

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

ĐIỀU KHIỂN PID-FUZZY CHO HỆ PENDUBOT

GVHD: TS. NGUYỄN VĂN ĐÔNG HẢI
SVTH: MẠCH VĂN LONG
LÊ QUANG HÒA



Tp.Hồ Chí Minh, tháng 8/2022

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

---o0o---



ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: ĐIỀU KHIỂN PID-FUZZY CHO
HỆ PENDUBOT

GVHD: TS. Nguyễn Văn Đông Hải

SVTH: Mạch Văn Long

MSSV: 18151088

SVTH: Lê Quang Hòa

MSSV: 18151075

Tp. Hồ Chí Minh, tháng 08 năm 2022

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA**

---oo---



**ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: ĐIỀU KHIỂN PID-PUZZY CHO
HỆ PENDUBOT**

GVHD: TS. Nguyễn Văn Đông Hải

SVTH: Mạch Văn Long

MSSV: 18151088

SVTH: Lê Quang Hòa

MSSV: 18151075

Tp. Hồ Chí Minh, tháng 08 năm 2022

LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em xin gửi lời tri ân và biết ơn sâu sắc đến thầy Nguyễn Văn Đông Hải, người trực tiếp hướng dẫn đồ án tốt nghiệp, giải đáp thắc mắc, tận tình giúp đỡ, động viên, khích lệ chúng em, tạo mọi điều kiện tốt nhất từ bố trí phòng thí nghiệm đến các thiết bị vật tư trong suốt quá trình nghiên cứu, thực hiện đề tài.

Em cũng xin gửi lời cảm ơn đến các quý thầy cô trong Khoa Chất Lượng Cao nói chung và các thầy cô ngành Công Nghệ Kỹ Thuật Điều Khiển Và Tự Động Hóa nói riêng- Trường Đại học Sư Phạm Kỹ Thuật đã nhiệt tình giảng dạy cho chúng em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt bốn năm học tập tại đây.

Xin cảm ơn gia đình và bạn bè luôn động viên, ủng hộ và giúp đỡ chúng em những lúc chúng em gặp khó khăn trong việc làm đồ án tốt nghiệp, để chúng em hoàn thành được đồ án như ngày hôm nay.

Cuối cùng chúng em xin kính chúc các thầy cô dồi dào sức khỏe và gặt hái được nhiều thành công trong sự nghiệp giảng dạy cao quý. Đồng kính chúc những người thân và bạn bè có nhiều sức khỏe và thành công trong cuộc sống.

Tp. Hồ Chí Minh, tháng 08 năm 2022

Người thực hiện đề tài

TÓM TẮT

Dựa vào những kiến thức trong suốt thời gian học đại học và những tìm hiểu về các hệ thống điều khiển tự động. Nhóm chúng em nhận thấy việc tìm hiểu và thực hành điều khiển các hệ thống mang một ý nghĩa rất quan trọng. Chúng ta có thể bắt gặp rất nhiều hệ thống được tự động hóa và điều khiển chỉ bằng một chiếc máy tính với chương trình điều khiển để vận hành cả một hệ thống, mà đáng ra phải tốn rất nhiều nhân công và công sức.

Khi được tiếp cận với các bài báo nói về Pendubot và mô hình mẫu của nhà trường. Nhóm chúng em rất thích thú và bắt tay vào nghiên cứu. Ở đồ án này, Nhóm tập trung điều khiển ổn định dùng bộ điều khiển PID và PID-Fuzzy. Hai bộ điều khiển rất phổ biến và tính áp dụng thực tế rất cao, và chúng em đã chọn đề tài “Điều khiển PID-Fuzzy cho hệ Pendubot.

May mắn có được những linh kiện có sẵn. Tuy có cái đã hư hỏng phải sửa và thay thế để phù hợp, nhóm đã hoàn thiện phần cứng để phù hợp hơn. Với mô hình này nhóm dùng vi xử lý STM32F4 để điều khiển. Và dùng phần mềm matlab để nhúng vi xử lý đó. Về phần cứng, mô hình này gồm những linh kiện không qua phức tạp và dễ tìm mua như: STM32F4, tủ điện, cầu H, các khớp của mô hình làm bằng mica,...

Sau khi xong mô hình, chúng em bắt đầu tính toán các phương trình động học, sau đó tiến hành xây dựng từng khối mô phỏng trên công cụ Simulink cả Matlab và tiến hành áp dụng bộ điều khiển PID vào. Nhóm đã mất rất nhiều thời gian để tìm ra hệ số phù hợp nhưng chưa đạt được hiệu quả cao. Vì vậy nhóm đã áp dụng *Giải thuật di truyền GA* để hỗ trợ, nhưng chỉ hỗ trợ được mô phỏng còn thực tế vì nhiều lý do nên chỉ có thể vận dụng kiến thức và kinh nghiệm để tìm ra hệ số phù hợp.

Có được kết quả của bộ điều khiển PID, Nhóm tiến hành mô phỏng PID-Fuzzy và bộ điều khiển Swingup. Trong quá trình đó nhóm đã có kết quả và cũng như là những hạn chế. Những điều đó sẽ được nhóm chúng em thể hiện một cách cụ thể và chi tiết trong các mục bên dưới của bài báo cáo.

MỤC LỤC

LỜI CẢM ƠN.....	i
TÓM TẮT.....	i
MỤC LỤC	iii
DANH MỤC CÁC CHỮ VIẾT TẮT.....	V
LIỆT KÊ CÁC HÌNH VẼ.....	vi
LIỆT KÊ CÁC BẢNG BIỂU	viii
CHƯƠNG 1. TỔNG QUAN	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu.....	1
1.3 Nội dung nghiên cứu	1
1.4 Giới hạn	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	3
2.1 Giới thiệu Pendubot.....	3
2.2 Xây dựng phương trình toán học.....	3
2.3 Xét tính điều khiển được	11
2.3.1 Vị Trí Top.....	11
2.3.2 Vị Trí Mid.....	12
2.3.3 Vị trí OX-Top	13
CHƯƠNG 3. THIẾT KẾ BỘ ĐIỀU KHIỂN	14
3.1 Giới thiệu bộ điều khiển	14
3.1.1 Khái niệm bộ điều khiển PID	14
3.2 Lý thuyết bộ điều khiển Fuzzy (mờ)	16
3.2.1 Khái niệm về tập mờ	16
3.2.2 Các phép toán của tập mờ.....	17
3.2.3 Bộ điều khiển mờ cơ bản.....	18
3.3 Thiết kế bộ điều khiển swingup.....	19
3.4 Kết luận về bộ điều khiển	21
CHƯƠNG 4. THIẾT KẾ CHƯƠNG TRÌNH MÔ PHỎNG TRÊN MATLAB/SIMULINK	22
4.1 Giới thiệu Giải thuật di truyền [8,127-130].....	22
4.2 Giới thiệu Anfis	23
4.3 Mô Phỏng bộ điều khiển PD	24

4.4 Mô phỏng bộ điều khiển PD-Fuzzy.....	27
4.5 Mô Phỏng bộ điều khiển Swing up và bộ điều khiển PD.....	32
CHƯƠNG 5. THỰC NGHIỆM VÀ KẾT QUẢ	35
5.1 Giới thiệu chung	35
5.2 Phần Cứng	35
5.2.1 Vi xử lý STM32F4	35
5.2.2 Mạch chuyển USB UART CP2102.....	36
5.2.3 Tủ điện	37
5.2.4 Lựa chọn động cơ	37
5.2.5 Board cầu H	40
5.2.6 Lựa chọn Encoder.....	43
5.2.7 Thiết kế các khớp.	44
5.2.8 Thiết kế mạch ra chân.....	46
5.2.9 Sản phẩm	47
5.3 Kết quả bộ điều khiển thực nghiệm.....	49
5.3.1 Wajjung.	49
5.3.2 Lưu đồ nguyên lý điều khiển.....	49
5.3.3 Kết quả điều khiển bộ PID.	50
5.3.3 Kết quả điều khiển bộ PID-Fuzzy.	56
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	59
6.1 Kết Luận	59
6.2 Hướng phát triển.....	59
TÀI LIỆU THAM KHẢO	60
PHỤ LỤC	61

DANH MỤC CÁC CHỮ VIẾT TẮT

STT	Chữ Viết Tắt	Ý Nghĩa
1	TĐH	Tự Động Hóa
2	ĐKTĐ	Điều khiển tự động
3	PID	Proportional Integral Derivative
4	LQR	Linear Quadratic Regulator
5	GA	Genetic algorithm- Giải thuật di truyền
6	SIMO	Single - Input Multi – Output : Một vào nhiều ra
7	MIMO	Multiple Input – Multiple Output : Nhiều vào nhiều ra
8	PWM	Pulse-Width Modulation
9	DSP	Digital Signal Processing

LIỆT KÊ CÁC HÌNH VẼ

Hình 2.1 Một số mô hình pendubot.....	3
Hình 2. 2 Mô hình hóa Pendubot.....	4
Hình 2. 3 Vị trí Top	12
Hình 2.4 Vị trí mid	12
Hình 2. 5 Vị trí OX-Top	13
Hình 3.1 Sơ đồ khối của bộ điều khiển PID	14
Hình 3.2 Bộ điều khiển tỉ lệ - P	15
Hình 3.3 Bộ điều khiển tỉ lệ và tích phân – P	15
Hình 3.4 Bộ điều khiển PD.....	16
Hình 3.5 Hàm liên thuộc của tập mờ.....	17
Hình 3. 6 Hàm liên thuộc của hợp hai tập mờ có cùng cơ sở.....	17
Hình 3. 7 Giao hai tập mờ cùng cơ sở.....	17
Hình 3.8 Phép bù 2 tập hợp mờ	18
Hình 3.9 Bộ điều khiển mờ cơ bản.....	18
Hình 3. 10 Cấu trúc bên trong của một bộ điều khiển mờ	19
Hình 3.11 Sơ đồ điều khiển swingup	20
Hình 4. 1 Sơ đồ của giải thuật GA	23
Hình 4.2 Mô phỏng bộ điều khiển PD.....	25
Hình 4.3 Khối điều khiển PD	25
Hình 4.4 Khối phương trình toán của hệ thống	26
Hình 4.5 Kết quả mô phỏng link 1 của bộ điều khiển PD trong 10 giây	26
Hình 4.6 Kết quả mô phỏng link 2 của bộ điều khiển PD trong 10 giây	27
Hình 4.7 Mô phỏng bộ điều khiển PD-Fuzzy.	27
Hình 4. 8 Các khối điều khiển trong bộ PD-Fuzzy.	28
Hình 4.9 Tạo hiệu cho khối fuzzy.	28
Hình 4.10 Khối anfisedit.	29
Hình 4.11 Input của khối Fuzzy sau khi được tạo.....	29
Hình 4.12 Output của khối Fuzzy sau khi được tạo	30
Hình 4. 13 Các luật mờ.....	30
Hình 4.14 Kết quả link1 của bộ mô phỏng PD-Fuzzy trong 10 giây	31
Hình 4.15: Kết quả link2 của bộ mô phỏng PD-Fuzzy trong 10 giây	31
Hình 4.16 Chương trình mô phỏng bộ điều khiển Swingup và PD	32
Hình 4.17 Kết quả mô phỏng vị trí link1 so với phương thẳng đứng trong 10 giây.....	33

Hình 4.18 Kết quả mô phỏng vị trí link2 so với phương link1 trong 10 giây.....	33
Hình 4.19 Điện áp U cần cấp cho động cơ trong 10 giây	34
Hình 5. 1 Vi xử lý STM32F407	35
Hình 5. 2 USB UART CP2102.....	36
Hình 5. 3 Tủ điện.....	37
Hình 5. 4 Động cơ servo NISCA NF5475	38
Hình 5. 5 Kích thước động cơ	39
Hình 5. 6 Vị trí encoder và bô trí dây.....	39
Hình 5. 7 Cầu H L298 và Cầu H IR2184	40
Hình 5.8 Cầu tạo cầu L298.....	42
Hình 5.9 Cầu tạo sơ đồ nguyên lý	42
Hình 5.10 Encoder và thông tin chân ra	43
Hình 5. 11 Kích thước mặt cắt nang Encoder	43
Hình 5. 12 Sơ đồ nguyên lý hoạt động của Encoder	44
Hình 5.13 Khớp 1 và khớp 2	45
Hình 5.14 File cắt mica và sản phẩm	45
Hình 5. 15 Sơ đồ khói.....	46
Hình 5.16 Sơ đồ nguyên lý	46
Hình 5. 17 Mạch ra chân	47
Hình 5. 18 Sản phẩm phần cứng hoàn chỉnh.....	47
Hình 5.19 Góc bên của hệ thống pendubot	48
Hình 5. 20 Bên trong tủ điện.	48
Hình 5.21 Khối Wajjung	49
Hình 5. 22 Sơ đồ nguyên lý.....	49
Hình 5. 23 Tổng quan bộ điều khiển PID.....	50
Hình 5. 24 Chương trình trong khối Read_Encoder	50
Hình 5. 25 Chương trình trong khối Fcn	51
Hình 5. 26 Chương trình khối UART.....	52
Hình 5. 27 Cấu trúc bộ điều khiển PID	52
Hình 5. 28 Chương trình trong khối Lim_and_rotation direction.....	53
Hình 5. 29 Chương trình khối Function_Direction	53
Hình 5. 30 Chương trình trong khối Lim1	53
Hình 5. 31 Chương trình trong khối Setup	54
Hình 5. 32 Chương trình trong khối PWM_Controller	54

Hình 5.33 Góc của khớp thứ nhất so với phương thẳng đứng.	55
Hình 5. 34 Góc của khớp thứ 2 so với phương thẳng đứng	55
Hình 5.35 Tổng quan bộ điều khiển PID-Fuzzy.	56
Hình 5.36 Bộ điều khiển PID-FUZZY cho Kp1 và Kp2.....	56
Hình 5.37 Góc của khớp thứ nhất so với phương thẳng đứng	57
Hình 5.38 Góc của khớp thứ hai so với phương thẳng đứng.	57

LIỆT KÊ CÁC BẢNG BIỂU

Bảng 2.1 Kí hiệu các thông số mô hình.....	5
Bảng 2.2 Giá trị các thông số hệ thống pendubot	9
Bảng 5.2 So sánh cơ thông số cơ bản hai mạch cầu.....	41

CHƯƠNG 1. TỔNG QUAN

1.1 Đặt vấn đề

Trong công cuộc xây dựng đất nước đang bước vào thời kỳ công nghiệp hóa, hiện đại hóa đất nước với những cơ hội thuận lợi và những khó khăn thách thức lớn. Cùng với sự phát triển của khoa học kỹ thuật, ngành tự động hóa đã có những bước tiến quan trọng. Quá trình đó góp phần không nhỏ vào tăng năng suất lao động, giảm giá thành, tăng chất lượng và độ đồng đều của chất lượng. Và Pendubot là ví dụ tiêu biểu cho các hệ thống hụt (số đầu vào điều khiển nhỏ hơn bậc của mô hình). Đây là một mô hình phi tuyến rất khó điều khiển, có thể sử dụng làm đối tượng thí nghiệm cho các hệ thống điều khiển mới.

Đối với nhóm tác giả [1] sử dụng bộ điều khiển PID-Mờ cho pendubot nhưng chưa được tối ưu các thông số của bộ điều khiển, dẫn tới việc chọn các thông số điều khiển rất nhiều dẫn đến việc chọn bộ thông số khó khăn và trong thực tế việc này mất rất nhiều thời gian. Theo bài báo của nhóm tác giả [2] cấu trúc logic mờ của bộ mô phỏng khá nặng và chiếm nhiều ô nhớ. Và phải có thông tin chuyên gia điều khiển 81 luật mờ và rất khó để test hết luật mờ. Tác giả Phan Văn Kiểm trong Luận văn Thạc sĩ [3] cũng đề xuất giải thuật mờ trượt điều khiển hệ Pendubot. Kết quả mô phỏng và thực nghiệm cho thấy bộ điều khiển hoạt động tốt tại vị trí upright, và không thể giải quyết vấn đề khi điểm làm việc thay đổi.

Nhận thấy những hạn chế đó, chúng em hướng tới xây dựng bộ điều khiển ổn định PD-Fuzzy hai vị trí là: MID và TOP với thời gian dài. Đồng thời khảo sát những kiến thức đã được học trong thời gian qua. Tìm hiểu những ứng dụng và hướng phát triển của những mô hình lớn hơn, giúp đỡ cho việc nghiên cứu và điều khiển các hệ thống sau này. Nên nhóm em đã chọn đề tài “Điều khiển ổn định PID-Fuzzy hệ Pendubot”.

1.2 Mục tiêu

Đề tài “Điều khiển ổn định PD-Fuzzy hệ Pendubot” được thực hiện với những mục tiêu cụ thể như sau:

- Tính toán, thiết kế lại phần cứng.
- Tìm hiểu Module STM32 và các kết nối Matlab.
- Tiến hành mô phỏng Matlab.
- Thiết kế bộ mô phỏng cho mô hình thực và tìm bộ thông số phù hợp giữ cân bằng ở vị trí Top.

1.3 Nội dung nghiên cứu

Đồ án bao gồm các nội dung chính sau:

Chương 1: Tổng quan: Đặt vấn đề cho đồ án, các mục tiêu thực hiện và giới hạn lại những gì làm được.

Chương 2: Cơ sở lý thuyết: Giới thiệu về mô hình và xây dựng phương trình toán cho Pendubot và xét các vị trí có thể điều khiển được như: Top, Mid và Ox-Top.

Chương 3: Thiết kế bộ điều khiển: Giới thiệu bộ điều khiển PID, Fuzzy và Swingup và đưa ra kết luận chung.

Chương 4: Thiết kế chương trình mô phỏng: Giới thiệu phần mềm và công cụ mô phỏng. Giải thích về giải thuật di truyền GA. Mô phỏng các bộ điều khiển PD, Fuzzy-PD, Swingup và sau đó đưa ra kết quả.

Chương 5: Thực nghiệm và kết quả: Giới thiệu phần cứng, Mô hình điều khiển thực tế và kết quả

Chương 6: Kết luận và hướng phát triển: Những gì đã làm được và các hạn chế khi thực hiện, đồng thời đưa ra hướng phát triển cho hệ thống.

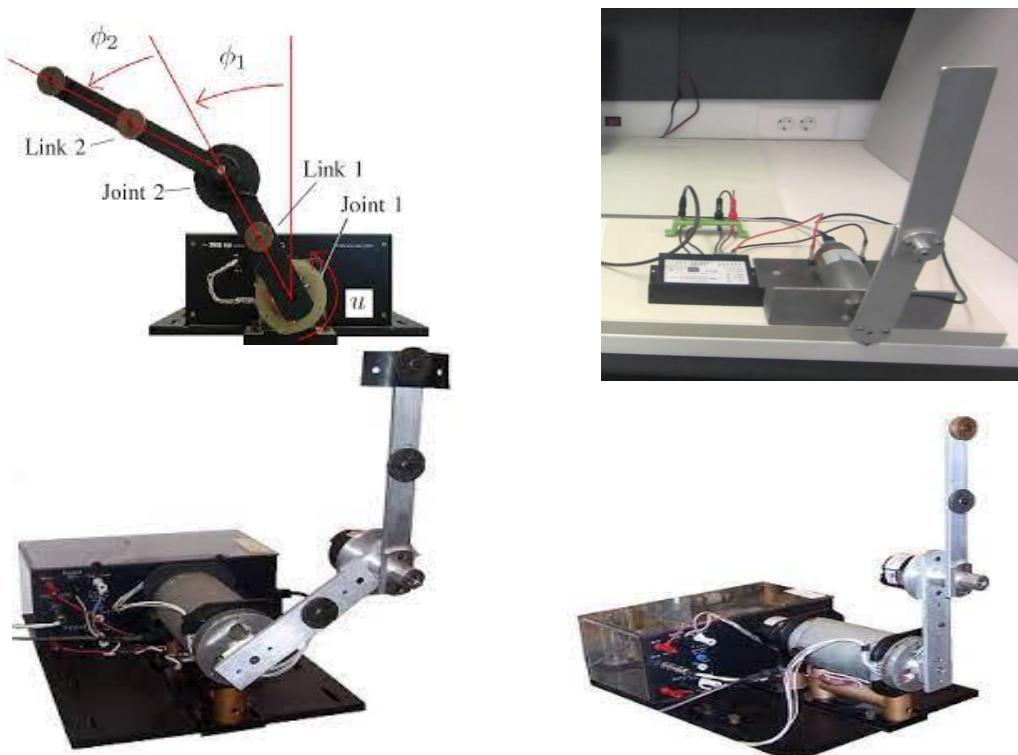
1.4 Giới hạn

Trong luận văn này, sinh viên chỉ tập trung vào việc điều khiển hệ Pendubot ổn định cân bằng sử dụng bộ điều khiển PD-Fuzzy. Thu thập số liệu thông qua mô phỏng và thực nghiệm. Từ đó, sinh viên đưa ra kết luận và hướng phát triển cho luận văn này.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu Pendubot

Mô hình Pendubot là mô hình có ngõ vào điều khiển ít hơn số bậc tự do, có độ phi tuyế̄n cao và rất khó để điều khiển. Pendubot với cấu trúc cơ khí không quá phức tạp nên được nhiều nhà nghiên cứu sử dụng để kiểm tra giải thuật điều khiển trong các phòng thí nghiệm. Trong bài báo cáo này, nhóm đã sử dụng giải thuật toàn phương tuyến tính dựa trên logic mờ (Fuzzy Linear Quadratic Regulator) và bộ điều khiển vi tích phân tỉ lệ PID để giải quyết bài toán cân bằng tại vị trí TOP của mô hình Pendubot. Ý tưởng chính của phương pháp này là sử dụng giải thuật PID kết hợp với giải thuật mờ để lựa chọn thông số điều khiển cho bộ điều khiển tuyến tính bậc hai LQR.

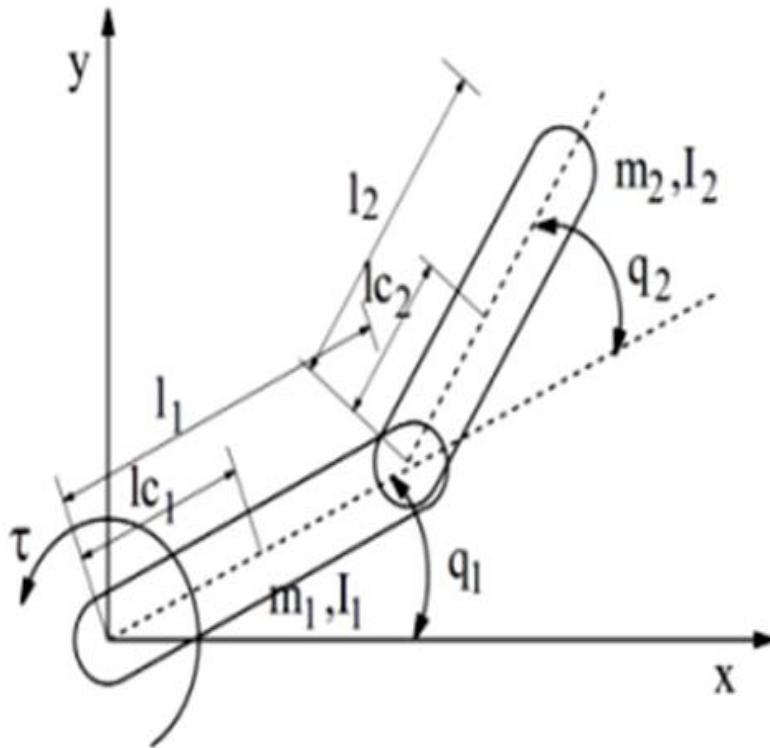


Hình 2.1 Một số mô hình pendubot.

2.2 Xây dựng phương trình toán học

Từ mô hình hệ thống ta tiến hành xây dựng hệ trực tọa độ để phù hợp với mô hình thực tế của nhóm và thuận tiện cho mô phỏng và điều khiển.

Sau khi có được hệ trực tọa độ hợp lý ta sử dụng những phương trình toán như động lực học, phương trình Euler để tính toán cho hệ thống pendubot để từ đó cái nhìn tổng quan và đúng đắn nhất cho hệ thống. Các bước được tiến hành như sau:



Hình 2. 2 Mô hình hóa Pendubot [1]

Ở Hình 2.2, ta chọn hệ trục tọa độ như hình, tung độ có hướng thẳng đứng hướng lên và hoành độ nằm ngang hướng sang phải.

Cấu trúc của một hệ thống Pendubot được chỉ ra trong hình 2.1 Hệ thống cơ khí kích thích dưới là một robot với một bộ truyền động ở khớp 1 và khớp 2 quay tự do quanh khớp 1.

Từ cấu tạo của Pendubot ta cần xây dựng mô hình toán học cho nó để phục vụ quá trình tổng hợp bộ điều khiển và mô phỏng trên máy tính một cách chính xác. Khi xây dựng phương trình động lực học. Ở đây ta sử dụng phương pháp Euler-Lagrange.

Phương trình Euler-Lagrange để mô tả chuyển động của một hệ bao toàn. Phương trình này thường được dùng để khảo sát những chuyển động cân bằng như dao động hay quỹ đạo của các hành tinh hệ cân bằng con lắc, xe cân bằng cũng như các hệ SIMO khác. Do vậy phương trình Euler – Larange đóng một vai trò rất quan trọng trong điều khiển nói chung và hệ pendubot của nhóm nói riêng.

Dạng tổng quát của phương trình Euler – Larange.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = \tau_j \quad (2.1)$$

Trong đó:

Chương 2. Cơ Sở Lý Thuyết

Các biến: q_1 , q_2 và τ . Trong đó, q_1 và q_2 là các tín hiệu đầu ra và τ là tín hiệu đầu vào. Do hệ pendubot có 1 tín hiệu vào và hai tín hiệu ra điều khiển nên τ_j là τ_1 . Khi đó τ_1 dựa trên bộ điều khiển được thiết kế.

Mỗi biến được định nghĩa như sau:

Ký hiệu	Đơn vị	Mô tả
q_1	rad	Góc khớp 1 với phương Ox
q_2	rad	Góc lệch khớp 2 so với khớp 1
l_1	m	Chiều dài khớp 1
l_2	m	Chiều dài khớp 2
I_1	kg. m^2	Moment quán tính của khớp 1
I_2	kg. m^2	Moment quán tính của khớp 2
g	m/ s^2	Gia tốc trọng trường
l_{c1}	m	Khoảng cách từ trục động cơ đến tâm của khớp 1
l_{c2}	m	Khoảng cách từ trục động cơ đến tâm của khớp 2
m_1	kg	Khối lượng khớp 1
m_2	kg	Khối lượng khớp 2

Bảng 2.1 Kí hiệu các thông số mô hình

Ta có bảng các biến số ký hiệu trên.

$$L = K - V \quad (2.2)$$

L: hàm Lagrange (Lagrange function)

K: động năng (kinetic energy)

V: Thế năng (potential energy)

$$\tau : \text{lực tác dụng (generalized forces)} \quad \tau = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}$$

Q: tọa độ suy rộng (generalized coordinates) $q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$

Với hệ toán học pendubot ta xét các thanh là đồng chất và có tâm của mỗi thanh ở vị trí chính giữa của mỗi khớp. Ta tiến hành xây dựng phương trình toán học cho hệ pendubot thông qua tính toán động năng và thế năng của hệ.

- Động năng của hệ

Tổng động năng của hệ thống pendubot như sau:

$$K = K_1 + K_2 \quad (2.3)$$

K_1 : động năng khớp 1

Với $K_1 = k_{11} + k_{12}$ Trong đó: ($k_{11} = \frac{1}{2}m_1v_1^2$) và ($k_{12} = \frac{1}{2}I_1\dot{q}_1^2$)

K_2 : động năng khớp 2

Với $K_2 = k_{21} + k_{22}$ Trong đó: ($k_{21} = \frac{1}{2}m_2v_2^2$) và ($k_{22} = \frac{1}{2}I_2\dot{q}_2^2$)

$$\implies K_1 = \frac{1}{2}m_1v_1^2 + \frac{1}{2}I_1\dot{q}_1^2; K_2 = \frac{1}{2}m_2v_2^2 + \frac{1}{2}I_2(\dot{q}_2 + \dot{q}_1)^2 \quad (2.4)$$

Từ hình 2.1 xét vị trí của các khớp:

Khớp 1:

$$x_1 = l_{c1} \cos(q_1); y_1 = l_{c1} \sin(q_1) \quad (2.5)$$

Khớp 2:

$$x_2 = l_{c1} \cos(q_1) + l_{c2} \cos(q_1 + q_2); y_2 = l_{c1} \sin(q_1) + l_{c2} \sin(q_1 + q_2) \quad (2.6)$$

Từ vị trí của các khớp ta đạo hàm cấp 1 để có vận tốc là:

$$v_2 = \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix}; v_1 = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} \quad (2.7)$$

Khi đó:

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2; v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 \quad (2.8)$$

Từ (2.6) (2.7) (2.8) và (2.2) ta thu được:

$$v^2_1 = l_{c1}^2 q_1^2 \quad (2.9)$$

$$v^2_2 = l_1^2 q_1^2 + l_{c2}^2 (\dot{q}_1 + \dot{q}_2)^2 + 2l_{c1}l_{c2}q_1(\dot{q}_1 + \dot{q}_2)\cos(q_2) \quad (2.10)$$

Thay (2.9) (2.10) vào phương trình (2.3) động năng của hệ là:

$$K = \frac{1}{2} \dot{q}_1(m_1 l_{c1}^2 + m_2 l_1^2 + I_2) + \frac{1}{2} (m_2 l_{c2}^2 + I_2)(\dot{q}_1 + \dot{q}_2)^2 \quad (2.11)$$

Thể năng của hệ pendubot là:

$$\begin{aligned} V &= V_1 + V_2 = m_1 gh_1 + m_2 gh_2 \\ \Leftrightarrow V &= m_1 g l_{c1} \sin(q_1) + m_2 g l_1 \sin(q_1) + m_2 g l_{c2} \sin(q_1 + q_2) \end{aligned} \quad (2.12)$$

Đặt:

$$\begin{aligned} \beta_1 &= m_1 l_{c1}^2 + m_2 l_1^2 + I_1; \quad \beta_2 = m_2 l_{c2}^2 + I_2; \quad \beta_3 = m_2 l_1 l_{c2}; \\ \beta_4 &= m_2 l_{c1} + m_2 l_1; \quad \beta_5 = m_2 l_{c2} \end{aligned} \quad (2.13)$$

Thay (2.13) lần lượt vào phương trình (2.11) và (2.12) thì động năng và thể năng của hệ là:

$$K = \frac{1}{2} \dot{q}_1^2 \beta_1 + \frac{1}{2} (\dot{q}_1 + \dot{q}_2)^2 + \beta_3 \cos(q_2) \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \quad (2.14)$$

$$V = \beta_4 g \sin(q_1) + \beta_5 g \sin(q_1 + q_2) \quad (2.15)$$

Từ phương trình (2.14) và (2.15) ta thu được mô hình động học của hệ:

$$\tau_1 = (\beta_2 + \beta_3 \cos(q_2)) \ddot{q}_1 + \beta_2 \ddot{q}_2 + \beta_3 \dot{q}_1^2 \sin(q_2) + g \beta_5 \cos(q_1 + q_2) \quad (2.16)$$

Mô hình động học hệ thống sẽ là:

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (2.17)$$

óí $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ là vector lực ma sát.

Xét giả định tiêu chuẩn tíc là không có ma sát, mô hình động học được suy ra theo dạng Euler-Lagrange như sau:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (2.18)$$

Trong đó: $D(q)$: ma trận quán tính

$C(q, \dot{q})$: coriolis/vector hướng tâm

$G(q)$: vector trọng lực

Với $q = [q_1 \ q_2]^T$; q_1 và q_2 là góc của khớp 1 so với phương ngang và góc giữa thanh 2 so với khớp 1; \dot{q} và \ddot{q} là vector vận tốc và vector gia tốc góc; $\tau = [\tau_1 \ 0]^T$, τ_1 là mô men xoắn bên ngoài đưa vào thanh 1.

Vì $D(q)$ là ma trận đối xứng định nghĩa dương nên $D(q)$ có thể viết dưới dạng

$$D(q) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \text{ trong đó } d_{12} = d_{21}$$

Từ công thức (2.17) và (2.18) các phần $D(q), C(q, \dot{q}), G(q)$ ta trình bày với năm thông số $\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5\}$ như sau:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (2.19)$$

$$\rightarrow d_{11} = \beta_1 + \beta_2 + 2\beta_3 \cos(q_2); d_{12} = \beta_2 + \beta_3 \cos(q_2); d_{21} = d_{12}; d_{22} = \beta_2$$

$$\rightarrow D(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos(q_2) & \beta_2 + \beta_3 \cos(q_2) \\ \beta_2 + \beta_3 \cos(q_2) & \beta_2 \end{bmatrix} \quad (2.20)$$

$$\rightarrow C(q) = \begin{bmatrix} -\beta_3 \dot{q}_2 \sin(q_2); -\beta_3 \dot{q}_2 \sin(q_2) - \beta_3 \dot{q}_1 \sin(q_2) \\ \beta_3 \dot{q}_1 \sin(q_2); 0 \end{bmatrix} \quad (2.21)$$

$$\text{Và: } G(q) = \begin{bmatrix} g_{11} \\ g_{12} \end{bmatrix} = \begin{bmatrix} \frac{\partial V}{\partial q_1} \\ \frac{\partial V}{\partial q_2} \end{bmatrix}$$

$$\rightarrow G(q) = \begin{bmatrix} \beta_4 g \cos(q_1) + \beta_5 g \cos(q_1 + q_2) \\ \beta_5 g \cos(q_1 + q_2) \end{bmatrix} \quad (2.22)$$

Những thông số sau là cần thiết diễn tả hệ thống:

Những thông số của các khớp có thể đo trực tiếp hoặc được tính bằng công thức từng thành phần riêng.

Chương 2. Cơ Sở Lý Thuyết

Moment của mỗi khớp có thể tính bằng công thức như sau: là moment từng phần ở trên khớp đó.

$$I_1 = I_{link1} + I_{sensor} = \frac{1}{3} m_1 l_1^2 + m_1 l_1^2 \rightarrow I_1 = \frac{1}{3} * 0.25 * 0.15^2 + 0.25 * 0.15^2 = 0.0189$$

Tương tự ta tính được $I_2 = 0.000409$

Với các thông số ta đo trực tiếp hoặc tính theo công thức ta được:

Thông số	Giá trị
l_1 (m)	0.15
l_2 (m)	0.25
I_1 ($\text{kg} \cdot \text{m}^2$)	0.0189
I_2 ($\text{kg} \cdot \text{m}^2$)	0.000409
g (m/s^2)	9.81
l_{c1} (m)	0.055
l_{c2} (m)	0.149
m_1 (kg)	0.25
m_2 (kg)	0.07

Bảng 2.2 Giá trị các thông số hệ thống pendubot

Thay vào các thông số ta được:

$$\begin{aligned} \beta_1 &= m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_1 = 0.0212; \quad \beta_2 = m_2 l_{c2}^2 + l_2 = 0.0020; \quad \beta_3 = m_2 l_1 l_{c2} = 0.0016; \\ \beta_4 &= m_2 l_{c1} + m_2 l_1 = 0.0243; \quad \beta_5 = m_2 l_{c2} = 0.0104. \end{aligned}$$

Điển tả không gian trạng thái hệ thống.

Bởi vì ma trận $D(q)$ là ma trận định nghĩa dương, ma trận nghịch đảo $D^{-1}(q)$ của nó tồn tại. nên ta viết lại phương trình hoạt động hệ thống dạng như sau:

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = D^{-1}(q)\tau - D^{-1}(q)C(q, \dot{q})\dot{q} - D^{-1}(q)G(q)$$

Ta có: $\begin{cases} x_1 = f_1(x) = q_1 \\ x_2 = f_2(x) = \dot{q}_1 \\ x_3 = f_3(x) = q_2 \\ x_4 = f_4(x) = \dot{q}_2 \end{cases} \rightarrow \begin{cases} \dot{x}_1 = x_1 \\ \dot{x}_2 = \ddot{q}_1 \\ \dot{x}_3 = x_2 \\ \dot{x}_4 = \ddot{q}_2 \end{cases}$ Và $x_5 = \tau_1$

$$(2.23)$$

Tính các thành phần trong (2.22):

$$D^{-1}(q) = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2(q_2)} \begin{bmatrix} \beta_2 & -\beta_2 - \beta_3 \cos(q_2) \\ -\beta_2 - \beta_3 \cos(q_2) & \beta_1 + \beta_2 + 2\beta_3 \cos(q_2) \end{bmatrix}$$

$$D^{-1}(q)\tau = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2(q_2)} \begin{bmatrix} \beta_2 & -\beta_2 - \beta_3 \cos(q_2) \\ -\beta_2 - \beta_3 \cos(q_2) & \beta_1 + \beta_2 + 2\beta_3 \cos(q_2) \end{bmatrix} \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}$$

$$\Leftrightarrow D^{-1}(q)\tau = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2(q_2)} \begin{bmatrix} \beta_2\tau_1 \\ -\beta_2 - \beta_3 \cos(q_2)\tau_1 \end{bmatrix}$$

$$D^{-1}(q)C(q, \dot{q})\dot{q} = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2(q_2)} \begin{bmatrix} -\beta_2\beta_3(\dot{q}_1 + \dot{q}_2)^2 \sin(q_2) - \beta_3^2 \dot{q}_1^2 \sin(q_2) \cos(q_2) \\ \beta_3(\beta_2 + \beta_3 \cos(q_2))(\dot{q}_1 + \dot{q}_2)^2 \sin(q_2) + \beta_3\beta_1 \dot{q}_1^2 \sin(q_2) \end{bmatrix}$$

$$D^{-1}(q)G(q) = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2(q_2)} \begin{bmatrix} -\beta_2\beta_3(\dot{q}_1 + \dot{q}_2)^2 \sin(q_2) - \beta_3^2 \dot{q}_1^2 \sin(q_2) \cos(q_2) \\ \beta_3(\beta_2 + \beta_3 \cos(q_2))(\dot{q}_1 + \dot{q}_2)^2 \sin(q_2) + \beta_3\beta_1 \dot{q}_1^2 \sin(q_2) \end{bmatrix}$$

$$\begin{bmatrix} \beta_4 g \cos(q_1) + \beta_5 g \cos(q_1 + q_2) \\ \beta_5 g \cos(q_1 + q_2) \end{bmatrix}$$

$$= \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2(q_2)} \begin{bmatrix} \beta_2\beta_4 g \cos(q_1) - \beta_3\beta_5 g \cos(q_1 + q_2) \\ \beta_5 g (\beta_1 + \beta_3 \cos(q_2)) \cos(q_1 + q_2) - \beta_4 g (\beta_2 + \beta_3 \cos(q_2)) \cos(q_1) \end{bmatrix}$$

Với: $\ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}$

$$\ddot{q}_1 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2(q_2)} \left[\beta_2\tau_1 + \beta_2\beta_3(\dot{q}_1 + \dot{q}_2)^2 \sin(q_2) + \beta_3^2 \dot{q}_1^2 \sin(q_2) \cos(q_2) \right. \\ \left. - \beta_2\beta_4 g \cos(q_1) + \beta_3\beta_5 g \cos(q_2) \cos(q_1 + q_2) \right] \quad (2.24)$$

$$\ddot{q}_2 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2(q_2)} \left[(-\beta_3(\beta_2 + \beta_3 \cos(q_2))(\dot{q}_1 + \dot{q}_2)^2 \sin(q_2) \right. \\ \left. - \beta_5 g (\beta_1 + \beta_3 \cos(q_2)) \cos(q_2 + q_1) + \beta_4 g (\beta_2 + \beta_3 \cos(q_2)) \cos(q_1) \right. \\ \left. - \beta_2 - \beta_3 \cos(q_2)) \tau_1 - \beta_3 \dot{q}_1^2 (\beta_1 + \beta_3 \cos(q_2)) \right] \quad (2.25)$$

Mô hình không gian trạng thái của hệ được diễn tả như sau:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & 0 & \frac{\partial f_2}{\partial x_3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_4}{\partial x_1} & 0 & \frac{\partial f_4}{\partial x_3} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial u} \\ 0 \\ \frac{\partial f_4}{\partial u} \end{bmatrix} u = Ax + Bu$$

2.3 Xét tính điều khiển được

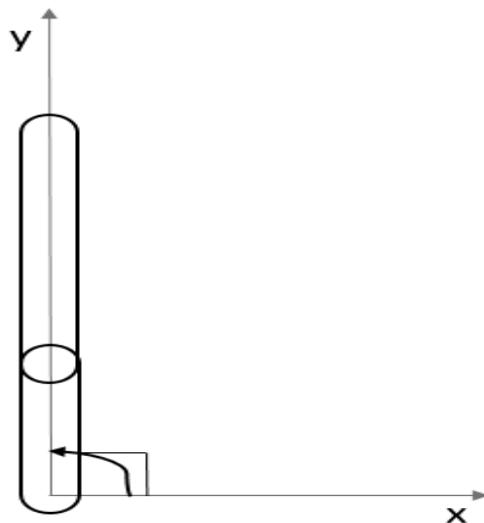
Hệ thống pendubot không phải ở bất kỳ vị trí nào cũng có thể điều khiển cho nên từ phương trình toán của hệ thống nhóm tiến hành xét tính điều khiển cho pendubot. Ở đây ta xét 3 trường hợp đặc trưng tiêu biểu như sau.

2.3.1 Vị Trí Top

Tuyến tính hóa hệ thống xung quanh điểm cân bằng Top.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(\beta_2\beta_4 - \beta_3\beta_5)g}{\beta_1\beta_2 - \beta_3} & 0 & \frac{-\beta_3\beta_5g}{\beta_1\beta_2 - \beta_3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\beta_5g(\beta_1 - \beta_3) - \beta_4g(\beta_2 + \beta_3)}{\beta_1\beta_2 - \beta_3} & 0 & \frac{\beta_5g(\beta_1 + \beta_3)}{\beta_1\beta_2 - \beta_3} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\beta_2}{\beta_1\beta_2 - \beta_3} \\ 0 \\ \frac{-\beta_2 - \beta_3}{\beta_1\beta_2 - \beta_3} \end{bmatrix} u \quad (2.26)$$

$$= Ax + Bu$$



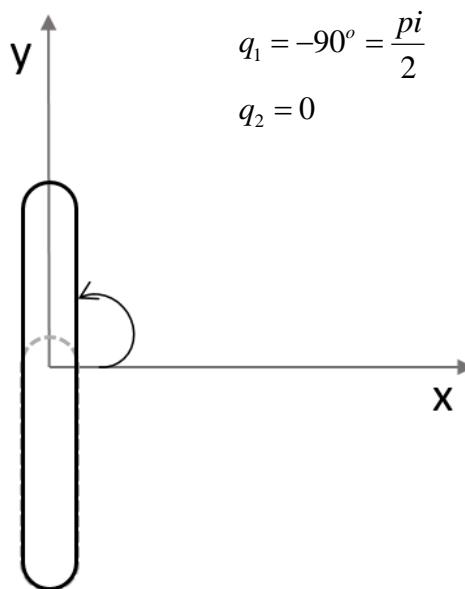
Hình 2. 3 Vị trí Top [2]

Với Hình 2. 3 vị trí của hai khớp là tại vị trí thẳng đứng cho nên góc điều khiển của mỗi khớp so với phương ngang và so với khớp 1 lần lượt là 90 độ và 0 độ.

Ta có tại vị trí cân bằng Top: $\text{Rank} = (B; AB; A^2B; A^3B) = 4$ Hạng của ma trận A

Vậy tại vị trí cân bằng Top có thể điều khiển được.

2.3.2 Vị Trí Mid



Hình 2.4 Vị trí mid [2]

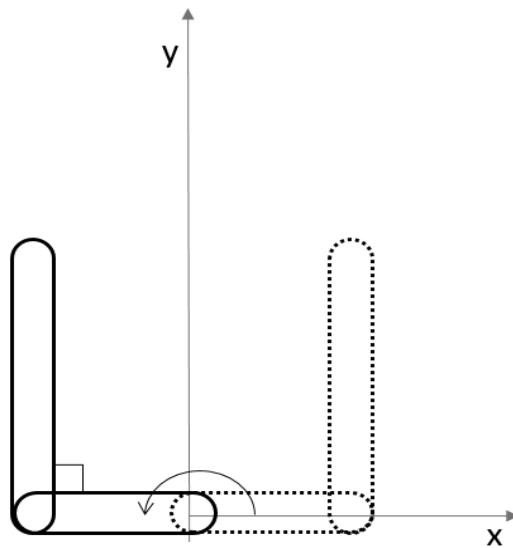
Ta có tại vị trí cân bằng Mid: $\text{Rank} = (B; AB; A^2B; A^3B) = 4$ Hạng của ma trận A

Vậy tại vị trí cân bằng Mid có thể điều khiển được

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(\beta_2\beta_4 - \beta_3\beta_5)g}{\beta_1\beta_2 - \beta_3} & 0 & \frac{-\beta_3\beta_5g}{\beta_1\beta_2 - \beta_3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\beta_5g(\beta_1 - \beta_3) - \beta_4g(\beta_2 + \beta_3)}{\beta_1\beta_2 - \beta_3} & 0 & \frac{\beta_5g(\beta_1 + \beta_3)}{\beta_1\beta_2 - \beta_3} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\beta_2}{\beta_1\beta_2 - \beta_3} \\ 0 \\ \frac{-\beta_2 - \beta_3}{\beta_1\beta_2 - \beta_3} \end{bmatrix} u$$

$$= Ax + Bu \quad (2.27)$$

2.3.3 Vị trí OX-Top



Hình 2. 5 Vị trí OX-Top [2]

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-(\beta_2\beta_4 - \beta_3\beta_5)g}{\beta_1\beta_2 - \beta_3} & 0 & \frac{\beta_3\beta_5g}{\beta_1\beta_2 - \beta_3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-\beta_5g(\beta_1 - \beta_3) + \beta_4g(\beta_2 + \beta_3)}{\beta_1\beta_2 - \beta_3} & 0 & \frac{\beta_5g(\beta_1 - \beta_3)}{\beta_1\beta_2 - \beta_3} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\beta_2}{\beta_1\beta_2 - \beta_3} \\ 0 \\ \frac{-\beta_2 + \beta_3}{\beta_1\beta_2 - \beta_3} \end{bmatrix} u$$

$$= Ax + Bu \quad (2.28)$$

Ta có tại vị trí cân bằng Mid-Top: $\text{Rank} = (B; AB; A^2B; A^3B) = 2 < 4$ Hạng của ma trận A

Vậy tại vị trí cân bằng này có thể điều khiển được.

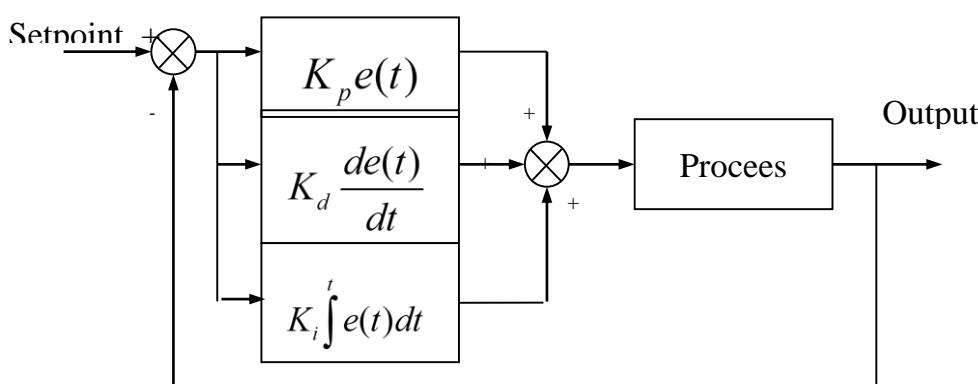
CHƯƠNG 3. THIẾT KẾ BỘ ĐIỀU KHIỂN

3.1 Giới thiệu bộ điều khiển

3.1.1 Khái niệm bộ điều khiển PID

- Bộ điều khiển PID là bộ điều khiển được sử dụng rộng rãi trong các hệ thống vòng kín mà có tín hiệu phản hồi. Bộ điều khiển sẽ tính toán sai số giá trị đặt mong muốn và các thông số biến đổi. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh các giá trị đầu vào (K_p , K_i , K_d). Để đạt được kết quả tốt nhất các thông số PID phải được điều chỉnh phù hợp với tính chất của hệ thống. Để cải thiện hệ thống ở xác lập và quá độ thì tín hiệu tác động:

$$U(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(t) dt \quad (3.1)$$



Hình 3.1 Sơ đồ khái niệm của bộ điều khiển PID [1]

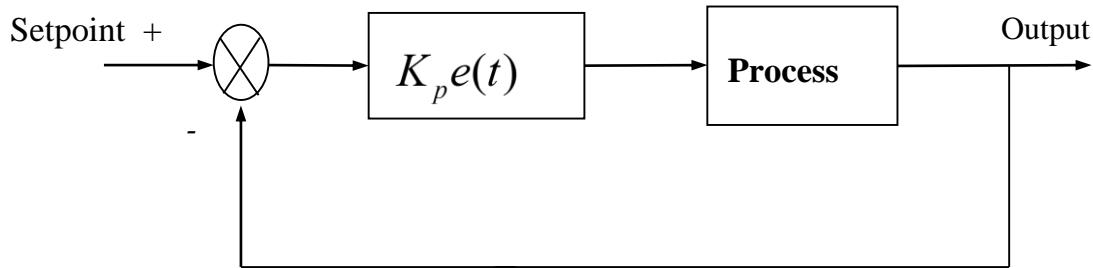
Trong Đó:

- P (Proportional): là phương pháp điều chỉnh tỉ lệ, giúp tạo ra tín hiệu điều chỉnh tỉ lệ với sai lệch đầu vào theo thời gian lấy mẫu.
- I (Integral): là tích phân của sai lệch theo thời gian lấy mẫu. Điều khiển tích phân là phương pháp điều chỉnh để tạo ra các tín hiệu điều chỉnh sao cho độ sai lệch giảm về 0. Từ đó cho ta biết tổng sai số tức thời theo thời gian hay sai số tích lũy trong quá khứ. Khi thời gian càng nhỏ thể hiện tác động điều chỉnh tích phân càng mạnh, tương ứng với độ lệch càng nhỏ.
- D (Derivative): là vi phân của sai lệch. Điều khiển vi phân tạo ra tín hiệu điều chỉnh sao cho tỉ lệ với tốc độ thay đổi sai lệch đầu vào. Thời gian càng lớn thì phạm vi điều chỉnh vi phân càng mạnh, tương ứng với bộ điều chỉnh đáp ứng với thay đổi.

3.1.2 Những bộ điều khiển khác

Bộ điều khiển tỉ lệ – P (Proportional Controller).

$$U(t) = K_p e(t) \quad (3.2)$$

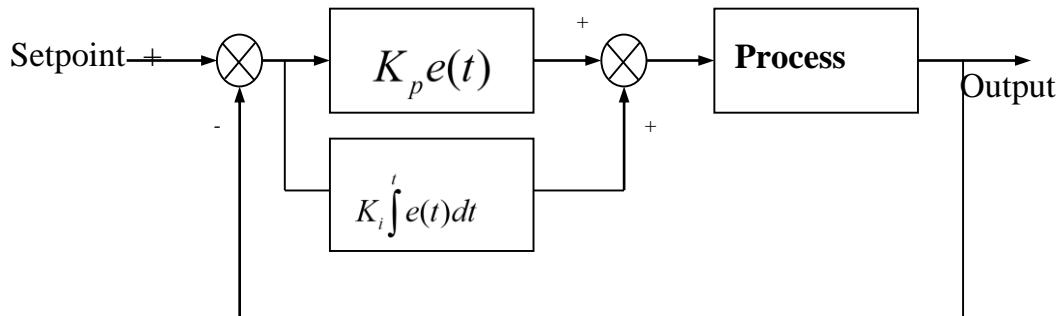


Hình 3.2 Bộ điều khiển tỉ lệ - P [1]

Điều khiển tỉ lệ cho phép nhanh chóng đạt trị số yêu cầu nhưng thường có sai lệch. Để giảm sai lệch người ta tăng độ lợi K , nếu tăng K quá dẫn đến vọt lố δ_{max} lớn và hệ có thể mất ổn định.

- Bộ điều khiển tỉ lệ và tích phân - PI (Proportional and Integral Controller)

$$U(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (3.3)$$

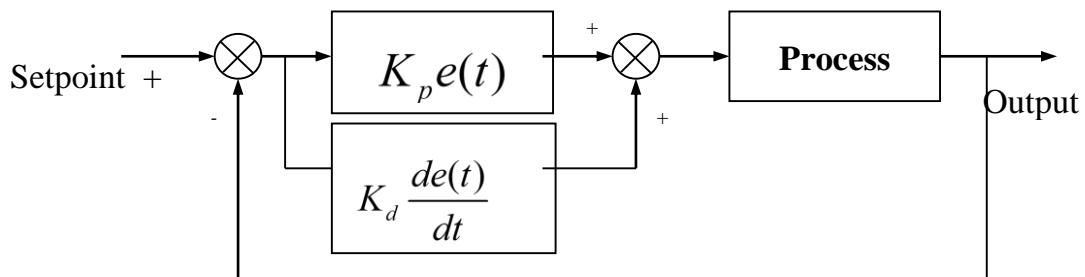


Hình 3.3 Bộ điều khiển tỉ lệ và tích phân – P [1]

Để nâng cao độ chính xác của hệ thống, người ta thêm khâu điều khiển tích phân. Nếu còn sai lệch, tín hiệu tác động còn duy trì để làm giảm sai lệch này. Điều khiển PI làm cho hệ hữu sai thành vô sai. Loại của hệ thống được tăng lên nghĩa là bậc của nó cũng tăng lên, do đó độ ổn định của hệ kém đi.

$$U(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (3.4)$$

- Bộ điều khiển đạo hàm - PD (Proportional and Derivative (PD) Controller)



Hình 3.4 Bộ điều khiển PD [1]

Nếu $C(t)$ tăng (độ vọt lồ lớn) thì $e(t)$ giảm $\Rightarrow \frac{de(t)}{dt} < 0$, nên $U(t) = K_p e(t) + K_d \frac{de(t)}{dt}$ giảm nhiều không cho $C(t)$ tăng quá. Vì vậy điều khiển PD làm giảm chấn của hệ thống tăng lên, giảm vọt lồ nhưng thời gian trễ sẽ lâu hơn.

Điều khiển PD chỉ ảnh hưởng tới sai số xác lập E_{xl} , nếu E_{xl} biến thiên theo thời gian ($\frac{d}{dt} \neq 0$) mà không ảnh hưởng nếu $E_{xl}(t) = C_{te}$. Nếu E_{xl} tăng theo t , tín hiệu tác động có thành phần tỉ lệ với $\frac{de(t)}{dt}$ làm giảm biên độ sai số. [1]

3.2 Lý thuyết bộ điều khiển Fuzzy (mờ)

Cũng như bộ điều khiển PID, bộ điều khiển Fuzzy cũng được sử dụng rộng rãi và được áp dụng nhiều trong các mô hình điều khiển trong đời sống và công nghiệp.

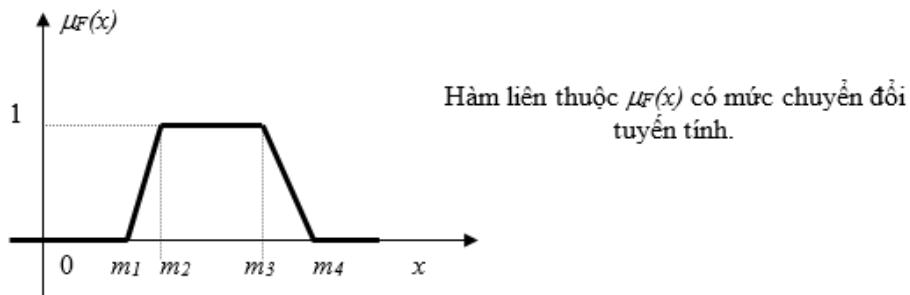
3.2.1 Khái niệm về tập mờ

Theo luận văn [4] ta có :

Tập mờ F xác định trên tập kinh điển M là một tập mà mỗi phần tử của nó là một cặp các giá trị $(x, \mu_F(x))$ trong đó $x \in M$ và μ_F là ánh xạ. $\mu_F: M \rightarrow [0, 1]$

Ánh xạ μ_F được gọi là hàm liên thuộc (hoặc hàm phụ thuộc) của tập mờ F . Tập kinh điển M được gọi là cơ sở của tập mờ F . Sử dụng các hàm liên thuộc để tính độ phụ thuộc của một phần tử x nào đó có hai cách: tính trực tiếp (nếu $\mu_F(x)$ ở dạng công thức tường minh) hoặc tra bảng (nếu $\mu_F(x)$ ở dạng bảng).

Các hàm liên thuộc $\mu_F(x)$ có dạng “tròn” được gọi là hàm liên thuộc kiểu S . Đối với hàm liên thuộc kiểu S , do các công thức biểu diễn $\mu_F(x)$ có độ phức tạp lớn nên thời gian tính độ phụ thuộc cho một phần tử lâu. Trong kỹ thuật điều khiển mờ thông thường, các hàm liên thuộc kiểu S thường được thay gần đúng bằng một hàm tuyến tính từng đoạn. Một hàm liên thuộc có dạng tuyến tính từng đoạn được gọi là hàm liên thuộc có mức chuyển đổi tuyến tính.



Hình 3.5 Hàm liên thuộc của tập mờ [4]

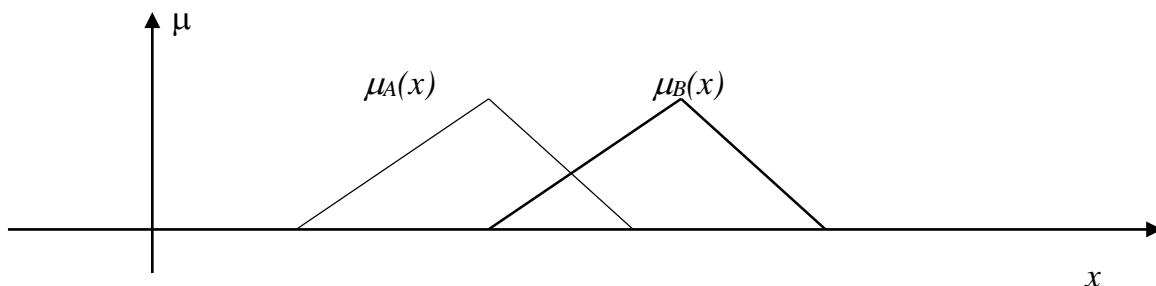
Hàm liên thuộc $\mu_F(x)$ như trên với $m_1 = m_2$ và $m_3 = m_4$ chính là hàm phụ thuộc của một tập kinh điển.

3.2.2 Các phép toán của tập mờ

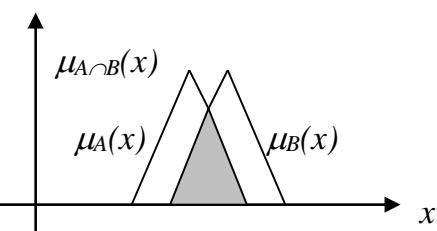
- Phép Hợp:

Hợp của hai tập mờ A và B có cùng cơ sở M là một tập mờ cũng xác định trên cơ sở M với hàm liên thuộc:

$$\mu_{A \cup B}(x) = \text{MAX}\{\mu_A(x), \mu_B(x)\}$$



Hình 3.6 Hàm liên thuộc của hợp hai tập mờ có cùng cơ sở [4].



Hình 3.7 Giao hai tập mờ cùng cơ sở [4].

Giao của hai tập mờ A và B có cùng cơ sở M là một tập mờ cũng xác định trên cơ sở M với hàm liên thuộc:

$$\mu_{A \cap B}(x) = \text{MIN}\{\mu_A(x), \mu_B(x)\}$$

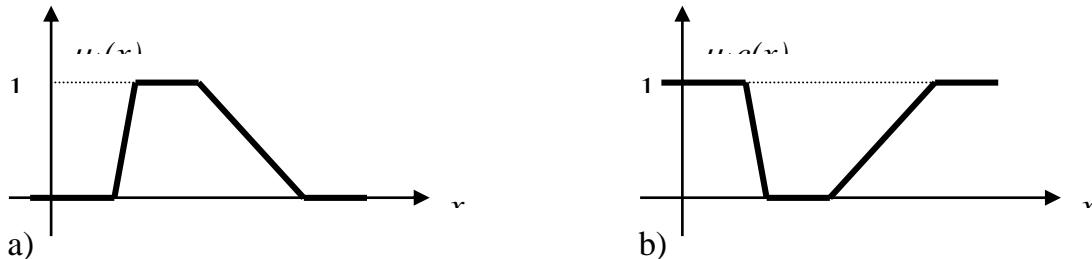
Trong công thức trên ký hiệu min được viết hoa thành MIN chỉ để biểu hiện rằng phép tính lấy cực tiểu được thực hiện trên tập mờ. Bản chất phép tính không có gì thay đổi.

- Phép Bù:

Chương 3. Thiết Kế Bộ Điều Khiển

Bù của tập mờ A có cơ sở M và hàm liên thuộc $\mu_A(x)$ là một tập mờ A^C xác định trên cùng cơ sở M với hàm liên thuộc:

$$\mu_{A^C}(x) = 1 - \mu_A(x).$$



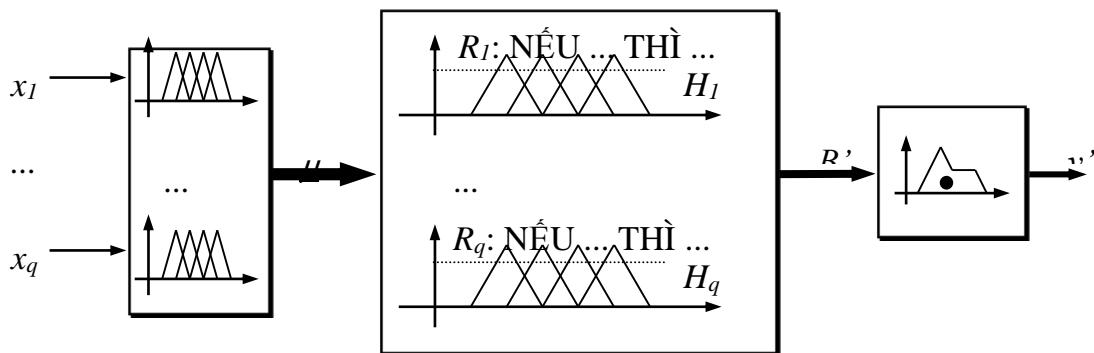
Tập bù A^C của tập mờ A .

a) Hàm liên thuộc của tập mờ A .

b) Hàm liên thuộc của tập mờ A^C .

Hình 3.8 Phép bù 2 tập hợp mờ [4].

3.2.3 Bộ điều khiển mờ cơ bản



Hình 3.9 Bộ điều khiển mờ cơ bản [4].

Những thành phần cơ bản của một bộ điều khiển mờ bao gồm khâu Fuzzy hóa, thiết bị thực hiện luật hợp thành và khâu giải mờ. Một bộ điều khiển mờ chỉ gồm ba thành phần như vậy có tên gọi là bộ điều khiển mờ cơ bản.

Do bộ điều khiển mờ cơ bản chỉ có khả năng xử lý các giá trị tín hiệu hiện thời nên nó thuộc nhóm các bộ điều khiển tĩnh. Tuy vậy để mở rộng miền ứng dụng của chúng vào các bài toán điều khiển động, các khâu động học cần thiết sẽ được nối thêm vào bộ điều khiển mờ cơ bản. Các khâu động đó chỉ có nhiệm vụ cung cấp thêm cho bộ điều khiển mờ cơ bản các giá trị đạo hàm hay tích phân của tín hiệu. Với những khâu động bổ sung này, bộ điều khiển cơ bản sẽ được gọi là bộ điều khiển mờ động.

- Tính phi tuyến của luật mờ.

Một bộ điều khiển mờ có ba khâu cơ bản gồm:

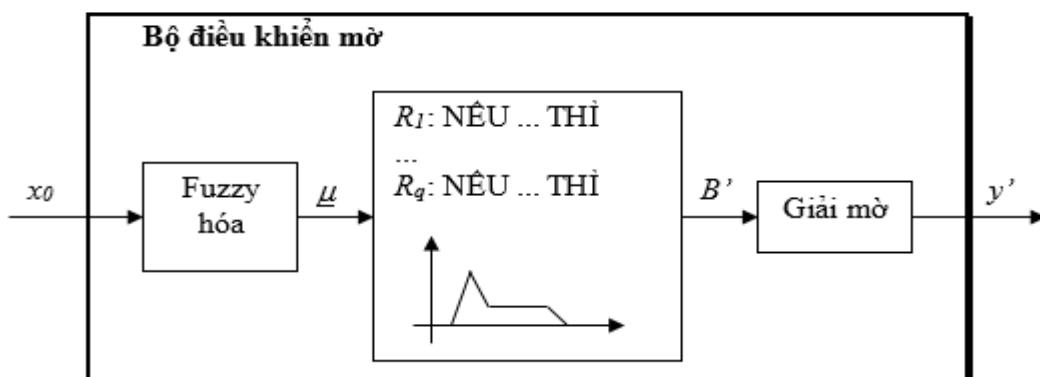
Chương 3. Thiết Kế Bộ Điều Khiển

- Khâu Fuzzy hóa có nhiệm vụ chuyển đổi một giá trị rõ đầu vào x_0 thành một vector μ gồm các độ phụ thuộc của giá trị rõ đó theo các giá trị mờ đã định nghĩa cho biến ngôn ngữ đầu vào,
- Khâu thực hiện luật hợp thành, có tên gọi là thiết bị hợp thành, xử lý vector μ và cho ra giá trị mờ B' của biến ngôn ngữ đầu ra,
- Khâu giải mờ, có nhiệm vụ chuyển đổi tập mờ B' thành một giá trị rõ y' chấp nhận được cho đối tượng (tín hiệu điều chỉnh).

Các bộ điều khiển mờ sẽ được phân loại dựa trên quan hệ vào/ra toàn cục của tín hiệu vào x_0 và tín hiệu ra y' . Quan hệ toàn cục đó được gọi là quan hệ truyền đạt.

Việc phân loại quan hệ truyền đạt một bộ điều khiển mờ dựa vào 7 tiêu chuẩn:

- Tĩnh hay động.
- Tuyến tính hay phi tuyến.
- Tham số tập trung hay tham số rải.
- Liên tục hay rời rạc.
- Tham số tĩnh hay tham số động.
- Tiền định hay ngẫu nhiên.
- Ôn định hay không ôn định.



Hình 3. 10 Cấu trúc bên trong của một bộ điều khiển mờ [4].

Xét từng khâu của bộ điều khiển mờ gồm các khâu Fuzzy hóa, thiết bị hợp thành và giải mờ, thì thấy rằng trong quan hệ vào/ra giá trị y' tại đầu ra chỉ phụ thuộc vào một mình giá trị x_0 của đầu vào chứ không phụ thuộc vào các giá trị đã qua của tín hiệu $x(t)$, tức là chỉ phụ thuộc vào giá trị của $x(t)$ tại đúng thời điểm đó.

3.3 Thiết kế bộ điều khiển swingup

Theo [6] bộ điều khiển swingup được thể kế như sau:

Phương trình mô tả hệ Pendubot (2.18) được biểu diễn lại như sau:

$$D_{11}\ddot{q}_1 + D_{12}\ddot{q}_2 + C_{11}\dot{q}_1 + C_{12}\dot{q}_2 + G_1 = \tau_m \quad (3.5)$$

$$D_{21}\ddot{q}_1 + D_{22}\ddot{q}_2 + C_{21}\dot{q}_1 + G_2 = 0 \quad (3.6)$$

Từ (3.4) ta tính được giá tốc của link 2:

$$\ddot{q}_2 = \frac{-D_{21}\ddot{q}_1 - C_{21}\dot{q}_1 - G_2}{D_{22}} \quad (3.7)$$

Thay (3.5) vào (3.6) ta được :

$$\bar{D}_{11}\ddot{q}_1 + \bar{C}_{11}\dot{q}_1 + \bar{C}_{12}\dot{q}_2 + \bar{G}_1 = \tau_m \quad (3.8)$$

Trong đó:

$$\begin{aligned} \bar{D}_{11} &= D_{11} - \frac{D_{12}D_{211}}{D_{22}} & \bar{C}_{11} &= C_{11} - \frac{D_{12}C_{21}}{D_{22}} \\ \bar{C}_{12} &= C_{12} & \bar{G}_1 &= G_1 - \frac{D_{12}G_2}{D_{22}} \end{aligned} \quad (3.9)$$

Bộ điều khiển vòng trong tuyền tính hóa q_1 được thiết kế như sau:

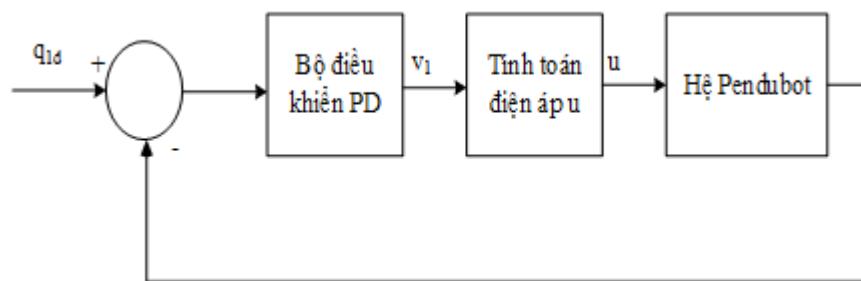
$$\tau_m = \bar{D}_{11}v + \bar{C}_{11}\dot{q}_1 + \bar{C}_{12}\dot{q}_2 + \bar{G}_1 \quad (3.10)$$

Hệ thống được mô tả lại như sau:

$$\begin{aligned} \ddot{q}_1 &= v \\ D_{22}\ddot{q}_2 + C_{21}\dot{q}_1 + G_2 &= -D_{21}v \end{aligned} \quad (3.11)$$

Thiết kế bộ điều khiển vòng ngoài để bám theo quỹ đạo của link 1 sử dụng bộ điều khiển PD hồi tiếp giá tốc của link 1:

$$v = K_D(\dot{q}_{1d} - \dot{q}_1) + K_P(q_{1d} - q_1) \quad (3.12)$$



Hình 3.11 Sơ đồ điều khiển swingup [4].

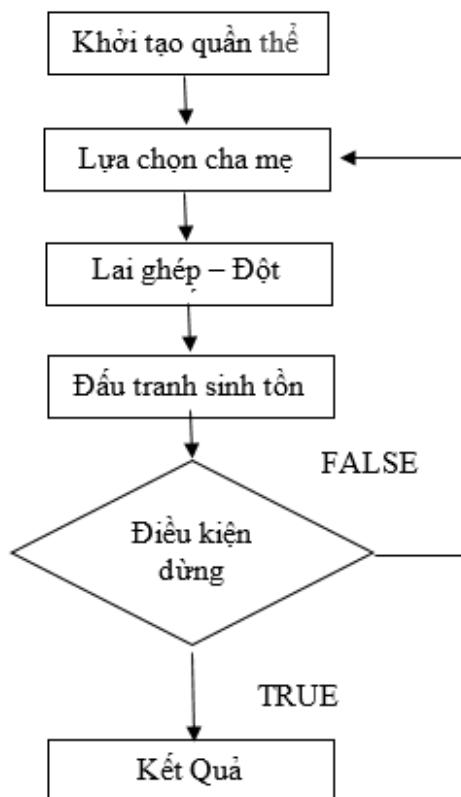
3.4 Kết luận về bộ điều khiển

- Qua quá trình tiến hành mô phỏng và khảo sát. Chúng tôi chọn bộ điều khiển PD-Fuzzy. Đối với bộ điều khiển nhóm đã thực hiện qua các bộ điều khiển như: PID, PD, PI và nhóm nhận thấy bộ điều khiển PD phù hợp với hệ thống .Bộ điều khiển PD giúp hệ thống dễ dàng thực hiện mô phỏng và điều khiển.
- Nhóm tiến hành thêm bộ mô phỏng PD-Fuzzy để khảo sát thêm kiến thức đã học trong thời gian qua. Đồng thời so sánh kết quả của từng bộ điều khiển với nhau.

CHƯƠNG 4. THIẾT KẾ CHƯƠNG TRÌNH MÔ PHỎNG TRÊN MATLAB/SIMULINK

4.1 Giới thiệu Giải thuật di truyền [8,127-130]

- Trong quá trình tìm các thông số điều khiển, nhóm nhận thấy cần rất nhiều thời gian để tìm các thông số phù hợp và điều khiển tốt. Vì vậy, nhóm đã thông qua việc sử dụng giải pháp Giải thuật di truyền để tìm các thông số một cách nhanh chóng và hiệu quả hơn.
- *Giải thuật di truyền (Genetic Algorithm – GA)* được Holland đưa ra năm 1975 là giải thuật tìm kiếm lời giải tối ưu trên nguyên tắc phỏng theo quá trình tiến hóa và quy luật di truyền của sinh vật trong tự nhiên. Bản chất toán học của GA là *giải thuật tìm kiếm theo xác suất*.
- *GA là phương pháp tối ưu hóa ngẫu nhiên tổng quát dùng để giải các bài toán tìm kiếm.* GA có một số khác biệt so với giải thuật tối ưu hóa và tìm kiếm thông thường. Thứ nhất, GA làm việc với quần thể gồm nhiều NST (tập hợp nhiều lời giải), tìm kiếm nhiều điểm cực trị cùng một lúc. Bằng các phép toán di truyền, GA trao đổi thông tin giữa những đỉnh cực trị, do đó hạn chế khả năng rơi vào cực trị cục bộ. Nói cách khác, GA là phương pháp tìm kiếm lời giải tối ưu toàn cục. Thứ hai, GA làm việc với chuỗi ký hiệu (chuỗi NST) chứ không làm việc với các thông số. Thứ ba, GA chỉ cần đánh giá hàm mục tiêu để định hướng quá trình tìm kiếm mà không cần bất cứ thông tin gì khác (như đạo hàm...). Do các đặc điểm trên, GA có khả năng giải quyết rất hiệu quả các bài toán tìm cực trị hàm phi tuyến, không khả vi, không liên tục...
- Giải thuật di truyền (GA) bắt nguồn từ ý niệm tiến hóa để tồn tại và phát triển trong tự nhiên. GA là phương thức giải quyết vấn đề bắt chước lối hành xử của con người để tồn tại và phát triển. Nó giúp tìm ra giải pháp tối ưu nhất trong điều kiện thời gian và không gian cho phép



Hình 4. 1 Sơ đồ của giải thuật GA [8]

Giải thích sơ đồ:

- Bước 1: Khởi tạo một quần thể ban đầu (Tập lời giải ban đầu của thuật toán).
- Bước 2: Tạo ra các quần thể mới bằng các phép toán duy truyền: Lai ghép chéo (crossover) từ các cá thể hiện tại có chọn lọc (selection), đột biến (mutation) các cá thể trong quần thể mới theo một xác suất nhất định.
- Bước 3: Đấu tranh sinh tồn, đánh giá độ thích nghi thông qua giá trị hàm mục tiêu (fitness) của mỗi cá thể trong quần thể. Các cá thể mới trong quần thể sinh ra được thay thế cho các cá thể trong quần thể cũ dựa trên đánh giá hàm thích nghi.
- Bước 4: Nếu điều kiện dừng thỏa mãn thì giải thuật dừng lại và trả về cá thể tốt nhất cùng với giá trị hàm mục tiêu của nó, nếu không thì quay lại bước 2

4.2 Giới thiệu Anfis

Theo tác giả [9] ta có:

- Khái Niệm:

ANFIS xuất phát từ tiếng Anh là Adaptive neuro-fuzzy inference system. Sử dụng tập dữ liệu vào/ra có sẵn, hàm anfis xây dựng nên hệ thống suy diễn mờ (FIS), các thông số hàm liên thuộc của nó được điều chỉnh nhờ sử dụng các thuật toán 62 huấn luyện của mạng nơron như thuật toán lan truyền ngược hoặc kết hợp lan truyền với phương pháp bình phương cực tiểu cho phép hệ mờ của ta "học" từ tập dữ liệu chúng được mô hình hóa.

- Cấu trúc và sự điều chỉnh tham số của FIS một kiểu mạng có cấu trúc tương tự mạng nơron, nó ánh xạ các đầu vào qua các hàm liên thuộc vào với các thông số tương ứng và sau đó là thông qua các hàm ra với các tham số tương ứng tạo nên các đầu ra có thể được sử dụng để giải thích ánh xạ vào/ra. Các thông số tương ứng với hàm liên thuộc sẽ thay đổi thông qua quá trình học. Việc tính toán các tham số này (hoặc việc điều chỉnh chúng) thực hiện dễ dàng bằng vec tơ gradient nó đưa ra giới hạn theo cách tốt cho hệ thống suy diễn mờ được mô hình hóa dữ liệu vào/ra theo tập các tham số nhất định. Ta đã biết, vec tơ gradient được áp dụng cho một vài thủ tục tối ưu cốt để điều chỉnh các tham số sao cho giảm nhỏ giá trị sai số (thường được định nghĩa bằng tổng bình phương sai lệch giữa đầu ra hiện thời và đầu ra mong muốn). Anfis sử dụng điều đó theo giải thuật lan truyền ngược hoặc kết hợp sự ước lượng bình phương cực tiểu và sự lan truyền ngược cho sự ước lượng tham số hàm liên thuộc.

- Các hạn chế của Anfis:

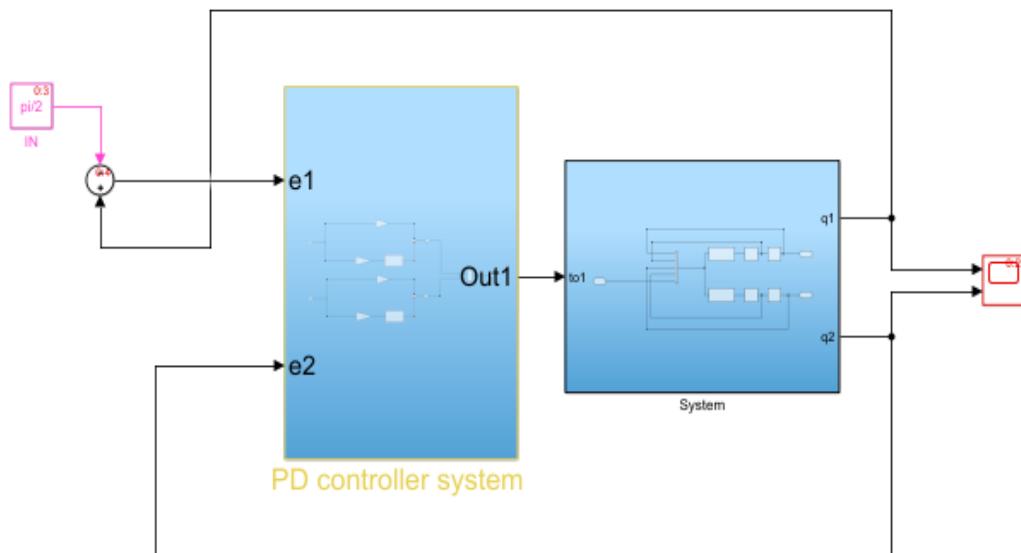
Anfis phức tạp hơn các hệ thống suy luận mờ mà chúng ta đã đề cập ở chương 1 rất nhiều, và cũng không sẵn có như các tùy chọn của hệ thống suy luận mờ. Đặc biệt, anfis chỉ hỗ trợ cho các hệ thống mờ theo mô hình Sugeno và chúng cần cài đặt những ràng buộc sau:

- ❖ Là các hệ thống loại Sugeno ở vị trí 0 hoặc 1.
- ❖ Có một đầu ra đơn, giải mờ bằng phương pháp trung bình trọng tâm. Tất cả các hàm liên thuộc đầu ra phải cùng loại, hoặc tuyến tính hoặc bất biến.
- ❖ Không chia sẻ luật điều khiển. Các luật khác nhau không thể chia sẻ cùng một hàm liên thuộc đầu ra, cụ thể là số lượng các hàm liên thuộc đầu ra phải bằng số lượng các luật.
- ❖ Có một trọng lượng nhất định (đồng nhất) cho mỗi một nguyên tắc. Khi không train đủ những ràng buộc trên, cấu trúc FIS sẽ bị sai số.

Hơn nữa, anfis không thể chấp nhận các tùy chọn thông thường mà suy luận mờ cơ bản cho phép. Vì vậy chúng ta không thể tùy ý tạo ra các hàm liên thuộc và các phương pháp giải mờ của mình mà phải sử dụng những chức năng đã cho.

4.3 Mô Phỏng bộ điều khiển PD

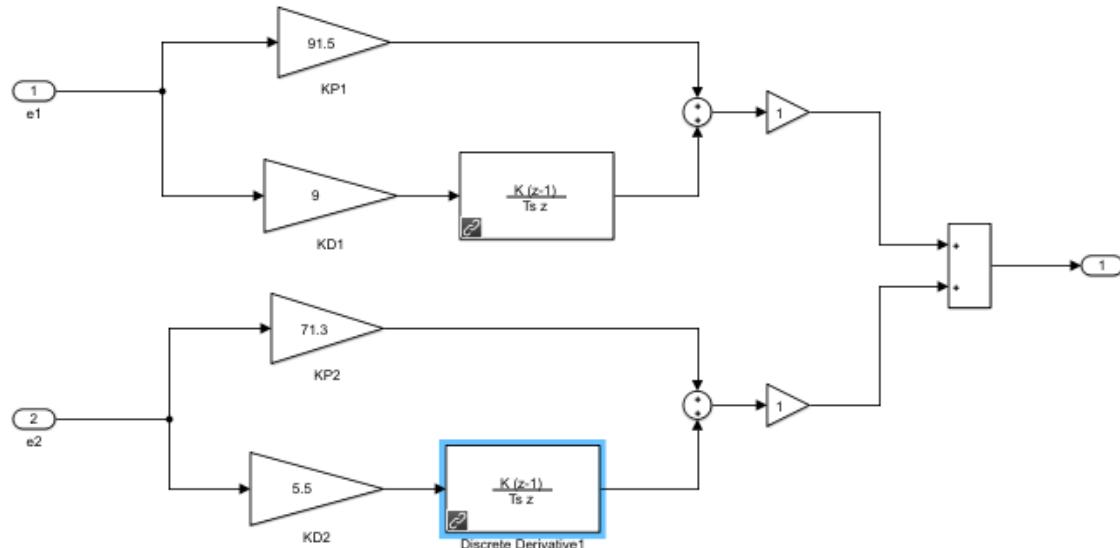
Dựa vào phương trình toán và thuật toán của bộ điều khiển PD như sau



Hình 4.2 Mô phỏng bộ điều khiển PD.

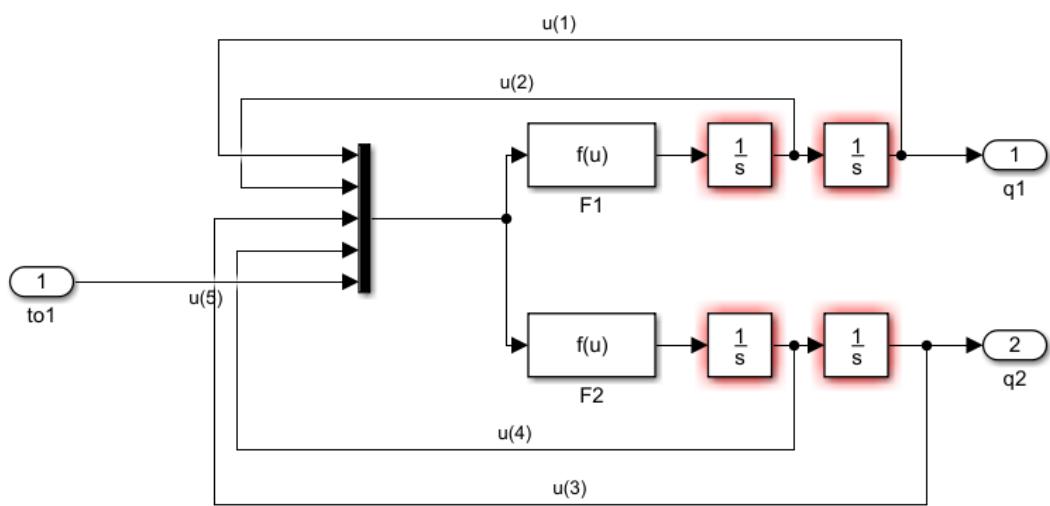
Chương trình mô phỏng được thể hiện trong

Hình 4.2, trong đó khối “System” chứa chương trình mô phỏng hệ thống theo phương trình() $-()$. Chương trình trong khối “PD controller system” chứa bộ điều khiển PD bao gồm các thông số điều khiển. Sau khi dùng giải thuật GA, nhóm thu được các thông số điều khiển như sau: $K_p1 = 91.5$, $K_d1 = 9$, $K_p2 = 71.3$, $K_d2 = 5.5$.



Hình 4.3 Khối điều khiển PD

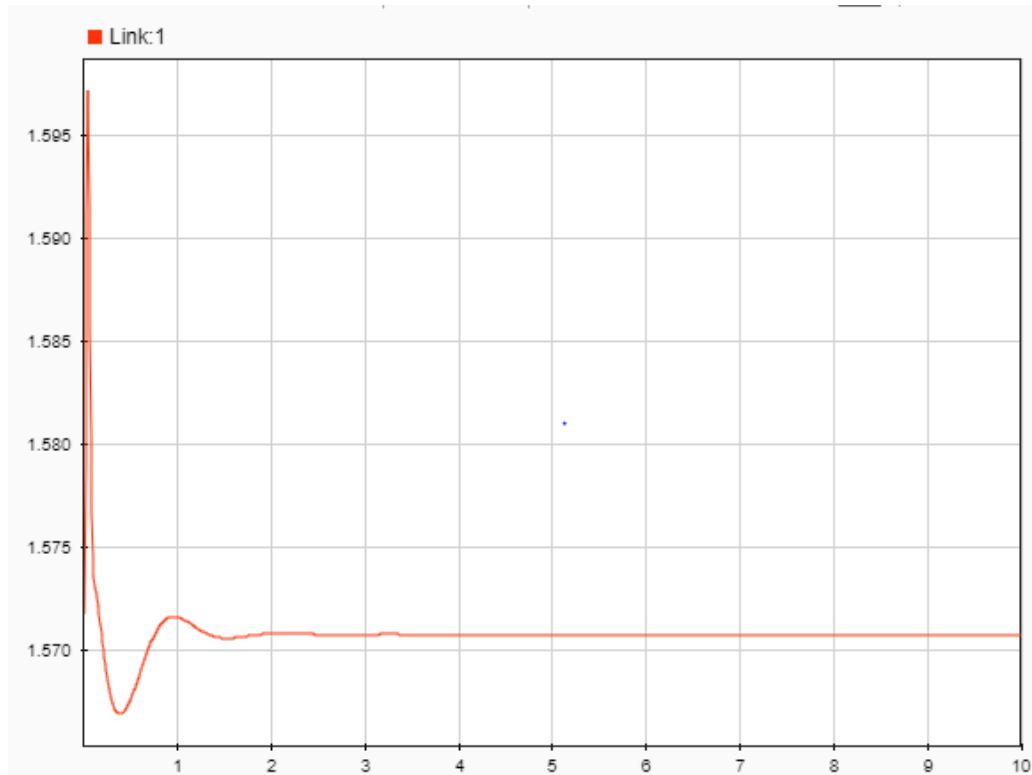
Trong khối System chứa phương trình toán của hệ thống:



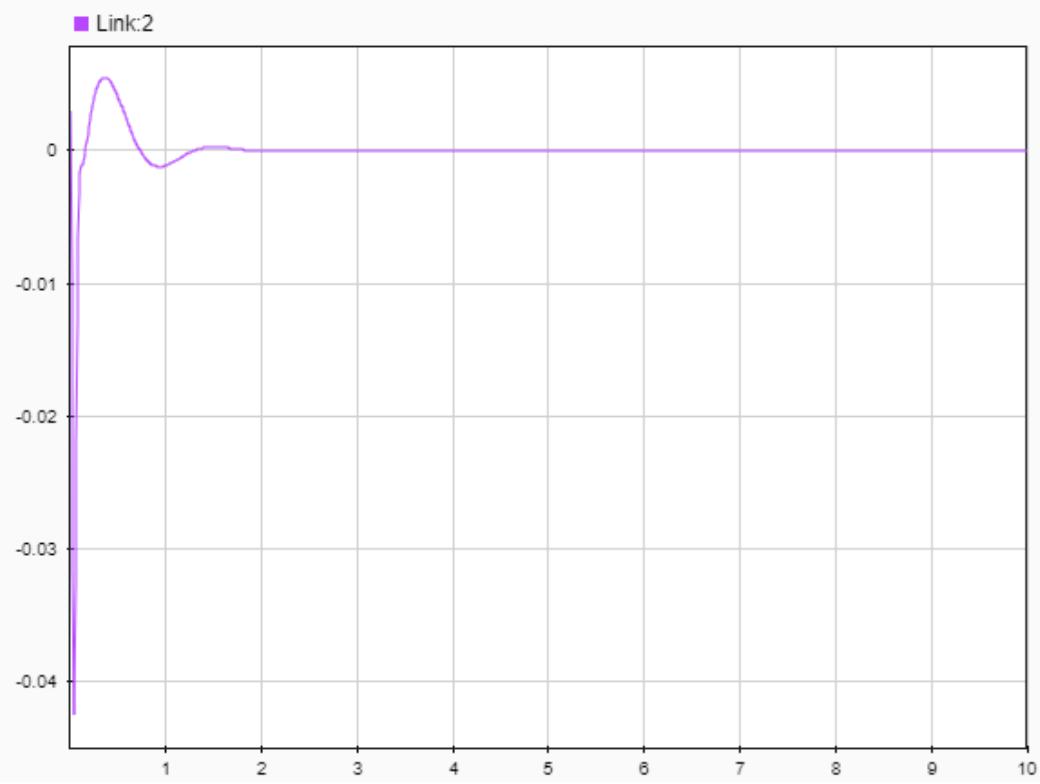
Hình 4.4 Khối phương trình toán của hệ thống

Trong đó 2 khối F1 với F2 là phương trình toán của hệ thống

Dựa vào bộ mô phỏng trên ta thu được kết quả như sau:



Hình 4.5 Kết quả mô phỏng link 1 của bộ điều khiển PD trong 10 giây

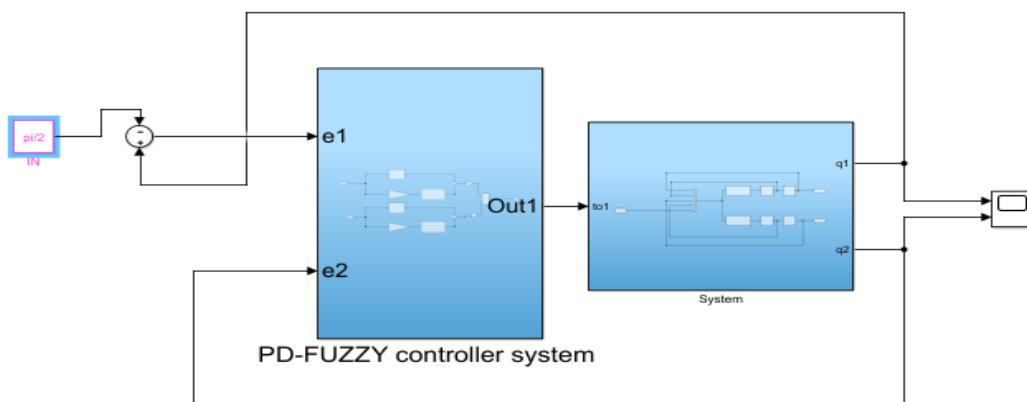


Hình 4.6 Kết quả mô phỏng link 2 của bộ điều khiển PD trong 10 giây

Với bộ mô phỏng trên, ta thu được kết quả như Hình 4.5, sau 2.36 giây link1 ổn định tại vị trí 1.571(rad). Và trong quá trình điều khiển link1 giao động trong khoảng từ 1.566- 1.598 (rad).

Theo hình Hình 4.6, link2 giao động trong khoảng từ -0.045 – 0.007(rad). Và sau 1.3 giây link2 ổn định. Và cách đặt vị trí link1 theo phuong thẳng đứng hướng lên và vị trí link2 theo phuong của link1. Dựa vào kết quả mô phỏng, hệ thống đã ổn định được với bộ điều khiển PD ở vị trí top.

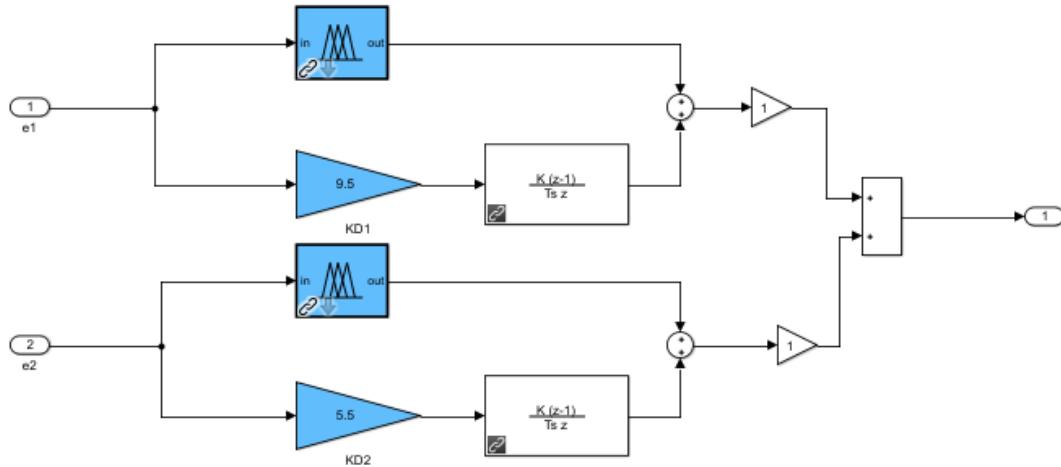
4.4 Mô phỏng bộ điều khiển PD-Fuzzy



Hình 4.7 Mô phỏng bộ điều khiển PD-Fuzzy.

Chương 4. Thiết Kế Chương Trình Mô Phỏng Trên Matlab/Simulink

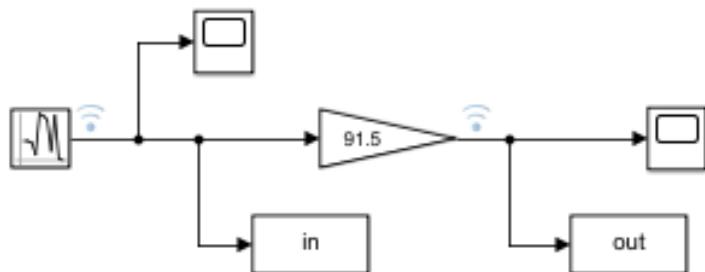
Bộ điều khiển PD-Fuzzy tương tự với bộ điều khiển PD, nhưng thông số Kp1 với Kp2 sẽ được thay bằng khối fuzzy như Hình 4.7. Bộ điều khiển này được tạo ra bởi khối Editanfis trong bộ điều khiển Matlab. Dựa vào sai số đầu vào nhân với thông số kp1 và kp2 để có được kết quả đầu ra.



Hình 4.8 Các khối điều khiển trong bộ PD-Fuzzy.

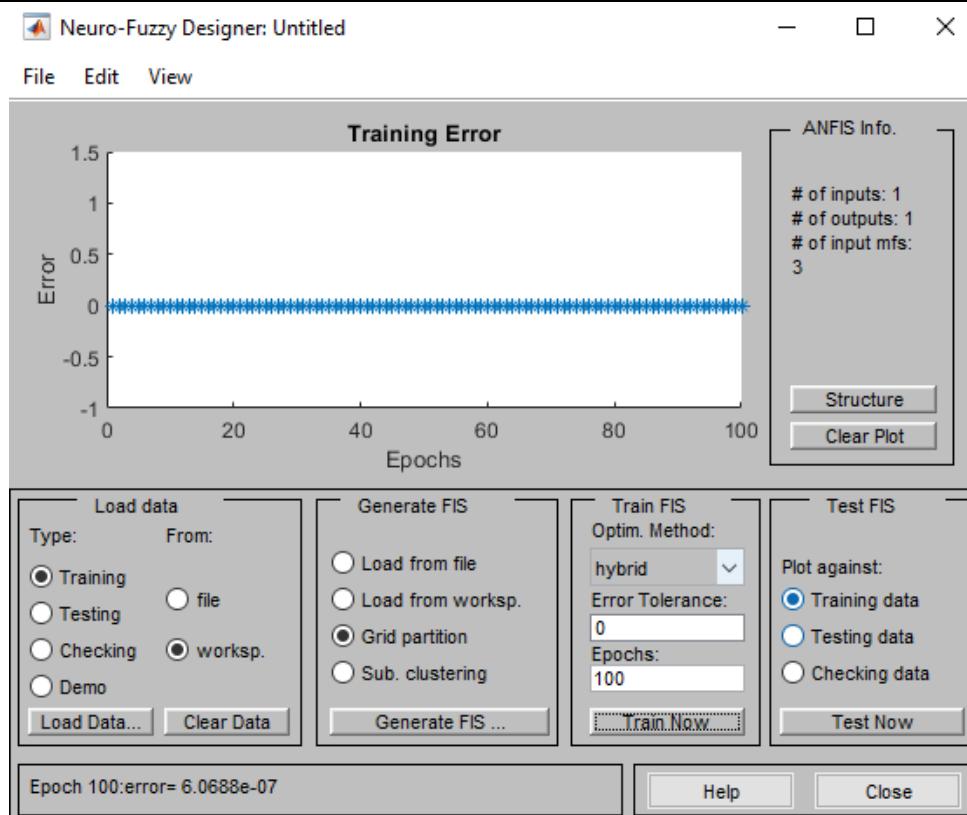
- Các bước tạo khái Fuzzy từ hàm *editanfis*:

Bước 1: Tạo ramdom tín hiệu đầu vào dựa vào thông số Kp1 để tạo tín hiệu đầu ra:



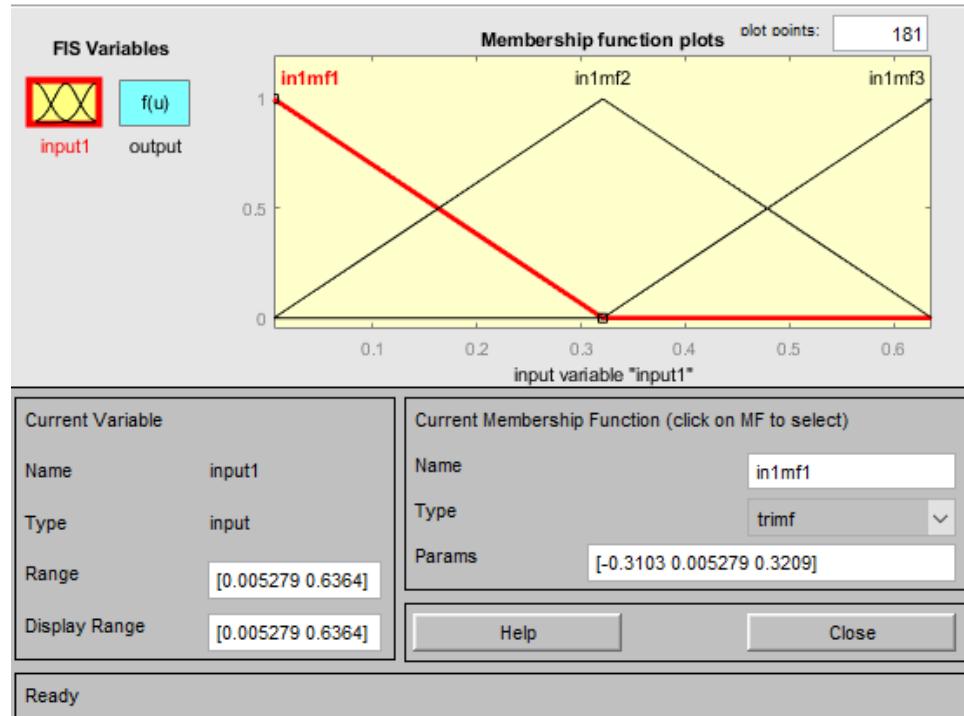
Hình 4.9 Tạo hiệu cho khái fuzzy.

Sau khi chạy bộ mô phỏng này ta sẽ thu được kết quả 2 giá trị là in và out. Sau đó 2 giá trị này sẽ được tạo thành ma trận và được đưa vào khái anfisedit để tạo thành bộ fuzzy cho thông số kp1. Và với cách tương tự ta sẽ thu được kết quả cho bộ fuzzy của kp2

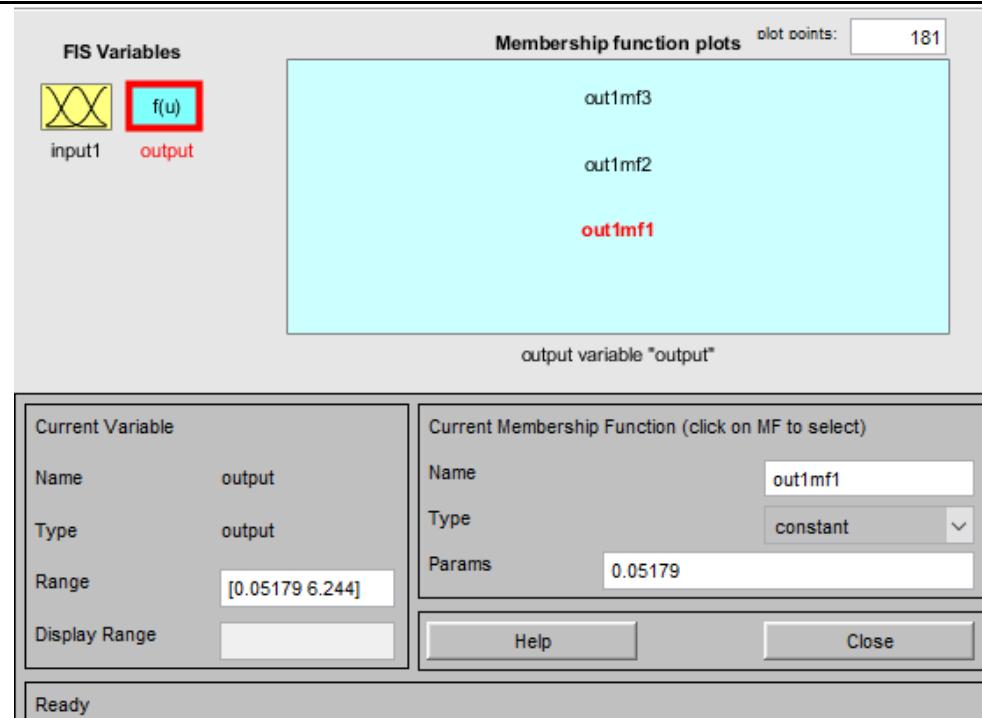


Hình 4.10 Khối anfisedit.

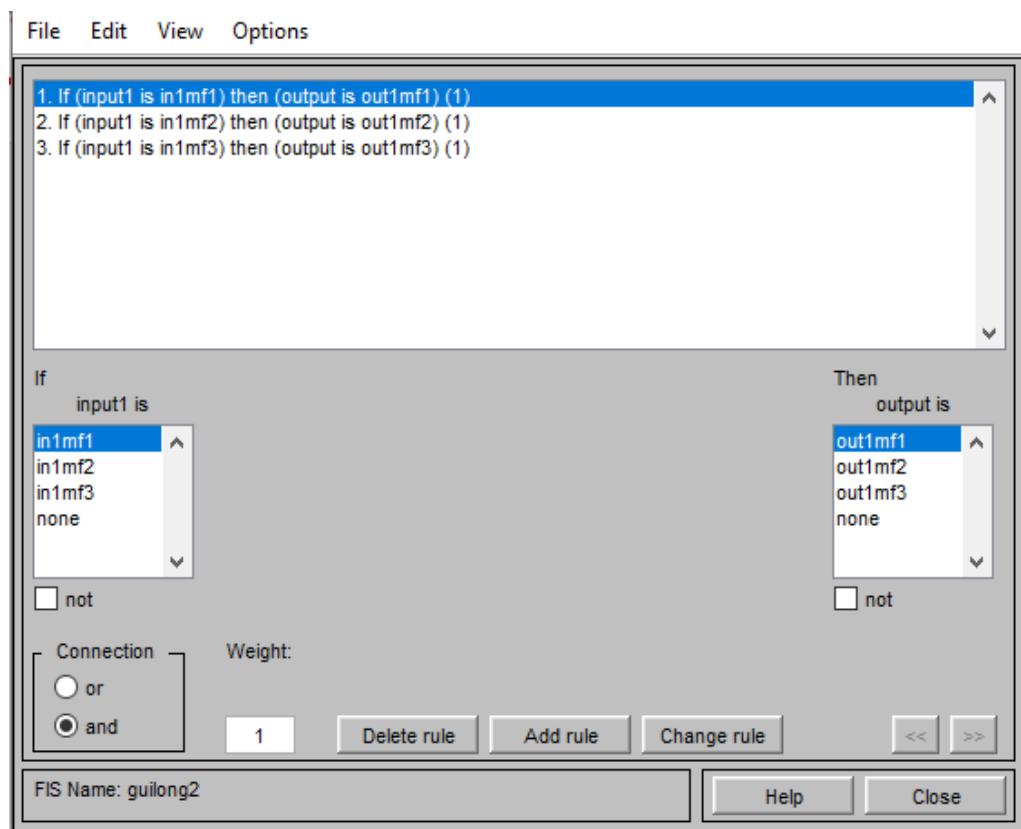
Sau khi tạo xong, chúng ta lưu lại thì sẽ có một khối fuzzy:



Hình 4.11 Input của khối Fuzzy sau khi được tạo

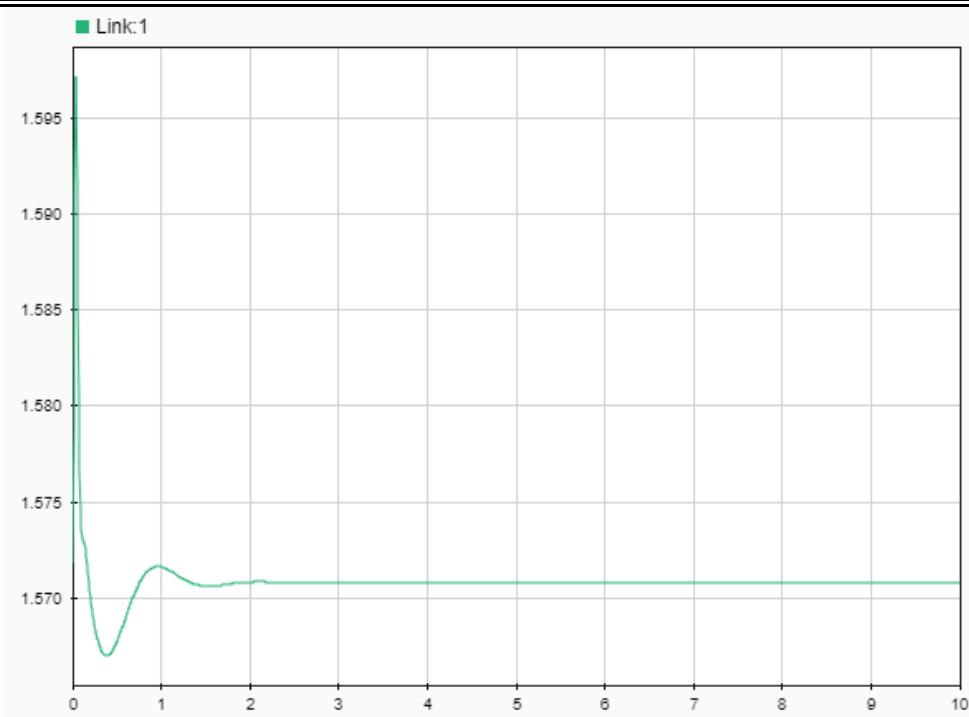


Hình 4.12 Output của khôi Fuzzy sau khi được tạo

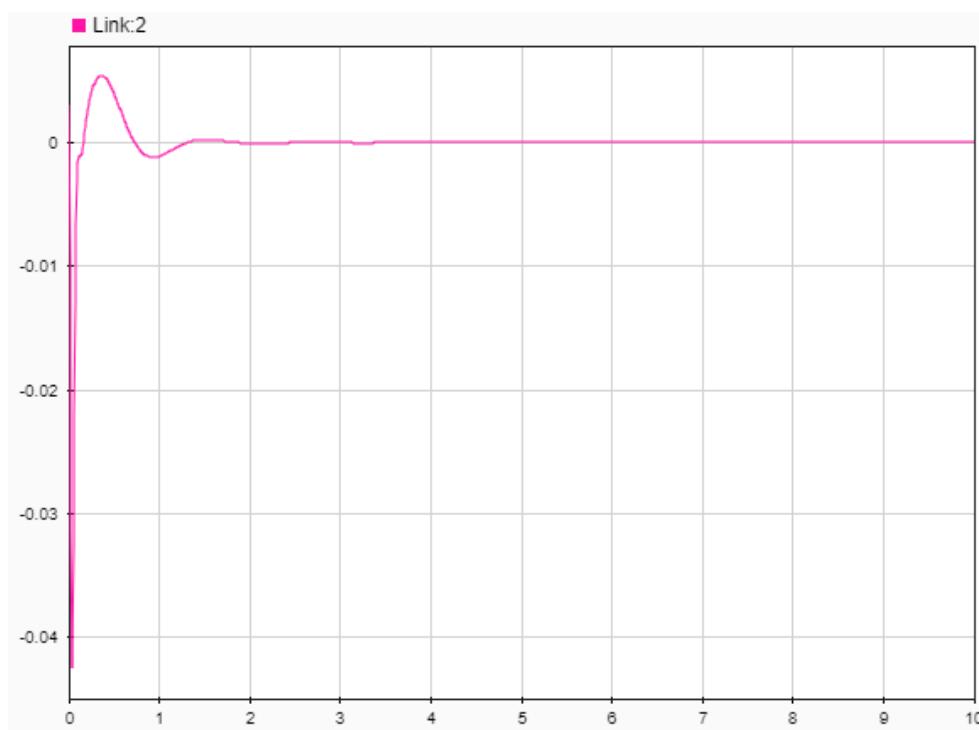


Hình 4. 13 Các luật mờ

- Kết quả của bộ mô phỏng PD-Fuzzy:



Hình 4.14 Kết quả link1 của bộ mô phỏng PD-Fuzzy trong 10 giây



Hình 4.15: Kết quả link2 của bộ mô phỏng PD-Fuzzy trong 10 giây

Ở Hình 4.14, ta thu được kết quả mô phỏng PD-Fuzzy trong 10 giây. Trong khoảng thời gian đó link1 giao động trong khoảng từ 1.566- 1.598 (rad). Và sau khoảng 2.37 giây link1 ổn định tại vị trí 1.571(rad) theo phương thẳng đứng.

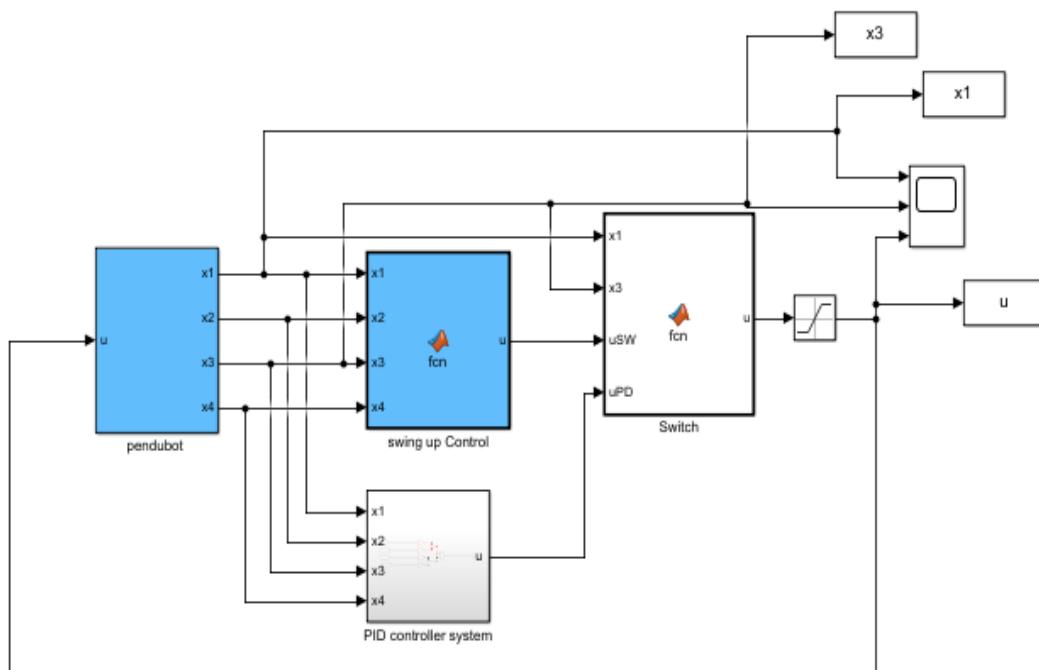
Đối với kết quả mô phỏng PD-Fuzzy của link2 trong Hình 4.15, ta thu được kết quả như sau : Trong 10 giây link2 giao động trong khoảng từ -0.045-0.007(rad), và sau 1.3 giây link2 ổn định tại 0(rad) theo phương link1.

Và ta nhận thấy được, bộ điều khiển PD-Fuzzy đã điều khiển ổn định được hệ thống.

Kết Luận: Hai bộ điều điều khiển đều điều khiển được hệ thống ổn định, mà bộ điều khiển PD-Fuzzy được học và xây dựng dựa trên cơ sở và số liệu của bộ điều khiển PD, nên kết quả thu được của hai bộ điều khiển gần như giống nhau. Thông qua các kết quả thu được ta kết luận được điều đó

4.5 Mô Phỏng bộ điều khiển Swing up và bộ điều khiển PD

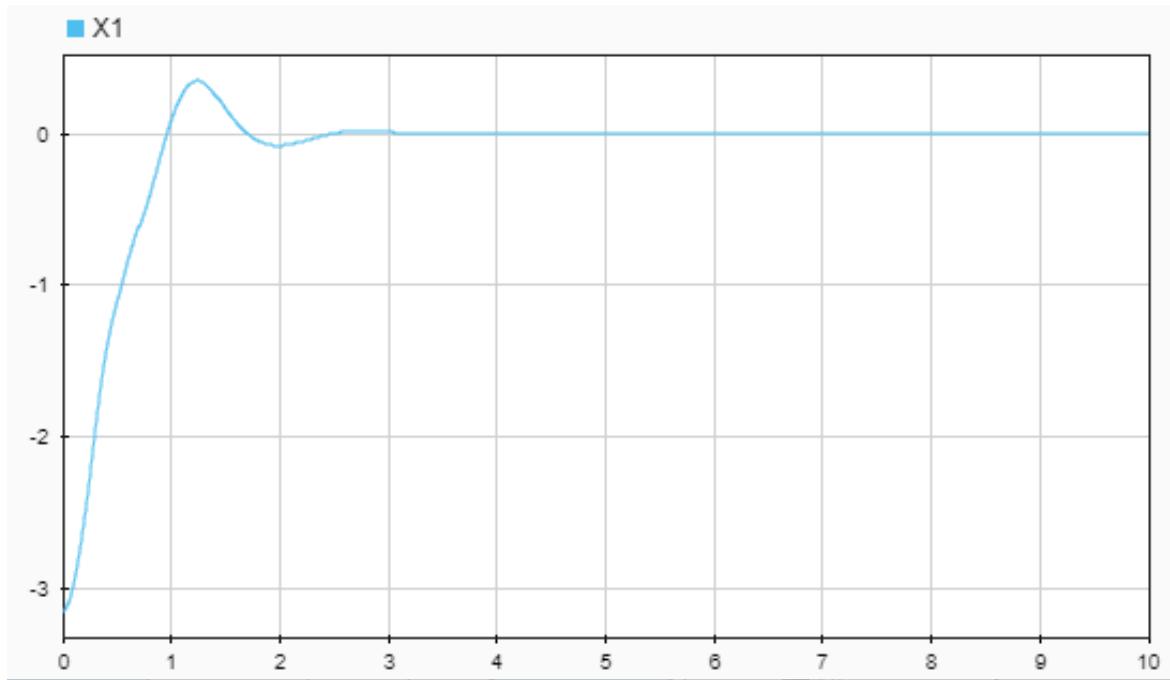
Theo công thức (3.8) và (3.9) Ta mô phỏng bộ điều khiển Swingup và PD:



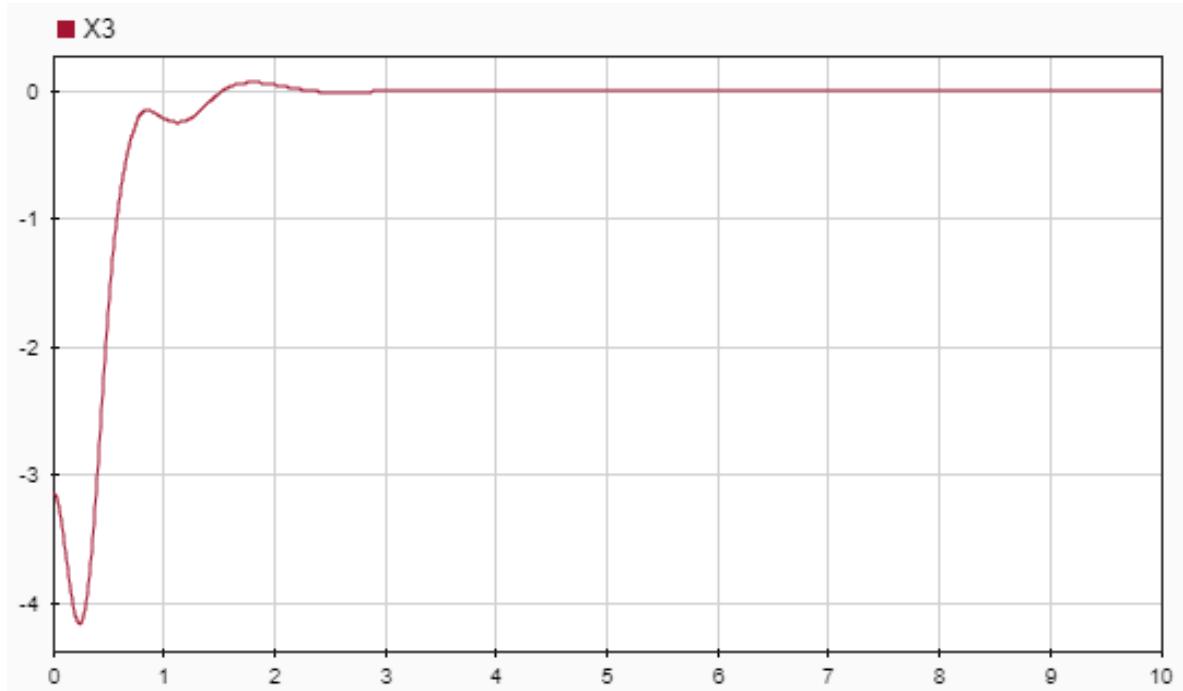
Hình 4.16 Chương trình mô phỏng bộ điều khiển Swingup và PD

Khối mô phỏng được thể hiện theo Hình 4.16. Trong đó khối “pendubot” chứa phương trình mô tả hệ thống (2.23)-(2.25), khối "swing up Control" chứa công thức tính toán để đưa hai link của pendubot từ vị trí downward (cả hai link hướng xuống) đến lân cận vị trí upright (cả hai link hướng lên) theo công thức (3.1) và (3.2). Khối "PD controller system" chứa các thông số kp và kd để điều khiển hệ thống ổn định hơn. Khối "Switch" lựa chọn tín hiệu điều khiển dựa vào hai biến trạng thái là x_1 & x_3 . Khối "Scope" quan sát biến trạng thái và tín hiệu điều khiển theo thời gian.

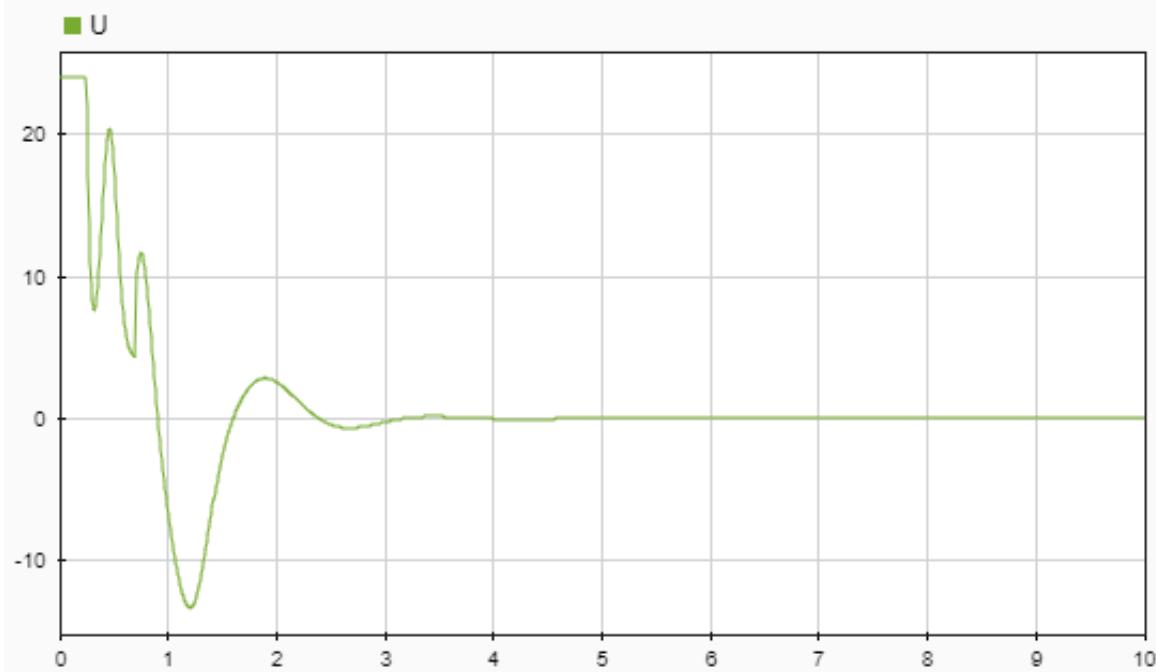
Kết quả của mô phỏng Swingup và PD:



Hình 4.17 Kết quả mô phỏng vị trí link1 so với phương thẳng đứng trong 10 giây



Hình 4.18 Kết quả mô phỏng vị trí link2 so với phương link1 trong 10 giây



Hình 4.19 Điện áp U cần cấp cho động cơ trong 10 giây

Trong Hình 4.17, link1 đã ổn định sau 2.5 giây, trong thời gian điều khiển link1 hoạt động trong khoảng từ $-3.1 - 0.3$ (rad) và sau khi điều khiển đã ổn định tại vị trí cân bằng là 0(rad).

Kết quả mô phỏng link2 trong Hình 4.18, đã cho ta thấy sau khoảng 2.4 giây thì link 2 đã được điều khiển ổn định. Đồng thời vị trí ban đầu chưa được điều khiển -4.1 (rad) và sau điều khiển thì dần trở về 0(rad).

Điện áp U cần để cấp cho hệ thống hoạt động được là 24(v), và ta nhận thấy được điều đó qua Hình 4.19.

Kết Luận: Thông qua các kết quả trên, bộ mô phỏng Swingup và PD là một bộ điều khiển khả thi và có thể điều khiển thực nghiệm. Tuy nhiên, về phần cứng của nhóm không đáp ứng đủ nguồn cung cấp nên bộ điều khiển này chỉ làm được mô phỏng. Và vì thời gian không còn nhiều nên nhóm không thể thay đổi phần cứng để đáp ứng nhu cầu làm bộ điều khiển trê

CHƯƠNG 5. THỰC NGHIỆM VÀ KẾT QUẢ

5.1 Giới thiệu chung

Mô hình pendubot khá đơn giản về phần cứng các bộ phận chính chủ yếu là động cơ, encoder và các khớp. Mặc dù đã được thày phụ trách hỗ trợ cho mượn phần cứng tuy nhiên nhóm vẫn thiết kế lại để hiểu rõ hơn về mô hình cũng như lấy các thông số tốt hơn cho bộ điều khiển mà nhóm thiết kế.

5.2 Phần Cứng

- ✓ Mô hình Pendubot được sinh viên xây dựng gồm có:
- ✓ Khớp 1 dài 15 cm được cắt bằng mica dày 10 mm, phía cuối link 1 được gắn Encoder có độ phân giải 600 xung.
- ✓ Khớp 2 dài 25 cm được cắt bằng mica dày 5 mm, được kết nối với khớp 1 thông qua Encoder.
- ✓ Động cơ DC 24V Nisca NF5475 được gắn Encoder có độ phân giải 200 xung.
- ✓ Tủ điện hình hộp chữ nhật để chứa các linh kiện như mạch cầu H, bộ nguồn, board mạch ...

5.2.1 Vi xử lý STM32F4

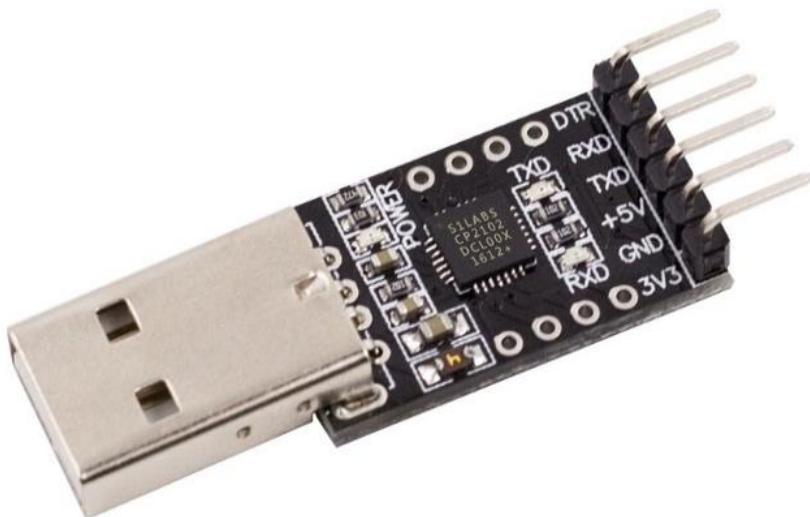


Hình 5. 1 Vi xử lý STM32F407 [6]

Kit STM32F407 Discovery hiện là loại kit được sử dụng ở rất nhiều trường đại học hiện nay trong giảng dạy vi điều khiển ARM, nếu so sánh về ngoại vi và sức mạnh của STM32 so với các dòng ARM của các hãng khác thì ở cùng 1 tầm giá, ARM của ST vượt trội về cấu hình và ngoại vi hơn rất nhiều.

- Vi điều khiển chính: STM32F407VGT6 microcontroller featuring 32-bit ARM Cortex-M4F core, 1 MB Flash, 192 KB RAM
- Tích hợp sẵn mạch nạp và Debug ST-LINKA/2
- Nguồn cấp từ cổng Mini USB qua các IC nguồn chuyển thành 3V3 để cấp cho MCU
- Có sẵn các chân nguồn: 3V và 5V

5.2.2 Mạch chuyển USB UART CP2102

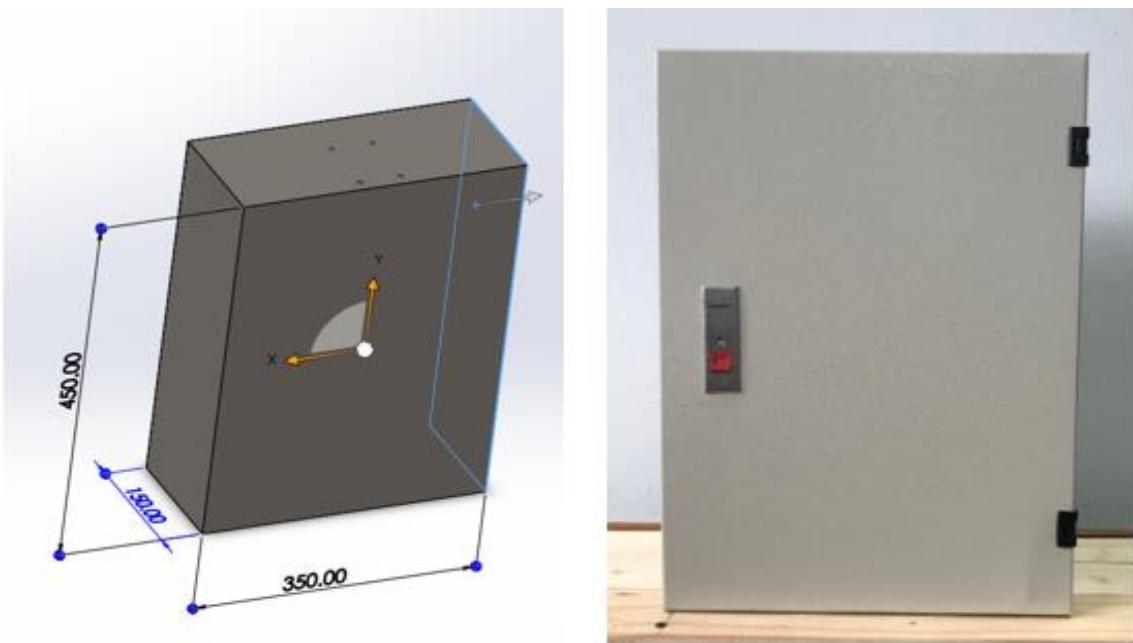


Hình 5. 2 USB UART *CP2102* [6]

Mô tả chân:

- ✓ TXD: chân truyền dữ liệu UART TTL (3.3VDC), dùng kết nối đến chân nhận RX của các module sử dụng mức tín hiệu TTL 3.3~5VDC.
- ✓ RXD: chân nhận dữ liệu UART TTL (3.3VDC), dùng kết nối đến chân nhận TX của các module sử dụng mức tín hiệu TTL 3.3~5VDC.
- ✓ GND: chân mass hoặc nối đất.
- ✓ 5V: Chân cấp nguồn 5VDC 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 3.3VDC (dòng cấp rất nhỏ tối đa 100mA), không sử dụng để cấp nguồn, thường chỉ sử dụng để thiết đặt mức tín hiệu Logic.

5.2.3 Tủ điện



Hình 5. 3 Tủ điện

Khung mô hình ở đây có thông số dựa trên hộp tủ điện có sẵn trên thị trường với kích thước. Vỏ tủ điện nối EMD 45x35x15cm sơn tĩnh điện trong công nghiệp Sử dụng trong công nghiệp hoặc xây dựng dân dụng, lắp đặt và bảo vệ các thiết bị khí cụ điện có trong tủ, tránh việc nắng mưa có thể xâm nhập làm hư hỏng ác thiết bị bên trong.

Thông số của tủ điện nối EMD:

- + Kích thước: 43x35x15cm
- + Màu sắc: màu xám
- + Vỏ tủ được sơn tĩnh điện và tay chốt đỏ mở khóa tủ ra bên ngoài.

Tủ được lắp đặt với thiết kế phù hợp cho việc điều khiển các thiết bị, che nắng mưa nhưng quý khách xin lưu ý trong quá trình sử dụng về không gian và lắp đặt tủ sao cho phù hợp với nhu cầu công việc của mỗi người là khác nhau, để tránh trường hợp khi mua tủ điện nối công nghiệp EMD không vừa không gian sử dụng.

5.2.4 Lựa chọn động cơ

Động cơ servo NISCA NF5475 encoder 200ppr là loại động cơ DC Brushed (Động cơ có chổi than) có thêm cơ cấu Encoder gắn liền phía sau của động cơ.



Hình 5. 4 Động cơ servo NISCA NF5475

Thông số kỹ thuật động cơ:

- Điện áp: 12-24V DC
- Công suất: 32W
- Dòng tải: 1.49A
- Dòng không tải: 0.27A
- Dòng khởi động: 8.264 A
- Moment khởi động: 464mN*m
- Moment tải: 71.07mN*m
- Tốc độ: 4500 vòng/phút

Encoder 200 xung cho độ chính xác cao. Ứng dụng làm máy CNC, điều khiển bánh xe robot, làm mô hình tự cân bằng, làm máy bắn các loại bóng bàn, tennis, cầu lông.

Thông số kích thước:

Đường kính trục động cơ: 8mm

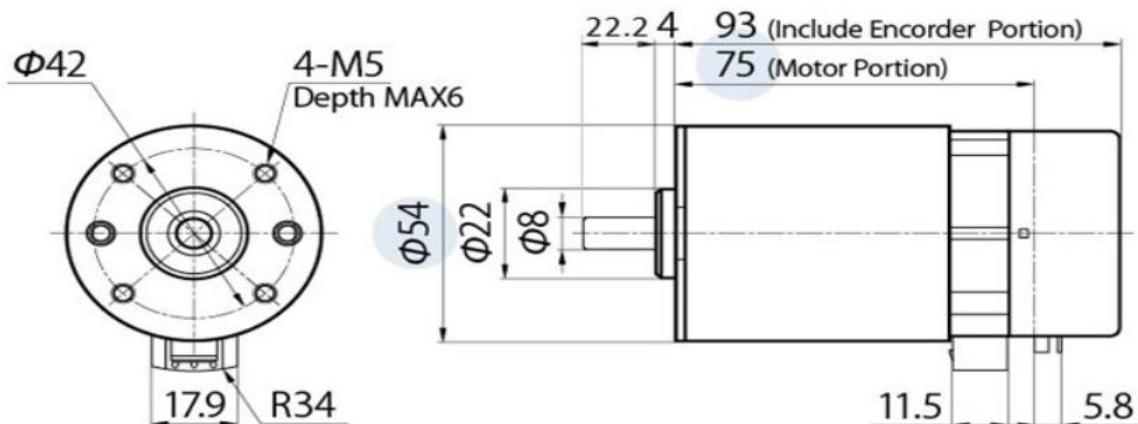
Đường kính thân động cơ: 54mm

Chiều dài động cơ không tính trực (bao gồm cả encoder): 93mm

Chiều dài không bao gồm encoder: 75mm

Lỗ mặt bích động cơ: Lỗ ren M5

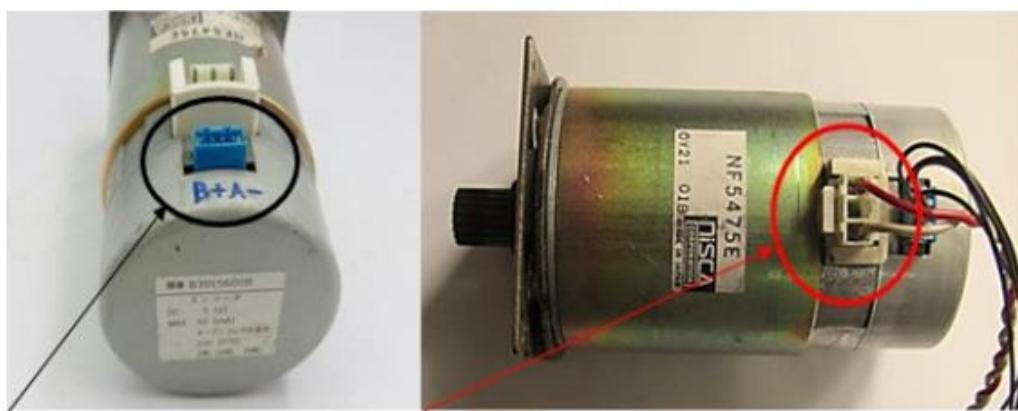
Chi tiết xem hình sau:



Hình 5. 5 Kích thước động cơ [6]

Thông số Encoder:

- Điện áp: 5VDC
- Số lượng dây :4 dây (4 wires)
- Số lượng kênh: 2 kênh A, B (A, B pluse channel)
- Tốc độ phản hồi: 20 KHz
- Phân giải xung: 200ppr (200 xung/vòng)



Bô trí dây Encoder:

(+) +5V Vcc

(A) Kênh xung A

(B) Kênh xung B

(-) 0V GND

Bô trí dây nguồn cấp động cơ

(Dây đỏ) VDD 12V hoặc 24 V

(Dây trắng) GND 0V

Hình 5. 6 Vị trí encoder và bô trí dây

Lý do chọn động cơ:

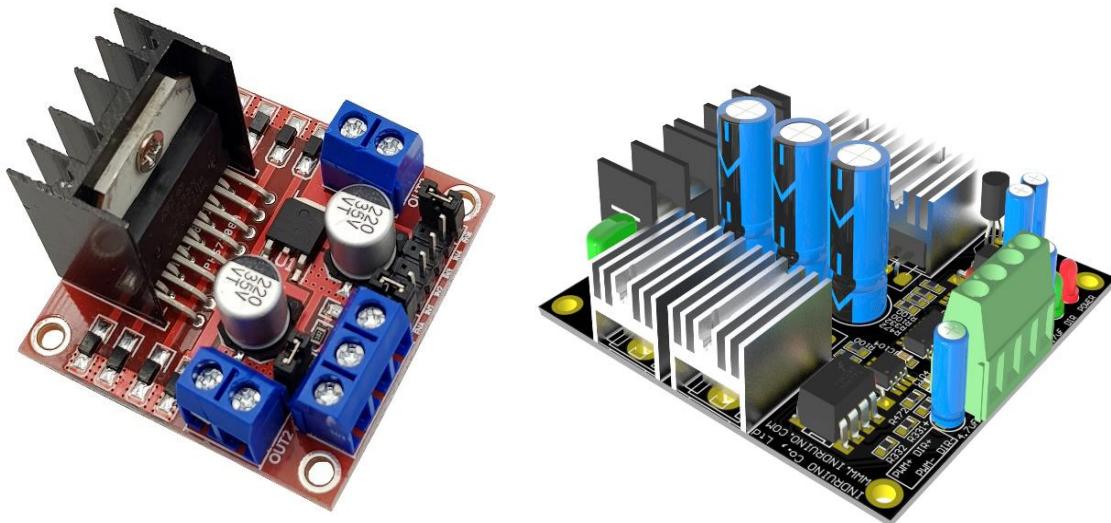
- Động cơ có tích hợp sẵn encoder 200 xung.

- Độ ổn định tốt.
- Độ nhiễu gây ra bởi động cơ lên mạch điều khiển thấp.
- Có tính chính xác cao.
- Chất lượng xung Encoder A, B hoàn hảo.
- Motor không phát nóng nhiều trong quá trình sử dụng.
- Bền bỉ, phù hợp với điều khiển pendubot.
- Giá thành vừa phải với sinh viên.

5.2.5 Board cầu H

Mạch cầu H là một mạch đơn giản dùng để điều khiển động cơ DC quay thuận hoặc quay nghịch. Trong thực tế, có nhiều kiểu mạch cầu H khác nhau tùy vào cách chúng ta lựa chọn linh kiện có dòng điện, áp điều khiển lớn hay nhỏ, tần số xung PWM... Và chúng sẽ quyết định đến khả năng điều khiển của cầu H.

Tuy vậy trong quá trình lựa chọn và thực nghiệm nhóm tiến hành nghiên cứu hai mạch cầu H là L298 và mạch cầu H công suất IR1284 như hình ở dưới:



Hình 5. 7 Cầu H L298 và Cầu H IR2184 [7]

Mỗi mạch cầu H thì đều có ưu điểm và nhược điểm riêng ta có bảng so sánh thông số kỹ thuật cơ bản sau:

	H-Bright L298	H-Bright IR1284
Điện áp đầu vào	5~30VDC	12~36VDC
Công suất tối đa	25W 1 cầu	150W
Phương pháp điều khiển	Băm xung PWM	Băm xung PWM
Dòng tối đa	2A 1 cầu	30 A
Kích thước	43x43x27mm	72x68.5x20mm

Bảng 5.1 So sánh cơ thông số cơ bản hai mạch cầu

Từ bảng trên ta thấy cầu IR1284 có nhiều ưu điểm hơn so với mạch cầu L298, tuy nhiên đối với hệ pendubot là một hệ có cấu trúc phần cứng khá đơn giản công suất cũng như dòng không lớn thì lựa chọn L298 trong mô hình của nhóm là hợp lý hơn và trong quá trình thực nghiệm nhóm đã có kiểm tra thử cầu IR1284 thì cầu này thường xuyên xảy ra hiện tượng nhiễu khiến góc trả về không đúng. Cho nên việc lựa chọn cầu L289 là hợp lý cho hệ thống pendubot.

Giới thiệu về cầu L298:

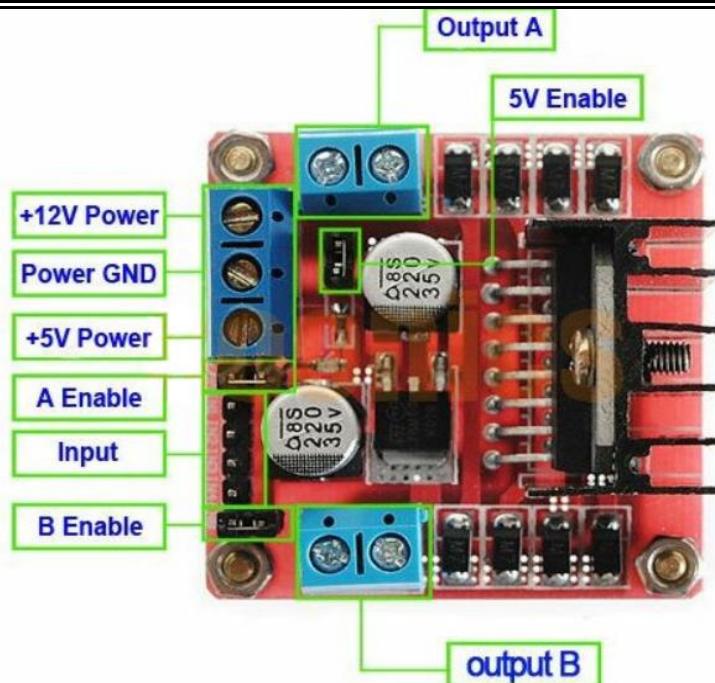
Mạch điều khiển động cơ DC L298 có khả năng điều khiển 2 động cơ DC, dòng tối đa 2A mỗi động cơ, mạch tích hợp diod bảo vệ và IC nguồn 7805 giúp cấp nguồn 5VDC cho các module khác.

Mạch điều khiển động cơ DC L298 dễ sử dụng, chi phí thấp, dễ lắp đặt, là sự lựa chọn tối ưu trong tầm giá.

Cấu tạo của module L298 bao gồm các chân sau:

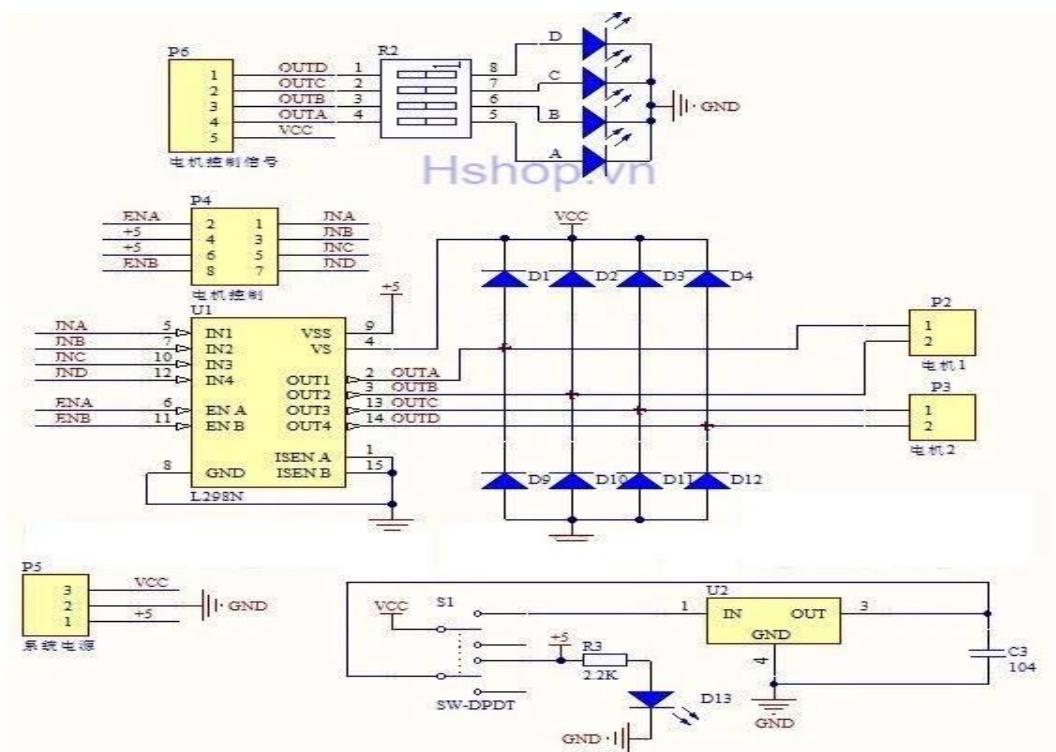
- Hai chân cấp nguồn trực tiếp đến động cơ là 12V power và 5V power. Ngoài ra bạn cũng có thể cấp nguồn 9 -12V ở 12V.
- Power GND là GND của nguồn cấp cho động cơ.
- 2 Jump A enable và B enable.
- 4 chân Input gồm có: IN1, IN2, IN3, IN4.
- Output A sẽ được nối với động cơ A. Khi nối bạn phải chú ý ký hiệu “+;-” vì khi bạn nối ngược động cơ sẽ chạy theo chiều ngược.
- Gồm hai phần điều khiển động cơ. Có thể điều khiển cho động cơ 4 dây và động cơ 6 dây.

Ở đây ta sử dụng L298 băm xung cho động cơ nên ta sử dụng chân A Enable hoặc B Enable để băm xung. Nếu dùng A Enable thì sử dụng IN1, IN2 để quy định chiều cho động cơ hoặc dùng B Enable thì dùng hai chân IN3 và IN4.



Hình 5.8 Cấu tạo cầu L298 [7]

Cấu tạo sơ đồ nguyên lý:



Hình 5.9 Cấu tạo sơ đồ nguyên lý

5.2.6 Lựa chọn Encoder



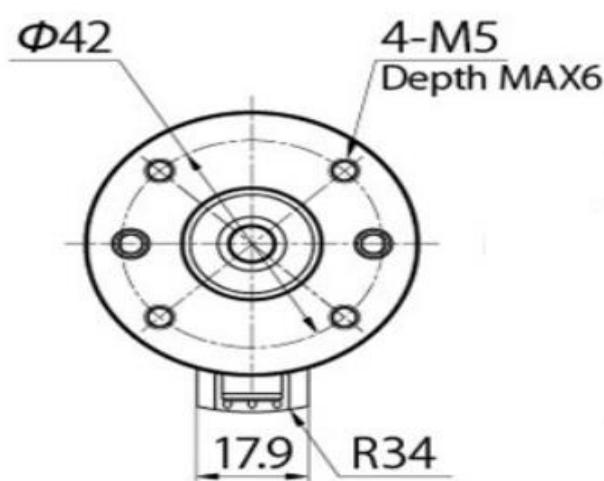
Hình 5.10 Encoder và thông tin chân ra

Encoder này dùng nhiều trong các thiết bị tự động

- Có thể dùng để đo quãng đường đi của robot tự động
- Khối lượng 100g.
- Nguồn 5VDC, hai pha A, B độ phân giải 200ppr (200xung/vòng)
- Đường kính trục 6mm. Đường kính vỏ ngoài 45mm
- Tốc độ 5.000.000 xung/phút.

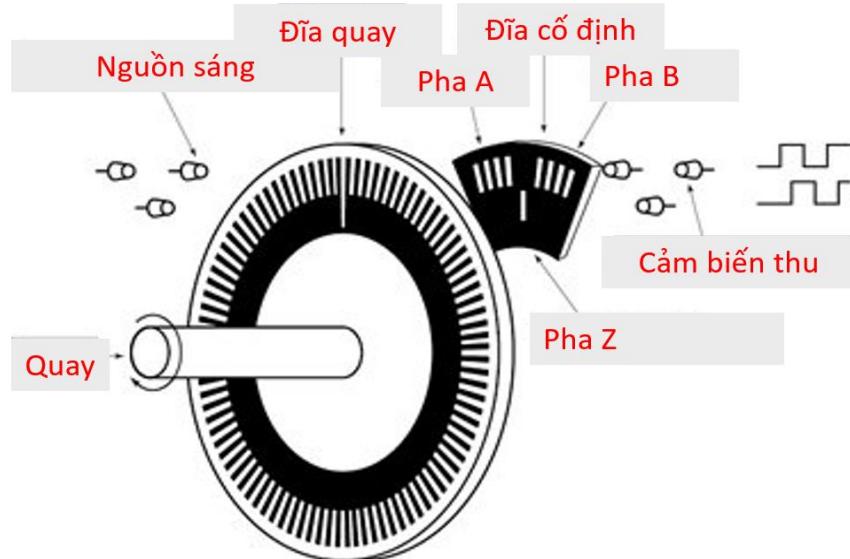
Encoder có 4 chân ra là Red, Blue, Yellow và chân Brown trong đó:

- Red cấp nguồn 5VDC
- Blue đấu nối với GND
- Yellow là chân đọc xung kênh B
- Brown là chân đọc xung kênh A



Hình 5. 11 Kích thước mặt cắt nang Encoder [7]

Giới thiệu về nguyên lý hoạt động của Encoder: Khi đĩa quay quanh trục, trên đĩa có các rãnh để tín hiệu quang chiếu qua (Led). Chỗ có rãnh thì ánh sáng xuyên qua được, chỗ không có rãnh ánh sáng không xuyên qua được. Với các tín hiệu có/không người ta ghi nhận đèn Led có chiếu qua hay không.



Hình 5. 12 Sơ đồ nguyên lý hoạt động của Encoder [7]

Số xung Encoder được quy ước là số lần ánh sáng chiếu qua khe. Ví dụ trên đĩa chỉ có 100 khe thì cứ 1 vòng quay, encoder đếm được 100 tín hiệu. Đây là nguyên lý hoạt động của loại Encoder cơ bản, còn đối với nhiều chủng loại khác thì đương nhiên đĩa quay sẽ có nhiều lỗ hơn và tín hiệu thu nhận cũng sẽ khác hơn.

Cảm biến thu ánh sáng sẽ bật tắt liên tục, từ đó:

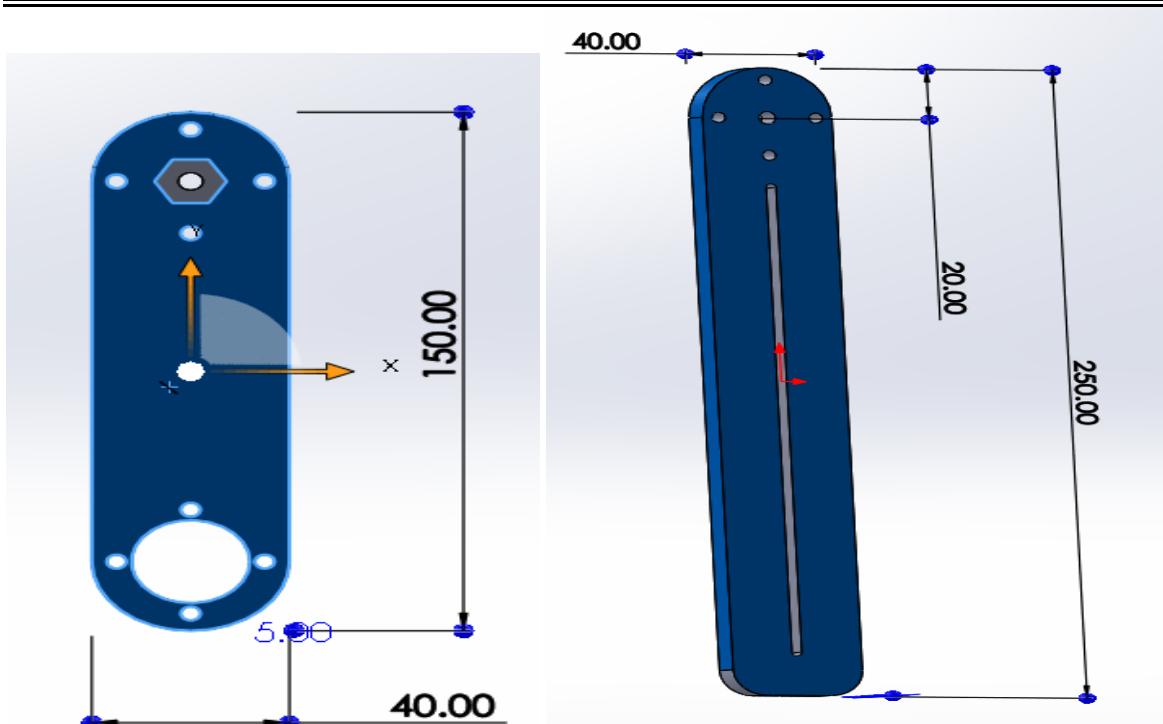
Tạo ra các tín hiệu dạng xung vuông.

Tín hiệu dạng xung sẽ được truyền về bộ xử lý trung tâm để đo đạc, xác định vị trí/ tốc độ của động cơ.

5.2.7 Thiết kế các khớp

Trong quá trình điều khiển bộ phận cứng mà thày cho sử dụng gấp một số ván đè trong quá trình điều khiển và mô phỏng cho nên nhóm quyết định thiết kế lại thông số các khớp cho phù hợp cũng như chủ động hơn trong quá trình mô phỏng và điều khiển.

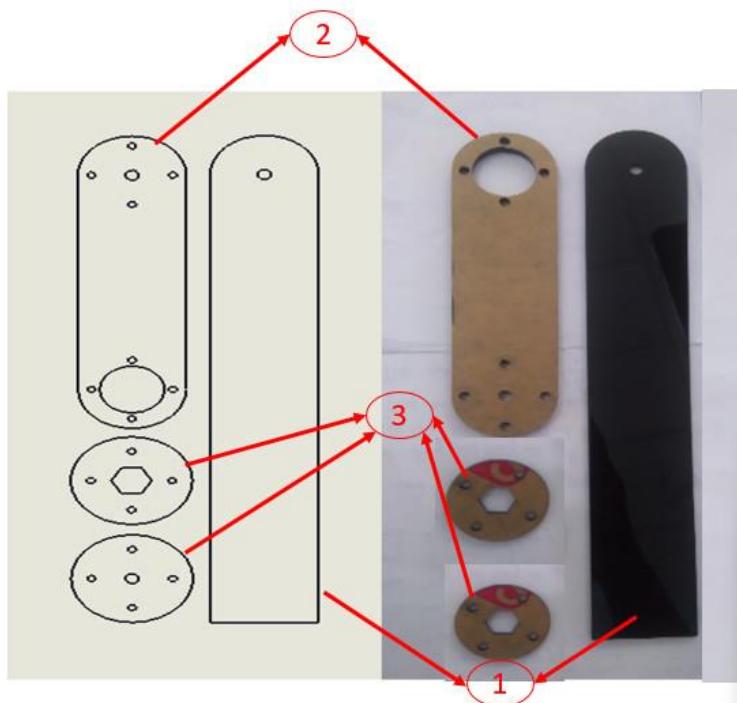
Thông số kích thước các khớp được thể hiện như trên hình vẽ với đơn vị là mm (milimet). Để tiết kiệm chi phí thì nhóm quyết định sử dụng cắt mica cho từng khớp.



Hình 5.13 Khớp 1 và khớp 2

Xuất bản vẽ cắt mica theo tỷ lệ 1:1

Khớp 1 có độ dày là 10mm, khớp 2 có độ dày là 5mm và gá gắn cố định có bề dày 3mm để giữ các khớp cố định trong quá trình điều khiển thì lực quán tính rất dễ làm cho các khớp bị lỏng đi do vậy gá này rất quan trọng đối với phần cứng



Hình 5.14 File cắt mica và sản phẩm

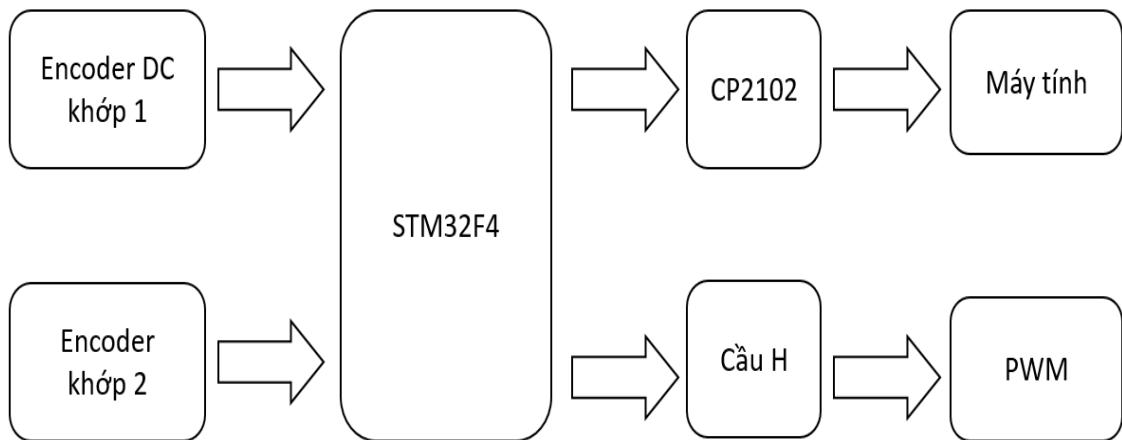
Trong Hình 5.14 ta có: “1” link1 của Pendubot được cắt từ vật liệu mica đen.

Chương 5. Thực Nghiệm Và Kết Quả

“2” là link2 được cắt từ vật liệu mica trắng. “3” là khớp nối giữa encoder và các link.

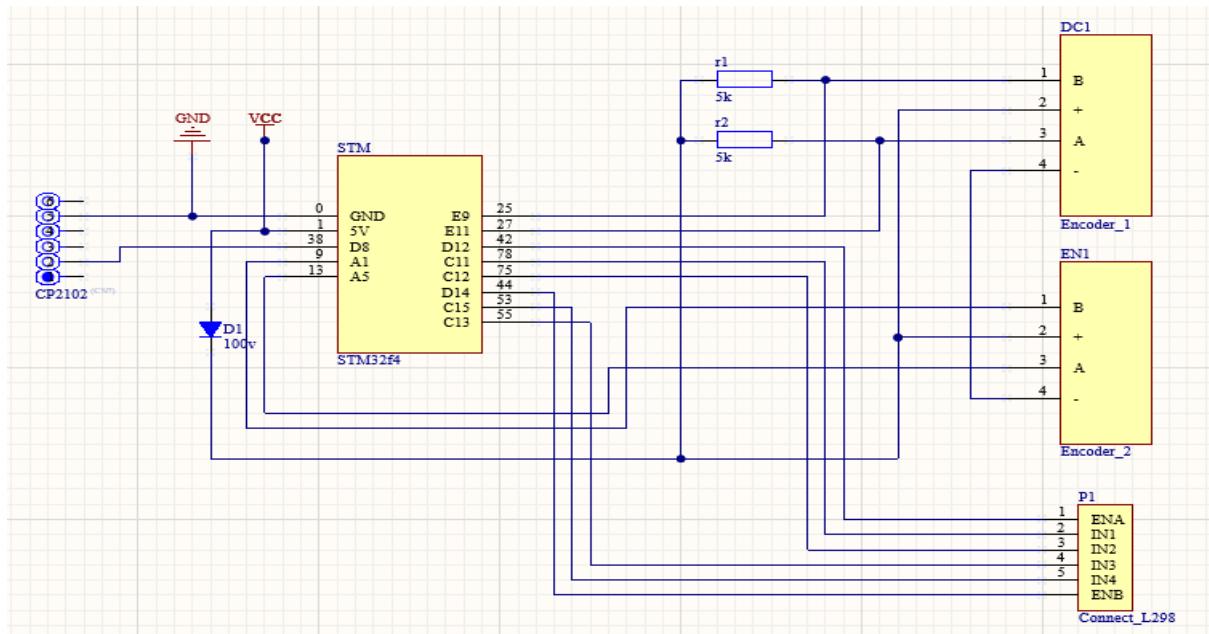
5.2.8 Thiết kế mạch ra chân

Sơ đồ khối phần điện tử của mô hình Pendubot được thể hiện trên Hình 5.16. Vì xử lý STM32F4 đọc giá trị từ cảm biến Encoder khớp 1 và khớp 2, tính toán rồi xuất tín hiệu để điều khiển động cơ thông qua cầu H L298, đồng thời giao tiếp với máy tính thông qua Module CP2102.

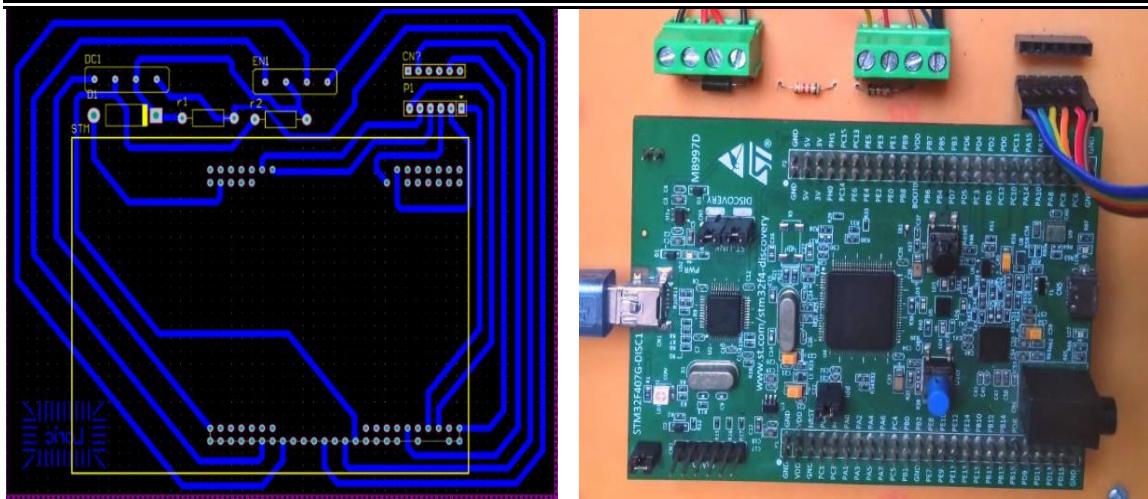


Hình 5. 15 Sơ đồ khối

Thiết kế sơ đồ nguyên lý trên Altium và tiến hành in làm mạch PCB.



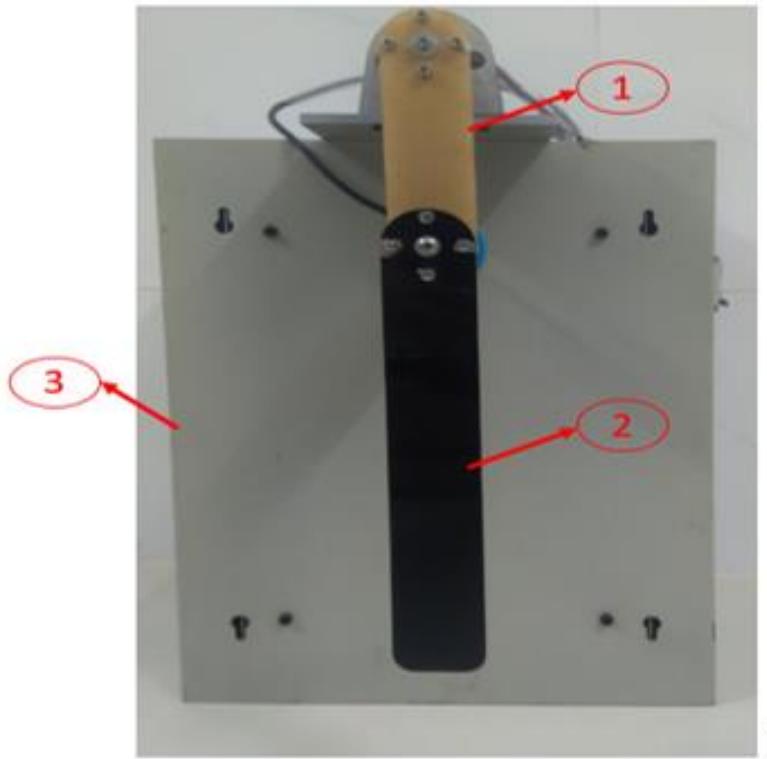
Hình 5.16 Sơ đồ nguyên lý



Hình 5. 17 Mạch ra chân

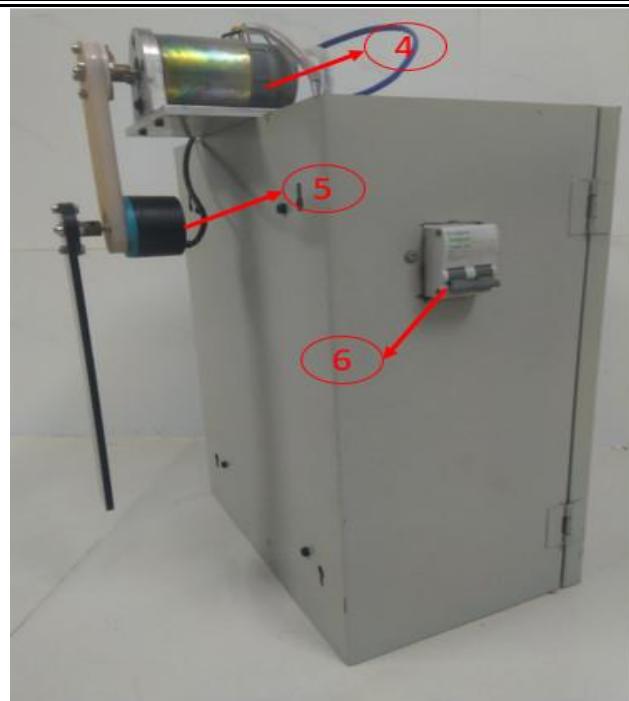
5.2.9 Sản phẩm

Sau khi thiết kế xong từng phần của mô hình, tiến hành lắp ghép mỗi phần lại với nhau theo cấu trúc đã thiết kế trong phần mềm SolidWork từ đó thu được sản phẩm hoàn chỉnh như hình dưới đây.



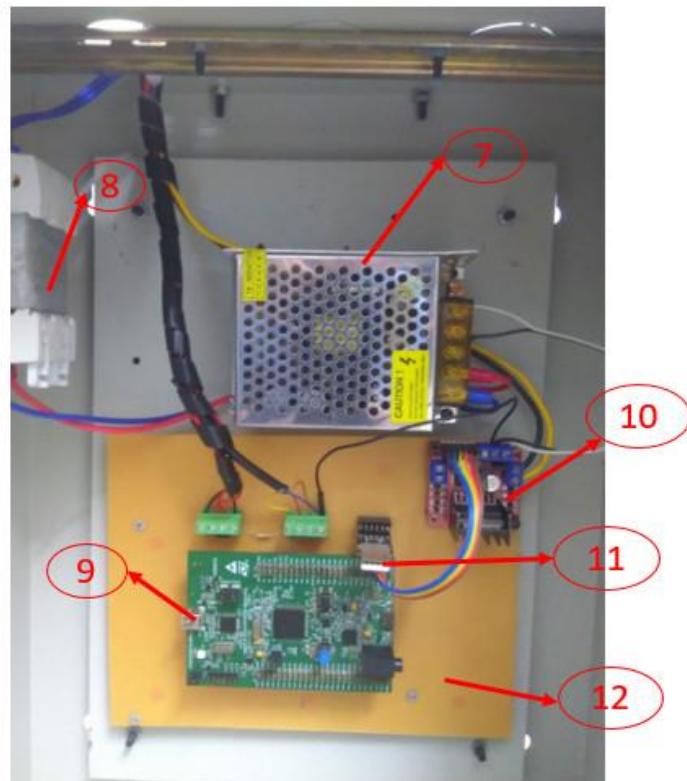
Hình 5. 18 Sản phẩm phần cứng hoàn chỉnh

Theo Hình 5. 18 : “1” là link1 của Pendubot, “2” là link thứ 2 và “3” là thủ điện chứa các linh kiện khác. Và trong đó link1 với 2 được cắt và làm từ vật liệu mica.



Hình 5.19 Góc bên của hệ thống pendubot

Trong Hình 5.19 ta có : “4” là động cơ servo NISCA NF5475 dùng để điều khiển link1 của động cơ, “5” là encoder để đọc thông tin về hệ thống. “6” là công tắc đóng mở của mô hình.



Hình 5. 20 Bên trong tủ điện.

Chương 5. Thực Nghiệm Và Kết Quả

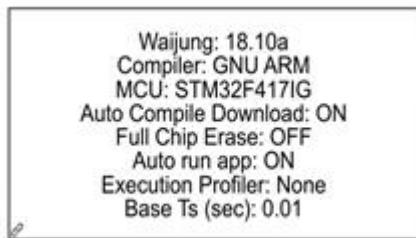
Các kí hiệu trong Hình 5. 20 gồm: “7” là nguồn tổ ong 12V , để cấp nguồn cho mô hình hoạt động, “8” là công tắc đóng ngắt nguồn cung cấp, “9” là vi xử lý STM34F4 dùng để điều khiển hệ thống. “10” là cầu H giúp điều khiển động cơ quay thuận nghịch, “11” là mạch chuyển USB UART CP2102. Và cuối cùng “12” là mạch in.

5.3 Kết quả bộ điều khiển thực nghiệm.

5.3.1 Wajjung

STM32F4 được biết đến là một KIT ARM có hỗ trợ DSP. Nên việc sử dụng KIT này để xây dựng các mô hình thí nghiệm phục vụ cho việc học tập lập trình và điều khiển là rất hữu ích. Trong những năm qua, một công ty Thái Lan đã phát triển một thư viện hỗ trợ cho việc lập trình KIT STM32F4 trên MATLAB Simulink. Điều này giúp cho việc sử dụng KIT này trong các dự án về điều khiển càng trở nên thuận lợi hơn.

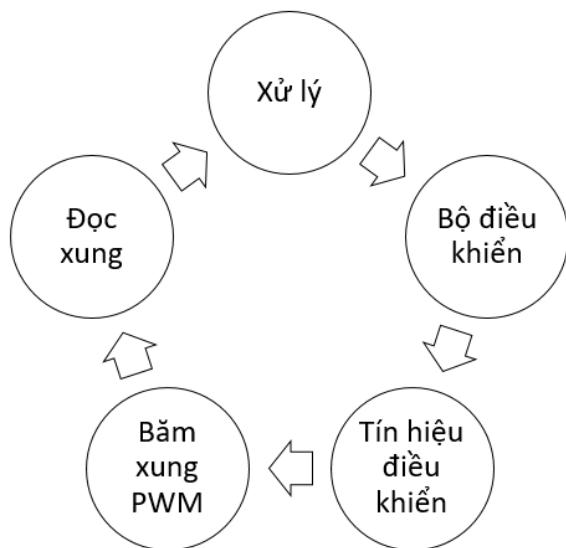
Khối Wajjung cho phép người sử dụng nạp trực tiếp code từ phần mềm MATLAB Simulink vào KIT STM32F4 mà không cần thông qua các phần mềm biên dịch khác như KeliC hoặc là ST-Link.



Hình 5.21 Khối Wajjung

5.3.2 Lưu đồ nguyên lý điều khiển

Nguyên lý điều khiển và phản hồi của hệ thống.

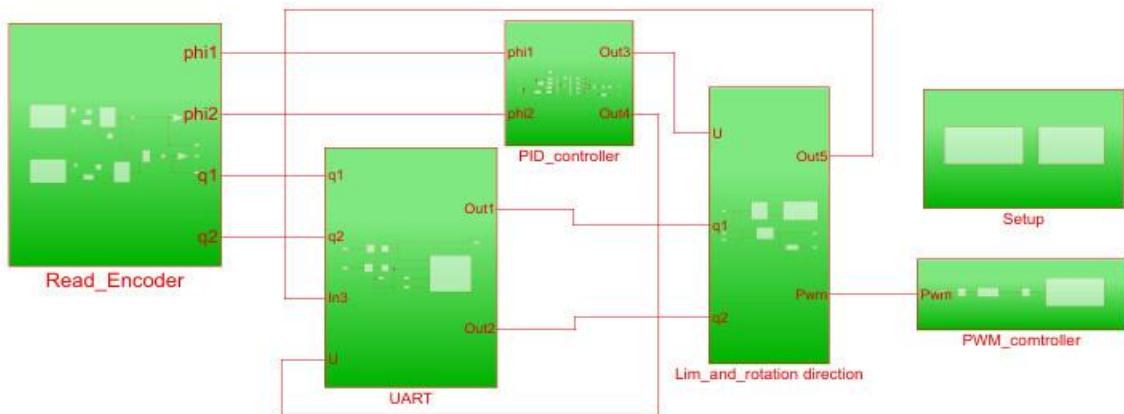


Hình 5. 22 Sơ đồ nguyên lý.

Chương 5. Thực Nghiệm Và Kết Quả

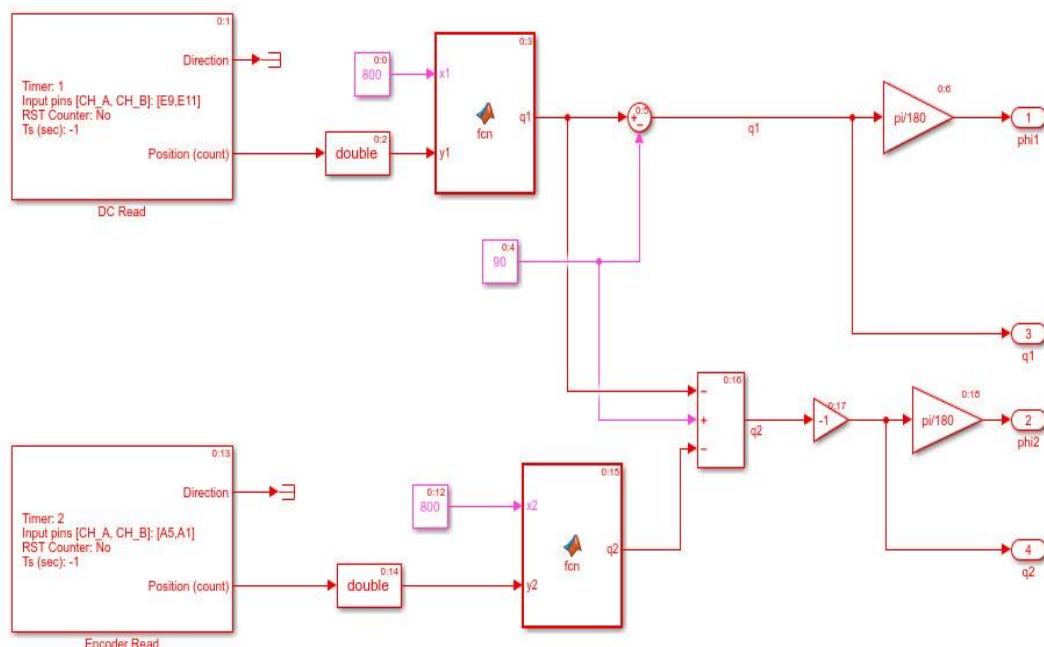
Hệ thống sẽ đọc xung từ encoder từ đó nhóm tiến hành xử lý quy đổi ra góc trong bộ biến đổi xử lý trung gian lập trình trong Matlab để đi vào bộ điều khiển, bộ điều khiển sẽ so sánh sai số từ đó xuất tín hiệu điều khiển là điện áp truyền vào LM35 cấp cho động cơ sau đó lại đọc góc về trở lại vòng lặp liên tục như vậy.

5.3.3 Kết quả điều khiển bộ PID.



Hình 5. 23 Tổng quan bộ điều khiển PID

Trong khối “Read_Encoder” ta có chương trình như sau:



Hình 5. 24 Chương trình trong khối Read_Encoder

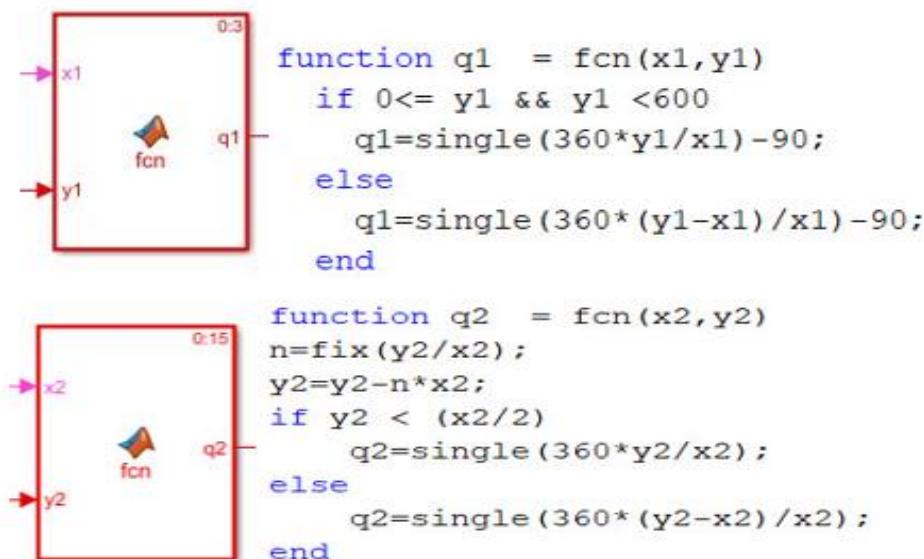
Theo hình Hình 5. 24 ta có các khối:

- Khối DC Read: Ở đây nhóm dùng timer 1 để đọc giá trị encoder của động cơ Dc. Và chân đọc giá trị về là chân E9 và E11. Vì không dùng đến Counter nên lệnh RST Counter sẽ chọn là “No”. Thời gian lấy mẫu là lệnh Ts(sec) và mang giá trị là “-1”, tức là thời gian lấy mẫu của động cơ giống với thời gian lấy mẫu của hệ thống.

Chương 5. Thực Nghiệm VÀ Kết Quả

- Khối Encoder Read: khối này nhóm dùng timer 2 để đọc giá trị về và tránh dùng trùng timer với các khối lệnh khác. Và ở đây hai chân đọc tín hiệu về là chân A5 và A1 trên vi xử lý STM32. Vì không dùng đến Counter nên lệnh RST Counter sẽ chọn “No”. Thời lấy mẫu Ts(sec) trùng với thời gian lấy mẫu của hệ thống nên mang giá trị là “-1”.
- Các khối Constant: Trong mô phỏng này các khối Constant mang giá trị là “800”. Giá trị encoder là 200 nhưng sau khi qua lệnh Wajung thì sẽ được khuếch đại gấp 4 lần là 800 để cho việc điều khiển chính xác hơn.
- Khối Double: giúp chuyển giữ liệu sang kiểu double để đồng nhất về kiểu dữ liệu.
- Các khối Gain: mang giá trị pi/180, giúp chuyển đổi giá trị từ độ sang rad.

Ngoài ra còn có 2 khối Fcn. Các khối đó chứa các code điều khiển được thể hiện sau đây:

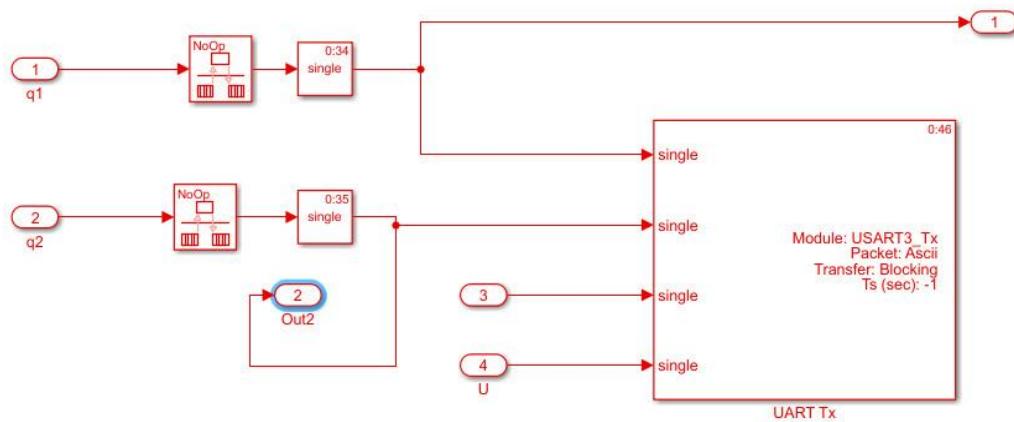


Hình 5. 25 Chương trình trong khối Fcn

Trong đó y1, y2 là xung đọc vào của encoder khớp 1,2. Và x1, x2 là độ phân giải sau khi khuếch đại của mỗi encoder. Nếu xung đọc vào của động cơ khớp 1 trong khoảng từ 600 – 900 thì góc “q1=single(360*(y1-x1)/x1)-90” và nếu từ 0-600 thì sẽ xét điều kiện còn lại.

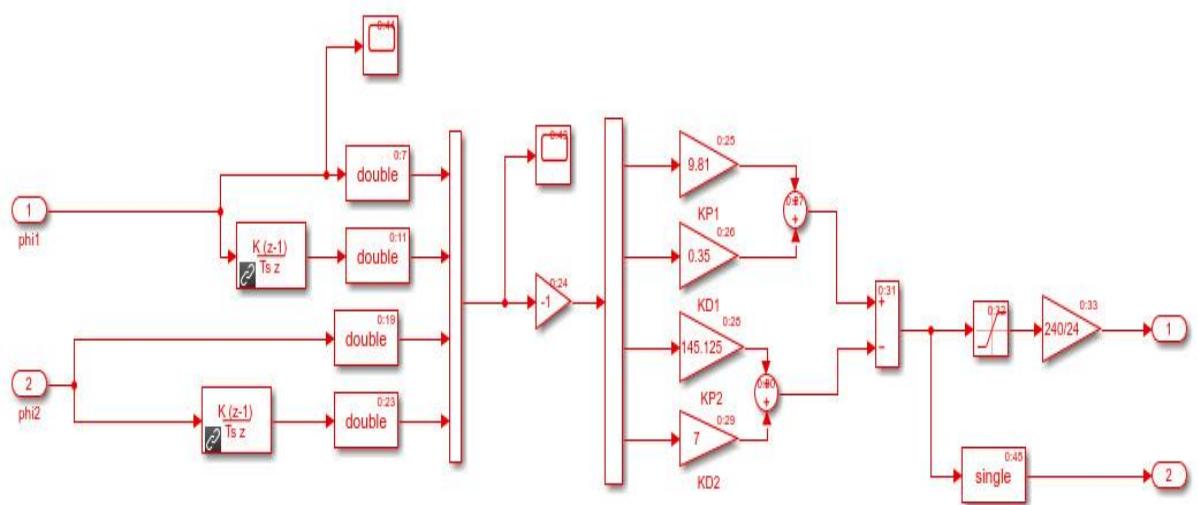
Còn nếu xung đọc vào của encoder khớp 2 bé hơn xung khớp 1 chia cho 2 thì góc “q2=single(360*y2/x2)” và nếu khác thì còn trường hợp còn lại.

Trong khối “UART” ta có các khối điều khiển sau:



Hình 5. 26 Chương trình khói UART

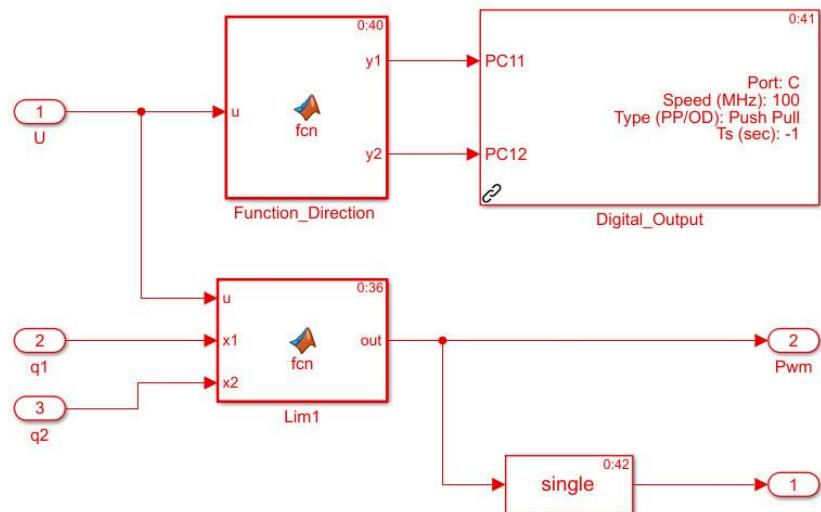
Cấu trúc bộ điều khiển PID bên trong khối “PID_controller”.



Hình 5. 27 Cấu trúc bộ điều khiển PID

Trong đó các thông số điều khiển lần lượt là: $kp1 = 9.81$, $kd1 = 0.35$, $kp2 = 145.125$ và $kd2 = 7$. Các thông số điều khiển này được tìm bằng kinh nghiệm và kỹ năng đã được học. Không dùng bất kì phương nào để hỗ trợ. Theo các thông số trên thì ta thu được các kết quả như ở dưới.

Chương trình trong khối “Lim_and_rotation direction”: dùng để giới hạn góc quay cả hệ thống.



Hình 5. 28 Chương trình trong khối Lim_and_rotation direction

Theo Hình 5. 28 ta có chức năng các khối:

- Khối “Function_Direction” có chương trình như sau:

```
function [y1,y2] = fcn(u)
if u<0
    y1=0;
    y2=1;
else
    y1=1;
    y2=0;
end
```

Hình 5. 29 Chương trình khối Function_Direction

Chương trình của Hình 5. 29 giúp đảo chiều động cơ dựa vào giá trị điện áp u trả về là dương hay âm.

- Khối “Lim1” có chương trình như sau:

```
function out = fcn(u, x1,x2)
if (-90 < x1) && (x1<90) && (x2 <30) && (x2>-30)

    out=u;

else
    out=0;
end
```

Hình 5. 30 Chương trình trong khối Lim1

Chương 5. Thực Nghiệm Và Kết Quả

Theo như khối “Lim1” thể hiện x1, x2 là góc của khớp 1,2. Nếu x1 nằm trong khoảng từ -90 – 90 độ và x2 từ -30 – 30 độ thì xuất tín hiệu điều khiển là out=u cho hệ Pendubot. Nằm ngoài khoảng đó thì out=0.

- Khối “Digital_Output”: với tần số điều khiển là 100Mhz, và thời gian lấy mẫu giống với thời gian lấy mẫu của hệ thống.

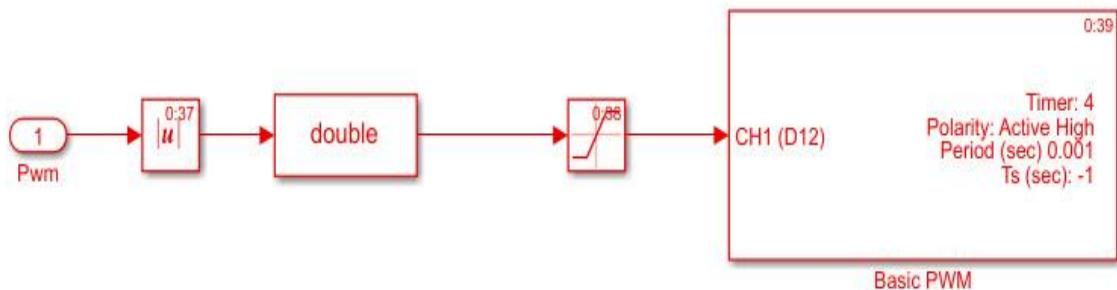
- **Khối “Setup”:**



Hình 5. 31 Chương trình trong khối Setup

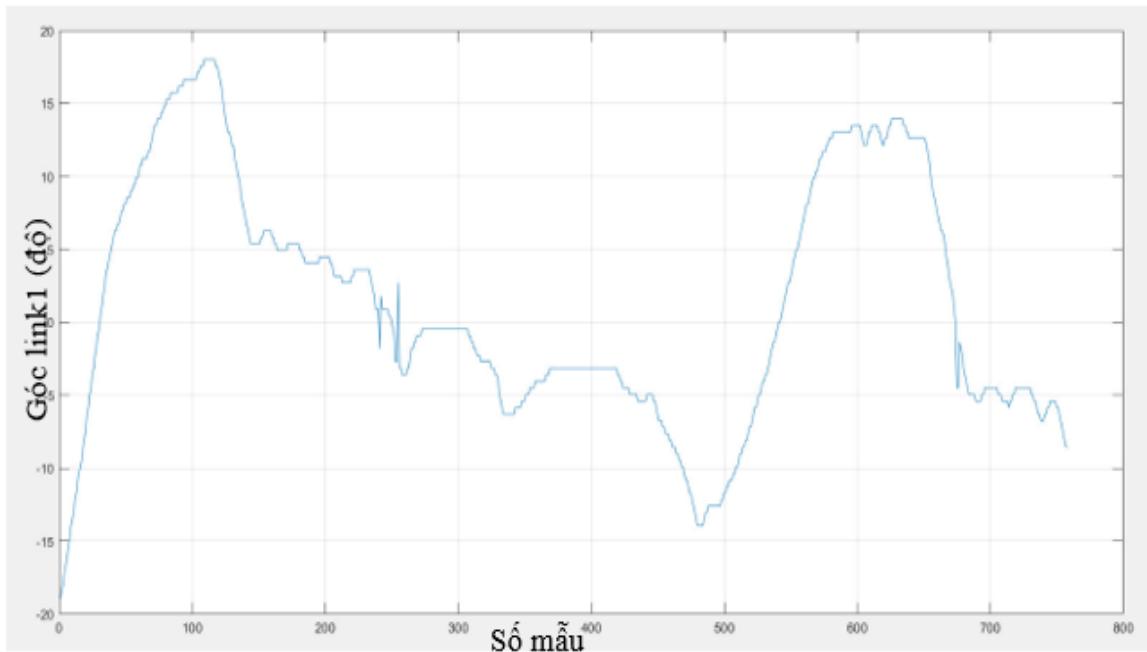
- Trong Hình 5. 31 gồm 2 khối là khối Waijung và khối USART. Khối Waijung hỗ trợ nhúng STM32F4 vào Matlab. Khối USART giúp đọc tín hiệu phản hồi.

- **Khối “PWM_Controller”:** Dùng để xuất tín hiệu điều khiển PWM.



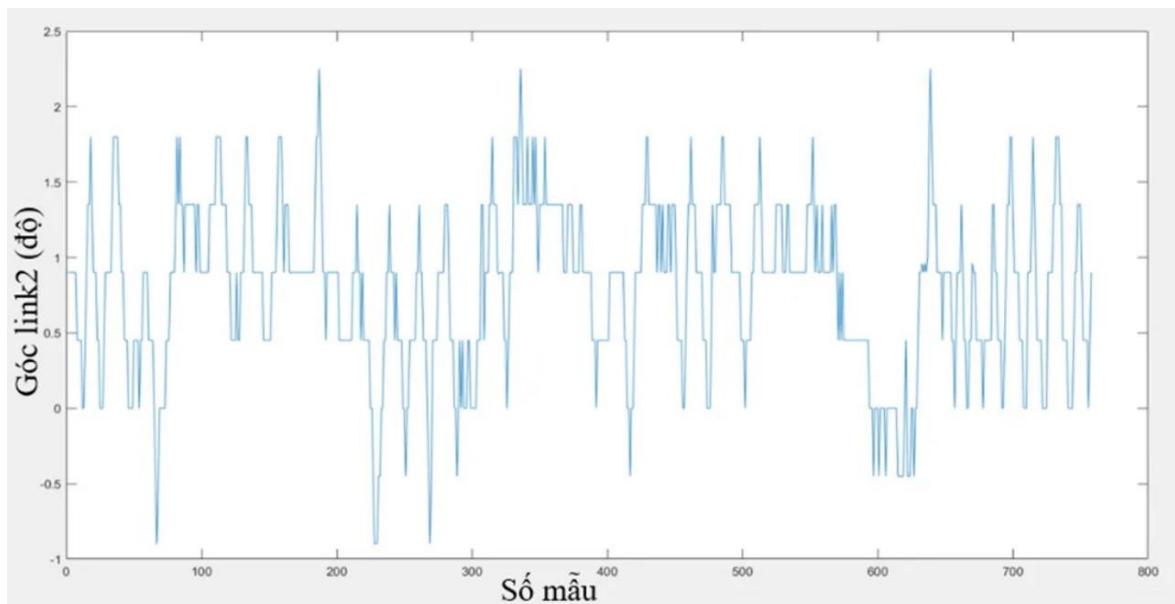
Hình 5. 32 Chương trình trong khối PWM_Controller

Trong đó, khối “Basic PWM” sử dụng timer4 với tín hiệu tác động mức cao. Với chu kỳ PWM là 0.001 giây là bội của thời gian lấy mẫu. Và thời gian lấy mẫu giống với thời gian lấy mẫu của hệ thống là 0.01



Hình 5.33 Góc của khớp thứ nhất so với phương thẳng đứng.

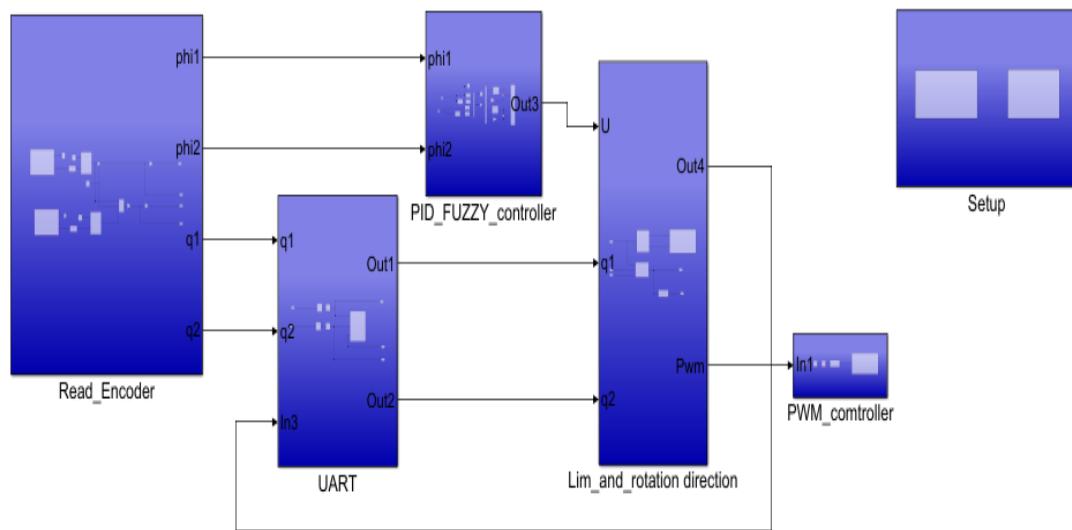
Theo Hình 5.33, Góc của khớp 1 so với phương thẳng đứng đang còn giao động nhiều và chưa ổn định ta có thể quan sát trên hình 5.21, góc của khớp này giao động chủ yếu xung quanh điểm cân bằng là 0. Số lượng mẫu là 800 tương ứng với khoảng thời gian cân bằng là 8 giây. Vì link 1 điều khiển link2 quanh vị trí cân bằng nên link1 sẽ hoạt động trong vùng rộng hơn và ở đây vùng giao động trong khoảng từ -15 tới 20 độ.



Hình 5. 34 Góc của khớp thứ 2 so với phương thẳng đứng

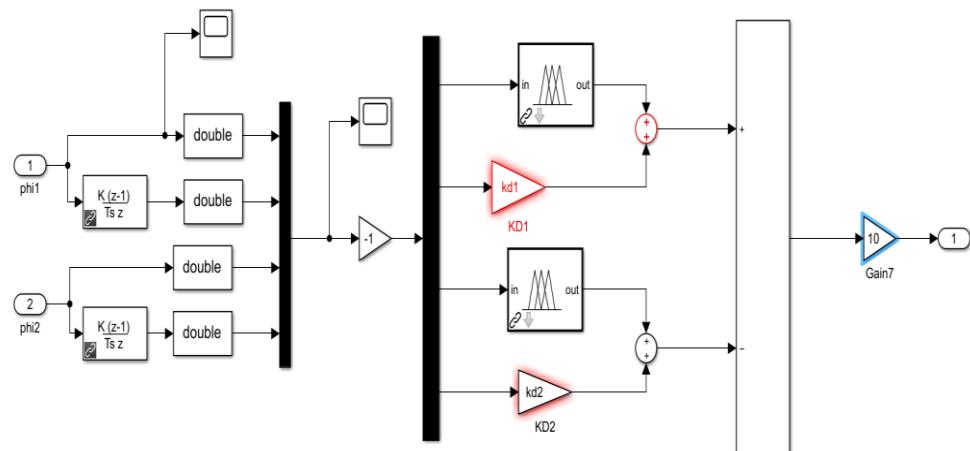
Từ Hình 5. 34 ta thấy góc của khớp thứ 2 so với phương thẳng đứng giao động quanh điểm cân bằng là 0 và góc sai số lớn nhất tầm khoảng từ -1 độ đến 2.45 độ. Số lượng mẫu là 800 mẫu, chu kỳ lấy mẫu là 0.01 giây tương đương với 8 giây.

5.3.3 Kết quả điều khiển bộ PID-Fuzzy.



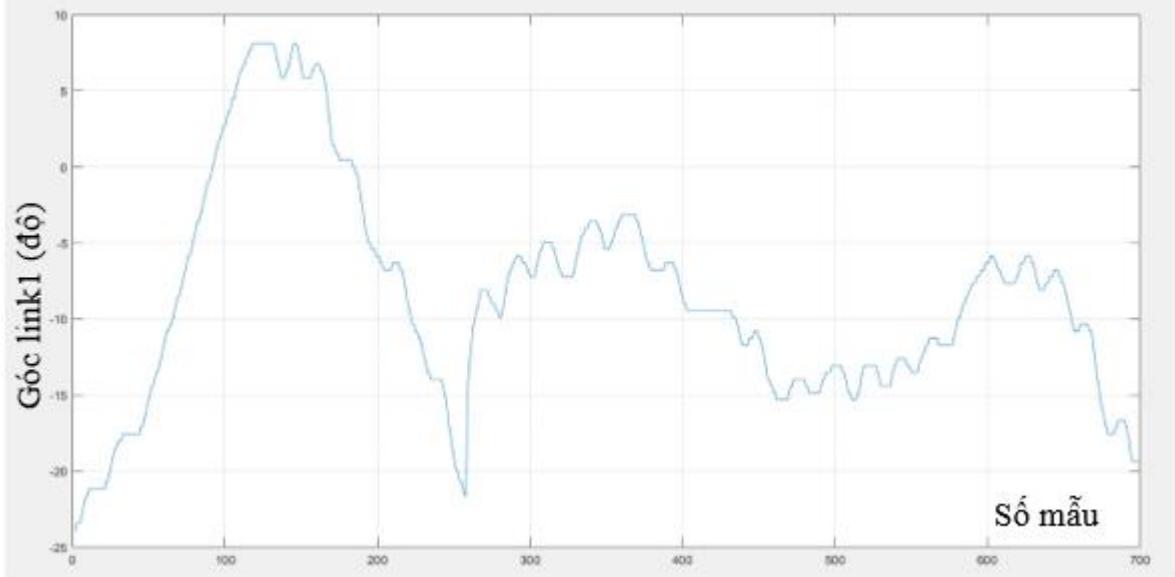
Hình 5.35 Tổng quan bộ điều khiển PID-Fuzzy.

Cấu trúc bên trong của bộ điều khiển PID-FUZZY trong khối PID_FUZZY_controller được biểu diễn trong hình dưới đây.



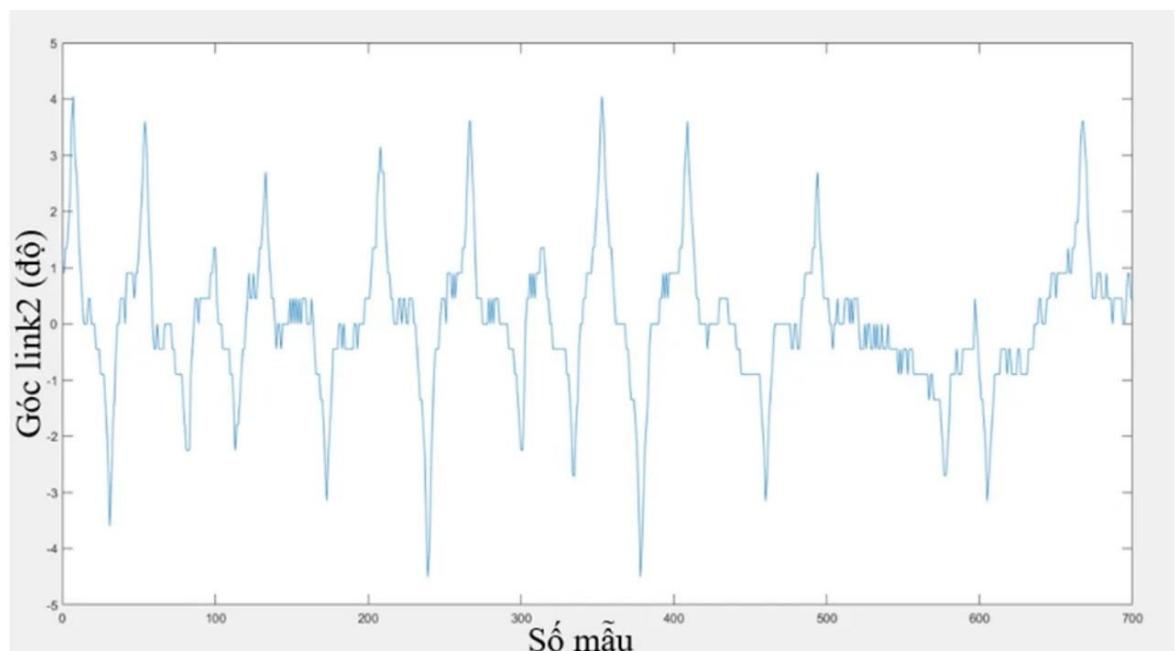
Hình 5.36 Bộ điều khiển PID-Fuzzy cho Kp1 và Kp2

Ở Hình 5.36, theo bộ điều khiển PID thì bộ điều khiển PID-Fuzzy được tạo thành bằng cách biến thông số kp thành khối Fuzzy thông qua hàm anfisedit. Và bộ Fuzzy ấy được tạo thành dựa trên thông số của kp. Các kết quả mô phỏng sẽ được thể hiện ở phí dưới.



Hình 5.37 Góc của khớp thứ nhất so với phương thẳng đứng

Từ Hình 5.37 với thời gian lấy mẫu là 7 giây ta thấy được góc q1 khi hệ dữ được ổn định cân bằng thì có xu hướng giao động lệch sang vị trí góc nhỏ hơn 0 và quan sát đồ thị thì góc này giao động gần như quanh điểm -11.5 độ.



Hình 5.38 Góc của khớp thứ hai so với phương thẳng đứng.

Quan sát đồ thị thu được của khớp thứ hai từ Hình 5.38 ta có thể thấy rằng góc q2 giao động tương đối ổn định quanh vị trí cân bằng 0 và sai số rơi vào khoảng từ -4 đến 4 độ có thể coi là ổn định

Trong Hình 5.33, Hình 5.34, Hình 5.37 và Hình 5.38 trực tung của đồ thị biểu diễn góc đơn vị là độ còn trực hoành của biểu đồ là số lượng mẫu.

Kết quả:

Chương 5. Thực Nghiệm Và Kết Quả

- Từ thực nghiệm cho thấy, do ta chú trọng việc điều khiển của góc của khớp thứ 2 nhiều hơn nên dẫn tới độ ổn định của góc q2 khớp thứ 2 tốt hơn hẳn so với góc q1 của khớp thứ nhất và đường như khớp thứ nhất có xu hướng lệch sang một bên.
- Từ việc so sánh hai bộ điều khiển PID và PID kết hợp Fuzzy nhóm nhận thấy mỗi bộ điều khiển đều có thể điều khiển cân bằng cho hệ thống pendubot. Tuy nhiên bộ PID có kết quả thu được khả quan hơn bộ PID kết hợp Fuzzy.
- Ở đây, Nhóm đã cố gắng thay đổi khâu Kp bằng một khối Fuzzy và học lại khâu Kp đó. Tuy việc điều khiển PID-Fuzzy là thành công, nhưng vẫn chưa thay thế được bộ điều khiển PID. Do đó bộ điều khiển đơn giản vẫn là phương pháp tối ưu nhất.
- Kết quả mô phỏng đã cho ta thấy hệ thống điều khiển ổn định và kết quả thực nghiệm đã chứng minh điều đó. Tuy nhiên kết quả thực nghiệm sẽ không được tốt như kết quả mô phỏng, vì mô hình còn phải chịu nhiều tác động và nhiễu bên ngoài. Tuy nhiên thực nghiệm vẫn chứng minh được bộ điều khiển PID và PID-Fuzzy có thể điều khiển ổn định cho Penduot trong thực tế.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết Luận

Qua việc mô phỏng cũng như chạy thực tế mô hình, sinh viên rút ra những kết luận:

- ❖ Bộ điều khiển PD và PD-Fuzzy điều khiển tốt hệ Pendubot tại vị trí cân bằng và hiệu quả điều khiển của hai bộ điều khiển là như nhau.
- ❖ Pendubot có phần cứng không quá phức tạp, tiết kiệm chi phí cho người nghiên cứu.
- ❖ Bộ điều khiển PD theo thời gian cần phải thay đổi nhỏ các thông số điều khiển kp và kp, do bộ điều khiển này còn phụ thuộc nhiều vào các yếu tố bên ngoài.
- ❖ Ứng dụng giải thuật di truyền giúp tối ưu thông số điều khiển và chất lượng điều khiển.
- ❖ Nắm rõ hơn và học hỏi thêm được cách nhúng vi điều khiển cụ thể là vi xử lý STM32F4 vào phần mềm MATLAB.
- ❖ Tuy nhiên do hạn chế về phần cứng nên không thể làm bộ điều khiển Swingup. Và bị ảnh hưởng bởi sợi dây đọc tín hiệu của encoder.
- ❖ Bộ điều khiển PID theo thời gian cần hiệu chỉnh thông số để hệ thống hoạt động tốt hơn.

6.2 Hướng phát triển

- ❖ Khắc phục sự sai số theo thời gian của bộ điều khiển PD.
- ❖ Thay đổi một số chi tiết phần cứng để dùng được bộ mô phỏng Swingup.
- ❖ Nâng cấp hệ Pendubot hai link thành hệ Pendubot ba link.
- ❖ Ứng dụng thêm các bộ điều khiển khác LQR, Backstepping, ... để so sánh và tìm kiếm bộ điều khiển tối ưu hơn cho hệ Pendubot.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Tran Hoang Chinh, Nguyen Minh Tam, Nguyen Van Dong Hai (10/2017). Ứng dụng điều khiển PID- Mờ cho hệ pendubot. *Journal of Technical Education Science No.44A*.
- [2] Nguyễn Thành Nguyên, Nguyễn Phong Lưu, Nguyễn Văn Đông Hải (06/2019). Điều khiển pendubot sử dụng dạng toàn phương tuyến tính dựa trên logic mờ. *Tạp chí khoa học*, số 38.
- [3] Phan Văn Kiêm (2018). *Điều khiển mờ trượt cho hệ Pendubot*, Luận văn Thạc sĩ kỹ thuật, Trường Đại Học Sư Phạm Kỹ Thuật TP.HCM.
- [4] Nguyễn Khánh Ngọc (2012). *Tìm hiểu Logic mờ và xây dựng ứng dụng điều khiển tự động tốc độ xe ô tô*, Luận văn Thạc Sĩ, Trường Đại học quốc gia TP.HCM
- [5] Võ Anh Khoa, Huỳnh Xuân Dũng, Nguyễn Minh Tâm và cộng sự (04/2020). Mô phỏng điều khiển mờ trượt thích nghi thứ bậc cho pendubot. *Tạp chí khoa học giáo dục kỹ thuật*, số 57,25.
- [6] Vũ Đình Đạt (2018). *Xây dựng điều khiển trượt tối ưu cho hệ Pendubot bám quy đạo*, Luận văn Thạc sĩ kỹ thuật, Trường Đại Học Sư Phạm Kỹ Thuật TP.HCM
- [7] Nguyễn Văn Đông Hải (2011). *Xây dựng bộ điều khiển nhúng tuyến hóa vào ra cho hệ xe con lắc ngược*, Luận văn Thạc sĩ Kỹ thuật, trường Đại học Bách Khoa.
- [8] Huỳnh Thái Hoàng (2014). *Hệ thống điều khiển thông minh*, Nhà xuất bản Đại học Quốc gia TP.HCM.

Tiếng Anh

- [9]. D.Zeharm, K. Benmohammed. Optimal sliding mode control of the pendubot. *International Research Journal of Computer Science and Information System*, Vol. 2(3),45-51, April 2013
- [10] Tran Vinh Toan, Tran Thu Ha, Tran Vi Do (2017). *Hybrid Control for Swing up and Balancing Pendubot System: An Experimental Result*

PHỤ LỤC

Code Giải thuật di truyền GA:

```
clc;%xoa bo man hinh  
clear all%xoa bo toan bo du lieu truoc do  
rand('state',sum(100*clock));  
%%%%%%%%%%%%%  
%  
Jmin=1000000;  
  
max_generation=20000; %so the he toi da trong qua trinh chay(chay toi toi da  
200 the he thi dung lai)  
max_stall_generation=50000;  
epsilon=0.0001; %J chuan(neu the he nao co J<=epsilon tuc la da tim duoc thog  
so thoa man, GA khong can chay tiep nua)  
pop_size=20; % so ca the trong qun the(so cha me)  
npar = 4; %co 6 nhanh sac the trong 1 ca the(6 nhanh sac the nay lan luot la  
Kp1,Ki1, Kd1, Kp2,Ki2,Kd2  
range=[ 0      0      0      0      ;...  
       100    100    100    100    ]; %tam cua cac NST  
dec=[2 2 2 2]; %vi tri dau cham thap phan  
sig=[3 3 3 3]; %so chu so co nghia trong moi nhanh sac the  
  
cross_prob = 0.6; %he so lai ghep  
mutate_prob = 0.4; %he so dot bien (he so dot bien + he so lai ghep =1)  
elitism = 1; %luon giu lai gia tri tot nhat  
  
rho=0.02; %trong so quyet dinh e1,e2 va u cai nao quan trong voi J hon  
  
par=Init(pop_size,npar,range); %khai tao 20 ca the cha me dau tien  
  
Terminal=0;          %KHOI DONG  
generation = 0;       %CAC GIA TRI  
stall_generation=0;%DAU TIEN TRUOC KHI CHAY ga  
  
for pop_index=1:pop_size  
%Khai tao  
    k1=par(pop_index,1);  
    %cac the he  
    %cha me
```

```

k2=par(pop_index,2);
%dau tien

k3=par(pop_index,3);
%(cho chay GA truoc
ra duoc cac cha me toi uu
                           %de

k4=par(pop_index,4);

sim('Swingup.mdl');

J=(x1'*x1)+(x3'*x3);

if (length(x1)>950) && (J<Jmin)

    k1
    k2
    k3
    k4

    Jmin=J
                           %

fitness(pop_index)=1/(J+eps);

%
else
%
J=10^100;
%
fitness(pop_index)=1/(J+eps);
%
end
%
end;
%
[bestfit,bestchrom]=max(fitness);
%
k10=par(bestchrom,1);           %cac
k20=par(bestchrom,2);           %cua cha me
k30=par(bestchrom,3);           %toi uu nhat
k40=par(bestchrom,4);           %do elitism=1
J0=1/bestfit+0.001;           %(do elitism = nen doi
hoi phai co 1 cha me tot nhat de so sanh voi cac he he sau, tim toi uu nhat)

while ~Terminal, %Terminal se bang 1 neu chay du 200 the he hoac trong qua
trinh chay co 1 the he con cai co bestfit thoa man sai so <epsilon

```

```

        generation = generation+1;
%truoc moi lan chay cho hien generation tuong ung voi lan chay thu
1,2,3,...(tuong ung generation 1,2,3... hay the he thu 1,2,3... )

        disp(['generation #' num2str(generation) ' of maximum '
num2str(max_generation)]); %

pop=Encode.Decimal.Unsigned(par,sig,dec); %ma
hoa thap phan (nhiem sac the cua cac ca the)

parent=Select.Linear.Ranking(pop,fitness,0.2,elitism,bestchrom); %sap
hang cha me tuyen tinh(cham tot thi xac suat co con se cao hon, cha me yeu
ot thich it con hoac ko co con)

child=Cross.Twopoint(parent,cross_prob,elitism,bestchrom); %con
cai se duoc sinh ra dua vao lai ghep da diem(trong truong hop nay la hai
diem)

pop=Mutate.Uniform(child,mutate_prob,elitism,bestchrom); %dot
bien theo dang phan bo deu

par=Decode.Decimal.Unsigned(pop,sig,dec); %giai ma ket qua ve lai dang nhiem sac the(la so duong)

for pop_index=1:pop_size
%Khoi tao

k1=par(pop_index,1); %cac the he %cha me

k2=par(pop_index,2); %dau tien

k3=par(pop_index,3); %(cho chay GA truoc
ra duoc cac cha me toi uu %de

k4=par(pop_index,4);

sim('Swingup.mdl');

J=(x1'*x1)+(x3'*x3);

if (length(x3)>950) && (J<Jmin)

    k1

    k2

    k3

    k4

    Jmin=J %

    fitness(pop_index)=1/(J+eps);

%
else %
    J=10^100;
%
end

```

```

        fitness(pop_index)=1/(J+eps);
%
end
%
end;

[bestfit(generation),bestchrom]=max(fitness);
%ca the nao co Kp1, Kil, Kd1, Kp2, Ki2, Kd2 lam cho J nho nhat, tuc best fit
lon nhat

%se duoc chon am ca the toi uu sau khi chay hoan thanh GA, tuc la ca the
bestchrom

if generation == max_generation                               %neu
chay du 200 the he rui ma chua co ca the thoa J <=epsilon (o dong thu 10)
    Terminal = 1;                                         %thi cho
Terminal=1, stop chuong trinh GA, chon ca the tot nhat trong cac truong hop
do lam bestchrom

elseif generation>1,
    if abs(bestfit(generation)-bestfit(generation-1))<epsilon,
        stall_generation=stall_generation+1;
        if stall_generation == max_stall_generation, Terminal = 1;end
%con trong qua trinh chay ma xuat hien ca the thoa J<=epsilon(dog thu 10) thi
cho Terminal =1, dung GA va lay gia tri do lam bestchrom
    else
        stall_generation=0;
    end;
end;

end; %While

plot(1./bestfit)

k1=par(bestchrom,1)      %hien thi
k2=par(bestchrom,2)      %Kp1,Kil, Kd1, Kp2,Ki2,Kd2
k3=par(bestchrom,3)      %cua ca the tim duoc nho qua trinh chay
k4=par(bestchrom,4)      %

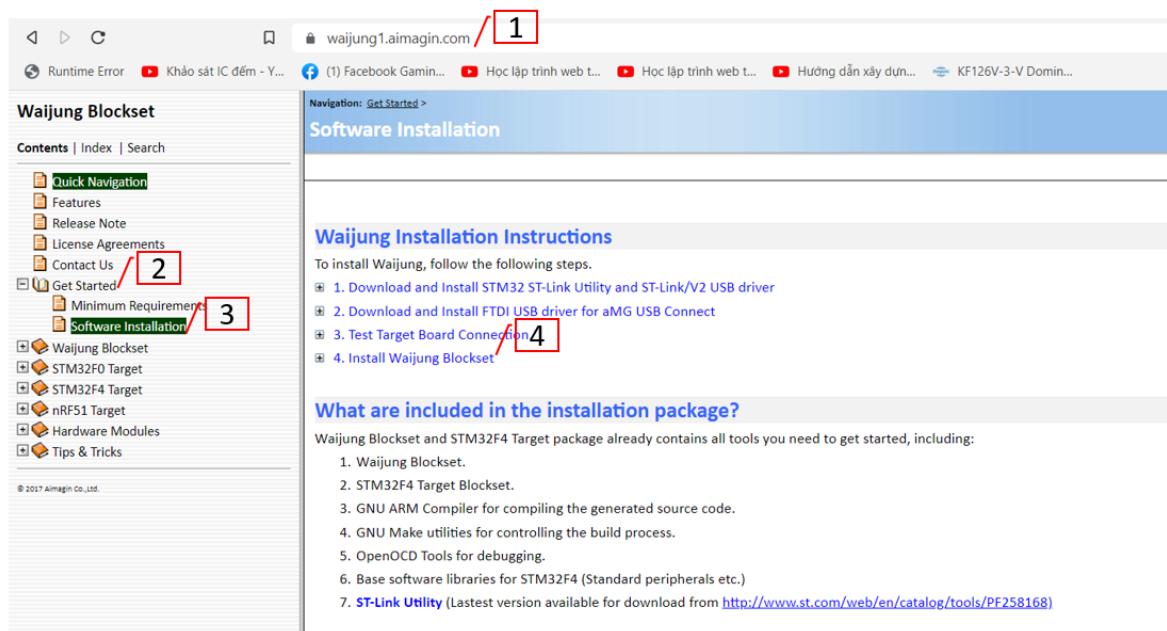
J=1/bestfit(end)  %ham tieu chuan tuong ung ca the con tot nhat do
sim('Swingup.mdl'); %tien hanh mo phong lai de kiem tra ca the con tot nhat
do cho dap ung he thong nhu the nao

```

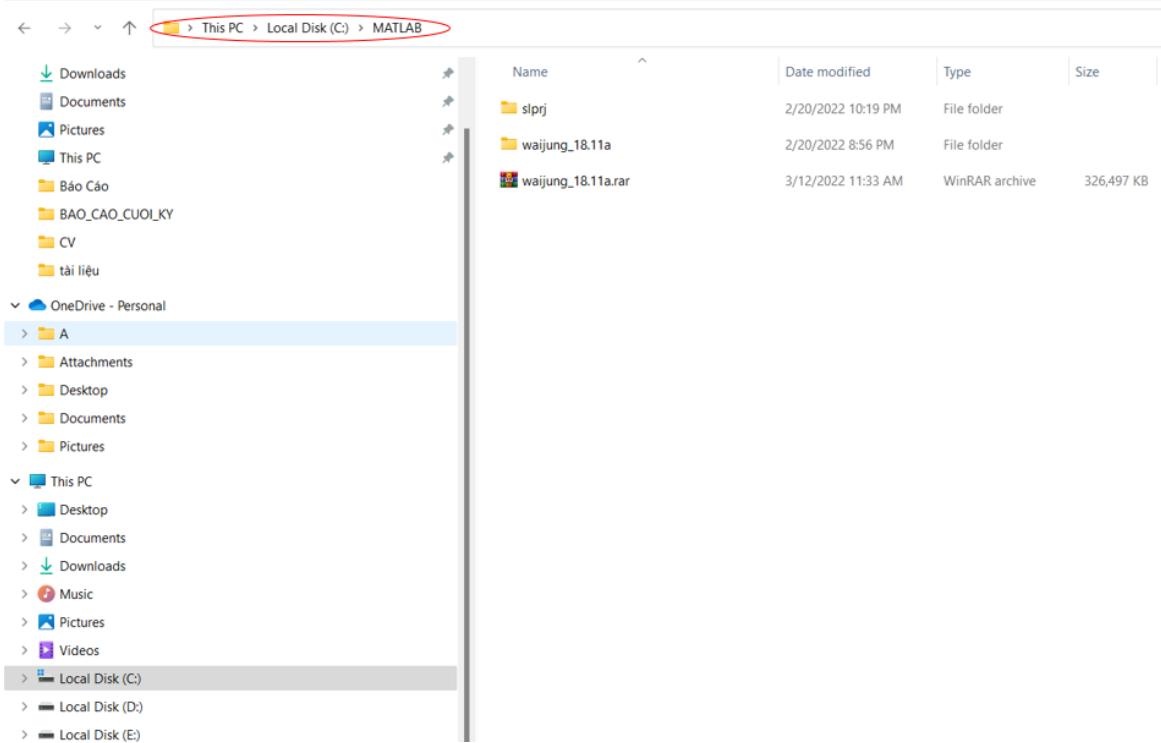
Cách cài đặt Waijung:

Cài đặt phần mềm Waijung để nhúng vi điều khiển STM32F4.

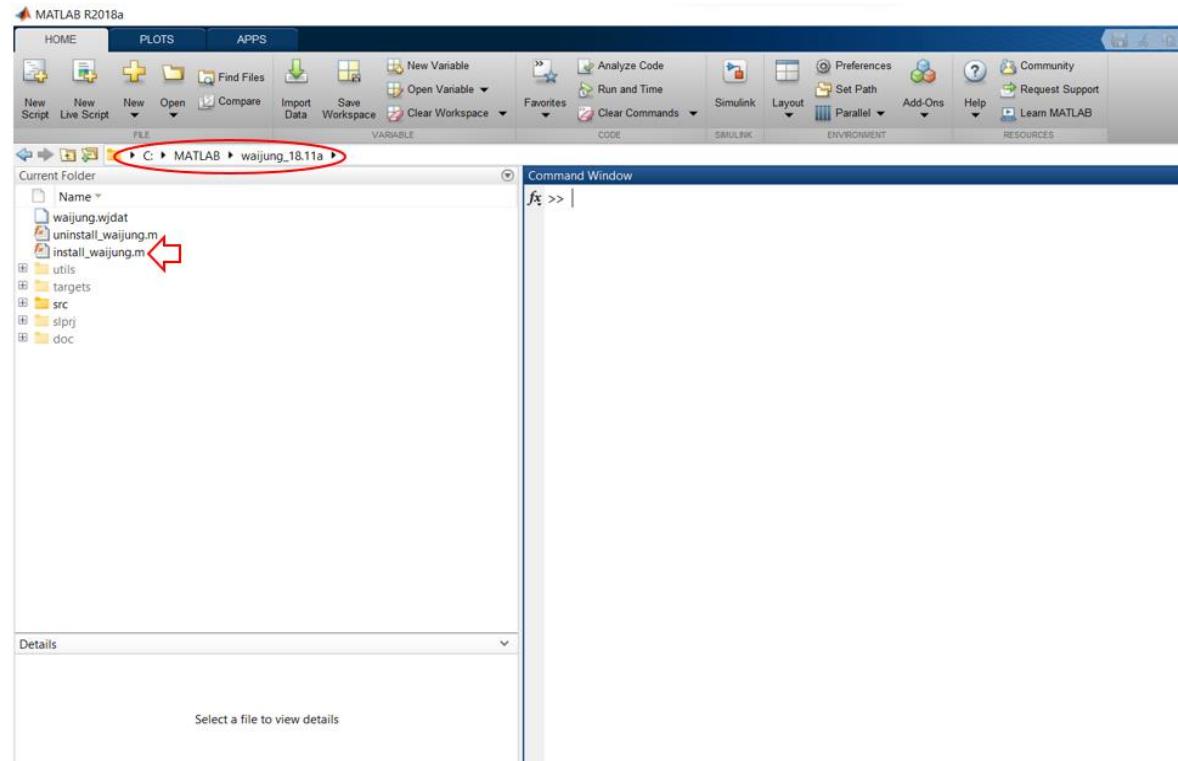
B1: Truy cập vào trang web <https://waijung1.aimagin.com/> ấn vào “Get Started” sau đó là “Software Installation” và tải “Install Waijung Blockset”.



B2: Trong ô địa C tạo thư mục MATLAB và giải nén thư mục vừa tải về vào đây.

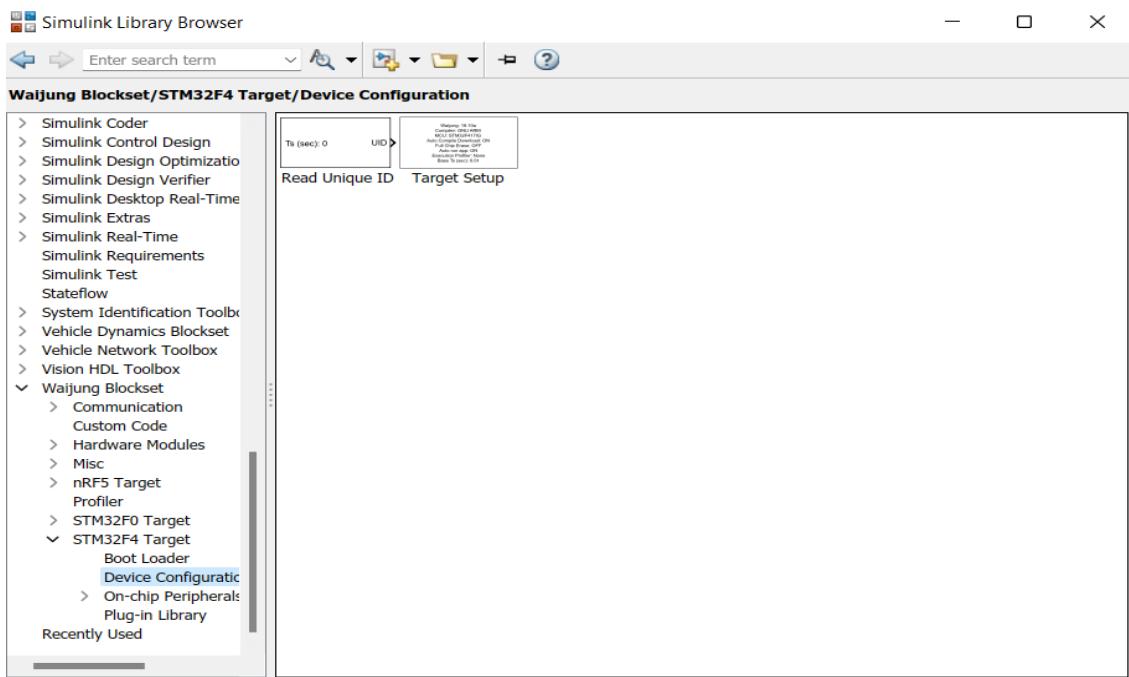


B3: Mở phần mềm Matlab trở tới đường dẫn tới Waijung sau đó click vào “install_waijung.m” để cài đặt. Sau khi cài thành công sẽ có dòng hiển thị ở Command Window: “Finish waijung Installation”



B4:

Kiểm tra xem đã có Board STM32F4 trên Simulink chưa. Nếu có rồi thì đã có thể tiến hành code và nạp vào vi điều khiển.

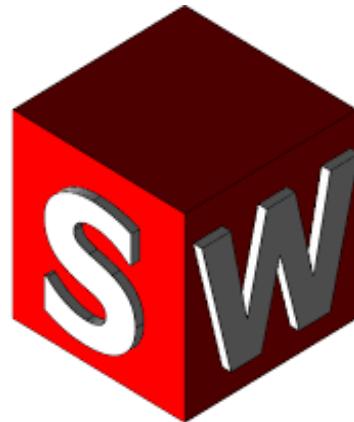


Link kết quả khi điều khiển thực tế được quay lại:

<https://www.youtube.com/shorts/GUrxFOPt4x1c>

Các phần mềm hỗ trợ:

- Phần mềm vẽ 3d hệ thống Solid work



- Phần mềm vẽ mạch cho hệ thống: Altium



