ECSE 211
Fall 2019

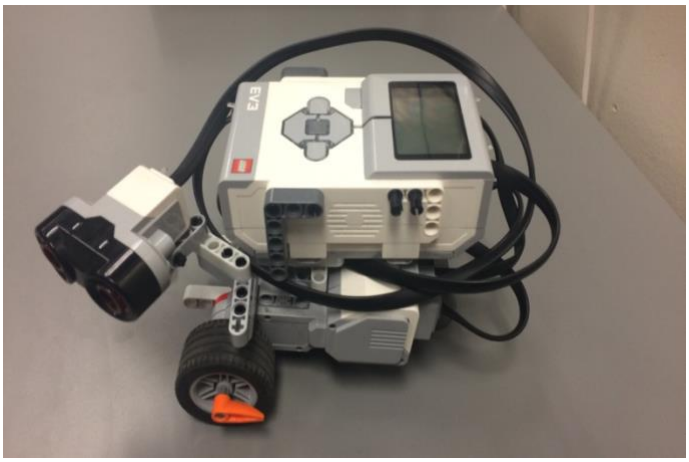
Lab 1 Report:
Wall Following

Group 21
Dafne Culha - 260785524
Suhas Udupa - 260867826

Culha Dafne 260785524
Udupa Suhas 260867826

Section 1: Design Evaluation

For this laboratory, we were required to develop a robot using the EV3 Mindstorm Kit which had the ability to follow a wall using the ultrasonic sensor and 2 motors.



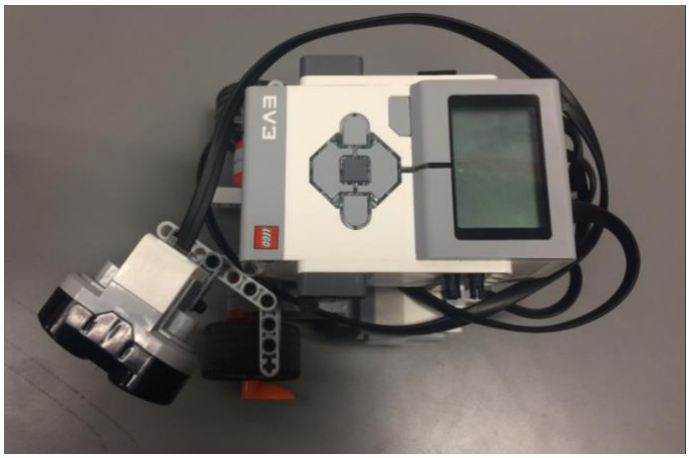


Figure 1: Our Hardware Design (Left and Top View)

The first step we took to develop our robot was to design our hardware. At first, we designed a very narrow robot, with the main and heavy part of the robot placed on the top of the ball at the back and with our sensor placed on the left side in the front at a 45-degree angle. Placing the sensor at this angle allowed the robot to follow the wall along its left side and also gave the robot the straight-ahead vision, which helped the robot anticipate concave turns. This initial design turned out to be highly unstable because the wheels were too narrow and lots of friction was caused by the ball at the back due to the heavy and uneven weight distribution. Thus, in our second design idea, we decided to flip the brick and spread the weight distribution more evenly and placed more weight on the wheels so that the latter would grip more firmly onto the ground. We also made the structure wider to further stabilize the robot while maintaining a compact figure so that the robot was still able to perform sharp turns.

The logic used to implement the bang-bang controller is shown in Figure 2 below.

Culha Dafne 260785524
Udupa Suhas 260867826

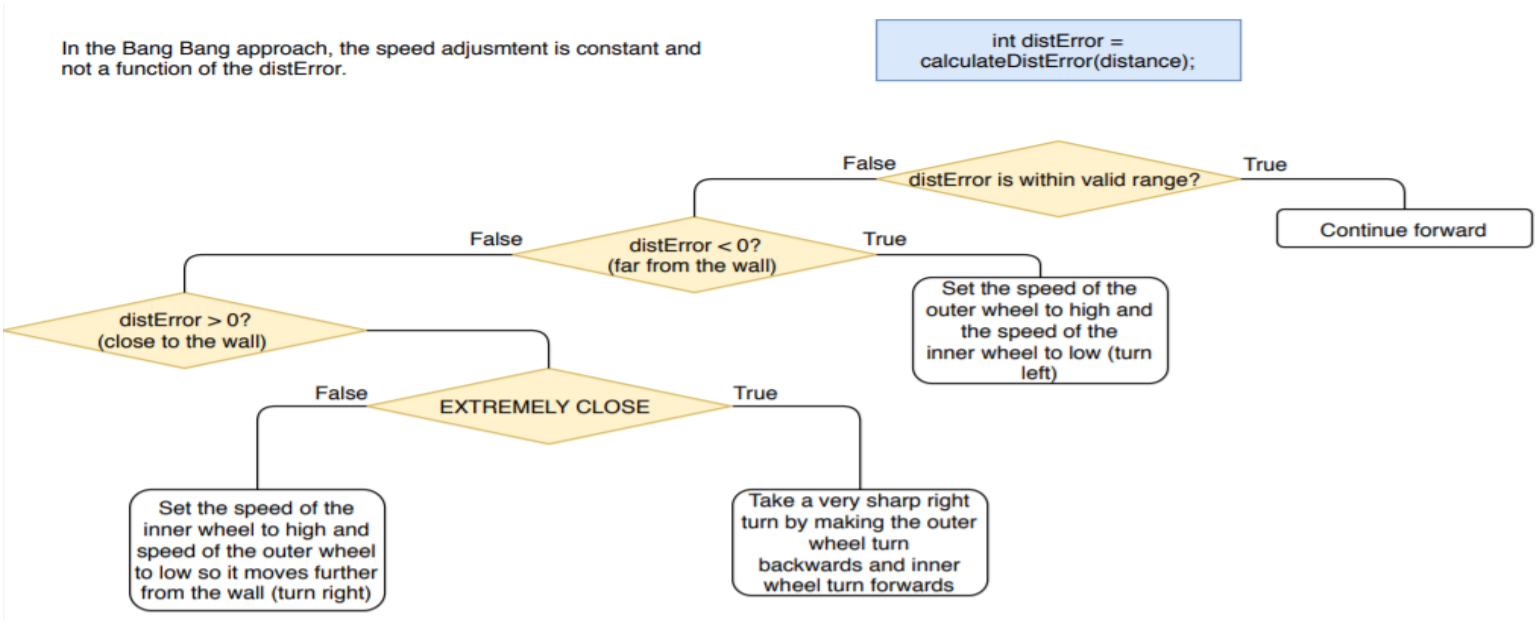


Figure 2: Our Bang-Bang Controller Implementation Logic

The software used to implement the p-type controller can be seen below in Figure 3:

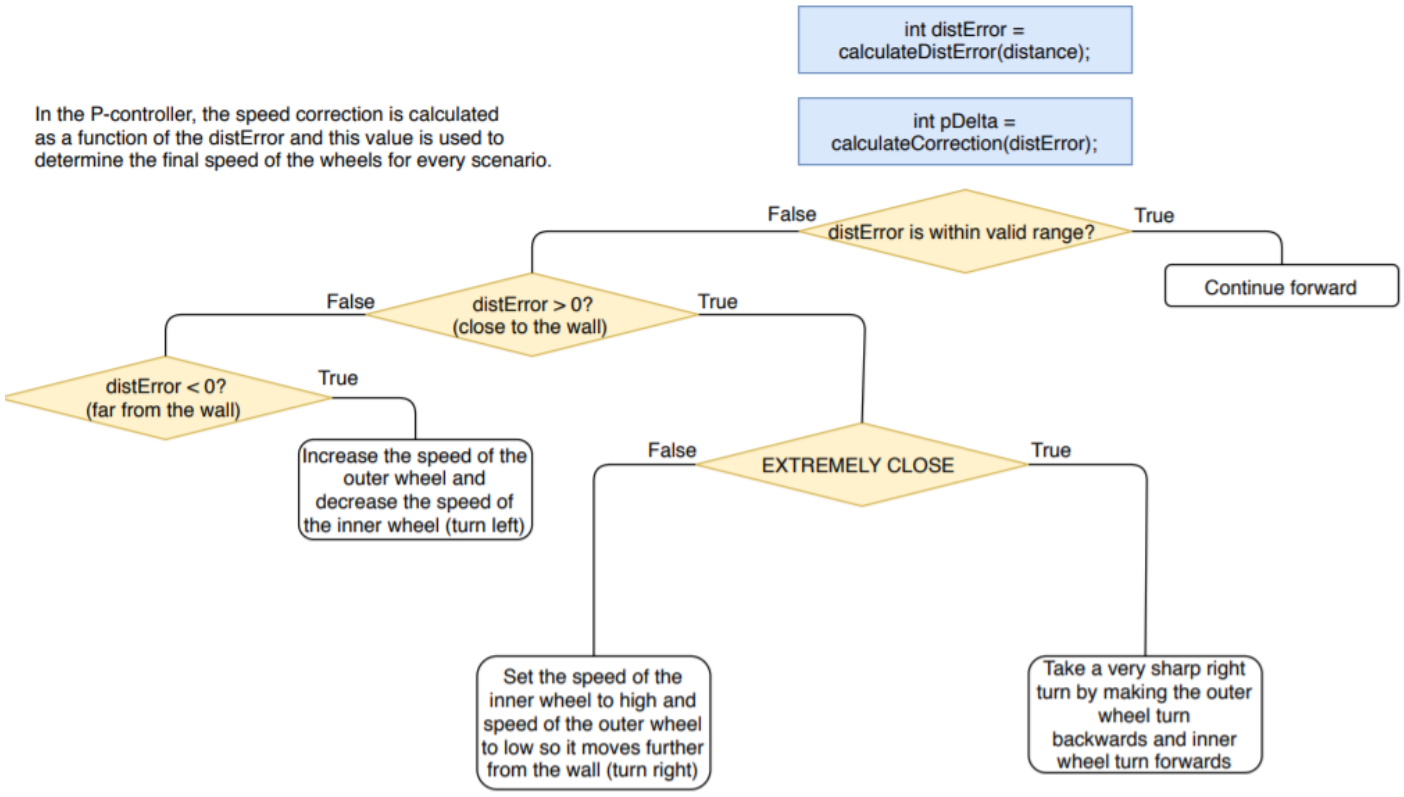


Figure 3: Our P-Controller implementation logic

Culha Dafne 260785524
Udupa Suhas 260867826

After testing several different values for the constants and recording how our robot responded, we were able to optimize them for each controller and our robot was able to follow a wall with gaps, convex and concave turns for each implementation.

For our bang-bang implementation, we set the motor speeds based on how wide or narrow we wanted the turns to be. We set the final band center to 27 and bandwidth to 3. Using these values, we found that the robot would adjust itself when it would get too close to the wall and would be able to take turns very smoothly without crashing into the walls.

For the p-type implementation, we also set the band center to 27 and bandwidth to 3. We used a p-constant of 5 to calculate the speed correction when the robot wasn't within the specified bounds. We also set a maximum correction speed of 80 so that when the robot was too far from the wall, very high values of speed correction would be avoided. Using these values, the robot maintained a band center distance from the wall with minimal oscillation and accurately followed the given structure.

Section 2: Test Data

Testing the P-type controller constant

Proportional Constant	**Battery Voltage**	**Successful Laps for an L-Shaped Wall**	**Behaviour**
2	8.0	0	Robot hits the wall
5 (used in our design)	8.0	2	Success
8	8.0	0	Robot hits the wall

Figure 4: Effects of Different P-type Controller Constants

As shown in Figure 2.1, using different P-type controller constants under the same testing environment rendered different results. In our design, a P-constant of 5 was determined to be optimal. When this value was increased to 8, significant oscillation was observed at every occurrence the distance would vary. This is because when the proportionality constant is

Culha Dafne 260785524
Udupa Suhas 260867826

increased, the sensitivity of the controller to error is also increased, which effectively causes an impairing in stability. Hence, the system approaches the behavior of on-off controlled systems and its response becomes oscillatory. As a matter of fact, before turning a convex corner, the robot would follow the wall correctly at a constant speed, but the instance when the ultrasonic sensor would no longer have the wall in its angle of vision, it would take a very sharp left turn at a high speed, thus causing the robot to crash into the wall. Furthermore, since the overall speed of the robot was increased, the resulting change in direction also became larger and this made it hard for the robot to maintain the band center which caused significant oscillation.

When the P-constant was lowered to 2, there was significantly less oscillation and the behavior of the robot was much smoother as it would take a much longer period of time to execute its maneuvers. This also allowed the robot to better maintain the band center as the correction was small and slow and the changes in direction weren't as drastic. However, using a lower P-constant still caused the robot to crash into the wall because of its inability to execute the turns in time. When using a P-constant of 2, the robot failed at a concave turn. It would turn too close and hit the wall.
Finally, when the robot was subject to a straight-line trajectory, both the lower and higher P-constant values didn't make a difference in the robot's behavior as it followed the wall properly. The trajectory was relatively smooth compared to the bang-bang controller and the robot did not oscillate too much along the band center. It was very similar to what we observed when using our tuned constant.

Bang-Bang controller test

Trial Number	**Battery Voltage**	**Change from Previous Trial**	**Successful Laps**	**Observations**

Culha Dafne 260785524
Udupa Suhas 260867826

1	7.1	Increased the given band center and bandwidth.	0	The robot followed the straight wall without any significant oscillations (although it would slowly creep closer to the wall and when it got too close, the robot would suddenly turn right to oscillate back to the band center), but it failed on the first convex turn. The sensor didn't seem to detect the left turn and did a very wide turn, which caused the sensor to then lose track of the wall. The robot also cut concave corners instead of approaching the forward wall and making a sharper turn.
2	8.0	Charged the robot to full capacity in order to ensure its proper functionality. Reduced the band center, bandwidth and motor speeds to accommodate our hardware design. Updated the logic structure in the code.	0	The robot failed to complete a full lap. It executed all concave turns smoothly and would oscillate after (successfully) completing a convex corner to readjust its position from the wall with respect to the band center), but it failed to detect a gap in the wall following a convex turn. By the time the sensor detected the hole, it had already turned into the wall.

Culha Dafne 260785524
Udupa Suhas 260867826

3	8.0	Slightly increased the band center. Lowered the speed of the wheels on very tight turns and changed the differential of speed between the inner/outer wheels on regular turns.	2	The robot performed the circuit smoothly and with minimal oscillations (which is expected from a Bang-Bang controller). It maintained an overall constant band center throughout the entire trajectory when placed initially at a distance equal to the band center.

Figure 5: Bang-Bang Controller Trials Data

As seen in Figure 5, we had to bring several changes to our code in order to implement the bang-bang controller successfully. Although there was no significant oscillation per say throughout the robot's trajectory, it was still present in the robot's behavior. This is however expected when using the bang-bang control as it is an optimal approach for reaching the desired set point (i.e. band center) but it is not very good for maintaining it. This is evident when the robot is following a straight wall. As mentioned in Figure 5, the robot would tend to slowly lean left on the wall but would then suddenly shift right to return to a distance from the wall equal to the band center. Hence, although the bang-bang controller is very simple, this control strategy has serious drawbacks. The robot never settles down to a perfectly steady state as it oscillates constantly between its two extreme states. As explained in the table above, the robot failed twice at a convex turn. The robot would take a wide turn and would find itself finishing the turn too close to the wall as the sensor would fail to read the distance adequately. In order to rectify this issue, we reduced the band center and motor speeds, which allowed the robot to run much smoother (especially at convex turns). However, based on the result from trial 2, we determined it was necessary to slightly increase the band center as the robot would turn into the wall following a convex turn when a gap was present. Lowering the speed of the wheels on very tight turns also helped solve this corner case.

P-type controller test

Trial Number	**Battery Voltage**	**Change from Previous Trial**	**Successful Laps**	**Observations**

Culha Dafne 260785524
Udupa Suhas 260867826

1	7.9	Used an arbitrary P-constant value (7).	0	The robot worked fine along straight walls and concave turns, and it did so in a fairly smooth manner and without any significant oscillation. But it failed at convex turns, as it would take too sharp of a turn and then hit the wall.
2	7.9	Reduced the P-constant and adjusted the motor speeds for each specific corner case (straight-line trajectory, convex turn, concave turn, sharp right turn). Increased the band center.	2	The robot completed the circuit with minimal oscillation. However, the robot failed to adequately follow the shape of the circuit and did more of a circular lap around the wall instead. The robot effectively didn't maintain the proper band center distance from the wall (for instance, at convex turns, the robot would barely approach the wall and begin turning right from too far, thus neglecting to accurately contour the wall).
3	7.8	Slightly increased the P-constant.	2	The robot executed the circuit successfully. Its trajectory was very smooth and showed very minimal oscillation. The robot performed all corners accurately.

Figure 6: P-Type Controller Trials Data

Culha Dafne 260785524
Udupa Suhas 260867826

As shown in Figure 6, we had to complete multiple trials in order to arrive at the optimal constants to use for the P-type controller. This control strategy is effectively more stable than the bang-bang controller and this is evident in the robot's reduced oscillation behavior. The logic behind the proportional control lies in the fact that the magnitude of the corrective action depends on the magnitude of the error. Consequently, while a small error would lead to a corresponding small adjustment, a larger error would result in a greater corrective action. By using different values for this constant, it was possible to increase the performance and reliability of the robot around the whole circuit.

In our first trial run, the robot failed at convex corners. The robot would take a very sharp turn around the corner and effectively crashed into the wall. We realized this was because we used a value that was too high for the P-constant, which was indeed impairing the controller's sensitivity. Hence, we made a few adjustments in the subsequent runs to address this issue. After reducing the proportionality constant among other changes, we noticed the robot completed the circuit without hitting the wall, but it failed to accurately follow the shape of the circuit, as explained in Figure 6. This was due to having used a value too small for the constant. Finally, we were able to rectify this issue in trial run 3 by slightly increasing the P-constant, which allowed the robot to perform the circuit successfully.

Section 3: Test Analysis

As explained in in the test data, using a different P-type constant than the one used in our demo rendered different robot trajectory oscillations and allowed us to visualize the effect of the proportionality constant. As a matter of fact, the P-type constant is one of the most sensitive constants in this laboratory. When this value was increased, we noticed the sensitivity of the controller to error also increased. But this impaired the controller's stability, which effectively resulted in the robot taking very sharp right turns at high speed at concave corners and consequently crashing into the wall. As the overall speed of the robot was increased, the resulting change in direction also became larger and this made it hard for the robot to maintain the band center (less reaction time) which caused significant oscillation. On the other hand, when the proportionality constant was reduced, there was significantly less oscillation and the robot better maintained the band center as the applied correction was smaller. However, this didn't prevent the robot from hitting the wall at a concave turn; because of the robot's inability to correct itself fast enough, it still crashed into to the wall on the turn, as mentioned in the test data. It also caused the robot to make very wide turns around the corners. As the distance separating the robot from the wall increased, the robot would lose track of its course.

To continue, the oscillation around the band center depended on the bandwidth. As recorded in the test data, there was much more oscillation associated with the bang-bang controller than with the proportional controller. This is because the bang-bang control strategy never settles down to a perfectly steady state (i.e. maintaining the band center) as it oscillates

Culha Dafne 260785524
Udupa Suhas 260867826

rapidly and constantly between its two extreme states. As a matter of fact, when the robot was following a straight wall, it would tend to lean left towards the wall and when it would get too close, the robot would suddenly reverse its wheels backwards to shift right to return to a distance from the wall equal to the band center. Hence, our oscillations around the band center were very high with the bang-bang controller. By reducing the bandwidth, we were able to slightly alleviate the robot's trajectory oscillation, but it would nonetheless still exceed the bandwidth by approximately 10 cm.

In contrast, the P-type controller had very mild oscillations and they would mainly occur in short bursts. For example, after turning a corner, the robot would sometimes oscillate repetitively for a few seconds while finding its way back to the band center and then continue with minimal oscillation. It would rarely exceed the bandwidth by more than 5 cm. Compared to the bang-bang controller which had more difficulty maintaining the band center due to its constant leftwards movement towards the wall and the lack thereof of a straight-driving state, the p-controller was effectively more stable and more consistent in staying near to the band center. This is because the correction applied to the speed of the wheels is effectively proportional to the difference between the band center and the distance from the wall.

Section 4: Observations and Conclusions

Based on your analysis, which controller would you use and why?

We would use the P-type controller because it is much more stable than the bang-bang controller, as explained in the section above, and it calculates the speed adjustment of the wheels as a function of the distance error, thus reducing significant oscillations. The speed correction in the p-type controller is much more efficient than using a constant delta to correct the speed of the wheels like in the bang-bang controller implementation.

Does the ultrasonic sensor produce false positives (detection of non-existent objects) and/or false negatives (failure to detect objects)? How frequent were they? Were they filtered?

Our sensor almost never produced false positives. However, it occasionally produced false negatives, especially when the battery was running low. They were very frequent and unfiltered. As a matter of fact, it was ignoring blocks and crashing into them. To filter out these false negatives, we required 20 readings of maximum distance before validating that there was really no blocks there. We also decreased the speed, increased the band center and the bandwidth so that the sensor would have more time to detect the objects and that the robot would adjust itself before crashing.

Culha Dafne 260785524
Udupa Suhas 260867826

Section 5: Further Improvements

We were able to produce a successful implementation (as required for this lab) for both controllers. However, our robot had its flaws.

What software improvements could you make to address the ultrasonic sensor errors? Give 3 examples.

The first suggestion would be to set bounds for the values read by the sensor, thus limiting false actions taken by the robot due to wrong and unfiltered sensor readings. Even when our brick was moving next to a block at normal distances (20-30 cm), our sensor would sometimes read very high values (20000cm) or very low values (2cm) which made our controllers (especially the p-type controller) very unstable. We could choose to set bounds and hence filter out values greater than the specified bounds.

Another improvement would be to make the sensor start before the motors. It takes our sensor about 5 seconds to start detecting distances. Thus, when the robot is placed facing a wall as a starting point, it crashes into it. To eliminate this error, it would be useful to implement a wait time for the motors to remain still before the sensor starts functioning.

Finally, defining a short period of time for which the sensor detects the distances, sorting these distances in an array and using the median value, or computing and using the mean value, would be another software improvement to address the ultrasonic sensor errors. This way, the readings would be more accurate.

What hardware improvements could you make to improve the controller performance? Give 3 examples.

The first hardware improvement would be to add a second ultrasonic sensor. We could place one ultrasonic sensor facing the wall directly to measure the lateral distance of the robot from the wall to better maintain the band center distance and then place the second sensor at the front of the robot to detect oncoming walls sooner in order to perform concave turns more efficiently.

Moreover, replacing the ball at the back of our robot with wheels would also be a very nice improvement. The robot's motion was hindered a lot in many of our trials due to the spinning and the friction caused by the use of the ball.

Lastly, placing the processor brick right on top of the wheels would really improve the stability and overall performance of our robot. This would increase the weight on the wheels

Culha Dafne 260785524
Udupa Suhas 260867826

and would allow for a more stable robot due to the increased grip of the wheels on the ground.

What other controller types could be used in place of the Bang-Bang or P-type?

A Proportional Integral Derivative Controller is another type of controller which could have been used to build a wall follower robot. It uses a specific formula that yields very accurate results and it consists of the sum of three terms: term P, term I and term D.

- Term P: The distance error multiplied by the proportionality constant;
- Term I: Integral of the error multiplied by the proportionality constant to adjust the applied correction according to the error and the number of times it was recorded
- Term D: Derivative of the error multiplied by the proportionality constant to reduce the amount of overshooting that occurs during the correction.

Summing these three components allows a deeper analysis of the error and makes the robot perform more accurately and smoothly within the accepted bounds.