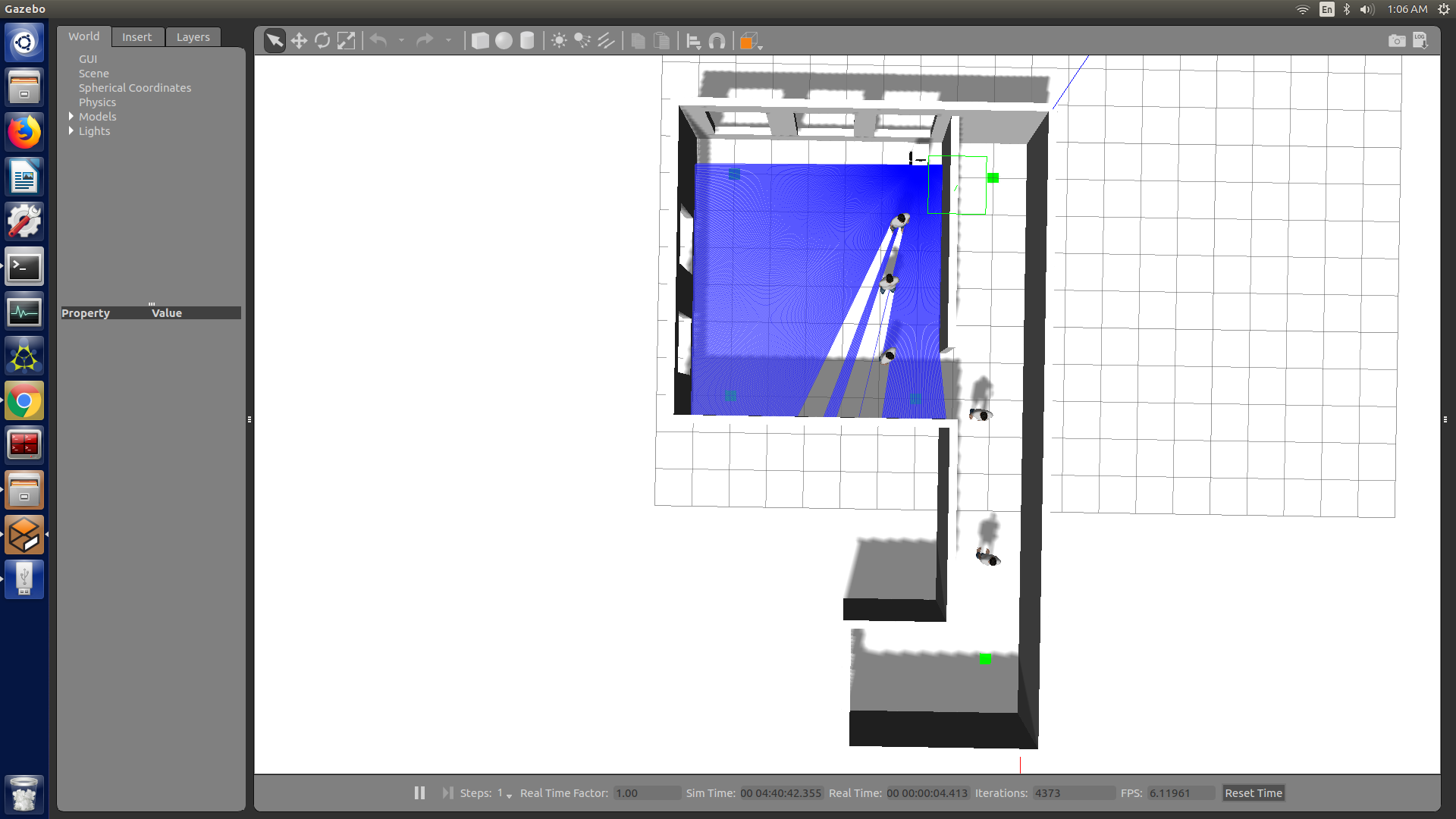
CPE 631 Final Project

Human Robot Interaction Ros Simulation

By Guanghua Zha

5/13/2019

1. **Introduction**

 In this project we are given a gazebo simulation environment with ROS package Pedsim and our goal is to design an algorithm that navigate the robot from start point to the end point. The simulation environment is consisted by several walls and five moving objects which represented the walking human. The robot that we are using is a customized Pioneer3AT wheel robot along with an RGB camera and a 180-degree laser sensor.

To navigate the robot from the start point to the target point, the two main problem is how to navigate and how to avoid people. In this simulation those ‘human’ are moving at a speed of 1.38 originally, to increase the successful rate of the navigation I reduce the speed of these ‘human’ to 1.0. Even with reduced speed they are still moving very fast. If we let the robot running in the open area, it will have a very high chance to encounter people from all directions. However, does not like human beings, the wheel robot has a better mobility in moving forward rather than any other direction, also when turning the robot must turn slowly to avoid roll over problem. When the robot encounter people in the open area it does not have enough time and space to deal with the fast-moving human especially when they meet head to head. On the other hand, in this simulation we are not given any high intelligence tools that can read the environment and make proper reaction with respect to the approaching human. With limited kinetic model of the robot and given tools, I developed a navigation algorithm that let the robot following the walls to reach the target while try to reduce its interaction with human beings.

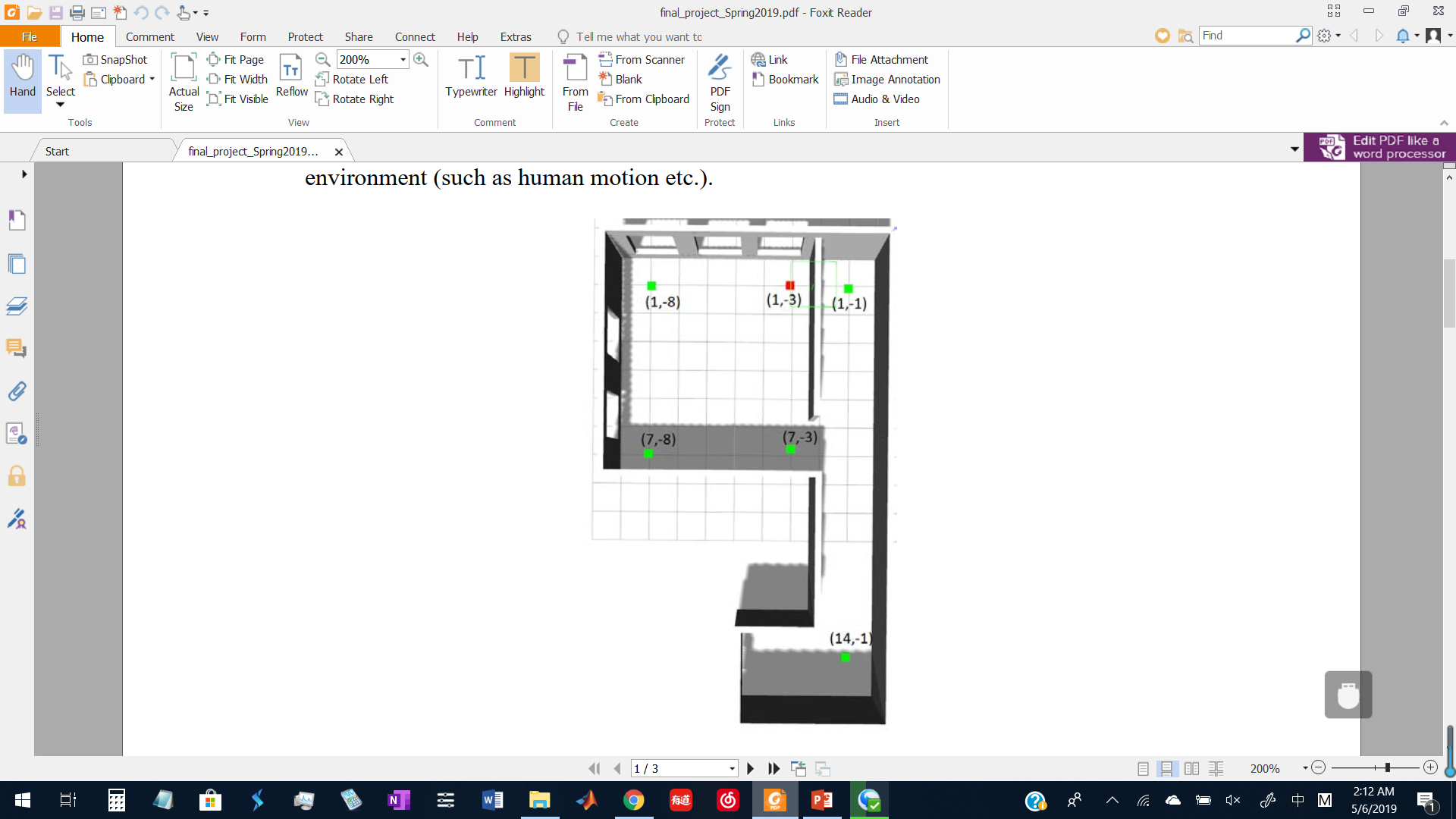
1. **Robot Motion Planning Algorithm**

The basic idea of the navigation algorithm that I developed is to follow the wall in the environment. In this environment, people tend to walk in the space between the wall and therefore by following the wall it is most likely to reduce the interaction with people. On the other hand, the location of the wall can be used to compute the shortest pass using visibility graph method as we learned from previous lecture, and the computation cost is relatively low compare to A\* or D\* since it only contains several points rather than the entire map.

The first step is to acquire the wall location from the gazebo simulator. This can be done using a one-time subscriber to that subscribe to the rostopic /pedsim\_simulator/simulated\_walls. Then the message from the topic can be used to form a list of points that indicated the location of the walls.

The second step is to use a python package pyvisigraph to compute the shortest path from the start point to the end points. The result path for the start point (1, -3) and end pint (15, -1) is [(1, -3) (5.5, -2.5) (15, -1)].

For any two adjacent point (x0, y0) and (x1, y1), if x0 equals x1 or y0 equals y1 that means the path is a straight line. Otherwise it is an oblique line. For an oblique line we formed two different paths as illustrate in the graph below. The green arrow represents the original path and the two right-angle arrows represent the two different paths.



Then we sampled some points within the two different path and choose the one which has more portion that lying on the wall rather than the open space. As shown from the right graph we choose the red pass rather than the blue one.

Then we need to add a small margin to the path to move it away from the wall. The sign(+ or -) of this margin is depend on the current and flowing action.

After this step we have constructed the basic movement sets which contains all the vertical and horizontal movement. All the action so far are only forwarding action.

The third step is to further proceed the action set. First, we defined several actions including 90-degree turn, U turn, S turn and calibration. ’90-degree turn’ is the normal left or right turn. U turn is 180-degree turn. S turn is the transaction to other side of the wall. Calibration is the action to move the robot close to the wall.

S turn

For some horizontal or vertical movement, if it is too short (<=0.5), we can replace such actions with the calibration action as it might cause the robot to hit on the wall since there are insufficient space for the robot to turn.

Here are two possible cases where the vertical or horizontal movement need to be replaced with the calibration action.

Then we insert the turning action in between of the two forwarding. Each turning will then follow by a calibration action to place the robot near the wall or the path that it supposes to follow.

Finally, the navigation stack will be looks like this: Calibration->Forwarding->Turning-> Calibration ->Forwarding->Turning-> Calibration ….

1. **Environment ang GUI**

For this simulation I am using a customized gazebo file that change the box indicator height to zero and relocate the robot to its start position. Also, the median speed of the human simulator is reduced to 1.0.

1. **Implementation Detail**

The actual motion is designed using while loop. Detail of implementation can be found inside the python script.

I also implement a simple collision avoidance system using the laser scanner data. The first step is by acquiring the data that publish on the “/laser/scanner” topic. Then select the data that within the of where the robot head oriented. Count the distance in the selected dataset that are smaller or equal to 0.9. If the count greater than zero, stop the robot. That is indicating something is in front of the robot, most likely to be human beings.

1. **Result and Performance**

I will upload several videos showing the result of the simulation. After lower the speed of human it greatly increase the chance for the robot to reach the destination. The collision avoidance system does prevent some collision between human and robot.

Total length

For my algorithm it took around 35 second in simulation time for the robot to reach the destination. I am not sure how that score compares to the others.

The robot usually has zero to two encounters to the people. With collision avoidance it can handle some of the encounter. However, the robot still gets hit from behind and side. And if the human and robot meet head to head, they cannot avoid each other.

At this point there is no clear definition of how the robot should behave in the human world as most of the robot are employed in the factory and only some small robot such as floor mopping robot that has been more commonly used in our daily life. Without any data collected from human beings, we cannot make any judgment of whether the algorithm that I developed in this project satisfied the human expectation of social manner.

From my personal point of view, I believe that the we should prepare the first stage of human-robot society by developing some algorithm that let robots work in human society without too much disturbance of human beings. After collected enough data of human feedback then we can move on to the second stage where the robot will have more active interaction with human beings. During the process we need to judge and redesign the robot model with advance kinematic capability and higher-level robot intelligence.

1. **Discussions**

This is the first time that I develop a path planning algorithm by myself and I learned a lot from how to write a ROS node to how laser scanner works. This is a very challenging task that many details of the path planning need to be carefully craft. However, further work is needed to complete the algorithm as there are many parts of the code were modified for this simulation environment or were simplified to save time from complex situation.

In conclusion this project allows me to apply what I have learned and make me think about what I can do in the future. Next, I would like to try to solve the problem by combing the machine learning method along with traditional robotics theory. Also, I would like to try out some computer vision method for SLAM and collision avoidance if possible. There is still lot to learn and many new things are coming on the way.