CIS 32.5 Fall 2009, Project 5

1 Description

With this project we finally get to run real localization — that is we have our Player controller figure out from the sensor readings where the robot is. We will use the Roomba again, and the robot is equipped with a laser, since in the current version of Player the only sensor that localization will work with is a laser.

2 Start up

1. Prof. Parsons will give you a folder called `project5` which contains a lot of familiar-looking files.
2. The control program is in the file `real-local.cc`, so first build that:

   `./build real-local`

   then run player on the config file `world5.cfg`:

   `player world5.cfg`
3. The robot, as it has at the start of previous projects, will trundle off to the south-east.
4. However, the display shows some new things.
5. The display is now dominated by a series of these:

```
X: 4.05511   Y: 4.68217   A: -0.116743   W: 0.005
X: -2.45264  Y: 2.15322   A: 0.076456    W: 0.0035
X: 2.55298   Y: 0.762078  A: -0.0410866  W: 0.0031
X: -5.30263  Y: 4.97457   A: -0.0395662  W: 0.0029
X: 6.02264   Y: 0.729982  A: 0.00262671  W: 0.0024
X: 4.01215   Y: -3.67916  A: -0.0180902  W: 0.0022
X: -5.04173  Y: 0.98516   A: -0.225971   W: 0.0019
X: -0.675089 Y: 5.43747   A: -0.0741516  W: 0.0018
```

6. Each of these is a *hypothesis* about where the robot is, with its x coordinate, y coordinate, angle and *weight*.
7. The weight is an estimate of how likely the robot is to be at that location.

3 Its all about configuration

1. Take a look at the configuration file `world5.cfg`.
2. The difference between this configuration file and the others we used before is in the last element.
3. The AMCL driver is a version of the particle filter that we talked about in the lecture on localization.
4. All those possible positions you see on the screen are (more or less) the particles.

4 Know where you are

1. You should notice the number of hypotheses reduces as the robot runs.
2. Your first task is to modify the controller so that the robot localizes properly.
3. That means so that the robot ends up with just one or two hypotheses.

4. If there is one, then clearly localization has figured out where it thinks the robot is.
5. If there are two, then as long as one has a probability of very close to 1, localization thinks it has figured out where the robot is.
6. Your job is to get the root so that localization not only has one or two hypotheses, but that at least one of them is reasonable. As in it is close to where the robot really is (which you can see from the simulator).
7. Don't expect localization to be correct, but it should be approximately correct.
8. You will probably want to make the robot moves in different ways to do this — moving robots localize quicker than stationary ones.

5 Know that you know where you are

1. When localization is generating just a couple of hypotheses, it has figured out where it thinks the robot is.
2. But does the robot control program, the bit in `main`, know that the position reported by the localization module is the right position?
3. Modify the controller so that when localization is down to a few hypotheses, the controller stops the robot moving.
4. Have the controller print out a message from `main` that says:

   `I am at:`

   and then the location and then:

   `I am N percent sure of my position`

   where `N` is the weight of the most likely hypothesis.
5. To do this you will have to modify the function `readPosition` since right now this just returns the location in the last hypothesis returned from localization (which is typically not the most likely one).
6. Once your controller stops the robot when the robot is localized, save your working program as **proj5-part1.cc**.

6 Now use this information

1. Modify your controller so that once the robot has figured out where it is, it then navigates to another position, using localization to keep track of its location.
2. Does it always know where it is? If not, you may have to modify your use of the localization information.
3. Once your robot can navigate across the simulator, save your working program as **proj5-part2.cc**.

7 Explore strange new worlds

1. If you have time, test to see how your robot controller localizes in different worlds.
2. To do this, edit the file `world2.world`.
3. If you look at this file in an editor, you'll see the instructions:

```
map
(
  bitmap "bitmaps/cave.png"
  size [16 16]
  name "cave"
)
```

This tells Player which map to load.

4. Look in the bitmaps folder using:

   `ls bitmaps`

   and try some other maps.

5. Do you have to modify your controller to get the robot to localize?