McGill University

Lab 5 - Search and Localization

Team 12
Charles Bourbeau, 260868653
Darien Muse-Charbonneau, 260791466
Leonidas Lintzeris, 260928670
Linwei Yuan, 260857954
Shuby Mao, 260800098
Yichen Wang, 260761601

McGill Faculty of Engineering
ECSE 211: Design Principles and Methods
Winter 2020

February 27th, 2020

Section 1: Design Evaluation

Workflow

● **How did you distribute tasks amongst team members? How did this distribution of work aid in the design process?**

We distribute tasks amongst team members based on our role assigned in the final project (according to the Capability document). By allocating different types of work to appropriate group members, the efficiency can be greatly improved. For example, our software team is mainly responsible for coding, while our hardware leader is responsible for robot construction for Lab5. What is more, we have our testing leader responsible for testing the reliability of our robot design as well as the codes. With clear task distribution, our lab is quite successful this time.

● **What was the timeline of your work? Did any aspect of the lab take you longer than expected?**

We first held a meeting, discussing the task assignment and brainstorming ideas for the lab. Then we have our hardware leader built a prototype for this lab based on our models from previous labs. After that, our testing leader has conducted several testing trials to see which ultrasonic and light sensors are more accurate than others and those sensors would be used (since there are 3 sets of sensors available after the group merge). Then, our software team started to write codes for this lab using previous classes we wrote such as Navigation and Localization. Finally the testing lead worked together with the software team to debug and calibrate the program, leading to a reliable and accurate result. Most of the parts took us less time than we expected, except for the testing part, where a lot of data were required for plots and documentation.

● **Did your design process or workflow change drastically with more group members? How did you deal with this change in the environment?**

The overall design process does not change drastically, it is more about how to utilize those resources more efficiently and correctly. Before becoming a group of six, the entire lab is done by only 2 people, so inevitably we will face some areas that we are not experienced in. However, with more group members participating, we could divide the whole design process into several smaller tasks and assign to those who are capable of doing it. And furthermore, with more group members, it is possible to tackle different tasks concurrently, saving a lot of time of doing tasks ‘sequentially’, as some tasks are parallel to each other. Therefore, to deal with this change, we believe the most important thing to do is to realize every group member’s pro and cons and then assign tasks based on everyone’s capability.

● **Include any flowcharts or graphics to help describe your workflow.**

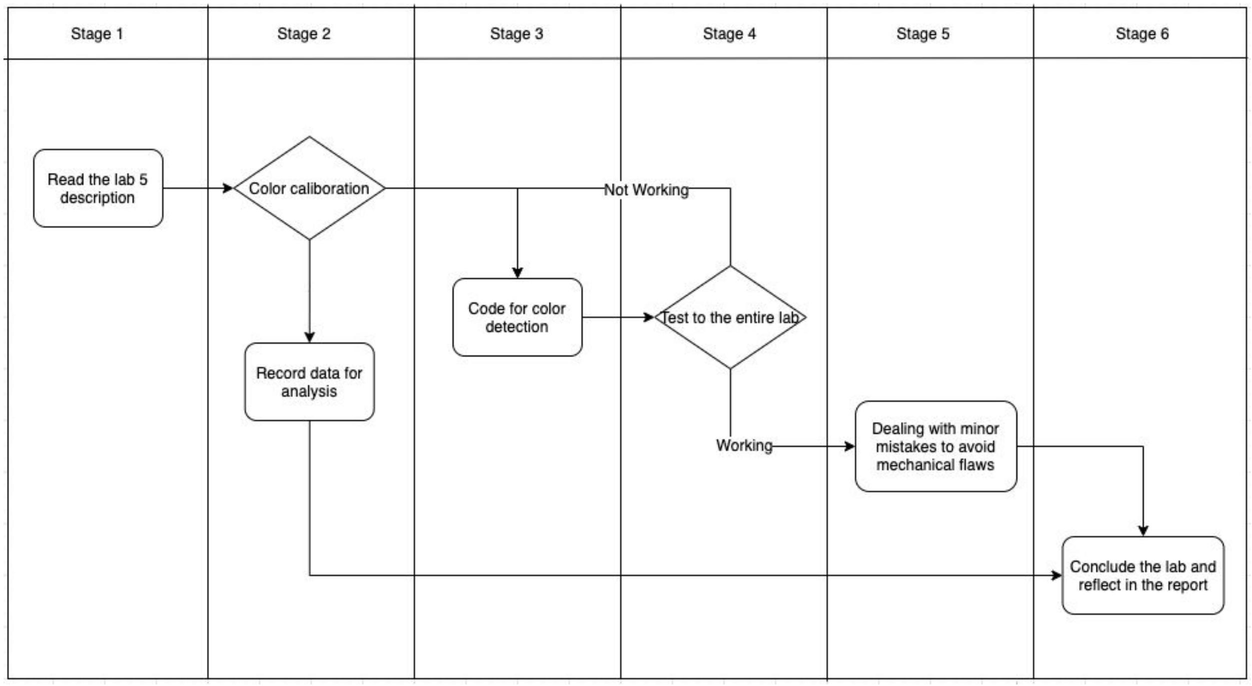

Figure 1: Lab 5 workflow flowchart

Hardware Design

● **Give a general overview of the hardware design. Include graphics or any images taken of your robot here.**

For the hardware design, the team all agreed on having a small base width compact robot. Before starting the build, all three groups presented their robots from previous labs. The robot that had the most compact robot and easiest to work off of was chosen. In fact, group 27's robot was used as the organ donor. The red motors mounted onto the EV3 brick were features that were conserved from group 27's robot. On the front of the robot, two L-shaped pieces were added to provide mounting points for the ultrasonic sensor, as well as the color detection sensor. The ultrasonic sensor was mounted in the front to facilitate the localization to point (1,1). The color sensor in the front was mounted high off the ground to be able to pass over the highest ring and detect its color. Also, a second light sensor was implemented on the back of the robot away from its center of rotation to facilitate light localization. Finally, a small spoiler on the back of the robot was kept from group 27's, in order to manage all the wires and keep them out of the EV3's path.

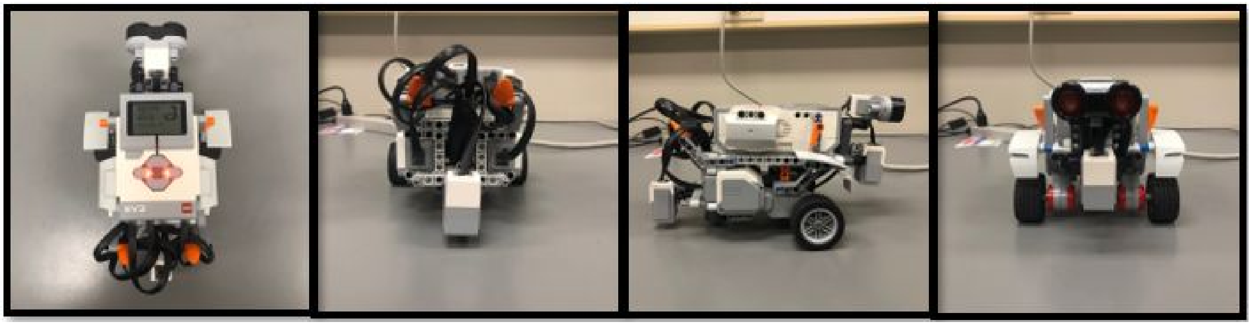

Figure 2: All Around Images of The Robot

● **How did you validate this hardware design prior to building it? If you did not, how could you include this step in the future?**

Initially, before any of the hardware was built some preliminary sketches were made by the hardware lead as can be seen in figure 3. Once the sketches were done, the sketches were shown to all team members and each member offered some pointers. The sketches then got modified and more detailed. Finally, once the sketches were approved by all team members the hardware design was built.

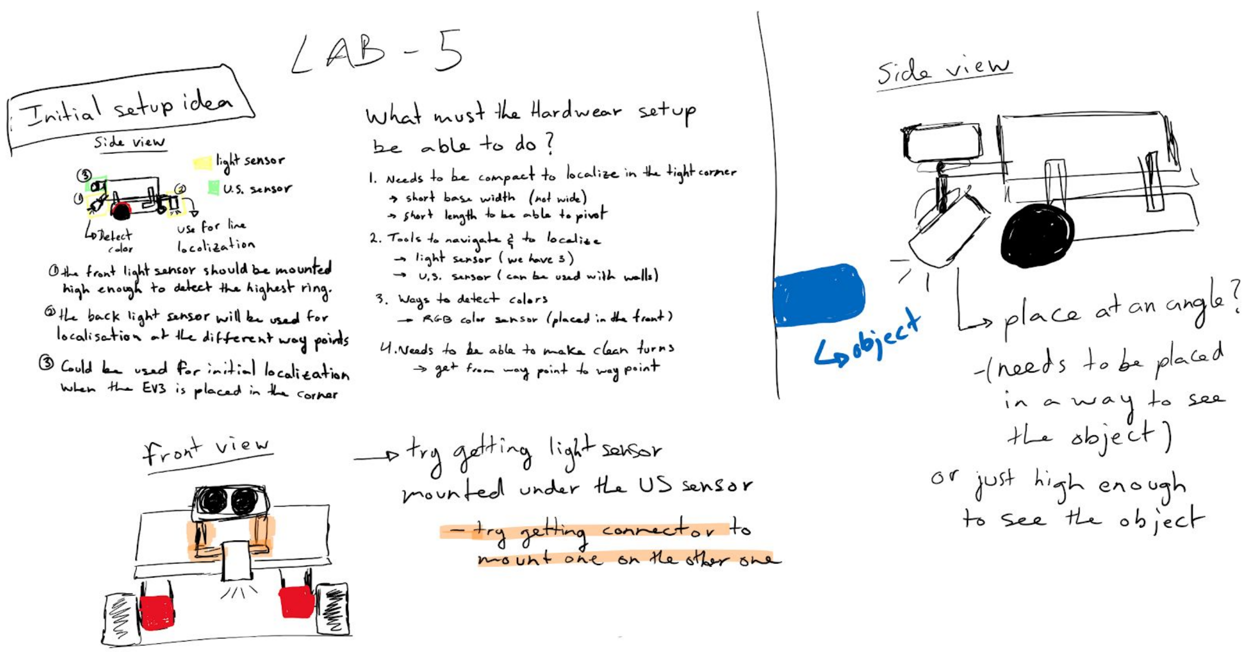

Figure 3: Hardware Sketches

Software Design

● **Give a general overview of the software design. Include graphics and diagrams here.**

The software is composed of many components and wrapped together in a single project. First we used the navigation system from lab 4, the odometer system from lab 2, the color and ultrasonic localization from lab 3. Aside from the previously developed component, we also developed a new system for the color detection which allows the robot to detect the ring color. To correctly detect the color under various different lighting conditions, we took samples from both labs and performed the Gaussian distribution on the RGB channel and computed a mean and the standard deviation for each color. After we establish the range value, when we measure a sample, we would compute the euclidean differences from each color and select the color in which the sample is closest to. After that, we combine all the components and integrate them together. Below is a flow diagram as well as the domain model of the software.

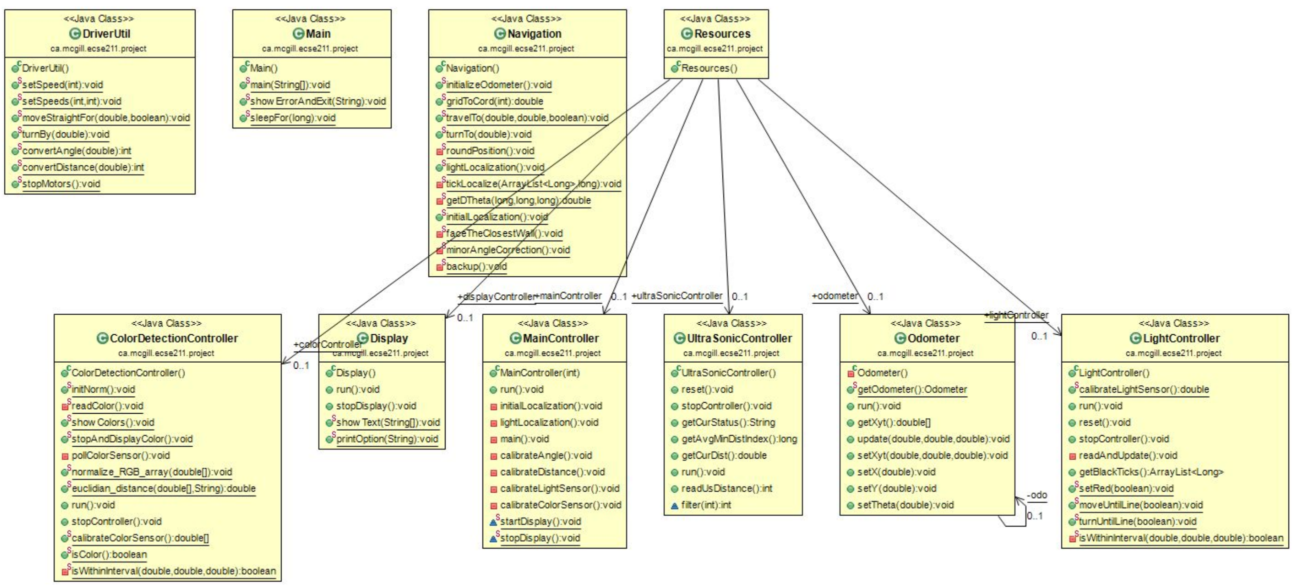

Figure 4: Software design

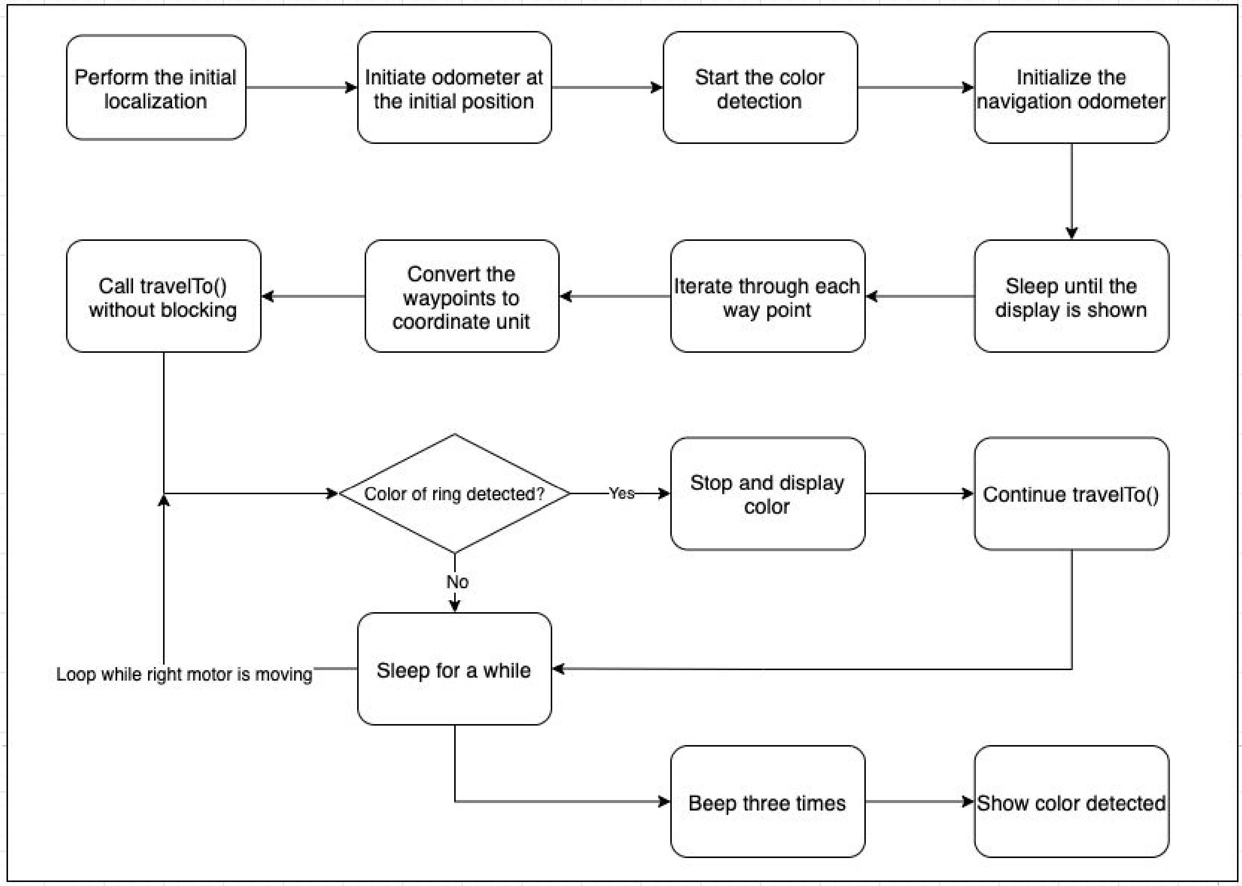

Figure 5: Flowchart of the main method of color detection

● **How did you validate your software design prior to running it on the robot? If you did not, how could you include this step in the future?**

We first validated the software through code review where team members review other member's code to make sure the logic is clear and is bug free. We also perform some unit tests on certain parts of the code such as the navigation to make sure that the logic behaves ideally. We tested the navigation logic on multiple sets of start and end locations to see if the method would provide the right direction and the correct distance to travel.

Section 2: Test Data

<u>Model Acquisition</u> (4x10 independent trials)

ROOM TR0110												
	Yellow ring			Blue ring			Orange ring			Green ring		
Trial	R	G	B	R	G	B	R	G	B	R	G	B
1	206	112	24	23	78	72	148	39	9	76	126	19
2	204	110	24	11	51	62	127	35	11	45	82	14
3	135	74	19	14	60	68	138	42	10	82	147	18
4	106	69	16	33	99	76	40	10	3	73	122	17
5	199	107	23	7	31	42	146	43	10	72	125	17
6	222	128	24	36	100	75	109	33	7	42	82	12
7	202	120	18	32	94	73	99	27	8	44	81	12
8	207	121	22	22	76	66	102	27	9	55	95	15
9	128	70	20	30	87	71	125	34	12	72	135	12
10	217	125	20	20	76	73	140	42	10	35	150	19

Table 1: test data of model acquisition trials in Room TR0110

Formulas:

$$\widehat{R}_S = \frac{R_S}{\sqrt{{R_S}^2+{G_S}^2+{B_S}^2}}$$

$$\widehat{G}_S = \frac{G_S}{\sqrt{{R_S}^2+{G_S}^2+{B_S}^2}}$$

$$\widehat{B}_S = \frac{B_S}{\sqrt{{R_S}^2+{G_S}^2+{B_S}^2}}$$

Color		Average over RGB in two rooms	Normalized average
Yellow ring	R	182.6	0.8654
	G	103.6	0.4910
	B	21	0.0995
Blue ring	R	22.8	0.2197
	G	75.2	0.7246
	B	67.8	0.6432
Orange ring	R	117.4	0.9597
	G	33.2	0.2714
	B	8.9	0.0728
Green ring	R	59.6	0.4584
	G	114.75	0.8807
	B	15.5	0.1192

Table 2: test data of 4 ring color identification in Room TR0110

Color and Position Identification (4 independent runs)

Trial/Rank	Detection	Return	R	G	B	Euclidean distance
1	Success	Blue	22	77	70	2.95
2	Success	Blue	28	65	72	12.19
3	Success	Blue	16	58	69	18.53
4	Success	Blue	24	99	56	26.59

Table 3: Data collected as the robot is detecting the blue ring

Rank	Detection	Return	R	G	B	Euclidean distance
1	Success	Yellow	177	108	17	8.17
2	Success	Yellow	180	89	15	15.99
3	Success	Yellow	156	78	17	37.13
4	Success	Yellow	132	114	20	51.67

Table 4: Data collected as the robot is detecting the yellow ring

Rank	Detection	Return	R	G	B	Euclidean distance
1	Success	Orange	118	42	10	8.89
2	Success	Orange	132	35	8	14.74
3	Success	Orange	144	39	9	27.23
4	Success	Orange	146	34	11	28.69

Table 5: Data collected as the robot is detecting the orange ring

Rank	**Detection**	**Return**	**R**	**G**	**B**	**Euclidean distance**
1	Success	Green	66	119	16	7.84
2	Success	Green	69	110	10	11.78
3	Success	Green	57	97	12	18.03
4	Success	Green	50	131	20	19.61

Table 6: Data collected as the robot is detecting the Green ring

Run	Rings detected	Rings correctly color-identified	Desired final position (cm)	Actual final position (cm)
1	4	4	(210,210)	(207.1,213.5)
2	4	4	(210,210)	(205.5, 204.5)
3	4	4	(210,210)	(210,215)
4	4	4	(210,210)	(208.5,213.5)

Table 7: Record of Robot’s position and number of rings detected and identified in the 4 runs

Section 3: Test Analysis

Color Calibration

Formulas:

$$d = \sqrt{(s_R - \mu_R)^2 + (s_G - \mu_G)^2 + (s_B - \mu_B)^2}$$

$$\mu_R = \frac{1}{\#trials} \cdot \sum_{i=1}^{\#trials} s_{R_i}$$

$$\mu_G = \frac{1}{10} \cdot \sum_{i=1}^{10} s_{G_i}$$

$$\mu_B = \frac{1}{10} \cdot \sum_{i=1}^{10} s_{B_i}$$

Color		Average over RGB	Normalized average	Standard deviation
Yellow ring	R	182.6	0.8654	42.291
	G	103.6	0.4910	23.453
	B	21	0.0995	2.828
Blue ring	R	22.8	0.2197	9.942
	G	75.2	0.7246	22.245
	B	67.8	0.6432	10.02
Orange ring	R	117.4	0.9597	32.449
	G	33.2	0.2714	10.02
	B	8.9	0.0728	2.514
Green ring	R	59.6	0.4584	17.161
	G	114.75	0.8807	27.184
	B	15.5	0.1192	2.877

Table 8: Mean, normalized average, and standard deviation of 4 rings in Room 0110

Based on Table 8, the following normal distributions are generated:

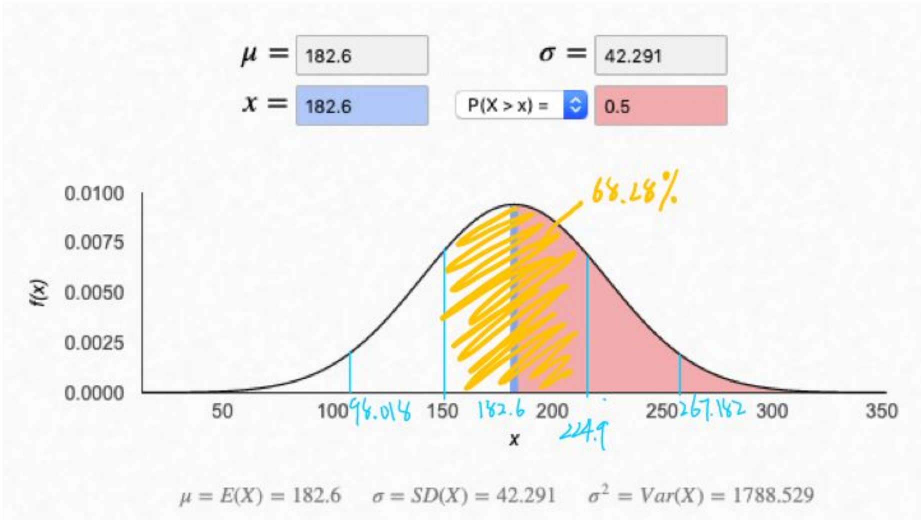

Chart 1: Normal distribution of R for the yellow ring

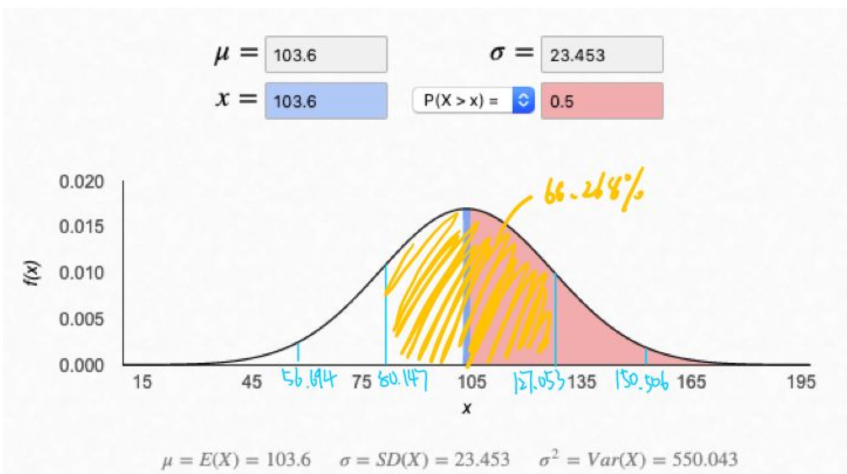

Chart 2: Normal distribution of G for the yellow ring

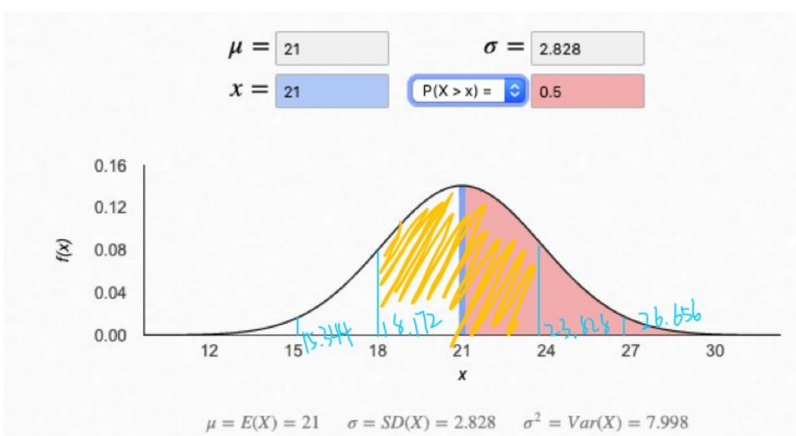

Chart 3: Normal distribution of B for the yellow ring

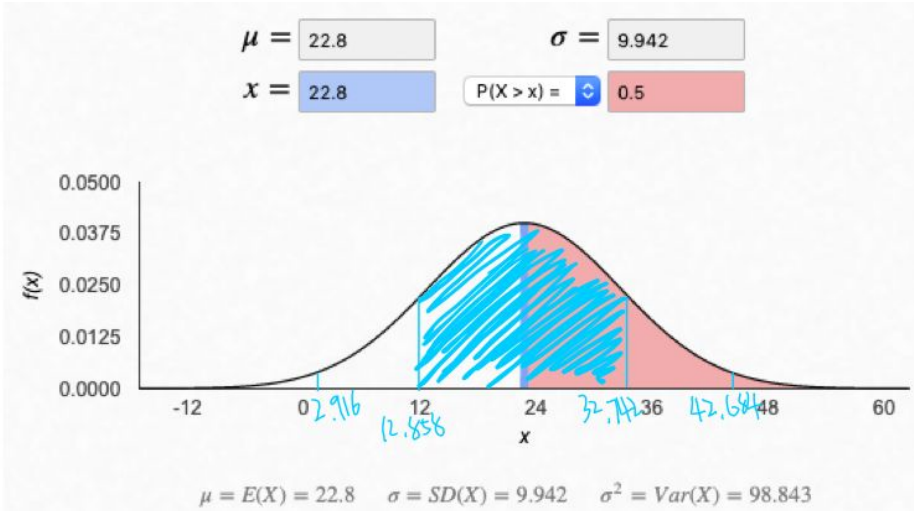

Chart 4: Normal distribution of R for the blue ring

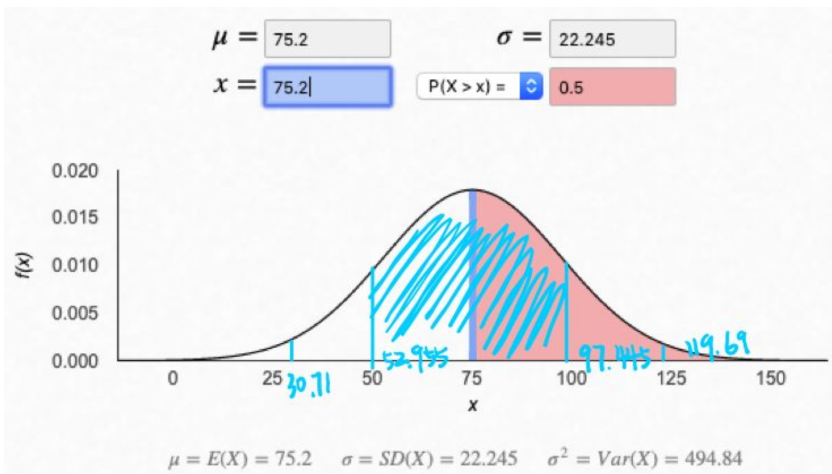

Chart 5: Normal distribution of G for the blue ring

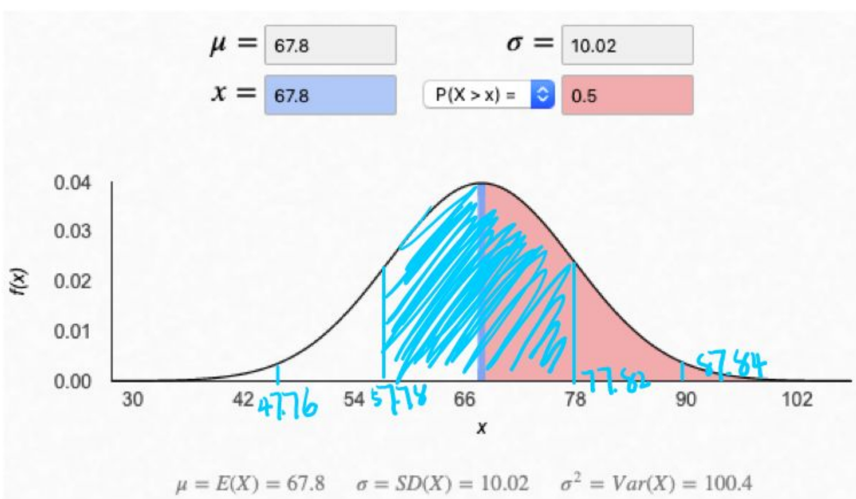

Chart 6: Normal distribution of B for the blue ring

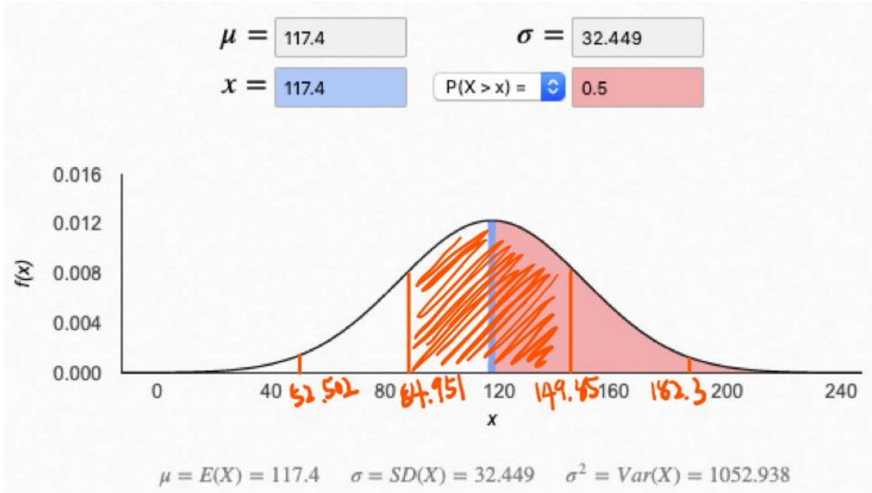

Chart 7: Normal distribution of R for the orange ring

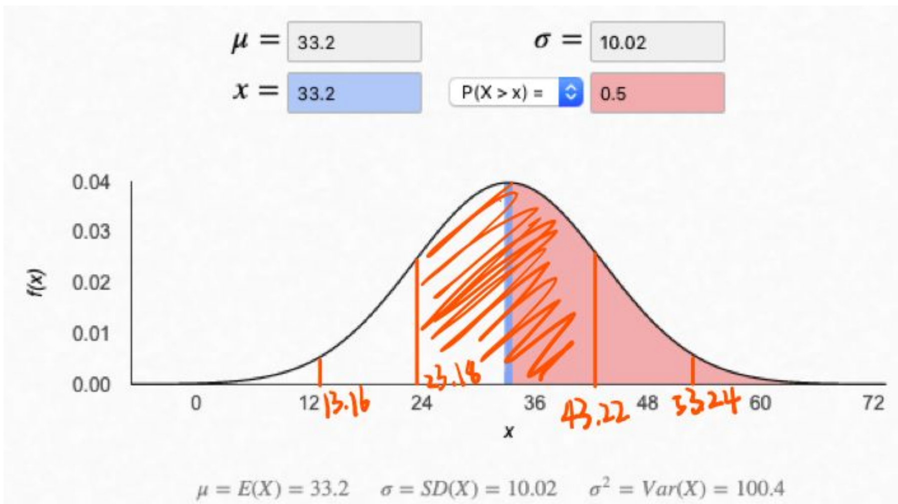

Chart 8: Normal distribution of G for the orange ring

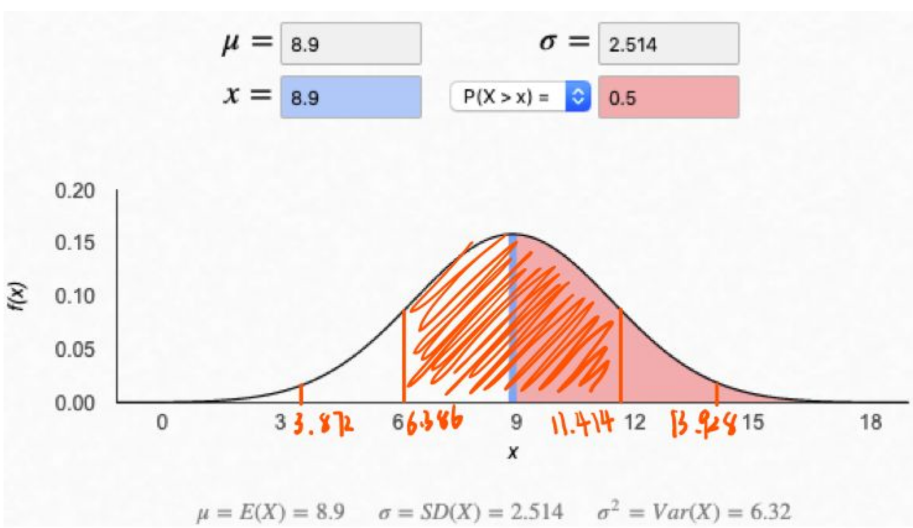

Chart 9: Normal distribution of B for the orange ring

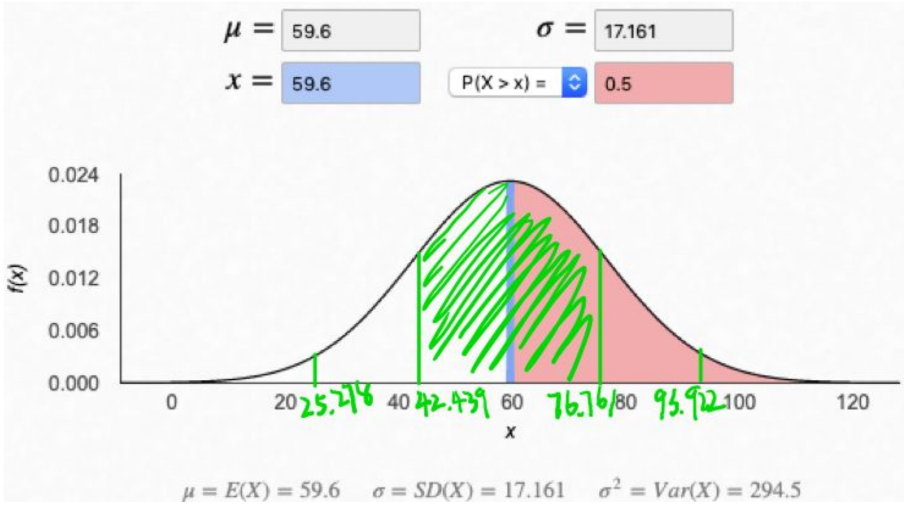

Chart 10: Normal distribution of R for the green ring

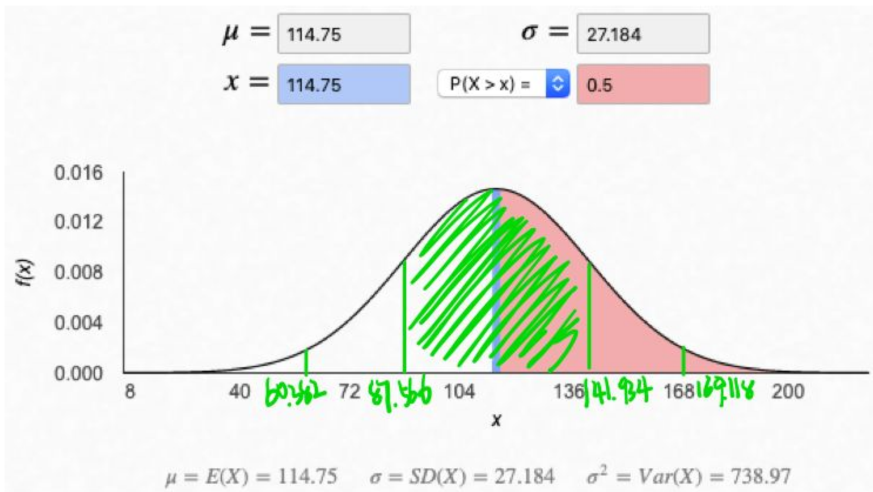

Chart 11: Normal distribution of G for the green ring

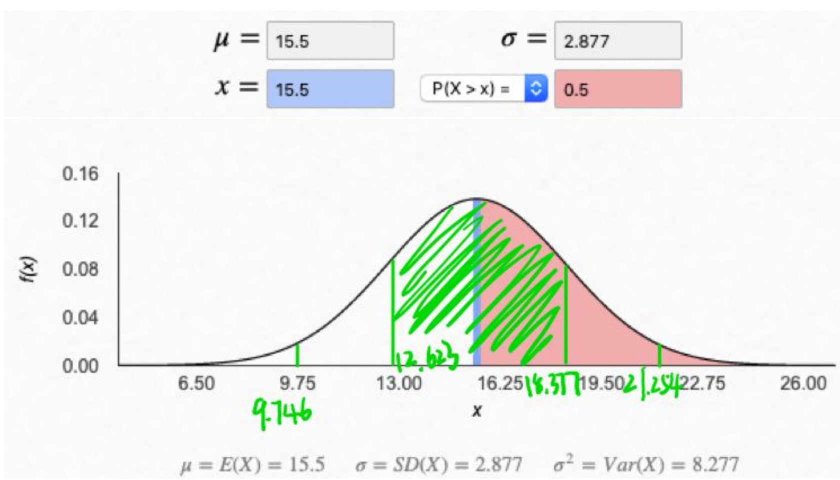

Chart 12: Normal distribution of B for the green ring

Color and position identification

The ranked Euclidean distance and the returned color of the algorithm are listed in Section 2.

Run	Actual final position (cm)	Expected final position (cm)	Euclidean distance
1	(210,210)	(207.1,213.5)	4.55
2	(210,210)	(205.5, 204.5)	7.11
3	(210,210)	(210,215)	5
4	(210,210)	(208.5,213.5)	4.03

Table 9: Euclidean distance between the actual and expected final position

Section 4: Observations and Conclusions

● **Are rank-ordering Euclidean distances a sufficient means of identifying ring colors? Explain in detail why or why not.**

The use of a rank-ordering Euclidean distance measurement has the advantage of providing one concise measurement of how close the three values (RGB) are. However, there are drawbacks that may affect the actual implementation of the method: This method does not allow us to see each of R, G, and B is close to the tested colors, which could give rise to problems in situations where 2 colors whose R, G and B values are very similar. In the actual implementation that requires to sample many different colors, the Euclidean distance rank-ordering method is very likely to encounter failure.

● **Is the standard deviation a useful metric for detecting false positives? In other words, if the ring color determined using the Euclidean distance metric, d, is incorrect, can this false positive be detected by using $\mu \pm 1\sigma$ or $\mu \pm 2\sigma$ values instead?**

Standard deviation is a useful metric on its own as it tells the variability of the set of data associated with the average value. In other words, it indicates how dispersedly distributed the sampled data is. Therefore, calculating the mean and standard deviation metric would help us to see whether an incorrect Euclidean distance sample obtained was tolerable. To illustrate this, assuming an incorrect sample is collected, the metric tells us that 68.2% of the data is within 1 standard deviation of each side of the mean and 95.4% of the data is within 2 standard deviations. So, there exists a false positive in the sample. The metric would allow us to evaluate how tolerable this Euclidean distance is. Therefore, using standard deviation and average is a useful metric for detecting false positives.

● **Under what conditions does the color sensor work best for correctly distinguishing colors?**

First, the positioning of the light sensor affected color detection. So, the first condition is to make sure the sensor is close enough to the ring to accurately measure its color. Then, observing from our further testing, the color sensor worked optimally when the distance is within one centimeter. This results from the fact that a large distance from the ring will allow the ambient light to interfere with the detection, reducing the accuracy of the returned values.

Section 5: Further Improvements

● **Depending on how you implemented your color classifier, can your results be improved by using one or more of the noise filtering methods discussed in class?**

We implemented the color classifier by taking several RGB measurements, taking the average of these measurements for each ring, and then normalizing the results to account for errors caused by ambient light. This means that we already filtered our data by taking the average and normalizing that average. One other noise filtering method that could easily be implemented is the removal of outliers that are present in the data collected over the sampling process. This would have the effect of centering the average towards an average that is more representing of the true absolute RGB average of any ring. Another noise filtering method that could be used is to take even more samples of the RGB values of the rings. By regression to the mean, the more samples we take, the more our average is going to approach the true absolute RGB average of the rings.

● **How could you improve the accuracy of your target detection?**

We could greatly improve the accuracy of our target detection by placing an ultrasonic sensor under the light sensor. This newly added sensor would have the purpose to detect objects in advance, and warn the color sensor that a target to be classified was detected. In our current version of the code, we only have the color sensor which is always polling the ground to see if the color is changing. This means that we totally rely on our color sensor to detect a target. Hence, adding the ultrasonic sensor would surely increase our accuracy because we would now have two sensors on the lookout for potential targets instead of one.