ENPM662 - Introduction to Robot Modeling

Project 2 Proposal
Fruit Harvesting Robot - GoodPick

Hritvik Choudhari – UID: 119208793

Abhimanyu Saxena - UID: 119342763

Date- 12/09/2022

Contents

1. Introduction and Organization:

GoodPick is an UGV designed to carry out wide variety of outdoor operations. It's high payload capacity and large lug-tread tires allows it to tackle challenging real-world terrain. Although GoodPick robot is mainly used for long distance tele-operation and navigation and path planning, we intend to mount a 3 DOF manipulator over it to extend its application in the agricultural domain especially for fruit and berry picking task with the help of high-resolution RGB camera. Appropriate controllers are added to make the manipulator attached to follow a trajectory based on the location of the fruit it intends to pick. This project models the functionality of the robot.

2. Motivation:

Agriculture has been humankind's oldest yet most important economic activity. According to World Population Prospects 2022 by the UN, the global population is estimated to reach 9 billion by the year 2050 and the agricultural production must twofold to meet the demands of food. Given the limited resources of land, water and labor, the involvement of robots in agriculture is helping achieve this goal. Today vision and GPS based robots find their place in almost all farming activities like plowing the soil and sowing seeds to stacking the shelves in a food market.

The purpose of this project is to address the issues faced in the harvesting season due to the long going labor shortage. The proposed design improves mobility, and the mounted manipulator can control the position of grippers that harvest fruits and vegetables with enhanced speed and accuracy. This reduces wastage and thereby increases yield. These robots can be deployed in numbers to tackle labor shortage and thereby increase profits in the harvest season.

3. Robot Description:

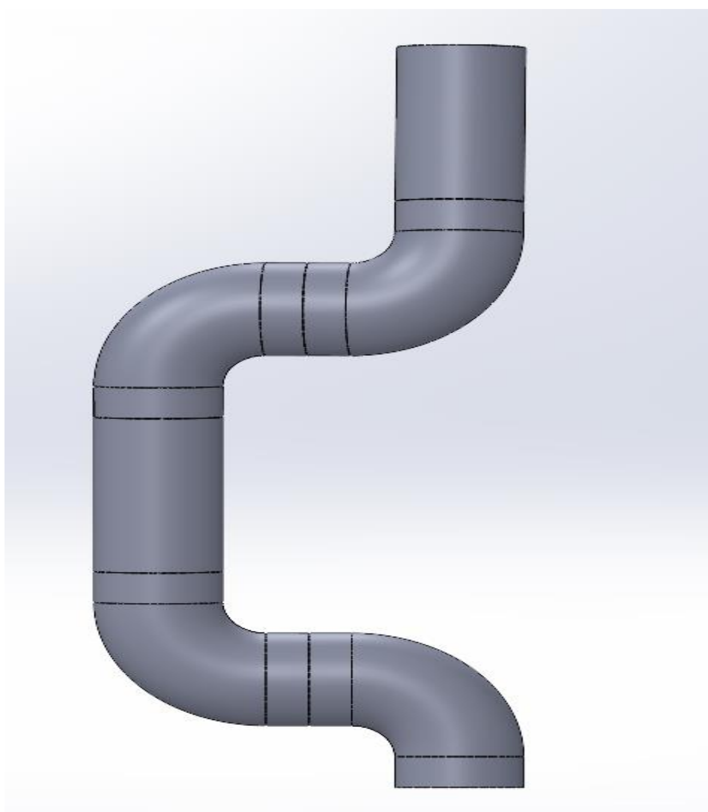
Our robot draws its design inspiration from the Husky unmanned bot from Clearpath Robotics. We propose a 4 wheel, all terrain robot with 2 DOF. One each in X and Y axis and Rotation about Z axis. This minimal design allows stable motion while performing the tasks in the agricultural field. The major component of our robot is a manipulator mounted on the top plate. It is a spherical 3 DOF manipulator which identifies and picks ripe fruits. A camera is attached to the gripper which will identify the fruit and guide the arm to its position. The manipulator consists of 3 revolute joints which give it a standard 3 DOF motion. The robot will also have a set of sensors to guide it autonomously in the field. There is also a vision sensor attached on the top of our manipulator that will help the robot in identifying the fruit and plan its trajectory according to its position and pick it using the suction gripper as the manipulator's end-effector. Below are the dimensions of 2 major components (Mobile base and manipulator):

Reference image of Husky from Clearpath Robotics which we plan to modify, model and simulate according to the scope.

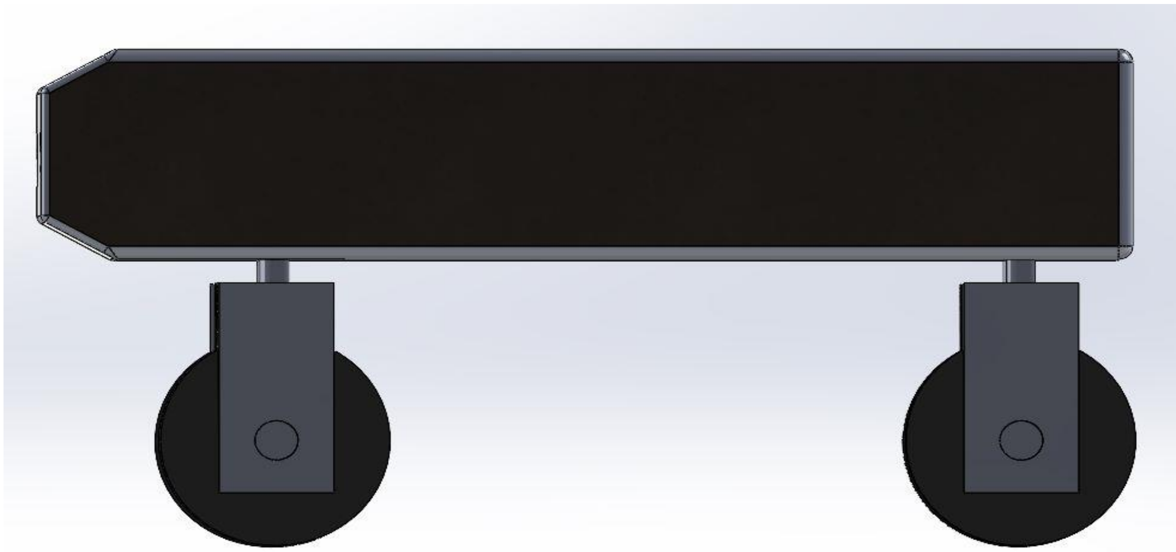
GoodPick (Fruit harvesting robot)		
Sr No	**Specification**	**Description**
1	Robot Dimensions	39 x 26.4 X 14.6 (in)
2	Wheel Dimensions	10 in diameter x 5 in wheelbase

5	Ground clearance	12 in
6	Attachments	Basket for fruit collection
7	DOF of robot base	2
8	DOF of manipulator (incl EE)	3
9	Gripper type	Suction
10	Applications	Fruit harvesting, weed removal etc
11	Vision sensors	RGB vision sensor

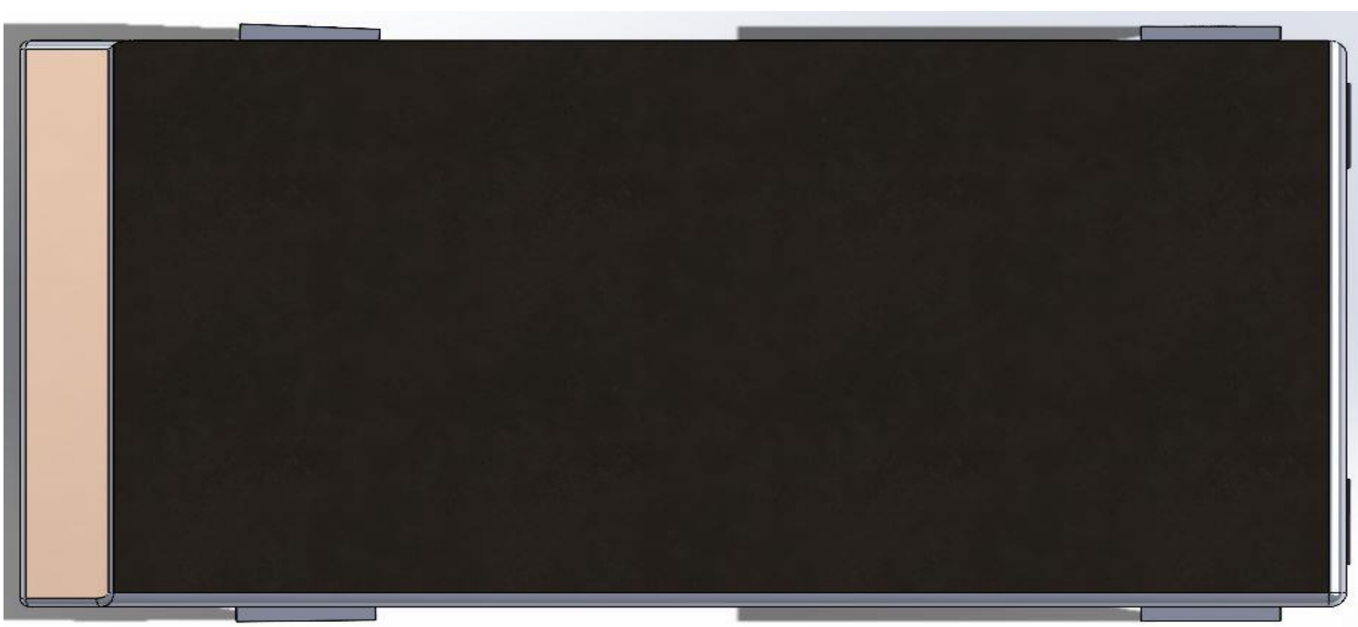
4. CAD Models:



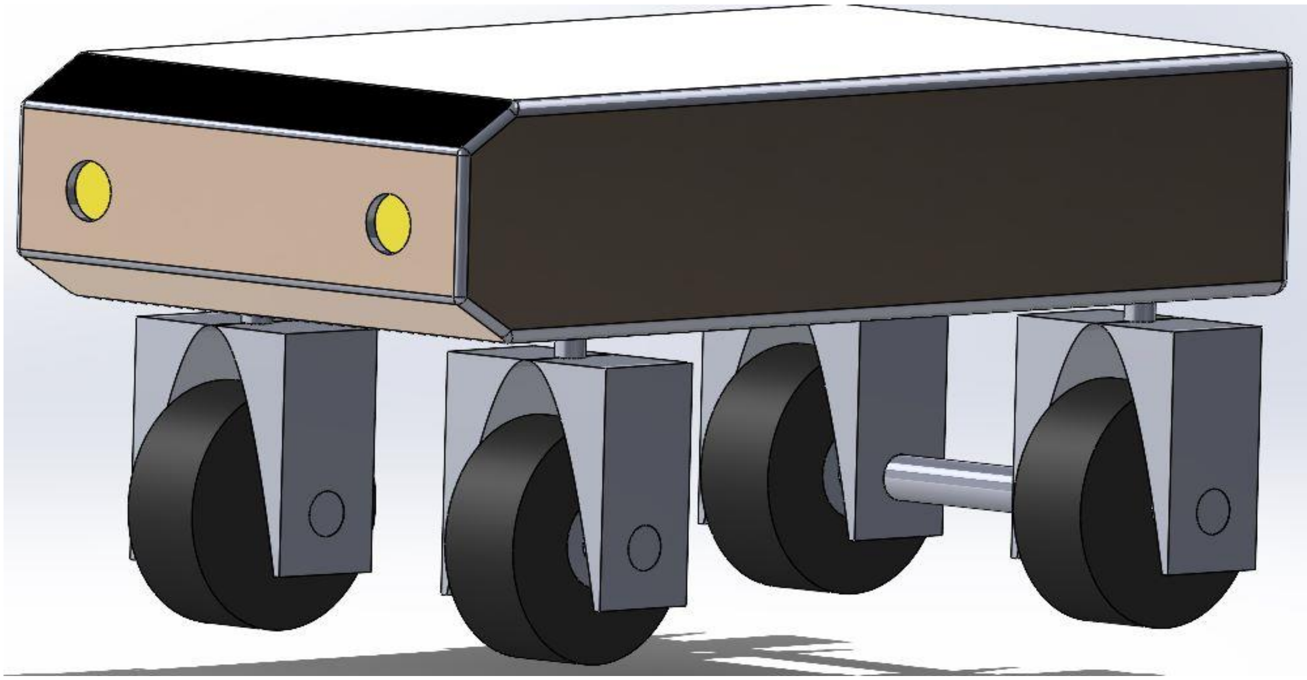
Manipulator Front view



Robot base side view



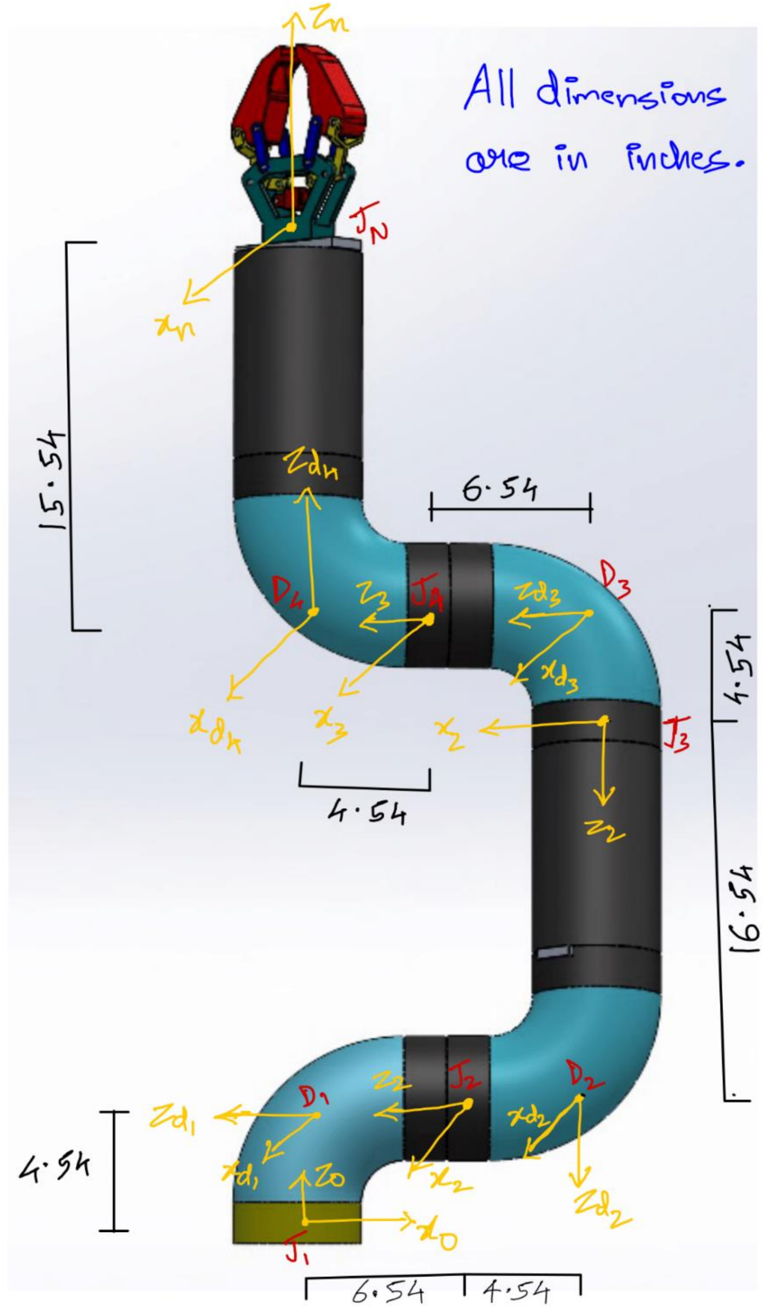
Robot base top view



Robot base

5. DH Parameters and Table:

The DH parameters were calculated using the spong's convention.



DH frames

Joints	α	d	a	Θ
J1	90	4.54	0	-90
D1	0	-6.54	0	0
J2	90	-4.54	0	Θ(2)
D2	0	-16.54	0	90
J3	-90	-4.54	0	Θ(3)-90
D3	0	6.54	0	0
J4	-90	4.54	0	Θ(4)
D4	0	15.54	0	0

DH parameter table.

6. Forward Kinematics and Validation:

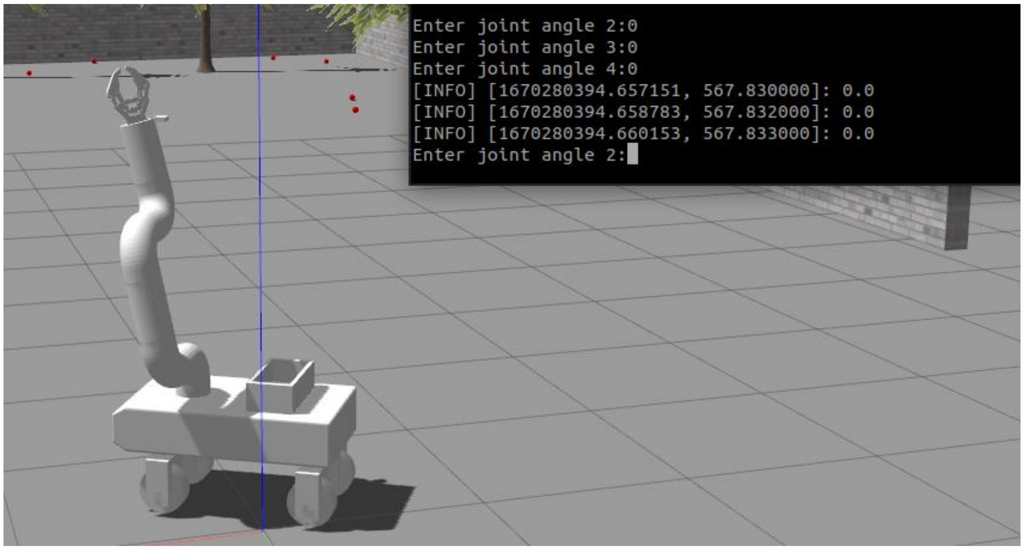
The forward kinematics was performed by plugging in the joint angles at a particular instance in the final transformation matrix to get the position coordinates of the end-effector. Below is the transformation matrix of end-effector with respect to its base T_N^0:

```
Matrix([[-sin(theta3)*cos(theta4), cos(theta3), sin(theta3)*sin(theta4), 15.54*si
n(theta3)*sin(theta4) - 11.08*cos(theta3) + 11.08], [sin(theta2)*sin(theta4) - co
s(theta2)*cos(theta3)*cos(theta4), -sin(theta3)*cos(theta2), sin(theta2)*cos(thet
a4) + sin(theta4)*cos(theta2)*cos(theta3), 15.54*sin(theta2)*cos(theta4) + 21.08*
sin(theta2) + 11.08*sin(theta3)*cos(theta2) + 15.54*sin(theta4)*cos(theta2)*cos(t
heta3)], [sin(theta2)*cos(theta3)*cos(theta4) + sin(theta4)*cos(theta2), sin(thet
a2)*sin(theta3), -sin(theta2)*sin(theta4)*cos(theta3) + cos(theta2)*cos(theta4),
-11.08*sin(theta2)*sin(theta3) - 15.54*sin(theta2)*sin(theta4)*cos(theta3) + 15.5
4*cos(theta2)*cos(theta4) + 21.08*cos(theta2) + 4.54], [0, 0, 0, 1]])
```

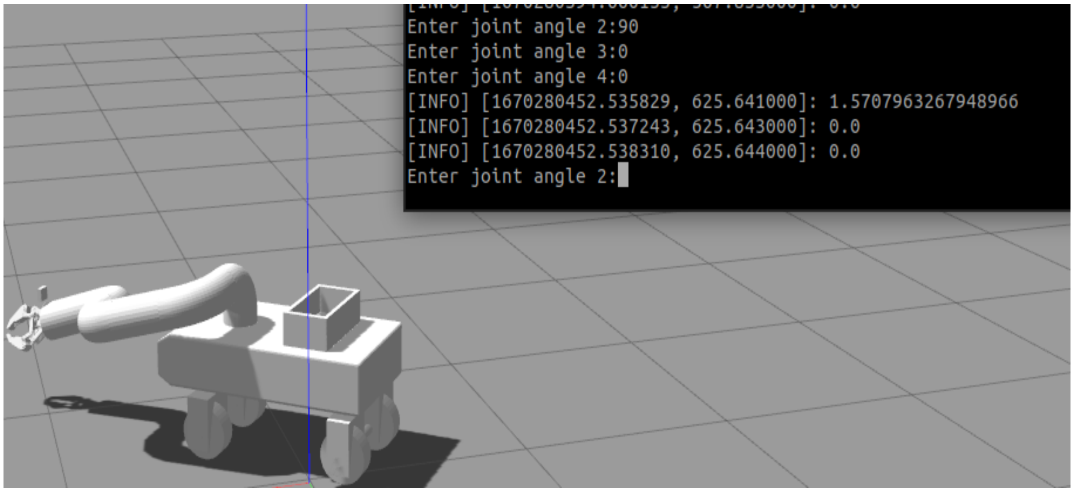
and below is transformation at robot's home position (all joint angles at 0):

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 41 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

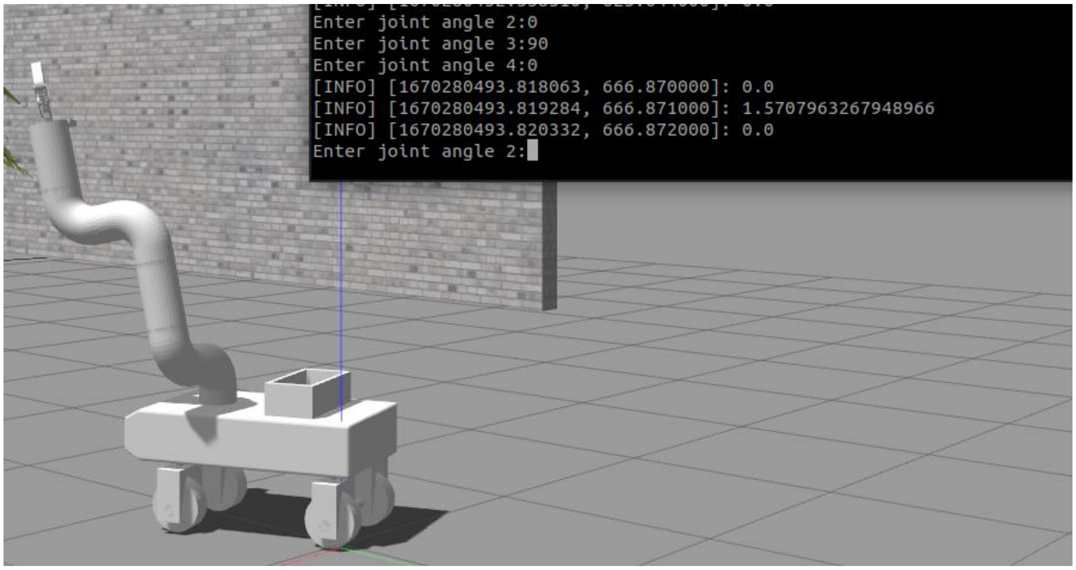
- J1 = 0° | J2 =0° | J3 = 0°



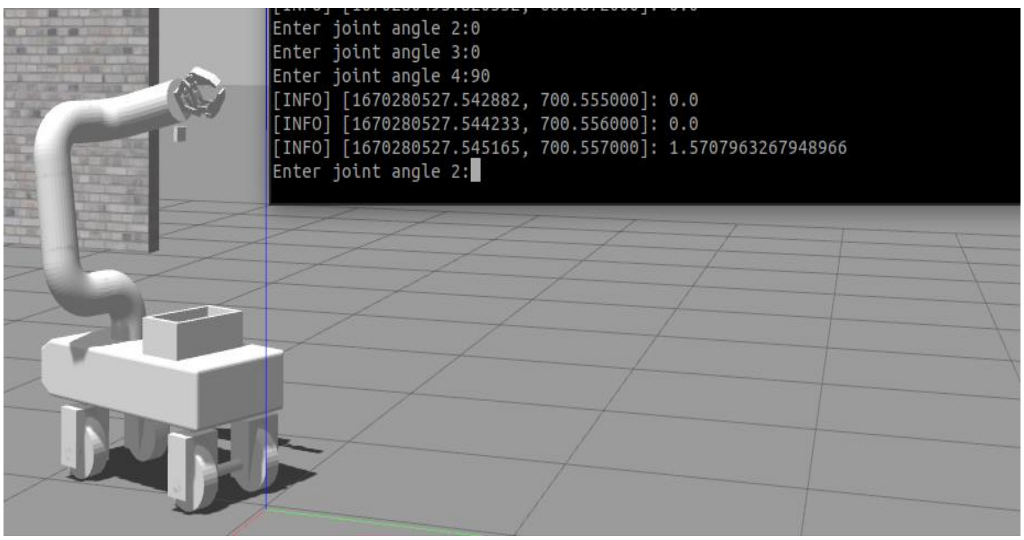
- J1 = 90° | J2 =0° | J3 = 0°



- J1 = 0° | J2 =90° | J3 = 0°



- J1 = 0° | J2 =0° | J3 = 90°



7. Inverse Kinematics and Validation:

The inverse kinematics calculation was performed using the concept of inverse Jacobian matrix which when multiplied by the position matrix gives us the respective joint angles of the manipulator for that position.

$$\mathbf{J}^{-1}\Delta\mathbf{r} = \Delta\boldsymbol{\theta}$$

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

Jacobian matrix calculation using partial derivative method

We have plotted the trajectory of the end-effector from home pose to the pick pose using the concept of both forward and inverse kinematics for validation purpose.

Code:

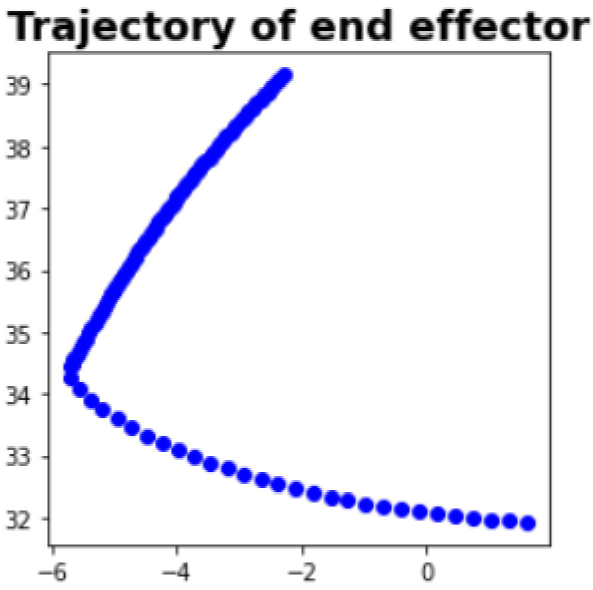
```
# importing all libraries

import numpy as np

from sympy import *

import matplotlib.pyplot as plt

# defining all variables

a1,a2,a3,a4= symbols('a1 a2 a3 a4')

q1,q2,q3,q4= symbols('q1 q2 q3 q4')

d1,d2,d3,d4= symbols('d1 d2 d3 d4')
```

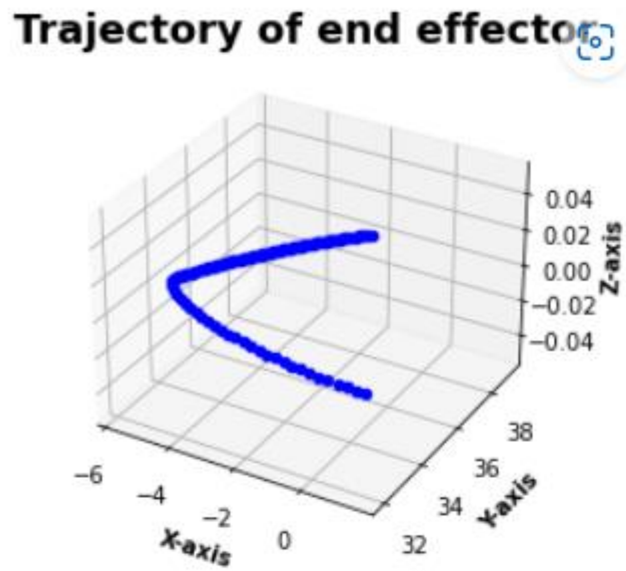
```
alpha1,alpha2,alpha3,alpha4= symbols('alpha1 alpha2 alpha3 alpha4')

theta1,theta2,theta3,theta4= symbols('theta1 theta2 theta3 theta4')

a,d,theta,alpha= symbols('a d theta alpha')

# creating the general D-H transformation matrix

R1= [1*cos(theta), -sin(theta)*cos(alpha), sin(theta)*sin(alpha), a*cos(theta)]

R2= [1*sin(theta), cos(theta)*cos(alpha), -cos(theta)*sin(alpha), a*sin(theta)]

R3= [0, 1*sin(alpha), 1*cos(alpha), 1*d]

R4= [0,0,0,1]

T= Matrix([R1,R2,R3,R4])

def Transformation_matrix(alpha, theta, di, ai):

    return Matrix([[cos(theta), -sin(theta)*cos(alpha), sin(theta)*sin(alpha), ai*cos(theta)],

    [sin(theta), cos(theta)*cos(alpha), -cos(theta)*sin(alpha), ai*sin(theta)],

    [0, sin(alpha), cos(alpha), di],

    [0, 0, 0, 1]])

T1 = Transformation_matrix(pi/2,-pi/2,4.54,0)

T2 = Transformation_matrix(0,0,-6.54,0)

T3 = Transformation_matrix(pi/2,theta2,-4.54,0)

T4 = Transformation_matrix(0,pi/2,-16.54,0)

T5 = Transformation_matrix(-pi/2,theta3-(pi/2),-4.54,0)

T6 = Transformation_matrix(0,0,6.54,0)

T7 = Transformation_matrix(-pi/2,theta4,4.54,0)

T8= Transformation_matrix(0,0,15.54,0)

# T_N= T1*T2*T3*T4*T5*T6*T7*T8

T10= T1

T20= T10*T2*T3

T30= T20*T4*T5

TN0= T30*T6*T7*T8

# Extracting all Zi and storing them in a list
```

```
T_list = [T10,T20,T30,TN0]
Z_list= []
for i in T_list:
    a= i.col(2)
    b= a.row_del(3)
    Z_list.append(a)
# Partially differentiating the Xp of T70 wrt thetas
thetas= [theta2,theta3,theta4]
pds= []
for t in thetas:
    pd= diff(TN0.col(3),t)
    d=  pd.row_del(3)
    pds.append(pd)
J1 = Matrix([pds[0], Z_list[0]])
J2 = Matrix([pds[1], Z_list[1]])
J3 = Matrix([pds[2], Z_list[2]])
J = Matrix([[J1, J2, J3]])          # Jacobian matrix
q= [0.5,-0.1,-1.5]
x = []
y = []
z = []
w=0.75
dt=1
fig = plt.figure(figsize=(4,4))
ax = fig.add_subplot(111, projection= '3d')
j4= q[2]
final= -0.5
#Plotting the trajectory points of the robot
```

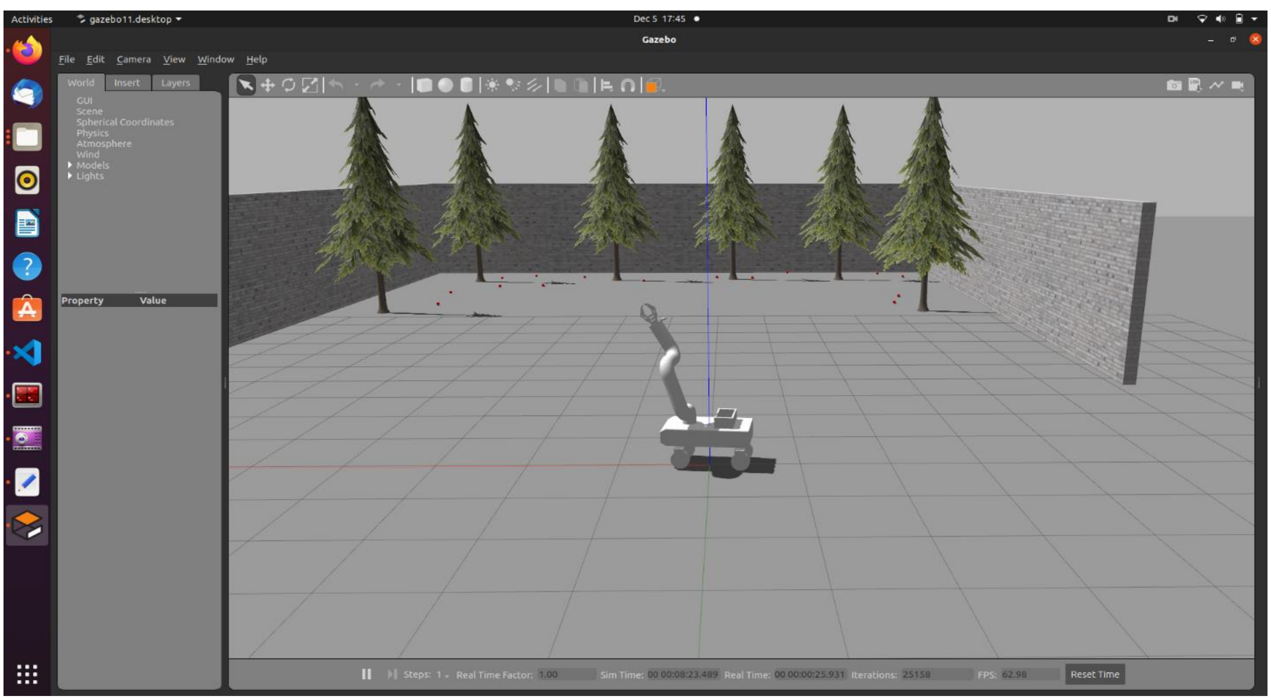
```
while j4!= final:

    Vx = -0.39*w*sin(-j4)

    Vy = 0

    Vz = 0.39*w*cos(-j4)

    X_dot = Matrix([[Vx], [Vy], [Vz],[0],[0],[0]])

    A = J.evalf(3,subs= {theta2:q[0], theta3:q[1], theta4:q[2]})

    X_new = TN0.subs({theta2:q[0], theta3:q[1], theta4:q[2]})

    x.append(X_new[3])

    y.append(X_new[7])

    z.append(X_new[11])

    ax.scatter(X_new[3],X_new[7],X_new[11], c='b')

    ax.set_xlabel('X-axis', fontweight ='bold')

    ax.set_ylabel('Y-axis', fontweight ='bold')

    ax.set_zlabel('Z-axis', fontweight ='bold')

    ax.set_title('Trajectory of end effector', fontsize = 18, fontweight ='bold')

    q_dot = A.pinv()*X_dot

    q_new = Matrix([q]).T + (q_dot*dt)

    q_new = q_new.evalf()

    q= q_new.T

     j4= q[2]
```

The below trajectory of end-effector is from home pose to pick pose.

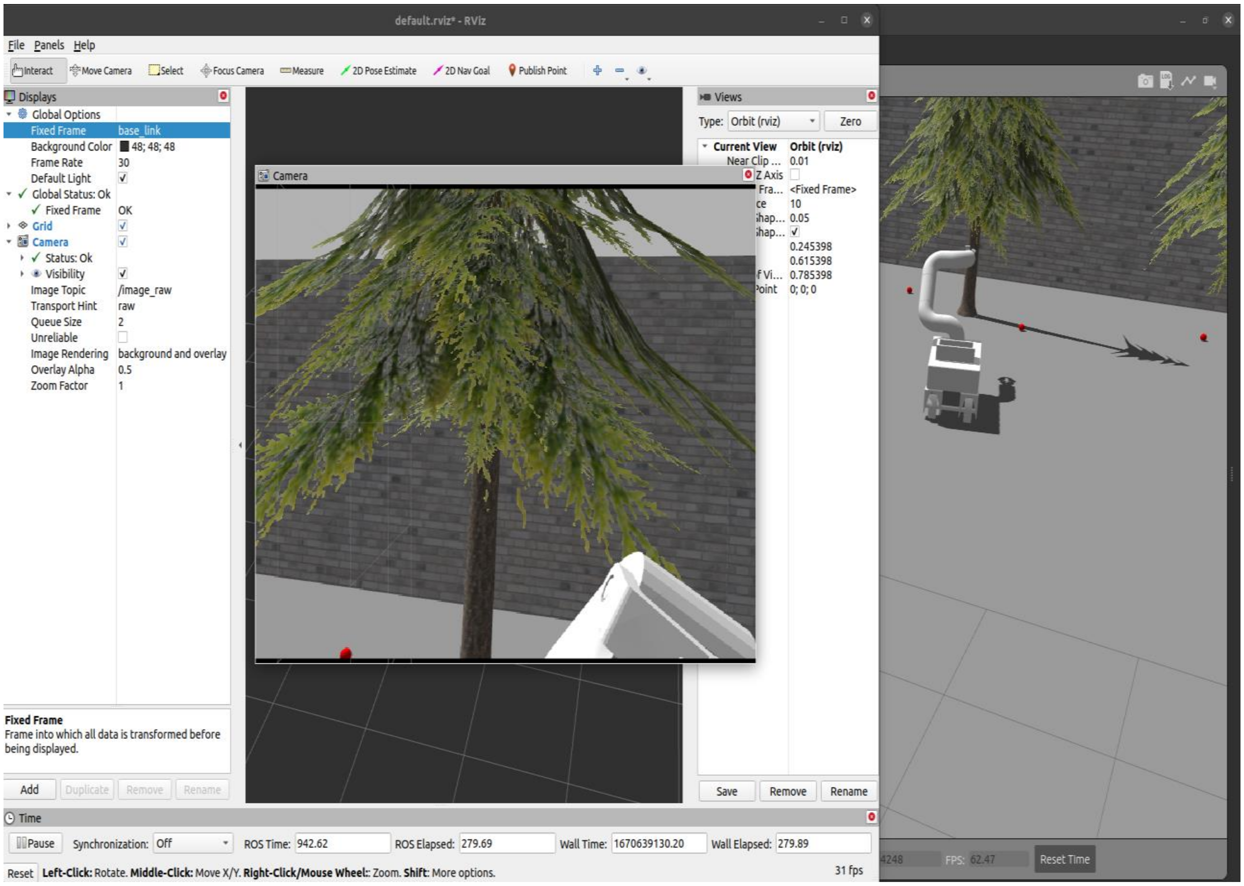




8. Gazebo and Rviz visualization: [link to video]

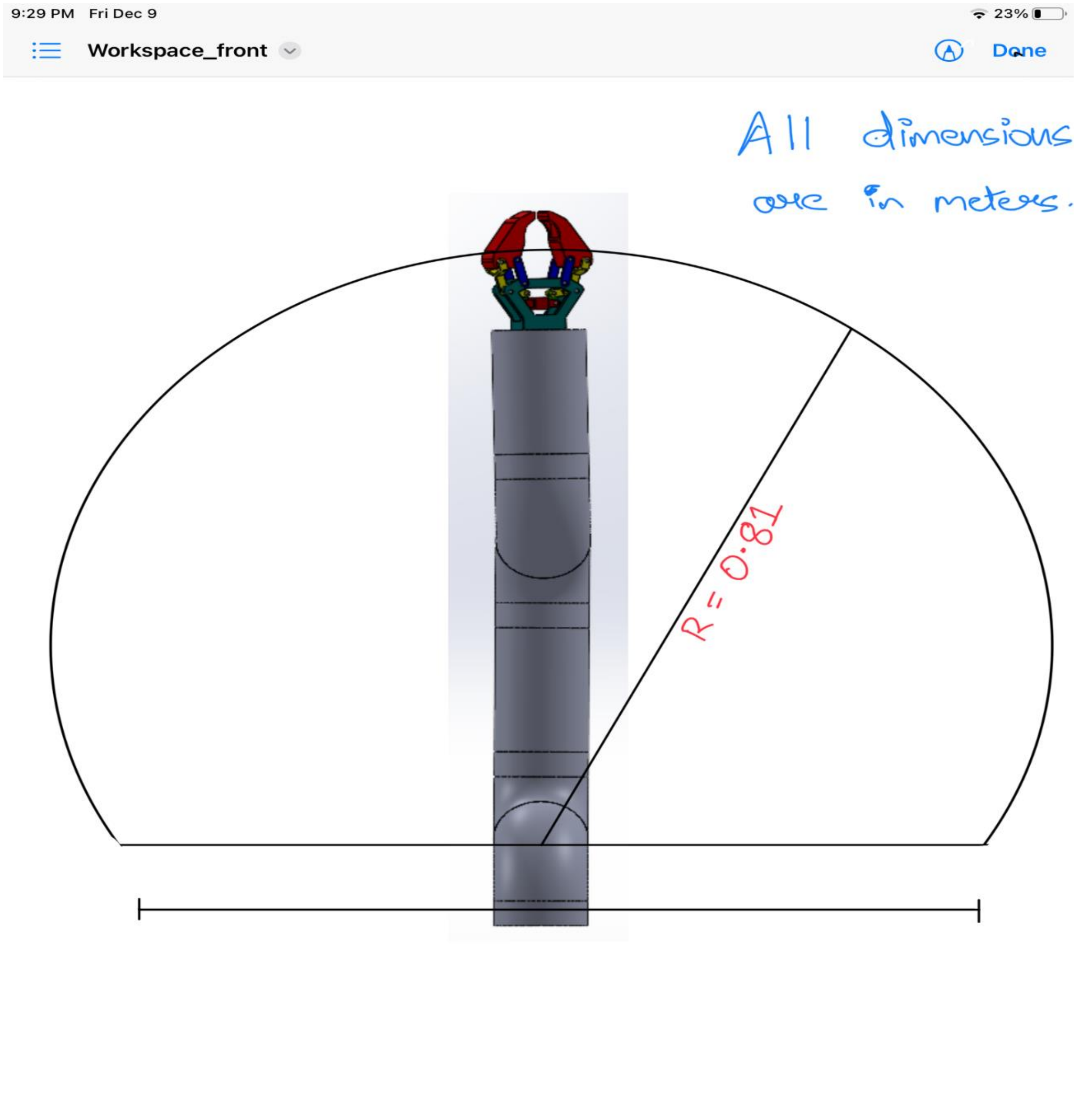


Robot spawned in Gazebo simulator

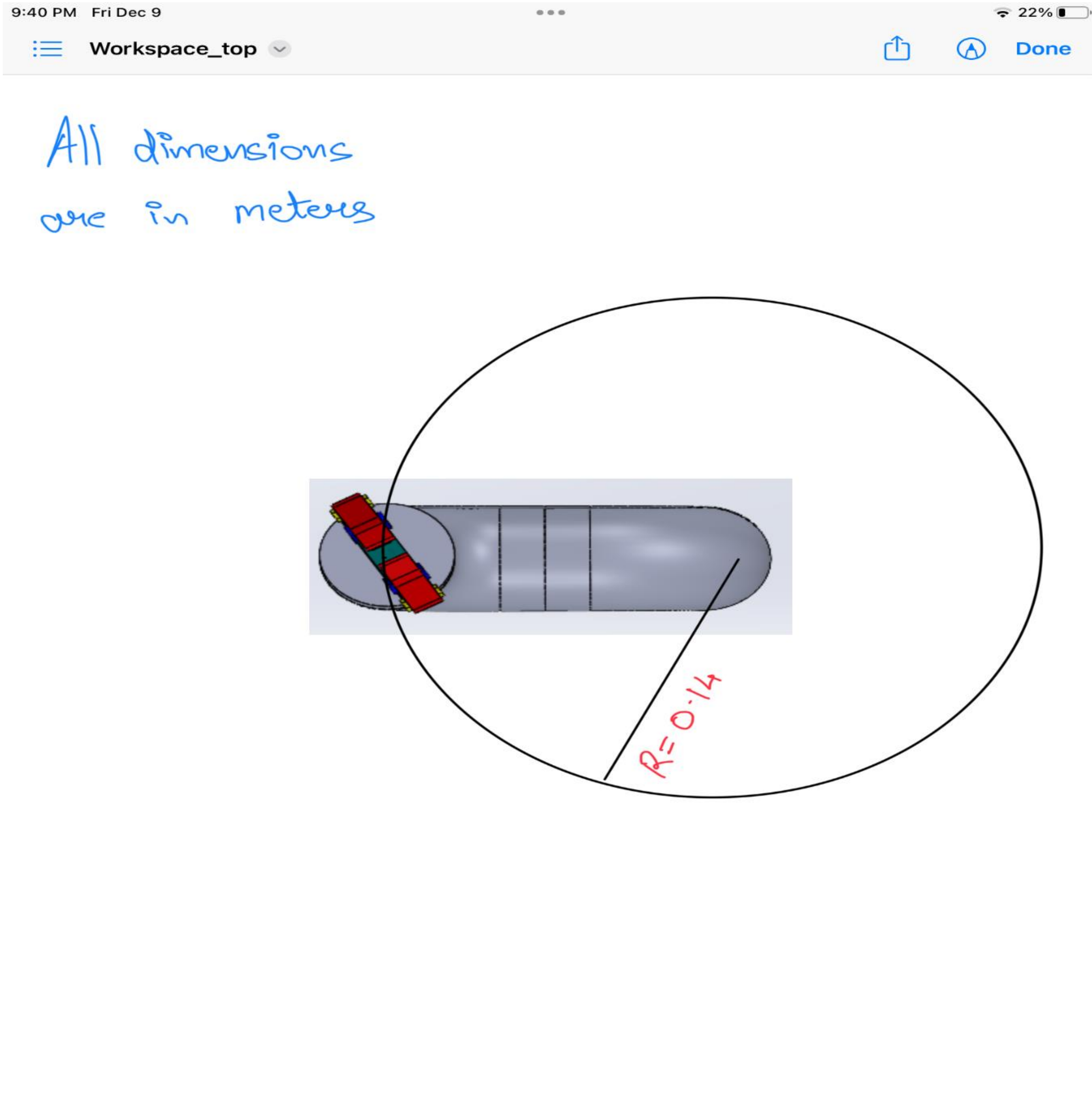


Robot visualization in Rviz

9. Work Space Analysis:



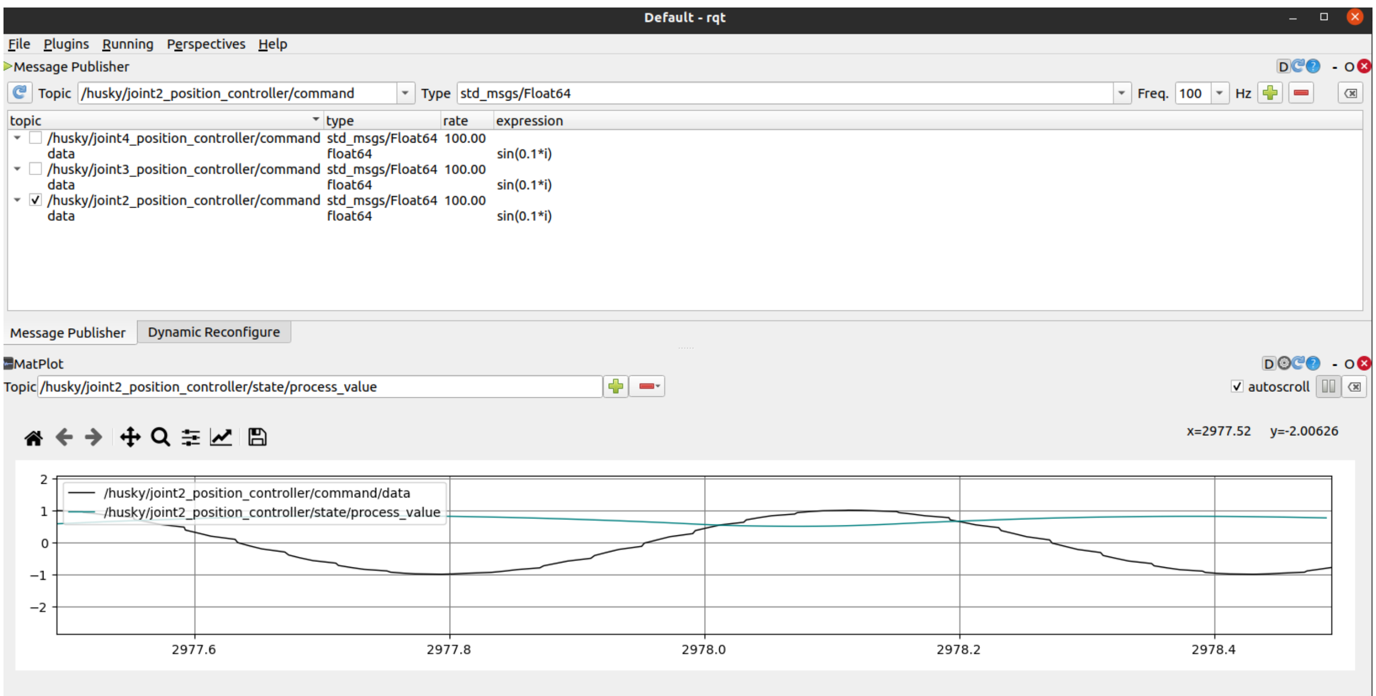
Workspace of robot (Front view)



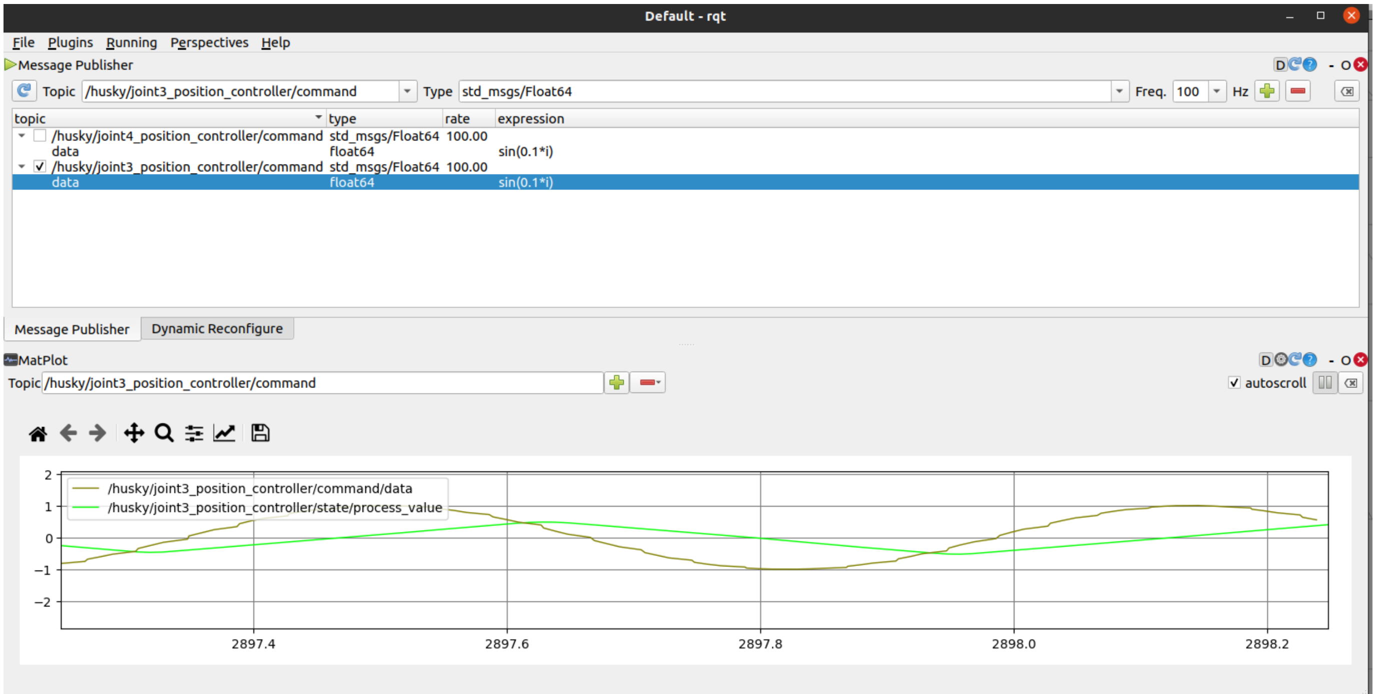
Workspace of robot (Front view)

10. Controllers:

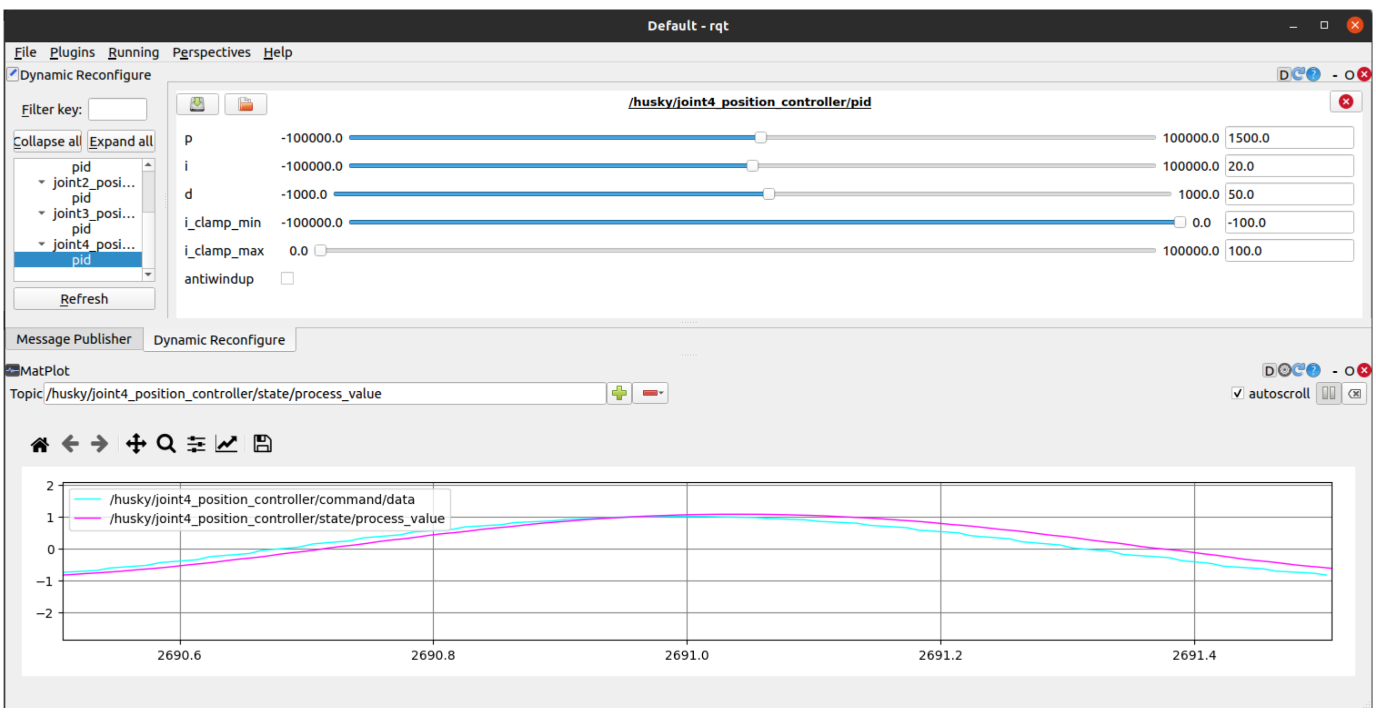
Since the mobile base use is same from Project1, the PID controllers were tuned only for the manipulator joints. For each manipulator joint, an input of sin(0.1*t) was given and the response was plotted using RQT visualizer. The P, I and D values were tuned accordingly so that response output is as close to the input given. Below are the rqt plots for controllers at initial stage and then PID values were tuned manually:



Tuned PID values for joint2: P= 1500 ; I= 15 ; D= 20



Tuned PID values for joint3: P= 1700 ; I= 20 ; D= 60



Tuned PID values for joint4: P= 1500 ; I= 20 ; D= 50

11. Robot Appropriateness for the task:

The proposed robot design is a combination of all terrain wheeled robot and a highly dexterous gripper as the end effector of a 3-DOF manipulator mounted on the robot. With a large wheelbase and balanced structure, the robot can maneuver almost all possible variations of terrain in an agriculture field. It will also provide stability needed to perform harvesting tasks and the 2-finger gripper will allow for accurate grip and firm hold of the object. The detachable basket on the back of the robot will temporarily store the harvested fruit which improves robot efficiency to work round the clock.

12. Model Assumptions:

Our robot model basically consists of 3 major parts (Husky bot, 3-DOF manipulator, Gripper) along with attached sensors. The husky robot will be a 4 wheeled robot capable of translating and rotating in a 2D plane. The manipulator attached will be primarily used to pick fruits and vegetables as commanded by the sensors. The gripper will consist of 3 fingers along with soft grip on the inside in order to avoid any damage caused to the ripened fruits. There are certain assumptions made during working of our robot:

- The Tyers have no slip condition. Other external disturbances are not taken into account.
- All parts obey rigid body mechanics and they do not deform under the action of applied forces.
- Sensors mounted on the robot are noiseless and data acquired through it is accurate.
- The weight distribution of the whole robot assembly is balanced and the Goodpick bot is able to carry the load of the mounted manipulator without having any impact in its trajectory.
- The initial three joints of the manipulator arm are the ones in particular that decide the position of the end effector and the last three decide the orientation.
- Considering the path followed by the robot will be the optimal solution among all the possible solutions it can have after kinematic calculations.
- No offset is present between the joints and all joints of the manipulator are revolute which can revolve 180 degrees.

13. Problems Faced:

- The gazebo simulator didn't provide us the opportunity to fix fruits on top of tree in contrast to other simulators.
- We faced issues while configuring our manipulator in MoveIt as there were issues associated with the moveit launch file in the Gazebo world. The robot did spawn in Rviz but not in Gazebo.
- The PID tuning was very difficult as the weight and inertial components impacted the movement of the manipulator very heavily.

14. Lessons learned:

- Designing of CAD models for a particular robot type along with the part assembly. Creating an URDF file and exporting it. Calculating the DH parameters according to the manipulator geometry.
- Spawning the URDF model into Gazebo simulator. Attaching other parts/components to the main body by adding the dummy link in the URDF. Defining the joint limits.
- Adding the PID controllers to the intended joints and tuning them using rqt plots. Integrating the controller file with the URDF and the launch file.
- Creating a custom Gazebo world according to the application environment of our robot.
- Motion and trajectory planning of the manipulator using the Forward and Inverse kinematics and creating a publisher and subscriber to validate it.

15. Conclusions:

- Successful implementation of GoodPick that can teleop to the required position and scan the object using camera sensor before picking it up using the attached end-effector gripper.
- The Forward and Inverse kinematics were verified using the robot poses and the trajectory plotted by the end-effector.
- The gazebo world was created according to the application environment of our robot.

16. Future Works:

- Attaching depth sensor to get the exact location of fruits.
- Developing image segmentation algorithm for the RGB camera attached.
- Designing and developing new manipulator with more DOF's to increase its reachability in the complex region of its workspace.
- Develop swarm of these and make them communicate with each other to make the process more efficient.

17. References:

1. https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/
2. https://www.theguardian.com/us-news/2022/may/28/robot-agriculture-farming-artificial-intellig ence
3. https://opengrasp.sourceforge.net/
4. Kuznetsova A, Maleva T, Soloviev V. Using YOLOv3 Algorithm with Pre- and Post-Processing for Apple Detection in Fruit-Harvesting Robot. Agronomy. 2020; 10(7):1016. https://doi.org/10.3390/agronomy10071016
5. Zhou, H., Wang, X., Au, W. et al. Intelligent robots for fruit harvesting: recent developments and future challenges. Precision Agric 23, 1856–1907 (2022). https://doi.org/10.1007/s11119-022-09913-3
6. Khare, D., Cherussery, S., Mohan, S. (2021). A Novel Design for an Autonomous Mobile Agricultural Fruit Harvesting Robot. In: Zeghloul, S., Laribi, M.A., Arsicault, M. (eds) Mechanism Design for Robotics. MEDER 2021. Mechanisms and Machine Science, vol 103. Springer, Cham. https://doi.org/10.1007/978-3-030-75271-2_5