McGill University

ECSE 211: Final design project

Testing Document

Philippe Papineau, Ali Sharif

supervised by
David A. Lowther

April 8, 2017

Contents

1 March 1, 2017

These tests were performed by Ali Sharif with the help of Philippe Papineau.

1.1 Goal

The LeJOS library provides two motor classes for controlling the EV3 motors, RegulatedMotor and UnregulatedMotor. This experiment was conducted with the hopes of discovering which motor performed better for use in the ball launcher.

1.2 Test 1: Unregulated vs Regulated Class

Hardware Version: Mark I

Software Version: Lab 5 Code

Procedure: The robot will be made to fire a ball from a distance of 7.7 tiles using the regulated motor class first and then the unregulated class. Five trial runs will be conducted. The robot will be placed in the lower left corner of the field, 40 cm away from the left wall and 30 cm away from the right wall, with the catapult facing north.

Results: The results showed that the Regulated class performed better than the Unregulated class. The Unregulated motor class, unlike the Regulated class, did not manage to hit the target. The results of the test are purely qualitative. There is no data.

1.3 Test 2: Unregulated vs Regulated Class (without lip)

Hardware Version: Mark I

Software Version: Lab 5 Code

Procedure: Using the results of Experiment 1, we will remove the robot's catapult's lip and repeat the above experiment.

Results: The results of this experiment showed that Unregulated class performed equally to the Regulated class.

1.4 Test 3: Greater arm length

Hardware Version: Mark I

Software Version: Lab 5 Code

Procedure: Repeat Experiments 1 & 2 using greater arm lengths for the catapult.

Results: This experiment brought to light a completely different issue regarding the catapult's arm and frame. The larger length made both the arm and frame highly susceptible to rotational torsion. The torsion would cause the arm to sway to the left or right during its upswing and throw the ball off to the side. Holding down the frame to minimize the frame's torsion we observed that Regulated motor class suffered greatly using larger arm lengths. Hence, we will use the Unregulated motor class for the time being.

Conclusion: We will use the unregulated class to power the catapult.

1.5 Test 4: CPU fan

Testers Romain, Philippe

Goal See if a 12 V CPU fan can deflect a red ball from its trajectory.

Procedure: *See also figure /// of the appendix* The ball will be throw in a bucket. We will put the fan 5 cm above the ground, first parallel with the ground and then with a 45°angle.

Results Non of the fan position worked. The ball went straight into the bucket. Because of its mass, the momentum is too important and the ball is not slowing down, even with the 12 V fan. A ping-pong ball, because of its smaller mass and dimension, could easily been deflected.

Conclusion We will not use a fan as a defence mechanism. Even if the test was successful, it would have created lots of new problems: how to put a 12 V source in the robot? How to make place for the fan? Are we allowed to have add all this additional material?

2 March 6, 2017

These tests were performed by Ali Sharif with the help of Philippe Papineau.

2.1 Goal

This is the testing of Mark II catapult and drive train.

2.2 Test 5a: Swing time adjusting

Hardware Version: Mark II – Arm length is 60 centimeters.

Software Version: Modified Lab 5 Code – Modified to make it easier to adjust the swing time between launches. Otherwise the code is identical.

Procedure: We will adjust the swing time and fire the ball to determine the best swing time for the motor.

Results: In each trial the ball fell short of the target.

Swing Time /ms	Successful?
500	N
550	N
600	N
650	N

Notes: The battery voltage was 7.8 Volts

2.3 Test 5b

Hardware Version: Mark II – Arm length is 50 centimeters.

Software Version: Modified Lab 5 Code – Identical to Test 1

Procedure: We will adjust the swing time and fire the ball to determine the best swing time for the motor.

Expected Results: We expect to see the ball enter the target.

Results: Our initial swing time selection of 500 milliseconds fired the ball perfectly into the target. Out of the 10 trials conducted, the ball passed through the target seven times.

Notes: The battery voltage was 7.6 Volts. See also the video "Test Demo 6th March.mp4"

2.4 Test 6: Floor gaps

Hardware Version: Mark II – Identical to Test 2.

Software Version: Modified Lab 4 – The navigation class was modified to move the robot more quickly. Otherwise the code is identical.

Procedure: The robot will be placed on the field and made to cross two gaps. A 1 & 5 millimeter gap.

Expected Results: The back wheel of the robot will fail to cross gap and get stuck.

Results: The robot managed to pass over both gaps without problems.

Notes: The battery voltage was 7.8 Volts

2.5 Test 7: Robot speed

Hardware Version: Mark II – Identical to Test 2.

Software Version: Modified Lab 4 – The navigation class was modified to move the robot in a straight line at full speed. Otherwise the code is identical.

Procedure: The robot will be traverse the field at full speed.

Expected Results: Because of the two bricks and the tall structure, the robot weights a lot more. Therefore, we expect it to go significantly slower.

Results: The robot went at about the same speed as the smaller ones during the lab. However we can hear the metal ball rubbing on the floor (most of the weight is put on the balls).

Notes: The battery voltage was 7.8 Volts

3 March 13, 2017

These tests were performed by Ali Sharif with the help of Philippe Papineau.

3.1 Test 8: Gear catapult

Hardware Version: Mark II - Catapult with gears.

Software Version: Specially created software for testing. It would just spin up the motors.

Procedure: This is identical to our previous catapult experiments. The only difference was that the arm was connected to a gear instead of directly to the motor. The gear would be turned by the motor. We expected this would limit the arm sway induced by the structure of the motor.

Results: The effect of the change on the side to side sway was negligible. The gears introduced other problems, however. The gears would constantly slip due to large acceleration of the motors.

Notes: The battery voltage was 8.0 volts.

3.2 Test 9: Elastic catapult

Hardware Version: Mark II - Catapult type with elastics

Software Version: Specially created software for testing. It would just spin up the motors.

Procedure: The launcher was placed on the field, and made to fire the ball. This would verify the feasibility of the design. The test is purely qualitative.

Expected Results: The launcher would be able to fire at a distance of 8 tiles.

Results: The distance fired was 4 tiles.

Notes: The battery voltage was 8.0 volts.

3.3 Test 10: Rotating Wheel Launcher

Hardware Version: Mark III - Rotating Wheel Launcher type

Software Version: Specially created software for testing. It would just spin up the motors

Procedure: The launcher was placed on the field, and made to fire the ball. This would verify the feasibility of the design. The test is purely qualitative.

Expected Results: The launcher would be able to fire at a distance of 8 tiles.

Results: The distance fired was 2 tiles.

Notes: The battery voltage was 8.0 volts.

4 March 15, 2017

These tests were performed by Ali Sharif with the help of Rami Djema.

4.1 Test 11: Travel test I

Hardware Version: Mark III

Software Version: 0.0.2 Alpha 3 - Using lab 4 Navigation

Procedure: We asked the robot to travel to (80,80).

Results: The robot did not manage to make it very far. There was a very strong bouncing effect.

4.2 Test 12: Travel test II

Hardware Version: Mark III

Software Version: 0.0.2 Alpha 3 - Using custom pilot Navigation

Procedure: We asked the robot to travel to (80,80).

Results: The robot manage to make it to (80,80) with a large degree of error. The results of this test were purely qualitative.

Note: This is a one shot based system, so calculations are off.

4.3 Test 13: Corner localization I

Hardware Version: Mark III

Software Version: 0.0.2 Alpha 3

Procedure: We placed the robot in a corner along a diagonal and used the ultrasonic localization.

Results: The robot manage to localize with an error less than 2 degrees. The results of this test were purely qualitative.

Note: Using falling edge localization.

5 March 18, 2017

These tests were performed by Romain Nith with the help of Philippe Papineau.

5.1 Test 14: Firing test I

Goal: We did a simple firing test to see how far the ball went.

Procedure: We placed the robot in the lower left corner of the field, 10 cm away from the left wall and 10 cm away from the right wall, with the crossbow facing north. We cocked the crossbow and then released the trigger.

Results: It went for about 12 tiles with rebound at 9 tiles.

5.2 Test 15: Trigger test I

We also tested the ratchet mechanism and the trigger mechanism.

Procedure: We pulled the rubber band until it was behind the trigger, and then released it. The trigger was robust enough to hold the rubber band in the firing position.

6 March 23, 2017

These tests were performed by Ali Sharif with the help of Rami Djema and Julien Courbebaisse.

6.1 Test 16: Wifi test I

Procedure: Run the wifi test code given to us by the professors.

Expected Results: The robot should connect the server and and display the information it receives.

Results: An exception regarding a missing class definitions was encountered. The test could not continue further.

Notes: We manually copied the missing JSON jar onto the brick to alleviate this problem

Results 2: The robot successfully connected and displayed all information.

6.2 Test 17: Corner localization II

Using Julien's lab 3 code, the robot pointed towards (0,0). It moved slowly, but was accurate.

7 March 24, 2017

These tests were performed by Ali Sharif with the help of Julien Courbebaisse and Romain Nith.

7.1 Test 18: Demo

Goal: Prepare for the competition in two weeks.

Procedure: The robot will perform basic tasks essential for the competition. In attacker mode, it will receive parameter using Wi-Fi and beep. In a corner, it will localize within 30 seconds and beep again. Finally, it will proceed to a launch point and launch the ball through the hoop.

Expected Results: We expected the robot to localized really slowly, but be able to navigate to the correct coordinates.

Results: The robot hit the wall twice. We observed that the right wheel was spinning without touching the ground. This must have been a considerable source of error for the odometer.

Conclusion: The time to localize was too long. We need to improve the overall structure so the firing mechanism can work properly. We need to change the rubber bands.

7.2 Test 19: Demo improvements

During the same day, we came back to the lab to work on the robot with the hope that at the end of the day, we could reproduce the demo, but this time with everything working as it should be.

Procedure: Same as the demo.

Results: Pathfinder was successful. See also the video "Test 19-Demo Improvements.mp4" under Videos folder.

8 March 25, 2017

These tests were performed by Rami Djema. Note. For tests 20 to 22, please also look at the Excel documents.

8.1 Test 20: Travel test III

Procedure: The navigation testing was done by simply starting the robot at (0, 0) and making it navigate to position (30, 30) without correction. The x and y errors were recorded.

Expected Results These values are meant to be somewhat off because the odometer correction is not activated. With our light sensor, odometry correction, these errors are expected to diminish drastically.

Results As expected, the values are a bit off. Also, the navigation is choppy, which hinders our angle slightly.

Conclusion: The navigation error might be able to be fixed using odometry correction.

8.2 Test 21: Travel test IV (with light sensor correction)

Procedure The navigation testing with correction was done by starting the robot at (0, 0) and making it navigate to position (60, 60) without correction. The x and y errors were recorded.

Expected Results: The error is expected to diminish drastically from the tests done without correction.

Results The error significantly increases rather than diminishing. This is likely to be a software implementation issue. Our navigation seems to be off which causes our robot to go over lines and mess up the odometer position. This causes significant overshoot and undershoot in our navigation.

Conclusion: The navigation code needs to be fixed before the we can continue any further. Nothing can be said about the effectiveness of the correction at this point.

8.3 Test 22: Firing Test II

Procedure The ball was launched 5 tiles away from the goal. The expected and actual bounce x and y values were recorded.

Expected Results: The ball is expected to bounce at (90, 90) and go through the net.

Results: The ball bounced near the expected bounce x and y coordinates. The success rate is quite low at 5/13 shots.

Conclusions: Further hardware adjustments are needed to the robot. Especially regarding the ball guide.

8.4 Test 23: Pathfinder Test

Procedure A simulation of segmented pathfinder was ran on eclipse with the inputs (600, 600) cm, and the coordinates printed to standard output.

Expected Results Inputting the coordinates into the pathfinder should break the navigation into x and y. Then it should break down each of those paths into segments of length equal to one tile.

Results The algorithm successfully segmented the path as explained in the expected results. See figure 4 in appendix A.

8.5 Test 24: Wifi Test II

Procedure Using the provided DPMserver, the forward team number, builder team start corner, defender team number, defender team start corner, forward line position, defender zone dimensions, ball dispenser position and ball dispenser orientation were provided through Wi-Fi. This data was printed onto the display of our Robot.

Expected Results All transmitted information will print onto the display.

Results: All transmitted information printed onto the display.

8.6 Test 25: Dispenser test

See Appendix B of the testing document.

8.7 Test 26: Launching limiter tuning

See Appendix B of the testing document.

9 March 4, 2017

9.1 Test 27: Final corner localization and navigation to a point on the field

Goal: Prepare for the final competition

Procedure: The robot will be placed in a corner. It will localize, go to (0,0)

Expected Results: We expected the robot to localized really slowly, but be able to navigate to the correct coordinates.

Results: The robot hit the wall twice. We observed that the right wheel was spinning without touching the ground. This must have been a considerable source of error for the odometer.

Conclusion: The time to localize was too long. We need to improve the overall structure so the firing mechanism can work properly. We need to change the rubber bands.

9.2 Test 28: Shooting

Goal: Verify if shooting is reliable and can sustain the stress from pulling back and forth the elastic.

Procedure: The robot will run the code from software v0.0.2-alpha.2. After the elastic is pulled for the first time, we load three balls, one in the chamber and 2 in the reloading mechanism.

Expected Results: The robot should fire the first ball, pull the elastic back, raise the loading structure and reload one ball in the chamber. It will this two times, until all the balls were shot.

Results:

Conclusion: The time to localize was too long. We need to improve the overall structure so the firing mechanism can work properly. We need to change the rubber bands.

Launching Limiter Tuning

Done by Romain Nith on the 27/03/2017

Problem: Crossbow launches the ball too far (9 tiles without hitting the ball)

Solution: Add a stopper to reduce the elastic band travel

Procedure: Robot is positioned at 7 tiles from target. Add a Lego piece in on the crossbow's barrel, launch and record distance. Increment the stopper until requirement is met.

Stopper position according to the exit of the launcher	Tile when it hits the floor	Hits target?
0	9	Yes (Direct hit)
2	8	No
4	7	No
5	6	Yes
6	6	Yes
7	5	No

Result: At position 5 and 6, the stopper managed to launch the ball at the required distance while hitting the floor within specifications. To prevent losing too much momentum when hitting the floor, the stopper will be placed at position 5.

Remarque: To prevent damaging the stopper Lego part, we chose a small piece that has a 3 block height for stronger structure.

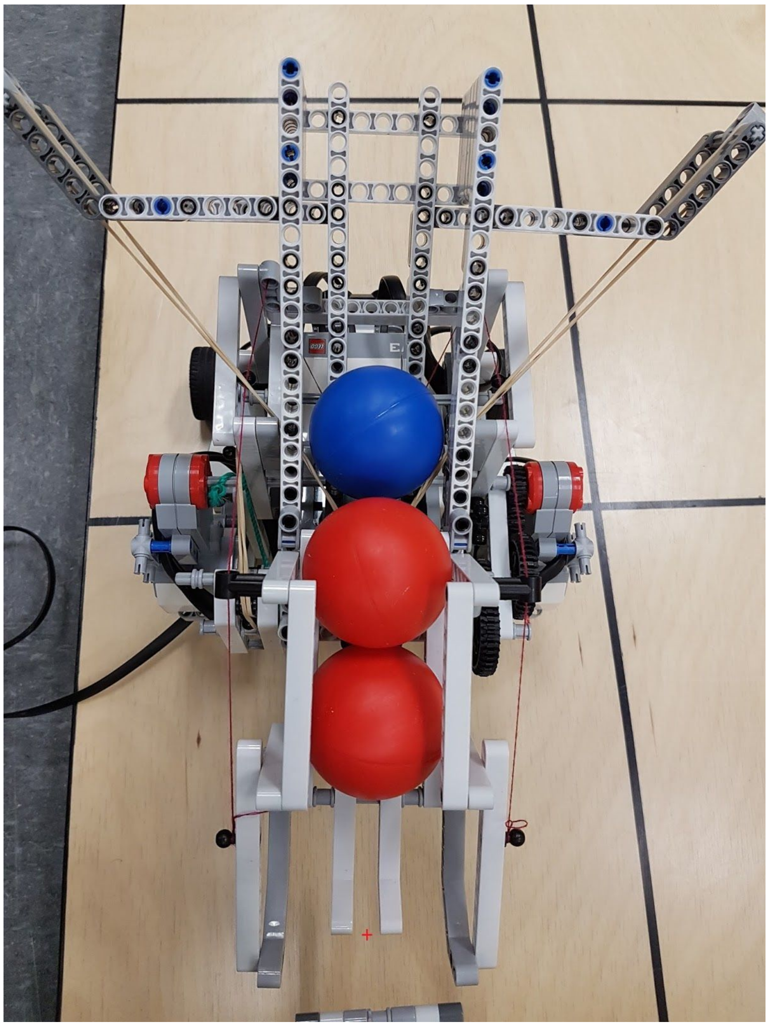
Dispenser Test

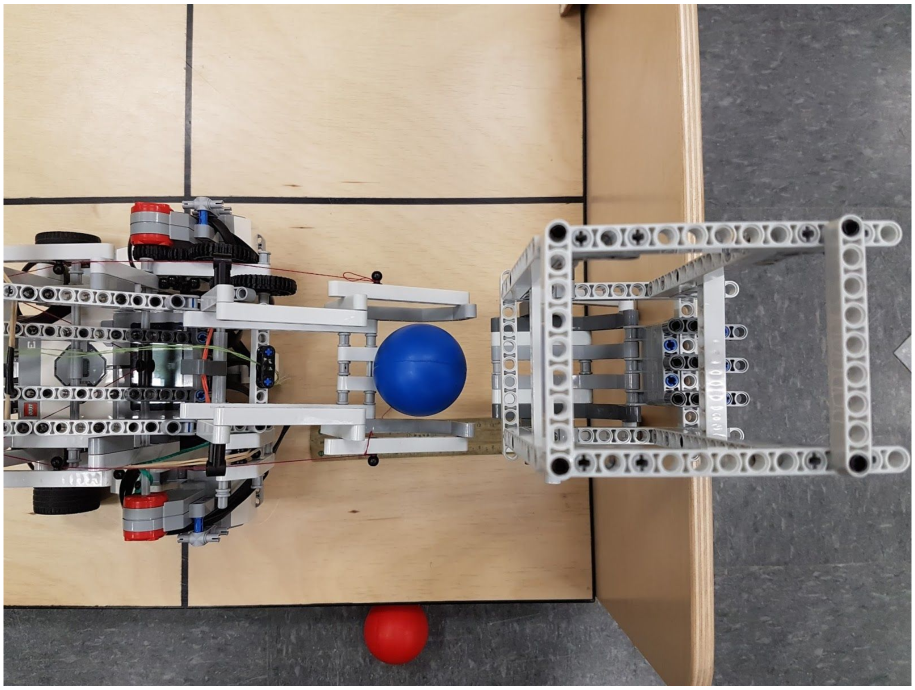
Done by Romain Nith on the 27/03/2017

Problem: Determine the allowed robot-dispenser distance error.

Setup: Robot place next to the dispenser. Let the ball fall into the dispenser and record for each incrementation in one axis the result

The origin is set on the robot arm's middle for the Y-axis and the edge for the X-axis (+)





Varying x and setting y at 0 cm

Distance between Robot and Dispenser (in cm)	Did the ball lend in the reservoir?*
0.5	Yes
1	Yes
1.5	Yes
2	Yes
2.5	Yes
3	No
2.9	No
2.8	Yes

Varying y and setting x at 2.5 cm

Distance between Robot and Dispenser (in cm)	Did the ball lend in the reservoir?*
0.5	Yes
1	Yes
2	Yes
3	Yes
3.5	Yes
4	No
3.75	No
3.5	No
3.4	No
3.3	Yes

(*) Is successful when 8 or more out of 10 trials are successful

Conclusion: the V2 “Ball Reloading and Reservoir Mechanism” has an acceptable error of [2.8; 3.3] cm.

APPENDICES

Contents

APPENDIX A: Images

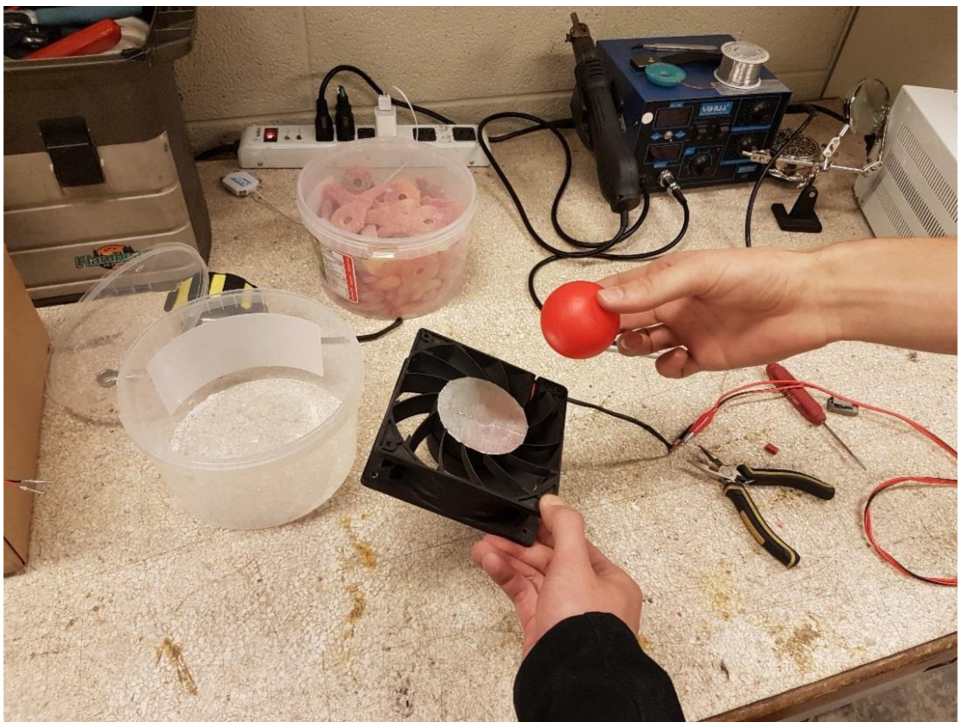


Figure 1 (Test 4): The bucket, the CPU fan, and the ball. Even with different angles, the ball went directly into the bucket. It did not deflect when it was position above the fan.

Figure 2 and 3 (Tests 8 and 9): We wanted to try other more powerful catapult mechanism. The first image shows the mechanism for a catapult with gears.

We tried different setup with the gears, but too much new problems were introduced. More specifically, because of the high motor speed, the gears were continuously slipping. We concluded that the use of gears was better suited for low speed use.

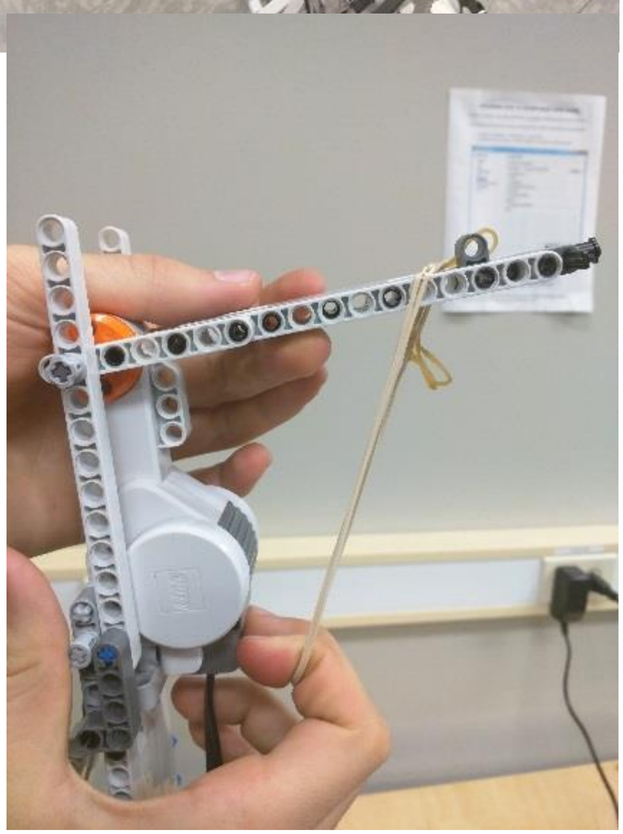


Figure 3: We also tried attaching rubber bands to the catapult, but we were not able to have a successful design.

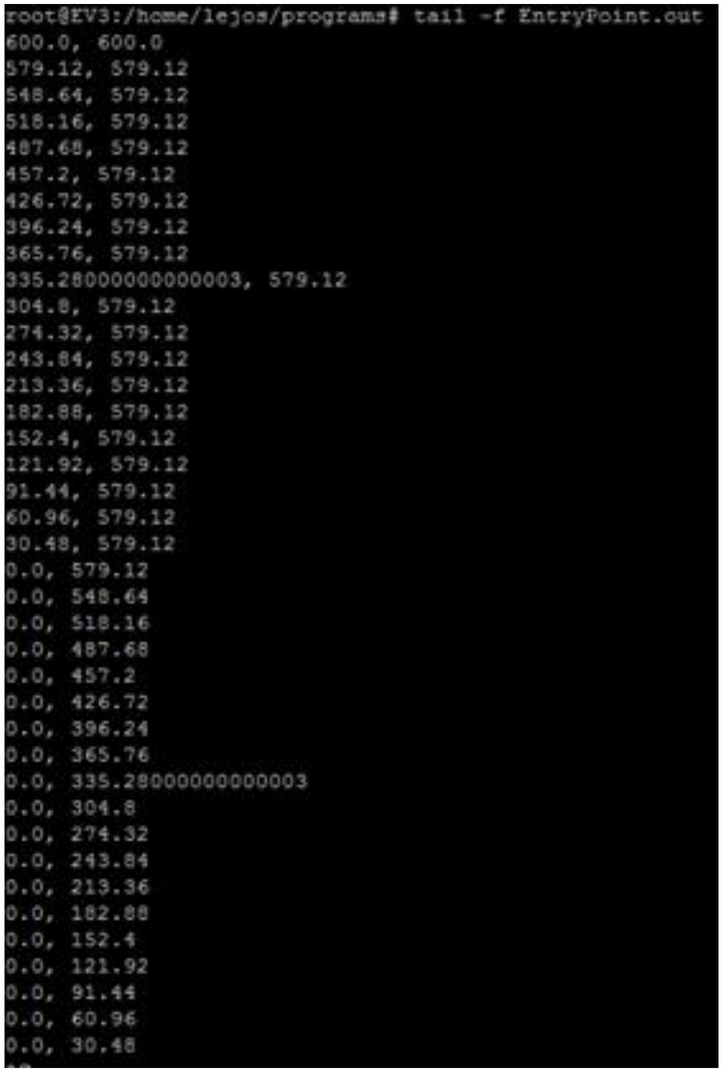


Figure 4: Results of test 23

APPENDIX B: Test results

See also Excel spreadsheets in Dropbox.

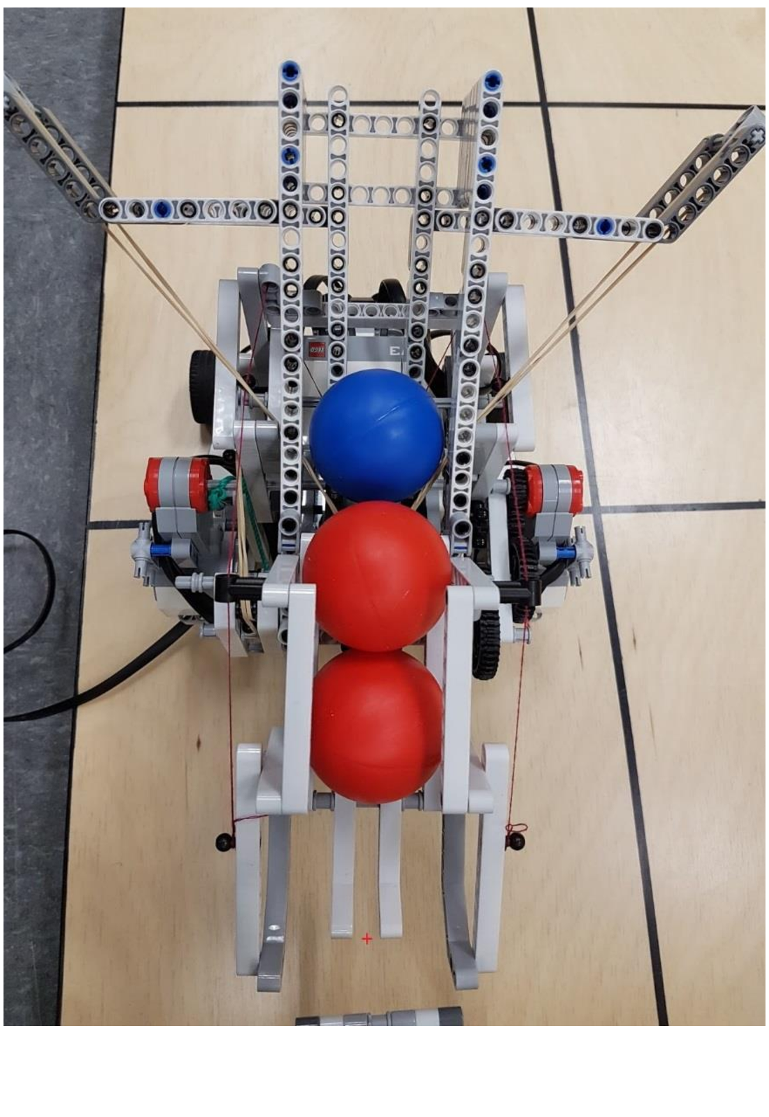
Dispenser Test

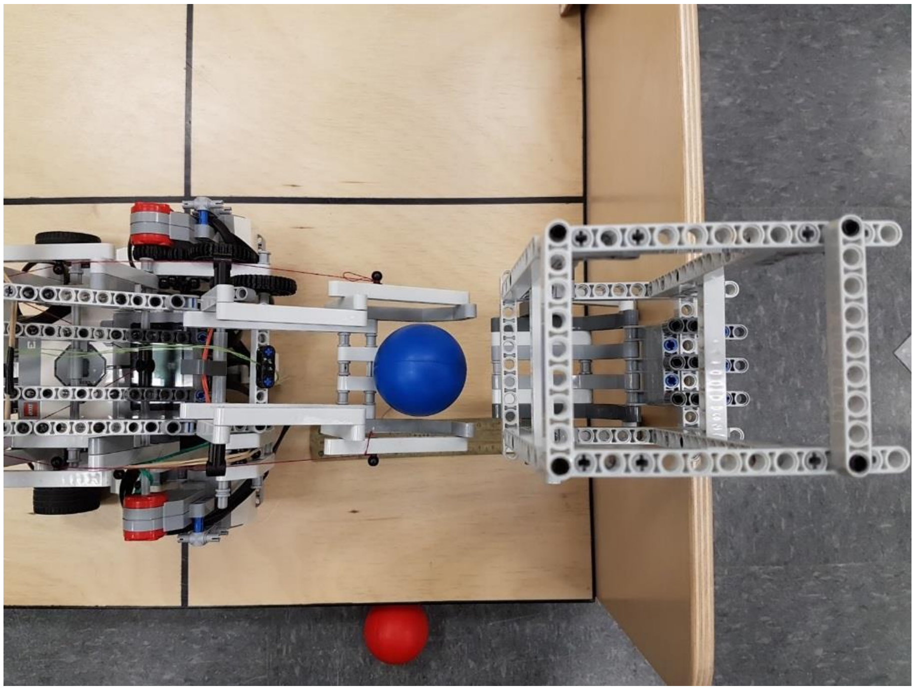
Done by Romain Nith on the 27/03/2017

Problem: Determine the allowed robot-dispenser distance error.

Setup: Robot place next to the dispenser. Let the ball fall into the dispenser and record for each incrementation in one axis the result

The origin is set on the robot arm's middle for the Y-axis and the edge for the X-axis (+)





Varying x and setting y at 0 cm

Distance between Robot and Dispenser (in cm)	Did the ball lend in the reservoir?*
0.5	Yes
1	Yes
1.5	Yes
2	Yes
2.5	Yes
3	No
2.9	No
2.8	Yes

Varying y and setting x at 2.5 cm

Distance between Robot and Dispenser (in cm)	Did the ball lend in the reservoir?*
0.5	Yes
1	Yes
2	Yes
3	Yes
3.5	Yes
4	No
3.75	No
3.5	No
3.4	No
3.3	Yes

(*) Is successful when 8 or more out of 10 trials are successful

Conclusion: the V2 “Ball Reloading and Reservoir Mechanism” has an acceptable error of [2.8; 3.3] cm.

Launching Limiter Tuning

Done by Romain Nith on the 27/03/2017

Problem: Crossbow launches the ball too far (9 tiles without hitting the ball)

Solution: Add a stopper to reduce the elastic band travel

Procedure: Robot is positioned at 7 tiles from target. Add a Lego piece in on the crossbow’s barrel, launch and record distance. Increment the stopper until requirement is met.

Stopper position according to the exit of the launcher	Tile when it hits the floor	Hits target?

0	9	Yes (Direct hit)
2	8	No
4	7	No
5	6	Yes
6	6	Yes
7	5	No

Result: At position 5 and 6, the stopper managed to launch the ball at the required distance while hitting the floor within specifications. To prevent losing too much momentum when hitting the floor, the stopper will be placed at position 5.

Remarque: To prevent damaging the stopper Lego part, we chose a small piece that has a 3 block height for stronger structure.

APPENDIX C: Software testing localization

See Word document “Software testing for localizer”