ME 599/AA546/EE546: Biology-Inspired Robot Control

University of Washington, Autumn 2016

Instructor: Prof. Sawyer B. Fuller

Problem Set 1

The idea of this problem set is to promote good software development practices for simulation code, and to explore dynamics and control simulations from a state-space perspective. You will need access to MATLAB, which is available through the Mechanical Engineering Department's remote desktop server (more information at `https://www.me.washington.edu/computing.html`) or for purchase for a fee at U. Wasahington's U-Ware site. You may also write software in another programming language, e.g. Python, but you will have to re-write the skeleton code that is provided with this problem set.

You are encouraged to work together to work out solutions, but note that you must submit your own work.

1. Please read Braitenburg1984, chapters 1–4, available under the `papers` section of the course website. You will be simulating the behavior of some of the vehicles he describes. Your vehicle will be moving on flat ground, such as a tabletop.

2. The first vehicle is vehicle 1, which follows a straight line. Let's suppose its sensor is responding in proportion to brightness of the light as detected by the sensor (rather than heat as suggested by the book). To implement this in simulation, suppose its state is given by its position and orientation, $q = [x, y, \theta]$, and it moves forward with a speed determined by the sensor reading. Skeleton code for this problem is provided online with the problem set in `braitenburg_1.m`. Please fill in the missing components so that your vehicle moves forward and its speed varies in response to the light brightness. Assume your sensor response is inversely proportional to the distance to the light source. Submit your updated software code and an image of the resulting motion of the vehicle.

3. Next, you will implement vehicle 2. For these robots, you must compute distances to the light source for two sensors, each placed in a different location. Skeleton code is in `braitenburg_2.m`. Submit your software code and plots showing light following and light avoiding maneuvers.

4. Next, you will implement vehicle 4. For these robots, you will need to implement an additional function on your vehicle 2 code that produces a non-monotonic response to an input light reading intensity. For example, your function could start at zero, rise up to a maximum of 1, and then decay back to zero, by implementing a series of `if-then-else` blocks. By adjusting parameters, see if you can get this robot to perform some sort of interesting motion around the light source, such as an orbit that takes it closer and farther from it. Submit your code and a representative figure. Note: your robot only receives feedback from its sensors, so it can't use any other information, such as its position, for feedback control. This robot's behavior can be tricky to reason about!

5. Last, we will make a robot that uses *memory* about previous sensor readings to perform maneuvers. Add two additional states to your robot's state vector q that encode the time integral of light intensity of the two light sensors. Use this to create a robot that stops once it has spent enough time in the vicinity of a light source. Submit your code and a representative figure.