School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

Physical Distancing with BLE

by

Archie McKay [z5462377], Allan Zhou [z5309515], John Van Pham [z3216645], Marko Wong [z5309371]

1. Introduction

Throughout the COVID-19 pandemic, there have been many issues and challenges raised by people around the globe. One challenge in particular that most, if not all of us faced was the problem with social distancing. This project aims to explore ways to improve Bluetooth Low Energy (BLE) distance measurement for automating self distance regulations of 2 metres. While this task is mainly self regulated and enforced within the communities through human judgement, it makes us wonder if there is a way we can achieve a more practical and automated solution with the use of technology. The project will also dive into the feasibility of applying methods of currently available BLE Technology for accurate contact tracing. By studying the potential for developing RSSI-based ranging algorithms and applying them into real world cases, we are increasing our preparedness for future pandemics thus reducing the extent of future outbreaks.

2. Background

2.1 Connection/Proximity Testing

Connection/proximity testing is based on recording ‘contacts’ whenever two devices are able to scan each other. As BLE allows for connections of up to 75 metres, this is not a viable method for reliably detecting contacts within 2 metres.

2.2 Fingerprinting

Fingerprinting relies on having a pre-deployed infrastructure of nodes that can be used to locate a user. Unlike triangulation it relies on having a pre-calculated dataset of recorded positions, which allows it much greater accuracy within a space. Fingerprinting models can allow for higher accuracy of position. The drawbacks of fingerprinting however are that because they use pre recorded data, the owner of the space must record this data by walking around the space and also limits their ability to change the layout of the space at a later date. They also rely on having a higher number of pre-deployed nodes, and this combination of increased cost and difficulty to set up has led us to use other options.

2.3 Radio Signal Strength Indicator (RSSI) Ranging

Radio Signal Strength Indicator (RSSI) ranging is a value that is provided by radio chips that indicates the power of a received signal. This requires users to be able to reliably transmit at a known power and be able to accurately measure a received power. A path loss propagation model can then be used to calculate the distance from the received signal.

The multipath fading effect occurs as transmitted waves can be reflected and travel across different paths. This leads to their interference creating zones of extremely high and low attenuation as the difference between the transmitter and receiver's distance changes. To account for these changes, pathloss propagation models such as the free-space propagation model or the log-distance path loss model can be used. Kalman filtering can be used to predict and account for variables in a measurement system, such as objects or body blocking [4].

2.4 Trilateration and Multilateration

Lateration can be used when ‘anchor’ nodes are available. Trilateration relies on three of these anchor nodes to be in place, and then uses trigonometry to determine a user's location. In previously set up systems, distance error has been as low as 0.5 metres. Due to the reduced complexity of calculation that is used in triangulation as compared to using neural networks/machine learning that are typically used in fingerprinting and other methods, this allows for greater energy efficiency and hence greater battery life, as well as reducing the cost of implementation for both pre installed hardware and the computational ability of the mobile device.

3. Methodology

The methodology used to achieve our aim of finding ways to improve BLE distancing for COVID-19 close contact tracing included using an arduino nano’s and phone’s RSSI values to infer distance. To find ways to improve BLE distancing for a more accurate close contact tracing, the team experimented with various situations and then compared this to the standard scenario. The experimental stage will be where the team attempts to find the ideal constants for the propagation model depending on the different situations. The practical demo stage will involve applying those ideal constants where applicable to the current situation to create alerts when two devices are within two metres of each other.

3.1 Hardware Used

The hardware utilised in this project included two arduino nanos and a phone with the “nRC Connect” mobile app. The reasoning behind this selection was that it required the least amount of setup time and that in turn allowed more time for experimentation. Since the team were taught how to use arduino nano and the nRC Connect, utilising these devices required the least amount of setup time compared to using other unfamiliar devices. For the experimental part of the project, the team used a phone and an arduino nano.

3.2 Communication Method

For the experimental part of this project, RSSI values were taken after manually connecting two BLE devices. This is much more reliable at continuously reading RSSI to infer distance compared to a connectionless approach where the devices need to continuously scan for all devices and read the RSSI of the target device after finding it. For our intended implementation of a solution, one nano assumes the role of peripheral and the other a central (depending on timing). Both devices are concurrently advertising and scanning for other devices. Whichever device finds the other first, will then connect and assume the role of central while the other the role of peripheral. When this connection is made, the central can write to the characteristics of the peripheral to give it its contextual information, and then the peripheral can make an estimate of distance taking these into account. Once complete, it can send this result back to the central through a notification on a characteristic that the central was subscribed to. This had complications as discussed later in this report and was ultimately not implemented fully.

3.2.1 Alert Method

For our implementation the alerts are made through a buzzer and LED and a print out to the serial monitor. Note however that we first looked into sending this with MQTT to AWS Cloud along with the address of the contact device and the time, however this had complications. An arduino nano peripheral can only be connected to one central at a time, and in order to send data via MQTT, a third device (our laptops) would need to be involved. The nano would have to break its scanning and advertising loops to do so, and then sync up with the laptop instead of other nanos, and this parallelism could not be implemented reliably.

3.3 Profiles / Situation

There were various different profiles/situations we tested to get the ideal constant for the propagation model for that particular situation. By getting these results we planned to make our practical demo adaptable to the current situation and ideally produce more accurate close contact tracing. In terms of which situations we tested, this includes: all variance in tilts (x, y and z axis), time of day, in pocket and on wrist. For each situation the team took 5 RSSI readings at 6 distances (0, 50, 100, 150, 200, 250 cm). The distance markings are marked by tape and were measured using a tape measure as seen in Fig 3.3.A. During experimentation the team kept every potential factor constant and only changed the factor being measured. For example, for the Y-axis situation the team set the phone on the angle we are measuring as seen in Fig 3.3.B then placed it at each distance marking, whilst keeping the devices used and position of objects the same.

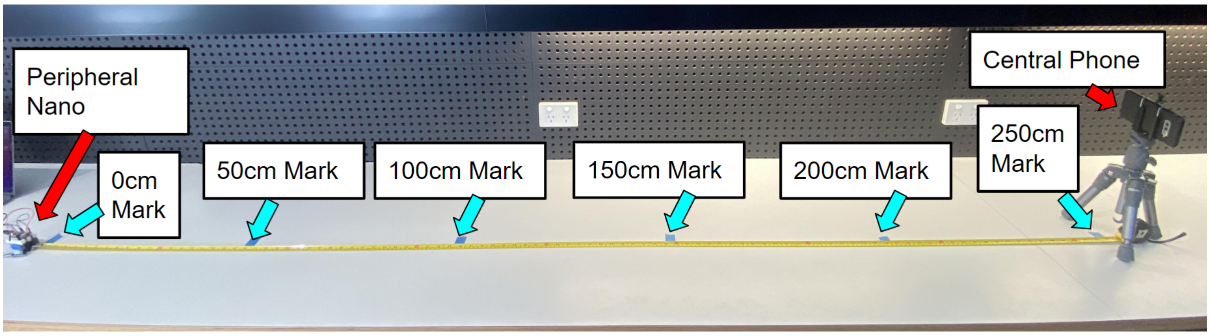


Fig 3.3.A: Experiment Setup

Fig 3.3.B Measuring Tilt Angle

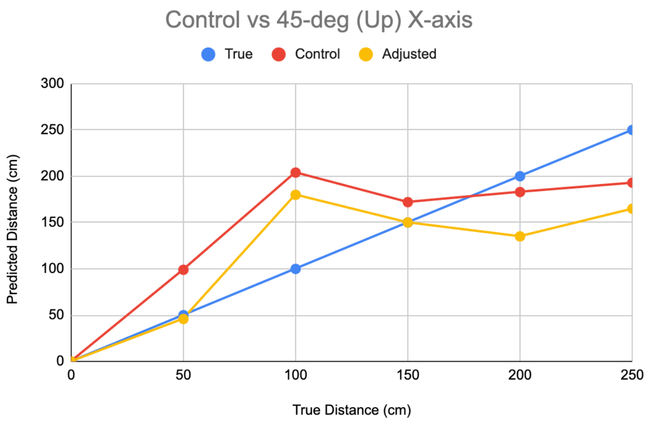
After collecting all the RSSI readings at different distances, the RSSI results were plotted against distance on a graph and the gradient and y-intercept were calculated. Using these gradient and y-intercept results from the different situations, different profiles are generated for each situation. The results are then compared to a basic (control) algorithm without the profiles.

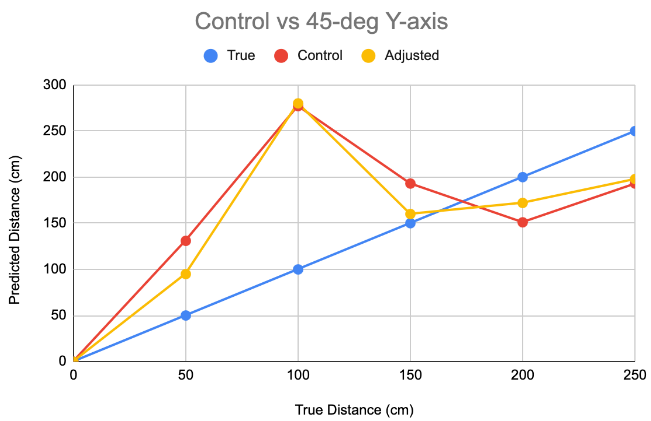
3.4 Environment

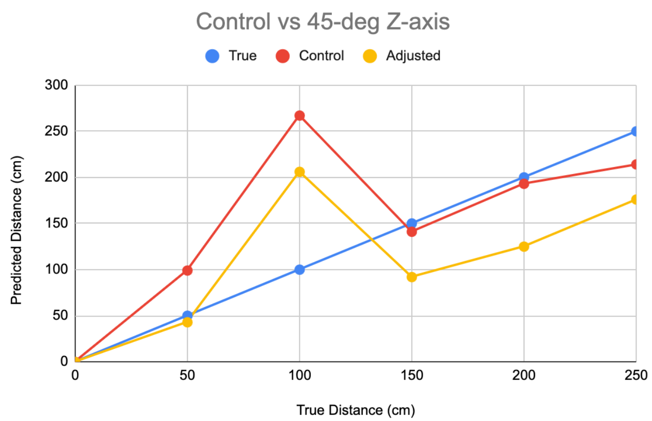
The environment the team performed these tests in was in UNSW Lyre Lab G12. This location had many other BLE devices active which simulates real world interference.

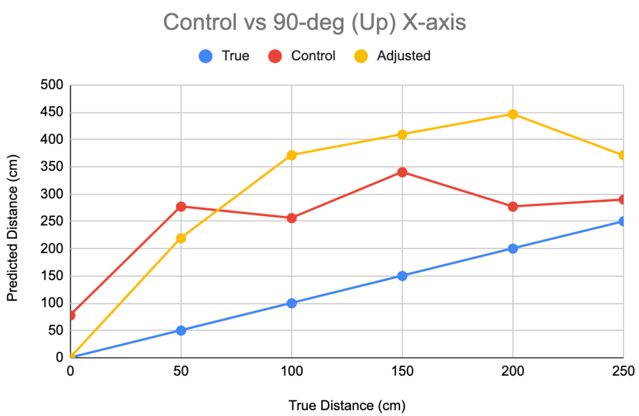
4. Results

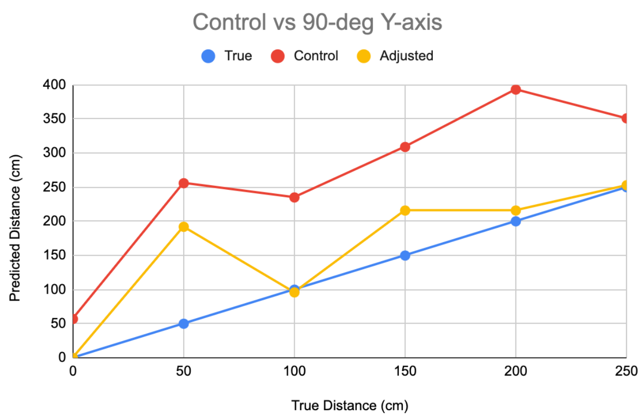
In this section we delve into understanding the reliability of using the adjusted RSSI Based BLE measurement algorithms. Below are some results of our Control algorithm and Adjusted algorithms displayed alongside true values. By tabulating and averaging distance predictions of the original control algorithm and adjusted algorithms against the true distance, we can find the percentage improvement in the new algorithm (or lack thereof). Refer to the appendix for the complete set of results.

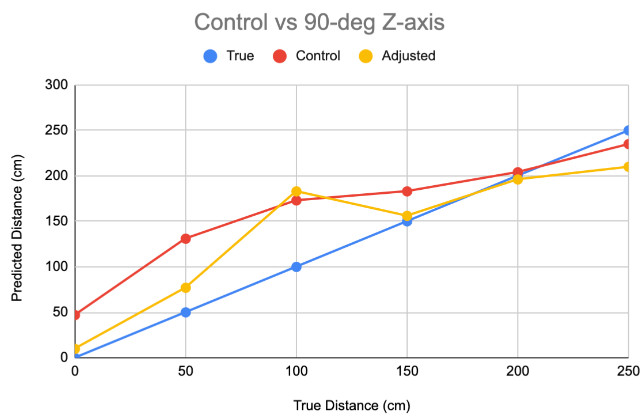


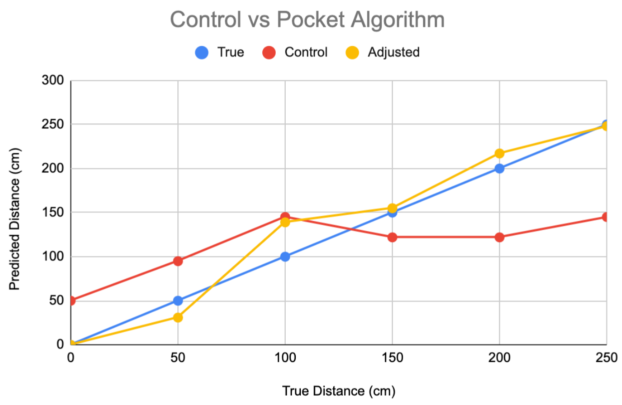


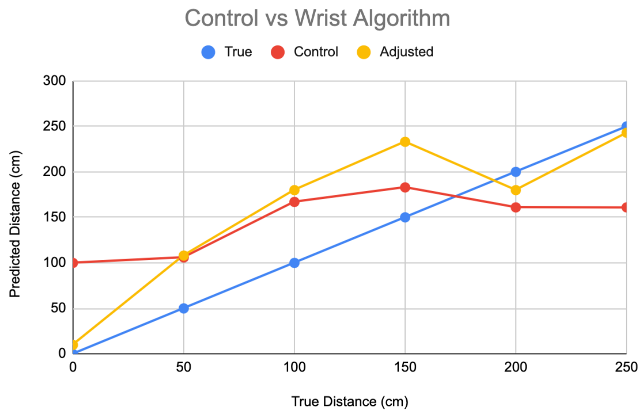


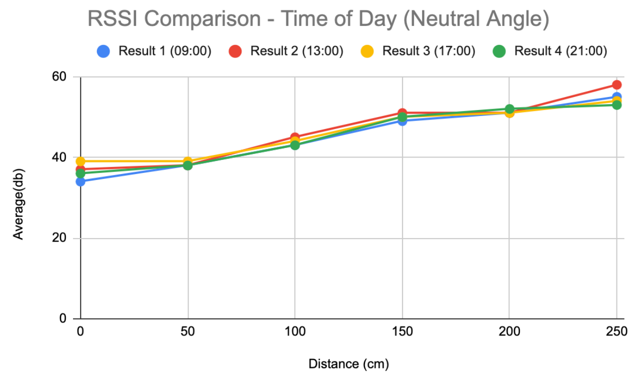






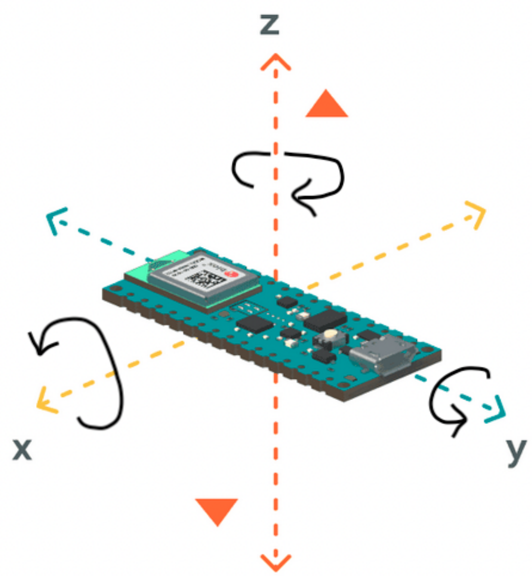






	RSSI = m * distance + c		AVERAGE ERROR		
DESCRIPTION	**m**	**c**	**CONTROL**	**ADJUSTED**	**CTRL - ADJ**
X 45-deg (down)	0.0544	29.9	52.51%	33.27%	19.25%
X 90-deg (down)	0.0488	40.5	123.69%	67.91%	55.78%
X 45-deg (up)	0.0672	31.9	49.59%	30.90%	18.69%
X 90-deg (up)	0.0264	39.2	158.23%	190.61%	-32.38%
Y 45-deg	0.0775	28.6	82.99%	62.29%	20.70%
Y 90-deg	0.0830	37.0	157.98%	68.24%	89.74%
Z 45-deg	0.0614	32.3	57.78%	45.15%	12.63%
Z 90-deg	0.0752	30.2	53.00%	31.80%	21.20%
Z 135-deg	0.0757	29.1	57.33%	37.77%	19.57%
Z 180-deg	0.0787	32.1	132.92%	71.07%	61.85%
Control	0.0954	23.5			

	RSSI = m * distance + c		AVERAGE ERROR		
DESCRIPTION	**m**	**c**	**CONTROL**	**ADJUSTED**	**CTRL - ADJ**
Pocket	0.0643	43.0	46.93%	17.93%	29.01%
Wrist	0.1120	37.8	51.22%	52.83%	-1.61%
Control	0.1080	28.9			



5. Discussion

5.1 General

The experiment produced mixed results with regards to being able to reveal trends and demonstrate accuracy. It was evident that noise and interference played a major role. It is noted that these results were primarily taken in the CSE Lyre Lab with the exception of the Time-of-Day results which were taken at home. We believe this had an impact on the stability of the results as those taken in the lab were erratic while those taken at home were comparably more stable and followed a more obvious trend.

5.2 Angles / Scenarios

5.2.1 X-axis: Rotation from 90-deg down to 90-deg up

It can be seen that the adjusted and control algorithms performed similarly for smaller angles, with the adjusted algorithms improving the distance estimate as expected. However, the results are less meaningful at large angles as can be seen from the deviations from the true distance. We noticed large errors at particular distances which was most likely due to interference.

5.2.2 Y-Axis: Rotation from 0-deg to 90-deg

Similar to the X-axis results, the adjusted algorithm performs slightly better than the control algorithm for the smaller angles but results are not meaningful for larger angles. There were also pronounced errors for certain distances.

5.2.3 Z-axis - Rotation from 0-deg to 180-deg

The adjusted algorithm produced improved results for most angles. Notably, the results did not worsen at any particular angle of rotation. Note that this is the only angle that cannot be controlled in a real-world scenario as two separate devices will never know their orientation in a global coordinate system, i.e. they do not know if they are facing each other, away from each other, or any angle in between.

5.2.4 Pocket, Wrist, and Time of Day

Adjusting for the pocket shows an improvement over the control while adjusting for the wrist performs similarly to the control. Also note that time-of-day (at least at home) did not affect our RSSI readings.

5.3 Accuracy and Noise

It is obvious from the results that there was a significant impact due to environment noise and interference. In our particular arrangement, we often noticed erratic results at the 100cm mark. We speculate that this is due to a number of reasons. The first being the arrangement of chairs and seating in the lab, and the other being the sheer number of devices in the vicinity of the experiment which interfered with the signal strength the Arduino Nano was receiving.

5.4 Main Prototype: Connection-Based

This section describes the implementation of RSSI-based ranging for a solution involving two Arduino Nano devices. We investigated this solution with various protocols for connecting and obtaining RSSI values and communication between the devices.

The intended main solution involved each nano behaving as both a peripheral and central at the same time. The reason for this was that the devices always need to be scanning for nearby relevant devices (scanned by name), and when one is found, one of the devices needs to assume the role of peripheral and the other central so that they can share relevant information such as angle of orientation and scenario (e.g. in pocket). The peripheral can then perform the calculation and notify the outcome with the central that was subscribed to the appropriate characteristic. Without this peripheral/central dual role for each nano there would be minimal information sharing and therefore the context is lost and the adjusted algorithms can not be implemented to their potential. Additionally, this connection allows the devices to take an average of RSSI values to produce a more stable result.

Although the above solution is theoretically ideal, our attempts at this solution ran into complications. Firstly, we found that the connection of one nano to another nano was unstable and we constantly lost connection. This did not happen when connecting with a mobile phone. It was a matter of probability if an arduino nano would connect to the other arduino nano with most attempts failing. The devices are designed to indefinitely search for other devices and on occasion did connect correctly, but were not as responsive as desired.

Another challenge was that there were concurrency issues. There is a delay when connecting to a peripheral and while this is happening the peripheral could be trying to make the same connection back to the central. This resulted in obscure bugs that were difficult to troubleshoot.

Due to this, one of our recommendations is that the Nano itself might not be the ideal tool to record contact, but a more reliable solution could be implemented through modern-day phones or wrist-watches. This was evident by how much more stable a connection from a mobile phone was to the nano. The experiments were conducted using a mobile connection to the nano for this very reason as it would hold this connection almost indefinitely, which was not possible between two nano devices. Similarly, for our presentation we will demonstrate with a manual connection to a nano using a mobile phone as this is the most reliable way to present the findings of the experiments.

5.5 Alternative: Advertise-Only Protocol

This alternative was trialled which involves devices only using advertised data to conduct distance-based ranging. This is likely a more robust solution since it does not rely on a connection. A contact-detecting device can be programmed with a specific service UUID so that performing a distance calculation for all possible BLE signals is prevented, and only devices with this UUID are investigated in terms of their distance to the current device.

Note that this may result in inconsistent results between two devices as they conduct their calculations independently of each other with little context as to the state of the other device (limited by the packet size that can be advertised).

5.6 Suitability

Our investigation and attempts at a contact-tracing solution using Arduino Nano BLE was met with a number of challenges as discussed. However, we believe there is still merit in further investigation.

Note that contact-tracing based on distance does not necessarily need to be a responsive and quick solution. Most regulatory requirements were based on detecting contact based on close proximity to another person over a period of minutes rather than seconds. Hence, it is only necessary for devices to make one good reading in order to detect the contact and they have the opportunity to do so over a period of minutes. It is probable that this opportunity will occur over that timeframe. As a result of this however, it is not advisable to overcomplicate the algorithms in this approach, as setting the sensitivity too high may result in excessive false positives that need to be investigated and cause further overhead. Additionally, our experimental results show that the added complexity with measuring and sharing angles and certain other variables does not always perform well and is unreliable. Based on this, a mobile or watch-based solution may be the most suitable approach since these devices are often unobstructed and do not require adjustment through complex algorithms.

6. Conclusion

6.1 Summary

Our investigations demonstrated that adjusting RSSI-based ranging algorithms for different angles and scenarios can produce more accurate results than keeping a fixed algorithm. However, due to interference and therefore the erratic nature of RSSI values, this may not be practical to implement for a real-world solution.

Between an advertise-only or connection-based approach to obtain distance measurements, it is likely that an advertise-only approach is more robust but will lack the full context of the other device. However, given the significant inaccuracy due to external variables, attempting to produce a more accurate distance measurement is not worthwhile.

An ideal solution may be through implementing distance-based contact tracing using mobile phones or wrist watches with an advertise-only approach. This is because it is likely these devices will have the opportunity to get a strong signal due to their constant exposure to other devices.

6.2 Recommendations and Future Work

Future work on this topic could focus specifically on either a mobile-mobile, mobile-smartwatch, or smartwatch-smartwatch implementation. Rather than focusing on making the distance algorithm more accurate for angles or scenarios, focus could be on how likely these devices are to catch a close-contact event in an ideal scenario: clear line of sight. Perhaps devices can advertise a value which indicates a calibration of signal strength so that the other device can adjust accordingly, as different devices produce different signal strengths. These signals can additionally be scaled depending on being indoors or outdoors, or for those environments that have major impact.

Bibliography

[1] Larsson, J. (2015). *Distance Estimation and Positioning Based on Bluetooth Low Energy Technology*. [online] Available at: https://www.diva-portal.org/smash/get/diva2:859549/FULLTEXT01.pdf.

[2] Admin (2020). *Can Bluetooth Be Used To Measure Distance?* [online] Bluetooth Tech World. Available at: https://bluetoothtechworld.com/can-bluetooth-measure-distance/.

[3] Etzlinger, B., Nußbaummüller, B., Peterseil, P. and Hummel, K. (2021). *Distance Estimation for BLE-based Contact Tracing -A Measurement Study*. [online] Available at: https://arxiv.org/pdf/2101.09075.pdf.

[4] Shue, S., Conrad, M., (n.d.). *Reducing the Effect of Signal Multipath Fading RSSI-Distance Estimation using Kalman Filters.* [online] Available at: https://dl.acm.org/doi/pdf/10.5555/2962686.2962691.

[5] Wu, R.-H., Lee, Y.-H., Tseng, H.-W., Jan, Y.-G. and Chuang, M.-H. (2008). *Study of characteristics of RSSI signal*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICIT.2008.4608603.

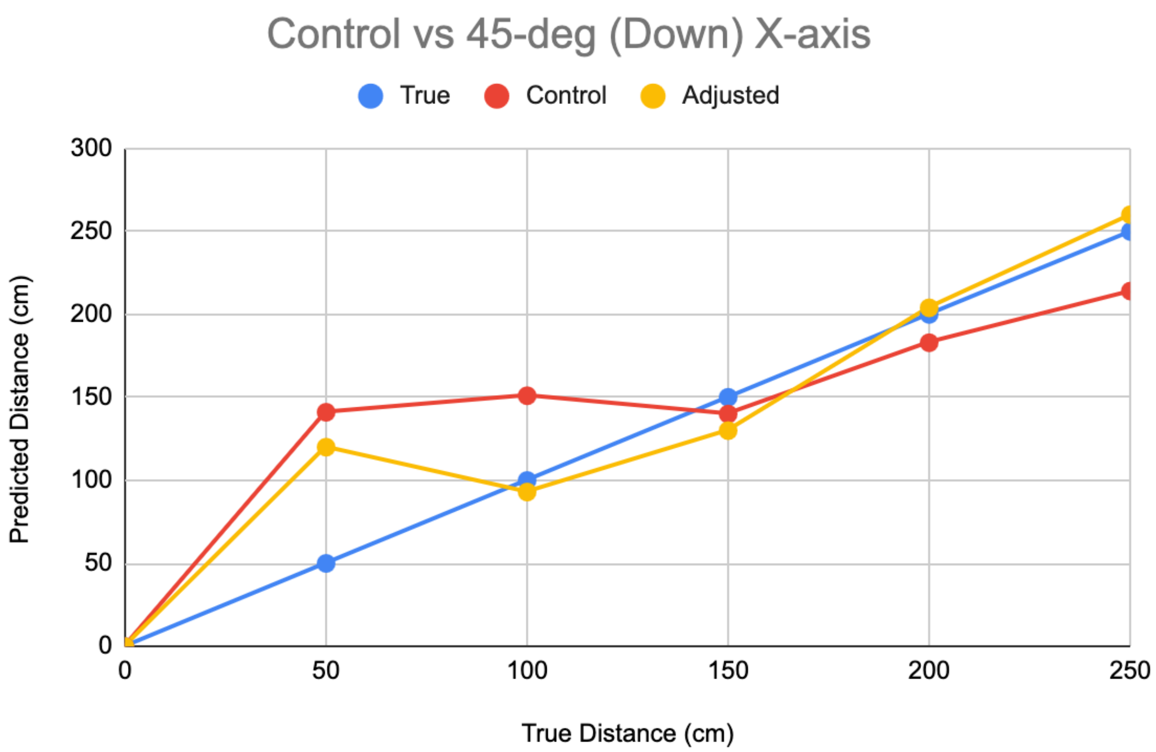
[6] Cannizzaro, D., Zafiri, M., Jahier Pagliari, D., Patti, E., Macii, E., Poncino, M. and Acquaviva, A. (2019). A Comparison Analysis of BLE-Based Algorithms for Localization in Industrial Environments. *Electronics*, 9(1), p.44. doi:https://doi.org/10.3390/electronics9010044.

[7] Rida, M.E., Liu, F., Jadi, Y., Algawhari, A.A.A. and Askourih, A. (2015). *Indoor Location Position Based on Bluetooth Signal Strength*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICISCE.2015.177.

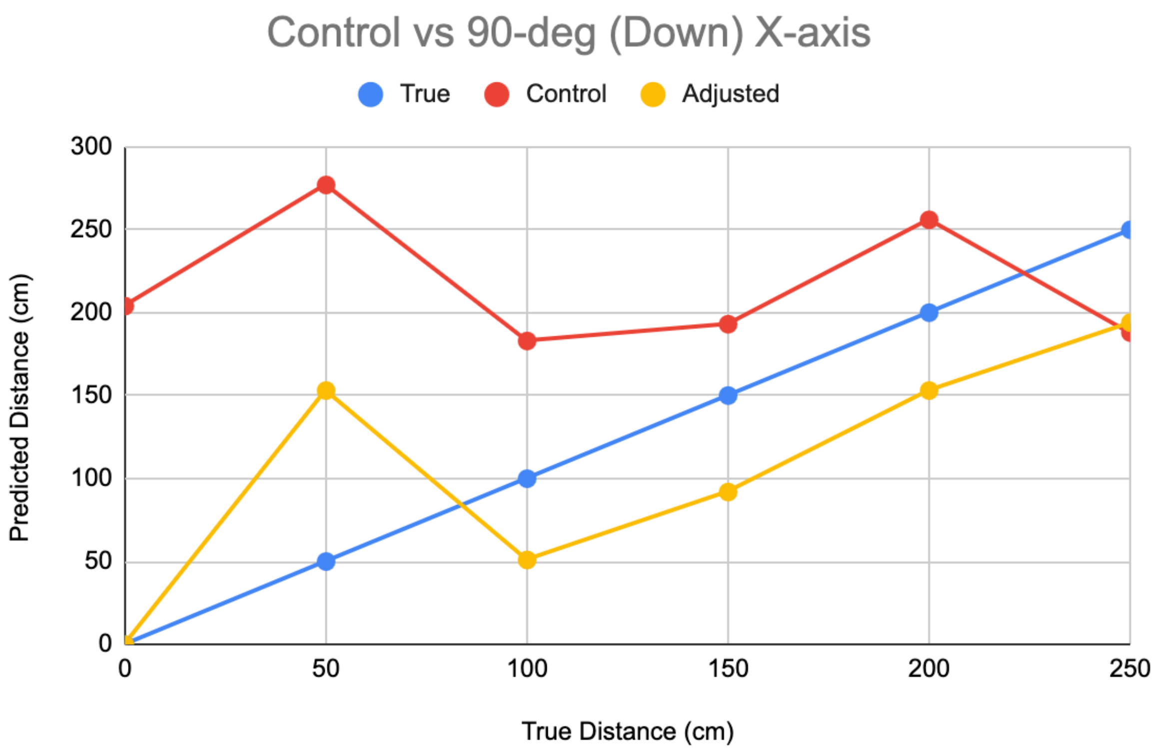
[8] Ng, P.C., Spachos, P. and N. Plataniotis, K. (2021). COVID-19 and Your Smartphone: BLE-Based Smart Contact Tracing. *IEEE Systems Journal*, pp.1–12. doi:https://doi.org/10.1109/jsyst.2021.3055675.

Appendix

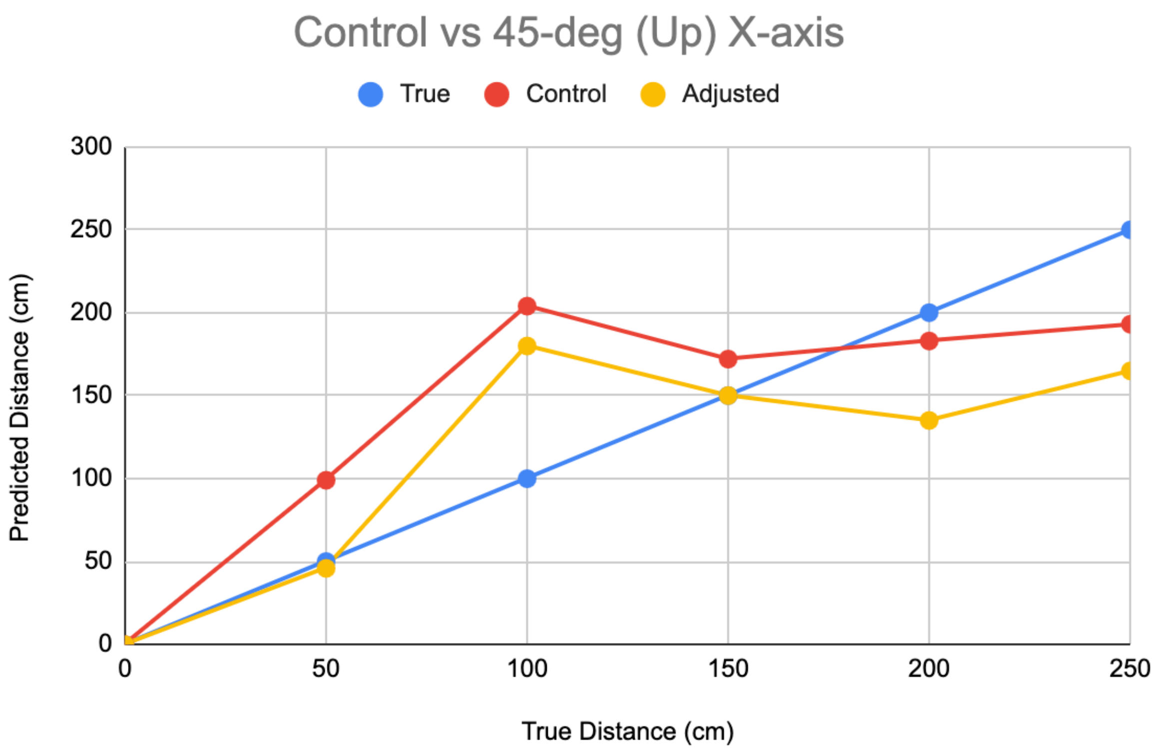
COMPARISON: CONTROL VS 45-DEG (DOWN) X-AXIS					
TRUE DIST	CONTROL ALGORITHM		ADJUSTED ALGORITHM		
Dist (cm)	Dist (cm)	Ctrl / True	Dist (cm)	Adj / True	|Ctrl| - |Adj|
0	0		0		
50	141	182.0%	120	140.0%	42.00%
100	151	51.0%	93	-7.0%	44.00%
150	140	-6.7%	130	-13.3%	-6.67%
200	183	-8.5%	204	2.0%	6.50%
250	214	-14.4%	260	4.0%	10.40%
				Average	19.25%



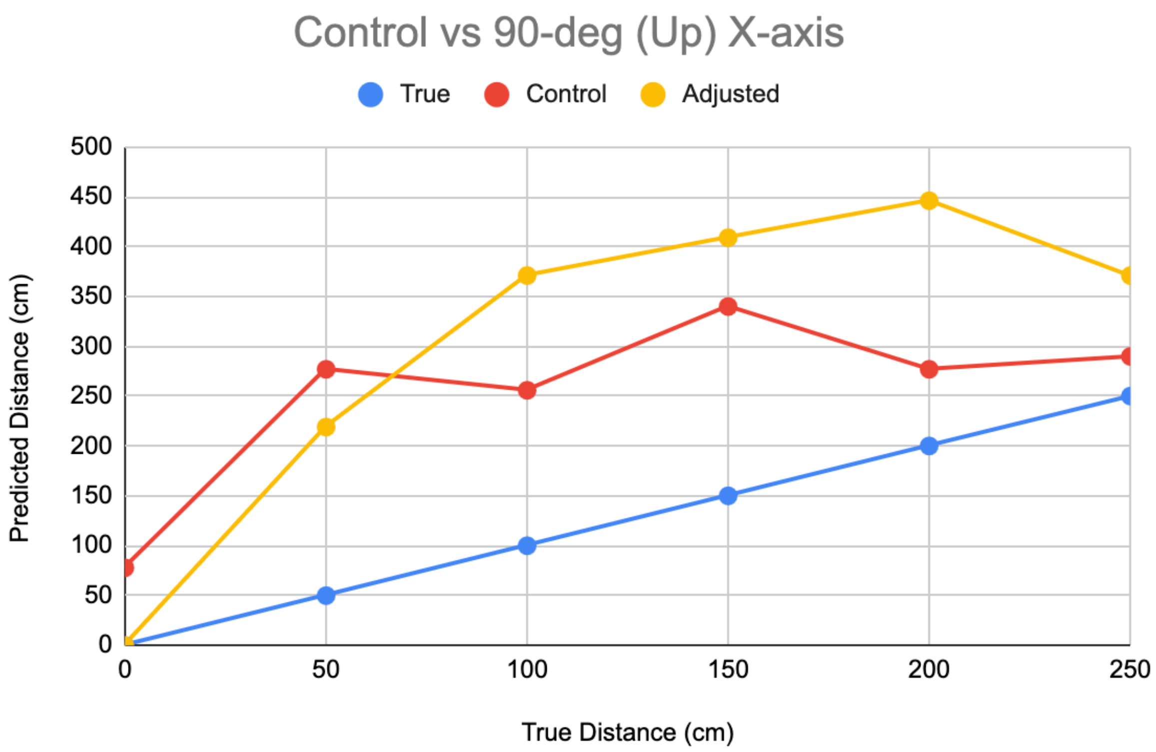
COMPARISON: CONTROL VS 90-DEG (DOWN) X-AXIS					
TRUE DIST	CONTROL ALGORITHM		ADJUSTED ALGORITHM		
Dist (cm)	Dist (cm)	Ctrl / True	Dist (cm)	Adj / True	|Ctrl| - |Adj|
0	204		0		
50	277	454.0%	153	206.0%	248.00%
100	183	83.0%	51	-49.0%	34.00%
150	193	28.7%	92	-38.7%	-10.00%
200	256	28.0%	153	-23.5%	4.50%
250	188	-24.8%	194	-22.4%	2.40%
				Average	55.78%



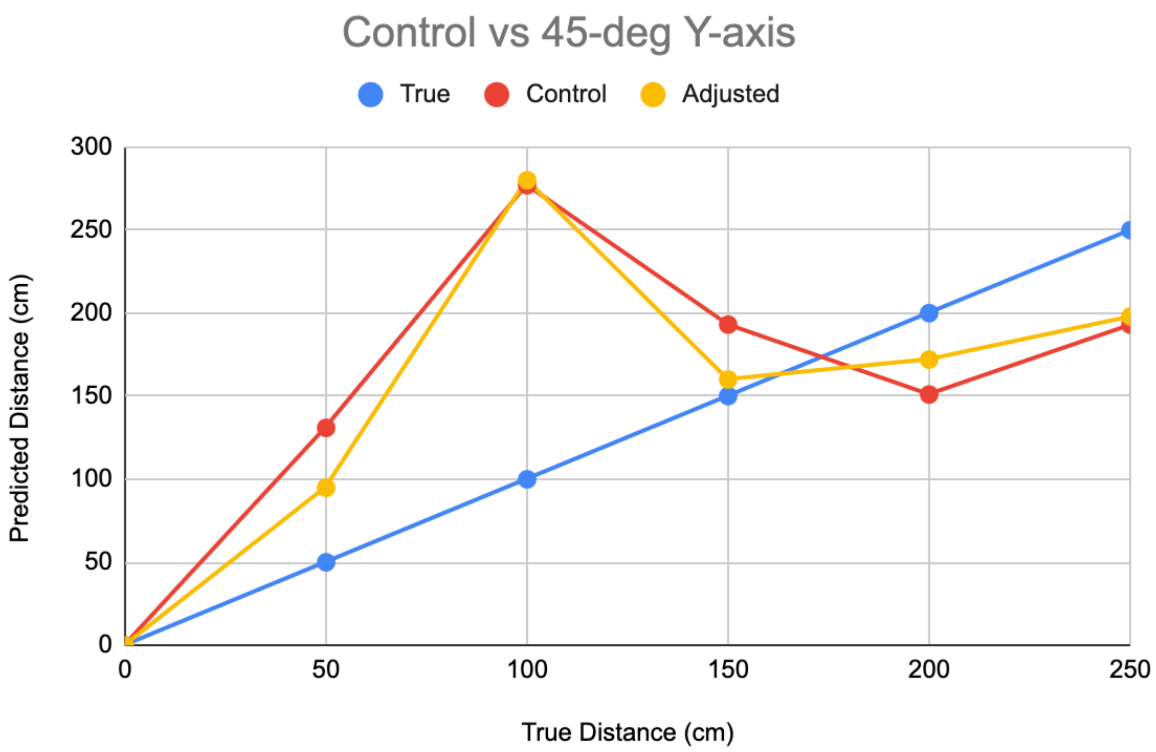
COMPARISON: CONTROL VS 45-DEG (UP) X-AXIS					
TRUE DIST	CONTROL ALGORITHM		ADJUSTED ALGORITHM		
Dist (cm)	Dist (cm)	Ctrl / True	Dist (cm)	Adj / True	|Ctrl| - |Adj|
0	0		0		
50	99	98.0%	46	-8.0%	90.00%
100	204	104.0%	180	80.0%	24.00%
150	172	14.7%	150	0.0%	14.67%
200	183	-8.5%	135	-32.5%	-24.00%
250	193	-22.8%	165	-34.0%	-11.20%
				Average	18.69%



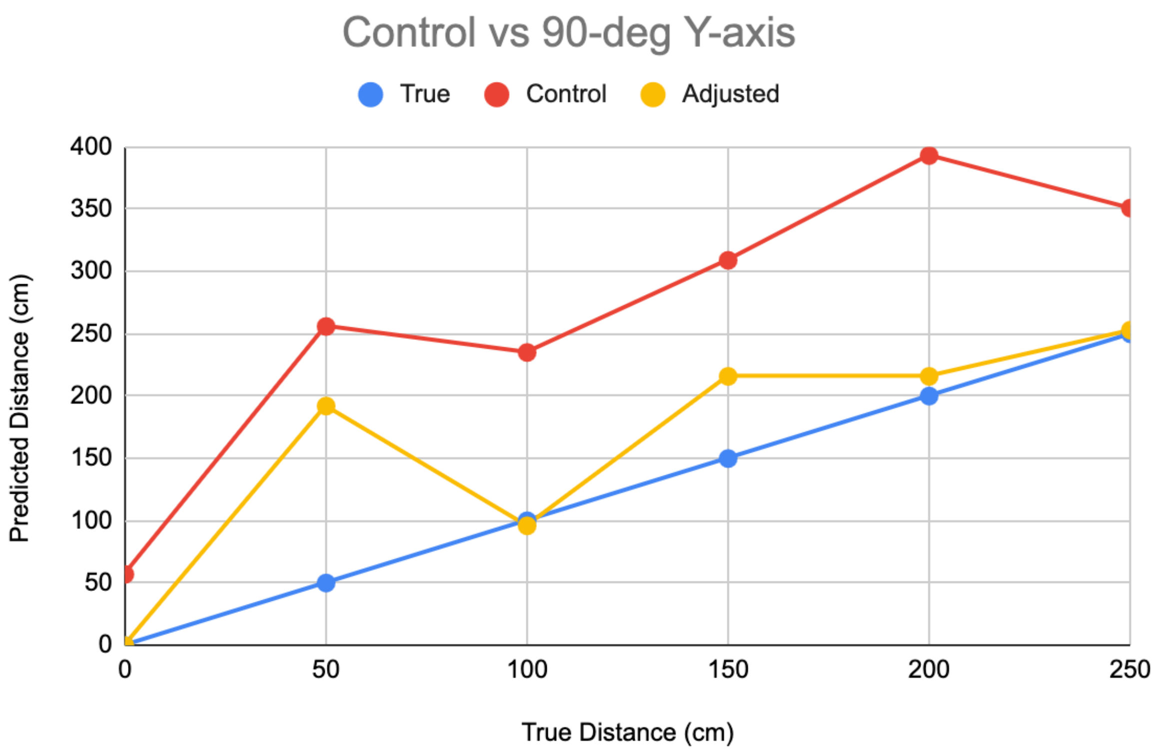
COMPARISON: CONTROL VS 90-DEG (UP) X-AXIS					
TRUE DIST	CONTROL ALGORITHM		ADJUSTED ALGORITHM		
Dist (cm)	Dist (cm)	Ctrl / True	Dist (cm)	Adj / True	|Ctrl| - |Adj|
0	78		0		
50	277	454.0%	219	338.0%	116.00%
100	256	156.0%	371	271.0%	-115.00%
150	340	126.7%	409	172.7%	-46.00%
200	277	38.5%	446	123.0%	-84.50%
250	290	16.0%	371	48.4%	-32.40%
				Average	-32.38%



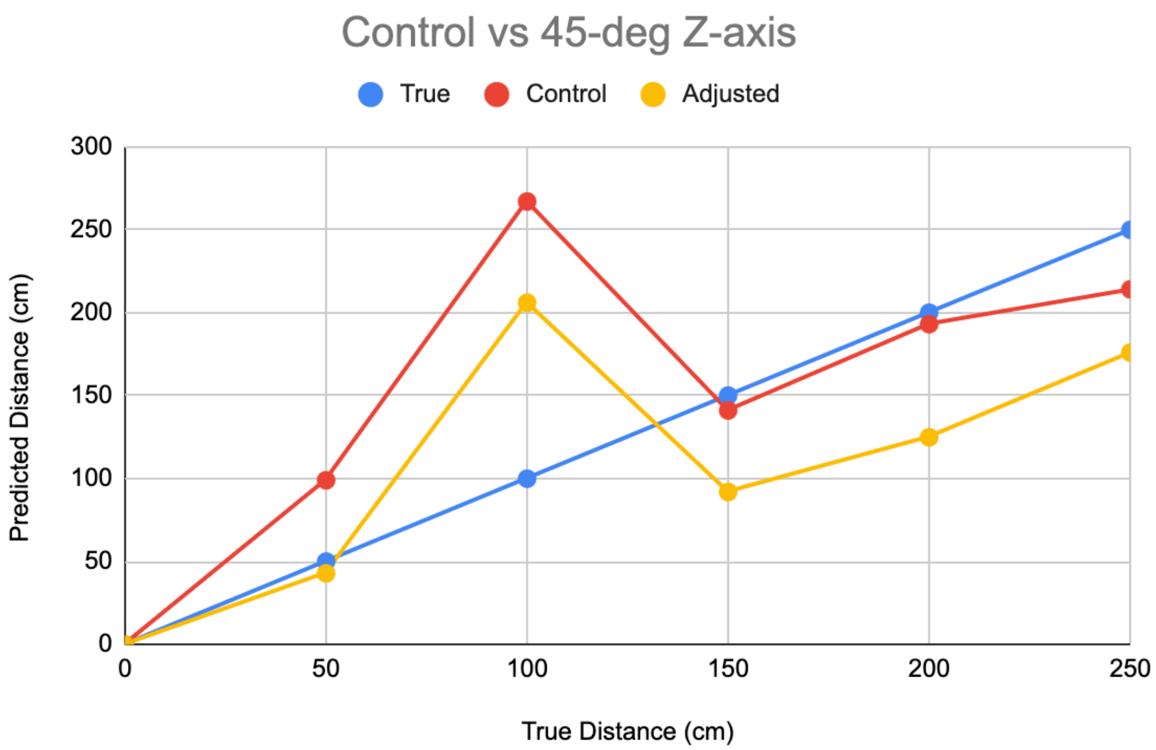
COMPARISON: CONTROL VS 45-DEG Y-AXIS					
TRUE DIST	**CONTROL ALGORITHM**		**ADJUSTED ALGORITHM**		
Dist (cm)	**Dist (cm)**	**Ctrl / True**	**Dist (cm)**	**Adj / True**	**|Ctrl| - |Adj|**
0	0		0		
50	131	162.0%	95	90.0%	72.00%
100	277	177.0%	280	180.0%	-3.00%
150	193	28.7%	160	6.7%	22.00%
200	151	-24.5%	172	-14.0%	10.50%
250	193	-22.8%	198	-20.8%	2.00%
				Average	20.70%



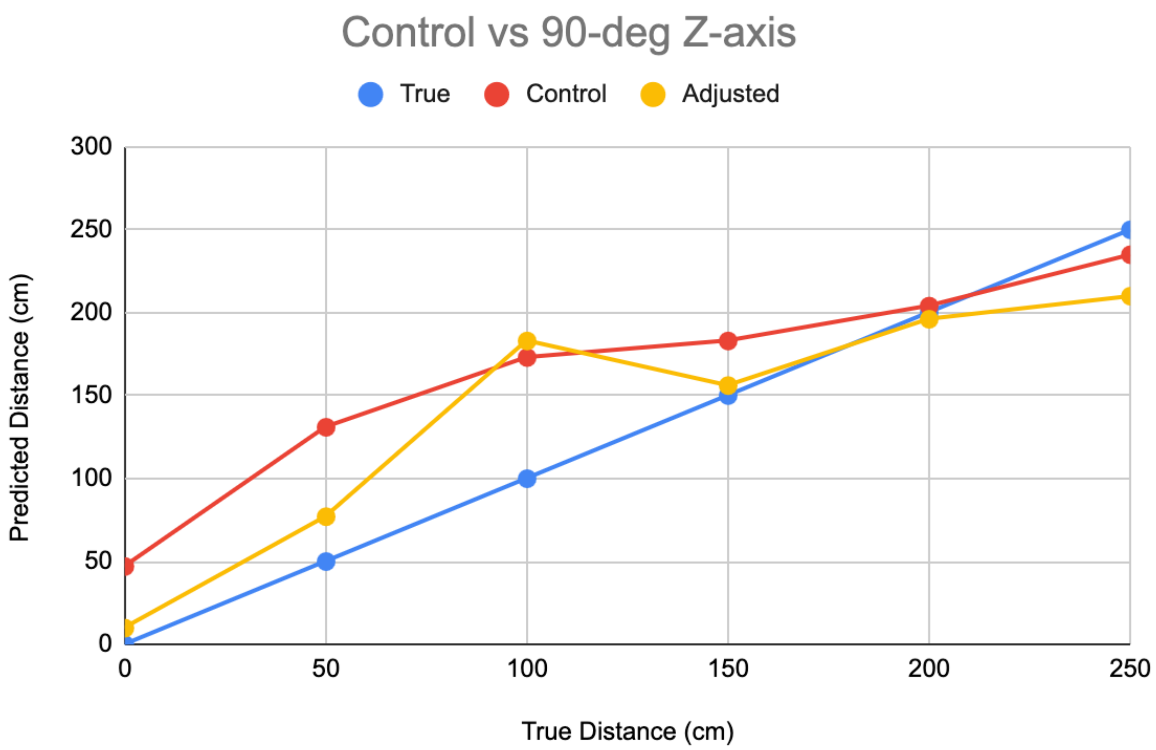
COMPARISON: CONTROL VS 90-DEG Y-AXIS					
TRUE DIST	**CONTROL ALGORITHM**		**ADJUSTED ALGORITHM**		
Dist (cm)	**Dist (cm)**	**Ctrl / True**	**Dist (cm)**	**Adj / True**	**|Ctrl| - |Adj|**
0	57		0		
50	256	412.0%	192	284.0%	128.00%
100	235	135.0%	96	-4.0%	131.00%
150	309	106.0%	216	44.0%	62.00%
200	393	96.5%	216	8.0%	88.50%
250	351	40.4%	253	1.2%	39.20%
				Average	89.74%



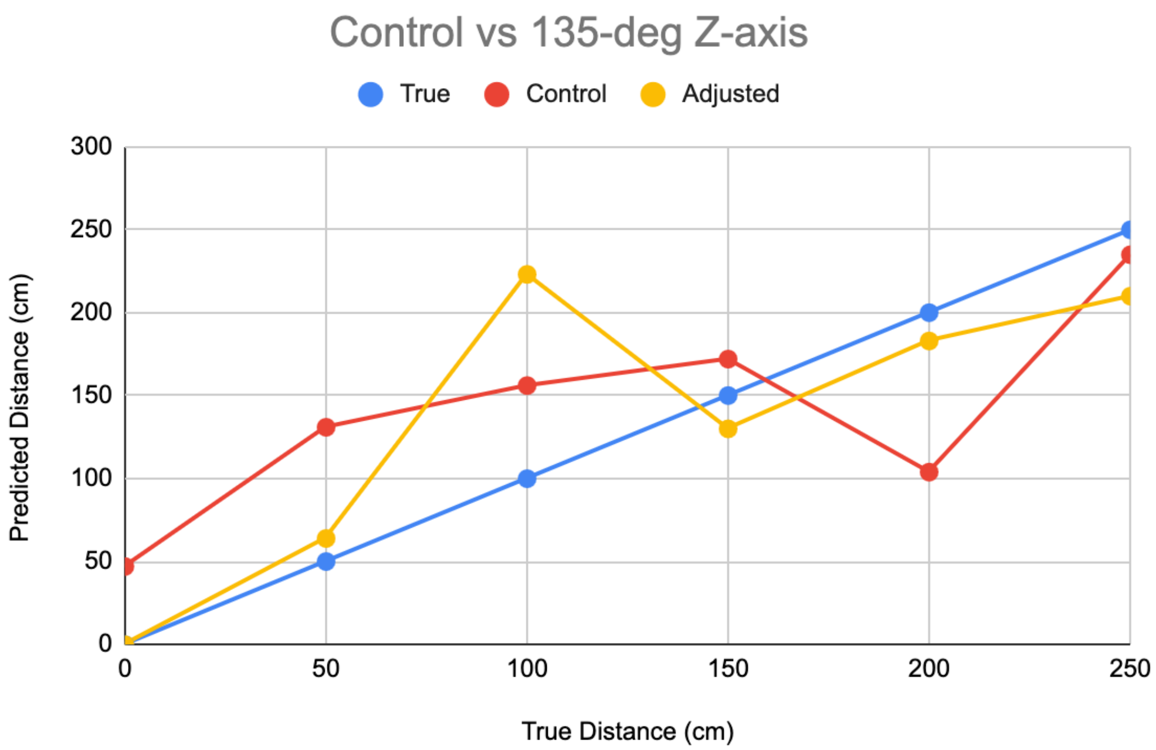
COMPARISON: CONTROL VS 45-DEG Z-AXIS					
TRUE DIST	**CONTROL ALGORITHM**		**ADJUSTED ALGORITHM**		
Dist (cm)	**Dist (cm)**	**Ctrl / True**	**Dist (cm)**	**Adj / True**	**|Ctrl| - |Adj|**
0	0		0		
50	99	98.0%	43	-14.0%	84.00%
100	267	167.0%	206	106.0%	61.00%
150	141	-6.0%	92	-38.7%	-32.67%
200	193	-3.5%	125	-37.5%	-34.00%
250	214	-14.4%	176	-29.6%	-15.20%
				Average	12.63%



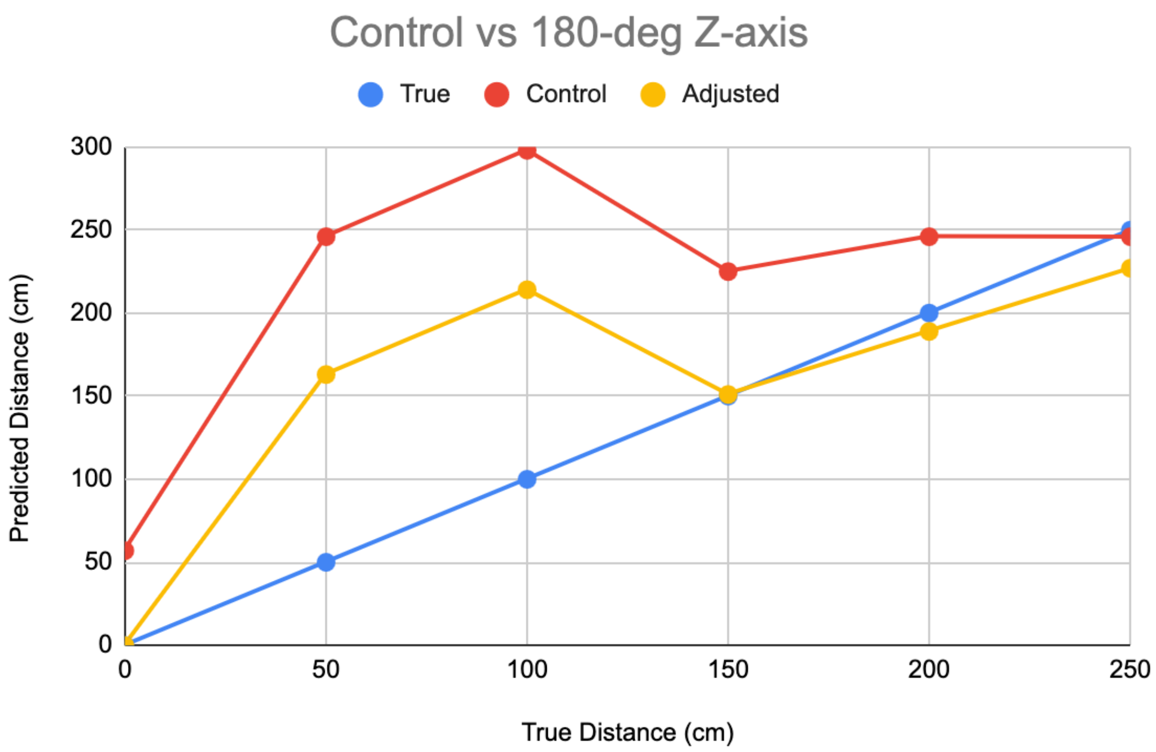
COMPARISON: CONTROL VS 90-DEG Z-AXIS					
TRUE DIST	CONTROL ALGORITHM		ADJUSTED ALGORITHM		
Dist (cm)	Dist (cm)	Ctrl / True	Dist (cm)	Adj / True	|Ctrl| - |Adj|
0	47		10		
50	131	162.0%	77	54.0%	108.00%
100	173	73.0%	183	83.0%	-10.00%
150	183	22.0%	156	4.0%	18.00%
200	204	2.0%	196	-2.0%	0.00%
250	235	-6.0%	210	-16.0%	-10.00%
				Average	21.20%



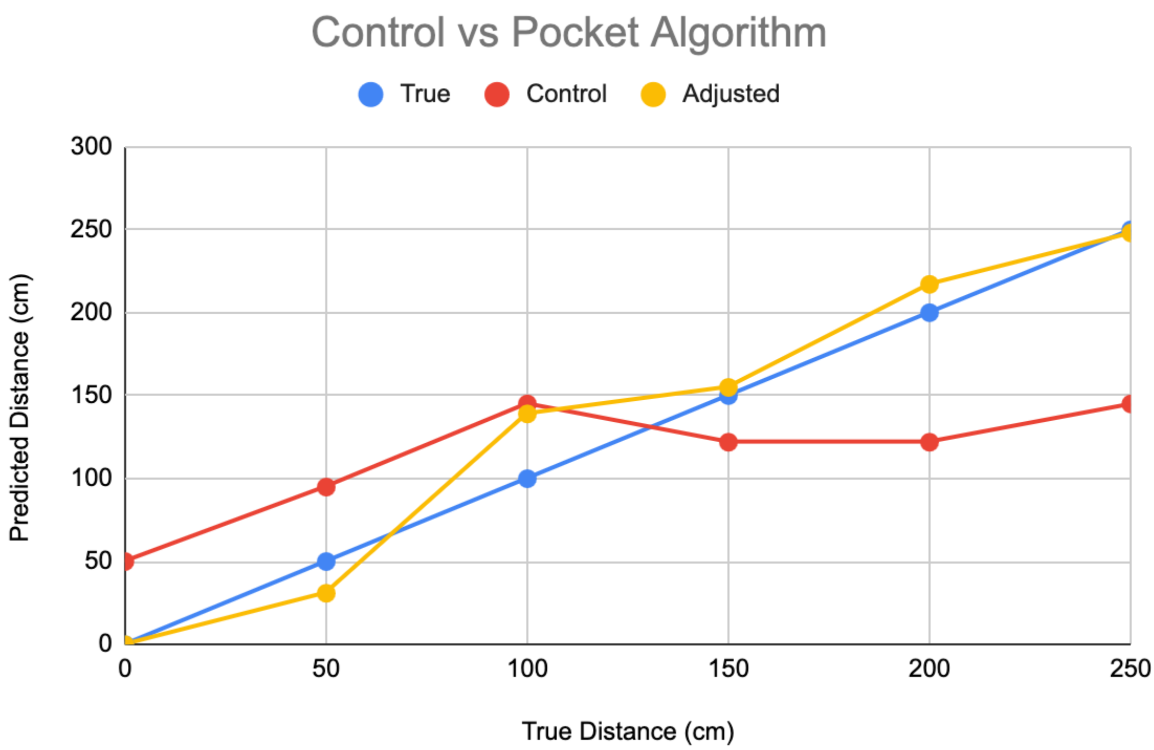
COMPARISON: CONTROL VS 135-DEG Z-AXIS					
TRUE DIST	CONTROL ALGORITHM		ADJUSTED ALGORITHM		
Dist (cm)	Dist (cm)	Ctrl / True	Dist (cm)	Adj / True	|Ctrl| - |Adj|
0	47		0		
50	131	162.0%	64	28.0%	134.00%
100	156	56.0%	223	123.0%	-67.00%
150	172	14.7%	130	-13.3%	1.33%
200	104	-48.0%	183	-8.5%	39.50%
250	235	-6.0%	210	-16.0%	-10.00%
				Average	19.57%



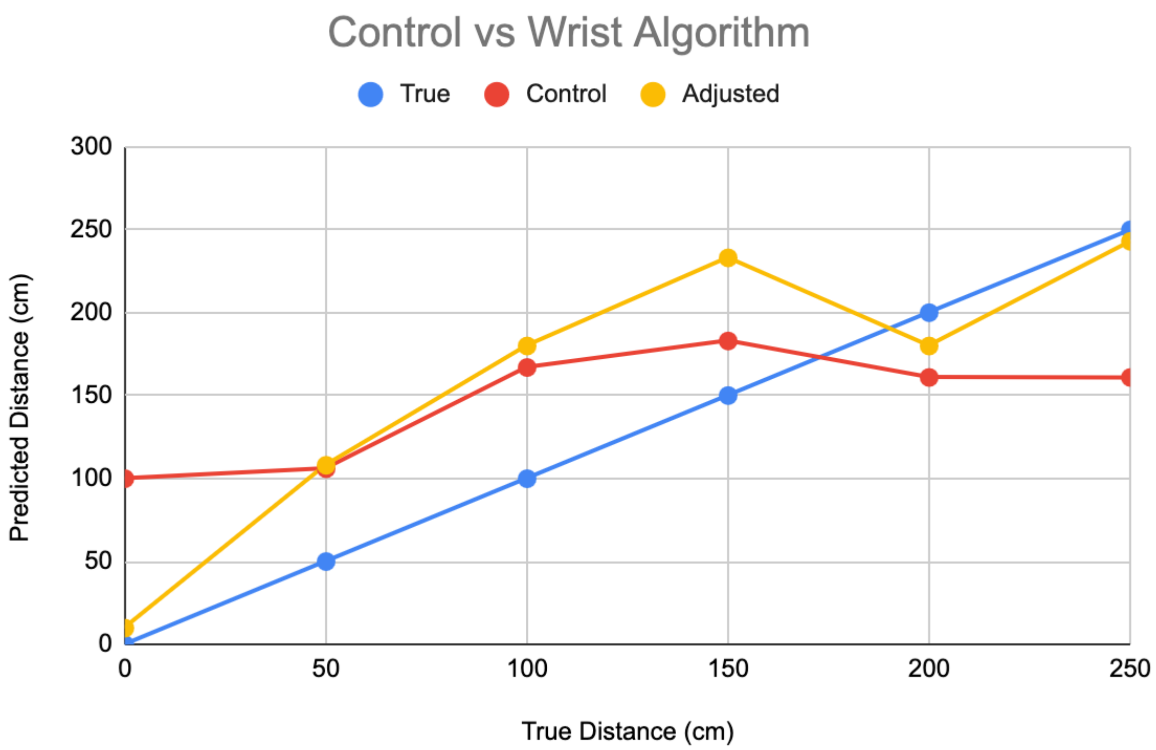
COMPARISON: CONTROL VS 180-DEG Z-AXIS					
TRUE DIST	CONTROL ALGORITHM		ADJUSTED ALGORITHM		
Dist (cm)	Dist (cm)	Ctrl / True	Dist (cm)	Adj / True	|Ctrl| - |Adj|
0	57		0		
50	246	392.0%	163	226.0%	166.00%
100	298	198.0%	214	114.0%	84.00%
150	225	50.0%	151	0.7%	49.33%
200	246	23.0%	189	-5.5%	17.50%
250	246	-1.6%	227	-9.2%	-7.60%
				Average	61.85%



COMPARISON: CONTROL VS POCKET ALGORITHM					
TRUE DIST	**CONTROL ALGORITHM**		**ADJUSTED ALGORITHM**		
Dist (cm)	**Dist (cm)**	**Ctrl / True**	**Dist (cm)**	**Adj / True**	**|Ctrl| - |Adj|**
0	50		0		
50	95	90.0%	31	-38.0%	52.00%
100	145	45.0%	139	39.0%	6.00%
150	122	-18.7%	155	3.3%	15.33%
200	122	-39.0%	217	8.5%	30.50%
250	145	-42.0%	248	-0.8%	41.20%
				Average	29.01%



COMPARISON: CONTROL VS WRIST ALGORITHM					
TRUE DIST	CONTROL ALGORITHM		ADJUSTED ALGORITHM		
Dist (cm)	Dist (cm)	Ctrl / True	Dist (cm)	Adj / True	|Ctrl| - |Adj|
0	100		10		
50	106	112.0%	108	116.0%	-4.00%
100	167	67.0%	180	80.0%	-13.00%
150	183	22.0%	233	55.3%	-33.33%
200	161	-19.5%	180	-10.0%	9.50%
250	161	-35.6%	243	-2.8%	32.80%
				Average	-1.61%



	RSSI			
Distance(cm)	**Result 1 (09:00)**	**Result 2 (13:00)**	**Result 3 (17:00)**	**Result 4 (21:00)**
0	34	37	39	36
50	38	38	39	38
100	43	45	44	43
150	49	51	50	50
200	51	51	51	52
250	55	58	54	53