Lab 04

Homing – Hybrid Control

Reading: *Ch. 3 of the text*

Read this entire lab procedure before coming to lab.

****

Purpose: The purpose of this lab is to use two photo resistors connected to your robot to implement light sensing behaviors. The light sensor will be used to implement a reactive controller similar to Braitenberg's vehicles 2 and 3. The light sensing will then be added as a layer to the robot's subsumption architecture and integrated into the random wander and obstacle avoidance state machine.

The second purpose is to use a type of locomotion called ***homing*** or ***docking*** with hybrid control to move the robot toward a light source. There will be a static light source placed in the environment which the robot can easily sense. The goal will be for the robot to move toward the beacon and stop just before hitting it. There will be no fixed path to the beacon, the robot should follow walls until the beacon is sensed, it should then leave the wall, keep track of its state and use the move to goal behavior to dock on the source while avoiding any obstacles along the way. Lastly, the robot should then turn 180 degrees and return to the wall to continue following as near as possible to the spot where it left. This will only be possible if the robot has kept track of its state.

Objectives:

At the conclusion of this lab, the student should be able to:

- Experiment with photoresistor sensors to determine a relationship between light conditions and change in resistance and voltage output
- Implement Valentino Braitenberg's *Vehicles* and examine the characteristics exhibited by the robot under simple motor-sensory couplings
- Implement a reactive controller and state machine on a mobile robot to integrate light sensing, random wander, and obstacle avoidance
- Implement a hybrid controller on a mobile robot to integrate wall following, path planning, homing and docking
- Write a technical memo to describe the purpose, method and conclusion of the lab

Equipment:

- Base Robot
- Range sensors
- 2 Photoresistors
- Flashlight
- 3 LEDs

Theory:

A photoresistor is a semiconductor device whose resistance is a function of light intensity. The schematic symbol for the photoresistor is shown in Figure 1. Because the resistance of the photoresistor varies with light intensity, the current that flows through it also varies with light intensity. However, we want to monitor a voltage, not a current, since the ADC (Analog-to-Digital Converter) on the micro-controller takes voltage measurements. We will be able to monitor a voltage from the photoresistor by creating a simple voltage divider circuit, as shown below.

The photoresistor used in this lab is designed to have a maximum resistance in the absence of light. As light intensity increases, its resistance decreases. As a result, as light intensity increases, the voltage, V_o, in the voltage divider circuit will also increase. You will monitor this voltage V_o as a measure of the light intensity seen by the photoresistor. On the CEENBoT, V_{ref} is the +5V supply and V_o is connected to an analog pin on the microcontroller.

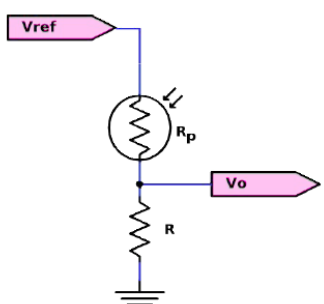

Figure 1: Photoresistor Wiring

In this configuration, the output voltage is given by,

$$V_o = \left(\frac{R}{R + R_p}\right) V_{ref}$$

Since the lighting environment and photoresistors will vary, your first task will be to measure the voltage output of your photoresistor for various dark and light settings.

Pre-Lab:

1. Create a state machine OR subsumption architecture OR flowchart for the Braitenberg Vehicles with obstacle avoidance and random wander.
2. Create a software design plan to implement the hybrid control architecture for wall following, homing, docking and path planning for your robot.

****

LAB PROCEDURE

****

Part I – Photoresistor test

1. Mount 2 photoresistors to analog pins on the front of the Arduino Robot. The sensors should face forward with one on the right in line with the right motor and one on the left in line with the left motor. You may need to use electrical, painters, or masking tape to anchor them to the desired location. If you have problems mounting your sensors, please go see the ECE technicians for help.
2. Next, you should write a program to read the analog value of the photoresistors to calibrate for dark and light values. You will use these values to account for ambient lighting and help you implement the reactive controllers. Complete Tables 1 and 2 for both sensors to give you some idea of how to use them for the next part. ***You should include these tables in your lab memo submission.***

Table 1: Environment Data

Conditions	Left Photoresistor (V)	Right Photoresistor (V)
Ambient light on the table		
Ambient light under the table		
Sensor covered		
In front of a flashlight or cell phone light		

Table 2: Distance and Angle of Incidence Data

		Left Photoresistor (V)			Right Photoresistor (V)		
Environment	**Distance (in)**	**Angle of Incidence -45°**	**Angle of Incidence 0°**	**Angle of Incidence 45°**	**Angle of Incidence -45°**	**Angle of Incidence 0°**	**Angle of Incidence 45°**
On the table	6						
On the table	12						
On the table	18						
On the table	24						
On the table	30						
Under the table	6						
Under the table	12						
Under the table	18						
Under the table	24						
Under the table	30						

Part II - Reactive Control

1. The first program you will write is a reactive controller inspired by Braitenberg's vehicle experiments. In this step, you will create a vehicle that is wired with excitatory connections where each sensor is connected to the motor on the same side. The program controls the left and right wheels based upon the light intensity seen by the left and right photoresistors (see Figure 2a). Turn on the RED, YELLOW, and GREEN LEDs when the behavior is active. ***The default motion for the robot is stopped until the light is sensed.***

2. How does the robot behave when (a) the light source is directly in front of the robot, (b) the light source is to one side of the robot? Is there anything about the robot's behavior that surprises you? ***Answer this question in the lab memo.***

3. Next, repeat parts 1 and 2 except that each sensor is connected in an inhibitory manner. This means the motor slows down as it gets closer to the light (see Figure 2b). Turn on the RED and YELLOW LEDs when the behavior is active. ***The default motion for the robot is driving forward slowly until the light is sensed.***

4. Next, repeat parts 1 and 2 except cross the connections between the motors and the sensors so that the left light sensor controls the right motor's speed and vice versa in an inhibitory manner (see Figure 2c). Turn on RED and GREEN LEDs when the behavior is active. ***The default motion for the robot is driving forward slowly until the light is sensed.***

5. Finally, repeat parts 1 and 2 with the connections still crossed between the motors and the sensors so that the left light sensor controls the right motor's speed and vice versa in and excitatory manner (see Figure 2d). Turn on the GREEN and YELLOW LEDs when the behavior is active. ***The default motion for the robot is stopped until the light is sensed.***

6. Braitenberg called these four light sensing behaviors, fear, aggression, love and explorer. These are the emergent behaviors that you did not explicitly program. Can you identify which of the four behaviors (fear, aggression, love, explorer) is exhibited for each of the prior motor/sensor connections? ***Answer this question in the lab memo.***

7. How did you decide on the position of the photoresistors? Were there certain lighting conditions that were more difficult or easier for the robot to sense? ***Answer this question in the lab memo.***

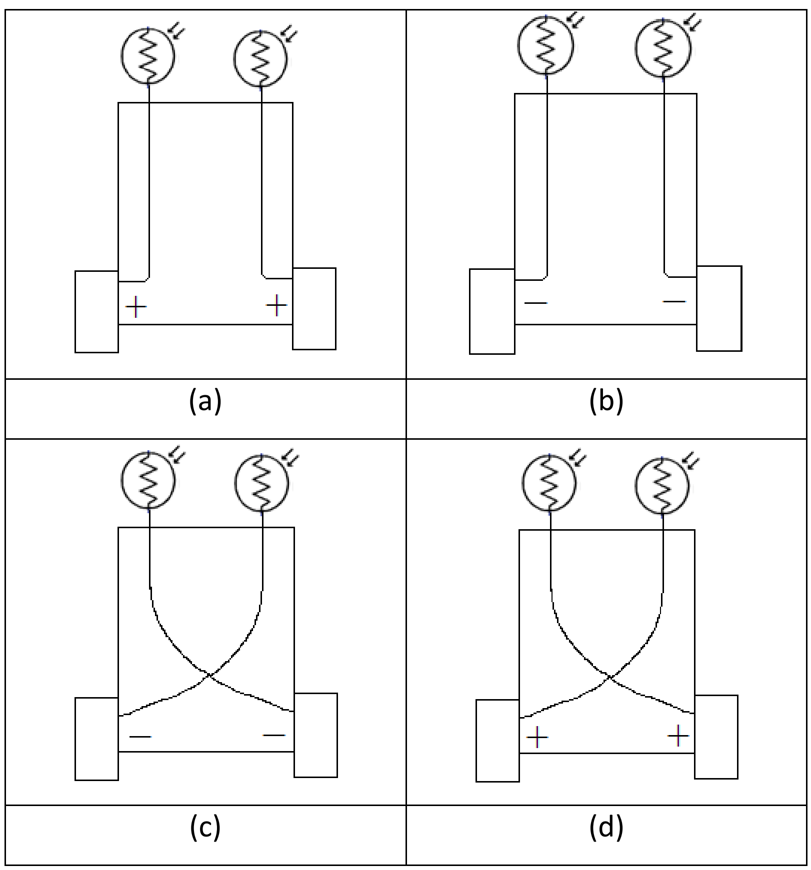

Figure 2: Valentino Braitenberg Vehicles

*Hints: In order to get variable speed on the robot stepper motors, you need to use **runSpeed()** or **runSpeedToPosition()**. You need to set the speed right just before you call these functions. Set the right and left motor speed proportional to the light. You will need to call runSpeed() and runSpeedToPosition() inside of the main loop as frequently as possible in order to see the variation. Do not use the RunToStop() function. You must use setMaxSpeed if setSpeed does not work to change the motor speed.*

Part III - Obstacle Avoidance

1. After testing each of the sensorimotor connections individually and confirming that they work correctly, you should add this light following behavior as layer on a subsumption architecture.

2. Layer 0 of the architecture should be avoid obstacle with collide and run away, layer 1 should be random wander with an input to avoid. Layer 2 should be light following which subsumes the output of random wander.

3. If the robot detects a light it should move with respect to the sensorimotor connections toward the light source while also avoiding obstacles. If the robot does not detect a light source or an obstacle then it should random wander. Figure 3 is an example of a subsumption architecture but it does not include random wander. Turn on RED LED to indicate an obstacle. Turn on a GREEN LED to indicate random wander.

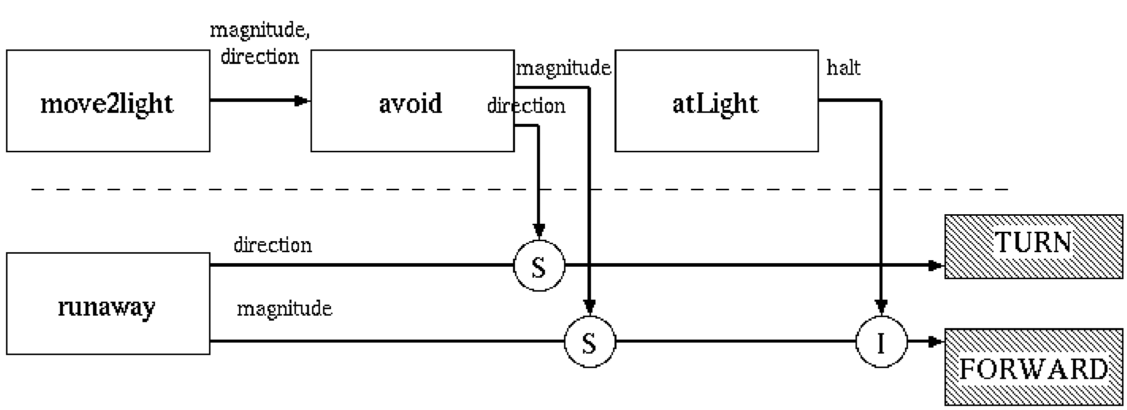

Figure 3: Sample Photophilic Architecture (no random wander)

Part IV – Homing or Docking

1. Next, you will design a complete different architecture for hybrid control. The hybrid control architecture that you will implement to home the robot includes a reactive layer (obstacle avoidance, wall following, move to goal (light following), path update), middle layer (arbitrator), and deliberative layer (current state, path plan, follow back to wall). This architecture is shown in Figure 4. Your code should be written in a modular fashion with functions such that it is evident where the planning, sensing and acting take place. Turn on a GREEN LED to indicate wall following.

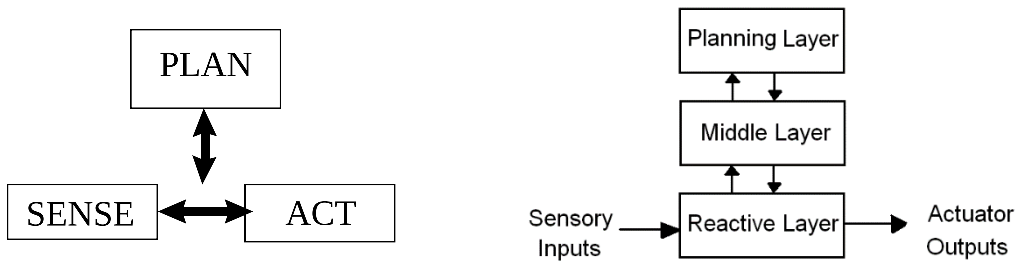

Figure 4: Homing Hybrid Control Architecture

2. The partial world map (representation) includes direction to the beacon and back to the wall with respect to the robot's current pose. This representation will be input into the deliberative layer for path planning. Updates to the path will be based upon feedback from the distance, heading and photoresistors. The middle layer will be used to make decisions about whether path updates are handled in the deliberative or reactive layer. The reactive layer will handle obstacle avoidance, wall following and move to goal behaviors. The robot should turn around and follow the path to drive back to the wall.

3. Based upon the above model, write code to home the CEENBoT robot to the light source (see Figure 5). The robot should come within one foot of the beacon without touching it. Turn on a YELLOW LED to indicate homing (moving to the goal, light following) and path planning (updates). Turn on a RED LED to indicate path following.

4. Test your final control algorithm for several different robot start points or beacon locations and summarize the results in your lab memo.

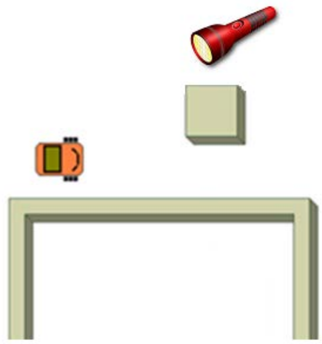

Figure 5: Robot homing

Part V – Docking the Robot and Return to the Wall

Improve the homing routine implemented in the previous part by docking the robot (back to the light) (see Figure 6). The robot should then follow the original path back to the wall to continue wall following. It is your choice which the direction the robot goes once it returns to the wall. Turn on the RED LED to indicate when the robot is path following back to the wall.

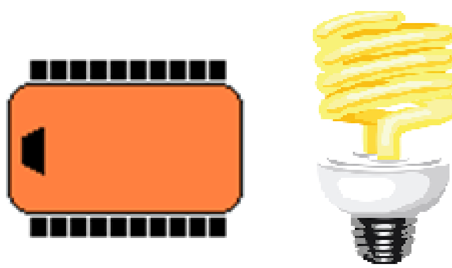

Figure 6: Robot Docking

Submission Requirements:

Software Design Plan

For each lab you will submit a software design plan which may be in the form of pseudocode, a flowchart, state machine or subsumption architecture. You will show this plan to the instructor during class, the plan will be graded and you will get feedback on implementation before designing the full system.

Demonstration

Bring your robot fully charged to class every day! Plug it in overnight.

This week's demonstration will involve exhibiting each of the four vehicles described in part 2 of the lab procedure. It would be advisable to write all of the reactive controllers in one program in a loop with a delay between switching to quickly transition from one to the other. The next part of the demonstration will involve showing the full state machine with the integration of light sensing, obstacle avoidance, and random wander. In the next part, the robot will be placed in the environment, wander until it finds a wall and then follow the wall until the beacon is detected. The robot should then move to goal and stop within one foot of the beacon. In the next part, the robot should turn and dock on the beacon. Lastly, the robot should return to the wall as close as possible to where it left off and continue to follow the wall.

Program:

The program should be properly commented and modular with each new behavior representing a new function call. The design of the subsumption architecture should be evident from the program layout. You should use the GUI, keypad, LCD and speech module as needed to illustrate robot state, input and output data.

Memo Guidelines:

Please use the following checklist to insure that your memo meets the basic guidelines.

- ✓ Format
  - o Begins with Date, To , From, Subject
  - o Font no larger than 12 point font
  - o Spacing no larger than double space
  - o Written as a combination of sentences or paragraphs and only bulleted list, if necessary
  - o No longer than three pages of text
- ✓ Writing
  - o Memo is organized in a logical order
  - o Writing is direct, concise and to the point
  - o Written in first person from lab partners
  - o Correct grammar, no spelling errors
- ✓ Content
  - o Starts with a statement of purpose
  - o Discusses the strategy or pseudocode for implementing the robot remote control (includes pseudocode, flow chart, state diagram, or control architecture in the appendix)
  - o Discusses the tests and methods performed
  - o States the results and or data tables including error analysis, if required

- Shows any required plots or graphs, if required
- Answers all questions posed in the lab procedure
- Clear statement of conclusions
- Include software design plan in the appendix (reference design plan in the text)

Questions to Answer in the Memo:

1. How reliable was the photoresistor at detecting the light in different environments, various distances and angles of incidence (head on, slightly left, slight right).
2. How significant was the difference in photoresistor voltages for the left and right sides. How did you use this difference to extract directional information to move the robot toward the beacon?
3. How did you integrate the light sensors into the obstacle avoidance behavior?
4. How reliable was the photoresistor at detecting the light at various angles and distances. Compare and contrast sensor data.
5. How significant was the difference in sensor data based upon distance from the source? How did you use this difference to extract distance information to move the robot toward the beacon?
6. What does the hybrid control architecture for your design look like? What was on the planning layer? Middle layer? Reactive layer?
7. What was your general strategy for planning the path back to the wall from the beacon?
8. How did the architecture respond to differences in robot start position or beacon location?
9. How did the robot's hybrid controller respond to dynamic changes in the environment (i.e. other robots and people) and compare this to purely deliberative control.
10. Were there any challenges in implementing the homing routine?
11. What could you do to improve the robot homing?
12. How did docking the robot modify the control architecture or algorithm?

Grading Rubric:

The lab is worth a total of 30 points and is graded by the following rubric.

Points	**Demonstration**	**Code**	**Memo**
10	Excellent work, the robot performs exactly as required	Properly commented, easy to follow with modular components	Follows all guidelines and answers all questions posed
7.5	Performs most of the functionality with minor failures	Partial comments and/or not modular with objects	Does not answer some questions and/or has spelling, grammatical, content errors
5	Performs some of the functionality but with major failures or parts missing	No comments, not modular, not easy to follow	Multiple grammatical, format, content, spelling errors, questions not answered
0	Meets none of the design specifications or not submitted	Not submitted	Not submitted

Upload Details:

You must submit your properly commented Sketch code & memo to the Moodle DropBox by midnight on Sunday. Check the course calendar for the lab demonstration due date.