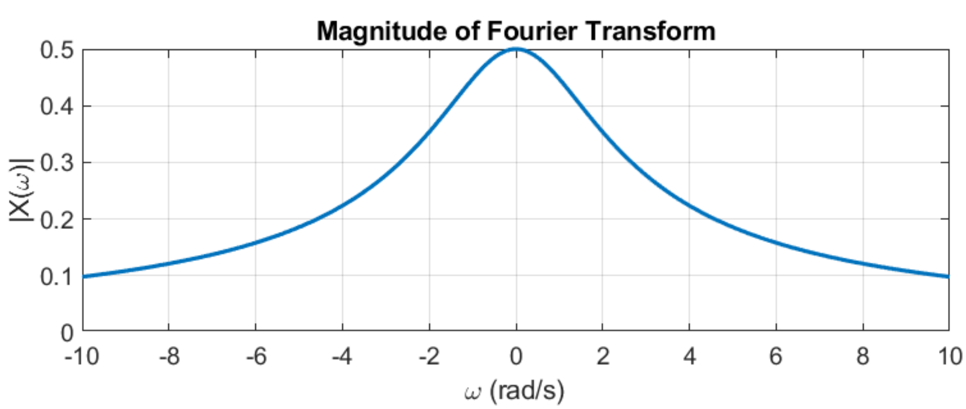
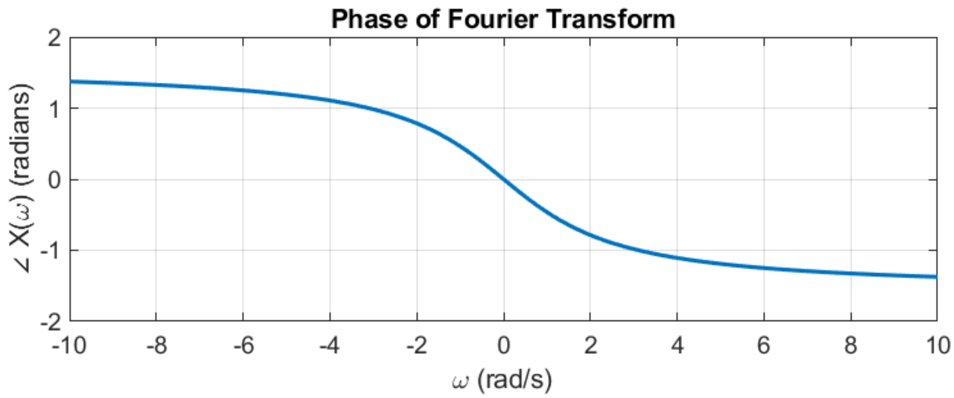
PROBLEM 2

MATLAB CODE

```
syms t w
u(t) = heaviside(t);    % defiined my unit step function.
x(t) = exp(-2*t)*u(t);  % our given function.

X = fourier(x(t),t,w);  % fourier transform of our given function.
disp(X);                % display of our fourier transform function.
X_mag = matlabFunction(abs(X));     % magnitude of our fourier transform.
X_phase = matlabFunction(angle(X)); % Phase of our fourier transform

w_vals = linspace(-10, 10, 1000);   %range of frequency for plotting.

X_mag_vals = X_mag(w_vals);         % evaluate magnitude of our function
X_phase_vals = X_phase(w_vals);     % evaluate phase of our function

% below code for plotting of magnitude and phase function.
subplot(2, 1, 1);
plot(w_vals, X_mag_vals, 'LineWidth', 1.5);
title('Magnitude of Fourier Transform');
xlabel('\omega (rad/s)');
ylabel('|X(\omega)|');
grid on;
```

MAGNITUDE AND PHASE PLOT



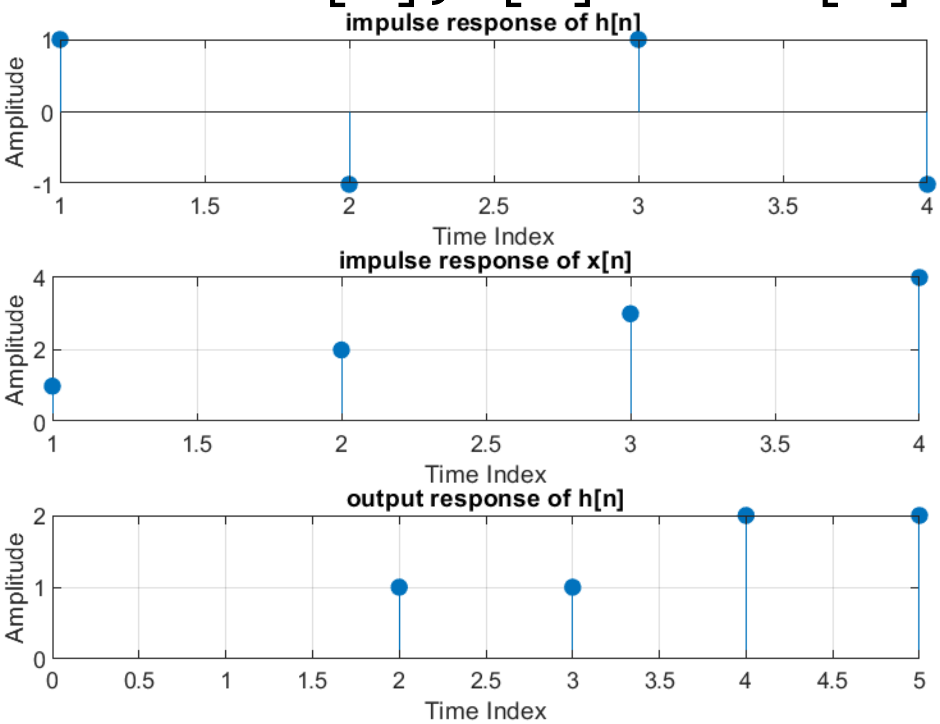


PROBLEM 3

MATLAB CODE

```
h = [1,-1,1,-1]; % System h[n]

nh = (1:length(h)); % no.of points in this system
x = [1,2,3,4];      % Input response x[n]

nx = ([1:length(x)]);
xlim([nh(1)-1 nh(end)+1]); % Time indices for h
xlim([nx(1)-1 nx(end)+1]); % Time indices for x
y = conv(x,h);             % convolution of x[n] and h[n]
ny = (nx(1) + nh(1) : nx(end) + nh(end)); %Time indices for output signal
figure,
% Plot for h[n]
subplot(3,1,1);
stem(nh,h,'filled');
grid on;
xlabel('Time Index');
ylabel('Amplitude');
title('impulse response of h[n]');
%Plot for x[n]
subplot(3,1,2);
stem(nx,x,'filled');
grid on;
xlabel('Time Index');
ylabel('Amplitude');
title('impulse response of x[n]');
%Plot for output response
subplot(3,1,3);
stem(ny,y,'filled');
grid on;
xlabel('Time Index');
ylabel('Amplitude');
title('output response of h[n]');
```

PLOT OF H[N],X[N] AND Y[N]

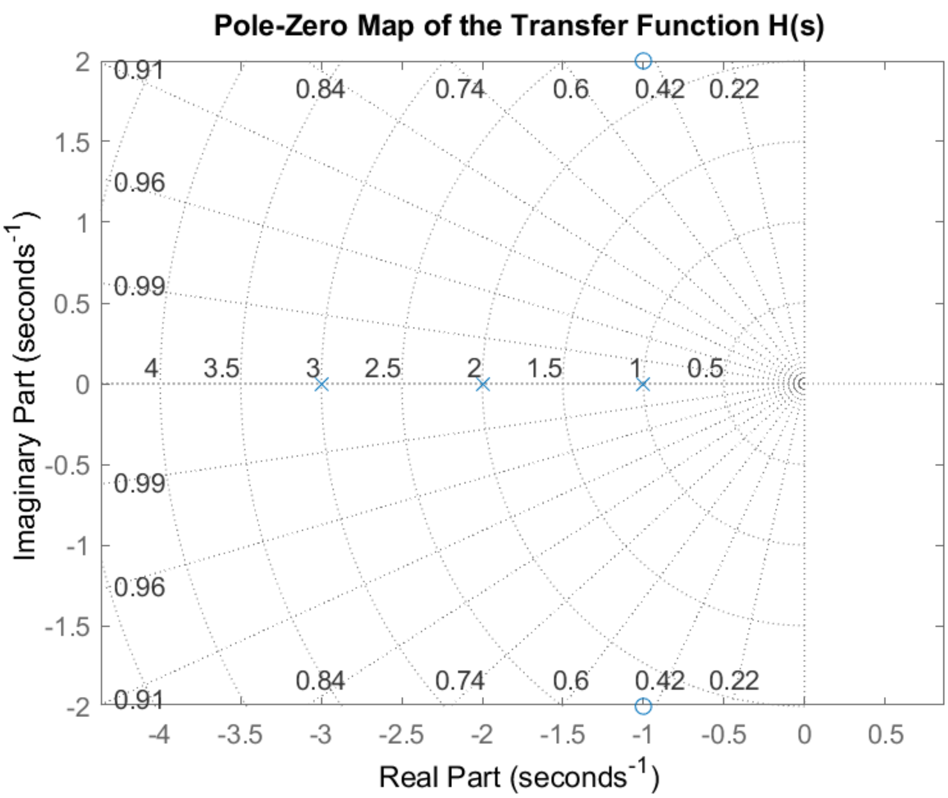


PROBLEM 4

MATLAB CODE

```
fs = 2000;           % Define Sampling frequency
ts = 1/fs;           % define time step
num1 = [1 2 5];      % Numerator coefficients of transfer function
den1 = [1 6 11 6];   % Denominator coefficients of tranfer function
H = tf(num1,den1);   %Create the transfer function H(s)
pzmap(H);            % Pole zero map of given tranfer function
% Plot of pole zero map
axis equal;
sgrid;
title('Pole-Zero Map of the Transfer Function H(s)');
xlabel('Real Part');
ylabel('Imaginary Part');
grid on;
```

POLE-ZERO PLOT OF TRANSFER FUNCTION



And by clearly looking at the PZ map we can see that all poles lies on open left half plane so our given transfer function is stable.

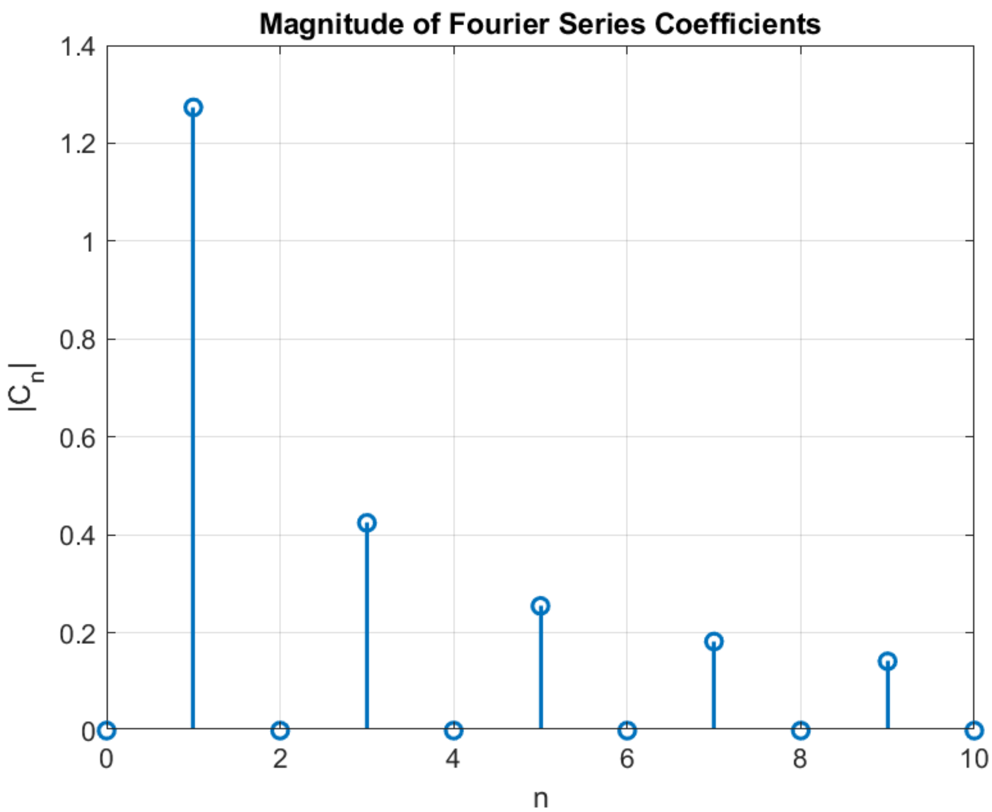
PROBLEM 5

MATLAB CODE

```
syms t
T = 2 * pi;
wo = 2 * pi / T;
x = piecewise(0 <= t & t < pi, -1, pi <= t & t < 2*pi, 1); % Define the piecewise
function x(t)
ao = (1/T) * int(x, t, 0, T);          % Compute the DC component a0

N = 10; % Number of Fourier terms
an = sym(zeros(1, N));
bn = sym(zeros(1, N));

% Computeing an and bn
for n = 1:N
    an(n) = (2/T) * int(x * cos(n * wo * t), t, 0, T);
```

```
    bn(n) = (2/T) * int(x * sin(n * wo * t), t, 0, T);
end

% Convert symbolic coefficients to double for numerical use
ao = double(ao);
an = double(an);
bn = double(bn);

% Display the Fourier series coefficients
disp('ao = ');
disp(ao);
disp('an = ');
disp(an);
disp('bn = ');
disp(bn);

% Compute the magnitudes of the coefficients
magnitude  = sqrt(an.^2 + bn.^2);

% Plot the magnitudes of the Fourier coefficients
figure;
stem(0:N, [ao, magnitude], 'LineWidth', 1.5);
title('Magnitude of Fourier Series Coefficients');
xlabel('n');
ylabel('|C_n|');
grid on;

% Reconstruct the signal using the first N terms
t_val = linspace(0, T, 1000);
x_reconstructed = ao * ones(size(t_val));

for k = 1:N
    x_reconstructed = x_reconstructed + an(k) * cos(k * wo * t_val) + bn(k) * sin(k *
wo * t_val);
end

% Define the original piecewise function numerically
x_original = zeros(size(t_val));
for i = 1:length(t_val)
    if 0 <= t_val(i) && t_val(i) < pi
        x_original(i) = -1;
    elseif pi <= t_val(i) && t_val(i) < 2*pi
        x_original(i) = 1;
    end
end

% Plotting the original and reconstructed signals
figure;
plot(t_val, x_original, 'r', 'LineWidth', 1.5); hold on;
plot(t_val, x_reconstructed, 'b--', 'LineWidth', 1.5);
legend('Original Signal', 'Reconstructed Signal');
title('Fourier Series Reconstruction of Square Wave');
xlabel('Time (t)');
ylabel('x(t)');
grid on;
```

PLOT OF FOURIER SERIES COEFFICIENT



PLOT OF ORIGINAL AND RECONSTRUCTUED SIGNAL