Introduction:

In this project, a 7-degree-of-freedom KUKA LBR iiwa 7 R800 robotic arm is programmed to carry out a certain function. The robot arm is attached to a base and fitted with an adaptor that features a rectangular item on one side and a camera on the other.
Using the robot's workspace, the objective is to move the rectangular object and place it precisely on top of a rectangle target. The camera detects an Aruco marker, which provides the target's direction and location.
The essential prerequisites are:

In order to place the rectangular object as close to the goal as feasible, we generate a trajectory in the joint space that moves the robot from its initial configuration.
Make sure the robot's motion stays away from the table or surface where the target is positioned to prevent accidents.
The robot has to move for precisely ten seconds in total and should stay at end position for last 2 seconds
Throughout the action, all of the robot joints' angular locations and velocities must stay within their designated ranges.

In addition to the position and orientation of the Aruco marker with respect to the camera's reference frame, the information sent also contains the robot arm's initial setup.
To put it briefly, the task requires the robotic arm to precisely place a rectangular object on a target within the workspace while abiding by predetermined joint restrictions and time constraints. This is accomplished by trajectory planning and execution.

Methodology:

The A matrices needed for the transformation are computed via the compute_A_matrices function.The following phases comprise the process for designing the combined trajectory:

2.1. Constants and Initialization

Joint offsets (d), rotation angles (alpha), link lengths (a), joint angles (qc), and joint offsets (d) are among the variables and constants that the code initialises. The rotation angles that are supplied are used to calculate the transformation matrices T2, T3, and T4. (where T2 is Tc T3 Euler's T4 endposition)

2.2. Kinematics Forward

The role of forward kinematics For every joint, FK calculates the transformation matrices.

The Tbt, Tef, and Tbe transformation matrices are computed.

2.3. Generation of Trajectories

It computes the desired end-effector posture (Tbe).
To get the required pose, an iterative method modifies the joint angles (Q).
To connect joint velocities to end-effector velocities, the Jacobian matrix (Ja) is calculated.

2.4. Coordinated Path Generation

Quintic polynomial interpolation is used to create the joint trajectory.
In order to interpolate between the initial and final joint configurations smoothly, polynomial coefficients are generated.
The trajectory that is obtained is saved in the PoseG matrix.

2.5. Calculating the Velocity Profile

The derivative of the joint trajectory is used to calculate the velocity profile for each joint.
The VeloG matrix contains the velocity profile.

3. Put into Practice

The above methodology is implemented by the code that is provided. It makes use of the forward kinematics, trajectory, and velocity profile computation functions. Smooth joint trajectories are produced by quintic polynomial interpolation.

4. Conclusion

The robotic arm can easily and efficiently achieve the intended end-effector attitude thanks to the specified trajectory. The methodology that has been put into practice offers a strong foundation for robotic system trajectory planning.

Qf = 1.7149 -1.3809 -2.0563 -2.6726 -3.017 -0.3974 2.0241