

## BÀI THÍ NGHIỆM 3

# ĐIỀU KHIỂN VỊ TRÍ ĐỘNG CƠ DC DÙNG BỘ TỰ CHỈNH STR (SELF-TUNING REGULATOR)

## I. MỤC ĐÍCH

Trong bài thí nghiệm này sinh viên sẽ khảo sát đáp ứng vị trí của động cơ DC theo thời gian khi tín hiệu đầu vào thay đổi để nhận dạng hàm truyền của động cơ. Dựa vào tham số nhận dạng sinh viên sẽ xây dựng bộ điều khiển thích nghi tự chỉnh định theo phương pháp đặt cực.

Ngoài ra, bài thí nghiệm còn giúp sinh viên hiểu rõ ảnh hưởng của hệ số quên lên tốc độ hội tụ của các tham số cần nhận dạng.

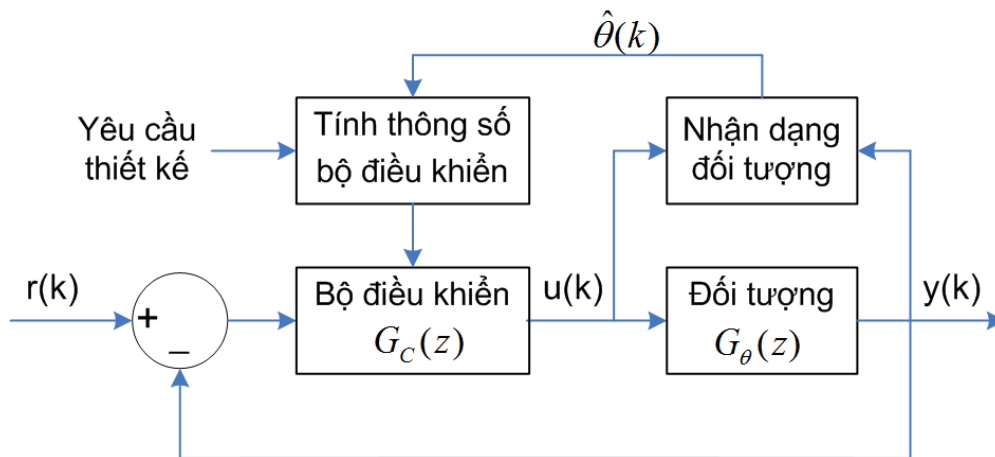
**Mục tiêu sau khi hoàn thành bài thí nghiệm này:**

- ☐ Biết cách xây dựng thuật toán nhận dạng off-line và on-line cho một đối tượng tuyến tính bậc 2.
- ☐ Biết cách xây dựng bộ điều khiển tự chỉnh định để điều khiển vị trí động cơ DC theo phương pháp đặt cực.

## II. CƠ SỞ LÝ THUYẾT

### 2.1. Điều khiển thích nghi tự chỉnh định STR (Self-Tuning Regulator)

Bộ điều khiển tự chỉnh định có sơ đồ cấu trúc như mô tả ở Hình 1. Đối tượng có hàm truyền rời rạc  $G_\theta(z)$ , trong đó  $\theta$  là một vector chứa các thông số chưa biết. Khi nhận dạng thu thập tín hiệu đầu vào  $u(k)$  và đầu ra  $y(k)$  của đối tượng để ước lượng các thông số của đối tượng  $\hat{\theta}(k)$ . Ngay khi hàm truyền rời rạc của đối tượng đã được tìm ra, khối Tính thông số bộ điều khiển sẽ tính ra được các thông số của bộ điều khiển dựa vào yêu cầu thiết kế cho trước (độ vọt lố, sai số xác lập,...).



Hình 1. Sơ đồ cấu trúc bộ điều khiển tự chỉnh định

## 2.2. Hàm truyền động cơ DC

Hàm truyền của động cơ DC với ngõ vào điện áp  $U$  và ngõ ra là tốc độ  $\Omega$ :

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K_T}{JL_a s^2 + (bL_a + JR_a)s + (bR_a + K_T K_b)} \quad (1)$$

Trong đó:

$R_a$	điện trở phần ứng
$L_a$	điện cảm phần ứng
$J$	mô men quán tính của trục động cơ
$b$	hệ số ma sát
$K_T$	hằng số mô men xoắn
$K_b$	hằng số sức phản điện (EMF)

Thông thường, điện cảm của động cơ tương đối nhỏ so với quán tính của động cơ nên có thể bỏ qua ở tần số thấp. Do đó, hàm truyền tốc độ động cơ DC có thể xấp xỉ như hệ thống bậc 1:

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K_T}{JR_a s + (bR_a + K_T K_b)} = \frac{K}{\tau s + 1} \quad (2)$$

Trong đó:

$$K = \frac{K_T}{(bR_a + K_T K_b)}, \quad \tau = \frac{JR_a}{(bR_a + K_T K_b)} \quad (3)$$

Từ hàm truyền tốc độ, ta suy ra hàm truyền vị trí động cơ DC:

$$G(s) = \frac{\Theta(s)}{U(s)} = \frac{K}{s(\tau s + 1)} \quad (4)$$

Vì vậy, hàm truyền rời rạc vị trí động cơ có dạng:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (5)$$

## 2.3. Nhận dạng hàm truyền rời rạc

Để nhận dạng các tham số  $a_1, a_2, b_1, b_2$  của hàm truyền vị trí động cơ ở (5), ta khai triển (5) dưới dạng đệ quy:

$$-a_1 y(k-1) - a_2 y(k-2) + b_1 u(k-1) + b_2 u(k-2) = y(k) \quad (6)$$

$$\text{Đặt: } \varphi(k) = [-y(k-1) \quad -y(k-2) \quad u(k-1) \quad u(k-2)] \quad (7)$$

$$\theta = [a_1 \quad a_2 \quad b_1 \quad b_2]^T \quad (8)$$

Thuật toán ước lượng đệ quy để tìm các thông số  $\hat{\theta}(k)$  ở thời điểm  $k$  như sau:

$$\begin{aligned}
 \hat{\theta}(k) &= \hat{\theta}(k-1) + L(k)\varepsilon(k) \\
 \varepsilon(k) &= y(k) - \varphi^T(k)\hat{\theta}(k-1) \\
 L(k) &= \frac{P(k-1)\varphi(k)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \\
 P(k) &= \frac{1}{\lambda} \left( P(k-1) - \frac{P(k-1)\varphi(k)\varphi^T(k)P(k-1)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \right)
 \end{aligned} \tag{9}$$

Trong đó các giá trị khởi tạo ban đầu được cài đặt như sau:

$$\begin{aligned}
 \hat{\theta}(0) &= rand(4,1) \\
 P(0) &= 10^5 eye(4)
 \end{aligned} \tag{10}$$

Hệ số quên  $\lambda$  ( $0 < \lambda \leq 1$ ) được cài đặt tùy thuộc vào tốc độ biến đổi theo thời gian của các thông số  $\theta$  trong hàm truyền  $G(z)$ . Nếu  $G(z)$  ít thay đổi theo thời gian thì ta cài đặt giá trị  $\lambda$  gần bằng 1,  $G(z)$  thay đổi theo thời gian càng nhiều thì ta cài đặt giá trị  $\lambda$  càng nhỏ. Thông thường, hệ số quên có giá trị trong khoảng  $\lambda = 0.8 \div 1$ .

## 2.4. Xây dựng bộ điều khiển theo phương pháp đặt cực

Khi hàm truyền đối tượng đã được nhận dạng, ta sẽ xây dựng bộ điều khiển cho đối tượng theo các thông số đã nhận dạng đó. Tùy theo yêu cầu thiết kế, ta có thể xây dựng bộ điều khiển theo các phương pháp khác nhau cho đối tượng như: PID, PID Ziegler-Nichols, bộ điều khiển đặt cực, bộ điều khiển LQ, .... Trong bài thí nghiệm này, sinh viên sẽ xây bộ điều khiển theo phương pháp đặt cực sử dụng 2 cách:

- Sử dụng cặp cực của hệ liên tục sau đó chuyển sang cặp cực hệ rời rạc
- Sử dụng trực tiếp cặp cực hệ rời rạc.

### 2.4.1. Sử dụng cặp cực hệ liên tục

- Yêu cầu thiết kế: Đáp ứng của hệ thống phải theo cặp cực quyết định

$$S_{1,2}^* = -\xi\omega \pm j\omega\sqrt{1-\xi^2} \tag{11}$$

- Luật điều khiển sẽ là:

$$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + (1-\gamma)u(k-1) + \gamma u(k-2) \tag{12}$$

Trong đó:

$$\begin{aligned}
 d_1 &= -2\exp(-\xi\omega T) \cos\left(\omega T \sqrt{1-\xi^2}\right) \quad \text{neu } \xi < 1 \\
 d_1 &= -2\exp(-\xi\omega T) \cosh\left(\omega T \sqrt{\xi^2-1}\right) \quad \text{neu } \xi \geq 1 \\
 d_2 &= \exp(-2\xi\omega T) \\
 r_1 &= (\hat{b}_1 + \hat{b}_2)(\hat{a}_1\hat{b}_1\hat{b}_2 - \hat{a}_2\hat{b}_1^2 - \hat{b}_2^2) \\
 s_1 &= \hat{a}_2 \left[ (\hat{b}_1 + \hat{b}_2)(\hat{a}_1\hat{b}_2 - \hat{a}_2\hat{b}_1) + \hat{b}_2(\hat{b}_1d_2 - \hat{b}_2d_1 - \hat{b}_2) \right]
 \end{aligned}$$

$$\gamma = q_2 \frac{\hat{b}_2}{\hat{a}_2}, \quad q_2 = \frac{s_1}{r_1}, \quad q_1 = \frac{\hat{a}_2}{\hat{b}_2} - q_2 \left( \frac{\hat{b}_1}{\hat{b}_2} - \frac{\hat{a}_1}{\hat{a}_2} + 1 \right),$$

$$q_0 = \frac{1}{\hat{b}_1} (d_1 + 1 - \hat{a}_1 - \gamma)$$

- $T$  là chu kỳ lấy mẫu hệ thống, thông thường trong điều khiển vị trí động cơ DC ta chọn  $T = 1 \div 5ms$ . Tùy theo yêu cầu về đáp ứng quá độ, ta chọn giá trị  $\xi$  và  $\omega$  cho phù hợp. Nếu cần đáp ứng nhanh ta cài đặt  $\xi$  nhỏ, nếu cần độ vọt lố nhỏ ta cài đặt  $\xi$  lớn.

#### 2.4.2. Sử dụng cặp cực hệ rời rạc

- Yêu cầu thiết kế: Đáp ứng của hệ thống phải theo 4 cực:

$$\begin{aligned} z_{1,2}^* &= \alpha \pm j\beta \\ z_{3,4}^* &= \alpha \end{aligned} \quad (13)$$

- Luật điều khiển sẽ là:

$$\begin{aligned} u(k) &= r_0 w(k) - q_0 y(k) - q_1 y(k-1) - q_2 y(k-2) \\ &\quad - p_1 u(k-1) - p_2 u(k-2) \end{aligned} \quad (14)$$

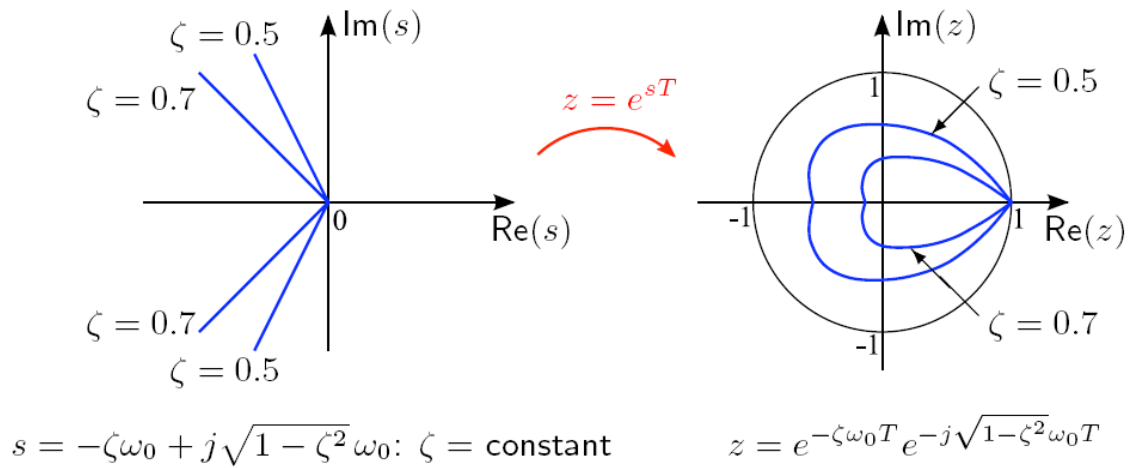
Trong đó:

$$\begin{aligned} r_0 &= \frac{x_1 + x_2 - x_3 + x_4}{\hat{b}_1 + \hat{b}_2}, \quad p_1 = \frac{r_5 - r_1}{r_1}, \quad p_2 = -\frac{r_5}{r_1} \\ q_0 &= r_0 - \frac{r_2 + r_3 + r_4}{r_1}, \quad q_1 = \frac{r_2 + r_3 + r_4 - r_6 - r_7}{r_1}, \quad q_2 = \frac{r_6 + r_7}{r_1} \end{aligned}$$

$$\begin{aligned} r_1 &= (\hat{b}_1 + \hat{b}_2)(\hat{a}_1 \hat{b}_1 \hat{b}_2 - \hat{a}_2 \hat{b}_1^2 - \hat{b}_2^2) & r_2 &= \hat{a}_1 \hat{b}_2 [\hat{b}_1 (x_2 - x_3 + x_4) - \hat{b}_2 x_1] \\ r_3 &= \hat{a}_2 \hat{b}_1 [\hat{b}_2 x_1 - \hat{b}_1 (x_2 - x_3 + x_4)] & r_4 &= (\hat{b}_1 + \hat{b}_2) [\hat{b}_1 x_4 + \hat{b}_2 (x_3 - x_4)] \\ r_5 &= \hat{b}_1 (\hat{b}_1^2 x_4 + \hat{b}_1 \hat{b}_2 x_3 + \hat{b}_2^2 x_2) - \hat{b}_2^3 x_1 & r_6 &= \hat{b}_1^2 (\hat{a}_2 x_3 + \hat{a}_1 x_4 - \hat{a}_2 x_4) \\ r_7 &= \hat{b}_2 [\hat{b}_1 (\hat{a}_1 x_4 + \hat{a}_2 x_2 - x_4) - \hat{b}_2 (\hat{a}_2 x_1 + x_4)] \end{aligned}$$

$$\begin{aligned} x_1 &= -4\alpha + 1 - \hat{a}_1 & x_2 &= 6\alpha^2 + \beta^2 + \hat{a}_1 - \hat{a}_2 \\ x_3 &= 2\alpha(2\alpha^2 + \beta^2) - \hat{a}_2 & x_4 &= \alpha^2(\alpha^2 + \beta^2) \end{aligned}$$

- $w(k)$  là tín hiệu đặt. Giá trị cặp cực  $z_{1,2}^*$  sẽ quyết định đến chất lượng điều khiển và phải thỏa mãn  $|z_{1,2}^*| < 1$  để hệ thống ổn định. Mối quan hệ giữa cực ở miền liên tục và cực ở miền rời rạc được mô tả như ở Hình 2. Do đó, khi thiết kế ta có thể chọn cặp cực ở miền liên tục  $s_{1,2}^*$ , sau đó tính  $z_{1,2}^* = e^{Ts_{1,2}^*}$  để tìm cặp cực ở miền rời rạc.



Hình 2. Mối quan hệ giữa cực ở miền liên tục và miền rời rạc

### III. MÔ TẢ THÍ NGHIỆM

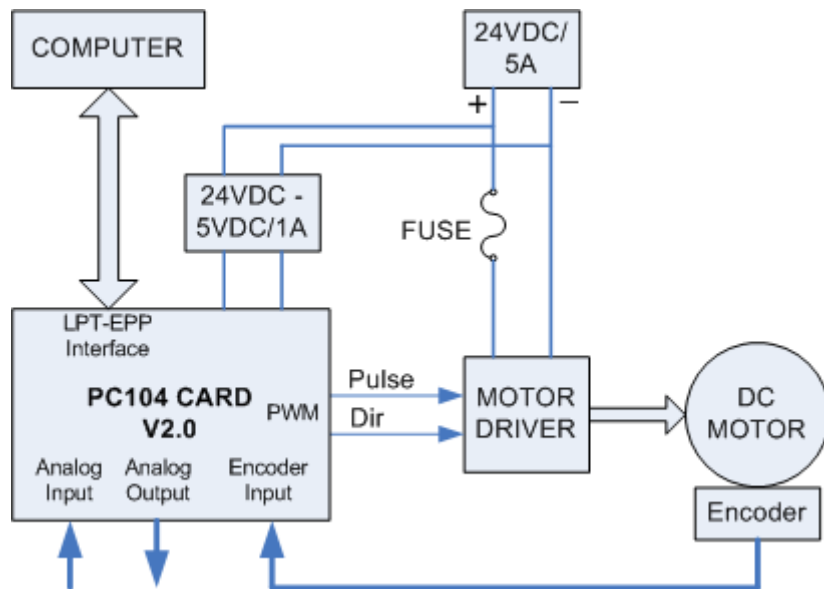
#### 3.1. Phần cứng

Sơ đồ khối phần cứng được mô tả như ở Hình 3. Trong đó, bo mạch PC104CARD là phần cứng trung tâm trong bài thí nghiệm. Bo mạch PC104CARD xây dựng trên nền tảng FPGA của Altera có khả năng giao tiếp với máy tính thông qua phần mềm Matlab/ Simulink/ Real-time Windows Target với các tính năng sau:

- Digital Input / Digital Output: 8 DI / 8 DO.
- Điều rộng xung (PWM): 4 kênh độ phân giải 0.1% (~ 10bit).
- Encoder: 4 kênh encoder x4.
- Analog Input: 4 kênh ADC 12bit tầm đo 0 – 10V.
- Analog Output: 3 kênh DAC 12bit tầm 0 – 2V.
- Giao tiếp với máy tính thông qua cổng máy in theo chuẩn EPP.

Động cơ DC sử dụng trong bài thí nghiệm có điện áp định mức 24VDC và encoder độ phân giải 400 xung/vòng. Tín hiệu điều rộng xung **Pulse** và hướng **Dir** sẽ điều khiển động cơ chạy với tốc độ và chiều quay mong muốn:

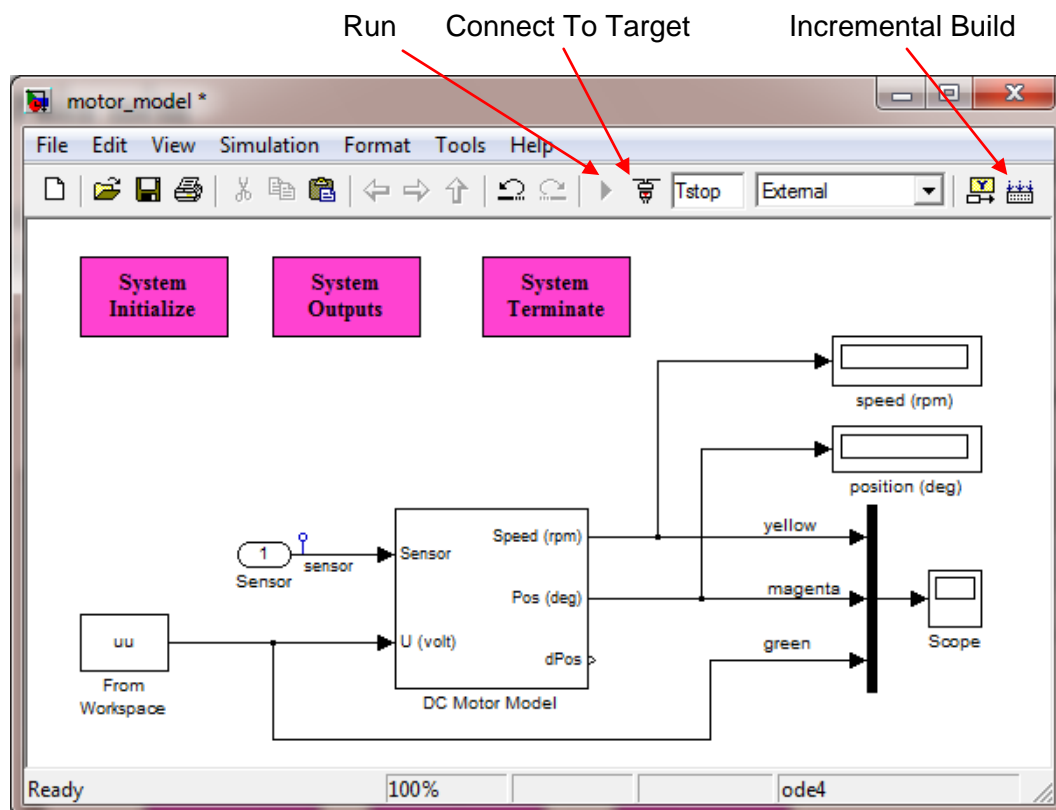
- Điện áp cấp cho động cơ sau khi qua Motor Driver sẽ bằng  $T_{PWM} * 24/1000$  (V), trong đó  $T_{PWM}$  là giá trị điều rộng xung từ (0 – 1000) tương ứng với độ phân giải 0.1%.
- Tốc độ của động cơ (đơn vị vòng/phút) được tính theo phương pháp đo thời gian một chu kỳ xung encoder A và hướng quay được xác định thông qua so sánh lệch pha của 2 tín hiệu encoder A và B.
- Vị trí của động cơ được tính toán qua mạch encoder x4, có nghĩa là khi động cơ quay 1 vòng ( $360^\circ$ ) ta sẽ đọc được  $4 \times 400 = 1600$  xung.



Hình 3. Sơ đồ phần cứng bài thí nghiệm

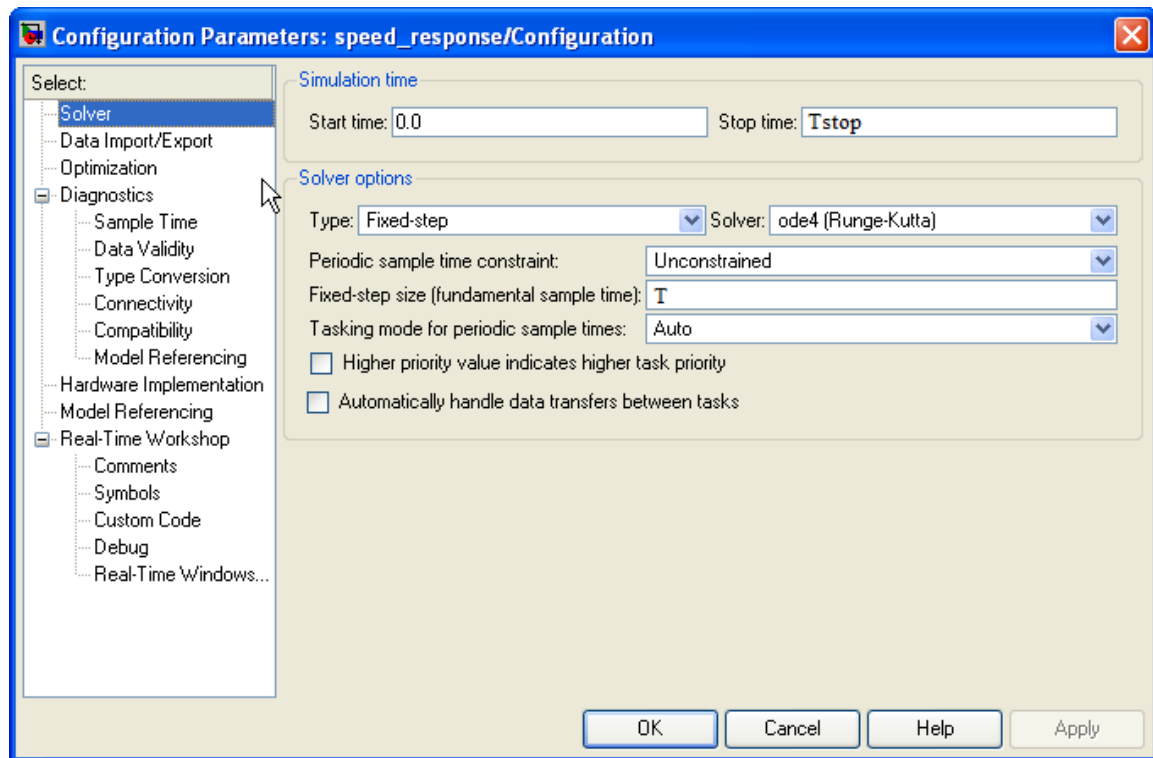
### 3.2. Phần mềm

Phần mềm sử dụng trong các bài thí nghiệm này là bộ phần mềm Matlab/ Simulink/ Real-time Windows Target. Công cụ Real-time Windows Target cho phép mô hình Simulink có khả năng kết nối với phần cứng bên ngoài và chạy theo thời gian thực. Để biên dịch và chạy mô hình Simulink liên kết với phần cứng sinh viên phải thực hiện các bước trình tự sau:



Hình 4. Mô hình Simulink kết nối phần cứng

- Tạo hoặc mở một file simulink như ở Hình 4.
- Vào menu Simulation -> Configuration, chọn mục Solver để cài đặt các thông số về thời gian mô phỏng (Simulation time) và phương pháp mô phỏng (Solver Options) như ở Hình 5.
- Vào menu Tool -> Real-Time Workshop -> Build Model (hoặc nhấn chuột trái vào biểu tượng Incremental Build) để biên dịch mô hình.
- Sau khi mô hình biên dịch thành công, vào menu Simulation -> Connect To Target (hoặc nhấn chuột trái vào biểu tượng Connect To Target) để kết nối mô hình Simulink tới phần cứng.
- Vào menu Simulation -> Run (hoặc biểu tượng Run) để chạy mô hình.



Hình 5. Cài đặt các thông số mô phỏng

## IV. CHUẨN BỊ TRƯỚC THÍ NGHIỆM

### 4.1. Thu thập dữ liệu đầu vào – đầu ra và nhận dạng hệ thống

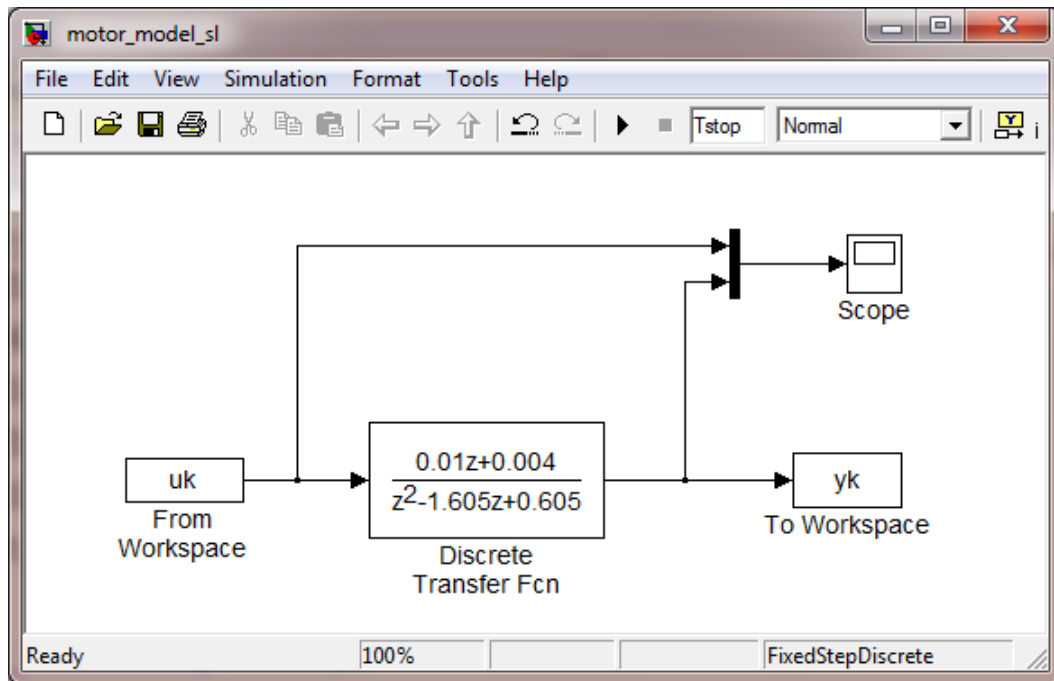
Để thực hiện tốt thí nghiệm trên lớp, sinh viên cần chuẩn bị trước phần lý thuyết và thực hiện mô phỏng trên máy tính.

Xây dựng mô hình Simulink để thu thập dữ liệu vị trí động cơ như ở Hình 6. Trong đó, đối tượng động cơ có hàm truyền:

$$G(z) = \frac{0.01z + 0.004}{z^2 - 1.605z + 0.605} \quad (15)$$

- Khối 'From Workspace' có tên biến là  $uk$  và khối 'To Workspace' có tên biến là  $yk$ . Cài đặt *Sample time* cho cả 2 khối này là 0.01s.

- Cài đặt thời gian lấy mẫu là  $T$  và thời gian mô phỏng là  $T_{stop}$ . Hai thông số này ta sẽ gán giá trị trong Matlab m-file.



Hình 6. Mô hình thu thập số liệu vị trí động cơ

- Tạo file mô hình simulink như ở Hình 6. Đặt tên file là 'motor\_model\_sl.mdl'.
- Tạo Matlab m-file với tên 'motor\_model\_init.m' để xuất tín hiệu  $uk$ , thu thập tín hiệu ngõ ra  $yk$ , và viết chương trình nhận dạng hàm truyền đối tượng như chương trình bên dưới.
- Khảo sát với nhiều giá trị  $\lambda$  khác nhau từ 0.8 đến 1. Nhận xét tốc độ hội tụ và sai số của  $\theta$  khi  $\lambda$  thay đổi.

```
% Cài đặt chu kỳ lấy mẫu T và thời gian mô phỏng Tstop
N = 1000;
T = 0.01;
Tstop = (N-1)*T;

% Tạo chuỗi tín hiệu điện áp đầu vào ngẫu nhiên uk
tt = 0:T:T*(N-1);
u = zeros(1,N);

u(1:90) = 1;
u(150:200) = 0.9;
u(270:350) = 0.8;
u(400:440) = 1;
u(480:500) = 0.7;
u(700:750) = 1;
u(800:900) = 0.6;
uk = [tt; u]';
```



```
% Chạy mô phỏng mô hình simulink vừa tạo ở Hình 6
sim('motor_model_sl.mdl');

% Vẽ tín hiệu đầu vào - đầu ra
plot(tt,u,'r',tt,yk,'b')
legend('uk','yk')

% Nhận dạng đối tượng với hệ số quên lamda
Theta = rand(4,N);
P = 1e5*eye(4);
lamda = 0.96;
for i=3:N
    PHI = [-yk(i-1) -yk(i-2) u(i-1) u(i-2)]';
    e = yk(i) - PHI'*Theta(:,i-1);
    L = P*PHI / (lamda + PHI'*P*PHI);
    P = 1/lamda * (P - P*PHI*PHI'*P / (lamda + PHI'*P*PHI));
    Theta(:,i) = Theta(:,i-1) + L*e;
end

% Vẽ đồ thị các hệ số a1, a2, b1, b2 vừa nhận dạng
figure;
subplot(2,1,1);
plot(tt,Theta(1,:), 'r',tt,Theta(2,:), ':r');
legend('a_1','a_2')
grid on
subplot(2,1,2);
plot(tt,Theta(3,:), 'b',tt,Theta(4,:), ':b')
legend('b_1','b_2');
grid on

% Hiện thị giá trị Theta cuối cùng
Theta(:,N)
```

#### 4.2. Xây dựng bộ điều khiển tự chỉnh STR theo phương pháp đặt cực

Sinh viên xây dựng mô hình simulink cho bộ điều khiển tự chỉnh theo phương pháp đặt cực miền liên tục như Hình 7 (Xem hướng dẫn ở Mục II). Trong đó, hàm truyền đối tượng cho trước ở (15). Yêu cầu sinh viên xây dựng thuật toán nhận dạng online ở khối 'Identification' và bộ điều khiển theo phương pháp đặt cực ở khối 'Self-Tuning Regulator'.

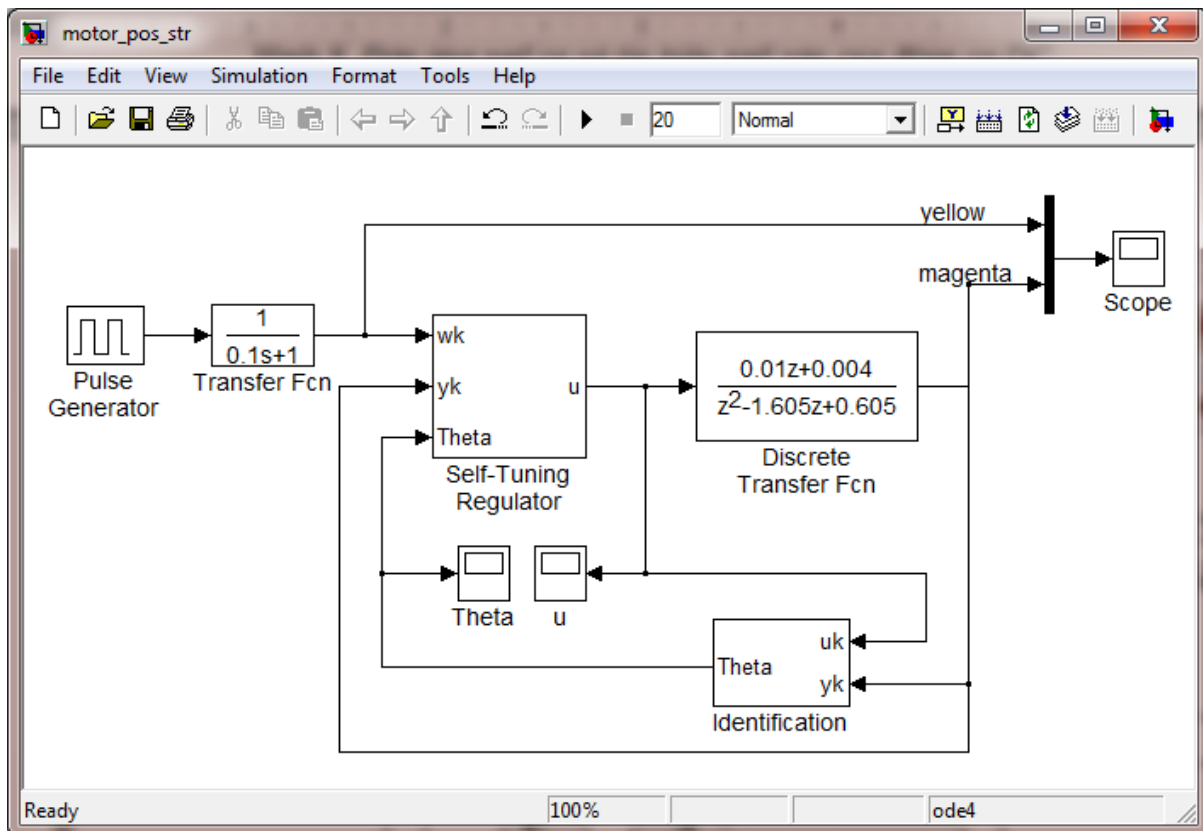
Hình 8 và 9 mô tả chi tiết cấu trúc khối 'Identification' và khối 'Self-Tuning Regulator', sinh viên tham khảo để xây dựng lại. Sinh viên sử dụng khối 'Embedded MATLAB Function' trong thư viện Simulink / User-Defined Functions để viết thuật toán nhận dạng và bộ điều khiển. Đặt tên hàm trong khối như sau:

- $[P, \Theta] = \text{Uocluong}(u_k, y_k, P\_ , \Theta\_ )$
- $u = \text{STR\_pp1}(\Theta, e, u\_ )$

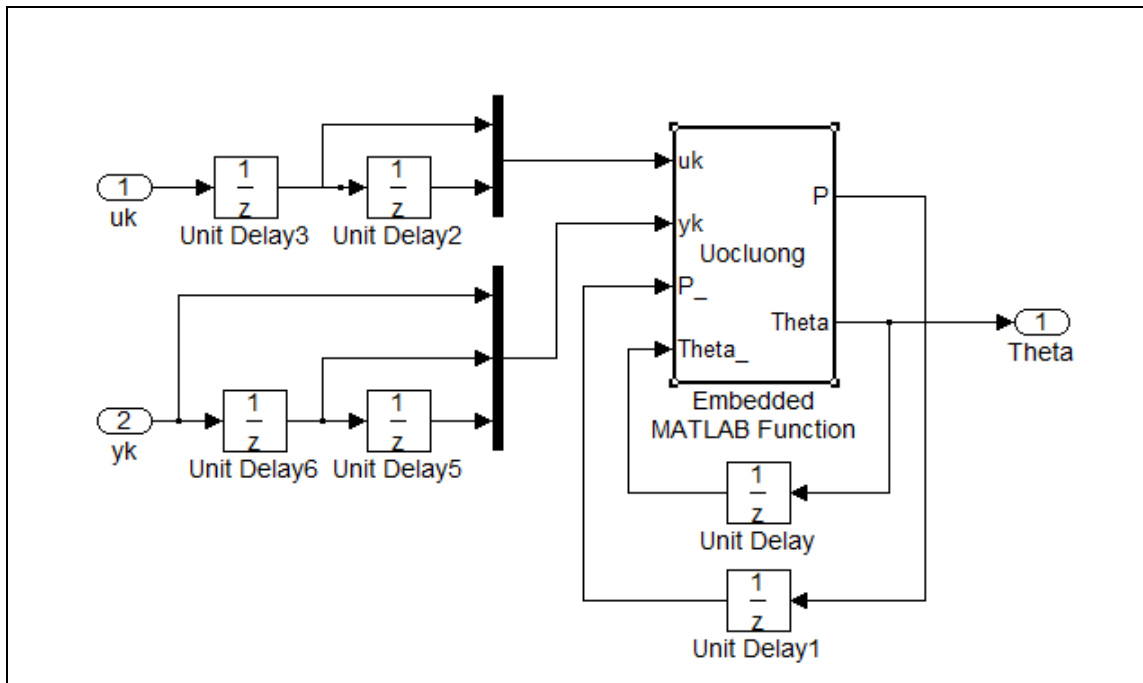
- Tạo file mô hình simulink như ở Hình 7, trong đó khối ‘Self-Tuning Regulator’ sử dụng bộ điều khiển tự chỉnh theo phương pháp đặt cực miền liên tục (Xem 2.4.1). Khối ‘Pulse Generator’ có  $Amplitude = 1$ ,  $Period = 4$ ,  $Pulse Width = 50$ .
- Cài đặt thời gian lấy mẫu  $T = 0.005s$  và thời gian mô phỏng 20s.
- Khảo sát với nhiều giá trị của  $\xi$ ,  $\omega$ . Nhận xét ảnh hưởng của cặp cực  $S_{1,2}^* = -\xi\omega \pm j\omega\sqrt{1-\xi^2}$  lên chất lượng bộ điều khiển. Mở Scope xem đáp ứng ngõ ra để thấy bộ điều khiển có tính thích nghi.
- Lặp lại Bước a) đến c) cho bộ điều khiển tự chỉnh theo phương pháp đặt cực miền rời rạc. Nhận xét ảnh hưởng của cực  $z_{1,2}^* = \alpha \pm j\beta$  lên chất lượng bộ điều khiển.
- Thay đổi mẫu số của hàm truyền đối tượng thành:

$$G(z) = \frac{0.02z + 0.004}{z^2 - 1.805z + 0.805} \quad (16)$$

Lặp lại Bước a) đến d) để chứng minh bộ điều khiển tự chỉnh định vẫn có chất lượng điều khiển tốt khi đối tượng thay đổi.



Hình 7. Hệ thống điều khiển tự chỉnh theo phương pháp đặt cực

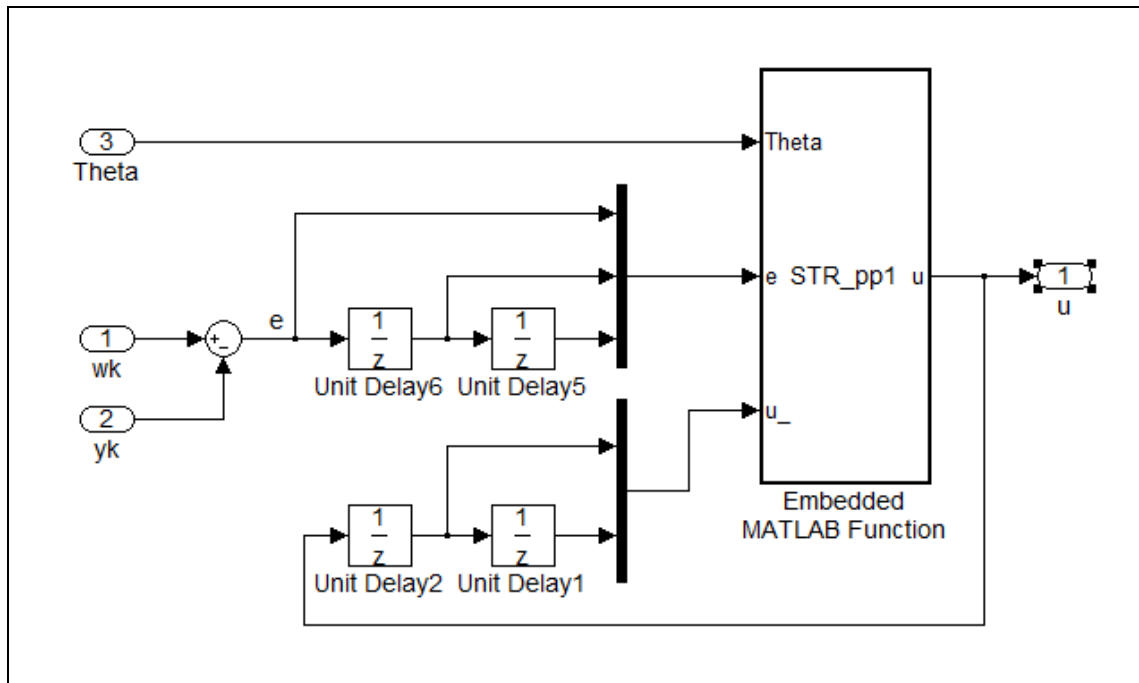


```
% Hàm nhận dạng thông số đối tượng
function [P,Theta] = Uocluong(uk,yk,P_,Theta_)

lamda = 0.96;

PHI = [-yk(2); -yk(3); uk(1); uk(2)];
e = yk(1) - PHI'*Theta_;
L = P_*PHI / (lamda + PHI'*P_*PHI);
P = 1/lamda*(P_ - P_*PHI*PHI'*P_ / (lamda + PHI'*P_*PHI));
Theta = Theta_ + L*e;
```

Hình 8. Sơ đồ khối 'Identification'. Giá trị khởi tạo của  $\Theta$  và  $P$  ở (10) được cài đặt trong khối 'Unit Delay' và 'Unit Delay1'



```
% Ham tinh thong so bo dieu khien theo pp dat cuc lien tuc
function u = STR_pp1(Theta,e,u_)
T0 = 0.005;
xi = 0.99;
wn = 20;

% Gan lai thong so de thuan tien viec tinh toan
a1 = Theta(1);
a2 = Theta(2);
b1 = Theta(3);
b2 = Theta(4);

% Tinh thong so bo dieu khien
if(xi<1)
    d1 = ...; % xi<1
else
    d1 = ...; % xi>1
end

d2 = ...;
r1 = ...;
s1 = ...;
q2 = ...;
gama = ...;
q1 = ...;
q0 = ...;

% Luat dieu khien
u = q0*e(1) + q1*e(2) + q2*e(3) + (1-gama)*u_(1) +
gama*u_(2);
```

Hình 9. Sơ đồ khối 'Self-Tuning Regulator'

## V. THỰC HIỆN THÍ NGHIỆM

### 5.1. Nhận dạng hàm truyền đối tượng

Trong phần này, sinh viên sẽ thu thập tín hiệu đầu vào và đầu ra của đối tượng và áp dụng công thức ở Mục 2.1 để nhận dạng hàm truyền vị trí của động cơ DC.

#### *Các bước thí nghiệm:*

- Mở file *motor\_model\_init.m* và chạy để khởi tạo tín hiệu điện áp đầu vào  $u_k$ , các thông số thời gian lấy mẫu  $T$  và thời gian mô phỏng  $T_{stop}$ . Sinh viên không cần cài đặt lại phần thiết lập trong Simulink -> Configuration.  
*Lưu ý: Khi chạy file này lần đầu tiên chương trình sẽ báo lỗi nhưng vẫn khởi tạo được giá trị bình thường.*
- Biên dịch và chạy file *motor\_model.mdl* để thu thập số liệu.
- Chạy lại file *motor\_model\_init.m* để tìm hàm truyền vị trí của động cơ. Thay đổi giá trị  $\lambda$  trong file *motor\_model\_init.m* sao cho đồ thị hiển thị các thông số ước lượng thẳng và hội tụ nhất.
- Lặp lại bước b) và c) với 5 lần chạy và ghi kết quả vào Bảng 1. Sau đó, tính giá trị trung bình của hàm truyền.

Bảng 1. Thông số hàm truyền  $G(z)$

Lần chạy	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_1$	$\hat{b}_2$
1				
2				
3				
4				
5				
<b>Trung bình</b>				

### 5.2. Xây dựng bộ điều khiển tự chỉnh theo phương pháp đặt cực miền liên tục

#### *Các bước thí nghiệm:*

- Mở file *motor\_pos\_str\_pp1.mdl*
- Mở khối 'Identification'. Tham khảo code ở Hình 8 để viết bổ sung vào cho đầy đủ.
- Mở khối 'Self-Tuning Regulator'. Tham khảo code ở Hình 9 để viết bổ sung vào cho đầy đủ (công thức (12))
- Sau khi hoàn thành việc viết code, biên dịch và chạy chương trình.
- Mở Scope để xem đáp ứng. Nhận xét chất lượng của hệ thống (độ vọt lố, thời gian quá độ, sai số xác lập).
- Thay đổi cặp cực quyết định bằng cách thay đổi giá trị của  $\xi$ ,  $\omega$  sao cho đáp ứng ngõ ra không có vọt lố. Nhận xét ảnh hưởng của  $\xi$ ,  $\omega$  lên chất lượng hệ thống.

### 5.3. Xây dựng bộ điều khiển tự chỉnh theo phương pháp đặt cực miền rời rạc

#### *Các bước thí nghiệm:*

- a) Mở file *motor\_pos\_str\_pp2.mdl*
- b) Mở khối 'Identification'. Tham khảo code ở Hình 8 để viết bổ sung vào cho đầy đủ.
- c) Mở khối 'Self-Tuning Regulator'. Xem hướng dẫn ở Mục 2.4.2 để xây dựng bộ điều khiển tự chỉnh theo phương pháp đặt cực miền rời rạc (công thức (14)).
- d) Sau khi hoàn thành việc viết code, biên dịch và chạy chương trình.
- e) Mở Scope để xem đáp ứng. Nhận xét chất lượng của hệ thống (độ vọt lố, thời gian quá độ, sai số xác lập).
- f) Thay đổi cặp cực quyết định bằng cách thay đổi giá trị của  $\alpha$ ,  $\beta$  sao cho đáp ứng ngõ ra không có vọt lố. Nhận xét ảnh hưởng của  $\alpha$ ,  $\beta$  lên chất lượng hệ thống.

## Báo cáo thí nghiệm

Họ và tên:

Nhóm:

Ngày:

1. Từ bảng số liệu ở Bảng 1, viết hàm truyền rời rạc vị trí động cơ. Nhận xét ảnh hưởng của hệ số quên  $\lambda$  lên sai số ước lượng.

.....

.....

.....

.....

.....

.....

2. Dựa vào kết quả thí nghiệm ở Mục 5.2, nhận xét chất lượng của hệ thống (độ vọt lố, thời gian quá độ, sai số xác lập).

.....

.....

.....

.....

.....

3. Dựa vào kết quả thí nghiệm ở Mục 5.3, nhận xét chất lượng của hệ thống (độ vọt lố, thời gian quá độ, sai số xác lập).

.....

.....

.....

.....

.....