**Obstacle Avoidance Testing 2 (wallFollower)**

Tester: Eric Vuong

Author: Eric Vuong

Hardware Version: 1.0

Software Version: V2.4

**Purpose**: Test the efficiency of the robot’s ability to avoid incoming obstacles when navigating to a specified waypoint.

**Decision**: The obstacle avoidance logic will be implemented into the final project if the robot successfully avoids all created scenarios.

**Measurements**:For each scenario, the robot will be tested on its ability to avoid an obstacle. Therefore, a failure occurs when the robot collides or rubs any obstacle.

**Test objective:** The objectives are to confirm the robot’s ability to avoid obstacles, to look for any weak-points and specific scenarios where the obstacle avoidance logic would fail.

**Assumptions:**

The obstacles will not be located at the edge of the competition area, because the robot may fall off the play area if it tries to avoid the obstacle.

**Procedures:**

For each run, the robot will be placed at grid location (1,1) facing north and will navigate to (1,3), and then (3,1).

There are a total of three scenarios: normal, extreme case 1 and extreme case 2.

Each scenario will have 10 runs. If the robot fails to avoid an obstacle, a detailed description will be provided for that run.

**Expectations:**

The robot should be capable of successfully passing all three cases. The normal case should be fairly easy for the robot. For the extreme case 1, the robot should be able to avoid easily, because a simple wall follower should allow it to pass the test. For the extreme case 2, the robot should also pass easily, because the robot should be able to spot incoming obstacles while in avoidance mode. Thus, when the robot’s orientation matches the current waypoint, it will exit obstacle avoidance and travel to the desired location.

Scenarios:  
Normal case: Simple block that prevents the robot from advancing forward.

Figure 1. Normal Case: Only one obstacle is present and the obstacle is placed at an angle perpendicular to the robot’s orientation when encountering the obstacle.

Figure 2. Extreme Case 2: Multiple obstacles, one after the other, placed in such a way that the robot will enter many obstacle avoidance mode.

Figure 3. Extreme Case 2: Multiple obstacles, one after the other, placed in such a way that the robot will enter many obstacle avoidance mode.

Test report:

**Table 1.** Obstacle avoidance run. The test was performed following the procedure as described above.

|  |  |  |  |
| --- | --- | --- | --- |
| Run | Normal Case | Extreme Case 1 | Extreme Case 2 |
| 1 | Pass | Pass | Pass |
| 2 | Pass | Pass | Fail |
| 3 | Pass | Pass | Pass |
| 4 | Pass | Pass | Pass |
| 5 | Pass | Pass | Pass |
| 6 | Pass | Pass | Pass |
| 7 | Pass | Pass | Pass |
| 8 | Pass | Pass | Pass |
| 9 | Pass | Pass | Pass |
| 10 | Pass | Pass | Pass |

**Analysis**:

**Normal case**

The robot was able to successfully avoid the obstacle and go to the destination. The robot almost rubbed the obstacle most of the time, but did not. Thus, maybe the constants set to the obstacle avoidance could be changed in a way so that the robot does not come too close to the obstacle during avoidance.

**Extreme Case 1**

The robot passed extreme case 1, where it entered obstacle avoidance everytime it sees an incoming obstacle and successfully avoided it just like in the normal case.

**Extreme Case 2**

The robot successfully avoided the obstacle 9 times out of the 10 run. The only time it failed was because the left wheel of the robot collided with the obstacle and the ultrasonic sensor couldn’t perceive it.

**Conclusion**:

The Obstacle avoidance V1.2 implemented in Software Version 2.4 has a reliable logic for avoiding obstacles. Even if the robot failed 1 run out of the 30 runs, it can be corrected easily by setting a further distance from the robot to the wall during obstacle avoidance.

**Action**:

The new obstacle avoidance ( Obstacle avoidance V1.2) will be implemented into the software V2.4 and onwards. It is important to note that the WallFollower’s safe distance from the wall will be increased in order to prevent any random collision with the obstacle once it exits. Thus, another test will be done to further improve it.

**Distribution: Software Development**

**Follow up: Eric Vuong Obstacle Avoidance Test 3**