

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HCM

KHOA CƠ KHÍ
BỘ MÔN CƠ ĐIỆN TỬ



ĐỒ ÁN TỐT NGHIỆP (CHUYÊN NGÀNH)

CHẾ TẠO VÀ ĐIỀU KHIỂN THIẾT BỊ HAPTIC CẢM NHẬN LỰC TƯỞNG TÁC VỚI VẬT THỂ ẢO

Giảng viên hướng dẫn: TS. NGUYỄN VIỄN QUỐC

Sinh viên thực hiện: NGUYỄN ANH QUỐC

Lớp: DHC DT11C

ĐOÀN TRƯỜNG XUÂN

Lớp: DHC DT11C

Thông tin liên lạc: ĐT: 0346363634

Email: dtruongxuan.cdt@gmail.com

TP. Hồ Chí Minh, tháng 12 năm 2019

Tp.HCM, ngày tháng năm 2019

NHIỆM VỤ ĐỒ ÁN CHUYÊN NGÀNH

Họ tên SV 1: ĐOÀN TRƯỜNG XUÂN MSSV:15088761 Lớp: DHCDT11C

Họ tên SV 2: NGUYỄN ANH QUỐC MSSV:15065461 Lớp: DHCDT11C

Tên đề tài: Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể áo

Nhiệm vụ (yêu cầu về nội dung và số liệu ban đầu):

Tìm hiểu haptic.

- Thiết kế và chế tạo thiết bị haptic dựa trên nguyên lý lực điện từ.
- Thiết kế và thực hiện bộ điều khiển trên nền vi điều khiển.
- Viết chương trình điều khiển cho vi điều khiển.
- Viết phần mềm giao diện trên máy tính.

Ngày giao nhiệm vụ ĐACN: 17/08/2019

Ngày kiểm tra giữa kỳ (50%): 25/10/2019

5.Ngày hoàn thành: 26/12/2019

6. Giáo viên hướng dẫn: TS. Nguyễn Viễn Quốc Phản hướng dẫn:.....

Ngày tháng năm 2019

CHỦ NHIỆM BỘ MÔN

GV HƯỚNG DẪN

TRƯỞNG KHOA

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Viễn Quốc là người đã tận tình chỉ dẫn và hỗ trợ chúng em trong suốt thời gian thực hiện đề tài.

Đồng thời chúng em cũng xin cảm ơn các giảng viên trường đại học công nghiệp thành phố Hồ Chí Minh đã tạo điều kiện cho chúng em nghiên cứu và học tập trong quãng thời gian vừa qua.

Trong quá trình thực hiện và hoàn thành luận văn này, do hạn chế về mặt kinh nghiệm cũng như thời gian, nên việc thiếu sót là không thể tránh khỏi. Chúng em mong quý thầy cô có thể vui lòng bỏ qua cũng như đóng góp xây dựng để đề tài được hoàn thiện hơn. Và đồng thời giúp chúng em được tích lũy thêm kinh nghiệm để hoàn thiện bản thân.

NHẬN XÉT GIÁO VIÊN HƯỚNG DẪN

MỤC LỤC

CHƯƠNG 1	TỔNG QUAN	1
1.	Giới thiệu hệ thống haptic	1
1.1	Đôi nét về cương nghệ phản hồi lực	1
2.	Các cơ cấu chấp hành haptic hiện nay	2
2.1	Bộ tạo rung	2
2.2	Bộ phản hồi lực	3
2.3	Vòng xoáy khí	3
2.4	Bộ tạo siêu âm	3
3.	Ứng dụng của haptic trong các lĩnh vực hiện nay	4
CHƯƠNG 2	CÁC ĐẶC TRƯNG THIẾT KẾ TRONG CƠ CẤU HAPTIC	5
1.	Tác động của tần số của cơ cấu cơ khí tới cảm nhận trong hệ thống haptic	5
1.1	Vùng hoạt động của cơ cấu tactile (chạm)	5
2.	Các dạng cảm nhận cầm nắm với cơ cấu haptic trong mô phỏng phản hồi lực và vị trí (kinaesthetic haptic)	7
2.1	Phân loại bộ phận tiếp xúc	7
CHƯƠNG 3	THIẾT KẾ VÀ CHẾ TẠO CƠ KHÍ	12
1.	Bản vẽ thiết kế	12
1.1	Thiết kế các bộ phận gia công	14
1.1.1	Thanh đỡ khung máy 1	14
1.1.2	Khung máy	15
1.1.3	Bàn máy	16
1.1.4	Óp chặn trực	17
1.1.5	Khớp đặt ngón cái	17
1.1.6	Bánh dẫn động cơ	18
1.1.7	Thanh đỡ khung máy 2	19
1.1.8	Bánh dẫn encoder	19
1.1.9	Khớp điều chỉnh vị trí encoder	20
1.1.10	Bánh dẫn động	21
1.1.11	Khớp đặt ngón trỏ	21
CHƯƠNG 4	THIẾT KẾ HỆ THỐNG GIÁM SÁT, ĐO LƯỜNG VÀ ĐIỀU KHIỂN MÔ HÌNH HAPTICS 1 BẬC TỰ DO TRÊN MÁY TÍNH	23

1.	Sơ đồ hệ thống	23
2.	Danh sách thiết bị	23
2.1	MẠCH CHUYỂN ĐỔI UART- USB	25
2.2	KIT PHÁT TRIỂN VI ĐIỀU KHIỂN STM32F103C8T6 BLUE PILL ..	25
2.3	CẢM BIẾN	25
2.4	Nguồn và mạch công suất	32
2.5	Động cơ	33
CHƯƠNG 5	THIẾT KẾ GIAO DIỆN PHẦN MỀM VÀ ĐỊNH NGHĨA KHUNG TRUYỀN TÍN HIỆU GIAO TIẾP VỚI VI ĐIỀU KHIỂN	35
1.	Thiết kế giao diện phần mềm	35
2.	Định nghĩa khung truyền tin giữa giao diện máy tính và vi điều khiển	35
CHƯƠNG 6	THIẾT KẾ VÀ LẮP RÁP MẠCH ĐIỀU KHIỂN	37
1.	Sơ đồ mạch nguyên lý tổng quát	37
2.	Tính toán thiết kế mạch đọc loadcell hai chiều cho bộ adc của vi điều khiển stm32f103c8t6:.....	38
CHƯƠNG 7	ĐO LƯỜNG VÀ GIA CÔNG CÁC GIÁ TRỊ CẢM BIẾN	40
1.	Đo lường lực tác động, dòng tiêu thụ động cơ bằng bộ chuyển đổi ADC của vi điều khiển STM32F103C8T6	40
1.1	Bộ lọc FIR (Finite Impulse Response).....	40
1.2	Bộ lọc nhiễu Kalman.....	44
1.3	So sánh bộ lọc FIR với bộ lọc Kalman	52
2.	Thí nghiệm calib Loadcell.....	52
CHƯƠNG 8	KHẢO SÁT ĐẶC TÍNH TẢI ĐỘNG CƠ	55
1.	Thí nghiệm: khảo sát hệ hở điều khiển động cơ.....	55
2.	Nhận xét:.....	56
3.	Kết luận:	57
4.	Thuật toán Linear Regression	57
4.1	Giới thiệu đôi nét về thuật toán Linear Regression	57
4.2	LR một biến.....	57
CHƯƠNG 9	THÍ NGHIỆM TÍNH THỜI GIAN THỰC THI CỦA CHƯƠNG TRÌNH	63
1.	Mục đích thí nghiệm:.....	63
2.	Bố trí thí nghiệm.....	63
3.	Tiến hành thí nghiệm:.....	63

CHƯƠNG 10 XÂY DỰNG CHƯƠNG TRÌNH ĐIỀU KHIỂN ĐỘNG CƠ	67
1. Hàm truyền của cơ cấu chấp hành haptic 1 bậc tự do	67
2. Phương pháp điều khiển động cơ	69
3. Tìm hiểu giải thuật điều khiển PID	69
4. Áp dụng bộ điều khiển PID vào điều khiển động cơ sinh lực trong thiết bị haptic:.....	72
4.1 Mô hình điều khiển:	72
4.2 Bộ điều khiển PID số dạng 1 (P-ID song song lý tưởng) ^[9] :	72
4.3 Bộ điều khiển PID số dạng 2 (PID song song) ^[9] :	73
4.4 Chọn thời gian lấy mẫu T và phân cấp ưu tiên thực thi trong chương trình C cho bộ điều khiển PID	75
4.5 Thí nghiệm khảo sát tác động bộ điều khiển PID để điều khiển động cơ sinh lực	75
CHƯƠNG 11 THÍ NGHIỆM KIỂM CHỨNG LỰC MÔ PHỎNG VÀ MÔ PHỎNG VỚI VẬT THỂ ĐÀN HỒI ẢO	82
1. Thí nghiệm tìm phương trình đặc tính của vật mô phỏng (lò xo)	82
2. Thí nghiệm mô phỏng lò xo	84
CHƯƠNG 12 KẾT LUẬN VÀ KIẾN NGHỊ	86
TÀI LIỆU THAM KHẢO.....	87
PHỤ LỤC	88
Phần (Code C) chương trình vi điều khiển	88
Phần (Code C#) giao diện máy tính:	102
Phần (Code MATLAB) quy hồi tuyến tính (RL).....	109
Thông số, datasheet thiết bị.....	112

DANH SÁCH HÌNH ẢNH, LUU ĐỒ

Hình 1.1: Động cơ ERM	2
Hình 1.2: Động cơ LRA	2
Hình 1.3: Bộ phản hồi lực của Microsoft.....	3
Hình 1.4: Mô hình bộ xoáy khí	3
Hình 1.5: Bộ tạo siêu áp.....	3
Hình 1.6: Cơ cấu haptic trong y học	4
Hình 1.7: Cơ cấu haptic trong đào tạo hàng không	4
Hình 2.1 Biểu đồ tần số cảm nhận con người trong cơ cấu haptic	5
Hình 2.2: Mô hình hệ tổng quan hệ thống haptic.....	6
Hình 2.3: Mô hình thí nghiệm phương pháp Haptic chạm	7
Hình 2.4: Kết quả thí nghiệm haptic chạm	7
Hình 2.5: Các cơ cấu cầm nắm tay	8
Hình 2.6: Thí nghiệm cầm viết góc 30 độ.....	8
Hình 2.7: Thí nghiệm cầm viết góc 90 độ.....	9
Hình 2.8: Thí nghiệm cầm viết góc 0 độ.....	9
Hình 2.9: Thí nghiệm nắm chặt khối trụ tròn	10
Hình 2.10: Thí nghiệm nắm chặt khối cầu.....	10
Hình 2.11: Thí nghiệm nắm chặt với vòng sắt.....	10
Hình 3.1: Bản vẽ thiết kế 3d	12
Hình 3.2: Bản thiết kế 3D	12
Hình 3.3: Bản vẽ chi tiết	13
Hình 3.4: Thanh đỡ khung máy 1	14
Hình 3.5: Khung máy	15
Hình 3.6: Bàn máy	16
Hình 3.7: Ốp chặn trực	17
Hình 3.8: Khớp đặt ngón cái	17
Hình 3.9: Bánh dẫn động cơ	18
Hình 3.10: Thanh đỡ khung máy 2	19
Hình 3.11: Bánh dẫn encoder.....	19
Hình 3.12: Khớp điều khiển vị trí	20
Hình 3.13: Bánh dẫn động	21
Hình 3.14: Khớp đặt ngón trỏ	21
Hình 4.1: USB UART	25
Hình 4.2: Mạch điều khiển trong thực tế	25
Hình 4.3: Sơ đồ chân mạch điều khiển	25
Hình 4.4: Bản vẽ kích thước encoder.....	26
Hình 4.5: Encoder thực tế	26
Hình 4.6: Kích thước loadcell	27
Hình 4.7: Loadcell thực tế.....	27
Hình 4.8: Sơ đồ nguyên lý loadcell.....	28
Hình 4.9: Sơ đồ chân mạch INA 128	29

Hình 4.10: Sơ đồ chân LM 358.....	31
Hình 4.11: Mạch cầu h trong thực tế.....	33
Hình 4.12: Động cơ trong thực tế	33
Hình 5.1: Giao diện sau khi chạy	35
Hình 5.2: Giao diện khi thiết kế	35
Hình 8.1: Biểu đồ động cơ quay thuận nghịch 1a.....	55
Hình 8.2:Đồ thị động cơ quay thuận nghịch 1c	56
Hình 8.3: Đồ thị động cơ quay thuận nghịch thí nghiệm 1b.....	56
Hình 8.4: Đồ thị linear động cơ quay nghịch.....	59
Hình 8.5: Đồ thị linear của động cơ quay thuận nghịch	58
Hình 8.6: Đồ thị linear động cơ quay thuận.....	59
Hình 8.7: Đồ thị so ba đường linear.....	60
Hình 8.8: Đường linear sau khi bỏ giá trị phi tuyến khi động cơ quay nghịch.....	60
Hình 8.9: Đồ thị linear động cơ quay thuận sau khi lọc	61
Hình 8.10: So sánh đường linear động cơ quay thuận và nghịch sau khi lọc	62
Hình 9.1: Lưu đồ giải thuật tính toán thời gian chương trình.....	63
Hình 9.2: Thời gian chạy không điều khiển động cơ.....	64
Hình 9.3: Thời gian chạy ở chế độ điều khiển vòng hở.....	64
Hình 9.4: Thời gian điều khiển khi sử dụng thuật toán PID	65
Hình 9.5: Thời gian điều khiển động cơ khít thay đổi hệ số PID	65
Hình 10.1: Mô hình truyền lực tay với vật đòn hồi.....	67
Hình 10.2: Mô hình truyền lực động cơ.....	68
Hình 10.3: Mô hình điều khiển Haptic	72
Hình 10.4: Mô hình điều khiển PID song song.....	72
Hình 10.5: Mô hình PID song song	74
Hình 10.6: Cấp độ ưu tiên thực hiện một chu kỳ lệnh.....	75
Hình 10.7: Mô hình thí nghiệm điều khiển lực sử dụng thuật toán PID	76
Hình 10.8 Đáp ứng khâu P bộ điều khiển pid số dạng 1 Kp=1000	76
Hình 10.9 Đáp ứng khâu P bộ điều khiển pid số dạng 1 Kp=2000	77
Hình 10.10 Đáp ứng khâu P bộ điều khiển pid số dạng 1 Kp = 18000	77
Hình 10.11 Đáp ứng kết hợp PI bộ điều khiển pid số dạng 1	78
Hình 10.12 Đáp ứng kết hợp PI bộ điều khiển pid số dạng 1	78
Hình 10.13 đáp ứng bộ PID dạng 1	79
Hình 10.14 đáp ứng bộ PID dạng 1	79
Hình 10.15 Đáp ứng khâu P, bộ PID dạng 2, Kp= 2000	80
Hình 10.16 Đáp ứng khâu P, bộ PID dạng 2, Kp= 2500	80
Hình 10.17 đáp ứng PD, bộ PID dạng 2	81
Hình 10.18 Đáp ứng PD, Bộ PID dạng 2.....	81

DANH SÁCH BẢNG

Bảng 3.1: Thông kê chi tiết.....	13
Bảng 3.2: Thông số thanh đỡ khung máy	14
Bảng 3.3: Thông số khung máy.....	15
Bảng 3.4: Thông số bàn máy	16
Bảng 3.5: Thông kê ốp chặn trực.....	17
Bảng 3.6: Khớp đặt ngón cái.....	18
Bảng 3.7: Bánh dẫn động cơ	18
Bảng 3.8: Bảng thông số Thanh đỡ khung máy 2	19
Bảng 3.9: Bảng thông số bánh dẫn encoder.....	20
Bảng 3.10: Thông số khớp điều chỉnh vị trí encoder	20
Bảng 3.11: Thông số bánh dẫn động	21
Bảng 3.12: Thông số khớp đặt ngón trỏ.....	21
Bảng 4.1: Thông kê chức năng thiết bị.....	23
Bảng 4.2: Datasheet Encoder	25
Bảng 4.3: Datasheet loadcell	28
Bảng 4.4: Datasheet INA128	30
Bảng 4.5: Chọn thông số R khuếch đại INA128.....	30
Bảng 4.6: Datasheet LM358	31
Bảng 4.7: Datasheet nguồn	32
Bảng 4.8: Datasheet mạch cầu h	33
Bảng 4.9: Datasheet động cơ	34
Bảng 5.1: Khung truyền dữ liệu UART từ máy tính đến vi điều khiển.....	36
Bảng 5.2: Khung truyền dữ liệu UART từ vi điều khiển đến máy tính.....	36
Bảng 7.1: Giá trị calib loadcell.....	53
Bảng 9.1: Thời gian các bước thực hiện chương trình điều khiển	66

DANH SÁCH THUẬT NGỮ VÀ CÁC TỪ VIẾT TẮT

Haptic:	Thiết bị mô phỏng tương tác giữa người và máy
Haptic feedback:	Phản hồi mô phỏng
Telerobotics:	robot điều khiển từ xa
Servomechanism:	Cơ cấu phản hồi sai số lệch góc trong hàng không
ERM:	Bộ truyền động lệch tâm
Taptic Engine:	Bộ phận mô phỏng chạm trên các thiết bị Apple
LRA:	Bộ truyền động cộng hưởng tuyến tính
Tactile:	Chạm, Xúc giác
Active haptics interaction:	Các thiết bị mô phỏng tương tác chủ động
Transition haptics interaction:	Các thiết bị mô phỏng tương tác trong giai đoạn chuyển tiếp
Passive haptics interaction:	Các thiết bị mô phỏng tương tác trong bị động
Kinaesthetic sensing:	Cảm nhận di chuyển, vị trí
Transmission delay:	Độ trễ đáp ứng của hệ thống haptic
Bandwidth:	Độ rộng của tín hiệu đáp ứng haptic
Resolution:	Độ phân giải của cơ cấu haptic
Dynamic range:	Phạm vi động năng của hệ haptic
Signal:	Tín hiệu
Noise:	Tín hiệu nhiễu
Contact grasps:	Tương tác chạm
Precision grasps:	Tương tác cầm nắm thông thường
Power grasps:	Tương tác cầm nắm chặt
UART:	Chuẩn giao tiếp nối tiếp dữ liệu máy tính
Driver:	Mạch công suất động cơ
PID:	Thuật toán điều khiển vi, tích phân, tỷ lệ
Duty:	Độ rộng xung
Lọc Kalman:	Bộ lọc nhiễu Kalman
FIR:	Phương pháp lọc nhiễu trung bình

ADC:	Bộ chuyển đổi tín hiệu tương tự sang số
Gaussian trung bình:	Bộ lọc Gaussian trung bình
Hiệp phương sai:	Độ sai lệch giữa hai trạng thái
PWM:	Phương pháp điều khiển động cơ bằng cách điều khiển độ rộng xung
Linear Regression:	Thuật toán hồi quy tuyến tính
Machine Learning:	Máy học
Supervised Learning:	Máy học có kiểm soát
Prescaller:	Bộ chia tỷ lệ trong điều khiển tần số mạch
Period:	Chu kỳ
Steady state error:	Sai số trạng thái tĩnh
Overshot :	Độ vọt lố

LỜI MỞ ĐẦU

Bối cảnh hiện nay

Thiết bị phản hồi xúc giác (Haptic devive) được sử dụng trong các mô hình, cơ cấu mô phỏng lại tương tác qua lại giữa người dùng và thiết bị chấp hành. Từ “Haptic” trong kỹ thuật được Oxford định rằng là sự quan hệ đến cảm giác của xúc giác, đặc biệt liên quan đến nhận thức và thao tác của các đối tượng (ở đây là người sử dụng) bằng cách sử dụng các giác quan cảm nhận và thực hiện quyền điều khiển. Trong tiếng Hy Lạp “Haptic” cũng có nghĩa là “Chạm”[1].

Với sự phát triển của công nghệ ngày nay các thiết bị ứng dụng công nghệ ngày càng trở nên phổ biến, nhằm mang lại cảm nhận tương tác trực quan hơn cho người sử dụng. Một trong những thiết bị phổ biến nhất hiện nay ứng dụng công nghệ này là điện thoại thông minh với ứng dụng công nghệ “3D touch”.

Với sự phát triển vượt bậc từ các công nghệ truyền thông và vi mạch thì độ trễ trong việc truyền tải dữ liệu ngày càng được xóa bỏ. Vì thế các ứng dụng haptic trong đời sống theo đó ra đời ngày càng nhiều. Đặc biệt công nghệ này còn được ứng dụng rất nhiều trong các lĩnh vực y học, thám hiểm, quân sự,... Đây là công nghệ cho phép chúng ta tiếp cận tương tác trực quan hơn với các môi trường khắc nghiệt, độc hại, nguy hiểm, hay các công việc có tính chất đặc biệt trong y học.

Mục đích của nghiên cứu

Đề tài nghiên cứu thiết kế, chế tạo một thiết bị có khả năng thu thập, mô phỏng lại các lực tương tác giữa các vật mẫu với tác động của người sử dụng thông qua mô trường thực tế ảo dựa trên nguyên lý lực điện từ. Với mục tiêu là cơ sở nền tảng cho những hệ thống vận hành từ xa có phản hồi lực, trong đó người sử dụng có thể cảm nhận được sự tương tác với môi trường; nhờ đó có thể cảm nhận và thao tác một cách chính xác và khéo léo hơn. Kết quả của đề tài này là cơ sở cho việc chế tạo hệ robot master-slave vận hành tương tác từ xa có phản hồi lực.

Phương pháp nghiên cứu của đề tài

Nghiên cứu lý thuyết của hệ phản hồi lực và mối tương quan giữa biến dạng của vật và lực, thiết kế một cơ cấu thể hiện mô phỏng hóa các mô hình mẫu đã được nghiên cứu từ trước.

Sử dụng mô hình thuật toán PID để điều khiển động cơ, trợ lực chuyển động của ngón tay vì sự tác động không mong muốn của cơ cấu cơ khí và mô phỏng lại lực tác động của vật trong môi trường ảo với ngón tay.

Ý nghĩa khoa học và thực tiễn của đề tài

Luận văn mang tính chất nghiên cứu, ứng dụng các lý thuyết nghiên cứu và ứng dụng khoa học. Từ đó mở ra các ứng dụng mang tính chất thực tiễn hơn vào công nghiệp cũng như trong đời sống. Đồng thời có thể là tài liệu tham khảo có các đối tượng quan tâm và nghiên cứu đến haptic.

TÓM TẮT ĐỀ TÀI

Đề tài nghiên cứu thiết kế, chế tạo một thiết bị có khả năng thu thập, mô phỏng lại các lực tương tác giữa các vật mẫu với tác động của người sử dụng thông qua mô trường thực tế ảo dựa trên nguyên lý lực điện từ. Với mục tiêu là cơ sở nền tảng cho những hệ thống vận hành từ xa có phản hồi lực, trong đó người sử dụng có thể cảm nhận được sự tương tác với môi trường; nhờ đó có thể cảm nhận và thao tác một cách chính xác và khéo léo hơn. Kết quả của đề tài này là cơ sở cho việc chế tạo hệ robot master-slave vận hành tương tác từ xa có phản hồi lực.

CHƯƠNG 1 TỔNG QUAN

1. Giới thiệu hệ thống haptic

1.1 Đôi nét về công nghệ phản hồi lực

1.1.1 Định nghĩa

Theo định nghĩa của viện công nghệ Massachusetts (Massachusetts Institute of Technology) Haptic là thiết bị cải thiện khả năng nhận dạng và nhận biết các vật thể thông qua xúc giác. Đây là cảm nhận chủ động, liên quan đến việc trải nghiệm một vật bằng tay chứ không phải cảm giác thụ động của một rung động hoặc lực trên da.^[1]

Haptic cũng được định nghĩa thiết bị phản hồi xúc giác (Haptic feedback) như sau. Công nghệ Haptic, còn được gọi là giao tiếp vị trí và chuyển động hoặc cảm ứng 3D, để cập đến bất kỳ công nghệ nào có thể tạo ra trải nghiệm chạm bằng cách áp dụng lực, rung hoặc chuyển động đến người dùng. Những công nghệ này có thể được sử dụng để tạo các đối tượng ảo trong mô phỏng máy tính, để điều khiển các đối tượng ảo và để tăng cường khả năng điều khiển từ xa các máy móc và thiết bị (telerobotics). Các thiết bị Haptic có thể kết hợp các cảm biến xúc giác đo các lực do người dùng tác động lên bề mặt vật thể. Từ haptic, từ tiếng Hy Lạp: ἀπτικός (haptikos), có nghĩa là "liên quan đến cảm giác chạm". Các thiết bị haptic đơn giản là phổ biến ở dạng bộ điều khiển trò chơi, cần điều khiển và vô lăng^[2].

Công nghệ Haptic tạo điều kiện cho việc điều tra về cách thức cảm giác của con người hoạt động bằng cách cho phép tạo ra các vật thể ảo haptic có kiểm soát. Hầu hết các nhà nghiên cứu phân biệt ba hệ thống cảm giác liên quan đến cảm giác chạm ở người: da, vị trí, chuyển động thông qua các khớp, cơ bắp và haptic. Tất cả các nhận thức qua trung gian bởi sự nhạy cảm của da và vị trí, chuyển động qua các cơ và khớp được gọi là nhận thức về xúc giác. Cảm giác chạm có thể được phân loại là thụ động và chủ động, và thuật ngữ "haptic" thường được liên kết với cảm ứng chủ động để giao tiếp hoặc nhận dạng đối tượng.

1.1.2 Đôi nét về lịch sử phát triển của công nghệ haptic

Chúng ta có thể kể đến những thiết bị ứng dụng haptic như sau. Một trong những ứng dụng đầu tiên của công nghệ haptic là trên các máy bay lớn sử dụng hệ thống servomechanism trên các máy bay lớn. Bằng sáng chế đầu tiên của Hoa Kỳ cho điện thoại ứng dụng haptic đầu tiên được cấp vào năm 1973. Năm 2015, Apple Watch ứng

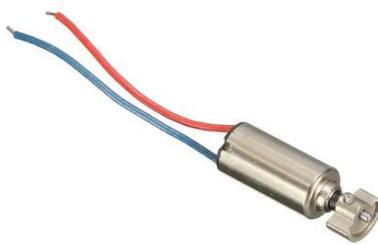
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo
dụng cảm biến chạm để đưa ra thông báo và cảnh báo từ điện thoại di động của người
sử dụng.^[3]

2. Các cơ cấu chấp hành haptic hiện nay

2.1 Bộ tạo rung

Ở loại cơ cấu chấp hành này ta có thể chia làm hai loại cơ cấu chấp hành nhỏ như
sau:

Bộ truyền động lệch tâm ERM (eccentric rotating mass), được sử dụng nhiều hiện
nay với các thiết bị chúng ta có thể thấy trong đời sống hàng ngày như: tay điều khiển
trên các loại máy chơi game, “Taptic Engine” trên các loại thiết bị thông minh của
Apple.



Hình 1.1: Động cơ ERM

Bộ truyền động cộng hưởng tuyến tính LRA (linear resonant actuator), với cơ chế
hoạt động gần giống như cơ chế hoạt động của loa. LRA có ưu điểm hơn so với ERM
ở khoảng thời gian phản hồi nhanh. Do đó có thể truyền đến người dùng cảm nhận
chính xác hơn.

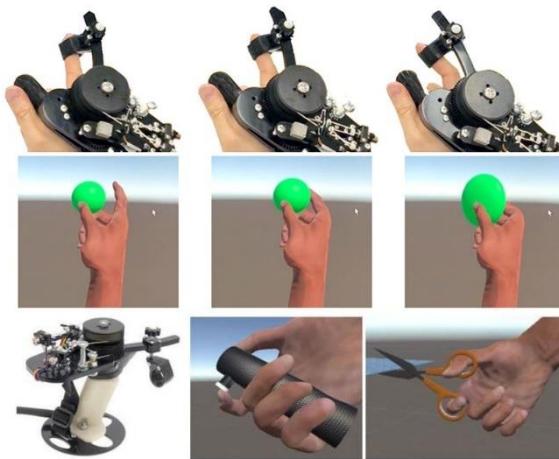


Hình 1.2: Động cơ LRA

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

2.2 Bộ phản hồi lực

Sử dụng động cơ để mô phỏng tạo ra các chuyển động tương tác của vật trong môi

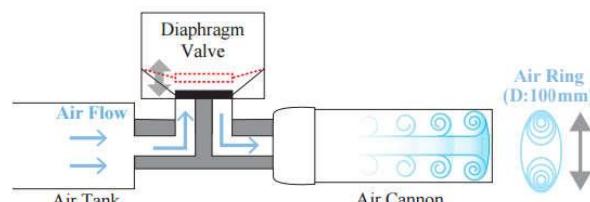


Hình 1.3: Bộ phản hồi lực của Microsoft

trường ảo lên đối tượng điều khiển. Đây cũng là cơ cấu trong phạm vi nghiên cứu của luận văn này.

2.3 Vòng xoáy khí

Cơ cấu này đang được nghiên cứu và phát triển bởi Microsoft và Disney. Với nguyên lý hoạt động sử dụng những luồng khí có hình dạng như chiếc bánh donut giúp người dùng cảm nhận được vật thể.

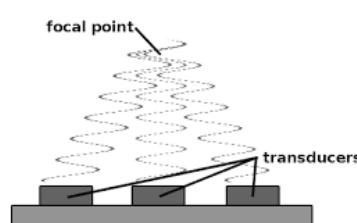


Generation of an air vortex ring

Hình 1.4: Mô hình bộ xoáy khí

2.4 Bộ tạo siêu âm

Nguyên lý hoạt động dựa trên sóng siêu âm để tạo ra cảm nhận cho người dùng.



Hình 1.5: Bộ tạo siêu âm

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

3. Ứng dụng của haptic trong các lĩnh vực hiện nay

Hệ thống Haptic đã được nghiên cứu và ứng dụng cho nhiều lĩnh vực khác nhau trong đời sống. Dưới đây là một số minh chứng về ứng dụng của haptic trong thực tế.

Trong y học



Hình 1.6: Cơ cấu haptic trong y học

Trong hàng không



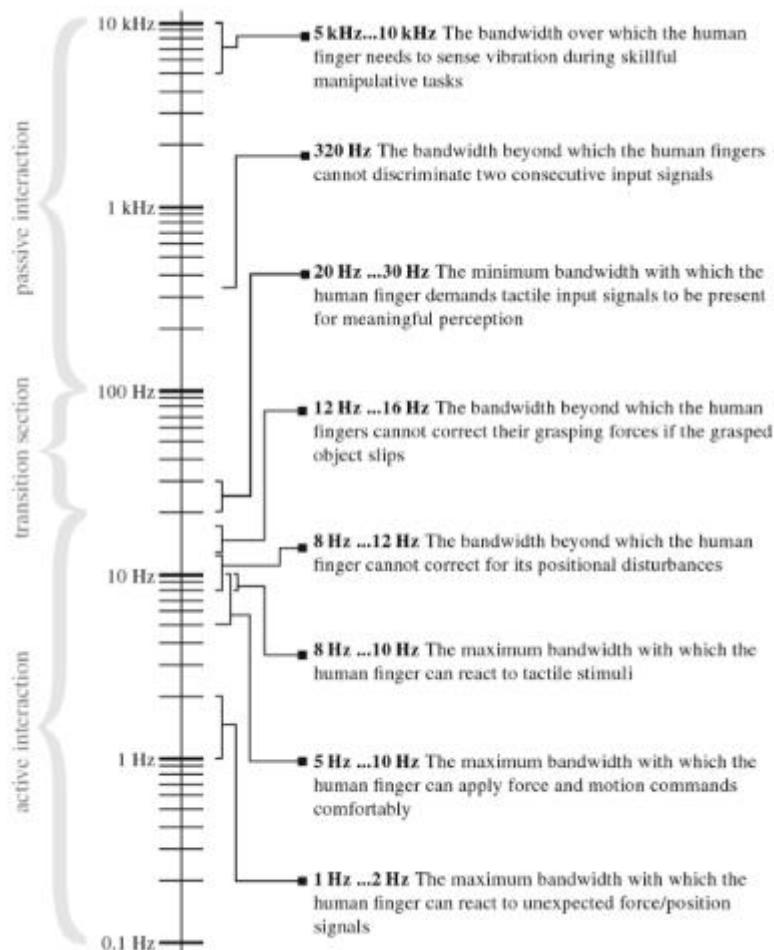
Hình 1.7: Cơ cấu haptic trong đào tạo hàng không

CHƯƠNG 2 CÁC ĐẶC TRƯNG THIẾT KẾ TRONG CƠ CẤU HAPTIC

1. Tác động của tần số của cơ cấu cơ khí tối cảm nhận trong hệ thống haptic

1.1 Vùng hoạt động của cơ cấu tactile (chạm)

Tần số cảm nhận chạm của các cơ cấu haptic có thể dao động lên đến 10kHz. Tần số cao nhất cảm nhận được nằm trong khoảng 30 đến 320Hz.

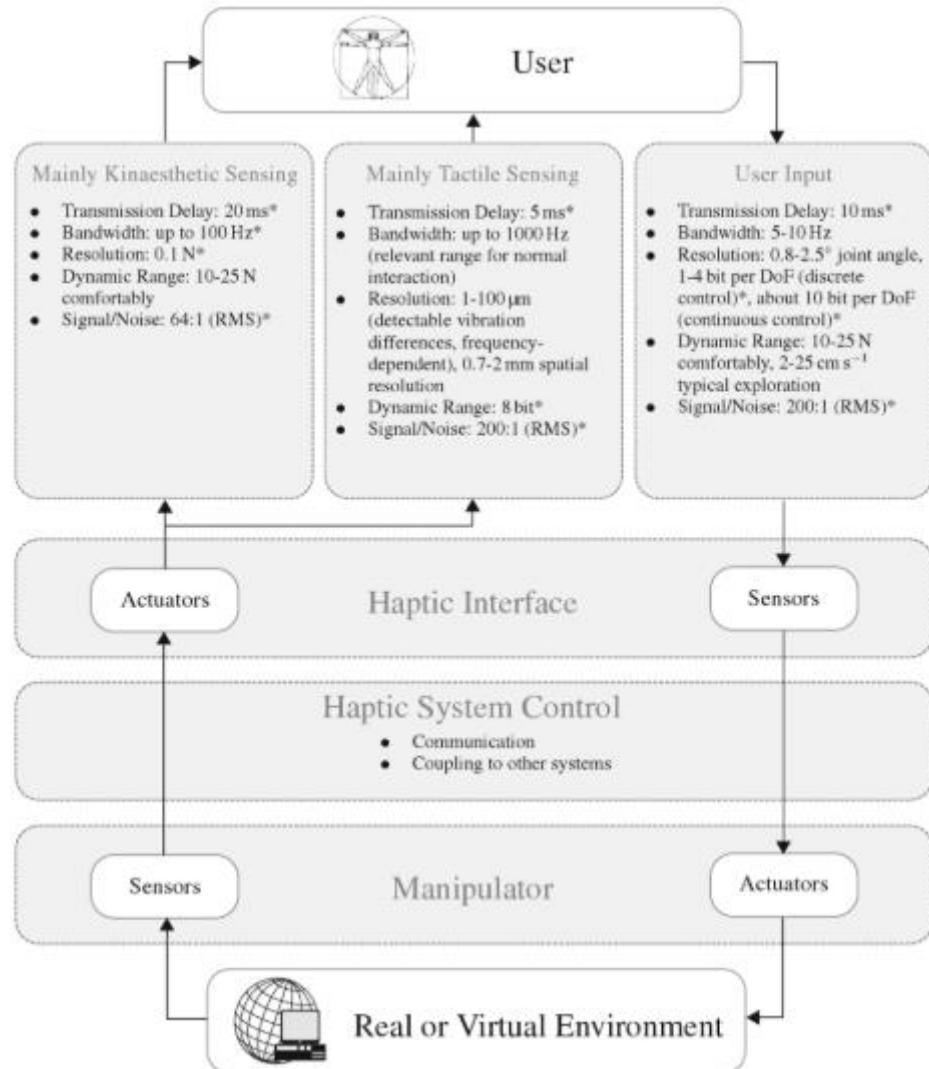


Hình 2.1 Biểu đồ tần số cảm nhận con người trong cơ cấu haptic

Khoảng tiếp nhận tần số của thiết bị cảm nhận chủ động, nó được định ra dựa trên khả năng cảm nhận và điều khiển của người sử dụng. Khả năng nhận biết rung động cao nhất nằm ở ngón tay con người và tần số lớn nhất có thể đạt được lên đến 10kHz. Khoảng chủ yếu thuộc đo khả năng cảm nhận của con người là vùng bị động (passive). Tuy nhiên, người sử dụng không thể tiếp nhận hay tương tác nằm trong khoảng bị động (passive), thực tế là khi con người cố gắng thực thi với tốc độ nhanh như là bấm vào cây viết thì khoảng tần số tối đa cũng chỉ nằm trong khoảng dao động ở mức thấp tức là mức chủ động (active haptics interaction). [3]

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

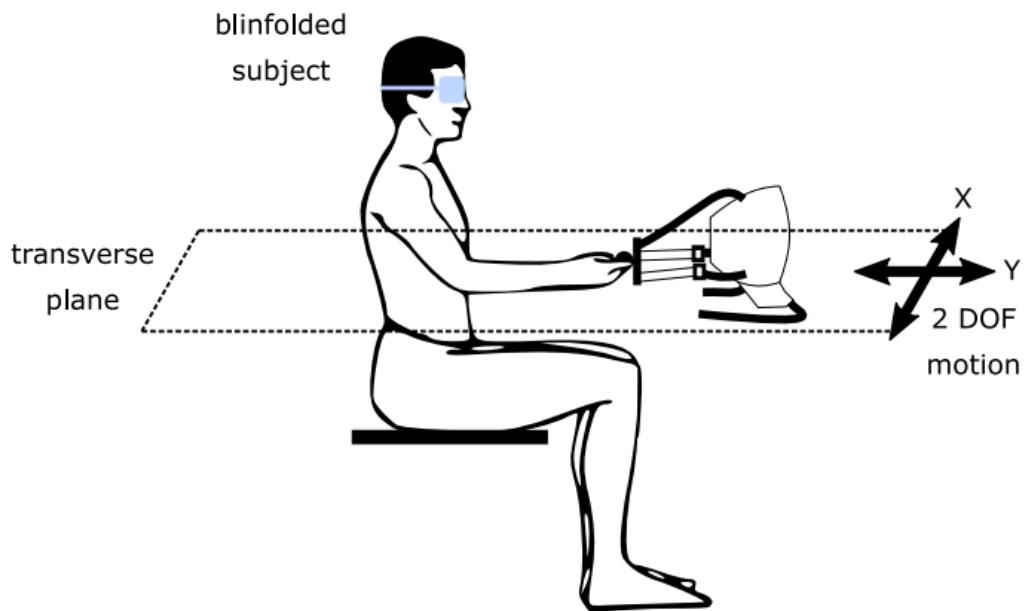
Dựa vào các khoảng tần số này ta có thể phân chia các dạng hồi tiếp và đáp ứng của một đối tượng sử dụng cơ cấu haptic thành các nhóm như biểu đồ sau:



Hình 2.2: Mô hình hệ thống haptic

Theo nghiên cứu về mức độ cảm nhận trong hệ thống cơ cấu haptic. Trong các thí nghiệm về nhận dạng hình học thông qua xúc giác ở tay cho kết quả như sau:

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 2.3: Mô hình thí nghiệm phương pháp Haptic chạm

Và kết quả đạt được như sau^[4]:

Active haptics			Passive haptics				
	answered (%)				answered (%)		
presented	□	△	○	presented	□	△	○
□	90	0	10	presented	20	0	80
△	0	100	0	△	20	60	20
○	10	0	90	○	20	20	60

Hình 2.4: Kết quả thí nghiệm haptic chạm

2. Các dạng cảm nhận cầm nắm với cơ cấu haptic trong mô phỏng phản hồi lực và vị trí (kinaesthetic haptic)

2.1 Phân loại bộ phận tiếp xúc

Cầm nắm tiếp xúc (Contact-grasps): biểu diễn sự tác động chạm lên đối tượng bằng cả bàn tay hoặc một phần của nó. Cơ cấu nút nhấn thông thường là cơ cấu cháp hành của loại tương tác này. Hướng tiếp xúc thường chỉ theo một chiều duy nhất.

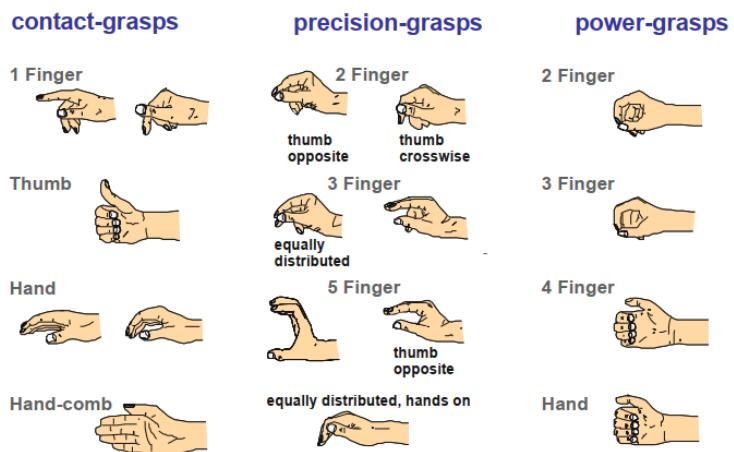
Cầm nắm chính xác (Precision-grasps): Biểu diễn quá trình cầm nắm phải sử dụng hai hay nhiều ngón tay cố định chiều chuyển động của vật. Phương pháp này thông

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

thường sẽ có định ít nhất một chiều chuyển động tự do. Chúng ta thường sử dụng cơ cấu chấp hành này khi vặn chìa khóa.

Cầm nắm chặt (Power-grasps): Biểu diễn sự tác động vào vật chỉ tối thiểu hai ngón tay. Các ngón tay này đóng vai trò cố định tất cả các bậc chuyển động của vật. Kiểu tiếp xúc này chúng ta thường dùng khi cầm nắm một vật nào đó.

Trong phạm vi đề tài này chúng ta sẽ nghiên cứu sâu hơn về cơ cấu cầm nắm chính xác và cầm nắm chặt bằng hai ngón tay. Chúng ta có hình minh họa cho các cơ cấu như sau:

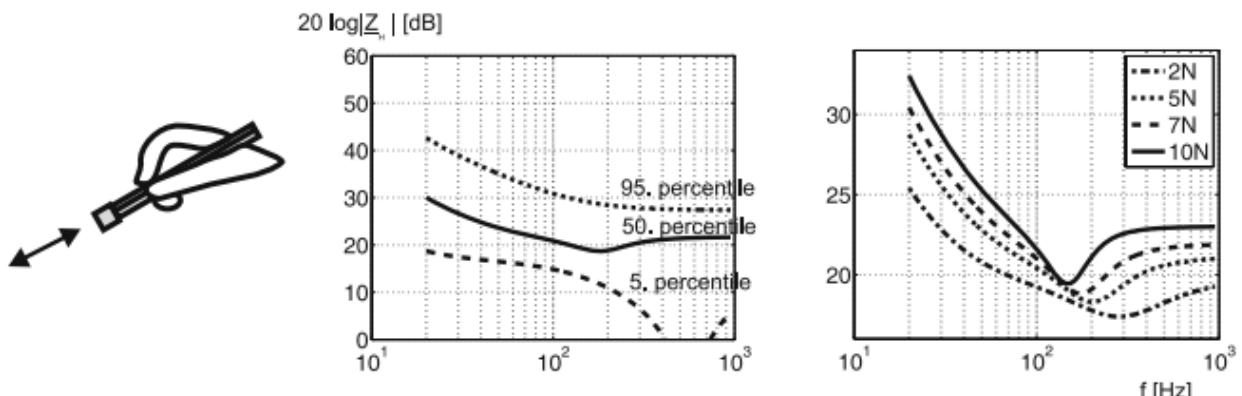


Hình 2.5: Các cơ cấu cầm nắm tay

2.1.1 Bài toán cầm nắm chính xác

Xét một thí nghiệm cầm nắm một cây viết ta có yêu cầu tối thiểu của tần số trong hệ điều khiển động cơ cần để đáp ứng trong bài toán cầm nắm chính xác là từ 20Hz–2kHz.

Thí nghiệm 1: Cầm viết với góc 30 độ

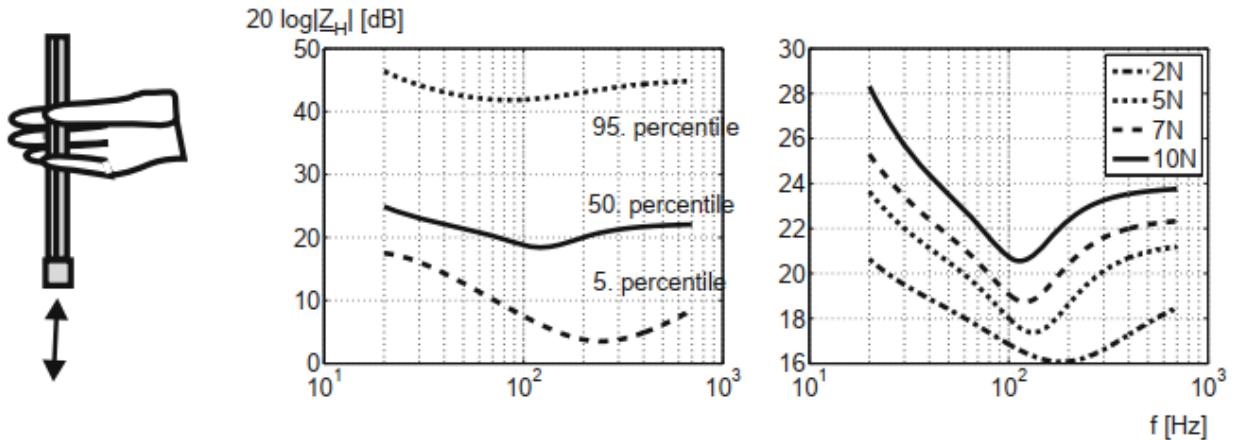


Hình 2.6: Thí nghiệm cầm viết góc 30 độ

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

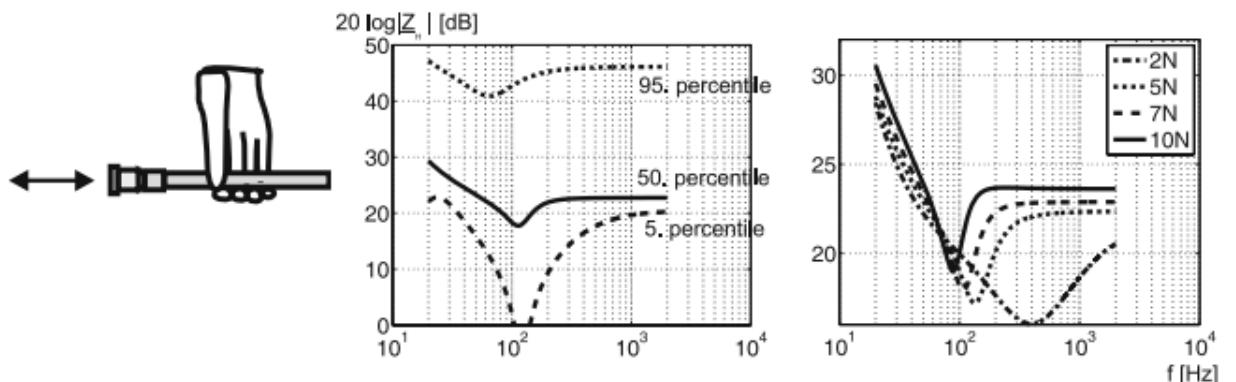
Trong thí nghiệm này tần số đáp ứng của hệ là từ 20Hz- 950Hz với cây viết có đường kính: 10mm

Thí nghiệm 2: Cầm viết với góc 90 độ



Hình 2.7: Thí nghiệm cầm viết góc 90 độ

Ở thí nghiệm này tần số đáp ứng của hệ là từ 20Hz – 700Hz với cây viết đường kính: 10mm



Hình 2.8: Thí nghiệm cầm viết góc 0 độ

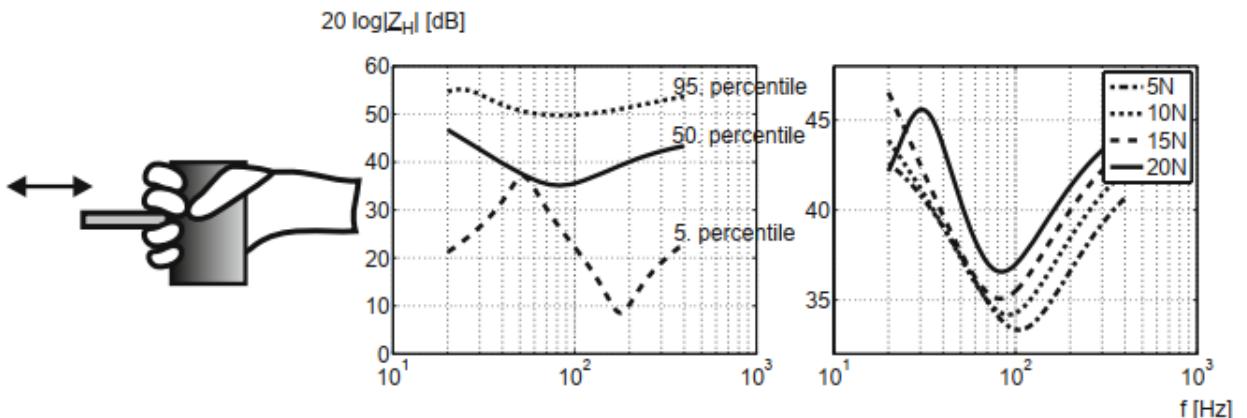
Thí nghiệm 3: Cầm viết với góc 0 độ

2.1.2 Bài toán cầm nắm chặt

Xét các thí nghiệm với bài toán cầm nắm chặt ta thu được kết quả tần số 20Hz – 2kHz

Thí nghiệm 1: Cầm chặt khói trụ

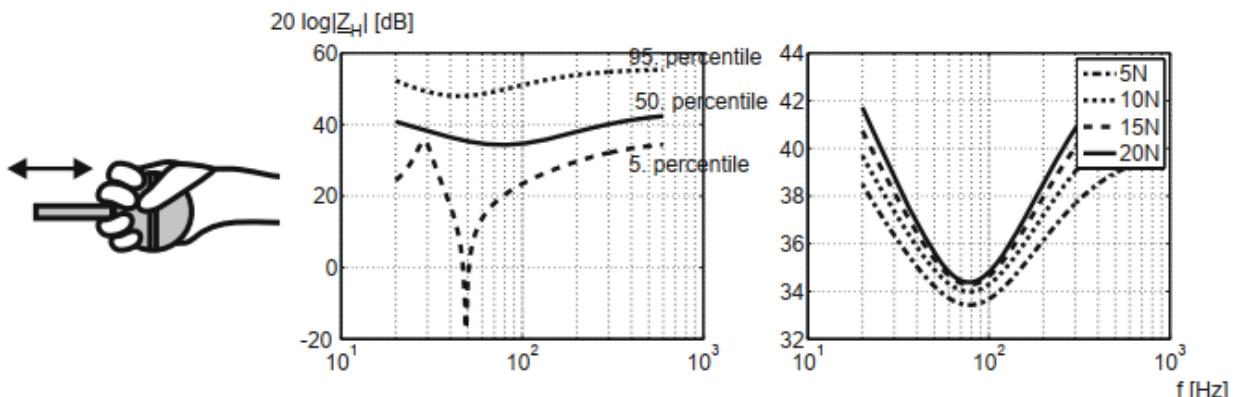
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 2.9: Thí nghiệm nắm chặt khối trụ tròn

Ta thu được kết quả như sau tần số của hệ từ 20Hz-400Hz.

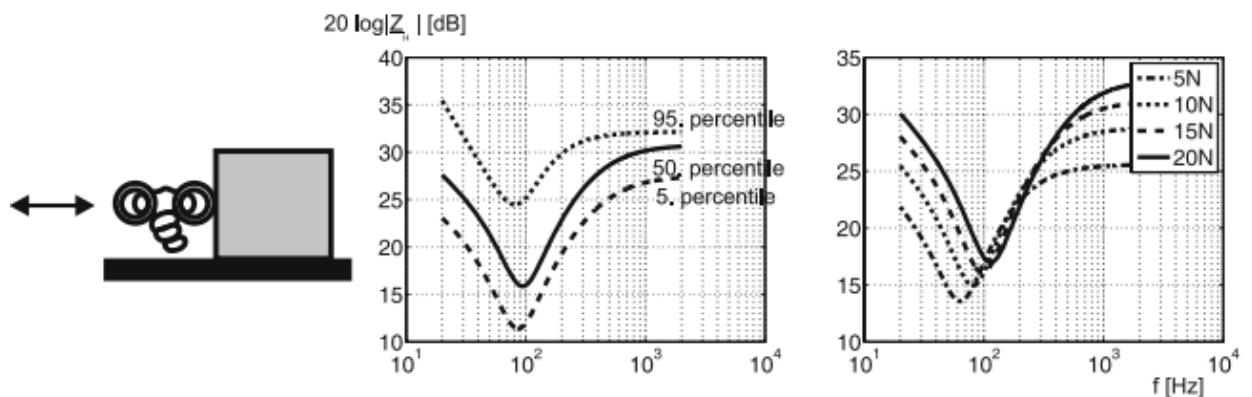
Thí nghiệm 2: Cầm chặt khối cầu



Hình 2.10: Thí nghiệm nắm chặt khối cầu

Ta thu được kết quả như sau tần số của hệ từ 20Hz-600Hz

Thí nghiệm 3: Cầm kéo hai vòng sắt có đường kính 25mm



Hình 2.11: Thí nghiệm nắm chặt với vòng sắt

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo
Ta thu được kết quả như sau tần số của hệ từ 20Hz-2kHz^[3].

Kết luận

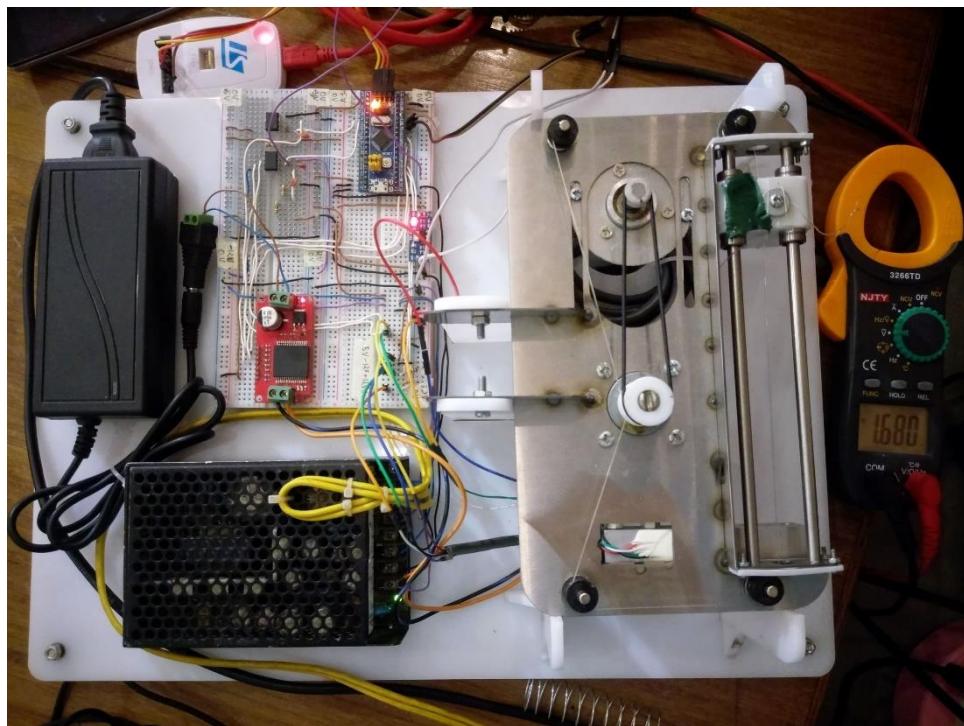
Từ các thí nghiệm với hệ bài toán cầm nắm chính xác và hệ bài toán cầm nắm chặt ta
có được tần số điều khiển động cơ từ 20Hz – 2kHz.

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

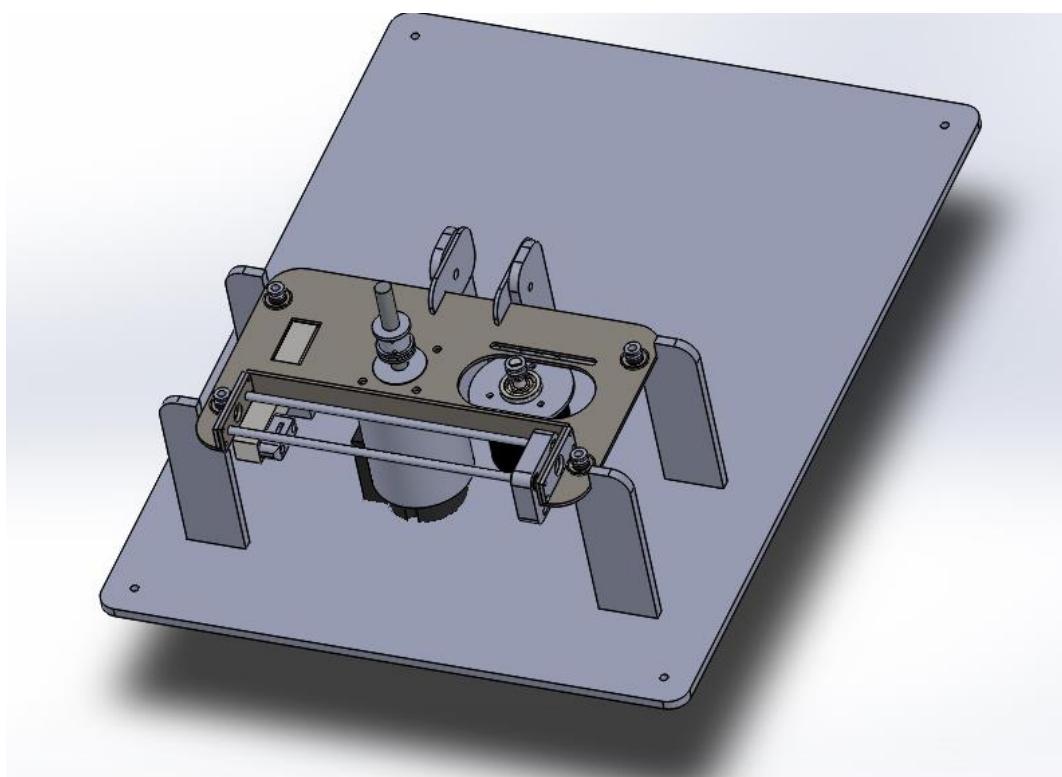
CHƯƠNG 3 THIẾT KẾ VÀ CHẾ TẠO CƠ KHÍ

1. Bản vẽ thiết kế

Bản vẽ tổng thể

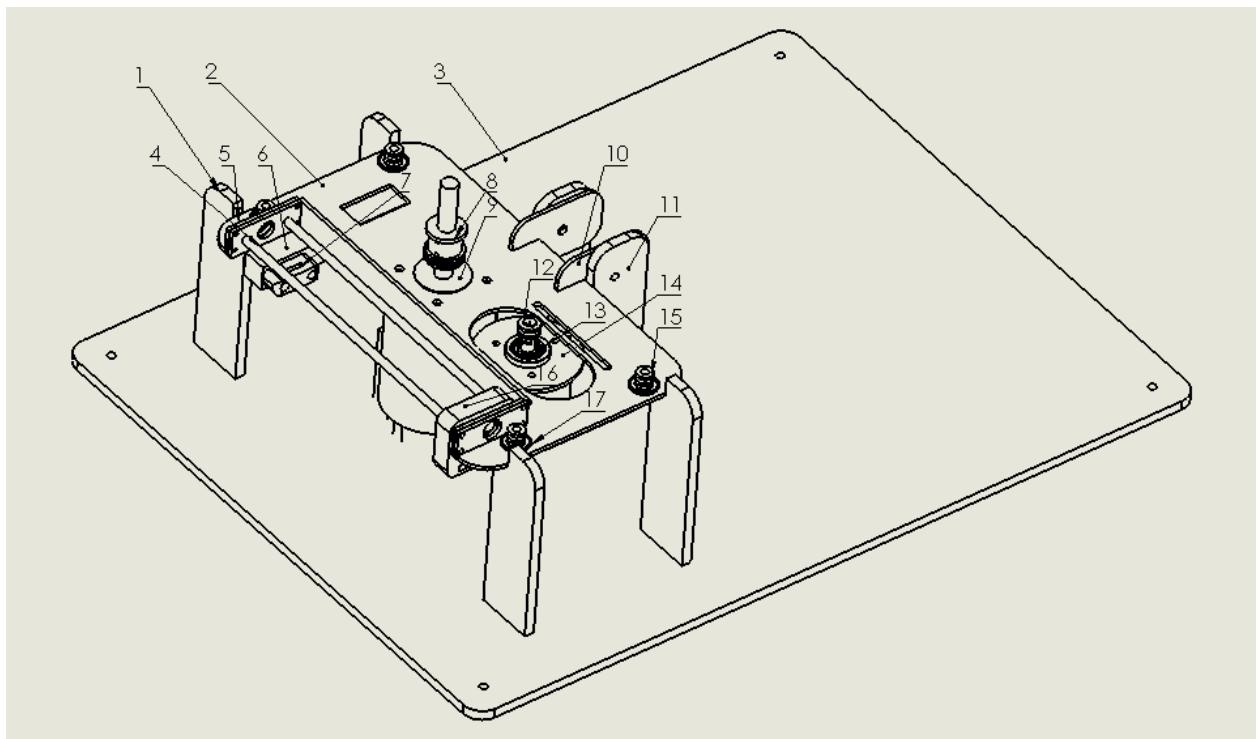


Hình 3.2: Bản thiết kế 3D



Hình 3.1: Bản vẽ thiết kế 3d

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 3.3: Bản vẽ chi tiết

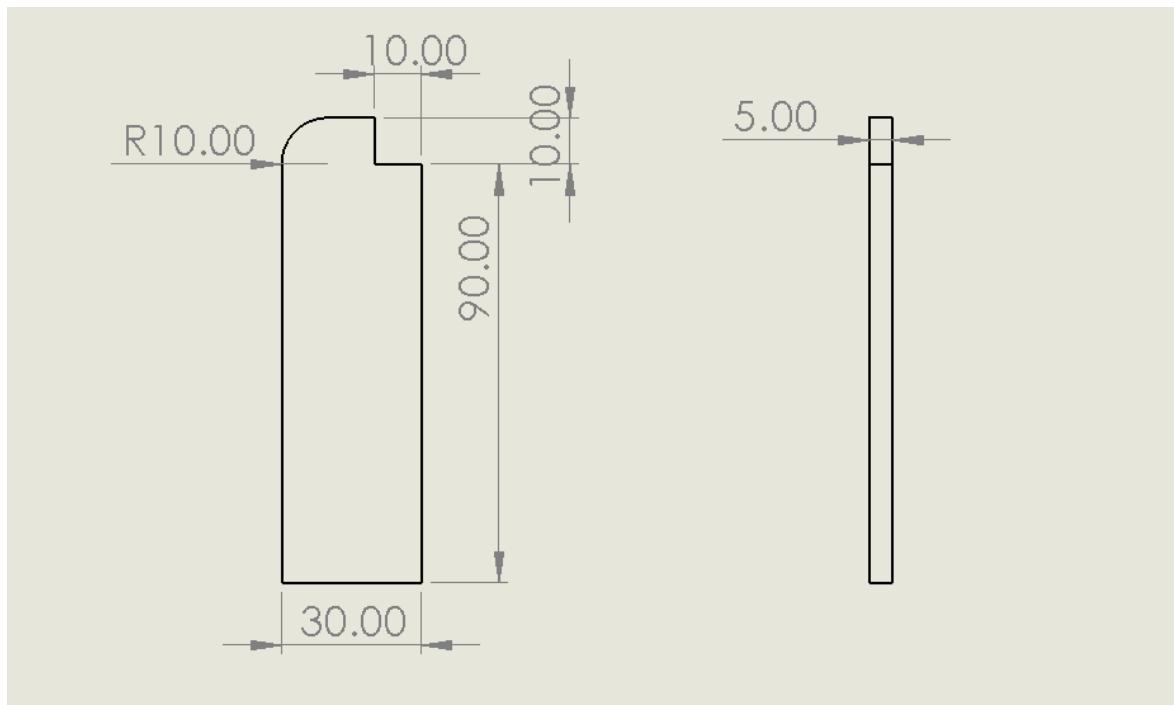
Bảng 3.1: Thông kê chi tiết

STT	Số lượng	Tên chi tiết	Chất liệu	Mô tả
1	4	Thanh đỡ khung máy 1	Mica	
2	1	Khung máy	Inox	
3	1	Bàn máy	Mica	
4	2	Trục dẫn Ø 5mm	Thép CT45	
5	2	Óp chặn trục	Nhựa PLA	
6	1	Loadcell		
7	1	Khớp đặt ngón cái	NhỰC PLA	
8	1	Bánh dẫn động cơ	NhỰC PLA	
9	1	Động cơ DC		
10	2	Khớp đỡ khung máy	Mica	
11	2	Thanh đỡ khung máy 2	Mica	
12	1	Bánh dẫn động encoder	NhỰC PLA	
13	1	Encoder		
14	1	Khớp điều chỉnh encoder		
15	4	Bánh dẫn động	Đồng thau	
16	1	Khớp đặt ngón trỏ		
17	4	Bạc lót Ø 4mm		

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

1. Thiết kế các bộ phận gia công

1.1 Thanh đỡ khung máy 1

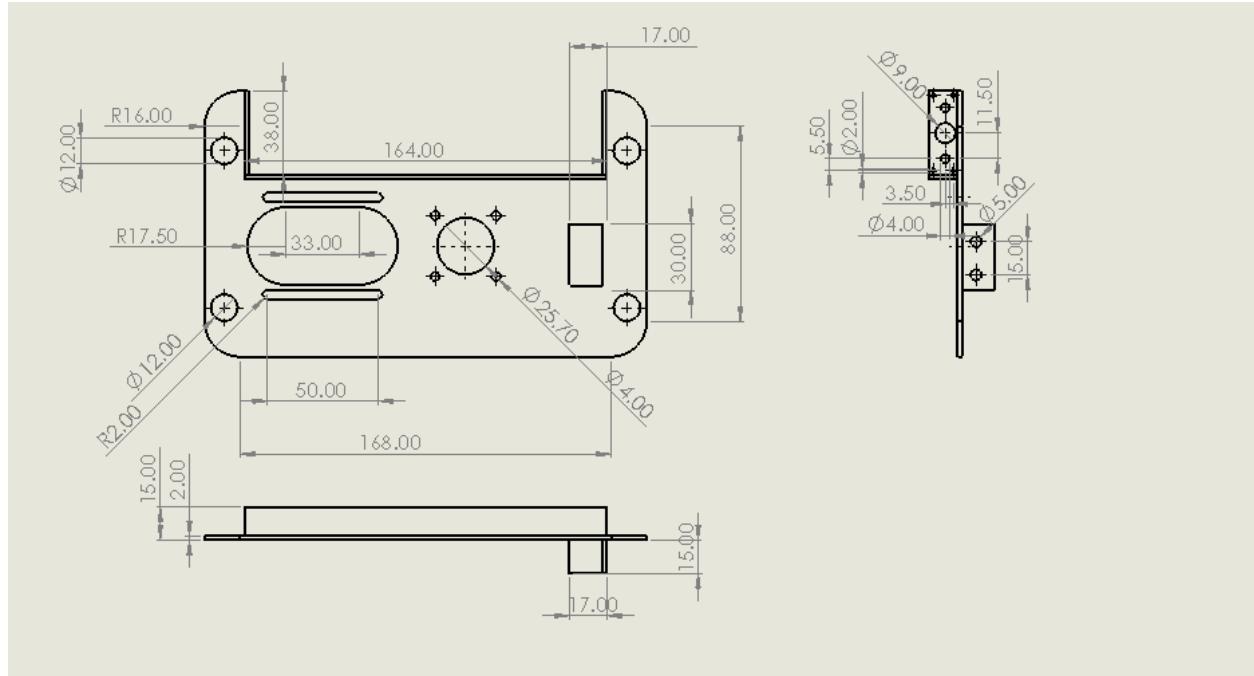


Hình 3.4: Thanh đỡ khung máy 1

Bảng 3.2: Thông số thanh đỡ khung máy

Thông số	Giá trị
Vật liệu	Mica
Hình thức gia công	Cắt laser CNC
Công dụng	Đỡ cố định khung máy

1.2 Khung máy

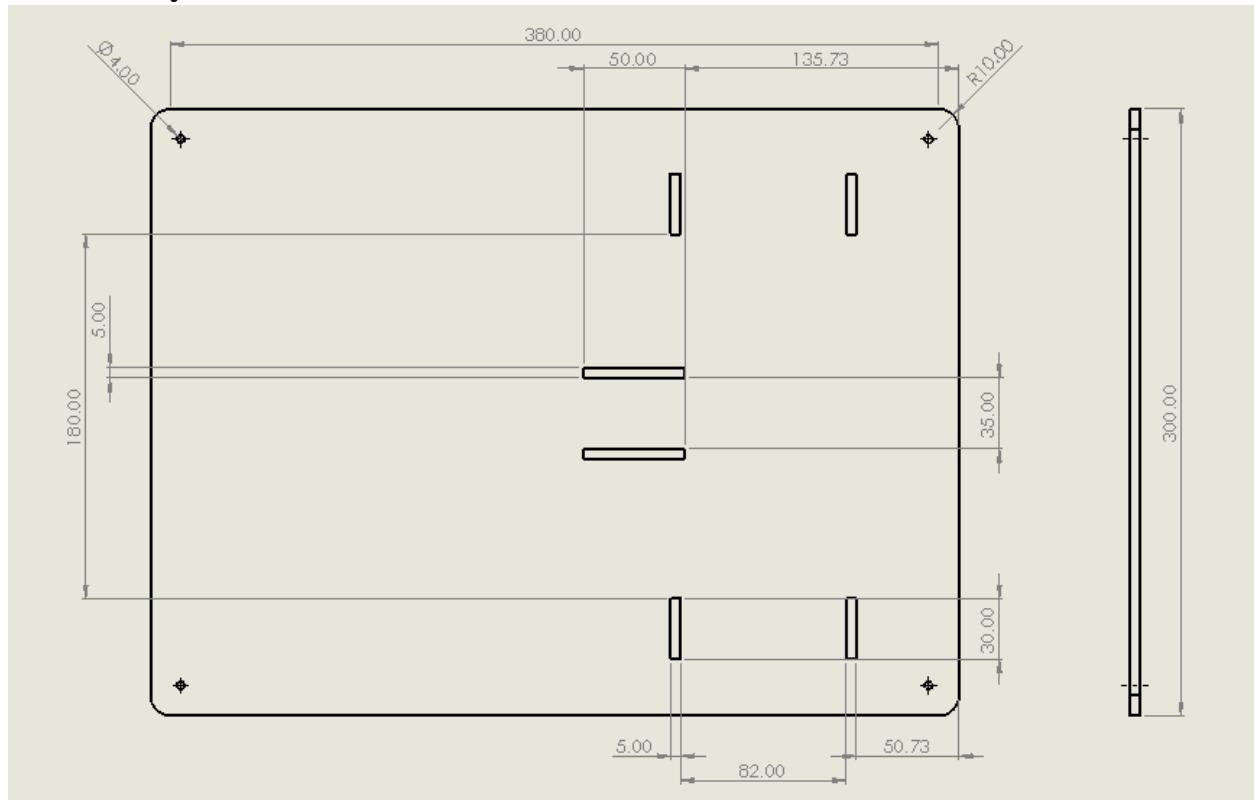


Hình 3.5: Khung máy

Bảng 3.3: Thông số khung máy

Thông số	Giá trị
Vật liệu	Inox
Hình thức gia công	Cắt laser CNC, khoan lỗ, hàn
Công dụng	Cố định các chi tiết.

1.3 Bàn máy



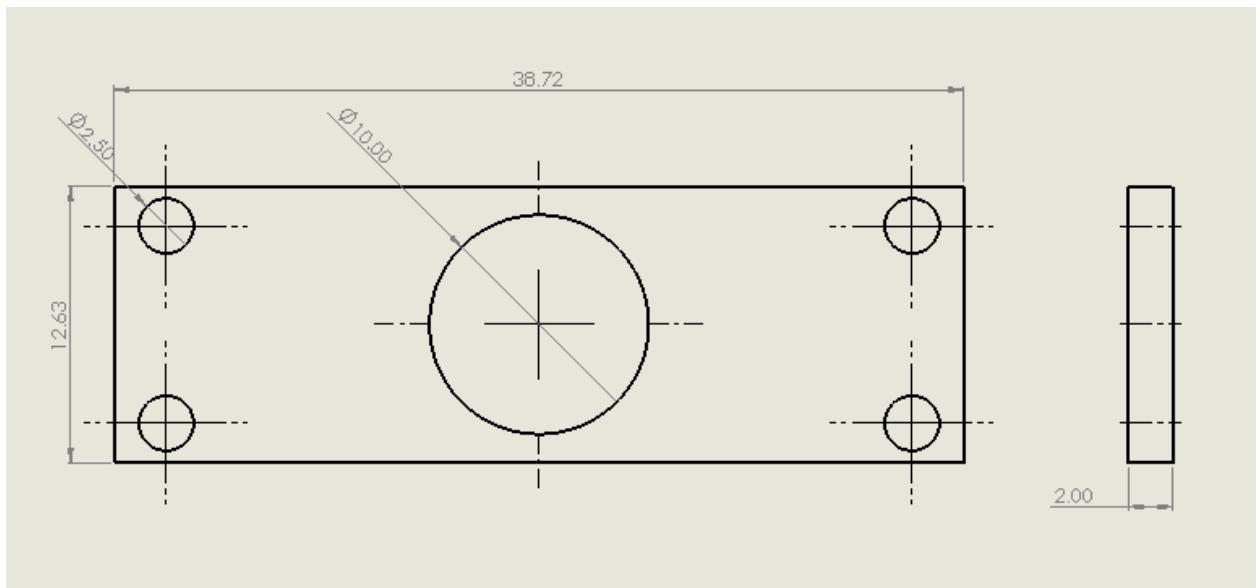
Hình 3.6: Bàn máy

Bảng 3.4: Thông số bàn máy

Thông số	Giá trị
Vật liệu	Mica
Hình thức gia công	Cắt laser CNC
Công dụng	Cố định các chi tiết

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

1.4 Ốp chặn trực

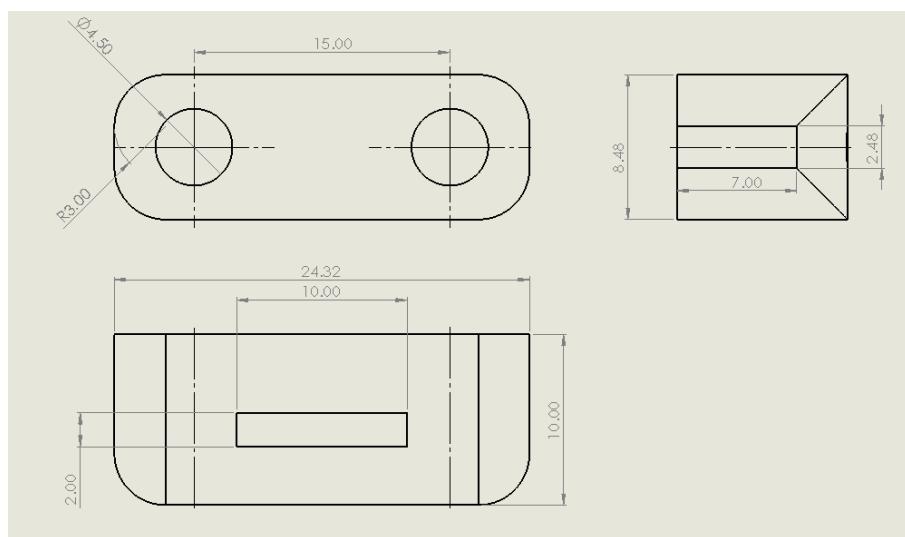


Hình 3.7: Ốp chặn trực

Bảng 3.5: Thông kê ốp chặn trực

Thông số	Giá trị
Vật liệu	Nhựa PLA
Hình thức gia công	In 3D
Công dụng	Chặn chiều chuyển động tịnh tiến của trực

1.5 Khớp đặt ngón cái

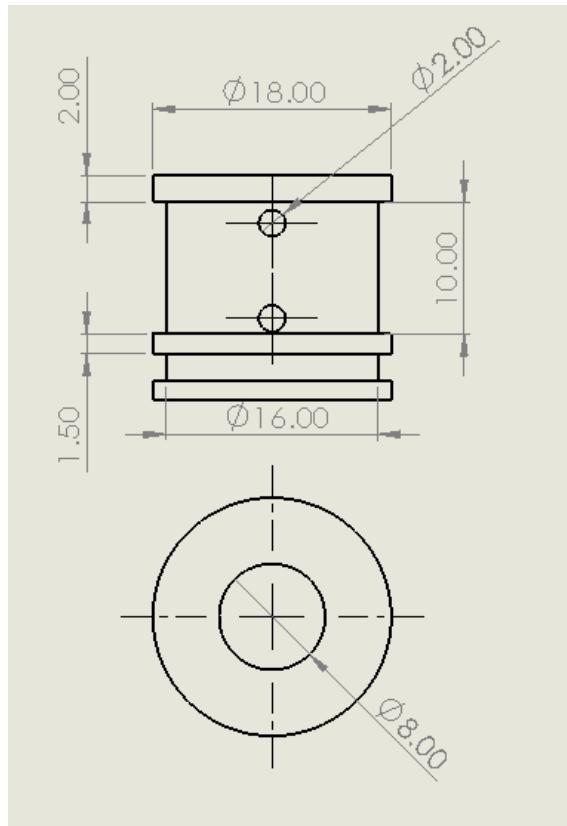


Hình 3.8: Khớp đặt ngón cái

Bảng 3.6: Khớp đặt ngón cái

Thông số	Giá trị
Vật liệu	Nhựa PLA
Hình thức gia công	In 3D
Công dụng	Nơi đặt ngón cái nối với loadcell

1.6 Bánh dẫn động cơ

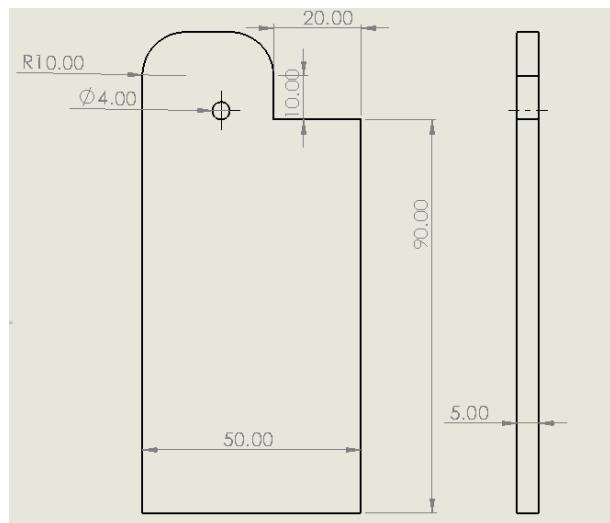


Hình 3.9: Bánh dẫn động cơ

Bảng 3.7: Bánh dẫn động cơ

Thông số	Giá trị
Vật liệu	Nhôm
Hình thức gia công	Tiện, khoan
Công dụng	Truyền động Moment xoắn động cơ

1.7 Thanh đỡ khung máy 2

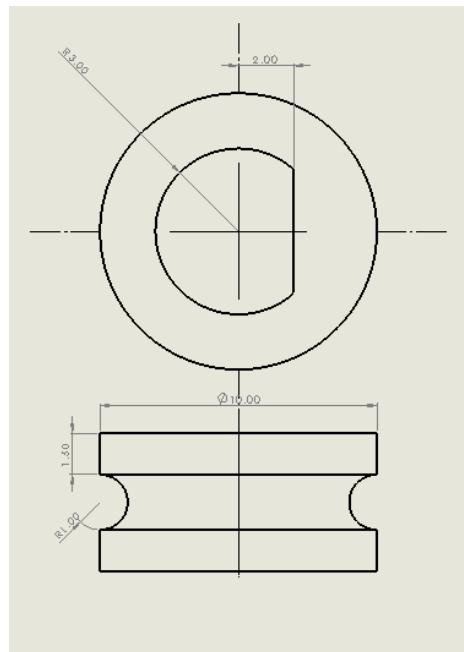


Hình 3.10: Thanh đỡ khung máy 2

Bảng 3.8: Bảng thông số Thanh đỡ khung máy 2

Thông số	Giá trị
Vật liệu	Mica
Hình thức gia công	Cắt laser CNC
Công dụng	Đỡ khung máy

1.8 Bánh dẫn encoder



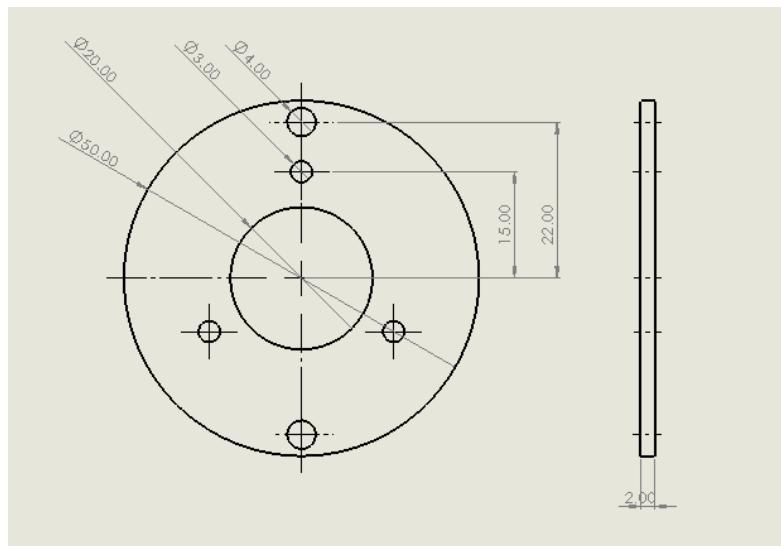
Hình 3.11: Bánh dẫn encoder

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Bảng 3.9: Bảng thông số bánh dẫn encoder

Thông số	Giá trị
Vật liệu	Nhựa PLA
Hình thức gia công	In 3D
Công dụng	Truyền động encoder

1.9 Khớp điều chỉnh vị trí encoder

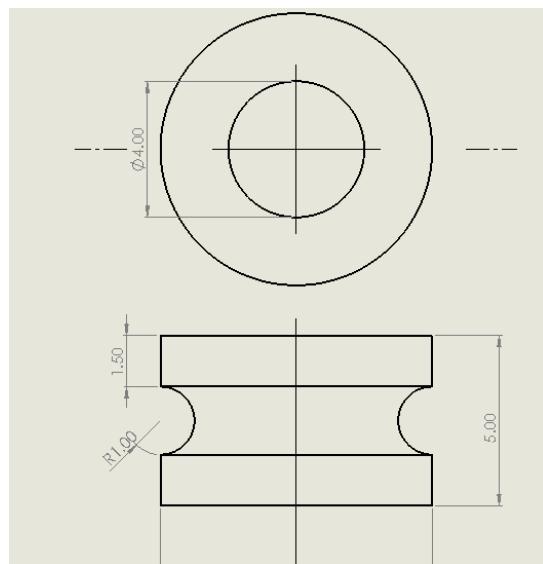


Hình 3.12: Khớp điều khiển vị trí

Bảng 3.10: Thông số khớp điều chỉnh vị trí encoder

Thông kê	Giá trị
Vật liệu	Inox
Hình thức gia công	Cắt laser CNC
Công dụng	Điều chỉnh vị trí encoder

1.10 Bánh dẫn động

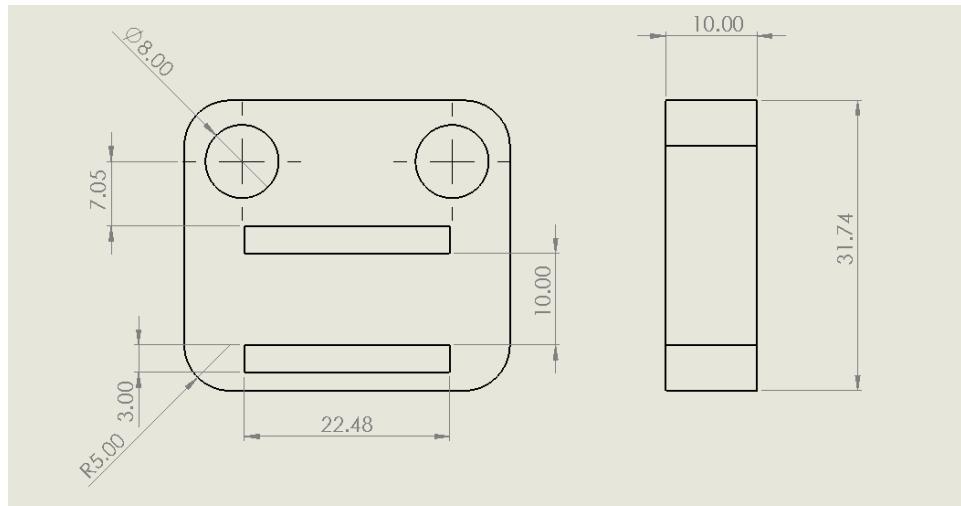


Hình 3.13: Bánh dẫn động

Bảng 3.11: Thông số bánh dẫn động

Thông số	Giá trị
Vật liệu	Nhựa PLA
Hình thức gia công	In 3D
Công dụng	Điều hướng truyền động của động cơ

1.11 Khớp đặt ngón trỏ



Hình 3.14: Khớp đặt ngón trỏ

Bảng 3.12: Thông số khớp đặt ngón trỏ

Thông số	Giá trị
Vật liệu	Nhựa PLA

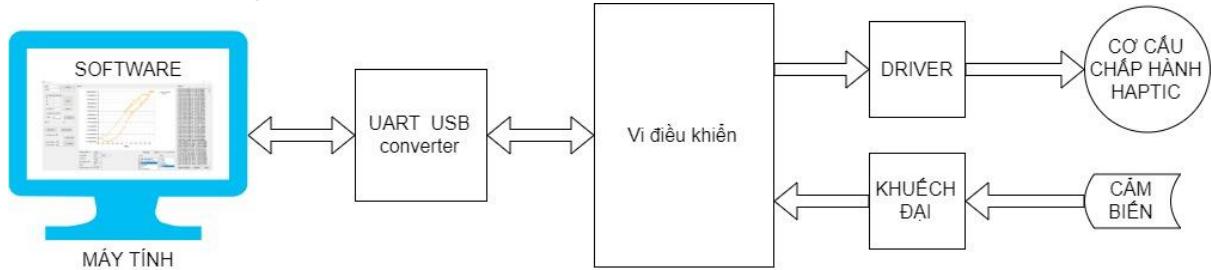
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Hình thức	gia công: In 3D
Công dụng	Nơi đặt ngón trỏ

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

CHƯƠNG 4 THIẾT KẾ HỆ THỐNG GIÁM SÁT, ĐO LUỒNG VÀ ĐIỀU KHIỂN MÔ HÌNH HAPTICS 1 BẬC TỰ DO TRÊN MÁY TÍNH

1. Sơ đồ hệ thống



Hình 4.1: Sơ đồ chức năng hệ thống

Để dễ dàng hơn trong việc xử lý dữ liệu, cần có một chương trình có thể quan sát, lưu trữ, ra lệnh điều khiển cho vi điều khiển.

Các tệp dữ liệu trong các thí nghiệm cần thiết được lưu lại trên máy tính để lợi dụng sức mạnh của máy tính và các phần mềm mã nguồn mở có sẵn để phân tích các tệp dữ liệu lớn.

2. Danh sách thiết bị

Bảng 4.1: Thông kê chức năng thiết bị

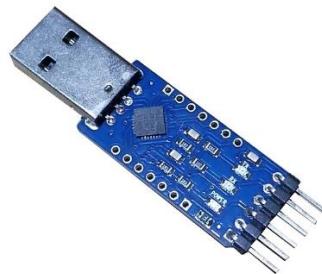
sst	Thành phần	Chức năng	Phần cứng
1	Phần mềm máy tính	<ul style="list-style-type: none">- Hiển thị biểu đồ- Hiển thị dữ liệu- lưu trữ dữ liệu- xử lý dữ liệu- ra lệnh điều khiển	<ul style="list-style-type: none">- Máy tính cá nhân
2	Mạch chuyển đổi UART-USB	Chuyển đổi mức giao tiếp uart sang usb làm cầu nối giao tiếp giữa máy tính và vi điều khiển	<ul style="list-style-type: none">- Module chuyển đổi USB-UART CP2102
3	Driver	Mạch công suất, khuếch đại tín hiệu điều khiển từ vi điều	<ul style="list-style-type: none">- Module mạch cầu H VNH2SP30 30A

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

		khiển để điều khiển cơ cấu chấp hành	
4	Khuếch đại	Khuếch đại tín hiệu đo lường của cảm biến	<ul style="list-style-type: none"> - INA128 - OPAMP LM325
5	Vị điều khiển	<ul style="list-style-type: none"> - nhận mệnh lệnh từ máy tính - Thực thi thuật toán điều khiển cơ cấu chấp hành - Thu thập dữ liệu cảm biến 	<ul style="list-style-type: none"> - STM32F103C8T6
6	Cơ cấu chấp hành haptic	<ul style="list-style-type: none"> - Sinh lực mô phỏng tác động lên tay người dùng 	<ul style="list-style-type: none"> - Động cơ DC - Cơ cấu tang trống kéo dây
7	Cảm biến	<ul style="list-style-type: none"> - Đo lực tác động lên cơ cấu chấp hành - Đo dòng tiêu thụ động cơ - Đo góc quay động cơ 	<ul style="list-style-type: none"> - Loadcell - Encoder

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

2.1 MẠCH CHUYỂN ĐỔI UART- USB

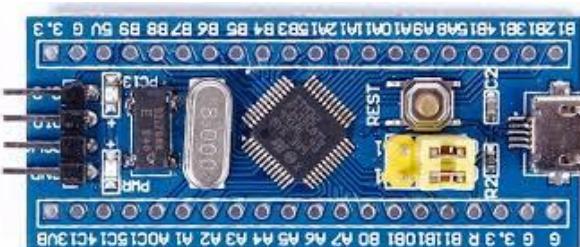


Hình 4.1: USB UART

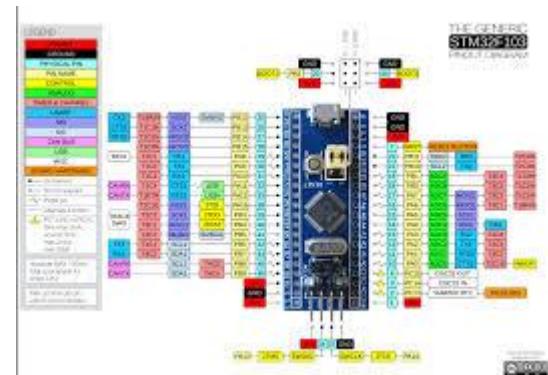
2.2 KIT PHÁT TRIỂN VI ĐIỀU KHIỂN STM32F103C8T6 BLUE PILL

Để thực hiện bài toán yêu cầu tốc độ đọc, xử lý cao và đọc Encoder thì việc sử dụng Arduino để giải quyết gần như là không thể. Vì vậy chúng ta cần một loại mạch điều khiển khác có tốc độ lớn hơn. Ta chọn mạch STM32F103C8T6 để có thể giải quyết được bài toán này.

2.3 CẢM BIẾN



Hình 4.2: Mạch điều khiển trong thực tế



Hình 4.3: Sơ đồ chân mạch điều khiển

Vì độ biến dạng của vật là rất thấp nên chúng ta cần một encoder có độ phân giải cao. Với độ phân giải 1000 xung/vòng tương ứng với độ phân giải 0.36 độ/vòng thì encoder Omron E6B2-CWZ6C gần như đáp ứng được đầy đủ.

Encoder Omron E6B2-CWZ6C

Bảng 4.2: Datasheet Encoder

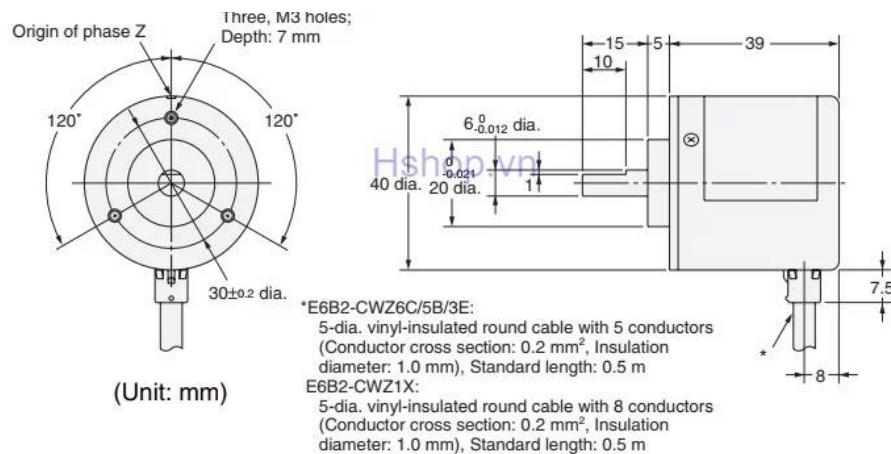
Thông số kỹ thuật	Giá trị
Điện áp sử dụng:	5~24VDC.
Dòng tiêu thụ:	max 80mA

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Số xung:	1000 xung / 1 vòng (1000 p/r)
Số kênh xung:	3 kênh xung riêng biệt A, B, Z.
Tần số đáp ứng tối đa:	100Khz
Dạng ngõ ra xung:	NPN cực thu hở (cân măc trơ treo lên VCC để tạo mức cao (High))
Đường kính trục:	6mm
Đường kính thân:	40mm



Hình 4.5: Encoder thực tế



Hình 4.4: Bản vẽ kích thước encoder

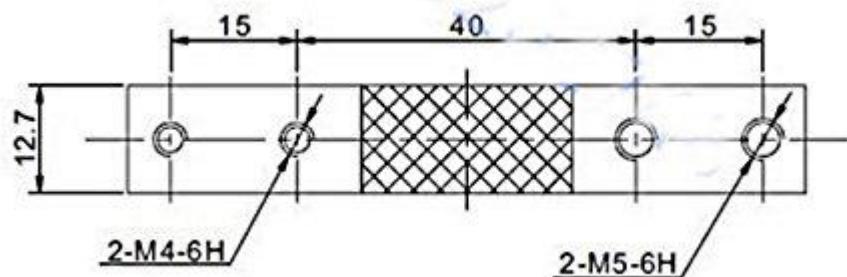
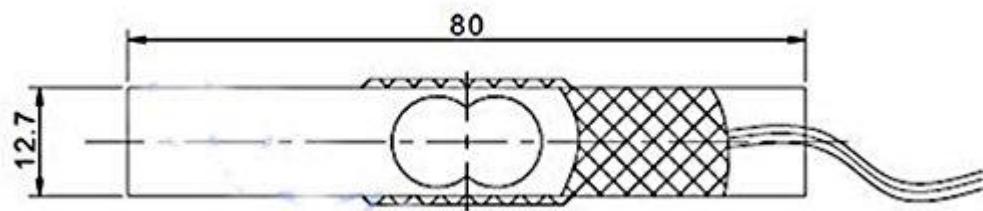
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

2.3.1 Loadcell

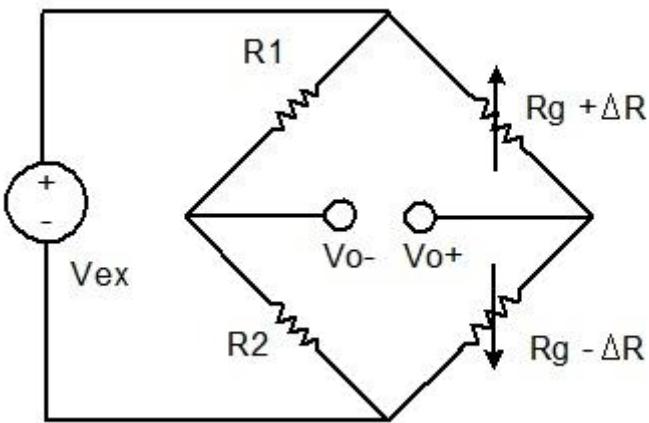
Ta cần chọn loadcell để làm việc với ngón tay và khả năng đáp ứng phù hợp với công suất động cơ bên trên, nên ta chọn loadcell 1kg là vừa đủ để đo đạt tính toán tác dụng lực ngón tay lên cơ cấu.



Hình 4.7: Loadcell thực tế



Hình 4.6: Kích thước loadcell



Hình 4.8: Sơ đồ nguyên lý loadcell

Bảng 4.3: Datasheet loadcell

Thông số kỹ thuật	Giá trị
Tải trọng	1Kg
Rated Output (mV/V)	1.0 +- 0.15
Độ lệch tuyến tính (%)	0.05
Creep (5min) %	0.1
Ảnh hưởng nhiệt độ tới độ nhạy %RO/ độ C	0.003
Ảnh hưởng nhiệt độ tới điểm không %RO/ độ C	0.02
Độ cân bằng điểm không %RO	+ -0.1
Trở kháng đầu vào (Ω)	1066 +- 20
Trở kháng ngõ ra (Ω)	1000 +- 20
Trở kháng cách li ($M\Omega$) 50V	2000
Điện áp hoạt động	5V
Nhiệt độ hoạt động	
-20 ~ 65 độ C	
Safe Overload %RO	120
Ultimate overload %RO	150
Chất liệu cảm biến	Nhôm
Độ dài dây	180mm
Dây đỗ	Ngõ vào (+)

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Dây đen	Ngõ vào (-)
Dây xanh Lá	Ngõ ra (+)
Dây trắng	Ngõ ra (-)

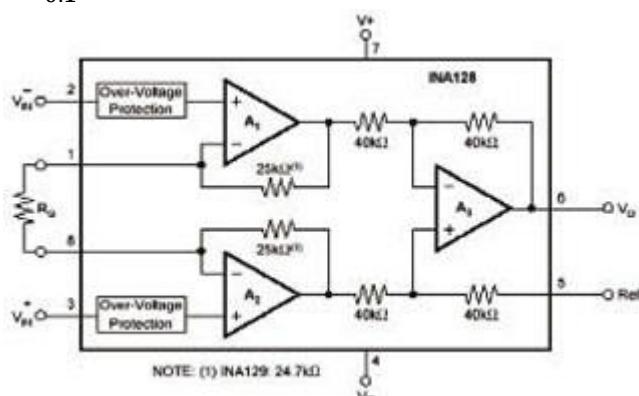
2.3.2 Mạch khuếch đại

a) INA 128P

Đầu ra của loadcell là đại lượng điện rất nhỏ so với tính hiệu điện mà mạch điều khiển có thể đọc. Vì vậy ta cần một mạch khuếch đại để đọc đại lượng điện của loadcell. Trong các mạch khuếch đại thì mạch khuếch đại đo lường là hiệu quả nhất. Chọn mạch khuếch đại cũng tránh thêm việc tăng thêm nội dung cần xử lý và làm chậm tốc độ đọc như HX711.

Độ phân giải vi điều khiển có thể đọc được:

$$3.3/(2^{12}-1) = 0.81 \text{ mV} > 0.1 \text{ mV} \text{ (Độ phân giải loadcell)} \Rightarrow \text{hệ số khuếch đại tín hiệu loadcell nhỏ nhất là } \frac{0.81}{0.1} = 8.1$$



Hình 4.9: Sơ đồ chân mạch INA 128

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Bảng 4.5: Chọn thông số R khuếch đại INA128

$$G = 1 + \frac{50 \text{ k}\Omega}{R_G} \quad G = 1 + \frac{49.4 \text{ k}\Omega}{R_G}$$

DESIRED GAIN (V/V)	INA128		INA129	
	R _G (Ω)	NEAREST 1% R _G (Ω)	R _G (Ω)	NEAREST 1% R _G (Ω)
1	NC	NC	NC	NC
2	50K	49.9K	49.4K	49.9K
5	12.5K	12.4K	12.35K	12.4K
10	5.556K	5.62K	5489	5.49K
20	2.632K	2.61K	2600	2.61K
50	1.02K	1.02K	1008	1K
100	505.1	511	499	499
200	251.3	249	248	249
500	100.2	100	99	100
1000	50.5	49.9	49.5	49.9
2000	25.01	24.9	24.7	24.9
5000	10	10	9.88	9.76
10000	5.001	4.99	4.94	4.87

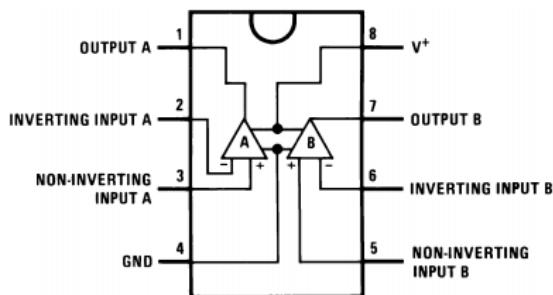
Bảng 4.4: Datasheet INA128

PARAMETER	TEST CONDITIONS	T _A = -55°C to +125°C			T _A = 175°C ⁽¹⁾			UNIT	
		MIN	TYP	MAX	MIN	TYP	MAX		
GAIN									
Gain equation			1 + (50 kΩ/R _G)			1 + (50 kΩ/R _G)		V/V	
Range of gain		1	10000		1	10000		V/V	
Gain error	G = 1	±0.01	±0.1		±0.1%	±0.5%		%FSR	
	G = 10	±0.02	±0.5		±0.5%	±1%			
	G = 100	±0.05	±0.7		±0.7%	±1.5%			
	G = 1000	±0.5	±2.5		±2%	±4%			
Gain vs temperature ⁽²⁾	G = 1	±1	±10		±75			ppm/°C	
50-kΩ resistance ⁽³⁾⁽⁴⁾		±25	±100		±75			ppm/°C	
Nonlinearity	V _O = ±13.6 V, G = 1	±0.0001	±0.001			±0.008		% of FSR	
	G = 10	±0.0003	±0.002			±0.01			
	G = 100	±0.0005	±0.002			±0.01			
	G = 1000	±0.001	See ⁽⁵⁾		±0.6	See ⁽⁵⁾			
OUTPUT									
Voltage	Positive	R _L = 10 kΩ	(V+) - 1.4	(V+) - 0.9	(V+) - 1.4	(V+) - 0.9		V	
	Negative	R _L = 10 kΩ	(V-) + 1.4	(V-) + 0.8	(V-) + 1.4	(V-) + 0.8			
Load capacitance stability			1000		1000			pF	
Short-circuit current			+6/-15		+6/-15			mA	
FREQUENCY RESPONSE									
Bandwidth, -3 dB	G = 1		1300		1100			kHz	
	G = 10		700		700				
	G = 100		200		190				
	G = 1000		20		17.5				
Slew rate	V _O = ±10 V, G = 10		4		4			V/μs	
Settling time, 0.01%	G = 1		7		7			μs	
	G = 10		7		7				
	G = 100		9		9				
	G = 1000		80		80				
Overload recovery	50% overdrive		4		4			μs	
POWER SUPPLY									
Voltage range			±2.25	±15	±18	±2.25	±15	±18	V
Current, total	V _{IN} = 0 V			±0.7	±1			±1	mA
TEMPERATURE RANGE									
Specification			-55	+125			175	°C	
Operating			-55	+125			175	°C	

b) OPAMP LM358

Giá trị ADC thường bị nhiễu bởi các loại nhiễu tần số cao hoặc sụt giảm điện áp do biến thiên của dòng điện tác dụng lên điện trở ngõ vào. Giải pháp cách ly dòng đo

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo lường bằng cách sử dụng một opamp ở chế độ mạch đệm làm tăng điện trở ngõ vào ADC của VDK, và lọc nhiễu bằng một bộ lọc thông thấp.



Hình 4.10: Sơ đồ chân LM 358

Bảng 4.6: Datasheet LM358

PARAMETER	TEST CONDITIONS	LM158A			LM358A			LM158, LM258			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
Power Supply	$V^+ = 5 \text{ V to } 30 \text{ V}$										
Rejection Ratio	(LM2904, $V^+ = 5 \text{ V to } 26 \text{ V}$), $T_A = 25^\circ\text{C}$	65	100		65	100		65	100		dB
Amplifier-to-Amplifier Coupling	$f = 1 \text{ kHz to } 20 \text{ kHz}$, $T_A = 25^\circ\text{C}$ (Input Referred), See ⁽⁵⁾		-120			-120			-120		dB
Output Current	Source	$V_{IN}^+ = 1 \text{ V}$,									
		$V_{IN}^- = 0 \text{ V}$,									
		$V^+ = 15 \text{ V}$,									
		$V_O = 2 \text{ V}$, $T_A = 25^\circ\text{C}$									
	Sink	$V_{IN}^+ = 1 \text{ V}$, $V_{IN}^- = 0 \text{ V}$									
		$V^+ = 15 \text{ V}$, $T_A = 25^\circ\text{C}$,									
		$V_O = 2 \text{ V}$									
		$V_{IN}^- = 1 \text{ V}$,									
		$V_{IN}^+ = 0 \text{ V}$									
		$T_A = 25^\circ\text{C}$, $V_O = 200 \text{ mV}$,									
Short Circuit to Ground		$V^+ = 15 \text{ V}$									
	Input Offset Voltage	See ⁽²⁾			4			5		7	mV
	Input Offset Voltage Drift	$R_S = 0\Omega$		7	15		7	20		7	$\mu\text{V}/^\circ\text{C}$
	Input Offset Current	$ I_{IN(+)} - I_{IN(-)} $			30			75		100	nA
	Input Offset Current Drift	$R_S = 0\Omega$		10	200		10	300		10	$\text{pA}/^\circ\text{C}$
	Input Bias Current	$I_{IN(+)} \text{ or } I_{IN(-)}$		40	100		40	200		40	300
	Input Common-Mode Voltage Range	$V^+ = 30 \text{ V}$, See ⁽⁴⁾ (LM2904, $V^+ = 26 \text{ V}$)		0	$V^+ - 2$		0	$V^+ - 2$		0	$V^+ - 2$
	Large Signal Voltage Gain	$V^+ = +15 \text{ V}$					15			25	
		$(V_O = 1 \text{ V to } 11 \text{ V})$									V/mV
		$R_L \geq 2 \text{ k}\Omega$									
Output Voltage	V_{OH}	$V^+ = +30 \text{ V}$	$R_L = 2 \text{ k}\Omega$	26			26			26	V
		$(\text{LM2904}, V^+ = 26 \text{ V})$	$R_L = 10 \text{ k}\Omega$	27	28		27	28		27	V
Swing	V_{OL}	$V^+ = 5 \text{ V}$, $R_L = 10 \text{ k}\Omega$		5	20		5	20		5	20
Output Current	Source	$V_{IN}^+ = +1 \text{ V}$, $V_{IN}^- = 0 \text{ V}$,		10	20		10	20		10	20
		$V^+ = 15 \text{ V}$, $V_O = 2 \text{ V}$									
	Sink	$V_{IN}^- = +1 \text{ V}$, $V_{IN}^+ = 0 \text{ V}$,		10	15		5	8		5	8
		$V^+ = 15 \text{ V}$, $V_O = 2 \text{ V}$									

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

2.4 Nguồn và mạch công suất

2.4.1 Nguồn VSF30-DD

Vì cần sử dụng mạch khuếch đại, nên ta xài nguồn triple +5, ± 12 VSF30-DD

Bảng 4.7: Datasheet nguồn

VSF30-TRIPLE SPECIFICATIONS

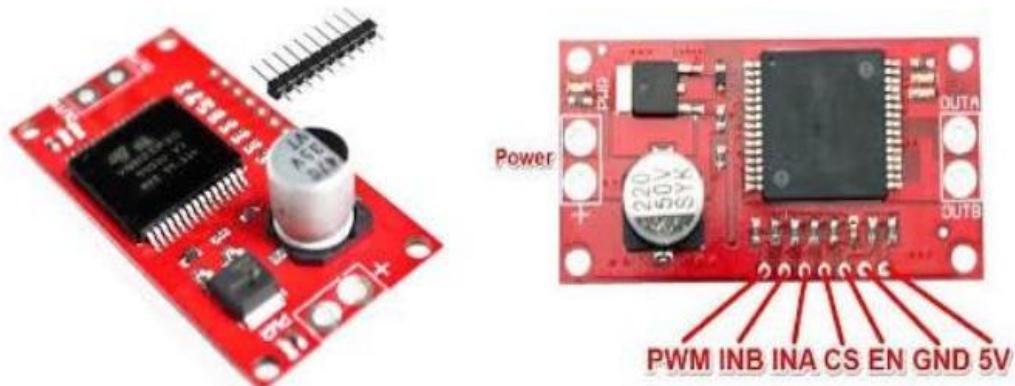
	MODEL		VSF30-BDB			VSF30-DD			VSF30-EE		
INPUT	VOLTAGE, FREQUENCY		AC100-240V (AC85 ~ 264V or DC125 ~ 370V), 50/60Hz (47 ~ 440Hz) Universal Input								
	CURRENT	110V				0.65 ($I=100\%$)					
		220V				0.35 ($I=100\%$)					
	EFFICIENCY	110V				66 typ			66 typ		
		220V							66 typ		
INRUSH CURRENT		20A typ(AC IN 110V, $I=100\%$), 40A typ(AC IN 220V, $I=100\%$) at cold start.									
OUTPUT	VOLTAGE	[V]	5	12	-5	5	12	-12	5	15	-15
	CURRENT	[A]	0.3 ~ 3	1	0.3	0.3 ~ 3	1.2	0.3	0.3 ~ 3	0.8	0.2
	REGULATION, LINE	[mV]	25 Max	60 Max	25 Max	25 Max	60 Max	60 Max	25 Max	75 Max	75 Max
	REGULATION, LOAD	[mV]	50 Max	120 Max	50 Max	50 Max	120 Max	120 Max	50 Max	150 Max	150 Max
	RIPPLE	[mVp-p]	50 Max	120 Max	50 Max	50 Max	120 Max	120 Max	50 Max	150 Max	150 Max
	RIPPLE, NOISE	[mVp-p]	100 Max	170 Max	100 Max	100 Max	170 Max	170 Max	100 Max	200 Max	200 Max
	TEMPERATURE DRIFT 0 → 50°C	[mV]	75 Max	180 Max	75 Max	75 Max	180 Max	180 Max	75 Max	225 Max	225 Max
	RISE TIME	[ms]	100 Max (AC IN 110V $I=100\%$)								
	HOLDING TIME	[ms]	10 typ (AC IN 110V $I=100\%$)								
PROTECTION CIRCUIT	OVER CURRENT PROTECTION		Works at over 110% of rating and recovers automatically(CH2,3 $I=100\%$)								
	OVER VOLTAGE PROTECTION		Works at 115 ~ 140% of rating								
ELECTRICALLY ISOLATED	INPUT-OUTPUT		AC 1,500V 1 minute current 10mA, DC 500V 100MΩ (At room temperature & Humidity)								
	INPUT-CASE, FG		AC 1,500V 1 minute current 10mA, DC 500V 100MΩ (At room temperature & Humidity)								
	OUTPUT-CASE, FG		AC 500V 1 minute current 100mA, DC 500V 100MΩ (At room temperature & Humidity)								
ENVIRONMENT	OPERATING TEMP AND HUMID		-10 ~ +50°C, 20 ~ 90% RH(Non condensing)								
	STORAGE TEMP AND HUMID		-20 ~ +75°C, 20 ~ 90% RH(Non condensing)								
SAFETY	VIBRATION		10 ~ 55Hz at 1G, 3 minutes period, 30 minutes along X, Y and Z axis								
	IMPACT		10G for 20ms once on each X, Y and Z axis								
	SAFETY REGULATION		Built to meet UL60950-1, EN60950-1								
	LINE CONDUCTED RF VOLTAGE		Built to meet EN55022-A								

Vì công suất nguồn bé hơn công suất động cơ nên ta cần phải sử dụng thêm nguồn ngoài với công suất 60w, điện áp 12v, dòng điện 5A.

2.4.2 Mạch cầu h VNH2SP30 30A

Mạch điều khiển động cơ VNH2SP30 30A được được sử dụng để điều khiển 1 động cơ với dòng tối đa lên đến 30A, mạch sử dụng IC Driver VNH2SP30 tích hợp cảm

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo
biến dòng tải, tự động ngắt khi quá công suất, mạch được thiết kế nhỏ gọn, chất lượng
tốt, dễ sử dụng.



Hình 4.11: Mạch cầu h trong thực tế

Bảng 4.8: Datasheet mạch cầu h

Thông số kỹ thuật	Giá trị
Điện áp hoạt động:	5.5 - 16VDC
Dòng định:	30A
Dòng liên tục:	14A
Trở kháng nội của MOSFET:	19mΩ
Tần số băm xung tối đa:	20kHz

2.5 Động cơ

2.5.1 Động cơ DME44BA



Hình 4.12: Động cơ trong thực tế

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

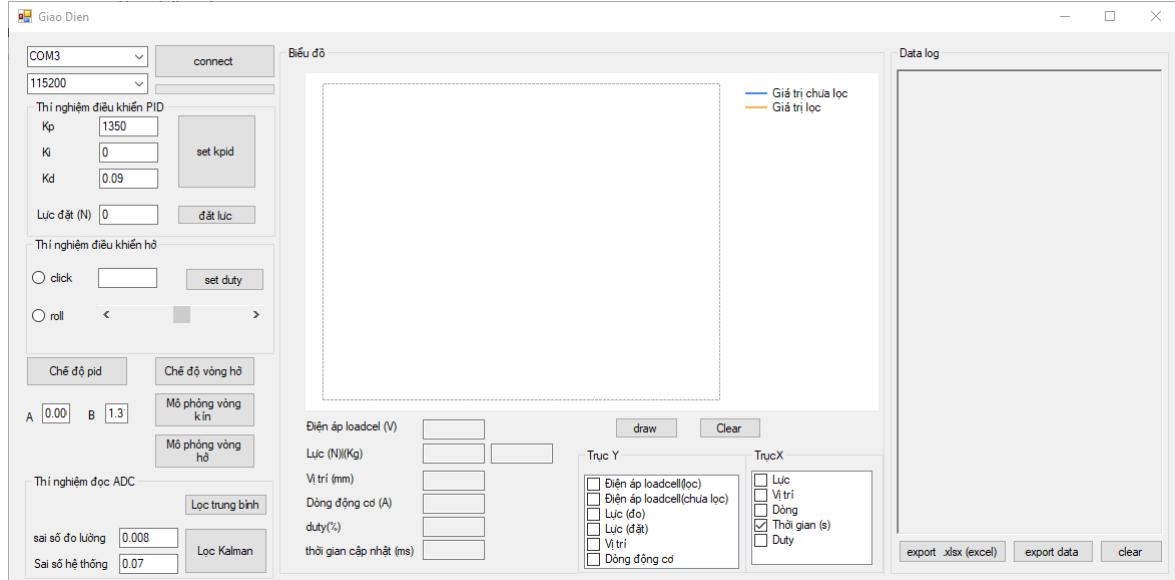
Bảng 4.9: Datasheet động cơ

Thông số kỹ thuật	Giá trị
Rated Voltage	12 V
Rated Output	14.8 W
Current Rating	940 mA
Rated Torque	39 mN·m
Rated Speed	3600 r/min
No Load Current	0.16 A
No Load Speed	4300 r/min
Diameter	44 mm
Weight	400 g
Stall Torque	250 mN·m

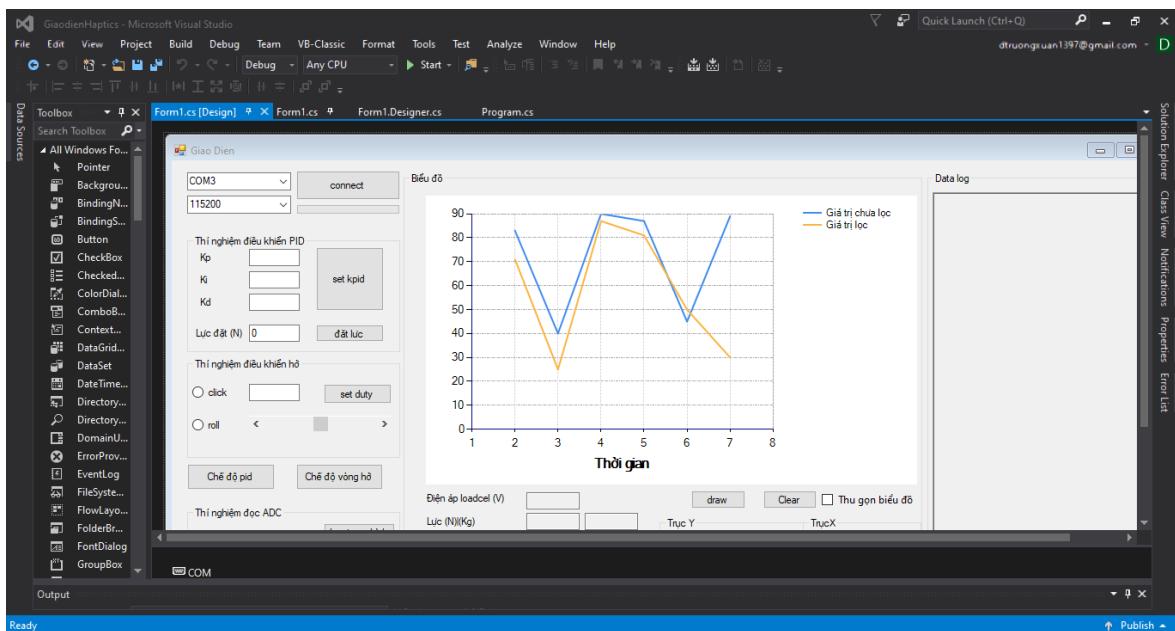
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

CHƯƠNG 5 THIẾT KẾ GIAO DIỆN PHẦN MỀM VÀ ĐỊNH NGHĨA KHUNG TRUYỀN TÍN HIỆU GIAO TIẾP VỚI VI ĐIỀU KHIỂN

1. Thiết kế giao diện phần mềm



Hình 5.1: Giao diện sau khi chạy



Hình 5.2: Giao diện khi thiết kế

Giao diện thiết kế bằng ngôn ngữ C#, dùng framework winform application.NET của Microsoft trên công cụ visual studio phiên bản 2015 của Microsoft

2. Định nghĩa khung truyền tin giữa giao diện máy tính và vi điều khiển

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Thông qua giao tiếp uart, mỗi 8 bit (1 kí tự ascii) truyền nhận bằng phần cứng được ghép lại thành một chuỗi ký tự có nghĩa được định nghĩa như dưới đây:

Khung truyền tin từ máy tính

Bảng 5.1: Khung truyền dữ liệu UART từ máy tính đến vi điều khiển

Ý nghĩa	Kí tự bắt đầu	dữ liệu					Kí tự kết thúc
Chạy ở chế độ PID mode	M	1					\n
Chạy ở chế độ điều khiển hở	M	0					
Chế độ bộ lọc trung bình FIR	f	Số mẫu					
Chế độ lọc Kalman	k	r	Trọng số sai số hệ thống	q	Trọng số sai số do lường		
Đặt lực đặt	F	Giá trị lực đặt					
Gửi thông số PID	K	P	Giá trị Kp	I	Giá trị Ki	D	Giá trị Kd
Gửi độ rộng xung	D	Giá trị độ rộng xung duty					

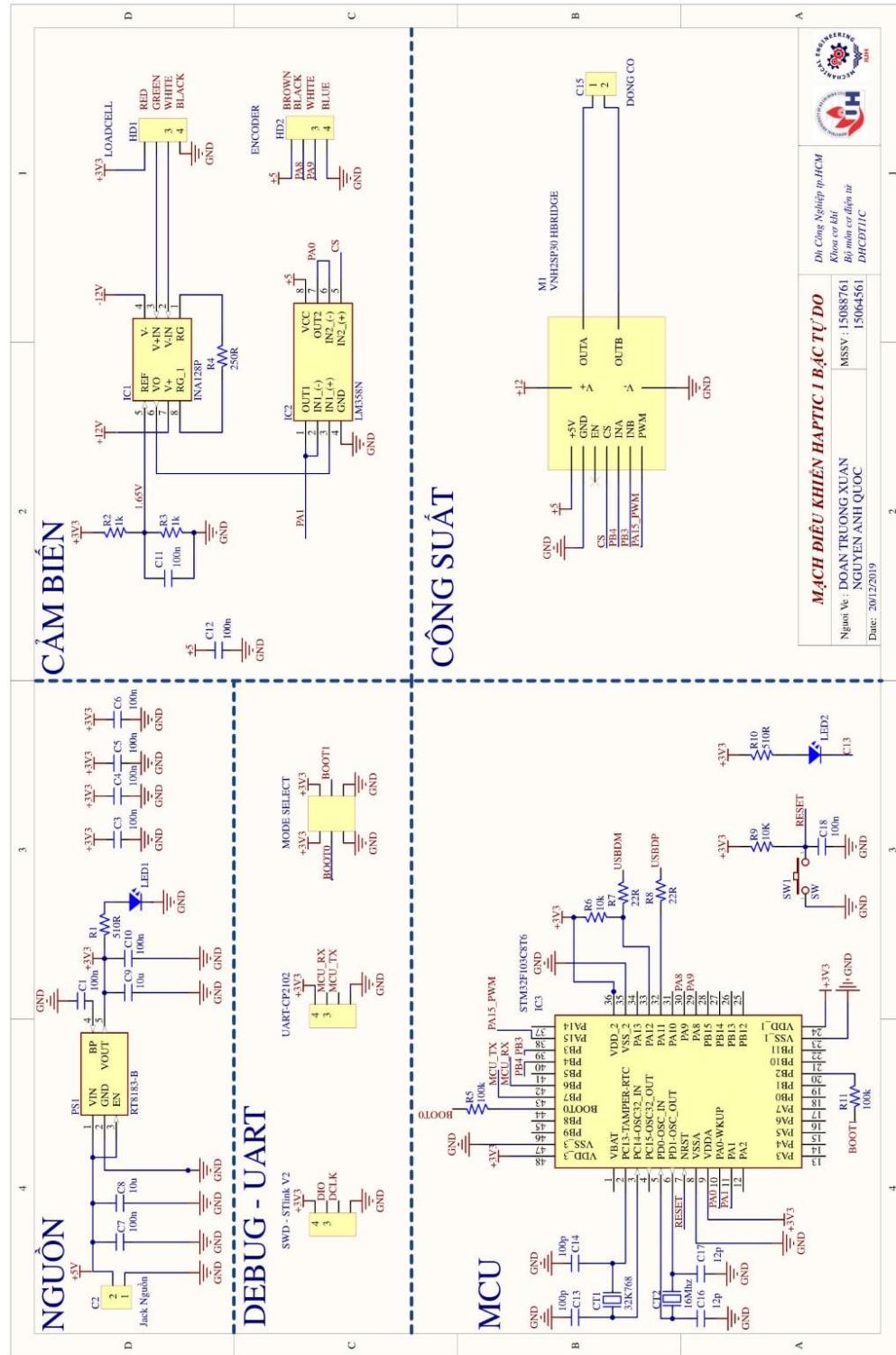
Khung truyền tin từ vi điều khiển

Bảng 5.2: Khung truyền dữ liệu UART từ vi điều khiển đến máy tính

Kí tự bắt đầu	Dữ liệu								Kí tự kết thúc	
(CR) 0x0D	điện áp loadcell chưa lọc	,	Điện áp loadcell đã lọc	,	Khoảng cách	,	Dòng động cơ	,	Độ rộng xung pwm	\n 0x0A

CHƯƠNG 6 THIẾT KẾ VÀ LẮP RÁP MẠCH ĐIỀU KHIỂN

1. Sơ đồ mạch nguyên lý tổng quát



Hình 6.1: Sơ đồ nguyên lý mạch điều khiển haptic

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Khối nguồn: Nguồn 5V được cấp từ nguồn tổ ong bên ngoài vào ic RT8183 có sẵn trong kit stm32f103c8t6-bluepill hạ áp xuống 3v3 để cấp nguồn cho vi điều khiển và loadcell.

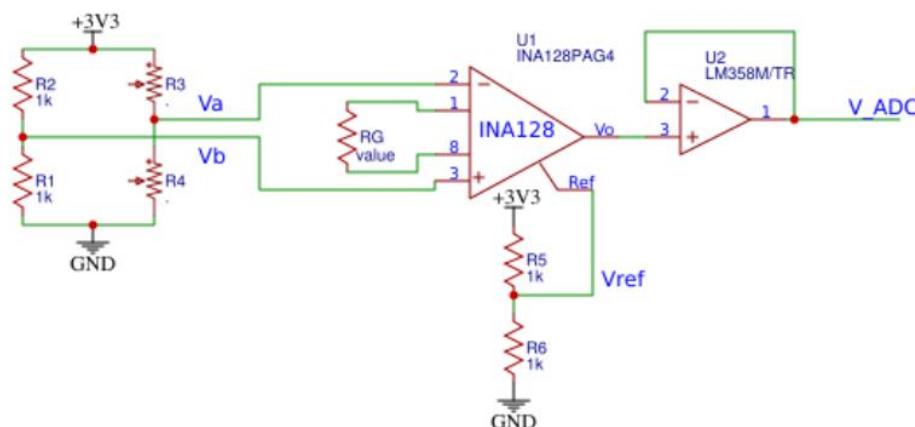
Khối Công suất: sử dụng một mạch cầu H tích hợp VNH2SP30 để điều khiển động cơ

Khối MCU: sử dụng module vi điều khiển stm32f103c8t6, tích hợp thạch anh 8Mhz tạo xung clock ngoài.

Khối debug-uart: gồm các cổng kết nối mạch nạp – debug stlink v2 và mạch giao chuyển đổi giao tiếp uart --usb cp2102.

Khối cảm biến: hai chân tín hiệu encoder gắn thêm hai điện trở treo, tạo mức logic cao là 3.3v. cảm biến loadcell được đưa vào bộ khuếch đại, được tính toán thiết kế như dưới đây.

2. Tính toán thiết kế mạch đọc loadcell hai chiều cho bộ adc của vi điều khiển stm32f103c8t6:



Hình 6.2: Sơ đồ nguyên lý kết nối Loadcell với INA128

Áp dụng mạch khuếch đại INA128 có:

$$v_o = G(v_+ - v_-) + v_{ref} \text{ V}$$

$$G = 1 + \frac{50K}{RG}$$

Đặt $v_{ab} = v_a - v_b$

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Trong khoảng đo -3Kg ÷ +3Kg ta có v_{ab} thay đổi trong khoảng -0.015V ÷ +0.015V
Điện áp tham chiếu bộ ADC vi điều khiển stm32f103c8t6 là 0V ÷ 3.3V. Do đó cần
khuếch đại và chuyển đổi phần điện áp âm thành điện áp dương tương ứng với
khoảng điện áp đo được bằng bộ ADC

Xét

$$v_{omin} = G \cdot v_{abmin} + v_{ref} = G \cdot (-0.015) + V_{ref} = 0V \quad (1)$$

$$v_{omax} = G \cdot v_{abmax} + v_{ref} = G \cdot 0.015 + V_{ref} = 3.3V \quad (2)$$

$$\text{Từ (1),(2)} \Rightarrow \begin{cases} G = 200 \Rightarrow \text{Chọn } R_g = 250 \Omega \\ V_{ref} = 1.65 V \Rightarrow \text{chọn } R5 = R6 = 1K \end{cases}$$

CHƯƠNG 7 ĐO LƯỜNG VÀ GIA CÔNG CÁC GIÁ TRỊ CẢM BIẾN

1. Đo lường lực tác động, dòng tiêu thu động cơ bằng bộ chuyển đổi ADC của vi điều khiển STM32F103C8T6

Để phản hồi lực điều khiển động cơ thì điều kiện tiên quyết đầu tiên là phải đọc được tín hiệu loadcell. Tuy nhiên vì tín hiệu đầu ra của loadcell là tín hiệu analog rất thấp. Vì vậy nó rất dễ bị ảnh hưởng nhiều từ các giá trị nhiễu bên ngoài mặc dù đã được đọc cách ly và lọc nhiễu đồng pha thông qua bộ khuếch đại INA128.

Các nguồn gây nhiễu trong đề tài:

- Nguồn xung: chất lượng của bộ nguồn xung không đảm bảo có thể dẫn đến các dao động điện áp đầu ra gây nhiễu
- Nhiễu do động cơ: tín hiệu điều khiển của động cơ là tín hiệu tần số cao, do đó gây nhiễu. Ngoài ra động cơ thực ra là một tải RL, cuộn dây phần ứng của động cơ sẽ gây ra các gai nhiễu trên nguồn

Nhiễu do chất lượng mạch không đảm bảo: điện áp tham chiếu của bộ adc không được cách ly tốt, không giữ được mức điện áp cố định.

Nhiễu phi tuyến: các loại nhiễu không xác định được nguyên nhân

Các giải pháp lọc nhiễu phần cứng được áp dụng trong đề tài:

- sử dụng các tụ lọc nguồn, các bộ lọc nhiễu thông thấp RC, vì lí do không xác định được tần số nhiễu nên trong đề tài này các tụ lọc thông thấp được chọn là 100nF.
- Tăng trở kháng ngõ vào adc bằng các mạch đệm phản hồi dòng dùng opamp

Các giải pháp lọc nhiễu phần cứng chỉ có thể giải quyết được các loại nhiễu cố định. Để đảm bảo chính xác hơn, cần xây dựng các bộ lọc nhiễu bằng phần mềm.

1.1 Bộ lọc FIR (Finite Impulse Response)

1.1.1 Cơ sở lý thuyết

Dựa trên cơ sở tín hiệu thực tế là hiệu giữa tín hiệu đo tín hiệu đo và tín hiệu nhiễu $S_{real} = S_{measure} + S_{noise}$. Và tín hiệu nhiễu luôn có xu hướng thay đổi ngẫu nhiên xung quanh điểm không $\lim_{x \rightarrow \infty} \sum s_{noise}(x) = 0$. Trên cơ sở đó khi ta cộng trung bình tĩnh hiệu càng lớn thì độ chính xác tín hiệu càng lớn.

Xây dựng bộ lọc FIR cho bộ chuyển đổi ADC:

Bước 1: Thiết lập số lượng lấy mẫu

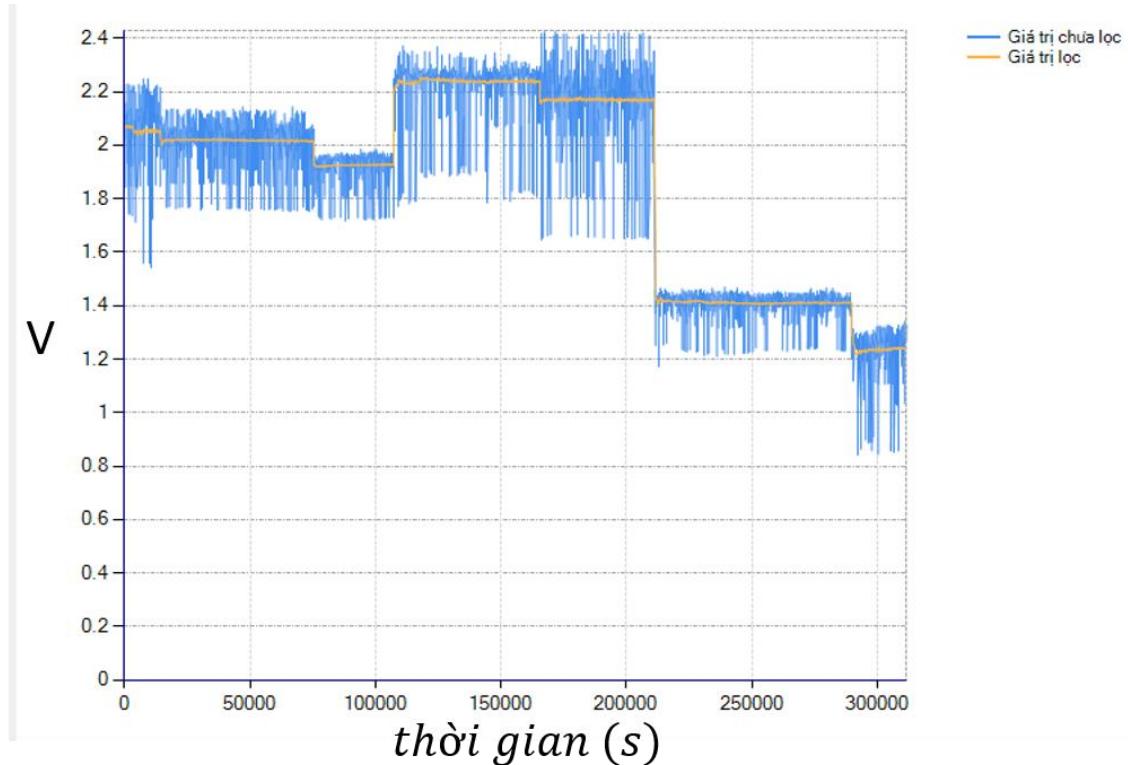
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Bước 2: Đọc tín hiệu từ đối tượng đưa vào tập mẫu

Bước 3: Tính toán trung bình giá trị trên bộ lọc

1.1.2 *Đồ thị bộ lọc FIR*

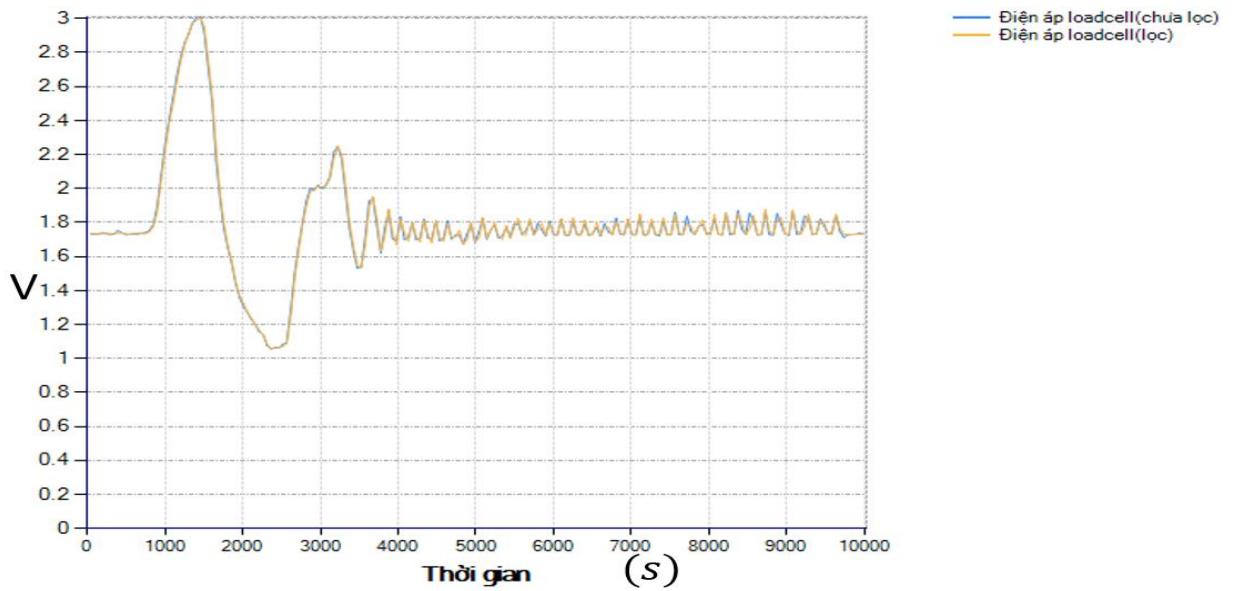
Thí nghiệm 1: Bộ lọc FIR với độ lớn tập mẫu là 1000, tần số lấy mẫu ADC là 50ms, tác dụng lực tay ngẫu nhiên.



Hình 7.1: *Thí nghiệm 1 lọc nhiễu FIR*

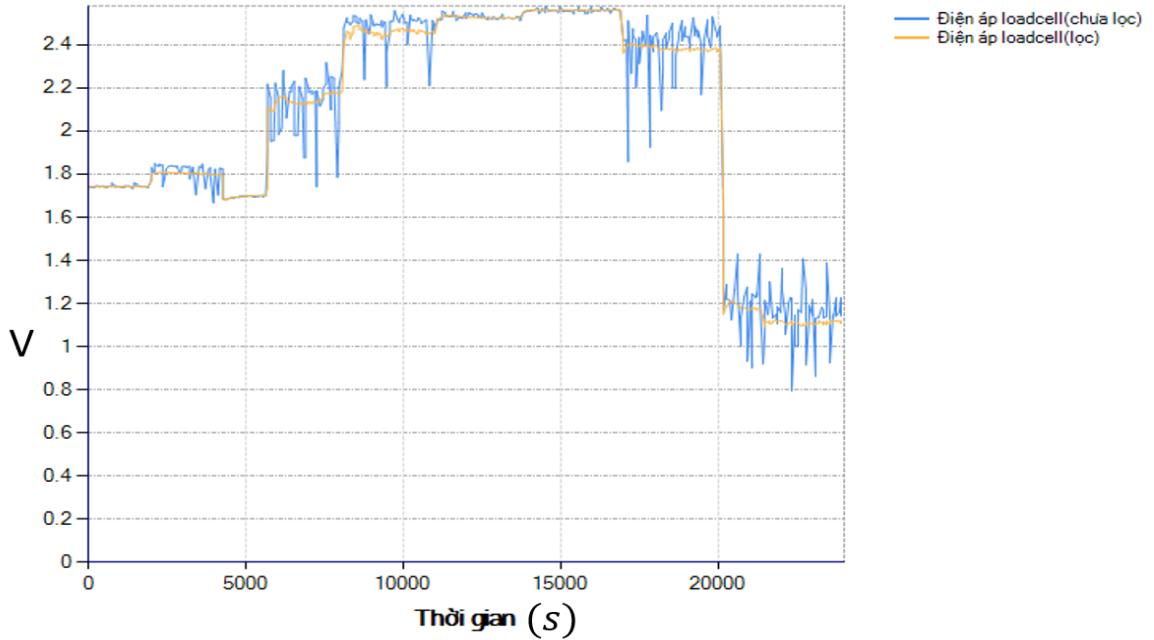
Thí nghiệm 2: Bộ lọc FIR với độ lớn tập mẫu là 100, tần số lấy mẫu ADC là 50ms thay đổi giá trị lực tác động ngẫu nhiên vào cảm biến bằng tay.

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 7.2: Thí nghiệm 2 lọc nhiễu FIR

Thí nghiệm 3: Bộ lọc FIR với độ lớn tần số lấy mẫu ADC là 50ms. Thay đổi giá trị lực tác động vào cảm biến bằng động cơ, bằng cách thay đổi độ rộng xung PWM điều khiển ngẫu nhiên.



Hình 7.3: Đồ thị thí nghiệm 3 bộ lọc FIR

1.1.3 Kết luận

Với bỘ mẪU SỐ là 1000 thì bộ lọc FIR chạy tương đối chính xác giá trị với đại lượng không đổi, tuy nhiên khi đại lượng đo thay đổi độ trễ của thuật toán xuất hiện rõ rệt.

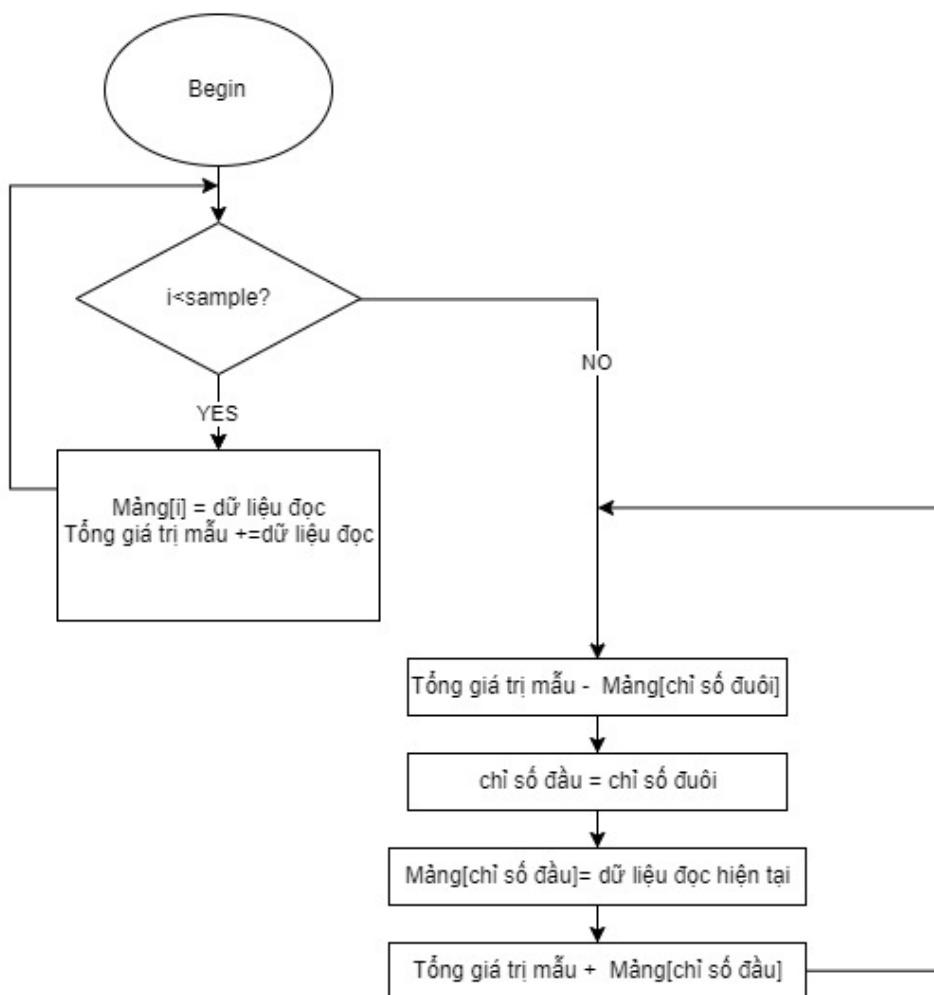
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Với bộ mẫu số 100 khi giá trị đại lượng đo thay đổi đột ngột thì độ chính xác của bộ lọc giảm rõ rệt tuy nhiên độ bám biển của nó lại tốt hơn bộ mẫu số 1000.

1.1.4 Kỹ thuật đọc dữ liệu Circle buffer

Từ kết quả thí nghiệm với bộ lọc FIR, ta nhận thấy với số mẫu càng lớn, độ lọc nhiễu của bộ lọc càng tăng, tuy nhiên khi số mẫu tăng lên cũng đồng nghĩa với số lần đọc giá trị trong chương trình phải thực hiện cũng nhiều lên, làm tăng thời gian thực thi chương trình. Để giải quyết vấn đề này, ta có thể sử dụng kỹ thuật circle buffer trong việc lấy mẫu dữ liệu rồi tính trung bình với số bước ít hơn thay vì sử dụng các vòng lặp để đọc đủ số lượng mẫu rồi mới tính giá trị trung bình.

lưu đồ giải thuật của kỹ thuật circle buffer áp dụng cho bộ lọc FIR sử dụng trong đề tài



Hình 7.4: Lưu đồ giải thuật FIR

1.2 Bộ lọc nhiễu Kalman

1.2.1 Cơ sở lý thuyết

Bộ lọc Kalman, được Rudolf (Rudy) E. Kálmán công bố năm 1960, là thuật toán sử dụng chuỗi các giá trị đo lường, bị ảnh hưởng bởi nhiễu hoặc sai số, để ước đoán biến số nhằm tăng độ chính xác so với việc sử dụng duy nhất một giá trị đo lường. Bộ lọc Kalman thực hiện phương pháp truy hồi đối với chuỗi các giá trị đầu vào bị nhiễu, nhằm tối ưu hóa giá trị ước đoán trạng thái của hệ thống. Với độ tin cậy cao giải thuật Kalman ngày nay được ứng dụng ngày một nhiều không chỉ riêng trong ngành kỹ thuật mà còn được áp dụng trong một số đặc thù trong kinh tế.

Bộ lọc Kalman được sử dụng để ước tính các trạng thái dựa trên các hệ thống phương trình không gian trạng thái tuyến tính. Mô hình dự đoán, ước tính của trạng thái từ thời điểm $k-1$ đến thời điểm k được biểu diễn bằng hệ phương trình như sau:

$$X_k = F_{xk-1} + B_{uk-1} + w_{k-1}$$

trong đó F là ma trận chuyển trạng thái được áp dụng cho vectơ trạng thái trước đó x_{k-1} , B là ma trận đầu vào điều khiển được áp dụng cho vectơ điều khiển u_{k-1} . Và w_{k-1} là vectơ nhiễu quá trình được giả sử là Gaussian trung bình bằng 0 với hiệp phương sai Q , Tức là $w_{k-1} \sim N(0, Q)$.

Mô hình quá trình được ghép nối với mô hình đo lường mô tả mối quan hệ giữa trạng thái và phép đo ở bước thời gian hiện tại k như:

$$z_k = H_k x_k + v_k$$

trong đó z_k là vectơ đo, H là ma trận đo lường, và v_k là vectơ nhiễu đo được giả sử là Gaussian trung bình bằng 0 với hiệp phương sai R , Tức là, $v_k \sim N(0, R)$.

Vai trò của bộ lọc Kalman là cung cấp ước tính của x_k tại thời điểm k , đưa ra ước tính ban đầu của x_0 , loạt phép đo, z_1, z_2, \dots, z_k và thông tin của hệ thống được mô tả bởi F, B, H, Q và R . Lưu ý rằng các chỉ số cho các ma trận này được bỏ qua ở đây bằng cách giả sử rằng chúng bất biến theo thời gian như trong hầu hết các ứng dụng. Mặc dù các ma trận hiệp phương sai được cho là phản ánh số liệu thống kê của các tạp âm, nhưng số liệu thống kê thực sự của các tạp âm không được biết hoặc không có Gaussian trong nhiều ứng dụng thực tế. Do đó, Q và R thường được sử dụng làm

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo thông số điều chỉnh mà người dùng có thể điều chỉnh để có được hiệu suất mong muốn.^[5]

1.2.2 Thuật toán lọc Kalman

Thuật toán lọc Kalman bao gồm hai giai đoạn: dự đoán giá trị và cập nhật trạng thái dự đoán. Thuật toán lọc Kalman được tóm tắt như sau:

Bước dự đoán:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k} + B u_k$$

$$P_{k|k-1} = F P_{k-1|k} F^T + Q$$

Trong đó : $\hat{x}_{k|k-1}$ là trạng thái tiên nghiệm được ước đoán ở thời điểm k;

$\hat{x}_{k-1|k}$ là trạng thái hậu nghiệm được ước đoán ở thời điểm k-1.

Ma trận $P_{k|k-1}$ là ma trận tương quan sai số tiên nghiệm;

Ma trận $P_{k-1|k}$ là ma trận tương quan sai số hậu nghiệm.

Quá trình hiệu chỉnh ước đoán là quá trình dựa vào giá trị đo lường để hiệu chỉnh giá trị ước đoán tiên nghiệm để được kết quả ước đoán hậu nghiệm chính xác. Kết quả này chính là giá trị ước lượng đầu ra của bộ lọc Kalman. Quá trình hiệu chỉnh ước đoán được thực hiện như sau:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + k_k * (z_k - H * \hat{x}_{k|k-1})$$

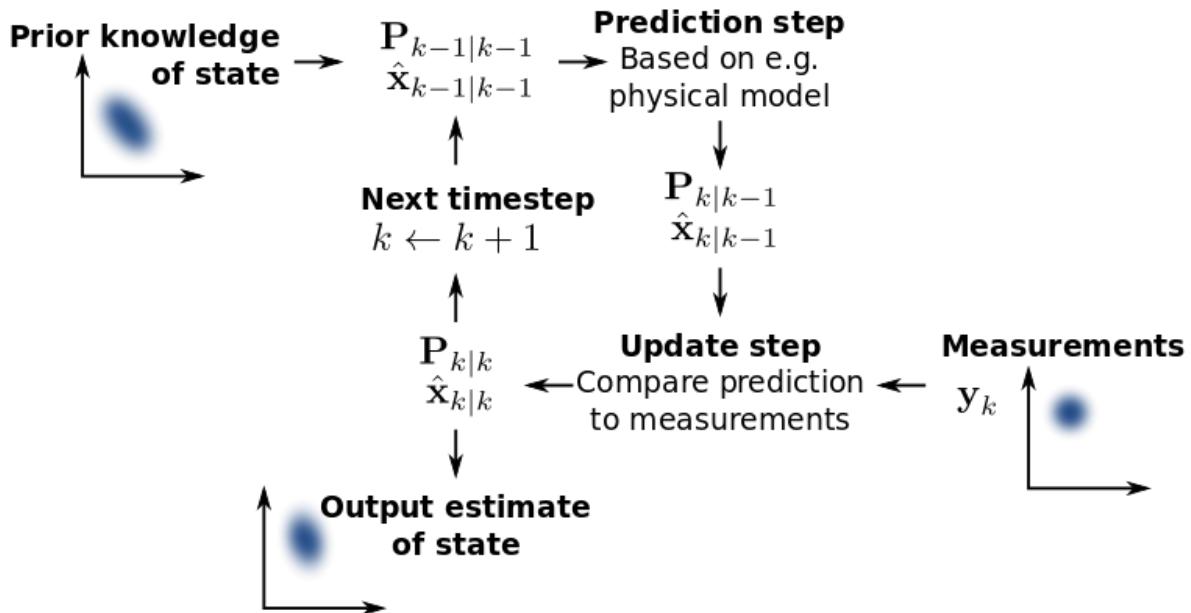
$$K_k = P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1}$$

$$P_{k|k} = (I - K_k H) P_{k|k-1}$$

Trong đó: K_k là độ lợi của mạch lọc Kalman, giá trị k_k thay đổi theo mỗi ước đoán k và được chọn sao cho tương quan sai số ước lượng hậu nghiệm là nhỏ nhất; I là ma trận đơn vị; $\hat{x}_{k|k}$ là trạng thái hậu nghiệm được ước đoán ở thời điểm thứ k. Đây chính là giá trị đầu ra của bộ lọc Kalman.^[6]

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

1.2.3 Lưu đồ giải thuật bộ lọc Kalman



Hình 7.5: Lưu giải thuật bộ lọc Kalman

1.2.4 Áp dụng bộ lọc Kalman cho bộ chuyển đổi ADC:

Phương trình trạng thái rời rạc mô tả tín hiệu đo adc được xác định như sau

$$X_k = F_{xk-1} + B_{uk-1} + w_{k-1}$$

$$z_k = H_k x_k + v_k$$

Phương trình tiên đoán giá trị:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k * (z_k - H * \hat{x}_{k|k-1})$$

$$K_k = P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1}$$

$$P_{k|k} = (I - K_k H) P_{k|k-1}$$

Trong đó:

X_k là ma trận giá trị đã lọc

F là ma trận chuyển đổi trạng thái của hệ thống

B là ma trận tác động của tín hiệu điều khiển

H là ma trận quan sát vi sai giữa trạng thái hiện tại và trạng thái trước.

u_k là vector điều khiển

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Z_k là ma trận giá trị đo lường (chưa lọc)

W_k là ma trận nhiễu hệ thống

V_k là ma trận nhiễu đo lường

K_k : là độ lợi của bộ lọc kalman

I là ma trận đơn vị

Đối với đối tượng cảm biến loadcell, hệ là hệ một đầu vào – một đầu ra, do đó ma trận $F = [1]$, $H = [1]$. tác động nhiễu do tín hiệu điều khiển có thể bỏ qua (do có cách ly điều khiển) nên $B=0$, $u=0$.

Từ đó ta lập được các bước thực hiện với bộ lọc Kalman với bài toán đọc giá trị loadcell như sau:

Bước 1: Trọng số sai số đo lường và sai số hệ thống được nhập từ máy.

Bước 2: Đọc giá trị điện từ loadcell

Bước 3: Tính hệ số Kalman với công thức sau:

$$K_{\text{gain}} = e_{\text{predict}} / (e_{\text{predict}} + e_{\text{measument}});$$

Bước 4: Tính giá trị loadcell.

$$V_k = V_{k-1} + K_{\text{gain}} \cdot (V_{\text{in}} - V_{k-1});$$

Bước 5: Cập nhật sai số dự đoán.

$$e_{\text{predict}} = (1 - K_{\text{gain}}) \cdot e_{\text{predict}} + |(V_{k-1} - V_k) \cdot e_{\text{system}}|$$

1.2.5 Thí nghiệm khảo sát bộ lọc Kalman

Mục đích thí nghiệm: Khảo sát ảnh hưởng của các trọng số trong bộ lọc Kalman đến kết quả đo ADC cho loadcell.

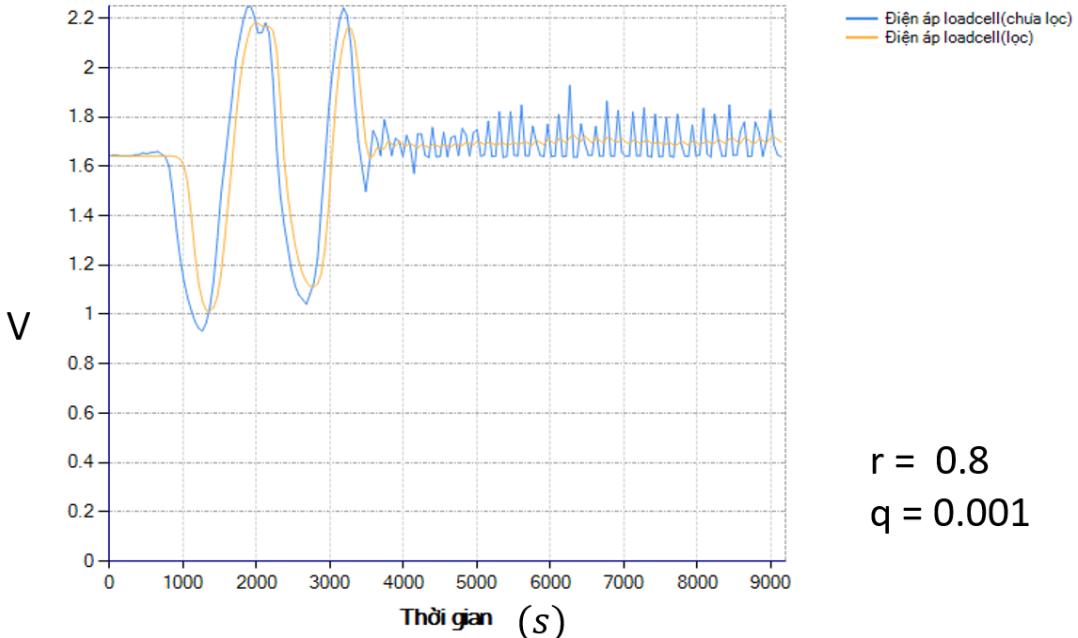
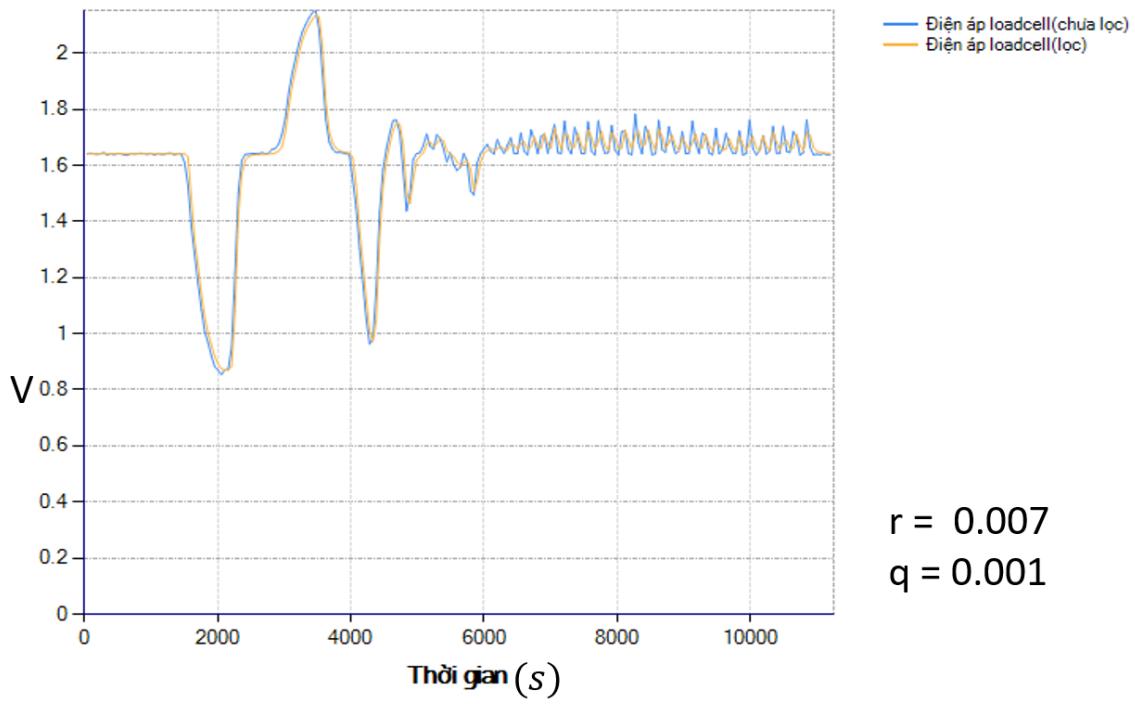
Bố trí thí nghiệm:

Áp dụng bộ lọc Kalman đọc điện áp loadcell với chu kỳ lấy mẫu ADC là 50ms, thay đổi các giá trị trọng số để xem xét ảnh hưởng của các trọng số tác động lên đầu ra bộ lọc.

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

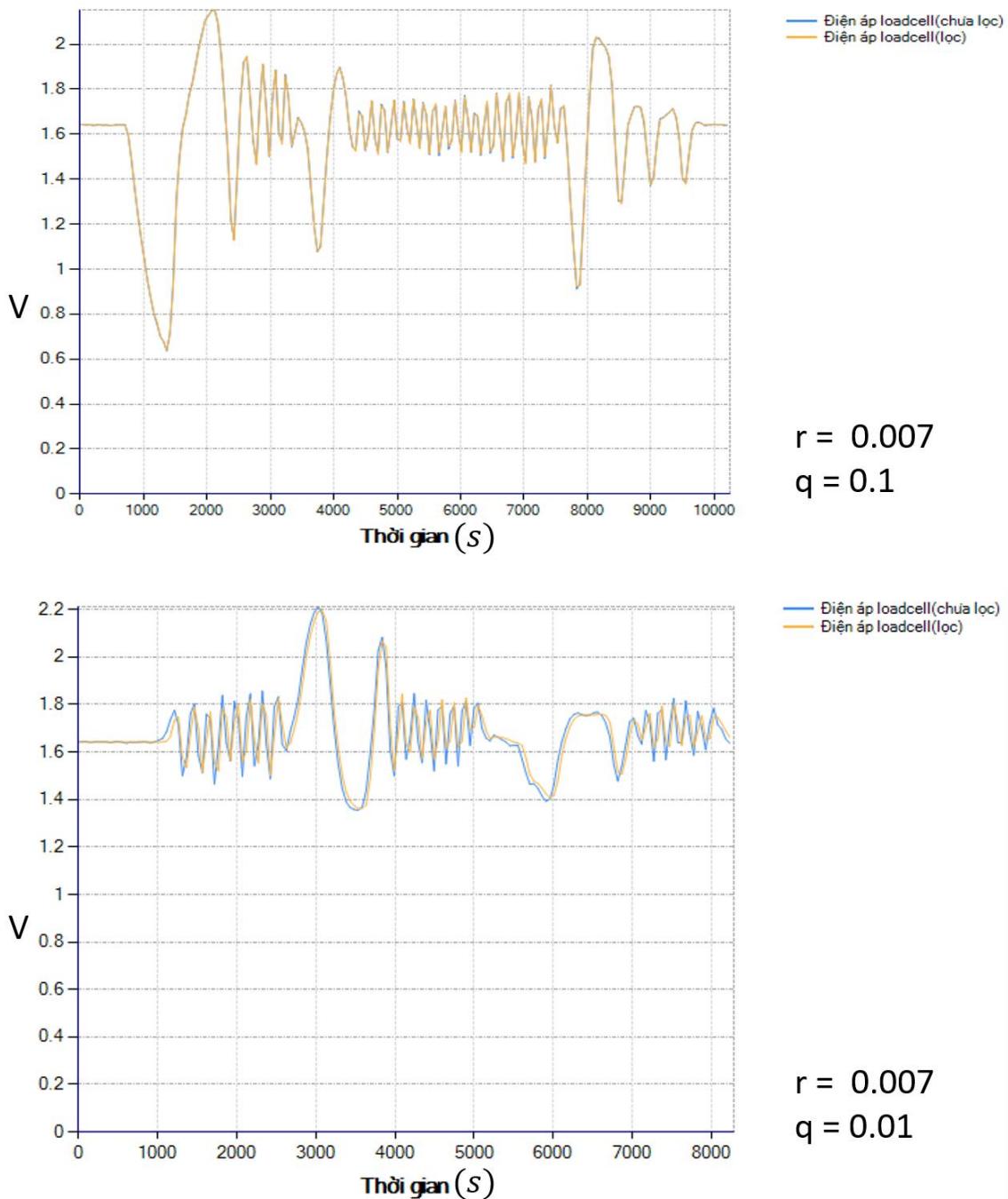
Thí nghiệm 1: dùng tay tác động vào cảm biến loadcell với các lực ngẫu nhiên và tần số thay đổi tăng dần.

Ảnh hưởng trọng số sai số hệ thống trong thí nghiệm 1



Ảnh hưởng trọng số sai số đo lường trong thí nghiệm 1

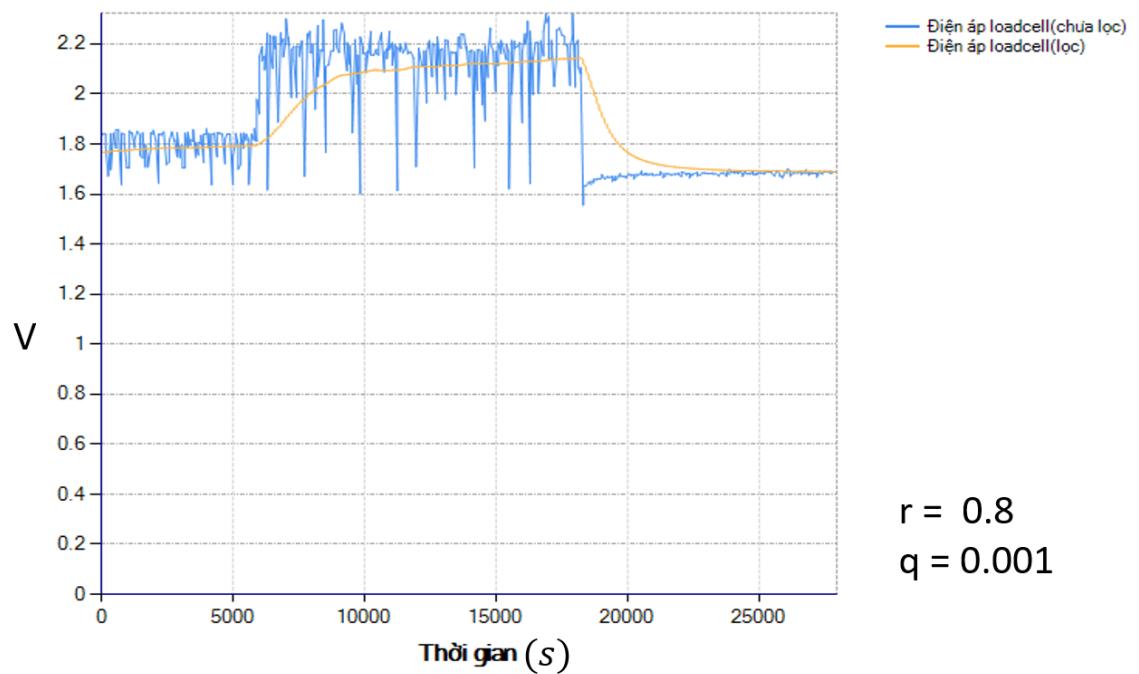
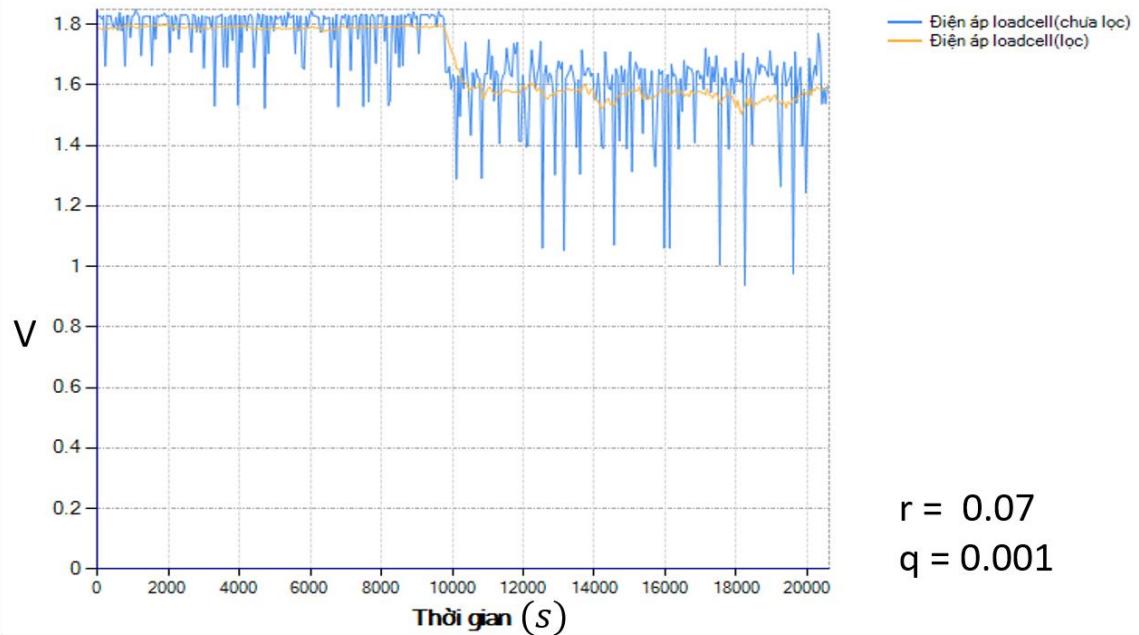
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Thí nghiệm 2: dùng động cơ tác động vào cảm biến loadcell với các giá trị độ rộng xung PWM thay đổi ngẫu nhiên.

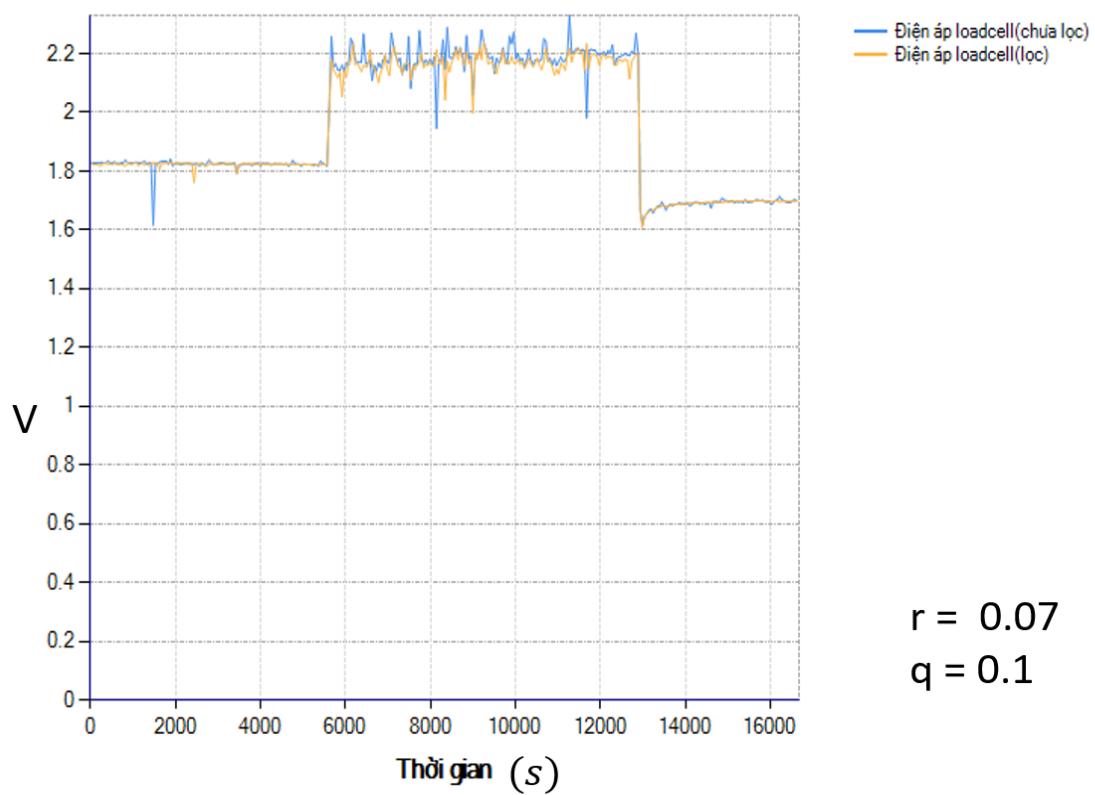
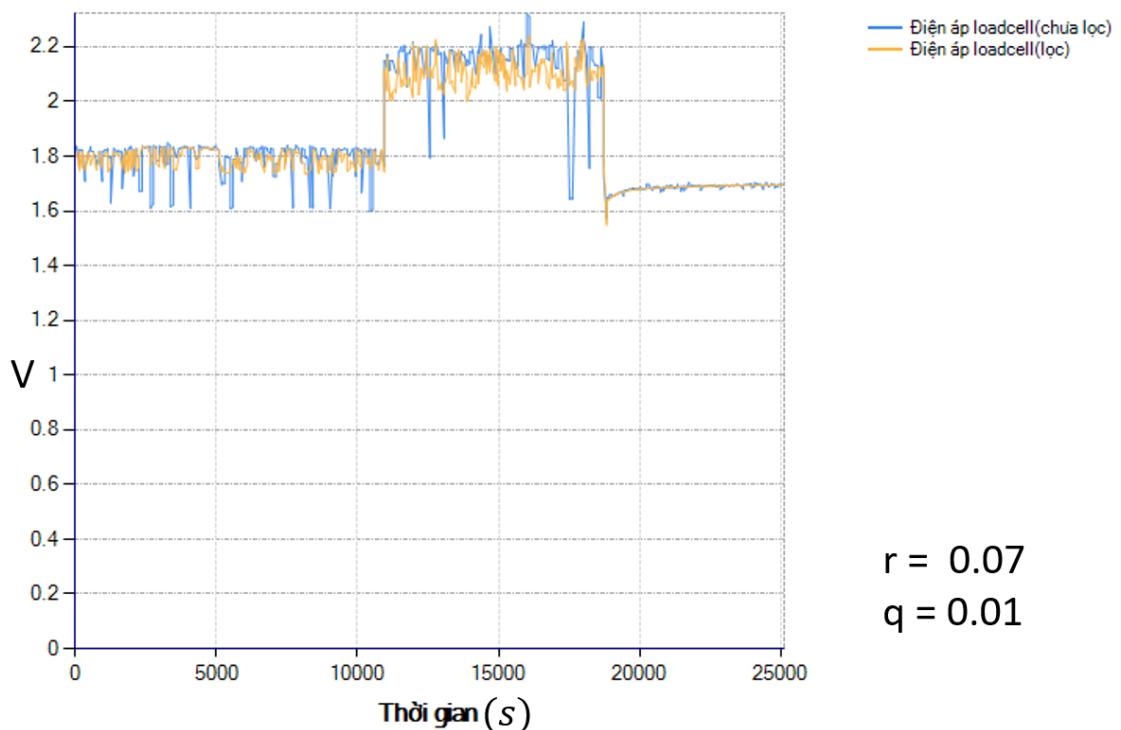
Ảnh hưởng trọng số sai số đo lường trong thí nghiệm 2

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Ảnh hưởng trọng số sai số hệ thống trong thí nghiệm 2

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



.Kết luận

Trọng số sai số đo lường càng lớn thì giá trị đầu ra của bộ lọc càng nhỏ so với giá trị đầu vào, đồng thời đáp ứng của đầu ra càng trễ so với độ đáp ứng của đầu vào (độ

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo thay đổi giữa các giá trị đầu ra với giá trị đầu vào nhỏ). Trọng số sai số hệ thống càng nhỏ giá trị đầu ra của bộ lọc càng nhỏ so với giá trị đầu vào. Đáp ứng không thay đổi đáng kể.

1.3 So sánh bộ lọc FIR với bộ lọc Kalman

Dựa vào các thông số của các đồ thị đã được thí nghiệm chúng ta dễ dàng nhận thấy nếu chọn đúng các thông số thì bộ lọc thuật toán Kalman hoàn toàn tốt hơn FIR. Từ biên độ đáp ứng đến độ bám biên đều nổi trội ở bộ lọc FIR.

2. Thí nghiệm calib Loadcell

Mục đích thí nghiệm: tìm ra đường đặc tuyến Lực – điện áp loadcell, từ đó có thể sử dụng loadcell đo lường lực bằng cách chuyển đổi giá trị điện áp thay đổi của loadcell theo đường đặc tuyến đã tìm ra trước đó.

Do nhóm không có điều kiện có được các mẫu trọng lượng chuẩn hoặc thiết bị đo lường chính xác nên nhóm sử dụng một cân điện tử có sẵn trên thị trường để làm chuẩn calib

Thông số cân làm chuẩn trong thí nghiệm: (thông tin chi tiết xem phụ lục)

Tên: Weiheng A11 Mini Retail Digital Weight Scale 25kg Mini Scale

Khoảng cân: -25Kg÷25 Kg

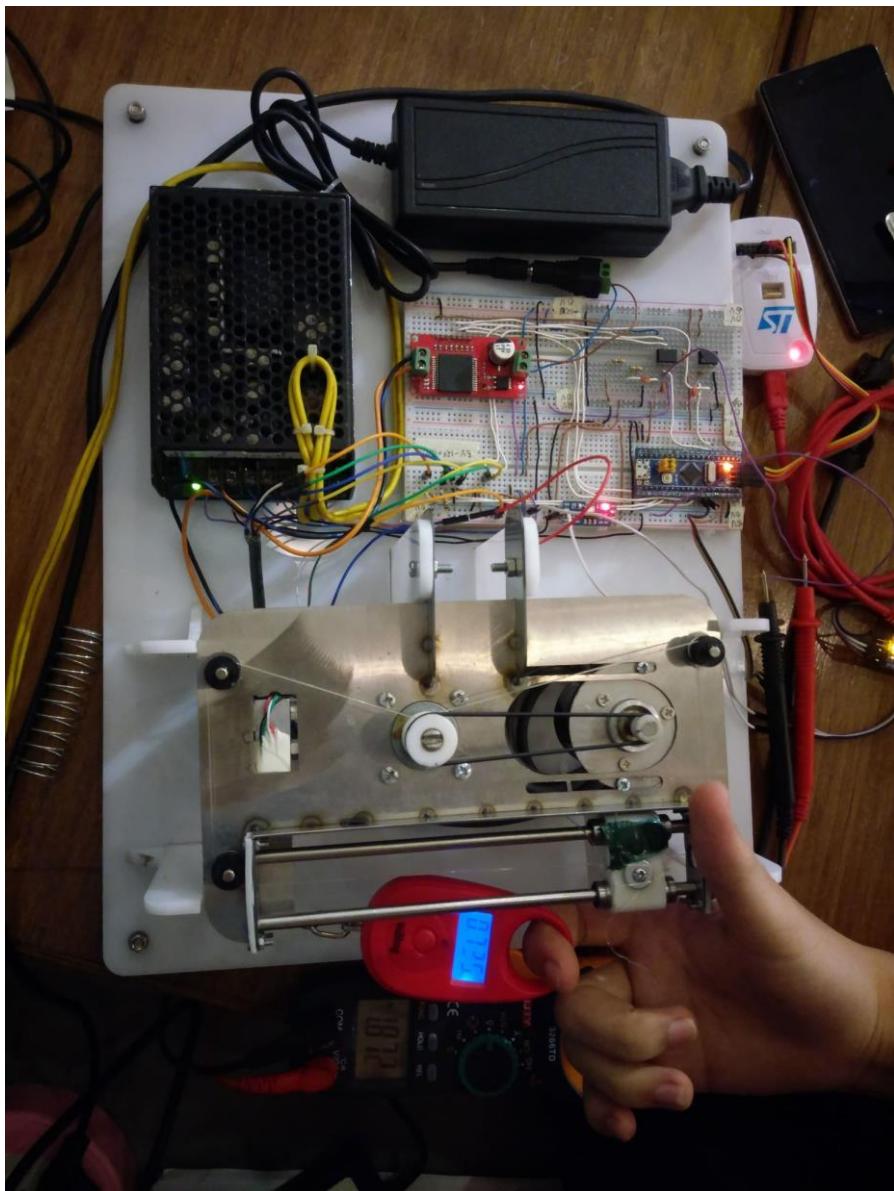
sai số: +-0.5% (5g)

Thí nghiệm 1 calib giá trị loadcell theo cân:

Dùng cân điện tử có sẵn trên thị trường. Tiến hành kéo và lấy mẫu như thí nghiệm sau:

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

.Sau khi thí nghiệm ta có được bảng giá trị như sau:



Hình 7.6: Bố trí mô hình thí nghiệm loadcell

Bảng 7.1: Giá trị calib loadcell

STT	Giá trị đo loadcell (V)	Giá trị lực kg.f
1	1.0588	-1.380
2	0.4865	-2.730
3	2.5714	2.190
4	0.6522	-2.340
5	1.3846	-0.615
6	0.4286	-2.875

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

7	2.1429	1.180
8	0.7368	-2.140
9	0.2308	-3.335
10	0.2857	-3.205
11	2.5714	2.190
12	2.4286	1.850
13	0.2353	-3.325
14	0.3478	-3.055
15	2.4286	1.851
16	1.3529	-0.685
17	0.9412	1.658
18	1.1765	-2.770
19	2.0769	1.020
20	2.1429	1.175

Từ các bảng giá trị trên ta có thể tính được phương trình sau:

$$F(V_{\text{Loadcell}}) = V_{\text{Loadcell}} * 2.3595 - 3.8796 \text{ kg.f}$$

CHƯƠNG 8 KHẢO SÁT ĐẶC TÍNH TẢI ĐỘNG CƠ

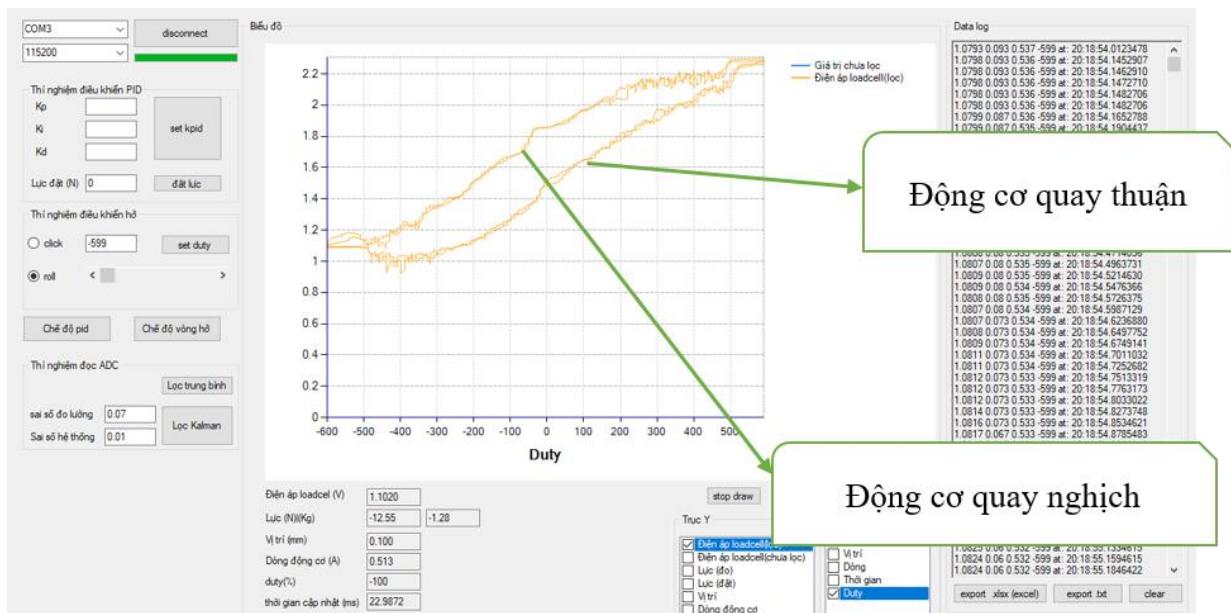
1. Thí nghiệm: khảo sát hệ hở điều khiển động cơ

Mục đích thí nghiệm: khảo sát đường đặc tính duty – điện áp loadcell, duty-lực đo được, điện áp loadcell - lực

Sử dụng phương pháp điều chế xung pwm điều khiển động cơ

bố trí thí nghiệm như sau: Dùng bu lông đai ốc nối cứng khớp ngón trỏ với loadcell. Sử dụng giao diện đo lường và điều khiển đã thiết kế trên máy tính để thay đổi giá trị duty theo hai chiều, từ thấp nhất đến cao nhất rồi ngược lại. Quan sát biểu đồ..

Kết quả thu được:

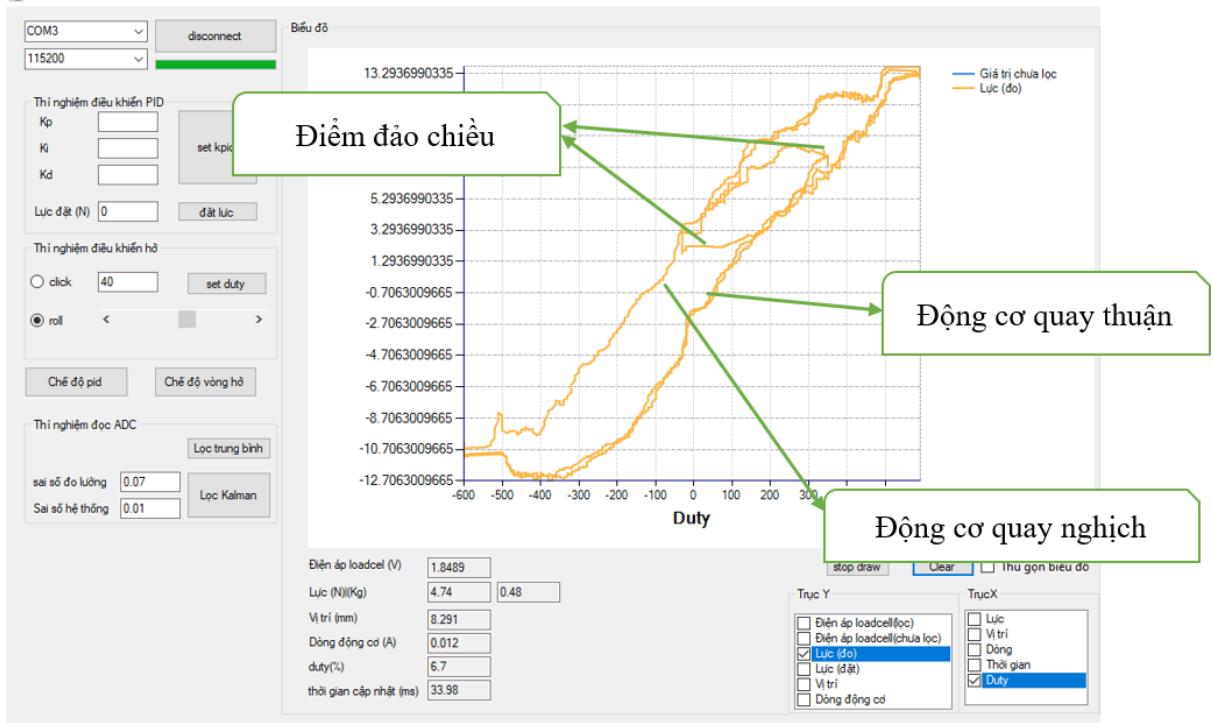


Hình 8.1: Biểu đồ động cơ quay thuận nghịch 1a

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 8.3: Đồ thị động cơ quay thuận nghịch thí nghiệm 1b



Hình 8.2: Đồ thị động cơ quay thuận nghịch 1c

2. Nhận xét:

từ hai biểu đồ trên ta có thể thấy rằng đường đáp ứng của loadcell khi thay đổi chiều của động cơ là khác nhau. Các đường đặc tính trong các lần thử nghiệm sau không hoàn toàn trùng lên đường đặc tính của lần thử nghiệm trước. Đường đặc tính duty-lực đo tuy nhiên trong khoảng giá trị lực từ -11N đến 12N, có nhiều phi tuyến ở các đầu cực đại và cực tiểu.

3. Kết luận:

Như vậy khi điều khiển động cơ sinh lực, khi đảo chiều động cơ sự thay đổi về lực sẽ có một độ trễ do có khoảng thay đổi của duty nhảy từ đường đặc tính này sang đường đặc tính kia. Do vậy khi không có bộ điều khiển phải xác định chiều quay để dùng đúng phương trình chuyển đổi giá trị điện áp loadcell – lực. Còn khi sử dụng các bộ điều khiển phải xem xét đến độ trễ này

4. Thuật toán Linear Regression

Với đường đồ thị phản hồi lực của động cơ theo duty là đường phi tuyến. Để dễ dàng thực hiện các khâu tính toán ta cần phải tuyến hóa phản hồi lực của động cơ theo duty.

4.1 Giới thiệu đôi nét về thuật toán Linear Regression

Thuật toán Linear Regression(LR) một thuật toán cơ bản trong Machine Learning. LR được phân loại vào nhóm thuật toán Supervised Learning, trong đó giá trị của output là một hàm tuyến tính của các biến trong input. Trong thực tế LR là một trong những thuật toán cơ bản và được ứng dụng nhiều trong các lĩnh vực đời sống và thực tiễn.

Để xét LR cho bài toán duty-lực hồi tiếp, ta ứng dụng thuật toán LR dành cho một biến (1 giá trị đầu vào là lực hồi tiếp – 1 giá trị đầu ra là duty).^[7]

4.2 LR một biến

Ta có phương trình đường thẳng tổng quát với bài toán như sau:

$$\hat{F}(x) = b_1 \cdot x + b_0 \quad (1)$$

Trong đó:

$\hat{F}(x)$: Hàm số biến thiên của lực theo duty

b_0, b_1 : Là bộ hệ số mà chúng ta cần tìm sao cho các sai số của các điểm dự đoán của LR so với thực giá trị thực tế là nhỏ nhất.

Với bài toán RL một biến ta có công thức như sau:

$$b_1 = \frac{\sum_{i=1}^n x_i \cdot y_i}{\sum_{i=1}^n x_i^2} = \frac{\bar{xy}}{\bar{x^2}} \quad (2)$$

Với x_i tương ứng với giá trị của lực theo thời gian y_i .^[8]

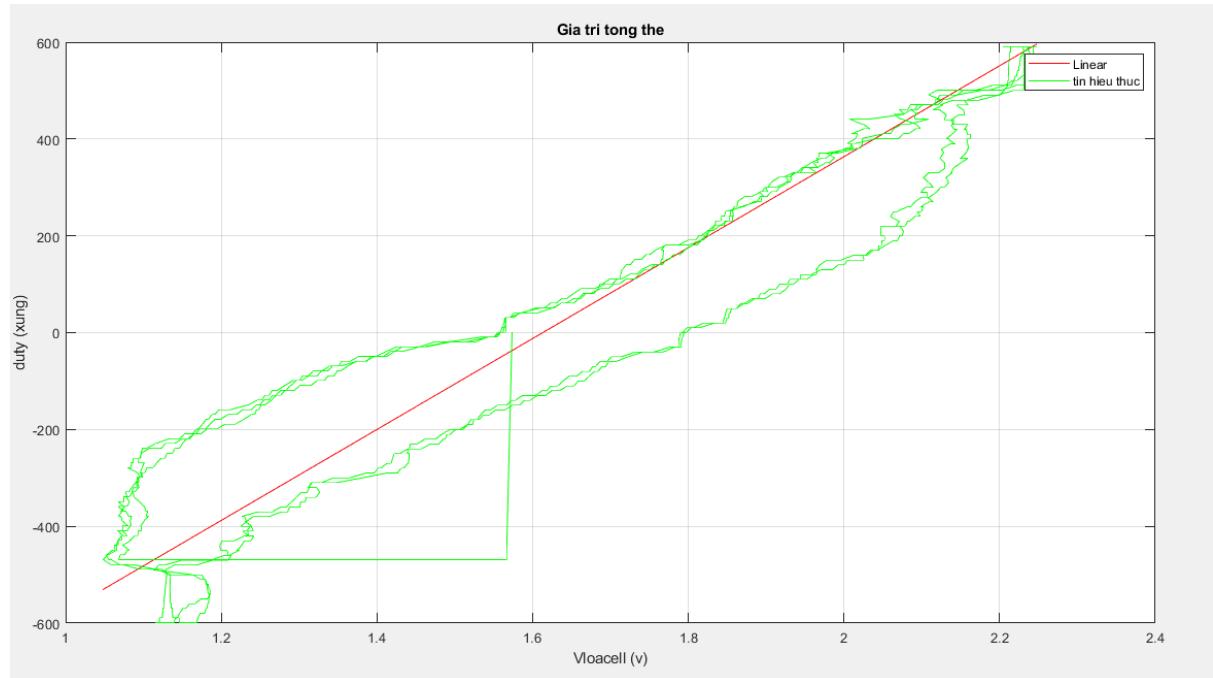
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Từ công thức (1), (2) tao có thể tính được b_0 như sau:

$$b_0 = \hat{F}(x) - b_1 = y(x) - \frac{\bar{xy}}{\bar{x}^2} [9]$$

Từ đó thị đường động cơ ta có các kết quả như sau:

Phương trình động cơ thuận nghịch có dạng:



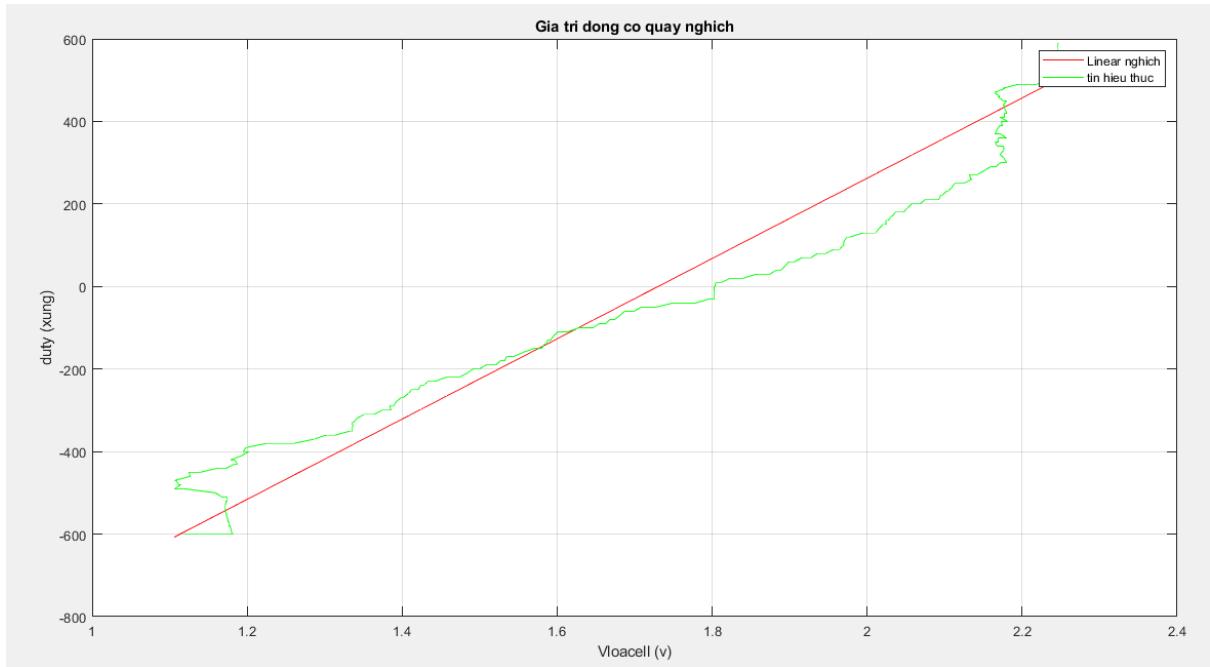
Hình 8.4: Đồ thị linear của động cơ quay thuận nghịch

$$y = 938.1x - 1514.8$$

Phương trình động cơ quay nghịch có dạng:

$$y = 972.7x - 1683.1$$

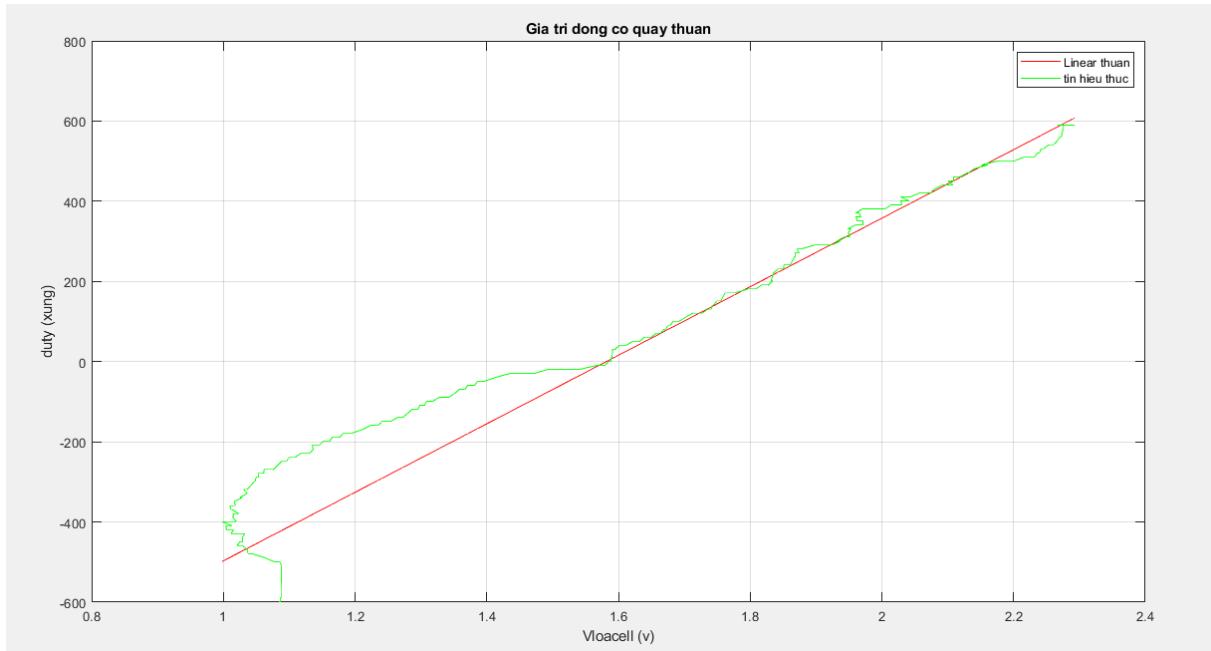
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 8.5: Đồ thị linear động cơ quay nghịch

Phương trình động cơ quay thuận:

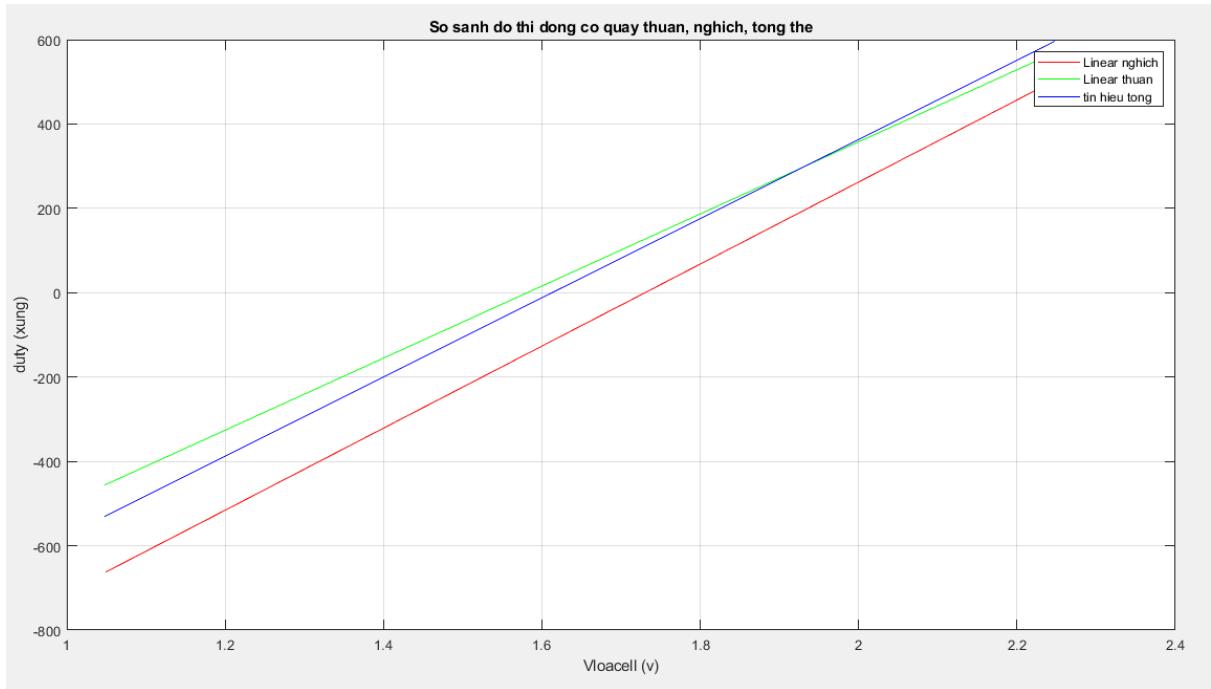
$$y = 854.6x - 1351.6$$



Hình 8.6: Đồ thị linear động cơ quay thuận

Đồ thị so sánh giữa đường tuyến tính quy hồi một biến như sau:

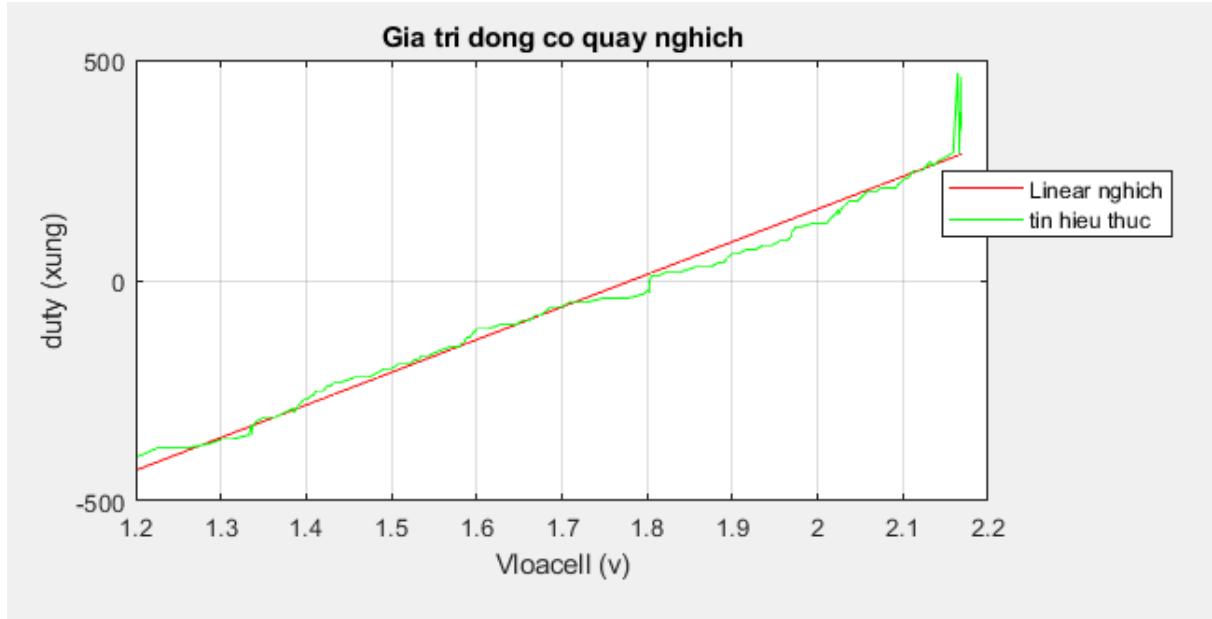
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 8.7: Đồ thị so sá̂o ba đường linear

Ta khảo sát lại hệ động cơ trên miền tuyến tính có được kết quả sau:

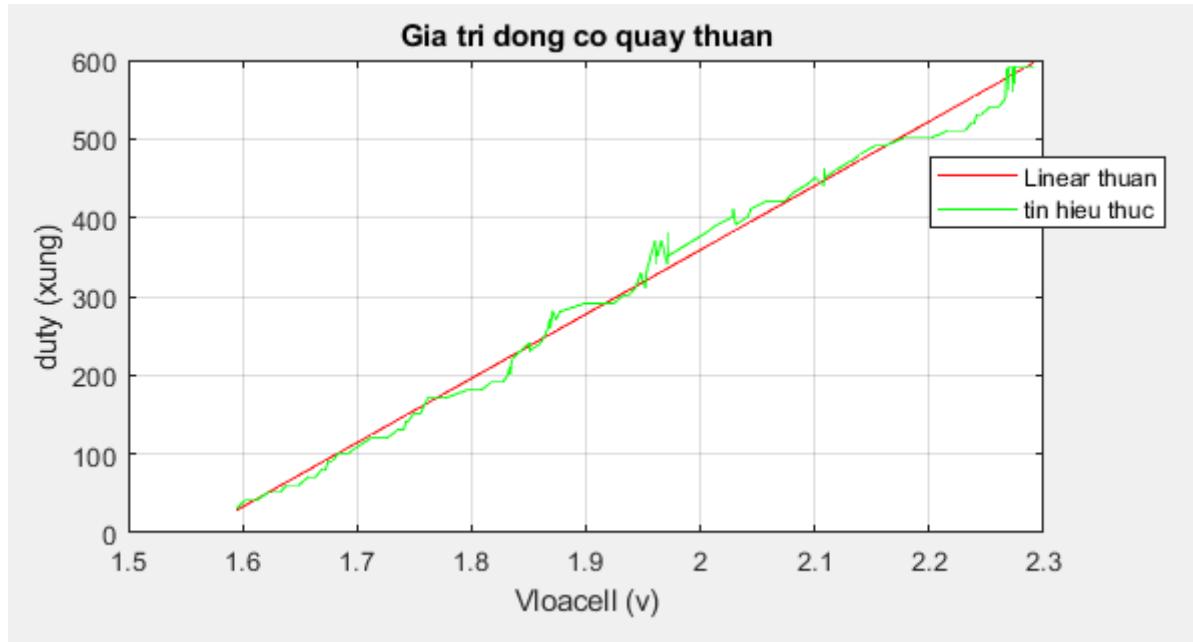
Xung tuyến tính với điện áp loadcell từ khoảng điện áp loadcell 1.2 đến 2.18 v với động cơ quay nghịch:



Hình 8.8: Đường linear sau khi bỏ giá trị phi tuyến khi động cơ quay nghịch

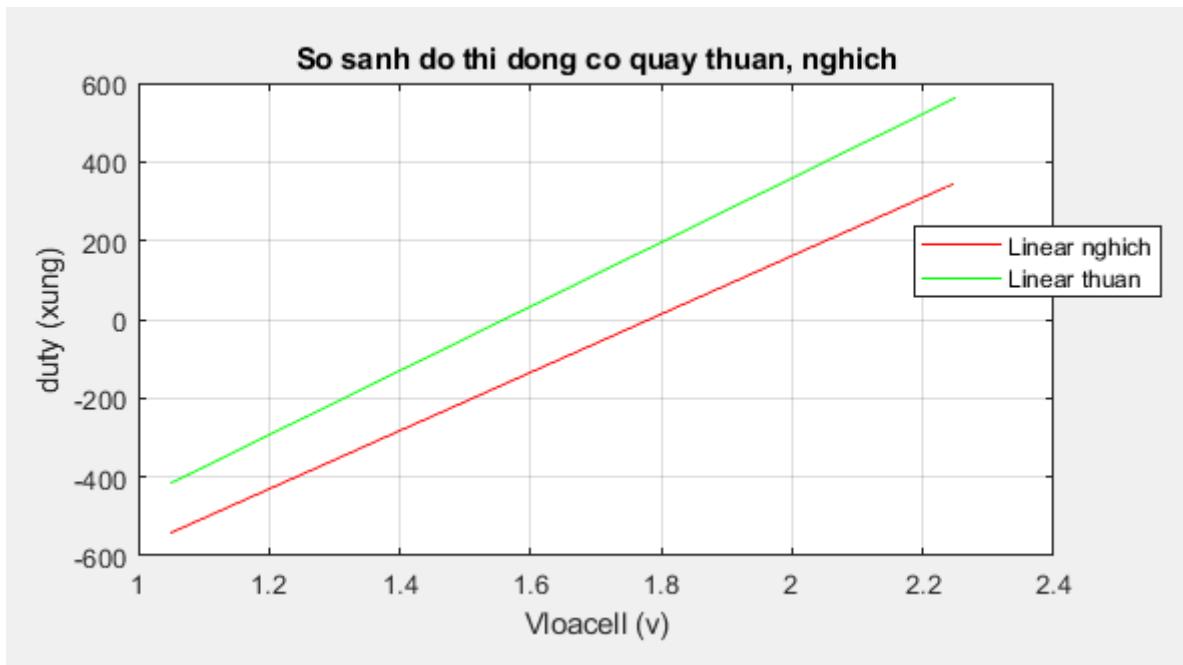
$$y = 740.3x - 1318.8$$

Xung tuyển tính với điện áp loadcell từ khoảng điện áp loadcell 1.6 đến 2.4 v với động cơ quay thuận:



Hình 8.9: Đồ thị linear động cơ quay thuận sau khi lọc

$$y = 815.4x - 1271.7$$



Hình 8.10: So sánh đường động cơ quay thuận và nghịch sau khi lọc

So sánh giữa hai đồ thị lúc động cơ quay thuận và quay nghịch:Nhận xét: Ta thấy rằng đường đặc tính động cơ của sau khi loại bỏ cái giá trị phi tuyến là gần như song song.

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

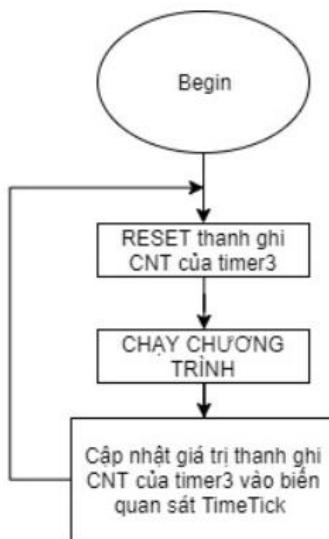
CHƯƠNG 9 THÍ NGHIỆM TÍNH THỜI GIAN THỰC THI CỦA CHƯƠNG TRÌNH

1. Mục đích thí nghiệm:

Vì trong chương trình ta điều khiển động cơ bằng thuật toán PID nên ta cần phải tính toán thời gian thực hiện trong một chu kỳ điều khiển của vi điều khiển.

2. Bố trí thí nghiệm

Sử dụng timer3 đếm thời gian thực thi chương trình nhằm tìm ra chu kỳ lấy mẫu nhỏ nhất cho các bộ điều khiển có trong chương trình điều khiển:



Hình 9.1: Lưu đồ giải thuật tính toán thời gian chương trình

3. Tiến hành thí nghiệm:

Chọn Tân số counter timer3: $f=\text{prescaller}/\text{PBH1clock}=1/36^6=0.028\mu\text{s}$

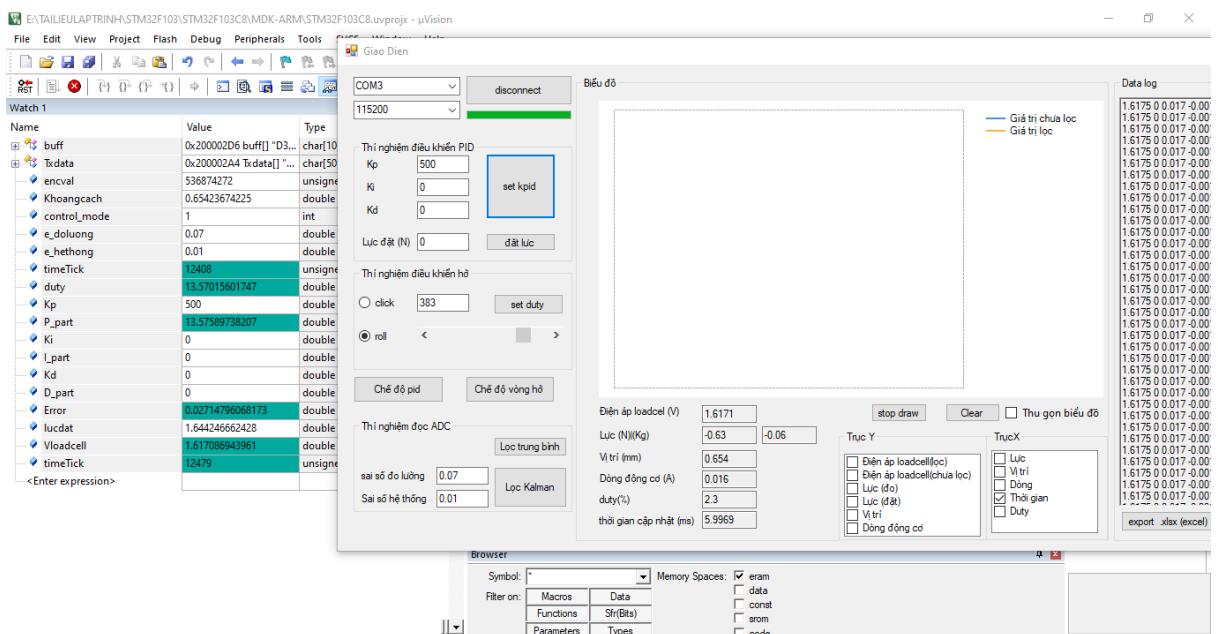
Cài đặt giá trị tối đa của thanh ghi CNT: period: 65535

$$\Rightarrow \text{Thời gian} = \text{CNT}/36^6 = \text{CNT} * 0.028 (\mu\text{s})$$

Sử dụng mạch nạp stlink v2, kết nối vi điều khiển với máy tính, dùng phần mềm keilc ở chế độ debug. Chạy hết tất cả các tính năng của chương trình đã viết, quan sát giá trị thanh ghi CNT của timer3 được cập nhật mỗi vòng lặp chương trình chính.

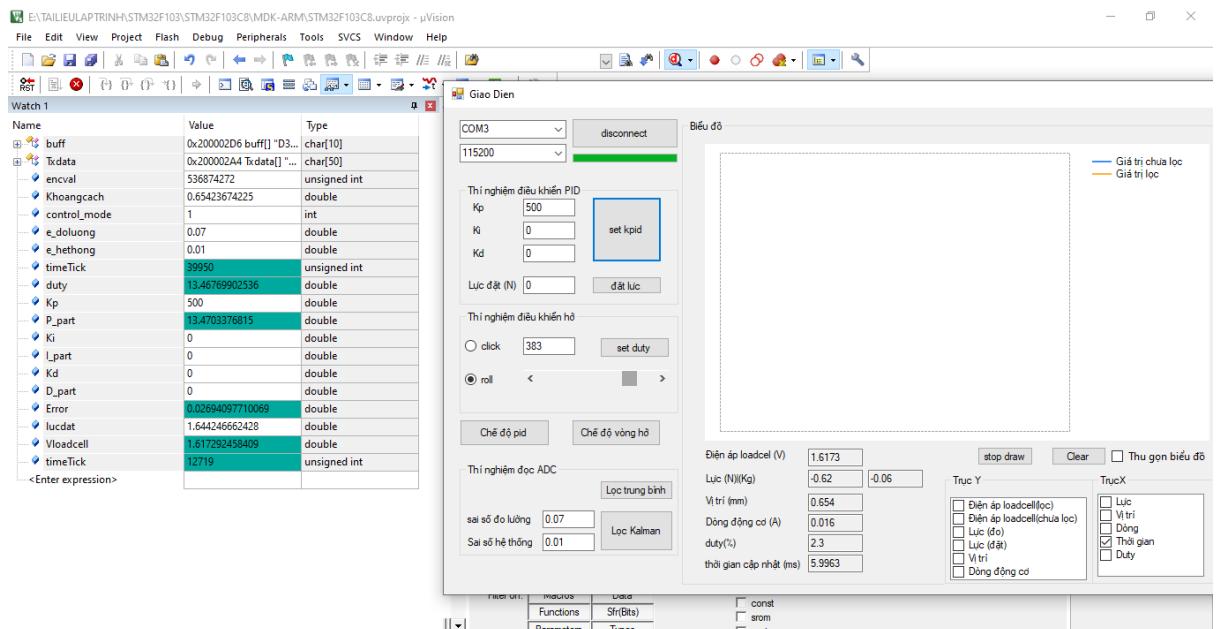
Kết quả: Quan sát khi kết nối với giao diện máy tính, không điều khiển động cơ, thời gian thực thi chương trình ổn định.

Ché tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 9.2: Thời gian chạy không điều khiển động cơ

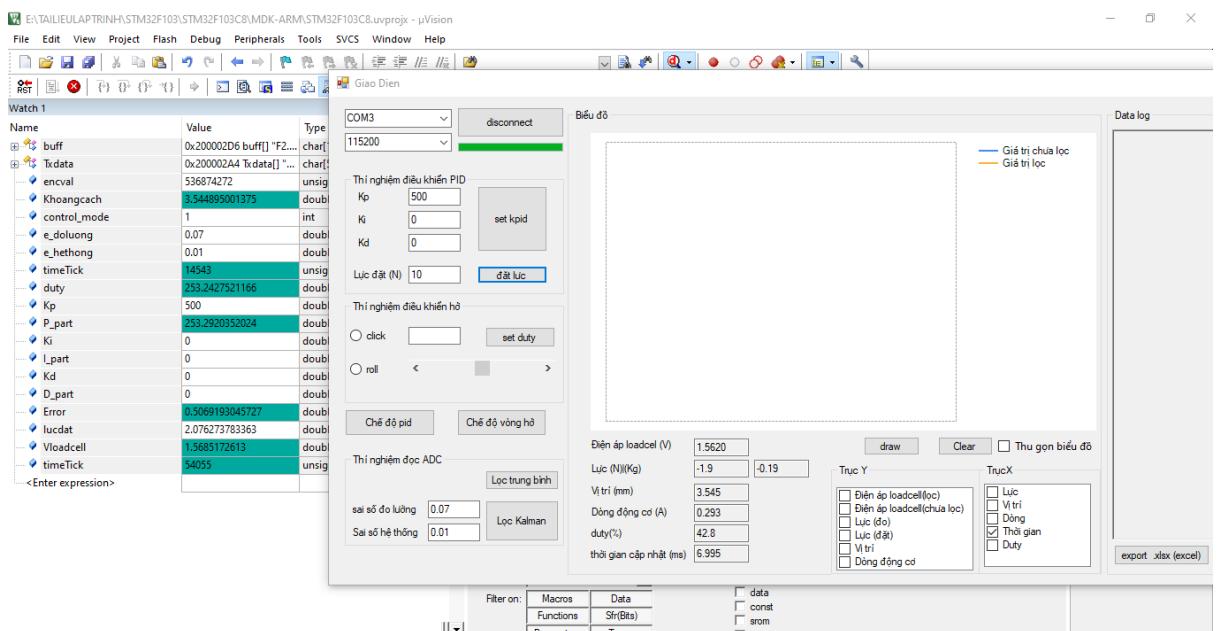
Ở Chế độ điều khiển hở, thay đổi giá trị độ rộng xung liên tục, thời gian thực thi chương trình có lúc tăng vọt.



Hình 9.3: Thời gian chạy ở chế độ điều khiển vòng hở

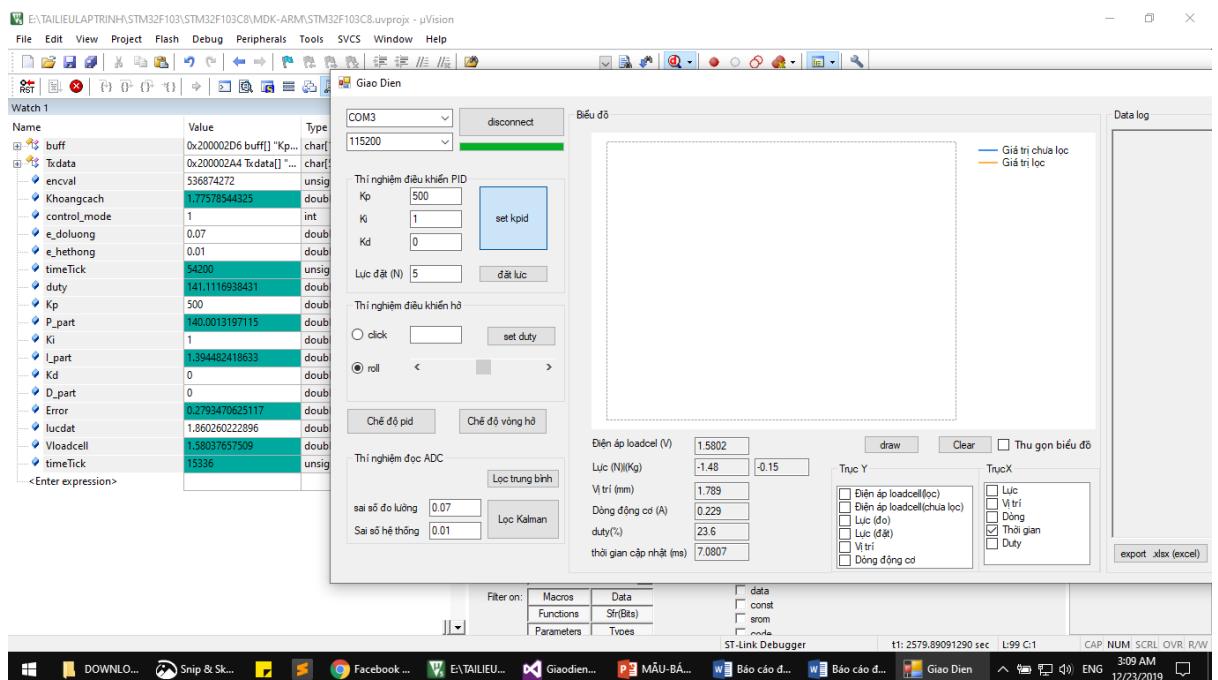
Ở chế độ điều khiển PID, không thay đổi các thông số, thời gian thực thi ổn định

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 9.4: Thời gian điều khiển khi sử dụng thuật toán PID

Khi điều chỉnh giá trị của điều khiển của khung PID thì giá trị thời gian xử lý của vi điều khiển tăng vọt



Hình 9.5: Thời gian điều khiển động cơ khit thay đổi hệ số PID

Bảng 9.1: Thời gian các bước thực hiện chương trình điều khiển

Thí nghiệm	Chế độ	Giá trị thanh ghi CNT timer3 (Cao nhất)	Tổng thời gian thực thi chương trình (μ s)
1	Điều khiển hở, không thay đổi các giá trị	12408	347.424
2	Điều khiển hở, thay đổi các giá trị điều khiển	39950	1118.6
3	PID, không thay đổi các thông số, giá trị đặt	14543	407.204
4	PID, thay đổi các giá trị thông số, các giá trị đặt	54200	1517.6

Kết luận:

từ bảng số liệu có được, thời gian thực thi của chương trình tùy thuộc vào độ phức tạp của chương trình điều khiển, giá trị lớn nhất đo được là 1.5176 ms , do đó trong đề tài này, các giá trị thời gian lấy mẫu cho bộ điều khiển PID, bộ lọc Kalman phải lấy lớn hơn 1.5176ms

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

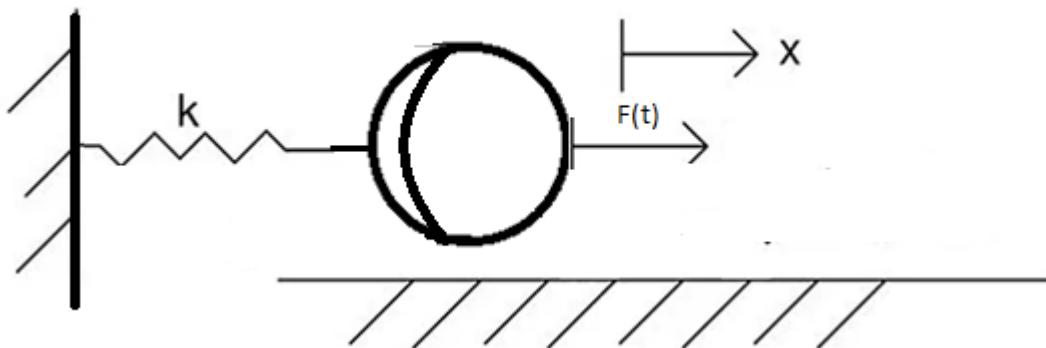
CHƯƠNG 10 XÂY DỰNG CHƯƠNG TRÌNH ĐIỀU KHIỂN ĐỘNG CƠ

1. Hàm truyền của cơ cấu chấp hành haptic 1 bậc tự do

Mô hình hóa hệ thí nghiệm với các đối tượng

Mô hình toán học tay với vật liệu đàn hồi.

làm mô hình thí nghiệm với các vật thể đàn hồi ta có được hệ phương trình có dạng như sau:



Hình 10.1: Mô hình truyền lực tay với vật đàn hồi

Từ đó ta có thể xây dựng hệ phương trình tổng quát như sau:

$$m\ddot{x}(t) + C\dot{x}(t) + kx(t) = F(t)$$

Với m là khối lượng của ngón tay trỏ (đóng vai trò như động cơ). Đơn vị kg

C là hệ số giảm chấn. Trong thí nghiệm trên không có bộ giảm chấn nên ta lấy $C = 0$.

K là hệ số đàn hồi. Trong thí nghiệm là hệ số đàn hồi của bóng ta đo được. Đơn vị N/m

Ta có phương trình trạng thái của cơ hệ bóng và ngón tay như sau:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{m} & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \cdot F(t)$$

Xây dựng cơ hệ của động cơ với ngón tay

$$e(t) = I_a R_a + L_a \frac{dI_a}{dt} + K_m \omega_m$$

$$T_e = K_m I_a$$

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

$$J \frac{d\omega_m}{dt} = T_e - T_L - B_m \omega_m$$

Với e là điện áp phản ứng. Đơn vị V

I_a là dòng điện định mức. Đơn vị Ampe

R_a Là điện trở nội của động cơ. Đơn vị Ω

L_a Cảm kháng của động cơ. Đơn vị H

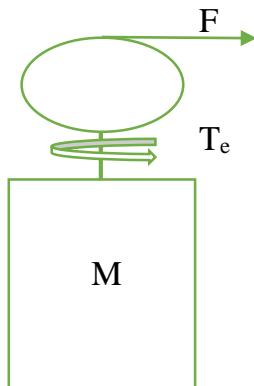
K_m Hằng số Moment xoắn của động cơ. Đơn vị N.m/A

ω_m Tốc độ động cơ. Đơn vị rad/s

T_e Moment xoắn của động cơ. Đơn vị N.m

T_L Moment xoắn của tải. Đơn vị N.m

B_M Hệ số từ thông của động cơ. Đơn vị T (Tesla)



Hình 10.2: Mô hình truyền lực động cơ

J Moment quán tính của động cơ. Đơn vị kg.m^2

Ta có mô hình động cơ điều khiển lực như sau:

Từ mô hình ta có thể viết được công thức như sau:

$$F = T_e / R$$

Trong đó:

F : Là lực kéo động cơ. Đơn vị N

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

T_e : Là Moment xoắn động cơ. Đơn vị N

R: Là bán kính đĩa xoay . Đơn vị m

Phương trình cân bằng lực ngón tay với lực sinh ra của động cơ:

$$\frac{K_m \cdot \frac{e(t)}{R_a}}{R} = F(t)$$

2. Phương pháp điều khiển động cơ

Để điều khiển động cơ ta áp dụng phương pháp điều khiển động cơ bằng PWM (*Pulse-width modulation*)

Ta lại có:

$$e(t) = u \cdot \frac{t}{T}$$

Trong đó:

T_{on} là tần số động cơ được mở. Đơn vị s

T là chu kỳ điều khiển. Đơn vị s

Từ đó ta có:

$$\frac{K_m \cdot u \cdot \frac{t}{T \cdot R_a}}{R} = F(t)$$

Từ phương trình trên và phương trình quan hệ giữa lực và duty của động cơ ta có thể sử dụng để điều khiển lực thông qua duty. Tuy nhiên từ thí nghiệm trong chương VIII ta thấy rằng độ chính xác của việc điều khiển bằng phương pháp này là không cao. Vì bản thân hệ cơ cấu cơ khí là không lý tưởng như điều kiện của phương trình và bản thân phương trình điều khiển lực theo duty của động cơ chỉ chính xác tương đối do các sai số trong đo lường từ các thí nghiệm trước.

3. Tìm hiểu giải thuật điều khiển PID

PID là cách viết tắt của các từ Propotional (tỉ lệ), Integral (tích phân), Derivative (đạo hàm) và là giải thuật điều khiển được dùng nhiều nhất trong các ứng dụng điều khiển

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo tự động với yêu cầu chính xác (accurate), nhanh (fast response), ổn định (small overshoot).

Một điều rất tự nhiên, với yêu cầu nhanh thì một cách đơn giản để công thức hóa ý tưởng này là dùng quan hệ tuyến tính:

$$F = K_p \cdot e(t)$$

Trong đó K_p là một hằng số dương nào đó mà chúng ta gọi là hệ số P (Propotional gain), e là sai số cần điều khiển. Mục tiêu điều khiển là đưa e (sai số) tiến về 0 càng nhanh càng tốt. Rõ ràng nếu K_p lớn thì tác động nhanh của điều khiển cũng càng lớn. Tuy nhiên, do quán tính mà việc điều khiển càng nhanh càng gây ra tính mất ổn định (do lực quán tính và lực điều khiển tạo ra cặp đôi lực xuất hiện ở hai khoảng thời gian liên tiếp nhau => chúng tạo ra dao động không kiểm soát được). Như vậy, ta sẽ sử dụng đạo hàm của sai số e để làm tăng giá trị nhưng ngược chiều của lực F (vì e đang giảm nhanh dần). Nếu sử dụng đạo hàm làm thành phần “thắng” thì có thể giảm được overshoot của quá trình. Thành phần “thắng” này chính là thành phần D (Derivative) trong bộ điều khiển PID mà chúng ta đang khảo sát. Thêm thành phần D này vào bộ điều khiển P hiện tại, chúng ta thu được bộ điều khiển PD như sau:

$$F = K_p \cdot e(t) + K_d \cdot (de/dt)$$

Trong đó (de/dt) là vận tốc thay đổi của sai số e và K_d là một hằng số không âm gọi là hệ số D (Derivative gain). Sự hiện diện của thành phần D làm giảm overshoot vật thể điều khiển khi nó tiến gần về vị trí cân bằng (vị trí ổn định), lực F gồm 2 thành phần $K_p \cdot e >= 0$ (P) và $K_d \cdot (de/dt) <= 0$ (D). Trong một số trường hợp thành phần D có giá trị lớn hơn thành phần P và lực F đổi chiều, “thắng” (hỗn tốc) lại, yếu tố cần điều khiển (ví dụ vận tốc, vị trí...) của vật thể giảm mạnh ở gần vị trí cân bằng. Một vấn đề này sinh là nếu thành phần D quá lớn so với thành phần P hoặc bán thân thành phần P nhỏ thì khi tiến gần điểm cân bằng (chưa thật sự đến vị trí này), vật thể có thể dừng hẳn, thành phần D bằng 0 (vì sai số e không thay đổi nữa), lực $F = K_p \cdot e$. Trong khi K_p và e lúc này đều nhỏ nên lực F cũng nhỏ và có thể không thắng được lực ma sát tĩnh. Sai số e trong tình huống này gọi là steady state error (tạm dịch là sai số trạng

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo thái tĩnh). Để tránh steady state error, người ta thêm vào bộ điều khiển một thành phần có chức năng “cộng dồn” sai số.

Khi steady state error xảy ra, 2 thành phần P và D mất tác dụng, thành phần điều khiển mới sẽ “cộng dồn” sai số theo thời gian và làm tăng lực F theo thời gian. Đến một lúc nào đó, lực F đủ lớn để thắng ma sát tĩnh và đẩy vật tiến tiếp về điểm cân bằng. Thành phần “cộng dồn” này chính là thành phần I (Integral - tích phân) trong bộ điều khiển PID. Vì chúng ta điều biết, tích phân một đại lượng theo thời gian chính là tổng của đại lượng đó theo thời gian. Bộ điều khiển đến thời điểm này đã đầy đủ là PID:

$$F = K_p e(t) + K_d \left(\frac{de}{dt} \right) + K_i \int_0^t e(\tau) d(\tau)$$

Như vậy, chức năng của từng thành phần trong bộ điều khiển PID giờ đã rõ. Tùy vào mục đích và đối tượng điều khiển mà bộ điều khiển PID có thể được lượt bớt để trở thành bộ điều khiển P, PI hoặc PD. Công việc chính của người thiết kế bộ điều khiển PID là chọn các hệ số K_p , K_d và K_i sao cho bộ điều khiển hoạt động tốt và ổn định (quá trình này gọi là PID gain tuning). Đây không phải là việc dễ dàng vì nó phụ thuộc vào nhiều yếu tố.

Tóm tắt một kinh nghiệm cơ bản khi chọn các hệ số cho PID như sau:

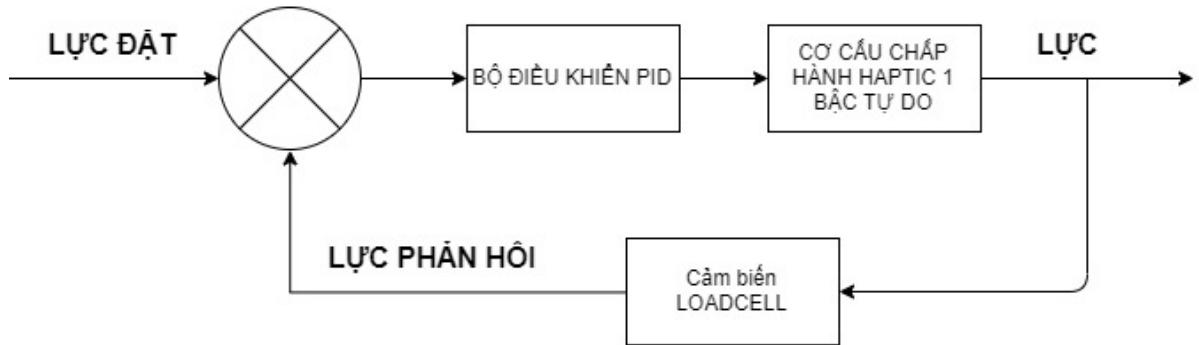
-Chọn K_p trước: thử bộ điều khiển P với đối tượng thật (hoặc mô phỏng), điều chỉnh K_p sao cho thời gian đáp ứng đủ nhanh, chấp nhận overshoot nhỏ. Thêm thành phần D để loại overshoot, tăng K_d từ từ, thử nghiệm và chọn giá trị thích hợp. Steady state error có thể sẽ xuất hiện.

-Thêm thành phần I để giảm steady state error. Nên tăng K từ bé đến lớn để giảm steady state error đồng thời không để cho overshoot xuất hiện trở lại. Có một phương pháp rất phổ biến dùng để chọn các hệ số cho bộ điều khiển PID gọi là Ziegler–Nichols.

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

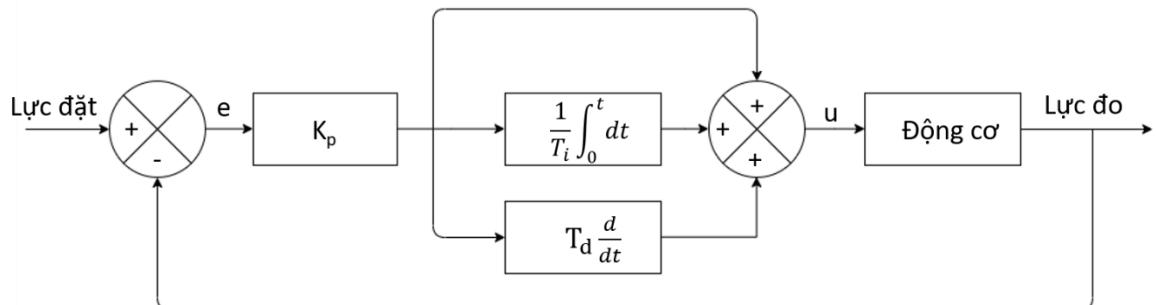
4. Áp dụng bộ điều khiển PID vào điều khiển động cơ sinh lực trong thiết bị haptic:

4.1 Mô hình điều khiển:



Hình 10.3: Mô hình điều khiển Haptic

4.2 Bộ điều khiển PID số dạng 1 (P-ID song song lý tưởng)^[9]:



Hình 10.4: Mô hình điều khiển PID song song

Từ bộ điều khiển PID liên tục:

$$G_{pid}(s) = K_p + \frac{K_i}{s} + K_d * s \quad (1)$$

Với:

$$K_p : hệ số tỉ lệ$$

$$K_i = \frac{K_p}{T_i}$$

$$K_d = K_p * T_d$$

T_i : Thời gian tích phân

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

T_d : Thời gian vi phân

K_i : Hệ số tích phân

K_d : Hệ số vi phân

Để áp dụng cho vi điều khiển, ta tiến hành chuyển đổi bộ điều khiển PID thành dạng rời rạc

Rời rạc hóa khâu tích phân(1/s): dùng biến đổi hình thang ta có

$$s \rightarrow \frac{2}{T} * \frac{z - 1}{z + 1}$$

Rời rạc hóa khâu vi phân(s): dùng biến đổi ngược ta có

$$s \rightarrow \frac{z - 1}{T * z}$$

Thay vào (1) ta có hàm truyền bộ điều khiển PID dạng số:

$$G_{pid}(z) = K_p + K_i * \frac{T}{2} * \frac{z + 1}{z - 1} + \frac{K_d}{T} * \frac{z - 1}{z}$$

$$G_{pid}(z) = \frac{a_0 + a_1 * z^{-1} + a_2 * z^{-2}}{1 - z^{-1}}$$

$$\text{Trong đó: đặt } a_0 = K_p + \frac{K_i T}{2} + \frac{K_d}{T}; a_1 = -K_p + k_i T - \frac{2K_d}{T}; a_2 = \frac{K_d}{T}$$

Như vậy, với bộ điều khiển pid số cho hệ haptic 1 bậc tự do, ta tính được tín hiệu điều khiển $u(k)$ khi có tín hiệu đầu vào $e(k)$

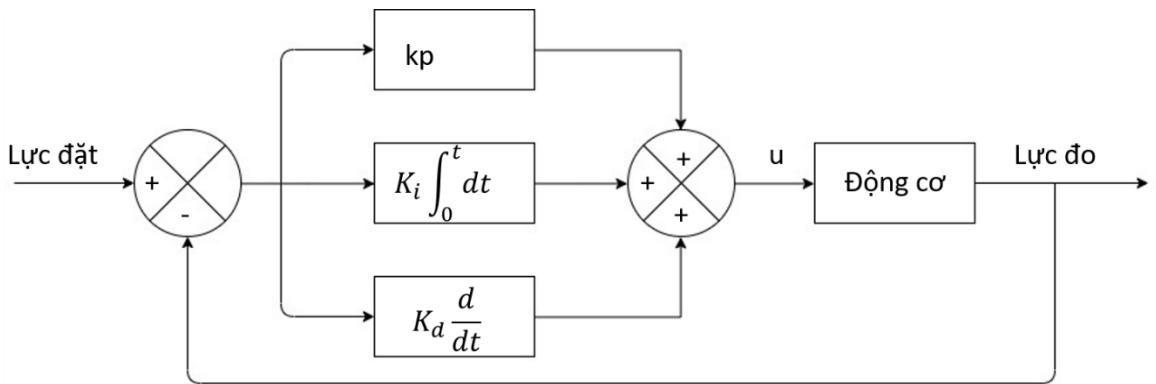
$$u(k) = G_{pid}(z) * e(k)$$

⇒ Ta có được công thức cập nhật giá trị PID sau:

$$u(k) = u(k - 1) + a_0 * e(k) + a_1 * e(k - 1) + a_2 * e(k - 2)$$

4.3 Bộ điều khiển PID số dạng 2 (PID song song) [9]:

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 10.5: Mô hình PID song song

Ta có :

$$\text{Khâu P rời rạc: } G_p(z) = K_p \Rightarrow P = u(k) = K_p e(k)$$

$$\text{Khâu I rời rạc: } G_i(z) = K_i \frac{\frac{T}{2} z^{-1}}{1 - z^{-1}} \Rightarrow I = u(k) = u(k-1) + K_i \frac{T}{2} (e(k) - e(k-1))$$

$$\text{Khâu D rời rạc: } G_d(z) = \frac{k_d}{T} (1 - z^{-1}) \Rightarrow D = u(k) = K_p \frac{T}{2} (e(k) - e(k-1))$$

Tổng ba khâu: tín hiệu điều khiển của bộ điều khiển PID là tổng của ba khâu P+I+D

Ta có mã giả của bộ PID rời rạc áp dụng cho vi điều khiển như sau:

previous_error = 0

integral = 0

loop:

 error = setpoint - measured_value

 integral = integral + error * dt

 derivative = (error - previous_error) / dt

 output = Kp * error + Ki * integral + Kd * derivative

 previous_error = error

 wait(dt)

 goto loop

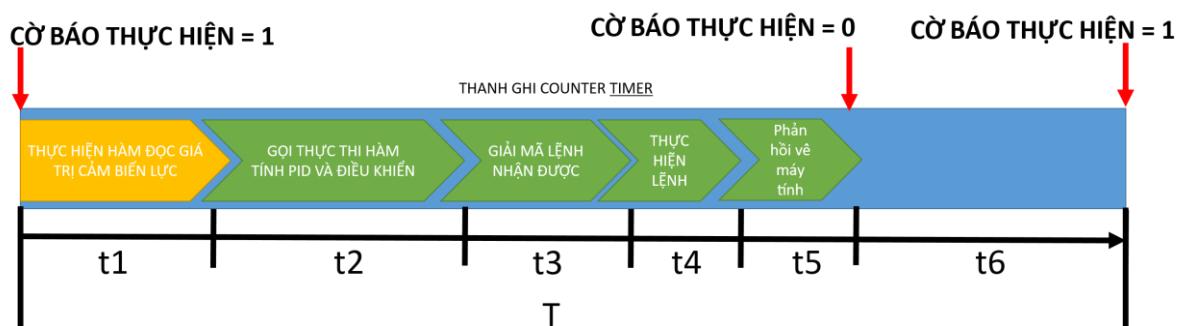
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

4.4 Chọn thời gian lấy mẫu T và phân cấp ưu tiên thực thi trong chường trình C cho bộ điều khiển PID

Ta thấy rằng hiệu quả bộ điều khiển pid rời rạc ngoài phụ thuộc vào các hệ số kp, ki, kd còn ảnh hưởng bởi thời gian lấy mẫu T, lý tưởng nhất là chọn được T với giá trị nhỏ nhất có thể.

Trong chương trình C, sử dụng một timer ngắn để định thời chu kỳ lấy mẫu, khi có ngắt xảy ra, ta bật một cờ báo thực hiện, khi chương trình thực hiện xong thì reset cờ báo đi, chờ cờ báo tiếp theo được bật lên. Chu kỳ lấy mẫu phải được chọn sao cho lớn hơn tổng thời gian thực thi tối đa của các hàm tác vụ chạy trong chương trình chính.

Cấp độ ưu tiên thực hiện các chương trình và thời gian thực thi trong một chu kỳ lấy mẫu T như hình dưới đây



Hình 10.6: Cấp độ ưu tiên thực hiện một chu kỳ lệnh

$$\text{Tổng thời gian } t_1 + t_2 + t_3 + t_4 + t_5 + t_6 \leq T$$

Trong đó t6 là khoảng thời gian dư, để phòng trường hợp thời gian thực thi của các hàm con tăng bất thường. Từ thí nghiệm trong chương trước, trong đề tài này T phải lớn hơn 1.5176ms

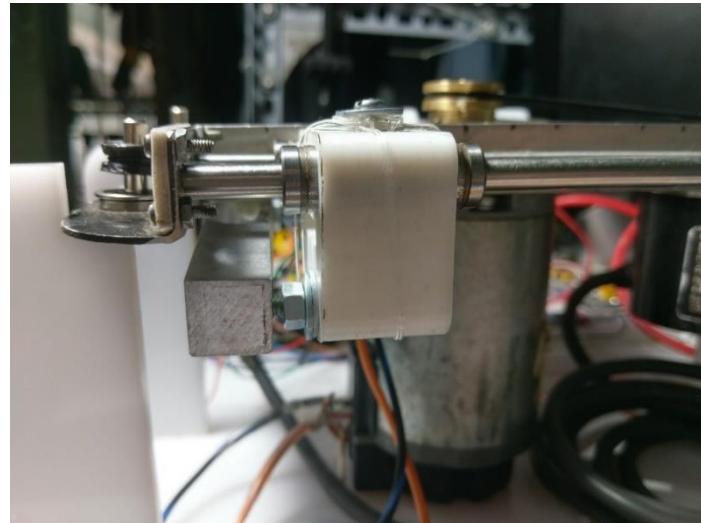
4.5 Thí nghiệm khảo sát tác động bộ điều khiển PID để điều khiển động cơ sinh lực

Mục đích thí nghiệm: tìm ra bộ số Kp, Ki, Kd phù hợp nhất cho việc điều khiển cơ cấu chấp hành haptic 1 bậc tự do sinh ra lực mô phỏng

Bố trí thí nghiệm:

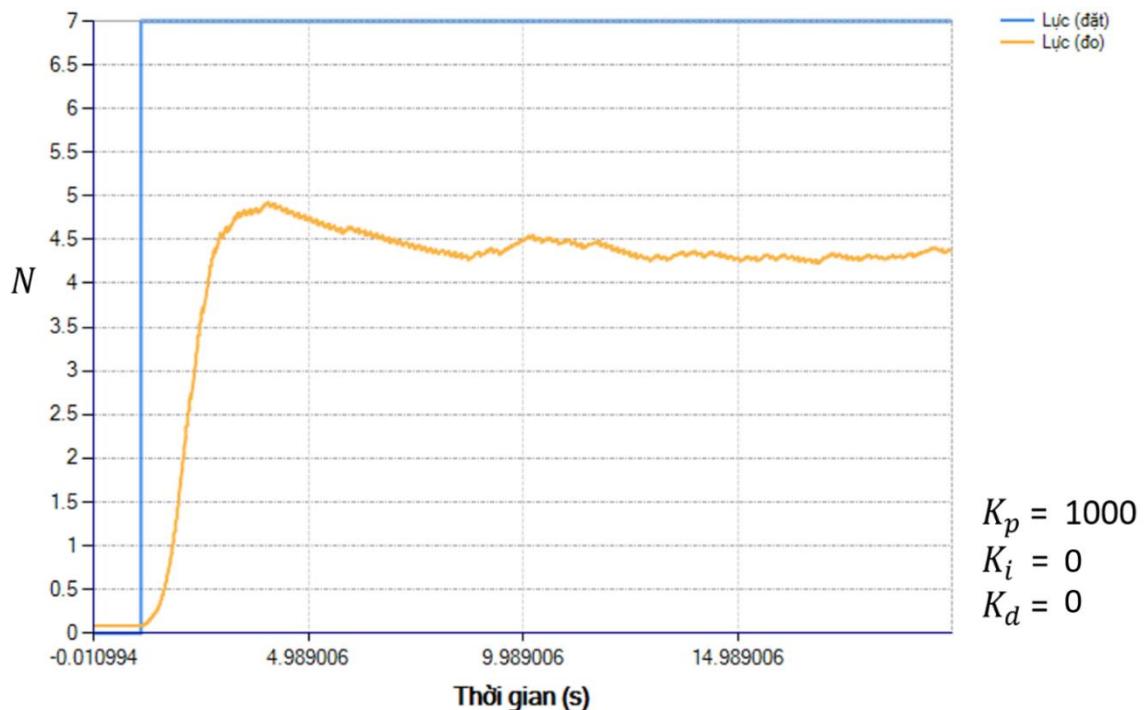
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Gắn cố định cơ cấu với loadcell, chọn chu kỳ lấy mẫu T là 10ms, kết nối với phần mềm máy tính, thay đổi các giá trị K_p , K_i , K_d và Giá trị lực đặt. Quan sát biểu đồ đáp ứng



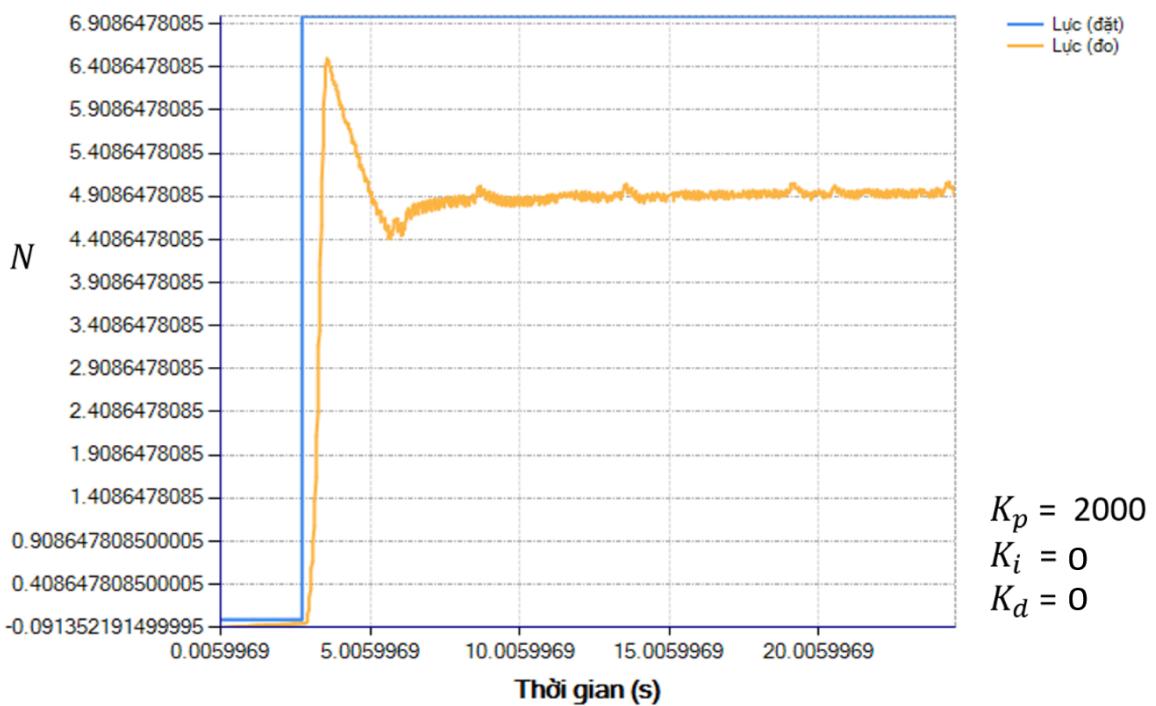
Hình 10.7: Mô hình thí nghiệm điều khiển lực sử dụng thuật toán PID

4.5.1 Đáp ứng của bộ PID số dạng 1 khi thay đổi các thông số:

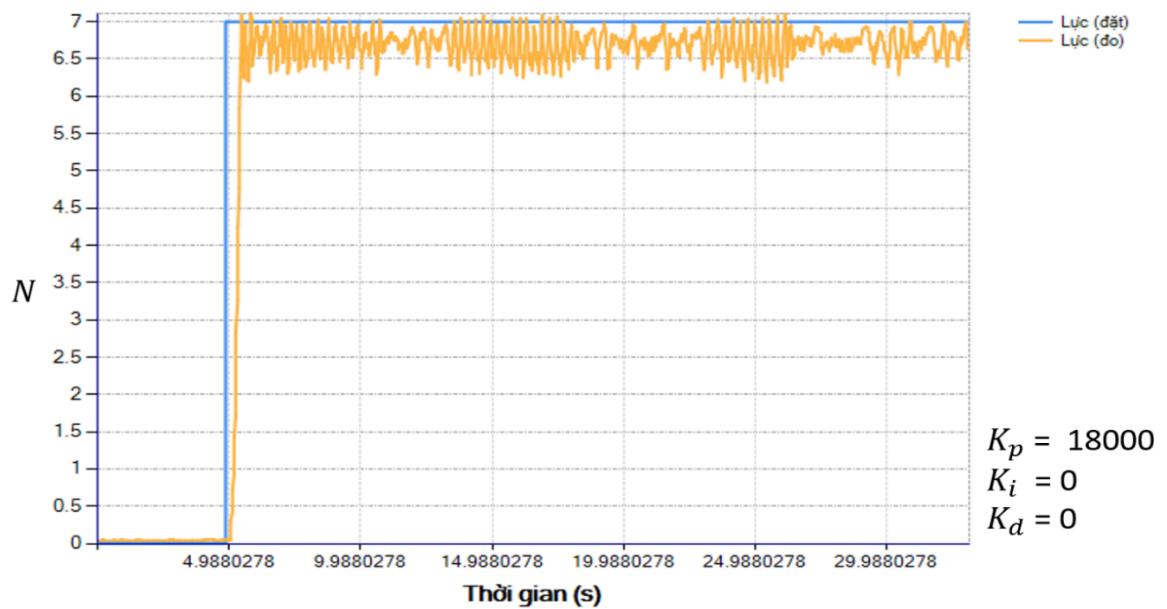


Hình 10.8 Đáp ứng khâu P bộ điều khiển pid số dạng 1 $K_p=1000$

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

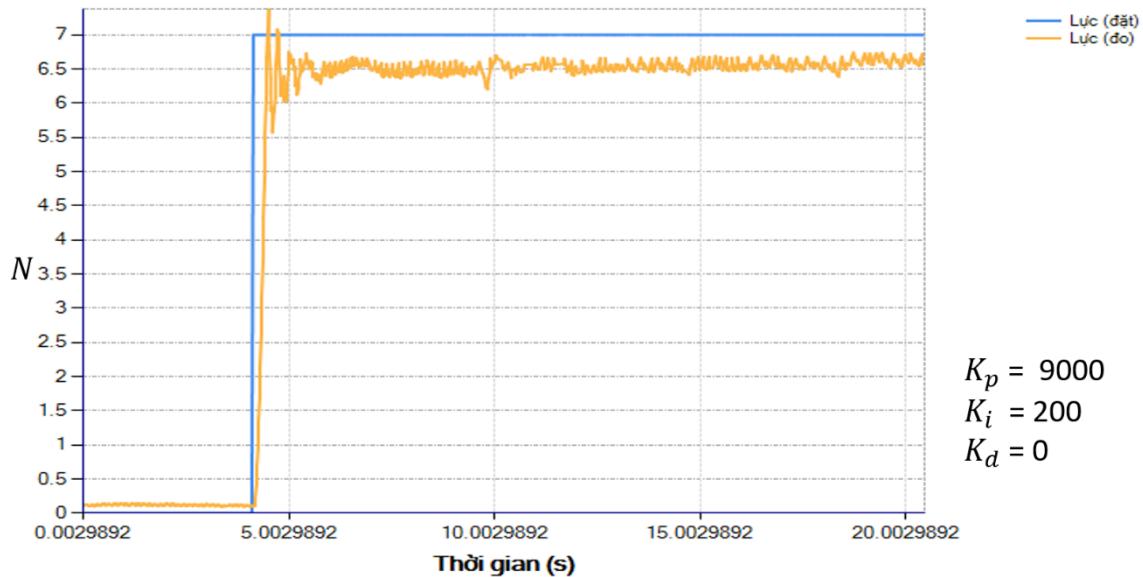


Hình 10.9 Đáp ứng khâu P bộ điều khiển pid số dạng I $K_p=2000$

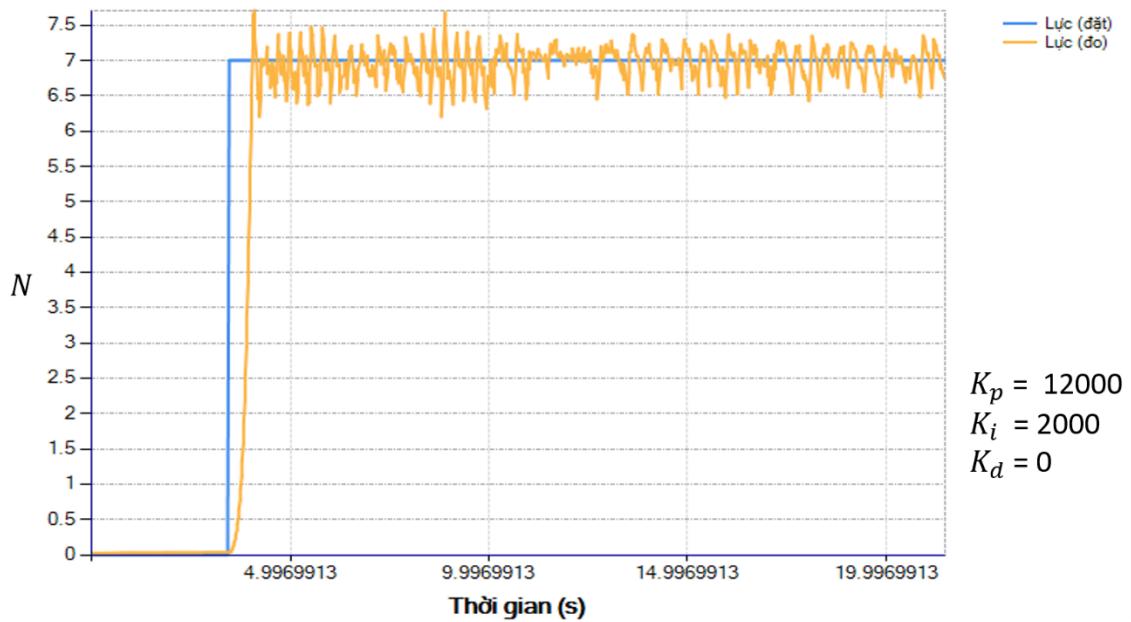


Hình 10.10 Đáp ứng khâu P bộ điều khiển pid số dạng I $K_p = 18000$

Kết hợp PI



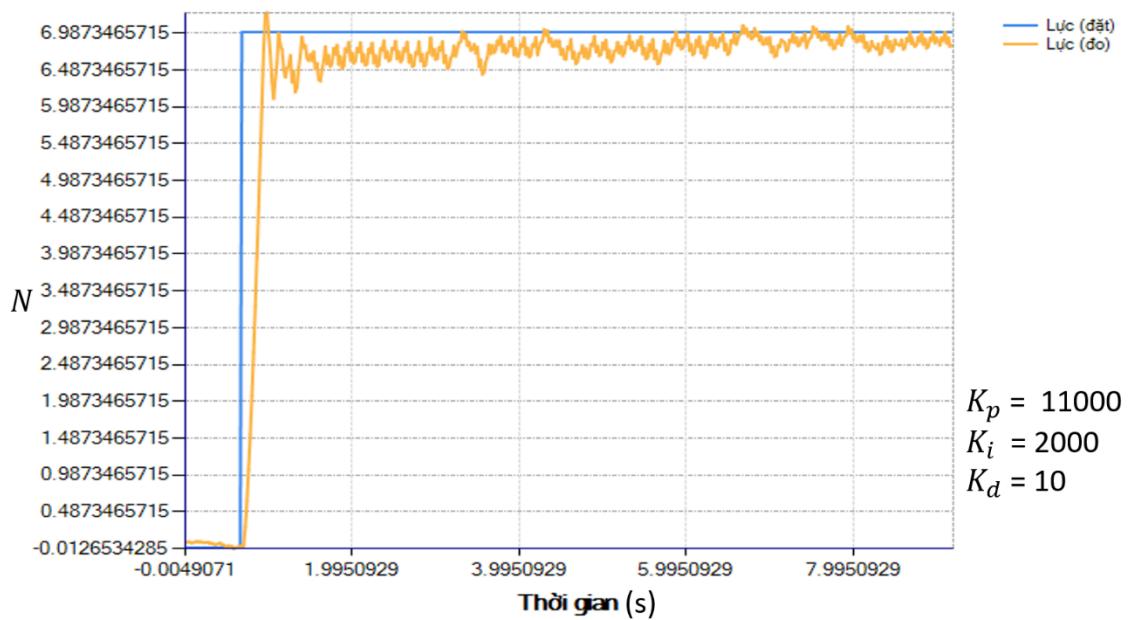
Hình 10.11 Đáp ứng kết hợp PI bô điều khiển pid số dạng 1



Hình 10.12 Đáp ứng kết hợp PI bô điều khiển pid số dạng 1

Kết hợp ba khâu PID:

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



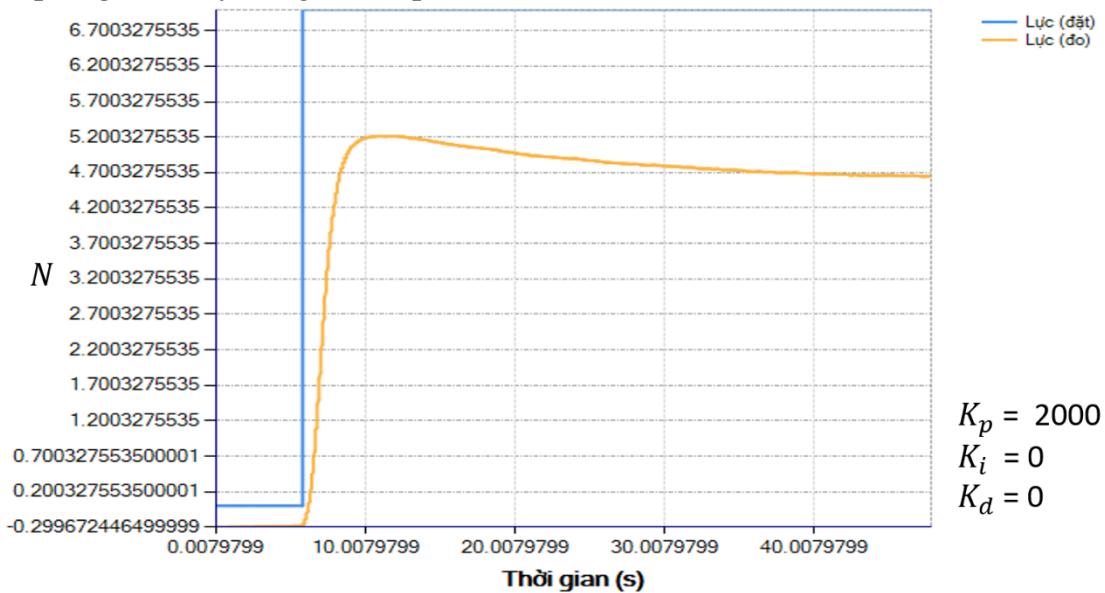
Hình 10.13 đáp ứng bộ PID dạng 1



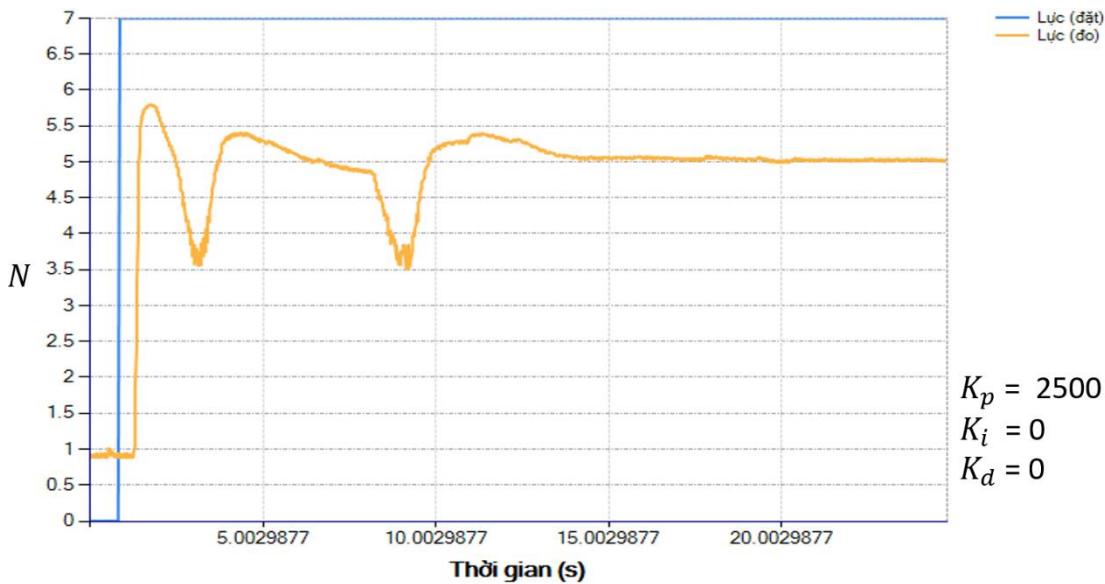
Hình 10.14 đáp ứng bộ PID dạng 1

4.5.2 Đáp ứng của bộ PID số dạng 2 khi thay đổi các thông số

Đáp ứng khi thay đổi giá trị K_p :



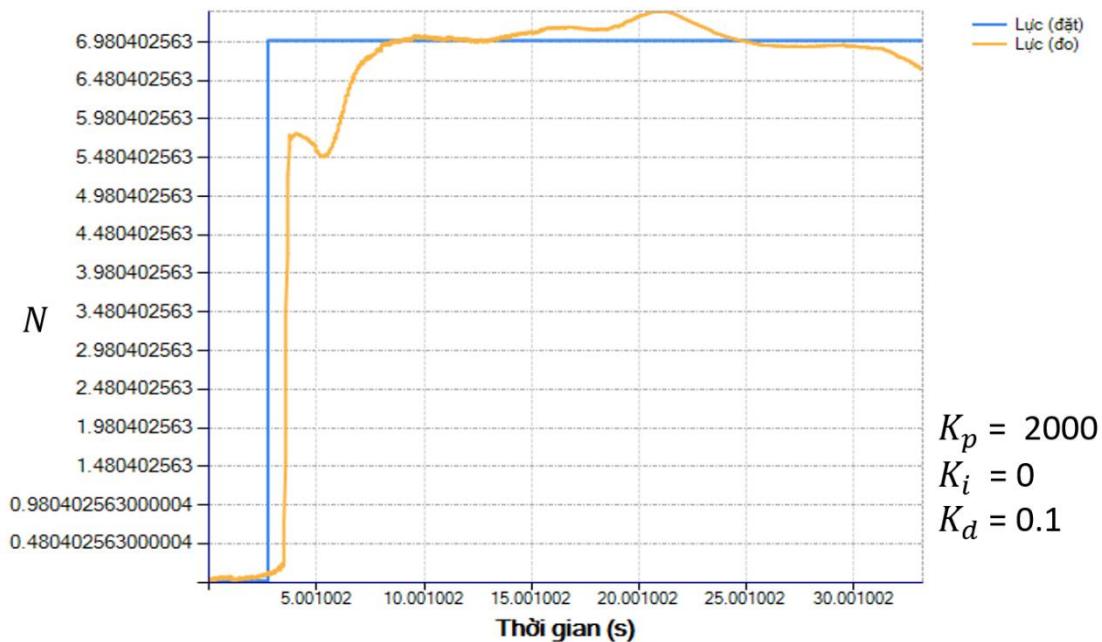
Hình 10.15 Đáp ứng khung P , bộ PID dạng 2, $K_p = 2000$



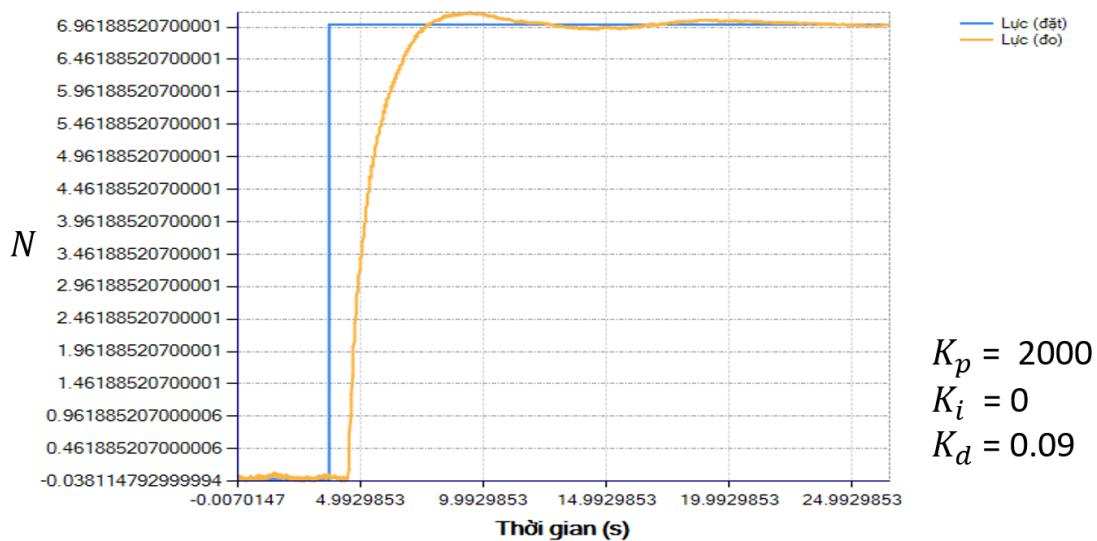
Hình 10.16 Đáp ứng khung P , bộ PID dạng 2, $K_p = 2500$

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Điều chỉnh theo PD:



Hình 10.17 Đáp ứng PD, bộ PID dạng 2



Hình 10.18 Đáp ứng PD, Bộ PID dạng 2

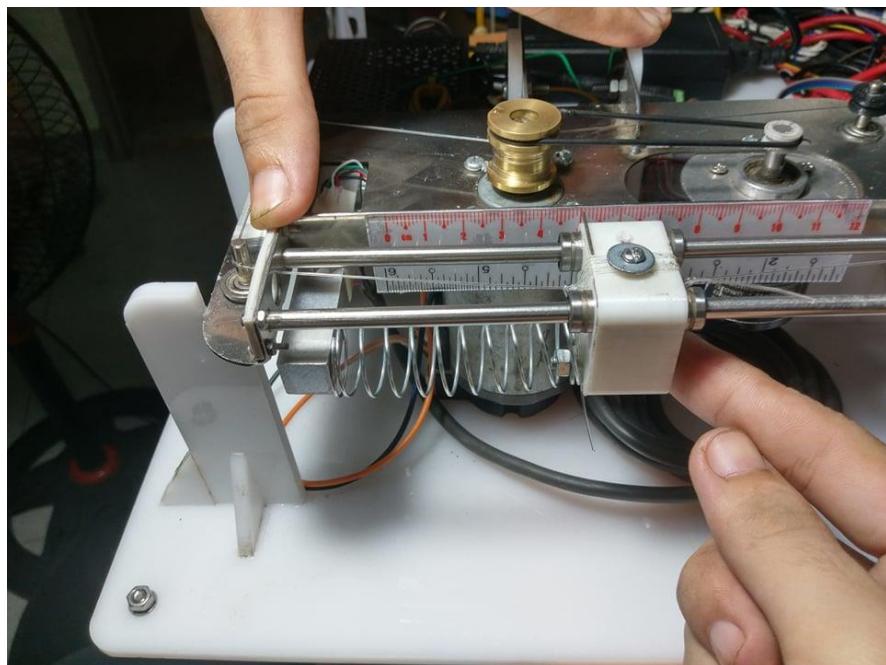
Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

CHƯƠNG 11 THÍ NGHIỆM KIỂM CHỨNG LỰC MÔ PHỎNG VÀ MÔ PHỎNG VỚI VẬT THỂ ĐÀN HỒI ẢO

1. Thí nghiệm tìm phương trình đặc tính của vật mô phỏng (lò xo)

Mục đích thí nghiệm: Tìm được phương trình đặc tính độ biến dạng – lực phản lực lò xo.

Bố trí thí nghiệm: bô trí thí nghiệm như hình. Đặt vị trí 0 tương ứng với điểm 0 trên thước. Đặt lò xo vào giữa loadcell và khớp đặt ngón trỏ, dùng tay ép biến dạng lò xo, trên giao diện quan sát biểu đồ và lưu lại dữ liệu. Dữ liệu này được xuất thành file data.dat đưa sang matlab. Áp dụng thuật toán quy hồi tuyến tính để tìm ra đường đặc tính với sai số nhỏ nhất bằng tập dữ liệu có được.



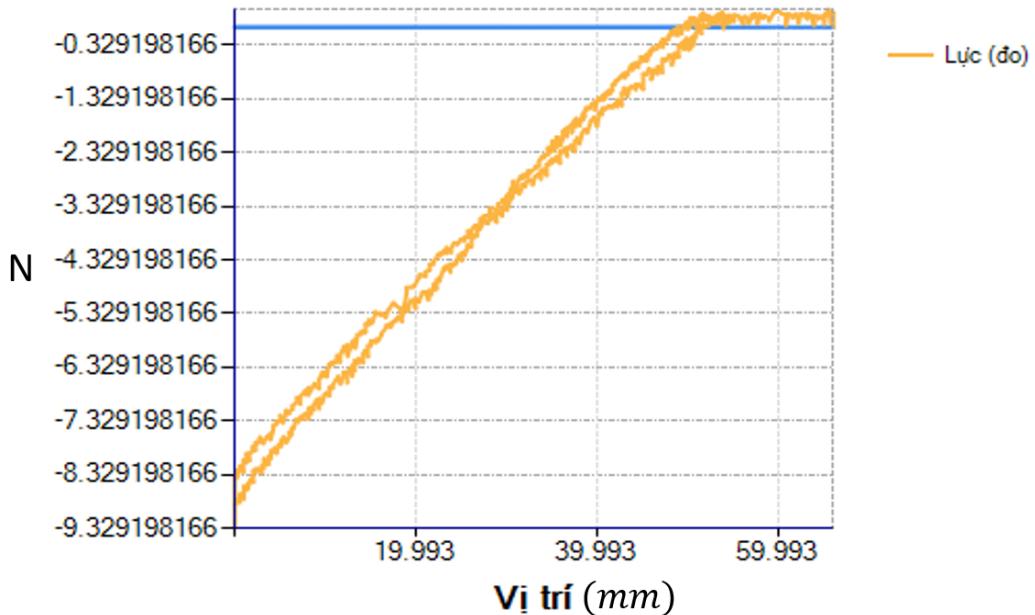
Hình 11.1: Mô hình thí nghiệm lực lò xo



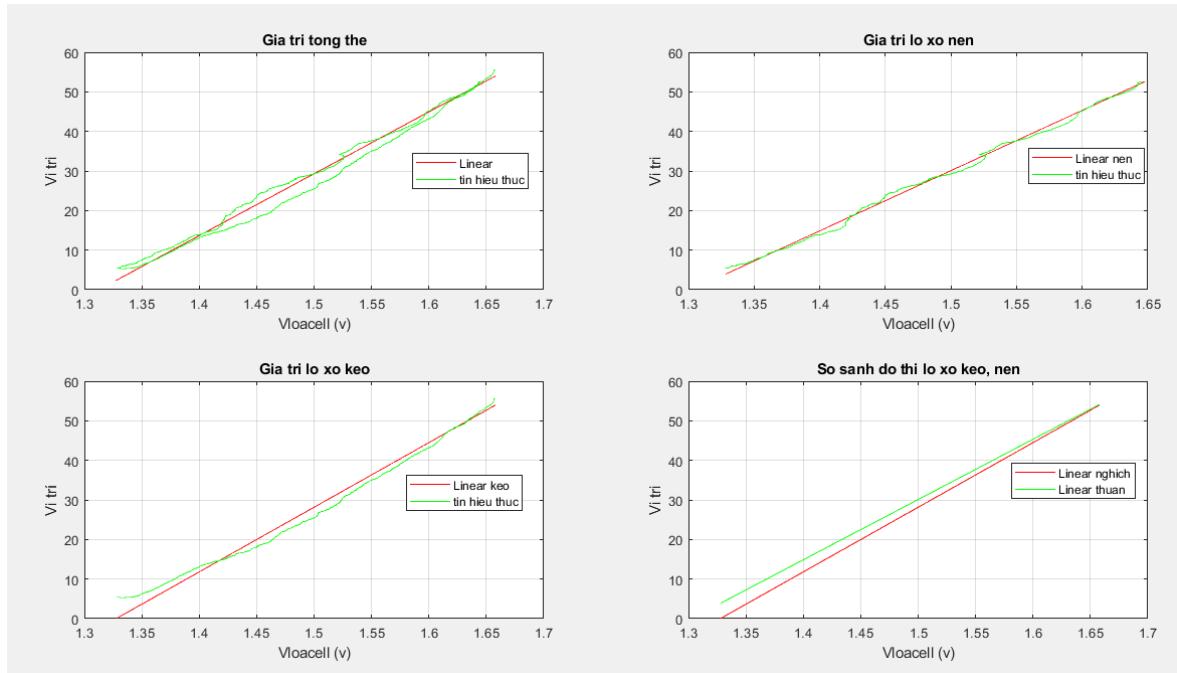
Hình 11.2: Lực tác dụng của lò xo lên loadcell

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

Ta thu được kết quả như sau:



Hình 11.1 Biểu đồ vị trí - lực



Hình 11.3: Đồ thị sau khi lấy Linear Regression

ta có phương trình điện áp loadcell \rightarrow vị trí:
 $y(x) = 156.39x - 205.28$ (mm)

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

=> phương trình vị trí → điện áp loadcell:

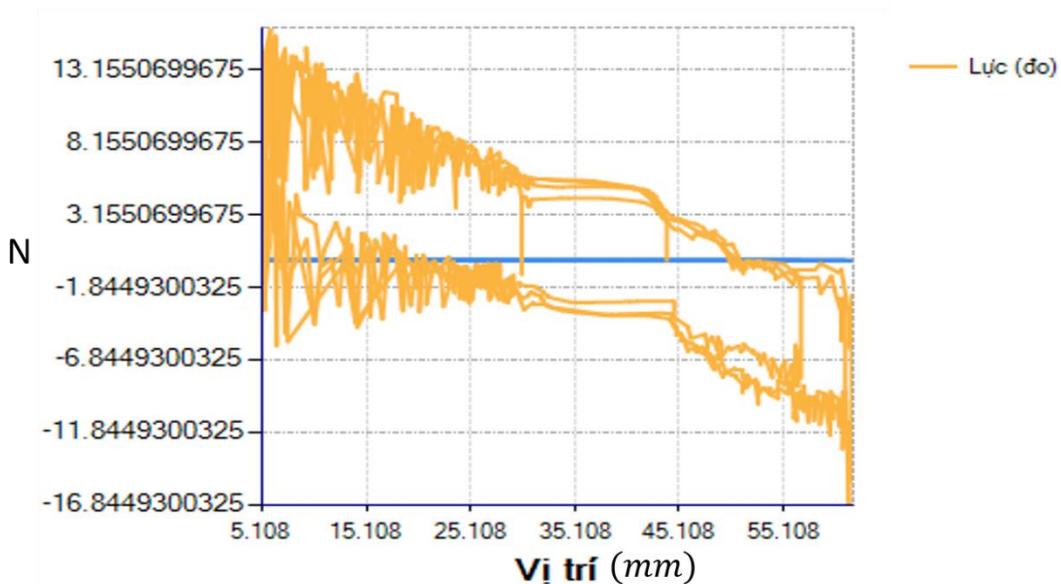
$$p(x) = 0.00636x + 1.31 \text{ (V)}$$

lực mô phỏng có chiều ngược với chiều của lực đàn hồi lò xo tác động lên loadcell, ta có phương trình vị trí – điện áp loadcell theo chiều ngược lại:

$$V(x) = 3.26 - (0.0636x + 1.31) \text{ (V)}$$

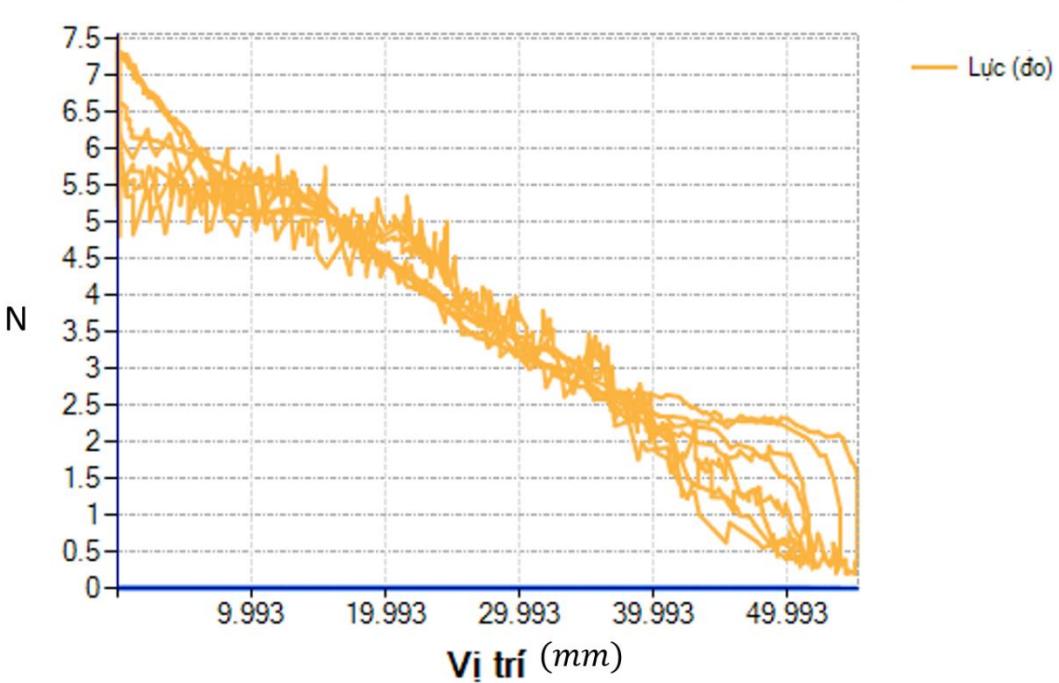
2. Thí nghiệm mô phỏng lò xo

Sau khi có được phương trình trên, bỏ vật ra khỏi cơ cấu và tiến hành mô phỏng vật ảo theo đặc tính đã tìm ra từ vật thật. Kết quả mô phỏng theo hai phương pháp điều khiển vòng hở và vòng kín theo lực phản hồi thu được các biểu đồ dưới đây



Hình 11.2 mô phỏng lò xo điều khiển hở

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo



Hình 11.3 mô phỏng với bộ điều khiển pid lực

Nhận xét:

Phương pháp vòng hở mô phỏng được vật ảo có đường đáp ứng bám theo hình dạng của đường đặc tuyến vật thật, tuy nhiên sai số lớn

Phương pháp vòng kín có đường đáp ứng bám theo hình dạng của đường đặc tuyến vật thật, sai số nhỏ

Về mặt cảm nhận vật ảo mô phỏng đúng cảm giác lực khi tay bóp lò xo thực.

CHƯƠNG 12 KẾT LUẬN VÀ KIẾN NGHỊ

Đề tài đã tìm hiểu các khái niệm và một số đặc trưng của hệ thống haptic.

Cơ khí: Chế tạo thành công cơ cấu haptic 1 bậc tự do dựa trên cơ cấu tang trống kéo dây.

Phần mềm giao diện: Hiển thị được biểu đồ cho phép lựa chọn dữ liệu hiển thị trên các trực thuẫn tiện sử dụng. dữ liệu được lưu trữ theo thời gian thực giúp việc phân tích kết quả dễ dàng sau các thí nghiệm.

Đo lường dữ liệu: áp dụng thành công các bộ lọc nhiễu cho bộ chuyển đổi adc của vi điều khiển

Điều khiển: áp dụng được bộ điều khiển PID (với khâu I=0) để điều khiển chính xác động cơ sinh lực

Đối với thiết bị haptic phản hồi lực: cường độ lực tối đa không cần cao mà cần độ chính xác, thời gian đáp ứng nhanh, ổn định. Một thiết bị haptic phản hồi lực cần đạt: độ phân giải 0.1N, khoảng trễ chuyển đổi lực tối đa 20ms thì hiệu quả cảm nhận mới tốt.

Khi sử dụng thiết bị haptic phản hồi lực 1 bậc tự do, để đảm bảo hiệu quả cảm nhận, cần khống chế các bậc tự do còn lại của bàn tay.

TÀI LIỆU THAM KHẢO

[1]: <https://mitpress.mit.edu/books/haptics>

[2]: <https://www.lexico.com/definition/haptic>

[3]: Christian Hatzfeld, Thorsten A. Kern (2014), Engineering Haptic Devices: A Beginner's Guide, Springer.

[4]: José-Luis Rodríguez, Ramiro Velázquez, Carolina Del-Valle-Soto, Sebastián Gutiérrez, Jorge Varona and Josué Enríquez-Zarate(2019), Active and Passive Haptic Perception of Shape Passive Haptics Can Support Navigation. Panamericana University, Mexico .

[5]:<http://zung.zetamu.net/2011/06/tim-hi%E1%BB%83u-v%E1%BB%81-1%E1%BB%8Dc-kalman-kalman-filter/>

[6]: Lê Ngọc Duy, Phan Đình Hiếu, Nguyễn Anh Tú, Lưu Vũ Hải (2018) . Nghiên cứu lọc thích nghi – Kalman cho cảm biến gia tốc trên xe lăn điện, , Tạp chí khoa học (Đại học công nghiệp Hà Nội).

[7]: David A. Freedman . Statistical Models: Theory and Practice (2009). Cambridge University Press. p. 26. A simple regression equation has on the right hand side an intercept and an explanatory variable with a slope coefficient. A multiple regression equation has two or more explanatory variables on the right hand side, each with its own slope coefficient.

[8]: <https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86>.

[9]: Waleed Ahmed Mohammed Hussein Ahmed(2003). PID Control of DC-Motors. University of Khartoum.

PHỤ LỤC

Phần (Code C) chương trình vi điều khiển

Chương trình chính: main.c

```
#include "main.h"
#include "stdio.h"
#include "string.h"

ADC_HandleTypeDef hadc1;
DMA_HandleTypeDef hdma_adc1;
TIM_HandleTypeDef htim1;
TIM_HandleTypeDef htim2;
TIM_HandleTypeDef htim3;
UART_HandleTypeDef huart1;

#define FIR_filter 0
#define Kalman_filter 1
#define PID_mode 1
#define Mohpongloxo_mode 2
#define linear_mode 3
#define OpenLoop_mode 0

volatile uint32_t encval;
volatile double duty=0;

/*khai bao cac bien xu ly pid*/
volatile double e_z =0;
volatile double e_z1=0;
volatile double e_z2=0;
volatile double a_0=0;
volatile double a_1=0;
volatile double a_2=0;
volatile double u_z=0;
volatile double u_z1=0;
volatile double tichphan = 0;
volatile double tichphan_pre = 0;
volatile double viphhan = 0;

volatile double Kp=0,Ki=0,Kd=0,kzero=1;
volatile double T=0.01; //10ms
volatile int pid_finish=1;

volatile double lucdat=0;
volatile double lucdo=0;

/*khai bao cac co bao*/
volatile int control_mode=PID_mode,timerFlag=0,method=0;
volatile uint8_t cmd=0;

/*khai bao cac mang phuc vu truyen nhan uart*/
char Txdata[50];
char Rxdata[1];
char buff[10];

/*Khai bao cac bien xu ly ADC, circle buffer*/
#define ringsize 100 // so mau lay trung binh
volatile uint16_t adc_value[2];
volatile double adc0_ring_buff[ringsize];
volatile double ring0_sum=0;
volatile int ring0_head=0,ring0_tail=ringsize-1;
volatile double adc1_ring_buff[ringsize];
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
volatile double ring1_sum=0;
volatile int ring1_head=0,ring1_tail=ringsize-1;
volatile double Vloadcell;
volatile double Vzero;
volatile double Vdongco;
volatile double Khoangcach;
double Vdat=1.644;

/* khai bao cac bien xu ly trung gian xu ly lenh tu pc*/
int i=0,buffsize=0;
uint16_t dem=0;
double so=0;
double num=0;
int dot,dotflag,negative_flag,buff_i=0;

/*khai bao cac bien bo loc Kalman*/
double Kalman_gain=0,output_hientai=0,output_truoc=0,e_uoclouong=0,e_doluong=0,e_hethong=0,input;

volatile uint32_t timeTick=0;

/*khai bao cac bien mo phong lo xo*/
volatile double heso_a=0;
volatile double heso_b=0;
volatile double lucdatloxo;
/*cac ham cau hinh cua thu vien*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_ADC1_Init(void);
static void MX_TIM1_Init(void);
static void MX_TIM2_Init(void);
static void MX_TIM3_Init(void);
static void MX_USART1_UART_Init(void);

double docKhoangCach(uint32_t encval);
double docDienAp_loadcell(uint32_t adcvalue,int method);
double docDienAp_Rshunt(uint32_t adcvalue);
double str2num(uint8_t *str,int fp,int ep);
double KalmanFilterADC(double input);
void pidcompute_type2(double luc );
void pidcompute_type1(double luc);
void openloopControl(double duty);
void readCMDPacket();
double MophongLoXo(double a, double b, double vitri);
double LinearControl(double a, double b, double vitri);
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart);

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_ADC1_Init();
    MX_TIM1_Init();
    MX_TIM2_Init();
    MX_TIM3_Init();
    MX_USART1_UART_Init();

    HAL_DMA_Init(&hdma_adc1);
    HAL_ADC_Start_DMA(&hadc1,(uint32_t*)adc_value,2);
    HAL_TIM_Base_Start_IT(&htim3);
    HAL_TIM_Encoder_Start(&htim1,TIM_CHANNEL_ALL);
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
HAL_TIM_PWM_Start(&htim2,TIM_CHANNEL_1);
HAL_UART_Receive_IT(&huart1,(uint8_t*)Rxdata,1);
HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_RESET);
__HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_1,0);

HAL_Delay(2000);

/*lay mau adc doc gia tri loadcel cho bo FIR*/
for(i=0;i<ringsize;i++){
    adc1_ring_buff[i]=adc_value[1]*3.3/4095;
    adc0_ring_buff[i]=adc_value[0]*3.3/4095;
    ring1_sum+=adc1_ring_buff[i];
    ring0_sum+=adc0_ring_buff[i];
}
ring0_head=ringsize-1;
ring0_tail=ringsize-2;
ring1_head=ringsize-1;
ring1_tail=ringsize-2;
Vloadcell=ring1_sum/ringsize;
Vzero=Vloadcell;
lucdat=Vzero;
Vdongco=ring0_sum/ringsize;

/*kalman init*/
output_truoc=Vzero;
e_uocluong=0.1;
e_doluong=0.008;
e_hethong=0.07;

while (1)
{
    /* dem thoi gian thuc thi*/
    //TIM3->CNT=0;
    /* Xu ly lenh nhan duoc tu PC*/
    if(cmd==1&&pid_finish==1){
        readCMDPacket();
        cmd=0;
    }
    /*Dieu khien*/
    if(timerFlag==1){
        //Dieu khien vong ho
        if(control_mode==OpenLoop_mode){
            pid_finish=1;
            Vloadcell=docDienAp_loadcelladc_value[1],Kalman_filter);
            Vdongco = docDienAp_Rshuntadc_value[0]);
            Khoangcach=docKhoangCach(TIM1->CNT);
            openloopControl(duty);
            dem++;
            timerFlag=0;
        }
        if(control_mode==linear_mode){
            pid_finish=1;
            Vloadcell=docDienAp_loadcelladc_value[1],Kalman_filter);
            Vdongco = docDienAp_Rshuntadc_value[0]);
            Khoangcach=docKhoangCach(TIM1->CNT);
            LinearControl(heso_a,heso_b,Khoangcach);
            openloopControl(LinearControl(heso_a,heso_b,Khoangcach));
            dem++;
            timerFlag=0;
        }
        // dieu khien PID
        if(control_mode==PID_mode){
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
/*ngat timer3: tinh toan pid*/
pid_finish=0;
Vdongco = docDienAp_Rshunt(adc_value[0]);
Khoangcach=docKhoangCach(TIM1->CNT);
Vloadcell=docDienAp_loadcell(adc_value[1],Kalman_filter);
//pidcompute(lucdat);
pidcompute_type1(lucdat);
dem++;
pid_finish=1;
timerFlag=0;
}
if(control_mode==Mohpongloxo_mode){
/*ngat timer3: tinh toan pid*/
pid_finish=0;
Vdongco = docDienAp_Rshunt(adc_value[0]);
Khoangcach=docKhoangCach(TIM1->CNT);

Vloadcell=docDienAp_loadcell(adc_value[1],Kalman_filter);
//pidcompute(lucdat);
lucdatloxo=3.28-MophongLoXo(heso_a,heso_b,Khoangcach);
pidcompute_type1(lucdatloxo);
dem++;
pid_finish=1;
timerFlag=0;
}
/*gui du lieu len may tinh*/
if(dem==1&&pid_finish==1){
Vloadcell=docDienAp_loadcell(adc_value[1],Kalman_filter);
Vdongco = docDienAp_Rshunt(adc_value[0]);
Khoangcach=docKhoangCach(TIM1->CNT);
sprintf(Txdata,"%0.3f,%0.4f,%0.3f,%0.3f,%f\n",adc_value[1]*3.3/4095,Vloadcell,Khoangcach,Vdongco,duty);
HAL_UART_Transmit_IT(&huart1,(uint8_t*)Txdata,strlen(Txdata));
dem=0;
}
//timeTick = TIM3->CNT; // tinh toan thoi gian thuc thi
}
}

void SystemClock_Config(void)
{
RCC_OscInitTypeDef RCC_OscInitStruct = {0};
RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

/** Initializes the CPU, AHB and APB busses clocks
*/
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
RCC_OscInitStruct.HSEState = RCC_HSE_ON;
RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
Error_Handler();
}
/** Initializes the CPU, AHB and APB busses clocks
*/
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV2;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
{
    Error_Handler();
}
PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_ADC;
PeriphClkInit.AdclkSelection = RCC_ADCPCLK2_DIV4;
if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
{
    Error_Handler();
}

static void MX_ADC1_Init(void)
{
    ADC_ChannelConfTypeDef sConfig = {0};
    hadc1.Instance = ADC1;
    hadc1.Init.ScanConvMode = ADC_SCAN_ENABLE;
    hadc1.Init.ContinuousConvMode = ENABLE;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.NbrOfConversion = 2;
    if (HAL_ADC_Init(&hadc1) != HAL_OK)
    {
        Error_Handler();
    }
    sConfig.Channel = ADC_CHANNEL_0;
    sConfig.Rank = ADC_REGULAR_RANK_1;
    sConfig.SamplingTime = ADC_SAMPLETIME_239CYCLES_5;
    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sConfig.Channel = ADC_CHANNEL_1;
    sConfig.Rank = ADC_REGULAR_RANK_2;
    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
    {
        Error_Handler();
    }
}
static void MX_TIM1_Init(void)
{
    TIM_Encoder_InitTypeDef sConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 0;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 65535;
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
    htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    sConfig.EncoderMode = TIM_ENCODERMODE_TI12;
    sConfig.IC1Polarity = TIM_ICPOLARITY_RISING;
    sConfig.IC1Selection = TIM_ICSELECTION_DIRECTTI;
    sConfig.IC1Prescaler = TIM_ICPSC_DIV1;
    sConfig.IC1Filter = 0;
    sConfig.IC2Polarity = TIM_ICPOLARITY_RISING;
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
sConfig.IC2Selection = TIM_ICSELECTION_DIRECTTI;
sConfig.IC2Prescaler = TIM_ICPSC_DIV1;
sConfig.IC2Filter = 0;
if (HAL_TIM_Encoder_Init(&htim1, &sConfig) != HAL_OK)
{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
}
static void MX_TIM2_Init(void)
{

    TIM_MasterConfigTypeDef sMasterConfig = {0};
    TIM_OC_InitTypeDef sConfigOC = {0};

    /* USER CODE BEGIN TIM2_Init 1 */
    /*PWM freq = 36*10^6/(1*600)=100Khz */
    /* USER CODE END TIM2_Init 1 */
    htim2.Instance = TIM2;
    htim2.Init.Prescaler = 0; //bo chia tan so cho pwm
    htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim2.Init.Period = 599; //do phan giao cua duty
    htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_PWM_Init(&htim2) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sConfigOC.OCMode = TIM_OCMODE_PWM1;
    sConfigOC.Pulse = 0;
    sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
    sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
    if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
    {
        Error_Handler();
    }
    HAL_TIM_MspPostInit(&htim2);
}

static void MX_TIM3_Init(void)
{
    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};
    /* USER CODE BEGIN TIM3_Init 1 */
    /*Timer 3 interrupt period time = (1*3600)/(36*10^6)*1=0.1ms=10^-4s*/
    /* USER CODE END TIM3_Init 1 */
    htim3.Instance = TIM3;
    htim3.Init.Prescaler = 3599;
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 99;
    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim3.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
{
    Error_Handler();
}
sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
if (HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig) != HAL_OK)
{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
}
static void MX_USART1_UART_Init(void)
{
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 115200;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart1) != HAL_OK)
    {
        Error_Handler();
    }
}
static void MX_DMA_Init(void)
{
    __HAL_RCC_DMA1_CLK_ENABLE();
    HAL_NVIC_SetPriority(DMA1_Channel1_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel1_IRQn);
}

static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3|GPIO_PIN_4, GPIO_PIN_RESET);
    GPIO_InitStruct.Pin = GPIO_PIN_3|GPIO_PIN_4;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_PULLUP;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
}

void pidcompute_type2(double luc){
    e_z= luc - Vloadcell;
    viphant = (e_z-e_z1)/T;

    if(e_z<0.005) tichphan = tichphan_pre + e_z*T;
    else if(e_z>-0.005) tichphan = tichphan_pre + e_z*T;
    else tichphan=0;
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
u_z = Kp*e_z + Ki*tichphan + Kd*viphan;
if(u_z>599) u_z=599;
if(u_z<-599) u_z=-599;

duty=u_z;
e_z1=e_z;
tichphan_pre=tichphan;

if(duty>0){
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_RESET);
    __HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_1,(uint32_t)duty);
}
else
{
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_SET);
    __HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_1,(uint32_t)(-duty));
}

void pidcompute_type1(double luc){

    e_z= luc-Vloadcell ; //cap nhat e(z)
    a_0=Kp+Ki*T/2+Kd/T;
    a_1=-Kp+Ki*T/2-2*Kd/T;
    a_2=Kd/T;
    u_z= u_z1+a_0*e_z+a_1*e_z1+a_2*e_z2 ; //cap nhat u(z)

    // khau bao hoa
    if(u_z>599) u_z=599;
    if(u_z<-599) u_z=-599;
    duty=u_z;

    u_z1=u_z;
    e_z2=e_z1;
    e_z1 =e_z;

    if(duty>0){
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_RESET);
        __HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_1,(uint32_t)duty);
    }
    else
    {
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_SET);
        __HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_1,(uint32_t)(-duty));
    }
}

/*dieu khien vong ho*/
void openloopControl(double duty){
    if(duty>0){
        __HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_1,(uint32_t)duty);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_RESET);
    }
    else if(duty<0)
    {
        __HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_1,(uint32_t)(-duty));
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_SET);
    }
}
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
        }
    else{
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_RESET);
    }
}

/*Chuyen doi chuoi string thanh so*/
double str2num(uint8_t *str,int fp,int ep){
    num=0;
    dot=0;
    dotflag=0;
    negative_flag=0;
    for(i=fp;i<ep;i++){
        if(str[i]=='.') dotflag=1;
        if(str[i]=='-') negative_flag=1;
        if(47<str[i]&&str[i]<58) num = num*10 + str[i]-'0';
        if(dotflag==1&&str[i]!='.'&&str[i]!='0') dot++;
    }
    for(i=0;i<dot;i++) num=num/10;
    if(negative_flag==1) num=-num;
    return num;
}

/*Xu ly ngat uart*/
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart){
    if(huart->Instance == USART1){
        if(cmd==0){
            buff[buff_i]=Rxdata[0];
            buff_i++;
            if(Rxdata[0]==10) {
                bufsize=buff_i;
                buff_j=0;
                cmd=1;
            }
        }
    }
    HAL_UART_Receive_IT(&huart1,(uint8_t*)Rxdata,1);
}

/*Bo loc KALMAN*/
double KalmanFilterADC(double read){
    input=read;
    // tinh toan do loi Kalman
    Kalman_gain= e_uocluong/(e_uocluong + e_doluong);
    // cap nhat gia tri output
    output_hientai = output_truoc + Kalman_gain*(input - output_truoc);
    if(output_truoc>output_hientai>0)
        e_uocluong = (1.0 - Kalman_gain)*e_uocluong + (output_truoc-output_hientai)*e_hethong;
    else
        e_uocluong = (1.0 - Kalman_gain)*e_uocluong - (output_truoc-output_hientai)*e_hethong;
    output_truoc=output_hientai;
    return output_hientai;
}

/*doc gia tri dien ap loadcell*/
double docDienAp_loadcell(uint32_t adcvalue,int method){
    // dung bo loc trung binh
    if(method==FIR_filter){
        ring1_sum=ring1_sum-adc1_ring_buff[ring1_tail];
        ring1_head=ring1_tail;
        adc1_ring_buff[ring1_head]=adc_value[1]*3.3/4095;
        ring1_sum=ring1_sum+adc1_ring_buff[ring1_head];
        ring1_tail--;
    }
}
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
if(ring1_tail<0) ring1_tail=ringsize-1;
    return ring1_sum/ringsize;
}
// dung bo loc KalMan
if(method==Kalman_filter){
    return KalmanFilterADCadcvalue*3.3/4095);
}
}

/*tinh gia tri khoang cach*/
double docKhoangCach(uint32_t encval){
/*tinh khoang cach*/
    double vitri= encval*(18.5*3.141593)/8000;
    if(vitri>110) vitri = 0;
    return vitri;
}

double MophongLoXo(double a, double b, double vitri){
    double luc = a*vitri+b;
    return luc;
}

double LinearControl(double a, double b, double vitri){
    return (3.28-(a*vitri+b))*972.7-1683.1;
}

/*Doc gia tri dien ap dien tro shunt*/
double docDienAp_Rshunt(uint32_t adcvalue){
    ring0_sum=ring0_sum-adc0_ring_buff[ring0_tail];
    ring0_head=ring0_tail;
    adc0_ring_buff[ring0_head]=adcvalue*3.3/4095;
    ring0_sum=ring0_sum+adc0_ring_buff[ring0_head];
    ring0_tail--;
    if(ring0_tail<0) ring0_tail=ringsize -1;
    return ring0_sum/ringsize;
}

/*xu ly packet nhan tu may tinh*/
void readCMDPacket(){
    // cap nhiet gia tri duty
    if(buff[0]=='D'){
        so=str2num((uint8_t *)buff,1,buffsize);
        duty=so;
    }
    // cap nhiet luc dat
    if(buff[0]=='F'){
        lucdat=str2num((uint8_t *)buff,1,buffsize);
    }
    if(buff[0]=='I'){
        control_mode=linear_mode;
        int a=0,b=0;
        for(int i=2;i<buffsize;i++){
            if(buff[i]=='a') a=i;
            if(buff[i]=='b') b=i;
        }
        heso_a=str2num((uint8_t *)buff,a+1,b);
        heso_b=str2num((uint8_t *)buff,b+1,buffsize);
    }
    // cap nhiet he so phuong trinh mo phong lo xo
    if(buff[0]=='L'){
        control_mode=Mohpongloxo_mode;
        int a=0,b=0;
        for(int i=2;i<buffsize;i++){
    
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
if(buff[i]=='a') a=i;
if(buff[i]=='b') b=i;
}
heso_a=str2num((uint8_t *)buff,a+1,b);
heso_b=str2num((uint8_t *)buff,b+1,buffsize);
}
// cap nhat he so kalman filter
if(buff[0]=='K'){
method=Kalman_filter;
int edl=0,eht=0;
for(int i=2;i<buffsize;i++){
if(buff[i]=='r') edl=i;
if(buff[i]=='q') eht=i;
}
e_doluong=str2num((uint8_t *)buff,edl+1,eht);
e_hethong=str2num((uint8_t *)buff,eht+1,buffsize);
}
// cap nhat he so FIR filter
if(buff[0]=='f'){
method=FIR_filter;
}
// cap nhat he so kpid
if(buff[0]=='K'){
e_z=0;
e_z1=0;
e_z2=0;
a_0=0;
a_1=0;
a_2=0;
u_z=0;
u_z1=0;
tichphan = 0;
tichphan_pre = 0;
viphap = 0;
control_mode=PID_mode;
int ipos=0,dpos=0;
for(int i=2;i<buffsize;i++){
if(buff[i]=='i') ipos=i;
if(buff[i]=='d') dpos=i;
}
if(ipos!=0) Kp=str2num((uint8_t *)buff,2,ipos);
if(dpos!=0) {
Ki=str2num((uint8_t *)buff,ipos+1,dpos);
Kd=str2num((uint8_t *)buff,dpos+1,buffsize);
}
buff[0]=0;
}
// cap nhat che do dk
if(buff[0]=='M'){
e_z=0;
e_z1=0;
e_z2=0;
a_0=0;
a_1=0;
a_2=0;
u_z=0;
u_z1=0;
tichphan = 0;
tichphan_pre = 0;
viphap = 0;
if(buff[1]=='1') control_mode = PID_mode;
if(buff[1]=='0') control_mode = OpenLoop_mode;
buff[0]=0;
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
}

}

void Error_Handler(void)
{
}

#ifndef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line)
{
}
#endif
```

Chương trình ngắn: stm32f1xx_it.h

```
#include "main.h"
#include "stm32f1xx_it.h"
extern DMA_HandleTypeDef hdma_adc1;
extern TIM_HandleTypeDef htim3;
extern UART_HandleTypeDef huart1;
extern int timerFlag;

void SysTick_Handler(void)
{
    HAL_IncTick();
}

void DMA1_Channel1_IRQHandler(void)
{
    HAL_DMA_IRQHandler(&hdma_adc1);
}

void TIM3_IRQHandler(void)
{
    HAL_TIM_IRQHandler(&htim3);
    timerFlag=1;
}

void USART1_IRQHandler(void)
{
    HAL_UART_IRQHandler(&huart1);
}
```

Khởi tạo chức năng ngoại vi:stm32f1xx_msp.h

```
#include "main.h"
extern DMA_HandleTypeDef hdma_adc1;
void HAL_TIM_MspPostInit(TIM_HandleTypeDef *htim);

void HAL_MspInit(void)
{
    __HAL_RCC_AFIO_CLK_ENABLE();
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_AFIO_REMAP_SWJ_NOJTAG();

}

void HAL_ADC_MspInit(ADC_HandleTypeDef* hadc)
{
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
GPIO_InitTypeDef GPIO_InitStruct = {0};
if(hadc->Instance==ADC1)
{
    __HAL_RCC_ADC1_CLK_ENABLE();

    __HAL_RCC_GPIOA_CLK_ENABLE();
    GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_1;
    GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
    hdma_adc1.Instance = DMA1_Channel1;
    hdma_adc1.Init.Direction = DMA_PERIPH_TO_MEMORY;
    hdma_adc1.Init.PeriphInc = DMA_PINC_DISABLE;
    hdma_adc1.Init.MemInc = DMA_MINC_ENABLE;
    hdma_adc1.InitPeriphDataAlignment = DMA_PDATAALIGN_HALFWORD;
    hdma_adc1.Init.MemDataAlignment = DMA_MDATAALIGN_HALFWORD;
    hdma_adc1.Init.Mode = DMA_CIRCULAR;
    hdma_adc1.Init.Priority = DMA_PRIORITY_LOW;
    if (HAL_DMA_Init(&hdma_adc1) != HAL_OK)
    {
        Error_Handler();
    }

    __HAL_LINKDMA(hadc,DMA_Handle,hdma_adc1);
}

void HAL_ADC_MspInit(ADC_HandleTypeDef* hadc)
{
    if(hadc->Instance==ADC1)
    {
        __HAL_RCC_ADC1_CLK_DISABLE();
        HAL_GPIO_DeInit(GPIOA, GPIO_PIN_0|GPIO_PIN_1);
        HAL_DMA_DeInit(hadc->DMA_Handle);
    }
}

void HAL_TIM_Encoder_MspInit(TIM_HandleTypeDef* htim_encoder)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    if(htim_encoder->Instance==TIM1)
    {
        __HAL_RCC_TIM1_CLK_ENABLE();

        __HAL_RCC_GPIOA_CLK_ENABLE();
        GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9;
        GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
        GPIO_InitStruct.Pull = GPIO_PULLUP;
        HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
    }
}

void HAL_TIM_PWM_MspInit(TIM_HandleTypeDef* htim_pwm)
{
    if(htim_pwm->Instance==TIM2)
    {
        __HAL_RCC_TIM2_CLK_ENABLE();
    }
}
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
void HAL_TIM_Base_MspInit(TIM_HandleTypeDef* htim_base)
{
    if(htim_base->Instance==TIM3)
    {
        __HAL_RCC_TIM3_CLK_ENABLE();
        HAL_NVIC_SetPriority(TIM3_IRQn, 0, 0);
        HAL_NVIC_EnableIRQ(TIM3_IRQn);
    }

}

void HAL_TIM_MspPostInit(TIM_HandleTypeDef* htim)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    if(htim->Instance==TIM2)
    {
        __HAL_RCC_GPIOA_CLK_ENABLE();
        GPIO_InitStruct.Pin = GPIO_PIN_15;
        GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
        GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
        HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

        __HAL_AFIO_REMAP_TIM2_PARTIAL_1();
    }

}

void HAL_TIM_Encoder_MspDeInit(TIM_HandleTypeDef* htim_encoder)
{
    if(htim_encoder->Instance==TIM1)
    {
        __HAL_RCC_TIM1_CLK_DISABLE();
        HAL_GPIO_DeInit(GPIOA, GPIO_PIN_8|GPIO_PIN_9);
    }

}

void HAL_TIM_PWM_MspDeInit(TIM_HandleTypeDef* htim_pwm)
{
    if(htim_pwm->Instance==TIM2)
    {
        __HAL_RCC_TIM2_CLK_DISABLE();
    }

}

void HAL_TIM_Base_MspDeInit(TIM_HandleTypeDef* htim_base)
{
    if(htim_base->Instance==TIM3)
    {
        __HAL_RCC_TIM3_CLK_DISABLE();
        HAL_NVIC_DisableIRQ(TIM3_IRQn);
    }

}

void HAL_UART_MspInit(UART_HandleTypeDef* huart)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    if(huart->Instance==USART1)
    {
        __HAL_RCC_USART1_CLK_ENABLE();
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
__HAL_RCC_GPIOB_CLK_ENABLE();
GPIO_InitStruct.Pin = GPIO_PIN_6;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

GPIO_InitStruct.Pin = GPIO_PIN_7;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

__HAL_AFIO_REMAP_USART1_ENABLE();
HAL_NVIC_SetPriority(USART1_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(USART1_IRQn);
}

}

void HAL_UART_MspDeInit(UART_HandleTypeDef* huart)
{
if(huart->Instance==USART1)
{
__HAL_RCC_USART1_CLK_DISABLE();

HAL_GPIO_DeInit(GPIOB, GPIO_PIN_6|GPIO_PIN_7);
HAL_NVIC_DisableIRQ(USART1_IRQn);
}
}
```

Phần (Code C#) giao diện máy tính:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.Timers;
using System.IO;

namespace GiaodienHaptics
{

    public partial class mainform : Form
    {
        string data;
        double Vdongco = 0;
        double Vloadcell = 0;
        double Vloadcellnotfill = 0;
        double Iucdo = 0;
        double Kgdo = 0;
        double KhoangCach = 0;
        double Iucdat = 0;
        double dutyread = 0;
        double TrucXvalue = 0.000001;
        double TrucYvalue = 0;
        double TrucX1value = 0.000001;
        double TrucY1value = 0;
        int draw = 0;
        DateTime timestamp = new DateTime();
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
DateTime timeend = new DateTime();
DateTime timestampdraw = new DateTime();
TimeSpan xtimetick = new TimeSpan();
TimeSpan timediff = new TimeSpan();
System.IO.StreamWriter out_file;
System.IO.StreamReader in_file;

public mainform()
{
    InitializeComponent();
    trucX.SelectedItemChecked(3, true);
    bieudo.Series[0].ToolTip = "Y = #VALY\nX = #VALX";

}

private void Form1_Load(object sender, EventArgs e)
{
    //scan port name
    comcbtxt.Items.AddRange(SerialPort.GetPortNames());
}

/*cau hinh va ket noi uart*/
private void comcbtxt_DropDown(object sender, EventArgs e)
{
    comcbtxt.Items.Clear();
    comcbtxt.Items.AddRange(SerialPort.GetPortNames());
}
private void connectbt_Click(object sender, EventArgs e)
{
    if (COM.isOpen == false)
    {
        try
        {
            COM.PortName = comcbtxt.Text;
            COM.BaudRate = int.Parse(baudcbtxt.Text);
            COM.Open();
            connectbt.Text = "disconnect";
            connectbar.Value = 100;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Can not open port:" + comcbtxt.Text + "baudrate: " + baudcbtxt.Text
, MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        finally
        {

        }
    }
    else if (COM.isOpen == true)
    {
        try
        {
            COM.Close();
            connectbar.Value = 0;
            connectbt.Text = "connect";
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "can not close port",
MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
        }
        finally
        {
    
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
        }
    }
}
/*xu ly du lieu gui den com*/
private void COM_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    FileStream dataFile = new FileStream("finalProject.dat", FileMode.OpenOrCreate, FileAccess.ReadWrite,
FileShare.ReadWrite);
    if (COM.IsOpen)
    {
        try
        {
            data = COM.ReadLine();
            timeend = DateTime.UtcNow;
            timeend = DateTime.UtcNow;
            timediff = timeend - timestart;
            timestart = timeend;
            xtimetick = DateTime.UtcNow - timestartdraw;
            string[] variables = data.Split('\n')[0].Split(',');
            Vloadcellnotfill = Convert.ToDouble(variables[0]);
            Vloadcell = Convert.ToDouble(variables[1]);
            KhoangCach = Convert.ToDouble(variables[2]);
            Vdongco = Convert.ToDouble(variables[3]);
            dutyread = Convert.ToDouble(variables[4]);
            kgdo = Vloadcell * 2.3595 - 3.8796;
            lucdo = kgdo*9.81;
            this.BeginInvoke((Action)() =>
            {
                dongco.Text = variables[3];
                loadcell.Text = variables[1];
                vitri.Text = variables[2];
                luc_txt.Text= Convert.ToString(Math.Round(lucdo, 2));
                kg_txt.Text= Convert.ToString(Math.Round(kgdo, 2));
                dutyread_txt.Text = Convert.ToString(Math.Round((dutyread / 599 * 100),1));
                if (trucY.GetItemCheckState(0) == CheckState.Checked )
                {
                    TrucY1value = Vloadcell;
                }
                if (trucY.GetItemCheckState(1) == CheckState.Checked)
                {
                    TrucYvalue = Vloadcellnotfill;
                }
                if (trucY.GetItemCheckState(3) == CheckState.Checked)
                {
                    TrucYvalue = lucdat;
                }
                if (trucY.GetItemCheckState(2) == CheckState.Checked)
                {
                    TrucY1value = lucdo;
                }
                if (trucY.GetItemCheckState(4) == CheckState.Checked)
                {
                    TrucYvalue = KhoangCach;
                }
                //if (trucX.GetItemCheckState(5) == CheckState.Checked)
                //{
                //    TrucXvalue = Vdongco;
                //}
                if (trucX.GetItemCheckState(0) == CheckState.Checked)
                {
                    TrucXvalue = lucdo;
                    TrucX1value = lucdo;
                }
            });
        }
    }
}
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
        }
        if (trucX.GetItemCheckState(1) == CheckState.Checked)
        {
            TrucXvalue = KhoangCach;
            TrucX1value = KhoangCach;
        }
        if (trucX.GetItemCheckState(2) == CheckState.Checked)
        {
            TrucXvalue = Vdongco;
            TrucX1value = Vdongco;
        }
        if (trucX.GetItemCheckState(3) == CheckState.Checked)
        {
            TrucXvalue = xtimetick.TotalSeconds;
            TrucX1value = xtimetick.TotalSeconds;
        }
        if (trucX.GetItemCheckState(4) == CheckState.Checked)
        {
            TrucXvalue = dutyread;
            TrucX1value = dutyread;
        }

        if (draw == 1)
        {
            RXdata_richtxt.AppendText(Vloadcell + " " + KhoangCach + " " + Vdongco+" "+dutyread+" at: "+
DateTime.Now.TimeOfDay+"\n");
            bieudo.Series[0].Points.AddXY(TrucXvalue, TrucYvalue);
            bieudo.Series[1].Points.AddXY(TrucX1value, TrucY1value);
        }
        dataupInterval_txt.Text = Convert.ToString(timediff.TotalMilliseconds);
    }));
}
catch(Exception ex)
{
}
}
}
}

/*Vélf Biê)%ou Å'ô)%*/
// bÄfât Å'ô)%u velf
private void draw_Click(object sender, EventArgs e)
{
    if (draw == 0)
    {
        timestamp = DateTime.UtcNow;
        timeend = DateTime.UtcNow;
        timestampdraw = DateTime.UtcNow;
        drawbt.Text = "stop draw";
        draw = 1;
    }
    else
    {
        drawbt.Text = "draw";
        draw = 0;
    }
}

//Chol%en trù)%c y vélf biê)%ou Å'ô)%
private void trucY_SelectedIndexChanged(object sender, EventArgs e)
{
    if (trucY.CheckedItems.Count < 3)
    {
        if (trucY.GetItemCheckState(trucY.SelectedIndex) == CheckState.Checked)
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
{  
    if (trucY.SelectedIndex == 0 || trucY.SelectedIndex == 2 )  
    {  
        bieudo.Series[1].Name = trucY.Items[trucY.SelectedIndex].ToString();  
    }  
    if (trucY.SelectedIndex == 1 || trucY.SelectedIndex == 3 || trucY.SelectedIndex == 4 || trucY.SelectedIndex  
== 5)  
    {  
        bieudo.Series[0].Name = trucY.Items[trucY.SelectedIndex].ToString();  
    }  
}  
else  
{  
    trucY.SetItemCheckState(trucY.SelectedIndex, CheckState.Unchecked);  
}  
}  
}  
//Chọn trục X và nếu bieber%ou Á'ô%  
private void trucX_SelectedIndexChanged(object sender, EventArgs e)  
{  
    if (trucX.GetItemCheckState(trucX.SelectedIndex) == CheckState.Checked)  
    {  
        bieudo.ChartAreas[0].AxisX.Title = trucX.Items[trucX.SelectedIndex].ToString();  
        for(int i = 0; i< 4; i++)  
        {  
            if (i != trucX.SelectedIndex)  
            {  
                trucX.SetItemCheckState(i, CheckState.Unchecked);  
            }  
        }  
    }  
}  
//hiển thị tọa độ điểm khi rê chuột  
private void bieudo_MouseMove(object sender, MouseEventArgs e)  
{  
    try  
    {  
        bieudo.Series[0].ToolTip = "Y = #VALY\nX = #VALX";  
        bieudo.Series[1].ToolTip = "Y = #VALY\nX = #VALX";  
    }  
    catch (Exception ex)  
    {  
    }  
}  
}  
//thu голen bieber%ou Á'ô%  
private void scrollbieudo_check_CheckedChanged(object sender, EventArgs e)  
{  
    if (scrollbieudo_check.Checked)  
    {  
        bieudo.ChartAreas[0].CursorX.AutoScroll = true;  
        bieudo.ChartAreas[0].AxisX.ScaleView.Zoomable = true;  
        bieudo.ChartAreas[0].AxisX.Interval = 5;  
        bieudo.ChartAreas[0].AxisX.ScaleView.Zoom(xtimetick.TotalMilliseconds, 10000);  
        bieudo.ChartAreas[0].AxisX.ScaleView.SmallScrollSize = 10;  
    }  
    else  
    {  
        bieudo.ChartAreas[0].CursorX.AutoScroll = false;  
    }  
}  
}  
//xóa bieber%ou Á'ô%  
private void clear_Click(object sender, EventArgs e)
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
{  
    bieudo.Series[0].Points.Clear();  
    bieudo.Series[1].Points.Clear();  
    bieudo.Series[0].Name = "nodata1";  
    bieudo.Series[1].Name = "nodata2";  
}  
/*XÆ°ì%o lyù khung data log*/  
private void cleardatalog_bt_Click(object sender, EventArgs e)  
{  
    try  
    {  
        RXdata_richtxt.Clear();  
    }  
    catch(Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
}  
/*Truyềñ cài ñc thông số*/  
// chon che do dieu khien PID  
private void pidmodebt_Click(object sender, EventArgs e)  
{  
    if (COM.IsOpen)  
    {  
        try  
        {  
            COM.WriteLine("M1" + dutytxt.Text);  
        }  
        catch {}  
    }  
}  
// chon che do voi ñng ho  
private void vonghomode_Click(object sender, EventArgs e)  
{  
    if (COM.IsOpen)  
    {  
        try  
        {  
            COM.WriteLine("M0");  
        }  
        catch {}  
    }  
}  
// GÆ°ì%oi BôùE loùEc trung biùnh  
private void FIRset_bt_Click(object sender, EventArgs e)  
{  
    if (COM.IsOpen)  
    {  
        try  
        {  
            COM.WriteLine("f");  
        }  
        catch {}  
    }  
}  
// Gui%oi thong soùBôùE loùEc kalman  
private void setKALMAN_bt_Click(object sender, EventArgs e)  
{  
    if (COM.IsOpen)  
    {  
        try  
        {  
            COM.WriteLine("k" + "r" + edoluong_txt.Text + "q" + ehetrong_Txt.Text);  
        }  
    }  
}
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
        }
        catch {}
    }
}
// Gửi giá trị lực
private void Fset_bt_Click(object sender, EventArgs e)
{
    if (COM.IsOpen)
    {
        try
        {
            //lucdo = Vloadcell * 2.3595 - 3.8796;
            double Fset = (Convert.ToDouble(Fdattxt.Text)/9.81 + 3.8796) / 2.3595;
            COM.WriteLine("F" + Fset);
            lucdat = Convert.ToDouble(Fdattxt.Text);
        }
        catch {}
    }
}
// Gửi thông số PID
private void setpidbt_Click(object sender, EventArgs e)
{
    if (COM.IsOpen)
    {
        try
        {
            if (kptxt.Text != "" && kitxt.Text != "" && kdtxt.Text != "")
                COM.WriteLine("Kp" + kptxt.Text + "I" + kitxt.Text + "D" + kdtxt.Text);
        }
        catch {}
    }
}
// Gửi ông xung duty
private void setdutybt_Click(object sender, EventArgs e)
{
    Changedutyvalue();
}
private void Changedutyvalue()
{
    if (roll_rdbt.Checked)
    {
        dutybar.Value = int.Parse(dutytxt.Text);
    }
    if (COM.IsOpen && click_rdbt.Checked)
    {
        try
        {
            if (dutytxt.Text != "")
                COM.WriteLine("D" + dutytxt.Text);
        }
        catch {}
    }
}
private void dutytxt_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar==(char)Keys.Enter)
    {
        Changedutyvalue();
    }
}
// cập nhật duty
private void dutybar_ValueChanged(object sender, EventArgs e)
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
{  
    if (COM.IsOpen && roll_rdbt.Checked)  
    {  
        try  
        {  
            dutytxt.Text = dutybar.Value.ToString();  
  
            COM.WriteLine("D" + dutytxt.Text);  
        }  
        catch (Exception ex)  
        {  
            MessageBox.Show(ex.Message);  
        }  
    }  
}  
  
private void button2_Click(object sender, EventArgs e)  
{  
    TextWriter writer = new StreamWriter(@"C:\Users\win10\Desktop\datalog\data.dat");  
    writer.WriteLine(RXdata_richtxt.Text);  
    writer.Close();  
}  
  
private void mophongloxbt_Click(object sender, EventArgs e)  
{  
    if (COM.IsOpen)  
    {  
        try  
        {  
            COM.WriteLine("La" + hesoAtxt.Text + "b" + hesoBtxt.Text);  
        }  
        catch (Exception ex)  
        {  
  
        }  
    }  
}  
  
private void button3_Click(object sender, EventArgs e)  
{  
    if (COM.IsOpen)  
    {  
        try  
        {  
            COM.WriteLine("la" + hesoAtxt.Text + "b" + hesoBtxt.Text);  
        }  
        catch (Exception ex)  
        {  
  
        }  
    }  
}
```

Phần (Code MATLAB) quy hồi tuyến tính (RL)

x = xlsread("Calibloadcell.xlsx", "A1:A2688");

y = xlsread("Calibloadcell.xlsx", "D1:D2688");

xThuan = xlsread("Thuan.xlsx", "A1:A179");

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
yThuan = xlsread("Thuan.xlsx", "D1:D179");
```

```
xNghich = xlsread("nghich.xlsx", "A1:A132");
```

```
yNghich = xlsread("nghich.xlsx", "D1:D132");
```

```
subplot(2, 2, 1);
```

```
n = numel(x);
```

```
sx = sum(x);
```

```
sy = sum(y);
```

```
sxy = sum(x.*y);
```

```
sxx = sum(power(x, 2));
```

```
B = (n*sxy - sx*sy)/(n*sxx- sx*sx);
```

```
A = (1/n)*sy-B*1/n*sx;
```

```
t = B*x+A;
```

```
plot(x, t, 'r', x, y, 'g');
```

```
title("Gia tri tong the");
```

```
xlabel('Vloacell (v)');
```

```
ylabel('duty (xung)');
```

```
legend('Linear', 'tin hieu thuc');
```

```
grid on;
```

```
subplot(2, 2, 2);
```

```
nThuan = numel(xThuan);
```

```
sx = sum(xThuan);
```

```
sy = sum(yThuan);
```

```
sxy = sum(xThuan.*yThuan);
```

```
sxx = sum(power(xThuan, 2));
```

```
BThuan = (nThuan*sxy - sx*sy)/(nThuan*sxx- sx*sx);
```

```
AThuan = (1/nThuan)*sy-BThuan*1/nThuan*sx;
```

```
tThuan = BThuan*xThuan+AThuan;
```

```
plot(xThuan, tThuan, 'r', xThuan, yThuan, 'g');
```

Chế tạo và điều khiển thiết bị haptic cảm nhận lực tương tác với vật thể ảo

```
title("Gia tri dong co quay thuan");
xlabel('Vloacell (v)');
ylabel('duty (xung)');
legend('Linear thuan', 'tin hieu thuc');
grid on;

subplot(2, 2, 3);
nNghich = numel(xNghich);
sx = sum(xNghich);
sy = sum(yNghich);
sxy = sum(xNghich.*yNghich);
sxx = sum(power(xNghich, 2));
BNghich = (nNghich*sxy - sx*sy)/(nNghich*sxx- sx*sx);
ANghich = (1/nNghich)*sy-BNghich*1/nNghich*sx;
tNghich = BNghich*xNghich+ANghich;
plot(xNghich, tNghich, 'r', xNghich, yNghich, 'g');
title("Gia tri dong co quay nghich");
xlabel('Vloacell (v)');
ylabel('duty (xung)');
legend('Linear nghich', 'tin hieu thuc');
grid on;

subplot(2, 2, 4);
tNghich = BNghich*x+ANghich;
tThuan = BThuan*x+AThuan;
plot(x, tNghich, 'r', x, tThuan, 'g', x, t, 'b');
title("So sanh do thi dong co quay thuan, nghich, tong the");
xlabel('Vloacell (v)');
ylabel('duty (xung)');
legend('Linear nghich','Linear thuan' , 'tin hieu thuc');
grid on;
```

Thông số, datasheet thiết bị

Thông số cụ thể cân dùng calib loadcell:

<https://www.made-in-china.com/showroom/melanshalu/product-detailRvwJIFgbLyhU/China-Weiheng-A11-Mini-Retail-Digital-Weight-Scale-25kg-Mini-Scale.html>

Phụ lục tham khảo datasheet STM32F103C6T8:

<https://datasheet.octopart.com/STM32F103C8T6-STMMicroelectronics-datasheet-14106722.pdf>

Phụ lục tham khảo datasheet động cơ:

<http://www.dynetics.eu/media/1229/dme44.pdf>

Phụ lục tham khảo datasheet loadcell:

<https://hshop.vn/products/cam-bien-loadcell-1kg>

Phụ lục tham khảo datasheet encoder:

https://www.ia.omron.com/data_pdf/cat/e6b2-c_ds_e_6_1_csm491.pdf?id=487

Phụ lục tham khảo datasheet INA128:

<http://www.ti.com/lit/gpn/ina128>

Phụ lục tham khảo datasheet USB-UART:

http://www.prolific.com.tw/UserFiles/files/ds_pl2303HXD_v1_4_4.pdf

Phụ lục tham khảo datasheet mạch cầu h VNH2SP30 30A:

<http://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/10832.pdf>

Phụ lục tham khảo datasheet OPAMP LM358:

<http://www.ti.com/lit/gpn/lm358-n>

Phụ lục tham khảo datasheet nguồn VSF30-DD:

http://kuds.co.kr/data/kud_Orientelectronics_kor-.pdf