RTES Project Report

Professor Matthew Campisi

Thursday 6:00-8:30 pm

Group 6 team 3

December 20, 2019

We divide the task in two part: avoidance of the obstacle and the determination of the position of beacon.

Basic modules

We used a four motors vehicle, with a PWM interface, two servo, two ultrasonic sensors, three infrared sensors and one ESP8266 board. The codes can be divided in four parts.

Motion module

The control of motors by using PCA9685 PWM I2C interface code are in move.h/move.cpp.

WiFi module

The connection and detection of RSSI part is in wifi.h/wifi.cpp.

Ultrasonic module

The codes of ultrasonic sensor used to detect the obstacle are in ultra.h/ultra.cpp.

Because we need to detect the reflection of ultrasonic in different direction, we need attach the sensors to servos. The servos' turning module is in servo.h/servo.cpp.

Beacon detecting module

The algorithm we used to determined beacon and related code are in algo.h/algo.cpp.

The main control codes is Src.ino

Avoidance of the obstacle

Situation 1:

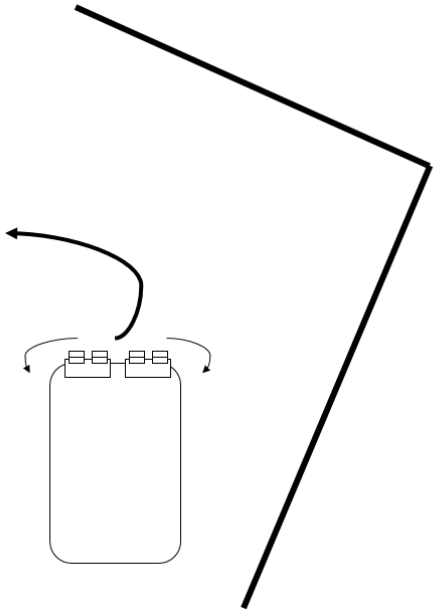


Fig Using Ultrasonic Sensor

When the ultrasonic sensors detect there is an obstacle in a threshold distance. We control the servo to rotate the sensors, separately record the distance of each distance. Then select a way with the longest distance and then turn to that way and go.

Using this method, we can avoid the majority of the obstacles.

Situation 2:

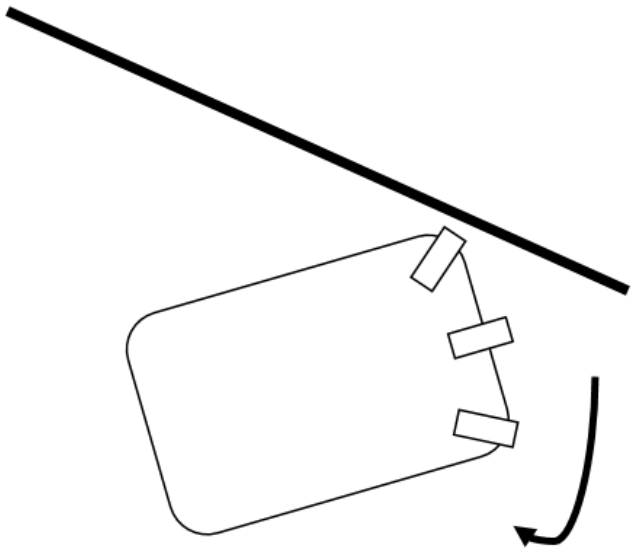


Fig 2

Determination of the beacon's position:

As we can see below, each time we can detect a RSSI from the board and we can using the function to calculate the distance from our board to the beacon:

$$D = 10^{\frac{(RSSI-A)}{N}}$$

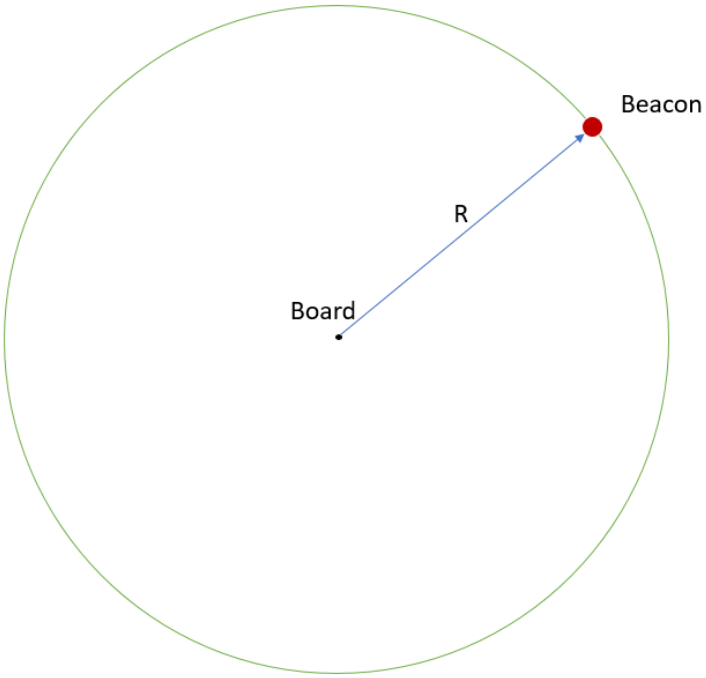
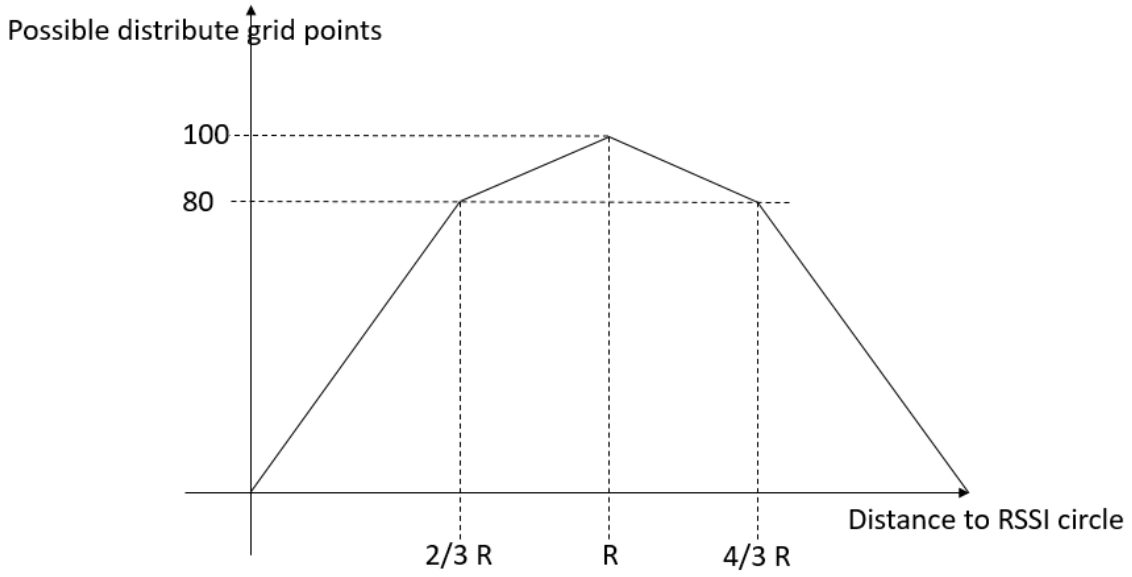


Fig 1: the possible circle of beacon to our board

Now we set a grid to simulate the onsite.

Because of the uncertainty of the RSSI and the approximately of using grid, we implemented a function to record the position:



Using the expression (1) we can calculate the approximate radius of possible circle. Then we update the grid points.

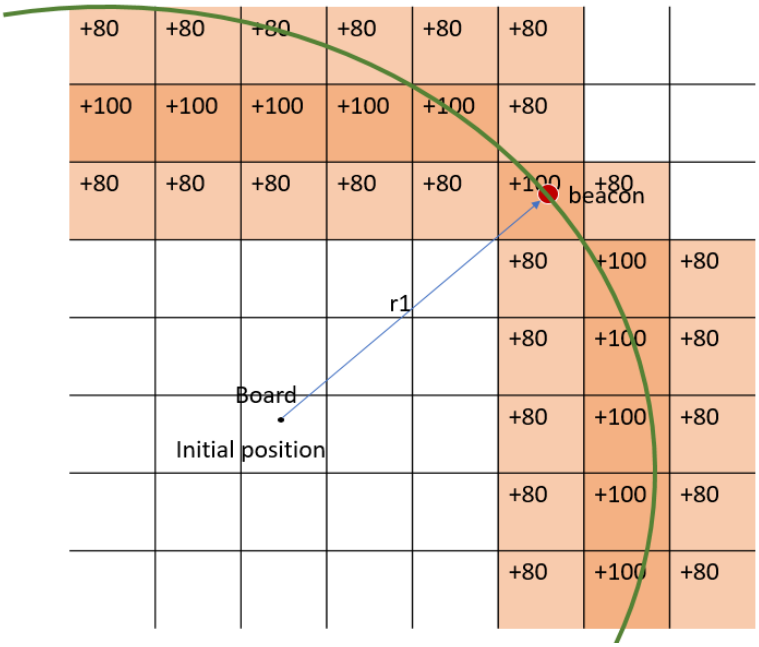
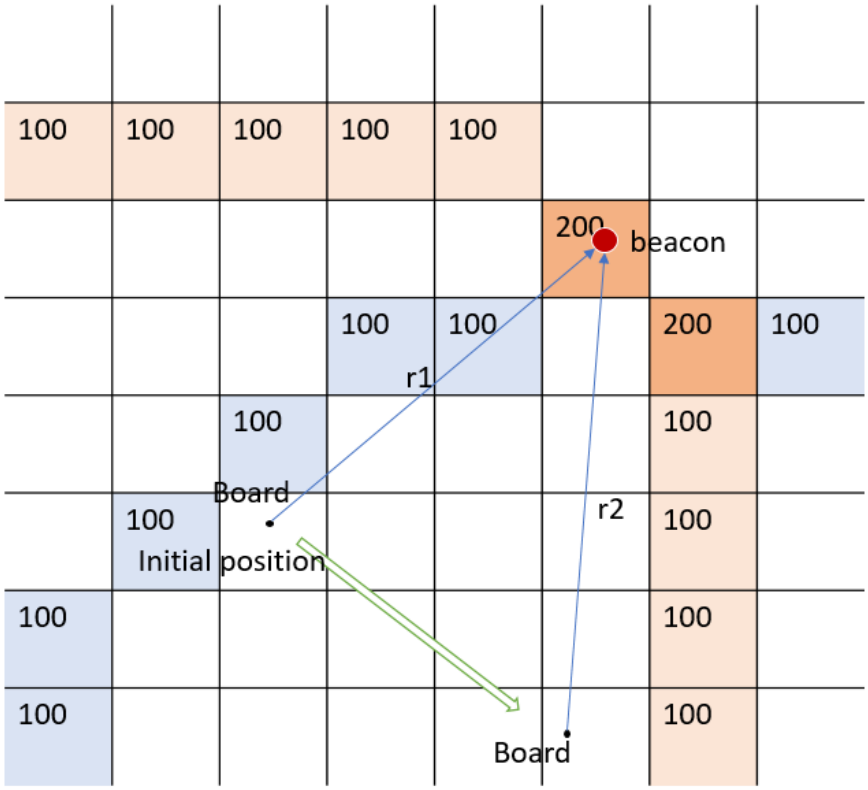


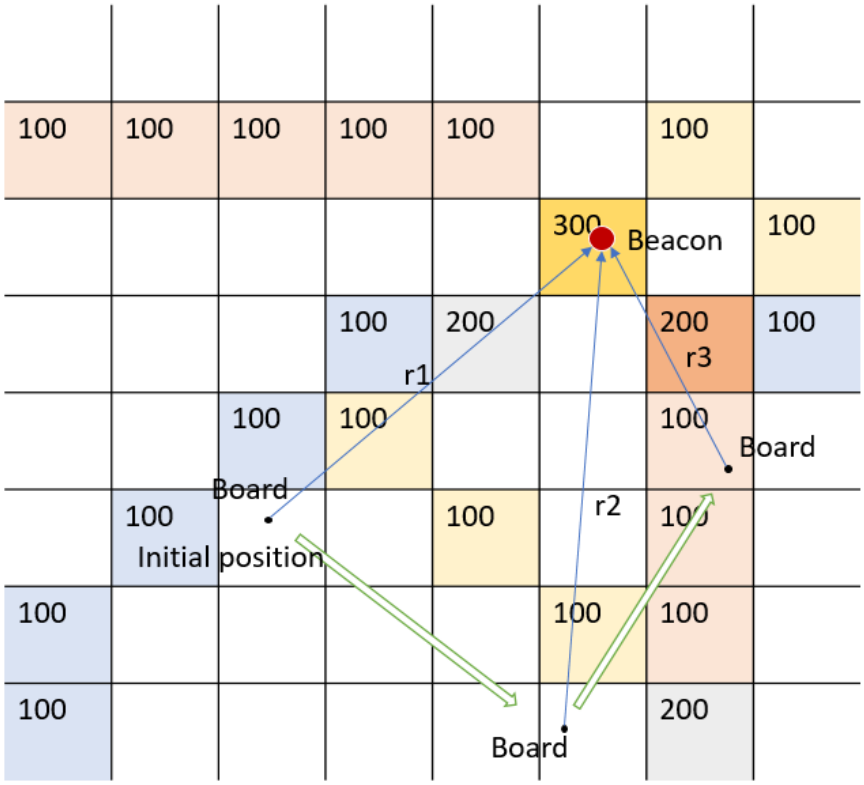
Fig 3 Using grid to store the possible circle

Basic Algorithm:

1. We start our vehicle randomly, record and update the gird.
2. Vehicle choose a direction to the highest points and move on a given distance. Calculate and update the gird.



3. Check the new RSSI, if it stronger than a threshold, stop, or repeat the step 2.


100
100
100
100
100
100
300
Beacon
100
100
200
200
r3
100
r1
100
100
100
Board
Board
100
Initial position
100
r2
100
100
100
100
200
Board