

# CS 493/790(X) – Advanced Topics in Robotics

Instructor: Monica Nicolescu

## Lab 4 – Handout

### 1. Introduction

In this lab you will use the behaviors you implemented in more complex controllers and have them compete against each other. It's play time!!

**!!! Important:** At the beginning of the lab, each student should take a few minutes to describe his/her own behavior, and demonstrate how it is used. Some behaviors require specific IDs for the fiducial markers. Please make sure that everybody knows what IDs your behavior is using. After this summary, everybody can start working on their own controllers, using the code provided for lab 3.

Below is the description of all the behaviors that were implemented in the previous lab, along with their descriptions.

- **Circular Patrol:** circular field around a fiducial marker. The field intensity reduces with distance. Field can be clockwise or counter clockwise.
- **Symmetric Circular Patrol:** similar with Circular patrol, but the field is symmetric with respect to a given direction
- **Wall Follow (left/right):** For a given side of the robot (left/right), the field is a sum of vectors parallel with the surface detected by the laser
- **Hide:** given 2 (possibly mobile) fiducial markers (robots), stay on the line formed by 1 and 2, such that 1 is between your robot and fiducial 2
- **Interfere:** given 2 (possibly mobile) fiducial markers (robots), stay on the line formed by 1 and 2, such that your robot is between 1 and 2.
- **Get into line of sight:** given 2 (possibly mobile) fiducial markers (robots), move away from the line formed by 1 and 2, such that 2 is visible from your robot.
- **Dodge:** given a fiducial marker, stay away from a path of some given direction that starts from the marker.
- **Intercept:** this is the opposite of dodge. Given a fiducial marker, move on a path of some given direction that starts at the marker.
- **Wall Avoid:** for any laser reading that reaches an obstacle, create a field of vectors perpendicular to the surface at each of those points.

### 2. Download

The source files for this lab's controller can be downloaded from:

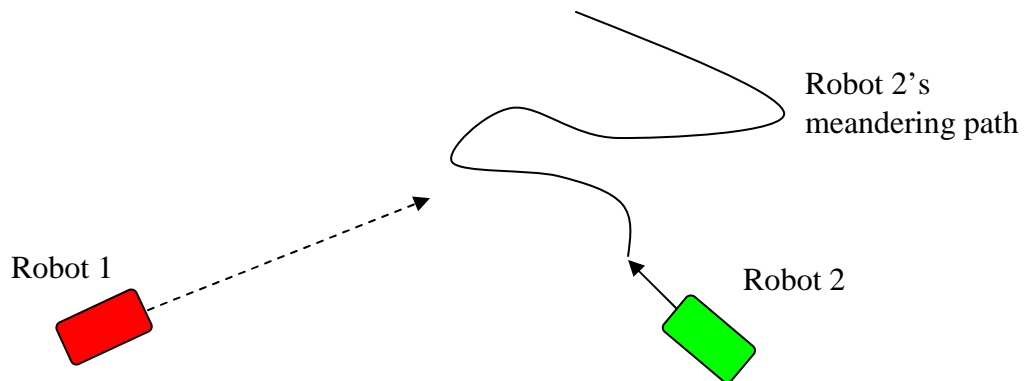
<http://www.cse.unr.edu/~monica/Courses/CS493-790/Labs/Lab3>

These files include all the behaviors currently developed. Please check the IDs of the behaviors in `behaviors.h`, such that you know how to properly set their weights. Since for this lab you will have to run controllers for more than one robot at a time, you should update the code to take the weights file name as an input parameter.

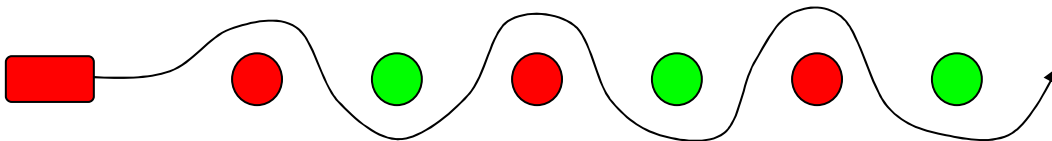
### 3. List of controllers

Below is the list of games that you will play with your developed behaviors. For each scenario, I provided a list of behaviors that your robots should use. This list is not fixed, but rather the guideline of what would be needed to accomplish the task. You may want to add/remove avoidance or wandering behaviors, which is possible if you believe that it would help the robot's performance.

**Playing Catch:** Create an environment with two robots, one running *Laser Avoid* and the *Intercept* behavior, the other running *Laser Avoid*, *Dodge* and *Wander* (figure below). Intercept should have as fiducial target the second robot, while Dodge should have as fiducial target the first robot. Run this scenario at least 5 times and report how well the robots are doing at intercepting each other (for robot 1) or staying away from the other robot's path (for robot 2).

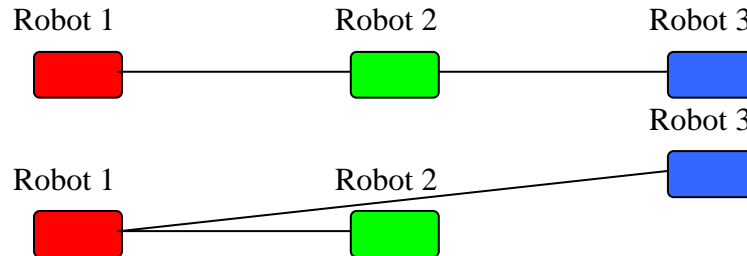


**Olympic Games Skier Robot:** Create an environment in which you have 6 fiducial markers, of two different colors (figure below). Place those markers in a straight line, alternating their colors, at a distance sufficient for the robot to go through. Modify the *Symmetric Circular Patrol* behavior, such that it turns clockwise around targets of color 1 and counterclockwise around targets of color 2. This should be done simply by adding a parameter to that behavior. Using the *Symmetric Circular Patrol* behavior, you should obtain a robot that slaloms through the targets like the Olympic Ski Champions! Report on the behavior of your skier robot and the weights of the behaviors that you used.



**Hide and Seek:** Create an environment with three robots. In this environment we will test the *Hide* and *Line of Sight* behaviors against each other. The first robot is using the *Hide* and *Laser Avoid* behaviors. The second robot is running *Wander* and *Laser Avoid* and the third robot is running *Line of Sight* and *Laser Avoid*. In this experiment, the hiding robot should attempt to hide behind robot 2, such that robot 3 cannot see it. At the

same time, robot 3 is trying to find robot 1, by getting a line of sight to it. First, start the robots in two situations: one in which robot 1 is in advantage (top figure below), in which 1 is hidden behind 2, and one in which robot 3 is in advantage, because it has a line of sight to 1. Report how well the first robot is doing in hiding from 3 and how well is robot 3 doing in seeking robot 1.



**Defend:** Create an environment with three robots. In this world we will test the *Intercept* and *Line of Sight* behaviors against each other. The first robot is using the *Wander* and *Laser Avoid* behaviors. The second robot is using *Intercept* and *Laser Avoid*, and the third robot is using *Line of Sight* and *Laser Avoid*. In this experiment, the second robot should attempt to get between robots 1 and 3, while robot 3 should strive to keep a line of sight to 1. As before, start the robots in two different configurations: one in which robot 2 is in advantage (is already between 1 and 3) and one in which robot 3 is in advantage (because it has a line of sight to 1). These configurations are identical to those in the figure above.

**Traverse a Maze:** Use the environment provided with the file `everything.cfg`, and `everything.world` to test the navigation behaviors. Using the existing behaviors (*Wander*, *LaserAvoid*, *WallFollow*, *WallAvoid*), find a combination of behaviors (and their weights) that would enable the robot to navigate in the office-like environment (e.g., go straight in corridors, make turns at a corner, etc.). Preferably the robot should not get stuck. Report on what behavior combination and behavior weights you have found to be working.

You should report the results on these controllers at the end of the lab next week (March 9).