

[Type here]

## ĐẠI HỌC BÁCH KHOA HÀ NỘI

### ĐỒ ÁN TỐT NGHIỆP

#### Thiết kế, mô phỏng và chế tạo hệ mô phỏng lái 2 bậc tự do

PHÙNG HOÀNG SƠN  
TRỊNH THẾ ĐỨC

Ngành Cơ điện tử  
Chuyên ngành Cơ điện tử

Giảng viên hướng dẫn: PGS. TS. Phạm Đức An

Chữ ký của GVHD

Khoa: Cơ điện tử  
Trường: Cơ khí

HÀ NỘI, 8/2022

[Type here]

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG CƠ KHÍ

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

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**  
*(NGÀNH CƠ ĐIỆN TỬ)*

**1. Thông tin về sinh viên:**

Họ và tên SV: **Phùng Hoàng Sơn** Lớp: **ME-NUT17**

ĐT: 0917.541.298

**Sơn**

Họ và tên SV: **Trịnh Thé Đức** Lớp: **ME-NUT17**

ĐT: 0822.251.399

**Đức**

Email : son.ph187488@sis.hust.edu.vn

Hệ đào tạo: Chính quy

Chuyên ngành: Cơ điện tử

Đồ án tốt nghiệp được thực hiện tại:

Thời gian làm ĐATN: Từ ngày / / đến / /

**2. Tên đề tài:**

Thiết kế, mô phỏng và chế tạo hệ mô phỏng lái 2 bậc tự do

*Hà Nội, ngày tháng năm  
Giáo viên hướng dẫn*

[Type here]

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG CƠ KHÍ

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

**NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP**  
*(Dành cho Giáo viên hướng dẫn)*

Tên đề tài: .....

Họ và tên SV: Nguyễn Văn A

Lớp: ...

Họ và tên SV: Nguyễn Văn B

Lớp: .....

Chuyên ngành: .....

Giáo viên hướng dẫn: .....

**NỘI DUNG NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**I. Tác phong làm việc**

.....

**II. Những kết quả đạt được**

.....

.....

.....

.....

**III. Hạn chế của đồ án**

.....

.....

.....

**IV. Kết luận**

Người hướng dẫn đề nghị cho phép sinh viên (không) được bảo vệ đề tài tốt nghiệp trước Hội đồng chấm đồ án tốt nghiệp.

**Đánh giá:** ..... điểm

Hà Nội, ngày tháng năm 20  
Giáo viên hướng dẫn  
(Ký và ghi rõ họ tên)

## NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

(Dành cho Giáo viên phản biện)

Tên đề tài: "Thiết kế, mô phỏng và chế tạo hệ mô phỏng lái 2 bậc tự do"

Họ và tên SV: Phùng Hoàng Sơn

Lớp: ME-NUT17-K63

Họ và tên SV: Trịnh Thế Đức

Lớp: ME-NUT17-K63

Chuyên ngành: Cơ điện tử (SIE)

## NỘI DUNG NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

### I. Những kết quả đạt được

Sinh viên đã hoàn thành nội dung đồ án cùng với 01 mô hình mô phỏng lái 2 bậc tự do, các nội dung đã thực hiện bao gồm:

- Xây dựng mô hình 3D của hệ thống mô phỏng lái 2 bậc tự do và nhúng vào môi trường Matlab Simulink để xác định giới hạn các góc chuyển động của bộ di động.
- Chế tạo được mô hình mô phỏng lái 2 bậc tự do.
- Lựa chọn, lắp đặt và kết nối các phần tử điều khiển theo sơ đồ mạch điện.
- Sử dụng Firmware SMC3 để nạp chương trình điều khiển nhúng cho Vi điều khiển và sử dụng phần mềm Simtool để thiết lập chương trình giả lập lái kết nối với hệ thống mô phỏng lái tạo ra các chuyển động mong muốn.

### II. Hạn chế của đồ án

- Mô hình hình học được xây dựng thiếu cơ sở và không có yêu cầu ban đầu.
- Lựa chọn các phần tử điều khiển cũng không có cơ sở và tính logic.
- Đáp ứng của mô hình điều khiển không diễn ra theo thời gian thực so với môi trường mô phỏng lái trên máy tính.

### III. Kết luận

Người duyệt đồng ý để nhóm sinh viên được bảo vệ đề tài tốt nghiệp trước Hội đồng chấm đồ án tốt nghiệp.

**Đánh giá:** 9.5/10 điểm

Hà Nội, ngày 12 tháng 8 năm 2023

Giáo viên phản biện

(Ký và ghi rõ họ tên)

Hoàng Sỹ Tuấn

## Lời mở đầu

Ngày nay, với sự phát triển của khoa học- kỹ thuật, các ứng dụng mô phỏng không chỉ đáp ứng yêu cầu to lớn của ngành công nghiệp giải trí mà còn phục vụ nhu cầu ngày càng tăng của đào tạo và huấn luyện trong cả quân sự và dân sự, ở nhiều lĩnh vực như: công nghiệp ô tô, máy bay, ... Một vài ứng dụng điển hình như là: đào tạo phi công lái máy bay giúp rút ngắn thời gian đào tạo, nghiên cứu phát triển hệ thống lái, đánh giá phân tích hành vi con người trong các tình huống rủi ro khi lái xe và ứng dụng trong các trò chơi mô phỏng lái.

Để có thể tự nghiên cứu, xây dựng và làm chủ các công nghệ trong nền tảng mô phỏng chuyển động nhóm em đã lựa chọn đề tài "**Thiết kế, mô phỏng và chế tạo hệ mô phỏng lái 2 bậc tự do**" làm đồ án tốt nghiệp.

Chúng em đã cố gắng tận dụng tất cả những kiến thức đã học ở trường cùng với những điều tự học và nghiên cứu để có thể hoàn thành tốt đồ án tốt nghiệp này. Kết quả sản phẩm đạt được trong ngày hôm nay tuy không lớn lao nhưng đó là thành quả của những năm tháng học tại trường là thành công đầu tiên của bản thân trước khi ra trường.

Chúng em muốn bày tỏ lòng biết ơn đối với sự giúp đỡ của nhà trường, Viện Cơ khí, đặc biệt đối với **PGS.TS.Phạm Đức An** – người hướng dẫn chúng em trong thời gian qua. Trong suốt quá trình hoàn thành đồ án, thầy luôn khuyến khích nghiên cứu tài liệu, giải đáp các khó khăn gặp phải. Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, đồ án này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để có điều kiện bổ sung, phục vụ tốt hơn trong công tác thực tế sau này.

Hà Nội, ngày..... tháng 8 năm 2023

## MỤC LỤC

<b>CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG MÔ PHỎNG CHUYỂN ĐỘNG .....</b>	<b>11</b>
1.1    Khái niệm về mô phỏng chuyển động.....	11
1.2    Các dạng robot song song được sử dụng để mô phỏng chuyển động ...	14
1.2.1    Khái niệm về robot song song.....	14
1.2.2    Các dạng robot song song thường được sử dụng trong mô phỏng chuyển động.....	14
<b>CHƯƠNG 2. THIẾT KẾ HỆ THỐNG MÔ PHỎNG LÁI 2DOF .....</b>	<b>17</b>
2.1    Tính toán thiết kế hệ thống mô phỏng lái 2 bậc tự do .....	17
2.1.1    Thiết kế mô hình hệ thống mô phỏng lái.....	17
2.1.2    Tính toán chọn động cơ .....	21
2.1.3    Kiểm tra bền các cơ cấu chịu tải trọng .....	23
2.2    Tính toán động học .....	24
2.2.1    Các bậc tự do trong mô phỏng lái .....	24
2.2.2    Tính toán quỹ đạo của hệ mô phỏng lái.....	25
<b>CHƯƠNG 3. MÔ PHỎNG MÔ HÌNH SỬ DỤNG CÔNG CỤ MATLAB SIMULINK - SIMMECHANICS .....</b>	<b>28</b>
3.1    Mô phỏng trên Simulink từ mô hình 3D Solidworks .....	28
3.1.1    Tạo file .xml từ mô hình trên Solidworks.....	29
3.1.2    Ý nghĩa của các khối:.....	30
3.2    Xây dựng sơ đồ mô phỏng.....	31
<b>CHƯƠNG 4. ỨNG DỤNG FIRMWARE SMC3 ĐỂ ĐIỀU KHIỂN ĐỘNG CƠ.....</b>	<b>35</b>
4.1    Công dụng của firmware SMC3 .....	35
4.1.1    Khái niệm .....	35
4.1.2    Sơ đồ khối chức năng điều khiển .....	35
4.2    Điều khiển động cơ bằng firmware SMC3.....	39
4.2.1    Các thiết bị kết nối.....	39
4.2.2    Đầu nối Arduino UNO R3 với Driver BTS7960 43A .....	43
4.2.3    Điều chỉnh PID sử dụng phần mềm SMC3 Utility .....	44

<b>CHƯƠNG 5. ĐIỀU KHIỂN HỆ THỐNG MÔ PHỎNG LÁI QUA SIMTOOLS .....</b>	<b>52</b>
5.1    Giới thiệu về Simtools .....	52
5.1.1    Khái niệm .....	52
5.1.2    Hướng di chuyển của xe trong Simtools .....	53
5.2    Sơ đồ khái quát của mô hình mô phỏng lái xe .....	56
5.3    Trình quản lý trò chơi Simtools GameManager.....	56
5.4    Công cụ Simtools GameEngine .....	58
5.4.1    Thiết lập giao diện kết nối (Interface Settings).....	58
5.4.2    Thiết lập trục giả lập (Axis Assignments) .....	59
5.4.3    Trung tâm điều chỉnh Simtools Tuning Center.....	61
<b>CHƯƠNG 6. THỰC THI VÀ THỬ NGHIỆM HỆ THỐNG MÔ PHỎNG LÁI 2 BẬC TỰ DO.....</b>	<b>63</b>
6.1    Thực thi, chế tạo mô hình mô phỏng lái 2 bậc tự do .....	63
6.2    Thử nghiệm mô hình .....	64
6.2.1    Thử nghiệm hệ thống với bậc tự do Pitch.....	64
6.2.2    Thử nghiệm với bậc tự do Roll .....	65
<b>CHƯƠNG 7. KẾT LUẬN .....</b>	<b>66</b>
7.1    Kết luận .....	66
7.2    Định hướng phát triển.....	66
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>1</b>

## **DANH MỤC HÌNH VẼ**

Hình 1.1 Hệ thống mô phỏng lái máy bay trong đào tạo phi công .....	12
Hình 1.2 Hệ thống ghế lái xe giải trí của Obutto.....	13
Hình 1.3 Hệ thống dạy lái xe qua mô phỏng ở Việt Nam .....	14
Hình 1.4 Bộ mô phỏng lái 2DOF.....	15
Hình 1.5 Bộ mô phỏng lái 3DOF.....	16
Hình 1.6 Bộ mô phỏng lái 6DOF.....	16
Hình 2.1 Tổng thể thiết kế hệ thống mô phỏng lái 2 bậc tự do.....	17
Hình 2.2 Các kích thước cơ bản của mô hình lái xe.....	18
Hình 2.3 Khung nhôm cố định .....	18
Hình 2.4 Bệ ghế .....	19
Hình 2.5 Nối bệ ghế với khung nhôm cố định .....	19
Hình 2.6 Mặt bích .....	19
Hình 2.7 Khớp nối Universal Joint .....	20
Hình 2.8 Tay đòn.....	20
Hình 2.9 Tay quay.....	21
Hình 2.10 Mô hình vị trí đặt tải trọng lớn nhất đặt lên động cơ .....	21
Hình 2.11 Mô hình đặt lực.....	22
Hình 2.12 Sơ đồ đặt lực tại B .....	22
Hình 2.13 Kết quả kiểm nghiệm cơ cấu cơ học tay quay .....	23
Hình 2.14 Kết quả kiểm tra biến dạng cơ cấu tay quay .....	24
Hình 2.15 Sáu bậc tự do trong chuyển động .....	24
Hình 2.16 . Mô hình hóa chuyển động hệ mô phỏng lái.....	25
Hình 2.17 Quỹ đạo chuyển động của điểm A quay quanh trục Oy.....	26
Hình 2.18 Quỹ đạo chuyển động của điểm A quay quanh trục Ox.....	26
Hình 3.1 Mô hình 3D Solidworks.....	28
Hình 3.2 Các file solid Part .....	29
Hình 3.3 Các bước tạo file .xml.....	29
Hình 3.4 Các khối mô phỏng trong Simulink .....	30
Hình 3.5 Thiết lập đầu vào cho Revolute Joint .....	32
Hình 3.6 Thiết lập đầu ra cho Revolute Joint.....	32
Hình 3.7 Sơ đồ Simulink của chương trình mô phỏng .....	33
Hình 3.8 Thiết lập thông số cho khối Sine Wave.....	33

Hình 3.9 Kết quả đạt được khi mô phỏng 2 động cơ quay cùng chiều theo góc từ -20 đến 20 độ .....	34
Hình 4.1 Sơ đồ khái niệm điều khiển của firmware SMC3 .....	35
Hình 4.2 Sơ đồ khái niệm tín hiệu điều khiển động cơ qua bộ PID .....	36
Hình 4.3 Sai số vi phân theo thời gian .....	37
Hình 4.4 Sai số của khâu tích phân theo thời gian .....	38
Hình 4.5 Sơ đồ khái niệm điều khiển hướng và xung ra cho động cơ.....	38
Hình 4.6 Arduino UNO R3 .....	39
Hình 4.7 Các chân của chiết áp .....	41
Hình 4.8 Sơ đồ đấu nối điều khiển 2 động cơ .....	44
Hình 4.9 Giao diện SMC3 Windows Utility .....	44
Hình 4.10 Chế độ xung ra Triangle .....	45
Hình 4.11 Chế độ đầu ra dạng Square .....	46
Hình 4.12 Chế độ đầu ra dạng Sine .....	46
Hình 4.13 Chế độ đầu ra Motion .....	46
Hình 4.14 Chế độ đầu ra Manual .....	47
Hình 4.15 Clip Limit và Max Limits trong SMC3 Utility .....	48
Hình 4.16 Setup cổng kết nối cho SMC3 Utility.....	49
Hình 4.17: Bộ điều khiển PID .....	51
Hình 5.1 Ứng dụng Live for Speed.....	52
Hình 5.2 Sáu bậc tự do trong chuyển động .....	53
Hình 5.3 Xe rẽ phải .....	53
Hình 5.4 Xe rẽ trái.....	54
Hình 5.5 Xe xuống dốc.....	54
Hình 5.6 Xe lên dốc.....	54
Hình 5.7 Xe nghiêng bên trái.....	55
Hình 5.8 Xe nghiêng bên phải .....	55
Hình 5.9 Sơ đồ kết nối và điều khiển hệ thống mô phỏng lái.....	56
Hình 5.10 Giao diện Simtools GameManager .....	57
Hình 5.11 Cài đặt các thông số ở giao diện Interface Settings trong Simtools	59
Hình 5.12 Thiết lập trực giá lập cho Simtools GameEngine .....	60
Hình 5.13 Cài đặt Axis Limiting .....	60
Hình 5.14 Output Testing .....	61
Hình 5.15 Menu Tools trong Simtools GameEngine .....	61

[Type here]

Hình 5.16 Simtools Tuning Center .....	62
Hình 6.1 Sản phẩm mô hình mô phỏng lái 2 bậc tự do .....	63
Hình 6.2 Bộ điều khiển mô hình.....	63
Hình 6.3 Thử nghiệm với bậc tự do Pitch dương.....	64
Hình 6.4 Thử nghiệm với bậc tự do Pitch âm .....	64
Hình 6.5 Thử nghiệm với bậc tự do Roll âm .....	65
Hình 6.6 Thử nghiệm với bậc tự do Roll dương .....	65

[Type here]

## **DANH MỤC BẢNG BIỂU**

Bảng 1.1 Bảng so sánh các ưu nhược điểm của các hệ mô phỏng.....	15
Bảng 3.1 Ý nghĩa các khối trong Solidworks.....	31

## CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG MÔ PHỎNG CHUYỂN ĐỘNG

### 1.1 Khái niệm về mô phỏng chuyển động

Mô phỏng chuyển động (Motion Simulation) hoặc nền tảng chuyển động (Motion Platform) là một cơ chế tạo cảm giác về chuyển động trong hệ thống mô phỏng lái (máy bay, ô tô,...) giúp người sử dụng có thêm cảm giác về chuyển động ngoài cảm giác có được từ các hình ảnh 3D, từ đó để cải thiện chất lượng mô phỏng cho hệ thống giúp người dùng có được trải nghiệm như đang lái xe trong môi trường thật. Trong một trình mô phỏng, chuyển động được đồng bộ hóa với màn hình/kính thực tế ảo hiển thị hình ảnh thế giới bên ngoài.

Trình mô phỏng chuyển động sử dụng trong lĩnh vực kỹ thuật để phân tích và xác minh hiệu suất thiết kế xe. Cụ thể là khả năng liên kết mô hình động dựa trên máy tính của một mẫu xe cùng với các chuyển động vật lý, mang đến cho người dùng khả năng cảm nhận chiếc xe mà không cần phải chế tạo các nguyên mẫu đắt tiền.

Mô phỏng chuyến bay cũng đã được các nhà sản xuất máy bay sử dụng để thử nghiệm phần cứng mới. Bằng cách kết nối buồng lái mô phỏng với màn hình trực quan cùng hệ thống điều khiển bay thực tế, tích hợp phi công với các thành phần điện, cơ khí và thủy lực tồn tại trên máy bay thật, các chuyên gia có thể tiến hành đánh giá hệ thống hoàn chỉnh trước khi thử nghiệm chuyến bay đầu tiên. Loại thử nghiệm này cho phép mô phỏng "lỗi hạt giống" (nghĩa là rò rỉ thủy lực có chủ ý, lỗi phần mềm hoặc tắt máy tính) nhằm xác thực rằng các tính năng thiết kế dự phòng của máy bay hoạt động đúng như dự định. Phi công thử nghiệm cũng có thể giúp xác định các thiếu sót của hệ thống, chẳng hạn như các chỉ số cảnh báo không đầy đủ hoặc thiếu, hoặc thậm chí là chuyển động của thanh điều khiển ngoài ý muốn. Thử nghiệm này là cần thiết để mô phỏng các sự kiện rủi ro cực kỳ cao không thể tiến hành trong chuyến bay.

Ngoài ra, tương tự như mô phỏng chuyến bay, mô phỏng lái xe được sử dụng như một công cụ học tập để đào tạo phi công, tương ứng là người lái xe. Ngoài ra còn là một công cụ phát triển cho ngành công nghiệp xe hơi. Nó cho phép thu được dữ liệu nhanh và hiệu quả trong các điều kiện kiểm tra liên tục.



Hình 1.1 Hệ thống mô phỏng lái máy bay trong đào tạo phi công

Có một loạt các ứng dụng có thể để mô phỏng lái xe. Nó được sử dụng để phân tích hành vi lái xe trong các tình huống rủi ro, hoặc để kiểm tra các yếu tố kiểm soát mới. Ví dụ :

- Giảm thời gian phát triển các hệ thống lái
- Đánh giá và điều chỉnh hệ thống trợ lý lái xe
- Kiểm tra các khái niệm an toàn
- Yếu tố con người
- Xử lý hệ thống thông tin
- Hành vi lái xe không chú ý
- Nhận thức của con người...

Một số trò chơi mô phỏng lái xe cho phép sử dụng các bộ điều khiển chuyên dụng như vô lăng, bàn đạp chân hoặc cần điều khiển. Trong những năm gần đây, bộ điều khiển trò chơi được thiết kế đã sử dụng công nghệ haptic để cung cấp phản hồi xúc giác, thời gian thực cho người dùng dưới dạng rung từ bộ điều khiển. Trình mô phỏng chuyển động thực hiện bước tiếp theo bằng cách cung cấp phản hồi xúc giác toàn thân cho người chơi. Ghế chơi game chuyển động có thể lăn sang trái và phải và ném về phía trước và lùi để mô phỏng các góc quay, tăng tốc và giảm tốc. Các nền tảng chuyển động cho phép trải nghiệm chơi game kích thích hơn và có khả năng thực tế hơn, thậm chí tương quan vật lý lớn hơn so với hình ảnh và âm thanh trong khi chơi trò chơi.



Hình 1.2 Hệ thống ghế lái xe giải trí của Obutto

Ở nước ta hiện nay, chúng ta thường mô phỏng lại các hệ thống thực đã có nhằm mục đích phục vụ đào tạo lái xe. Trong các trường đào tạo lái xe các kỹ sư xây dựng một hệ thống mô phỏng dạy lái xe ô tô bằng công nghệ thực tại ảo 3D được chế tạo nhằm tạo ra một cabin giống như cabin thật của xe ô tô. Người học sẽ sử dụng chiếc xe mô phỏng này để thực hiện các bài tập của mình trong một thế giới ảo 3D đợt lặp trình sẵn tương tự như thế 10 giới thật mà khi thi lái xe hoặc tham gia giao thông chúng ta sẽ gặp phải. Hệ thống đào tạo lái xe cơ bản giống với chiếc xe thực sự về mặt phần cứng vật lý như vô lăng, chân ga, ly hợp, phanh, hộp số, đồng hồ đo và phơong pháp hoạt động. Vì vậy hệ thống đào tạo này cung cấp cho người dùng tất cả những cảm giác như lái xe một chiếc xe thật. Hệ thống góp phần giải quyết những vấn đề phức tạp mà người học lái xe thường gặp phải, không đủ thời gian để tham gia vào công tác đào tạo lái xe, thiếu kinh nghiệm, giáo viên khó tính, áp lực ganh đua học lái xe, tâm lý căng thẳng, kết quả kiểm tra thấp, và sau khi có giấy phép lái xe không dám lái xe trên đường,... Hệ thống đào tạo mô phỏng lái xe ô tô cho phép học trong sự thuận tiện và thoải mái, giải quyết được các vấn đề bất cập nói trên



Hình 1.3 Hệ thống dạy lái xe qua mô phỏng ở Việt Nam

## 1.2 Các dạng robot song song được sử dụng để mô phỏng chuyển động

### 1.2.1 Khái niệm về robot song song

Robot song song là nhóm robot có các khâu chuyển động song song tương đối với nhau. Thông thường gồm có một đê cố định và một đê di động. Tùy thuộc vào số lượng các nhánh của robot song song mà ta có thể phân loại chúng với nhau.

Về ưu điểm, robot song song có khả năng chịu tải cao, độ cứng vững cao do kết cấu hình học của chúng. Thực hiện được các thao tác phức tạp với độ chính xác cao ( sai số chỉ phụ thuộc vào sai số đọc trực của các cụm cơ cầu chân riêng lẻ và sai số không bị tích lũy ). Giá thành của các robot song song ứng dụng trong gia công cơ khí ít hơn so với máy CNC có tính năng tương đương.

Tuy nhiên các robot song song cũng có những nhược điểm nhất định khi so sánh với robot chuỗi. Điểm hình là khoảng không gian làm việc nhỏ và khó thiết kế. Giải các bài toán động học, động lực học rất phức tạp. Có nhiều điểm kỳ dị trong không gian làm việc.

### 1.2.2 Các dạng robot song song thường được sử dụng trong mô phỏng chuyển động

Như đã giới thiệu trong phần trên, các robot song song được ứng dụng rất nhiều trong việc mô phỏng chuyển động. Khả năng mô phỏng phụ thuộc vào số bậc tự do ( DOF ) mà robot có thể cung cấp. Bậc tự do càng nhiều thì mô phỏng càng chân thực, chất lượng mô phỏng càng đảm bảo, nhưng cũng gây khó khăn cho khâu thiết kế và tính toán.

[Type here]

Nội dung so sánh	2 - DOF	3 - DOF	4 - DOF	6 - DOF
Tịnh tiến theo trục X	Không	Không	Không	Có
Tịnh tiến theo trục Y	Không	Không	Không	Có
Tịnh tiến theo trục Z	Không	Có	Có	Có
Quay theo trục X	Có	Có	Có	Có
Quay theo trục Y	Có	Có	Có	Có
Quay theo trục Z	Không	Không	Có	Có
Số biến tính toán động học	2	3	4	6
Tốc độ xử lý tính toán	Nhanh	Trung bình	Chậm	Rất chậm
Chi phí thực hiện dự án	Nhỏ	Trung bình	Lớn	Rất lớn

Bảng 1.1 Bảng so sánh các ưu nhược điểm của các hệ mô phỏng

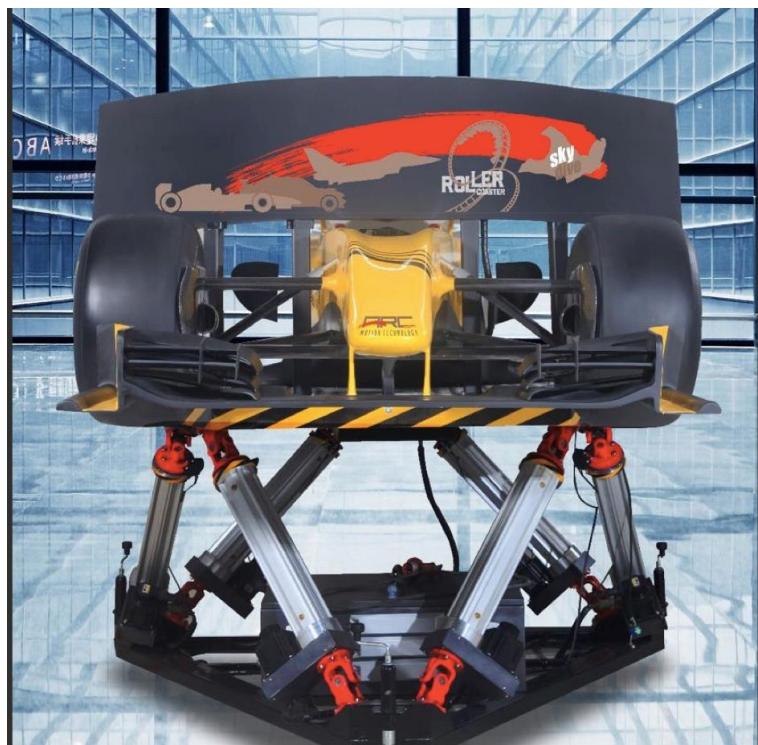


Hình 1.4 Bộ mô phỏng lái 2DOF

[Type here]



Hình 1.5 Bộ mô phỏng lái 3DOF



Hình 1.6 Bộ mô phỏng lái 6DOF

Sau khi nghiên cứu kỹ lưỡng về các mô hình mô phỏng lái xe hai bậc tự do cùng với sự học tập các thiết kế đã có sẵn trên thị trường trong nước và trên thế giới, căn cứ vào điều kiện thực tế, khả năng gia công chế tạo của mình, nhóm đề tài đã lựa chọn hệ thống mô phỏng lái 2 bậc tự do.

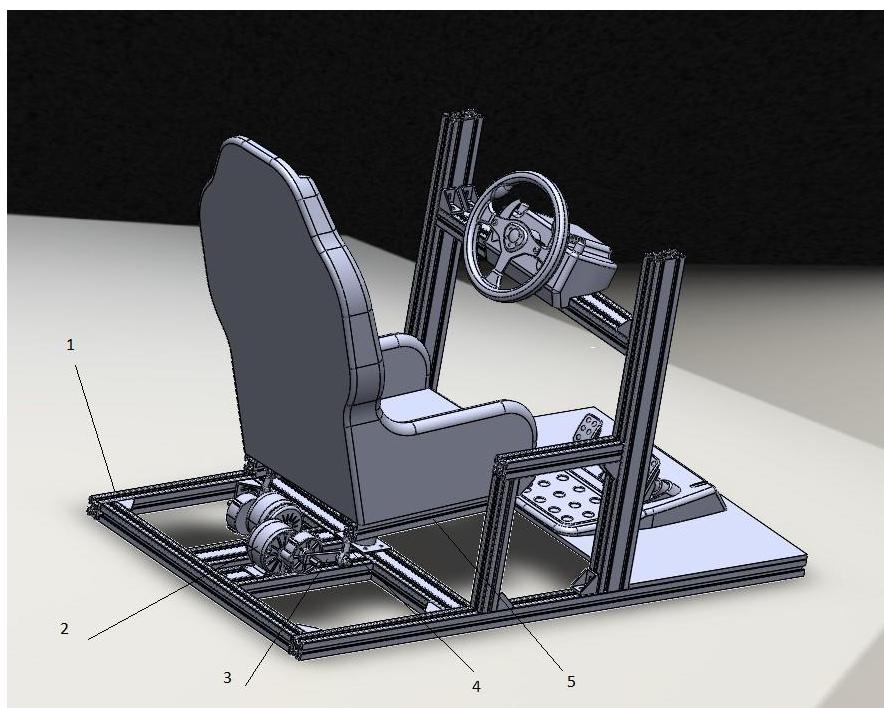
## CHƯƠNG 2. THIẾT KẾ HỆ THỐNG MÔ PHỎNG LÁI 2DOF

### 2.1 Tính toán thiết kế hệ thống mô phỏng lái 2 bậc tự do

Việc tính toán thiết kế mô hình lái xe bao gồm tính chọn công suất động cơ phù hợp với tải trọng yêu cầu và thiết kế các cơ cấu đảm bảo được độ bền cơ học, độ an toàn, tin cậy trong quá trình vận hành đồng thời cũng giảm được chi phí và thời gian trong quá trình chế tạo và thử nghiệm mô hình. Công đoạn tính toán này đặc biệt quan trọng và cần thiết đối với những cơ cấu làm việc ở điều kiện chịu tải trọng lớn, còn với các cơ cấu chịu tải trọng nhẹ hoặc không chịu tải trọng thì chỉ tiêu độ bền, độ cứng vững có tầm quan trọng thấp hơn so với các tiêu chí khác nhờ tiêu chí về giá thành, tiêu chí thẩm mỹ, và tiêu chí gọn nhẹ.

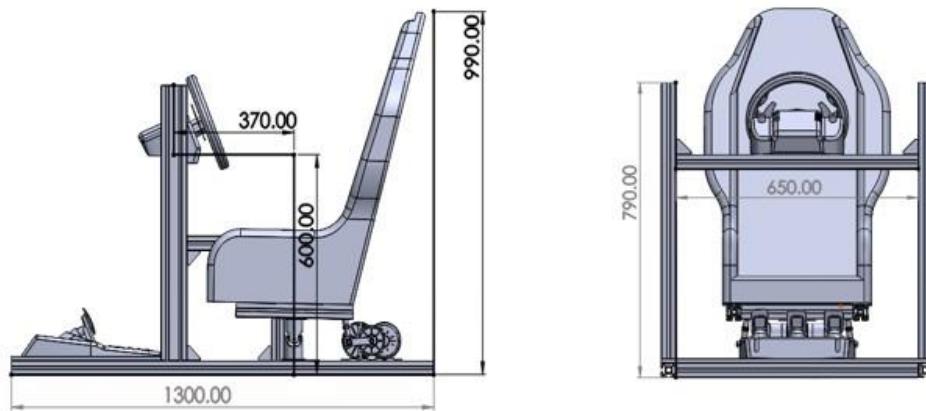
#### 2.1.1 Thiết kế mô hình hệ thống mô phỏng lái

Mô hình được xây dựng trong đề tài có kết cấu gồm các phần chính: phần đế, phần thân ghế, khớp cardan đóng vai trò là trực quay, động cơ, tay quay, tay đòn . Qua quá trình nghiên cứu các kết cấu, mô hình ghế học lái xe và các mô hình mô phỏng lái xe đã có trong nước và trên thế giới, kế thừa từ những tìm hiểu đó và căn cứ với điều kiện thực tế có thể của mình, hệ thống mô phỏng lái trong đồ án sẽ được thiết kế như sau:



Hình 2.1 Tổng thể thiết kế hệ thống mô phỏng lái 2 bậc tự do

1. Khung nhôm cố định 2. Động cơ
3. Tay quay 4. Tay đòn 5. Bệ ghế



Hình 2.2 Các kích thước cơ bản của mô hình lái xe

#### 2.1.1.1. Khung nhôm cố định

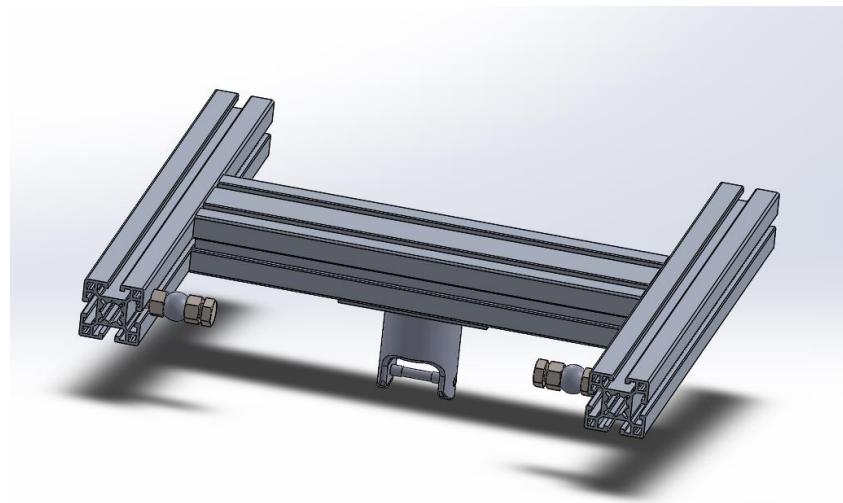


Hình 2.3 Khung nhôm cố định

Khung nhôm cố định bao gồm 2 loại nhôm định hình 40x40 và 40x80 được lắp ráp với nhau có nhiệm vụ chính để

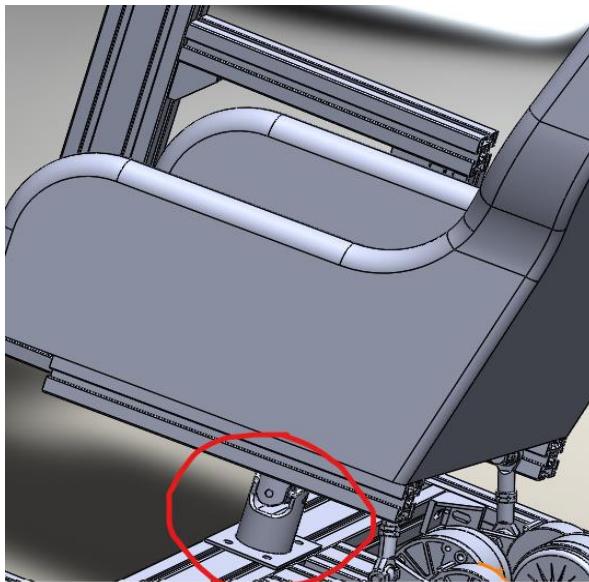
- Làm trụ vững cho cả hệ thống
- Là vị trí lắp động cơ, mạch điều khiển, vô lăng, chân ga và các thiết bị kết nối.

### 2.1.1.2. Bệ ghế

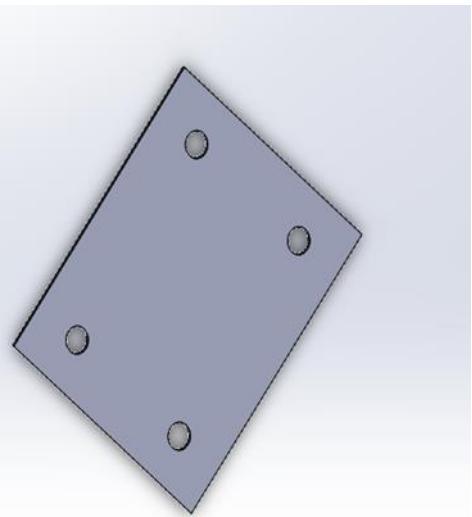


Hình 2.4 Bệ ghế

Bệ ghế được cấu tạo từ 3 thanh nhôm định hình 40x80, được kết nối với khung nhôm cố định qua 1 khớp cardan. Khớp nối này được hàn chặc xuống thanh nhôm qua mặt bích như hình 2.4.



Hình 2.5 Nối bệ ghế với khung nhôm cố định



Hình 2.6 Mặt bích

### 2.1.1.3. Khớp cardan

Tâm quay của mô hình là cơ cấu giúp cho ghế lái có thể chuyển động được theo các hướng khác nhau theo ý đồ mô phỏng. Để đáp ứng được yêu cầu này, cơ cấu được chọn là khớp cardan trong hệ thống dẫn động của ô tô.

Khớp cardan bao gồm một cặp bản lề nằm gần nhau, được nối với nhau bằng một trục chữ thập. Khớp này có một bậc tự do quay duy nhất bị hạn chế

[Type here]

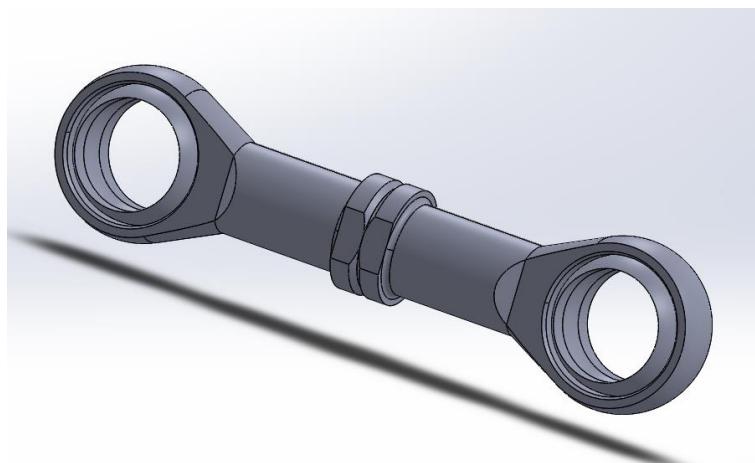
cũng như tất cả các phép tịnh tiến tương đối, tạo ra một khớp phô quát 2 bậc tự do (2-DOF)



Hình 2.7 Khớp nối Universal Joint

#### 2.1.1.4. Tay đòn

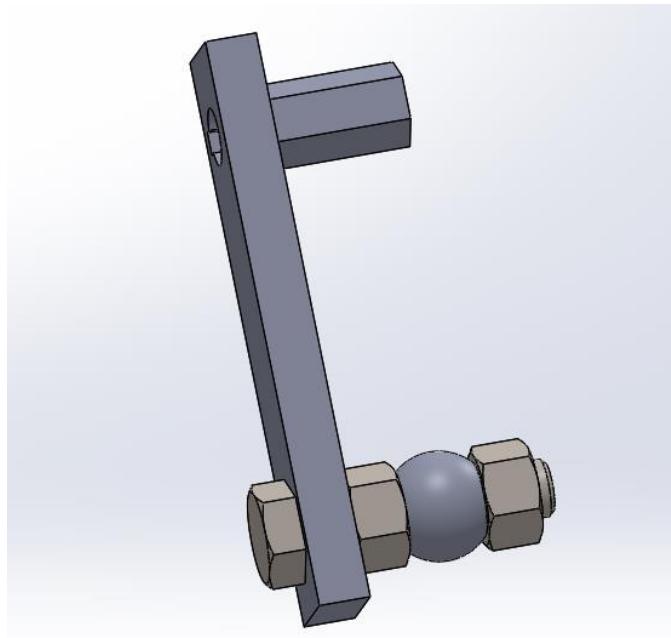
Cấu tạo gồm 2 khớp cầu mắt trâu M10 (tên Tiếng Anh là Rod End Bearing) được gắn với nhau. Có thể là khớp cầu ren ngoài (SA, SAL) hoặc ren trong (SI, SIL).



Hình 2.8 Tay đòn

#### 2.1.1.5. Tay quay

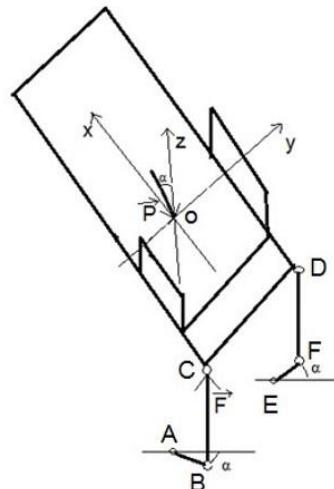
Tay quay được chế tạo từ phôi nhôm đặc có kích thước 150x30x12 (mm). Ghép nối với trực động cơ bằng một bulong cố định và một bulong kẹp chặt. Ghép nối với khớp cầu qua bulong và ốc.



Hình 2.9 Tay quay

### 2.1.2 Tính toán chọn động cơ

Để chọn được động cơ có đủ công suất cần xác định vị trí của mô hình mà tại đó tải trọng đặt lên động cơ là lớn nhất, tính công suất động cơ cần thiết đáp ứng được tải trọng tại vị trí đó rồi nhân với một hệ số an toàn sẽ được công suất của động cơ cần chọn.

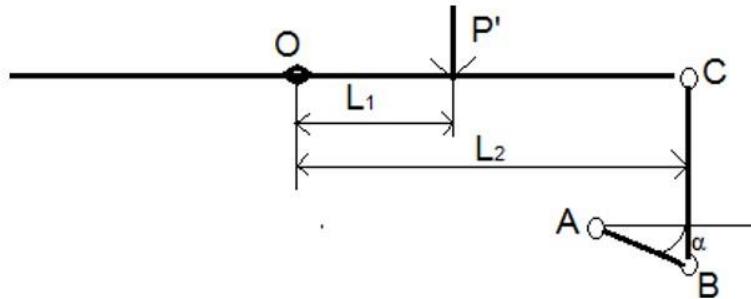


Hình 2.10 Mô hình vị trí đặt tải trọng lớn nhất đặt lên động cơ

Đặt vào tâm quay O của mô hình một hệ trục tọa độ Oxyz, lấy vị trí gốc của hệ trục tọa độ là vị trí mô hình ở trạng thái cân bằng, tại vị trí đó có trục Ox trùng với phương chiều dài mô hình, Oy là trục trùng với phương chiều ngang của mô hình, Oz trùng với phương chiều cao của mô hình. Tải trọng lớn nhất mà mô hình lái xe đặt lên động cơ là vị trí mà ghế lái xe có độ nghiêng lớn nhất (vị trí mà một động cơ tại vị trí thấp nhất, một động cơ ở vị trí cao nhất trong quỹ

[Type here]

đạo chuyển động của nó) được mô tả như trên Hình 2.10. Chiều các thành phần tải trọng lên mặt phẳng Oxz, các thành phần lực được biểu diễn như trong hình:



Hình 2.11 Mô hình đặt lực

Trong đó:

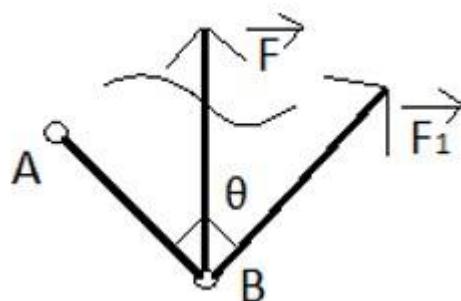
- $P' = P \cos(\alpha)$  là trọng lực của cơ thể người và thân ghế ( $P = 1500\text{N}$ ,  $\alpha=45^\circ$ );
- $F$  là lực đẩy động cơ tác động lên thân ghế ( $\text{N}$ );
- $M$  là momen của động cơ ( $\text{Nm}$ );
- $L_1, L_2$  là khoảng cách của các điểm đặt lực tới tâm quay ( $L_1=9.5\text{ cm}$ ,  $L_2=38\text{ cm}$ );
- $AB$  là cánh tay đòn (cơ cấu phát động,  $AB = 8\text{cm}$ );
- $BC$  là thanh truyền ( $15\text{ cm}$ ).

Viết phương trình cân bằng momen tại tâm quay O ta có:

$$\begin{aligned} P' \cdot L_1 &= F \cdot L_2 \Leftrightarrow P \cos 45^\circ \cdot L_1 = F \cdot L_2 \\ \Leftrightarrow F &= \frac{P \cos 45^\circ \cdot L_1}{L_2} = 265,1 (\text{N}) \end{aligned}$$

Vậy lực cần đặt vào điểm C là  $F = 265,1(\text{N})$

Xét các lực đặt tại điểm B như hình vẽ:



Hình 2.12 Sơ đồ đặt lực tại B

Trong đó:

- $F$  là lực động cơ truyền cho thanh dẫn động BC ( $F = 265,1 \text{ N}$ );
- $F_1$  là lực do momen động cơ tạo ra;
- $\theta$  là góc lệch giữa  $F_1$  và  $F$  ( $\theta = 45^\circ$ ).

Theo sơ đồ đặt lực như hình vẽ ta có thể xác định lực  $F_1$ :

[Type here]

$$F_1 = \frac{F}{\cos\theta} = \frac{265,1}{\cos 45^\circ} = 374,9 \text{ (N)}$$

Theo công thức tính công suất động cơ:

$$P = M \cdot \omega$$

Trong đó:

- $M$  là momen được tạo ra bởi động cơ:  $M = F_1 \cdot R$ ;
- $F_1$  là lực phát động từ động cơ (374,9 N);
- $R$  độ dài cơ cấu phát động (0.025 m)

$$\Rightarrow M = F_1 \cdot R = 9,37 \text{ (Nm)};$$

- $\omega$  là vận tốc góc của động cơ.

Dữ liệu từ chương trình điều khiển gửi vị trí mới của động cơ xuống mạch điều khiển động cơ sau mỗi 110 (ms). Động cơ có quỹ đạo chuyển động trong phạm vi góc bằng  $90^\circ$ . Có nghĩa là, trong trường hợp động cơ phải quay từ vị trí cao nhất xuống vị trí thấp, động cơ phải hoàn thành trong khoảng 110(ms)  $\Rightarrow$  động cơ cần có vận tốc góc  $\omega = 14,3 \text{ (rad/s)}$ . Gọi  $k$  là hệ số an toàn, lấy  $k = 1,5$ .

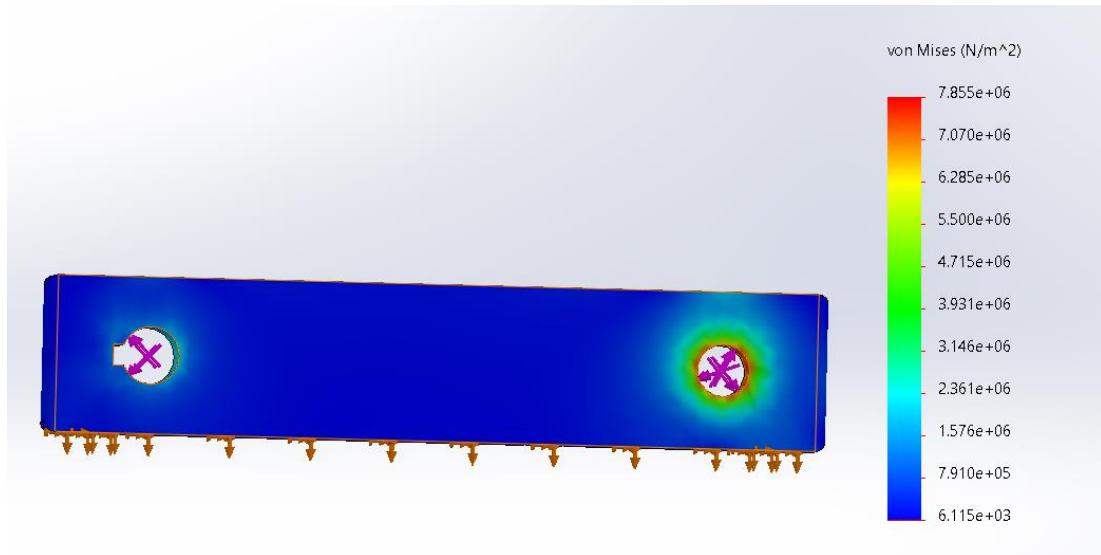
Công suất động cơ yêu cầu:

$$P = M \cdot \omega \cdot k = 9,37 \cdot 14,3 \cdot 1,5 = 200,98 \text{ (W)}$$

Như vậy, cần chọn động cơ có công suất tối thiểu là 200,98(W) để đáp ứng đủ tải mô hình lái xe có thể hoạt động. Từ đó chọn động cơ MY1016Z2.

### 2.1.3 Kiểm tra bền các cơ cấu chịu tải trọng

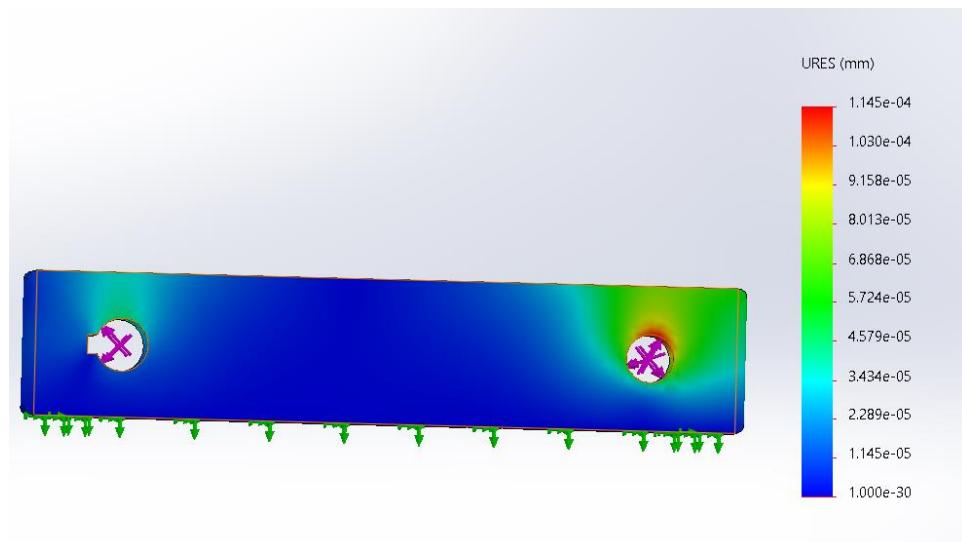
Cơ cấu tay quay được thiết kế bằng phần mềm solidworks và sử dụng solidworks để tính toán độ bền biến dạng và độ bền chịu tải trọng dựa trên tải trọng đã được tính toán ở phần 2.1.1 và thu được kết quả như sau:



Hình 2.13 Kết quả kiểm nghiệm cơ cấu cơ học tay quay

Cơ cấu tay quay được chế tạo từ phôi nhôm đặc có kích thước 150x30x12 (mm). Và theo như kết quả kiểm nghiệm bằng phần mềm Solidwork, cơ cấu có thể chịu ứng suất tối đa  $7,855 \cdot 10^6$

Khi mô phỏng, dựa vào sự phân bố các vùng màu trên cơ cấu, kết quả có một số điểm nguy hiểm trên cơ cấu nhưng cơ cấu vẫn có thể hoạt động an toàn với tải yêu cầu.



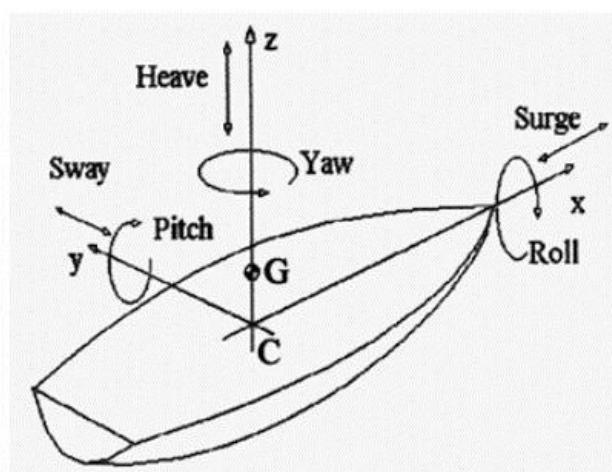
Hình 2.14 Kết quả kiểm tra biến dạng cơ cấu tay quay

Theo kết quả kiểm nghiệm cơ cấu có biến dạng lớn nhất tại vị trí chịu lực, độ biến dạng lớn nhất là  $1,145 \cdot 10^{-4}$  mm.

## 2.2 Tính toán động học

### 2.2.1 Các bậc tự do trong mô phỏng lái

Trong tự nhiên, xe di chuyển trên đường dưới tác động của nội lực và ngoại lực tác động lên xe, làm cho xe có xu hướng chuyển động theo sáu bậc tự do như hình:



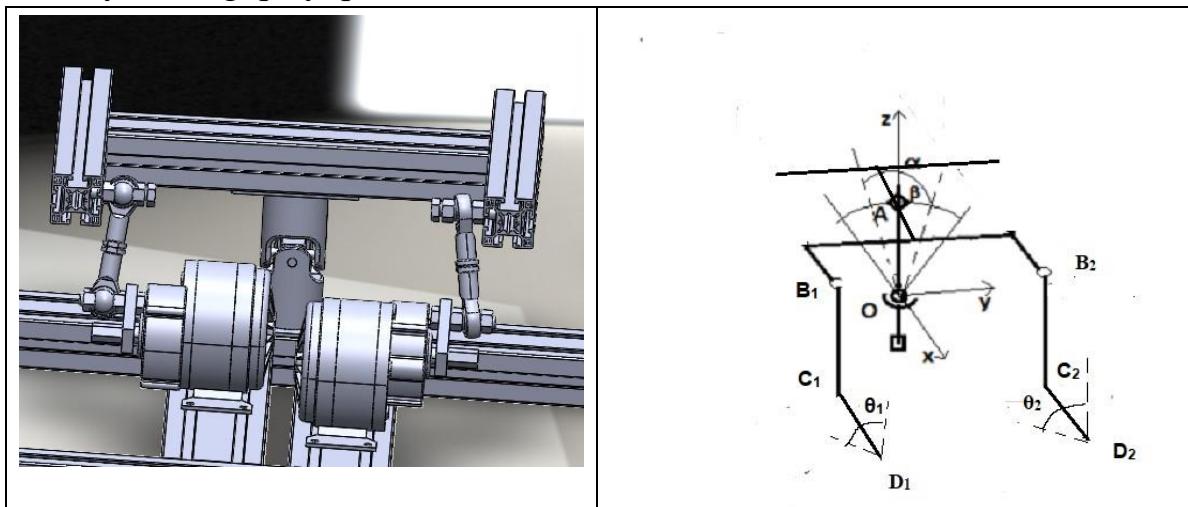
Hình 2.15 Sáu bậc tự do trong chuyển động

Đồ án thiết kế về hệ mô phỏng lái 2 bậc tự do nên sẽ chỉ đề cập đến 2 bậc tự do bao gồm:

- **Roll:** là bậc tự do mô phỏng các lực làm cho xe quay quanh trục X. Chuyển động của bậc tự do này có thể có khi xe di chuyển trên đường gồ ghề, 28 mảnh mô, rẽ trái, rẽ phải hoặc do xe chịu tác động của ngoại lực làm cho xe có chuyển động quay quanh trục X.
- **Pitch:** là bậc tự do các lực làm cho xe quay quanh trục Y gây nên. Chuyển động của bậc tự do này có thể có được khi xe lên dốc, xuống dốc theo dao động của đường, do tác động của lực quán tính lên xe hoặc các ngoại lực khác làm cho xe có chuyển động quay quanh trục Y.

### 2.2.2 Tính toán quỹ đạo của hệ mô phỏng lái

Hệ thống mô phỏng lái 2 bậc tự do sẽ mô phỏng các lực đặt vào xe làm cho xe chuyển động quay quanh trục X, Y.



Hình 2.16 . Mô hình hóa chuyển động hệ mô phỏng lái

Trong đó:

- $D_1C_1, D_2C_2$  là tay quay của động cơ
- $C_1B_1, C_2B_2$  là tay đòn của động cơ
- $\theta_1, \theta_2$  là góc quay của động cơ
- Điểm gốc O của hệ trục tọa độ Oxyz đặt tại khớp cardan (trục Ox, Oy lần lượt trùng với trục chuyển động của mô hình theo bậc tự do Pitch và Roll)
- Điểm A là trọng tâm của bệ ghế
- Góc  $\alpha, \beta$  là các góc tạo thành khi bệ ghế quay quanh trục Oy, Ox

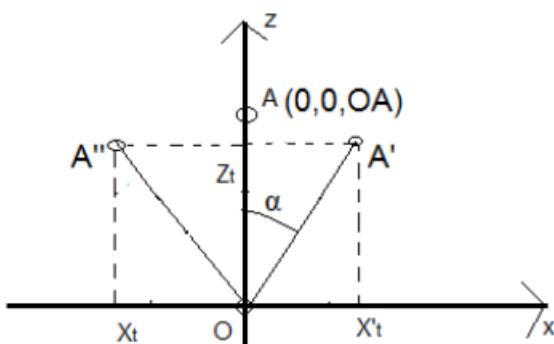
Tại vị trí cân bằng của hệ thống là vị trí các góc  $\theta_1=0, \theta_2=0$ . Quy ước chiều dương là khi tay quay  $D_1C_1, D_2C_2$  quay cùng chiều kim đồng hồ tạo các góc  $\theta_1, \theta_2$  có giá trị dương. Các vị trí của điểm A trên quỹ đạo chuyển động có

[Type here]

thể được xác định trên hệ trục tọa độ Oxyz thông qua vị trí ban đầu của điểm A tại vị trí cân bằng và các góc quay  $\theta_1, \theta_2$

Trong quá trình hoạt động, điểm A chuyển động theo hai bậc tự do là quay quanh trục Ox, và quay quanh trục Oy tạo thành các góc  $\alpha, \beta$  so với vị trí cân bằng từ đó có thể xác định được vị trí của điểm A trên hệ trục tọa độ Oxyz. Khi cả 2 động cơ quay cùng chiều nhau một góc  $\theta$ , điểm A sẽ quay quanh trục Oy một góc  $\alpha = \theta$

Quỹ đạo chuyển động của điểm A trên mặt phẳng Oxz được biểu diễn trên hình vẽ sau:



Hình 2.17 Quỹ đạo chuyển động của điểm A quay quanh trục Oy

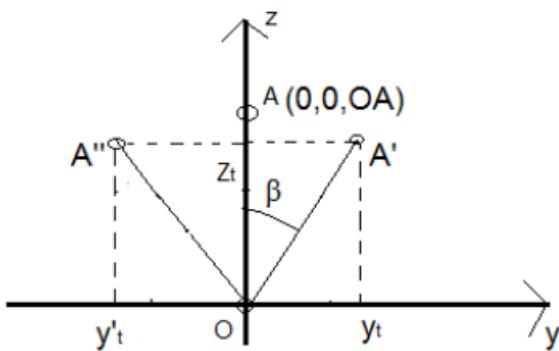
Vị trí của điểm A tại thời điểm t bất kỳ khi chuyển động quay quanh trục Oy được xác định theo công thức:

$$x_{A(t)} = OA \cdot \sin(\alpha t)$$

$$y_{A(t)} = 0$$

$$z_{A(t)} = OA \cdot \cos(\alpha t)$$

Khi cả 2 động cơ quay ngược chiều nhau một góc  $\theta$ , điểm A sẽ quay quanh trục Ox một góc  $\beta = \theta$ . Tương tự như trên, quỹ đạo của điểm A khi quay quanh trục Ox được biểu diễn như hình vẽ sau:



Hình 2.18 Quỹ đạo chuyển động của điểm A quay quanh trục Ox

[Type here]

Vị trí của điểm A tại thời điểm t bất kỳ khi chuyển động quay quanh trục Ox được xác định theo công thức:

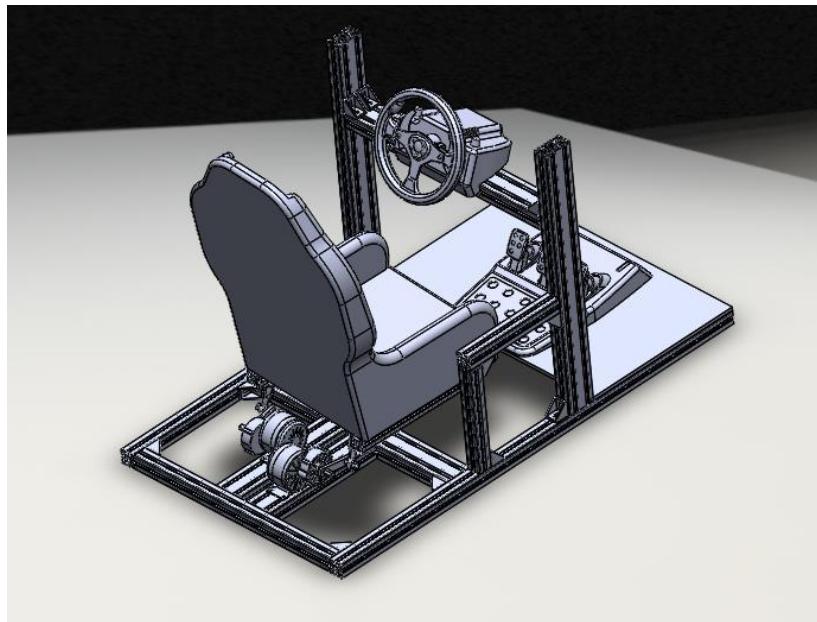
$$x_{A(t)} = 0$$

$$y_{A(t)} = OA \cdot \sin(\beta t)$$

$$z_{A(t)} = OA \cdot \cos(\beta t)$$

## CHƯƠNG 3. MÔ PHỎNG MÔ HÌNH SỬ DỤNG CÔNG CỤ MATLAB SIMULINK - SIMMECHANICS

### 3.1 Mô phỏng trên Simulink từ mô hình 3D Solidworks

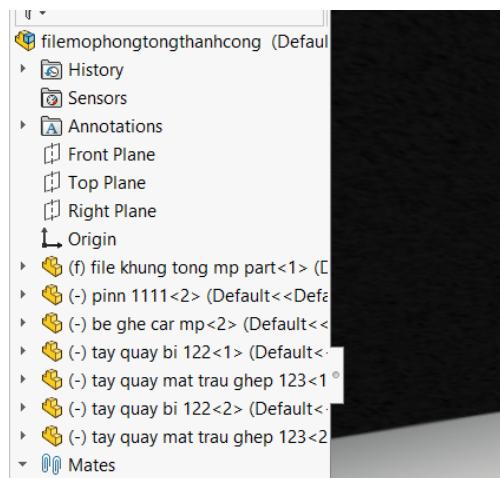


Hình 3.1 Mô hình 3D Solidworks

Mô hình 3D đã được thiết kế trong môi trường SolidWorks với đầy đủ các ràng buộc về ghép nối khớp xoay (Revolute) và khớp cầu (Spherical). Bằng công cụ Simscape Multibody Link, ta sẽ xuất ra được một tập tin .xml với biểu thị đầy đủ cấu trúc của robot trong môi trường Simulink.

Một mẹo nhỏ để sơ đồ các khối trong môi trường Simulink được tối giản nhất là loại bỏ các ràng buộc không cần thiết trong mô hình như ràng buộc (Mate) của khớp nối bulong đai óc, ràng buộc của các file Solid Part trong một file Solid Assembly con. Như hình vẽ ở dưới, ta thấy tất cả các mô hình Assembly con đã được lưu về file Solid Part.

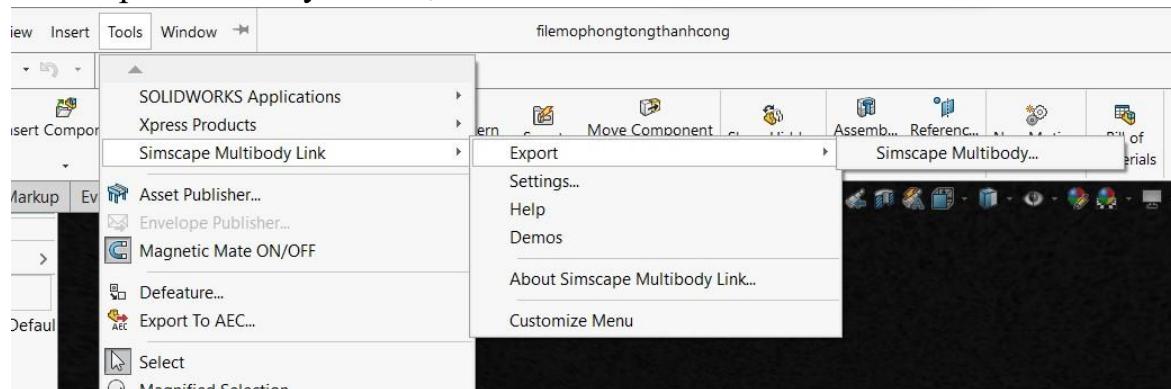
[Type here]



Hình 3.2 Các file solid Part

### 3.1.1 Tạo file .xml từ mô hình trên Solidworks

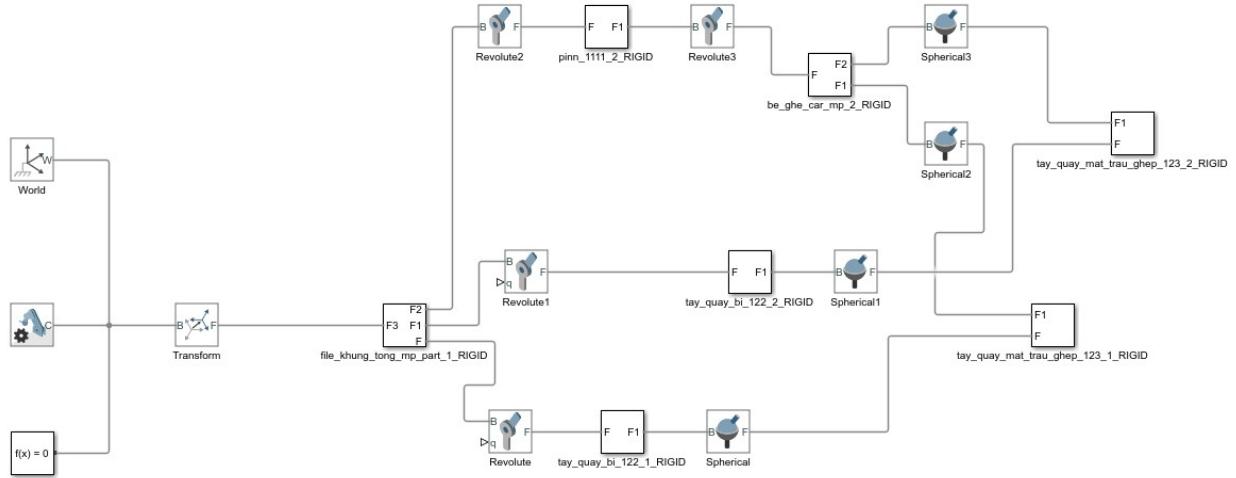
- Bước 1: Vào Tools ► Simscape Multibody Link ► Export ► Simscape Multibody... để tạo file .xml



Hình 3.3 Các bước tạo file .xml

- Bước 2: Trong giao diện Matlab, mở folder vừa Export file .xml của mô hình robot trong cửa sổ Current Folder. Trong cửa sổ Command Window, ta gõ lệnh smimport ('Tên .xml file '). Ví dụ smimport ('Assem2dof.xml').

Giao diện các khối trong Simulink như sau:

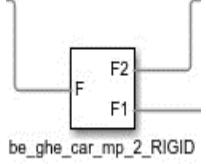
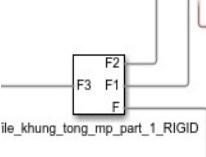
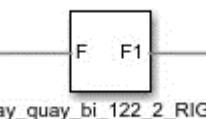
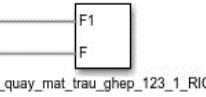


Hình 3.4 Các khối mô phỏng trong Simulink

### 3.1.2 Ý nghĩa của các khối:

Khối	Tên khối	Ý nghĩa
	Transform	Xác định một chuyển đổi 3D cố định và cứng giữa hai khung hoặc cho phần mở rộng động học. Các thành phần xác định các phần tịnh tiến và quay cho mô hình và GDLs của nó
	World Frame	Xây dựng tham chiếu của mô hình cơ khí. World Frame là nền của tất cả liên kết cơ học.
	Mechanism Configuration	Cung cấp các tham số cơ học và mô phỏng cho một khối khép kín gồm các Simscape Multibody được kết nối với nhau. Các tham số bao gồm trọng lực và một delta tuyến tính hóa.
	Solver Configuration	Xác định các giá trị cấu hình cho mô phỏng, các tham số của dung sai nhất quán (consistent tolerance), loại bộ giải (solver type), thời gian mẫu, đại số tuyến tính và độ trễ dung lượng bộ nhớ

[Type here]

		Khối đại diện cho bệ ghế trong mô hình Solidworks
		Khối đại diện cho khung nhôm cố định trong Solidworks
		Khối đại diện cho tay quay trong Solidworks
		Khối đại diện cho tay đòn trong Solidworks
	Revolute Joint	Điển tả cho khớp 2 bậc tự do quay
	Spherical joint	Điển tả cho khớp cầu

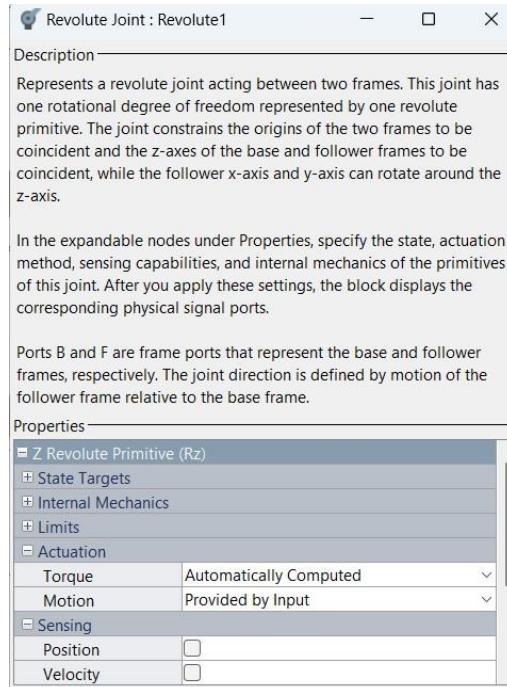
Bảng 3.1 Ý nghĩa các khối trong Solidworks

### 3.2 Xây dựng sơ đồ mô phỏng

Đưa tín hiệu góc xoay cho động cơ bằng cách kích đúp vào mỗi khối Revolute Joint. Chọn mục Actuation, ta thiết lập đầu vào như sau:

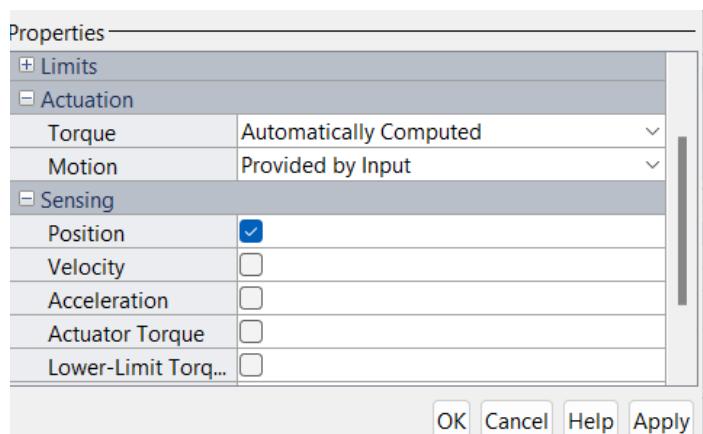
- Torque (Momen xoắn) : Automatically Computed
- Motion (Chuyển động): Provided by Input

[Type here]



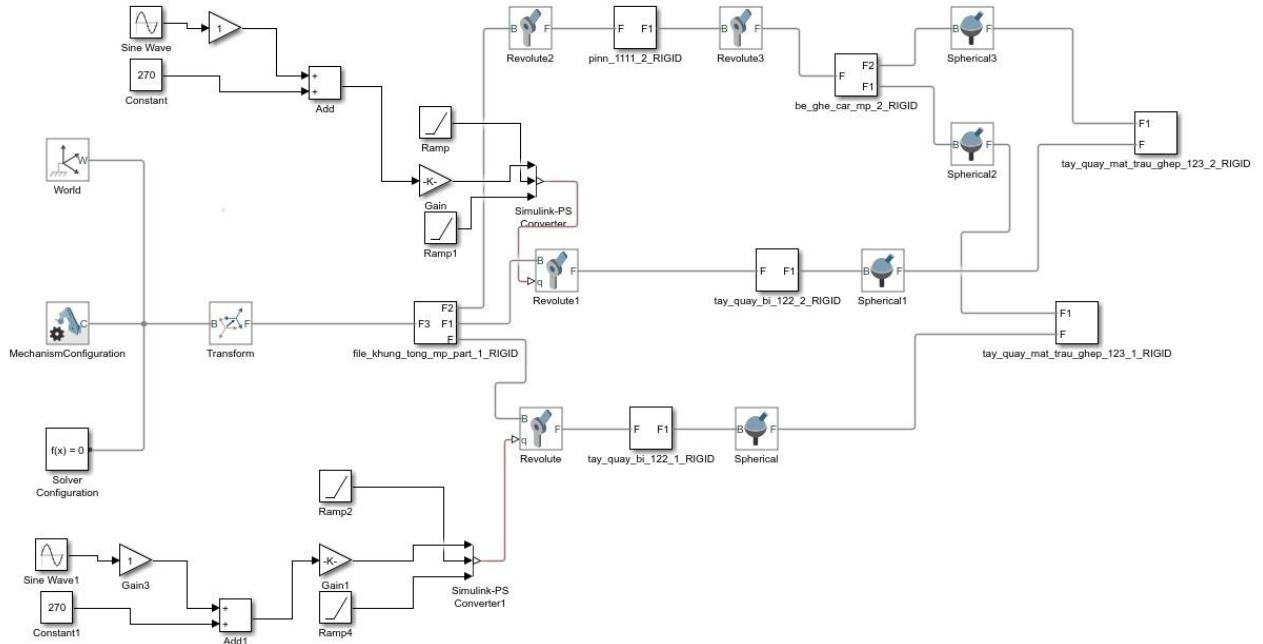
Hình 3.5 Thiết lập đầu vào cho Revolute Joint

Ở mục Sensing, ta chọn tín hiệu đầu ra đưa ra giá trị góc xoay. Tích vào Position.



Hình 3.6 Thiết lập đầu ra cho Revolute Joint

Ta lập được sơ đồ mô phỏng như sau

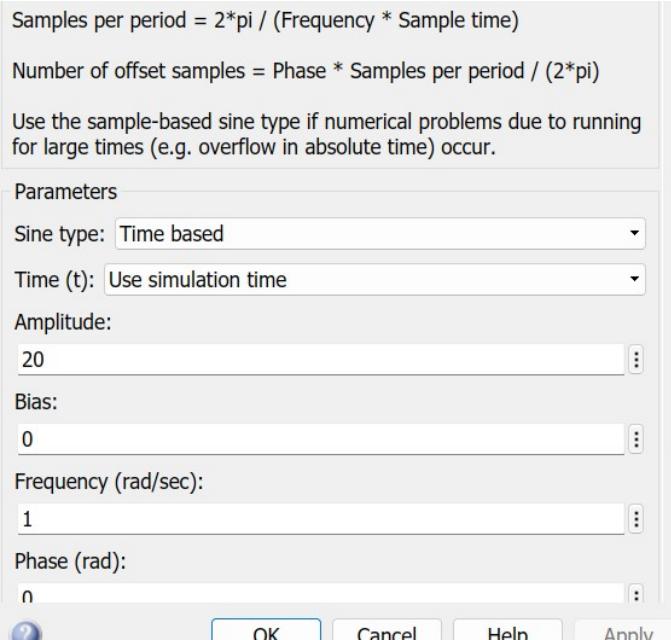


Hình 3.7 Sơ đồ Simulink của chương trình mô phỏng

Trong đó:

- Khối Simulink-PS Converter dùng để dịch các yếu tố từ Second Generation sang First Generation của Simulink để giải thích dữ liệu và tín hiệu. Được dùng để theo dõi các Revolute Joint.
- Khối Constant cung cấp tín hiệu là một hằng số
- Khối Gain để khuếch đại tín hiệu
- Khối Add: tổng 2 tín hiệu
- Khối Sine Wave: đưa thông số đầu ra bằng một sóng hình sin

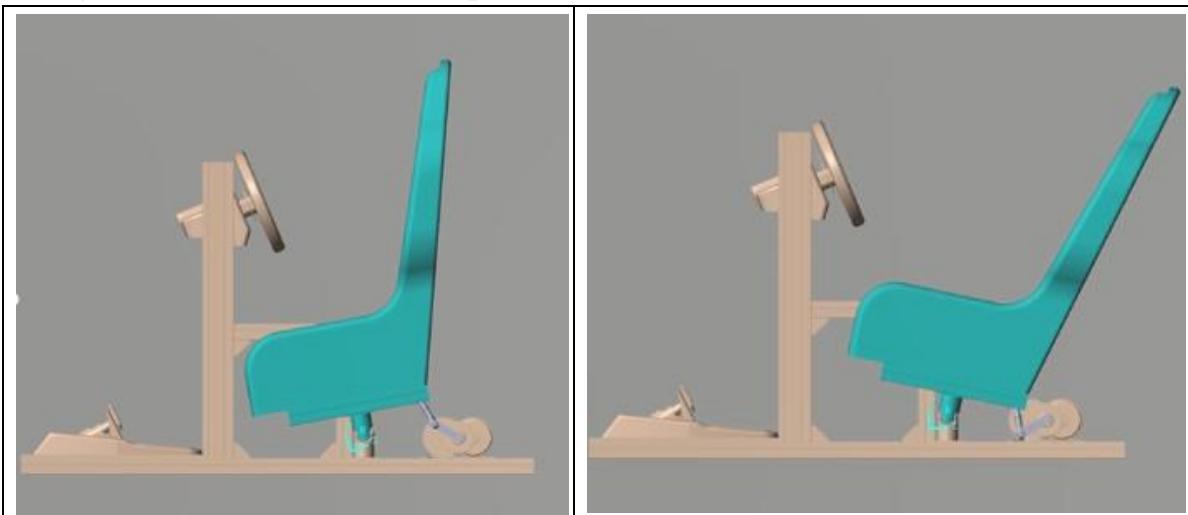
Cài đặt thông số của khối Sine Wave để đặt thông số đầu vào:



Hình 3.8 Thiết lập thông số cho khối Sine Wave

[Type here]

Trong hình 3.8, ta đặt thông số đầu vào cho 2 động cơ quay cùng chiều theo góc từ  $-20^\circ$  đến  $20^\circ$ . Kết quả đạt được như hình vẽ:



Hình 3.9 Kết quả đạt được khi mô phỏng 2 động cơ quay cùng chiều theo góc từ  $-20^\circ$  đến  $20^\circ$  độ

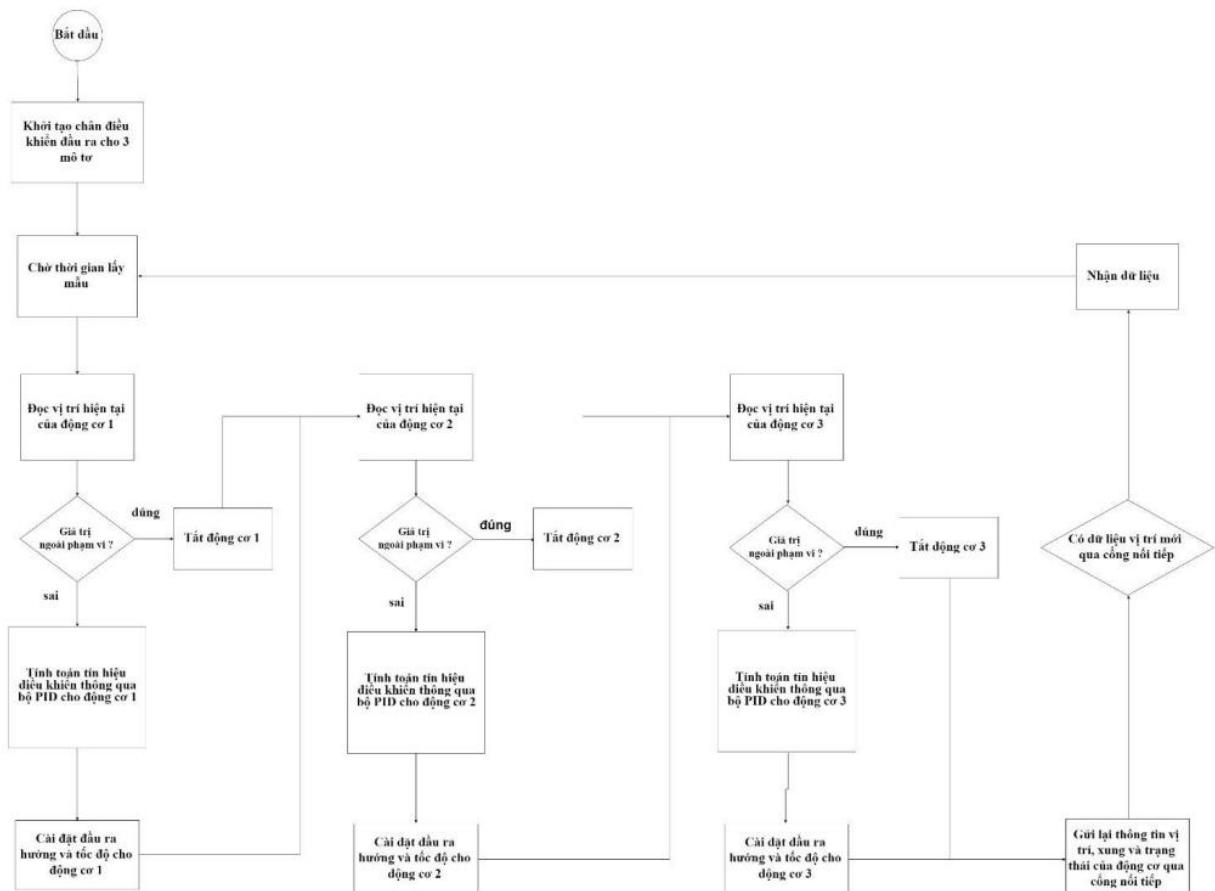
## CHƯƠNG 4. ỨNG DỤNG FIRMWARE SMC3 ĐỂ ĐIỀU KHIỂN ĐỘNG CƠ

### 4.1 Công dụng của firmware SMC3

#### 4.1.1 Khái niệm

SMC3 là firmware cho phép điều khiển đồng thời tối đa 3 động cơ bằng phản hồi tương tự được viết cho Arduino UNO R3 và sử dụng bộ điều khiển PID để điều khiển vị trí của động cơ. SMC3 được thiết kế riêng biệt để sử dụng với phần mềm điều khiển chuyên động Simtools, độ chính xác 10 bit (1024 bước) và điều khiển đảo ngược động cơ nếu động cơ di chuyển vượt quá giới hạn cho phép.

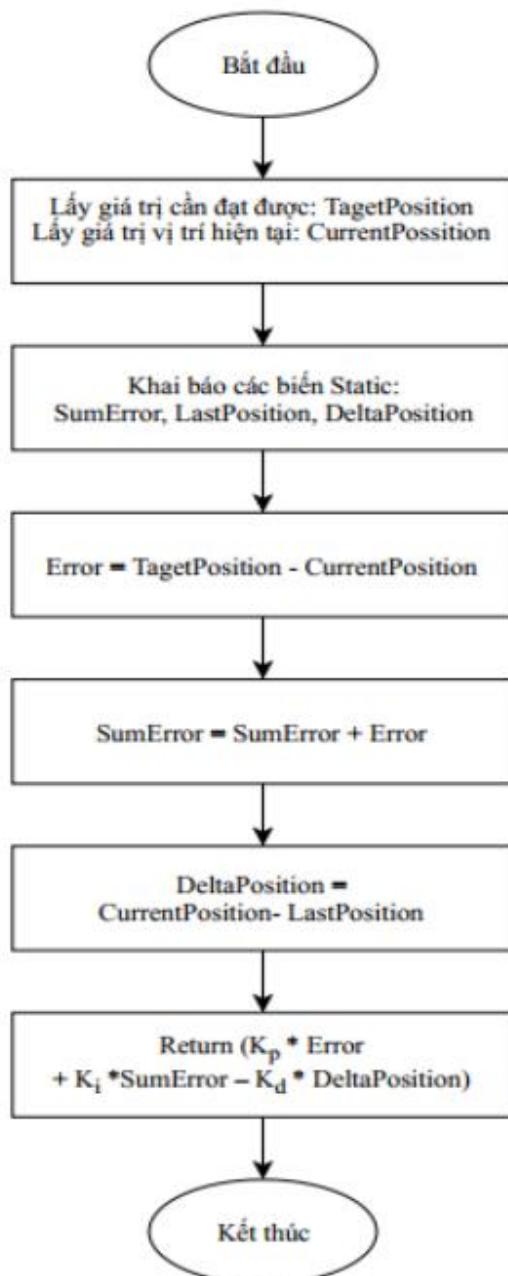
#### 4.1.2 Sơ đồ khái niệm năng điều khiển



Hình 4.1 Sơ đồ khái niệm năng điều khiển của firmware SMC3

Khi bắt đầu, arduino sẽ cấu hình các chân để có thể điều khiển được động cơ và đọc được tín hiệu analog. Chờ khoảng thời gian lấy mẫu, Arduino sẽ đọc tín hiệu vị trí hiện tại của động cơ 1. Nếu vị trí nằm trong khoảng cho phép, arduino sẽ tính toán tín hiệu điều khiển thông qua bộ điều khiển PID và xuất ra điện áp điều khiển vị trí cũng như hướng quay của động cơ. Nếu ngoài khoảng

cho phép thì sẽ tắt động cơ. SMC3 sẽ tương tự các bước trên đối với động cơ 2 và 3. Tiếp theo, Arduino sẽ kiểm tra xem có lệnh điều khiển từ Simtools thông qua kiểm tra trạng thái cổng truyền thông UART, nếu có sẽ thực hiện theo trình tự nhận dữ liệu, phân tích và xử lý dữ liệu nhận được, phản hồi lại các thông tin về Simtools. Nếu không có tín hiệu từ Simtools, Arduino sẽ bỏ qua các bước trên gửi lại phản hồi ngay lập tức. Chu trình được lặp lại trong khoảng thời gian 250 micro giây. Đối với lần bắt đầu khởi động, arduino sẽ khởi tại giá trị vị trí điều khiển là giá trị vị trí cân bằng nên khi hệ thống có điện tất cả các động cơ sẽ được điều khiển về vị trí cân bằng sau đó mới nhận các lệnh điều khiển từ Simtools.

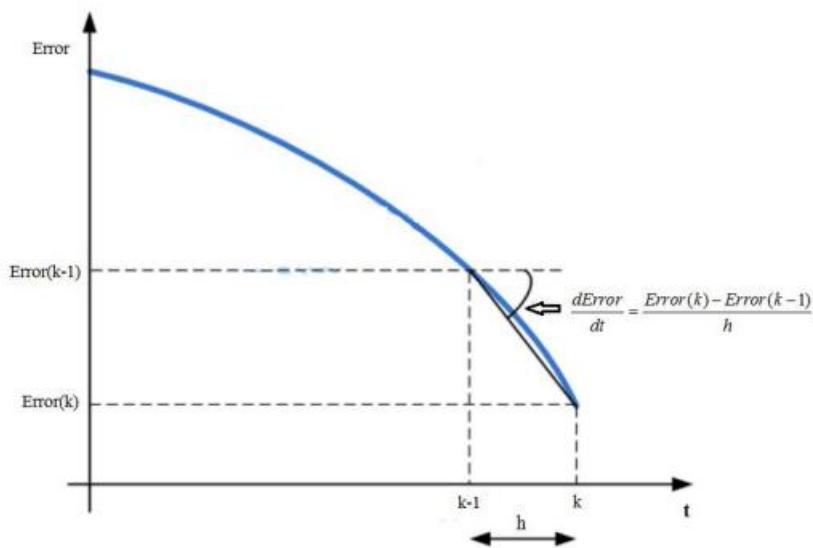


Hình 4.2 Sơ đồ khái niệm tín hiệu điều khiển động cơ qua bộ PID

Để tính toán tín hiệu đầu ra của bộ điều khiển PID, đặt vị trí hiện tại của mô tơ (CurrentPosition) và vị trí cần đặt được (TargetPosition). Error là sai số, tương đương với hiệu của CurrentPosition và TargetPosition nên có tín hiệu tuyến tính

$$u_1 = K_p \times \text{Error}.$$

Tiếp theo tính toán tín hiệu vi phân, vì thời gian lấy mẫu cho các bộ điều khiển thường rất bé nên có thể xấp xỉ đạo hàm bằng sự thay đổi của Error trong 2 lần lấy mẫu liên tiếp:



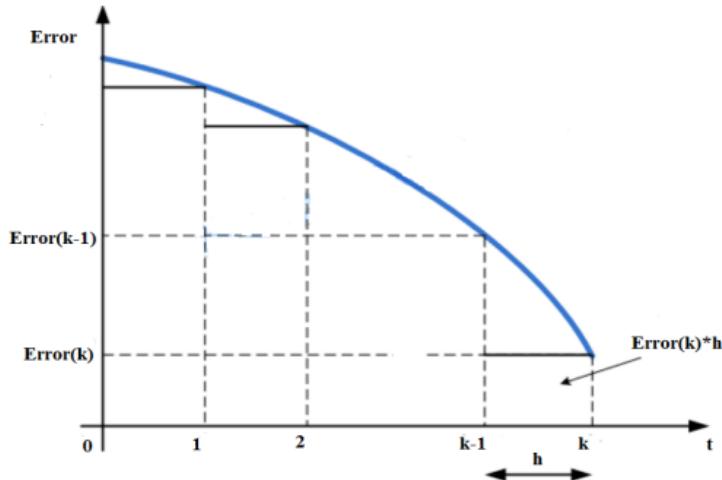
Hình 4.3 Sai số vi phân theo thời gian

$$\begin{aligned} \frac{dError}{dt} &= \frac{Error(k) - Error(k-1)}{h} \\ &= \frac{\text{TargetPosition} - \text{CurrentPosition} - (\text{TargetPosition} - \text{LastPosition})}{h} \\ &= \frac{\text{LastPosition} - \text{CurrentPosition}}{h} \\ u_2 &= K_d * \frac{dError}{dt} = K_d * \frac{\text{LastPosition} - \text{CurrentPosition}}{h} = -K_d * \Delta Position \end{aligned}$$

Trong đó: h là thời gian lấy mẫu

Tín hiệu tích phân được tính bằng tổng diện tích các hình chữ nhật tại mỗi thời điểm đang xét. Mỗi hình chữ nhật có chiều rộng bằng thời gian lấy mẫu h và chiều cao là giá trị sai số Error tại thời điểm đang xét.

[Type here]

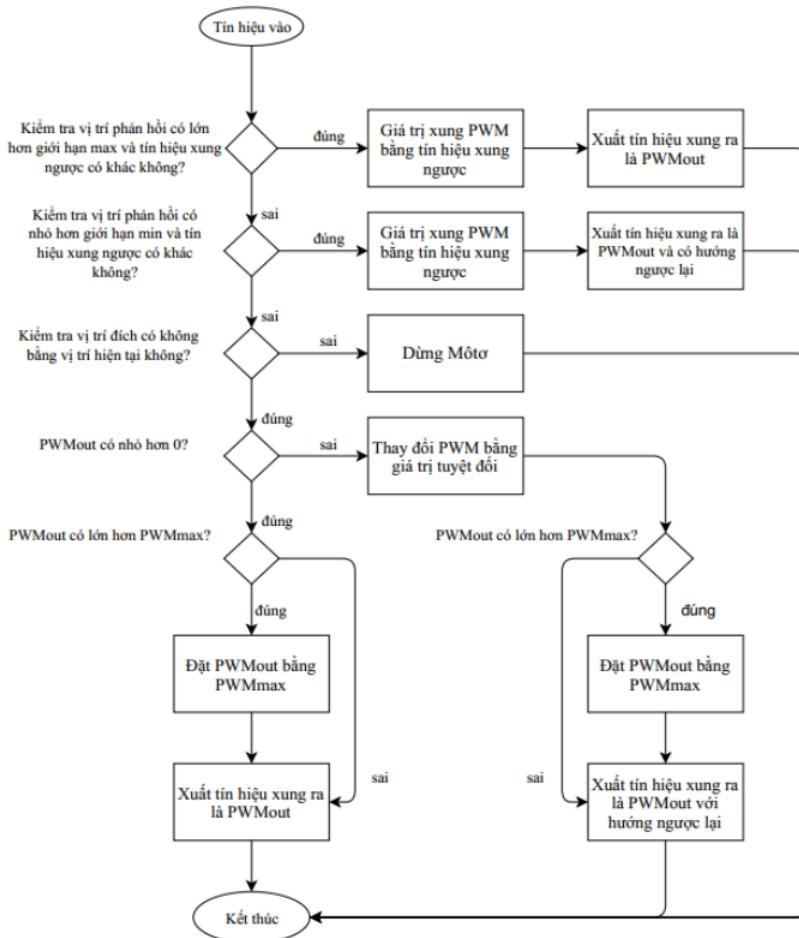


Hình 4.4 Sai số của khâu tích phân theo thời gian

$$\int_0^t Error * dt = \sum_0^k Error(k) * h = SumError$$

$$u_3 = K_i * SumError$$

$$\text{Vậy } u = K_p * Error + K_i * SumError + K_d * DeltaPosition$$



Hình 4.5 Sơ đồ khói điều khiển hướng và xung ra cho động cơ

Khi bắt đầu, SMC3 sẽ kiểm tra vị trí hiện tại của động cơ 3 có nằm trong phạm vi Clip Input max không, nếu nằm trong phạm vi này Arduino sẽ xuất ra tín hiệu điều khiển với hướng ngược lại để cản trở động cơ đến giới hạn Max Limits trên. Tương tự Arduino kiểm tra xem vị trí hiện tại của động cơ có nằm trong phạm vi Clip Input min, nếu nằm trong phạm vi này arduino sẽ xuất ra tín hiệu điều khiển để cản trở động cơ đến giới hạn Max Limits dưới. Nếu tín vị trí cần đạt được và vị trí hiện tại không bằng nhau, kiểm tra tín hiệu điều khiển có lớn hơn giá trị điều khiển lớn nhất, nếu đúng thì gán tín hiệu điều khiển bằng với tín hiệu lớn nhất. Đối với tín hiệu điều khiển nhỏ hơn 0 và giá trị tuyệt đối lớn hơn giá trị max, gán giá trị điều khiển bằng giá trị lớn nhất và xuất ra tín hiệu hướng ngược lại. Trường hợp vị trí mục tiêu và tín hiệu hiện tại bằng nhau, arduino sẽ dừng động cơ.

## 4.2 Điều khiển động cơ bằng firmware SMC3

### 4.2.1 Các thiết bị kết nối

#### 4.2.1.1. Arduino UNO R3

Arduino UNO R3 là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng. Arduino hiện nay đã được biết đến một cách rộng rãi tại Việt Nam, và trên thế giới thì nó đã quá phổ biến. Sức mạnh của chúng ngày càng được chứng tỏ theo thời gian với vô vàn các ứng dụng mở (open source) độc đáo được chia sẻ rộng rãi. Với Arduino bạn có thể ứng dụng vào những mạch đơn giản như mạch cảm biến ánh sáng bật tắt đèn, mạch điều khiển động cơ... hoặc cao hơn nữa bạn có thể làm những sản phẩm như: máy in 3D, Robot, khinh khí cầu, máy bay không người lái...



Hình 4.6 Arduino UNO R3

Mạch Arduino UNO R3 được sử dụng để nhận dữ liệu vị trí từ máy tính, kết nối qua cổng UART và điều khiển động cơ DC qua chân I/O và chân bấm xung vào mạch cầu để điều khiển động cơ và nhận phản hồi vị trí qua các chân analog. Thông số kỹ thuật của Arduino UNO R3:

- Điện áp hoạt động: 5-12V
- Dòng tiêu thụ: ~ 30mA - Số chân digital I/O: 14 (6 chân PWM)
- Số chân Analog: 6 (độ phân giải 10 bit)
- Dòng ra tối đa (5V): 500 mA
- Dòng ra tối đa (3.3V): 50mA
- Dòng tối đa mỗi chân trên I/O: 30Ma

#### *4.2.1.2. Động cơ giảm tốc MY1016Z2*

Các thông số của động cơ:

- Điện áp: 24V
- Công suất: 250W
- Vận tốc không tải: 3850 vòng/phút
- Vận tốc có tải: 3000 vòng/phút
- Hiệu suất: >78%
- Tỉ số truyền 9,78

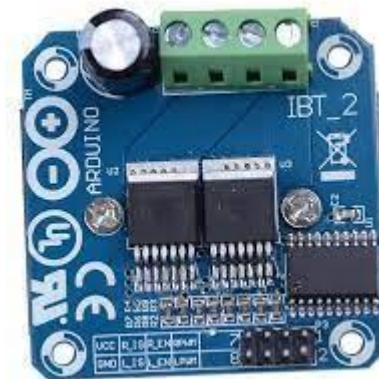
#### *4.2.1.3. Chiết áp B503-50k*

Chiết áp là biến trở có 3 chân VCC, GND và 1 chân lấy điện áp thay đổi. Chiết áp được nối với trục động cơ, khi động cơ quay làm num vặn quay, làm cho điện áp thay đổi chân Output. Điện áp này được kết nối với chân analog của Arduino để làm tín hiệu phản hồi đo vị trí của động cơ. Nếu num vặn ở vị trí 1 thì điện áp ra là thấp nhất, do điện trở là cao nhất, tương ứng với điện áp đầu Output là 0V, nếu num vặn ở vị trí 3 thì điện áp ra là cao nhất, do điện trở là thấp nhất, tương ứng với điện áp đầu ra chân Output là 5V. Khi truyền điện áp chân Output vào chân analog của Arduino sẽ biến đổi điện áp (0V ~ 5V) thành tín hiệu analog (0 ~ 1023).



Hình 4.7 Các chân của chiết áp

#### 4.2.1.4. Driver BTS7960 43A



Hình: Driver BTS7960 43A

- **Thông số kỹ thuật của Driver:**

- Nguồn cấp: 6 – 27V
- Dòng điện tối đa: 43A
- Nguồn điều khiển: 3,3-5V
- Tần số điều khiển tối đa: 25kHz
- Driver BTS7960 43A sẽ tự động ngắt khi điện áp thấp. Nếu điện áp lớn hơn 5,5V, mạch cầu sẽ tự ngắt điện và sẽ mở lại khi điện áp nhỏ hơn 5,5V

- **Chức năng các chân của Driver:**

- Chân 1: Chân đảo chiều hoặc cấp PWM cho nửa cầu H phải (RWPM)
- Chân 2: Chân đảo chiều hoặc cấp PWM cho nửa cầu H trái (LWPM)
  - + RPWM = 1 và LPWM = 0 : Mô tơ quay thuận.
  - + RPWM = 0 và LPWM = 1 : Mô tơ quay nghịch
  - + RPWM = 1 và LPWM = 1 hoặc RPWM = 0 và LPWM = 0 : Dừng.
- Chân 3(R\_EN): Chân enable nửa cầu H phải. ( R\_EN = 0 : Disable nửa cầu H phải. R\_EN=1: Enable nửa cầu H phải. )
- Chân 4(L\_EN): Chân enable nửa cầu H trái.( L\_EN = 0 : Disable nửa cầu H trái. L\_EN=1: Enable nửa cầu H trái. )
- Chân 5(R\_IS): Giới hạn dòng nửa cầu H phải
- Chân 6(L\_IS): Giới hạn dòng nửa cầu H trái
- Chân 7(VCC): 5VDC
- Chân 8(GND): Nối đất
- Chân B+: Nguồn VCC (5.5V-27V)
- Chân B- : GND
- Chân M+: Nguồn ra VCC cho động cơ
- Chân M-: Nguồn ra GND cho động cơ
- **Driver BTS7960 43A có 2 chế độ điều khiển**
- Chế độ 1:
  - R\_EN và L\_EN nối với nhau và nối với nguồn 5V
  - LPWM: tín hiệu xung PWM để điều khiển tốc độ, motor sẽ quay cùng chiều kim đồng hồ
  - RWPM: tín hiệu xung PWM để điều khiển tốc độ, motor sẽ quay ngược chiều kim đồng hồ
- Chế độ 2:
  - R\_EN và L\_EN nối với nhau và nối với tín hiệu PWM để điều khiển tốc độ

- LPWM: mức 5V động cơ sẽ chạy
- RWPM: mức 5V sẽ đảo chiều động cơ

SMC3 điều khiển động cơ với chế độ 2: RPWM,LPWM nối với GPIO để điều khiển chiều quay của động cơ. Chân R\_EN , L\_EN nối chung lại rồi nối với PWM để điều khiển tốc độ động cơ.

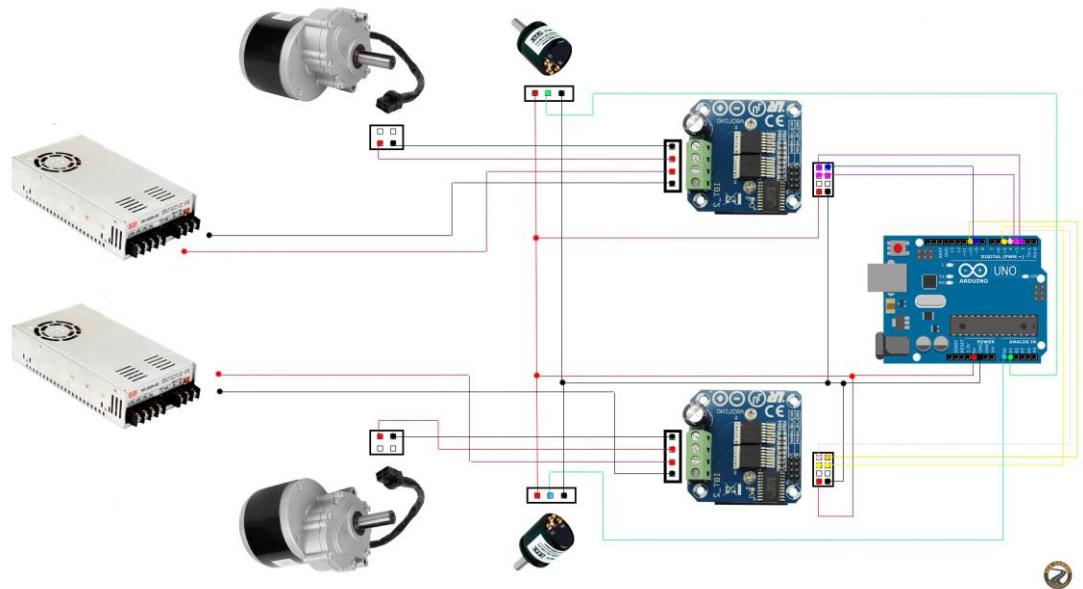
#### 4.2.1.5. Nguồn 24V 15A

- Điện áp đầu vào: 110 – 220VAC chỉnh bằng công tắc gạt.
- Công suất: 360W.
- Dòng đầu ra tối đa: 15A.
- Nhiệt độ làm việc: -10 ~ 60 độ C.

#### 4.2.2 Đầu nối Arduino UNO R3 với Driver BTS7960 43A

- Chân VCC của chiết áp và chân VCC của Driver BTS7960 nối vào nguồn 5V của Arduino UNO R3.
- Chân GND của chiết áp và chân GND của Driver BTS7960 nối vào GND của Arduino UNO R3.
- Chân Analog của chiết áp nối với động cơ 1 và 2 nối lần lượt vào các chân A0, A1 của Arduino UNO R3
- Chân RWPM của Driver động cơ 1 nối với chân 6 của Arduino UNO R3
- Chân LWPM của Driver động cơ 1 nối với chân 9 của Arduino UNO R3
- Chân R\_EN và L\_EN của Driver động cơ 1 nối với chân 4 của Arduino UNO R3
- Chân RWPM của Driver động cơ 2 nối với chân 3 của Arduino UNO R3
- Chân LWPM của Driver động cơ 2 nối với chân 10 của Arduino UNO R3
- Chân R\_EN và L\_EN của Driver động cơ 2 nối với chân 2 của Arduino UNO R3

[Type here]



Hình 4.8 Sơ đồ đấu nối điều khiển 2 động cơ

#### 4.2.3 Điều chỉnh PID sử dụng phần mềm SMC3 Utility

##### 4.2.3.6. Phần mềm SMC3 Windows Utility

SMC3 Windows Utility là một phần mềm chuyên dụng cài đặt cho firmware SMC3. SMC3 Windows Utility giúp kiểm tra chiều quay của từng động cơ để lắp đặt cho đúng, điều chỉnh các thông số Kp, Ki, Kd cho bộ điều khiển PID. Sau đây là các thông số chi tiết của phần mềm:



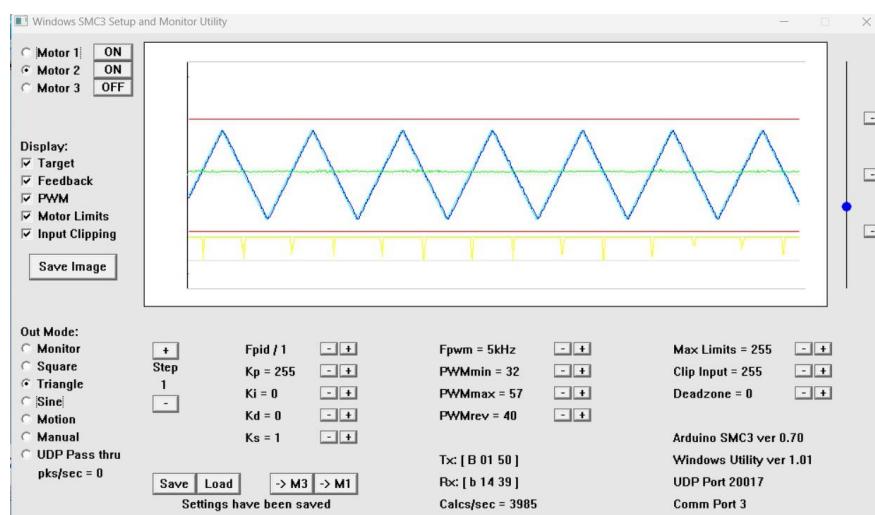
Hình 4.9 Giao diện SMC3 Windows Utility

- Motor 1, Motor 2, Motor 3 nằm trong giới hạn điều khiển thì tín hiệu là ON, nằm ngoài giới hạn điều khiển là OFF. Giới hạn điều khiển này có giá trị từ Max Limits đến 1023 - Max Limits.
- **Target**: Vị trí đích động cơ cần tiến tới.
- **Feedback**: Giá trị phản hồi của chiết áp.
- **PWM**: Giá trị xung PWM.
- **Motor Limits**: có giá trị nhỏ hơn Max Limits và lớn hơn 1023 – Max Limits động cơ sẽ tắt.
- **Input Clipping**: có giá trị nhỏ hơn Clip Input và lớn hơn 1023 – Clip Input

động cơ được cấp xung có giá trị ngược lại với  $\text{PWM} = \text{PWMrev}$

Các chế độ đầu ra trong SMC3 Windows Utility:

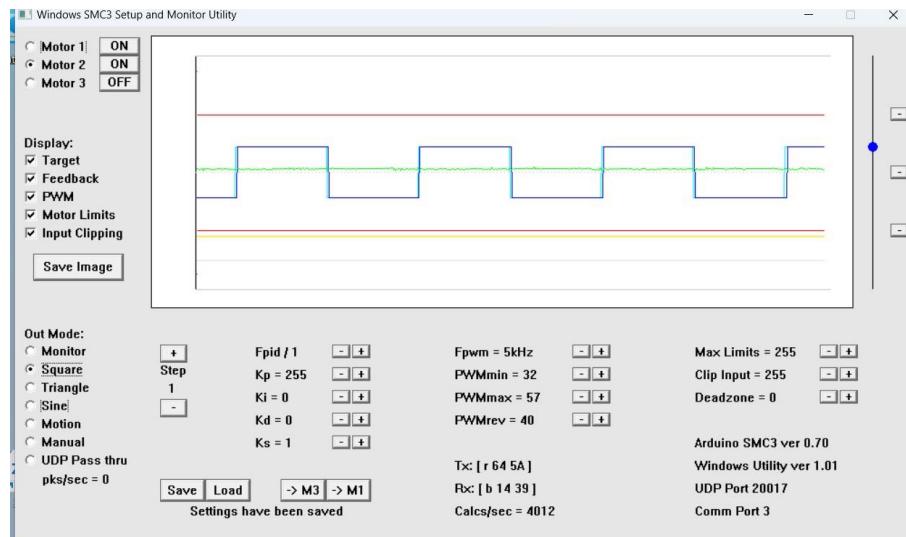
- **Monitor**: Chế độ đầu ra dạng đường thẳng.
- **Triangle**: Chế độ đầu ra dạng xung tam giác.



Hình 4.10 Chế độ xung ra Triangle

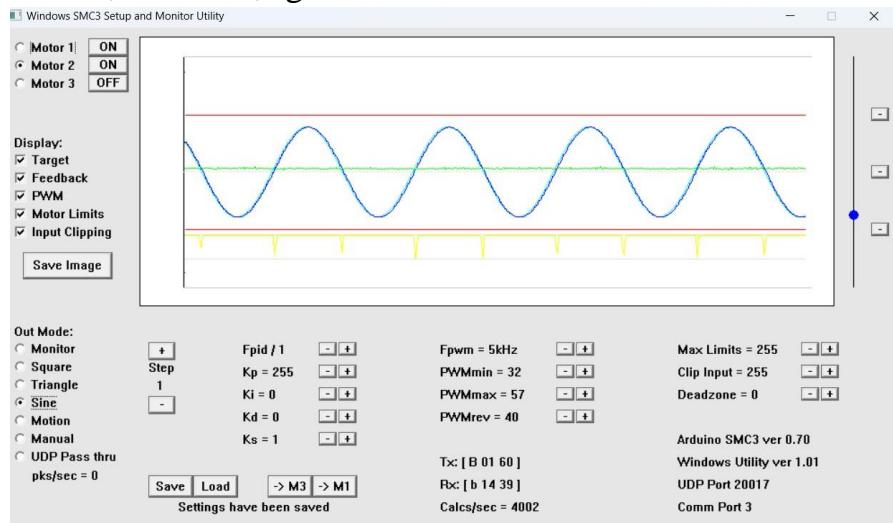
- **Square**: Chế độ đầu ra dạng xung vuông

[Type here]



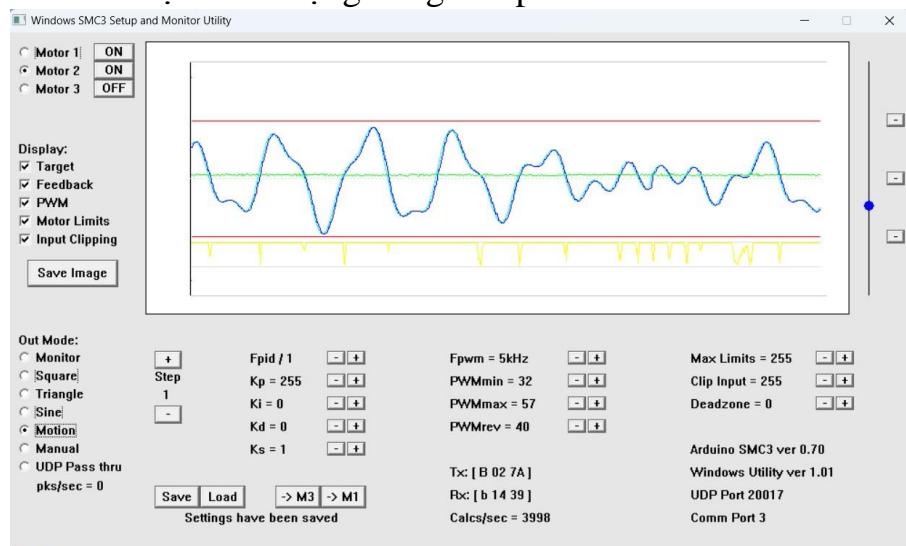
Hình 4.11 Chế độ đầu ra dạng Square

- **Sine:** Chế độ đầu ra dạng hình sin



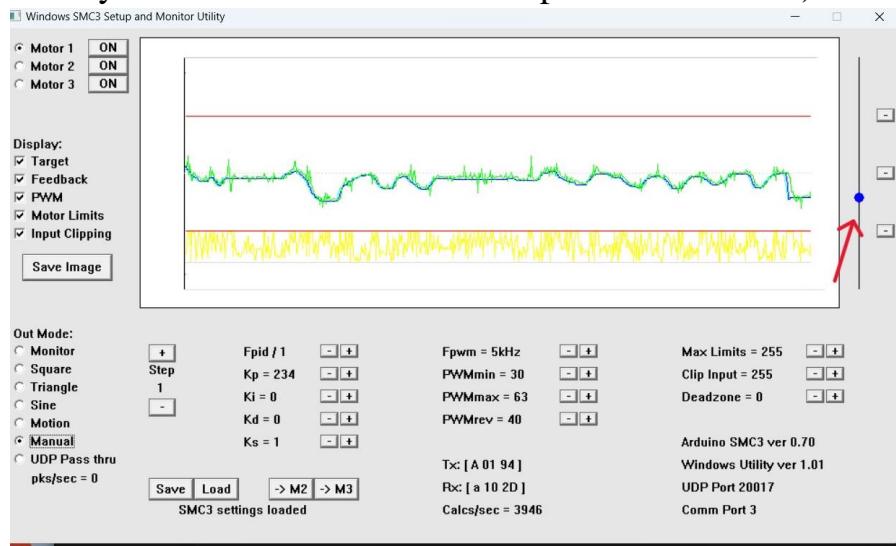
Hình 4.12 Chế độ đầu ra dạng Sine

- **Motion:** Chế độ đầu ra dạng xung của phần mềm đưa ra



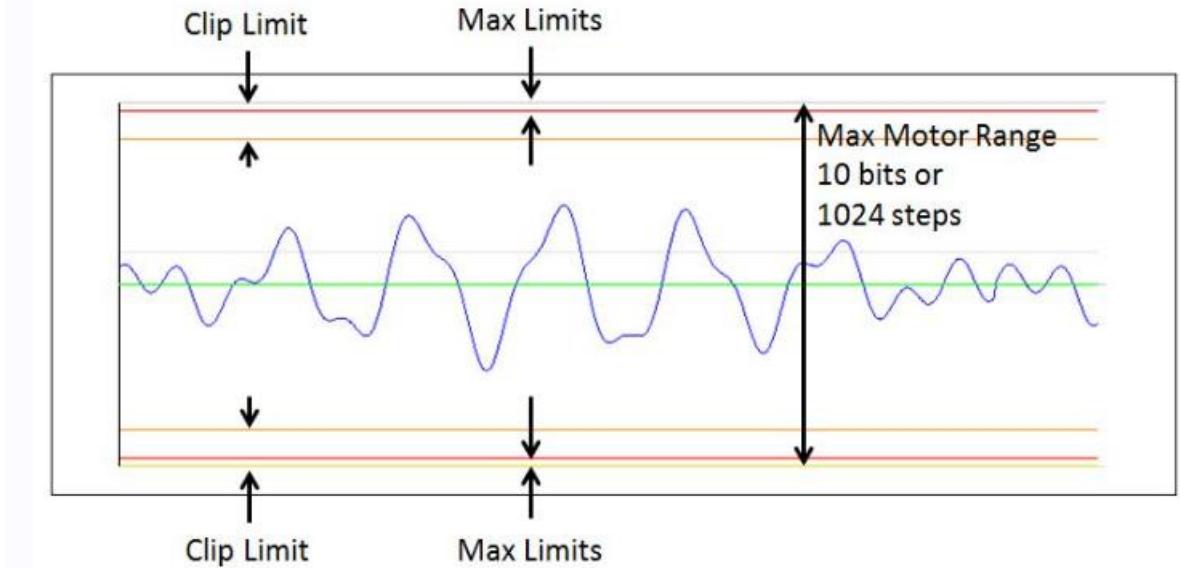
Hình 4.13 Chế độ đầu ra Motion

- **Manual:** Chế độ đầu ra dạng xung do người dùng tự tạo. ( Tạo bằng cách di chuyển chấm tròn màu xanh bên phải như hình vẽ )



Hình 4.14 Chế độ đầu ra Manual

- **Clip Input:** thông số này cài đặt hai giá trị giới hạn mà mọi giá trị vị trí vượt quá giới hạn này được gửi tới SMC3 sẽ bị cắt bỏ. Ngoài ra, nếu phản hồi từ chiết áp vượt quá phạm vi Clip Input thì SMC3 sẽ hãm động cơ bằng cách đẩy ngược động cơ lùi ra khỏi giá trị giới hạn. Đây ngược động cơ được áp dụng với các giá trị vị trí nằm giữa Clip Input và Max Limits. Giá trị Clip Input được cài đặt trong phạm vi từ 0 đến 255. Ví dụ, đặt Clip Input bằng 23, thì những giá trị nhỏ hơn 23 và lớn hơn 1000 (1023 - 23) sẽ bị cắt SMC3 cắt bỏ.
- **Max Limits:** thông số này cài đặt hai giá trị giới hạn mà mọi giá trị vị trí vượt quá giới hạn này (thường là do quán tính) được gửi tới SMC3 sẽ tự động tắt trình điều khiển cho đến khi được đặt lại. Đây thực chất là một cơ chế an toàn nếu có sự cố. Giá trị cài đặt nằm trong phạm vi 0-255 (tuy nhiên không thể lớn hơn cài đặt Clip Input).



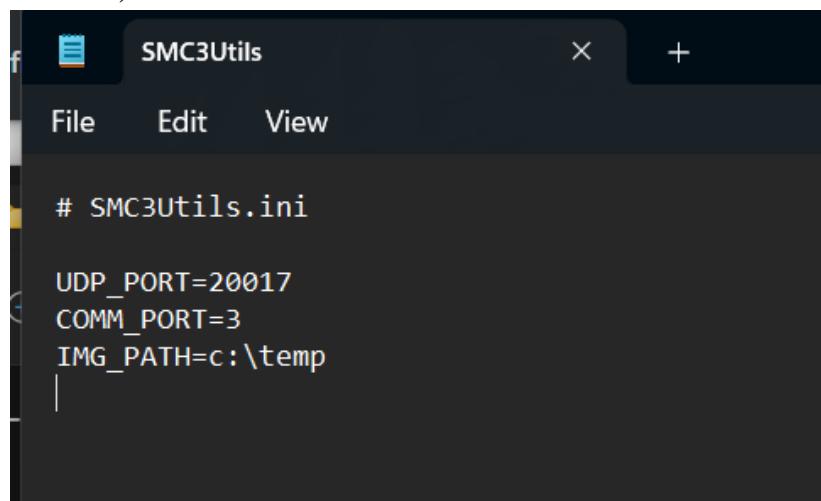
Hình 4.15 Clip Limit và Max Limits trong SMC3 Utility

- **Deadzone:** thông số này tạo vùng trễ cho phản hồi của động cơ. Thông số này càng cần lớn thì trong thiết lập cơ khí có độ trễ càng cao.
- **Fpwm:** Tần số của Motor
- **PWMmin:** Chu kỳ xung tối thiểu được sử dụng để điều khiển các động cơ đến vị trí đích đến. Ví dụ: nếu các vị trí đã ở gần đích, động cơ sẽ không cần phải được điều khiển nhanh như chúng ở xa đích. Lý do chúng ta không sử dụng 0 là do ma sát, tải trọng và các yếu tố khác, động cơ sẽ cần một dòng điện tối thiểu nhất định chỉ để chuyển động. Giá trị cài đặt nằm trong phạm vi 0-255 tuy nhiên không thể lớn hơn cài đặt PWMmax.
- **PWMmax:** Chu kỳ xung tối đa được sử dụng để điều khiển các động cơ đến vị trí đích. Khi vị trí đích đến càng xa vị trí hiện tại, SMC3 sẽ điều khiển tăng tốc độ động cơ để đưa nhanh đến vị trí đích. Chu kỳ xung PWM sẽ tăng cho đến khi đạt được mức tối đa PWMmax.
- **PWMrev:** Chu kỳ xung được sử dụng để điều khiển động cơ đảo ngược lại nếu vượt quá giới hạn Clip Input. Thông số này có thể gây giật đột ngột khi thay đổi hướng. Nếu muốn vô hiệu hóa đảo ngược động cơ thì đặt thông số này bằng 0. Giá trị cài đặt nằm trong phạm vi 0-255.
- **Fpid:** Theo mặc định, vòng lặp PID thực hiện 4000 cập nhật tính toán mỗi giây. Bộ chia này cho phép giảm số lượng tính toán PID mỗi giây bằng số lượng được chia.
- **K<sub>p</sub>:** Hệ số tỷ lệ của vòng lặp PID
- **K<sub>i</sub>** Hệ số tích phân của vòng lặp PID.

- **K<sub>d</sub>**: Hệ số vi phân của vòng lặp PID  
(Việc hiệu chỉnh các thông số K<sub>p</sub>, K<sub>i</sub>, K<sub>d</sub> sẽ được đề cập ở phần 4.2.3b)
- **K<sub>s</sub>**: Đây là hệ số cập nhật cho K<sub>d</sub>. Ví dụ: K<sub>s</sub> bằng 5 có nghĩa là K<sub>d</sub> sẽ chỉ được cập nhật một lần sau mỗi 5 lần các tham số khác được tính toán.

#### 4.2.3.7. Các bước hiệu chỉnh thông số SMC3 trong phần mềm SMC3 Windows Utility

- Bước 1: Ngắt kết nối nguồn điện động cơ.
- Bước 2: Turn off Simtool.
- Bước 3: Kết nối Arduino (đã cài đặt SMC3) với máy tính qua cổng USB
- Bước 4: Chạy phần mềm SMC3 Windows Utility (Lưu ý cần setup đúng cổng kết nối qua dòng lệnh COMM\_PORT trong file SMC3Utils.ini )



Hình 4.16 Setup cổng kết nối cho SMC3 Utility

- Bước 5: Đặt tất cả các thông số K<sub>p</sub>, K<sub>i</sub>, K<sub>d</sub>, PWMmin, PWMmax, PWMrev về 0 cho tất cả động cơ để đảm bảo các động cơ không di chuyển.
- Bước 6: Đặt Max Limits và Clip Input bằng 255 (giúp cho người dùng có nhiều thời gian nếu có vấn đề xảy ra trong quá trình cài đặt)
- Bước 7: Bật nguồn cho động cơ (lúc này động cơ chưa chạy)
- Bước 8: Đặt K<sub>p</sub> bằng 400
- Bước 9: Tăng từ từ PWMmax đến một thời điểm động cơ bắt đầu quay. Nếu phản hồi Feedback di chuyển về Target thì động cơ và dây nối là chính xác - tiến hành kiểm tra các động cơ khác. Nếu động cơ quay

ngay thì tắt nguồn động cơ ngay lập tức (hoặc nhanh chóng giảm trở lại PWMmax). Trong trường hợp này cần đảo ngược dây dẫn động cơ.

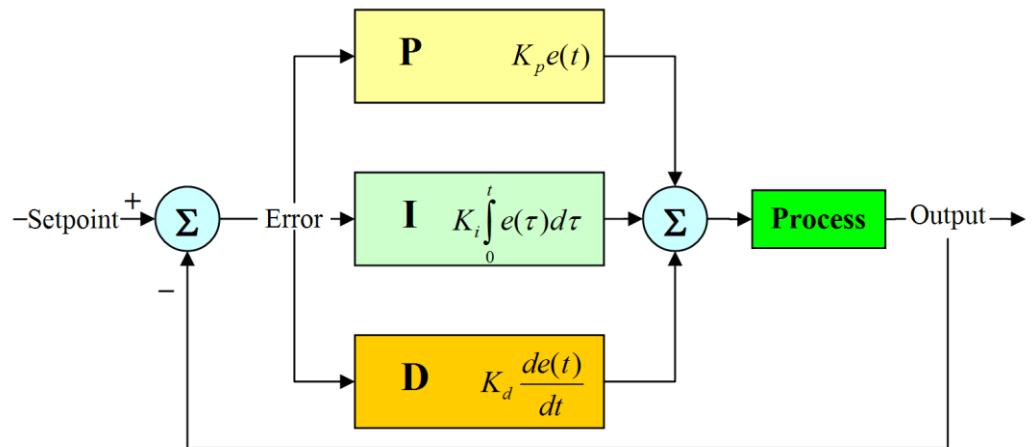
- Bước 10: Điều chỉnh các thông số Kp, Ki, Kd của bộ điều khiển PID để hệ thống chạy ổn định.

#### 4.2.3.8. Bộ điều khiển PID

Mặc dù những tiến bộ của lý thuyết và các phương pháp thiết kế đang liên tục phát triển trong lĩnh vực điều khiển tự động, các bộ điều khiển Tỉ lệ-Tích phân-Vi phân (PID - Proportional, Integral, Derivative control) vẫn được sử dụng rộng rãi nhất trong công nghiệp do ưu điểm về tỉ số giá thành và lợi nhuận của chúng đem lại. Thực tế, mặc dù chúng tương đối đơn giản trong việc sử dụng, nhưng vẫn có khả năng thỏa mãn về thực thi trong nhiều tác vụ điều khiển quá trình. Thực ra quá trình lịch sử lâu dài và các kiến thức được đúc lại qua nhiều năm đã làm cho cách sử dụng chúng như một bộ điều khiển phản hồi chuẩn mực. Tuy nhiên khả năng đáp ứng cao của các bộ vi xử lý, các công cụ phần mềm và sự tăng yêu cầu nâng cao chất lượng sản phẩm đồng thời giảm giá thành sản phẩm đã thúc đẩy các nhà nghiên cứu phát minh ra các phương pháp mới cho việc cải tiến khả năng hoạt động và sự đơn giản trong sử dụng các bộ điều khiển.

Các bộ điều khiển PID đã được phát triển trong quá trình phát triển công nghệ, và ngày nay nó được hiện thực rất phổ biến dưới dạng số với các phần tử điện hoặc khí nén,... Nó có thể thấy trong hầu hết các loại thiết bị điều khiển như một bộ điều khiển đơn - độc lập hoặc dưới dạng các khối chức năng trong các bộ điều khiển PLC (Programmable Logic Controller) hoặc DCS (Distributed Control System).

Các bộ điều khiển PID là công cụ chuẩn dùng cho tự động hóa công nghiệp. Sự linh hoạt của bộ điều khiển tạo cho nó khả năng sử dụng trong nhiều trường hợp. Các bộ điều khiển cũng có thể được sử dụng trong điều khiển tầng và các cấu hình bộ điều khiển khác. Nhiều vấn đề điều khiển đơn giản có thể được giải quyết rất tốt bởi điều khiển PID - với những yêu cầu chất lượng không quá cao. Thuật toán PID được đóng gói trong các bộ điều khiển tiêu chuẩn cho điều khiển quá trình và cũng là cơ sở cho nhiều hệ thống điều khiển đơn giản.



Hình 4.17: Bộ điều khiển PID

Các tham số của bộ điều khiển PID:

- $K_p$ : Hệ số tỉ lệ. Khi  $K_p$  càng lớn sẽ giúp hệ thống đáp ứng càng nhanh nhưng lại gây ra sai số xác lập lớn. Vì vậy nếu  $K_p$  quá lớn sẽ gây ra vọt lố, hệ thống mất ổn định.
- $K_i$ : Hệ số tích phân.  $K_i$  càng lớn sẽ giúp sai số xác lập càng nhỏ nhưng nếu  $K_i$  quá lớn sẽ gây ra vọt lố, mất ổn định.
- $K_d$ : Hệ số vi phân.  $K_d$  lớn sẽ làm giảm độ vọt lố nhưng lại làm hệ thống đáp ứng chậm.

## CHƯƠNG 5. ĐIỀU KHIỂN HỆ THỐNG MÔ PHỎNG LÁI QUA SIMTOOLS

### 5.1 Giới thiệu về Simtools

#### 5.1.1 Khái niệm

SimTools là một phần mềm mô phỏng chuyển động chung có thể điều khiển nhiều giao diện phần cứng như bộ truyền động Arduino, SimAxe, JRK, SCN5.

Phần mềm này là liên kết kết nối giữa trò chơi và trình mô phỏng chuyển động. Nó trích xuất các giá trị trò chơi như lực tốc độ, hướng chuyển động, chuyển số, tốc độ quay của động cơ, lực tăng tốc và nhiều hơn nữa. Sau đó, tất cả các giá trị này sẽ được trộn vào một tập dữ liệu được sử dụng để điều khiển các động cơ của trình mô phỏng chuyển động.



Hình 5.1 Ứng dụng Live for Speed

Có những sản phẩm phần mềm khác như SimTools trên thị trường(BFF, X-Sim), nhưng SimTools sở hữu một số ưu điểm:

- Rất dễ sử dụng
- Cộng đồng hỗ trợ lớn.
- Hỗ trợ hơn 95% các trò chơi đua xe và lái máy bay phổ biến nhất hiện nay.
- Hỗ trợ đa dạng các bộ truyền động, nhiều DOF,...

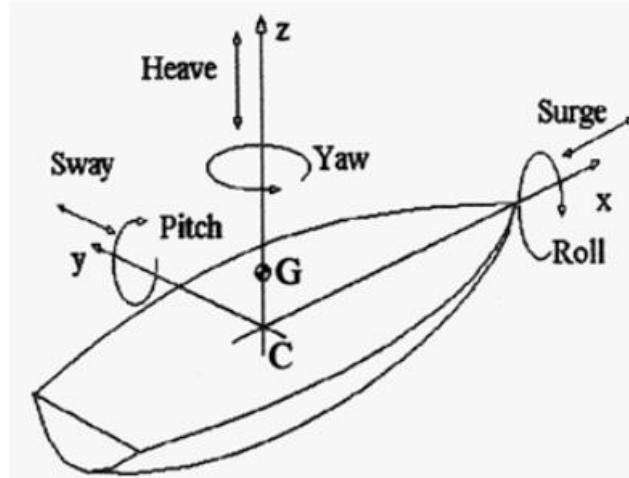
Bộ công cụ simtools bao gồm 3 tệp thực thi chính:

- Game PluginUpdater: cài đặt plugin trò chơi.
- Game Manager: cài đặt đường dẫn kết nối với game, cài đặt trích xuất dữ liệu từ game, Ip và cổng UDP để Game Plugin gửi dữ liệu.

- Game Engine: Cài đặt trực, tạo bộ lọc, cài đặt giao diện giao tiếp(NET,SCN,SER...), chạy thử kiểm tra trực.

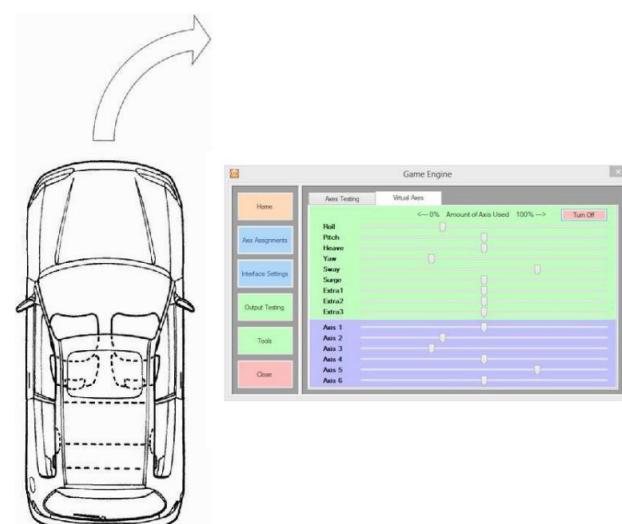
### 5.1.2 Hướng di chuyển của xe trong Simtools

Mỗi chuyển động của xe được phân tích thành một hoặc nhiều chuyển động cơ bản.



Hình 5.2 Sáu bậc tự do trong chuyển động

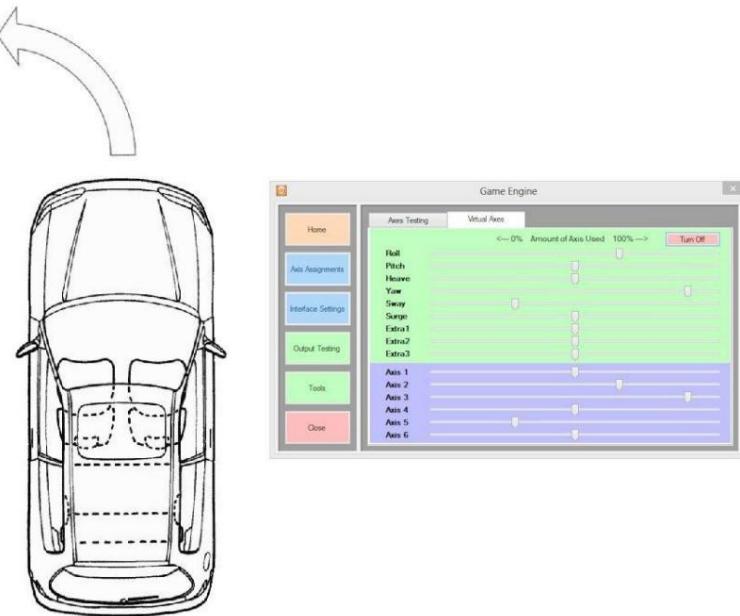
- Tăng tốc – Thanh Pitch chuyển động sang trái, thanh Surge chuyển động sang phải.
- Giảm tốc (phanh) – Thanh Pitch chuyển động sang phải, thanh Surge chuyển động sang trái.
- Rẽ phải – Thanh Roll di chuyển sang trái, thanh Sway di chuyển sang phải và thanh Yaw di chuyển sang trái



Hình 5.3 Xe rẽ phải

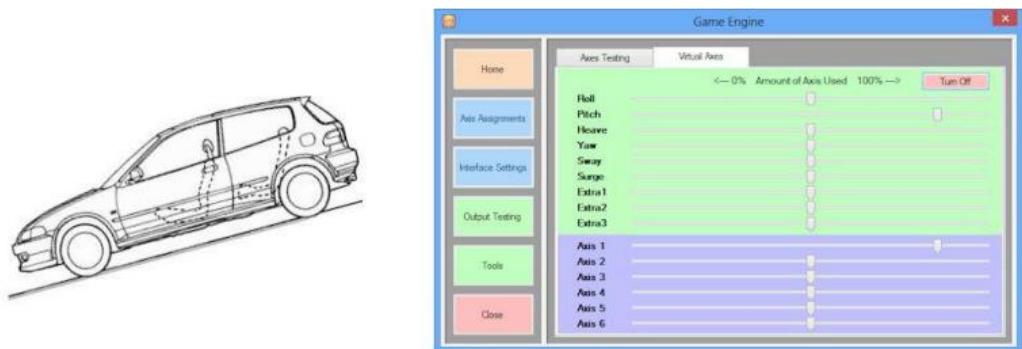
[Type here]

- Rẽ trái – Thanh Roll di chuyển sang phải, thanh Sway di chuyển sang trái và thanh Yaw di chuyển sang phải.



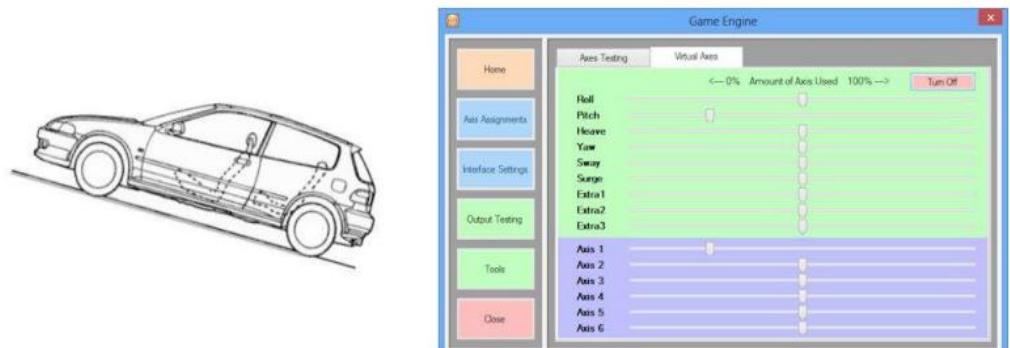
Hình 5.4 Xe rẽ trái

- Xuống dốc – Thanh Pitch di chuyển sang phải.



Hình 5.5 Xe xuống dốc

- Lên dốc – Thanh Pitch di chuyển sang trái.



Hình 5.6 Xe lên dốc

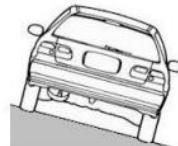
[Type here]

- Xe nghiêng bên trái – Thanh roll di chuyển sang trái.



Hình 5.7 Xe nghiêng bên trái

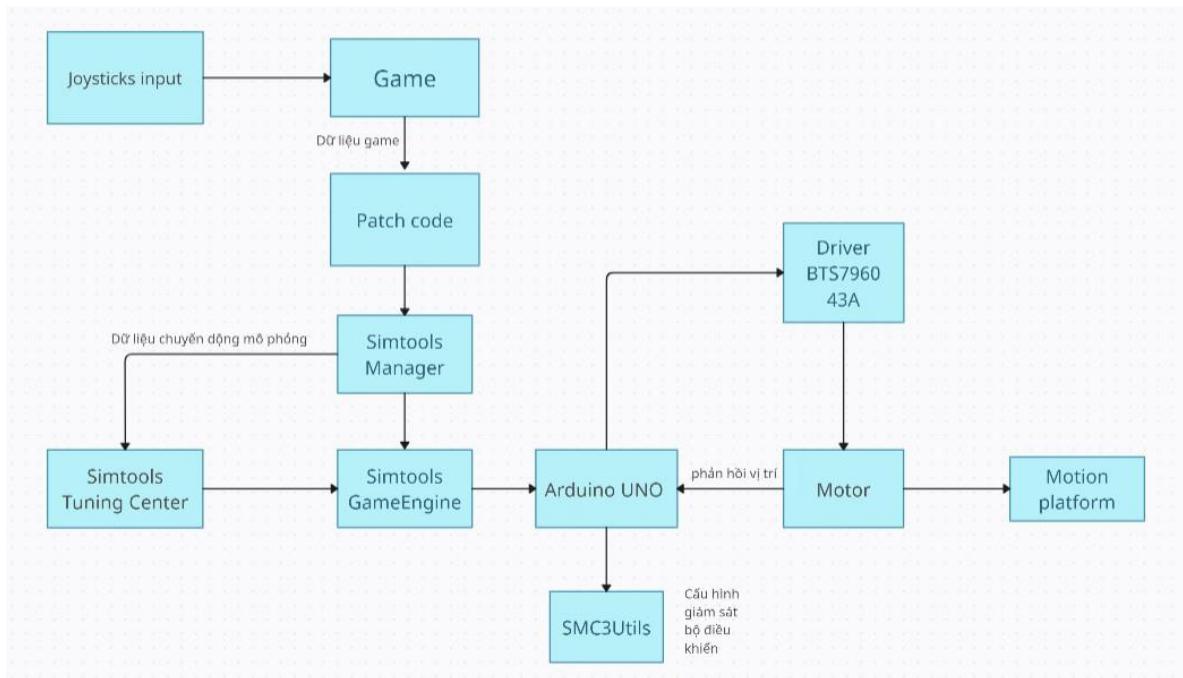
- Xe nghiêng bên phải – Thanh Roll di chuyển sang phải.



Hình 5.8 Xe nghiêng bên phải

Đồ án này thiết kế về hệ mô phỏng lái 2 bậc tự do nên chỉ mô phỏng chuyển động cho 2 chuyển động Roll và Pitch, còn 4 chuyển động Heave, Yaw, Sway, Surge sẽ được bỏ qua.

## 5.2 Sơ đồ khái quát của mô hình mô phỏng lái xe



Hình 5.9 Sơ đồ kết nối và điều khiển hệ thống mô phỏng lái

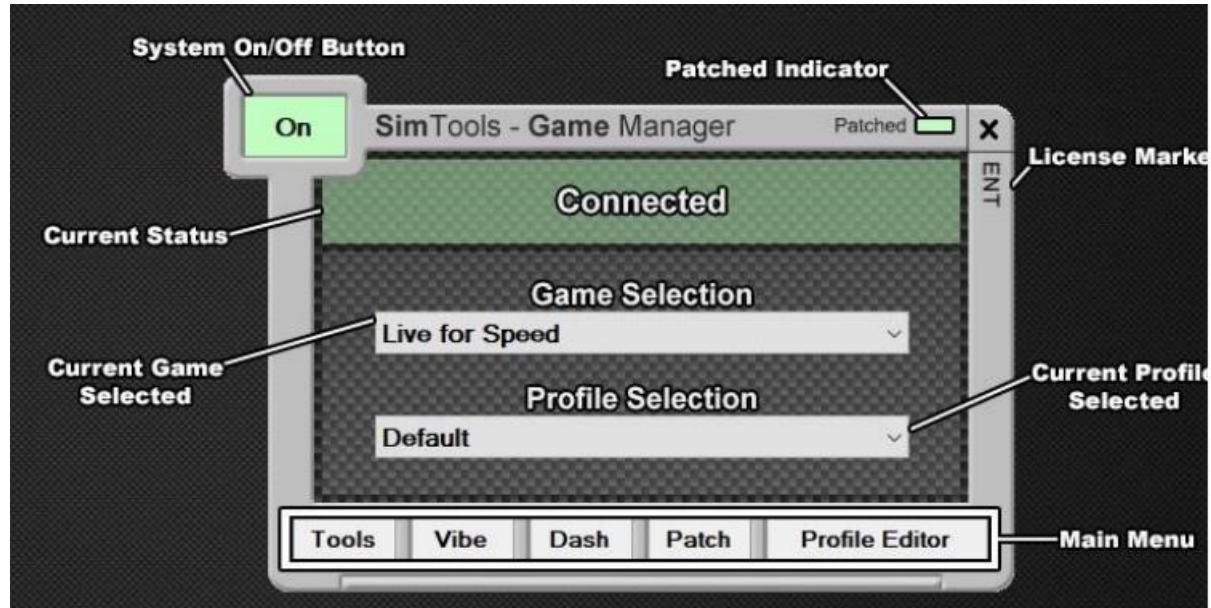
Trình quản lý trò chơi Simtools GameManager sẽ sử dụng các gói hỗ trợ trong thư viện GamePlugin để phân tích dữ liệu từ game thành các dữ liệu mô phỏng chuyển động (Roll, Pitch, Yaw, Sway, Surge, Heave,...) và được gửi tới công cụ Simtools GameEngine. Tại đây dữ liệu mô phỏng được so sánh với giá trị max và min (được điều chỉnh trong Tuning Center, sẽ được đề cập ở phần sau) để đảm bảo rằng dữ liệu chuyển động được xuất ra cho bộ mô phỏng là tốt nhất có thể. Sau đó dữ liệu này sẽ được gửi qua cổng serial cho Arduino. Arduino được nạp sẵn firmware SMC3 – bộ điều khiển chuyển động cho 3 động cơ. Nó sử dụng vòng điều khiển PID để điều khiển động cơ qua Driver BTS7960.

## 5.3 Trình quản lý trò chơi Simtools GameManager

Simtools GameManager phần quản lý các chương trình game được kết nối với phần mềm mô phỏng. Khi một trong số các game đã được kết nối với phần mềm mô phỏng thì nó có nhiệm vụ kết nối game với phần game engine để thực hiện truyền nhận dữ liệu mô phỏng. Simtools GameManager có ba trạng thái được báo trên màn hình

- **Not connect:** trạng thái báo hai phần Simtools GameManager và Simtools GameEngine không đợc kết nối với nhau.
- **Connected:** trạng thái báo hai phần của simtool đã được kết nối với nhau, sẵn sàng cho việc chạy chương trình game.

- **Game running:** trạng thái báo Simtools kết nối thành công với chương trình game và đang gửi dữ liệu cần mô phỏng tới Simtools GameEngine.



Hình 5.10 Giao diện Simtools GameManager

Để có thể thực hiện được quá trình mô phỏng, cần phải cài đặt game đua xe cho phép liên kết với phần mềm Simtools và kết nối với phần mềm. Phần mềm Simtools là một phần mềm mô phỏng đa năng, có thể mô phỏng được rất nhiều các bậc tự do và rất nhiều chuyển động trong tự nhiên vì thế nó cũng có thể liên kết với rất nhiều các ứng dụng khác nhau. Trong lĩnh vực mô phỏng lái xe, Simtools cho phép kết nối với khoảng 20 ứng dụng trong đó có thể kể tới các ứng dụng thu hút nhiều người quan tâm như Race 07, Race room, Live for speed, Dirt . . . Trong phần này, nhóm đồ án sẽ sử dụng ứng Live for speed để phục vụ cho đề tài.

Các bước liên kết game (Live for Speed) cho Simtools GameManager:

- Bước 1: Chọn mục “Patch” trên giao diện GameManager
- Bước 2: Chọn “Patch Game”

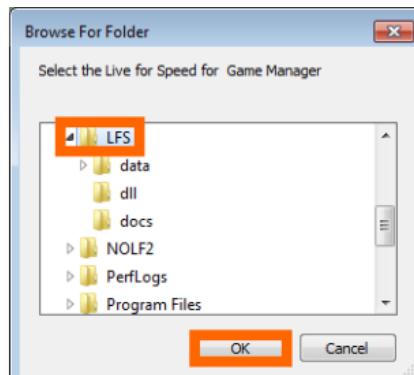


- Bước 3: Nhấn OK

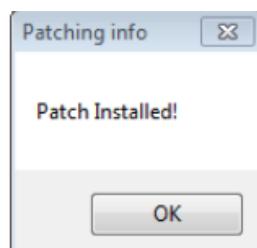
[Type here]



- Bước 4: Chọn thư mục chứa dữ liệu game Live For Speed đã cài đặt (Ví dụ C:\LFS)



- Bước 5: Khi cửa sổ Patch Installed hiện ra, nhấn OK để hoàn thành cài đặt.



## 5.4 Công cụ Simtools GameEngine

### 5.4.1 Thiết lập giao diện kết nối (Interface Settings)

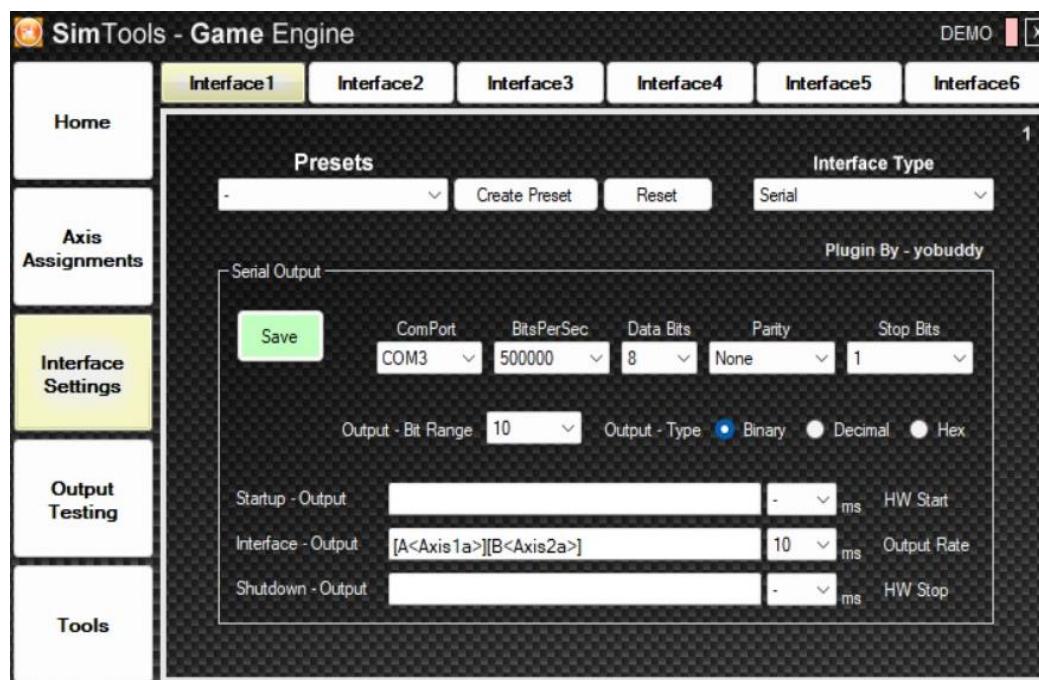
Interface Settings là phần giao diện cài đặt các thông số truyền nhận giữ liệu, giao tiếp giữa phần cứng và phần mềm. Phần mềm Simtools GameEngine có thể lưu được nhiều giao diện giao tiếp cho nhiều mục đích mô phỏng khác nhau, nhưng nhìn chung các giao diện giao tiếp đều bao gồm các thông số giống nhau, cần thiết cho việc giao tiếp, truyền nhận dữ liệu giữa chương trình game và phần cứng.

Simtools hỗ trợ nhiều giao thức giao tiếp khác nhau. Các loại giao thức có thể chọn là Net - Giao thức cần được kết nối qua Địa chỉ IP, SCN - để sử dụng bộ truyền động tuyến tính servo SCNS / SCN6, cuối cùng là tất cả các thiết bị được kết nối Ser- serial. Trong đồ án chọn giao thức giao tiếp Serial – một giao thức giao tiếp theo từng chuỗi dữ liệu để giao tiếp với Arduino.

Cài đặt các thông số khác:

- ComPort: chọn cổng máy tính đang kết nối với Simtools

- BitsPerSec: tốc độ truyền nhận số bit truyền nhận trên một giây
- DataBits: Số bit dữ liệu trong một khung dữ liệu. Thường dc set là 8 bit
- Parity: Chọn bit chẵn lẻ. Đây là phương pháp phát hiện lỗi trong quá trình truyền dữ liệu (Thường chọn là None).
- Stop Bits:1
- Output – Bit Range: 10
- Interface Output: Nhập định dạng đầu ra. Trong Simtools, mỗi cơ cấu chấp hành(động cơ) được xem là 1 trục (axis) và có thể xử lý tối đa 6 bộ truyền động (“Axis1a” đến “Axis6a” )



Hình 5.11 Cài đặt các thông số ở giao diện Interface Settings trong Simtools

#### 5.4.2 Thiết lập trực giả lập (Axis Assignments)

Axis Assignments: là phần giao diện thiết lập thông số đầu vào, phân chia các trục, chọn trục, các bậc tự do cần mô phỏng.

Việc thiết lập trực giả lập cho phép người dùng cài đặt cấu hình kết hợp các lực cho mỗi trục để tạo ra chuyển động hợp lí cho hệ thống mô phỏng.

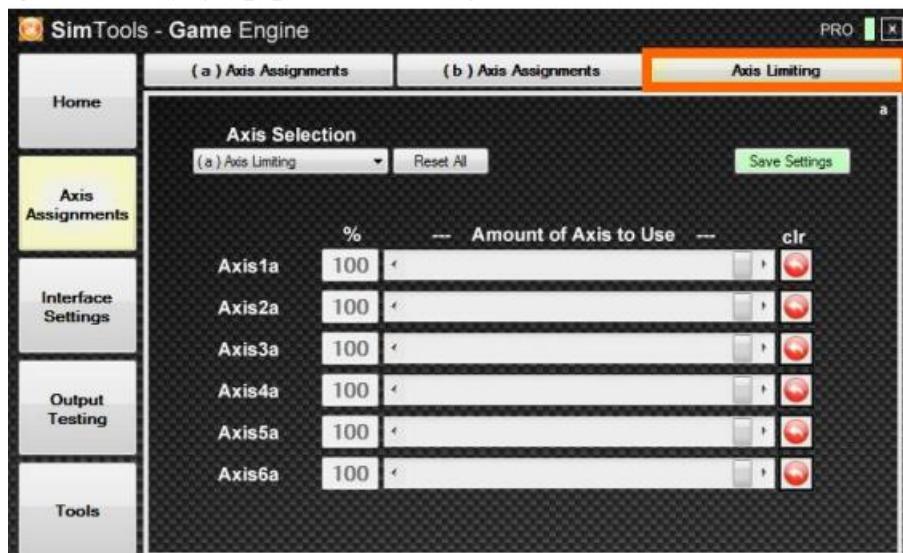
- Dir: Hướng mà lực được tác dụng
- Force: Các chuyển động cơ bản(Roll, Pitch, Yaw,.....)
- %: Số phần trăm một trục sẽ mô phỏng một bậc tự do so với tín hiệu đầu vào của bậc tự do đó

[Type here]



Hình 5.12 Thiết lập trực giả lập cho Simtools GameEngine

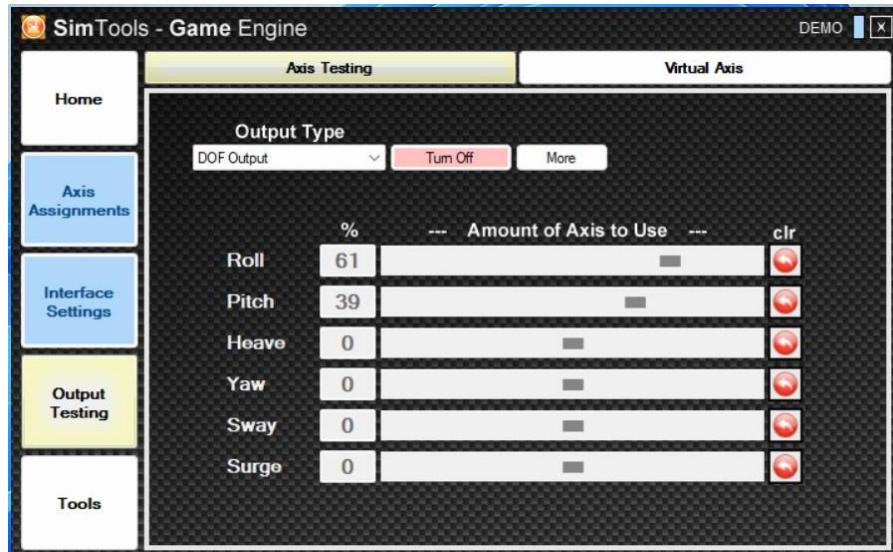
**Axis Limiting:** cho phép người dùng chọn tỉ lệ phần trăm của một trục để sử dụng. Điều này giúp phần cài đặt giới hạn của trục khi điều khiển.



Hình 5.13 Cài đặt Axis Limiting

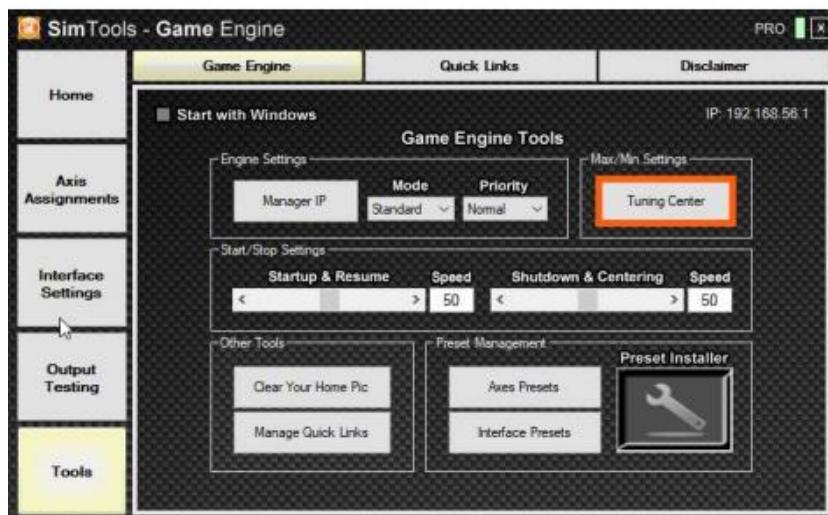
Sau khi thiết lập xong Axis Assignments, có thể kiểm tra đầu ra ở mục Output Testing. Khi bật chế độ lên, chương trình cho phép ta giả tạo tín hiệu đầu vào và kiểm tra tín hiệu ra trên phần cứng. Kiểm tra xem các động cơ đã quay đúng chiều hay không, nếu chưa đúng chiều cần đổi lại thông số Dir trong Axis Assignment.

[Type here]



Hình 5.14 Output Testing

### 5.4.3 Trung tâm điều chỉnh Simtools Tuning Center



Hình 5.15 Menu Tools trong Simtools GameEngine

Sử dụng nó sau khi hoàn tất thiết lập cơ bản của Trình quản lý trò chơi và Công cụ trò chơi. Điều chỉnh các giá trị (Max/min) của trò chơi cho trình giả lập để đảm bảo rằng chuyển động xuất ra cho trình giả lập là tốt nhất có thể.

Việc tăng các giá trị Min / Max lên số cao hơn sẽ dẫn đến bộ mô phỏng ít chuyển động hơn, nghĩa là các lực mô phỏng sẽ hiếm khi đạt đến giá trị tối đa hoặc tối thiểu của nó dẫn đến chuyển động tổng thể ít hơn nhưng sẽ có phạm vi hoặc đầu ra nhiều hơn.

Giảm các giá trị Min / Max dẫn đến bộ mô phỏng có thể đạt được các cài đặt Min / Max đó nhanh hơn rất nhiều dẫn đến chuyển động rõ nét hơn và nhanh hơn nhưng với phạm vi tổng thể ít hơn cho đầu ra.

[Type here]



Hình 5.16 Simtools Tuning Center

## CHƯƠNG 6. THỰC THI VÀ THỬ NGHIỆM HỆ THỐNG MÔ PHỎNG LÁI 2 BẬC TỰ DO

### 6.1 Thực thi, chế tạo mô hình mô phỏng lái 2 bậc tự do

Dựa trên quá trình đã nghiên cứu của mình, nhóm đã tiến hành chế tạo hệ thống mô phỏng lái 2 bậc tự do và thu được các kết quả, sản phẩm như sau:



Hình 6.1 Sản phẩm mô hình mô phỏng lái 2 bậc tự do

Xây dựng được bộ điều khiển mô hình:

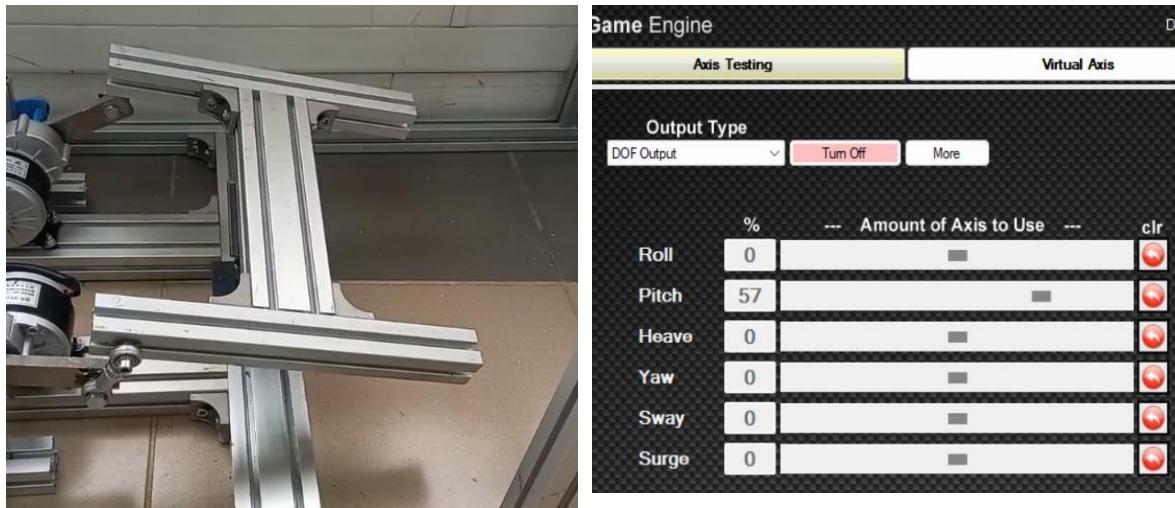


Hình 6.2 Bộ điều khiển mô hình

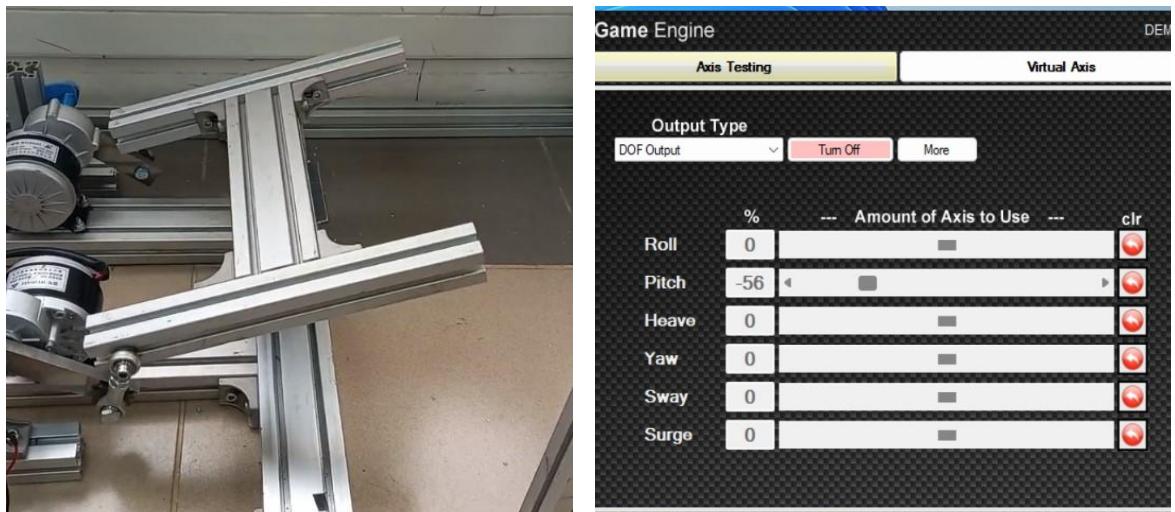
## 6.2 Thử nghiệm mô hình

Sau khi xây dựng xong mô hình phần cứng và kết nối với phần mềm mô phỏng, nhóm đã tiến hành thử nghiệm mô hình.

### 6.2.1 Thử nghiệm hệ thống với bậc tự do Pitch

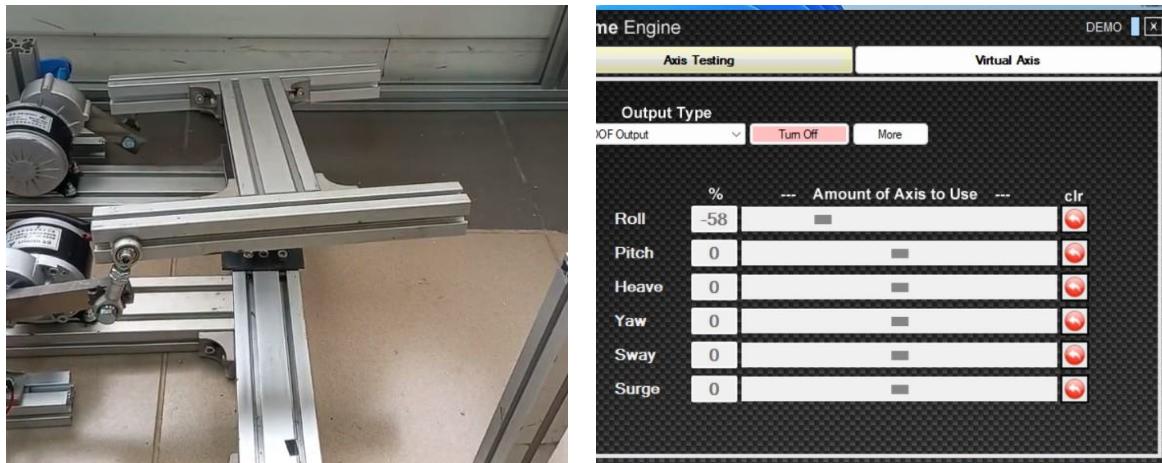


Hình 6.3 Thử nghiệm với bậc tự do Pitch dương

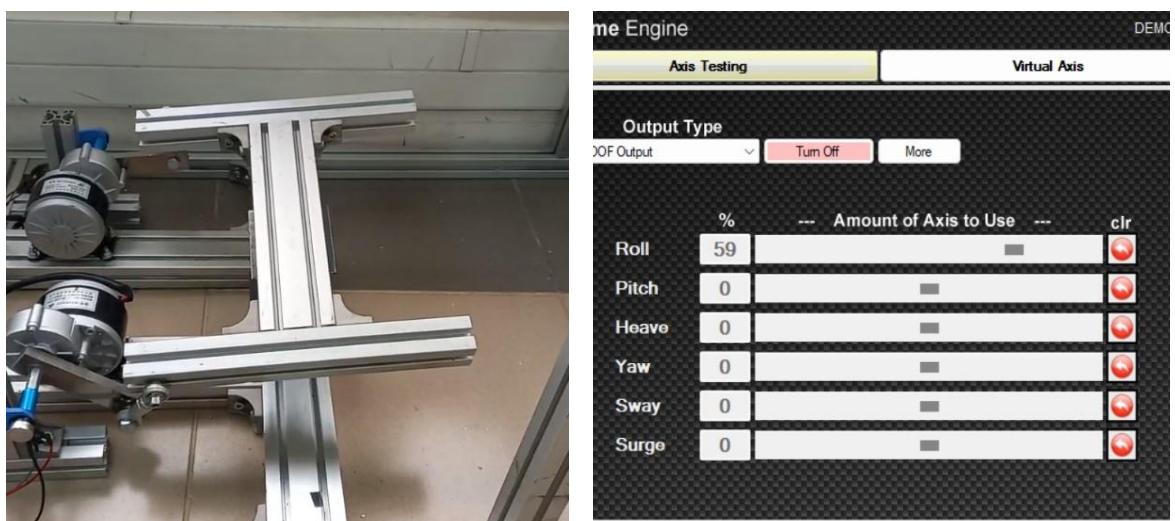


Hình 6.4 Thử nghiệm với bậc tự do Pitch âm

### 6.2.2 Thử nghiệm với bậc tự do Roll



Hình 6.5 Thử nghiệm với bậc tự do Roll âm



Hình 6.6 Thử nghiệm với bậc tự do Roll dương

## CHƯƠNG 7. KẾT LUẬN

Sau hơn 3 tháng thực hiện đồ án, nhóm đồ án đã hoàn thành đồ án thiết kế hệ thống mô phỏng lái 2 bậc tự do. Chúng em xin được trình bày các kết quả đạt được cũng như những hạn chế chưa giải quyết được như sau:

### 7.1 Kết luận

Đồ án đã hoàn thành được các nhiệm vụ sau:

- Thiết kế mô hình 3D và gia công, chế tạo thành công hệ mô phỏng lái.
- Mô phỏng sơ bộ hệ thống trên Matlab - Simulink
- Xây dựng được bộ điều khiển, ứng dụng firmware SMC3 để điều khiển động cơ.
- Kết nối được phần cứng với phần mềm mô phỏng
- Kết xuất dữ liệu từ ứng dụng Live for Speed để điều khiển hệ thống mô phỏng lái.
- Chạy thử nghiệm

Tuy nhiên, đồ án vẫn còn nhiều hạn chế:

- Còn một số khuyết điểm về cơ khí khiến hệ thống chưa được linh hoạt
- Chưa đưa được tính toán động học vào trong mô phỏng qua Matlab-Simulink
- Chưa giải quyết được bài toán động học nghịch, tĩnh học, phân tích độ cứng vững.

### 7.2 Định hướng phát triển

- Tối ưu hóa các vấn đề về cơ khí
- Xây dựng phần mềm riêng để phục vụ những mục đích khác nhau trong thực tế

## **TÀI LIỆU THAM KHẢO**

- [1] Xsimulator, “SimToolsUserManual-v2.4,” Available:  
<https://www.xsimulator.net/>.
- [2] Xsimulator <https://www.xsimulator.net/community/threads/sm3-arduino-3dof-motor-driver-and-windows-utilities.4957/>
- [3] Xsimulator <https://www.xsimulator.net/community/threads/my-finished-2dof-project-videotutorial.16214/>