

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH



ĐỒ ÁN TỐT NGHIỆP
NGÀNH CNKT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

SO SÁNH CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN CÂN
BẰNG CHO PENDUBOT

GVHD: TS. NGUYỄN VĂN ĐÔNG HẢI
SVTH: TRỊNH XUÂN CHINH
LÊ DIỆP THUỲ DƯƠNG



Tp.Hồ Chí Minh, tháng 8/2022

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



ĐỒ ÁN TỐT NGHIỆP

**SO SÁNH CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN
CÂN BẰNG CHO PENDUBOT**

SVTH: TRỊNH XUÂN CHINH **MSSV:** 18151055
LÊ DIỆP THUỲ DƯƠNG **MSSV:** 18151061
Khoa: 2018
Ngành: Công nghệ Kỹ thuật Điều khiển và Tự động hóa
GVHD: TS. NGUYỄN VĂN ĐÔNG HẢI

Tp. Hồ Chí Minh, tháng 08 năm 2022

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



ĐỒ ÁN TỐT NGHIỆP

**SO SÁNH CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN
CÂN BẰNG CHO PENDUBOT**

SVTH: TRỊNH XUÂN CHINH **MSSV:** 18151055
LÊ DIỆP THUỲ DƯƠNG **MSSV:** 18151061
Khoa: 2018
Ngành: Công nghệ Kỹ thuật Điều khiển và Tự động hóa
GVHD: TS. NGUYỄN VĂN ĐÔNG HẢI

Tp. Hồ Chí Minh, tháng 08 năm 2022

LỜI CẢM ƠN

Trong quá trình thực hiện đề tài, nhóm sinh viên đã nhận được sự giúp đỡ của Quý Thầy/Cô và các bạn, nhờ vậy đề tài đã hoàn thành đúng thời gian quy định. Những kiến thức và kinh nghiệm chúng em đã tích luỹ được trong quá trình làm đồ án này là vô cùng quý báu. Nhóm sinh viên đã trao đổi khả năng làm việc nhóm, phân quản lý công việc và quản lý thời gian. Và hơn hết, những kỹ năng cứng lẫn kỹ năng mềm này chính là hành trang để chúng em có thêm động lực để vững bước trong quãng đường sự nghiệp trong tương lai.

Chúng em xin chân thành gửi lời cảm ơn đến TS. Nguyễn Văn Đông Hải, người đã tận tình hướng dẫn, định hướng nghiên cứu, hỗ trợ các trang thiết bị, cùng vốn kiến thức chuyên môn để nhóm sinh viên có thể hoàn thành đồ án tốt nghiệp đúng thời gian.

Nhóm xin trân trọng gửi lời cảm ơn chân thành đến Quý Thầy/Cô Khoa Đào tạo Chất lượng cao và Khoa Điện – Điện tử đã cung cấp những kiến thức nền cơ bản.

Nhóm sinh viên xin gửi lời cảm ơn đến gia đình đã tạo điều kiện thuận lợi về vật chất cũng như tinh thần để nhóm sinh viên sớm hoàn thành đồ án này. Rất cảm ơn gia đình đã luôn bên cạnh và ủng hộ chúng em khi gặp phải các rào cản tinh thần khi thử nghiệm thất bại trong suốt quá trình thực hiện đề tài này.

Tuy đã cố gắng hết sức, luận văn này vẫn còn nhiều thiếu sót, nhóm sinh viên rất mong nhận được sự góp ý từ Quý Thầy/Cô để đề tài được hoàn thiện hơn.

Tp Hồ Chí Minh, ngày tháng 08 năm 2022

Sinh viên thực hiện

Trịnh Xuân Chinh

Lê Diệp Thuỳ Dương

TÓM TẮT

Với sự phát triển của công nghiệp, các nhóm ngành thuộc về kỹ thuật như Công nghệ thông tin, Cơ – Điện tử, Điện – Điện tử, Điều khiển tự động ... đã và đang đạt được nhiều thành tựu mới. Việc đơn giản hóa các mô hình điều khiển tự động thực tế rất cần thiết với mục đích tìm ra giải thuật điều khiển tối ưu nhất để hệ thống có thể vận hành mượt mà và thời gian ổn định nhanh nhất.

Trong môi trường nghiên cứu, các mô hình chuẩn đóng vai trò rất quan trọng, thay thế các mô hình thực tế vốn có phương trình toán phức tạp về các dạng mô hình đơn giản. Hệ Pendubot là một trong số các hệ thống mang tính phi tuyến phổ biến nhất trong phòng thí nghiệm, là mô hình lý tưởng để nghiên cứu về các giải thuật điều khiển.

Suốt thời gian học đại học, nhóm sinh viên đã được học về nhiều giải thuật điều khiển, từ phi tuyến, tuyến tính đến thông minh, tuy nhiên chỉ có cái nhìn tổng quát chứ chưa thật sự hiểu sự khác biệt giữa các cách thức điều khiển này. Vì lẽ đó, với đề tài tốt nghiệp, nhóm sinh viên muốn đi vào các giải thuật, để có thể hiểu hơn về các phương pháp điều khiển, cũng như về mô hình Pendubot. Đó là lý do nhóm sinh viên chọn đề tài “So sánh các phương pháp điều khiển cân bằng cho hệ Pendubot”.

Trong đề tài này, các giải thuật nhóm sinh viên chọn để khảo sát bao gồm LQR – Điều khiển tối ưu (giải thuật tuyến tính), Tuyến tính hoá hồi tiếp (giải thuật phi tuyến), Fuzzy – Điều khiển mờ (giải thuật thông minh). Nhóm sinh viên tiến hành mô phỏng trên công cụ MATLAB/Simulink kết hợp với các phần mềm hỗ trợ, đồng thời xây dựng mô hình thực tế để có thể quan sát trực quan hơn về cách hoạt động của mô hình.

Đề tài này chỉ khảo sát hệ Pendubot hoạt động trong giới hạn lân cận vị trí cân bằng của vị trí TOP – cả 2 thanh của hệ đứng thẳng lên theo hướng ngược chiều trọng lực. Bộ xử lý trung tâm sẽ điều khiển động cơ, đồng thời đưa tín hiệu về máy tính, kết hợp với module UART để thu thập dữ liệu. Các dữ liệu thu thập được sẽ phục vụ cho việc vẽ đồ thị.

Nhóm sinh viên xây dựng phần cứng dựa trên mô hình có sẵn. Toàn bộ phần điện bao gồm vi điều khiển, board mạch... đã tự làm lại dưới sự chứng kiến của TS. Nguyễn Văn Đông Hải. Nhóm sinh viên xin cam đoan, đề tài “So sánh các phương pháp điều khiển cân bằng cho hệ Pendubot” là thành phẩm của riêng nhóm. Các số liệu lấy làm cơ sở ban đầu, các kết quả đạt được là hoàn toàn trung thực, dựa trên các tài liệu đã được công bố.

MỤC LỤC

Trang phụ bìa	TRANG
Chương 1 TỔNG QUAN	1
1.1 ĐẶT VÂN ĐỀ	1
1.2 MỤC TIÊU.....	1
1.3 NỘI DUNG NGHIÊN CỨU	2
1.4 GIỚI HẠN.....	2
Chương 2 PHƯƠNG TRÌNH TOÁN HỌC	4
2.1 GIỚI THIỆU	4
2.1.1 Các mô hình SIMO trong phòng thí nghiệm.....	4
2.1.2 Các vị trí cân bằng của Pendubot	7
2.2 PHƯƠNG TRÌNH TOÁN HỌC HỆ PENDUBOT	7
Chương 3 GIẢI THUẬT ĐIỀU KHIỂN	14
3.1 GIẢI THUẬT ĐIỀU KHIỂN TỐI UU – LQR	14
3.1.1 Giới thiệu.....	14
3.1.2 Xây dựng bộ điều khiển LQR cho hệ Pendubot.....	16
3.2 GIẢI THUẬT TUYẾN TÍNH HOÁ HỒI TIẾP	16
3.2.1 Giới thiệu.....	16
3.2.2 Xây dựng Bộ điều khiển Tuyến tính hoá hồi tiếp cho hệ Pendubot	18
3.3 GIẢI THUẬT ĐIỀU KHIỂN MỒ - FUZZY	19
3.3.1 Giới thiệu.....	19
3.3.2 Xây dựng bộ điều khiển Fuzzy cho hệ Pendubot.....	20
3.4 GIẢI THUẬT DI TRUYỀN – GA	23
3.4.1 Giới thiệu.....	23
3.4.2 Áp dụng giải thuật GA tối ưu hoá các bộ điều khiển.....	24
Chương 4 MÔ PHỎNG	27
4.1 THÔNG SỐ MÔ PHỎNG	27

4.1.1	Thông số cho Bộ điều khiển LQR.....	27
4.1.2	Thông số cho Bộ điều khiển Tuyến tính hoá hồi tiếp	27
4.1.3	Thông số cho Bộ điều khiển Fuzzy	28
4.2	CHƯƠNG TRÌNH MÔ PHỎNG	28
4.2.1	Chương trình mô phỏng Bộ điều khiển LQR	28
4.2.2	Chương trình mô phỏng Bộ điều khiển Tuyến tính hoá hồi tiếp	29
4.2.3	Chương trình mô phỏng Bộ điều khiển Fuzzy	30
4.3	KẾT QUẢ MÔ PHỎNG	31
4.3.1	Kết quả mô phỏng Bộ điều khiển LQR.....	31
4.3.2	Kết quả mô phỏng Bộ điều khiển Tuyến tính hoá hồi tiếp	32
4.3.3	Kết quả mô phỏng Bộ điều khiển Fuzzy	34
4.4	NHẬN XÉT	35
	Chương 5 THỰC NGHIỆM	36
5.1	MÔ HÌNH	36
5.1.1	Thiết kế phần điện	36
5.1.2	Thiết kế phần cơ	42
5.2	Thư viện và Phần mềm	46
5.2.1	Wajung Blockset	46
5.2.2	Hyper Terminal	54
5.2.3	Thiết kế chương trình điều khiển	55
5.3	KẾT QUẢ THỰC NGHIỆM	60
5.4	NHẬN XÉT	69
	Chương 6 KẾT LUẬN - HƯỚNG PHÁT TRIỂN	70
6.1	KẾT LUẬN.....	70
6.2	HƯỚNG PHÁT TRIỂN	70
	TÀI LIỆU THAM KHẢO	72
	PHỤ LỤC	73

DANH MỤC CÁC TỪ VIẾT TẮT

GA – Genetic Algorithm: phương pháp tìm kiếm tối ưu ngẫu nhiên bằng cách mô phỏng quá trình tiến hoá của sinh vật theo học thuyết Darwin

LQR – Linear Quadratic regulator: Bộ điều khiển tối ưu tuyến tính hoá toàn phương, hay gọi tắt là Bộ điều khiển tối ưu

MIMO – Multiple-Input & Multiple-Output: hệ thống nhiều ngõ vào - nhiều ngõ ra

PID – Proportional Integral Derivative: Bộ điều khiển vi tích phân tỉ lệ

PWM – Pulse-width modulation: phương pháp điều chỉnh điện áp đầu ra dựa trên sự thay đổi độ rộng của xung

SIMO – Single-Input & Multiple-Output: hệ thống một ngõ vào – nhiều ngõ ra

SMC – Sliding Mode Control: Bộ điều khiển trượt

UART – Universal Asynchronous Receiver/Transmitter: vi mạch trong vi điều khiển, hỗ trợ truyền dữ liệu nối tiếp giữa 2 thiết bị

DANH SÁCH BẢNG BIỂU

Bảng 1.1. Bảng phân chia công việc	2
<hr/>	
Bảng 2.1. Thông số hệ thống	8
<hr/>	
Bảng 3.1. Các biến ngõ vào và ngõ ra của Bộ điều khiển Fuzzy	21
<hr/>	
Bảng 4.1. Biến trạng thái ban đầu	27
Bảng 4.2. Thông số của Bộ điều khiển Tuyến tính hoá hồi tiếp	28
Bảng 4.3. Thông số của Bộ điều khiển Fuzzy	28
<hr/>	
Bảng 5.1. Bảng logic trạng thái hoạt động động cơ	40
Bảng 5.2. Thông số kỹ thuật Cầu H IR2184	41

DANH SÁCH HÌNH ẢNH, BIỂU ĐỒ

Hình 2.1. Mô hình toán học hệ Xe con lắc ngược	4
Hình 2.2. Mô hình toán học hệ Con lắc ngược quay	5
Hình 2.3. Hệ Cân bằng quả bóng trên thanh đỡ	5
Hình 2.4. Mô hình toán học hệ Acrobot.....	6
Hình 2.5. Các vị trí cân bằng của Pendubot	7
Hình 2.6. Mô hình toán học hệ Pendubot.....	8
Hình 2.7. Cấu trúc của động cơ DC.....	10
<hr/>	
Hình 3.1. Sơ đồ nguyên lý hệ thống được điều khiển bởi Bộ điều khiển LQR.....	14
Hình 3.2. Sơ đồ nguyên lý của hệ thống Pendubot được điều khiển bởi Bộ điều khiển LQR	16
Hình 3.3. Sơ đồ nguyên lý hệ thống được điều khiển bởi Bộ điều khiển Fuzzy.....	20
Hình 3.4. Luật mờ ngõ vào x_1	22
Hình 3.5. Luật mờ ngõ vào x_2	22
Hình 3.6. Luật mờ ngõ vào x_3	22
Hình 3.7. Luật mờ ngõ vào x_4	23
Hình 3.8. Luật mờ ngõ ra u	23
<i>Hình 3.10. Lưu đồ chương trình GA.....</i>	25
Hình 3.11. Sơ đồ nguyên lý hệ thống được điều khiển bởi Bộ điều khiển GA kết hợp....	26
<hr/>	
Hình 4.1. Chương trình mô phỏng Bộ điều khiển LQR cho hệ Pendubot	29
Hình 4.2. Chương trình mô phỏng Bộ điều khiển Tuyến tính hoá hồi tiếp cho hệ Pendubot	29
Hình 4.3. Chương trình mô phỏng Bộ điều khiển Fuzzy cho hệ Pendubot.....	30
Hình 4.4. Chương trình của khối Fuzzy Controller.....	30
Hình 4.5. Kết quả mô phỏng LQR cho góc thanh 1	31
Hình 4.6. Kết quả mô phỏng LQR cho vận tốc thanh 1	31
Hình 4.7. Kết quả mô phỏng LQR cho góc thanh 2	31

Hình 4.8. Kết quả mô phỏng LQR cho vận tốc thanh 2	32
Hình 4.9. Kết quả mô phỏng Tuyến tính hoá hồi tiếp cho góc thanh 1	32
Hình 4.10. Kết quả mô phỏng Tuyến tính hoá hồi tiếp cho vận tốc thanh 1	33
Hình 4.11. Kết quả mô phỏng Tuyến tính hoá hồi tiếp cho góc thanh 2.....	33
Hình 4.12. Kết quả mô phỏng Tuyến tính hoá hồi tiếp cho vận tốc thanh 2.....	33
Hình 4.13. Kết quả mô phỏng Fuzzy cho góc thanh 1	34
Hình 4.14. Kết quả mô phỏng Fuzzy cho vận tốc thanh 1	34
Hình 4.15. Kết quả mô phỏng Fuzzy cho góc thanh 2	34
Hình 4.16. Kết quả mô phỏng Fuzzy cho vận tốc thanh 2	35
<hr/>	
Hình 5.1. Sơ đồ kết nối phần điện của hệ thống.....	36
Hình 5.2. Hình ảnh thực tế STM32F407VG	37
Hình 5.3. Hình ảnh thực tế động cơ NF5475E	38
Hình 5.4. Hình ảnh minh họa nguyên lý hoạt động của encoder	39
Hình 5.5. Sơ đồ cấu tạo cơ bản của mạch cầu H	40
Hình 5.6. Hình ảnh board Cầu H IR2184	41
Hình 5.7. Hình ảnh thực tế Module chuyển đổi CP2102	42
Hình 5.8. Hình chụp phần cứng khi đang ở trạng thái nghỉ	43
Hình 5.9. Hình chụp phần cứng khi đang cân bằng ở vị trí TOP	43
Hình 5.10. Hình chụp góc nghiên của hệ Pendubot	44
Hình 5.11. Hình chụp hộp điện của hệ Pendubot	45
Hình 5.13. Thư viện Waijung Blocket trong thư viện chính của MATLAB/SIMULINK	46
Hình 5.14. Khối Target Setup.....	47
Hình 5.15. Cài đặt cho khối Target Setup	48
Hình 5.16. Khối UART Setup	48
Hình 5.17. Cài đặt cho khối UART Setup	49
Hình 5.18. Khối UART Tx	49
Hình 5.19. Cài đặt cho khối UART Tx	50
Hình 5.20. Khối Encoder Read.....	50

Hình 5.21. Cài đặt cho khối Encoder Read	51
Hình 5.22. Khối Basic PWM.....	51
Hình 5.23. Cài đặt cho khối Basic PWM	52
Hình 5.24. Khối Digital Output	52
Hình 5.25. Cài đặt cho khối Digital Output.....	53
Hình 5.26. Giao diện chính của phần mềm Hyper Terminal.....	54
Hình 5.27. Chương trình điều khiển Pendubot bằng Bộ điều khiển LQR thực tế	55
Hình 5.28. Nội dung khối Subsystem “System setup”	56
Hình 5.29. Nội dung khối Subsystem “Pendubot”	56
Hình 5.30. Cài đặt cho khối Encoder 1	56
Hình 5.31. Cài đặt cho khối Encoder 2	57
Hình 5.32. Nội dung khối Subsystem “H Brigde Controlling”	57
Hình 5.33. Cài đặt đảo chiều động cơ	58
Hình 5.34. Cài đặt điều xung cho động cơ	58
Hình 5.35. Nội dung khối Subsystem “UART reader”	59
Hình 5.36. Cài đặt đọc UART	59
Hình 5.37. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR	60
Hình 5.38. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR	61
Hình 5.39. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã tăng K1.....	61
Hình 5.40. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã tăng K1	62
Hình 5.41. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã giảm K1	62
Hình 5.42. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã giảm K1	63
Hình 5.43. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã tăng K2	63
Hình 5.44. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã tăng K2	64
Hình 5.45. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã giảm K2	64
Hình 5.46. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã giảm K2	65
Hình 5.47. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã tăng K3	65
Hình 5.48. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã tăng K3	66
Hình 5.49. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã giảm K3	66

Hình 5.50. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã giảm K_3	67
Hình 5.51. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã tăng K_4	67
Hình 5.52. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã tăng K_4	68
Hình 5.53. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã giảm K_4	68
Hình 5.54. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã giảm K_4	69

Chương 1

TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Mô hình Pendubot, gọi đầy đủ là Pendulum Robot, là một hệ mô hình cổ điển nhưng đóng vai trò hết sức quan trọng trong việc nghiên cứu giải thuật trong các phòng thí nghiệm Điều khiển tự động. Đây là cơ sở để tạo ra các hệ thống tự cân bằng trong thực tế. Ví dụ như hệ thống cân bằng giàn khoan trên biển, hệ thống cân bằng khi phóng tàu vũ trụ và điều khiển đi đúng quỹ đạo... Khi lý thuyết áp dụng trên các mô hình chuẩn (hệ Pendubot, hệ con lắc ngược quay...) hoàn thiện, các nhà khoa học sẽ áp dụng các giải thuật tương tự cho các hệ thống thực tế.

Trong tài liệu [4], tác giả Phan Việt Hùng khảo sát 2 bộ điều khiển PID và LQR kết hợp giải thuật GA để tối ưu hóa các thông số điều khiển cho hệ Pendubot tại vị trí làm việc tĩnh. Tác giả Xiao Qing Ma ở tài liệu [8] đề cập đến phương pháp điều khiển hệ Pendubot bằng giải thuật Fuzzy. Tuy nhiên cả 2 tài liệu [4] và [8] đều chỉ tiến hành nghiên cứu trên mô phỏng, chưa có mô hình thực tế để kiểm tra tính đúng đắn của các phép toán đã chứng minh được.

Ở [7], tác giả Vũ Đình Đạt đã điều khiển rất tốt hệ Pendubot bằng giải thuật SMC, có khả năng swing-up và giữ cân bằng hệ thống thực tế khi cho thanh 1 di chuyển theo quỹ đạo hình sin. Tuy nhiên, tác giả chỉ đi sâu vào 1 giải thuật, chưa bao quát các phương pháp điều khiển cân bằng.

Nhìn thấy khả năng ứng dụng và tính mở rộng của mô hình, nhóm sinh viên tiến hành nghiên cứu và so sánh các phương pháp điều khiển cân bằng cho hệ Pendubot từ mô phỏng đến mô hình thực tế thông qua các lý thuyết điều khiển LQR – Điều khiển tối ưu (giải thuật tuyến tính), Tuyến tính hóa hồi tiếp (giải thuật phi tuyến), Fuzzy – Điều khiển mờ (giải thuật thông minh). Song song đó, nhóm sinh viên kết hợp giải thuật GA để tìm ra các hệ số tối ưu nhất cho mô hình.

1.2 MỤC TIÊU

Để hoàn thành đề tài này, nhóm sinh viên hoàn thành các nhiệm vụ sau:

- Xây dựng mô hình toán cho hệ Pendubot

CHƯƠNG 1. Tổng quan

- Thiết kế các bộ điều khiển Fuzzy, Tuyến tính hoá hồi tiếp và LQR
- Mô phỏng điều khiển hệ Pendubot bằng công cụ MATLAB Simulink
- Thiết kế mô hình hệ Pendubot thực tế
- Lập trình cho vi điều khiển

Mục tiêu là cân bằng thành công mô hình thực tế hệ Pendubot ở vị trí Top (cả 2 thanh đều dựng thẳng lên). Thông qua các lý thuyết đã chứng minh, nhóm sinh viên tiến hành so sánh phương thức điều khiển cho hệ Pendubot, từ đó đưa ra các kết luận.

1.3 NỘI DUNG NGHIÊN CỨU

Trong quyển báo cáo này, nhóm sinh viên chia thành 6 chương để có nghiên cứu và phân tích tỉ mỉ hơn. Các chương bao gồm:

- Chương 1. Tổng quan
- Chương 2. Phương trình toán học
- Chương 3. Giải thuật điều khiển
- Chương 4. Mô phỏng
- Chương 5. Thực nghiệm
- Chương 6. Kết luận và Hướng phát triển

1.4 GIỚI HẠN

Trong giới hạn Luận văn tốt nghiệp, nhóm sinh viên không thể khảo sát hết tất cả các phương thức điều khiển cân bằng cho hệ Pendubot, vì thế chỉ lấy điển hình 1 bộ điều khiển trong 3 phương pháp điều khiển phổ biến nhất (tuyến tính, phi tuyến và thông minh) để so sánh tổng quát.

Về các vấn đề phân chia công việc, nhóm sinh viên đã cùng nhau hoàn thành các nhiệm vụ đã liệt kê theo *Bảng 1.1*.

Bảng 1.1. Bảng phân chia công việc

STT	Nội dung công việc	Sinh viên thực hiện
1	Xây dựng phương trình toán học của hệ Pendubot	Lê Diệp Thùy Dương

CHƯƠNG 1. Tổng quan

2	Thiết kế và mô phỏng điều khiển hệ Pendubot với Bộ điều khiển Fuzzy	Lê Diệp Thùy Dương
3	Thiết kế và mô phỏng điều khiển hệ Pendubot với Bộ điều khiển LQR	Lê Diệp Thùy Dương
4	Thiết kế và mô phỏng điều khiển hệ Pendubot với Bộ điều khiển Tuyến tính hoá hồi tiếp	Trịnh Xuân Chinh
5	Thiết kế mô hình Pendubot thực tế	Lê Diệp Thùy Dương
6	Lập trình điều khiển hệ Pendubot trên vi xử lý	Trịnh Xuân Chinh
7	Viết báo cáo tốt nghiệp	Lê Diệp Thùy Dương Trịnh Xuân Chinh

Chương 2 PHƯƠNG TRÌNH TOÁN HỌC

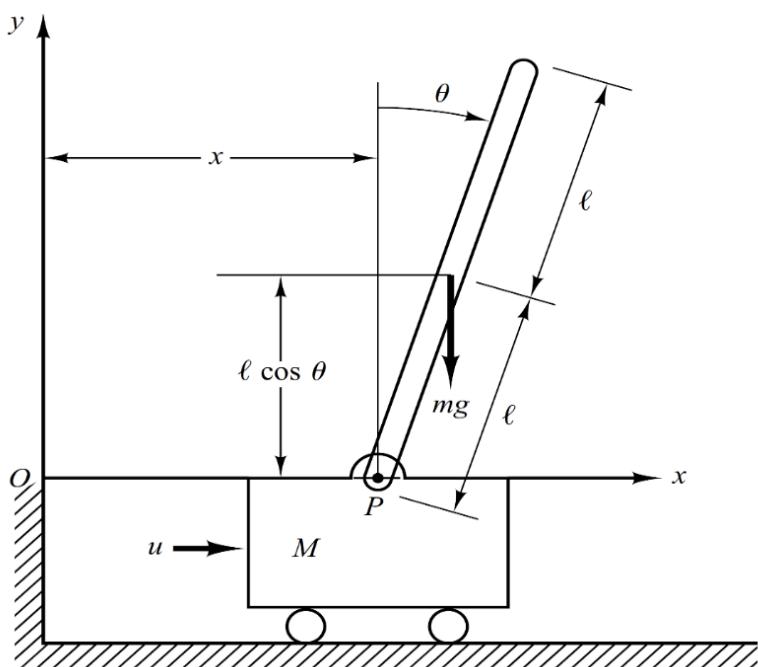
2.1 GIỚI THIỆU

Mô hình chuẩn là dạng đơn giản hóa của hệ thống cân bằng thực tế. Mô hình chuẩn mang phương trình toán học đơn giản hơn, được điều khiển bởi các thuật toán cân bằng. Sau khi đã xây dựng bộ điều khiển, nhà nghiên cứu sẽ áp dụng bộ điều khiển đó vào các ứng dụng thực tế, đồng thời có các thay đổi, cũng như chỉnh sửa các thông số sao cho phù hợp hơn với từng hệ thống cụ thể. Có thể nói, các mô hình ở phòng thí nghiệm là tiền đề cho các ứng dụng thực tế, giúp giảm thiểu rủi ro và tiết kiệm hơn trong quá trình nghiên cứu.

2.1.1 Các mô hình SIMO trong phòng thí nghiệm

Đối với các mô hình dạng SIMO, tín hiệu đầu vào thường là điện áp cấp cho động cơ hoặc momen do động cơ tạo ra, tín hiệu đầu ra là các góc và vận tốc cần điều khiển. Các mô hình SIMO điển hình là hệ Xe con lắc ngược, hệ Con lắc ngược quay, hệ Cân bằng quả bóng trên thanh đỡ, hệ Acrobot, hệ Pendubot...

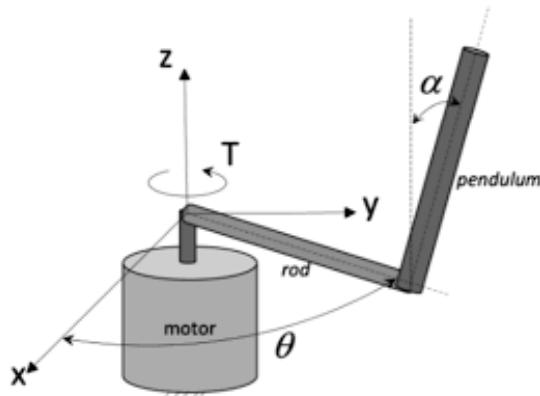
Hệ Xe con lắc ngược – Inverted Pendulum on a Cart



Hình 2.1. Mô hình toán học hệ Xe con lắc ngược

Mô hình toán học hệ Xe con lắc ngược được trình bày theo *Hình 2.1*. Hệ gồm thanh con lắc với một đầu gắn cố định vào xe. Động cơ điều khiển xe di chuyển trên một phương trong giới hạn cụ thể. Hệ thống điều khiển thanh con lắc đứng thẳng lên, đồng thời xe sẽ di chuyển về vị trí 0 đã định sẵn.

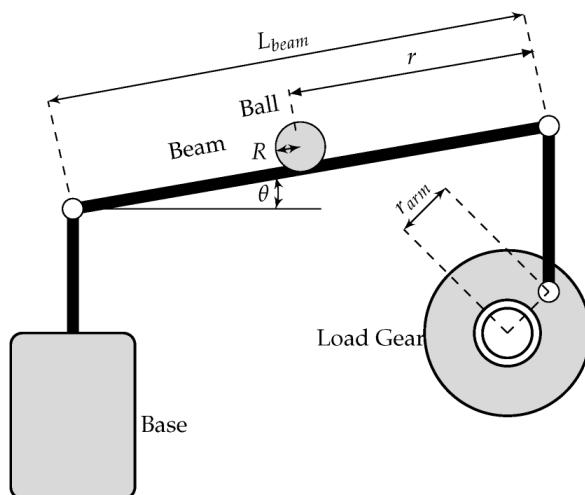
Hệ Con lắc ngược quay – Rotary Inverted Pendulum



Hình 2.2. Mô hình toán học hệ Con lắc ngược quay

Mô hình của hệ Con lắc ngược quay được minh họa như *Hình 2.2*. Hệ này có cấu trúc đơn giản, gồm một thanh cánh tay gắn vuông góc với trực động cơ, đầu còn lại của thanh cánh tay được gắn với thanh con lắc. Thanh con lắc có khả năng quay tự do trong mặt phẳng vuông góc với thanh cánh tay. Đồng thời, cả 2 thanh đều được trang bị encoder để đo chính xác góc quay. Điều khiển hệ Con lắc ngược quay là ổn định thanh con lắc ở vị trí thẳng đứng và thanh cánh tay về được đúng vị trí đặt trước.

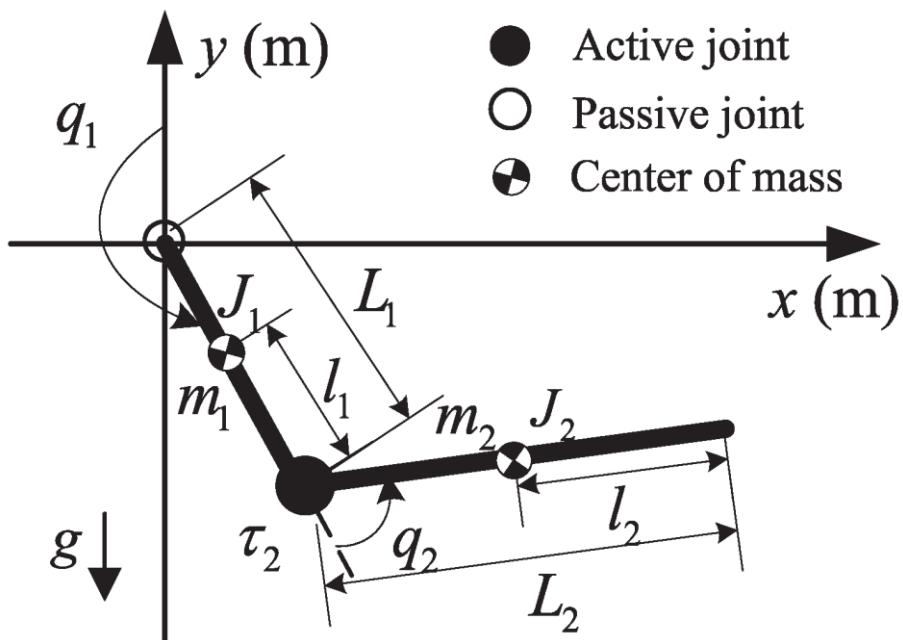
Hệ Cân bằng quả bóng trên thanh đỡ – Ball and Beam



Hình 2.3. Hệ Cân bằng quả bóng trên thanh đỡ

Hệ Cân bằng quả bóng trên thanh đỡ (*Hình 2.3*) là hệ thống dùng để thực hiện bài toán ẩn định vị trí. Theo như tên gọi, mô hình bao gồm hai thành phần chính, một thanh ngang, gọi là beam, thường có độ dài dao động từ 0.5 đến 1 m và quả bóng – ball. Ngoài ra, cơ cấu động học của hệ thống còn có động cơ điều khiển độ nghiêng của thanh ngang, cảm biến đọc vị trí quả bóng và cảm biến xác định độ nghiêng của thanh ngang.

Hệ Acrobot



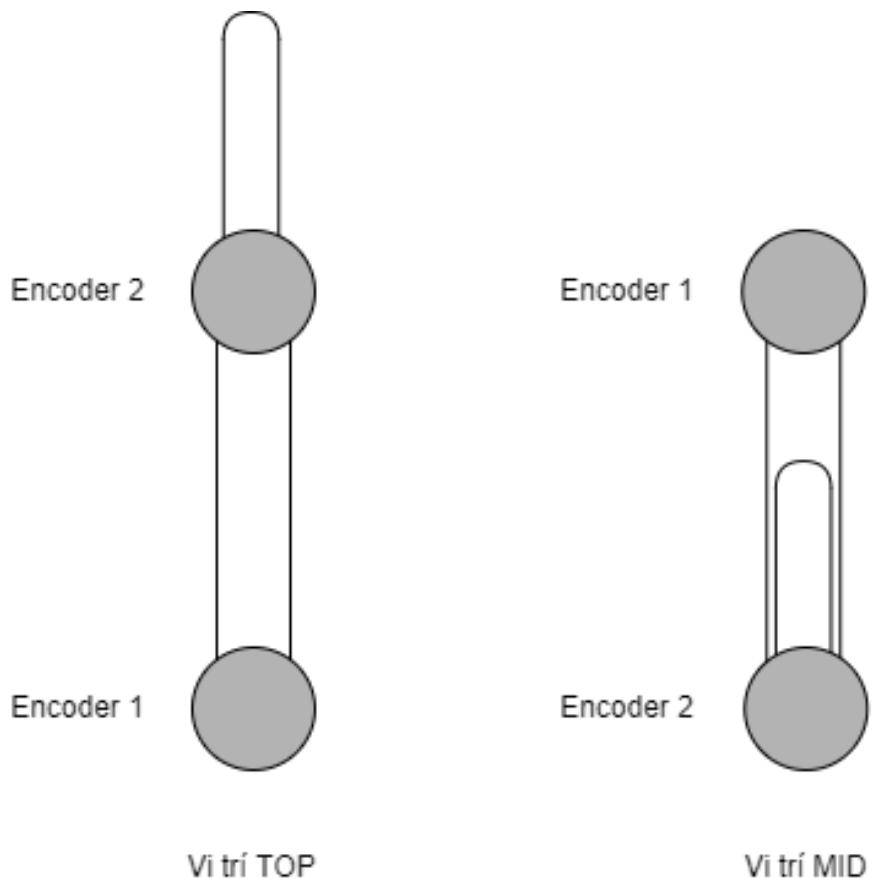
Hình 2.4. Mô hình toán học hệ Acrobot

Hệ Acrobot là cánh tay robot 2 thanh hoạt động trong mặt phẳng đứng với cơ cấu truyền động ở khớp nối thanh 1 và thanh 2. Tên của hệ này lấy cảm hứng từ vận động viên nhào lộn trên dây (tiếng Anh: Acrobat). Vận động viên nhào bằng và điều khiển chuyển động qua dây đeo ở thắt lưng. Đó cũng là nguyên lý hoạt động của acrobot, sử dụng momen xoắn ở khuỷu tay (khớp thứ 2) để di chuyển và cân bằng hệ thống.

Hệ Pendubot

Cũng như hệ Acrobot, hệ Pendubot cũng là cánh tay máy 2 thanh liên kết với nhau bởi các khớp nối, nhưng thay vì khớp truyền động ở thanh 2, Pendubot sử dụng cơ cấu truyền động đặt ở gốc, tức là khớp ở đầu thanh 1. Mô hình và các phương trình toán học của Pendubot sẽ được trình bày ở phần 2.2.

2.1.2 Các vị trí cân bằng của Pendubot



Hình 2.5. Các vị trí cân bằng của Pendubot

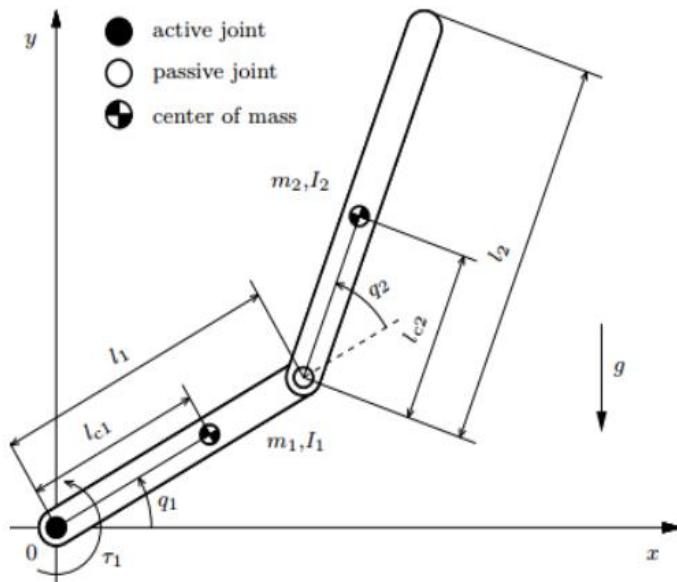
Pendubot có thể cân bằng ở 2 vị trí, minh họa ở *Hình 2.5*

- Vị trí TOP: thanh 1 và thanh 2 đều ở vị trí hướng lên
- Vị trí MID: thanh 1 hướng xuống theo chiều trọng trường, thanh 2 hướng lên

2.2 PHƯƠNG TRÌNH TOÁN HỌC HỆ PENDUBOT

Thiết kế bộ điều khiển LQR phải dựa trên các phân tích và mô tả theo phương trình toán học của hệ thống. Dựa trên phương trình toán, thông số hệ thống và phương trình Riccati, ta tính toán được ma trận K. Ma trận K vừa tìm được là tín hiệu điều khiển hồi tiếp cho bộ điều khiển.

Sơ đồ cấu trúc Pendubot được thể hiện như *Hình 2.6*. Hệ bao gồm 2 thanh được liên kết với nhau bởi passive joint (khớp bị động), đây là vị trí đặt encoder, đọc giá trị góc giữa 2 thanh. Đầu còn lại của thanh 1 gọi là active joint (khớp chủ động), là vị trí đặt đầu trực của động cơ DC có chức năng truyền động cho cả hệ thống.



Hình 2.6. Mô hình toán học hệ Pendubot

Các biến gồm có q_1 , q_2 và τ_1 . Trong đó, tín hiệu đầu vào là momen động cơ $\tau_1 = [\tau_m \ 0]$, tín hiệu đầu ra là góc thanh 1 q_1 so với phương Ox và góc thanh 2 q_2 so với thanh 1. Các biến này không có giá trị xác định, sẽ thay đổi qua từng vòng điều khiển tiếp.

Các thông số hệ thống được trình bày ở *Bảng 2.1*. Trong đó, khối lượng của các thanh đã bao gồm khối lượng encoder.

Các thông số phần cứng có được qua quá trình nhận dạng động cơ, tuy nhiên nhóm sinh viên vẫn chưa tự làm phần này nên đã dùng lại bộ thông số của tài liệu [6]

Bảng 2.1. Thông số hệ thống

Kí hiệu	Giá trị	Đơn vị	Mô tả
m_1	0.16	kg	Khối lượng thanh 1
l_1	0.2	m	Chiều dài thanh 1
l_{c1}	0.1	m	Khoảng cách từ gốc hệ thống đến trọng tâm thanh 1
I_1	0.00222	$kg \cdot m^2$	Momen quán tính thanh 1
m_2	0.066	kg	Khối lượng thanh 2

CHƯƠNG 2. Phương trình toán học

l_2	0.22	m	Chiều dài thanh 2
l_{c2}	0.11	m	Khoảng cách từ passive joint đến trọng tâm thanh 2
I_2	0.00106	$kg \cdot m^2$	Momen quán tính thanh 2
K_t	0.0198	Nm	Hằng số momen động cơ
K_b	0.0198	V	Hằng số phản điện của động cơ
R_m	6.835	Ω	Điện trở phần ứng
J_m	0.000134	$kg \cdot m^2$	Momen quán tính của động cơ
C_m	0.000048	Nm	Hệ số ma sát nhót

Phương trình Euler – Lagrange dùng để tính toán phương trình động học của hệ Pendubot:

$$\frac{d}{dt} \frac{\partial}{\partial \dot{q}_k} L(q, \dot{q}) - \frac{\partial}{\partial q_k} L(q, \dot{q}) = \tau \quad (1)$$

Trong đó hàm Lagrange được định nghĩa là hiệu của động năng K và thế năng P của hệ Pendubot.

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) \quad (2)$$

Từ (1), tìm được phương trình chuyển động như sau:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = \tau \quad (3)$$

Với:

$$M(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos q_2 & \beta_2 + \beta_3 \cos q_2 \\ \beta_2 + \beta_3 \cos q_2 & \beta_2 \end{bmatrix} \quad (4)$$

$$V(q, \dot{q}) = \begin{bmatrix} -\beta_3 \sin q_2 \dot{q}_2 & -\beta_3 \sin q_2 \dot{q}_2 - \beta_3 \sin q_2 \dot{q}_1 \\ \beta_3 \sin q_2 \dot{q}_1 & 0 \end{bmatrix} \quad (5)$$

$$G(q) = \begin{bmatrix} \beta_4 \cos q_1 + \beta_5 \cos(q_1 + q_2) \\ \beta_5 \cos(q_1 + q_2) \end{bmatrix} \quad (6)$$

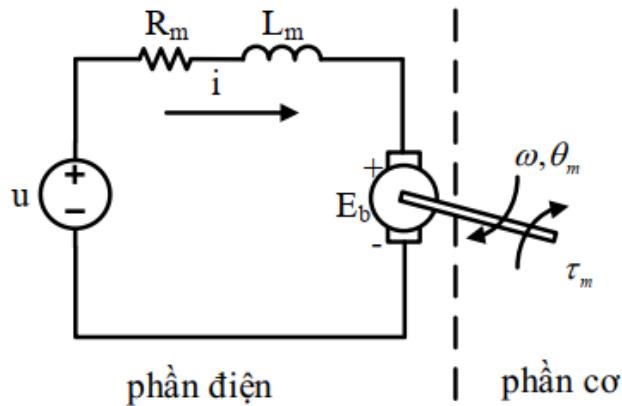
Trong đó:

$$\beta_1 = m_1 l_1^2 + m_2 l_1^2 \quad \beta_4 = (m_1 + m_2)l_1 g$$

$$\beta_2 = m_2 l_2^2 \quad \beta_5 = m_2 l_2 g$$

$$\beta_3 = m_2 l_1 l_2$$

Theo lý thuyết, tín hiệu ngõ vào của hệ thống là momen từ động cơ tác động vào thanh 1. Nhưng trong thực tế, điện áp cung cấp cho động cơ quay để sinh ra momen. Vì thế, ngõ vào thực tế của hệ Pendubot là điện áp u . Phải chuyển đổi ngõ vào momen τ_m về điện áp u để điều khiển chính xác ngõ vào thực tế.



Hình 2.7. Cấu trúc của động cơ DC

Theo *Hình 2.7*, cấu trúc của động cơ được chia làm 2 phần: phần điện và phần cơ. Từ đó sẽ thu được 2 phương trình (7) và (8) với E_b là suất phản điện động, T là momen điện từ, τ_m là momen tải, $\omega = q_1$ do trực động cơ được gắn trực tiếp với thanh 1.

$$u = iR_m + L_m \frac{di}{dt} + E_b \quad (7)$$

$$J \frac{d\omega}{dt} = T - C_m \omega - \tau_m \quad (8)$$

$$\text{Trong đó: } E_b = K_b \omega \quad T = K_i i \quad (9)$$

Do: $L_m \frac{di}{dt} \ll u \rightarrow L_m \frac{di}{dt} \approx 0$

$$(7) \rightarrow i = \frac{u - K_b \omega}{R_m} \quad (10)$$

Thay (9)(10) vào (8), ta được phương trình mới có dạng như sau:

$$\tau_m = a_1 u - a_2 \dot{q}_1 - a_3 \ddot{q}_1 \quad (11)$$

Với:

$$a_1 = \frac{K_t}{R_m} \quad a_2 = \frac{K_t K_b}{R_m} \quad a_3 = J_m$$

Giải phương trình (3), thu được hệ phương trình Pendubot có dạng như sau:

$$\ddot{q}_1 = \frac{-1}{2\beta_3^2 \cos^2 q_2 - 2\beta_1\beta_2 + 4\beta_2\beta_3 \cos q_2 - 4\beta_2\beta_3 \cos q_2 + q_2^2 + 2\dot{q}_1\dot{q}_2) + \beta_3^2 \sin(2q_2) \dot{q}_1^2 - 2\beta_2\beta_4 g \cos q_1 + \beta_3\beta_5 (\cos q_1 + \cos(q_1 + 2q_2)) - 2\beta_2\tau_1]} \quad (12)$$

$$\ddot{q}_2 = \frac{1}{\beta_3^2 \cos^2 q_2 - \beta_1\beta_2 + 2\beta_2\beta_3 \cos q_2 - 2\beta_2\beta_3 \cos q_2 - \beta_5 \cos(q_1 + q_2) - \beta_4 g \cos q_1 + \beta_3 \sin q_2 (\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2) + (\beta_3 \sin q_2 \dot{q}_1^2 + \beta_5 \cos(q_1 + q_2))(\beta_1 + \beta_2 + 2\beta_3 \cos q_2)} \quad (13)$$

Trong đề tài này, nhóm sinh viên tập trung khảo sát hệ Pendubot ở vị trí TOP. Các biến trạng thái sẽ được định nghĩa lại như sau:

$$\begin{aligned} x_1 &= \frac{\pi}{2} - q_1 & x_2 &= \dot{x}_1 = -\dot{q}_1 \\ x_3 &= \frac{\pi}{2} - (q_1 + q_2) & x_4 &= \dot{x}_3 = -\dot{q}_1 - \dot{q}_2 \end{aligned} \quad (14)$$

Việc đổi biến sẽ thay thế x cho q , tuyến tính hoá hệ thống xung quanh vị trí cân bằng $x_{CB} = [x_1 \ x_2 \ x_3 \ x_4] = [0 \ 0 \ 0 \ 0]$ để áp dụng giải thuật LQR. x_1 và x_3

CHƯƠNG 2. Phương trình toán học

tương ứng với góc thanh 1 và thanh 2 so với phương Oy. x_2 và x_4 lần lượt là vận tốc góc của thanh 1 và thanh 2.

Thay (11)(14) vào (12)(13), thu được hệ phương trình:

$$\dot{x}_1 = x_2 \quad (15)$$

$$\dot{x}_2 = -\ddot{q}_1 = f_1(x, u) \quad (16)$$

$$\dot{x}_3 = x_4 \quad (17)$$

$$\dot{x}_4 = -\ddot{q}_1 - \ddot{q}_2 = f_2(x, u) \quad (18)$$

Trong đó:

$$f_1(x, u) = \frac{1}{\beta_2 a_3 - \beta_3^2 (\cos(x_3 - x_1))^2 + \beta_1 \beta_2} [\beta_2 \beta_3 x_4^2 \sin(x_3 - x_1) + \beta_3^2 x_2^2 \sin(x_3 - x_1) \cos(x_3 - x_1) + \beta_2 \beta_4 \sin x_1 - \beta_3 \beta_5 \cos(x_3 - x_1) \sin x_3 - \beta_2 a_2 x_2 - a_1 \beta_2 u] \quad (19)$$

$$f_2(x, u) = \frac{1}{\beta_2 a_3 - \beta_3^2 (\cos(x_1 - x_3))^2 + \beta_1 \beta_2} [-\beta_2 \beta_3 x_4^2 \sin(x_3 - x_1) - \beta_3^2 x_4^2 \sin(x_3 - x_1) - \beta_1 \beta_3 x_2^2 \sin(x_3 - x_1) - \beta_3^2 x_2^2 \sin(x_3 - x_1) \cos(x_3 - x_1) - a_3 \beta_5 \cos x_3 - \beta_1 \beta_5 \cos x_3 - \beta_3 \beta_5 \cos x_3 \cos(x_3 - x_1) - \beta_2 \beta_4 \sin x_1 - \beta_3 \beta_4 \sin x_1 \cos(x_3 - x_1) + a_2 \beta_2 x_2 + a_2 \beta_3 x_2 \cos(x_3 - x_1) - a_3 \beta_3 x_2^2 \sin(x_3 - x_1) + \beta_2 u + \beta_3 \cos(x_3 - x_1) u] \quad (20)$$

Hệ thống được mô tả bằng phương trình biến trạng thái có dạng như sau:

$$\dot{x} = Ax + Bu \quad (21)$$

Trong đó:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial \partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \end{bmatrix}_{\substack{x=0 \\ u=0}} \quad (22)$$

$$B = \begin{bmatrix} 0 \\ \frac{\partial f_1}{\partial u} \\ 0 \\ \frac{\partial f_2}{\partial u} \end{bmatrix}_{\substack{x=0 \\ u=0}} \quad (23)$$

Từ các số liệu cung cấp ở Bảng 2.1, thay vào (22)(23), tính toán được ma trận A, B có kết quả như ở (24)(25).

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 20.2972 & -0.0048 & -5.9275 & 0 \\ 0 & 0 & 0 & 1 \\ -18.4520 & 0.0044 & 49.9795 & 0 \end{bmatrix} \quad (24)$$

$$B = \begin{bmatrix} 0 \\ -0.1326 \\ 0 \\ 0.1205 \end{bmatrix} \quad (25)$$

Xét tính điều khiển được của hệ thống:

Với ma trận tuyến tính A, B, tìm được ma trận điều khiển theo công thức (26) và đưa ra được kết quả trình bày ở (27).

$$M = [B \ AB \ A^2B \ A^3B] \quad (26)$$

$$M = \begin{bmatrix} 0 & -0.1326 & 0.0006 & -3.4059 \\ -0.1326 & 0.0006 & -3.4059 & 0.0329 \\ 0 & 0.1205 & -0.0006 & 8.4714 \\ 0.1205 & -0.0006 & 8.4714 & -0.0558 \end{bmatrix} \quad (27)$$

Ma trận điều khiển M có $rank = 4$, kết luận hệ Pendubot có thể điều khiển được ở vị trí TOP.

Chương 3

GIẢI THUẬT ĐIỀU KHIỂN

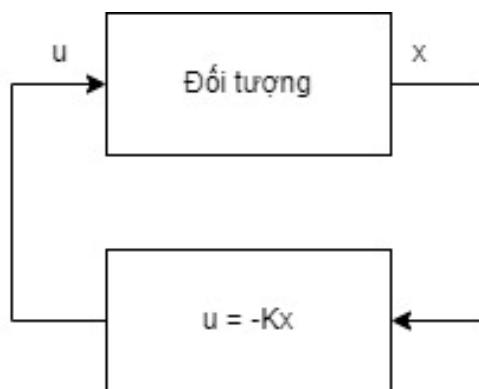
3.1 GIẢI THUẬT ĐIỀU KHIỂN TỐI UỐU – LQR

3.1.1 Giới thiệu

Bộ điều khiển LQR (Linear Quadratic Regulator – Điều khiển tối ưu) được xây dựng dựa trên cơ sở nguyên lý phản hồi trạng thái. LQR thường được dùng để điều khiển cho hệ SIMO. Tổng quát, bộ điều khiển nhận ngõ vào là trạng thái của hệ thống, tiến hành tính toán và chuyển thành tín hiệu điều khiển cho đối tượng.

Tín hiệu điều khiển hệ thống theo (28), với x là ngõ vào của Bộ điều khiển, là ma trận các giá trị biến trạng thái, K là ma trận hồi tiếp, u là ngõ ra của Bộ điều khiển, tức là tín hiệu điều khiển của hệ thống – điện áp hoặc momen. Sơ đồ nguyên lý hệ thống được điều khiển bởi Bộ điều khiển LQR được minh họa ở *Hình 3.1*.

$$u(t) = -Kx(t) \quad (28)$$



Hình 3.1. Sơ đồ nguyên lý hệ thống được điều khiển bởi Bộ điều khiển LQR

Xét hệ thống có tín hiệu điều khiển ($u \neq 0$):

$$\dot{x} = Ax + Bu \quad (29)$$

Ma trận K được xác định qua phương trình Riccati có dạng:

$$K = R^{-1}B^T S \quad (30)$$

Trong đó:

CHƯƠNG 3. Giải thuật điều khiển

- Q là ma trận trọng số với các hệ số dương có dạng như (31). Từng hệ số trong ma trận Q đại diện cho giá trị điều khiển cho một giá trị tương ứng, ví dụ tinh chỉnh Q_1 để điều khiển x_1 .

$$Q = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_n \end{bmatrix} \quad (31)$$

- R ($R > 0$) là hệ số năng lượng, R càng cao thì hệ thống đáp ứng càng nhanh, bù lại động cơ rung mạnh hơn, gai trong quá trình điều khiển PWM cũng xuất hiện nhiều hơn.
- S là nghiệm của phương trình đại số Riccati. Ma trận S tìm được bằng cách giải phương trình Riccati (32)

$$A^T S + S A - S B R^{-1} B^T S + Q = 0 \quad (32)$$

Từ (30), phương trình (28) được viết lại như sau:

$$u(t) = -R^{-1}B^T S x(t) \quad (33)$$

Xét tại điểm làm việc của hệ thống đã được tuyến tính hóa (các giá trị góc, vận tốc góc đã được quy đổi về 0), dựa vào các biểu thức toán học và mô phỏng thử nghiệm, ta sẽ tìm ra được A và B . Sau khi có được ma trận trạng thái, tiến hành chọn ma trận Q và hệ số R sao cho hàm mục tiêu là nhỏ nhất. Từ các giá trị đã tìm được, xác định được ma trận tối ưu $K = [K_1 \ K_2 \ K_3 \ K_4]$ qua việc giải phương trình Riccati.

Với công cụ MATLAB/Simulink, việc tính toán hệ số K cho Bộ điều khiển LQR không còn phức tạp nữa. Dùng lại ở bước tính A , B và chọn các hệ số cho ma trận Q , R , chỉ cần 1 dòng lệnh, hệ số K sẽ được chương trình tính toán theo các phép tính đã được lập trình sẵn, sau đó lưu vào bộ nhớ Workspace.

- Đối với hệ liên tục, câu lệnh tính toán K sẽ là:

$$K = lqr(A, B, Q, R) \quad (34)$$

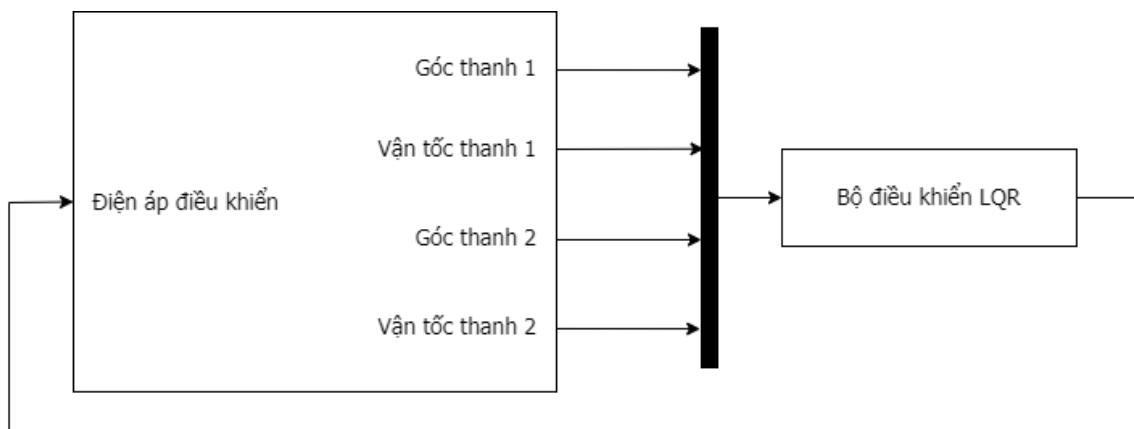
- Đối với hệ rời rạc, cần phải chuyển đổi từ liên tục sang rời rạc cho các giá trị ma trận trạng thái bằng câu lệnh (35), sau đó tiến hành tính K cho hệ rời rạc bằng câu lệnh $dlqr$ (36)

$$[Ad, Bd] = c2d(A, B, 0.01) \quad (35)$$

$$K = dlqr(Ad, Bd, Q, R) \quad (36)$$

3.1.2 Xây dựng bộ điều khiển LQR cho hệ Pendubot

Với cấu trúc đơn giản nhưng đảm bảo tính ổn định cao, bộ điều khiển LQR thường là ưu tiên hàng đầu cho các hệ thống điều khiển cân bằng như Pendubot. Ngõ vào bộ điều khiển là các biến góc và vận tốc thanh 1, góc và vận tốc thanh 2. Ngõ ra là giá trị điện áp.



Hình 3.2. Sơ đồ nguyên lý của hệ thống Pendubot được điều khiển bởi Bộ điều khiển LQR

3.2 GIẢI THUẬT TUYẾN TÍNH HOÁ HỒI TIẾP

3.2.1 Giới thiệu

Điều khiển Tuyến tính hoá hồi tiếp là một cách tiếp cận để thiết kế bộ điều khiển cho hệ phi tuyến. Trong đó, tín hiệu điều khiển xuất hiện tuyến tính trong phương trình trạng thái. Kỹ thuật hồi tiếp tuyến tính hoá biến đổi hệ phi tuyến thành hệ tuyến tính bằng cách hồi tiếp trạng thái và biến đổi hệ toạ độ. Phần này được tham khảo ở tài liệu [1] trang 307-312

Xét hệ thống phi tuyến SISO được mô tả bởi phương trình trạng thái có dạng:

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (37)$$

Trong đó:

- $x = [x_1, x_2 \dots x_n]^T \in R^n$ là vecto trạng thái hệ thống

CHƯƠNG 3. Giải thuật điều khiển

- $f(x), g(x) \in R^n$ là vecto mô tả động học của hệ thống
- $u \in R^n$ là tín hiệu ngõ vào
- $y \in R^n$ là tín hiệu ngõ ra
- $h(x) \in R^n$ là một hàm vô hướng của vecto x . Xác định quan hệ giữa biến trạng thái và tín hiệu ngõ ra
- n là bậc tương đối của hệ thống

Đạo hàm Lie của hàm $h(x)$ theo phương $f(x)$ được định nghĩa như sau:

$$L_f h(x) = \frac{\partial h(x)}{\partial x} f(x) = \left[\frac{\partial h(x)}{\partial x_1} \dots \frac{\partial h(x)}{\partial x_n} \right] [f_1(x) \dots f_n(x)]^T \quad (38)$$

Đạo hàm Lie của hàm $h(x)$ theo phương $f(x)$ k lần được định nghĩa như (39), với $L_f^0 h(x) = h(x)$

$$L_f^k h(x) = \frac{\partial L_f^{k-1} h(x)}{\partial x} f(x) \quad (39)$$

Đạo hàm Lie của hàm $h(x)$ theo phương $f(x)$ k lần, sau đó lấy đạo hàm Lie lần nữa theo phương $g(x)$ được định nghĩa như sau:

$$L_g L_f^k h(x) = \frac{\partial L_f^k h(x)}{\partial x} g(x) \quad (40)$$

Theo **quy tắc điều khiển tuyến tính hóa hồi tiếp**, ngõ ra của hệ thống được xác định theo công thức tổng quát (41)(42)

$$\frac{d^{n-i}y}{dt^{n-i}} = L_f^{n-i} h(x) + L_g L_f^{n-i-1} h(x) u = L_f^{n-i} h(x) \quad (i \in (1, n-1)) \quad (41)$$

$$\frac{d^n y}{dt^n} = L_f^n h(x) + L_g L_f^{n-1} h(x) u \quad (L_g L_f^{n-1} h(x) \neq 0) \quad (42)$$

Đặt: $\alpha(x) = L_f^n h(x)$

$$\beta(x) = L_g L_f^{n-1} h(x)$$

Biểu thức (42) được viết lại như sau:

$$y^{(n)} = \alpha(x) + \beta(x)u \quad (43)$$

Từ (43), tín hiệu điều khiển tuyến tính hoá hệ thống được xác định như ở (44), trong đó $v(t)$ là vecto tín hiệu ngõ vào của hệ thống.

$$u(x) = \frac{v(t) - \alpha(x)}{\beta(x)} \quad (44)$$

3.2.2 Xây dựng Bộ điều khiển Tuyến tính hoá hồi tiếp cho hệ Pendubot

Đối với hệ Pendubot, nhóm sinh viên chọn quan tâm cho biến góc lệch của các thanh. Mặt khác, Bộ điều khiển Tuyến tính hoá hồi tiếp chỉ áp dụng cho hệ SISO, tức là hệ thống có số ngõ vào bằng số ngõ ra, tuy nhiên hệ Pendubot là hệ SIMO. Vì thế, cần chọn ngõ ra $y = h(x)$ là tổ hợp các biến ngõ ra.

Biểu thức (44) tính tín hiệu điều khiển hệ Pendubot – momen. Hệ có 4 biến cần điều khiển, tức là hệ bậc 4, vậy $n = 4$.

Quan hệ tuyến tính được tạo ra giữa biến ngõ ra y và biến ngõ vào mới v . v có phương trình như sau:

$$v = -Kz \quad (45)$$

Trong đó:

- $K = [K_1 \ K_2 \ K_3 \ K_4]$ là ma trận với các hệ số dương
- $z = \begin{bmatrix} h(x) \\ L_f^1 h(x) \\ L_f^2 h(x) \\ L_f^3 h(x) \end{bmatrix}$ là biến trạng thái

Các giá trị đạo hàm Lie được tính toán như (46) → (49)

$$L_f^2 h(x) = \left[\frac{\partial L_f h(x)}{\partial x_1} \ \ \frac{\partial L_f h(x)}{\partial x_2} \ \ \frac{\partial L_f h(x)}{\partial x_3} \ \ \frac{\partial L_f h(x)}{\partial x_4} \right] * f \quad (46)$$

$$L_f^3 h(x) = \left[\frac{\partial L_f^2 h(x)}{\partial x_1} \ \ \frac{\partial L_f^2 h(x)}{\partial x_2} \ \ \frac{\partial L_f^2 h(x)}{\partial x_3} \ \ \frac{\partial L_f^2 h(x)}{\partial x_4} \right] * f \quad (47)$$

$$L_f^4 h(x) = \begin{bmatrix} \frac{\partial L_f^3 h(x)}{\partial x_1} & \frac{\partial L_f^3 h(x)}{\partial x_2} & \frac{\partial L_f^3 h(x)}{\partial x_3} & \frac{\partial L_f^3 h(x)}{\partial x_4} \end{bmatrix} * f \quad (48)$$

$$L_g L_f^3 h(x) = \begin{bmatrix} \frac{\partial L_f^3 h(x)}{\partial x_1} & \frac{\partial L_f^3 h(x)}{\partial x_2} & \frac{\partial L_f^3 h(x)}{\partial x_3} & \frac{\partial L_f^3 h(x)}{\partial x_4} \end{bmatrix} * g \quad (49)$$

3.3 GIẢI THUẬT ĐIỀU KHIỂN MỜ - FUZZY

3.3.1 Giới thiệu

Khái niệm

Lý thuyết logic mờ được đề cập đầu tiên bởi giáo sư Lotfi Zadeh của Trường Đại học California - Mỹ vào năm 1965. Lý thuyết này giải quyết các bài toán rất gần với cách tư duy của con người. Tới nay, lý thuyết logic mờ đã phát triển rất mạnh mẽ và được ứng dụng trong nhiều lĩnh vực của cuộc sống.

Theo logic truyền thống, một biểu thức chỉ nhận một trong hai giá trị: True (đúng) tương đương với mức logic 1 và False (sai) tương đương mức logic 0. Khác với logic truyền thống, logic mờ đưa ra kết quả của biểu thức ở một trong vô số giá trị số thực nằm trong khoảng giữa 0 và 1. Nói cách khác, logic mờ cho ta một giá trị lưỡng chừng, tùy theo mức độ nhiều hay ít của các điều kiện đầu vào.

Giá trị của các biến không phải là các con số cụ thể mà chỉ là các khái niệm đơn giản, “nóng”, “lạnh” hay “nhanh” “chậm”... Do đó cách tính toán của logic mờ rất gần với cách tư duy của con người.

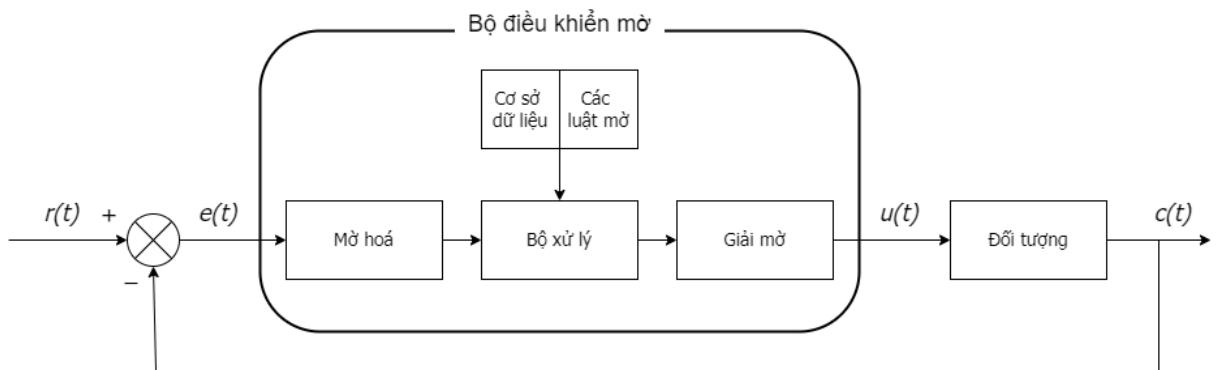
Nguyên lý hoạt động

Cấu tạo của một Hệ thống được điều khiển mờ bao gồm các khối đầu vào/đầu ra và một loạt các quy tắc xử lý giữa chúng. Từ một giá trị cụ thể đầu vào, bộ điều khiển trung tâm sẽ tiến hành “mờ hoá”, tức là đánh giá xem tín hiệu đó đang thuộc mức độ nào trong các biến đã định trước. Sau khi “mờ hoá”, giá trị đầu và sẽ mang các khái niệm chứ không còn là các con số cụ thể. Ở bước tiếp theo, các giá trị khái niệm này được xử lý theo các quy tắc của bộ điều khiển đã được thiết đặt sẵn và cho ra kết quả là một miền giá trị được tạo bởi các biến đầu ra. Từ các giá trị này, khâu giải mờ sẽ biến đổi các khái niệm thành một giá trị cụ thể đầu ra.

Theo *Hình 3.3*, một bộ điều khiển mờ hoàn chỉnh bao gồm:

CHƯƠNG 3. Giải thuật điều khiển

- Khâu mờ hoá và các khâu tiền xử lý để hiệu chỉnh các giá trị đầu vào như tỉ lệ, tích phân, vi phân...
- Bộ xử lý bao gồm các dữ liệu đã được tính toán và các luật mờ được đưa ra qua quá trình thực nghiệm
- Khâu giải mờ và khâu hậu xử lý trực tiếp để phù hợp với đối tượng điều khiển.



Hình 3.3. Sơ đồ nguyên lý hệ thống được điều khiển bởi Bộ điều khiển Fuzzy

Tuy Fuzzy đã tồn tại lâu và có nhiều phát triển, các nguyên tắc, tính ổn định, chất lượng đáp ứng, sự ảnh hưởng của nhiễu... vẫn còn là vấn đề lớn cho giới khoa học. Việc thiết kế Bộ điều khiển Fuzzy chủ yếu dựa trên kinh nghiệm và tinh chỉnh từng giá trị trong quá trình thực nghiệm. Độ phi tuyến của hệ thống đôi khi lại làm bất ổn định tín hiệu điều khiển.

3.3.2 Xây dựng bộ điều khiển Fuzzy cho hệ Pendubot

Dựa trên 81 luật mờ đã được đề cập trong tài liệu [1] trang 182, nhóm sinh viên thay đổi hệ quy chiếu của hệ Pendubot sao cho phù hợp với hệ quy chiếu đã chọn của tác giả cho hệ Xe con lắc ngược. Giá trị các biến ngõ vào và ngõ ra của bộ điều khiển được trình bày theo *Bảng 3.1*. Các hàm liên thuộc được thể hiện ở *Hình 3.4* → *Hình 3.8*

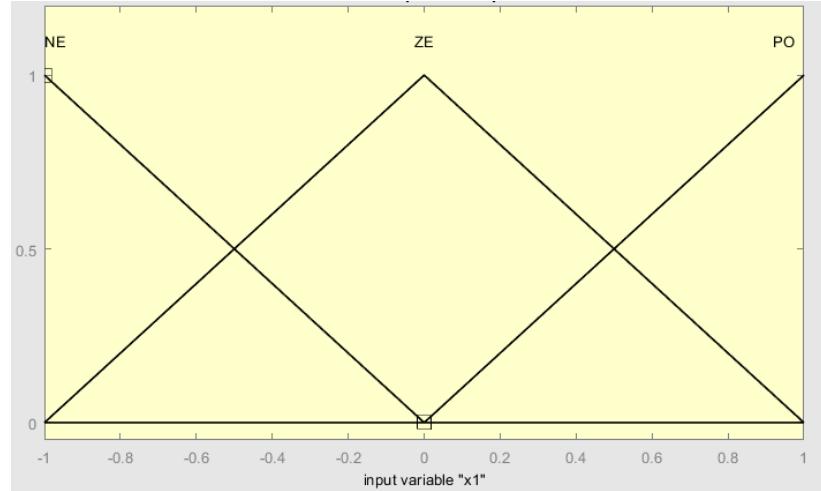
Giá trị ngõ vào sẽ biến đổi từ 0 đến 1 trong khoảng từ -1 đến 1, sau đó sẽ được nhân với hệ số tiền xử lý cho từng biến để khuếch đại tín hiệu đủ lớn để bộ điều khiển xử lý. Tương tự cho giá trị ngõ ra, sẽ biến đổi dần từ 0 đến 1 rồi nhân với hệ số hậu xử lý để đạt được tín hiệu điều khiển đúng như mong muốn. Các thông số tiền xử lý và hậu xử lý là số ngẫu nhiên thuộc tập hợp số thực ($K \in R$).

CHƯƠNG 3. Giải thuật điều khiển

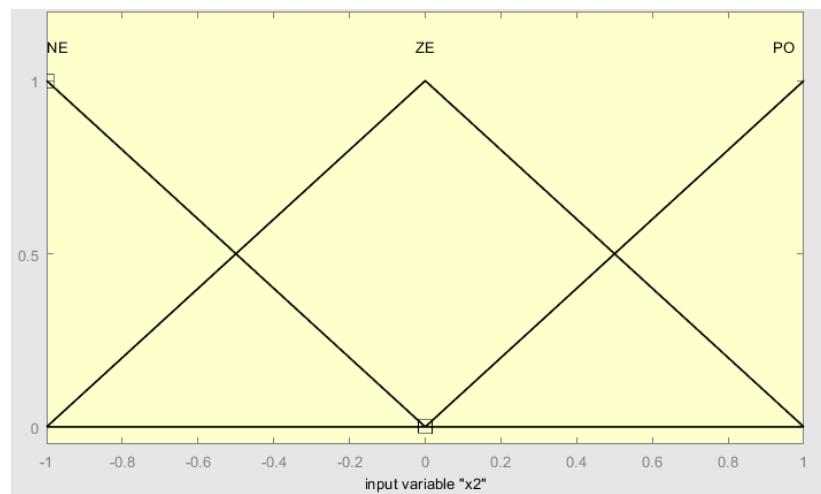
Bảng 3.1. Các biến ngõ vào và ngõ ra của Bộ điều khiển Fuzzy

Kí hiệu	Chú thích
Giá trị cho ngõ vào	
NE	Negative – Âm
ZE	Zero – Không
PO	Positive – Dương
Giá trị cho ngõ ra	
NB	Negative big – Âm nhất
NM	Negative medium – Âm trung bình
NS	Negative small – Âm ít
ZE	Zero – Không
PS	Positive small – Dương ít
PM	Positive medium – Dương trung bình
PB	Positive big – Dương nhất

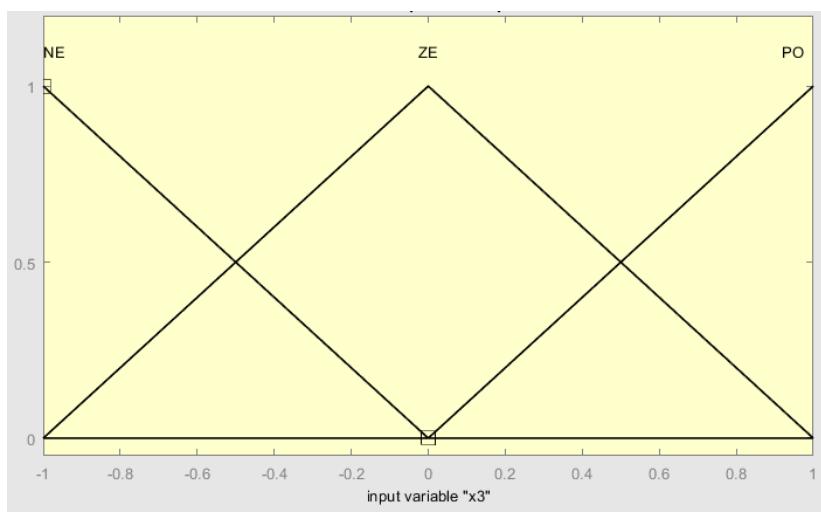
CHƯƠNG 3. Giải thuật điều khiển



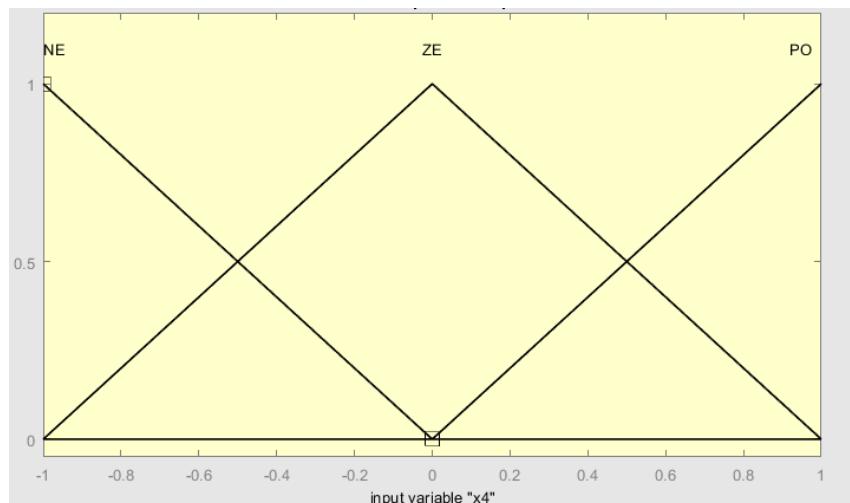
Hình 3.4. Hàm liên thuộc ngõ vào x_1



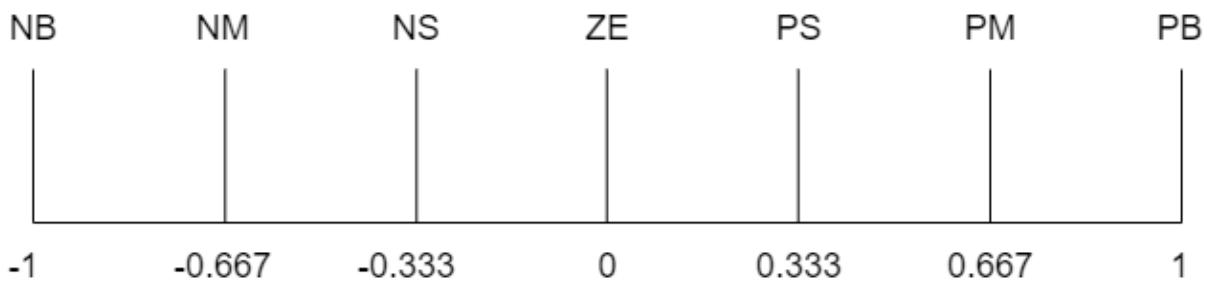
Hình 3.5. Hàm liên thuộc ngõ vào x_2



Hình 3.6. Hàm liên thuộc ngõ vào x_3



Hình 3.7. Hàm liên thuộc ngõ vào x_4



Hình 3.8. Hàm liên thuộc ngõ ra u

3.4 GIẢI THUẬT DI TRUYỀN – GA

3.4.1 Giới thiệu

Khái niệm

Di truyền là quá trình tiến hoá của một quần thể sinh học qua những thế hệ nối tiếp nhau, trong đó thế hệ sau sẽ giữ lại một số đặc tính của thế hệ trước và xuất hiện các thay đổi để phù hợp hơn với điều kiện môi trường. Dựa trên đặc tính di truyền của sinh học trong tự nhiên, vào năm 1975, nhà bác học John Henry Holland lần đầu đưa ra Giải thuật di truyền (Genetic Algorithm – GA) với tư tưởng chính là mô phỏng các hiện tượng tự nhiên, là kế thừa và chọn lọc tự nhiên.

Nguyên lý hoạt động của GA

Việc tìm kiếm giả thuyết thích hợp được bắt đầu với một quần thể. Các cá thể trong quần thể tạo ra thế hệ tiếp theo qua các hoạt động lai ghép và đột biến – được lấy mẫu sau

CHƯƠNG 3. Giải thuật điều khiển

các quá trình tiến hoá sinh học. Ở mỗi bước tiến hoá, các giả thuyết trong quần thể hiện tại được ước lượng với đại lượng thích nghi, các giả thuyết phù hợp nhất được chọn theo xác suất là hạt giống cho việc sản sinh thế hệ kế tiếp. Cá thể nào mạnh hơn, thích nghi tốt hơn với môi trường sẽ được tồn tại và tiếp tục vòng lặp tiến hoá, phần còn lại sẽ bị đào thải ra khỏi vòng tròn sinh học.

Dựa theo cách vận hành của sự tiến hoá trong tự nhiên, GA được tạo ra để giải quyết các bài toán thông qua các quá trình cơ bản: lai tạo, đột biến và chọn lọc. Lai tạo để thu được cá thể mang các đặc trưng của thế hệ đi trước. Đột biến để tạo ra các biến thể đặc biệt với các ưu điểm vượt trội. Chọn lọc để đào thải các cá thể không phát triển được trong môi trường tự nhiên. Áp dụng GA vào một phép tính toán đòi hỏi người thiết kế phải xác định rõ ràng các yếu tố sau:

- Quần thể ban đầu là các lời giải của bài toán
- Hàm mục tiêu – đánh giá lời giải theo mức độ thích nghi
- Các phép toán di truyền tạo hàm sinh sản

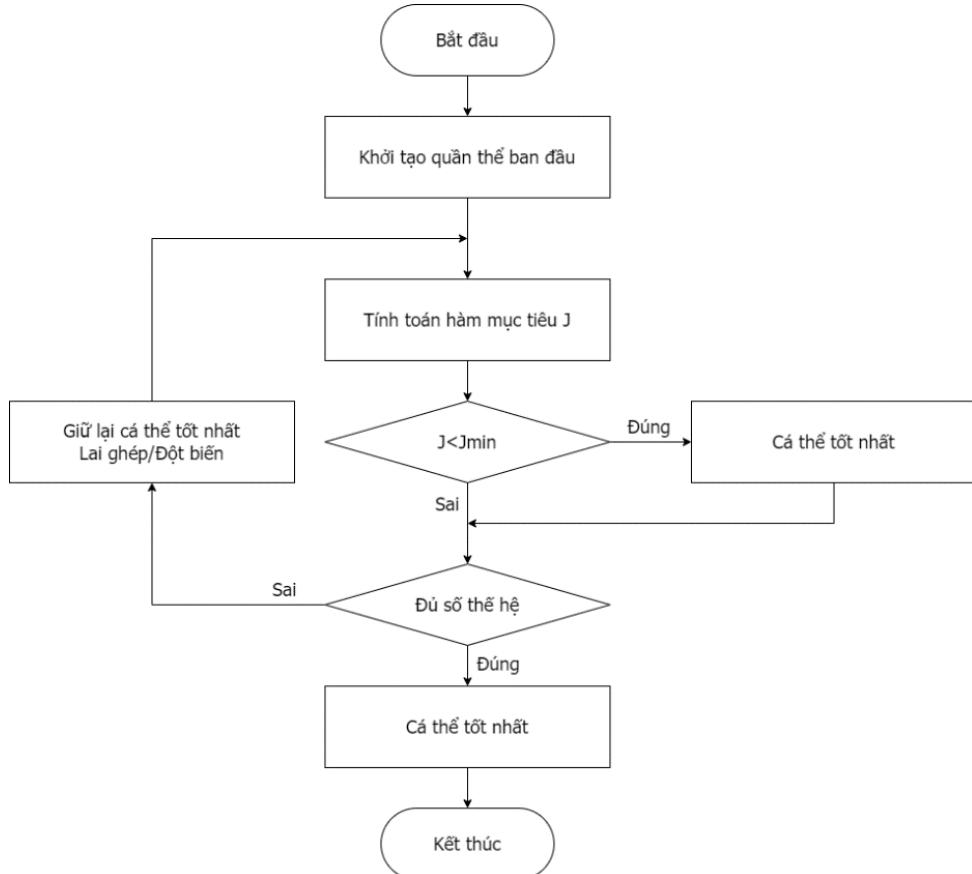
Ưu điểm của GA

Sự khác biệt lớn nhất giữa việc chọn lọc bằng GA và các phép chọn lọc khác là GA có khả năng xử lý một khối lượng dữ liệu lớn, gọi là quần thể. Quần thể bao gồm nhiều nhiễm sắc thể, tức là lời giải của phép tính. Bằng các phép toán di truyền, GA trao đổi thông tin giữa những đỉnh cực trị, hạn chế tối đa khả năng kết quả tìm được rơi vào cực trị cục bộ. Song song đó, GA đánh giá tính tối ưu của kết quả qua hàm mục tiêu mà không cần bất cứ thông tin nào thêm. Do đó, GA có khả năng giải quyết hiệu quả các phép tính toán cho hàm phi tuyến, không khả vi...

3.4.2 Áp dụng giải thuật GA tối ưu hoá các bộ điều khiển

Phương pháp dùng GA để tối ưu hoá các bộ điều khiển được trình bày theo dạng lưu đồ như ở *Hình 3.9*. Tìm kiếm giả thuyết ban đầu được khởi tạo với một quần thể, là tập hợp có chọn lọc các giải thuyết đầu vào, là tập hợp các chuỗi nhiễm sắc thể - đại diện cho mỗi cá thể riêng biệt. Để đánh giá các cá thể, phải tính toán một hàm thích nghi, gọi là hàm mục tiêu J. Qua quá trình chọn lọc, thế hệ nào có độ thích nghi cao thì khả năng tồn tại cao hơn, khả năng sinh sản để tạo ra thế hệ sau có xu hướng thích nghi tốt hơn thế hệ trước. Qua một thời gian tiến hoá với số lần lai tạo nhất định, GA sẽ tìm ra cá thể tốt nhất, tức là lời giải tốt nhất cho bài toán đầu vào. Càng nhiều lần lai tạo, lời giải của bài toán dần tiến đến kết quả tối ưu.

CHƯƠNG 3. Giải thuật điều khiển



Hình 3.9. Lưu đồ chương trình GA

Hàm mục tiêu được chọn như sau:

$$J = \sum_1^n x_1^T x_1 + x_3^T x_3 \quad (50)$$

Trong đó, x_1 và x_3 tương ứng với góc thanh 1 và thanh 2 so với phuong Oy, n là tổng số mẫu. Ví dụ chương trình viết trên phần mềm Matlab SIMULINK cho khởi chạy trong 5 giây và thời gian lấy mẫu là 0.01, vậy $n = 501$.

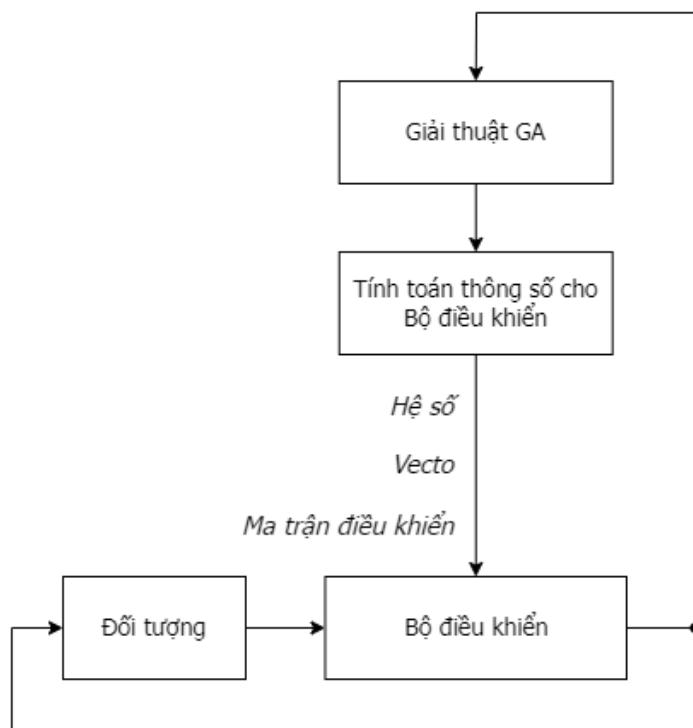
Giải thuật GA tối ưu hệ số cho Bộ điều khiển

Đối với Bộ điều khiển Tuyến tính hoá hồi tiếp: các hệ số quyết định đáp ứng của hệ thống là ma trận K.

Đối với Bộ điều khiển LQR: ma trận điều khiển K cần được tối ưu. K bị ảnh hưởng bởi ma trận trọng số Q và hệ số năng lượng R. Đối tượng cần dùng GA để tìm ra các hệ số tốt nhất là Q và R.

CHƯƠNG 3. Giải thuật điều khiển

Dối với Bộ điều khiển Fuzzy: cần xác định đúng các hệ số tiền xử lý và hậu xử lý để ngõ ra của hệ thống về 0 nhanh nhất và ổn định nhất.



Hình 3.10. Sơ đồ nguyên lý hệ thống được điều khiển bởi Bộ điều khiển GA kết hợp

Theo Hình 3.10, đầu vào của giải thuật GA là đáp ứng của hệ thống. Sau các phép lai tạo và chọn lọc, dựa trên giá trị hàm mục tiêu, giá trị các hệ số thích hợp nhất sẽ được đưa vào bộ điều khiển để đưa ra tín hiệu điều khiển (giá trị điện áp hoặc momen) thích hợp cho hệ Pendubot cân bằng.

Chương 4 MÔ PHỎNG

4.1 THÔNG SỐ MÔ PHỎNG

Các thông số cơ học của hệ thống đã được liệt kê đầy đủ ở *Bảng 2.1*. Giá trị ban đầu của các biến trạng thái được đặt như *Bảng 4.1*.

Bảng 4.1. Biến trạng thái ban đầu

Giá trị	Thông số
\dot{x}_1_{init}	0.05
\dot{x}_2_{init}	0.05
\dot{x}_3_{init}	0.05
\dot{x}_4_{init}	0.05

4.1.1 Thông số cho Bộ điều khiển LQR

Nhóm sinh viên chú trọng vào độ dao động của các thanh, do đó sẽ tinh chỉnh cái giá trị Q_2 và Q_4 . Từ phương trình toán (19)(20), tính được ma trận tuyến tính (22) →(25). Chọn ma trận trọng số Q theo (51) và hệ số năng lượng R theo (52), tính được ma trận điều khiển K (53).

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 350 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 350 \end{bmatrix} \quad (51)$$

$$R = 0.01 \quad (52)$$

$$K = 10^3 \times [-0.0313 \quad 0.1970 \quad 3.5168 \quad 0.5409] \quad (53)$$

4.1.2 Thông số cho Bộ điều khiển Tuyến tính hoá hồi tiếp

Kết hợp GA và Bộ điều khiển tuyến tính hoá hồi tiếp, nhóm sinh viên tìm được bộ ma trận K điều khiển ổn định nhất theo *Bảng 4.2*.

CHƯƠNG 4. Mô phỏng

Bảng 4.2. Thông số của Bộ điều khiển Tuyến tính hoá hồi tiếp

Kí hiệu	Giá trị
K1	62.1111
K2	941.4010
K3	96.5010
K4	62.5310

4.1.3 Thông số cho Bộ điều khiển Fuzzy

Nhóm sinh viên tiến hành dùng GA để tìm các thông số tối ưu nhất cho Bộ điều khiển Fuzzy của hệ Pendubot. Dựa trên lý thuyết và các phép tính toán, các hệ số tiền xử lý và hậu xử lý được trình bày ở Bảng 4.3.

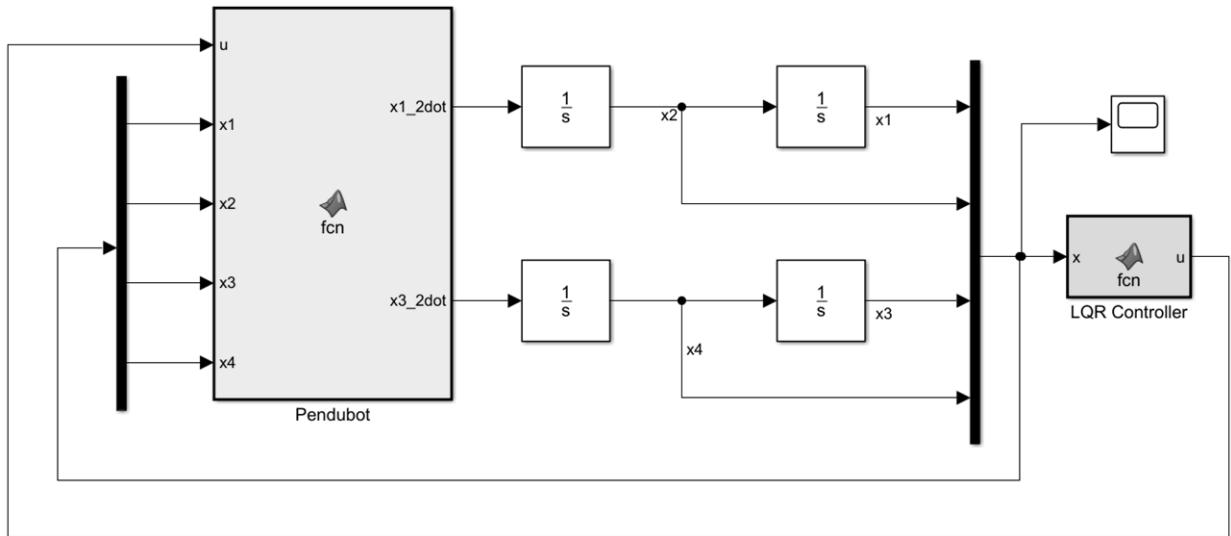
Bảng 4.3. Thông số của Bộ điều khiển Fuzzy

Kí hiệu	Giá trị
K1	-4.6
K2	1.4
K3	-4.3
K4	-2.0
K5	-20

4.2 CHƯƠNG TRÌNH MÔ PHỎNG

4.2.1 Chương trình mô phỏng Bộ điều khiển LQR

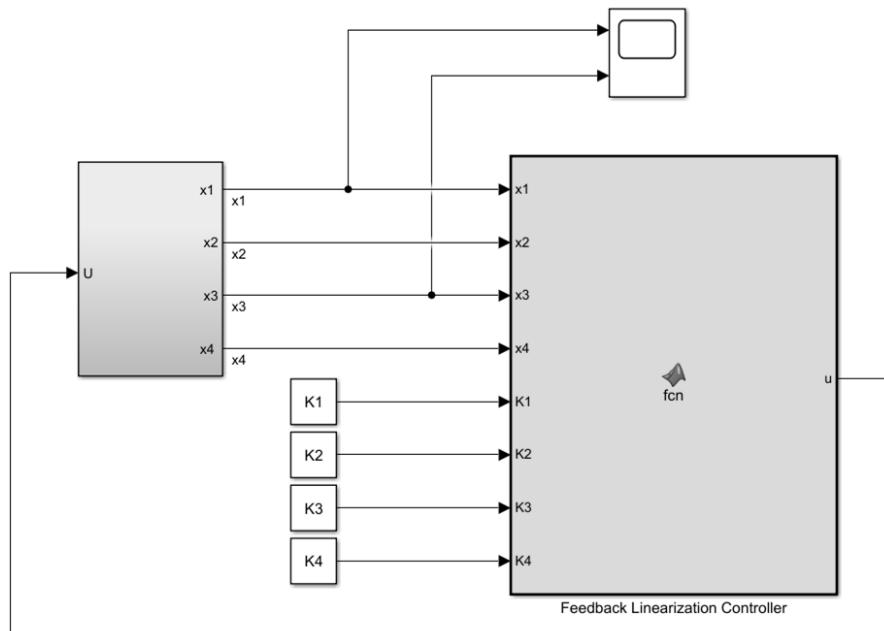
Chương trình mô phỏng Bộ điều khiển LQR cho hệ Pendubot được trình bày như Hình 4.1 với các thông số đã được nêu ở (53). Trong đó, khối “Pendubot” chứa các phương trình (14)→(20) mô tả phần cứng hệ thống, khối “LQR Controller” là khối chứa Bộ điều khiển LQR với ngõ vào là đáp ứng hệ thống sau khi điều khiển và ngõ ra là điện áp u .



Hình 4.1. Chương trình mô phỏng Bộ điều khiển LQR cho hệ Pendubot

4.2.2 Chương trình mô phỏng Bộ điều khiển Tuyến tính hóa hồi tiếp

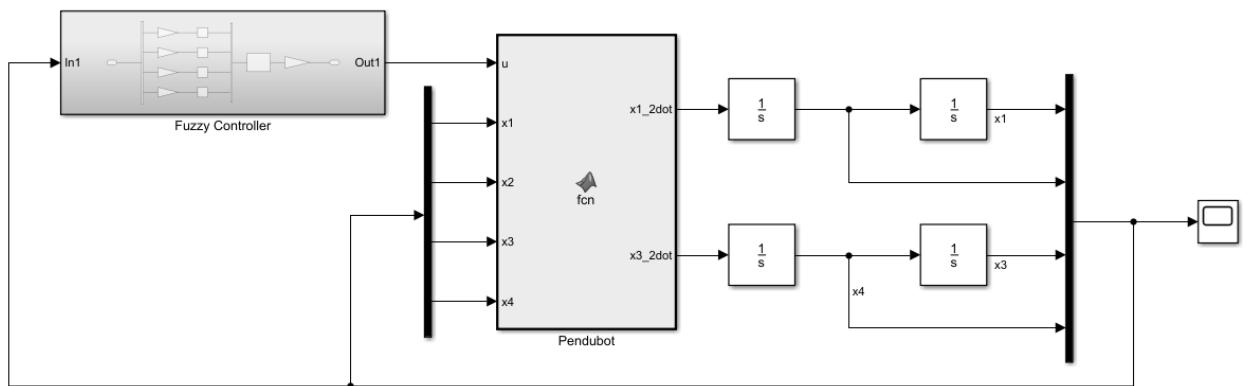
Chương trình mô phỏng Bộ điều khiển Tuyến tính hóa hồi tiếp cho hệ Pendubot được trình bày như *Bảng 4.2*. Trong đó, khối “Pendubot” chứa các phương trình (14)→(20) mô tả phần cứng hệ thống, khối “Feedback Linearization Controller” là khối chứa Bộ điều khiển Tuyến tính hóa hồi tiếp với ngõ vào là đáp ứng hệ thống sau khi điều khiển và các hệ số của ma trận K đã trình bày ở *Bảng 4.2*, ngõ ra là momen của động cơ.



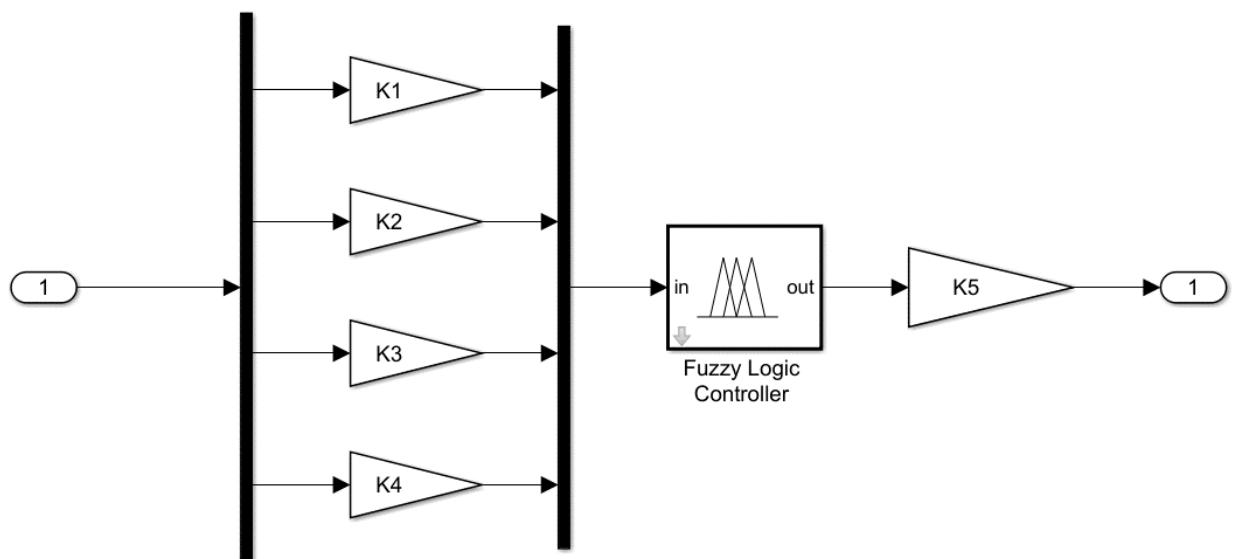
Hình 4.2. Chương trình mô phỏng Bộ điều khiển Tuyến tính hóa hồi tiếp
cho hệ Pendubot

4.2.3 Chương trình mô phỏng Bộ điều khiển Fuzzy

Chương trình mô phỏng Bộ điều khiển Fuzzy cho hệ Pendubot được trình bày như *Hình 4.3* với các thông số đã được xử lý qua GA và được ghi rõ ở *Bảng 4.3*. Trong đó, khối “Pendubot” chứa các phương trình (14)→(20) mô tả phần cứng hệ thống, khối “Fuzzy Controller” là khối chứa Bộ điều khiển Fuzzy với ngõ vào là đáp ứng hệ thống sau khi điều khiển và ngõ ra là điện áp u . Các khối lệnh bên trong khối “Fuzzy Controller” được thể hiện như ở *Hình 4.4*.



Hình 4.3. Chương trình mô phỏng Bộ điều khiển Fuzzy cho hệ Pendubot



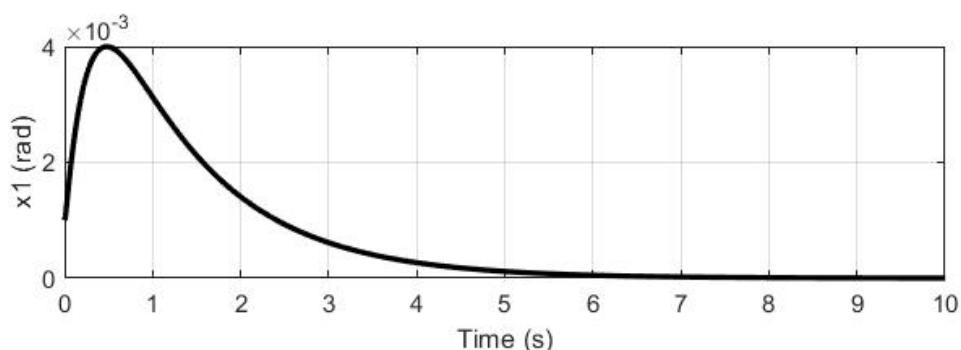
Hình 4.4. Chương trình của khối Fuzzy Controller

4.3 KẾT QUẢ MÔ PHỎNG

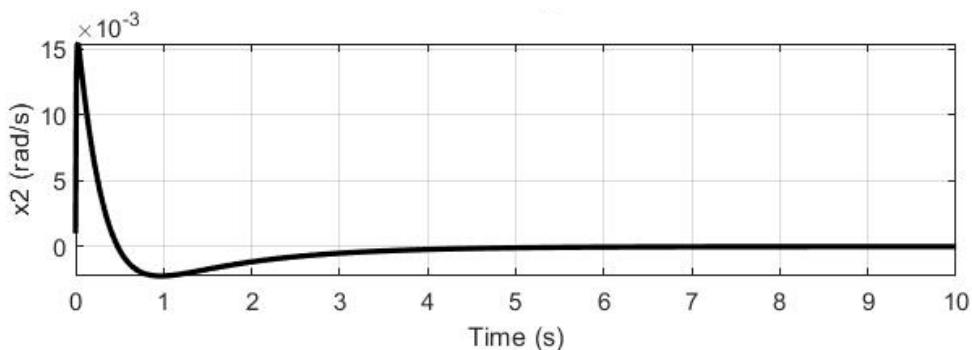
4.3.1 Kết quả mô phỏng Bộ điều khiển LQR

Kết quả mô phỏng điều khiển hệ Pendubot bằng Bộ điều khiển LQR được biểu diễn theo *Hình 4.5* → *Hình 4.8*. Trong đó:

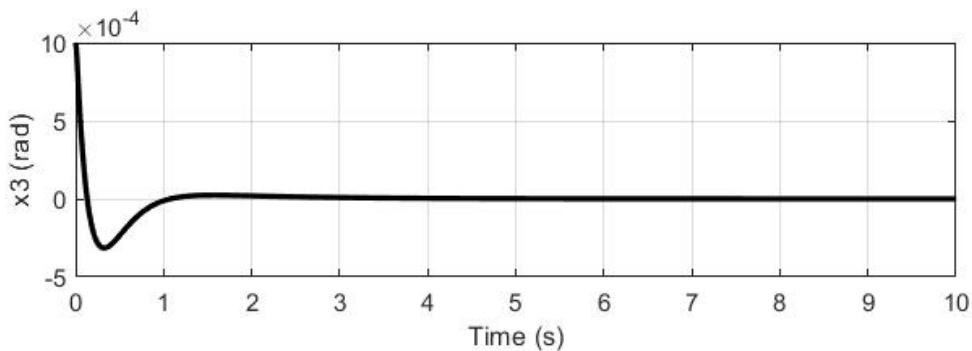
- *Hình 4.5* là kết quả mô phỏng cho góc thanh 1, kí hiệu là x_1 , đơn vị tính là rad
- *Hình 4.6* là kết quả mô phỏng cho vận tốc thanh 1, kí hiệu là x_2 , đơn vị tính là rad/s
- *Hình 4.7* là kết quả mô phỏng cho góc thanh 2, kí hiệu là x_3 , đơn vị tính là rad
- *Hình 4.8* là kết quả mô phỏng cho vận tốc thanh 2, kí hiệu là x_4 , đơn vị tính là rad/s



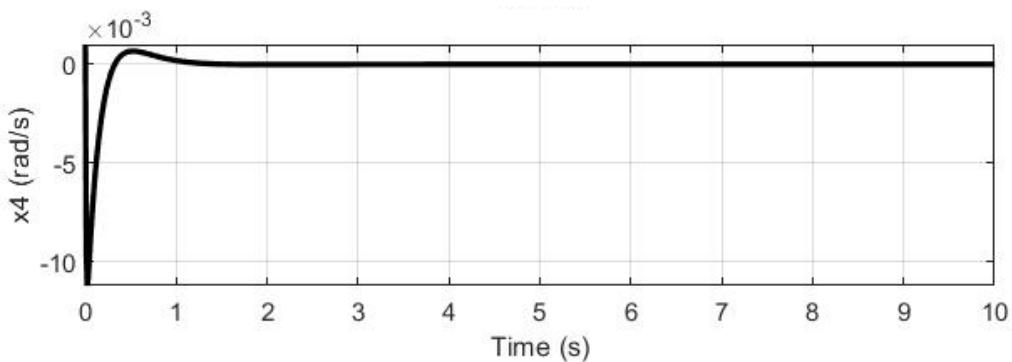
Hình 4.5. Kết quả mô phỏng LQR cho góc thanh 1



Hình 4.6. Kết quả mô phỏng LQR cho vận tốc thanh 1



Hình 4.7. Kết quả mô phỏng LQR cho góc thanh 2



Hình 4.8. Kết quả mô phỏng LQR cho vận tốc thanh 2

Nhận xét

Tại vị trí ban đầu đã đặt theo *Bảng 4.1*, kết quả của các biến ngõ ra trả về đều ổn định ở vị trí 0:

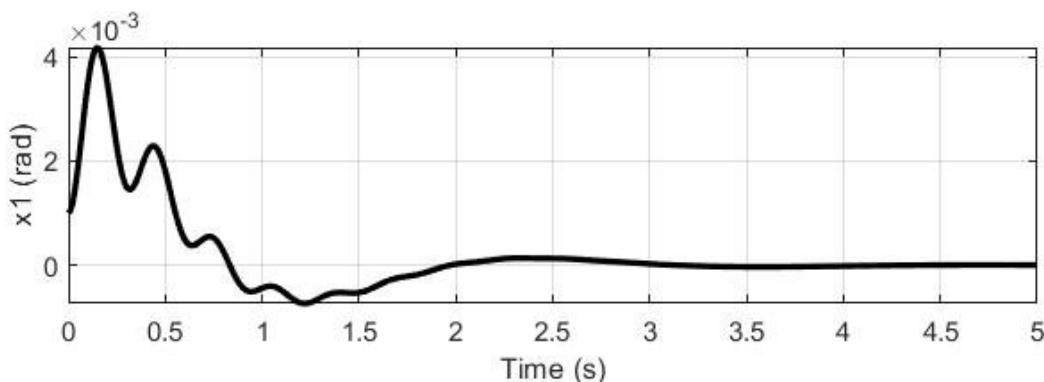
- Thanh 1 đứng thẳng lên và ngừng chuyển động sau 6s
- Thanh 2 đứng thẳng lên và ngừng chuyển động sau 1.5s

Như vậy, bộ điều khiển LQR đã cân bằng hệ Pendubot sau 6s.

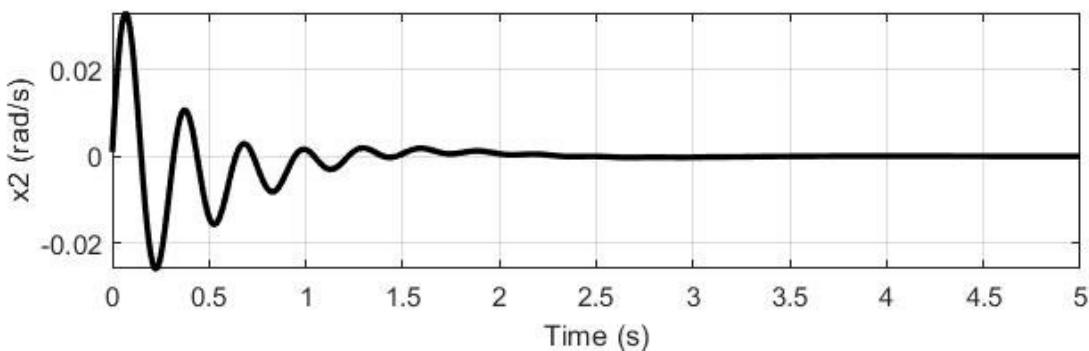
4.3.2 Kết quả mô phỏng Bộ điều khiển Tuyến tính hóa hồi tiếp

Kết quả mô phỏng điều khiển hệ Pendubot bằng Bộ điều khiển Tuyến tính hóa hồi tiếp được biểu diễn theo *Hình 4.9 → Hình 4.12*. Trong đó:

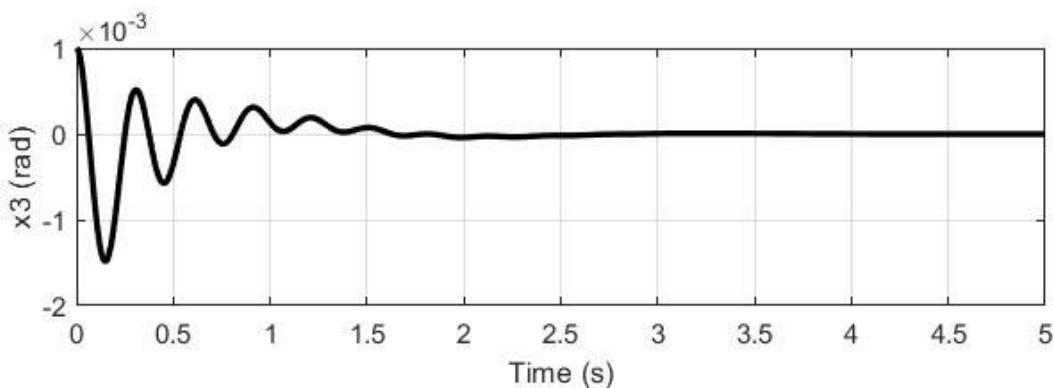
- *Hình 4.9* là kết quả mô phỏng cho góc thanh 1, kí hiệu là x_1 , đơn vị tính là rad
- *Hình 4.10* là kết quả mô phỏng cho vận tốc thanh 1, kí hiệu là x_2 , đơn vị tính là rad/s
- *Hình 4.11* là kết quả mô phỏng cho góc thanh 2, kí hiệu là x_3 , đơn vị tính là rad
- *Hình 4.12* là kết quả mô phỏng cho vận tốc thanh 2, kí hiệu là x_4 , đơn vị tính là rad/s



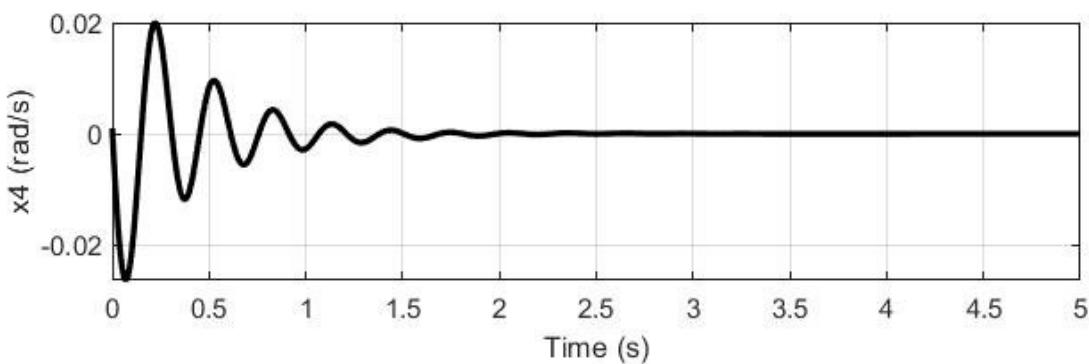
Hình 4.9. Kết quả mô phỏng Tuyến tính hóa hồi tiếp cho góc thanh 1



Hình 4.10. Kết quả mô phỏng Tuyến tính hoá hồi tiếp cho vận tốc thanh 1



Hình 4.11. Kết quả mô phỏng Tuyến tính hoá hồi tiếp cho góc thanh 2



Hình 4.12. Kết quả mô phỏng Tuyến tính hoá hồi tiếp cho vận tốc thanh 2

Nhận xét

Tại vị trí ban đầu đã đặt theo *Bảng 4.1*, kết quả của các biến ngõ ra trả về đều ổn định ở vị trí 0:

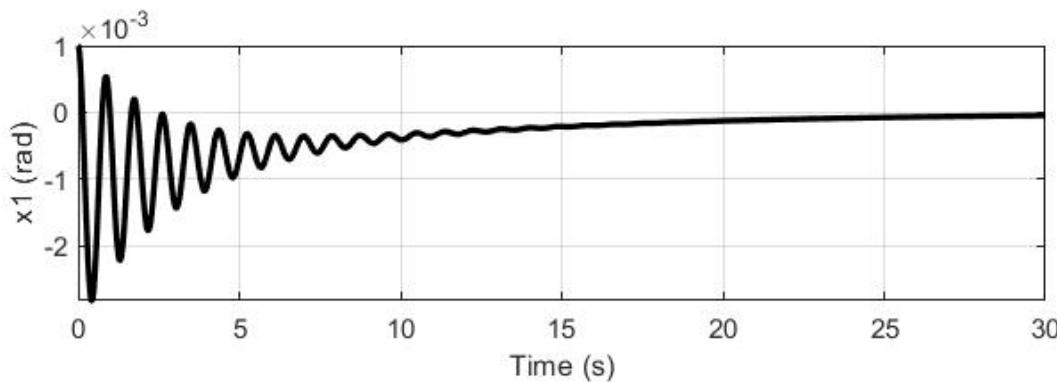
- Thanh 1 đứng thẳng lên và ngừng chuyển động sau 3.5s
- Thanh 2 đứng thẳng lên và ngừng chuyển động sau 2.5s

Như vậy, bộ điều khiển Tuyến tính hoá hồi tiếp cân bằng hệ Pendubot sau 3.5s.

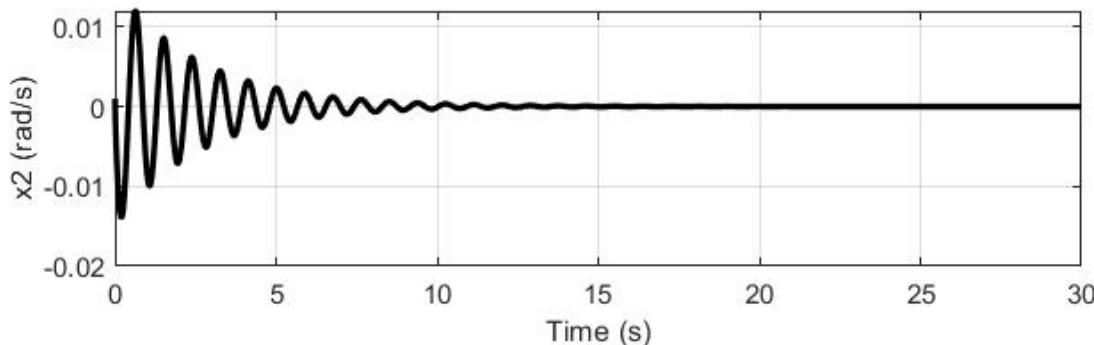
4.3.3 Kết quả mô phỏng Bộ điều khiển Fuzzy

Kết quả mô phỏng điều khiển hệ Pendubot bằng Bộ điều khiển Fuzzy được biểu diễn theo *Hình 4.13 → Hình 4.16*. Trong đó:

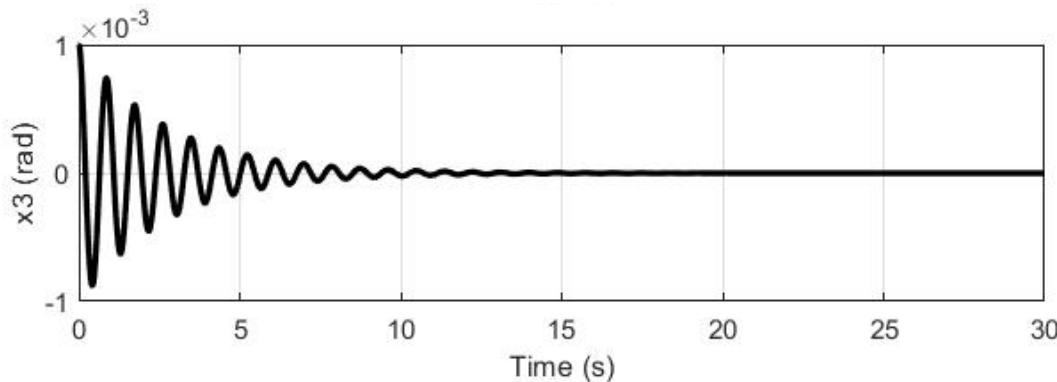
- *Hình 4.13* là kết quả mô phỏng cho góc thanh 1, kí hiệu là x_1 , đơn vị tính là rad
- *Hình 4.14* là kết quả mô phỏng cho vận tốc thanh 1, kí hiệu là x_2 , đơn vị tính là rad/s
- *Hình 4.15* là kết quả mô phỏng cho góc thanh 2, kí hiệu là x_3 , đơn vị tính là rad
- *Hình 4.16* là kết quả mô phỏng cho vận tốc thanh 2, kí hiệu là x_4 , đơn vị tính là rad/s



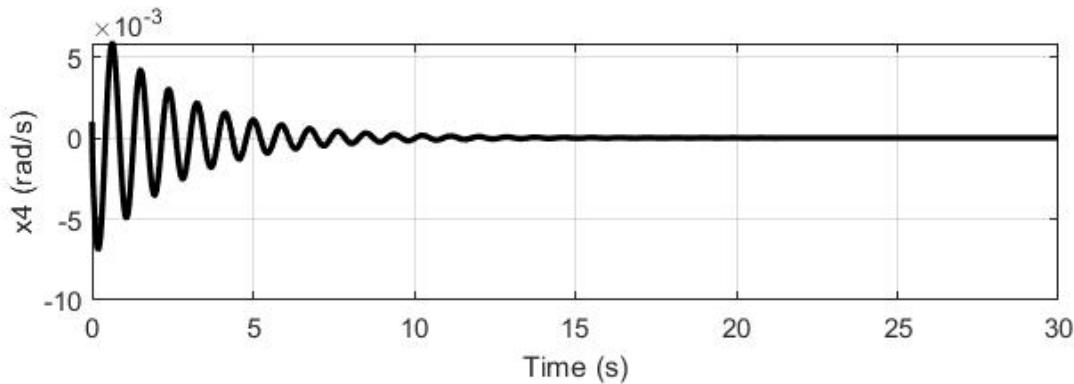
Hình 4.13. Kết quả mô phỏng Fuzzy cho góc thanh 1



Hình 4.14. Kết quả mô phỏng Fuzzy cho vận tốc thanh 1



Hình 4.15. Kết quả mô phỏng Fuzzy cho góc thanh 2



Hình 4.16. Kết quả mô phỏng Fuzzy cho vận tốc thanh 2

Nhận xét

Tại vị trí ban đầu đã đặt theo *Bảng 4.1*, kết quả của các biến ngõ ra trả về đều ổn định ở vị trí 0:

- Thanh 1 đứng thẳng lên và ngừng chuyển động sau 25s
- Thanh 2 đứng thẳng lên và ngừng chuyển động sau 15s

Như vậy, bộ điều khiển Fuzzy cân bằng hệ Pendubot sau 25s.

4.4 NHẬN XÉT

Nhìn chung, các bộ điều khiển đều cân bằng được thành công hệ Pendubot. Trong đó, Bộ điều khiển LQR ít dao động nhất, Bộ điều khiển Tuyến tính hoá hồi tiếp đáp ứng nhanh nhất.

Dáp ứng ngõ ra từ phương pháp điều khiển Fuzzy dao động nhiều quanh điểm 0 và thời gian quá độ dài (25s). Do đó khả năng điều khiển thực nghiệm cho hệ Pendubot bằng phương pháp điều khiển này không khả thi.

Bộ điều khiển LQR và Tuyến tính hoá hồi tiếp là 2 phương pháp mà nhóm sinh viên chọn để tiến hành điều khiển trên mô hình thực tế.

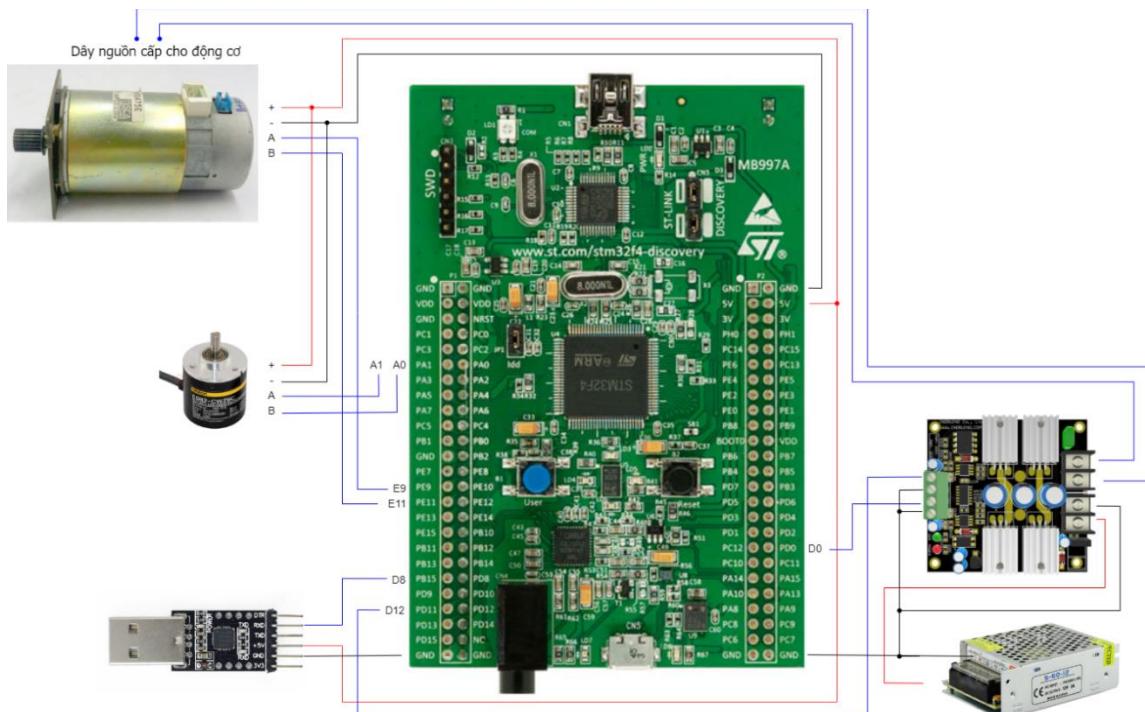
Chương 5 THỰC NGHIỆM

5.1 MÔ HÌNH

5.1.1 Thiết kế phần điện

Phần điện của hệ thống Pendubot gồm 7 khối chính:

- Vi điều khiển đóng vai trò là bộ xử lý trung tâm của hệ thống
- Động cơ
- Encoder thanh 1
- Encoder thanh 2
- Cầu H
- Module chuyển đổi USB – UART
- Bộ nguồn



Hình 5.1. Sơ đồ kết nối phần điện của hệ thống

Vi điều khiển STM32F4 đọc các giá trị từ 2 encoder đưa về, sau đó sẽ tính toán và cấp tín hiệu ra cầu H để điều khiển động cơ sao cho góc thanh 1 và thanh 2 đều về 0. Song song đó, STM32F4 sẽ giao tiếp với máy tính thông qua module chuyển đổi USB – UART.

Sơ đồ đấu dây chi tiết được trình bày như *Hình 5.1*. Encoder thanh 1 gắn trực tiếp vào động cơ. Nhóm sinh viên sử dụng nguồn tò ong cấp nguồn cho cầu H để điều khiển động cơ.

Vi điều khiển STM32F407VG

Kit STM32F407G-DISC1 là loại kit thông dụng nhất ở nhiều trường đại học hiện nay trong học tập và giảng dạy vi điều khiển ARM. Ưu điểm của loại vi điều khiển này là về khả năng may đo theo nhu cầu người lập trình (ví dụ như tuỳ chỉnh thời gian cho tín hiệu xung)



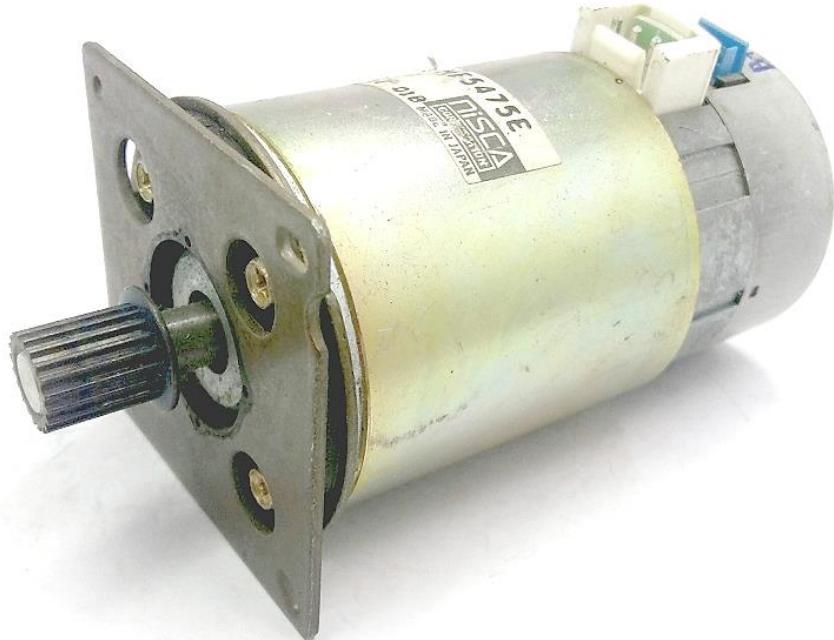
Hình 5.2. Hình ảnh thực tế STM32F407VG

- MCU STM32F407VG có lõi 32-bit ARM Cortex-M4F, Flash 1MB, RAM 192KB trong một gói LQFP100
- ST-LINK/V2 được thiết kế có khả năng hoạt động ở tốc độ cao, thực hiện hàng triệu lệnh mỗi giây (Million Instructions per second – MIPS)
- Nguồn cấp: thông qua USB hoặc từ nguồn điện 5V bên ngoài
- STM32F407VG có 5 port chính A, B, C, D, E, mỗi port có 16 chân và được ký hiệu từ 0 đến 15, trong đó các chân PD12 → PD15 được kết nối sẵn với 4 LED được trang bị trên bề mặt kit.
- STM32F407VG bao gồm 8 chip LED:
 - LD1 (đỏ/xanh) để hiển thị giao tiếp USB
 - LD2 (màu đỏ) hiển thị nguồn 3.3 VDC

CHƯƠNG 5. Thực nghiệm

- 4 LED có thể lập trình điều khiển, bao gồm: LD3 (màu cam) nối với ngõ ra PD13, LD4 (màu xanh lá) nối với ngõ ra PD12, LD5 (màu đỏ) nối với ngõ ra PD14 và LD6 (màu xanh dương) nối với ngõ ra PD15.
- 2 đèn LED USB OTG LD7 (màu xanh lá) VBus và LD8 (màu đỏ) biểu hiện quá dòng
- 2 nút nhấn: Reset và User
- USB OTG FS với cổng micro-AB.

Động cơ Nisca NF5475E



Hình 5.3. Hình ảnh thực tế động cơ NF5475E

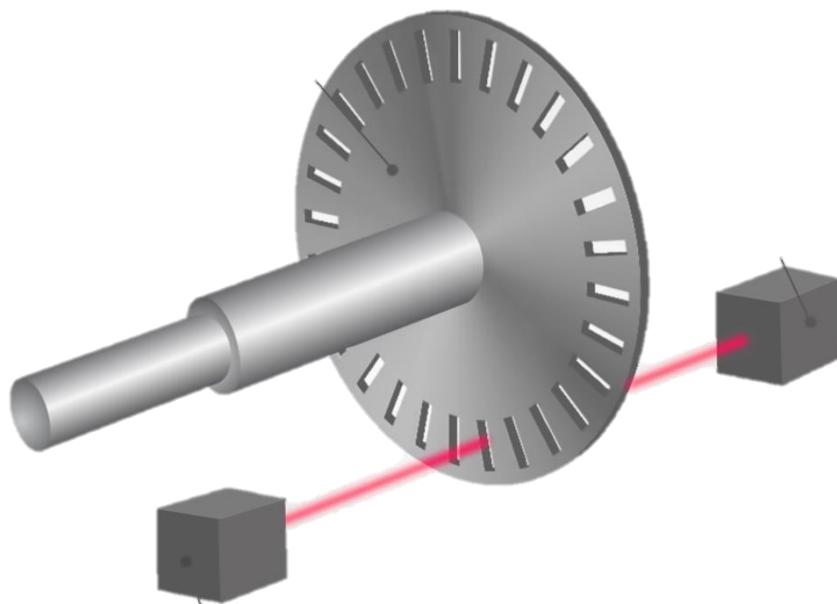
Đây là động cơ DC có tích hợp sẵn encoder đọc xung tốc độ cao, với các thông số chính như sau:

- Trục động cơ Ø8 mm
- Điện áp: 12 – 24 V
- Công suất < 40 W
- Tốc độ quay tối đa 4800 vòng/phút
- Encoder gồm 2 kênh, phân giải 200 xung/vòng khi cấp nguồn 5V cho encoder

So với các động cơ DC thông dụng trên thị trường, NF5475E có phần trục động cơ to hơn. Điều này giúp dễ dàng cho việc phát triển mô hình thành ứng dụng thực tế.

Encoder

Encoder dùng để chuyển đổi vị trí góc hoặc chuyển động của trục encoder thành tín hiệu đầu ra analog/digital. Encoder được dùng để phát hiện vị trí, hướng di chuyển, tốc độ... của động cơ bằng cách đếm số vòng quay được của trục.



Hình 5.4. Hình ảnh minh họa nguyên lý hoạt động của encoder

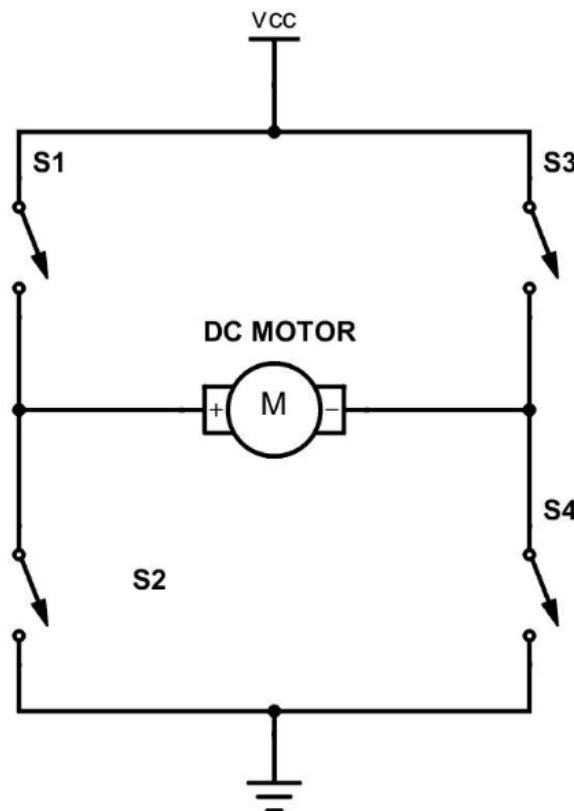
Theo *Hình 5.4*, một encoder cơ bản bao gồm đĩa quang, nguồn sáng và cảm biến quang thu tín hiệu. Đĩa quang là đĩa kim loại tròn, có đục các lỗ nhỏ quanh trục encoder và được gắn chặt với trục encoder. Nguồn sáng là LED, được đặt đối xứng với bộ thu tín hiệu qua vị trí các lỗ của đĩa quang. Khi cấp nguồn cho encoder, LED sáng lên, phát tín hiệu cho bộ thu. Khi đĩa quay, bộ thu sẽ nhận được tín hiệu qua các lỗ đã đục sẵn. Ngược lại, vị trí không có lỗ sẽ làm ngắt quãng tín hiệu. Đây là nguyên lý hoạt động của encoder. Từ các tín hiệu có/không, kết quả thu được sẽ là số xung của encoder.

Cảm biến quang sẽ liên tục nhận các tín hiệu 0/1, từ đó tạo ra các tín hiệu xung vuông. Tín hiệu này sẽ được truyền về bộ xử lý trung tâm, tiến hành các phép tính toán, đo đặc, xác định vị trí, tốc độ ...

Cầu H IR2184

Cầu H là trung gian để giao tiếp giữa vi điều khiển và động cơ, dùng để điều tốc và đảo chiều động cơ. Động cơ quay thuận hay nghịch tùy thuộc vào việc mắc dây A và B cho động cơ.

Sơ đồ cấu tạo của một mạch cầu H được trình bày như *Hình 5.5*. Một mạch cầu H gồm 4 công tắc, 4 công tắc này có thể là relay hoặc các khoá bán dẫn (transistor, mosfet hoặc IGBT). Bảng logic các tín hiệu điều khiển cho cầu H và trạng thái hoạt động của động cơ được liệt kê ở *Bảng 5.1*.



Hình 5.5. Sơ đồ cấu tạo cơ bản của mạch cầu H

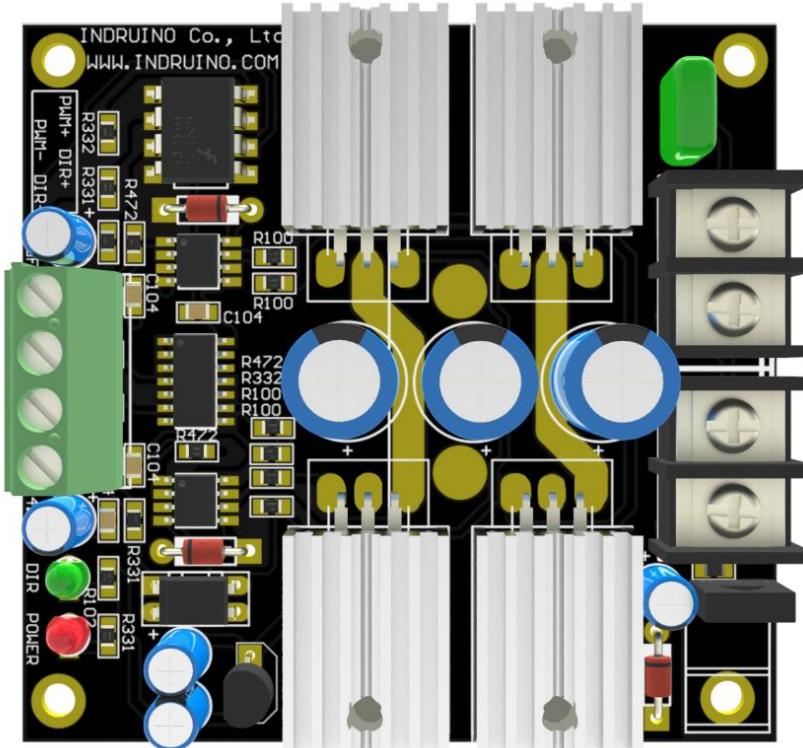
Bảng 5.1. Bảng logic trạng thái hoạt động động cơ

S1	S2	S3	S4	Trạng thái động cơ
0	0	0	0	Không hoạt động
1	0	0	1	Quay thuận
0	1	1	0	Quay nghịch
1	1	X	X	Ngắt mạch
X	X	1	1	Ngắt mạch

CHƯƠNG 5. Thực nghiệm

Theo *Bảng 5.1*, S1 S2 S3 và S4 lần lượt là trạng thái các công tắc, 0 là trạng thái đóng, 1 là trạng thái mở, X là trạng thái không quan tâm. Động cơ có 2 dây (+) và (-), tùy vào cách thứ tự đóng ngắt của các công tắc mà quy định chiều quay của trục động cơ.

Trong đề tài này, nhóm sinh viên chọn cầu H Mosfet do có khả năng điều khiển động cơ có công suất lớn, đồng thời không gây hao phí qua nhiệt năng trong quá trình sử dụng cũng như khả năng bảo vệ mạch điều khiển ở các khâu phía sau.



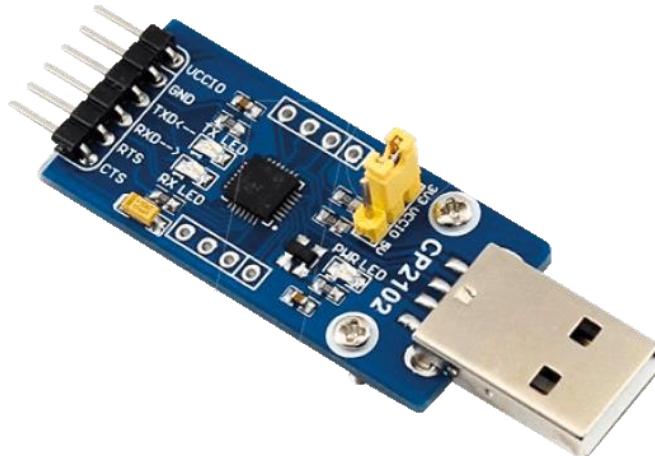
Hình 5.6. Hình ảnh board Cầu H IR2184

Bảng 5.2. Thông số kỹ thuật Cầu H IR2184

Điện áp cấp	12 – 36 VDC
Điện áp logic	0.8 V (H), 2.7 V (L)
Dòng định	30 A
Tần số đóng cắt an toàn	40 kHz

Module chuyển đổi USB – UART CP2102

Vi điều khiển STM không giao tiếp được với máy tính, do đó cần module chuyển đổi dữ liệu để trực quan hơn trong quá trình điều khiển và thu thập dữ liệu. Module CP2101 rất dễ sử dụng, hỗ trợ nhiều tốc độ đọc khác nhau, tương thích trên nhiều hệ điều hành.



Hình 5.7. Hình ảnh thực tế Module chuyển đổi CP2102

CP2102 gồm 6 chân cắm, chức năng của các chân lần lượt là:

- TXD: chân truyền dữ liệu UART TTL (3.3 VDC), dùng kết nối đến chân nhận RX của các module sử dụng mức tín hiệu TTL 3.3 – 5 VDC
 - RXD: chân nhận dữ liệu UART TTL (3.3 VDC), dùng kết nối đến chân truyền dữ liệu TX của các module sử dụng mức tín hiệu TTL 3.3 – 5 VDC
 - GND: chân mass hoặc nối đất.
 - 5V: Chân cấp nguồn 5 VDC từ cổng USB, tối đa 500mA.
 - DTR: Chân tín hiệu DTR, thường được dùng để cấp tín hiệu Reset nạp chương trình cho mạch Arduino.
 - 3.3V: Chân nguồn 3.3 VDC (dòng cấp rất nhỏ tối đa 100mA), không sử dụng để cấp nguồn, thường chỉ sử dụng để thiết đặt mức tín hiệu Logic.

5.1.2 Thiết kế phần cơ

Mô hình Pendubot được cấu tạo từ mica, các chi tiết được lắp ghép bằng ốc, con tán và keo. Mô hình bao gồm:

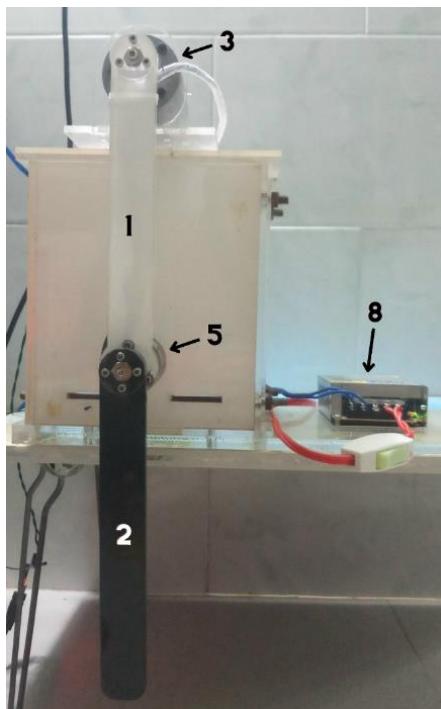
- Khung hộp bằng các tấm mica 5 mm ghép lại, tạo không gian trống bên trong để chứa các board mạch. Bên phải mô hình còn có không gian trống để đặt bộ nguồn 12 VDC cấp nguồn cho động cơ hoạt động

CHƯƠNG 5. Thực nghiệm

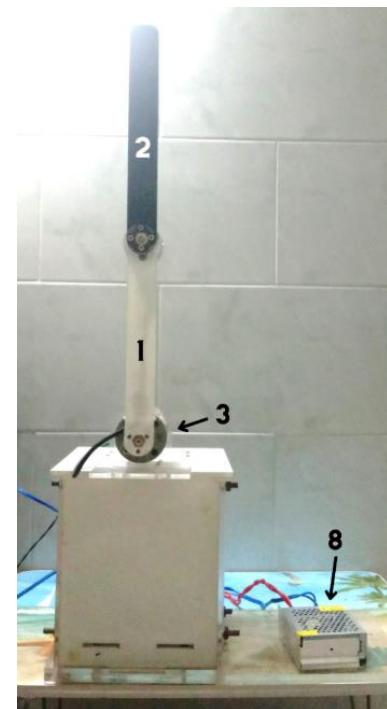
- Động cơ được gắn trực tiếp lên hộp, phía đuôi động cơ được tích hợp sẵn encoder với độ phân giải 200 xung
- Thanh 1 dài 20 cm dày 5 mm bằng mica, 1 đầu gắn với trục động cơ thông qua mặt bích, đầu còn lại liên kết với thanh 2
- Thanh 2 dài 22 cm dày 5 mm bằng mica. 1 đầu kết nối với thanh 1 qua mặt bích và encoder 600 xung, đầu còn lại để trống; thanh 2 có khả năng quay tự do trong mặt phẳng.

Hình 5.8 → Hình 5.11 là hình chụp thực tế của hệ Pendubot. Các thành phần của mô hình được chú thích chi tiết trên từng hình, bao gồm:

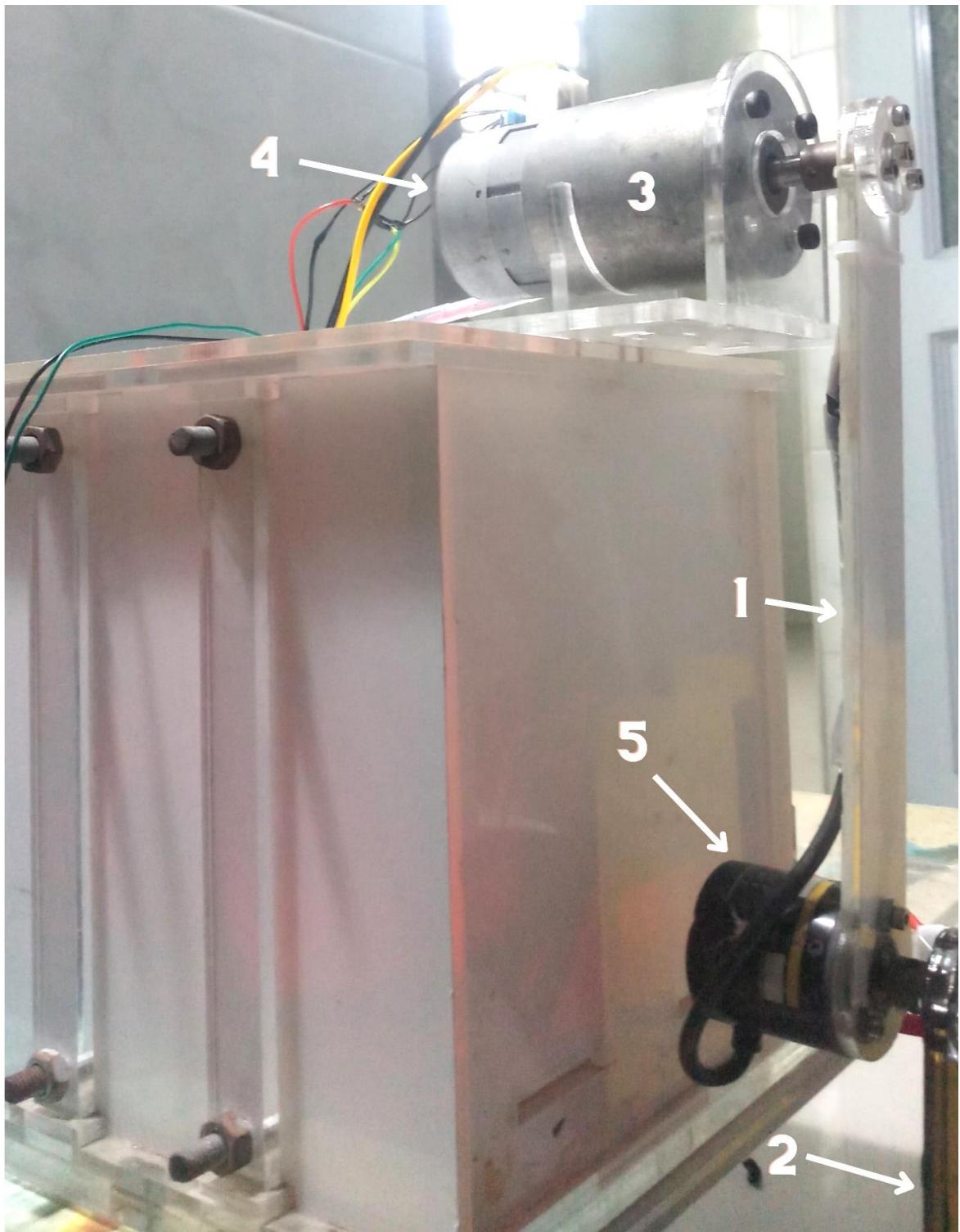
- 1 – Thanh 1
- 2 – Thanh 2
- 3 – Động cơ Nisca NF5475E
- 4 – Encoder 1
- 5 – Encoder 2
- 6 – Cầu H IR2184
- 7 – STM32F407
- 8 – Bộ nguồn 12 VDC



Hình 5.8. Hình chụp phần cứng khi đang ở trạng thái nghỉ

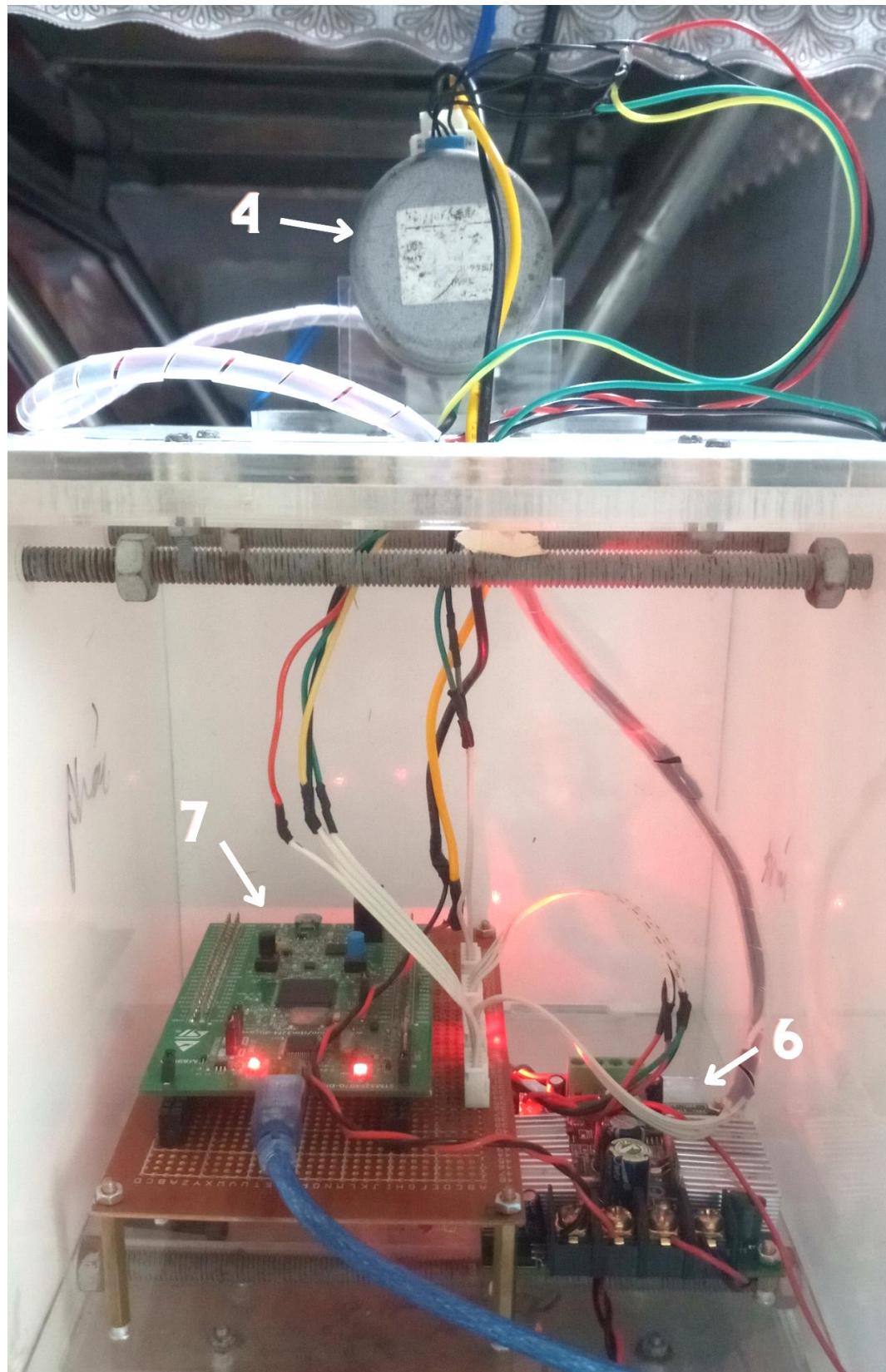


Hình 5.9. Hình chụp phần cứng khi đang cân bằng ở vị trí TOP



Hình 5.10. Hình chụp góc nghiên của hệ Pendubot

CHƯƠNG 5. Thực nghiệm



Hình 5.11. Hình chụp hộp điện của hệ Pendubot

5.2 Thư viện và Phần mềm

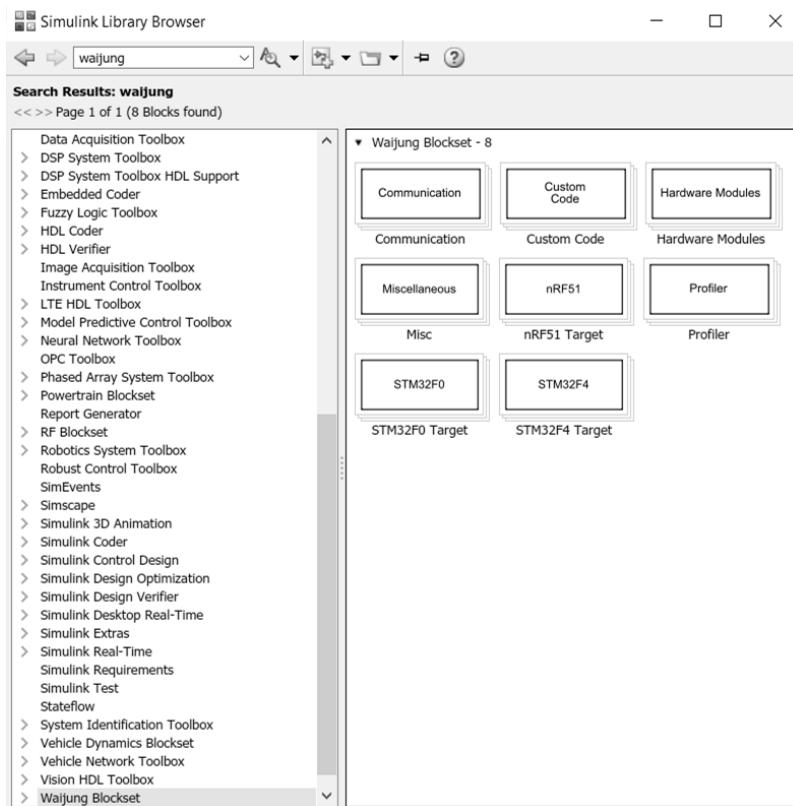
5.2.1 Waijung Blockset

Waijung hoặc “ໄວຈັງ”, tiếng lóng của Thái Lan, nghĩa là “nhanh”, là thư viện cho công cụ MATLAB/Simulink gồm các khối lệnh được xây dựng trên nền tảng C và được phát triển bởi công ty Aimagin – công ty chuyên về nhúng và sản xuất phần mềm của Thái Lan. Mục đích của thư viện này là cho phép cấu hình cho vi điều khiển và thiết bị ngoại vi.

Thư viện Waijung có 2 phiên bản: Waijung 1 được thiết kế riêng cho vi điều khiển dòng STM32F4, Waijung 2 để phục vụ cho ứng dụng IoT qua vi điều khiển ESP32.

Các phiên bản Waijung đều được đăng trên trang web chính thức của Công ty Aimagin. Việc tìm kiếm và cài đặt dễ dàng, hỗ trợ rất nhiều trong quá trình dạy học tại các cơ sở giáo dục kỹ thuật.

Waijung 1 chỉ có thể lập trình trên MATLAB từ bản 2009a đến 2018a. Waijung 1 chứa các khối lập trình phục vụ đầy đủ các ứng dụng của vi điều khiển, bao gồm Bộ định thời (Timer)/Bộ đếm (Counter), điều khiển động cơ bước (Stepper motor) và động cơ servo (Servo motor), giao tiếp I2C, SPI, UART....



Hình 5.12. Thư viện Waijung Blockset trong thư viện chính của MATLAB/SIMULINK

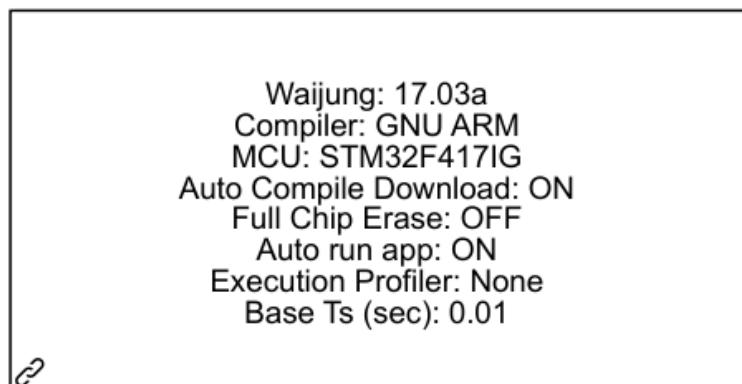
CHƯƠNG 5. Thực nghiệm

Trong đề tài này, nhóm sinh viên sẽ sử dụng các khôi lệnh của thư viện Waijung, bao gồm:

- Target Setup
- UART Setup
- UART Tx
- Encoder Read
- Basic PWM
- Digital Output

Việc kết hợp khôi lệnh của thư viện Waijung và các khôi lệnh khác của MATLAB/Simulink sẽ tạo ra chương trình mô phỏng hoàn chỉnh, dễ dàng hơn rất nhiều so với việc viết các dòng code C như truyền thống.

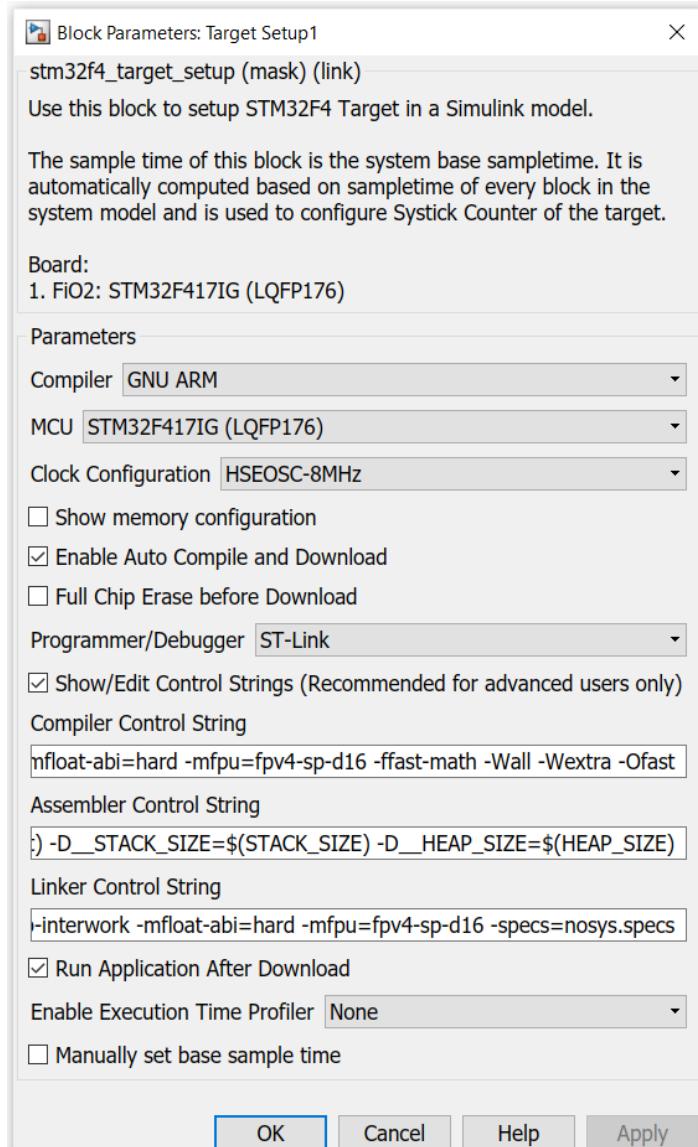
Khôi Target setup: khôi này được đưa vào mỗi chương trình Simulink để thiết lập các thông tin cài đặt cho vi điều khiển



Hình 5.13. Khôi Target Setup

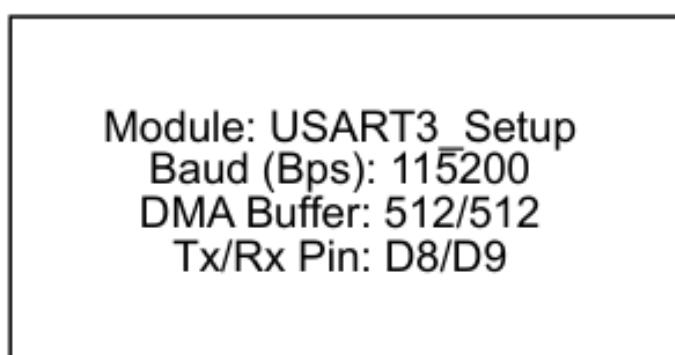
- Compiler: chọn dòng vi điều khiển GNU ARM, MDK ARM hoặc EW ARM
- MCU: chọn dòng vi xử lý, bao gồm các dòng của STM32F4
- Clock Configuration: thiết lập xung nhịp clock
- Programmer/Debugger: chọn trình gỡ rối cho vi điều khiển

CHƯƠNG 5. Thực nghiệm



Hình 5.14. Cài đặt cho khối Target Setup

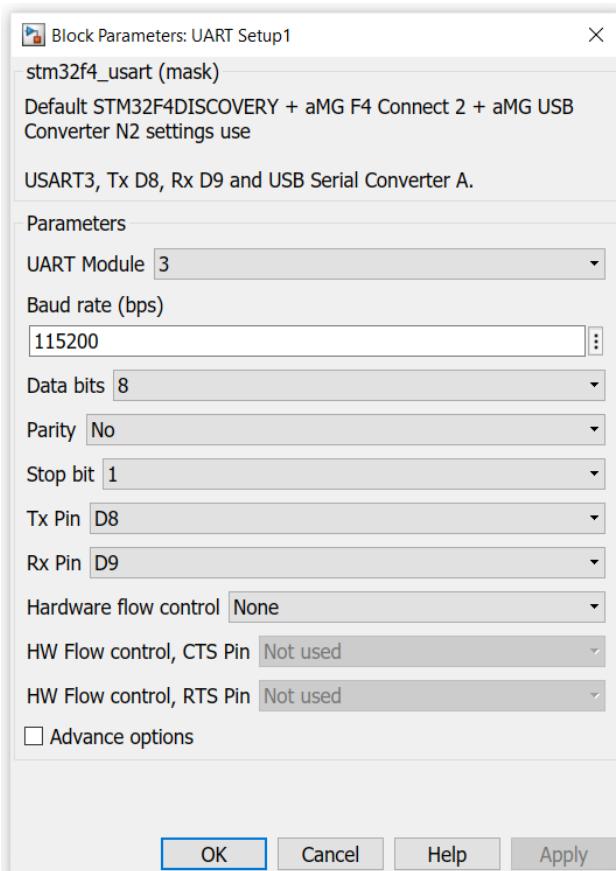
Khối UART Setup: cài đặt UART cho vi điều khiển



Hình 5.15. Khối UART Setup

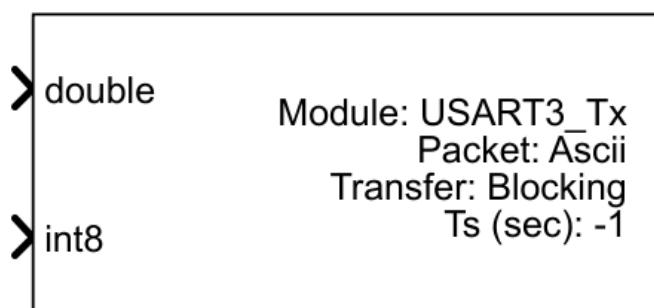
CHƯƠNG 5. Thực nghiệm

- UART Module: chọn module UART trong trường hợp sử dụng nhiều UART trong 1 chương trình
- Baud rate: chọn tốc độ truyền/nhận dữ liệu (số bit/giây)
- Tx Pin: chọn chân truyền dữ liệu đến thiết bị khác
- Rx Pin: chọn chân nhận dữ liệu về từ thiết bị khác



Hình 5.16. Cài đặt cho khối UART Setup

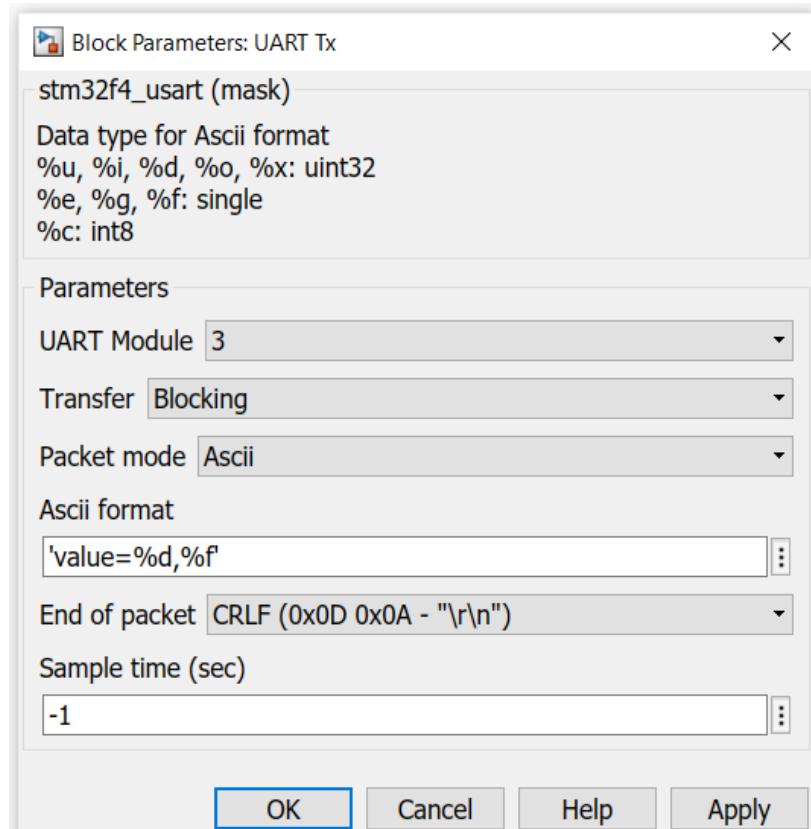
UART Rx: cài đặt cho quá trình nhận dữ liệu khi vi điều khiển giao tiếp với các thiết bị thông qua giao thức UART



Hình 5.17. Khối UART Tx

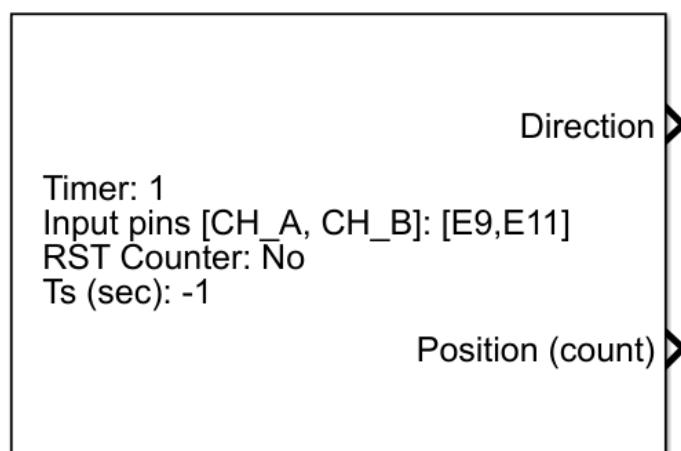
CHƯƠNG 5. Thực nghiệm

- UART Module: chọn module UART, phải chọn module giống ở phần này của khối UART Setup
- Transfer: chọn phương thức truyền
- Packet mode: chọn dạng dữ liệu truyền đi, ASCII hoặc Binary (mã nhị phân)



Hình 5.18. Cài đặt cho khối UART Tx

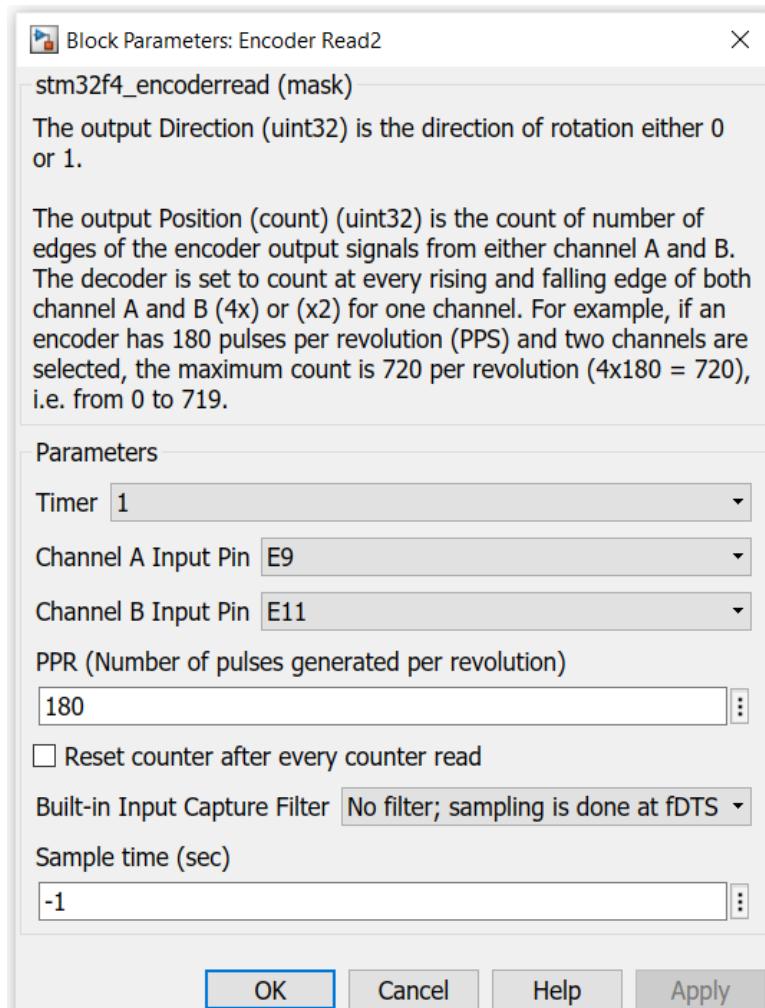
Encoder read: đọc đầu ra của bộ encoder



Hình 5.19. Khối Encoder Read

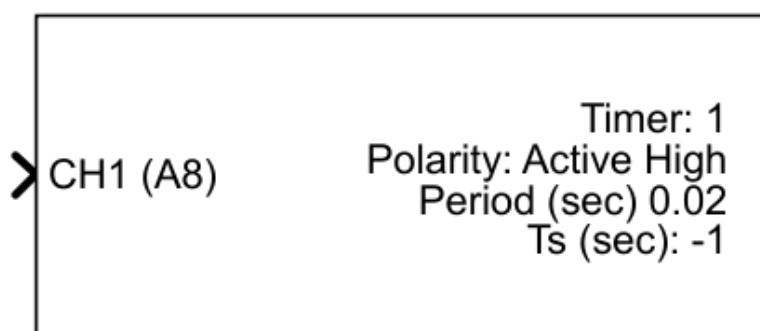
CHƯƠNG 5. Thực nghiệm

- Timer: chọn timer hoạt động cho encoder
- Channel A/B Input Pin: chọn 2 chân đọc tín hiệu A/B của encoder
- PPR: nhập độ phân giải của encoder (xung/vòng)



Hình 5.20. Cài đặt cho khối Encoder Read

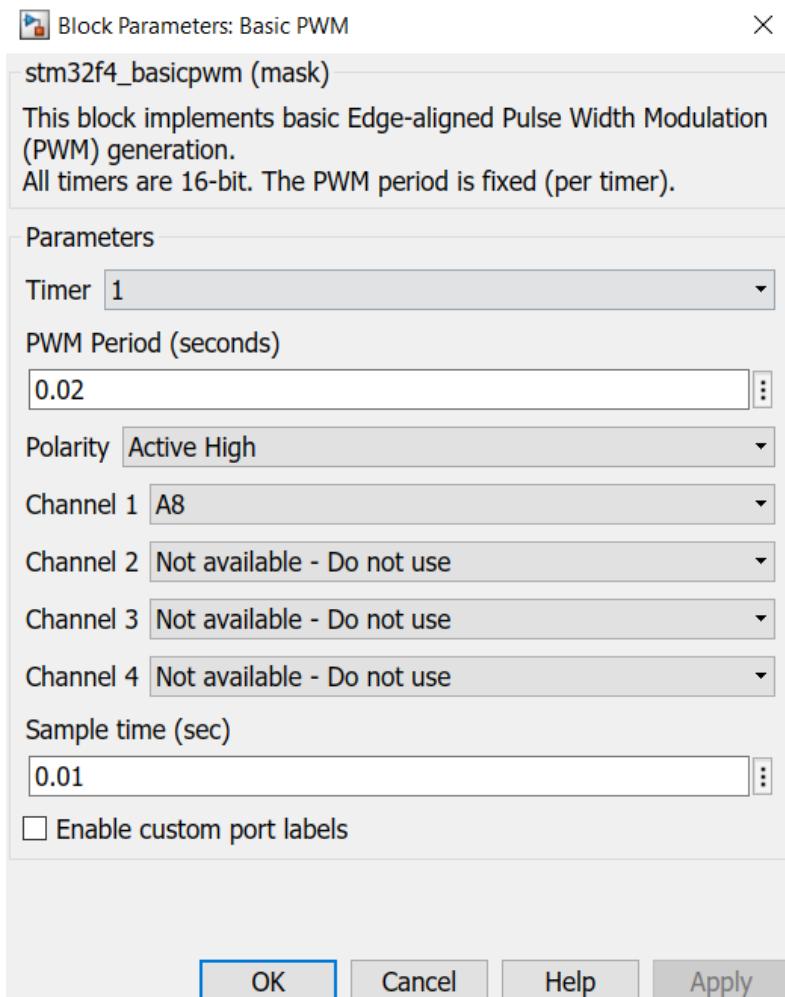
Basic PWM: cài đặt các thông số băm xung cho bộ xử lý



Hình 5.21. Khối Basic PWM

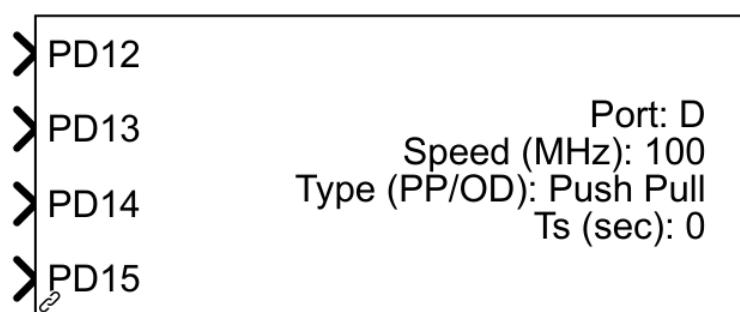
CHƯƠNG 5. Thực nghiệm

- Timer: chọn timer hoạt động cho khối PWM
- PWM Period: thời gian của một chu kì băm xung
- Channel 1 → 4: chọn chân nhận tín hiệu PWM từ vi điều khiển



Hình 5.22. Cài đặt cho khối Basic PWM

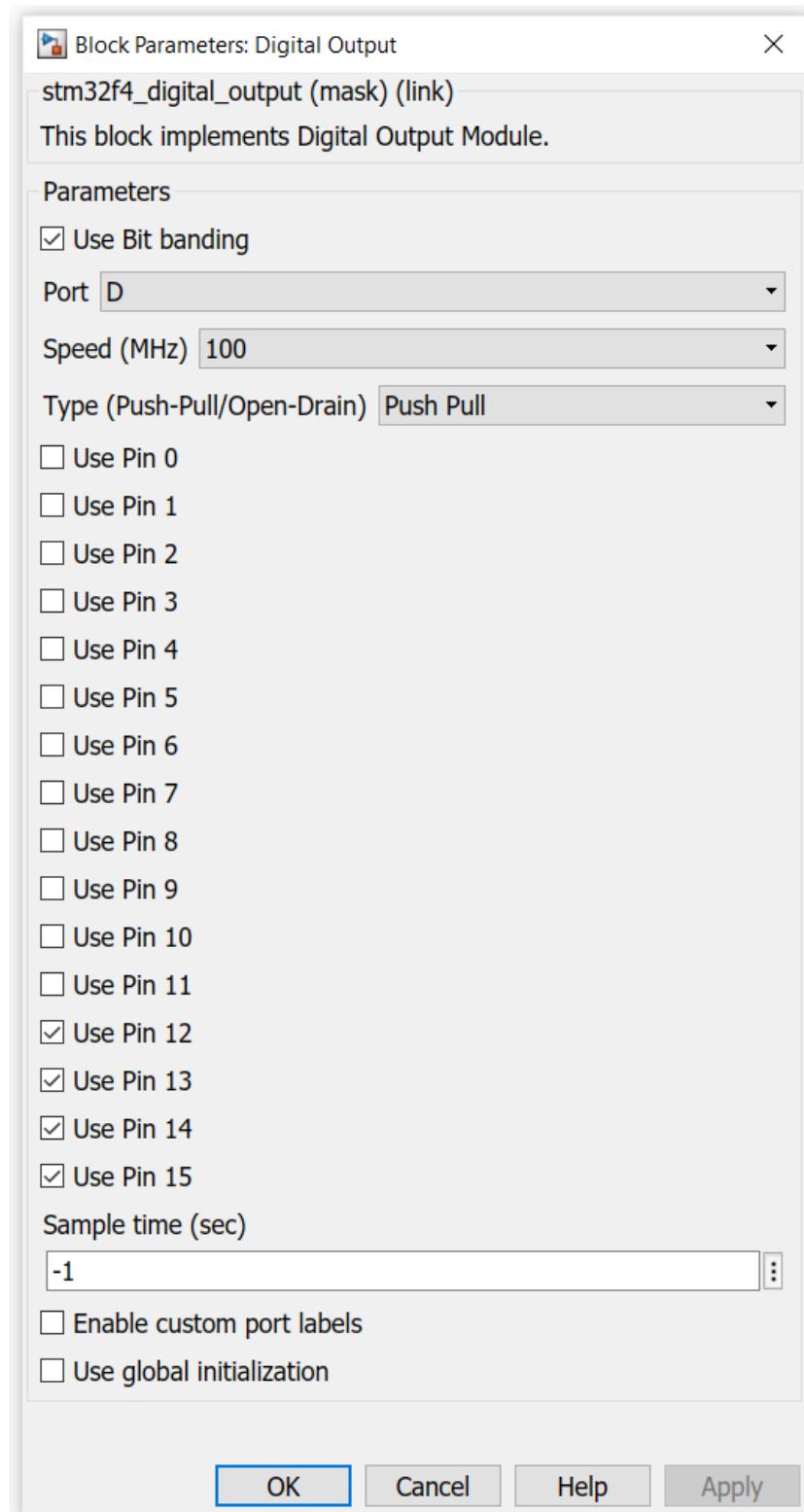
Digital Output: khối tạo logic 0 1 cho chân ngõ ra



Hình 5.23. Khối Digital Output

CHƯƠNG 5. Thực nghiệm

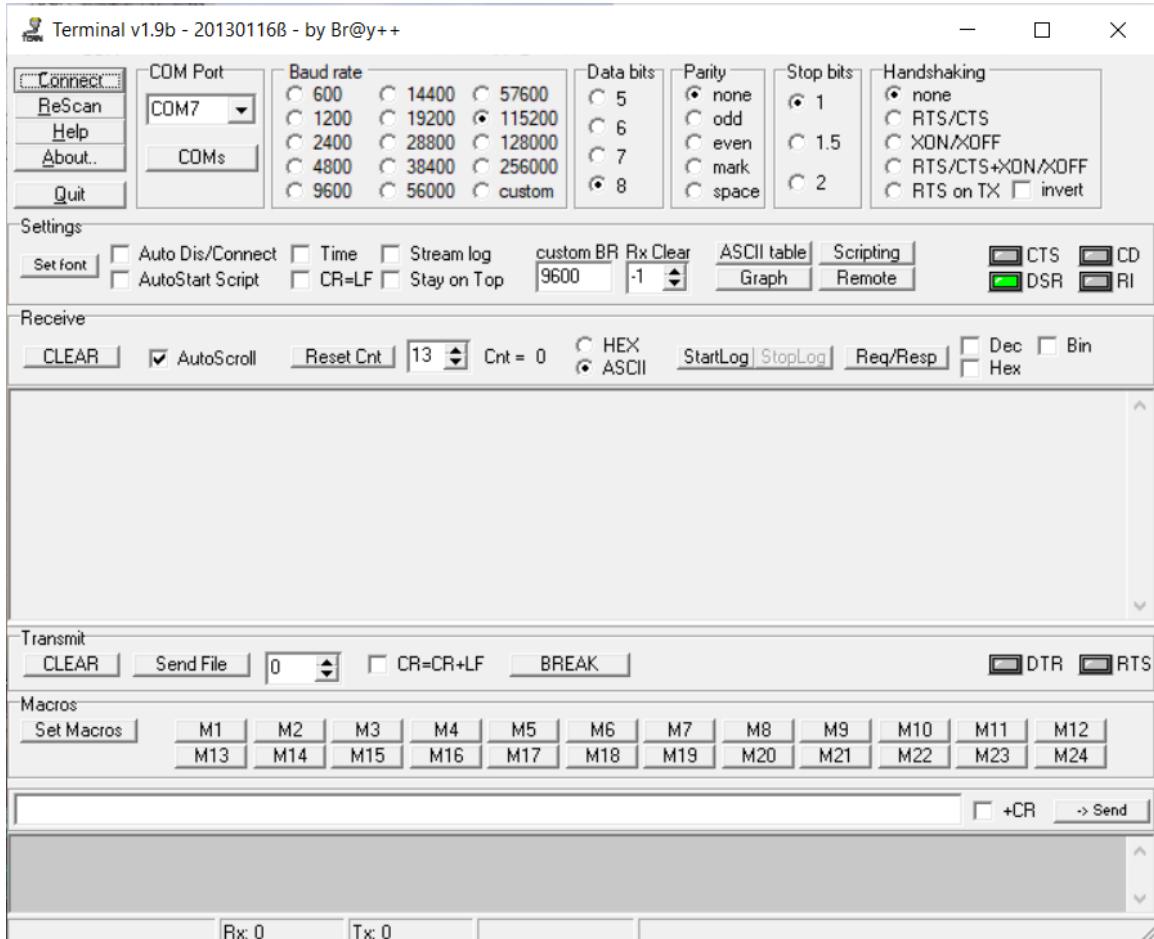
Chọn port chứa chân cần truyền tín hiệu. Đánh dấu vào ô cần đưa tín hiệu ra từ vi điều khiển.



Hình 5.24. Cài đặt cho khối Digital Output

5.2.2 Hyper Terminal

Hyper Terminal là phần mềm thông dụng trong việc giao tiếp giữa máy tính và vi điều khiển qua cổng COM.



Hình 5.25. Giao diện chính của phần mềm Hyper Terminal

Các bước cài đặt cho phần mềm Hyper Terminal:

- *Bước 1:* Chọn cổng kết nối với module UART ở ô *COM Port*
- *Bước 2:* Chọn tốc độ truyền/nhận dữ liệu ở ô *Baud rate*. Ở bước này, phải chọn giống với giá trị đã cài đặt tại ô Baud rate ở khôi UART Setup trong công cụ MATLAB/Simulink
- *Bước 3:* Chọn số bit dữ liệu ở ô *Data bits*

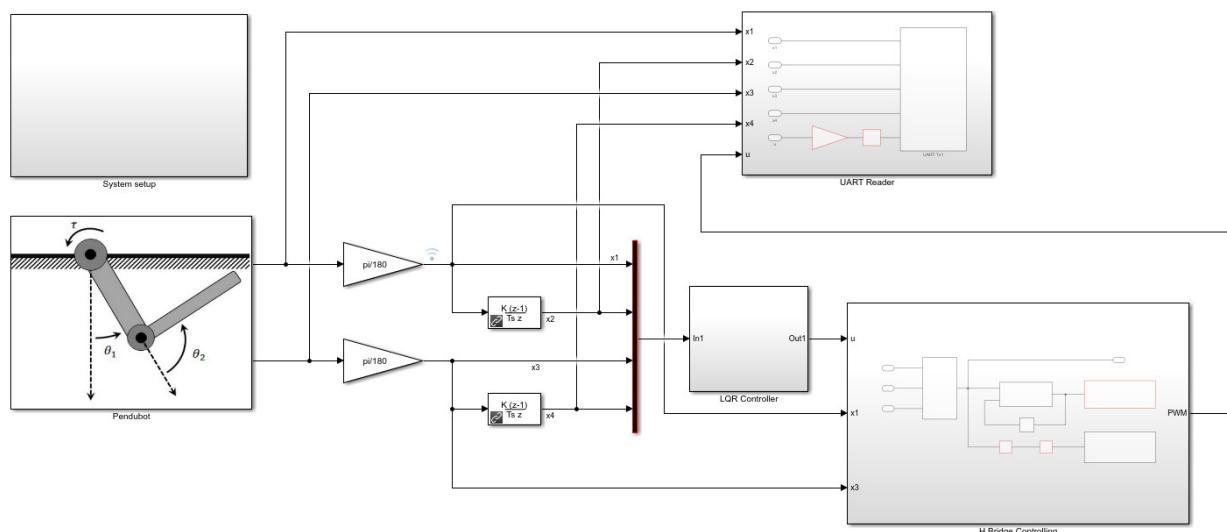
Sau khi đã hoàn thành xong các bước cài đặt, chọn Connect để tiến hành giao tiếp máy tính và vi điều khiển.

5.2.3 Thiết kế chương trình điều khiển

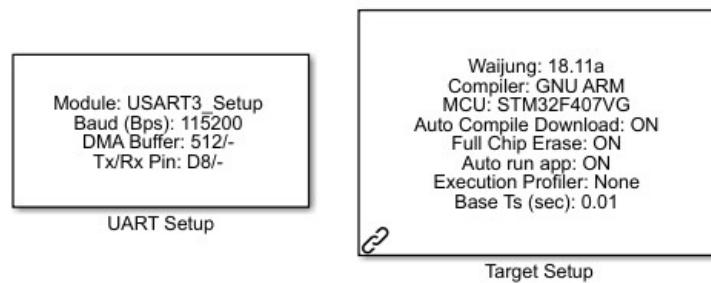
Chương trình điều khiển được viết trên công cụ MATLAB/Simulink và nhúng vào board STM32F407 qua thư viện Waijung.

Chương trình tổng quát được trình bày ở *Hình 5.26*, bao gồm các khối System setup, Pendubot, LQR controller, H Brigde controlling và UART Reader.

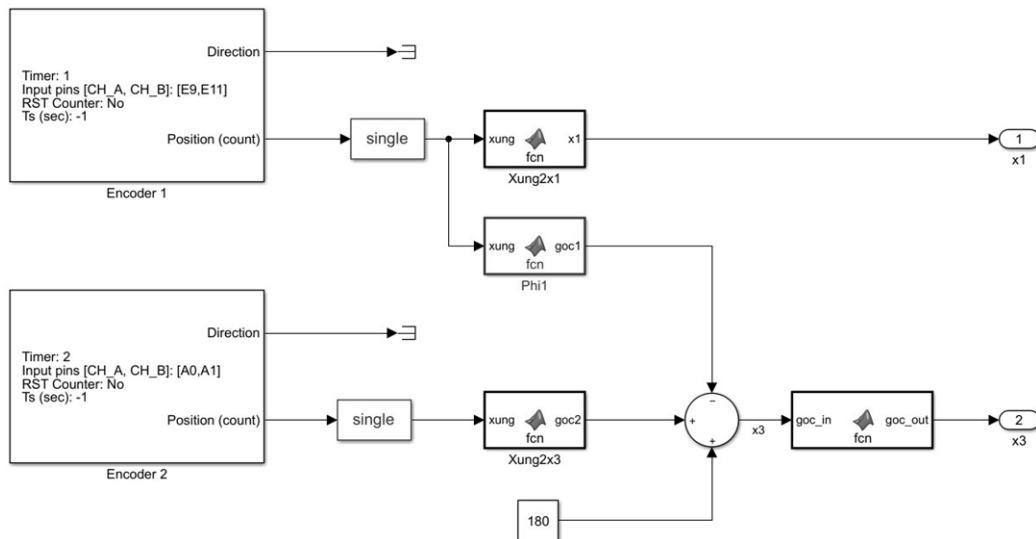
- Khối System Setup (*Hình 5.27*) chứa các thông tin cài đặt cho vi điều khiển: khai báo vi xử lý, khai báo dòng vi điều khiển, chọn thời gian lấy mẫu, chọn chân đọc UART ...
- Khối Pendubot (*Hình 5.28*) chứa các lệnh đọc xung từ encoder thanh 1 và encoder thanh 2 và các phép biến đổi để góc đọc được theo như giả thiết
- Khối H Bridge Controlling (*Hình 5.31*) chứa các lệnh điều khiển cho cầu H, bao gồm điều khiển đảo chiều động cơ qua chân DIR và điều xung bằng chân PWM
- Khối UART reader (*Hình 5.34*) chứa cài đặt đọc UART và thông tin hiển thị ra phần mềm hỗ trợ đọc UART Hyper Terminal



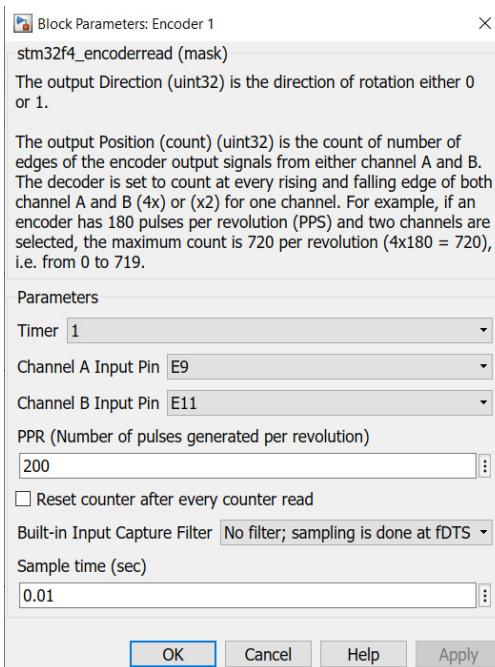
Hình 5.26. Chương trình điều khiển Pendubot thực tế



Hình 5.27. Nội dung khói Subsystem “System setup”

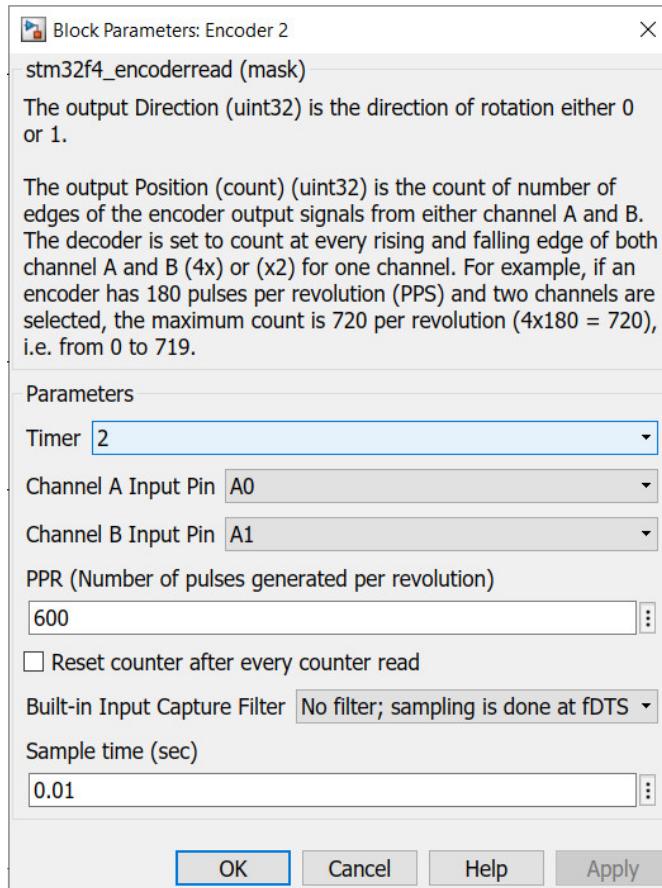


Hình 5.28. Nội dung khói Subsystem “Pendubot”

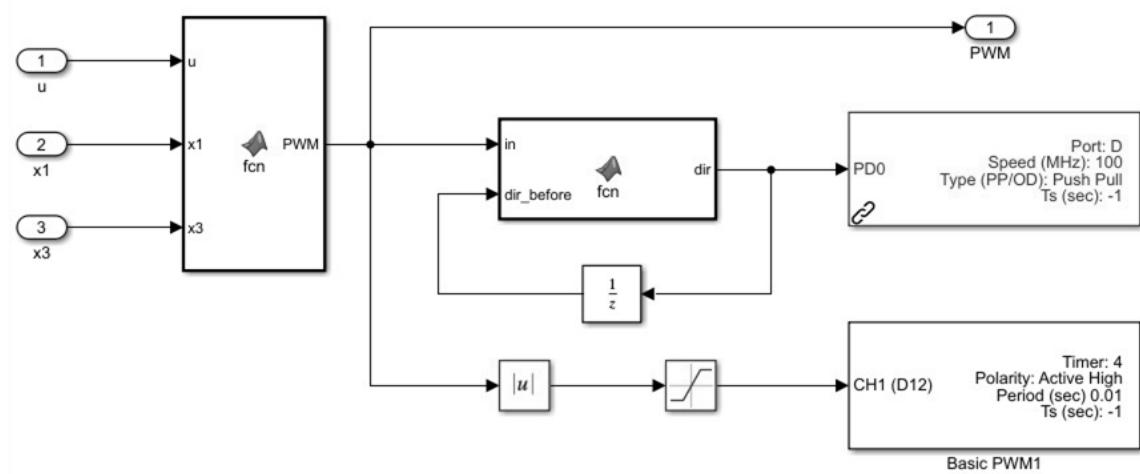


Hình 5.29. Cài đặt cho khói Encoder 1

CHƯƠNG 5. Thực nghiệm

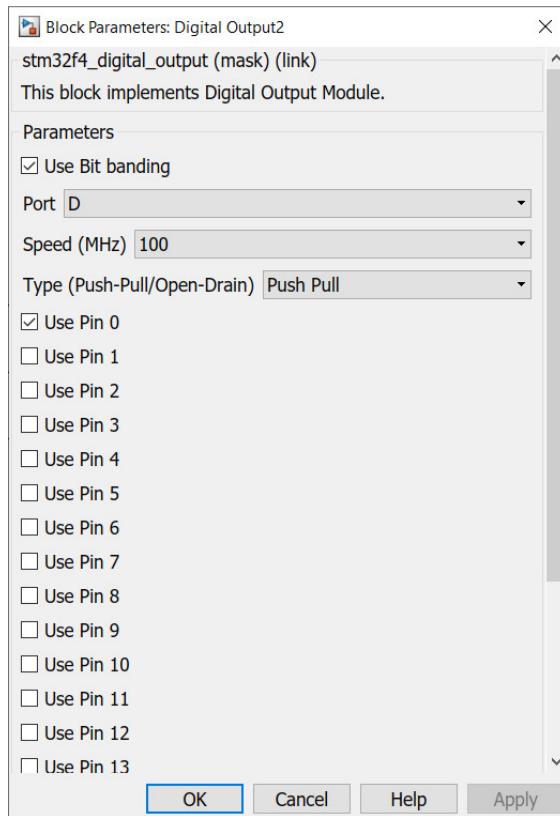


Hình 5.30. Cài đặt cho khối Encoder 2

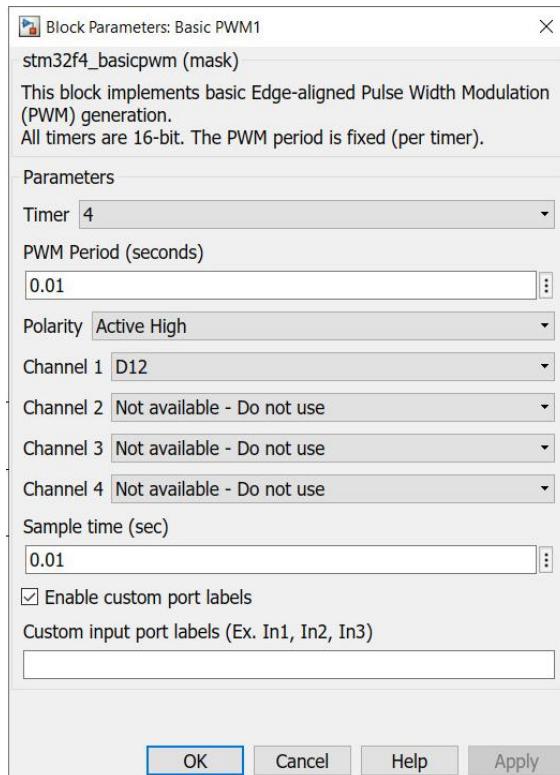


Hình 5.31. Nội dung khối Subsystem “H Brigde Controlling”

CHƯƠNG 5. Thực nghiệm

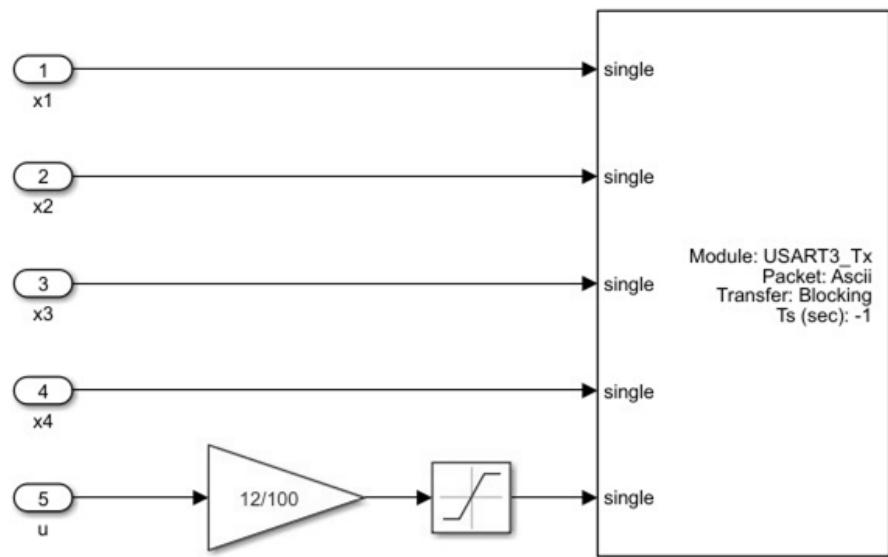


Hình 5.32. Cài đặt đảo chiều động cơ

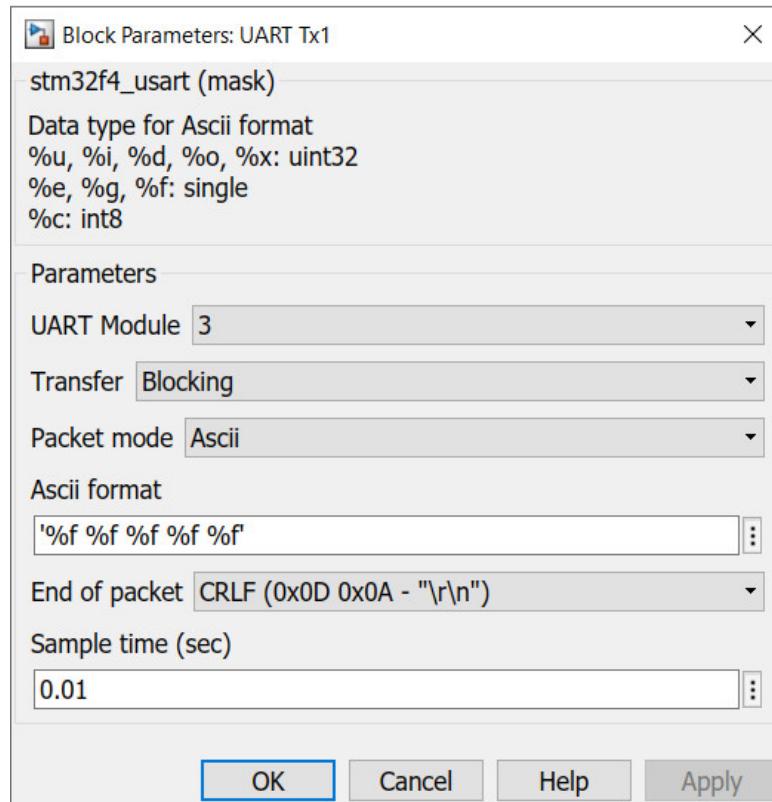


Hình 5.33. Cài đặt điều xung cho động cơ

CHƯƠNG 5. Thực nghiệm



Hình 5.34. Nội dung khối Subsystem “UART reader”



Hình 5.35. Cài đặt đọc UART

5.3 KẾT QUẢ THỰC NGHIỆM

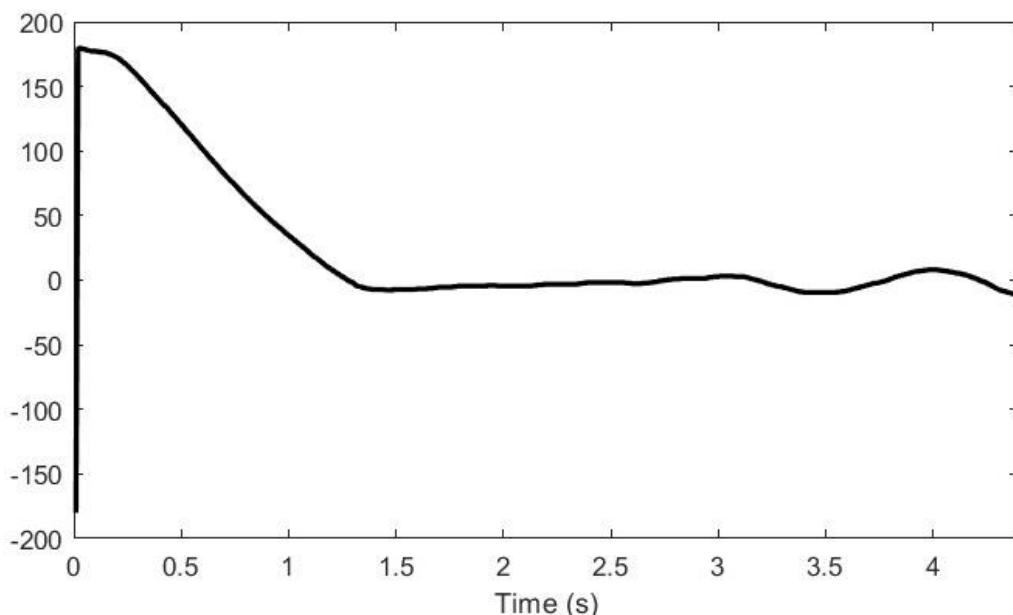
Đáp ứng ngõ ra của Bộ điều khiển Fuzzy không ổn định và thời gian đáp ứng dài (dựa trên kết quả mô phỏng đã trình bày ở mục 4.3.3), nhóm sinh viên bỏ qua điều khiển thực tế đối với phương pháp điều khiển này. Song song đó, nhóm sinh viên đã thực hiện điều khiển hệ Pendubot bằng Bộ điều khiển Tuyến tính hoá hồi tiếp, nhưng không thành công nên không đưa vào quyền báo cáo tóm tắt. Chính vì vậy, ở mục này chỉ trình bày kết quả thực nghiệm đối với Bộ điều khiển LQR.

Mục tiêu của đề tài này là cân bằng hệ Pendubot ở vị trí TOP, do đó điều kiện ban đầu là phải đưa cả 2 thanh về vị trí 0 (theo chiều hướng lên ngược chiều trọng trường). Theo *Hình 5.36* và *Hình 5.37*, trong 1.5 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (2 thanh xuôi theo chiều trọng trường) lên vị trí lân cận 0.

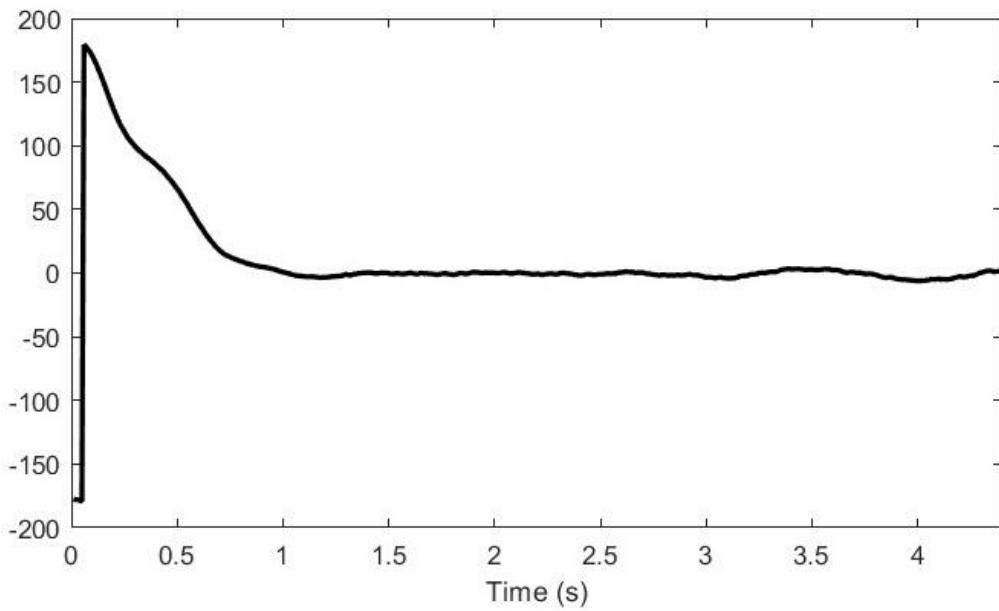
Giá trị K thực nghiệm giống như ở mô phỏng:

$$K = 10^3 \times [-0.0313 \quad 0.1970 \quad 3.5168 \quad 0.5409]$$

Bộ thông số tìm được hoạt động ổn định trong mô phỏng, tuy nhiên khi thực nghiệm, bộ thông số này chỉ giữ thăng bằng hệ trong khoảng 2.5 giây, sai số khi ổn định của thanh 1 là 9.45° , thanh 2 là 3.95° .



Hình 5.36. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR



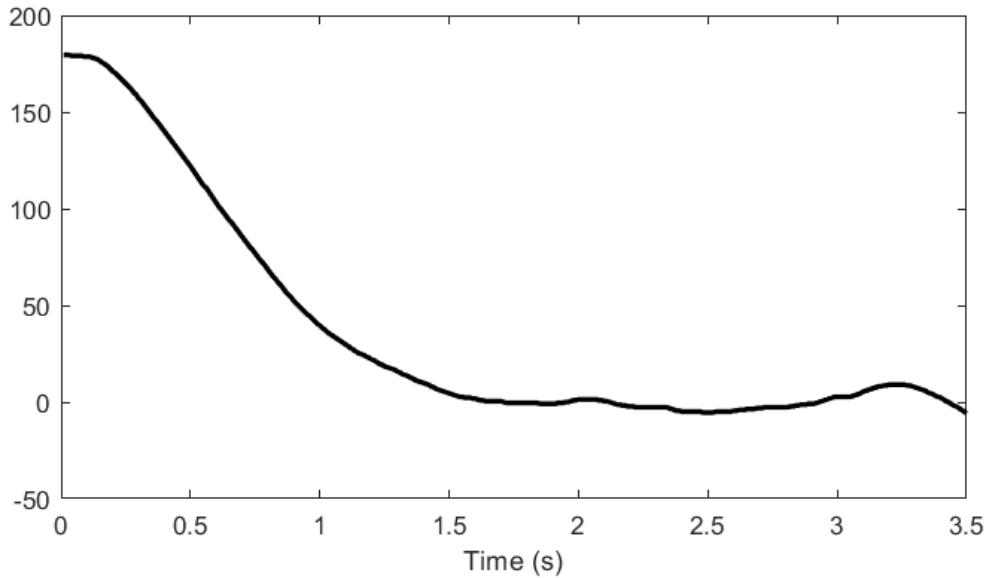
Hình 5.37. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR

Nhận thấy thời gian ổn định của mô hình quá ngắn và đáp ứng chưa thật sự bám vị trí cân bằng, nhóm sinh viên xem cấu trúc điều khiển của Bộ điều khiển LQR tương tự như Bộ điều khiển PID, tinh chỉnh trực tiếp trên các giá trị của ma trận K .

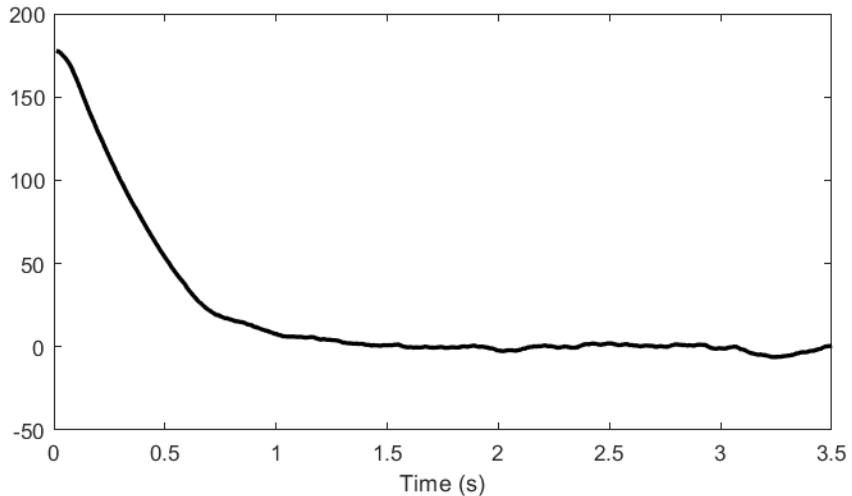
Tăng K_1

Tăng giá trị K_1 của ma trận điều khiển. Ma trận mới trở thành:

$$K = 10^3 \times [0.0313 \quad 0.1970 \quad 3.5168 \quad 0.5409]$$



Hình 5.38. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã tăng K_1



Hình 5.39. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã tăng K₁

Kết quả nhận được như *Hình 5.38* và *Hình 5.39*. Trong 1.5 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (vị trí xuôi theo chiều trọng trường) lên vị trí cân bằng.

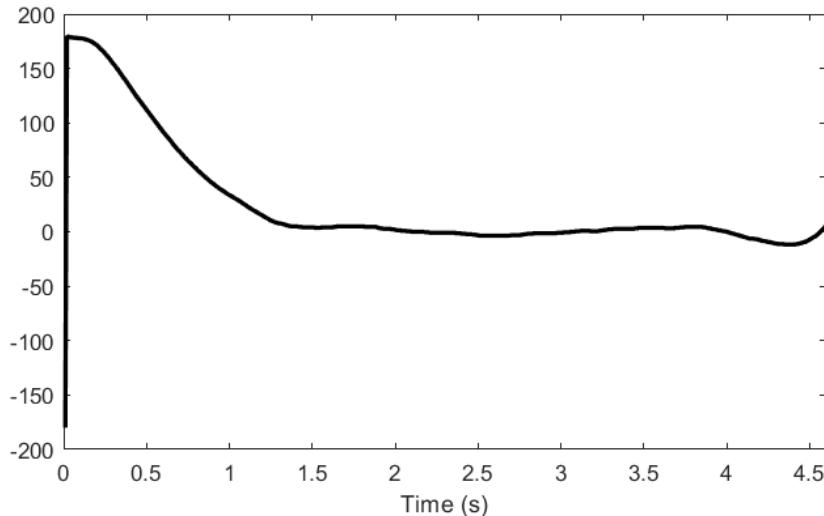
Khi tăng K₁ ở giá trị -0.313 thành 0.313, hệ Pendubot giữ cân bằng ở vị trí TOP trong 2 giây, sai số khi ổn định tối đa của thanh 1 là 9°, thanh 2 là 6.12°.

Giảm K₁

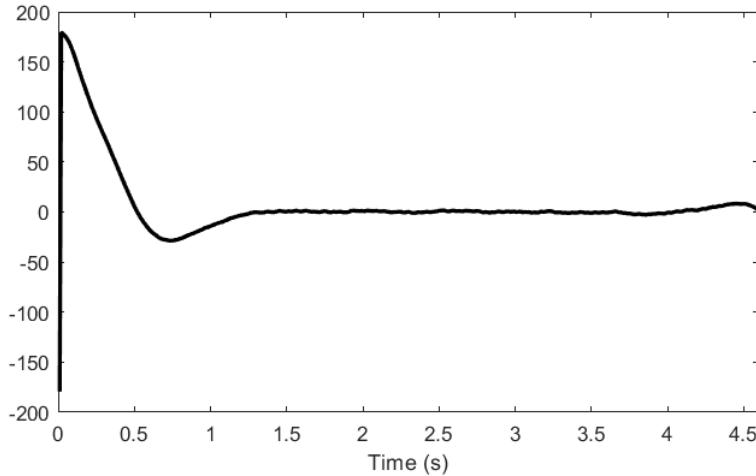
Giảm giá trị K₁ của ma trận điều khiển. Ma trận mới trở thành:

$$K = 10^3 \times [-0.0913 \quad 0.1970 \quad 3.5168 \quad 0.5409]$$

Kết quả nhận được như *Hình 5.40* và *Hình 5.41*. Trong 1.5 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (vị trí xuôi theo chiều trọng trường) lên vị trí cân bằng.



Hình 5.40. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã giảm K₁



Hình 5.41. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã giảm K_1

Khi giảm K_1 ở giá trị -0.313 thành -0.913 , hệ Pendubot giữ cân bằng ở vị trí TOP trong 3 giây, sai số khi ổn định tối đa của thanh 1 là 4.95° , thanh 2 là 1.62° .

Nhân xét:

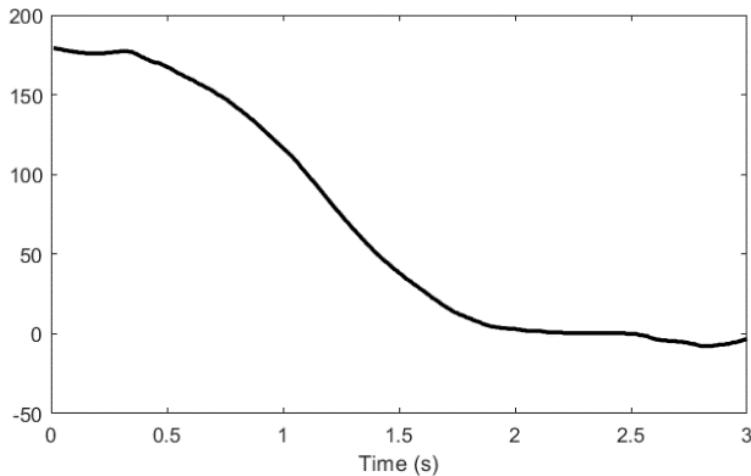
Khi tinh chỉnh giá trị K_1 thì hệ thống ổn định hơn với giá trị K_1 giảm, độ sai số khi ổn định thấp hơn, thời gian cân bằng hệ thống có tăng nhưng không đáng kể

Tăng K_2

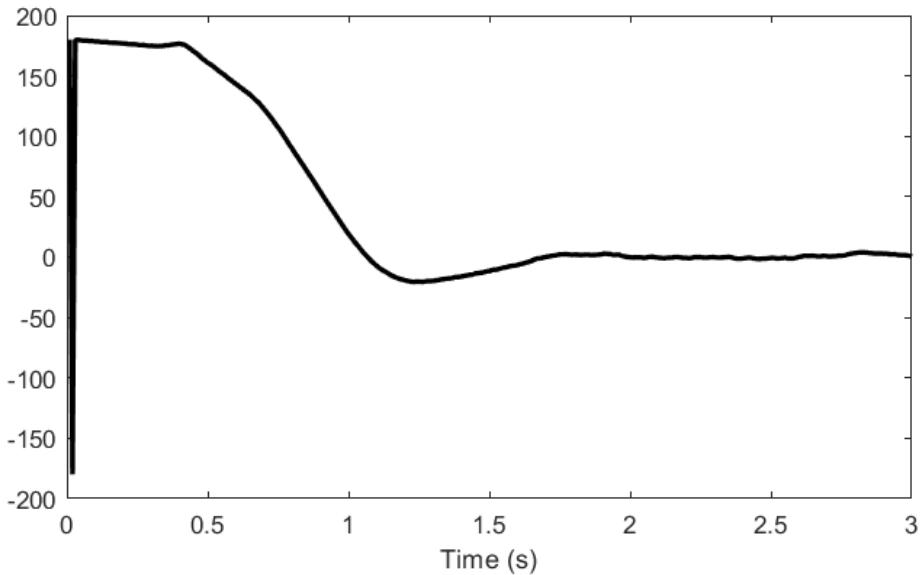
Tăng giá trị K_2 của ma trận điều khiển. Ma trận mới trở thành:

$$K = 10^3 \times [-0.0313 \quad 0.3970 \quad 3.5168 \quad 0.5409]$$

Kết quả nhận được như *Hình 5.42* và *Hình 5.43*. Trong 1.6 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (vị trí xuôi theo chiều trọng trường) lên vị trí cân bằng.



Hình 5.42. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã tăng K_2



Hình 5.43. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã tăng K₂

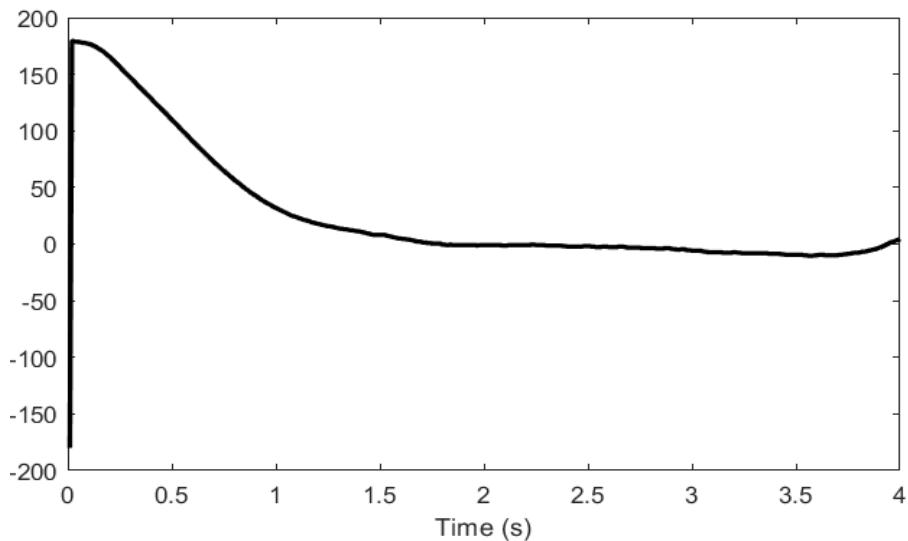
Khi tăng K₂ ở giá trị 0.1970 thành 0.3970, hệ Pendubot giữ cân bằng ở vị trí TOP trong 1 giây, sai số khi ổn định tối đa của thanh 1 là 7.65°, thanh 2 là 3.86°.

Giảm K₂

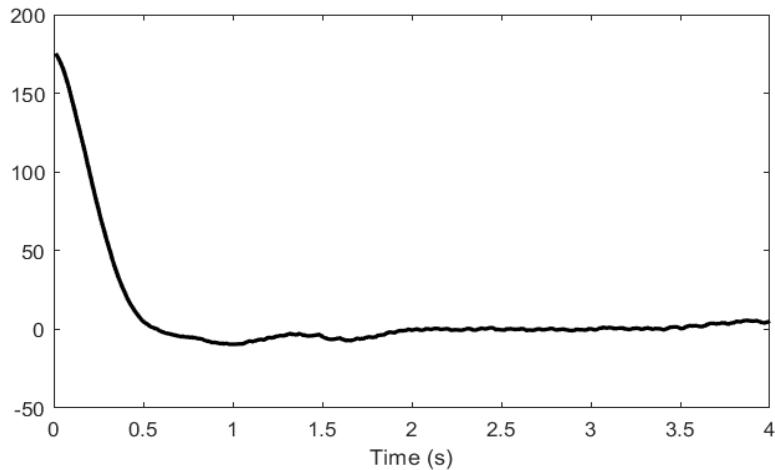
Giảm giá trị K₂ của ma trận điều khiển. Ma trận mới trở thành:

$$K = 10^3 \times [-0.0313 \quad -0.1970 \quad 3.5168 \quad 0.5409]$$

Kết quả nhận được như *Hình 5.44* và *Hình 5.45*. Trong 1.5 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (vị trí xuôi theo chiều trọng trường) lên vị trí cân bằng.



Hình 5.44. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã giảm K₂



Hình 5.45. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã giảm K_2

Khi giảm K_2 ở giá trị 0.1970 thành -0.1970 , hệ Pendubot giữ cân bằng ở vị trí TOP trong 1 giây, sai số khi ổn định tối đa của thanh 1 là 9.9° , thanh 2 là 4.6° .

Nhân xét

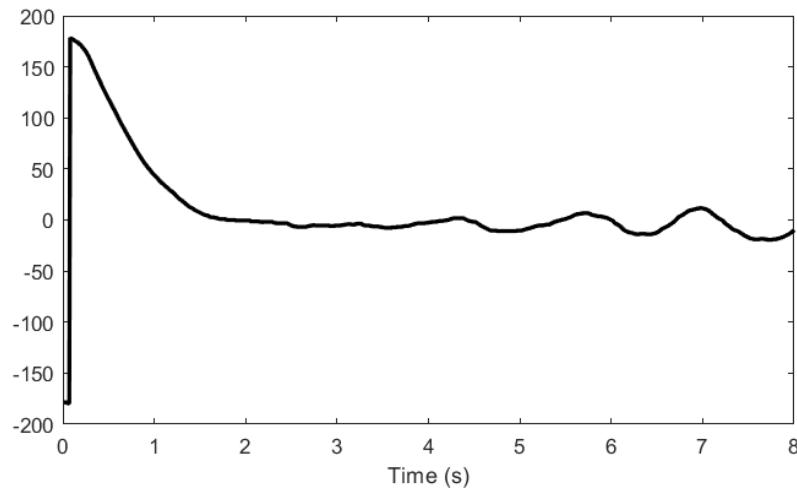
Khi tinh chỉnh giá trị K_2 thì hệ thống không ổn định hơn, đồng nghĩa với thời gian điều khiển hệ thống vẫn chưa có tiến triển.

Tăng K_3

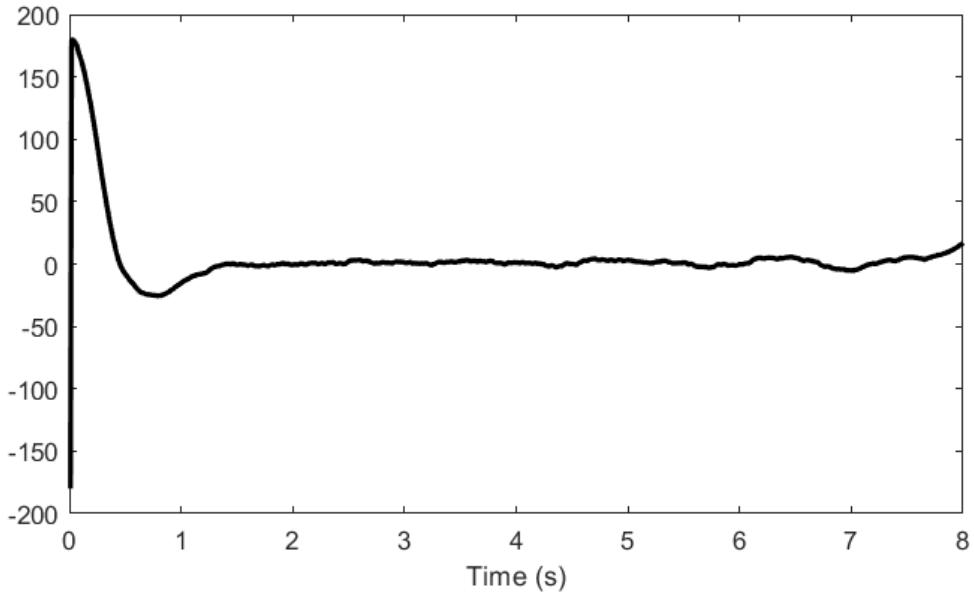
Tăng giá trị K_3 của ma trận điều khiển. Ma trận mới trở thành:

$$K = 10^3 \times [-0.0313 \quad 0.1970 \quad 4.5168 \quad 0.5409]$$

Kết quả nhận được như *Hình 5.46* và *Hình 5.47*. Trong 2 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (vị trí xuôi theo chiều trọng trường) lên vị trí cân bằng.



Hình 5.46. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã tăng K_3



Hình 5.47. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã tăng K_3

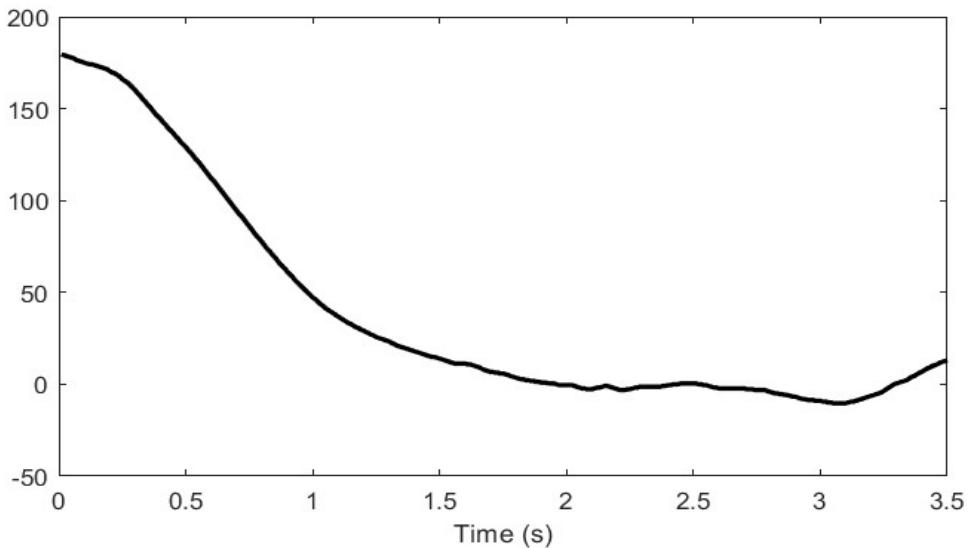
Khi tăng K_3 ở giá trị 3.5168 thành 4.5168, hệ Pendubot giữ cân bằng ở vị trí TOP trong 6.5 giây, sai số khi ổn định tối đa của thanh 1 là 11.75° , thanh 2 là 5.3° .

Giảm K_3

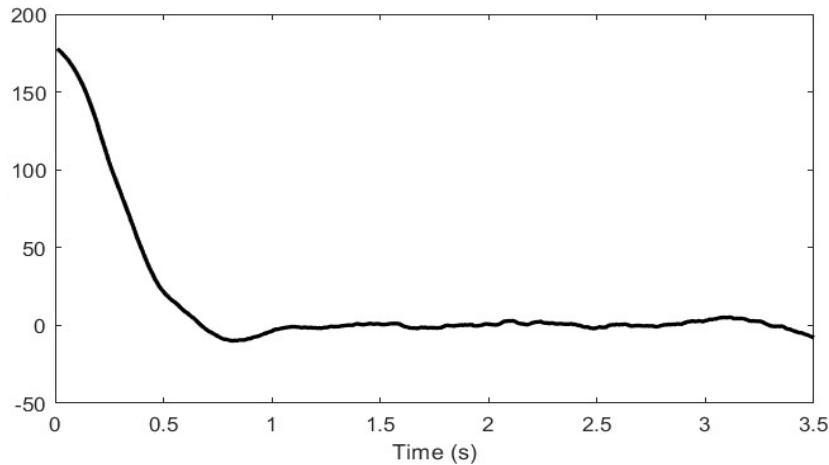
Giảm giá trị K_3 của ma trận điều khiển. Ma trận mới trở thành:

$$K = 10^3 \times [-0.0313 \quad 0.1970 \quad 2.5168 \quad 0.5409]$$

Kết quả nhận được như *Hình 5.48* và *Hình 5.49*. Trong 1 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (vị trí xuôi theo chiều trọng trường) lên vị trí cân bằng.



Hình 5.48. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã giảm K_3



Hình 5.49. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã giảm K_3

Khi giảm K_3 ở giá trị 3.5168 thành 2.5168, hệ Pendubot giữ cân bằng ở vị trí TOP trong 2.5 giây, sai số khi ổn định tối đa của thanh 1 là 10.38° , thanh 2 là 5.13° .

Nhận xét

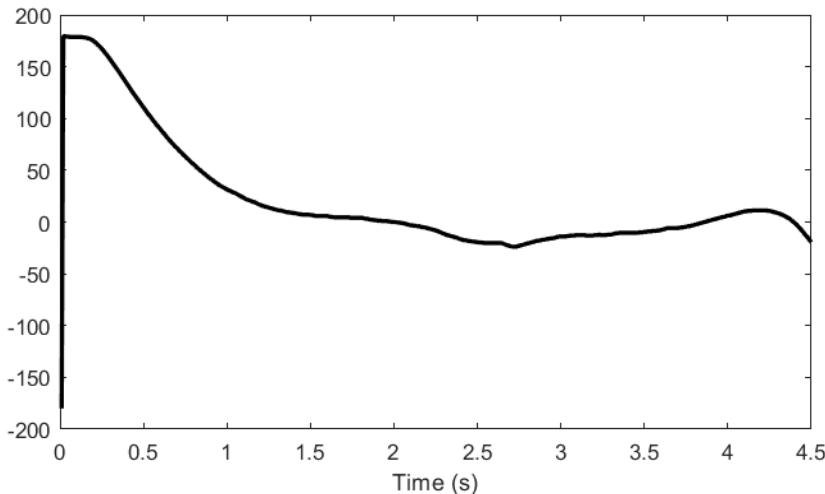
Khi tinh chỉnh giá trị K_3 thì hệ thống ổn định hơn với giá trị K_3 giảm, độ sai số khi ổn định cao hơn, bù lại thời gian cân bằng hệ thống thấp.

Tăng K_4

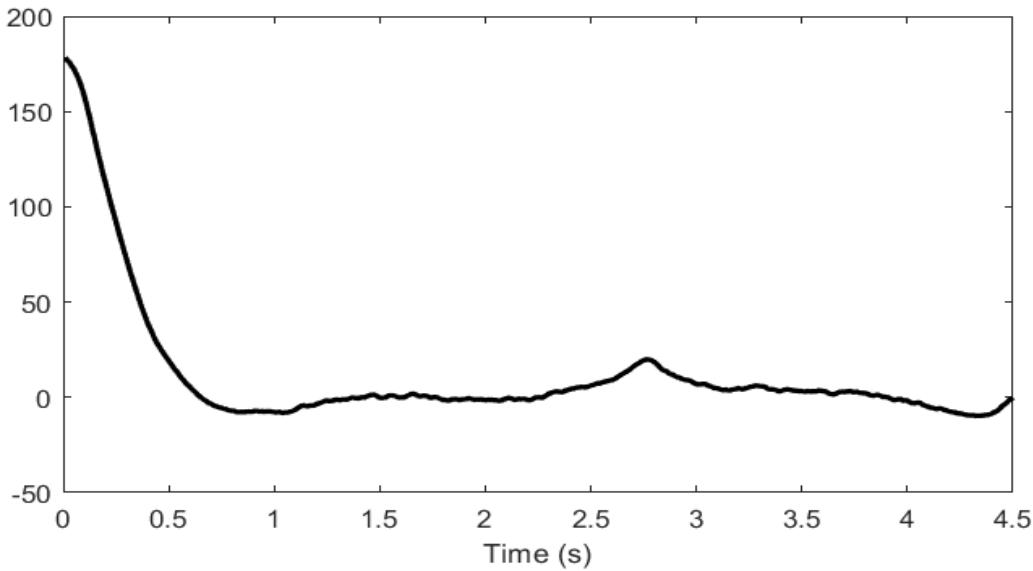
Tăng giá trị K_4 của ma trận điều khiển. Ma trận mới trở thành:

$$K = 10^3 \times [-0.0313 \quad 0.1970 \quad 3.5168 \quad 0.6409]$$

Kết quả nhận được như *Hình 5.50* và *Hình 5.51*. Trong 1 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (vị trí xuôi theo chiều trọng trường) lên vị trí cân bằng.



Hình 5.50. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã tăng K_4



Hình 5.51. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã tăng K_4

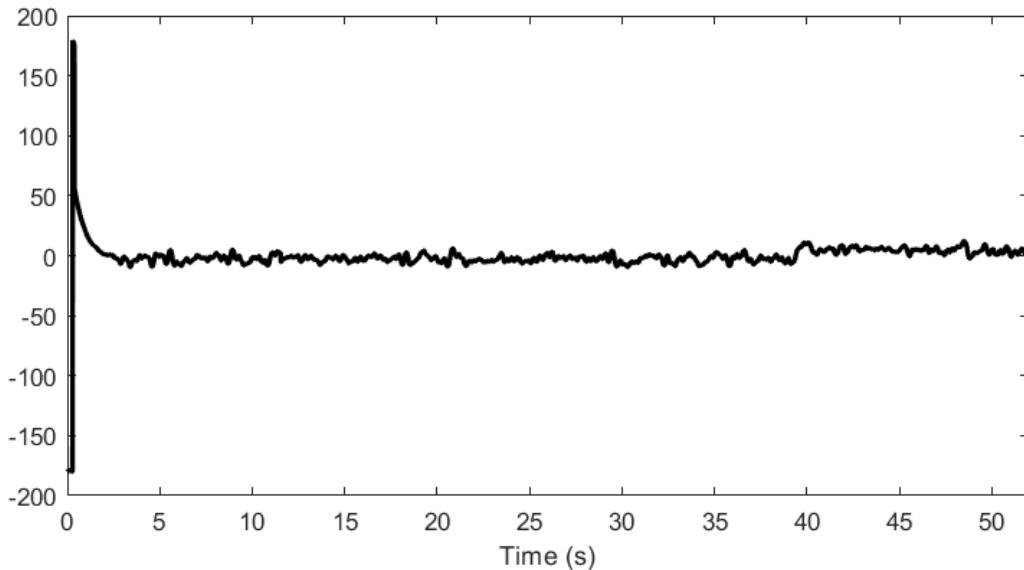
Khi tăng K_4 ở giá trị 0.5409 thành 0.6409, hệ Pendubot giữ cân bằng ở vị trí TOP trong 3.5 giây, sai số khi ổn định tối đa của thanh 1 là 23.4° , thanh 2 là 18.57° .

Giảm K_4

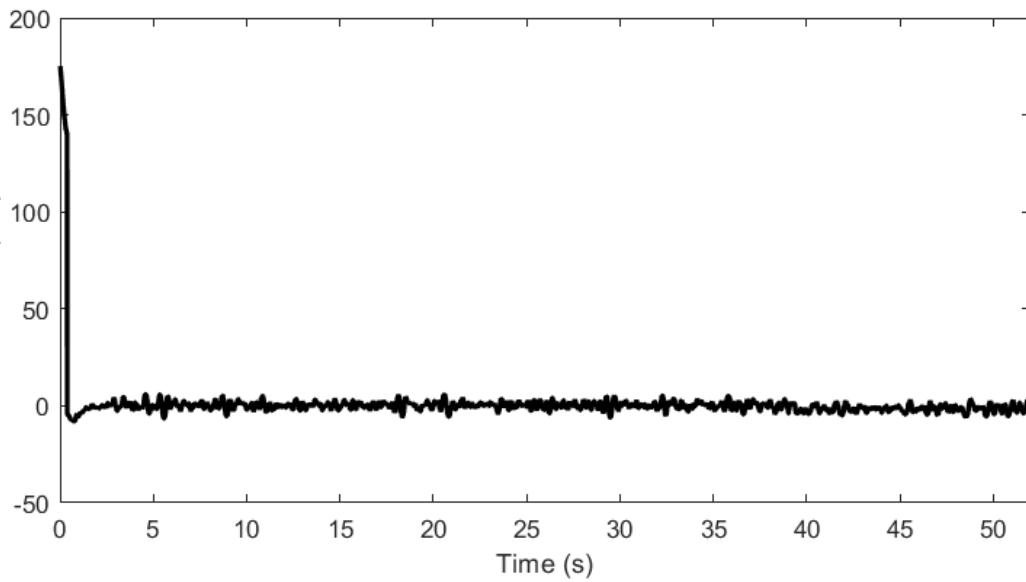
Giảm giá trị K_4 của ma trận điều khiển. Ma trận mới trở thành:

$$K = 10^3 \times [-0.0313 \quad 0.1970 \quad 3.5168 \quad 0.2409]$$

Kết quả nhận được như *Hình 5.52* và *Hình 5.53*. Trong 1 giây đầu, nhóm sinh viên nâng 2 thanh từ trạng thái nghỉ (vị trí xuôi theo chiều trọng trường) lên vị trí cân bằng.



Hình 5.52. Góc thanh 1 điều khiển bởi Bộ điều khiển LQR đã giảm K_4



Hình 5.53. Góc thanh 2 điều khiển bởi Bộ điều khiển LQR đã giảm K_4

Khi giảm K_4 ở giá trị 0.5409 thành 0.2409, hệ Pendubot giữ cân bằng ở vị trí TOP trong gần 1 phút, sai số khi ổn định tối đa của thanh 1 là 6.5° , thanh 2 là 4.5° .

Nhận xét

Khi tinh chỉnh K_4 thì hệ thống điều khiển rất ổn định với giá trị K_4 giảm. Thời gian cân bằng dài vượt trội hơn so với K tìm được từ GA hay các phép tinh chỉnh các giá trị thành phần của K .

5.4 NHẬN XÉT

Kết quả tốt nhất sau khi tinh chỉnh các giá trị của ma trận K là khi giảm giá trị K_4 . Sau thời gian nâng 2 thanh của hệ Pendubot về vị trí cân bằng, bộ xử lý trung tâm nhanh chóng xử lý và điều khiển động cơ để giữ thanh 2 về vị trí cân bằng rất tốt, có khả năng giữ thanh 1 của hệ quanh vùng cân bằng tại vị trí thẳng đứng hướng lên, sai số trong phạm vi là 6.5° .

Sự sai lệch giữa mô phỏng và thực nghiệm có thể gây nên do sai số trong quá trình nhận dạng động cơ. Trường hợp các thông số động cơ chính xác, có thể kết quả K từ mô phỏng có thể điều khiển trực tiếp mô hình thực tế mà không cần tinh chỉnh.

Chương 6

KẾT LUẬN - HƯỚNG PHÁT TRIỂN

6.1 KẾT LUẬN

Trên mô phỏng, hệ thống hầu như đã hoạt động ổn định với cả 3 bộ điều khiển LQR, Fuzzy và Tuyến tính hoá hồi tiếp. Theo như kết quả đã trình bày, không khó để nhận ra LQR là bộ điều khiển thích hợp nhất cho hệ Pendubot. Bộ điều khiển LQR hoạt động dựa trên phương trình toán của hệ thống để tìm ra ma trận điều khiển, do vậy việc tính toán phương trình toán học cũng như các thông số động cơ không thể xảy ra sai sót.

Qua quá trình điều khiển thực tế, nhóm sinh viên đã hiểu hơn về cách thức hoạt động của Bộ điều khiển LQR để có thể tinh chỉnh các thông số sao cho phù hợp với hệ thống. Việc xem cấu trúc của Bộ điều khiển LQR tương tự như Bộ điều khiển PID cũng không phải là phương thức tối ưu nhất, tuy nhiên sự khác biệt giữa quá trình tính toán lý thuyết và vận hành thực tế duy nhất chỉ ở các thông số động cơ, do đó đây có thể là nguyên nhân chính dẫn đến các sai lệch giữa mô phỏng và thực nghiệm.

Trong đề tài này, nhóm sinh viên đã điều khiển thành công hệ Pendubot cân bằng tại vị trí 0, đưa ra các kết luận so sánh giữa các bộ điều khiển phi tuyến (Bộ điều khiển LQR), tuyến tính (Bộ điều khiển Tuyến tính hoá hồi tiếp) và thông minh (Bộ điều khiển Fuzzy), đồng thời kết hợp được Giải thuật GA để tối ưu hóa chất lượng điều khiển của hệ thống.

6.2 HƯỚNG PHÁT TRIỂN

Đối với mô hình, nhóm sinh viên cần nhận dạng động cơ để có bộ thông số phần cứng chính xác, giúp các phương pháp điều khiển có thể cho ra đáp ứng nhanh và ổn định hơn mà không cần xem cấu trúc Bộ điều khiển LQR như giải thuật PID. Bên cạnh đó, bộ thông số mô hình cũng ảnh hưởng lớn đến việc mô hình không hoạt động được bằng phương pháp Fuzzy và Tuyến tính hoá hồi tiếp.

Đối với phần cơ của Pendubot, có thể phát triển thành hệ Pendubot 3 thanh. Khi đó cần xác định lại phương trình toán cũng như tính toán lại thông số của các bộ điều khiển.

Đối với các phương pháp điều khiển, trong tương lai, nhóm sinh viên mong muốn có thể thiết kế Bộ điều khiển Swing-up, hệ Pendubot sẽ tự động đưa 2 thanh về vị trí cân bằng dù đang ở bất kì vị trí nào mà không cần con người phải trực tiếp tác động như trong

CHƯƠNG 6. Kết luận – Hướng phát triển

đề tài này. Song song đó, với phương pháp Fuzzy, nhóm sinh viên sẽ tìm hiểu thêm tài liệu để có thể tự xây dựng một bộ điều khiển mờ của riêng mình mà không cần dùng lại bộ 81 luật mờ đã được tác giả đã công bố ở [1] trước đó.

TÀI LIỆU THAM KHẢO

- [1] Dương Hoài Nghĩa (2011), “Hệ thống điều khiển đa biến”, *Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh*
- [2] Huỳnh Thái Hoàng (2006), “Hệ thống điều khiển thông minh”, *Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh*
- [3] Nguyễn Thị Phương Hà (2016), “Lý thuyết điều khiển hiện đại”, *Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh, trang 3-35*
- [4] Nguyễn Văn Đông Hải (2011), “Xây dựng Bộ điều khiển nhúng Tuyến tính hoá vào ra cho hệ Xe con lắc ngược”, *Luận văn Thạc sĩ*, Trường Đại học Bách khoa Thành phố Hồ Chí Minh
- [5] Phan Việt Hùng (2013), “Nghiên cứu điều khiển cánh tay robot thiêu dẫn động hai bậc tự do – Pendubot”, *Luận văn Thạc sĩ*, Trường Đại học Đà Nẵng
- [6] Võ Anh Khoa, Nguyễn Minh Tâm, Trần Vi Đô, Nguyễn Văn Đông Hải, “Xây dựng và điều khiển mô hình con lắc ngược quay cho phòng thí nghiệm”, *Tạp chí Khoa học Giáo dục Kỹ thuật – Đại học Sư phạm Kỹ thuật TP Hồ Chí Minh*, số 69, trang 26-35
- [7] Võ Minh Tài, Trần Vi Đô, Vũ Văn Phong, Trần Đức Thiện, Nguyễn Minh Tâm, Ngô Văn Thuyên, Trần Thanh Phong, Nguyễn Văn Đông Hải (2018), “Xây dựng Bộ điều khiển Hồi tiếp tuyến tính hoá vào – ra cho hệ con lắc ngược quay”, *Tạp chí Khoa học Giáo dục Kỹ thuật – Đại học Sư phạm Kỹ thuật TP Hồ Chí Minh*, số 49, trang 32-40
- [8] Vũ Đình Đạt (2018), “Xây dựng bộ điều khiển trượt tối ưu cho hệ Pendubot bám quỹ đạo”, *Đồ án tốt nghiệp*, Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh
- [9] Xiao Qing Ma (2001), “Fuzzy Control for an Under-actuated Robotic Manipulator: Pendubot”, *Master Thesis in the Department of Mechanical Engineering*, Concordia University, Canada

PHỤ LỤC

Bảng 81 luật mò cho hệ Con lắc ngược trên xe

STT	θ	$\Delta\theta$	x	Δx	u
1	NE	NE	NE	NE	NB
2	NE	NE	NE	ZE	NB
3	NE	NE	NE	PO	NM
4	NE	NE	ZE	NE	NB
5	NE	NE	ZE	ZE	NM
6	NE	NE	ZE	PO	NE
7	NE	NE	PO	NE	NM
8	NE	NE	PO	ZE	NE
9	NE	NE	PO	PO	ZE
10	NE	ZE	NE	NE	NB
11	NE	ZE	NE	ZE	NM
12	NE	ZE	NE	PO	NE
13	NE	ZE	ZE	NE	NM
14	NE	ZE	ZE	ZE	NE
15	NE	ZE	ZE	PO	ZE
16	NE	ZE	PO	NE	NE
17	NE	ZE	PO	ZE	ZE
18	NE	ZE	PO	PO	PO

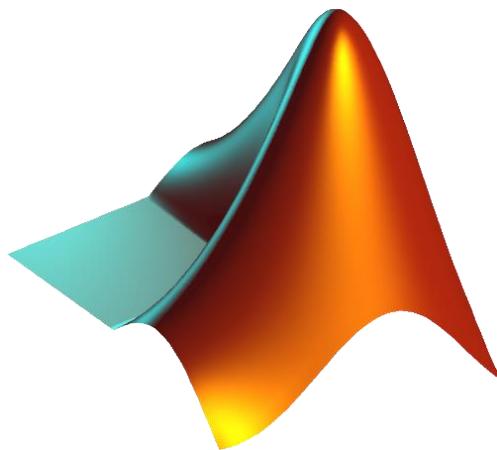
19	NE	PO	NE	NE	NM
20	NE	PO	NE	ZE	NE
21	NE	PO	NE	PO	ZE
22	NE	PO	ZE	NE	NE
23	NE	PO	ZE	ZE	ZE
24	NE	PO	ZE	PO	PO
25	NE	PO	PO	NE	ZE
26	NE	PO	PO	ZE	PO
27	NE	PO	PO	PO	PM
28	ZE	NE	NE	NE	NB
29	ZE	NE	NE	ZE	NM
30	ZE	NE	NE	PO	NE
31	ZE	NE	ZE	NE	ZE
32	ZE	NE	ZE	ZE	NE
33	ZE	NE	ZE	PO	ZE
34	ZE	NE	PO	NE	NE
35	ZE	NE	PO	ZE	ZE
36	ZE	NE	PO	PO	PO
37	ZE	ZE	NE	NE	NM
38	ZE	ZE	NE	ZE	NE
39	ZE	ZE	NE	PO	ZE

40	ZE	ZE	ZE	NE	NE
41	ZE	ZE	ZE	ZE	ZE
42	ZE	ZE	ZE	PO	PS
43	ZE	ZE	PO	NE	ZE
44	ZE	ZE	PO	ZE	PS
45	ZE	ZE	PO	PO	PS
46	ZE	PO	NE	NE	PM
47	ZE	PO	NE	ZE	NS
48	ZE	PO	NE	PO	PS
49	ZE	PO	ZE	NE	ZE
50	ZE	PO	ZE	ZE	PS
51	ZE	PO	ZE	PO	PM
52	ZE	PO	PO	NE	PS
53	ZE	PO	PO	ZE	PM
54	ZE	PO	PO	PO	PB
55	PO	NE	NE	NE	NM
56	PO	NE	NE	ZE	NS
57	PO	NE	NE	PO	ZE
58	PO	NE	ZE	NE	NS
59	PO	NE	ZE	ZE	ZE
60	PO	NE	ZE	PO	PS

61	PO	NE	PO	NE	ZE
62	PO	NE	PO	ZE	PS
63	PO	NE	PO	PO	PM
64	PO	ZE	NE	NE	NS
65	PO	ZE	NE	ZE	ZE
66	PO	ZE	NE	PO	PS
67	PO	ZE	ZE	NE	ZE
68	PO	ZE	ZE	ZE	PS
69	PO	ZE	ZE	PO	PM
70	PO	ZE	PO	NE	PS
71	PO	ZE	PO	ZE	PM
72	PO	ZE	PO	PO	PB
73	PO	PO	NE	NE	ZE
74	PO	PO	NE	ZE	PS
75	PO	PO	NE	PO	PM
76	PO	PO	ZE	NE	PS
77	PO	PO	ZE	ZE	PM
78	PO	PO	ZE	PO	PB
79	PO	PO	PO	NE	PM
80	PO	PO	PO	ZE	PB
81	PO	PO	PO	PO	PB

Giới thiệu phần mềm MATLAB

MATLAB là phần mềm cung cấp IDE (Integrated Development Environment – môi trường phát triển tích hợp) để viết code tính toán và lập trình, được ra mắt lần đầu tiên vào năm 1970 do Cleve Moler, chủ nhiệm khoa Máy tính tại Đại học New Mexico. Ở những năm thập niên 70, MATLAB chỉ được dùng nội bộ trong trường Đại học Stanford. Mãi đến năm 1983, Jack Little viết lại MATLAB ở ngôn ngữ C, đồng thời xây dựng thêm các thư viện, hộp công cụ (Tool box)... Một năm sau, Jack Little kết hợp với Cleve Moler cùng Steve Bangert – một nhà hoạt động kinh doanh, đã quyết định đưa MATLAB thành dự án thương mại. Từ đây, một bước ngoặt lớn trong lịch sử của MATLAB được hình thành, đó chính là sự ra đời của công ty MathWorks, công ty đại diện cho MATLAB đến tận bây giờ.



Logo phần mềm MATLAB

MATLAB cho phép tính toán các phương trình từ đơn giản đến phức tạp. Trong môi trường giáo dục ở các cơ sở kỹ thuật, MATLAB là công cụ hàng đầu trong việc nghiên cứu ở các phòng thí nghiệm, đóng vai trò hết sức quan trọng trong việc dạy và học. Trong công nghiệp, MATLAB là lựa chọn ưu tiên trong quá trình nghiên cứu các công nghệ, giải thuật ... trong các mảng chính yếu của nền công nghiệp như điều khiển tự động, xử lý ảnh ...



Logo MATLAB/Simulink

Bên cạnh tạo ra một môi trường để lập trình, MATLAB còn cung cấp cho người dùng môi trường để tạo ra các mô phỏng, các phép phân tích động học dựa theo các khôi lệnh đã được lập trình sẵn, gọi là công cụ Simulink. Các thư viện đã được các lập trình viên tính toán và lập trình, người dùng chỉ cần lấy khôi lệnh từ thư viện, thực hiện qua các thao tác đơn giản, kết nối các khôi lệnh lại với nhau là có thể xây dựng được một mô phỏng hoàn chỉnh. Giao diện sử dụng của MATLAB/Simulink vô cùng đơn giản. Có thể thấy, công cụ Simulink đã được MATLAB đầu tư hết sức tỉ mỉ, giúp người dùng đơn giản hóa quá trình tính toán, đẩy nhanh tiến độ hoàn thành các dự án.

Lý thuyết nhận dạng động cơ

Cấu trúc 1 động cơ DC đã được minh họa ở *Hình 2.7*. Phương pháp nhận dạng được nhóm sinh viên tham khảo dựa trên tài liệu [3].

Thiết lập phương trình tính toán giá trị cường độ dòng điện trong 1 đơn vị thời gian lấy mẫu:

$$\frac{di(t)}{dt} \approx \frac{i(t_{n+1}) - i(t_n)}{\Delta t} \quad (54)$$

Phương trình cân bằng điện áp và các bước triển khai được trình bày như ở (55)→(59)

$$u(t_n) = i(t_n) \times R_m + \frac{di(t)}{\Delta t} \times L_m + K_b \times \omega(t_n) \quad (55)$$

$$\Leftrightarrow u(t_n) = i(t_n) \times R_m + \frac{i(t_{n+1}) - i(t_n)}{\Delta t} \times L_m + K_b \times \omega(t_n) \quad (56)$$

$$\begin{aligned} &\Leftrightarrow \Delta t \times u(t_n) \\ &= i(t_n) \times R_m \times \Delta t + [i(t_{n+1}) - i(t_n)] \times L_m \\ &+ K_b \times \omega(t_n) \times \Delta t \end{aligned} \quad (57)$$

$$\begin{aligned} &\Leftrightarrow \Delta t \times u(t_n) \\ &= i(t_n) \times R_m \times \Delta t + i(t_{n+1}) \times L_m - i(t_n) \times L_m \\ &+ K_b \times \omega(t_n) \times \Delta t \end{aligned} \quad (58)$$

$$\Leftrightarrow i(t_{n+1}) = \frac{L_m - R_m \times \Delta t}{L_m} \times i(t_n) - \frac{K_b \times \Delta t}{L_m} \times \omega(t_n) + \frac{\Delta t}{L_m} \times u(t_n) \quad (59)$$

Phương trình cân bằng momen được trình bày và triển khai theo như các biểu thức:

$$i(t_n) \times K_b = \frac{d\omega(t_n)}{dt} \times J_m + \omega(t_n) \times C_m + K_f \times sgn(\omega) \quad (60)$$

$$\Leftrightarrow i(t_n) \times K_b = \frac{\omega(t_{n+1}) - \omega(t_n)}{\Delta t} \times J_m + \omega(t_n) \times C_m + K_f \times sgn(\omega) \quad (61)$$

$$\begin{aligned} \Leftrightarrow \Delta t \times i(t_n) \times K_b &= [\omega(t_{n+1}) - \omega(t_n)] \times J_m + \Delta t \times \omega(t_n) \times C_m \\ &\quad + \Delta t \times K_f \times sgn(\omega) \end{aligned} \quad (62)$$

$$\begin{aligned} \Leftrightarrow \Delta t \times i(t_n) \times K_b &= w(t_{n+1}) \times J_m - \omega(t_n) \times J_m + \Delta t \times \omega(t_n) \times C_m \\ &\quad + \Delta t \times K_f \times sgn(\omega) \end{aligned} \quad (63)$$

$$\Leftrightarrow \omega(t_{n+1}) = \frac{J_m - C_m \times \Delta t}{J_m} \omega(t_n) + \frac{K_b \times \Delta t}{J_m} i(t_n) - \frac{K_f \times \Delta t}{J_m} sgn(\omega) \quad (64)$$

Đặt:

$$a_1 = \frac{L_m - R_m \times \Delta t}{L_m} \quad a_2 = \frac{K_b \times \Delta t}{L_m} \quad a_3 = \frac{\Delta t}{L_m}$$

$$b_1 = \frac{K_b \times \Delta t}{J_m} \quad b_2 = \frac{J_m - C_m \times \Delta t}{J_m} \quad b_3 = -\frac{K_f \times \Delta t}{J_m}$$

Phương trình (59) và (64) được viết gọn thành:

$$i(t_{n+1}) = a_1 \times i(t_n) - a_2 \times \omega(t_n) + a_3 \times e(t_n) \quad (65)$$

$$\omega(t_{n+1}) = b_1 \times i(t_n) - b_2 \times \omega(t_n) + b_3 \times sgn(\omega(t_n)) \quad (66)$$

Trong đó:

- $i(t_{n+1})$ là dòng điện qua động cơ DC tại thời điểm t_{n+1}
- $i(t_n)$ là dòng điện qua động cơ DC tại thời điểm t_n
- $\omega(t_{n+1})$ là tốc độ góc của trục động cơ tại thời điểm t_{n+1}
- $\omega(t_n)$ là tốc độ góc của trục động cơ tại thời điểm t_n
- $e(t_n)$ là điện áp cấp cho động cơ tại thời điểm t_n

- $sgn(\omega(t_n))$ là dấu của tốc độ góc của trục động cơ DC tại thời điểm t_n
- Δt là thời gian lấy mẫu

Viết (65)(66) về dạng ma trận, thu được kết quả như sau:

$$\begin{bmatrix} i(2) \\ i(3) \\ i(4) \\ \dots \\ i(t_{n+1}) \end{bmatrix} = \begin{bmatrix} i(1) & \omega(1) & e(1) \\ i(2) & \omega(2) & e(2) \\ i(3) & \omega(3) & e(3) \\ \dots & \dots & \dots \\ i(t_n) & \omega(t_n) & e(t_n) \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (67)$$

$$\begin{bmatrix} \omega(2) \\ \omega(3) \\ \omega(4) \\ \dots \\ i(t_{n+1}) \end{bmatrix} = \begin{bmatrix} i(1) & \omega(1) & sgn(\omega(1)) \\ i(2) & \omega(1) & sgn(\omega(2)) \\ i(3) & \omega(3) & sgn(\omega(3)) \\ \dots & \dots & \dots \\ i(t_n) & \omega(t_n) & e(t_n) \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (68)$$

Từ biểu thức ma trận (67)(68), thiết lập chương trình nhận dạng bằng công cụ MATLAB/Simulink và dùng vi điều khiển để thu thập số liệu đo được từ động cơ. Tính toán $a_1, a_2, a_3, b_1, b_2, b_3$ và thay vào công thức (67)(68) để xác định các thông số $L_m, R_m, K_b, K_f, C_m, J_m$ của động cơ DC.

$$L_m = \frac{\Delta t}{a_3} \quad K_b = -\frac{a_2}{a_3} \quad R_m = \frac{(1 - a_1)}{a_3}$$

$$|T_f| = K_f = \frac{b_3 a_2}{b_1 a_3} \quad J_m = -\frac{a_2 \Delta t}{a_3 b_1} \quad C_m = \frac{a_2 (b_2 - 1)}{a_3 b_1}$$

