





TÓM TẮT LUẬN VĂN

Luận văn này trình bày về đề tài điều khiển con lắc ngược một bậc tự do. Luận văn sẽ trình bày về cơ sở các lý thuyết được áp dụng, chi tiết các phần như thiết kế cơ khí của mô hình, tìm hiểu và xây dựng mô hình toán, xây dựng và tiến hành mô phỏng mô hình trên Matlab, xây dựng các khối điều khiển trên thư viện nhúng và thiết kế giao diện điều khiển.

Việc thiết kế cơ khí của mô hình chủ yếu dựa vào các mẫu mô hình hiện có từ các nghiên cứu về con lắc ngược trong và ngoài nước để thiết kế lên mô hình đơn giản và phù hợp nhất với nhu cầu của luận văn, sau đó xây dựng lên mô hình toán để tiến hành mô phỏng trên chương trình Simulink của Matlab để khảo sát các đặc tuyến và thông số của mô hình, và dựa vào các thông số đó mà tiến hành thiết kế khâu điều khiển mô hình mô phỏng theo nhu cầu của đề tài.

Tiếp đến, để chạy mô hình thực tế dựa trên bộ điều khiển mô phỏng, ta kết nối qua thư viện nhúng hỗ trợ trên Matlab Simulink. Người thực hiện sử dụng thư viện Waijung, một thư viện hỗ trợ nhúng từ Matlab cho kit ứng dụng STM32F4 DISCOVERY của hãng STMicroelectronics, để xây dựng các khối nhúng xử lý các tín hiệu đầu vào và xuất tín hiệu đầu ra, đồng thời sử dụng các khối chức năng hỗ trợ sẵn có trên thư viện nhúng để xây dựng giải thuật điều khiển theo mô hình thực tế. Sau khi đã xây dựng xong mô hình các khối chức năng, người thực hiện sẽ tiến hành biên dịch các khối đó thành các đoạn mã Code C và nạp xuống vi điều khiển để chạy ứng dụng. Đồng thời, những người thực hiện sẽ xây dựng giao diện người dùng trên máy tính để giám sát và đánh giá chất lượng điều khiển thực tế. Các tín hiệu hồi tiếp thực tế sẽ được vi điều khiển thu thập và gửi lên máy tính qua khối truyền thông nối tiếp sử dụng IC FT232 USB – Serial. Việc so sánh các tín hiệu đó với các đặc tuyến mô phỏng sẽ giúp cho những người thực hiện chỉnh định các thông số của bộ điều khiển để hệ thống đạt được chất lượng tốt nhất.

Và cuối cùng, khi việc điều khiển mô hình thực tế đã thực hiện tốt, những người thực hiện sẽ đánh giá chất lượng điều khiển và rút ra kết luận của luận văn, đồng thời đề ra hướng phát triển của luận văn.

MỤC LỤC

ΤĆ	M T	ÅT LUẬN VĂN	III
Μl	Ų C L I	ŲC	IV
DA	NH S	SÁCH HÌNH MINH HỌA	VII
1.	СН	ƯƠNG 1: GIỚI THIỆU	1
	1.1	Tổng quan	1
	1.2	Tình hình nghiên cứu trong và ngoài nước	3
	1.3	Ý nghĩa thực tiễn của đề tài	4
	1.4	Mục tiêu của luận văn	5
	1.5	Phương pháp nghiên cứu	6
	1.6	Nội dung luận văn	7
2.	СН	ƯƠNG 2: CƠ SỞ LÝ THUYẾT	10
2	2.1	Tổng quan	10
2	2.2	Xây dựng mô hình toán đối tượng.	10
	2.2.	1 Mô hình vật lý	10
	2.2.	2 Mô hình toán	12
2	2.3	Phương pháp điều khiển LQR	14
	2.3.	1 Khái niệm	15
	2.3. (điề	2 Ôn định Lyapunov đối với hệ thống tuyến tính – Tiêu chuẩn ổn định thứ hai của Lyap cu kiện đủ)	
	2.3. Rice	3 Điều khiển tối ưu hệ tuyến tính với chỉ tiêu chất lượng dạng toàn phương – Phương trì cati đối với hệ liên tục	
	2.3.	4 Phương trình Riccati đối với hệ rời rạc	19
	2.3.	5 Các bước thiết kế phương pháp điều khiển toàn phương tuyến tính hệ thống liên tục	20
2	2.4	Phương pháp điều khiển mờ	21
	2.4	1 Tổng quan và Logia mờ	21

	2.4.2	Khái niệm về tập mờ	22
	2.4.3	Các phép toán trên tập mờ và giải mờ	24
	2.4.4	Biến ngôn ngữ và giá trị của biến ngôn ngữ	26
	2.4.5	Luật hợp thành mờ	26
	2.4.6	Phương pháp giải mờ	28
	2.4.7	Cấu trúc bộ điều khiển mờ	31
3.	CHƯƠN	NG 3: MÔ PHỎNG ĐIỀU KHIỂN HỆ THỐNG CON LẮC NGƯỢC QUAY	33
	3.1 Tổn	ng quan	33
	3.2 Xây	dựng mô hình mô phỏng hệ thống con lắc ngược quay	33
	3.3 Giả	i thuật điều khiển con lắc ngược quay	36
	3.3.1	Lưu đồ giải thuật	36
	3.3.2	Giải thuật Swing up	37
	3.3.3	Giải thuật điều khiển cân bằng LQR	39
	3.3.4	Giải thuật cân bằng dùng bộ điều khiển mờ trực tiếp	45
4.	CHƯƠN	NG 4: THIẾT KẾ VÀ THỰC HIỆN MÔ HÌNH	53
	4.1 Lựa	ı chọn mô hình	53
	4.1.1	Mô hình xe con lắc ngược	53
	4.1.2	Mô hình con lắc ngược quay	54
	4.1.3	Các cấp mô hình con lắc ngược	55
	4.2 Thi	ết kế mô hình con lắc ngược quay một bậc tự do	56
	4.3 Các	thiết bị trong mô hình	58
	4.3.1	Mô hình cơ khí	58
	4.3.2	Động cơ	58
	4.3.3	Cảm biến góc quay – encoder	59
	4.3.4	Mạch điều khiển	60
	4.3.5	Mạch cầu H	65
	4.3.6	Khối truyền thông nối tiếp	69
_	CHIICM	NC 5. THIỆT VỀ CHƯƠNG TĐÌNH ĐIỀU VIUỆN	71

ĐIỀU KHIỂN CON LẮC NGƯỢC QUAY

5.1 Th	iết kế chương trình điều khiển thời gian thực	71	
5.1.1	Giới thiệu Matlab simulink và thư viện Waijung	71	
5.1.2	Giới thiệu các khối sử dụng trong đề tài	72	
5.1.3	Xây dựng chương trình điều khiển	74	
5.2 Th	iết kệ giao diện người dùng	79	
6. CHƯƠ	NG 6: KẾT QUẢ - KẾT LUẬN – HƯỚNG PHÁT TRIỂN CỦA LUẬN VĂN	80	
6.1 Kế	t quả thực hiện	80	
6.2 Kế	t luận	86	
6.3 Đớ	ng góp của luận văn	86	
6.4 Hu	rớng phát triển của luận văn	87	
TÀI LIỆU T	THAM KHẢO	88	
PHŲ LŲC A	A: Bảng ký hiệu và thuật ngữ	89	
PHỤ LỤC B: Chương trình Matlab mô hình con lắc ngược			
PHŲ LŲC (C: Giao diện điều khiển	91	
PHU LUC I): Mô hình điều khiển thực hệ thống con lắc ngược quay	93	

DANH SÁCH HÌNH MINH HỌA

Hình 1.1: Các hệ thống trong điều khiển	1
Hình 1.2 Mô hình Robot Raven II- mô hình hiện đại được ứng dụng trong ngành Y khoa	2
Hình 1.3 Ứng dụng của mô hình con lắc ngược	4
Hình 1.4 Các mục tiêu của luận văn	5
Hình 1.5 Sơ đồ các chương trong luận văn	9
Hình 2.1 Mô hình vật lý hệ thống con lắc ngược quay	11
Hình 2.2 Mô hình đơn giản hệ con lắc ngược quay	11
Hình 2.3 Sơ đồ hệ thống điều khiển	15
Hình 2.4 Giáo sư Lotfi A.Zadeh	
Hình 2.5 Hàm liên thuộc μBx của tập mờ B	22
Hình 2.6 Độ cao, miền xác định, miền tin cậy của tập mờ	23
Hình 2.7 Các dạng hàm liên thuộc của tập mờ	24
Hình 2.8 Hợp của hai tập mờ toán tử dùng toán tử MAX	25
Hình 2.9: Giao của hai tập mờ có cùng cơ sở theo quy tắc Min (a) và theo tích (b)	25
Hình 2.10 Bù của tập mờ	
Hình 2.11 Mờ hóa biến "tốc độ"	26
Hình 2.12 Mô tả hàm liên thuộc của luật hợp thành	
Hình 2.13 Các nguyên lý giải mò theo phương pháp cực đại	
Hình 2.14 So sánh các phương pháp giải mờ	
Hình 2.15 Mô hình điều khiển mờ	
Hình 2.16 Bộ điều khiển mờ cơ bản	
Hình 3.1: Mô hình mô phỏng hệ con lắc ngược quay	
Hình 3.2: Mô hình phi tuyến mô phỏng hệ con lắc ngược quay	
Hình 3.3: Mô hình toán tuyến tính hệ con lắc ngược quay	
Hình 3.4 Mô hình mô phỏng khi chưa có bộ điều khiển	
Hình 3.5 Kết quả mô phỏng mô hình tuyến tính và phi tuyến	
Hình 3.6 Đáp ứng vị trí con lắc và cánh tay khi không có bộ điều khiển	
Hình 3.7 Lưu đồ giải thuật điều khiển hệ con lắc ngược quay	
Hình 3.8 Lưu đồ giải thuật chương trình bộ điều khiển Swing up	
Hình 3.9 Mô phỏng bộ Swing-up trong Simulink	
Hình 3.10 Đáp ứng của hệ qua bộ Swing up	
Hình 3.11 Mô phỏng hệ con lắc ngược quay trên Matlab Simulink	
Hình 3.12 Đáp ứng của góc lệch con lắc và cánh tay	
Hình 3.13 Đáp ứng của góc lệch con lắc và cánh tay khi tăng q_{33}	
Hình 3.14 Mô phỏng kết hợp hai bộ điều khiển	43
Hình 3.15 Kết quả mô phỏng vị trí góc con lắc và góc cánh tay	44
Hình 3.16 Kết quả mô phỏng vận tốc con lắc và vận tốc cánh tay	
Hình 3.17 Sơ đồ khối bộ điều khiển mờ trực tiếp điều khiển hệ con lắc ngược quay	45
Hình 3.18 Các tập mờ của e_1 , e_2 , e_3 , e_4	46
Hình 3.19 Các tập mờ của Vm	46
Hình 3.20 Mặt đặc tính của bộ mờ với tín hiệu góc lệch và vận tốc góc con lắc	
Hình 3.21 Mặt đặc tính bộ mờ với tín hiệu vị trí cánh tay và vận tốc cánh tay	

ĐIỀU KHIỂN CON LẮC NGƯỢC QUAY

Hình 3.22 Kết quả mô phỏng bộ mờ với $K_0 = 98.484, \alpha = \frac{\pi}{20}$	50
Hình 3.23 Mô phỏng bộ điều khiển mờ với $K_0 = -200, \alpha = \frac{\pi}{20}$	
20	
Hình 3.24 Sơ đồ mô phỏng bộ điều khiển mờ 1 ngõ vào	52
Hình 3.25 Kết quả mô phỏng bộ điều khiển mờ 1 ngõ vào với $K_0 = -200, \alpha = \frac{\pi}{20}$	52
This 3.25 Ret qualito phong by the kineti mo 1 ngo vuo voi $K_0 = 200, \alpha = 20$	32
Hình 4.1 Mô hình xe con lắc ngược quay trên ray cố định	53
Hình 4.2 Mô hình xe con lắc ngược di chuyển tự do	54
Hình 4.3 Mô hình con lắc ngược quay	55
Hình 4.4 Mô hình con lắc ngược quay 2 bậc và 3 bậc tự do	56
Hình 4.5 Động cơ DC Servo	
Hình 4.6 Encoder tương đối	60
Hình 4.7 Sơ đồ chân của vi điều khiển ARM STM32F4	
Hình 4.8 Mạch STM32F4 DISCOVERY	
Hình 4.9 Các linh kiện trên bản mạch STM32F4 DISCOVERY của STMicroelectronics	
Hình 4.10 Sơ đồ mạch cầu H đơn giản	
Hình 4.11 MOSFET IRF540n	
Hình 4.12 IC MC33883 và sơ đồ chân	
Hình 4.13 Khối điều khiển cầu H và cầu H	
Hình 4.14 Khối mạch chống trùng dẫn	
Hình 4.15 Khối mạch cách ly	
Hình 4.16 Cầu H sử dụng IC MC33883 tích hợp	
Hình 5.1 Quan hệ giữa Matlab – thư viện Waijung và mạch điều khiển	
Hình 5.2 Khối Target setup trong Waijung	
Hình 5.3 Khối tạo xung PWM sử dụng Timer1	
Hình 5.4 Khối đếm xung encoder sử dụng Timer1	
Hình 5.5 Các khối của UART trong Waijung	
Hình 5.6 Sơ đồ thiết kế bộ điều khiển thời gian thực	
Hình 5.7 Giao diện điều khiển thiết kế trên máy tính	
Hình 6.1 Các đáp ứng của mô hình điều khiển với tín hiệu đặt là zero	
Hình 6.2 Các đáp ứng của mô hình điều khiển với tín hiệu đặt là xung vuông	
Hình 6.3 Đáp ứng của góc cánh tay với tín hiệu điều khiển	
Hình 6.4 Điện áp điều khiển	
Hình 6.5 Đáp ứng của góc con lắc, góc cánh tay và điện áp điều khiển	
Hình 6.6 Đáp ứng của góc cánh tay, góc con lắc và điện áp điều khiển	

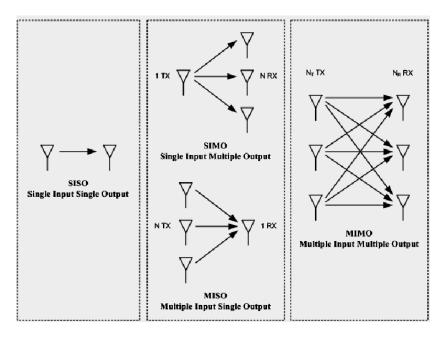
DANH SÁCH BẢNG SỐ LIỆU

Bảng 3.1 Các quy tắc giữ cân bằng hệ con lắc ngược	48
Bảng 4.1 Thông số cơ bản của mô hình con lắc ngược quay	58
Bảng 4.2 Thông số cơ bản của encoder sử dụng trong đề tài	60
Bảng 4.3 Thông số các bộ Timer của vi điều khiển ARM STM32F4	62
Bảng 4.4 Các khối Timer được sử dụng trong đề tài	62
Bảng 4.5 Thông số của các bộ USART/UART	63
Bảng 4.6 Các Pin của mạch điều khiển sử dụng trong đề luận văn	65
Bảng 4.7 Các thông số cơ bản của MOSFET IRF540n	66

CHƯƠNG 1: GIỚI THIỆU

1.1 Tổng quan

Mô hình không gian trạng thái có ưu điểm là mô tả được đặc tính động học bên trong hệ thống (các biến trạng thái) và có thể dễ dàng áp dụng cho hệ MIMO và hệ thống biến đổi theo thời gian. Lý thuyết điều khiển hiện đại ban đầu được phát triển chủ yếu cho hệ tuyến tính, sau đó được mở rộng cho hệ phi tuyến bằng cách sử dụng lý thuyết của Lyapunov.

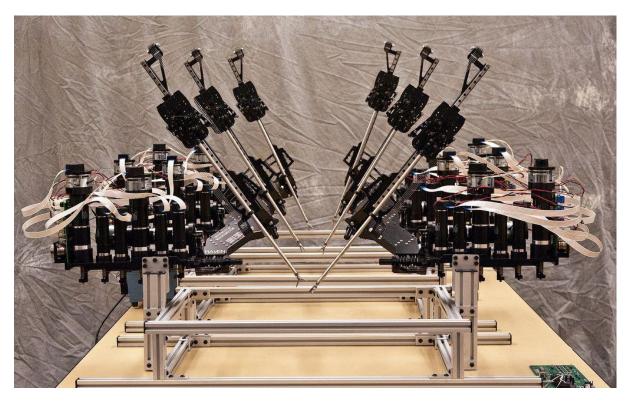


Hình 1.1: Các hệ thống trong điều khiển

Bộ điều khiển được sử dụng chủ yếu trong thiết kế hệ thống điều khiển hiện đại là bộ điều khiển hồi tiếp trạng thái. Tùy theo cách tính vector hồi tiếp trạng thái mà ta có phương pháp phân bố cực, điều khiển tối ưu, điều khiển bền vững...Với sự phát triển của lý thuyết điều khiển số và hệ thống rời rạc, lý thuyết điều khiển hiện đại rất thích hợp để thiết kế các bộ điều khiển là các chương trình phần mềm chạy trên vi điều khiển và máy tính số. Điều này cho phép thực thi được các bộ điều khiển có đặc tính động phức tạp hơn cũng như hiệu quả hơn so với các bộ điều khiển đơn giản như PID hay sớm trễ pha trong lý thuyết cổ điển.

Trong ba thập niên gần đây, lĩnh vực nghiên cứu về robot ứng dụng điều khiển bằng lý thuyết điều khiển hiện đại đã có những bước tiến vượt bậc cả về lý thuyết và ứng dụng. Nhiều nghiên cứu robot điều khiển bằng phương pháp điều khiển mờ, điều khiển LQR... đã được thiết kế cho các mục đích ứng dụng khác nhau. Trong đó phần lớn cánh tay robot được sử dụng trong lĩnh vực điều khiển robot như lấp đặc IC vào bo mạch, hàn và sơn khung xe trên

những dây chuyền láp ráp, kiểm tra và sửa chữa cấu trúc trong lò phản ứng hạt nhân, thám hiểm dưới biển và thăm dò dưới lòng đất đòi hỏi những sự chính xác cao.Lĩnh vực của robot rất rộng từ những việc đơn giản như tháo lắp thiết bị điều khiển đến những việc phức tạp đòi hỏi sự an toàn chính xác trong môi trường khắc nghiệt như kiểm tra độ phóng xạ, thám hiểm vũ trụ....đòi hỏi phải ứng dụng những lý thuyết mới để tăng cường sự chính xác, giảm sai số. Sự phát triển gần đây của lý thuyết điều khiển hiện đại là trong nhiều lĩnh vực điểu khiển tối ưu của các hệ thống ngẫu nhiên và tiền định. Những robot gần đây áp dụng lý thuyết điều khiển hiện đại vào ngay cả những ngành kỹ thuật như: sinh học, y học...



Hình 1.2 Mô hình Robot Raven II- mô hình hiện đại được ứng dụng trong ngành Y khoa

Và một mô hình điển hình cho việc ứng dụng các bộ điều khiển hiện đại, thường được sử dụng trong các phòng thí nghiệm của các trường đại học, các trung tâm nghiên cứu là mô hình con lắc ngược. Hệ thống con lắc ngược là hệ thống phức tạp, không ổn định, có tính phi tuyến cao. Việc điều khiển giữ cho con lắc cân bằng hay thiết kế bộ điều khiển Swing-up là những vấn đề rất được quan tâm trong lĩnh vực điều khiển tự động. Con lắc ngược là loại robot ứng dụng một trong những vấn đề quan trọng lý thuyết điều khiển và được đề cập nhiều trong các tài liệu về điều khiển. Mô hình thực tế con lắc ngược có thể kiểm chứng lại các lý thuyết điều khiển như PID, Fuzzy, Neural Network, các lý thuyết điều khiển hiện đại... Tuy

nhiên con lắc ngược cũng đặt ra nhiều thách thức đối với lý thuyết điều khiển cũng như các thiết bị điều khiển chúng.

Dòng ARM STM32F4 là dòng vi điều khiển mới và rất mạnh của hãng STMicroelectronics. Đây là vi điều khiển 32-bit với bộ xử lý dấu chấm động (FPU) được tích hợp sẵn, cùng với cấu trúc DSP và tốc độ xử lý lên đến 168Mhz rất thích hợp cho các ứng dụng đòi hỏi nhanh và chính xác. Đặc biệt, ARM STM32F4 còn được hỗ trợ bởi phần mềm Debug chuyên dụng St-Link, giúp ta dễ dàng nhúng các giải thuật điều khiển từ các thư viện trên máy tính xuống vi điều khiển. STM32F4 có bộ nhớ Flash lên đến 1Mbyte và 192Kbytes trên chip RAM, 17 Timer 16-bit và 2 Timer 32-bit có tần số lên đến 168Mhz, tích hợp sẵn bộ điều rộng xung PWM và bộ đếm encoder rất phù hợp với nhu cầu điều khiển động cơ. Ngoài ra, vi điều khiển này còn hỗ trợ 24 kênh ngõ vào ADC 12 bits, 3 kênh giao tiếp I²C, 3 kênh truyền thông SPI 37.5Mbits/s, 2 giao diện CAN. Với những thuận lợi nên trên, nhóm đã tiến hành tìm hiểu về vi điều khiển ARM STM32F4 trên board mạch STM32F4 DISCOVERY của hãng STMircoelectronics và tiến hành thiết kế các bộ điều khiển để thực hiện đề tài "Điều khiển con lắc ngược quay".

1.2 Tình hình nghiên cứu trong và ngoài nước

Vì các đặc tính rất riêng của hệ thống con lắc ngược nên việc nghiên cứu và thiết kế bộ điều khiển cho mô hình này là một thách thức và rất thú vị trong lĩnh vực điều khiển tự động. Bên cạnh đó, với sự ứng dụng của mô hình trong nhiều lĩnh vực thực tế đời sống hiện đại ngày nay, mô hình con lắc ngược đang nhận được sự quan tâm của rất nhiều các ngành nghề khác nhau như quân sự, xây dựng, hàng không vũ trụ...

Đã có rất nhiều các công trình nghiên cứu khoa học, các báo cáo chuyên ngành trong và ngoài nước đề cập đến mô hình con lắc ngược này. Nhiều nhà nghiên cứu đã sử dụng các thuật toán điều khiển khác nhau để điều khiển hệ thống như thuật toán điều khiển PID, điều khiển trượt, thuật toán điều khiển tối ưu LQR, điều khiển logic mờ Fuzzy hay ứng dụng mạng Neural để điều khiển đã thu được một số thành công đáng kể. Từ thành công trong việc điều khiển con lắc ngược đơn giản với một bậc tự do, nhiều nhà nghiên cứu đã phát triển lên các bậc cao hơn như con lắc ngược 2 bậc tự do, 3 bậc tự do tự cân bằng, mở ra hướng phát triều ứng dụng cao hơn về điều khiển cân bằng trong thực tế cuộc sống.

1.3 Ý nghĩa thực tiễn của đề tài

Trong cuộc sống hiện đại ngày nay, nhiều ứng dụng thực tế của mô hình con lắc ngược như: xe hai bánh tự cân bằng, robot di chuyển bằng hay chân, ổn định tháp vô tuyến, ổn định giàn khoan ngoài biển, điều khiển cân bằng cầu trục nâng hạ hàng, điều khiển các thiết bị quân sự...và nhiều ứng dụng khác đang dần phổ biến và mang lại nhiều lợi ích cho con người. Các nghiên cứu và ứng dụng phổ biến của mô hình con lắc ngược tại Việt Nam là: điều khiển xe hai bánh tự cân bằng, công trình giảm dao động rung lắc giàn khoan DK1, điều khiển cầu trục nâng hạ container tại cảng...

Mặt khác, trong lĩnh vực nghiên cứu, con lắc ngược là mô hình phổ biến sử dụng để nghiên cứu phát triển các phương pháp điều khiển mới, so sánh và kiểm tra các lý thuyết điều khiển, tạo ra phương pháp học tập nghiên cứu bằng mô hình cụ thể.

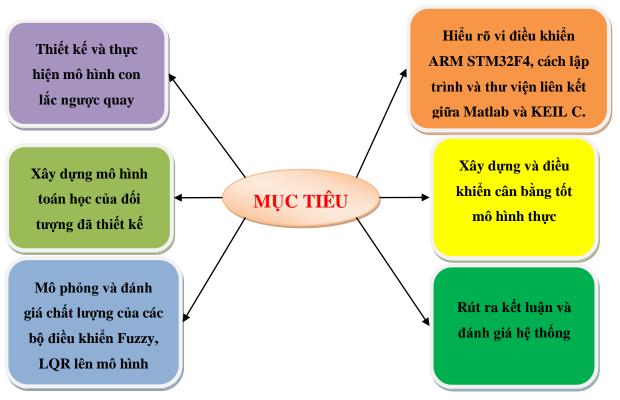




Hình 1.3 Ứng dụng của mô hình con lắc ngược

1.4 Mục tiêu của luận văn

Mục tiêu của đề tài là khảo sát phương trình động lực học, xây dựng mô hình toán học, mô phỏng đặc tuyến hoạt động của hệ con lắc ngược quay, tổng quan các công trình nghiên cứu về hệ con lắc ngược quay, khảo sát một số phương pháp điều khiển có thể áp dụng cho hệ con lắc ngược quay.



Hình 1.4 Các mục tiêu của luận văn

Đề tài đi sâu vào nghiên cứu các giải thuật điều khiển sử dụng công nghệ tính toán mềm ứng dụng cho hệ con lắc ngược quay , tiến hành khảo sát tổng hợp thiết kế điều khiển và xây dựng các mô hình mô phỏng của giải thuật điều khiển tính toán mềm ứng dụng vào hệ con lắc ngược, so sánh các kết quả mô phỏng đạt được về đặc tuyến làm việc, thời gian đáp ứng xác lập, sự ổn định của hệ thống rồi ứng dụng phương pháp vào chạy thực tế trên mô hình con lắc.

Kết quả mô phỏng sẽ cho chúng ta thấy ứng dụng giải thuật tính toán mềm vào hệ con lắc ngược. giúp chúng ta phân tích được những mặt ưu điểm và mặt khuyết điểm từ đó rút ra phương pháp hợp lý cho hệ con lắc ngược và ứng dụng vào điều khiển mô hình thực.

Dựa và mô hình mô phỏng, tiến hành thiết kế và xây dụng mô hình thực ứng dụng vào hệ con lắc ngược. Khảo sát chi tiết các thành phần cấu tạo nên mô hình thực của hệ con lắc

ngược và thiết kế thi công mô hình con lắc ngược quay . Xây dựng được mô hình thực của hệ con lắc ngược có nhúng thuật toán điều khiển sử dụng công nghệ tính toán mềm, giải quyết được những trường hợp nhiễu hệ thống và thực hiện được giải thuật swing-up, giải thuật giữ cân bằng được hệ con lắc ngược ở bất kì vị trí nào.

So sánh giữa kết quả lý thuyết và thực tiễn, ta tiến hành kiểm chứng lại vấn đề của công trình rồi xây dụng hướng phát triển của công trình hoàn thiện hơn. Đồng thời, mở rộng phạm vi ứng dụng của giải thuật điều khiển trên tất cả hệ thống thiếu cơ cấu truyền động. phương pháp hợp lý cho hệ con lắc ngược và ứng dụng vào điều khiển.

Bên cạnh đó, đề tài cũng nghiên cứu về các vi điều khiển hiện đại, các chương trình lập trình mới, các mạch điều khiển động cơ và truyền thông để ứng dụng vào việc điều khiển, đồng thời áp dụng các biện pháp xử lý tín hiệu để chống nhiễu nhằm mang lại chất lượng điều khiển tốt nhất.

1.5 Phương pháp nghiên cứu

Nội dung luận văn này nhằm đi sâu vào nghiên cứu hệ con lắc ngược một bậc tự do và các giải pháp điều khiển nó. Xây dựng mô hình mô phỏng hệ con lắc trên các giải pháp đó bằng phần mềm MATLAB để nghiên cứu đặc tính làm việc, thời gian xác lập của giải thuật đưa lên (swing- up) và giải thuật cân bằng hệ con lắc ngược ở vị trí bất kì. Đồng thời, dựa vào kết quả thu thập được qua quá trình mô phỏng của giải thuật điều khiển, ta tiến hành so sánh đánh giá ưu khuyết điển của các giải thuật mà nhóm đã sử dụng trong luận văn, từ đó cải thiện thêm phương pháp điều khiển mô hình con lắc ngược quay.

Sau khi đã thiết kế và thi công xong mô hình cơ khí con lắc ngược quay, ta sẽ xây dựng mô hình toán của đối tượng đã thiết kế. Ta sẽ dựa vào các thông số cơ khí của mô hình, các đặc tính truyền động của động cơ, chuyển động của cánh tay và con lắc để xây dựng nên mô hình toán chính xác nhất mô tả hệ thống con lắc ngược quay. Mô hình toán này có dạng hệ phương trình vi phân bậc hai, đòi hỏi kỹ năng ứng dụng tốt các kiến thức toán cao cấp để giải quyết bài toán. Tiếp đến, ta sẽ sử dụng chương trình mô phỏng Simulink của Matlab để mô phỏng mô hình toán đã xây dựng, sau đó đánh giá hệ thống và tiến hành thiết kế khâu điều khiển.

Trước khi thiết kế bộ điều khiển con lắc ngược quay, ta phải nghiên cứu kỹ các giải thuật điều khiển khác nhau để chọn ra những giải thuật, phương pháp điều khiển thích hợp nhất với mô hình đã xây dựng. Bài toán điều khiển mô hình này gồm hai bước: điều khiển đưa

con lắc ngược từ vị trí cân bằng dưới lên vị trí cân bằng thẳng đứng (Swing-up) và điều khiển cân bằng con lắc tại vị trí đó. Những người thực hiện đề tài này sẽ tiến hành nghiên cứu các phương pháp điều khiển và các bộ điều khiển đã được thiết kế, sau đó sẽ đánh giá và chọn ra bộ điều khiển phù hợp nhất hoặc tự thiết kế một bộ điều khiển mới cho mô hình.

Tiếp đến, ta sẽ mô phỏng mô hình toán và bộ điều khiển để đánh giá chất lượng, và tiến hành nhúng thuật toán để điều khiển mô hình thực tế khi quá trình mô phỏng đã được được các tiêu chuẩn chất lượng đề ra. Mô hình thực tế sẽ tự thu thập các số liệu hồi tiếp và các đáp ứng với bộ điều khiển, các đáp ứng đó sẽ được so sánh với các số liệu và đặc tuyến mô phỏng để đánh giá và chỉnh định các thông số. Sau khi chỉnh định, mô hình thực tế phải đáp ứng tốt với bộ điều khiển như đã mô phỏng.

1.6 Nội dung luận văn

Nội dung của luận văn gồm các chương sau:

Chương 1: Giới thiệu

Nội dung chương một trình bày sơ lược về các phương pháp điều khiển hiện đại của hệ thống con lắc ngược quay và tổng quan về các công trình nghiên cứu liên quan đến đề tài từ các nguồn tài liệu trong và ngoài nước. Chương này cũng trình bày những mục tiêu, nhiệm vụ và phương pháp nghiên cứu được sinh viên thực hiện trong đề tài này. Phần cuối của chương trình bày sơ lược nội dung của luận văn.

Chương 2: Cơ sở lý thuyết

Chương hai trình bày khái quát các cơ sở lý thuyết phương pháp điều khiển được sử dụng trong đề tài và mô hình toán học của mô hình. Đây là nền tảng cơ sở lý thuyết để xây dựng thuật toán điều khiển cho hệ con lắc ngược quay.

Chương 3: Thuật toán điều khiển mô hình con lắc ngược quay

Chương này trình bày phương pháp nghiên cứu mô hình toán học của con lắc ngược quay và xây dựng mô hình mô phỏng dựa vào mô hình toán trên chương trình Matlab Simulink. Chương ba cũng trình bày việc thiết kế giải thuật Swing-up và giải thuật điều khiển cân bằng dựa trên mô hình mô phỏng và tiến hành đánh giá, kiểm tra chất lượng của hệ thống điều khiển.

Chương 4: Thiết kế và thi công mô hình con lắc ngược quay

Chương bốn khái quát việc thiết kế và thi công mô hình con lắc ngược quay, từ phần cơ khí đến việc nhúng các giải thuật điều khiển lên vi điều khiển để chạy ứng dụng. Phần cuối của chương trình bày kết quả thực tế khi chạy mô hình và thu thập tín hiệu hồi tiếp để vẽ các đặc tuyến thực tế.

Chương 5: Thiết kế chương trình điều khiển

Chương này trình bày về thư viện Waijung liên kết Matlab với chương trình biên dịch Code C, giải thuật điều khiển đổ xuống vi điều khiển và giao diện người dùng trên máy tính. Đồng thời chương này cũng giải thích các giải thuật xử lý tín hiệu đầu vào và đầu ra của hệ thống con lắc ngược.

Chương 6: Kết quả – Kết luận – Hướng phát triển của luận văn.

Chương này trình bày về những kết quả đạt được trong luận văn , ưu điểm và nhược điểm của giải thuật điều khiển, những đóng góp và đề xuất hướng phát triển tiếp theo để hoàn thiện và mở rộng của luận văn.



Hình 1.5 Sơ đồ các chương trong luận văn

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Tổng quan

Chương hai trình bày việc nghiên cứu mô hình toán của đối tượng và các giải thuật điều khiển mô hình con lắc ngược quay. Giải thuật điều khiển được nghiên cứu trong luận văn này là phương pháp điều khiển tối ưu toàn phương tuyến tính LQR và phương pháp điều khiển mờ FUZZY- một phương pháp điều khiển thông minh. Nhóm lựa chọn hai phương pháp điều khiển này vì đây là hai phương pháp điều khiển hiện đại được sử dụng rộng rãi trong các ứng dụng điều khiển tự động ngày nay, và đây cũng là những phương pháp thích hợp nhất để điều khiển mô hình con lắc ngược quay.

2.2 Xây dựng mô hình toán đối tượng.

Trong điều khiển, dù là bộ điều khiển thiết kế là bộ điều khiển hiện đại hay thông minh, thì việc xây dựng mô hình đối tượng là điều rất cần thiết. Việc xây dựng mô hình toán đối tượng giúp chúng ta hiểu được các đặc tính của mô hình và dễ dàng hơn trong việc thiết kế bộ điều khiển cho đối tượng đó.

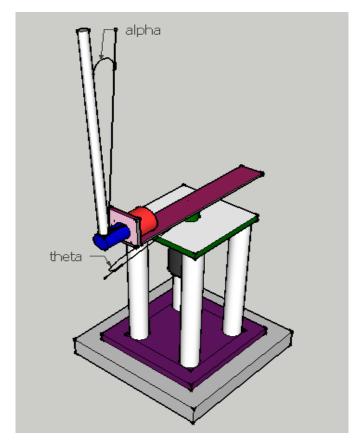
Hệ thống con lắc ngược bao gồm bộ điều khiển, cánh tay, con lắc, bộ chấp hành (động cơ DC) và hai increment encoder. Bộ điều khiển có nhiệm vụ giữ cân bằng con lắc trên cánh tay bằng cách quay cánh tay quanh trục được gắn trên giá đỡ. Động cơ cung cấp năng lượng để quay cánh tay. Vị trí cánh tay và con lắc được đọc từ encoder.

2.2.1 Mô hình vật lý

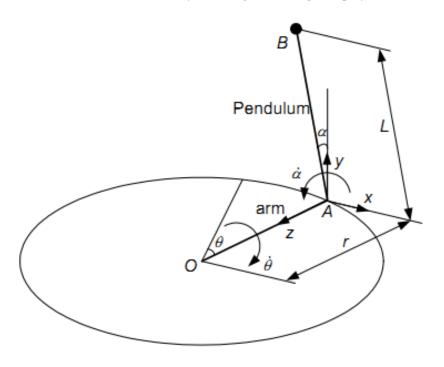
Những giả thiết khi phát triển mô hình vật lý:

- Các thanh rắn đồng chất.
- Hê hai bâc tư do
- Đáp ứng của các encoder đủ nhanh và có thể xem như là tức thời.
- Đáp ứng động cơ đủ nhanh và có thể xem như là tức thời.
- Để cho đơn giản, các hệ số ma sát được quy về tương đương ở khóp nối giữa cánh tay và động cơ.

Mô hình vật lý và mô hình cơ học đơn giản của hệ thống con lắc ngược được mô tả trên Hình 2.1 và Hình 2.2.



Hình 2.1 Mô hình vật lý hệ thống con lắc ngược quay



Hình 2.2 Mô hình đơn giản hệ con lắc ngược quay

Trên hình 2.2 ta có:

Con lắc chiều dài 2L, trọng tâm đặt tại điểm giữa B. Cánh tay có chiều dài 2r, trục quay đặt tại điểm giữa O. Con lắc sẽ dịch chuyển một góc α trong khi cánh tay quay một góc θ . Hệ tọa độ xyz được đặt ở điểm A. Con lắc sẽ quay trong mặt phẳng Axy. Ta sẽ dùng mô hình ở Hình 2.2 để thiết lập mô hình toán học của hệ con lắc ngược.

Các ký hiệu và thuật ngữ dùng trong biểu thức toán được trình bày ở Phụ lục A.

2.2.2 Mô hình toán

Phương trình vi phân động học mô tả hệ thống con lắc ngược quay được thiết lập dựa trên phương trình Euler-Lagrange của chuyển động:

$$\frac{d}{dt}(\frac{\partial T}{\partial \dot{q}_i}) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \tag{2.1}$$

Trong đó:

- $-q_i = \text{các tọa độ suy rộng}$
- Q_i = vector ngoại lực hay vector tải
- T = tổng động năng của hệ con lắc
- -V = tổng thế năng của hệ con lắc

Từ mô hình ở Hình 2.2, ta xác định được:

Động năng của cánh tay quay quanh trục O là:

$$T_0 = \frac{1}{2} J_{eq} \dot{\theta}^2 \tag{2.2}$$

Động năng của con lắc quay quanh A là:

$$T_{1} = \frac{1}{2}m(\dot{x}_{B}^{2} + \dot{y}_{B}^{2}) + \frac{1}{2}J_{B}\dot{\alpha}^{2}$$

$$= \frac{1}{2}m[(r\dot{\theta} - L\cos(\alpha)\dot{\alpha})^{2} + (-L\sin(\alpha)\dot{\alpha})^{2}] + \frac{1}{2}J_{B}\dot{\alpha}^{2}$$

$$= \frac{1}{2}mr^{2}\dot{\theta}^{2} + \frac{2}{3}mL^{2}\dot{\alpha}^{2} - mrL\dot{\theta}\cos(\alpha)\dot{\alpha}$$
(2.3)

Trong đó: $J_B = \frac{1}{12} m (2L)^2 = \frac{1}{3} m L^2$ là momen quán tính của con lắc đối với trục quay ở trọng tâm. Tổng động năng của hệ:

$$T = T_0 + T_1 = \frac{1}{2} (J_{eq} + mr^2) \dot{\theta}^2 + \frac{2}{3} mL^2 \dot{\alpha}^2 - mLr \cos(\alpha) \dot{\theta} \dot{\alpha}$$
 (2.4)

Tổng thế năng của hệ:

$$V = V_{pend} = mgL\cos(\alpha) \tag{2.5}$$

Do ta có hai tọa độ suy rộng θ và α nên theo phương trình Lagrange, ta có hai phương trình mô tả chuyển động của hệ:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}} \right) - \frac{\partial T}{\partial \theta} + \frac{\partial V}{\partial \theta} = M - B_{eq} \dot{\theta}$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\alpha}} \right) - \frac{\partial T}{\partial \alpha} + \frac{\partial V}{\partial \alpha} = 0$$
(2.6)

Thay (2.4) và (2.5) vào (2.6), ta được phương trình chuyển động của hệ:

$$(J_{eq} + mr^{2})\ddot{\theta} - mLr\cos(\alpha)\ddot{\alpha} + mLr\sin(\alpha)\dot{\alpha}^{2} = M - B_{eq}\dot{\theta}$$
$$-mLr\cos(\alpha)\ddot{\theta} + \frac{4}{3}mL^{2}\ddot{\alpha} - mgL\sin(\alpha) = 0$$
(2.7)

Trong đó:

$$M = T_{m} - J_{m} \ddot{\theta}_{m}$$

$$= \eta_{m} K_{t} I_{m} - J_{m} \ddot{\theta}$$

$$= \eta_{m} K_{t} \frac{V_{m} - K_{m} \ddot{\theta}}{R_{m}} - J_{m} \ddot{\theta}$$

$$= \frac{\eta_{m} K_{t}}{R_{m}} V_{m} - \frac{\eta_{m} K_{t} K_{m}}{R_{m}} \dot{\theta} - J_{m} \ddot{\theta}$$

$$(2.8)$$

Thay (2.8) vào (2.7), ta được mô hình phi tuyến của hệ:

$$a\ddot{\theta} - b\cos(\alpha)\ddot{\alpha} + b\sin(\alpha)\dot{\alpha}^2 + e\dot{\theta} = fV_m$$

$$-b\cos(\alpha)\ddot{\theta} + c\ddot{\alpha} - d\sin(\alpha) = 0$$
(2.9)

Trong đó:

$$a = J_{eq} + mr^{2} + J_{m}$$

$$b = mLr$$

$$c = \frac{4}{3}mL^{2}$$

$$d = mgL$$

$$e = B_{eq} + \frac{\eta_{m}K_{t}K_{m}}{R_{m}}$$

$$f = \frac{\eta_{m}K_{t}}{R_{m}}$$
(2.10)

Giải hệ (2.9) cho hai ẩn $\ddot{\alpha}$ và $\ddot{\theta}$ ta được :

$$\ddot{\theta} = \frac{1}{ac - b^2 \cos^2(\alpha)} \begin{vmatrix} -b \sin(\alpha)\dot{\alpha}^2 - e\dot{\theta} + fV_m & -b\cos(\alpha) \\ d \sin(\alpha) & c \end{vmatrix}$$

$$= \frac{1}{ac - b^2 \cos^2(\alpha)} (-bc \sin(\alpha)\dot{\alpha}^2 + bd \sin(\alpha)\cos(\alpha) - ce\dot{\theta} + cfV_m)$$

$$\ddot{\alpha} = \frac{1}{ac - b^2 \cos^2(\alpha)} \begin{vmatrix} a & -b \sin(\alpha)\dot{\alpha}^2 - e\dot{\theta} + fV_m \\ -b\cos(\alpha) & d\sin(\alpha) \end{vmatrix}$$

$$= \frac{1}{ac - b^2 \cos^2(\alpha)} (ad \sin(\alpha) - b^2 \sin(\alpha)\cos(\alpha)\dot{\alpha}^2 - be\cos(\alpha)\dot{\theta} + bf\cos(\alpha)V_m)$$
(2.11)

Tuyến tính hóa (2.9) với $\alpha \approx 0$ và $\dot{\alpha} \approx 0$, ta được mô hình tuyến tính:

$$a\ddot{\theta} - b\ddot{\alpha} + e\dot{\theta} = fV_m$$

$$-b\ddot{\theta} + c\ddot{\alpha} - d\alpha = 0$$
(2.12)

Giải hệ (2.12) cho $\ddot{\alpha}$ và $\ddot{\theta}$ ta được :

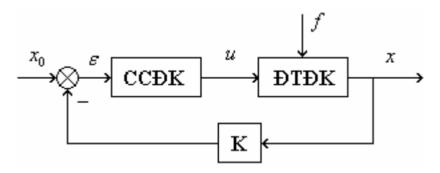
$$\ddot{\theta} = \frac{1}{ac - b^2} \begin{vmatrix} -e\dot{\theta} + fV_m & -b \\ d\alpha & c \end{vmatrix} = \frac{1}{ac - b^2} (bd\alpha - ce\dot{\theta} + cfV_m)$$

$$\ddot{\alpha} = \frac{1}{ac - b^2} \begin{vmatrix} a & -e\dot{\theta} + fV_m \\ -b & d\alpha \end{vmatrix} = \frac{1}{ac - b^2} (ad\alpha - be\dot{\theta} + bfV_m)$$
(2.13)

2.3 Phương pháp điều khiển LQR

Trong phần này chúng ta sẽ xem xét phương pháp xây dựng bài toán tổng hợp các hệ tuyến tính với chỉ tiêu chất lượng dạng toàn phương.

2.3.1 Khái niệm



Hình 2.3 Sơ đồ hệ thống điều khiển

Hệ thống điều khiển như hình 2.3 bao gồm các phần tử chủ yếu: đối tượng điều khiển (ĐTĐK), cơ cấu điều khiển (CCĐK), tín hiệu hồi tiếp (K) và các ký hiệu được chú thích trong phụ lục A.

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

Chỉ tiêu chất lượng J của một hệ thống có thể được đánh giá theo sai lệch của đại lượng được điều khiển x so với trị số mong muốn x0, lượng quá điều khiển (trị số cực đại xmax so với trị số xác lập $x(\infty)$ tính theo phần trăm), thời gian quá độ... hay theo một chỉ tiêu hỗn hợp trong điều kiện làm việc nhất định như hạn chế về công suất, tốc độ, gia tốc... Do đó việc chọn luật điều khiển và cơ cấu điều khiển để đạt được chế độ làm việc tối ưu còn tùy thuộc vào lượng thông tin ban đầu mà ta có được.

2.3.2 Ôn định Lyapunov đối với hệ thống tuyến tính – Tiêu chuẩn ổn định thứ hai của Lyapunov (điều kiện đủ)

Xét hệ thống được mô tả bởi phương trình trạng thái:

$$\dot{x} = f(x_1, x_2, x_3, x_4)$$

Nếu tìm được một hàm V(x) với mọi biến trạng thái $x_1, x_2...x_n$ là một hàm xác định dương, sao cho đạo hàm $\frac{dV(x)}{dt}$ của nó dựa theo phương trình vi phân của chuyển động bị nhiễu cũng là hàm xác định dấu, song trái dấu với hàm V(x) thì chuyển động không bị nhiễu sẽ ổn định tiệm cận.

V(x). $\dot{V}(x) < 0$: Với mọi biến trạng thái x_i , $i = \overline{1,n}$ thì hệ thống ổn định tiệm cận.

V(x). $\dot{V}(x) = 0$: Với mọi biến trạng thái x_i , $i = \overline{1, n}$ thì hệ thống ổn định.

V(x). $\dot{V}(x) > 0$: Với mọi biến trạng thái x_i , $i = \overline{1,n}$ thì hệ thống không ổn định.

Phương trình Lyapunov

Xét hệ tuyến tính mô tả bởi phương trình trạng thái:

$$\dot{x} = Ax$$

Yêu cầu cực tiểu hóa chỉ tiêu chất lượng J:

$$J = \int_0^\infty x^T \, Qx dt$$

với ma trận Q là ma trận vuông xác định dương.

Chon hàm năng lương V(x) xác đinh dương:

$$V(x) = x^T S x$$

trong đó ma trận S là ma trận vuông xác định dương. $\dot{V}(x)$ có dạng:

$$V(x) = \dot{x}^T S x + x^T S \dot{x} + x^T \dot{S} \dot{x}$$
$$= (Ax)^T S x + x^T S A x + x^T \dot{S} \dot{x}$$
$$= x^T A^T S x + x^T S A x + x^T \dot{S} \dot{x}$$
$$= x^T (A^T S + S A + \dot{S}) \dot{x}$$

Do V(x) xác định dương nên để hệ thống ổn định thì $\dot{V}(x)$ phải là xác âm. Ta chọn $V(x) = -x^T Q x$ (do Q là ma trận xác định dương nên $\dot{V}(x)$ sẽ là xác định âm).

$$\Rightarrow Q = -(A^TS + SA + \dot{S})$$

Điều kiện cần và đủ để trạng thái cân bằng x=0 ổn định tiệm cận: cho trước bất kỳ một ma trận xác định dương Q và ma trận A ổn định, tồn tại một ma trận xác định dương S thỏa mãn phương trình:

$$A^TS + SA + \dot{S} = -Q$$

$$\Rightarrow -\dot{S} = A^T S + S A + Q \qquad (1)$$

Phương trình (1) được gọi là phương trình Lyapunov.

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

$$0 = A^T S + SA + Q$$

Chỉ tiêu chất lượng J được tính như sau:

$$J = \int_0^\infty x^T Qx dt = -x^T(\infty) Sx(\infty) + x^T(0) Sx(0)$$

Khi tất cả các phần tử của ma trận A âm, ta có $x(\infty) \to 0$. Do đó:

$$I = x^{T}(0)Sx(0)$$

2.3.3 Điều khiển tối ưu hệ tuyến tính với chỉ tiêu chất lượng dạng toàn phương – Phương trình Riccati đối với hệ liên tục.

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

$$\dot{x} = Ax + Bu \tag{2}$$

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

$$u(t) = -Kx(t)$$
 (3)

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

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \qquad (4)$$

Trong đó: Q là ma trận xác định dương (hoặc bán xác định dương), R là ma trận xác định dương. Chú ý: thành phần $u^T R u$ ở phương trình trên xác định năng lượng tiêu tốn của tín hiệu điều khiển.

Chúng ta sẽ chứng minh luật điều khiển tuyến tính cho bởi phương trình (3) là luật điều khiển tối ưu. Khi đó, nếu ma trận K được xác định để tối thiểu hóa chỉ tiêu chất lượng J thì luật điều khiển u(t) sẽ tối ưu với mọi trạng thái ban đầu x(0).

Từ phương trình (2) và (3), ta có:

$$\dot{x} = Ax - BKx = (A - BK)x$$

Thay u(t) = -Kx(t) vào phương trình (4):

$$J = \int_0^\infty (x^T Qx + u^T K^T R K x u) dt$$
$$= \int_0^\infty (x^T Qx + K^T R K) x dt$$

Bây giờ ta chọn hàm năng lượng:

$$V(x) = x^T S x \ v \ a \ V(x) > 0. \ \forall x$$

Với S là ma trận vuông xác định dương.

$$V(x) = \dot{x}^T S x + x^T S \dot{x} + x^T \dot{S} \dot{x}$$

= $x^T (A - BK)^T S x + x^T \dot{S} \dot{x} + x^T \dot{S} \dot{x} + x^T \dot{S} (A - BK) x$
= $x^T [(A - BK)^T S x + \dot{S} + S(A - BK)] x$

Do V(x) xác định dương nên để hệ thống ổn định thì $\dot{V}(x)$ phải là xác định âm. Ta đặt:

$$\dot{V}(x) = \frac{d}{dt}(x^T S x) = -x^T (Q + K^T R K) x$$

(do Q và R là ma trận xác định dương nên ma trận $(Q + K^T RK)x$ cũng xác định dương, từ đó $\dot{V}(x)$ sẽ là ma xác định âm).

$$\Rightarrow x^T (Q + K^T R K) x = -x^T [(A - BK)^T S x + S(A - BK) + \dot{S}] x$$

$$Q + K^T R K = (A - BK)^T S x + \dot{S} + S(A - BK)$$
 (5)

Theo tiêu chuẩn ổn định thứ hai của Lyapunov, nếu ma trận (A-BK) ổn định thì tồn tại một ma trận xác định dương S thỏa mãn phương trình (5).

Chỉ tiêu chất lượng bây giờ có thể được xác định như sau:

$$J = \int_0^\infty (x^T Qx + u^T Ru) dt = -x^T(\infty) Sx(\infty) + x^T(0) Sx(0)$$

Lưu ý rằng $x(\infty) = 0$

$$\Rightarrow J = x(0)^T S x(0)$$

Đặt $R = T^T T(0)$ phương trình (5) trở thành:

$$(A^{T} - K^{T}B^{T})S + S(A - BK) + \dot{S} + Q + K^{T}T^{T}TK = 0$$

Phương trình trên có thể viết lại như sau:

$$A^{T}S + SA + [TK - (T^{T})^{-1}B^{T}S]^{T} [TK - (T^{T})^{-1}B^{T}S]S - SBR^{-1}B^{T}S + \dot{S} + Q = 0$$
 (6)

Chỉ tiêu chất lượng J đạt giá trị cực tiểu khi biểu thức:

 $x^T[TK - (T^T)^{-1}B^TS]^T[TK - (T^T)^{-1}B^TS]x$ đạt giá trị cực tiểu.

Hệ hở:
$$Q = 0$$
, $u_k^* = \frac{1-a}{b(1-a^{2N})} (r_N - a^N x_0) a^{N-k-1}$. Hệ ổn định: $r_N = 0$

Khi đó:

$$TK = (T^T)^{-1}B^TS$$

 $\Rightarrow K = T^{-1}(T^T)^{-1}B^TS = R^{-1}B^TS)$ (7)

Phương trình (7) cho ta ma trận tối ưu K. Như vậy, luật điều khiển tối ưu cho bài toán điều khiển tối ưu dạng toàn phương với chỉ tiêu chất lượng cho bởi phương trình (7) là tuyến tính và có dạng:

$$u(t) = -Kx(t) = -R^{-1}B^{T}Sx(t)$$

Ma trận S khi đó phải thỏa mãn phương trình (6), được viết lại như sau:

$$A^{T}S + SA - SBR^{-1}B^{T}S + Q = -\dot{S}$$
 (8)

Phương trình (8) được gọi là phương trình Riccati.

Khi S không thay đổi theo thời gian thì $\dot{S} = 0$, ta có phương trình đại số Riccati (ARE: Elgebraic Riccati Equation):

$$A^TS + SA - SBR^{-1}B^TS + Q = 0$$

2.3.4 Phương trình Riccati đối với hệ rời rạc

Xét hệ rời rạc:

$$x_{k+1} = A_k x_k + B_k u_k$$

Với $x_k \in R^n v$ à $u_k \in R^m$

Chỉ tiêu chất lượng J được định nghĩa trong khoảng [1,N] có dạng:

$$J_{i} = \sum_{k=1}^{N-1} (x_{k}^{T} Q_{k} x_{k} + u_{k}^{T} R_{k} u_{k})$$

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

$$u_k = -K_k x_k$$

Với K_k được xác định như sau:

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

Trong đó S_k phải thoải mãn phương trình:

$$S_k = A_k^T [S_{k+1} - S_{k+1} B_k (B_k^T S_{k+1} B_k + R_k)^{-1} B_k^T S_{k+1}] A_k + Q_k$$

Phương trình trên chính là phương trình Riccati cho hệ rời rạc. Khi $|S_k| \neq 0$ với $\forall k$, ta có thể dùng bổ đề ma trận nghịch đảo để viết lại phương trình như sau:

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

2.3.5 Các bước thiết kế phương pháp điều khiển toàn phương tuyến tính hệ thống liên tục

Bước 1: Thiết lập hệ phương trình trạng thái:

$$\begin{cases} \dot{x} = Ax + Bu \\ C = Dx \end{cases}$$

Xác định các thông số A, B, D.

Bước 2: Xác định ma trận trọng lượng Q, R từ chỉ tiêu chất lượng J cho dưới dạng toàn phương tuyến tính.

Bước 3: Tìm nghiệm S của phương trình Riccati:

Đối với hệ liên tục:

$$A^TS + SA - SBR^{-1}B^TS + O = -\dot{S}$$

Đối với hệ rời rạc:

$$S_k = A_k^T [S_{k+1} - S_{k+1} B_k (B_k^T S_{k+1} B_k + R_k)^{-1} B_k^T S_{k+1}] A_k + Q_k$$

Bước 4: Chỉ tiêu chất lượng tối ưu đối với hệ dừng:

$$J_{min} = x^T(0)Sx(0)$$

Bước 5: Luật điều khiển tối ưu:

Đối với hệ liên tục:

$$u = -R^{-1}B^TSx$$

Đối với hệ rời rạc:

$$u_k = -(B_k^T S_{k+1} B_k + R_k)^{-1} B_k^T S_{k+1} A_k x_k$$

2.4 Phương pháp điều khiển mờ

2.4.1 Tổng quan về Logic mờ

Từ năm 1965 đã ra đời một lý thuyết mới đó là lý

thuyết tập mờ (Fuzzy set theory) do giáo sư Lotfi A. Zadeh ở trường đại học Califonia - Mỹ đưa ra. Từ khi lý thuyết đó ra đời nó được phát triển mạnh mẽ qua các công trình khoa học của các nhà khoa học như: Các công ty của Nhật bắt đầu dùng lôgic mờ vào kỹ thuật điều khiển từ năm 1980. Nhưng do các phần cứng chuẩn tính toán theo giải thuật lôgic mờ rất kém nên hầu hết các ứng dụng đều dùng các phần cứng chuyên về logic mờ. Một trong những ứng dụng dùng lôgic mờ đầu tiên tại đây là nhà máy xử lý nước của Fuji Electric vào năm 1983, hệ thống xe điện ngầm của Hitachi vào năm 1987.



Hình 2.4 Giáo sư Lotfi A.Zadeh

Logic mờ là logic của con người, thể hiện được sự nhận định và kinh nghiệm của con người trong việc điều khiển đối tượng nên logic mờ có thể điều khiển đối tượng chỉ dựa trên những thông tin không đầy đủ để điều khiển đầy đủ đối tượng một cách chính xác nhất. Logic mờ nói về sự không chắc chắn, dùng để ước lượng và giải quyết các vấn đề phức tạp.

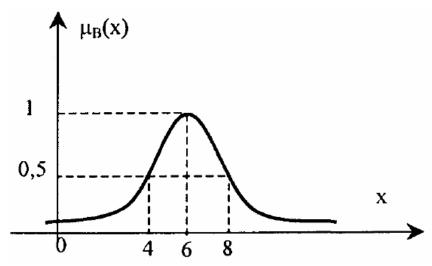
Để thực thi lôgic của con người trong kỹ thuật cần phải có một mô hình toán học của nó. Từ đó lôgic mờ ra đời như một mô hình toán học cho phép mô tả các quá trình quyết định và ước lượng của con người theo dạng giải thuật. Dĩ nhiên cũng có giới hạn, đó là lôgic mờ không thể bắt chước trí tưởng tượng và khả năng sáng tạo của con người. Tuy nhiên, lôgic mờ cho phép ta rút ra kết luận khi gặp những tình huống không có mô tả trong luật nhưng có sư tương đương. Vì vậy, nếu ta mô tả những mong muốn của mình đối với hệ thống trong những trường hợp cụ thể vào luật thì lôgic mờ sẽ tạo ra giải pháp dựa trên tất cả những mong muốn đó.

2.4.2 Khái niệm về tập mờ

Trong khái niệm tập hợp kinh điển hàm phụ thuộc $\mu A(x)$ của tập A, chỉ có một trong hai giá trị là "1" nếu $x \in A$ hoặc "0" nếu $x \notin A$. Cách biểu diễn hàm phụ thuộc như trên sẽ không phù hợp với những tập được mô tả "mờ" như tập B gồm các số thực gần bằng 5:

$$B = \{x \in R \mid x \approx 5\}$$

Khi đó ta không thể khẳng định chắc chắn số 4 có thuộc B hay không? mà chỉ có thể nói nó thuộc B gao nhiều phần trăm. Để trả lời được câu hỏi này, ta phải coi hàm phụ thuộc $\mu B(x)$ có giá trị trong khoảng từ 0 đến 1 tức là: $0 \le \mu B(x) \le 1$.



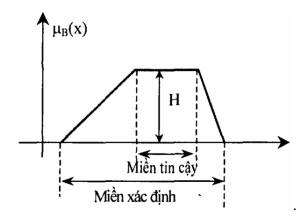
Hình 2.5 Hàm liên thuộc $\mu_B(x)$ của tập mờ B

Từ phân tích trên ta có định nghĩa: Tập mờ B xác định trên tập kinh điển M là một tập mà một phần tử của nó được biểu diễn bởi một cặp giá trị $(x,\mu B(x))$. Trong đó $x\in M$ và $\mu B(x)$ là ánh xạ. Ánh xạ $\mu B(x)$ được gọi là hàm liên thuộc của tập mờ B. Tập kinh điển M được gọi là cơ sở của tập mờ B.

Các thông số đặc trưng cho tập mờ:

Độ cao của một tập mờ B (Định nghĩa trên cơ sở M) là giá trị lớn nhất trong các giá trị
 của hàm liên thuộc.

$$H = \sup_{X \in M} \mu_B(x)$$



Hình 2.6 Độ cao, miền xác định, miền tin cậy của tập mờ

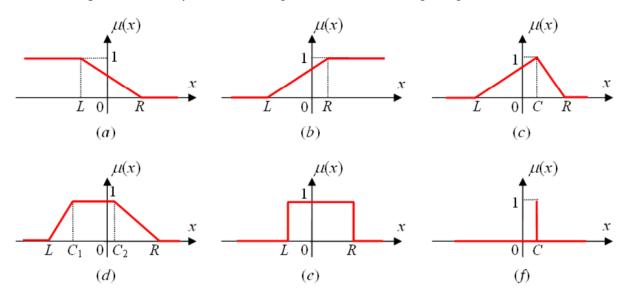
- Một tập mờ có ít nhất một phần tử có độ phụ thuộc bằng 1 được gọi là tập mờ chính tắc (H = 1). Ngược lại, một tập mờ B với H < 1 gọi là tập mờ không chính tắc.
- Miền xác định của tập mờ B (Định nghĩa trên cơ sở M) được ký hiệu bởi S là tập con của M có giá trị hàm liên thuộc khác không:

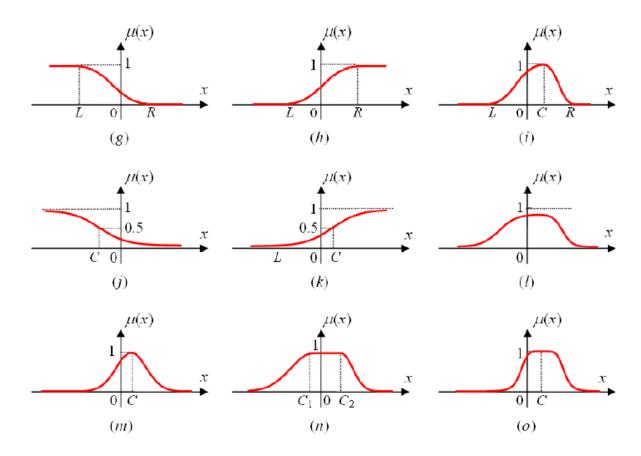
$$S = \{x \in M | \mu_B(x) > 0\}$$

Miền tin cậy của tập mờ B (định nghĩa trên cơ sở M) được ký hiệu bởi T, là tập con của M có giá trị hàm liên thuộc bằng 1:

$$T = \{x \in M | \mu B(X) = 1\}.$$

Các dạng hàm liên thuộc của tập mờ: Có rất nhiều cách khác nhau để biểu diễn hàm liên thuộc của tập mờ. Dưới đây là một số dạng hàm liên thuộc thông dụng:





Hình 2.7 Các dạng hàm liên thuộc của tập mờ

2.4.3 Các phép toán trên tập mờ và giải mờ

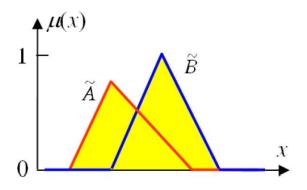
Trên tập mờ có 3 phép toán cơ bản là phép hợp, phép giao, và phép bù.

a. Phép hợp hai tập mờ

Hợp của hai tập mờ có cùng cơ sở: Hợp của hai tập mờ A và B có cùng cơ sở M là một tập mờ cùng xác định trên cơ sở M với hàm liên thuộc được xác định theo một trong các công thức sau:

$$\tilde{A} \cup \tilde{B} : \mu_{\tilde{A} \cup \tilde{B}}(x) = S\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}$$

S có thể là toán tử MAX(cực đại) hoặc toán tử BSUM (tổng bị chặn)



Hình 2.8 Hợp của hai tập mờ toán tử dùng toán tử MAX

b. Phép giao hai tập mờ

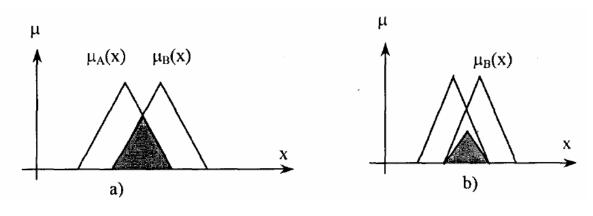
Giao của hai tập mờ A và B có cùng cơ sở M là một tập mờ cũng xác định trên cơ sở M với hàm liên thuộc $\mu A \cap B(x)$ được tính:

$$\tilde{A} \cup \tilde{B} : \mu_{\tilde{A} \cup \tilde{B}}(x) = T\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}$$

T có thể toán tử MIN(cực tiểu) hay toán tử PROD (tích)

$$- \mu A \cap B(x) = Min \{\mu A(x), \mu B(x)\};$$

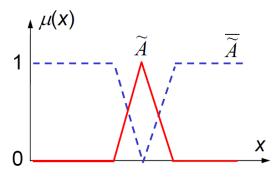
$$- \mu A \cap B(x) = \mu A(x) \cdot \mu B(x) \text{ (tích)};$$



Hình 2.9: Giao của hai tập mờ có cùng cơ sở theo quy tắc Min (a) và theo tích (b)

c. Phép bù tập mờ

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: $\mu_A(x) = 1 - \mu_A(x)$



Hình 2.10 Bù của tập mờ

2.4.4 Biến ngôn ngữ và giá trị của biến ngôn ngữ

Mộ biến có thể gán bởi các từ trong ngôn ngữ tự nhiên làm giá trị của nó gọi là biến ngôn ngữ. Một biến ngôn ngữ thường bao gồm 4 thông số: X, T, U, M. Với:

- X: Tên của biến ngôn ngữ;
- T: Tập của các giá trị ngôn ngữ;
- U: Không gian nền mà trên đó biến ngôn ngữ X nhận các giá trị rõ;
- M: Chỉ ra sự phân bố của T trên U.

Ví dụ miền các giá trị ngôn ngữ của biến "tốc độ" N

N = [rất chậm, chậm, trung bình, nhanh, rất nhanh].

Miền các giá trị vật lý $V = \{x \in R (x \ge 0)\}.$

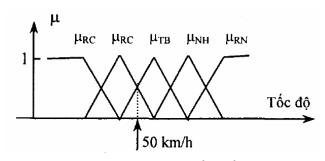
Mỗi giá trị ngôn ngữ (mỗi phần tử của Ni có tập nền là miền giá trị vật lý V. Từ một giá trị vật lý của biến ngôn ngữ ta có được một vécto μ gồm các độ phụ thuộc của x:

$$X \rightarrow \mu T = \left[\mu r \text{ \hat{a} t châm μ châm μ $trung b inh μ nhanh μ r \hat{a} t nhanh}\right]$$

ánh xạ trên được gọi là quá trình fuzzy hoá giá trị rõ x.

Ví dụ: ứng với tốc độ 50 km/h ta có:

Véc-to
$$\mu(50) = \begin{cases} 0 \\ 0.5 \\ 0.5 \\ 0 \\ 0 \end{cases}$$



Hình 2.11 Mờ hóa biến "tốc độ"

2.4.5 Luật hợp thành mờ

Xét hai biến ngôn ngữ χ và γ ; Biến χ nhận giá trị (mờ) A có hàm liên thuộc μ A(x) và γ nhận giá trị (mờ) B có hàm liên thuộc μ B(x) thì hai biểu thức: χ = A; γ = B được gọi là hai mệnh đề.

Luật Điều khiển: nếu $\chi = A$ thì $\gamma = B$ được gọi là mệnh đề hợp thành. Trong đó $\chi = A$ gọi là mệnh đề điều kiện và $\gamma = B$ gọi là mệnh đề kết luận. Một mệnh đề hợp thành có thể có nhiều mệnh đề điều kiện và nhiều mệnh đề kết luận, các mệnh đề liên kết với nhau bằng toán tử "và". Dựa vào số mệnh đề điều kiện và số mệnh đề kết luận trong một mệnh đề hợp thành mà ta phân chúng thành các cấu trúc khác nhau:

 Cấu trúc SISO (một vào, một ra): Chỉ có một mệnh đề điều kiện và một mệnh đề kết luân.

Ví dụ: nếu
$$\chi = A$$
 thì $\gamma = B$.

 Cấu trúc MISO (Nhiều vào, một ra): Có từ 2 mệnh đề điều kiện trở lên và một mệnh đề kết luân.

Ví dụ: nếu
$$\chi_1 = A_1 \text{ và } \chi_2 = A_2 \text{ thì } \gamma = B.$$

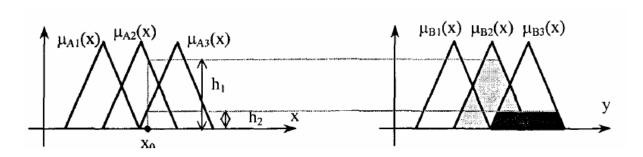
 Cấu trúc MIMO (Nhiều vào, nhiều ra): Có ít nhất 2 mệnh đề điều kiện và 2 mệnh đề kết luận.

Ví dụ: nếu
$$\chi_1 = A_1$$
 và $\chi_2 = A_2$ thì $\gamma_1 = B_1$ và $\gamma_2 = B_2$

Luật hợp thành là tên chung gọi mô hình R biểu diễn (một hay nhiều) hàm liên thuộc $\mu A => B(x, y)$ cho (một hay nhiều) mệnh đề hợp thành $A \Rightarrow B$. Một luật hợp thành chỉ có 1 mệnh đề hợp thành gọi là luật hợp thành đơn, có từ 2 mệnh đề hợp thành trở lên gọi là luật hợp thành phức.

Xét luật hợp thành R gồm 3 mệnh đề hợp thành:

- R1: Nếu x = A1 Thì y = B1 hoặc
- R2: Nếu x = A2 Thì y = B2 hoặc
- R3: Nếu x = A3 Thì y = B3 hoặc



Hình 2.12 Mô tả hàm liên thuộc của luật hợp thành

Với mỗi giá trị rõ x0 của biến ngôn ngữ đầu vào, ta có 3 tập mờ ứng với 3mệnh đề hợp thành R1, R2, R3 của luật hợp thành R. Gọi hàm liên thuộc của các tập mờ đầu ra là: $\mu_{B'_1}(y)$; $\mu_{B'_2}(y)$; $\mu_{B'_3}(y)$ thì giá trị của luật hợp thành R ứng với x0 là tập mờ B' thu được qua phép hợp 3 tập mờ: B' = B'1 \cup B'2 \cup B'3.

Tuỳ theo cách thu nhận các hàm liên thuộc $\mu_{B'_1}(y)$; $\mu_{B'_2}(y)$; $\mu_{B'_3}(y)$ và phương pháp thực hiện phép hợp để nhân tập mờ B' mà ta có tên gọi các luật hợp thành khác nhau:

- Luật hợp thành MAX-MIN nếu $\mu_{B'_1}(y)$; $\mu_{B'_2}(y)$; $\mu_{B'_3}(y)$ thu được qua phép lấy Min còn phép hợp thực hiện theo luật Max;
- Luật hợp thành MAX-PROD nếu $\mu_{B'_1}(y)$; $\mu_{B'_2}(y)$; $\mu_{B'_3}(y)$ thu được qua phép PROD còn phép hợp thực hiện theo luật Max;
- Luật hợp thành SUM-MIN nếu $\mu_{B'_1}(y)$; $\mu_{B'_2}(y)$; $\mu_{B'_3}(y)$ thu được qua phép lấy Min còn phép hợp thực hiện theo luật SUM;
- Luật hợp thành SUM PROD $\mu_{B'_1}(y)$; $\mu_{B'_2}(y)$; $\mu_{B'_3}(y)$ thu được qua phép lấy PROD còn phép hợp thực hiện theo Lukasiewicz.

Vậy, để xác định hàm liên thuộc $\mu B'(y)$ của giá trị đầu ra B' của luật hợp thành có n mệnh đề hợp thành R1, R2,... ta thực hiện theo các bước sau:

- + Xác định độ thoả mãn hj.
- + Tính $\mu_{B'}$; $\mu_{B'}$; $\mu_{B'}$ theo quy tắc Min hoặc Prod
- + xác định μ B'(y) bằng cách thực hiện phép hợp các $\mu_{B'}(y)$

2.4.6 Phương pháp giải mờ

Từ một giá trị rõ x₀ ở đầu vào, sau khi qua khối luật hợp thành ta có tập mờ đầu ra B'. Vấn đề đặt ra là cần phải xác định giá trị rõ y₀ từ tập mờ đầu ra đó. Muốn vậy ta cần thực hiện việc giải mờ.

Giải mờ là quá trình xác định một giá trị rõ y₀ nào đó có thể chấp nhận được từ hàm liên thuộc μ_{B'}(y) của giá trị mờ B' (tập mờ B'). Có hai phương pháp giải mờ chính là phương pháp cực đại và phương pháp điểm trọng tâm.

• Giải mờ theo phương pháp cực đại

Để giải mờ theo phương pháp cực đại, ta cần thực hiện 2 bước:

- Xác định miền chứa giá trị rõ y₀ (miền G): Đó là miền mà tại đó hàm liên thuộc μB'(y) đạt giá trị cực đại (độ cao H của tập mờ B'), tức là miền:

$$G = \{ y \in Y | \mu_{B'}(y) = H \}$$

- Xác định yo có thể chấp nhận được từ G. Hình 2.13 là tập mờ đầu ra của một luật hợp thành gồm 2 mệnh đề hợp thành:

$$R_1$$
: $N \in u \chi = A_1 Thi \gamma = B_1$

$$R_2$$
: N ế $u \chi = A_2 Thì \gamma = B_2$

Miền chứa giá trị rõ G là khoảng [y1, y2] của miền giá trị của tập mờ đầu ra B2 của luật điều khiển:

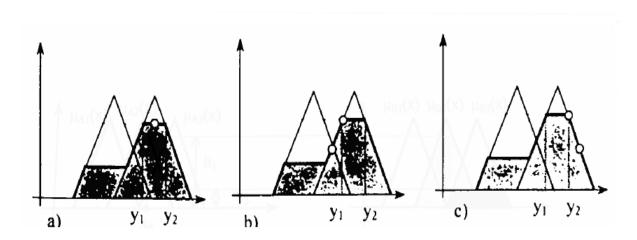
R2: Nếu
$$\chi = A2$$
 Thì $\gamma = B2$

với y1 là điểm cận trái của G $(y_1 = \inf_{y \in G}(y))$

và y2 là điểm cận phải của G $(y_2 = \sup_{y \in G}(y))$. Khi đó, luật R2 được gọi là luật Điều khiển quyết định.

Vậy luật điều khiển quyết định là luật Rk, $k \in \{1, 2, ..., p\}$ mà giá trị mở đầu ra của nó có độ cao lớn nhất (Bằng độ cao H của B').

Để xác định y0 trong khoảng [y1, y2] ta có thể áp dụng theo một trong ba nguyên lý: Nguyên lý trung bình; nguyên lý cận trái và nguyên lý cận phải.



Hình 2.13 Các nguyên lý giải mò theo phương pháp cực đại

a) Nguyên lý trung bình: giá trị rõ y₀ sẽ là trung bình cộng của y₁ và y₂

$$y_0 = \frac{y_1 + y_2}{2}$$

b) Nguyên lý cận trái: Giá trị rõ y0 được lấy bằng cận trái y1 của G

$$(y_1 = \inf_{y \in G}(y))$$

c) Nguyên lý cận phải: Giá trị rõ y0 được lấy bằng cận phải y2 của G

$$(y_2 = \sup_{y \in G}(y))$$

Phương pháp điểm trọng tâm

Giải mờ theo phương pháp điểm trọng tâm sẽ cho ra kết quả y' là hoành độ của điểm trọng tâm miền được bao bởi trục hoành và đường $\mu B'(y)$. Công thức xác định y0 theo phương pháp điểm trọng tâm như sau:

$$y' = \frac{\int_{S} y \mu_{B'}(y) dy}{\int_{S} \mu_{B'}(y) dy}$$

Với s là miền xác định của tập mờ B'.

O Phương pháp điểm trọng tâm cho luật hợp thành MAX-MIN

Giả sử có q luật điều khiển được triển khai. Khi đó mỗi giá trị mờ B' tại đầu ra của bộ điều khiển sẽ là tổng của q giá trị mờ đầu ra của từng luật hợp thành. Ký hiệu giá trị mờ đầu ra của luật điều khiển thứ k là μ B'K(y) với k = 1,2,...,q. Với quy tắc SUM- MIN, hàm liên thuộc μ B'(x) sẽ là:

$$\mu_{B'}(y) = \sum_{k=1}^{q} \mu_{B'k}(y)$$

Sau khi biến đổi, ta có:

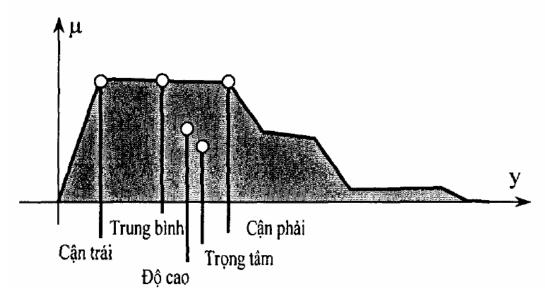
$$y' = \frac{\int_{S} \left[y \sum_{k=1}^{q} \mu_{B'k}(y) \right] dy}{\int_{S} \sum_{k=1}^{q} \mu_{B'k}(y) dy} = \frac{\sum_{k=1}^{q} \left[\int_{S} y \mu_{B'k}(y) \right] dy}{\sum_{k=1}^{q} \left[\int_{S} \left[\mu_{B'k}(y) \right] dy \right]}$$

Phương pháp độ cao

Sử dụng công thức

$$y' = \frac{\int_{S} \left[y \sum_{k=1}^{q} \mu_{B'k}(y) \right] dy}{\int_{S} \sum_{k=1}^{q} \mu_{B'k}(y) dy} = \frac{\sum_{k=1}^{q} \left[\int_{S} y \mu_{B'k}(y) \right] dy}{\sum_{k=1}^{q} \left[\int_{S} \left[\mu_{B'k}(y) \right] dy \right]}$$

cho cả hai luật hợp thành MAX-MIN và SUM-MIN với thêm một giả thiết là mỗi tập mờ μ B'K(y) được xấp xỉ bằng một cặp giá trị (yk, Hk) duy nhất (singleton), trong đó Hk là độ cao của μ B'K(y) và yk là một điểm mẫu trong miền giá trị của μ B'K(y).

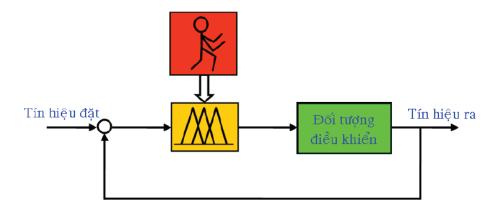


Hình 2.14 So sánh các phương pháp giải mờ

Chú ý: Tuỳ hình dạng hàm liên thuộc B' mà sai khác giữa các phương pháp giải mờ có khác nhau. Hình 2.14 cho biết kết quả các phương pháp giải mờ ứng với một hàm liên thuộc B' cụ thể.

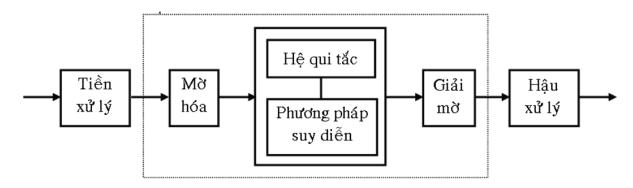
2.4.7 Cấu trúc bộ điều khiển mờ

Hoạt động của một bộ điều khiển mờ phụ thuộc vào kinh nghiệm và phương pháp rút ra kết luận theo tư duy của con người sau đó được cài đặt vào máy tính trên cơ sở logic mờ.



Hình 2.15 Mô hình điều khiển mờ

Bộ điều khiển mờ cơ bản bao gồm 3 khối cơ bản: khối mờ hóa, khối quy tắc và hệ suy diễn, và khối giải mờ.



Hình 2.16 Bô điều khiển mờ cơ bản

- **Khối mờ hoá** có chức năng chuyển mỗi giá tri rõ của biến ngôn ngữ đầu vào thành véctơ μ có số phần tử bằng số tập mờ đầu vào.
- -Thiết bị hệ quy tắc và phương pháp suy diễn mà bản chất của nó sự triển khai luật hợp thành R được xây dựng trên cơ sở luật điều khiển.
- **Khối giải mờ** có nhiệm vụ chuyển tập mờ đầu ra thành giá trị rõ y0 (ứng với mỗi giá tri rõ x0 đề điều khiển đối tượng.
- Khối tiền xử lý thực hiện việc tổng hợp và chuyển đổi tin hiệu vào (từ tương tự sang số), ngoài ra còn có thể có thểm các khâu phụ trợ đê thực hiện bài toán động như tích phân, vi phân....
- Khối hậu xử lý thực hiện chuyển đổi tín hiệu ra (từ số sang tương tự) để điều khiển đối tượng.

Nguyên tắc tổng hợp một bộ điều khiển mờ hoàn toàn dựa vào những phương pháp toán học trên cơ sở định nghĩa các biến ngôn ngữ vào/ra và sự lựa chọn những luật điều khiển. Do các bộ điều khiển mờ có khả năng xử lý các giá trị vào/ra biểu diễn dưới dạng dấu phẩy động với độ chính xác cao nên chúng hoàn toàn đáp ứng được các yêu cầu của một bài toán điều khiển "rõ ràng" và "chính xác"

Để tổng hợp được các bộ Điều khiển mờ và cho nó hoạt động một cách hoàn thiện ta cần thực hiện qua các bước sau:

1- Khảo sát đối tượng, từ đó định nghĩa tất cả các biến ngôn ngữ vào, ra và miền xác định của chúng. Trong bước này chúng ta cần chú ý một số đặc điểm cơ bản của đối tượng điều khiển như: Đối tượng biến đổi nhanh hay chậm? có trễ hay không? tính phi tuyến nhiều hay ít?,... Đây là những thông tin rất quan trọng để quyết định miền xác định của các biến ngôn ngữ đầu

vào, nhất là các biến động học (vận tốc, gia tốc,...). Đối với tín hiệu biến thiên nhanh cần chọn miền xác định của vận tốc và gia tốc lớn và ngược lại.

- 2- Mờ hoá các biến ngôn ngữ vào/ra: Trong bước này chúng ta cần xác định số lượng tập mờ và hình dạng các hàm liên thuộc cho mỗi biến ngôn ngữ. Số lượng các tập mờ cho mỗi biến ngôn ngữ được chọn tuỳ ý. Tuy nhiên nếu chọn ít quá thì việc điều chỉnh sẽ không mịn, chọn nhiều quá sẽ khó khăn khi cài đặt luật hợp thành, quá trình tính toán lâu, hệ thống dễ mất ổn định. Hình dạng các hàm liên thuộc có thể chọn hình tam giác, hình thang, hàm Gaus,...
- 3- Xây dựng các luật điều khiển (mệnh đề hợp thành): Đây là bước quan trọng nhất và khó khăn nhất trong quá trình thiết kế bộ điều khiển mờ. Việc xây dựng luật điều khiển phụ thuộc rất nhiều vào tri thức và kinh nghiệm vận hành hệ thống của các chuyên gia. Hiện nay ta thường sử dụng một vài nguyên tắc xây dựng luật hợp thành đủ để hệ thống làm việc, sau đó mô phỏng vả chỉnh định dần các luật hoặc áp dụng một số thuật toán tối ưu.
- 4- Chọn thiết bị hợp thành (MAX-MIN hoặc MAX-PROD hoặc SUMMIN hoặc SUM-PRROD) và chọn nguyên tắc giải mờ (Trung bình, cận trái, cận phải, điểm trọng tâm, độ cao).
- 5- Tối ưu hệ thống: Sau khi thiết kế xong bộ điều khiển mờ, ta cần mô hình hoá và mô phỏng hệ thống để kiểm tra kết quả, đồng thời chỉnh định lại một số tham số để có chế độ làm việc tối ưu. Các tham số có thể điều chỉnh trong bước này là. Thêm, bớt luật điều khiển; Thay đổi trọng số các luật; Thay đổi hình dạng và miền xác định của các hàm liên thuộc.

CHƯƠNG 3: MÔ PHỎNG ĐIỀU KHIỂN HỆ THỐNG CON LẮC NGƯỢC QUAY

3.1 Tổng quan

Chương ba trình bày quá trình thực hiện mô phỏng hệ thống con lắc ngược quay và các bộ điều khiển dựa trên các lý thuyết đã trình bày ở chương hai. Sau khi mô phỏng, những người thực hiện sẽ tiến hành đánh giá và chọn ra phương pháp điều khiển thích hợp nhất để ứng dụng trên mô hình thực tế.

3.2 Xây dựng mô hình mô phỏng hệ thống con lắc ngược quay

Mô hình mô phỏng hệ thống con lắc ngược quay được xây dựng trên Matlab Simulink dựa vào mô hình toán học đã xây dựng từ các phương trình (2.11) và (2.13).



Non-linear Rotary Inverted Pendulum

f(u)
ac-b^2cos(u1)

f(u)
bcsin(u1)u(3)^22

f(u)
bdsin(u1)cos(u1)

f(u)
alpha_dot_dot integrator alphazero

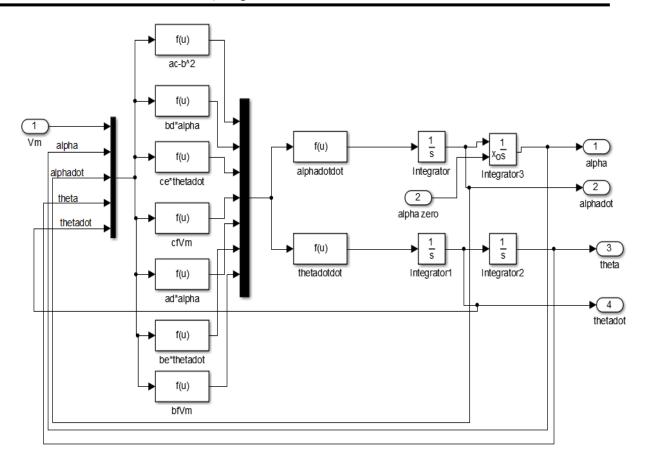
1
Vm

adsin(u1) f(u)

f(u) becos(u1)u(4) f(u)

Hình 3.1: Mô hình mô phỏng hệ con lắc ngược quay

Hình 3.2: Mô hình phi tuyến mô phỏng hệ con lắc ngược quay

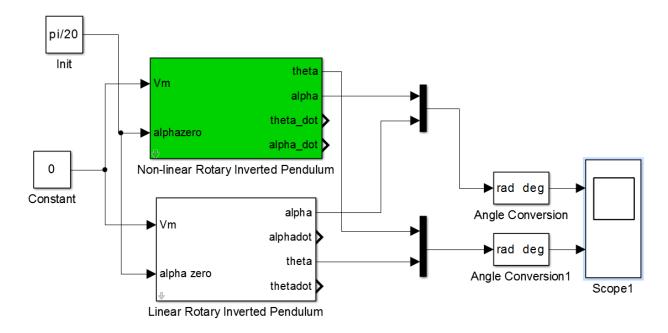


Hình 3.3: Mô hình toán tuyến tính hệ con lắc ngược quay

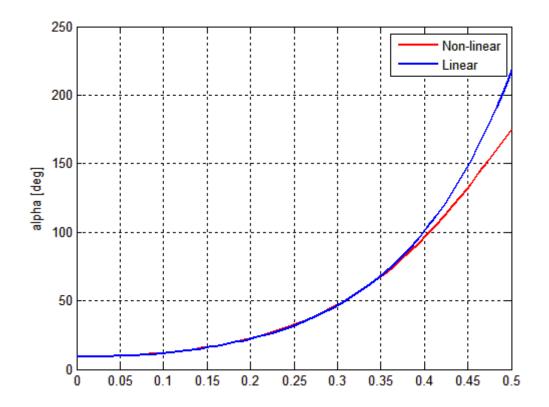
Để biết được độ chính xác của mô hình tuyến tính so với mô hình phi tuyến, ta tiến hành mô phỏng đặc tính động học của hệ thống sử dụng cả hai mô hình phi tuyến và tuyến tính sau đó so sánh kết quả mô phỏng với nhau. Việc mô phỏng vừa giúp ta kiểm chứng mô hình phi tuyến vừa có thể biết được ngưỡng của góc α trong mô hình tuyến tính.

Sơ đồ và kết quả mô phỏng được trình bày ở Hình 3.4 và Hình 3.5

Từ đồ thị ta có thể thấy mô hình tuyến tính mô tả đúng đặc tính động học của con lắc trong khoảng 0.3 giây đầu tiên. Phóng to khoảng 0.3 giây đầu tiên, ta thấy rằng mô hình phi tuyến có thể mô tả chính xác hệ thống trong khoảng 50 độ đầu tiên.



Hình 3.4 Mô hình mô phỏng khi chưa có bộ điều khiển

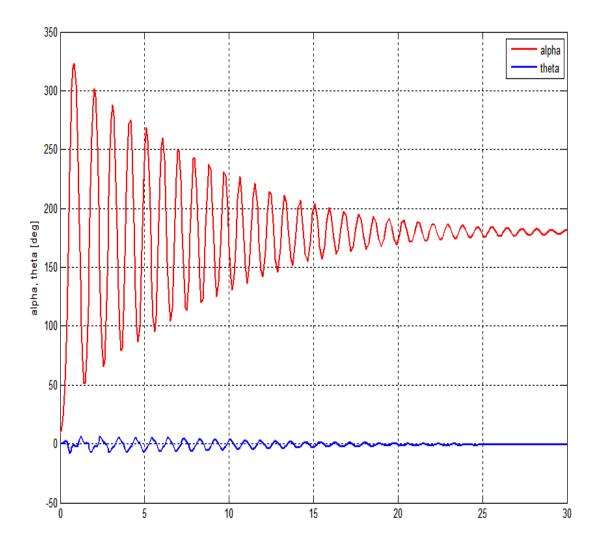


Hình 3.5 Kết quả mô phỏng mô hình tuyến tính và phi tuyến

Xét điều kiện đầu:
$$\theta = 0, \alpha = \frac{\pi}{18}, \dot{\theta} = 0, \dot{\alpha} = 0$$

Điện áp điều khiển: $V_a = 0$

Kết quả đáp ứng góc con lắc và góc cánh tay:



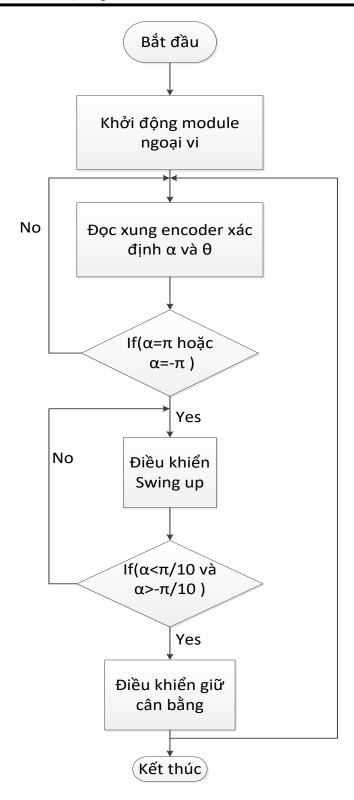
Hình 3.6 Đáp ứng vị trí con lắc và cánh tay khi không có bộ điều khiển

Khi chưa có bộ điều khiển, kết quả đáp ứng của con lắc như Hình 3.6, con lắc sẽ dao động tắt dần quanh điểm cân bằng ở dưới. Ta phải sử dụng bộ điều khiển hồi tiếp để điều khiển con lắc cân bằng ở trên, phương pháp sử dụng trong luận văn này là phương pháp điều khiển tối ưu toàn phương tuyến tính LQR và phương pháp điều khiển mờ.

3.3 Giải thuật điều khiển con lắc ngược quay

3.3.1 Lưu đồ giải thuật

Hình sau trình bày giải thuật điều khiển hệ thống con lắc ngược quay:



Hình 3.7 Lưu đồ giải thuật điều khiển hệ con lắc ngược quay

3.3.2 Giải thuật Swing up

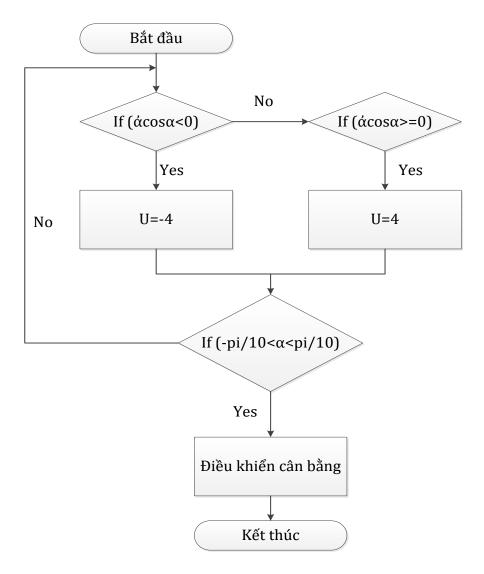
Bộ điều khiển Swing up được sử dụng để đưa con lắc từ vị trí cân bằng dưới lên vị trí cân bằng trên khi con lắc rời xa điểm cân bằng trên, ví dụ như khi con lắc rời do trọng lực. Bộ điều khiển sử dụng trong đề tài dựa trên bộ điều khiển được nghiên cứu bởi Astrom và Furuta

có tên "Swing up a pendulum by energy control", đó là phương pháp swing up dựa vào năng lượng để tối thiểu thời gian điều khiển.

Theo đó, luật điều khiển dựa trên năng lượng có dạng như sau:

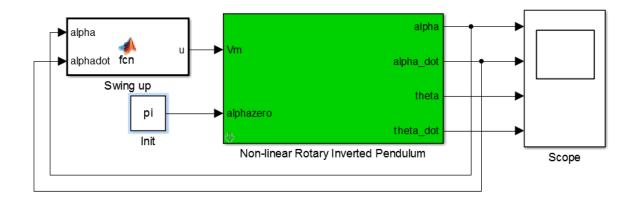
$$u(t) = -u_m sign(\dot{\alpha}(t)\cos(\alpha(t)))$$

Tín hiệu điều khiển có chiều phụ thuộc vào dấu của tích $\dot{\alpha}(t)\cos(\alpha(t))$, ta có lưu đồ giải thuật:

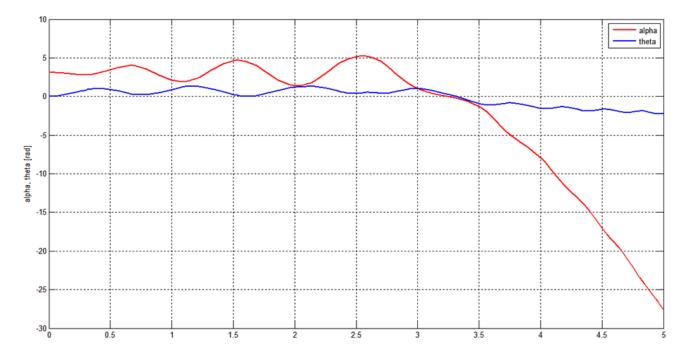


Hình 3.8 Lưu đồ giải thuật chương trình bộ điều khiển Swing up

Mô phỏng bộ điều khiển swing up trong Matlab Simulink



Hình 3.9 Mô phỏng bộ Swing-up trong Simulink



Hình 3.10 Đáp ứng của hệ qua bộ Swing up

Ta thấy trong khoảng 3.5 giây, bộ Swing up đưa con lắc từ vị trí π về vị trí 0. Do chưa có bộ điều khiển cân bằng nên con lắc chưa giữ cân bằng được. Tiếp sau ta sẽ thiết kế bộ điều khiển cân bằng để giữ cân bằng con lắc.

3.3.3 Giải thuật điều khiển cân bằng LQR

Hệ con lắc ngược là hệ thống có tính phi tuyến rất cao, tuy nhiên ở vị trí cân bằng, với góc lệch nhỏ, ta có thể tuyến tính hóa mô hình phi tuyến để đưa ra được mô hình tuyến tính, từ đó áp dụng bộ điều khiển LQR vào việc điều khiển cân bằng con lắc tại vị trí thẳng đứng hướng lên.

Xét mô hình tuyến tính (3.13):

$$\begin{split} \ddot{\theta} &= \frac{1}{ac - b^2} \begin{vmatrix} -e\dot{\theta} + fV_m & -b \\ d\alpha & c \end{vmatrix} = \frac{1}{ac - b^2} (bd\alpha - ce\dot{\theta} + cfV_m) \\ \ddot{\alpha} &= \frac{1}{ac - b^2} \begin{vmatrix} a & -e\dot{\theta} + fV_m \\ -b & d\alpha \end{vmatrix} = \frac{1}{ac - b^2} (ad\alpha - be\dot{\theta} + bfV_m) \end{split}$$

Đặt các biến trạng thái: $x_1 = \alpha, x_2 = \dot{\alpha}, x_3 = \theta, x_4 = \dot{\theta}$

Phương trình trạng thái phi tuyến:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{ad}{ac - b^2} & 0 & 0 & \frac{-be}{ac - b^2} \\ 0 & 0 & 0 & 1 \\ \frac{bd}{ac - b^2} & 0 & 0 & \frac{-ce}{ac - b^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{bf}{ac - b^2} \\ 0 \\ \frac{cf}{ac - b^2} \end{bmatrix} V_m$$

Thay số cụ thể hệ con lắc ngược quay với các giá trị trong bảng ở phụ lục A

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 64.3128 & 0 & 0 & -5.8248 \\ 0 & 0 & 0 & 1 \\ 8.6128 & 0 & 0 & -8.8942 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 2.8074 \\ 0 \\ 4.2867 \end{bmatrix} V_m$$

Thiết kế bộ điều khiển LQR

Giả thiết:

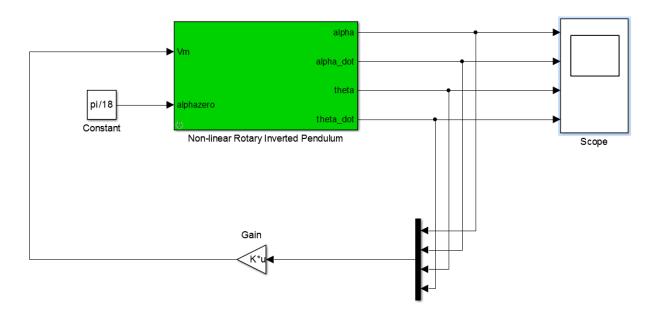
- Đặc tính động của hệ con lắc ngược có thể được mô tả bởi hệ phương trình biến trạng thái tuyến tính. Điều này đã được kiểm chứng ở phần trước.
- Hệ thống phản hồi trạng đầy đủ, nghĩa là có thể đo được 4 biến trạng thái (góc lệch α , góc quay θ , vận tốc góc lệch $\dot{\alpha}$ và vận tốc góc quay $\dot{\theta}$).
- Không có nhiễu tác động vào hệ thống.

Thiết kế dùng Matlab: >> K = lqr(A, B, Q, R)

- Tùy theo độ lớn tương đối giữa trọng số Q và R mà hệ thống có đáp ứng quá độ và năng lượng tiêu tốn khác nhau.
- Muốn hệ thống đáp ứng nhanh thì tăng thành phần Q tương ứng

- Muốn giảm năng lượng thì tăng R.

Mô phỏng điều khiển LQR hệ con lắc ngược quay trên Matlab Simulink:



Hình 3.11 Mô phỏng hệ con lắc ngược quay trên Matlab Simulink

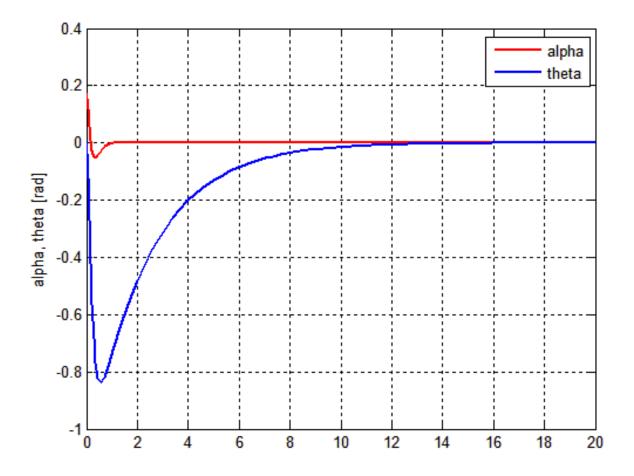
Kết quả mô phỏng điều khiển LQR hệ con lắc ngược quay:

Với:

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

$$R = 1$$

$$K = \begin{bmatrix} 104.2012 & 13.3756 & -1.0000 & -4.7431 \end{bmatrix}$$



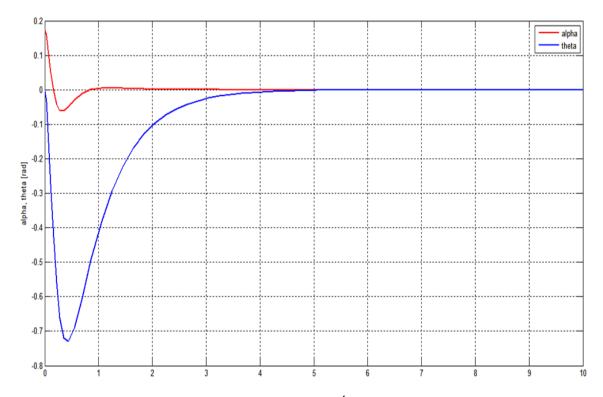
Hình 3.12 Đáp ứng của góc lệch con lắc và cánh tay

Góc lệch con lắc được giữ cân bằng tốt nhưng góc cánh tay dao động khá lớn.

Tăng trọng số q_{33} (ứng với góc lệch cánh tay): cánh tay dao động ít hơn

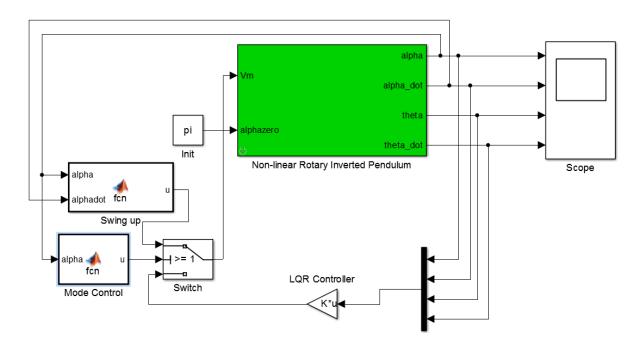
$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R = 1$$

$$K = \begin{bmatrix} 115.6256 & 14.8638 & -3.1623 & -5.5173 \end{bmatrix}$$



Hình 3.13 Đáp ứng của góc lệch con lắc và cánh tay khi tăng q_{33}

Kết quả khi tăng trọng số q_{33} ứng với vị trí cánh tay sẽ làm cho đáp ứng của vị trí cánh tay tăng lên và làm cho con lắc về vị trí cân bằng nhanh hơn.

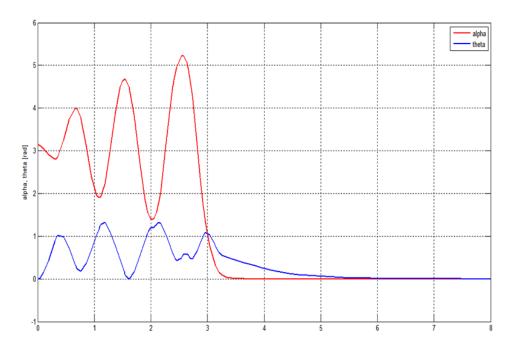


Hình 3.14 Mô phỏng kết hợp hai bộ điều khiển

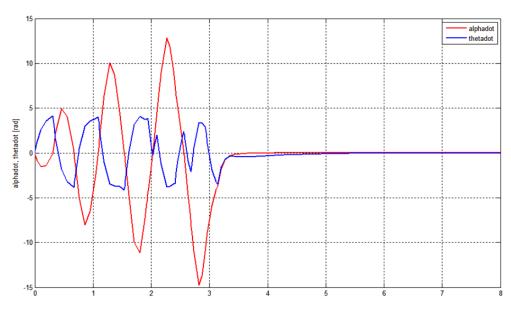
Mô hình mô phỏng kết hợp bộ điều khiển swing up và cân bằng được thể hiện chi tiết trong hình 3.14. Để chuyển đổi tín hiệu từ bộ điều khiển swing up sang bộ điều khiển cân

bằng ta sử dụng một khối Mode Control. Khối này có nhiệm vụ điều khiển khối Switch để chọn bộ điều khiển như sau: nếu góc alpha lớn hơn góc 20^{0} , hệ thống sẽ chạy thuật toán của bộ điều khiển swing up, ngược lại, nếu góc alpha nhỏ hơn 20^{0} hệ thống sẽ chạy thuật toán bộ điều khiển cân bằng.

Kết quả mô phỏng:



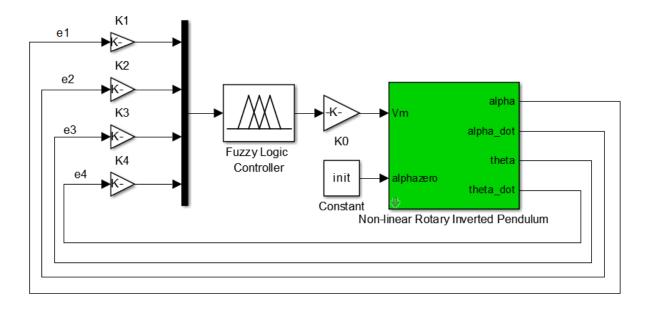
Hình 3.15 Kết quả mô phỏng vị trí góc con lắc và góc cánh tay



Hình 3.16 Kết quả mô phỏng vận tốc con lắc và vận tốc cánh tay

Kết quả mô phỏng cho thấy đáp ứng hệ thống rất tốt, con lắc từ vị trí cân bằng phía dưới đã được điều khiển tới vị trí cân bằng phía trên khá nhanh và ổn định.

3.3.4 Giải thuật cân bằng dùng bộ điều khiển mờ trực tiếp



Hình 3.17 Sơ đồ khối bộ điều khiển mờ trực tiếp điều khiển hệ con lắc ngược quay

Giống như bộ điều khiển LQR, bộ điều khiển mờ cho hệ con lắc ngược quay cũng có 4 ngõ vào và 1 ngõ ra. Bốn ngõ vào của bộ điều khiển mờ là sai số vị trí góc con lắc e_1 , sai số vận tốc con lắc e_2 , sai số vị trí cánh tay e_3 , sai số vận tốc cánh tay e_4 .

Yêu cầu đặt ra là thiết kế bộ điều khiển để giữ cân bằng con lắc ở vị trí thẳng đứng, tín hiệu đặt là vị trí của cánh tay. Ta chọn sơ đồ điều khiển mờ trực tiếp theo kinh nghiệm chuyên gia, nghĩa là so sánh vị trí cánh tay với tín hiệu đặt để điều khiển sao cho độ sai lệch giữa 2 tín hiệu này giảm về 0, trong khi đó vẫn phải giữ thăng bằng cho con lắc thẳng đứng.

Ta sử dụng bộ điều khiển mờ Mamdani có ngõ ra dạng singleton. Chọn 3 tập mờ cho mỗi biến vào, các tập mờ này được phân hoạch mờ trên tập cơ sở chuẩn hóa và hàm liên thuộc có dạng tam giác. Số tập mờ cho biến ngõ ra được chọn bằng 7.

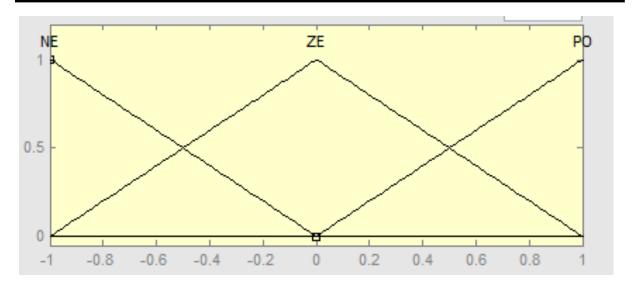
Ta sử dụng phương pháp suy diễn Max-Min và phương pháp giải mờ trọng tâm (COG).

Ta sẽ sử dụng các thông số của bộ LQR để thành lập bộ điều khiển mờ.

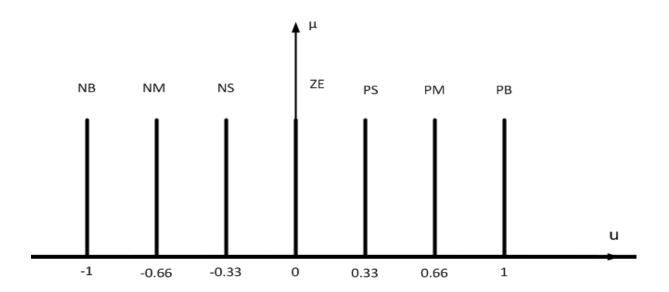
a. Giải thuật cân bằng dùng bộ điều khiển mờ trực tiếp

Chọn tập mờ cho các biến vào ra

Úng với mỗi biến vào, ta định nghĩa 3 giá trị ngôn ngữ là NE, ZE và PO. Biến ra được định nghĩa 7 giá trị ngôn ngữ NB, NM, NS, ZE, PS, PM, PB.



Hình 3.18 Các tập mờ của e_1 , e_2 , e_3 , e_4



Hình 3.19 Các tập mờ của Vm

Xây dựng luật mờ

Để xây dựng luật mò, ta xét từng trường hợp, sau đây là những trường hợp tiêu biểu:

Nếu góc lệch của con lắc so với phương thẳng đứng là ZE, vận tốc góc của con lắc là ZE, vị trí của cánh tay so với vị trí đặt là ZE và vận tốc góc cánh tay bằng ZE thì ta không kích hoạt động cơ, ngõ ra sẽ là ZE. Như vậy luật mờ sẽ được viết như sau:

Nếu
$$(e_1=ZE)$$
 và $(e_2=ZE)$ và $(e_3=ZE)$ và $(e_4=ZE)$ thì $(Vm=ZE)$

Nếu góc lệch của con lắc so với phương thẳng đứng là ZE, vận tốc góc của con lắc là ZE, vị trí của cánh tay so với vị trí đặt là NE và vận tốc góc cánh tay bằng NE thì ta cần kích hoạt động cơ chạy ngược lại một cách chậm rãi. Như vậy luật mờ sẽ được viết như sau:

$$N\tilde{e}u\ (e_1=ZE)\ va\ (e_2=ZE)\ va\ (e_3=NE)\ va\ (e_4=NE)\ thi\ (Vm=NM)$$

Nếu góc lệch của con lắc so với phương thẳng đứng là ZE, vận tốc góc của con lắc là ZE, vị trí của cánh tay so với vị trí đặt là PO và vận tốc góc cánh tay bằng ZE thì ta cần kích hoạt động cơ chạy thuận một cách chậm rãi. Như vậy luật mờ sẽ được viết như sau:

Nếu
$$(e_1 = ZE)$$
 và $(e_2 = ZE)$ và $(e_3 = PO)$ và $(e_4 = ZE)$ thì $(Vm = PS)$

Nếu góc lệch của con lắc so với phương thẳng đứng là PO, vận tốc góc của con lắc là NE, vị trí của cánh tay so với vị trí đặt là PO và vận tốc góc cánh tay bằng NE thì ta không cần kích hoạt động cơ. Như vậy luật mờ sẽ được viết như sau:

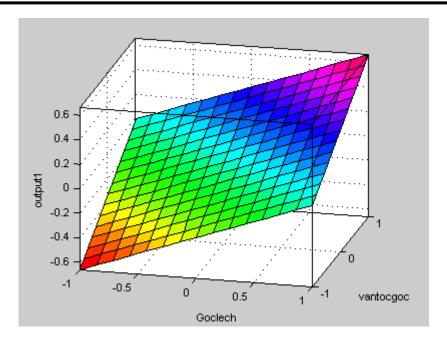
Nếu
$$(e_1 = PO)$$
 và $(e_2 = NE)$ và $(e_3 = PO)$ và $(e_4 = NE)$ thì $(Vm = ZE)$

Ta cứ lần lượt xét các trường hợp để xây dựng tập luật mờ

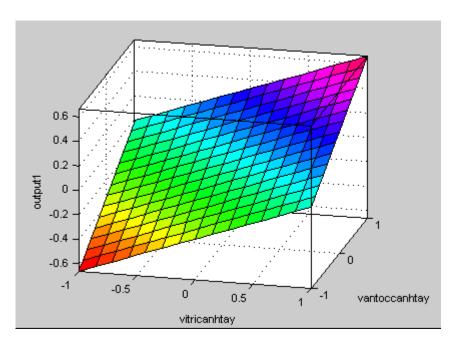
Bảng quy tắc giữ cân bằng hệ con lắc ngược:

STT	e_1	e_2	e_3	e_4	V_m	STT	e_1	e_2	e_3	e_4	V_m
1	NE	NE	NE	NE	NB	42	ZE	ZE	ZE	PO PO	PS
2	NE	NE	NE	ZE	NB	43	ZE	ZE	PO	NE	ZE
3	NE	NE	NE	PO	NM	44	ZE	ZE	РО	ZE	PS
4	NE	NE	ZE	NE	NB	45	ZE	ZE	РО	РО	PM
5	NE	NE	ZE	ZE	NM	46	ZE	PO	NE	NE	NS
6	NE	NE	ZE	PO	NS	47	ZE	PO	NE	ZE	ZE
7	NE	NE	PO	NE	NM	48	ZE	PO	NE	PO	PS
8	NE	NE	PO	ZE	NS	49	ZE	PO	ZE	NE	ZE
9	NE	NE	PO	PO	ZE	50	ZE	PO	ZE	ZE	PS
10	NE	ZE	NE	NE	NB	51	ZE	PO	ZE	PO	PM
11	NE	ZE	NE	ZE	NM	52	ZE	PO	PO	NE	PS
12	NE	ZE	NE	PO	NS	53	ZE	PO	PO	ZE	PM
13	NE	ZE	ZE	NE	NM	54	ZE	PO	PO	PO	PB
14	NE	ZE	ZE	ZE	NS	55	PO	NE	NE	NE	NM
15	NE	ZE	ZE	PO	ZE	56	PO	NE	NE	ZE	NS
16	NE	ZE	PO	NE	NS	57	PO	NE	NE	PO	ZE
17	NE	ZE	PO	ZE	ZE	58	PO	NE	ZE	NE	NS
18	NE	ZE	PO	PO	PS	59	PO	NE	ZE	ZE	ZE
19	NE	PO	NE	NE	NM	60	PO	NE	ZE	PO	PS
20	NE	PO	NE	ZE	NS	61	PO	NE	PO	NE	ZE
21	NE	PO	NE	PO	ZE	62	PO	NE	PO	ZE	PS
22	NE	PO	ZE	NE	NS	63	PO	NE	PO	PO	PM
23	NE	PO	ZE	ZE	ZE	64	PO	ZE	NE	NE	NS
24	NE	PO	ZE	PO	PS	65	PO	ZE	NE	ZE	ZE
25	NE	PO	PO	NE	ZE	66	PO	ZE	NE	PO	PS
26	NE	PO	PO	ZE	PS	67	PO	ZE	ZE	NE	ZE
27	NE	PO	PO	PO	PM	68	PO	ZE	ZE	ZE	PS
28	ZE	NE	NE	NE	NB	69	PO	ZE	ZE	PO	PM
29	ZE	NE	NE	ZE	NM	70	PO	ZE	PO	NE	PS
30	ZE	NE	NE	PO	NS	71	PO	ZE	PO	ZE	PM
31	ZE	NE	ZE	NE	NM	72	PO	ZE	PO	PO	PB
32	ZE	NE	ZE	ZE	NS	73	PO	PO	NE	NE	ZE
33	ZE	NE	ZE	PO	ZE	74	PO	PO	NE	ZE	PS
34	ZE	NE	PO	NE	NS	75	PO	PO	NE	PO	PM
35	ZE	NE	РО	ZE	ZE	76	РО	PO	ZE	NE	PS
36	ZE	NE	РО	PO	PS	77	РО	PO	ZE	ZE	PM
37	ZE	ZE	NE	NE	NM	78	РО	PO	ZE	PO	PB
38	ZE	ZE	NE	ZE	NS	79	РО	PO	PO	NE	PM
39	ZE	ZE	NE	PO	ZE	80	РО	PO	PO	ZE	PB
40	ZE	ZE	ZE	NE	NS	81	PO	PO	PO	PO	PB
41	ZE	ZE	ZE	ZE	ZE						

Bảng 3.1 Các quy tắc giữ cân bằng hệ con lắc ngược



Hình 3.20 Mặt đặc tính của bộ mờ với tín hiệu góc lệch và vận tốc góc con lắc



Hình 3.21 Mặt đặc tính bộ mờ với tín hiệu vị trí cánh tay và vận tốc cánh tay

Nhận xét: Mặt đặc tính của bộ điều khiển mờ có dạng tuyến tính

• Tính toán các hệ số xử lý

Dựa trên ý tưởng "sao chép" bộ điều khiển LQR, ta sẽ tính toán các hệ số xử lý theo các hệ số độ lợi của bộ LQR. Tăng các hệ số tiền xử lý dẫn đến tăng "độ lợi" của bộ mờ, tăng hệ số hậu xử lý tương ứng cũng làm tăng "độ lợi" của bộ mờ. Do đó, xấp xỉ độ lợi trên cặp ngõ vào thứ i-ngõ ra là K_iK_0 , vậy để sao chép độ lợi k_i của bộ LQR, ta chọn:

$$K_i K_0 = k_i$$

Ta đã có các thông số của bộ LQR là: $k_1 = -196.986, k_2 = -25.456, k_3 = 8.944, k_4 = 9.760$

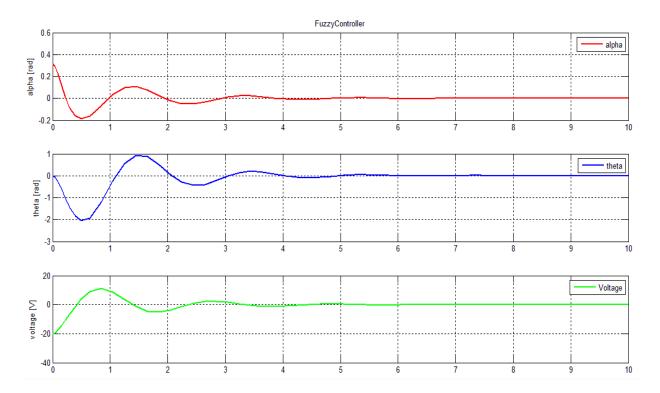
Với
$$Q = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix}, R = 1$$

- Đầu tiên ta sẽ chọn hệ số K_1 cho ngõ vào là α . Với khoảng giá trị α trong khoảng [-0.5, 0.5] rad thì $K_2=2$.
- Với $K_2 = 2$ ta được hệ số hậu xử lý là $K_0 = \frac{k_1}{K_1} = \frac{-196.968}{2} = 98.484$

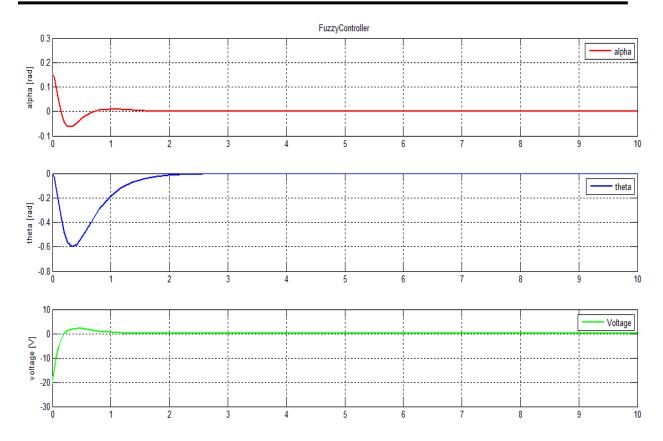
•
$$K_2 = \frac{k_2}{K_0} = \frac{-25.456}{-98.484} = 0.259$$

•
$$K_3 = \frac{k_3}{K_0} = \frac{8.944}{-98.484} = -0.091$$

•
$$K_4 = \frac{k_4}{K_0} = \frac{9.760}{-98.484} = -0.099$$



Hình 3.22 Kết quả mô phỏng bộ mờ với $K_0 = 98.484, \alpha = \frac{\pi}{20}$



Hình 3.23 Mô phỏng bộ điều khiển mờ với $K_0 = -200, \alpha = \frac{\pi}{20}$

Nhận xét: Khi tăng độ lớn của hệ số hậu xử lý thì đáp ứng bộ điều khiển mờ nhanh hơn.

b. Bộ điều khiển mờ 1 ngõ vào

Như trên ta thấy, để thiết kế bộ điều khiển mờ 4 ngõ vào, với mỗi ngõ vào gồm 3 tập mờ ta phải thiết lập 81 luật điều khiển, nếu tăng số tập mờ ở mỗi ngõ vào thì số luật điều khiển sẽ là rất lớn gây khó khăn trong việc thiết lập. Để giải quyết tình trạng trên, một nhóm nghiên cứu sinh ở trường đại học Kyungpook, Korea đã đề xuất phương pháp điều khiển mờ chỉ dùng 1 ngỗ vào.

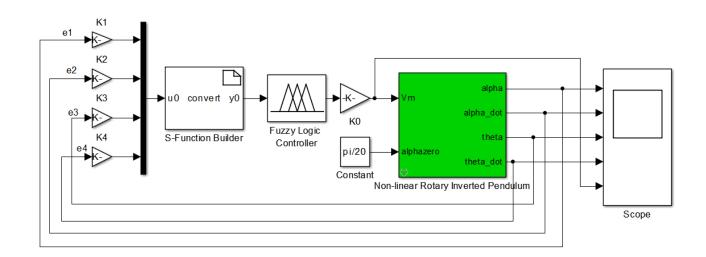
Theo đó, ta có thể biểu diễn 4 ngõ vào của bộ điều khiển mờ theo 1 ngõ vào d duy nhất như sau:

$$d = \frac{e_1 + \lambda_1 e_2 + \lambda_2 e_3 + \lambda_3 e_4}{\sqrt{1 + \lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

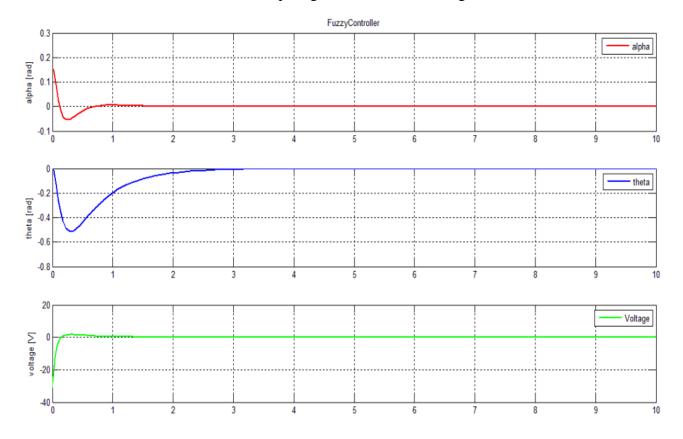
Từ đó, ta có luật mờ cho bộ điều khiển mờ 1 ngõ vào với 7 tập mờ:

d	NB	NM	NS	ZE	PS	PM	PB
V_{m}	NB	NM	NS	ZE	PS	PM	PB

Ta chọn các hệ số $\lambda_1=\lambda_2=\lambda_3=1$, các hệ số xử lý giống như đối với bộ mờ 4 ngõ vào. Kết quả mô phỏng được trình bày trên Hình 3. Ta thấy đáp ứng của hệ thống không khác nhiều so với bộ điều khiển mờ 4 ngõ vào, thậm chí đáp ứng còn tốt hơn.



Hình 3.24 Sơ đồ mô phỏng bộ điều khiển mờ 1 ngõ vào



Hình 3.25 Kết quả mô phỏng bộ điều khiển mờ 1 ngõ vào với $K_0 = -200, \alpha = \frac{\pi}{20}$

CHƯƠNG 4: THIẾT KẾ VÀ THỰC HIỆN MÔ HÌNH

Chương này trình bày thiết kế mô hình con lắc ngược quay, thiết kế phần cứng bao gồm phần cơ khí mô hình, các mạch điều khiển và mạch công suất, thiết kế phần mềm bao gồm chương trình nhúng cho vi điều khiển và giao diện người dùng. Phần cơ khí bao gồm khung giá đỡ và con lắc, động cơ điều khiển và 2 cảm biến encoder xác định vị trí. Phần mạch điện gồm vi điều khiển trung tâm, mạch cầu H để lái động cơ và mạch giao tiếp UART với máy tính. Và phần mềm được thiết kế dựa trên thư viện nhúng của chương trình Matlab 2013, còn phần giao diện người dùng được thiết kế trên chương trình Visual Studio 2010, sử dụng ngôn ngữ C# rất trực quan và sinh động.

4.1 Lua chon mô hình

Mô hình con lắc ngược là một đối tượng nghiên cứu rất phổ biến và có rất nhiều cấp độ và dạng khác nhau. Xét về cấp độ, mô hình đơn giản nhất là con lắc ngược một bậc tự do, phức tạp hơn thì có con lắc ngược 2 bậc tự do, 3 bậc tự do. Xét về dạng thì phổ biến nhất là mô hình con lắc ngược quay và mô hình xe con lắc ngược.

4.1.1 Mô hình xe con lắc ngược

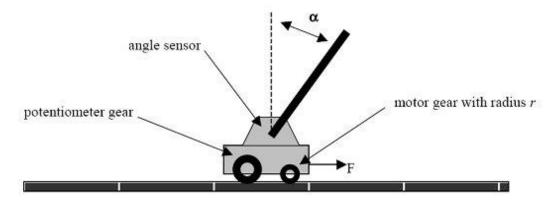
Mô hình xe con lắc ngược có hai dạng chính, đó là dạng xe chạy trên đường ray cố định và xe chạy tự do. Điểm chung của hai mô hình này là con lắc được gắn qua trục của encoder cố định trên xe, con lắc quay trên một mặt phẳng cố định.



Hình 4.1 Mô hình xe con lắc ngược quay trên ray cố định

Về tổng quan, mô hình xe con lắc ngược di chuyển trên ray cố định bao gồm một con lắc chuyển động tự do được gắn trên xe thông qua khớp xoay, xe có thể di chuyển dễ dàng dọc theo một thanh ray. Vị trí của chiếc xe này được điều khiển bởi 1 động cơ DC truyền động qua dây curoa. Các tín hiệu ngõ ra của hệ thống được hồi tiếp về bộ điều khiển là vị trí của chiếc xe và góc lệch của con lắc. Những khó khăn của việc thiết kế mô hình là các thiết bị của mô hình khó kiếm, kết cấu của hệ thống xe con lắc ngược cồng kềnh, phức tạp. Thi công cơ khí đòi hỏi khả năng chính xác và mô hình chiếm không gian lớn dành cho mô hình.

Còn mô hình xe con lắc ngược di chuyển tự do, xe gắn con lắc được gắn 4 bánh xe giống được truyền động từ động cơ thông qua cơ cấu hộp số. Xe chỉ di chuyển theo 1 phương cố định, chỉ tiến hoặc lui để giữ cân bằng con lắc tại vị trí thẳng đứng. Vị trí của chiếc xe cũng được xác định dựa trên cảm biến encoder gắn trên động cơ truyền động. Đây là mô hình khắc phục được các nhược điểm của mô hình xe con lắc di chuyển trên ray cố định, nhỏ gọn và tính di động cao. Nhưng do di chuyển tự do nên tính ổn định kém, dễ rung lắc khi xe di chuyển làm ảnh hưởng rất nhiều đến việc giữ cân bằng, đồng thời việc thiết kế mô hình xe cũng rất phức tạp, đòi hỏi phải tích hợp tất cả các mạch điều khiển lên xe và phải có nguồn điện dự trữ (pin) để cấp nguồn cho xe.



Hình 4.2 Mô hình xe con lắc ngược di chuyển tự do

4.1.2 Mô hình con lắc ngược quay

Mô hình con lắc ngược quay bao gồm một khung cố định dùng để gắn động cơ. Con lắc gắn trực tiếp với encoder được điều khiển bởi một cánh tay được gắn trực tiếp với trục động cơ. Ưu điểm của mô hình là dễ thiết kế, không bị giới hạn về không gian chuyển động của cánh tay nằm ngang, nhỏ gọn, chiếm không gian ít. Các dây tín hiệu trong hệ thống đi dây khá phức tạp là nhược điểm chính của mô hình.



Hình 4.3 Mô hình con lắc ngược quay

Như vậy, nếu thực hiện phép so sánh giữa hai mô hình thì thiết kế hệ con lắc ngược quay có thể đơn giản hơn nhờ giảm bớt một số chi tiết như công tắc hành trình, dây curoa, thanh ray, xe chở con lắc...Phần mạch điện của hệ thống được thiết kế đơn giản hơn so với hệ thống xe con lắc. Như vậy, **mô hình Con lắc ngược quay được những người thực hiện lựa chọn để thiết kế và điều khiển trong đề tài này.**

4.1.3 Các cấp mô hình con lắc ngược

Như đã đề cập ở trên, mô hình con lắc ngược có nhiều cấp khác nhau. Ở cấp cao hơn thì số bậc tự do của con lắc cũng cao hơn, đồng thời giải thuật điều khiển cũng phức tạp hơn rất nhiều. Ở mô hình con lắc ngược hai bậc tự do, ngoài một con lắc tự do quay như các mô hình ở trên, còn có thêm một con lắc tự do nữa gắn trên con lắc thứ nhất qua một cảm biến encoder thứ hai. Động cơ sẽ điều khiển vị trí của xe hoặc cánh tay, đồng thời phải giữ cân bằng cho cả hai con lắc tự do chồng lên nhau. Tương tự, con lắc ngược bậc ba thì có ba bậc tự do, con lắc thứ ba sẽ gắn trên con lắc thứ hai cũng qua một cảm biến encoder.



Hình 4.4 Mô hình con lắc ngược quay 2 bậc và 3 bậc tự do

Giải thuật và phương pháp điều khiển cho con lắc nhiều hơn một bậc rất phức tạp, đòi hỏi người nghiên cứu phải có một thời gian dài để tìm hiểu và thực hiện. Bên cạnh đó, theo nhận định của những người làm đề tài này, để điều khiển thật tốt các mô hình con lắc ngược nhiều bậc như trên thì phải thực hiện tốt mô hình đơn giản nhất trước, chính là mô hình con lắc ngược một bậc tự do. Vì thế, trong phạm vi của luận văn này, nhóm chỉ giới thiệu các mô hình con lắc ngược nhiều bậc mà không đi sâu vào tìm hiểu cũng như các giải thuật điều khiển.

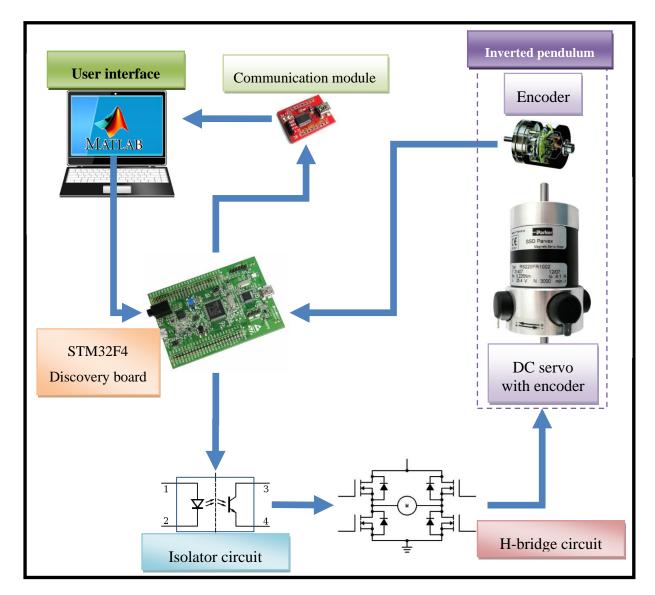
Những người thực hiện luận văn chọn thực hiện mô hình con lắc ngược một bậc tự do vì tính khả thi của đề tài so với khả năng hiện có của nhóm. Đồng thời với việc nghiên cứu giải thuật điều khiển con lắc ngược một bậc tự do, nhóm sẽ có được cái nhìn cơ bản nhất về con lắc ngược và dễ dàng hơn trong việc phát triển lên đề tài cao hơn là những mô hình nhiều bâc hơn.

4.2 Thiết kế mô hình con lắc ngược quay một bậc tự do

Hệ thống con lắc ngược quay một bậc tự do trong đề tài bao gồm:

- Khung đỡ bằng nhôm
- Động cơ DC servo gắn sẵn encoder dùng để truyền động trực tiếp cho cánh tay
- Encoder gắn trên cánh tay mô hình để xác định vị trí con lắc
- Con lắc được gắn trên trục encoder, quay trong mặt phẳng thẳng đứng

- Mach điều khiển STM32F4 DISCOVERY
- Mạch cách ly và mạch cầu H
- Mạch giao tiếp nối tiếp
- Máy tính với giao diện người dùng



- Khối xử lý gồm chip ARM STM32F4 DISCOVERY có nhiệm vụ nhận tín hiệu từ các encoder, xử lý thuật toán điều khiển và xuất tín hiệu điều khiển đồng thời gửi dữ liệu lên máy tính để hiển thị và đánh giá.
- Khối cảm biến gồm 2 Encoder để đo vị trí và vận tốc của con lắc và cánh tay rồi gửi tín hiệu về mạch điều khiển.
- Khối giao tiếp UART có nhiệm vụ thu nhận dữ liệu gửi lên từ khối xử lý và truyền
 lên máy tính để thu thập dữ liệu hiển thị trên giao diện người dùng.

- Khối cách ly có nhiệm vụ cách ly mạch điều khiển xử lý tín hiệu) với mạch cầu H
- điều khiển đông cơ.
- Khối động lực là mạch cầu H điều khiển động cơ.
- Máy tính chỉ có nhiệm vụ nạp chương trình xuống vi điều khiển và thu thập dữ liệu điều khiển được truyền về từ khối giao tiếp nối tiếp rồi hiển thị qua giao diện người dùng, không tham gia vào việc điều khiển mô hình.

4.3 Các thiết bị trong mô hình

4.3.1 Mô hình cơ khí

Nhận thấy thiết kế cơ khí là một phần quan trọng bậc nhất trong đề tài,vì thế người thực hiện đã dành nhiều thời gian để tìm hiểu các mô hình con lắc ngược quay trên thế giới cũng như trong nước để tham khảo và từ đó tiến hành thiết kế . Mô hình cơ khí hệ thống con lắc ngược phải đảm bảo chắc chắn,có các thông số hợp lý cũng như mang tính thẩm mỹ.

Mô tả	Thông số mô hình				
Khối lượng con lắc	55.62g				
Chiều dài con lắc	32 cm				
Chiều dài cánh tay	22.5 cm				
Khối lượng cánh tay	179.57 g				
Độ cao mô hình	43 cm				
Khối lượng mô hình	12.5 Kg				

Bảng 4.1 Thông số cơ bản của mô hình con lắc ngược quay

4.3.2 Động cơ

Có 3 loại động cơ có thể thích hợp cho đề tài này đó là: động cơ bước, động cơ DC servo và động cơ RC servo. Hai yếu tố quan trọng cần phải xem xét khi lựa chọn động cơ đó là phải có tốc độ lớn và moment lớn. Động cơ phải có tốc độ lớn để có thể chuyển động nhanh hơn sự rơi của con lắc, động cơ phải có moment lớn để có thể đảo chiều nhanh, từ đó mới có thể giữ cân bằng con lắc. Cùng với đó là yêu cầu có thể hồi tiếp được tốc độ (hay vị trí) của trục động cơ. Việc lựa chọn động cơ cũng còn liên quan đến việc lựa chọn công suất nguồn cung cấp.

- Động cơ bước có thể có moment lớn nhưng không đạt yêu cầu về tốc độ.
- Động cơ RC servo thì ngược lại có thể đạt được tốc độ lớn nhưng bị hạn chế về moment. Cùng với đó là động cơ servo chỉ có thể quay trong phạm vi $\pm 180^{\circ}$.
 - Động cơ DC servo có thể có được tốc độ lớn cùng với moment lớn.

Như vậy, động cơ DC servo là phù hợp cho ứng dụng này. Sau khoảng thời gian lựa chọn và tìm kiếm thì người thực hiện đề tài đã chọn được động có DC servo của hãng SANYO có thông số như sau: 24VDC, 3000rpm, 23W,encoder 1000p/r, phù hợp cho nhu cầu của luận văn.



Hình 4.5 Động cơ DC Servo

4.3.3 Cảm biến góc quay – encoder

Để phát hiện vị trí của cánh tay cũng như của con lắc thì có thể sử dụng các loại cảm biến *cảm biến góc quay, encoder tương đối, encoder tuyệt đối, cảm biến vận tốc góc*.

- Cảm biến góc quay (Rotation Angle Sensor –Potentionmeter): Ngõ ra là điện áp, như vậy cần phải thiết kế mạch xử lý tín hiệu trước khi đưa vào ngõ vào của ADC để xử lý. Bên cạnh đó cũng cần lưu ý rằng, giá trị điện áp tại ngõ ra có thể quan hệ phi tuyến với góc quay và khả năng đáp ứng của bộ điều khiển có thể bị giới hạn bởi tần số lấy mẫu của bộ ADC.
- Cảm biến vận tốc góc (gyro): cảm biến vận tốc góc cũng có hai loại ngõ ra là analog hoặc loại truyền theo các chuẩn I2C, SPI. Như vậy, nếu chúng ta sử dụng cảm biến gyro thì có thể chọn theo hai loại để mua. Nếu sử dụng loại ngõ ra là analog cần phải thiết kế mạch xử lý tín hiệu trước khi đưa vào ngõ của ADC để xử lí trước khi đưa vào bộ lọc Kagman. Nếu sử dụng loại truyền theo chuẩn I2C, SPI chúng ta sẽ phân tích tín hiệu trước khi đưa vào bộ lọc Kagman. Một điều quan tâm nữa là giá thành của gyro mắc.

- Encoder tương đối (Incremental Encorder):có thể sử dụng module QEP trên STM32F4 DISCOVERY, điểm mạnh của chip ARM STM32F4 là có bộ nhân 4 cho mỗi xung nên không đòi hỏi encoder tương đối phải có độ phân giải cao.
- Encoder tuyệt đối (Absolute Encorder): Là lý tưởng cho những ứng dụng xác định góc quay với khả năng có thể "nhớ" chính xác vị trí của trục quay. Tuy nhiên cũng có thể cân nhắc bởi giá thành của nó trên thị trường.

Do ảnh hưởng của một số yếu tố khách quan (cảm biến góc quay lại rất "hiếm" trên thị trường, điều kiện tài chính đối với Encorder tuyệt đối và cảm biến vận tốc góc ...), nên Incremental Encorder là sự lựa chọn trong ứng dụng này.



Hình 4.6 Encoder tương đối

Bảng 4.2 Thông số cơ bản của encoder sử dụng trong đề tài

	Thông số encoder sử dụng trong đề tài							
Encoder	Độ phân giải	Ngõ ra	Tần số hoạt động tối đa	Tốc độ tối đa	Moment trên trục	Nguồn cung cấp		
ROTARY ENCODER	1024 xung/vòng	Open collector	75KHz	3000rpm	3g.cm ²	5VDC		

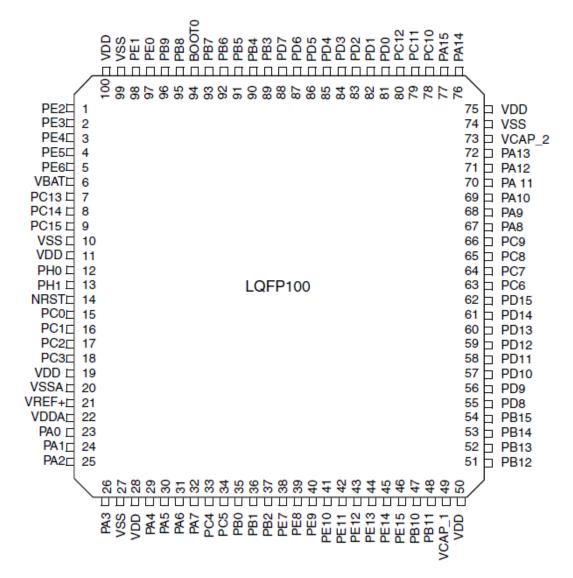
4.3.4 Mạch điều khiển

Đề tài sử dụng Chip **ARM Cortex-M4 STM32F4** (**LQFP100**) của hãng STMicroelectronics làm khối xử lý trung tâm để nhúng các giải thuật điều khiển. Dữ liệu chưa chuẩn hóa từ các cảm biến encoder (đo vị trí và vận tốc góc của hai thanh) đưa qua những khối tiền xử lý sẽ được chuẩn hóa và thu được thông tin về vị trí và vận tốc góc quay của hai thanh trong hệ con lắc ngược quay.

Tổng quan về vi điều khiển STM32F407

- ARM 32-bit CortexTM-M4 CPU với FPU và Adaptive real-time accelerator (xử lý thời gian thực).
- Tần số hoạt động lên đến 168MHz nên xử lý rất nhanh
- Hỗ trợ cấu trúc DSP
- Bộ nhớ trên chip: lên đến 1Mbyte bộ nhớ Flash, 192+4 Kbytes SRAM bao gồm
 64Kbytes CCM (core coupled memory)

- Cung cấp nguồn ứng dụng và I/Os từ 1.8 3.6V
- 3 bô ADC 12-bit, 2 bô DAC 12-bit.
- 12 timers 16-bit and 2 timers 32-bit lên đến 84 MHz, có 4 chức năng cơ bản là
 IC/OC/PWM hay đếm xung tốc độ cao (incremental) từ encoder.
- Có đến 15 kênh giao tiếp:
 - o 3 kênh giao tiếp I²C (SMBus/PMBus)
 - 4 kênhUSARTs/2 kênh UARTs (10.5 Mbit/s, ISO 7816 interface, LIN, IrDA, modem control)
 - o 3 kênh SPIs (37.5 Mbits/s)
 - o 2 CAN interfaces (2.0B Active)



Hình 4.7 Sơ đồ chân của vi điều khiển ARM STM32F4

Trong luận văn này, nhóm sử dụng các khối Timer để đọc encoder và phát xung PWM cho động cơ qua các chân I/O của vi điều khiển và 1 kênh UART để truyền nhận dữ liệu nối

tiếp giữa vi điều khiển và máy tính. Các thông số cơ bản của các khối Timer và UART được trình bày trong các bảng sau:

Bảng 4.3 Thông số các bộ Timer của vi điều khiển ARM STM32F4

Timer type	Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/ compare channels	Complementary output	Max interface clock (MHz)	Max timer clock (MHz)
Advanced- control	TIM1, TIM8	16-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	Yes	84	168
	TIM2, TIM5	32-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	No	42	84
General purpose	TIM3, TIM4	16-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	No	42	84
	TIM9	16-bit	Up	Any integer between 1 and 65536	No	2	No	84	168
	TIM10, TIM11	16-bit	Up	Any integer between 1 and 65536	No	1	No	84	168
	TIM12	16-bit	Up	Any integer between 1 and 65536	No	2	No	42	84
	TIM13, TIM14	16-bit	Up	Any integer between 1 and 65536	No	1	No	42	84
Basic	TIM6, TIM7	16-bit	Up	Any integer between 1 and 65536	Yes	0	No	42	84

Bảng 4.4 Các khối Timer được sử dụng trong đề tài

Timer	Chức năng sử dụng	Nhiệm vụ
TIM1	Bộ đếm 16-bit	Đếm xung của encoder con lắc
TIM2	Bộ đếm 16-bit	Đếm xung của encoder động cơ
TIM3	Bộ đếm 16-bit	Phát xung PWM để điều khiển động cơ

Bảng 4.5 Thông	số của các b	ộ USART/UART

USART name	Standard features	Modem (RTS/CTS)	LIN	SPI master	irDA	Smartcard (ISO 7816)	Max. baud rate in Mbit/s (oversampling by 16)	in Mbit/s	APB mapping
USART1	х	Х	х	х	x	Х	5.25	10.5	APB2 (max. 84 MHz)
USART2	х	Х	х	х	х	х	2.62	5.25	APB1 (max. 42 MHz)
USART3	х	Х	х	Х	х	Х	2.62	5.25	APB1 (max. 42 MHz)
UART4	х	-	х	-	х	-	2.62	5.25	APB1 (max. 42 MHz)
UART5	х	-	х	-	х	-	2.62	5.25	APB1 (max. 42 MHz)
USART6	х	Х	Х	Х	х	Х	5.25	10.5	APB2 (max. 84 MHz)

Tổng quan về board mạch điều khiển STM32F4 DISCOVERY

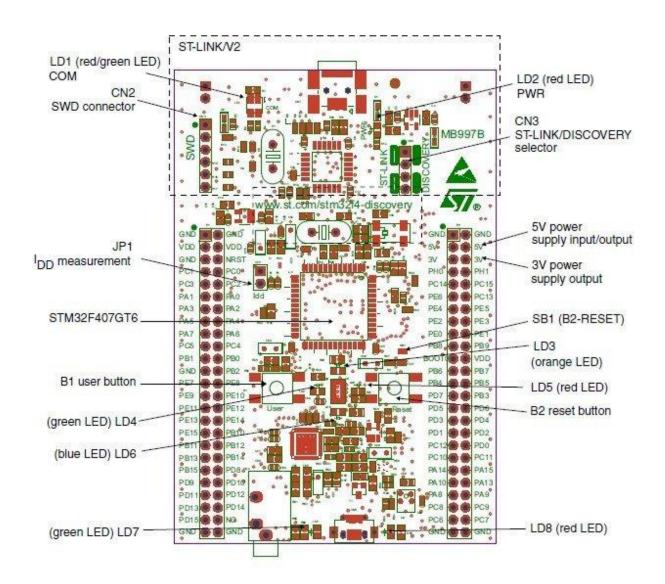
Đây là kit ứng dụng được phát triển dựa trên vi điều khiển ARM STM32F4 của hãng STMicroelectronics, đây là một kit hoàn toàn mới, đang rất phổ biến trong việc học tập và nghiên cứu hiện nay với ưu điểm là khả năng xử lý rất cao và cực kì nhanh, hỗ trợ rất tốt người dùng bởi các khối giao tiếp tích hợp sẵn trên kit. Bên cạnh đó, cộng đồng người nghiên cứu và ứng dụng kit này đang ngày càng tăng vì những ưu điểm và ưu thế của nó so với các kit phát triển khác. Nhận thấy được điều đó, nhóm đã quyết định chọn kit STM32F4 DISCOVERY làm vi điều khiển chính trong đề tài điều khiển con lắc ngược quay.



Hình 4.8 Mach STM32F4 DISCOVERY

Đối với kit STM32F4 DISCOVERY này, việc nạp code và debug hết sức đơn giản. Chúng ta chỉ việc cài driver cho máy tính và công cụ debug nhúng ST-link là có thể thể kết nối với kit thông qua cổng USB.

Tất cả các I/O trên mạch đều được đưa ra Header để dễ dàng kết nối với các thiết bị ngoại vi. Trên mạch có hai nút nhấn, nút màu xanh là nút người dùng, có thể làm ngoại vi trong các ứng dụng. Còn nút màu đen là nút RESET của mạch, dùng để khởi động lại các ứng dụng đang chạy trên kit khi cần thiết.



Hình 4.9 Các linh kiên trên bản mạch STM32F4 DISCOVERY của STMicroelectronics

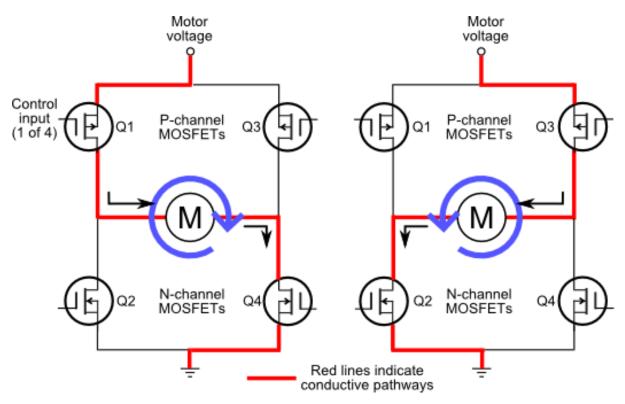
Úng dụng trong luận văn này, Nhóm sử dụng các chân điều khiển được liệt kê trong bảng sau:

Pin	Mô tả
Pin PE9	Kênh A encoder con lắc
Pin PE11	Kênh B encoder con lắc
Pin PB6	Chân Tx, gửi dữ liệu UART
Pin PB7	Chân Rx, nhận dữ liệu UART
Pin PA6	Chân PWM điều khiển động cơ
Pin PD12	Chân DIR để đảo chiều động cơ
Pin PB3	Kênh A encoder động cơ
Pin PA5	Kênh B encoder động cσ

Bảng 4.6 Các Pin của mạch điều khiển sử dụng trong đề luận văn

4.3.5 Mạch cầu H

Để điều khiển được động cơ, ta phải sử dụng một mạch driver. Mạch driver thông dụng nhất để điều khiển động là mạch cầu H. Mạch cầu H đơn giản bao gồm 4 con Transitor công suất làm khóa điện tử để dẫn dòng qua cuộn dây của động cơ (đối tượng), qua đó điều khiển được tốc độ và chiều quay của động cơ.



Hình 4.10 Sơ đồ mạch cầu H đơn giản

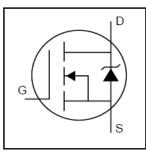
Vi điều khiển sẽ cấp tín hiệu tới cực điều khiển của các Transitor để thực hiện việc đóng/mở các cổng, từ đó thay đổi chiều quay của động cơ cũng như tốc độ động cơ thông qua độ rộng xung điều khiển (PWM). Một điều hết sức lưu ý và nếu ta tích cực 2 transitor cùng 1

bên trong cùng 1 khoảng thời gian, sẽ gây ra hiện tượng ngắn mạch hay còn gọi là hiện tượng trùng dẫn. Hiện tượng này tạo nên dòng rất lớn làm hư hỏng các thiết bị. Vì thế khi sử dụng cầu H, cần có biện pháp phòng chống trường hợp này.

Mạch cầu H điều khiển động cơ DC Servo được sử dụng trong ứng dụng này phải có khả năng đảo chiều nhanh và dẫn dòng lớn do đặc tính của mô hình, đồng thời các transistor phải hoạt động được ở tần số đóng ngắt cao nhằm giảm thiểu tối đa sự rung lắc của động cơ khi hoạt động. Vì thế, trong mạch cầu H, nhóm sử dụng IC MC33883 để làm driver cho MOSFET trong cầu H. Với nguyên lý hoạt động của mình,các IC driver được sử dụng trong mạch sẽ nhận tín hiệu PWM và sẽ đóng các MOSFET tương ứng để làm cho động cơ quay theo chiều xác đinh.

MOSFET sử dụng trong mạch cầu H này là transistor công suất IRF540n với các đặc tính rất nổi trội như: khả năng tải dòng liên tục lên đến 30A, tần số hoạt động khoảng 30kHz...đáp ứng rất tốt so với nhu cầu sử dụng của đề tài. Một số đặc tính cơ bản của MOSFET IRF540n được tóm tắt như sau:





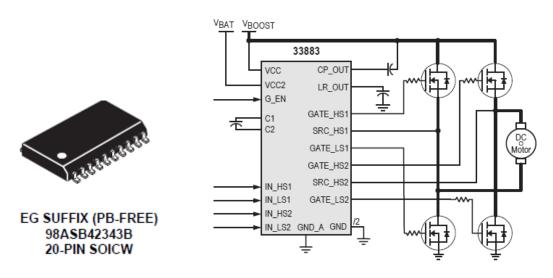
$$V_{DSS} = 100V$$
 $R_{DS(on)} = 44m\Omega$
 $I_D = 33A$

Hình 4.11 MOSFET IRF540n

Bảng 4.7 Các thông số cơ bản của MOSFET IRF540n

	Parameter	Max.	Units
I _D @ T _C = 25°C	Continuous Drain Current, V _{GS} @ 10V	33	
I _D @ T _C = 100°C	Continuous Drain Current, V _{GS} @ 10V	23	Α
I _{DM}	Pulsed Drain Current ①	110	
P _D @T _C = 25°C	Power Dissipation	130	W
	Linear Derating Factor	0.87	W/°C
V_{GS}	Gate-to-Source Voltage	± 20	V
I _{AR}	Avalanche Current①	16	Α
E _{AR}	Repetitive Avalanche Energy①	13	mJ
dv/dt	Peak Diode Recovery dv/dt ③	7.0	V/ns
T _J	Operating Junction and	-55 to + 175	
T _{STG}	Storage Temperature Range		°C
	Soldering Temperature, for 10 seconds	300 (1.6mm from case)	
	Mounting torque, 6-32 or M3 srew	10 lbf•in (1.1N•m)	

Còn về IC MC33883, thì đây là một IC lái cầu H chuyên dụng (được biết đến với tên gọi cầu H full-bridge) với các kênh lái cổng mức thấp và cao độc lập. Các kênh lái được điều khiển độc lập với nhau bởi bốn đầu vào riêng biệt, do đó cho phép thiết bị được tùy chọn cấu hình để điều khiển, gồm hai cổng lái mức cao và hai cổng lái mức thấp. Trong phạm vi đề tài này, nhóm chỉ sử dụng hai cổng lái mức cao để điều khiển vận tốc và chiều quay của động cơ, còn hai cổng lái mức thấp được nối xuống GND – không xử dụng.



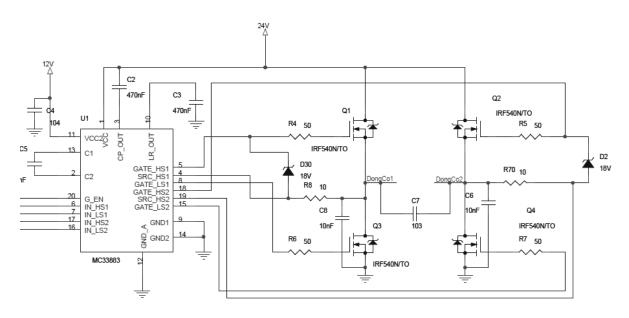
Hình 4.12 IC MC33883 và sơ đồ chân

Các thông số cơ bản của IC MC33883:

Điện áp vận hành VCC: 5.5V đến 55V

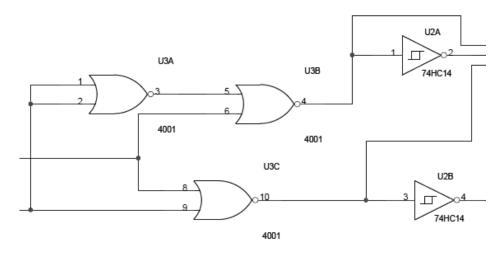
Điện áp VCC2: 5.5V đến 28V

Hỗ trợ tần số PWM lên đến 100kHz



Hình 4.13 Khối điều khiển cầu H và cầu H

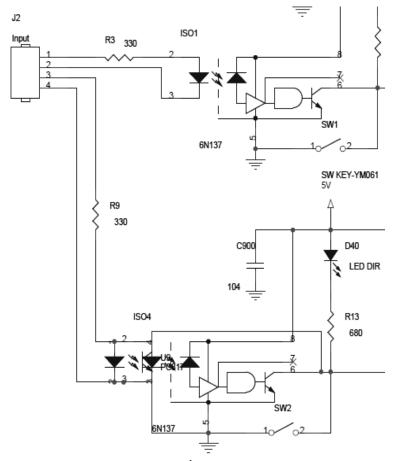
Hình trên là sơ đồ mạch cầu H được sử dụng trong đề tài này, gồm bốn MOSFET IRF540n và IC lái MC33883. Đồng thời để đảm bảo an toàn cho việc sử dụng, tránh hiện tượng trùng dẫn gây hư hại cho các board mạch, nhóm còn sử dụng thêm các cách ly và khối xử lý tín hiệu trước khi đưa vào IC lái cầu H.



Hình 4.14 Khối mạch chống trùng dẫn

Hình trên là khối xử lý tín hiệu trước khi đưa vào IC lái cầu H, bao gồm các cổng logic NAND và NOT Trigger nhằm tránh tình trạng trùng dẫn. Và hình bên là khối cách ly giữa mạch công suất và mạch điều khiển. Các tín hiệu điều khiển từ vi điều khiển được đưa vào các opto xung 6N137 rồi mới vào IC lái cầu H, do đó tránh được tình trạng hư hỏng mạch điều khiển và đảm bảo tính độc lập giữa các thành phần linh kiện.

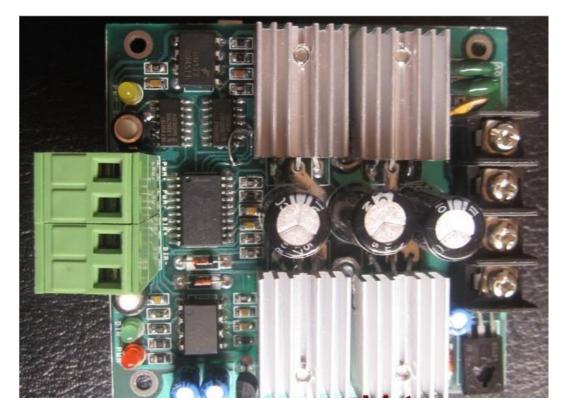
Và để tiết kiệm chi phí thi công mạch điện và gọn gàng hơn, những người thực hiện đề tài này đã tích hợp khối mạch lái



Hình 4.15 Khối mạch cách ly

và cầu H, khối xử lý tín hiệu đầu vào và khối cách ly trên cùng một board mạch. Như vậy việc

điều khiển sẽ hết sức đơn giản, chỉ việc cấp tín hiệu điều khiển cho cầu H từ vi điều khiển, cấp nguồn và đấu dây động cơ. Sơ đồ tổng thể mạch cầu H như sau:



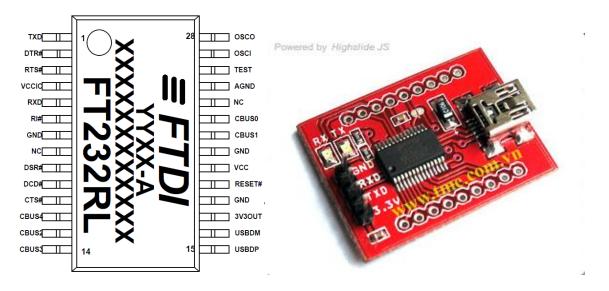
Hình 4.16 Cầu H sử dụng IC MC33883 tích hợp

4.3.6 Khối truyền thông nối tiếp

Trong đồ án này, nhóm sử dụng mạch truyền thông nối tiếp bất đồng bộ (UART) dùng IC FT232R của hãng FTDI's. Với các đặc tính sau:

- Chuyển đổi USB sang UART
- Tích hợp sẵn bộ giao động và EFPROM
- Tốc độ truyền từ 300 bauds đến 3Mbaud
- Bộ đệm nhận dữ liệu 128 bytes và bộ đệm truyền dữ liệu 256 bytes
- Hỗ trợ truyền thông 7 hoặc 8 bits, 1 hay 2 bits stop và parity

Nhóm sử dụng mạch UART của hãng TME, dùng IC FT232R để thực hiện truyền nhận dữ liệu nối tiếp giữa vi điều khiển và máy tính. Ta chỉ cần cấp nguồn 3.3V cho mạch UART này, kết nối chân TX của PIC với chân RX của UART, chân RX của vi điều khiển với chân TX của UART. Sau đó kết nối mạch UART với máy tính thông qua cổng USB là chúng ta có thể sử dụng được.



Nếu máy tính không tự động nhận mạch UART, ta phải tiến hành cài driver cho máy tính, việc này thực hiện rất đơn giản vì thông qua trang wed của FTDI's: http://www.ftdichip.com/Drivers/VCP.htm.

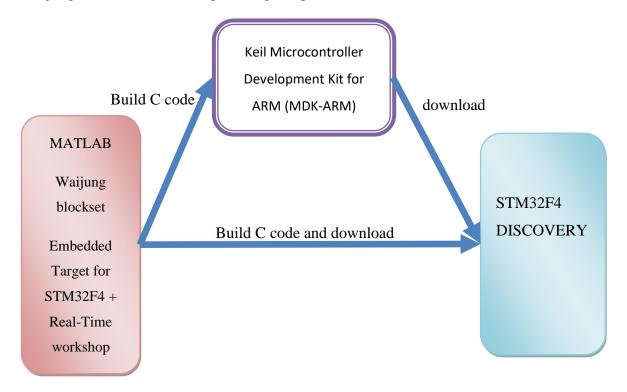
Sau khi máy tính đã nhận mạch UART, ta cần xác định xem máy tính cấp cho UART địa chỉ cổng COM nào. Việc này thực hiện dễ dàng thông qua chức năng Device Manager trong Window. Sau khi xác định được địa chỉ cổng COM, ta sẽ thực hiện dễ dàng việc truyền nhận dữ liệu giữa vi điều khiển và máy tính.

CHƯƠNG 5: THIẾT KẾ CHƯƠNG TRÌNH ĐIỀU KHIỂN

5.1 Thiết kế chương trình điều khiển thời gian thực

5.1.1 Giới thiệu Matlab simulink và thư viện Waijung

Matlab simulink là một chương trình mô phỏng hệ thống rất phổ biến trong việc học tập và nghiên cứu. Đây là một chương trình rất mạnh mẽ, hỗ trợ rất tốt người dùng trong việc thiết kế và đánh giá các hệ thống điều khiển. Với một thư viện hết sức phong phú, Matlab hỗ trợ hầu hết các nhu cầu cần thiết đối với việc thiết kế một hệ thống điều khiển tự động. Chính vì sự phổ biến và tính năng mạnh mẽ của Matlab, rất nhiều các công ty và cá nhân đã viết nhiều ứng dụng và thư viện nhằm nâng cao hơn nữa tính ứng dụng của nó. Và thư viện Waijung Blockset là một trong các ứng dụng đó.



Hình 5.1 Quan hệ giữa Matlab – thư viện Waijung và mạch điều khiển

Thư viện Waijung Blockset hô trợ cho STM32F4 được cung cấp bởi công ty AIMAGIN là một công cụ có thể được sử dụng dễ dàng và tự động tạo code C từ chính chương trình mô phỏng trong Matlab/simulink, do đó việc lập trình trở nên đơn giản hơn. Trước hết người dùng sẽ thiết kế mô hình mô phỏng trên chương trình Matlab Simulink, sau đó sử dụng các thư viện Waijung để tạo các khối điều khiển thích hợp. Khi đã hoàn thành việc mô phòng, người dùng chỉ việc sử dụng chức năng biên dịch có sẵn của Matlab, từ đó các

khối giải thuật sẽ được tự động biên dịch sang code C và có thể sử dụng cho bất kỳ chương trình lập trình vi điều khiển nào, hoặc có thể nạp trực tiếp lên vi điều khiển để chạy ứng dụng.

Thư viện Waijung Blockset hỗ trợ rất tốt các ứng dụng cơ bản trên vi điều khiển STM32F4, các ứng dụng này được đóng sẵn thành các khối và tích hợp trong chương trình Matlab Simulink khi chúng ta cài đặt thư viện Waijung. Việc sử dụng cũng hết sức đơn giản, chúng ta chỉ việc kéo thả các khối chức năng theo ý muốn, nối với các I/O cần thiết và thực hiện biên dịch là có thể download xuống vi điều khiển để chạy ứng dụng.

5.1.2 Giới thiệu các khối sử dụng trong đề tài

a. Khối Target setup

Waijung: 13.10a
Compiler: GNU ARM
MCU: STM32F407VG
Auto Compile Download: ON
Full Chip Erase: OFF
Auto run app: ON
Execution Profiler: None
Base Ts (sec): 0.01

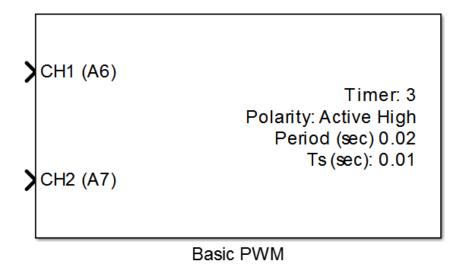
Target Setup

Hình 5.2 Khối Target setup trong Waijung

Đây là khối bắt buộc phải có trong mọi mô hình Matlab Simulink sử dụng vi điều khiển STM32F4. Khối này có các chức năng cấu hình cho vi điều khiển như sau:

- Tạo code C cho đối tượng vi điều khiển được chọn (MCU: STM32F407VG)
- Lựa chọn và điều khiển bộ biên dịch sang code C theo các chương trình khác nhau
 như GNU-ARM hoặc MDK-ARM...
- Thiết lập đồng bộ hệ thống cho từng đối tượng khác nhau
- Điều khiển toàn bộ quá trình từ tự động biên dịch cho đến download xuống vi điều khiển.

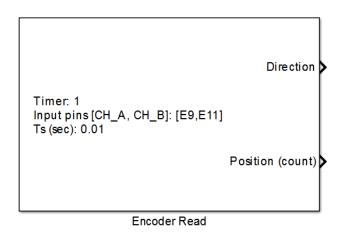
b. Khối PWM



Hình 5.3 Khối tạo xung PWM sử dụng Timer1

- PWM là viết tắt của Pulse Width Modulation điều chế biên độ rộng xung
- Dùng để điều khiển tốc độ động cơ
- Các chân CH1-CH4 là các chân cấp xung đầu ra trên mạch điều khiển

c. Khối đọc tín hiệu encoder

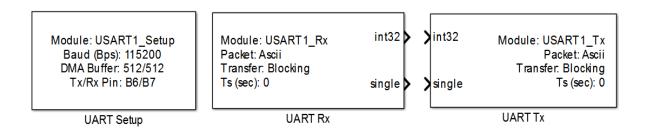


Hình 5.4 Khối đếm xung encoder sử dụng Timer1

- Dùng để đọc tín hiệu encoder đưa về
- Sử dụng các Timer để làm bộ đếm xung encoder
- Có 2 đầu vào là xung A và xung B của encoder

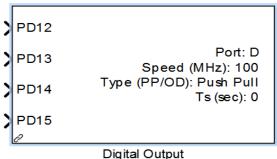
Khối UART d.

- Khối UART Setup phải được đặt trong mô hình simulink để kích hoạt việc sử dụng module UART khi ứng dụng cần gửi hoặc nhận dữ liệu từ thiết bị bên ngoài thông qua việc sử dụng giao thức UART.
- Khối UART Rx dùng để nhận dữ liệu UART khi ứng dụng cần giao tiếp giữa 2 thiết bị thông qua giao thức UART
- Khối UART Tx dùng để truyền dữ liêu đến UART khi ứng dưng cần giao tiếp giữa 2 thiết bị thông qua giao thức UART



Hình 5.5 Các khối của UART trong Waijung

Khối Digital Output e.



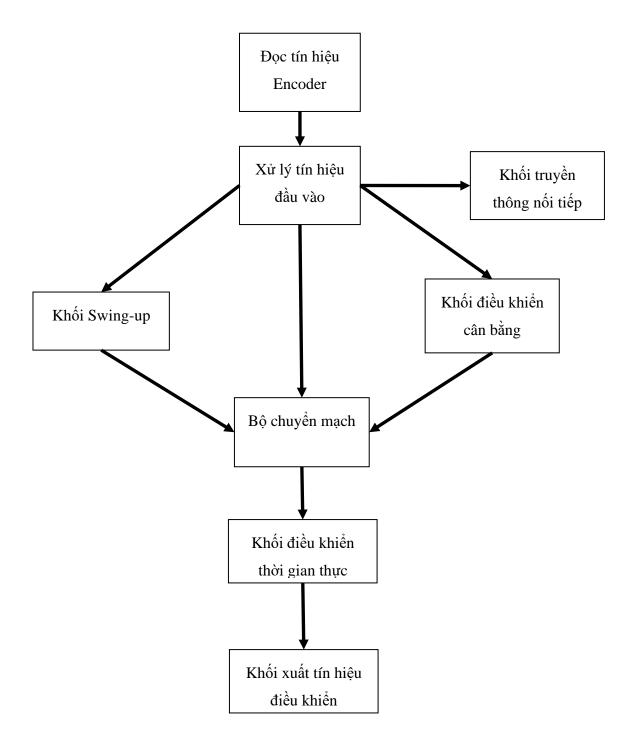
- Khối này dùng để tạo mức logic cho các chân output của vi điều khiển
- Sử dụng khối này để tạo tín hiệu đổi chiều động cơ cấp vào chân DIR của mạch cầu H

5.1.3 Xây dựng chương trình điều khiển

Để xây dưng chương trình điều khiển của hệ thống con lắc ngược quay, ta sử dụng các bộ điều khiển đã thiết kế ở chương 3 cùng với các khối chức năng trong thư viện Waijung đã trình bày ở trên trong Matlab.

Mô hình điều khiển thực hệ thống con lắc ngược quay được xây dựng trong Matlab Simulink được thể hiện chi tiết ở Hình 5.6 bao gồm khối Encoder để xác định 4 giá trị vị trí ,vận tốc của con lắc và cánh tay như đã miêu tả ở phần trên.Quan trọng ở đây là 3 khối điều

khiển bao gồm khối điều khiển Swing – up ,khối điều khiển Switch để chuyển từ điều khiển Swing sang điều khiển cân bằng và khối điều khiển cân bằng dùng thuật toán LQR .Chương trình được xây dựng dựa vào các kết quả mô phỏng thu được trong chương ba.



Hình 5.6 Sơ đồ thiết kế bộ điều khiển thời gian thực

Nhiệm vụ của các khối:

Đọc tín hiệu encoder: đọc về giá trị hiện tại của encoder con lắc và encoder cánh tay và chuyển dữ liệu thu được cho bộ xử lý tín hiệu vào.

Xử lý tín hiệu đầu vào: Từ dữ liệu đọc về, tính toán vị trí cánh tay và vị trí con lắc, chuyển về đơn vị Radian để việc điều khiển được dễ dàng và chính xác hơn. Khối này sẽ chuyển tín hiệu cho các khối: Swing-up, điều khiển cân bằng và khối chuyển mạch điều khiển.

Khối Swing-up: Đã được nhúng giải thuật Swing-up mô phỏng ở chương ba, nhận tín hiệu đầu vào đã được xử lý và tính toán tín hiệu điều khiển đầu ra.

Khối điều khiển cân bằng: là bộ điều khiển LQR, nhận tín hiệu đầu vào đã được xử lý, tính toán xuất tín hiệu đầu ra điều khiển con lắc cân bằng.

Bộ chuyển mạch: Vì nhóm sử dụng hai bộ điều khiển để điều khiển hệ thống và sử dụng chúng với mục đích khác nhau tùy trạng thái đầu vào của hệ, nên bộ chuyển mạch dùng để chọn bộ điều khiển thích hợp.

Khối điều khiển thời gian thực: nhiệm vụ của khối này là nhận tín hiệu điều khiển từ khối điều khiển Swing-up hoặc cân bằng, tính toán theo mô hình thực tế của hệ thống và xuất tín hiệu đầu ra cho các ngoại vi.

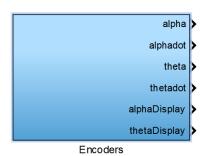
Khối xuất tín hiệu đầu ra: chọn các đầu ra của mạch điều khiển STM32F4 DISCOVERY để làm các Pin điều khiển.

Khối truyền thông nối tiếp: Nhận các tín hiệu đầu vào đã được xử lý để truyền lên máy tính qua giao diện người dùng để giám sát và đánh giá.

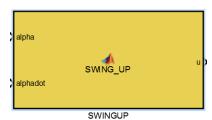
Dựa vào sơ đồ đã thiết kế ở trên, nhóm tiến hành xây dựng bộ điều khiển thời gian thực trên chương trình Simulink của Matlab với thư viện hỗ trợ của Waijung.

Trong mô hình này, nhóm sử dụng các khối chức năng cụ thể như sau:

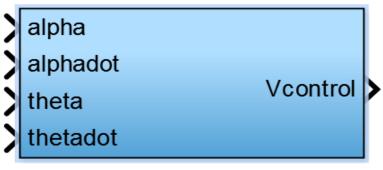
- Khối Encoder: bao gồm khối đọc tín hiệu và xử lý tín hiệu đầu vào. Ở đây nhóm đọc về số xung của hai encoder và tính toán ra bốn tín hiệu bao gồm: góc con lắc (alpha), vận tốc con lắc (alphadot), góc cánh tay (theta) và vận tóc góc cánh tay (thetadot).



- Khối Swing-up: nhúng giải thuật điều khiển Swingup, nhận tín hiệu đầu vào là góc và vận tốc con lắc (alpha và alphadot) để điều khiển. Bộ điều khiển này sẽ điều khiển sao cho alpha và alphadot dần về 0.

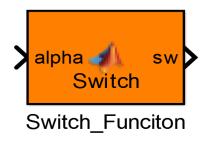


Khối LQR control: Khối điều khiển cân bằng đã gắn
 hệ số K, nhận bốn tín hiệu đã xử lý để tính toán và xuất tín hiệu điều khiển cân bằng con lắc.

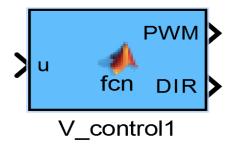


LQR Controller

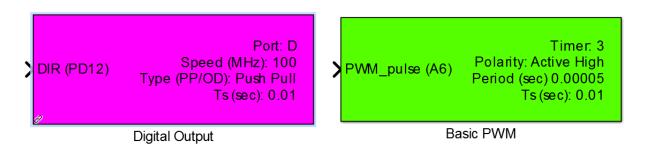
Khối Switch_Function và Switch là bộ chuyển mạch tín hiệu điều khiển từ hai khối Swing-up và LQR. Switch_Function sẽ nhận tín hiệu góc con lắc (alpha) đã được xử lý để tính toán tín hiệu chuyển mạch cho bộ chuyển Switch. Nếu alpha nằm trong ngoài khoảng (-10⁰,10⁰) thì sw = 1, bộ điều khiển được chọn để điều khiển là bộ Swing-up. Nếu alpha nằm trong khoảng (-10⁰,10⁰) thì bộ điều khiển cân bằng LQR sẽ được chọn để điều khiển.



Khối V_control là bộ điều khiển thời gian thực. Bộ này sẽ nhận tín hiệu điều khiển từ
hai bộ điều khiển và chuẩn hóa tín hiệu đầu ra cho phù hợp với mô hình thực tế.



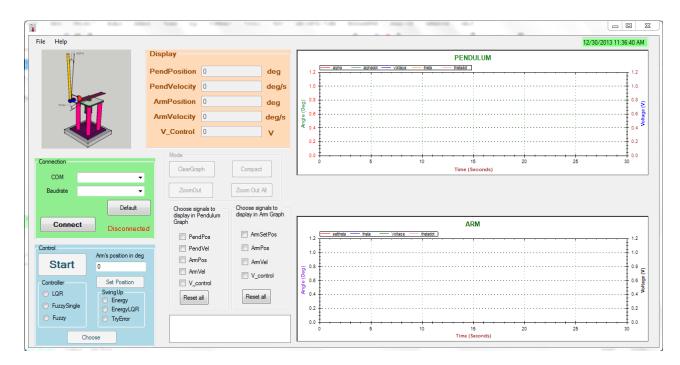
 Khối BasicPWM và Digital Output là hai khối chức năng của thư viện Waijung, có nhiệm vụ xuất xung PWM và tín hiệu DIR để điều khiển tốc độ và chiều quay của động cơ.



Mô hình điều khiển thực hệ thống con lắc ngược quay được trình bày trong PHỤ LỤC D của luận văn.

5.2 Thiết kệ giao diện người dùng

Giao diện người dùng được thiết kế dựa trên ngôn ngữ C# trên chương trình Visual Studio 2010 của hãng Microsoft. Máy tính sẽ nhận dữ liệu gửi về từ vi điều khiển qua cổng truyền thông nối tiếp và hiển thi lên giao diện để người điều khiển dễ dàng quan sát và đánh giá chất lượng điều khiển. Các phần và hướng dẫn sử dụng giao diện người dùng sẽ được trình bày trong PHỤ LỤC C của luận văn.



Hình 5.7 Giao diện điều khiển thiết kế trên máy tính

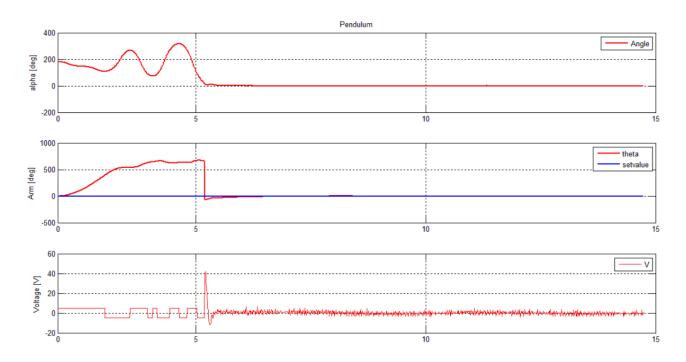
CHƯƠNG 6: KẾT QUẢ - KẾT LUẬN – HƯỚNG PHÁT TRIỀN CỦA LUẬN VĂN

6.1 Kết quả thực hiện

Với mô hình đã thiết kế ở trên, ta tiến hành biên dịch sang Code C nhờ công cụ hỗ trợ của Waijung và download xuống vi điều khiển. Sau đó, ta kết nối vi điều khiển với mô hình và máy tính sau đó chạy chương trình. Ta sẽ tiến hành điều khiển và đánh giá khi thay đổi tín hiệu đặt là vị trí cánh tay của mô hình.

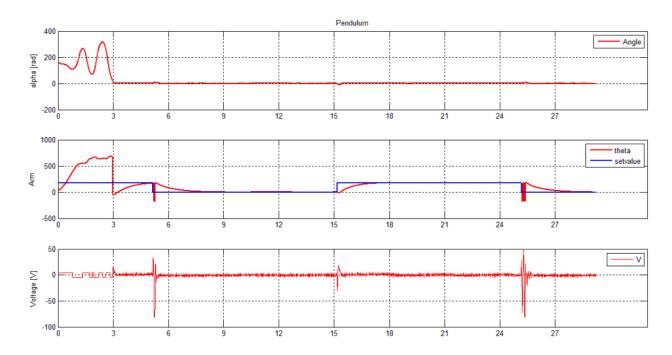
Nhóm tiến hành lần lượt 5 thí nghiệm để khảo sát sự cân bằng và ổn định của hệ thống với tín hiệu đặt là zero, xung vuông và xung hình sin.

Thí nghiệm 1: Tín hiệu đặt góc cánh tay là zero, sử dụng bộ điều khiển LQR và bộ Swing Up EnergyLQR ta được kết quả:

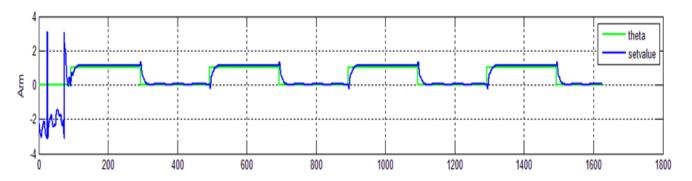


Hình 6.1 Các đáp ứng của mô hình điều khiển với tín hiệu đặt là zero Nhận xét: từ vị trí cân bằng dưới (180^0) , con lắc đã di chuyển lên vị trí cân bằng trên (0^0) trong khoảng thời gian là 5.2 giây. Với tín hiệu đặt là zero, góc cánh tay dao động rất nhỏ, có thể xem như là đứng yên. Đáp ứng của hệ thống rất tốt.

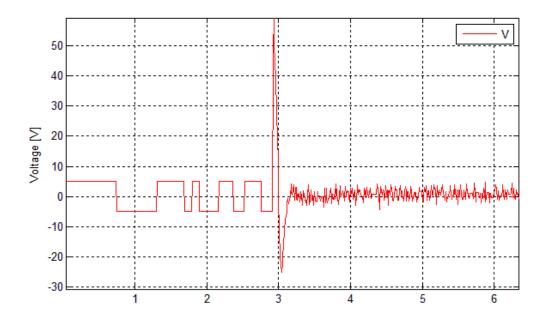
Thí nghiệm 2: Tín hiệu đặt góc cánh tay là xung vuông sử dụng bộ điều khiển LQR và bộ SwingUp EnergyLQR, ta được kết quả:



Hình 6.2 Các đáp ứng của mô hình điều khiển với tín hiệu đặt là xung vuông



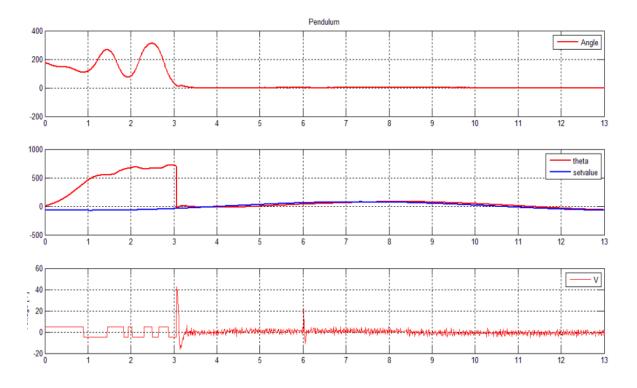
Hình 6.3 Đáp ứng của góc cánh tay với tín hiệu điều khiển



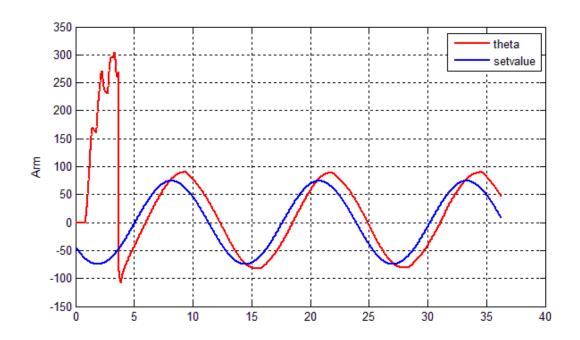
Hình 6.4 Điện áp điều khiển

Nhận xét: Với tín hiệu đặt là xung vuông biên độ 3 rad, chu kỳ 20 giây, con lắc giữ cân bằng khá tốt, tín hiệu góc cánh tay θ bám rất tốt theo tín hiệu đặt, hệ thống có rung lắc nhẹ.

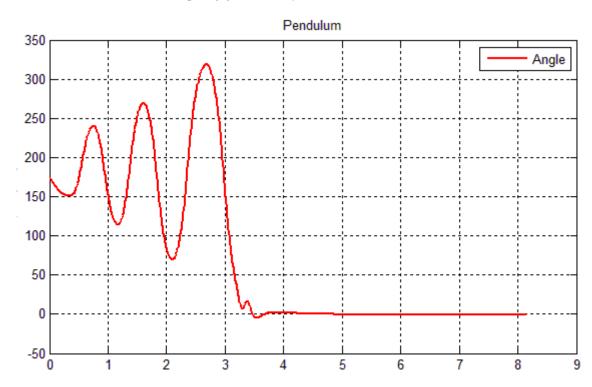
Thí nghiệm 3: Tín hiệu đặt góc cánh tay là xung hình sin sử dụng bộ điều khiển LQR và bộ SwingUp EnergyLQR, ta được kết quả:



Hình 6.5 Đáp ứng của hệ thống đối với tín hiệu đặt hình sin



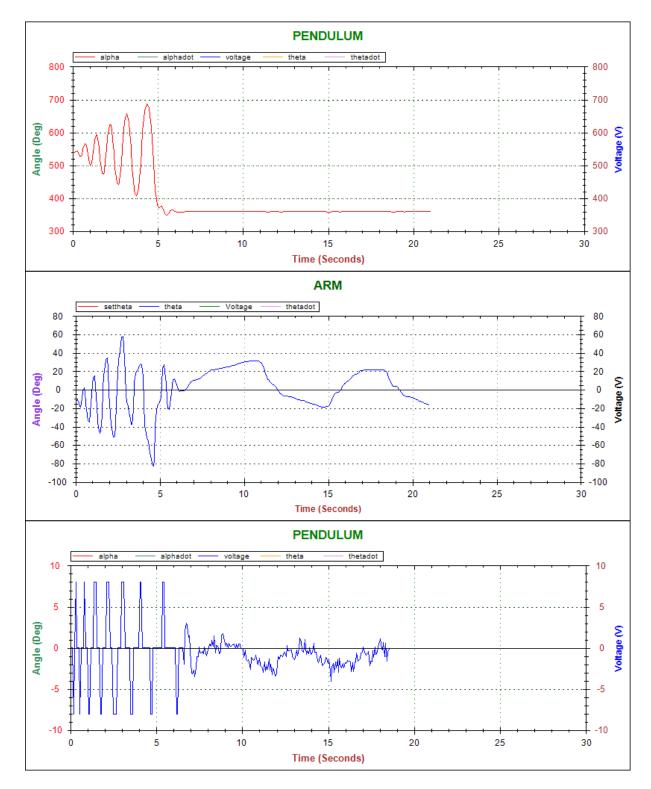
Hình 6.6 Đáp ứng góc cánh tay theta đối với tín hiệu đặt hình sin



Hình 6.7 Đáp ứng góc con lắc đối với tín hiệu xung hình sin

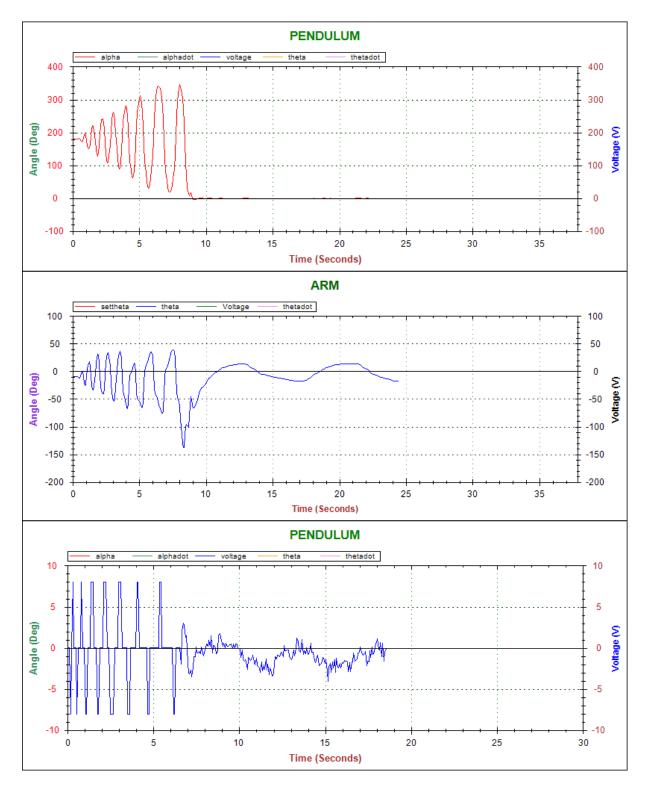
Nhận xét: Đối với tín hiệu xung hình sin, biên độ 75 độ, tần số ω =0.5 rad/s, góc cánh tay bám chưa tốt, sai số xác lập khoảng 10 độ; góc con lắc vẫn giữ cân bằng tốt.

Thí nghiệm 4: Điều khiển hệ thống sử dụng bộ điều khiển mờ bốn ngõ vào và bộ SwingUp TryError, ta được kết quả:



Hình 6.5 Đáp ứng của góc con lắc, góc cánh tay và điện áp điều khiển Nhận xét: Con lắc đạt vị trí cân bằng sau khoảng 4.9 giây, cánh tay cồn dao động lớn do ngõ vào chỉ có ba tập mờ nên bộ điều khiển mờ chưa tốt.

Thí nghiệm 5: Điều khiển hệ thống sử dụng bộ điều khiển mờ một ngõ vào và bộ SwingUp TryError. Ta được kết quả:



Hình 6.6 Đáp ứng của góc cánh tay, góc con lắc và điện áp điều khiển

Nhận xét: Con lắc đạt vị trí cân bằng sau 9 giây, góc cánh tay dao động ít hơn bộ điều khiển mờ 4 ngõ vào do số lượng tập mờ ngõ vào là 7.

Nhận xét chung: Con lắc luôn giữ cân bằng tốt với các tín hiệu đặt khác nhau và bộ điều khiển khác nhau, đáp ứng góc cánh tay tốt đối với xung zero và xung vuông. Đối với xung hình sin, góc cánh tay còn dao động với sai số tương đối lớn. Nhìn chung đáp ứng của hệ thống là tốt, đạt yêu cầu đề ra.

6.2 Kết luận

Với đề tài này, nhóm đã hoàn thành được những công việc sau:

- Nghiên cứu và thiết kế thành công mô hình con lắc ngược quay, xây dựng mô hình toán học và nắm vững các đặc tính động học của hệ con lắc ngược quay. Thi công mới mô hình hệ con lắc ngược.
- Nghiên cứu chip STM32F4, nghiên cứu lập trình C và Matlab cho kit STM32F4
 Discovery.
- Nghiên cứu các giải thuật của lý thuyết điều khiển hiện đại.
- Mô hình hóa và mô phỏng hệ con lắc ngược trên Matlab Simulink.
- Thiết kế và vẽ đồ thị thời gian thực mô hình con lắc ngược quay.
- Xuất tín hiệu và vẽ đồ thị lên máy tính để xem đáp ứng và tính ổn định của hệ thi công.
- Đã tiến hành thí nghiệm kiểm tra đáp ứng của hệ thống với các tín hiệu đặt khác nhau và nhìn chung, đáp ứng của hệ thống tốt, bám tốt theo tín hiệu đặt với dao động là rất nhỏ.

Điểm han chế:

- Mô hình còn vướng dây vì thế cánh tay di chuyển hạn chế.
- Con lắc còn rung khi ở vị trí cân bằng, tuy nhiên rung rất nhỏ.

6.3 Đóng góp của luận văn

- Luận văn đã đề xuất, đóng góp được một phương pháp điều khiển hệ con lắc ngược quay.
- Khái quát, ứng dụng một số lý thuyết điều khiển hiện đại vào điều khiển phi tuyến như
 hệ con lắc ngược quay.
- Đã xây dựng được phương pháp điều khiển để hệ con lắc ngược quay ổn định tại mọi vị trí của cánh tay.

6.4 Hướng phát triển của luận văn

Để con lắc ngược đi vào mức ổn định cao ta cần phát triển đồng thời cả phần cứng và phần mềm của mô hình con lắc ngược quay:

- Về phần cứng thiết kế thêm bộ chổi than để hệ thống khỏi bị vướng dây
- Về phần mềm nghiên cứu thêm nhiều giải thuật điều khiển như Fuzzy, Neural...
- Xây dựng bộ điều khiển tổng quát cho mô hình con lắc ngược n bậc (n<5).
- Thiết kế bộ đọc tín hiệu encoder không dây dùng RF, dùng xử lý ảnh để xử lý tín hiệu encoder.
- Hệ con lắc ngược là một hệ thống thiếu cơ cấu truyền động bậc hai nên có thể phát triển phương pháp điều khiển của luận văn này cho các hệ thống thiếu cơ cấu truyền động bậc cao hơn.

TÀI LIỆU THAM KHẢO

- [1] Slávka Jadlovská and Ján Sarnovský. "Modelling of classical and rotary pendulum systems –a genveralized approach," in Journal of ELECTRICAL ENGINEERING, VOL. 64, NO. 1, 2013, pp. 12-19.
- [2] Andrew K. Stimac. "Standup and Stabilization of the Inverted Pendulum," Massachusetts Institute of Technology, June 1999.
- [3] Marvin Bugeja. "Non-Linear Swing-Up and Stabilizing Control of an Inverted Pendulum System," EUROCON 2003 Ljubljana, Slovenia.
- [4] K. J. Astrom and K. Furuta. "Swing up a pendulum by energy control," presented at IFAC 13th World Congress, San Francisco, California, 1996.
- [5] Kevin M. Passino and Stephen Yurkovich. "Case Studies in Design and Implementation," in *Fuzzy Control*, Addison Wesley Longman, Inc. 1998.
- [6] Chunha Ryu, Byung-Jae Choi and Bong-Yeol Choi. "Design of a Fuzzy Logic Controller for a Rotary-type Inverted Pendulum System", International Journal of Fuzzy Logic and Intelligent Systems, vol 2, no.2 June 2002, pp 109-114.
- [7] Nguyễn Thị Phương Hà. Lý thuyết điều khiển hiện đại. Nhà xuất bản Đại học quốc gia TP Hồ Chí Minh, 2010.
- [8] Huỳnh Thái Hoàng. *Hệ thống điều khiển thông minh*. Nhà xuất bản Đại học quốc gia TP Hồ Chí Minh, 2006.
- [9] Vũ Duy Cường. *Cơ lý thuyết*. Nhà xuất bản Đại học quốc gia TP Hồ Chí Minh, 2011.
- [10] Lê Hồng Khiêm . Hệ thống điều khiển con lắc ngược quay , luận văn tốt nghiệp 2012.

Các website

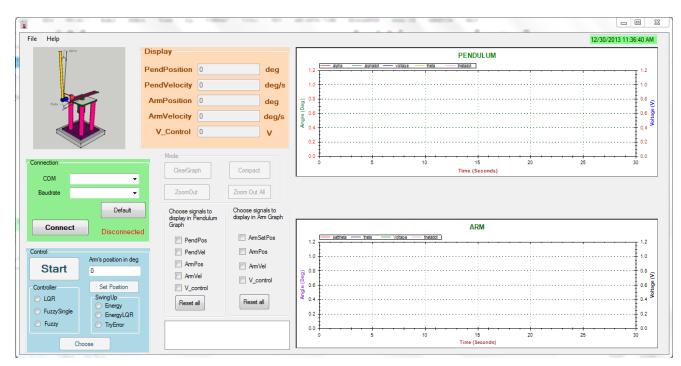
- [11] Waijung Blockset: http://waijung.aimagin.com/
- [12] Tài liệu ARM-Cortex M4 STM32F40x: http://www.st.com/
- [13] Matwork MATLAB: http://www.mathwork.com
- [14] Keil ARM: http://www.keil.com
- [15] Các trang web khác <u>www.wikipedia.org</u>, <u>www.dreamincode.net</u>, <u>www.codeproject.com</u>, www.social.msdn.microsoft.com, ...

PHỤ LỤC A: Bảng ký hiệu và thuật ngữ

Ký hiệu	Mô tả	Biến Matlab	Giá trị (đơn vị SI)
X ₀	Tín hiệu đầu vào		
u	Tín hiệu điều khiển		
X	Tín hiệu đầu ra		
$\varepsilon = \mathbf{x} - \mathbf{x}_0$	Tín hiệu sai lệch		
f	Tín hiệu nhiễu		
α	Vị trí con lắc		•••
ά	Vận tốc con lắc		•••
$\ddot{\alpha}$	Gia tốc con lắc		•••
B_{eq}	Hệ số ma sát nhớt tương đương	B_{eq}	0.0444
g	Gia tốc trọng trường	g	9.81
I_m	Dòng điện ở mạch phần ứng		
$J_{\scriptscriptstyle B}$	Momen quán tính của con lắc đối với trọng tâm.		•••
$J_{\it eq}$	Momen quán tính tương của cánh tay và con lắc quanh trục của θ	Jeq	0.004036
$oldsymbol{J}_m$	Momen quán tính của rotor động cơ	Jm	0.001
K_{m}	Hằng số sức điện động	Km	0.084026
K_{t}	Hằng số momen động cơ	Kt	0.084039
L	Nửa chiều dài con lắc	L	0.1254
m	Khối lượng con lắc	M	0.05562
r	Nửa chiều dài cánh tay	r	0.1095
$R_{_m}$	Điện trở phần ứng	Rm	2.6
T_m	Momen sinh bởi động cơ		•••
θ	Vị trí cánh tay		•••
$\dot{ heta}$	Vận tốc cánh tay		•••
$\ddot{ heta}$	Gia tốc cánh tay		•••
$V_{\scriptscriptstyle m}$	Điện áp ngõ vào mạch phần ứng		•••
$\dot{x}_{\scriptscriptstyle B}$	Vận tốc trọng tâm của con lắc theo phương x		
$\dot{\mathcal{Y}}_{B}$	Vận tốc trọng tâm của con lắc theo phương y		
$\eta_{_m}$	Hiệu suất động cơ	eta	0.69

PHỤ LỤC B: Chương trình Matlab mô hình con lắc ngược

```
%% - General constants for Rotary Inverted Pendulum ------
g = 9.81; % [m/s^2]
L = 0.1254; % [m]
                 % [m]
% [Kg]
% [m]
m = 0.05562;
r = 0.1095;
T = 0.1093,
Jeq = 0.004036; % [Kg-meter^2]
Beq = 0.0444; % [Nm/(rad/s)]
n = 0.69; % hieu suat
n = 0.05;
Kt = 0.084039; % [N-m/Amp]
Km = 0.084026; % [Volt-sec/rad]
Rm = 2.6; % dien tro phan ung (Ohm)
Rm = 2.6; % alen tro p...... % momen quan tinh dong co
%% ***** constants of the model -----
a = Jeq + m*r*r+Jm;
b = m*L*r;
c = (4/3) *m*L*L;
d = m*g*L;
e = Beq + (n*Kt*Km)/Rm;
f = n*Kt/Rm;
%% **** the linearized Rotary Inverted Pendulum system around the straight
up position LQR
A=[0,
                                                       0;
   a*d/(a*c-b*b),
                      Ο,
                                     0, -b*e/(a*c-b*b);
                       Ο,
                                     Ο,
   b*d/(a*c-b*b),
                      Ο,
                                     0, -c*e/(a*c-b*b)]
B = [
                0;
  b*f/(a*c-b*b);
    c*f/(a*c-b*b)]
Q = diag ([50 1 80 20]);
%Q = [5 0 0]
                          0;
           20
    0
                   0
                           0;
     0
           0
                  0
                           0;
     0
           0
                  0
                          0];
R = 1;
%P=dare(A,B,Q,R)
K = lqr(A, B, Q, R)
```



PHỤ LỤC C: Giao diện điều khiển

1) Connection

- COM: Chọn cổng giao tiếp trên máy tính
- Baudrate: Chọn tốc độ Baud cho cổng giao tiếp
- Default: chọn cổng COM đầu tiên khi kết nối với tốc độ Baud mặc định là 256000
- Connect: kết nối với mạch giao tiếp qua cổng COM đã chọn, trạng thái kết nối sẽ hiện thi ở bên canh "Connect" button.

2) Control

- Controller: chọn bộ điều khiển để điều khiển cân bằng con lắc ngược, bao gồm ba bộ điều khiển: LQR, điều khiển mờ một ngõ vào và điều khiển mờ bốn ngõ vào.
- SwingUp: chọn bộ điều khiển Swing Up con lắc ngược, bao gồm ba bộ:
 - Energy: bộ điều khiển tối thiểu thời gian Swing Up sử dụng phương pháp năng lượng.
 - Energy: bộ điều khiển tối thiểu thời gian Swing Up sử dụng phương pháp năng lượng cho bộ điều khiển cân bằng LQR.
 - O TryError: Bộ điều khiển Swing Up bằng phương pháp thử sai.
 - O Choose: chọn các bộ điều khiển được đánh dấu ở trên.
- Arm's Position in deg: khung nhập vị trí để điều khiển cánh tay
- Set Position: điều khiển cánh tay theo vị trí nhập ở khung "Arm's Position in deg"

3) Display

Phần Display gồm bốn phần:

- Khung Display:
 - o PendPosition: hiển thị vị trí hiện tại của góc con lắc theo độ (deg).
 - o PendVelocity: hiển thị vận tốc góc hiện tại của con lắc (deg/s).
 - o ArmPosition: hiển thị vị trí hiện tại của góc cánh tay theo độ (deg).
 - O ArmVelocity: hiển thị vận tốc góc hiện tại của cánh tay (deg/s).
 - O V Control: điện áp điều khiển hiện tại (Volt).
- Khung Mode: tác động lên đồ thị hiển thị
 - o ClearGraph: Xóa toàn bộ khung đồ thị đã hiển thị
 - o Compact: Chế độ hiển thị Full Plot
 - o ZoomOut: Thu nhỏ đồ thị hiển thị.
 - o ZoomOutOn: Trở về trạng thái hiển thị ban đầu.
 - o Các khung CheckBox: chọn tín hiệu muốn hiển thị lên đồ thị
- Khung đồ thị:
 - o Pendulum: đồ thị hiển thị các tín hiệu liên quan đến con lắc
 - Arm: đồ thị hiển thị các tín hiệu liên quan đến cánh tay, bao gồm cả tiến hiệu đặt.
- 4) Thanh Menu chức năng:
 - File:
 - Video: Load video
 - Save Graph: lưu đồ thị hiển thị của cánh tay và con lắc
 - Exit: Thoát chương trình
 - Help: thông tin về chương trình

PHỤ LỤC D: Mô hình điều khiển thực hệ thống con lắc ngược quay

