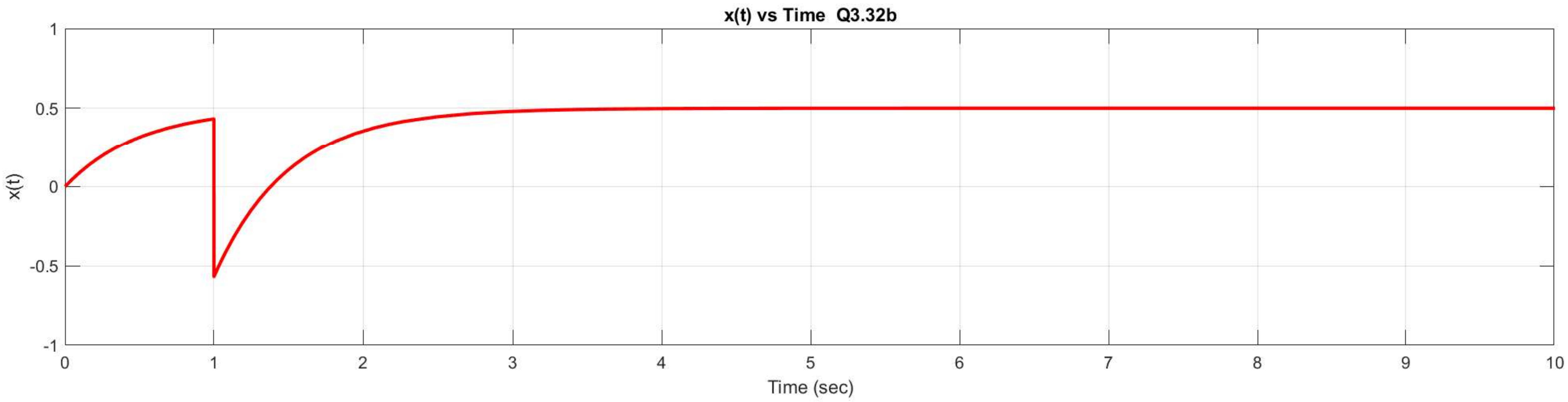
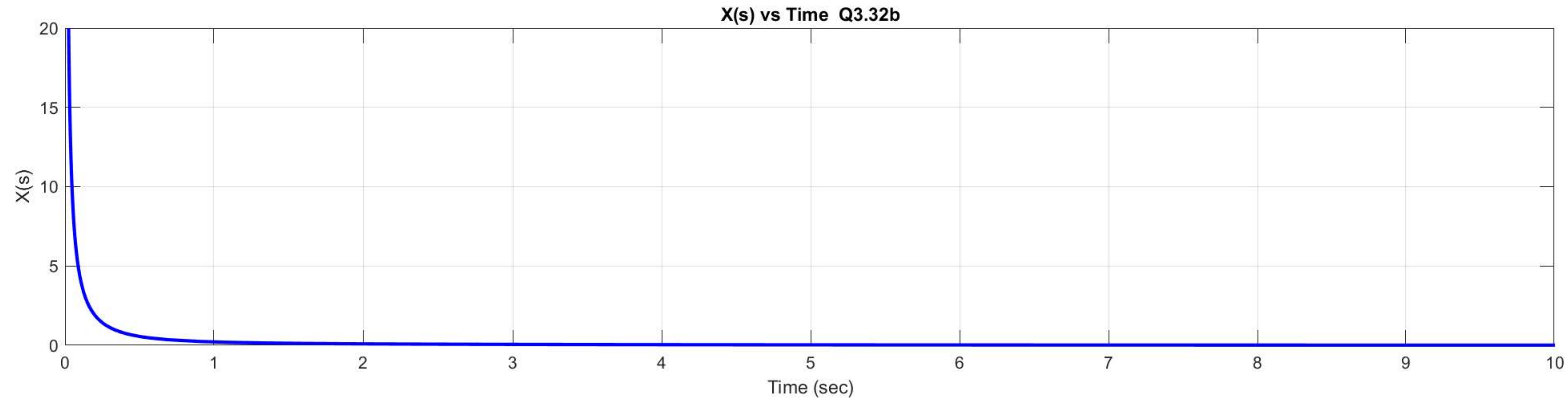
```
% Name: Lamin Jammeh
% Class: EE480 Online
% Semster: Fall 2023
% HW_6

% Basic Problems
%% ********** 3.23a **********
clear;
clc;
% Step 1 define a syms function with t and s
syms t s;
% defien the impulse response of the LTI system
h_t = exp(-2*t) * heaviside(t);
% define the input of the system;
x_t = heaviside(t) - heaviside(t-3);
% Step 2 change both the H(t) and x(t) to laplace transform
H_s = laplace(h_t);
X_s = laplace(x_t);
%define Y(s)
Y1_s = H_s * X_s;
Y_simplified = simplifyFraction(Y1_s);
% Step 3 change Y(s) to y(t) to get the output in time domain
y_t = ilaplace(Y1_s);

%% ********** 3.22a **********
clear;
clc;
%define syms interms of t and s;
syms t s;
% define x(t)
x_t = heaviside(t) - 2*heaviside(t-1) + heaviside(t-2);
% define X(s) as laplace transform of x(t)
X_s = laplace(x_t);
%define Y(s)
Y1_s = ((s+2)*(1-exp(-s))^2) / (s*(s+1)^2);
%define H(s) as impulse response in laplace domain
H_s = Y1_s/X_s;
H_simplified = simplifyFraction(H_s);

%% ********** MATLAB Problems 3.32a **********
%Find the inverse laplace transform of X(s)
% define the syms as t and s
clear;
clc;
syms t s;
%define X(s)
X_s = (s^2 + 2*s + 1) / (s*(s+1)*(s^2 + 10*s + 50));
x_t = ilaplace(X_s);
x_simplified = simplifyFraction(x_t);
```

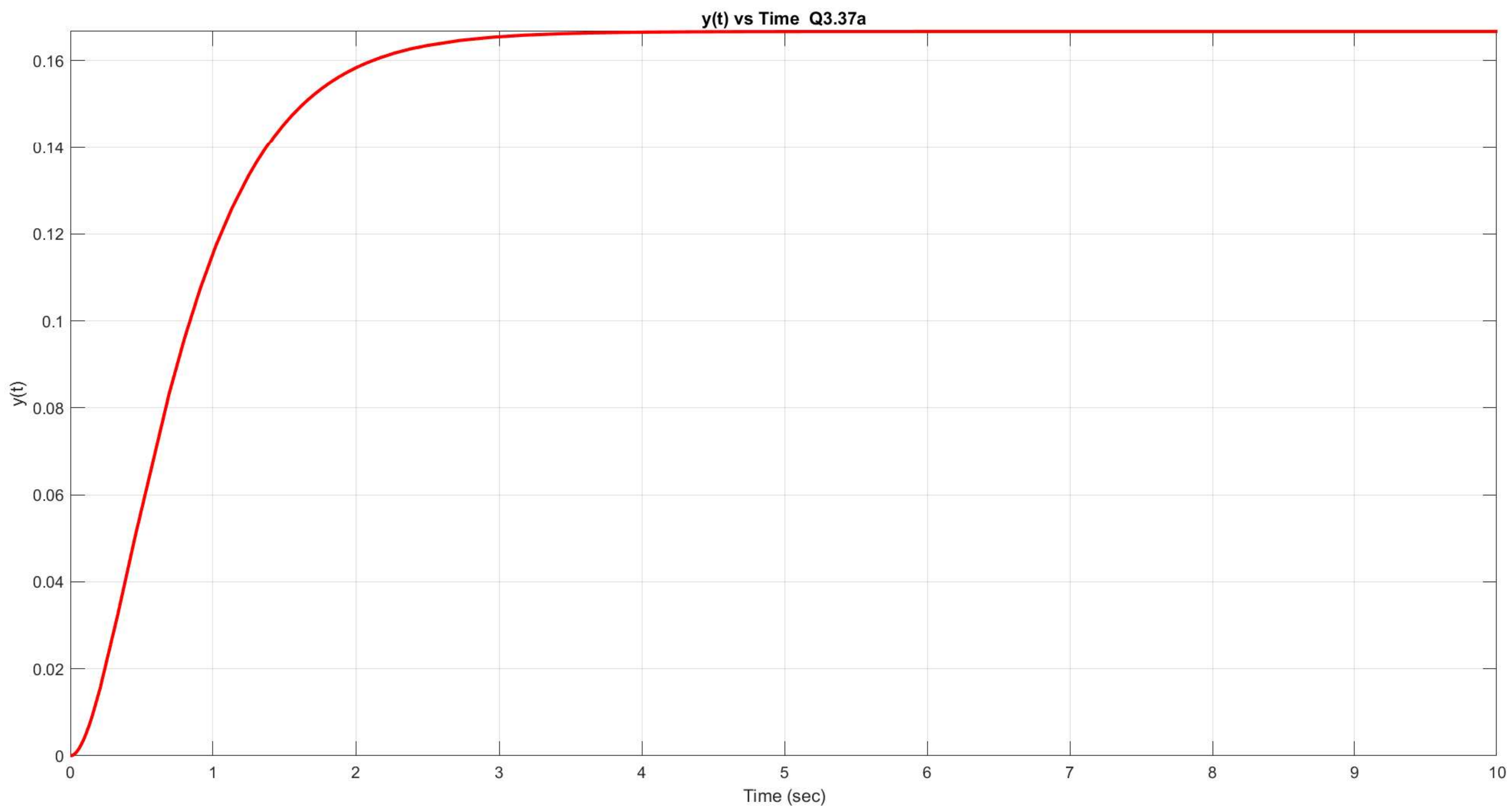
```matlab
%% ********** MATLAB Problems 3.32b **********
%Find the inverse laplace transform of X(s)
clear;
clc;
% define the syms as t and s
syms t s;
X_s = (1-s*exp(-s)) / (s*(s+2)); % define X(s)
x_t = ilaplace(X_s); % define x(t)
x_simplified = simplifyFraction(x_t);
subplot(2,1,1)
fplot(x_simplified,[0,10],'r',"LineWidth",2); % Plot x(t) as a function plot
xlabel('Time (sec)');
ylabel("x(t)");
title("x(t) vs Time  Q3.32b");
ylim([-1,1]) % assign limits to y axis
grid on;
subplot(2,1,2)
fplot(X_s,[0,10],'b',"LineWidth",2); %Plot X(s) as a functi0n plot
xlabel('Time (sec)');
ylabel("X(s)");
title("X(s) vs Time  Q3.32b");
ylim([0,20]) % assign limits to y axis
grid on;

%% ********** MATLAB Problems 3.35 **********
clear;
clc;
% determine h(t) from Y(s)
%Step 1 determine the numerator & denuminator of the H(s)
num = [1,2]; %defines the numerator
denum = [1,2,1]; %define the denuminator
%Step 2 define the transform function of the impulse response H(s)
H_s = tf(num, denum); % use the tf function for the H(s)
%Step 3 use the pole function to determine the poles
poles = pole(H_s);
%condition for BIBO stable
if pole(H_s) < 0
    disp('The system is BIBO stable')
else
    disp('The system is not BIBO stable')
end

%% ********** MATLAB Problems 3.37a **********
clear;
clc;
b1 = 5;
b0 = 6;
%Define syms function;
```

```
syms t s;
%Define the input of the system
x_t = heaviside(t);
X_s = laplace(x_t);
%Define H(s)
H_s = 1 / (s^2 + b1*s + b0);
%Define Y(s) = X(s)*H(s)
Y1_s = H_s * X_s;
y_t = ilaplace(Y1_s);
%plot y(t) using a fplot
fplot(y_t,[0,10],'r',"LineWidth",2); % Plot y(t) as a function plot
xlabel('Time (sec)');
ylabel("y(t)");
title("y(t) vs Time  Q3.37a");
grid on;

%% ********** MATLAB Problems 3.39b **********
clear;
clc;
%define syms as t and s
syms t s;
%define input of the system x(t)
x_t = heaviside(t);
X_s = laplace(x_t);

Y1_s = (s+1) / (s*(s^2 + 2*s + 4));
y1_t = ilaplace(Y1_s);
%define the transfer function of the system
H1_s = simplifyFraction(Y1_s / X_s)
% plot the poles and zeros
num1 = [1,1];
denum1 = [1,2,4];
H1_s = tf(num1,denum1);
figure
pzmap(H1_s)
title("pzmap of H1(s)");
grid on;
% define x and y axis limits
xlim([-2,0]) % assign limits to y axis
ylim([-2,2]) % assign limits to y axis


Y2_s = 1 / (s+2)^2;
y2_t = ilaplace(Y2_s);
H2_s = simplifyFraction(Y2_s / X_s)
% plot the poles and zeros
num1 = [1];
denum1 = [1,2,4];
H2_s = tf(num1,denum1);
```

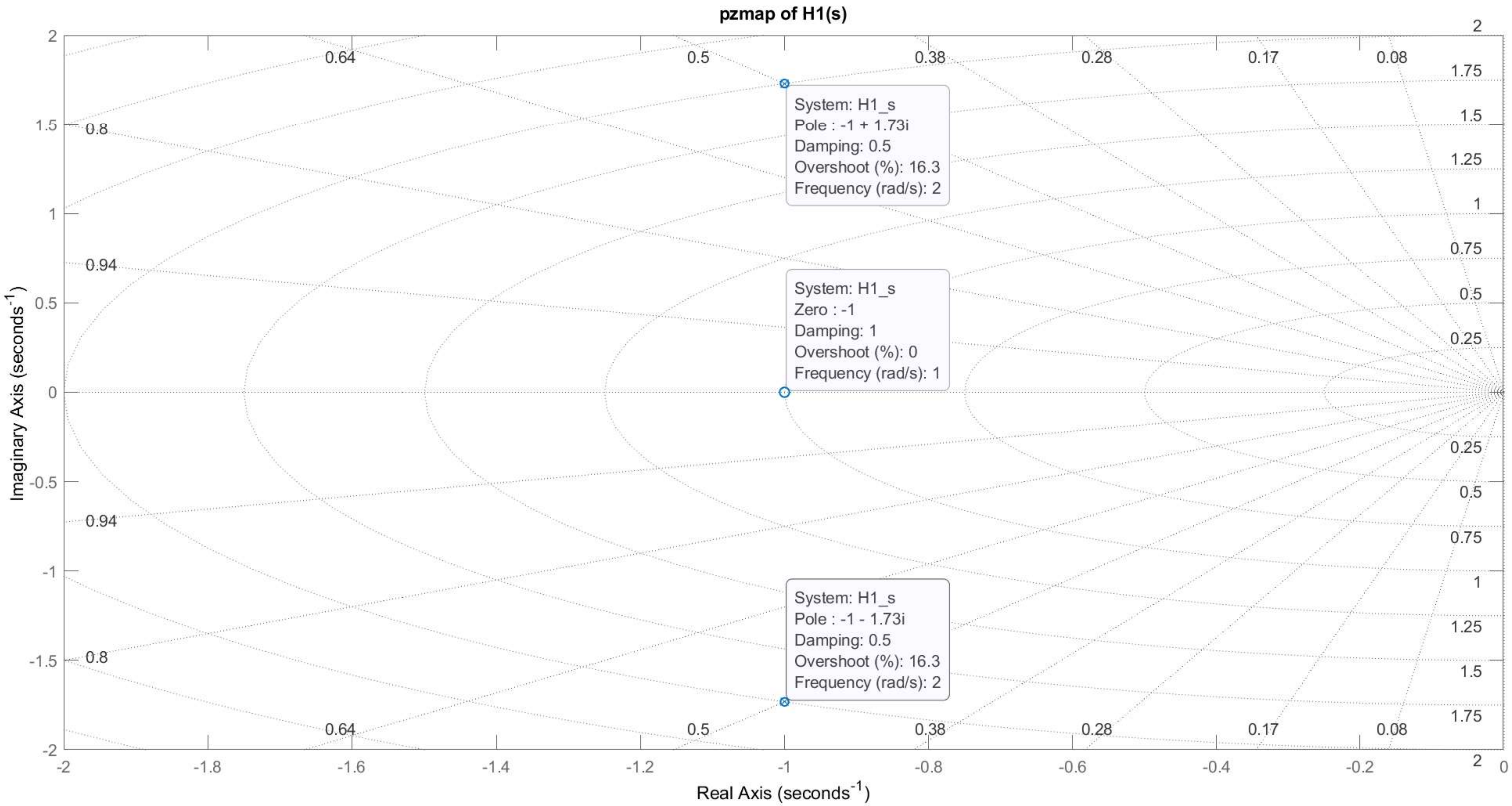
```
figure
pzmap(H2_s)
title("pzmap of H2(s)");
grid on;
% define x and y axis limits
xlim([-2,2]) % assign limits to y axis
ylim([-2,2]) % assign limits to y axis


Y3_s = (s-1) / (s^2*((s+1)^2 + 9));
y3_t = ilaplace(Y3_s);
H3_s = simplifyFraction(Y3_s / X_s)
% plot the poles and zeros
num1 = [1,-1];
denum1 = [1,2,10,0];
H3_s = tf(num1,denum1);
figure
pzmap(H3_s)
title("pzmap of H3(s)");
grid on;
% define x and y axis limits
xlim([-2,2]) % assign limits to y axis
ylim([-2,2]) % assign limits to y axis
```

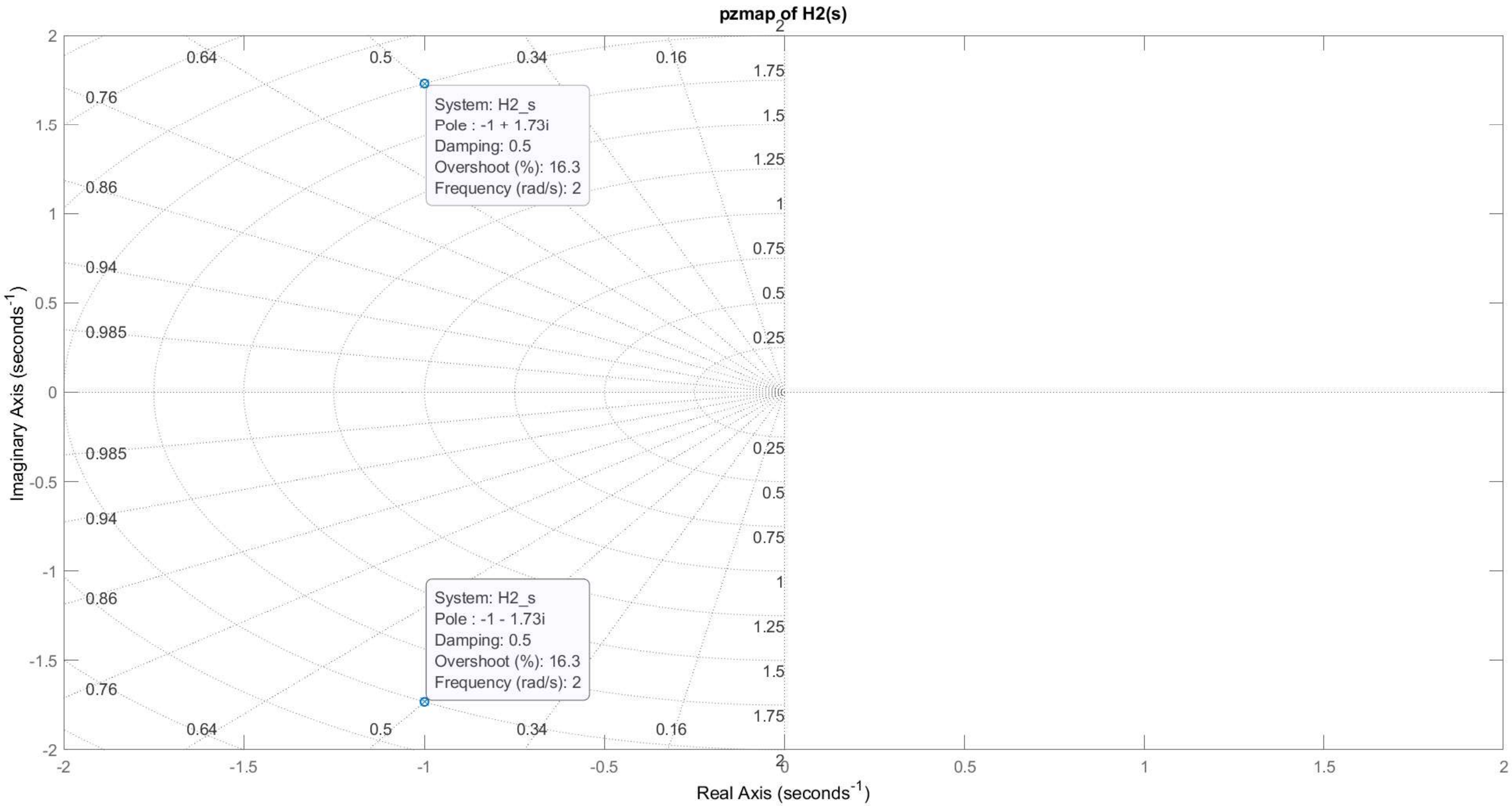

x(t) vs Time Q3.32b
x(t)
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
7
8
9
10
Time (sec)

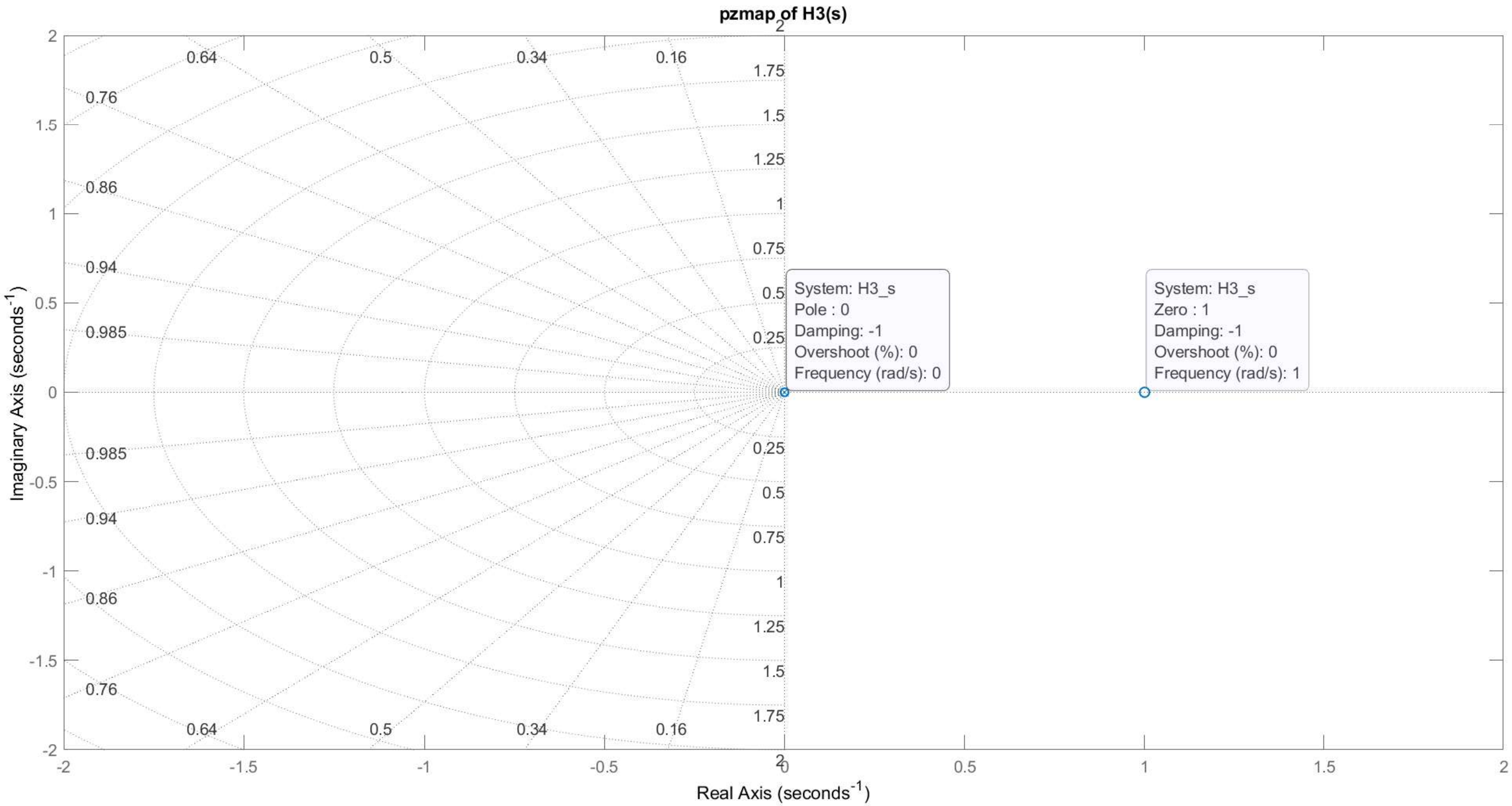

X(s) vs Time Q3.32b
X(s)
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
10
Time (sec)



y(t) vs Time Q3.37a
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
y(t)
0
1
2
3
4
5
6
7
8
9
10
Time (sec)


pzmap of H1(s)
System: H1_s
Pole : -1 + 1.73i
Damping: 0.5
Overshoot (%): 16.3
Frequency (rad/s): 2
System: H1_s
Zero : -1
Damping: 1
Overshoot (%): 0
Frequency (rad/s): 1
System: H1_s
Pole : -1 - 1.73i
Damping: 0.5
Overshoot (%): 16.3
Frequency (rad/s): 2
Real Axis (seconds^-1)
Imaginary Axis (seconds^-1)
0.64
0.5
0.38
0.28
0.17
0.08
0.8
0.94
2
1.75
1.5
1.25
1
0.75
0.5
0.25
-2
-1.8
-1.6
-1.4
-1.2
-1
-0.8
-0.6
-0.4
-0.2
0
1.5
0.5
-0.5
-1
-1.5
-2


pzmap of H2(s)
System: H2_s
Pole : -1 + 1.73i
Damping: 0.5
Overshoot (%): 16.3
Frequency (rad/s): 2
System: H2_s
Pole : -1 - 1.73i
Damping: 0.5
Overshoot (%): 16.3
Frequency (rad/s): 2
Real Axis (seconds^{-1})
Imaginary Axis (seconds^{-1})


pzmap of H3(s)
System: H3_s
Pole : 0
Damping: -1
Overshoot (%): 0
Frequency (rad/s): 0
System: H3_s
Zero : 1
Damping: -1
Overshoot (%): 0
Frequency (rad/s): 1
Real Axis (seconds^-1)
Imaginary Axis (seconds^-1)

3.23a

Monday, October 2, 2023 8:38 PM

$h(t) = e^{-2t}u(t)$ $\qquad$ $x(t) = u(t) - u(t-3)$

$\mathcal{L}\{x(t) * h(t)\} = X(s) \cdot H(s)$

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\mathcal{L}\, y(t)}{\mathcal{L}\, x(t)}$$

$$Y(s) = X(s) \cdot H(s)$$

Check Matlab Code for the rest

$$\boxed{\frac{[e^{-3s} \cdot [e^{3s} - 1]]}{s(s+2)}} = \frac{e^{-3s+3s} - e^{-3s}}{s(s+2)} = \boxed{\frac{1 - e^{-3s}}{s(s+2)}}$$

Matlab answer = text Book's answer

3.20a

Monday, October 2, 2023 9:23 PM

$$y(t) = \alpha x(t-T) + \alpha^3 x(t-3T)$$

Input is $x(t)$.

$$\mathcal{L}\, h(t) = \frac{\mathcal{L}\, y(t)}{\mathcal{L}\, X(t)} = \frac{Y(s)}{X(s)}$$

$$\mathcal{L}\, x(t) = X(s)$$

$$\mathcal{L}\, y(t) = \left[\alpha e^{Ts} + \alpha^3 e^{3Ts}\right] X(s) = Y(s)$$

$$\frac{Y(s)}{X(s)} = \alpha\left[e^{Ts} + \alpha^2 e^{3Ts}\right]$$

$$H(s) = \alpha\left[e^{Ts} + \alpha^2 e^{3Ts}\right]$$

$$h(t) = \alpha\left[\delta(t-T) + \alpha^2 \delta(t-3T)\right]$$

b) $H(s) = \alpha\left[e^{Ts} + \alpha^2 e^{3Ts}\right]$

The system has no poles

The system is BIBO Stable

3.35c

Tuesday, October 3, 2023 9:55 PM

$$Y(s) = \frac{X(s)}{s^2+2s+3} + \frac{s+1}{s^2+2s+3}$$

$$(s^2+2s+3)\,Y(s) = X(s) + (s+1)$$

$$\frac{(s+2)\cancel{(s+1)}}{\cancel{(s+1)}}\,Y(s) = \frac{X(s)}{(s+1)} + \frac{\cancel{(s+1)}}{\cancel{(s+1)}}$$

$$(s+2)\,Y(s) = \frac{X(s)}{(s+1)} + 1$$

$$(s+2)(s+1)\,Y(s) - (s+1)$$

$$[s^2+2s+3]\,Y(s) - (s+1) = X(s)$$

$$s^n F(s) = \frac{d^n}{dt^n} f(t)$$

$$\frac{d^2}{dt^2} y(t) + 2\frac{d}{dt} y(t) + 3y(t) - \text{Initial Condition} = x(t)$$

$$x(t) = \frac{d^2}{dt^2} y(t) + 2\frac{d}{dt} y(t) + 3y(t)$$