COMS 4733, Computational Aspects of Robotics, Fall 2019

Lab 2: Bug2 Path Planning. Due (electronically via courseworks): Monday, Oct. 7, 11:59 PM

The purpose of this lab is to implement the Bug 2 algorithm discussed in class. In this lab you will get your turtlebot to move from a start position to a goal position along the m-line. If your turtlebot senses an obstacle, then it invokes a contour following behavior until it reaches the m-line again, at which point it proceeds towards the goal. This behavior (follow m-line, follow contour, re-acquire m-line) continues as long as there are obstacles or until the goal position is reached.

In this lab we will be using the **turtlebot_gazebo package**. You can find this package and its installation in the text Programming Robots with ROS: A Practical Introduction to the Robot Operating System:

(from section 7.1): **sudo apt-get install ros-indigo-turtlebot-gazebo**

Test your installation by running: **roslaunch turtlebot_gazebo turtlebot_world.launch**

We will change the world model from the standard world by specifying our own world with obstacles that the turtlebot needs to traverse:

roslaunch turtlebot_gazebo turtlebot_world.launch world_file:=$PWD/bug2_0.world

NOTE: use an absolute address for the world _file parameter

You can download the bug world files to your own machine. Your Bug needs to successfully complete all the worlds, and extra credit (+10 points) is given for completing bug2_extra.world.

Once you have loaded the bug2_0.world in Gazebo, you can then run your Bug2 program to move from a starting position (which is (0,0,0) – at the origin and heading down the X-axis) to a goal position 10 meters down the X-axis: (10,0,0). You are required to use the nav_msgs/Odometry messages to update your position of the robot (using odometry reference: section 7.8 of the book ROS By Example, A Do-It-Yourself Guide to the Robot Operating System). Your turtlebot has a laser scanner (Kinect-like device) that can be used to sense when obstacles are near. The use of the laser is explained in chapter 7 (Wander-bot) of the book Programming Robots with ROS: A Practical Introduction to the Robot Operating System . A possible strategy you can implement for the Bug algorithm is this (feel free to use other strategies):

1. Follow m-line (go forward) until obstacle encountered.
2. When obstacle encountered:
    1. Store hit point
    2. Turn left until the object is no longer detected to the right of the robot.
    3. Follow the obstacle as below until m-line reached or hit point reached. If hit point reached, conclude impossible.
        1. Move forward a small distance
        2. If object not detected on right, turn slightly right
        3. If object close on right, turn left and move forward
3. Continue along m-line until obstacle encountered or goal reached
4. You may have to experiment with how far you translate and rotate each time before you re-check your laser for an obstacle. Here are some example Bug2 videos.