Complete Testing

**Test 1**

**Date :** 2021/4/6

**Tester:** Shichang Zhang

**Author:** Shichang Zhang

**Hardware version:** 1.5 (in Part 2.5 of [Hardware Document](https://docs.google.com/document/d/11jkA_S_xBqyCbcn2NyMuM-OMDEybDfRy/edit#))

**Software version:** 2.0 (in Part 7.0 of [Software Document](https://docs.google.com/document/d/19JaY5629aUu4Y4rjoQJ-jWyeQLqNSAcr/edit))

**Test Purpose:**

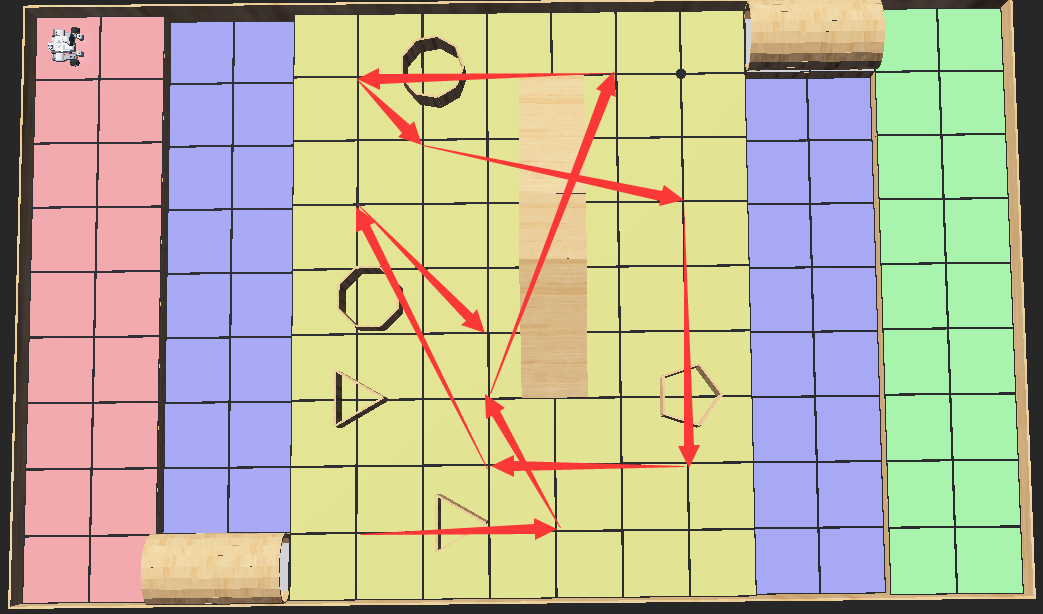
Determine whether the robot meets the client's needs.

**Test Procedure:**

1. The robot is placed at the corner 3 of the competition field.
2. The robot is placed with an initial angle 80°.
3. Start the program. Enable the WiFi parameter delivery.
4. Records the performance of the robot on each step defined by client needs.
5. If the robot succeeded in returning to the start point within the time limit, stop the program and measure the final translation Euclidean Distance Error.

**Test Data:**

Test map:



Test waypoints: {(5.1),(8,1),(7,3),(9,8),(5,8),(6,7),(10,6),(10,2),(7,2),(5,6),(7,4)}

**Expected Result:**

The robot finishes all client needs and returns to the start point with translation error within 5cm.

**Test Results:**

|  |  |
| --- | --- |
| Trial# | Comment |
| 1 | 1. The robot successfully received the current game parameters.  2. The robot successfully localized to the grid and issued a sequence of 3 beeps.  3. The robot was successfully navigated to the corresponding bridge.  4. The robot successfully passed though the bridge and reached the mainland.  5. The robot was travelling to (5,1). The robot was successfully navigated to (5,1)  6. The robot was travelling to (8,1). The robot successfully avoided the obstacle between (5,1) and (8,1). The robot was navigated to (8,1).  7. The robot was travelling to (7,3). The robot was successfully navigated to (7,3). However, the robot had errors in translation. It was not exactly located at (7,3).  8. The robot was travelling (9,8). The robot successfully went to the startpoint of the overpass but with some translation error and huge angle error since the robot touched the overpass when it tried to move to the startpoint of the overpass from (7,3). The robot was trying to go over the overpass. Due to the large angle error, the robot failed to go over the overpass and crashed. |
| Figure 1.1 The robot fell down the overpass due to errored moving direction |
| 2 | 1. The robot successfully received the current game parameters.  2. The robot successfully localized to the grid and issued a sequence of 3 beeps.  3. The robot was successfully navigated to the corresponding bridge.  4. The robot successfully passed though the bridge and reached the mainland.  5. The robot was travelling to (5,1). The robot was successfully navigated to (5,1)  6. The robot was travelling to (8,1). The robot successfully avoided the obstacle between (5,1) and (8,1). The robot was navigated to (8,1).  7. The robot was travelling to (7,3). The robot was successfully navigated to (7,3).  8. The robot was travelling (9,8). The robot successfully travelled to the start point of the bridge with few translation errors. The robot normally climbed the overpass and went down the overpass. The robot successfully detected the intersection between overpass and ground so it successfully finished the go overpass action. The robot tried to travel to the (9,8). However, due to unknown reasons, the robot turned to an error angle and travelled to (9,8) with relatively large translation error.  9. The robot was travelling to (5,8). Since the robot’s position was deviated at last waypoint, the robot hit the bridge during travelling. Then the robot crashed. |
| Figure 1.2 The robot failed to localize after going across the overpass |
| 3 | 1. The robot successfully received the current game parameters.  2. The robot successfully localized to the grid and issued a sequence of 3 beeps.  3. The robot was successfully navigated to the corresponding bridge.  4. The robot successfully passed though the bridge and reached the mainland.  5. The robot was travelling to (5,1). The robot was successfully navigated to (5,1) but with small transition errors.  6. The robot was travelling to (8,1). The robot successfully avoided the obstacle between (5,1) and (8,1). The robot was navigated to (8,1). But also, with small translation errors.  7. The robot was travelling to (7,3). The robot was successfully navigated to (7,3) also with small translation error.  8. The robot was travelling (9,8). The robot successfully travelled to the start point of the bridge with inaccurate y-value. Although the robot started from an errored position when it tried to go over the overpass, it still normally climbed the overpass and went down the overpass. The robot successfully detected the intersection between overpass and ground so it successfully finished the go overpass action. The robot tried to travel to the (9,8). However, similar to trial 2, the robot turned to an errored angle and travelled to (9,8) with relatively large translation error.  9. The robot was travelling to (5,8). Since the robot’s position was deviated at the last waypoint (9,8), the robot hit the bridge during travelling. Then the robot crashed. |
| Figure 1.3 The robot hit the bridge because incorrect position in real world |

**Test Report:**

The pass rate of this map is 0%. The robot could not even complete half of the tasks. From the test output, we could see that the robot performed excellent for localization and passing bridge and performed unpleasant for running laps on the main island. For trial 1, the error made by avoiding the obstacle influenced the following actions of the robot. The robot climbed the overpass with an obvious angle error so it fell down from the overpass and crashed. For trial 2 and 3, the critical error occured when the robot finished going overpass action. The robot turned to an errored angle and failed to travel to the next waypoint accurately. The deviated position rendered the robot hit the overpass and crashed when the navigation system led the robot to the next waypoint. Overall, our design performs favorably for localization and going across the bridge. But for the actions on the mainland, we need to improve our algorithms to avoid the occurrence of accumulated errors and critical errors in some cases.

**Conclusion:** Fail

**Action:**

1. Revise the algorithm for repairing the odometer after finishing going over the overpass.
2. Revise the obstacle avoidance algorithm to reduce the error generated during the avoid action.
3. Retest this map after correcting the limits listed above.

**Distribution:** software development, hardware development

**Test 2**

**Date :** 2021/4/6

**Tester:** Junjian Chen

**Author:** Junjian Chen

**Hardware version:** 1.5 (in Part 2.5 of [Hardware Document](https://docs.google.com/document/d/11jkA_S_xBqyCbcn2NyMuM-OMDEybDfRy/edit#))

**Software version:** 2.0 (in Part 7.0 of [Software Document](https://docs.google.com/document/d/19JaY5629aUu4Y4rjoQJ-jWyeQLqNSAcr/edit))

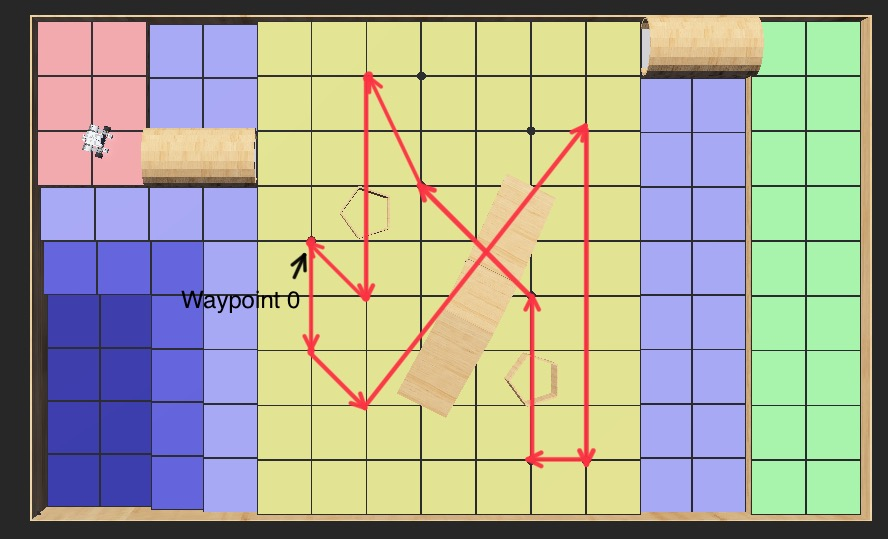
**Test Purpose:**

Test whether the robot can finish the whole process in the given map.

**Test Procedure:**

1. The robot is placed at the upper left corner of the competition field.
2. The robot is placed with an initial angle 80°.
3. Start the program. Enable the WiFi parameter delivery.
4. Records the performance of the robot on each step defined by client needs.
5. If the robot succeeded, return to the start point within the time limit, stop the program and measure the final translation Euclidean Distance Error.

**Test Data:**

Test Map:

Input waypoints: [(5, 5), (5, 3), (6, 2), (10, 7), (10, 1), (9, 1), (9, 4), (7, 6), (6, 8), (6, 4)]

**Expected Result:**

The robot finishes all client needs and returns to the start point with translation error within 5cm.

**Test Results:**

|  |  |
| --- | --- |
| Trial# | Comment |
| 1 | 1. The robot successfully received the current game parameters. 2. The robot successfully localized to the grid and issued a sequence of 3 beeps. 3. The robot was successfully navigated to the corresponding bridge. 4. The robot successfully passed though the bridge and reached the mainland. 5. The robot was travelling to (5,5). The robot was successfully navigated to (5,5) 6. The robot was travelling to (5,3). The robot was successfully navigated to (5,3) 7. The robot was travelling to (6,2). The robot was successfully navigated to (6,2) 8. The robot recognize it should pass the overpass 9. The robot passes the overpass successfully 10. The robot was travelling to (10,7). The robot was successfully navigated to (10,7) 11. The robot was travelling to (10,1). The robot was successfully navigated to (10,1) 12. The robot was travelling to (9,1). The robot was successfully navigated to (9,1) 13. The robot was travelling to (9,4). The robot detects and avoids the obstacle. Then the robot successfully navigates to (9,4) 14. The robots recognize it should go underpass 15. The robot crushes to the edge of the underpass when passing |
| Figure 2.1 The robot hit the edge of the underpass when going underpass |
| 2 | 1. The robot successfully received the current game parameters. 2. The robot successfully localized to the grid and issued a sequence of 3 beeps. 3. The robot was successfully navigated to the corresponding bridge. 4. The robot successfully passed though the bridge and reached the mainland. 5. The robot was travelling to (5,5). The robot was successfully navigated to (5,5) 6. The robot was travelling to (5,3). The robot was successfully navigated to (5,3) 7. The robot was travelling to (6,2). The robot was successfully navigated to (6,2) 8. The robot recognize it should pass the overpass 9. The robot fails to pass the overpass |
| Figure 2.2 The robot fell down the overpass |
| 3 | 1. The robot successfully received the current game parameters. 2. The robot successfully localized to the grid and issued a sequence of 3 beeps. 3. The robot was successfully navigated to the corresponding bridge. 4. The robot successfully passed though the bridge and reached the mainland. 5. The robot was travelling to (5,5). The robot was successfully navigated to (5,5) 6. The robot was travelling to (5,3). The robot was successfully navigated to (5,3) 7. The robot was travelling to (6,2). The robot was successfully navigated to (6,2) 8. The robot recognize it should pass the overpass 9. The robot passes the overpass successfully 10. The robot was travelling to (10,7). The robot was successfully navigated to (10,7) 11. The robot was travelling to (10,1). The robot was successfully navigated to (10,1) 12. The robot was travelling to (9,1). The robot was successfully navigated to (9,1) 13. The robot was travelling to (9,4). The robot detects and avoids the obstacle. Then the robot successfully navigates to (9,4) 14. The robots recognize it should go underpass 15. The robot crushes to the edge of the underpass when passing |
| Figure 2.3 The robot hit the edge of the underpass when going underpass |

**Test Report:**

Trial 1 and 3 fail because there is an obstacle near the exit of the underpass. After it passes the underpass, when it prepares to go to the next waypoint, it detects the obstacle. To avoid it, the robot will turn right and go another path. However, the robot turns right and hits the edge of the underpass.

Trial 2 fails because the hardware default. When it goes on the overpass, the ball caster is stuck for a moment and the angle of the robot deviates, which causes the robot to fall down.

**Conclusion:** The test fails.

**Action:**

1. Revise the obstacle avoidance
2. Revise the position of ball caster of hardware design

**Distribution:** Software Development, Hardware Design

**Test 3**

**Date :** 2021/4/11

**Tester:** Shichang Zhang

**Author:** Shichang Zhang

**Hardware version:** 1.5 (in Part 2.5 of [Hardware Document](https://docs.google.com/document/d/11jkA_S_xBqyCbcn2NyMuM-OMDEybDfRy/edit#))

**Software version:** 2.3 (in Part 7.0 of [Software Document](https://docs.google.com/document/d/19JaY5629aUu4Y4rjoQJ-jWyeQLqNSAcr/edit))

**Test Purpose:**

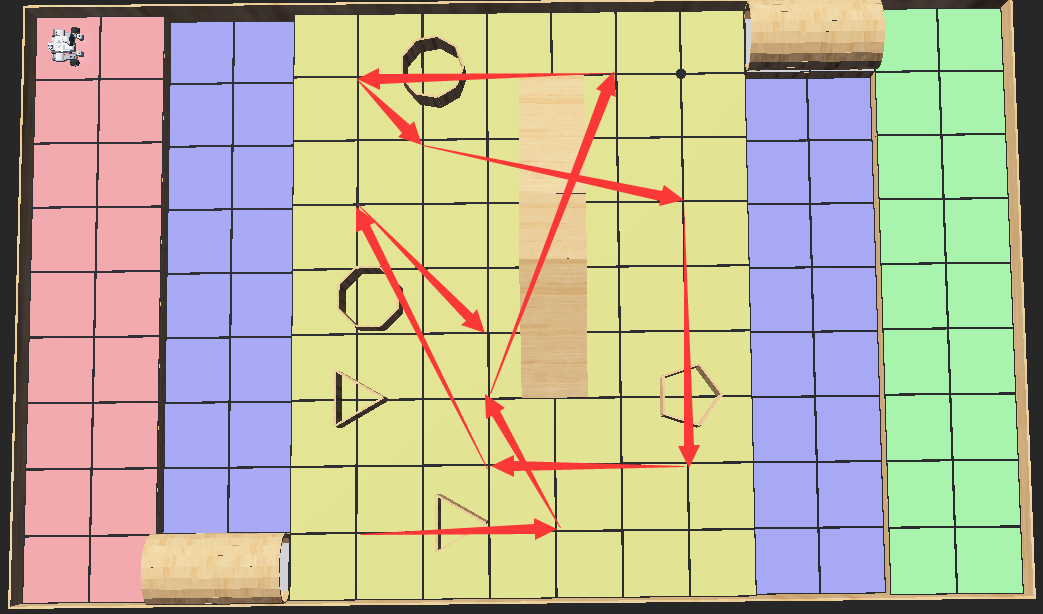
Determine whether the robot meets the client's needs. (In Test1, the robot failed to pass this map, so retest it again)

**Test Procedure:**

1. The robot is placed at the corner 3 of the competition field.
2. The robot is placed with an initial angle 80°.
3. Start the program. Enable the WiFi parameter delivery.
4. Records the performance of the robot on each step defined by client needs.
5. If the robot succeeded in returning to the start point within the time limit, stop the program and measure the final translation Euclidean Distance Error.

**Test Data:**

Test map:



Test waypoints: {(5.1),(8,1),(7,3),(9,8),(5,8),(6,7),(10,6),(10,2),(7,2),(5,6),(7,4)}

**Expected Result:**

The robot finishes all client needs and returns to the start point with translation error within 5cm.

**Test Results:**

|  |  |
| --- | --- |
| Trial# | Comment |
| 1 | 1. The robot successfully received the current game parameters.  2. The robot successfully localized to the grid and issued a sequence of 3 beeps.  3. The robot was successfully navigated to the corresponding bridge.  4. The robot successfully passed though the bridge and reached the mainland.  5. The robot was travelling to (5,1). The robot was successfully navigated to (5,1)  6. The robot was travelling to (8,1). The robot successfully avoided the obstacle between (5,1) and (8,1). The robot was navigated to (8,1).  7. The robot was travelling to (7,3). The robot was successfully navigated to (7,3). This time, the robot accurately stopped at (7,3).  8. The robot was travelling (9,8). The robot successfully went to the startpoint of the overpass accurately. The robot was trying to go over the overpass. The robot normally climbed the overpass and went down. The robot passed the overpass accurately. The robot tried to localize to a point to reduce odometer errors and then travel to (9,8). However, the localization point is too close to the wall, the robot hit the wall and crashed. |
| Figure 3.1 The robot failed to localize after going overpass |
| 2 | 1. The robot successfully received the current game parameters.  2. The robot successfully localized to the grid and issued a sequence of 3 beeps.  3. The robot was successfully navigated to the corresponding bridge.  4. The robot successfully passed though the bridge and reached the mainland.  5. The robot was travelling to (5,1). The robot was successfully navigated to (5,1)  6. The robot was travelling to (8,1). The robot successfully avoided the obstacle between (5,1) and (8,1). The robot was navigated to (8,1).  7. The robot was travelling to (7,3). The robot was successfully navigated to (7,3).  8. The robot was travelling (9,8). The robot successfully went to the startpoint of the overpass accurately. The robot was trying to go over the overpass. However, the robot’s ball casters got stuck for a while when the robot tried to climb the overpass. The robot’s motion was influenced so the moving direction of the robot was errored. The robot climbed the overpass with an errored angle. Finally the robot fell down the overpass while traveling on the overpass. |
| Figure 3.2 The robot fell down the overpass due to the errored angle |

**Test Report:**

The pass rate of the retest is 0%. The robot still could not complete half of the tasks. From the test output, the robot currently still can hardly handle the localization problem after going over the overpass. In trial 1, the robot figured out the wrong localization point (on the wall) so it just travelled to it and crashed. However, for trial 2, the robot failed to climb the overpass. From our previous unit tests on this part, the robot is able to climb the overpass given it arrives at the previous waypoint accurately. In this trial, the pre-condition was satisfied, the robot travelled to (7,3) and the overpass start point accurately. But for unknown reasons, maybe due to the hardware, the ball casters got stuck when the robot tried to climb. Finally the robot fell down the overpass and crashed. Overall, the robot can hardly localize to a point after going over the overpass and the robot sometimes will have problems when climbing the overpass.

**Conclusion:** Fail

**Action:**

1. Improve the algorithm for the localization after going down the overpass.
2. Check which factor causes the ball caster to get stuck when climbing the overpass.(may be worked later, since it is time consuming to figure out)
3. Test other maps. Some functions are not tested due to the failure.

**Distribution:** software development, hardware development

**Test 4**

**Date :** 2021/4/11

**Tester:** Shichang Zhang

**Author:** Shichang Zhang

**Hardware version:** 1.5 (in Part 2.5 of [Hardware Document](https://docs.google.com/document/d/11jkA_S_xBqyCbcn2NyMuM-OMDEybDfRy/edit#))

**Software version:** 2.3 (in Part 7.0 of [Software Document](https://docs.google.com/document/d/19JaY5629aUu4Y4rjoQJ-jWyeQLqNSAcr/edit))

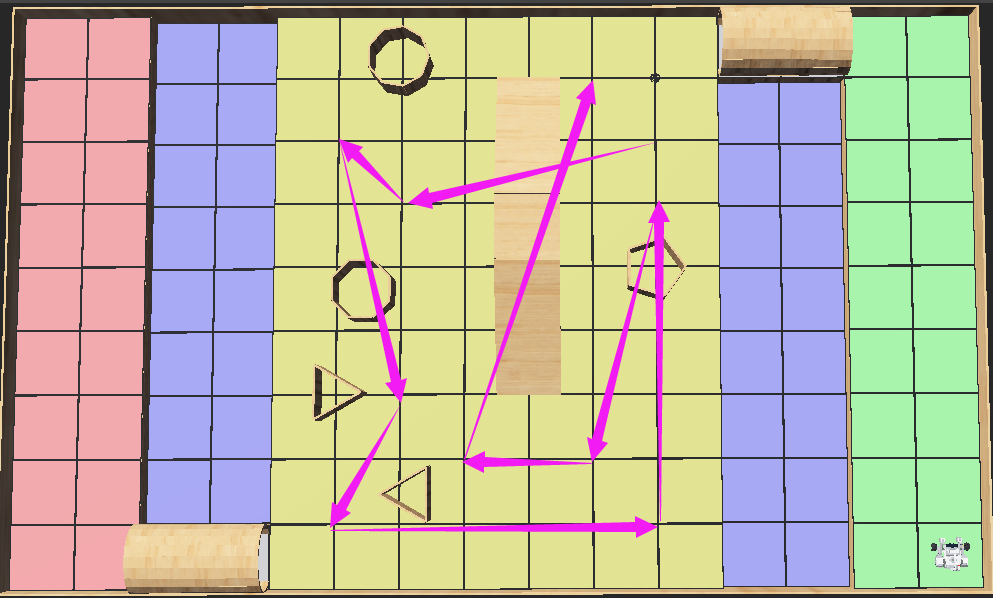
**Test Purpose:**

Determine whether the robot meets the client's needs. (In Test1 and Test 3, the robot failed to pass the overpass so other functions of the robot cannot be tested. In this test, we will input a series of waypoints that asks the robot to pass the overpass at the end. So other functions of the robot will be tested first.)

**Test Procedure:**

1. The robot is placed at the corner 1 of the competition field.
2. The robot is placed with an initial angle 45°.
3. Start the program. Enable the WiFi parameter delivery.
4. Records the performance of the robot on each step defined by client needs.
5. If the robot succeeded in returning to the start point within the time limit, stop the program and measure the final translation Euclidean Distance Error.

**Test Data:**

Test map:

Test waypoints: {(10.7),(6,6),(5,7),(6,3),(5,1),(10,1),(10,6),(9,2),(7,2),(9,8)}

**Expected Result:**

The robot finishes all client needs and returns to the start point with translation error within 5cm.

**Test Results:**

|  |  |
| --- | --- |
| Trial# | Comment |
| 1 | 1. The robot successfully received the current game parameters.  2. The robot successfully localized to the grid and issued a sequence of 3 beeps.  3. The robot was successfully navigated to the corresponding bridge.  4. The robot successfully passed though the bridge and reached the mainland.  5. The robot was travelling to (10,7). The robot localized to the nearest point after going across the tunnel. The robot was successfully navigated to (10,7).  6. The robot was travelling to (6,6). The robot realized that it should go underpass during the trip. The robot successfully went to the front of the overpass accurately. The robot passed the underpass, but with some translations. The robot tried to travel to (6,6). The robot arrived (6,6) with a larger y-coordinate than expectation.  7. The robot was travelling to (5,7). The robot successfully stopped at (5,7). But due to the accumulated errors, the translation errors were larger.  8. The robot was travelling to (6,3). The robot successfully detected that there was an obstacle on the path. The robot tried to avoid the obstacle, The robot called the avoid methods too late, there was no space for the robot to rotate. The robot hit the obstacle and crashed. |
| Figure 4.1 The robot failed to avoid obstacle in time |
| 2 | 1. The robot successfully received the current game parameters.  2. The robot successfully localized to the grid and issued a sequence of 3 beeps.  3. The robot was successfully navigated to the corresponding bridge.  4. The robot successfully passed though the bridge and reached the mainland.  5. The robot was travelling to (10,7). The robot localized to the nearest point after going across the tunnel. The robot was successfully navigated to (10,7).  6. The robot was travelling to (6,6). The robot realized that it should go underpass during the trip. The robot successfully went to the front of the overpass accurately. The robot passed the underpass, but with some translations. The robot tried to travel to (6,6). The robot arrived (6,6) with a larger y-coordinate than expectation.  7. The robot was travelling to (5,7). The robot successfully stopped at (5,7) with some translation errors.  8. The robot was travelling to (6,3). The robot successfully detected that there was an obstacle on the path. The robot tried to avoid the obstacle successfully. The robot tried to travel to (6,3). The robot successfully reached (6,3).  9. The robot was trying to travel to (5,1). The robot detected the obstacle on the right by mistake. The robot tried to avoid the obstacle on the right. The robot then travelled to (5,1).  10. The robot was trying to travel to (10,1). The robot successfully detected that there was an obstacle on the left. The robot avoided the obstacle successfully. The robot tried to travel to the (10,1). The robot successfully stopped at (10,1).  11. The robot was trying to travel to (10,6). The robot successfully detected that there was an obstacle on the path. The robot successfully avoided it. The robot tried to travel to (10,6). The robot successfully arrived (10,6).  12. The robot was trying to travel to (9,2). The robot successfully detected that there was an obstacle on the path. The robot tried to avoid the obstacle. The robot’s tail hit the overpass while avoiding the obstacle. The robot tried to travel to (9,2). The robot arrived (9,2) with relatively large translation errors and a huge angle error.  13. The robot was trying to travel to (7,2). Because at the last waypoint the robot’s angle was errored considerably, the robot totally deviated from the planned path. |
| Figure 3.2 The robot totally deviate from planned path |

**Test Report:**

The pass rate of the retest is 0%. In this test, the robot completed more tasks than in Test 1 and Test 3. From the test output, we found that the failures were both caused by the obstacle avoidance. In trial 1, the robot did not avoid the obstacle in time. This was due to the inaccuracy of the ultrasonic reading. In trial 2, the robot’s tail hit the overpass and influenced the next actions. Our algorithm cannot plan the accurate avoiding path according to the obstacle size. So sometimes the robot will hit other obstacles while avoiding. We also found that the going underpass action would generate some translation errors on the robot. Overall, the robot is able to pass underpass, but with small translation errors. The robot is also able to avoid obstacles generally, but it cannot get a very accurate avoid path.

**Conclusion:** Fail

**Action:**

1. Improve the obstacle avoidance algorithms’ flexibility. (planned an accurate path according to the position and size of the obstacle.)
2. Improve the going underpass algorithm. Try to limit the error after going underpass.

**Distribution:** software development, hardware development