Test Document

Project: LIBERTY

Task: Test the bridge navigation

Document Version Number: **1.0**
Date: 29/11/2017
Author: Edward Son
Editor: Andi-Camille Bakti

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

McGill

TABLE OF CONTENTS

1. TESTS:

1.1 Bridge test

Test 1: Corner 2 bridge crossing test

Date: 21/11/2017
Tester: Edward Son
Author: Edward Son

1) This test will validate the functionality of the bridge navigation, and if navigation class properly creates a path to cross the river.
2) This test should make the robot successfully complete bridge navigation. The angle, X and Y position error will be measured at each point in the river, i.e (SH_LL_x, SH_LL_y), (SV_LL_x, SV_LL_y) and (SV_UR_x, SV_UR_y). 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 on the first point near corner 2 (7,7), position facing -180 degrees, as it would be facing after localization. The robot receives data over wifi, the bridge coordinates (SH_LL_x, SH_LL_y), (SV_LL_x, SV_LL_y) and (SV_UR_x, SV_UR_y). Once data is acquired, the command "start" is used on the server GUI and the robot starts navigating. It first navigates to the points (SH_LL_x, SH_LL_y) and localizes. The robot proceeds to navigate to the bridge point (SV_UR_X - 1), (SV_UR_Y - 1), where it localizes, and the X, Y and angle error are measured. A button press gives it the go to reach the end of the bridge, and stops for another angle check. Finally, localization is run and angle error is measured at the end.
4) The brick should end up at the end of the bridge, and localize at the (SV_LL_x, SV_LL_y) point.
5)

Test Run #	Starting Position (x,y)	SH angle reloc. (°)	SH_LL_x (cm)	SH_LL_y (cm)	SV angle reloc. (°)	SV_UR_x (cm)	SV_UR_y (cm)	SV angle reloc. (°)	SV_LL_x (cm)	SV_LL_y (cm)
1	(7,7)	1.0	1.0	1.0	3.0	0.5	0.5	1.0	1.5	1.0

2	(7,7)	0.0	0.5	1.0	2.0	N/A	N/A	2.5	N/A	N/A
3	(7,7)	1.5	0.0	0.5	1.0	0.5	0.5	3.0	0.5	2.0
4	(7,7)	2.0	1.0	0.5	0.0	3.5	3.5	5.0	N/A	N/A
5	(7,7)	0.0	1.0	1.0	2.0	N/A	N/A	4.0	N/A	N/A
6	(7,7)	1.0	1.5	0.5	0.5	N/A	N/A	3.5	N/A	N/A
7	(7,7)	2.0	0.5	1.0	1.0	1.5	2.5	2.0	N/A	N/A
8	(7,7)	0.0	0.0	1.5	2.0	2.0	0.5	1.0	N/A	N/A
9	(7,7)	1.0	1.0	1.0	1.5	4.5	3.0	7.5	N/A	N/A
10	(7,7)	3.0	0.0	0.5	0.0	1.0	1.0	3.0	N/A	N/A

Figure 1: Bridge navigation test (positioning precision)

	X (cm)		**Y (cm)**		**Angle (°)**	
	Standard Deviation	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**	**Mean**
SH_LL	0.5	0.7	0.3	0.9	1.0	1.2
SV_UR	1.5	1.9	1.3	1.6	1.0	1.3
SV_LL	0.7	1.0	0.7	1.5	2.1	3.3

Figure 2: Standard Deviation and Mean of successful localizations

6) The brick successfully navigates to the start of the bridge 100% of the time. It also localizes with a deviation and mean of near 1, which is very satisfactory. It makes it to the next point 70% of the time, due to some angle turning imprecision. When it does localize, the values for the deviation and mean are between 1 and 2, which is still quite satisfactory. Finally, it arrives at the end of the bridge only 20% of the time, since the error accumulated throughout due to the imprecise angle turning throws it off, and it localizes to the tile next to the objective point instead of the point itself. The standard

deviation and mean when it successfully localizes is around 1 for x and y and around 2-3 for the angle, which is still considered acceptable as it is under 2 cm and 4 degrees.

7) There needs to be a fix implemented in the method that defines the path for the bridge in robotControl class, as it does not seem to travel properly after the midway point. This may be due to the turning being off, therefore the wheelbase value of the robot may also need to be modified for more precise turning.

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

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

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

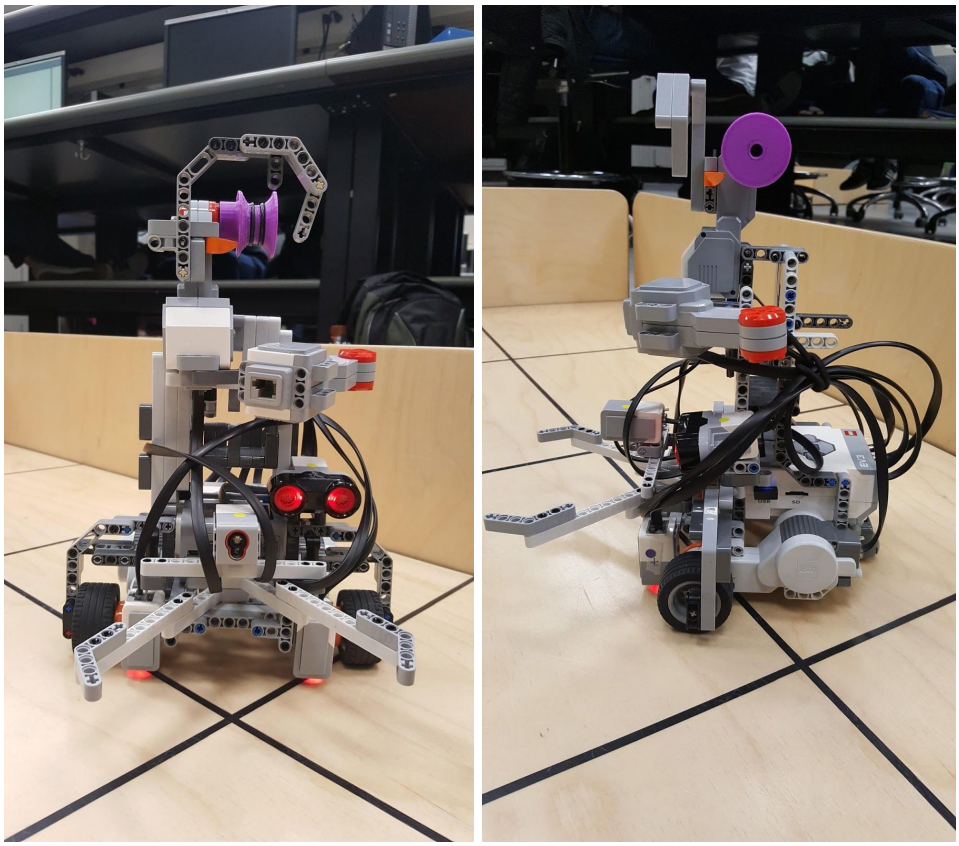
$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

$\overline{x}$ = *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: `941f1c2c2ba3f6c1ed7a9073f08fd72fd2747e2e`