Test Document

Project: LIBERTY

Task: Test the system playing as Green

Document Version Number: 2.0
Date: 30/11/2017
Author: Xijun Liu
Editor: Andi-Camille Bakti

Edit History: https://github.com/Gabetn/DPM_01_Project_Documentation

TABLE OF CONTENTS

1. TESTS:

1.1 Green integration corner three test

Test 1: Full playthrough integration test

Date: 23/11/2017
Tester: Xijun Liu
Author: Edward Son

1) This test will validate the functionality of the integrated code so far, in terms of being a green team player, and running through the whole course.
2) This test should make the robot successfully complete zip-line traversal, capturing, bridge navigation, and return to starting position. The angle, X and Y position error will be measured between each of these components. The angle should be under 4 degrees of error. The X and Y positions should be under 2 cm of error.
3) The brick should be positioned near corner 1, in any orientation. The robot receives data over wifi, the zip-line coordinates (ZC_G_x, ZC_G_y) and (ZC_R_x, ZC_R_y), as well as the search zone and bridge coordinates. Once data is acquired, the command "start" is used on the server GUI and the robot starts localization. After localization, the angle is measured, and a button press continues to navigation. It first navigates to the points (ZO_G_x, ZO_G_y) and localizes. Then the X, Y and angle errors will be measured. The robot proceeds to mount onto the zip-line at point (ZC_G_x, ZC_G_y), and will dismount and re-localize itself at point (ZO_R_x, ZO_R_y). X, Y and angle error will be measured. The following button presses allow it to sweep through the red zone and capture the right flag and navigate it through the bridge back to the starting point.
4) The brick should end up back at the starting corner after having successfully completed the whole playthrough (zip-line traversal, capturing, bridge traversal).
5)

	localization(offset)			waypoint (offset)			mounting	localization(after dismounting) (offset)			Capturing	Bridge Traversal
Trial	x (+/- 1cm)	y (+/- 1cm)	theta(+/- 1 deg)	x (+/- 1cm)	y (+/- 1cm)	theta(+/- 1 deg)		x (+/- 1cm)	y (+/- 1cm)	theta(+/- 1 deg)		
1	0.3	0.5	2.3	0.7	0.6	3.4	yes	1	1.2	1.4	Failed	Failed
2	0.5	1.1	3.2	2.3	0.4	2.3	yes	0.2	0.5	1.5	Failed	Failed
3	1.2	0.3	1.1	1.2	2.1	1.5	yes	2.1	0.7	1.8	Success	Failed
4	0.4	0.2	0.6	0.5	0.3	4.5	yes	0.3	0.7	2.9	Failed	Failed
5	2.2	0.6	1.8	0.8	0.9	2.0	yes	0.5	1.1	3.1	Success	Failed
6	0.3	0.6	1.9	0.6	1.6	2.2	yes	1.5	1.5	1.2	Failed	Failed
7	1.1	0.3	1.1	1.0	0.4	6.1	Failed					
8	0.3	0.6	5.4	0.3	1.2	7.2	Failed					
9	0.5	0.7	2.3	0	1.7	5.9	Failed					
10	0.6	0.9	0.8	0.9	0.8	3.2	yes	0.8	0.5	3.7	Success	Failed
average	0.74	0.58	2.05	0.83	1	3.83		1.83	2.17	1.95		
ST	1.22	0.52	2.21	1.23	1.58	3.53		1.07	0.53	1.96		

Green Team Integration Test

Fig. 1 Testing Result For Green Team Integration Test.

6) The brick successfully localizes with 100% success rate. It can complete the zip-line traversal task with a success rate of 70%. However, it can only capture the right flag for three times among the ten trials, which is 30%. Finally, it fails to traverse the bridge, and return to the starting point. The standard deviation and mean for initial localization are relatively small, which is considered very good. For the re-localization at point (ZO_G_x, ZO_G_y), there are three trials of which the angle error is relatively large and causes failure to mount on to the zip-line consequently. Overall, the robot runs fairly well up to the localization after dismounting. The following tasks (i.e. capturing and bridge traversal) failed almost for all trials. In conclusion, the result of integration of the green team is not ideal. What seems to be the reason for the testing failure is that the robot seems not to keep tracking its position properly during the process.

7) There should be a bridge test individually. In addition, robot should keep tracking its position at the red zone.

The average (AM) was calculated by using the following formula:

$$Average = \frac{1}{n}\sum_{i=1}^{n} a_i = \frac{1}{n}\left(a_1 + a_2 + \cdots + a_n\right)$$

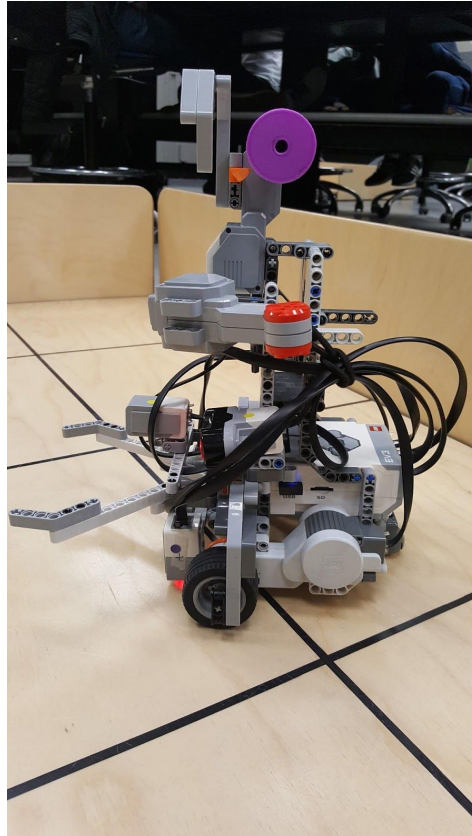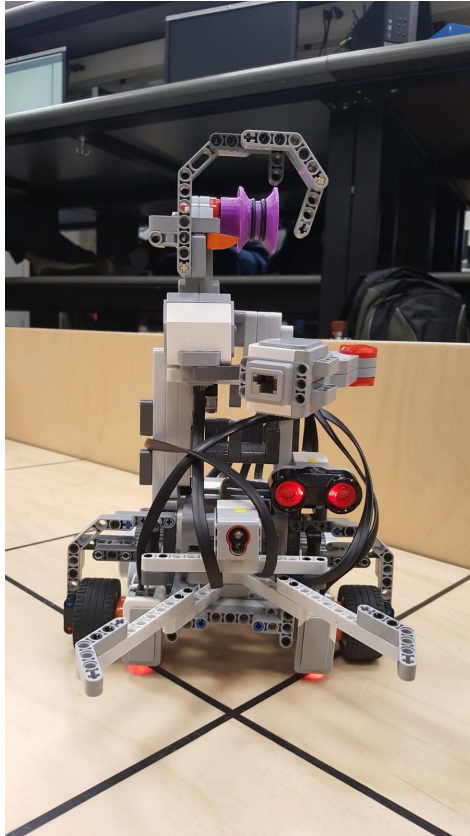We use the sample standard deviation formula (see below) to calculate the sample standard deviation.

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

$\bar{x}$ = Mean
N = Sample size
x_i = Sample at i

2. HARDWARE

See **HARDWARE - 2.0.**

3. Source Code used

See github group repository at commit: d4f57913294490a9db953a5af5995e3eea75ffd5