

THE PENNSYLVANIA STATE UNIVERSITY  
SCHREYER HONORS COLLEGE

DEPARTMENT OF MECHANICAL ENGINEERING

Application of Video-Based Technology in Tracking Hand Movements for Medical Training  
Simulations

CYNTHIA BUDZINSKI  
SPRING 2024

A thesis  
submitted in partial fulfillment  
of the requirements  
for a baccalaureate degree  
in Mechanical Engineering  
with honors in Mechanical Engineering

Reviewed and approved\* by the following:

Jason Moore  
Professor of Mechanical Engineering  
Thesis Supervisor

Daniel Cortes  
Associate Professor of Mechanical Engineering  
Honors Adviser

\* Electronic approvals are on file

## **ABSTRACT**

This thesis explores the need for individualized feedback in medical simulation training for procedures such as endotracheal intubations. Current pass-fail assessments lack the specificity required for skill refinement, leading to potential complications in critical procedures. This thesis investigates the application of visual hand tracking technology, specifically Google's MediaPipe, to offer quantitative feedback. Using a wooden hand model on a linear motor with a depth camera, this experiment explores tracking hand movements in real time. Google's MediaPipe software tracks 3D hand motions, generating data points for analysis. Results demonstrated consistent patterns in hand movements corresponding to linear motor actions. Total distance traveled by each point of interest and error analysis provided insights into the accuracy of visually measured hand movements. Visual hand tracking, such as MediaPipe, may prove to be a promising tool for refining operator techniques in critical medical procedures. The experiments explore the technology's accuracy and underscores its potential application in medical training, addressing a gap in current practices and emphasizing the importance of tailored feedback for improved patient outcomes.

## TABLE OF CONTENTS

LIST OF FIGURES .....	iii
LIST OF TABLES.....	iv
ACKNOWLEDGEMENTS.....	v
Chapter 1 – Introduction and Background .....	1
Introduction and Background .....	1
Neonatal Endotracheal Intubation.....	4
Intubation Training and Skill Maintenance .....	5
Previous work with MediaPipe hand tracking.....	7
Conclusion.....	8
Chapter 2 - Methods and Materials.....	9
Materials.....	9
Experimental Design and Setup.....	9
Chapter 3 – Results .....	13
Chapter 4 – Discussion and Future Work .....	20
Challenges and Limitations .....	21
Applications to Existing Medical Simulations .....	22
Recommendations for Future Work.....	23
Appendix A: Python Code.....	25
Python Code for Retrieving Position Data using MediaPipe .....	25
Appendix B: MATLAB Code .....	28
MATLAB Code for XY Trials .....	28
MATLAB Code for YZ Trials.....	33
Appendix C: Additional Subplots for Trials .....	37
BIBLIOGRAPHY.....	41

## LIST OF FIGURES

Figure 1. Correct Placement of an Endotracheal Tube.....	2
Figure 2. Any TIAE, severe TIAE and severe oxygen desaturations by number of TI attempts	5
Figure 3. Data point labels for MediaPipe hand reconstruction.....	7
Figure 4. Manikin Hand Model on Linear Motor.....	10
Figure 5. Workflow of Experiment .....	11
Figure 6. Manikin hand on linear motor for YZ trials.....	12
Figure 7. X-Position vs. Time for Points of Interest 4, 8, 12, 26, and 20 for Trial 1 .....	13
Figure 8. Error in mm at each pause for each POI.....	16
Figure 9. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 1 .....	17
Figure 10. Error in mm at each pause for each POI .....	19
Figure 11. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 2.....	37
Figure 12. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 3.....	37
Figure 13. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 4.....	38
Figure 14. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 5.....	38
Figure 15. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 2 .....	39
Figure 16. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 3 .....	39
Figure 17. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 4 .....	40
Figure 18. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 5 .....	40

## **LIST OF TABLES**

Table 1. Total Distance Traveled by Points 4, 8, 12, 16 and 20 during XY Trial 1.....	14
Table 2. Total Distance Traveled by Points 8 and 12 during YZ Trial 1.....	18

## ACKNOWLEDGEMENTS

I would first like to express my gratitude to my family for their unwavering support throughout this journey. To my mom, Sharon, dad, Matthew, and brother, Steven, your encouragement and belief in me have been invaluable. To Michael Parrish, I am forever grateful for your support through this entire process.

I am extremely grateful to Dr. Jason Moore for his guidance and mentorship throughout this project. Your insights and encouragement have been pivotal in shaping my research. Additionally, I would like to extend my appreciation to Hang-Ling Wu for his endless advice and willingness to help with any bugs I may have encountered.

I am thankful to the Schreyer Honors College, the College of Engineering, and Penn State for their support and sponsorship, without which, this project would not have been possible.

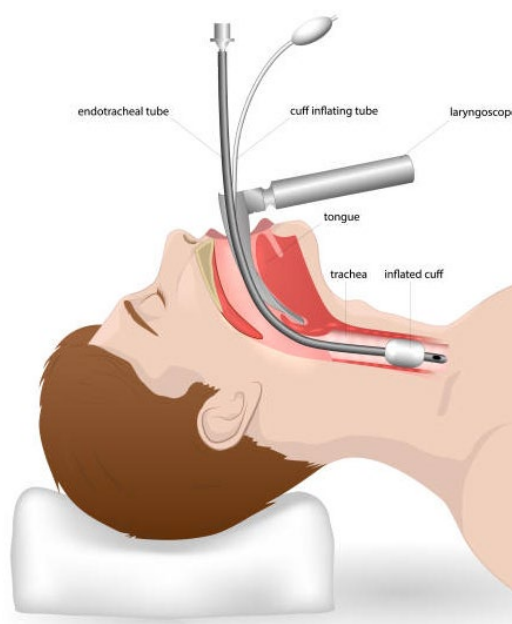
Finally, to my friends, thank you for your support and for being there through these past four years.

Research reported in this publication was supported by the National Institute of Diabetes and Digestive and Kidney Diseases of the National Institutes of Health under Award Number R01DK137230. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

## **Chapter 1 – Introduction and Background**

### **Introduction and Background**

An endotracheal intubation is a common, and sometimes lifesaving, procedure that opens a patient's airway to help prevent suffocation. Endotracheal intubations are routine for many hospitals and surgical procedures; however, they may also be performed in emergency settings. While this procedure may be common, positioning a laryngoscope (the device used in an endotracheal intubation) is difficult. Figure 1 shows the pathway for a laryngoscope and endotracheal tube that are used during the procedure. The operator must align the laryngeal, pharyngeal, and oral axes using a laryngoscope. Human variability can be an additional factor affecting health care providers' ability to successfully place an endotracheal tube. Risks associated with failed intubations include, but are not limited to, damage to the teeth or throat, bleeding, injury to the lungs, and hypoxia which can lead to brain injury or death [1].



**Figure 1. Correct Placement of an Endotracheal Tube**

Complications and misplaced intubation tubes are more common when performed in emergency situations. A study conducted by the emergency department of the University of Pittsburgh's hospital characterized an endotracheal intubation error to be one of the following: a misplacement of the breathing tube, four or more placement attempts, or a failure to place a tube at all. Of the 1,953 patients that received an endotracheal intubation, 22% of them had one or more of these errors [2], [3]. A later study by the same group found that of the failure cases with negative outcomes, 73% of the patients died [3]. Additionally, Katz and Falk conducted a study in which 108 patients arrived in the emergency department after a paramedic performed an intubation. Of these 108 patients, 25% of them had been given an improperly placed endotracheal tube [4]. It can be concluded from these studies that errors occur more often in emergency situations or with operators that perform less endotracheal intubations per year.



It has also been shown that an improvement in technology does not directly correlate to an improvement in skill and patient outcomes. A video-laryngoscopy is an indirect form of laryngoscopy that uses a fiber optic camera to gain visualization of the airway [5]. While the technology is steadily improving, direct laryngoscopy is still preferred over indirect. A study published in *The Laryngoscope* shows that the increased use of video-laryngoscopes has correlated to an increase in oropharyngeal injuries [6]. When comparing the rate of oropharyngeal injuries from direct laryngoscopy and video-laryngoscopy, the difference between the two of them was found to be statistically significant ( $p < 0.005$ ). Therefore, it can be concluded that the improved technology being used in hospitals does not always translate into better outcomes for patients.

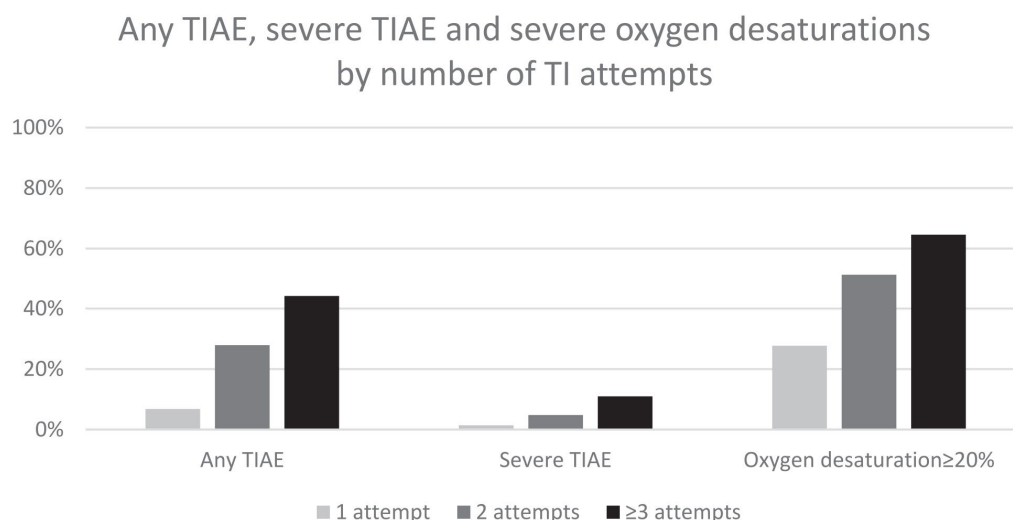
Currently, no universal standard exists for endotracheal intubation training. The proficiency of training and maintenance of skill must be evaluated by individual healthcare practices [1]. Many hospitals have simulation training facilities with mannequins for residents to learn procedures and other professionals to practice their skills. Before graduation, residents are expected to complete several sessions of training within their simulation center [7]. While these training opportunities can help operators learn and practice the techniques of endotracheal intubations, the operators are graded on a binary pass-fail basis. For this reason, little opportunity exists for individual feedback or quantitative data for how to improve the operator's skills.

It is important to include training with quantitative feedback for operators. For instance, having the ability to analyze hand movements in real time would provide valuable information on the flaws in an operator's technique. This feedback could give healthcare providers a better skillset to improve their patients' outcomes.

## Neonatal Endotracheal Intubation

One area of medicine in which endotracheal intubations are particularly difficult is neonatal medicine. Specifically with endotracheal intubations, infants are particularly difficult because of their unique and less developed airway anatomy [8]. In addition, endotracheal intubations require the patient's neck to be positioned in a certain way. Neonatal patients have abnormal neck mobility and may have head abnormalities [9]. Thus, conducting an endotracheal intubation on a neonatal patient requires more training and specialized skill maintenance compared to more common airway anatomies.

A study published in the Journal of Perinatology analyzed the relationship between the number of intubation attempts and number of negative outcomes from intubations conducted in the delivery room and neonatal intensive care unit (NICU). Data were collected from 17 different training hospitals over a three-year period. 22% of the 7,708 patients requiring intubation required three or more attempts to properly intubate [8]. Even in cases with adult patients, an increased number of intubation attempts can have detrimental outcomes. These may include severe oxygen loss, trauma or injury to the teeth or lips, or even life-threatening soft tissue damage [1]. For neonatal patients, additional intubation attempts can have long lasting negative effects on the patient's immature, developing brain. The study concluded that increasing number of attempts to intubate directly correlated to increased risk of adverse tracheal intubation associated events (TIAEs) and severe oxygen desaturation [8]. The results of this study and the trends drawn from it are included in Figure 2.



**Figure 2. Any TIAE, severe TIAE and severe oxygen desaturations by number of TI attempts**

As shown in Figure 2, as the number of intubation attempts increases, so does the potential of a (TIAE) and/or severe oxygen desaturation. Therefore, it is of particular importance to improve the training of health care providers that will be performing neonatal endotracheal intubations. Furthermore, neonatal training simulations are more difficult to have access to, so it is important to make the simulation training as efficient and helpful to the trainees as possible.

### **Intubation Training and Skill Maintenance**

Health care providers trained to perform endotracheal intubations include anesthesiologists, paramedics, respiratory therapists, medical students, and others. However, all these professionals have different training standards and perform this procedure at different frequencies. While an anesthesiologist may perform several intubations a week, a medical student may only conduct a handful of intubations before graduation and therefore, not be

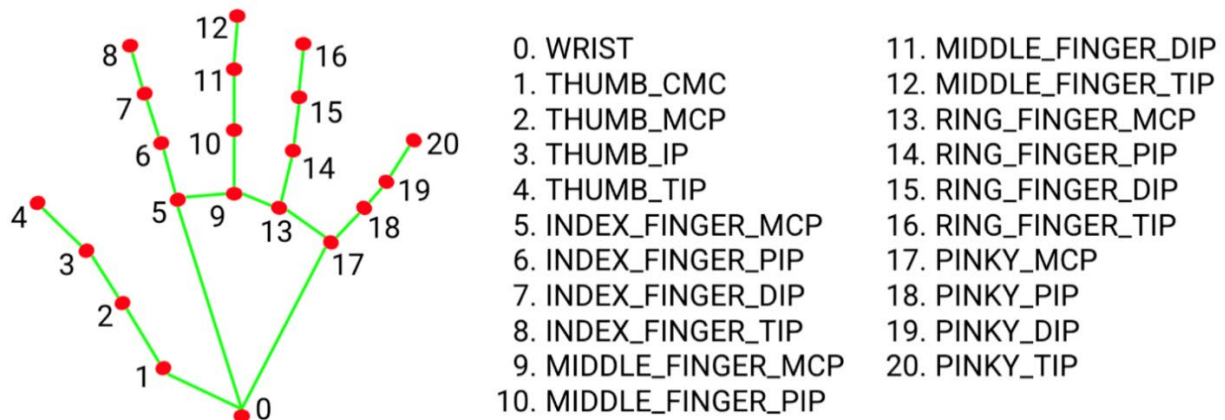
qualified to conduct this procedure [7], [10], [11]. Proficiency with this procedure is not acquired until the health care professional has had sufficient experience and exposure [12].

When surveying respiratory therapists about their training and skill maintenance, it was found that there was “significant variance in the training methods, skill maintenance, and recertification requirements” among all the study participants [13]. Furthermore, 24% of the respiratory therapists that responded disagreed with the statement “I am satisfied with my institution’s current program for providing endotracheal intubation training.” This dissatisfaction is not isolated to respiratory therapists. A study conducted on medical student interns analyzed the competence tests that medical students must pass to provide patient care. On several of the procedures, 37% of residents (even upon graduation) could not achieve the minimum passing score without making more than one attempt [11]. Many tests that students undergo are just pass/fail and do not offer quality feedback on how a student could improve their score. There is not a sufficient framework for medical students to demonstrate competency with certain procedures before residency.

On the other hand, studies have been conducted analyzing the effect of simulation training on medical students’ skills and confidence. Simulation training, especially with high fidelity simulations, has been found to improve a student’s skillset and confidence level with the procedure [14]. Medical students were surveyed before and after taking a simulation class about managing septic shock. Not only were their mean post-test examinations statistically higher than the pre-test examinations ( $p < 0.001$ ), but their mean confidence level in managing septic shock increased compared to before the class ( $p < 0.001$ ) [14]. Improving skillsets and confidence of medical professionals is vital to decreasing error rates with endotracheal intubation procedures.

## Previous work with MediaPipe hand tracking

To track the hands in a 3D space, software from Google’s MediaPipe was used. This technology is open-source and has been used in several projects to track hands and place coordinates on hands in a camera frame [15], [16]. Data can be retrieved about each point that is placed on the hand. These data can then be reconstructed into a computational model of the hand. An example of the data points and hand reconstruction is in Figure 3. Each point on each finger is given a number and label. Individual motion data is recorded for each point when using MediaPipe. Thus, 3D position data can be extracted and analyzed about one point on the hand, an entire finger, or the entire hand.



**Figure 3. Data point labels for MediaPipe hand reconstruction**

One study using MediaPipe specifically looked at the movements of the hands “through simple tasks commonly used in clinical practice” [16]. This study validated the initial use of MediaPipe in a clinical setting and further explained the uses of it for research. Moreover, a very recent study used MediaPipe’s hand tracking technology to analyze Parkinson patients before and after medication. In addition to using the video hand tracking features of MediaPipe, the

experiment used accelerometers on the study participants' hands. The accelerometers were of a known accuracy, so the data recorded from the accelerometers could be compared to the data recorded by MediaPipe's hand tracking capabilities. The data from the video hand tracking "correlated well" with the measurements from the accelerometers [17]. Although still a new and developing technology, using MediaPipe to track the movements of hands during a procedure or training has been proven to be valid and accurate.

## **Conclusion**

The error rates for endotracheal intubations point to a flaw in the way that intubations are taught and tested. Training and testing on a pass-fail basis are no longer the most effective way for an anesthesiologist to practice their skill. There is a gap between the knowledge and the technology being used to reduce endotracheal intubation related injuries. However, as this thesis suggests, improving training for operators may reduce the frequency of oropharyngeal injuries. Additionally, the uniqueness of intubation of neonatal patients' anatomy adds an additional complication for healthcare providers. Therefore, it is of importance to develop high level training simulations to give quantitative and individualized feedback for healthcare providers about their intubation techniques. Having the ability to assess hand movements in real time would provide valuable information on the flaws in an operator's technique. This feedback would give healthcare professionals the skillset to deliver better outcomes to their patients. Finally, MediaPipe has not yet been sufficiently explored for applications to the medical field. This thesis presents an opportunity to examine the accuracy of MediaPipe and potential future applications for training of other medical procedures.

## **Chapter 2 - Methods and Materials**

### **Materials**

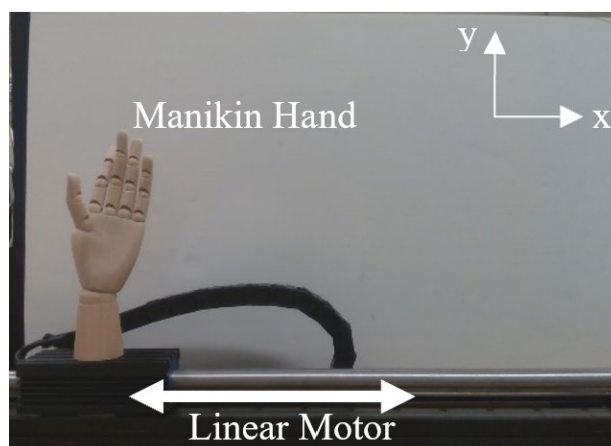
The experiment was conducted, and all materials were sourced from, the Precision Medical Device Laboratory at the Pennsylvania State University. The model for this experiment is a wooden hand that has moving joints. This type of hand model is typically used by artists because of its ability to be repositioned – and stay in place – into many different hand positions. This model was chosen not only because of how closely it resembled a hand, but also to allow us to reposition the hand if necessary. The hand model was placed on top of a linear motor from Dunkermotoren linear systems. An Intel® RealSense™ Depth Camera D435 was used to capture the movement during the experiment. A camera tripod was used to position the depth camera at an appropriate height and distance from the lab bench in order to be level with the hand. Several computer programs including MATLAB, Python, and LabView were used in this experiment.

### **Experimental Design and Setup**

To track the hands in a 3D space, software from Google's MediaPipe was used. This technology is open-source and is used to track the hand and place coordinates on it in a camera frame [5], [6]. MediaPipe was accessed by using Python. As stated earlier, by using MediaPipe, individual motion data can be recorded for each of the 21 points on the hand. Using Python, MediaPipe was installed and activated. Within the Python script, several points of interest were identified. In order to reduce the processing time, 5 points of interest were chosen from the 21

options seen in Figure 3. The points of interest that were chosen were points 4, 8, 12, 16, and 20. These points were chosen because they are the tips of the fingers on the hand. It was assumed that these points would give us the best idea of the accuracy of the data because they are at the tip of the digits. The Python script is located in Appendix A.

The depth camera was positioned on the tripod at an appropriate height and distance away from the test bench. The linear motor program, LabView, was started and the linear motor turned on to ready. LabView is an application that helps build test systems [18]. It also controls the linear motor that is being used for the experiment. The test system that was used is basic, but effective. Once the program begins, the user can input the distance in millimeters from the homing point that he or she wants the linear motor to travel. The homing point, in relation to the depth camera's point of view, is the right-most side of the linear motor. Once told which position to travel to, the linear motor moves at a constant speed to the desired location. The hand, placed in an open-palm position, was placed on top of the linear motor as seen in Figure 4.

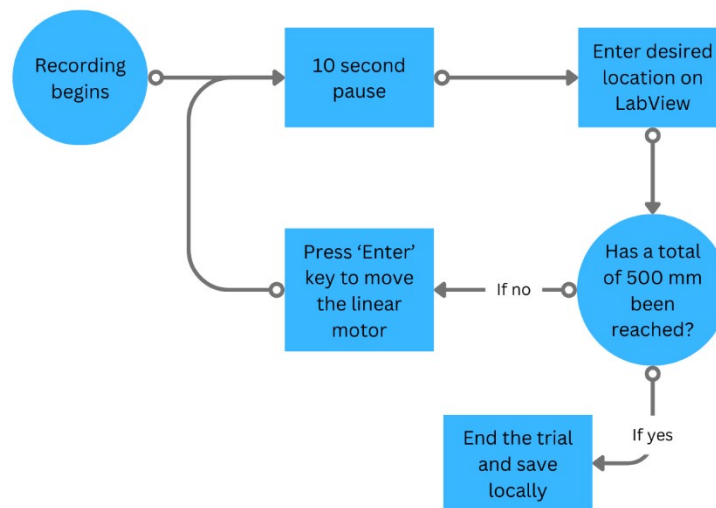


**Figure 4. Manikin Hand Model on Linear Motor**



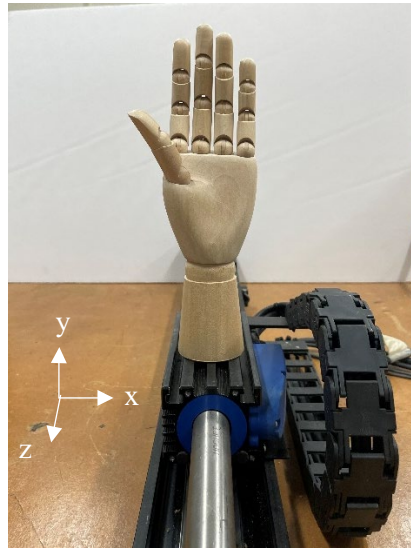
Once the hand was positioned and the linear motor program ready, the following sequence was performed. The recording began, followed by a 10-second pause, and then the linear motor and hand system was moved 100 millimeters in the negative x direction of the depth camera. That is, the linear motor's starting position was at the home point, on the right side of the depth camera's frame. This process was repeated until a total distance of 500 millimeters was covered. The recording was stopped, and the acquired data was locally saved in an Excel file.

Figure 5 graphically represents how the sequence of events was performed.



**Figure 5. Workflow of Experiment**

This experimental process was repeated five times in the XY plane (relative to the depth camera frame) and five times in the YZ plane (relative to the depth camera frame). Figure 6 below shows the orientation of the depth camera, manikin hand, and linear motor for the trials in the YZ plane.



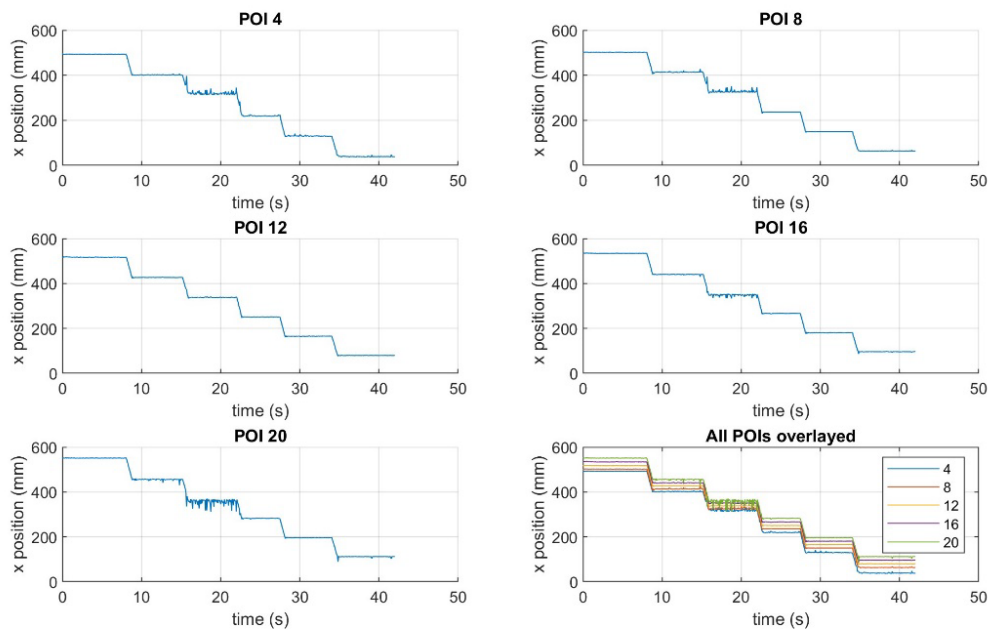
**Figure 6. Manikin hand on linear motor for YZ trials**

After all trials were completed, the recorded data underwent processing in MATLAB. The MATLAB code can be found in Appendix B. First, any zero values that may have been present due to noise in the data acquisition were filtered out. These “zero” values indicate times in the experiment where the depth camera may have picked up on something else or misidentified it as the point of interest we were targeting. Then, an animated line representation was generated in a 3D space to visualize the movement of the knuckles of interest throughout the experiment. This line represents what all the data points during the experiment look like in a 3D space.

Additionally, a plot was created illustrating the x-position of each point of interest over the duration of the experiment. The total distance traveled by the points of interest was calculated and recorded. The results of all ten trials can be found in the results section.

### Chapter 3 – Results

The first five trials recorded the manikin hand moving in the XY plane relative to the depth camera. Results of the first trial as a subplot consisting of six position vs. time graphs are shown in Figure 7. There is one subplot for each point of interest, and the sixth plot overlays all five points of interest over each other. As stated before, points 4, 8, 12, 16, and 20 were chosen as points of interest because they are the tips of each digit on the hand manikin model.



**Figure 7. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 1**

Additional subplots with position vs. time data for the subsequent 4 trials can be found in Appendix C. In each of the subplot graphs, the five movements of the linear motor and corresponding pause can be distinctly seen. This pattern is consistent with all five points of interest and across all trials in the XY plane. Data underwent processing using MATLAB and Excel. First, the total distance traveled for each point of interest was recorded and a bias factor was calculated. The total distance that each point

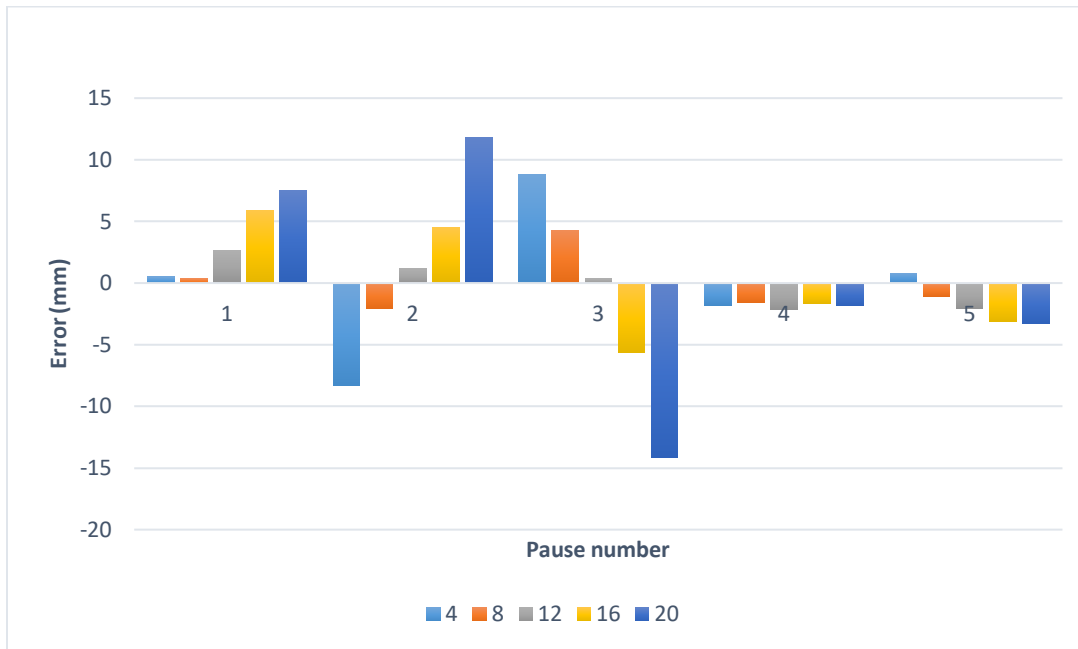
traveled in Trial 1 is as follows: 509 mm, 489 mm, 491 mm, 499 mm, and 512 mm for points 4, 8, 12, 16, and 20, respectively when bias factor, eliminating the steady state error, was added. This steady state error factor was necessary as a calibration factor for the camera system. Additionally, the average distance at each pause was recorded and used to calculate the distance between pauses. Table 1 below shows the total distance traveled by each POI during the duration of the experiment after the elimination of steady state error. While total distance traveled may vary by point for each trial, across all trials, the distance for an individual point remains relatively precise. For example, the data from trial one, had a range from 489 to 512 mm, or a range of about 23 mm. However, focusing on just POI 4, across all five trials, the range is from 508 to 514, or a range of 6 mm. This pattern is reflected across all five POIs. From this data, it can be concluded that across several trials, the distance a point travels remains precise. Therefore, for future trials, the bias factor should be calculated for each point individually rather than for each trial.

**Table 1. Total Distance Traveled by Points 4, 8, 12, 16 and 20 during XY Trial 1**

<i>Point of Interest</i>	<i>Trial Number</i>	<i>Total Distance Traveled (mm)</i>
<b>4</b>	1	509
	2	511
	3	508
	4	513
	5	514
<b>8</b>	1	489
	2	485
	3	488
	4	486

	5	485
<b>12</b>	1	491
	2	488
	3	490
	4	484
	5	486
<b>16</b>	1	499
	2	495
	3	513
	4	488
	5	488
<b>20</b>	1	512
	2	520
	3	563
	4	528
	5	514

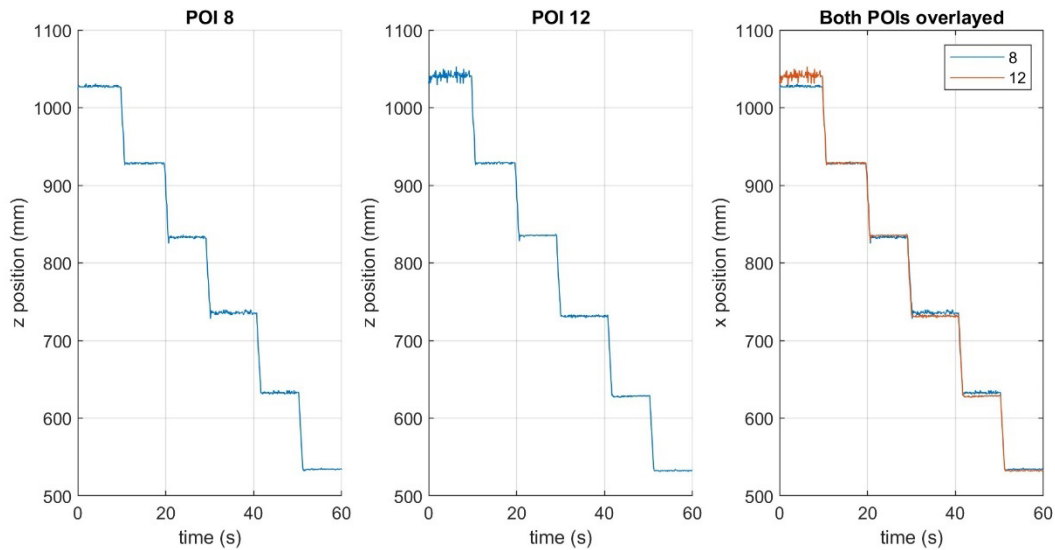
Figure 8 shows the error with eliminated steady state error in the distance between each pause for each of the five points of interest for Trial 1. The error ranged from an absolute value of 0.358 mm to 14.1 mm with an average value of 3.89 mm. Over a 500 mm span, this inaccuracy ranges from 0.0716% to 2.83% with an average value of 0.778%. This is an acceptable range of inaccuracy for developing a nominal model for hand tracking.



**Figure 8. Error in mm at each pause for each POI**

For the YZ plane trials, only two points of interest, 8 and 12, were analyzed. When capturing initial data, the depth camera was struggling to capture clean data on more than two points at a time. This could be because of the machine learning algorithms being slightly different for capturing data in this direction compared to the XY plane. Therefore, it may be more challenging to collect distance data compared to two-dimensional movement. For the purposes of this thesis – that is, to analyze the accuracy of MediaPipe’s output – data was examined from only the two points of interest that were consistently being identified. Results of the first trial as a subplot consisting of three z-position vs. time graphs are shown in Figure 9. There is one subplot for each point of interest, and the third plot overlays both points of interest over each other. As with the XY trials, the five movements of the linear motor and corresponding pause can be distinctly seen. This pattern is consistent with both points of interest and across all

five trials in the YZ plane. Additional subplots with Z-position vs. time data for the subsequent 4 trials can be found in Appendix C.



**Figure 9. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 1**

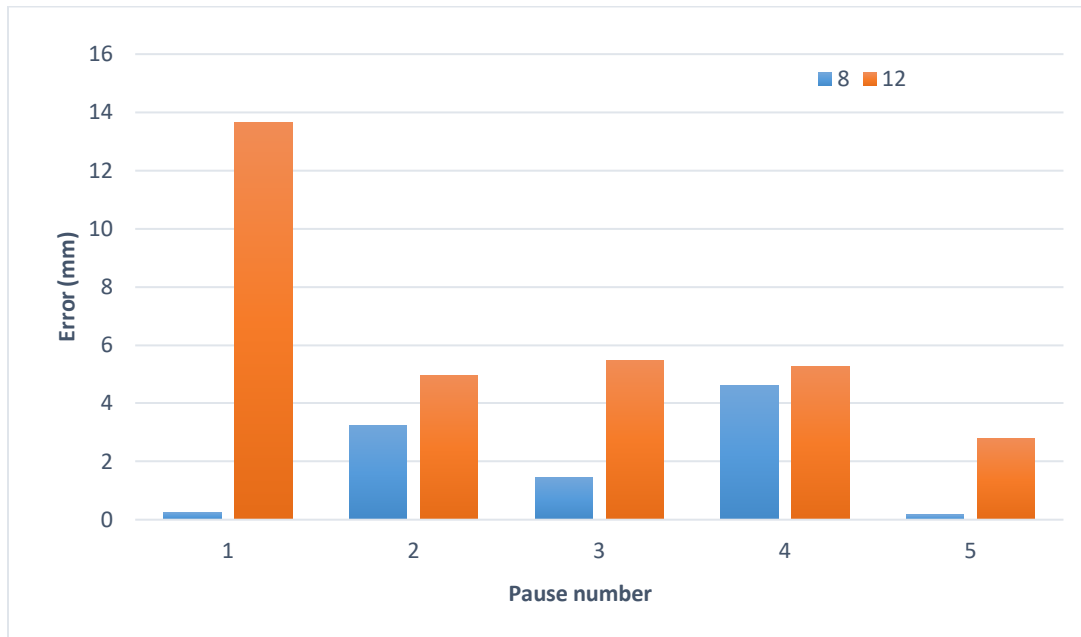
Data for this trial underwent the same processing as for the XY trials. The total distance that each point traveled in Trial 1 for the YX plane was 499 mm and 522 mm for points 8 and 12, respectively. Notably, for all five trials in the YZ plane, the distance traveled by point 8 was extremely accurate, whereas the distance traveled by point 12 was consistently off by about 25 mm. Similar to the XY results, this could mean that the bias factor should be applied to each point individually as opposed to across an entire trial. The total distance traveled by each point in the YZ trials can be seen in Table 2.

**Table 2. Total Distance Traveled by Points 8 and 12 during YZ Trial 1**

<i>Point of Interest</i>	<i>Trial Number</i>	<i>Total Distance Traveled (mm)</i>
<b>8</b>	1	499
	2	499
	3	500
	4	501
	5	500
<b>12</b>	1	522
	2	528
	3	529
	4	527
	5	523

Additionally, the average distance at each pause was recorded and used to calculate the distance between pauses. Figure 10 shows the error with eliminated steady state error in the distance between each pause for each of the five points of interest. For all of the trials combined, the error ranged from an absolute value of 0.162 mm to 13.67 mm with an average value of 4.19 mm. Over a 500 mm span, this inaccuracy ranges from 0.0324% to 2.734% with an average value of 0.838%.





**Figure 10. Error in mm at each pause for each POI**

## Chapter 4 – Discussion and Future Work

For the XY plane trials, the highest error out of all the points of interest was POI 20. Some possible reasons for that may be that POI 20 resides on the smallest digit of the hand, and it resides closer to other points of interest. Additionally, there may be limitations to the accuracy of some points of interest depending on the training data that was used to develop the MediaPipe model. However, this high-level accuracy supports the idea that highly accurate automated assessments of medical training could be conducted.

For the YZ plane trials, POI 8 had a consistently high accuracy and did not need to accommodate for steady state error. This interesting finding helps to support the idea of using MediaPipe for highly accurate medical simulation training procedures. Additionally, while the distance readings for POI 12 were not as accurate, they were consistent with a total range of only 7 mm from each other. Therefore, with the correct bias factor calibration, we could see an accurate and consistent distance reading for both POIs 8 and 12 over the entire range of motion.

From this data, it can be concluded that MediaPipe can be used to give highly accurate results regarding changes in position over time. Thus, it can be used to give quantitative feedback to medical professionals about their approach to endotracheal intubation or colonoscopy. The use of MediaPipe in a simulation training environment would address existing gaps in current practices and has the potential to enhance patient outcomes. Additionally, because the materials needed to use MediaPipe in training are not hardware intensive, it can be integrated into already existing training programs. This can set a precedence for high level assessments of performance in medical training without the need for extra hardware intensive equipment.

## Challenges and Limitations

An initial challenge that I had with this project was learning how to properly use MediaPipe for my intents. This was partly because I am unfamiliar with Python as a programming language and there was a significant learning curve to understand its differences from MATLAB, a language with which I am thoroughly comfortable. This challenge was also due in part to MediaPipe having a range of functions to choose from. After a while, I learned how to use Python and the MediaPipe features to my advantage, and I wrote a script that was able to capture and record the data I needed to capture using the depth camera.

A second challenge I encountered when trying to collect data, was the depth camera sometimes “missing” the points on the hand. I quickly realized this was due to the number of things in the background of the camera frame that were in the lab. Having too many things in the background was hindering the camera’s, and MediaPipe’s, ability to locate the points on the hand. To resolve this, I placed a white poster board behind the linear slide. This blocked out the other things in the background of the camera frame, and in addition, it added an aspect of contrast to the frame. Both helped the camera and MediaPipe to pick up on the points on the hand better. After this remedy, the data was much easier to collect and clearer after processing.

One limitation to this experiment is the processing speed and bandwidth of the device being used to store the data. If the speed is too slow, the sampling frequency may be inadequate to capture the correct data. This problem becomes apparent whenever trying to capture data about all 21 points on the hand at once. This is why the decision to only capture data about five points was made. This ensured we captured data about all five digits on the hand, and it also allowed us to use an adequate processing speed and sampling frequency. However, once data is

captured, there is no way to go back and add points that you may want to extract data about. In other words, I programmed Python to use MediaPipe to collect position data on points 4, 8, 12, 16, and 20. After running the trials, it is impossible for me to open a data file and analyze position data for point 13 for that specific trial. In order to get data for point 13, the Python code would need to be edited, and a new trial would need to be run. This is a limitation because if we are trying to visualize the hand pathway during a procedure in future work, it would be impossible to find certain data unless it was captured in the initial collection.

Finally, some limitations of MediaPipe that we found include the need for calibration and a bias factor. This is easy enough to complete during the data analysis process of the experiment, but it is an additional limitation of the technology. Additionally, the depth aspect of MediaPipe's hand tracking capabilities does not seem to be as fleshed out as the other capabilities. This may be a limitation of the software, and hopefully future versions of it will have this more fleshed out. However, the depth data acquisition may be improved depending on the kind of camera that is in use.

### **Applications to Existing Medical Simulations**

Adding MediaPipe to existing medical simulations could be done quite seamlessly. A depth camera on a tripod can be set up off to the side of the simulation in the clinical training facility. There is very little hardware setup that would need to be added to the existing simulation space. Therefore, adding MediaPipe to a training simulation space could be done across many different training hospitals and with a variety of simulated procedures. The camera can capture the movement of the points of interest on the hand and can model the pathway of them.

Additionally, after a practice attempt is complete, an animated line can be shown to the trainee to show the movement over time of their hand during the procedure.

### **Recommendations for Future Work**

As the first student on this project, there is still much more to be discovered about MediaPipe and its applications to medical simulation training. Firstly, more work needs to be done to explore the high error and noise that is seen in the YZ plane data. The distance (Z-direction) readings that were captured from the depth camera were not as accurate as the X and Y data, and they were noisier than the X and Y data as well. The Z (“distance”) data is a different variable output for the depth camera, so more work into how that affects the results should be explored. Additionally, this project examined the accuracy of MediaPipe over a larger range of motion. Most endotracheal intubation procedures will not likely require a range of motion of 500 mm. After reporting the accuracy of MediaPipe over this larger range of motion, accuracy of MediaPipe over a small range of motion should be explored. This can give insight into whether MediaPipe may or may not be equipped to capture motion data over a range of motion that more closely resembles that of an endotracheal intubation procedure.

Then, the testing should move from using an open palm manikin hand to a hand that is gripping a laryngoscope or similar instrument. This testing will be able to provide insight into how the accuracy of MediaPipe may change if part of the hand is obstructed from an instrument, much like it may be during an endotracheal intubation procedure simulation.

Finally, research should be conducted filming a clinician’s hands moving with a device, such as a laryngoscope, in hand. This could further track how MediaPipe collects position data

with an obstructed view. In addition, this future work could help plot the proper pathway of the hand through the procedure. This data can then be used as a potential baseline for comparing training attempts to. Overall, this project is still in the early stages of its development, so there are plenty of avenues for future work to follow.

## Appendix A: Python Code

### Python Code for Retrieving Position Data using MediaPipe

```
import cv2
import mediapipe as mp
import pyrealsense2
from realsense_depth import *
import pandas as pd
import time

# Initialize MediaPipe hand components
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands

# Initialize Camera Intel Realsense
dc = DepthCamera()

# Initialize MediaPipe Hands
with mp_hands.Hands(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:
    data = []
    frame_number = 0

    while True:
        ret, depth_frame, color_frame = dc.get_frame()

        if not ret:
            print("Failed to grab frame.")
            break

        # Convert the BGR image to RGB before processing
        color_frame_rgb = cv2.cvtColor(color_frame, cv2.COLOR_BGR2RGB)

        # Process the frame and get hand landmarks
        results = hands.process(color_frame_rgb)

        # Draw the hand landmarks on the color frame
        if results.multi_hand_landmarks:
            for idx, hand_landmarks in enumerate(results.multi_hand_landmarks):
```

```

mp_drawing.draw_landmarks(
    color_frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)

# Get hand type (Left/Right)
hand_type = results.multi_handedness[idx].classification[0].label
print(f"Detected {hand_type} hand.")

# Get distance for knuckles 4, 8, 12, 16, 20
knuckles = [4, 8, 12, 16, 20] # you can change which knuckle you
want here

for knuckle in knuckles:
    x, y = int(hand_landmarks.landmark[knuckle].x *
color_frame.shape[1]), int(
        hand_landmarks.landmark[knuckle].y *
color_frame.shape[0])
    cv2.circle(color_frame, (x, y), 10, (0, 255, 0), -1)

    # Get depth at the knuckle point
    distance = depth_frame[y, x]
    now = time.localtime()
    current_time = time.strftime("%H:%M:%S", now)
    print(f"{hand_type} hand - Knuckle {knuckle} Coordinate:
({x}, {y}), Distance: {distance} mm")
    data.append({
        "Time": current_time,
        "Frame": frame_number,
        "HandType": hand_type,
        "KnuckleID": knuckle,
        "CoordinateX": x,
        "CoordinateY": y,
        "Distance": distance
    })

# Display the resulting frame with hand landmarks and depth information
cv2.imshow('Hand Landmarks with Depth', color_frame)
frame_number = frame_number + 1
if cv2.waitKey(33) == ord('q'): # press the 'q' key to end the trial
    break

# Release the capture and destroy OpenCV windows
cv2.destroyAllWindows()

```



```
# Create a DataFrame and save it to Excel
df = pd.DataFrame(data)
excel_file_path = r'C:\Users\cindy\OneDrive\Documents\Spring 24\ME
494H\thesis\realsense\realsense\hand_tracking_data.xlsx'
df.to_excel("hand_tracking_data.xlsx", index=False)
```

## Appendix B: MATLAB Code

### MATLAB Code for XY Trials

```

clear
clc
close all
% code for analyzing hand tracking data

% pull the excel file that you are reading data from
data = readtable('hand_tracking_data_f_15.xlsx');

% create variables for each column of data from excel file
time = table2array(data(:, "Time"));
knuckleID = table2array(data(:, "KnuckleID"));
hand_type = table2array(data(:, "HandType"));
CoordinateX = table2array(data(:, "CoordinateX"));
CoordinateY = table2array(data(:, "CoordinateY"));
depth = table2array(data(:, "Distance"));

% reconstruct movement of knuckles - initialize animated lines
an_4 = animatedline('Marker', '.', 'Color', 'c');
an_8 = animatedline('Marker', '.', 'Color', 'r');
an_12 = animatedline('Marker', '.', 'Color', 'b');
an_16 = animatedline('Marker', '.', 'Color', 'm');
an_20 = animatedline('Marker', '.', 'Color', 'g');

figure(1) % this figure will show the data markers in 3D during the movement
xlabel('x')
ylabel('y')
zlabel('z')
title('Movement of Points of Interest 4, 8, 12, 16, and 20 vs time')
legend('POI 4', 'POI 8', 'POI 12', 'POI 16', 'POI 20')
grid on

% plotting the different knuckle movements over time
for iii = 1:length(time)
    iii;
    if knuckleID(iii) == 4
        x_4(iii) = CoordinateX(iii);
        y_4(iii) = CoordinateY(iii);
        z_4(iii) = depth(iii);
        addpoints(an_4, x_4(iii), y_4(iii), z_4(iii));
        % drawnow
        pause(0.01)
    elseif knuckleID(iii) == 8
        x_8(iii) = CoordinateX(iii);
        y_8(iii) = CoordinateY(iii);
        z_8(iii) = depth(iii);

```

```

        addpoints(an_8,x_8(iii),y_8(iii),z_8(iii));
    % drawnow
    pause(0.01)
elseif knuckleID(iii) == 12
    x_12(iii) = CoordinateX(iii);
    y_12(iii) = CoordinateY(iii);
    z_12(iii) = depth(iii);
    addpoints(an_12,x_12(iii),y_12(iii),z_12(iii));
    % drawnow
    pause(0.01)
elseif knuckleID(iii) == 16
    x_16(iii) = CoordinateX(iii);
    y_16(iii) = CoordinateY(iii);
    z_16(iii) = depth(iii);
    addpoints(an_16,x_16(iii),y_16(iii),z_16(iii));
    % drawnow
    pause(0.01)
    iii+1;
else
    x_20(iii) = CoordinateX(iii);
    y_20(iii) = CoordinateY(iii);
    z_20(iii) = depth(iii);
    addpoints(an_20,x_20(iii),y_20(iii),z_20(iii));
    % drawnow
    pause(0.01)
    iii+1;
end
iii;
end

%%

% filter the position data to only have non-zero results
nonZeroIndices_4= x_4 ~=0;
x_4 = x_4(nonZeroIndices_4);
nonZeroIndices_8= x_8 ~=0;
x_8 = x_8(nonZeroIndices_8);
nonZeroIndices_12= x_12 ~=0;
x_12 = x_12(nonZeroIndices_12);
nonZeroIndices_16= x_16 ~=0;
x_16 = x_16(nonZeroIndices_16);
nonZeroIndices_20= x_20 ~=0;
x_20 = x_20(nonZeroIndices_20);

%% the res of this code is for finding the time and averages
elapsedTime = 42; % time of experiment in seconds
time_4 = linspace(0,elapsedTime,length(x_4));
time_8 = linspace(0,elapsedTime,length(x_8));
time_12 = linspace(0,elapsedTime,length(x_12));
time_16 = linspace(0,elapsedTime,length(x_16));
time_20 = linspace(0,elapsedTime,length(x_20));

```

```

%% create subplot of all 5 points of interest and one overlaying
figure(2)
subplot(3,2,1)
hold on
xlabel('time (s)')
ylabel('x position (mm)')
title('POI 4')
grid on
plot(time_4,x_4)

subplot(3,2,2)
hold on
xlabel('time (s)')
ylabel('x position (mm)')
title('POI 8')
grid on
plot(time_8,x_8)
hold off

subplot(3,2,3)
hold on
xlabel('time (s)')
ylabel('x position (mm)')
title('POI 12')
grid on
plot(time_12,x_12)
hold off

subplot(3,2,4)
hold on
xlabel('time (s)')
ylabel('x position (mm)')
title('POI 16')
grid on
plot(time_16,x_16)
hold off

subplot(3,2,5)
hold on
xlabel('time (s)')
ylabel('x position (mm)')
title('POI 20')
grid on
plot(time_20,x_20)
hold off

subplot(3,2,6)
grid on
plot(time_4,x_4)
hold on

```

```

plot(time_8,x_8)
hold on
plot(time_12,x_12)
hold on
plot(time_16,x_16)
hold on
plot(time_20,x_20)
hold off
grid on
xlabel('time (s)')
ylabel('x position (mm)')
title('All POIs overlayed')
legend('4', '8', '12', '16', '20')

% overlaying all plots onto one
figure(3)

grid on
plot(time_4,x_4)
hold on
plot(time_8,x_8)
hold on
plot(time_12,x_12)
hold on
plot(time_16,x_16)
hold on
plot(time_20,x_20)
hold off
xlabel('time (s)')
ylabel('x position (mm)')
title('All POIs overlayed')
legend('4', '8', '12', '16', '20')

% finding average values

% Example: Assuming start_time and end_time are the start and end times of your
desired range
start_time = 36.5;
end_time = 41.3;
start_index = find(time_4 >= start_time, 1);
end_index = find(time_4 <= end_time, 1, 'last');

% Extract the data within the specified time range - change to x_? for each
data_range_4 = x_4(1,start_index:end_index);
data_range_8 = x_8(1,start_index:end_index);
data_range_12 = x_12(1,start_index:end_index);
data_range_16 = x_16(1,start_index:end_index);
data_range_20 = x_20(1,start_index:end_index);

% Calculate the average value
average_value_4 = mean(data_range_4)

```

```

average_value_8 = mean(data_range_8)
average_value_12 = mean(data_range_12)
average_value_16 = mean(data_range_16)
average_value_20 = mean(data_range_20)
%
% finding distance traveled - used this to quantify and analyze
%
x_distance_4 = max(x_4) - min(x_4); % in mm
% y_distance_4 = y_4(1,iii-1) - y_4(1,1) % in mm
% z_distance_4 = z_4(1,iii-1) - z_4(1,1) % in mm
%
x_distance_8 = max(x_8) - min(x_8); % in mm
% y_distance_8 = y_12(1,iii-1) - y_8(1,1) % in mm
% z_distance_8 = z_12(1,iii-1) - z_8(1,1) % in mm
%
x_distance_12 = max(x_12) - min(x_12); % in mm
% y_distance_12 = y_12(1,iii-1) - y_12(1,1) % in mm
% z_distance_12 = z_12(1,iii-1) - z_12(1,1) % in mm
%
x_distance_16 = max(x_16) - min(x_16); % in mm
% y_distance_16 = y_12(1,iii-1) - y_16(1,1) % in mm
% z_distance_16 = z_12(1,iii-1) - z_16(1,1) % in mm
%
x_distance_20 = max(x_20) - min(x_20); % in mm
% y_distance_20 = y_12(1,iii-1) - y_20(1,1) % in mm
% z_distance_20 = z_12(1,iii-1) - z_20(1,1) % in mm

```

## MATLAB Code for YZ Trials

```

clear
clc
close all
% starter code for analyzing hand tracking data

% pull up and read desired excel book
data = readtable('htd_s_13.xlsx');

time = table2array(data(:, "Time"));
knuckleID = table2array(data(:, "KnuckleID"));
hand_type = table2array(data(:, "HandType"));
CoordinateX = table2array(data(:, "CoordinateX"));
CoordinateY = table2array(data(:, "CoordinateY"));
depth = table2array(data(:, "Distance"));

% reconstruct movement of knuckles
an_4 = animatedline('Marker', '.', 'Color', 'c');
an_8 = animatedline('Marker', '.', 'Color', 'r');
an_12 = animatedline('Marker', '.', 'Color', 'b');
an_16 = animatedline('Marker', '.', 'Color', 'm');
an_20 = animatedline('Marker', '.', 'Color', 'g');

figure(1)
xlabel('x')
ylabel('y')
zlabel('z')
title('Movement of POIs 4, 8, 12, 16, and 20 vs time')
legend('POI 4', 'POI 8', 'POI 12', 'POI 16', 'POI 20')
grid on

% plotting the different knuckle movements over time
% comment out the addpoints commands to skip this step
for iii = 1:length(time)
    iii;
    if knuckleID(iii) == 4
        x_4(iii) = CoordinateX(iii);
        y_4(iii) = CoordinateY(iii);
        z_4(iii) = depth(iii);
        %addpoints(an_4,x_4(iii),y_4(iii),z_4(iii));
        drawnow
        %pause(0.001)
    elseif knuckleID(iii) == 8
        x_8(iii) = CoordinateX(iii);
        y_8(iii) = CoordinateY(iii);
        z_8(iii) = depth(iii);
        %addpoints(an_8,x_8(iii),y_8(iii),z_8(iii));
        % drawnow
        %pause(0.001)
    elseif knuckleID(iii) == 12

```

```

        x_12(iii) = CoordinateX(iii);
        y_12(iii) = CoordinateY(iii);
        z_12(iii) = depth(iii);
        %addpoints(an_12,x_12(iii),y_12(iii),z_12(iii));
        % drawnow
        % pause(0.001)
    elseif knuckleID(iii) == 16
        x_16(iii) = CoordinateX(iii);
        y_16(iii) = CoordinateY(iii);
        z_16(iii) = depth(iii);
        %addpoints(an_16,x_16(iii),y_16(iii),z_16(iii));
        % drawnow
        % pause(0.001)
        iii+1;
    else
        x_20(iii) = CoordinateX(iii);
        y_20(iii) = CoordinateY(iii);
        z_20(iii) = depth(iii);
        %addpoints(an_20,x_20(iii),y_20(iii),z_20(iii));
        % drawnow
        %pause(0.001)
        iii+1;
    end
    iii;
end

%%
% need to filter the position data so that we only have non-zero results
nonZeroIndices_4= z_4 ~=0;
z_4 = z_4(nonZeroIndices_4);
nonZeroIndices_8= z_8 ~=0;
z_8 = z_8(nonZeroIndices_8);
nonZeroIndices_12= z_12 ~=0;
z_12 = z_12(nonZeroIndices_12);
nonZeroIndices_16= z_16 ~=0;
z_16 = z_16(nonZeroIndices_16);
nonZeroIndices_20= z_20 ~=0;
z_20 = z_20(nonZeroIndices_20);

elapsedTime = 60; % time of experiment in seconds
time_4 = linspace(0,elapsedTime,length(z_4));
time_8 = linspace(0,elapsedTime,length(z_8));
time_12 = linspace(0,elapsedTime,length(z_12));
time_16 = linspace(0,elapsedTime,length(z_16));
time_20 = linspace(0,elapsedTime,length(z_20));

%%
figure(2)

subplot(1,3,1)
hold on

```



```

xlabel('time (s)')
ylabel('z position (mm)')
title('POI 8')
grid on
plot(time_8,z_8)
hold off

subplot(1,3,2)
hold on
xlabel('time (s)')
ylabel('z position (mm)')
title('POI 12')
grid on
plot(time_12,z_12)
hold off

subplot(1,3,3)
grid on
plot(time_8,z_8)
hold on
plot(time_12,z_12)

hold off
grid on
xlabel('time (s)')
ylabel('x position (mm)')
title('Both POIs overlayed')
legend('8','12')

% finding distance traveled
z_distance_4 = max(z_4) - min(z_4); % in mm
% y_distance_4 = y_4(1,iii-1) - y_4(1,1) % in mm
% z_distance_4 = z_4(1,iii-1) - z_4(1,1) % in mm
%
z_distance_8 = max(z_8) - min(z_8); % in mm
% y_distance_8 = y_12(1,iii-1) - y_8(1,1) % in mm
% z_distance_8 = z_12(1,iii-1) - z_8(1,1) % in mm
%
z_distance_12 = max(z_12) - min(z_12); % in mm
% y_distance_12 = y_12(1,iii-1) - y_12(1,1) % in mm
% z_distance_12 = z_12(1,iii-1) - z_12(1,1) % in mm
%
z_distance_16 = max(z_16) - min(z_16); % in mm
% y_distance_16 = y_12(1,iii-1) - y_16(1,1) % in mm
% z_distance_16 = z_12(1,iii-1) - z_16(1,1) % in mm
%
z_distance_20 = max(z_20) - min(z_20); % in mm
% y_distance_20 = y_12(1,iii-1) - y_20(1,1) % in mm
% z_distance_20 = z_12(1,iii-1) - z_20(1,1) % in mm

%% overlaying all of them
figure(3)

```

```

grid on
plot(time_4,z_4)
hold on
plot(time_8,z_8)
hold on
plot(time_12,z_12)
hold on
plot(time_16,z_16)
hold on
plot(time_20,z_20)
hold off
xlabel('time (s)')
ylabel('z position (mm)')
title('All POIs overlayed')
legend('4', '8','12','16','20')

%% finding average values

% Example: Assuming start_time and end_time are the start and end times of your
desired range
start_time = 52.9;
end_time = 58;
start_index = find(time_8 >= start_time, 1);
end_index = find(time_8 <= end_time, 1, 'last');

% Extract the data within the specified time range - change to x_? for each

data_range_8 = z_8(1,start_index:end_index);
data_range_12 = z_12(1,start_index:end_index);

% Calculate the average value
average_value_8 = mean(data_range_8)
average_value_12 = mean(data_range_12)

```

### Appendix C: Additional Subplots for Trials

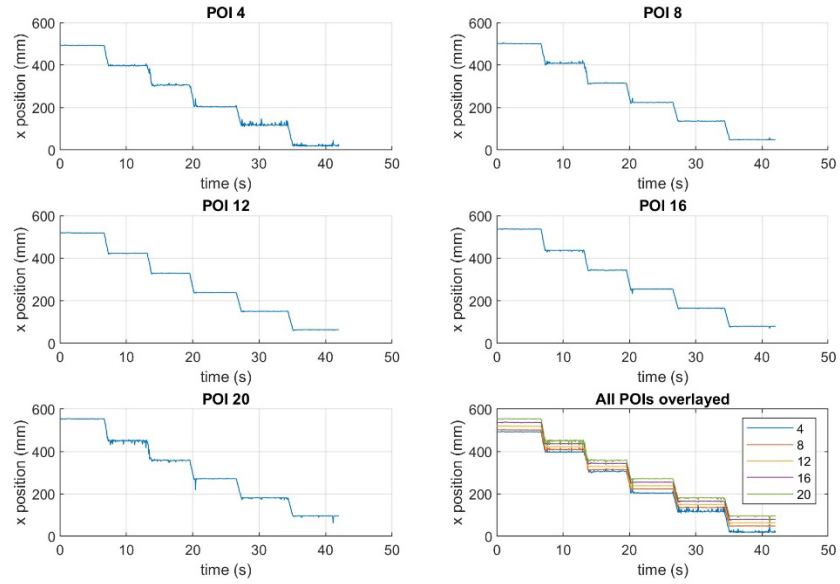


Figure 11. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 2

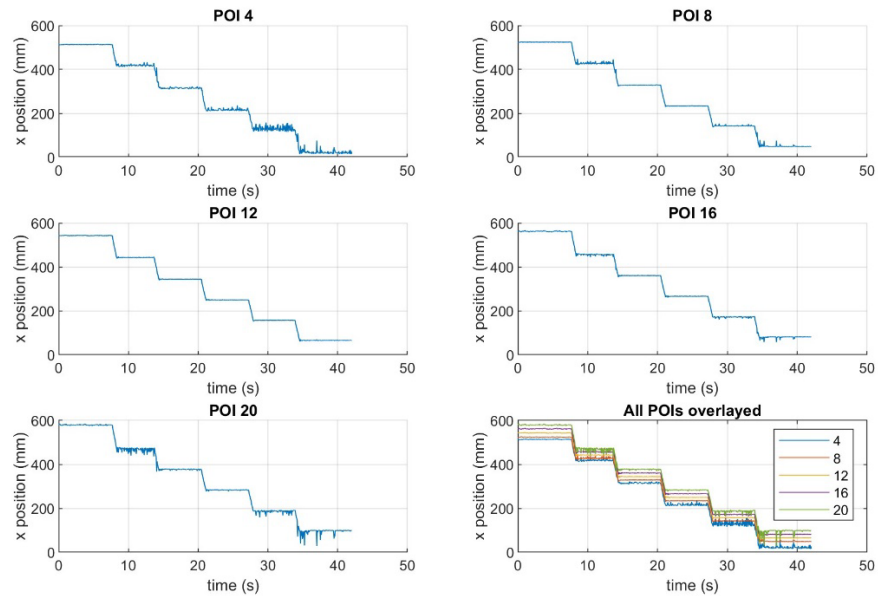
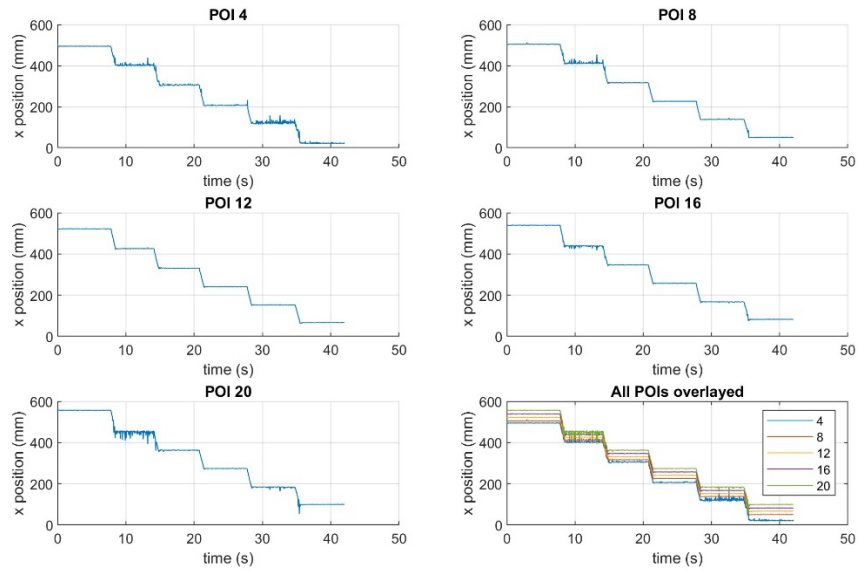
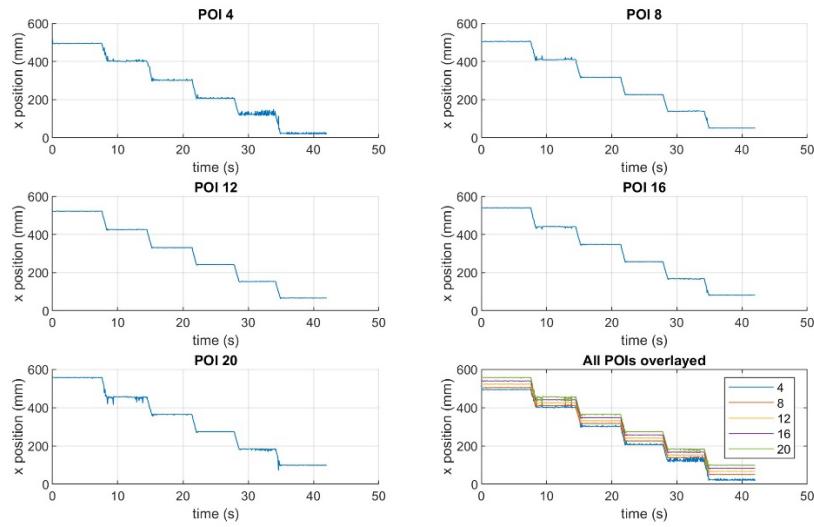


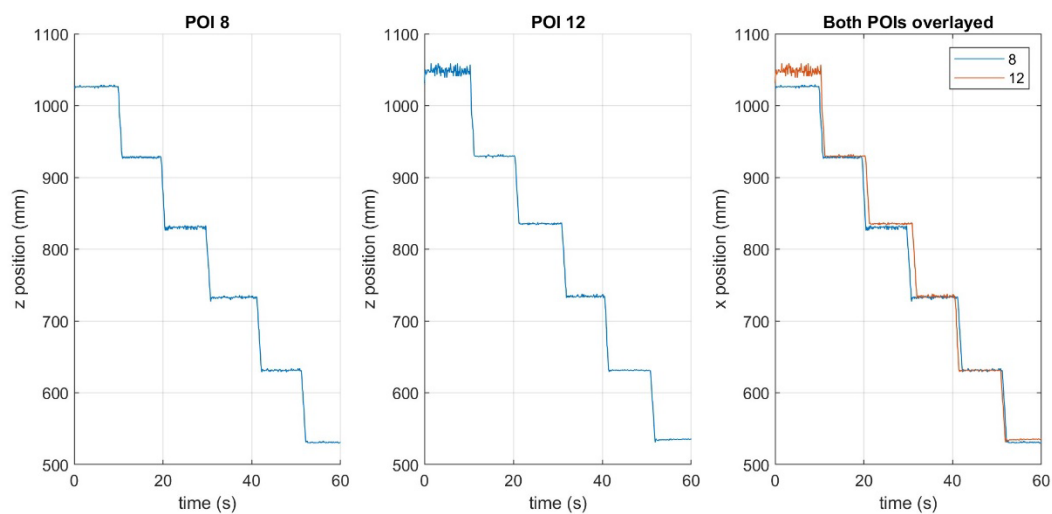
Figure 12. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 3



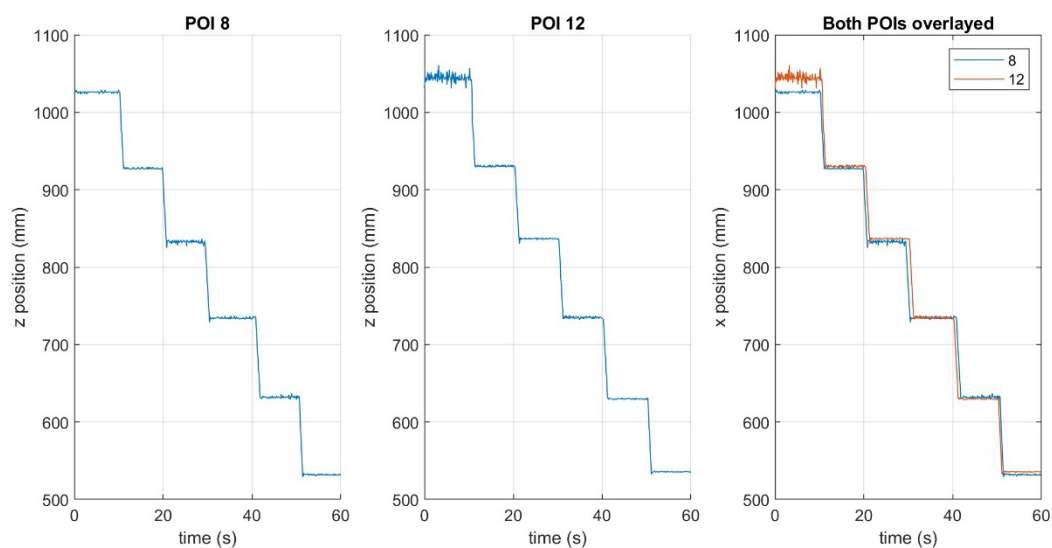
**Figure 13. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 4**



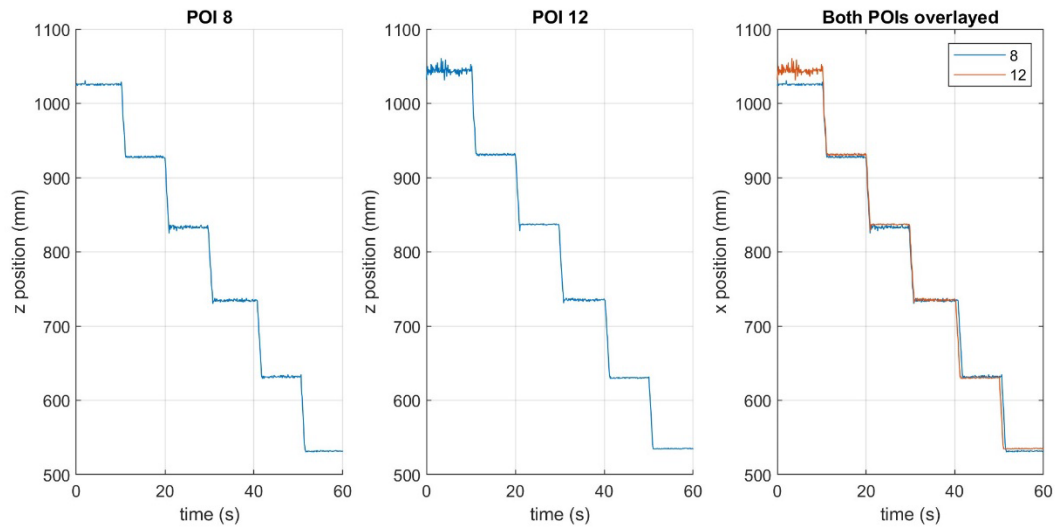
**Figure 14. X-Position vs. Time for Points of Interest 4, 8, 12, 16, and 20 for Trial 5**



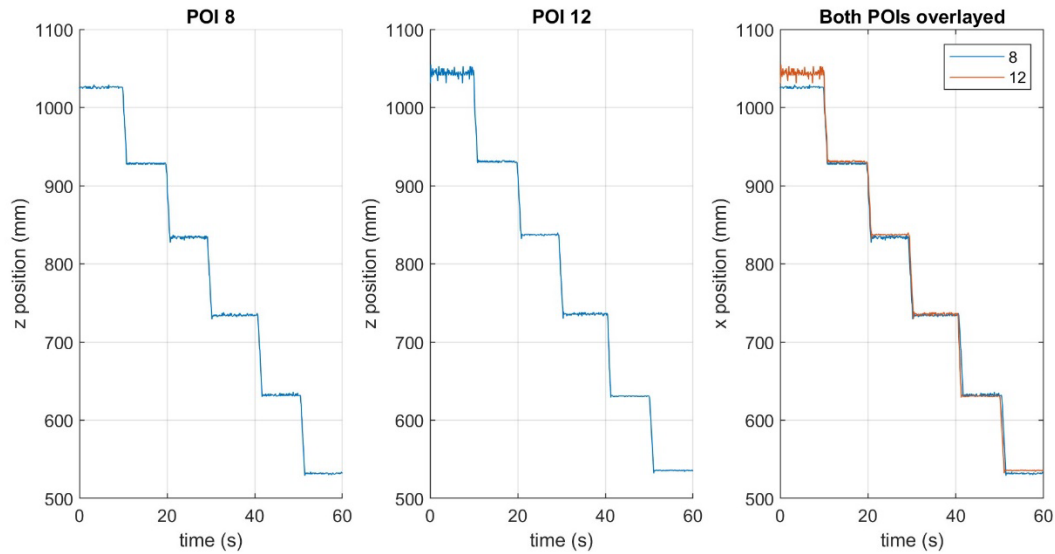
**Figure 15. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 2**



**Figure 16. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 3**



**Figure 17. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 4**



**Figure 18. Z-Position vs. Time for Points of Interest 8 and 12 for Trial 5**

## BIBLIOGRAPHY

- [1] A. Sturgeon, E. Sarraf, and J. Moore, "MOTION BASED FEEDBACK TRAINING SYSTEM FOR ENDOTRACHEAL INTUBATION." [Online]. Available: <http://asmedigitalcollection.asme.org/BIOMED/proceedings-pdf/DMD2022/84815/V001T11A008/6877148/v001t11a008-dmd2022-1024.pdf>
- [2] H. E. Wang, J. R. Lave, C. A. Sirio, and D. M. Yealy, "Paramedic intubation errors: Isolated events or symptoms of larger problems?," *Health Aff*, vol. 25, no. 2, pp. 501–509, Mar. 2006, doi: 10.1377/HLTHAFF.25.2.501.
- [3] H. E. Wang, L. J. Cook, C. C. H. Chang, D. M. Yealy, and J. R. Lave, "Outcomes after out-of-hospital endotracheal intubation errors," *Resuscitation*, vol. 80, no. 1, pp. 50–55, Jan. 2009, doi: 10.1016/J.RESUSCITATION.2008.08.016.
- [4] S. H. Katz and J. L. Falk, "Misplaced endotracheal tubes by paramedics in an urban emergency medical services system," *Ann Emerg Med*, vol. 37, no. 1, pp. 32–37, Jan. 2001, doi: 10.1067/MEM.2001.112098.
- [5] R. Chemsian, S. Bhananker, and R. Ramaiah, "Videolaryngoscopy," *Int J Crit Illn Inj Sci*, vol. 4, no. 1, p. 35, 2014, doi: 10.4103/2229-5151.128011.
- [6] D. Greer, K. E. Marshall, S. Bevans, A. Standlee, P. McAdams, and W. Harsha, "Review of videolaryngoscopy pharyngeal wall injuries," *Laryngoscope*, vol. 127, no. 2, pp. 349–353, Feb. 2017, doi: 10.1002/LARY.26134.
- [7] "EM\_Key\_Index\_Procedure\_Minimums\_103117".

- [8] N. Singh *et al.*, “Impact of multiple intubation attempts on adverse tracheal intubation associated events in neonates: a report from the NEAR4NEOS,” *J Perinatol*, vol. 42, no. 9, pp. 1221–1227, Sep. 2022, doi: 10.1038/S41372-022-01484-5.
- [9] G. Berisha, A. M. Boldingh, E. W. Blakstad, A. E. Rønnestad, and A. L. Solevåg, “Management of the Unexpected Difficult Airway in Neonatal Resuscitation,” *Front Pediatr*, vol. 9, Oct. 2021, doi: 10.3389/fped.2021.699159.
- [10] “Case Minimum Requirements: Surgical Critical Care Effective Beginning Academic Year 2019-2020.”
- [11] J. Yee, C. San Miguel, S. Khandelwal, D. Way, and A. Panchal, “Procedural Curriculum to Verify Intern Competence Prior to Patient Care,” *Western Journal of Emergency Medicine*, vol. 24, no. 1, pp. 8–14, Dec. 2022, doi: 10.5811/westjem.2022.11.58057.
- [12] J. T. Mulcaster *et al.*, “Laryngoscopic Intubation Learning and Performance,” 2003. [Online]. Available: <http://pubs.asahq.org/anesthesiology/article-pdf/98/1/23/407139/0000542-200301000-00007.pdf>
- [13] A. G. Miller, “Endotracheal Intubation Training and Skill Maintenance for Respiratory Therapists,” *Respir Care*, vol. 62, no. 2, pp. 156–162, Feb. 2017, doi: 10.4187/RESPCARE.05037.
- [14] V. Vattanavanit, J. Kawla-Ied, and R. Bhurayanontachai, “High-fidelity medical simulation training improves medical students’ knowledge and confidence levels in septic shock resuscitation,” *Open Access Emergency Medicine*, vol. 9, pp. 1–7, 2017, doi: 10.2147/OAEM.S122525.



- [15] V. Chunduru, M. Roy, D. Romit N. S, and R. G. Chittawadigi, “Hand Tracking in 3D Space using MediaPipe and PnP Method for Intuitive Control of Virtual Globe,” *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 1–6, 2021.
- [16] G. Amprimo, C. Ferraris, G. Masi, G. Pettiti, and L. Priano, “GMH-D: Combining Google MediaPipe and RGB-Depth Cameras for Hand Motor Skills Remote Assessment,” in *Proceedings - 2022 IEEE International Conference on Digital Health, ICDH 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 132–141. doi: 10.1109/ICDH55609.2022.00029.
- [17] G. Güney *et al.*, “Video-Based Hand Movement Analysis of Parkinson Patients before and after Medication Using High-Frame-Rate Videos and MediaPipe,” *Sensors (Basel)*, vol. 22, no. 20, Oct. 2022, doi: 10.3390/s22207992.
- [18] “What is NI LabVIEW? Graphical Programming for Test & Measurement - NI.” Accessed: Mar. 18, 2024. [Online]. Available: <https://www.ni.com/en/shop/labview.html>