**To:** Dr. Berry  
**From:** Peter Heath, Matthew Schack, and Data  
**Date:** 1/21/16  
**RE:** Lab 06 hybrid control and homing

The purpose of this lab was to make Data able to follow a wall, go to a light, dock with the light, and return to the wall. To do this we implemented hybrid control with a state machine. Each state was a different stage of the path such as: following wall or docking with the light. While the state told Data what to do the hybrid control corrected for little errors that came up on the way.

The planning layer of the hybrid control looked like a state machine. Depending on what state Data was in Data would look at different sensors and move in different ways. The middle layer controlled the switching between the states. It would look at sensor readouts and decide if we had reached the end of the state or not. The reactive layer looked at different sensors to correct for errors that may have popped up along the way.

To plan a path back from the wall we had Data turn 180° then move forward till the forward sensors found the wall. This worked as a path because the light sensor could only see light in a line; this means that when going to the light Data would move in about a straight line. In order to have Data retrace the steps all we had to do was turn around and move straight.

The photoresistor was not reliable at detecting anything. The only way we could get it to properly find a light was if the light was super bright and Data was in the dark. This ended up just flooding the sensor, so while we could see a light we could not see distance, size, or location well.

The left and right photoresistors had slight differences if one of them was only partially in light. We used this information to slightly correct our path in the Light Follow state. We did this by taking the difference in sensor values and multiplying it by a Kp for light. We added this number to a base speed for the left wheel and subtracted it from the base speed for the right wheel.

The value did not change much for the photoresistors as it approached the light, so in order to tell the distance to the beacon we used our front sonar sensor to look for an object while we were in the Light Follow state.

The architecture responds well to different start positions and beacon locations. Data can follow the wall on the left or the right depending on which wall is closer when the program starts, and Data will be able to see the light at any angle as long as the light is somewhat in front of Data.

The hybrid control did not respond well to dynamic changes in the environment. Data was expecting specific things depending on the state and the sensor readouts were used to correct for error. If the light data was going to was removed Data will continue moving forward searching for both the light and the physical beacon. While Data did not respond great to the changes it still responded better than if it was purely deliberative control. The hybrid control is able to respond to little bits of error and correct Data while deliberative control would not be able to correct like that.

There were some challenges implementing the light homing routine, mainly making an equation that would correct based off of the light while maintaining a fairly constant speed. However, we managed to make an equation that worked off of the differences between the light sensors and was able to make data home into the light. We could improve the method by making Data turn faster in the beginning of the light homing method.

To make Data dock with the light we just added in one more state in our control architecture. Because our control architecture is so modular we are able to easily add any number of states to our program.

State Control

Possible states:

* Wall following
* Turn towards light
* Light homing
* Docking
* Return to wall

Motor output

Sensor inputs

Motor Controller

Motor speed is based off of state and sensor inputs

State Changer

Reads sensor inputs and decides if the state should be changed