Anza Khan
Ryan Chalmers 260581055
Lab1

Data Analysis:
The BangBang controller did keep the robot at a distance 'bandCentre' from the wall, this value for us was 38. The robot repeatedly oscillates with the BangBang and PControllers because both controllers make changes when the robot reaches the maximum allowable deviation from 'bandCentre'. These changes are not intended to put the robot at bandCentre, rather, they are intended to ensure the robot does not exceed the allowed deviation from the bandCentre that is set by the bandwidth.

Observations and Conclusions:
Our earlier implementations of the PController would face many issues of incorrect filtering i.e. our robot would pick the maximum value and instead of it filtering it out, it would speed up in the opposite direction. The maximum value was 2147483647. This was heavily influenced by the ultrasonic sensor detecting false negatives.

Another issue that we had to deal with throughout the course of Lab1 was the ultrasonic sensor itself. Since the sensor used was not of the highest quality there was a limit to the number of times it could ping per second i.e. it had a limited frequency. This caused problems; most noticeable when the error was small in the BangBang controller. The sensor was unable to read the distance quickly enough so when the robot entered the threshold region, the sensor could not indicate in sufficient time that there is no error and that both motor speeds should be equal to make the robot go straight. The sensor occasionally reported objects (such as chairs in the lab room) as much closer than they actually were. This erroneous reporting tended to persist so long as the robot was facing the same object, so it wasn't easily filterable. Additionally the ultrasonic sensor was producing a large number of incorrect values, especially after rounding a corner. We observed that as the ultrasonic sensor once again found the wall in range, it would produce values that were much too high, forcing the robot to continue the turn until a collision occurred.

Further Improvements:
The first improvement that we felt could be done is to lower the height of the ultrasonic sensor mount. Our design had the US sensor mounted near the upper threshold height of the wall. It is possible that some if the incorrect readings from the US sensor could have been a result of this. Another hardware improvement would have been to improve the stability of our robot, this we could have done by adding a heavier counterweight at the back which would provide sufficient support to the front two wheels. For our robot we used a simple ball bearing at the back which works as a decent solution but if we made another wheel, it would have increased the stability further as it would have been able to counteract the weight of the front wheels. While we were demo-ing in front of the TA, our ultrasonic sensor was placed vertically as opposed to horizontally, this caused the sensor face to at times miss certain corners and only detect them at the last possible moment, a solution to this would be to place the sensor horizontally.

Another major improvement for our BangBang controller would have been to add additional code lines in it to allow us to rotate the motor that held our sensor, this would have allowed the sensor to constantly rotate and detect obstacles in its path quickly and move much smoother as opposed to a stationary position. We found that it was possible to maneuver the course without this, although it was tough to find the ideal angle for which the sensor could detect all obstacles quick enough for a response. Sticking to software based improvements another improvement would have been to use doubles as opposed to integer values in our code, that would have given us a greater range of values to play with and hence improved accuracy.

We could have used better controllers. Proportional + Integral + Derivative (PID) controller may have better outcomes than the bang-bang and P-type controllers. This is because the PID controller combines the proportional, integral and derivative controllers which allow it to have more flexibility and accuracy. Derivative controllers are implemented to account for future values by taking the derivative, while the integral controllers are used to account for past performance and correct it based on past errors. Therefore, for these reasons, by combining the three controllers to form a PID controller may have better outcomes than just the bang-bang and P-type controllers alone.