Robotic Simulation: Advancing Robotics Research through Virtual Environment

Ruyel T. Rodrigues, Mentor: Dr. Lili Ma
Department of Computer Engineering Technology, CUNY New York City College of Technology, New York, US

Abstract

The study's objective was to simulate a robot in a virtual environment, enabling the observation and modification of its behavior without the need to run the physical robot. This approach bypasses the constraints of studying robotics using an actual robot, such as limited battery power and the absence of a suitable testing environment.

Through simulations, the robot's behavior can be accurately replicated in diverse environments without any inherent limitations. The research findings validate the viability of simulation as a method, demonstrating that the simulated robot mirrors the input and output behaviors of a real robot, yet operates without constraints and is primed for numerous test sessions.

This method offers the advantage of comprehensive testing and modification of the robot's behavior in a controlled and adaptable environment, yielding valuable insights for further development and deployment.

Methods

The research commenced with the installation of Oracle VM Virtualbox and Ubuntu 18.04 Linux OS as a virtual machine, according to the guidance provided by the ROBOTIS e-manual. After the setup process we Installed the necessary ROS OS which for our case was ROS Melodic. Following that, instead of setting up a real robot through SBC setup, we had to download and use the catkin_make command to build a gazebo simulation workspace.

The simulation package came with both TurtleBot3 models, the burger and the waffle, along with a good range of simulated worlds. The ROS package also came with a tele-operation command that allowed the user to use WASD and X keys to control the robot movement in the virtual world. We proceed to build the SLAM map which uses the G-mapping slam method by default. Then we used the teleoperation node to control the robot and make a map of the environment that was saved for later use.

Methods Cont'd

With the saved map we can run a navigation node that uses the map file on another simulated world to understand the robot's relative location within the world through the LDS sensor data. The initial pose estimation helps locate the robot on the map with the LDS sensor that then allows the robot to navigate the map with any given vector command. The Gazebo application comes with basic shapes we can use to add to the map and redo the SLAM operation to test how the robot navigates with the given changes.

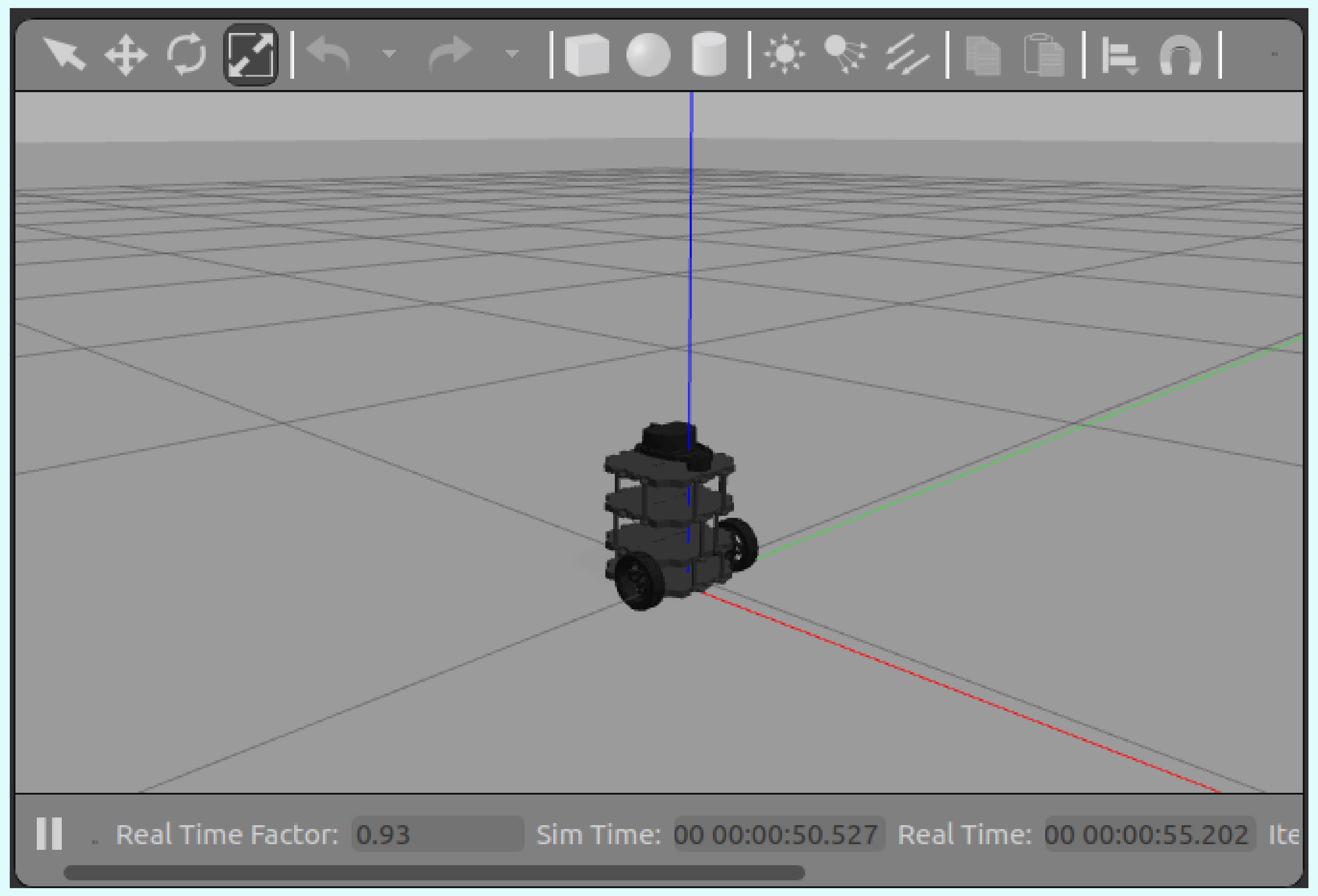

Fig 1. Simulated TurtleBot3_Burger in an empty world. Adapted from "Machine Learning with Turtlebot3." by Ruyel T. Rodrigues

Results

Through this research, I found that the implementation of the Turtlebot3 in a simulated environment is nearly identical to it being deployed in the real-world. That is because the Gazebo simulation package provides the necessary interfaces to simulate a robot in Gazebo using ROS messages, services and dynamic reconfigure. There are some model.sdf files in the folder named turtlebot3_gazebo that describe multiple attributes by categories such as collision, visuals, links, inertia and more.

```
<?xml version="1.0" ?>
<sdf version="1.5">
  <model name="turtlebot3_burger">

    <link name="base">

      <inertial>
        <pose>-0.032 0 0.070 0 0 0</pose>
        <inertia>
          <ixx>0.001</ixx>
          <ixy>0.000</ixy>
          <ixz>0.000</ixz>
          <iyy>0.001</iyy>
          <iyz>0.000</iyz>
          <izz>0.001</izz>
        </inertia>
        <mass>1.0</mass>
      </inertial>

      <collision name="base_collision">
        <pose>-0.032 0 0.070 0 0 0</pose>
        <geometry>
          <box>
            <size>0.140 0.140 0.140</size>
          </box>
        </geometry>
      </collision>

      <visual name="base_visual">
        <pose>-0.032 0 0 0 0 0</pose>
        <geometry>
          <mesh>
            <uri>model://turtlebot3_common_meshes/burger_base.dae</uri>
            <scale>0.001 0.001 0.001</scale>
          </mesh>
        </geometry>
      </visual>

      <collision name="caster_collision">
        <pose>-0.081 0 -0.004 0 0 0</pose>
        <geometry>
```

Fig 2. File showing model information for the simulation robot. Adapted from, "Machine Learning with Turtlebot3." by Ruyel T. Rodrigues

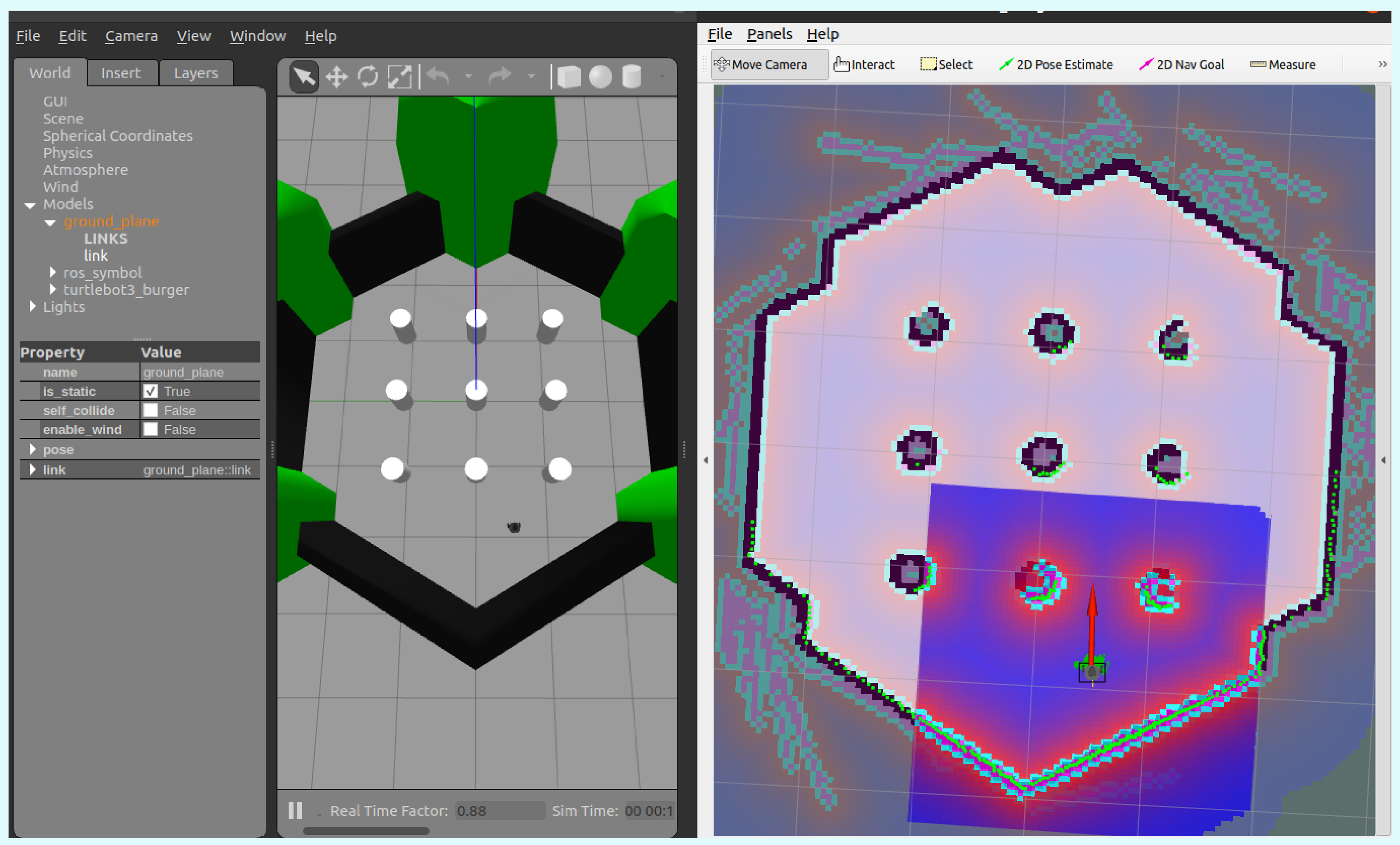

Fig 3. Gazebo simulation of the robot in a scanned world(left), Rviz simulation of the robots mapping and sensor data used for navigation. Adapted from "Machine Learning with Turtlebot3." by Ruyel T. Rodrigues

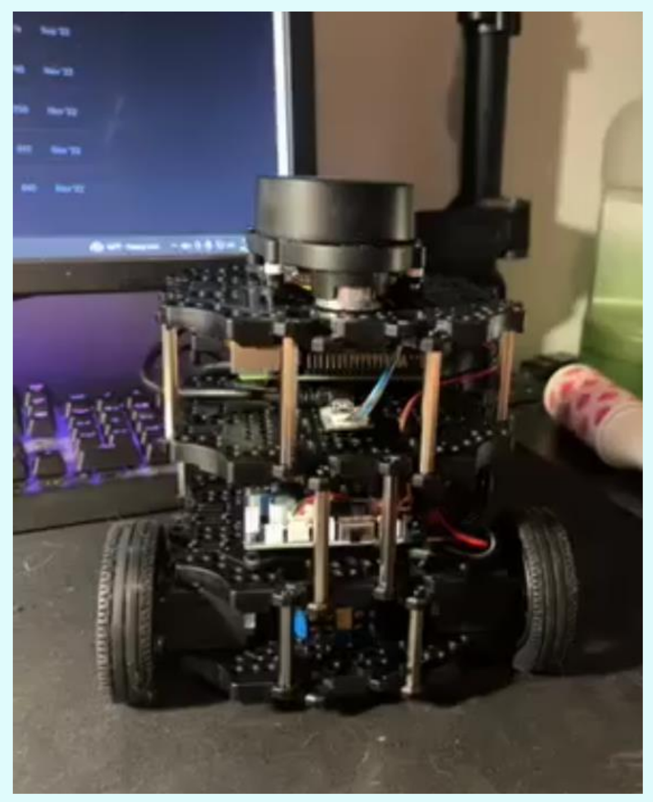

Fig 4. Physical TurtleBot3 Robot demo. Adapted from "Machine Learning with Turtlebot3." image by Christian Rosa

Results Cont'd

They all essentially describe how the simulation of the turtlebot should be initiated with properties like base_visual, lidar, left_wheel etc. all having their position and axis given. We noticed that everything that is being simulated has files describing its attributes including the world, objects, walls etc.

The robot's movement and sensory behavior is identical to the actual robot because the simulation still uses ROS, which is the same ROS used by the real-world counterpart. In fact, all the functions that can be used with a physical burgerbot must be used by the simulation too because the simulated robot is not a new object.

Conclusion

In conclusion, the Gazebo simulation serves as an accurate visual representation of real-world scenarios, enabling the execution of experiments using logic like that employed by physical robots. This research successfully recreated a simulation with all the functionalities of the real-world burgerbot, offering an adaptable environment that mirrors real-world conditions.

The flexibility of simulations allows for convenient execution, pausing, and restarting as needed, facilitating more complex research, particularly in the realm of machine learning. This capability is essential for iterative processes, as it permits the reset of both the robot and its environment, paving the way for extensive and in-depth exploration within the field of machine learning.

References

[2] "Machine Learning with Turtlebot3." Edited by Ruyel T. Rodrigues and Christian Rosa, City Tech OpenLab Site Wide Activity RSS, openlab.citytech.cuny.edu/groups/a-i-integration-with-turtlebot3-temp/. Accessed 30 Nov. 2023.

[4] Son, Will. "Robotis E-Manual." Manual, 27 July 2017, emanual.robotis.com/docs/en/platform/turtlebot3/machine_learning/#set-parameters.