Chapter 2: Kinematics

2.1 Forward Kinematics

2.1.1 Introduction

In this section, we calculate the position and orientation of the End Effector frame relative to the base frame using forward kinematics. The strategy involves computing the individual transformations for each joint and then combining them to determine the overall transformation from the base frame to the End Effector frame.

2.1.2 Coordinate Frames

In the following Figure 2.1 , the coordinate frames of the Kinova Gen3 Lite manipulator are depicted. These frames are associated with each actuator, the base and the End Effector, illustrating their respective positions and orientations in space. Understanding these frames is fundamental for analyzing the robot's kinematic behavior and determining its movements within its workspace.

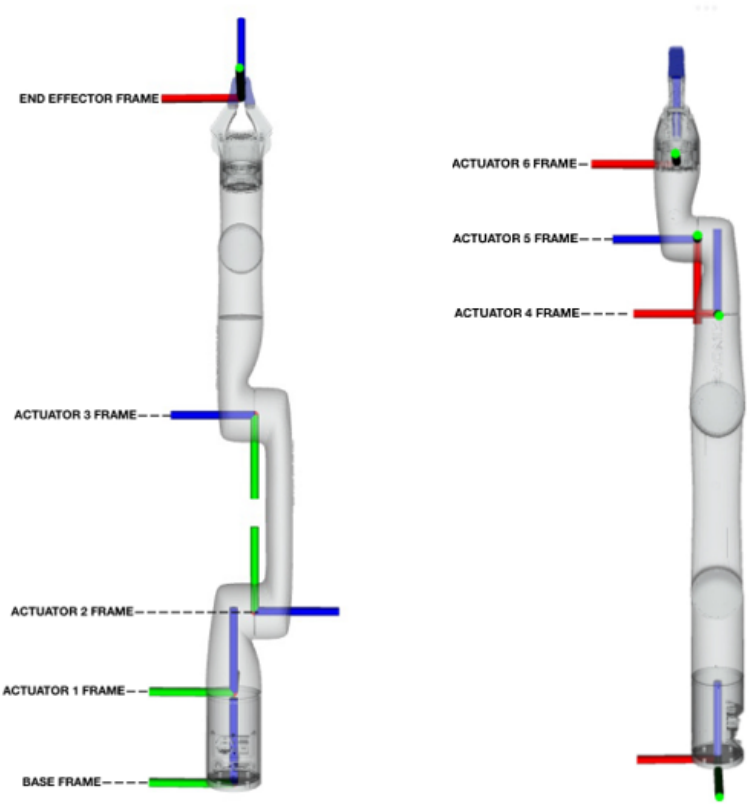


Figure 2.1: Manipulator Frames

2.1.3 Transformation Matrices

The overall transformation from base frame to the tool frame is given by:

$$^{Base}T_{Tool} = {}^{Base}T_1^{*\,1}T_2^{*\,2}T_3^{*\,3}T_4^{*\,4}T_5^{*\,5}T_6^{*\,6}T_{EE}$$

Here:

$$^{i-1}T_i^* = {}^{i-1}T_i \cdot R_z(q_i)$$

where:

- $^{i-1}T_i^*$ is the matrix for the general transformation from frame $[i-1]$ to frame $[i]$.
- $^{i-1}T_i$ is the transformation from the previous frame $[i-1]$ to the current frame $[i]$ when $q_i = 0$ (set angle for joint $i = 0$).
- $R_z(q_i)$ is the transformation matrix for a rotation of q_i around joint i (the z-axis

for the joint frame is always defined to be along the joint axis of rotation).

The rotation matrix $R_z(q_i)$ is given by:

$$R_z(q_i) = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 & 0 \\ \sin(q_i) & \cos(q_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In the table2.1 below, we can see the transformation matrices that describe the relationship between consecutive frames of the manipulator. These matrices include both the general transformation matrices (${}^{i-1}T_i^*$), which incorporate the joint angles q_i, and the specific transformation matrices (${}^{i-1}T_i$) for the case where $q_i = 0$. Each matrix captures the rotation and translation parameters required to transition from one frame to the next.

Transformation	${}^{i-1}T_i$	${}^{i-1}T_i^*$
Base to frame 1	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \cos(q_1) & \sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 1 to frame 2	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -L_2 \\ 0 & 1 & 0 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & 0 \\ 0 & 0 & -1 & -L_2 \\ \sin(q_2) & \cos(q_2) & 0 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 2 to frame 3	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & L_4 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & 0 \\ -\sin(q_3) & -\cos(q_3) & 0 & L_4 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 3 to frame 4	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -L_5 \\ 0 & 1 & 0 & L_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \cos(q_4) & -\sin(q_4) & 0 & 0 \\ 0 & 0 & -1 & -L_5 \\ \sin(q_4) & \cos(q_4) & 0 & L_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 4 to frame 5	$\begin{bmatrix} 0 & 0 & 1 & L_7 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & L_8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & L_7 \\ \sin(q_5) & \cos(q_5) & 0 & 0 \\ -\cos(q_5) & \sin(q_5) & 0 & L_8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 5 to frame 6	$\begin{bmatrix} 0 & 0 & -1 & -L_9 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & L_{10} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -1 & -L_9 \\ \sin(q_6) & \cos(q_6) & 0 & 0 \\ \cos(q_6) & -\sin(q_6) & 0 & L_{10} \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Frame 6 to End Effector frame	$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & L_{11} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	

Table 2.1: Transformation Matrices

2.1.4 Forward Kinematics Computation

For the purposes of our work, the overall transformation from the base frame to the End Effector frame is computed by focusing on the first three joints of the manipulator. To simplify the process, from joint 3 onward, the transformation matrices are evaluated with $q_i = 0$. As a result, the transformation from the base frame to the End Effector frame is given by:

$$^{Base}T_{EE} = {}^{Base}T_1^{*\,1}T_2^{*\,2}T_3^{*\,3}T_{EE}$$

where:

$$^{3}T_{EE} = {}^{3}T_4\,{}^{4}T_5\,{}^{5}T_6\,{}^{6}T_{EE}$$

with:

$$^{3}T_{EE} = \begin{bmatrix} 0 & -1 & 0 & d_1 \\ 0 & 0 & -1 & -d_2 \\ 1 & 0 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where d_1 and d_2 are parameters related to the links, as represented in Table2.2 below. The final transformation matrix $^{Base}T_{EE}$ can be expressed as:

$$^{Base}T_{EE} = \left[\begin{array}{ccc|c} & & & X_E \\ & R_E & & Y_E \\ & & & Z_E \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

where:

- R_E is the rotation matrix for the End Effector orientation.
- X_E, Y_E, and Z_E represent the position coordinates of the End Effector.

$$R_E = \begin{bmatrix} -\sin(q_1) & -\cos(q_2 - q_3)\cos(q_1) & -\sin(q_2 - q_3)\cos(q_1) \\ \cos(q_1) & -\cos(q_2 - q_3)\sin(q_1) & -\sin(q_2 - q_3)\sin(q_1) \\ 0 & -\sin(q_2 - q_3) & \cos(q_2 - q_3) \end{bmatrix}$$

and

$$X_E = \cos(q_1)\left(d_1\cos(q_2 - q_3) - d_2\sin(q_2 - q_3) - d_3\sin(q_2)\right) + d_4\sin(q_1) \tag{2.1}$$

$$Y_E = \sin(q_1)\left(d_1\cos(q_2 - q_3) - d_2\sin(q_2 - q_3) - d_3\sin(q_2)\right) - d_4\cos(q_1) \tag{2.2}$$

$$Z_E = d_1\sin(q_2 - q_3) + d_2\cos(q_2 - q_3) + d_3\cos(q_2) + d_5 \tag{2.3}$$

Table 2.2: Definitions of parameters in relation to the links.

Parameter	Relation to Links	Value (m)
d_1	L7+L10	0.057
d_2	L5+L8+L9+L11	0.48
d_3	L4	0.28
d_4	L2-L6	0.01
d_5	L1+L3	0.2435

2.2 Differential Forward Kinematics

2.2.1 Introduction

This section covers **Differential Forward Kinematics**, focusing on the relationship between joint velocities and the resulting linear and angular velocities of the End Effector. Through the **Kinematic Jacobian Matrix**, we map joint movements to the motion of the End Effector in Cartesian space. This matrix is divided into:

- **Linear Velocity Jacobian (J_v)**: linking joint velocities to the End Effector's

linear motion,

- **Angular Velocity Jacobian (J_ω)**: linking joint velocities to its rotational motion.

2.2.2 Linear and Angular Velocity Jacobian

Now that we know the equations of the linear motion of the End Effector frame, we can differentiate them to find the Jacobian Matrix.

$$J_v(q) = \begin{bmatrix} \frac{\partial X_E}{\partial q_1} & \frac{\partial X_E}{\partial q_2} & \frac{\partial X_E}{\partial q_3} \\ \frac{\partial Y_E}{\partial q_1} & \frac{\partial Y_E}{\partial q_2} & \frac{\partial Y_E}{\partial q_3} \\ \frac{\partial Z_E}{\partial q_1} & \frac{\partial Z_E}{\partial q_2} & \frac{\partial Z_E}{\partial q_3} \end{bmatrix}$$

$$\begin{aligned}
\frac{\partial X_E}{\partial q_1} &= \sin(q_1) \cdot (-d_1 \cos(q_2 - q_3) + d_2 \sin(q_2 - q_3) + d_3 \sin(q_2)) + d_4 \cos(q_1) \\
\frac{\partial X_E}{\partial q_2} &= -\cos(q_1) \cdot (d_1 \sin(q_2 - q_3) + d_2 \cos(q_2 - q_3) + d_3 \cos(q_2)) \\
\frac{\partial X_E}{\partial q_3} &= \cos(q_1) \cdot (d_1 \sin(q_2 - q_3) + d_2 \cos(q_2 - q_3)) \\
\frac{\partial Y_E}{\partial q_1} &= \cos(q_1) \cdot (-d_1 \cos(q_2 - q_3) + d_2 \sin(q_2 - q_3) + d_3 \sin(q_2)) + d_4 \sin(q_1) \\
\frac{\partial Y_E}{\partial q_2} &= -\sin(q_1) \cdot (d_1 \sin(q_2 - q_3) + d_2 \cos(q_2 - q_3) + d_3 \cos(q_2)) \\
\frac{\partial Y_E}{\partial q_3} &= \sin(q_1) \cdot (d_1 \sin(q_2 - q_3) + d_2 \cos(q_2 - q_3)) \\
\frac{\partial Z_E}{\partial q_1} &= 0 \\
\frac{\partial Z_E}{\partial q_2} &= d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3) - d_3 \sin(q_2) \\
\frac{\partial Z_E}{\partial q_3} &= -\left(d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3)\right)
\end{aligned}$$

To compute the Angular Velocity Jacobian (J_w), the transformation matrices ($^{Base}T_1$, $^{Base}T_2$, $^{Base}T_3$) are derived to determine the orientation and position of each joint. The Z-axis vectors (z_0, z_1, z_2) are extracted from the rotational components of these matrices, representing the axes of rotation for each joint. Finally, the angular Jaco-

bian is constructed by combining the Z-axis vectors into a matrix as $J_w = [z_1\ z_2\ z_3]$, mapping joint angular velocities to the End Effector angular velocity.

$$^{Base}T_1 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \rho_0 = \begin{bmatrix} 0 \\ 0 \\ L_1 \end{bmatrix}.$$

$$^{Base}T_2 = \begin{bmatrix} \cos(q_1)\cos(q_2) & -\cos(q_1)\sin(q_2) & \sin(q_1) & \rho_{1x} \\ \sin(q_1)\cos(q_2) & -\sin(q_1)\sin(q_2) & -\cos(q_1) & \rho_{1y} \\ \sin(q_2) & \cos(q_2) & 0 & \rho_{1z} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad z_1 = \begin{bmatrix} \sin(q_1) \\ -\cos(q_1) \\ 0 \end{bmatrix},$$

$$\rho_1 = \begin{bmatrix} 0.03\sin(q_1) \\ -0.03\cos(q_1) \\ 0.2435 \end{bmatrix}.$$

$$^{Base}T_3 = \begin{bmatrix} \cos(q_2-q_3)\cos(q_1) & \sin(q_2-q_3)\cos(q_1) & -\sin(q_1) & \rho_{2x} \\ \cos(q_2-q_3)\sin(q_1) & \sin(q_2-q_3)\sin(q_1) & \cos(q_1) & \rho_{2y} \\ \sin(q_2-q_3) & -\cos(q_2-q_3) & 0 & \rho_{2z} \\ 0 & 0 & 0 & 1 \end{bmatrix}, z_2 = \begin{bmatrix} -\sin(q_1) \\ \cos(q_1) \\ 0 \end{bmatrix},$$

$$\rho_2 = \begin{bmatrix} 0.03\sin(q_1) - 0.28\cos(q_1)\sin(q_2) \\ -0.03\cos(q_1) - 0.28\sin(q_1)\sin(q_2) \\ 0.28\cos(q_2) + 0.2435 \end{bmatrix}.$$

Thus, the Angular Velocity Jacobian J_w can be represented as:

$$J_w = \begin{bmatrix} z_0 & z_1 & z_2 \end{bmatrix}$$

Now, the complete Jacobian J is given by:

$$J = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} \frac{\partial X_E}{\partial q_1} & \frac{\partial X_E}{\partial q_2} & \frac{\partial X_E}{\partial q_3} \\ \frac{\partial Y_E}{\partial q_1} & \frac{\partial Y_E}{\partial q_2} & \frac{\partial Y_E}{\partial q_3} \\ \frac{\partial Z_E}{\partial q_1} & \frac{\partial Z_E}{\partial q_2} & \frac{\partial Z_E}{\partial q_3} \\ 0 & \sin(q_1) & -\sin(q_1) \\ 0 & -\cos(q_1) & \cos(q_1) \\ 1 & 0 & 0 \end{bmatrix} \tag{2.4}$$

2.3 Kinematics Validation

To verify the forward and differential kinematics equations we defined in the previous sections, we developed a simulation setup using ROS (we will discuss further in Chapter 5).

Our goal was to compare the real motion data of the robot, such as joint velocities and End Effector Pose with the values computed using our own kinematics equations. The real data was collected through the `/joint_states` topic, and `/tf` transform tree, both provided by the robot's software. By comparing the real and computed values, we were able to evaluate whether our equations accurately describe the robot's actual motion in simulation.

In Figure 2.2, we present a simplified flow diagram of the validation system we built. The flow is explained in more detail in the paragraph below.

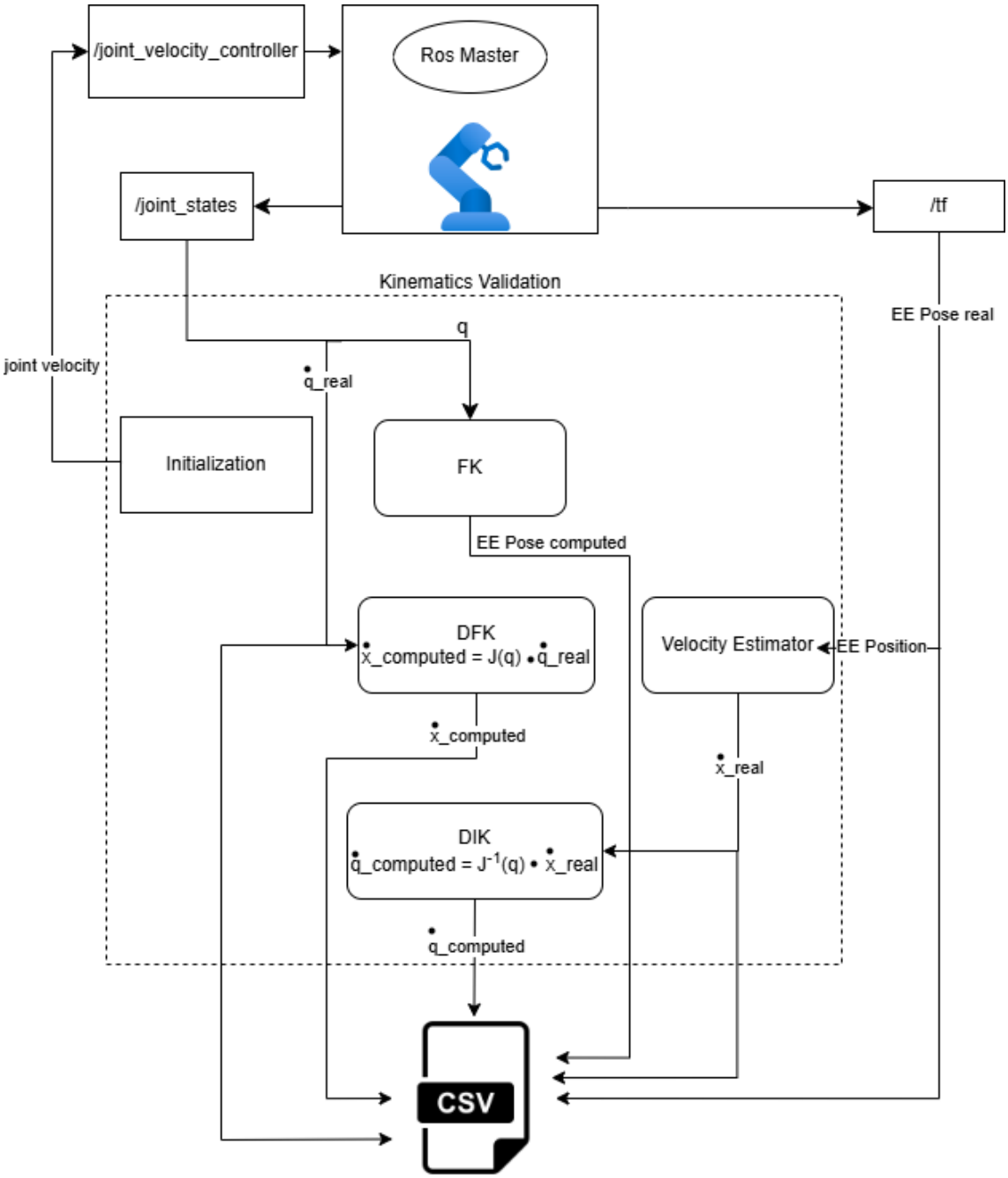


Figure 2.2: Kinematics Validation Flow Chart

The kinematics validation flow is organized into two main parts:

Primary Flow: The process begins with an initialization phase where a predefined constant joint velocity is set for the manipulator. This velocity is continuously published to the `/joint_velocity_controller` topic, allowing the robot to move

at a constant rate throughout the experiment. The simulation runs until the user decides to terminate it.

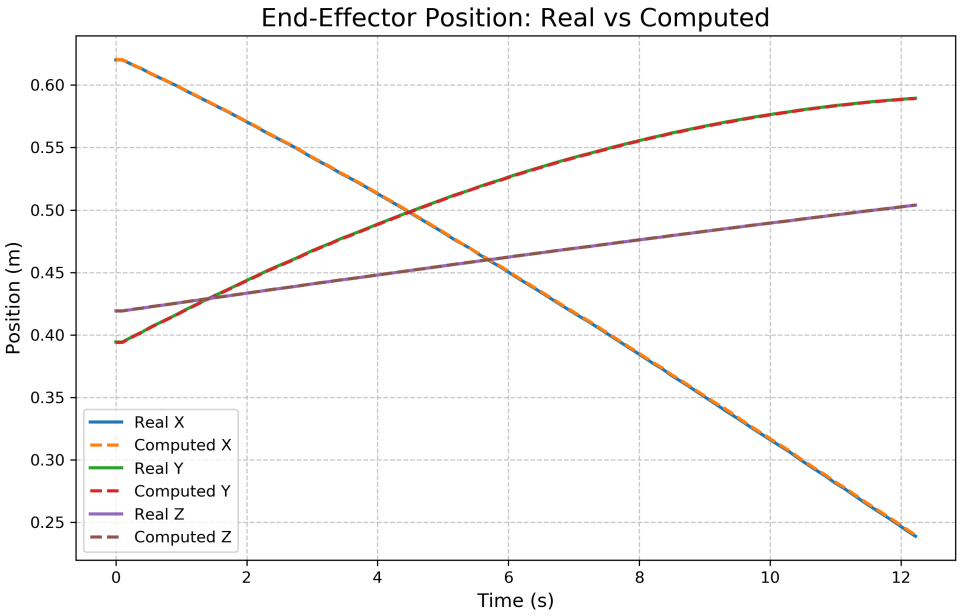
Secondary Flow: While the robot is moving, the kinematics validation node remains active and constantly subscribes to both the `/joint_states` and `/tf` topics. These provide:

- Joint positions and velocities from `/joint_states`
- Real-time End Effector Pose (position and orientation) from the `/tf` transform tree

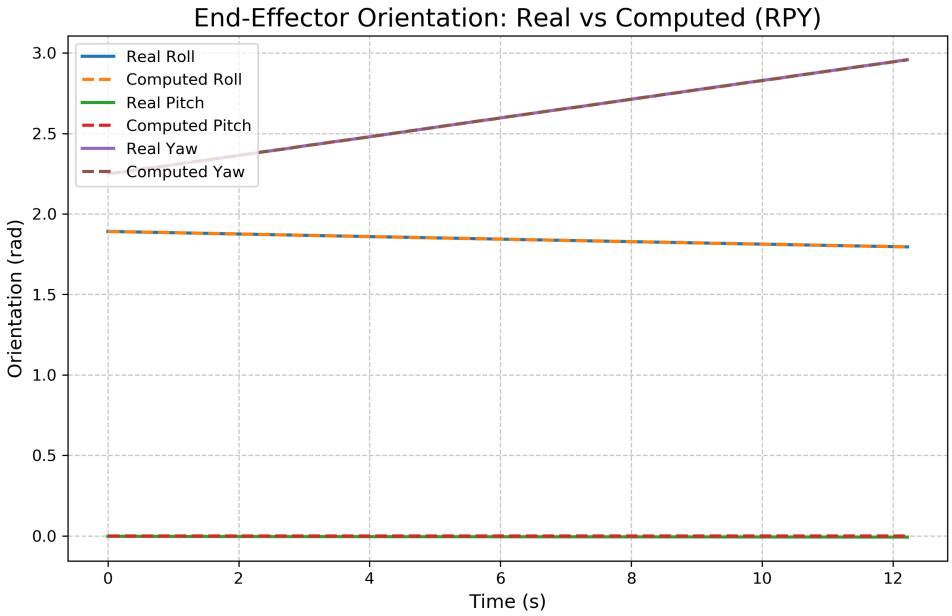
This data serves as input for a series of computations and comparisons:

- **FK (Forward Kinematics):** Calculates the End Effector Pose based on current Joint positions using our forward kinematics equations. This computed pose is compared against the real pose obtained from the `/tf` topic.
- **DFK (Differential Forward Kinematics):** Uses our Jacobian matrix $J(q)$ and joint velocities $\dot{q}_{\text{real}}$ to compute the End Effector velocity $\dot{x}_{\text{computed}}$.
- **Velocity Estimator:** Estimates the real End Effector velocity $\dot{x}_{\text{real}}$ based on changes in position over time. This is used to compare with the velocity computed by DFK.
- **DIK (Differential Inverse Kinematics):** Applies the inverse of the Jacobian $J^{-1}(q)$ and the estimated End Effector velocity to compute the expected joint velocities $\dot{q}_{\text{computed}}$, which are compared to the real joint velocities from the `/joint_states` topic.
- **CSV Logging:** All real and computed values are logged into a CSV file for later analysis.

Finally, we use the logged data to calculate various errors (joint velocity error, End Effector velocity error, End Effector Pose error) and generate plots that visually evaluate the accuracy of our kinematic equations.



(a) End Effector Position



(b) End Effector Orientation

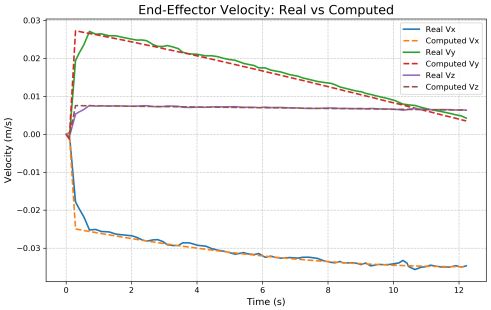
Figure 2.3: End Effector Pose Comparison

Figure 2.3 presents a comparison between the real and computed End Effector Pose data.

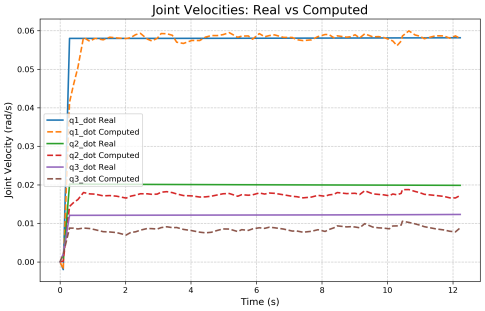
- **(a) End Effector Position:** The real and computed position curves along the X, Y, and Z axes align perfectly. This demonstrates that our Forward

Kinematics (FK) computations accurately reproduce the End Effector's position as obtained from the `/tf` topic in the simulation environment.

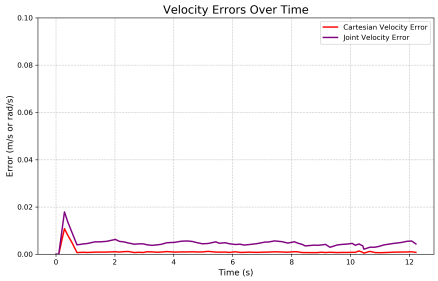
- **(b) End Effector Orientation:** The orientation comparison, based on Roll, Pitch, and Yaw (RPY) values, also shows excellent agreement between the real and computed data.



(a) End Effector Velocity



(b) Joint Velocities



(c) Velocity Errors

Figure 2.4: Velocities Comparison

Figure 2.4 illustrates the comparison of velocities obtained from both real sensor data and our differential kinematics computations.

- **(a) End Effector Velocity:** The computed Cartesian velocity of the End Effector is obtained using the differential forward kinematics (DFK), specifically through the Jacobian $\dot{x}_{\text{computed}} = J(q) \cdot \dot{q}_{\text{real}}$. This is compared with the estimated real End Effector velocity, which is calculated using a velocity estimator based on the time-differentiated End Effector position.

- **(b) Joint Velocities:** In this subplot, we compare the real joint velocities (received from the `/joint_states` topic) with those computed through differ-

ential inverse kinematics (DIK). These are calculated using the inverse of the Jacobian:

$$\dot{q}_{\text{computed}} = J^{-1}(q) \cdot \dot{x}_{\text{real}}$$

The close match between the real and computed joint velocities demonstrates the validity of our DIK implementation.

- **(c) Velocity Errors:** This plot presents the magnitude of the velocity errors over time. The joint velocity error and the Cartesian velocity error are both shown, calculated as the norm of the difference between real and computed values. Specifically:

$$\text{velocity\_error\_joint} = \|\dot{q}_{\text{computed}} - \dot{q}_{\text{real}}\|$$

and

$$\text{velocity\_error} = \|\dot{x}_{\text{computed}} - \dot{x}_{\text{real}}\|$$

The errors remain consistently low (generally below 0.01), confirming the accuracy of our kinematic equations.

2.4 Inverse Kinematics

2.4.1 Introduction

In this section, we explore *Inverse Kinematics*, the process of determining the joints angles required to achieve a desired Pose of the End Effector in space. Unlike forward kinematics, where the position and orientation is calculated based on given joints angles, IK involves solving equations to compute joints values that bring the robot's End Effector to the target Pose.

2.4.2 Derivation of Joints Angles Equations

Deriving Expressions for q_1

To find the equation for the joint angle q_1, we use End Effector position equations X_E 2.1 and Y_E 2.2 based on the forward kinematics.

$$X_E = \cos(q_1) \cdot (d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3) - d_3 \sin(q_2)) + d_4 \sin(q_1)$$

$$Y_E = \sin(q_1) \cdot (d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3) - d_3 \sin(q_2)) - d_4 \cos(q_1)$$

Initially, we rewrite X_E and Y_E as:

$$X_E - d_4 \sin(q_1) = \cos(q_1) \, (d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3) - d_3 \sin(q_2))$$

$$Y_E + d_4 \cos(q_1) = \sin(q_1) \, (d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3) - d_3 \sin(q_2))$$

Then, by dividing both equations, we eliminate the common terms and obtain:

$$\frac{X_E - d_4 \sin(q_1)}{Y_E + d_4 \cos(q_1)} = \frac{\cos(q_1)}{\sin(q_1)}$$

By rearranging terms, we have:

$$X_E \sin(q_1) - d_4 \sin^2(q_1) = Y_E \cos(q_1) + d_4 \cos^2(q_1)$$

$$\Rightarrow \quad X_E \sin(q_1) - Y_E \cos(q_1) = d_4 \sin^2(q_1) + d_4 \cos^2(q_1)$$

$$\Rightarrow \quad X_E \sin(q_1) - Y_E \cos(q_1) = d_4 \cdot \left(\sin^2(q_1) + \cos^2(q_1)\right)$$

$$\Rightarrow \quad X_E \sin(q_1) - Y_E \cos(q_1) = d_4$$

Now, we can express this as:

$$a \sin(q_1) + b \cos(q_1) = c, \quad where \quad a = X_E, \quad b = -Y_E \quad and \quad c = d_4.$$

Using the trigonometric Formula:

$$a\sin(q_1) + b\cos(q_1) = R\sin(q_1 + \varphi)$$

where:

$$\varphi = \arctan\left(\frac{b}{a}\right), \quad R = \sqrt{a^2 + b^2}$$

Thus,

$$R\sin(q_1 + \varphi) = c \Rightarrow \sin(q_1 + \varphi) = \frac{c}{R}$$

Final Solutions:

$$q1 = \begin{cases} q_{1.1} = \sin^{-1}\left(\frac{c}{R}\right) - \varphi \\ q_{1.2} = \pi - \sin^{-1}\left(\frac{c}{R}\right) - \varphi \end{cases} \tag{2.5}$$

Deriving Expressions for q_{23}

Initially, we use Z_E 2.3 to express $\cos(q_2)$:

$$Z_E = d_1\sin(q_2 - q_3) + d_2\cos(q_2 - q_3) + d_3\cos(q_2) + d_5$$

$$\Rightarrow \quad \cos(q_2) = \frac{Z_E - d_1\sin(q_2 - q_3) - d_2\cos(q_2 - q_3) - d_5}{d_3} \tag{2.6}$$

Next, we aim to find an expression for $\sin(q_2)$ by using X_E 2.1 or Y_E 2.2.

- If $\sin(q_1) \neq 0$, we manipulate Y_E as follows:

$$Y_E = \sin(q_1)\cdot(d_1\cos(q_2 - q_3) - d_2\sin(q_2 - q_3) - d_3\sin(q_2)) - d_4\cos(q_1)$$

$$\Rightarrow \quad \frac{Y_E + d_4\cos(q_1)}{\sin(q_1)} = d_1\cos(q_2 - q_3) - d_2\sin(q_2 - q_3) - d_3\sin(q_2)$$

We define:

$$A = \frac{Y_E + d_4 \cos(q_1)}{\sin(q_1)}$$

Rewriting the expression as:

$$A = d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3) - d_3 \sin(q_2)$$

$$\Rightarrow \quad A - d_1 \cos(q_2 - q_3) + d_2 \sin(q_2 - q_3) = -d_3 \sin(q_2)$$

Thus, we have:

$$\sin(q_2) = \frac{A - d_1 \cos(q_2 - q_3) + d_2 \sin(q_2 - q_3)}{-d_3} \tag{2.7}$$

- If $\sin(q_1) = 0$ and $\cos(q_2) \neq 0$, we manipulate X_E as follows:

$$X_E = \cos(q_1) \cdot (d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3) - d_3 \sin(q_2)) + d_4 \sin(q_1)$$

$$\Rightarrow \quad \frac{X_E}{\cos(q_1)} = d_1 \cos(q_2 - q_3) - d_2 \sin(q_2 - q_3) - d_3 \sin(q_2)$$

In this case, we can use expression (2) by setting $A = \frac{X_E}{\cos(q_1)}$.

Since we now know have expressions for both $\cos(q_2)$ 2.6 and $\sin(q_2)$ 2.7, we can use the trigonometric identity $\sin^2(q_2) + \cos^2(q_2) = 1$.

Thus, we have:

$$\sin^2(q_2 - q_3) + \cos^2(q_2 - q_3) = 1$$

$$\Rightarrow \quad \Gamma \sin(q_2 - q_3) + \Delta \cos(q_2 - q_3) = E$$

where:

$$\Gamma = d_2 A - d_1 B$$

$$\Delta = -d_1 A - d_2 B$$

$$E = -A^2 - B^2 - d_1^2 - d_2^2 + d_3$$

$$B = Z_E - d_5$$

We define R and φ as:

$$R = \sqrt{\Gamma^2 + \Delta^2} \quad \text{and} \quad \varphi = \arctan\left(\frac{\Delta}{\Gamma}\right)$$

Using the trigonometric formula, we find:

$$q_{23} = \begin{cases} q_{23.1} = \sin^{-1}\left(\frac{E}{R}\right) - \varphi \\ q_{23.2} = \pi - \sin^{-1}\left(\frac{E}{R}\right) - \varphi \end{cases} \tag{2.8}$$

Deriving Expressions for q_2

We will use Z_E 2.3 equation as follows:

$$Z_E = d_1 \sin(q_2 - q_3) + d_2 \cos(q_2 - q_3) + d_3 \cos(q_2) + d_5$$

$$\Rightarrow \quad \cos(q_2) = \frac{Z_E - d_1 \sin(q_2 - q_3) - d_2 \cos(q_2 - q_3) - d_5}{d_3}$$

Letting:

$$k = \frac{Z_E - d_1 \sin(q_2 - q_3) - d_2 \cos(q_2 - q_3) - d_5}{d_3}$$

The solutions for q_2 are:

$$q_2 = \begin{cases} q_{2.1} = \cos^{-1}(k) \\ q_{2.2} = -\cos^{-1}(k) \end{cases} \tag{2.9}$$

Deriving Expressions for q_3

Using trigonometric identities:

$$\sin(q_2)\sin(q_3) + \cos(q_2)\cos(q_3) = \cos(q_2 - q_3)$$

We define:

$$a = \sin(q_2), \quad b = \cos(q_2), \quad \text{and} \quad c = \cos(q_2 - q_3)$$

Thus, we have:

$$a\sin(q_3) + b\cos(q_3) = c$$

where:

$$\varphi = \arctan\left(\frac{b}{a}\right) \quad \text{and} \quad R = \sqrt{a^2 + b^2} = 1$$

Finally, the solutions for q_3 are:

$$q_3 = \begin{cases} q_{3.1} = \sin^{-1}(c) - \varphi \\ q_{3.2} = \pi - \sin^{-1}(c) - \varphi \end{cases} \tag{2.10}$$

2.4.3 Validation and Observations of Inverse Kinematics

Overview of the Approach

To evaluate the performance of the inverse kinematics equations derived above, we developed and implemented a series of algorithms (detailed in Appendix B.1). These algorithms systematically calculate all possible joint configurations for a given target position, evaluate the reachable workspace, and analyze the manipulator's flexibility.

Observations from Algorithm Application

After implementing the above algorithms, we observed several important insights regarding the reachable workspace of the manipulator and the dexterity of the joints in achieving a desired End Effector position:

1. **Multiple Solutions for a Single Position**

   The manipulator can reach a given target position (X_E, Y_E, Z_E) using up to

four distinct configurations of the joints. This variation arises due to the degrees of freedom in the system, particularly in joints 2 and 3, which allow for two main configurations:

- **Elbow-up** and **elbow-down** configurations, as shown in the accompanying images. These configurations determine whether the arm's elbow is raised above or lowered below the plane formed by the shoulder and the target position.

In addition, Joint 1 contributes additional flexibility in reaching positions by orienting the manipulator in two ways:

- **Face-front configuration**: Joint 1 is oriented at $q_{1.1} = a$, causing the manipulator to approach the target with a forward-facing orientation relative to the base.
- **Face-back configuration**: Joint 1 is oriented at $q_{1.2} = a - \pi$, creating a symmetric orientation of the manipulator relative to the base, while still reaching the same target position.

Algorithm 1 was used to systematically calculate all possible joint configurations for a given target position, enabling us to identify these variations.

2. **Statistical Analysis of the Workspace**

   To evaluate the manipulator's flexibility across the workspace, we applied Algorithm 2 to generate and evaluate **one million target positions** randomly distributed in the reachable space. The results, presented in the table below, provide insights into the number of valid joint configurations for each target position:

Configuration Type	Count	Percentage
Single configuration	26047	2.60%
Two configurations	241959	24.20%
Four configurations	694094	69.41%
No configurations	37900	3.79%

These statistics, analyzed with Algorithm 3, show that the manipulator most commonly has **four valid configurations** for a target position, emphasizing its high flexibility and adaptability. The small percentage of no solutions indicates unreachable positions due to physical constraints or joint limits. It is important to note that the range of Joint 1 is restricted and does not span 2π, meaning that for some angle values $q_{1.1} = a$, the symmetric face-back configuration $q_{1.2} = a - \pi$ may not exist.

3. **Example of Four Distinct Configurations**

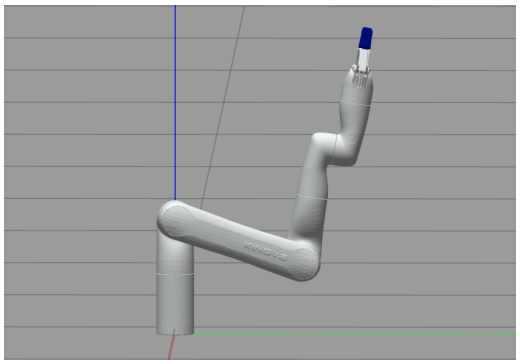
As demonstrated in the example below, for the target position:

$$X_E = 0.1000, \quad Y_E = 0.4000, \quad Z_E = 0.8000$$

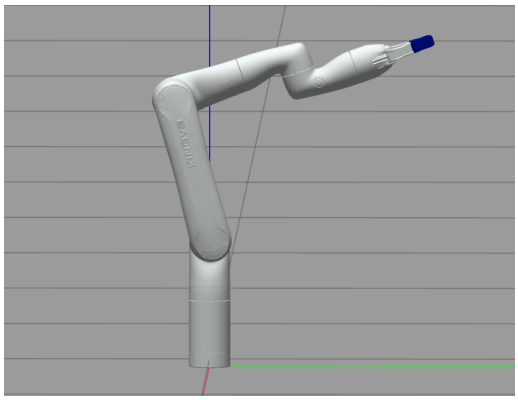
The manipulator can reach this position using the following four joint configurations:

$$\begin{aligned}
&\text{Configuration 1:} \quad q_1 = 1.6457,\ q_2 = -1.9027,\ q_3 = -1.7365 \\
&\text{Configuration 2:} \quad q_1 = 1.6457,\ q_2 = 0.2604,\ q_3 = 1.5001 \\
&\text{Configuration 3:} \quad q_1 = -1.5458,\ q_2 = -0.2604,\ q_3 = -1.7365 \\
&\text{Configuration 4:} \quad q_1 = -1.5458,\ q_2 = 1.9027,\ q_3 = 1.5001
\end{aligned}$$

These configurations correspond to combinations of **elbow-up, elbow-down, face-front, and face-back** positions of the joints. Images visualizing these configurations are provided below.

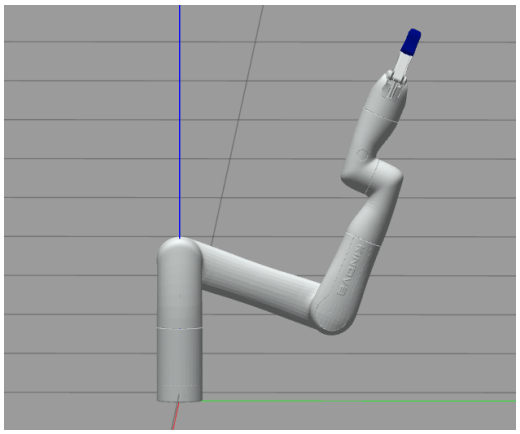


(a) Face Front Elbow Down Configuration

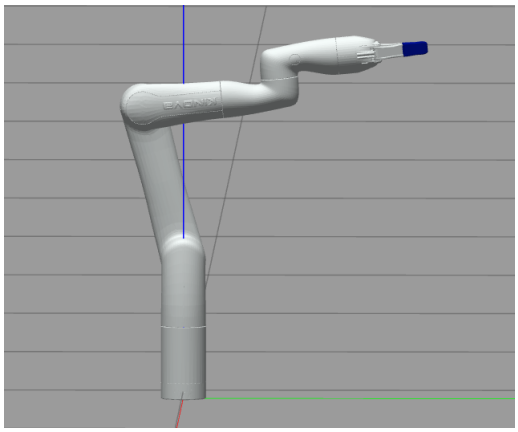


(b) Face Front Elbow Up Configuration

Figure 2.5: Comparison of Face Front Configurations



(a) Face Back Elbow Down Configuration

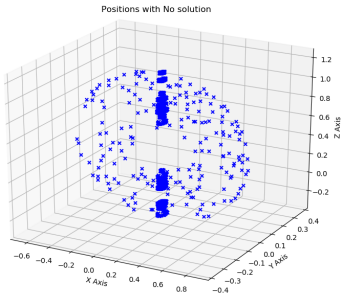


(b) Face Front Elbow Up Configuration

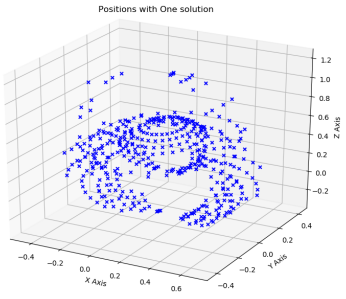
Figure 2.6: Comparison of Face Back Configurations

4. **Visualization of the Reachable Workspace and Configurations**

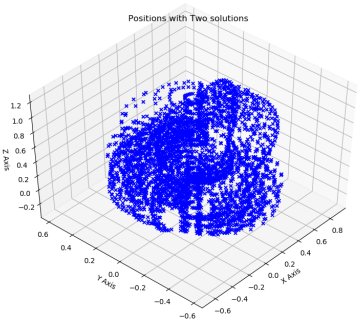
   Based on the results obtained from Algorithm 3, we generated visualizations of the reachable workspace. These plots illustrate the distribution of reachable positions in the workspace along with their corresponding configuration counts. Specifically, the following categories are depicted in the images below:



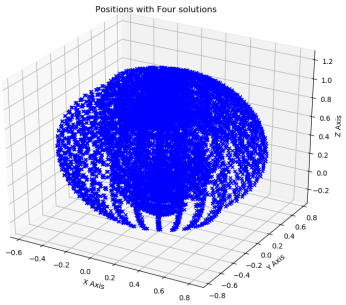
(a) No Configuration



(b) One Configuration



(c) Two Configurations



(d) Four Configurations

Figure 2.7: Visualization of Configuration Categories.