CS148
Building Intelligent Robots

Week 4 : Subsumption Architecture *Out: 18 Feb 2003*

Preliminary Tasks *before 20 Feb 2003, 9am*

The purpose of this week's lab and project is to introduce the idea of programming in brickOS using subsumption architecture. The following reading will be essential to understanding the lab, and should be completed before coming to lab on Thursday:

- Pages 206-223 *Martin*

For this assignment you should build a generic tankbot with a touch sensor and a light sensor. You can use your light sensor from project 3 if you want.

Lab 4: Subsumption Architecture *20 Feb 2003*

You will be given three modules and your task will be to write an arbitration module between the three. The first module, *wander*, is responsible for having the robot wander randomly in its environment. The second module, *avoid*, will monitor the robot's area and if it's in a "bad spot", will perform an evasive action. A bad spot is any spot where a light sensor detects a bright light. The third module, *halt*, will halt the robot if the touch sensor is pressed. The *arbitrator* module should control the outputs so that the robot moves appropriately while avoiding bad spots.

You will be given standard brickOS code that performs these functions. You will also be given a file which has the wander module, to which you should add an avoid, halt, and arbitrator module. This module should abstract out the motor control for the robot and should subsume the modules in the following order:

- Highest priority: *halt*
- Medium priority: *avoid*
- Lowest priority: *wander*

You can copy the stencil code by typing this in your working directory (from a cygwin shell)

```
cp -r l:/asgn/subsumption/* .
```

Task: Finish the reading if you have not done it!

Task: Write the avoid, halt, and arbitration modules so that your robot will perform the behaviors in a prioritized manner as described above. Make sure your LCD will output the name of the process that currently has the highest priority. Note: The avoid module does not have to be smart. All it has to do is turn away from the light.

Project 4: Subsumption Architecture *due 27 Feb 2003, in class*

Specification: In this project, you will be creating a robot that will have multiple behaviors. You must implement these behaviors using subsumption. These behaviors are:

1. *Wander*: When your robot has nothing better to do, your creature may wander, dance, or just generally move about in any fashion you feel fit.
2. *Obstacle-Avoidance*: When your robot runs into something, it should try to avoid it.
3. *Light Seeker*: When your robot detects a strong light source in its enviornment, it should move towards it. A flashlight shinning on the floor a short distance away from the robot will create the light source.

Some notes to consider:

- Which of the three behaviors should have the highest priority? Which behavior should have the lowest priority? Note that your robot should never get stuck anywhere.
- You have already written an obstacle-avoidance robot, how can you integrate that code with the other behaviors using subsumption?
- How many light sensors will you need for the light seeking behavior? What would your algorithm be if you were using two light sensors? (Think back to the obstacle-avoidance design, why would you want to use two bump sensors in the front as opposed to one bump sensor.)
- How will you be able to detect a 'strong' light source in the environment that will be noticable to your robot? What kind of calibration do you need?

Final Note: As part of your robot design, please make sure a motor controls each tread/wheel. Attach a rotation sensor to each motor by using a small gear train (2 gears is fine). You will not be needing the rotation sensor for this project but you will need it for next weeks lab (PID), therefore we want you to include these rotation sensors as part of your robot design. The sensors don't need to be hooked up to the sensor ports for this project, just make sure they are attached to the treads/wheels.

Paper Handin: Please staple your individual written report and write your group number on the front page. Your paper handin should describe your final design (both hardware and software) as well as any failed attempts you made. This would also include the number and type of sensors used and how you used them. You should explain your light seeking algorithm. Pseudocode is not necessary for this explanation, but recommended. In addition, explain what your robot does for calibration, including how it gathers data and what it does (statistically) once it gathers it. The arbitration module should also be explained in detail: which process subsumes which? How? How do they communicate if they need to?

Please electronically hand in your code as well **/course/cs148/bin/cs148_handin subsumption**. One copy of your code will be sufficient for the group. (Handin script only works from a Linux machine.)

Grading: Both your handin and your actual robot will determine your grade. Your robot will be scored in the following manner:

Wanders	2%
Follows light	15%
Calibration	8%
Avoids obstacles	5%
Proper priorities	20%
Total	**50%**

Your paper handin will be graded as follows:

Design explanations and process	5%
Light-following algorithm	10%
Calibration	10%
Bump detection/wander algorithm	5%
Subsumption algorithm/explanation	20%
Total	**50%**