

**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**



**BÀI TẬP LỚN: NHẬP MÔN ĐIỀU KHIỂN THÔNG MINH**  
**ĐIỀU KHIỂN VỊ TRÍ ĐỘNG CƠ DÙNG BỘ ĐIỀU KHIỂN PI MỜ**

**LỚP L01 - HK 242**

**Giảng viên hướng dẫn: PGS.TS Huỳnh Thái Hoàng**

<b>Sinh viên thực hiện</b>	<b>Mã số sinh viên</b>
Nguyễn Tuấn Anh	2210109
Trương Tuấn An	2210041
Võ Quế Long	2211910

*Thành phố Hồ Chí Minh – 2025*

## MỤC LỤC

<b>GIỚI THIỆU .....</b>	<b>1</b>
<b>I. LÝ DO CHỌN ĐỀ TÀI .....</b>	<b>1</b>
<b>II. MỤC TIÊU &amp; PHẠM VI NGHIÊN CỨU .....</b>	<b>2</b>
2.1. Mục tiêu .....	2
2.2. Phạm vi nghiên cứu .....	2
<b>NỘI DUNG .....</b>	<b>3</b>
<b>I. CƠ SỞ LÝ THUYẾT .....</b>	<b>3</b>
1.1. Đặc tính của động cơ điện một chiều .....	3
1.1. Bộ điều khiển PI mờ .....	6
<b>II. THIẾT KẾ BỘ ĐIỀU KHIỂN PI FUZZY .....</b>	<b>9</b>
2.1. Chuẩn hóa các biến vào ra .....	9
2.2. Định nghĩa các giá trị ngôn ngữ cho các biến vào/ ra .....	10
2.3. Lập quy tắc mờ .....	12
<b>III. MÔ PHỎNG HỆ THỐNG .....</b>	<b>14</b>
3.1. Mô hình .....	14
3.2. Fuzzy logic trong matlab .....	15
3.3. Kết quả mô phỏng .....	17
<b>IV. ỨNG DỤNG THỰC NGHIỆM .....</b>	<b>20</b>
4.1. Chuẩn bị .....	20
4.2. Kết nối phần cứng .....	21
4.3. Cấu hình và lập trình phần mềm .....	22
4.4. Kết quả .....	34
<b>KẾT LUẬN .....</b>	<b>37</b>
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>	<b>38</b>

# GIỚI THIỆU

## I. LÝ DO CHỌN ĐỀ TÀI

Lĩnh vực điều khiển tự động ngày càng phát triển, đặc biệt là điều khiển chính xác, đã trở thành một phần không thể thiếu của nền công nghiệp hiện đại. Phần lớn các loại máy móc, thiết bị dân dụng hay trong công nghiệp sử dụng động cơ điện, từ động cơ điện trong các máy công cụ, máy CNC, các cánh tay robot, ... đến trong những thiết bị gia dụng như máy giặt, điều hòa, máy hút bụi, ngay cả trong máy vi tính. Những thiết bị như vậy yêu cầu độ chính xác cao, tiết kiệm năng lượng, tuổi thọ và chu kì bảo dưỡng dài. Một trong những yêu cầu cần được đáp ứng để đạt những chỉ tiêu trên đây là điều khiển được động cơ điện một cách ổn định, đáp ứng nhanh, vận hành trơn tru khi xác lập và khi thay đổi trạng thái.

Việc ứng dụng những thuật toán kinh điển vào vấn đề điều khiển tốc vị trí cơ đã đạt được nhiều kết quả khả quan. Ví dụ như sử dụng bộ điều khiển PI, PID cho kết quả tốt ở một số đối tượng động cơ. Chính định tham số cho bộ điều khiển PID kinh điển cũng có nhiều phương pháp. Tuy nhiên, với các thuật toán, phương pháp kinh điển, ta phải biết chính xác về đối tượng, hoặc mô hình hóa tương đối chi tiết đối tượng. Một điểm nữa là trong quá trình vận hành, nếu như đối tượng thay đổi thì hệ thống có thể mất ổn định hoặc chất lượng điều khiển không còn đáp ứng được yêu cầu. Do đó, auto-tuning là một trong những hướng đi khả quan của điều khiển tự động. Trong điều khiển hiện đại, lý thuyết mờ cung cấp cho ta một hướng đi mới, xây dựng những hệ điều khiển mờ thuần túy hoặc nhưng hệ mờ lai với mục đích nâng cao chất lượng các bộ điều khiển kinh điển, cũng như điều khiển những đối tượng chưa biết hoặc khó nhận dạng.

Với những lý do trên, nhóm đã chọn đề tài “Điều khiển vị trí động cơ dùng bộ điều khiển PI mờ” ở bài tập lớn Lý thuyết điều khiển nâng cao trong HK242 này.

## **II. MỤC TIÊU & PHẠM VI NGHIÊN CỨU**

### **2.1. Mục tiêu**

Trong khuôn khổ bài tập lớn, nhóm xin trình bày về thuật toán Fuzzy-PI xây dựng các bộ điều khiển này trên nền vi điều khiển STM32F103C8T6 , mô phỏng được bộ điều khiển PI mờ điều khiển vị trí motor trên Matlab Simulink và tiến hành làm thực nghiệm để đánh giá hiệu suất bộ điều khiển.

### **2.2. Phạm vi nghiên cứu**

Nghiên cứu xây dựng, khảo sát động cơ DC, phương pháp điều khiển vị trí động cơ.

Mô phỏng hệ thống điều khiển trên MATLAB – Simulink.

Thiết kế, xây dựng mô hình thực nghiệm.

Đánh giá kết quả nghiên cứu.

## NỘI DUNG

### I. CƠ SỞ LÝ THUYẾT

#### 1.1. Đặc tính của động cơ điện một chiều

##### 1.1.1. Mô hình toán học động cơ DC sử dụng nam châm vĩnh cửu

Mô hình điện của động cơ DC tập trung vào các phần tử mạch, phản ánh mối quan hệ giữa điện áp, dòng điện và điện trở. Trong khi đó, mô hình cơ khí khai thác động học của động cơ, bao gồm moment xoắn, ma sát và quán tính để dự đoán chuyển động quay và hành vi của động cơ.

Động cơ DC có thể được mô hình hóa như một hệ thống động bậc hai, được mô tả bởi phương trình sau:

$$\Gamma = KV = J\ddot{\theta} + v\dot{\theta} + \tau_{nlf} \quad (1)$$

Trong đó:

$\Gamma$ : Moment xoắn của động cơ.

K: Hằng số động cơ.

V: Điện áp đầu vào.

J: Moment quán tính

v: Hệ số ma sát nhớt.

$\tau^{nlf}$ : Mô men ma sát tĩnh Coulomb.

$\dot{\theta}$ : Vận tốc góc của rotor.

Ở: Gia tốc góc.

Khi động cơ DC sử dụng nam châm vĩnh cửu không tải, tải cơ khí bên ngoài gần như bằng không, nên moment quán tính toàn hệ thống chỉ bao gồm moment quán tính của chính rotor và các thành phần quay bên trong động cơ. Khi động cơ gắn thêm tải moment quán tính tổng thể của hệ thống tăng lên đáng kể.

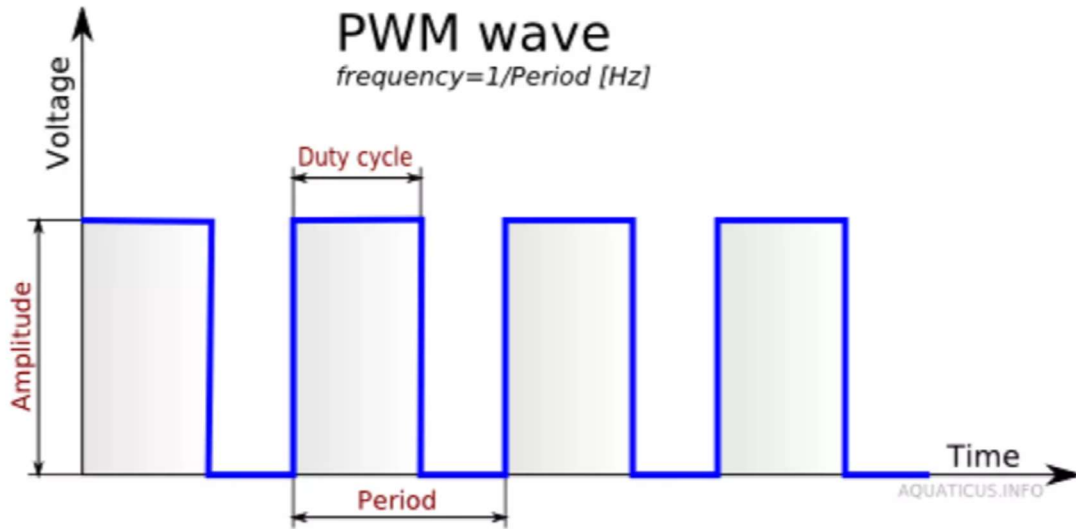
### **1.1.2. Phương pháp nhận dạng trực tuyến mô hình động cơ**

Ta sử dụng encoder để chuyển đổi từ số vòng quay của động cơ thành số xung với một hệ số tỉ lệ nào đó, phụ thuộc vào độ phân giải của encoder. Các xung này được đưa vào vi xử lý và truyền lên máy tính, vẽ đồ thị đáp ứng vòng hở của mô hình động cơ. Từ việc phân tích đồ thị, sử dụng các phương pháp nhận dạng trong lý thuyết điều khiển, ta có mô hình xấp xỉ các thông số của động cơ. Vì lí do phạm vi môn học, nên phương pháp nhận dạng trực tuyến này không được đề cập ở đây. Đối với động cơ sử dụng trong bài tập lớn là động cơ nam châm vĩnh cửu, loại 25GA 370, sử dụng điện áp định mức 12V, công suất 3W, encoder quay 224 xung/vòng (kết quả thực nghiệm).

Thông số động cơ DC được ước lượng thực nghiệm để mô phỏng như sau: hằng số động cơ  $K = 0.2082 \text{ Nm/V}$ , mô men quán tính rotor lúc không tải  $J = 0.001 \text{ kg.m}^2$ , hệ số ma sát nhớt  $\nu = 0.0204 \text{ N.m.s}$ . Điện áp cấp cho phản ứng động cơ tối đa 12V moment ma sát tĩnh Coulomb  $= 0 \text{ Nm}$ .

### **1.1.3. Phương pháp điều khiển vị trí động cơ**

Động cơ DC dùng nam châm vĩnh cửu ta coi là có từ thông không đổi. Phương trình (1) biểu diễn quan hệ giữa vận tốc động cơ với điện áp đặt vào hai đầu cuộn dây phản ứng. Mặt khác, tác động về mặt cơ học của động cơ là tương đối nhanh. Do đó ta sử dụng phương pháp điều khiển điện áp phản ứng để thay đổi tốc độ động cơ DC, cụ thể là sử dụng phương pháp điều chế độ rộng xung PWM kết hợp với thuật toán điều khiển vị trí để đảm bảo động cơ đạt đến vị trí mong muốn chính xác.



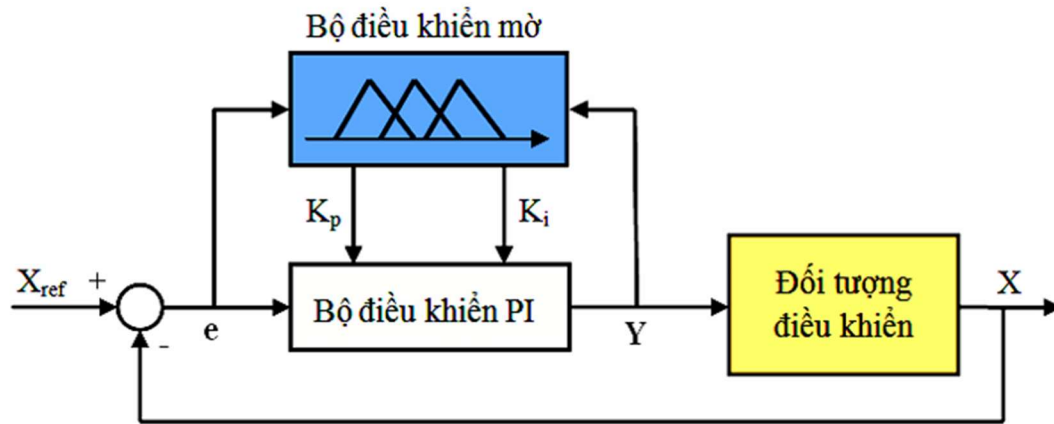
Hình 1.1.1. Nguyên lý điều chế độ rộng xung PWM

Giá trị điện áp trung bình trên tải:

$$U_t = U_{max} \frac{T_{on}}{T_{off}} = Amplitude * \frac{Duty Cycle}{Perio} \quad (2)$$

Ta thấy giá trị điện áp trên tải phụ thuộc vào tỉ số  $\gamma = \frac{Duty Cycle}{Perio}$  do đó ứng với mỗi tần số xung, ta có thể điều chỉnh Duty cycle để điều chỉnh điện áp. Đối với vi điều khiển STM32F103C8T6 sử dụng thạch anh 72MHz, ta có các timer 16 bit hỗ trợ chế độ PWM với độ phân giải cao. Theo lý thuyết, ta có thể tạo xung vuông với tần số từ vài Hz đến hàng trăm kHz. Tuy nhiên, để tạo xung PWM, cần điều khiển thông qua giá trị duty cycle, nên dải tần số PWM sẽ bị giới hạn bởi độ phân giải của bộ đếm và yêu cầu về độ mịn của điều khiển. Khi lựa chọn tần số của xung PWM, cần chọn sao cho đáp ứng cơ học của động cơ đủ mịn để không gây ra hiện tượng giật hoặc rung do sự thay đổi điện áp không liên tục.

## 1.1. Bộ điều khiển PI mờ



Hình 1.2.1. Mô hình tổng quát của hệ thống điều khiển

### 1.2.1. Cấu trúc bộ điều khiển PI

Bộ điều khiển PI được thực hiện bởi Johnson M.A. & M.H. Moradi, 2005:

$$Y(t) = K_p e(t) + K_i \int_0^t e(t) dt$$

$$e(t) = X_{ref}(t) - X(t)$$

trong đó,  $K_p$  và  $K_i$  lần lượt là độ lợi tỉ lệ và độ lợi tích phân của bộ điều khiển. Nhiệm vụ của người thiết kế bộ điều khiển PI, là chọn lựa bộ giá trị  $K_p$ ,  $K_i$  thỏa mãn các yêu cầu về chất lượng điều khiển.

Các thông số độ lợi  $K_p$  và  $K_i$  ảnh hưởng đến thời gian tăng, độ vọt lố và thời gian xác lập của đáp ứng được cho trong bảng 1 (Kanagaraj *et. al.*, 2008). Các thông tin này được xem là cơ sở tri thức để xác định luật chỉnh định mờ bộ điều khiển PI.



Độ lợi	Thời gian tăng	Độ vọt lố	Thời gian xác lập	Sai số xác lập
$K_p$	Giảm	Tăng	Thay đổi ít	Giảm
$K_i$	Giảm	Tăng	Tăng	Triệt tiêu

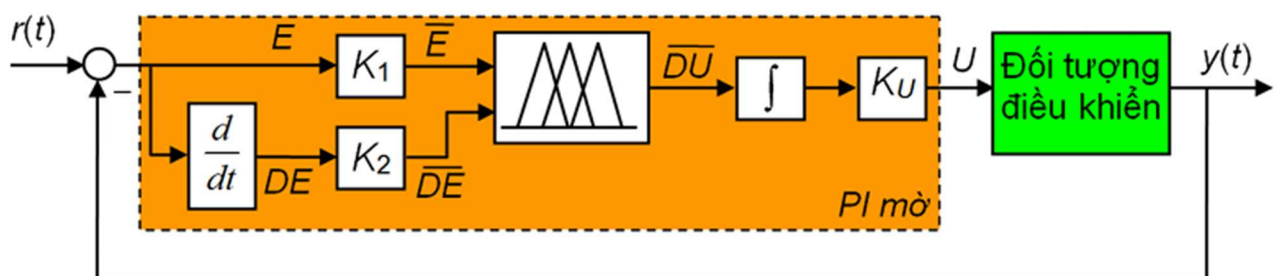
Bảng 1: Ảnh hưởng của việc tăng các thông số độ lợi của bộ điều khiển PI

### 1.2.2. Thiết kế bộ điều khiển PI mờ.

Không giống như các bộ điều khiển thông thường dựa trên các mô hình toán học chính xác, bộ điều khiển logic mờ sử dụng phương pháp dựa trên dữ liệu mô phỏng lý luận của con người, làm cho chúng đặc biệt phù hợp với các hệ thống có những bất định và phi tuyến tính vốn có. Khả năng thích ứng này cho phép bộ điều khiển logic mờ xử lý các biến đổi trong động lực của mô tơ và nhiễu bên ngoài dễ dàng hơn. Thêm vào đó, khả năng tích hợp kiến thức chuyên gia trực tiếp vào chiến lược điều khiển cho phép bộ điều khiển logic mờ đạt được hiệu suất và độ tin cậy cao, ngay cả trong các môi trường điều khiển phức tạp. Kết quả là, bộ điều khiển logic mờ cung cấp một lựa chọn trực quan và linh hoạt cho việc điều khiển động cơ DC, nâng cao cả hiệu quả và độ ổn định trong một loạt các ứng dụng.

Bộ điều khiển PI mờ nếu thiết kế tốt có thể điều khiển điều khiển đối tượng trong miền làm việc rộng với sai số xác lập bằng 0. Tuy nhiên cũng cần để ý rằng bộ điều khiển PI làm chậm đáp ứng của hệ thống và trong nhiều trường hợp làm cho quá trình quá độ có dao động.

So sánh	
Bộ điều khiển FUZZY PI	Bộ điều khiển PI
<ul style="list-style-type: none"> <li>Có thể làm việc với tín hiệu đầu vào kém chính xác hơn.</li> <li>Không cần bộ xử lý nhanh.</li> <li>Việc điều chỉnh bộ điều khiển fuzzy PI dễ dàng, mạnh mẽ hơn so với các bộ điều khiển phi tuyến khác.</li> <li>Bộ điều khiển fuzzy có độ ổn định tốt hơn, quá điều chỉnh nhỏ, và đáp ứng nhanh.</li> </ul>	<ul style="list-style-type: none"> <li>Thuật toán bộ điều khiển PI truyền thống đơn giản, ổn định, dễ điều chỉnh và độ tin cậy cao.</li> <li>Không cần bộ xử lý.</li> <li>Việc điều chỉnh các thông số điều khiển PI rất khó khăn, độ mạnh kém, do đó khó đạt được trạng thái tối ưu trong điều kiện thực tế tại hiện trường sản xuất.</li> <li>Khi tải thay đổi, hệ thống trở nên không ổn định, gây ra quá điều chỉnh lớn hơn.</li> </ul>



Hình 1.2.2. Sơ đồ khối hệ thống điều khiển

Trình tự thiết kế bộ điều khiển PI mờ như sau:

Bước 1: Vẽ sơ đồ khối hệ thống điều khiển (HTĐK). Xác định tầm giá trị của:

Biến vào: sai số ( $E$ ) và vi phân sai số ( $DE$ ).

Biến ra: vi phân của tín hiệu điều khiển ( $DU$ ).

Bước 2: Xác định các hệ số chuẩn hóa biến vào và biến ra về miền giá trị  $[-1,1]$   $[-1, 1]$ .

Bước 3: Định nghĩa các giá trị ngôn ngữ cho các biến vào và biến ra, định lượng các giá trị ngôn ngữ bằng các tập mờ.

Bước 4: Xây dựng hệ qui tắc mờ bằng các hình vẽ minh họa để có ý tưởng đưa ra một số qui tắc điển hình, sau đó áp dụng tính liên tục của hệ mờ và tính đối xứng để đưa ra các qui tắc còn lại.

Bước 5: Chọn phương pháp suy luận mờ MAX–MIN hoặc MAX–PROD.

Bước 6: Chọn phương pháp giải mờ: Phương pháp trọng tâm hoặc trung bình có trọng số.

Bước 7: Mô phỏng hoặc thực nghiệm đánh giá chất lượng điều khiển, tinh chỉnh các thông số của bộ điều khiển mờ để đạt chất lượng mong muốn.

## **II. THIẾT KẾ BỘ ĐIỀU KHIỂN PI FUZZY**

### **2.1. Chuẩn hóa các biến vào ra**

Do đối tượng không có khâu tích phân lý tưởng nên dùng bộ điều khiển PI mờ.

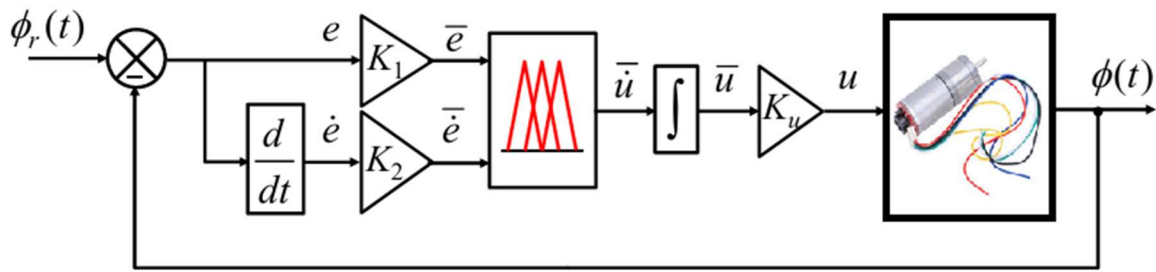
Tín hiệu đặt:  $0 \leq \varphi_r \leq 2\pi$  (rad)

Các biến vào/ra của bộ điều khiển PI mờ cơ bản:

2 biến vào: sai số và tốc độ biến thiên sai số.

Biến ra: biến thiên mômen điều khiển.

Sơ đồ khối hệ thống điều khiển:



Hình 2.1.1. Sơ đồ khối hệ thống điều khiển

Chuẩn hóa biến vào/ ra của bộ điều khiển:

$$\text{Sai số: } e(t) = \phi_r(t) - \phi(t) \quad -2\pi \leq e \leq 2\pi \text{ (rad)} \Rightarrow K_1 = \frac{1}{2\pi}$$

$$\text{Biến thiên sai số: } \frac{1}{6} \leq \dot{e} \leq \frac{1}{6} \text{ (rad/s)} \Rightarrow K_2 = \frac{6}{\pi}$$

$$\text{Mômen điều khiển: } -12 \leq u \leq 12 \text{ (Nm)} \Rightarrow K_u = 12$$

Khi mô phỏng và tiến hành thực nghiệm các hệ số cần tinh chỉnh.

## 2.2. Định nghĩa các giá trị ngôn ngữ cho các biến vào/ ra

Một bộ điều khiển mờ kiểu Mamdani sẽ được thiết kế để giải quyết thách thức hiện tại. Khi làm như vậy, số lượng hàm thuộc cho mỗi đầu vào và đầu ra, hình dạng và phạm vi của chúng, cơ sở luật, phương pháp khử mờ và các hệ số tiền và hậu tỷ lệ đều là những cân nhắc quan trọng. Vì mục đích ứng dụng nhỏ này, một số lựa chọn điển hình sẽ được thực hiện. Kết quả là, 3 hàm liên thuộc cho đầu vào  $e$ , 5 hàm liên thuộc cho đầu vào tích phân  $\dot{e}$  và 5 hàm liên thuộc cho đầu ra tín hiệu điều khiển được chọn. Hình dạng và phạm vi của các hàm liên thuộc đó như sau.

Xây dựng quy luật

Bảng quy luật mờ được xác định như sau:

Các mức sai số  $e$  bao gồm:

- 1- Âm                      Neg
- 2- Zero                      Zero
- 3- Dương                      Pos

Các mức biến thiên sai số  $\epsilon$  bao gồm:

- 4- Âm lớn                      NegBig
- 5- Âm nhỏ                      NegSml
- 6- Zero                      Zero
- 7- Dương nhỏ                      PosSml
- 8- Dương lớn                      PosBig

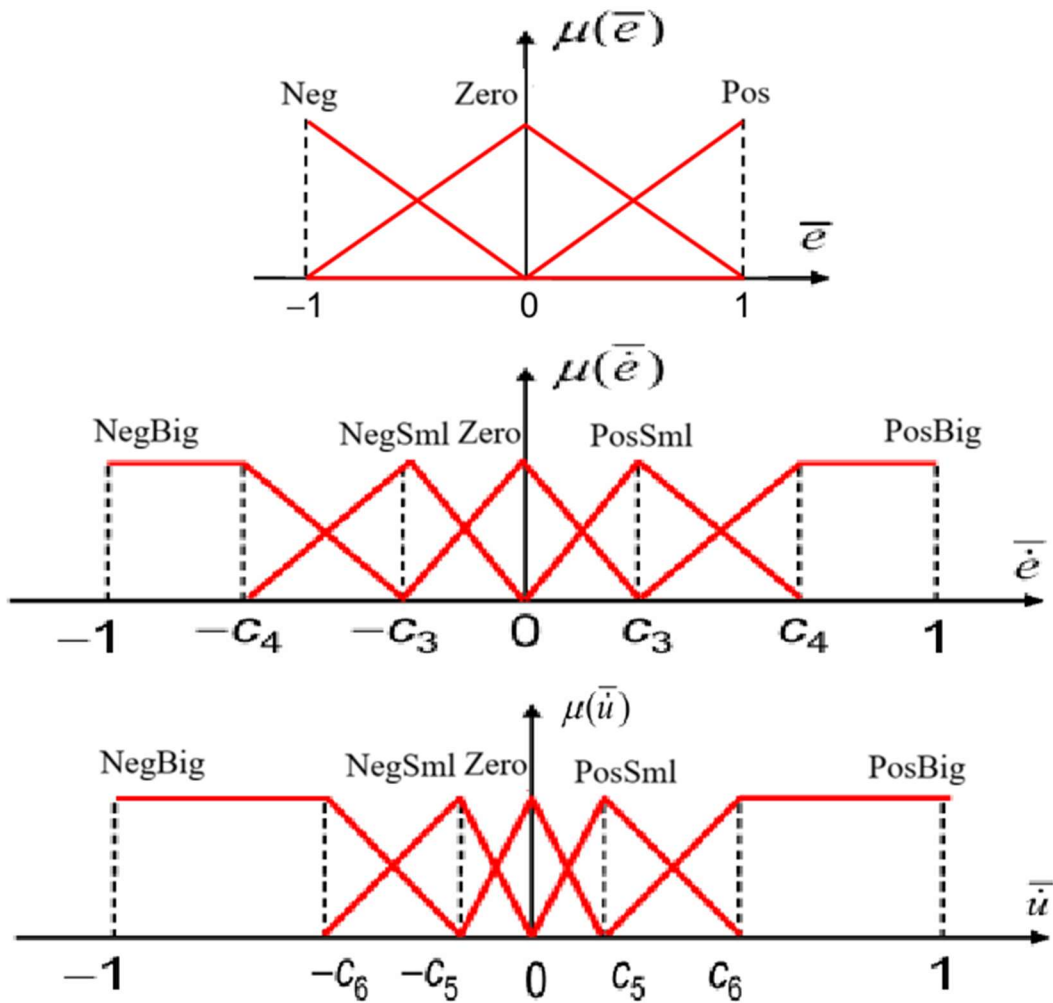
Do tốc độ động cơ tỉ lệ thuận với điện áp nên ta cũng có 6 mức độ điện áp  $u$ :

- 9- Rất thấp                      NegBig
- 10-Thấp                      NegSml
- 11-Zero                      Zero
- 12-Cao                      PosSml
- 13-Rất cao                      PosBig

Từ đây ta tiến hành đặt quy luật:

<b><math>e</math></b>	<b><math>\epsilon</math></b>	<b><math>u</math></b>
Neg	NegBig	NegBig
Zero	NegSml	NegSml
Pos	Zero	Zero
	PosSml	PosSml
	PosBig	PosBig

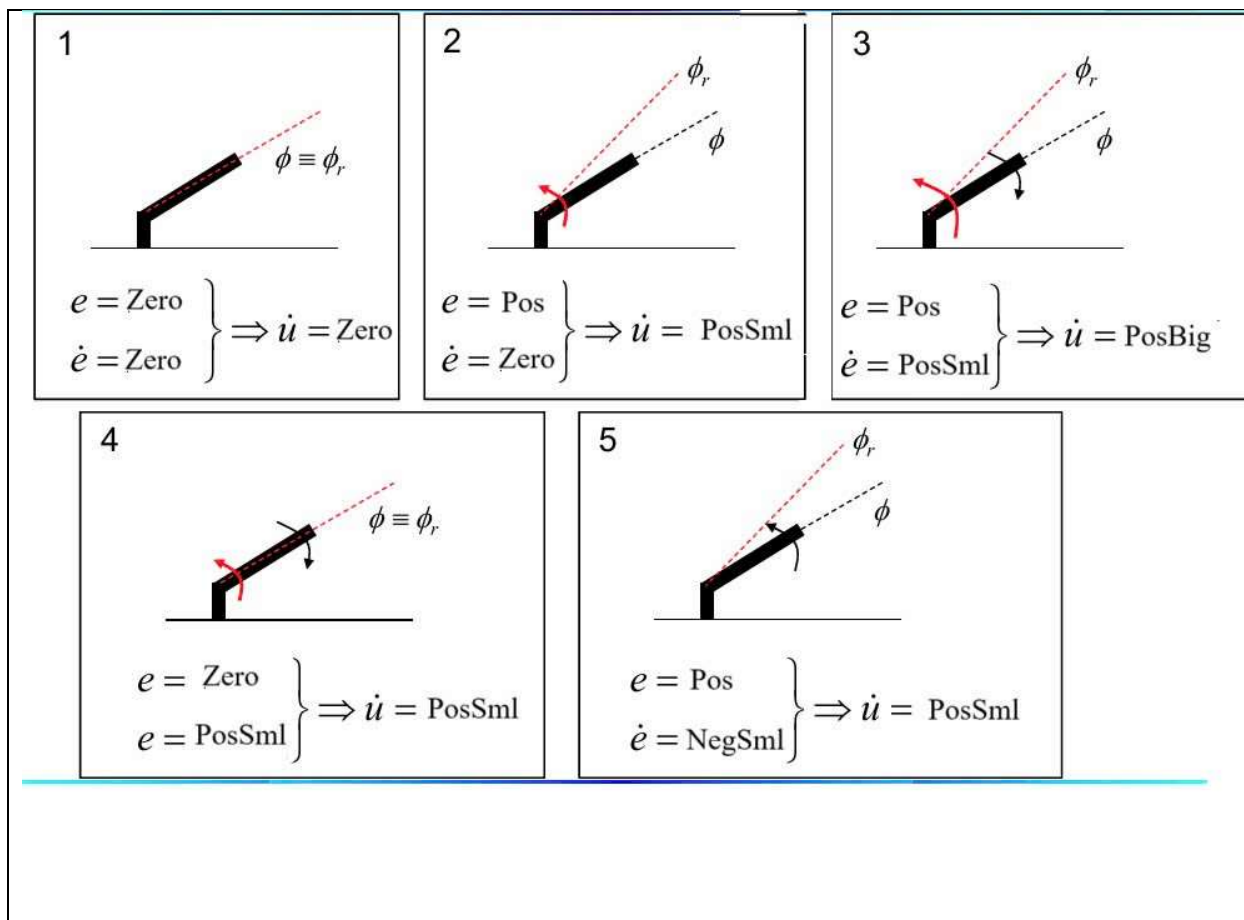
Bảng 2. Thông số bảng quy luật



Hình 2.3.1. Các giá trị ngôn ngữ cho biến vào ra

### 2.3. Lập quy tắc mờ

Qui tắc điều khiển mờ được đưa ra dựa vào kinh nghiệm.



Hình 2.4.1. Vẽ hình minh họa ý tưởng đưa ra 5 luật điều khiển mờ khác nhau.

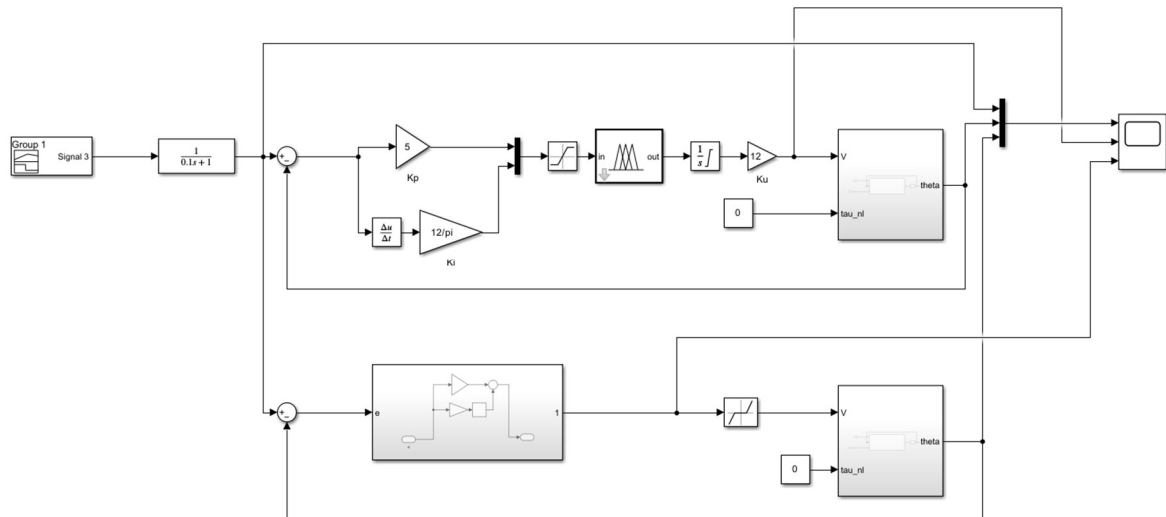
$\dot{u}$		$\dot{e}$		
		Neg	Zero	Pos
E	NegBig	NegBig	NegSml	NegSml
	NegSml	NegBig	NegSml	NegSml
	Zero	NegSml	Zero	PosSml
	PosSml	PosSml	PosSml	PosBig
	PosBig	PosSml	PosSml	PosBig

Bảng 3. Hệ qui tắc mờ được đưa ra dựa vào kinh nghiệm

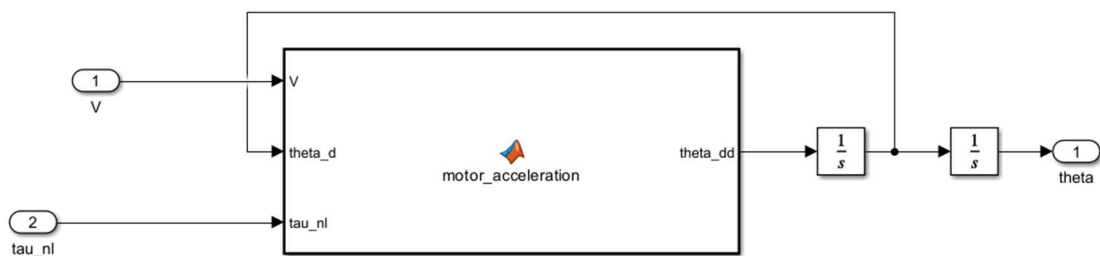
### III. MÔ PHỎNG HỆ THỐNG

#### 3.1. Mô hình

Trong phần này, nhóm sẽ tiến hành mô phỏng để đánh giá hiệu suất của bộ điều khiển PI mờ. Các mô phỏng sẽ được thực hiện bằng phần mềm MATLAB Simulink. Sơ đồ khối của hệ thống điều khiển được minh họa trong Hình 3.1.



Hình 3.1.1. Sơ đồ khối hệ thống điều khiển

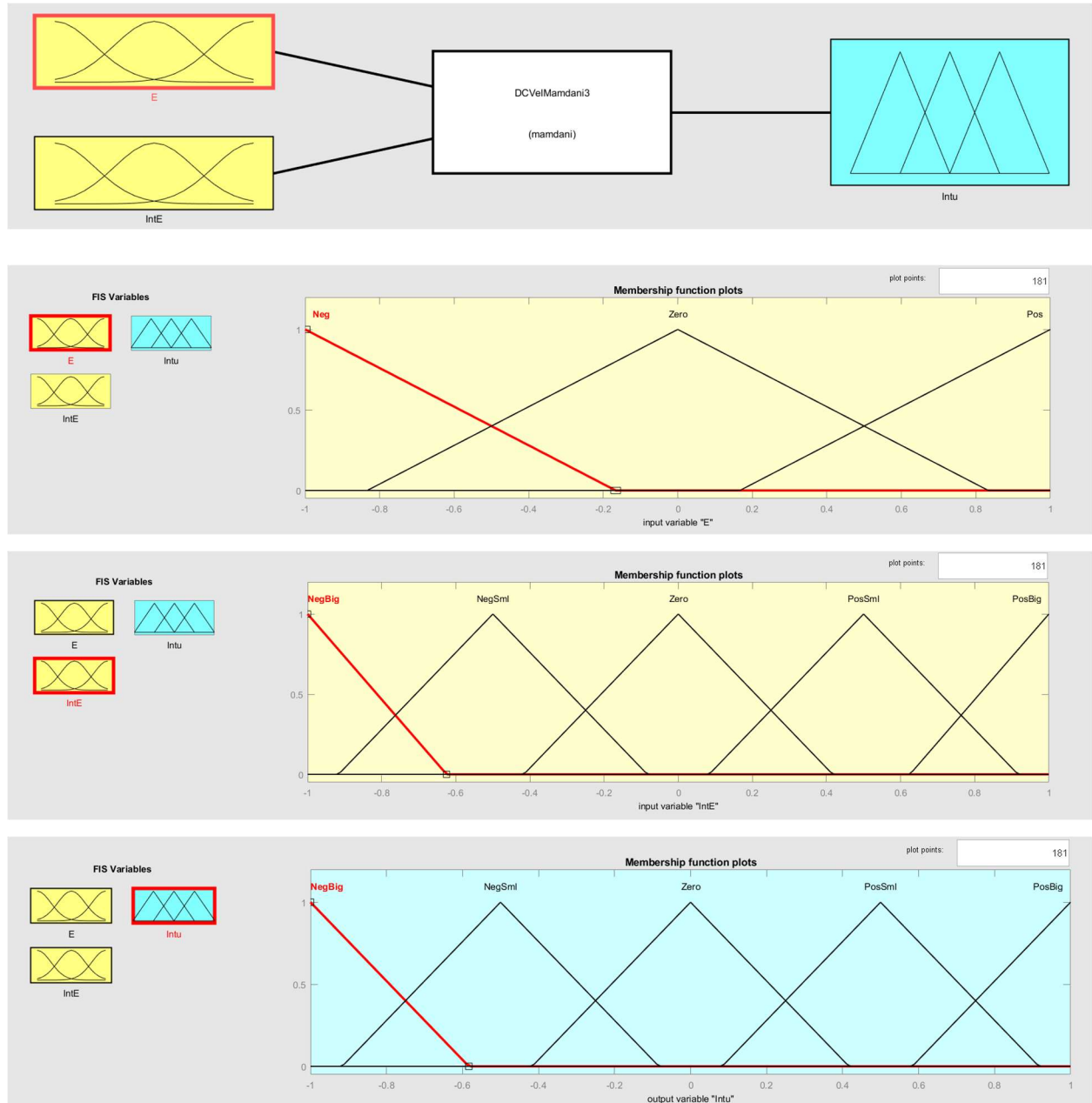


Hình 3.1.2. Mô hình đối tượng Motor DC

Mô phỏng hệ thống điều khiển với  $K_1 = 5$ ,  $K_2 = \frac{12}{\pi}$ ,  $K_u = 12$



### 3.2. Fuzzy logic trong matlab



Hình 3.2.1. Các giá trị hàm liên thuộc

1. If (E is Neg) and (IntE is NegBig) then (u is NegBig) (1)  
2. If (E is Zero) and (IntE is NegBig) then (u is NegSml) (1)  
3. If (E is Pos) and (IntE is NegBig) then (u is NegSml) (1)  
4. If (E is Neg) and (IntE is NegSml) then (u is NegBig) (1)  
5. If (E is Zero) and (IntE is NegSml) then (u is NegSml) (1)  
6. If (E is Pos) and (IntE is NegSml) then (u is PosSml) (1)  
7. If (E is Neg) and (IntE is Zero) then (u is NegSml) (1)  
8. If (E is Zero) and (IntE is Zero) then (u is Zero) (1)  
9. If (E is Pos) and (IntE is Zero) then (u is PosSml) (1)  
10. If (E is Neg) and (IntE is PosSml) then (u is NegSml) (1)  
11. If (E is Zero) and (IntE is PosSml) then (u is PosSml) (1)  
12. If (E is Pos) and (IntE is PosSml) then (u is PosBig) (1)  
13. If (E is Neg) and (IntE is PosBig) then (u is PosSml) (1)  
14. If (E is Zero) and (IntE is PosBig) then (u is PosSml) (1)  
15. If (E is Pos) and (IntE is PosBig) then (u is PosBig) (1)

If

E is

Neg  
Zero  
Pos  
none

☐ not

and

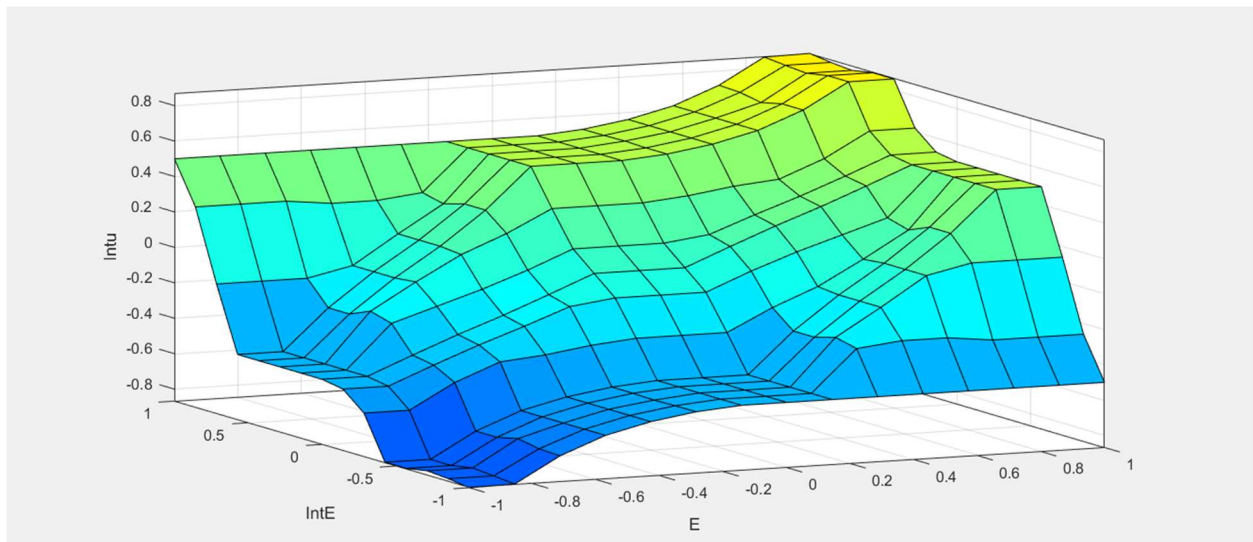
IntE is

NegBig  
NegSml  
Zero  
PosSml  
PosBig  
none

☐ not

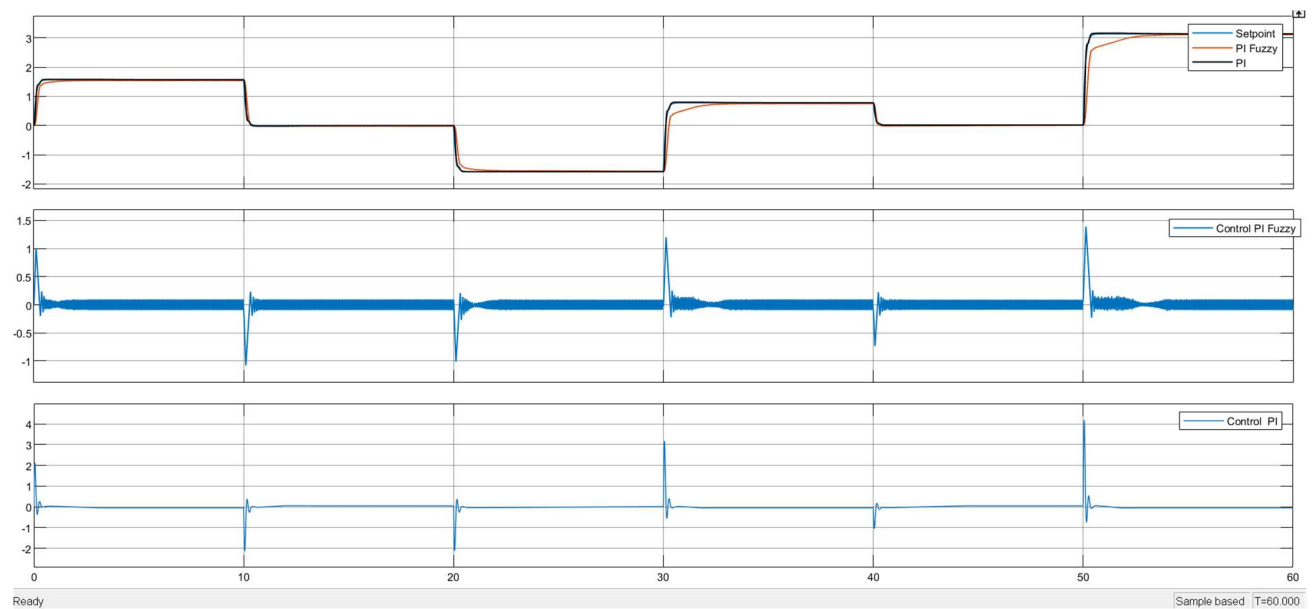
Hình 3.2.2. Hệ qui tắc mờ điều khiển được đưa ra dựa vào kinh nghiệm

Cơ sở luật cùng với phương pháp khử mờ trọng tâm dẫn đến mặt điều khiển như sau:

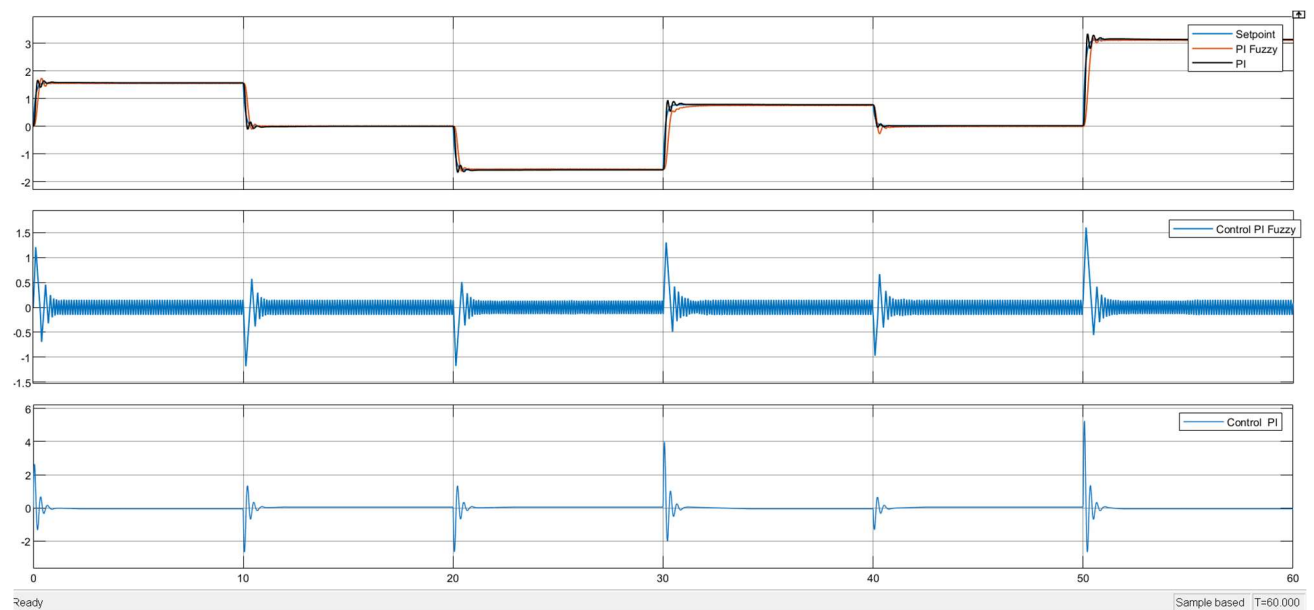


Hình 3.2.3. Mặt điều khiển

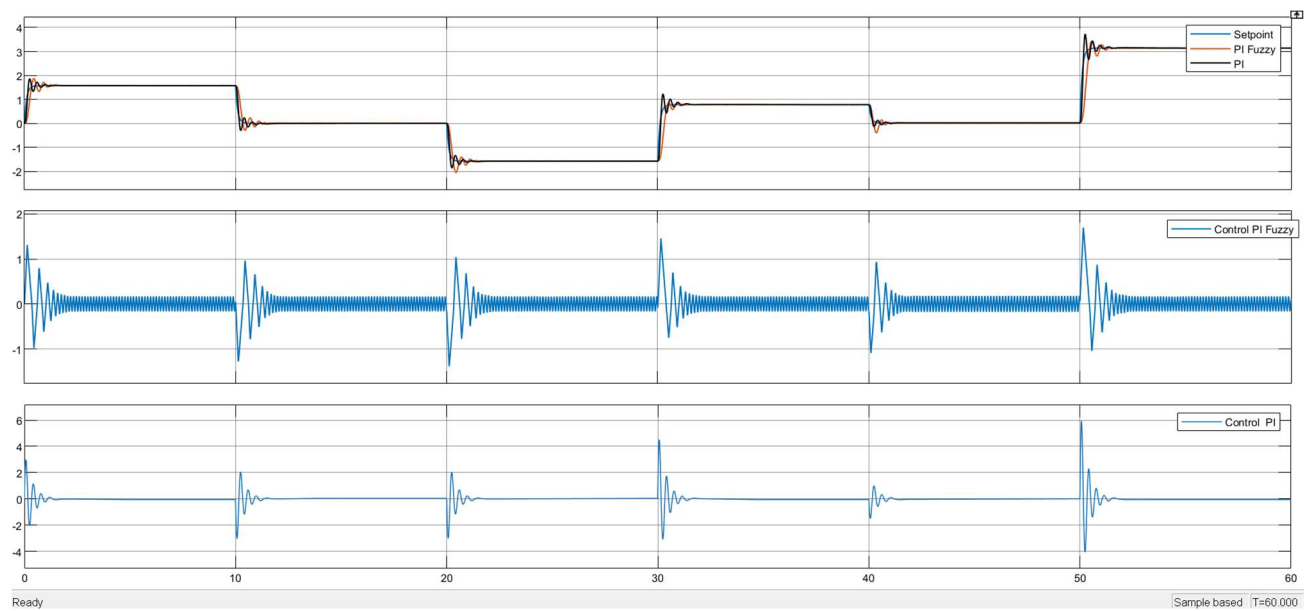
### 3.3. Kết quả mô phỏng



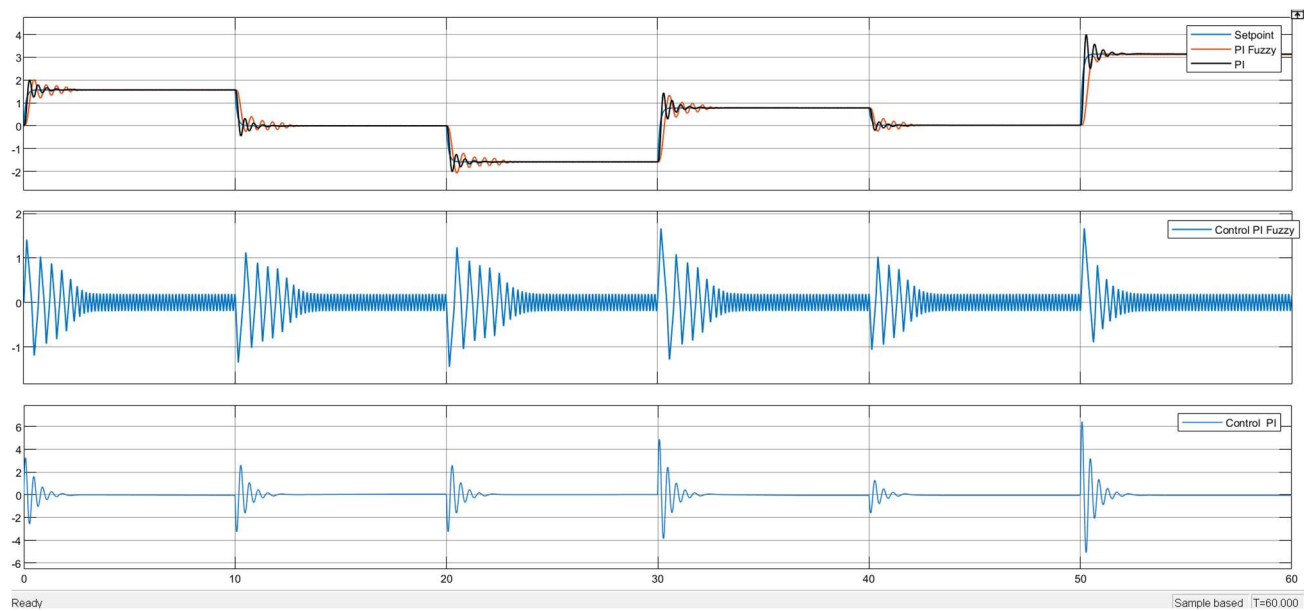
Hình 3.3.1. Kết quả mô phỏng  $J=0.001 \text{ kg.m}^2$



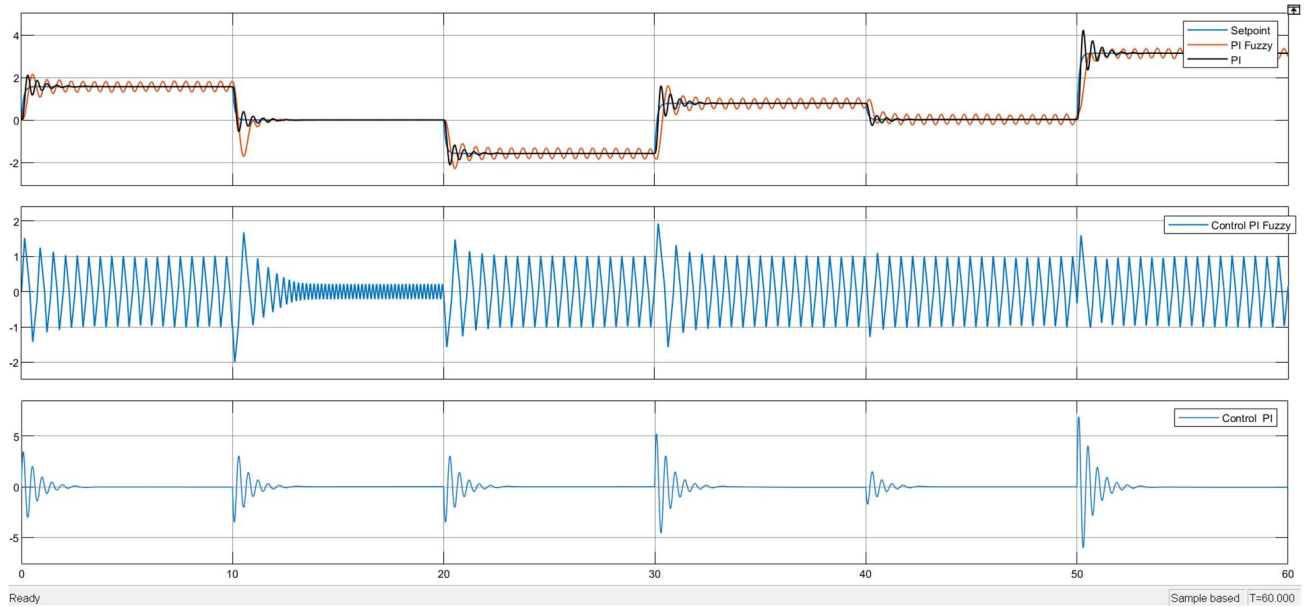
Hình 3.3.2. Kết quả mô phỏng  $J=0.002 \text{ kg.m}^2$



Hình 3.3.3. Kết quả mô phỏng  $J=0.003 \text{ kg.m}^2$



Hình 3.3.4. Kết quả mô phỏng  $J=0.004 \text{ kg.m}^2$



Hình 3.3.5. Kết quả mô phỏng  $J=0.005 \text{ kg.m}^2$

Kết quả trong Hình 3.3.1 khi không tải moment quán tính rotor  $J = 0.001 \text{ kg.m}^2$  cho thấy đầu ra của bộ điều khiển PI mờ, chỉ ra rằng vị trí motor đã bám tốt theo giá trị tham chiếu, với vị trí ban đầu tại với thời gian lấy mẫu là 0.01 giây. Các đường màu xanh dương biểu thị giá trị tham chiếu, trong khi các đường màu đỏ thể hiện giá trị thực tế của motor. Có thể quan sát thấy hệ thống mất khoảng chưa đến 2 giây để đáp ứng đầu ra đạt trạng thái ổn định, khả năng bám tốt theo giá trị đặt.

Khi hệ số moment quán tính rotor  $J$  tăng dần từ 0.002 đến 0.005  $\text{kg.m}^2$ , có thể quan sát thấy hệ thống có xu hướng đáp ứng chậm hơn. Cụ thể, thời gian để động cơ đạt đến trạng thái ổn định tăng lên đáng kể so với trường hợp ban đầu  $J = 0.001 \text{ kg.m}^2$ . Điều này là do moment quán tính lớn hơn khiến động cơ khó tăng tốc nhanh khi có sự thay đổi từ tín hiệu điều khiển. Đồng thời, độ vọt lố của hệ thống có xu hướng giảm dần khi  $J$  tăng, phản ánh quán tính lớn giúp hạn chế sự tăng tốc đột ngột. Tuy vậy, trong tất cả các trường hợp, hệ thống vẫn duy trì được khả năng bám theo giá trị đặt với sai số xác lập nhỏ, chứng minh hiệu quả và tính thích ứng tốt của bộ điều khiển PI mờ. Kết quả mô phỏng cũng cho thấy bộ điều khiển hoạt động ổn định ngay cả khi tham số động học thay

đổi, khẳng định ưu điểm nổi bật của điều khiển mờ trong việc xử lý những bất định và biến động của hệ thống.

Sử dụng điều khiển logic mờ cho bài toán điều khiển vị trí động cơ mang lại nhiều ưu điểm so với các phương pháp truyền thống dựa trên mô hình. Điều khiển logic mờ có tính bền vững cao trước các bất định và sự biến thiên của tham số hệ thống, giúp nó thích nghi tốt hơn và chịu đựng nhiều hiệu quả. Nó đơn giản hóa quá trình thiết kế nhờ dựa vào các luật trực quan thay vì các mô hình toán học phức tạp, và xử lý phi tuyến một cách tự nhiên.

Tuy nhiên, trong trường hợp này, kết quả điều khiển PI lại cho hiệu quả tốt hơn điều khiển fuzzy PI. Trước hết, thiết kế bộ điều khiển mờ chưa tối ưu có thể là nguyên nhân chính. Bộ điều khiển fuzzy phụ thuộc nhiều vào các yếu tố như tập luật, hàm thành viên và các hệ số tỷ lệ. Nếu các thành phần này không được điều chỉnh hợp lý và tối ưu, bộ điều khiển fuzzy sẽ không thể phát huy hết ưu điểm vốn có, thậm chí có thể cho kết quả kém hơn so với bộ PI truyền thống.

Ngoài ra, đặc điểm của hệ thống mô phỏng cũng đóng vai trò quan trọng. Bộ điều khiển fuzzy PI thường thể hiện ưu thế vượt trội khi hệ thống có tính phi tuyến cao hoặc xuất hiện nhiều nhiễu, lệch tham số. Tuy nhiên, nếu hệ thống mô phỏng chỉ là động cơ DC với mô hình tuyến tính đơn giản, bộ điều khiển PI truyền thống hoàn toàn có thể đạt hiệu quả tốt hơn hoặc tương đương fuzzy PI. Đây là một yếu tố then chốt cần xem xét khi không có mô hình hệ thống và yêu cầu đáp ứng phải thích nghi với sự thay đổi.

## **IV. ỨNG DỤNG THỰC NGHIỆM**

### **4.1. Chuẩn bị**

Để thực hiện thí nghiệm điều khiển, nhóm đã chuẩn bị các thành phần phần cứng cần thiết gồm:

Vi điều khiển STM32F103C8T6, đóng vai trò là bộ xử lý trung tâm, thực hiện thuật toán điều khiển PI mờ và phát tín hiệu PWM điều khiển động cơ.

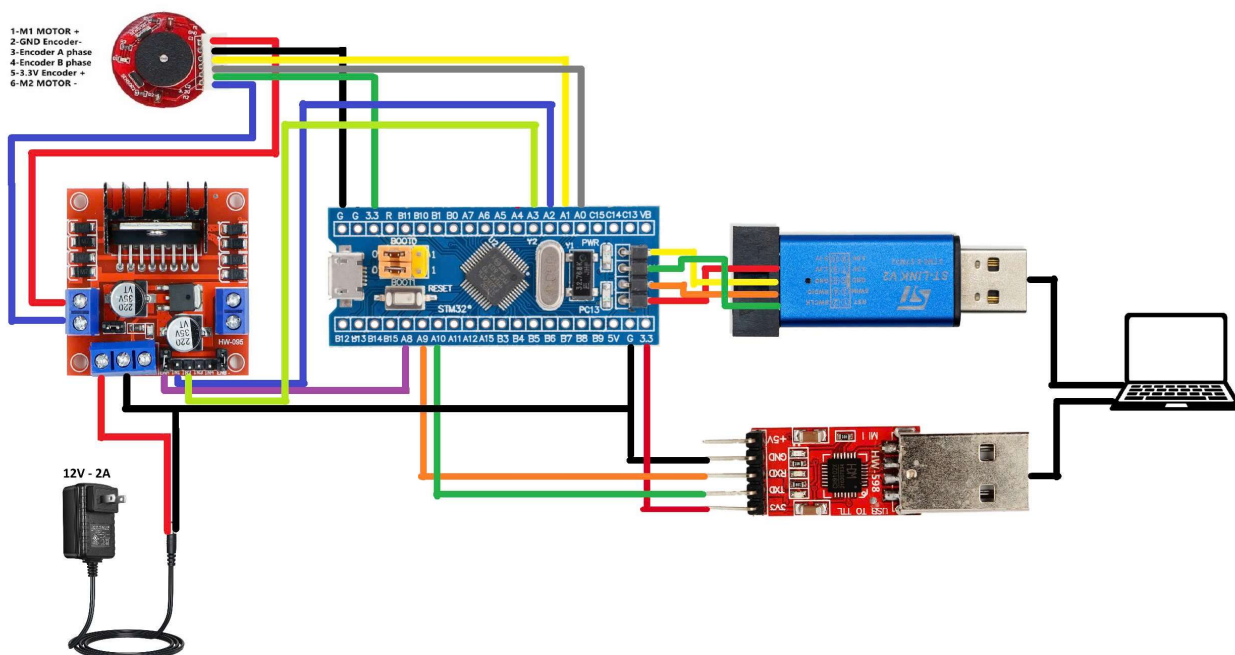
Mạch cầu H L298N, được sử dụng để điều khiển chiều quay và điều chỉnh tốc độ động cơ DC. Mạch này hỗ trợ dòng điện và điện áp đủ lớn để đáp ứng yêu cầu của động cơ, đồng thời cho phép giao tiếp dễ dàng với vi điều khiển thông qua các chân điều khiển.

Nguồn cấp 12V DC, cung cấp điện áp ổn định cho động cơ DC và mạch cầu H L298N.

Động cơ DC 12V có gắn encoder loại 25GA 370, là đối tượng điều khiển chính trong thí nghiệm. Encoder gắn liền cho phép thu thập dữ liệu phản hồi vị trí chính xác, giúp thực hiện điều khiển vòng kín hiệu quả.

Tất cả thiết bị và linh kiện đều được kiểm tra kỹ lưỡng trước khi kết nối nhằm đảm bảo hệ thống vận hành an toàn, ổn định và đạt kết quả mong muốn.

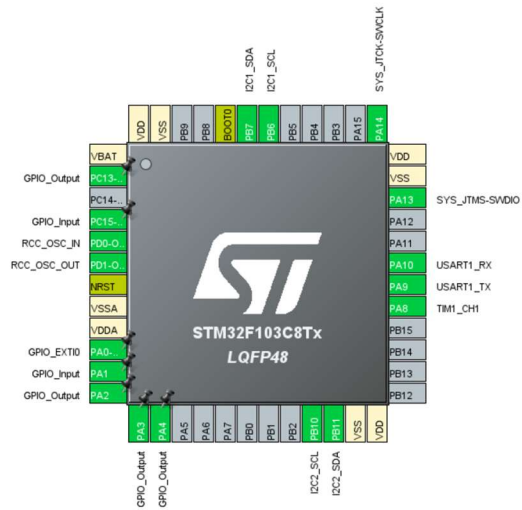
## 4.2. Kết nối phần cứng



Hình 4.2.1. Sơ đồ đấu dây

## 4.3. Cấu hình và lập trình phần mềm

### 4.3.1. Cấu hình



Timer 1 dùng để tạo xung PWM



### TIM1 Mode and Configuration

Mode

Slave Mode Disable ▼  
 Trigger Source Disable ▼  
 Clock Source Internal Clock ▼  
 Channel1 PWM Generation CH1 ▼

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ GPIO Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

🔍

⏪
⏩
i

▼ Counter Settings
 

Prescaler (PSC - 16 bits value) 0  
 Counter Mode Up  
 Counter Period (AutoReload Regi... 65535  
 Internal Clock Division (CKD) No Division  
 Repetition Counter (RCR - 8 bits v... 0  
 auto-reload preload Disable

▼ Trigger Output (TRGO) Parameters
 

Master/Slave Mode (MSM bit) Disable (Trigger input effect not delayed)  
 Trigger Event Selection Reset (UG bit from TIMx\_EGR)

▼ Break And Dead Time management - BR...
 

BRK State Disable  
 BRK Polarity High

Timer 2 dùng để đọc số xung encoder

### TIM2 Mode and Configuration

Mode

Slave Mode Disable ▼

Trigger Source Disable ▼

Clock Source Internal Clock ▼

Configuration

Reset Configuration

✔ Parameter Settings
✔ User Constants
✔ NVIC Settings
✔ DMA Settings

Configure the below parameters :
 

⏪ ⏩
i

▼

Counter Settings

Prescaler (PSC - 16 bits value)	72-1
Counter Mode	Up
Counter Period (AutoReload Regis...	10000-1
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable

▼

Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

Timer 3 dùng để định thời khi vẽ setpoint thay đổi theo thời gian

24

TIM3 Mode and Configuration

Mode

Slave Mode Disable ▼

Trigger Source Disable ▼

☒ Internal Clock

Channel1 Disable ▼

Channel2 Disable ▼

Channel3 Disable ▼

Channel4 Disable ▼

Combined Channels Disable ▼

☐ XOR activation

☐ One Pulse Mode

Configuration

Reset Configuration

✔ Parameter Settings
✔ User Constants
✔ NVIC Settings
✔ DMA Settings

Configure the below parameters :

Search (Ctrl+F) ⏪ ⏩ i

▼ Counter Settings

Prescaler (PSC - 16 bits value)	6000-1
Counter Mode	Up
Counter Period (AutoReload Regi...	60000-1
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable

### 4.3.2.Lập trình phần mềm

Setpoint (số xung) thay đổi theo thời gian cho cả hai bộ điều khiển:

```
void Active_Inactive_Status_Tracking ()
{
    Timer3_Delay_Flag = 0;
    count ++;
```

```

    if (count == 1)
    {
        setpoint = 0;
    }

    if (count == 2)
    {
        setpoint = 50;
    }

    if (count == 3)
    {
        setpoint = 100;
    }

    if (count == 4)
    {
        setpoint = 150;
        count = 0;
    }
}

```

Bộ điều khiển PI:

```

//PID CACULATION & ACTIVE TIMER INTERRUPT//

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if (htim == &htim2)

```

```

{
    error = setpoint - pulse;
    integral = integral + (error * Delta_t);
    derivative = (error - pre_error) / Delta_t;
    duty_cycle_output = (Kp * error) + (Ki * integral);

    if (duty_cycle_output > 100)
    {
        duty_cycle_output = 100;
    }

    if (duty_cycle_output < -100)
    {
        duty_cycle_output = -100;
    }

    pre_error = error;

    Control_Position();
}

if (htim == &htim3)
{
    Timer3_Delay_Flag = 1;
}
}

```

Bộ điều khiển PI Fuzzy:

```
//TRIANGULAR MEMBERSHIP FUNCTION
```

```
float trimf(float x, float a, float b, float c)
```

```
{
    if (x == b)
        return 1.0f;
    if (x < a || x > c)
        return 0.0f;
    if (x < b)
        return (x - a) / (b - a);
    else
        return (c - x) / (c - b);
}
```

```
//MEMBERSHIP STRUCTURE
```

```
typedef struct
```

```
{
    float a, b, c; //Triangular
} MF;
```

```
MF e_mf[NUM_MF] =
```

```
{
    {-1, -1, -0.66}, {-1, -0.66, -0.33}, {-0.66, -0.33, 0}, {-0.33, 0, 0.33},
    {0, 0.33, 0.66}, {0.33, 0.66, 1}, {0.66, 1, 1}
};
```

```
MF de_mf[NUM_MF] =
```

```
{
```

```

    {-1, -1, -0.66}, {-1, -0.66, -0.33}, {-0.66, -0.33, 0}, {-0.33, 0, 0.33},
    {0, 0.33, 0.66}, {0.33, 0.66, 1}, {0.66, 1, 1}
};

float u_mf_centers[NUM_MF] = {-1.0, -0.66, -0.33, 0.0, 0.33, 0.66, 1.0};

//FUZZY RULES
int rule_table[NUM_MF][NUM_MF] =
    {
        {0, 0, 0, 0, 1, 2, 3},
        {0, 0, 0, 1, 2, 3, 4},
        {0, 0, 1, 2, 3, 4, 5},
        {0, 1, 2, 3, 4, 5, 6},
        {1, 2, 3, 4, 5, 6, 6},
        {2, 3, 4, 5, 6, 6, 6},
        {3, 4, 5, 6, 6, 6, 6}
    };

//INFERENCE FUNCTION
float fuzzy_inference(float e, float de)
    {
        float weights[NUM_RULES] = {0};
        float outputs[NUM_RULES] = {0};
        int idx = 0;

        for (int i = 0; i < NUM_MF; i++) {
            float mu_e = trimf(e, e_mf[i].a, e_mf[i].b, e_mf[i].c);

```

```

for (int j = 0; j < NUM_MF; j++) {
    float mu_de = trimf(de, de_mf[j].a, de_mf[j].b, de_mf[j].c);
    float w = mu_e < mu_de ? mu_e : mu_de;
    int out_idx = rule_table[i][j];
    weights[idx] = w;
    outputs[idx] = u_mf_centers[out_idx];
    idx++;
}
}

//DEFUZZY USE WEIGHTED AVERAGE TECHNIQUE
float sum_w = 0, sum_u = 0;
for (int i = 0; i < NUM_RULES; i++) {
    sum_w += weights[i];
    sum_u += weights[i] * outputs[i];
}
if (sum_w == 0) return 0;
return sum_u / sum_w;

```



```

void PID_Calculation ()
{
    PID_Calculation_Flag = 0;

    //=====BEGIN MATLAB=====//

    error = setpoint - pulse;
    error = error / 150;
    if (error > 1.0f) error = 1.0f;
    if (error < -1.0f) error = -1.0f;

    integral = integral + (error * Delta_t);
    integral = integral / 1.5;

    derivative = (error - pre_error) / Delta_t;
    derivative = derivative / 1.5;
    if (derivative > 1.0f) derivative = 1.0f;
    if (derivative < -1.0f) derivative = -1.0f;

    //FUZZY FUNCTION BLOCK
    u_dot = fuzzy_inference(error, derivative);

    //INTEGRAL BLOCK
    u = u + (u_dot * Delta_t);

    //SATURATION BLOCK
    if (u > 1)
    {
        u = 1;
    }
}

```

```

    }

    if (u < -1)
    {
        u = -1;
    }

    if (fabs(setpoint - pulse) <= 2)
    {
u = 0;
    }

    //GAIN BLOCK
    duty_cycle_output = u*100;

    //SATURATION BLOCK
    if (duty_cycle_output > 100)
    {
        duty_cycle_output = 100;
    }

    if (duty_cycle_output < -100)
    {
        duty_cycle_output = -100;
    }

    //DEAD ZONE
    if (fabs(duty_cycle_output) < 5) duty_cycle_output = 0;

```

```
//=====END MATLAB=====//
```

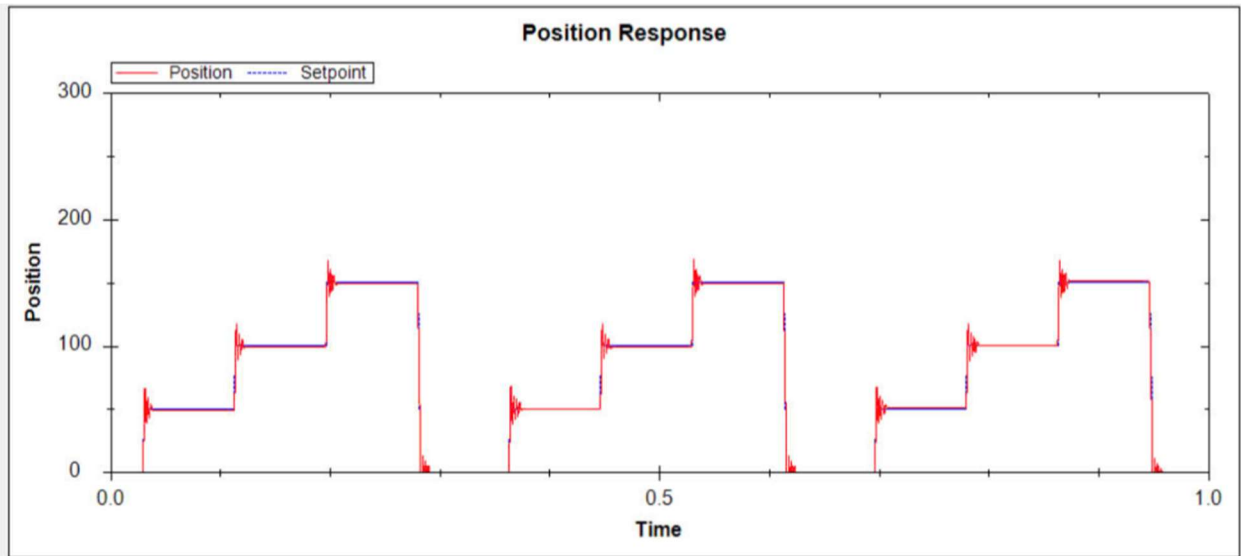
```
pre_error = error;
```

```
Control_Position();
```

## 4.4. Kết quả

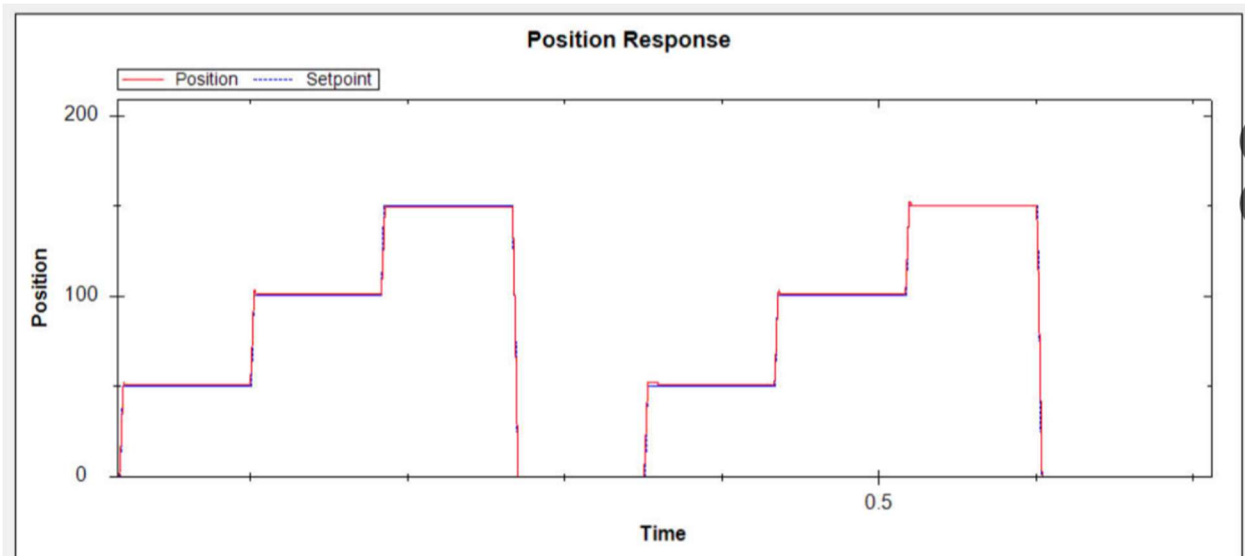
### 4.4.1. Bộ điều khiển PI

Trường hợp không tải



Hình 4.4.1.  $K_1 = 10$  &  $K_2 = 0.5$

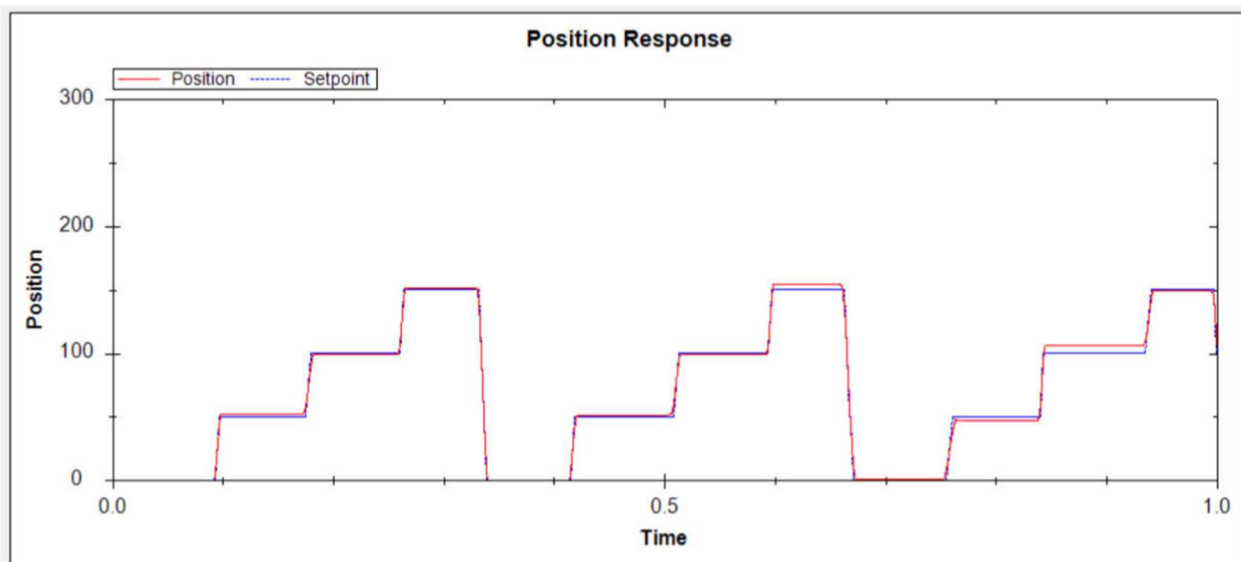
Trường hợp có tải nặng 1.2kg



Hình 4.4.2.  $K_1 = 10$  &  $K_2 = 0.5$

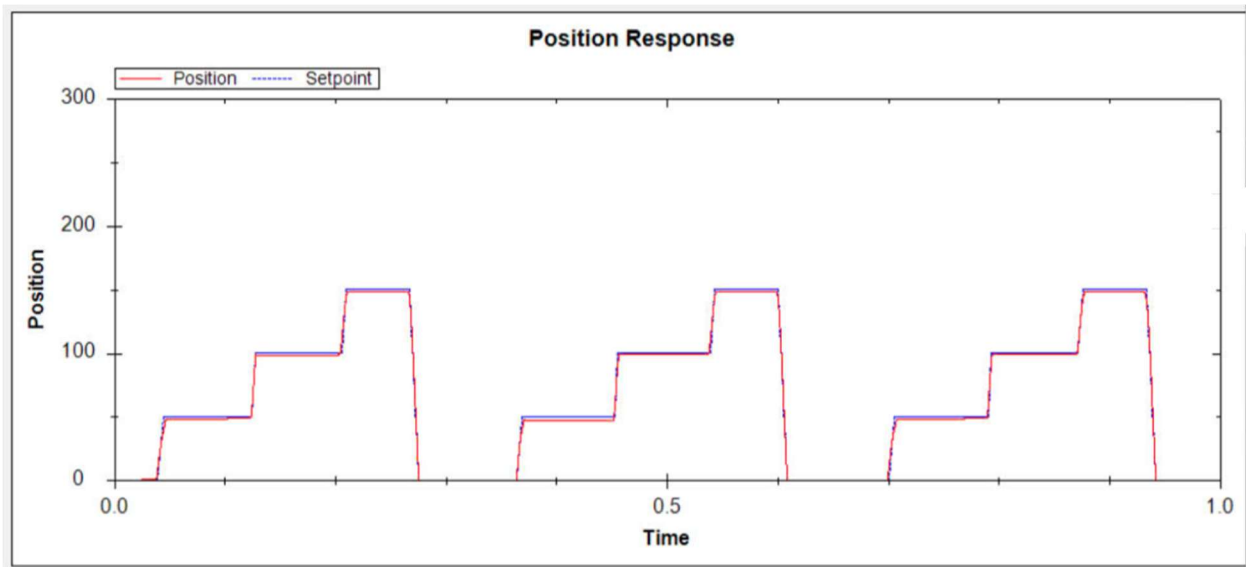
#### 4.4.2. Bộ điều khiển PI Fuzzy

Trường hợp không tải



Hình 4.4.3.  $K_1 = 1/300$  &  $K_2 = 1/1.5$

Trường hợp có tải nặng 1.2 kg



Hình 4.4.4.  $K_1 = 1/200$  &  $K_2 = 1/1.5$

#### 4.4.3. Đánh giá đáp ứng của hai bộ điều khiển

Trong trường hợp không tải của bộ điều khiển PI, hệ thống cho thấy vọt lố đáng kể dẫn đến thời gian xác lập tăng. Ngoài ra chất lượng điều khiển không cao, có khả năng xuất hiện dao động, và độ bám chỉ đạt mức trung bình để đáp ứng ban đầu chậm. Khi có tải, có cùng tham số vọt lố vẫn tồn tại, thời gian xác lập ngắn hơn và độ bám ít bị ảnh hưởng.

Trong trường hợp không tải, bộ điều khiển PI Fuzzy cho thấy vọt lố nhỏ hơn nhiều so với PI, nhờ khả năng điều chỉnh tham số. Thời gian xác lập nhanh hơn, không có vọt lố, chất lượng điều khiển cao với ít dao động, và độ bám khá tốt. Khi có thêm tải, PI Fuzzy vẫn giữ ưu thế so với bộ điều khiển PI với độ vọt lố gần như không có, thời gian xác lập nhanh và độ bám tốt.

So với PI, bộ điều khiển PI mờ vượt trội ở hầu hết các tiêu chí. Đầu tiên là độ vọt lố được giảm đáng kể nhờ khả năng thích nghi của logic mờ. Tiếp đến là thời gian xác thực ngắn hơn, đặc biệt trong điều kiện tải, do PI Fuzzy điều chỉnh tham số dựa trên sai số và nhiễu. Ngoài ra chất lượng điều khiển và độ bám của PI Fuzzy cũng cao hơn, đảm bảo ổn định trong môi trường phức tạp.

## KẾT LUẬN

Trong đề tài này, nhóm đã xây dựng thành công một mô hình bộ điều khiển PI mờ dựa trên động lực học. Thông qua quá trình mô phỏng trên MATLAB/Simulink, bộ điều khiển được thiết kế cho thấy hiệu quả rõ rệt trong việc giúp vị trí motor bám sát các quỹ đạo tham chiếu với sai số nhỏ và thời gian đáp ứng nhanh.

Ưu điểm của phương pháp này là thiết kế và thực hiện bộ quan sát mờ rất đơn giản mà không cần quan tâm đến cấu trúc bên trong của đối tượng điều khiển. Khoảng tinh chỉnh các thông số của bộ điều khiển PI được thiết lập đủ rộng xung quanh giá trị đạt được từ phương pháp thực nghiệm. Kết quả kiểm nghiệm trên motor 25GA 370 cho thấy bộ điều khiển được đề xuất trong nghiên cứu này có hiệu quả, đáp ứng của các thiết bị có thời gian tăng và thời gian xác lập phù hợp, độ vọt lố không đáng kể và sai số xác lập được triệt tiêu. Ngoài ra, bộ điều khiển còn đáp ứng được yêu cầu về thời gian thực, cũng như sự ổn định trước tác động của nhiễu vào hệ thống.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Yasar, C. F. (2025). *DC-Motor-Position-with-Fuzzy-Logic-Control* (Version 1.0.0) [MATLAB code]. MathWorks File Exchange. Truy cập: <https://www.mathworks.com/matlabcentral/fileexchange/167991-dc-motor-position-with-fuzzy-logic-control>
- [2] Er, M.J. and Y.L. Sun, 2001. *Hybird fuzzy proportional-intergral plus conventional derivative control of linear and nonlinear systems*. IEEE Transacions on Industrial Electronics, vol. 48, no. 6, pp. 1109-1117
- [3] PGS. Ts. Huỳnh Thái Hoàng, *Giáo trình điều khiển thông minh*
- [4] PGS. Ts. Huỳnh Thái Hoàng, *Hệ thống điều khiển thông minh*, NXB Đại Học Quốc Gia
- [5] PGS. Ts. Nguyễn Thị Phương Hà, *Lý Thuyết Điều Khiển Hiện Đại*, NXB Đại Học Quốc Gia