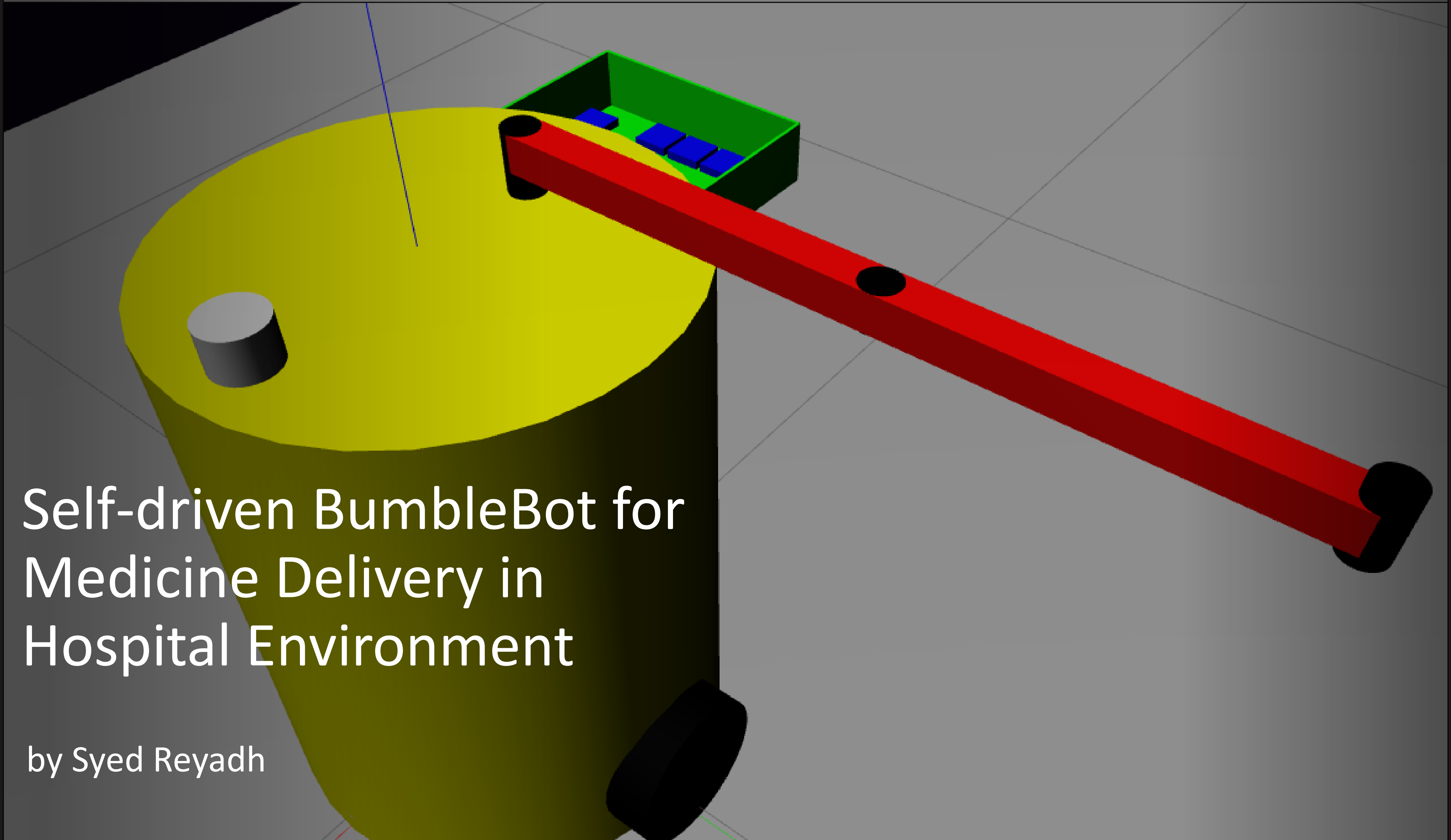


Self-driven BumbleBot for Medicine Delivery in Hospital Environment

by Syed Reyadh

Introduction & Background

- The Self-driven BumbleBot delivers medications to hospital patients autonomously, addressing the need for safe, efficient, and reliable medical supply delivery.
- Autonomous robots offer a feasible solution to the time-consuming, error-prone, and hazardous traditional delivery methods, with their rapid advancements in robotics technology.
- It is a leading-edge breakthrough that represents cutting-edge technology in the healthcare sector.
- The simulation robot, equipped with advanced sensors, mapping tech, and autonomous navigation, can safely deliver supplies to their destination in complex hospital environments.
- It navigates and delivers medical supplies in difficult hospital environments safely and effectively by utilising the most recent robotics, AI, and mapping technology.

Technical Content Methodology

Methods for Data Collection

- User feedback
- Historical data analysis
- Sensor data collection
- Performance monitoring
- Simulation and testing

Used software

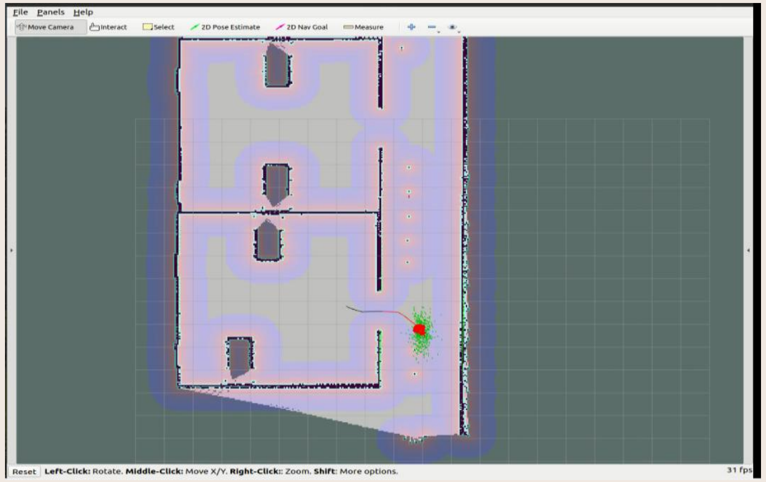
- Rviz
- Gazebo
- SweetHome3D
- Slam gmapping package for map creation
- Python coding for robot operation

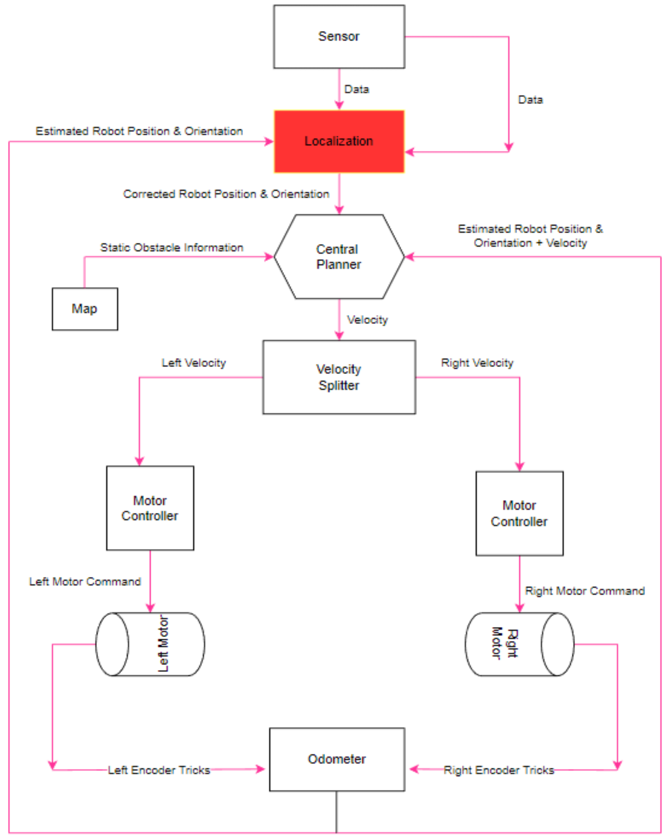
Used methods

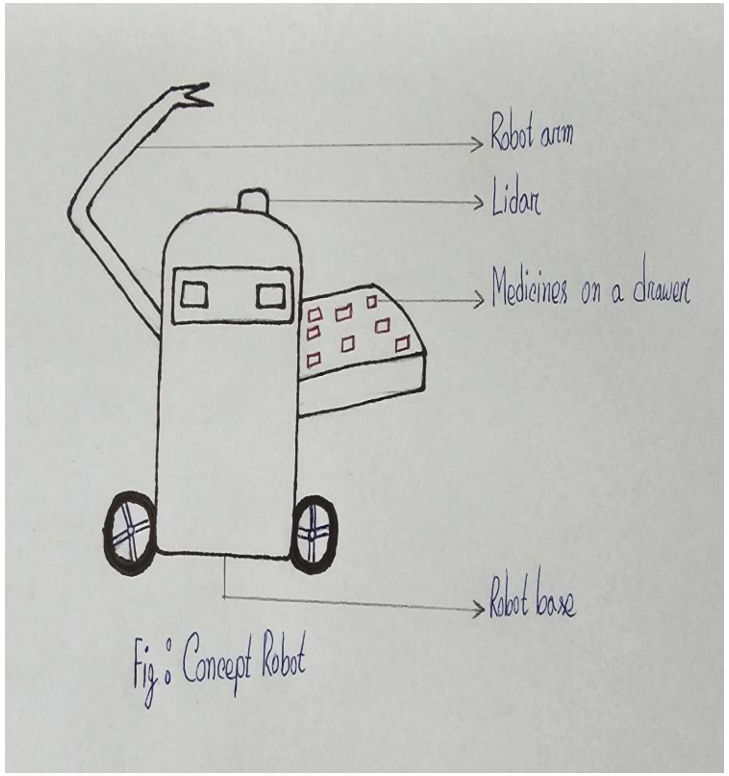
- ROS
- 3D Design
- Perception
- Navigation
- Localization and mapping
- Path planning
- Grasping and manipulation
- Task management

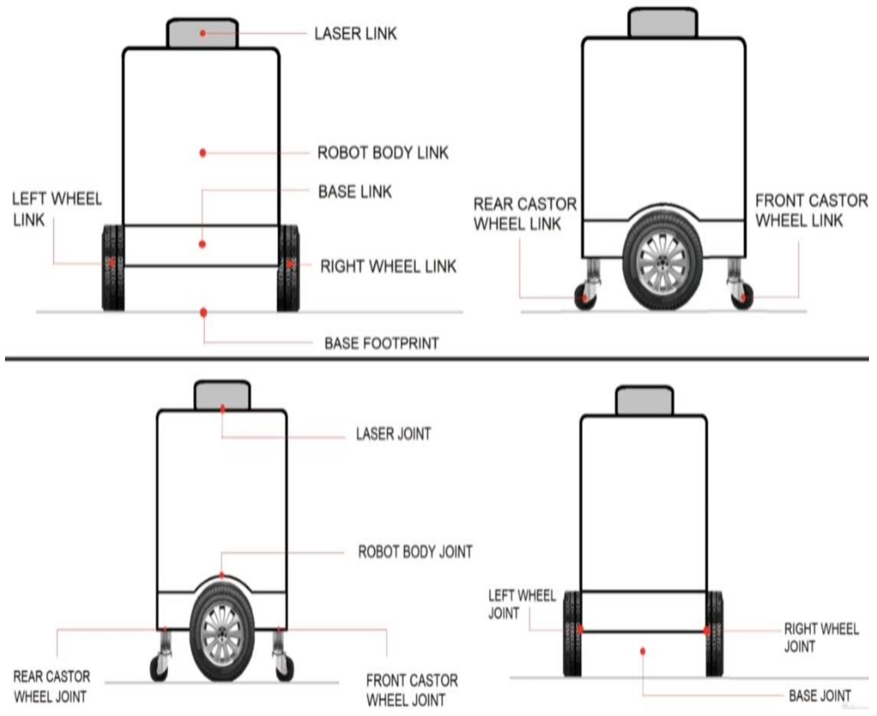
Reasons for choosing used software and methods

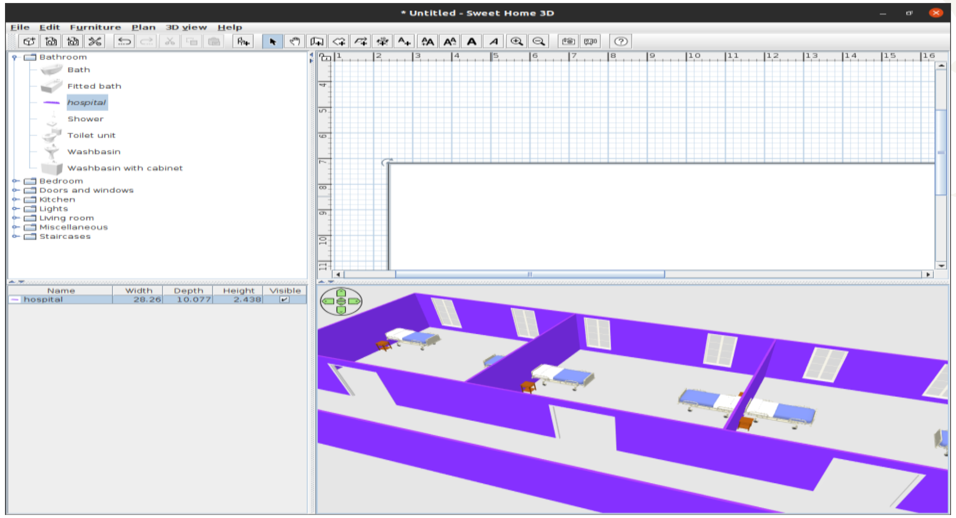
- ROS was the whole operating system.
- Rviz and SweetHome3D were used for robot and hospital world designing
- Navigation to move the robot from one point to another and it involves the usage of lidar to create a map.
- Path planning is the method to determine most efficient path between two points.
- Mapping was used to create a map of the wards and later it was used for robot’s navigation and localization.
- Task management was used to enable the robot to manage it’s tasks.
- Grasping and manipulation method was used to pickup the medicine and put it on the table.



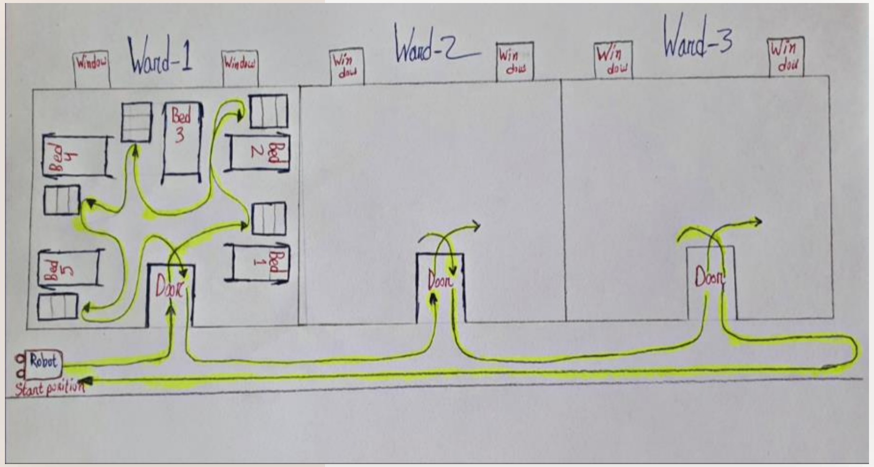


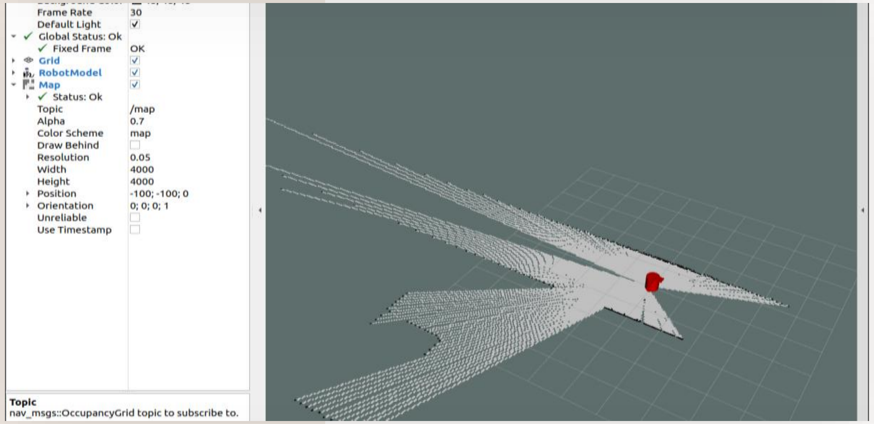






Robot and Environment Design





Path Planning and Mapping

- During the initial stage a hand-written path was made to get the idea.
- According to the plan 5 beds in a ward wasn't used in actual simulation to avoid complexity.
- "Slam gmapping" package is used to create a map of the whole gazebo world and Rviz displayed the world.
- The navigation stack provided by ROS was used and configured according to the hospital ward.

Some Performance Functions on Python

```python
def goal(a,b,c):
    goal = MoveBaseGoal()
    goal.target_pose.header.frame_id = "odom"
    goal.target_pose.header.stamp = rospy.Time.now()
    goal.target_pose.pose.position.x = a
    goal.target_pose.pose.position.y = b
    quaternion = tf.transformations.quaternion_from_euler(0.0, 0.0, c)
    goal.target_pose.pose.orientation.x = quaternion[0]
    goal.target_pose.pose.orientation.y = quaternion[1]
    goal.target_pose.pose.orientation.z = quaternion[2]
    goal.target_pose.pose.orientation.w = quaternion[3]

    client.send_goal(goal)
    wait = client.wait_for_result()
    if not wait:
        rospy.logerr("Action server not available!")
        rospy.signal_shutdown("Action server not available!")
    else:
        return client.get_result()
```

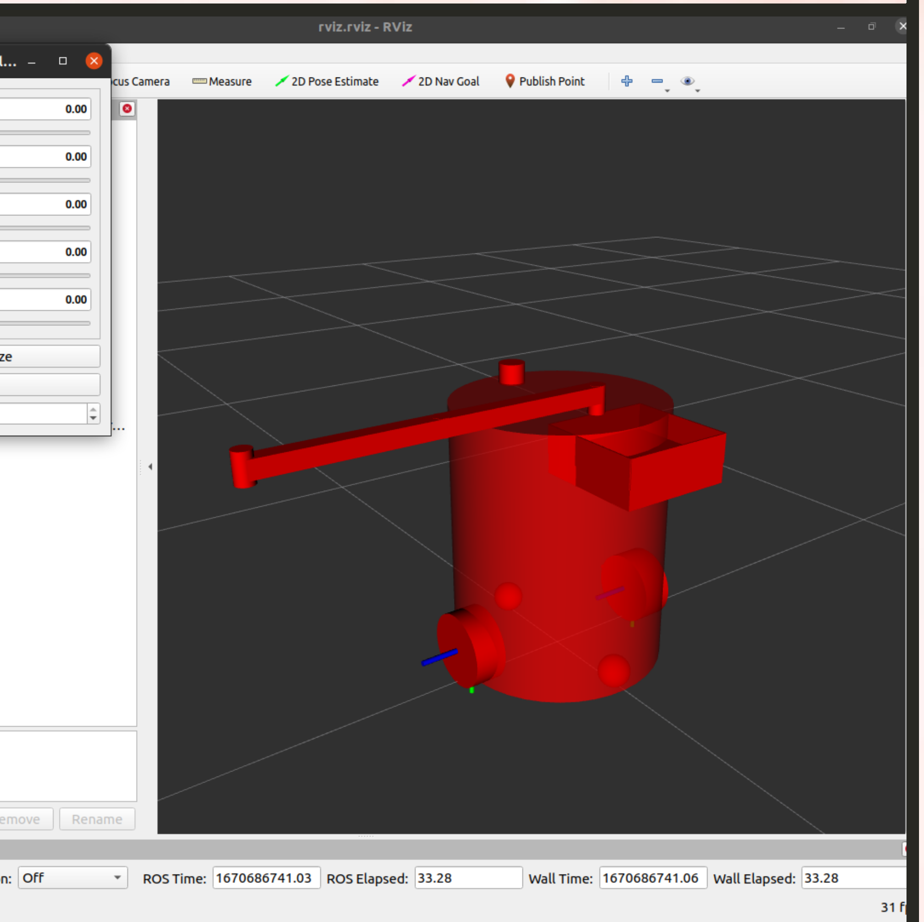
Image-1

A python script is written where a sequence of different tasks was programmed. Image-1 shows the goal function and Image-2 displays the carry function for the robot to deliver the medicine.

```python
def carry(th1,th2,medicine):
    for x in range(300):
        grip = rospy.ServiceProxy('/gazebo/set_model_state', SetModelState)
        arm = rospy.ServiceProxy('/gazebo/get_link_state', GetLinkState)
        state = arm("hospital_robot::link_3","")
        state_msg = ModelState()
        state_msg.model_name = medicine
        state_msg.pose.position.x = state.link_state.pose.position.x
        state_msg.pose.position.y = state.link_state.pose.position.y
        state_msg.pose.position.z = state.link_state.pose.position.z
        grip_state = grip(state_msg)
        pub_1.publish(th1)
        pub_2.publish(th2)
        rospy.sleep(0.01)
```

Image-2

Results



What is done so far-

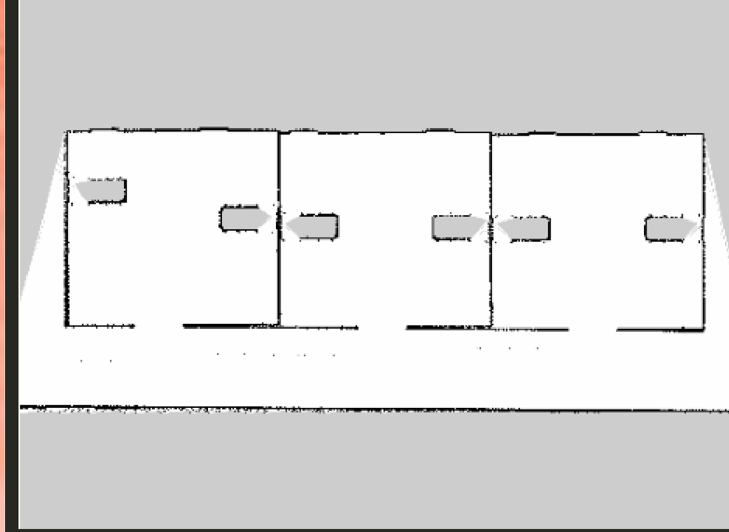
- A simulation delivery robot
- A simulated hospital environment
- A complete map for navigation

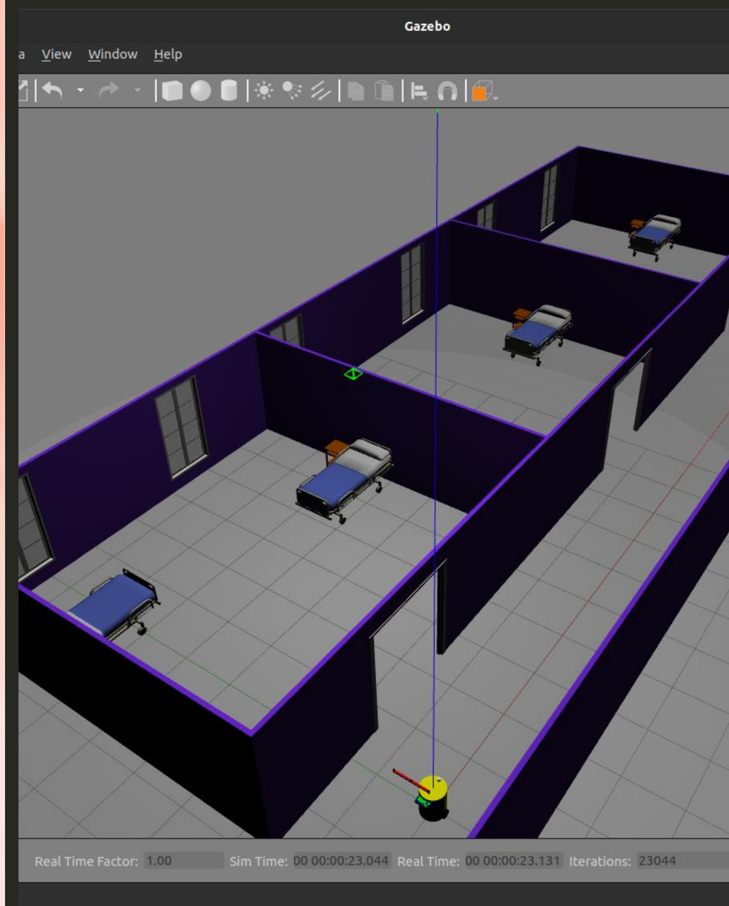
Key findings-

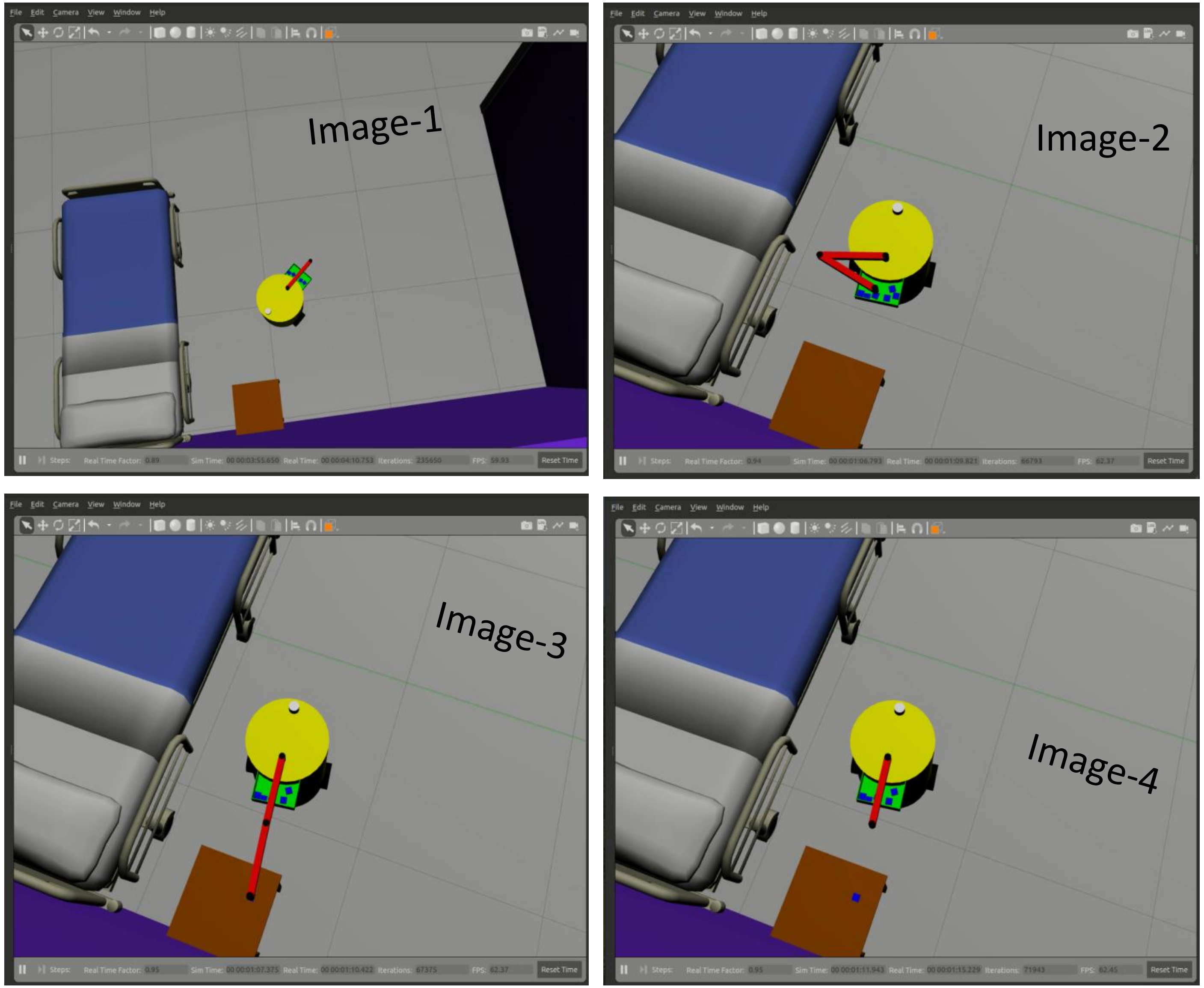
- Robotics technology(ROS)
- Healthcare delivery
- Safety and risk management
- Data analysis and management

Datas still needed to collect-

- Patients data
- Environment data
- User data
- Performance data





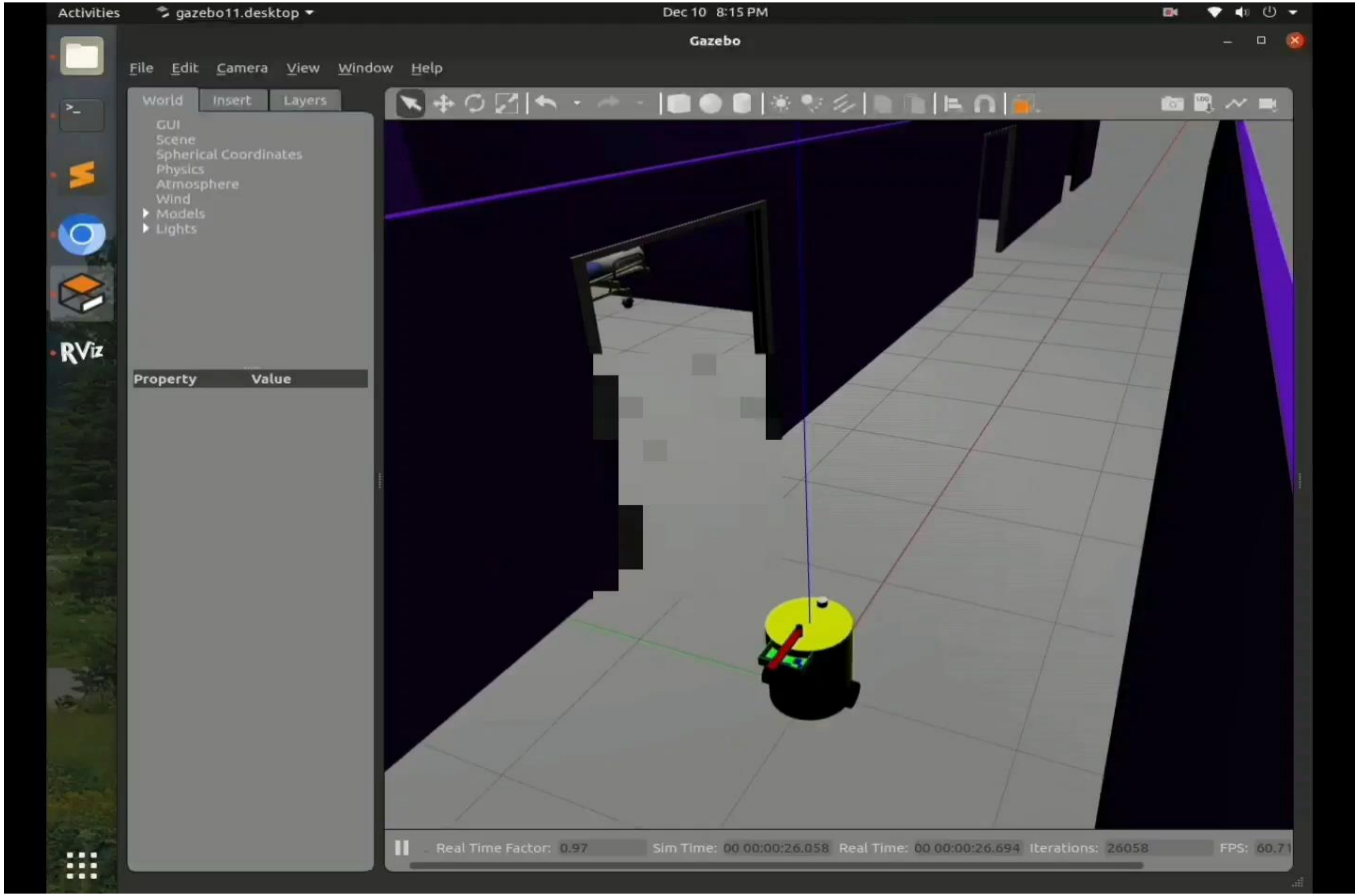


Results

Robot operation

- Image-1 shows the robot is approaching to its first goal
- On Image-2 the robot has figured out its position and has started delivering
- Image-3 shows the robot is delivering the medicine.
- Lastly on Image-4 the robot is leaving the current location to its next location for doing the same task
- The robot repeats the process for all the beds in the three wards and then returns back to its starting point.

Results



Video Demonstration

Evaluation

1. The project was successfully completed with a noble goal and am confident in its real-life implementation.

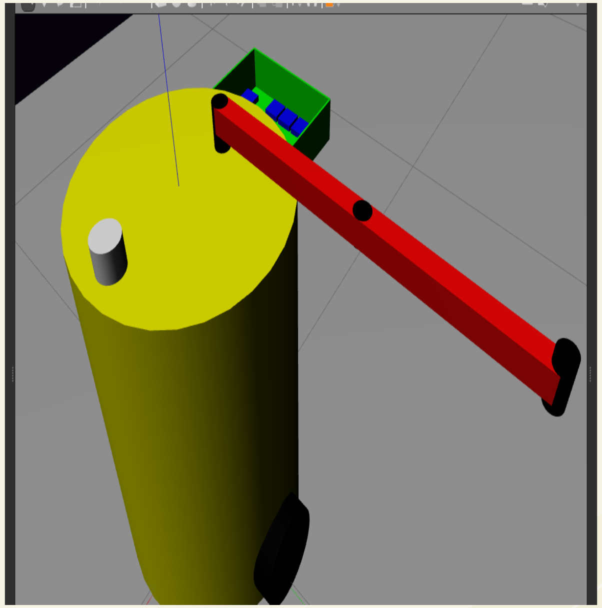
2. During this project evaluation criteria like functionality, efficiency, safety, user interface and cost were kept under consideration. My project was implementation of ROS in a hospital environment. It would be a cost-effective solution if implemented in the actual environment.

3. I relied on the knowledge of healthcare experts, robotics engineers, and researchers with experience in the field of healthcare robotics to ensure the robustness of my findings.

4. Any project, even those integrating robotics and healthcare, may experience unforeseen problems that have an impact on its viability. The simple, cost saving and efficient approach for this project has made it different in its own class.

Some of future development for this project includes-

- An actual delivery robot that can do the task in an actual hospital.
- Integration with electronic health record system.
- Usage of more advanced sensors and algorithms.
- Making the robot multi-tasker.
- Increasing ability to deliver more beds.
- Able to implement more customize features.



Reference

- NHS staff shortages put long-term vision for primary and community care at risk. Available at: https://www.health.org.uk/news-and-comment/news/nhs-staff-shortages-put-long-term-vision-for-primary-and-community-care-at-risk (Accessed: 23 October 2022).
- Workplace robotics safety. Available at: https://en.wikipedia.org/wiki/Workplace_robotics_safety (Accessed: 23 October 2022).
- AGV in Hospitals. Autonomous Mobile Robots Disrupting Healthcare Automation. Available at: https://www.agvnetwork.com/Automation-Hospitals-AGV-Autonomous-Mobile-Robots (Accessed: 01 November 2022).
- This Incredible Hospital Robot Is Saving Lives. Available at: https://www.wired.com/2015/02/incredible-hospital-robot-saving-lives-also-hate/ (Accessed: 02 November 2022).
- Panasonic Autonomous Delivery Robots - HOSPI - Aid Hospital Operations at Changi General Hospital. Available at: https://news.panasonic.com/global/topics/4923 (Accessed: 04 November 2022)
- Building a Visual Model with URDF from scratch. Available at: http://wiki.ros.org/urdf/Tutorials/Building%20a%20Visual%20Robot%20Model%20with%20URDF%20from%20Scratch (Accessed: 15 December 2022)
- Ros.org(navigation). Available at: http://wiki.ros.org/navigation (Accessed: 04 December 2022)
- Using slam navigation. Available at: http://wiki.ros.org/cob_tutorials/Tutorials/Navigation%20%28slam%29 (Accessed: 04 December 2022)