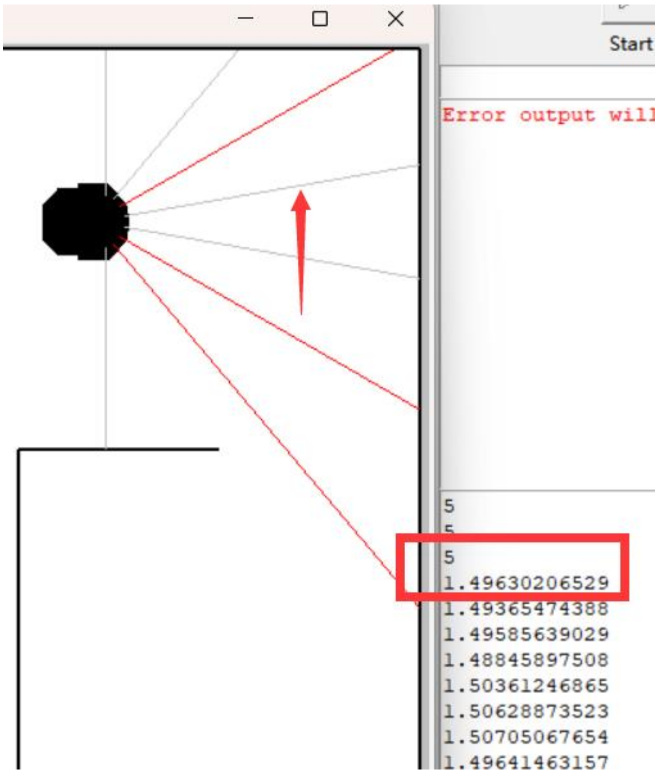
Checkoff 1.

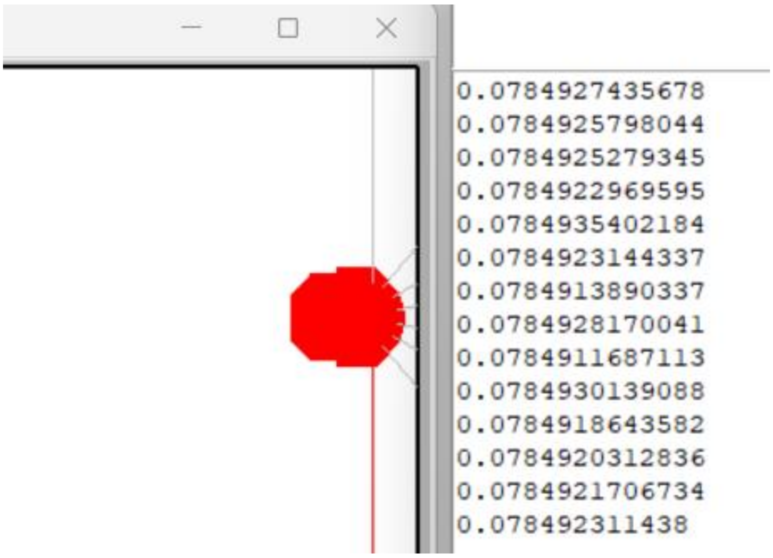
1.From how far away can you get reliable distance readings? What happens when the closest thing is farther away than that?

Answer：We can read the sonar measurement distance of the object up to 1.5 meters, at which point the sonar output is valid, and beyond this distance the uniform output is 5 meters.



2. What happens with things very close to the sensor?

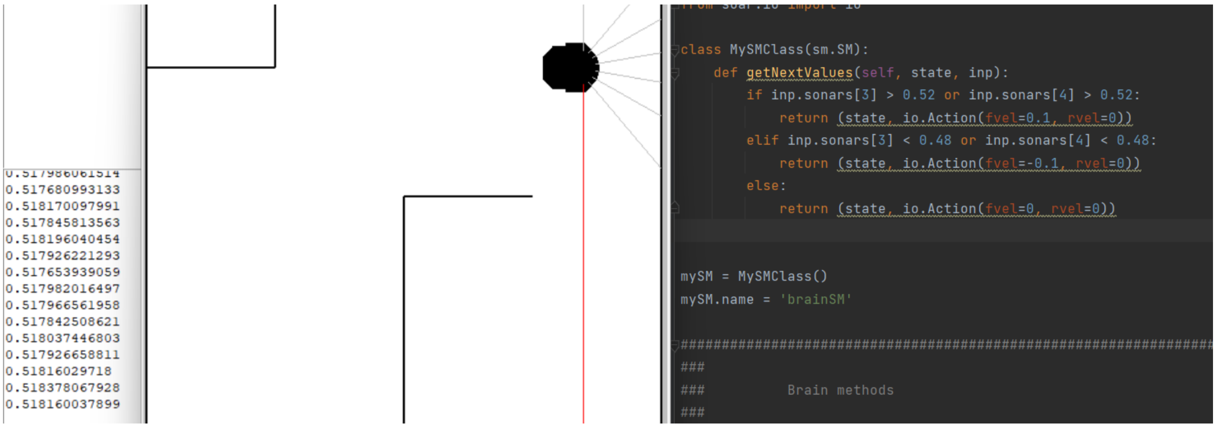
Answer：When the effective output of any sonar is less than 0.078m, the robot will be forced to stop.



3. Does changing the angle between the sonartransducer and the surface that it is pointed toward affect the readings? Does this behavior depend on the material of the surface? Try bubble wrap versus smooth foam core.

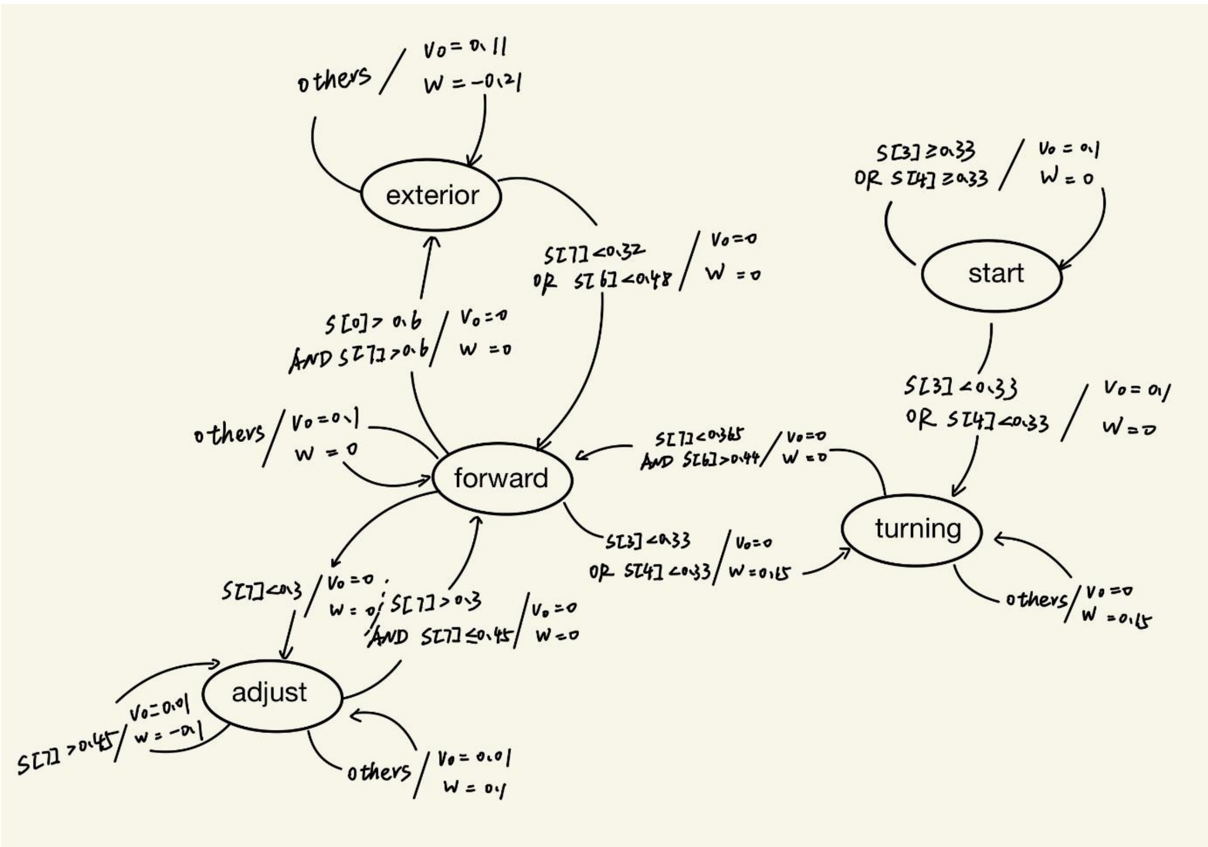
Answer: Due to the environment of the first experiment, this test was not performed.

Checkoff 2.



Idea is to let the robot always maintain a state judge, without introducing a new state, at each sampling time measuring 3 and 4 sensor output value (front approximate distance) , either one output and the target of 0.5 m was beyond the difference of error is allowed, for the corresponding forward or backward, until nearly reached 0.5 m away from obstacles.

Checkoff 3.



We take the effective output distance sampled by sonar as the state transition input of the state machine, perform the corresponding condition judgment so as to refresh the running state of the robot, and perform appropriate input and output to complete the desired task.

Checkoff 4.

We divided the whole task into three parts: initial wall sticking, inner and outer corner turning and position correction.

The robot can recognize the situation of the inner corner and the outer corner, and keep the error distance straight with the right wall in no condition. In order to correct the distance difference between the robot and the wall accumulated by the sensor error, a correction state is introduced.

Firstly, we clarify the role played by each sonar in the state machine. s3 and s4 are approximately regarded as sensors to measure the distance in front of the robot, which is mainly responsible for whether the robot reaches the initial state of wall sticking or internal corner turning； s7, as the most important sonar for judging the distance between the right side of the robot and the wall, is responsible for confirming whether the turning of the robot is completed and posture correction, while s6 assists s7 in position judgment. s0 acts with s7 to determine whether the robot reaches the outer corner.

We subdivide the status of the robot into five states: ‘start’, ‘turning’, ‘forward’, ‘adjust’, and ‘exterior’.

At the instant of all state transitions, the output is 0 to reduce error.

‘start’ is to complete the task of the robot walking in a straight line to find the wall in the initial state. The robot keeps the straight line at a constant speed without refreshing the state until s3 and s4 judge the existence of obstacles.

‘turning’ completes the inner corner turn. When obstacles are encountered in the front of the start or forward state, an indefinite Angle rotation is completed (we consider the first wall climbing and the subsequent outer corner turning as the same state). The exit is judged by s6 and s7 and uniformly converted to forward

‘forward’ is the center state of the robot and communicates with the other four states. When the sudden change of s0 and s7 data exceeds a certain range, it is judged as the turning state of the outer corner and converted to the turning state of the outer corner. When s7 ranging is too small, enter the adjust state to conduct attitude correction and ensure that the robot keeps straight driving; s3 and s4 will enter the turning state when they judge that there is an obstacle in front of them again

‘exterior’ completes the outer corner turn,only the exit judgment is the distance judgment between s7 and s6. Other cases continue to make a nearly circular turn.

‘adjust’ complete attitude correction, when the robot is toward the wall, to prevent the robot from hitting the wall in the forward state, according to s7 to judge the exit, the frame range is 0.3-0.45meters.

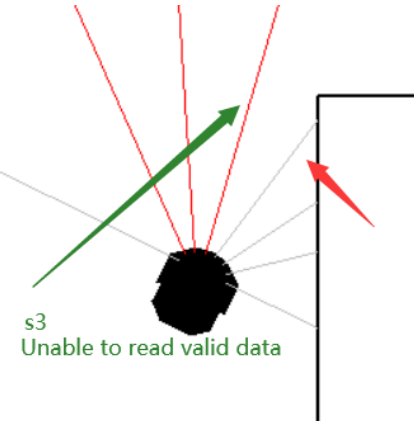
Summary

1. Why are the starting conditions s3 and s4 of internal corner turning judged by OR？

```
if state == 'start':
    action = io.Action(fvel = 0.1)
    if inp.sonars[3] < 0.33 or inp.sonars[4] < 0.33:
        newState = 'turning'
```

```
elif state == 'forward':
    if inp.sonars[3] < 0.33 or inp.sonars[4] < 0.33:
        newState = 'turning'
        action = io.Action(fvel=0, rvel=0.15)
```

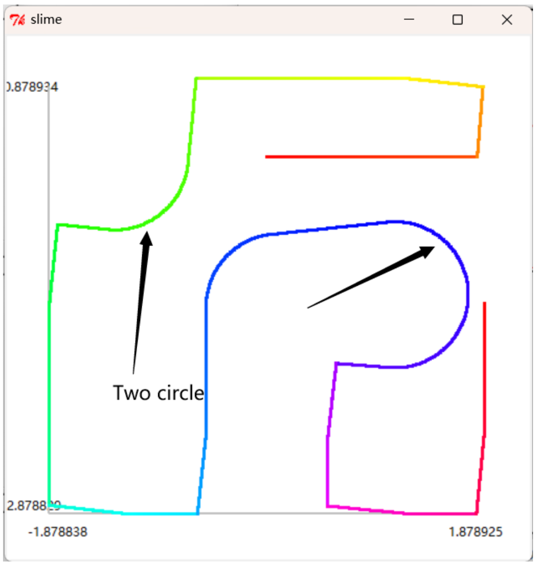
Because s3 and s4 have a certain Angle with the axial direction of the small vehicle, if AND is used to judge, it cannot deal with the problem that a certain sensor cannot obtain effective data when the axial direction of the robot has a large Angle with the plane, and can effectively deal with the first time when the robot is leaning against the wall in the case of an Angle.



2. How to realize the outer corner turning？

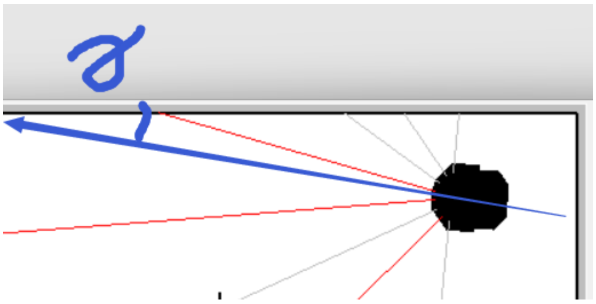
The turning state of the outer corner is judged by the sudden change of s7 and s0 data . At this point, the s7 sonar points to the outer corner cusp .We expect the robot outer corner cusp on a standard circle, the orientation of the robot is the direction of linear velocity, $\omega R = v$, under the condition of the angular velocity determined roughly determines speed and judged by s7 to make state exit. However, in the actual coding, we found that s6 was used to determine the exit in advance to prevent excessive turning of the robot and dynamically adjust the data of ω and v at the same time

```
elif state == 'exterior':
    if inp.sonars[7] < 0.32 or inp.sonars[6] < 0.48:
        newState = 'forward'
        action = io.Action(fvel=0, rvel=0)
    else:
        newState = 'exterior'
        action = io.Action(fvel=0.11, rvel=-0.21)
    return (newState, action)
```

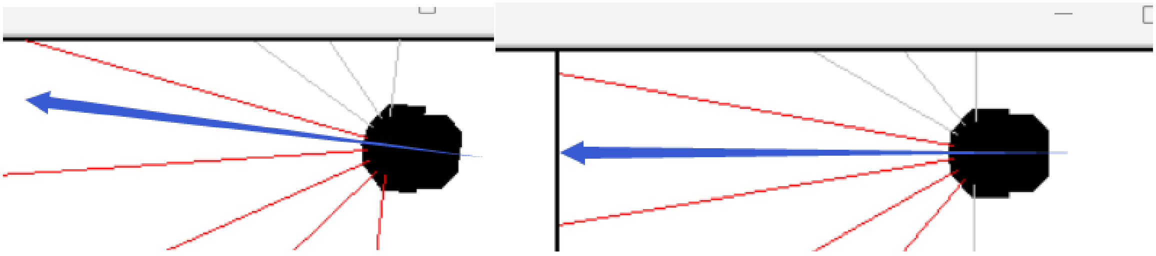


3. Only s7<0.3 was set at the adjust entrance, but why did you judge s7>0.45 when it was in condition?

At the turning state exit, the parameters set are conservative. After the inner corner turning is completed, the axis of the robot has an acute Angle with the wall. We only need to consider the situation that the robot is too close to the wall, and do not need to worry about the distance from the wall. Since our 'adjust' runs by the center rotation of the robot, there is a situation where s7 is always less than 0.3meters. We retain a certain linear velocity of 0.01， when abnormal, the robot will deviate from the wall until 0.45meters, in which case the robot can still be properly corrected.



Conservative outer Angle rotation exit judgment



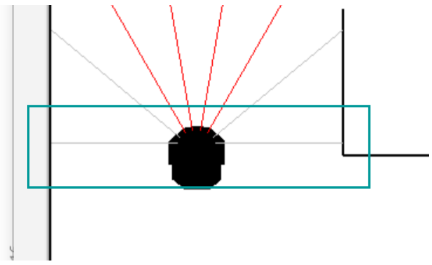
Realization of attitude adjustment

4. Why is s6 and s7 conditions set for ‘turning‘ exits？

   When the robot is centered, the sonar rotation is not constant centered. When turning the outer corner, the setting of 0.33 (0.365) for s7 will be reached in advance without sufficient rotation, and s6 is set to ensure that the robot completes as full a rotation as possible.

5. Why is the initial state set to ‘start‘ instead of cforward‘？

   In the initial state, the robot is far away from the wall and to prevent misjudgment for the outer corner condition



6. For the robot and the wall inclined acute Angle ($\alpha < 90°$), whether it can also run stably?

   ‘adjust‘ makes it possible