P545 Lab 2

Bryce Himebaugh

September 23, 2008

1 Introduction

In the square driver, you designed a heading controller that adjusts the turn radius of the steering subsystem on each clock cycle to minimize the difference between the current compass heading and the desired heading. In this lab you will utilize this component to follow a course defined by an arbitrary set of GPS coordinates.

For the square driver, you adjusted the desired heading onto 4 fixed headings - 0, 90, 180, and 270 degrees. In this lab, you will be continuously computing a new desired heading based on the cart's current gps coordinates and the coordinates of the current waypoint. The heading controller will then hold this course until the cart is within a defined distance from the waypoint. Once the cart has come close enough to the waypoint, it starts to head toward the next waypoint in the list.

2 Waypoints

A waypoint is described by a 2D circle with center at a defined lat/lon coordinate and radius defined by a parameter call the lateral boundary offset (LBO). Each waypoint will be numbered and will include a speed limit for the segment. In the references section, a link has been included to the DARPA RDDF Format Description. This document is a good overview regarding the format, however, we have made a few unit adjustments. LBO is in meters and speed is measured in m/s.

3 Angle and Distance to a Waypoint

In the design of your GPS driver, you will need to take into account the distance and angle to the next waypoint. This will take your current position and desired position as inputs and return the azimuth and distance to the waypoint.

There are a number of different way to make this computation. I have found that using the geopy module provides an acceptable solution. A link to the geopy site is provided in the References section. You may make this computation using this technique or using a technique of your choice.

```
from geopy import distance

current_latlon = (39.181903,-86.522041)
waypoint_latlon = (39.182169,-86.522007)

meters_to_target = distance.distance(current_latlon,waypoint_latlon).kilometers * 1000
heading_to_target = distance.distance(current_latlon,waypoint_latlon).forward_azimuth
```

4 Assignment Overview

Design a GPS driver that follows a course defined by a list of gps waypoints. As with the square assignment, you will not be controlling speed with this driver. Use a fixed throttle setting that keeps the speed of the cart in the 1-3 m/s range. When the cart reaches the end of the list, start again at the top of the list of points.

```
waypoints = [[1,39.181917,-86.5221208333,1.5,3.0],\
                 [2,39.1818975,-86.521724,1.5,3.0],\
                 [3,39.182143,-86.5217033333,1.5,3.0],\
                 [4,39.182199,-86.5220985,1.5,3.0],\
                 [5,39.1819156667,-86.522309,1.5,3.0],\
                 [6,39.1819645,-86.522398,1.5,3.0],\
                 [7,39.1820415,-86.5223095,1.5,3.0],\
                 [8,39.1821313333,-86.5223926667,1.5,3.0],\
                 [9,39.1822116667,-86.522302,1.5,3.0]]
```

5 References

1 RDDF Format

2 Geopy

6 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 on the same graph with a plot of the straight line course.

Post this code to your svn lab directory for lab 2. Drop me an email when this is complete. This should be complete by 10/03/08.