Assignment 7 ROS

Rafal Grasman

**Manual**

The goal in this assignment is to learn Behavior Based Robotics.

To use the package:

1. Extract the package into your catkin workspace
2. Follow the preparation steps of Assignment7.docx
3. Follow <http://wiki.ros.org/catkin/Tutorials/using_a_workspace> to build the package
4. roslaunch stage\_behavior behavior.launch

Then use in another terminal that also has access to the catkin workspace:

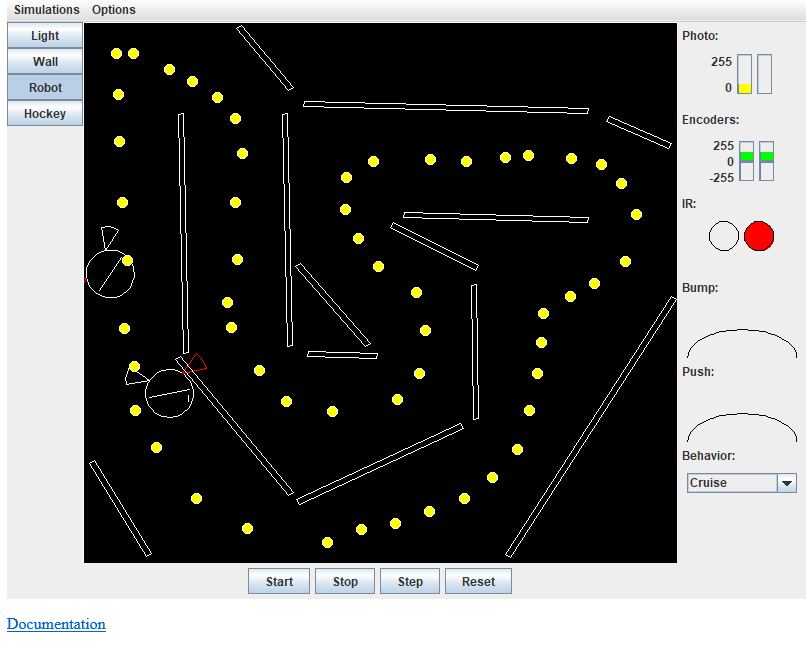
|  |
| --- |
| rostopic pub /move\_base\_simple/goal geometry\_msgs/PoseStamped "header:  seq: 0  stamp:  secs: 0  nsecs: 0  frame\_id: ''  pose:  position:  x: <X position>  y: <Y position>  z: 0.0  orientation:  x: 0.0  y: 0.0  z: 0.0  w: 0.0" |

To set a goal for the robot

**Documentation**

**1**

The basic behavior of a robot could be turning away from a wall as demonstrated in the simulator:



It is also possible to trigger different behaviors based on light and touch.

**2**

The files from mines-ros have been copied and used in assignment 7.

**2.1. Arbiter**

The arbiter has been made so that the current not-yet-published message is always the highest priority:

|  |
| --- |
| void Arbiter::velocityUpdate(const geometry\_msgs::Twist::ConstPtr& cmd, int prio)  {  if (prio < priority)  {  priority = prio;  vel = \*cmd;  }  } |

Then on tick it’s published and reset to lowest prio:

|  |
| --- |
| if (priority != velocities)  {  priority = velocities;  pub.publish(vel);  } |

**2.2. Seek Behavior**

The assignment3 follow\_carrot has been used as seek\_behavior because it’s ready to use code and can follow a line, and has been modified to accept PoseStamped instead of Goal/Path.

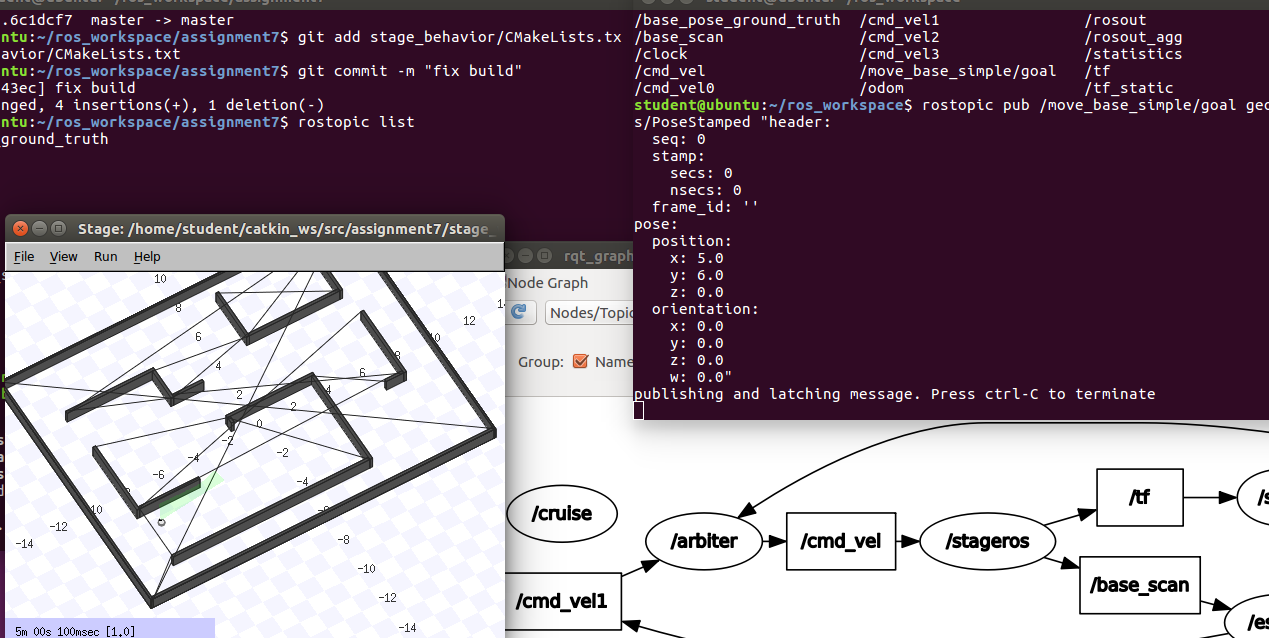
The tf package and seek\_behavior node have been added to stage\_behavior package (+build options) and seek\_behavior has been added to the launch file as priority 2 (higher than cruise, lower than escape).

Topics:

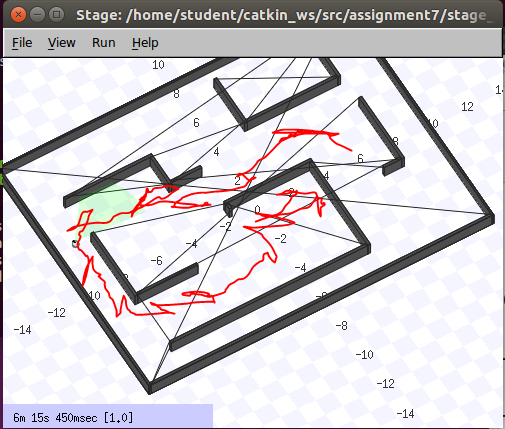
|  |
| --- |
| rostopic list  /base\_pose\_ground\_truth  /base\_scan  /clock  /cmd\_vel  /cmd\_vel0  /cmd\_vel1  /cmd\_vel2  /cmd\_vel3  /move\_base\_simple/goal  /odom  /rosout  /rosout\_agg  /tf  /tf\_static |

The robot moves after publishing a goal.

When publishing the goal 5,6,0 then because of the follow carrot / line PID controller it keeps going to a wall then turning, and repeating, eventually reaching the destination:



The path looks like this:



This is inefficient and could be improved, but the concept works very well by combining multiple behaviors.