Assignment 4: Prediction and Dashboard

Objective

The objective of this assignment is to integrate optimization and learning into a complete trajectory planning pipeline. You will train a machine learning model to approximate optimized joint-space trajectories and deploy an interactive dashboard that allows users to explore and compare optimized and learned trajectories.
This assignment demonstrates how learning-based methods can be used to accelerate trajectory generation in real-world robotic systems.

Problem Description

Consider the same 2-link planar robotic arm used throughout the project. Optimized joint-space trajectories have already been generated in Assignment 3 using numerical optimization.
In this assignment, you will use those optimized trajectories as training data to learn a mapping from start and end joint configurations to a full joint-space trajectory. The trained model should be able to quickly predict a smooth trajectory for new start and end configurations without running an optimizer.

Dataset Preparation

You should:

- Generate multiple optimized trajectories for different start and end joint configurations
- Store joint trajectories as training data
- Split the dataset into training and testing sets

Each data sample should consist of:

- Input: start and end joint angles
- Output: joint trajectory over time

Learning-Based Trajectory Prediction

You are required to:

- Train a neural network to predict joint trajectories
- Use a simple architecture such as a multilayer perceptron
- Evaluate prediction performance using an appropriate error metric (for example, mean squared error)

The focus should be on understanding the learning pipeline rather than achieving perfect accuracy.

Interactive Dashboard

You must build a simple interactive dashboard that:

- Accepts start and end joint angles as user input
- Displays the optimized trajectory
- Displays the neural-network-predicted trajectory
- Allows visual comparison using joint-angle plots and end-effector paths

The dashboard may be implemented using Streamlit or any similar Python-based interface.

Comparison and Analysis

You should analyze:

- Differences between optimized and learned trajectories
- Situations where the learned model performs well or poorly
- Trade-offs between prediction accuracy and computation time

A short discussion should explain when learning-based approaches are preferable to direct optimization.

Submission Requirements

Your submission must include:

- Python code for dataset generation and model training
- Python code for the interactive dashboard
- Plots comparing optimized and learned trajectories
- A short written report (8–10 sentences) summarizing results and observations

Submission Deadline - 12th January EOD