Vishnu K

S5

Roll No : 71

SIGNALS AND SYSTEMS ASSIGNMENT

Q1: A trapezoidal signal is described as follows

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

Simulate the following and generate the corresponding plot.

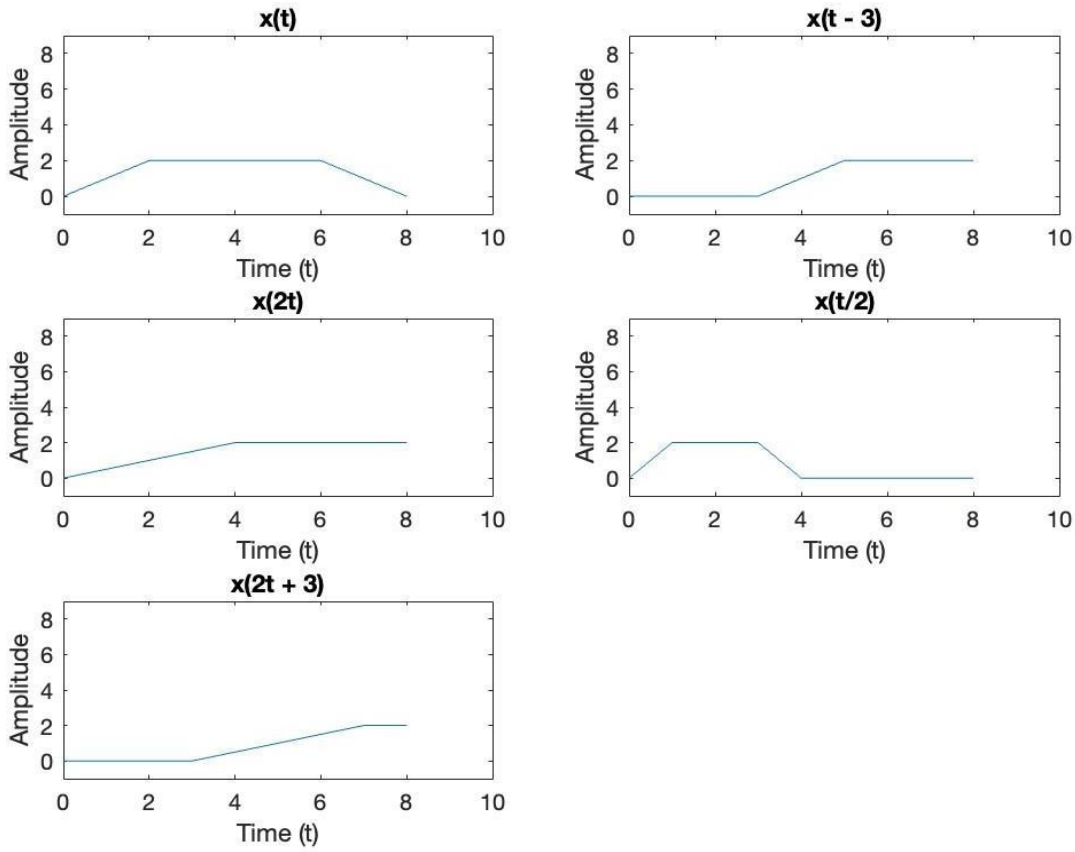
(a) $x(t)$
(b) $x(t-3)$
(c) $x(2t)$
(d) $x(\frac{1}{2}t)$
(e) $x(2t+3)$

Code

```
% 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 1/2 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 $h(t)$:

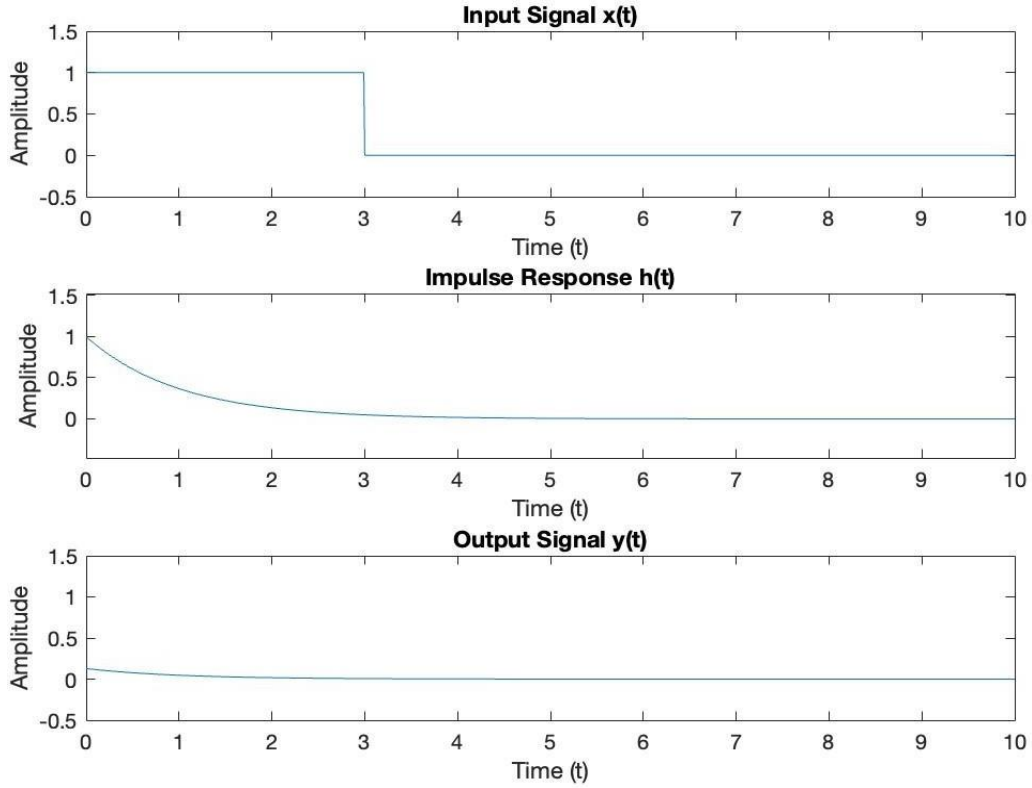
$$x(t) = \begin{cases} 1 & 0 \leq t < 3 \\ 0 & \text{otherwise} \end{cases}$$

$$h(t) = e^{-t}, \quad t \geq 0$$

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

Code

```
% 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]);
```



(d) **System Behavior**: Convolution of x(t) (input) with h(t)(impulse response) represents the **output response** of a Linear Time-Invariant (LTI) system.

Implications: If h(t)=e−t, it indicates an **exponential decay** function, suggesting that the system responds strongly at first and then gradually diminishes. This behavior is typical of systems with a damping effect.The system behaves as a **low-pass filter**, attenuating high-frequency components while preserving low-frequency ones.

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

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

(a) Define the input square wave in MATLAB.

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

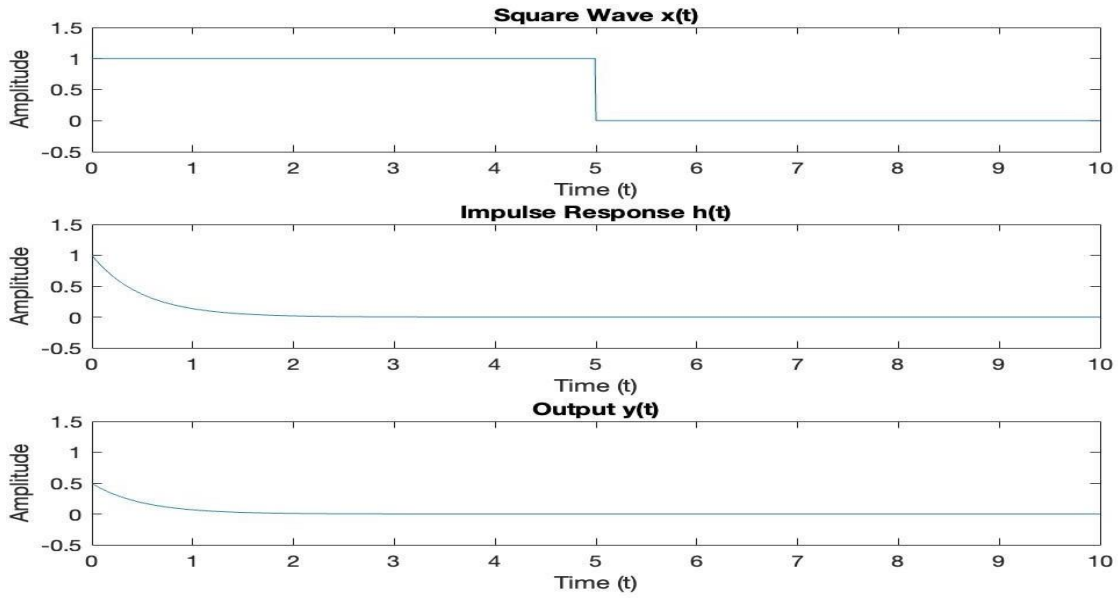
(c) Plot the input, impulse response, and output signals.

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

Code

```
% 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]);
```

Output



(d) S**ystem Response**: Convolution with h(t)=e−2t applies a faster decay compared to e−t making this system respond more quickly but also decay faster.

Square Wave Input: The square wave is a non-smooth periodic signal with both high and low frequencies.

Result: The system's faster decay results in **smoothing** the square wave's sharp edges, making it look more like a rounded wave, removing high frequencies and oscillating initially before stabilizing.

Conclusion: This system also acts like a low-pass filter, but with a **quicker response and faster decay**, producing a less sustained output.

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:

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

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

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

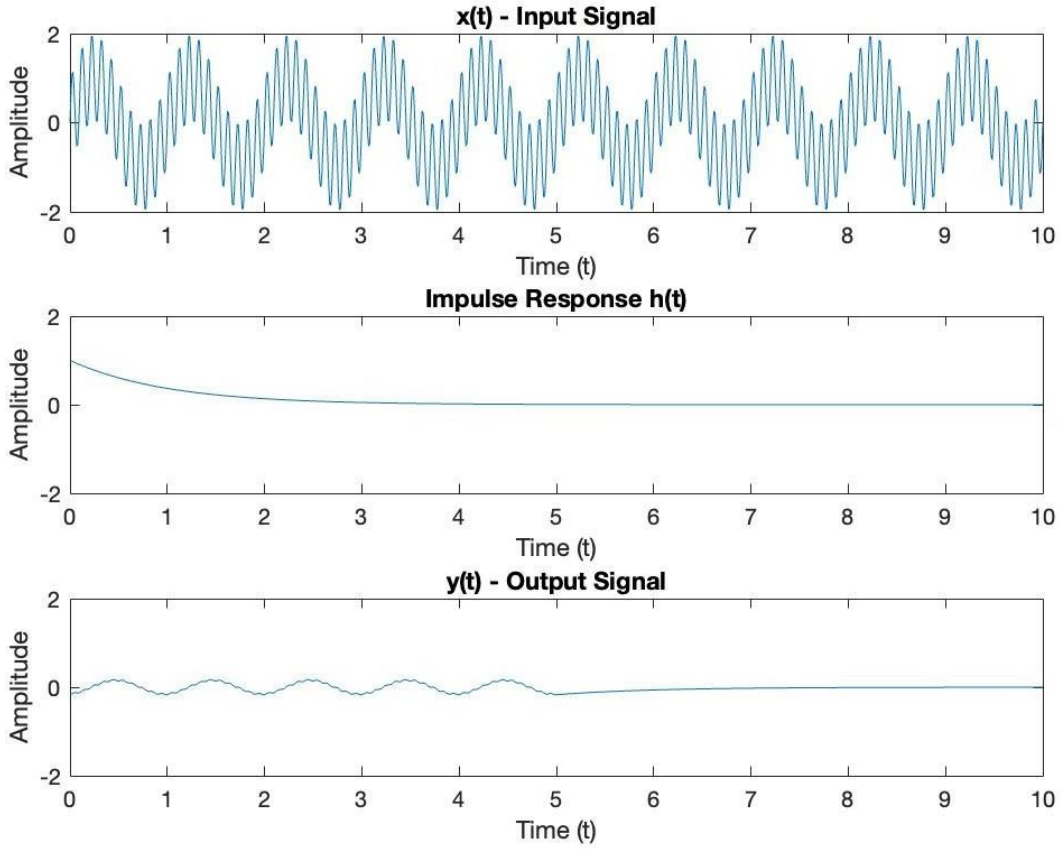
(c) Plot the output signal $y(t)$.

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

Code

```
% 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]);
```

Output



(d) **Low-Pass Filter Effect**: Given h(t)=e−t this impulse response decays gradually, meaning that low frequencies will pass through more effectively than high frequencies.

Low-Frequency Component (f1=1 Hz): This component will pass through relatively unchanged since it aligns more closely with the system's low-pass characteristics.

High-Frequency Component (f2=10 Hz): The higher frequency will be attenuated, resulting in a reduced amplitude in the output.

The system will filter out most of the high-frequency sinusoid, allowing the low-frequency sinusoid to pass through, which aligns with typical low-pass filtering behavior.

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.

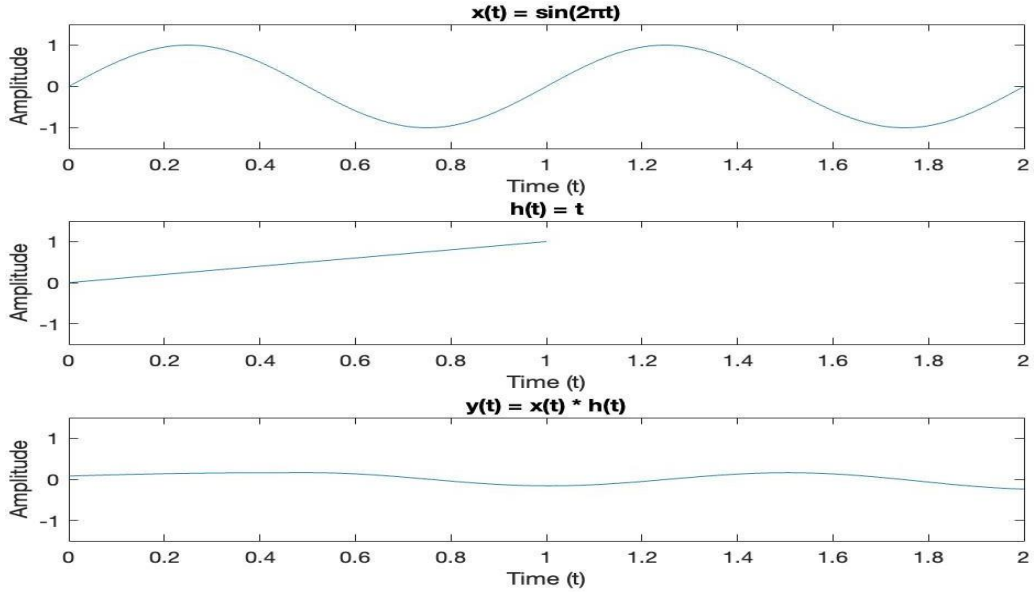
(b) Compute the convolution $y(t) = x(t) * h(t)$ using MATLAB's `conv` function.

(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.

Code

```
% 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]);
```



(d) **Signal Convolution Interpretation**: Convolution of x(t)=sin(2πt) with h(t)=t produces an output y(t) that represents the **cumulative effect** of applying h(t) to x(t).

Effect of h(t)=t The linear ramp function t represents a **linearly increasing weight** over time. This means that as time progresses, more weight is given to past values of x(t), emphasizing its earlier parts.

Result: The output y(t) will gradually build up in amplitude, reflecting a weighted accumulation of x(t).

Physically, this convolution might represent a system where responses increase linearly over time, which could simulate processes like heating or loading where the effect grows over time.

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.

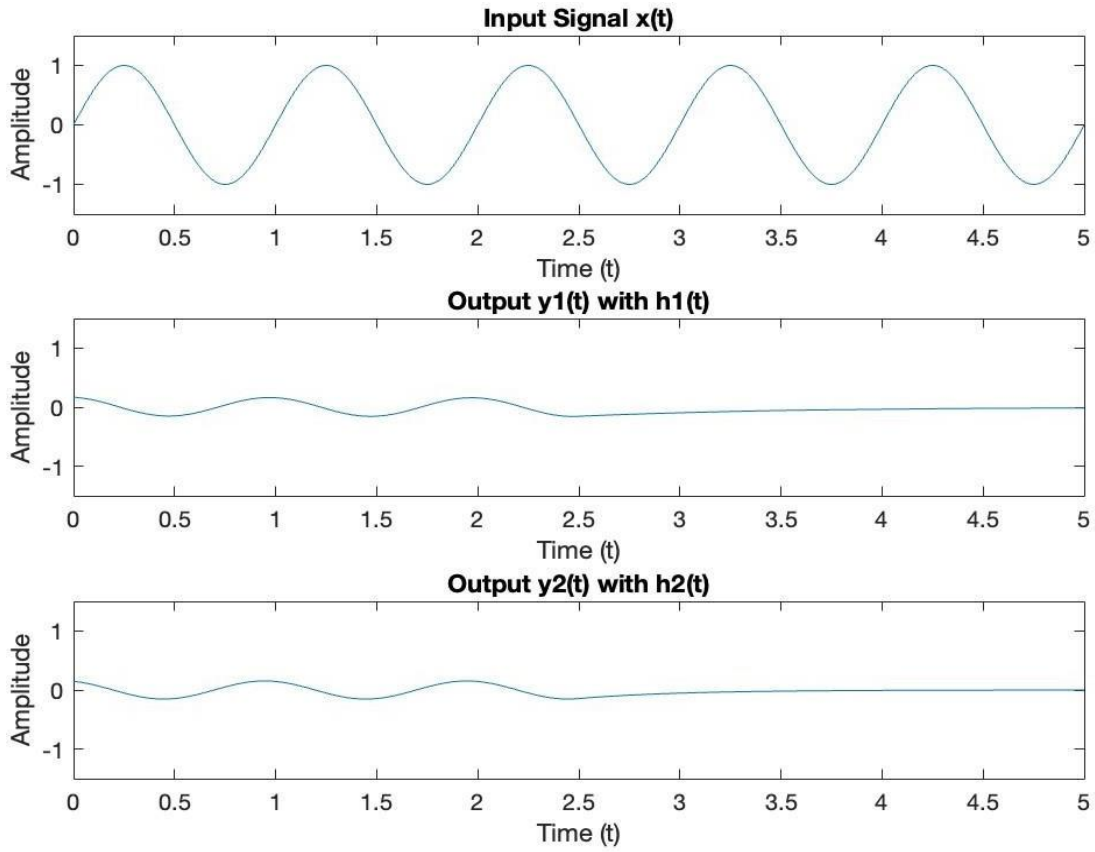
(b) Plot the output signals for both cases.

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

Code

```
% 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]);
```



(c) **Impulse Responses**: The two impulse responses given are h1(t)=e−t and h2(t)=e−2t

h1(t)=e−t: This is a slower-decaying function, meaning it has a **longer-lasting impact** on the output. It retains more of the input signal over time.

h2(t)=e−2t: This decays faster, so it **dampens the response more quickly**, leading to a shorter-lived impact of the input signal on the output.

Output Comparison:

- With h1(t) the output will have a **smoother, more sustained response**.
- With h2(t), the output will respond and decay more rapidly, reflecting a faster drop-off in signal.

Changing the impulse response affects the output by controlling how long past inputs affect the system's current output. A slower decay (as in h1(t)) means longer influence, while a faster decay (as in h2(t)) produces a quicker but shorter-lasting response.

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

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

(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 for the given number of terms.

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

(7)

(a) $x(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{n\omega_0 t}{T}\right) + b_n \sin\left(\frac{n\omega_0 t}{T}\right) \right)$

$\omega_0 = \frac{2\pi}{T}$

since $T = 2\pi$, $\omega_0 = 1$

$$a_0 = \frac{1}{T}\int_0^T x(t)\,dt = \frac{1}{2\pi}\int_0^{\pi} 1\,dt + \int_{\pi}^{2\pi} -1\,dt$$

$$= \frac{1}{2\pi}\left(\pi + (-\pi)\right) = \frac{1}{2\pi} \times 0 = 0 \text{ //}$$

$$a_n = \frac{1}{\pi}\int_0^{2\pi} x(t)\cos(nt)\,dt = \frac{1}{\pi}\int_0^{\pi} 1\cdot\cos nt\,dt + \int_{\pi}^{2\pi} (-1)\cos nt\,dt$$

$$= \frac{1}{\pi}\int_0^{\pi} \cos(nt)\,dt - \int_{\pi}^{2\pi} \cos(nt)\,dt$$

$$= \frac{1}{\pi}\left(\frac{\sin(n\pi)}{n} - 0 - \left(\frac{\sin(2n\pi)}{n} - \frac{\sin(n\pi)}{n} \right) \right) = 0$$

$a_n = 0$ for all n

$$b_n = \frac{1}{\pi}\int_0^{2\pi} x(t)\sin(nt)\,dt = \frac{1}{\pi}\int_0^{\pi} 1\cdot\sin(nt)\,dt + \int_0^{2\pi} (-1)\sin(nt)\,dt$$

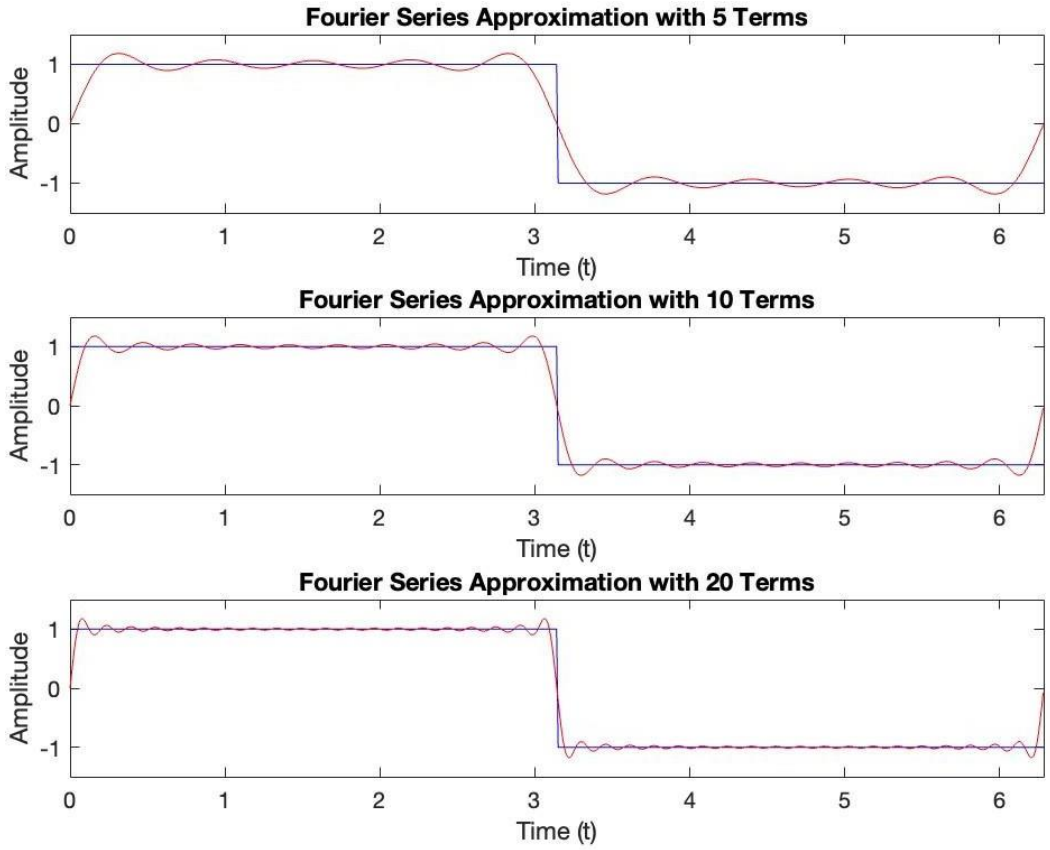
$$= \frac{1}{\pi}\int_0^{\pi} \sin(nt)\,dt + -\int_{\pi}^{2\pi} \sin(nt)\,dt$$

$$= \frac{1}{\pi}\left(\frac{-\cos(n\pi)}{n} + \frac{1}{n} + \frac{\cos(n\pi)}{n} - \frac{1}{n} \right)$$

$$b_n = \frac{2(-1)^{n+1}}{n\pi}$$

Code

```
% 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
```



(d) As the number of terms in the Fourier series increases, the approximation of the square wave improves. Initially, the series captures the main features of the wave, such as its amplitude and period. However, the convergence can be analyzed further:

1. **Gibbs Phenomenon:** Near the discontinuities of the square wave (at t=0 and t=π), the Fourier series exhibits overshoot or ringing, known as the Gibbs phenomenon. This overshoot approaches approximately 9% of the amplitude of the jump, regardless of the number of terms used.
2. **Uniform Convergence:** While the Fourier series converges to the square wave at all points where the function is continuous, it converges to the average of the left and right limits (0 in this case) at points of discontinuity.

3. **Rate of Convergence:** The Fourier series converges to the square wave more quickly away from discontinuities and slower near them. For a square wave, the convergence is slower due to the sharp transitions between values.

Q8: A periodic sawtooth wave is defined by:

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

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 Gibbs phenomenon occurs at discontinuities.

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

(8a) $x(t) = t/\pi$

$$a_0 = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\,dt = \frac{1}{2\pi^2}\left(\frac{\pi^2}{2} - \frac{\pi^2}{2}\right) = 0$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} \frac{t}{\pi}\cos|nt|\,dt = \frac{1}{\pi^2}\int_{-\pi}^{\pi} t\cos(nt)\,dt$$

$a_n = 0$ for all n

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} x(t)\sin|nt|\,dt = \frac{1}{\pi}\int_{-\pi}^{\pi} \frac{t}{\pi}\sin|nt|\,dt$$

Applying integration by parts.

$$b_n = \frac{1}{\pi^2} \times 0 = 0 \qquad b_n = \frac{2(-1)^{n+1}}{n}$$

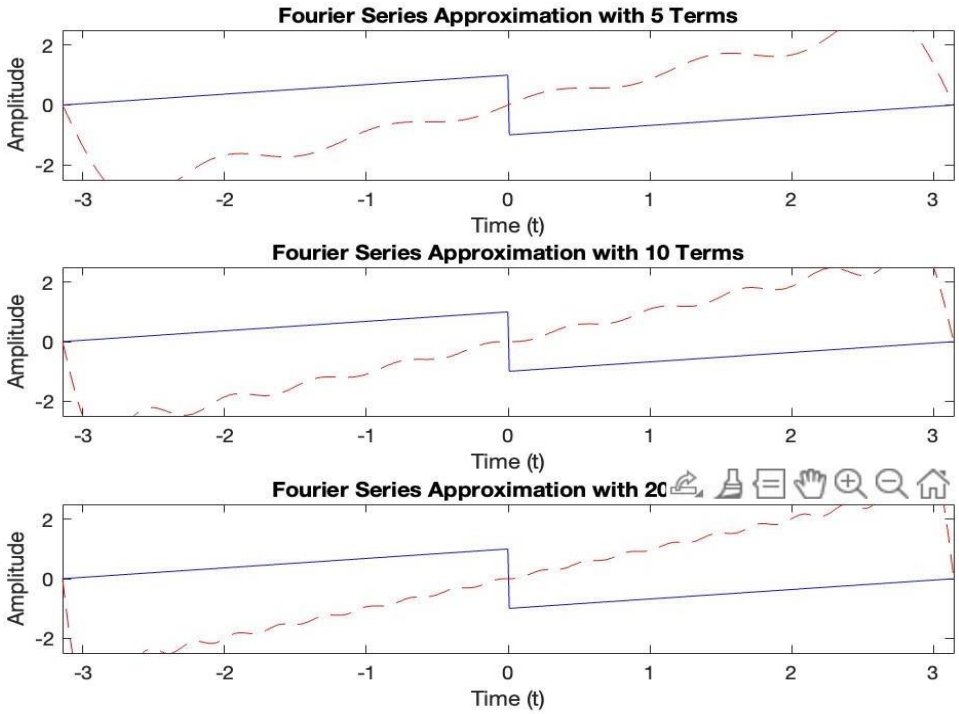
due to nature of sawtooth wave

$$x(t) = \sum_{n=1}^{\infty} \frac{2(-1)^{n+1}}{n}\sin(nt)$$

=

Code

```
% 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
```



(C) The accuracy of approximation, particularly in the context of Fourier series, refers to how well the series represents a given function. For smooth functions, Fourier series can converge uniformly to the function. However, at points of discontinuity, the series often exhibits the Gibbs phenomenon.

The Gibbs phenomenon occurs when approximating a function with a Fourier series that has a jump discontinuity. As the number of terms in the series increases, the series approaches the function, but there is a persistent overshoot near the discontinuity.

The Gibbs phenomenon arises because Fourier series, composed of smooth functions, struggle to accurately represent sharp changes in a function. The resulting oscillations lead to overshooting at points of discontinuity, reflecting the inherent limitations of using continuous functions to approximate discontinuous ones.

(D) Increasing the number of harmonics in a Fourier series generally enhances the quality of the approximation by improving accuracy and reducing truncation errors. However, it can also lead to artifacts, especially near discontinuities, and introduce oscillations that may negatively affect the approximation's visual quality. The key is to find an optimal balance between the number of harmonics and the desired fidelity of the approximation for a specific application.

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 MATLAB with 5, 10, and 20 terms.

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

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

$= \pi$

(a) $a_0 = \frac{1}{T}\int_0^T x(t)\,dt = \frac{1}{2\pi}\int_0^{2\pi} x(t)\,dt$

$\int_0^{2\pi} x(t)\,dt = \frac{\pi}{2} + \frac{\pi}{2} = \frac{2\pi}{2} = \pi$

$a_0 = \frac{1}{2\pi}\cdot\pi = 1/2$

$=$

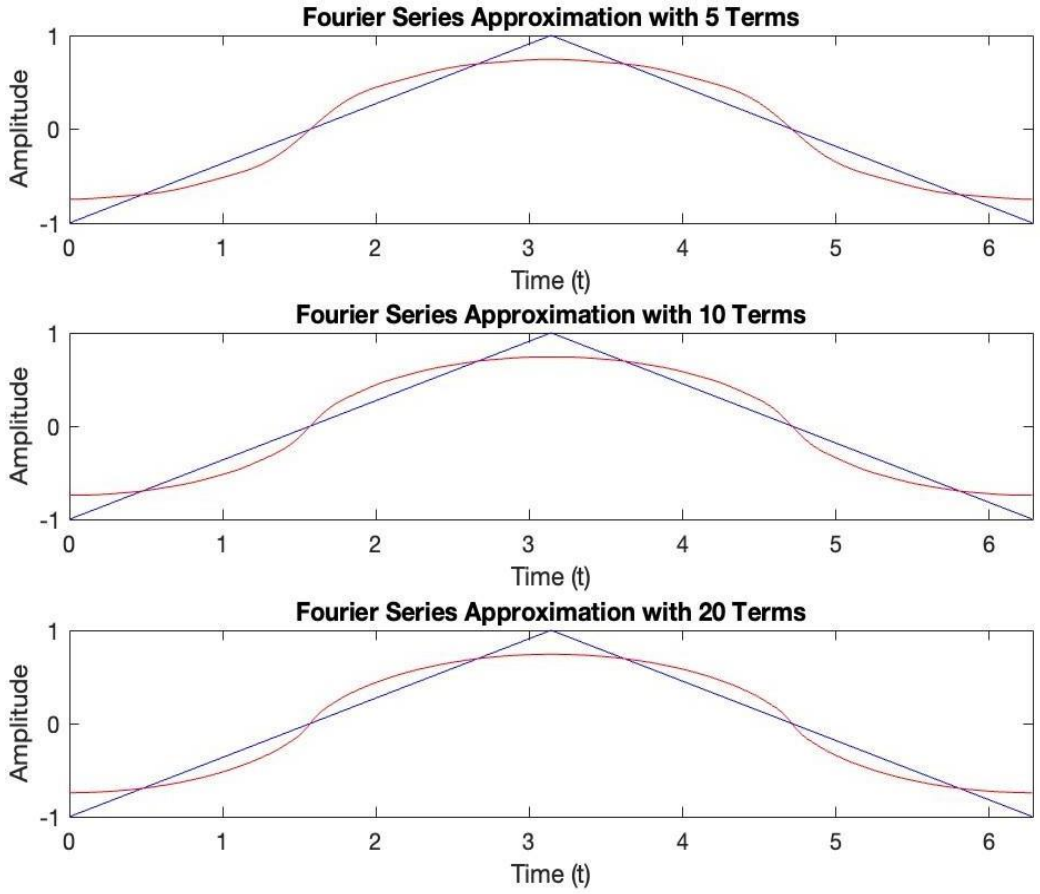
$a_n = \frac{1}{\pi}\int_0^{2\pi} x(t)\cos(nt)\,dt = \frac{1}{\pi}\int_0^{\pi} \frac{t}{\pi}\cos(nt)\,dt + \int_0^{2\pi}\left(-t/\pi + 2\right)\cos(nt)\,dt$

Simplify using boundary condition $a_n = \begin{cases} 0 & n \text{ even} \\ 4/n\pi^2 & n \text{ odd} \end{cases}$, $b_n = \frac{1}{\pi}\int_0^{2\pi} x(t)\sin(nt)\,dt$

$b_n = -(-1)^n/n + 2(-1)^n/n = (-1)^n/n$ //

Code

```
% 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
```



(c) **Odd Symmetry**: A triangular wave is often defined to be odd, meaning it exhibits symmetry about the origin. This means that $f(-t)=-f(t)$. Because of this odd symmetry, all the cosine coefficients in the Fourier series will be zero. Cosine functions are even, so they do not contribute to the series representation of an odd function.

Periodicity: A triangular wave is periodic, repeating its shape at regular intervals. This periodicity allows it to be represented using a Fourier series.

Impact on fourier coefficients:-

Fourier Series Representation: The Fourier series of a function f(t) can be expressed as:

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(nt) + b_n \sin(nt))$$

- **a0 Coefficient** For an odd function like the triangular wave, the average value over one period is zero. Therefore, a0=0

- **Cosine Coefficients an**:
  - Since the triangular wave is odd, all cosine coefficients an=0. This is a direct consequence of the wave's symmetry, as cosine functions cannot represent odd functions.

- **Sine Coefficients bn:**
  - The sine coefficients, bn, will be non-zero and can be calculated using the formula:

$$b_n = \frac{2}{T} \int_0^T f(t) \sin\left(\frac{2\pi nt}{T}\right) dt$$

  - The sine functions are odd, which aligns with the properties of the triangular wave. Thus, the Fourier series for the triangular wave will only contain sine terms.

(d) 1. **Convergence near Discontinuities**:

- **Square Wave**: The convergence is slow near the discontinuities due to the sharp transitions, leading to pronounced oscillations (Gibbs phenomenon).
- **Triangular Wave**: The convergence is relatively smooth and faster near its endpoints because of its continuous nature.

2. **Overall Behavior**:

- **Square Wave**: Although it converges pointwise, the oscillations persist near discontinuities, leading to slower effective convergence.
- **Triangular Wave**: The Fourier series converges uniformly to the function, especially as more terms are added. The smoothness of the triangular wave leads to a more stable convergence behavior

The Fourier series for the triangular wave converges more quickly and smoothly compared to that of the square wave. The continuous nature of the triangular wave allows for better approximation with fewer oscillations, while the square wave's sharp transitions cause slower convergence and more noticeable Gibbs phenomenon effects.

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

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

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

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

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

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

(10) (a)

$$x(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(nt) + b_n \sin(nt)$$

$$a_0 = \frac{1}{T}\int_0^T x(t)\,dt = \frac{1}{2\pi}\int_0^{\pi} \sin t\,dt = -\cos(\pi) + \cos(0)$$

$$= 1 + 1 = 2$$

$$a_0 = \frac{1}{2\pi} \times 2 = \frac{1}{\pi}$$

$$a_n = \frac{1}{\pi}\int_0^{\pi} \sin t \cos(nt)\,dt = \frac{1}{2\pi}\int_0^{\pi} \sin(1+n)t\,dt + \int_0^{\pi} \sin(1-n)t\,dt$$

Simplifying, $a_n = 0$ (for odd n) and $a_n = \frac{1}{n\pi}$ (for even n)

$$b_n = \frac{1}{\pi}\int_0^{\pi} \sin(t)\sin(nt)\,dt + \int_{\pi}^{2\pi} 0 \cdot \sin(nt)\,dt$$

$$= \frac{1}{\pi}\int_0^{\pi} \sin t \sin(nt)\,dt$$

$$= \frac{1}{2\pi}\left[\frac{\sin(n-1)t}{(n-1)}\right]_0^{\pi} - \left[\frac{\sin(n+1)t}{(n+1)}\right]_0^{\pi}$$

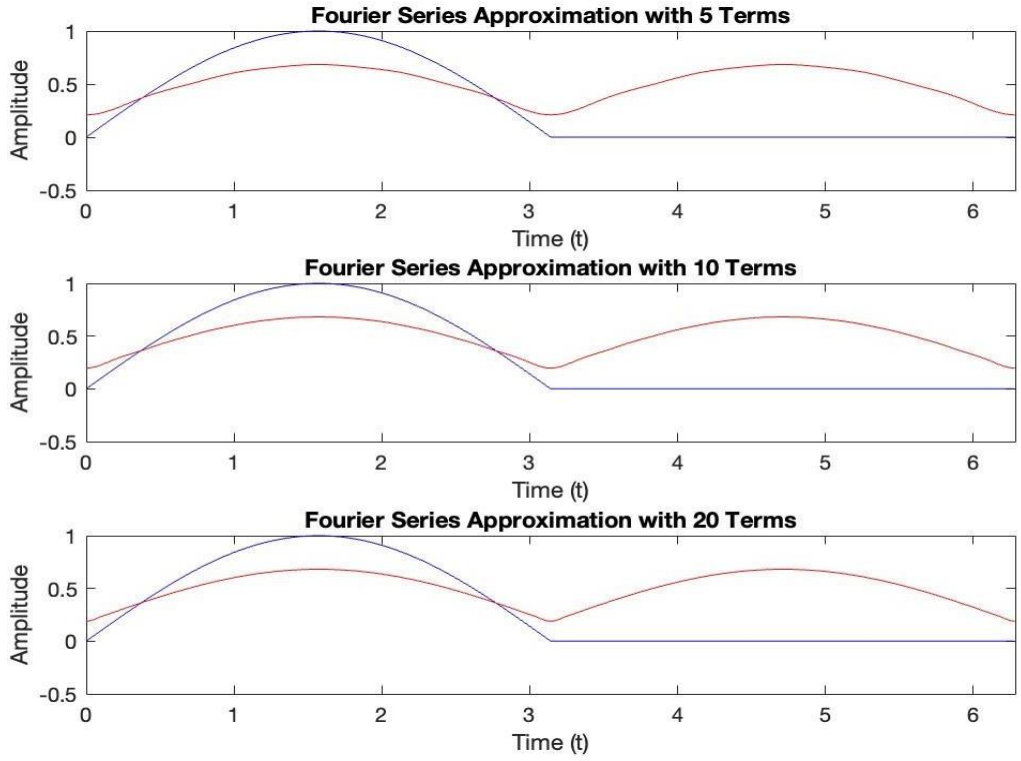
Since $\sin(k\pi) = 0$ for any k.

$$b_n = 0$$

$$x(t) = \frac{1}{\pi} + \sum_{k=0}^{\infty} \frac{2}{(2k+1)\pi} \sin(2k+1)t$$

Code

```
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
```



(c) Since the Fourier series consists only of **sine terms** (odd harmonics), the frequency spectrum will have components at odd multiples of the fundamental frequency. The presence of both sine and cosine terms would typically suggest both even and odd harmonics, but in this case:

- **Only sine terms** are present because the original signal x(t) is odd about π(it only has positive values in the first half of the period).
- The spectrum thus has contributions from odd harmonics only (e.g., $1\omega_0, 3\omega_0, 5\omega_0, \ldots$), where $\omega_0 = 2\pi/T = 1$.

(d) The Fourier coefficients bn indicate the amplitude of each harmonic component in the signal. Physically:

- **Lower-frequency terms (e.g., b1)** have higher amplitudes, which means that the fundamental frequency dominates the shape of the signal.

- **Higher-frequency terms** contribute to the sharper edges of the rectified wave. Increasing the number of terms in the Fourier series improves the approximation, capturing more of the signal's details.

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

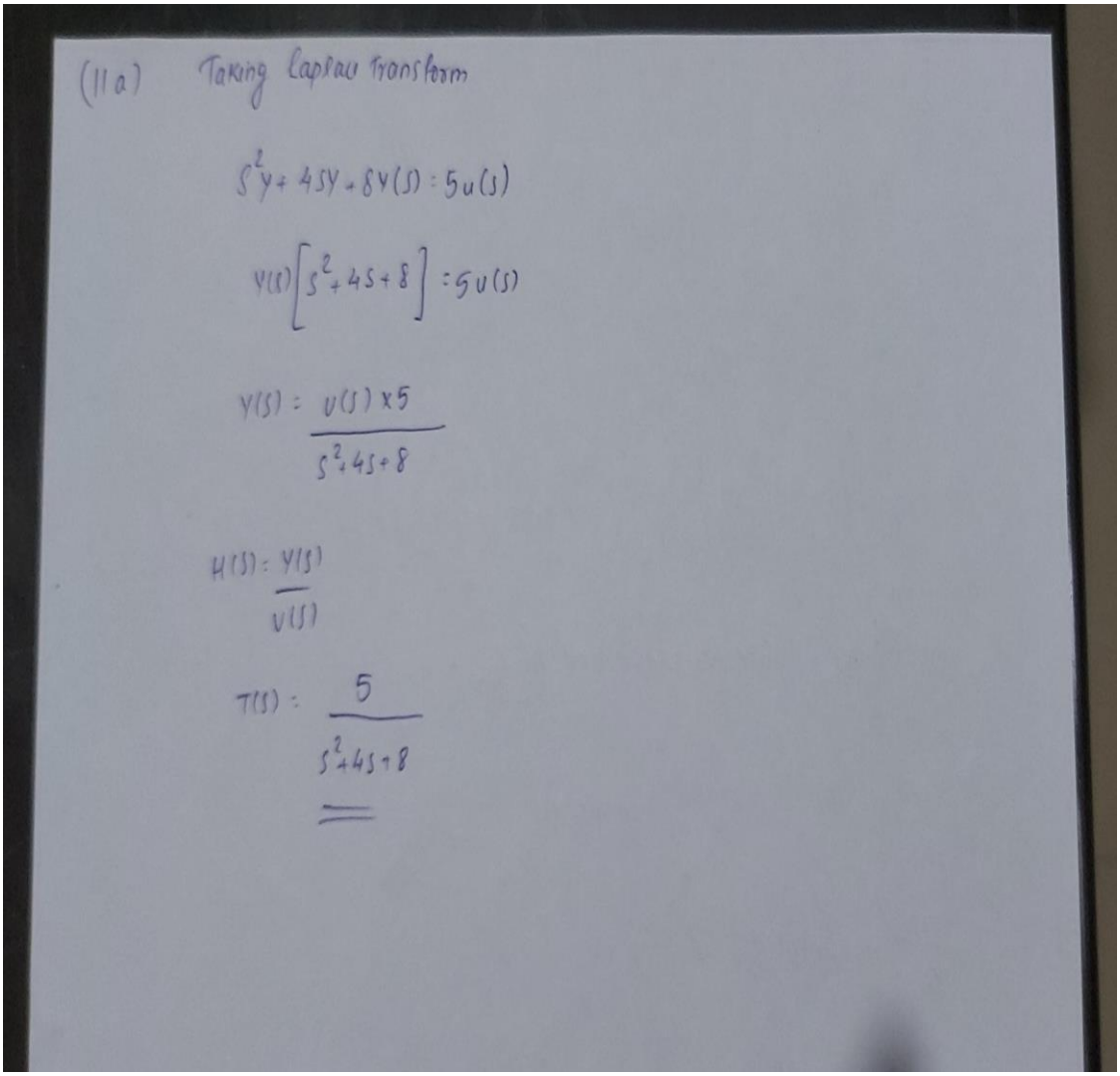
$$\frac{d^2y(t)}{dt^2} + 4\frac{dy(t)}{dt} + 8y(t) = 5u(t)$$

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.

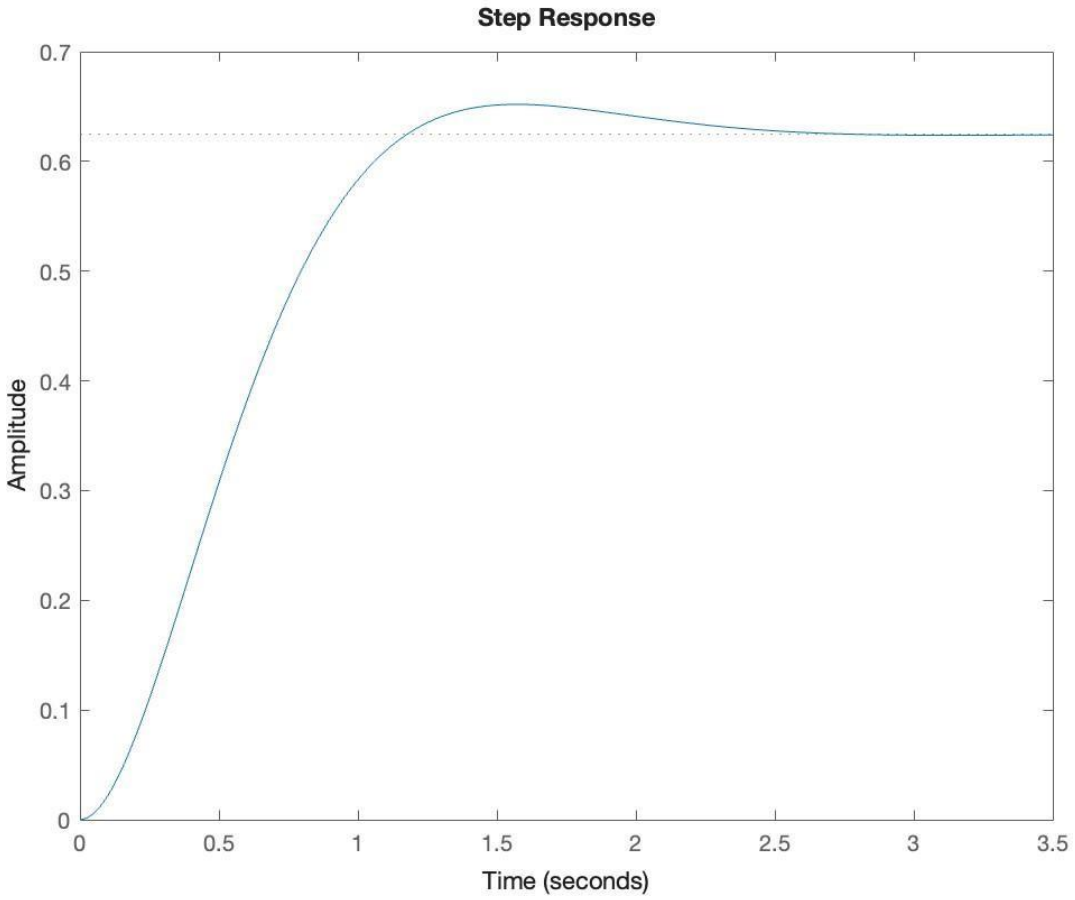
(b) Simulate the step response of the system using MATLAB.

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


(11 a) Taking Laplace transform
$s^2Y + 4sY + 8Y(s) = 5u(s)$
$Y(s)\left[s^2 + 4s + 8\right] = 5u(s)$
$Y(s) = \frac{u(s) \times 5}{s^2 + 4s + 8}$
$H(s) = \frac{Y(s)}{U(s)}$
$T(s) = \frac{5}{s^2 + 4s + 8}$

(b) Code

```
% Define transfer function
s = tf('s');
H = 5 / (s^2 + 4*s + 8);

% Step response
figure; step(H);
title('Step Response');
```



(c) Since the system is stable (all poles have negative real parts), the step response will eventually reach a steady-state value.

The imaginary components (±2j) suggest that the system will exhibit **damped oscillations** as it approaches steady state. Therefore, the response will oscillate and then settle to a final, steady value. Yes, the system **reaches a steady state** because it is stable. The step response will

oscillate initially due to the complex poles but will eventually settle to a constant value as t→∞.

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 $T_d(s)$.

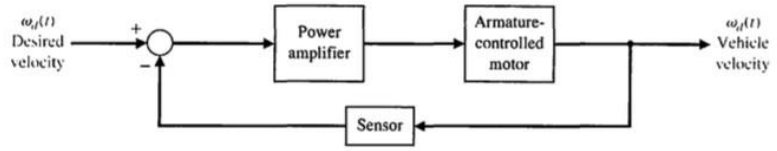


Figure 1: Speed control of an electric motor traction

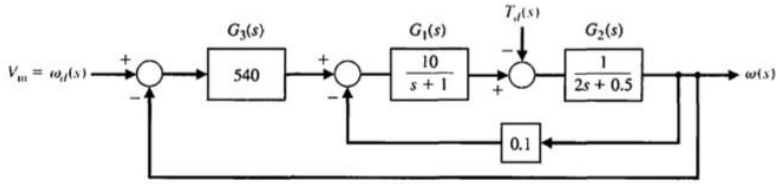
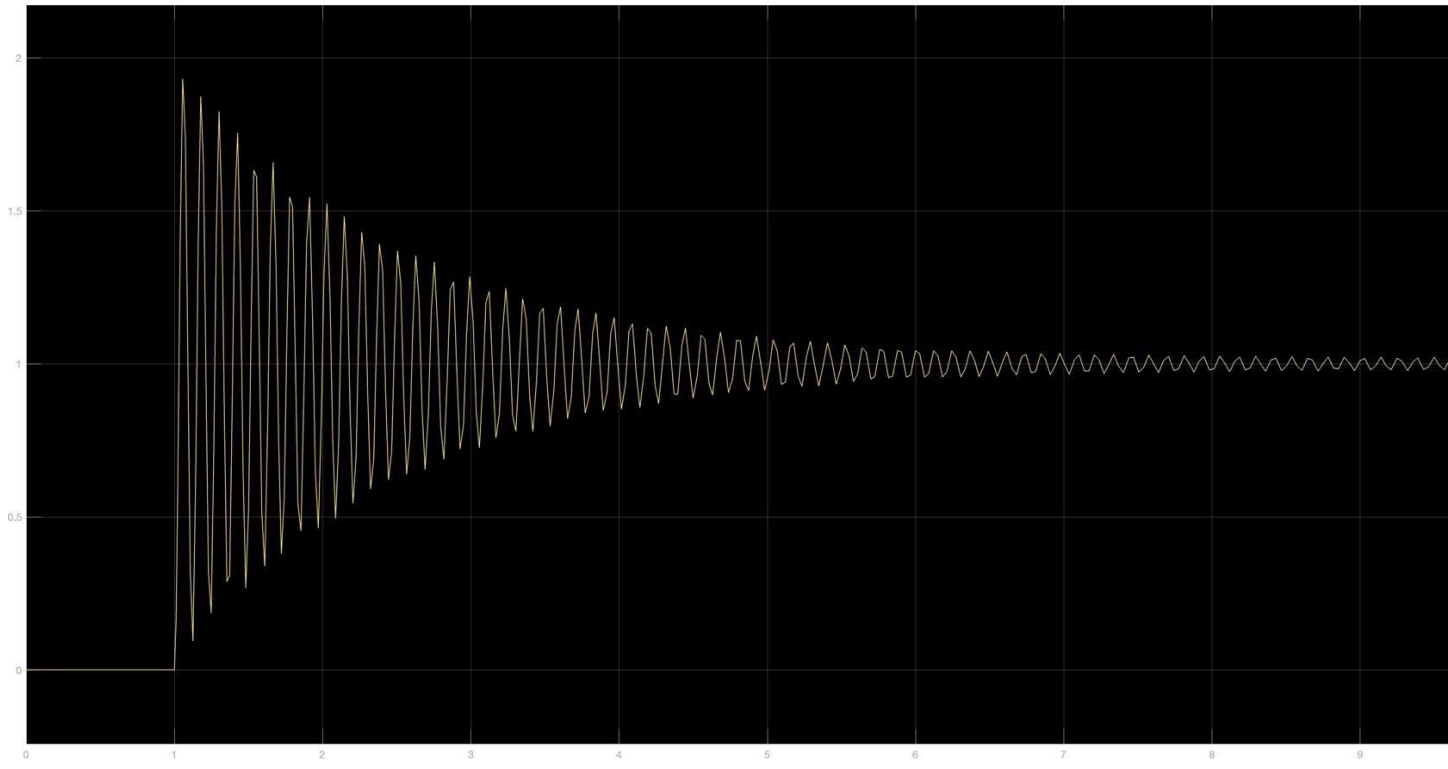


Figure 2: Speed control of an electric motor traction after substituting system parameters

(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 in SIMULINK and find the overall transfer function.

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



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.

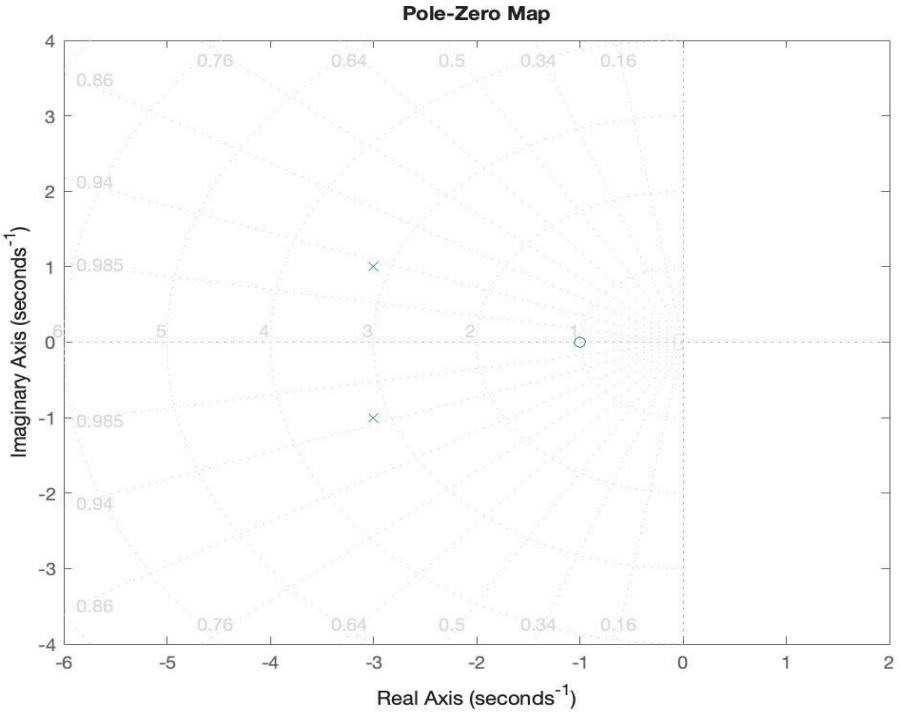
(c) Discuss the stability of the system based on the pole locations.

(a) Zeros are the values of s that make the numerator zero the zero is at s=−1

Poles are the values of s that make the denominator zero. For s2+6s+10 solve the quadratic eqn. Therefore, the poles are at s=−3+j s=−3-j

(b) Code

```
% 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;
```



(c) The **zero** is at s=−1. The **poles** are at s=−3+j s=−3-j confirming the system is **stable** because all poles are in the left half of the complex plane.

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 the response.

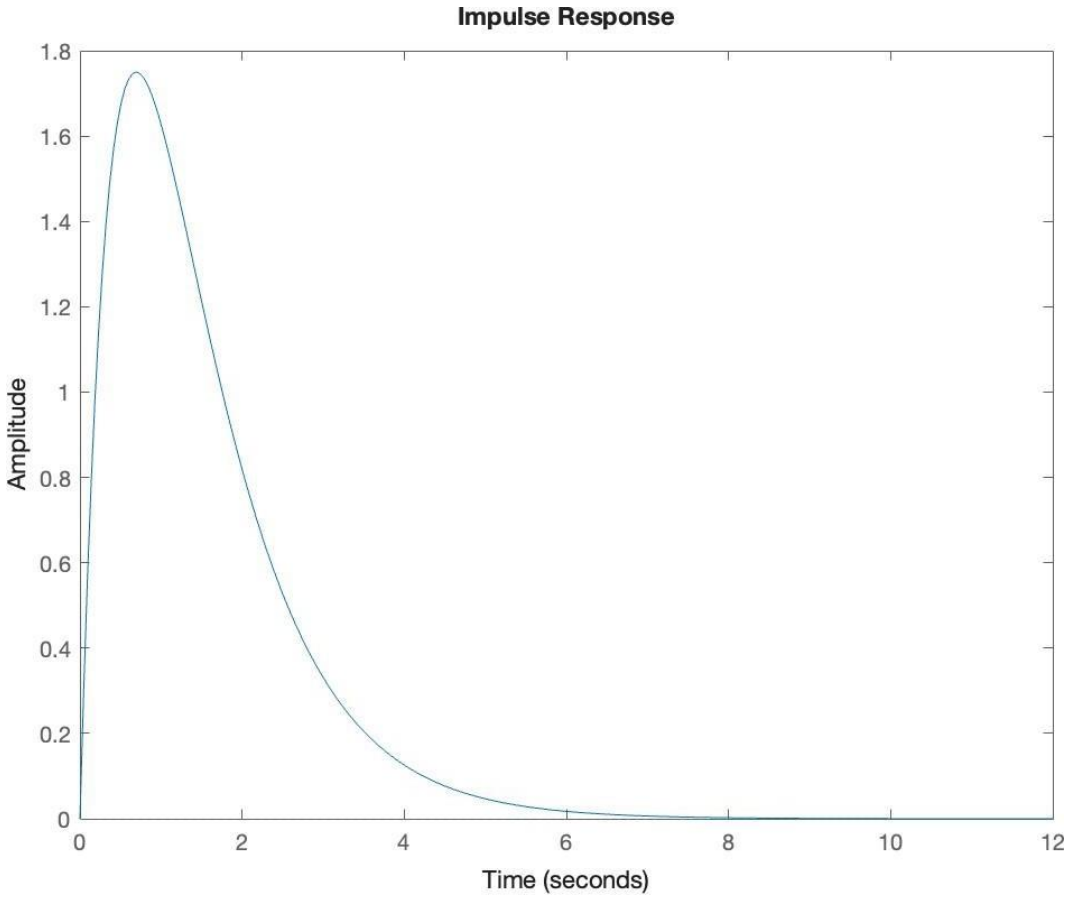
(a) Poles are the values of ss that make the denominator zero.

For s2+3s+2=0 we can factor the quadratic equation: s2+3s+2=(s+1) (s+2) =0.

This gives us poles at s=−1 and s=−2

(b) Code

```
% Define transfer function
H = 7 / (s^2 + 3*s + 2);

% Impulse response
figure; impulse(H);
title('Impulse Response');
```



(c) Given the poles $s=-1$ and $s=-2$, both poles are in the left half of the complex plane, which means the system step response will settle to a steady-state value over time.

This aligns with stability, so the system is stable, as observed in the **bounded and convergent nature** of the step 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.

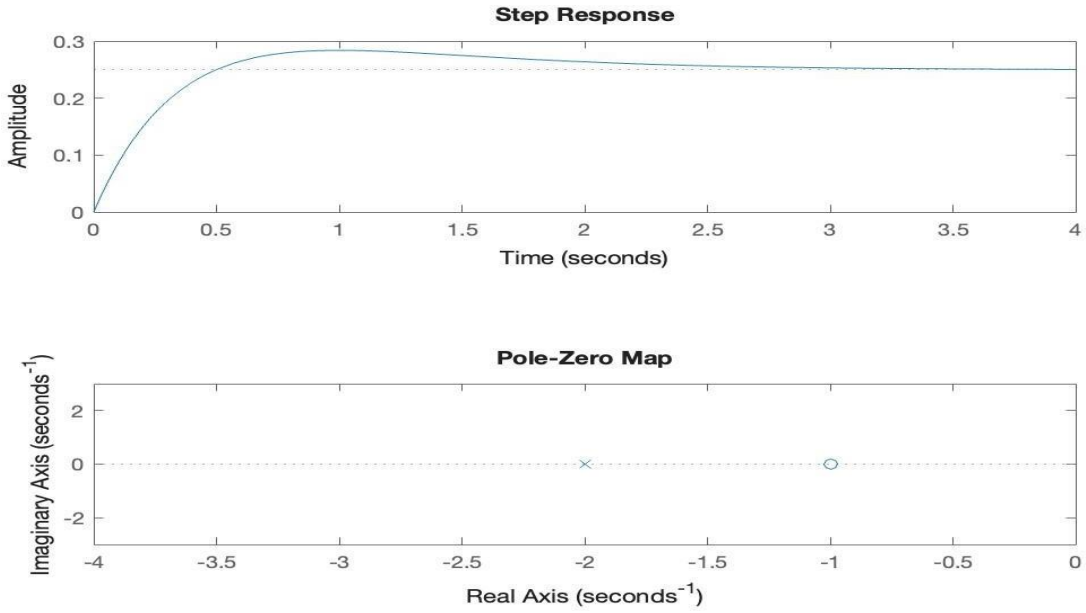
(a) For s+1=0 the zero is at s=−1.

For s2+4s+4=0 factor the quadratic equation: s2+4s+4=(s+2) (s+2) =0.

This gives us a **repeated pole** at s=−2

(b) **Code**

```
% 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
```



(c) The **zero** is at s=−1 and the **poles** are both at s=−2.

Since all poles are on the left half-plane, the system is **stable**, but the repeated pole at s=−2 could lead to a slower decay in the step response.