

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



ĐÒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ KỸ THUẬT Ô TÔ

NGHIÊN CỨU, LẬP TRÌNH VẬN HÀNH MÔ HÌNH
ĐỘNG CƠ PMSM ACMC CỦA DSPACE

GVHD: TH.S. NGUYỄN TRUNG HIẾU
SVTH: HOÀNG KHẢI
ĐỖ THÀNH LONG



Tp. Hồ Chí Minh, tháng 6 năm 2024

TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT
TP. HỒ CHÍ MINH
KHOA CƠ KHÍ ĐỘNG LỰC

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập – Tự Do – Hạnh Phúc

TP. Hồ Chí Minh, ngày 23 tháng 02 năm 2024

NHIỆM VỤ ĐO ÁN TỐT NGHIỆP

Họ tên sinh viên: 1. Hoàng Khải

MSSV: 20145159

(E-mail: 20145159@student.hcmute.edu.vn Điện thoại: 0919632036)

2. Đỗ Thành Long

MSSV: 20145705

(E-mail: 20145705@student.hcmute.edu.vn Điện thoại: 0981067357)

Ngành: Công nghệ Kỹ thuật ô tô

Khóa: 2020

Lớp: 201452

1. **Tên đề tài:** Nghiên cứu, lập trình vận hành mô hình động cơ PMSM AC/DC của dSPACE

2. **Nhiệm vụ đề tài**

Hiểu rõ phương pháp Model-Based Design
Có kiến thức về các phần mềm phần cứng hãng dSPACE
Thiết kế, chế tạo module (FPGA) của MicroLabBox
Vận hành mô hình PMSM AC/DC

3. Sản phẩm của đề tài

Module FPGA của MicroLabBox
Mô hình PMSM AC/DC

4. Ngày giao nhiệm vụ đề tài: 23/02/2024

5. Ngày hoàn thành nhiệm vụ: 10/6/2024

TRƯỞNG BỘ MÔN

Lê Quang Vũ

CÁN BỘ HƯỚNG DẪN

Bộ môn: Điện tử ô tô

PHIẾU NHẬN XÉT ĐỘ ÁN TỐT NGHIỆP

(Dành cho giảng viên hướng dẫn)

Họ và tên sinh viên: Hoàng Khải

MSSV: 20145159

Họ và tên sinh viên: Hồ Thành Lực

MSSV: 20145703

Tên đề tài: Nghiên cứu, lập trình vận hành mô hình động cơ PMSM ACMC của dSPACE

Ngành đào tạo: Công nghệ Kỹ thuật ô tô

Họ và tên GV hướng dẫn: Nguyễn Trung Hiếu

Ý KIẾN NHẬN XÉT

1. Nhận xét về tinh thần, thái độ làm việc của sinh viên

Cả 02 sinh viên đã nỗ lực, chăm chỉ và cẩn thận trong quá trình thực hiện đề án, làm việc với tinh thần chủ động và cầu tiến.

2. Nhận xét về kết quả thực hiện của DATN

2.1. Kết cấu, cách thức trình bày DATN:

Trình bày sạch sẽ, rõ ràng bao gồm chất lượng và theo mẫu của khoa.

2.2. Nội dung đồ án:

(Cơ sở lý luận, tính thực tiễn và khả năng ứng dụng của đồ án, các hướng nghiên cứu có thể tiếp tục phát triển)

Đồ án mang tính mở rộng việc khai thác và sử dụng mô hình ACMC để nghiên cứu quan tâm chuyên ngành. Văn bản bài báo có lối viết nghiêm túc, có khung định hướng rõ ràng. Phép tu từ phàn áy noli rõ ràng, dễ hiểu.

2.3. Kết quả đạt được:

- 01 thuyết trình
- 01 Dashboard điều khiển ACMC
- 01 mجلد nêu rõ

2.4. Những tồn tại (nếu có).

.....
.....
.....

3. Dánh giá:

TT	Mục đánh giá	Điểm tối đa	Điểm đạt được
1.	Hình thức và kết cấu DATN	30	26
	Dung format với đầy đủ cả hình thức và nội dung của các mục	10	10
	Mục tiêu, nhiệm vụ, tổng quan của đề tài	10	10
	Tính cấp thiết của đề tài	10	10
2.	Nội dung DATN	50	45
	Khả năng ứng dụng kiến thức toán học, khoa học và kỹ thuật, khu vực xã hội...	5	4
	Khả năng thực hiện/phân tích/tổng hợp/danh giá	10	9
	Khả năng thiết kế chế tạo một hệ thống, thành phần, hoặc quy trình đáp ứng yêu cầu đưa ra với những ràng buộc thực tế.	15	14
	Khả năng cải tiến và phát triển	15	14
	Khả năng sử dụng công cụ kỹ thuật, phần mềm chuyên ngành...	5	4
3.	Dánh giá về khả năng ứng dụng của đề tài	10	10
4.	Sản phẩm cụ thể của DATN	10	10
	Tổng điểm	100	95

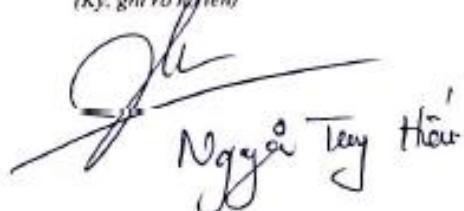
4. Kết luận:

- Được phép bảo vệ
 Không được phép bảo vệ

TP.HCM, ngày tháng 06 năm 2024

Giảng viên hướng dẫn

(Ký, ghi rõ họ tên)



Nguyễn Tuy Thieu

Bộ môn: Điện tử ô tô

PHIẾU NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

(Dành cho giảng viên phản biện)

Họ và tên sinh viên: Hoàng Khải

MSSV: 20145159

Họ và tên sinh viên: Đỗ Thành Long

MSSV: 20145705

Tên đề tài: Nghiên cứu, lập trình vận hành mô hình động cơ PMSM AC/DC của dSPACE

Ngành đào tạo: Công nghệ Kỹ thuật ô tô

Họ và tên GV phản biện (Mã GV): Vũ Đình Huân (2961)

Ý KIẾN NHẬN XÉT

1. Kết cấu, cách thức trình bày ĐATN:

.....Trình bày rõ ràng, khoa học, minh mẫn.....
.....
.....

2. Nội dung đồ án:

(Cơ sở lý luận, tính thực tiễn và khả năng ứng dụng của đồ án, các hướng nghiên cứu có thể tiếp tục phát triển)

.....Nội dung cơ bản lý thuyết đầy đủ.....
.....Khả năng ứng dụng cao.....
.....

3. Kết quả đạt được:

.....Nhóm đã hoàn thành đề tài trong thời gian.....
.....
.....

4. Những thiếu sót và tồn tại của ĐATN:

.....
.....
.....

5. Câu hỏi

.....
.....
.....
.....
.....
.....
.....

6. Đánh giá:

TT	Mục đánh giá	Điểm tối đa	Điểm đạt được
1.	Hình thức và kết cấu ĐATN	30	30
	Đúng format với đầy đủ cả hình thức và nội dung của các mục	10	
	Mục tiêu, nhiệm vụ, tổng quan của đề tài	10	
	Tính cấp thiết của đề tài	10	
2.	Nội dung ĐATN	50	48
	Khả năng ứng dụng kiến thức toán học, khoa học và kỹ thuật, khoa học xã hội...	5	
	Khả năng thực hiện/phân tích/tổng hợp/danh giá	10	
	Khả năng thiết kế chế tạo một hệ thống, thành phần, hoặc quy trình đáp ứng yêu cầu đưa ra với những ràng buộc thực tế.	15	
	Khả năng cải tiến và phát triển	15	
	Khả năng sử dụng công cụ kỹ thuật, phần mềm chuyên ngành...	5	
3.	Đánh giá về khả năng ứng dụng của đề tài	10	10
4.	Sản phẩm cụ thể của ĐATN	10	6
	Tổng điểm	100	94

7. Kết luận:

- Được phép bảo vệ
 Không được phép bảo vệ

TP.HCM, ngày tháng 06 năm 2024

Giảng viên phản biện

(Ký, ghi rõ họ tên)


Vu Anh Huan

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ
THUẬT TP. HCM
KHOA CƠ KHÍ ĐỘNG LỰC

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

Bộ môn: Điện tử Ô tô

XÁC NHẬN HOÀN THÀNH ĐỒ ÁN

Tên đề tài: NGHIÊN CỨU, LẬP TRÌNH VẬN HÀNH MÔ HÌNH ĐỘNG CƠ PMSM
ACMC CỦA DSPACE

Họ và tên Sinh viên: Hoàng Khải MSSV: 20145159

Họ và tên Sinh viên: Đỗ Thành Long MSSV: 20145705

Ngành: Công nghệ Kỹ thuật ô tô

Sau khi tiếp thu và điều chỉnh theo góp ý của GV hướng dẫn, GV phản biện và các thành viên trong Hội đồng bảo vệ. Em xin cam kết và chịu trách nhiệm về nội dung và hình thức Đồ án tốt nghiệp đã được hoàn chỉnh đúng các quy định và theo yêu cầu.

Chủ tịch Hội đồng:

Vũ Đình Huân

Đỗ Thành Long

GV hướng dẫn:

NGUYỄN TRUNG HIẾU

Đỗ Thành Long

GV phản biện:

Vũ Đình Huân

Đỗ Thành Long

Tp. Hồ Chí Minh, ngày tháng 6 năm 2024

LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn chân thành nhất đến gia đình, thầy hướng dẫn và bạn bè đã đồng hành và hỗ trợ tôi trong suốt quá trình thực hiện đồ án tốt nghiệp này. Đồ án sẽ không có được thành công này nếu thiếu đi sự giúp đỡ cũng như động viên từ mọi người.

Đầu tiên, tôi cũng muốn gửi lời cảm ơn đến gia đình đã luôn bên cạnh, động viên và hỗ trợ tôi suốt quá trình học tập. Sự ủng hộ không điều kiện của họ là động lực giúp tôi vực qua những thử thách và đạt được mục tiêu của bản thân.

Tiếp đến, tôi muốn bày tỏ lòng biết ơn sâu sắc tới Thầy Nguyễn Trung Hiếu đã dành nhiều thời gian, tâm huyết và sự hỗ trợ tận tình để chỉ dẫn tôi qua mỗi bước của đồ án. Những lời khuyên vô cùng quý báu và sự chỉ dẫn không ngừng nghỉ của Thầy đã giúp tôi ngày càng trưởng thành, phát triển cũng như hoàn thiện bản thân.

Cuối cùng, tôi muốn cảm ơn các thành viên trong Lab AML đã tạo cho tôi một môi trường làm việc cũng như điều kiện tốt nhất để tôi có thể thực hiện đồ án này một cách suôn sẻ và hoàn thiện nhất.

Lời cảm ơn này không thể diễn tả hết những sự biết ơn của tôi đến tất cả mọi người. Tôi sẽ luôn ghi nhớ những đóng góp và sự hỗ trợ quý báu của mọi người trong suốt cuộc hành trình sau này của bản thân.

TÓM TẮT

Bài nghiên cứu này tập trung về việc nghiên cứu phần cứng dSPACE như MicroLabBox, RapidPro Power Unit trong việc ứng dụng phương pháp Model-based Design và nghiên cứu mô hình động cơ PMSM ACMC của dSPACE, sử dụng thuật toán điều khiển có sẵn để vận hành cho mô hình sử dụng các phần mềm Matlab/Simulink và dSPACE. Mô hình này mô tả cụ thể về quá trình thực hiện quy trình Rapid Control Prototyping (RCP), một phần trong phương pháp Model-Based Design. Cụ thể, mô hình bao gồm 3 phần: **động cơ PMSM, MicroLabBox, RapidPro Power Unit**. RapidPro Power Unit đóng vai trò như một module điều khiển động cơ, cần phải sử dụng phần mềm để cấu hình cho RapidPro trở thành full-bridge, half-bridge, low side hoặc high side tùy thuộc vào nhu cầu sử dụng. Trong khi đó, MicroLabBox lại có vai trò như một con microcontroller (vi điều khiển) để xử lý tín hiệu từ cảm biến và xuất tín hiệu PWM để điều khiển động cơ PMSM. Sau khi nghiên cứu kĩ mô hình, tiến hành thiết kế giao diện điều khiển trên phần mềm ControlDesk và chạy thực nghiệm, ghi nhận kết quả thực nghiệm, từ đó đưa ra các nhận xét và đánh giá về mô hình. Ngoài ra, khai thác thêm một số chức năng cơ bản của MicroLabBox như: đầu vào, đầu ra tín hiệu dạng số, analog bằng cách chế tạo MicroLabBox GPIO Module, mở rộng các chức năng. Cuối cùng, kết luận và đưa ra hướng phát triển cho tương lai.

MỤC LỤC

LỜI CẢM ƠN.....	i
TÓM TẮT.....	ii
MỤC LỤC.....	iii
Danh mục các chữ viết tắt và kí hiệu	vi
Danh mục các hình.....	vii
Danh mục các bảng.....	xii
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục đích của đề tài	2
1.3. Phương pháp nghiên cứu	2
1.4. Đối tượng nghiên cứu	2
1.4.1. Phạm vi nghiên cứu	2
1.4.2. Bố cục đồ án	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	4
2.1. Tổng quan về động cơ điện.....	4
2.1.1. Giới thiệu động cơ điện	4
2.1.2. Cấu tạo và nguyên lí hoạt động chung.....	5
2.1.3. Phân loại động cơ điện.....	6
2.2. Động cơ điện PMSM	7
2.2.1. Cấu tạo	7
2.2.1.1. Rotor	8
2.2.1.2. Stator	8
2.2.2. Nguyên lí hoạt động.....	9
2.2.3. Ưu điểm và nhược điểm.....	10
2.2.3.1. Ưu điểm	10
2.2.3.2. Nhược điểm.....	10
2.2.4. Ứng dụng	10

2.3.	Tổng quan về phương pháp Model-Based Design (MBD).....	11
2.3.1.	Khái niệm.....	11
2.3.1.1.	Ưu điểm của MBD.....	13
2.3.1.2.	Nhược điểm của MBD.....	14
2.3.1.3.	Quy trình làm việc của MBD.....	14
2.3.2.	Yêu cầu hệ thống (System Requirements).....	15
2.3.3.	Cấu trúc hệ thống (System Architecture)	16
2.3.4.	Thiết kế hệ thống (System Design).....	16
2.3.5.	Mô phỏng và triển khai mô hình (Model Simulation and Implementation)..	16
2.3.6.	Xác nhận và đánh giá (Verification & Validation).....	17
2.3.6.1.	Model-in-the-Loop Simulation (MIL) Testing	17
2.3.6.2.	Software-in-the-Loop Simulation (SIL) Testing	18
2.3.6.3.	Processor-in-the-Loop Simulation (PIL) Testing	19
2.3.6.4.	Rapid Control Prototyping (RCP) Testing	20
2.3.6.5.	Hardware-in-the-Loop Simulation (HIL) Testing	21
2.3.7.	Sản xuất (Production)	22

CHƯƠNG 3: MÔ HÌNH PMSM ACMC CỦA DSPACE23

3.1.	Tổng quan về Matlab/Simulink	23
3.2.	Tổng quan về dSPACE Software.....	25
3.2.1.	Real-time Interface (RTI)	25
3.2.2.	ControlDesk	26
3.2.3.	ConfigurationDesk for RapidPro	31
3.3.	Tổng quan về dSPACE MicroLabBox và RapidPro System.....	33
3.3.1.	MicroLabBox.....	33
3.3.1.1.	Tổng quan MicroLabBox.....	34
3.3.1.2.	Mặt trước MicroLabBox.....	37
3.3.1.3.	Mặt sau MicroLabBox	45
3.3.1.4.	Kết nối MicroLabBox với Host PC	47
3.3.2.	RapidPro System.....	49
3.3.2.1.	Giới thiệu RapidPro System	49
3.3.2.2.	RapidPro Power Unit	54

3.3.2.3. PS-HCHBD 2/2 Module	65
3.4. Mô hình động cơ PMSM ACMC của dSPACE	74
3.4.1. Tổng quan về mô hình PMSM ACMC	74
3.4.2. Sơ đồ nối dây mô hình	77
CHƯƠNG 4: VẬN HÀNH MÔ HÌNH PMSM ACMC CỦA DSPACE.....	89
4.1. Nghiên cứu, chế tạo module mở rộng tín hiệu digital và analog của MicroLabBox	89
4.1.1. Ý tưởng và thiết kế.....	90
4.1.2. Chế tạo module hoàn chỉnh	92
4.1.3. Vận hành module	97
4.2. Vận hành mô hình PMSM ACMC của dSPACE	101
4.2.1. Mô hình bộ điều khiển PMSM trên Matlab/Simulink	101
4.2.2. Xây dựng giao diện điều khiển trên ControlDesk	112
4.2.3. Chạy thực nghiệm trên mô hình và đánh giá kết quả	114
4.2.3.1. Chế độ điều khiển tốc độ	114
4.2.3.2. Chế độ điều khiển dòng	117
4.2.3.3. Chế độ điều khiển vòng lặp mở	118
4.2.3.4. Chế độ thêm tải	119
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	123
5.1. Kết luận	123
5.2. Hướng phát triển trong tương lai	124
TÀI LIỆU THAM KHẢO	125

Danh mục các chữ viết tắt và kí hiệu

PMSM: Permanent Magnet Synchronous Motors

ACMC: AC Motor Control

MBD: Model-based Design

RCP: Rapid Control Prototyping

GPIO: General Purpose Input/Output

GUI: Graphical User Interface

MIL: Model-in-the-Loop Simulation

SIL: Software-in-the-Loop Simulation

PIL: Processor-in-the-Loop Simulation

HIL: Hardware-in-the-Loop Simulation

RTI: Real-time Interface

ADC: Analog Digital Converter

DAC: Digital Analog Converter

PGI: Programmable Generic Interface

FPGA: Field-Programmable Gate Array

PS: Power Stage

SC: Signal Conditioning

UCB: Unit Connection Bus

Danh mục các hình

Hình 2.1. Sự ứng dụng rộng rãi của động cơ điện	4
Hình 2.2. Chuyển động của rotor	5
Hình 2.3. Stator (bên phải) và rotor (bên trái) tương ứng trong động cơ điện	6
Hình 2.4. Phân loại động cơ điện	6
Hình 2.5. Cấu tạo bên trong của động cơ PMSM	7
Hình 2.6. Hai cách bố trí stator và rotor khác nhau trong PMSM	8
Hình 2.7. PMSM nam châm trên bề mặt	8
Hình 2.8. PMSM nam châm bên trong	8
Hình 2.9. Stator cuộn dây phân bố	9
Hình 2.10. Stator cuộn dây tập trung	9
Hình 2.11. Từ trường quay tạo ra trong stator	9
Hình 2.12. Ứng dụng rộng rãi của động cơ PMSM	11
Hình 2.13. Sự khác biệt giữa phương pháp truyền thống và MBD	12
Hình 2.14. Mô hình DC-DC Converter trong Simulink	13
Hình 2.15. MBD theo tiêu chuẩn V-model	15
Hình 2.16. Sơ đồ MIL	18
Hình 2.17. Sơ đồ SIL	19
Hình 2.18. Sơ đồ PIL	20
Hình 2.19. Sơ đồ RCP	21
Hình 2.20. Sơ đồ HIL	22
Hình 3.1. Giao diện chính của Matlab	23
Hình 3.2. Giao diện chính của Simulink	24
Hình 3.3. RTI sử dụng cho MicroLabBox	26
Hình 3.4. Ứng dụng của ControlDesk	27
Hình 3.5. ControlDesk hỗ trợ truy cập một số phần cứng dSPACE	28
Hình 3.6. Giao diện GUI trên ControlDesk	29
Hình 3.7. Thu thập dữ liệu bằng độ thị trong ControlDesk	30
Hình 3.8. Tổng quan giao diện ControlDesk	31
Hình 3.9. Giao diện chính của Configuration for RapidPro	33
Hình 3.10. MicroLabBox của dSPACE	34

Hình 3.11. Sơ đồ 2 board DS1202 và DS1302	36
Hình 3.12. Các cổng kết nối ở mặt trước của MicroLabBox.....	37
Hình 3.13. Các pin analog I/O cổng kết nối A	38
Hình 3.14. Mạch đầu vào đơn giản analog input Class 1	39
Hình 3.15. Các pin analog I/O cổng kết nối B.....	39
Hình 3.16. Mạch đầu vào đơn giản analog input Class 2	40
Hình 3.17. Sơ đồ đơn giản analog output Class 1.....	40
Hình 3.18. Các pin digital I/O kênh A	41
Hình 3.19. Sơ đồ đơn giản kênh 2 chiều Digital Class 1	42
Hình 3.20. Các pin digital I/O kênh B	42
Hình 3.21. Sơ đồ đơn giản kênh 2 chiều Digital Class 2	43
Hình 3.22. Các pin dành cho tiêu chuẩn kết nối RS232	43
Hình 3.23. Các pin dành cho tiêu chuẩn kết nối RS422	44
Hình 3.24. Các pin dành cho tiêu chuẩn kết nối RS485	44
Hình 3.25. Các pin dành cho Resolver	45
Hình 3.26. Các pin dành cho kết nối giao tiếp CAN	45
Hình 3.27. Các cổng kết nối ở mặt trước của MicroLabBox.....	46
Hình 3.28. Cổng kết nối Ethernet trên MictoLabBox	47
Hình 3.29. Giao diện web của MicroLabBox	48
Hình 3.30. Ứng dụng MicroLabBox trong lĩnh vực y tế	48
Hình 3.31. RapidPro System.....	49
Hình 3.32. Cấu tạo bên ngoài của RapidPro SC Unit.....	50
Hình 3.33. Cấu tạo bên trong của RapidPro SC Unit	51
Hình 3.34. Cấu tạo bên ngoài của Power Unit.....	52
Hình 3.35. Cấu tạo bên trong của RapidPro Power Unit	52
Hình 3.36. Cấu tạo bên ngoài của RapidPro Control Unit	53
Hình 3.37. Cấu tạo bên trong của RapidPro Control Unit.....	54
Hình 3.38. Sơ đồ khối của RapidPro Power Unit	55
Hình 3.39. RountingID 44205	56
Hình 3.40. RountingID 60766	56
Hình 3.41. Kết nối Sub-D của F1 Connector.....	57

Hình 3.42. Kết nối Sub-D của F2 Connector.....	59
Hình 3.43. Kết nối USB đến máy tính.....	60
Hình 3.44. Kết nối Sub-D của R1 Connector	61
Hình 3.45. Kết nối Sub-D của R2 Connector	62
Hình 3.46. Vị trí LED trên RapidPro Power Unit	63
Hình 3.47. Sơ đồ mạch chế độ General purpose, single channel và double channel	67
Hình 3.48. Sơ đồ mạch chế độ eletric drive, two half-bridges	68
Hình 3.49. Sơ đồ mạch kênh 2 của chế độ eletric drive, half-bridge and brake chopper..	69
Hình 3.50. Điều khiển high-side	72
Hình 3.51. Mạch điều khiển low-side	73
Hình 3.52. Điều khiển động cơ BLDC/BLAC	74
Hình 3.53. MicroLabBox trong việc điều khiển động cơ	75
Hình 3.54. Tổng quan mô hình PMSM ACMC.....	76
Hình 3.55. Thông số hoạt động của PMSM	76
Hình 3.56. Tổng quan kết nối giữa RapidPro và PMSM.....	78
Hình 3.57. Cỗng Rig	78
Hình 3.58. Sơ đồ khối mô hình động cơ PMSM ACMC	79
Hình 3.59. Sơ đồ nối dây hệ thống phanh của PMSM	80
Hình 3.60. Sơ đồ nối dây Encoder và Hall	81
Hình 3.61. Sơ đồ nối dây giữa RapidPro và PMSM.....	82
Hình 3.62. Sơ đồ nối dây giữa Rig và L1_R2	83
Hình 3.63. Sơ đồ nối dây giữa Rig và L1_R1	84
Hình 3.64. Dây kết nối	85
Hình 3.65. Các cổng giao tiếp Digital/Analog.....	85
Hình 3.66. Dây nguồn	86
Hình 3.67. Sơ đồ khối kết nối cổng Rig tới MicroLabBox	86
Hình 3.68. Sơ đồ nguồn đi vào cổng Rig.....	87
Hình 3.69. Sơ đồ kết nối cổng Rig đến cổng kết nối Digital kênh A	88
Hình 3.70. Sơ đồ kết nối cổng Rig đến cổng kết nối Analog kênh A	89
Hình 4.1. Module thiết kế bằng Solidworks (2D).....	91
Hình 4.2. Module thiết kế bằng Solidworks (3D).....	92

Hình 4.3. Kích thước của module	93
Hình 4.4. Bản vẽ in MICA.....	94
Hình 4.5. Sơ đồ đi dây cho các chân digital và LED.....	94
Hình 4.6. Đi dây cho các chân analog output	95
Hình 4.7. Sơ đồ đi dây cho analog input.....	95
Hình 4.8. Mô hình hoàn thiện	96
Hình 4.9. Dây cáp kết nối	96
Hình 4.10. Sơ đồ kết nối	97
Hình 4.11. Sơ đồ đấu dây cho module (thực tế)	97
Hình 4.12. Thuật toán trên Matlab.....	98
Hình 4.13. Điều khiển tín hiệu digital output	98
Hình 4.14. Tín hiệu Analog Output	99
Hình 4.15. Đọc tín hiệu analog input.....	100
Hình 4.16. Bộ điều khiển module trên ControlDesk	100
Hình 4.17. Đồ thị điện áp của biến trở.....	101
Hình 4.18. Mô hình PMSM Controller trên Matlab/Simulink	102
Hình 4.19. Các khối xử lí bên trong Control	103
Hình 4.20. Khối xử lí tín hiệu đầu vào Input.....	103
Hình 4.21. Khối xử lí tín hiệu Controller	104
Hình 4.22. File init đuôi .m chứa các thông số đầu vào ban đầu.....	105
Hình 4.23. PMSM Control.....	106
Hình 4.24. Khối Speed_Controller	106
Hình 4.25. PI Control của điều khiển tốc độ	107
Hình 4.26. Khối Current_controller.....	107
Hình 4.27. Điều khiển i_d bằng PI Control	108
Hình 4.28. Điều khiển i_q bằng PI Control	108
Hình 4.29. Thuật toán điều chỉnh I_set_source_SW	109
Hình 4.30. Thuật toán điều chỉnh V_set_source_SW.....	109
Hình 4.31. Thuật toán điều chỉnh tải	110
Hình 4.32. Khối torque_2_duty_cycle.....	110
Hình 4.33. Bên trong khối voltage_2_duty_cycle	111

Hình 4.34. Thuật toán tính toán vị trí	111
Hình 4.35. Khối tín hiệu đầu ra PMSM Output.....	112
Hình 4.36. Giao diện điều khiển chung	113
Hình 4.37. Đồ thị điều khiển tốc độ 300 rpm	115
Hình 4.38. Giảm tốc độ động cơ từ 300rpm đến -700 rpm	116
Hình 4.39. Chế độ điều khiển bằng dòng 0.95A.....	117
Hình 4.40. Chế độ điều khiển dòng 1.2A	117
Hình 4.41. Chế độ Open Loop hoạt động ổn định 500rpm	118
Hình 4.42. Chế độ Open Loop hoạt động ổn định 1000rpm	119
Hình 4.43. Chế độ thêm tải Manual Load.....	120
Hình 4.44. Điều khiển tải với 0.4 Nm và 0.2 Hz	121
Hình 4.45. Điều khiển tải với 1 Nm và 0.2 Hz	121
Hình 4.46. Điều khiển tải với 1 Nm và 0.4 Hz	122

Danh mục các bảng

Bảng 3.1. Thông số ADC Class 1 và ADC Class 2	37
Bảng 3.2. Đặc điểm DIO Class 1 và DIO Class 2	40
Bảng 3.3. Thông số các loại nguồn cảm biến từ MicroLabBox	46
Bảng 3.4. Bảng tín hiệu của F1 Connector	58
Bảng 3.5. Bảng tín hiệu của F2 Connector	59
Bảng 3.6. Bảng tín hiệu của R1 Conneter	61
Bảng 3.7. Bảng tín hiệu của R2 Connector.....	62
Bảng 3.8. Module có sẵn trên RapidPro Power Unit.....	64
Bảng 3.9. Các chế độ điều khiển đầu ra.....	66
Bảng 3.10. Thông số kỹ thuật module [1]	69
Bảng 3.11. Thông số kỹ thuật module [2]	71
Bảng 3.12. Thông số kỹ thuật chi tiết động cơ APM-SB03ADK-9	77

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Phương pháp Model-Based Design (MBD) là một xu hướng hiện đại trong lĩnh vực kỹ thuật điều khiển và thiết kế hệ thống. Một trong những lợi ích lớn nhất của phương pháp MBD là tăng tính hiệu quả và độ chính xác trong quá trình phát triển các hệ thống phức tạp. Bằng cách sử dụng mô hình hóa, các kỹ sư có thể mô phỏng và kiểm tra các kịch bản hoạt động khác nhau trước khi sản xuất thực tế. Điều này giúp giảm thiểu sai sót và lỗi thiết kế, đồng thời tăng cường tính chính xác của sản phẩm cuối cùng. Việc này không chỉ cải thiện chất lượng sản phẩm mà còn giảm đáng kể chi phí và thời gian phát triển. Những ưu điểm này đặc biệt quan trọng trong bối cảnh cạnh tranh khốc liệt và yêu cầu cao về chất lượng và độ tin cậy của các hệ thống công nghiệp ngày nay.

MicroLabBox, một thiết bị phần cứng tiên tiến, là một công cụ mạnh mẽ trong việc hỗ trợ MBD. Với khả năng thực hiện các tác vụ phức tạp với độ chính xác cao, MicroLabBox là một lựa chọn lý tưởng cho các ứng dụng điều khiển và thiết kế hệ thống. Thiết bị này hỗ trợ nhiều giao thức kết nối và chuẩn giao tiếp khác nhau, giúp dễ dàng tích hợp vào các hệ thống hiện có cũng như mở rộng thêm các tính năng mới. Hơn nữa, MicroLabBox cho phép hiện thực hóa các mô hình MBD trực tiếp trên phần cứng, điều này có nghĩa là các mô phỏng và kiểm tra có thể được thực hiện trong môi trường thực tế, giúp tăng cường độ tin cậy và tính thực tiễn của các mô hình thiết kế. Sự kết hợp giữa MBD và MicroLabBox không chỉ tạo ra các hệ thống điều khiển hiệu quả mà còn đảm bảo rằng các giải pháp phát triển luôn được kiểm chứng và tối ưu hóa ngay từ giai đoạn đầu.

Trong đó, Permanent Magnet Synchoronous Motor (PMSM) là loại động cơ điện xoay chiều có nhiều ưu điểm vượt trội so với các loại động cơ khác như hiệu suất cao, mật độ moment cao, tiết kiệm năng lượng, độ ồn thấp với độ tin cậy cao. Việc sử dụng các thiết bị phần cứng và phần mềm dSPACE cụ thể là MicroLabBox và RapidPro trong việc nghiên cứu PMSM sẽ giúp cho việc vận hành, lập trình điều khiển trở nên hiệu quả hơn. Qua đó có thể mô phỏng và đánh giá hiệu suất PMSM trong các điều kiện hoạt động khác nhau trước khi chế tạo và thử nghiệm thực tế.

1.2. Mục đích của đề tài

Hiểu rõ về cấu trúc, các thông số kỹ thuật, vận hành, ưu điểm, nhược điểm và xu hướng phát triển của động cơ PMSM.

Có kiến thức về phương pháp Model-Based Design (MBD) trong việc xây dựng và phát triển hệ thống nhúng.

Biết cách sử dụng các phần cứng dSPACE: RapidPro system, MicroLabBox.

Sử dụng thuật toán có sẵn của mô hình động cơ PMSM ACMC để vận hành mô hình và đánh giá hiệu suất.

Thiết kế, chế tạo module mở rộng GPIO của MicroLabBox.

Sử dụng ControlDesk để thiết kế giao diện điều khiển (GUI) cho mô hình động cơ PMSM ACMC.

Luyện tập kỹ năng sử dụng phần mềm Matlab/Simulink và dSPACE.

1.3. Phương pháp nghiên cứu

Nghiên cứu tài liệu của dSPACE để tìm hiểu về nguyên lý hoạt động, cách vận hành của phần cứng và cách sử dụng các chức năng của phần mềm.

Nghiên cứu tài liệu liên quan để hiểu rõ nguyên lý hoạt động và ứng dụng của mô hình động cơ PMSM ACMC của dSPACE.

Nghiên cứu, ứng dụng một phần nhỏ của phương pháp thiết kế dựa vào mô hình (Model-Based Design), Rapid Control Prototyping(RCP) cho mô hình điều khiển động cơ.

1.4. Đối tượng nghiên cứu

Mô hình động cơ PMSM ACMC của dSPACE và MicroLabBox.

1.4.1. Phạm vi nghiên cứu

Nghiên cứu và phân tích cơ sở lý thuyết động cơ PMSM và tính ứng dụng trong các lĩnh vực ô tô, công nghiệp, ...

Nghiên cứu một số phần cứng : MicroLabBox, RapidPro Power Unit và phần mềm: ControlDesk, ConfigurationDesk for RapidPro của dSPACE.

Nghiên cứu thuật toán điều khiển động cơ PMSM có sẵn bằng các phần mềm của dSPACE và Matlab/Simulink.

Vận hành mô hình động cơ PMSM ACMC để đánh giá hiệu suất.

Đề xuất các hướng phát triển để cải tiến và tối ưu hóa mô hình động cơ PMSM phù hợp với ứng dụng thực tế.

1.4.2. Bộ cục đồ án

Chương 1: Tổng quan đề tài

Chương 2: Cơ sở lý thuyết

Chương 3: Mô hình PMSM ACMC của dSPACE

Chương 4: Vận hành mô hình PMSM ACMC của dSPACE

Chương 5: Kết luận và hướng phát triển

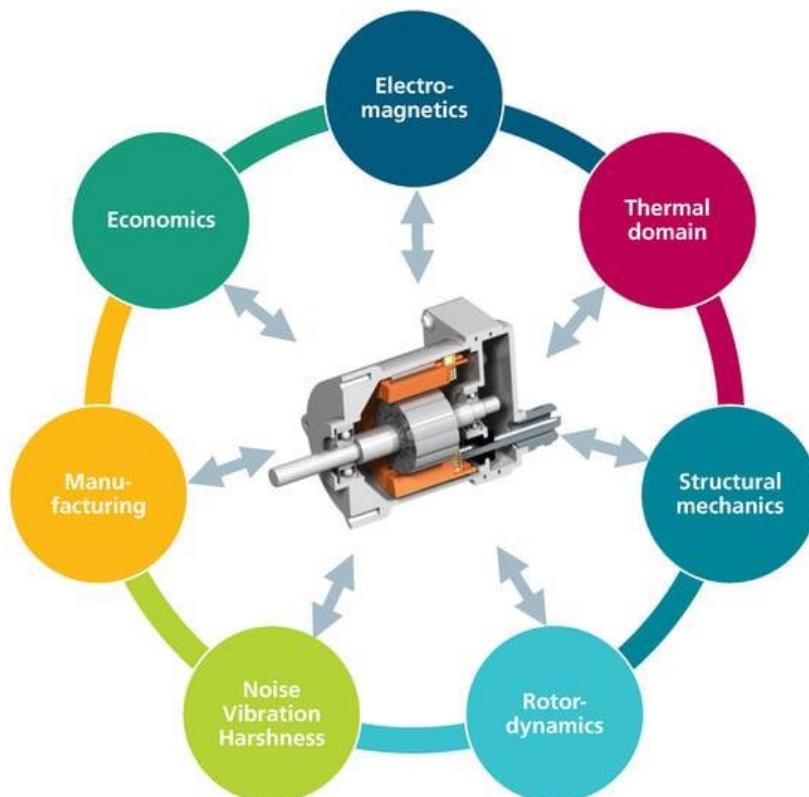
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về động cơ điện

2.1.1. Giới thiệu động cơ điện

Động cơ điện là một loại máy điện được sử dụng rộng rãi nhất trong rất nhiều các ứng dụng từ dân dụng đến công nghiệp. Động cơ điện có khả năng chuyển đổi năng lượng điện trở thành chuyển động cơ học. Các chuyển động này sẽ được đánh giá bằng các biến công suất cơ học, chẳng hạn như tốc độ, moment xoắn và mô lực. Hầu hết các loại động cơ điện hoạt động được nhờ vào sự tương tác qua lại giữa từ trường của động cơ và dòng điện đi qua các cuộn dây để ra lực ở dưới dạng moment xoắn áp lên trục của động cơ. Về cơ bản, động cơ điện cũng tương tự máy phát điện, nhưng hoạt động ngược lại, nếu như máy phát điện chuyển đổi cơ năng thành điện năng thì động cơ điện chuyển đổi điện năng để biến thành cơ năng.

Các động cơ điện có thể hoạt động được nhờ nguồn cung cấp năng lượng bằng cả dòng điện một chiều (DC) như là pin, bộ chỉnh lưu hoặc bằng nguồn điện xoay chiều (AC) như mạng lưới điện, máy biến tần hoặc máy phát điện.

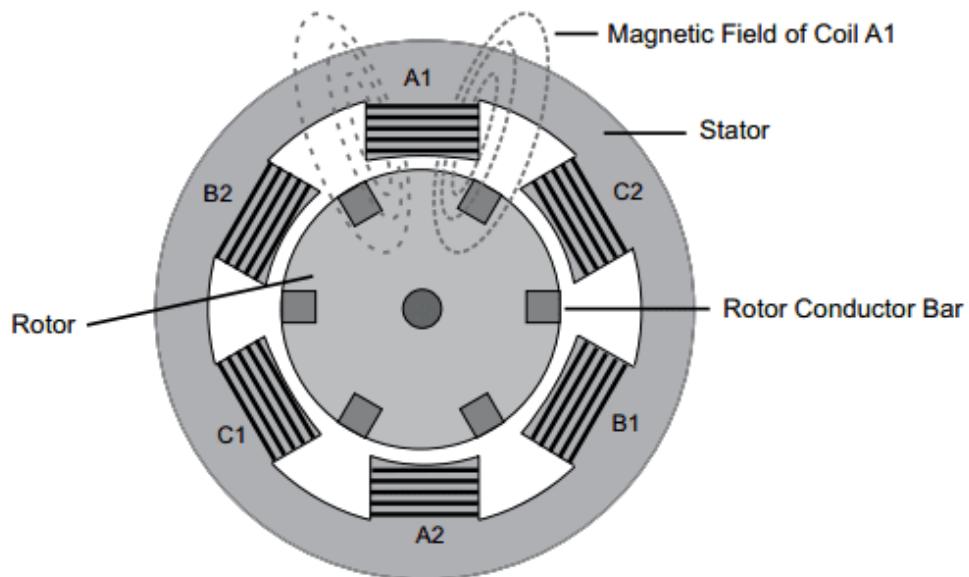


Hình 2.1. Sự ứng dụng rộng rãi của động cơ điện

Động cơ điện được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ các ngành công nghiệp nặng đến những ứng dụng nhỏ trong đời sống hàng ngày. Trong các ngành công nghiệp, chúng là nguồn cung cấp sức mạnh để vận hành máy móc cũng như quy trình sản xuất, có thể kể đến các ứng dụng như quạt công nghiệp, máy bơm, máy công cụ, robot. Trong cuộc sống hàng ngày, động cơ điện là trái tim của các thiết bị gia đình, từ máy giặt, tủ lạnh, quạt đến các máy khoan điện hay máy hút bụi. Động cơ điện đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày cũng như sự phát triển của công nghiệp.

2.1.2. Cấu tạo và nguyên lý hoạt động chung

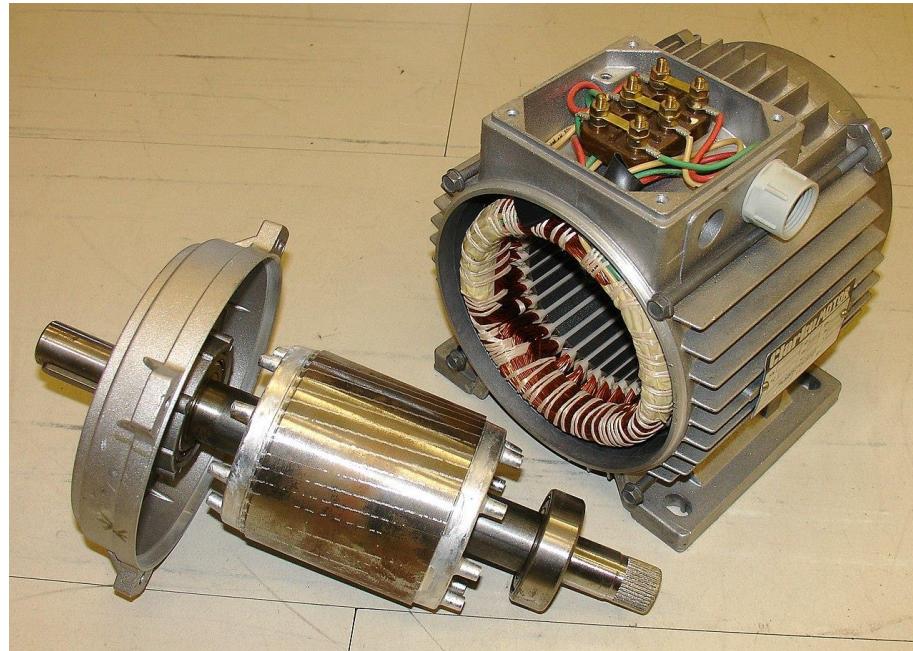
Một động cơ điện thông thường có cấu tạo cơ khí gồm hai bộ phận: đó là *rotor*, bộ phận chuyển động và *stator*, bộ phận đứng yên. Xét về mặt điện tử, động cơ điện sẽ bao gồm hai phần: phần nam châm từ trường và phần ứng, một phần sẽ được gắn vào rotor và phần còn lại gắn vào stator. Kết hợp với nhau chúng sẽ tạo thành một mạch từ, các nam châm sẽ tạo ra từ trường đi qua phần ứng. Nam châm đó có thể là nam châm điện hoặc nam châm vĩnh cửu. Phần nam châm sẽ được nằm trên stator, còn phần ứng sẽ ở rotor, nhưng ở một số trường hợp có thể đảo ngược tùy vào ứng dụng.



Hình 2.2. Chuyển động của rotor

- Rotor: là bộ phận chuyển động để cung cấp năng lượng cơ học cho động cơ điện. Nó chuyển động nhờ vào sự tương tác giữa cuộn dây và từ trường từ đó tạo ra moment xoắn quanh trục.

- Stator: là bộ phận đứng im trong động cơ điện. Stator bao quanh rotor, thường chứa các nam châm từ trường, đó có thể là nam châm điện (vòng dây quấn quanh lõi sắt) hoặc là nam châm vĩnh cửu. Nó tạo ra từ trường đi qua phần ứng của rotor, làm quay phần rotor.

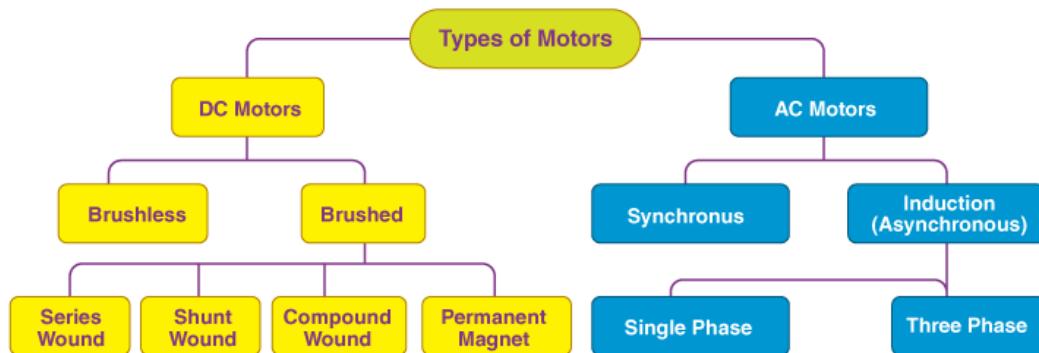


Hình 2.3. Stator (bên phải) và rotor (bên trái) tương ứng trong động cơ điện

2.1.3. Phân loại động cơ điện

Có nhiều cách phân loại động cơ điện khác nhau nhưng thường thấy nhất là phân loại theo nguồn cung cấp là nguồn một chiều (DC) hoặc nguồn xoay chiều (AC).

- DC Motor: gồm các loại động cơ có chổi than và động cơ không chổi than.
- AC Motor: gồm các loại động cơ đồng bộ và động cơ không đồng bộ.



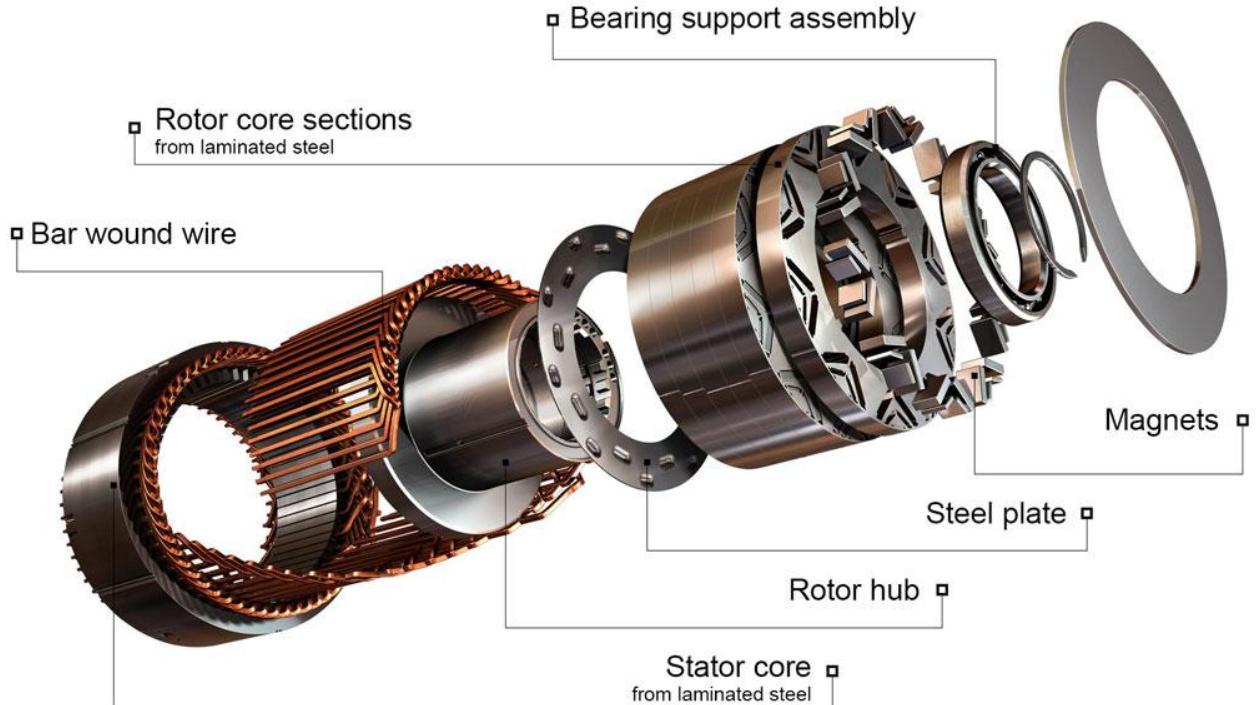
Hình 2.4. Phân loại động cơ điện

2.2. Động cơ điện PMSM

Permanent Magnet Synchronous Motor (PMSM) là một loại động cơ đồng bộ sử dụng nam châm vĩnh cửu cố định, thường được sử dụng cho những mục đích điều khiển cần hiệu suất cũng như sự chính xác ở mức cao. Khác với các loại động cơ điện khác sử dụng những phương pháp điều khiển đơn giản, động cơ PMSM sử dụng các thuật toán điều khiển phức tạp và chuyên sâu cùng với đó là mức chi phí và hiệu suất cao. Nhờ vào tính linh hoạt và hiệu suất cao, động cơ PMSM có mặt trong hầu hết các ứng dụng cần độ chính xác và độ tin cậy cao.

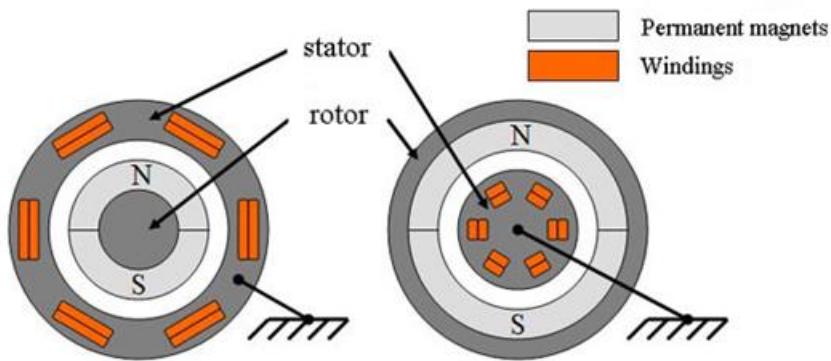
2.2.1. Cấu tạo

Nhu bát kì các loại động cơ điện, động cơ PMSM cũng có hai bộ phận chính: bộ phận đứng yên stator và bộ phận chuyển động rotor.



Hình 2.5. Cấu tạo bên trong của động cơ PMSM

Thông thường, rotor sẽ được bố trí ở bên trong stator, tuy nhiên cũng có vài cấu trúc động cơ ngược lại với stator bên trong và rotor ở bên ngoài.



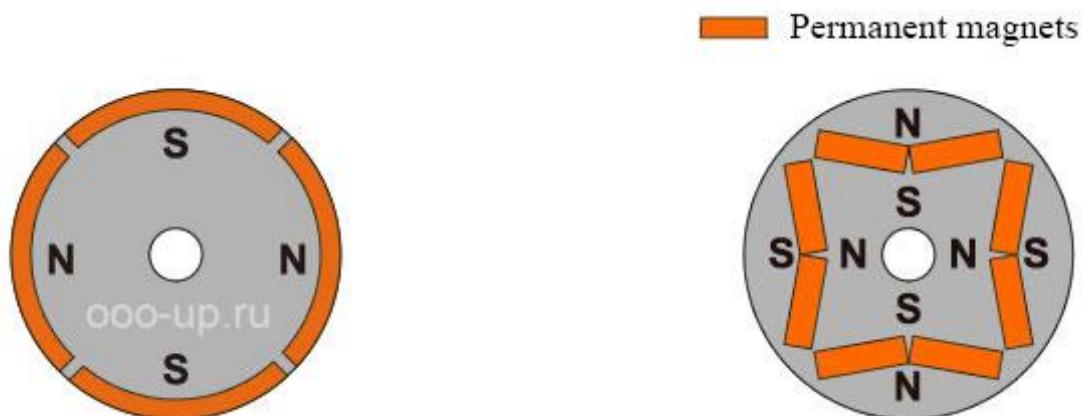
Hình 2.6. Hai cách bố trí stator và rotor khác nhau trong PMSM

2.2.1.1. Rotor

Phần rotor của động cơ PMSM sẽ bao gồm các nam châm vĩnh cửu. Vật liệu thường dùng làm nam châm là những vật liệu có từ tính và độ cứng bức cao.

Dựa vào những cách thiết kế khác nhau của rotor, sẽ có 2 loại động cơ PMSM:

- Động cơ PMSM có nam châm trên bề mặt
- Động cơ PMSM có nam châm bên trong



Hình 2.7. PMSM nam châm trên bề mặt

Hình 2.8. PMSM nam châm bên trong

2.2.1.2. Stator

Phần stator trong động cơ PMSM sẽ bao gồm phần khung bên ngoài và cuộn dây quấn. Thiết kế phổ biến nhất được sử dụng là cuộn dây hai pha và ba pha.

Sẽ có hai cách thiết kế stator khác nhau:

- Stator với cuộn dây phân bố
- Stator với cuộn dây tập trung



Hình 2.9. Stator cuộn dây phân bố

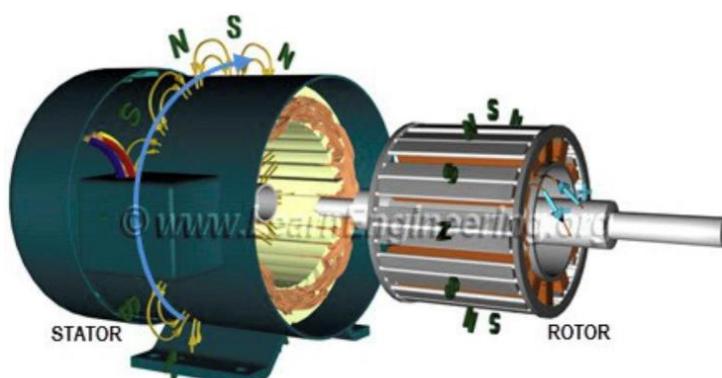


Hình 2.10. Stator cuộn dây tập trung

2.2.2. Nguyên lí hoạt động

Nguyên lí hoạt động của động cơ PMSM rất đơn giản, nhanh và hiệu quả khi so sánh với các động cơ điện thông thường khác. Động cơ PMSM hoạt động dựa vào từ trường quay của stator và từ trường cố định của rotor.

Dòng điện ba pha được đưa vào cuộn stator của động cơ. Sau khi dòng điện được đưa vào, một từ trường quay sẽ được tạo ra trong cuộn stator của động cơ. Vì nam châm cố định được lắp trên rotor nên cực từ của nam châm sẽ được cố định. Theo nguyên lí cực từ cùng loại sẽ đẩy, khác loại sẽ hút nhau, từ trường quay được tạo ra trong stator sẽ đẩy cho rotor quay. Và cuối cùng tốc độ quay của rotor bằng tốc độ quay của từ trường quay tạo ra trong stator. Vì vậy, quá trình khởi động của một động cơ PMSM được coi là bao gồm một giai đoạn khởi động không đồng bộ và sau đó là giai đoạn kéo đồng bộ. Trong giai đoạn khởi động không đồng bộ, tốc độ của động cơ sẽ tăng dần đến khi đạt được tốc độ của từ trường xoay.



Hình 2.11. Từ trường quay tạo ra trong stator

2.2.3. Ưu điểm và nhược điểm

2.2.3.1. Ưu điểm

Động cơ PMSM có nhiều ưu điểm vượt trội khi so sánh với các loại động cơ không đồng bộ truyền thống khác, một vài điểm nổi bật có thể kể đến như:

- Hiệu suất cao: động cơ PMSM sử dụng nam châm vĩnh cửu nên không cần cấp dòng điện vào rotor để tạo từ trường, từ đó loại bỏ hoàn toàn hao tốn trên rotor. Khi so sánh với các loại động cơ điện khác, động cơ PMSM yêu cầu dòng điện thấp hơn trên stator và có hệ số công suất lớn hơn, dẫn đến định mức dòng điện trên bộ điều khiển sẽ nhỏ hơn và làm tăng hiệu suất tổng thể của hệ thống, tiết kiệm chi phí năng lượng.
- Kích thước nhỏ: động cơ PMSM thường có mật độ công suất cao hơn các loại động cơ truyền thống khác, nghĩa là chúng có thể cung cấp một công suất tương đương hoặc lớn hơn với một kích thước và trọng lượng nhỏ hơn.
- Moment khởi động lớn: động cơ PMSM thường tạo ra moment khởi động lớn, khiến chúng trở nên vượt trội trong các ứng dụng yêu cầu khởi động nhanh hoặc phải xử lý các tải lớn.
- Kiểm soát vị trí và tốc độ chính xác: Do từ trường của động cơ PMSM được tạo ra bởi nam châm vĩnh cửu nên việc lấy vị trí của rotor sẽ chính xác hơn, từ đó phù hợp với các ứng dụng yêu cầu điều khiển vị trí và tốc độ, chẳng hạn như xe điện và các loại máy móc công nghiệp.
- Chi phí bảo trì thấp: động cơ PMSM thường có ít các chi tiết nên sẽ giảm thời gian và chi phí bảo trì.

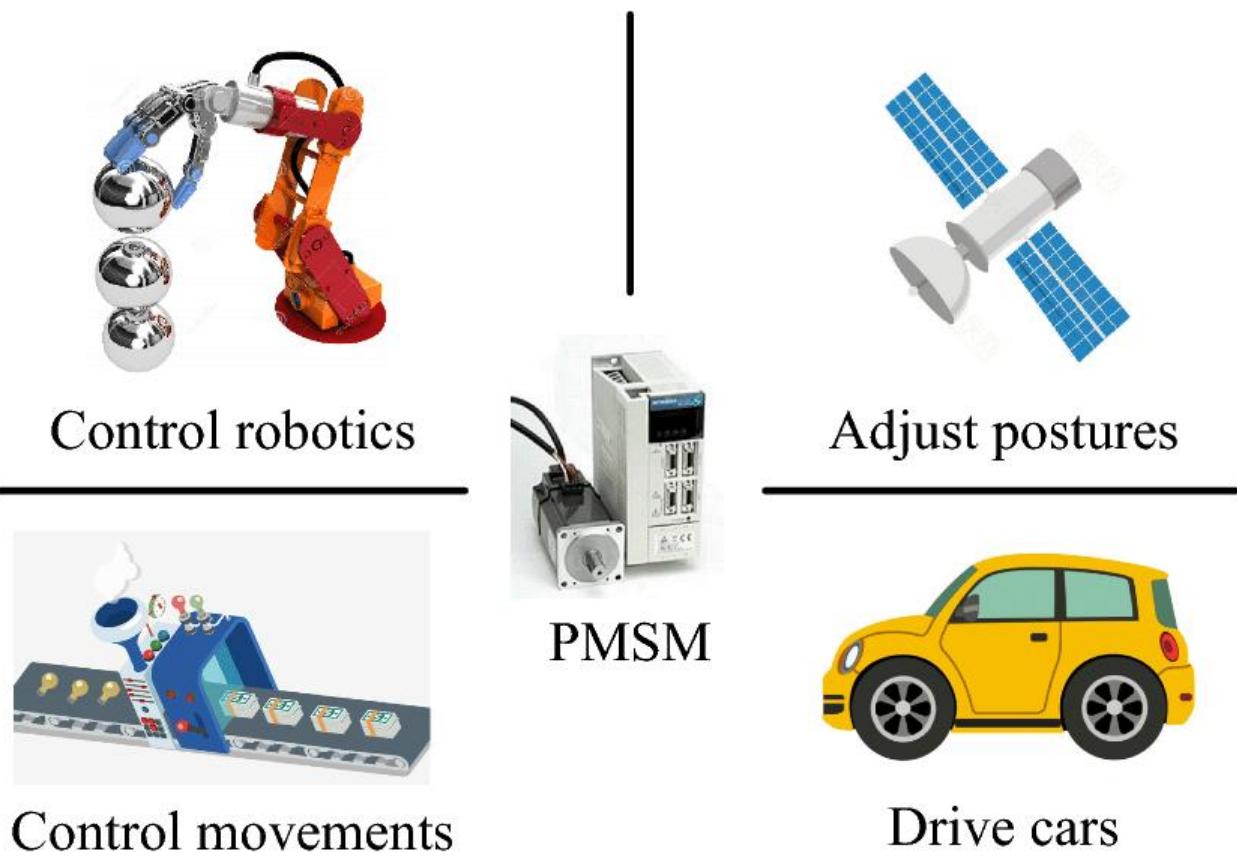
2.2.3.2. Nhược điểm

- Chi phí ban đầu cao, thường cao gấp 3 lần động cơ không đồng bộ thông thường.
- Vật liệu làm nam châm là đất hiếm như neodymium và samarium, đất tiền và khó khai thác
- Yêu cầu cao về mạch điều khiển, việc lắp đặt và vận hành trở nên phức tạp.

2.2.4. Ứng dụng

Động cơ PMSM được ứng dụng trên nhiều lĩnh vực khác nhau, từ gia dụng đến các ngành công nghiệp nặng:

- Gia dụng: PMSM được sử dụng trong các loại quạt, điều hòa, thông gió, các loại máy bơm do hiệu suất năng lượng cao và vận hành ở các tốc độ thay đổi.
- Xe điện và xe điện Hybrid: PMSM thường được sử dụng do có hiệu suất cao, kích thước nhỏ gọn và có khả năng cung cấp moment xoắn cao ở tốc độ thấp.
- Tự động hóa công nghiệp: PMSM được sử dụng rộng rãi trong các ứng dụng công nghiệp cần độ chính xác cao như robot, máy CNC, các máy móc trong quy trình sản xuất.
- Hàng không vũ trụ: PMSM được dùng trong các động cơ điện trên máy bay và vệ tinh, nhờ thiết kế nhẹ, mật độ năng lượng và hiệu suất cao.



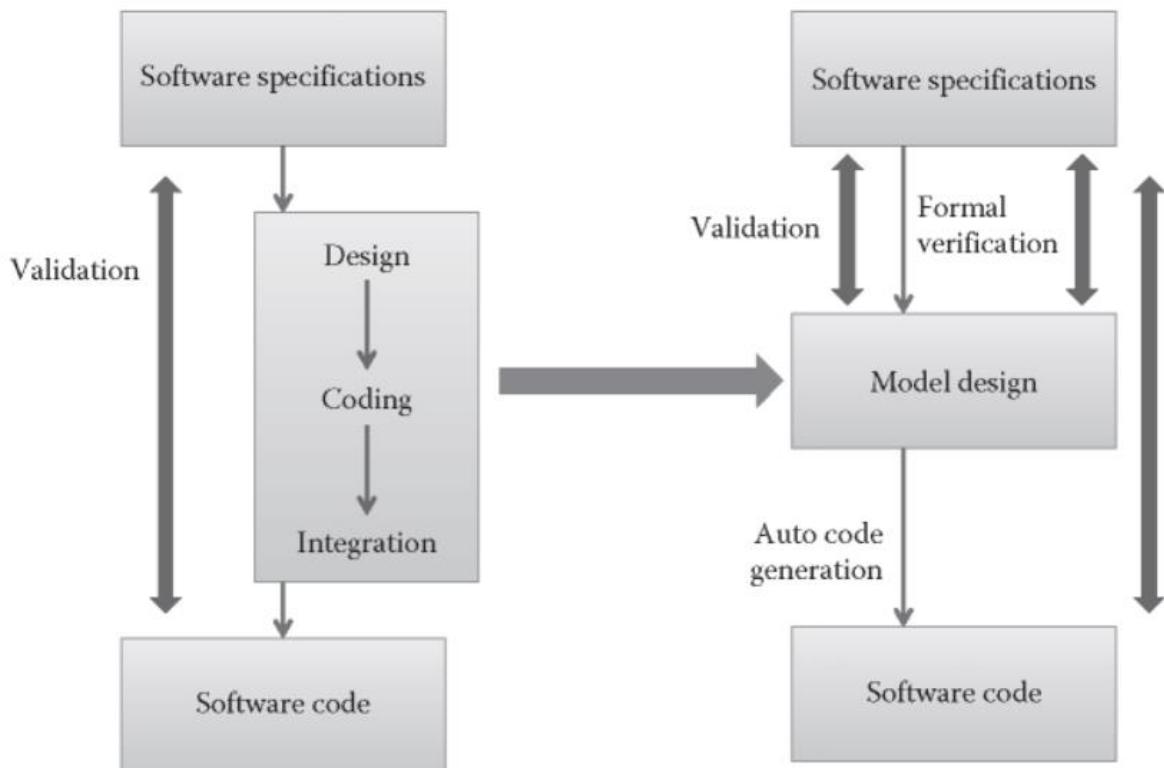
Hình 2.12. Ứng dụng rộng rãi của động cơ PMSM

2.3. Tổng quan về phương pháp Model-Based Design (MBD)

2.3.1. Khái niệm

Model-Based Design, hay còn được gọi là phát triển dựa trên mô hình, là một phương pháp thiết kế hệ thống để chuyển các thông tin của hệ thống nằm trên giấy tờ, tài liệu và tập tin sang các mô hình số hóa được lưu trữ trong môi trường mô phỏng. Mô hình này

chứa toàn bộ thông tin về quá trình thiết kế được tiến hành, bao gồm các yêu cầu, quy định thiết kế, triển khai, xác nhận & xác thực, triển khai. MBD được sử dụng để giải quyết các vấn đề liên quan đến việc thiết kế các hệ thống phức tạp trong các ngành công nghiệp như hàng không vũ trụ và quốc phòng, ô tô, robot học, IoT,... với nhiều lợi ích đáng kể bao gồm: tăng năng suất kỹ sư, giảm chi phí, rút ngắn thời gian phát triển, và nhiều lợi ích khác.

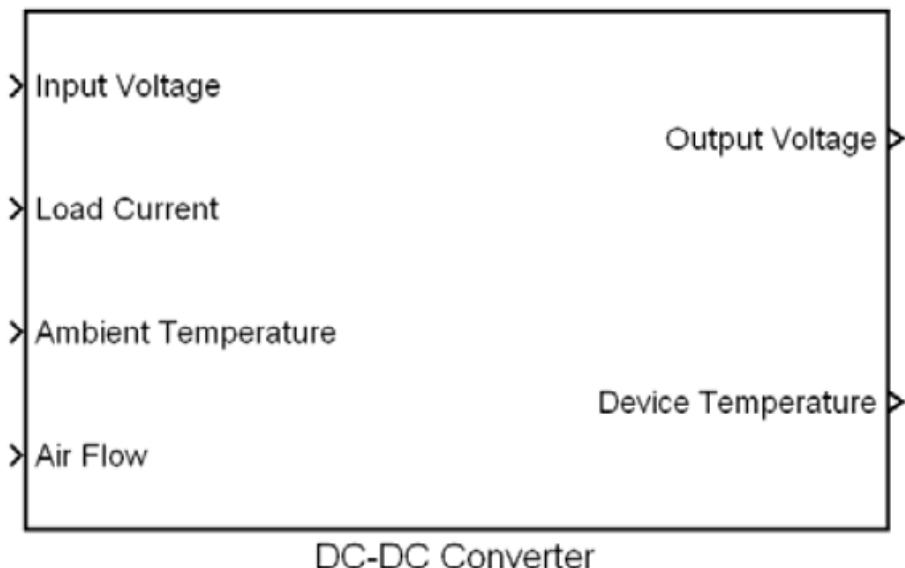


Hình 2.13. Sự khác biệt giữa phương pháp truyền thống và MBD

Từ sơ đồ khái trên, có thể hình dung ra được sự khác biệt giữa 2 phương pháp. Đối với phương pháp truyền thống, từ thông số kĩ thuật và yêu cầu của hệ thống, trải qua các giai đoạn thiết kế, lập trình, tích hợp vào phần cứng, kiểm tra và so sánh, đánh giá với yêu cầu ban đầu, việc này khá là mất thời gian khi thực hiện những hệ thống lớn, có độ phức tạp cao. Mặc khác, với MBD, từ các thông số kĩ thuật và yêu cầu hệ thống, tạo ra một mô hình đại diện cho các yêu cầu trên, xem xét đánh giá mô hình phù hợp với yêu cầu. Tiếp theo, từ mô hình thiết kế, tự động tạo mã C nguồn để tích hợp vào phần cứng, chạy thử nghiệm và lại kiểm tra cầu ban đầu, nếu yêu cầu không thỏa mãn, chỉ cần thực hiện lại từ bước thiết kế mô hình, không cần phải thực hiện lại toàn bộ quá trình, điều này giúp tiết

kiệm rất nhiều thời gian và chi phí. Ngoài ra, MBD còn có nhiều ưu điểm vượt trội so với phương pháp truyền thống như: tính nhất quán, tính đa nhiệm, tính kế thừa,...

Hiểu một cách đơn giản hơn, MBD là phương pháp dựa vào mô hình để mô tả các thông số kỹ thuật, cách vận hành cũng như hiệu suất của một thành phần hoặc toàn bộ hệ thống. Thay vì các thông tin kỹ thuật này được lưu trữ dưới dạng tài liệu giấy, giờ đây nó được lưu trữ trên một mô hình được thiết kế trên phần mềm, với đầy đủ các thông tin cần thiết. Ví dụ cụ thể 12V-5V DC-DC Converter, với phương pháp truyền thống, ta cần có một tài liệu với đầy đủ thông số kỹ thuật về đầu vào, đầu ra của điện áp, dòng điện, các thông số về nhiệt, hiệu suất,... Ngược lại với MBD, ta chỉ cần một mô hình để chứa đầy đủ các thông tin về các thông số kỹ thuật này.



Hình 2.14. Mô hình DC-DC Converter trong Simulink

Với mô hình, ta không những chứa thông tin chi tiết hơn, mà còn có thể kiểm tra trạng thái hoạt động, mối liên hệ giữa các thông số bằng các cổng input/output. Và mô hình này có thể chia sẻ với người khác, hoặc sử dụng ở nhiều hệ thống khác nhau.

2.3.1.1. Ưu điểm của MBD

- Code generation:** MBD cho phép tự động tạo mã nguồn từ mô hình thiết kế, giúp tăng tính nhất quán và giảm thiểu lỗi do sai sót con người trong quá trình viết mã.

- **Giảm thời gian phát triển:** MBD giúp sửa lỗi cũng như điều chỉnh yêu cầu nhanh chóng trong quá trình phát triển.
- **Tiết kiệm chi phí:** MBD giảm chi phí phát triển bằng cách giảm thiểu việc sản xuất các phiên bản vật lý của hệ thống trong quá trình phát triển.
- **Tính linh động:** MBD cho phép dễ dàng điều chỉnh và thay đổi mô hình để phản ánh các yêu cầu và thay đổi trong quá trình phát triển.
- **Hiểu biết về hệ thống:** bằng cách tạo mô hình chi tiết của hệ thống, MBD giúp tăng hiểu biết về cấu trúc và hoạt động của hệ thống, từ đó cải thiện quá trình phát triển và bảo trì sau này.

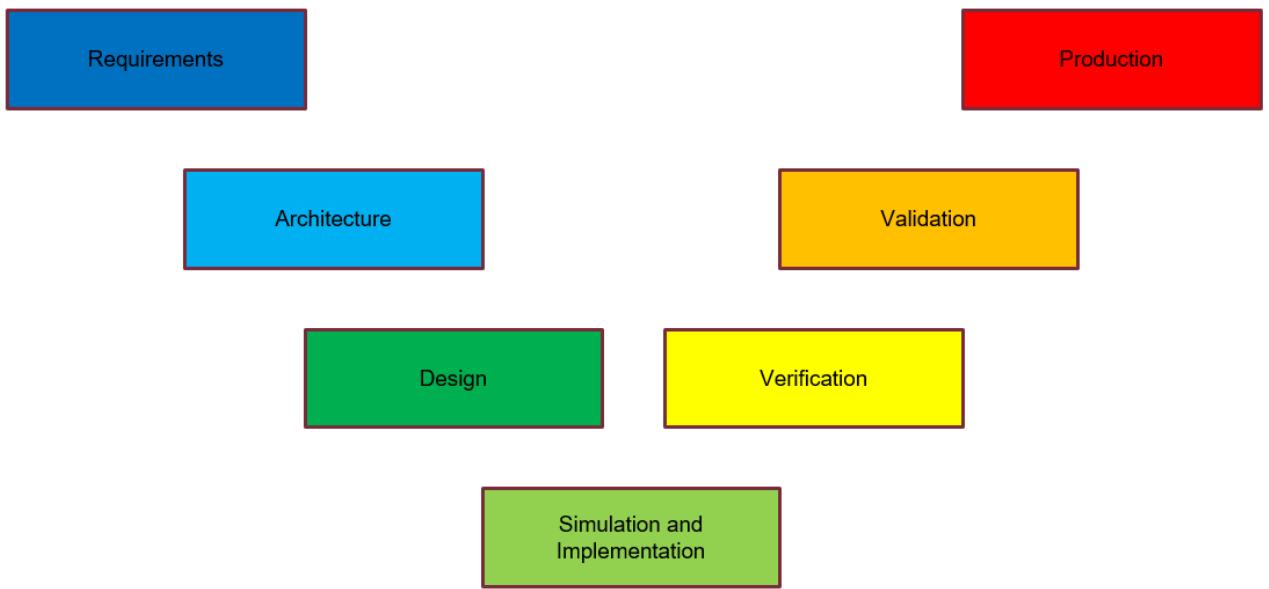
2.3.1.2. Nhược điểm của MBD

- **Đòi hỏi kỹ năng chuyên môn cao:** MBD yêu cầu các kỹ sư có kỹ năng sử dụng các công cụ và ngôn ngữ mô hình hóa phức tạp.
- **Đòi hỏi tài nguyên cao:** MBD cần các thiết bị và phần mềm mạnh mẽ để phục vụ quá trình phát triển và triển khai mô hình.
- **Rủi ro không tương thích:** Nếu mô hình không chính xác hoặc không tương thích với hệ thống thực tế, có thể dẫn đến lỗi hoặc hiệu lầm trong quá trình phát triển.

2.3.1.3. Quy trình làm việc của MBD

MBD thường được thiết kế theo tiêu chuẩn vòng đời phát triển của mô hình chữ V (V-model). V-model được coi là bảng nâng cấp hoặc mở rộng của mô hình thác nước (waterfall), tuy nhiên trong MBD, mỗi bước phát triển thường được liên kết với kiểm thử. Với sự liên kết mạnh mẽ với quá trình kiểm thử lặp đi lặp lại, MBD còn được gọi với một tên khác là mô hình Verification & Validation (V&V-model).

Các bước chính của V-model bao gồm: yêu cầu hệ thống, cấu trúc của hệ thống, thiết kế hệ thống, mô phỏng và thực nghiệm hệ thống, xác nhận và đánh giá, sản xuất.



Hình 2.15. MBD theo tiêu chuẩn V-model

Thông thường, quy trình này sẽ bắt đầu từ nhánh bên trái, đi từ yêu cầu sản phẩm, sau đó tiến hành xây dựng kiến trúc hệ thống, thiết kế hệ thống và kết thúc bằng việc mô phỏng và thực nghiệm. Với mỗi giai đoạn như vậy đều liên kết với nhánh bên phải để xác nhận và đánh giá đúng với yêu cầu sản phẩm, và kết thúc vào việc đưa hệ thống vào sản xuất. Lưu ý, quá trình V&V sẽ lặp đi lặp lại liên tục để đảm bảo sự chính xác ở các giai đoạn ở nhánh bên trái. Tất cả các giai đoạn sẽ được liên kết với nhau tạo thành một hệ thống, có sự trao đổi thông tin qua lại một khi có sự sửa đổi hoặc những chức năng mới xuất hiện.

2.3.2. Yêu cầu hệ thống (System Requirements)

Phân tích yêu cầu hệ thống bao gồm đề cập đầy đủ về các yếu tố cần thiết để triển khai một hệ thống thành công. Giai đoạn này bao gồm phân tích để ra các yêu cầu về thành phần vật lý như bộ phận điện, cơ khí cũng như các yêu cầu về phần mềm như hệ điều hành và bộ xử lý cần thiết để chạy phần mềm nhúng, và một số thành phần khác đối với chức năng dự kiến của hệ thống.

MBD yêu cầu các yêu cầu của hệ thống được đặt trong một cơ sở dữ liệu của phần mềm thiết kế, trong đó mỗi yêu cầu là một đối tượng và liên kết trực tiếp với yêu cầu tổng thể của sản phẩm. Các yêu cầu phải được truyền từ yêu cầu cấp sản phẩm xuống yêu cầu

hệ thống cá nhân, các hệ thống nhỏ hơn và các thành phần. Khi quá trình phát triển sản phẩm tiếp tục các giai đoạn sau, hiểu biết về tổng thể về yêu cầu hệ thống được cải thiện, yêu cầu hệ thống có thể được lặp lại để bổ sung. Tuy nhiên, tất cả các kết nối phải được duy trì và cập nhật.

2.3.3. Cấu trúc hệ thống (System Architecture)

Trong MBD, cấu trúc hệ thống là một bản đồ chi tiết về chức năng của các thành phần phần cứng và phần mềm cũng như sắp xếp theo cấp bậc tự hệ thống cao nhất, các hệ thống phụ và cuối cùng là các đơn vị con.

Kiến trúc hệ thống thường được thực hiện bằng ngôn ngữ mô hình hóa hệ thống, ngôn ngữ này hỗ trợ việc chỉ định, thiết kế, xác nhận và đánh giá hệ thống. Đối với MBD, việc xác định kiến trúc hệ thống tốt sẽ cho phép các kỹ sư kết hợp các hệ thống, hệ thống con và các thành phần một cách chính xác đảm bảo hiệu suất tối ưu và giảm thiểu thời gian quản lý. Ứng dụng sử dụng ngôn ngữ hệ thống có đầy đủ các yếu tố trên như Matlab/Simulink.

2.3.4. Thiết kế hệ thống (System Design)

Thiết kế hệ thống là quá trình hình thành, xác định và mô tả các mô-đun, thành phần và các đơn vị khác nhau của một hệ thống. Quá trình thiết kế đòi hỏi tối ưu về hiệu suất và sẽ đề cập đến mọi thứ về một hệ thống ở một mức độ trừu tượng đủ cao để một kỹ sư vẫn có thể hiểu và đánh giá các yếu tố chính của rủi ro. Trong bước này, kỹ sư sẽ dùng các công cụ mô hình hóa và mô phỏng để tạo ra các phương trình toán học hoặc vật lý của tất cả hệ thống con, bộ phận, thành phần, đường dẫn và thuật toán.

Mô hình sau khi thiết kế sơ bộ, sẽ được dùng để xem xét các yếu tố như thời gian sử dụng, độ tin cậy, chi phí, xác xuất thành công,...Mô hình sẽ được thực hiện lặp đi lặp lại để giải quyết các rủi ro trước khi tiến hành giai đoạn tiếp theo. Điều này rất quan trọng, đảm bao cho mô hình ngày càng tiến gần hơn với vấn đề và yêu cầu của người dùng, nếu không, thực hiện các thay đổi hợp trong khi chi phí thay đổi vẫn còn thấp.

2.3.5. Mô phỏng và triển khai mô hình (Model Simulation and Implementation)

Sau khi hoàn thành quy trình thiết kế, mô hình có độ phức tạp tương đối thấp. Thông qua việc mô phỏng, thử nghiệm các trường hợp kiểm tra khác nhau, kỹ sư sẽ thiết kế lại mô hình có độ phức tạp cao hơn để kết quả mô phỏng sát với thực tế nhất có thể. Các mô

hình phức tạp hơn này hướng dẫn quyết định thông qua quá trình tối ưu hóa thiết kế, xác định các yếu tố nào quan trọng nhất, xem xét biến số và hi sinh một số yếu tố không quan trọng để phù hợp với yêu cầu hệ thống.

- **Auto Code Generation**

Phần mềm nhúng thường được viết bằng C, C++. C là một ngôn ngữ lập trình đa dụng và rất phù hợp cho phần mềm nhúng vì nó độc lập với bộ xử lý, có thể di chuyển, nhanh chóng và cung cấp truy cập trực tiếp vào quản lý bộ nhớ. Mặc dù C không thấp hơn ngôn ngữ máy, nhưng nó thường thân thiện với người dùng hơn so với các ngôn ngữ cấp cao hơn. Do đó, nhiều công ty sẽ chọn sử dụng các công cụ tự động tạo code để tối ưu hóa quy trình chuyển đổi từ mô hình sang mã.

Lợi ích của việc tự động tạo code là sau mỗi lần có điều chỉnh, thay đổi ở mô hình, phần code có thể được cập nhật tự động và triển khai ngay lập tức trên phần cứng. Điều này không phụ thuộc vào độ phức tạp của hệ thống hay thuật toán.

2.3.6. Xác nhận và đánh giá (Verification & Validation)

V&V liên quan đến tất cả quy trình, thử nghiệm thực tế, đánh giá kết quả, ứng dụng trực tiếp và các bản thử nghiệm nhằm đảm bảo rằng hệ thống được thiết kế và phát triển đúng với yêu cầu được đưa ra. Verification là quá trình xác thực mã code của bộ điều khiển hoạt động tốt trong môi trường mô phỏng cũng như áp dụng trên phần cứng thực tế. Validation kiểm tra hiệu suất tổng thể của hệ thống so với các yêu cầu được đặt ra ban đầu.

Verification & Validation bắt đầu từ việc xem xét tất cả các yêu cầu của hệ thống. Mỗi yêu cầu sẽ có một kế hoạch để kiểm tra, đánh giá xem mô hình có thỏa mãn những yêu cầu đó không? Kiểm tra mô hình dựa vào vòng lặp là một phần không thể thiếu trong V&V. Nó bao gồm việc xây dựng các trường hợp thử nghiệm mô phỏng, thực nghiệm, kiểm tra sự thỏa mãn yêu cầu với pass/fail. Quy trình V&V thông thường trải qua các trường hợp kiểm tra như MIL, SIL, PIL, RCP, HIL.

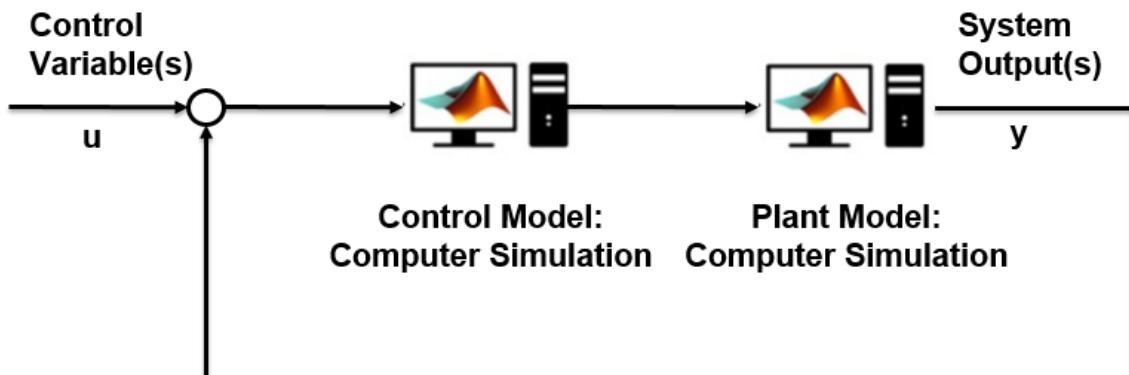
Giả sử một hệ thống đơn giản bao gồm 2 phần chính: bộ điều khiển (controller) và cơ cấu chấp hành (actuator). Hai bộ phận này sẽ được thiết kế thành mô hình, từ đơn giản nhất đến phức tạp nhất, từ trạng thái lý tưởng đi đến hệ thống thực tế, với quy trình kiểm tra được mô tả ở hình sau:

2.3.6.1. Model-in-the-Loop Simulation (MIL) Testing

Dựa vào cơ cấu chấp hành trong thực tế (ví dụ: động cơ,...) mong muốn điều khiển, xây dựng một mô hình cơ cấu chấp hành trong môi trường Simulink (có thể mô phỏng lại các hoạt động của động cơ trong thực tế). Mô hình này có đầy đủ các đặc điểm thiết yếu của phần cứng ngoài thực tế, bao gồm động học, hạn chế, hành vi trong nhiều điều kiện khác nhau.

Sau khi hoàn thành mô hình động cơ trong môi trường Simulink, tiến hành xây dựng mô hình bộ điều khiển để điều khiển động cơ. Mô hình này chứa các logic điều khiển tương tác với mô hình động cơ để đưa ra đầu ra mong muốn (ví dụ: điều khiển tốc độ động cơ).

Model-in-the-Loop Simulation (MIL) Testing là phương pháp chạy mô phỏng biểu diễn sự tương tác bộ điều khiển và động cơ, quan sát kết quả và nhận xét kết quả. Nếu kết quả không được như mong muốn, điều chỉnh lại logic điều khiển, sau đó tiếp tục chạy mô phỏng. Nếu kết quả điều khiển đúng như mong đợi, lưu lại dữ liệu của quá trình kiểm tra để sử dụng ở các bài kiểm tra sau.



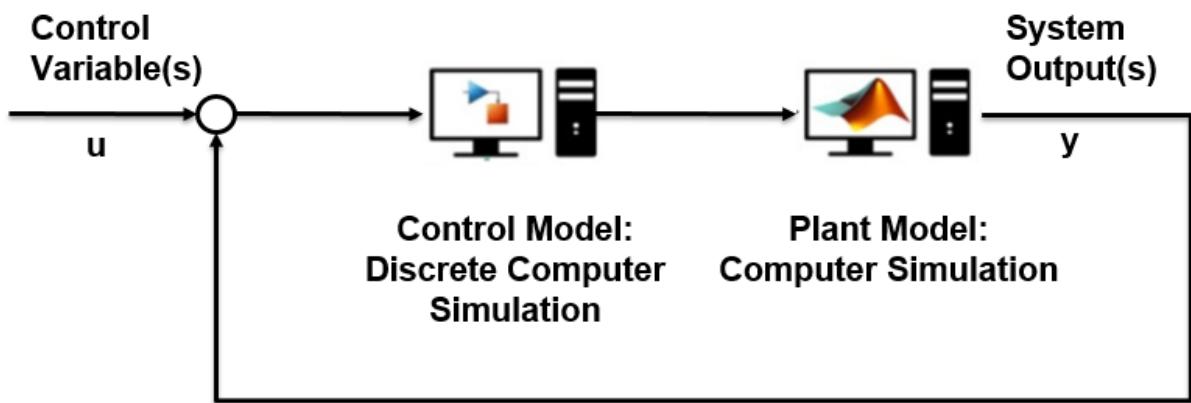
Hình 2.16. Sơ đồ MIL

Đây là sơ đồ của Model-in-the-loop (SIL), có thể thấy cả 2 mô hình động cơ và bộ điều khiển đều ở trong cùng một môi trường mô phỏng, nhằm đánh giá hiệu quả của hệ thống vòng lặp kín.

2.3.6.2. Software-in-the-Loop Simulation (SIL) Testing

Sau khi mô hình đã hoàn thành kiểm tra MIL Testing, bước tiếp theo chính là SIL Testing. Ở giai đoạn này, mô hình động cơ được giữ nguyên so với giai đoạn MIL, bộ điều

khiển sẽ được biến đổi thành mã nguồn (source C code) để tương thích với chương trình nhúng vào vi điều khiển thực tế. Ta thay thế bộ điều khiển ở MIL Testing bằng một bộ điều khiển mới chứa mã C vừa được tạo ở trên, sau đó tiến hành chạy mô phỏng. Quan sát kết quả, so với với kết quả đã lưu ở giai đoạn trước để đánh giá bộ điều khiển về khả năng tạo mã C và có thể được thực hiện trên phần cứng được không? Nếu thu được kết quả tương đồng với giai đoạn trước, thì có thể tiến tới giai đoạn tiếp theo. Ngược lại, nếu có sự khác biệt giữa 2 giai đoạn, quay lại giai đoạn MIL để chỉnh sửa và kiểm tra lại.



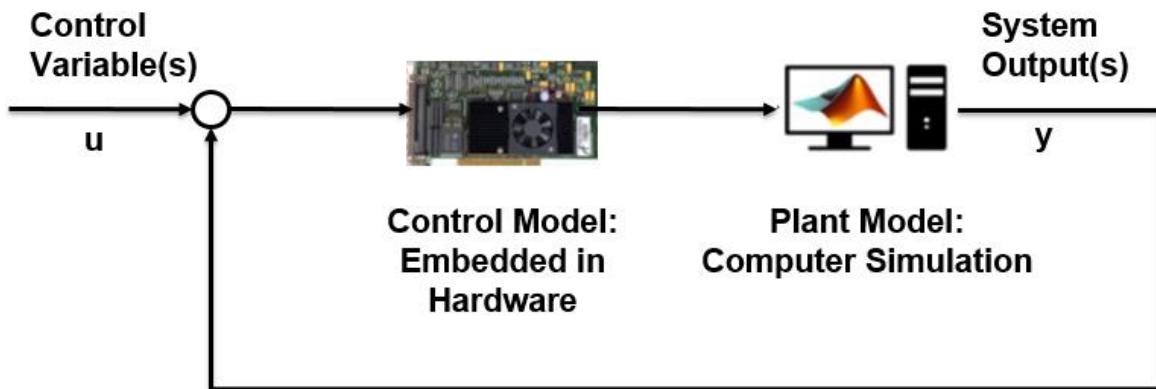
Hình 2.17. Sơ đồ SIL

Sơ đồ của Software-in-the-Loop Simulation (SIL) Testing, bộ điều khiển được tự động tạo mã C bằng phần mềm, vẫn chạy trên cùng môi trường mô phỏng với mô hình động cơ, để đánh giá về khả năng thực hiện bộ điều khiển trên phần cứng.

2.3.6.3. Processor-in-the-Loop Simulation (PIL) Testing

Processor-in-the-Loop (PIL) Testing hiểu đơn giản là kiểm tra đoạn mã C của bộ điều khiển có thực sự tương thích với board điều khiển thực tế không? Ở giai đoạn này, đoạn mã C được tạo tự động từ giai đoạn SIL, sẽ được nạp vào board điều khiển thực tế để chạy trực tiếp trên đó, và được đại diện bởi khối PIL trong môi trường Simulink. Mô hình động cơ vẫn giữ nguyên mô hình mô phỏng như ban đầu. Lưu ý, giai đoạn này giao tiếp giữa mô hình động cơ và PIL vẫn là trên môi trường ảo, không thông qua cổng kết nối I/O của

board điều khiển. Sau khi chạy mô phỏng, tiến hành so sánh kết quả với giai đoạn trước, nếu có sự cố hay lỗi, quay lại giai đoạn MIL và SIL, sửa và tiếp tục lặp lại quy trình.



Hình 2.18. Sơ đồ PIL

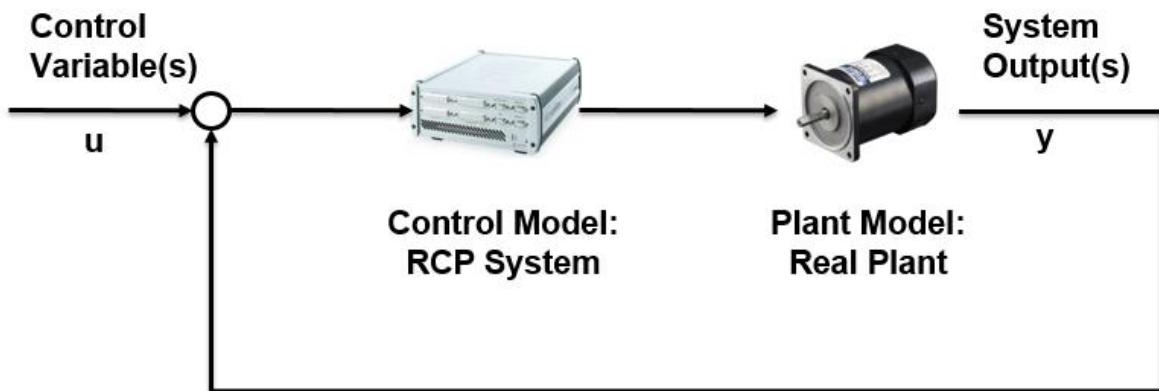
Sơ đồ của Processor-in-the-Loop Simulation (PIL) Testing, mô hình động cơ trong môi trường mô phỏng được điều khiển bởi mã C chạy trên board mạch thực tế (không thông qua I/O connectors).

2.3.6.4. Rapid Control Prototyping (RCP) Testing

Rapid Control Prototyping (RCP) là một quy trình làm việc để kiểm tra hành vi của bộ điều khiển với mục đích nhanh chóng độ thiêt kế, thử nghiệm và tối ưu hóa logic điều khiển. Đây là một bước quan trọng trong vòng đời phát triển sản phẩm nhúng, dùng để xác thực lại mã C điều khiển mà không cần phải nhúng trên board điều khiển thực tế, trong môi trường có sẵn các cảm biến và cơ cấu chấp hành thực tế.

Hệ thống RCP thường sử dụng phần cứng và phần mềm đặt biệt để tạo ra một môi trường thử nghiệm mà thuật toán có thể được thực thi thời gian thực với độ trễ tín hiệu thấp và độ tin cậy cao.

Với quy trình này, mô hình bộ điều khiển được tạo mã C tự động và nạp xuống RCP System có sẵn như: MicroLabBox, MicroAutoBox, DS1104,... Mô hình động cơ được thay thế bằng một động cơ thực tế, với tín hiệu điều khiển được lấy từ RCP System.



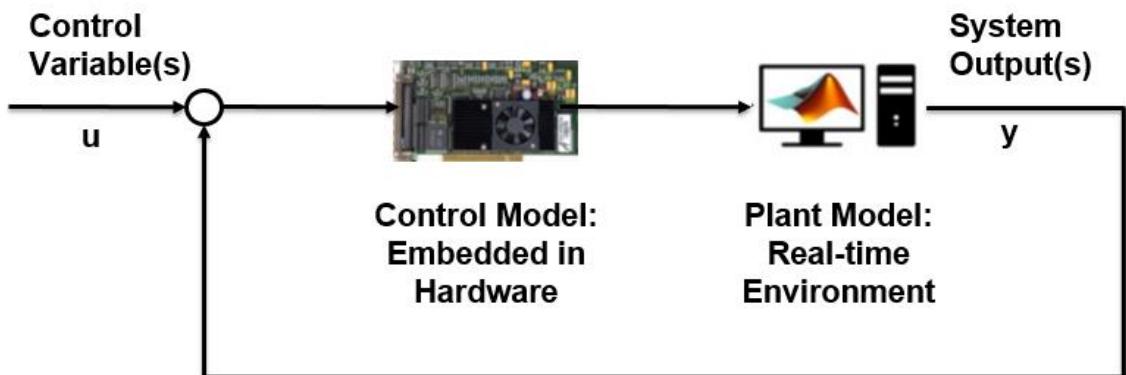
Hình 2.19. Sơ đồ RCP

Sơ đồ của Rapid Control Prototyping (RCP) Testing, sử dụng mô hình động cơ thực tế, được điều khiển bằng hệ thống RCP thông qua các cổng kết nối I/O và các giao tiếp ngoại vi.

2.3.6.5. Hardware-in-the-Loop Simulation (HIL) Testing

Trước khi kết nối board điều khiển vào động cơ thực tế, cần thử nghiệm mô hình động cơ trên một hệ thống thời gian thực. Hệ thống thời gian thực thực hiện các mô phỏng xác định và có kết nối vật lý với board điều khiển thực tế như kết nối I/O, giao tiếp CAN, serial, SPI,... Điều này giúp xác định các vấn đề liên quan đến giao tiếp giữa các cổng với nhau, ví dụ có thể dẫn đến truyền dữ liệu lỗi, độ trễ,... làm cho bộ điều khiển mất ổn định. Những vấn đề này sẽ không được phát hiện trong quá trình thử nghiệm với mô hình động cơ trong môi trường mô phỏng. Nếu có sự khác biệt về kết quả so với các giai đoạn trước đó, cần xem xét và điều chỉnh lại ở giai đoạn trước và lặp lại quy trình.

Giai đoạn HIL thường được sử dụng trong những ứng dụng hệ thống nhúng cần có độ an toàn và chính xác cao, đòi hỏi khắt khe về chất lượng. Một số lĩnh vực cụ thể như ô tô, hàng không, vũ trụ,...



Hình 2.20. Sơ đồ HIL

Sơ đồ của hardware-in-the-Loop (HIL) Testing, mô hình động cơ trong thời gian thực được điều khiển bởi board điều khiển thông qua các cổng kết nối I/O và các giao tiếp ngoại vi.

2.3.7. Sản xuất (Production)

Quá trình sản xuất bao gồm một loại hình kiểm tra có sự tham gia của con người trong môi trường thực tế (ví dụ lái thử chiếc xe). Mục tiêu là đánh giá lại toàn bộ chức năng của hệ thống từ đầu đến cuối với sự hiện diện của con người, sẵn sàng can thiệp nếu có sự cố xảy ra. Trong giai đoạn này, có sự so sánh, đánh giá giữa hiệu suất của hệ thống trong mô hình so với hệ thống thực. Nếu cần cải thiện cho mô hình hệ thống, sự chỉnh sửa sẽ được thực thi trong công cụ xây dựng MBD.

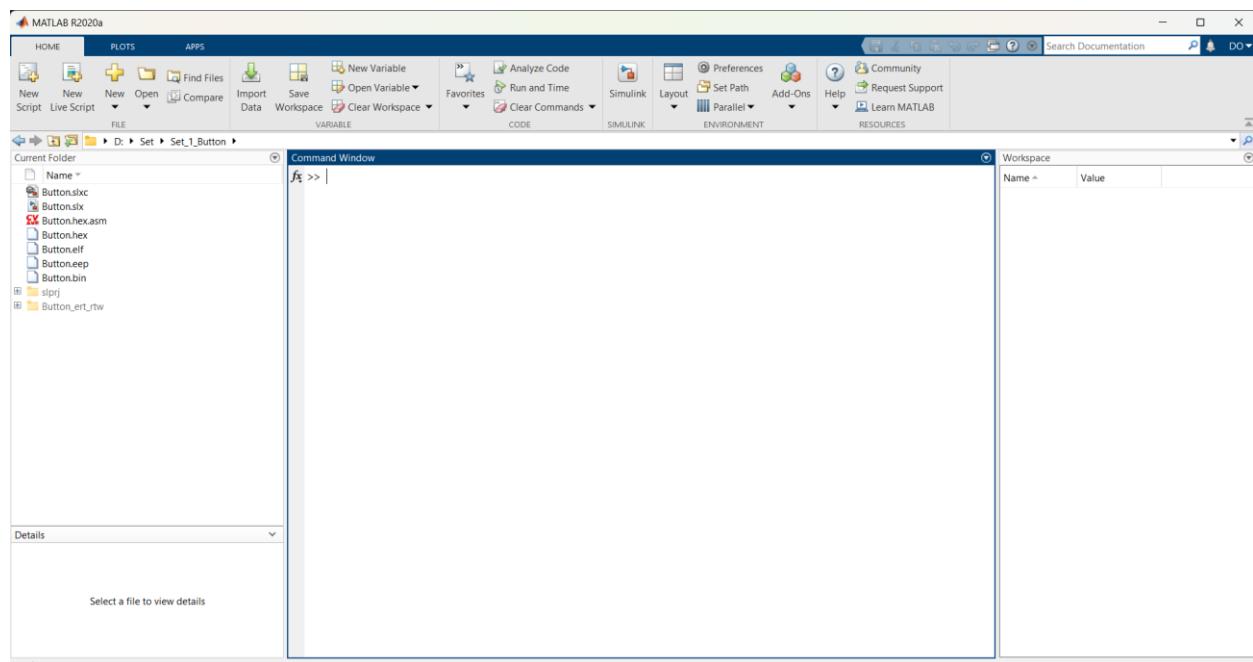
Tương tự, khi cần cập nhật cho bộ điều khiển, mô hình sẽ được làm mới, các mô phỏng được tiến hành bằng cách sử dụng dữ liệu lịch sử, xác minh và xác thực được thực hiện, và các cải tiến được triển khai đến các sản phẩm. Quá trình lặp lại này đảm bảo rằng hệ thống tiến triển và thích nghi với điều kiện thực tế, từ đó tăng cường hiệu suất và độ tin cậy của nó.

CHƯƠNG 3: MÔ HÌNH PMSM ACMC CỦA DSPACE

3.1. Tổng quan về Matlab/Simulink

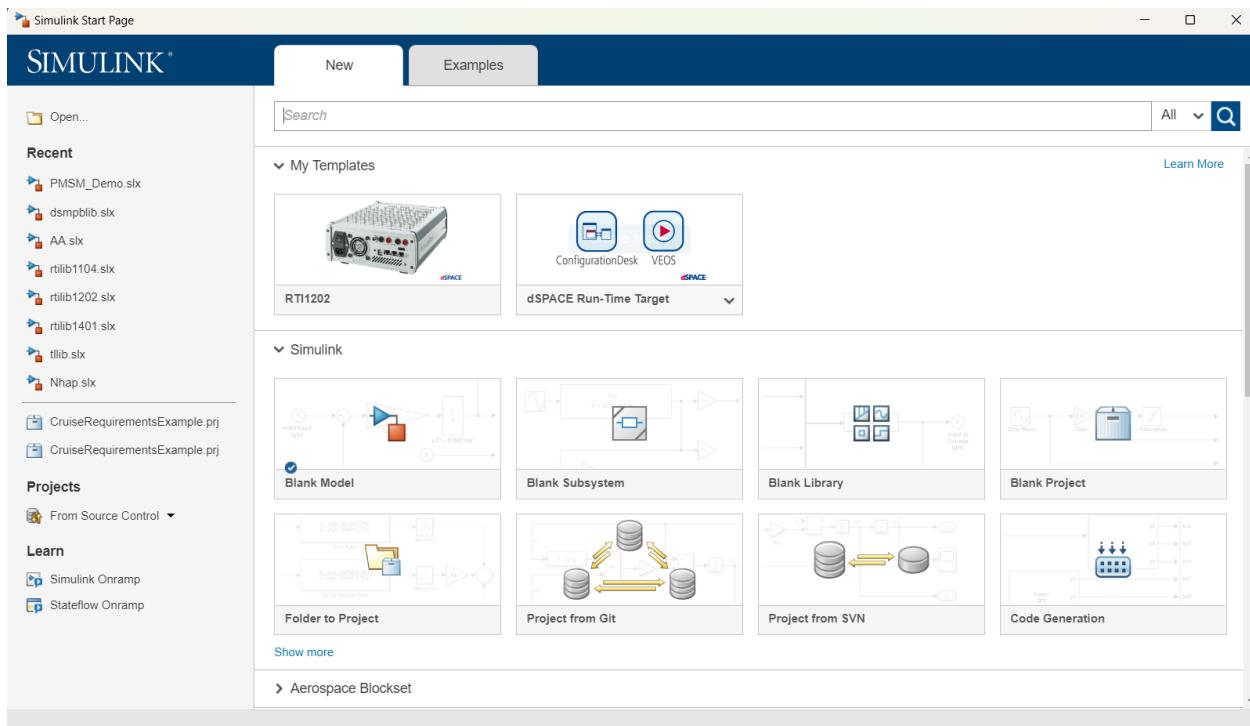
Matlab/Simulink, công cụ được phát triển bởi công ty phần mềm Matworks là một trong những công cụ phổ biến nhất được sử dụng trong lĩnh vực kỹ thuật, đặc biệt là trong mô hình hóa (MBD), mô phỏng và điều khiển hệ thống. Matlab là một môi trường tính toán số và lập trình kỹ thuật mạnh mẽ, trong khi Simulink là một công cụ mô phỏng đồ họa được tích hợp sâu vào Matlab, cho phép người dùng mô hình hóa hệ thống động và thực hiện phân tích, thiết kế và kiểm tra các hệ thống điều khiển.

Các tính năng chính của Matlab bao gồm khả năng thực hiện các phép tính toán phức tạp, xử lý tín hiệu, và phân tích dữ liệu, cũng như hỗ trợ cho nhiều loại dữ liệu và biểu đồ. Nó cũng cung cấp các công cụ lập trình mạnh mẽ để phát triển và triển khai các thuật toán và ứng dụng tương thích với nhiều máy tính được viết bởi đa dạng các loại ngôn ngữ khác nhau.



Hình 3.1. Giao diện chính của Matlab

Simulink cho phép người dùng mô hình hóa hệ thống (MBD) bằng cách sử dụng các khối chức năng và kết nối chúng để tạo ra một biểu đồ dễ hiểu và thân thiện. Nó cũng cung cấp rất nhiều công cụ hỗ trợ cho việc tính toán, ước tính tham số, so sánh số liệu, đánh giá kết quả phục vụ cho quá trình thiết kế, mô phỏng. Điều này cho phép người dùng mô phỏng và kiểm tra các hệ thống phức tạp trước khi triển khai chúng trong thực tế.



Hình 3.2. Giao diện chính của Simulink

Trong Simulink, bạn có thể tích hợp các thuật toán MATLAB vào các mô hình và xuất kết quả mô phỏng MATLAB để phân tích thêm. Simulink hỗ trợ thiết kế một hệ thống mô phỏng mà tự động tạo mã để kiểm tra và xác định các hệ thống nhúng. Ngoài ra, có một số phần mềm hỗ trợ khác được cung cấp bởi MathWorks và các sản phẩm phần cứng và phần mềm của bên thứ ba có sẵn để sử dụng với Simulink. Dưới đây là một số phần mềm này:

- Simulink Test: Hỗ trợ tự động hóa kiểm nghiệm mô hình của Simulink.
- Simulink Code: Tự động chuyển đổi mô hình Simulink thành ngôn ngữ C/C++.
- Simulink Design Verifier: Sử dụng để kiểm tra tự động mô hình Simulink để phát hiện lỗi và lập luận về tính chính xác của mô hình.
- Stateflow: Môi trường thiết kế và kiểm thử cho các hệ thống điều khiển dựa trên trạng thái.
- Embedded Coder: Tự động tạo mã C cho các hệ thống nhúng từ mô hình Simulink và MATLAB.
- Matlab Compiler: Chuyển đổi mã MATLAB thành ứng dụng độc lập.
- Simulink Real-time: Cho phép mô hình Simulink chạy trực tiếp trên phần cứng thời gian thực.

Simulink hỗ trợ hầu như toàn bộ các quy trình được áp dụng trong mô hình hóa hệ thống (MBD), bao gồm:

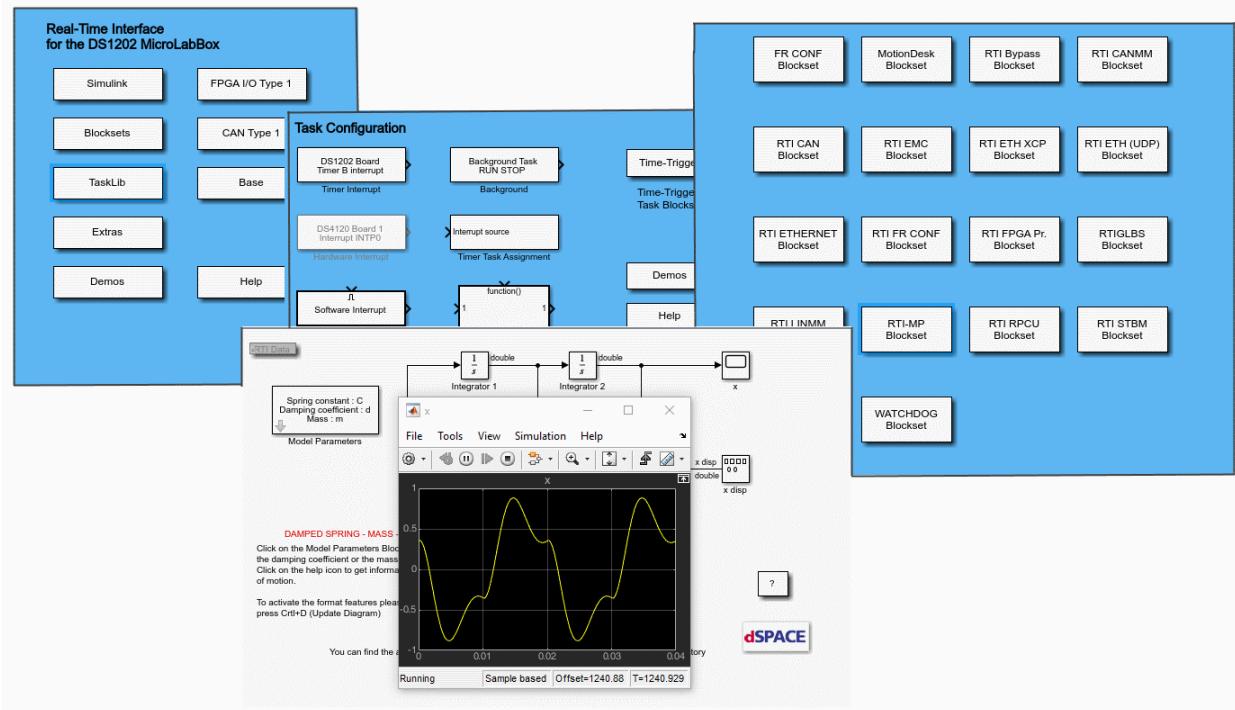
- Cấu trúc hệ thống (system-level design): thiết kế toàn bộ hệ thống theo cấp độ cao nhất, bao gồm việc tạo ra các sơ đồ khối để biểu diễn các phần của hệ thống, xác định các liên kết giữa chúng và xây dựng một hình ảnh toàn diện về cách các phần hoạt động cùng nhau.
- Mô phỏng (simulation): Simulink cung cấp một môi trường mô phỏng đa miền, thử nghiệm hoạt động của hệ thống trong các điều kiện khác nhau mà không cần phải triển khai trên phần cứng thực tế. Bằng cách này, bạn có thể đánh giá hiệu suất và kiểm tra tính đúng đắn của hệ thống trước khi triển khai.
- Tự động tạo mã (auto code generation): Simulink có khả năng tự động tạo mã từ mô hình thiết kế. Một số công cụ hỗ trợ như Simulink Code, Embedded Coder,... Điều này có nghĩa là có thể chuyển đổi mô hình thành mã ngôn ngữ lập trình như C hoặc C++ một cách tự động, giúp tiết kiệm thời gian và giảm thiểu lỗi trong quá trình triển khai.
- Xác nhận và đánh giá (Verification & Validation): Simulink cho phép kiểm tra và xác minh tính đúng đắn của hệ thống nhúng của mình thông qua việc tạo ra các tập lệnh kiểm tra tự động, mô phỏng hành vi của hệ thống trong các điều kiện khác nhau và phân tích kết quả để đảm bảo rằng hệ thống hoạt động như mong đợi và đáp ứng được các yêu cầu cụ thể. Simulink Test, Harness Test,... là những công cụ tuyệt vời để giúp làm những công việc tạo ra các trường hợp kiểm tra và đánh giá này.

3.2. Tổng quan về dSPACE Software

3.2.1. Real-time Interface (RTI)

Real-time Interface (RTI) là một công cụ giúp chúng ta tập trung hoàn toàn vào quá trình thiết kế thực sự và thực hiện các vòng lặp thiết kế nhanh chóng. Nó mở rộng trình biên dịch C Simulink CoderTM (trước đây là Real-Time Workshop[®]) để triển khai mô hình Simulink và Stateflow của chúng ta trên phần cứng thời gian thực một cách mượt mà và tự động.

Với Real-time Interface, ta có thể xây dựng các ứng dụng thời gian thực cho Rapid Control Prototyping (RCP) hoặc là Hardware-in-the-loop (HIL). Đây có thể coi là một sự liên kết giữa phần cứng dSPACE và các phần mềm phát triển cụ thể là MATLAB/Simulink.



Hình 3.3. RTI sử dụng cho MicroLabBox

Các khối RTI sẽ bao gồm:

- Mục đích chung, như xử lý các tác vụ.
- Triển khai ứng dụng cho hệ thống đa nhân và đa bộ xử lý.
- Kết nối mô hình với các bo mạch I/O của dSPACE.
- Cung cấp các tính năng đặc biệt, như giao tiếp qua bus, lập trình FPGA hoặc điều khiển động cơ điện (việc hỗ trợ phụ thuộc vào phần cứng của dSPACE).

Với các khối RTI, việc mô phỏng các hệ thống cũng như kết nối với các phần cứng dSPACE sẽ trở nên thuận lợi và dễ dàng hơn.

3.2.2. ControlDesk

ControlDesk là một phần mềm ứng dụng của dSPACE, được tích hợp các chức năng thường đòi hỏi nhiều công cụ chuyên biệt. Nó cung cấp quyền truy cập vào các nền tảng mô phỏng và hệ thống bus được kết nối, đồng thời có thể thực hiện đo lường, hiệu chuẩn và chẩn đoán trên ECU, ví dụ như thông qua giao diện ASAM tiêu chuẩn. Cấu trúc module linh hoạt của ControlDesk mang lại khả năng mở rộng cao để đáp ứng các yêu cầu của

từng trường hợp ứng dụng cụ thể. Điều này mang lại những lợi thế rõ ràng về mặt xử lý, lượng đào tạo cần thiết, công suất máy tính cần thiết và chi phí.

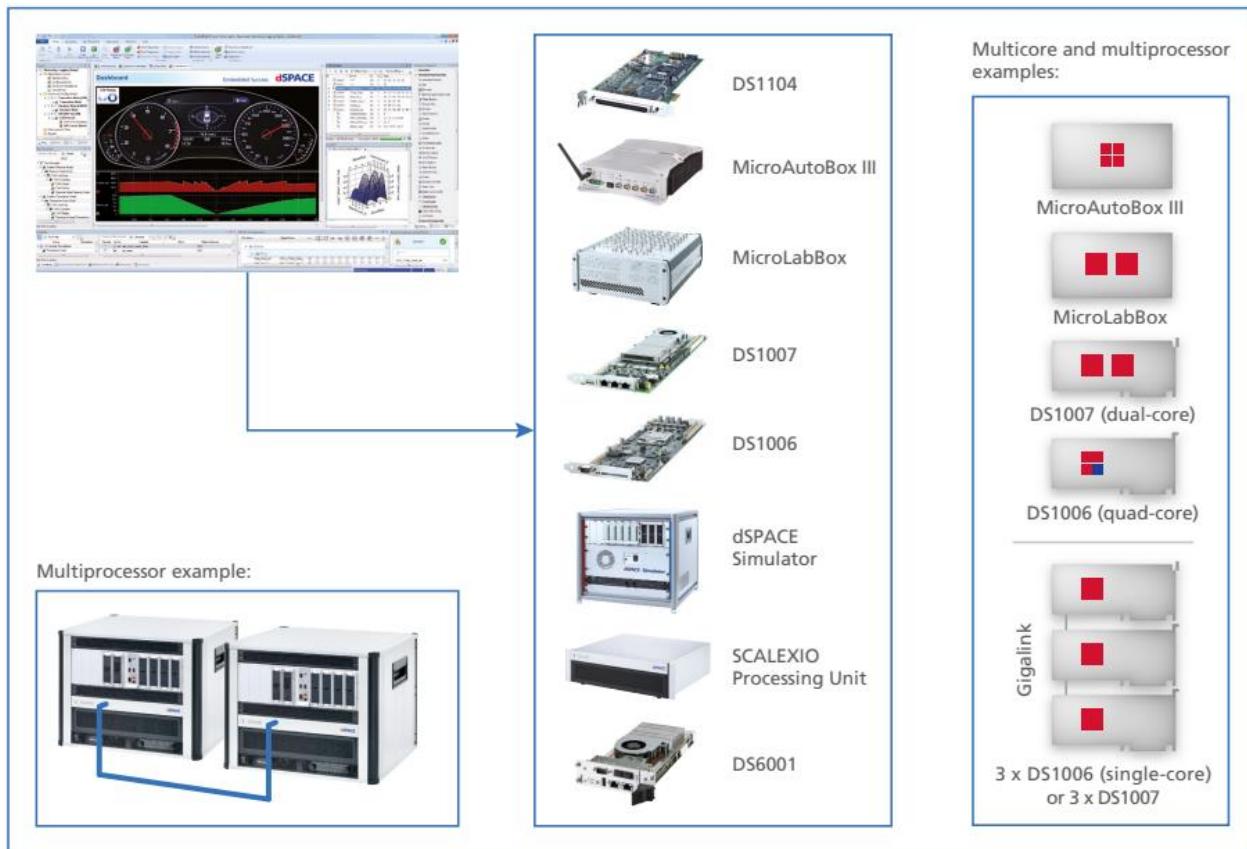
ControlDesk phù hợp với nhiều trường hợp sử dụng, ví dụ như:

- Rapid Control Prototyping (RCP) (fullpass, bypass)
- Hardware-in-the-loop Simulation (HIL)
- Đo lường, hiệu chỉnh và tính toán ECU
- Truy cập vào hệ thống bus trên xe (CAN, CAN FD, LIN, Ethernet)
- Xác thực ảo với các sản phẩm dSPACE như VEOS, SCALEOX



Hình 3.4. Ứng dụng của ControlDesk

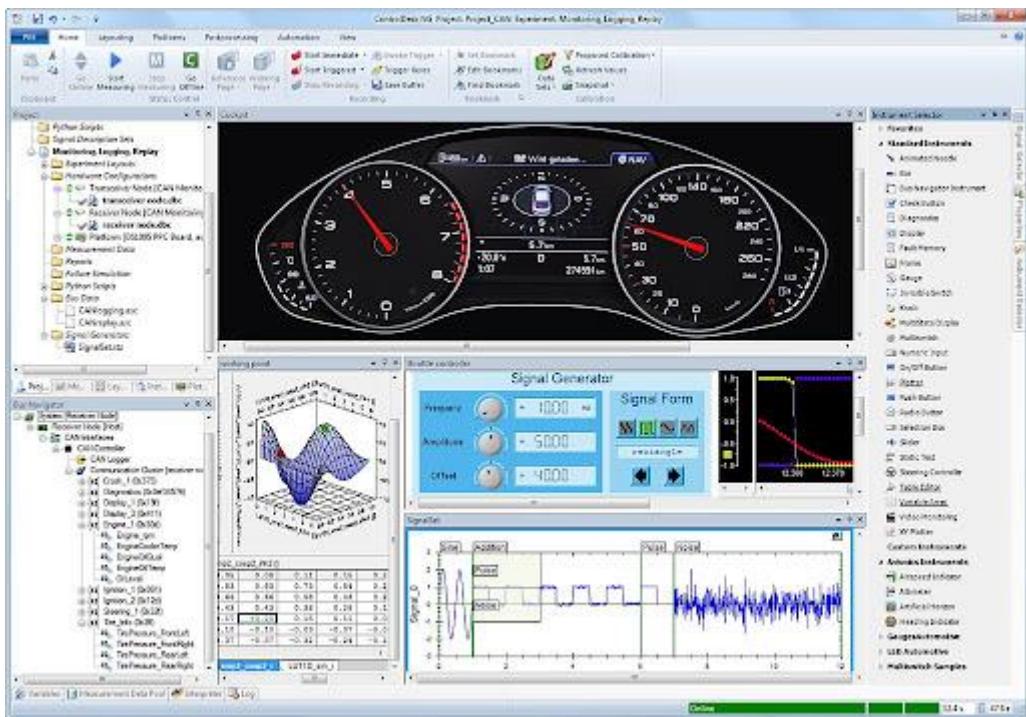
ControlDesk cung cấp khả năng truy cập đến các phần cứng thời gian thực của dSPACE, hỗ trợ các ứng dụng cho bộ xử lí đơn và thiết lập bộ xử lí đa lõi phức tạp hơn cho các ứng dụng cần nhiều công suất, các mô phỏng lớn. Với các ứng dụng cần công suất tính toán cao, có thể dùng 2 bộ xử lí cùng một lúc, ví dụ như kết nối cùng lúc với hai phần cứng SCALEXIO. Với việc truy cập này, có thể thuận tiện thực hiện các phép đo và điều chỉnh các giá trị tham số trên phần cứng của dSPACE trực tiếp từ ControlDesk.



Hình 3.5. ControlDesk hỗ trợ truy cập một số phần cứng dSPACE

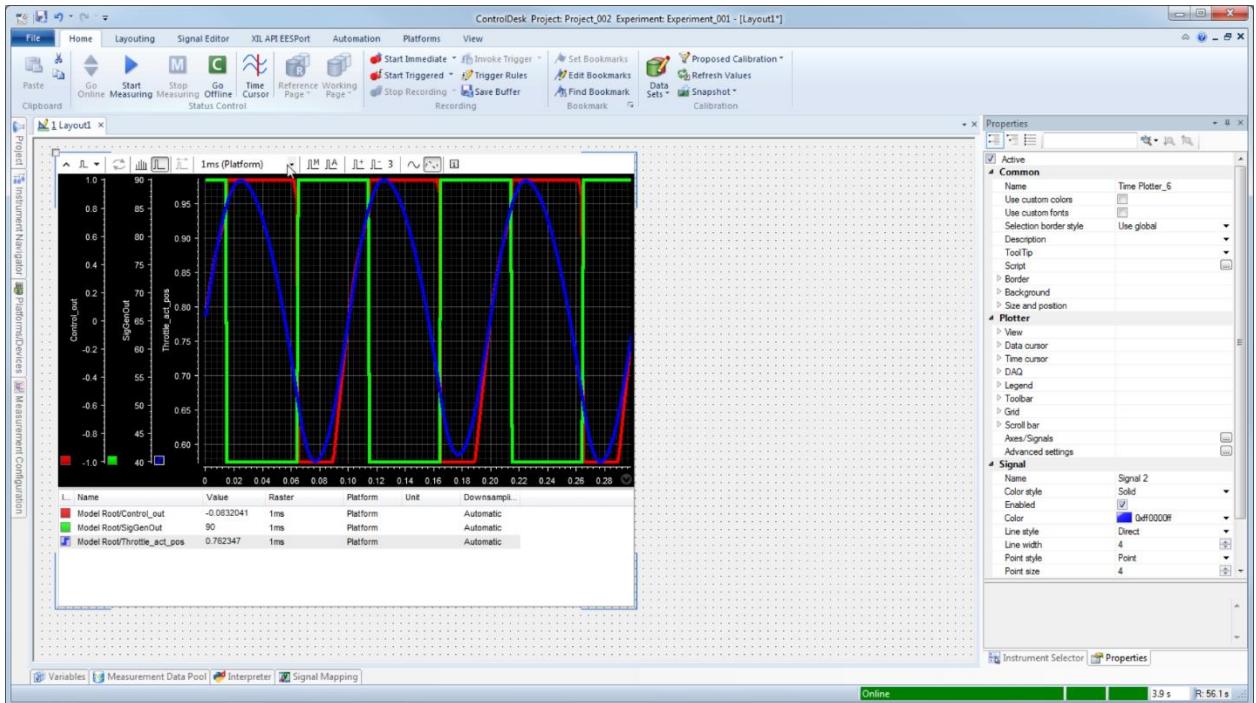
Ứng dụng của ControlDesk vào Rapid Control Prototyping (RCP):

- Thiết kế giao diện người dùng (GUI):** có thể tạo các bảng điều khiển, đồ thị, biểu tượng, thanh trượt, nút nhấn và các yếu tố giao diện khác để tương tác với hệ thống và theo dõi dữ liệu thời gian thực. Khả năng tùy chỉnh cao của controlDesk cho phép tạo ra các giao diện người dùng trực quan và dễ sử dụng, phù hợp với nhu cầu cụ thể của từng dự án RCP.



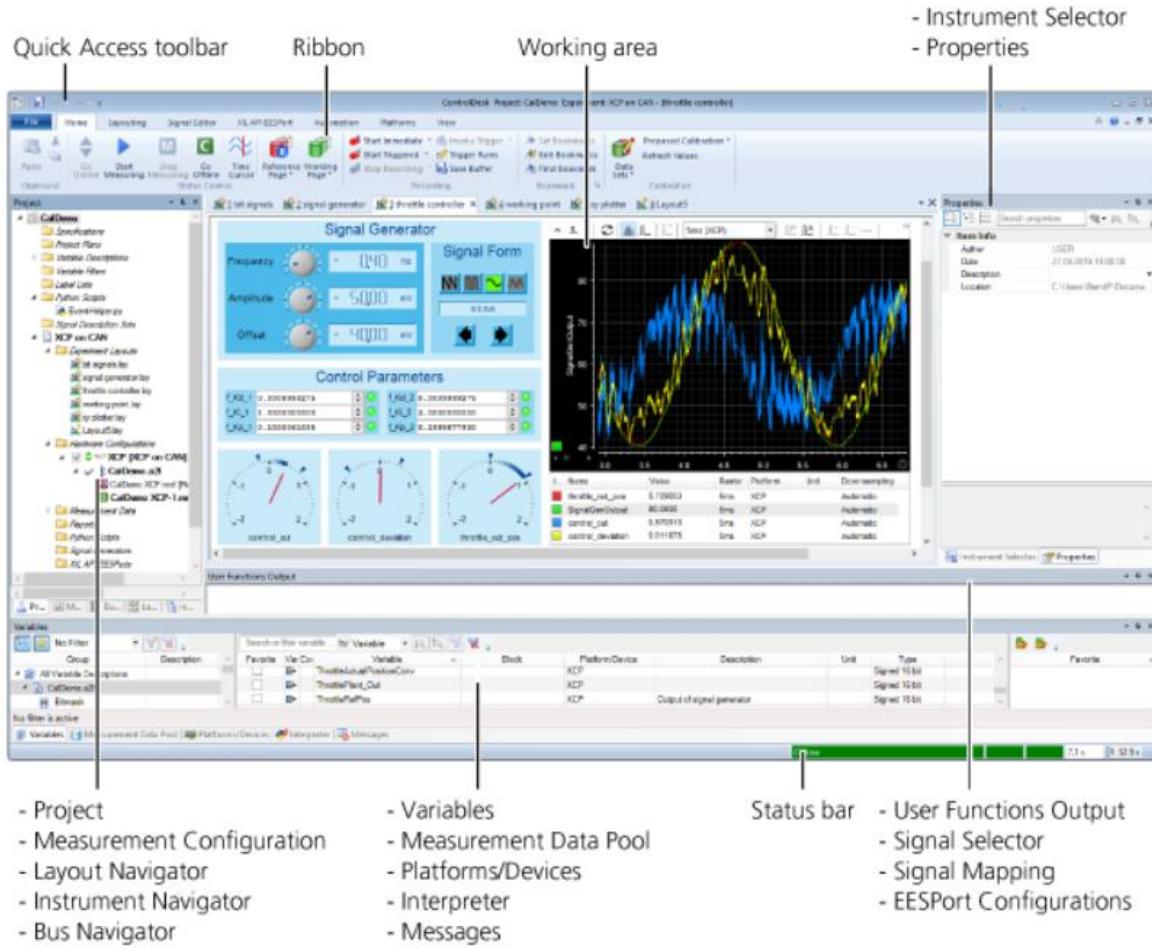
Hình 3.6. Giao diện GUI trên ControlDesk

- Phát triển và tích hợp thuật toán điều khiển:** sử dụng các ngôn ngữ lập trình như Python, C++ hoặc MATLAB để viết mã cho các thuật toán điều khiển, sau đó tích hợp chúng vào giao diện người dùng. Điều này giúp đơn giản hóa quá trình phát triển và thử nghiệm các thuật toán điều khiển, cho phép nhanh chóng lặp lại và tối ưu hóa hiệu suất hệ thống.
- Giao tiếp và thu thập dữ liệu:** có thể sử dụng các giao thức truyền thông như RS-232, USB, CAN hoặc Ethernet để kết nối với các thiết bị này và thu thập dữ liệu thời gian thực. Dữ liệu thu thập được có thể được hiển thị trực tiếp trên giao diện người dùng hoặc lưu trữ để phân tích sau.



Hình 3.7. Thu thập dữ liệu bằng độ thị trong ControlDesk

- Mô phỏng và thử nghiệm hệ thống:** có thể tạo các mô hình mô phỏng cho các thành phần khác nhau của hệ thống, bao gồm cả mô hình cơ cấu chấp hành và mô hình bộ điều khiển. Mô phỏng cho phép thử nghiệm các thuật toán điều khiển và xác định các vấn đề tiềm ẩn trước khi kết nối với phần cứng thực tế, tiết kiệm thời gian và chi phí.
- Triển khai và vận hành hệ thống:** có thể tạo các tệp mã nhúng để chạy thuật toán điều khiển trên bộ vi điều khiển hoặc các thiết bị phần cứng khác. controlDesk cũng cung cấp các công cụ để giám sát và điều khiển hệ thống từ xa, giúp vận hành hệ thống hiệu quả và dễ dàng.



Hình 3.8. Tổng quan giao diện ControlDesk

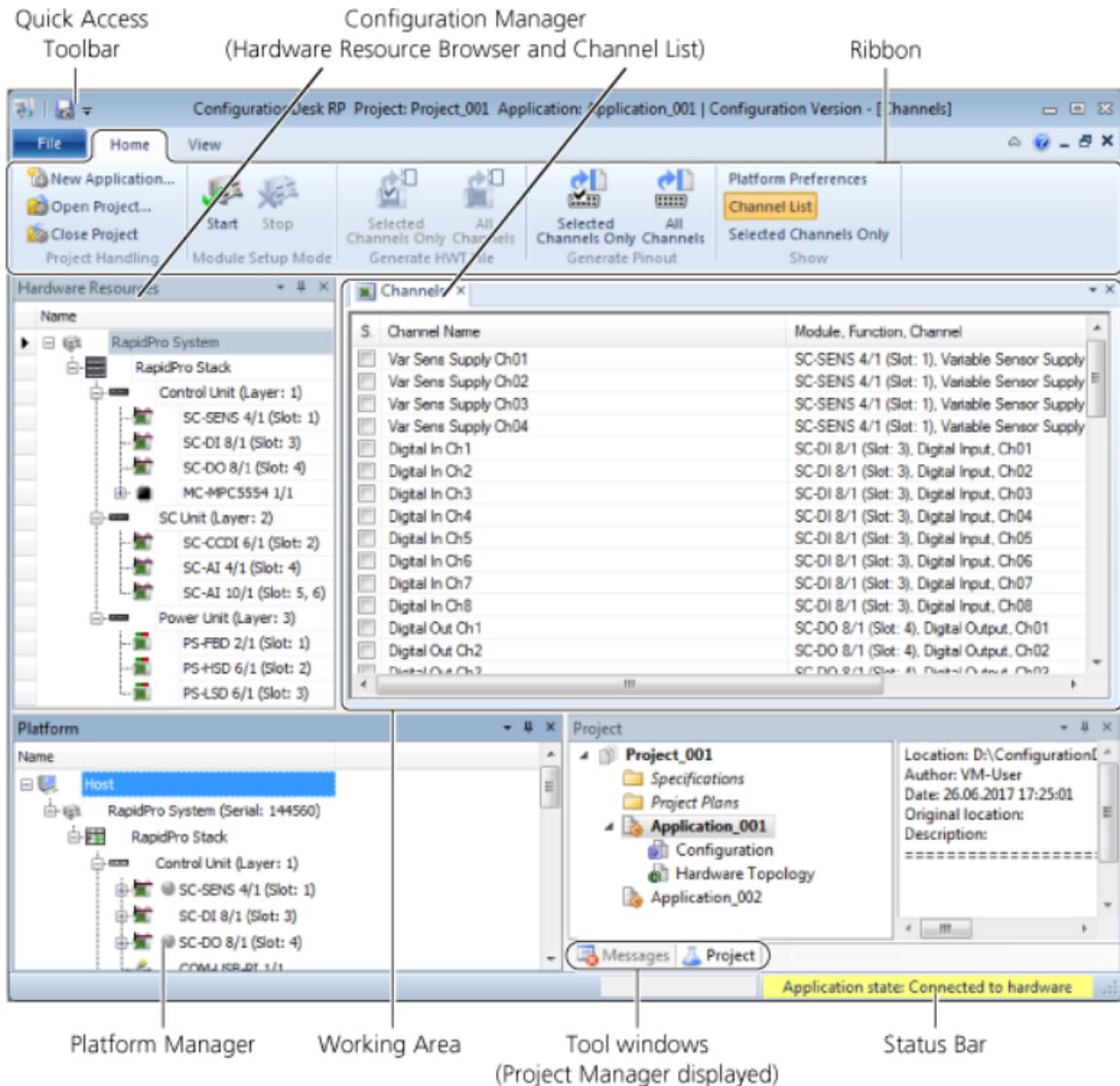
3.2.3. ConfigurationDesk for RapidPro

ConfigurationDesk for RapidPro là một phần mềm cho phép dễ dàng cấu hình và giám sát phần cứng của RapidPro, xuất dữ liệu, thiết lập các dự án và ứng dụng. Phần mềm này giúp điều chỉnh, quản lý các module hiệu chỉnh tín hiệu (SC module) hay module cấp nguồn (PS module) phù hợp với từng mục đích, yêu cầu của ứng dụng. Dữ liệu có thể được xuất ra sử dụng cho ứng dụng khác (RTI) hoặc để người dùng tham khảo và các dữ liệu đã cấu hình cho một module sẽ được lưu trữ.

Configuration for RapidPro cung cấp một bộ tính năng hữu ích cho việc cấu hình phần cứng RapidPro:

- **Cấu hình phần cứng RapidPro:** Cấu hình các thiết bị RapidPro, mô-đun và các kênh của chúng. Có thể cấu hình ngay cả khi phần cứng chưa được kết nối.

- **Xử lý chẩn đoán:** Hiển thị các thông báo chẩn đoán về kết nối của cảm biến và bộ cơ cấu chấp hành, chẳng hạn như quá dòng, ngắn mạch, quá tải, quá nhiệt, quá áp và dưới áp. Điều này giúp dễ dàng phát hiện và xác định vị trí lỗi.
- **Kết nối RapidPro (Hot plugging):** Phàn cứng có thể được kết nối và ngắt kết nối trong khi chương trình đang chạy. Việc kết nối phàn cứng RapidPro sẽ khởi chạy quá trình quét phàn cứng tự động, giúp phát hiện phàn cứng được kết nối. Thông tin này sẽ ngay lập tức được hiển thị trong Trình quản lý nền tảng (Platform Manager).
- **Xác định phàn cứng:** Trình quản lý nền tảng của ConfigurationDesk cung cấp thông tin chi tiết về cấu trúc phàn cứng (bao gồm ID cấu trúc phàn cứng - TopologyID), các mô-đun SC và PS được sử dụng, chẳng hạn như số serial của chúng.
- **Quản lý dự án:** Trình quản lý dự án của ConfigurationDesk cho phép tổ chức tất cả thông tin dự án liên quan, chẳng hạn như cấu hình và dữ liệu cụ thể của ứng dụng.
- **Giám sát giá trị tín hiệu và hệ thống:** Giám sát giá trị tín hiệu (trạng thái logic, điện áp đầu vào/ra) hỗ trợ bạn kết nối cảm biến và bộ truyền động với phàn cứng RapidPro khi hệ thống được đưa vào vận hành. Giám sát giá trị hệ thống (điện áp bên trong, nhiệt độ) phục vụ mục đích khắc phục sự cố.
- **Xuất dữ liệu:** có thể xuất ác file HWT để giới thiệu phàn cứng RapidPro với RTI RapidPro Control Unit Blockset hoặc thông tin ghim cần thiết để chế tạo dây cáp.



Hình 3.9. Giao diện chính của Configuration for RapidPro

3.3. Tổng quan về dSPACE MicroLabBox và RapidPro System

3.3.1. MicroLabBox

MicroLabBox là một công cụ phát triển hệ thống của dSPACE dành cho các hoạt động nghiên cứu, phát triển trong phòng thí nghiệm với sự nhỏ gọn, hiệu suất cùng độ linh hoạt cao. Với MicroLabBox, chúng ta có thể thiết lập các ứng dụng điều khiển, kiểm thử hoặc đo lường một cách nhanh chóng và hiệu quả.

3.3.1.1. Tổng quan MicroLabBox



Hình 3.10. MicroLabBox của dSPACE

MicroLabBox là một hệ thống RCP nhỏ gọn, dễ dàng và an toàn khi sử dụng. MicroLabBox được thiết kế để dễ dàng thao tác trong môi trường phòng thí nghiệm. Ví dụ, nó có thể được kết nối trực tiếp với nguồn điện mà không cần sử dụng thêm nguồn cấp điện hay biến áp. Ngoài ra, hệ thống này cung cấp các giao diện tiêu chuẩn đến các thiết bị bên ngoài như kết nối Ethernet và USB. Để tạo và đo các tín hiệu I/O, bo mạch cung cấp các kênh đầu vào và đầu ra dạng analog và digital với tích hợp điều hòa tín hiệu.

MicroLabBox bao gồm hai bảng mạch:

1. DS1202

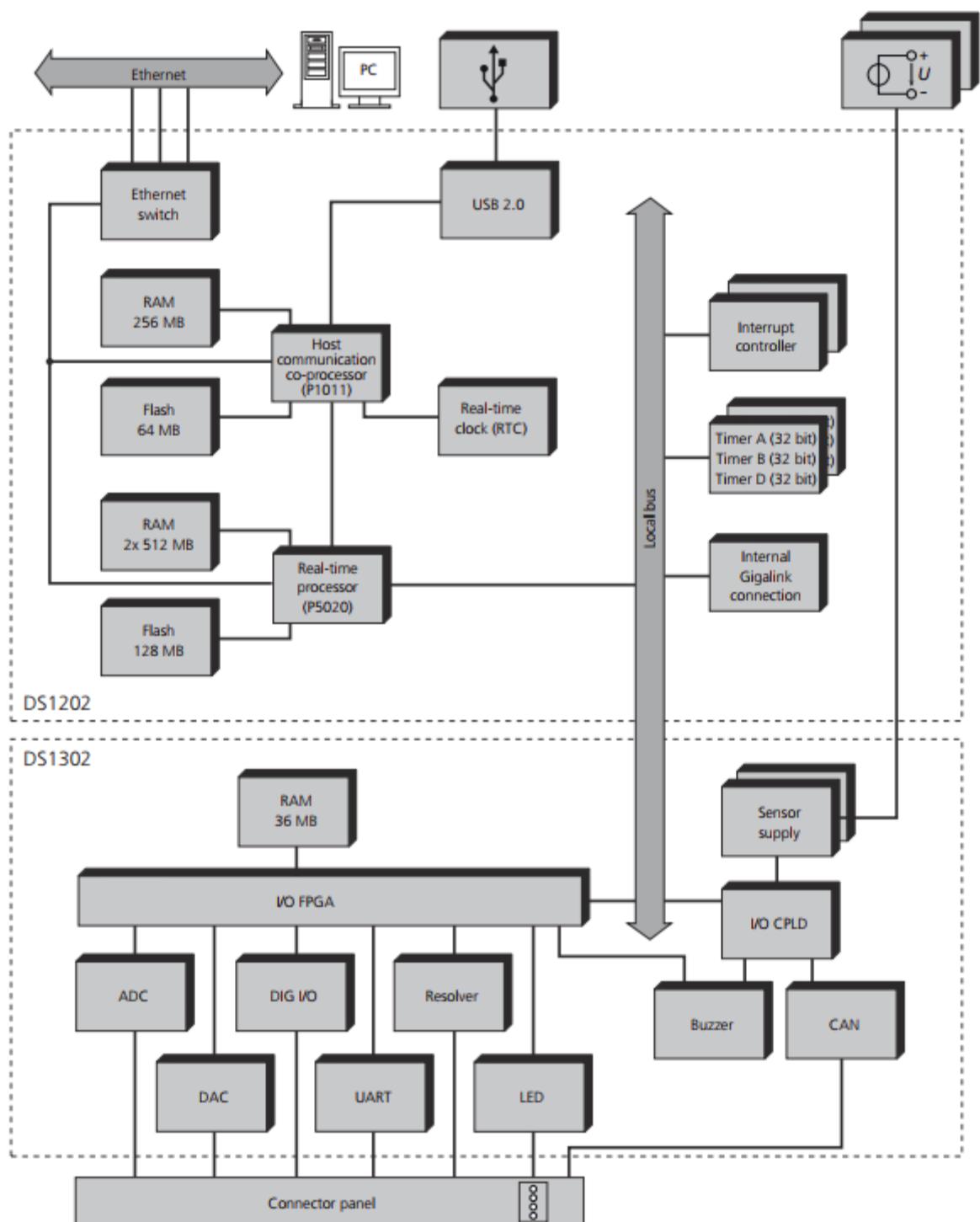
Đây là bảng mạch chính của MicroLabBox, dựa trên công nghệ Freescale Power Architecture®. Nó cung cấp các tính năng về giao tiếp và tính toán. Bảng DS1202 sẽ kiểm soát:

- Giao diện Ethernet, bao gồm cấu hình switch cho giao tiếp với máy chủ và truy cập I/O
- Giao diện USB để ghi dữ liệu và khởi động ứng dụng thông qua thiết bị lưu trữ USB
- Quản lý flash để khởi động MicroLabBox và tải các ứng dụng thời gian thực từ bộ nhớ flash
- Giao tiếp và trao đổi dữ liệu với Bảng I/O DS1302

2. DS1302

Đây là bảng mạch I/O của MicroLabBox. Nó cung cấp các tính năng I/O tiêu chuẩn của bảng mạch. Bảng DS1302 sẽ kiểm soát:

- Các kênh I/O Analog
- Các kênh I/O Digital
- Giao diện nối tiếp (RS232 và RS422/485)
- Giao diện CAN
- Hai đầu ra cấp nguồn cho cảm biến khác nhau
- Các đèn LED tùy chỉnh
- Còi báo hiệu
- Giao diện resolver



Hình 3.11. Sơ đồ 2 board DS1202 và DS1302

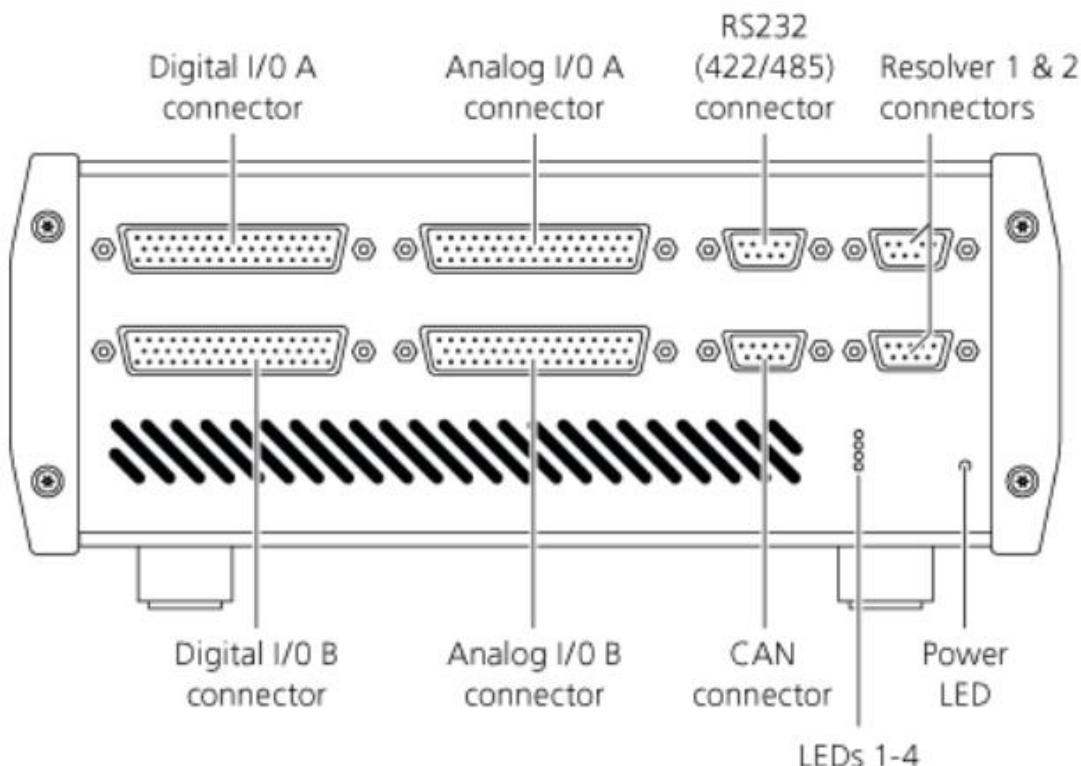
Thông số kỹ thuật MicroLabBox phần cứng:

- Kích thước vật lí: 310 x 250 x 110 mm, 4.5 kg.
- Điện áp nguồn: 100 ~ 240 V AC, 50 ~ 60 Hz.

- Công suất tiêu thụ tối đa: 125W.
- Bộ nhớ: 1GB DRAM, 128 MB flash memory.
- Sử dụng 2 board cơ bản: DS1202 và DS1302.

3.3.1.2. Mặt trước MicroLabBox

Có các cổng kết nối I/O cho các tín hiệu Digital, Analog, CAN, Resolver, các kết nối UART cũng như các LED hiển thị trạng thái.



Hình 3.12. Các cổng kết nối ở mặt trước của MicroLabBox

3.3.1.2.1. Analog I/O

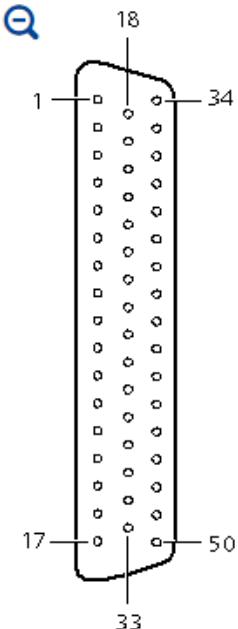
MicroLabBox cung cấp hai đơn vị chuyển đổi A/D với các đặc điểm khác nhau: ADC Class 1 và ADC Class 2. Ngoài ra, MicroLabBox còn cung cấp một đơn vị chuyển đổi D/A là DAC Class 1.

Bảng 3.1. Thông số ADC Class 1 và ADC Class 2

	ADC Class 1	ADC Class 2	DAC Class 1
Độ phân giải	16 bit	14 bit	16 bit
Số kênh chuyển đổi A/D	24 kênh	8 kênh	16 kênh
Tốc độ lấy mẫu	1MSPS	10 MSPS	-

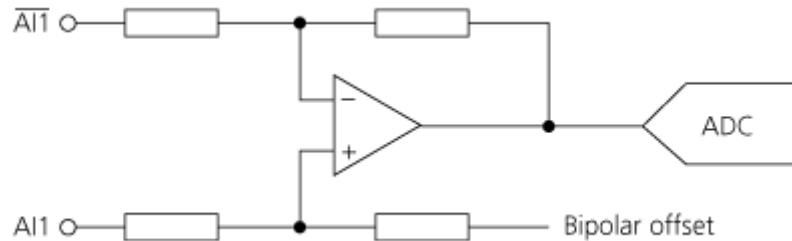
Khoảng điện áp đầu vào	-10 ~ 10 V	-10 ~ 10 V	-
Khoảng điện áp đầu ra	-	-	-10 ~ 10 V
Dòng điện đầu ra	-	-	-8 ~ 8 mA

Các kênh ADC và DAC sẽ được chia đều vào 2 Connector A và B để đưa ra các chân giao tiếp đến các thiết bị ngoại vi.

Analog I/O A Connector	Pin	Signal	Pin	Signal	Pin	Signal
	1	GND			34	AI1 ch24
	2	AI1 ch23	18	AI1 ch24	35	AI1 ch23
	3	AI1 ch22	19	AI1 ch22	36	AI1 ch21
	4	AI1 ch20	20	AI1 ch21	37	AI1 ch20
	5	AI1 ch19	21	AI1 ch19	38	AI1 ch18
	6	AI1 ch17	22	AI1 ch18	39	AI1 ch17
	7	AI1 ch16	23	AI1 ch16	40	AI1 ch15
	8	AI1 ch14	24	AI1 ch15	41	AI1 ch14
	9	AI1 ch13	25	AI1 ch13	42	AI1 ch12
	10	AI1 ch11	26	AI1 ch12	43	AI1 ch11
	11	AI1 ch10	27	AI1 ch10	44	AI1 ch9
	12	AI1 ch8	28	AI1 ch9	45	AI1 ch8
	13	AI1 ch7	29	AI1 ch7	46	AI1 ch6
	14	AI1 ch5	30	AI1 ch6	47	AI1 ch5
	15	AI1 ch4	31	AI1 ch4	48	AI1 ch3
	16	AI1 ch2	32	AI1 ch3	49	AI1 ch2
	17	AI1 ch1	33	AI1 ch1	50	GND

Hình 3.13. Các pin analog I/O cỗng kết nối A

Trong đó: $AI1 ch < x >$ là đầu vào analog không đảo, còn $\overline{AI1 ch < x >}$ cho đầu vào analog đảo ngược, nó chính là điện áp tham chiếu cho chân AII .

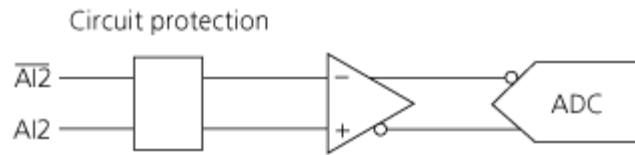


Hình 3.14. Mạch đầu vào đơn giản analog input Class 1

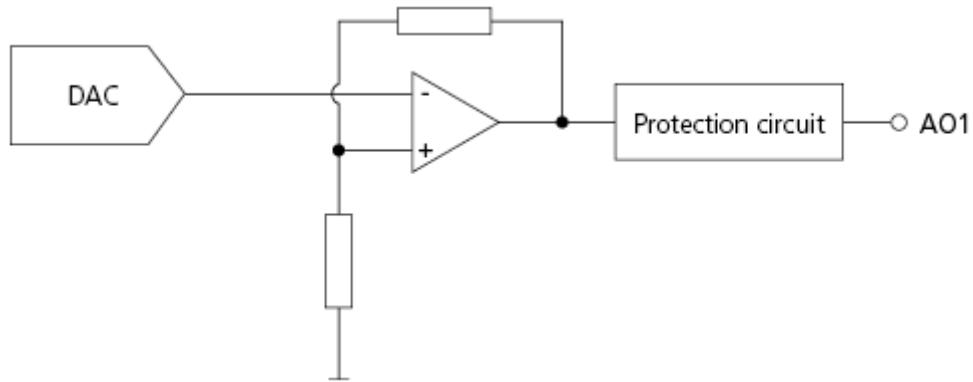
Analog I/O B Connector	Pin	Signal	Pin	Signal	Pin	Signal
	1	GND			34	AO1 ch16ref
	2	AO1 ch15ref	18	AO1 ch16	35	AO1 ch15
	3	AI2 ch8	19	$\overline{AI2 ch8}$	36	AO1 ch14ref
	4	AO1 ch13ref	20	AO1 ch14	37	AO1 ch13
	5	AI2 ch7	21	$\overline{AI2 ch7}$	38	AO1 ch12ref
	6	AO1 ch11ref	22	AO1 ch12	39	AO1 ch11
	7	AI2 ch6	23	$\overline{AI2 ch6}$	40	AO1 ch10ref
	8	AO1 ch9ref	24	AO1 ch10	41	AO1 ch9
	9	AI2 ch5	25	$\overline{AI2 ch5}$	42	AO1 ch8ref
	10	AO1 ch7ref	26	AO1 ch8	43	AO1 ch7
	11	AI2 ch4	27	$\overline{AI2 ch4}$	44	AO1 ch6ref
	12	AO1 ch5ref	28	AO1 ch6	45	AO1 ch5
	13	AI2 ch3	29	$\overline{AI2 ch3}$	46	AO1 ch4ref
	14	AO1 ch3ref	30	AO1 ch4	47	AO1 ch3
	15	AI2 ch2	31	$\overline{AI2 ch2}$	48	AO1 ch2ref
	16	AO1 ch1ref	32	AO1 ch2	49	AO1 ch1
	17	AI2 ch1	33	$\overline{AI2 ch1}$	50	GND

Hình 3.15. Các pin analog I/O công kết nối B

Trong đó, $AO1 ch< x > ref$ là chân cung cấp đầu ra Analog, có nối đất tín hiệu (chân tham chiếu để kết nối cáp phù hợp).



Hình 3.16. Mạch đầu vào đơn giản analog input Class 2



Hình 3.17. Sơ đồ đơn giản analog output Class 1

3.3.1.2.2. Digital I/O

MicroLabBox cung cấp 2 đơn vị chuyển đổi Digital I/O với các đặc điểm khác nhau: DIO Class 1 và DIO Class 2.

- Có thể điều chỉnh điện áp đầu ra ở các mức:
 - 2.5 V
 - 3.3 V
 - 5.0 V

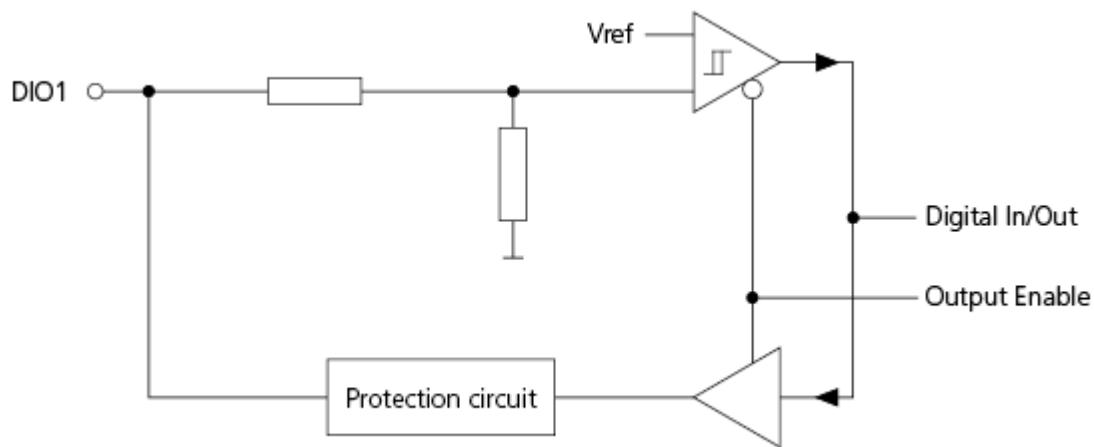
Bảng 3.2. Đặc điểm DIO Class 1 và DIO Class 2

	DIO Class 1	DIO Class 2
Số kênh	48 kênh 2 chiều	12 kênh 2 chiều
Hỗ trợ RTI	Có hỗ trợ	Không hỗ trợ

Digital I/O A	Pin	Signal	Pin	Signal	Pin	Signal
	1	GND			34	DIO1 ch32
	2	DIO1 ch16	18	GND	35	DIO1 ch31
	3	DIO1 ch15	19	GND	36	DIO1 ch30
	4	DIO1 ch14	20	GND	37	DIO1 ch29
	5	DIO1 ch13	21	GND	38	DIO1 ch28
	6	DIO1 ch12	22	GND	39	DIO1 ch27
	7	DIO1 ch11	23	GND	40	DIO1 ch26
	8	DIO1 ch10	24	GND	41	DIO1 ch25
	9	DIO1 ch9	25	GND	42	DIO1 ch24
	10	DIO1 ch8	26	GND	43	DIO1 ch23
	11	DIO1 ch7	27	GND	44	DIO1 ch22
	12	DIO1 ch6	28	GND	45	DIO1 ch21
	13	DIO1 ch5	29	GND	46	DIO1 ch20
	14	DIO1 ch4	30	GND	47	DIO1 ch19
	15	DIO1 ch3	31	GND	48	DIO1 ch18
	16	DIO1 ch2	32	GND	49	DIO1 ch17
	17	DIO1 ch1	33	GND	50	GND

Hình 3.18. Các pin digital I/O kênh A

Trong đó, *DIO1 ch<x>* là chân cung cấp đầu vào hoặc đầu ra digital class 1.

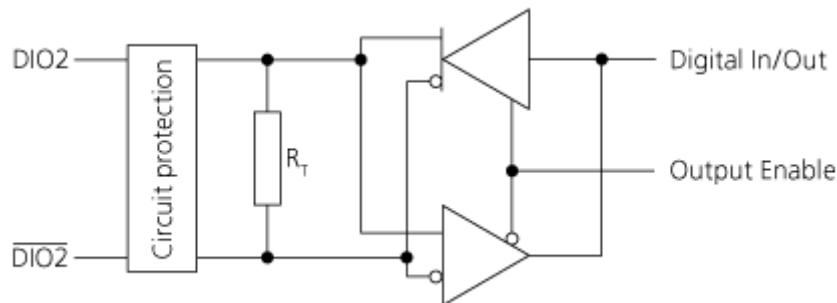


Hình 3.19. Sơ đồ đơn giản kênh 2 chiều Digital Class 1

Digital I/O B	Pin	Signal	Pin	Signal	Pin	Signal
	1	GND	18	GND	34	GND
	2	DIO1 ch48	18	GND	35	<u>DIO2 ch12</u>
	3	DIO1 ch47	19	DIO2 ch12	36	<u>DIO2 ch11</u>
	4	DIO1 ch46	20	DIO2 ch11	37	<u>DIO2 ch10</u>
	5	DIO1 ch45	21	DIO2 ch10	38	GND
	6	DIO1 ch44	22	GND	39	<u>DIO2 ch9</u>
	7	DIO1 ch43	23	DIO2 ch9	40	<u>DIO2 ch8</u>
	8	DIO1 ch42	24	DIO2 ch8	41	<u>DIO2 ch7</u>
	9	DIO1 ch41	25	DIO2 ch7	42	GND
	10	DIO1 ch40	26	GND	43	<u>DIO2 ch6</u>
	11	DIO1 ch39	27	DIO2 ch6	44	<u>DIO2 ch5</u>
	12	DIO1 ch38	28	DIO2 ch5	45	<u>DIO2 ch4</u>
	13	DIO1 ch37	29	DIO2 ch4	46	GND
	14	DIO1 ch36	30	GND	47	<u>DIO2 ch3</u>
	15	DIO1 ch35	31	DIO2 ch3	48	<u>DIO2 ch2</u>
	16	DIO1 ch34	32	DIO2 ch2	49	<u>DIO2 ch1</u>
	17	DIO1 ch33	33	DIO2 ch1	50	GND

Hình 3.20. Các pin digital I/O kênh B

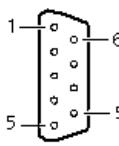
Trong đó $DIO2\ ch< x >$ là chân cung cấp đầu vào hoặc ra digital không đảo, còn chân $\overline{DIO2\ ch < x >}$ cho đầu vào hoặc ra digital đảo ngược.



Hình 3.21. Sơ đồ đơn giản kênh 2 chiều Digital Class 2

RS232 (422/485) Connector	Pin	Pin Name	Signal	Description
	1	RXD ch2	RXD	Channel 2: Receive data
	2	RXD ch1	RXD	Channel 1: Receive data
	3	TXD ch1	TXD	Channel 1: Transmit data
	4	TXD ch2	TXD	Channel 2: Transmit data
	5	GND	—	Signal ground
	6	RTS ch2	RTS	Channel 2: Request to send
	7	RTS ch1	RTS	Channel 1: Request to send
	8	CTS ch1	CTS	Channel 1: Clear to send
	9	CTS ch2	CTS	Channel 2: Clear to send

Hình 3.22. Các pin dành cho tiêu chuẩn kết nối RS232

RS232 (422/485) Connector	Pin	Pin Name	Signal	Description
	1	RXD ch2	\overline{RX}	Channel 2: Receive data, inverted input
	2	RXD ch1	\overline{RX}	Channel 1: Receive data, inverted input
	3	TXD ch1	\overline{TX}	Channel 1: Transmit data, inverted input
	4	TXD ch2	\overline{TX}	Channel 2: Transmit data, inverted input
	5	GND	—	Signal ground
	6	RTS ch2	TX	Channel 2: Transmit data, non-inverted input
	7	RTS ch1	TX	Channel 1: Transmit data, non-inverted input
	8	CTS ch1	RX	Channel 1: Receive data, non-inverted input
	9	CTS ch2	RX	Channel 2: Receive data, non-inverted input

Hình 3.23. Các pin dành cho tiêu chuẩn kết nối RS422

RS232 (422/485) Connector	Pin	Pin Name	Signal	Description
	1	RXD ch2	—	Not connected
	2	RXD ch1	—	Not connected
	3	TXD ch1	BM	Channel 1: Bus minus
	4	TXD ch2	BM	Channel 2: Bus minus
	5	GND	—	Signal ground
	6	RTS ch2	BP	Channel 2: Bus plus
	7	RTS ch1	BP	Channel 1: Bus plus
	8	CTS ch1	—	Not connected
	9	CTS ch2	—	Not connected

Hình 3.24. Các pin dành cho tiêu chuẩn kết nối RS485

Resolver Connector	Pin	Signal	Pin	Signal
	1 2 3 4 5	GND COS SIN EXC GND	6 7 8 9	$\overline{\text{COS}}$ $\overline{\text{SIN}}$ GND $\overline{\text{EXC}}$

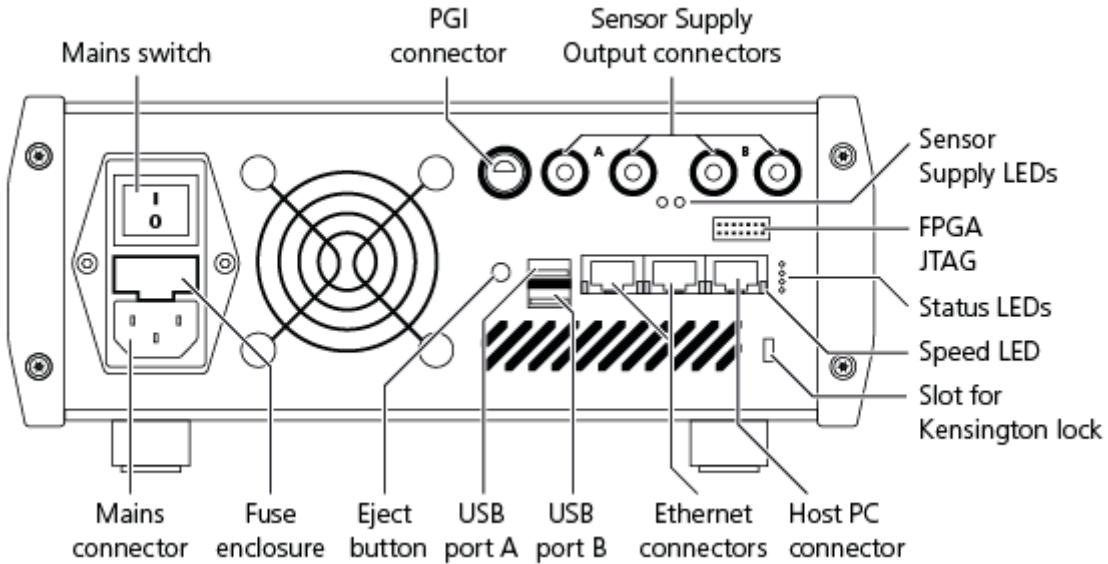
Hình 3.25. Các pin dành cho Resolver

CAN Connector	Pin	Signal	Pin	Signal
	1 2 3 4 5	Not connected CANL ch1 GND CANL ch2 Not connected	6 7 8 9	Not connected CANH ch1 CANH ch2 Not connected

Hình 3.26. Các pin dành cho kết nối giao tiếp CAN

3.3.1.3. Mặt sau MicroLabBox

Có các công tắc và các cổng kết nối với USB, với cảm biến, cổng kết nối Host PC, Ethernet, các LED hiển thị trạng thái.



Hình 3.27. Các cổng kết nối ở mặt trước của MicroLabBox

- **Mains Switch & mains connector:** công tắc và vị trí dây nguồn 220V.
- **Fuse enclosure:** vị trí cầu chì và nắp đậy bảo vệ.
- **USB port A/B:** MicroLabBox cung cấp một đầu nối USB 2.0 với hai cổng. Cổng USB (A) có thể được sử dụng để ghi lại dữ liệu và khởi động một ứng dụng. Cổng USB thứ hai (B) được dự phòng dành cho việc sử dụng trong tương lai.
- **Eject button:** Trước khi ngắt kết nối thiết bị USB, hãy nhấn nút Eject. Khi đèn LED của nút Eject tắt thì ta có thể ngắt kết nối thiết bị USB một cách an toàn.
- **PGI Connector:** Cổng kết nối PGI. PGI là viết tắt của Programmable Generic Interface là một giải pháp về phần cứng và phần mềm của dSPACE cho phép người dùng sử dụng các cảm biến và cơ cấu chấp hành với các giao diện và giao thức khác nhau kết nối đến hệ thống.
- **FPGA/JTAG:** các chân thiết lập FPGA và giao diện JTAG cho phép người dùng chuẩn đoán lỗi.
- **Sensor Supply/Output connectors:** Có 2 kênh để MicroLabBox cấp nguồn cho cảm biến và các thiết bị bên ngoài

Bảng 3.3. Thông số các loại nguồn cảm biến từ MicroLabBox

Loại nguồn cảm biến	Thang đo	Điều kiện	Min.	Typ.	Max.	Đơn vị
----------------------------	-----------------	------------------	-------------	-------------	-------------	---------------

Fixed voltage supply	Điện áp	$I_{out} = 0 A$	11.2	12	12.6	V
		$I_{out} = 0.25A$	11.0			V
	Dòng điện				0.25	A
	Nguồn				3	W
Adjustable Voltage Supply	Điện áp		2		20	V
	Dòng điện				0.2	A
	Nguồn				1	W
	Độ sai lệch	$V_{out} > 3 V$		2	5	%

3.3.1.4. Kết nối MicroLabBox với Host PC

Để có thể sử dụng MicroLabBox cho hoạt động phát triển hệ thống, ta cần kết nối với một host PC để có thể truy cập và nạp các ứng dụng vào MicroLabBox, các bước thiết lập như sau:

- **Bước 1: Chuẩn bị host PC**

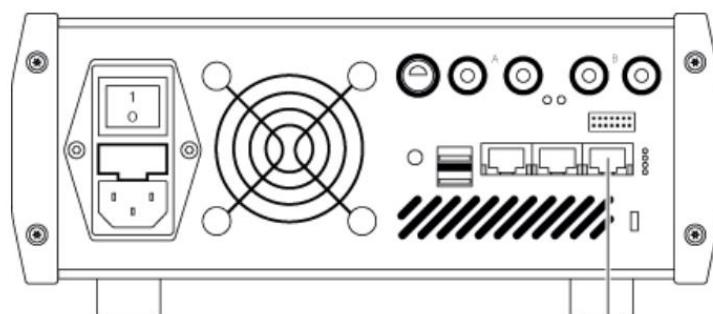
Đặt IPv4 address: Vì IP mặc định của MicroLabBox là 192.168.140.7 nên IP máy Host sẽ trong khoảng 192.168.140.1 → 192.168.140.254 (subnet mask: 255.255.255.0)

Các phần mềm cần chuẩn bị trên host PC để hoạt động với MicroLabBox:

1. Matlab: Simulink
2. dSPACE: RTI (Real-Time Interface), a C-code cross compiler
3. ControlDesk khi cần chạy thực nghiệm với real-time application.

- **Bước 2: Kết nối**

Vì việc giao tiếp của MicroLabBox và PC là giao tiếp Ethernet nên ta cần kết nối dây Ethernet vào cả hai thiết bị.

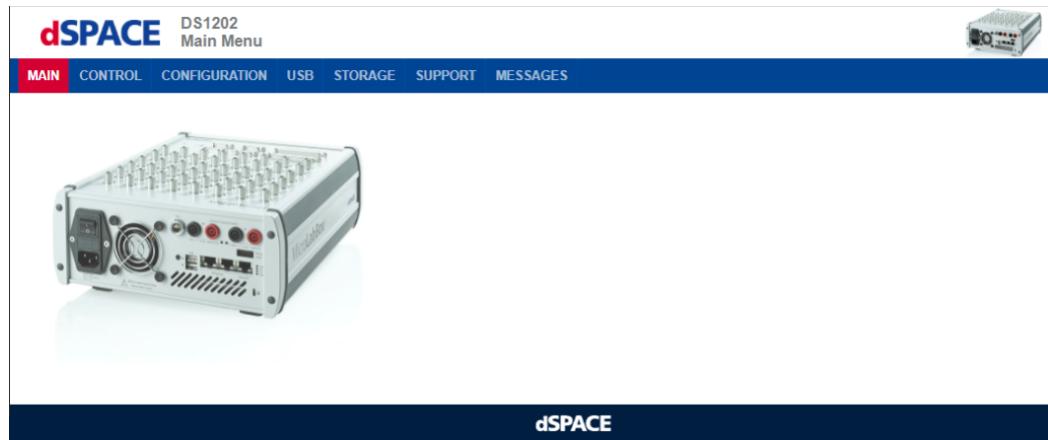


Hình 3.28. Cổng kết nối Ethernet trên MictoLabBox

- Bước 3: Kiểm tra kết nối**

Kết nối dây nguồn MicroLabBox vào nguồn. Bật Switch. MicroLabBox bắt đầu boot, sau khi boot xong sẽ hiện thị LED green.

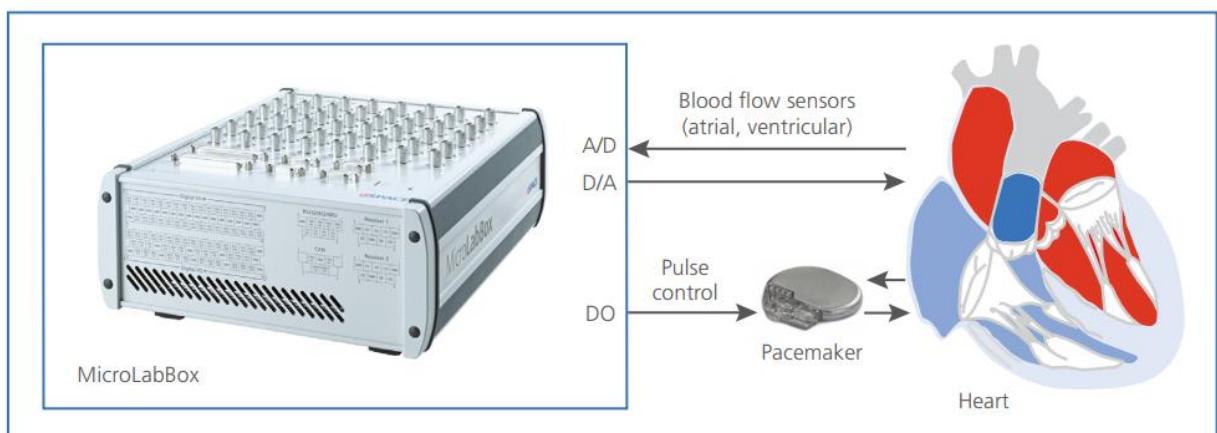
Ở Host PC, mở trình duyệt và nhập địa chỉ IP của MicroLabBox, nếu có giao diện web hiện lên như bên dưới thì đã kết nối thành công và có thể bắt đầu sử dụng.



Hình 3.29. Giao diện web của MicroLabBox

e) Một số lĩnh vực ứng dụng MicroLabBox

- Nghiên cứu phát triển điều khiển động cơ điện: MicroLabBox là công cụ lý tưởng để phát triển các thuật toán điều khiển các động cơ điện như động cơ không đồng bộ, BLDC, PMSM.
- Thử nghiệm và phát triển các thiết bị y tế: Các thiết bị y tế được phát triển bằng MicroLabBox luôn đảm bảo sự an toàn với độ tin cậy và chính xác cao.



Hình 3.30. Ứng dụng MicroLabBox trong lĩnh vực y tế

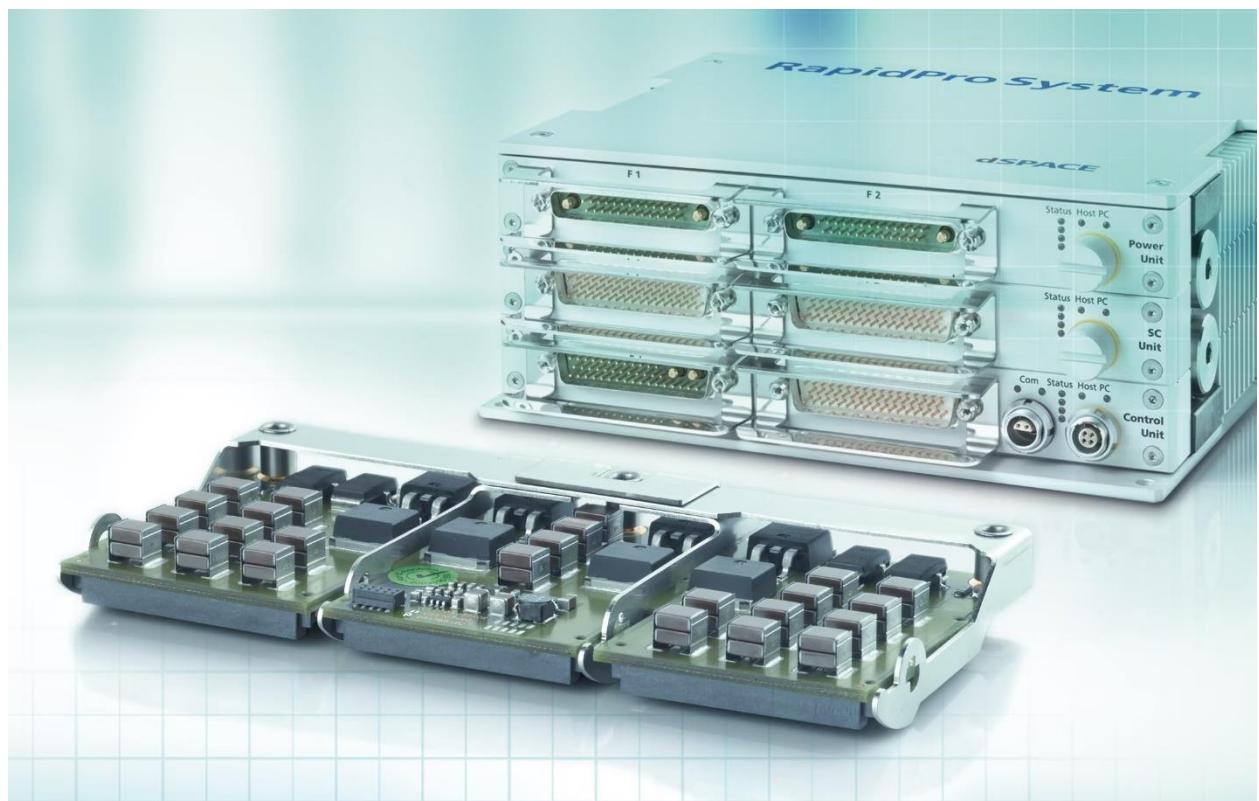
- Robotics: Các cổng giao tiếp đa dạng làm cho MicroLabBox trở thành lựa chọn lý tưởng cho nhiều ứng dụng robot.

- Một số lĩnh vực công nghiệp cần độ tin cậy và chính xác như ứng dụng về khử tiếng ồn và chống rung, hàng không vũ trụ, ...

3.3.2. RapidPro System

3.3.2.1. Giới thiệu RapidPro System

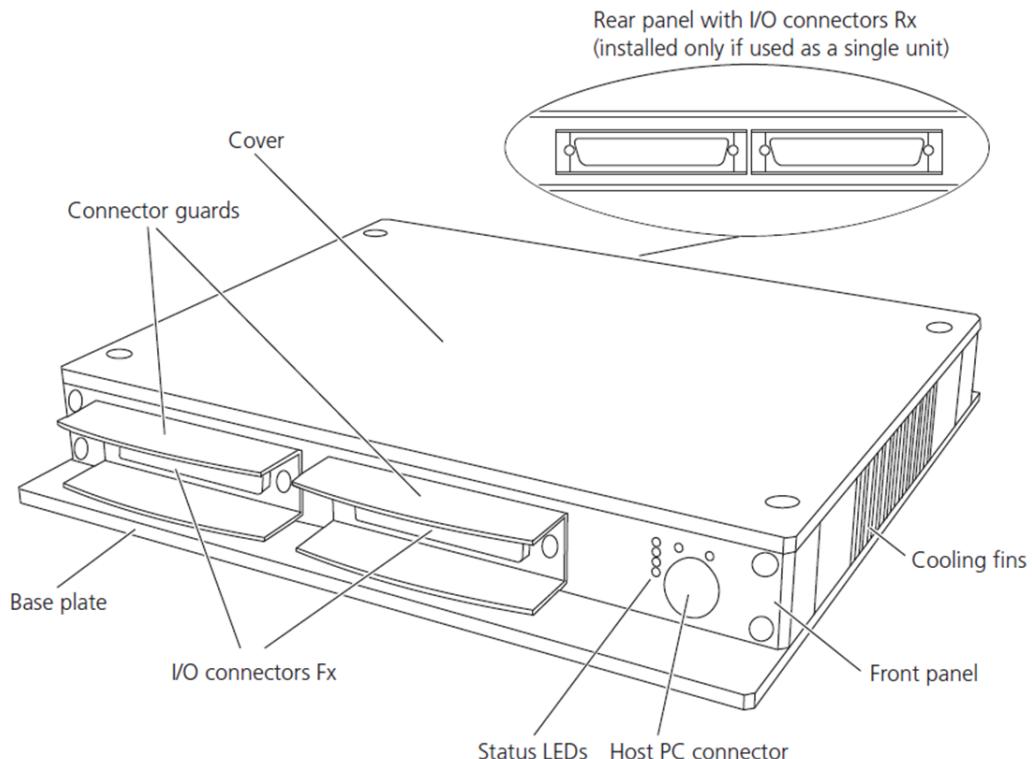
Phần cứng RapidPro hoạt động như một **phân mỏ rộng** cho các hệ thống RCP của dSPACE. Bọc ngoài của nó được thiết kế có thể sử dụng các RapidPro Unit một cách độc lập hoặc kết nối nhiều unit thành một khối thống nhất để sử dụng như một đơn vị vật lý duy nhất. Các mô-đun hiệu chỉnh tín hiệu (SC) và mô-đun nguồn (PS) có thể được cài đặt trong các đơn vị để thiết lập các hệ thống cá nhân phù hợp nhất với nhu cầu của một ứng dụng cụ thể. Các mô-đun có thể cấu hình phần cứng và phần mềm, có nghĩa là tất cả các thành phần có thể được tái sử dụng, cấu hình lại hoặc mở rộng mà không cần nhiều công sức, ví dụ như trong các dự án sau này hoặc nếu yêu cầu thay đổi.



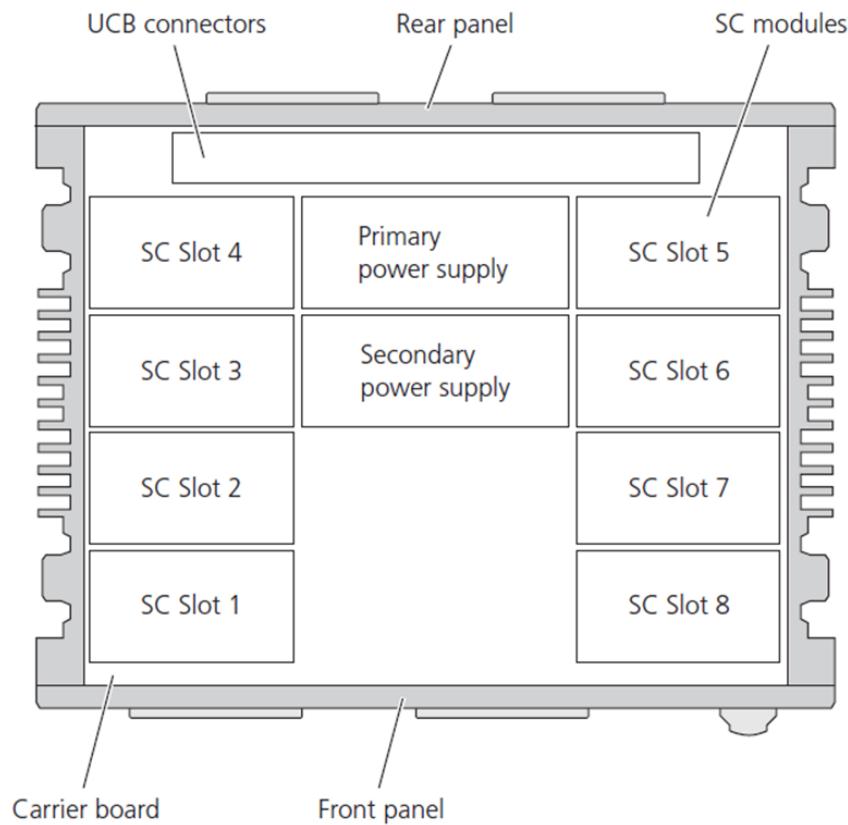
Hình 3.31. RapidPro System

Hệ thống RapidPro gồm có 3 thành phần cơ bản: RapidPro SC Unit, RapidPro Power Unit và RapidPro Control Unit. Tùy thuộc vào yêu cầu của sản phẩm mà ta có thể sử dụng riêng lẻ từng thành phần hoặc ghép các thành phần với nhau

- **RapidPro SC Unit:** chứa các module xử lí và chuyển đổi tín hiệu nhận được từ cảm biến (sensor) bao gồm các chức năng như khuếch đại, lọc nhiễu, biến đổi dạng tín hiệu,...Module này thường được gọi là Signal Conditioning Modules. RapidPro SC Unit chứa lên đến 8 module này và cấu hình bằng phần mềm ConfigurationDesk trên máy tính thông qua kết nối USB.

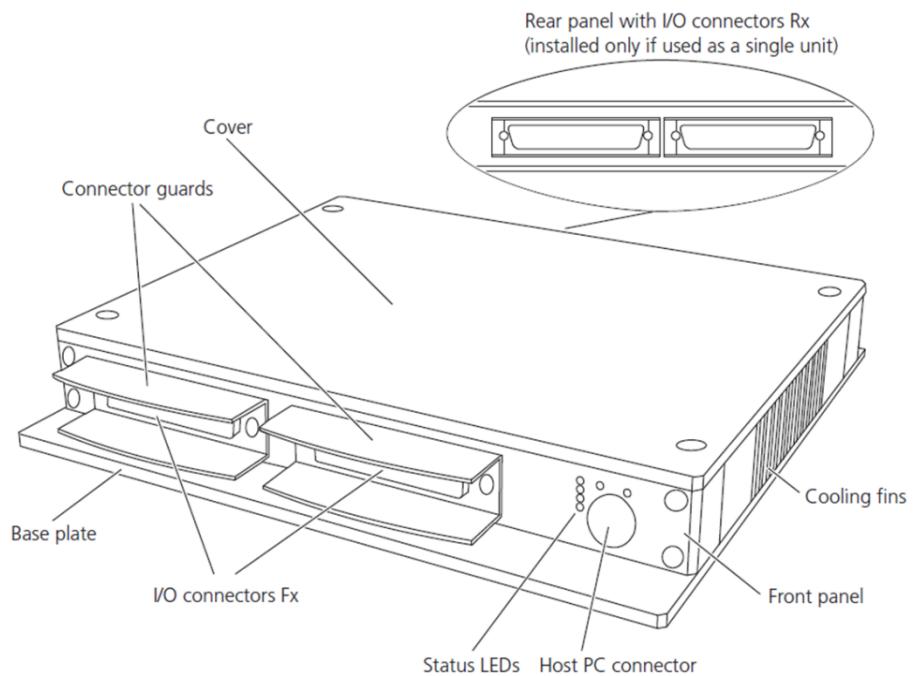


Hình 3.32. Cấu tạo bên ngoài của RapidPro SC Unit

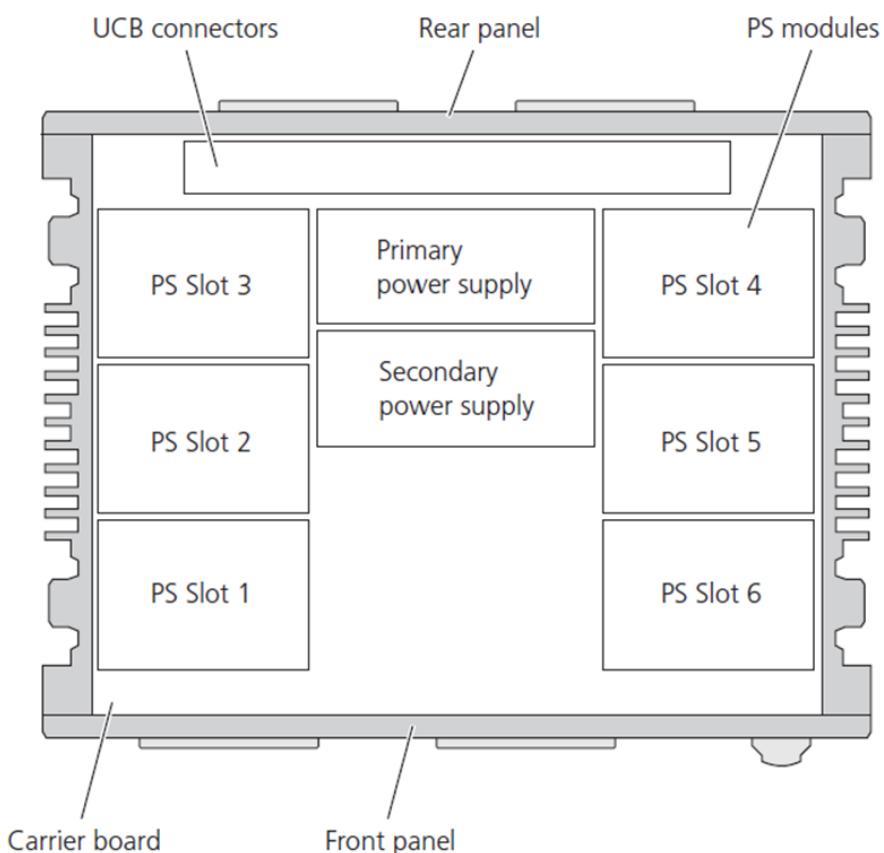


Hình 3.33. Cấu tạo bên trong của RapidPro SC Unit

- **Rapid Power Unit:** chứa các module có chức năng cung cấp nguồn và tín hiệu điều khiển cho cơ cấu chấp hành (actuator). Module này được gọi là Power Stage Modules. RapidPro Power Unit chứa 6 PS module, được cấu hình bằng phần mềm ConfigurationDesk, giao tiếp thông qua cổng USB.

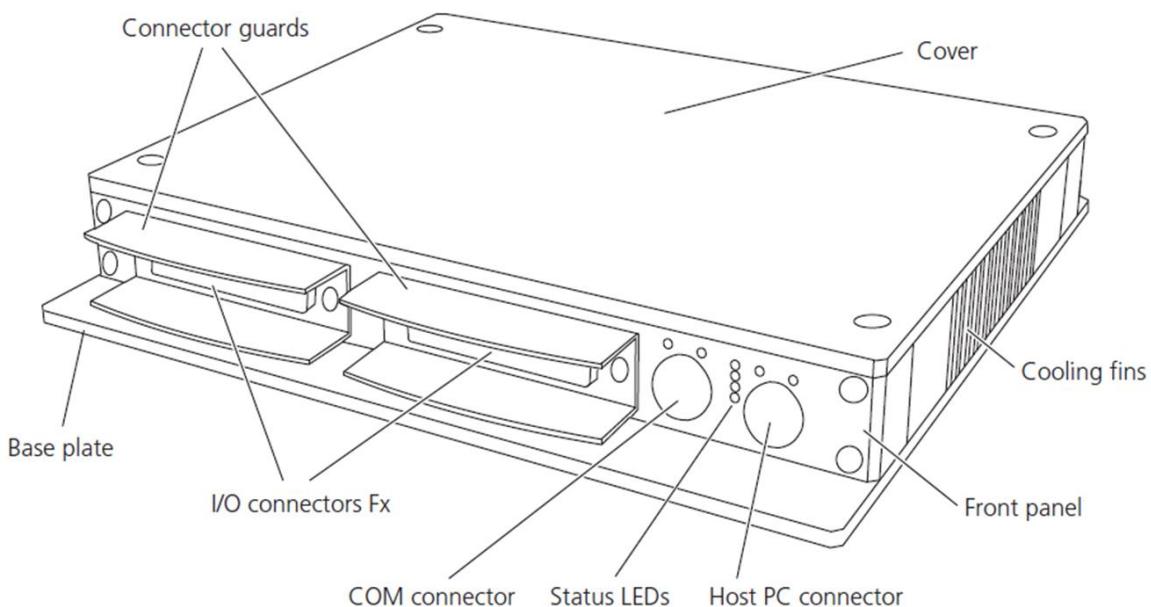


Hình 3.34. Cấu tạo bên ngoài của Power Unit

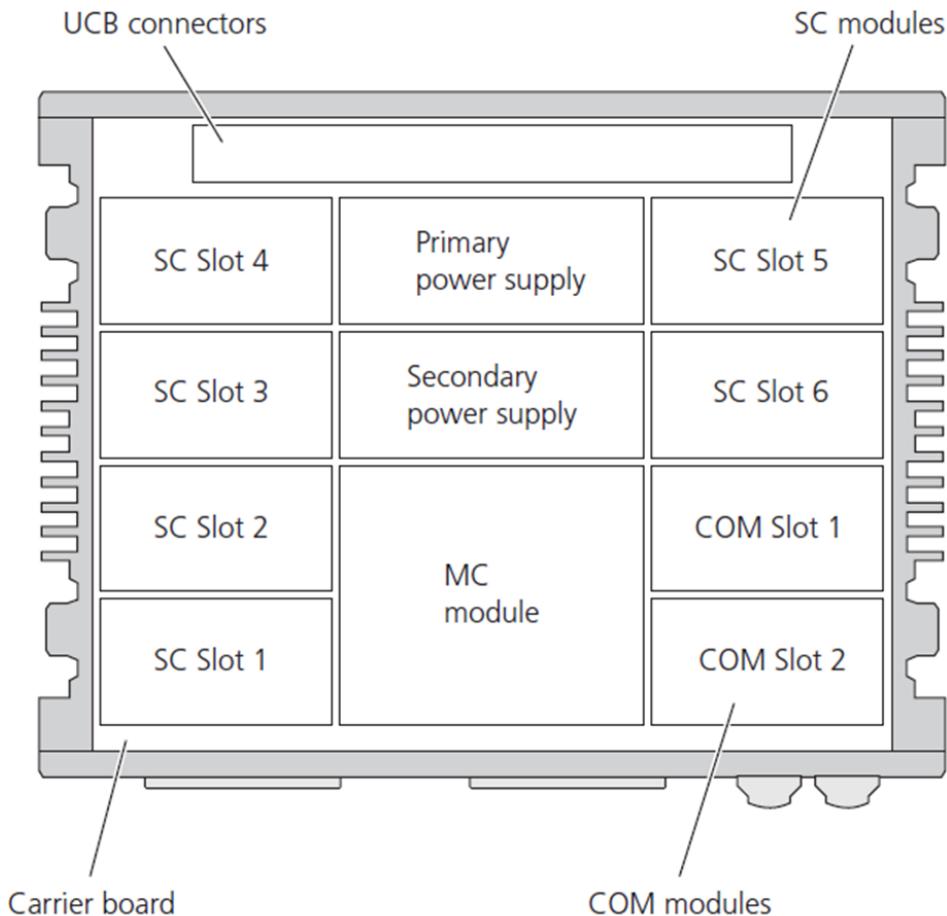


Hình 3.35. Cấu tạo bên trong của RapidPro Power Unit

- **RapidPro Control Unit:** chứa các SC modules để xử lí và chuyển đổi tín hiệu giống như RapidPro SC Unit, nhưng đồng thời có chứa vi điều khiển MPC565 và các cổng giao tiếp COM để giao tiếp với máy tính hoặc hệ thống RCP. Tùy vào ứng dụng, RapidPro Control Unit có thể đóng vai trò như một bộ vi điều khiển, hoặc chỉ có vai trò như một cổng kết nối I/O mở rộng cho hệ thống RCP. Unit này bao gồm 6 SC modules cấu hình bằng ConfigurationDesk thông qua kết nối USB, 2 COM module giao tiếp mạng LAN để truyền nhận dữ liệu đến máy tính/RCP , MPC565 module.



Hình 3.36. Cấu tạo bên ngoài của RapidPro Control Unit



Hình 3.37. Cấu tạo bên trong của RapidPro Control Unit

- **Chức năng của các cổng kết nối**

I/O Connector Fx: kết nối với nguồn (UBAT) và thiết bị bên ngoài (sensor, actuator).

Host PC Connector: giao tiếp với máy tính thông qua giao tiếp USB để cấu hình phần cứng hoặc chẩn đoán lỗi (nếu dùng unit riêng lẻ).

Status LED: ghi lại trạng thái nguồn và các trạng thái hoạt động, báo lỗi.

I/O Connector Rx: kết nối với hệ thống RCP của dSPACE (MicroLabBox, MicroAutoBox hoặc một số module khác).

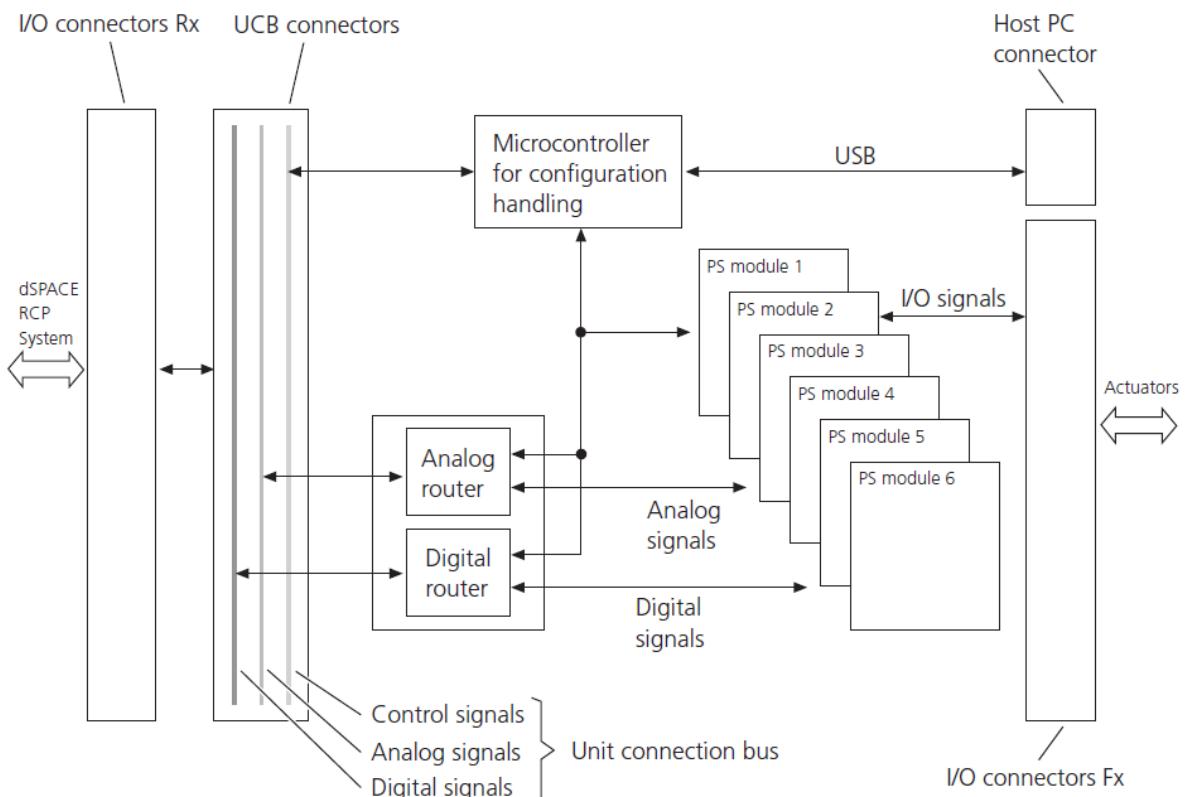
COM Connectors: để kết nối với hệ thống RCP như MicroLabBox (đối với trường hợp là RapidPro Control Unit sử dụng như một kết nối I/O mở rộng).

3.3.2.2. RapidPro Power Unit

RapidPro Power Unit là một hệ thống nhiều module có tính linh hoạt, thông minh được thiết kế cho việc tạo mẫu thuật toán điều khiển một cách nhanh chóng (rapid control prototyping) trong các ứng dụng về lĩnh vực ô tô. Nó không chỉ phù hợp sử dụng trên xe,

mà còn được ưa chuộng trong phòng thí nghiệm, hoặc đơn giản chỉ là những trường hợp thử nghiệm. Phần cứng này được thiết kế rất nhỏ gọn, chắc chắn với công nghệ kim loại tiên tiến, bên trong chứa các PS module. Tất cả đầu vào và đầu ra của nó đều có mạch bảo vệ. RapidPro Power Unit có thể được trang bị chỉ một hoặc nhiều PS module tùy thuộc vào nhu cầu sử dụng (tối đa 6 module). Toàn bộ Unit có thể được bật/tắt từ xa hoặc tắt cả đầu vào/đầu ra được kích hoạt/vô hiệu hóa từ hệ thống RCP. Giao tiếp USB có sẵn để giao tiếp với máy tính, sử dụng dSPACE ConfigurationDesk để cấu hình lại RapidPro Power Unit và chẩn đoán, sử lí lỗi.

RapidPro Power Unit, kết hợp với PS module tiêu chuẩn hoặc các module đã được điều chỉnh, cho phép đáp ứng nhiều loại cơ cấu chấp hành khác nhau (ví dụ động cơ, kim phun, van, tải cảm ứng, relay,...) cho các hệ thống tạo mẫu cụ thể.



Hình 3.38. Sơ đồ khái niệm của RapidPro Power Unit

Các bộ định tuyến tín hiệu số và tín hiệu analog được điều khiển bởi vi điều khiển tích hợp. Vi điều khiển này cần mã định tuyến (do dSPACE cung cấp) chứa thông tin các chân tương ứng trên các mô-đun được lắp đặt. Mã định tuyến này xác định mô-đun PS nào có thể được lắp vào khe cắm nào.

Single Power Unit	Module	PS1	PS2	PS3	PS4	PS5	PS6
	PS-FBD 2/1 (DS1661)	✓	✓	✓	✓	✓	✓
	PS-HSD 6/1 (DS1663)	✓	✓	✓	✓	✓	✓
	PS-LSD 6/1 (DS1662)	✓	✓	✓	✓	✓	✓
	PS-DINJ 2/1 (DS1664)	✓		—	—	✓	
	PS-HCFBD 1/1 ^{1), 2)} (DS1667)	✓				✓	
	PS-HCHBD 2/1 ^{1), 2)} (DS1668)	✓				✓	
	PS-HCFBD 1/2 ¹⁾ (DS1767)	✓				✓	
	PS-HCHBD 2/2 ^{1), 3)} (DS1768)	✓				✓	

Hình 3.39. RountingID 44205

Single Power Unit	Module	PS1	PS2	PS3	PS4	PS5	PS6
	PS-FBD 2/1 (DS1661)	—	—	—	—	—	—
	PS-HSD 6/1 (DS1663)	—	—	—	—	—	—
	PS-LSD 6/1 (DS1662)	—	—	—	—	—	—
	PS-DINJ 2/1 (DS1664)	—	—	—	—	—	—
	PS-HCFBD 1/1 ^{1), 2)} (DS1667)	✓				✓	
	PS-HCHBD 2/1 ^{1), 2)} (DS1668)	✓				✓	
	PS-HCFBD 1/2 ¹⁾ (DS1767)	✓				✓	
	PS-HCHBD 2/2 ¹⁾ (DS1768)	✓				✓	

Hình 3.40. RountingID 60766

(1): Mô-đun PS-HCxBD x/x yêu cầu ba khe liền kề. Bạn có thể gắn mô-đun vào rail 1 (các khe PS1 ... PS3) hoặc vào rail 2 (các khe PS4 ... PS6).

(2): Mô-đun PS-HCxBD x/1 chỉ sử dụng các tín hiệu I/O có sẵn của khe PS2 (nếu được gắn vào rail 1) hoặc của khe PS5 (nếu được gắn vào rail 2). Các tín hiệu I/O của khe PS1 và PS3 (hoặc PS4 và PS6) không được kết nối với mô-đun.

(3): Mã định tuyến này chỉ có giá trị cho các mô-đun PS-HCHBD 2/2 trong chế độ sử dụng General purpose, single channel hoặc General purpose, double channel.

Kết nối UCB bao gồm các tín hiệu số, tín hiệu analog và tín hiệu điều khiển. Khi RapidPro Power Unit được sử dụng đơn lẻ, các tín hiệu trên UCB sẽ có sẵn trên đầu nối

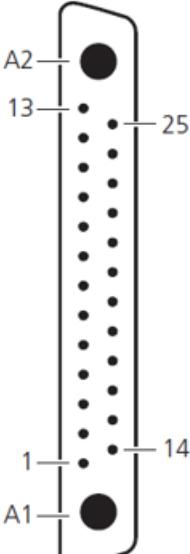
I/O Rx để giao tiếp với hệ thống RCP. Nếu RapidPro Power Unit được sử dụng stack với các RapidPro khác, kết nối UCB để giao tiếp giữa các thiết bị này.

RapidPro Power Unit có một phần điều khiển và một phần cấp nguồn để cung cấp cho các mô-đun PS. Phần cấp nguồn phải cung cấp đủ năng lượng cho các mô-đun PS để chuyển đổi tải, v.v. Phần cấp nguồn được chia thành hai đường ray riêng biệt, một đường với các khe 1 ... 3 và một đường với các khe 4 ... 6. Mỗi đường ray phải được kết nối riêng biệt với một nguồn điện áp, ví dụ như UBAT.

3.3.2.2.1. F1 Connector

Cổng kết nối F1 cung cấp quyền truy cập đến tín hiệu I/O của các PS module được lắp vào khe cắm 1 ... 3, để kết nối chúng với các thiết bị bên ngoài. Ngoài ra, nó còn có các đầu vào cho nguồn cấp điện bên ngoài của các module PS 1 ... 3.

F1 Connector	Pin	Signal	Pin	Signal
A2	25	UBAT1		
13	13	S2_IO1		
	12	S2_IO3	25	S2_IO2
	11	S2_IO5	24	S2_IO4
	10	S2_IO7	23	S2_IO6
	9	S1_IO1	22	S2_IO8
	8	S1_IO3	21	S1_IO2
	7	S1_IO5	20	S1_IO4
	6	S1_IO7	19	S1_IO6
	5	S3_IO1	18	S1_IO8
	4	S3_IO3	17	S3_IO2
	3	S3_IO5	16	S3_IO4
	2	S3_IO7	15	S3_IO6
	1	KL15	14	S3_IO8
A1		GND1		



Hình 3.41. Kết nối Sub-D của F1 Connector

Trong đó: Sx_IOy - chân tín hiệu đầu ra(x là thứ tự khe cắm, y là thứ tự kênh I/O). KL15 - tín hiệu đầu vào bật/tắt từ xa. UBAT1, GND1 - cấp nguồn cho module từ 1 đến 3.

Bảng 3.4. Bảng tín hiệu của F1 Connector

Tín hiệu	Mô tả	Thông số kỹ thuật	
Sx_IOy	Tín hiệu đầu ra	Điện áp đầu ra	Tùy vào PS module
		Dòng điện đầu ra	Tùy vào PS module và tải.
KL15	Tín hiệu điều khiển từ xa để bật RapidPro Power Unit (ví dụ tín hiệu đánh lừa,...)	Điện áp ON	> 4V
		Điện áp OFF	< 3V
		Bảo vệ quá áp	+/-100V
UBAT1	Đầu vào nguồn cung cấp bên ngoài (ví dụ, từ pin ô tô) cho các khe mô-đun 1 ... 3 và tất cả các điện áp cung cấp nội bộ.	Điện áp đầu vào	6V đến 54V
		Dòng điện dự phòng	<10mA (8mA tối ưu)
		Năng lượng tiêu thụ	Xấp xỉ 10W tùy vào PS module
		Bảo vệ quá áp	+/-100V
GND1	Dây nối đất nguồn cung cấp cho các khe mô-đun 1 ... 3 và tất cả các điện áp cung cấp nội bộ.		

3.3.2.2.2. F2 Connector

Cổng kết nối F1 cung cấp quyền truy cập đến tín hiệu I/O của các PS module được lắp vào khe cắm 4 ... 6, để kết nối chúng với các thiết bị bên ngoài. Ngoài ra, nó còn có các đầu vào cho nguồn cung cấp điện bên ngoài của các module PS 4 ... 6.

F2 Connector	Pin	Signal	Pin	Signal
A2		UBAT2		
13		S6_IO1	25	S6_IO2
		S6_IO3	24	S6_IO4
11		S6_IO5	23	S6_IO6
10		S6_IO7	22	S6_IO8
9		S5_IO1	21	S5_IO2
8		S5_IO3	20	S5_IO4
7		S5_IO5	19	S5_IO6
6		S5_IO7	18	S5_IO8
5		S4_IO1	17	S4_IO2
4		S4_IO3	16	S4_IO4
3		S4_IO5	15	S4_IO6
2		S4_IO7	14	S4_IO8
1		Reserved		
A1		GND2		

Hình 3.42. Kết nối Sub-D của F2 Connector

Trong đó: Sx_IOy - chân tín hiệu đầu vào/đầu ra (x là thứ tự khe cắm, y là thứ tự kênh I/O). UBAT1,GND1 - cấp nguồn cho module từ 4 đến 6.

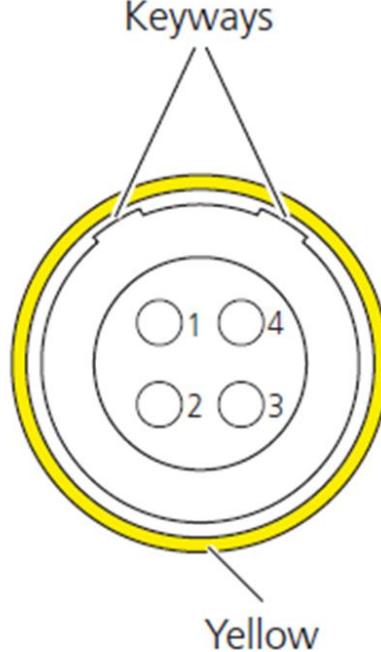
Bảng 3.5. Bảng tín hiệu của F2 Connector

Tín hiệu	Mô tả	Thông số kỹ thuật	
Sx_IOy	Tín hiệu đầu ra	Điện áp đầu ra	Tùy vào PS module
		Dòng điện đầu ra	Tùy vào PS module và tải.
UBAT2	Đầu vào nguồn cung cấp bên ngoài (ví dụ, từ pin ô tô) cho các khe mô-đun 4 ... 6 và tất cả các điện áp cung cấp nội bộ.	Điện áp đầu vào	6V đến 54V
		Dòng điện dự phòng	<10mA (8mA tối ưu)
		Năng lượng tiêu thụ	Xấp xỉ 10W tùy vào PS module
		Bảo vệ quá áp	+/-100V

GND2	Dây nối đất nguồn cung cấp cho các khe mô-đun 4 ... 6 và tất cả các điện áp cung cấp nội bộ.	
------	--	--

3.3.2.2.3. Host PC Connector

Cung cấp khả năng truy cập vào giao diện USB của RapidPro Power Unit, để cấu hình module từ máy tính.

Host PC Connector	Pin	Signal
	1	VBUS
	2	D+
	3	D-
	4	GND

Hình 3.43. Kết nối USB đến máy tính

Trong đó: VBUS,GND - cung cấp nguồn cho kết nối USB. D+, D- - chân tín hiệu của USB.

3.3.2.2.4. R1 Connector

Được sử dụng để truyền/nhận tín hiệu số (digital) giữa RapidPro Power Unit và hệ thống RCP (ví dụ như MicroLabBox, MicroAutoBox,...).

R1 Connector	Pin	Signal	Pin	Signal	Pin	Signal
18	1	DIG_1	18	DIG_18	34	DIG_34
1	2	DIG_2	19	DIG_19	35	DIG_35
17	3	DIG_3	20	DIG_20	36	DIG_36
33	4	DIG_4	21	DIG_21	37	DIG_37
	5	DIG_5	22	DIG_22	38	DIG_38
	6	DIG_6	23	DIG_23	39	DIG_39
	7	DIG_7	24	DIG_24	40	DIG_40
	8	DIG_8	25	DIG_25	41	DIG_GND
	9	DIG_9	26	DIG_26	42	DIG_GND
	10	DIG_10	27	DIG_27	43	DIG_GND
	11	DIG_11	28	DIG_28	44	DIG_GND
	12	DIG_12	29	DIG_29	45	DIG_GND
	13	DIG_13	30	DIG_30	46	DIG_GND
	14	DIG_14	31	DIG_31	47	SPI_MOSI
	15	DIG_15	32	DIG_32	48	SPI_MISO
	16	DIG_16	33	DIG_33	49	SPI_CLK
	17	DIG_17			50	SPI_CS

Hình 3.44. Kết nối Sub-D của R1 Connector

Bảng 3.6. Bảng tín hiệu của R1 Connetor

Tín hiệu	Mô tả	Thông số kĩ thuật	
DIG_x	Tín hiệu số đầu vào/ đầu ra	Điện áp đầu vào	0V đến 5V
		Điện áp đầu ra	0V đến 3.3V
DIG_GND	Đường trả về tín hiệu số	-	
SPI_MISO	Đầu ra dữ liệu của giao tiếp SPI	-	
SPI_MOSI	Đầu vào dữ liệu của giao tiếp SPI	-	
SPI_CLK	Chân xung nhịp của giao tiếp SPI	-	
SPI_CS	Chân chọn chip của giao tiếp SPI	-	

3.3.2.2.5. R2 Connector

Được sử dụng để truyền/nhận tín hiệu analog giữa RapidPro Power Unit và hệ thống RCP (ví dụ như MicroLabBox, MicroAutoBox, ...).

R2 Connector	Pin	Signal	Pin	Signal	Pin	Signal
	1	ANA_1	18	ANA_18	34	ANA_34
	2	ANA_2	19	ANA_19	35	ANA_35
	3	ANA_3	20	ANA_20	36	ANA_36
	4	ANA_4	21	ANA_21	37	ANA_37
	5	ANA_5	22	ANA_22	38	ANA_38
	6	ANA_6	23	ANA_23	39	ANA_39
	7	ANA_7	24	ANA_24	40	ANA_40
	8	ANA_8	25	ANA_25	42	ANA_GND
	9	ANA_9	26	ANA_26	43	ANA_GND
	10	ANA_10	27	ANA_27	44	ANA_GND
	11	ANA_11	28	ANA_28	45	ANA_GND
	12	ANA_12	29	ANA_29	46	ANA_GND
	13	ANA_13	30	ANA_30	47	ANA_GND
	14	ANA_14	31	ANA_31	48	WDT
	15	ANA_15	32	ANA_32	49	KL15
	16	ANA_16	33	ANA_33	50	ENABLE

Hình 3.45. Kết nối Sub-D của R2 Connector

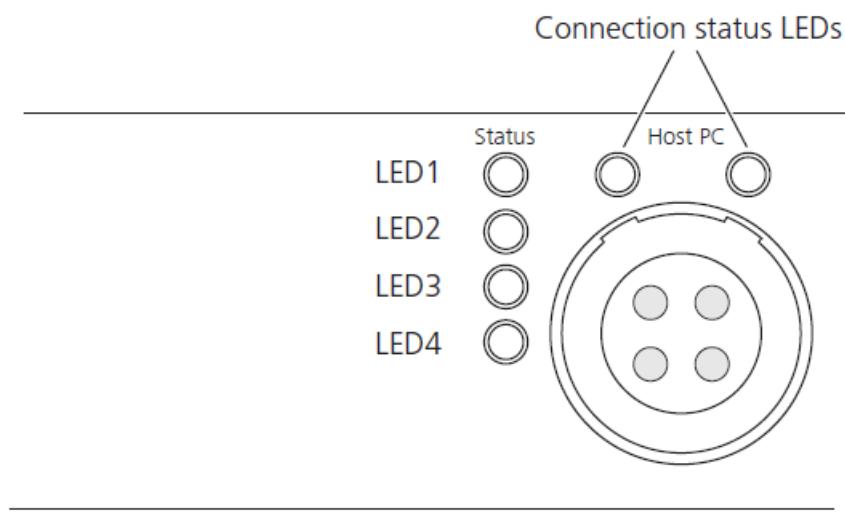
Bảng 3.7. Bảng tín hiệu của R2 Connector

Tín hiệu	Mô tả	Thông số kỹ thuật	
ANA_x	Tín hiệu analog đầu vào/đầu ra	Điện áp đầu vào/đầu ra	-10V đến 10V
ANA_GND	Đường trả về của tín hiệu analog	-	
WDT	Watchdog timer: Nếu không có tín hiệu, điều khiển đầu ra của tắt	Điện áp đầu vào	0V đến 5V

	cả các mô-đun đã cài đặt sẽ ở chế độ tắt.	Loại tín hiệu	Xung vuông
		Tần số tín hiệu	10Hz đến 100kHz
ENABLE	Nếu tín hiệu này ở mức thấp, tắt cả tín hiệu đầu ra của module sẽ tắt	Điện áp đầu vào	0V đến 5V DC ON: 2.5V OFF: 1.5V
KL15	Tín hiệu điều khiển từ xa để bật RapidPro Power Unit (ví dụ tín hiệu đánh lửa,...)	Điện áp ON Điện áp OFF	>4V <3V
		Bảo vệ quá áp	+50V

3.3.2.2.6. Status LEDs

RapidPro Power Unit được trang bị với 6 đèn LED chỉ trạng thái hoạt động. Trong đó, 4 LED chỉ trạng thái lỗi đầu vào/đầu ra của tín hiệu, 2 LED chỉ trạng thái kết nối giao tiếp.



Hình 3.46. Vị trí LED trên RapidPro Power Unit

LED bên trái: sáng xanh lá nếu kết nối được thiết lập và giao diện sẵn sàng cho việc giao tiếp.

LED bên phải: chớp xanh lá nếu dữ liệu được gửi hoặc được nhận.

LED1: chỉ trạng thái nguồn cấp bên ngoài. LED sáng màu xanh lá chỉ hành vi đúng, màu cam hoặc đỏ chỉ hành vi sai.

LED2...LED4: hiển thị các lỗi của hệ thống RapidPro hoặc lỗi ứng dụng tại các thiết bị được kết nối với hệ thống RapidPro.. LED sáng màu xanh lá chỉ hành vi đúng, màu cam hoặc đỏ chỉ hành vi sai.

3.3.2.2.7. PS Module

Hiện nay, trên RapidPro Power Unit có sẵn 8 module với đa dạng các chức năng phù hợp cho nhiều ứng dụng khác nhau:

Bảng 3.8. Module có sẵn trên RapidPro Power Unit

PS Module	Mô tả	Order Number
PS-FBD 2/1	Là một module công suất full-bridge với 2 kênh để điều khiển tải hai chiều như động cơ DC. Mỗi kênh cung cấp hai đầu vào điều khiển và có thể điều khiển một tải được kết nối tại hai đầu ra.	DS1661
PS-LSD 6/1	Là một module công suất low-side, cung cấp 6 kênh để điều khiển các tải được kết nối với UBAT, như van, rơ-le, cuộn đánh lửa hoặc thiết bị làm nóng.	DS1662
PS-HSD 6/1	Là một module công suất high-side, cung cấp 6 kênh để chuyển tải đến UBAT mà đầu còn lại được kết nối với GND. Module này có thể điều khiển các tải như van, rơ-le, cuộn đánh lửa, thiết bị làm nóng, và đèn.	DS1663
PS-HCFBD 1/1	Là một module công suất full-bridge dòng cao với một kênh để điều khiển tải hai chiều như động cơ DC (ví dụ, trong các hệ thống cung cấp điện 12 V). Module này cung cấp hai đầu vào điều khiển và có thể điều khiển một tải được kết nối tại hai đầu ra.	DS1667
PS-HCHBD 2/1	Là một module công suất half-bridge dòng cao với hai kênh để điều khiển các tải ở các chế độ khác nhau. Mỗi kênh cung cấp một đầu vào điều khiển và có thể điều khiển một tải được kết nối tại đầu ra ở các chế độ: half-bridge, low-side, high-side.	DS1668
PS-DINJ 2/1	Có hai kênh để điều khiển van phun điện từ.	DS1664

PS-HCHFBD 1/2	Là một module công suất full-bridge dòng cao với một kênh để điều khiển tải hai chiều như động cơ DC (12 V hoặc 24 V). Module này cung cấp hai đầu vào điều khiển và có thể điều khiển một tải được kết nối tại hai đầu ra.	DS1767
PS-HCHBD 2/2	Là một module công suất half-bridge dòng cao với hai kênh để điều khiển các tải ở các chế độ khác nhau trong các hệ thống cung cấp điện 12 V hoặc 24 V.	DS1768

3.3.2.3. PS-HCHBD 2/2 Module

Mô-đun PS-HCHBD 2/2 (DS1768) là mô-đun trạng thái nguồn h-bridge dòng cao với hai kênh để điều khiển tải ở các chế độ khác nhau trong hệ thống cung cấp điện 12 V hoặc 24 V.

Các tính năng chính của mô-đun PS-HCHBD 2/2 như:

- Hai kênh điều khiển tải.
- Các chế độ vận hành đáp ứng đầu ra của đa dạng các trường hợp khác nhau như: h-bridge, low-side, high-side và đặc biệt điều khiển động cơ điện 3 pha (BLDC/BLAC).
- Dòng điện đầu ra: tối đa 60A cho một giây.
- Hạn chế dòng điện của bộ chopper cho dòng điện đầu ra cho mỗi kênh (cấu hình băng phần mềm).
- Thiết bị đo dòng với bộ lọc low-pass filter của mỗi kênh (cấu hình băng phần mềm).
- Đo điện áp để giám sát UBAT và điện áp đầu ra.
- Chức năng chẩn đoán.
- Bảo vệ chống lại quá áp và quá nhiệt.
- Điện áp vận hành: 6V đến 30V.

Các chức năng I/O của mô-đun phụ thuộc vào sự kết hợp của các chế độ hoạt động có thể cấu hình bằng phần mềm và các chế độ điều khiển đầu ra. Các chế độ hoạt động có sẵn bao gồm:

- **General purpose, single channel**

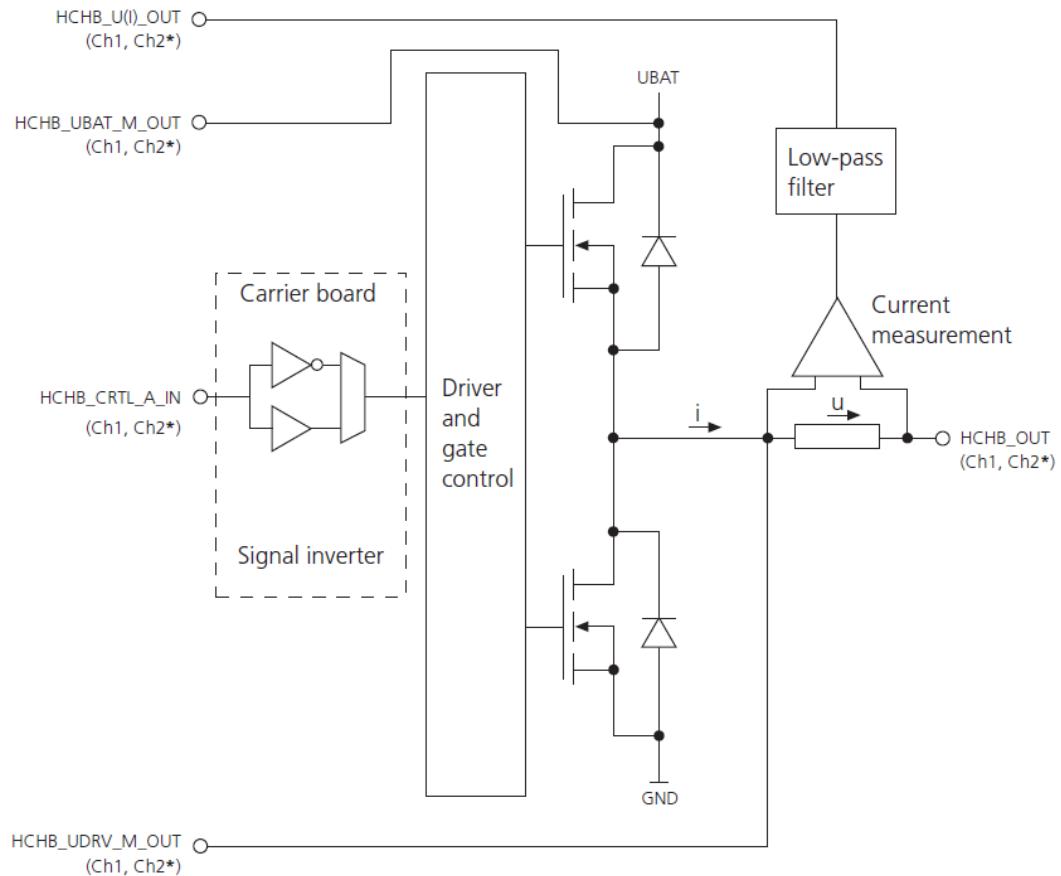
Chỉ có kênh 1 của mô-đun có sẵn để sử dụng. Kênh 2 sẽ bị khóa để cung cấp dòng điện đầu ra cho kênh 1 tối đa lên đến 60A. Có thể cấu hình các chế độ điều khiển đầu ra khác nhau ở chế độ vận hành này.

Bảng 3.9. Các chế độ điều khiển đầu ra

Chế độ điều khiển đầu ra	Mô tả
Half – bridge	Tín hiệu đầu vào chuyển đổi tải kết nối từ UBAT sang GND và ngược lại.
Low – side	Điều khiển tải được kết nối với UBAT
High - side	Điều khiển tải được kết nối với GND

- **General purpose, double channel**

Cả 2 kênh điều hoạt động, có thể cấu hình riêng từng chân bằng phần mềm. Mức dòng điện đầu ra tối đa ở mỗi kênh là 30A. Có thể cấu hình các chế độ điều khiển đầu ra khác nhau ở chế độ vận hành này.

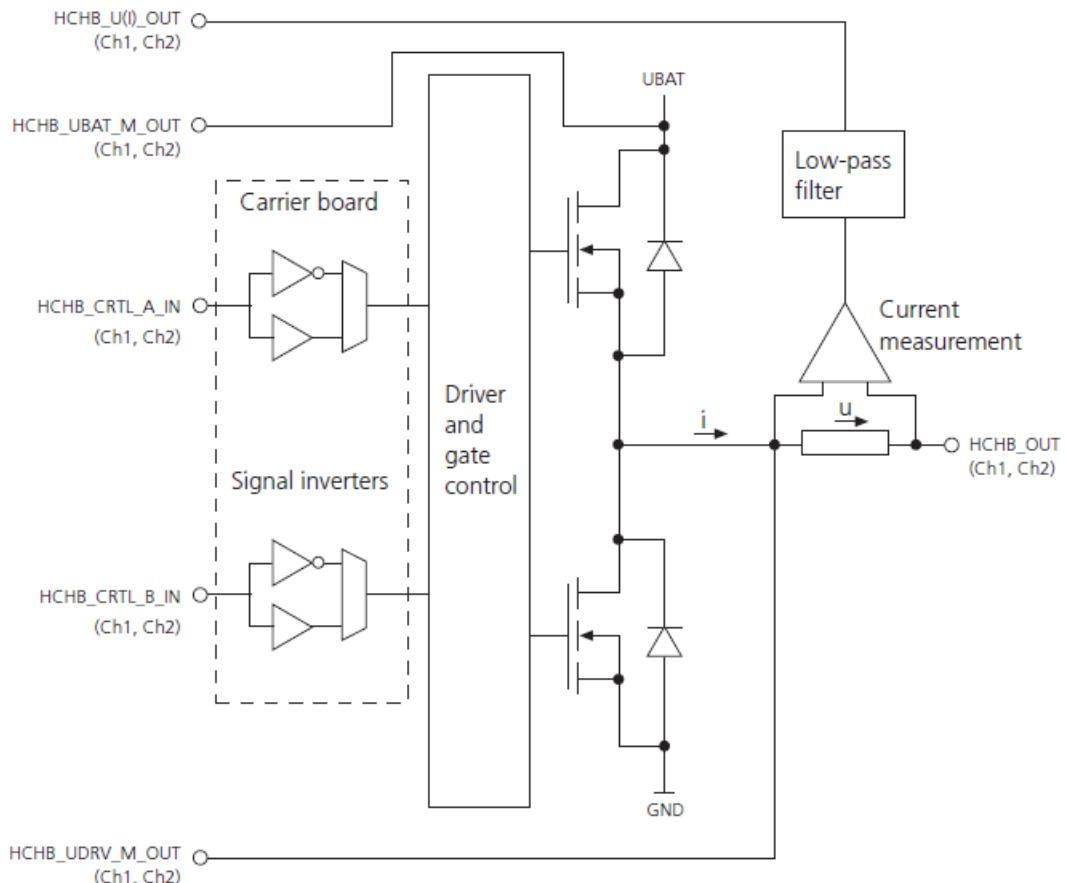


* Channel 2 is deactivated in **General purpose, single channel** mode.

Hình 3.47. Sơ đồ mạch chế độ General purpose, single channel và double channel

- **Electric drive, two half bridges**

Trong chế độ này, có thể chuyển đổi tải kết nối sang UBAT, GND hoặc trạng thái trở kháng cao. Tổng dòng điện cho hai kênh của một mô-đun này không được vượt quá 60A.

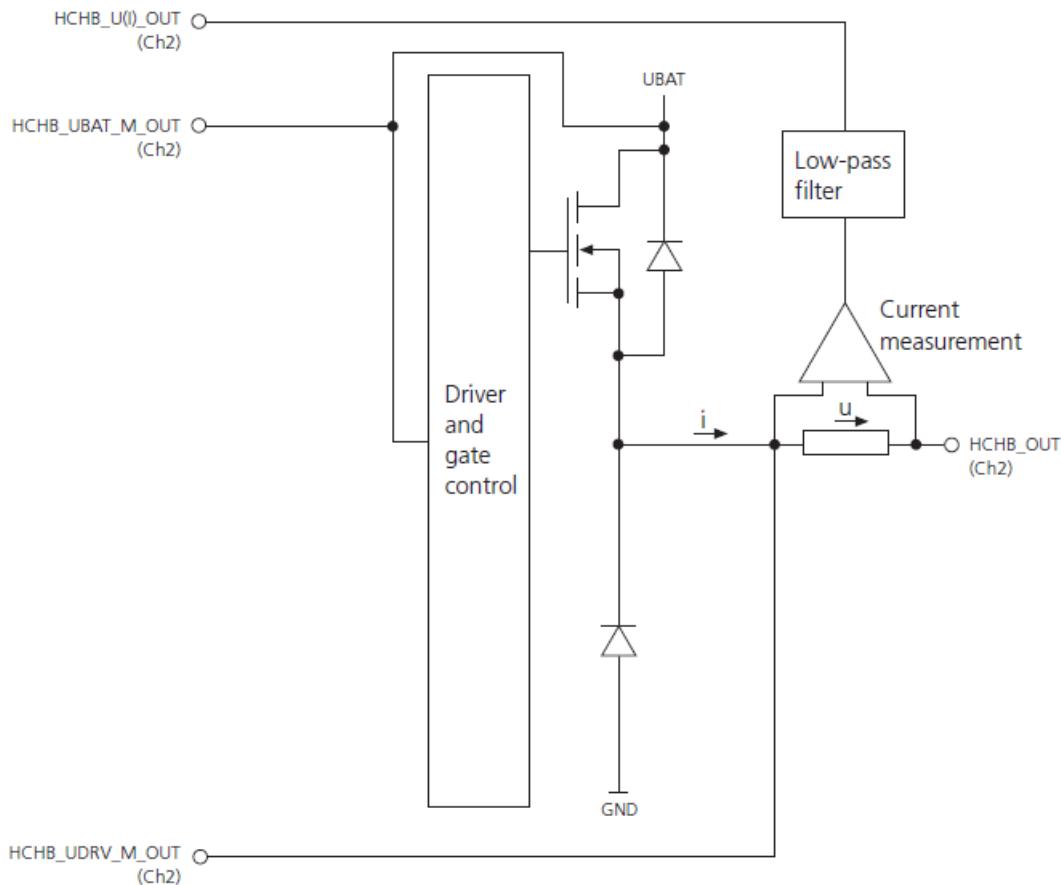


Hình 3.48. Sơ đồ mạch chế độ eletric drive, two half-bridges

- **Combination: Electric drive, two half-bridges and electric drive, half bridge and brake chopper**

Kết hợp cả hai chế độ hoạt động được sử dụng để điều khiển các động cơ điện ba pha (BLDC/BLAC), bao gồm một bộ cắt phanh tùy chọn. Trong trường hợp này, cần hai mô-đun PS-HCHBD 2/2 trong cùng một RapidPro Power Unit. Một mô-đun sẽ được cài ở chế độ eletric drive, two half-bridges, và mô-đun còn lại được cài ở chế độ eletric drive, half-bridge and brake chopper. Dòng điện đầu ra tối đa của mỗi kênh là 60A.

Mạch I/O của kênh 1 trong chế độ eletric drive, half-bridge and brake chopper giống với mạch I/O của chế độ eletric drive, two half-bridges. Mạch I/O của kênh 2 trong chế độ này có thể được sử dụng như một bộ cắt phanh để kiểm soát điện áp trên UBAT nếu động cơ hoạt động như một máy phát điện. Sơ đồ mạch của kênh 2 chế độ này như sau:



Hình 3.49. Sơ đồ mạch kênh 2 của chế độ eletic drive, half-bridge and brake chopper

Bảng 3.10. Thông số kỹ thuật module [1]

Parameter	Range/Setting	Factory Default Setting	Description	Valid
Điện áp đầu ra	0...+5V -5V ... +5V	0...+5V	Single Unit: Cho phép điều chỉnh các đầu ra analog để phù hợp với các loại bảng I/O dSPACE khác nhau.	Mọi kênh
			Stack: điện áp đầu ra được cố định ở mức 0...+5V.	

Giới hạn dòng điện	General purpose, double channel 5.0A...30.0A	30A	Thiết lập dòng điện đầu ra tối đa có thể để điều chỉnh phù hợp với các thông số kỹ thuật của tải được kết nối.	Mỗi kênh
	General purpose, single channel 5.0A...60.0A	60A		
	Electric drive, two half bridges 5.0A...60.0A	30A		
	Combined mode 5.0A...60.0A	30A		
Điện áp ngưỡng kích hoạt chopper	0.0V...30.0V	30V	Nếu điện áp đầu vào vượt quá ngưỡng, bộ cắt phanh bật	Chỉ điều chỉnh kênh 2
Điện áp ngưỡng vô hiệu hóa chopper	0.0V...30.0V	28V	Nếu điện áp đầu vào thấp hơn ngưỡng, bộ cắt phanh tắt	với chế độ kết hợp. Giá trị chênh lệch không thấp hơn 2V
Phân cực tín hiệu	Non-inverted hoặc inverted	Non-inverted	Có thể thiết lập phân cực tín hiệu của chân HCHB_CTRL_A_IN và HCHB_CTRL_B_IN	Mỗi kênh
Kích hoạt low-pass filter	True False	True	Cả phần cứng và phần mềm của bộ lọc hoạt động giúp lọc tín	Mỗi kênh

			hiệu hiệu quả hơn, nhưng không thể lọc các tín hiệu có tần số cao. Cần vô hiệu hóa lọc phần mềm để lọc tín hiệu tần số cao.	
Tần số low-pass filter	100Hz... 150000Hz	6500Hz	Có thể thiết lập tần số cắt của low-pass filter, là một phần của nhánh đo dòng điện của môđun.	Mỗi kênh
Thứ tự low-pass filter	1 st order 2 nd order	1 st order	Có thể thiết lập thứ tự của low-pass filter, là một phần của nhánh đo dòng điện của môđun.	Mỗi kênh

Bảng 3.11. Thông số kỹ thuật module [2]

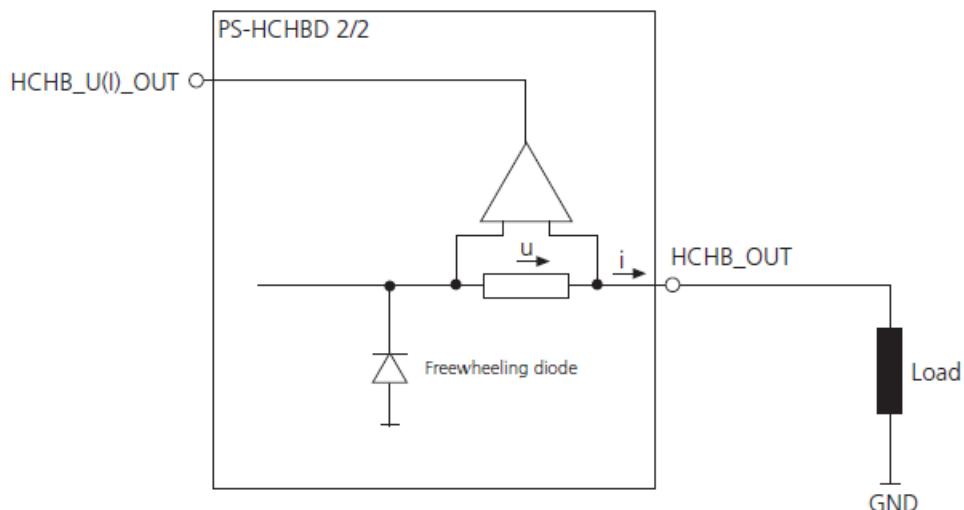
Chế độ vận hành	Chế độ điều khiển đầu ra	HCHB_CTRL _A_IN	HCHB_CTL _B_IN	HCHB _OUT	Chiều dòng điện
General purpose, single channel General purpose, double channel	Half-bridge	0		Low	+
		1		High	-
	Low-side	1		Tristate	Không dòng điện
		0		Low	+
	High-side	0		Tristate	Không dòng điện
		1		High	-
Electric drive, two half bridges		0	0	Tristate	Không dòng điện
		1	0	High	-

Eletric drive, half-bridge and brake chopper		0	1	Low	+
		1	1	Tristate	Không dòng điện

Các trường hợp kết nối với tải:

- Điều khiển tải được kết nối với GND

Nếu muốn điều khiển các tải được kết nối với GND, như van, rơ-le, các tải liên quan đến nhiệt, v.v., có thể sử dụng mô-đun PS-HCHBD 2/2 như một bộ điều khiển high-side. Để làm điều này, kết nối tải với các chân HCHB_OUT và GND.

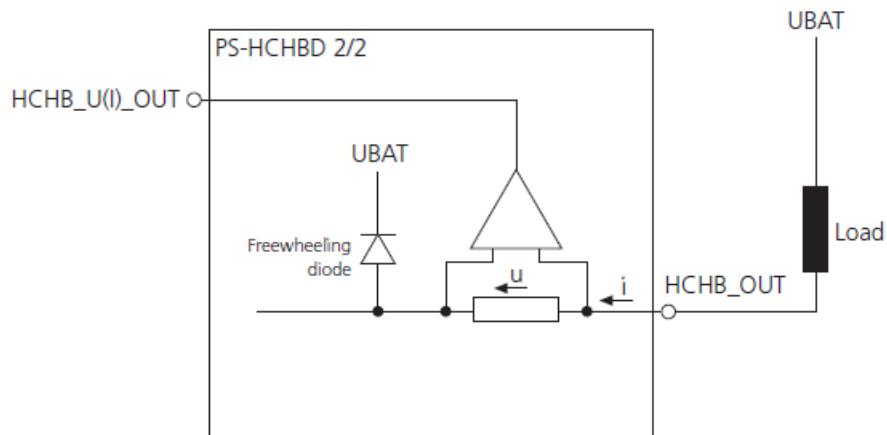


Hình 3.50. Điều khiển high-side

Bộ điều khiển high-side có sẵn trong các chế độ General purpose, single channel và General purpose, double channel. Đầu ra HCHB_OUT có sẵn trong F1 hoặc F2 Connectors tùy vào ray được chọn cho mô-đun.

- Điều khiển tải được kết nối với UBAT

Nếu muốn điều khiển các tải được kết nối với UBAT, như van, rơ-le, các tải liên quan đến nhiệt, v.v., có thể sử dụng mô-đun PS-HCHBD 2/2 như một bộ điều khiển low-side. Để làm điều này, kết nối tải với các chân HCHB_OUT và UBAT.

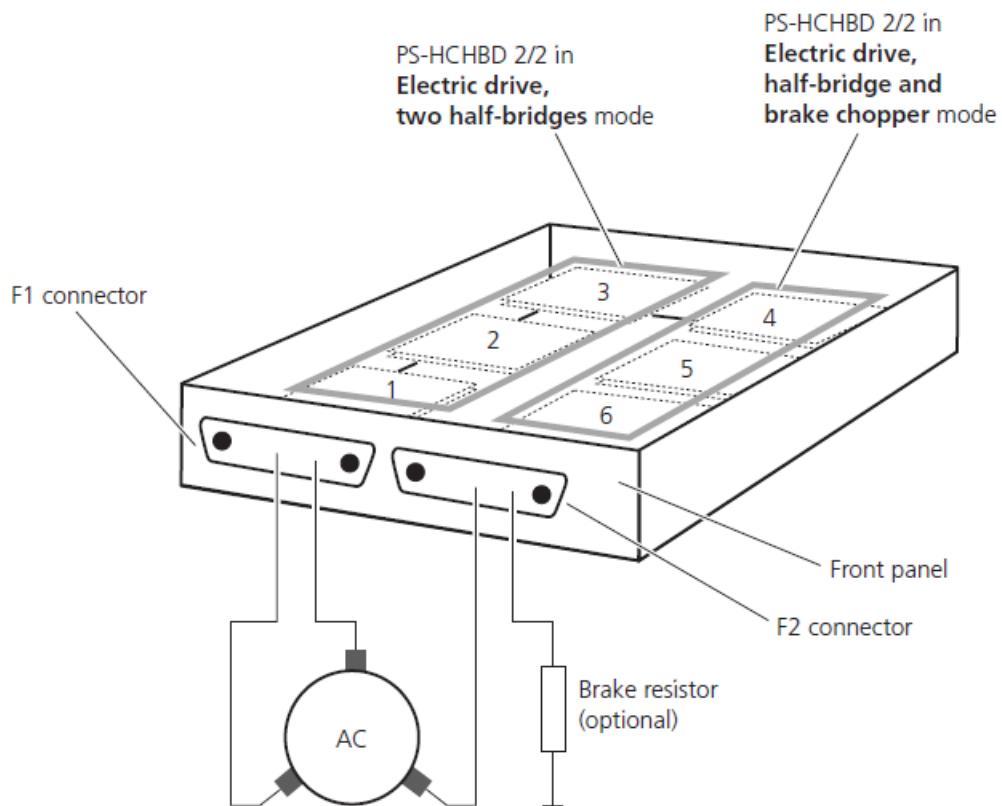


Hình 3.51. Mạch điều khiển low-side

Bộ điều khiển low-side có sẵn trong các chế độ General purpose, single channel và General purpose, double channel. Đầu ra HCHB_OUT có sẵn trong F1 hoặc F2 Connectors tùy vào ray được chọn cho mô-đun.

- Điều khiển động cơ BLDC/BLAC

Với trường hợp này, cần phải thiết lập 2 mô-đun PS-HCHBD 2/2 để có thể điều khiển được động cơ BLDC/BLAC. Cả 2 mô-đun được thiết lập ở cùng một RapidPro Power Unit. Một mô-đun sẽ được cài ở chế độ eletric drive, two half-bridges, và mô-đun còn lại được cài ở chế độ eletric drive, half-bridge and brake chopper. Ba kênh của hai mô-đun dùng để điều khiển ba pha của động cơ. Kênh 2 của ray 4...6 không dùng đến.

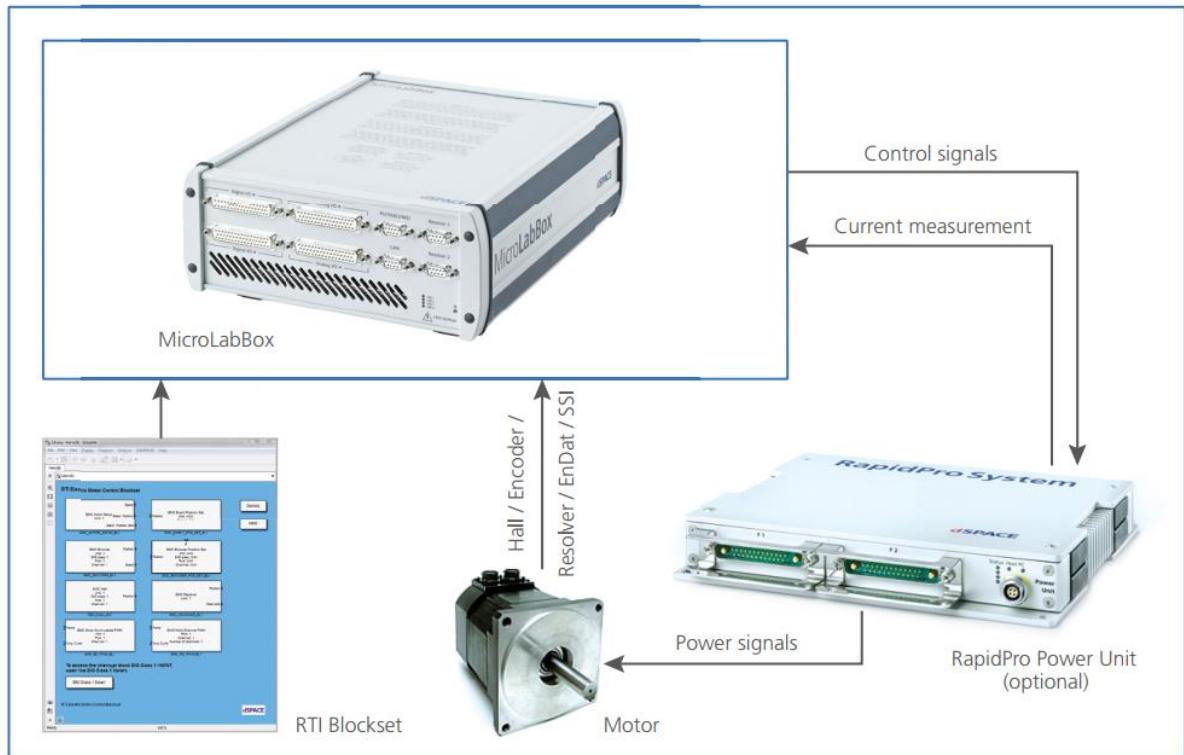


Hình 3.52. Điều khiển động cơ BLDC/BLAC

3.4. Mô hình động cơ PMSM ACMC của dSPACE

3.4.1. Tổng quan về mô hình PMSM ACMC

Mô hình PMSM ACMC là mô hình điều khiển động cơ PMSM sử dụng **MicroLabBox**, **RapidPro Power Unit** và sử dụng giao diện được xây dựng trên ứng dụng **ControlDesk** để điều khiển.



Hình 3.53. MicroLabBox trong việc điều khiển động cơ

Mô hình này cung cấp một môi trường thí nghiệm linh hoạt cho việc nghiên cứu và phát triển các ứng dụng điều khiển động cơ với việc cho phép liên tục điều khiển theo nhiều chế độ khác nhau như điều khiển bằng chiều, vận tốc hoặc điện áp. Cùng với đó, việc kết hợp với các phần cứng và phần mềm của dSPACE và xây dựng mô hình bộ điều khiển trên Matlab/Simulink giúp cho việc điều khiển trở nên chính xác, đáng tin cậy từ đó có thể nghiên cứu, phát triển các ứng dụng liên quan đến điều khiển động cơ PMSM.

Các thiết bị phần cứng sử dụng:

- MicroLabBox
- RapidPro Power Unit
- PMSM có cảm biến Encoder/Hall

- Power Supply (220V và 12V)



Hình 3.54. Tôđng quan mô hình PMSM ACMC

Các phần mềm sử dụng:

- Matlab/Simulink
- ControlDesk
- ConfigurationDesk for RapidPro

Thông số PMSM sử dụng:



Hình 3.55. Thông số hoạt động của PMSM

- Hãng sản xuất: LS Electric

- Mã motor: APM-SB03ADK-9
- Điện áp/Dòng điện đầu vào: 3~, 30V, 9.62A
- Công suất tối đa: 300W
- Vận tốc tối đa: 3000rpm
- Encoder: Inc. 2500p/r

Bảng 3.12. Thông số kỹ thuật chi tiết động cơ APM-SB03ADK-9

	APM-SB03ADK-9	Unit
Rated Power	300	W
Pole Number	8	Pole
Rated Torque	0.95	N.m
	9.74	Kgf.cm
Rated Speed	3000	rpm
Voltage Constant	9.7	V/Krpm
Inertia	0.328	Gfcms ²
	0.321	Kgm ² .10 ⁻⁴
Torque Constant	1.574	Kgfcm/A
Phase-to-Phase Inductance	0.742	mH
Phase-to-Phase Resistance	0.181	Ω
Rated Current	6.31	A
Rated Voltage	29.7	V
No-load Current	0.12	A
Encoder Type	2500	P/r
Power Rate	28.4	KW/s
Mechanical time constant	0.704	ms

3.4.2. Sơ đồ nối dây mô hình

Mô hình động cơ PMSM ACMC đã được nối dây sẵn để phù hợp cho ứng dụng điều khiển động cơ PMSM. Chi tiết về sơ đồ nối dây được trình bày cụ thể dưới đây:

- Kết nối giữa động cơ PMSM và RapidPro Power Unit

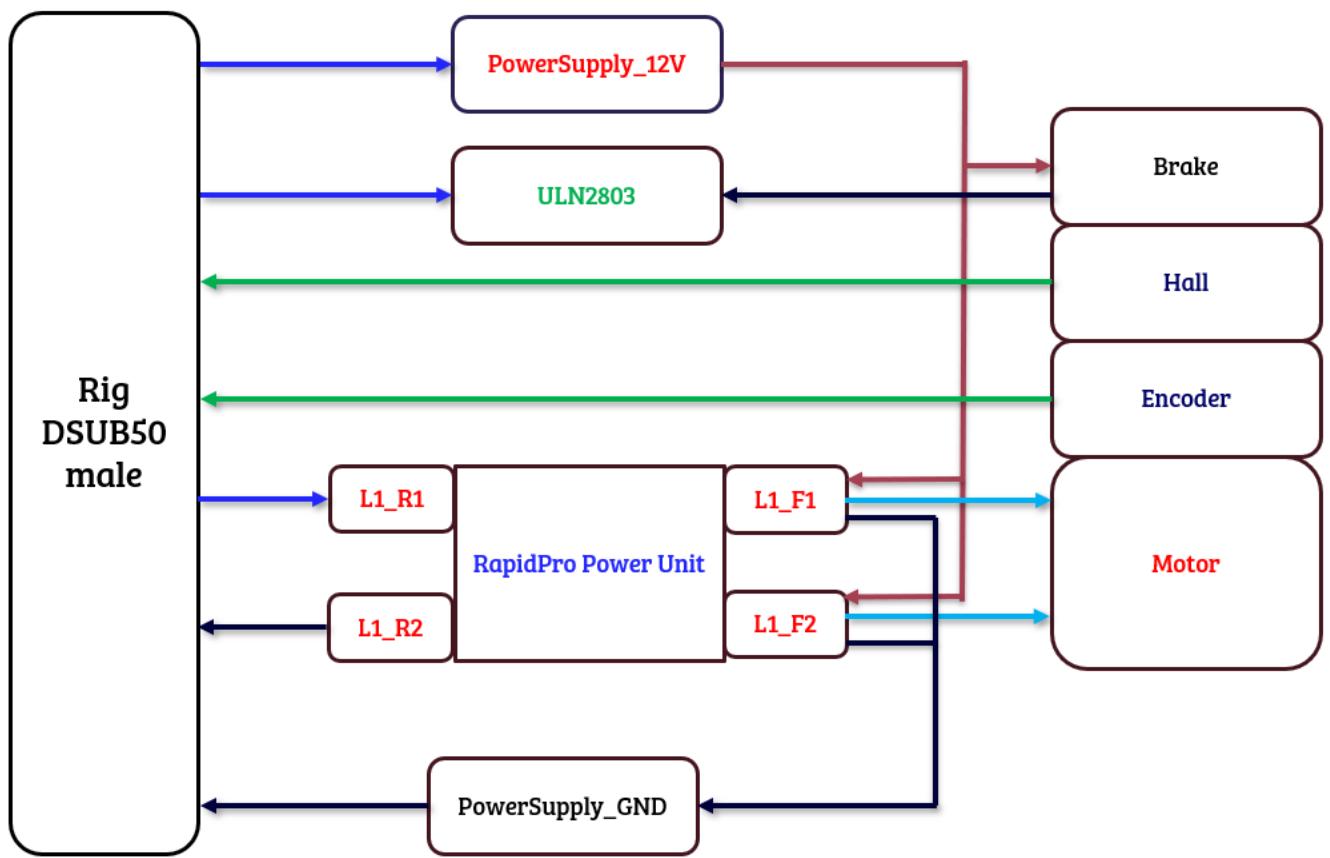


Hình 3.56. Tổng quan kết nối giữa RapidPro và PMSM

Các chân kết nối giữa RapidPro Power Unit và PMSM sẽ được tổng hợp và kết nối vào cổng Rig để từ cổng Rig bắt đầu giao tiếp tín hiệu với các thiết bị bên ngoài.

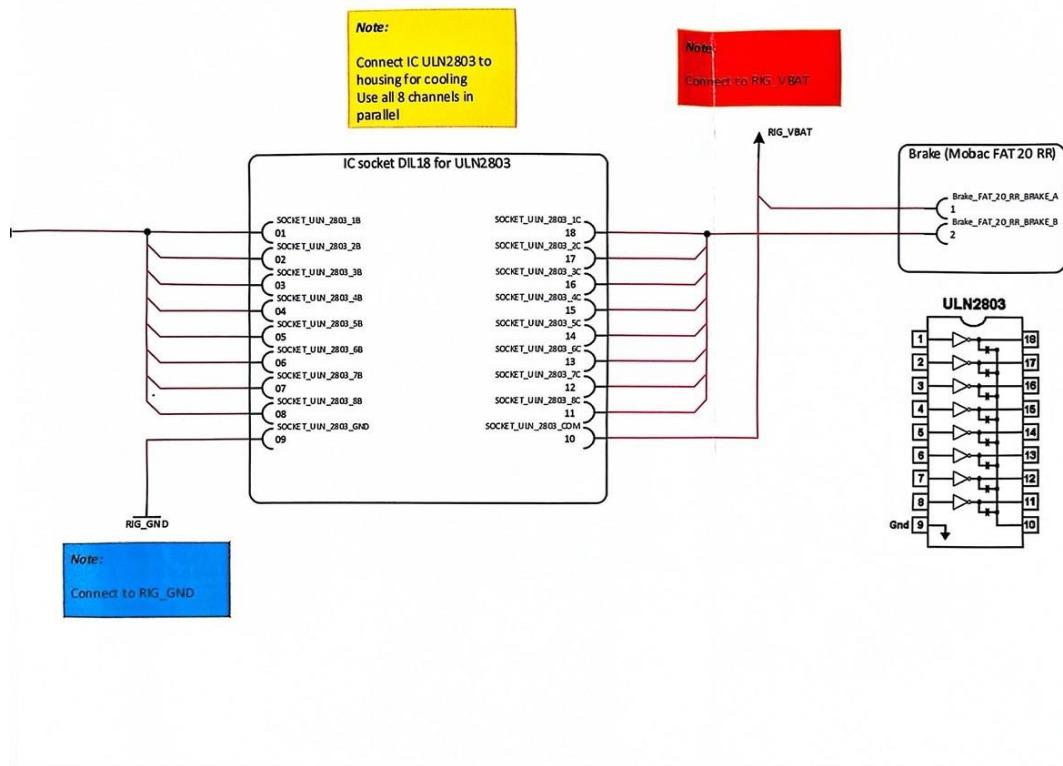


Hình 3.57. Cổng Rig



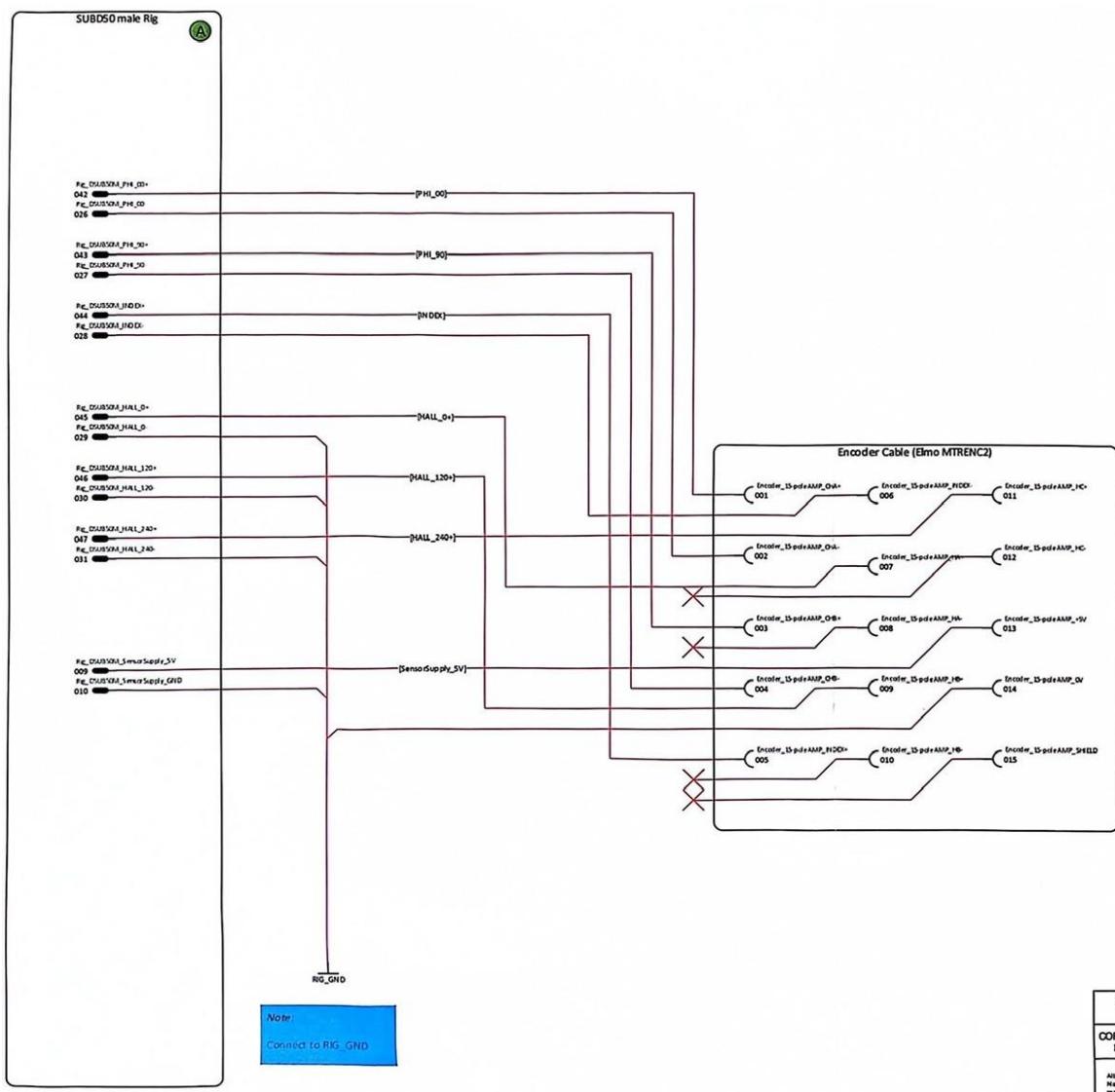
Hình 3.58. Sơ đồ khái niệm về mô hình động cơ PMSM AC/DC

Đây là sơ đồ khái niệm tổng quát các kết nối giữa các module của RapidPro Control Unit và động cơ PMSM tổng hợp kết nối ra cổng Rig. Sơ đồ gồm có: PMSM, các cảm biến Hall/Encoder, phanh, dây nguồn, RapidPro và các module của nó.



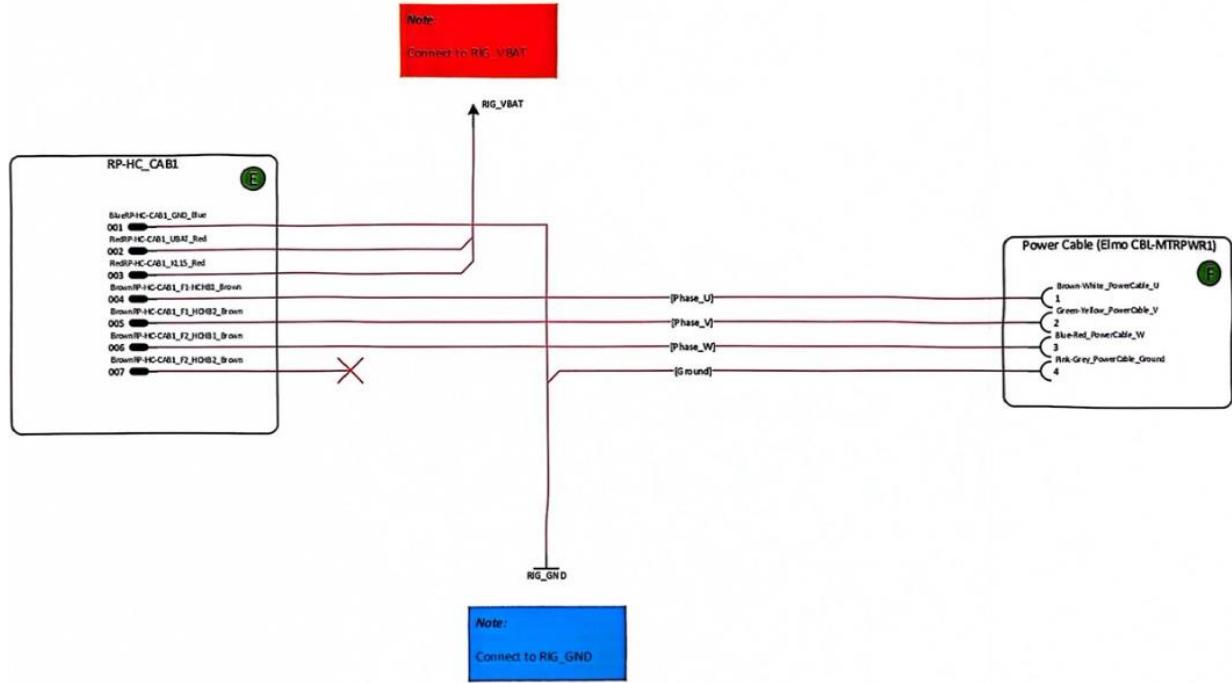
Hình 3.59. Sơ đồ nối dây hệ thống phanh của PMSM

Hệ thống phanh của động cơ PMSM sẽ được điều khiển thông qua IC ULN2803 và được kết nối với cổng Rig để nhận tín hiệu. Nguồn cung cấp cho hệ thống phanh hoạt động sẽ là nguồn 12V.



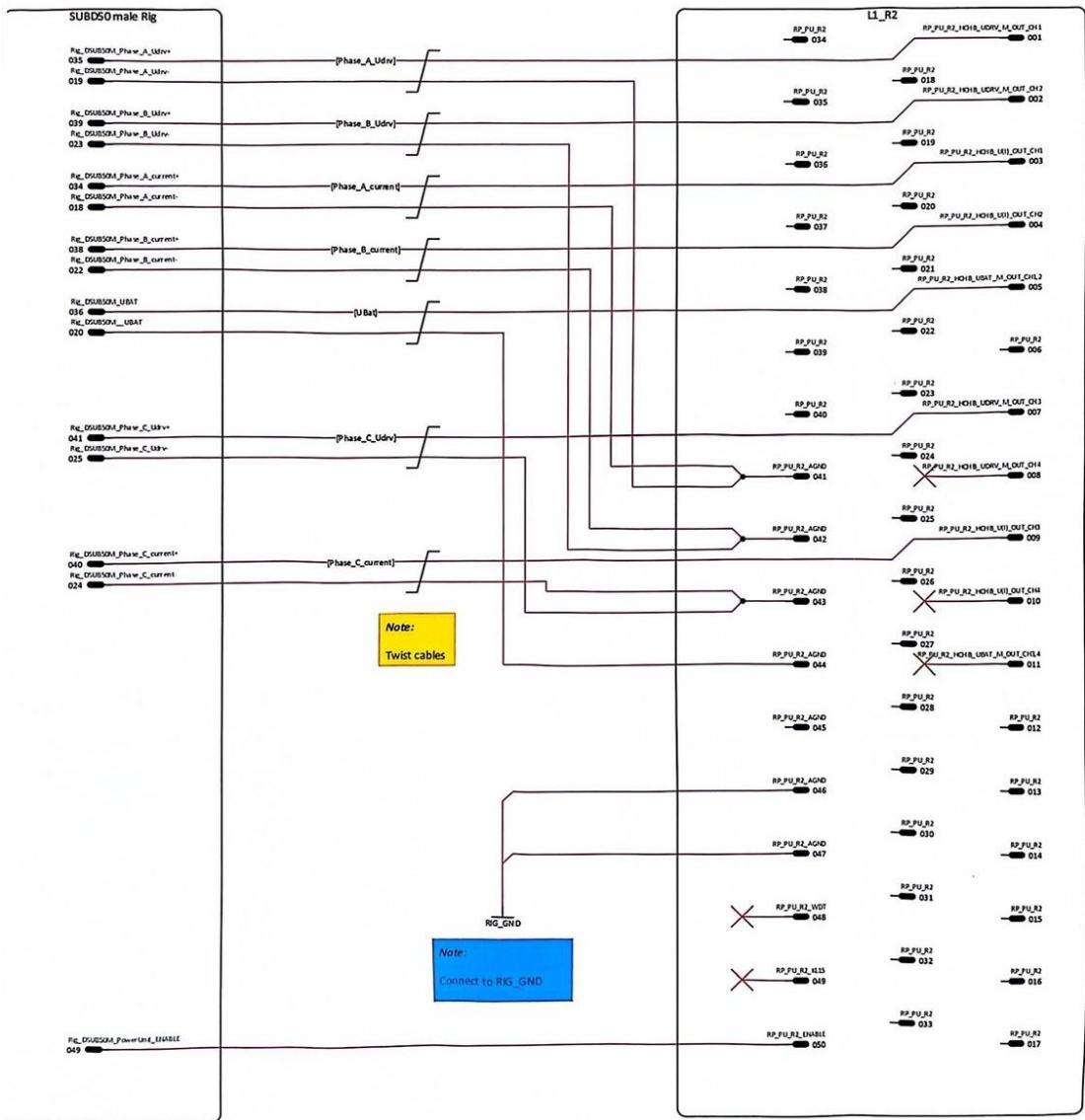
Hình 3.60. Sơ đồ nối dây Encoder và Hall

Sơ đồ dây kết nối cảm biến Hall/Encoder bao gồm việc cấp nguồn cho cảm biến và nhận các tín hiệu đưa về MicroLabBox thông qua cổng Rig.



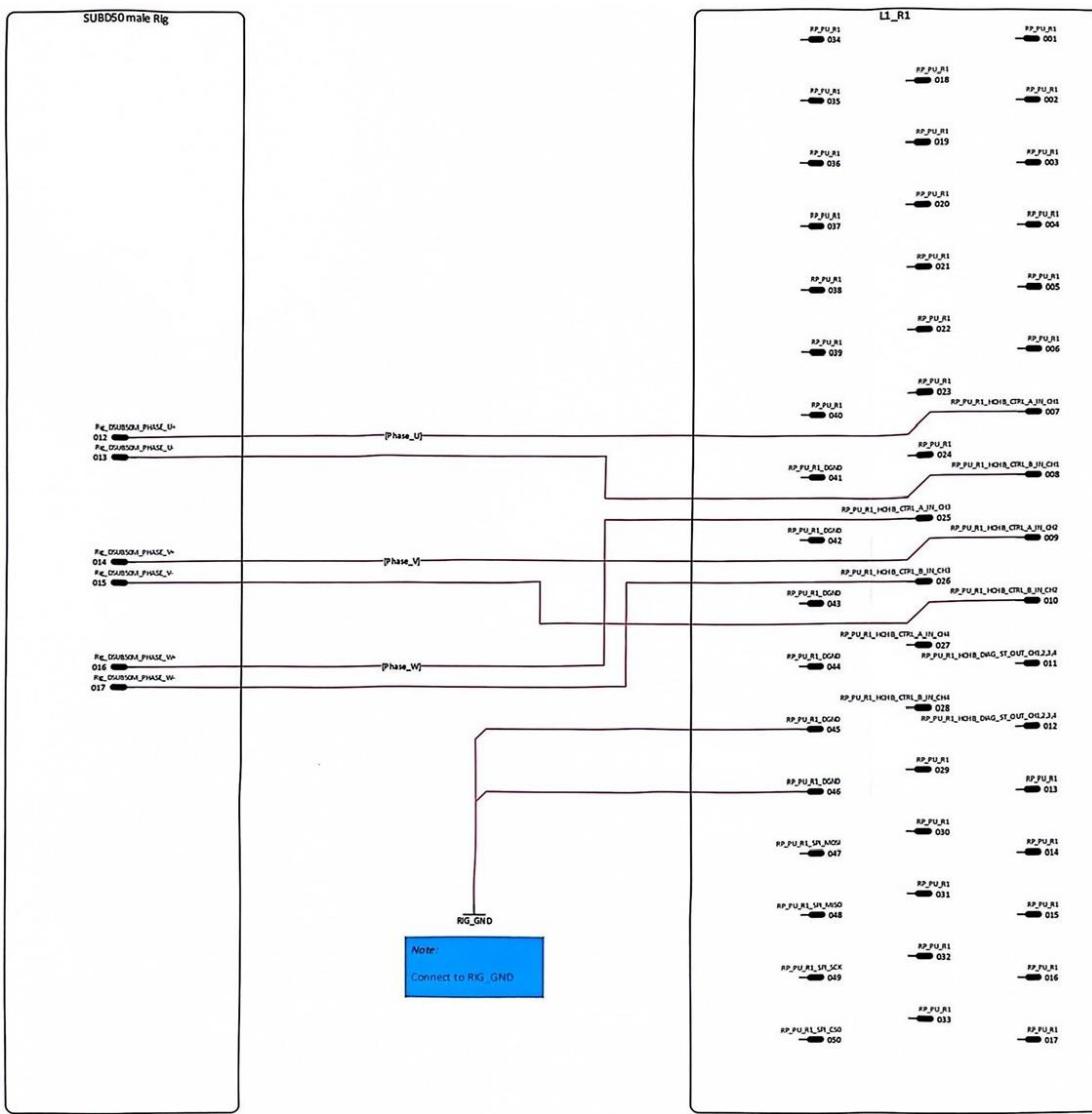
Hình 3.61. Sơ đồ nối dây giữa RapidPro và PMSM

Các tín hiệu điều khiển động cơ từ MicroLabBox sẽ được xử lí và khéch đại thông qua các cầu - H và đi đến động cơ.



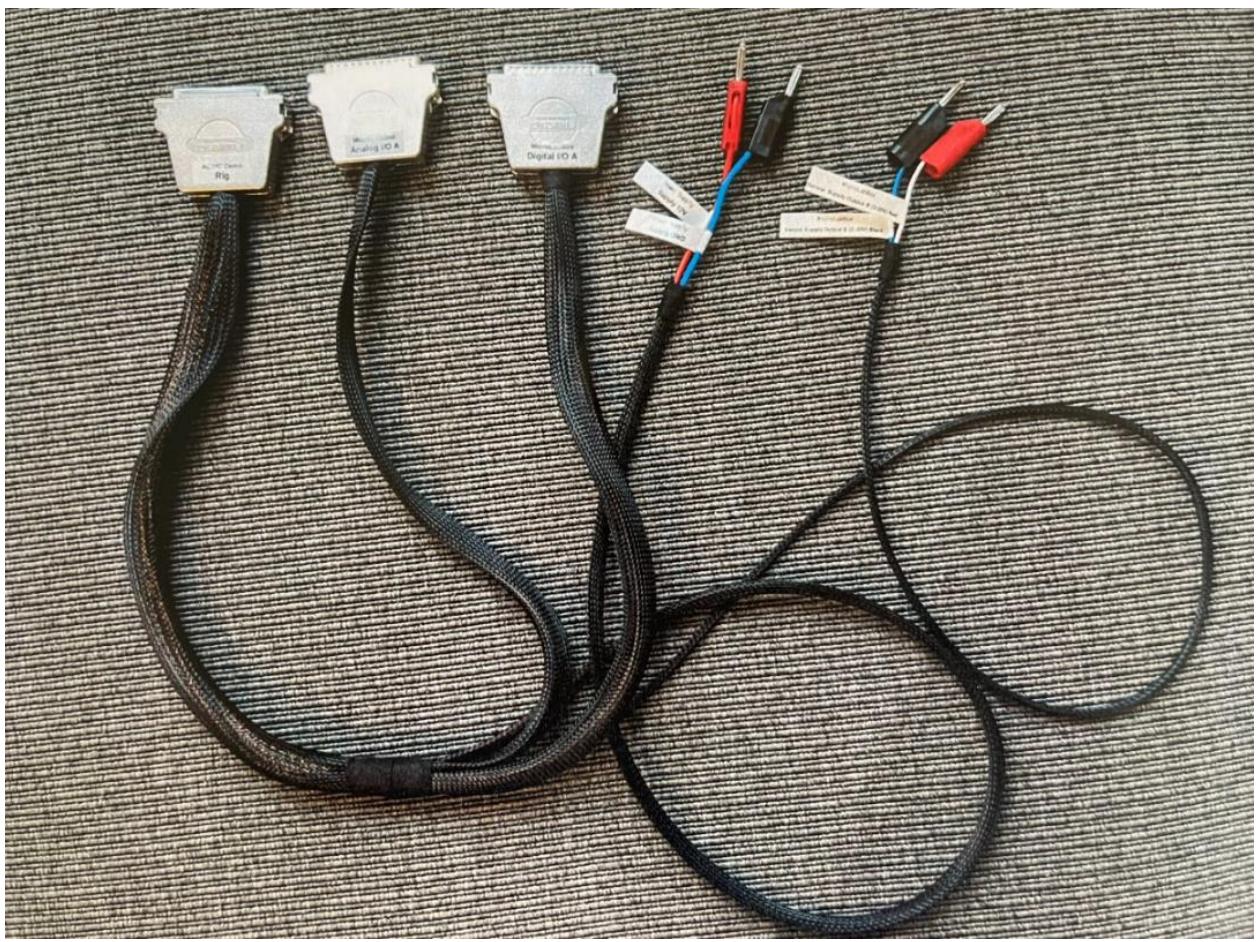
Hình 3.62. Sơ đồ nối dây giữa Rig và L1_R2

Công kết nối L1_R2 gửi tín hiệu dòng điện và điện áp đo đạc được từ 3 phase của động cơ gửi về MicroLabBox.

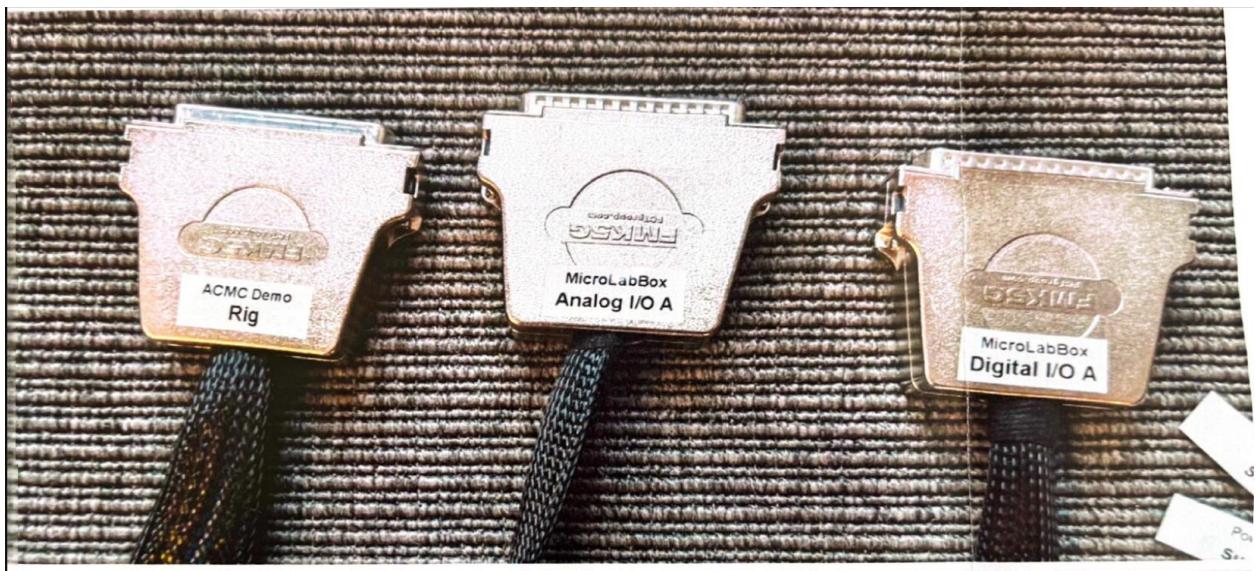


Hình 3.63. Sơ đồ nối dây giữa Rig và L1_R1

Công kết nối L1_R1 nhận tín hiệu điều khiển động cơ dạng PWM từ MicroLabBox đưa vào RapidPro.



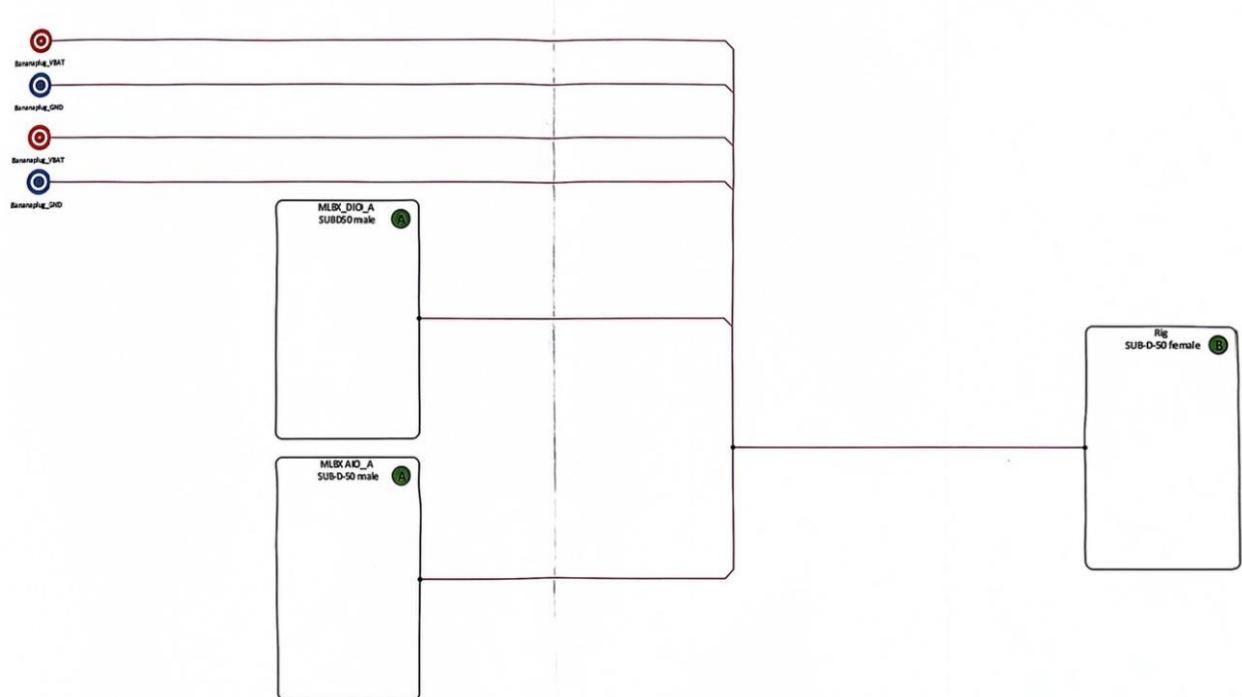
Hình 3.64. Dây kết nối



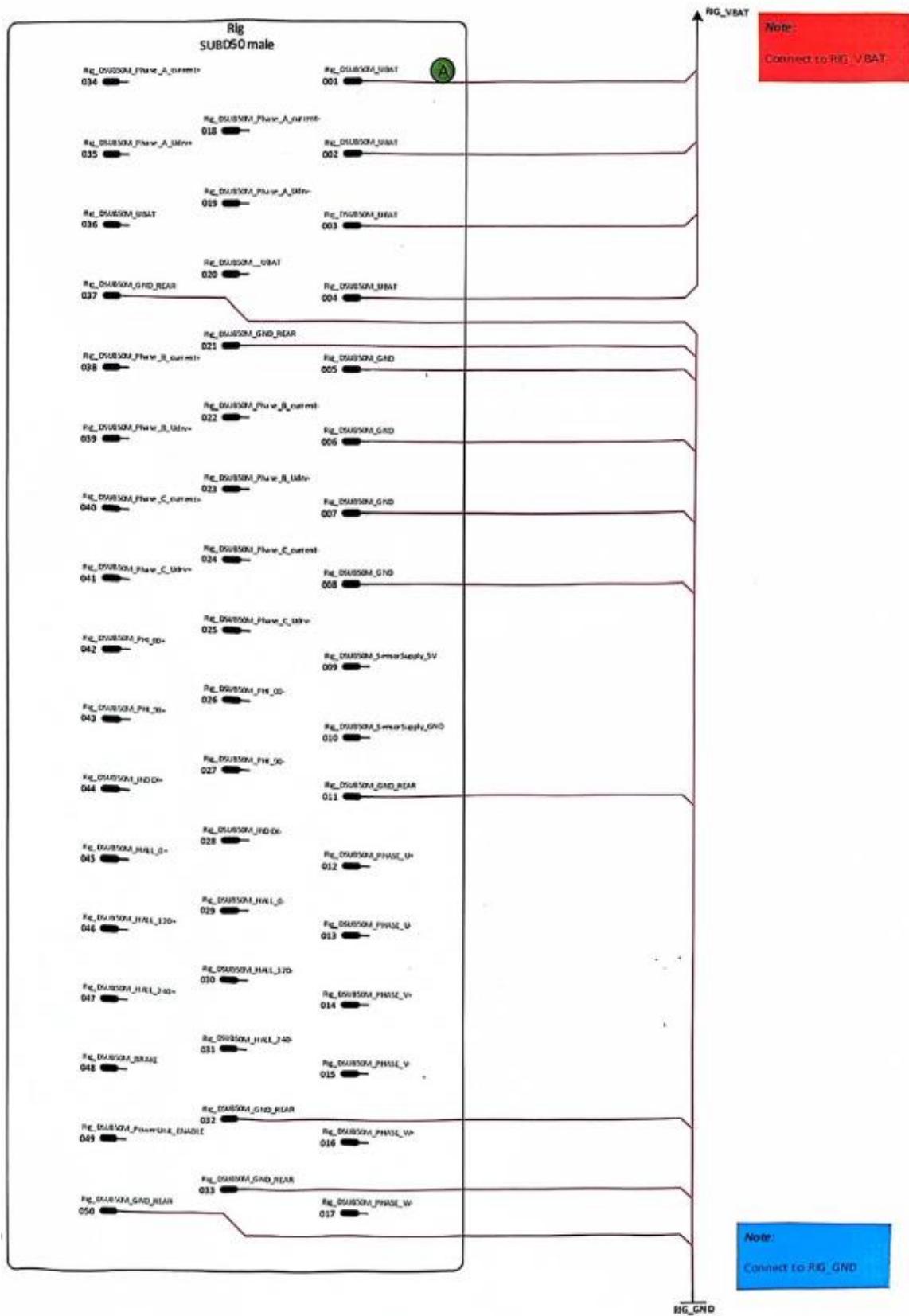
Hình 3.65. Các cổng giao tiếp Digital/Analog



Hình 3.66. Dây nguồn

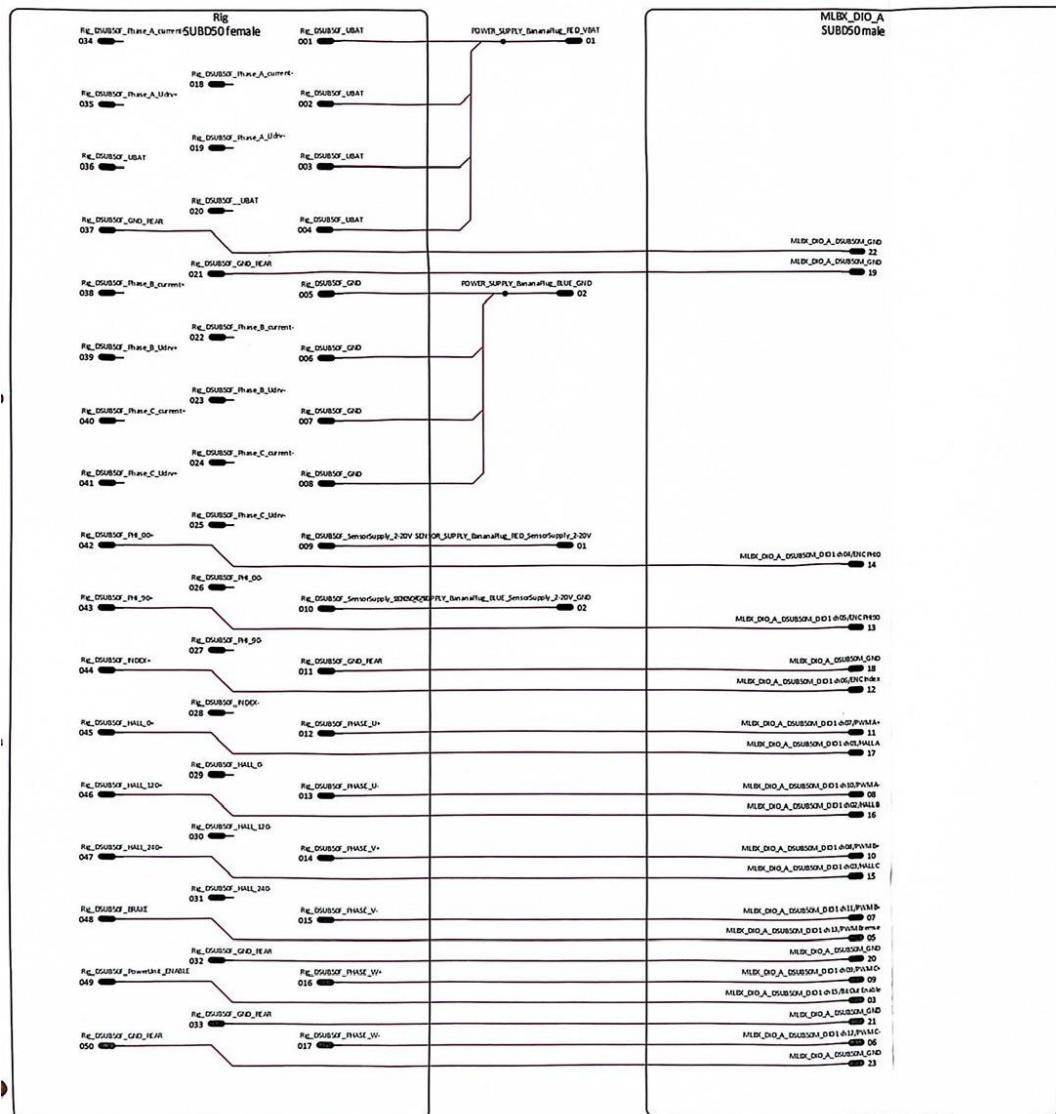


Hình 3.67. Sơ đồ khôi kết nối cổng Rig tới MicroLabBox



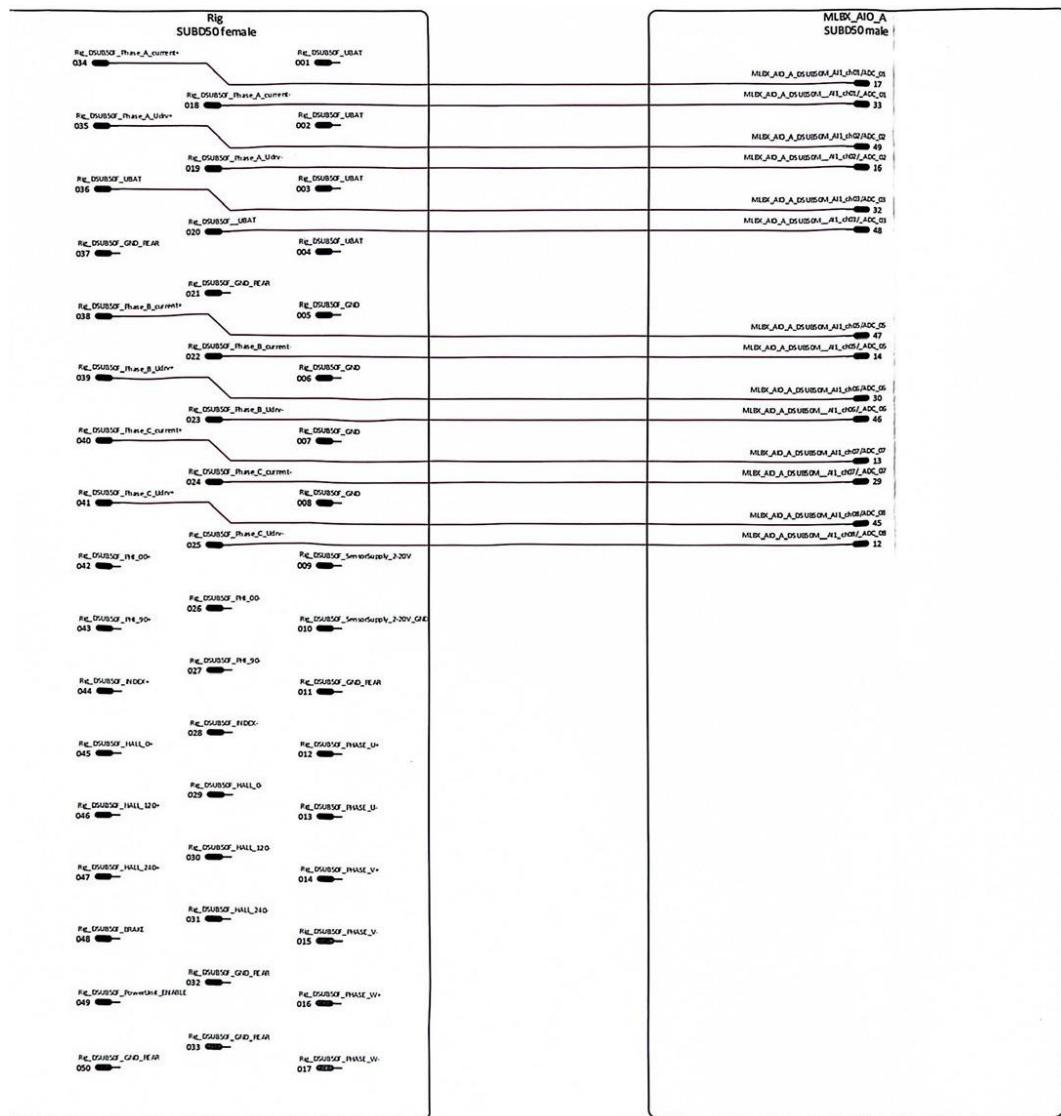
Hình 3.68. Sơ đồ nguồn đi vào công Rig

Nguồn 12V được lấy từ bên ngoài cấp vào cổng Rig. Nguồn này dùng để vận hành RapidPro, cấp nguồn cho động cơ và Encoder_Hall.



Hình 3.69. Sơ đồ kết nối cổng Rig đến cổng kết nối Digital kênh A

Cổng kết nối digital kênh A sẽ nhận các tín hiệu dạng số từ Encoder và cảm biến Hall đưa về MicroLabBox để xử lý, đồng thời các tín hiệu xung PWM điều khiển được gửi xuống điều khiển động cơ.



Hình 3.70. Sơ đồ kết nối Rig đến công két nối Analog kênh A

Công két nối analog kênh A nhận các giá trị dòng điện của 3 phase U, V, W của động cơ vào MicroLabBox.

CHƯƠNG 4: VẬN HÀNH MÔ HÌNH PMSM ACMC CỦA DSPACE

4.1. Nghiên cứu, chế tạo module mở rộng tín hiệu digital và analog của MicroLabBox

MicroLabBox cung cấp rất nhiều đầu ra I/O như digital và analog, với một MicroLabBox cơ bản đã có 60 kênh tín hiệu digital và 48 kênh tín hiệu analog. Tuy nhiên, các chân tín hiệu này thường được quy hoạch riêng biệt từng giắc cắm như việc 32 kênh tín hiệu digital vào một giắc sub-DB50 hoặc 32 chân tín hiệu analog vào một giắc sub-DB50. Việc này gây khó khăn trong việc kết nối đến các hệ thống phức tạp với nhiều thiết bị, cảm biến ở các vị trí cách xa nhau. Hoặc đơn giản hơn, với mục đích giáo dục, các giắc

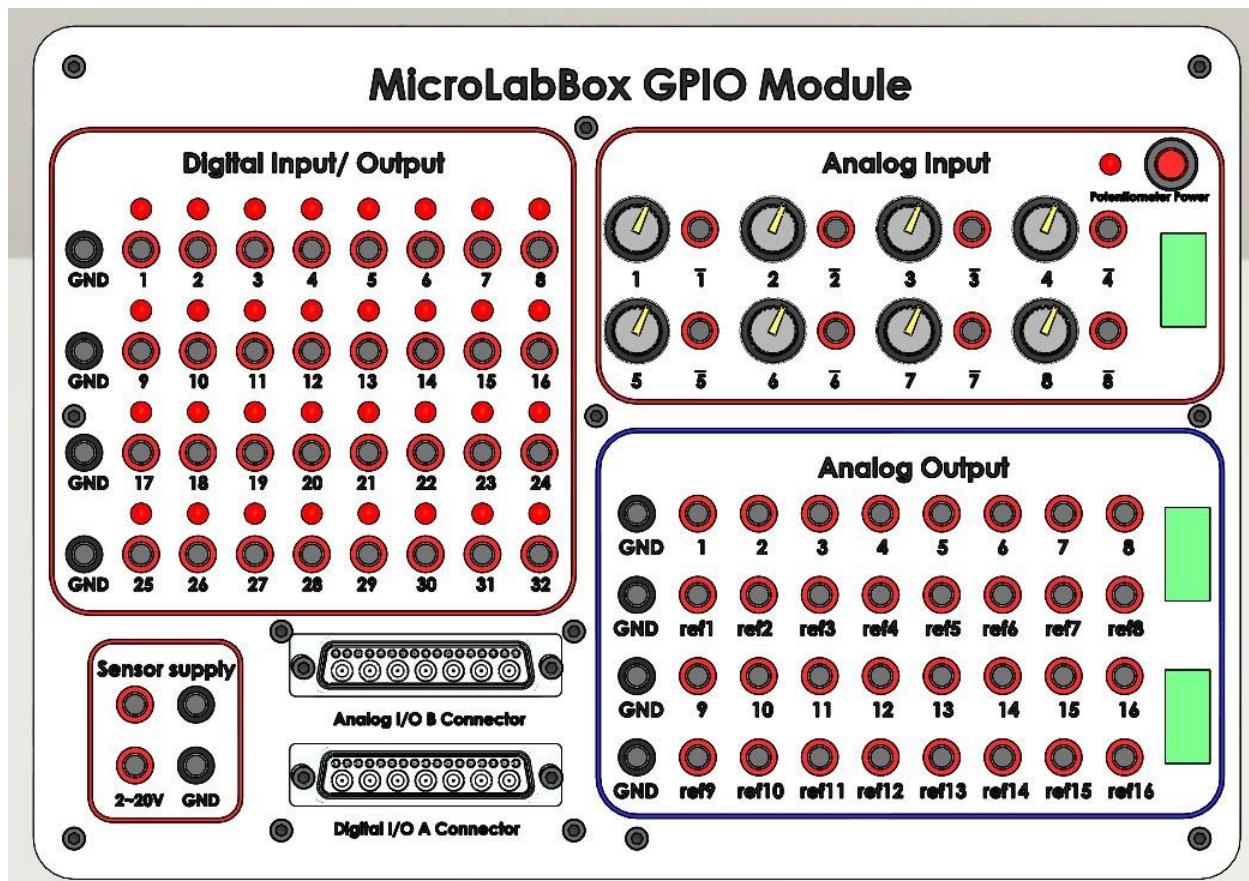
cắm MicroLabBox làm cho học giả khó tiếp cận với các I/O, đặc biệt là với những người bắt đầu tìm hiểu về phần cứng này. Vì vậy, để sử dụng MicroLabBox một cách dễ dàng, thuận tiện hơn, nhóm đã lên ý tưởng thiết kế một module mở rộng một số chân I/O cơ bản.

4.1.1. Ý tưởng và thiết kế

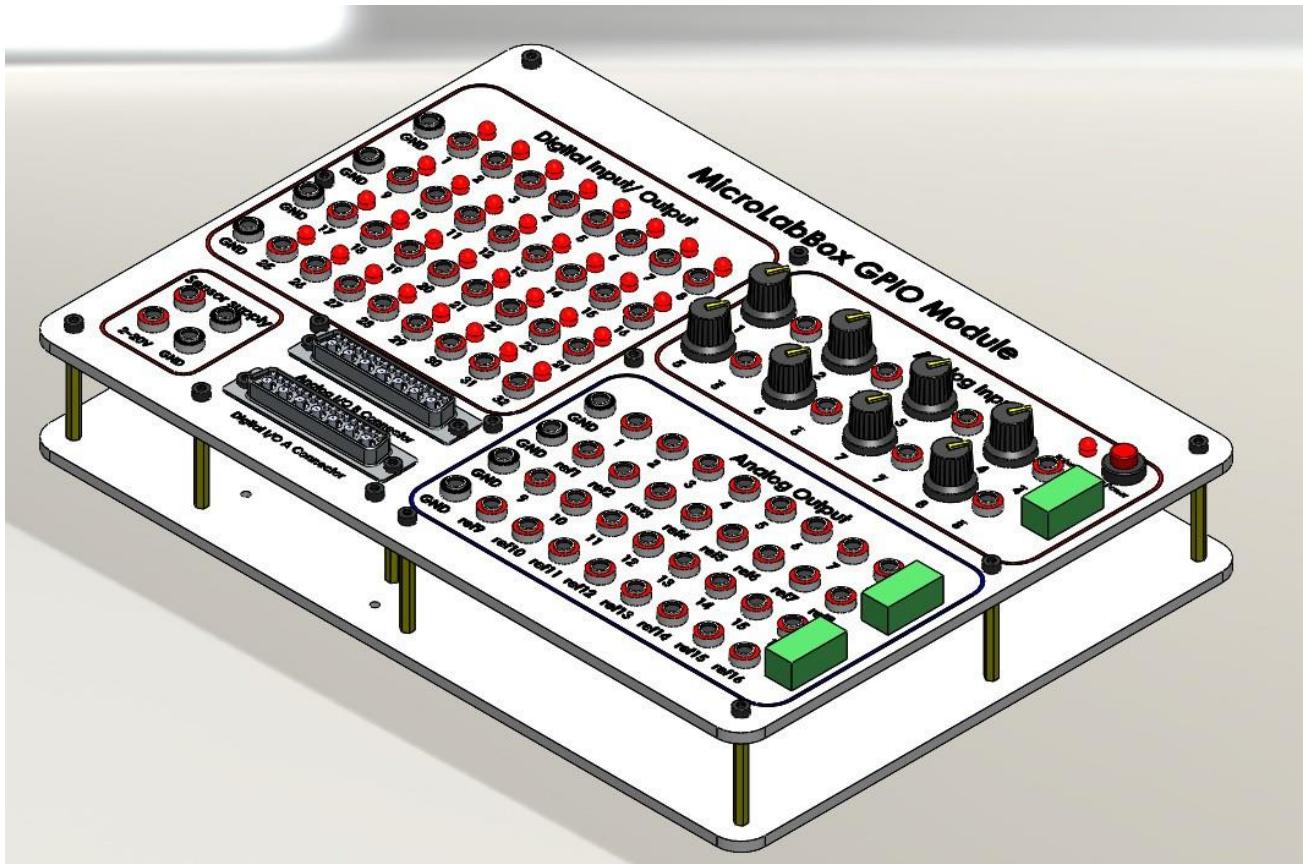
MicroLabBox có bốn giắc cắm tín hiệu I/O gồm Digital I/O A Connector, Digital I/O B Connector, Analog I/O A Connector và Analog I/O B Connector với đầy đủ các chức năng cơ bản của các tín hiệu analog, digital. Sau khi suy nghĩ và lên ý tưởng, nhóm đã quyết định mở rộng hai giắc Digital I/O A Connector và Analog I/O B Connector vì một số lí do sau đây: chỉ cần mở rộng hai giắc này đã có đầy đủ các chức năng đầu vào/đầu ra của tín hiệu digital và analog, đồng thời tiết kiệm được 2 giắc còn lại để sử dụng cho các mục đích khác yêu cầu cần đầy đủ một giắc cắm dạng digital hoặc analog.

Module này bao gồm:

- 32 chân lấy tín hiệu digital I/O class 1 kèm với 32 con LED
- 8 tín hiệu analog input được tích hợp sẵn với 8 biến trở
- 8 chân đảo điện áp của analog input
- 16 chân lấy tín hiệu analog output
- 16 chân tham chiếu điện áp của analog output
- 1 header 8 chân ra của analog input (trường hợp không dùng biến trở)
- 2 header 16 chân ra của analog output
- Công tắc nguồn biến trở (ngắt biến trở để sử dụng các cảm biến bên ngoài)
- Nguồn từ Sensor Supply B cấp cho biến trở và các thiết bị khác (từ 2-20V tùy chỉnh bằng Matlab/Simulink)
- 2 sub-DB50 để kết nối với MicroLabBox

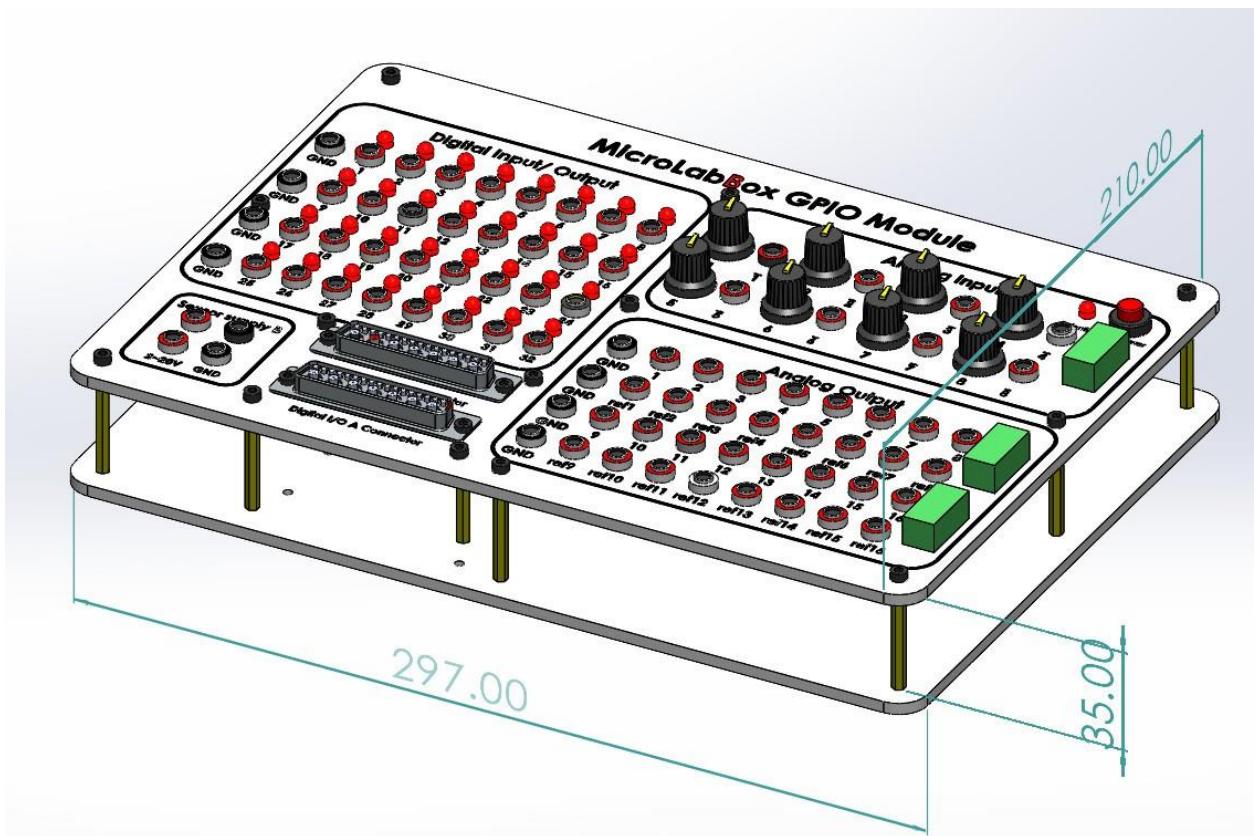


Hình 4.1. Module thiết kế bằng Solidworks (2D)



Hình 4.2. Module thiết kế bằng Solidworks (3D)

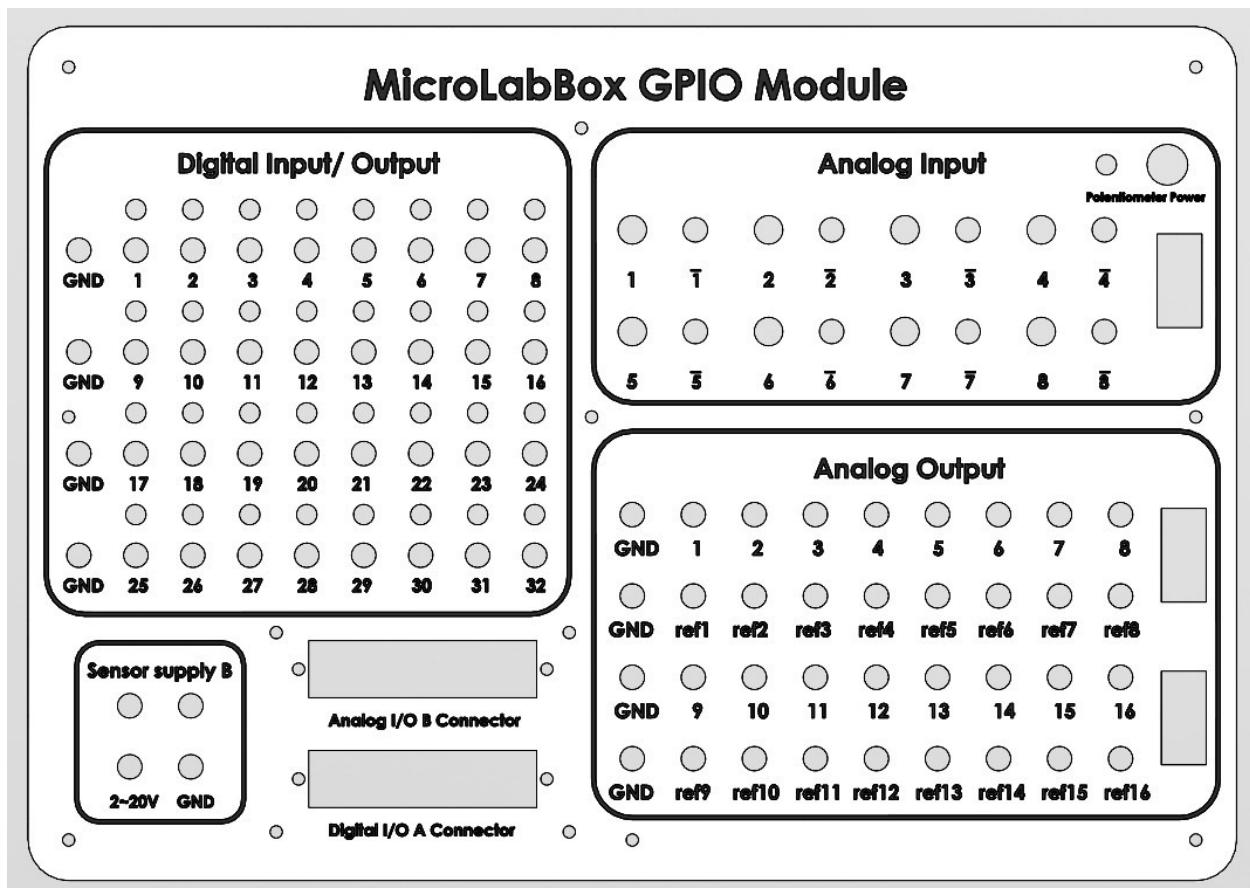
4.1.2. Chế tạo module hoàn chỉnh



Hình 4.3. Kích thước của module

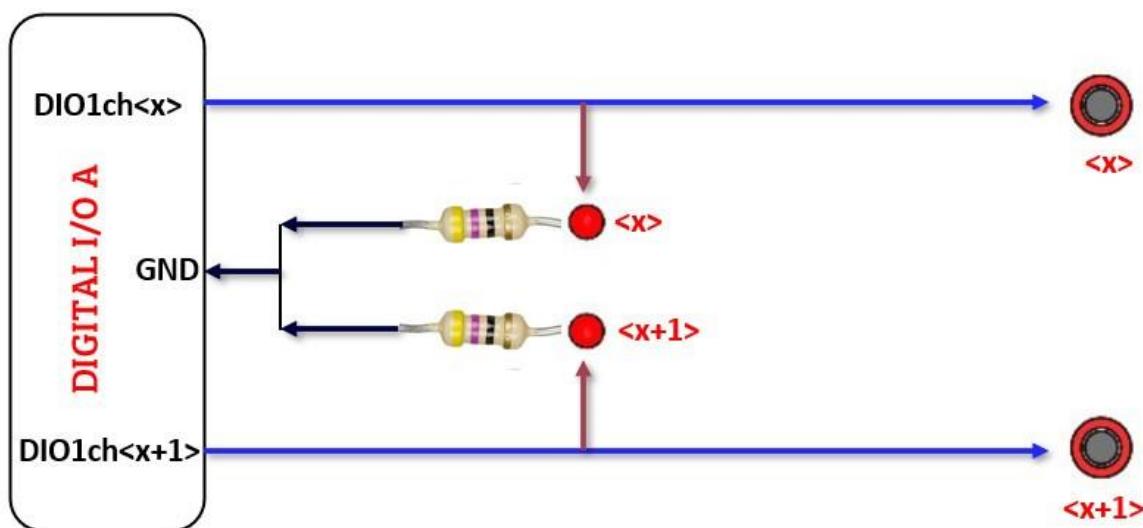
Các linh kiện cần thiết:

- x80 Đế banana 2mm cái dài 23mm
- x33 LED 5mm
- X32 điện trở 220Ω và 1 điện trở $1k\Omega$
- x3 header 8 slots
- x1 nút nhấn
- x8 biến trở $10k\Omega$
- x2 sub-db50 cái
- x17 Trụ đồng HEX-M3 5mm + ốc HEX-M3
- 2 tấm MICA 297x210mm được đục theo kích thước thiết kế sẵn.



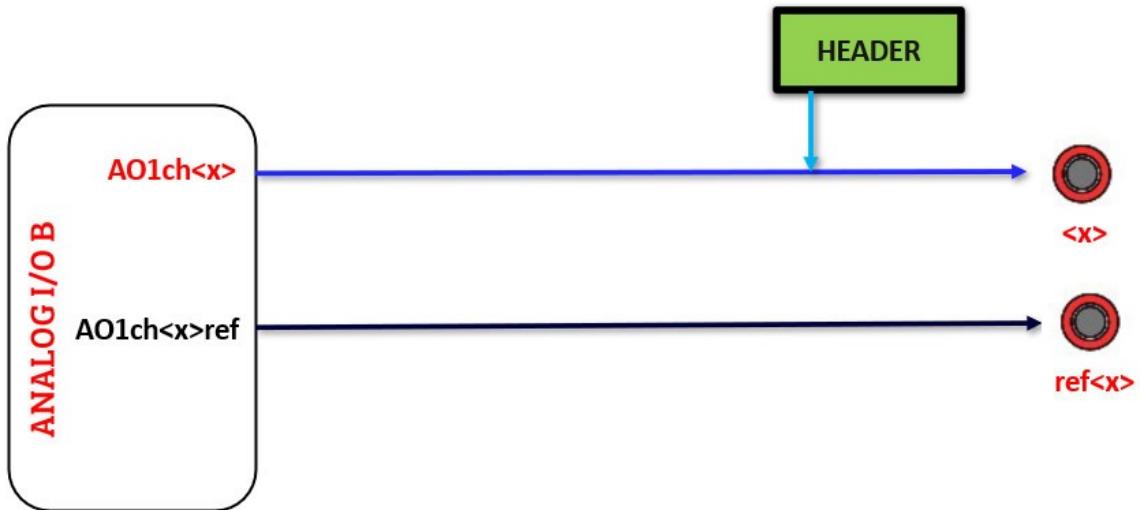
Hình 4.4. Bản vẽ in MICA

Sau khi thiết kế và chuẩn bị đầy đủ các linh kiện và thiết bị để thực hiện chế tạo module. Nhóm tiến hành đi dây và lắp đặt module



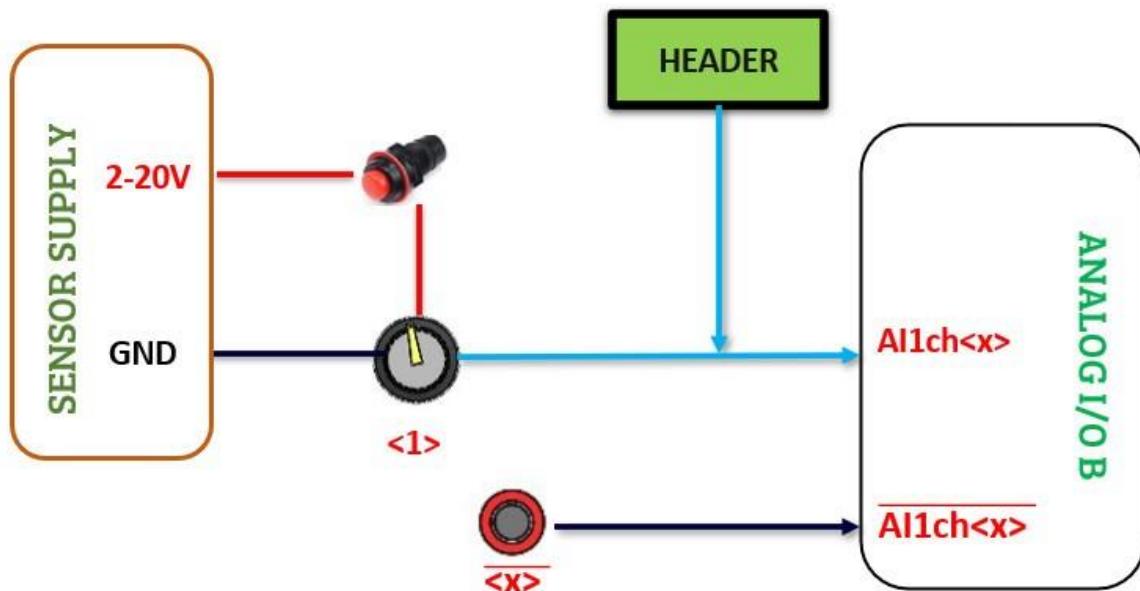
Hình 4.5. Sơ đồ đi dây cho các chân digital và LED

Các chân digital của I/O A Connector được đấu dây qua module như hình vẽ trên, lưu ý 2 LED sử dụng digital gần nhau sử dụng chung 1 chân GND gần nhất. Không nên dùng 1 chân GND cho nhiều LED để tránh gây mất ổn định tín hiệu.



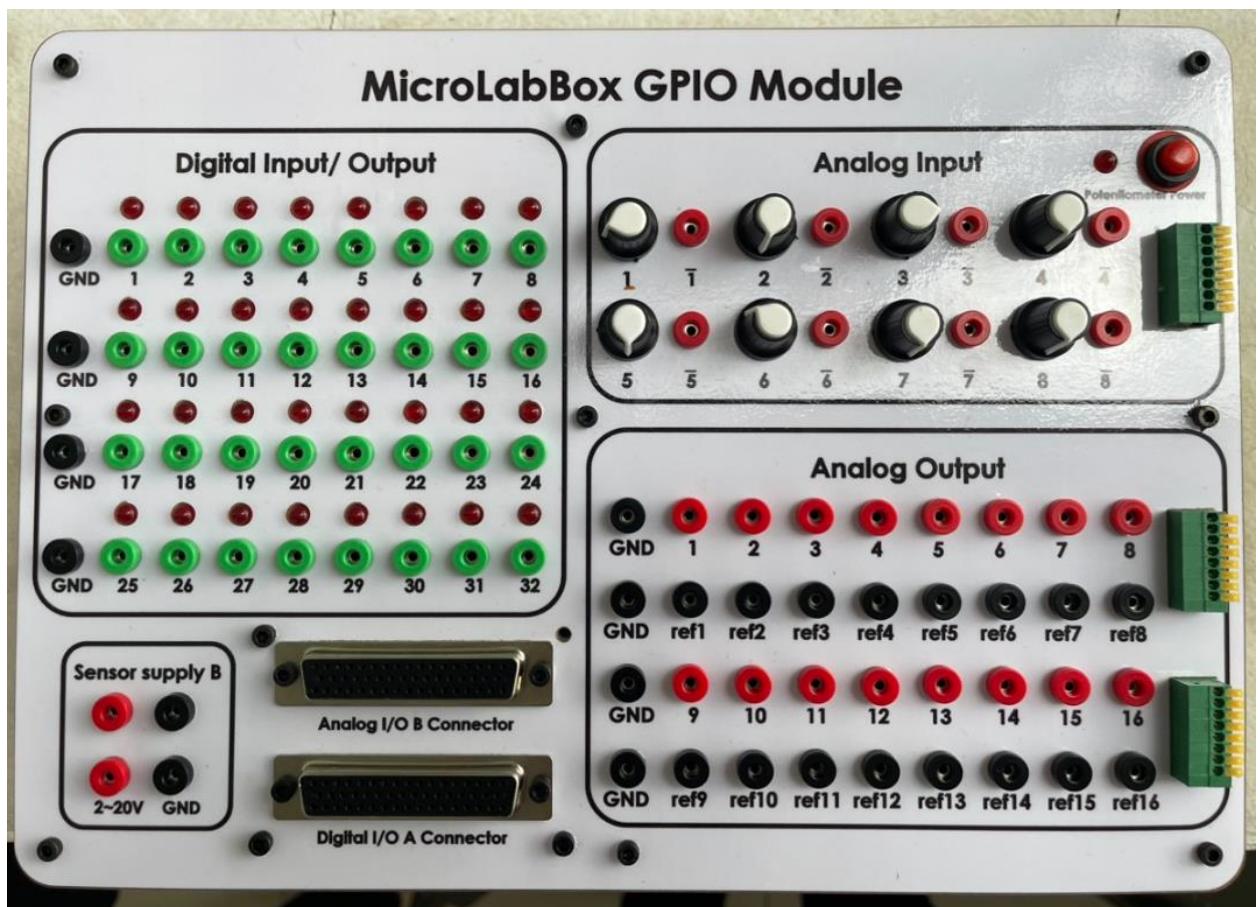
Hình 4.6. Đi dây cho các chân analog output

Tín hiệu analog output được đi dây như hình vẽ trên, mỗi chân analog output có một chân tham chiếu tương ứng để tránh việc nhiễu tín hiệu.



Hình 4.7. Sơ đồ đi dây cho analog input

Tín hiệu analog input được đi dây như hình vẽ trên, chân tín hiệu biến trở là các chân analog input, nguồn biến trở được cấp từ sensor supply. Nút nguồn dùng để ngắt tín hiệu từ cảm biến và lấy analog input từ bên ngoài (cảm biến,...). Chân \bar{x} là chân điện áp đảo ngược, giúp cho tín hiệu analog input có thể đảo thành giá trị âm.



Hình 4.8. Mô hình hoàn thiện

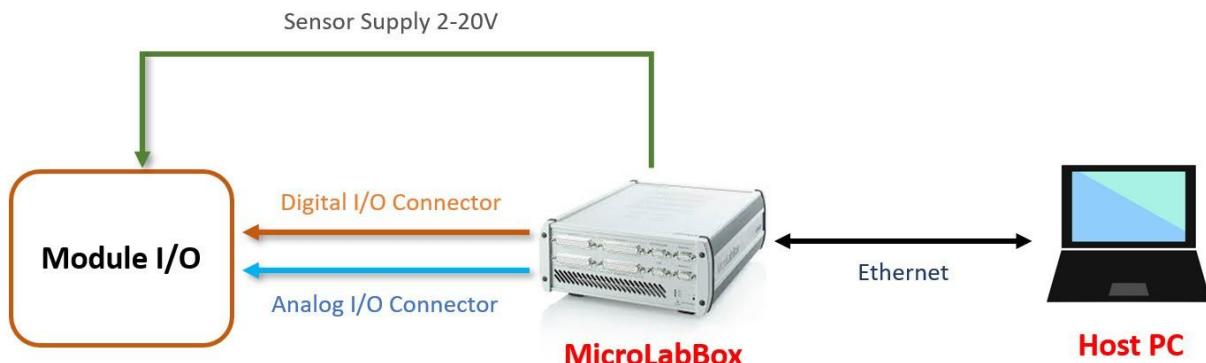
Sau đó, tiến hành chế tạo dây cáp kết nối giữa MicroLabBox và module.



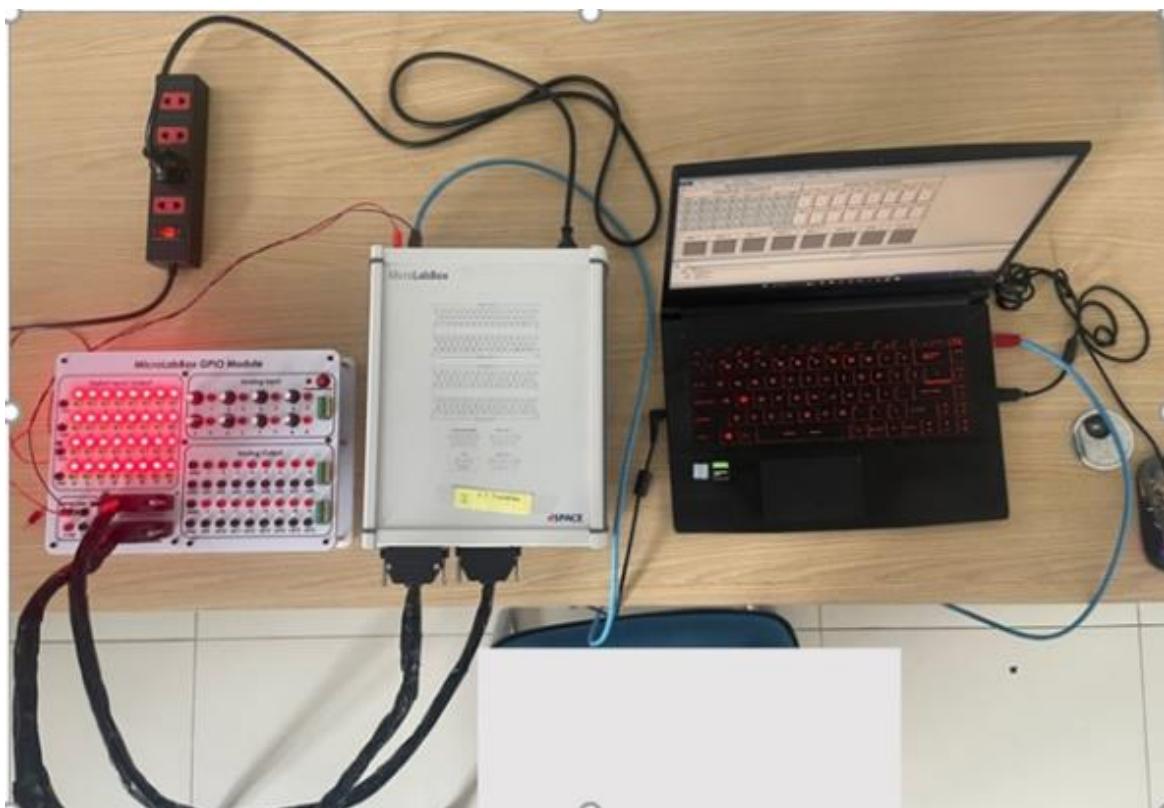
Hình 4.9. Dây cáp kết nối

4.1.3. Vận hành module

Sơ đồ đấu dây của mô hình:

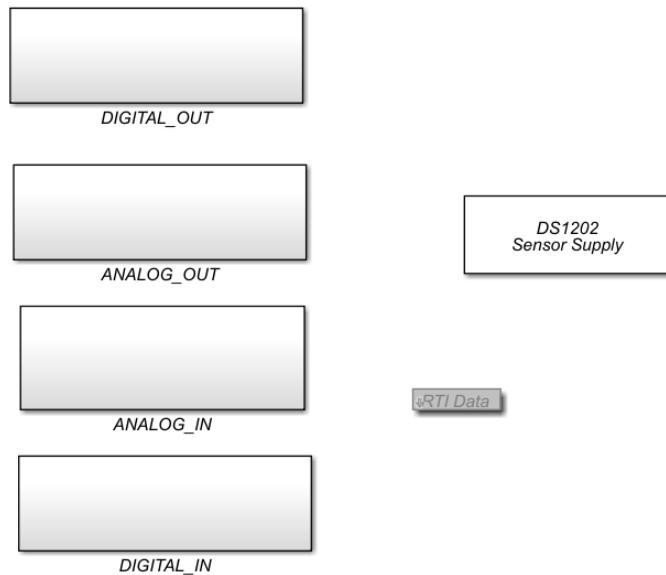


Hình 4.10. Sơ đồ kết nối



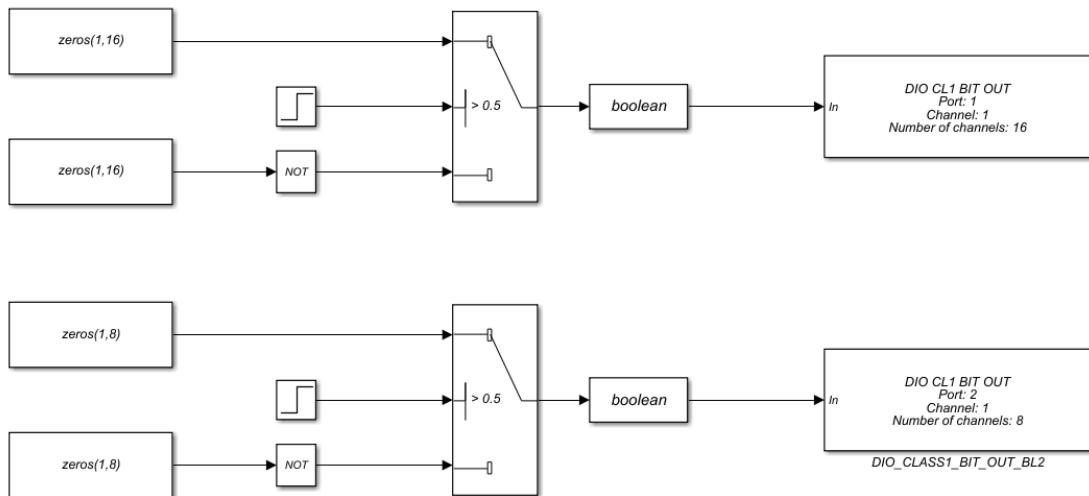
Hình 4.11. Sơ đồ đấu dây cho module (thực tế)

Để điều khiển được các chức năng cơ bản của module như: xuất tín hiệu dạng digital, analog, nhận tín hiệu từ cảm biến như biến trở, cảm biến nhiệt độ, ... cần phải thiết kế thuật toán điều khiển bằng matlab và thiết kế bộ điều khiển bằng ControlDesk.



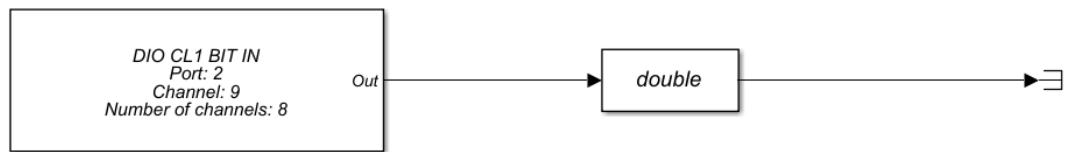
Hình 4.12. Thuật toán trên Matlab

Thuật toán điều khiển gồm 3 tín hiệu chính và một khối điều chỉnh điện áp cung cấp cho biến trở và nguồn của module.



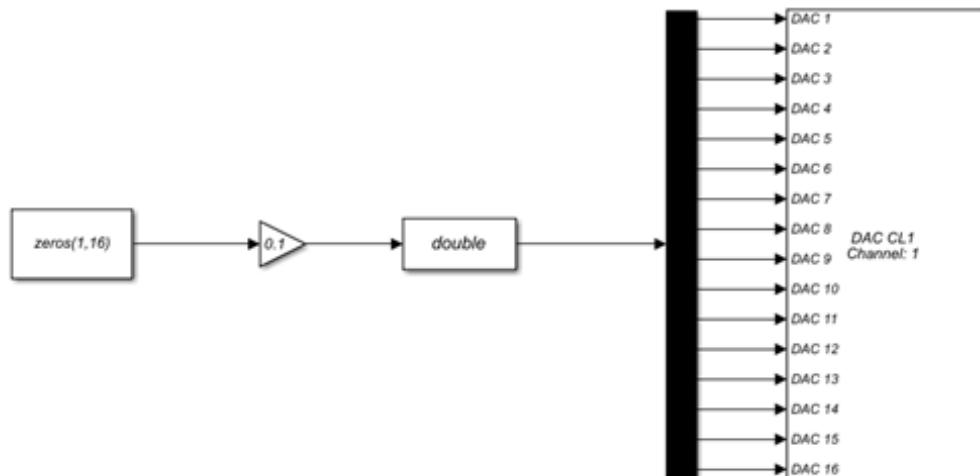
Hình 4.13. Điều khiển tín hiệu digital output

Với tín hiệu digital input, 24 cổng được cài ở chế độ đầu ra. Khi bắt đầu vận hành module, 24 LED sẽ sáng 2 giây rồi tắt để kiểm tra hoạt động của LED. Sau đó việc sáng/tắt LED phụ thuộc vào bộ điều khiển.



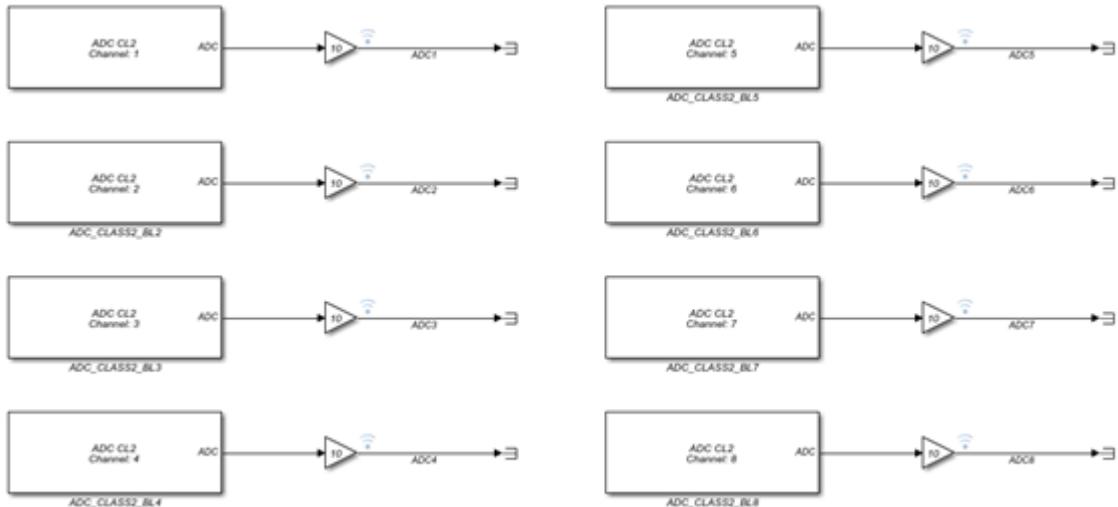
Đọc tín hiệu digital input

Tín hiệu digital input được cài đặt ở 8 chân cuối cùng của module(25-32), dùng để đọc tín hiệu HIGH/LOW.



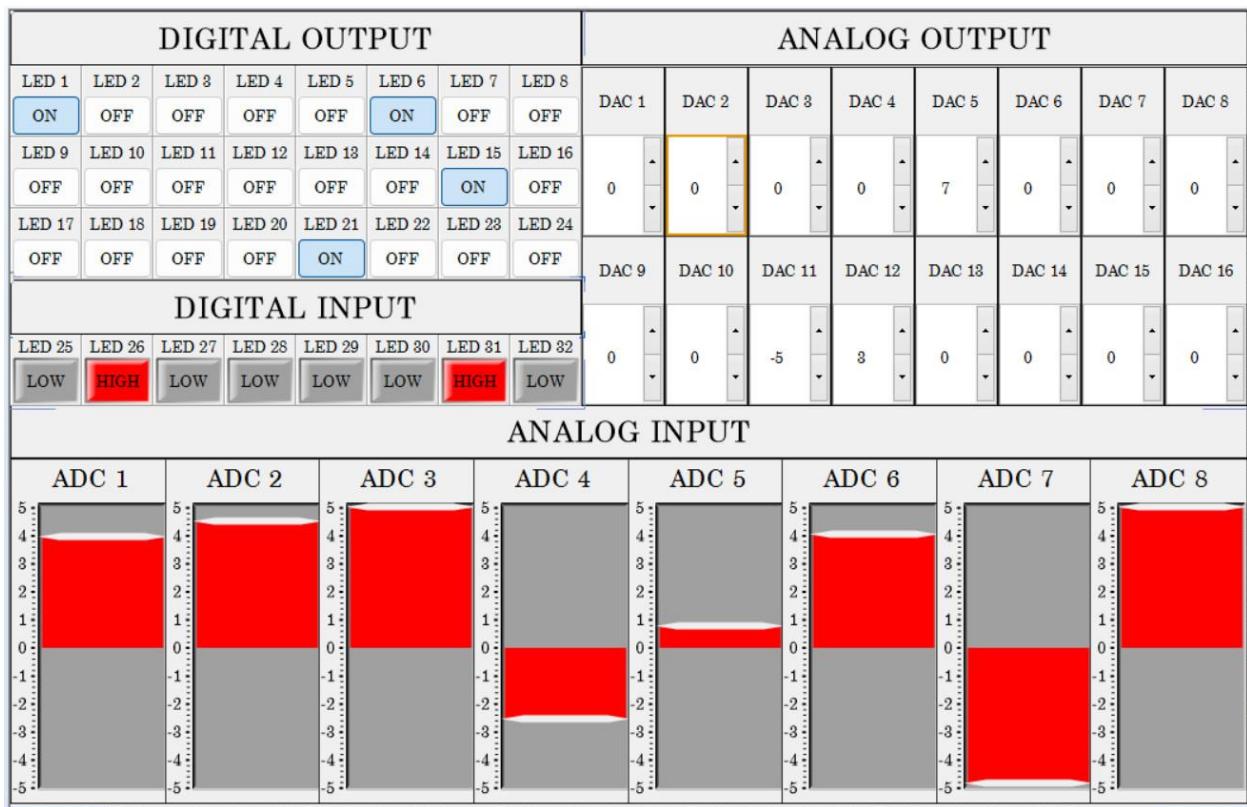
Hình 4.14. Tín hiệu Analog Output

Các tín hiệu analog output đều được cài ở mức 0V khi mô hình bắt đầu vận hành.



Hình 4.15. Đọc tín hiệu analog input

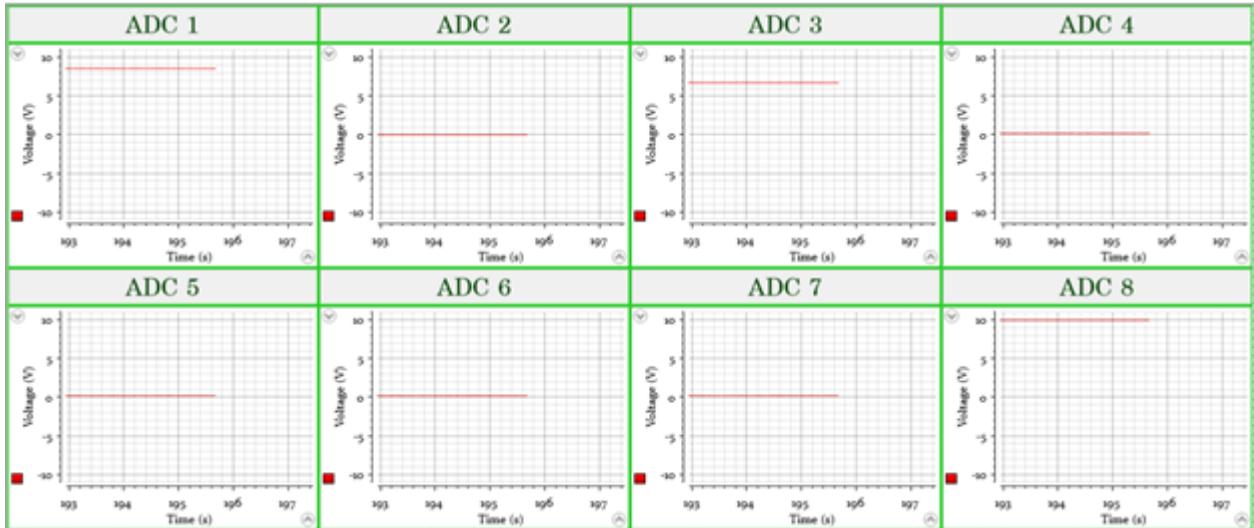
Các tín hiệu của biến trở được đọc và gửi về bộ điều khiển.



Hình 4.16. Bộ điều khiển module trên ControlDesk

Với bộ điều khiển, có thể điều chỉnh tín hiệu digital output của module cùng với các trạng thái LED tương ứng và hiển thị trạng thái tín hiệu digital input. Các tín hiệu analog output cũng được tùy chỉnh ở các mức điện áp -10 đến 10V tùy vào nhu cầu sử dụng. Đồng

thời, tín hiệu analog input sẽ được hiển thị dưới dạng điện áp bằng các thanh trượt hiển thị hoặc các đồ thị.



Hình 4.17. Đồ thị điện áp của biến trở

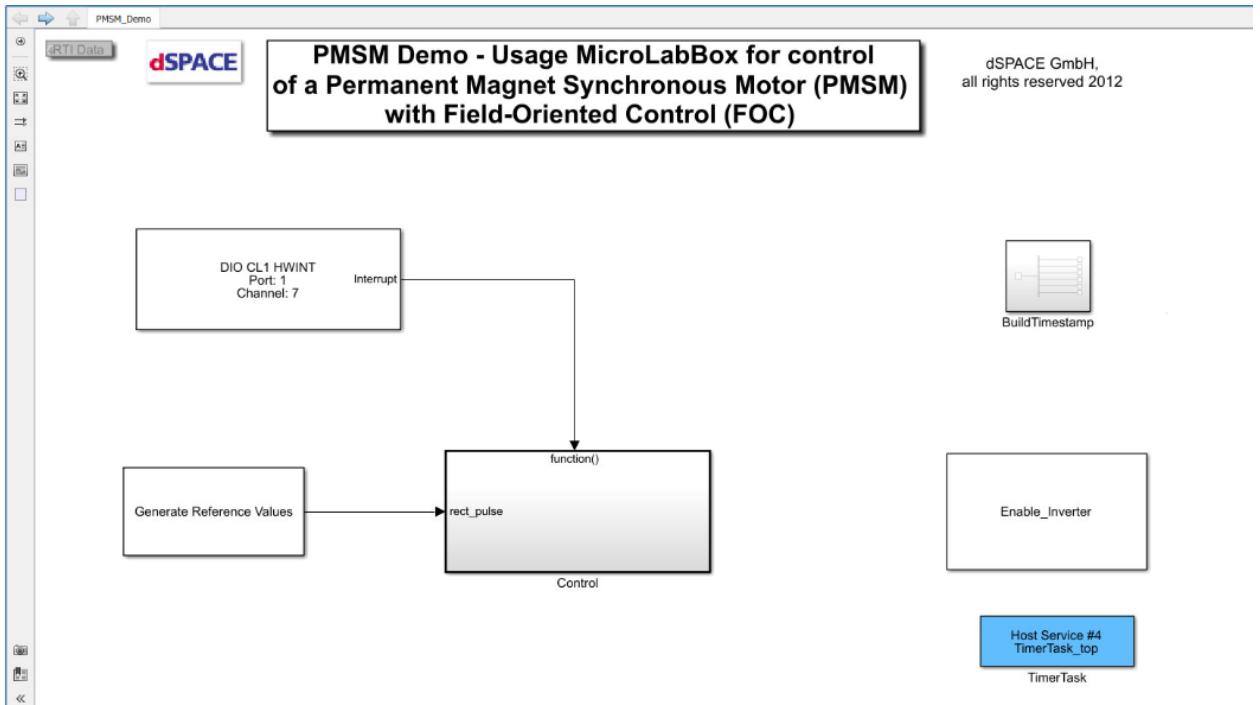
Để kiểm tra các chức năng của module hoạt động bình thường, ta dùng các cách sau đây:

- Với digital output, bật/tắt đèn bằng bộ điều khiển và quan sát module đáp ứng thực tế, đồng thời dùng đồng hồ đo điện áp các chân tương ứng xem các trạng thái HIGH/LOW của chân digital.
- Với digital input, khi cấp tín hiệu HIGH/LOW vào thì đèn LED sẽ sáng/tắt, đồng thời trạng thái của chân sẽ được hiển thị trên bộ điều khiển.
- Với analog output, điều điện áp đầu ra mong muốn của từng chân trên bộ điều khiển, sau đó dùng đồng hồ đo điện áp tương ứng.
- Với analog input, vặn biến trở và quan sát trên đồ thị hoặc các thanh trượt hiển thị khả năng đọc tín hiệu.

4.2. Vận hành mô hình PMSM ACMC của dSPACE

4.2.1. Mô hình bộ điều khiển PMSM trên Matlab/Simulink

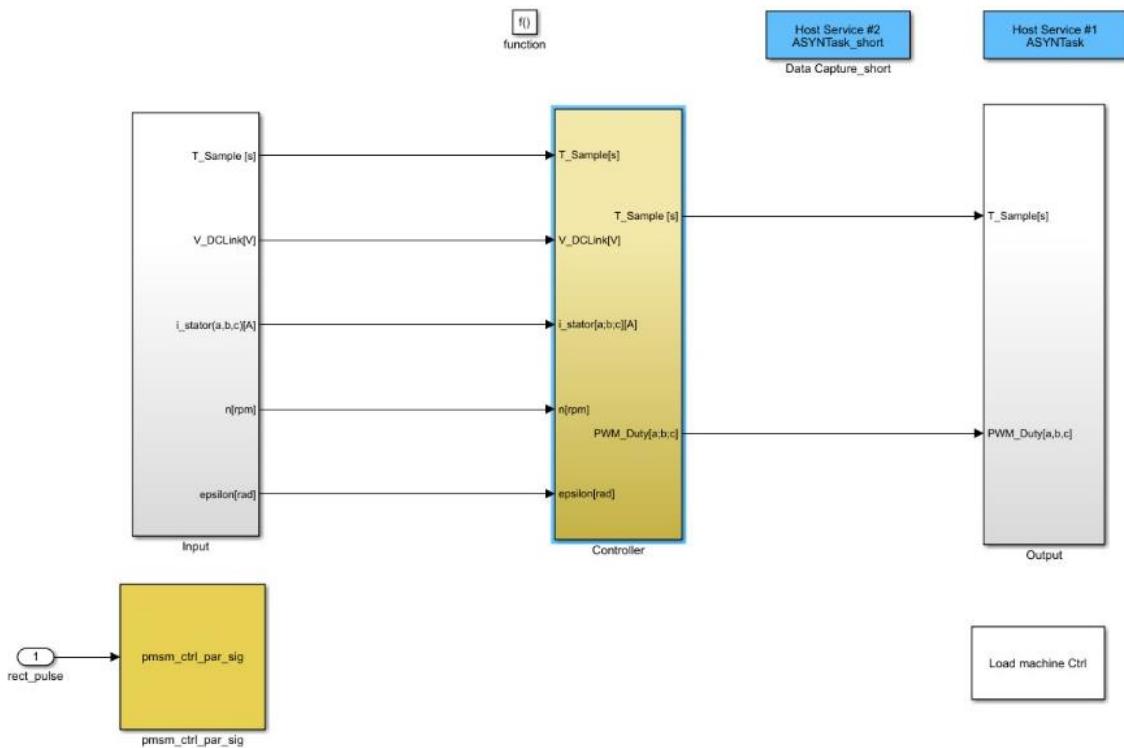
Để có thể hoạt động và điều khiển được mô hình PMSM, chúng ta sử dụng mô hình bộ điều khiển (Controller) đã được xây dựng trên Matlab/Simulink cung cấp bởi dSPACE. Ở đây bộ điều khiển PMSM được xây dựng theo phương pháp điều khiển Field-Oriented Control (FOC).



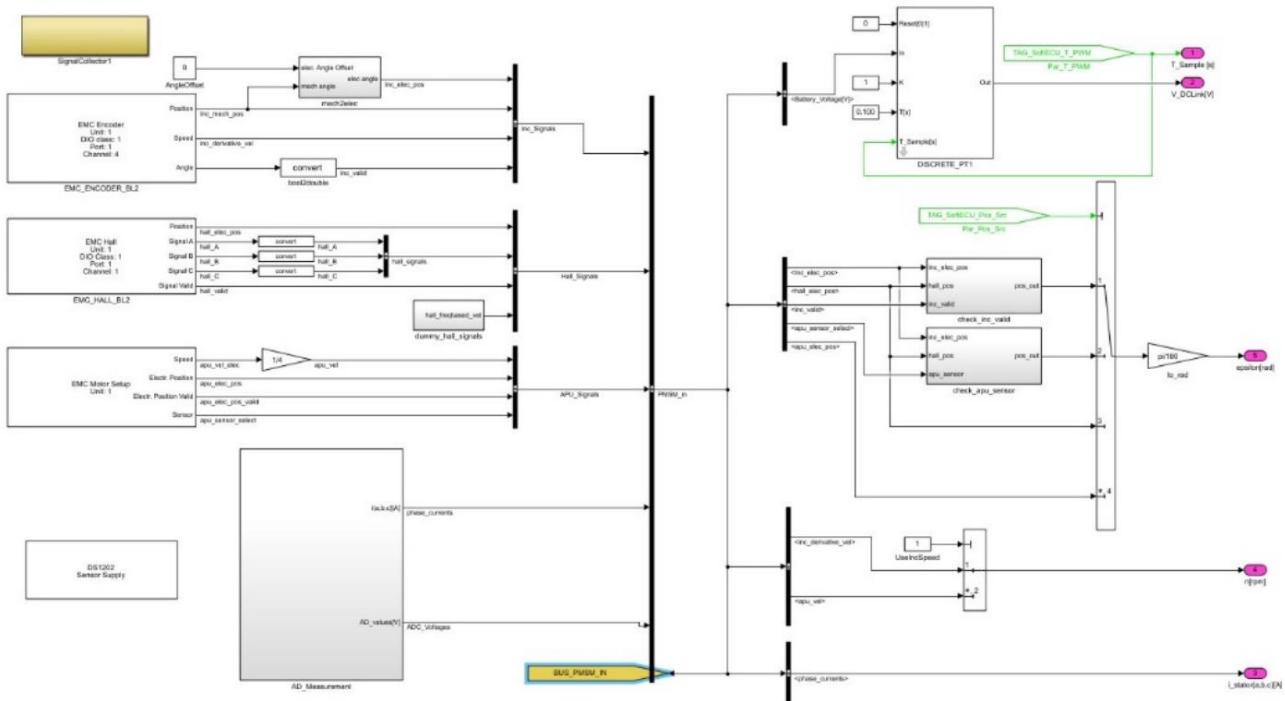
Hình 4.18. Mô hình PMSM Controller trên Matlab/Simulink

Các mô hình điều khiển chính bên trong khối Control sẽ được xử lí khi nhận được một tín hiệu ngắn từ chân tín hiệu ngắn *DIO CL1 HWINT Port 1 Channel 7*:

- Input: nhận các tín hiệu đầu vào, các thông số thiết lập motor.
- Controller: chứa các thuật toán xử lí tín hiệu, các phương pháp tính toán, điều khiển bằng PID, FOC.
- Output: đưa ra Duty cylce 3 pha của PMSM.



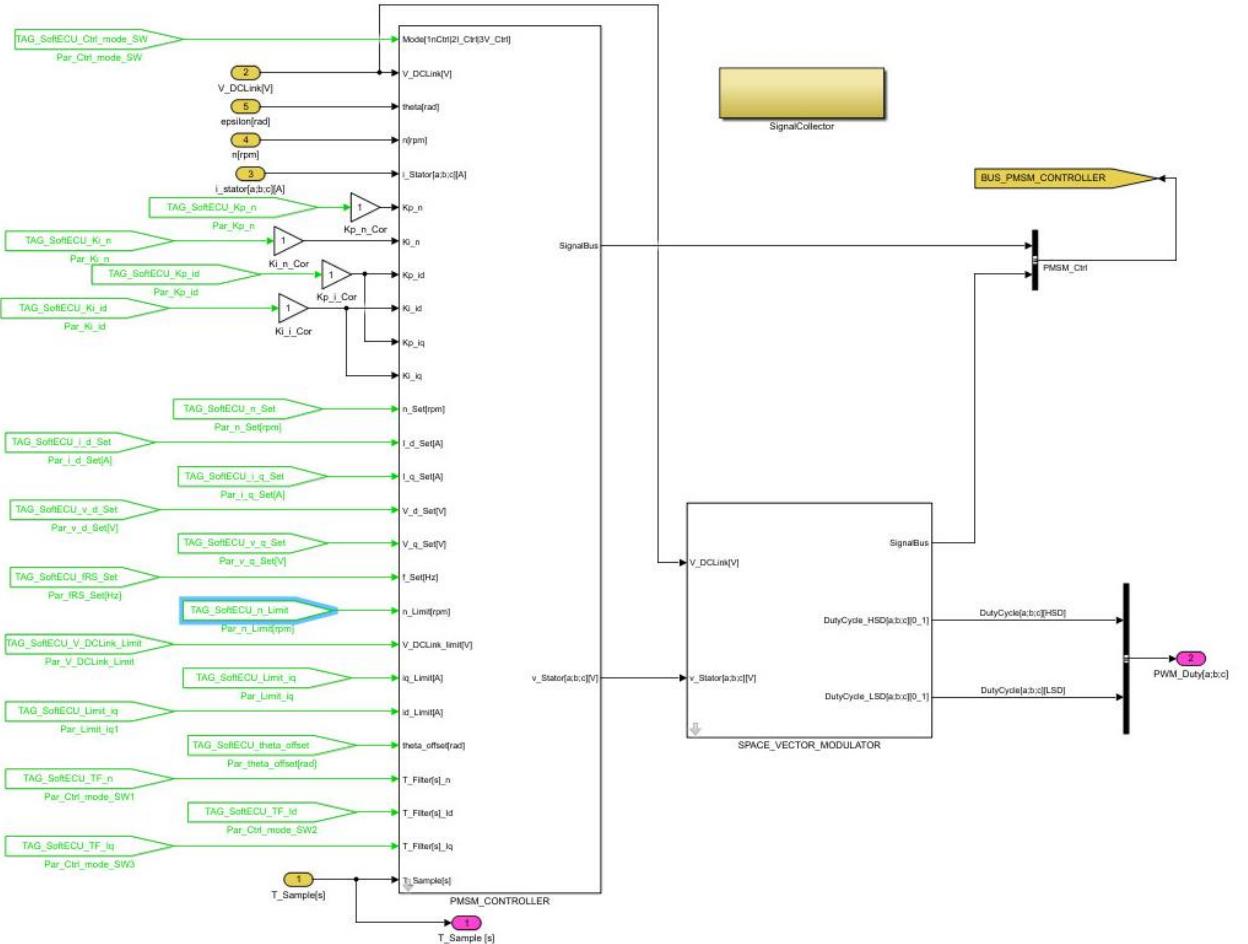
Hình 4.19. Các khối xử lí bên trong Control



Hình 4.20. Khối xử lí tín hiệu đầu vào Input

Ở khối đầu vào (Input) sẽ bao gồm việc xử lí các tín hiệu đầu vào từ các cảm biến Hall, cảm biến Encoder được sử dụng, các tín hiệu các chân điện áp ADC, ... Các tín hiệu

này sẽ được xử lý bước đầu trong khối Input và sau đó sẽ được các tín hiệu cần thiết như $i_{\text{stator}}(a,b,c)[A]$, $n[\text{rpm}]$, $\epsilon[\text{rad}]$ đưa vào khối điều khiển (Controller).



Hình 4.21. Khối xử lí tín hiệu Controller

Bên trong khối Controller sẽ xử lí các tín hiệu được đưa vào từ Input, thêm vào đó Controller sẽ nhận thêm các tín hiệu set (tín hiệu đặt bởi người dùng) như n_{Set} , $i_{\text{q}}_{\text{Set}}$, $i_{\text{d}}_{\text{Set}}$, $v_{\text{d}}_{\text{Set}}$, n_{Limit} , ... Từ việc điều chỉnh các tín hiệu đó, người dùng có thể điều chỉnh PMSM hoạt động theo ý muốn.

Các thông số đầu vào của khối được lấy từ một file khởi tạo các giá trị ban đầu.

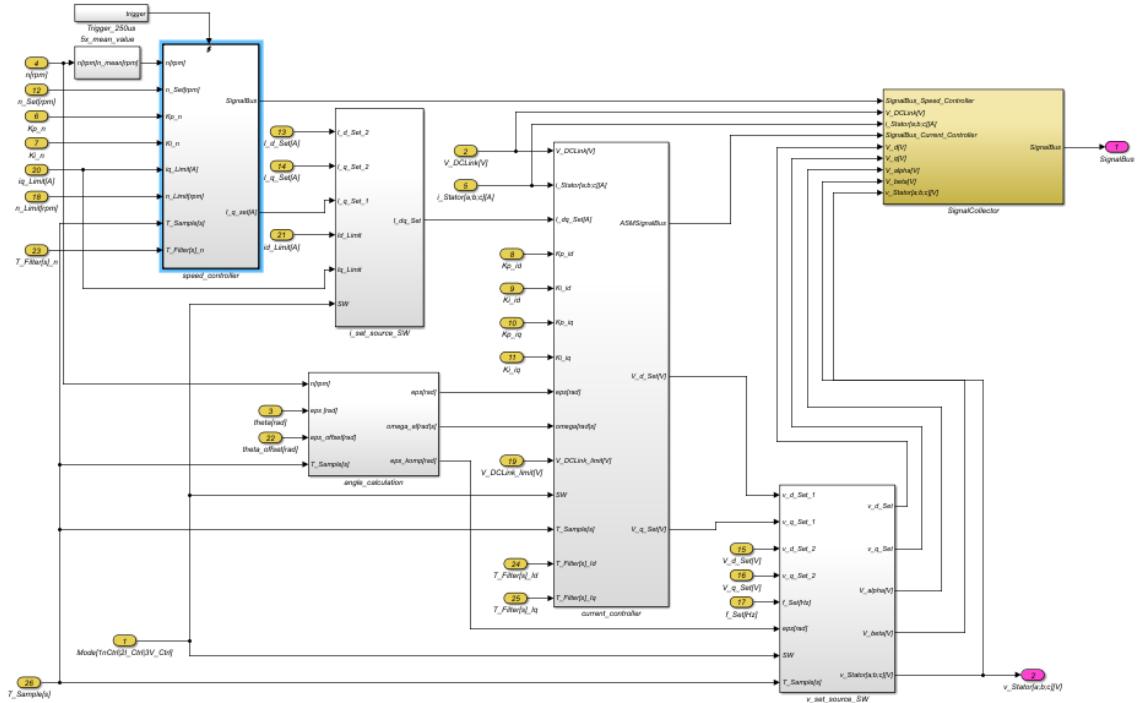
```

PMSM_Demo_ini.m × +
1 %rtiver rti1005;
2 %addpath(strcat(pwd, '\ASM_ElectricComponents_lib_DEV\MATLAB\ASM\Libraries'));
3 %addpath(strcat(pwd, '\ASM_ElectricComponents_lib_DEV\MATLAB\ASM\MFcns'));
4 %addpath(strcat(pwd, '\ASM_ElectricComponents_lib_DEV\MATLAB\Local'));
5
6 % Clear Workspace
7 %clear all;
8
9 - SIM.StepSize.v = 5e-6; % Step Size for plant simulation
10 - T_PWM = 50e-6; % PWM periode = Controller step size
11 - SIM.ControllerStepSize.v = 50e-6;
12 - SIM.MDLStepSize.v = 5e-6;
13 - SIM.StepSize.v = SIM.MDLStepSize.v;
14 %=====
15 % Set your Machine parameters in the following area!
16 %=====
17
18 - L = 0.4e-3; %Stator inductance
19 - R = 0.2; %Stator resistance
20 - Psi = 0.017; %Flux induced by magnet Psi
21 - PolePairs = 4.0; %PolePairs
22 - Km = 1.5*Psi*PolePairs; %1.5*Psi*Pp
23 - UD = 15.0; %DC link voltage
24 - IQ_MAX = 10.6; %maximal value of i_q (ADAPT !!)
25 - N_MAX = 3000.0; %max. mechanical velocity (ADAPT !!)
26 - ID_MAX = 5.6; %maximal value of i_d (maxFieldWeaking Current)
27
28 %=====
29 % Controller settings
30 %=====

```

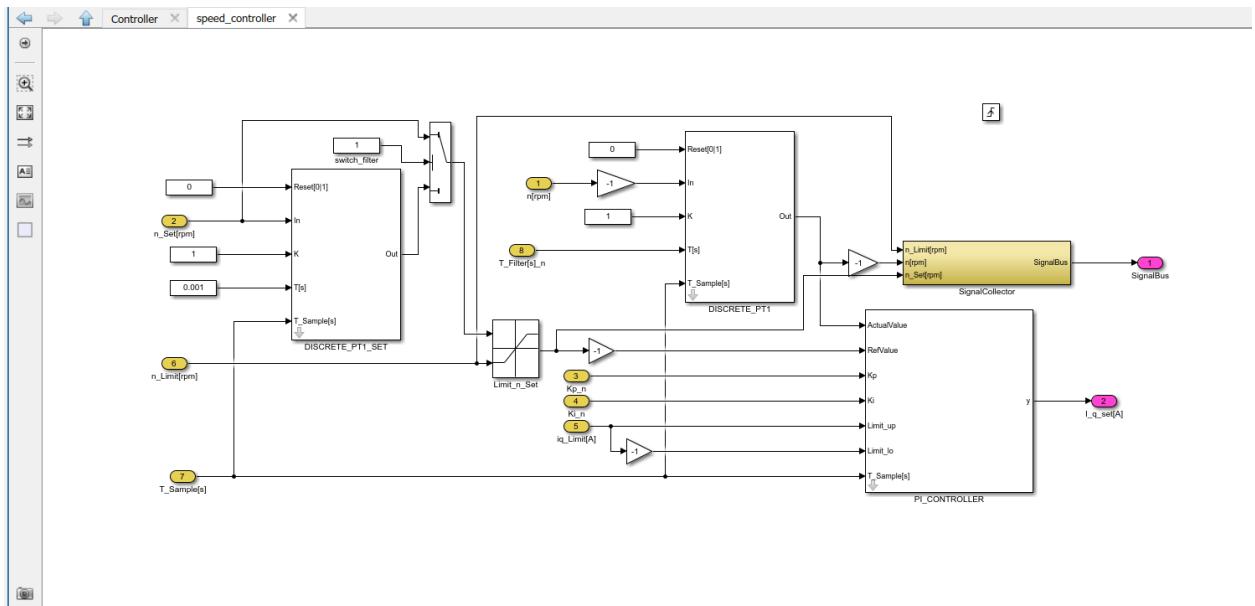
Hình 4.22. File init đuôi .m chứa các thông số đầu vào ban đầu

Với thuật toán này, động cơ PMSM được điều khiển ở 2 trường hợp: điều khiển tốc độ (speed_control) và điều khiển dòng điện (current_control).



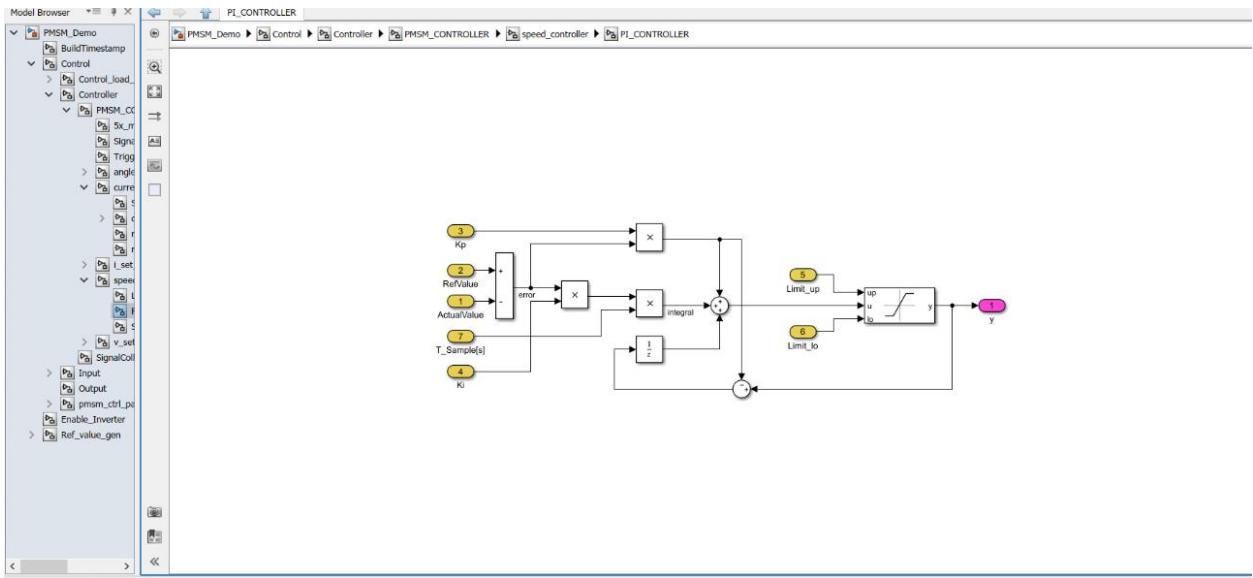
Hình 4.23. PMSM Control

- Điều khiển tốc độ



Hình 4.24. Khối Speed_Controller

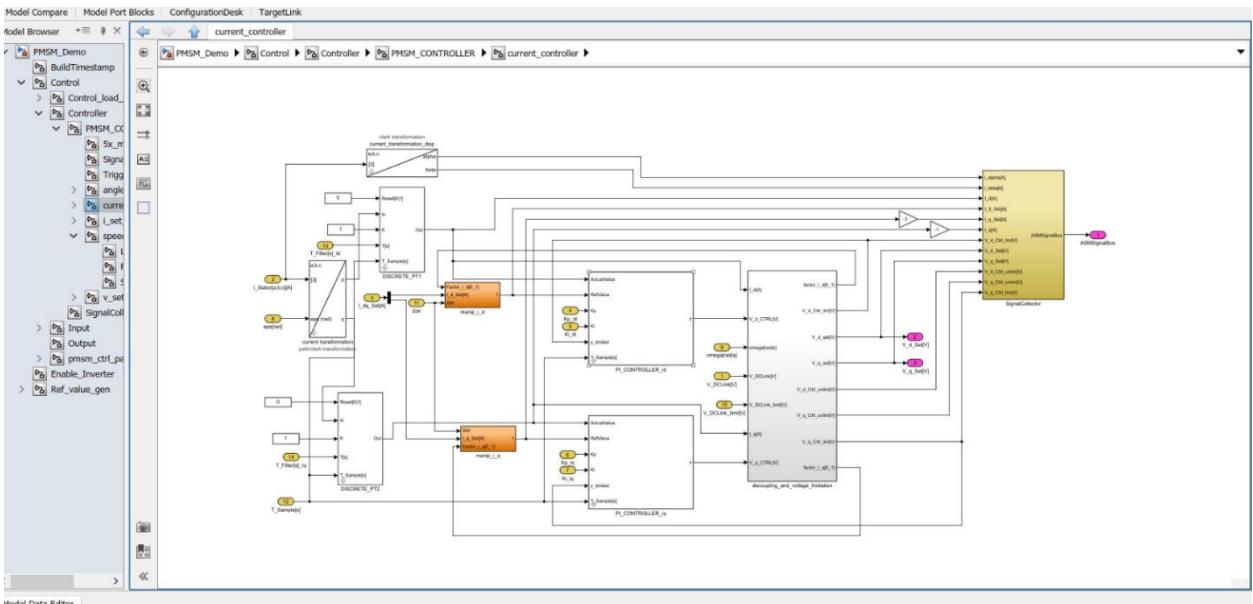
Các giá trị tốc độ mong muốn và thực tế sau khi được xử lý phù hợp với yêu cầu của bài toán, được đưa vào PI Control để tính toán và đưa ra giá trị I_q_Set phù hợp với động cơ.



Hình 4.25. PI Control của điều khiển tốc độ

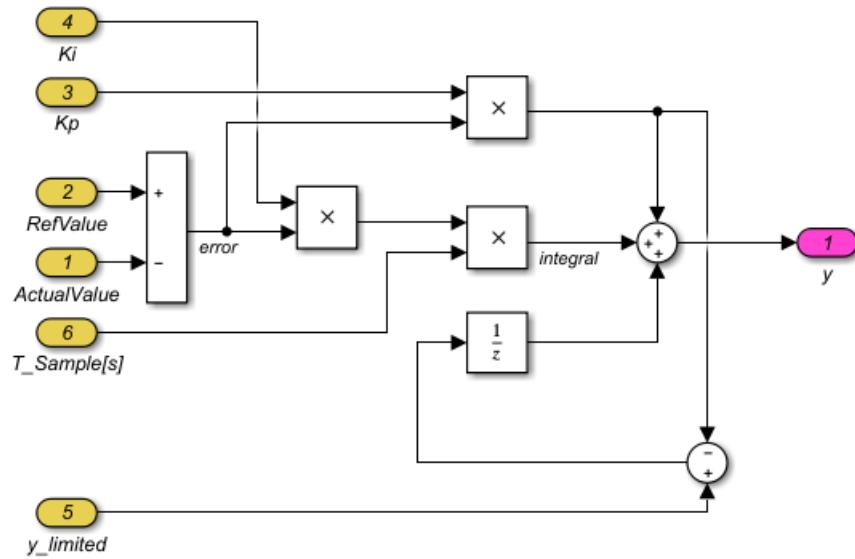
Các giá trị Kp, Ki có thể được thay đổi, tùy chỉnh để có thể đáp ứng khả năng điều khiển tốc độ một cách tốt nhất.

- Điều khiển dòng điện

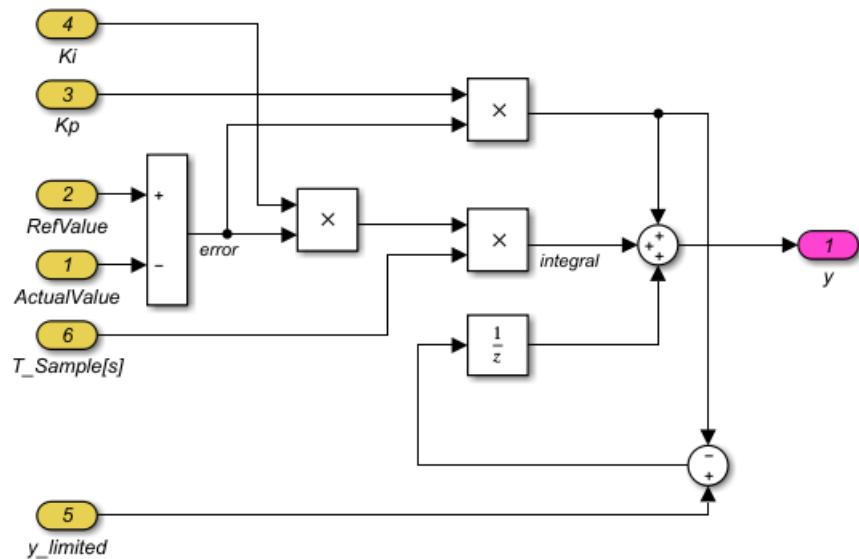


Hình 4.26. Khối Current_controller

Các giá trị i_d và i_q sẽ được bộ điều khiển xử lý và điều khiển phù hợp với yêu cầu của bài toán, đưa vào PI Control để đưa ra các giá V_d và V_q phù hợp.

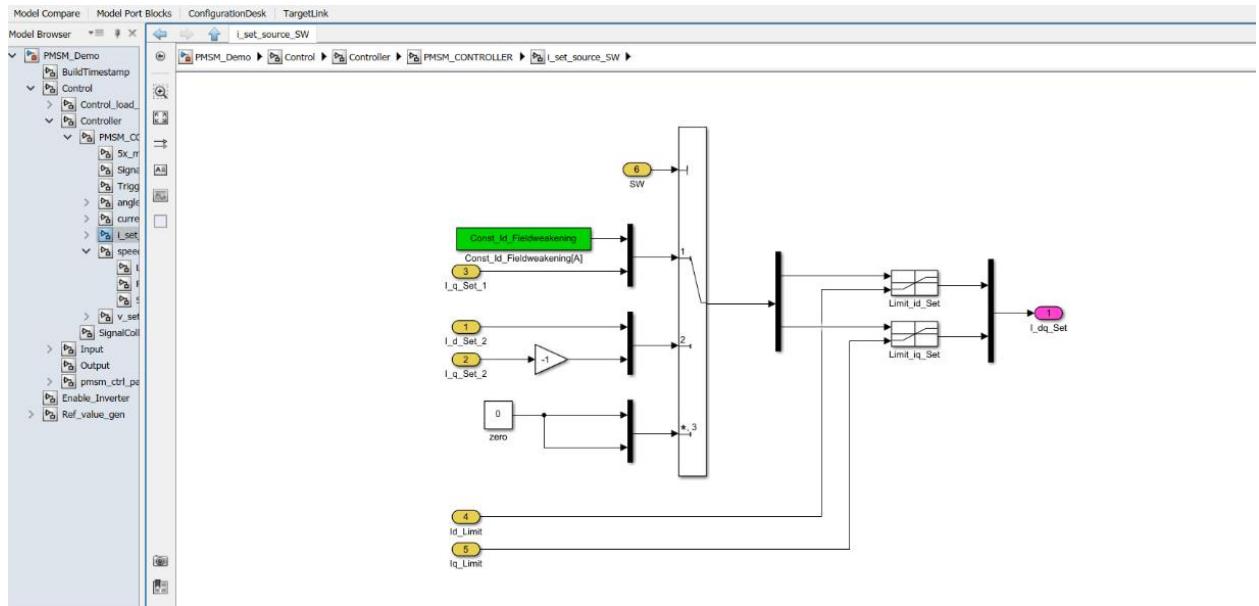


Hình 4.27. Điều khiển i_d bằng PI Control

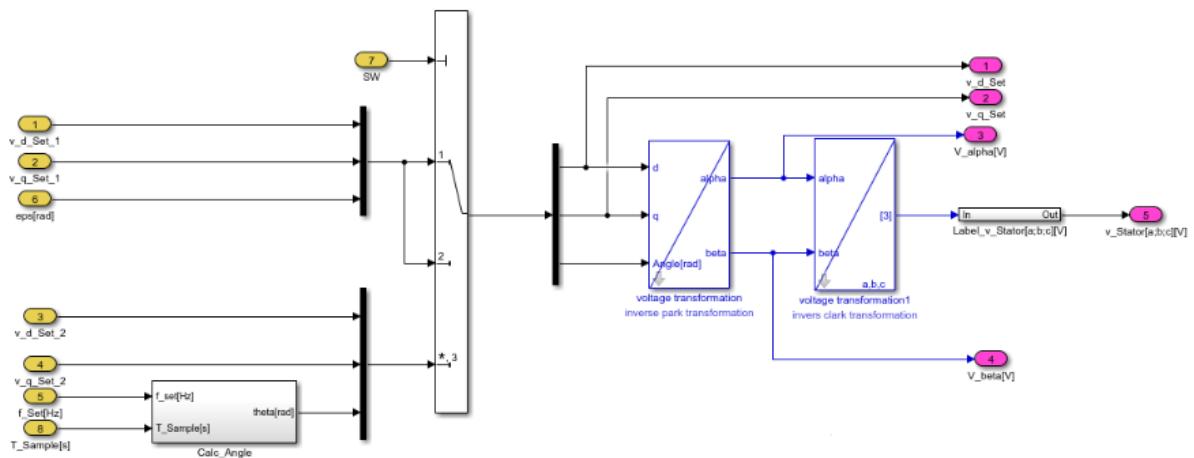


Hình 4.28. Điều khiển i_q bằng PI Control

Ngoài ra, trong khói PMSM Controller còn có các khói `i_set_source_SW` và `v_set_source_SW` dùng để thiết lập các giá trị i và v của động cơ tùy thuộc vào từng trường hợp điều khiển.

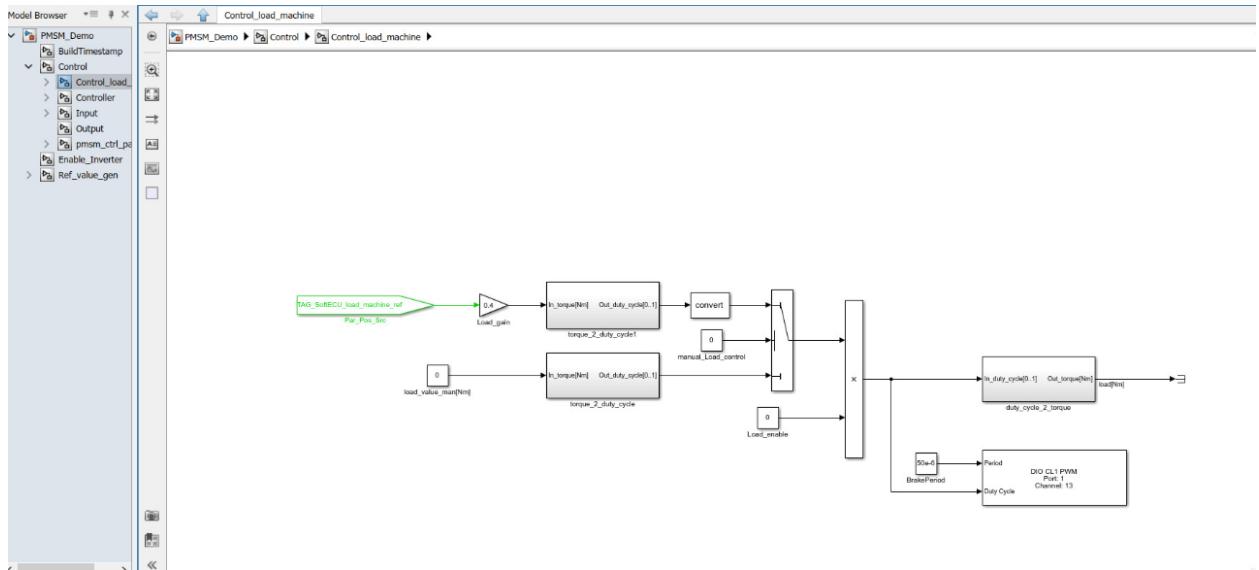


Hình 4.29. Thuật toán điều chỉnh $I_{set_source_SW}$



Hình 4.30. Thuật toán điều chỉnh $V_{set_source_SW}$

Ở đây có 1 thông số SW, đây là thông số chọn chế độ điều khiển cho động cơ. SW có giá trị 1 tương ứng với điều khiển tốc độ, giá trị 2 tương ứng với điều khiển dòng điện và 3 tương ứng với điều khiển vòng lặp hở. Với mỗi chế độ được chọn, các thông số điều khiển liên quan đến chế độ đó sẽ được đưa vào động cơ để tiến hành điều khiển động cơ

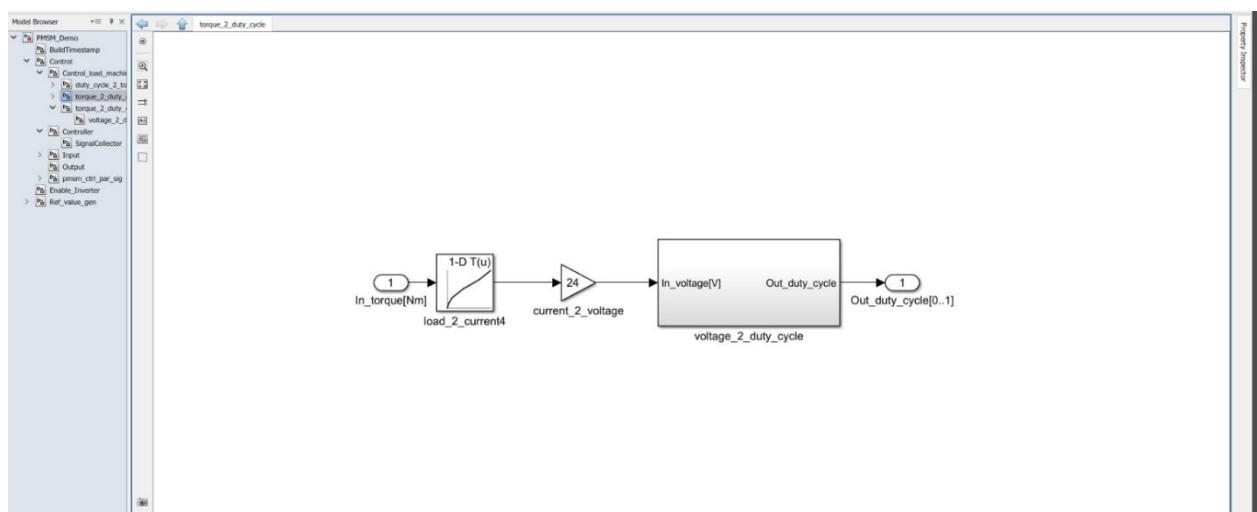


Hình 4.31. Thuật toán điều chỉnh tải

Hình 4.26 thể hiện điều chỉnh các giá trị tải, trong đó ta có thể tiến hành điều chỉnh biến Load_enable thành 0 hoặc 1 tương ứng với việc Bật/Tắt chế độ tải.

Ở block manual_Load_control có thể được điều chỉnh để chọn các mode điều khiển khác nhau: giá trị 0 ứng với period mode và khác 0 với manual mode.

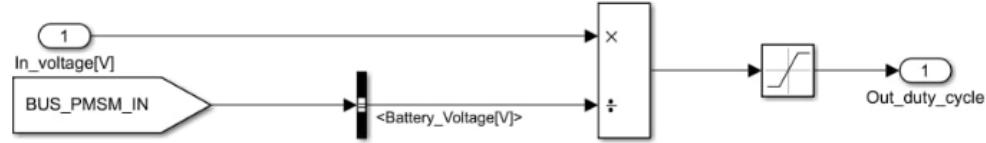
Ở chế độ period thì tải sẽ được điều chỉnh ở Load_gain = 0.4 theo từng chu kỳ, còn ở manual mode thì tải sẽ được thêm trực tiếp liên tục vào động cơ và ta có thể điều chỉnh các thông số load_value_man để thay đổi giá trị tải khi ở chế độ manual mode.



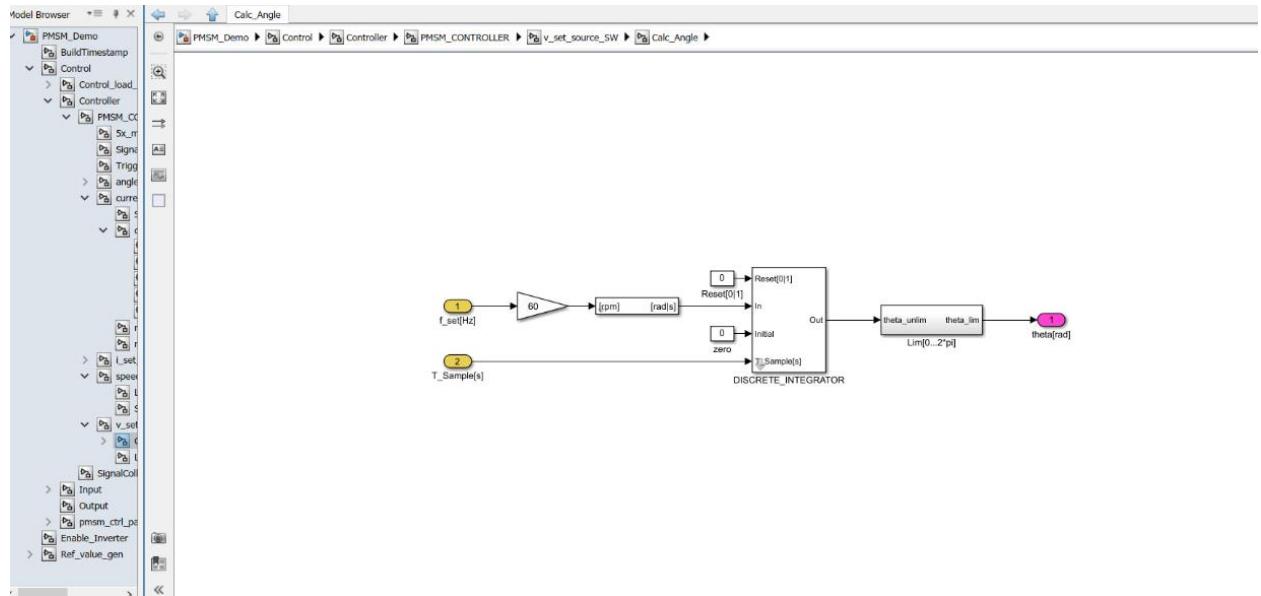
Hình 4.32. Khối torque_2_duty_cycle

Khối torque_2_duty_cycle là khối chuyển đổi từ Input_torque trở thành duty_cycle ở đầu ra output. Trong khối sẽ có look-up table để đổi từ tải sang giá trị dòng điện, tiếp tục

chuyển đổi từ giá trị dòng điện sang điện áp và cuối cùng là khối chuyển đổi từ điện áp trở thành duty_cycle.

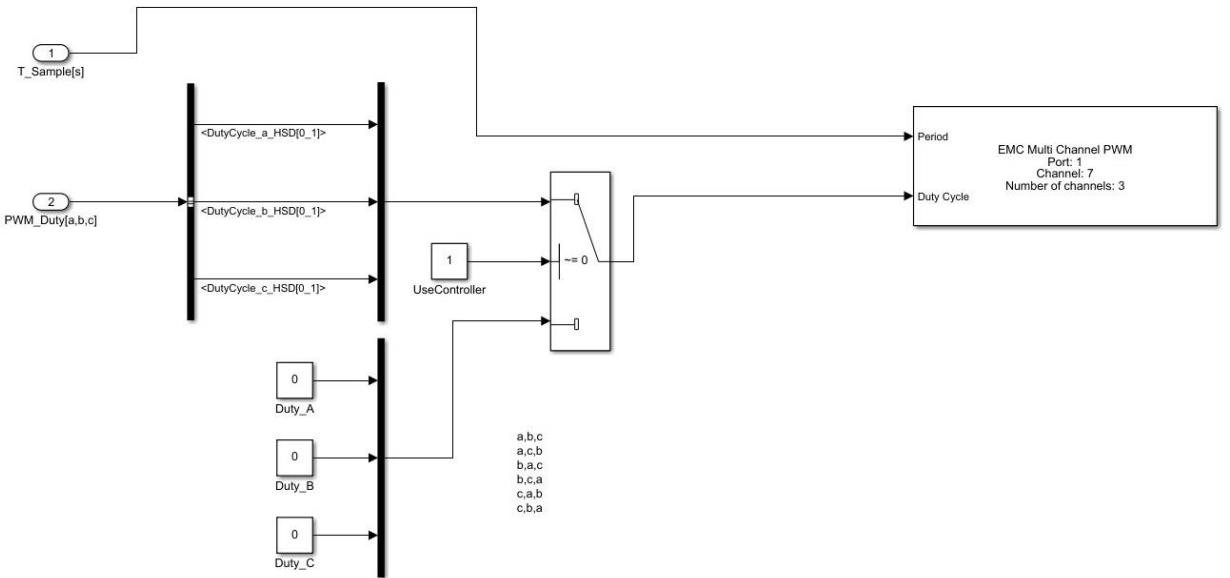


Hình 4.33. Bên trong khối voltage_2_duty_cycle



Hình 4.34. Thuật toán tính toán vị trí

Từ các tín hiệu nhận được, khối Controller tiến hành thực hiện các thuật toán PID, FOC để xử lý tín hiệu, đưa ra các duty cycle để PMSM hoạt động ở khối Output.

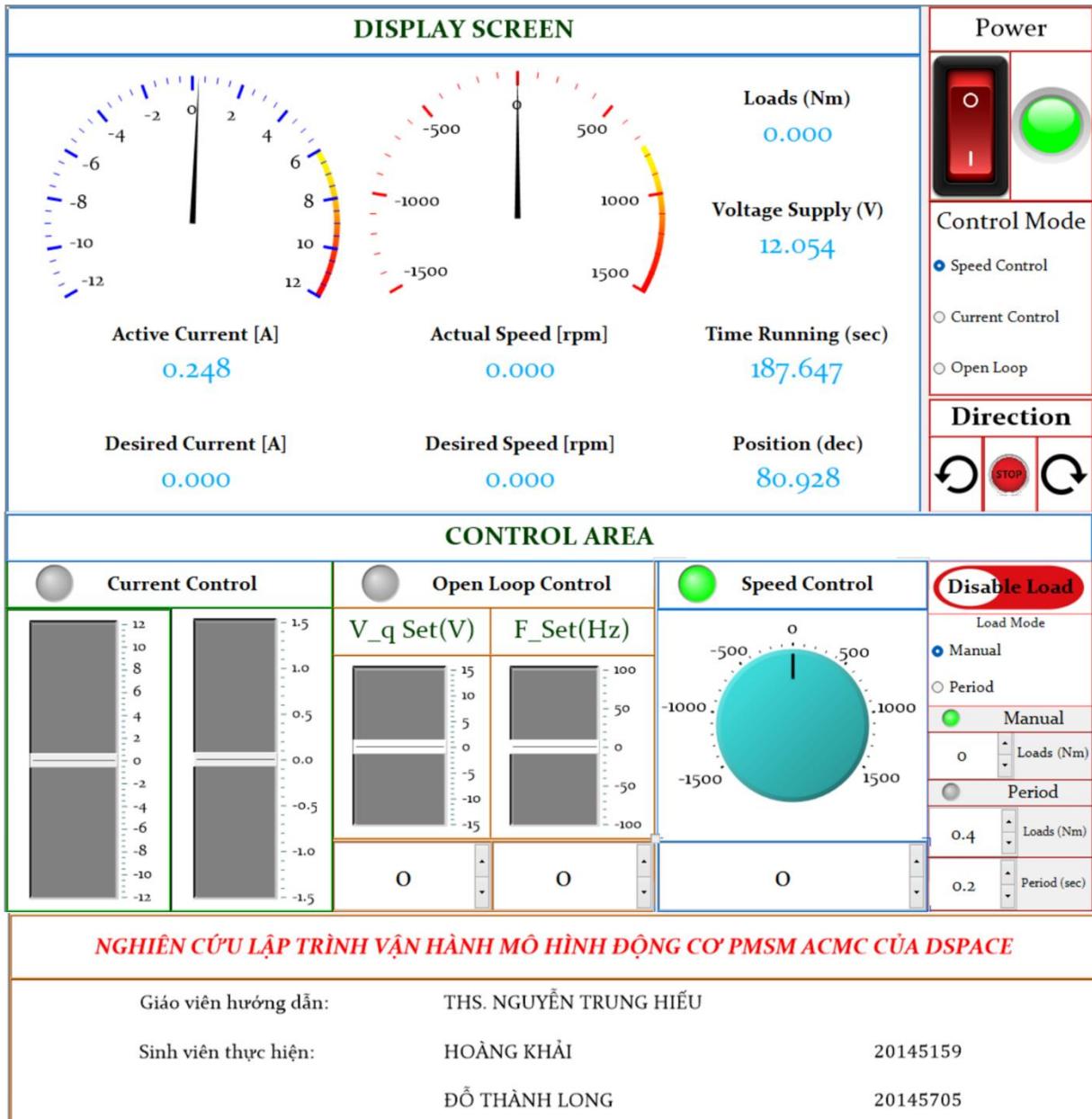


Hình 4.35. Khối tín hiệu đầu ra PMSM Output

Ở khối Output sẽ có 2 chế độ điều khiển để điều chỉnh DutyCycle, chế độ 1 là sử dụng Controller để điều chỉnh, chế độ 0 là chế độ tự điều chỉnh DutyCycle không thông qua bộ điều khiển. Sau đó các tín hiệu Duty Cycles sẽ được đưa vào 3 channels của khối EMC Multi Channel PWM cho PMSM hoạt động và Period sẽ là tín hiệu từ T_Sample[s].

Để tiến hành chạy mô hình PMSM qua Matlab/Simulink sau khi đã kết nối với MicroLabBox, ta tiến hành Build Model thông qua lệnh Ctrl + B. Sau đó hệ thống sẽ tự động tạo ra file .sdf giúp kết nối các giá trị có thể điều chỉnh giữa phần cứng MicroLabBox và các phần mềm mô phỏng điều khiển như giữa Matlab/Simulink và ControlDesk.

4.2.2. Xây dựng giao diện điều khiển trên ControlDesk



Hình 4.36. Giao diện điều khiển chung

Một số tính năng chính của GUI như: nguồn mô hình, chọn các chế độ điều khiển, điều khiển tốc độ, dòng, điện áp và tần số, chọn chế độ tải và điều khiển tải, chế độ hướng của động cơ, hiển thị các thông số và các đồ thị.

- Màn hình hiển thị: khu vực này dùng để hiển thị các thông số cơ bản như: nguồn cung cấp, thời gian chạy mô hình và thông số điều khiển của mô hình động cơ như tốc độ, dòng điện và tải trọng, đồng thời hiển thị vị trí của động cơ. Các thông số này được hiển thị bằng đồng hồ kim, hoặc đơn giản là các LED hiển thị.

- Khu vực điều khiển: chia ra làm 4 loại điều khiển chính bao gồm: điều khiển tốc độ, điều khiển dòng điện, điều khiển vòng lặp mở bằng cách thay đổi điện áp và tầm số của động cơ và cuối cùng là điều khiển tải trong tác dụng lên động cơ. Các thông số này có thể điều chỉnh được bằng nút vặn, thanh trượt hoặc đơn giản hơn là nhập trực tiếp thông số vào.
- Khu vực khác: ngoài 2 khu vực chính, còn có khu vực nút nguồn để bật/tắt nguồn cấp cho động cơ, khu vực chọn chế độ điều khiển, và khu vực điều khiển tốc độ động cơ cùng/ngược chiều, hoặc dùng theo một tốc độ cố định.
- Đồ thị: ControlDesk cũng hỗ trợ vẽ đồ thị các thông số mong muốn với độ chính xác cao. Một số đồ thị có thể vẽ để đánh giá hiệu suất của động cơ như: đồ thị vận tốc, đồ thị dòng, đồ thị điện áp stator 3 pha, đồ thị dòng điện 3 pha, đồ thị sai số, đồ thị vị trí động cơ,...

4.2.3. Chạy thực nghiệm trên mô hình và đánh giá kết quả

Sau khi hoàn tất các khâu chuẩn bị cho mô hình: kết nối các thành phần với nhau, cấp nguồn cho các bộ phận. Tiếp đó, tiến hành kiểm tra các tín hiệu LED báo lỗi trên MicroLabBox và RapidPro Power Unit. Nếu không phát sinh lỗi gì, tiến hành vận hành mô hình theo từng chế độ hoạt động.

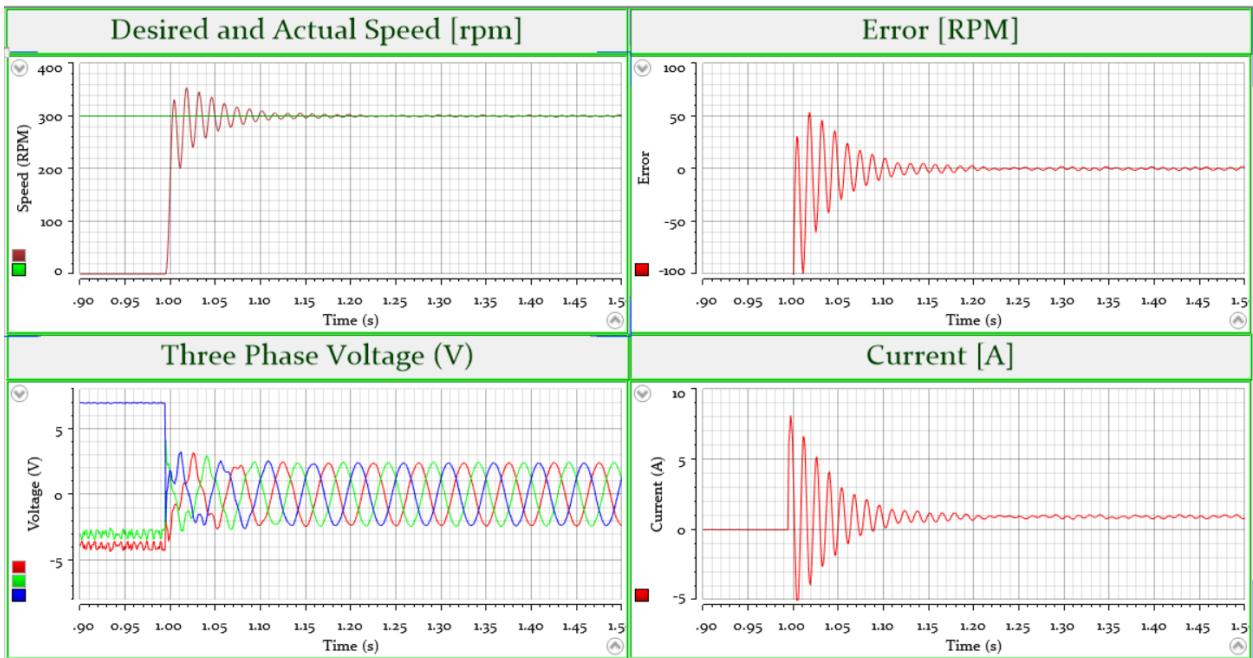
4.2.3.1. Chế độ điều khiển tốc độ

Giá trị biến điều khiển: PMSM_Ctrl_n_Set

Chọn MODE: Speed Control trên bảng chọn Control Mode

4.2.3.1.1. Thực nghiệm cho PMSM khởi động với tốc độ 300 rpm

Điều chỉnh bảng điều khiển tốc độ được xây dựng trên ControlDesk: Chọn Desire Speed 300 rpm



Hình 4.37. Đồ thị điều khiển tốc độ 300 rpm

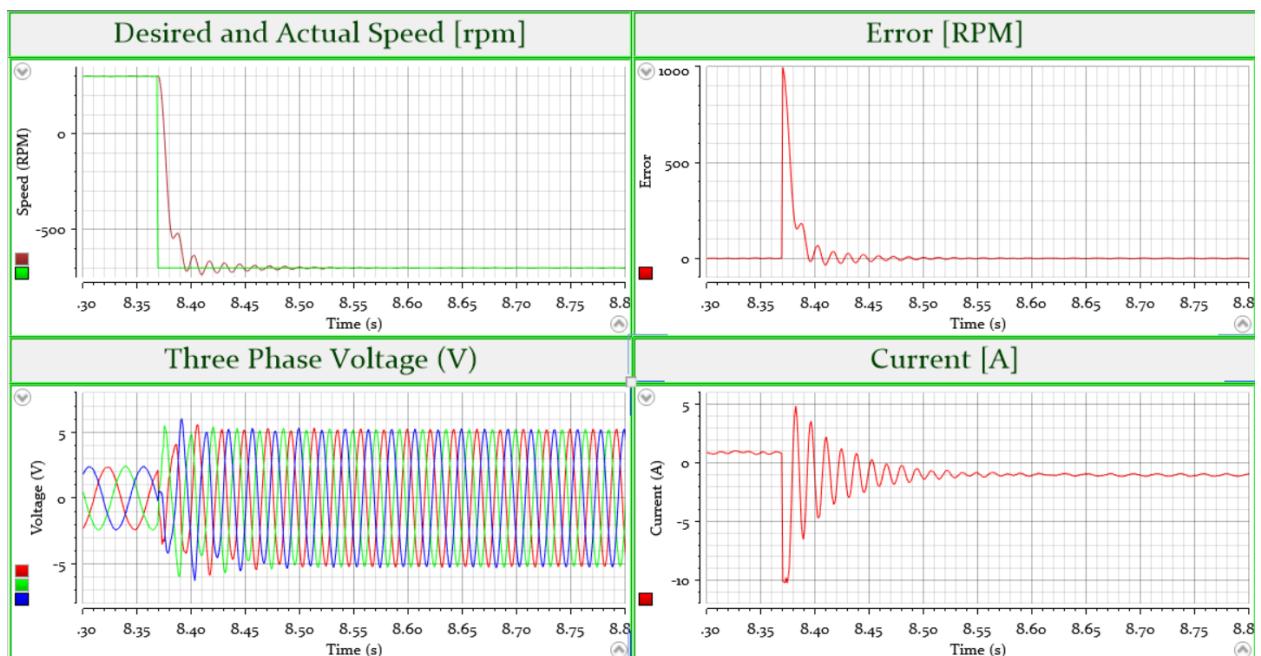
Quan sát tốc độ thực tế của động cơ hoạt động và tốc độ Actual Speed hiển thị kết hợp đồ thị Desired/Actual Speed:

- Đường màu xanh là Desired Speed, màu đỏ là Actual Speed.
- Ở thời điểm điều chỉnh gần 1s tiến hành điều chỉnh Desired Speed thành 300 rpm, hệ thống phản ứng với tốc độ Rise Time rất nhanh, khoảng 0.01s, có hiện tượng

Overshoot sau đó giảm dần và dao động xung quanh giá trị mong muốn trước khi ổn định ở 300 rpm.

- Ở đồ thị Error biểu diễn sai số giữa Desired và Actual Speed. Tại thời điểm bắt đầu thay đổi tốc độ, sai số tăng đột ngột, có khi đạt giá trị cao nhất khoảng 90 rpm. Sau đó, sai số giảm dần và tiệm cận về 0 khi hệ thống hoạt động ổn định ở 300 rpm.
- Đồ thị “Three Phase Voltage” biểu diễn điện áp ba pha của động cơ, ở thời điểm tăng tốc, điện áp ở ba pha bắt đầu tăng và ổn định ở mức $\pm 2.5V$ tương ứng với mức 300 rpm.
- Còn ở đồ thị Current, tại thời điểm thay đổi tốc độ, dòng điện tăng đột ngột để đáp ứng, cao nhất có thời điểm 7A, sau đó giảm dần và duy trì ổn định ở mức 1A khi tốc độ ổn định.

4.2.3.1.2. Giảm tốc độ động cơ từ 300 rpm đến -700 rpm



Hình 4.38. Giảm tốc độ động cơ từ 300rpm đến -700 rpm

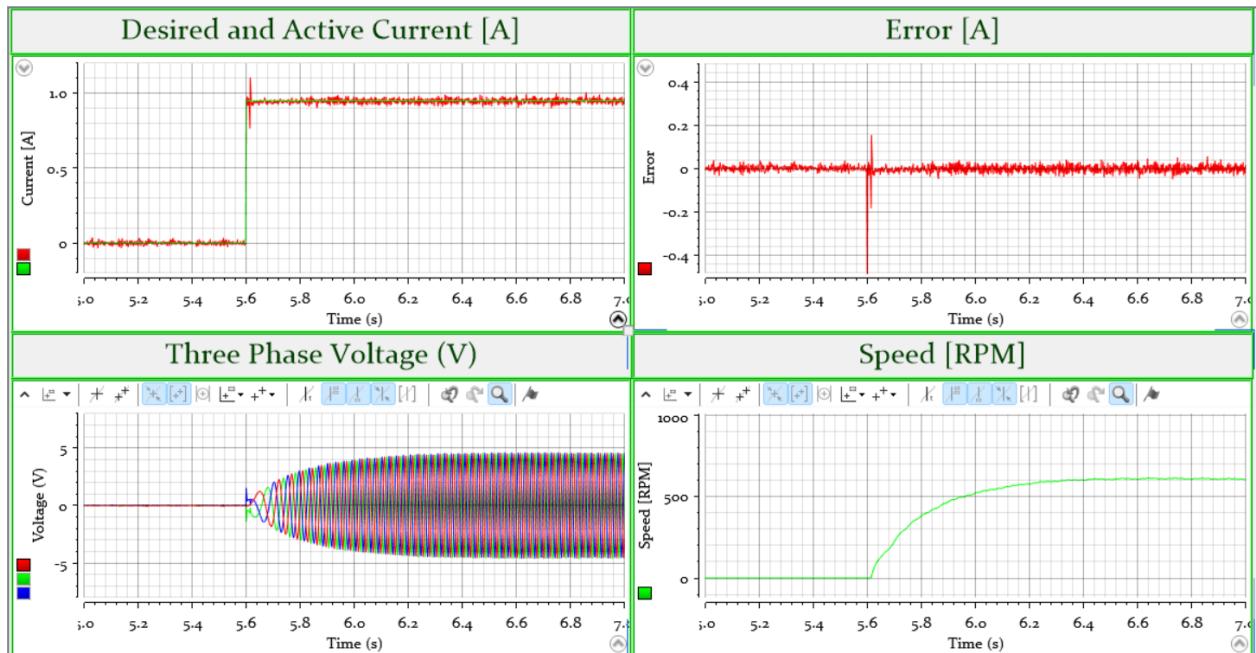
Ta tiến hành cho giảm tốc độ động cơ đang hoạt động ổn định với tốc độ 300 rpm đột ngột xuống -700 rpm.

- Tại thời điểm thay đổi tốc độ cũng như chiều quay của hệ thống (300 rpm cùng chiều thành 700 rpm ngược chiều), hệ thống phản ứng với tốc độ khoảng 0.05s, sau đó quay ngược chiều ở 700 rpm với độ ổn định.

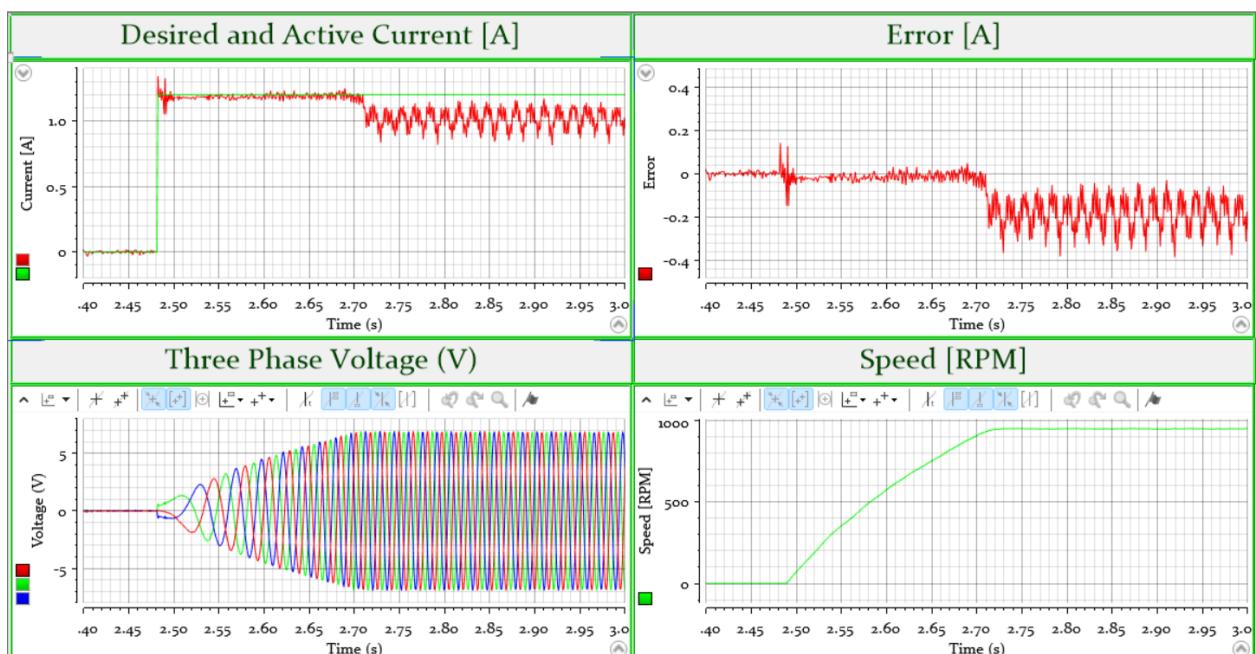
- Các đồ thị Error, Current, Three Phase Voltage cũng có sự thay đổi lớn lúc thay đổi giá trị điều khiển, nhưng nhanh chóng ổn định với tốc độ cho trước

4.2.3.2. Chế độ điều khiển dòng

Giá trị biến điều khiển: PMSM_Ctrl_i_q_Set



Hình 4.39. Chế độ điều khiển bằng dòng 0.95A



Hình 4.40. Chế độ điều khiển dòng 1.2A

Tiến hành điều chỉnh dòng điện của động cơ ở các mức 0.95 và 1.2:

- Khi ta thay đổi dòng điện điều khiển thì dòng điện trong hệ thống thay đổi một cách gần như chính xác và lập tức (0.05s) đáp ứng nhu cầu điều khiển, cùng với đó tốc độ quay của động cơ cũng như điện áp 3 pha duy trì ổn định sau khi dòng điện ổn định
- Ở đồ thị điều khiển bằng dòng 1.2A, ở giai đoạn đầu hệ thống dùng dòng 1.2A để tăng tốc độ, nhưng khi đạt đến ngưỡng tốc độ lớn nhất của hệ thống (~950 rpm) thì dòng điện không cần đạt mức cao như 1.2A để duy trì. Khi đó, dòng điện sẽ được duy trì ở mức có thể duy trì tốc độ tối đa (ở mức dòng điện 0.9 → 1.1).

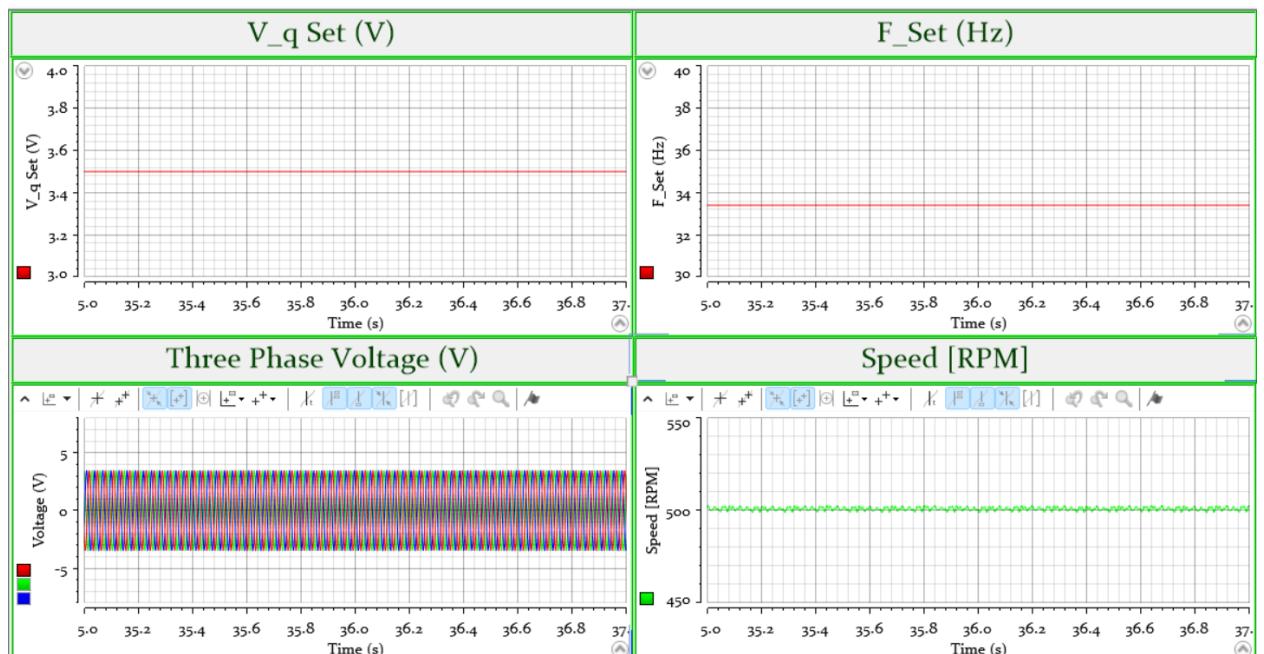
4.2.3.3. Chế độ điều khiển vòng lặp mở

Chọn MODE: điều khiển open loop trên bảng chọn chế độ điều khiển.

Giá trị biến điều khiển: PMSM_Ctrl_v_q_Set và PMSM_Ctrl_fRS_Set.

4.2.3.3.1. Điều chỉnh cho động cơ vận hành ổn định mức 500 RPM

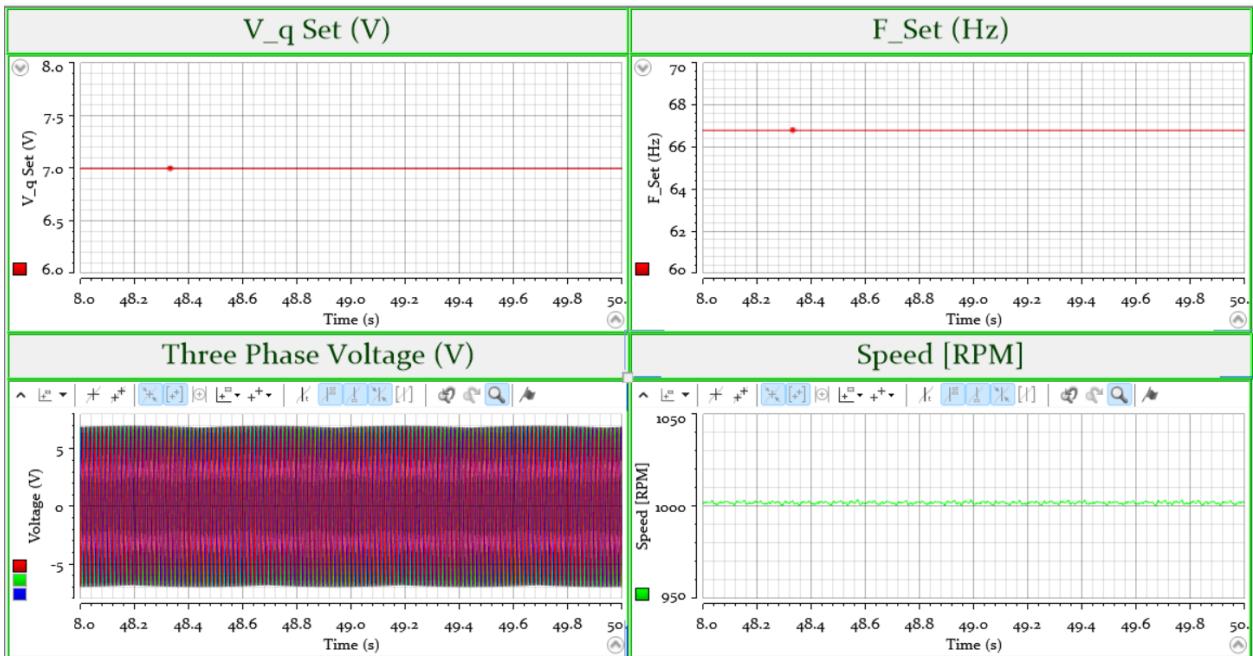
Tiến hành điều chỉnh hai thông số của điện áp và tần số của mô hình điều khiển sao cho động cơ hoạt động êm ái, ổn định ở mức 500 RPM



Hình 4.41. Chế độ Open Loop hoạt động ổn định 500rpm

4.2.3.3.2. Điều chỉnh cho động cơ vận hành ổn định mức 1000 RPM

Tiến hành điều chỉnh hai thông số của điện áp và tần số của mô hình điều khiển sao cho động cơ hoạt động êm ái, ổn định ở mức 1000 RPM



Hình 4.42. Chế độ Open Loop hoạt động ổn định 1000rpm

Nhận xét:

- Để động cơ hoạt động êm ái ở từng chế độ, ta cần điều chỉnh thông số điện áp và tần số một cách hợp lý.
- Động cơ hoạt động ở điện áp và tần số không hợp lý sẽ dẫn đến động cơ bị giật, kẹt, tạo ra tiếng ồn.
- Tuy nhiên, chế độ vòng lặp mở (Open Load) thường làm giảm sự ổn định của hệ thống do không có giá trị trả về cho bộ Controller xử lí, từ đó dẫn đến không điều khiển được Error gây ra sự dao động với tần suất lớn.
- Điều khiển bằng vòng lặp mở thường không đem lại mức độ hiệu quả cao, thích hợp với những ứng dụng yêu cầu độ chính xác tương đối.

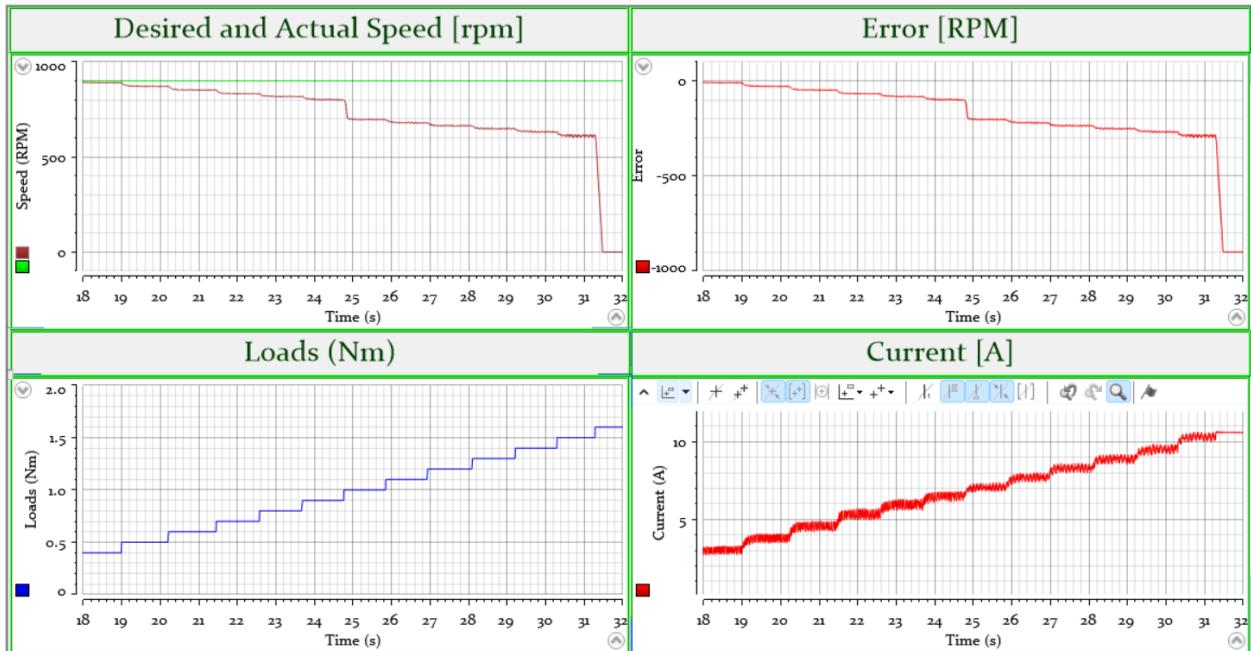
4.2.3.4. Chế độ thêm tải

Có thể áp đặt tải lên động cơ PMSM bằng 2 chế độ: manual và period. Với chế độ manual, tải được đưa xuống động cơ với một giá trị mong muốn trong xuyên suốt thời gian hoạt động của động cơ. Trong khi đó, chế độ period sẽ thay đổi chu kỳ chịu tải theo 1

khoảng thời gian nhất định. Thiết lập tần số càng nhỏ thì thời gian chịu tải càng lớn và ngược lại.

4.2.3.4.1. Manual

Ở chế độ này, ta sẽ thực hiện Enable chế độ tải (Load) và tiến hành thêm tải từ từ cho động cơ PMSM đang chạy ở tốc độ 900 RPM. Các mức thêm tải sẽ là mỗi giá trị 0.1, tiến hành từ 0 đến khi động cơ không còn đủ khả năng hoạt động.



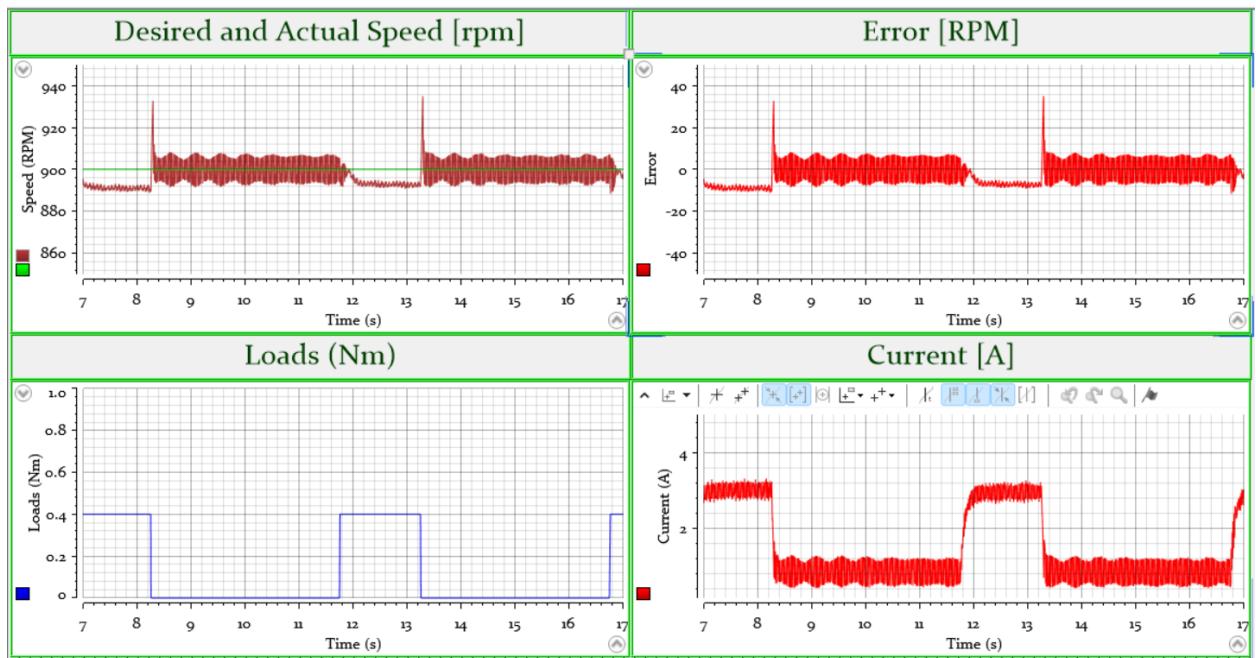
Hình 4.43. Chế độ thêm tải Manual Load

Nhận xét:

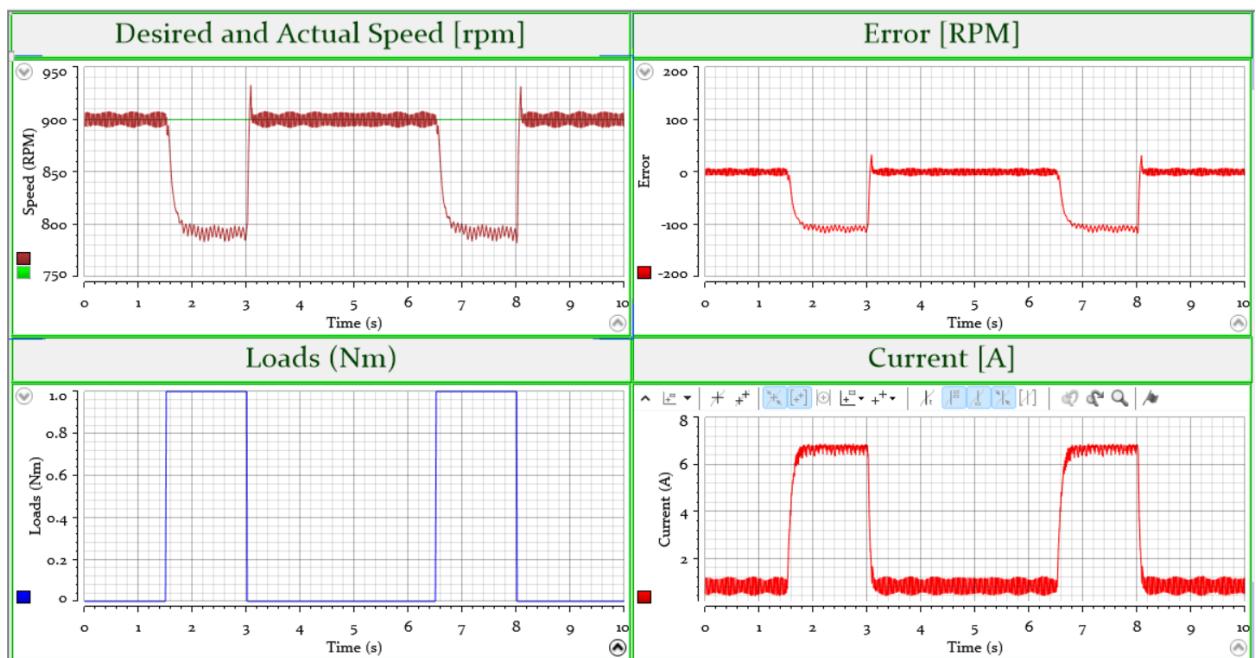
- Khi ta thực hiện tăng tải cho động cơ, hệ thống sẽ tự động tăng cường độ dòng điện (I_q) để tiến hành đảm bảo tốc độ quay của động cơ. Càng tăng tải càng cao thì cường độ dòng điện được hệ thống tăng lên càng lớn.
- Tuy nhiên, mức cường độ dòng điện hệ thống tăng lên vẫn không đủ đáp ứng duy trì tốc độ ban đầu, nó chỉ giữ cho tốc độ không bị giảm một lượng lớn và đột ngột.
- Khi mức tải chạm đến mức quá cao (>1.6) thì động cơ sẽ quá nặng không thể hoạt động được nữa.

4.2.3.4.2. Period

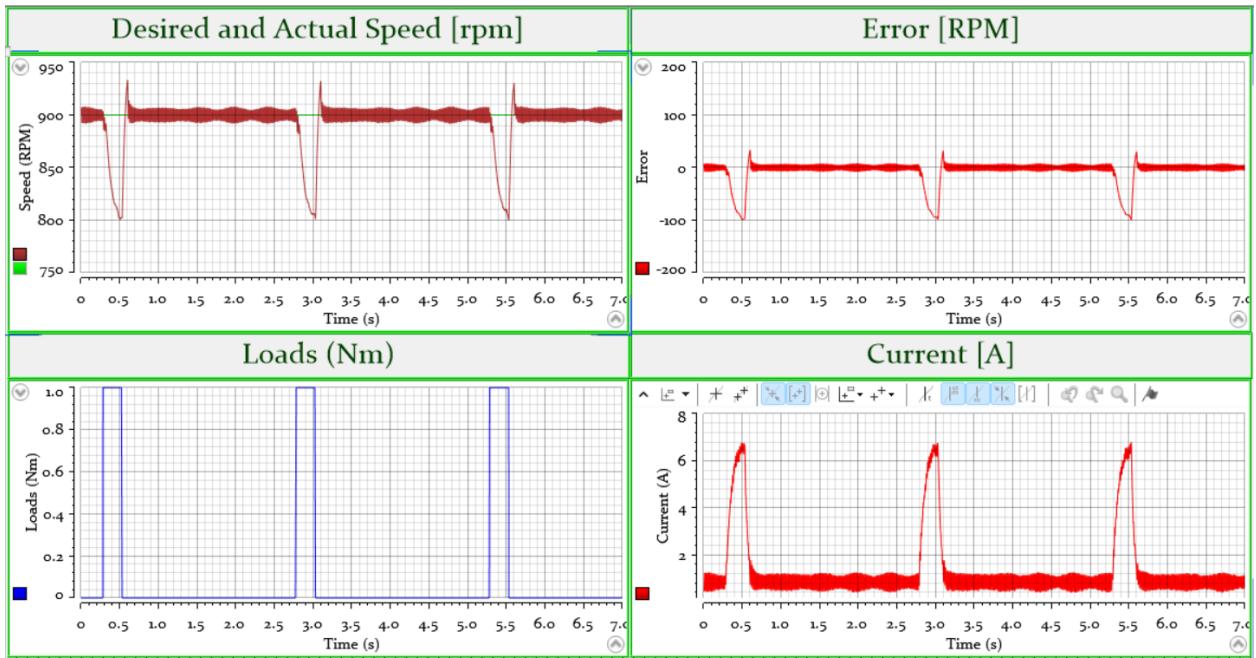
Ở chế độ này, cho động cơ chạy với tốc độ 900 RPM, sau đó kích hoạt tải 0.4 Nm với tần số 0.2 Hz và 1Nm với tần số 0.2 Hz.



Hình 4.44. Điều khiển tải với 0.4 Nm và 0.2 Hz



Hình 4.45. Điều khiển tải với 1 Nm và 0.2 Hz



Hình 4.46. Điều khiển tải với 1 Nm và 0.4 Hz

Nhận xét:

- Khi áp tải, dòng điện có xu hướng tăng lên để bù lại lượng momen tổn thất do tải gây ra, giúp di trì tốc độ.
- Đôi khi với tải trọng cao, dòng điện tăng đến cực đại nhưng vẫn không di trì được tốc độ động cơ.
- Tần số càng lớn, thời gian áp tải cho động cơ càng nhỏ.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Bằng cách tìm hiểu, nghiên cứu và ứng dụng phương pháp Model-Based Design để vận hành hệ thống điều khiển cho động cơ PMSM thông qua phần cứng điều khiển MicroLabBox của dSPACE, nhóm nghiên cứu đã đạt được những kết quả:

- Có sự hiểu biết về các loại động cơ điện, hiểu rõ cấu trúc và các thông số kỹ thuật của động cơ PMSM, đồng thời nhận thức được các ưu điểm và nhược điểm của loại động cơ này. Động cơ PMSM có hiệu suất cao, kích thước nhỏ gọn và độ bền tốt, tuy nhiên cũng đòi hỏi hệ thống điều khiển phức tạp hơn so với các loại động cơ khác.
- Hiểu rõ về phương pháp Model-Based Design trong việc xây dựng và phát triển hệ thống, những ưu điểm của nó khi so với các phương pháp truyền thống. Bằng ứng dụng Matlab/Simulink biết cách thức xây dựng các thuật toán điều khiển và kiểm chứng chúng thông qua mô phỏng trước khi triển khai trên phần cứng thực tế. Điều này giúp giảm thiểu rủi ro và tiết kiệm thời gian trong quá trình phát triển hệ thống.
- Có kiến thức về các phần cứng của dSPACE như RapidPro, MicroLabBox cũng như các phần mềm như ControlDesk, ConfigurationDesk trong việc xây dựng và phát triển các hệ thống lớn, nhỏ.
- Tiến hành thực nghiệm vận hành mô hình PMSM ACMC, cho thấy mô hình hệ thống hoạt động ổn định và đáp ứng tốt các yêu cầu điều khiển. Thông qua giao diện ControlDesk, nhóm nghiên cứu có thể giám sát và điều chỉnh hệ thống một cách dễ dàng, đồng thời thu thập dữ liệu để đánh giá hiệu suất của hệ thống. Các kết quả thực nghiệm cho thấy hệ thống điều khiển PMSM được thiết kế có khả năng điều khiển tốc độ và dòng điện chính xác, đáp ứng nhanh với các thay đổi tải và điều kiện hoạt động khác nhau. Hệ thống đã chứng minh được tính ứng dụng cao trong các lĩnh vực như ô tô, công nghiệp và các ứng dụng yêu cầu độ chính xác cao.
- Thiết kế ra một module giao tiếp I/O với MicroLabBox giúp việc sử dụng MicroLabBox trở nên dễ dàng, an toàn hơn.

5.2. Hướng phát triển trong tương lai

Từ những kết quả đạt được, nhóm nghiên cứu đề xuất một số hướng phát triển trong tương lai:

- Cải tiến giao diện điều khiển trên ControlDesk để càng thân thiện hơn với người dùng, cung cấp thêm các tính năng giám sát và điều khiển nâng cao.
- Nghiên cứu nâng cao các thuật toán điều khiển để cải thiện hiệu suất của hệ thống PMSM ACMC.
- Nghiên cứu các hướng phát triển thêm module giao tiếp I/O, cải thiện chất lượng về mặt cơ khí.
- Thủ nghiệm và triển khai xây dựng và phát triển các hệ thống khác sử dụng MicroLabBox cũng như các phần cứng, phần mềm của dSPACE.

TÀI LIỆU THAM KHẢO

- [1] MathWorks, “What are MIL, SIL, PIL, and HIL, and how do they integrate with the Model-Based Design approach?”

<https://www.mathworks.com/matlabcentral/answers/440277-what-are-mil-sil-pil-and-hil-and-how-do-they-integrate-with-the-model-based-design-approach>

- [2] aboutmotors, “Permanent Magnet Synchronous Motor”

<https://about-motors.com/motorcontrol/pmsm/>

- [3] Collimator, “Model Based Design Overview for System Development”

<https://www.collimator.ai/post/model-based-development>

- [4] DSPACE, “RapidPro System Hardware Installation Guide”

https://www.dspace.com/en/pub/home/support/documentation.cfm?helpsetid=RapidProSystemHardwareInstallationGuide&externalid=Book_ov-00000004-0000-36cc-0201-000000000003_--&Language=en-us&Release=RLS2023-A

- [5] DSPACE, “dSPACE ControlDesk Brochure 12_2021_English”

https://www.dspace.com/en/pub/home/medien/product_info/controldesk_next_generation.cfm

- [6] DSPACE, “MicroLabBox Features”

<https://www.dspace.com/en/pub/home/products/hw/microlabbox.cfm>

- [7] DSPACE, “MicroLabBox Hardware Installation and Configuration”

https://www.dspace.com/en/pub/home/support/documentation.cfm?helpsetid=MicroLabBoxHardwareInstallationConfigurationGuide&externalid=Book_b08a73e1-5a56-47d7-be29-194d79da926f_--&Language=en-us&Release=RLS2023-A

