Ballistic Launcher

ECSE 211 - Design Principles and Methods

Lab 5 Report

Team 5

Xirui Zhang : 260656808
Xinran Li : 260774237
Cheng Chen: 260775674
Arianit Vavla: 260868601
Tony Ou: 260867785
Hongtao Xu: 260773785

1. Design Evaluation

Hardware Design

In this lab, we use the design of large-based robot with two motors which gives more energy than accuracy in this lab. In figure 1.2, two motors were placed in the front to provide force for the robot to move forward. Also, a metal ball was placed at the back of the robot to provide stability for the robot while it's moving. A light sensor and an ultrasonic sensor were used in this lab for localization. The ultrasonic sensor was placed in the front and pointing straight ahead to detect the walls. The light sensor was placed at the back and pointing downwards to detect the black lines on the floor. To launch the ball, a catapult style ball launcher was designed. We used two motors to provide force for the catapult. When we launch the ball, the two motors will provide a great acceleration to the catapult. After a few tests, we found that no matter what speed or acceleration we set for the motors, we cannot launch the ball to the distance described in the requirements document. After doing some research on the internet, we decided to buy some elastic bands and fix one end to the catapult and fix another end to the robot, then we were able to launch the ball far enough. However, the robot shakes after every launch. We decided to add more weight on the robot by adding one metal ball to each side of the robot and one motor to each side of the robot. After some tests, everything was working fine.

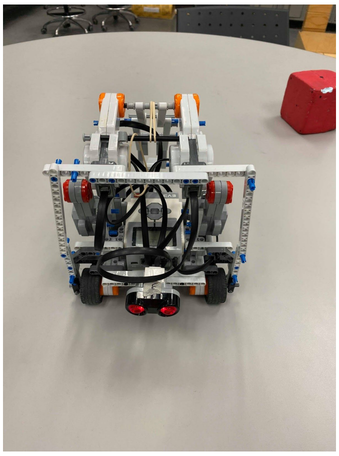


Figure 1.1 Front View of Hardware

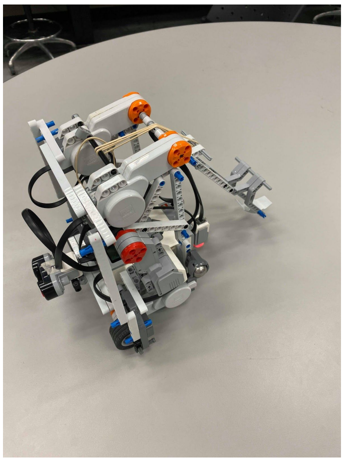


Figure 1.2 Lateral View of Hardware

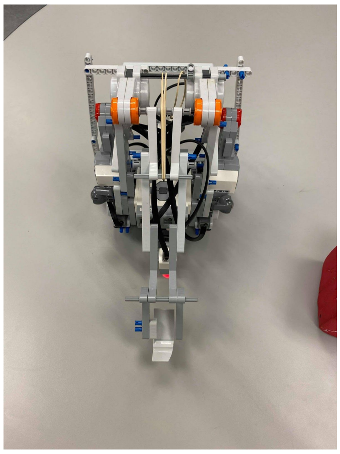


Figure1.3 Top view of Hardware

Software Design
In this lab, Navigation and Localizer classes drive the falling edge and localize to (0,0) point like we did in lab 3 and 4. Odometer class will return the current coordinates of the robot and Printer will print these coordinates on the screen.

Launcher class is implemented to control the motors to use the catapult. For the first few tests, we set the speed of the motors to 500 and acceleration to 600. We found that the ball could not reach the target. Then we set both the speed and the acceleration of the motors to maximum, the ball still could not reach the target. After some research, we used some elastic bands to exert a larger force on the catapult since the motors could not provide enough power. After some tests with the elastic bands, we were able to launch the ball to the target. For the mobile launcher part, we set the target point in the Resource class. The navigation class will move the robot to the launching point after localization. Once the robot reaches the launching point, the launcher class will launch the ball to the target.

When the main class is called, ultrasonicLocalizer starts to run as a thread and calls LightLocalizer when it finishes falling edge. Then the light localizer starts to navigate and correct itself at (0,0) point. After this we calculate the euclidean error distance between robot and targetpoint to determine if the robot is in the range of non-shooting circle. In this case, there are two conditions, if the robot is in the cycle, we calculate four points by plus or minus X, Y coordinates and navigate robot to the point which is not out of bound. If the robot is outside the cycle, we make the robot travel the euclidean error distance minus 120 and navigate towards the target point. After the launching point has been reached the Launcher class is called to run.

The launcher class has two methods. One called launchPosition is used to fold the robot back to the position ready to launch, it has lower speed. The other is called launchBal is used to launch the ball, the speed and acceleration speed has been tested and fixed in faster speed than launch Position.

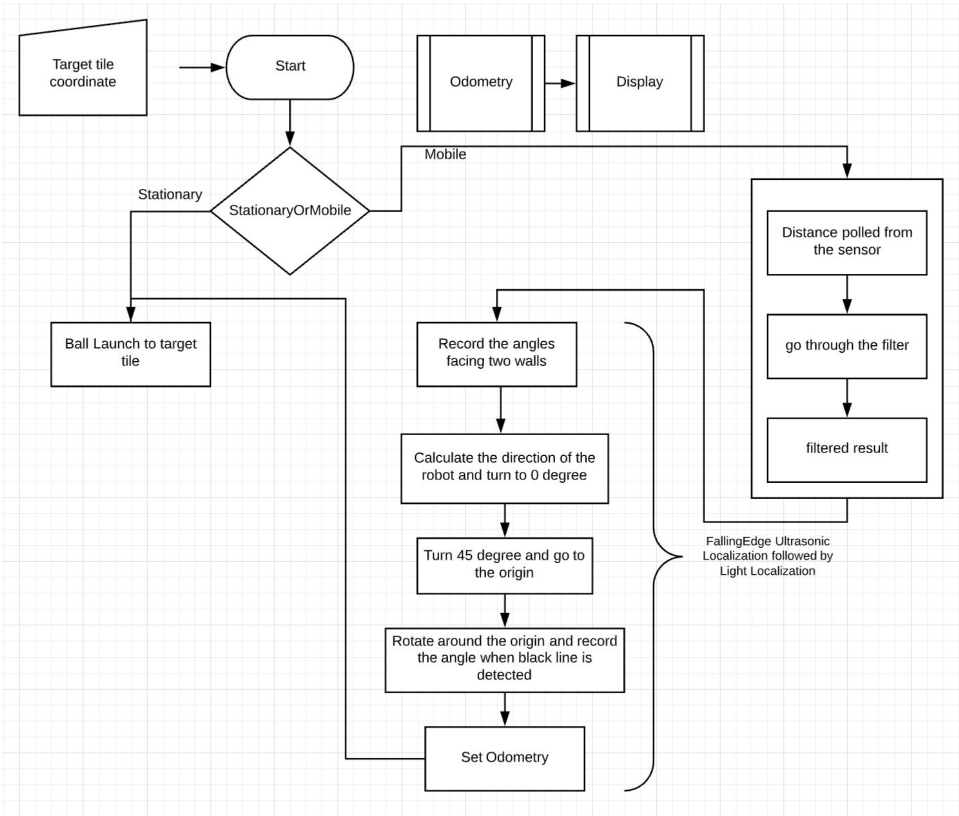


Figure 2. Flowchart of Software Architecture

2. Test data

The following table shows the test data for a total of 20 trials.

Note: Assume target coordinates is at the middle of the target tile

Target tile: (2,7)

Table 2.1. Test Data for Stationary Launch

Launch #	X coordinate (cm)	Y coordinate (cm)	X target (cm)	Y target (cm)
1	54.09	197.51	45.50	197.15
2	46.07	209.30	45.50	197.15
3	49.65	208.79	45.50	197.15
4	44.45	197.21	45.50	197.15
5	51.75	210.61	45.50	197.15
6	49.03	205.30	45.50	197.15
7	44.31	207.14	45.50	197.15
8	47.33	209.02	45.50	197.15
9	49.67	195.84	45.50	197.15
10	49.01	211.66	45.50	197.15
11	50.05	207.69	45.50	197.15
12	44.63	204.55	45.50	197.15
13	51.00	210.50	45.50	197.15
14	43.09	203.18	45.50	197.15
15	55.11	201.61	45.50	197.15
16	49.12	208.93	45.50	197.15
17	46.16	204.86	45.50	197.15
18	44.79	211.14	45.50	197.15
19	42.37	210.18	45.50	197.15
20	48.92	205.93	45.50	197.15

Note: Assume target coordinates is at the middle of the target tile

Target tile 1: (8,8)   Target tile 2: (7,6)

Table 2.2 Test Data for Mobile Launch

Run #	X coordinate (cm)	Y coordinate (cm)	X target (cm)	Y target (cm)
1	205.12	250.12	227.48	227.48
2	214.37	246.58	227.48	227.48
3	239.02	203.74	227.48	227.48
4	207.94	229.56	227.48	227.48
5	204.13	231.80	227.48	227.48
6	185.19	171.39	197.15	166.82
7	193.55	142.16	197.15	166.82
8	173.78	153.01	197.15	166.82
9	195.03	146.86	197.15	166.82
10	211.27	156.88	197.15	166.82

3. Test analysis

Note: Assume target coordinates is at the middle of the target tile

Target tile: (2,7)

Table 3.1. Stationary Launch Error with Center of Target Tile

Launch #	X error (cm)	Y error (cm)	X in target tile	Y in target tile
1	8.59	0.36	Yes	Yes
2	0.57	12.15	Yes	Yes
3	4.15	11.64	Yes	Yes
4	1.05	0.06	Yes	Yes
5	6.25	13.46	Yes	Yes
6	3.53	8.15	Yes	Yes
7	1.19	9.99	Yes	Yes
8	1.83	11.87	Yes	Yes
9	4.17	1.31	Yes	Yes
10	3.51	14.51	Yes	Yes
11	4.55	10.54	Yes	Yes
12	0.87	7.40	Yes	Yes
13	5.50	13.35	Yes	Yes
14	2.41	6.03	Yes	Yes
15	9.61	4.46	Yes	Yes
16	3.62	11.78	Yes	Yes
17	0.66	7.71	Yes	Yes
18	0.71	13.99	Yes	Yes
19	3.13	13.03	Yes	Yes
20	3.42	8.78	Yes	Yes

Note: Assume target coordinates is at the middle of the target tile

Target tile 1: (8,8) Target tile 2: (7,6)

Table 3.2. Mobile Launch Error with Center of Target Tile

Launch #	X error (cm)	Y error (cm)	X in target tile	Y in target tile
1	22.36	22.64	No	No
2	13.11	19.10	Yes	No
3	11.54	23.74	Yes	No
4	19.54	2.08	No	Yes
5	23.35	4.32	No	Yes
6	11.96	4.57	Yes	Yes
7	3.60	24.66	Yes	No
8	23.37	13.81	No	Yes
9	2.12	19.96	Yes	No
10	14.12	9.94	Yes	No

Table 3.3. Mean and standard deviation for the euclidean distance error (Stationary & Mobile)

Variable	Mean (cm)	Standard deviation (cm)
Stationary Euclidean Distance Error	10.33	3.66
Mobile Euclidean Distance Error	22.70	5.44

For Stationary Launch, the normal distribution We need the area under the curve which the ball hits 95% of the time. It is represented by figure 7.1. The shaded region shows that euclidean error must be between 4.0 cm and 22.0 cm in order to be considered a successful hit 95% of the time for the stationary launch. Similarly, As for the travel then shoot, we have determined that the error must be between 11.4 cm and 33 cm a shown in figure 7.2.

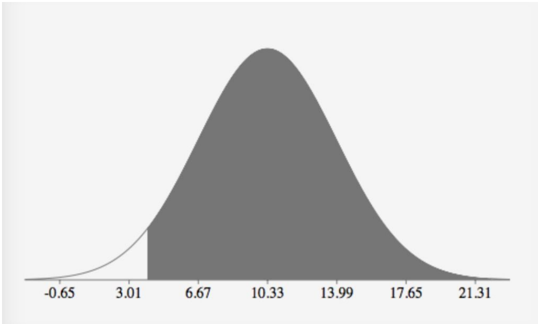


Figure7.1 Normal Distribution of Stationary Launch

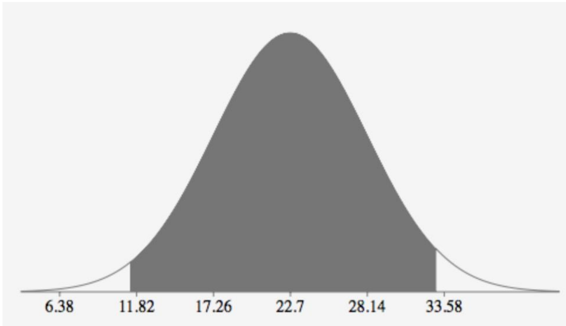
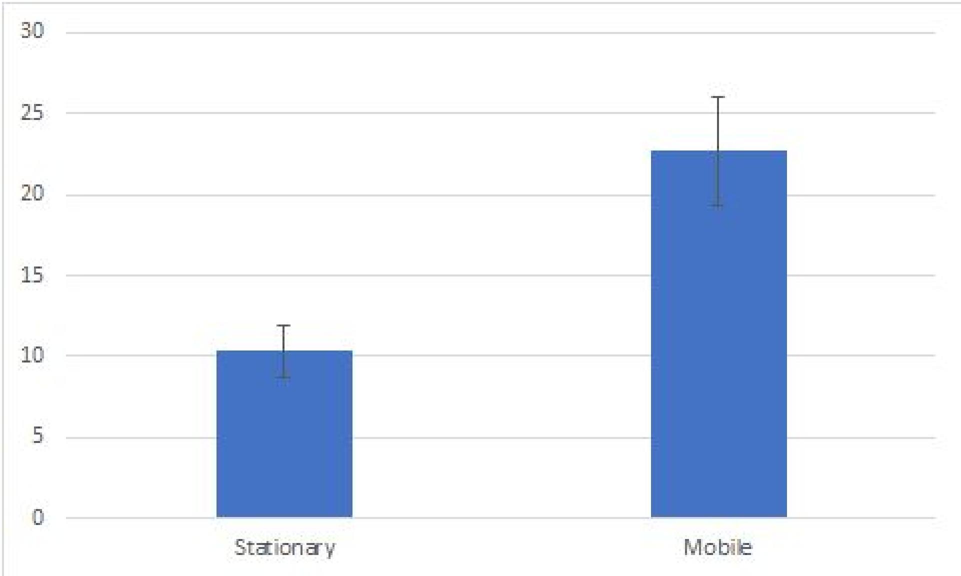


Figure7.2 Normal Distribution of Mobile Launch



Graph 3.4. Stationary & Mobile Launcher 95% Confidence Interval

4. Observations and Conclusions

Which statistic is most closely related to the repeatability of your mechanism? Explain.

The standard deviation of the landing position is the statistic most closely related to the repeatability because it shows how consistent the mechanism is over the different number of trials and the lower the standard deviation is, it means that the results of the different repeats are consistent for that mechanism.

Would you expect this statistic to be sensitive to error in launch position? If so why? if not, why not?

Yes, this statistic is expected to be sensitive to error in launch position because it affects the position in the landing position and the statistic is not taking in consideration the distance traveled by the ball, but only the landing position is taken as data. Also if the error of the angle at launch position is not correct, the landing position is also affected because the distance the ball has to travel is significantly large enough for the angle to play a big role in the error of the landing position.

Using the statistical model (above) you computed for your stationary launcher, determine the probability of a successful hit, for a confidence interval of +/- 15cm. How does this compare to your actual success rate.

We use the normal distribution formula to find the probability of a successful hit using the confidence interval of $\pm$ 15 cm knowing the mean is 10.33 cm and the standard deviation being 3.66 cm and the number of trials being 20.

$$10.33 + Z \times \frac{3.66}{\sqrt{20}} = 15$$

$$Z = 5.71$$

Looking at the normal distribution probability table, our Z is significantly larger than the largest Z on the table, which means that the probability of successful hit for a confidence interval of $\pm$ 15 cm is virtually 100%, which corresponds to our actual success rate, which means that it is well represented.

Repeat for the case of the mobile launcher.

The same formula applies for the mobile launcher but now the mean is 22.70 cm, the standard deviation is 5.44 cm and the number of trials is 10.

$$22.70 + Z \times \frac{5.44}{\sqrt{10}} = 15$$

$$Z = -4.48$$

Looking at the normal distribution probability table, our Z is significantly lower than the lowest Z on the table, which means that the probability of successful hit for a confidence interval of $\pm$ 15 cm is virtually 0%, which does not correspond to our actual 10% success rate, which means that trial 6 was an exception and has beaten the odds.

5. Further Improvements

Given your observations, what changes to your design would likely improve the repeatability of your results?

From the observations of the launching processes, we found that our robot was not stabilized while launching the ball (it vibrates), which led to unstable results and failure in the demo. In details, the center of gravity of our robot was in the middle and front part in the state of rest, while the center of gravity changed to the back part while launching the ball. When the robot was launching the ball, the stick to launching the ball moved largely and the robot also vibrated smally. In this way, the center of gravity moved to the back part, and the launching results would be inaccurate. To maintain the center of gravity in the front part, I think we need to add some heavy elements in the front part of the robot, like the heavy motors we added for testing data (in the following pics). In this way, our robot becomes more stable when it is launching the ball, and it will improve the repeatability of our results.

Also, for our mobile launch, some of them didn't work well and it is because of the navigation sometimes failed. But unfortunately we didn't have much time to plan on testing and fixing this problem. So in our Final project, we are going to do more testing on our navigation and how it doesn't work.

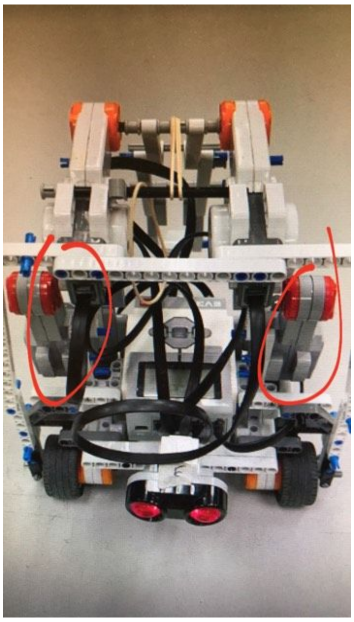


Figure 3. Motor Weights

The alternative to modifying the design is to consider a strategy that increases the likelihood of success. Given the statistics determined for your launcher in both the stationary and mobile cases, calculate the minimum number of launches required to guarantee at least one successful hit for each case. (Hint: Binomial Distribution).

binomial distribution: $p(X = s) = \binom{n}{s} p^s (1-p)^{n-s}$

Stationary Launch:

From Table 3.1, we have a probability of 100% to hit the target and our total trial is 20 so guaranteed a successful hit, so only 1 number of trials we need calculate to get a guaranteed successful hit.

Travel and Launch:

From table 3.2 we have a probability of 40% to hit the target, the total of shots we need in order to guaranteed to have a probability of 99.5% that one of the trials is a success is shown below:

For Binomial calculation: $p(X = s) = \binom{n}{s} p^s (1-p)^{n-s}$

Let f=1, p =0.4, k =1, so

$n = 4$

So we need 4 trials to get at least one successful shot.

(Extra calculation)

For independent trial:

$(1-0.4)^x = 0.005$

$0.6^x = 0.005$

$x * log(0.6) = log(0.005)$

$x = 5.86449 = 6$

Using calculating independent trials, we need 6 trials to get one success shot.

An alternative to modifying the design is changing the way of launching the ball. In our test, we launched our ball within rubber bands which was connected to the launching sticks. We change the number and thickness of rubber bands to regulate its launching speed and accuracy. There is an alternative launching way which is using an ejector (like the following pic).

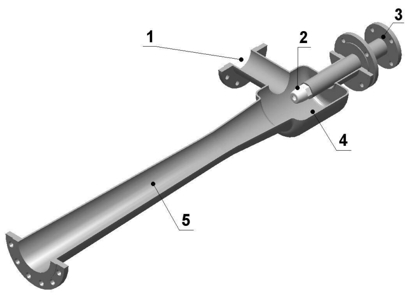


Figure 4. Ejector