**Obstacle Avoidance Testing 3 (3 Ultrasonic Sensor & PController)**

Tester: Eric Vuong

Author: Eric Vuong

Hardware Version: 3.X

Software Version: V3.9

**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 and fails to arrive at waypoint.

**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 random obstacles will only be placed on the random obstacle field of the play area and none will be present on the shooting area, the starting location and the safe zones.

**Procedures:**

For each run, the robot will be placed at grid location (0,0) facing north and will navigate to (6,6)

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 robot will detect any incoming obstacles located at 20cm from the robot at a full range of 180 degrees.

Scenarios:

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: The corner of an obstacle is placed in the front sensor’s view so that it is hard to perceive the obstacle as ultrasonic waves will be deflected to the left and right

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, if the robot sees 2 walls consecutively, it will avoid the obstacle the other way to prevent moving to the set path.

Test report:

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

|  |  |  |  |
| --- | --- | --- | --- |
| Run | Normal Case (error) | Extreme Case 1 (error) | Extreme Case 2 (error) |
| 1 | Pass 3cm | Pass 3cm | Pass 7cm |
| 2 | Pass 2cm | Pass 3cm | Pass 9cm |
| 3 | Pass 3cm | Pass 4cm | Pass 6cm |
| 4 | Pass 3cm | Pass 6cm | Pass 7cm |
| 5 | Pass 3cm | Pass 3cm | Pass 8cm |
| 6 | Pass 3cm | Pass 2cm | Pass 7cm |
| 7 | Pass 4cm | Pass 3cm | Pass 12cm |
| 8 | Pass 3cm | Pass 3cm | Pass 9cm |
| 9 | Pass 4cm | Pass 3cm | Pass 9cm |
| 10 | Pass 3cm | Pass 4cm | Pass 9cm |

**Analysis**:

**Normal case**

The robot was able to successfully avoid the obstacle and go to the destination. Accuracy tolerable of ~3cm.

**Extreme Case 1**

The robot passed the test as it was able to detect the incoming obstacle. However, the robot detects the obstacle at a distance closer than expected. Thus, for angled obstacles, the robot detects at a closer distance. Accuracy tolerable of ~3.5cm.

**Extreme Case 2**

The robot successfully avoided the obstacle, and arrived near destination. Upon detecting two consecutive walls, the robot knows that it is a dead end, and will avoid from the other side by following the initial obstacle. It is important to notice that the error at the end of the obstacle avoidance is ~9cm, which is fairly significant. Thus, the odometer’s accuracy is clearly affected by the amount of time spent in avoiding an obstacle.

**Conclusion**:

The Obstacle avoidance implemented in Software Version 3.9 has a reliable logic for avoiding obstacles. However, it is important to realize that the longer the robot stays in obstacle avoidance, less accurate the odometer will be.

**Action**:

The new obstacle avoidance will be implemented into the software V3.9 and onwards. Further improvement on its obstacle avoidance algorithm will be improved post-beta demonstration to the client. Also, the time will be recorded to ensure that the obstacle avoidance is done in the fastest time.

**Distribution: Software Development and Hardware integration**

**Follow up: Eric Vuong Obstacle Avoidance Test 4 (Post-Beta Demo result adjustments)  
 Eric Vuong, Adrian Pan, Gwyneth 🡪 Speed test on obstacle avoidance.**