DPM Lab 1: Wall Following

Group No. 36
Student Names: Saima Haneef (260744058)
Yubei (Rebecca) Xiong (260711605)
Date of Submission: January 19, 2017

Data Analysis:

After many attempts and trials, the wall follower robot was able to travel continuously at a distance bandCentre away from the wall without colliding into the blocks or drifting too far. This was achieved by positioning the ultrasonic sensor at a suitable angle, and adding the filter, as well as adjusting the bandCentre for the most ideal performance.

It was expected that the wall follower robot would repeatedly oscillate from one side of the band to the other with the bang-bang and P-type controllers because the robot adjusted its motor behaviour according to the ultrasonic responses it was receiving. The robot should oscillate so that when it got too close to the wall, it drove itself away from it to avoid crashing into the blocks. On the other hand, when it got too far away from the wall, it tried to approach the blocks so that it could stay at the same distance of the bandCentre from the wall. This kind of oscillation is achieved by varying the speed of the motors in two different ways. In bang-bang mode, the robot switched from one form to another with a set speed assigned to each motor. For a p-type controller, the robot adjusted its motor speeds proportional to the error, which was the difference between its current distance from the wall and the bandCentre. Therefore, it was expected that the robot would oscillate from one side of the band to the other for both these controllers.

Observations and Conclusions:

There were a few errors that the ultrasonic sensor in the wall follower robot experienced. One error produced was that when the robot got too close to the wall, the robot gave out an error signal that exceeded its normal range in distance. It would produce a very large value, which was clearly incorrect. Another error was that, whenever the robot passed a gap, the ultrasonic sensor would suddenly detect a mounting distance, which caused it to turn towards the gap as opposed to ignoring it. In order to solve these problems, a filter (BangBangController.java line 31) was added to the code, which allowed the robot to ignore the responses that passed the normal distance threshold for a short period of time. These errors were thus filterable.

The ultrasonic sensor did not produce false positives, but at one point, it produced false negatives. It failed to detect certain objects and would crash into them. This problem was solved by adjusting the bandCentre to allow it to follow a path far enough from the wall so that the robot did not get too close to the wall.

Further Improvements:

Some improvements that could have been made to improve the physical and software designs would be, first, to decrease the distance between the two wheels in the robot. This improvement would make it easier for the robot to turn around corners which would make the robot more efficient. Another improvement would have been to use tires with better traction, especially as the tires used in the lab eventually lost some traction due to dust. Using tires with better traction would allow the robot to go faster and be more effective in completing the path it had to follow. A final improvement that could have been made would have been to add an extra piece of code that would allow the robot to reverse whenever it got too close to the wall, and thus continue its course without scraping or crashing into the wall.

A controller type that may have better outcomes than bang-bang and p-type would be a PID controller (Avery). A PID controller uses derivatives and integrals to assess the errors produced in certain situations and the controller reacts accordingly (Avery). This type of controller is much better than a bang-bang and p-type controller because it keeps track of its movement and adjusts its error based the accumulative data with more accurate calculations.

Works Cited:

Avery, Paul. "Introduction to PID Control." Machine Design | Home, 1 Mar. 2009, machinedesign.com/sensors/introduction-pid-control. Accessed 16 Jan. 2017