WORLD ROBOTICS OLYMPIAD - ADVANCED ROBOTICS CHALLENGE

A PREPRINT

Mayank Kyal
180414
kyalmynk@iitk.ac.in

Prakhar Katiyar
180525
prakhark@iitk.ac.in

Shivang Shekhar
180724
shivang@iitk.ac.in

June 6, 2019

ABSTRACT

The project demands a robot that could fulfill the purpose of gardening in a greenhouse environment.For this we are required to perform a group of tasks based on which the evaluation will be made.We are going to implement everything as a simulation first.For that we are using gazebo simulator - which provides wide range of environments,sensors and plugins to work with, RViz - which is a 3D visualization tool and will help us interprete the messages received from the husky sensors and ROS framework .Finally We are going to write our programs in the ROS framework.

1 INTRODUCTION

The challenge is to make a robot that can score as many points as possible in a robot version of the autonomous operation of gardening in a green house. We need to design and program robot that can do localization and mapping for the playground that simulate different environment of a green house in order to garden the green house.A greenhouse is a closed structure that provide different regulated climate for plants. The greenhouse will be represented by an open square table. The greenhouse contains plants in the growth stage which are represented by small cubes laying on the top of wooden boxes. The boxes represent different climate which differs in the illumination and warming conditions.In order to fulfill the purpose of gardening there is a game planned or particularly a group of task that the robot has to perform in the arena .The documentation ahead is divided into three sections as Game Description , Simulation and ROS framework.

2 GAME DESCRIPTION

There are 5 boxes of 5 different colors and on top of each box there are 4 small cubes which not from the same color of the box and they are located in the four known direction (right, left, forward and backward). The boxes do not have a fixed position inside the table. The robot must start from a parking zone represented by white box that have open side. the positions of the boxes are randomly defined.The robot's task is to move some plants from climate to another one which is represented in the playground as moving the small cubes from top of the boxes to another one. The robot should identify only three needed cubes on the boxes by the following rules and move them to another box:

1. The first cube is located always on the blue box on that side of the box which is the opposite side to the side which is facing the closest wall.
2. The color of the cube on the blue box identifies the second box. So, the first cube should be moved to this box.
3. The second cube is located on the second box on that side of the box which is the opposite side to the side which is facing the closest wall.
4. The color of the cube on the second box identifies the third box. So, the second cube should be moved to this box.

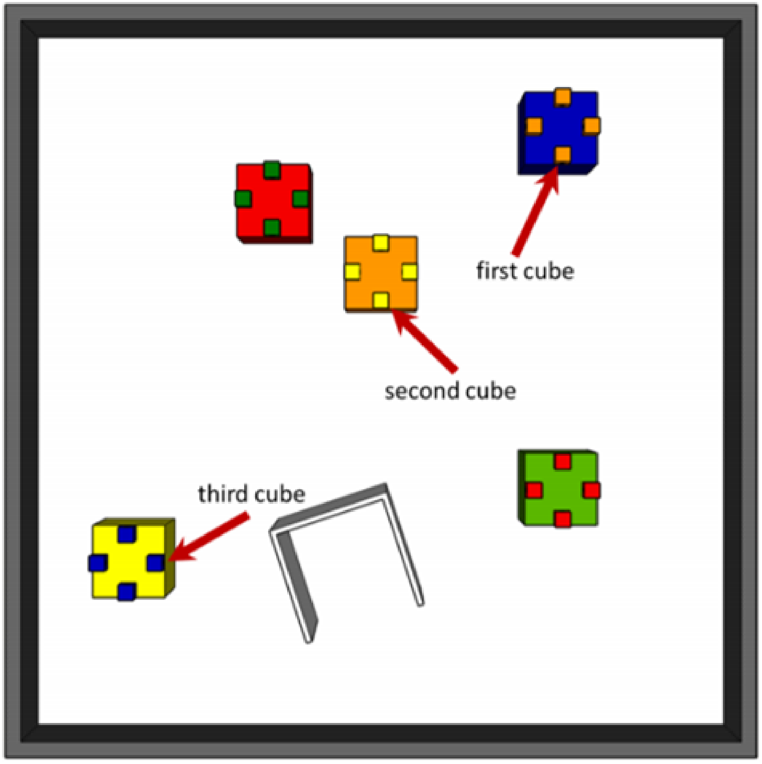

Figure 1: Arena

5. The third cube is located on the third box on that side of the box which is the opposite side to the side which is facing the closest wall.
6. The third cube must be moved to the parking box (could be on top of or within the robot).

3 SIMULATION

Simulation basically means imitation of the operation of a process or system.The act of simulating , first requires a model is developed. This model is a well-defined description of the simulated subject, and represents its key characteristics, such as its behaviour, functions and abstract or physical properties. The model represents the system itself, whereas the simulation represents its operation over time.

A simulator in robotics is used to create an application for a physical robot without depending on the actual machine, thus saving cost and time. In some case, these applications can be transferred onto the physical robot (or rebuilt) without modifications.For our project we are using the renowned Gazebo Simulator and a visualization tool named RViz.

3.1 Gazebo

Gazebo is an open-source 3D robotics simulator.It provides realistic rendering of environments including high-quality lighting, shadows, and textures. It can model sensors that "see" the simulated environment, such as laser range finders, cameras (including wide-angle), Kinect style sensors, etc.It provides a ROS interface to work with and is extensive with a wide range of plugins.It includes a database of many robots and environments(called gazebo worlds).We will build our arena(world) in this simulator and use the inbuilt husky and its different sensors available for the simulation of our task.

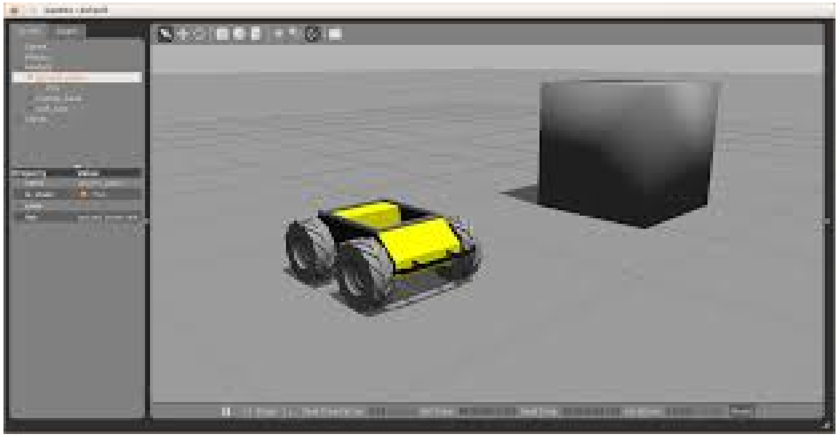

Figure 2: Gazebo

3.2 RViz

RViz (ROS Visualization) is a 3D visualizer for displaying sensor data and state information given by ROS.It offers different camera views like orthographic , top-down , etc .It is extensible with many plugins.We can save our configuration and load it later.It has interactive tools to publish the user information.It gains these informations by subscribing to the ROS topics and visualizing their message contents.

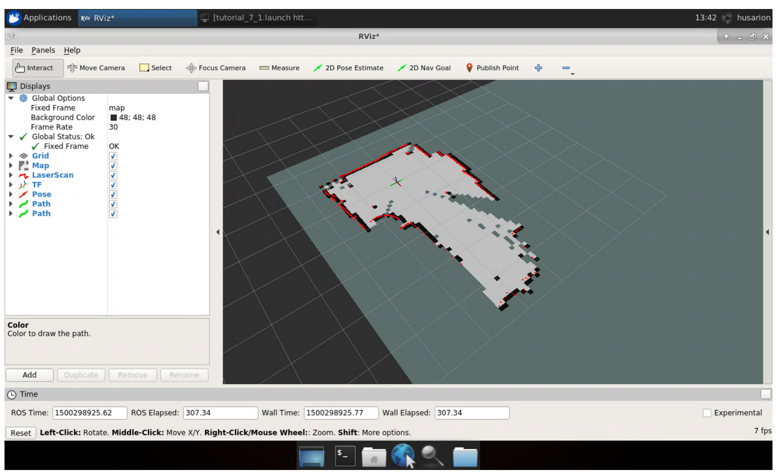

Figure 3: RViz

4 ROS

Robot Operating System or ROS is an open-source, meta-operating system for robots. It provides the services we would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.We are going to use ROS framework for fulfilling our project needs.

4.1 What we have learnt so far

- ROS architecture: Master, nodes, topics, messages, services, parameters and actions
- Console commands: Navigating and analyzing the ROS system and the catkin workspace
- Creating ROS packages: Structure, launch-files, and best practices
- ROS C++ client library (roscpp): Creating your own ROS C++ programs
- Simulating with ROS: Gazebo simulator, robot models (URDF) and simulation environments (SDF)

- Working with visualizations (RViz) and user interface tools (rqt)
- Inside ROS: TF transformation system, time, bags

4.2 What we have implemented

4.2.1 Driving Husky

First we need to launch husky gazebo and then we can publish the desired velocity on the ros topic . The other way is a bit interesting as we need to write a launch file which has an extension of .launch and in which we specify the nodes to be launched . Husky can also be controlled by teleop-twist-keyboard , a ros package . We created a launch file such that on running it , it launches the husky simulator as well as the teleop-twist-keyboard simultaneously and allows controlled movement with the help of defined keys.

4.2.2 Laser Scan

We visualized the laser scan messages recieved by huskyin gazebo simulator to trace the boundary of objects in its path. For this , first we created a package which depended on the predefined packages in ros namely roscpp and sensor-msgs and added these dependencies in the CMakelists.txt and package.xml file.Secondly , we created a subscriber node that subscribed to the laser scan topic of the husky robot and then callback a function which would output the smallest distance at which any obstacle is detected by the laser fitted to the husky.Finally , we created a launch file that will launch the Husky , RViz teleop-twist-keyboard and simultaneously displaying the smallest distance obstacle as the husky moves on the ground. This was possible by inspecting the message type of the laser scan messages published by husky in the laser scan topic and the parameter file which was loaded in the launch file (which specified the topic name and queue size).

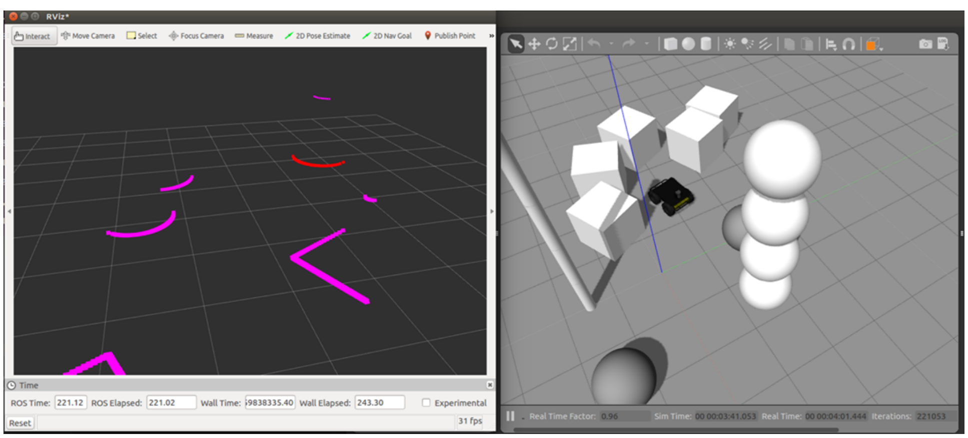

Figure 4: RViz visualization

5 GAZEBO WORLD

Using the feature of build editor in gazebo we designed the arena which resembles the arena for the competition .We are going to use it for our simulation .It looks like as shown in the figure below.

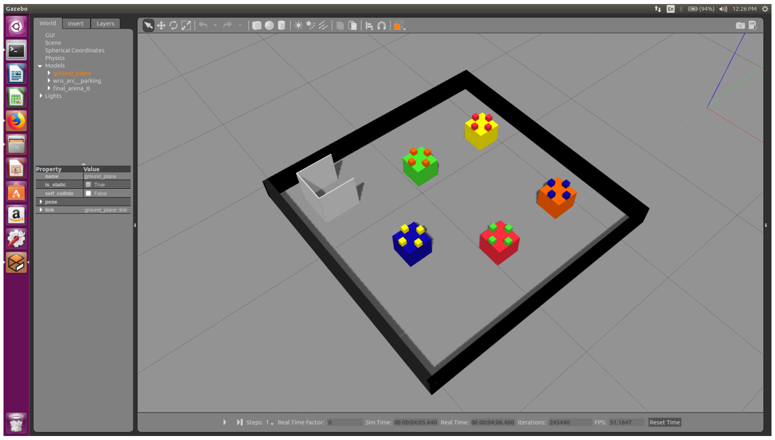

Figure 5: World

References

[1] https://www.ros.org/

[2] ETHzurich Department of Mechanical and Process Engineering Institute of Robotics and Intelligent Systems ://www.rsl.ethz.ch/education-students/lectures/ros.html

[3] Gazebo Tutorials http://gazebosim.org/