To: Dr. Berry
From: Team Millhouse
Date: 4-10-2009
Subject: ~~Lab4 Memo~~ PID Control Lab 4

Great job
10/10

Statement of purpose:

The purpose of Lab 4 was to use feedback control to improve the Traxster's odometry. To help accomplish this goal the pmc function was supplied which provided PID control of the motors. Additionally, we needed to try and improve the square motion experiment using the pmc function and PID concepts. Lastly, we needed to use these concepts to improve wall following.

Strategy:

Our strategy for part one was to improve the square path code from lab2 by utilizing the PID control obtained from the pmc function. For this part of the lab we simply copied the square function over to the supplied lab4 program and replaced the dmc command with pmc commands. In order to make the improvements to the wall following function, we decided to start with the lab 2 code. We created a copy of the lab2 code that included our wall following routine, and then added the pmc functionality to the program. Finally, we changed the drive commands in the wall following function to utilize PD control.

Methods:

Overall for the square routine we used the pmc function set at a specified distance of 12inches to make the robot drive the sides of the square and then used a call to pmc to get the Traxster to rotate 90degrees. The only other modification to the original code was the removal of the thread.sleep calls. After we had completed the modified square function for the Traxster, we than began to tweak the gains of the pmc function in order to get the Traxster to drive the square. The final proportional, derivative, and integral constants used were 1, 0 and, 2 respectively along with a motor speed of 55. Finally, we then conducted trials and measured the distance between the starting points to the ending points of the square path.

To improve the wall following routine we decided to not use the pmc function and to replace the set speed values that were sent in the dmc.drive calls to be based on a PD function. The PD function that we created used the distance read from the ultrasonic sensors to determine the speed that was passed to the dmc.drive function. We then used our same state code created for lab2 and put in the PD controlled speed function. Additionally, we removed any calls to thread.sleep, and created another state that allowed the Traxster to keep from hitting walls by detected when it was too close. The PD functions used are shown below in figure 1, the functions used for the left motor and right motor only differed by a sign. The goal variable listed in the equation is set to the distance the robot should stay form the wall, the Kp and Kd variables represent the constants used for proportional and derivative control respectively. Lastly, we tweaked the goal, Kp, and Kd values until we were able to get the robot to correctly follow the wall. The final values used for were 0.1 for Kp, 10 for Kd, and 3inches for the goal.

```
Right Motor = (50 - Kp * (r - goal) + Kd * (r - pr)
Left Motor = (50 - Kp * (r - goal) - Kd * (r - pr)
```

Figuer 1:PD left and right motor equations

Results:

Our results from the odometry test were poor when compared to the capability of a machine with a control algorithm. Table 1 shows our results and the average calculated odometry error. Our results using the pmc function were much better. Table 2 shows the results and an average error of only 1.38inches. As for the wall following algorithm, we were able to follow a wall around corners and obstacles will maintaining a specified distance away from the wall.

Trial	Distance(inches)
1	3.375
2	4.25
3	7.25
4	1.75
Average:	4.156

Table 1: Ending measurements

Trial	Distance(inches)
1	2.00
2	1.25
3	0.75
4	1.50
Average:	1.375

Table 2: Ending measurements

Questions:

What controller gains did you get for the P, PI, PD, PID controller when you tuned it in Part 1?

We found the appropriate P, I, and D gains to be 1, 0 and 2 respectively.

What were the results of the different gains, how did you decide on which one to use?

We noticed that the proportional controller had a lot to do with getting the robot to respond to the right speed and distance, while the derivative control was too sensitive to effectively adjust. It had to either be 1 or 0. We also found that the integral controller to have small effect until it was larger than the proportional controller gain.

Compare and contrast the robot's accuracy for the 12 inch distance based upon the controller used (P, PI, PD, PID).

By adding a PID controller to the robot we were able to reduce the amount of error from its motion; however the time required building a PID controller and fine tune it was intensive and fairly difficult.

How does the odometry error for the square motion compare to the error from Lab 1?

The overall error was greatly reduced by using PID control in lieu of dead reckoning. A reduced error of approximately three times was accomplished by adding PID control.

What plan did you use to implement the feedback controller to improve your wall following and line following behaviors?

We used PD control to accomplish feedback control for wall following. Proportional control was accomplished by assigning a constant that multiplied by the error from the wall and a goal target distance, and converted this to motor speed. Similarly, we accomplished derivative control by adjusting the motor speed based upon previous IR distance measurements.

Compare the results of the improved wall following and line following to the implementations without feedback control.

The resulting improvements produced a smoother more reliable line following robot. It was capable of following the wall a goal distance and do so smoothly. Instead of turning towards the wall ever two or three seconds, it made smart adjustments based upon actual measurements from the wall.

What was the general plan you used to implement the feedback control on the robot?

Please see the question above about PD control.

How could you improve the line following and wall following algorithms that you developed in this lab?

We could have added Integral Control in order to improve the wall following robot. In addition other control architectures may have been used to make faster and more accurate decisions.

Conclusion:

In conclusion, our algorithms were much improved from the previous versions. The square path function was much more repeatable and considerably more accurate. The wall following routine was able to follow the wall while not making contact with it. Future improvements for these algorithms include integral control and some sort of learning or improvement over time so that the system would tweak the constants depending on the environment.