

Lab 3: Robot Control Methods: Sequencing and Subsumption

Reminder: Mark Micire will talk about robots at the WTC on 9/25, 3:00-4:00, Olsen 311.

Out: Thursday, 18 September 2003

Due: Thursday, 25 September 2003, at start of class

This is an individual lab. Unless expressly stated otherwise, all labs are to be completed individually.

Overview: In this lab, we'll explore two new robot control methods: sequencing and subsumption. Sequencing allows us to create finite state machines to control our robot behaviors. Subsumption will allow us to create a behavior-based robot in which higher level behaviors will take precedence over lower ones. (Next week in lab, we'll continue to explore behavior-based control using fuzzy logic to combine behaviors.)

Part I: Sequencing Behaviors

Read the Pyro Sequencing Control module (linked on the course web page). This section describes a two state brain that drives in a square. The *turn* state will make a 90 degree turn, sending control to the edge state when it finishes. The *edge* state drives forward for one robot length, then sends control back to the turn state. Copy the code from the module and run the square code in simulation. (You may also try it out on the robot if you'd like.)

Modify the code to avoid obstacles. For this, you'll need to write at least one avoid state (you may find that it makes more sense to have two avoid states – one that is called from the turn state and another that is called from the edge state). Edge and turn should move into an avoid state if an obstacle is encountered before it has completed. After the robot has cleared the obstacle in the avoid state, it should move back to the edge or turn state.

Part II: Subsumption in Pyro

Read the beginning of the Pyro Module on Behavior Based Control (linked on the course web page). For this lab, we'll just be looking at the section on vertical behaviors. Try running the Subsumption.py in simulation (and on the robot if you'd like).

Instead of Exercise 1 as described in the Pyro module, extend Subsumption.py to include wall following. Think carefully about how to order the Wander, Avoid, and WallFollow behaviors. Which should have the highest priority? Which should have the lowest?

What to Turn In

For both part of the lab, turn in your (well commented) code. Also, you should demonstrate both programs for me on a robot before the start of class next week.