COS 495 – Autonomous Robot Navigation
Fall 2011

Lab 2 Manual

2.1 Goals

The programming application goal of this lab is to:

- Use P Control to steer the vehicle to longitude/latitude points.

2.2 Recover your P control code

Dig up your old P control code from the previous code. Find the location where you entered 90 degrees as the desired angle to track. Create a variable of type double called `theta_desired`, (if you don't already have one).

2.3 Modify your P control code

Now you will add to your code the ability to change your desired angle to track, `theta_desired`. The desired angle will be based on current GPS location, and 3 desired locations to track. Here is what the program should do:

1. Set a desired task point location to be Burr Hall
2. Set the desired angle according to the desired task point location and current location
3. For 30 seconds, use P control to steer towards the desired location (only yaw control)
4. Set a desired location to be the Frick Chemistry Lab
5. Set the desired angle according to the desired task point location and current location
6. For 30 seconds, use P control to steer towards the desired location (only yaw control)
7. Set a desired location to be the Von Neumann Hall
8. Set the desired angle according to the desired task point location and current location
9. For 30 seconds, use P control to steer towards the desired location (only yaw control)

Note: atan2 might be useful!

To accomplish this you will modify the lab 1 base code (downloadable from the website). A few notes about the base code:

1. There are global variables that can be used as the desired way points the vehicle should try to point to. For example, …. (note these are California Long / Lat's)

```
double[] desiredLong = {-120.657939,-120.66282,-120.657939 }; // x axis
double[] desiredLat = {35.299558, 35.300346, 35.299558 }; // Y axis
double currentLong = -120.662348;
double currentLat = 35.30046;
double currentHeading = 0;
```

2. The local variable desPointIndex has been added to let you keep track of which the desiredLong and desiredLat points you are tracking. You don't have to use it if you don't want to.

```
int desPointIndex = 0;
```

3. The currentLat, currentLong, and currentHeading variables are set for you with simulator functions. To simulate a stationary robot in the lab, the currentLat and currentLong stay constant. The currentHeading changes as if the vehicle could actually be steered with the yawFin actuated. The following code is already in place for you:

```
// Get localMeasurements
currentLat = simulateLat();//vehiState.getLatitude();
currentLong = simulateLong();//vehiState.getLongitude();
currentHeading = simulateHeading();//compState.getHeading();
```

4. Keep the log box logging the desiredHeading, currentHeading, and yawFin. This will allow the instructor and students to debug what is happening. Other things can be logged as well. That is, keep the line of code:

```
AddToLogBox(" Desired Point Index = " + desPointIndex + ", desiredHeading = "
+desiredHeading + ", Current Heading = "+ currentHeading+", yawFin = "+yawFin);
```

5. You can add most of your code (setting the desiredHeading based on Long/Lat, as well as your P control code after the statement "`// Add your code here to set yawFin!`". Don't forget to set your yawFin!
6. If all is working, when you run the code, the currentHeading should be steered toward the desired destinations.

2.4 Deliverables

Demonstrate your simulator to the instructor on Monday, October 3rd. Remember, it must use the ACTUAL latitude/longitude from the GPS receiver.