Robotics-Based Medication Delivery System in Hospitals

BY:

Vedansh Surjan
Margi Nasit
Vrunda Patel

Problem Statement

- Health Center face challenges such as staff shortages, high workloads, and time-intensive tasks like fetching equipment or delivering medication.

- The goal is to automate repetitive tasks using robotics to improve efficiency and reduce human error.

Key Challenges

- Navigating crowded and dynamic health center environments.
- Reliable object navigation including key challenges such as inventory management and delivering items.

Proposed Solution

Core Idea:

An autonomous robotic system designed to:

- Navigate health center environments efficiently.
- Manage storage and dispatch of medical supplies and equipment.
- Integrate seamlessly into health center workflows.

Key Features:

- SLAM Mapping: Simultaneous Localization and Mapping for real-time navigation.
- Object Recognition: Identify and interact with health center objects like dispatch units and storage.
- Autonomous Decision-Making: Dynamic path planning and task prioritization.

System Design

Core Components:

1. **OrderController**:
    - Processes logistics orders for medication and supplies.
2. **Robotic Manipulation**:
    - Automated fetching and handling of supplies.
3. **Inventory Management System**:
    - Tracking of number of medicines at different medicine supply units.
4. **Navigation System**:
    - Integrates real-time data from sensors like **LiDAR** and cameras.
    - Uses **MoveBaseAction** under ROS for autonomous mobility.

Key Features:

- Interaction with **medicine supply units and target locations**.
- Task execution in **dynamic and obstacle-filled environments**.

Environment Setup

Tools Used:

- Gazebo for virtual environment simulation.
- ROS for model control and navigation.
- Movement and path planning using move_base package in ROS.

Simulation Setup:

- Dynamic hospital environment with obstacles.
- Use of technologies like LiDAR
- ModelController for handling logistics operations.

Made with Gamma

Defining Models

- Structuring , defining, visualizing models
- Models Used : Medicines, Storage units, Dispatch areas

Manages Models:
Loads models from files into memory.
Spawns models into the Gazebo simulation.
Deletes models from the simulation.

Navigation

Planning for Object Interaction:

- Calculating inventory in specific storage units and deciding if there exists a feasible path that satisfies the requirements of the request

Optimal Path Planning:

- Plan an optimal path to the target object or area using LiDAR and Local and Global path planner maps provided.
- Path Planning: Employs global planners (A*, Dijkstra) and local planners (DWA) for dynamic, obstacle-free movement.

Object Identification and Pose Estimation:

- Utilizes pre-defined object descriptors (e.g., dispatch units, storage units) from the hospital simulation environment.

Navigation Workflow:

- Obstacle Avoidance: Real-time costmap updates for bypassing unexpected obstacles.

Technical Details

- **OrderController:**

The OrderController class manages the flow of orders in a robotic hospital system. It is responsible for creating, queuing, and dispatching orders for the robots.

- **ModelController:**

The ModelController is responsible for loading models from files, spawning them in the simulation environment, and deleting them when no longer needed.

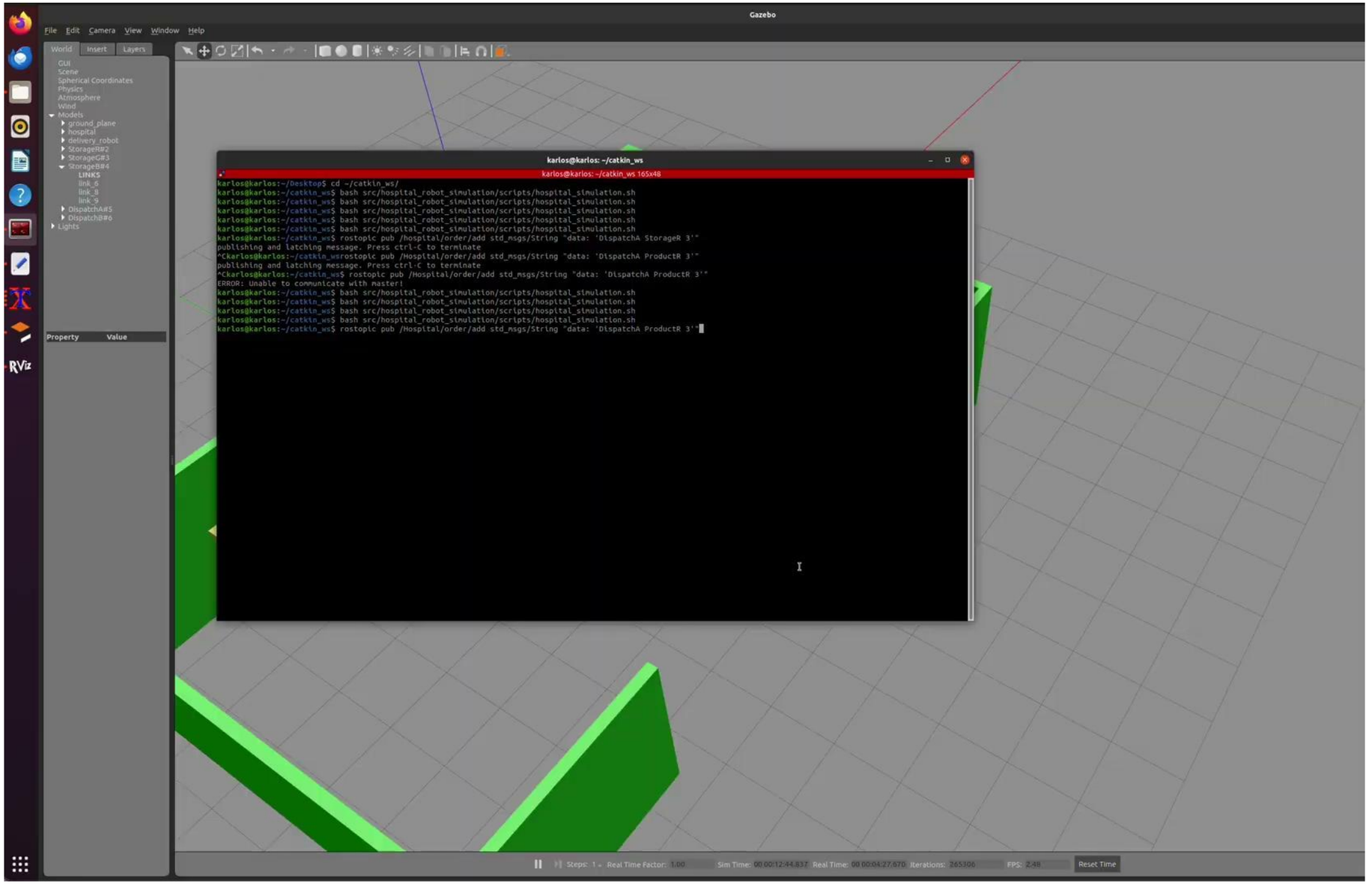
- **HospitalSimulation:**

It includes functionalities for spawning and managing models like storage units, dispatch units, and robots, as well as handling orders and ensuring a clean shutdown process.

- **Robot:**

The Robot class is responsible for managing robot operations in a hospital environment. It interacts with orders, storage units, dispatch units, and movement controllers to automate order processing.

Demo


Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
hospital
delivery_robot
StorageR#2
StorageG#3
StorageB#4
LINK5
link_6
link_8
link_9
DispatchA#5
DispatchB#6
Lights
Property Value
karlos@karlos: ~/catkin_ws
karlos@karlos: ~/catkin_ws 165x48
karlos@karlos:~/Desktop$ cd ~/catkin_ws/
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ rostopic pub /Hospital/order/add std_msgs/String "data: 'DispatchA StorageR 3'"
publishing and latching message. Press ctrl-C to terminate
^Ckarlos@karlos:~/catkin_wsrostopic pub /Hospital/order/add std_msgs/String "data: 'DispatchA ProductR 3'"
publishing and latching message. Press ctrl-C to terminate
^Ckarlos@karlos:~/catkin_ws$ rostopic pub /Hospital/order/add std_msgs/String "data: 'DispatchA ProductR 3'"
ERROR: Unable to communicate with master!
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ bash src/hospital_robot_simulation/scripts/hospital_simulation.sh
karlos@karlos:~/catkin_ws$ rostopic pub /Hospital/order/add std_msgs/String "data: 'DispatchA ProductR 3'"
Steps: 1 Real Time Factor: 1.00 Sim Time: 00 00:12:44.837 Real Time: 00 00:04:27.670 Iterations: 265306 FPS: 2.48
Reset Time

Conclusion

Revolutionizing Hospital Efficiency with Robotics

- Through our innovative Robotics-Based Medication Delivery System, we have taken significant strides in improving hospital operations by automating the delivery of medications. Utilizing ROS-based SLAM, our robot navigates complex hospital corridors efficiently, reducing the dependency on manual labor and minimizing human error.

Key Achievements:

- **Autonomous Navigation:** Success in real-time SLAM implementation allowing precise localization and mapping in dynamic hospital environments.
- **Real Time Inventory Management:** Decide the feasibility of path planning and request completion in real time before spending any effort.
- **Robust System Architecture:** Developed a scalable and modular framework that supports easy updates and integration with existing hospital systems.

THANK YOU