ECE355L Project 2

Chase A. Lotito, *SIUC Undergraduate*

1 Exercise 1

We are given the following differential equation with initial conditions, $y(0) = 1$, $y'(0) = 1$, and $y''(0) = 0$:

$$y''' + 8y'' + 2521y' + 5018y = f'' + 5018f \tag{1}$$

We can solve for a symbolic expression of y using the *dsolve* method which is apart of the Symbolic Math Toolbox extension in MATLAB. This requires us to create a symbolic character variable for $y(t)$, then using the *diff* method to differentiate our signals as necessary.

```
% Chase Lotito - SIUC - Spring 2024
% ECE355L - Project 2
% Q1: dsolve to solve for zero-input response
% y0 = dsolve('D3y + 8*D2y + 2521*Dy + 5018*y = 0', 'y(0)=1', 'Dy(0)=1', 'D2y(0)=0');

% system variable
syms y(t)           % make y a function of t

% initial conditions
Dy = diff(y,t);          % define D operator
D2y = diff(y,t,2);       % define D2 operator
cond = [y(0) == 1, Dy(0) == 1, D2y(0) == 0]; % set initial conditions

y0 = diff(y, t, 3) + 8 * diff(y, t, 2) + 2521 * diff(y, t) + 5018 * y == 0;
S = dsolve(y0, cond); % solve diff eq. with ICs

% get latex of it
chr = latex(S);
```

In the "Command Window":

```
S =

(exp(-3*t)*(7489*sin(50*t) - 700*cos(50*t) + 125750*exp(t)))/125050
```

The code also provides us a symbolic LaTeX equivalent for $y_0(t)$ in *chr*:

$$y_0(t) = \frac{\mathrm{e}^{-3\,t}\,\left(7489\,\sin\left(50\,t\right) - 700\,\cos\left(50\,t\right) + 125750\,\mathrm{e}^{t}\right)}{125050} \tag{2}$$

2 Exercise 2

We can then plot the zero-input response for the system in the previous exercise via this additional code. Here *fplot* is a function in provided by Symbolic Math Toolbox that plots symbolic expressions:

```
% ... Exercise 1 Code Above...
fplot(S, [0,4]);    % fplot(<function>, [<xmin>, <xmax>])
xlabel('t');
ylabel('y0(t)');
title('Zero-Input Response');
```

Remember that from Exercise 1, S was set equal to the symbolic expression for $y_0(t)$. Then we get the following plot:

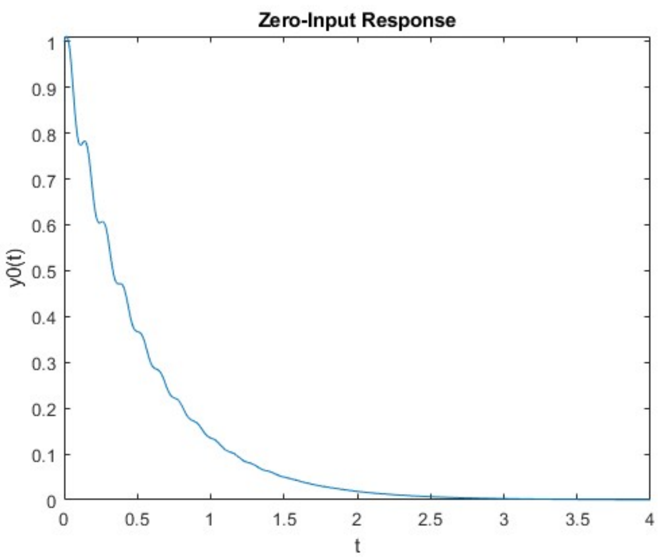

Figure 1: Zero-Input response of system

3 Exercise 3

Create a system object using tf for the system from part 1 and obtain the zero-state impulse and step response using impulse and step function.

Solution.

Again, here is the differential equation that describes our system (w.r.t. time):

$$y''' + 8y'' + 2521y' + 5018y = f'' + 5018f \tag{3}$$

To get to the transfer function, we need to take the Laplace transform, omitting any initial conditions:

$$s^3Y(s) + 8s^2Y(s) + 2521sY(s) + 5018Y(s) = s^2F(s) + 5018F(s) \tag{4}$$

This we can rearrange to get the transfer function $H(s)$:

$$H(s) = \frac{Y(s)}{F(s)} = \frac{s^2 + 5018}{s^3 + 8s^2 + 2521s + 5018} \tag{5}$$

Using the Control Systems Toolbox extension in MATLAB, we can take the numerator and denominator of $H(s)$ and solve for the step and impulse responses.

```
% Chase Lotito - 355L Project 2 - Exercise 3
% Define the numerator and denominator coefficients
num = [1 0 5018];
den = [1 8 2521 5018];
% Create the transfer function object
TFsys = tf(num, den);
% Remove the roots from the transfer function
TFsys_no_roots = tzero(TFsys);
% Define the time vector
t_vec = 0:0.01:10; % Time vector from 0 to 10 with a step size of 0.01
% Calculate the step response
[ystep, t_step] = step(TFsys_no_roots, t_vec);
% Calculate the impulse response
[h, t_impulse] = impulse(TFsys_no_roots, t_vec);
% Plot the step response
subplot(2, 1, 1);
plot(t_vec(1:length(ystep)), ystep); % Adjust the length of t_vec
title('Step Response');
xlabel('t');
ylabel('y_{step}(t)');
% Plot the impulse response
subplot(2, 1, 2);
```

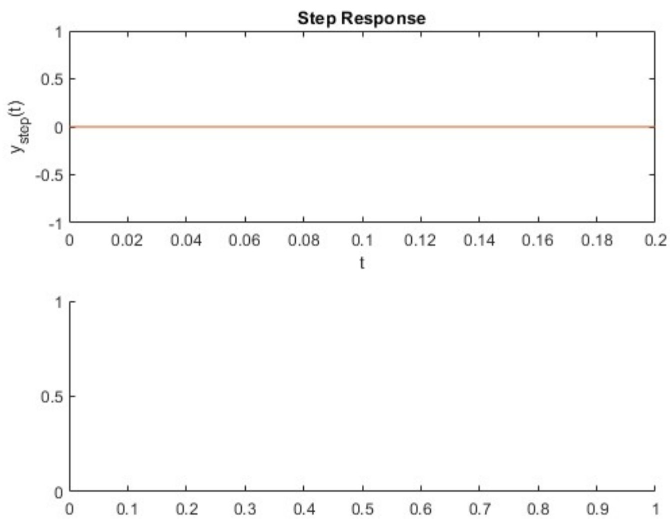

Figure 2: Step-Response and Impulse-Response