CSCE 374 Robotics

Nate Fuller, Silas Rubinson, Matt Farich (Team 2)

Project 3 Report

24 October 2013

The purpose of this project was to have the robot perform *wall following* behavior by using the PID controller.

We decided to reuse our library of code from the first project and add to it. Some of the code that is part of “functionsForRobot.c” came from the iRobot website. We included the necessary libraries and our “functionsForRobot.h,” which lays out our functions we will use, defines that some of the random numbers are replaced by variables that say what the numbers are used for, op codes, bit masks, and LED function numbers. The primary data type we used were integers in the “CSCE374P3a.c” code. The while loop runs forever, that way the robot will constantly read the infrared sensor data. We correctly read the value from this sensor and assembled into a 16-bit integer.

Description:

Continuing on from the above, we arrived at our value for the set point and gains by means of trial and error. We consistently tested and retested the code and functionality of the robot to make sure the robot was driving as parallel to the wall as possible. During this trial and error process, some of the conflicts we had were overcoming scratches on the wall and other non-homogeneous faults in the wall. We also made sure that if the robot came close enough to a wall, the sensor would pick it up and run along it. When the robot bumps into a wall it turns left until either it stops hitting the wall or the wall sensor picks up the wall signal. To deal with right turns in the wall we are following we let the PID controller find the wall. If we have a left turn we hit the wall and turn left until the wall sensor picks up the wall.

Evaluation:

Our program works, but we believe there may be a chink in our armor: the robot hits the wall at the entrance of the lab because the surface scatters the infrared signal so the robot thinks that the wall is missing. We would recommend a PID control to someone else because the code is not that difficult to implement and when the gains are tuned it follows walls smoothly. The difficulty is finding the gains that work for your robot and environment.

Allocation of Effort:

Silas and Matt worked on the coding and debugging of the robot functionality and Nate wrote up the report. Silas and Matt showed Nate how the robot works and the unobvious parts of code that were done while Nate was unable to meet when we made the robot work. Nate was there the first time to help, but wasn’t the second, so needed to understand what we had done.