ECSE 211
Lab 1: Wall Follower
Lab Report

Group 20

Linwei Yuan (260857954)

Le-Li Mao (260800098)

Section 1: Design Evaluation

Brief Description

The concept for the hardware design is trying to be as simplified as possible. Therefore, we simply have the essential four parts - main frame, front wheels, back ball, and the ultrasonic sensor. First, the ultrasonic sensor is attached in the left front of the brick and in a 45 degree using the angled part so that the robot can detect wall reflection from both the left and the front. Second, we decided to use two-wheel structure because it is more convenient to control to make turns. So, two wheels and motors are jointed with the main brick as a start. Next, we consider that a third-point support is needed to support the brick while moving. So, we have two design choices at that point: add a third wheel or other support components that can move with the front wheels. Finally, the choice of using an all-direction-moving iron ball so that the robot can smoothly turn without extra friction. However, the upcoming problem is that the brick is not horizontal due to the small size of the iron ball. We consider to solve this by adding a main frame structure at the bottom to raise the level of the back of the brick. At this point, our final frame version of robot is completed.

Two types of controller exist the software, Bang bang and P-type. Bang bang uses constant and will perform the same turn regardless of the distance away from the band center. For a given range within the band center, the robot will go straight with both wheel in the same speed. When the detected distance is less than a threshold, the robot will decrease the right wheel to a constant value to perform a right turn to get away from the wall. On the other hand, the robot will decrease the left motor speed to a constant when the detected distance is greater than the threshold to get closer to the wall.

P- type uses a P-type coefficient to calculate the speed decrease in proportional to the distance away from the band center. After it compute the delta and the speed decrease, we perform a min operation with the predefined max decrease value to ensure that the speed decrease is not too large at extreme cases. Similar to the bang bang we will decrease the corresponding speed based on the turn requirement.

Workflow

1. After setting up the environment for Lejos EV3 programming (configuring the operating system), and installing Legos EV3(reformatting the brick), the first problem we encounter was the brick cannot connect to DPM Wifi network. The problem was solved by reformatting the SD card and some reconfiguration.

2. Then, the hardware design of our robot was implemented. The robot was built according to the design concept described. (Assemble the base of the robot and attach the motor to the base. Attach the ultrasonic sensor to the left front side of the bot and in a 45 degree using the angled part.)

3. Third, we developed the bang bang controller and the p-type controller’s base code according to the class diagram. Also, separate coefficient for different controller types were added. The program is now implemented using arbitrary constant that had not been tested yet.

4. Last, we adjusted P-coefficient by continuously checking the turning speed and bandwidth with an objective to avoid touching the wall. If the bandwidth is not appropriate, P-coefficient will be changed accordingly with holding other constants consistent. Also, when continuously testing and tuning the coefficients for correct behavior, we often adjusted one constant once.

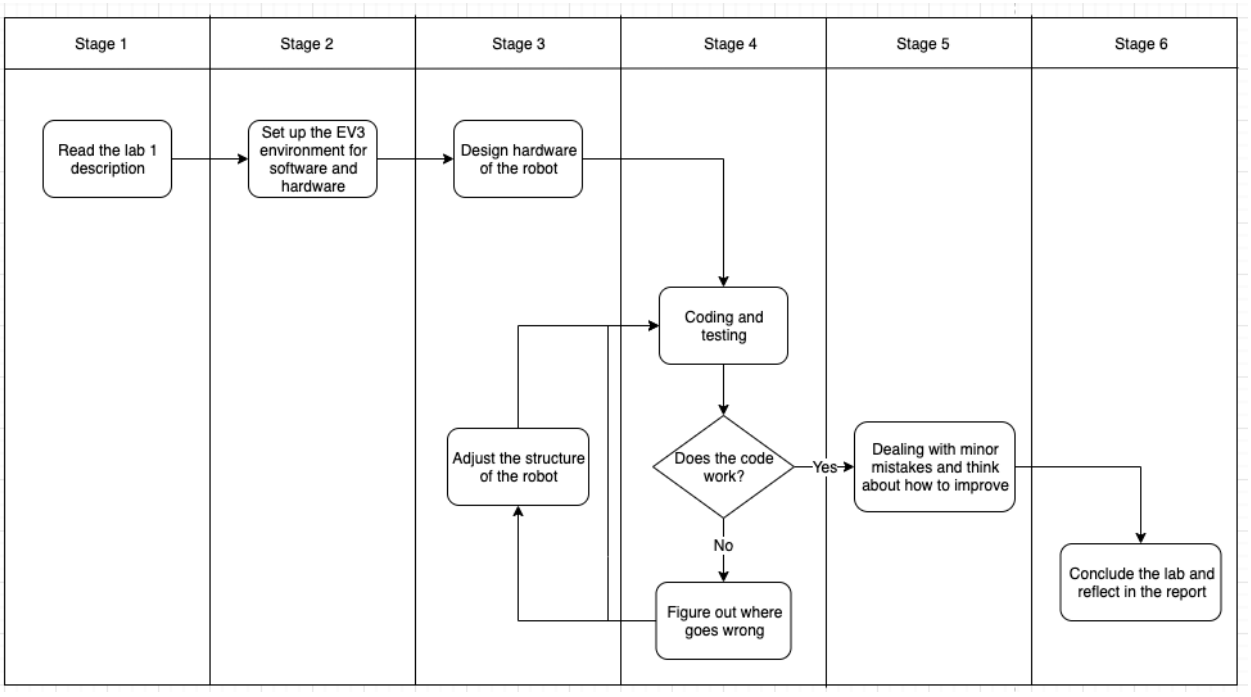

Figure 1: Workflow diagram

Photos of Hardware Design

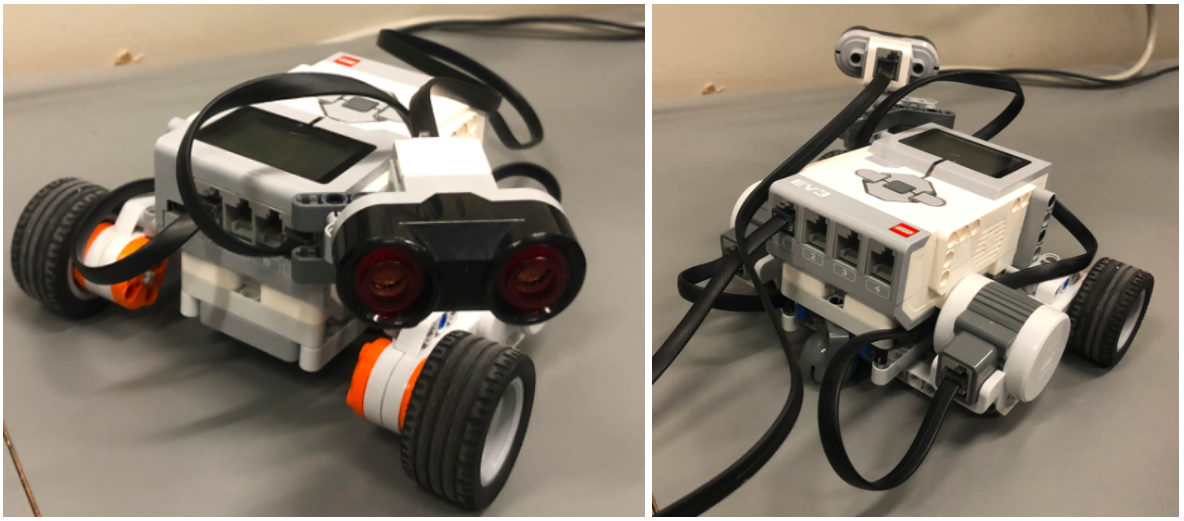

Figure 2: Front view (left) and back view (right) of the robot

Visuals for the Software Design

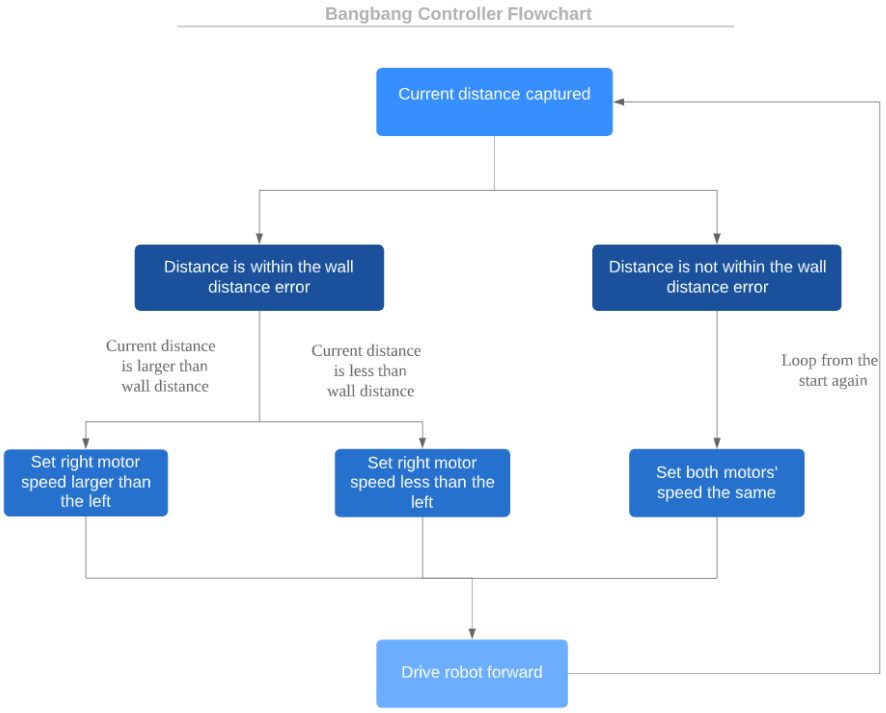

Figure 3: BangBang controller flow chart

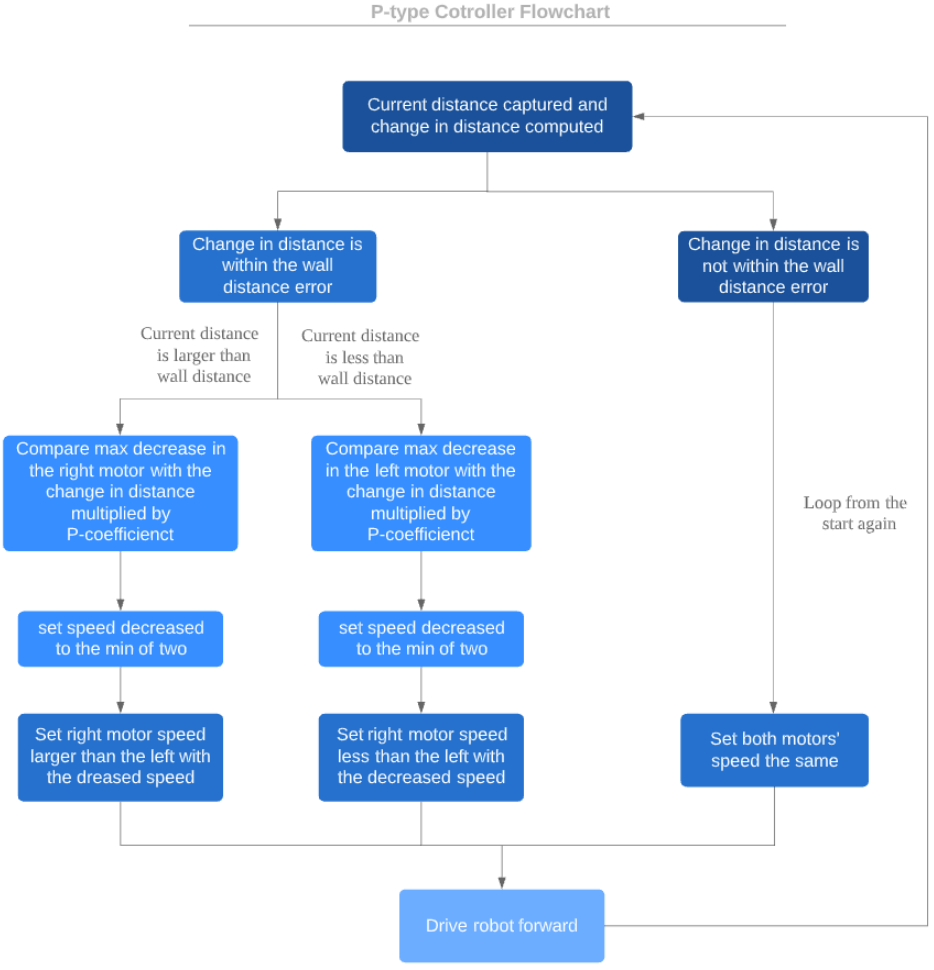

Figure 4: P-type controller flow chart

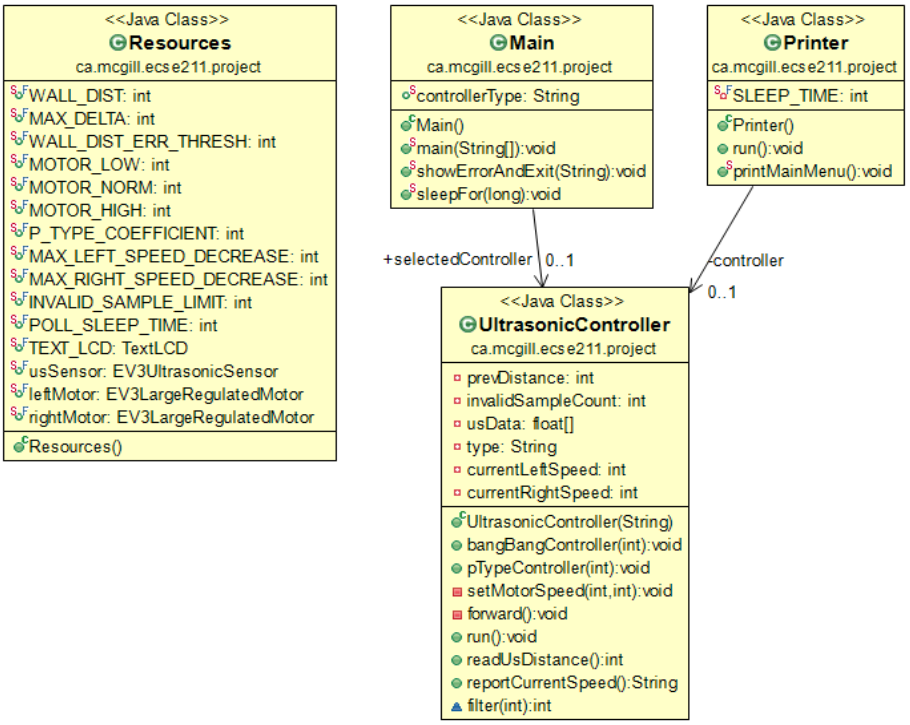

Figure 5: Class diagram of the project

Section 2: Test data

Test the P-type controller constant

P-Type Coefficient Demo uses: 19	Behavior
5	- Low amount of oscillation and less correction. - The robot is unable to react to tight turn with consistency. - U-turn reasonably well without being too close to the wall. - The robot stay on the band center reasonably well except for tight concave turn where it hit the wall.
25	- High amount of oscillation and correction. - The robot is able to react to tight turn with consistency. - In some cases, the robot would turn too quickly into the U-turn and hit the wall. - The robot is unstable on the band center.

Table 1: P-coefficient data

Bang-Bang controller test

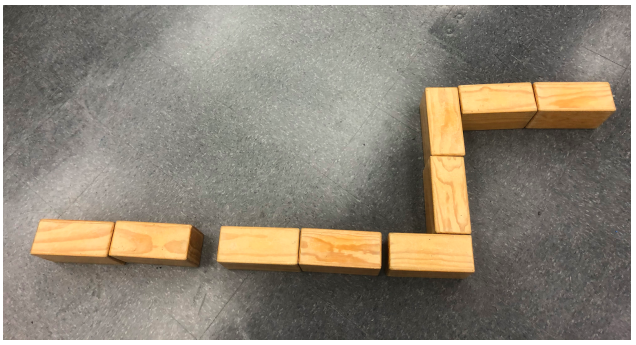

Figure 6: The track to test the robot in

	Behavioral
Trial 1	- Stayed at a safe distance except for the U-turn - Hit the U-turn due too fast turning speed - Medium amount of oscillation around the band center (<10 cm)
Trial 2	- Stayed at a safe distance throughout the laps except one turn - Almost hit the concave turn due too slow turning speed - Medium amount of oscillation around the band center (<9 cm)
Trial 3	- Stayed at a safe distance throughout the laps - Medium amount of oscillation around the band center (<7cm)
Summary	Among the three trials we did for the Bang-bang controller, 2 out of 3 turned out to finish a lap without touching the wall.

Table 2: Bang bang controller data

P-type controller test

We used the same track as above for our p-type testing. See figure 5.

	Behavioral
Trial 1	- Stayed at a safe distance throughout the laps - Almost hit the concave due too slow turning speed - Low amount of oscillation around the band center (<7 cm)
Trial 2	- Stayed at a safe distance throughout the laps - Almost hit the U-turn due to wall being in the blind spot - Low amount of oscillation around the band center (<5 cm)
Trial 3	- Stayed at a safe distance throughout the laps - Low amount of oscillation around the band center (<5 cm)
Summary	In all 3 trials, the robot successfully navigated around the lap but have minor

	danger at certain situations.

Table 2: P-type controller data

Section 3: Test Analysis

What happens when your P-type constant is different from the one used in the demo?

When we lowered the P-type constant from 15 to 5, the robot is unable to perform tight concave turn and will end up running into the wall. Also, the robot experiences less hunting behavior due to the small adjustment of the low P-type constant. When we change the P-type constant from 15 to 25, the robot moves with volatile movement and oscillate drastically in the band center and have too much hunting behavior. Also, the robot turns significantly too quickly at the U-Turn position and hit the wall at the sensor's blind spot.

How much does your robot oscillate around the band center?

For p-type controller, the robot has less oscillation and stayed on the band center reasonably well. In contrast, the bang bang controller shown a lot more hunting behavior and relatively more oscillation because it often over adjust in some scenario.

Did it ever exceed the bandwidth? If so, by how much?

In the worse case, P-type have exceeded the bandwidth by around 7cm. On the other hand the bang bang exceed the bandwidth by 10cm.

Describe how this occurs qualitatively for each controller.

The robot is able to adjust its speed accordingly as long as it detects a wall distance in the Bang-Bang Controller so that it can still maintain the bandwidth stably. In the P-type Controller, the robot always gradually moves back into bandwidth range because the speed is adjusted by a proportional response (ie. controlled by a P-coefficient). Generally speaking, the reason why the robot sometimes moves away from bandwidth is that its routine is not regulated and it can only response whenever the ultrasonic sensor detects a wall. Therefore, there exists a delay that allows the robot to exceed the bandwidth.

Section 4: Observations and Conclusions

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

Based on the analysis of the two controller, we would consider the P-type controller to have a better functionality than the Bang-Bang controller since it is observed fewer oscillations in the test. Despite the fact that the Bang-Bang controller can move back to the bandwidth quicker, the adjustment of the robot's position is often more than the position that it should be at. So, there are lots of unnecessary reorientation and movement. In contrast, the P-type controller successfully overcome this problem by gradually increase of the motor speed. However, the problem for P-type is the overdamped behavior.

As mentioned above, the P-type controller is more desirable because its mechanism speed is proportional to distance. In the lab, we did not observe extra significant oscillation behaviors for both the P-type controller and the Bang-bang controller while keeping a constant band center. However, the main difference between the two that we consider to judge the optimal choice is how stable when the robot makes U-turns: our results show that the Bang-Bang Controller has much drastic oscillations than the P-type controller does. The reason is that the robot is reorienting and adjusting its movement more frequently than usual.

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

The ultrasonic sensor has not produced false positive but sometimes produces false negative. This sometimes causes some delayed reaction to steer away from the wall and causes the robot to be really close to the wall. The false negative occurs only occasionally (1-2 times a lap). The false negative is sometimes filtered by the filter system given in the source code but in extreme cases, they still cause an impact.

Section 5: Further Improvements

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

1) Implement an algorithm that can filter the unnecessary adjustment of tuning left and right when the robot is moving along a straight wall.
2) Create a different method to execute when the robot is turning in a convex corner so the robot can minimize the errors in detection and follow the convex corner more strictly.
3) Improve the algorithm to a more complex one that involves moving backwards in order to avoid unexpected and unanticipated crashing into the wall.

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

1) Add a touch sensor to the robot so that it can move backwards when detecting collision with the wall.
2) Change the robot to a four-wheeled front-wheel-drive one to minimize the error due to inadequate mechanical grip.
3) Install a rotary sensor to have a defined rotation axis and let our robot know at which angle it should move directly prior to going forward.

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

We can use proportional-derivative (PD) and proportional-integral (PI) controller. They use more complicated algorithms than the P-type controller by adding more tuning parameters. For the P-type controller, there always exists a steady state error. To eliminate that, PI controller can be used to integrate the error over a period of time until the error value reaches zero. However, PI controller does not have the capability to predict the future behavior of error. PD controller overcomes this problem by adding a feature that anticipates future behavior of the error. Its output depends on the rate of change of error with respect to time, multiplied by derivative constant. Therefore, by combining P, PI and PD control types, the robot can use distance history to predict future events and apply proportional action to achieve a higher degree of precision.

Citation: https://www.elprocus.com/the-working-of-a-pid-controller/