TESTING DOCUMENT

Project: ECSE 211 Final Design Project – Team 6
Document Version Number: 6.2
Date: 19/10/2017
Original Author: Alex Hale

Edit History:
[19/10/2017] Alex Hale: created document
[19/10/2017] Xu Hai: Recorded the information about Test 1.
[20/10/2017] Xianyi Zhan: Recorded the information about Test 2.
[21/10/2017] Xu Hai: Recorded the information about Test 3.
[21/10/2017] Xu Hai: Recorded the information about Test 4.
[21/10/2017] Xu Hai: Recorded the information about Test 5.
[22/12/2017] Alex Hale: transferred to Google Drive for easier collaboration; made some grammatical and formatting changes
[23/10/2017] Xu Hai: Recorded the information about Test 6.
[23/10/2017] Xu Hai: Created the weekly status summary #1.
[28/10/2017] Xu Hai: Recorded the information about Test 7-10.
[29/10/2017] Xianyi Zhan: Created the weekly status summary #2.
[30/10/2017] Alex Hale: grammar touch-ups for weekly submission.
[05/11/2017] Frederic Cyr: Recorded the information about Test 11.
[05/11/2017] Xianyi Zhan: Recorded the information about Test 12-14,Created weekly summary #3.
[05/11/2017] Alex Hale: added proper references to hardware and software versions; grammar and formatting adjustments for Test 11-14
[06/11/2017] Xianyi Zhan: Recorded the information about Test 15-16.
[06/11/2017] Alex Hale: added details for testing plan; made grammar edits to Test 15-16
[06/11/2017] Xu Hai: Recorded the information about Test 17-18; added some pictures for Test 11-19.
[06/11/2017] Frederic Cyr: Recorded the information for Test 19.
[06/11/2017] Alex Hale: formatting and grammar edits for tests 16-19
[06/11/2017] Xu Hai: Updated the Testing Plan.
[12/11/2017] Xianyi Zhan: Recorded the information about Test 20.
[12/11/2017] Xu Hai: Recorded the information about Test 21-22.
[18/11/2017] Xu Hai: Recorded the information about Test 23; Updated some graphs for Test 21-23.
[18/11/2017] Xianyi Zhan: Created the weekly status summary #4 and #5.
[19/11/2017] Alex Hale: Revised tests 20-23 and weekly summaries #4 and #5
[21/11/2017] Frederic Cyr: Recorded the information for Test #24.
[23/11/2017 Alex Hale: revised test #24, expanded section 1.0

Table of Contents

1.0 Requirements

1.1 Project Requirements

See Requirements Document

1.2 Testing Requirements

- Each test should note the date, tester(s), report writer, hardware version, software version, goal, procedure, expected result, test report, conclusion, action and distribution.
- Each test should have at least 10 independent trials. Results from each trial should be recorded in a table. Compute Euclidean distance, mean value and standard deviation if possible.
- Potential weak points should be tested.
- Extreme cases of the specifications should be tested.
- The tester should have a clear expected outcome for each test.

1.3 Characteristics that Require Testing

- Test whether the hardware design keeps the robot stable when moving, ziplining, and turning (extreme cases included, e.g. high speed).
- Test whether the hardware design allows the robot to mount, traverse, and dismount the zip line successfully (extreme cases included, such as a suboptimal approach angle).
- Test whether the robot gets stuck when it finishes traversing the zip line.
- Test the precision of the localization procedure of the robot.
- Test the success rate of the robot when searching for the flag.

2.0 Testing Plan

An outline of all the project components that must be tested can be found below. Items indicated [FC] do not have to be completed until after the Beta Demonstration, but must be completed before the Final Competition (FC). All other tests must be complete at least once before the Beta Demonstration.

2.1 Hardware

- Hardware stability (Completed)
    - The robot must not tip over. The centre of gravity needs to be centralized enough that the robot does not lean to one side, affecting navigation.
- Zipline crossing (Completed, but there are some potential problems that when the robot dismounts the zipline, it always leans to left, we think it has something to do with the zipline wheel speed and the gravity, further tests should be operated later)
    - The robot must be able to mount, traverse, and dismount the zip line in a consistent, stable fashion.
- Ultrasonic sensor accuracy and stabilization (Completed)
    - The ultrasonic sensor must be accurate enough to meet the performance requirements of the robot. The mounting of the ultrasonic sensor must hold the sensor stable enough to take accurate readings.
- Colour sensor accuracy and stabilization (Completed)
    - The colour sensors must be accurate enough to meet the performance requirements of the robot. The mounting of the colour sensors must hold the sensors stable enough to take accurate readings.
- Medium motor performance (Completed, but according to the test of zipline crossing, further tests need to be operated to test the motor's speed stability)
    - The medium motor must be strong and accurate enough to pivot the front-facing colour sensor and ultrasonic sensor.
- Large motor performance (Completed)
    - The large motors must be strong and accurate enough to move the robot around the field and to move the robot across the zip line. The two motors selected to move the robot around the field must be very similar in performance so the robot moves and turns the same amount in both directions.

2.2 Software

- Navigation reliability (Completed)
    - The navigation algorithm must consistently move the robot to the required location on the field. The movements must be precise enough to allow the localization algorithm to re-localize the robot to a grid intersection.
- Localization reliability (Completed)
    - The ultrasonic localization algorithm must be accurate enough to orient the robot such that it can more accurately find the grid intersection using the light localization algorithm. The light localization algorithm must be accurate enough to

align the robot to mount the zip line. This will require an error angle of under 2 degrees. The entire localization algorithm must be fast enough to finish the initial corner localization in under 30 seconds.

- Odometer reliability (Completed)
  - The odometer must be accurate enough to let the navigation algorithm move the robot within a small margin of error of the intended destination. This margin of error must be small enough that the light localization algorithm can re-localize the robot to a grid intersection.
- Flag detection (Incomplete, software hasn't been finished)
  - The flag detection system must be able to find a flag (block) in space, move toward it, and identify its colour with the colour sensor.
- Searching efficiency (Incomplete, software hasn't been finished)
  - The searching algorithm must be efficient enough that the robot can find the flag and return to its home base within the game time limit.
- Obstacle avoidance (Incomplete, software hasn't been finished)
  - The obstacle avoidance system must be capable of detecting objects and navigating around them. Contact with objects is allowed, but undesirable, as it might cause the robot to fall over or lose its odometer calibration.
- Zipline traversal system (Competed)
  - The zipline traversal system must be capable of aligning with, mounting, traversing, and dismounting the zip line. The system must be able to detect the transitions between states.

2.3 Extra Cases

- Weaknesses from Lab 5 must be ironed out, especially aligning with the zip line (Completed)
- Localization after zipline dismount has never been used in a lab, so must be tested (Completed)
- [FC] Localization after capturing the flag has never been used in a lab, so must be tested
- [FC] Determine the level of battery depletion during a long run (maximum 7 minutes for the competition)
- [FC] Ensure that *all* software components that have been developed in isolation can be integrated successfully

3.0 Tests

<u>Test #1 - Hardware Stability</u>

Date: 19 October 2017

Tester: Xu Hai

Report Writer: Xu Hai

Hardware version: R&D Laboratory Robot (Hardware Design Document, Section 5)

Software version: N/A

Goal: Determine if the hardware design can keep stable when the robot is moving and turning.

Procedure: The robot is placed on the ground and then get instructed to move forward, backward and turn around. Each direction should also be tested with speed of 100, 200 and 300, each case is tested at least 5 times.

Expected Result: The hardware design should keep stable all the time.

Test Report: The test was performed totally 60 times following the protocol described above. During the test, the robot can keep stable at most the time (57/60). However, sometimes left motor can drop off when put the robot on the ground. In summary, the left motor dropped off on 3 of the 60 tests.

Conclusion: The hardware design can keep stable when the robot is moving and turning, but the design of left motor connection is not sufficiently stable, potential problem exists.

Action: This test report is sent to the hardware team to review the hardware design. The Gantt chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #2 - Hardware Stability II</u>
Date: 20 October 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xianyi Zhan
Hardware version: R&D Laboratory Robot (Hardware Design Document, Section 5)
Software version: N/A
Goal: Determine if the hardware design can keep stable when the robot is moving and turning. Check if the left motor dropping off problem has been solved.

Procedure: The robot is placed on the ground and then get instructed to move forward, backward and turn around. Each direction should also be tested with speed of 100, 200 and 300, each case is tested at least 5 times.

Expected Result: The hardware design should keep stable all the time without any hardware issue.

Test Report: The test was performed totally 60 times following the protocol described above. During the test, the robot can keep stable at most the time (60/60). During the entire testing process, the left motor does not dropping off.

Conclusion: The hardware design can keep stable when the robot is moving and turning, the left motor dropping off problem has been solved.

Action: This test report is sent to the project manager and documentation manager. The Gantt chart should be updated to show the project can move on to the next stage. The test team will start to testing the zipline crossing ability of the robot.

Distribution: Project management, Documentation management, Testing team.

Test #3 - Zipline Crossing
Date: 20 October 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xu Hai
Hardware version: R&D Laboratory Robot (Hardware Design Document, Section 5)
Software version: N/A
Goal: Determine if the hardware design can make the robot mount and traverse the zip line successfully.

Procedure: The robot is placed in front of the zip line and then get instructed to mount and traverse it. The test is performed with speed of 200 and 300. Each case is tested at least 10 times.

Expected Result: The hardware design can make the robot mount and traverse the zip line successfully all the time.

Test Report: The test was performed totally 20 times following the protocol described above. In summary, with the speed of 200, the robot dropped off the zip line on 1 of 10 runs. With the speed of 300, the robot dropped off the zip line on 0 of 10 runs. Notice that the supporting structure of the zip line travelling motor is unstable.

Conclusion: The robot performance did not meet the specified outcomes. The hardware design for the zip line is unreliable.

Action: This test report is sent to the hardware team to review the hardware design. The Gantt chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

Test #4 - Zip Line Crossing
Date: 20 October 2017
Testers: Xu Hai, Xianyi Zhan

Report Writer: Xu Hai
Hardware version: R&D Laboratory Robot (Hardware Design Document, Section 5)
Software version: N/A
Goal: Determine if the hardware design can make the robot mount and traverse the zip line successfully.

Procedure: The robot is placed in front of the zip line and then get instructed to mount and traverse it. The test is performed with speed of 200 and 300. Each case is tested at least 10 times.

Expected Result: The hardware design can make the robot mount and traverse the zip line successfully all the time.

Test Report: The test was performed totally 20 times following the protocol described above. In summary, the robot mounts and traverses the zip line successfully all the time.

Conclusion: The robot performance met the specified outcomes. The hardware design for the zip line crossing is reliable.

Action: This test report will be sent to the hardware team to see the result. The Gantt chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #5 - Zip Line Sticking at end of Crossing</u>
Date: 20 October 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xu Hai
Hardware version: R&D Laboratory Robot (Hardware Design Document, Section 5)
Software version: N/A
Goal: Determine if the hardware design can make the robot avoid stuck when it finishes traversing the zip line.

Procedure: The robot is placed in front of the zip line and then get instructed to mount and traverse it. The test is performed at least 10 times

Expected Result: The hardware design should make the robot avoid stuck when it finishes traversing the zip line all the time.

Test Report: The test was performed totally 10 times following the protocol described above. In summary, the robot did not get stuck during the test.

Conclusion: The robot performance met the specified outcomes. The hardware design for the zip line crossing is reliable.

Action: This test report is sent to the hardware team to see the result. The Gantt chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #6 - Pose orientation and position accuracy</u>
Date: 23 October 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xu Hai
Hardware version: R&D Laboratory Robot (Hardware Design Document, Section 5)
Software version: R&D Laboratory 5 codebase - see Lab 5 report and code submission
Goal: Determine if the robot can reach (X_o, Y_o), which in this test, is (1, 6)

Procedure: The robot is placed in corner 0, as shown in the Lab 5 instructions, at position $(a,\ a),\ 0 < a < 1$. The robot is instructed to localize and navigate to $(X_0,\ Y_0) = (1, 6)$.The test is performed at least 10 times. The data collected during the test is recorded, and the standard deviation and mean value of the Euclidean distance error is computed.

Expected Result: The robot should finish the localization successfully and navigate to (X_o, Y_o) within an error tolerance of ± 2.5 degree and ± 2 cm.

Test Report: The test was performed 10 times following the protocol described above. The results can be found in the Lab 5 report. In summary, the robot cannot reach the destination precisely.

Conclusion: The robot performance does not meet the specified outcomes. Localization and navigation are not reliable.

Action: This test report is sent to the software team to review the localization and navigation process. The Gantt chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #7 - Hardware Stability for design #2</u>
Date: 27 October 2017
Tester: Xu Hai
Report Writer: Xu Hai
Hardware version: Final Robot Draft #1 (Hardware Design Document, Section 7)

Software version: N/A
Goal: Determine if the hardware design can keep stable when the robot is moving and turning.

Procedure: The robot is placed on the ground and instructed to move forward, backward and turn around. Each direction is tested with speeds of 100, 200 and 300, and each case is tested at least 5 times.

Expected Result: The hardware design should keep stable all the time.

Test Report: The test was performed 60 times following the protocol described above. During the test, the design can keep the robot stable when it moves backward (15 runs), but it can not keep the robot in balance when it tried to move forward (10 runs) or turns around (30 runs).

Conclusion: The hardware design can keep stable when the robot is moving backward, but the design for the centre of gravity is not reliable. In addition, it seems that the wheels come in touch with the support beam above them, which may impose resistance to the rotation of wheels.

Action: This test report is sent to the hardware team to review the hardware design. The Gantt chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #8 - Hardware Stability II for design #2</u>
Date: 27 October 2017
Testers: Xu Hai
Report Writer: Xu Hai
Hardware version: Final Robot Draft #1 (Hardware Design Document, Section 7)
Software version: N/A
Goal: Determine if the hardware design can keep stable when the robot is moving and turning. Check if the centre of gravity problem has been solved.

Procedure: The robot is placed on the ground and instructed to move forward, backward and turn around. Each direction is tested with speed of 100, 200 and 300, and each case is tested at least 5 times.

Expected Result: The hardware design should keep stable all the time without any hardware issues.

Test Report: The test was performed 60 times following the protocol described above. During the test, the robot can keep stable all the time (60/60 successful tests).

Conclusion: The hardware design can keep stable when the robot is moving and turning, and the centre of gravity problem has been solved. In addition, the design near the wheels has been rebuilt: the wheels don't come in touch with the beam anymore.

Action: This test report is sent to the project manager and documentation manager. The Gantt chart should be updated to show the project can move on to the next stage.

Distribution: Project management, Documentation management.

<u>Test #9 - Zip Line Crossing I</u>
Date: 27 October 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xu Hai
Hardware version: Final Robot Draft #1 (Hardware Design Document, Section 7)
Software version: N/A
Goal: Determine if the hardware design can make the robot mount and traverse the zip line successfully.

Procedure: The robot is placed in front of the zip line and instructed to mount and traverse it. The test is performed with speeds of 200 and 300. Each case is tested at least 10 times.

Expected Result: The hardware design can make the robot mount and traverse the zip line successfully all the time.

Test Report: The test was only performed 3 times. During the test, we found that the height of the zip line arm was not high enough to mount the zip line.

Conclusion: The zip line arm was not high enough to mount the zip line. The hardware design for the zip line arm is not reliable.

Action: This test report will be sent to the hardware team to review the hardware design. The Gantt chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #10 - Zip Line Crossing II</u>
Date: 27 October 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xu Hai
Hardware version: Final Robot Draft #1 (Hardware Design Document, Section 7)
Software version: N/A

Goal: Determine if the hardware design can make the robot mount and traverse the zip line successfully.

Procedure: The robot is placed in front of the zip line and instructed to mount and traverse it. The test is performed with speeds of 200 and 300. Each case is tested at least 10 times.

Expected Result: The hardware design can make the robot mount and traverse the zip line successfully all the time.

Test Report: The test was performed 20 times following the protocol described above. In summary, the robot mounts and traverses the zip line successfully all the time.

Conclusion: The robot performance met the specified outcomes. The hardware design for the zip line crossing is reliable.

Action: This test report will be sent to the hardware team to see the result. The Gantt chart should be updated to show the revised tasks.

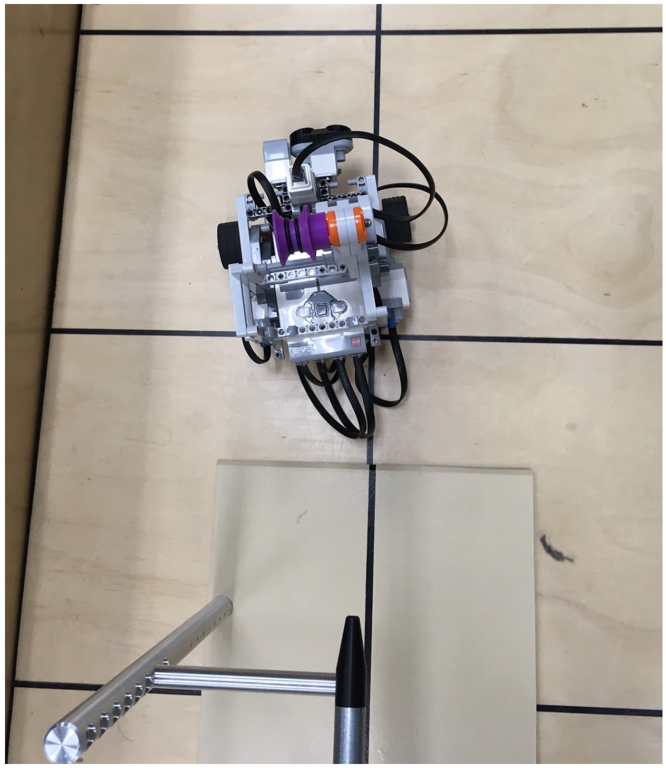
Distribution: Hardware development, project management.

<u>Test #11 - Zip Line Crossing III</u>
Date: 05 November, 2017
Testers: Xu Hai, Frederic Cyr
Report Writer: Frederic Cyr
Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)
Software version: N/A
Goal: Determine if the hardware design can make the robot mount and traverse the zip line successfully.

Procedure: The robot is placed in front of the zip line and instructed to mount and traverse it. The test is performed with speeds of 200 and 300. Each case is tested at least 10 times.

Expected Result: The hardware design can make the robot mount and traverse the zip line successfully all the time.

Test Report: The test was performed 10 times following the protocol described above. The height of the floor-facing light sensors was raised since they were colliding with the small platform placed below the zipline. In addition, when landing at the end of the zipline, the robot tends to go to the left.



Conclusion: The robot performance met the specified outcomes, except for the landing. The hardware design for the zip line crossing is reliable but the finish should be revised.

Action: This test report will be sent to the hardware team to see the result. The GANTT chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #12 - Medium Motor Accuracy</u>
Date: 05 November, 2017
Testers: Xu Hai, Xianyi Zhan, Frederic Cyr
Report Writer: Xianyi Zhan
Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)
Software version: N/A
Goal: Determine which one of three medium motors that is the most accurate during turning.

Procedure: Set the medium motor at 0 degrees (along the heading direction of the robot). Activate the medium motor and let it turn a 30 degree angle, then measure the actual turning angle. The test is performed with motor speed of 200. Each motor is tested at least 10 times.

Expected Result: The medium motor should read about 30 degree (error within 4 degrees), and the motor with the lowest standard deviation should be chosen.

Test Report: The test was performed 10 times for each motor, following the protocol described above. The data and calculations are shown in the table below.

motor No.	1st	2nd	3rd
#1	20	25.9	23.5
#2	21.1	30.4	26.5
#3	17.7	24.9	26
#4	19.9	25	25.7
#5	21.6	28.3	29.2
#6	15.4	26.7	29
#7	22.3	24.9	25.1
#8	21.7	27.2	25.4
#9	15.3	26.9	26.1
#10	26.1	26	26.7
STD.	3.292905	1.731281	1.715161
AVE.	20.11	26.62	26.32

From the test result, we can observe that the 2nd and 3rd motor both have readings within the error tolerance range. The third motor has the lowest standard deviation.

Conclusion: The third medium motor is chosen to be the component of the final robot.

Action: This test report will be sent to the hardware team to make adjustment. The GANTT chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #13 - Ultrasonic Sensor Accuracy I</u>
Date: 05 November 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xianyi Zhan
Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)
Software version: N/A
Goal: Determine which one of three ultrasonic sensors is the most accurate during measuring fixed distance.

Procedure: The robot is placed in front of the wall with distance of one grid length and two grid lengths. Record the distance obtained by the ultrasonic sensor. Perform the test 10 times for each ultrasonic sensor.

Expected Result: The ultrasonic sensors will read the distance from the wall within the error tolerance (error within 2cm). The sensor with the smallest standard deviation will be chosen.

Test Report: The test was performed 30 times total, following the protocol described above. The data and calculations are shown in the table below.

sensor No	1st sensor		2nd sensor		3rd sensor	
real distanc	30.48	60.96	30.48	60.96	30.48	60.96
#1	37	75	37	77	37	78
#2	37	75	36	76	39	80
#3	35	76	40	78	36	81
#4	37	74	37	77	40	80
#5	37	75	40	77	37	86
#6	36	75	40	77	40	80
#7	37	76	39	76	40	78
#8	37	75	38	78	39	77
#9	37	74	37	76	37	77
#10	36	75	38	77	36	76
STD.	0.699206	0.666667	1.47573	0.737865	1.66333	2.869379
AVE.	36.6	75	38.2	76.9	38.1	79.3

Reading the data, it is evident that none of the sensors can read the distance within the error tolerance. Therefore, we will compare them by their standard deviation and use software to account for this error. The second motor has the smallest standard deviation.

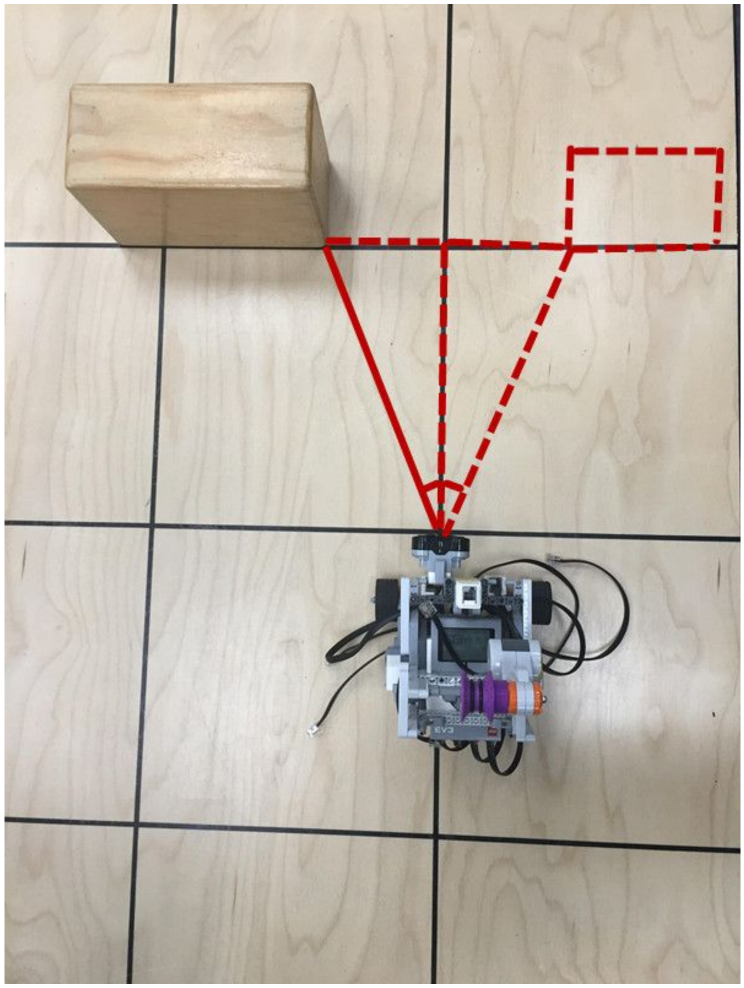
Conclusion: The second sensor is chosen to be the component for the final robot.

Action: This test report will be sent to the hardware team to make adjustment. The error will be sent to the software team to analyze and implement adjustments. The GANTT chart should be updated to show the revised tasks.

Distribution: Hardware development, Software Development, Project management.

Test #14 - Ultrasonic sensor Accuracy II
Date: 05 November 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xianyi Zhan
Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)
Software version: N/A
Goal: Using the ultrasonic sensor chosen in Test 13, find the detection angle.

Procedure: A block is placed in front of the robot at distances of one, two, and three grid lengths. Move the block horizontally until the moment that the ultrasonic sensor cannot detect the object. Perform the same test for the left and right directions. Record the maximum angle of detection. The test is performed 10 times for each grid length.



Expected Result: The value should be consistent for all the grid lengths.

Test Report: The test was performed 30 times following the protocol described above. The data and calculations are shown in the table below.

degree(1 grid away	left	right	degree(2 grids away	left	right	degree(3 grid away)	left	right
#1	21.5	23.9	#1	21.2	22.3	#1	21.2	22
#2	22.1	24.3	#2	21.3	22.1	#2	21.3	21.9
#3	21.4	24.1	#3	21.4	22.1	#3	21.8	21.6
#4	21.7	23.7	#4	21.6	22.5	#4	21.9	22.1
#5	21.2	23.8	#5	22.1	22.4	#5	22.1	21.7
AVE.	21.58	23.96	AVE.	21.52	22.28	AVE.	21.66	21.86

From the data, we can see that the sensor can detect the object within the angle range of 45 degrees, and this result is constant for each distance.

Conclusion: The test result of maximum angle of detection is valid. Record the average value for future use.

Action: This test report will be sent to the hardware team to make adjustments. The constant data will be sent to the software team for application. The GANTT chart should be updated to show the revised tasks.

Distribution: Hardware development, Software Development, Project management.

<u>Test #15 - Wheel Motor Accuracy</u>
Date: 06 November, 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xianyi Zhan
Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)
Software version: N/A
Goal: Determine which two of the six available wheel motors are the most accurate during turning.

Procedure: Set the wheel motor at 0 degrees.Turning the wheel manually by 360 degrees, clockwise. Display the reading of the tachometer and record the value. Each motor is tested at least 10 times.

Expected Result: The wheel motors should read 360 degrees (error margin within 2 degrees), and the motors with the lowest standard deviation should be chosen.

Test Report: The test was performed 10 times for each motor, following the protocol described above. The data and calculations are shown in the table below.

Motor No.	1	2	3	4	5	6
#1	360.2	360.7	361.6	362.8	362	361.4
#2	368.7	360.6	362	362.3	361	360.1
#3	363.8	361.1	360.9	362.7	363	362.3
#4	365	359.8	361.7	362.1	361.2	362.5
#5	364.4	360.3	361.5	363.3	359.6	360.5
#6	365.8	360.2	360.7	362.4	362.2	362
#7	365.3	360.1	361.3	362.4	360.7	361.9
#8	364.9	360.2	360.8	362.7	361.1	361.6
#9	364.2	360.1	361.2	363.1	360.2	360.9
#10	365.4	360.3	361.1	362.5	361.3	362.1
STD.	2.096054	0.368782	0.42111	0.36833	0.985506	0.798679
AVE.	364.77	360.34	361.28	362.63	361.23	361.53

From the test results, we can observe that the 2nd, 3rd, 5th and 6th motor have readings within the error tolerance range. The 2nd and 3rd motors have the lowest standard deviation.

Conclusion: The second and third wheel motor are chosen to be the components of the final robot.

Action: This test report will be sent to the hardware team to make use of the proper motors. The GANTT chart should be updated to show the revised tasks.

Distribution: Hardware development, project management.

<u>Test #16 - Zipline Motor Accuracy</u>
Date: 06 November, 2017
Testers: Xu Hai, Xianyi Zhan
Report Writer: Xianyi Zhan
Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)
Software version: N/A
Goal: Determine which one of the six available zipline motors is the most accurate during turning.

Procedure: Set the zipline motor at 0 degrees.Turning the zipline wheel manually by 360 degrees, clockwise. Display the reading of the tachometer and record the value. Each motor is tested at least 10 times.

Expected Result: The zipline motors should read about 360 degrees (error margin within 2 degrees), and the motor with the lowest standard deviation should be chosen.

Test Report: The test was performed 10 times for each motor, following the protocol described above. The data and calculations are shown in the table below.

Motor No.	1	2	3	4	5	6
#1	362.1	362.4	360.9	361.2	361.1	361.1
#2	361.4	362.2	360.4	362.5	361.4	361.3
#3	361	362.3	361.4	362.3	360.8	361.1
#4	361.6	361.8	360.6	360.7	360.5	360.7
#5	362.4	362.4	360.3	361.4	361.2	360.6
#6	362.1	361.7	361.1	362.2	361.1	362.2
#7	361.3	362.3	360.6	360.9	360.4	361.3
#8	361.6	362.1	360.2	361.3	361.6	360.9
#9	361.3	361.9	360.3	362.2	360.6	361.2
#10	361.8	362.2	360.7	360.6	360.7	360.8
STD.	0.437671	0.249666	0.38658	0.711883	0.400555	0.451664
AVE.	361.66	362.13	360.65	361.53	360.94	361.12

From the test results, we can see that the 1st, 3rd, 4th,5th and 6th motor have readings within the error tolerance range. The 3rd motor has the lowest standard deviation among these 5 motors.

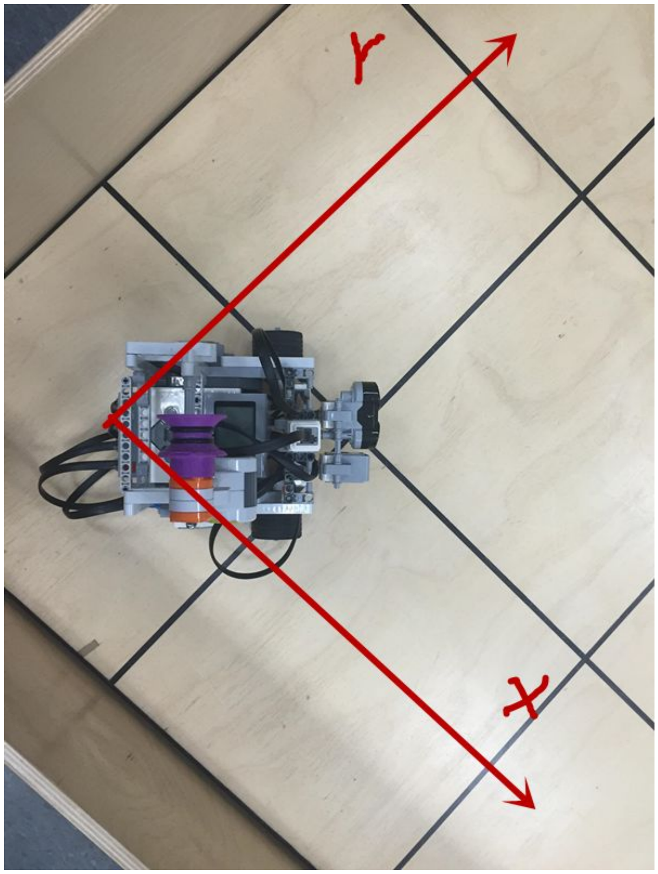
Conclusion: The third zipline motor is chosen to be the component of the final robot.

Action: This test report will be sent to the hardware team to implement the correct motor. The GANTT chart should be updated to show the completed tasks.

Distribution: Hardware development, project management.

<u>Test #17 - Localization Reliability</u>
Date: 06 November, 2017
Testers: Xu Hai, Justin Tremblay
Report Writer: Xu Hai
Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)
Software version: Final Project Draft #1 (see "Final_Project_Draft_1.zip")
Goal: Determine the reliability of localization.

Procedure: The procedure here is similar to the test in Lab 4: place the robot in a corner, with the centre of rotation on top of the 45 degree line and a random orientation (see the picture below). Run the localization software (we used falling edge in this test). Note the angle of the robot after ultrasonic localization and compute the Euclidean distance error and final angle error. The mean value and the standard deviation should be computed.



Expected Result: All the error measurements, mean values, and standard deviation calculations should be close to 0.

Test Report: The test was performed 10 times following the protocol described above. The data and calculations are shown in the table below.

The original point is (0, 0)

	Ultrasonic localization			Light localization					
Trials	**Expected angle (degree)**	**Actual angle (degree)**	**Error angle (degree)**	**X_F (cm)**	**Y_F (cm)**	**Euclidean distance error**	**Actual angle (degree)**	**Expected angle (degree)**	**Error angle (degree)**
1	0	-1	-1	0.2	0	0.20	-1.9	0	-1.9
2	0	-0.7	-0.7	-1.8	-0.3	1.83	0.3	0	0.3
3	0	-1.1	-1.1	-1	-0.5	1.12	-1.5	0	-1.5
4	0	-0.9	-0.9	0	0	0.00	-1	0	-1
5	0	0.1	0.1	0.1	-0.2	0.22	-0.9	0	-0.9
6	0	-0.5	-0.5	0.16	-0.2	0.26	-1	0	-1
7	0	-0.6	-0.6	-1.5	-0.7	1.66	-0.6	0	-0.6
8	0	-0.7	-0.7	-1	-2	2.24	0.1	0	0.1
9	0	-1.2	-1.2	1.1	-0.4	1.17	-0.4	0	-0.4
10	0	-0.7	-0.7	-0.2	0.1	0.22	0.3	0	0.3
Mean	0	-0.73	-0.73	-0.394	-0.42	0.892	-0.66	0	-0.66
Standard deviation	0	0.35	0.35	0.85	0.58	0.77	0.71	0	0.71

All the angle errors are smaller than 5 degrees, and nearly all the Euclidean distance errors (9 out of 10) are smaller than 2 cm. The standard deviations of all the measurements are small enough.

Conclusion: Localization is reliable.

Action: This test report will be sent to the software team. The GANTT chart should be updated to show the completed tasks.

Distribution: Software development, project management.

<u>Test #18 - Odometer Reliability</u>

Date: 06 November, 2017

Testers: Xu Hai, Justin Tremblay

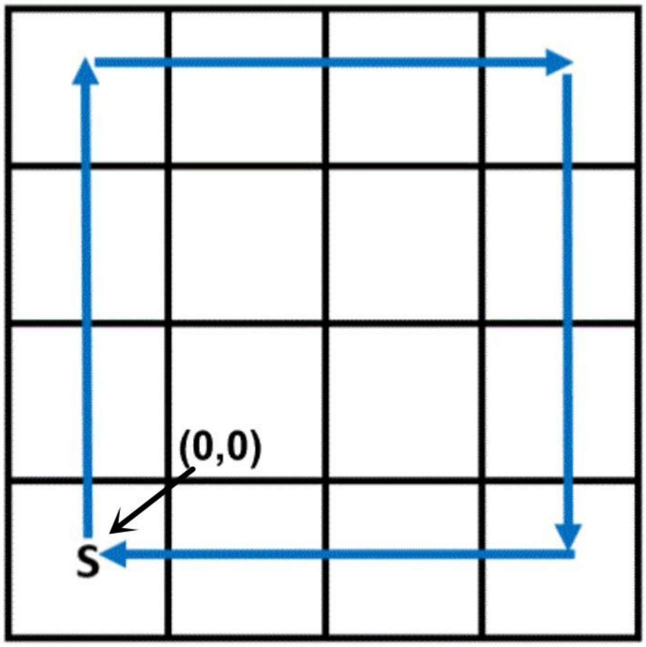
Report Writer: Xu Hai

Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)

Software version: Final Project Draft #1 (see “Final_Project_Draft_1.zip”)

Goal: Determine the reliability of the odometer.

Procedure: The procedure here is similar to the test in Lab 2. Set the robot at (0,0) and run the robot in a 3-by-3 tile square (shown in the diagram below). Measure its resulting signed (X_f, Y_f) position with respect to the starting position. Note the reported values of X and Y shown on the odometer. Compute the Euclidean error distance of the position for each test, as well as the mean value and the standard deviation.



Expected Result: The Euclidean distance error should be as small as possible, especially smaller than 2. The standard deviation should also be as small as possible.

Test Report: The test was performed 10 times following the protocol described above. The data

	Resulting signed values (cm)		Reported values (cm)	
Trial	**X_F**	**Y_F**	**X**	**Y**
1	-1.2	-1.7	0.97	-1.1
2	1.1	-2.1	0.86	-1.24
3	-0.6	-0.8	1.32	0.10
4	-0.10	-1.4	-1.38	-0.93
5	-0.6	-1.1	0.74	0.40
6	-2.2	-2.2	-1.32	-1.21
7	-1.8	-0.4	-0.76	1.35
8	-0.2	-0.6	-1.21	1.21
9	-0.60	-1.3	-1.12	-0.65
10	-1.30	-1.2	-0.87	-0.07

and calculations are shown in the table below.

	Odometer tests		
Trial	**X (cm)**	**Y (cm)**	**Euclidean error distance**
1	0.97	-1.1	2.25
2	0.86	-1.24	0.89
3	1.32	0.10	2.12
4	-1.38	-0.93	1.36
5	0.74	0.40	2.01
6	-1.32	-1.21	1.32
7	-0.76	1.35	2.04
8	-1.21	1.21	2.07
9	-1.12	-0.65	0.83
10	-0.87	-0.07	1.21
Mean	-0.277	-0.214	1.61
Standard deviation	1.04	0.922	0.5157

Although the mean value of the Euclidean error distance is 1.61 (smaller than 2), half of the trials produced Euclidian error distances above 2.

Conclusion: The performance of odometer is good on average, but consistency should be improved.

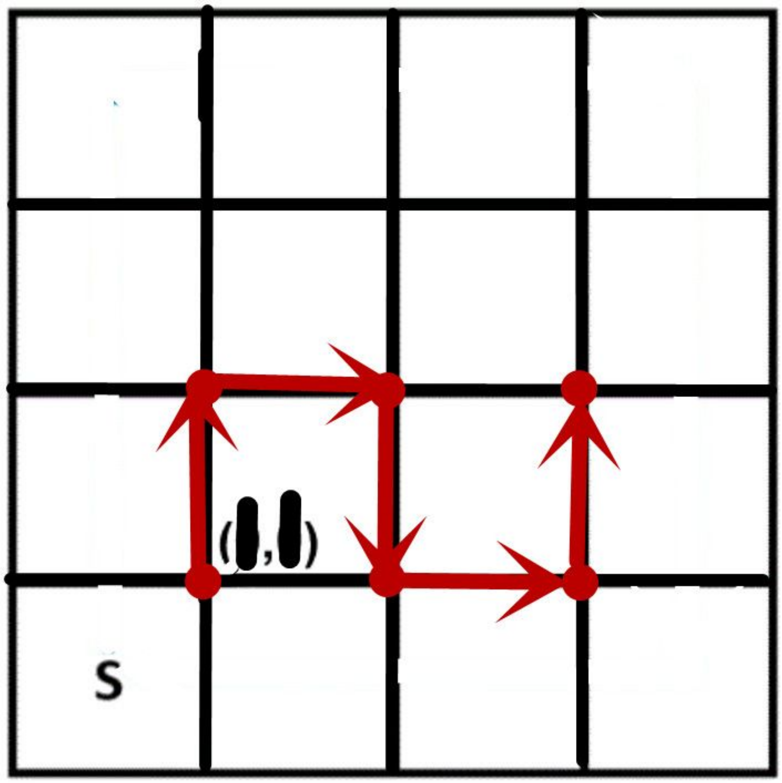
Action: This test report will be sent to the software team. The GANTT chart should be updated to show the completed tasks.

Distribution: Software development, project management.

Test #19 - Navigation Reliability
Date: 06 November, 2017
Testers: Xu Hai, Justin Tremblay
Report Writer: Frederic Cyr
Hardware version: Final Robot Draft #2 (Hardware Design Document, Section 8)
Software version: Final Project Draft #1 (see “Final_Project_Draft_1.zip”)

Goal: Determine the reliability of the navigation algorithm.

Procedure: Starting from the (1,1) point on the 4x4 tile grid, the robot must travel through 5 waypoints. The robot must stop at each of these following waypoints: (1,2), (2,2), (2,1), (3,1), and (3,2). The robot should finally stop at the last waypoint, which has the coordinates of (90 cm, 60 cm). The values displayed on the odometer and the actual distance from the starting position should be measured.



Expected Result: The final point should be very close to the expect coordinate point, (90.0 cm, 60.0 cm). The difference between the measured value and the value displayed on the odometer should be very low. All the standard deviation calculations should be close to 0.

Test Report: The test was performed a total of 10 times for the same navigation path, following the protocol described above. The data and calculations are shown in the table below.

Trial #	Measured final X position (cm)	Measured final Y position (cm)	Final X position on Odometer (cm)	Final Y position on Odometer (cm)
1	89.44	60.26	91.35	59.86
2	89.94	60.06	91.36	59.96
3	89.84	60.36	91.36	59.91
4	89.84	60.26	91.35	59.94

5	89.94	60.06	91.42	59.96
6	90.24	59.96	91.33	59.94
7	90.14	60.36	91.31	59.89
8	89.54	60.06	91.31	59.98
9	90.14	60.26	91.37	59.96
10	89.54	59.96	91.33	59.95
Mean	89.86	59.96	91.35	59.94
Standard deviation	0.278	0.156	0.0325	0.0372

From the test results, we can observe that the difference between the measured final position and the distance shown on the odometer is within 2 centimeters. However, the standard deviation is quite accurate, especially for the position displayed on the odometry.

Conclusion: Here, we can observe that the consistency is better than the accuracy, meaning improvements can be made for future tests.

Action: This test report will be sent to the software team. The GANTT chart should be updated to show the completed tasks.

Distribution: Software development, project management.

<u>Test #20 - Odometer Reliability II</u>

Date: 12 November, 2017

Testers: Xu Hai, Xianyi Zhan

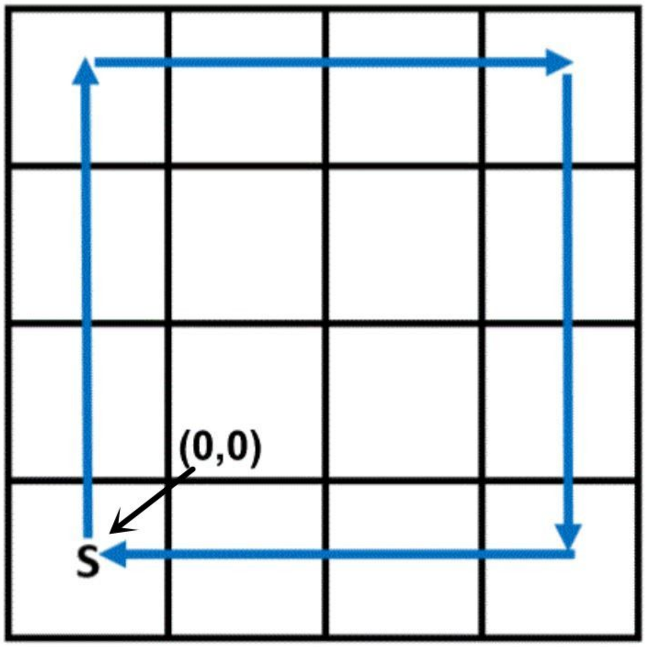
Report Writer: Xianyi Zhan

Hardware version: Final Robot Draft #3 (Hardware Design Document, Section 9)

Software version: Final Project Draft #2 (see “Final_Project_Draft_2.zip”)

Goal: Determine the reliability of the odometer using on the new wheelbase parameter.

Procedure: Similar to Test #18. Set the robot at (0,0) and run the robot in a 3-by-3 tile square (shown in the diagram below). Measure its resulting signed (X_f, Y_f) position and angle theta with respect to the starting position. Note the reported values of X, Y and theta shown on the odometer. Compute the Euclidean error distance of the position for each test, as well as the mean value and the standard deviation.



Expected Result: The Euclidean distance error should be as small as possible, ideally smaller than 2 cm. The angle error should be less than 10 degrees. The standard deviation should also be as small as possible.

Test Report: The test was performed 10 times following the protocol described above. The data and calculations are shown in the table below.

trial	x actual	y actual	x shown	y shown	Xf	Yf	Error	actual	shown	difference
1	0	1.1	1.17	-1.25	1.17	0.15	1.18	0	356.89	3.11
2	0.5	0.7	2.01	-2.51	1.51	1.81	2.36	0	354.46	5.54
3	1.2	0	3.16	-1.72	1.96	1.72	2.61	1	353.97	5.03
4	0.4	0.8	1	2.01	0.6	-2.81	2.87	1	354.21	4.79
5	1.6	1.1	3.04	-3.12	1.44	2.02	2.48	2	352.91	5.09
6	0.5	2	2.55	-2.68	2.05	0.68	2.16	1	356.22	2.78
7	1.3	1.6	2.81	-3.01	1.51	1.41	2.07	1	355.88	3.12
8	1.1	1.7	2.31	-3.02	1.21	1.32	1.79	0	356.93	3.07
9	1.4	1.5	1.83	-2.97	0.43	1.47	1.53	0	356.25	3.75
10	1.1	1.3	1.34	-2.31	0.24	1.01	1.04	1	356.14	3.86
						AVE.	2.01		AVE.	4.014
						STD.	0.61		STD.	4.1044

The mean and standard deviation values of x,y error are improved and, in general, the results are more accurate and consistent. The mean angle error is within the tolerance range, but the standard deviation is too large. In general, the measured actual angles are very precise, within 2 degrees.

Conclusion: The performance of the odometer is better on average, with improved consistency. Angle error is acceptable, but the consistency should be improved. The software team will be consulted to determine whether to keep the original width parameter or use the new parameter.

Action: This test report will be sent to the software team. The GANTT chart will be updated to show the completed tasks.

Distribution: Software development, project management.

<u>Test #21 - Integration Test for Beta Demo</u>

Date: 12 November, 2017

Testers: Xu Hai, Xianyi Zhan, Frederic Cyr

Report Writer: Xu Hai

Hardware version: Final Robot Draft #3 (Hardware Design Document, Section 9)

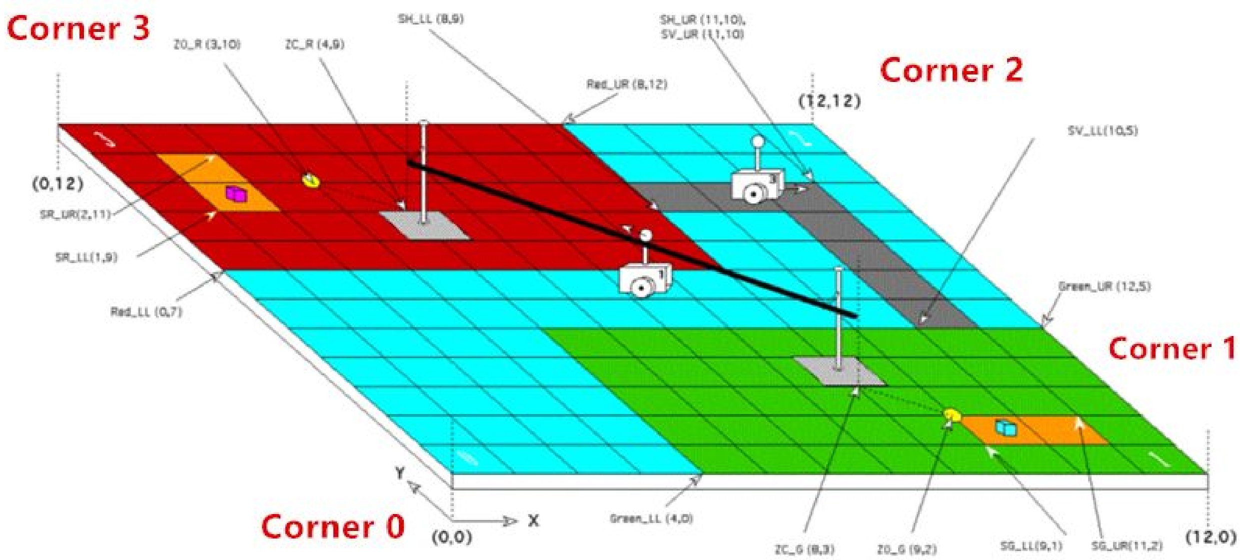
Software version: Final Project Draft #2 (see “Final_Project_Draft_2.zip”)

Goal: Determine the reliability of the robot ahead of the for beta demo.

Procedure: According to the instructions of the beta demo, set the robot in an 8-by-8 beta demo floor as designed by a professor or TA. The beta demo floor is similar to the final competition floor, with a smaller size. Use the final competition floor as a reference (shown in the diagram below). Set Z0_G to be (1,5), ZC_G to be (2,5), ZC_R to be (6,5), Z0_R to be (7,5), SR_LL to be (7,4) and SR_UR to be (6,2). Place the robot in a corner (0 to 3), and set the robot to be the green team. The robot is turned on and cannot be touched again until the end of the run. Use the server to send the robot the information shown in the diagram via the wifi link. The robot will perform the following:

1. Localize
2. Navigate to the zip line
3. Mount the zip line
4. Cross the zip line and dismount
5. Localize
6. Navigate to a specified location (the opponent’s flag zone)
7. Stop when it arrives at the location

Each corner is tested 5 times.



Expected Result: The robot should finish each part of the procedure successfully. The Euclidean distance error at the stopping point should be less than 2 cm.

Test Report: The test was performed 20 times following the protocol described above. The data and calculations are shown in the table below (rotated sideways).

	starting coner	localizaition	error angle	X error	Y error	EU error	navigation to p1	error angle	X error	Y error	EU error	mounting	demounting	localization 2	error angle	X error	Y error	EU error
trial 1	0		1.3	0.9	0.7	1.1402		1.8	0.7	0.5	0.86023	Yes	Yes		1.6	1.1	0.3	1.14018
trial 2	0		2.3	1.1	1.1	1.5556		1.6	1.3	1.5	1.98494	Yes	Yes		2.4	1.3	0.4	1.36015
trial 3	0		1.7	0.7	0.4	0.8062		2.7	0.9	1.1	1.42127	Yes	Yes		1.7	0.9	0.6	1.08167
trial 4	0		4.5	1.1	0.4	1.1705		2.3	1.3	0.7	1.47648	Yes	Yes		0.9	1.1	0.3	1.14018
trial 5	0		2.7	0.8	1	1.2806		1.2	1.4	1.2	1.84391	Yes	Yes		1.6	1.5	0.7	1.65529
trial 6	1		3.8	1.5	1.2	1.9209		2.1	0.9	1.1	1.42127	Yes	Yes		2.1	1.2	0.4	1.26491
trial 7	1		2	1.3	0.8	1.5264		2.3	1.5	0.7	1.65529	Yes	Yes		2	1.1	0.6	1.253
trial 8	1		3	0.9	0.8	1.2042		1.7	1	1	1.41421	Yes	Yes		2.5	1.4	0.4	1.45602
trial 9	1		3	1.2	1.2	1.6971		0.6	1.3	1.4	1.9105	Yes	Yes		1.8	1.9	0.8	2.06155
trial 10	1		3.7	0.9	0.9	1.2728		2.1	1.4	1.3	1.9105	Yes	Yes		1.5	2.1	0.4	2.13776
trial 11	2		1.9	0.5	0.3	0.5831		2.7	1.3	2.1	2.46982	Yes	Yes		2.7	1.2	0.7	1.38924
trial 12	2		2.3	0.6	0.7	0.922		1.8	1.5	1.3	1.98494	Yes	Yes		2.1	1.3	0.8	1.52643
trial 13	2		2.7	0.8	0.1	0.8062		1.1	1.6	1.4	2.12603	Yes	Yes		1.7	1.3	0.5	1.39284
trial 14	2		3.2	0.5	0.7	0.8602		1.8	1.4	1.5	2.05183	Yes	Yes		2.3	1.2	1.1	1.62788
trial 15	2		2	1.1	0.6	1.253		1.7	0.9	1.3	1.58114	Yes	Yes		0.7	1.8	1.1	2.1095
trial 16	3		2.2	0.8	0.7	1.063		2.1	1.5	0.9	1.74929	Yes	Yes		1.8	2.2	1.3	2.55539
trial 17	3		3	1.1	0.6	1.253		0.7	1.9	1.3	2.30217	Yes	Yes		1.4	1.2	0.9	1.5
trial 18	3		2	0.8	0.6	1		1.2	1.3	1.3	1.83848	Yes	Yes		1.8	2.3	0.7	2.40416
trial 19	3		2.5	0.8	0.7	1.063		2.6	1.4	1.6	2.12603	Yes	Yes		1.3	2.5	1	2.69258
trial 20	3		3	0.9	0.5	1.0296		1.9	1.6	1.3	2.06155	Yes	Yes		1.3	2.7	1.7	3.19061

According to the results, the robot finishes each part of the process successfully from each starting corner. However, the average Euclidean distance error starting from corners 2 and 3 is greater than 2 cm, which does not meet the test criteria.

Conclusion: The performance of the robot is good in general. It can finish each part of the demo successfully. Although the Euclidean distance error starting from corner 2 and 3 is too large, improvements can be made in the future, and the results are good enough to finish the beta demo.

Action: This test report will be sent to the software team. The GANTT chart should be updated to show the completed tasks.

Distribution: Software development, project management.

Test #22 - Reliability of the Localization After the Zipline Dismount

Date: 12 November, 2017

Testers: Xu Hai, Xianyi Zhan, Frederic Cyr

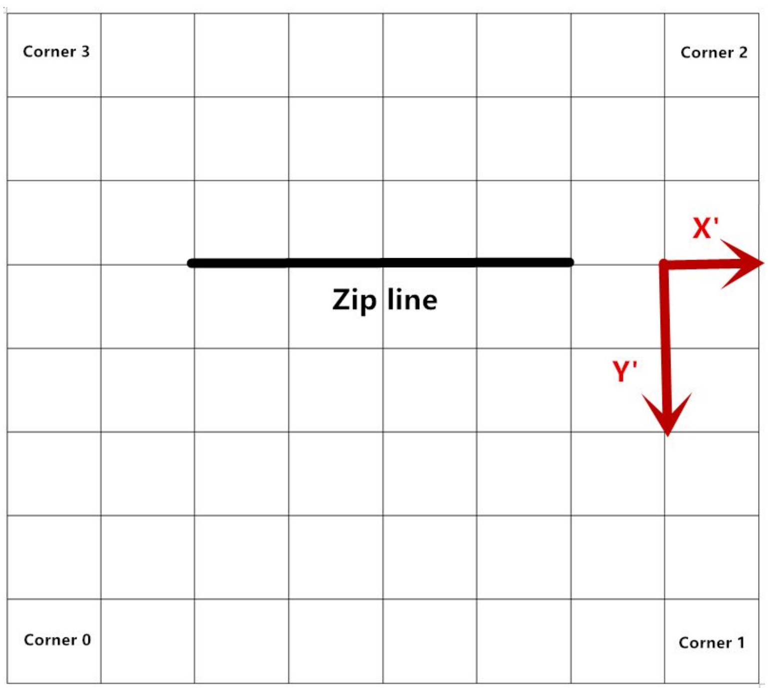
Report Writer: Xu Hai

Hardware version: Final Robot Draft #3 (Hardware Design Document, Section 9)

Software version: Final Project Draft #2 (see “Final_Project_Draft_2.zip”)

Goal: Determine the reliability of localization after the zipline dismount.

Procedure: According to the instructions of the beta demo, set the robot in a 8-by-8 competition floor as designed by a professor or TA. Place the robot in a corner (0 to 3) and the robot to the green team. The robot is turned on and cannot be touched until the end of the run. Use the server to send the robot the game board information via the wifi link. After the robot dismounts from the zipline, it will relocalize. Record the performance of navigation after dismount.



Expected Result: All the error measurements, mean values, and standard deviation calculations should be close to 0. Error angle should be smaller than 5 degrees. The Euclidean distance error should be smaller than 2 cm.

Test Report: The test was performed 20 times following the protocol described above. The data and calculations are shown in the table below.

	X_F (cm)	Y_F (cm)	Euclidean distance error	Actual angle (degree)	Expected angle (degree)	Error angle (degree)
corner 0	1.1	0.3	1.1402	1.6	0	1.6
corner 0	1.3	0.4	1.3602	2.4	0	2.4
corner 0	0.9	0.6	1.0817	1.7	0	1.7
corner 0	1.1	0.3	1.1402	0.9	0	0.9
corner 0	1.5	0.7	1.6553	1.6	0	1.6
corner 1	1.2	0.4	1.2649	2.1	0	2.1
corner 1	1.1	0.6	1.253	2	0	2
corner 1	1.4	0.4	1.456	2.5	0	2.5
corner 1	1.9	0.8	2.0616	1.8	0	1.8
corner 1	2.1	0.4	2.1378	1.5	0	1.5
corner 2	1.2	0.7	1.3892	2.7	0	2.7
corner 2	1.3	0.8	1.5264	2.1	0	2.1
corner 2	1.3	0.5	1.3928	1.7	0	1.7
corner 2	1.2	1.1	1.6279	2.3	0	2.3
corner 2	1.8	1.1	2.1095	0.7	0	0.7
corner 3	2.2	1.3	2.5554	1.8	0	1.8
corner 3	1.2	0.9	1.5	1.4	0	1.4
corner 3	2.3	0.7	2.4042	1.8	0	1.8
corner 3	2.5	1	2.6926	1.3	0	1.3
corner 3	2.7	1.7	3.1906	1.3	0	1.3
Mean	1.565	0.735	1.7469	1.76	0	1.76
Standard deviation	0.534	0.366	0.5948	0.511345	0	0.511

According to the results, all error angles are smaller than 5 degrees and the Euclidean distance error when starting in corners 0 and 1 are smaller than 2 cm. The Euclidean distance errors when starting in corners 2 and 3 are sometimes bigger than 2 cm. The standard deviations of all data are smaller than 0.6.

Conclusion: The performance of post-zipline localization when starting in corners 0 and 1 are good enough, while the performances when starting in corners 2 and 3 can be improved. The localization after dismount is reliable, but further improvement should better be carried out.

Action: This test report will be sent to the software team. The GANTT chart will be updated to show the completed tasks.

Distribution: Software development, project management.

<u>Test #23 - Integration Test for the Improvements From Beta Demo</u>

Date: 16 November, 2017

Testers: Xu Hai, Xianyi Zhan, Justin Tremblay

Report Writer: Xu Hai

Hardware version: Final Robot Draft #3 (Hardware Design Document, Section 9)

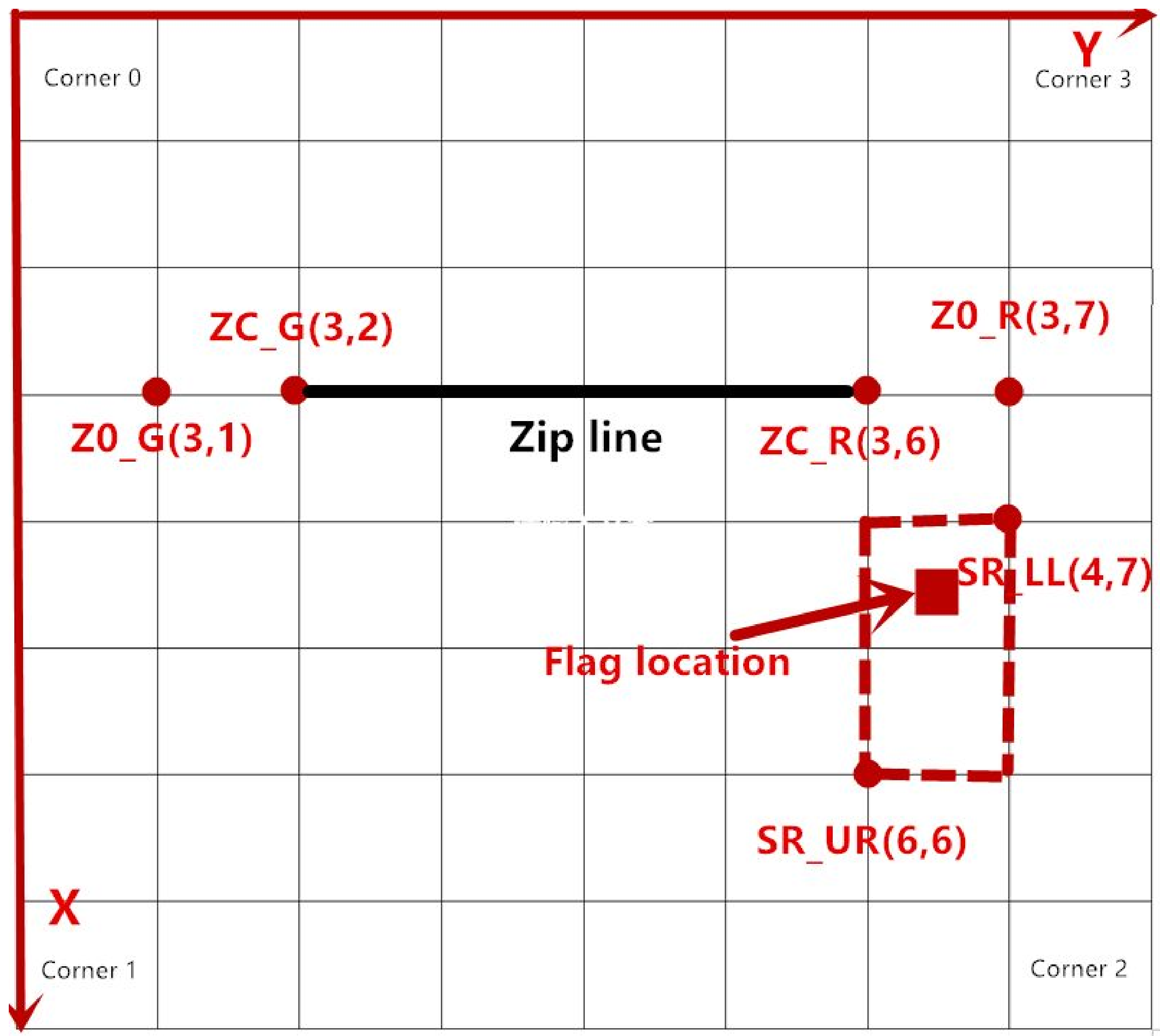
Software version: Final Project Draft #3 (see “Final_Project_Draft_3.zip”)

Goal: Test for the improvements from the beta demo.

Procedure: According to the instructions of the beta demo, set the robot in an 8-by-8 beta demo according to the diagram below. Place the robot in a corner (0 to 3), and assign the robot to the green team. The robot is turned on and cannot be touched again until the end of the run. Use the server to send the robot the information shown in the diagram via the wifi link. The robot will perform the following:

1. Localize
2. Navigate to the zip line
3. Mount the zip line
4. Cross the zip line and dismount
5. Localize
6. Navigate to a specified location (the opponent’s flag zone)
7. Stop when it arrives at the location

Each corner is tested 5 times. The time taken to complete the run is recorded.



Expected Result: The robot should finish each part successfully. The Euclidean distance error for the navigation and localization should be smaller than 2 cm. The remaining time should be longer than 2.5 min.

Test Report: The test was performed 20 times following the protocol described above. The data and calculations are shown in the table below (rotated).

	starting coner	localizaition	error angle	X error	Y error	EU error	navigation to p1	error angle	X error	Y error	EU error	mounting	demounting	localization 2	error angle	X error	Y error	EU error	navigation to area	time remain
trial 1	0		2	0	0.5	0.5		0	0.5	0.5	0.70711	Yes	Yes		4	1.2	1	1.56205	Yes	2:20
trial 2	0		4	0.3	1.1	1.1402		2	1.5	1.3	1.98494	Yes	Yes		2	1.5	1.2	1.92094	Yes	2:18
trial 3	0		2	1.2	1.1	1.6279		2	0.8	0.4	0.89443	Yes	Yes		2	1.2	1	1.56205	Yes	2:24
trial 4	0		3	1.1	0.5	1.2083		2.5	0.7	1.1	1.30384	Yes	Yes		1	1.4	1.3	1.9105	Yes	2:30
trial 5	0		3	1.2	0.7	1.3892		1.5	1	0.5	1.11803	Yes	Yes		1.5	1.3	0.7	1.47648	Yes	2:26
trial 6	1		4	1.3	0.9	1.5811		2	0.6	1	1.16619	Yes	Yes		1	0	1	1	Yes	2:26
trial 7	1		2.5	1.2	1	1.562		1.5	1	0.9	1.34536	Yes	Yes		2	1.1	1.3	1.70294	Yes	2:30
trial 8	1		3	1.2	0.8	1.4422		1	1.2	1	1.56205	Yes	Yes		2	1.2	1.3	1.76918	Yes	2:26
trial 9	1		2.5	0.9	1.2	1.5		0	1.1	1.3	1.70294	Yes	Yes		2.5	1.5	1.3	1.98494	Yes	2:20
trial 10	1		3.5	0.6	0.9	1.0817		1	1.4	1.2	1.84391	Yes	Yes		1.5	1.2	1.6	2	Yes	2:29
trial 11	2		2	1.2	0	1.2		2.5	0.5	2.5	2.54951	Yes	Yes		2.5	1.4	1.3	1.9105	Yes	1:48
trial 12	2		3	0.8	0.5	0.9434		2	1.7	1.4	2.20227	Yes	Yes		1.5	1.7	1.5	2.26716	Yes	1:55
trial 13	2		2.5	1.4	0.2	1.4142		12	12	14	18.4391	No	N/A		N/A	N/A	N/A	N/A	N/A	N/A
trial 14	2		3	1.1	0.7	1.3038		3	1.3	1.5	1.98494	Yes	Yes		1.5	1	1	1.41421	Yes	1.46
trial 15	2		2	0.5	0.7	0.8602		1.5	0.9	2.6	2.75136	No	N/A		N/A	N/A	N/A	N/A	N/A	N/A
trial 16	3		2.5	0.8	0.7	1.063		2.5	2.1	0.9	2.28473	Yes	Yes		2	2	1.3	2.38537	Yes	1:50
trial 17	3		2.5	1.1	0.6	1.253		1.5	1.8	1.2	2.16333	Yes	Yes		2.5	1.6	1.8	2.40832	Yes	1:48
trial 18	3		2	0.9	0.7	1.1402		1	1.4	1.3	1.9105	Yes	Yes		2.2	3	1.1	3.19531	Yes	1:56
trial 19	3		3	0.8	0.6	1		3	1.3	1.7	2.14009	Yes	Yes		1.4	2.7	1	2.87924	Yes	1:44
trial 20	3		3	0.7	1.2	1.3892		2.5	2.1	1.3	2.46982	Yes	Yes		1.3	2.5	1.2	2.77308	Yes	1:45

According to the results, when the robot starts from corner 0 and 1, each part of the procedure is always finished successfully, the Euclidean distance errors is smaller than 2, and the remaining time is close to 2.5 min. However, there the performances when starting from corner 2 and 3 is less satisfying. The robot sometimes fails to mount the zip line when it starts from corner 2 (2 out of 5 trials). When the robot finishes the procedure successfully, the remaining time is shorter than 2 min, and the Euclidean distance errors are bigger than 2cm. Although the robot finishes each part of the procedure when starting from corner 3, the remaining time is smaller than 2 min, and the Euclidean distance errors are bigger than 2 cm.

Conclusion: The performances of the robot starting from corner 0 and 1 are good. Although the case in corner 2 and 3 are not so meaningful for the final competition, since when starting from these corners the robot will be assigned to the red team and will cross the bridge first, improvement should be carried out. In addition, the speed of the robot should be increased if possible since the time limitation of the final competition is 5 min, and arrival at the searching zone roughly marks the halfway point.

Action: This test report will be sent to the software team. The GANTT chart will be updated to show the completed tasks.

Distribution: Software development, project management.

<u>Test #24 - Zipline Wheel</u>
Date: 21 November, 2017
Testers: Xu Hai, Frederic Cyr
Report Writer: Frederic Cyr
Hardware version: Final Robot Draft #3 (Hardware Design Document, Section 9)
Software version: Final Project Draft #3 (see “Final_Project_Draft_3.zip”)

Goal: Using the revised zip line wheel, determine the maximum angle of approach to the zipline available such that the robot is able to mount the zip line.

Procedure: Start with the robot aligned with the zip line (0°) and a motor speed of 300. Increase the angle of incidence with the zip line by 1° at a time in one direction until the robot fails to mount the zip line. Decrease the angle of incidence by 0.5° and see if the robot can mount the zip line. Record the highest angle of incidence where the robot was still able to mount the zip line. Repeat the process in the other direction.

Expected Result: There should be a larger margin of error when the robot approaches the zip line moving from the left to the right of the zip line start point. This is because the extended funnel on the left side of the zipline wheel is larger.

Test Report: The test was performed on each side of the robot until it failed to mount the zipline. The last angle registered in the table is the maximum margin of error in degrees.

Trial #	Leftward error angle (degrees)	Succeeded to mount the zipline?	Rightward error angle (degrees)	Succeeded to mount the zipline?
1	1.0	YES	3.0	YES
2	2.0	YES	4.0	YES
3	3.0	YES	5.0	NO
4	4.0	NO	4.5	YES
5	3.5	YES		

The largest leftward angle error where the robot can mount the zip line is 3.5 degrees, whereas the largest rightward angle error where the robot can mount the zip line is 4.5 degrees.

Conclusion: Adding plastic funnels to the zipline wheel increases the robot's chances of successfully mounting and traversing the zipline.

Action: This test report will be sent to the software team. The GANTT chart should be updated to show the completed tasks.

Distribution: Software development, project management.

4.0 Weekly Status Summary

Weekly Status Summary #1

After analyzing the requirements of Lab 5, we decided that the robot for the final project will be heavily based off of the robot from Lab 5. To ensure that our final project robot is reliable, we needed to built a reliable Lab 5 robot, and to accomplish that goal, we conducted a wide variety of testing. By the end of the week, the hardware and software components of the Lab 5 robot were completely tested. The hardware design is adequately reliable, but the robot did not perform as well in the software tests. It seems that there are errors in the odometry and navigation systems. Next week, we should consider fixing our software issues and confirming that our software is performing correctly with more testing.

Weekly Status Summary #2

According to the results from Lab 5, our localization was not accurate enough. To resolve this, we decided to change our method of localization - therefore, the hardware design needed to be rebuilt. By the end of this week, tests for the entirety of the basic hardware design were finished. Next, we will test motors and sensors, rank them by performance, and select the optimal components for the final robot. Since the WiFi code was provided this week, tests of WiFi connection and transfer capabilities will be conducted next week. In addition, the values used in software (e.g. wheel base and wheel radius constants) should be determined next week.

Weekly Status Summary #3

Some constants like wheel base have been determined by software team, so corresponding tests are not necessary in this case. Many software components were developed this week, so software testing increased accordingly. The zip line crossing and localization algorithms were tested for accuracy and reliability. On the hardware side, the components were tested to find the best-performing options available. Tests were conducted on the three kits' worth of ultrasonic sensors, large motors, and medium motors, and those with the highest performance were included in the complete robot. In the upcoming week, more software tests on the remaining state machine and algorithms will be conducted ahead of the upcoming Beta Demonstration. Preliminary testing on the integration of software components that were developed in isolation will be conducted, but not all integration testing will be completed. The results of the Beta Demonstration will play heavily into deciding the final testing plan for the final weeks of the project.

Weekly Status Summary #4

The second odometer test (with the revised wheelbase constant) was completed, and that constant has been passed to the software team to make the necessary adjustment. Integration testing of the robot and software for the beta demo is done - everything is ready for the beta demo. Next week's testing will depend on the result of the beta demo. In particular, software adjustments based on the results of the beta demo will be conducted next week. Furthermore, avoidance, searching, and integration testing for the final competition needs to be completed within the next two weeks.

Weekly Status Summary #5

The results of the beta demo revealed a mismatch between the coordinate system in our software and the coordinate system provided by the game server. We revised the software based on the proper coordinate system of the final competition. The revised software passed its testing, and the problem experienced in the beta demo is fixed. However, extreme cases of the integrated software system should be performed to improve the success rate of the robot ahead of the final competition. Since the software is not yet complete, tests of avoidance, searching, and final integration could not be completed. Once the software implementation is complete, the remaining tests will be done and this document will be updated.

5.0 Glossary of terms

None required.