

4 Bringing Up RazBot Simulator

Previous: [3 Creating the RazBot Control Configuration](#)

Next: [5 Import Solidworks Data into RazBot Model](#)

Create `razbot_gazebo` package, used to load the RazBot into the Gazebo simulator. You can find the output for this stage of the tutorials in https://github.com/clearpathrobotics/razbot_tutorials/tree/ready-for-sim

1. Create the skeleton for the package:

```
$ catkin_create_pkg razbot_gazebo roslaunch
$ mkdir razbot_gazebo/urdf razbot_gazebo/launch
```

2. Modify `razbot_gazebo/package.xml`. This file contains dependencies and other relevant information about our package:

```
<?xml version="1.0"?>
<package>
  <name>razbot_gazebo</name>
  <version>0.0.1</version>
  <description>RazBot Simulator bringup</description>

  <maintainer email="pbovbel@clearpathrobotics.com">Paul Bovbel</maintainer>
  <license>BSD</license>

  <buildtool_depend>catkin</buildtool_depend>
  <build_depend>roslaunch</build_depend>
  <run_depend>gazebo_plugins</run_depend>
  <run_depend>gazebo_ros</run_depend>
  <run_depend>gazebo_ros_control</run_depend>
  <run_depend>razbot_control</run_depend>
  <run_depend>razbot_description</run_depend>
  <run_depend>robot_state_publisher</run_depend>
</package>
```

3. Modify `razbot_gazebo/CMakeLists.txt`. This file tells catkin, the ROS build system, how to build our package:

```
cmake_minimum_required(VERSION 2.8.3)
project(razbot_gazebo)

find_package(catkin REQUIRED COMPONENTS roslaunch)
catkin_package()

roslaunch_add_file_check(launch)

install(
  DIRECTORY urdf launch
  DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}
)
```

4. Create `razbot_gazebo/urdf/razbot.gazebo.xacro`

```

<?xml version="1.0"?>
<robot name="razbot" xmlns:xacro="http://ros.org/wiki/xacro">

  <!-- Include the razbot robot description -->
  <xacro:include filename="$(find razbot_description)/urdf/razbot.urdf.xacro" />

  <!-- Gazebo plugin for ROS Control -->
  <gazebo>
    <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
      <robotNamespace></robotNamespace>
    </plugin>
  </gazebo>

  <!-- Additional plugins, like an IMU sensor, can be loaded here -->
  <gazebo>
    <plugin name="imu_controller" filename="libhector_gazebo_ros_imu.so">
      <robotNamespace></robotNamespace>
      <updateRate>50.0</updateRate>
      <bodyName>base_link</bodyName>
      <topicName>imu/data</topicName>
      <accelDrift>0.005 0.005 0.005</accelDrift>
      <accelGaussianNoise>0.005 0.005 0.005</accelGaussianNoise>
      <rateDrift>0.005 0.005 0.005 </rateDrift>
      <rateGaussianNoise>0.005 0.005 0.005 </rateGaussianNoise>
      <headingDrift>0.005</headingDrift>
      <headingGaussianNoise>0.005</headingGaussianNoise>
    </plugin>
  </gazebo>

</robot>

```

5. Create `razbot_gazebo/launch/simulator.launch`:

```

<?xml version="1.0"?>
<launch>
  <!-- Load the RazBot's control configuration -->
  <include file="$(find razbot_control)/launch/control.launch"/>

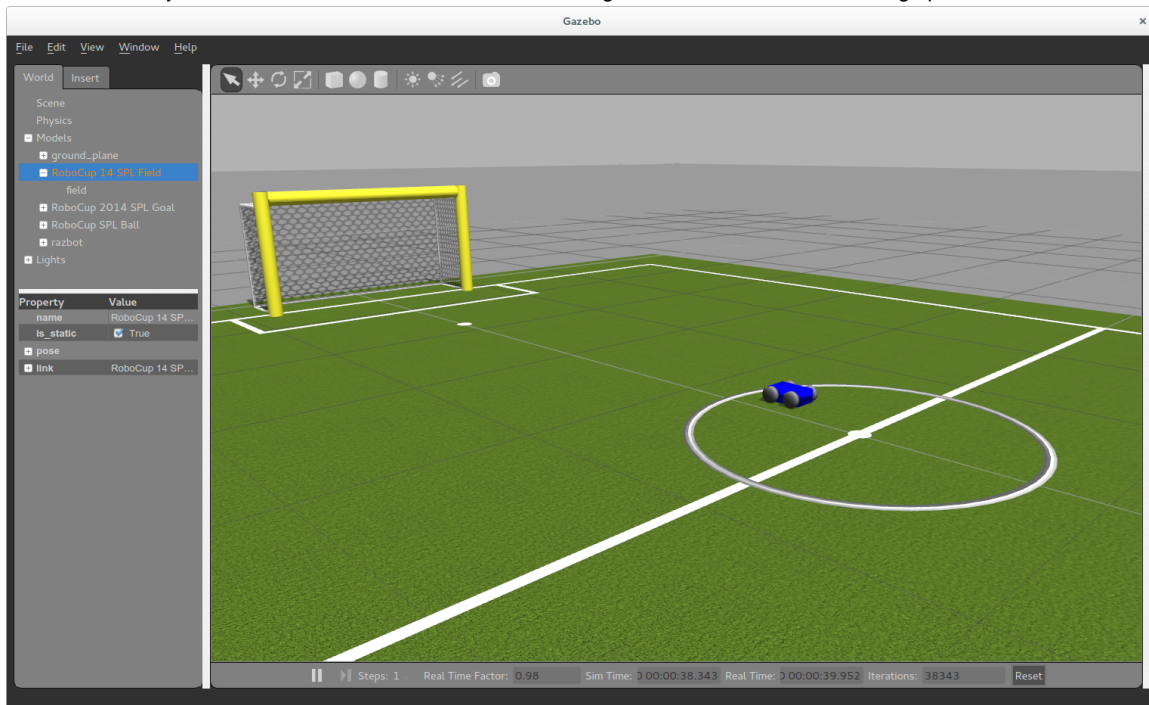
  <!-- Start the Gazebo simulator, loading a world -->
  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="world_name" value="worlds/robocup14_spl_field.world"/> <!--
world_name is wrt GAZEBO_RESOURCE_PATH environment variable -->
  </include>

  <!-- Send the RazBot Gazebo URDF to param server -->
  <param name="robot_description" command="$(find xacro)/xacro.py '$(find
razbot_gazebo)/urdf/razbot.gazebo.xacro' " />

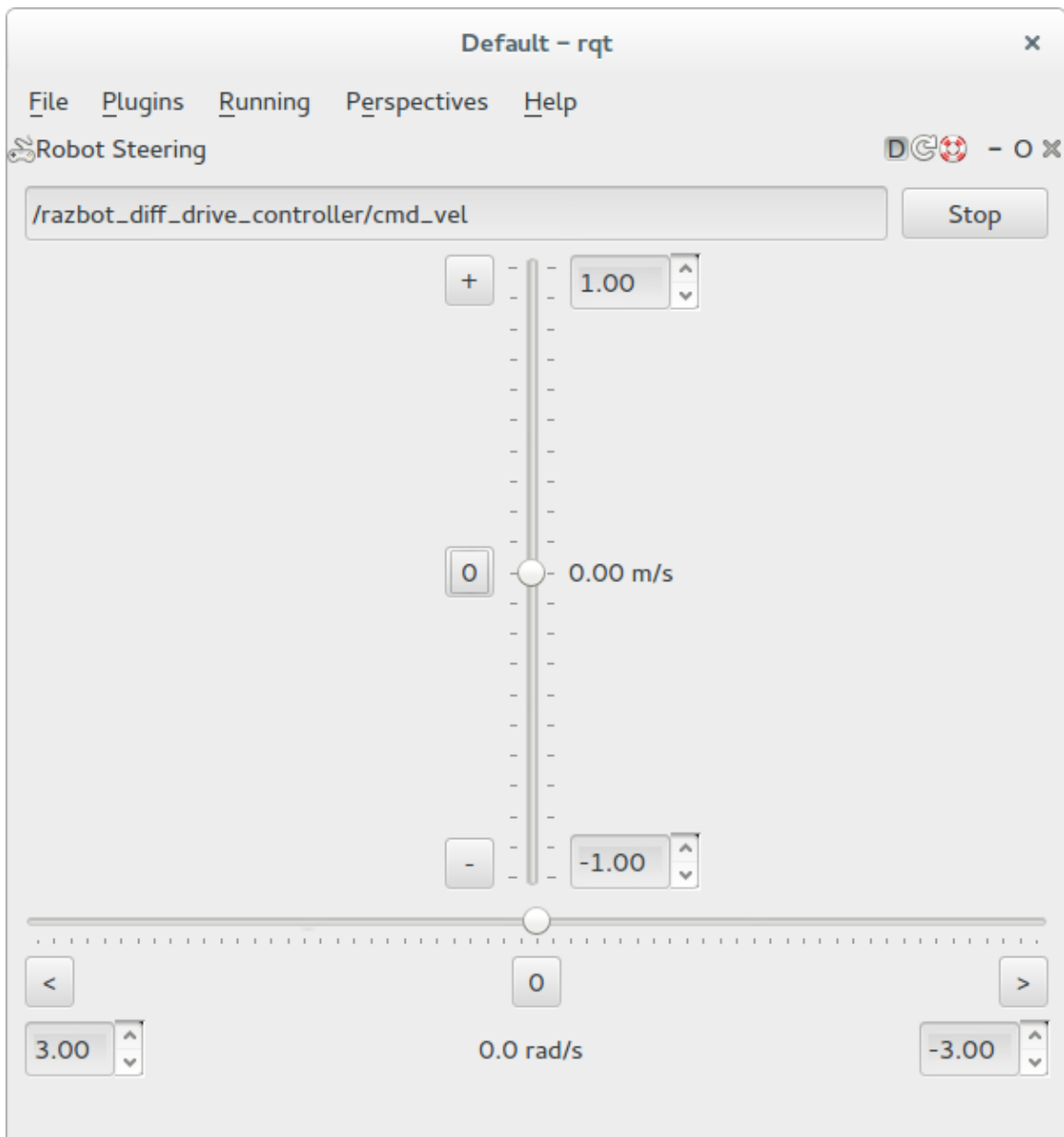
  <!-- Spawn RazBot in gazebo -->
  <node name="spawn_razbot" pkg="gazebo_ros" type="spawn_model" args="-unpause
-urdf -param robot_description -model razbot"/>
  <node name="robot_state_publisher" pkg="robot_state_publisher"
type="robot_state_publisher" />
</launch>

```

6. We are now ready to start the simulator! Run `roslaunch razbot_gazebo simulator.launch` to bring up an instance of the Gazebo simulator.



7. To control the robot, run `roslaunch rqt_robot_steering rqt_robot_steering` in a separate terminal. Set the output topic to `/razbot_diff_drive_controller/cmd_vel` as below, and use the sliders to drive the robot.



Explore some alternative ways to control the robot:

- http://wiki.ros.org/teleop_twist_joy
- http://wiki.ros.org/interactive_marker_twist_server