Design Principles and Methods Michael Li and Cecilia Jiang

ECSE 211 (Fall 2020) 260869379, ######, Group #36

**Lab 1: Wall Following**

Prof. Frank Ferrie

McGill Faculty of Engineering

Due September 18th, 2019

**Software design**

Turning right in the P-controller is design such that the right well turns backwards and the left wheel turns forward. We wanted to avoid accelerating the left wheel, since it could touch the wall if it is too close from it. Therefore, when we turn right, we try to stay at the same position as much as possible and only changing the orientation of the robot.

There are two types of turns, gentle turns and sharp turns

Add class diagram and choose controller > USSensor > thread sleep

Class diagram for software control

**Hardware design**

The sensor was fixed at a specific degree that was tested in order to get consistent results. The wheel are considerably distant from each other in order to turn better and not having to have a big speed difference between the wheels in order to turn. Finally, the are attached so that the vehicle occupies less space as possible

**Section 2: Test Data**

**Testing the P-type controller constant**

In the case where the P-type controller’s constant is 15 (higher than the one used in the demo i.e. 4), its performance was bad and inconsistent. The robot went through multiple long oscillation periods as every time it is turning, it overturns especially on convex and concave corners. Our robot is moving in a counterclockwise manner along the path. Every time it turns left to get around a concave corner, it turns too sharply and consequently gets too close to the wall. Afterwards, it tries to readjust its position by turning right but it moves too far from the wall. Consequently, its path is remarkably unstable and characterized by various unexpected oscillations especially after sharp turns. It moves between the two extremes from being either too close or too far from the wall and is not able to maintain a constant distance from the wall.

In the case where the P-type controller’s constant is 2 (smaller than the one used in the demo i.e. 4), its performance was decent but still have room for improvement. The robot was able to maintain a consistent band center from the wall once it reaches the set offset distance and is able stabilize itself after various oscillations. It experienced very short oscillation period after sharp turns but was remarkably stable overall. However, when turning convex and concave corners, it does not turn sharply enough to imitate the shape of the path (i.e. it goes forward almost as a straight line for short corners) and it is lacking responsiveness as it occasionally turned a bit too late. In addition, when it is too close from the wall, it moves away from the wall by turning 180 degrees on itself instead of drive away from the wall slowly. However, after moving away from the wall, it maintained a consistent band center from the obstacles without any oscillation.

**Bang-Bang controller test**

Test 1:

At the beginning of the lap, the bandcenter was stable until the convex corner where the robot had to turn right. It turns right sharply and started experiencing oscillations. Afterwards, it turns left at a concave corner. The left turn was executed too roughly and the robot was getting to close from the wall. Consequently, a sharp right turn was triggered and the robot turns almost 180 degrees on itself before turning left once again to stay on the path. In addition, when the robot is too close from the wall, it experienced violent vibrations by turn away and towards the wall constantly. It is very inconsistent when it is too close from the obstacles and seemed to have trouble getting back to the set offset distance where the machine is relatively stable. Finally, the robot successfully completed a lap without touching the wall.

Test 2:

For the second trial, the band center was consistent for most of the lap. Its behaviour is very stable and is able to make small adjustments to preserve its distance from the wall. However, the robot behaves strangely at concave corners. On the first occurrence, the robot started by moving away from the wall by sharply turning right before turning left on the concave corner. On the second occurrence, the robot turns left too early and detects the wall at a very close distance. Hence, a sharp right turn is triggered and the robots turns almost 180 degrees to move away before turning left again. Consequently, almost each time it is executing a left “U” turn, it experiences intense oscillations. Finally, the robot successfully completed a lap without touching the wall.

Test 3:

On the third trial, the robot still makes multiple adjustments while getting around concave corners. Therefore, there is a heavy oscillation period during left sharp turns. However, it is very stable when going straight. When it gets too close from the wall, the robot makes a 180 degrees right turn to move away from it. In addition, on convex corners, it tends to make a 180 degrees right turn to move away from the wall to stabilize itself at the bandcenter. Finally, the robot successfully completed a lap without touching the wall.

**P-type controller test**

Test 1:

The robot is very stable during most of the itinerary, it makes small adjustments in order to maintain the band center distance (32 cm). The robot alternates well between going forward and turning. However, its right turn is still not perfect. Instead of making a 90 degrees right turn and stay perpendicular to the wall, the robot makes a 135 degrees right turn on itself every time that it detects a convex corner. The robot can imitates the shape of the path almost perfectly and maintain its equilibrium state with small adjustments. Finally, the robot successfully completed a lap without touching the wall.

Test 2:

The robot moves parallel to the wall and maintains a bandcenter of 32 cm most of the time. However, its performance on left turns is mediocre. Each left turn is accompanied by a sharp right turn that moves the robot away from the wall, because it was getting too close from the obstacle. Then, the left turn resumes since the robot is now detecting an infinitely big distance after the sharp right turn. Although this is an issue, the robot is capable of completing any circuit without any problems. Finally, the robot successfully completed a lap without touching the wall.

Test 3:

The behaviour of the robot is similar to the previous trials. It imitates well the shape of the path and maintains a constant bandcenter to the wall by making adjustments when necessary. Every correction is proportional to the magnitude of the error since the robot is capable of identifying to which extent it needs to turn in order to preserve the set offset distance from the wall. Important oscillations occur on convex corners, but the robot is able to get back to the equilibrium state quickly. Finally, the robot successfully completed a lap without touching the wall.

**Section 3: Test Analysis**

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

When the P-type constant is bigger than the one used in the demo, the robot is going through oscillation periods on concave and convex corners, because of excessive adjustments. Oscillations occur during each sharp turn, because the robot turns either too close or too far from the wall, therefore it needs to readjust itself. On concave corners, its left turn is too sharp. Consequently, the robot is getting to close from the wall. Hence, an excessive sharp right turn is triggered to move away from the wall. On convex corners, the robot makes a 180 degrees turn to move away from the wall which is unnecessary. Afterwards, it makes a sharp left turn to get back to the set offset distance where the robot is relatively stable. Therefore, its path is incredibly unstable around concave and convex corners, but rather stable when going straight. In fact, it switches between the two extremities (too close or too far from the wall) while making sharp turns.

When the P-type constant is smaller than the one used in the demo, its performance is stable but not optimal. The robot was able to maintain its bandcenter when going straight and return to the equilibrium state on various minimal oscillations. It experienced short oscillation periods on sharp turns but was not as intense compared to when the constant is very big. Nonetheless, it does not turn sharply enough to imitate the shape of the path when getting around convex and concave corners (i.e. it goes forward almost as a straight line for short corners). In addition, it is lacking responsiveness as it occasionally turned a bit too late. When the robot is too close from the wall, it moves away from the wall by turning 180 degrees on itself instead of driving slowly away from the wall. However, after distancing itself from the wall, it maintains a consistent band center from the obstacles without any oscillation.

**How much does your robot oscillate around the band center?**

The band center of the robot is 32 cm and its band width, 6 cm. This allows the robot to not make any corrections when the measured distance is within [26, 38] cm. This margin of 12 cm is relatively big and allows the robot to stay stable when it is around the bandcenter. Therefore, the band width is big enough to eliminate any unnecessary oscillations and small enough to trigger adjustments when the error is too bigger. Hence, the robot is relatively stable around the band center.

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

As the robot approaches a convex or concave corner, the absolute value of the error (i.e. the difference between the bandcenter and the measured distance) exceeds the bandwidth. For convex corners, the error exceeds the bandwidth by 5 to 25 cm (i.e. the bandwidth is 6 cm and the calculated error varies between 11 cm to 26 cm). The closer the robot is from the wall, the greater this error will be, therefore the more it would exceed the bandwidth. For concave corners, the error is negative, hence the absolute error is considered in this case. When the robot is executing a sharp left turn, the error exceeds the bandwidth by a value in the range of [1,∞[, because at some point the ultrasonic sensor is detecting nothing so it outputs an infinitely big distance. In this case, the absolute value of the error is infinitely big which exceeds the bandwidth.

When going straight, the robot sometimes exceeds the bandwidth of 1 to 5 cm, therefore the controllers will trigger small adjustments.

During sharp turns it exceeds bandwidth by almost 20

Does excee the bandwidth. For small corrections it exceeds by 5-8cm , for sharp turns, bigger than 8cm, it makes sharp turns.

**Describe how this occurs qualitatively for each controller.**

Bang-Bang is less stable compared to the as each time the error is bigger than the bandwidth it makes an adjustment.

The error is the distance between the set offset distance and the mesured distance by the US sensor. When getting around the corner, sharp turn, too close from the wall therefore distance is small, and exceeds the bandwidth

**Software improvements**

Implemeenting a PID controller

Change filter function

Change sensorpolling class

Implement a timer class instead of thread.sleep …

**Hardware improvements**

Fix sensor at 45% degree and lower to be able to see front and wall as much as possible. It is technically better

Put more weight to lower center of gravity

This is definitely a glitch we need to fix. Right turn 180 degrees

Better weight distribution for better traction on the wheels

Better sensor position at exactly 45 degrees