FINAL REPORT

Harshal Salian
Course: MEEG671: Introduction to Robotics

Introduction

The KUKA LBR iiwa 7 R800 robotic arm, equipped with seven degrees of freedom, is central to this project, tasked with the precise positioning of a rectangular object onto a target within its operational workspace. This sophisticated robot starts from a predetermined configuration, noted as q1, which is [58.2686, 75.3224, 11.7968, 45.9029, −22.1081, −31.2831, −42.3712] degrees. The goal is to accurately place the object such that its corners A, B, C, and D align perfectly with those of the target.

Mounted on the robot flange is an adapter which holds both a camera and the rectangular object on opposite sides. This setup utilizes real-time image processing facilitated through an Aruco marker to determine the target's precise position and orientation within the robot's workspace. The Aruco marker software provides detailed feedback on the position and orientation of the marker with respect to the camera, with coordinates given by translations along the X, Y, and Z axes of .205780720039398(m), -0.109793029482687 (m), and 0.561252115509121 (m), respectively, and orientation angles in Euler ZYX notation of Roll x: 174.1750404305652 degrees, Pitch y: -17.3967534123935degrees, and Yaw z: -1.9587388578232degrees.

The challenge of this project extends beyond simple object placement. The trajectory must be meticulously planned in joint space to move the arm from its starting configuration to the final position over the target, ensuring each movement adheres to the robot's mechanical constraints and operational limits. The trajectory planning is critical as it must account for smoothness and precision to avoid any abrupt movements or collisions, making use of the robot's advanced control capabilities to adjust dynamically to the feedback provided by the camera and Aruco marker system.

❖ Methods

System Setup and Initial Configuration

The robotic setup includes a camera mounted strategically on the robot's flange to detect an Aruco marker. This marker's detection is crucial for determining the target's exact position within the robot's workspace. The robot begins from a predetermined configuration q1, which sets the stage for all subsequent transformations and movements.

❖ Transformation Matrices

The project utilizes several key transformation matrices:

- End Effector to Camera (TEC): Adjusts for the offset between the robot's end effector and the camera, essential for accurate target detection and interaction.
- Camera to Aruco Marker (TCA): Translates and rotates the camera's frame to align with the Aruco marker, crucial for defining the target's location in space accurately.
- Aruco Marker to Target (TAT): Establishes a fixed spatial relationship between the marker and the target, pivotal for the precise placement of the object.

The multiplication of these matrices (T0E * TEC * TCA * TAT) provides the complete transformation from the robot's base to the target, facilitating exact calculation of the necessary position and orientation at the end effector.

❖ Inverse Kinematics

Inverse kinematics (IK) in this project is handled through an iterative numerical approach designed to refine the robot's joint angles, ensuring that the end effector reaches the designated target position and orientation accurately. The MATLAB code utilizes a Jacobian-based method, which is pivotal for relating changes in joint angles to changes in the end effector's position and orientation.

- Error Calculation: At each iteration, the difference between the current end effector's position (and orientation) and the target position (and orientation) is calculated. This error is used to adjust the joint angles in the subsequent iteration.

- **Jacobian Matrix Computation: The Jacobian matrix, which is recalculated in each iteration, is a fundamental component of this process. It describes how slight changes in each of the joint angles affect the position and orientation of the end effector. In the code, the Jacobian is computed using the current state of the robot's configuration.**
- **Pseudo-Inverse of Jacobian: To find the optimal change in joint angles that minimizes the positional error, the pseudo-inverse of the Jacobian matrix is used. This mathematical tool allows the system to handle cases where a direct inverse is not possible (which is common in robotics due to the system being over-determined or under-determined). The computed pseudo-inverse is then used to determine the necessary adjustments to the joint angles to move the end effector towards the target position and orientation.**
- **Update Rule: The joint angles are updated based on the product of the pseudo-inverse of the Jacobian and the positional error, scaled by a gain factor to ensure smooth convergence. This update rule is applied iteratively until the error is reduced below a predefined threshold, or a maximum number of iterations is reached.**

❖ Trajectory Planning

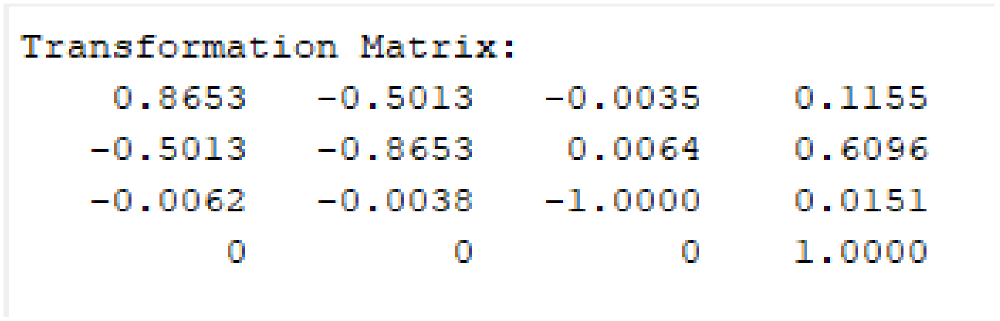
Trajectory planning in the project ensures that the movement from the initial to the final joint configuration is smooth and adheres to the robot's kinematic and dynamic constraints. The MATLAB code achieves this through a carefully designed cubic polynomial trajectory, which describes how each joint angle should change over time.

- **Cubic Polynomial Parameters: The trajectory for each joint is defined by a cubic polynomial, chosen for its smooth acceleration and deceleration characteristics. The coefficients of these polynomials are calculated based on the initial and final joint configurations, as well as initial and desired velocities (typically set to zero at both ends for smooth starting and stopping).**
- **Discrete Time Simulation: The trajectory is evaluated at discrete time intervals, allowing the control system to generate a series of intermediate joint configurations, which lead the robot from its starting pose to the target pose. This method ensures that all movements are within the robot's operational limits, such as maximum allowable velocities and accelerations for each joint.**

- Velocity and Acceleration Constraints: To ensure that the movement is not only smooth but also feasible, the velocities and accelerations calculated from the trajectory are checked against the robot's specifications. This is crucial to prevent mechanical overloads and ensure the longevity and safety of the robotic system.

❖ Results

The implementation of the MATLAB code successfully directs the robotic arm to position the rectangular object precisely over the target. The trajectory followed by the robot is verified to be smooth and within all operational limits. Simulations confirm the accuracy of the target placement, with no violations of the predefined mechanical constraints of the robot.



```
Transformation Matrix:
    0.8653   -0.5013   -0.0035    0.1155
   -0.5013   -0.8653    0.0064    0.6096
   -0.0062   -0.0038   -1.0000    0.0151
         0         0         0    1.0000
```

❖ Conclusion

This project successfully demonstrates the precise execution of advanced robotic control techniques, culminating in the accurate achievement of the final transformation of the KUKA LBR iiwa 7 R800 robotic arm. By seamlessly integrating inverse kinematics and trajectory planning, laid a robust foundation for tasks requiring exceptional precision