

Design Lab 9: Visually Attractive

You can use any computer that runs Python to use **CMax**. You will also need a robot and a lab laptop.

- **Athena machine:** Do `athrun 6.01` update and add `-f 6.01`.
- **Lab laptop:** Do `athrun 6.01` update.
- **Personal laptop:** Download design lab 9 zip file from course web page.

Code is in `lab9/designLab/`.

See the last page of this handout for connector pin-outs.

Our ultimate goal is to build a 'head' that we can put on the robot that will be able to sense and track light. Last week, you started considering various circuit designs to accomplish this task. Today, you will complete this.

1 Pointing Circuit

- **If you have not completed checkoff 3 (the reference design) from last week's lab, you should do that now.**
- **If you have completed checkoff 3 (the reference design) from last week's lab but you have not completed the alternative design (upload 8), then skip checkoff 1 for now and build the reference design. You can complete checkoff 1 after you finish the other checkoffs.**

Your partner and you should each have designs and simulations for two pointing circuits.

Check Yourself 1. Discuss your circuit and simulations with your partner. Pick which circuit you are going to build.

You can start to lay out your circuit using CMax while you wait for a staff member to come and do the following check-off with you.

Checkoff 1. Discuss your designs and circuit simulations for each of the designs with a staff member. Explain your results. Why did you choose your design?

Lay out the design you chose. You can run CMax by going to a Terminal window, navigating to lab9/designlab and typing

```
> python CMax.py
```

Or, you can start Idle with `-n`, open the file `CMax.py` and do **Run Module**. If you open `CMax.py` through Idle, close it down by killing Idle.

In your circuit layout use a Robot Connector to provide power and ground for your circuit; do not use separate power and ground for your board.

Check Yourself 2. Simulate your circuit in CMax using the same eyeServo test that you used last week. Make sure that the result of the simulation matches your previous results.

- Build the circuit on a proto board, to match your CMax layout. Start with a gain of 1 but be sure your circuit can be simply modified to obtain higher gains. Connect your proto-board to a head with a red cable and to a robot with the yellow cable that is already attached to the robot.

Make sure that there is no black wire connecting your proto board to the motor.

Verify that your circuit works by measuring the voltages across the motor, being sure that they behave appropriately as you move the head by hand. Demonstrate the correct behavior to a staff member and they will give you a black cable to connect the motor.

- Plug in the black cable. **If the head turns to one side and slams against one of the stops because the voltages are not correct, turn off the robot IMMEDIATELY.** Test the robot head by moving a light in front of it and making sure it is behaving reasonably.
- Connect the center pin of the neck potentiometer (pin 2 on the head connector) to the first analog input (pin 1 on the robot connector). You will also need to connect the power and ground pins of the neck potentiometer to power and ground on your protoboard.
- Use the brain `turnToLightAnalogBrain.py` to plot the neck potentiometer as a function of time, and report the settle time of the signal. Follow these steps:
 1. Unplug the lego motor.
 2. Turn the head 45 degrees from the light (in either direction).
 3. Start the brain and wait a second or two.
 4. Plug in the motor.
 5. Stop the brain a second or two after the head has stopped moving, or it is clear that it will not.

You should see a graph appear and an estimated settle time will be printed on the Soar window. You should verify the printed settle time by examining your graph closely. (Sometimes our settle-time estimator is confused if there are minor variations at the end of your graph). The settle time is the number of steps it takes for the head to converge on its final direction (ignoring the initial period in which it is constant, before the cable is plugged in).

- Pick a good gain for your circuit so that you get as fast a response as possible (it shouldn't have significant oscillations, but a little overshoot is fine) over a range of distances from the light. Gather data with at least two different gains in your circuit and at least two distances. Save the graphs and settle times for each.

Checkoff 2. Illustrate your circuit and its performance at two different distances with the two gains you investigated. How does the settling time behave with gain and with distance? **Keep your plots and measurements to discuss in your interview.**

2 Analog Bull's Eye

The actual photoresistors in the heads have some variation in their response to light. You saw this in design lab 7, when you needed to adjust the inputs from the eyes to get the robot to point reliably at the light. Consider how to add one potentiometer to your circuit so as to improve the pointing accuracy of your circuit. Be prepared to discuss your approach for dealing with this.

Tune your circuit so that the laser lands reliably on the lampshade.

Checkoff 3. Discuss your circuit and demonstrate its accuracy in pointing the laser at a light.

3 If you have time... Pet Robot

We would like our robot to follow a bright light around the room. The head you have just built is capable of turning much more quickly than the (heavy) robot, so we will construct a two-level control system in which the head turns quickly to track moving light and the robot body turns in such a way as to keep the head pointing forward relative to the body. This is analogous to the human visual system in which the eyes can move quickly to track motion and the head turns in the direction the eyes are gazing.

Use the head as configured in the previous part, so that it automatically turns toward a bright light. Mount the robot head on the robot body so that the head faces forward (in the same direction that the sonar array points).

1. Design a behavior for that robot that will use signals from the robot head to turn the robot toward a bright light.

Check Yourself 3. What control variable from the head is important for turning the robot toward the light? Explain.

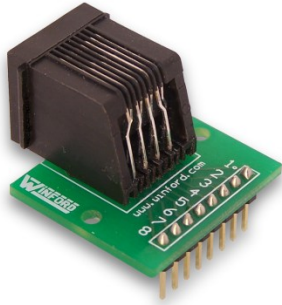
2. You can read the voltages from pins 1, 3, 5, and 7 of the robot connector from a soar brain as the list of four values `inp.analogInputs`, where `inp` is an instance of the `io.SensorInput` class.
3. Write a soar brain to implement your controller. The output of the controller should be an `Action` that specifies the rotational velocity `rvel` of the robot. We have provided a skeleton in `roverBrainSkeleton.py`

Hint: You should debug the robot behavior by tilting the robot backwards so that the wheels do not touch the ground and watching to see that the behavior is reasonable before unleashing your robot on the world. Also, start with the black cable disconnected, so that you can manually turn the head and observe if the wheels turn correctly.

Check Yourself 4. Demonstrate that your robot turns toward a bright light. What is the highest gain (in the software control loop) for which you get stable responses?

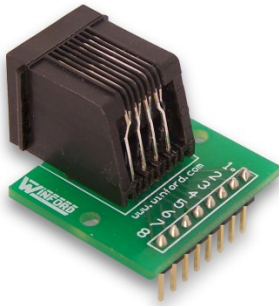
4. Rework the brain to make the robot behave as a well-trained pet. Your robot's behavior depend on proximity to the light. If the light is off, the robot should stand still in obedience. If the light is on, the robot should approach the light, positioning itself approximately half a meter from the bulb. For this part, the soar brain will need access to not just the neck pot but also some measure of the light intensity. Figure out which voltages on your circuit can provide this information. Connect the relevant voltage(s) to one or more of pins 1, 3, 5, and 7 on the robot connector. **This will be easier or harder depending on which design you implemented for your pointing circuit. Talk to an instructor for ideas if you don't see how to do it for your circuit.**

Robot connector



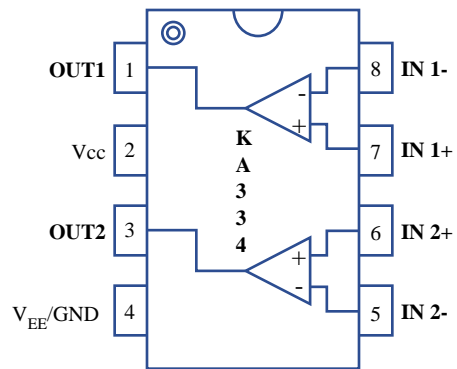
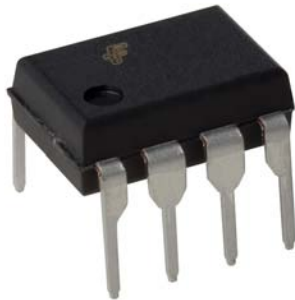
- pin 1: V_{i1} analog input #1
- pin 2: +10 V power (limited to 0.5 A)
- pin 3: V_{i2} analog input #2
- pin 4: ground
- pin 5: V_{i3} analog input #3
- pin 6: V_o analog output
- pin 7: V_{i4} analog input #4
- pin 8: +5 V power (limited to 0.5 A)

Head connector



- pin 1: neck pot (top)
- pin 2: neck pot (center)
- pin 3: neck pot (bottom)
- pin 4: photoresistor (left)
- pin 5: photoresistor (common)
- pin 6: photoresistor (right)
- pin 7: V_{M+} Motor drive +
- pin 8: V_{M-} Motor drive -

Op-Amp



(8-DIP)

Figure by MIT OpenCourseWare.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.01 Introduction to Electrical Engineering and Computer Science I
Fall 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.