91.451, Robotics II
Spring 2005
Prof. Yanco

Sample Midterm Exam Questions

Here's a set of problems that are similar to types of questions for the midterm. Note that this is most likely shorter than the midterm will be (more short answer questions, possibly a "what will this code do" question).

Problem 1: Name and briefly describe (using no more than three sentences and/or a diagram) a cooperative method for behavior combination.

Problem 2: Write direct control code for a robot to search for open space in front of the robot and move towards it. The robot should turn towards the area with the most open space. You do not need to worry about obstacle avoidance or backing up; if all of the front sensors have readings closer than 1.0 robot units, just have the robot stop moving.

Problem 3: Write fuzzy rules in Pyro to equalize the distance of the robot from two obstacles: one in front of the robot and one in back of the robot. If the obstacle in front is closer than the back, the robot should back up. If the obstacle in back is closer than the front, the robot should move forward. If the obstacles are at equal distances from the front and back of the robot, the robot should stop.

You do not need to write a full program. Instead, write the rules needed to control the robot and any supporting statements that are needed in those rules (for example, if you use a variable name in your fuzzy rule, you need to have a line of code that gives a value to that variable).

Problem 4: In this problem, you will write a Pyro program to play hide and seek. In this game, the robot will hide. Once the robot is found, it should wait, then try to find its opponent. Once the opponent is found, the roles reverse again and the robot hides.

Assume that the following behaviors exist:

    HideBehavior
    FindBehavior
    WaitBehavior

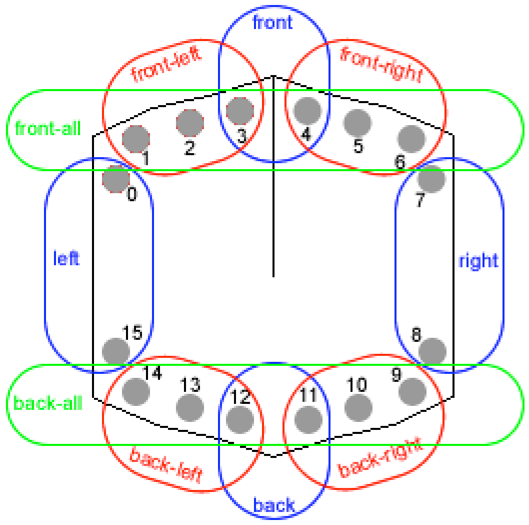
None of these behaviors take any parameters.

You can also assume that global variables called opponentFound and selfFound exist and are updated automatically for you.

A. Draw the state diagram for your robot's program.

B. Write the Pyro code to implement the state diagram you drew in part A. You do not need to write the behaviors given above, nor do you need to write code to compute values of opponentFound or selfFound.

For reference: The mapping of sonars on the Pioneer.



front
front-left
front-right
front-all
left
right
back-all
back-left
back-right
back
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15