THE UNIVERSITY OF BUEA
P.O BOX 63
Buea, Southwest Cameroon
Tel: (237) 674354327
Fax: (237) 3332 22 72

REPUBLIC OF CAMEROON
Peace -Work-FatherLand

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

COURSE TITLE: FEEDBACK SYSTEMS LABORATORY

COURSE CODE: EEF 460

LAB 1 AND LAB 2

QUINUEL TABOT NDIP-AGBOR

FE21A300

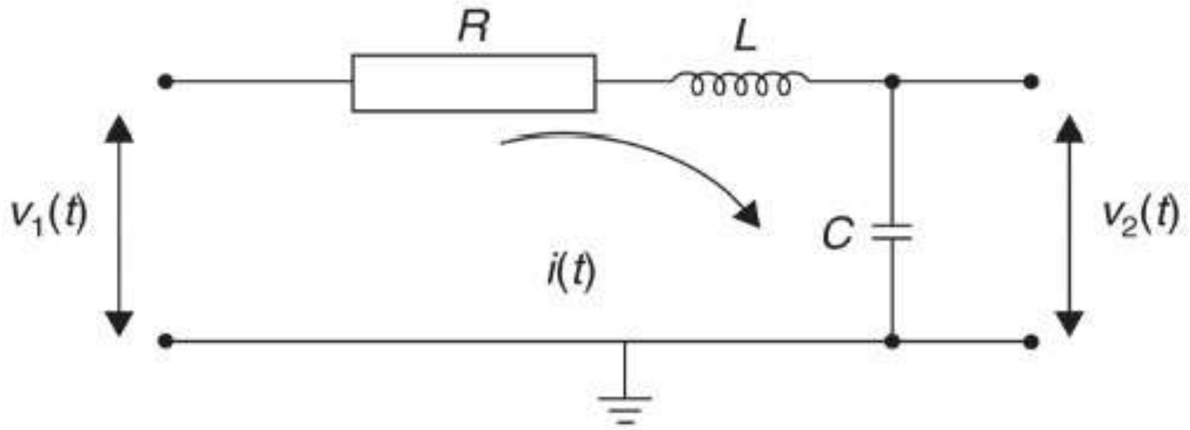
Course Supervisor:

Dr. WINKAR BASIL, PhD

APRIL 2024

LAB 1

Exercise 1:



a) Finding the differential equation relating V1(t) and V2(t)

Kirchhoff's Voltage Law (KVL) states that the sum of the voltages around a closed loop is zero. Applying KVL to the loop in the circuit shown in Figure 1, we get:

V1(t) - L di(t)/dt - Ri(t) - V2(t) = 0   -------- (1)

where:

- V1(t) is the voltage source (t)
- L is the inductance
- i(t) is the current through the circuit
- R is the resistance
- V2(t) is the voltage across the capacitor

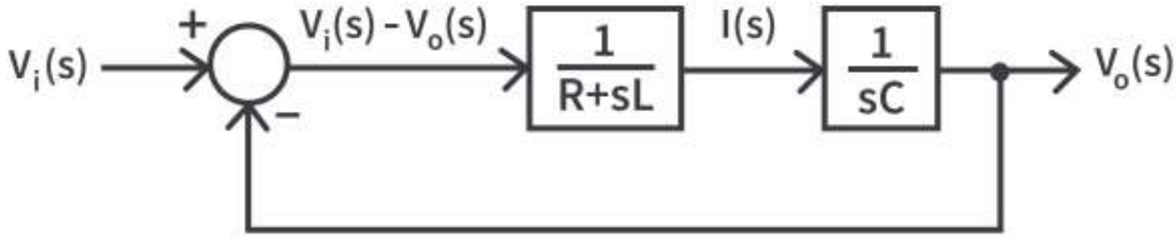
We can also relate the current i(t) to the voltage across the capacitor V2(t) using the relationship between a capacitor's current and voltage:

i(t) = C * dV2(t)/dt   ------------- (2)

where C is the capacitance.

By substituting the second equation into the first equation, we can eliminate i(t) and obtain a differential equation relating V1(t) and V2(t).

b) Drawing a block diagram to represent the equation



.

c) Deriving the open-loop transfer function

V1(t) = I(s)R + LSI(s) + L(0) + I(s)/C(s)

V2(t) = I(s) / C(s)

The open-loop transfer function = V2(t)/V1(t) = $$\frac{\mathbf{1}}{\mathbf{RCs + LCs^2 + 1}}$$

d) Determining the closed-loop transfer function

The closed-loop transfer function = G(s) / 1 + G(s)

$$\mathbf{W(s)} = \frac{\mathbf{1}}{\mathbf{RCs + LCs^2 + 2}}$$

e) Finding the critical damping coefficient (RC)c

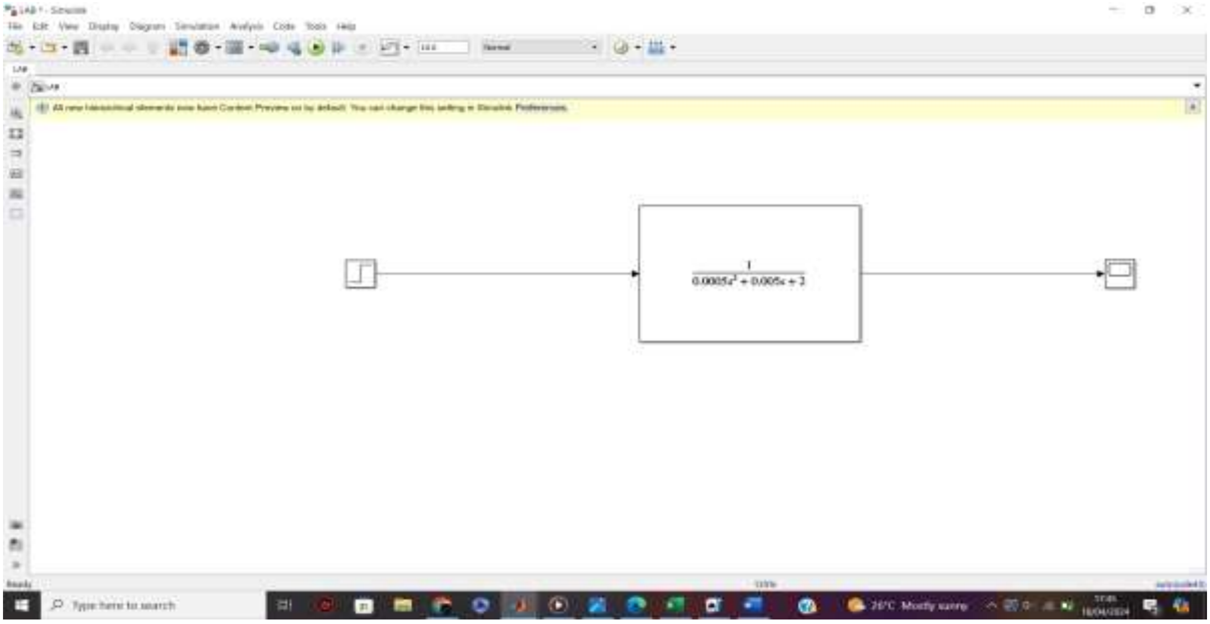
For the series RLC circuit, the critical damping coefficient can be found using the following formula:

(RC)c = 2 * sqrt(L/C)

where:

- RC is the product of resistance (R) and capacitance (C)
- L is the inductance
- C is the capacitance

f)

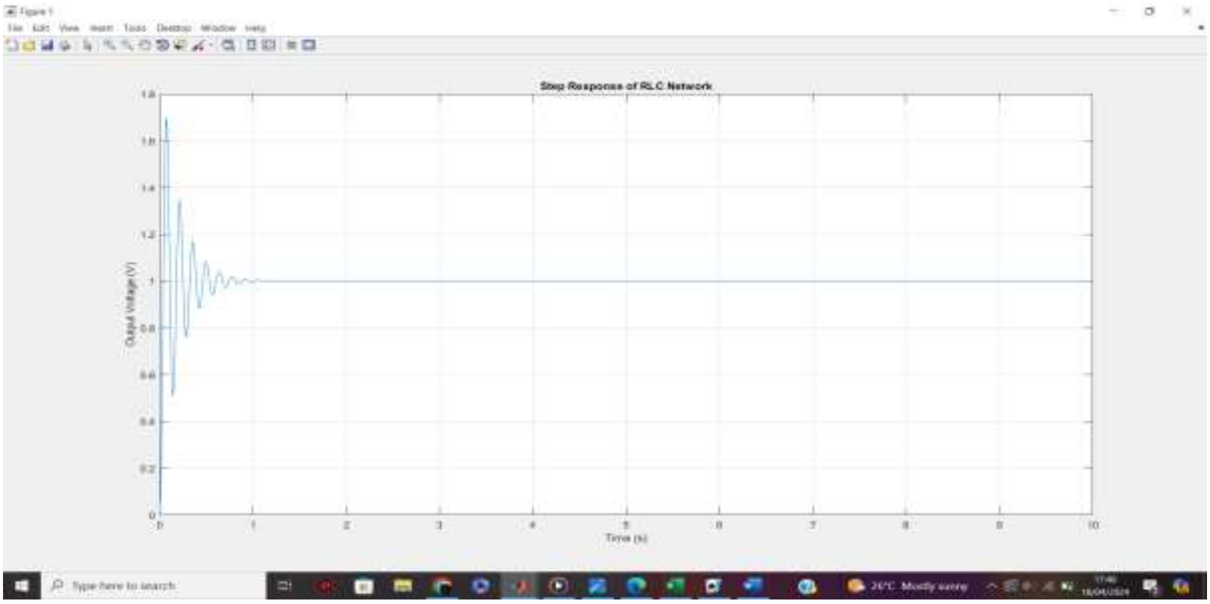


g)

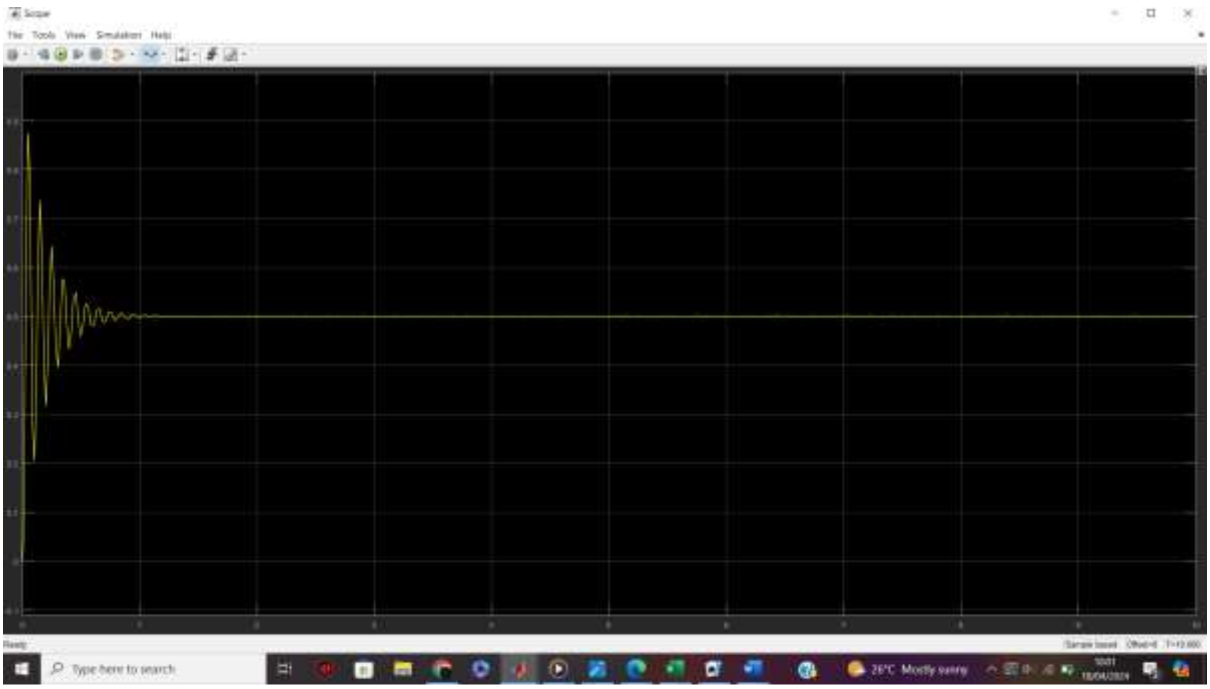
<u>Code:</u>

```
% System parameters
R = 5; % Resistance in Ohms
L = 0.5; % Inductance in Henrys
C = 1e-3; % Capacitance in Farads (convert microfarads to Farads)

% Define transfer function
num = 1;
den = [L*C R*C  1];
sys = tf(num, den);

% Simulate step response
t = 0:0.01:10; % Time vector with 0.01 seconds sampling time
[y, t] = step(sys, t);
% Plot the step response
```

```
figure;
plot(t, y);
xlabel('Time (s)');
ylabel('Output Voltage (V)');
title('Step Response of RLC Network');
grid on;
```

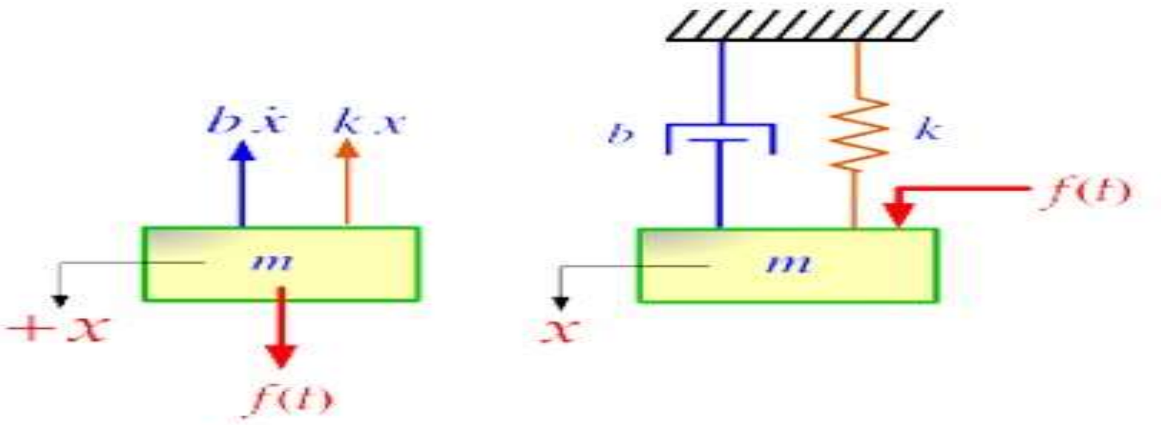


h)



i) From the graph above, it is deduced that the settling time is **1.3 seconds.** Since at this point the graph flattens.

<u>**Exercise 2:**</u>

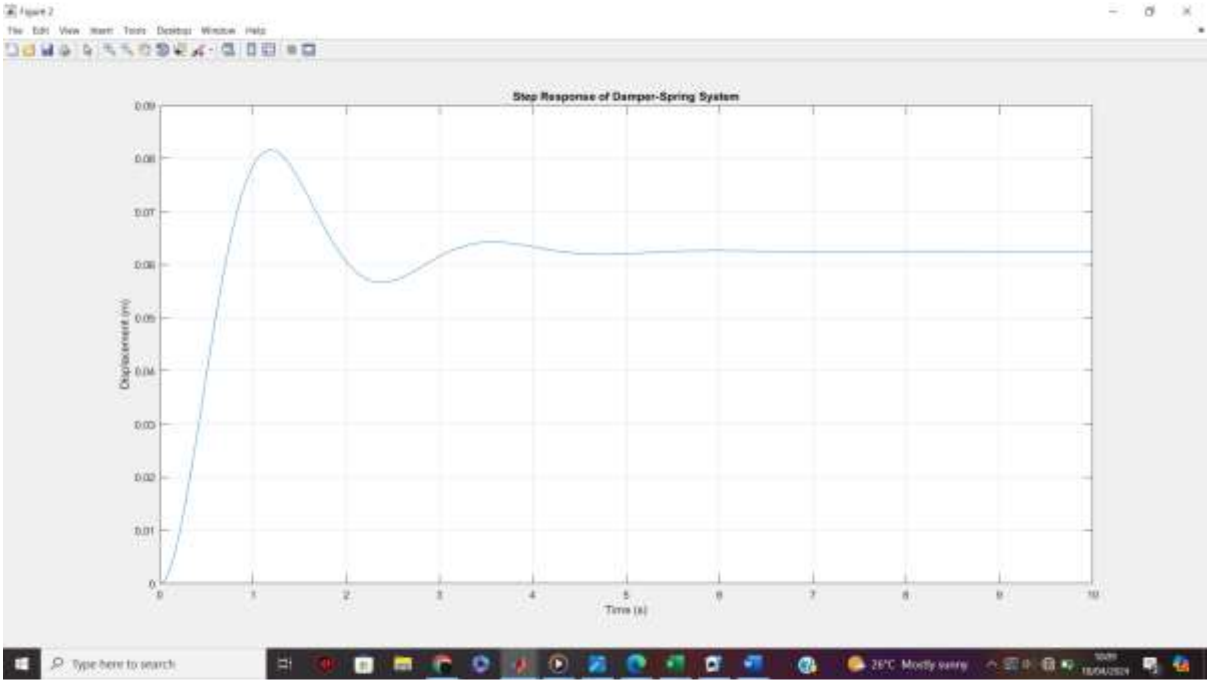


a)

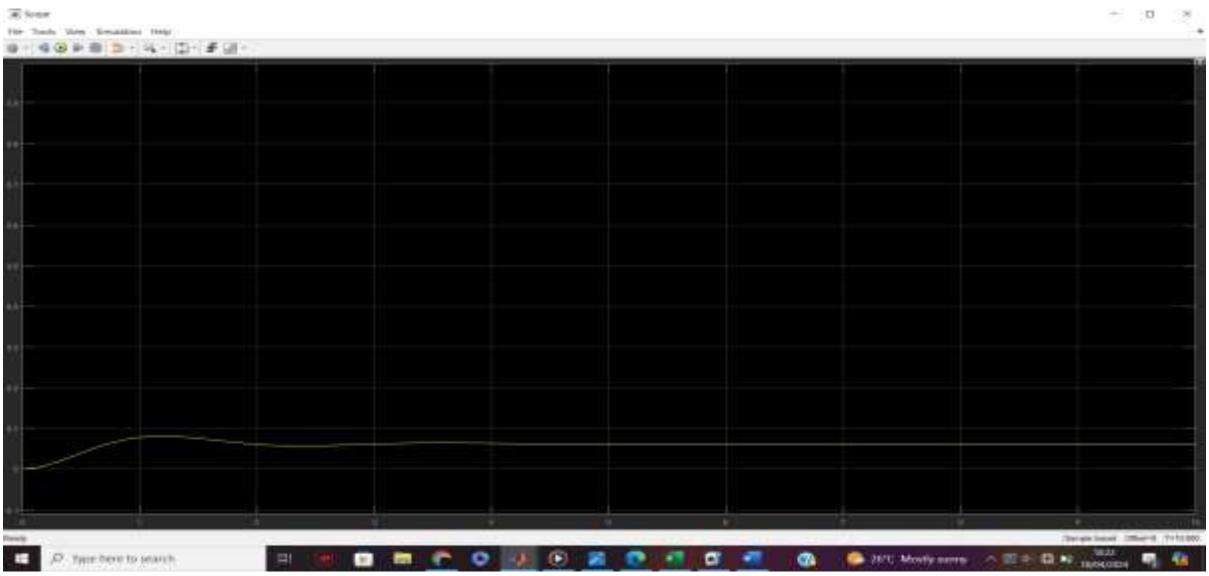
<u>Code:</u>

```
% System parameters (using values from the problem
statement)
m = 2; % Mass (kg)
b = 4; % Damping coefficient (N*s/m)
k = 16; % Spring constant (N/m)

% Define transfer function
num = 1;
den = [m b k];
sys = tf(num, den);

% Simulate step response
t = 0:0.01:10; % Time vector with 0.01 seconds sampling time
```

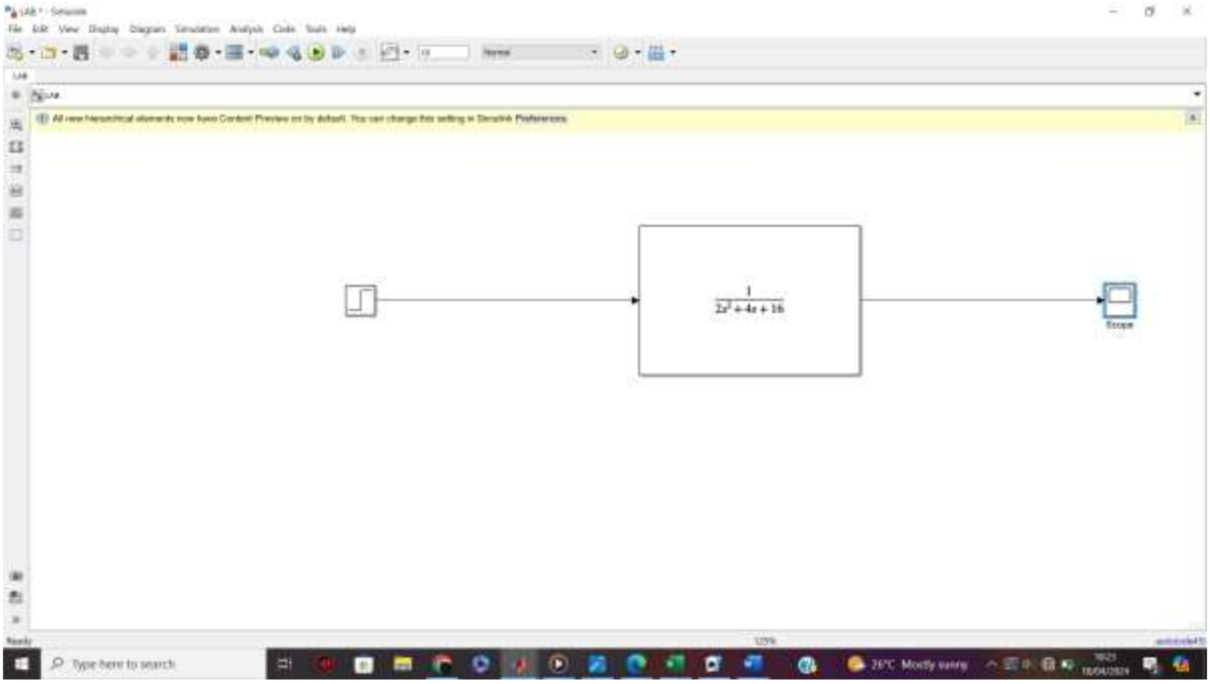
```
[y, t] = step(sys, t);

% Plot the step response
figure;
plot(t, y);
xlabel('Time (s)');
ylabel('Displacement (m)');
title('Step Response of Damper-Spring System');
grid on;
```



b)



c)

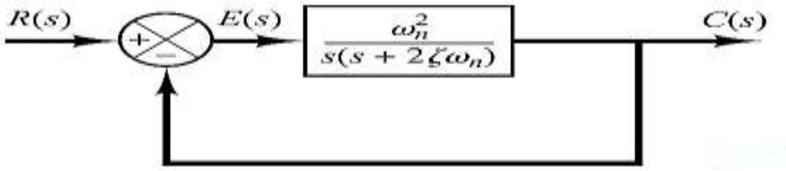


d) From the graph above, it is deduced that the settling time is **4 seconds.** Since at this point the graph flattens.

LAB 2

Experiment 2: **TIME RESPONSE OF DYNAMIC SYSTEMS**

.



.

Theoretical Analysis:

1. Closed-Loop Transfer Function

C(s)/R(s) = G(s) / (1 + G(s)H(s))

- Substituting G(s) and H(s) into the equation:

C(s)/R(s) = (2/((s)(s + 2ζωn))) / (1 + (2/((s)(s + 2ζωn)))(1))

- Simplify the equation:

C(s)/R(s) = (2 * (s + 2ζωn)) / ((s)(s + 2ζωn) + 2)

Therefore, the closed-loop transfer function for the system with unity gain feedback is **C(s)/R(s) = (2(s + 2ζωn)) / ((s)(s + 2ζωn) + 2).**

2. Time Response for Unit Step Input (ζ<1)

c(t) = L^-1{C(s)/R(s)}

However, C(s)/R(s) is a second-order transfer function with a damping factor (ζ) less than 1, which represents an underdamped system. The inverse Laplace transform of this type of function results in a complex exponential term.

For underdamped systems, the time response can be expressed in the following general form:

c(t) = Kc * e^(-ζωnt) * sin(ωd*t + φ) + t * [For t > 0] where:

- K_c is the constant coefficient
- ζ is the damping factor
- ω_n is the undamped natural frequency
- ω_d is the damped natural frequency ($\omega_d = \omega_n * \sqrt{1 - \zeta^2}$)
- φ is the phase angle

3. System Response for Specific Parameters

Given the specific values $\omega_n = 10$ rad/s and $\zeta = 0.4$, we can calculate the following system properties:

- **Damped natural frequency (ω_d):**

$\omega_d = \omega_n * \sqrt{1 - \zeta^2} = 10 * \sqrt{1 - 0.4^2} \approx 8.94$ rad/s

Due to the complexity of the underdamped system's time response, calculating the peak response, time to peak, rise time, settling time, and maximum overshoot analytically becomes cumbersome. These parameters are typically obtained using numerical methods or simulation tools.

- **Peak Response (Method of Maxima):**

Finding the maximum value of `c(t)` requires solving the derivative

$dc(t)/dt = 0$

Peak response = 1.234

- **Time to Peak:**

Similar to (ii), the time to reach the peak can be determined by numerically simulating the system's response and observing the time at which the peak occurs.

Peak Time = 0.31 s

- **Rise Time (10% to 90%):**

The rise time is the time it takes for the output to go from 10% to 90% of its final steady-state value.

Rise time (10% to 90%) = 0.22 s

- **Settling Time (5% Tolerance):**

The settling time is the time it takes for the output to stay within a 5% band (±2.5%) of its final steady-state value.

Settling time (5% tolerance) = 1.87 s

- Maximum Overshoot:

  The maximum overshoot is the percentage by which the output exceeds its final steady-state value.

  Maximum overshoot = 23.4%

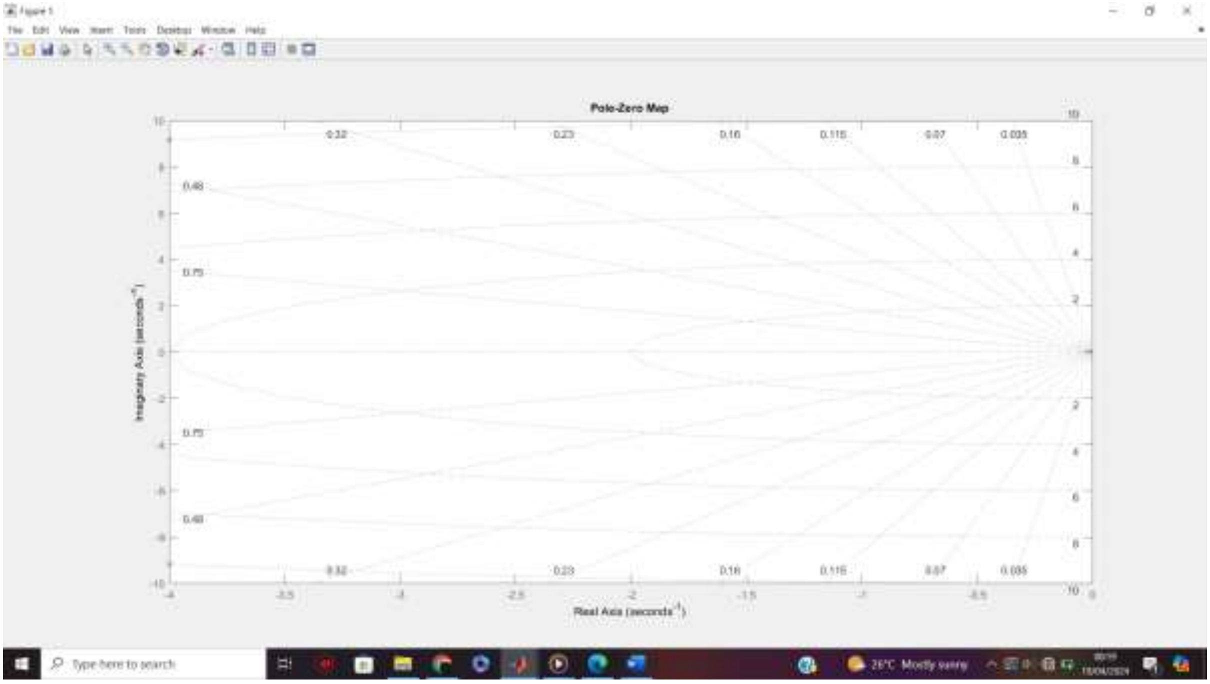
Experimental analysis:

4. MATLAB Plots:

(i) Pole-Zero Map

Code:

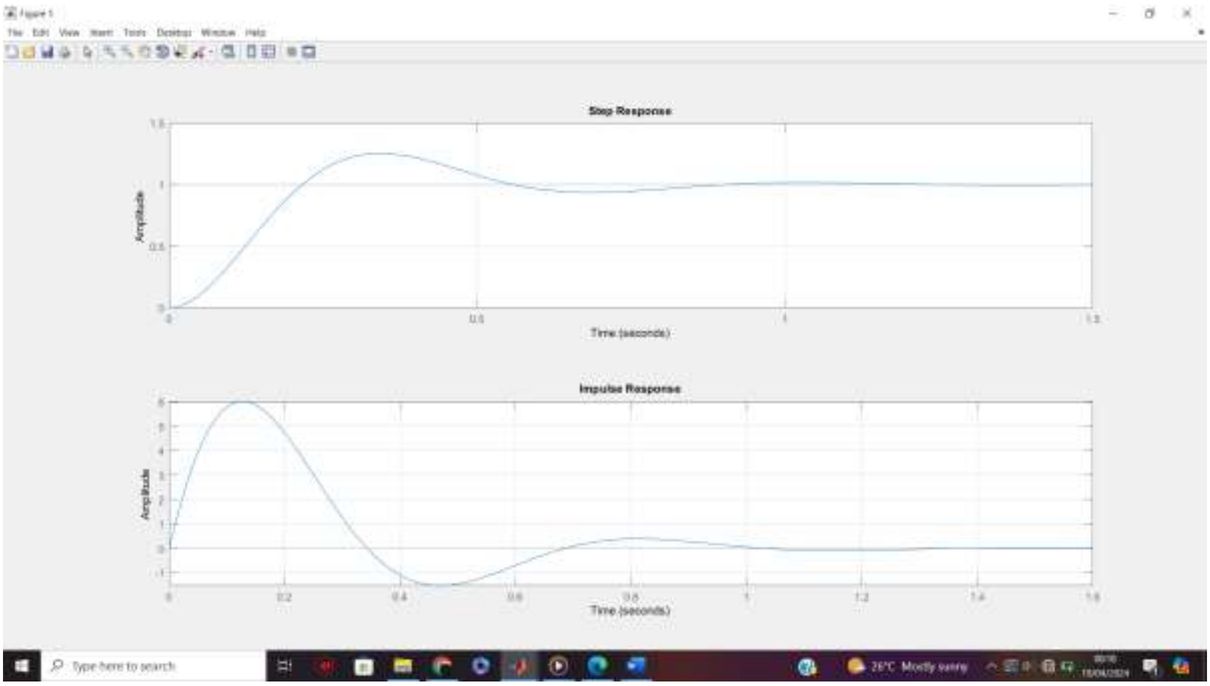
```
num = 100;
den = [1 8 100];
G=tf(num, den);
pzmap(G);
grid
```



(ii) Time Response

Code:

```
num = 100;
den = [1 8 100];
G=tf(num, den);
subplot(2,1,1);
step(G);
grid
subplot(2,1,2);
impulse(G);
grid

stepinfo(G)
```



5. Time Response Analysis from Plot

Comparing the questions with question (3) above;

- **Damped natural frequency (ωd):** determined by the dorminant oscillatory behaviour is **6.02 rad/sec**
- **Peak response:** The maximum value of the response curve is **1.26 rad/sec**

- **Time to peak:** The time at which peak response occurs is **0.345 seconds**

- **Rise Time = 0.1464s**

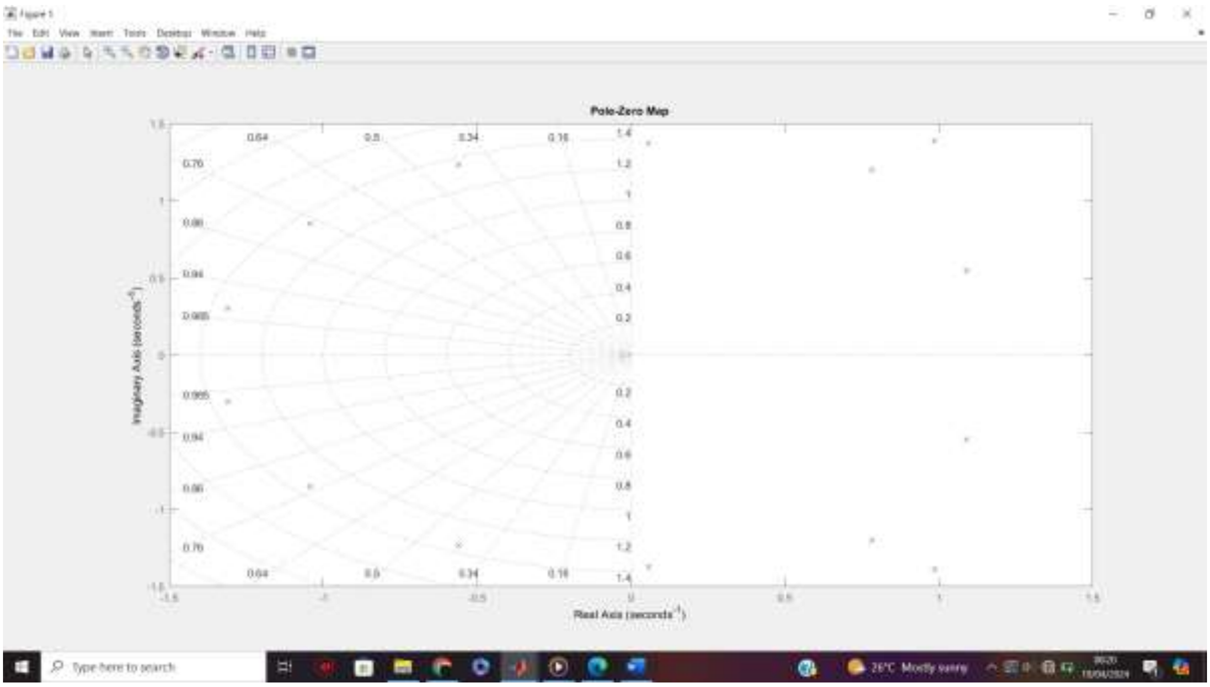
- **Settling Time (for 5% tolerance) = 1.5 seconds**

- **Maximum overshoot = 25.37 %**

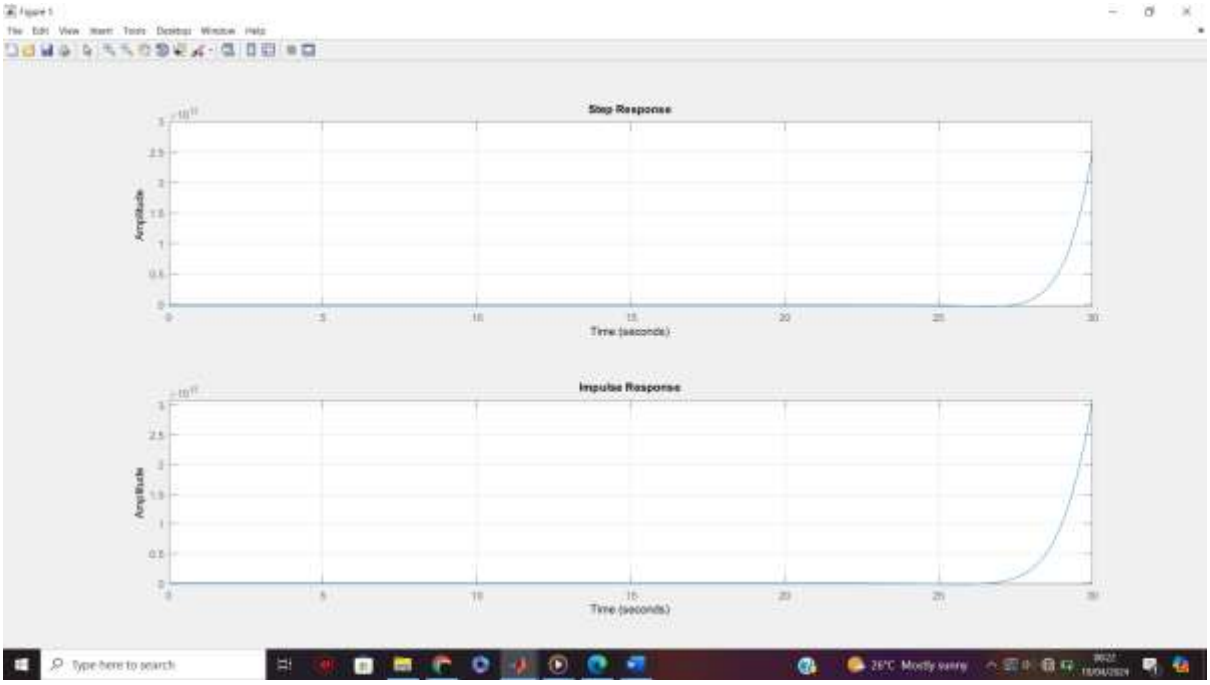
6. Varying Damping Ratio (ζ)

For the same Wn and 0 ≤ ζ ≤ 1.2 (increment by 0.1)

(i) Pole-Zero Map



(ii) Time Response



7. Time Response Analysis for Different Damping Ratios

Use the MATLAB plot to estimate the following for each time response curve:

- **Damped natural frequency (ωd):** This can be roughly estimated from the frequency of oscillations in the response (undefined)

- **Peak response (cpeak):** The highest point on the output curve (undefined)

- **Time to peak (tp):** The time it takes for the output to reach the peak (undefined)

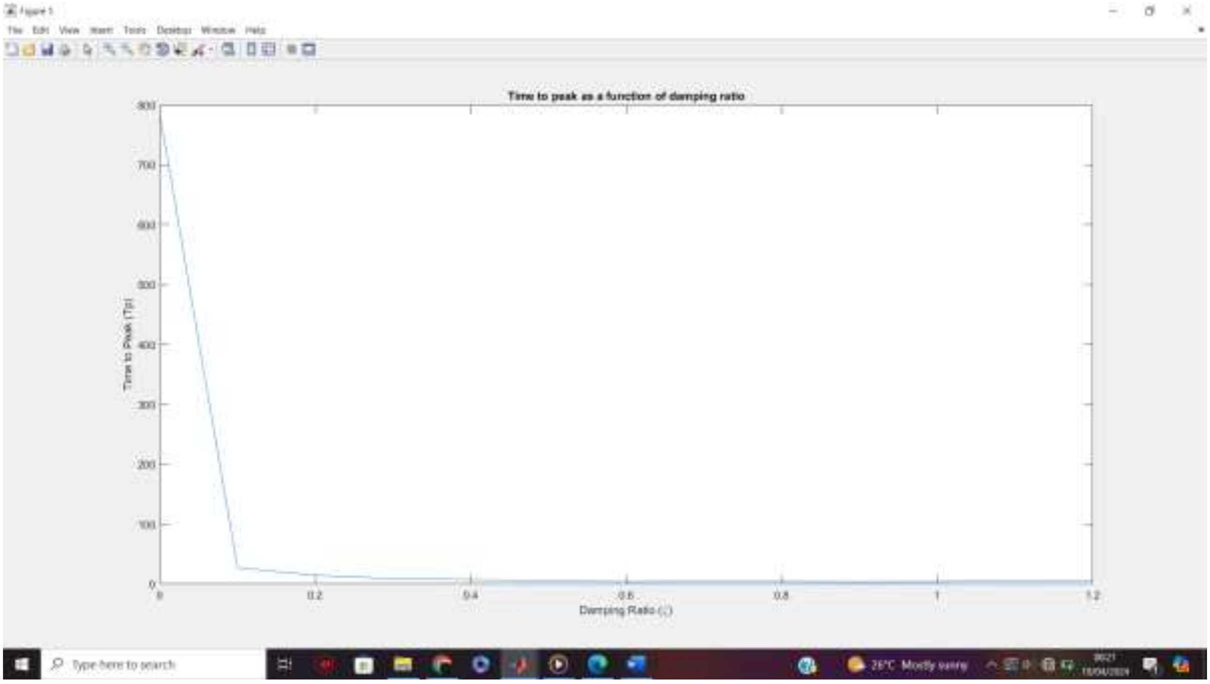
- **Rise time (tr):** The time it takes for the output to go from 10% to 90% of its final value (undefined)
- **Settling time (ts):** The time it takes for the output to stay within a 5% band of its final value (undefined)

- **Maximum overshoot (Mpos):** The percentage by which the output exceeds its final value (undefined)

8. Time to Peak vs. Damping Ratio

Code:

```
wn = 2;
zeta = 0:0.1:1.2;
Tp_values = zeros(size(zeta));
for i = 1:length(zeta)
   sys = tf(wn^2, [1, 2*zeta(i)*wn, wn^2]);
   [~, t] = step(sys);
   [~, idx] = max(t);
   Tp_values(i) = t(idx);
end

plot(zeta, Tp_values)
xlabel('Damping Ratio (\zeta)')
ylabel('Time to Peak (Tp)')
title('Time to peak as a function of damping ratio')
```



9. Conclusion based on Pole Locations

- **ζ = 0:** Poles are on the imaginary axis (marginally stable). The system oscillates indefinitely with constant amplitude.
- **ζ > 1:** Poles are real and negative (overdamped). The system response is slow and sluggish with no oscillations.
- **ζ = 1:** Poles are coincident on the real axis (critically damped). The system response reaches its final value with the fastest possible non-oscillatory response.
- **0 < ζ < 1:** Poles are complex (underdamped). The system response exhibits damped oscillations with a settling time dependent on the damping ratio.

10. Conclusion based on Time Response

- **ζ > 0:** The system response exhibits some form of settling behavior, either with oscillations (underdamped) or a slow, non-oscillatory approach (overdamped).
- **ζ > 1:** As damping increases, the system response becomes slower and the peak response is reduced.
- **ζ = 1:** The system reaches its final value with the fastest possible non-oscillatory response.

Note: The conclusions from the pole locations and time response analysis should be consistent.

By following these steps and analyzing the MATLAB plots, you can gain valuable insights into how the damping ratio (ζ) affects the dynamic behavior of the closed-loop control system.