POSITION OR FORCE CONTROL FOR ROBOTIC MANIPULATORS
CLOSING BOTTLE CAP WITH OPENMANIPULATOR-X

IITGN ROBOTICS

Introduction

Robotic manipulators, or robotic arms, are crucial in modern automation, enhancing efficiency from industrial assembly lines to everyday tasks. This project explores the intricate control mechanisms required to perform the seemingly simple task of closing a bottle cap using the OpenMANIPULATOR-X robotic arm. Although humans execute this effortlessly, replicating it with robotics involves complex control strategies. Our research focuses on position control, aiming to advance the capabilities of robotic manipulators. By implementing and analyzing these control approaches on the OpenMANIPULATOR-X, we aim to enhance precision and automation in robotic tasks, demonstrating significant implications for future applications.

C-space of OpenMANIPULATOR-X

The configuration space (C-space) of an n-joint manipulator, such as OpenMANIPULATOR-X, encompasses all possible positions and orientations the robot can achieve. For an n-joint manipulator, the C-space is n-dimensional, with each dimension corresponding to a joint variable. Determining the degrees of freedom (DoF) involves identifying the minimal set of independent parameters needed to describe all possible configurations of the robot within its operational limits. Gruebler's formula helps us to calculate the Dof for a system:

$$F = 6 \times (L - 1 - J) + \sum_{i=1}^{J} f_i$$

Joint type	dof f	Constraints c between two planar rigid bodies	Constraints c between two spatial rigid bodies
Revolute (R)	1	2	5
Prismatic (P)	1	2	5
Helical (H)	1	N/A	5
Cylindrical (C)	2	N/A	4
Universal (U)	2	N/A	4
Spherical (S)	3	N/A	3

Thus, we can represent the C-space as $C = \{\theta_1, \theta_2, \theta_3, \theta_4\}$.

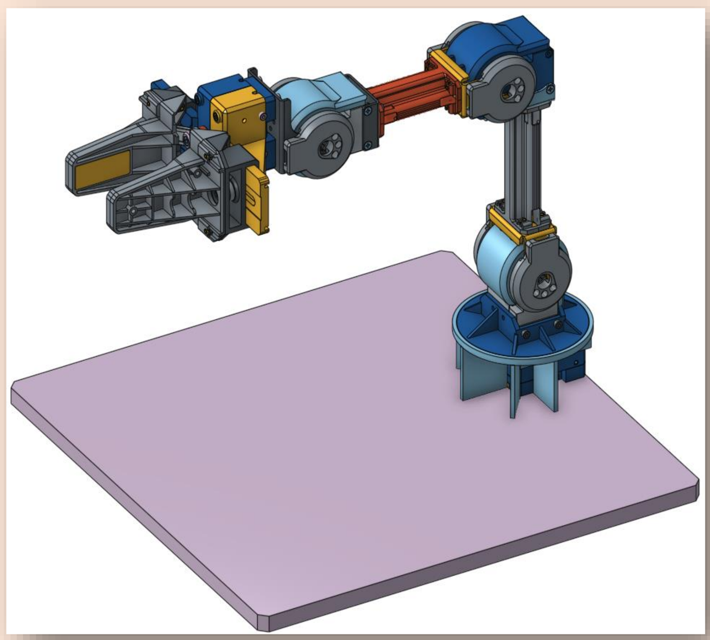
Forward and Inverse Kinematics

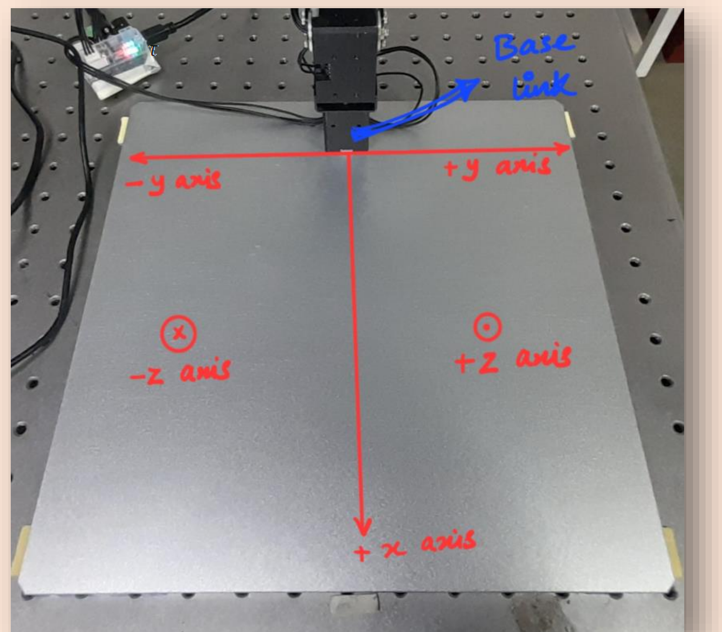
Forward kinematics for OpenMANIPULATOR-X with four revolute joints computes the transformation matrix T from the base to the end-effector frame:

$$T = T_1 T_2 T_3 T_4$$

where Ti represents the transformation matrix for each joint, describing both rotation and translation. Inverse kinematics involves determining the joint angles $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ that achieve a desired end-effector pose by solving $T = T_1 T_2 T_3 T_4$.

Advanced solvers such as KDL, IKFast, and MoveIt! are utilized for efficient computation of inverse kinematics solutions in applications involving OpenMANIPULATOR-X. These solvers automate the process, enabling precise positioning and trajectory planning for robotic tasks.





Force Reading Through Manipulator Jacobian

The Jacobian matrix in robotics is critical for relating joint torques τ to the forces f_{tip} experienced at the end-effector of OpenMANIPULATOR-X:

$$\tau = J^T f_{tip}$$

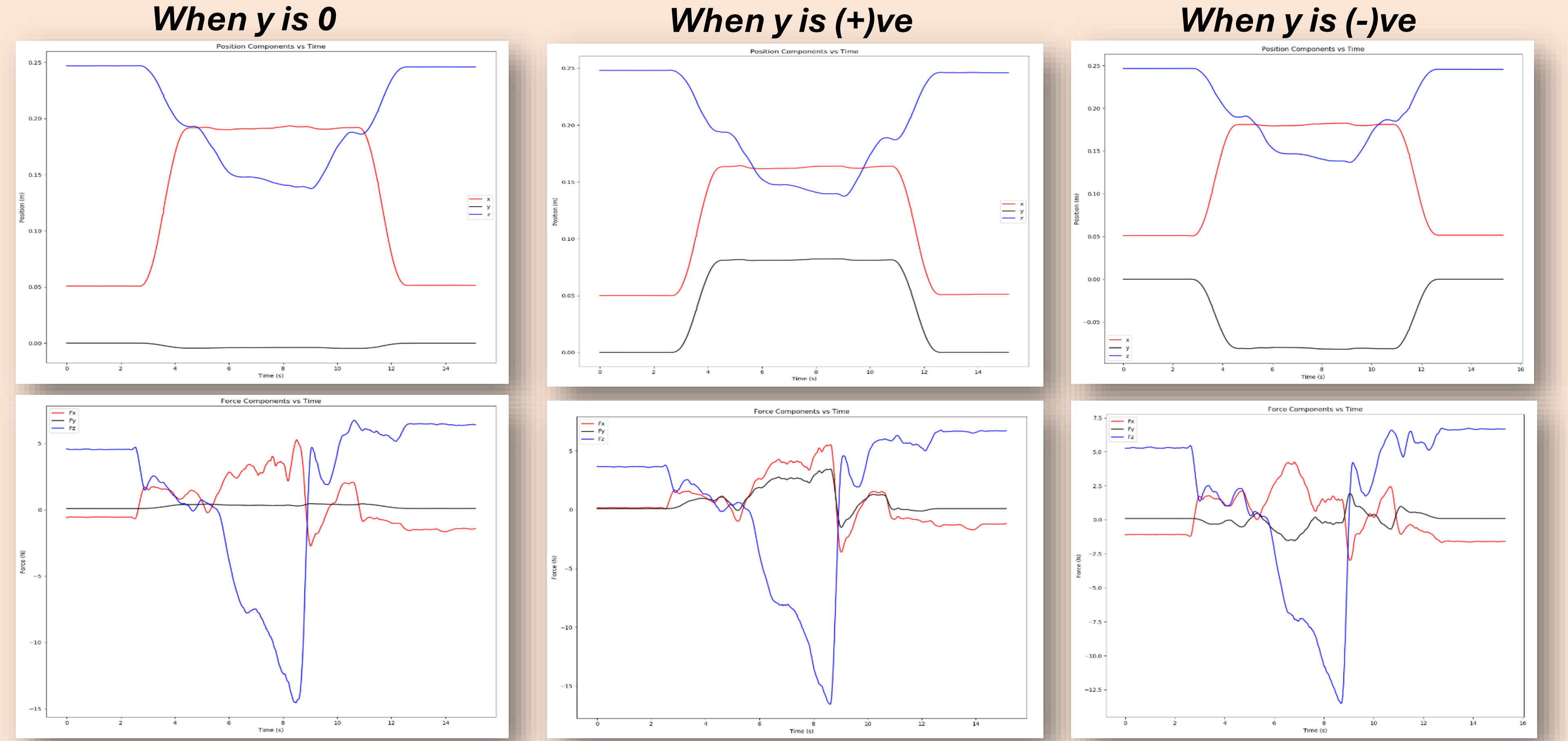
Here, τ represents the joint torque matrix (4x1), and f_{tip} denotes the end-effector force matrix (6x1), consisting of $\{f_x, f_y, f_z, \tau_x, \tau_y, \tau_z\}^T$. The transpose of the Jacobian matrix (J^T) has dimensions (4x6) for OpenMANIPULATOR-X, relating how forces exerted at the end-effector translate into torques at the joints. The torque matrix can't be directly obtained from the inbuilt ROS-topics, hence we get the current of the joint motors and then use the performance graph to obtain the joint torques.

Methodology

1. Moving the end-effector from point A to B using IK. The OpenMANIPULATOR-X has certain inbuilt services to use them to set the pose and the orientation of the two points
2. We use the help of OpenCV to get the coordinates of the centre point (x,y) in pixels. We use the `cv2.HoughCircles()` function to figure out circles under the webcam's frame.
3. We use transformation equations to convert the (x,y) coordinates of the detected cap (in pixels) into coordinates (in metres) of the cap concerning the base link of the OpenMANIPULATOR-X.
4. Now, we apply force control to ensure the manipulator closes the cap tightly. We can also use position control to do the same, for which an algorithm needs to devised.
5. While performing position control IK solvers may not solve certain poses directly but, when the previous pose is very similar it would be able to solve it easily. So, uniform motion of orientation and z-coordinate from initial to required pose is incorporated.

Results

When y is 0 | *When y is (+)ve* | *When y is (-)ve*



Conclusion

This research optimized bottle cap closure using position control with OpenMANIPULATOR-X. It applied forward and inverse kinematics for precise movement control. ROS tools helped real-time IK without manual computations. A tailored position control algorithm effectively managed cap closure, ensuring accurate end-effector positioning. Overall, the study underscores the efficacy of position control in complex tasks, offering insights for future advancements in robotic automation.

SRIP POSTER BY: TAMIZHANBAN A G, 22110269, B.TECH ELECTRICAL ENGINEERING, IITGN