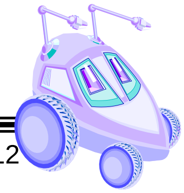

Lab 4

Reactive Control - Light Sensing

Reading: *Introduction to AI Robotics (Ch. 4)*

(Demonstration due in class on **Thursday**)

(Code and Memo due in Moodle drop box by midnight on **Sunday at midnight**)

Read this entire lab procedure before coming to lab.

****

Purpose: The purpose of this lab is to use two photo resistors connected to implement light sensing on the CEENBoT. The light sensor will then be used to implement a reactive controller related to Braitenberg's vehicles 2 and 3.

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

- Experiment with a photo-resistor sensors to determine a relationship between light conditions and change in resistance and voltage output
- Implement Valentino Braitenberg's *Vehicles* to see the impact of simple reactive controllers and the characteristics exhibited by the robot under simple motor-sensory couplings

Equipment: Base Robot

2 photoresistors

Software: AVR Studio 4 (32-bit) available at

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

WinAVR GCC toolchain (http://winavr.sourceforge.net/)

CEENBoT API static library available at

http://www.digital-brain.info/downloads/capi324v221-v1.09.002R.zip

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 photo-resistor 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 photo-resistor by creating a simple voltage divider circuit, as shown below.

The photo-resistor 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, Vo, in the voltage divider circuit will also increase. You will monitor this voltage Vo as a measure of the light intensity seen by the photoresistor. On the CEENBoT, Vref is the +5V supply and Vo is connected to the ADC on the microcontroller.

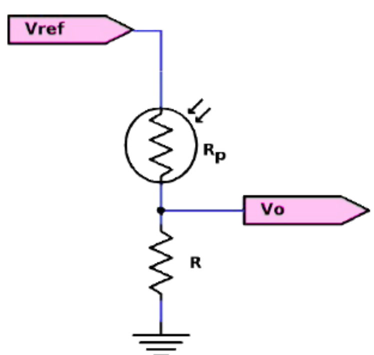

Figure 1: Photoresistor Wiring

In this configuration, the output voltage is given by,

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

On the CEENBoT, R is 10 kΩ and the photo resistor has a resistance range from 5 kΩ (maximum light reception) to 20 kΩ (no light reception). Then the voltage output is given by (no light) $1.67\ V \ \leq\ V_o \leq 3.33\ V$ (maximum light)

Since the lighting environment and photoresistors will vary, your first task will be to measure the voltage output of your photoresistor for a dark and light setting. The ADC will read the analog voltage data by using the API and this code has been written for you in the *RangeSensors* project.

Purposeful navigation requires cognition and it must have a specific method for avoiding obstacles while continuing to make progress toward the goal. Three techniques to avoid obstacles while continuing to make progress toward the goal are the Bug1, Bug2 and Tangent Bug. The **Bug1** is the simplest algorithm although not the most efficient because it involves following the contour of an obstacle until it is fully circled and recalling the point when it was the closest to the goal. The robot then returns to that point and leaves the obstacle to continue toward the goal. Figure 2 provides an example of the Bug1 algorithm.

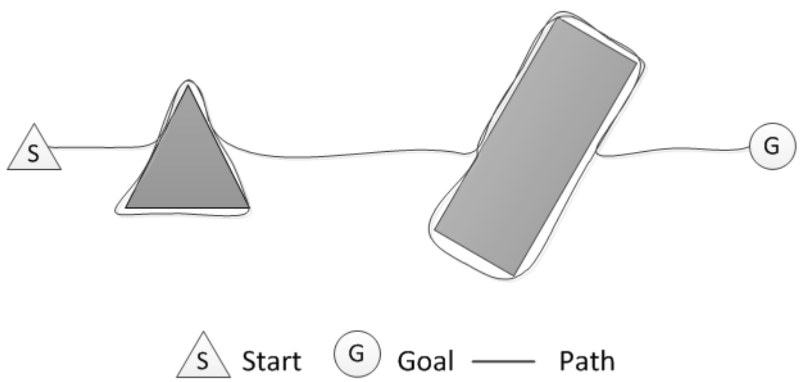

Figure 2: Bug1 Obstacle Avoidance

The **Bug2** algorithm involves always following the obstacle on the left or right side and leaving the obstacle if a direct connection to the goal location is intersected. This method will have a shorter travel time than Bug1. Bug1 is an exhaustive search algorithm while the Bug2 will move toward the goal at the first opportunity. Figure 3 shows an example of obstacle avoidance while moving to a goal by using the Bug2 algorithm.

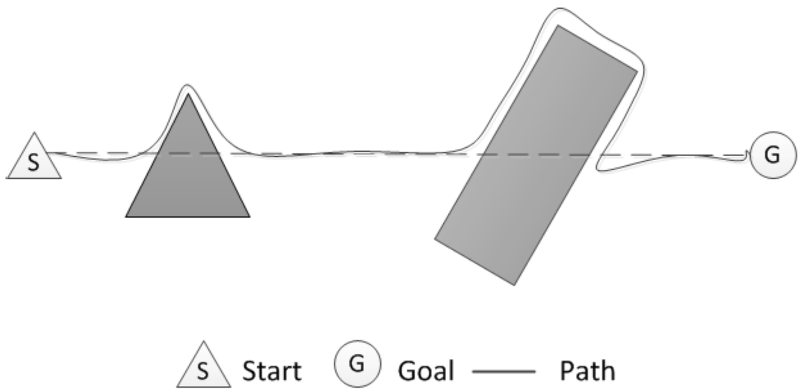

Figure 3: Bug2 Obstacle Avoidance

For the **<u>Tangent Bug</u>** algorithm, the robot will move toward the goal until an obstacle is detected and then follow along the tangential surface of the obstacle until it can once again can move toward the goal. This type of obstacle avoidance approaches the globally optimal path. Figure 4 provides an example of obstacle avoidance with the Tangent Bug algorithm.

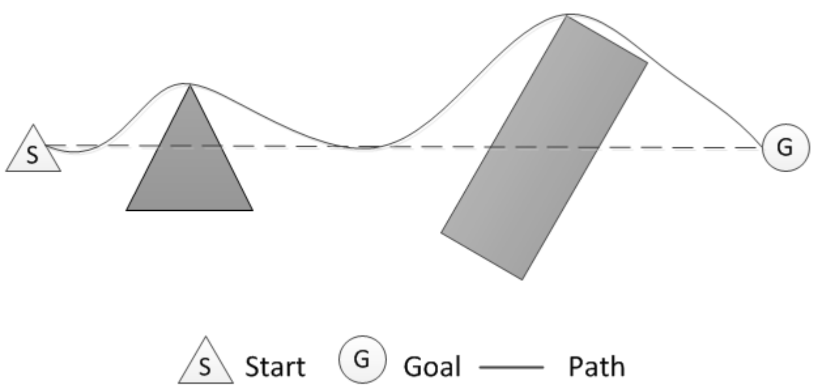

Figure 4: Tangent Bug Obstacle Avoidance

****

LAB PROCEDURE

****

Part 1 – Photoresistor test

1. Originally, your photoresistors will be mounted on the breadboard on the robot. You will need to adjust them and selecting a mounting location to maximize light and dark sensing. You may need to use masking tape in order to anchor them to the desired location.

2. Next, you should use the range sensors program to read the values from each sensor for dark and light readings to get a feel for how to use them to implement the controllers. Complete the following table for both sensors. You should include this table in your lab memo submission.

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

Part 2 - 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 5a).

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?
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 5b).
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 5c).
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 5d).

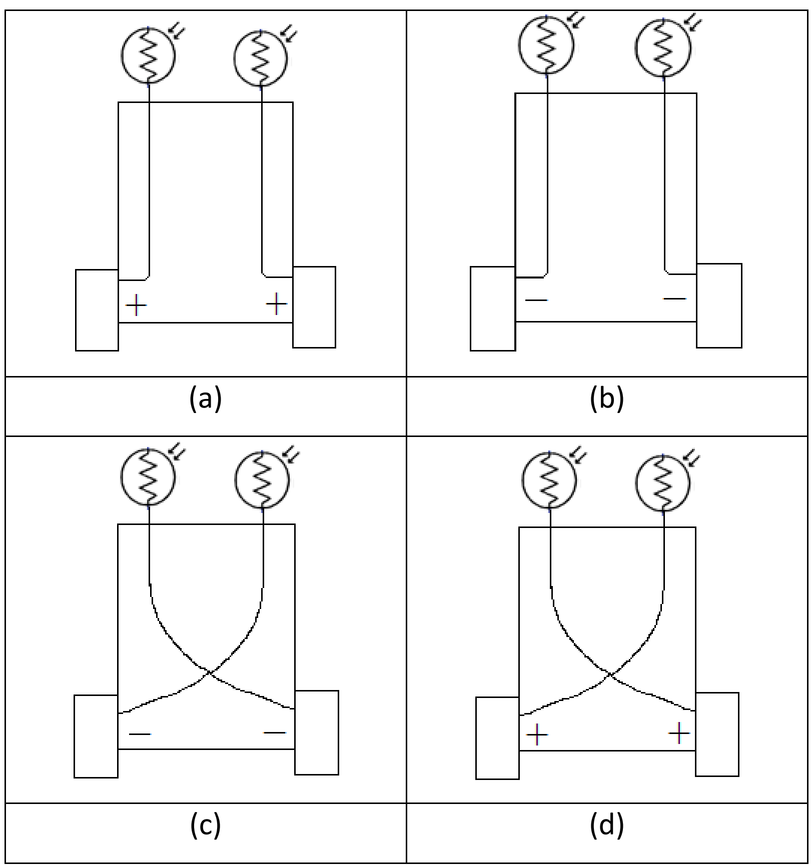

Figure 5: Valentino Braitenberg Vehicles

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?

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?

Part 3 - Obstacle Avoidance (layer 0)

1. After testing each of the sensorimotor connections individually and confirming that they work correctly, you should make this Layer 1 of the subsumption architecture.

2. Similar to the prior 2 labs, Layer 0 of the architecture should be obstacle avoidance. The robot should move with respect to the sensorimotor connections with respect to the light source while also avoiding obstacles. If the robot does not detect a light source or an obstacle then it should remain still (see Figure 3).

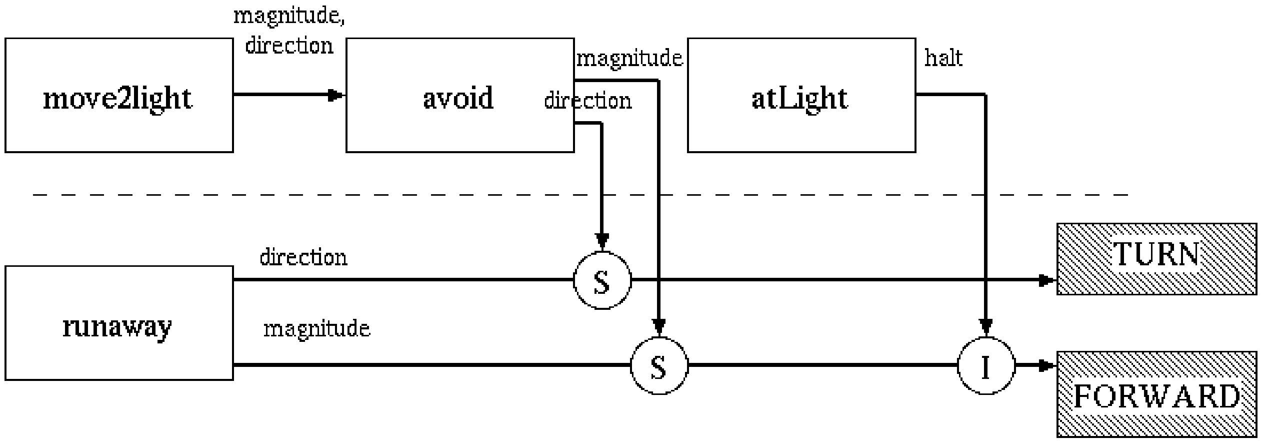

Figure 3: Sample Photophilic Architecture

3. As an alternative to part 2, you can program the robot to avoid obstacles by using the Bug1, Bug2 or Tangent Bug algorithms. It is your choice whether to use feelforce or one of these algorithms in your final demonstration.

Demonstration:

This week's demonstration will involving 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 that is selectable by pressing a push button on the robot to quickly transition from one to the other. The second part of the demonstration should include some type of obstacle avoidance, such as feelforce, Bug1, Bug2 or Tangent Bug.

Bring your robot fully charged to class on Thursday for the demonstration. Note that you always must re-flash the factory firmware and plug in the AC adapter in order for the robot to charge. Alternately, you can put the robot battery in the RC car battery charger. Note that this is a fast charger and will not last as long as the outlet charge.

See prior labs for information on the program, memo, submission requirements and grading rubric.

Submission Requirements:

You must submit you properly commented code as a zipped folder of the C file and the lab memo in a zipped folder by **11:59 pm on Sunday** to the Angel Course Drop box. Your code should be modular with functions and classes in order to make it more readable. You should use the push buttons, buzzers and LCD to command the robot and indicate the robot state during program execution.