Advance Software Engineering
Assignment 3
5/21/2017

PID Controller

Marco Karier

Approach

The tests had to consist of a different combination of speeds, angles, origin points and whether or not the agent is a instantaneous deflection type or a non-instantaneous deflection type. I set up an array with the different combinations and used for loops to make sure that the different permutations happened. The file output is named first by speed, then angle, true/false(relative to the deflection), and finally the origin points. This way when the files are organized they will be grouped together based off of similar characteristics.

To create the actual PID Controller I wanted to check to make sure that the agent is with in a certain range and if they the agent was close enough I want my PID to do nothing and only regulate it if it is out of range. So I created several "if" statements in ascending order to check how far away the agent is from 0 and used set Pitch Delta Target. I use the built agent function isProximate(double x, double y , double distance); which returns the boolean about the distance.

Outcome

From initial glance at the graphs they all seem to make the same shape when the deflection is set to false. The only thing to note would be the range changes and the frequency changes with speed and angle which is what was expected.

The average is the absolute value of the y value of the agent to represented the height to show how far away from the y = 0 the agent is flying. The program prints out the different averages of the different tests. However the average of the averages is about 46.1278 with rounding after the 4th place value.

The standard deviation is about 56.1618 with rounding after the 4th place. The standard deviation is so high because some tests have smaller y values and others have extremely high y values. This wasn't the best way to get the average but the easiest the so many tests begin run.

The step count to achieve stability(or unrecoverable loss of control) was not reached in any of the tests. The test ended after 5000 iterations and the way for the step counter to show how many steps that the agent had to have a y axis of zero which was not reached. This was the way the test were run for lack of a better method.

The distance that is printed out to the screen it taken by subtracting the initial x value form the absolute value of the final x value. The average of the distances is 4396.2874 with rounding to the 4th place value. That is a very accurate measure because some distances are negative because the agent went backwards and others are small and have only gone 10 spaces.

The area under the curve was calculated using the Runge-Kutta method. It is a algorithm used for finding a basic integral. The area under the curve for each graph is printed out to the screen when the program runs. The average of the areas under the curve is about -8.6370 with rounding at the 4th place. Again not a very accurate account because of the difference in the flight patterns in the tests.