ENPM661 Project 3

Aaradhana Kannan Subramanian

ReadMe

The project is implemented using ROS Kinetic and Turtlebot in Ubuntu 16.04. Listed below are the steps to follow to set up the environment.

Files included:

moveBot.py – The main file to run
ROS_test.py – The python file that is imported into moveBot.py. This contains the Astar Algorithm.
turtlebot.launch – The launch file
Aara_turtlebot_video – The video output for navigation from (0,0) to (4,-3)

Making the workspace:

```
mkdir catkin_ws
cd catkin_ws
mkdir src
cd src
```

Creating the package:

```
catkin_create_pkg planning
```

Setting up the workspace:

```
cd ~/catkin_ws
catkin_make
source devel/setup.bash
```

Create a folder named ‘launch’ and paste the turtlebot.launch file here.
Create a folder named ‘map’ and paste the rrl_map in this folder.
The final folder structure will be
catkin_ws -> src -> planning-> launch-> turtlebot.launch
catkin_ws -> src -> planning-> moveBot.py and ROS_test.py
catkin_ws -> src -> planning->map->rrl_map

The following modules need to be imported to run the files:
In moveBot.py:

```
import rospy
from geometry_msgs.msg  import Twist
import math
import ROS_test
import sys
```

In ROS_test.py:

```
import numpy as np
import cv2
import math
from heapq import *
import sys
```

Commands to run to launch the turtlebot in gazebo at a desired location:

Open the terminal (Ctrl+Alt+T)

Source the workspace using:

```
cd ~/catkin_ws
source devel/setup.bash
ROBOT_INITIAL_POSE:=”-x 0 -y 0” roslaunch planning turtlebot.launch
```

Here, the value after -x is taken as the initial x position and the value after -y is the initial y position.
The coordinate system assumed is similar to gazebo’s coordinate system. It has (0,0) at the center of the map.

Commands to run the python file to navigate in gazebo using A-star:

Open a new terminal after spawning the robot in gazebo.

```
cd ~/catkin_ws
source devel/setup.bash
```

To run the code, the python file must be made an executable. To do that, write the following command.

```
cd catkin_ws/src/planning
chmod +x moveBot.py
```

After this executes successfully, source the workspace using:

```
cd ~/catkin_ws
source devel/setup.bash
```

Now, you can run the file using:

```
rosrun planning moveBot.py
```

The node will prompt you to enter the start x position and start y position again.
It will then prompt you to enter the goal x and goal y position.
The algorithm will then execute and the turtlebot will navigate accordingly.
The threshold defined is 1m. If the turtlebot in within a distance of 1m from the end node, it will stop.
This is done to ensure that the bot doesn’t get stuck in tight spaces or hit obstacles.
The velocities are [10,5] in rad/s.
If the start node or goal node provided are in an obstacle or will cause the bot to hit an obstacle, it will display a message and exit the program automatically.