Gazebo and ROS - part 2

Kjartan Halvorsen

Overview

1. Position control of the Pioneer2dx robot
2. Adding a depth camera to the robot
3. Using the parameter server
4. Ideas for coming ROS seminars

Sources

- http://gazebosim.org/
- http://gazebosim.org/tutorials/?tut=ros_comm
- http://docs.ros.org/kinetic/api/gazebo_msgs/html/index-msg.html
- http://gazebosim.org/tutorials?tut=ros_depth_camera&cat=connect_ros

A simple controller (reusing code)

Download the code

```
~$ roscd pioneer_gazebo/src
~/catkin_ws/src/pioneer_gazebo/src$ wget \
> http://alfkjartan.github.io/files/robot_controller.cpp

~/catkin_ws/src/pioneer_gazebo/src$ cd ..
~/catkin_ws/src/pioneer_gazebo$ wget \
> http://alfkjartan.github.io/files/CMakeLists.txt
```

A simple controller - usage

Build the package etc, then

```
~$ rosrun pioneer_gazebo pioneer_controller
```

Adding a depth camera to the pioneer

1. Make a new gazebo depth camera that publishes its data to ROS topics (add plugin)
2. Attach the depth camera to the pioneer2dx robot

Add ROS plugin to depth camera

1. In gazebo, insert a depth camera from the model database
2. Quit gazebo

Add ROS plugin to depth camera, contd

We now have the definition of the depth camera stored locally. Let's make a copy and modify.

3. Create a copy of the depth camera model

```
~$ cd ~/.gazebo/models
~$ cp -R depth_camera ros_depth_camera
```

4. Change the name of the model in the `model.config` file and the `model.sdf` file.
5. Add the `<plugin>` tag in the `model.sdf` file

Add ROS plugin to depth camera, contd

7. Add the <plugin> tag in the model.sdf file right before thex closing </sensor> tag

```
<plugin name="camera_plugin" filename="libgazebo_ros_openni_kinect.so">
  <baseline>0.2</baseline>
  <alwaysOn>true</alwaysOn>
  <updateRate>0.0</updateRate>
  <cameraName>camera</cameraName>
  <imageTopicName>/camera/depth/image_raw</imageTopicName>
  <cameraInfoTopicName>/camera/depth/camera_info</cameraInfoTopicName>
  <depthImageTopicName>/camera/depth/image_raw</depthImageTopicName>
  <depthImageInfoTopicName>/camera/depth/camera_info</depthImageInfoTopi
  <pointCloudTopicName>/camera/depth/points</pointCloudTopicName>
  <frameName>camera_link</frameName>
  ...
  </plugin>
```

Adding a depth camera to the pioneer

1. Make a new gazebo depth camera that publishes its data to ROS topics (add plugin)
2. Attach the depth camera to the pioneer2dx robot

Attaching the depth camera

Modify the `pioneer.world` file

```
~$ roscd pioneer_gazebo/worlds
~/catkin_ws/src/pioneer_gazebo/worlds$ gedit pioneer.world
```

Add the depth camera to the definition of the pioneer model. Right before ending `</model>` tag:

```
<include>
  <uri>model://ros_depth_camera</uri>
  <pose>0.2 0 0.2 0 0 0</pose>
</include>
<joint name="depth_camera_joint" type="fixed">
  <child>ros_depth_camera::link</child>
  <parent>chassis</parent>
</joint>
```

Attaching the depth camera, contd

Save the world to a new file (e.g. `pioneer-depth-camera.world`), and modify the launch file:

```
~/catkin_ws/src/pioneer_gazebo/worlds$ gedit ../launch/pioneer.launch
```

Change the world to load.

Attaching the depth camera, contd

Change the world to load.

```
<launch>
  <!-- Use the logic in empty_world.launch. Just launch another world -->
  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="world_name" value="$(find pioneer_gazebo)/worlds/pioneer-dept
    <!-- More params can be changed here -->
  </include>
</launch>
```

Data from the depth camera

Close any running gazebo and launch our new world

```
~$ roslaunch pioneer_gazebo pioneer.launch
```

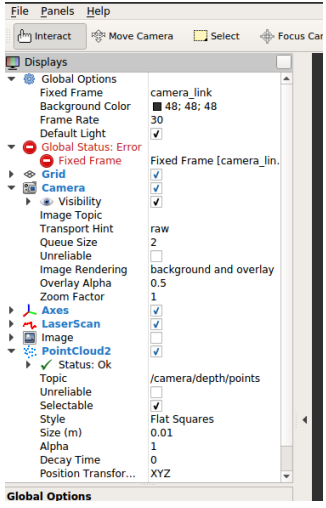
Place some objects around the robot

Visualize data in `rviz`

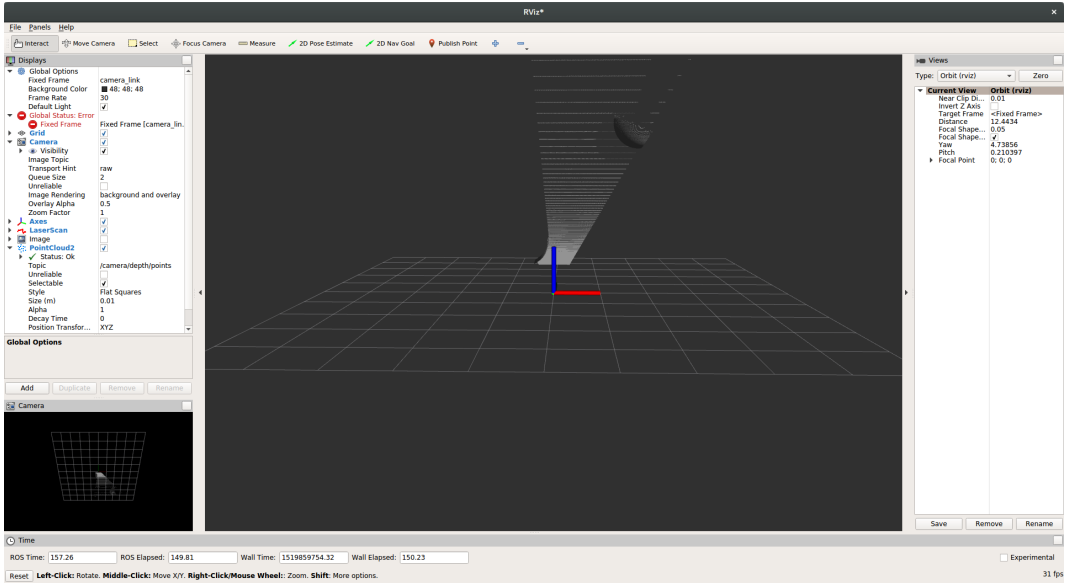
Start `rviz`

```
~$ rosrun rviz rviz
```

Visualize data in `rviz`, contd



Visualize data in `rviz`, contd



Look around

Make the robot rotate and see the changing view in `rviz`

Look around

Make the robot rotate and see the changing view in `rviz`

```
~$ rostopic pub /pioneer2dx/cmd_vel geometry_msgs/Twist \
> '{angular: {z: 0.3}}'
```

The ROS parameter server

Which of the following parameters is not available (hint: `rosparam`)

`/camera/depth/image_raw/compressed/format`
`/camera/imager_rate`
`/gazebo/gravity_x`
`/gazebo/gravity_y`
`/gazebo/gravity_z`
`/gazebo/max_contacts`
`/gazebo/min_contacts`
`/gazebo/time_step`
`/rosdistro`
`/rosversion`

Setting and getting parameters

What is the current setting of the gravity vector?

Setting and getting parameters, contd

```
~$ rosparam get /gazebo/gravity_z
-9.8
```

Setting and getting parameters, contd

```
~$ rosparam set /gazebo/gravity_z -1.62
```

Reading parameters in our own node

```
...
ros::NodeHandle n;
...

double Kv = 0.1;
if (n.getParam("Kv", Kv)) {
    ROS_INFO("Setting velocity gain to %f", Kv);
} else {
    ROS_INFO("Velocity gain parameter not found");
}
...
```

Setting parameters in the launch file

```
<launch>
  <!-- Use the logic in empty_world.launch. Just launch another world -->
  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="world_name" value="$(find pioneer_gazebo)/worlds/pioneer-dept
    <!-- More params can be changed here -->
  </include>
  <param name="tol" type="double" value="0.2" />
  <param name="Kv" type="double" value="0.05" />
  <param name="Ka" type="double" value="0.1" />
</launch>
```

Ideas for future ROS seminars

- Point cloud library
- Navigation
- ROS control package