# Assignment 2: Joint-Space Trajectory Generation and Smoothness Analysis

---

## Objective

The objective of this assignment is to move beyond static robot poses and understand how a robot transitions between configurations over time. Students will learn how to generate **joint-space trajectories** between two robot configurations and analyze the effect of different trajectory choices on motion smoothness.

This assignment introduces the concept of **trajectory as a function of time**, which is a critical foundation for trajectory optimization and learning-based methods explored later in the project.

---

## Problem Description

Consider the same **2-link planar robotic arm** studied in Assignment 1, with link lengths l1, l2. The robot starts at an initial joint configuration:
(q_1^{start}, q_2^{start}
and must move to a final joint configuration:
(q_1^{end}, q_2^{end})
Instead of directly visualizing only the start and end poses, you are now required to generate **continuous joint trajectories** that describe how the robot moves between these two configurations over a fixed time duration TTT.

---

## Trajectory Representation

A joint-space trajectory is defined as:
q1(t), q2(t),   t ∈ [0, T]
You will generate and compare **two types of trajectories**:

1. **Linear joint-space trajectory**
2. **Smooth polynomial joint-space trajectory**

---

# Expected Work

## Linear Joint-Space Trajectory

In this approach, each joint angle varies linearly with time between its start and end values. This results in a simple trajectory but may introduce abrupt changes in velocity at the start and end of motion.

Students should:

- Define a time vector $t \in [0,T]$
- Linearly interpolate $q1(t)$ and $q2(t)$
- Plot joint angles as functions of time

---

## Smooth Polynomial Trajectory

To achieve smoother motion, students will generate a **polynomial trajectory** (cubic or quintic) that ensures smooth start and stop conditions.

The trajectory should:

- Start and end at the specified joint angles
- Have zero velocity at the start and end of motion
- Produce smoother joint angle profiles compared to linear interpolation

Students should:

- Compute polynomial coefficients for each joint
- Plot the resulting joint trajectories over time
- (Optional) Compute and plot joint velocities to observe smoothness

---

# Analysis and Comparison

Students must compare the two trajectories by:

- Plotting joint angle vs. time for both methods
- Observing differences in smoothness near the start and end points
- Discussing which trajectory is more suitable for a real robotic system and why

No animation is required. The emphasis is on **trajectory plots and interpretation**.

---

# Submission Requirements

Each submission must include:

1. A Python script or Jupyter notebook containing:
    - Linear trajectory generation
    - Polynomial trajectory generation
2. Clearly labeled plots showing:
    - Joint angles vs. time (for both trajectories)
3. A short written discussion (5–7 sentences) addressing:
    - Differences between linear and smooth trajectories
    - Why smoothness is important for real robots

Ps. Assignment is to be submitted on github, the instructions on how to submit is in the updated readme
**Deadline - 23rd December, EOD.**