# Report

## 1. Wi-Fi Navigation Algorithm:

Our group has come across 3 algorithms for Wi-Fi navigation.

The first one is to first take 4 wifi signal values at the front, back, right, left of the robot. And then compare them with the medium value. If there is only one value above the medium, then just go that direction. If there are two or three values above the medium, then choose the direction between them.

The second algorithm is to make the robot circle for 1 round to get 8 signal values. And then circle again to get to the point where you got the maximum value, and turn 90 degrees right. That's the right direction.

The third algorithm is to go straight ahead and get the wifi signal periodically. Once the robot detects the wifi signal is decreasing, then turn left. To be more specific in the algorithm, the robot will record the latest three values. Xh1, is the oldest value, while xh3 is the latest value. If xh2<xh1, and xh3<xh1, then turn left. Once the signal value reaches to a point, it stops.

We chose the third one instead since the power of our motor performs very differently between full power and low power. During experiments, after we got the correct value for motors, battery power going down, and the motor which rotates faster in full power rotates slower than the other one when it's in low power. The problem is the same in the second algorithm. Considering the actual situation, we tried to make it go as simple as possible. So we chose the third algorithm.

## 2. Noise Filter

When we test our robot, we find it can find the right direction, as once it passing by the beacon, it will turn left. However, it also happens that the robot sometimes will turn left when it's not supposed to turn left. So what will happen is when the robot is just on the right way to our beacon, if it turns left in mistake, then it will turn left for another 3 times to find our beacon.

After debugging, we find there is always a bigger value happen. Though xh2<xh1 and xh3<xh1 will not let the robot passing by the beacon in mistake, the robot will turn left when it is not supposed to be if this unusual big value is updated to xh1( ).

So we add a judgement when updating xh1, xh2, xh3. If xh2>xh1+2, then xh1=xh1+2. By doing so, it will filter unusual wifi values.

## 3. Obstacle avoidance

If it detects an obstacle in a very short distance, it will go back. And then using the ultrasonic sensor to detect whether there is an obstacle in the left and right. If there is nothing on left, then turn left. If there is something in the left, but nothing in the right, then turn right. Otherwise, go back.

## 4. Combination

For combination, obstacle avoidance is always of the first priority. If there are no obstacles in the front, then it will go straight ahead and enter into wifi navigation. In wifi navigation, if the signal value is increasing, then go to the next loop. Otherwise, turn to the left direction. It won't go straight ahead at this point till the next loop when it makes sure there is no obstacle in front of it.