ENPM662 INTRODUCTION TO ROBOT MODELING

PROJECT REPORT

HUMAN SUPPORT ROBOT(HSR)

BALAJI SELVAKUMAR **118545745**

VYSHNAV ACHUTHAN **119304815**

Acknowledgement

We express our heartfelt gratitude to respected Dr. Reza Monafredi for providing us with the opportunity to work on the final project. We also thank him for his support, guidance and the knowledge imparted through lectures which enabled us to come with this idea and further work on it. We extend our deep gratitude to Teaching Assistants Mr. Adarsh Malapaka and Mr. Pavan Mandtripragada, who helped us through all the problems and errors we faced while working on the project and, their valuable suggestions to enhance the quality of the project. Without their support this project would not have taken its present shape.

Balaji Selvakumar

Vyshnav Achuthan

Table of Contents

Serial Number	Topics	Page Number
1	Introduction	**4**
2	Application	**5**
3	Robot Description	**6**
4	CAD Model	**7**
5	Denavit-Hartenberg Representation	**8**
6	Forward Kinematics	**10**
7	Forward Kinematics Validation	**12**
8	Inverse Kinematics	**14**
9	Inverse Kinematics Validation	**18**
10	Workspace Study	**22**
11	Assumptions	**24**
12	Control Method	**24**
13	Gazebo & RViz Visualization	**25**
14	Problems Faced	**28**
15	Lessons Learned	**29**
16	Conclusion	**29**
17	References	**30**

Introduction

There has been an increasing interest in mobile manipulators that can perform physical work in living spaces worldwide corresponding to the aging population with reduction in birthrates with the expectation of improving quality of life. As society's aging rapidly advances, the shortage of workers and care workers has become a major issue. This project aims to augment the caregiving industry by designing and simulating a support system which helps the dependents in their day-to-day activities. As per CDC, 13.7 % of the population in the US has mobility disability and this number is only going to double by 2050, as WHO cites that the world's population of people over 60 years is anticipated to double by 2050. Th Human Support Robot (HSR) we proposed, is a domestic mobile manipulator robot which holds functions of physical work. This project is inspired from the HUMAN SUPPORT ROBOT by Toyota Motors which helps in day to day tasks such as fetching and carrying of daily necessities and tidying rooms up. Hence, we designed a robotic system that will address these issues.

The robot consists of a mobile base in which a manipulator is attached. The manipulator we chose is a FANUC arm with a two-finger end effector. The mobile base has 2 degrees of freedom, and the manipulator has 6 degrees of freedom. We simulated the system in a custom world environment where it can pickup the object from one table and place it in the other table. There can be multiple real-life scenarios which can be emulated by the robot model.

The report is organized in a way that it goes over every aspect of the project in a chronological order. In the first section, we discuss the applications of the support system, which is followed by the description of the robot which mentions the type of robot and specifications of the robot like the degrees of freedom and dimensions. The third section mentions the key factors in the CAD modelling steps and the procedures involved in it. In forward kinematics, we have created the Denavit Hartenburg table using the Spong convention and then finally calculated the final transformation matrix. The inverse kinematics includes the calculation of the Jacobian and Jacobian inverse matrices, which is used to find joint velocities and given an initial configuration. This section is closely followed by the validation of both forward and inverse kinematics. Then, we refer to the workspace of the robot system which is augmented with screenshots from the gazebo visualization snapshots. The assumptions taken throughout the study are stated explicitly in the following section. In the tenth section, we talk about the control method used in the system. We follow through with the actual gazebo simulation and the observations made from it. We also include the problems faced by us throughout the project and the learning outcomes from it. The report ends with the conclusion of the report and the future works. The references used throughout the project work are cited at the end of the document

Application

Human Support Robot is a capable system designed to help and assist humans with day-to-day activities. As mentioned in the previous section, certain aspects of taking care of elderly people are something that needs to be solved while considering specific scenarios such as healthcare and performing daily domestic tasks.

The application of the robot system can be consolidated as follows

- Elder care **:** Human support robots can be used to help care for elderly and disabled people, providing assistance with mobility, monitoring health, and providing companionship.
- Domestic Assistance**:** It can also be used to help with mundane household tasks such as laundry, vacuuming, and other basic chores.
- Assistance with disabilities**:** It can also be used to help people with disabilities carry out everyday tasks such as opening door and reaching items.

The field of robotics is at its peak and there are numerous possibilities in every application, especially when we all are going facing these tough times(pandemic), we believe we should try and augment our health sector with the advent of technology.

HSR can be implemented autonomously as well as remotely which is much useful when it comes to assisting people with disabilities

Robot Description

The HSR consists of a mobile robot base with 2 DOF and a FANUC manipulator (commercially sold by FANUC Manufacturing. It has a 6 DOF arm with a custom build two finger End effector.

The robot base has 4 wheels and we have used skid steer drive to control the wheel movements.

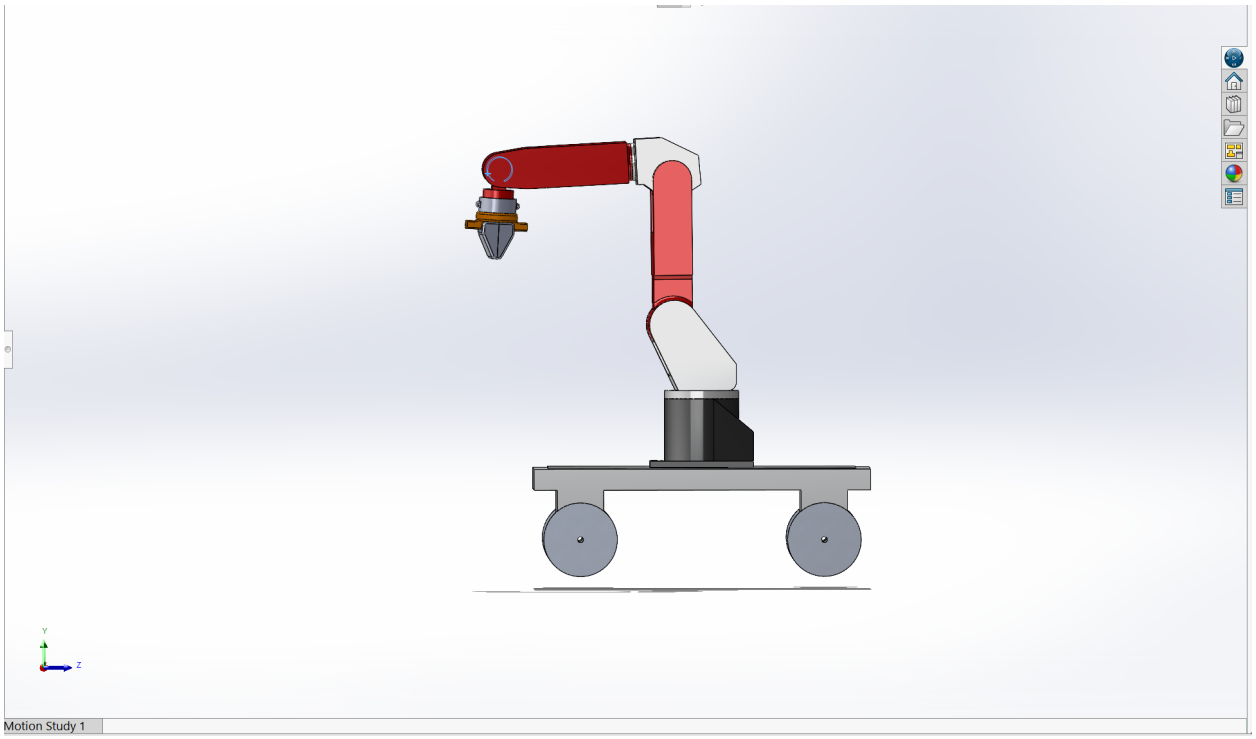
The manipulator has 5 revolute joints to provide a wide range of motion. The end effector is a two finger gripper with prismatic joint which is used to pick and place the objects.

The following table lists the robot specifications:

Degrees of Freedom	8
Mass	56.73 KG
Dimensions [LxBxH]	1450.22 x 621.14 x 1578
Number of Links	8
Number of Joints	6
End effector Velocity	2m/s
Payload	10 kgs
Applications	Domestic and hospital health care
Maximum Joint Angular Displacement(varies as per link)	360 degrees

CAD MODEL

The robot assembly has two main components. The mobile base consists of 5 parts which are used to create a body and 4 wheels. The manipulator consists of 9 parts which are used to create 9 links joined by 7 joints. The system also has a two-finger gripper as end effector. All joints in the arm are revolute joints. The end effector has a prismatic joint. Appropriate joint limits were specified to avoid joint locking. The assembly process took a while as the regular mates were not enough for the assembly. Advanced mates such as profile center was used to achieve the correct functionality of the system. The base link was exported with a reference coordinate system which was aligned according to gazebo coordinate system.



Snapshot from Solid works assembly file

Denavit-Hartenberg Representation:

The axes have been designed in a way that the z-axis is aligned along the direction of motion and the x-axis is towards the front of the robot. This would allow us to easily extract DH parameters of the manipulator through either Sponge or Craig convention. The axes naming starts from base and has 6 axes moving upward. The base axis is called Z base and the end effector axis is called Z Tool axis.

DH Table:

i	θ	α	a	d
1	Θ_1	90	69.55	342.96
2	90 + θ_2	0	350.5	0
3	Θ_3	90	47.38	0
4	Θ_4	270	0	410.45
5	270+θ_5	90	0	0
6	Θ_6	0	0	127

$\boldsymbol{a}$i : link length

$\boldsymbol{di}$: link offset.

$\boldsymbol{\alpha i}$: twist of the link.

$\boldsymbol{\theta i}$ **:** joint angle.

Now the above table shows that the 6 joints have the above base configuration. Then the values are substituted into transformation matrices and multiplied to produce the following matrix which is the transformation matrix from the base to the end effector. The basic rules followed to construct the DH parameter table are as follows:

1) θ n is said to be the rotation along Z_{n-1} to make X_{n-1} point and is in the same direction as X_n
2) α is the rotation alone X_n to make Z_{n-1} point in the same direction as Z n

3) a is the displacement along X_n
4) d is the displacement alone Z_{n-1}

The DH table tells us that most of the joints have a different axes than the ones either preceding or the next joints and yet still have their axes in the direction similar to other joints.

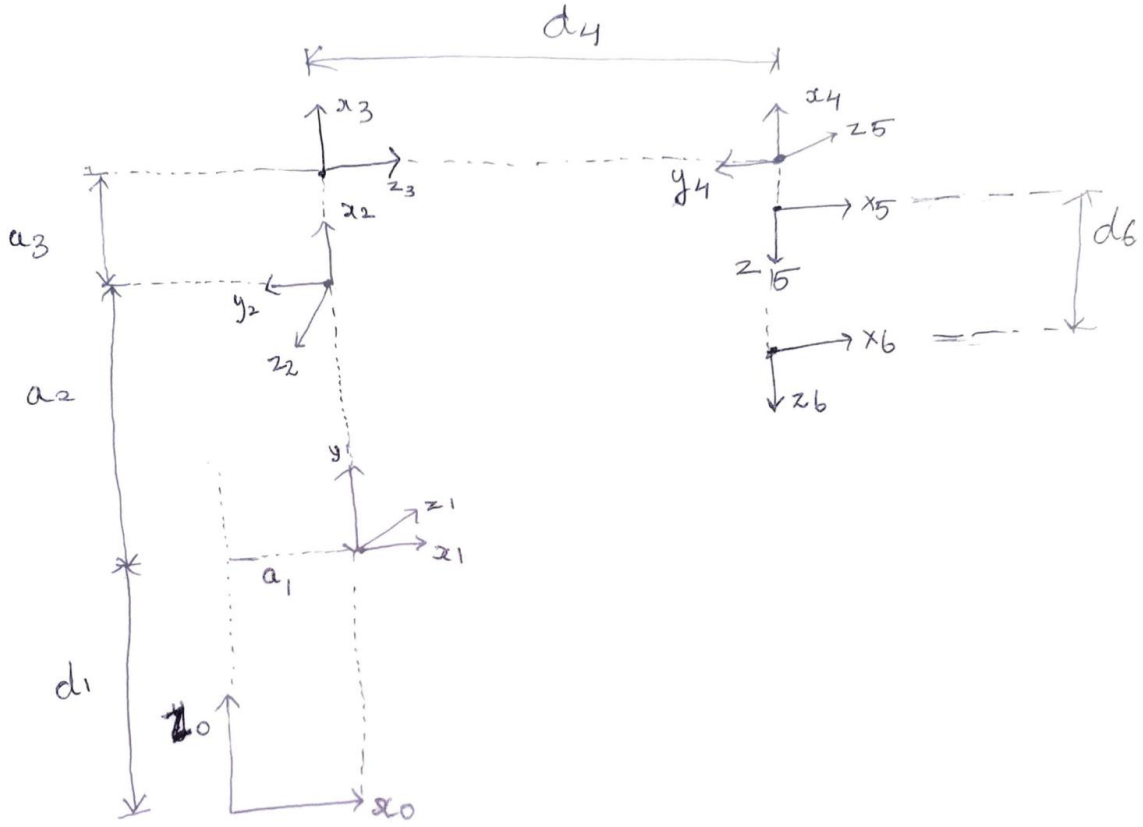


Fig : DH Diagram at zero position

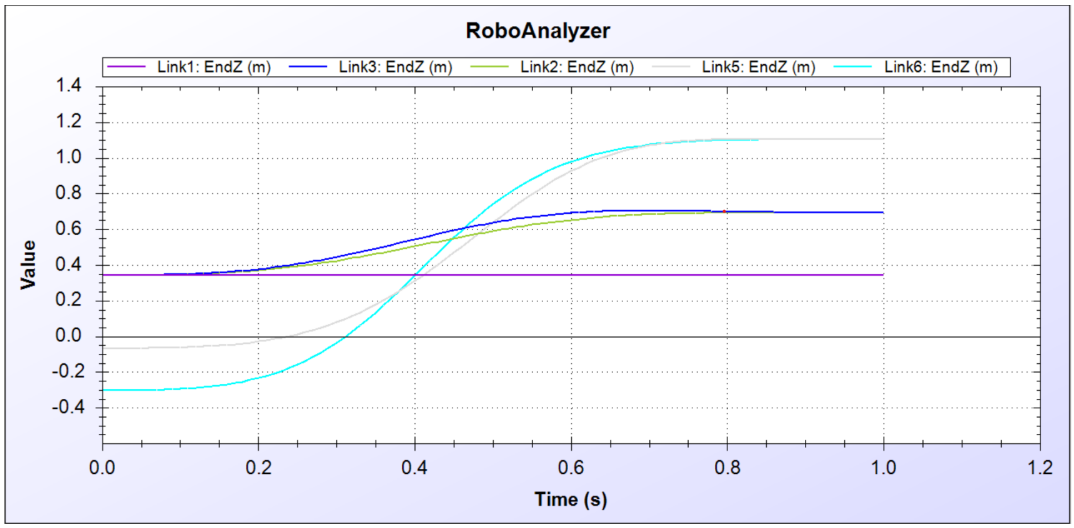


Fig: Analysis of End Effector position with respect to individual links

Forward Kinematics:

Forward kinematics is one of the main kinematic constraints used to analyze robot motion. Given the robot position, we can use forward kinematics to verify our robot position. Since we had used velocity controllers due to some problem with effort controllers in the package and then to reach these positions the limits to the revolute joints were given what were the same as the theta inputs to the Transformation matrix. This would mean that the joints would essentially be in the same position as they would be had we used effort controllers and with this we had tried multiple different configurations for the manipulator and then geometrically checked the end effector positions to see if they matched. To verify our forward kinematics, we need to follow three steps in total . We first assign frames to all 6 dof of the manipulator excluding the end effector using the conventions followed in Mark W Spong book :

- Each X_n axis is always perpendicular to, and intersects with Z_{n-1} axis
- Revolute joints always rotate around the Z_n axis, while prismatic joints move along the Z_n axis.

Upon calculating the frame , we then tabulate the DH table constructed as per the rules.The final step is to generate the final transformation matrix which will give us the pose and orientation of the end effector. First we need to calculate individual transformation matrix for each of the joints by the DH table values in the transformation matrix below

$$\begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using the generic representation, we now calculate the transformation matrix for all the links from base to end effector

$$^{0}_{1}T = \begin{pmatrix} c_{\theta_1} & -s_{\theta_1} & 0 & a_0 \\ s_{\theta_1}c_{\alpha_0} & c_{\theta_1}c_{\alpha_{01}} & -s_{\alpha_0} & -d_1 s_{\alpha_0} \\ s_{\theta_1}s_{\alpha_0} & c_{\theta_1}s_{\alpha_0} & c_{\alpha_0} & d_1 c_{\alpha_0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad ^{0}_{1}T = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

from base to second link, the transformation matrix is given by

$$^{1}_{2}T = \begin{pmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & a_1 \\ s_{\theta_2}c_{\alpha_1} & c_{\theta_2}c_{\alpha_1} & -s_{\alpha_1} & -d_2 s_{\alpha_1} \\ s_{\theta_2}s_{\alpha_1} & c_{\theta_2}s_{\alpha_1} & c_{\alpha_1} & d_2 c_{\alpha_1} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad ^{01}_{2}T = \begin{pmatrix} c_2 & -s_2 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

repeating the same steps till T6 we the following transformation matrix:

$$^{2}_{3}T = \begin{pmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & a_2 \\ s_{\theta_3}c_{\alpha_2} & c_{\theta_3}c_{\alpha_2} & -s_{\alpha_2} & -d_3 s_{\alpha_2} \\ s_{\theta_3}s_{\alpha_2} & c_{\theta_3}s_{\alpha_2} & c_{\alpha_2} & d_3 c_{\alpha_2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad ^{2}_{3}T = \begin{pmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{3}_{4}T = \begin{pmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & a_3 \\ s_{\theta_4}c_{\alpha_3} & c_{\theta_4}c_{\alpha_3} & -s_{\alpha_3} & -d_4 s_{\alpha_3} \\ s_{\theta_4}s_{\alpha_3} & c_{\theta_4}s_{\alpha_3} & c_{\alpha_3} & d_4 c_{\alpha_3} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad ^{3}_{4}T = \begin{pmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{4}_{5}T = \begin{pmatrix} c_{\theta_5} & -s_{\theta_5} & 0 & a_4 \\ s_{\theta_5}c_{\alpha_4} & c_{\theta_5}c_{\alpha_4} & -s_{\alpha_4} & -d_5 s_{\alpha_4} \\ s_{\theta_5}s_{\alpha_4} & c_{\theta_5}s_{\alpha_4} & c_{\alpha_4} & d_5 c_{\alpha_4} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad ^{4}_{5}T = \begin{pmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{5}_{6}T = \begin{pmatrix} c_{\theta_6} & -s_{\theta_6} & 0 & a_5 \\ s_{\theta_6}c_{\alpha_5} & c_{\theta_6}c_{\alpha_5} & -s_{\alpha_5} & -d_6 s_{\alpha_5} \\ s_{\theta_6}s_{\alpha_5} & c_{\theta_5}s_{\alpha_5} & c_{\alpha_5} & d_6 c_{\alpha_5} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad ^{5}_{6}T = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

To find the homogeneous matrix of the end effector with respect to the base, we multiply all the transformation matrices in order . This is given by

$$^{0}_{2}T = {}^{0}_{1}T \text{ X } {}^{1}_{2}T$$

$$^{0}_{2}T = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{X} \begin{pmatrix} c_2 & -s_2 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_1c_2 & -c_1s_2 & -s_1 & c_1a_1 \\ s_1c_2 & -s_1s_2 & c_1 & s_1a_1 \\ -s_2 & -c_2 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{0}_{3}T = {}^{0}_{2}T \text{ X } {}^{2}_{3}T$$

$$^{0}_{3}T = \begin{pmatrix} c_1c_2 & -c_1s_2 & -s_1 & c_1a_1 \\ s_1c_2 & -s_1s_2 & c_1 & s_1a_1 \\ -s_2 & -c_2 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{0}_{3}T = \begin{pmatrix} c_1c_2c_3 - c_1s_2s_3 & -(c_1c_2s_3 + c_1s_2c_3) & -s_1 & c_1c_2a_2 + c_1a_1) \\ s_1c_2c_3 - s_1s_2s_3 & -(s_1c_2s_3 + s_1s_2c_3) & c_1 & s_1c_2a_2 + s_1a_1) \\ -(s_2c_3 + c_2s_3) & s_2s_3 - c_2c_3 & 0 & -s_2a_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{0}_{3}T = \begin{pmatrix} c_1c_{23} & -c_1s_{23} & -s_1 & c_1(c_2a_2 + a_1) \\ s_1c_{23} & -s_1s_{23} & c_1 & s_1(c_2a_2 + a_1) \\ -s_{23} & -c_{23} & 0 & -s_2a_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{4}_{6}T = {}^{4}_{5}T \text{ X } {}^{5}_{6}T$$

$$^{4}_{6}T = \begin{pmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_5c_6 & -c_5s_6 & -s_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ s_5\,c_6 & -s_5\,s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{3}_{6}T = {}^{3}_{4}T \text{ X } {}^{4}_{6}T$$

$$^{3}_{6}T = \begin{pmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_5c_6 & -c_5s_6 & -s_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ s_5\,c_6 & -s_5\,s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{3}_{6}T = \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & 0 \\ s_5\,c_6 & -s_5\,s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{0}_{6}T = {}^{0}_{3}T \text{ X } {}^{3}_{6}T$$

$$^{0}_{6}T = \begin{pmatrix} c_1c_{23} & -c_1s_{23} & -s_1 & c_1(c_2a_2 + a_1) \\ s_1c_{23} & -s_1s_{23} & c_1 & s_1(c_2a_2 + a_1) \\ -s_{23} & -c_{23} & 0 & -s_2a_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & 0 \\ s_5\,c_6 & -s_5\,s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Upon getting the transformation matrix, we can use this to find and verify the end effector position for a known value and position.

Forward Kinematics Validation:

The geometric location of the end effector aligned with the coordinates obtained in the Transformation matrix for different orientations given by respective values of theta angles. For our verification, we used one pose, that's our base pose of the robot where the joint angles are zero with respect to each other. We were then able to confirm that the Transformation Matrix obtained by the dh table gives us the correct coordinates to the assembled model.

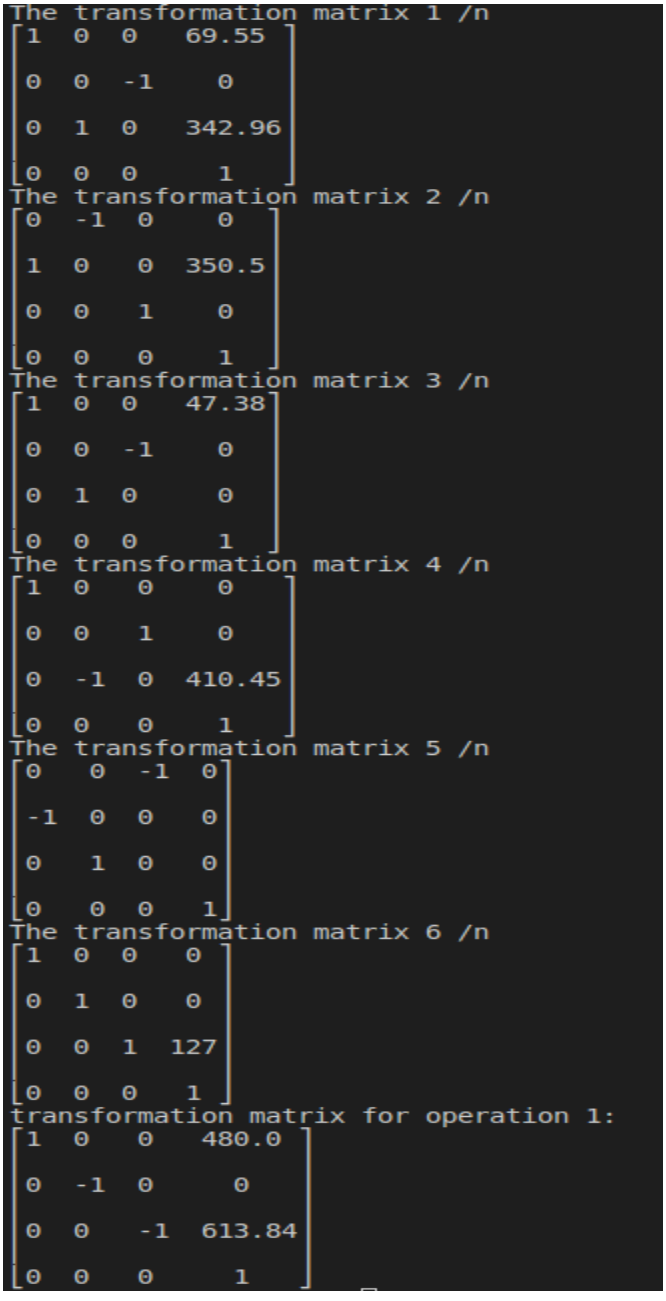


Fig: Forward Kinematics using Manual Coding

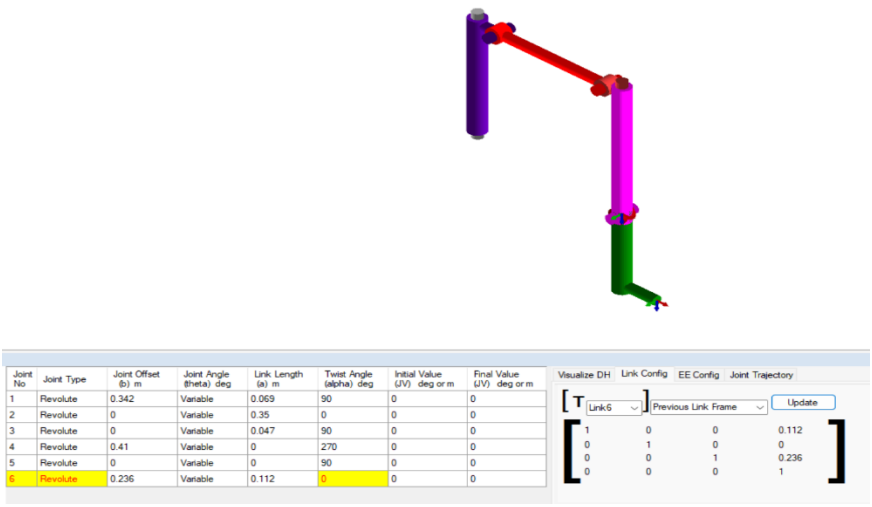


Fig: Verification of the Forward Kinematics using ReDySim

Upon using hardcoding, we found the individual transformation matrix to be matching with the forward kinematics we derived upon simulating our model using the ReDySim algorithm (RoboAnalyzer). We fed the dh params and the angle to the solver and obtained coordinates of the end effector, which matched with the value we obtained via the hardcode solver.

Inverse kinematics:

Inverse kinematics can be used to predict how a robot will move in order to arrive at a specific place. We define inverse kinematics equations for the same to understand the angles the robot takes to reach a particular position. Inverse kinematics can be solved using both analytical methods and numerical methods. For our 6 DOF robot, we have taken the approach of solving the inverse kinematics problem using a numerical method.

The simulations run in the Moveit package uses KDL solver , which is an analytical solver method. To compute inverse kinematics, we need to first compute the Jacobian matrix. The Jacobian Matrix can be computed using two methods. The Jacobian matrix efficiently transfers the rate of change of joint angles to the rate of change of the actual location of the end effector through the gradient of a vector-valued function. The Jacobian maps the change in vector value in the joint space $\Delta\theta$, using the formula :

$$\Delta r = J\Delta\theta$$

where J is the Jacobian Matrix.

$$J_{v_i} = \begin{cases} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{cases}$$

where the first representation is used for revolute joints and the second is used for prismatic joints. Now, since all our joints are revolute in nature, we will use the first notation to calculate Jacobian at the ith instance.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \cdots & \frac{\partial x}{\partial \theta_n} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \cdots & \frac{\partial y}{\partial \theta_n} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \cdots & \frac{\partial z}{\partial \theta_n} \end{bmatrix}$$

As the Jacobian is the differentiation of x with respect to joint angles theta, we need to calculate z matrix and differentiate with respect to input theta joints. Z matrix can be calculated using the transformation matrix as follows :

$$^{n}_{0}T = \begin{bmatrix} ^{n}_{0}R & & ^{n}_{0}P \\ 0 \quad 0 & 0 & 1 \end{bmatrix}$$

Where P is the origin of the end effector and P can be given as :

$$h(q_1, q_2, \ldots, q_n) = X_p(q_1, q_2, \ldots, q_n)$$

From this ,we can see that we can take the last row values of the final homogenous matrix that we generate from the joints :

$$^{n}_{0}T = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

we then use the xyz values to construct a matrix by using the position values :

$$z = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$ {}^{0}_{n}J = \begin{bmatrix} \frac{\partial {}^{0}X_p}{\partial q_1} & \cdots & \frac{\partial {}^{0}X_p}{\partial q_i} & \cdots & \frac{\partial {}^{0}X_p}{\partial q_n} \\ {}^{0}Z_1 & \cdots & {}^{0}Z_i & \cdots & {}^{0}Z_n \end{bmatrix} $$

Now on tabulating the values for our robot configuration, we obtain the following values as Jacobian Matrix

$$0.127\left(\left(-\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)-\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\cos\left(\theta_4\right)-\sin\left(\theta_1\right)\sin\left(\theta_2\right)\sin\left(\theta_4\right)\right)\sin\left(\theta_5\right)$$
$$-0.127\left(\sin\left(\theta_1\right)\sin\left(\theta_3\right)\cos\left(\theta_2\right)-\cos\left(\theta_1\right)\cos\left(\theta_3\right)\right)\cos\left(\theta_5\right)+0.4104\sin\left(\theta_1\right)\sin\left(\theta_3\right)\cos\left(\theta_2\right)-0.047\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)-0.35\sin$$
$$\left(\theta_1\right)\cos\left(\theta_2\right)-0.069\sin\left(\theta_1\right)-0.047\sin\left(\theta_3\right)\cos\left(\theta_1\right)-0.4104\cos\left(\theta_1\right)\cos\left(\theta_3\right)$$
$$0.127\left(\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_1\right)\right)\sin\left(\theta_5\right)$$
$$-0.127\left(-\sin\left(\theta_1\right)\cos\left(\theta_3\right)-\sin\left(\theta_3\right)\cos\left(\theta_1\right)\cos\left(\theta_2\right)\right)\cos\left(\theta_5\right)-0.047\sin\left(\theta_1\right)\sin\left(\theta_3\right)-0.4104\sin\left(\theta_1\right)\cos\left(\theta_3\right)-0.4104\sin\left(\theta_3\right)\cos$$
$$\left(\theta_1\right)\cos\left(\theta_2\right)+0.047\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+0.35\cos\left(\theta_1\right)\cos\left(\theta_2\right)+0.069\cos\left(\theta_1\right)$$
$$0$$
$$-\sin\left(\theta_2\right)\cos\left(\theta_1\right)$$
$$-\sin\left(\theta_1\right)\sin\left(\theta_2\right)$$
$$\cos\left(\theta_2\right)$$

Fig : Column 1 of the Jacobian Matrix

The Jacobian Matrix

$$0.127\left(-\sin\left(\theta_2\right)\cos\left(\theta_1\right)\cos\left(\theta_3\right)\cos\left(\theta_4\right)+\sin\left(\theta_4\right)\cos\left(\theta_1\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_5\right)-0.127\sin\left(\theta_2\right)\sin\left(\theta_3\right)\cos\left(\theta_1\right)\cos\left(\theta_5\right)+0.4104\sin\left(\theta_2\right)\sin$$
$$\left(\theta_3\right)\cos\left(\theta_1\right)-0.047\sin\left(\theta_2\right)\cos\left(\theta_1\right)\cos\left(\theta_3\right)-0.35\sin\left(\theta_2\right)\cos\left(\theta_1\right)$$

$$0.127\left(-\sin\left(\theta_1\right)\sin\left(\theta_2\right)\cos\left(\theta_3\right)\cos\left(\theta_4\right)+\sin\left(\theta_1\right)\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_5\right)-0.127\sin\left(\theta_1\right)\sin\left(\theta_2\right)\sin\left(\theta_3\right)\cos\left(\theta_5\right)+0.4104\sin\left(\theta_1\right)\sin$$
$$\left(\theta_2\right)\sin\left(\theta_3\right)-0.047\sin\left(\theta_1\right)\sin\left(\theta_2\right)\cos\left(\theta_3\right)-0.35\sin\left(\theta_1\right)\sin\left(\theta_2\right)$$

$$.127\left(\sin\left(\theta_2\right)\sin\left(\theta_4\right)+\cos\left(\theta_2\right)\cos\left(\theta_3\right)\cos\left(\theta_4\right)\right)\sin\left(\theta_5\right)+0.127\sin\left(\theta_3\right)\cos\left(\theta_2\right)\cos\left(\theta_5\right)-0.4104\sin\left(\theta_3\right)\cos\left(\theta_2\right)+0.047\cos\left(\theta_2\right)\cos\left(\theta_3\right)$$
$$+0.35\cos\left(\theta_2\right)$$
$$-\sin\left(\theta_1\right)\cos\left(\theta_3\right)-\sin\left(\theta_3\right)\cos\left(\theta_1\right)\cos\left(\theta_2\right)$$
$$-\sin\left(\theta_1\right)\sin\left(\theta_3\right)\cos\left(\theta_2\right)+\cos\left(\theta_1\right)\cos\left(\theta_3\right)$$
$$-\sin\left(\theta_2\right)\sin\left(\theta_3\right)$$

Fig : Column 2 of the Jacobian Matrix

The Jacobian Matrix

$$0.127\left(-\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\sin\left(\theta_4\right)+\sin\left(\theta_2\right)\cos\left(\theta_1\right)\cos\left(\theta_4\right)\right)\sin\left(\theta_5\right)$$
$$0.127\left(-\left(\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\sin\left(\theta_4\right)+\sin\left(\theta_1\right)\sin\left(\theta_2\right)\cos\left(\theta_4\right)\right)\sin\left(\theta_5\right)$$
$$0.127\left(-\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_3\right)-\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right)\sin\left(\theta_5\right)$$
$$\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\sin\left(\theta_4\right)-\sin\left(\theta_2\right)\cos\left(\theta_1\right)\cos\left(\theta_4\right)$$
$$\left(\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\sin\left(\theta_4\right)-\sin\left(\theta_1\right)\sin\left(\theta_2\right)\cos\left(\theta_4\right)$$
$$\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_3\right)+\cos\left(\theta_2\right)\cos\left(\theta_4\right)$$

Fig : Column 3 of the Jacobian Matrix

The Jacobian Matrix

$$0.127\left(\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_1\right)\right)\cos\left(\theta_5\right)$$
$$+0.127\left(-\sin\left(\theta_1\right)\cos\left(\theta_3\right)-\sin\left(\theta_3\right)\cos\left(\theta_1\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_5\right)$$

$$0.127\left(\left(\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_1\right)\sin\left(\theta_2\right)\sin\left(\theta_4\right)\right)\cos\left(\theta_5\right)$$
$$+0.127\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)\cos\left(\theta_2\right)+\cos\left(\theta_1\right)\cos\left(\theta_3\right)\right)\sin\left(\theta_5\right)$$

$$0.127\left(\sin\left(\theta_2\right)\cos\left(\theta_3\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\cos\left(\theta_5\right)-0.127\sin\left(\theta_2\right)\sin\left(\theta_3\right)\sin\left(\theta_5\right)$$

$$\left(\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_1\right)\right)\sin\left(\theta_5\right)$$
$$-\left(-\sin\left(\theta_1\right)\cos\left(\theta_3\right)-\sin\left(\theta_3\right)\cos\left(\theta_1\right)\cos\left(\theta_2\right)\right)\cos\left(\theta_5\right)$$
$$\left(\left(\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_1\right)\sin\left(\theta_2\right)\sin\left(\theta_4\right)\right)\sin\left(\theta_5\right)-\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)\cos\left(\theta_2\right)+\cos\left(\theta_1\right)\cos\left(\theta_3\right)\right)\cos$$
$$\left(\theta_5\right)$$
$$\left(\sin\left(\theta_2\right)\cos\left(\theta_3\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_5\right)+\sin\left(\theta_2\right)\sin\left(\theta_3\right)\cos\left(\theta_5\right)$$

Fig : Column 4 of the Jacobian Matrix

The Jacobian Matrix

$$0$$

$$0$$

$$0$$

$$-\left(\left(\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_1\right)\right)\cos\left(\theta_5\right)+\left(-\sin\left(\theta_1\right)\cos\left(\theta_3\right)-\sin\left(\theta_3\right)\cos\left(\theta_1\right)\cos\left(\theta_2\right)\right)\right.$$
$$\left.\left(\theta_6\right)+\left(\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\sin\left(\theta_4\right)-\sin\left(\theta_2\right)\cos\left(\theta_1\right)\cos\left(\theta_4\right)\right)\cos\left(\theta_6\right)\right.$$
$$-\left(\left(\left(\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_1\right)\sin\left(\theta_2\right)\sin\left(\theta_4\right)\right)\cos\left(\theta_5\right)+\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)\cos\left(\theta_2\right)+\cos\left(\theta_1\right)\cos\left(\theta_3\right)\right)\mathrm{s}\right.$$
$$\left.\left(\theta_6\right)+\left(\left(\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\sin\left(\theta_4\right)-\sin\left(\theta_1\right)\sin\left(\theta_2\right)\cos\left(\theta_4\right)\right)\cos\left(\theta_6\right)\right.$$
$$-\left(\left(\sin\left(\theta_2\right)\cos\left(\theta_3\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\cos\left(\theta_5\right)-\sin\left(\theta_2\right)\sin\left(\theta_3\right)\sin\left(\theta_5\right)\right)\sin\left(\theta_6\right)+\left(\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_3\right)+\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right.$$
$$\left.\left(\theta_6\right)\right.$$

Fig : Column 5 of the Jacobian Matrix

The Jacobian Matrix

$$0$$

$$0$$

$$0$$

$$-\left(\left(\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_1\right)\right)\cos\left(\theta_5\right)+\left(-\sin\left(\theta_1\right)\cos\left(\theta_3\right)-\sin\left(\theta_3\right)\cos\left(\theta_1\right)\cos\left(\theta_2\right)\right)\right.$$
$$\left.\left(\theta_6\right)+\left(\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)+\cos\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)\right)\sin\left(\theta_4\right)-\sin\left(\theta_2\right)\cos\left(\theta_1\right)\cos\left(\theta_4\right)\right)\cos\left(\theta_6\right)\right.$$
$$-\left(\left(\left(\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\cos\left(\theta_4\right)+\sin\left(\theta_1\right)\sin\left(\theta_2\right)\sin\left(\theta_4\right)\right)\cos\left(\theta_5\right)+\left(-\sin\left(\theta_1\right)\sin\left(\theta_3\right)\cos\left(\theta_2\right)+\cos\left(\theta_1\right)\cos\left(\theta_3\right)\right)\mathrm{si}\right.$$
$$\left.\left(\theta_6\right)+\left(\left(\sin\left(\theta_1\right)\cos\left(\theta_2\right)\cos\left(\theta_3\right)+\sin\left(\theta_3\right)\cos\left(\theta_1\right)\right)\sin\left(\theta_4\right)-\sin\left(\theta_1\right)\sin\left(\theta_2\right)\cos\left(\theta_4\right)\right)\cos\left(\theta_6\right)\right.$$
$$-\left(\left(\sin\left(\theta_2\right)\cos\left(\theta_3\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\cos\left(\theta_5\right)-\sin\left(\theta_2\right)\sin\left(\theta_3\right)\sin\left(\theta_5\right)\right)\sin\left(\theta_6\right)+\left(\sin\left(\theta_2\right)\sin\left(\theta_4\right)\cos\left(\theta_3\right)+\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right)$$
$$\left(\theta_6\right)$$

Fig : Column 6 of the Jacobian Matrix

Inverse Kinematics Validation:

Now we can compute the Jacobian matrix using the homogeneous transformation matrix we obtained from forward kinematics. For our use, we are assuming that joint 1 is fixed where the joint connects the robot base and the manipulator base. Now, the jacobian matrix is divided into two, the upper half consists of all the linear velocity values whereas the lower half consists of the angular velocity vector.

For validation purposes , we used one robot pose for where link 3 and link 5 are upright or $\theta_3 = 90^\circ \; and \; \theta_5 = 90^\circ$. Using this we now tally the inverse kinematics and verify using the trajectory we get out of the robot. The velocity kinematics of the equation is assumed to be in the form of a curve of radius 0.1m , and hence the trajectory of the expected action from the base position to the above mentioned position should be curved.

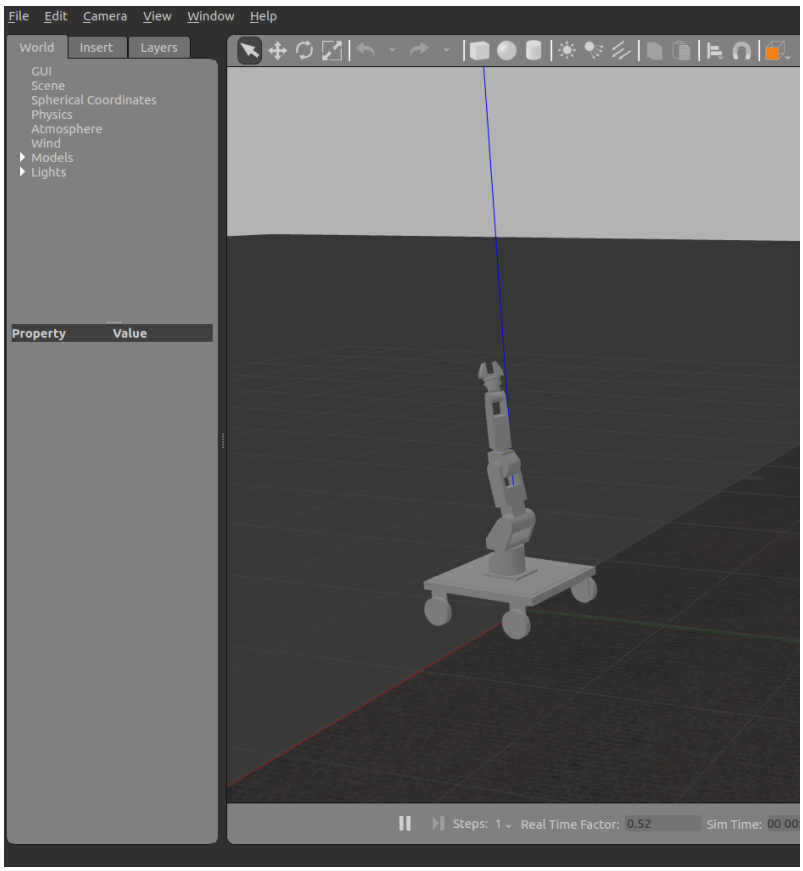


Fig : Described pose achieved in Gazebo

To calculate the inverse kinematics, we need the change in theta in a constant iteration . The maximum angle the joint can cover is 180 degree and keeping that in mind the velocity equation is defined for 5 seconds as $r \cdot \frac{\pi}{5}$, where r is the radii of the curve.

$$\dot{q} = X * J^{-1}$$

In our validation method, we will be verifying if the robot is moving to our desired pose when the constant change in $\Delta\theta$ is fed to the manipulator when it's at home . We tracked the trajectory and identified that the robot trajectory was similar to the curve and the end effector position values were similar as expected.

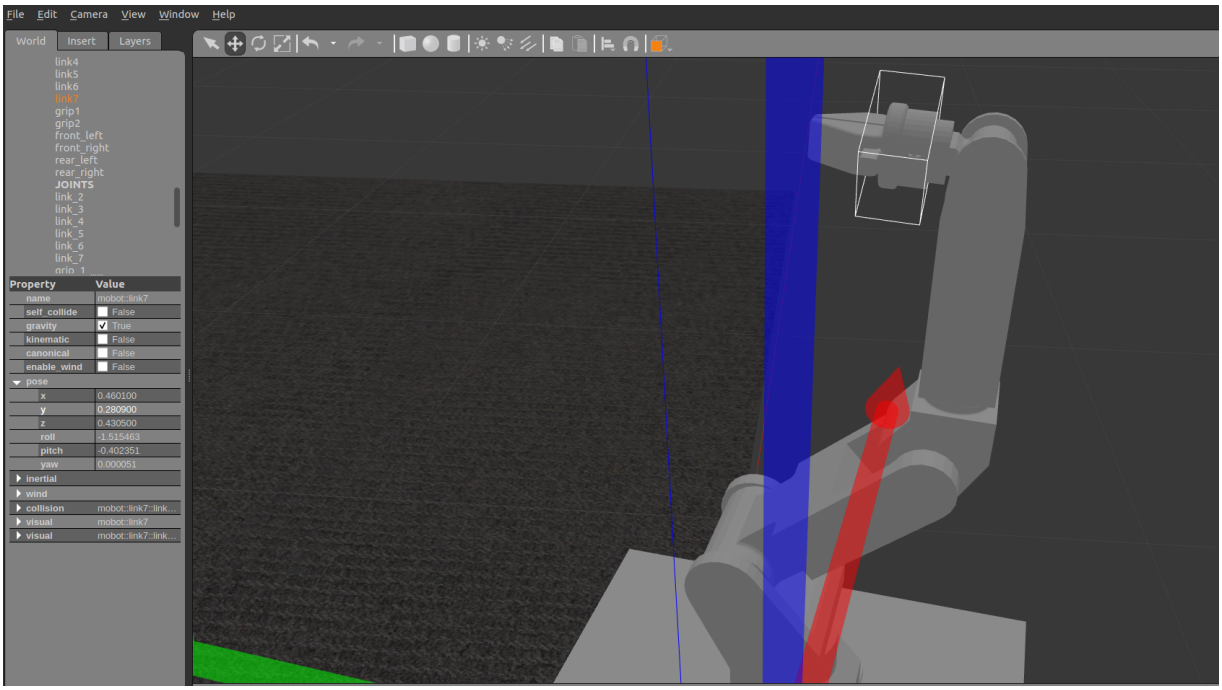


Fig : The coordinates of end effector after the pose completion

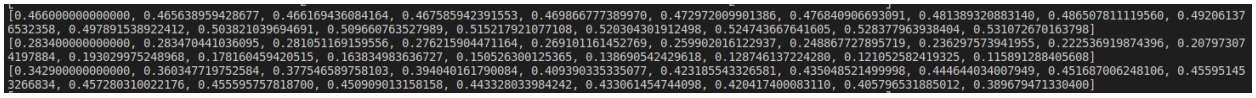


Fig: List of end effector values at each step presented

We can see that the end effector values in gazebo (0.481,0.280,0.4360) matches with the 9th value in the list. Below given the trajectory followed by the robot during the motion, when given a semi circular trajectory.

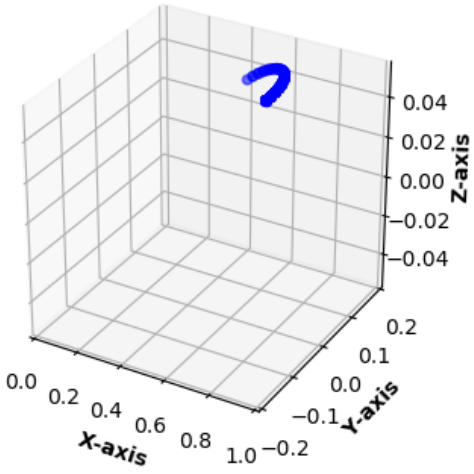


Fig: The followed trajectory of End Effector

We verified the inverse kinematics using RoboAnalyzer software to verify the correctness of the values we got. The IK values we got were like the values we got from our python code. Manually, Inverse kinematics can be calculated as the product of Jacobian Inverse and the set of known robot positions.

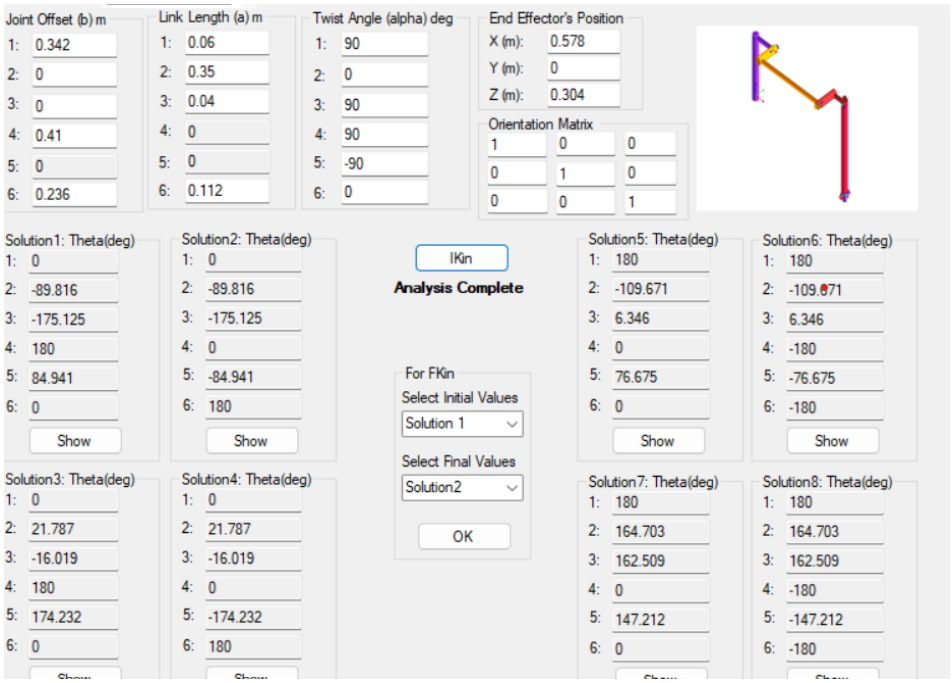


Fig: Verification of the Inverse Kinematics using ReDySim

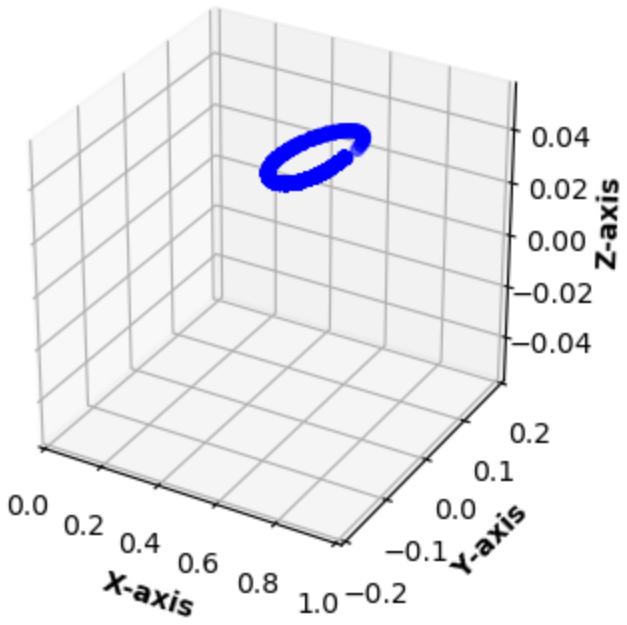


Fig: Trajectory of the robot when the trajectory equation is circle

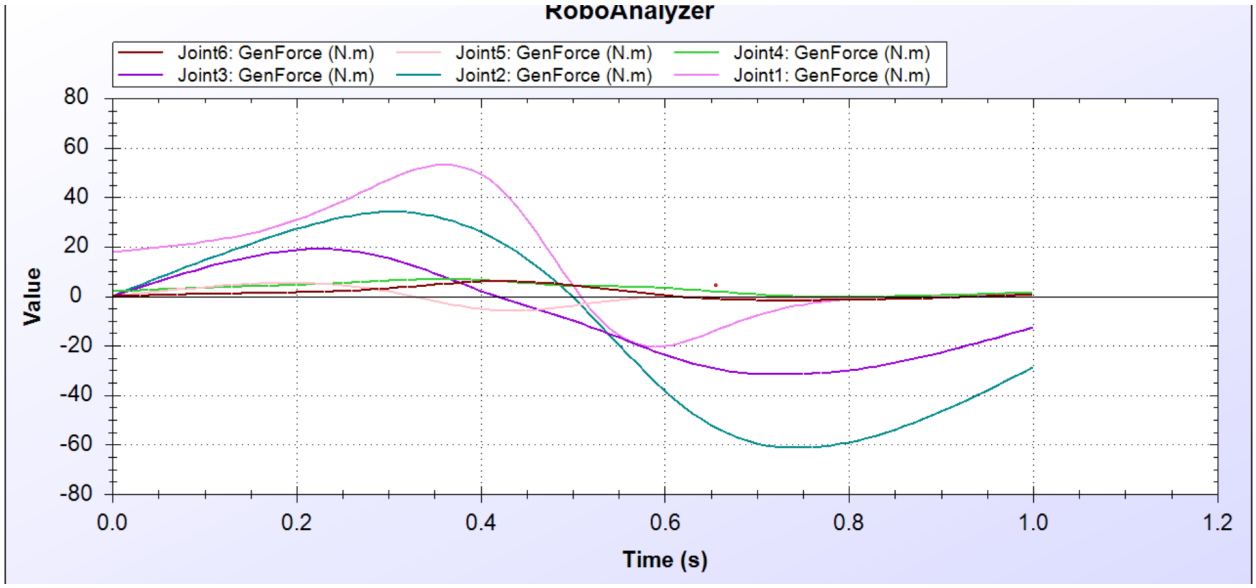


Fig : Force on the end effector after Inverse Kinematics Calculation

Workspace Study:

The workspace of the system is spherical in shape with an approximate diameter of 1.5 meters which is proved by the below images. The below images were produced by fully extending the robot arm in 1 direction in the xy plane and then rotating the first joint for full 360 degrees and then bending the elbow joints (total 3 elbow joints) to produce a spherical workspace. The workspace is similar to the image shown below:

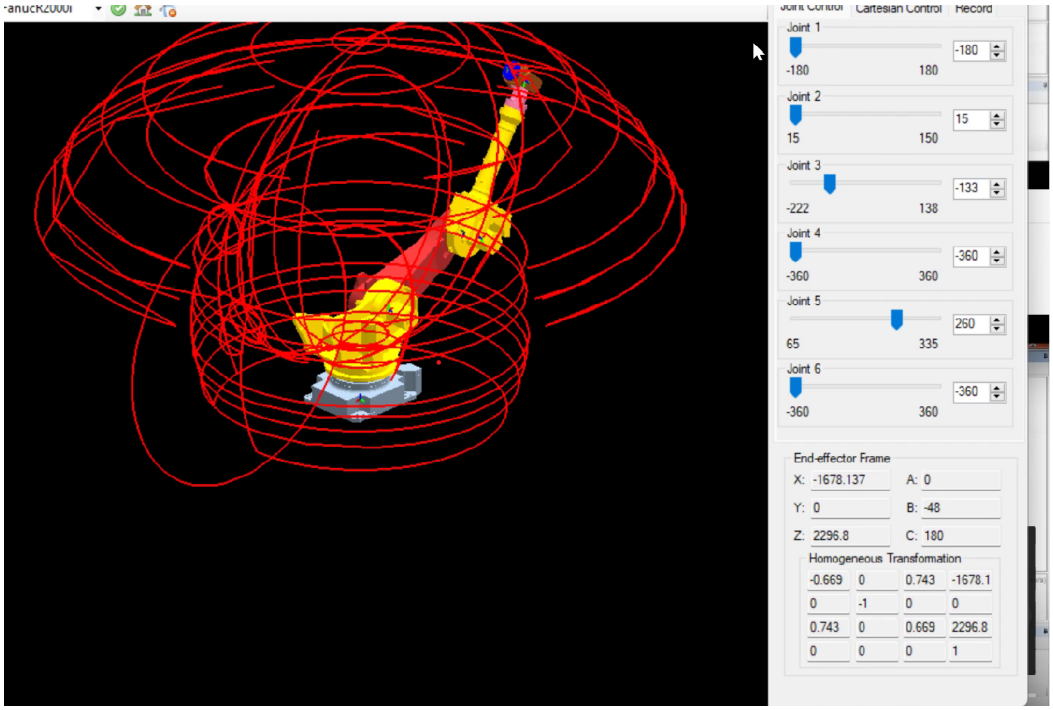


Fig: Workspace Generation

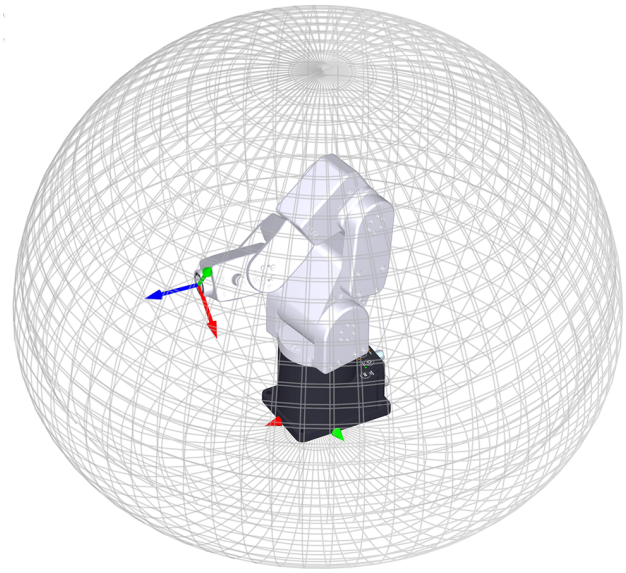


Fig: Original Robot Workspace

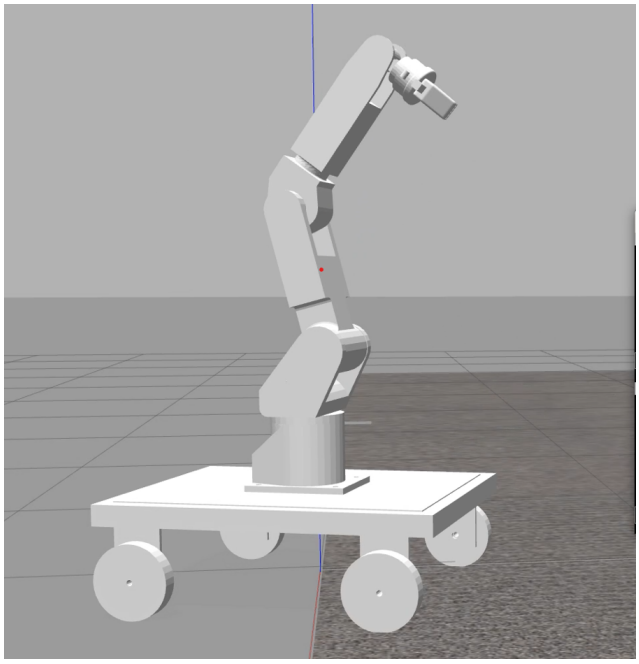


Fig: Robot deployed in environment

Assumptions :

The robot has 6 DOF and for the entirety of the robot system simulation, we assume the following holds true:

- All the links and joints have good structural integrity.
- The friction and other external vibrations are not considered.
- The objects that are picked and placed will have well defined edges and shapes.
- The path of the robot arm is defined but may not be the ideal path.
- The actuators are assumed to have higher torqued for maneuverability
- Thermal effects are ignored
- Robot material isn't fixed and is subjected to changes

Control Method:

As for the method of control we have used a simple PID controller as it satisfies the scope of the project. Other accurate or adaptive controllers could be used in the future depending upon the implementation and the precision required. The PID used were tuned based on the responses that we could see from the joint movements for the different PID values and the generic oscillations of joints. The PID values were in fact manually tuned and the changes to the controllers were shown through the rqt_gui tool offered in the ROS package.

As we are using manipulators, we used Effort Controllers. Initially we tried implementing Velocity Controllers, we found that using the Velocity controller was throwing multiple errors. The error seemed to be because we had used continuous arguments for each of the joints. And then after that we tried to use revolute joints with essentially a full 360 degree of freedom that was said to rectify the error but to no avail. We then proceeded to finally use Effort controllers and even though they seemed to work we could not publish joint positions as inputs to them.

We had learnt to work around that by giving the joints limits and then using the controller input to make them reach those positions. This allowed us to orient the joints in the exact position instead of trying to manipulate the joints manually. This method of manual movement of joints might help us in reaching a certain point but we had to tune the limits in the joint to make sure that we could move them to the exact pose. This method of orienting the joints to the pose lets us easily validate the forward kinematics.

Gazebo & RViz Visualization:

The simulation has been done on gazebo by grasping objects using tele-op to control the end effector to pick up various objects. The usual 2 finger end effector can pick up various objects. The end effector in our simulation video is shown to fully grab an object and then proceed to lift it up and move it and place it in a different location. We observed that it is handy to use a 2-finger gripper in this pick and place situation.

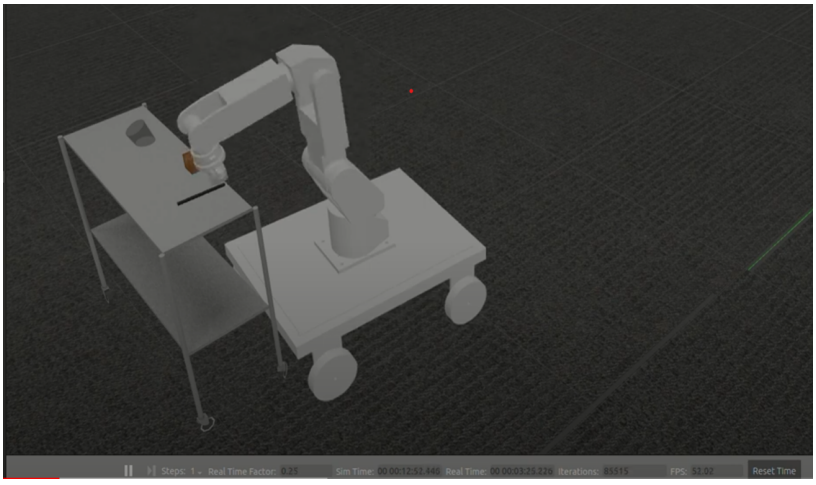


Fig : Robot performing Object Picking

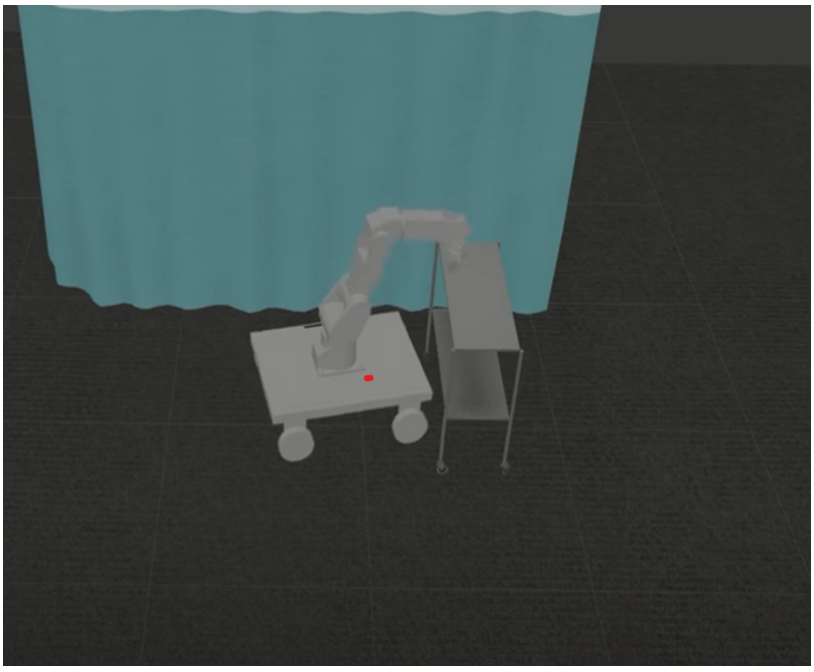


Fig : Robot attempting to place the object

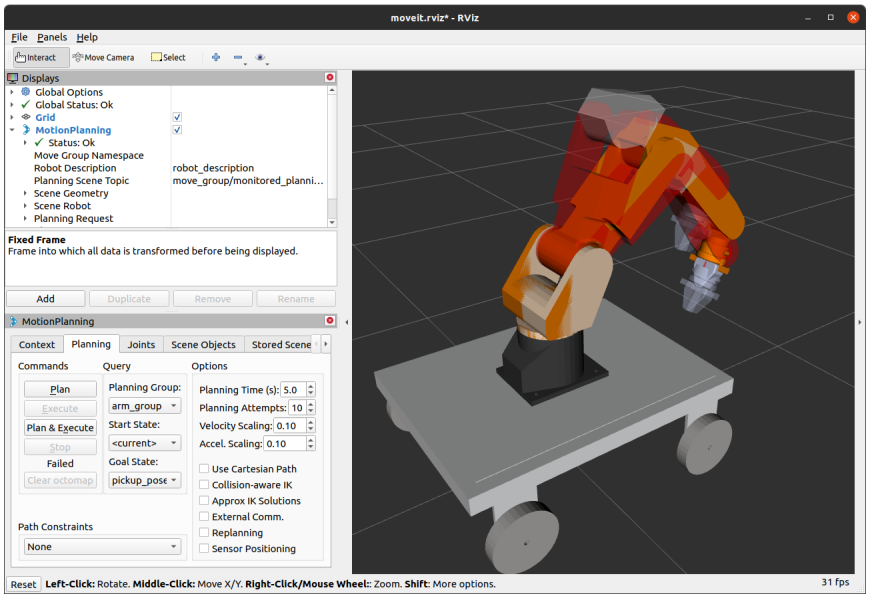


Fig : Rviz visualization of the robot picking up the object using a motion planner

https://drive.google.com/drive/folders/1FWGGrsOYWW4DuOndEivPhxbhFenLA6eq?usp=share_link

RQT_GRAPH

A GUI plugin called rqt graph makes it possible to see the ROS computation graph.Its components are designed to be generic so that this package offers all the mapping of nodes and topics when its publishes and also gives the details of who has subscribed the topics and the nodes. It also gives information about all the active topics and controllers.

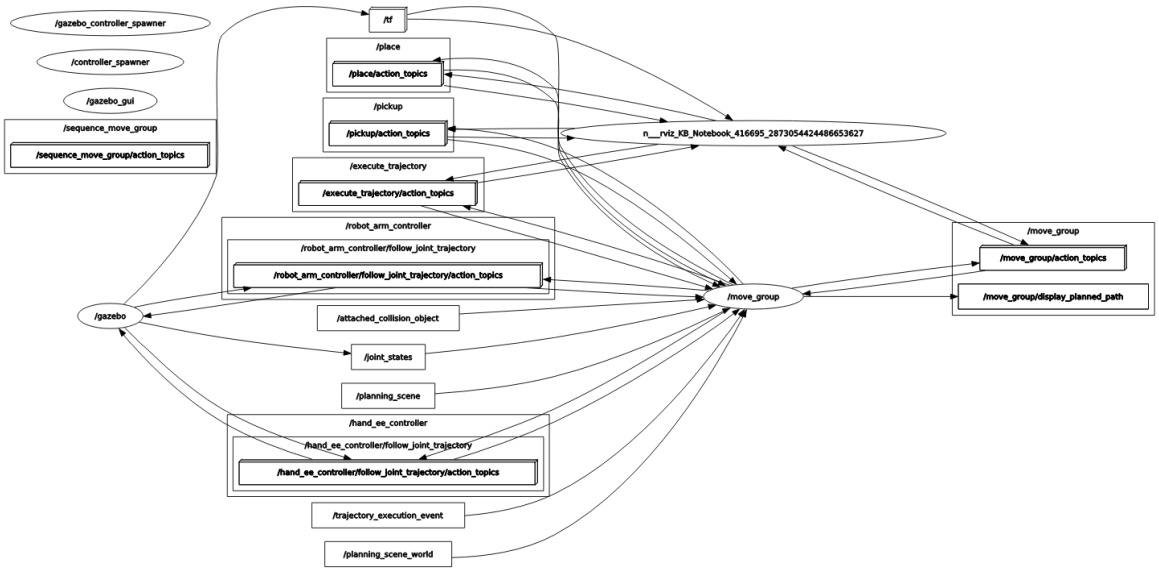


Fig 1(a): Representation of RQT graph when Moveit is deployed

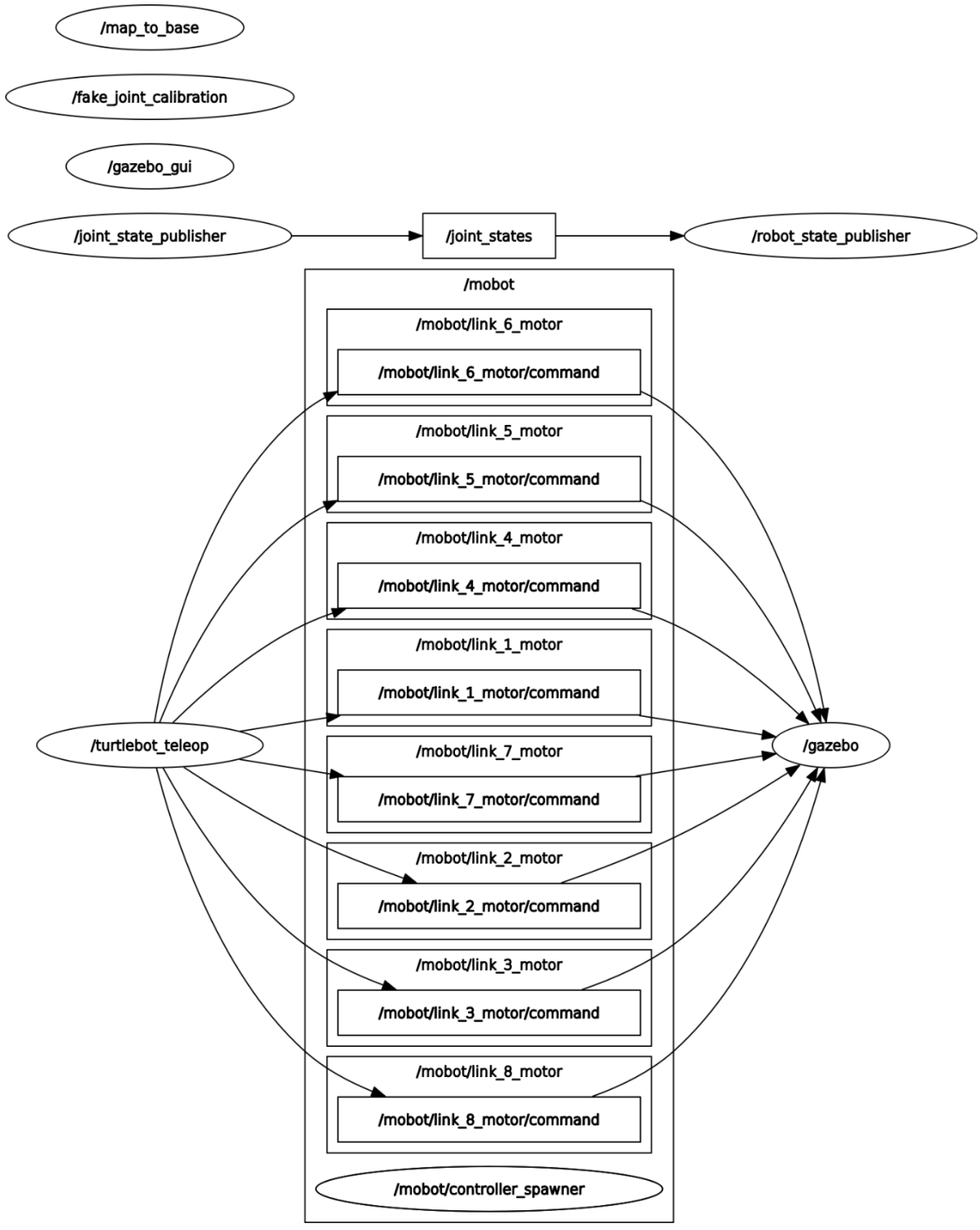


Fig: RQT Graph of the teleoperation of the robot in an environment

Problems Faced

● Our initial plan was to use a CAD model directly imported from the FANUC manipulator website; however, the model had many issues while converting to URDF. We later used the same parts with small tweaks. This was then re-oriented and assemblies and sub-assemblies were done.

● We still faced issues while initially spawning the robot with masses of links and their moments which made the robot fall to the ground. We manually tuned in values using the hit and trial method. Then we had to change the coordinate system with the CAD model compatible to Gazebo coordinate system.

● PID tuning took a while as we manually tuned in the values. We used a graphical tool later for completing this step. The tool used was rqt_gui which allows us to send a function as an input to any of the controllers and then visualize the output in comparison to the desired behavior.

● Our initial plan was to use Velocity controllers for the manipulator, however, there was an issue while loading those controllers on our systems and eventually we had to switch to Effort controllers. However, we have used Velocity controllers for our robot base.

● Finally, the simulation took very long because orienting the end effector to pick the object up from the table was difficult because of the problem we faced to tune PID values. The grasping function took very long, because the object would slip off the end effector.

Lessons Learned

● We improved our skills in CAD modeling and re orienting the parts and assembly according to the coordinate system in Gazebo.

● We got to learn about some advanced mates and used reference shapes to mate parts.

● PID tuning can be done conveniently with the use of the ROS graphical tool(rqt_gui).

● The base of the robot needs to be strong enough to support the links above it, therefore, URDF should be checked properly for all the links.

● The URDF file can be manually tweaked to resolve minor spawning errors.

● Choice of controllers is based on the type of joints, like for a continuous joint, velocity controller would be correct and for a revolute joint, Effort controller is preferred.

● The tele-op file must be a well thought implementation especially when the number of joints to be controlled are very high. Similarly, the key bindings need to be intuitive so that it is easy for the human to maneuver the bot.

● We also learned how to create two separate packages for the manipulator arm and end effector and then integrate both in the same world using Xacro. However, we did not use this method as the process became very complex and debugging the errors became a hassle

Future Works:

One of the major works that can be implemented on the current implementation is to reduce the mechanical instability and have a compact design to serve in extreme environments. This model can be used as an add on for the existing hospital nursing bots as the use of manipulators extends the current usage of these bots. The usage of a manipulator can help the robot to be deployed in extreme conditions as in surgical procedures and also in hospital warehouses.

Along with design changes, the reduction of wheels from four to two and decreasing body size is also one of the works that has been done by the Toyota institute. The usage of manipulator in these systems has been a question mark, but the research team is exploring avenues to implement a model with high reachability and having greater workspace.

Conclusion:

We were able to assemble the robot from the parts that we extracted from the official website for the WAM developers and despite the differences in the origin of different parts with respect to their joint positions, we shifted them to the required axes for the final assembly.

The URDF was particularly tricky as we had to make changes to the URDF constantly to make sure the robot was able to properly spawn, and the weights and the joint types were tuned according to the spawn stability. The joint types were changed from initial continuous to revolute and consequently were given joint and velocity limits based on the kinematic configurations.

The forward kinematics of the robot was rather easy as we already had the exact dimensions and followed the conventions to get the dh table for the calculation. Then the validation for which was done through geometric observation and the tuning of joint limits as mentioned above.

The inverse kinematics of the robot had more than the number of columns needed for a perfect square matrix and its inverse was only calculated using the pseudo inverse method. As for the validation we fixed 2 redundant joints and were able to make the Jacobian a perfect square and then proceed to make the robot trace a straight line which was plotted by the values provided.

In the simulation we showed that our robot was able to pick up one of the objects by individually controlling the end effector joints forming them into an appropriate shape for which the key bindings had 20 keys. The end effector can form complex shapes by manipulating the individual joints which it can use to pick up differently shaped objects.

Although we were able to pick up objects we wanted, we were not able to model the environment in which we could simulate scenarios. We will further work on making the robot follow trajectories and pick up highly complicated objects. We want to implement a controller that lets us have a better movement of the robot and hence allows for high precision applications such as wound dressing, etc.

References

1. Implementation of Automated Guided Vehicle System in Healthcare Facility; Marko,Pedana; Milan,Gregora ;Dariusz,Plintab https://www.sciencedirect.com/science/article/pii/S1877705817326619

2. Guidance needed during the project to set up an environment in Gazebo and to learn about other new topics: https://www.youtube.com/ & https://docs.ros.org

3. Visual servo control of cable-driven soft robotic manipulator;Hesheng Wang; Weidong Chen; Xiaojin Yu; Tao Deng; Xiaozhou Wang; Rolf Pfeifer: https://ieeexplore.ieee.org/document/6696332/authors#authors

4. https://grabcad.com/library/simulation-of-6-dof-fanuc-manipulator-1

5. https://www.theconstructsim.com/

6. https://mag.toyota.co.uk/toyota-human-support-robot/

7. https://answers.ros.org/questions/

8. https://robotics.stackexchange.com/