xuất hiện lần đầu tiên tại triển lãm Tokyo 2005, Iswing là biểu tượng cho phương tiện cá nhân trong tương lai bởi nó khác xa so với ô tô thông thường. Sự di chuyển, điều khiển, giao tiếp giữa người và xe đều mang tính nhân bản. Chẳng hạn như khi đặt ghế ở chế độ "thân xe mở tối thiểu", tầm mắt tài xế sẽ ngang với người đang đứng nên rất dễ trò chuyện và tạo cảm giác thân thiện. I-swing được điều khiển thông qua tay nắm và các nút, giống với cách con người sử dụng các thiết bị điện tử hàng ngày.



Hình 1.10: Xe Iswing

1.2.3 Tình hình nghiên cứu robot hai bánh tự cân bằng trong nước

Trường Đại học Bách Khoa là trường Đại học tiên phong trong phòng trào nghiên cứu và chế tạo xe hai bánh tự cân bằng tại Việt Nam, có nhiều luận văn đại học và cao học đã thực hiện thành công mô hình xe hai bánh tự cân bằng:

Điều khiển xe hai bánh tự cân bằng theo thích nghi theo độ dốc của học viên Nguyễn Trung Hiếu [6]. Đề tài này đã thực hiện thành công giải thuật điều khiển LQR cho hệ xe tự cân bằng. Tuy nhiên, giải thuật điều khiển LQR chỉ đảm bảo xe đứng yên tại điểm làm việc. Khả năng điều khiển xe quẹo trái, quẹo phải, đi thẳng bị bỏ ngỏ.

Điều khiển xe hai bánh tự cân bằng dùng PID auto-tunning của học viên Hoàng Anh Vũ [7] đã điều khiển thành công hệ xe hai bánh tự cân bằng có thông số PID tự thay đổi thích nghi. Tuy nhiên, thông số điều khiển PID phải được chỉnh định trước và phải rất gần giá trị tối ưu để có thể nhanh chóng chỉnh về giá trị tối ưu trước khi xe bị mất cân bằng thực sự. Ngoài ra, luận văn trên cũng bỏ ngỏ khả năng điều khiển đi thẳng, quẹo trái, quẹo phải của xe.

Ngoài ra, một số đồ án tốt nghiệp của sinh viên các trường khác như đại học Sư phạm Kỹ thuật TPHCM cũng đã tiến hành chế tạo thử nghiệm nhưng chưa thành công : xe chỉ giữ thẳng bằng trong một khoảng thời gian ngắn thì bị mất cân bằng.

Đề tài học viên thực hiện là đề tài đầu tiên của trường Đại học Công nghệ TPHCM nên khó tránh khỏi sai sót. Tuy nhiên, đề tài đã thành công trong việc mô phỏng thành công hệ thống xe hai bánh tự cân bằng. Việc mô phỏng thành công trên Matlab/Simulink. Ngoài ra, xe hai bánh tự cân bằng trong khuôn khổ luận văn có khả năng tự cân bằng tại điểm làm việc rất tốt, cho dù bị tác động tương đối mạnh. Mặt khác, xe có khả năng tự quẹo trái, quẹo phải. Đây là những điều mới mẻ hơn so với các luận văn trước đó.

1.3 Giới hạn đề tài

- Mặt phẳng hoạt động của hệ xe hai bánh tự cân bằng là phẳng hoàn toàn, có độ nghiêng bằng 0 và có độ bám nhất định đủ để bánh xe không bị trượt trong quá trình hoạt động.

- Hai động cơ gắn với hai bánh có thông số giống nhau xem như hoàn toàn.
- Dao động rung của xe xem như không ảnh hưởng nhiều đến giá trị đọc về từ cảm biến độ nghiêng.
- Xem như cả chiếc xe là một hình hộp chữ nhật hoàn hảo (thực tế hình dạng xe không là một hình hộp chữ nhật hoàn hảo). Trong đó, khối lượng xe được phân bố đều trong hình hộp trên.

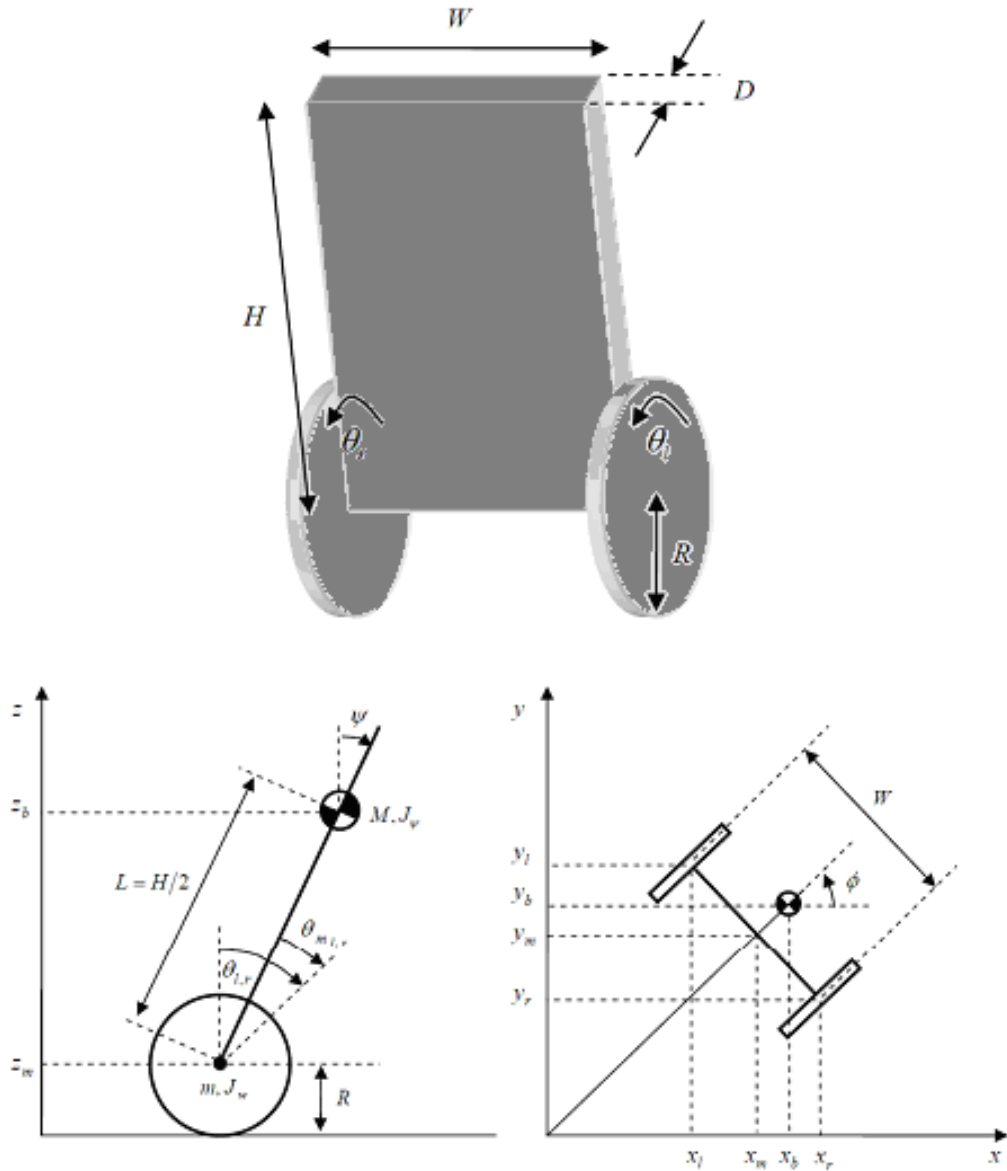
CHƯƠNG 2

CÁC CƠ SỞ LÝ THUYẾT

2.1 Đặc tính động lực học

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

Xây dựng hệ phương trình trạng thái mô tả hệ thống robot 2 bánh tự cân bằng.



ψ : góc nghiêng của xe $\theta_{l,r}$: góc tới lần lượt của bánh trái, bánh phải

Hình 2.1: Mô hình robot 2 bánh tự cân bằng trên mặt phẳng

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

Bảng 2.1: Ký hiệu và ý nghĩa của các đại lượng

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

i_l, i_r	A	Dòng điện động cơ bánh trái, phải
v_l, v_r	V	Điện áp động cơ bánh trái, phải

Theo tài liệu tham khảo [6], phương trình phi tuyến của robot hai bánh tự cân bằng được xây dựng như sau:

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

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

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

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

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

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

Tọa độ bánh trái trong hệ qui chiếu :

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

Tọa độ bánh phải trong hệ qui chiếu :

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

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

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

Phương trình động năng của chuyển động tịnh tiến:

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

Phương trình động năng của chuyển động quay :

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

Với

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

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

Phương trình thế năng:

$$U = mg z_l + mg z_r + mg z_b \quad (2.10)$$

Phương trình Lagrange :

$$L = T_1 + T_2 - U \quad (2.11)$$

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

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

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

Lấy đạo hàm L theo các biến ta được:

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

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

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

Momen động lực do động cơ DC sinh ra:

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

Và:

$$F_l = nK_t i_l + f_m(\psi - \theta_l) - f_w \theta_l \quad (2.19)$$

$$F_r = nK_t i_r + f_m(\psi - \theta_r) - f_w \theta_r \quad (2.20)$$

$$F_\psi = -nK_t i_l - nK_t i_r - f_m(\psi - \theta_l) - f_m(\psi - \theta_r) \quad (2.21)$$

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

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

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

$$i_{l,r} = \frac{v_{l,r} + K_b(\psi - \theta_{l,r})}{R_m} \quad (2.23)$$

Từ đó, các moment lực sinh ra:

$$F_\theta = \alpha(v_l + v_r) - 2(\beta + f_w)\theta + 2\beta\psi \quad (2.24)$$

$$F_\psi = -\alpha(v_l + v_r) + 2\beta\theta - 2\beta\psi \quad (2.25)$$

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

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

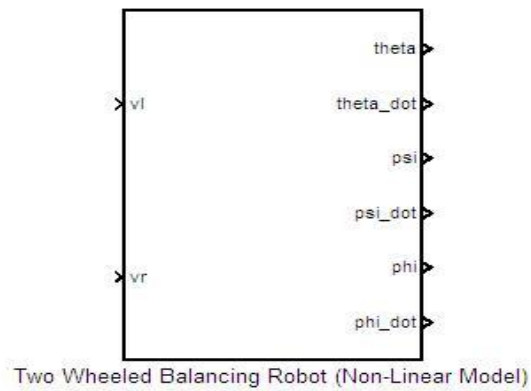
Thu được phương trình động lực học mô tả chuyển động của robot như sau:

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

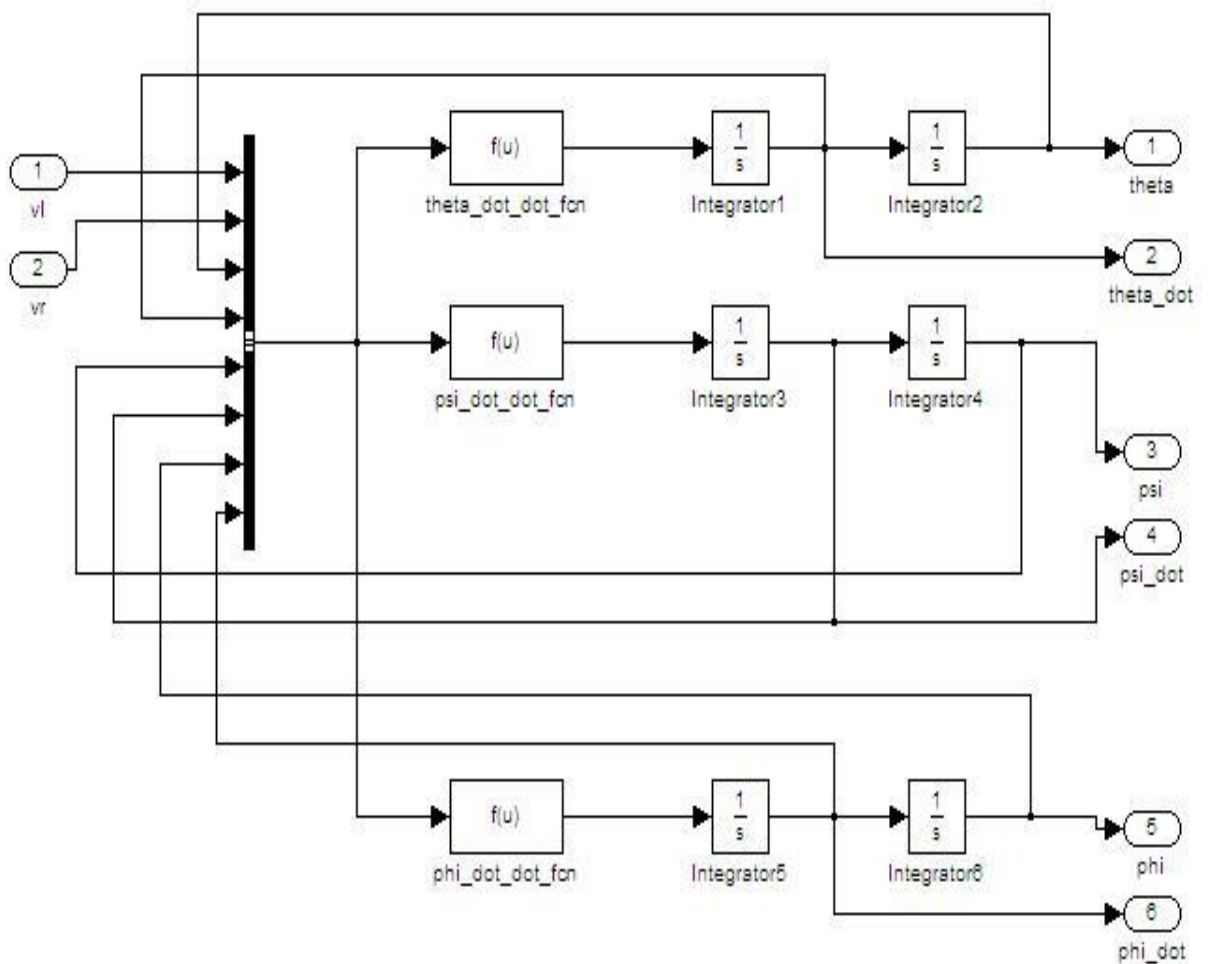
$$(MLR \cos \psi - 2n^2 J_m) \ddot{\theta} + (ML^2 + J_\psi + 2n^2 J_m) \ddot{\psi} - MgL \sin \psi - ML^2 \dot{\phi}^2 \sin \psi \cos \psi = -\alpha(v_l + v_r) + 2\beta\theta - 2\beta\psi \quad (2.29)$$

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

2.1.2 Mô hình hóa robot trong Simulink



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

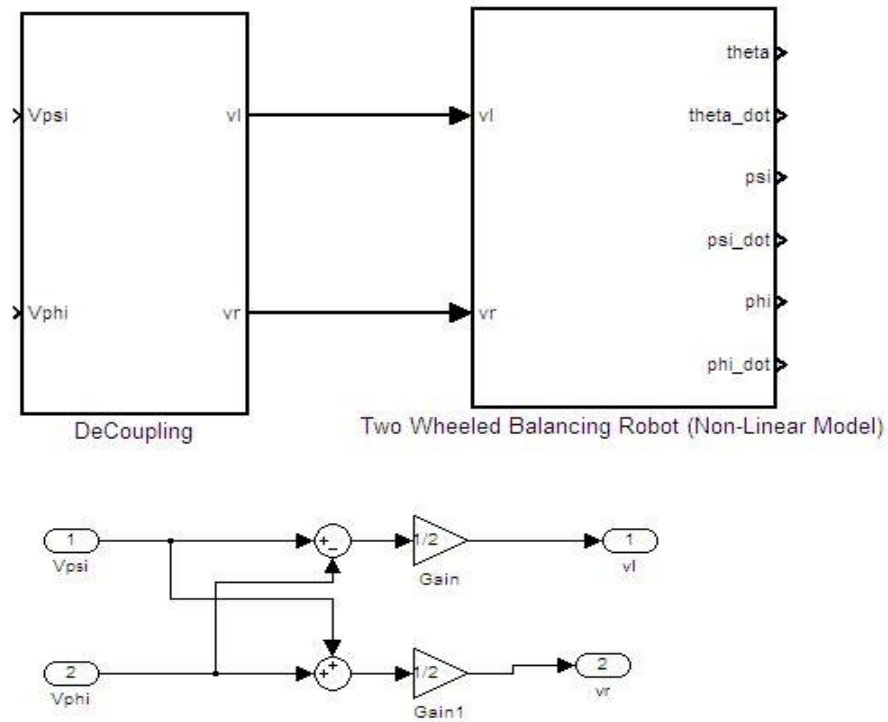


Hình 2.3: Bên trong khối Two Wheeled Balancing Robot (Non-Linear Model)

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

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

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



Hình 2.4: Bên trong khối “DeCoupling”

2.2 Giới thiệu về bộ lọc Kalman ứng dụng trong đọc cảm biến độ nghiêng

Theo nguồn Internet [9] có hỗ trợ đọc cảm biến độ nghiêng cho IMU MPU6050 truyền về cho Arduino. Mặt khác, học viên tiến hành lập trình Arduino đọc giá trị cảm biến độ nghiêng thu được từ Arduino về DSP TMS320F28335 thông qua cổng SCI-B Giải thuật lọc Kalman đã được tích hợp bên trong chương trình đọc cảm biến độ nghiêng giao tiếp giữa Arduino và IMU MPU6050.

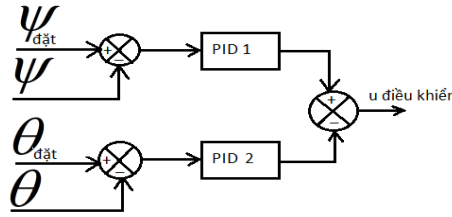
2.3 Giải thuật điều khiển

Ở đây, tác giả sử dụng cả hai giải thuật PID và LQR để cho thấy sự so sánh hai giải thuật với nhau. Giải thuật PID là giải thuật thử sai không phụ thuộc vào sự hiểu biết cấu trúc hệ thống (khó để tìm các thông số tốt, nên tác giả chỉ dùng giải thuật cho xe đứng yên). Giải thuật LQR là giải thuật ổn định dựa vào sự hiểu biết phương trình toán học, thông số cũng như cấu trúc hệ thống. Việc kiểm chứng thành công hai giải thuật trên trong mô phỏng và thực nghiệm sẽ củng cố hơn nữa hiểu biết về hai loại điều khiển trên cả về lý thuyết và thực tế. Từ đó, tác giả rút ra được các ưu điểm, khuyết điểm của hai loại điều khiển so với nhau.

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

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

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



Hình 2.5: Cấu trúc bộ điều khiển PID cho hệ robot hai bánh tự cân bằng

Hàm truyền đạt bộ điều khiển PID liên tục:

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

Rời rạc hóa đạo hàm theo thời gian:

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

Rời rạc hóa tích phân theo thời gian:

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

Phép biến đổi rời rạc (z-Transform)

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

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

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

Do đó

$$\begin{aligned} y(k) &= \frac{f(k) - f(k-1)}{T_s} \xrightarrow{\mathcal{Z}} Y[z] = \frac{F[z] - z^{-1}F[z]}{T_s} \\ \Rightarrow \frac{Y[z]}{F[z]} &= \frac{z-1}{zT_s} \end{aligned} \quad (2.57)$$

Và

$$\begin{aligned} y(k) &= y(k-1) + \frac{f(k) - f(k-1)}{2} \cdot T_s \xrightarrow{\mathcal{Z}} Y[z](1 - z^{-1}) = \frac{T_s}{2} F[z](1 + z^{-1}) \\ \Rightarrow \frac{Y[z]}{F[z]} &= \frac{T_s}{2} \frac{z+1}{z-1} \end{aligned} \quad (2.58)$$

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

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

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

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

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

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

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

Cơ sở chọn bộ điều khiển thông qua quá trình thử sai các thông số K_p , K_i , K_d theo quy tắc sau: chọn K_p , thêm K_d , K_i . Sau đó, xem xét thay đổi dần các thông số đó :

a/ Điều chỉnh K_p sao cho thời gian đáp ứng đủ nhanh(chấp nhận vọt lồ nhỏ), nếu độ vọt lồ tăng thì giảm K_p , để giảm thời gian xác lập thì tăng K_p ,

b/ Tăng K_d từ từ chọn giá trị thích hợp(có thể xuất hiện lỗi ổn định trạng thái), để hệ thống ổn định thì tăng K_d .

c/ Việc tăng K_i từ bé đến lớn để giảm lỗi mất ổn định trạng thái, đồng thời không cho vọt lố trở lại, triệt tiêu sai số xác lập thì tăng K_i .

2.3.2 Bộ điều khiển LQR

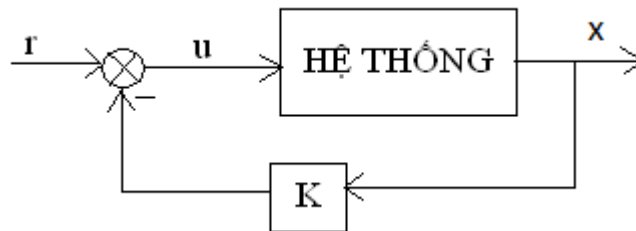
2.3.2.1 Bộ điều khiển LQR liên tục

Một hệ điều khiển được thiết kế ở chế độ làm việc tốt nhất là hệ 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, vào sự hiểu biết về đối tượng và các tác động lên đối tượng , vào điều kiện làm việc của hệ điều khiển

Xét hệ thống có tác động ngoài ($u \neq 0$):

$$\dot{x} = Ax + Bu \quad (2.63)$$

Sơ đồ điều khiển cần thiết kế:



Hình 2.6: Sơ đồ bộ điều khiển LQR.

Ma trận thông số hồi tiếp K được tính toán dựa vào ma trận A , B và hai ma trận trọng số Q và R bằng cách giải phương trình Riccati. Việc giải phương trình Riccati là rất khó khăn. Tuy nhiên, Matlab đã hỗ trợ hàm `lqr` và `dlqr` để tính toán ra ma trận K tương ứng.

Trong đó Q là ma trận xác định dương (hoặc bán xác định dương) , R là ma trận xác định dương . Trong đó, Q , R là các ma trận trọng lượng tương ứng với biến trạng thái và tín hiệu ngõ vào. Nếu muốn thành phần nào được ưu tiên đạt tối ưu khi điều khiển, ta chọn ma trận trọng lượng tương ứng thành phần đó có giá trị lớn.

Chúng ta cần tìm ma trận K của vector điều khiển tối ưu :

$$u(t) = -Kx(t)$$

thỏa mãn chỉ tiêu chất lượng J đạt giá trị cực tiểu.

Ma trận hồi tiếp trạng thái tương ứng K được tính như sau:

$$K = R^{-1} B^T S \quad (2.65)$$

Trong đó,

$$A^T S + SA - SBR^{-1}B^T S + Q = -\dot{S} \quad (2.66)$$

Khi S không thay đổi theo thời gian $\dot{S} = 0$, ta có phương trình đại số *Riccati*:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \quad (2.67)$$

2.3.2.2 Điều khiển LQR rời rạc

Xét hệ rời rạc:

$$x_{k+1} = A_k x_k + B_k u_k \quad (2.68)$$

$$\text{với } x_k \in R^n \text{ và } u_k \in R^m. \quad (2.69)$$

Nếu hệ ban đầu là hệ liên tục thì công việc của ta là rời rạc hoá hệ liên tục đó để có thể sử dụng phương pháp điều khiển LQR rời rạc.

Khi đó, luật điều khiển tối ưu của tín hiệu điều khiển có dạng:

$$u_k = -K_k x_k \quad (2.70)$$

với K_k được xác định như sau:

$$K_k = (B_k^T S_{k+1} B_k + R_k)^{-1} B_k^T S_{k+1} A_k \quad (2.71)$$

Trong đó, S_k phải thỏa mãn phương trình:

$$S_k = A_k^T (S_{k+1}^{-1} + B_k R_k^{-1} B_k^T) A_k + Q_k \quad (2.72)$$

Để xây dựng bộ điều khiển, trước tiên, ta phải biết rõ thông số hệ thống. Đưa hệ thống về dạng:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ f_2(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ f_3(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ f_4(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ f_5(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ f_6(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \end{bmatrix} + \begin{bmatrix} g_1(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ g_2(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ g_3(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ g_4(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ g_5(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \\ g_6(x_1, x_2, x_3, x_4, x_5, x_6, u_1, u_2) \end{bmatrix} \quad (2.73)$$

Lúc này, dạng phương trình hệ thống là hệ phi tuyến (các hệ thống trong tự nhiên đa phần đều là các hệ thống phi tuyến. Hệ xe hai bánh tự cân bằng có đặc trưng phi tuyến khá rõ ràng).

Sau đó, ta tiến hành tuyến tính hóa hệ thống quanh điểm cân bằng để tìm ma trận tuyến tính A, B như sau:

$$A = \begin{bmatrix} \left. \frac{\partial f_1}{\partial x_1} \right|_{x=0;u=0} & \left. \frac{\partial f_1}{\partial x_2} \right|_{x=0;u=0} & \left. \frac{\partial f_1}{\partial x_3} \right|_{x=0;u=0} & \left. \frac{\partial f_1}{\partial x_4} \right|_{x=0;u=0} & \left. \frac{\partial f_1}{\partial x_5} \right|_{x=0;u=0} & \left. \frac{\partial f_1}{\partial x_6} \right|_{x=0;u=0} \\ \left. \frac{\partial f_2}{\partial x_1} \right|_{x=0;u=0} & \left. \frac{\partial f_2}{\partial x_2} \right|_{x=0;u=0} & \left. \frac{\partial f_2}{\partial x_3} \right|_{x=0;u=0} & \left. \frac{\partial f_2}{\partial x_4} \right|_{x=0;u=0} & \left. \frac{\partial f_2}{\partial x_5} \right|_{x=0;u=0} & \left. \frac{\partial f_2}{\partial x_6} \right|_{x=0;u=0} \\ \left. \frac{\partial f_3}{\partial x_1} \right|_{x=0;u=0} & \left. \frac{\partial f_3}{\partial x_2} \right|_{x=0;u=0} & \left. \frac{\partial f_3}{\partial x_3} \right|_{x=0;u=0} & \left. \frac{\partial f_3}{\partial x_4} \right|_{x=0;u=0} & \left. \frac{\partial f_3}{\partial x_5} \right|_{x=0;u=0} & \left. \frac{\partial f_3}{\partial x_6} \right|_{x=0;u=0} \\ \left. \frac{\partial f_4}{\partial x_1} \right|_{x=0;u=0} & \left. \frac{\partial f_4}{\partial x_2} \right|_{x=0;u=0} & \left. \frac{\partial f_4}{\partial x_3} \right|_{x=0;u=0} & \left. \frac{\partial f_4}{\partial x_4} \right|_{x=0;u=0} & \left. \frac{\partial f_4}{\partial x_5} \right|_{x=0;u=0} & \left. \frac{\partial f_4}{\partial x_6} \right|_{x=0;u=0} \\ \left. \frac{\partial f_5}{\partial x_1} \right|_{x=0;u=0} & \left. \frac{\partial f_5}{\partial x_2} \right|_{x=0;u=0} & \left. \frac{\partial f_5}{\partial x_3} \right|_{x=0;u=0} & \left. \frac{\partial f_5}{\partial x_4} \right|_{x=0;u=0} & \left. \frac{\partial f_5}{\partial x_5} \right|_{x=0;u=0} & \left. \frac{\partial f_5}{\partial x_6} \right|_{x=0;u=0} \\ \left. \frac{\partial f_6}{\partial x_1} \right|_{x=0;u=0} & \left. \frac{\partial f_6}{\partial x_2} \right|_{x=0;u=0} & \left. \frac{\partial f_6}{\partial x_3} \right|_{x=0;u=0} & \left. \frac{\partial f_6}{\partial x_4} \right|_{x=0;u=0} & \left. \frac{\partial f_6}{\partial x_5} \right|_{x=0;u=0} & \left. \frac{\partial f_6}{\partial x_6} \right|_{x=0;u=0} \end{bmatrix} \quad (2.74)$$

$$B = \begin{bmatrix} \left. \frac{\partial g_1}{\partial x_1} \right|_{x=0;u=0} & \left. \frac{\partial g_2}{\partial x_1} \right|_{x=0;u=0} \\ \left. \frac{\partial g_1}{\partial x_2} \right|_{x=0;u=0} & \left. \frac{\partial g_2}{\partial x_2} \right|_{x=0;u=0} \\ \left. \frac{\partial g_1}{\partial x_3} \right|_{x=0;u=0} & \left. \frac{\partial g_2}{\partial x_3} \right|_{x=0;u=0} \\ \left. \frac{\partial g_1}{\partial x_4} \right|_{x=0;u=0} & \left. \frac{\partial g_2}{\partial x_4} \right|_{x=0;u=0} \\ \left. \frac{\partial g_1}{\partial x_5} \right|_{x=0;u=0} & \left. \frac{\partial g_2}{\partial x_5} \right|_{x=0;u=0} \\ \left. \frac{\partial g_1}{\partial x_6} \right|_{x=0;u=0} & \left. \frac{\partial g_2}{\partial x_6} \right|_{x=0;u=0} \end{bmatrix} \quad (2.75)$$

Sau đó, ta tiến hành chọn ma trận :

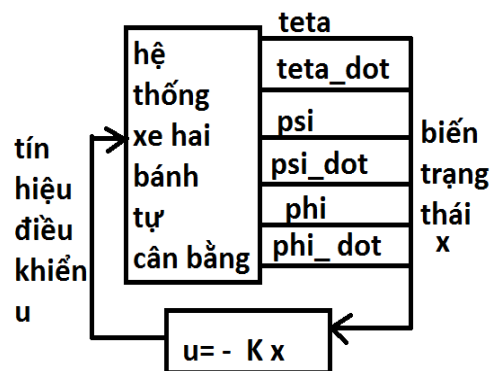
$$Q = \begin{bmatrix} Q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_6 \end{bmatrix}; \text{ ma trận } R = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

Đối với bộ điều khiển LQR, với các thông số $Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, R_1$ và R_2 để tinh chỉnh cho bộ điều khiển LQR. Trong đó, thông số $Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, R_1$ và R_2 lần lượt là các trọng số tốt tương ứng cho các giá trị $teta, teta_dot, psi, psi_dot, phi, phi_dot$, tín hiệu điều khiển cho góc tiến tới và tín hiệu điều khiển cho góc xoay. Nếu muốn biến trạng thái nào được ưu tiên trong việc điều khiển thì ta cho thông số trọng số tương ứng đó lớn hơn tương đối so với các thông số trọng số khác.

Sau khi chọn được thông số điều khiển tương ứng, ta tiến hành xây dựng thông số

$$\text{hồi tiếp K với tín hiệu điều khiển } u = -K \cdot x \text{ với } x = \begin{bmatrix} \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ \phi \\ \dot{\phi} \end{bmatrix}$$

Thông số K được tính dựa vào phương trình Riccati ở (2.71). Tuy nhiên, Matlab đã hỗ trợ ta hàm `lqr()` để tính ra thông số K khi đã biết ma trận A, B, Q, R.



Hình 2.7: Sơ đồ chi tiết bộ điều khiển LQR cho hệ xe hai bánh tự cân bằng

2.3.3 Các thành phần chính của mô hình

2.3.3.1 Nguồn

Nguồn cung cấp cho robot sử dụng nguồn pin lypo. Pin lypo có ưu điểm nhỏ gọn và hoạt động lâu. Hiện thời, pin này đang dần thay thế các loại acquy trong các ứng dụng.

2.3.3.2Vi xử lý

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



Hình 2.8: DSP TMS320F28335

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

2.3.3.3 Cảm biến IMU

Robot sử dụng cảm biến IMU (Inertial Measurement Unit-cảm biến đo lường quán tính) để tính toán, ước lượng giá trị góc nghiêng và vận tốc góc nghiêng thânrobot, tên đầy đủ là Razor IMU 9 Degrees Of Freedom, là sự tích hợp của cảm biến gia tốc ADXL 345 (triple-axis accelerometer), cảm biến con quay hồi chuyển ITG3200 (MEMS triple-axis gyro) và cảm biến từ trường HMC5883L (triple-axis magnetometer) trên cùng 1 bo mạch.

Việc sử dụng board Arduino để đọc tín hiệu cảm biến về rồi mới truyền dữ liệu lên board DSP TMS320F28335 là vì Arduino sử dụng một bộ thư viện mở. Thư

viện trên đã hỗ trợ các chương trình đọc tín hiệu cảm biến của IMU MPU6050, bao gồm cả chương trình lọc Kalman phức tạp trong đó. Việc lập trình DSP để đọc trực tiếp tín hiệu cảm biến độ nghiêng là rất khó khăn.

Tốc độ đọc về của cảm biến tương ứng tốc độ xử lý của chương trình là theo chu kỳ 0.01s tức 100Hz



Hình 2.9: IMU MPU6050

Cảm biến MPU6050 (GY-521): MPU6050 là cảm biến gia tốc 3 trục công suất thấp của hãng InvenSense thích hợp cho các ứng dụng di động, độ phân giải cao (13bit), với tầm đo $\pm 16g$ và có ngõ ra theo chuẩn truyền dữ liệu I2C . Ở đây, tác giả sử dụng một vi điều khiển Arduino có chương trình đọc và phân tích sẵn giá trị cảm biến độ nghiêng (chương trình được hỗ trợ trên mạng dưới dạng code). Sau khi nạp chương trình, tín hiệu được truyền về DSP thông qua cổng SCI.

2.3.3.4 Bo mạch Arduino Pro Mini:

Bo mạch Arduino Pro Mini là một board mạch dựa trên vi điều khiển ATmega328. Nó có 14 chân I/O số (trong đó có 6 chân có thể điều xung PWM), 8 kênh đọc giá trị Analog.

Đặc tính kỹ thuật:

- Vi điều khiển: ATmega328
- Điện áp làm việc: 3.3V ~ 5V
- Điện áp đầu vào: 3.35V ~ 12V
- Ngõ I/O số: 14 chân (6 chân điều xung PWM)

- Ngõ vào Analog: 8
- Dòng điện qua mỗi chân: 40mA
- Bộ nhớ FLASH: 16KB
- SRAM: 1KB
- EEPROM: 512 bytes
- Thạch Anh: 16 MHz

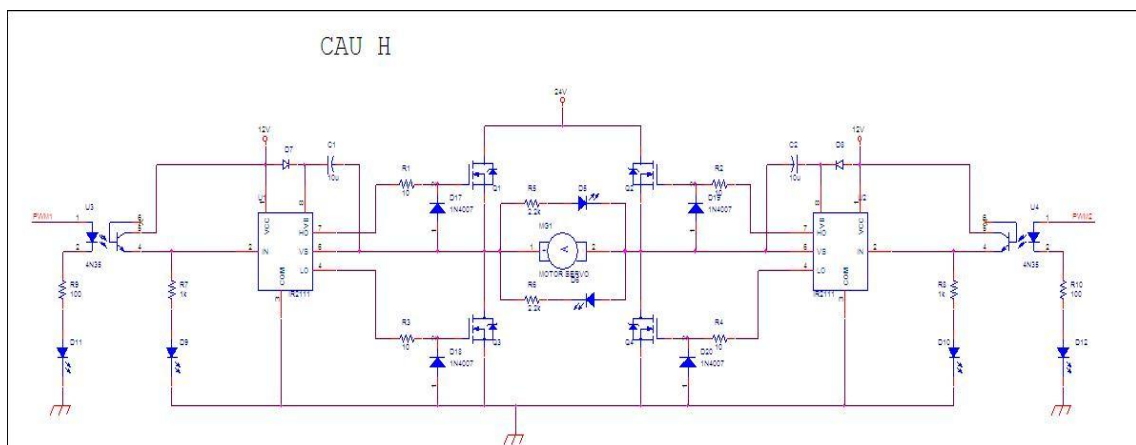
Chức năng một số chân:

- UART: RX (chân 0), TX (chân 1)
- Ngắt ngoài: gồm 2 ngắt ngoài là chân 2, chân 3.
- Điều xung PWM: Gồm 6 chân 3, 5, 6, 9, 10 và 11. Hỗ trợ điều xung độ phân giải 8 bit.
- Chuẩn giao tiếp: SPI gồm chân 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).
- Chuẩn giao tiếp I²C gồm chân A4 (SDA) và A5 (SCL).

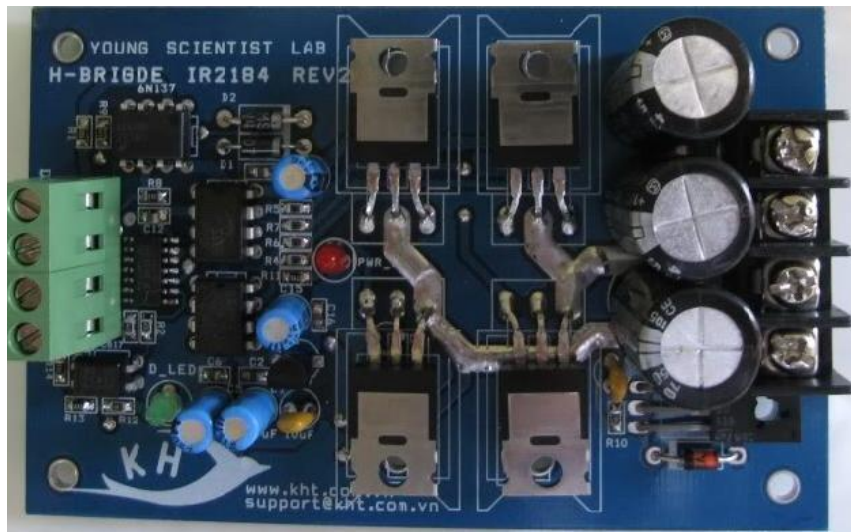
Ở đây, tác giả đã giao tiếp Arduino với MPU6050 qua chuẩn I2C để tính toán ra góc nghiêng cụ thể. Sau khi tính toán được góc nghiêng, Arduino truyền dữ liệu với DSP thông qua chuẩn UART, cụ thể là kênh SCI B của DSP.

2.3.3.5 Mạch cầu H

Gồm 4 Mosfet có tác dụng đảo chiều động cơ



Hình 2.10: Sơ đồ nguyên lý mạch cầu H



Hình 2.81: Mạch cầu H thực tế

Mạch cầu H chịu được dòng 10 A và điều xung được 95%. Có ngõ vào điều xung PWM và ngõ vào chọn chiều quay của động cơ. Ngõ điều xung được lấy từ kênh điều xung của DSP, ngõ chọn chiều quay được kích từ chân GPIO của DSP.

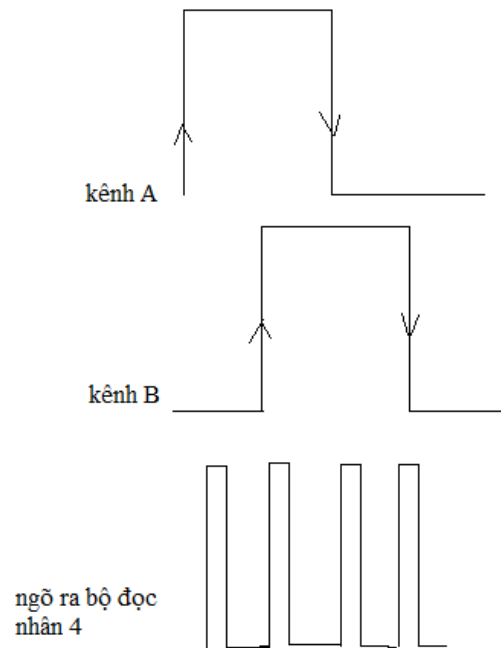
2.3.3.6 Động cơ

Mô hình sử dụng 2 động cơ không hộp số, 12VDC, tỉ số truyền 1, moment xoắn 2Nm

Encoder 250 xung/vòng gắn trên thân robot, tiếp xúc không trượt với bánh xe robot, tỉ lệ số vòng bánh xe encoder và số vòng bánh xe robot là 1:1 để đo vị trí và vận tốc di chuyển của robot.

Theo thử nghiệm, encoder này được nuôi nguồn 3v3 vẫn cho ngõ ra số rất ổn định, có hai ngõ ra A và B báo hiệu chiều quay cũng như số vòng quay theo chiều tương ứng.

Để tăng độ chính xác của encoder, thay vì một trạng thái tương ứng 1 xung cạnh lên và cạnh xuống của kênh A và một xung cạnh lên và một xung cạnh xuống của kênh B thì trong bộ TMS320F28335 có một bộ đọc cả xung cạnh lên và cạnh xuống của cả kênh A và kênh B. Như vậy, cứ mỗi trạng thái sẽ được bộ đọc này xuất ra 4 xung. Như vậy, độ chính xác đã được tăng lên bốn lần.



Hình 2.92: Nguyên lý hoạt động bộ đọc nhân 4

Do DSP có bộ nhân 4 nên số xung báo về cho một vòng quay bánh xe là 1000 xung/ vòng. Độ phân giải bánh xe quay đạt được là $360^\circ / 1000 = 0.36^\circ$.



Hình 2.103: Động cơ

CHƯƠNG 3

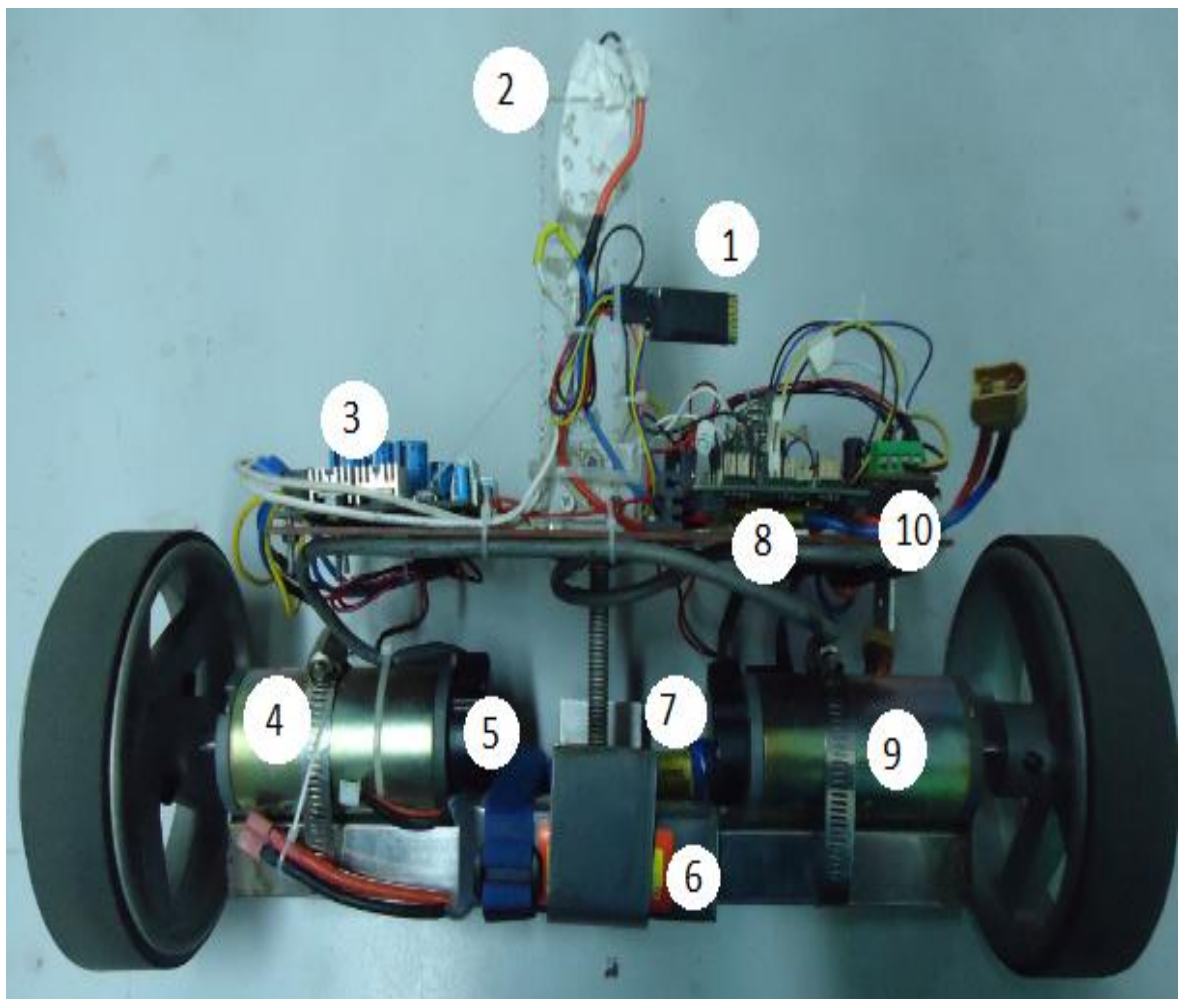
THIẾT KẾ HỆ THỐNG

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

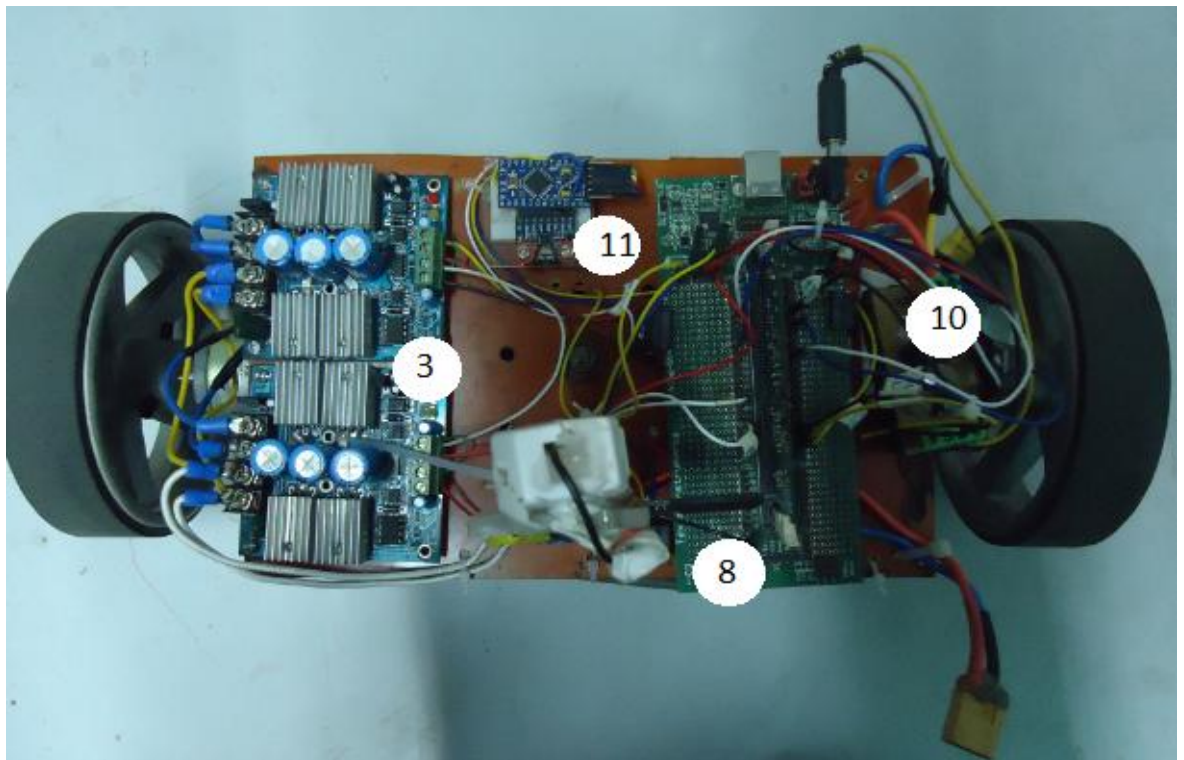
3.1.1 Thiết kế cơ khí

Khung sườn sử dụng thanh sắt hộp, có 1 ren sắt khoan đồng trục. Cơ cấu chuyển động encoder áp bánh và bố trí mạch theo chiều dựng đứng.

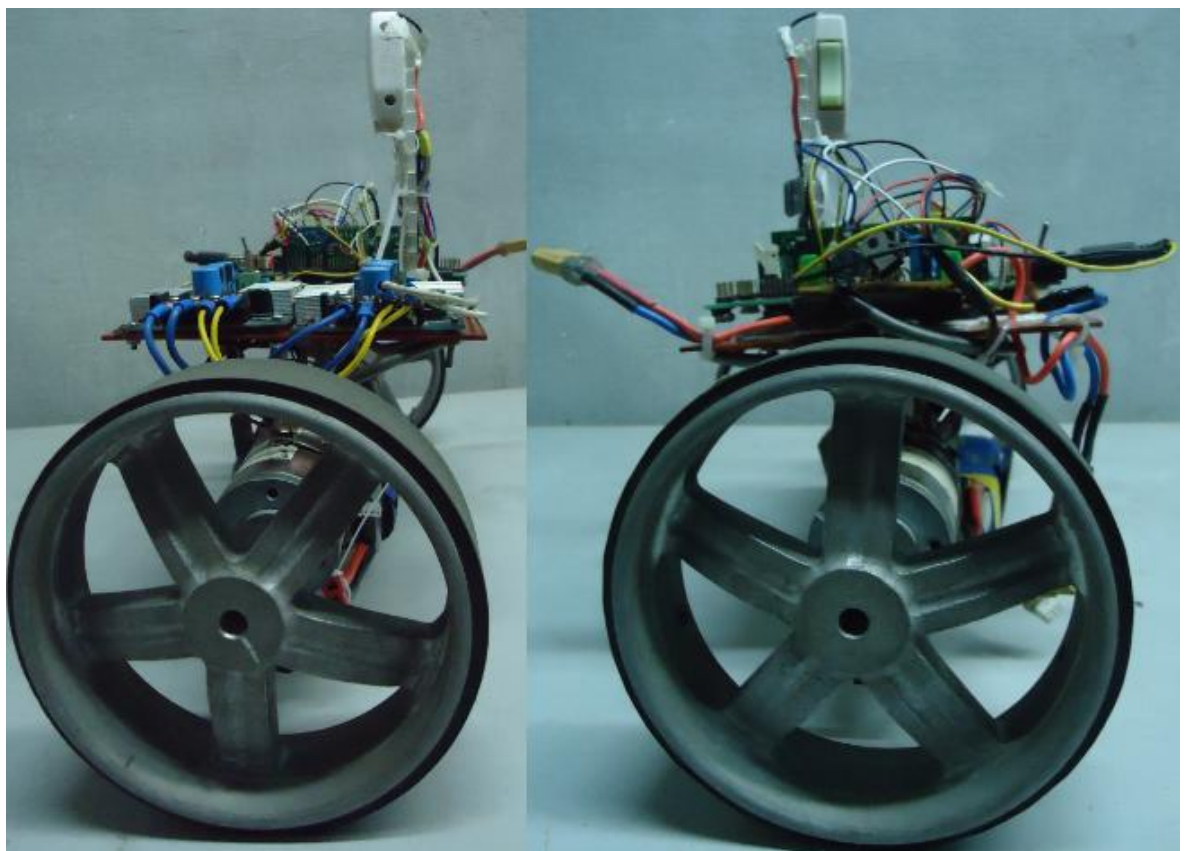
Hình ảnh thực tế của mô hình robot 2 bánh tự cân bằng



Hình 3.1: Mô hình robot nhìn từ trước



Hình 3.2: Hình robot chụp nhìn từ trên xuống



Hình 3.3: Hình robot nhìn ở hai bên

Chú thích hình vẽ:

- 1: module thu phát Blue Tooth
- 2: công tắc cấp nguồn pin cho động cơ
- 3: cầu H
- 4 và 9: động cơ bánh trái, bánh phải
- 5 và 7: encoder bánh trái và phải
- 6: pin cấp nuôi động cơ và DSP
- 8: board điều khiển DSP
- 10: board chuyển đổi nguồn 12 volt sang 5 volt để nuôi board DSP
- 11: cảm biến độ nghiêng kết nối với Arduino để truyền tín hiệu độ nghiêng về cho DSP

3.1.2 Cấu trúc điều khiển phản cứng

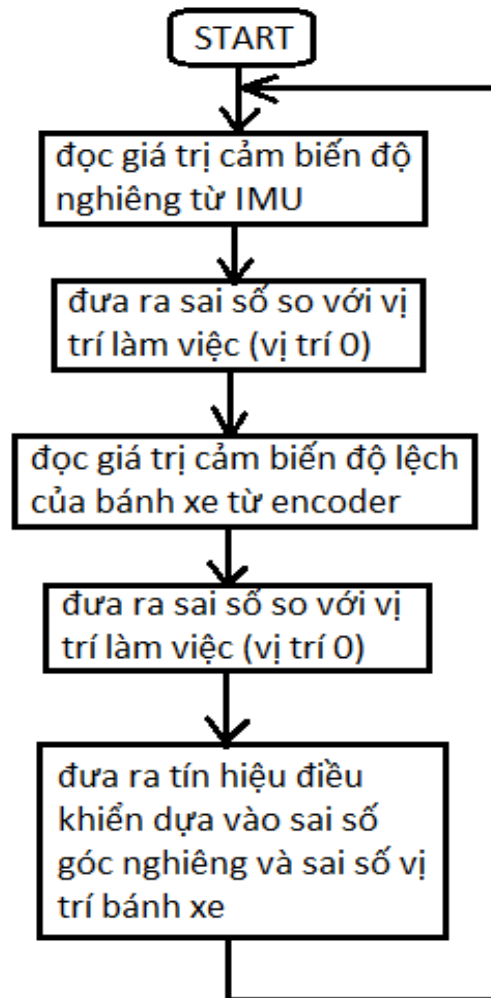
Encoder bánh phải, encoder bánh trái báo về góc tiến tới của bánh trái và góc tiến tới của bánh phải. Từ các tín hiệu trên, ta suy ra góc θ và ϕ của xe hái bánh cân bằng (theo công thức 2.1). Mặt khác, cảm biến MPU6050 báo về giá trị độ nghiêng ψ của xe. Chương trình điều khiển được lập trình nhúng vào chip DSP. Dựa vào tín hiệu góc nghiêng và tín hiệu góc xoay mà tín hiệu điều khiển được xuất ra tương ứng với từng cầu H, điều khiển các động cơ trái và phải. Nếu bánh trái chạy nhanh hơn bánh phải, ta có xe đang ở tình trạng quẹo phải. Nếu bánh trái chạy chậm hơn bánh phải ta có tình trạng quẹo phải. Nếu bánh trái và bánh phải cùng tiến tới với một tốc độ xác định ta có tình trạng đi tới. Ngược lại, ta có tình trạng đi lùi.

Tín hiệu xử lý của cảm biến có chu kì là 0.01s, tức 100Hz. Đáp ứng của động cơ là 1MHz (tùy thuộc vào đáp ứng của FET trên cầu H) nên động cơ hoàn toàn đáp ứng được tốc độ xử lý của chương trình (chu kì lấy mẫu 0.01s)

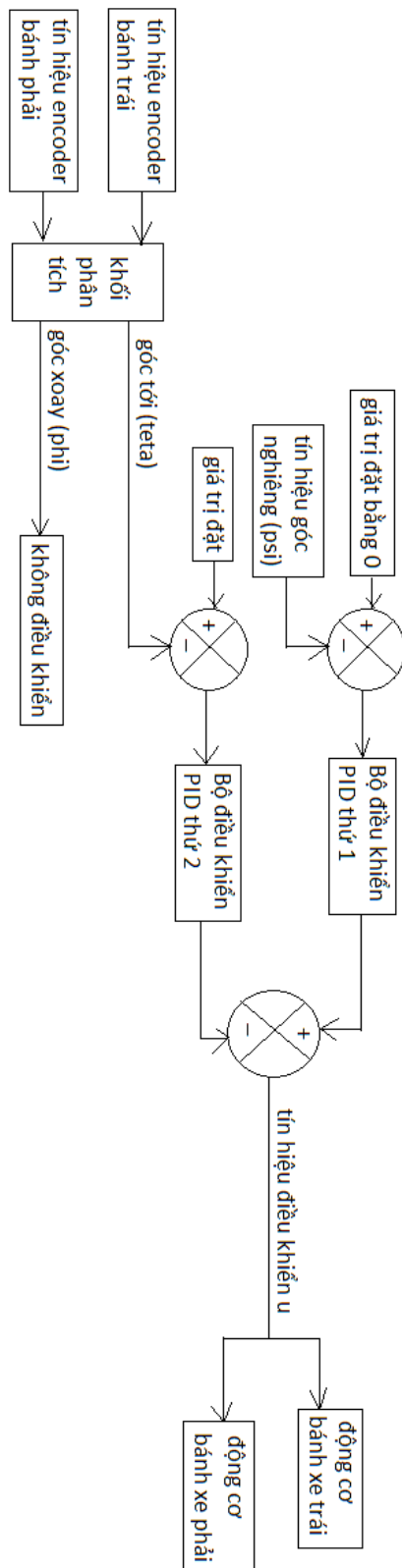
3.2 Thiết kế phần mềm

3.2.1 Lưu đồ giải thuật

- Điều khiển PID



Hình 3.4: Lưu đồ giải thuật điều khiển PID



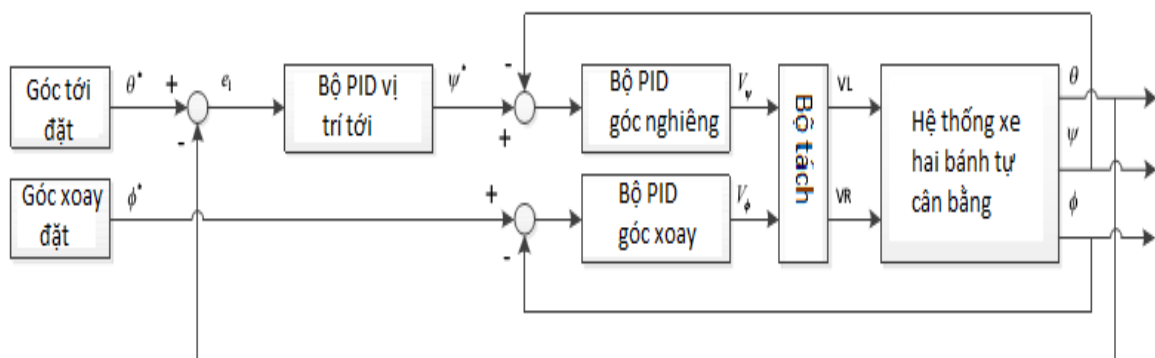
Hình 3.5: Sơ đồ khối bộ điều khiển PID

Bộ điều khiển PID là bộ điều khiển cho hệ một vào một ra. Do đó, ở đây, tác giả chỉ điều khiển thỏa hiệp giữa góc nghiêng (ψ) và góc tới ($teta$). Tức một bộ điều khiển PID để điều khiển góc $teta$ và một bộ PID điều khiển góc ψ . Tín hiệu điều khiển PID cuối cùng có được là tổng tín hiệu điều khiển PID từ hai bộ PID trên. Do hai động cơ là hoàn toàn giống nhau và cùng được cấp tín hiệu điện áp giống nhau trong cùng một thời điểm nên góc xoay ψ là không đổi trong quá trình điều khiển PID.

Nếu ta muốn động cơ chạy tới 1 khoảng nhất định thì ta đặt một giá trị đặt tương ứng với góc tới ($teta$). Giá trị đặt sẽ dương trong trường hợp ta muốn xe chạy tới và giá trị đặt sẽ âm trong trường hợp ta muốn xe chạy lùi. Còn đối với góc nghiêng (ψ), ta luôn muốn xe đứng thẳng hướn lên nên giá trị đặt đối với góc ψ luôn bằng 0.

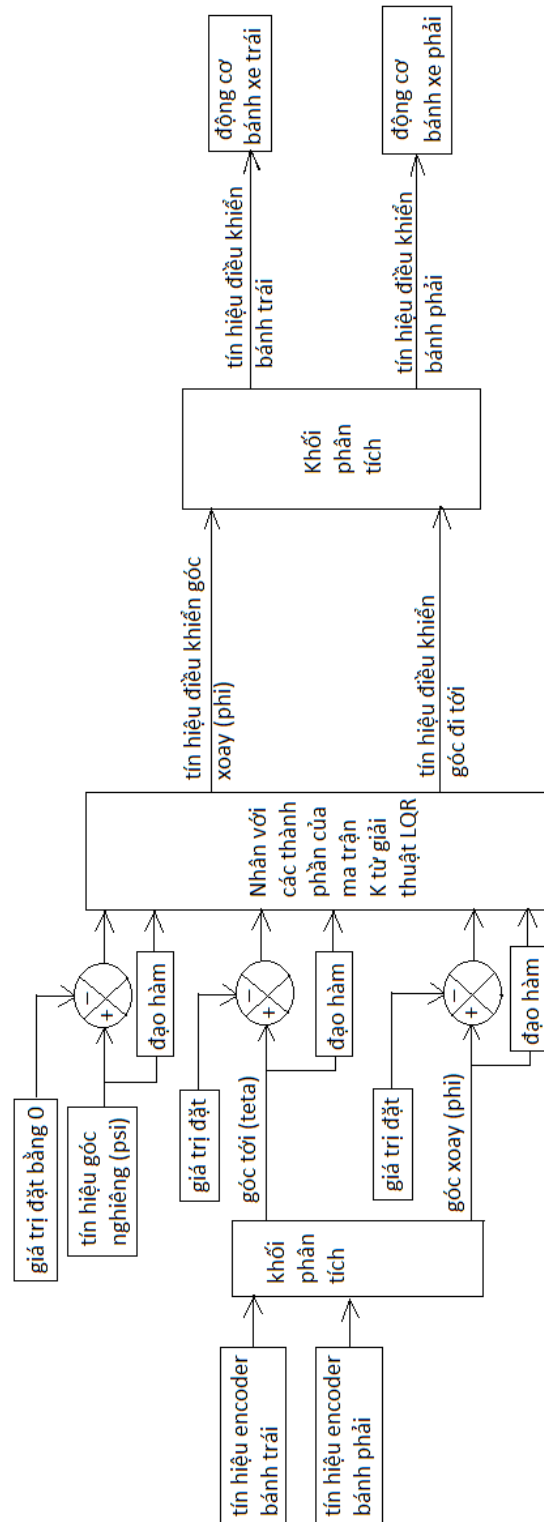
Ngoài ra, có một phương pháp điều khiển PID được đề cập ở [8]. Tuy nhiên, các tác giả trên cũng chưa áp dụng thành công bộ điều khiển trên để chạy thực tế được. Mô tả điều khiển PID theo dạng này như là ba bộ PID được sử dụng để điều khiển robot hai bánh tự cân bằng, bao gồm:

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



Hình 3.6: Cấu trúc bộ điều khiển PID cho hệ robot hai bánh tự cân bằng

- Điều khiển LQR



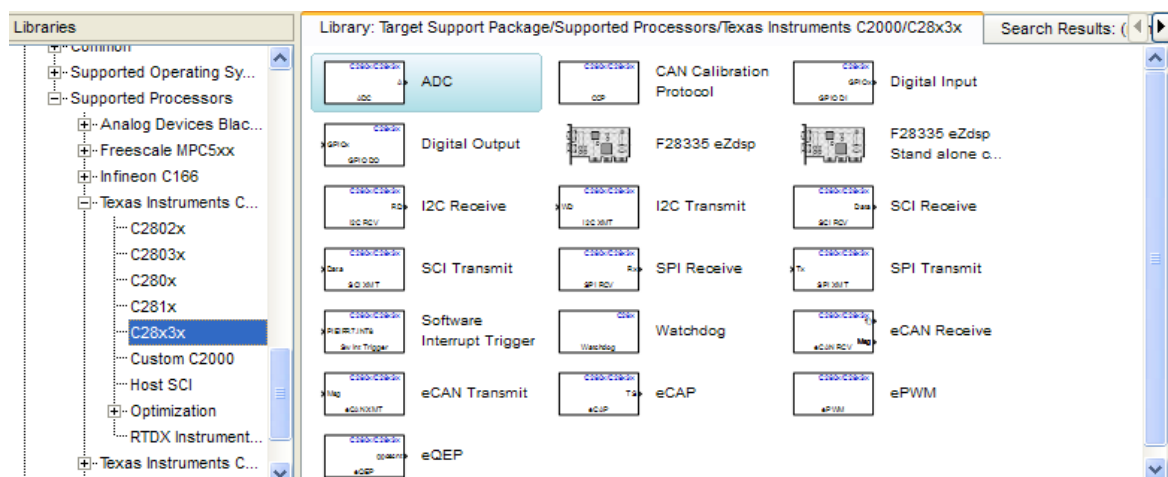
Hình 3.7: Sơ đồ khối bộ điều khiển LQR

3.2.2 Bộ điều khiển nhúng cho robot 2 bánh tự cân bằng

3.2.2.1 Giới thiệu

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

- MATLAB



Hình 3.8: Thư viện Target Support Package cho DSP F28335

Matlab được sử dụng là Version 2012b. Chương trình được viết hoàn toàn trên Simulink/ Matlab. Sau đó, ta tiến hành tạo Project tương ứng trong CCS thông qua thao tác Build all chương trình trong Simulink. Khi Project được tạo thành công trong CCS thì ta tiếp tục Build project trên trong CCS để tạo ra file .out tương ứng nạp vào chip TMS320F28335 (tương tự như file .hex nạp cho chip 89c51).

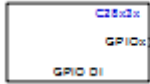
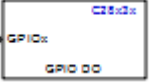
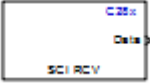


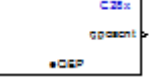
- Chương trình CCS

Chương trình CCS được sử dụng là phiên bản 3.3 do hãng Texas Instrument xây dựng, được kèm theo khi mua board DSP 28335 gồm hai phần :

- Chương trình cài đặt cấu hình cho CCS
- Chương trình nạp chip cho DSP

3.2.2.2 Các khối chức năng sử dụng trong chương trình

Bảng 3.1: Các khối chức năng sử dụng trong chương trình

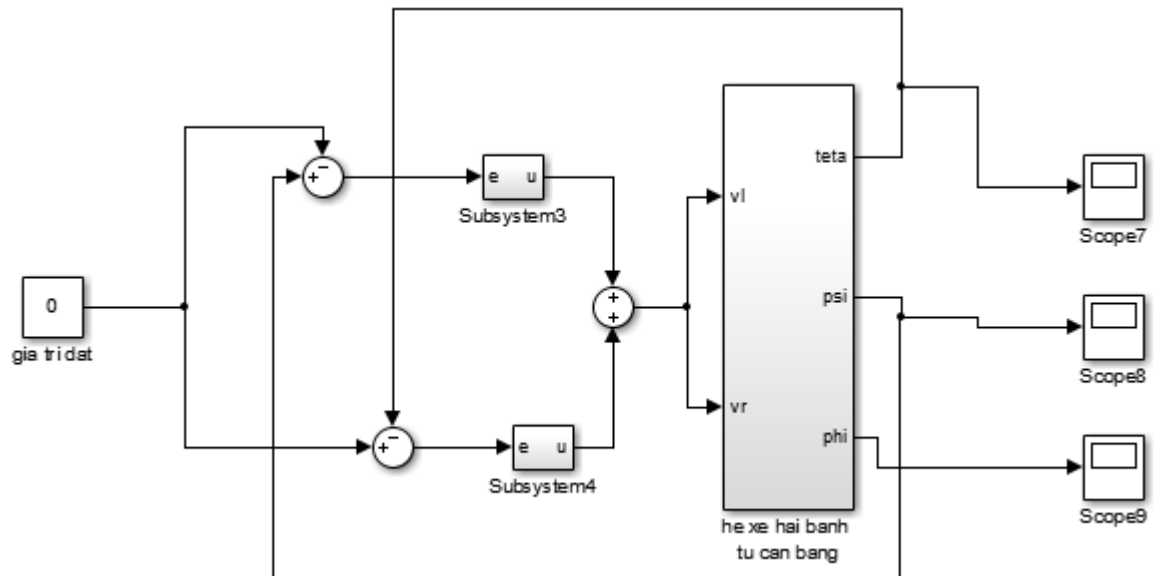
Biểu tượng	Chức năng
	Nhận tín hiệu digital vào: nút start, stop, calibrate
	Xuất tín hiệu digital: các đèn hiển thị
	Nhận tín hiệu truyền thông UART: đọc cảm biến IMU
	Xuất tín hiệu truyền thông UART: gửi các giá trị về máy tính để theo dõi, giám sát
	Điều chế độ rộng xung, đảo chiều động cơ
	Đọc encoder

CHƯƠNG 4

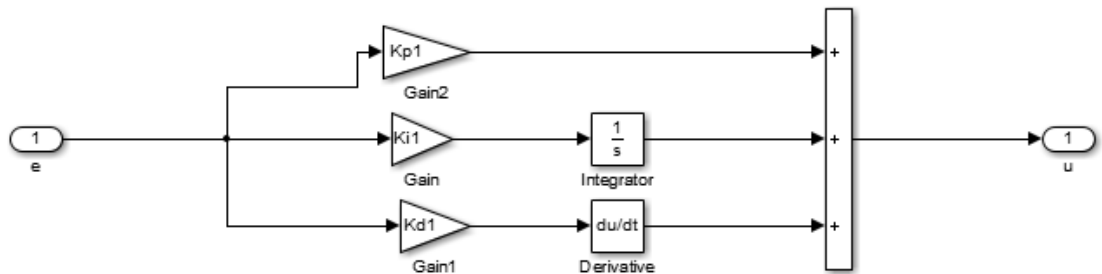
KẾT QUẢ MÔ PHỎNG

4.1 Giải thuật PID

4.1.1 Chương trình mô phỏng



Hình 4.1: Sơ đồ lập trình thực cho bộ điều khiển PID



Hình 4.2: Sơ đồ khối điều khiển PID

4.1.2 Kết quả mô phỏng

Giá trị khởi tạo:

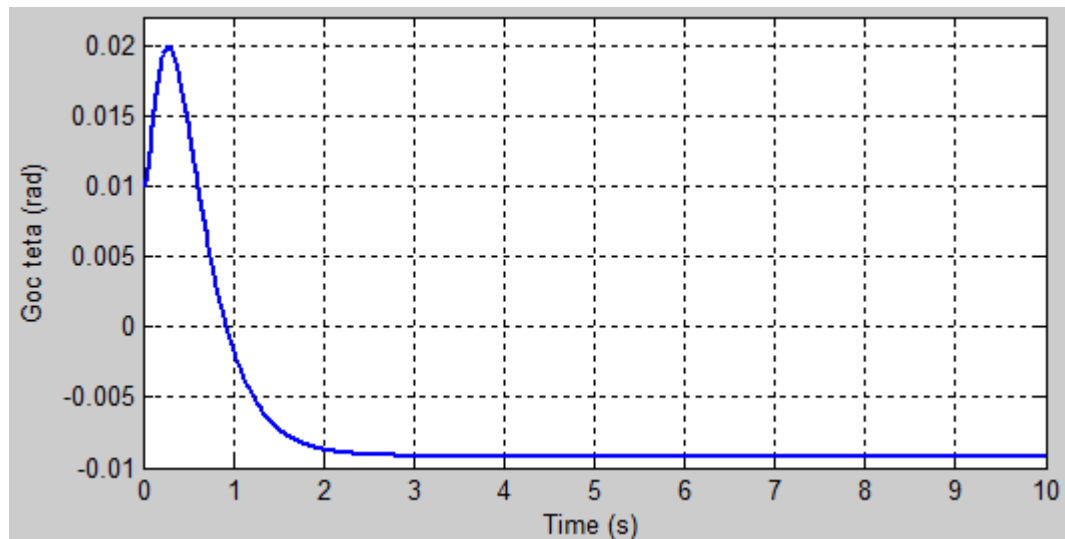
$$\theta_{init} = 0.001(rad) ; \dot{\theta}_{init} = -0.0012(rad / s) ; \psi_{init} = 0.002(rad) ; \dot{\psi}_{init} = -0.002(rad / s) ;$$

$$\phi_{init} = 0.002(rad) ; \dot{\phi}_{init} = 0.002(rad) ;$$

Việc thử sai các giá trị thông số điều khiển PID theo nguyên tắc. Nếu chọn K_{p1} , K_{i1} , K_{d1} lớn hơn tương đối so với K_{p2} , K_{i2} , K_{d2} thì độ ổn định và chính xác của góc nghiêng ψ được ưu tiên và ngược lại. Nếu xe dao động quá mạnh tương

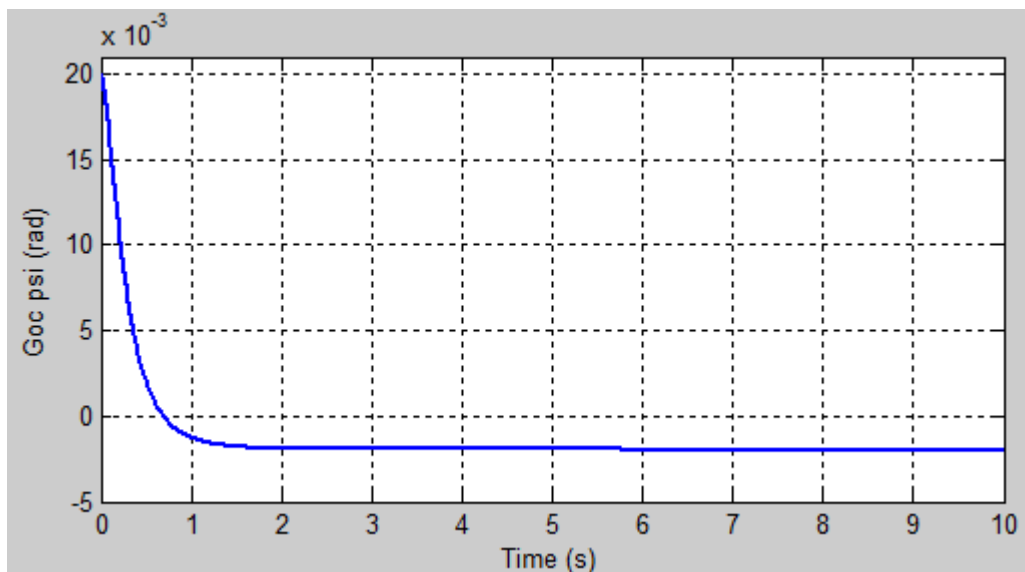
ứng giảm K_p , tăng K_d . Nếu muốn giảm sai số xác lập thì tăng K_i . Nếu muốn tăng thời gian đáp ứng, giảm sai số xác lập thì tăng K_p . Qua quá trình thực nghiệm thử sai theo qui tắc trên, các thông số điều khiển được chọn là:

$$\begin{aligned} K_{p1} &= 447.4000; K_{i1} = 9.7000; K_{d1} = 72.4000; \\ K_{p2} &= 92.5000; K_{i2} = 2.2000; K_{d2} = 11.6000; \end{aligned}$$



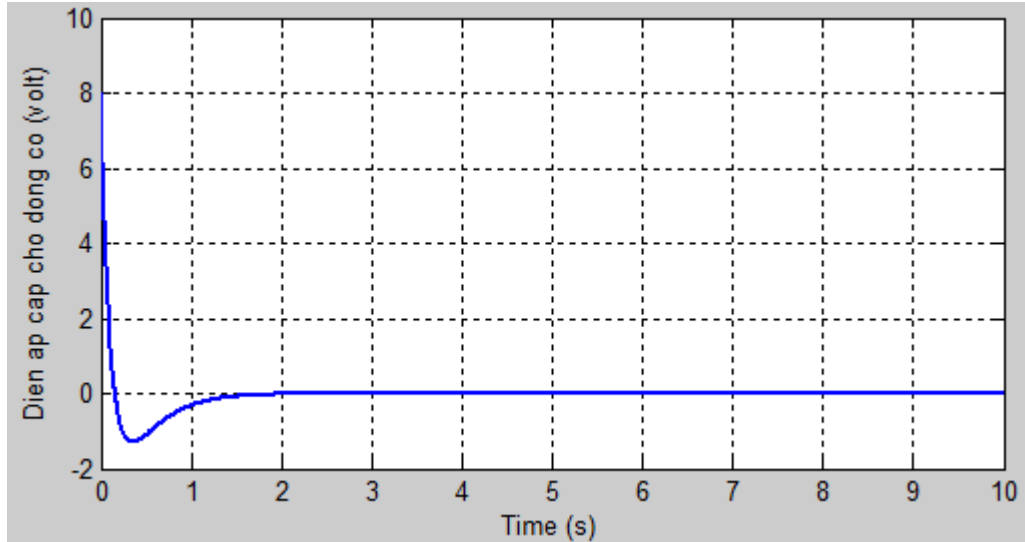
Hình 4.3: Tín hiệu vị trí góc quay của bánh xe θ (rad)

Hình 4.3 cho thấy sự thay đổi góc đi tới của xe: sau một khoảng quá độ, góc tới xe đã về vị trí xác lập là -9mm (gần bằng -0.01m). Kết quả này có sai lệch so với góc tới mong muốn là 0m



Hình 4.4: Tín hiệu góc nghiêng của bánh xe ψ (rad)

Hình 4.4 cho thấy sau khoảng 1.5 s dao động thì góc nghiêng của xe ổn định ở -2×10^{-3} rad. Giá trị trên vẫn còn sai lệch so với góc nghiêng lý tưởng là 0 rad. Tuy nhiên, sai lệch trên là không đáng kể



Hình 4.5: Tín hiệu điện áp điều khiển (Volt)

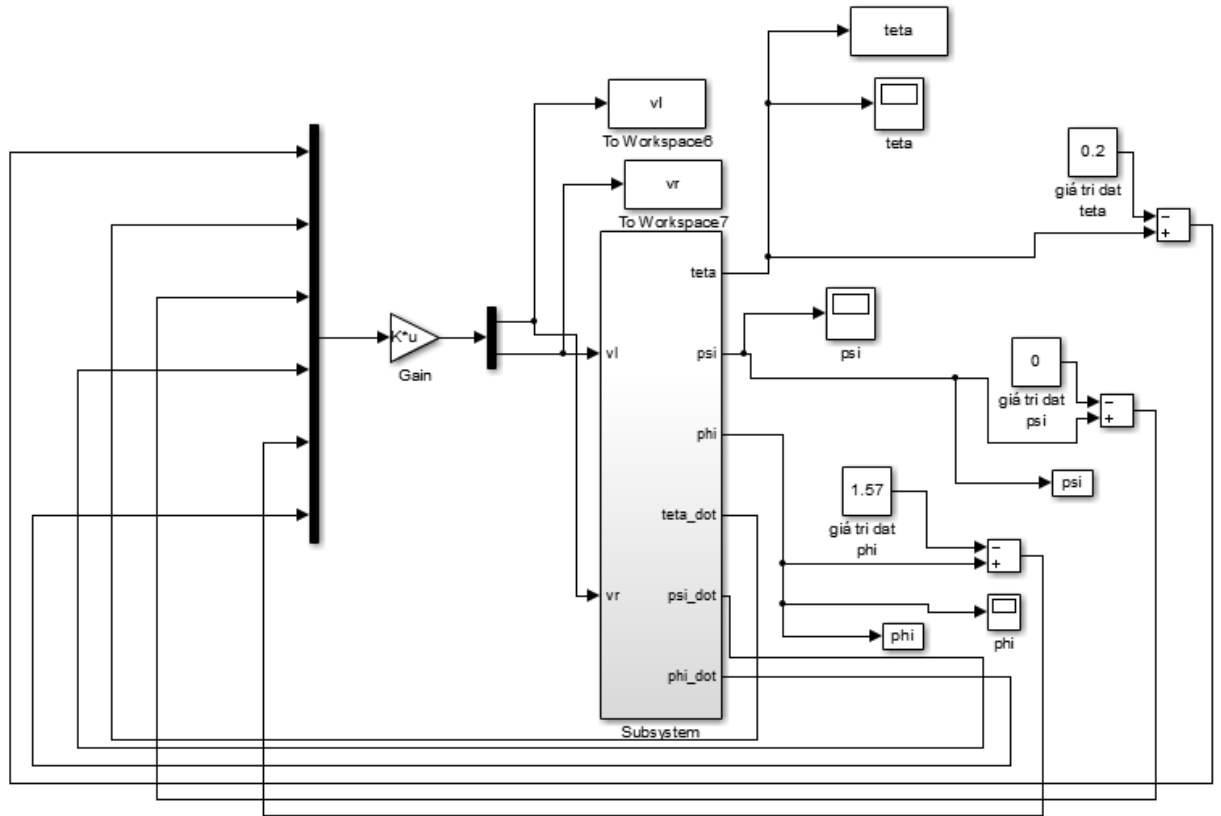
Hình 4.5 cho thấy sau khoảng 1.5s thì tín hiệu điều khiển về 0 vì tất cả các biến trạng thái đều đã xác lập nên không cần xuất tín hiệu điều khiển nữa.

4.1.3 Nhận xét

Ở đây, ta đặt mong muốn là xe ở đúng vị trí mong muốn (vị trí 0) và xe thẳng đứng hướng lên (góc $\psi=0$). Do đó, tác giả sử dụng bộ PID kết hợp (ngõ vào là θ, ψ) và ngõ ra là tín hiệu điện áp điều khiển. Ở đây, thông số PID được điều chỉnh nhưng chưa tối ưu nên hệ thống ổn định nhưng vẫn còn tồn tại sai số xác lập và thời gian để xác lập còn quá lớn. Mặt khác, ta không điều khiển được góc xoay ϕ .

4.2 Giải thuật LQR

4.2.1 Chương trình mô phỏng



Hình 4.6: Chương trình lập trình LQR

4.2.2 Kết quả mô phỏng

Giá trị đầu được chọn gần với điểm làm việc. Điểm làm việc là vị trí xe đứng yên tại vị trí 0 ($teta=0$), không nghiêng ($psi=0$), không xoay ($phi=0$)

Từ đó, ta chọn giá trị đầu:

$$\theta_{init} = 0.0001(rad) \quad ; \quad \dot{\theta}_{init} = -0.00012(rad/s) \quad ; \quad \psi_{init} = 0.0002(rad) \quad ;$$

$$\dot{\psi}_{init} = -0.0002(rad/s) \quad ; \quad \phi_{init} = 0.0002(rad) \quad ; \quad \dot{\phi}_{init} = 0.0002(rad) \quad ;$$

Mã trận trọng số LQR được chọn như sau:

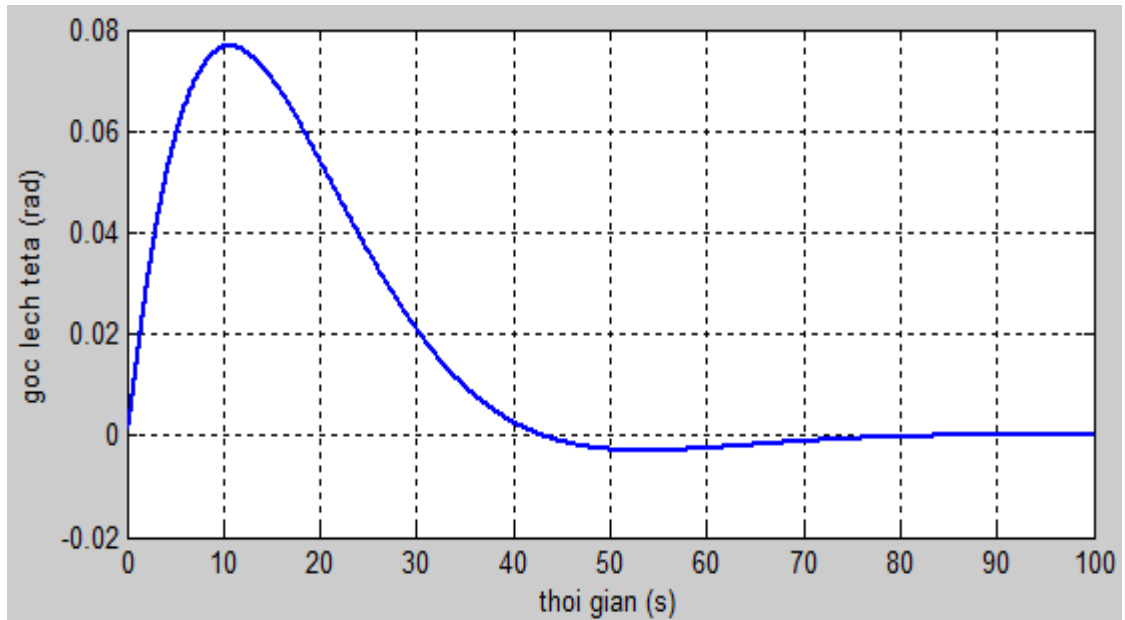
$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Lúc này, ta được ma trận K theo lệnh

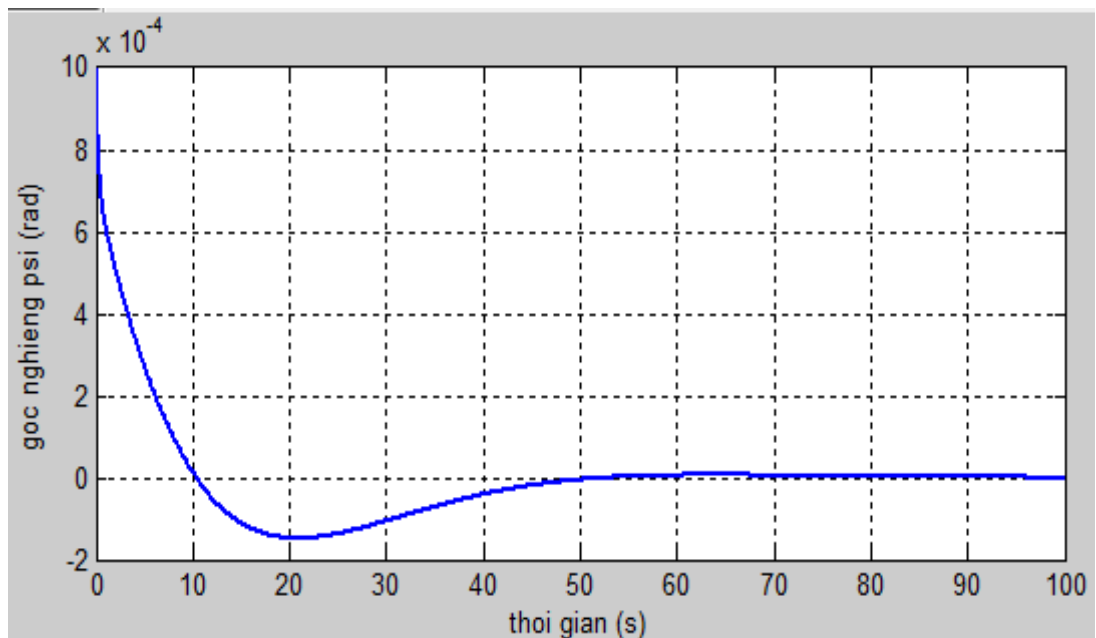
$$K = \text{lqr}(A, B, Q, R)$$

$$K = \begin{bmatrix} -0.7 & 236.3 & -6066 & -1390.9 & -0.7 & -10.4 \\ -0.7 & 236.3 & -6066 & -1390.9 & 0.7 & 10.4 \end{bmatrix}$$

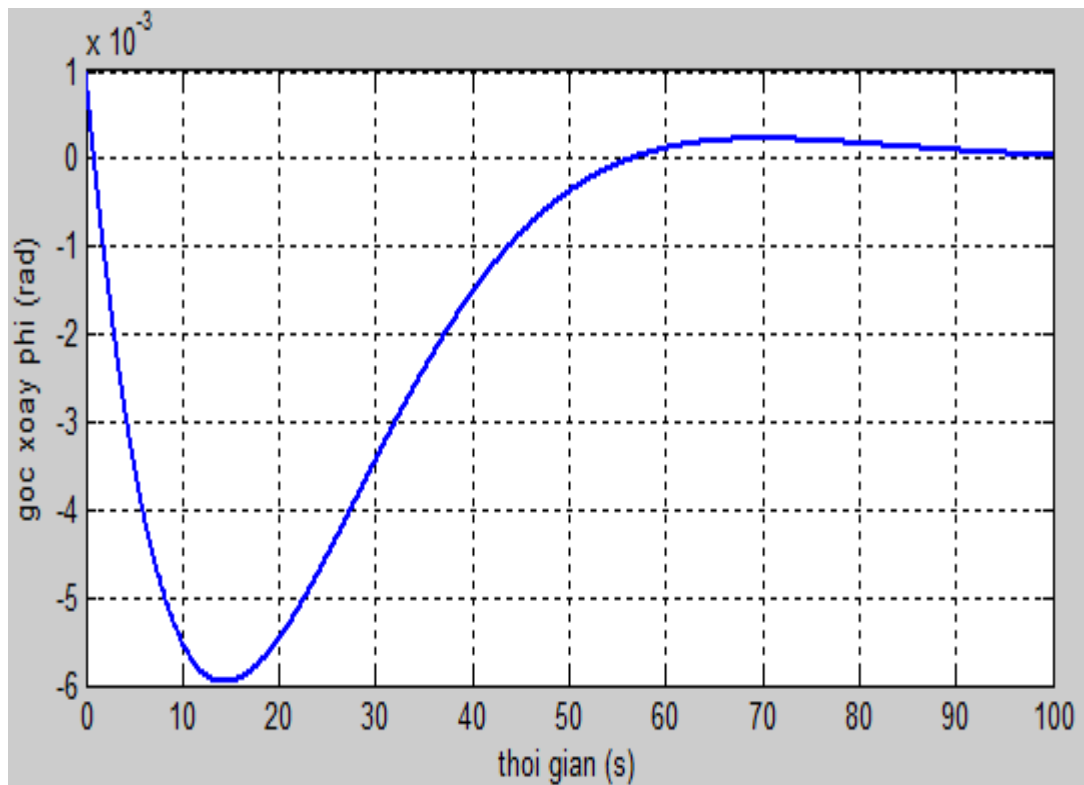
4.2.2.1 Giá trị đặt bằng 0



Hình 4.7: Góc bánh xe θ (rad)

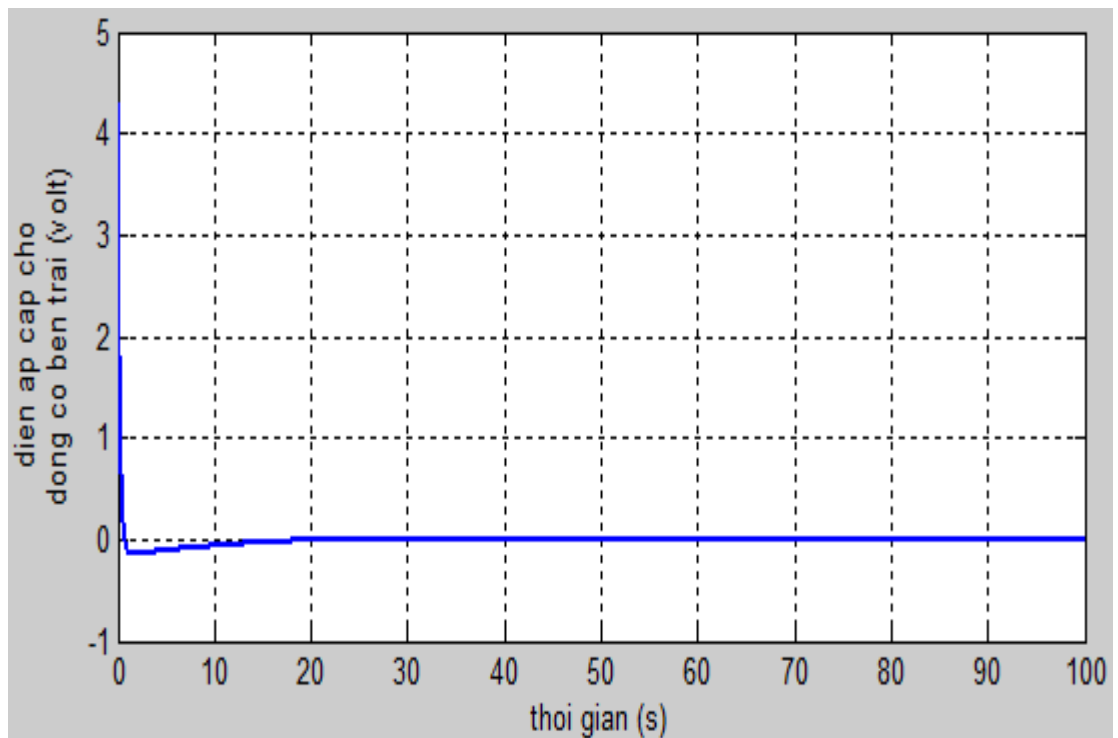


Hình 4.8: Góc nghiêng ψ (rad)



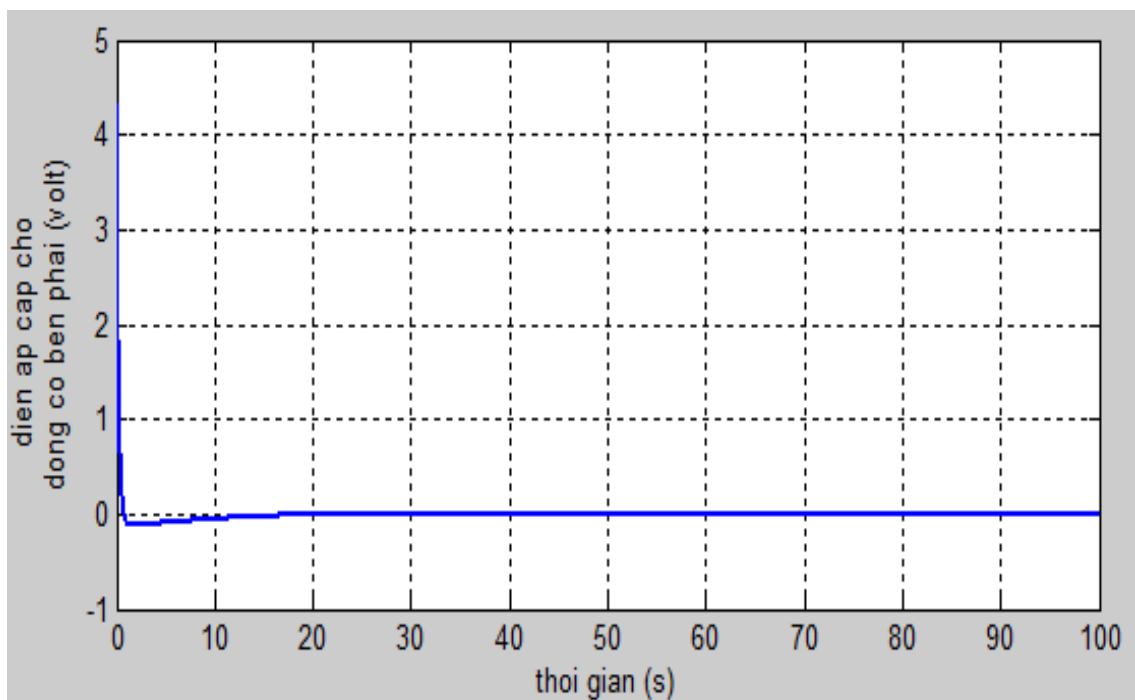
Hình 4.9: Góc xoay ϕ (rad)

Sau khoảng 90s thì góc xoay phi đã về vị trí xác lập là 0. Trong trường hợp này, góc xoay đã quay về đúng vị trí lý tưởng và không bị sai số xác lập. Tuy nhiên, do thông số điều khiển chọn được chưa phải tối ưu nhất nên thời gian xác lập còn quá lâu (hơn 1 phút)



Hình 4.10: Điện áp cấp cho động cơ trái (Volt)

Hình 4.10 cho thấy điện áp cấp cho động cơ về giá trị 0 khi hệ thống đã về được trạng thái ổn định xác lập.

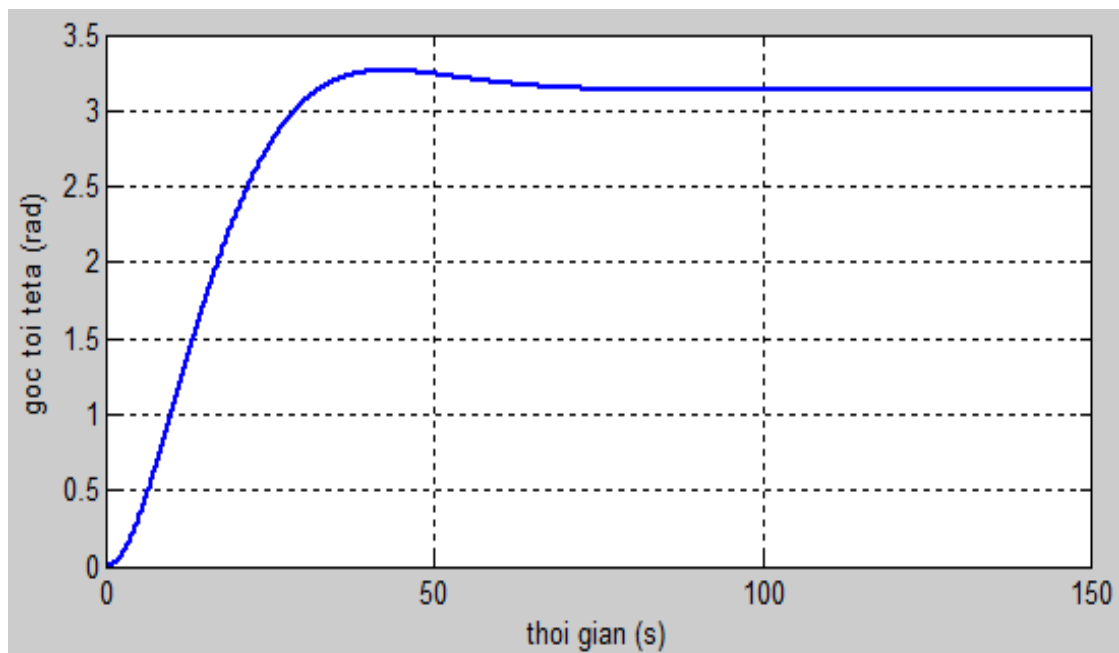


Hình 4.11: Điện áp cấp cho động cơ phải (Volt)

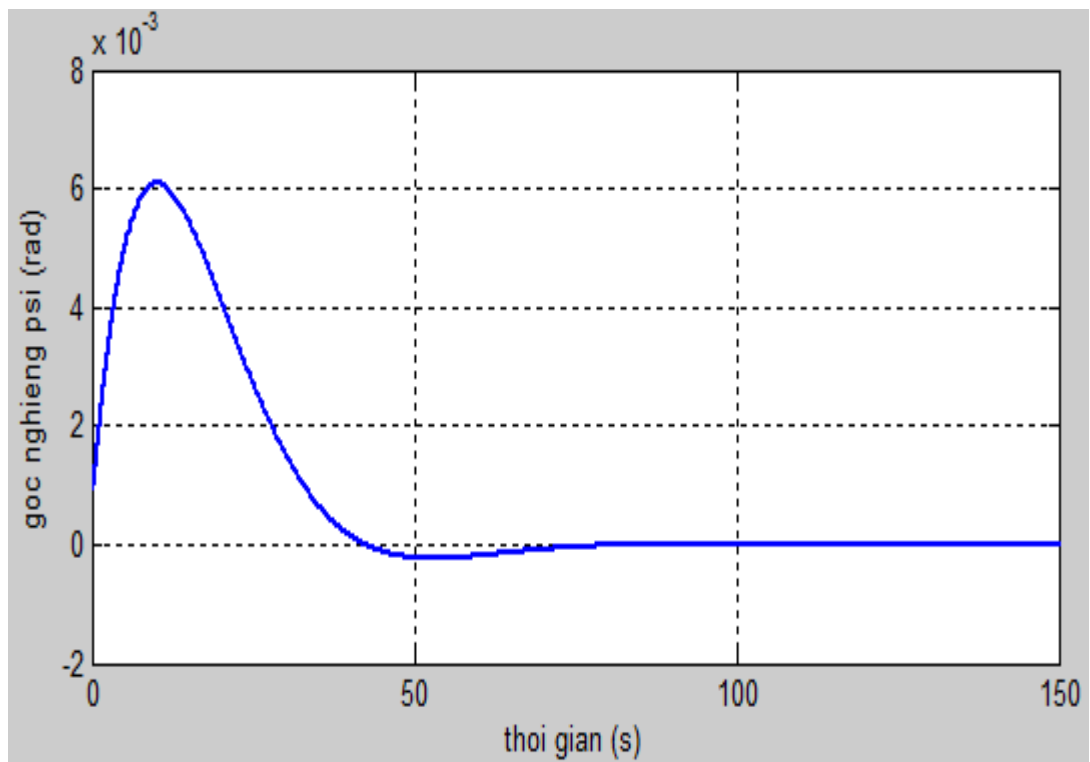
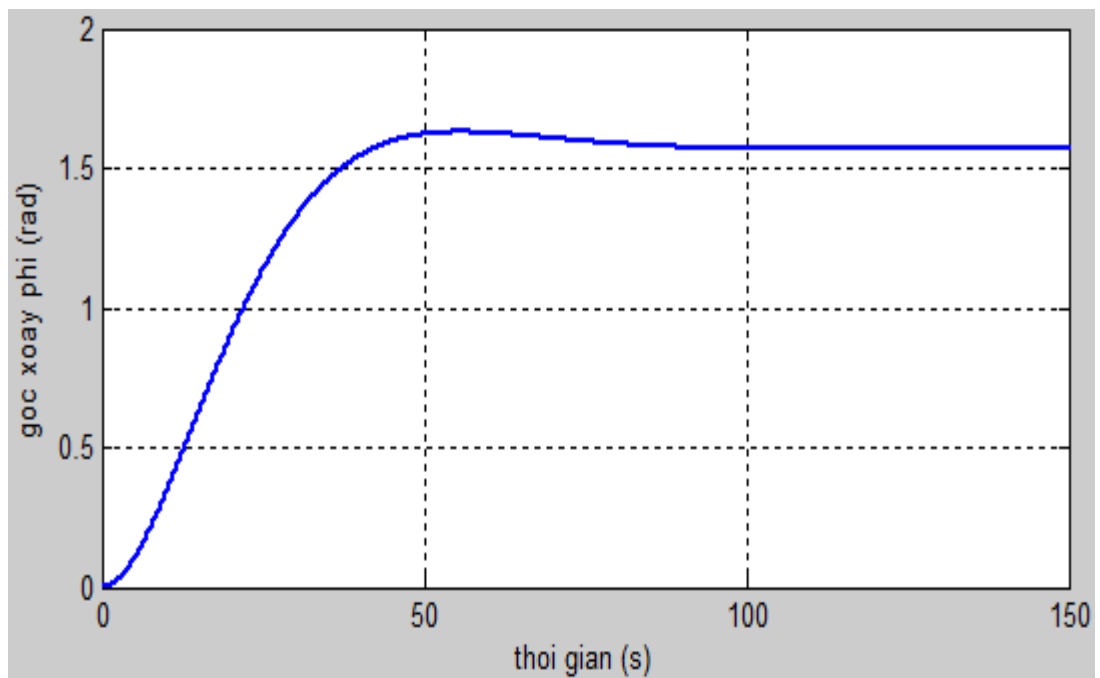
Hình 4.11 cho thấy điện áp cấp cho động cơ về giá trị 0 khi hệ thống đã về được trạng thái ổn định xác lập.

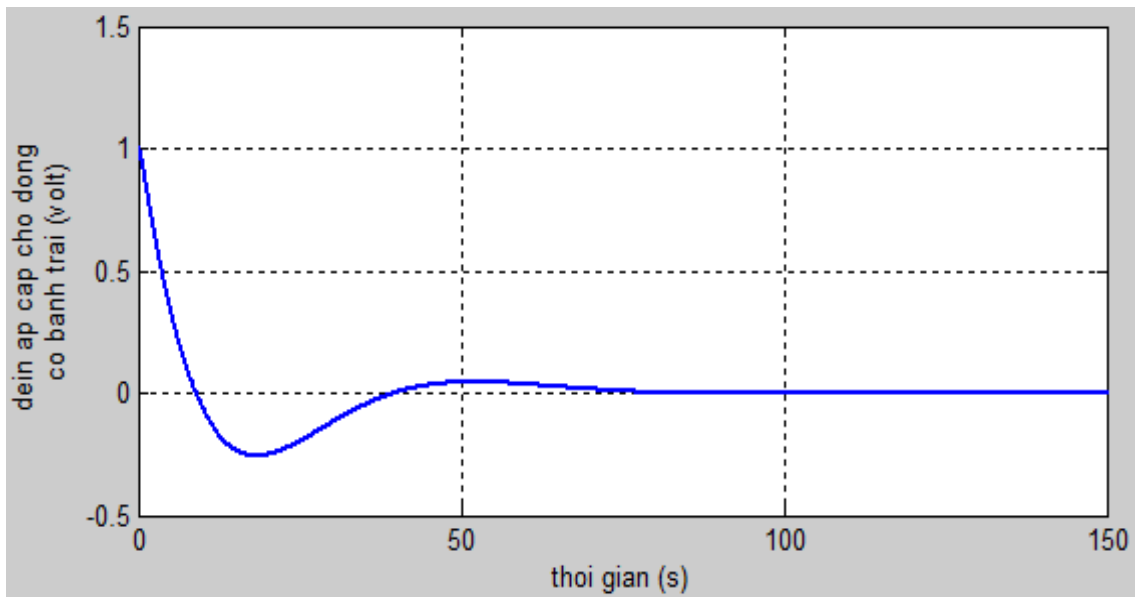
4.2.2.2 Giá trị đặt khác 0

Chọn giá trị đặt của teta là 3.14 rad, giá trị đặt của phi là 1.57(rad) (xe tiến tới 1 đoạn bằng chiều quay của một vòng hai bánh xe và quay 1 góc 90 độ quanh trục của xe)

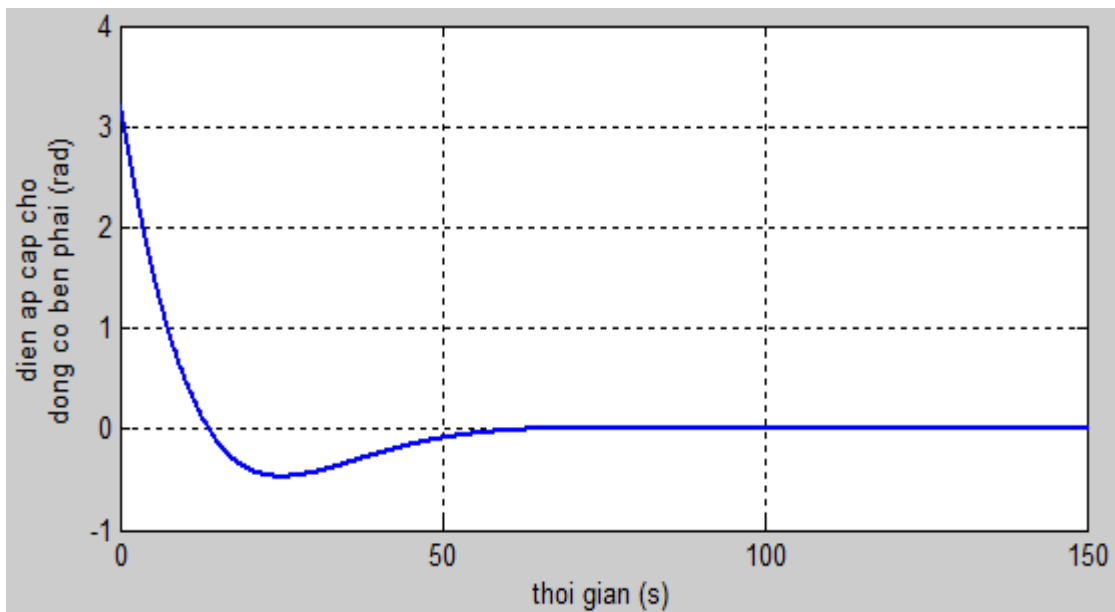


Hình 4.12: Góc bánh xe θ (rad)

Hình 4.13: Góc nghiêng ψ (rad)Hình 4.14: Góc xoay ϕ (rad)



Hình 4.15: Tín hiệu điện áp cấp cho động cơ trái (volt)



Hình 4.16: Tín hiệu điện áp cấp cho động cơ bánh phải (volt)

4.2.3 Nhận xét

Ta nhận thấy với các thông số của ma trận hồi tiếp K được chọn thì hệ thống ổn định. Tuy nhiên, ta hoàn toàn có thể chỉnh đỉnh từng thành phần của ma trận Q và ma trận R tương ứng để có thể ổn định hơn các thành phần tùy chọn trong các thành phần θ , ϕ , ψ , $\dot{\theta}$, $\dot{\psi}$, $\dot{\phi}$ và điện áp cấp cho từng thành phần động cơ.

Ví dụ:

$$\text{ta có ma trận } Q = \begin{bmatrix} Q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_6 \end{bmatrix}; \text{ ma trận } R = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \text{ với } x = \begin{bmatrix} \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ \phi \\ \dot{\phi} \end{bmatrix}$$

Nếu ta muốn giá trị θ được nhanh chóng ổn định hơn, ít dao động hơn, ta có thể tăng tương đối Q_1 so với các thành phần khác. Nếu ta muốn ψ được nhanh chóng ổn định, ít dao động hơn thì ta tăng tương đối Q_3 so với các thành phần khác.

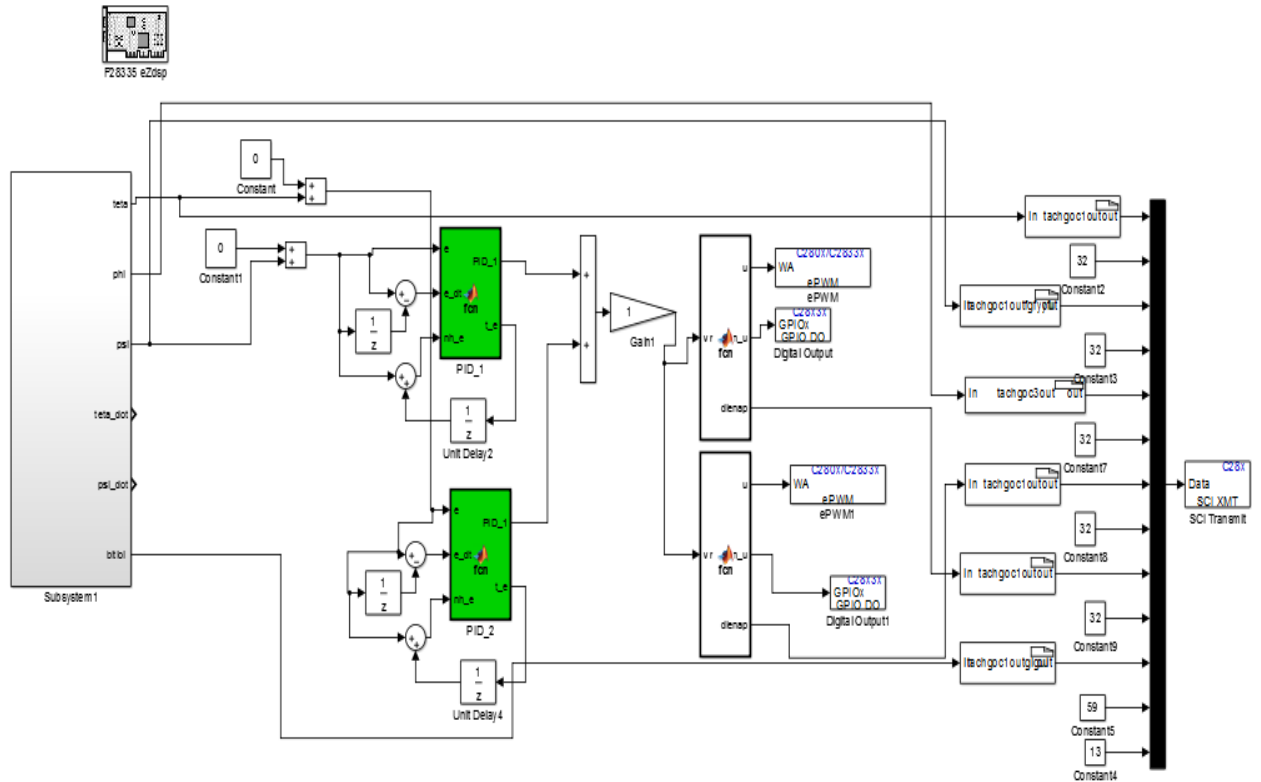
Điều khiển LQR cho đáp ứng tốt hơn hẳn đáp ứng bộ điều khiển PID. Mặt khác, LQR có thể điều khiển được hệ có số ngõ vào và ngõ ra là tùy ý. Trong khi đó, giải thuật điều khiển PID chỉ áp dụng được cho hệ có số ngõ vào và ra la bằng nhau. Ngoài ra, điều khiển PID ở đây là điều khiển thỏa hiệp giữa góc psi và góc teta nên việc điều khiển chỉ tốt khi hệ thống hoạt động quanh vị trí cân bằng (teta=0, psi=0). Đối với LQR là điều khiển chính xác nên điều này được cải thiện hơn. Tuy nhiên, do LQR được thiết kế sau khi đã tuyến tính hóa hệ thống nên thực ra, LQR cũng chỉ hoạt động tốt quanh vị trí làm việc.

CHƯƠNG 5

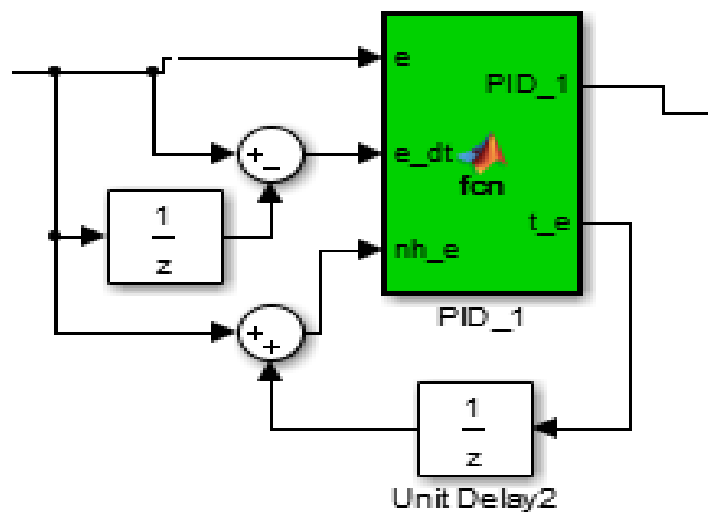
KẾT QUẢ THỰC NGHIỆM

5.1 Điều khiển PID

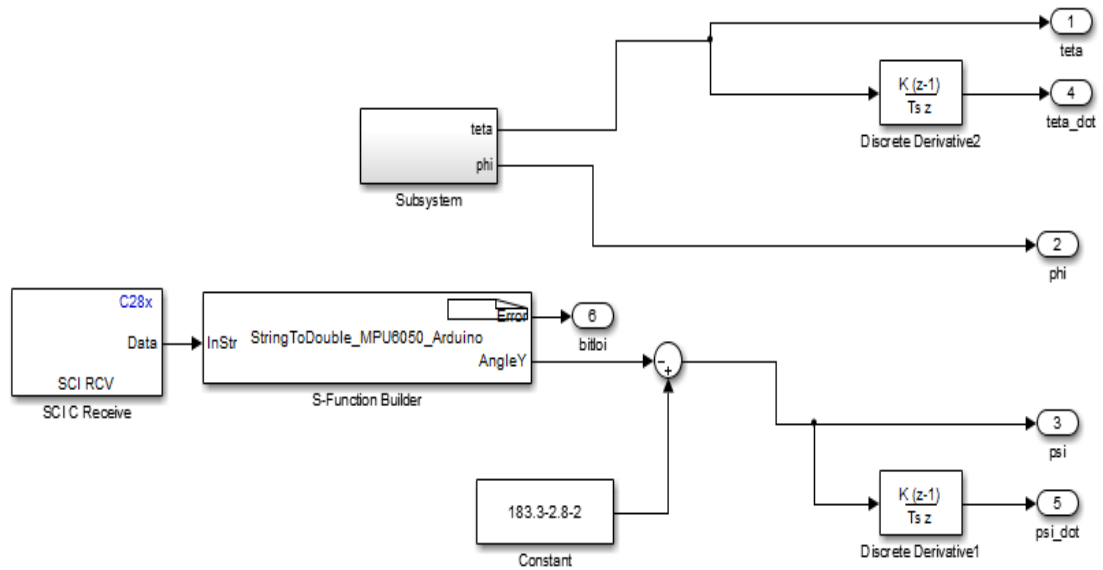
5.1.1 Điều khiển đứng yên tại vị trí cân bằng



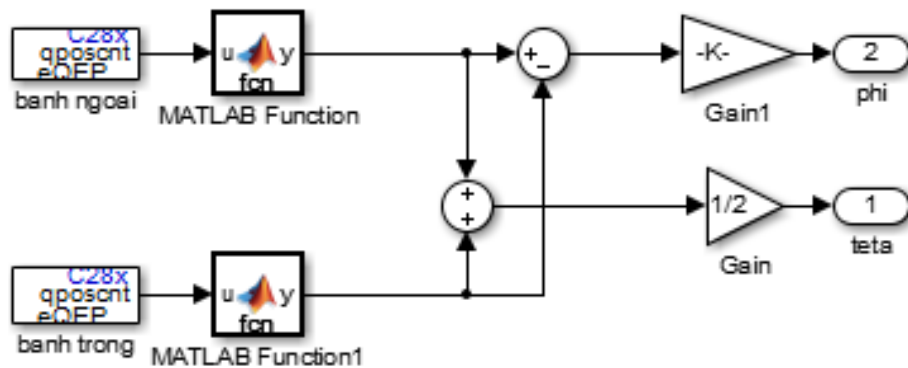
Hình 5.1: Chương trình thực tế điều khiển PID hệ thống



Hình 5.2: Khối điều khiển PID



Hình 5.3: Khối phân tích các thành phần psi, teta, phi

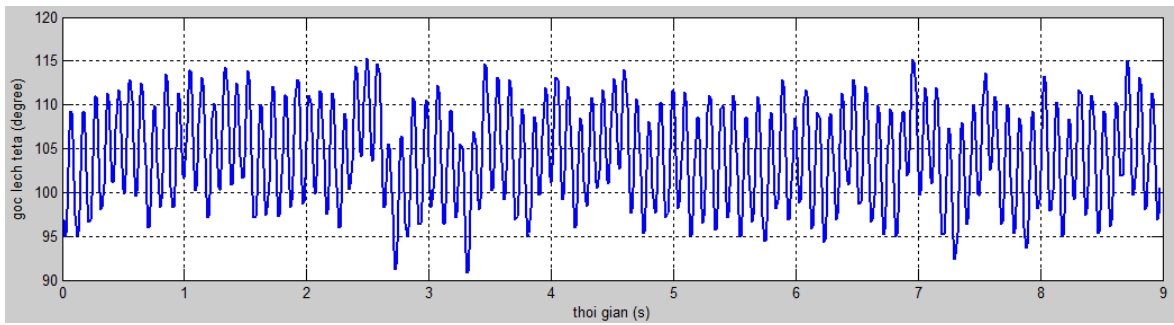


Hình 5.4: Khối phân tích để cho ra các thành phần phi và teta

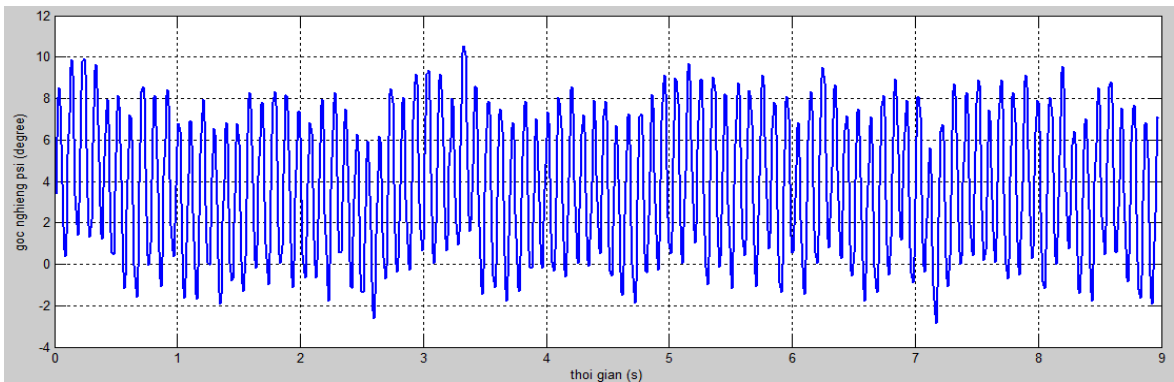
Việc thử sai các giá trị thông số điều khiển PID theo nguyên tắc. Nếu chọn K_{p1} , K_{i1} , K_{d1} lớn hơn tương đối so với K_{p2} , K_{i2} , K_{d2} thì độ ổn định và chính xác của góc nghiêng psi được ưu tiên và ngược lại. Nếu xe dao động quá mạnh tương ứng giảm K_p , tăng K_d . Nếu muốn giảm sai số xác lập thì tăng K_i . Nếu muốn tăng thời gian đáp ứng, giảm sai số xác lập thì tăng K_p . Qua quá trình thực nghiệm thử sai theo qui tắc trên, các thông số điều khiển được chọn là:

$$K_{p1}=150; K_{i1}=0; K_{d1}=10;$$

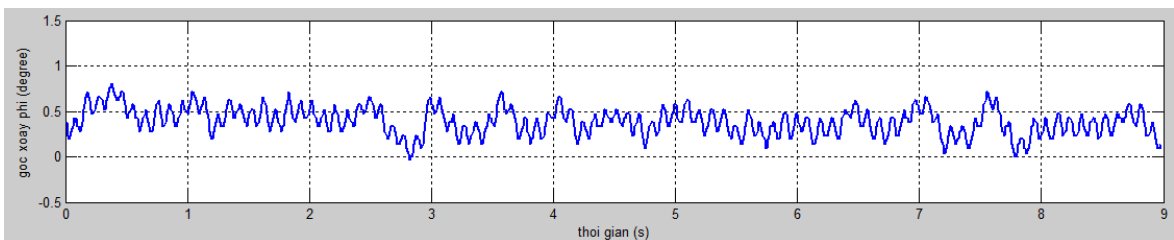
$$K_{p2}=3; K_{i2}=0; K_{d2}=2;$$



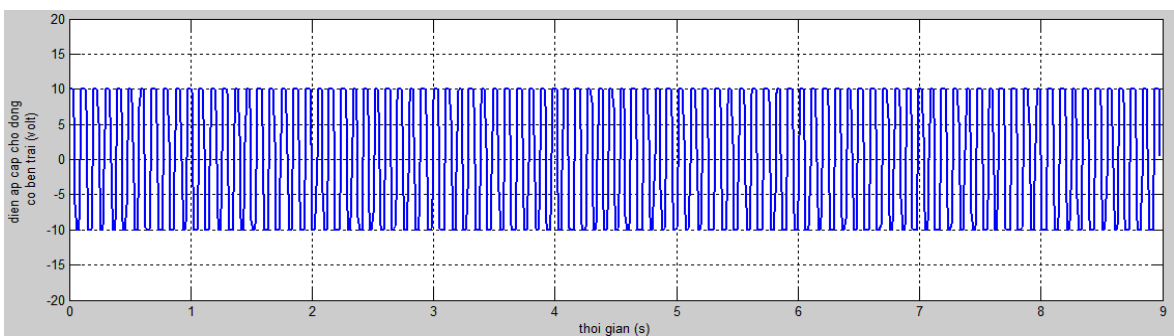
Hình 5.5: Góc teta (degree)



Hình 5.6: Góc psi (degree)



Hình 5.7: Góc phi (degree)



Hình 5.85: Điện áp cấp cho hai động cơ (volt)

5.1.2 Nhận xét

Ta thấy góc tới teta vẫn ổn định nhưng dao động quanh giá trị 100^0 . Một vòng bánh xe là 360^0 nên sự sai lệch này tương ứng gần $1/3$ vòng bánh xe hai bánh cân

bằng. Sự sai lệch này không đáng kể. Xe có xu hướng luôn ngã về phía trước (góc nghiêng psi dao động quanh góc 2 độ). Góc xoay lệch khoảng 5 độ (rất nhỏ). Các sự sai lệch này bù trừ cho nhau để xe vẫn cân bằng vì điều khiển PID ta đang xét là PID kết hợp. Qua kết quả thực nghiệm, ta nhận thấy kết quả thực tế là xe vẫn ổn định nhưng không về xác lập được là vì mô hình cơ khí của xe còn chưa hoàn thiện nên tự bản thân xe bị rung trong quá trình hoạt động. Ngoài ra, thời gian thử sai cũng chưa đủ nhiều để đạt giá trị điều khiển PID tốt hơn.

5.2 Điều khiển LQR

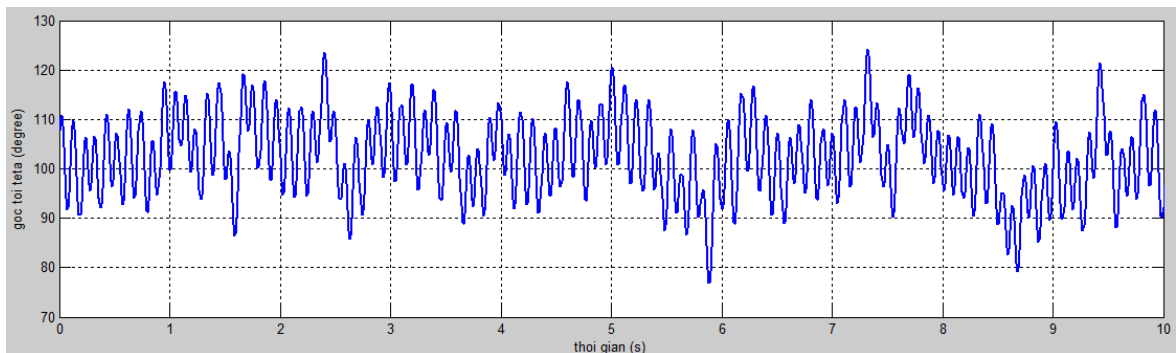
Trong khuôn khổ điều khiển LQR, ta chọn ma trận:

$$Q = \begin{bmatrix} 100000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}, \text{ ma trận } R = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}.$$

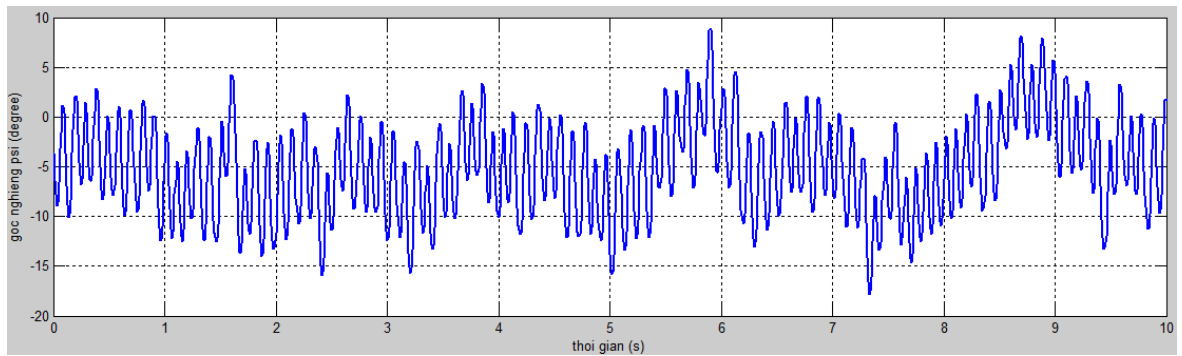
Như vậy, với thông số hệ thống thực tế, ta tính toán ra

$$K = \begin{bmatrix} -50 & -18.0527 & -902.7874 & -152.2351 & -158.1139 & -16.0239 \\ -50 & -18.0527 & -902.7874 & -152.2351 & 158.1139 & -16.0239 \end{bmatrix}$$

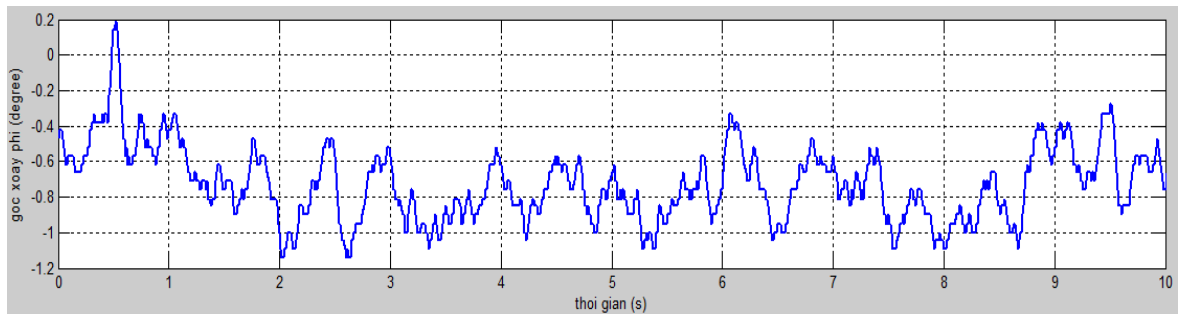
5.2.1 Điều khiển đứng yên tại vị trí cân bằng



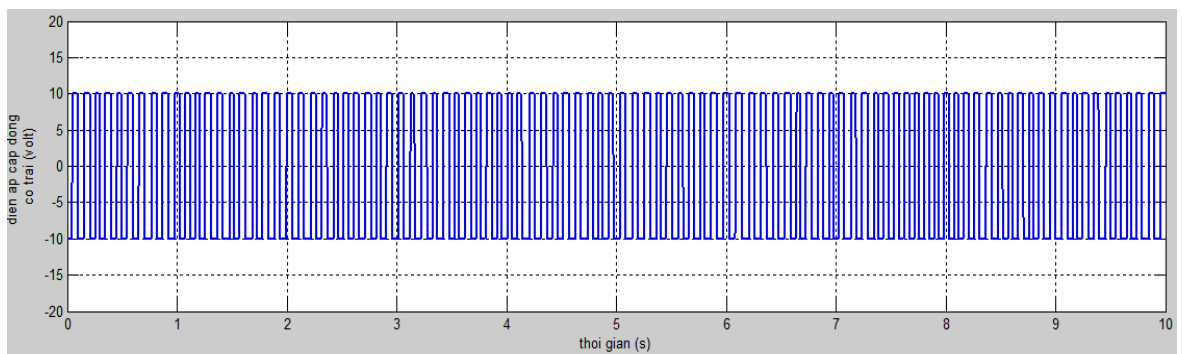
Hình 5.9 : Góc tới teta (độ)



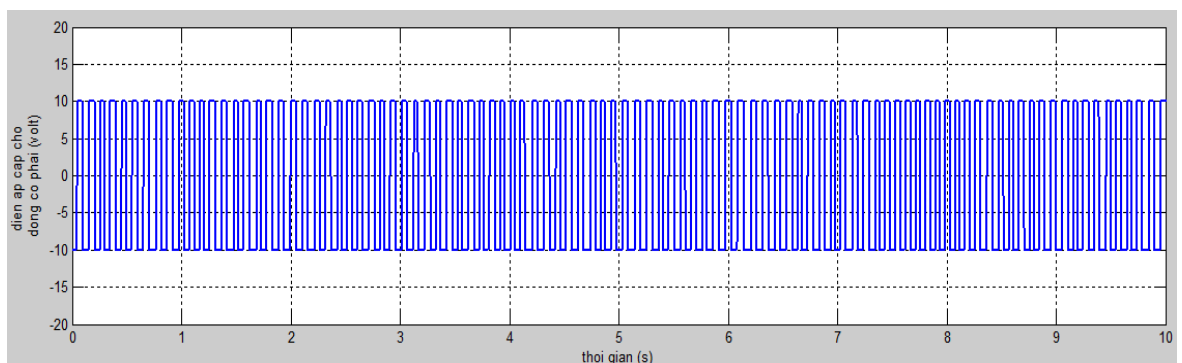
Hình 5.10: Góc nghiêng ψ (độ)



Hình 5.11: Góc xoay ϕ (độ)



Hình 5.12: Điện áp cấp cho động cơ trái (volt)



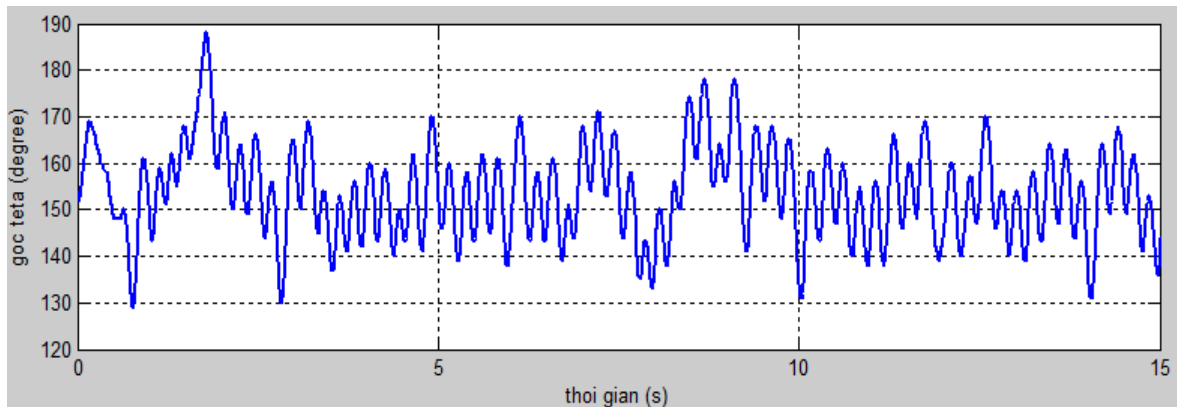
Hình 5.13: Điện áp cấp cho động cơ phải (volt)

5.2.2 Nhận xét: Ta thấy với các thông số K vừa tìm được thì bộ điều khiển LQR cho đáp ứng tốt nhưng góc teta vẫn còn lệch 100° so với giá trị 0 (tức sai lệch chưa tới 1/3 vòng quay của hai bánh xe vì một bánh xe quay hết 1 vòng tức 360°). Góc xoay phi dao động quanh vị trí 0.8° . Góc nghiêng teta còn lệch 1 góc 5° so với phương thẳng đứng. Việc lựa chọn thông số K rất quan trọng và mang tính bù trừ cho các thông số biến trạng thái (ví dụ muốn teta ổn định, chính xác thì tang K_1 lên, tuy nhiên làm như vậy sẽ làm các biến trạng thái khác bị điều khiển kém đi...). Trên cơ sở đó, ta cần thử sai để chọn được bộ trọng số K phù hợp. Các kết quả thực nghiệm cho thấy hệ xe cân bằng. Tuy nhiên, đáp ứng ngõ ra còn dao động vì thông số điều khiển không thể chọn được hoàn toàn chính xác vì mô hình cơ khí không chính xác hoàn toàn như lý thuyết được (ví dụ theo lý thuyết xe là một khối hộp đặc nhưng thực tế thì không được như vậy, các thông số động cơ, hệ số ma sát... là rất khó xác định) và việc tinh chỉnh cảm biến độ nghiêng ban đầu là có sai lệch.

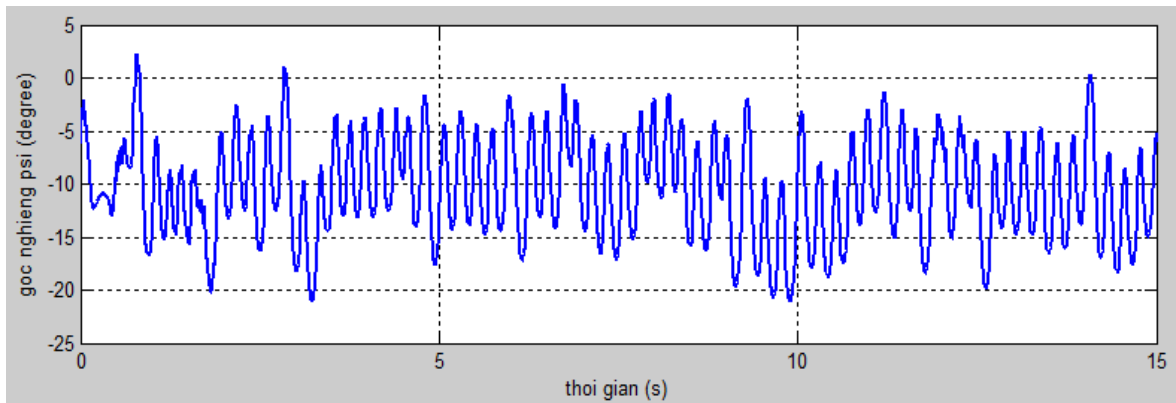
5.2.3 Điều khiển với vị trí đặt khác 0

Thí nghiệm 1: Điều khiển queo trái. Chọn góc đặt queo trái là 90° ($\varphi_{set} = 90^\circ$).

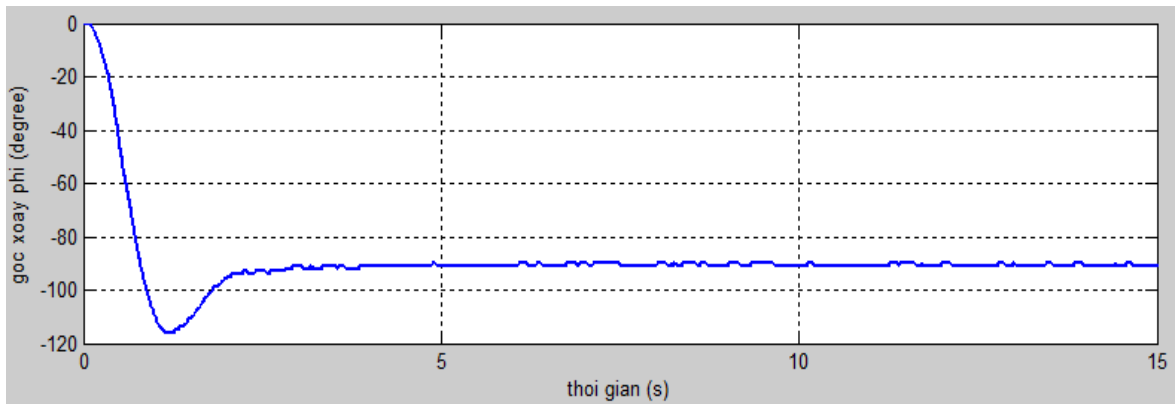
Cũng với thông số K như ở mục 5.2.1, ta có kết quả thực nghiệm như sau:



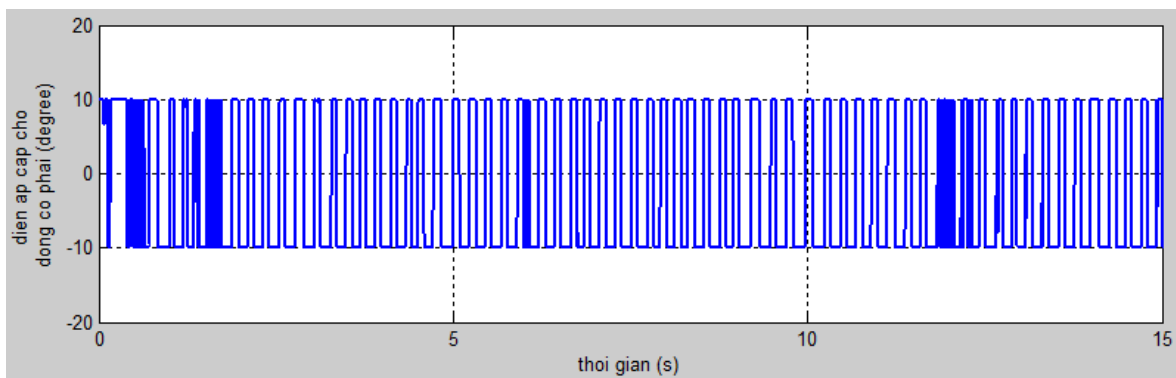
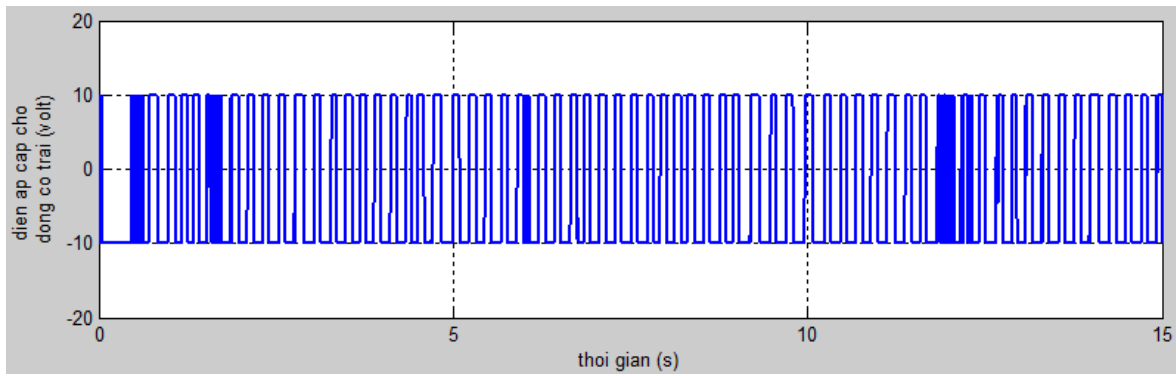
Hình 5.14: Góc tới teta (độ)



Hình 5.15: Góc nghiêng ψ (độ)



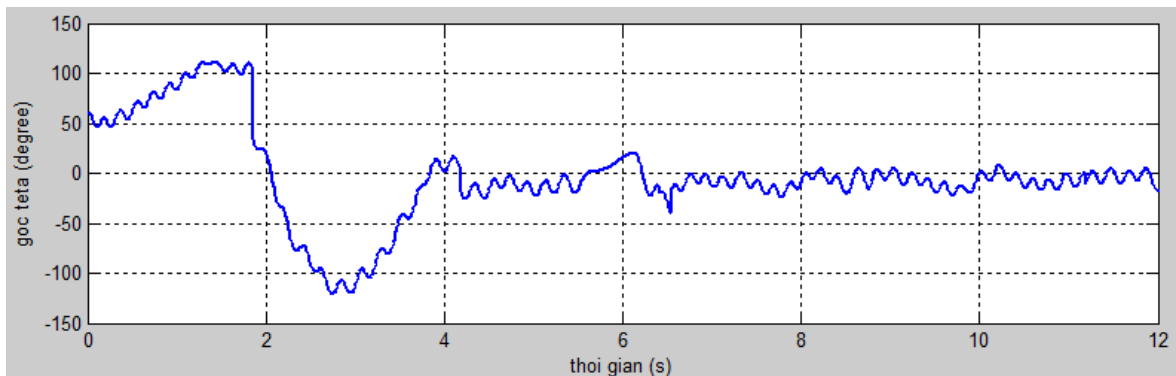
Hình 5.16: Góc xoay ϕ (độ)



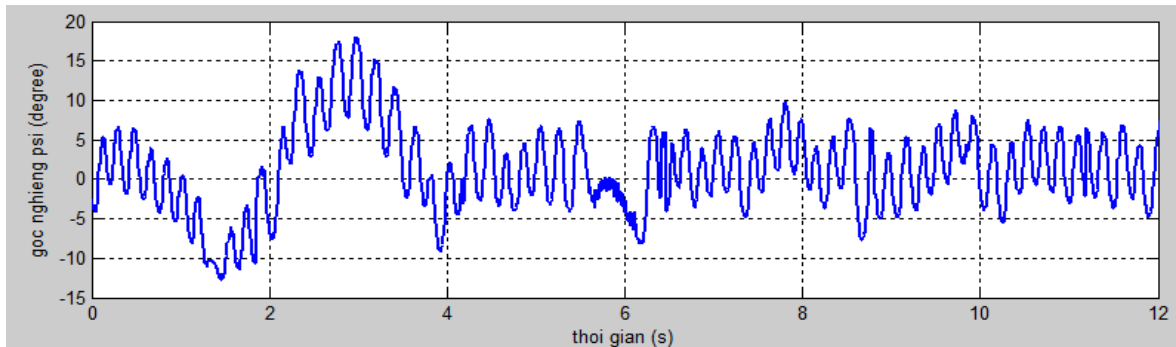
Hình 5.17: Điện áp cấp cho động cơ phải (volt)

Nhận xét: Giá trị xác lập của góc phi là 90^0 , trùng khớp với giá trị set_point. Điều này cho thấy thông số K được chọn là rất phù hợp để điều khiển góc xoay trái, giữ được cân bằng cho hệ thống. Tuy nhiên, góc nghiêng psi và góc tới teta còn lệch nhiều so với giá trị xác lập (tương ứng 10^0 và 150^0). Tức xe có xu hướng lệch về phía trước khoảng 10^0 và bánh xe không ở đúng vị trí cân bằng mà có xu hướng di chuyển lên khoảng hơn 1/3 vòng của bánh xe (vì một vòng quay của hai bánh là 360^0). Trong thí nghiệm này, xe đã xoay trái thành công.

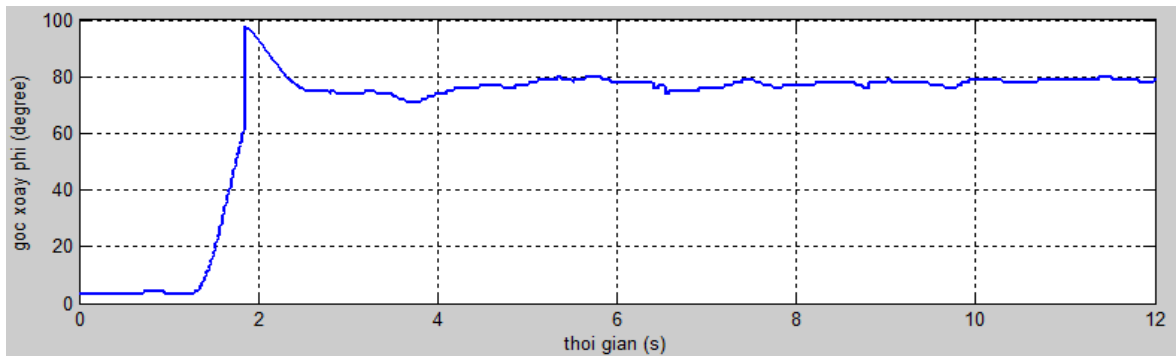
Thí nghiệm 2: Điều khiển quẹo phải, giá trị góc đặt φ là -90^0



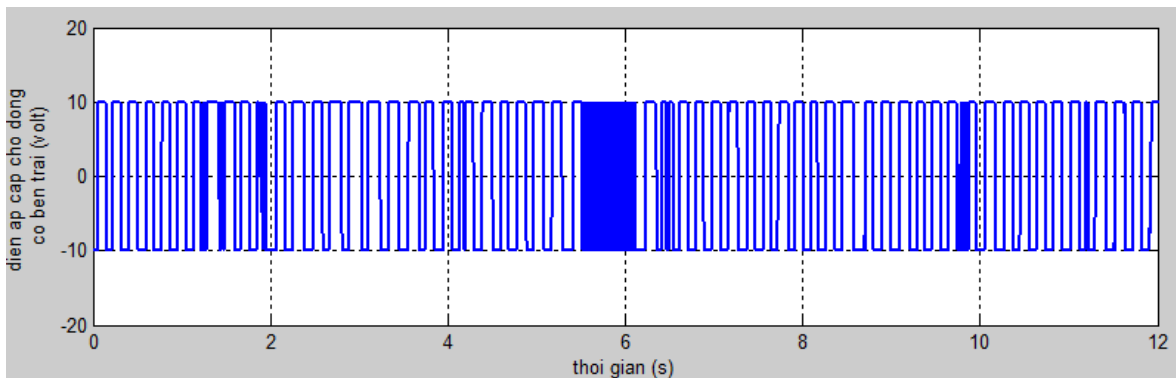
Hình 5.18: Góc tới teta (độ)



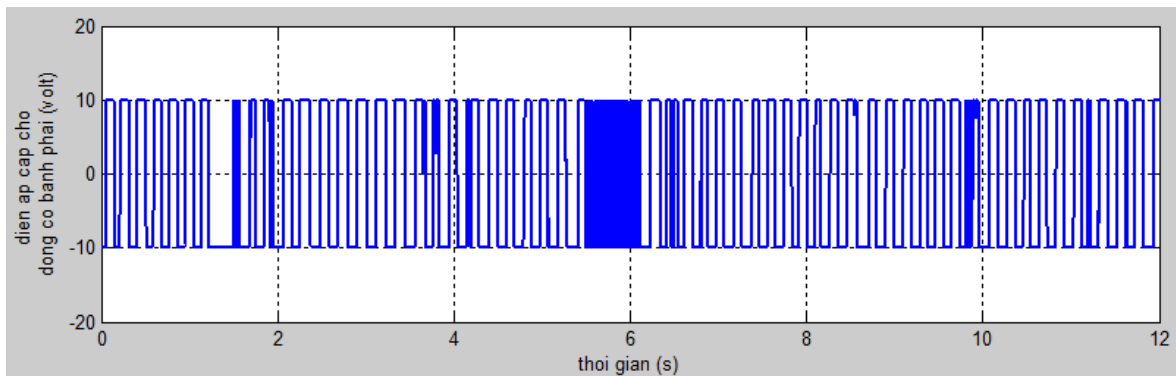
Hình 5.19: Góc nghiêng psi (độ)



Hình 5.20: Góc xoay phi (độ)



Hình 5.21: Điện áp cấp cho động cơ trái (volt)

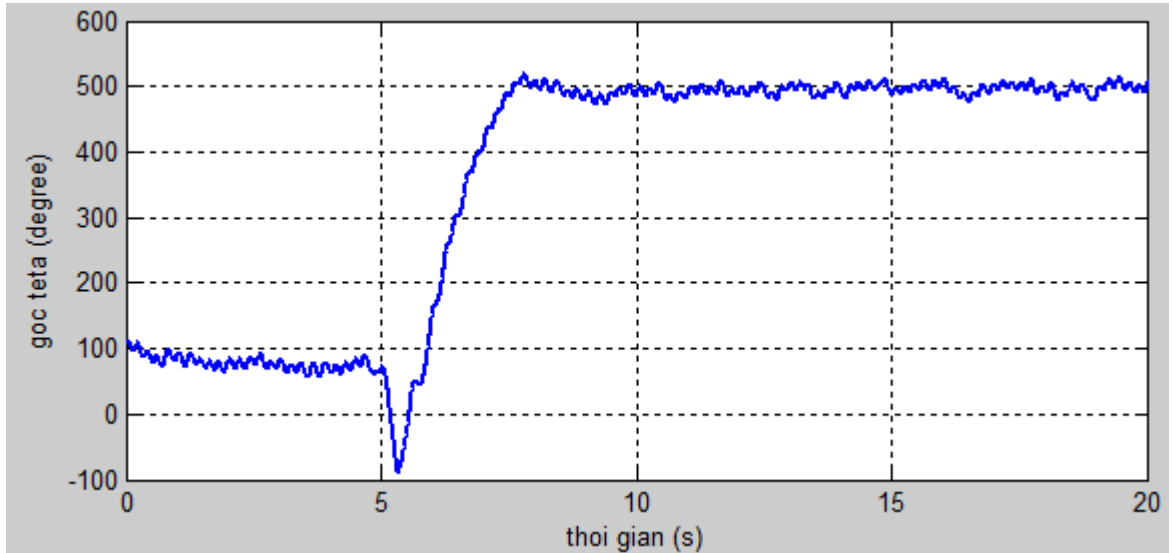


Hình 5.22: Điện áp cấp cho động cơ phải (volt)

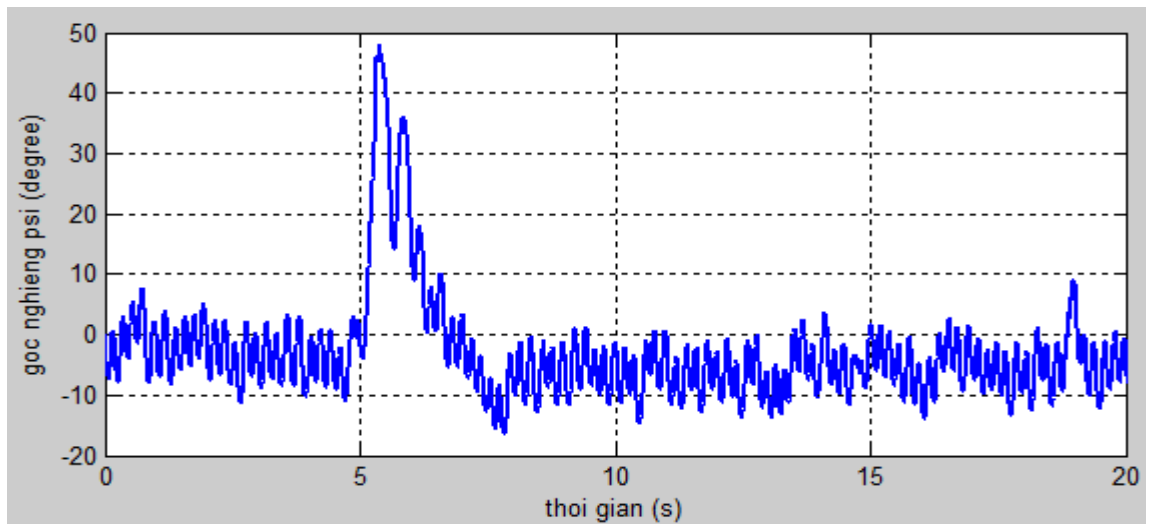
Nhận xét: Ta thấy giá trị φ thực tế bị lệch so với giá trị đặt khoảng 10^0 (80^0 so với 90^0) và có xuất hiện vọt lố. Tuy nhiên, giá trị góc nghiêng và góc tới luôn dao động quanh 0^0 và hệ thống giữ được ổn định. Như vậy, thông số điều khiển điều khiển tốt giá trị góc xoay phi theo yêu cầu nhưng vẫn còn tồn tại sai lệch. Các biến trạng thái còn lại đều hội tụ rất tốt. trong trường hợp này xe đã quẹo phải thành công.

Thí nghiệm 3: Điều khiển đi tới, giá trị góc tới đặt là 400^0 , tức ta muốn xe di chuyển tới khoảng hơn 1 vòng bánh xe (của cả hai bánh) (một vòng bánh xe di chuyển là

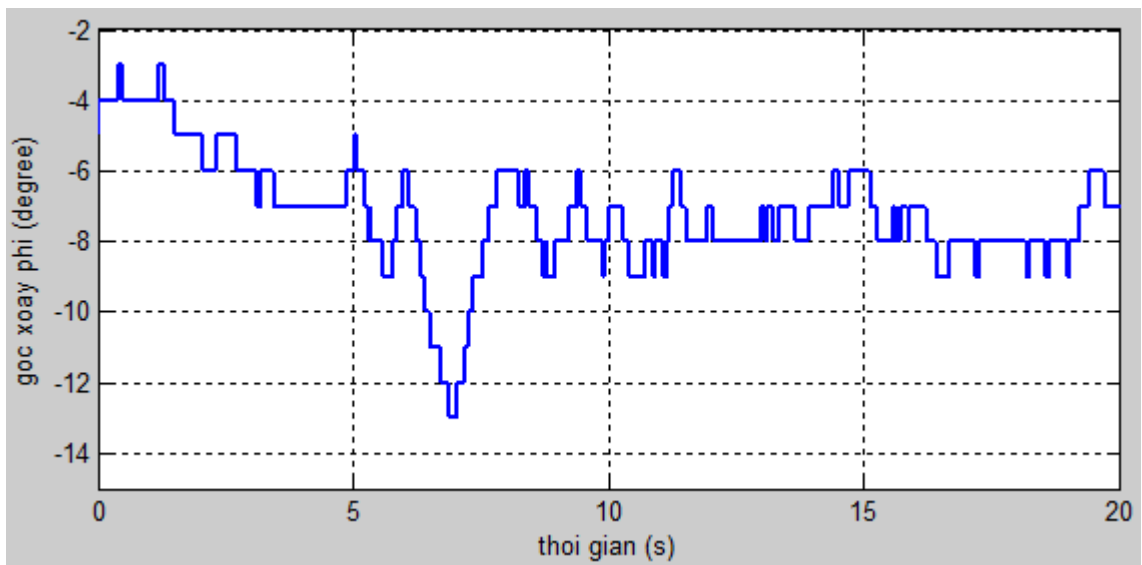
360^0). Sau 5s, bộ điều khiển tiến hành thực iện tác vụ đi tới. Dưới đây là kết quả thực nghiệm.



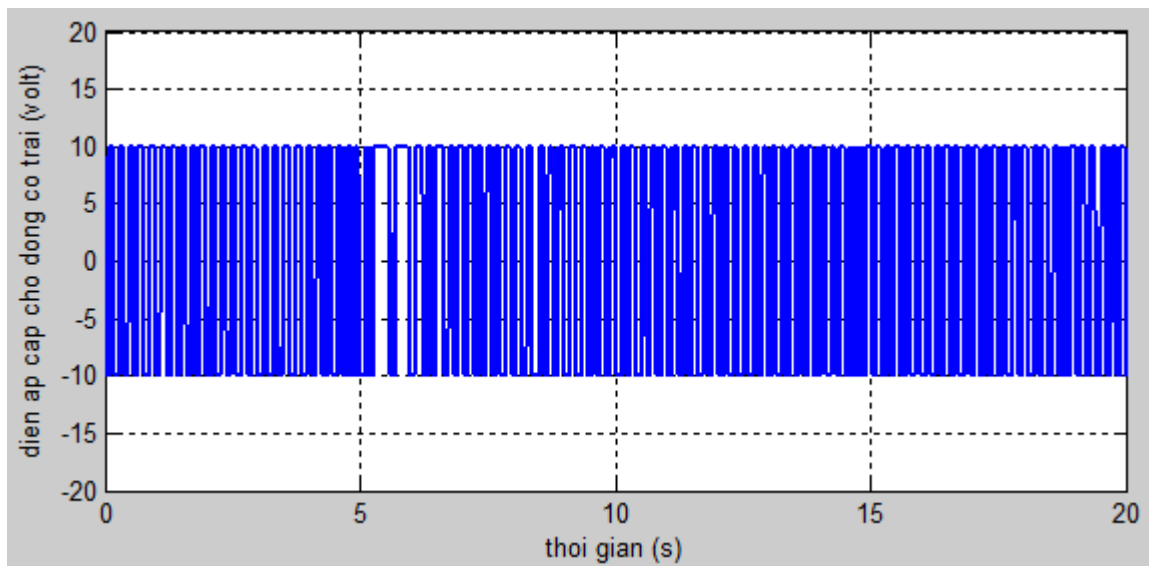
Hình 5.23 : Góc tới teta (độ)



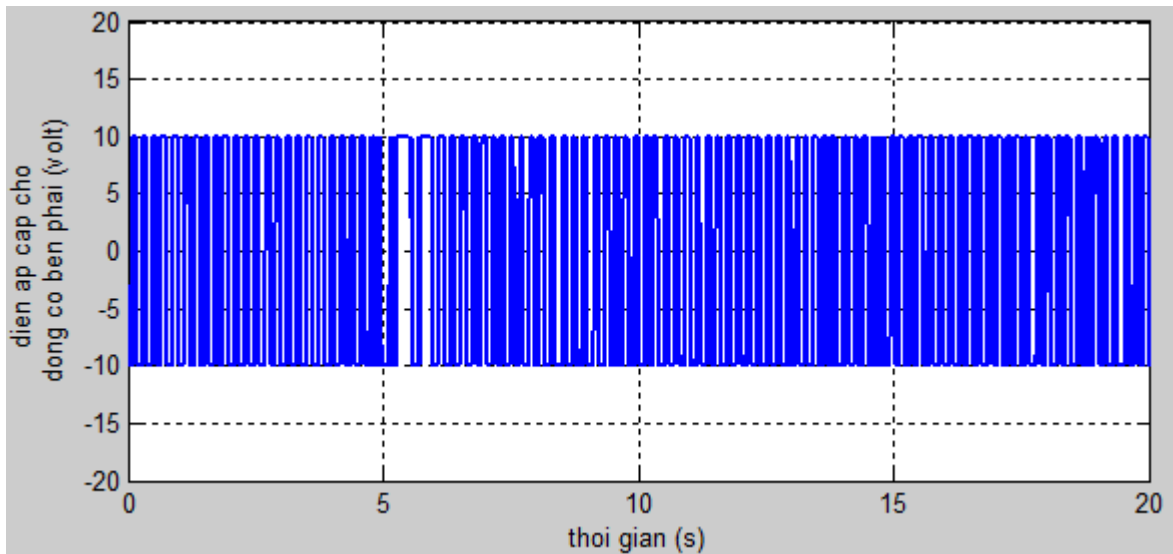
Hình 5.24: Góc nghiêng psi (độ)



Hình 5.25: Góc xoay phi (độ)



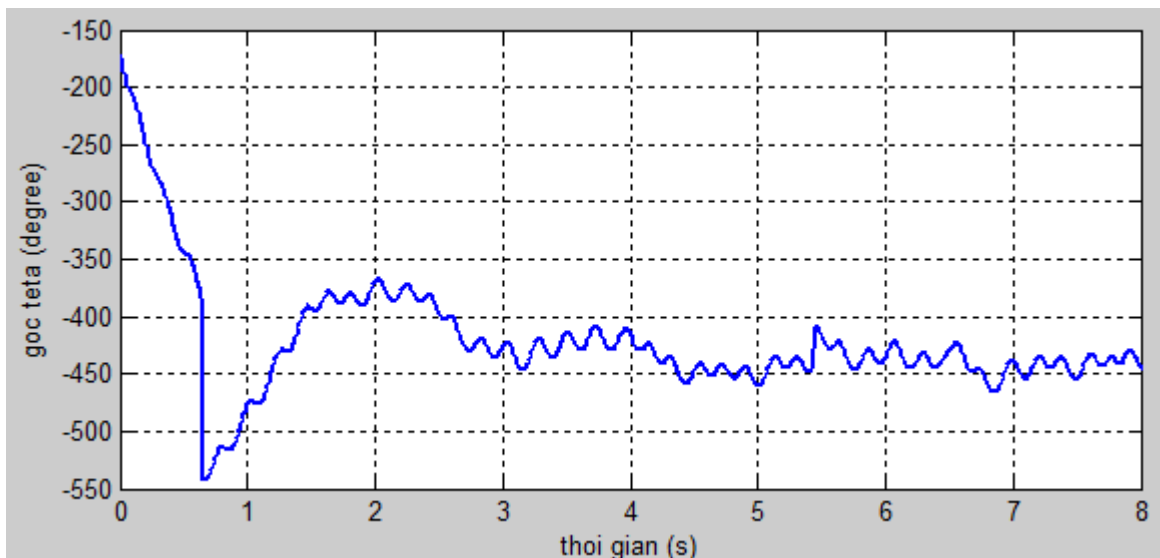
Hình 5.26: Điện áp cấp cho động cơ bên trái (volt)



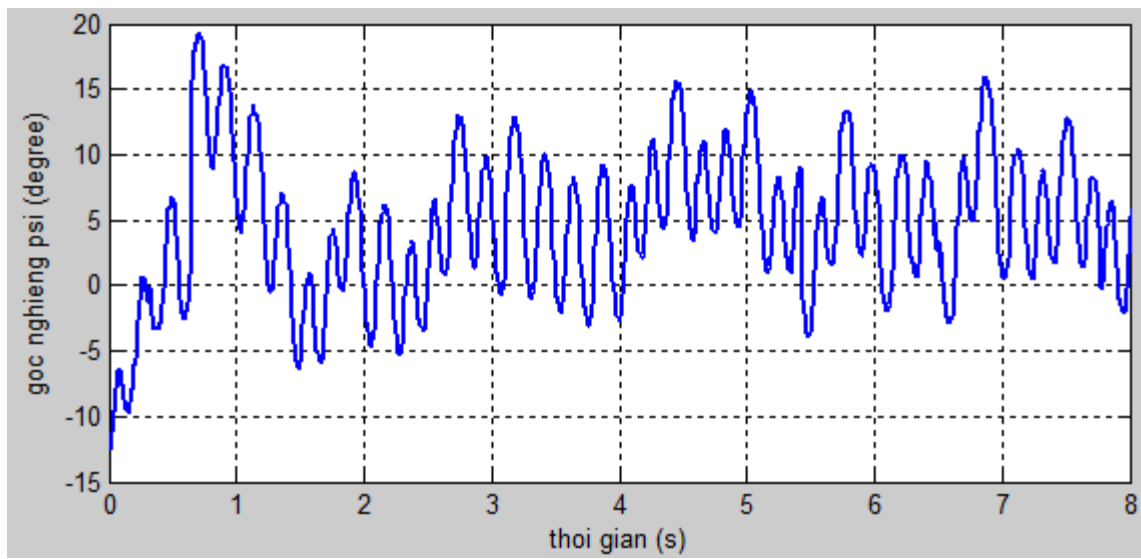
Hình 5.27: Điện áp cấp cho động cơ bên phải (volt)

Nhận xét: Tại thời gian 5s, bộ điều khiển tiến hành cho xe đi tới. Tuy độ vọt lố của góc teta không lớn nhưng sai số xác lập là khá tương đối (500^0 so với giá trị đặt là 400^0). Giá trị góc xoay còn lệch khoảng 8^0 so với phương chuẩn. Hệ thống vẫn giữ ổn định đối với thông số K được chọn. Trong thí nghiệm này, xe đã đi tới thành công.

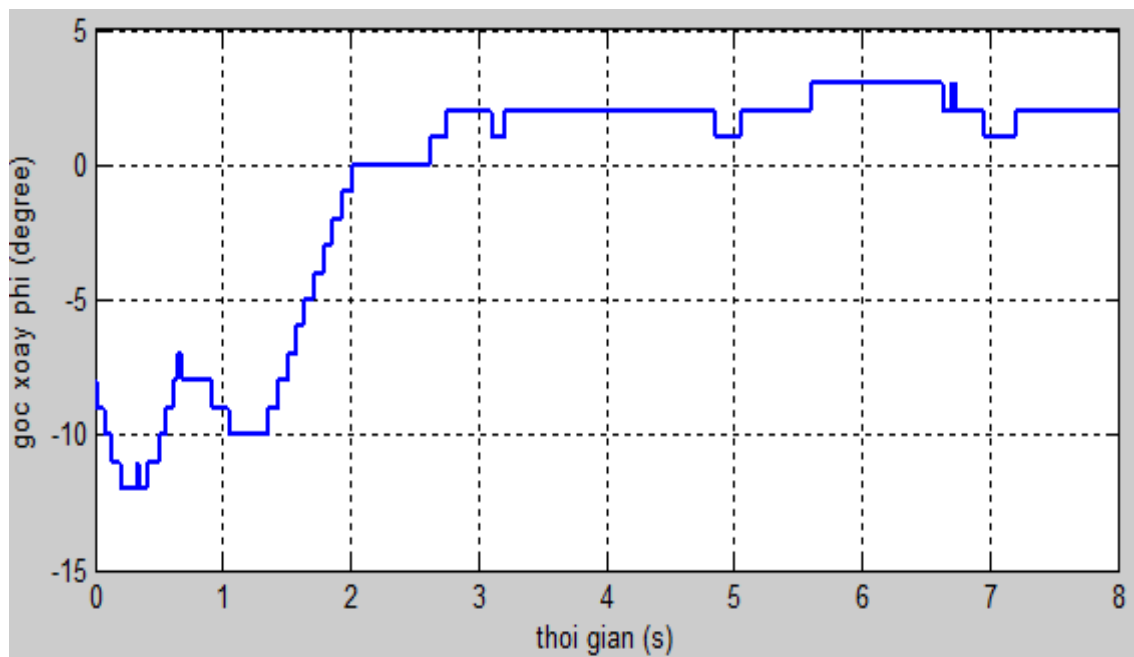
Thí nghiệm 4: Điều khiển đi lùi, giá trị lùi là 400^0 (tức xe sẽ lùi lại khoảng hơn 1 vòng bánh xe). Các thành phần góc nghiêng, góc xoay là không đổi. Dưới đây là kết quả thực nghiệm.



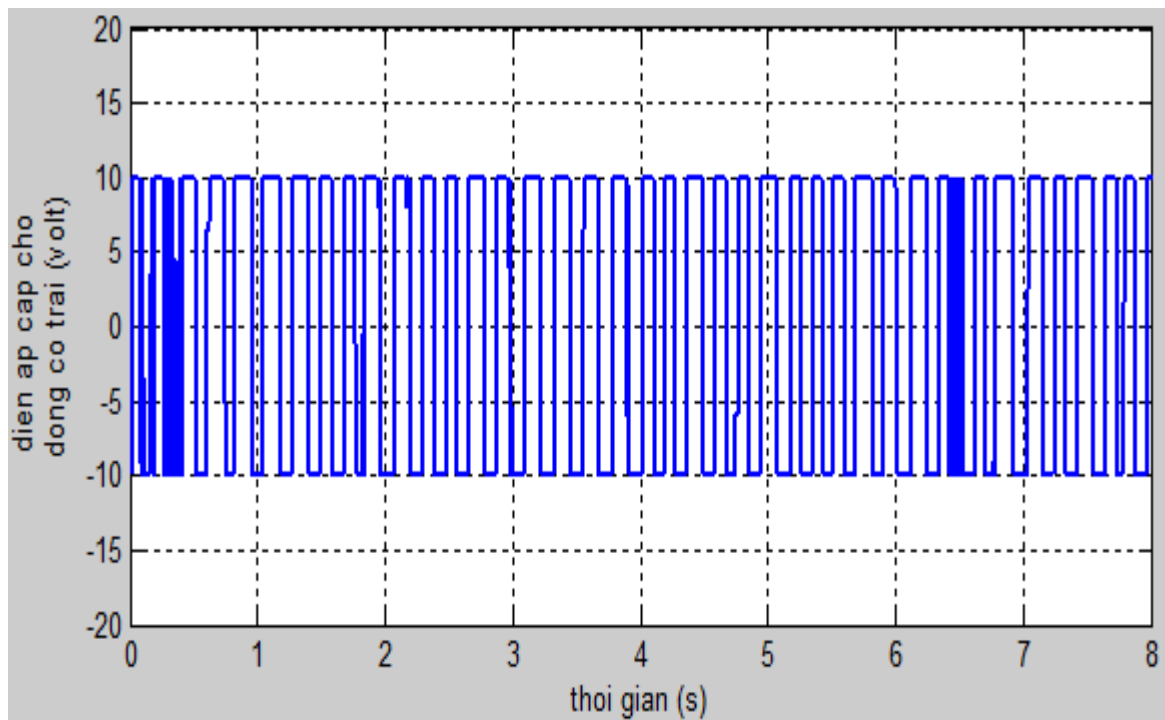
Hình 5.28: Góc tới teta (độ)



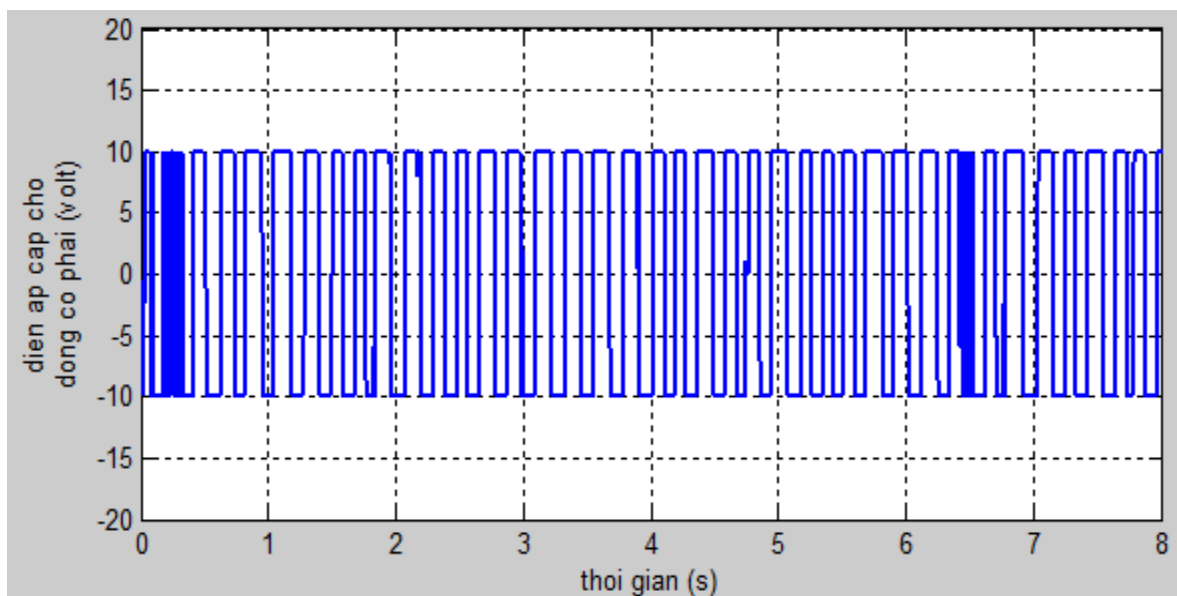
Hình 5.29: Góc nghiêng ψ (độ)



Hình 5.30: Góc xoay ϕ (độ)



Hình 5.31: Điện áp cấp cho động cơ trái (volt)



Hình 5.32: Điện áp cấp cho động cơ phải (volt)

Nhận xét: Ta thấy với thông số K được chọn, giá trị góc tới thực tế vẫn lệch 50° so với góc đặt (450° so với góc đặt là 400°). Các giá trị góc nghiêng và góc xoay cũng sai lệch. Hệ thống vẫn giữ được ổn định. Trong thí nghiệm này xe, đã đi lùi thành công.

5.2.4 Nhận xét:

Trong các trường hợp điều khiển đi tới, đi lùi, quẹo trái, quẹo phải, hệ xe hai bánh cân bằng thực hiện tốt (giá trị góc teta trong trường hợp tới lùi bám sát giá trị đặt, sai lệch ít hơn 50^0 , giá trị góc phi trong trường hợp xoay trái phải bám sát giá trị đặt, sai lệch ít hơn 10^0). Điều khiển LQR là điều khiển thỏa hiệp nên ta có thể tinh chỉnh các trọng số ma trận K tương ứng các biến trạng thái mà ta quan tâm để vẫn đạt giá trị điều khiển tối ưu hơn. Các thông số tìm được trong mô phỏng không thể áp dụng cho đối tượng thực tế vì việc tìm kiếm thông số mô hình thực tế là rất khó khăn và trong trường hợp này, tác giả đã giả lập nhiều giá trị nên thông số điều khiển K chỉ đạt giá trị gần chính xác chứ không chính xác hoàn toàn. Từ đó, kết quả thực nghiệm và kết quả mô phỏng cũng không giống nhau. Tuy nhiên, trong các trường hợp mô phỏng và thực nghiệm thì hệ xe vẫn cân bằng ổn định.

5.3 Chương trình thu thập dữ liệu

Ở đây, tác giả sử dụng phần mềm Hyperterminal để nhận tín hiệu từ DSP gửi lên máy tính trong quá trình hoạt động. trên JTAG của DSP đã có IC chuyển đổi giao tiếp từ cổng USB sang cổng RS232. Do đó, dữ liệu có thể chuyển đổi qua lại giữa máy tính và DSP thông qua cổng USB.

CHƯƠNG 6

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Trong khuôn khổ luận văn, học viên đã hoàn thành các nhiệm vụ sau:

- Xây dựng thành công chương trình mô phỏng điều khiển PID và điều khiển LQR cho hệ xe hai bánh tự cân bằng ổn định quanh vị trí làm việc. .
- Xây dựng thành công mô hình thực tế (cơ khí và mạch điện) đối với một hệ thống xe hai bánh tự cân bằng.
- Điều khiển thành công hệ xe hai bánh tự cân bằng ổn định quanh vị trí cân bằng theo giải thuật PID và LQR
- Thiết kế thành công giao diện giao tiếp giữa máy tính và vi xử lý DSP để điều khiển LQR và PID cho hệ xe hai bánh tự cân bằng.

Các hạn chế của đề tài:

- Do thời gian eo hẹp nên học viên chưa chọn được thông số điều khiển PID và LQR tối ưu hơn.
- Chương trình truyền nhận dữ liệu giữa máy tính và DSP còn bị treo trong một số trường hợp.

Điều khiển LQR là phương pháp điều khiển rất tốt nếu ta biết rõ phương trình toán học của hệ thống và biết rõ các thông số mô hình. Đây là bộ điều khiển đảm bảo hệ thống hoạt động tốt quanh vị trí làm việc. Một hệ MIMO bất kì nếu thỏa các điều kiện vừa nêu ra thì đều có thể điều khiển được bằng giải thuật LQR. Ưu điểm của giải thuật LQR là đơn giản, thực hiện được cho hệ có số ngõ vào ra là bất kì. Nhược điểm chính của giải thuật LQR là ta cần có phương trình toán học chính xác cũng như thông số hệ thống của đối tượng.

Tuy nhiên, nếu một hệ thống chưa biết rõ thông số, có thể tạm chấp nhận số ngõ ra hệ thống bằng số ngõ vào (trong trường hợp không thỏa mãn như trên thì vẫn có thể dùng PID thỏa hiệp), ta hoàn toàn có thể điều khiển hệ thống bằng giải thuật PID thông thường. Tuy nhiên, thời gian thử sai là đáng kể trên cơ sở dựa vào đáp ứng ngõ ra mà tinh chỉnh K_p , K_i , K_d sao cho phù hợp.

Luận văn đã điều khiển thành công giải thuật LQR và PID cho hệ xe hai bánh cân bằng, từ đó đúc kết được phương pháp tính toán chọn lựa phù hợp để có thông số điều khiển được hệ thống, từ đó phát triển thêm các ứng dụng cho hệ xe hai bánh. Mặt khác, qua quá trình điều khiển thì các ưu khuyết điểm của giải thuật PID và LQR cũng được xem xét, kiểm tra và kết luận, kết quả được trình bày trong luận văn.

6.2 Hướng phát triển

Đề tài xe hai bánh tự cân bằng tuy đã được nghiên cứu nhiều trên thế giới và cũng đã được thực hiện ở Việt Nam nhưng vẫn còn nhiều hướng để mở rộng:

- Điều khiển thích nghi để hệ thống xe có thể thực nghiệm cho người đứng lên với các cân nặng khác nhau và thay đổi trong quá trình hoạt động của xe.
- Nhận dạng thông số hệ thống bằng các phương pháp nhận dạng để có thể áp dụng giải thuật LQR.
- Ứng dụng các giải thuật điều khiển thông minh như mờ, dùng mạng neuron,... trong điều khiển hệ thống.
- Nâng cấp giải thuật điều khiển để áp dụng cho hệ xe một bánh tự cân bằng...
- Hướng phát triển sắp tới mà học viên dự kiến là sẽ áp dụng các phương pháp thích nghi, tự dò tìm thông số điều khiển phù hợp thông qua quá trình điều khiển online. Ngoài ra, một hướng phát triển khác là dùng các phương pháp nhận dạng để xác định thông số mô hình động cơ. Từ đó, tính toán giá trị LQR phù hợp nhất để có thể điều khiển xe chạy tới, lui, quẹo trái, quẹo phải trong thực tế.

TÀI LIỆU THAM KHẢO

- [1] Huỳnh Thái Hoàng, *Lý thuyết điều khiển tự động*, Nhà xuất bản Đại học Quốc Gia, 2005.
- [2] Huỳnh Thái Hoàng; *Mô Hình Hóa Và Nhận Dạng Hệ Thống*, Đại học bách khoa Tp. HCM.
- [3] Nguyễn Phùng Quang; *Matlab và Simulink*, NXB khoa học và kỹ thuật.
- [4] Dương Hoài Nghĩa, *Điều khiển hệ thống đa biến*, Nhà xuất bản Đại học Quốc Gia
- [5] Nguyễn Thị Phương Hà, Huỳnh Thái Hoàng, *Lý thuyết điều khiển hiện đại*, NXB Đại học quốc gia TP.HCM, 2007.
- [6] Nguyễn Trung Hiếu, *Điều khiển robot hai bánh tự cân bằng trên địa hình không phẳng*, luận văn cao học Đại học Bách Khoa TPHCM, 2012.
- [7] Hoàng Anh Vũ, *Robot hai bánh tự cân bằng*, luận văn cao học Đại học Bách Khoa TPHCM, 2012.
- [8] Cù Minh Phước, Nguyễn Khắc Mẫn, *Xây dựng chế tạo robot hai bánh tự cân bằng*, đồ án tốt nghiệp Đại học SPKT TPHCM, 2013.
- [9] <http://arduino.cc/>
- [10] A. N. K. Nasir, M. A. Ahmad, R. M. T. Raja Ismail, *The Control of a Highly Nonlinear Two-wheels Balancing Robot: A Comparative Assesment between LQR and PID-PID Control Schemes*, World Acadamy of Science, Engineering and Technology 46, 2010.
- [11] k. M. Goher, M. O. Tokhi and N. H. Siddique, *Dynamic Modeling and Control of a two wheeled Robotic Vehicle with a Virtual Payload*, ARPN Journal of Engineering and Applied Science, 2011.
- [12] Spencer Burdette, *A Zilog ZNEO based Self-Balancing Robot with PID Control*, Master Thesis of The George Washigton University, 2007.
- [13] Peter Miller, *Building a Two Wheeled Balancing Robot*, Dissertation of University of Sourthern Queensland, 2008.

- [14] Maia R.Bageant, *Balancing a Two-wheeled Segway Robot*, Bachelor Thesis of Massacuhsetts Institute of Technology, 2011.
- [15] Mahadi Hasan, Chanchal Saha, Md. Mostafizur, Md. Rabiual Islam Sarker and Subrata K.Aditya, *Balancing of an Invertedd Pendulum Using PD Controller*, Dhaka University Journal of Science, 2012.
- [16] Umar Adeel, K. S. Alimgeer, Omair Inam, Ayesha Hmmed, Mehmood Qureshi, Mehmood Ashraf, *Autonomous Dual Wheel Self Balancing Robot Based on Microcontroller*, Journal of Basic and Applied Scientific Research, 2013.