ELECTRONICS & ROBOTICS CLUB
IIT Bombay

erciitbombay@gmail.com
erciitb.github.io
+91 824 722 7509 | +91 953 820 7404

Control Theory Bootcamp

Assignment 1: Basics of Control Systems

The Perfect Pour Coffee Robot Problem — Submission Deadline: 23:59, 16 June 2025

Scenario:

You're on the engineering team at *Chaayos+*, a startup building smart coffee machines. Your current prototype uses a robotic arm to pour **exactly 200mL** of coffee from a carafe into a cup.

However, your pouring arm faces issues:

- **Overshoot** (spills),
- **Slow settling** (arm takes too long to stop),
- **Oscillations** (wobbly pouring).

The tilt of the carafe is adjusted by a motor. Your team lead wants you to design a control system that ensures smooth, accurate pouring—every single time.

Part 1: Engineering Insight

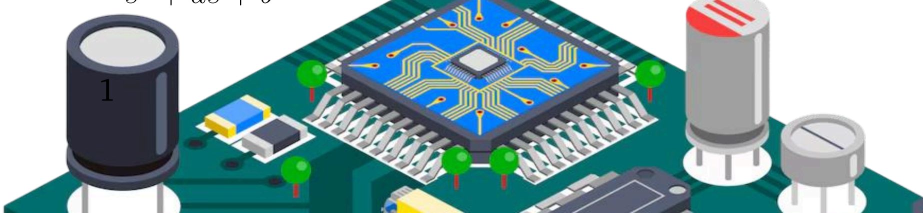
1. **System Definition:**
   In your own words, what is the *system* you're controlling? What are the input and output variables for this system? What should the feedback loop measure?

2. **Controller Intuition:**
   Given the observed issues (overshoot, oscillations, lag), describe why a **Proportional**, **Integral**, and **Derivative** component might be helpful in a controller—without using math.

Part 2: Mathematical Modeling

3. **System Modeling (Laplace Domain):**
   Assume the tilt motor and pouring dynamics can be modeled as a second-order system. Propose a generic transfer function:

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



Institute Technical Council
Students Gymkhana
IIT Bombay, Powai
Bombay 76

What do the parameters K, a, and b represent physically in the context of pouring coffee?

4. **Pole-Zero Analysis:**
   Sketch a rough s-plane and mark where the poles might lie in the case of:
   - Overshoot,
   - Oscillations,
   - Sluggishness.

   Also, describe the impact of having a zero in the transfer function on system response.

Part 3: Interactive Simulation

Use Python to simulate your transfer function:

$$G(s) = \frac{10}{s^2 + as + b}$$

- Plot the step response of this system.
- Use interactive sliders (e.g., via `ipywidgets`) to tune:
  - Damping ratio (ζ),
  - Natural frequency (ω_n),
  - Gain (K).
- Observe how the response changes in terms of overshoot, settling time, and oscillations. **Hint:** Think about the time you want each cup to fill up in to figure out the constants a and b.

Deliverables:

- Answers to the 4 conceptual and mathematical questions.
- Python code with step response plot and interactive sliders.
- Commentary on how system behavior changes with damping and frequency.