ECSE 211: Design Principles and Methods

Lab 1: Wall Following

Arianit Vavla : 260868601

Cheng Chen: 260775674

Group 20

September 2019

Hardware Design

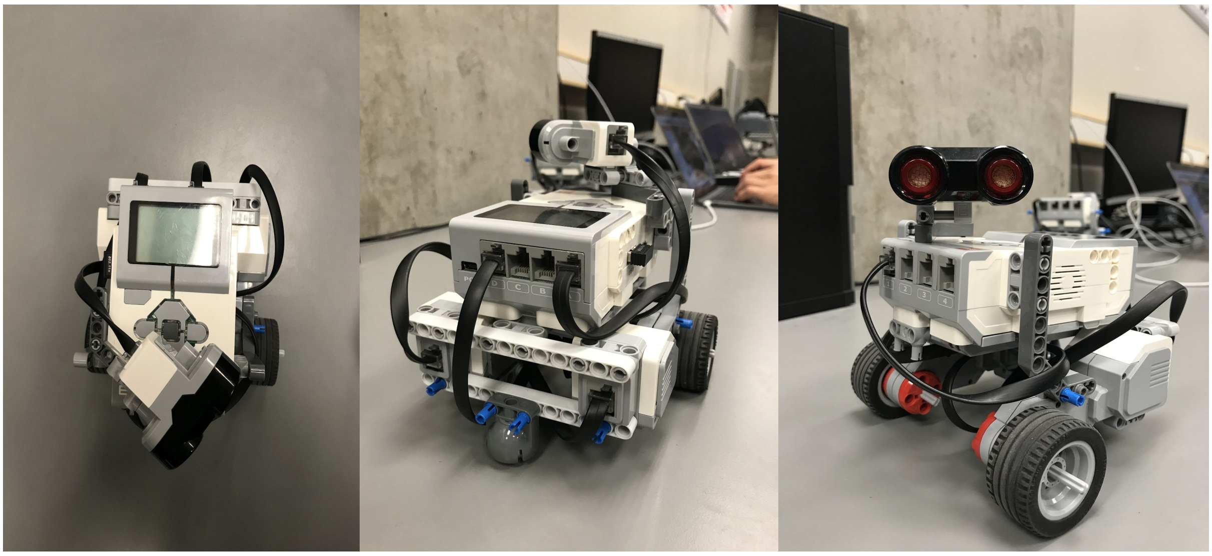

Figure 1: Car Image

First of all, we wanted to make the robot as small and as compact as possible because it will make it easier for us to design on the software side. We fixed the motor to the sides of the EV3 controller and connect the motor to two tires in the front. On the back of the robot, we used the metal ball for the robot to gain more stability and the metal ball has less friction between the floor than other materials provided. Since there is a lot of empty space between two motors, we are able to put the wires inside the robot. After a few trials, we decided to put the ultrasonic sensor on the front and top of the robot because the distance that the ultrasonic sensor measured can reflect the distance between the robot and the wall very well.

Software Design

There are 7 java files in this package, which are Main, BangBangController, PController, Printer, Resources, UltrasonicController and UltrasonicPoller. The following two flow charts will show both PController and BangBangController.

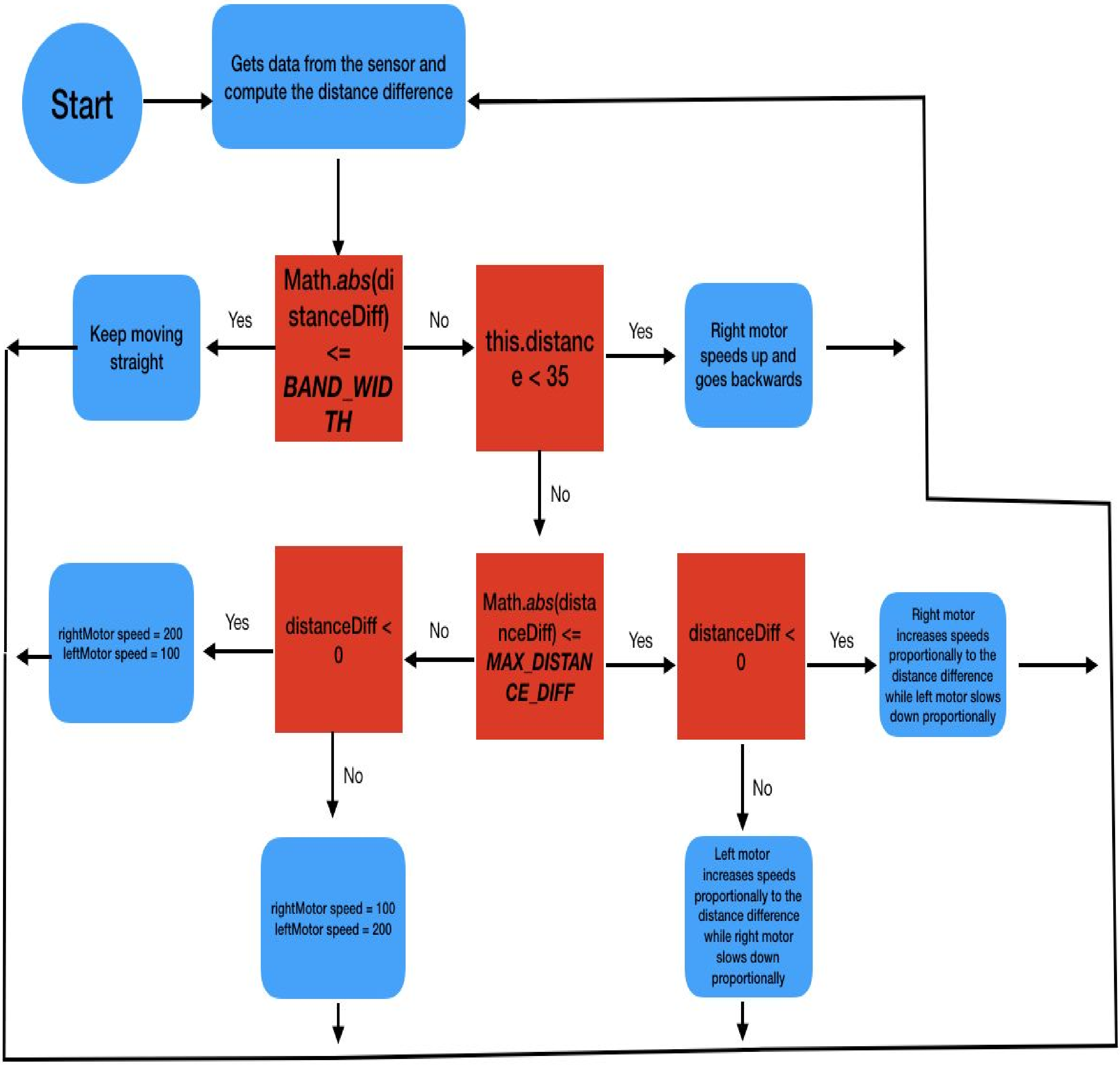

Figure 2: PController Flow Chart

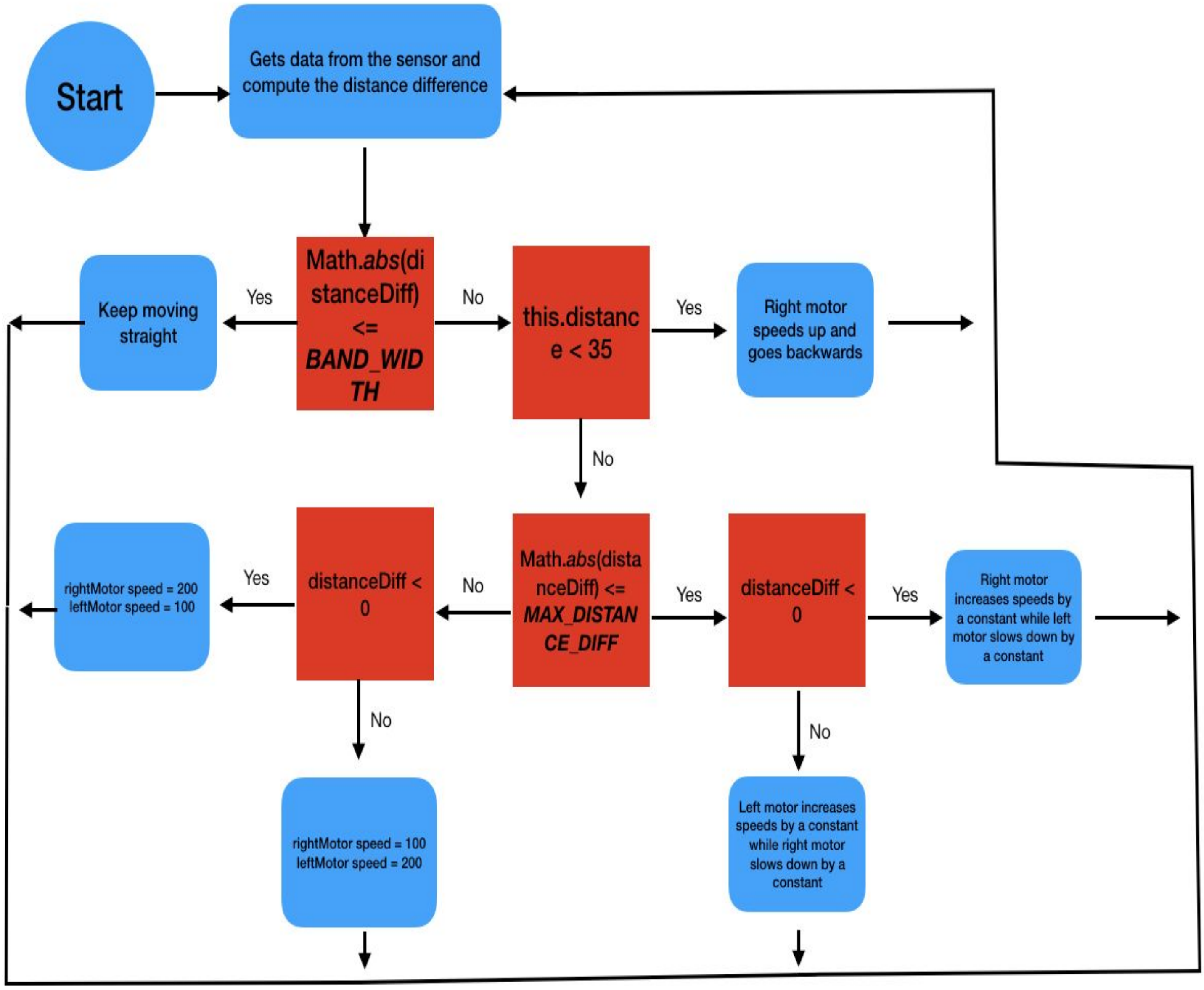

Figure 3: BangBangController

<u>Section 2: Test Data</u>

<u>Testing the P-type controller constant:</u>

On this test, we changed the constant on the P-type controller. Our initial constant was 3, so we used the constants 2 and 7. For constant 2, the machine took more time to correct its error than the initial constant because it took more time to go to its band center. So the distance was often further or closer from the wall than the band center that we set. The oscillations were almost similar as the initial constant, which did not oscillate a lot either. For constant 7, the oscillation were really frequent because the wheels had a bigger difference when correcting the error, so when the new ultrasonic sensor data was taken, there was still an error but the opposite way, so the robot went the other way to correct it, but the error was reduced slower than with the initial constant..

<u>Bang-Bang controller test:</u>

	Lap Complete	Band Center	Oscillations
Trial 1	Yes	It was constantly around the band center	It was oscillating a lot, with a lot of amplitude and it doesn’t become much smaller
Trial 2	Yes	It was constantly around the band center, but when it had to do the convex turn, it took awhile for the robot to realize it and go back following the wall	It was oscillating a lot, with a lot of amplitude and it doesn’t become much smaller
Trial 3	Yes	It was constantly around the band center	It was oscillating a lot, with a lot of amplitude and it doesn’t become much smaller

P-type controller test:

	Lap Complete	Band Center	Oscillations
Trial 1	Yes	The robot constantly stayed around the band center, including gaps, concave and convex corners	There were not many oscillations and the amplitudes were becoming smaller and smaller
Trial 2	Yes	The robot constantly stayed around the band center, including gaps, concave and convex corners	There were not many oscillations and the amplitudes were becoming smaller and smaller
Trial 3	Yes	The robot constantly stayed around the band center, including gaps, concave and convex corners	There were significantly more oscillations than the two previous trials, and we think it is maybe a problem with the sensor

Section 3: Test analysis

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

When the constant is smaller than when we did it in the demo, there were less oscillations, but the robot was more time off the band center because it took time to go back to it, but the correction of the error after each oscillation was more efficient because the robot had the time to take the data and react to it more easily as the robot was slower. When the constant was larger than the one in the demo, there were more oscillations because the robot did not have time to correct its error on time, and as a result, the error from the band center was corrected slower after each oscillation because of the reaction time, but the total time that the robot was off the band center is less than when the constant was smaller than the demo.

How much does your robot oscillate around the band center?

When the bang-bang controller was used, the robot oscillated during the whole lap and with a very large amplitude compared to the p-type controller. This is due to the fact that the difference of rotation between the left and right tire is constant in the bang-bang controller, which means that it has a harder time correcting the error from the band center as compared to the p-type controller, which the difference of rotation between the left and right tire is proportional to the error from the band center, which means if the error is small, then the robot would turn more slowly and have an easier time correcting its error.

Did it exceed the bandwidth? If so, by how much? Describe how this occurs qualitatively for each controller.

The robot exceeded the bandwidth sometimes when using the bang-bang controller by around 1-2 cm because the robot would take time to react to the data and change direction, but it did not happen often. Overall it stayed within the bandwidth. The robot stayed within the bandwidth for the p-type controller because the error was adjusted each time the robot was taking data from the ultrasonic sensor.

Section 4: Observations and Conclusions

Based on our previous analysis, P controller is much better. Based on our observations, the p-type controller had a smoother, more accurate and more esthetic motion. Also, if we think about it mathematically, it can adjust its speed according to the distance difference while the bang bang controller can only change its speed by a constant.

We saw some false positives from the EV3 control console. When the robot is too close to the wall, the ultrasonic sensor could not read the correct values. While it is moving, we saw that it read some strange values, these values might come from ultrasonic sensor from other robots as there are many other robots in the same room or the waves sent by our ultrasonic sensors got reflected. The filter method definitely filtered out some of the false positives but definitely not all of them.

Section 5: Further Improvements

Software improvements

- We could always improve the filter of the two sensors so they would both do better turns around a convex corner
- We could make a timer so each thread is better coordinated with one another and the robot would react better towards the environment
- Change the thread sleep time for the sensor so it would optimize the trajectory of the robot so it would do a path with less oscillations but also would not get out of the bandwidth as often

Hardware improvements

- We could lower the center of gravity of the robot so it would turn better and react faster
- We could change the tires because the one we had in disposition were used and would slip on the ground sometimes
- We could add a touch sensor so if ever there is something happening with the ultrasonic sensor and there would be another option for the robot to turn right in case of a concave corner

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

Another type of controller can be used is Proportional, Integral and Derivative Controller. For example, the speed change of the robot will be proportional to how fast the robot is approaching the walls (i.e. the derivative of the distance difference), which will give the robot a smoother motion.