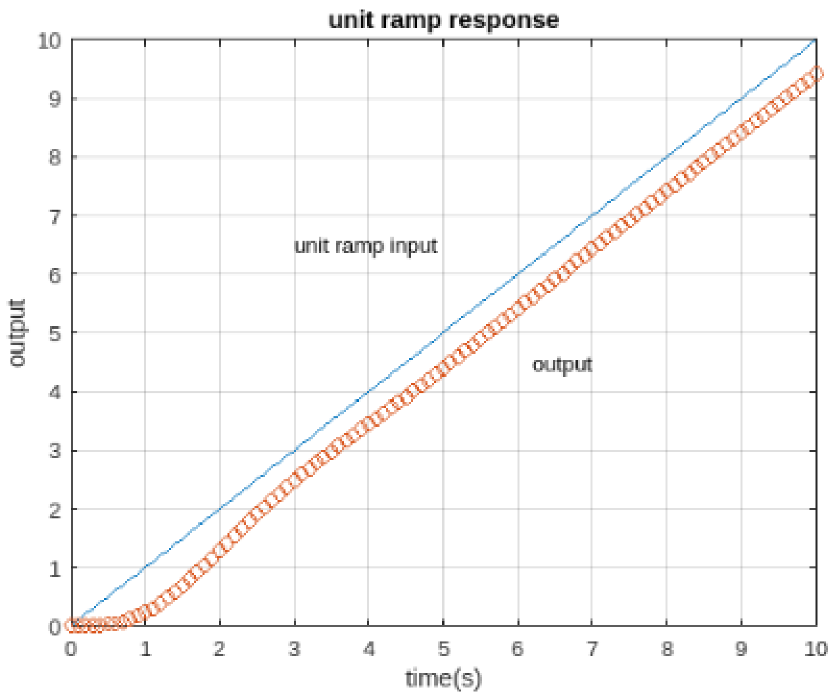
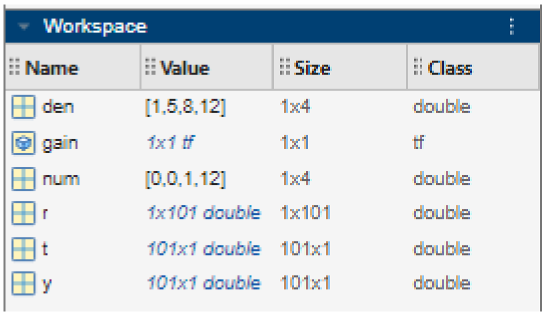
1. Find out the response for the transfer function TF = $\frac{C(s)}{R(s)} = \frac{s+12}{s^3+5s^2+8s+12}$ for unit ramp input.

<u>Code:</u>

```
num=[0 0 1 12]; % Defining the numerator polynomial of the transfer function
den=[1 5 8 12]; % Defining the denominator polynomial of the transfer function
t=0:0.1:10;% Creating a time vector from 0 to 10 seconds with a step size of 0.1
r=t; % Creating a unit ramp input signal
gain=tf(num,den); % Creating the transfer function object
[y,t]=lsim(gain,t,r); % Simulating the system response to the unit ramp input
plot(t, r, '-', t, y, 'o'); % Plotting the unit ramp input and its output
title('unit ramp response'); % Adding title to the plot
xlabel('time(s)'); % Adding x-label to the plot
ylabel('output'); % Adding y-label to the plot
text(3.0,6.5,'unit ramp input'); % Adding text annotations to indicate the input
text(6.2,4.5,'output'); % Adding text annotations to indicate the output signal
grid on; % Turning on the grid for better readability
```

<u>Output:</u>





Workspace

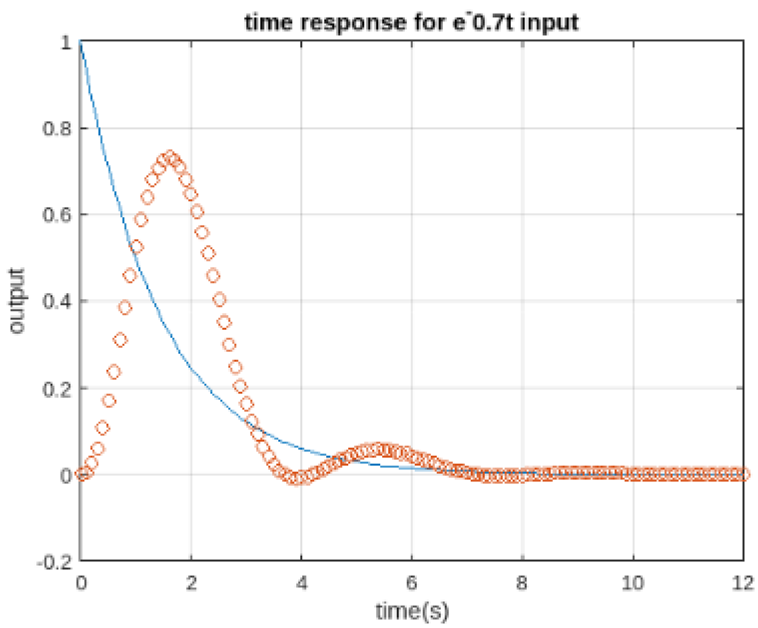
Name	Value	Size	Class
den	[1,5,8,12]	1x4	double
gain	1x1 tf	1x1	tf
num	[0,0,1,12]	1x4	double
r	1x101 double	1x101	double
t	101x1 double	101x1	double
y	101x1 double	101x1	double

2. **Find out the response for the transfer function TF = $\frac{C(s)}{R(s)} = \frac{s+12}{s^3+5s^2+8s+12}$ for exponential input.**

<u>Code:</u>

```
num=[0 0 1 12]; % Defining the numerator polynomial of the transfer function
den=[1 5 8 12]; % Defining the denominator polynomial of the transfer function
t=0:0.1:12;% Creating a time vector from 0 to 12 seconds with a step size of 0.1
r=exp(-0.7*t);; % Creating a exponential input signal
gain=tf(num,den); % Creating the transfer function object
[y,t]=lsim(gain,t,r); % Simulating the system response to the exponential input
plot(t, r, '-', t, y, 'o'); % Plotting the exponential input and its output
title('time response for e^-0.7t input'); % Adding title to the plot
xlabel('time(s)'); % Adding x-label to the plot
ylabel('output'); % Adding y-label to the plot
grid on; % Turning on the grid for better readability
```

<u>Output:</u>



Workspace

Name	Value	Size	Class
den	[1,5,8,12]	1x4	double
gain	1x1 tf	1x1	tf
num	[0,0,1,12]	1x4	double
r	1x121 double	1x121	double
t	121x1 double	121x1	double
y	121x1 double	121x1	double

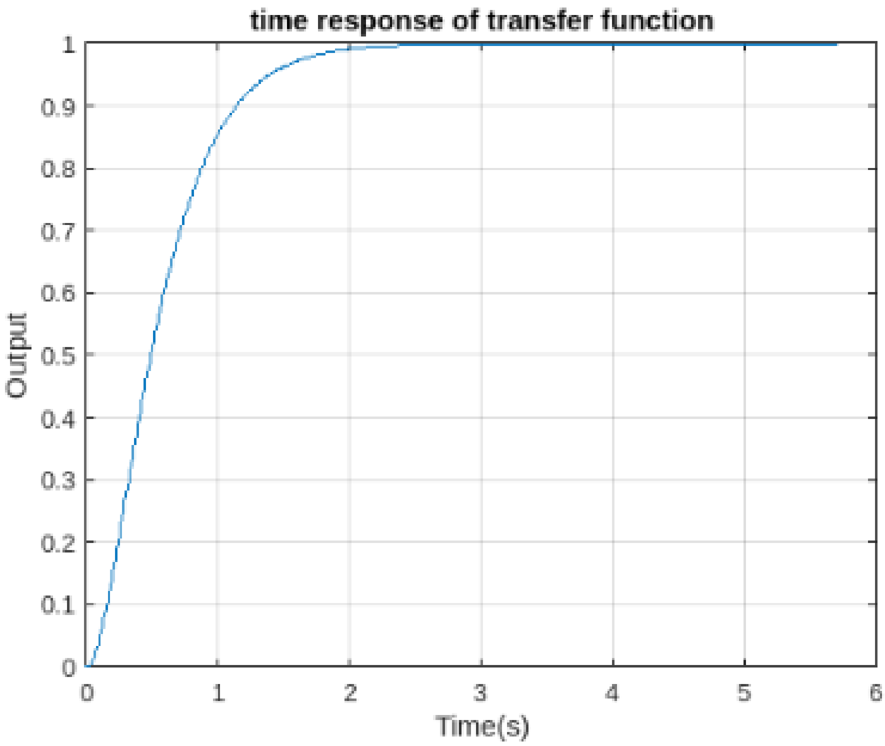
3. Find out the response and time parameters for the transfer function $TF_1 = \frac{C(s)}{R(s)} = \frac{12}{s^2+7s+12}$ for step input using MATLAB and Simulink.

MATLAB Code:

```
% Defining system parameters
R=2.8; % Resistance (ohms)
L=400*10^-3; % Inductance (henries)
C=0.2083;  % Capacitance (farads)
% Calculating natural frequency (wn) and damping ratio (zeta)
wn=1/(sqrt(L*C));
zetaa=(R/2)*(sqrt(C/L));
% Calculating settling time (ts), peak time (tp), rise time (tr), and
percent overshoot (mp)
ts=4/(zetaa*wn);
tp=pi/(wn*sqrt(zetaa^2-1));
x=atan(sqrt(zetaa^2-1)/zetaa);
y=deg2rad(x);
tr=(pi-y)/(wn*(sqrt(zetaa^2-1)));
mp=exp((-pi*zetaa/sqrt(zetaa^2-1)))*100;
gain=tf([12],[1 7 12]); % Defining transfer function
t=0:0.01:ts*5; % Generating time vector
% Calculating and plot step response
[y,t]=step(gain,t);
plot(t,y);
% Adding plot title, x-label, y-label and grid
title('time response of transfer function');
xlabel('Time(s)');
ylabel('Output');
grid on; % Turning on the grid for better readability
% Displaying calculated parameters
disp('Calculated Parameters:');
disp(['Natural Frequency (wn): ', num2str(wn)]);
disp(['Damping Ratio (zetaa): ', num2str(zetaa)]);
disp(['Settling Time (ts): ', num2str(ts)]);
disp(['Peak Time (tp): ', num2str(tp)]);
disp(['Rise Time (tr): ', num2str(tr)]);
disp(['Percent Overshoot (mp): ', num2str(mp)]);
```

Output:

```
Calculated Parameters:
Natural Frequency (wn): 3.4644
Damping Ratio (zetaa): 1.0103
Settling Time (ts): 1.1429
Peak Time (tp): 6.3075
Rise Time (tr): 6.3025
Percent Overshoot (mp): 2.5851e-08
```

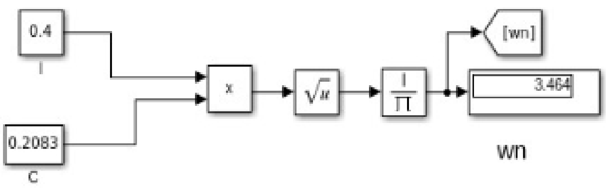


Workspace

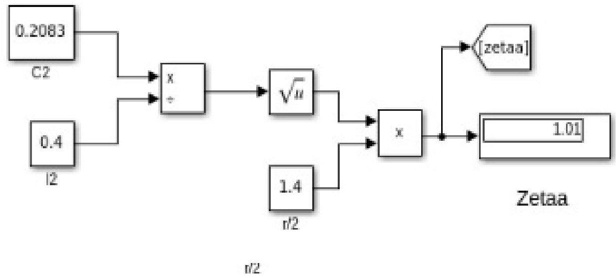
Name	Value	Size	Class
C	0.2083	1x1	double
L	0.4000	1x1	double
R	2.8000	1x1	double
den	[1,5,8,12]	1x4	double
gain	*1x1 tf*	1x1	tf
mp	2.5851e-08	1x1	double
num	[0,0,1,12]	1x4	double
r	*1x121 double*	1x121	double
t	*229x1 double*	229x1	double
tp	6.3075	1x1	double
tr	6.3025	1x1	double
ts	1.1429	1x1	double
wn	3.4644	1x1	double
x	0.1414	1x1	double
y	*229x1 double*	229x1	double
zetaa	1.0103	1x1	double

Simulink:

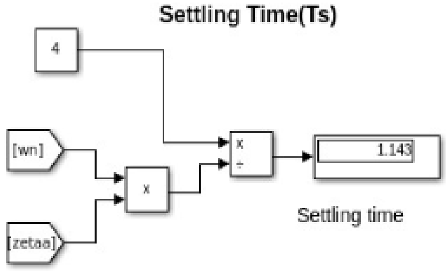
Natural Frequency(wn)



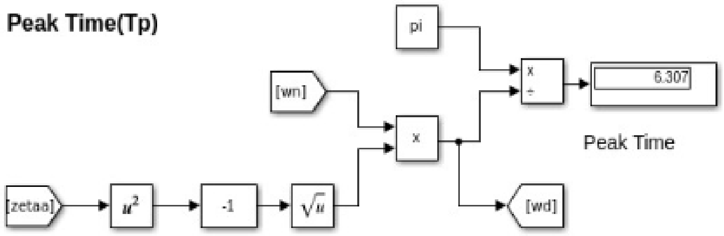
Damping Ratio(zetaa)



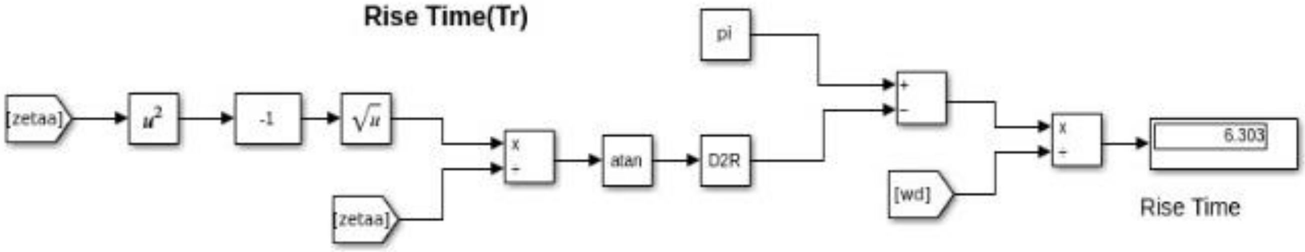
Settling Time(Ts)



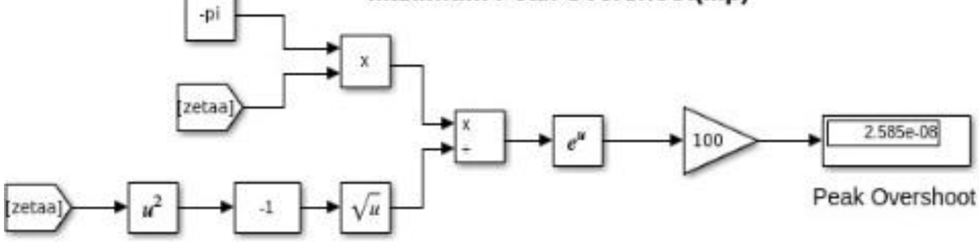
Peak Time(Tp)



Rise Time(Tr)



Maximum Peak Overshoot(Mp)



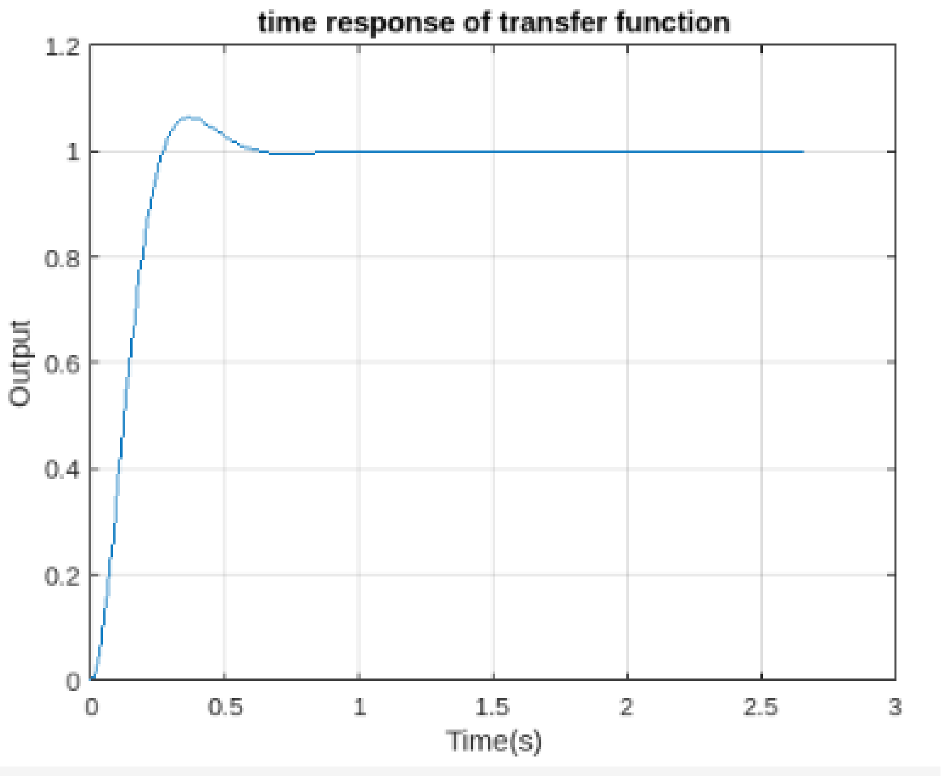
4. Find out the response and time parameters for the transfer function $TF_2 = \frac{C(s)}{R(s)} = \frac{130}{s^2+15s+130}$ for step input using MATLAB and Simulink.

<u>MATLAB Code:</u>

```
% Defining system parameters
R=6; % Resistance (ohms)
L=400*10^-3; % Inductance (henries)
C=0.0192;  % Capacitance (farads)
% Calculating natural frequency (wn) and damping ratio (zeta)
wn=1/(sqrt(L*C));
zetaa=(R/2)*(sqrt(C/L));
% Calculating settling time (ts), peak time (tp), rise time (tr), and
percent overshoot (mp)
ts=4/(zetaa*wn);
tp=pi/(wn*sqrt(1-zetaa^2));
x=atan(sqrt(1-zetaa^2)/zetaa);
y=deg2rad(x);
tr=(pi-y)/(wn*(sqrt(1-zetaa^2)));
mp=exp((-pi*zetaa/sqrt(1-zetaa^2)))*100;
gain=tf([130],[1 15 130]); % Defining transfer function
t=0:0.01:ts*2; % Generating time vector
% Calculating and plot step response
[y,t]=step(gain,t);
plot(t,y);
% Adding plot title, x-label, y-label and grid
title('time response of transfer function');
xlabel('Time(s)');
ylabel('Output');
grid on; % Turning on the grid for better readability
% Displaying calculated parameters
disp('Calculated Parameters:');
disp(['Natural Frequency (wn): ', num2str(wn)]);
disp(['Damping Ratio (zetaa): ', num2str(zetaa)]);
disp(['Settling Time (ts): ', num2str(ts)]);
disp(['Peak Time (tp): ', num2str(tp)]);
disp(['Rise Time (tr): ', num2str(tr)]);
disp(['Percent Overshoot (mp): ', num2str(mp)]);
```

<u>Output:</u>

Calculated Parameters:
Natural Frequency (wn): 11.4109
Damping Ratio (zetaa): 0.65727
Settling Time (ts): 0.53333
Peak Time (tp): 0.36531
Rise Time (tr): 0.36357
Percent Overshoot (mp): 6.4584

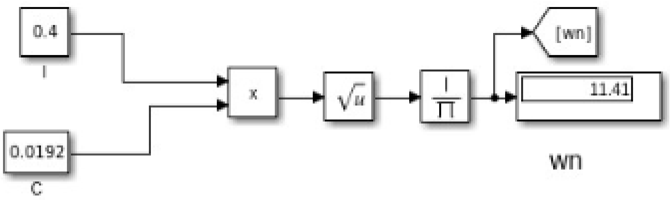


Workspace

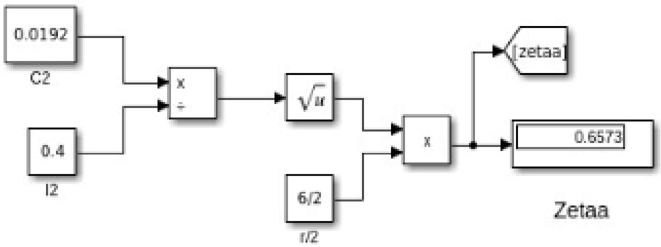
Name	Value	Size	Class
C	0.0192	1x1	double
L	0.4000	1x1	double
R	6	1x1	double
gain	*1x1 tf*	1x1	tf
mp	6.4584	1x1	double
t	*107x1 double*	107x1	double
tp	0.3653	1x1	double
tr	0.3636	1x1	double
ts	0.5333	1x1	double
wn	11.4109	1x1	double
x	0.8536	1x1	double
y	*107x1 double*	107x1	double
zetaa	0.6573	1x1	double

Simulink:

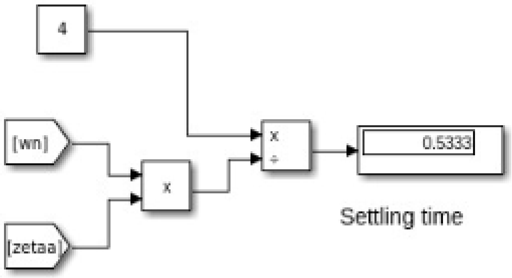
Natural Frequency(wn)



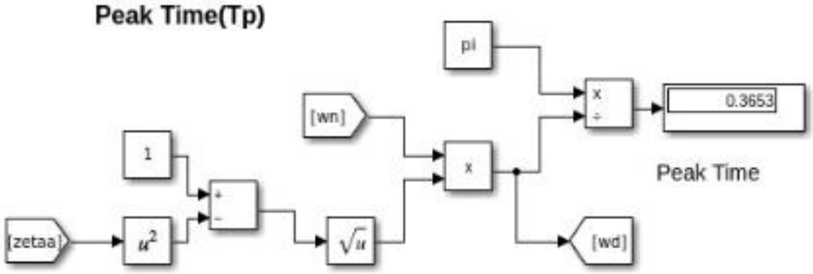
Damping Ratio(zetaa)



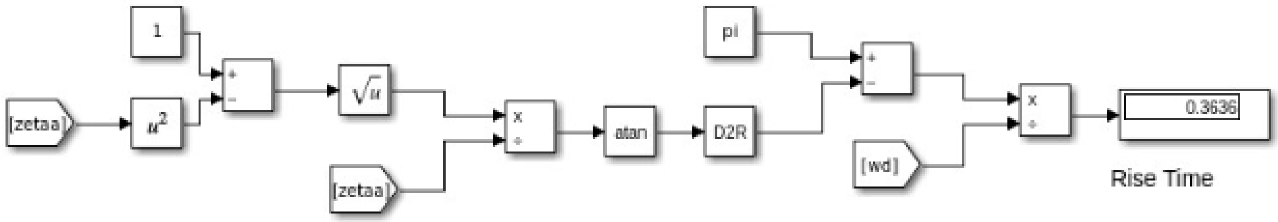
Settling TIme(Ts)



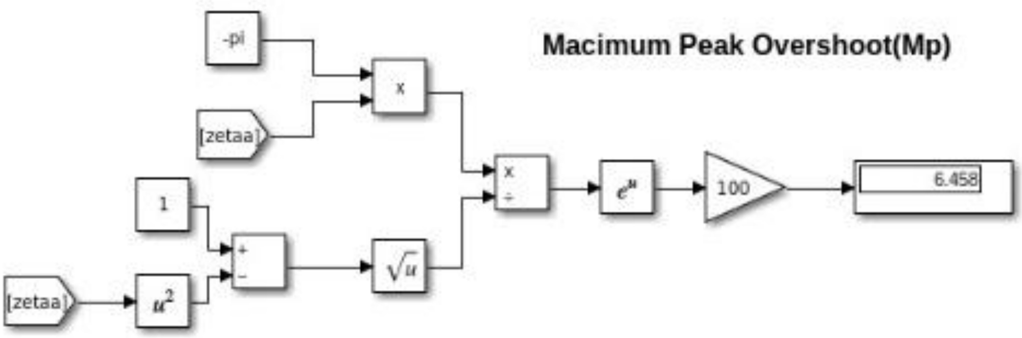
Peak Time(Tp)



Rise Time(Tr)



Macimum Peak Overshoot(Mp)



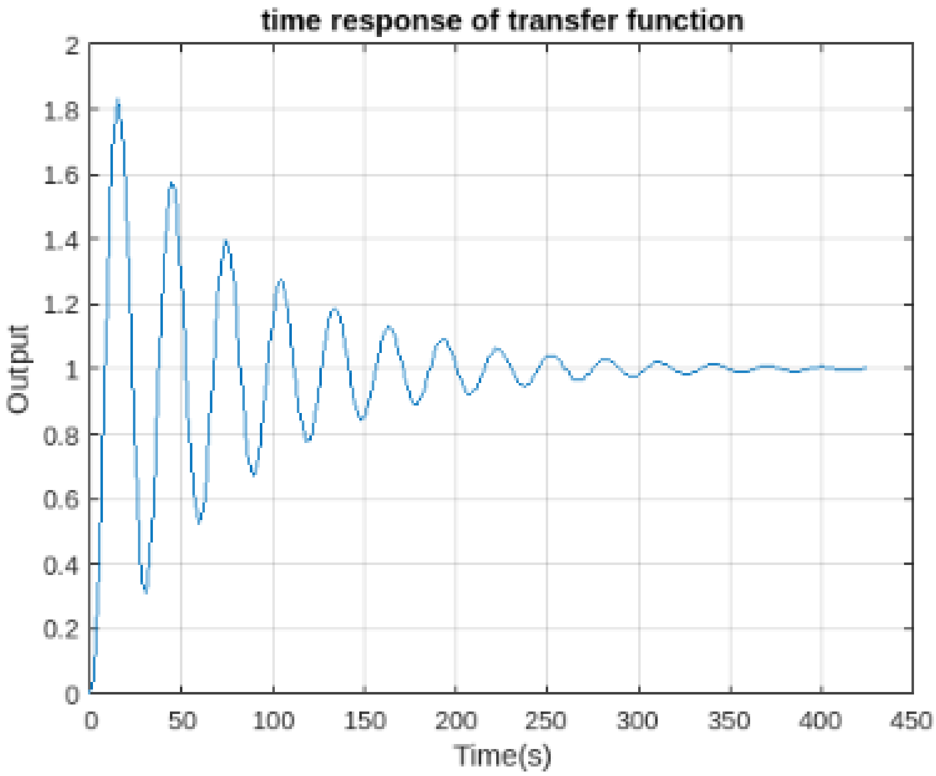
5. Find out the response and time parameters for the transfer function $TF_2 = \frac{C(s)}{R(s)} = \frac{0.045}{s^2+0.025s+0.045}$ for step input using MATLAB and Simulink.

<u>MATLAB Code:</u>

```
% Defining system parameters
R=0.01; % Resistance (ohms)
L=400*10^-3; % Inductance (henries)
C=55.5555;  % Capacitance (farads)
% Calculating natural frequency (wn) and damping ratio (zeta)
wn=1/(sqrt(L*C));
zetaa=(R/2)*(sqrt(C/L));
% Calculating settling time (ts), peak time (tp), rise time (tr), and
percent overshoot (mp)
ts=4/(zetaa*wn);
tp=pi/(wn*sqrt(1-zetaa^2));
x=atan(sqrt(1-zetaa^2)/zetaa);
y=deg2rad(x);
tr=(pi-y)/(wn*(sqrt(1-zetaa^2)));
mp=exp((-pi*zetaa/sqrt(1-zetaa^2)))*100;
gain=tf([0.045],[1 0.025 0.045]); % Defining transfer function
t=0:0.01:ts*2; % Generating time vector
% Calculating and plot step response
[y,t]=step(gain);
plot(t,y);
% Adding plot title, x-label, y-label and grid
title('time response of transfer function');
xlabel('Time(s)');
ylabel('Output');
grid on; % Turning on the grid for better readability
% Displaying calculated parameters
disp('Calculated Parameters:');
disp(['Natural Frequency (wn): ', num2str(wn)]);
disp(['Damping Ratio (zetaa): ', num2str(zetaa)]);
disp(['Settling Time (ts): ', num2str(ts)]);
disp(['Peak Time (tp): ', num2str(tp)]);
disp(['Rise Time (tr): ', num2str(tr)]);
disp(['Percent Overshoot (mp): ', num2str(mp)]);
```

<u>Output:</u>

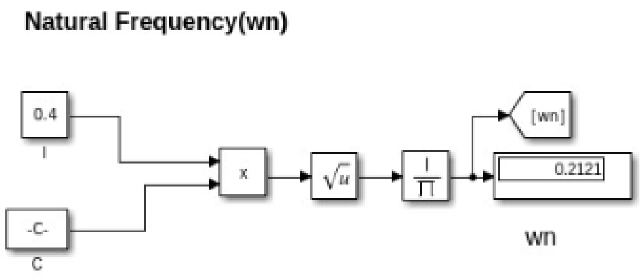
```
Calculated Parameters:
Natural Frequency (wn): 0.21213
Damping Ratio (zetaa): 0.058926
Settling Time (ts): 320
Peak Time (tp): 14.8354
Rise Time (tr): 14.7108
Percent Overshoot (mp): 83.0737
```

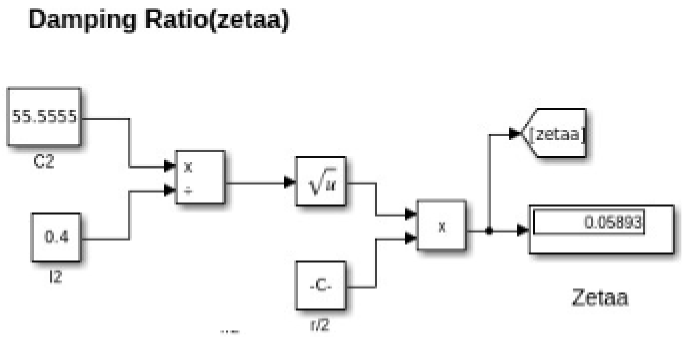


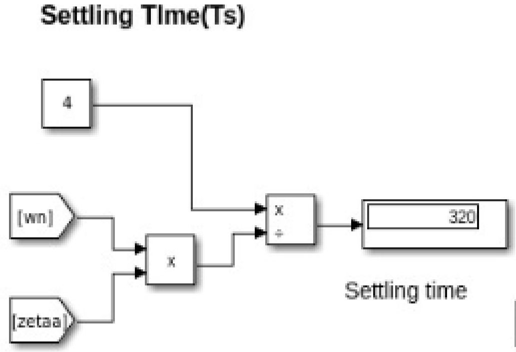
Workspace

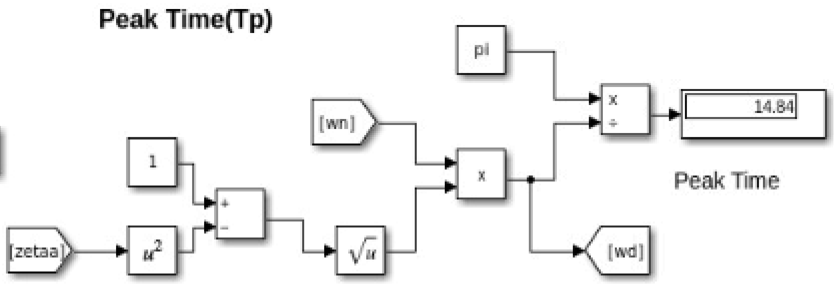
Name	Value	Size	Class
C	55.5555	1x1	double
L	0.4000	1x1	double
R	0.0100	1x1	double
gain	*1x1 tf*	1x1	tf
mp	83.0737	1x1	double
t	*64001x1 do...*	64001x1	double
tp	14.8354	1x1	double
tr	14.7108	1x1	double
ts	320	1x1	double
wn	0.2121	1x1	double
x	1.5118	1x1	double
y	*64001x1 do...*	64001x1	double
zetaa	0.0589	1x1	double

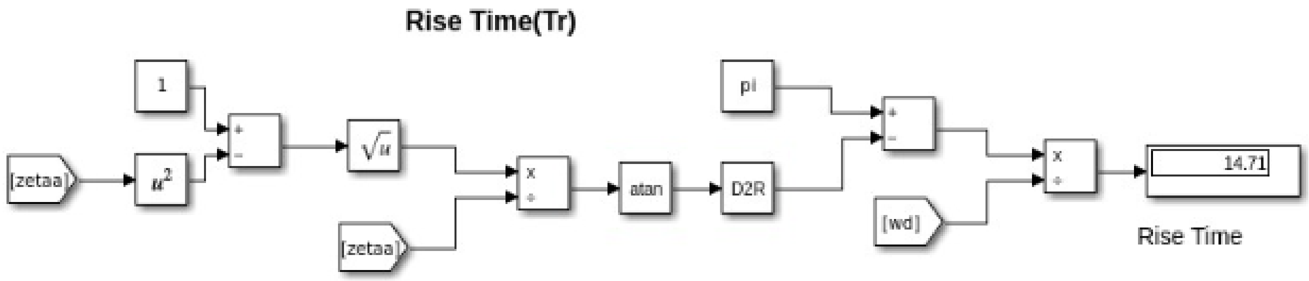
<u>Simulink:</u>

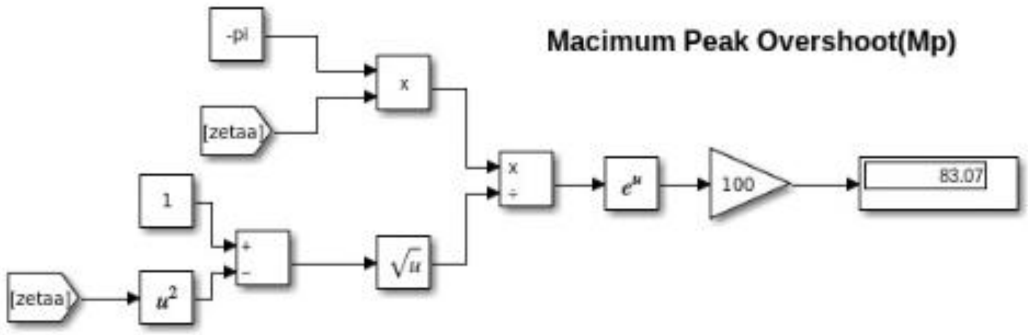

Natural Frequency(wn)
0.4
I
-C-
C
x
[wn]
0.2121
wn


Damping Ratio(zetaa)
55.5555
C2
0.4
I2
x
÷
-C-
r/2
x
[zetaa]
0.05893
Zetaa


Settling TIme(Ts)
4
[wn]
[zetaa]
x
x
÷
320
Settling time


Peak Time(Tp)
pi
[wn]
1
[zetaa]
+
-
x
x
÷
14.84
Peak Time
[wd]


Rise Time(Tr)
1
[zetaa]
+
-
x
÷
[zetaa]
atan
D2R
pi
+
-
x
÷
[wd]
14.71
Rise Time


Macimum Peak Overshoot(Mp)
-pi
[zetaa]
x
1
[zetaa]
+
-
x
÷
100
83.07