Assignment 1

EET305- Signals and Systems

B.Tech Fifth Semester
Department of Electrical and Electronics Engineering
Government Engineering College, Barton Hill, Thiruvananthapuram

Questions

Q1: A trapezoidal signal is described as follows

$$x(t) = \begin{cases} t, & 0 \le t < 2 \\ 2, & 2 \le t \le 6 \\ 8-t, & 6 < t \le 8 \\ 0, & \text{otherwise} \end{cases}$$

Simulate the following and generate the corresponding plot.

(a) $x(t)$

(b) $x(t-3)$

(c) $x(2t)$

(d) $x(\frac{1}{2}t)$

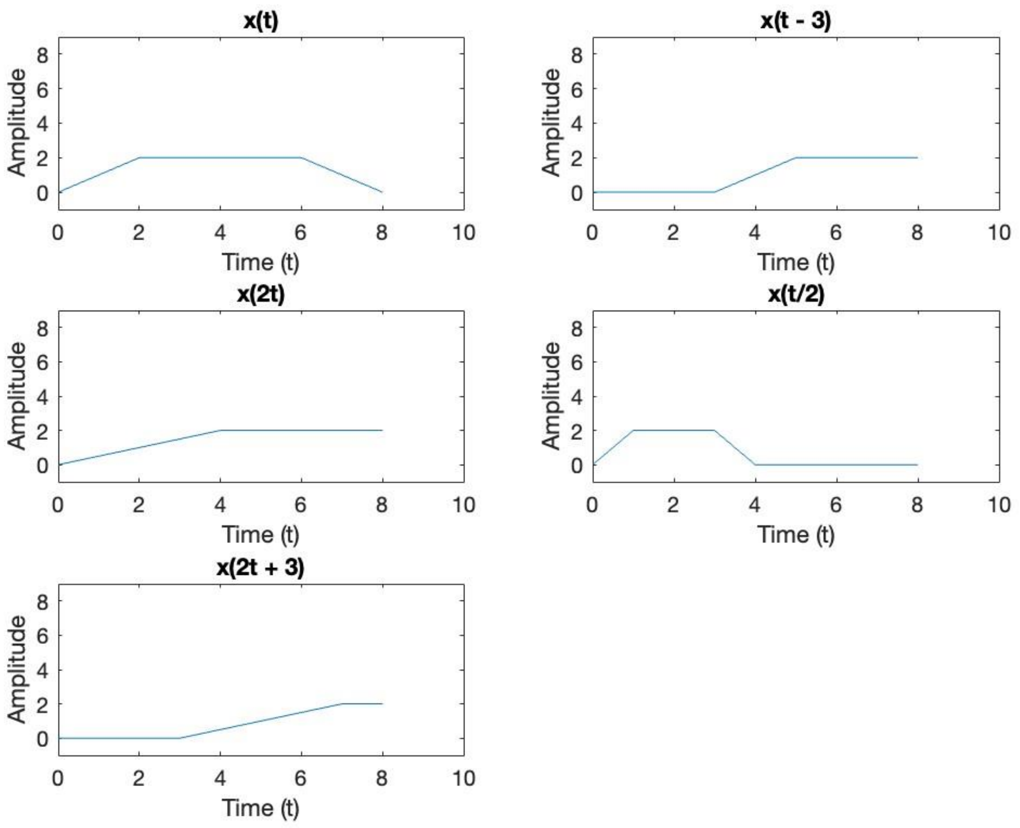
(e) $x(2t+3)$

```
% Define time vector t with a fine step for smooth plots

T = 0:0.01:8;



% Define the original x(t) as a piecewise function

X = (t>=0 & t<2).*t + (t>=2 & t<=6).*2 + (t>6 & t<=8).*(8-t);



% Set axis limits

X_limits = [0 10]; % Adjust based on the range of t for
transformations
```

```
Y_limits = [-1 9]; % Adjust based on the amplitude of x(t)


% Plot the original x(t)

Subplot(3,2,1);

Plot(t, x);

Title('x(t)');

Xlabel('Time (t)');

Ylabel('Amplitude');

Axis([x_limits y_limits]); % Apply axis limits to view entire graph


% Transformation 1: Time Shift x(t – 3)

T_shifted = t – 3; % Shift time vector by +3 units to the left

X_shifted = (t_shifted>=0 & t_shifted<2).*t_shifted + (t_shifted>=2 &
t_shifted<=6).*2 + (t_shifted>6 & t_shifted<=8).*(8 – t_shifted);

Subplot(3,2,2);

Plot(t, x_shifted);

Title('x(t – 3)');

Xlabel('Time (t)');

Ylabel('Amplitude');

Axis([x_limits y_limits]);


% Transformation 2: Time Scaling x(2t)
```

```
T_scaled = t / 2; % Scale down the time vector by ½ for compression

X_scaled = (t_scaled>=0 & t_scaled<2).*t_scaled + (t_scaled>=2 & t_scaled<=6).*2 + (t_scaled>6 & t_scaled<=8).*(8 - t_scaled);

Subplot(3,2,3);

Plot(t, x_scaled);

Title('x(2t)');

Xlabel('Time (t)');

Ylabel('Amplitude');

Axis([x_limits y_limits]);


% Transformation 3: Time Scaling x(t/2)

T_expanded = t * 2; % Scale up the time vector by 2 for expansion

X_expanded = (t_expanded>=0 & t_expanded<2).*t_expanded + (t_expanded>=2 & t_expanded<=6).*2 + (t_expanded>6 & t_expanded<=8).*(8 - t_expanded);

Subplot(3,2,4);

Plot(t, x_expanded);

Title('x(t/2)');

Xlabel('Time (t)');

Ylabel('Amplitude');

Axis([x_limits y_limits]);
```

```
% Transformation 4: Combined Scaling and Shifting x(2t + 3)

T_combined = (t - 3) / 2; % Scale down and shift time vector

X_combined = (t_combined>=0 & t_combined<2).*t_combined + (t_combined>=2 & t_combined<=6).*2 + (t_combined>6 & t_combined<=8).*(8 - t_combined);

Subplot(3,2,5);

Plot(t, x_combined);

Title('x(2t + 3)');

Xlabel('Time (t)');

Ylabel('Amplitude');

Axis([x_limits y_limits])
```



•

Q2: Given the following input signal $x(t)$ and impulse response $h(t)$:

$$x(t) = \begin{cases} 1 & 0 \le t < 3 \\ 0 & \text{otherwise} \end{cases} \quad h(t)$$

$$= e^{-t}, t \ge 0$$

(a) Plot the input signal $x(t)$ and the impulse response $h(t)$.

(b) Perform the convolution of $x(t)$ and $h(t)$ using MATLAB.

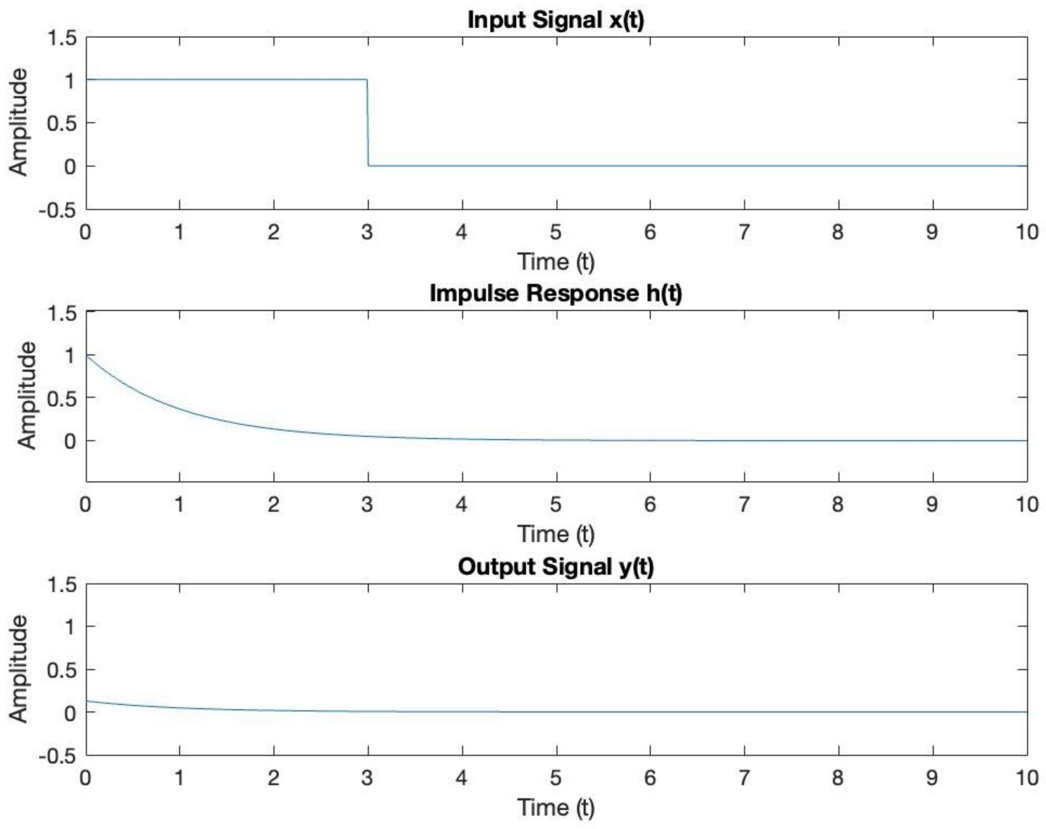
(c) Plot the output signal $y(t)$ obtained after convolution.

(d) Analyze the system's behavior based on the convolution result.

```
Answer:% Define time vector

T = 0:0.01:10;


% Define x(t) and h(t)

X = (t>=0 & t<3);

H = exp(-t).*(t>=0);


% Perform convolution

Y = conv(x, h, 'same') * 0.01;


% Set axis limits

X_limits = [0 10];
```

```
Y_limits = [-0.5 1.5]; % Adjust based on amplitude of y(t)

% Plot results
Subplot(3,1,1); plot(t, x); title('Input Signal x(t)');
Xlabel('Time (t)'); ylabel('Amplitude');
Axis([x_limits y_limits]);

Subplot(3,1,2); plot(t, h); title('Impulse Response h(t)');
Xlabel('Time (t)'); ylabel('Amplitude');
Axis([x_limits y_limits]);

Subplot(3,1,3); plot(t, y); title('Output Signal y(t)');
Xlabel('Time (t)'); ylabel('Amplitude');
```

```
Axis([x_limits   y_limits]);
```



Q3: A system with an impulse response $h(t) = e^{-2t}$ for $t \geq 0$ is excited by a square wave input:

$$x(t) = 1 \quad \text{for} \quad 0 \leq t < 5, \qquad x(t) = 0 \quad \text{otherwise}$$

(a) Define the input square wave in MATLAB.

(b) Perform the convolution of $x(t)$ with $h(t)$ to find the output $y(t)$.

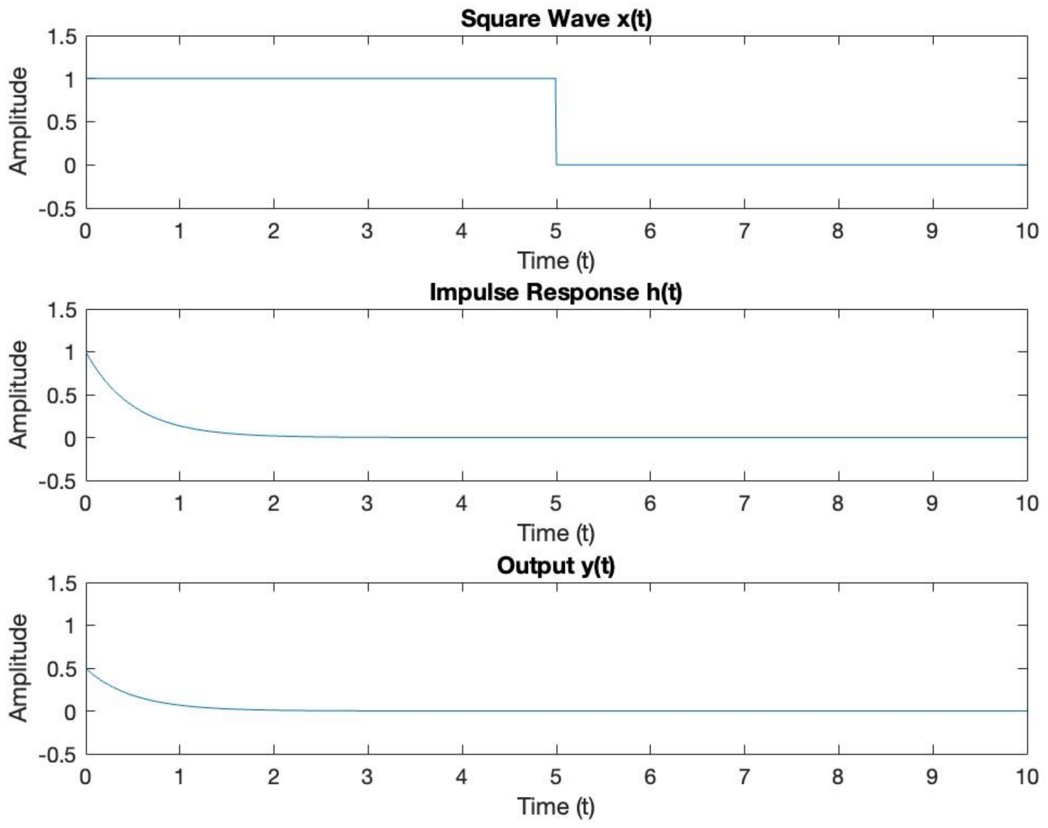
(c) Plot the input, impulse response, and output signals.

(d) Discuss the system's response to the square wave.

```
% Define time vector

T = 0:0.01:10;
```

```
% Define x(t) as a square wave and h(t) as the impulse
response

X = (t>=0 & t<5);

H = exp(-2*t).*(t>=0);


% Perform convolution

Y = conv(x, h, 'same') * 0.01;


% Set axis limits

X_limits = [0 10];

Y_limits = [-0.5 1.5]; % Adjust based on amplitude of y(t)


% Plot the signals

Subplot(3,1,1); plot(t, x); title('Square Wave x(t)');

Xlabel('Time (t)'); ylabel('Amplitude');

Axis([x_limits y_limits]);


Subplot(3,1,2); plot(t, h); title('Impulse Response h(t)');

Xlabel('Time (t)'); ylabel('Amplitude');

Axis([x_limits y_limits]);


Subplot(3,1,3); plot(t, y); title('Output y(t)');
```

```
Xlabel('Time (t)'); ylabel('Amplitude');

Axis([x_limits y_limits]);
```



Q4: Consider a system with an impulse response $h(t) = e^{-t}$ representing a low-pass filter. The input signal is a sum of two sinusoids:

$$x(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t), \qquad f_1 = 1\text{Hz}, \qquad f_2 = 10\text{Hz}$$

(a) Plot the input signal $x(t)$ for $0 \le t \le 10$ seconds.

(b) Perform the convolution of $x(t)$ with $h(t)$ in MATLAB.

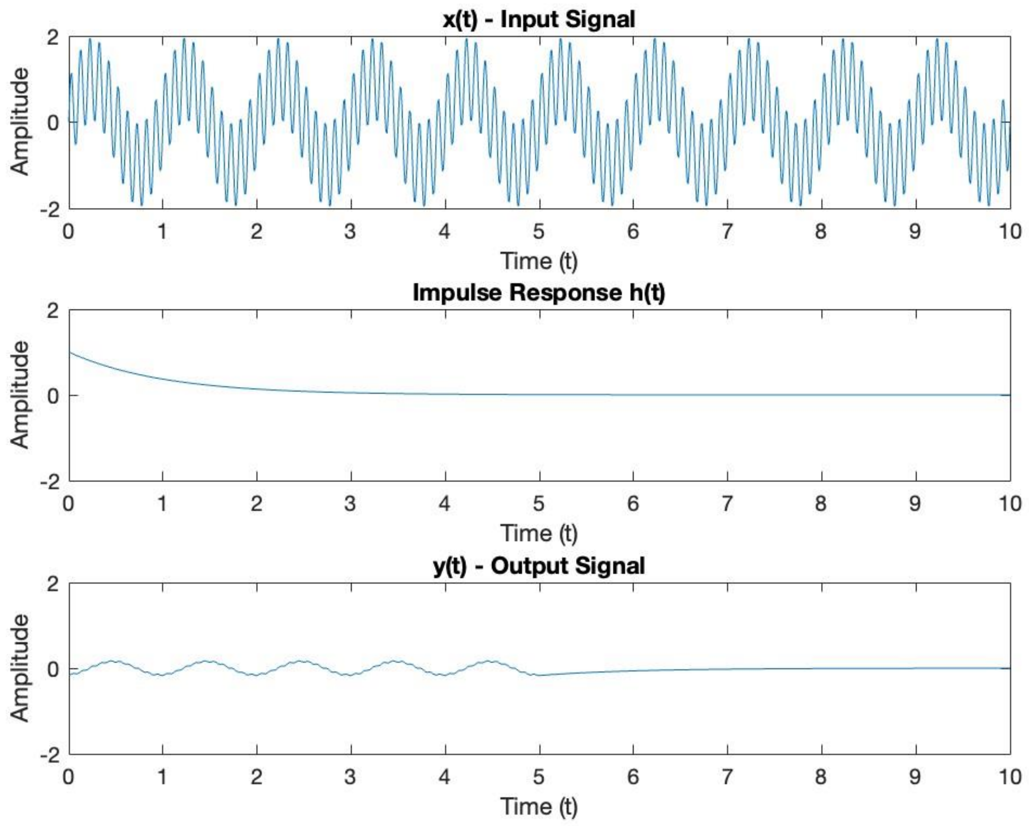
(c) Plot the output signal $y(t)$.

(d) Discuss the effect of the system on the two sinusoidal components (low-passfiltering behavior).

```
% Define parameters
```

```
F1 = 1; f2 = 10;
T = 0:0.01:10;

% Define x(t) and h(t)
X = sin(2*pi*f1*t) + sin(2*pi*f2*t);
H = exp(-t).*(t>=0);

% Perform convolution
Y = conv(x, h, ‘same’) * 0.01;

% Set axis limits
X_limits = [0 10];
Y_limits = [-2 2]; % Adjust based on the amplitude of y(t)

% Plot results
Subplot(3,1,1); plot(t, x); title(‘x(t) – Input Signal’);
Xlabel(‘Time (t)’); ylabel(‘Amplitude’);
Axis([x_limits y_limits]);

Subplot(3,1,2); plot(t, h); title(‘Impulse Response h(t)’);
Xlabel(‘Time (t)’); ylabel(‘Amplitude’);
Axis([x_limits y_limits]);
```

```
Subplot(3,1,3); plot(t, y); title('y(t) - Output Signal');

Xlabel('Time (t)'); ylabel('Amplitude');

Axis([x_limits y_limits]);
```



Q5: Given two arbitrary continuous-time signals:

$$x(t) = \sin(2\pi t), \quad 0 \le t \le 2, \quad h(t) = t, \quad 0 \le t \le 1$$

(a) Write MATLAB code to define $x(t)$ and $h(t)$ as functions.

(b) Compute the convolution $y(t) = x(t) * h(t)$ using MATLAB's conv function.

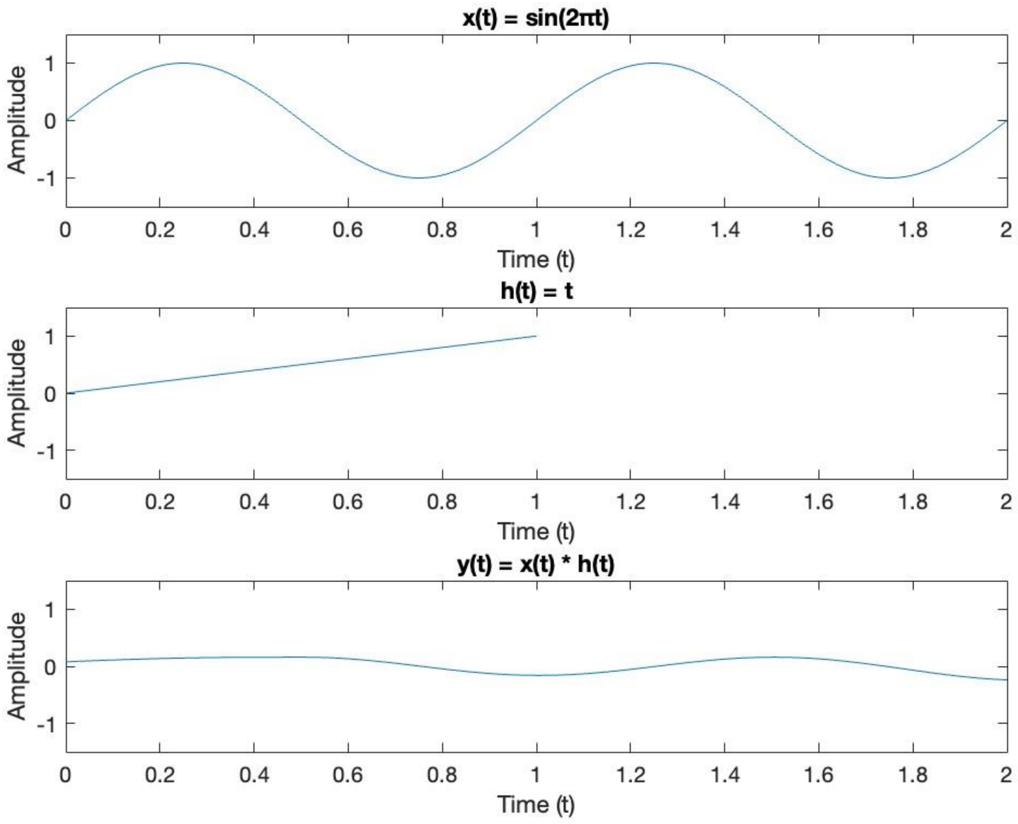
(c) Plot the original signals $x(t)$ and $h(t)$, as well as the output $y(t)$.

(d) Interpret the physical meaning of the convolution in this case.

```
% Define time vectors

T1 = 0:0.01:2;
```

```
T2 = 0:0.01:1;


% Define x(t) and h(t)

X = sin(2*pi*t1);

H = t2;


% Perform convolution

Y = conv(x, h, ‘same’) * 0.01;


% Set axis limits

X_limits = [0 2];

Y_limits = [-1.5 1.5];   % Adjust y-limits to include negative values


% Plot each signal and the convolution result

Subplot(3,1,1); plot(t1, x); title(‘x(t) = sin(2πt)’);

Xlabel(‘Time (t)’); ylabel(‘Amplitude’);

Axis([x_limits y_limits]);


Subplot(3,1,2); plot(t2, h); title(‘h(t) = t’);

Xlabel(‘Time (t)’); ylabel(‘Amplitude’);

Axis([x_limits y_limits]);
```

```
Subplot(3,1,3);  plot(t1,  y(1:length(t1)));  title('y(t)  =  x(t)  *  h(t)');

Xlabel('Time  (t)');  ylabel('Amplitude');

Axis([x_limits  y_limits]);
```



Q6: Consider a system with the following impulse response $h(t)$:

$$h_1(t) = e^{-t} \quad \text{for} \quad t \geq 0$$

$$h_2(t) = e^{-2t} \quad \text{for} \quad t \geq 0$$

The input signal is $x(t) = \sin(2\pi t)$ for $0 \leq t \leq 5$.

(a) Compute the convolution of $x(t)$ with both impulse responses $h_1(t)$ and $h_2(t)$ in MATLAB.

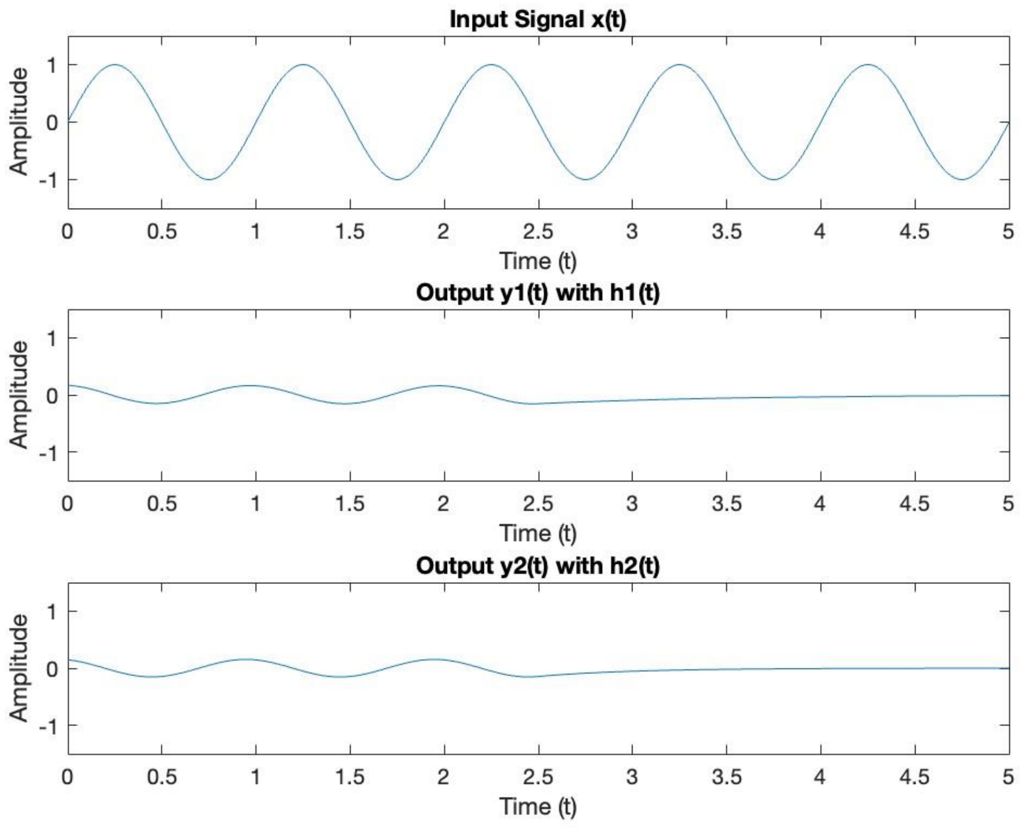
(b) Plot the output signals for both cases.

(c) Compare and contrast the outputs based on the different impulse responses.Discuss how the change in the impulse response affects the output.

```
% Define time vector and input signal x(t)
```

```
T = 0:0.01:5;

X = sin(2*pi*t);


% Define impulse responses h1(t) and h2(t)

H1 = exp(-t).*(t>=0);

H2 = exp(-2*t).*(t>=0);


% Perform convolution for both cases

Y1 = conv(x, h1, ‘same’) * 0.01;

Y2 = conv(x, h2, ‘same’) * 0.01;


% Set axis limits

X_limits = [0 5];

Y_limits = [-1.5 1.5];   % Adjust y-limits for all subplots


% Plot results

Subplot(3,1,1); plot(t, x); title(‘Input Signal x(t)’);

Xlabel(‘Time (t)’); ylabel(‘Amplitude’);

Axis([x_limits y_limits]);


Subplot(3,1,2); plot(t, y1); title(‘Output y1(t) with h1(t)’);

Xlabel(‘Time (t)’); ylabel(‘Amplitude’);

Axis([x_limits y_limits]);
```

```
Subplot(3,1,3); plot(t, y2); title('Output y2(t) with h2(t)');

Xlabel('Time (t)'); ylabel('Amplitude');

Axis([x_limits y_limits]);
```



Q7: A periodic square wave $x(t)$ with period $T = 2\pi$ is defined as:

$$x(t) = \begin{cases} 1, & 0 \le t < \pi \\ -1, & \pi \le t < 2\pi \end{cases}$$

(a) Compute the Fourier series coefficients a_n, b_n, and a_0 for the square wave.

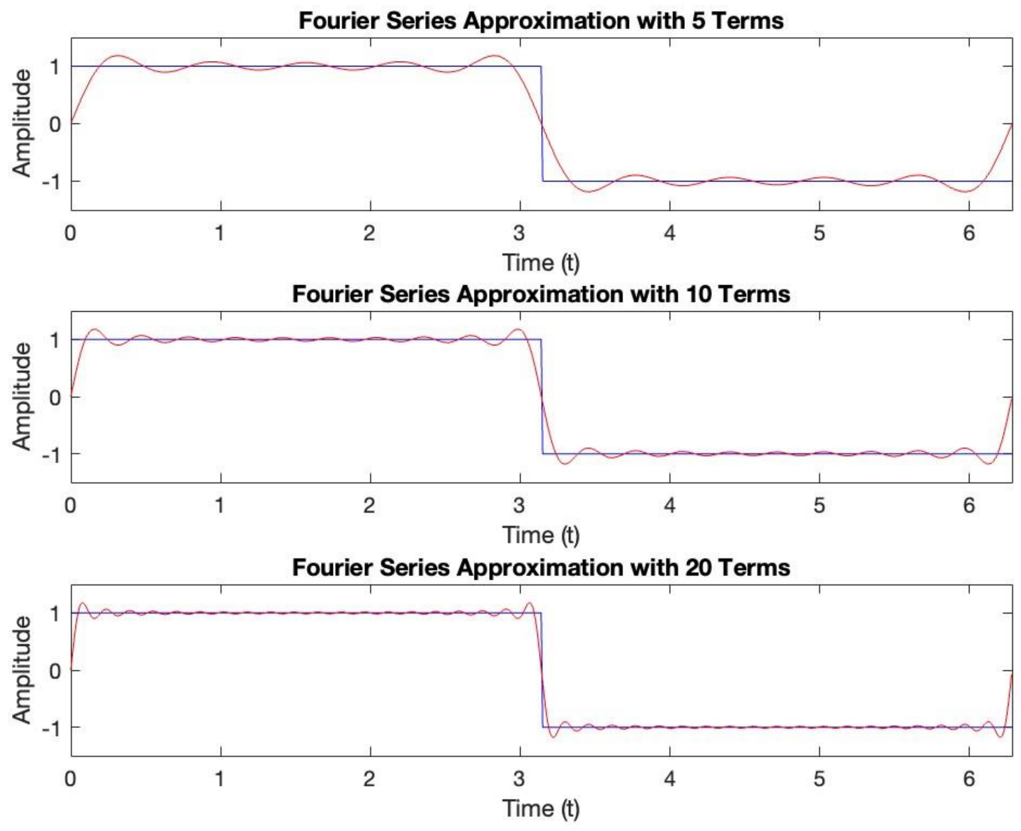
(b) Plot the square wave and its Fourier series approximation using the first 5, 10,and 20 terms of the series.

(c) Use MATLAB to compute the Fourier series and plot the approximations forthe given number of terms.

(d) Discuss the convergence of the Fourier series to the square wave as the numberof terms increases.

```
% Define time vector
```

```
T = 0:0.01:2*pi;


% Define original square wave

X = square(t);


% Fourier series approximation for 5, 10, 20 terms

N = [5, 10, 20];

X_limits = [0 2*pi];

Y_limits = [-1.5 1.5];


For I = 1:3

        Approx = 0;

        For n = 1:N(i)

                Approx = approx + (4/pi) * (sin((2*n-1)*t) / (2*n-1));

        End

        Subplot(3,1,i);

        Plot(t, x, ‘b’, t, approx, ‘r’);

        Title([‘Fourier Series Approximation with ‘, num2str(N(i)), ‘
Terms’]);

        Xlabel(‘Time (t)’);

        Ylabel(‘Amplitude’);

        Axis([x_limits y_limits]);

End
```



Q8: A periodic sawtooth wave is defined by:

$$x(t) = \frac{t}{\pi}, \quad -\pi \le t < \pi$$

This signal repeats with a period $T = 2\pi$.

(a) Derive the Fourier series coefficients for the sawtooth wave.

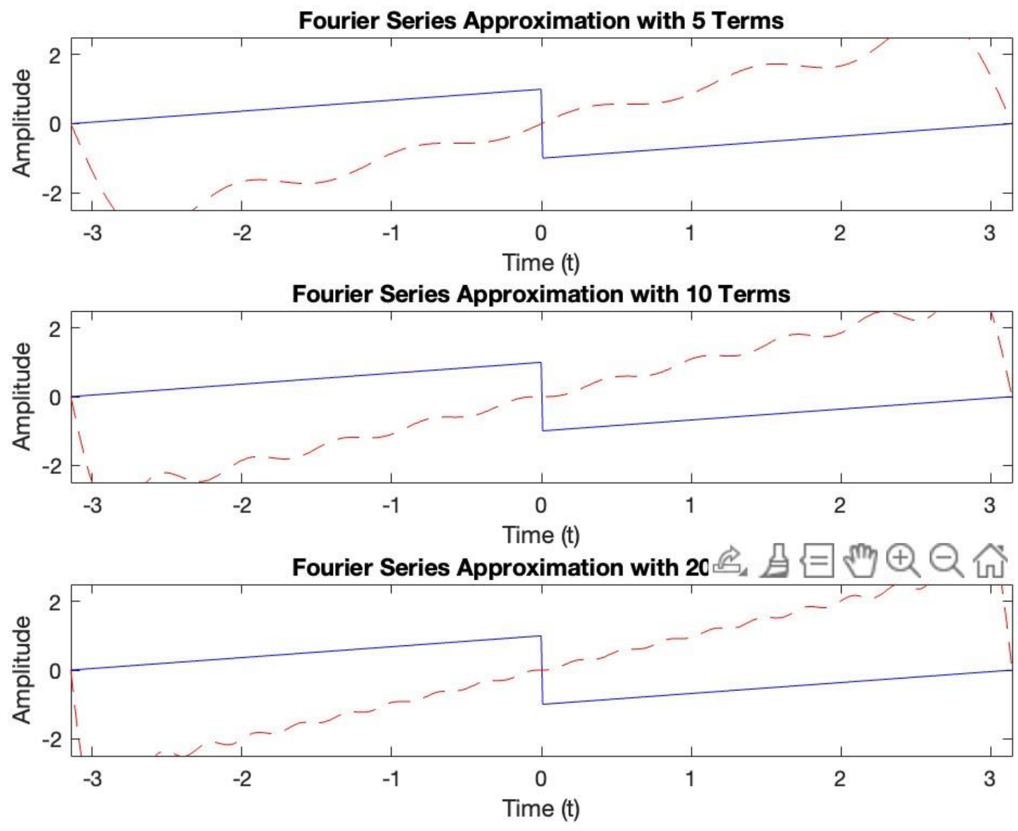
(b) Using MATLAB, plot the original sawtooth wave and its Fourier series approximations using 5, 10, and 20 terms.

(c) Comment on the accuracy of the approximation and explain why the Gibbsphenomenon occurs at discontinuities.

(d) Analyze the impact of the number of harmonics on the quality of the approximation.

```
% Define time vector and original sawtooth wave

T = -pi:0.01:pi;

X = sawtooth(t);
```

```
% Fourier series approximation for 5, 10, 20 terms

N = [5, 10, 20];

Y_limits = [-2.5 2.5];


For I = 1:3

        Approx = 0;

        For n = 1:N(i)

                Approx = approx + (2*(-1)^(n+1)/n) * sin(n*t);

        End

        Subplot(3,1,i);

        Plot(t, x, ‘b’, t, approx, ‘r–‘);

        Title([‘Fourier Series Approximation with ‘, num2str(N(i)), ‘
Terms’]);

        Xlabel(‘Time (t)’);

        Ylabel(‘Amplitude’);

        Axis([-pi pi y_limits]);
```



End

Q9: A triangular wave $x(t)$ has period $T = 2\pi$ and is defined as:

$$x(t) = \begin{cases} \frac{t}{\pi}, & 0 \leq t \leq \pi \\ -\frac{t}{\pi} + 2, & \pi \leq t \leq 2\pi \end{cases}$$

(a) Compute the Fourier series coefficients for the triangular wave.

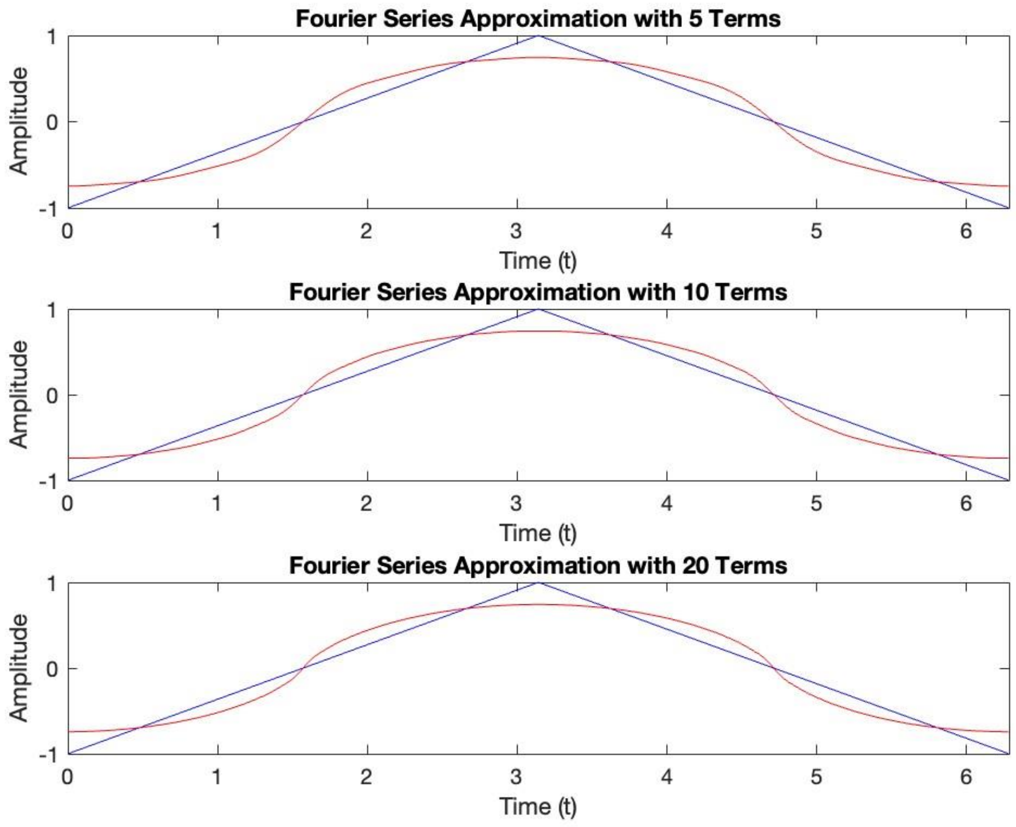
(b) Plot the triangular wave and its Fourier series approximations using MATLABwith 5, 10, and 20 terms.

(c) Discuss the symmetry properties of the triangular wave and their impact onthe Fourier coefficients.

(d) Compare the rate of convergence of the Fourier series for the triangular wavewith that of the square wave.

```
% Define time vector and original triangular wave

T = 0:0.01:2*pi;

X = sawtooth(t, 0.5);
```

```
% Fourier series approximation for 5, 10, 20 terms

N = [5, 10, 20];

Y_limits = [-1 1];


For I = 1:3

        Approx = 0;

        For n = 1:N(i)

                Approx = approx + ((-1)^(n) / (2*n-1)^2) * cos((2*n-
1)*t);

        End

        Approx = approx * (8/pi^2);

        Subplot(3,1,i);

        Plot(t, x, 'b', t, approx, 'r');

        Title(['Fourier Series Approximation with ', num2str(N(i)), '
Terms']);

        Xlabel('Time (t)');

        Ylabel('Amplitude');

        Axis([0 2*pi y_limits]);

End
```



Q10: A half-wave rectified sine wave is defined by:

$$x(t) = \begin{cases} \sin(t), & 0 \le t \le \pi \\ 0, & \pi < t \le 2\pi \end{cases}$$

(a) Derive the Fourier series coefficients for the half-wave rectified sine wave.

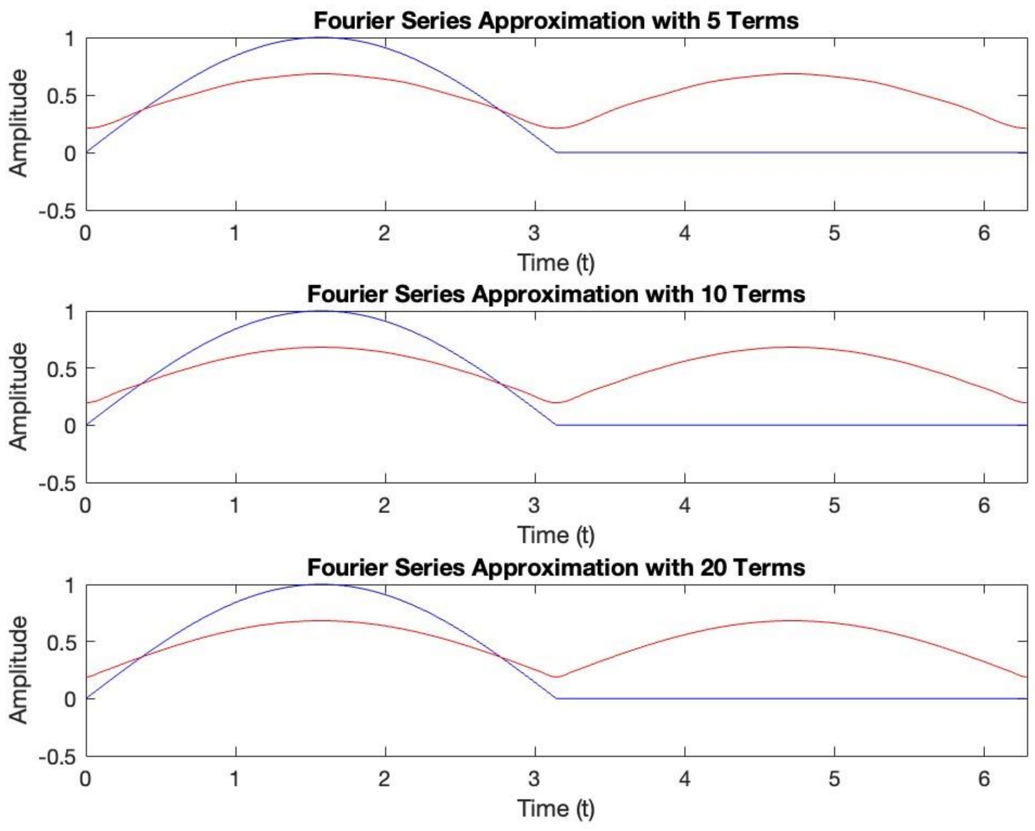
(b) Using MATLAB, plot the original signal and its Fourier series approximationsfor 5, 10, and 20 terms.

(c) Analyze the frequency spectrum of the signal and explain the presence of bothsine and cosine terms in the Fourier series.

(d) Comment on the physical interpretation of the Fourier coefficients.

```
T = 0:0.01:2*pi;

X = max(0, sin(t)); % Half-wave rectified sine wave



N = [5, 10, 20];
```

```
Y_limits  =  [-0.5   1];


For  I  =  1:3
        Approx  =  0;
        For  n  =  1:N(i)
                Approx  =  approx  +  ((2/pi)  *  (1/(1  -  (2*n)^2)))  *
cos(2*n*t);
        End
        Approx  =  0.5  +  approx;
        Subplot(3,1,i);
        Plot(t,  x,  'b',  t,  approx,  'r');
        Title(['Fourier  Series  Approximation  with  ',  num2str(N(i)),  '
Terms']);
        Xlabel('Time  (t)');
        Ylabel('Amplitude');
        Axis([0  2*pi  y_limits]);
End
```



Q11: A system is described by the following second-order differential equation:

$$\frac{d^2y(t)}{dt^2} + 4\frac{dy(t)}{dt} + 8y(t) = 5u(t)$$

where $u(t)$ is the input, and $y(t)$ is the output.

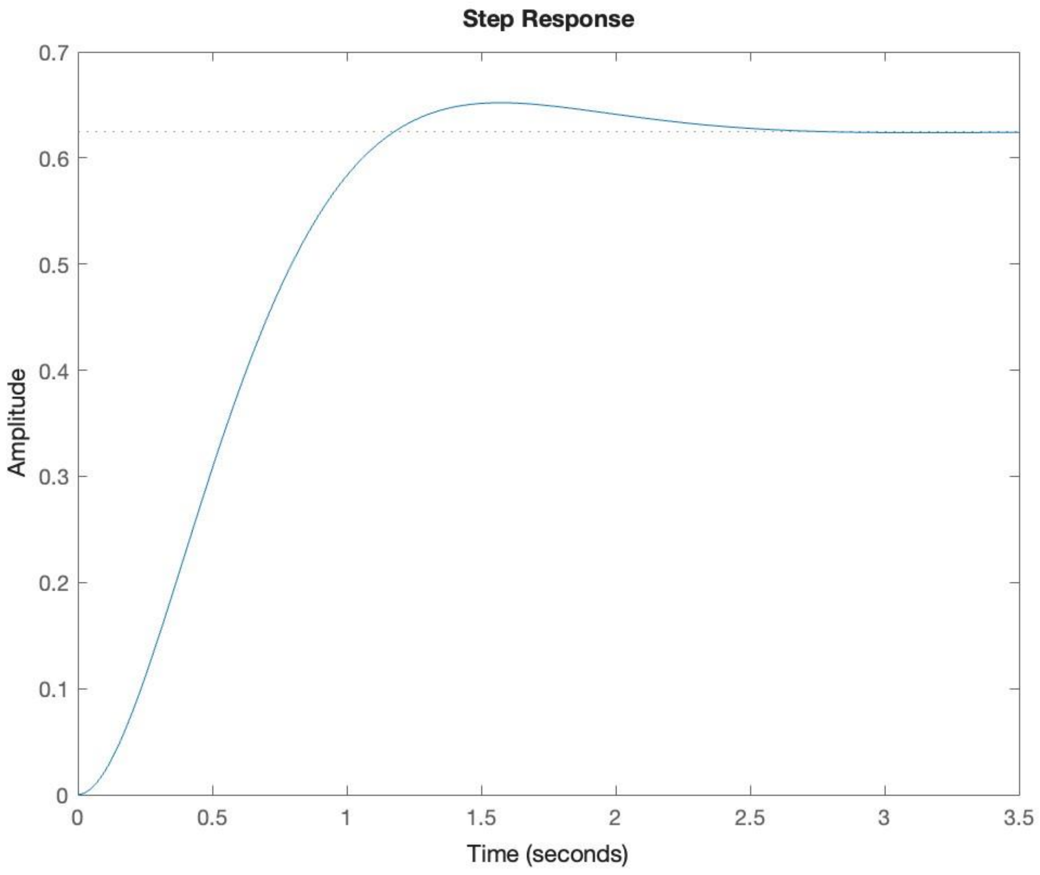
(a) Derive the transfer function $H(s) = \frac{Y(s)}{U(s)}$ for the system.

(b) Simulate the step response of the system using MATLAB.

(c) Plot the step response and determine if the system reaches a steady state.

```
% Define transfer function

S = tf(‘s’);

H = 5 / (s^2 + 4*s + 8);



% Step response

Figure; step(H);
```

Title('Step   Response');



Q12: A majority of modern trains and local transit vehicles utilize electric traction motors. The electric motor drive for a railway vehicle is shown in block diagram form in Figure 1, incorporating the necessary control of the velocity of the vehicle. After solving the differential equations and substituting system parameters we get 2. Ignore the disturbance torque $T_d(s)$.

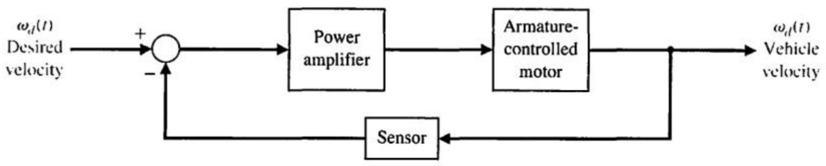


Figure 1: Speed control of an electric motor traction

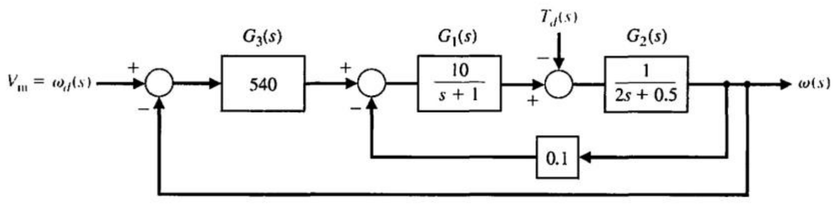


Figure 2: Speed control of an electric motor traction after substituting system parameters

(a) Find the overall transfer function of the system$\frac{\omega(s)}{\omega_d(s)}$.

(b) Implement the block diagram representation of the system shown in 2 inSIMULINK and find the overall transfer function.

(c) Simulate the step response and plot the figure.

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<Types xmlns=http://schemas.openxmlformats.org/package/2006/content-types>

<Default ContentType="application/vnd.mathworks.simulink.graphicalInterface+json" Extension="json"/>

<Default ContentType="image/png" Extension="png"/>

<Default ContentType="application/vnd.openxmlformats-package.relationships+xml" Extension="rels"/>

<Default ContentType="application/vnd.mathworks.simulink.mdl+xml" Extension="xml"/>

<Override ContentType="application/vnd.openxmlformats-package.core-properties+xml" PartName="/metadata/coreProperties.xml"/>

<Override ContentType="application/vnd.mathworks.package.coreProperties+xml" PartName="/metadata/mwcoreProperties.xml"/>

<Override ContentType="application/vnd.mathworks.package.corePropertiesExtension+xml" PartName="/metadata/mwcorePropertiesExtension.xml"/>

<Override ContentType="application/vnd.mathworks.package.corePropertiesReleaseInfo+xml" PartName="/metadata/mwcorePropertiesReleaseInfo.xml"/>

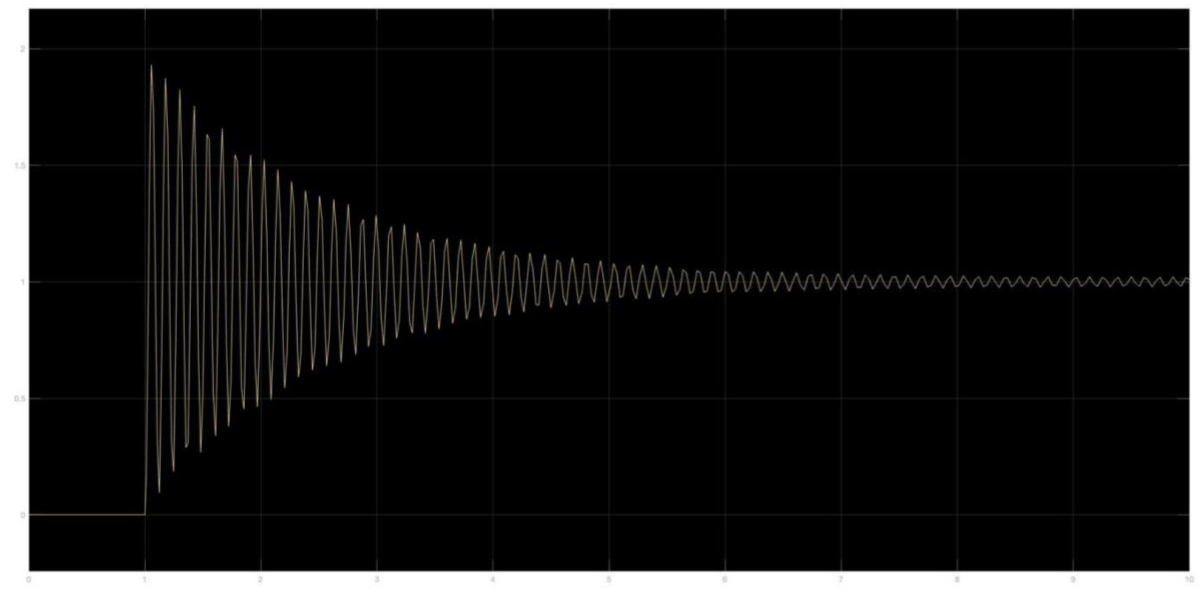
<Override ContentType="application/vnd.mathworks.simulink.configSet+xml" PartName="/simulink/configSet0.xml"/>

<Override ContentType="application/vnd.mathworks.simulink.configSetInfo+xml" PartName="/simulink/configSetInfo.xml"/>

<Override ContentType="application/vnd.mathworks.simulink.hdlparams+xml" PartName="/simulink/hdlParams.xml"/>

<Override ContentType="application/vnd.mathworks.simulink.mf0+xml" PartName="/simulink/modelDictionary.xml"/>

```
  <Override
ContentType=”application/vnd.mathworks.simulink.ScheduleCore+xml”
PartName=”/simulink/ScheduleCore.xml”/>
  <Override
ContentType=”application/vnd.mathworks.simulink.ScheduleEditor+xml”
PartName=”/simulink/ScheduleEditor.xml”/>
  <Override
ContentType=”application/vnd.mathworks.simulink.blockDiagram+xml”
PartName=”/simulink/windowsInfo.xml”/>
</Types>
```



Q13: Given the transfer function of a system:

$$H(s) = \frac{10(s+1)}{(s^2+6s+10)}$$

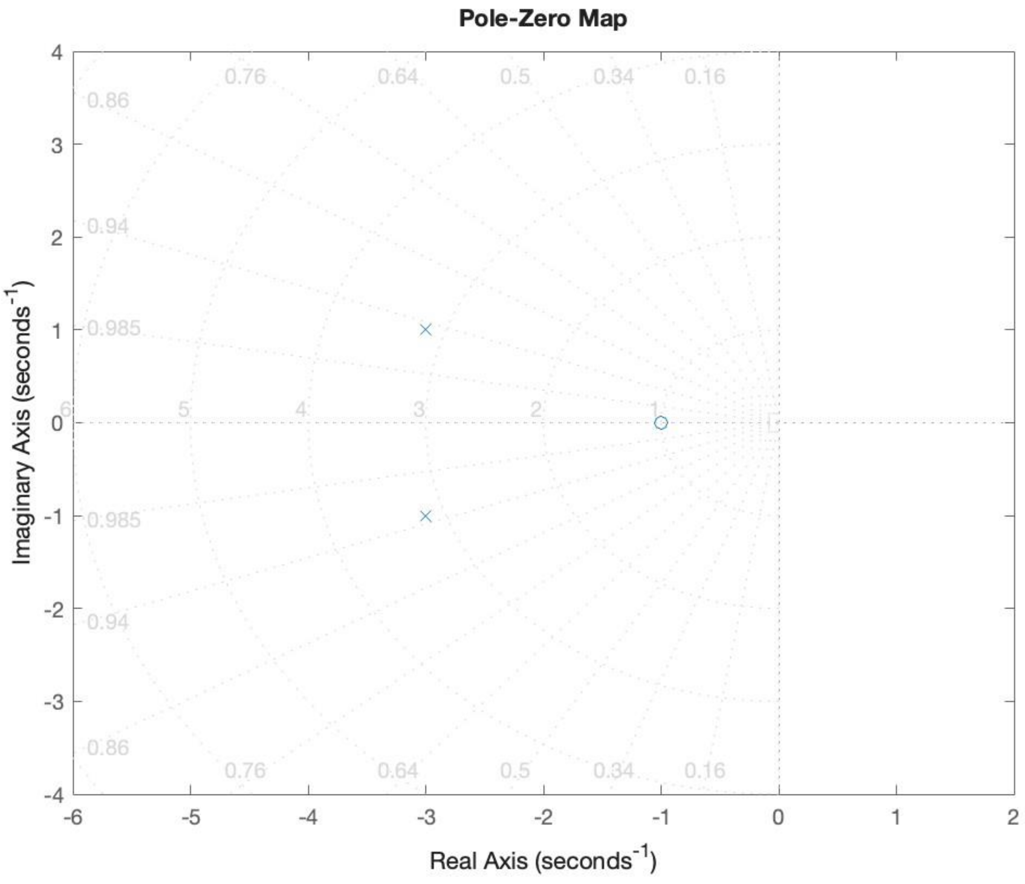
(a) Find the poles and zeros of the system.

(b) Plot the pole-zero map in MATLAB.

(c) Discuss the stability of the system based on the pole locations.

```
% Define transfer function

H = 10 * (s + 1) / (s^2 + 6*s + 10);


% Pole-zero map

Figure;

Pzmap(H);
```

```
Axis([-6  2  -4  4]);  %  Adjust  axis  limits

Title('Pole-Zero  Map');

Grid  on;
```



Q14: Consider the following transfer function:

$$H(s) = \frac{7}{s^2 + 3s + 2}$$

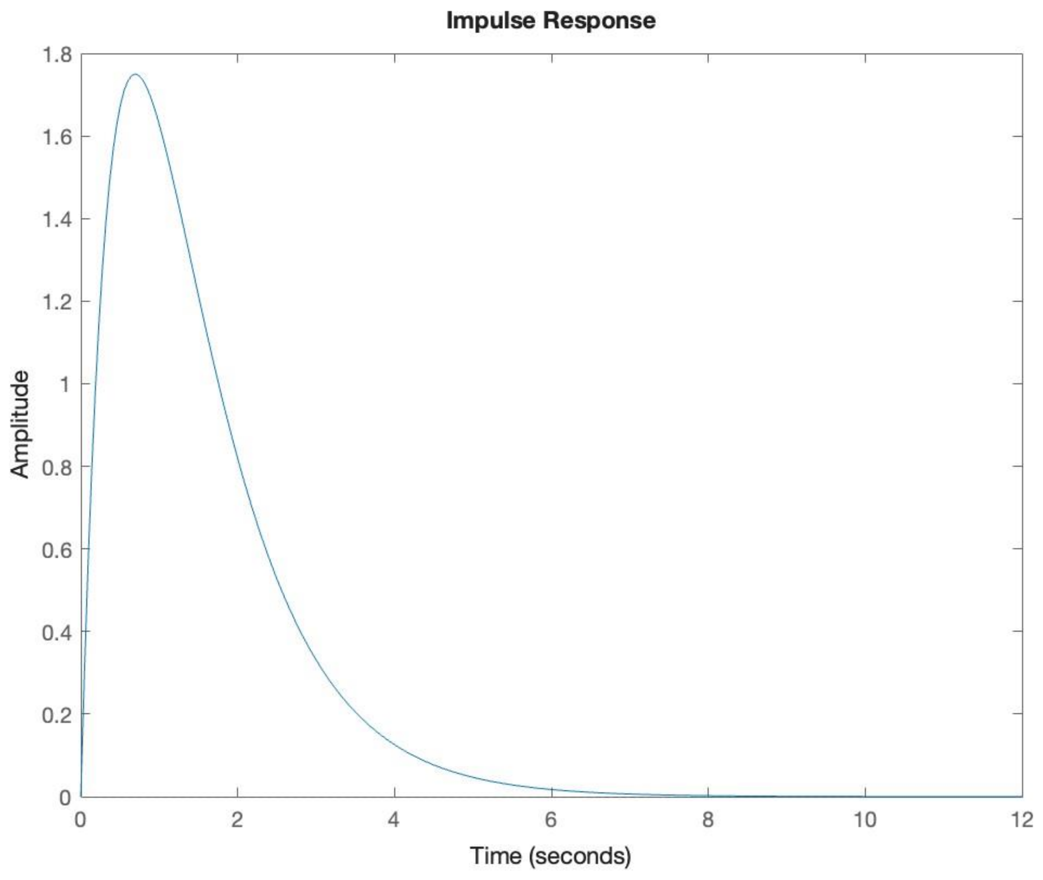
(a) Find the poles of the system.

(b) Simulate the impulse response of the system.

(c) Plot the impulse response and analyze if the system is stable based on theresponse.

```
% Define  transfer  function

H  =  7  /  (s^2  +  3*s  +  2);



% Impulse  response
```

```
Figure;  impulse(H);

Title('Impulse  Response');
```



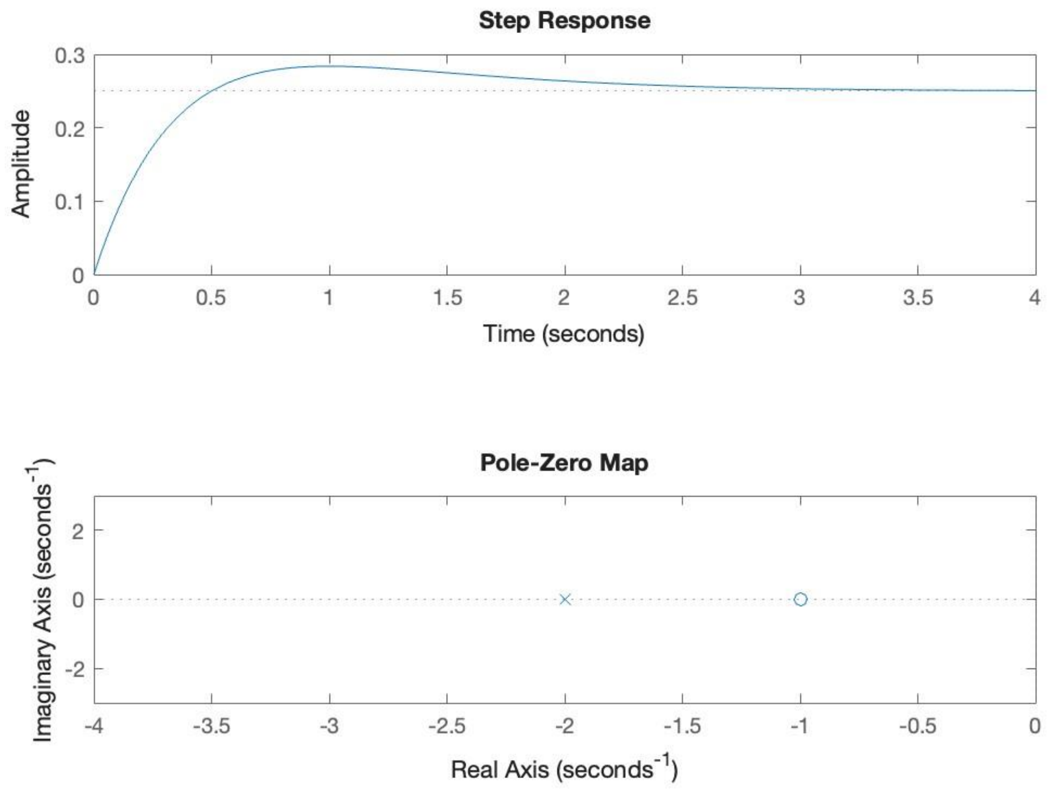
Q15: Given a system with the transfer function:

$$H(s) = \frac{(s+1)}{(s^2+4s+4)}$$

(a) Find the poles and zeros of the system.
(b) Plot the step response of the system.
(c) Generate the pole-zero map and comment on the system's stability.

```
% Define transfer function
H = (s + 1) / (s^2 + 4*s + 4);
```

```
% Step response and Pole-Zero Map
Figure;
Subplot(2,1,1); step(H);
Title('Step Response');
Subplot(2,1,2); pzmap(H);
Title('Pole-Zero Map');
Axis([-4 0 -3 3]); % Adjust axis limits for better visualization
```


Step Response
Amplitude
Time (seconds)
Pole-Zero Map
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})