

P545 Lab 4 Obstacle Avoidance II

Bryce Himebaugh

October 17, 2008

1 Introduction

In this lab, you will be adding capability to your driver to allow it to avoid obstacles as they are sensed. We will be using a synthetic obstacle sensor allow us to get confidence in our avoidance/planning algorithm before we tackle the real sensor.

2 Synthetic Sensor Component

2.1 Overview

Figure 1 provides an illustration of the sensing area of a physical laser sensor that we will be using in Lab 5. This laser can detect obstacles to a range of 4 meters and has a 145 degree sweep angle. However, it will introduce noise and other complications that we do not want to be confronted with initially. To defer dealing with these complications, we will be using a CartFS component that synthesizes the ideal properties of the physical laser.

2.2 SynLaser CartFS Component

The SynLaser component is designed to dynamically produce an obstacle list based on a static list of potential obstacles. Using the GPS location of the cart and the cart's heading, the synlaser component decides which obstacles in the list would be in view of the physical laser. All of the obstacles that are in view at a given instant are provided as a `obstacle_list` in the `synlaser_s` file.

The obstacle list will be a list of lists where each element of the main list is in the format `[tag, (lat,lon), lbo]`. If there are no obstacles in view `obstacle_list` will be `[]`. The tag string is the name of the obstacle as noted on the map in Figure 2.

```
obstacle_list = [{"cone5", (39.182154, -86.522190), 0.25},
                 {"cone6", (39.182172, -86.522218), 0.25}]
```

You should start monitoring these obstacles in your driver by adding a reader for the `./synlaser/synlaser_s` file in your code. You can model this addition after the way that it is done for the `gps_s`.

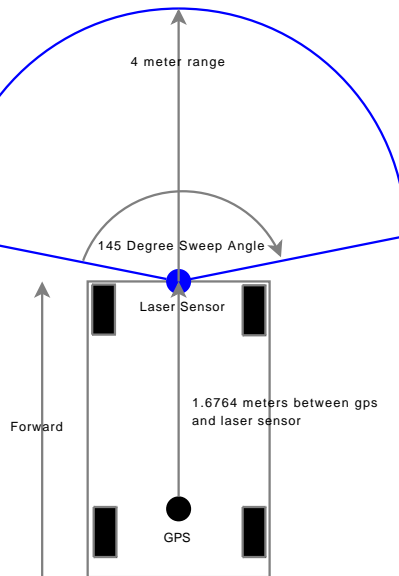


Figure 1: Cart Synthetic Obstacle Sensor

We will be using the same base course that we used in Lab 3 with the addition of 8 obstacles - the obstacle from Lab 3 is still in the course. See Figure 2 for reference. For reference and to aid in your prototyping, the coordinates of the obstacles are provided below. Please do not statically plan the route for the driver as this will limit the utility of your driver when we start using the physical laser.

```
obstacle_list = [{"cone1", (39.181933, -86.521931), 0.25},
                  {"cone2", (39.182036, -86.521747), 0.25},
                  {"cone3", (39.182033, -86.521654), 0.25},
                  {"cone4", (39.182119, -86.522024), 0.25},
                  {"cone5", (39.182154, -86.522190), 0.25},
                  {"cone6", (39.182172, -86.522218), 0.25},
                  {"cone7", (39.182015, -86.522356), 0.25},
                  {"cone8", (39.181957, -86.522167), 0.25},
                  {"cone9", (39.181909, -86.522167), 0.25}]
```

2.3 Map of Obstacles

3 What to turn in

Write a concise summary of the testing that was performed and the results. Using collected gps data, include a plot of the path that the cart took as it traversed the course for at least two laps. Post this code to your svn lab directory

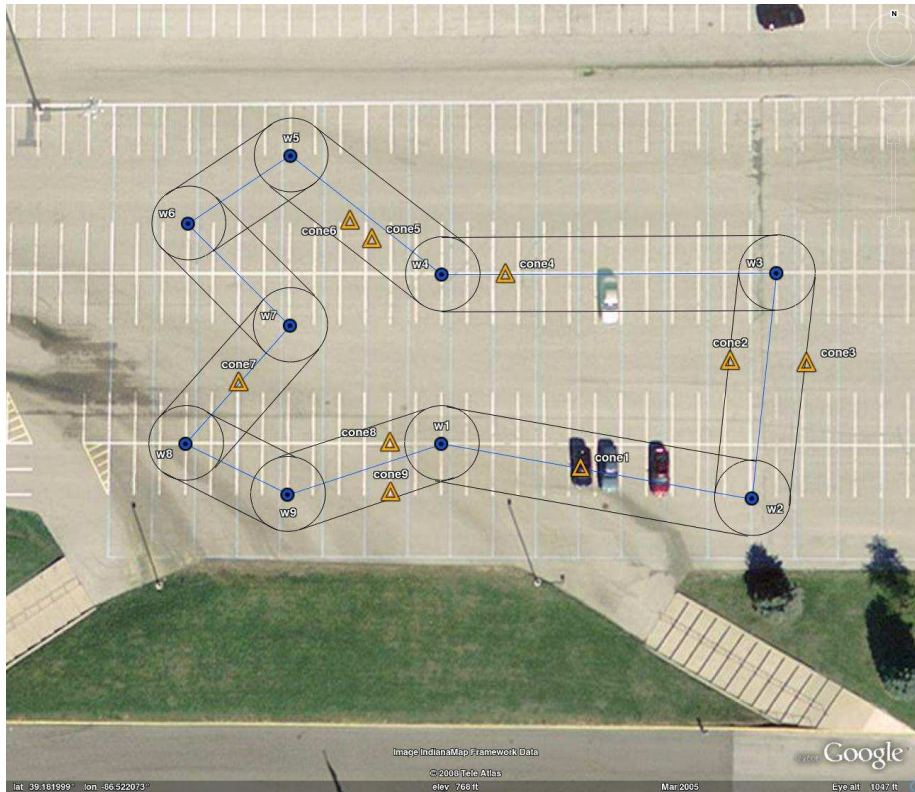


Figure 2: Route Path from Lab3 with additional obstacles

for lab 4. Drop me an email when this is complete. This should be complete by 10/31/08.