Essential Tremor Measurement and Analysis

Hayden Burt hayden.burt5@gmail.com

Poojit Hegde phalpha1729@gmail.com

Amanda Smalfus amandasmalfus@yahoo.com

Alex Weiner*

New Jersey Governor's School of Engineering and Technology

21 July 2017

*Corresponding Author

***Abstract*— A computer sensor network is developed to monitor hand position of those with *essential tremor*, a nervous system disorder which causes uncontrollable shaking, primarily in the hands and upper body [1]. The network collects three-dimensional positional data using two ZX Distance and Gesture Sensors, an Arduino Uno board, and a Raspberry Pi. The ZX Distance and Gesture Sensors, which use two infrared (IR) light emitting diodes (LEDs) and a receiver to collect positional data, are connected to the Arduino. From there, the data is transmitted to the Raspberry Pi via an I2C multiplexer. Data from a series of carefully selected hand movements performed by a subject without essential tremor is collected and analyzed using Python to determine network accuracy. Two-dimensional graphs of raw data in the form of x-, y-, or z-coordinates over time, moving data features, and three-dimensional plots are produced. Theoretical plots are also generated to outline ideal hand motion. Average deviation is calculated for each task performed and found to be no higher than 0.24 inches. These values quantify the shakiness of the normal hand relative to ideal paths of motion. With data collected from real tremor patients, this computer sensor network could revolutionize the ability to monitor and classify essential tremor for the millions of patients who struggle to find treatment.**

I. Introduction

Computer sensor networks, consisting of sensors to monitor physical conditions and computers to analyze and process the sensor data, are used for a wide range of consumer, industrial, military, and healthcare applications. The computer sensor network developed in this study has the potential to influence the modern healthcare system.

Computer sensor networks have an integral application in the diagnosis of and care for the essential tremor, a common, but misunderstood, nerve disorder that causes shaking in the upper body [2]. With new computer sensor technology available, essential tremor can be measured quantitatively using positional data, contrary to current tools which rely largely upon qualitative information [3]. This study looks to develop a sophisticated computer sensor network capable of collecting and analyzing two- and three-dimensional positional data using infrared sensors and a microcomputer. The network allows for comparison, classification, and analysis of multiple data features collected from tremor patient hand motion.

II. Background

A. *Essential Tremor Causes and Characteristics*

The essential hand tremor refers to the involuntary shaking of the hands, legs, head, or vocal area due to uncontrollable muscle contraction. The symptoms of this nerve disorder are most pronounced during active movement, contrary to the symptoms of related conditions, such as Parkinson's Disease, whose tremors are most pronounced during periods of rest [4].

The essential tremor can affect up to ten million people in the United States alone on any given day and can make basic activities extraordinarily difficult. The symptoms can be detrimental to daily activities and independent function, including the ability to speak with clarity. In particular, the hand tremors associated with this condition make it very difficult and frustrating for patients to complete basic tasks such as eating, drinking, putting on makeup, shaving, or writing. Though it has many causes and can appear at any age, the essential tremor is most common among the elderly population [4].

Despite its widespread occurrence, relatively little is known about the root cause of essential tremors. It is thought that around half of the cases, known as familial tremors, are caused by genetic mutation. It is believed that the thalamus, which helps to control muscular activity in the brain may play a role in processing these abnormal signals. Its symptoms are worsened by external factors such as over caffeination, lack of sleep, and stress. Nonetheless, predicting the development of this condition is extraordinarily difficult because so much about the essential tremor remains unknown [4].

Currently, there are a very limited number of tools for measurement and analysis of essential tremors, despite the high numbers of patients who suffer from this condition. The Essential Tremor Rating Assessment Scale (TETRAS) created by the Tremor Research Group [5] is one such tool. Many of the current tools, TETRAS included, rely upon tests such as handwriting tasks and the dot approximation task, in which a patient is asked to hold the tip of a pen just above a small dot. The results are calculated and recorded using a qualitative scale based on how "noticeable" the tremor is in each test. Qualitative data can be affected by personal bias and is difficult to compare from case to case. Prior to this study, the state of the art essential tremor measurement systems consisted of assessment scales that assign a rating based on a doctor's perception of the tremor [5]. However, the computer sensor network designed and tested in this study collects quantitative, rather than qualitative, information and allows the essential tremor to be analytically compared to a set of baseline data. Developing a more quantitative method to measure and analyze tremors would be a significant improvement in decreasing subjectivity that can lead to invalid or deceptive analysis.

B. ZX Gesture and Distance Sensors

The ZX Distance and Gesture Sensors [6] are composed of two infrared (IR) LEDs, an infrared receiver, and the PIC16F1823 microcontroller. The sensors measure an object's distance through the use of two infrared light emitting diodes. The emitted infrared light reflects off of an object located above the sensor, in this case a hand or small rectangular plate held by the subject. The receiver in the center of the ZX Distance and Gesture Sensor then detects the returning infrared light.

These sensors are able to measure distance of an object approximately twelve inches vertically above the sensor (z-axis) and six inches horizontally across the sensor (x-axis) and output samples at a rate of 1200 samples per second, or 9600 baud [6].

C. Arduino Uno and Raspberry Pi 3

The Arduino Uno SMD Edition [7] is a microcontroller approximately the size of a playing card. It is usually the central component in embedded systems. Equipped with digital and analog pins, an Arduino board is capable of connecting to a variety of sensors as well as other microcomputers, such as the Raspberry Pi [8]. The job of the Arduino board used in this study is to collect inputs from the environment via the attached sensors . The Arduino runs on C script and works with the Linux computer operating system installed on the Raspberry Pi.

The Raspberry Pi 3 is a simple yet powerful microcomputer designed to be affordable and easy to use. Like Arduino, Raspberry Pi models can be used to produce thousands of unique projects and are programmed for specific purposes. Both are designed to be accessible to programmers of all experience levels. Raspberry Pi computers include multiple USB ports, an HDMI port, and an 8 GB microSD (secure digital) card as a memory source for storing incoming and saved information. The USB is used for receiving serial data from the Arduino, which is stored on the microSD. When connected to a monitor and keyboard, a Raspberry Pi has the capability to perform as an average laptop or PC. As a microcomputer with a central processing unit, a Raspberry Pi has much more sophisticated capabilities for data storage and acquisition than the average Arduino board [9].

D. Serial Communication and Inter-Integrated Circuit Protocol (I2C)

In a computer sensor network, such as the one in this study, it is necessary to transmit data from one device to another. Data can be sent from an Arduino to a Raspberry Pi device using a variety of serial communication methods [10]. One method of transmitting data is through the inter-integrated circuit (I2C) serial protocol [11]. The protocol makes it possible for many "slave" devices to communicate with a single "master" device over a short distance. The slave devices output the data to the master device, while the master device receives and records the data. The I2C bus consists of the serial clock line (SCL) and serial data line (SDL). The SCL synchronizes the devices and the SDL transmits the data itself [13].

The Arduino Uno board by itself is not equipped to receive data from two ZX Gesture Sensors with the same address on the same SDA/SCL pins [14]. As a solution to this issue, this study uses an TCA9548A 1-to-8 I2C multiplexer [15], which allows the Arduino Uno board to collect data from both sensors simultaneously. The multiplexer is capable of directing commands to 8 different I2C devices, even if they have the same address. With successful computer code, the multiplexer can direct commands to specific pins on the I2C bus.

E. Programming with C and Python

C is a compiled programming language, which means that a compiler is necessary to translate the code before it can be used. On the other hand, Python is considered an interpreted language, which means that the instructions from the code are executed directly. Despite these differences, both languages are useful for a different purpose in the network in this study.

The Arduino Uno is programmed using the Arduino language, which is essentially comprised of a set of functions in C. The language C is often used to interface with hardware, making it particularly useful for embedded systems programming. Therefore, it is primarily used to collect data from the Arduino microcontroller.

The Raspberry Pi computers typically use the high level programming language Python. Since Python places a focus on readability of the code, it is commonly used in the field of robotics and data analysis. In this study, a Python script is used to log the serial data from the Arduino as well run the analysis of the collected data.

F. Data Analysis

When performing data analysis it is of great importance to compare multiple sets of derived values, or features. Features include various calculations from the original data set, such as average value, standard deviation, variance, and root mean square. When looking at positional data collected from a test subject, even the smallest of hand tremors are collected and displayed in a graphical analysis. As a result, the raw data tends to be noisy and difficult to study. It is necessary to compute features because raw data may not be enough to capture movement that is out of the ordinary or it may be lost amidst the small, natural discrepancies that those without essential tremor also display. For example, collected data sets may seem to differ by just a few percentage points, but the root mean square of those data sets may be drastically different. Without the derived data features, data comparison at face value may not reveal differences that could be integral in classification and analysis.

All features derived in this study are moving features. The purpose of completing moving calculations is to clearly depict how data changes over time as well as to reduce the impact of outliers as a result of hardware or software errors. The moving average is a processed representation of fluctuating individual sensor data that presents more valuable information than the original data. This is done by sectioning, or windowing the data. Rather than obtaining one average, standard deviation, root mean square, etc. for a collection of thousands of data points, a moving function produces a value for every window of data, revealing how the calculation fluctuates over the course of data collection.

In addition to derived features, it is necessary to quantitatively assess the extent to which the collected data deviates from a theoretical ideal. This can be done by summing the difference between the raw and theoretical data at each point and dividing by the total number of samples to get an average deviation value. This equation can produce numerical values indicative of the extent to which the raw data varies relative to the theoretical data. The deviation equation,

$$\frac{\sum |f(t)-g(t)|}{s} \tag{1}$$

is used, where f(t) represents the raw data line over time, g(t) represents the ideal data line over time, and s represents the number of samples collected. This can be used to quantitatively measure the extent of hand instability, and in the case of this study identify a baseline for “normal” hand movement, which by no means can be considered perfectly smooth motions.

III. Experimental Approach: Developing A Computer Sensor Network

The computer sensor network consists of multiple elements to collect, log, and analyze positional data. The diagram in Fig. 3 illustrates the path of these elements in the network. Hand position is measured using two ZX sensors. The data from these sensors is collected using an Arduino Uno. The data is then sent serially from the Arduino to a Raspberry Pi 3 using the I2C protocol. A Python script is used to log the collected data onto a file on the Raspberry Pi where it is then analyzed and graphed.

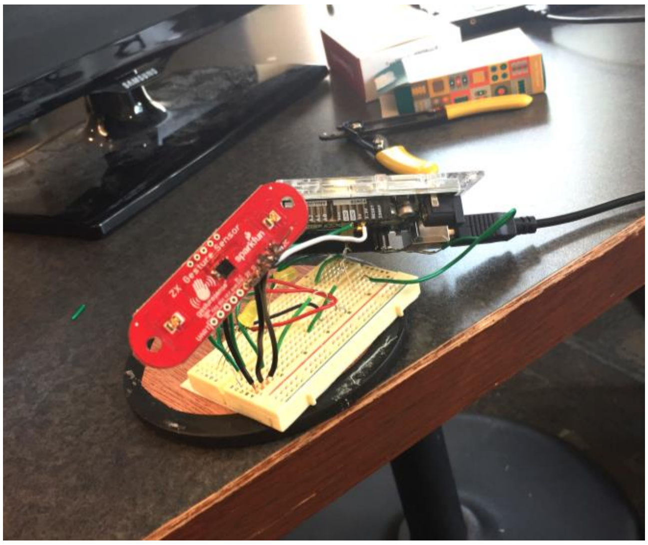

Fig. 1 Preliminary setup of one ZX Gesture Sensor to an Arduino Uno using a breadboard, used in early stages of network development.

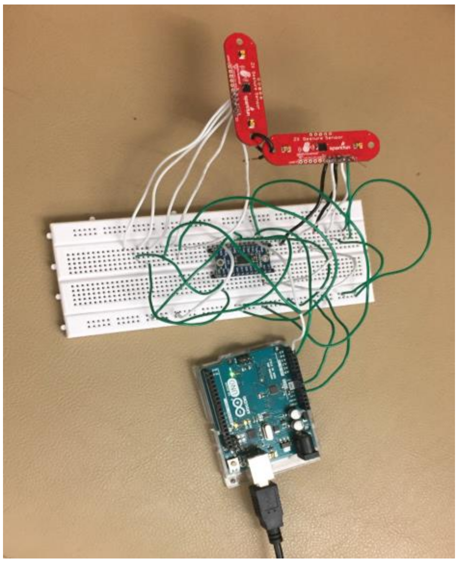

Fig. 2 Final computer sensor network design. Two ZX Gesture Sensors are arranged perpendicular to one another and connected to an I2C multiplexer, which is wired to an Arduino Uno using a breadboard. The Arduino sends the sensor data to a Raspberry Pi, not depicted in this figure.

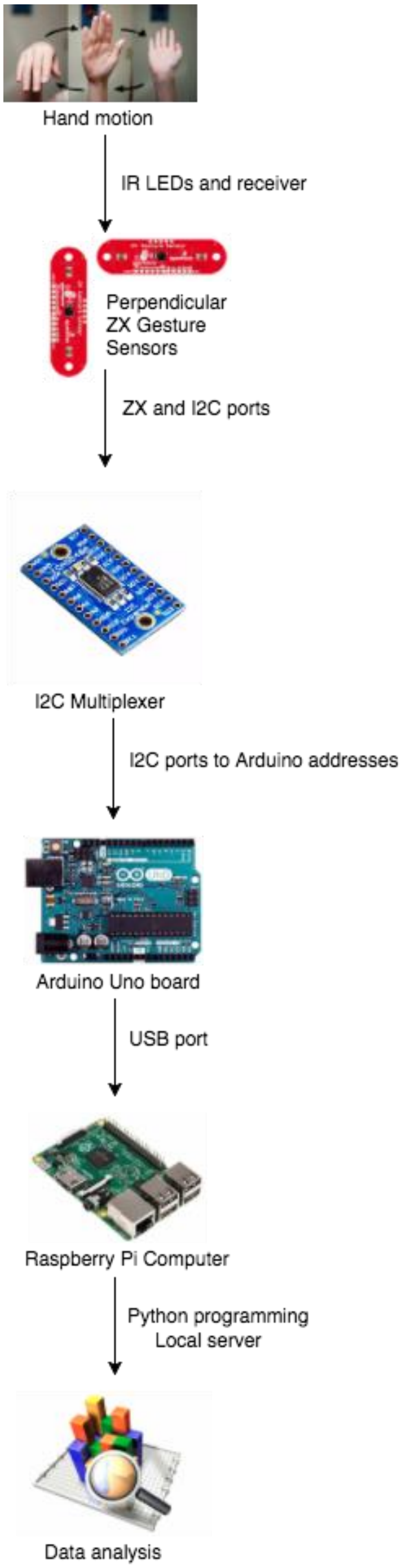

Fig. 3 Flowchart depicting a simplified outline of the computer sensor network proposed by this paper.

A. Assembling the Hardware Components

The computer sensor network begins with two ZX Distance and Gesture Sensors, which use infrared LEDs and the PIC16F1823 microcontroller to measure the reflected infrared (IR) light. These sensors are wired to the TCA9548A 1-to-8 I2C multiplexer to allow communication from two sensors across a single address. A breadboard is used to wire the ZX Gesture Sensors to the Arduino Uno. The ZX Gesture Sensors were secured in a perpendicular arrangement by bending their attached wires at the correct angle, tying them together with a small piece of wire, and mounting on cardboard to maintain stability. The Arduino Uno was connected to the Raspberry Pi via the Raspberry Pi's USB port.

B. Collecting Three-Dimensional Positional Data

The two ZX Gesture Sensors were used to collect three-dimensional positional data of a hand or object. A single sensor is capable of collecting data along two axes, but in order to obtain three-dimensional data, two sensors must be used. Placing two sensors perpendicular to one another allows for the collection of data from an x-, y-, and z-axis. The layout of the two ZX sensors can be seen in Fig. 2. The diagram in Fig. 4 outlines the range of each sensor as well as the x and y coordinates being used for measurement.

Each sensor recorded values from 0 to 240 [6] both vertically and horizontally. When the hand was out of range for one of these sensors, a negative one was recorded. This way, the sensors continued to collect data even when the hand was out of range. The data recorded while the hand was outside the sensing range was filtered out of the data set prior to analysis using Python script.

In this study, a subject without essential tremor was asked to complete a series of specifically defined tasks to test network functionality. This was done in order to generate a baseline, or set of control data, that accounts for natural hand unsteadiness unrelated to essential tremor.

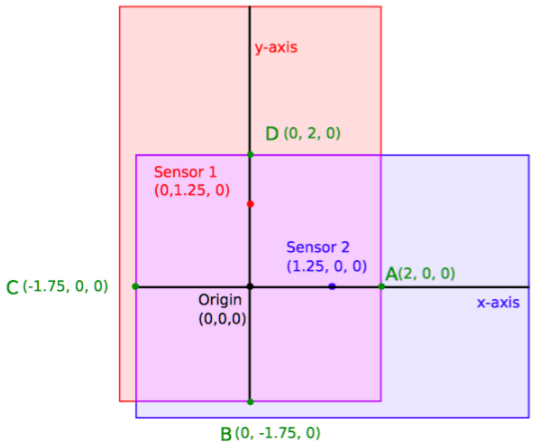

Fig. 4 Diagram depicting the range of each sensor in a three-dimensional coordinate system.. Sensor 1's range is represented by the red rectangle and Sensor 2's range is represented by the blue rectangle and the purple area is where the two sensing ranges intersect.

Fig. 4 shows how a pair of ZX sensors together can measure position in three dimensions. The physical layout of the sensors is mapped to the Cartesian coordinate plane $\mathbf{R}^3$. The origin of the coordinate plane is depicted in Fig. 4. The actual sensing range includes twelve inches of depth in the z-direction.

One sensor is placed such that the center of the sensor is at (0,1.25,0) and the sensor is lying on the x-axis. The other sensor is placed at (1.25,0,0) and the sensor is lying on the y-axis. The first sensor can measure in the y-dimension and z-dimension in $\mathbf{R}^3$, and the second sensor can measure in the x-dimension and z-dimension. The z-value measured by both sensors should theoretically be the same or close to the same because both the sensors are on the plane z = 0. A position P on $\mathbf{R}^3$ in our network is measured as follows:

$$P = (x, y, z) \; = \; (x_{senors2}\,, x_{sensor1}\,, \hat{z}\,) \qquad (2)$$

In this equation, $x_{sensor2}$ is the x sensor value from the second sensor, and $x_{sensor1}$ is the x sensor value from the first sensor. $\hat{z}$ is equal to $z_{sensor1}$ if only sensor 1 detects an object, $z_{sensor2}$ if only sensor 2 detects an object, or the average of the two values, if both sensors detect an object.

As shown in Fig. 4, each sensor has a range of six inches in the sensor's x direction and a range of twelve inches in the sensor's z direction. Through heuristic measurement, each sensor's y direction range was measured to be at least four inches. Thus, the volume of space of each sensor's range is at least 288 cubic inches.

The purple rectangle in Fig. 4 depicts the region in which both sensors can detect an object. The area of this rectangle is 14.0625 square inches, and the volume of the intersecting region between the sensing ranges of the two sensors is 168.75 cubic inches. Because the horizontal cross section of this volume is limited, the motions which are solely included in the sensing range of both sensors is also limited.

C. Motions for Analysis

For each task, the subject was asked to begin with the hand flat, palm down, fingers together, and elbow at a right angle, keeping the forearm parallel to the ground. The subject was instructed to perform each of these tasks by moving only the arm and hand, keeping the rest of the torso and body steady. This ensured that the data collected was a reflection of arm and hand movement only.

For the first task, the subject was asked to place the center of the palm at the origin of the purple detection range on the sensor layout diagram in Fig. 4. Keeping the palm and parallel with the ground, the subject moved their hand up seven inches, paused for a second, and then brought the hand back down to the origin. The subject was then asked to repeat this up and down motion.

In the second task, the subject was asked to place the hand so that the center of the palm was directly over the left edge of the sensor detection range. Keeping the palm straight and the forearm parallel to the ground, the subject was instructed to move the hand horizontally left to right until the center of the palm was directly over the right edge of the detection range. The subject then waited one second and moved their hand back toward the left until the center of the palm was once again in its initial position. This left to right horizontal movement was repeated.

In the third task, the subject was asked to trace a circle shape with a radius of two inches in the clockwise direction. Placing the palm at the 9:00 position using the center of the palm as the tracing point, the subject was asked to trace a circle in the clockwise direction in the horizontal plane, making sure the center of the palm did not extend past the purple detection range.

In the fourth task, the subject was asked to place the center of the palm over the origin, making sure the tips of the fingers were facing the breadboard connecting the ZX Gesture Sensors. Moving only at the wrist, the subject pressed down beginning with the fingers, tilting the palm toward the ground. After pausing for one second, the subject brought the palm back to neutral and then completed the reverse motion, pushing the fingertips upward and consequently tilting the palm in the opposite direction. After pausing again in this position for one second, the subject was asked to return to neutral and repeat this set of motions.

In the final task, the subject was asked to hold a rectangular plate steady in one place for ten seconds. Initially, the subject held a dowel instead of a plate. It was decided during testing, however, that using a plate provided steadier data and reduced the margin of error due to its flat surface and increased surface area. The subject was asked to hold the plate against the palm with the fingers, placing the center of the plate over the origin of the detection range. The subject was then instructed to hold the plate steady, without moving, for ten seconds.

D. Sending Data to Raspberry Pi

After the sensor data was taken in by the Arduino, it was sent serially to the Raspberry Pi via USB connection and saved to a file [16]. The data from each was saved as a vector with x and z positions ranging from 0 to 240 units from each of the two sensors [17]. It was then converted to inches before graphical analysis. Within the Python script, one sensor was designated as x and z positioning and the other as y and z positioning. This file is later tied to for analysis and graphing.

E. Analyzing and Plotting Data

The data was received by the Raspberry Pi and saved as a file on the computer. However, before analyzing the data, it was mapped from the Arduino analog domain, which is recorded as a value from 0 to 240, to the real-world domain by converting to inches. In the z direction, there is a sensing range of zero to twelve inches. In both the x and y directions, there is a sensing range of -1.75 to 4.25 inches. The conversion from the analog to real world domain was done using division and subtraction, as shown in Equation 3, where x, y, and z are the positions in terms of Arduino analog data and x', y', and z' are positions in terms of inches.

$$x' = \frac{x}{40} - 1.75$$
$$y' = \frac{y}{40} - 1.75$$
$$z' = \frac{z}{20} \qquad (3)$$

Several functions were then written in Python to conduct mathematical calculations on the data including arithmetic mean, standard deviation, variance, and root mean square. A function was written to turn each analytic function into a moving version of itself. This moving function was applicable to four major mathematical functions: average, standard deviation, root mean square, and variance.

For every task, three-dimensional parametric graphs were created from the coordinates of hand position detected from the sensors. These graphs included x position versus y position versus z position, x position versus time, y position versus time, and z position versus time. For each task, moving average, moving standard deviation, moving root mean square, and moving variance of x position, y position, and z position were calculated and graphed. Each graph was created using raw experimental data as well as moving calculations to counteract outliers and produce a smoother, more comprehensible graph. The Python script that graphed the data utilized Matplotlib, a plotting library for Python. The graphs were generated using a three-dimensional plotting toolkit from the Matplotlib library that created three axes on which the positional data could be graphed.

E. Analysis of Deviation and Percent Error

After the raw data and the moving average for each task were graphed together, average deviation was calculated as a measure of hand shakiness and sensor inaccuracy. This was done by summing the difference between raw and theoretical data at each point and dividing by the total number of samples to get an average deviation value. Equation [1] is used to calculate these values. The results obtained from this equation were used to quantitatively define the extent to which the raw data varies relative to the theoretical data. These results were calculated with the understanding that some level of sensor inaccuracy contributed to the extent of deviation.

IV. Results

The proposed computer sensor network was tested on a subject without essential tremor in order to establish the functionality and accuracy of the network as well as provide a control for future data to be compared against. Positional data was collected from five tasks completed by the test subject, and multiple data features were generated and graphed for each task.

A. Motion 1: Up and Down Vertical Motion

The first task required the subject to move the hand up and down vertically while holding the palm flat. Specifically, a hand was moved from approximately (0,0,0) to (0,0,10) and back down to (0,0,0) in the three-dimensional coordinate plane of both ZX Gesture Sensors. This motion was completed twice. Fig. 5 shows the raw data, the data filtered using a moving average, and theoretical ideal data.

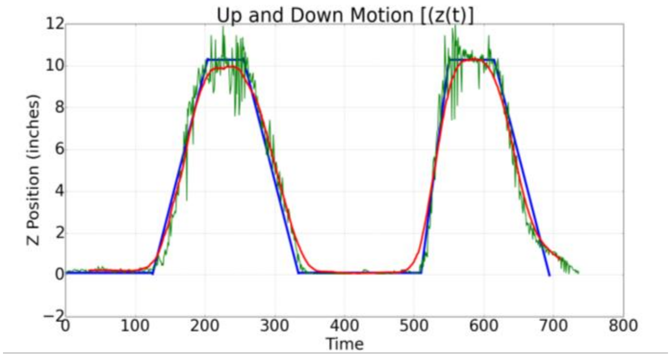

Fig. 5 The green line represents the raw data from the sensor and red represents the moving average of the raw data. This data was collected from a subject performing and up and down vertical motion. The blue line represents theoretical ideal data for this motion.

The theoretical line (depicted in blue) has a zero slope during the parts when the hand is held still, and it has a nonzero but constant slope when the hand is moving down or up. As shown in the graph, both the moving average and the raw data match the theoretical path closely, but the deviation is a bit greater when the hand is held at 10 inches above the sensors.

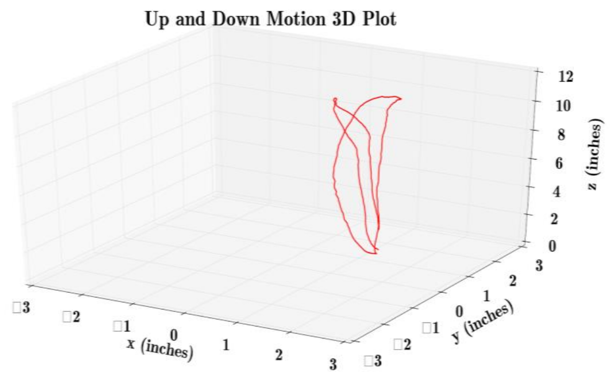

Fig. 6 A three-dimensional plot of data collected from a subject performing the up and down vertical motion. This graph depicts the motion path taken by the subject's hand in three-dimensional space.

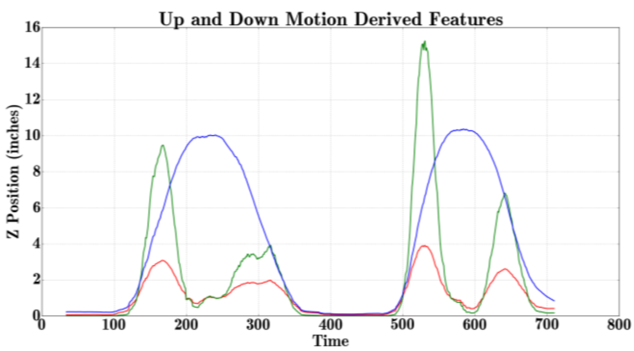

Fig. 7 The green line represents the moving variance, the blue line represents moving root mean square, and the red line represents moving standard deviation. This data was collected from the subject performing an up and down vertical hand motion.

In Fig. 6, a three-dimensional plot of the x-, y-, and z-axis data is shown for the up and down motion. The motion depicted in the plot is not perfectly up and down, which is expected due to sensor error and hand shakiness. Overall, however, the motion depicted is going from a height of zero inches to a height of ten inches. Fig. 7 shows the moving standard deviation, root mean square, and variance of the data. All of these serve as metrics of analysis to visualize characteristics of the data and could be used to compare to other data sets or baseline results.

B. Motion 2: Left to Right Horizontal Motion

The second task required the subject to move the hand in a left to right path twice. This data is graphed in Fig. 8. In the figure, the x-position is compared against time. The increasing regions of the graph represent the hand moving to the right, increasing along the x-axis. The plateaus represent the pause after each left or right motion, and the decreasing sections represent the hand moving to the left, decreasing along the x-axis. The raw data is graphed alongside moving average, which provides a smoother visual as well as a theoretical path drawn in blue.

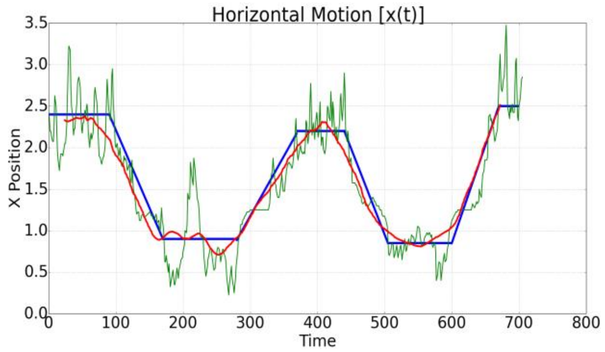

Fig. 8 The green line represents raw data and the red line represents moving average of left-to-right horizontal motion. The blue line represents theoretical ideal data for the horizontal motion. Each plateau signifies a pause in the motion at the far left and far right of the motion path.

Likewise, from the up and down motion, the theoretical line has a zero slope during the parts when the hand is held still, and it has a nonzero but constant slope when the hand is moving horizontally. The moving average and the raw data seem to match the theoretical path less for the horizontal motion along the x-axis than for the up and down motion.

C. Motion 3: Circular Motion

The third task required the subject to move their hand in a clockwise circular motion. Fig. 9 and Fig. 10 show the results from data collection. Again, raw data was graphed alongside moving average, which provides a smoother graph by helping to reduce the impact of extreme outliers on the graph's shape. The theoretical paths are sinusoidal waves [18] because the x-component and y-component of a circular motion over time oscillate between their maximum and minimum points in a sinusoidal fashion. The actual movement was not in a constant speed, so the actual motion does not fit the theoretical sine wave completely. The motion would be modeled more closely by sine waves if the speed and rotational acceleration were constant throughout the circular motion.

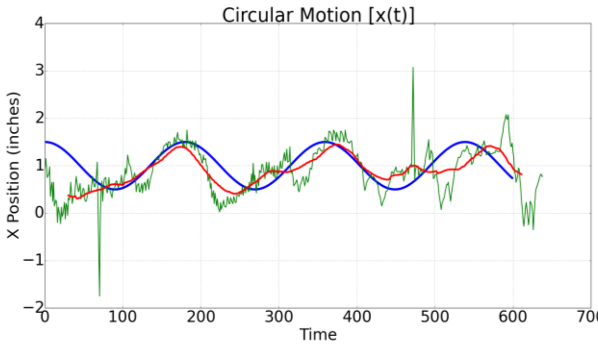

Fig. 9 The green line represents the raw data and the red line the moving average of the x-value versus time data for the circular motion. The blue line, a sinusoidal function, represents the theoretical ideal data for this motion.

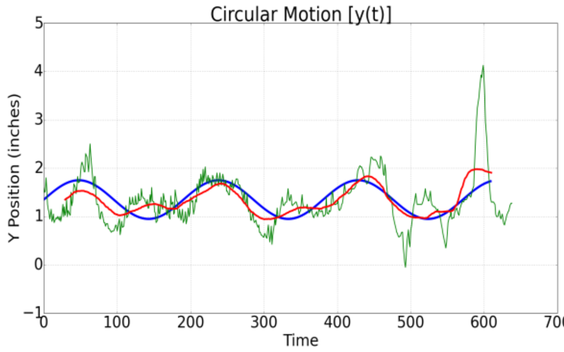

Fig. 10 The green line represents the raw data and the red line represents the moving average for the y-value versus time of the circular motion. The blue line, a sinusoidal function, represents the theoretical ideal data for this motion.

D. Motion 4: Tilting of Hand

The fourth task required the subject to bend the wrist, tilting the palm up and down. The raw data and moving average are depicted in Fig. 11. The computer sensor network is much less reliable for this motion, largely due to the irregular nature of the hand position during the motion. Therefore, there are significant fluctuations.

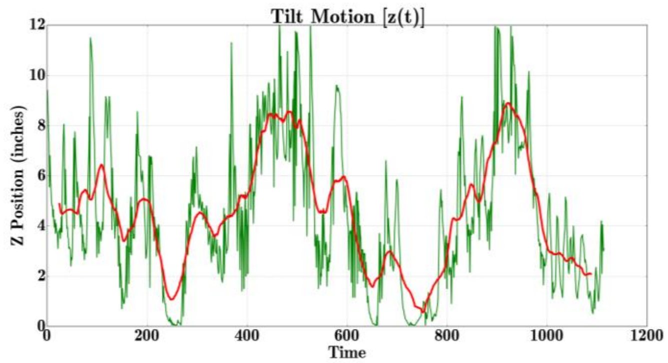

Fig. 11 The green line represents the raw data and the red line represents the moving average of the data collected from the hand-tilting motion.

E. *Motion 5: Steady Hold*

The final task required the test subject to hold the hand steady in one position for a given amount of time. Fig. 12 shows the graph of this data, including raw data and moving average as well as theoretical data. The purpose of this task was to illustrate not only the minute shaking that occurs in the normal hand, but also to illustrate the way the computer sensor network could be successful in collecting and analyzing data from a variety of different types of motions. The theoretical paths are straight lines because for a hand held steady, the coordinate values do not change. However, there is a slight fluctuation around the expected position of about (0,1,8).

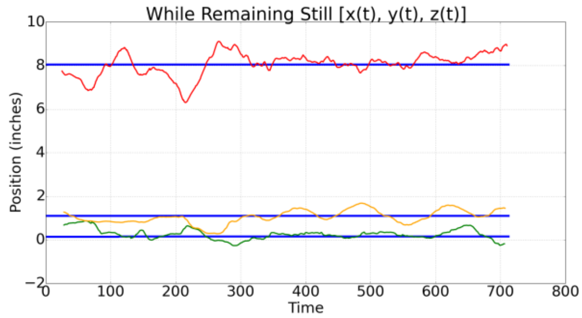

Fig. 12 The red line represents the moving average of the z-axis values, the yellow line represents the moving average of the y-axis values, and the green line represents the moving average of the x-axis value. This data was obtained from holding a hand steady for a certain amount of time. The blue, zero slope lines represent the ideal data for this motion.

In this study, the deviation from the ideal path was calculated as an average deviation. Table I lists the average deviation values for each task. The deviation was calculated by using Equation (1).

TABLE I

AVERAGE DEVIATION OF EACH TASK

Task	Calculated Average Deviation (inches)
Up and down motion - Z values	0.168
Holding still motion - X values	0.155
Holding still motion - Y values	0.202
Holding still motion - Z values	0.239
Circular motion - X values	0.193
Circular motion - Y values	0.144
Horizontal motion - X values	0.114

V. DISCUSSION OF RESULTS

For all of the motions, the average deviations were small, ranging between 0.114 inches for the x-position of the horizontal motion task and 0.239 inches for the z-position while holding steady. Although the deviations were generally larger when measuring the z-position over time, the relative error is less because the z-range is greater.

The three-dimensional plot was expected to look much closer to the actual path of motion performed by the subject. However, due to the natural shakiness of the human hand and the sensors' lack of precision, the graph did not follow the strict motion expected theoretically. While some general shape can still be seen the three-dimensional graph shown in Fig. 7, it provides a less valuable analysis of the data than two dimensional graphs of individual x-, y-, or z-axis position plotted against time.

In most cases, the two-dimensional data aligned with the expected results with minimal deviation. For instance, as can be seen in Fig. 11, when the y-position of the circular motion was plotted against time, it generated an approximately sinusoidal wave as expected from circular motion. However, the speed was not constant, so the frequency of the sinusoidal wave from the actual motion was not constant. This caused deviations from the theoretical path, which is constant frequency sinusoidal wave.

For some of the motions, using a single sensor provided more valuable data. The two sensors each offered a limited individual range, and limitations are exaggerated when dealing with intersecting ranges. Using a single sensor for two-dimensional motion, such as the task requiring the subject to move in a simple left to right line, provided better data for graphical analysis.

While the computer sensor network is capable of collecting and analyzing positional data, the measurements do have some uncertainty. The primary reason for slight inconsistencies in measurements is the limited capabilities of the sensors. The ZX sensors have a limited range, making motions which take up a lot of space difficult. The usage of two sensors makes the union of the sensing ranges greater, but the intersecting region of the sensing ranges is limited. Therefore, there are intervals of time when only one or no sensors may detect an object. As a result, there are gaps in the data. These are accounted in the analysis, but they do limit the capabilities of position detection. In addition, the sensors do not measure distances very precisely. This is evident as even when an object is held still, the amount of deviation is noticeable. This can be partially accounted for by the natural shakiness in a person's hand. A reference was provided for the subject in the form of a two dimensional grid placed below the sensors, as depicted in Fig. 4. However, this is not enough to eliminate all error due to inexact path following. Therefore, our network's error is the combined error of the sensor and the person who is using it.

Even with the multiple sources of error, the network still has significant potential to measure and quantify hand tremors. Let P_{M1} , P_{M2} be the measured positions for two persons using the network for a given position and time. Also let P_T represent the theoretical position of an object for the given position and given time.

$$P_{M1} = P_T \pm Error_{sensors} \pm Error_{Person\ 1}$$

$$P_{M2} = P_T \pm Error_{sensors} \pm Error_{Person\ 2} \tag{4}$$

If the error due to the sensors at that position and time are the same, then subtracting the equations (and adding absolute values) yields:

$$|P_{M1} - P_{M2}| = |Error_{Person\ 1} - Error_{Person\ 2}|. \tag{5}$$

Therefore, the positions measured by the sensors by the two people can be used to relate the errors due to the hands of the two people. The further apart the measured positions of the two people are, then the further apart the errors of the two people are. Equation (5) can be used to compare the hand position of a person with essential tremor and a person without essential tremor. The differences between the positions for a person with essential tremor and a person without essential tremor can be used to measure the amount of error that the hand tremors cause. In the future, this may be used to diagnose, classify, and/or categorize essential tremor with the use of baseline tests similar to those used in the analysis of this network.

However, the differences in position are reflective of the differences in intensity of hand tremors only if the sensor errors are small and relatively equal. Therefore, the most significant way to increase the reliability of the network is by increasing the quality of the sensors. In addition, it must be ensured that the tasks are replicated exactly to isolate error to that from the hand tremors rather than from differences in performance of the tasks.

VI. Conclusion

This paper proposes a computer sensor network capable of collecting and analyzing data in regards to human hand tremors. This is achieved through the connection of two ZX Gesture Sensors, an I2C multiplexer, an Arduino Uno, and a Raspberry Pi computer using the computer programming languages C and Python. Two and three-dimensional positional data was collected, graphed, and compared to the derived data features, such as moving average, standard deviation, variance, and root mean square. The motions were compared against theoretical ideal paths to calculate an average deviation. The graphical analysis reveals the natural shakiness in the hands and arms of those without essential tremor. This can serve as a baseline of comparison between those with and without essential tremor. In the future, this computer sensor network can be improved to analyze and classify the movements of essential tremor patients in order to better understand and potentially treat this disorder.

For future work with this network, using higher quality sensors and developing a better way to demonstrate the paths the subject needs to follow, perhaps by using augmented reality technology to superimpose the path above the sensor, would eliminate some of the deviations unrelated to hand tremor. Additionally, it would be of value to measure and compile motion data from a large sample size of subjects without essential tremor to create control data. This control data could be extraordinarily helpful to compare against data from essential tremor patients to aid in diagnosis and a better understanding of the disorder.

The network is not limited to the world of essential tremor, and could be applied to areas such as athletics or sports medicine that require detailed analysis of even the slightest physical movements. This network is unique in its ability to comprehensively collect and analyze data in an objective way, and could revolutionize the way tremors are assessed by medical professionals and contribute to the expanding field of computer-based motion analysis.

VII. Acknowledgements

The authors gratefully acknowledge project mentor Alex Weiner as well as residential teaching assistant and research coordinator Jaimie Swartz, who assisted and directed them in this project from conception to realization.

The authors would also like to sincerely thank Dean Ilene Rosen, New Jersey Governor's School of Engineering and Technology Director and Dean Jean Patrick Antoine, New Jersey Governor's School of Engineering and Technology Associate Director, without whom this study would not be possible.

Additional thanks are extended to the many generous sponsors of the New Jersey Governor's School of Engineering and Technology including Rutgers University, Rutgers School of Engineering, the state of New Jersey, Lockheed Martin, Silverline Windows, and the New Jersey Governor's School of Engineering and Technology Alumni.

References

[1] "The Brain and Essential Tremor," *WebMD*. [Online]. Available: http://www.webmd.com/brain/essential-tremor-basics#1. [Accessed: 20-Jul-2017].

[2] Mayo Clinic Staff, "Essential tremor," *Mayo Clinic*, 20-Jan-2016. [Online]. Available: http://www.mayoclinic.org/diseases-conditions/essential-tremor/symptoms-causes/dxc-20177829. [Accessed: 20-Jul-2017].

[3] J. P. Varkey, D. Pompili, and T. A. Walls, “Erratum to: Human motion recognition using a wireless sensor-based wearable system,”*Personal and Ubiquitous Computing*, vol. 16, no. 7, pp. 911–911, 2012.
[4] “Essential tremor - Genetics Home Reference,”*U.S. National Library of Medicine*, 18-Jul-2017. [Online]. Available: https://ghr.nlm.nih.gov/condition/essential-tremor. [Accessed: 20-Jul-2017].
[5] R. J. Elble, “The Essential Tremor Rating Assessment Scale,”*The Essential Tremor Rating Assessment Scale | JNEUROLOGY*, 21-Jul-2016. [Online]. Available: http://www.jneurology.com/articles/the-essential-tremor-rating-assessment-scale.html. [Accessed: 20-Jul-2017].
[6] S. Hymel, “ZX Distance and Gesture Sensor Hookup Guide,”*Learn at SparkFun Electronics*. [Online]. Available: https://learn.sparkfun.com/tutorials/zx-distance-and-gesture-sensor-hookup-guide#arduino-gesture-example. [Accessed: 16-Jul-2017].
[7] D. Purdie, “RPi and Arduino based data logger,”*OpenLabTools*, 2013. [Online]. Available: http://openlabtools.eng.cam.ac.uk/Resources/Datalog/RPi_Arduino/. [Accessed: 20-Jul-2017].
[8] A. Sentosa, “Raspberry Pi - Arduino Serial Communication,”*Instructables.com*, 12-May-2016. [Online]. Available: http://www.instructables.com/id/Raspberry-Pi-Arduino-Serial-Communication/. [Accessed: 20-Jul-2017].
[9] “Teach, Learn, and Make with Raspberry Pi,”*Raspberry Pi*. [Online]. Available: https://www.raspberrypi.org/. [Accessed: 19-Jul-2017]
[10] “Serial communication,” *SparkFun*. [Online]. Available: https://learn.sparkfun.com/tutorials/serial-communication. [Accessed: 19-Jul-2017].
[11] “I2C,” *SparkFun*. [Online]. Available: https://learn.sparkfun.com/tutorials/i2c. [Accessed: 19-Jul-2017].
[12] “Software I2C library (master mode only),”*Arduino Playground - SoftwareI2CLibrary*. [Online]. Available: https://playground.arduino.cc/Main/SoftwareI2CLibrary. [Accessed: 20-Jul-2017].
[13] “Adafruit TCA9548A 1-to-8 I2C Multiplexer Breakout,”*Wiring & Test | Adafruit TCA9548A 1-to-8 I2C Multiplexer Breakout | Adafruit Learning System*, 28-Nov-2016. [Online]. Available: https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout/wiring-and-test. [Accessed: 19-Jul-2017].
[14] “Port Registers,”*Arduino - PortManipulation*. [Online]. Available: https://www.arduino.cc/en/Reference/PortManipulation. [Accessed: 20-Jul-2017].
[15] “Adafruit TCA9548A 1-to-8 I2C Multiplexer Breakout,”*Overview | Adafruit TCA9548A 1-to-8 I2C Multiplexer Breakout | Adafruit Learning System*, 11-Sep-2015. [Online]. Available: https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout/overview. [Accessed: 19-Jul-2017].
[16] “Grab serial - program to read a serial port and send the data to stdout,”*Grab serial - program to read a serial port and send the data to stdout*. [Online]. Available: https://raw.githubusercontent.com/tbird20d/grabserial/master/grabserial. [Accessed: 19-Jul-2017].
[17] S. Hymel, “ShawnHymel/HoverPong,”*GitHub*, 08-Jul-2015. [Online]. Available: https://github.com/ShawnHymel/HoverPong/blob/master/HoverPong/HoverPong.ino. [Accessed: 19-Jul-2017].
[18] “Unit Circle Showing Sine Graph,”*Desmos Graphing Calculator*. [Online]. Available: https://www.desmos.com/calculator/kxfekf0kgb. [Accessed: 19-Jul-2017].
[19] Elble, R., Comella, C., Fahn, S., Hallett, M., Jankovic, J., Juncos, J. L., LeWitt, P., Lyons, K., Ondo, W., Pahwa, R., Sethi, K., Stover, N., Tarsy, D., Testa, C., Tintner, R., Watts, R. and Zesiewicz, T. “Reliability of a New Scale for Essential Tremor,” in *Movement Disorder Society*, 2014.