Advanced Robot Navigation
Homework 1: Simple Reactive Navigation

Sapan Agrawal (ssagrawal@wpi.edu)

02 February 2020

1 Problem Statement

To write a computer program that simulates the motion of a robot in a 2D environment. The robot should be able to move in any direction with a constant velocity, while avoiding obstacles (Random motion).

2 Robot Configuration

The simulation is performed in the Gazebo7 simulator along the ROS Kinetic. Turtlebot2 is used as the robot. It comprises of a Hokuyo Lidar Sensor. The sensor provides the Euclidean distance of the objects along 8 directions (separated by 45 deg) with 1m collision threshold. The sensor has a sensing range of 0.2 - 10 meters. The robot moves at a constant velocity of 0.5 m/s.

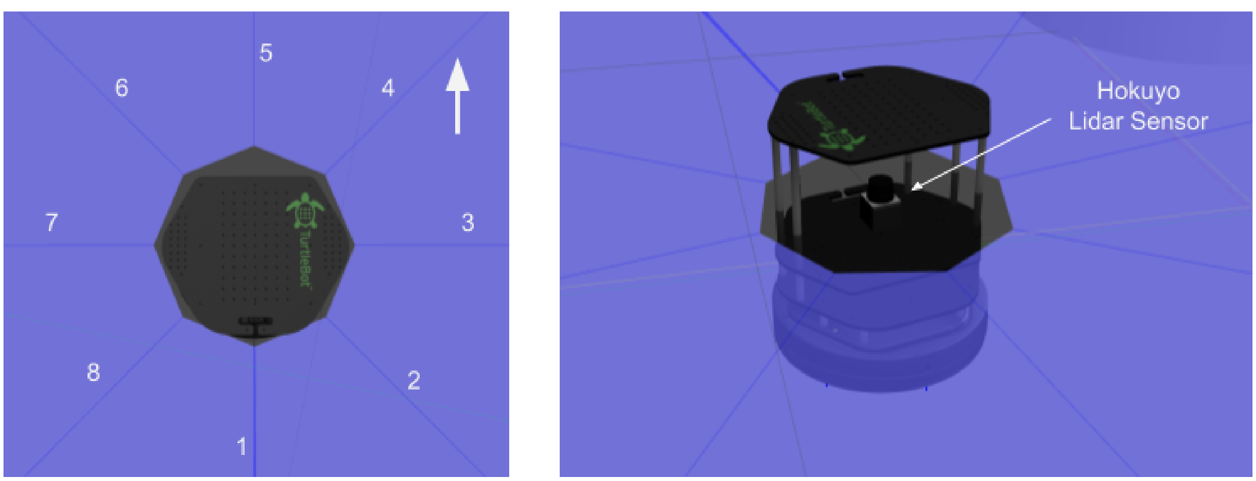

Figure 1: Turtlebot2 with Hokuyo Lidar Sensor in Gazebo Simulator

2.1 State feedback from Hokuyo Sensor

```
Following is the sample feedback from the Hokuyo Sensor:

header:
  seq: 686
  stamp:
    secs: 254
    nsecs:  20000000
  frame_id: "hokuyo_link"
angle_min: -3.1457
angle_max: 3.1457
angle_increment: 0.7864
time_increment: 0.0
scan_time: 0.0
range_min: 0.2
range_max: 10.0
ranges: [1.6132, 2.4566, 1.8732, 2.6634, 2.0183, 3.1427, 3.6013, 4.271]
intensities: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

3 Environment Map

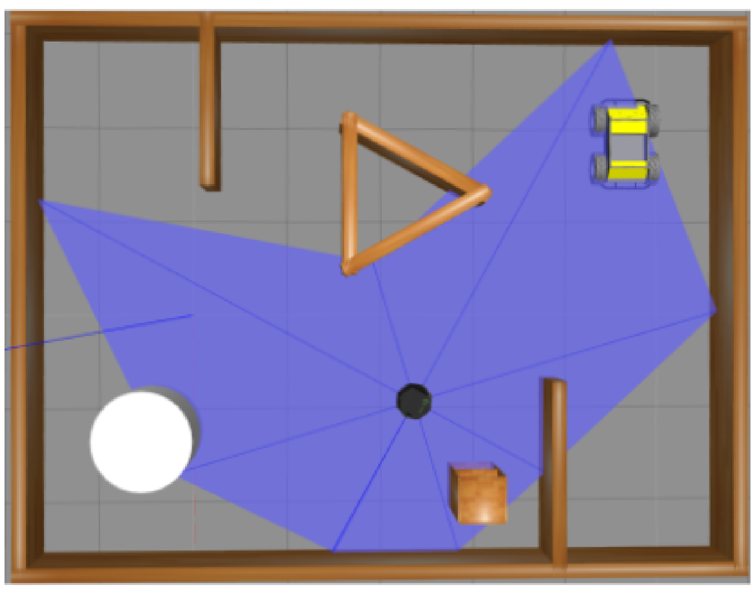

Figure 2: Environment Map built using Gazebo Building Editor

4 Reactive Controller

The controller must be reactive to the obstacles in the environment and try its best to avoid them. I used a simple controller which rotates the robot anticlockwise by 60 deg whenever it detects the obstacle within its threshold at beams {4,5,6} directed outwards at front of the robot as shown in fig[1].

5 Results

Running the simulation with this simple strategy following observation can be drawn:

1. The robot is able to avoid all obstacles at 0.5 m/s
[Video : https://drive.google.com/open?id=1XadEDoS96RbtDqNBqtdqDx2-rrT_XlgT].

2. Increasing the robot speed to 1m/s , the robot collides with the obstacle but recovers quickly
[Video : https://drive.google.com/open?id=10H12tJ9Lzk-qVz-b7kc_3jdk0p6HRWmT].

3. The robot is able to move randomly in all direction while avoiding the obstacles.

4. The robot can move as fast as 0.75 m/s without colliding with any obstacle.

6 Code

The complete code with instructions to run the demo can be found at:
Github: https://github.com/sapan-ostic/Turtlebot_random_walker

7 Simulator tools

1) To stream the position and orientation of the robot in the world, run in new terminal:

```
rostopic echo /gazebo/model_states

position:
  x: -2.08338521141
  y: 4.64392465507
  z: 0.14276554273
orientation:
  x: 4.20383758137e-06
  y: 2.44141824345e-05
  z: -0.271457643144
  w: 0.962450386963
```

2) To stream translation of the robot, run in new terminal:

```
rostopic echo /odom

header:
  seq: 290828
  stamp:
    secs: 3145
    nsecs:          0
  frame_id: "odom"
child_frame_id: "base_footprint"
pose:
  pose:
    position:
      x: -0.753089803332
```

```
      y: 2.27876056889
      z: 0.0
    orientation:
      x: 0.0
      y: 0.0
      z: -0.92043953088
      w: 0.390884983075
  covariance: [0.1, 0.0, 0.0, 0.0, 0.0, 0.0, ... 0.0, 0.05]
twist:
  twist:
    linear:
      x: 5.96787146828e-06
      y: 0.0
      z: 0.0
    angular:
      x: 0.0
      y: 0.0
      z: 0.0421866765713
  covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ... 0.0, 0.0]
```

3) To save the data, run the simulation with following additional argument:

```
roslaunch turtlebot_random_walker walkler.launch runRosbag:=true
```

Check the bags folder for a ROSbag file called walker.bag. This file has a record of all topics from the simulated turtlebot. To access the data from the bag file run:

```
cd ~/catkin_ws/src/turtlebot_walker/bags
rosbag play walker.bag
```