CS1114 Section 2: Making Friends with Robots

January 30th, 2013

1 Live Demo

2 Robot arena

We’ve set up a virtual robot arena to test your code in case all of the physical robots are in use. For today, you’ll use the virtual robot arena to get familiar with the Matlab commands that allow you to control and program the robots.

1. First, you’ll need to start the virtual arena. Open a Terminal, then change to the arena directory and start the program using the following commands:

   $ cd /courses/cs1114/section/robot_arena
   $ ./run.sh

   Note that the dollar sign ($) is used to indicate a terminal prompt, while a pair of right angle-brackets (>>) will be used to indicate the Matlab command prompt.

   This will open up a window containing an default robot arena. The arena contains a robot (a circular disk with a black dot indicating which direction it’s facing) and a glow stick.

2. Next, open up Matlab and use the robotInit function to create a virtual robot and connect to it. For the robot arena, this function takes three arguments. In Matlab, type:

   >> r = robotInit(’localhost’, ’virtual’, 8000);

   This creates a connection to the virtual robot and stores it in the variable r.

3. Now you can issue the virtual robot any of the robot commands. Any robot you use in this course, be it the Sony Aibo, Wowwee Rovio, or the virtual robots, will understand the following commands:

   robotInit
   robotDriveStraight
   robotTurn
   robotGetFrame
To learn the details of what these functions do and how to call them, you can use the `help` command. For example:

```
>> help robotInit
```

Try initializing your robot and issuing each of the commands. You should see the virtual robot in the arena react to the commands.

4. Now that you’ve gotten a handle on the commands, use them to get a picture of the left and right side of the wall in the virtual arena. Store these images in variables `left` and `right`.

5. Sending these commands one at a time to the robot is fun, but it can get tedious. Suppose you’re programming your robot to explore unknown environments, such as nuclear meltdown sites or Mars. Something you might want to do frequently is take stock of the robot’s surroundings by viewing pictures in front, behind, and to the sides of the robot’s current location. Write a function that does the following for each direction in 90-degree increments:

   - Take an image
   - Display it using `imshow`
   - Pause for two seconds (use the `pause` command) to give you time to look at the image.

Your function should leave the robot in the same orientation that it started, and should have the following signature:

```
function robotShowSurroundings(robot)
```

Write your program using a `for` loop—this will save you a little bit of typing. If you were instead rotating in 10-degree increments, it would save you a lot of typing!

6. Finally, modify your function to take a second and third parameter to specify the number of turns you want the robot to make, and the number of seconds to pause between photos. This new signature would look like:

```
function robotShowSurroundings(robot, numberOfTurns, pauseTime)
```

so that calling

```
robotShowSurroundings(myRobot, 36, 1)
```

would turn the robot in 10-degree increments and display a sequence of 36 pictures, pausing for one second between images.

7. Any remaining time may be used to work on Assignment 1.