Name- Himanshu Shete(23B0770)

Part 1: Engineering Insight

1. System Definition

System: The coffee pouring mechanism controlled by a robotic arm adjusting the tilt of a carafe.
Input variable: Applied voltage/current to the tilt motor (control input).
Output variable: Amount of coffee flowed (or alternatively, tilt angle if linearly mapped).
Feedback loop needs to monitor: Actual tilt angle or flow rate (indirectly proportional to volume) in order to maintain accurate control to stop at 200mL without overshooting.

2. Controller Intuition

Proportional (P): Reacts to current error. If the arm is not tilted enough, P increases the motor effort. Helps reduce steady-state error but too much P can cause overshoot.
Integral (I): Accounts for accumulated past errors. Ensures the cup eventually reaches exactly 200mL even if the system has biases (like friction). But too much I leads to slow correction and instability.
Derivative (D): Looks at the rate of error change. It damps the response and prevents overshoot by predicting future error trends. Helps control oscillations and improve settling time.

Part 2: Mathematical Modeling

3. System Modeling (Laplace Domain)

Let the transfer function be:

$$G(s) = \frac{K}{s^2 + as + b}$$

- **K (gain):** Represents system responsiveness — how much output per unit input.
- **a (damping factor):** Represents friction/resistance in the motor and system.
- **b (stiffness or inertia):** Reflects the system's natural tendency to return to equilibrium (e.g., gravity effects, mechanical spring-like behavior).

4. Pole-Zero Analysis

Sketch (not drawn here) is a complex s-plane:

- **Overshoot:** Poles lie in the **left half-plane** but close to the **imaginary axis** (low damping).
- **Oscillations:** Poles are **complex conjugates** with small real parts (Re ≈ 0), indicating underdamped behavior.

- **Sluggishness:** Poles are **real and negative**, but close to origin (small real part ⇒ slow response).

Effect of a Zero:
A zero adds a frequency where the numerator becomes zero. Depending on location, it can:

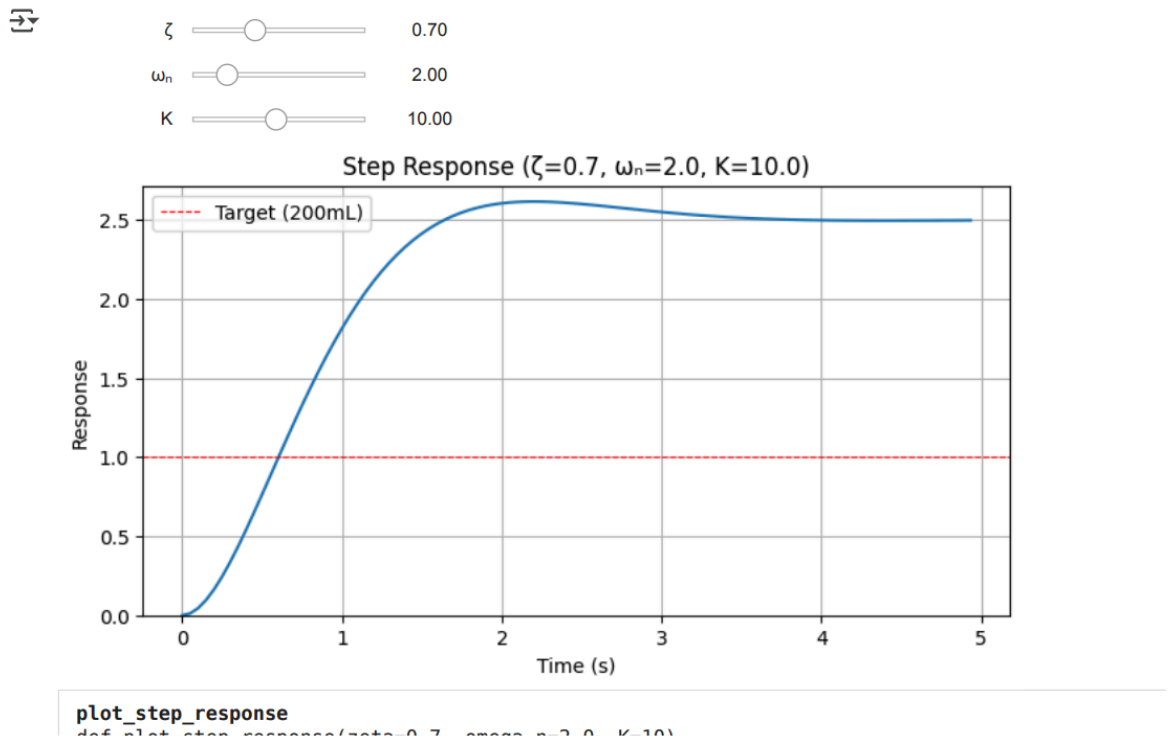
- **Speed up** the response (zero near origin)
- **Introduce peaking** or amplify certain frequencies
- Make controller design more complex

Part 3: Interactive Simulation in Python

$$G(s) = \frac{K}{s^2 + as + b}$$

. K=10;
https://github.com/himu23/ls-25/blob/main/The%20Control%20Theory%20Bootcamp/control_theory_ass1%20(1).ipynb

```
import numpy as np
import matplotlib.pyplot as plt
from control import TransferFunction, step_response
from ipywidgets import interact, FloatSlider


def plot_step_response(zeta=0.7, omega_n=2.0, K=10):
    a = 2 * zeta * omega_n
    b = omega_n**2

    num = [K]
    den = [1, a, b]

    sys = TransferFunction(num, den)
    t, y = step_response(sys)

    plt.figure(figsize=(8, 4))
    plt.plot(t, y)
    plt.title(f'Step Response (ζ={zeta}, ωₙ={omega_n}, K={K})')
    plt.xlabel('Time (s)')
    plt.ylabel('Response')
    plt.grid(True)
    plt.ylim(0, max(1.5, max(y) + 0.1))
    plt.axhline(1, color='r', linestyle='--', linewidth=0.8, label='Target (200mL)')
    plt.legend()
    plt.show()


interact(
    plot_step_response,
    zeta=FloatSlider(value=0.7, min=0, max=2, step=0.05, description='ζ'),
    omega_n=FloatSlider(value=2.0, min=0.1, max=10, step=0.1, description='ωₙ'),
    K=FloatSlider(value=10, min=1, max=20, step=1, description='K')
)
```



- Increase in Damping: Reduces overshoot and oscillations, faster settling
- Decrease in Damping: Increases oscillation, potential overshoot
- Increase in natural frequency: Faster response
- Decrease in natural frequency: Slower response, may reduce overshoot