HARDWARE V3.01 Navigation Constants Precision Test

GROUP 4

2nd December 2016
Verison 1

Abstract

This test is to as accurately as possible identify the constants for the track radius and wheel radius of Hardware version 3.01.

Contents

1 BACKGROUND

1.1 Edit History

Jake Zhu: 2016-10-28, Initial set up

Mamoun Benchekroun: 2016-11-11, Added abstract section

Quentin Norris: 2016-11-16, Added test information, procedure, expected result, and conclusion.

Quentin Norris: 2016-11-23, Filled in action and distribution

Jake Zhu: 2016-11-30, Edited Procedure and Software Version

1.2 Test Information

Tester:Quentin Norris, Mamoun Benchekroun, Tristan

Author: Quentin Norris

Hardware Version: 3.01

Software Version: be35803

2 GOAL

Identify as accurately as possible the wheel radius and track radius.

3 PROCEDURE

Setup - Test 1: Wheel radius:

1. Place robots wheels on the intersection of the black lines closest to one of the corners of the 8x8 board.
2. Orient it so its facing the black line away from the wall, so if it were to drive forward it would make it to the next side.
3. Upload WheelRadiusTest.java
4. Hit the run button to run the code.
5. After the robot has reached a complete stop, measure the distance from the intersection of the black lines from the robot. This intersection should be the one the robot traveled closest to.

6. Enter the distance away from the table. If the value is less than 1 centimeter, decrease the wheel radius. If it more than 1 centimeter increase the wheel radius.
7. Repeat the test until a value within 1 centimeters for wheel radius is consistently recorded.

Setup - Test 2: Track width:

1. Place robot on one of the tiles and mark where the two wheels are.
2. Upload TrackWidthTest.java
3. Hit the run button to run the code.
4. The robot is coded to spin 10 times, let it finish.
5. Record the angle, measured with a protractor, the two wheels are off by from their initial position. Record this angle in the data table.
6. If the angle is larger than +15 degrees, reduce the track length to decrease the angle. If the angle is smaller than -15 degrees, increase the track length to increase the angle.
7. Repeat the process until a consistent track length has been found where the angle is within +/- 15 degrees.

Test:

1. For both test 1 and test 2 run the their respective codes and hit the right arrow.

4 EXPECTED RESULT

We expect the initial tests to be somewhat inaccurate but by the final test we will have accuracy within +/- 1 centimeter for wheel radius, for the first test. For test 2 for track accuracy we expect the final result to be within +/- 15 degrees of the initial position.

5 TEST REPORT

5.1 Test 1: Wheel Radius

Trial	Above/Below	Distance	Wheel Radius (cm)
1	Below	8.7	2.127
2	Below	8	2.127
3	Below	6	2.10
4	Below	2	2.05
5	Below	1.1	2.04
6	Below	1.1	2.04
7	Below	1	2.04
8	Below	0	2.03
9	Below	0.1	2.03
10	Above	0.1	2.03

As shown by the table, the best wheel radius we found is 2.03.

5.2 Test 2: Track Width

Trial	Angle	Track Width (cm)
1	+135	11.6
2	+185	12
3	+45	11.5
4	+45	11.45
5	35	11.4
6	30.5	11.4
7	10	11.25
8	-10	11.25
9	-3	11.25
10	8	11.25

As shown by the table, the best track is clearly 11.25.

6 CONCLUSION

From the values gathered during the various trials we did for both wheel radius and track length, we have concluded that the most precise wheel radius is 2.03 and the most precise track length is 11.25.

7 ACTION

For any future hardware changes involving the chassis of the robot or any major changes, this test should be done again to re-calibrate the wheel radius and track length of the new hardware version of the robot.

8 DISTRIBUTION

This information will be sent to the software team so they can update these values in their code for the final project. This information will also be sent to the rest of the testing team so they know what values are most accurate to use when testing various codes.