Lab 1: Wall Follower

AHMED ELEHWANY (260707540)

BARRY CHEN

Group 10

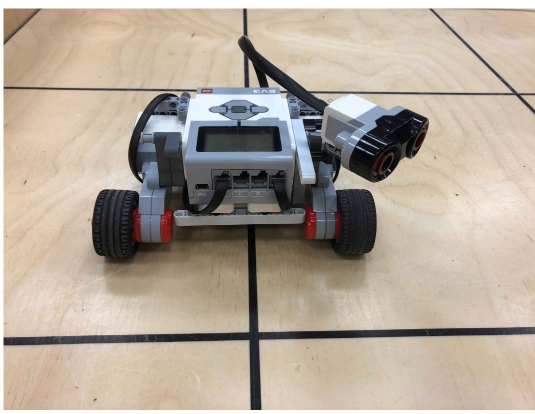
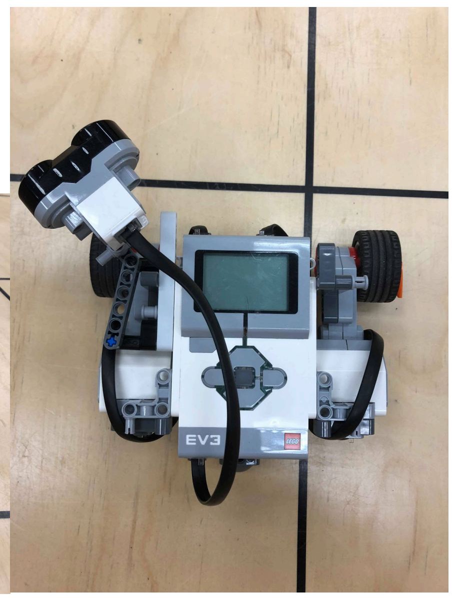
ECSE 211

Sep 21st 2018

Section 1: Design Evaluation

Hardware

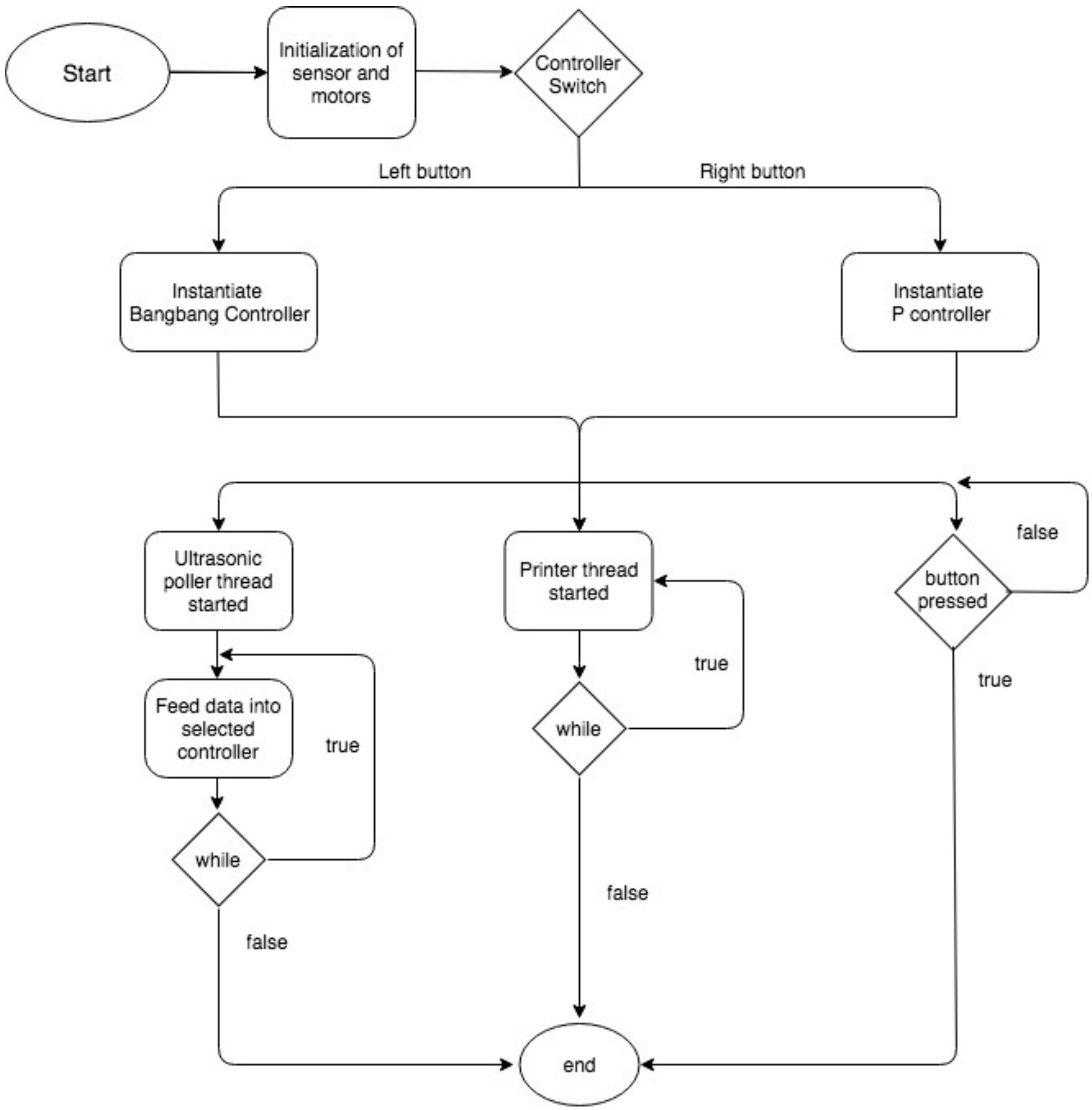
This robotic designs contains four major components: the main brick, two EV3 motors and one ultrasonic sensor. The motors were placed on both side of the brick, each motor is responsible for its respective wheel. We placed the sensor on front left of the brick with a tilt of 45 degrees, since the wall is always on the left-hand side of the robot. The tilted angle will create a wider range for the sensor which allows it to detect barriers easier, especially at corners and at U-turns.

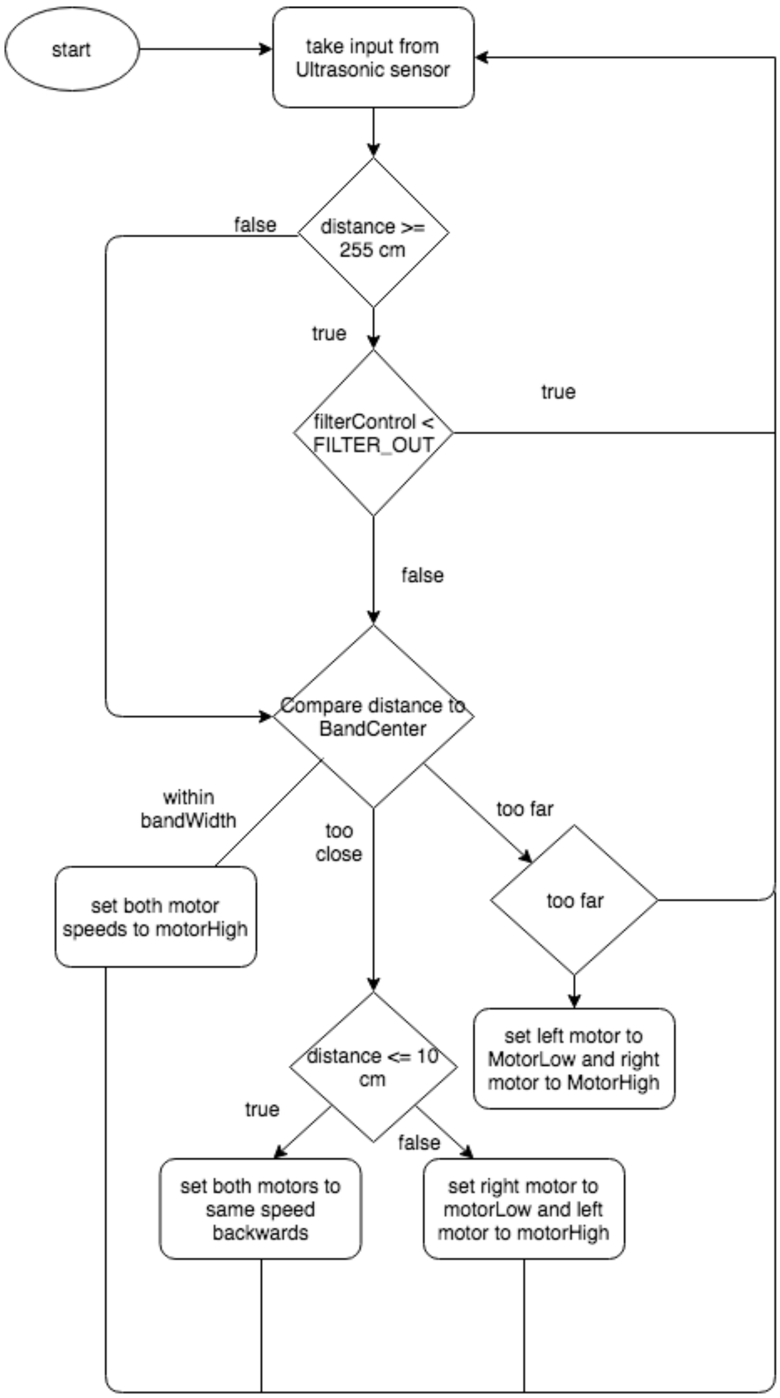
Software

The software comprises of a main class which initializes the motors, the sensors, printer class and ultrasonic sensor class for polling readings, an ultrasonic controller interface class and 2 classes for the actual implementation of the p-controller and the bang-bang controller. The two controller classes are where most of the logic and the data processing occurs.

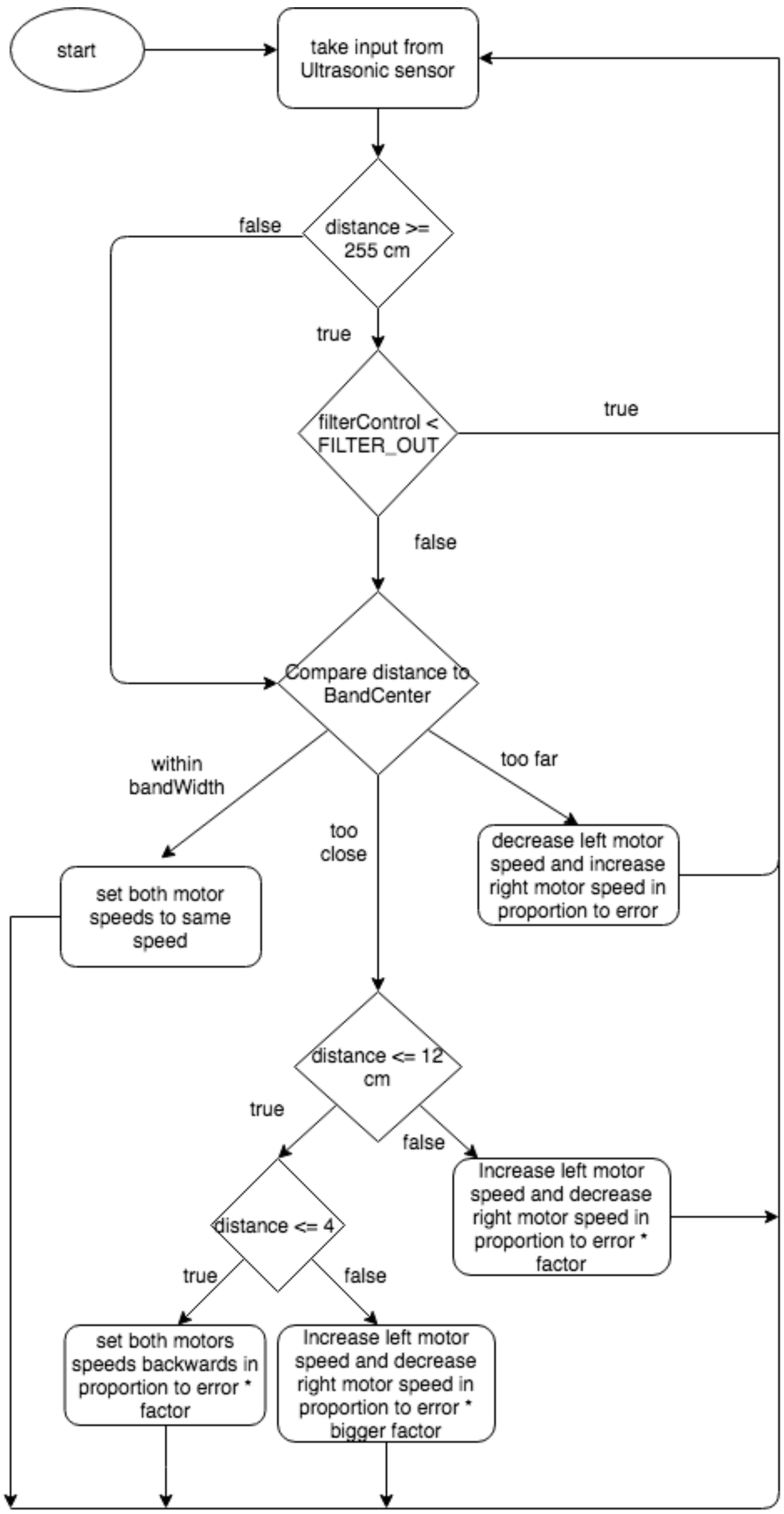
System Flow Diagram



Bang-Bang Controller Flow Diagram



P-type Controller Flow Diagram



Section 2: Test Data

Bang-Bang Controller Test

The Bang-Bang Controller keeps the robot stable within the band center distance from the wall, while ensuring in the first place it doesn't hit a block or gets off-track. It ignores slight deviations within the band width value (3 cm in our case). If the robot gets beyond the range (band center + band width), the motors apply different speeds to the wheel such that the left wheel slows down and the right wheel speeds up to tilt the car back to its original track.

Table of 3 independent runs performed on the same track from figure 1.1

Run Number	Band Center/cm	Observation
1	15	The robot showed more oscillations and was too close to the wall and bumping into it from time to time.
2	22 (used)	The robot moves with minimal oscillations and barely hitting the wall now.
3	30	Oscillations were minimal too but the robot takes very wide curves which worked perfect with U-turns but caused it to hit before fully turning at corners.

P-type Controller Test

In P-type controller, the motor speeds are controlled in proportion to the error value (band center – distance). Therefore, when the robot gets far away from the range (band center ± band width), more power is applied by the motor to abruptly readjust the robot back on track however, when it gets slightly away from the range, it adjusts back smoothly to avoid overshooting into the opposite side. This allows the robot to always maintain a distance closest to the band center value which was in our case was (22±3) cm and minimizes the chances of the robot getting off-track.

Table of 3 independent runs performed on the same track from *figure 1*

Run Number	Band Center/cm	Observation
1	15	The robot showed more oscillations and was too close to the wall and bumping into it from time to time.
2	22 (used)	The robot moves with minimal oscillations and barely hitting the wall now.
3	30	Oscillations were minimal too but the robot takes very wide curves which worked perfect with U-turns but caused it to hit before fully turning at corners.

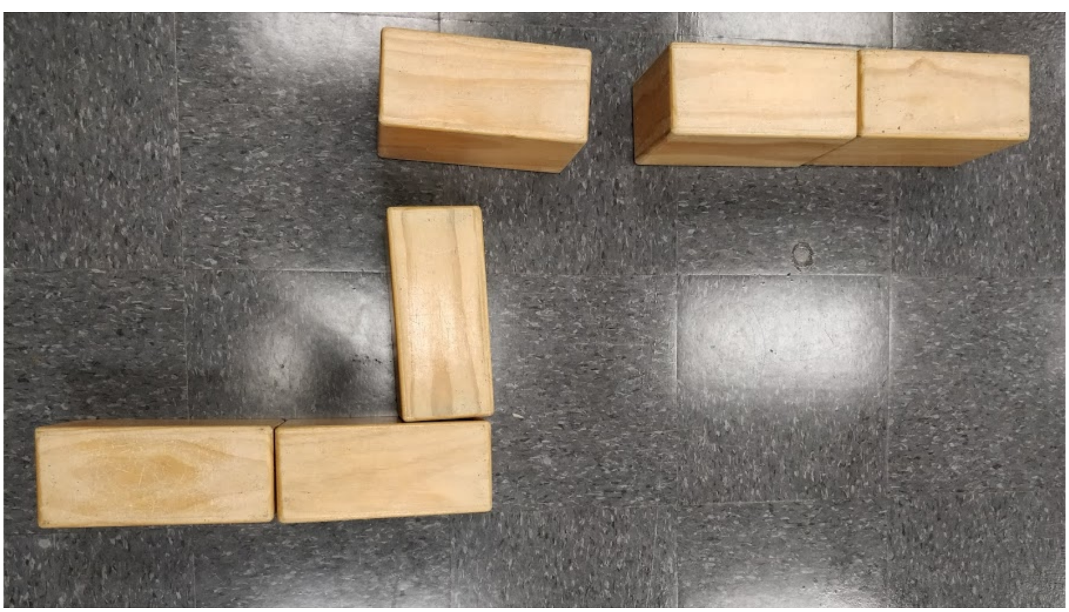
P-type Constant Test

Changing the constant for the p-type controller had an impact on the smoothness of the curves taken by the robot and the stability of the oscillation of the robot's distance from the wall.

Table of 3 independent runs performed on the same track from *figure 1.1*

Run Number	Constant	Observation
1	2	Robot took curves that were too wide and deviated from track when handling U-turns (sharp turns).
2	3 (used)	Robot took smooth curves and U-turns while having stable distance oscillation from the wall.
3	4	Robot's movement had sharp oscillations from the wall and took sharp U-turns that made it hit the wall.

figure 1 Track used for test data



Section 3: Test Analysis

For the P-type Controller, changing the constant affects the smoothness and reaction speed of the car to changes in its distance. Using a higher constant in the demo would cause the car to take sharper turns which might cause it sometimes to turn around itself while performing 180 degree turns (U-turns). Using a lower constant would result in the car taking wider curves which causes it to hit at concave corners due to not having enough time to perform a 90 degree turn.

Oscillations have occurred in all runs which exceeded the bandwidth for both controllers. For the Bang-Bang Controller, the robot oscillated within ±15 cm. For the P-type Controller, oscillations were within ±7~10 cm. Most of the big deviations or oscillations happen on sharp turns where the robot gets deviated from its course and tries to adjust itself back within the band center distance.

Section 4: Observations and Conclusions

Based on the previous test data analysis, the P-type controller showed more stable behavior and reliable readings across different runs. Robot running on P-type controller showed ability to readjust and recover back from deviations faster than Bang-Bang controller. P-type controller also had less oscillations and was able to maintain a closer range difference from the band center than Bang-Bang controller. This is mainly due to the fact that P-type controllers readjust themselves by reacting proportionally to the error difference, so they fix themselves quickly before they get further away, therefore resulting in smaller oscillations.

The Bang-Bang controller showed good performance as long as it did not go too far from track. However, sometimes at U-turns and sharp turns it went too far and, unlike P-type controller, it had difficulty getting back on track.

There were a few errors or abnormal behavior that the ultrasonic sensor experienced throughout the process. For example, while following a wall, the sensor would produce a maximum measurement while it's supposed to be taking normal measurements, which was clearly incorrect and confusing. However, the error was fixed by ignoring the maximum measurement taken, until it passes a filter. The filter caused measurement of the maximum distance to be ignored unless the sensor could detect it for at least 20 times in a row.

Furthermore, from our observations, the ultrasonic sensor took approximately 6 seconds longer than the motors for it to function. This problem is corrected by placing the robot 15 cm further from the starting point where the test should start. This gave ample time to the sensor to activate.

Section 5: Further Improvements

Hardware Improvements

1. Adding an extra sensor

The ultrasonic sensor has a beam width range of 30 cm, which means its limits are 15 cm to the right and to the left. Therefore, putting a sensor at 45 degrees still doesn't guarantee receiving reliable readings from neither the front nor the left side of the robot since they lie on the boundaries of the sensor range at this angle. Adding an extra sensor towards the front would improve various features of the robot including detecting obstacles in front of it and detecting corners and turning earlier before bumping.

2. Improve the design for mounting the Ultrasonic sensor

Implementing a sensor that can rotate or change angles while the robot moves so that its always facing the wall could improve the sensor readings and help better in detecting gaps in the wall.

3. Implement a more compact and cylindrical shape

Further Optimization to the robot design and wire connections to build a smaller robot could actually minimize the incidents of the robot scratching the wall from the side or wires touching the wall at corners and sharp turns

Software Improvements

1. Better filtering Algorithm

From time to time, the sensor received some readings in the middle of the runs that did not make any sense. A filtering algorithm could eliminate such readings and process readings to provide a feature for detecting gaps and sharp turns where special code behavior needs to be applied at such cases.

2. Spread Spectrum Phased Array Sonar

Two ultrasonic sensors could be hacked to build a spread spectrum phased array which provides improved sensor readings and monitoring.

3. Apply machine learning

Apply an algorithm that allows the robot to learn from its mistakes such as hitting a wall or getting off-track and adjusts the robot's turning angles and speeds so that it improves its course over time.

Controller Type Alternatives

PID Controllers

Using PID controller may have more accurate and reliable results over Bang-Bang and P-type controllers since it could monitor its movement and adjust its mistake based on collective data and integral and differential calculations. PID controllers are widely used in various engineering disciplines and processes.