

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CƠ KHÍ CHẾ TẠO MÁY**

**BỘ MÔN CƠ ĐIỆN TỬ**



**HCMUTE**

**THỰC TẬP SERVO**

# **BÁO CÁO 1**

**GVHD      ThS: VÕ LÂM CHƯỜNG**

<b>SVTH</b>	<b>Trần Ngọc Hiếu</b>	<b>20146127</b>
	<b>Đỗ Sĩ Hoài</b>	<b>20146491</b>
	<b>Phạm Quang Huy</b>	<b>20146126</b>
	<b>Tiết Nguyễn Hoàng Tấn Đạt</b>	<b>20146488</b>
	<b>Dương Nhật Huy</b>	<b>20146125</b>
	<b>Lê Nhựt Linh</b>	<b>20146159</b>

<b>Lê Văn Mạnh Quỳnh</b>	<b>20146147</b>
--------------------------	-----------------

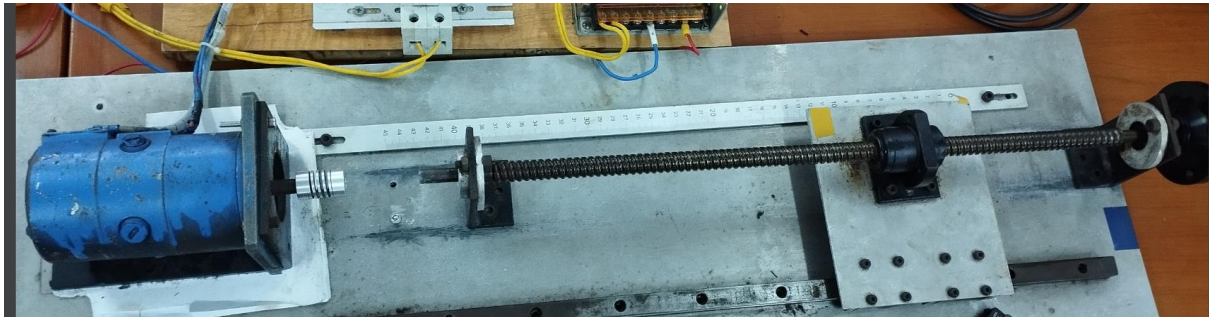
## MODULE 3 AND 4: HỆ THỐNG SERVO MỘT TRỤC SỬ DỤNG ĐỘNG CƠ DC SERVO

### 1. Nội dung thí nghiệm.

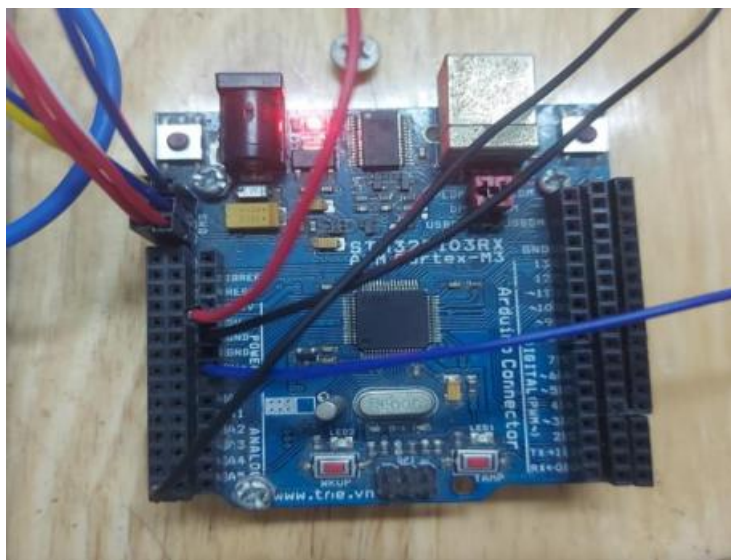
- Mô hình hóa hệ thống servo một trục.
- Sử dụng Simulink / Matlab để minh họa hiệu suất hệ thống.
- Lập trình bằng STM32F103.
- Kiểm soát vận tốc của động cơ servo DC.
- Kiểm soát vị trí của hệ thống servo một trục.

### 2. Giới thiệu hệ thống.

#### 2.1 Các bộ phận chính:



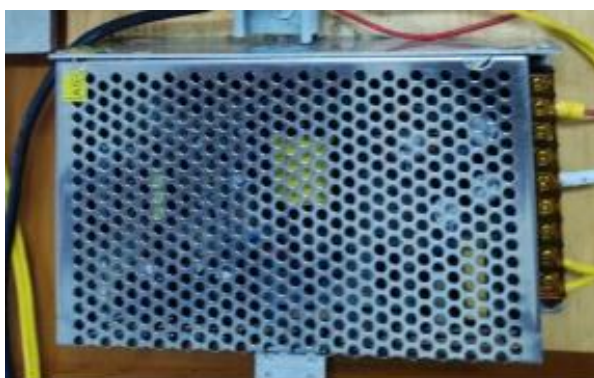
**Hình 1: Động cơ servo DC và encoder**



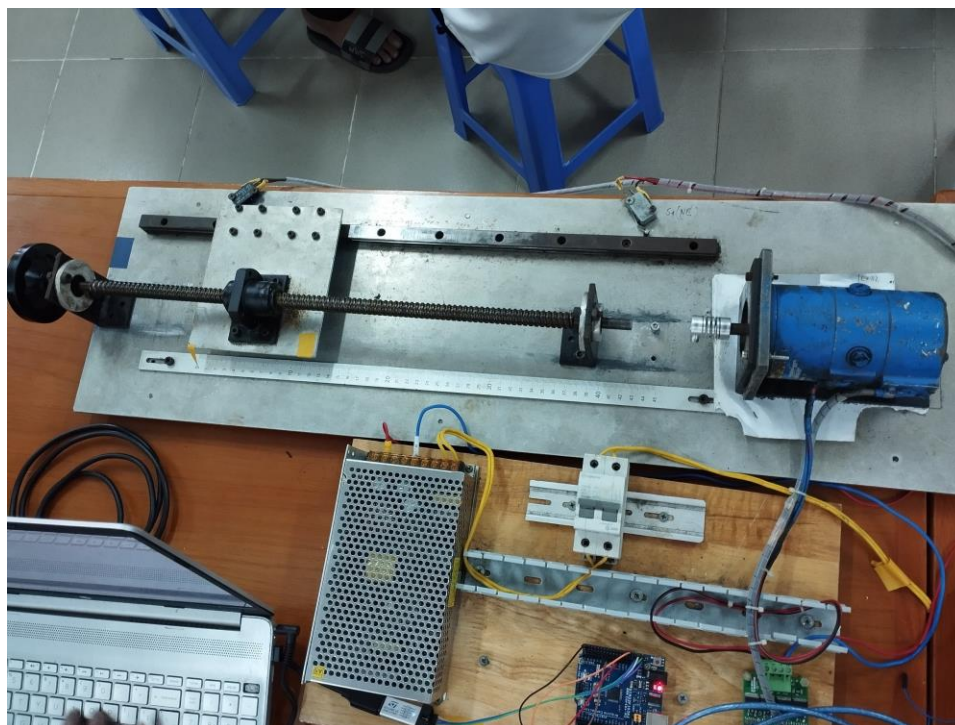
**Hình 2: Vi điều khiển STM32F103**



**Hình 3: Mạch cầu H**



**Hình 4: Bộ nguồn**

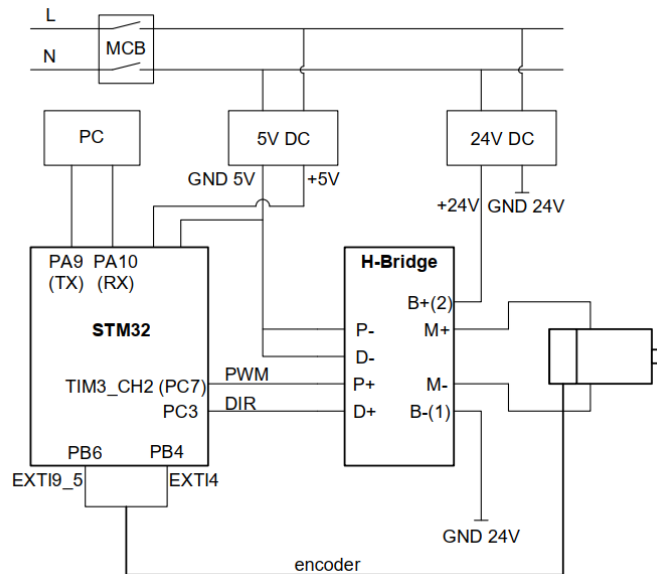


**Hình 5: Tổng quan hệ thống**

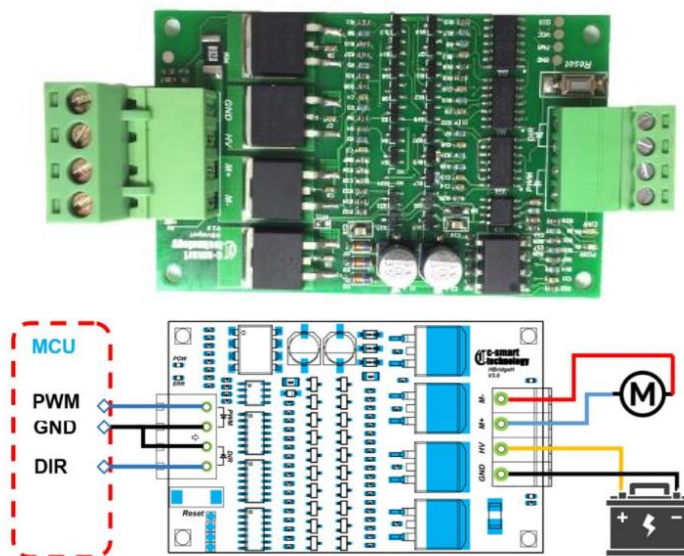


## 2.2. Sơ đồ nối dây

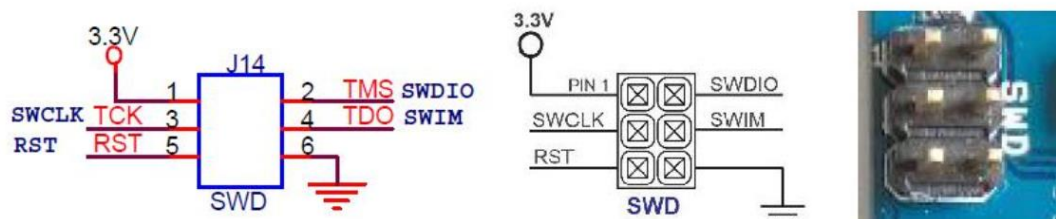
Sơ đồ phân cứng hệ thống servo một trục sử dụng động cơ servo DC:



Mạch cầu H:



Programmer/Debugger:



[illegible]

### 3.1. Xác định hàm truyền hệ thống

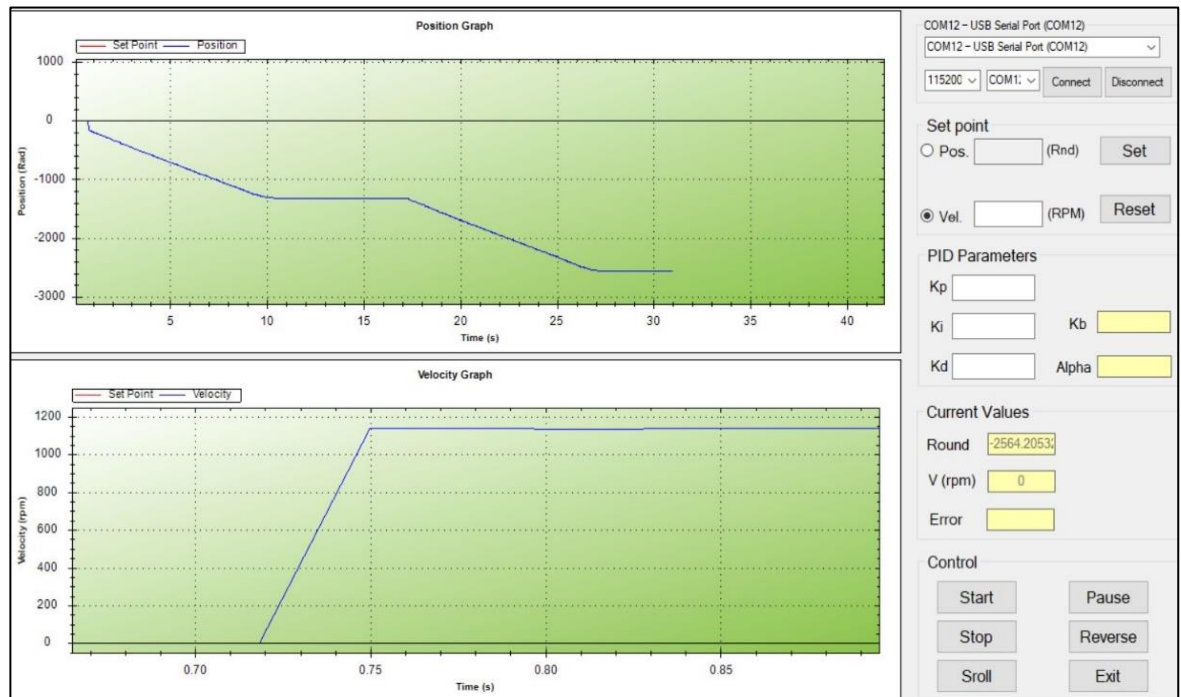
B1: Tăng MV (PWM) từ 0  $\rightarrow$  100%

**B2: Quan sát đồ thị và tìm ra được  $V_{\max} = 1140 \text{ RPM}$**

### B3: Tìm hàm truyền hệ thống

Ta có:

- Tại thời điểm  $v = 0 \text{ RPM} = 0 \text{ rad/s}$ ,  $\tau = 0,7185\text{s}$
- Tại thời điểm  $v = 1140 \text{ RPM} = 119,38 \text{ rad/s}$ ,  $\tau_{100\%} = 0,75\text{s}$
- Chọn gốc  $\tau = 0,7185\text{s}$  làm thời điểm bắt đầu, ta có  $\tau_{0\%} = 0\text{s}$ ,  $\tau_{100\%} = 0,0315\text{s}$
- Căn cứ vào kết quả biểu đồ trên C#, ta thấy được  $\tau_{63\%} = 0,0198\text{s}$



- Ta có hàm truyền tổng quát của hệ bậc 1:

$$G(s) = \frac{Kc}{Ts+1}$$

$$Kc = \frac{\Delta CV}{\Delta MV} = \frac{119,38}{100} = 1,1938$$

$$T = \tau_{63\%} = 0,0198\text{s}$$

Hàm truyền của hệ thống:  $G(s) = \frac{1.1938}{0.01938s+1}$

### 3.2. Module 3: Kiểm soát vận tốc của hệ Servo một trục

#### 3.2.1. Tính toán hệ số cho bộ điều khiển vận tốc (dùng bộ PI)

Sử dụng phương pháp IMC để thiết kế bộ điều khiển PID:

**IMC controller settings for Parallel-Form PID controller (Chien and Fruehauf,1990);  $K=K_p$ ;  $\tau=\tau_p$ ;  $\theta=\theta_p$**

Case	Model	$K_c K$	$\tau_I$	$\tau_D$
A	$\frac{K}{\tau s + 1}$	$\frac{\tau}{\tau_c}$	$\tau$	–
B	$\frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$	$\frac{\tau_1 + \tau_2}{\tau_c}$	$\tau_1 + \tau_2$	$\frac{\tau_1 \tau_2}{\tau_1 + \tau_2}$
C	$\frac{K}{\tau^2 s^2 + 2\zeta \tau s + 1}$	$\frac{2\zeta \tau}{\tau_c}$	$2\zeta \tau$	$\frac{\tau}{2\zeta}$
D	$\frac{K(-\beta s + 1)}{\tau^2 s^2 + 2\zeta \tau s + 1}, \beta > 0$	$\frac{2\zeta \tau}{\tau_c + \beta}$	$2\zeta \tau$	$\frac{\tau}{2\zeta}$
E	$\frac{K}{s}$	$\frac{2}{\tau_c}$	$2\tau_c$	–
F	$\frac{K}{s(\tau s + 1)}$	$\frac{2\tau_c + \tau}{\tau_c^2}$	$2\tau_c + \tau$	$\frac{2\tau_c \tau}{2\tau_c + \tau}$
G	$\frac{K e^{-\theta s}}{\tau s + 1}$	$\frac{\tau}{\tau_c + \theta}$	$\tau$	–
H	$\frac{K e^{-\theta s}}{\tau s + 1}$	$\frac{\tau + \frac{\theta}{2}}{\tau_c + \frac{\theta}{2}}$	$\tau + \frac{\theta}{2}$	$\frac{\tau \theta}{2\tau + \theta}$
I	$\frac{K(\tau_3 s + 1)e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)}$	$\frac{\tau_1 + \tau_2 - \tau_3}{\tau_c + \theta}$	$\tau_1 + \tau_2 - \tau_3$	$\frac{\tau_1 \tau_2 - (\tau_1 + \tau_2 - \tau_3)\tau_3}{\tau_1 + \tau_2 - \tau_3}$
J	$\frac{K(\tau_3 s + 1)e^{-\theta s}}{\tau^2 s^2 + 2\zeta \tau s + 1}$	$\frac{2\zeta \tau - \tau_3}{\tau_c + \theta}$	$2\zeta \tau - \tau_3$	$\frac{\tau^2 - (2\zeta \tau - \tau_3)\tau_3}{2\zeta \tau - \tau_3}$
K	$\frac{K(-\tau_3 s + 1)e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)}$	$\frac{\tau_1 + \tau_2 + \frac{\tau_3 \theta}{\tau_c + \tau_3 + \theta}}{\tau_c + \tau_3 + \theta}$	$\tau_1 + \tau_2 + \frac{\tau_3 \theta}{\tau_c + \tau_3 + \theta}$	$\frac{\tau_3 \theta}{\tau_c + \tau_3 + \theta} + \frac{\tau_1 \tau_2}{\tau_1 + \tau_2 + \frac{\tau_3 \theta}{\tau_c + \tau_3 + \theta}}$
L	$\frac{K(-\tau_3 s + 1)e^{-\theta s}}{\tau^2 s^2 + 2\zeta \tau s + 1}$	$\frac{2\zeta \tau + \frac{\tau_3 \theta}{\tau_c + \tau_3 + \theta}}{\tau_c + \tau_3 + \theta}$	$2\zeta \tau + \frac{\tau_3 \theta}{\tau_c + \tau_3 + \theta}$	$\frac{\tau_3 \theta}{\tau_c + \tau_3 + \theta} + \frac{\tau^2}{2\zeta \tau + \frac{\tau_3 \theta}{\tau_c + \tau_3 + \theta}}$
M	$\frac{K e^{-\theta s}}{s}$	$\frac{2\tau_c + \theta}{(\tau_c + \theta)^2}$	$2\tau_c + \theta$	–
N	$\frac{K e^{-\theta s}}{s}$	$\frac{2\tau_c + \theta}{\left(\tau_c + \frac{\theta}{2}\right)^2}$	$2\tau_c + \theta$	$\frac{\tau_c \theta + \frac{\theta^2}{4}}{2\tau_c + \theta}$
O	$\frac{K e^{-\theta s}}{s(\tau s + 1)}$	$\frac{2\tau_c + \tau + \theta}{(\tau_c + \theta)^2}$	$2\tau_c + \tau + \theta$	$\frac{(2\tau_c + \theta)\tau}{2\tau_c + \tau + \theta}$

Vì hệ thống không có độ trễ và hàm truyền có dạng tương thích với mô hình của case

A  $\Rightarrow$  Chọn phương pháp tính toán của case A để tính các thông số PID .

Bước 1: Chọn  $\tau_c$ :

- *Several IMC guidelines for  $\tau_c$  have been published for the FOPTD .*

+  $\tau_c > 0.8\theta_p$ ;  $\tau_c > 0.1\tau_p$  (Rivera et al., 1986)

+  $\tau_p > \tau_c > \theta_p$  (Chien and Fruehauf, 1990)

$$\Rightarrow \text{Chọn } \tau_c = \tau_p * 0.8 = \tau * 0.8 = 0.0198 * 0.8 = 0.01584$$

Bước 2: Tính  $\tau_i$

$$\Rightarrow \tau_i = \tau = 0.0198$$

Bước 3: Tính  $K_c$

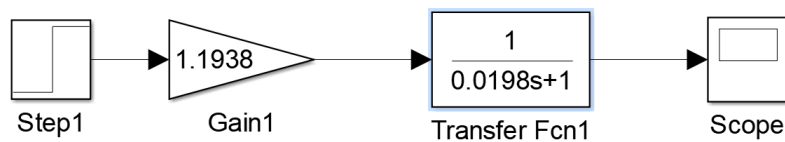
$$\Rightarrow K_c = (\tau / \tau_c) / K = 0.0198 / (1.1938 * 0.01584) = 1.04$$

Kết luận:

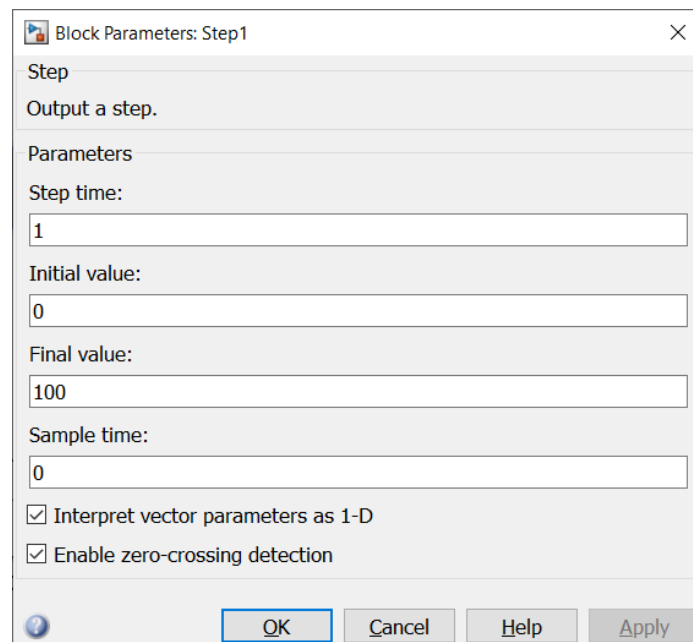
$$K_p = K_c = 1.04$$

$$K_i = K_c / \tau_i = 1.04 / 0.02 = 52$$

### 3.2.2 Kết quả chạy mô phỏng

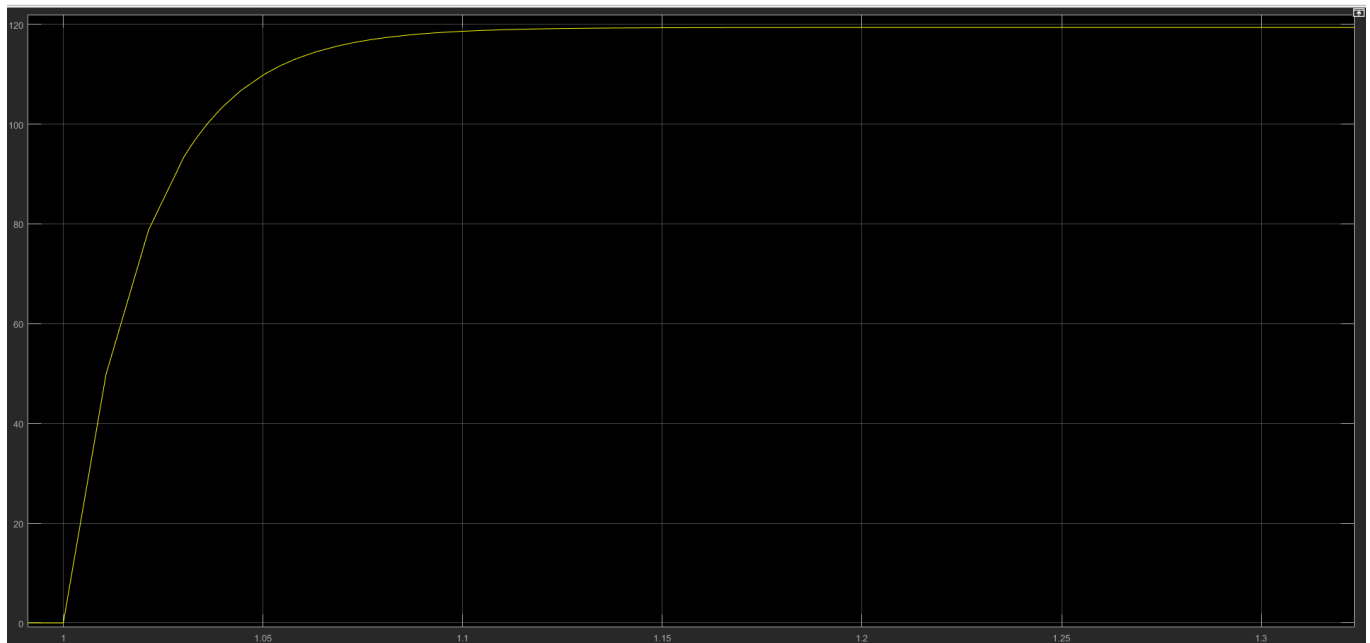


**Hình 6: Mô phỏng hàm truyền**



**Hình 7: Hàm step có giá trị tăng từ 0 lên 100**

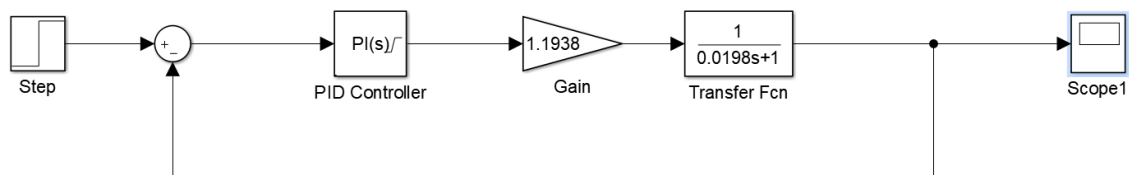




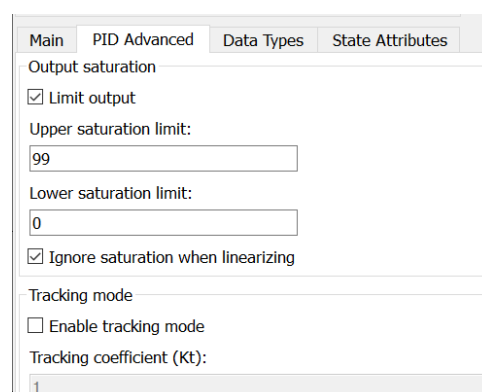
**Hình 8: Kết quả chạy mô phỏng trong 10 giây**

- Ta thấy kết quả mô phỏng khá sát với thực tế khi giá trị đạt được khoảng 119 rad/s (thực tế là 119.38 rad/s)
- Thời gian mà CV tăng từ 0 lên 119 rad/s khoảng 0,1s, hơi lệch so với thực tế (thực tế là 0,0315). Sai lệch có thể là do biểu đồ thời gian trên C# không chính xác so với thực tế cũng như thời gian lấy mẫu chưa đủ chính xác.

### Mô phỏng bộ điều khiển PI:



**Hình 9: Mô phỏng hệ thống điều khiển vận tốc dùng bộ PI**



**Hình 10: Thiết lập giá trị max, min cho output**

Controller: PI Form: Parallel

Time domain:  
☒ Continuous-time  
☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Source: internal [Compensator](#)

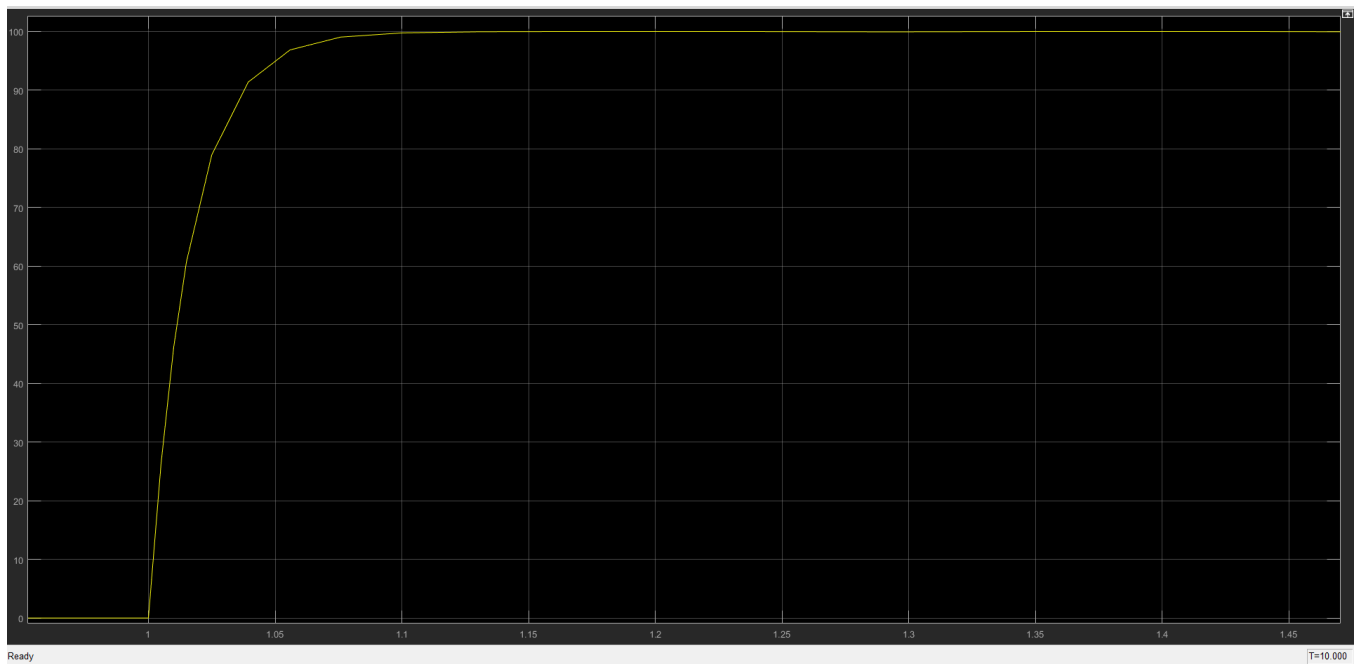
Proportional (P): 1.04

Integral (I): 52  $P + I \frac{1}{s}$

Tune...

Initial conditions

**Hình 11: Set thông số cho bộ điều khiển PI**



**Hình 12: Kết quả chạy mô phỏng bộ điều khiển vận tốc**

### 3.2.3. Code điều khiển vận tốc

```
#define pi 3.1415
#define p2r pi/2000

float Kp = 1.0;
float Ki = 0.3;
float Kd = 0.0;
float dt = 0.005;

#define MAX_PWM 99
#define MIN_PWM 0
```

```

void test_motor_control(enum_dir_t direction, int16_t pwm)
{
    // Turn on PC3
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3, (uint8_t)(direction));
    __HAL_TIM_SetCompare(&htim3, TIM_CHANNEL_2, pwm); // set pwm
}

```

```

int16_t PID(float pos_sp, float pos_cv)
{
    error = pos_sp - pos_cv;

    integral = integral + error * dt;

    int16_t mv = (int16_t)(Kp*error + Ki*integral);

    if (mv > 99)
    {
        mv = 99;
    }
    else if (mv < 0)
    {
        mv = 0;
    }

    return mv;
}

```

```

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) { // ngat timer 4 tinh van toc
    if(htim->Instance==TIM4) // ngat do timer 4 5ms
    {
        CurPos = PosCnt*2*pi+CountValue*p2r; // Position calculation
        Cnttmp = CntVel;
        CntVel = 0;
        RealVel = Cnttmp*6; //RPM
        CurVel = Cnttmp*pi/5; //rad/s

        if (run == true)
        {
            setpoint_rad = (setpoint * 2.0 * pi) / 60.0;
            cv_rad = (float)((RealVel * 2.0 * pi) / 60.0);
            mv_pwm = PID(setpoint_rad, cv_rad);
            test_motor_control(LEFT_DIRECTION, mv_pwm);
        }
        else
        {
            test_dc_motor_off();
        }

        return;
    }
}

```

### 3.2.4. Kết quả chạy thực nghiệm

- Cho Set point vận tốc thực tế là 100 rad/s, kết quả thu được ổn định ở mức khoảng 90 rad/s, vận tốc ổn định.

### 3.4. Module 4: Kiểm soát vị trí của hệ Servo một trục

#### 3.4.1. Tính toán hệ số cho bộ điều khiển vị trí (dùng bộ PD)

Ta có hàm truyền có dạng:

$$\frac{K}{(\tau s + 1)s}$$

Bước 1: Chọn  $\tau_c$ :

- *Several IMC guidelines for  $\tau_c$  have been published for the FOPTD .*
- +  $\tau_c > 0.8\theta_p$ ;  $\tau_c > 0.1\tau_p$  (Rivera et al., 1986)
- +  $\tau_p > \tau_c > \theta_p$  (Chien and Fruehauf, 1990)

$$\Rightarrow \text{Chọn } \tau_c = \tau_p/3 = 0.0067$$

Bước 2: Tính  $K_c$

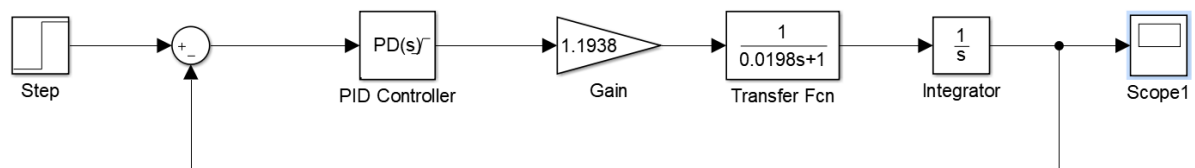
$$\Rightarrow K_c * K = (2 * \tau_c + \tau) / \tau_c^2$$

$$\Rightarrow K_c = 611.68$$

Bước 3: Tính  $\tau_d$

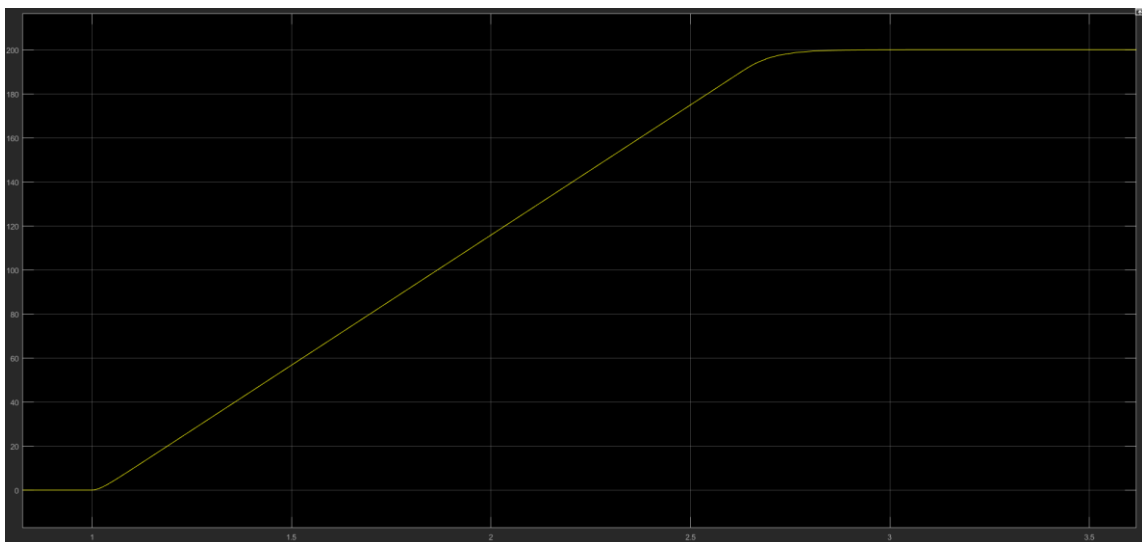
$$\Rightarrow \tau_d = (2 * \tau_c * \tau) / (2 * \tau_c + \tau) = 0.008$$

#### 3.4.2 Kết quả chạy mô phỏng



**Hình 13: Mô phỏng hệ thống điều khiển vị trí dùng bộ PD**

Main	PID Advanced	Data Types	State Attr
Output saturation			
<input checked="" type="checkbox"/> Limit output			
Upper saturation limit:			
<input type="text" value="99"/>			
Lower saturation limit:			
<input type="text" value="-99"/>			
<input checked="" type="checkbox"/> Ignore saturation when linearizing			



Hình 14: Kết quả mô phỏng hệ với set point là vị trí 200 rad

### 3.4.3. Code điều khiển vị trí

```
#define pi 3.1415
#define p2r pi/2000

float Kp = 2.0;
float Ki = 0.0;
float Kd = 0.1;
float dt = 0.005;

#define MAX_PWM 99
#define MIN_PWM 0
```



```

int16_t PID(float pos_sp, float pos_cv)
{
    error = pos_sp - pos_cv;

    integral = integral + error * dt;

    derivative = (error - prev_error) / dt;

    int16_t mv = (int16_t)(Kp*error + Ki*integral + Kd*derivative);

    prev_error = error;

    if (mv > 99)
    {
        mv = 99;
    }
    else if (mv < -99)
    {
        mv = -99;
    }

    // Return value
    return mv;
}

```

```

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) { // ngat timer 4 tinh van toc
    if(htim->Instance==TIM4) // ngat do timer 4 5ms
    {
        CurPos = PosCnt*2*pi+CountValue*p2r; // Position calculation
        Cnttmp = CntVel;
        CntVel = 0;
        RealVel = Cnttmp*6; //RPM
        CurVel = Cnttmp*pi/5; //rad/s

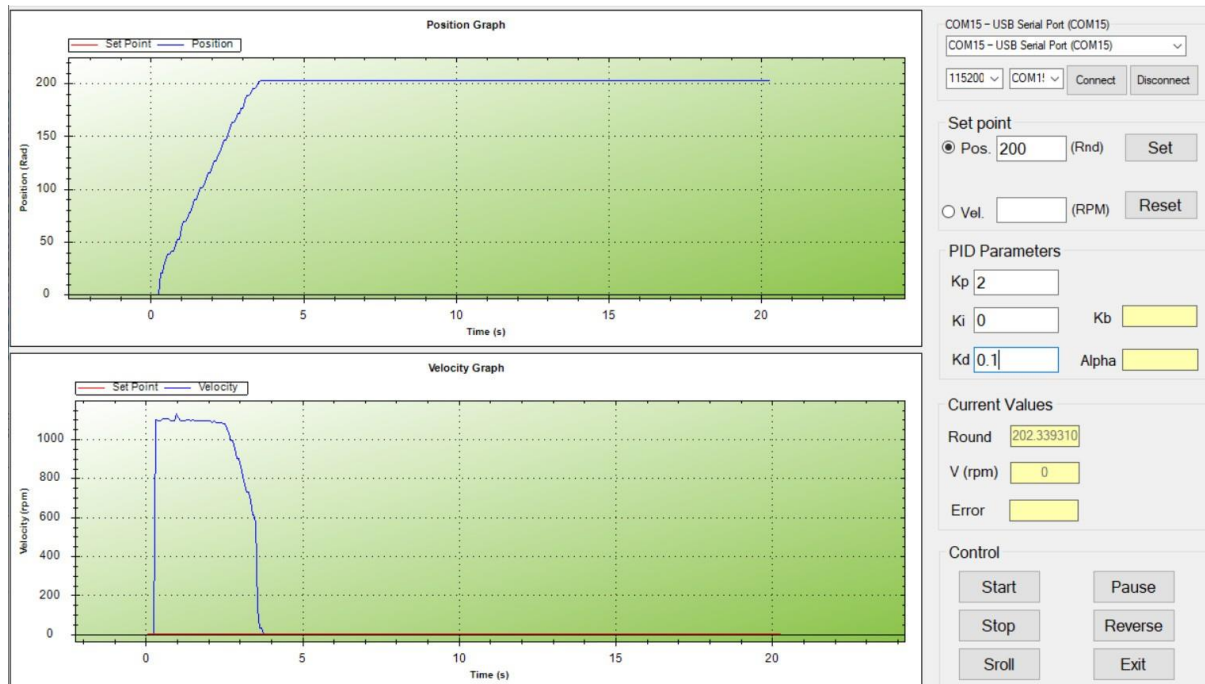
        if (run == true)
        {
            mv_pwm = PID(setpoint_rad, CurPos);

            if (mv_pwm > 0)
            {
                test_motor_control(LEFT_DIRECTION, mv_pwm);
            }
            else
            {
                test_motor_control(RIGHT_DIRECTION, mv_pwm*(-1));
            }
        }
        else
        {
            test_dc_motor_off();
        }

        return;
    }
}

```

### 3.4.4. Kết quả chạy thực nghiệm



**Hình 14: Kết quả chạy thực tế với setpoint là 200 rad**

- Ta thấy rằng kết quả điều khiển vị trí không xảy ra vọt lố, sai lệch khác nhỏ (khoảng 2 rad)