ENPM 662: INTRODUCTION TO ROBOT MODELING

Project Report:

SEARCH AND RESCUE BOT

By:
FNU Koustubh (dir id:koustubh)
Aryan S Mishra (dir id:amishr17)

December 2023

TABLE OF CONTENTS — Pg No.

1. INTRODUCTION

SARB or Survey and Rescue Bot is a robotics platform that can be used for various rescue and surveying operations. It consists of a mobile robot with a 3dof arm attached on the top of a mobile base. The robot is also integrated with a LIDAR and depth camera module for surveying the environment. The robot can be particularly helpful in surveying collapsed structures and possibly find trapped victims in confined spaces. The robot arm attached to the robot can be used to manipulate the environment such as operating switches or can be used to drop supplies such as medicine, food or a GPS beacon for the rescue team to locate a victim.

1.1 MOBILE ROBOT

Mobile Robot's integrate electrical and mechanical systems into a vehicular form. Encompassing an extensive suite of software functionalities. While their fundamental designs are often straightforward, typically consisting of a platform mounted on wheels or caterpillar tracks, their software applications engender a plethora of innovative possibilities. By leveraging their advanced sensory capabilities, mobile robots effectively perceive their surroundings, enabling enhanced navigation and control strategies for entire fleets. The fundamental applications of mobile robots in manufacturing encompass the transportation of raw materials, components, tools, and other essential items. Mobile platforms can be programmed to seamlessly transport production items from storage areas to workstations for further processing. Upon completion, mobile robots efficiently convey items to manual quality inspection stations before transporting them to designated storage locations.

1.2 ROBOTIC ARM

A 3-DOF (degree of freedom) robotic arm is a mechanical arm that has three independent axes of rotation. This means that it can move in three different directions: up and down, left and right, and forward and backward. 3-DOF robotic arms are relatively simple and inexpensive, making them popular for a variety of applications, including:

- **Pick and Place:** 3-DOF robotic arms are often used in pick and place applications, such as assembling parts or packaging products.
- **Machine Tending:** 3-DOF robotic arms can be used to load and unload machines, as well as to monitor and maintain equipment.
- **Welding:** 3-DOF robotic arms are commonly used in welding applications, where they can precisely position the welding torch and work-piece.
- **Surgical Procedures:** 3-DOF robotic arms are used in minimally invasive surgical procedures, where they can provide surgeons with greater precision and dexterity.

We are combining the Mobile Robot with a 3-DOF arm. The arm will be mounted upon the Mobile Robot.

2 APPLICATIONS

Mobile robots equipped with a 3-DOF arm offer a versatile combination of mobility and manipulation capabilities, making them suitable for a wide range of applications :

2.1 PRIMARY APPLICATION:

Search and Rescue: Mobile robots with 3-DOF arms can navigate hazardous environments, conduct searches, and manipulate objects, making them ideal for locating and rescuing survivors in disaster zones.
The aim of our project is to simulate a world, where there is a certain building structure and the robot has to pick up an emergency beacon and drop it at a location where the rescue is needed. Apart from this general application, the robot is equipped with mapping hardware such as a LIDAR and a IntelRealSense depth camera that can be used to map the environment in the future by adding the required software plugins.
SARB platform apart from being a rescue and survey platform can also be used for a variety of applications as listed below.

2.2 SECONDARY APPLICATIONS:

<u>Here are some of the possible applications that SARB can be used for:</u>

• Manufacturing and Assembly: Mobile robots with 3-DOF arms can automate repetitive tasks in manufacturing and assembly lines, such as pick-and-place operations, part feeding, and tool handling.
• Warehouse and Logistics: In warehouses and logistics centers, mobile robots with 3-DOF arms can handle material handling tasks, such as loading and unloading pallets, transporting goods between storage areas, and
assisting in order fulfillment.
• Service Robotics and Hospitality: Mobile robots with 3-DOF arms can be used in service industries, such as hospitality and retail. They can perform tasks such as delivering food, cleaning tables, and restocking shelves, enhancing customer service and reducing labor costs.
• Healthcare and Assisted Living: In healthcare settings, mobile robots with 3-DOF arms can assist with patient care, medication administration, and specimen handling. They can also provide companionship and support for elderly individuals in assisted living facilities.
• Agriculture and Precision Farming: Mobile robots with 3-DOF arms can be deployed in agriculture for precision farming applications. They can perform tasks such as delicate plant manipulation, sample collection, and targeted pesticide application, reducing labor costs and improving agricultural productivity.
• Maintenance and Repair: In industrial settings, mobile robots with 3-DOF arms can assist in maintenance and repair tasks. They can reach hard-to-access areas, perform tool

manipulation, and carry out tasks such as valve turning, component replacement, and equipment inspection.

3 Robot Type

The robot is a mobile robot with four wheels attached to a chassis. The front wheels have independent joint control that can be used for steering the robot. The rear two wheels have the drive capability for moving the robot.
There is a three dof arm attached to the robot in RRR configuration. Three revolute joints (or rotating joints) are arranged in series to form the basic kinematic structure in this configuration. Each joint permits rotation about an axis, and the configuration is frequently described as having a shoulder, elbow, and wrist joint, similar to the construction of the human arm.
The arm is also equipped with a vacuum gripper as an end effector that can enable it to pick any object or tool.The vacuum gripper is a reliable option for our application as it is fast, reliable as well as can be used to handle delicate objects like glass, electronics or food items.
The sensor array on the robot consists of a lidar and Intel realsense camera.

4 Dof And Dimensions

The robot has a total of six degrees of freedom, three from the mobile manipulator base and three from the manipulator arm. There is an additional degree of freedom available for the wrist joint in the vacuum gripper end effector but for keeping the simplicity of the robot, it is locked to represent a fixed joint.

The dimensions of the robot are as follows:

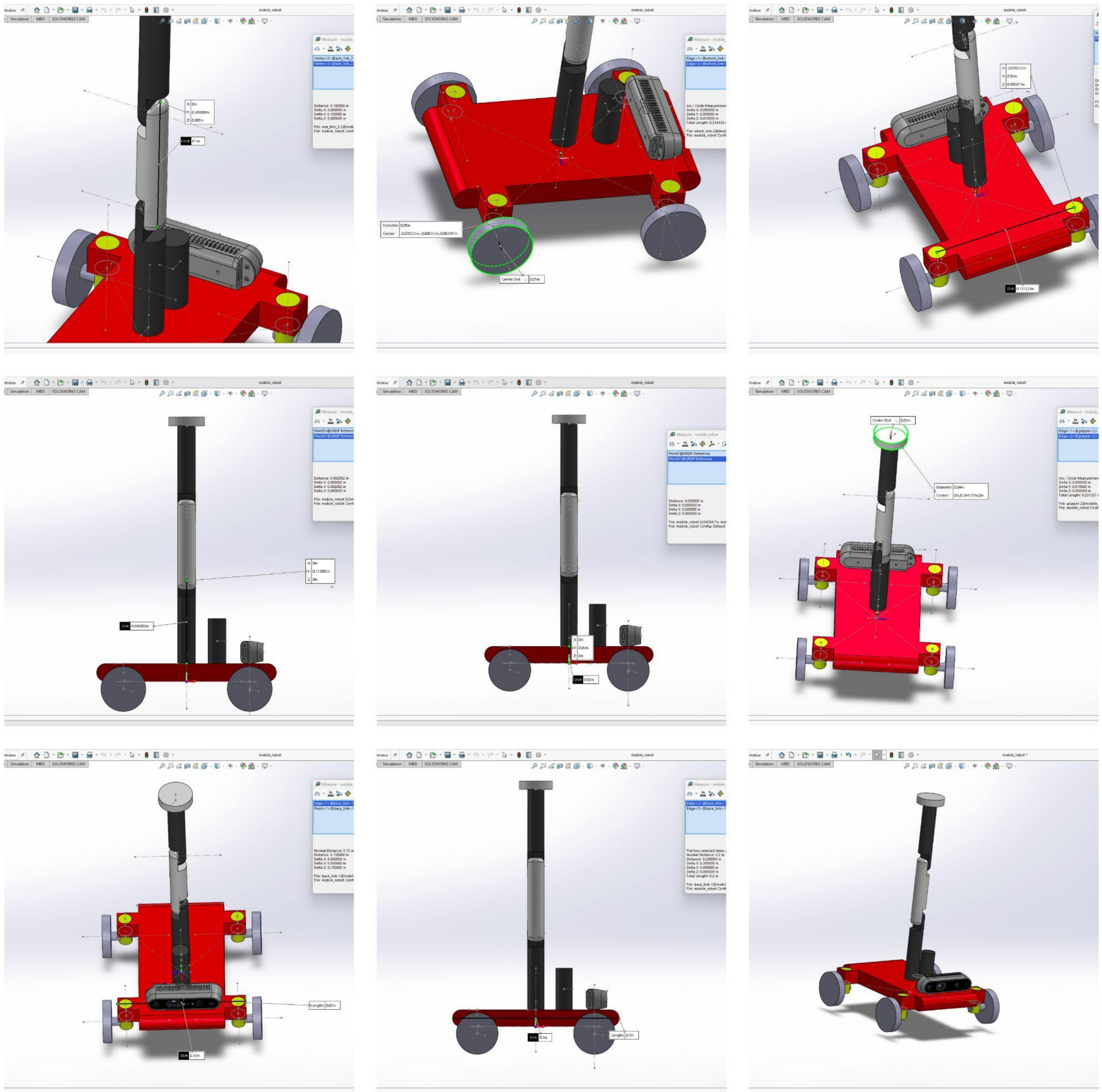
Dimensions of the mobile robot:
Length of the chassis: 0.1 m
Width of the chassis: 0.15 m
Height of the mobile base (including wheel assembly): 0.0532 m
Diameter of the wheel: 0.05 m
Width of the wheel: 0.01 m

Dimensions of the manipulator:
Length of the arm1: 0.092 m
Length of the arm2: 0.080 m
Length of the arm3: 0.092 m

Dimensions of the vacuum gripper:
Diameter of the vacuum gripper: 0.05 m

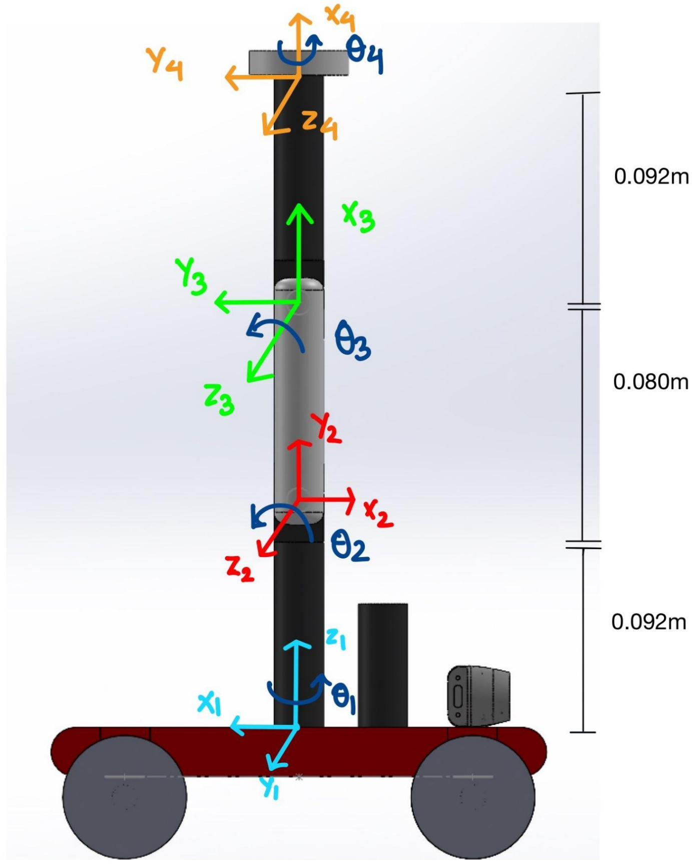
Width of the vacuum gripper: 0.01m

5 CAD MODEL



6 D-H Parameter

Frame:



Link	a	d	α	θ
1-2	0	0.092	$\frac{\pi}{2}$	$\theta_1 + \pi$
2-3	0.08	0	0	$\frac{\pi}{2} + \theta_2$
3-4	0.092	0	0	θ_3

7 Forward Kinematics:

7.1 Forward Kinematics Of The Mobile Base:

The forward kinematics of the mobile base is given by the following python code. The following equations are used to calculate the x and y coordinates of the robots

$$V_x = (\omega r)cos(\psi)$$
$$V_y = (\omega r)sin(\psi)$$

Where,
V_x = Velocity along the x direction
V_y = Velocity along the y direction
ω = Angular velocity of the wheel
r = Radius of the wheel
ψ = Yaw of the robot

```python
x = msg.orientation.x
        y = msg.orientation.y
        z = msg.orientation.z
        w = msg.orientation.w

        self.yaw = math.atan2(2 * (w * z + x * y), 1 - 2 * (y*y + z*z))
        self.x_vel = (self.wheel_angular_velocity*0.025)*math.cos(self.yaw)
        self.y_vel = (self.wheel_angular_velocity*0.025)*math.sin(self.yaw)
        self.x += self.x_vel*(1/100)
        self.y += self.y_vel*(1/100)
```

7.2 Forward Kinematics Of The Mobile Manipulator:

The forward kinematics of the robot manipulator is given by using the transformation matrices. The transformation matrices for the manipulator are set using the following python code

```
Python
def homo_matrix(self,a,d,alpha,theta):
        T=sp.Matrix([[sp.cos(theta),-sp.sin(theta)*sp.cos(alpha),sp.sin(theta)*sp.sin(alpha), a
* sp.cos(theta)],[sp.sin(theta), sp.cos(theta)*sp.cos(alpha),-sp.cos(theta)*sp.sin(alpha), a *
sp.sin(theta)],[0, sp.sin(alpha), sp.cos(alpha), d],[0, 0, 0, 1]])
        return T
```

The final transformation matrix comes out to be:

8 INVERSE KINEMATICS:

To calculate the Jacobian matrix we first extract the translation matrix(x, y, z) and then partial differentiate the x, y, and z values with qi
After this, we calculate the Zi column vectors from the transformation matrices and put them in the Jacobian matrix.

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

The Jacobian matrix calculated is by using the following python code:

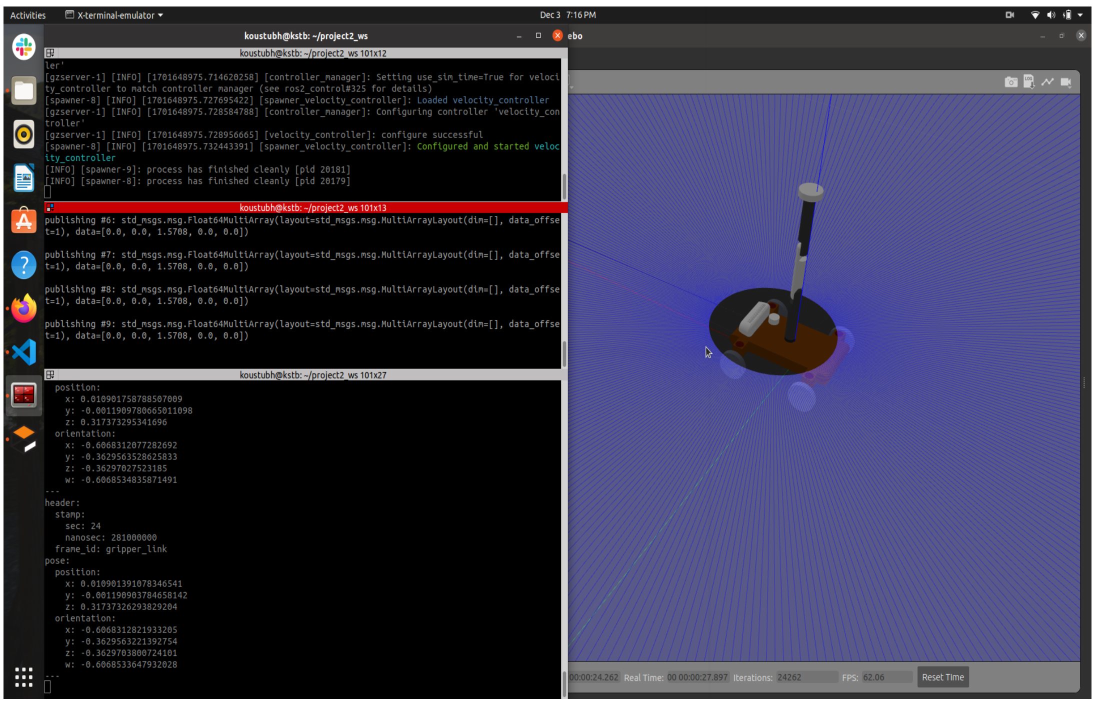
```
Python
J_w =
sp.Matrix([[T_01[0,2],T_02[0,2],T_03[0,2]],[T_01[1,2],T_02[1,2],T_03[1,2]],[T_01[2,2],T_0
2[2,2],T_03[2,2]]])
P_x = T_03[0,3]
P_y = T_03[1,3]
P_z = T_03[2,3]
J_v = sp.Matrix([[0,0,0],[0,0,0],[0,0,0]])
J_v[0,0] = sp.diff(P_x, theta1)
J_v[1,0] = sp.diff(P_y, theta1)
J_v[2,0] = sp.diff(P_z, theta1)
J_v[0,1] = sp.diff(P_x, theta2)
J_v[1,1] = sp.diff(P_y, theta2)
J_v[2,1] = sp.diff(P_z, theta2)
J_v[0,2] = sp.diff(P_x, theta3)
```

```
J_v[1,2] = sp.diff(P_y, theta3)
J_v[2,2] = sp.diff(P_z, theta3)

J = sp.Matrix.vstack(J_v, J_w)
```

9 Forward Kinematics Validation:

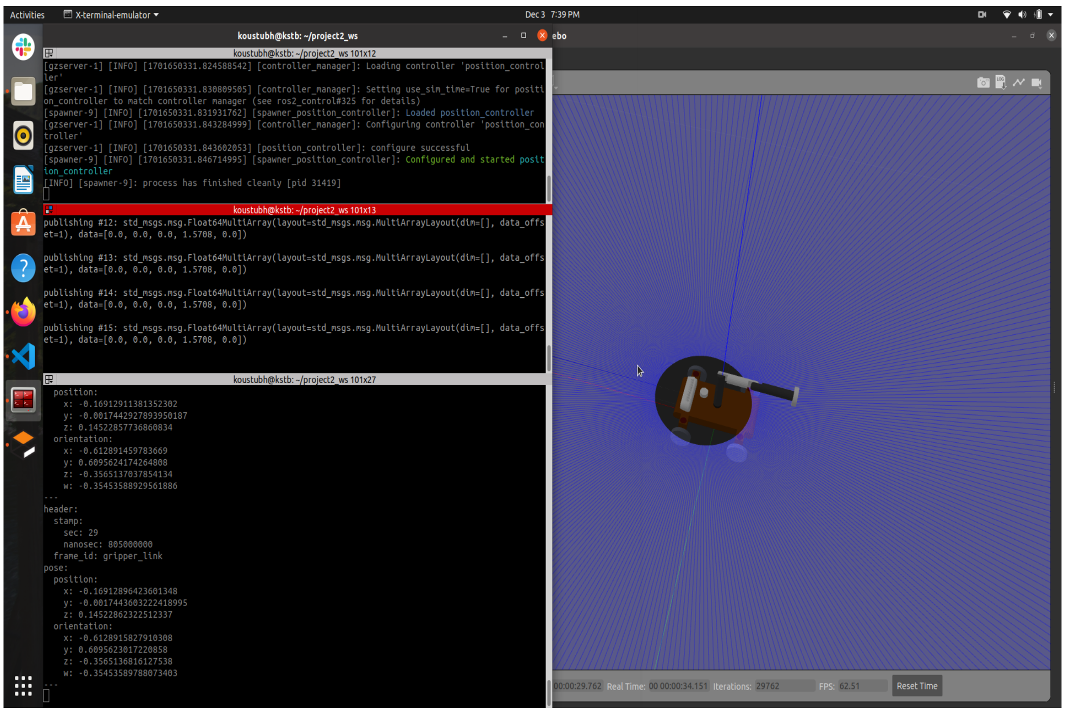
9.1 Joint 1 (θ_1) is rotated by an angle of $\pi/2$



For geometric validation, when theta1 is 90 degrees, the arm only rotates on its axis and the x, y, z coordinates of the end effector remain the same. The z coordinate is calculated as height of mobile base and sum of all the arm link lengths as 0.0532+0.092+0.08+0.092 = 0.3172

The same is verified by the odometry plugin as x = 0, y = 0 and z = 0.3172

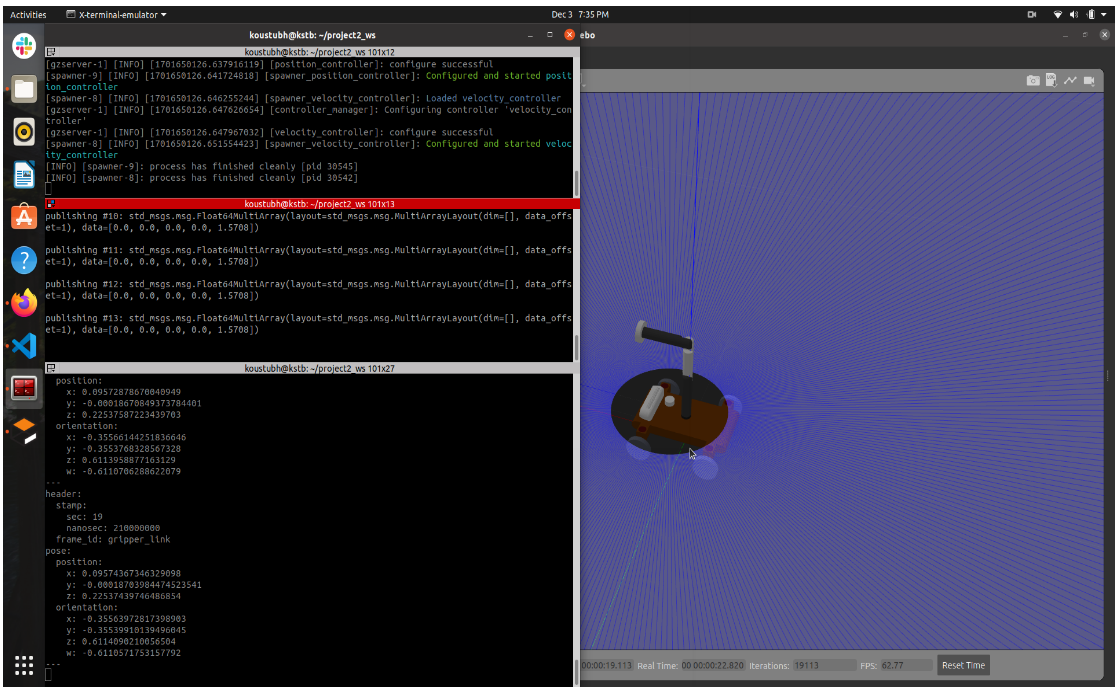
9.2 Joint 2 (θ_2) is rotated by an angle of $-\pi/2$



In this case the whole arm2 and arm3 links are aligned along the negative x axis. As a result the end effector travels a distance of 0.08+0.092 = 0.172 in the negative x direction and z coordinate as sum of mobile base height and arm1 link length as 0.0532+0.092 = 0.1452 .
Hence, the overall coordinates of the robot end effector should be as x = -0.172, y = 0, z = 0.1452.

The same is verified by the odometry plugin.

9.3 Joint 1 (θ_3) is rotated by an angle of $\pi/2$



When the third joint is rotated by 90 degrees, the arm 3 link is aligned with the positive x axis and the arm 1 and arm2 are still aligned with the z axis. Hence the z coordinate is the sum of mobile base height, and arm 1 and 2 link lengths as 0.0532+0.092+0.08 = 0.2252 and x coordinate is the length of arm 3 as 0.092. Hence, the final coordinates of end effector are x = 0.092, y = 0 and z = 0.2252

The same is verified by the odometry plugin.

```
position:
  x: 0.09574367346329098
  y: -0.00018703984474523541
  z: 0.22537439746486854
orientation:
  x: -0.35563972817398903
  y: -0.35539910139496045
  z: 0.6114090210056504
  w: -0.6110571753157792
---
```

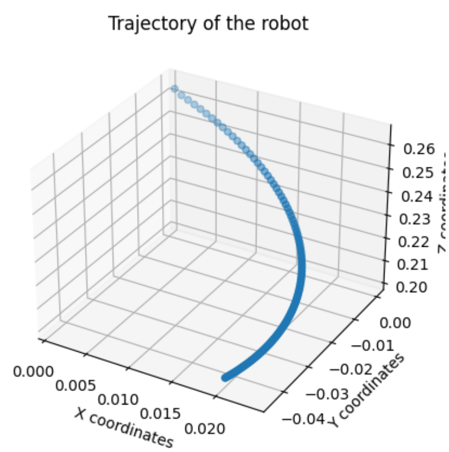
10 Inverse Kinematics Validation

The robot end effector starts with the initial position of x = 0, y = 0 and z = 0.3172. Our goal is to make the end effector reach a position required to pick up a GPS beacon placed on a box at (0,0,0.225).

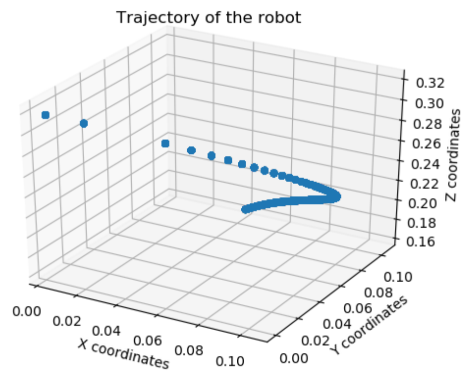
The tool should follow a semi circular path to reach the object. In order to reach the target, we assumed that the end effector should be given some fixed velocities for a certain amount of time to reach the target.

The required tool velocities come out to be as V_x = 0.092 m/s and V_z = -0.092 m/s for duration of 10 seconds.

We ran the simulation in an independent python script as well as in ROS2, the robot manipulator was successfully able to reach the desired destination in the stipulated time. The output trajectory from the python script is shown below



The output trajectory from the actual robot end effector in Gazebo is as follows:



Hence, we can conclude that the inverse kinematics are working correctly for the robot.
The video for the motion of the arm can be found here
https://www.youtube.com/watch?v=TTmBDSYedSo

11 Workspace Study

Since we have used an articulated manipulator on the top of the mobile robot. The workspace of the robot is given below

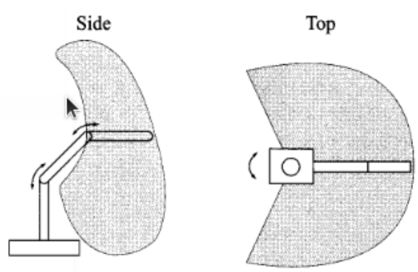


Figure 1.10: Workspace of the elbow manipulator. The elbow manipulator provides a larger workspace than other kinematic designs relative to its size.

Pic Source: Spong and Hutchinson [2]

The maximum reach of the robot in the z direction(the axis pointing towards the top) is 0.3172. Because of a mobile base, the robot can reach almost any position in x and y directions as long as the robot has the space to orient itself correctly with respect to the target coordinates.

12 Assumptions

The practical application of the SARB requires it to be able to navigate any type of terrain, we have assumed that using the current mobile platform it will be able to navigate the terrain.

While designing the model in Solidworks, we have taken the material of every link to be PE Low Density Film, so as to keep the model physics simple and lightweight. This would not be true in a real life scenario, where the different parts of the robot would require different materials to be used.

In Gazebo simulation, we have also ignored any self collisions that the robot has with itself. Although, for the demonstration purpose, we have planned the trajectory so that the robot avoids collisions with its own link. In a real life scenario, this would need a planner to avoid any collisions while simultaneously following the best path required.

Also, for the purpose of simulation, we have assumed that for goal navigation a simple P controller will suffice and can produce excellent results. But in a real life scenario, a PID controller might be required so as to minimize the oscillations and any steady state errors associated with the hardware used.

Talking about the end effector, we have assumed that the object can be always picked up using a vacuum gripper, this might not be true in practice, as the robot might have to pick up objects that might not align with the end effector or might have contours that do nor necessarily stick to the gripper.

13 CONTROL METHOD

The SARB has two separate control mechanisms for its two different parts, the mobile base has a closed loop P controller that can be used for goal navigation and the arm has used a position controller from [1] package. The ros_control plugin does have an inbuilt PID controller for the position control and the gains can be tuned.
For our application, we are using the default configuration of the position controller.

The PID controller would be used

14 GAZEBO AND RVIZ VISUALIZATION

The simulations for the robot are as follows.

14.1 Robot goal pose navigation

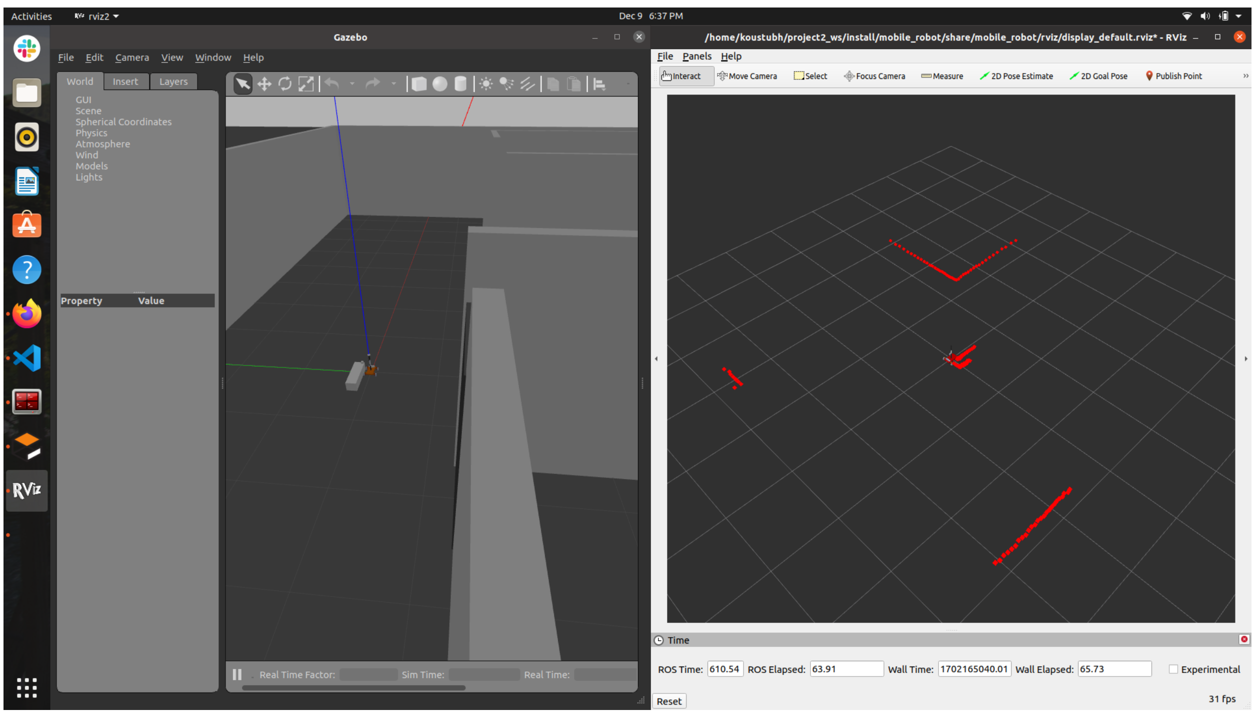
In this mission, the robot is given a goal coordinate of x=5 and y=6. The robot navigates to the desired position using a closed loop controller. Please find the url for the video below.
https://www.youtube.com/watch?v=B1DWkmJjCEA&feature=youtu.be

14.2 Robot Pick and Place

The robot implements the movement of pick and place where the robot is programmed to pick up a block on the table and drops at a location. Please find the url for the video below.
https://youtu.be/s4FvZ7kkXAc

14.3 Robot Visualisation in Rviz

The robot has a LIDAR and a depth camera for visualizing the environment. The Rviz visualization for the robot is as shown below.



In the image above, we can see the laser data being visualized wherever there is an obstacle.
The sensors can be used for SLAM and autonomous navigation in the future.

15 PROBLEMS FACED

We faced the following problems while implementing the project:

- While exporting the URDF from solidworks, the origins and the axis of the links were not generated correctly. To fix the problem, we completely removed any reference geometry that was present in the model and generated the origins and the axis of rotation using the auto-generate feature of the URDF exporter tool.

- The gazebo environment was not launching properly due to error in the spawn entity process. To fix the issue we verified and corrected the indentation and any errors in the xml tags in the urdf file.
- The odometry plugin was not working properly and there was no message being published to the "/odom" frame for the end effector pose. To fix the issue, we made the end effector joint to the arm a "revolute" joint, instead of a fixed joint. After making the joint as revolute, the odometry plugin started to work correctly.
- The launch files do not launch the simulation properly every time. The gazebo often freezes and displays a black screen. To circumvent this issue, we used the commands "killall gzserver" and "killall gzclient". This is an inherent problem with the gazebo that comes with the ROS2 Galactic distro.
- While launching the ros_slamtoolbox, the map data was not being published to the topic because of transformation error between the frames.

16 LESSONS LEARNED

The project helped us learn the various aspects of Robot Modelling such as CAD design, URDF, DH parameters, Forward and Inverse Kinematics, ROS2 and Gazebo simulations.

- **CAD modeling:** The project helped us to understand the intricacies of designing a real life bot which in reality can be achieved by designing simple parts and assembling them to get the desired model.
- **UDRF :** Exporting the URDF from the CAD Model. URDF enables us to define various kinds of joints that are very commonly used in Robotics.
- **DH Parameters :** Enabled us to understand the complexities of Robot joints and frames by utilizing concepts to attain the transformation matrices.
- **Forward and Inverse Kinematics:** Learned to derive the forward kinematic equations from transformation matrices and Inverse kinematic equations by understanding the concept of Jacobian.
- **ROS2 and Gazebo:** provide invaluable robotics lessons by emphasizing communication protocols for seamless hardware and software integration, demonstrating distributed system architectures to handle complex tasks across multiple nodes, enabling accurate simulation and testing of robots in various environments, fostering understanding of robot behavior and control mechanisms, providing tools for debugging and visualization, and emphasizing the importance of community support. Mastering these platforms entails fundamental principles in robotics, system design, communication paradigms,

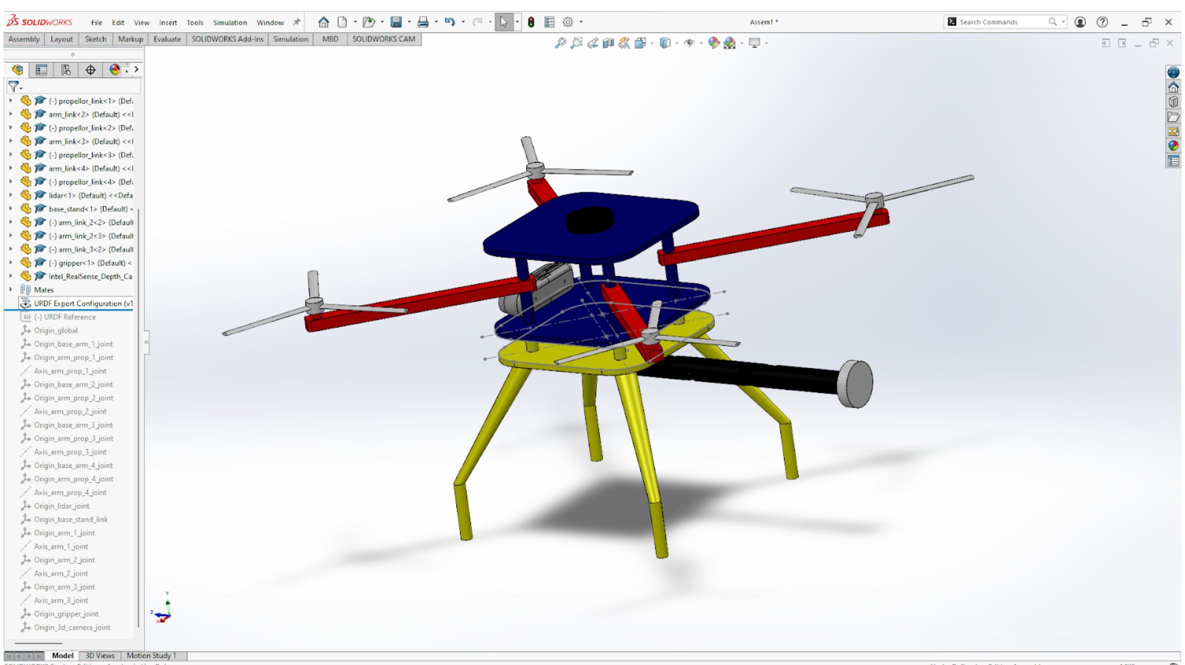
and simulation methodologies necessary for constructing resilient and efficient robotic systems.

17 Conclusion

The Search and Rescue bot(SARB) is a beacon of innovation in the field of robotics, thanks to its adaptive design and versatile capabilities. Its modular nature allows it to be customized to suit distinct contexts, allowing it to overcome obstacles, pick up and precisely drop goods. With mapping and obstacle recognition capabilities, this versatile bot exhibits efficiency and dependability in critical situations. As an embodiment of optimism in rescue missions, its mobility and precision ensure that it is at the cutting edge of technical innovation in assisting and improving search and rescue operations.

18 Future works

The SARB project is supposed to be a modular and customisable platform of mobile robots, with variations of ground based, air based and water based mobile robots.Many other types of sensors such as night vision/IR cameras, gas leakage sensors, and geiger counters can be integrated in the robot.A prototype SARB quadcopter with the same sensor array and manipulator arm repurposed from the ground based mobile robot as shown in the gazebo simulation.



19 References

1. Gazebo : Tutorial : ROS control
2. Project 1 Documentation — ENPM 662
3. Robot Modeling and Control, 2nd Edition | Wiley