

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**BÁO CÁO THỰC HÀNH**  
**MÔN MÔ HÌNH HÓA VÀ MÔ PHỎNG**



**LAB 1: MODELING AND SIMULATION IN MATLAB/SIMULINK**  
**RLC CIRCUIT – MÔ PHỎNG HOẠT ĐỘNG MẠCH RLC**

**Giảng viên hướng dẫn : Nguyễn Hồng Thịnh**

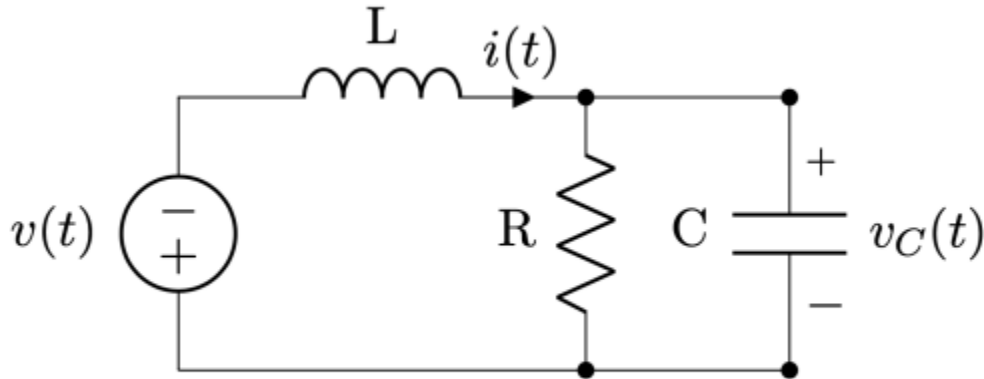
**Sinh viên thực hiện : Hoàng Nhật Minh**

**Mã sinh viên : 21020696**

**Lớp : ELT2031E 21**

## 1. Prelab

### 1.1. Modeling a simple RLC circuit



Hình 1: Mạch RLC

\*Xác định phương trình vi phân mô tả mối quan hệ giữa dòng điện  $i(t)$ , và điện áp ở tụ điện  $v_C(t)$ ?

Ta có:

- Phương trình dòng điện: 
$$\begin{cases} i = i_L = i_R + i_C \\ i_C = C \cdot \frac{dv_C}{dt} \\ i_R = \frac{v_R}{R} = \frac{v_C}{R} \end{cases} \Leftrightarrow i = \frac{v_C}{R} + C \cdot \frac{dv_C}{dt}$$

$$\Leftrightarrow \frac{di}{dt} = \frac{1}{R} \cdot \frac{dv_C}{dt} + C \cdot \frac{d^2v_C}{dt^2}$$

- Phương trình điện áp: 
$$\begin{cases} V = V_L + V_C \\ V_R = V_C \end{cases} \Leftrightarrow V = L \cdot \frac{di}{dt} + V_C = L \cdot \frac{1}{R} \cdot \frac{dv_C}{dt} + L \cdot C \cdot \frac{d^2v_C}{dt^2} + V_C$$

→ phương trình vi phân mô tả mối quan hệ giữa dòng điện  $i(t)$ , và điện áp ở tụ điện  $v_C(t)$ :

$$V(t) = L \cdot \frac{di(t)}{dt} + v_C(t) = L \cdot \frac{1}{R} \cdot \frac{dv_C(t)}{dt} + L \cdot C \cdot \frac{d^2v_C(t)}{dt^2} + v_C(t)$$

\*Sử dụng biến đổi Laplace, xác định hàm truyền  $F(s) = \frac{v_C(t)}{V_S}$  giữa biến đổi Laplace của điện áp ở tụ điện  $V_C(t)$  và điện áp lối vào  $V_S$

Ta có: 
$$V(s) = L \cdot \frac{1}{R} \cdot s \cdot V_C(s) + L \cdot C \cdot s^2 \cdot V_C(s) + V_C(s) = V_C(s) \cdot \left( \frac{L}{R} \cdot s + L \cdot C \cdot s^2 + 1 \right)$$

→ Hàm truyền 
$$F(s) = \frac{v_C(t)}{V_S} = \frac{1}{\frac{L}{R} \cdot s + L \cdot C \cdot s^2 + 1}$$

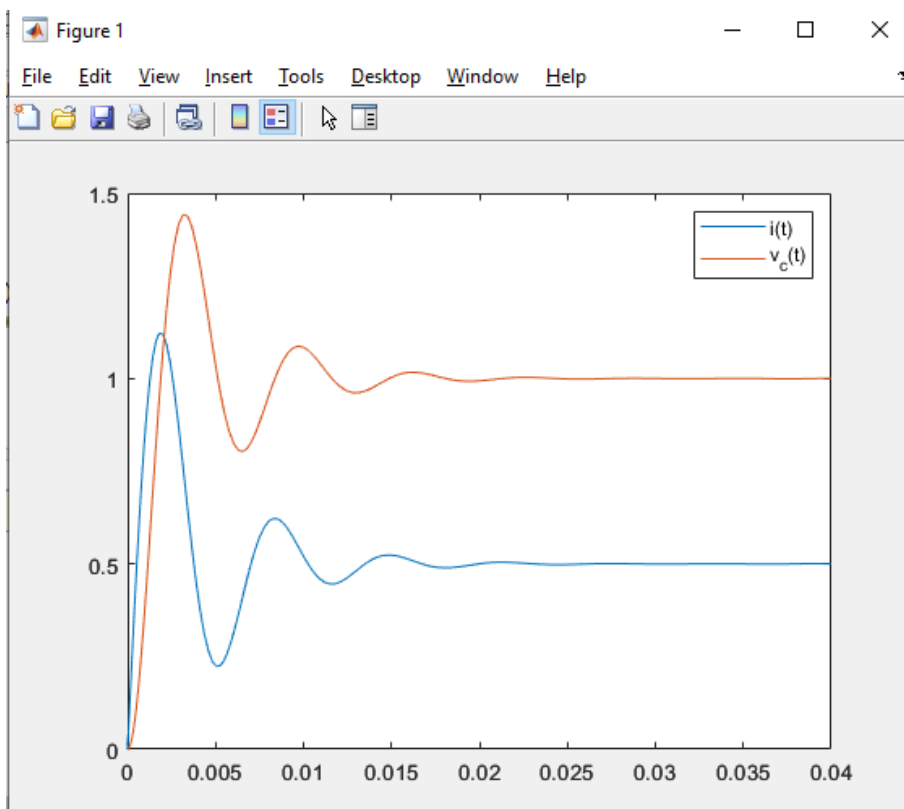
## 2. Lab

### 2.1. Mô phỏng sử dụng source-code

- Sử dụng lệnh ode45, dựa trên một phương pháp xấp xỉ Runge-Kutta:

```
1. function xdot = RLCss(t,x)
2. R=2;
3. L=10*10^-4;
4. C=10*10^-4;
5. u=1;
6. xdot(1) = 0*x(1)+(-1/L)*(x(2))+(1/L)*u;
7. xdot(2)=(1/C)*x(1) +(-1/R/C)*x(2);
8. xdot = xdot';
9. %% run in command window
10. [t,x] = ode45("RLCss", [0 4*10^-2], [0, 0];
11. plot(t,x);
12. legend("i(t)", "v_C(t)")
```

Sau khi chạy đoạn code trên, đồ thị biểu thị sự thay đổi của  $i(t)$  và  $V_C(t)$ :



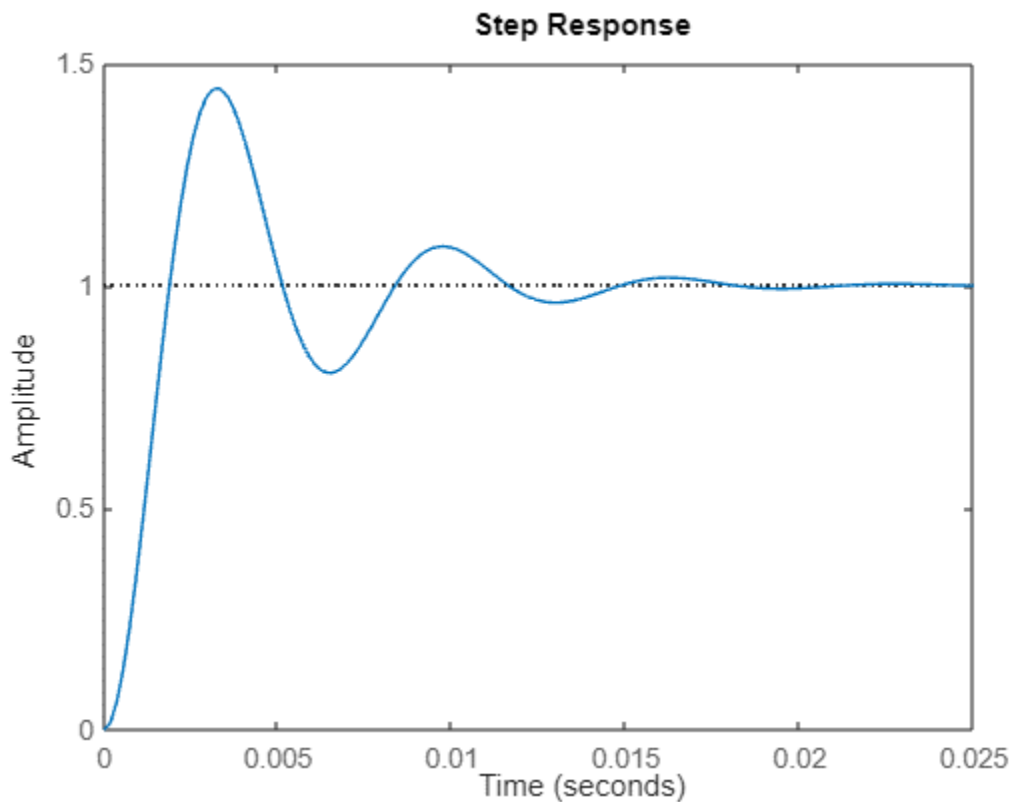
## 2.2. Mô phỏng sử dụng mô hình hàm truyền

Để biểu diễn hàm truyền của hệ thống liên tục với thời gian trong MATLAB, ta sử dụng lệnh  $tf(num, den)$  trong đó  $num$  và  $den$  lần lượt là vector hệ số của tử số và mẫu số của hàm truyền:  $num = [b_M, b_{M-1}, \dots, b_0]$ ;  $den = [a_N, a_{N-1}, \dots, a_0]$ .

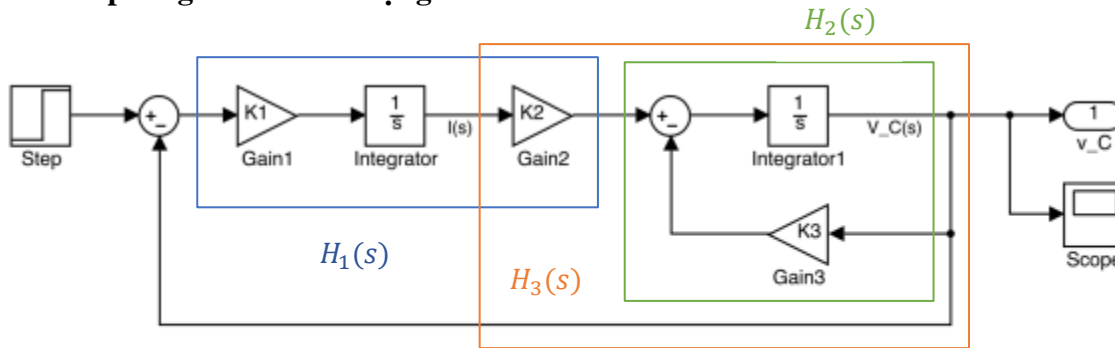
- Chương trình tính hàm truyền RLCft của hệ thống RLC dựa theo công thức hàm truyền  $F(s) = \frac{V_C(t)}{V_S} = \frac{1}{\frac{L}{R}s + LCs^2 + 1}$ , sử dụng lệnh  $RCLFT = tf(\dots, \dots)$

```
1. R=2;  
2. L=10*10^-4;  
3. C=10*10^-4;  
4. num = 1;  
5. den = [L*C, L/R, 1];  
6. RLCFT = tf(num, den);  
7. step(RLCFT);
```

- Sau khi sử dụng lệnh  $step(RLCFT)$ , ta có đáp ứng lồi ra với tín hiệu lối vào là xung nhảy bậc  $u(t)$ :



### 2.3. Mô phỏng mô hình sử dụng Simulink



Hình 2: Mô hình Simulink của mạch RLC trên

- Tính  $K_1, K_2, K_3$

Có:  $H_1(s) = \frac{K_1 K_2}{s}$ ;

Có  $H_2(s)$  ở dạng ghép nối phản hồi âm  $\rightarrow H_2(s) = \frac{\frac{1}{s}}{1 + K_3 \frac{1}{s}} = \frac{1}{s + K_3}$

$\rightarrow H_{total}(s) = \frac{H_1(s).H_2(s)}{1 + H_1(s).H_2(s)} = \frac{A}{B}$

Có:  $A = H_1(s).H_2(s) = \frac{K_1 K_2}{s} \cdot \frac{1}{s + K_3} = \frac{K_1 K_2}{s^2 + K_3 s}$

Có:  $B = 1 + H_1(s).H_2(s) = 1 + \frac{K_1 K_2}{s^2 + K_3 s}$

$\rightarrow H_{total}(s) = \frac{\frac{K_1 K_2}{s^2 + K_3 s}}{1 + \frac{K_1 K_2}{s^2 + K_3 s}} = \frac{1}{\frac{s^2 + K_3 s}{K_1 K_2} + 1} = \frac{K_1 K_2}{s^2 + K_3 s + K_1 K_2}$

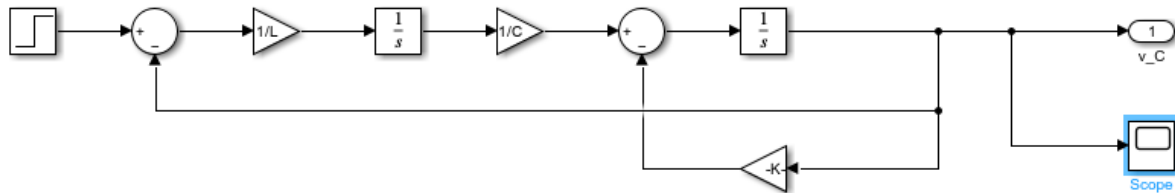
Mà lại có dạng hàm truyền  $F(s) = \frac{V_C(t)}{V_S} = \frac{1}{\frac{L}{R} \cdot s + L \cdot C \cdot s^2 + 1} = \frac{L \cdot C}{s^2 + \frac{1}{RC} s + \frac{1}{LC}}$  ở Prelab

$\rightarrow \begin{cases} K_1 \cdot K_2 = \frac{1}{LC} \\ K_3 = \frac{1}{RC} \end{cases} \rightarrow \begin{cases} K_1 = \frac{1}{L} \\ K_2 = \frac{1}{C} \\ K_3 = \frac{1}{RC} \end{cases}$

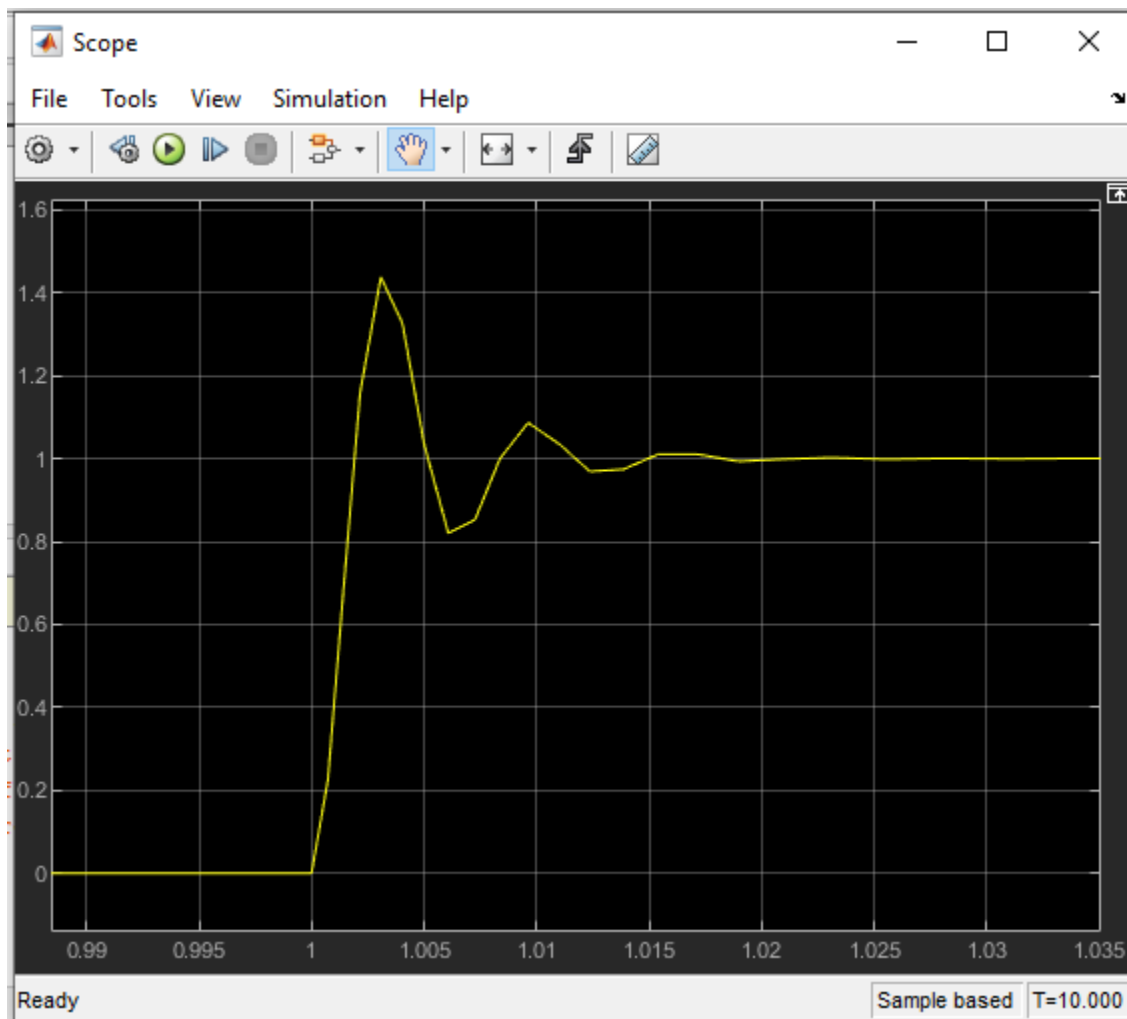
- Tại sao mô hình này không sử dụng các khối đạo hàm (differentiation blocks)?

- Sự không ổn định: Nếu đạo hàm của một hàm số phức tạp được tính toán không chính xác, kết quả của mô hình có thể không chính xác. Điều này có thể dẫn đến các vấn đề như dự đoán sai hoặc phân loại sai.
- Tốc độ tính toán: Tính đạo hàm của một hàm số phức tạp có thể tốn nhiều thời gian và tài nguyên tính toán. Điều này có thể làm chậm quá trình đào tạo và triển khai mô hình.
- Sự phức tạp: Các khối đạo hàm có thể làm tăng thêm sự phức tạp của mô hình, khiến cho mô hình khó hiểu và khó bảo trì. Điều này có thể làm giảm khả năng bảo trì và phát triển mô hình trong tương lai.

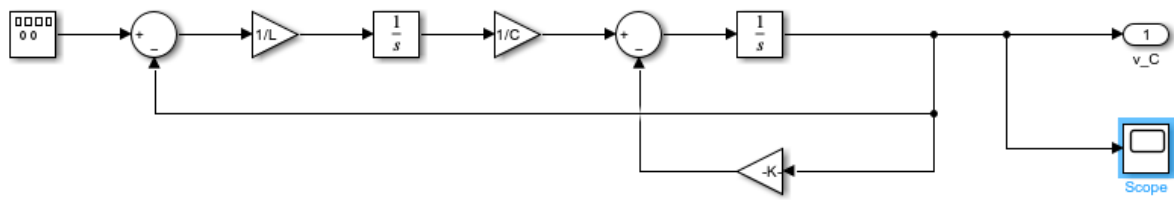
- Nhiều: Sử dụng khối đạo hàm có thể dẫn đến tăng nhiễu trong tín hiệu đầu ra, dẫn đến sự không ổn định và kết quả không chính xác.
- Giá trị khởi tạo của bộ integrator blocks in Simulink là 0. Thay đổi giá trị này tương ứng với giá trị điện áp của tụ điện  $v_c = 1$ . Mô phỏng hệ thống và vẽ lỗi ra
- Mô phỏng hệ thống



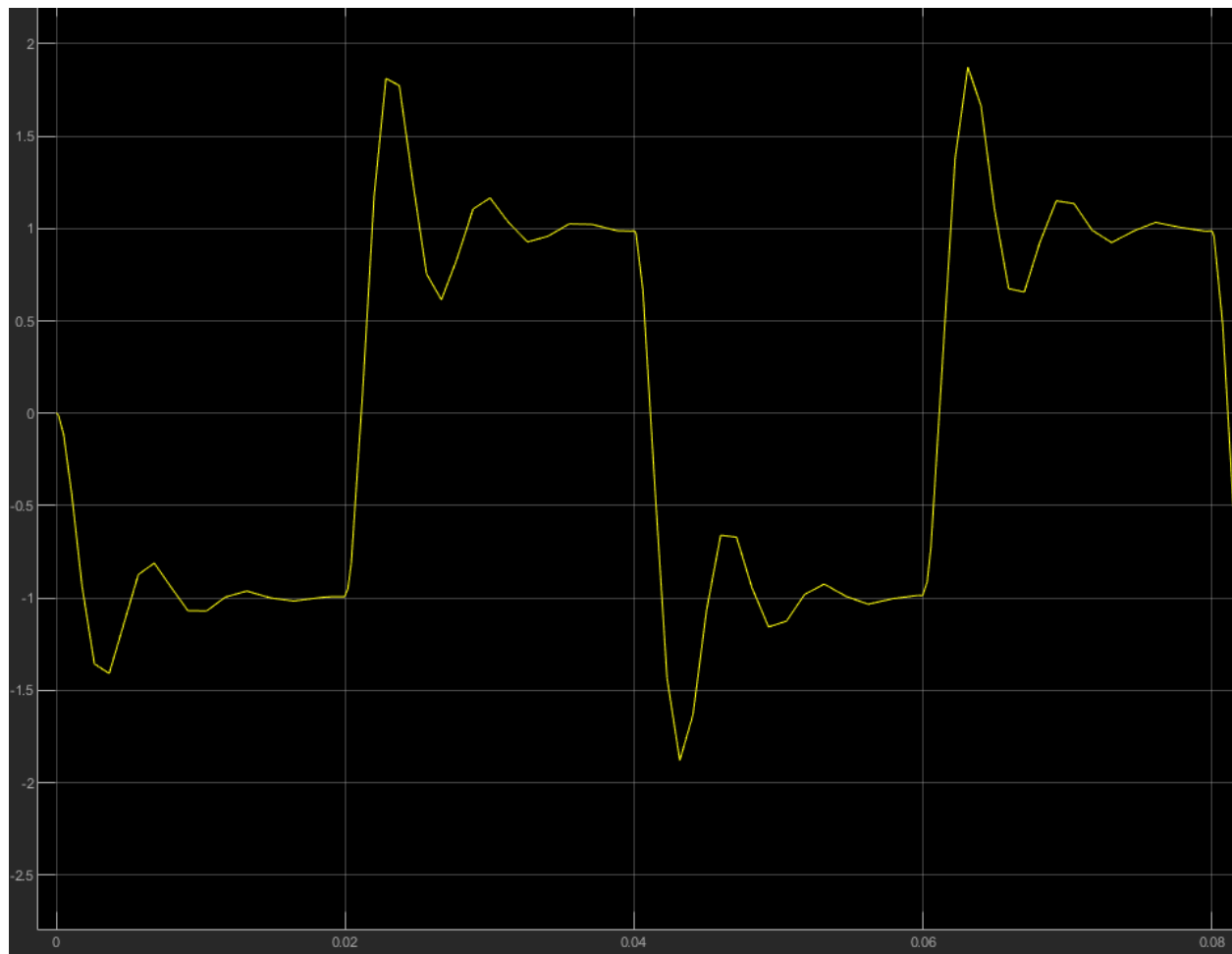
- Vẽ lỗi ra



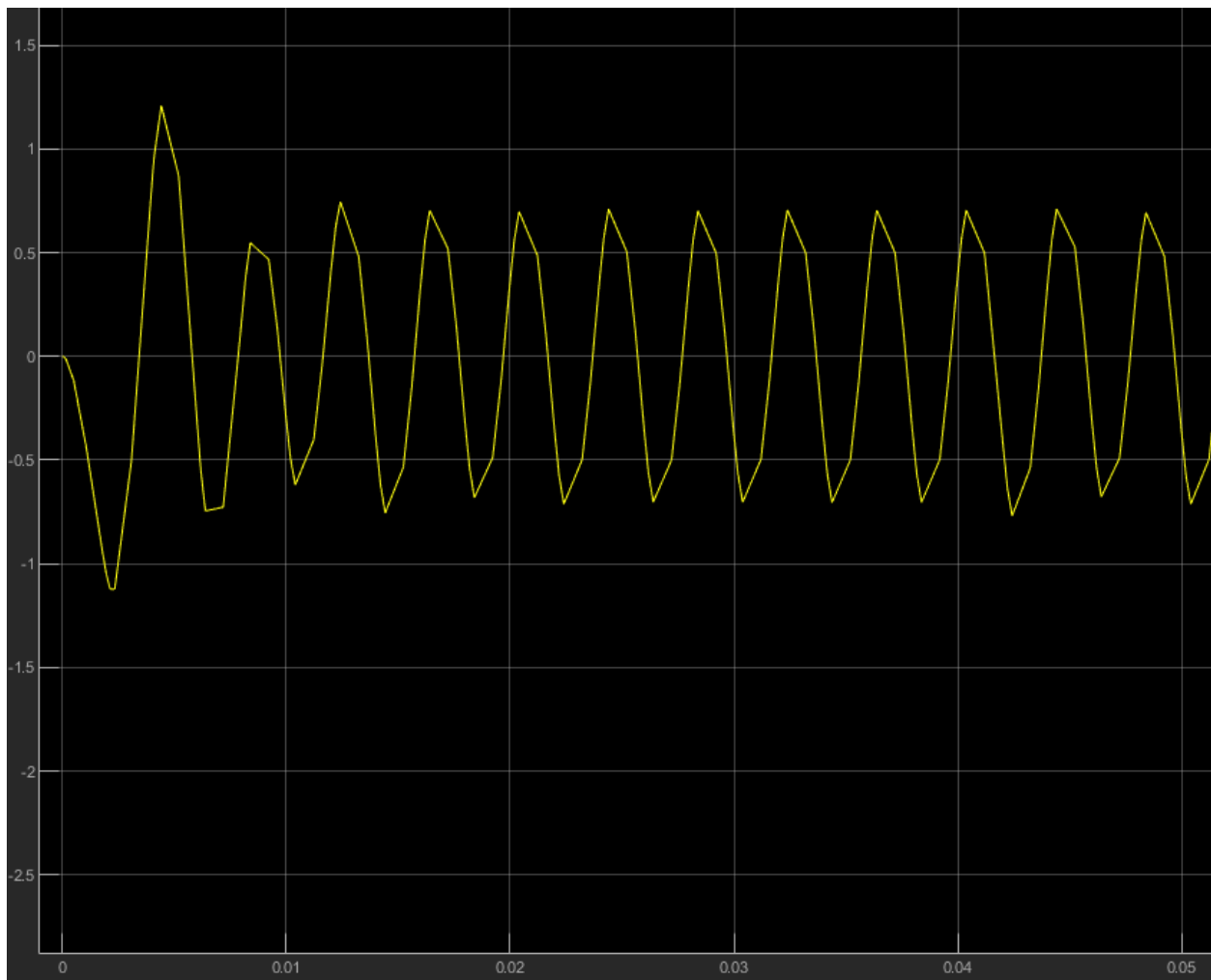
- Thay thế khối step-input trong sơ đồ trên bởi khối tạo xung vuông có biên độ 1V và tại lần lượt có các tần số  $f_1 = 25 \text{ Hz}$  và  $f_2 = 250 \text{ Hz}$ .



- $f_1 = 25 \text{ Hz}$



- $f_2 = 250 \text{ Hz}$



Có sự khác nhau về đầu ra của hai tín hiệu khi thay đổi tần số 25Hz và 250Hz. Sự khác nhau này do tần số của sóng lỗi vào ảnh hưởng đến hình dáng của tín hiệu lỗi ra. Cụ thể, khi tần số là 25Hz, tín hiệu lỗi ra có dạng gần giống sóng sine vì điện áp trên tụ có đủ thời gian để tích tụ và phóng điện; còn khi tần số là 250Hz, tín hiệu lỗi ra có dạng răng cưa vì điện áp trên tụ không đủ thời gian để kịp tích tụ.

### 3. Summary

#### 3.1. Các cách biểu diễn trên có thể sử dụng cho các hệ thống phi tuyến không?

Simulink có thể sử dụng cho các hệ thống phi tuyến vì nó cung cấp các khối hàm phi tuyến, khối bộ điều khiển phi tuyến. Tuy nhiên, các cách biểu diễn này có thể không chính xác nếu mô hình phi tuyến của hệ thống không được mô tả chính xác, và có thể không hiệu quả nếu mô hình phi tuyến của hệ thống phức tạp. Ngoài ra, việc mô hình và mô phỏng hệ thống phi tuyến trong Simulink có thể phức tạp hơn so với hệ thống tuyến tính.



### 3.2. Mô hình nào phù hợp cho các hệ thống có trạng thái ban đầu khác nhau?

- Mô hình sử dụng phương trình vi phân hoặc sử dụng hàm ode trong command window cho phép ta khởi tạo giá trị ban đầu của hệ thống.
- Mô hình sử dụng hàm truyền chỉ phụ thuộc vào công thức hàm truyền, không thể thiết lập giá trị khởi tạo cho hệ thống.
- Mô phỏng sử dụng Simulink cho phép ta thiết lập các giá trị khởi tạo của các block.

### 3.3. Ưu điểm và nhược điểm của các cách biểu diễn trên

- **Mô hình sử dụng phương trình vi phân hoặc sử dụng hàm ode:**
  - Ưu điểm: Xây dựng mối liên hệ giữa các biến trực quan, thiết lập khởi tạo giá trị ban đầu.
  - Nhược điểm: Dễ thiết lập sai phương trình liên hệ vì có thể phương trình vi phân sẽ rất phức tạp.
- **Mô hình sử dụng hàm truyền chỉ phụ thuộc vào công thức hàm truyền:**
  - Ưu điểm: sử dụng hàm tf đơn giản với các thông số đặc tả  $K_1, K_2, K_3, \dots$  của hàm truyền đã được tính toán trước.
  - Nhược điểm: cần tính toán trước các thông số đặc tả  $K_1, K_2, K_3, \dots \rightarrow$  cần giải hệ phương trình toán học.
- **Mô hình mô phỏng sử dụng Simulink:**
  - Ưu điểm: Dùng các ký hiệu, khối hàm mà MATLAB đã trang bị sẵn, thiết lập mô hình một cách trực quan, dễ dàng điều chỉnh mô hình và giá trị khởi tạo.
  - Nhược điểm: Xây dựng, chọn các khối hàm phải đúng thì mới ra kết quả đúng được.