5-4-2021

Transforming the Chinese Pole Circus Apparatus into an Interactive Musical Instrument

Fosse Lin-Bianco
flinbian@lion.lmu.edu

Hossein Asghari
Loyola Marymount University, mohammadhossein.asghari@lmu.edu

Evan Mitchell
Loyola Marymount University, emitch12@lion.lmu.edu

Transforming the Chinese Pole Circus Apparatus into an Interactive Musical Instrument

A thesis submitted in partial satisfaction

of the requirements of the University Honors Program

of Loyola Marymount University

by

Fosse Lin-Bianco

May 4, 2021

Transforming the Chinese Pole Circus Apparatus into an Interactive Musical Instrument

Fosse Lin-Bianco and Evan Mitchell
Department of Electrical and Computer Engineering
Loyola Marymount University
April 30, 2021

Abstract

The objective of this project is to create a modified version of the Chinese pole circus apparatus in order to artistically musicalize and visualize a circus performer's movement in real time. Wireless, wearable inertial measurement units (IMUs) allow for tracking the position of the performer's hands and feet. The vertical height of the performer is then used to play a corresponding pitch on a musical scale, while the position with respect to the other two dimensions is used to produce a bird's-eye-view visualization. Radio-frequency identification (RFID) tags added to the pole improve the accuracy of the IMU position tracking by providing anchor points with which to recalibrate an IMU's position in space. This works to reduce the effect of drift, the result of small inaccuracies in the acceleration data collected by the IMUs which are compounded when integrating over time to determine velocity and position. This project allows an audience to experience the movement of a circus performer from a new perspective. In addition to the choreography, the audience is left with a unique musical composition and visual art pieces they can remember. More importantly, the audience can experience the movement from a perspective that cannot be experienced without the use of this technology.

Table of Contents

I. Introduction

In the past decade, many industries have been touched by the digital world, including the entertainment industry. While the circus arts and dance industries have incorporated technology in the form of projections and moving stage platforms, for example, these technologies many times distract from the focus of the performers on stage and do not add significant value to their movement. However, we believe it is possible to use technology to aid in the artistic performance of a circus or dance piece instead of distracting from it. One way to create this effect is to reveal a new perspective of a performer's movement that could not have been previously experienced without the technological aid.

In 2014, Spanish artist and product designer Lesia Trubat created a video showcasing her work called "E-TRACES. Memories of Dance." Trubat sewed a number of different sensors into ballet shoes and displayed the movement of a ballet dancer as brush strokes on the screen. When the dancer would perform small turns, or chaîné turns, smooth, black brush strokes would overlay the video, forming the same circles as the turns [1]. Fig. 1 shows an image from this video demonstrating her work [2].

Fig. 1. Screenshot from "E-TRACES. Memories of Dance" by Lesia Trubat.

Team member Fosse Lin-Bianco was intrigued by Trubat's work because she used technology to create a novel way to view dance. The digital aspect of the work was not merely an accessory for entertainment, but a necessary tool to view this new perspective. With her work in mind, Fosse participated in a Research Fellowship with the University Honors Program during the summer of 2020 to create his own version of this project, but with a focus on circus performers and the movements of acrobatics. During this research, Fosse worked under the mentorship of Dr. Barbara Marino.

In this project, entitled Acrobat Acceleration, an accelerometer was sewn into an acrobatics shoe to track the acceleration of a circus performer as they performed different acrobatics skills. This data was then used to generate digital art pieces, with the created art responding to the performer's movement. Fig. 2 is an image of the final built acrobatics shoe, and Fig. 3 demonstrates a sample art piece that was generated based on an acrobatic performance of a forward roll. In Fig. 3, the placement of the circles on the canvas are random. However, each circle represents one data point, with different colors for x-, y-, and z-axis acceleration. The diameter of each circle corresponds to the magnitude of acceleration. When the test begins, small circles begin to appear on the screen and when the performer begins to accelerate, larger circles begin to appear. In the end, we are left with an art piece that encapsulates the acrobatic skill, and the audience can leave with a memory to remember the skill by.

Fig. 2. Image of acrobatic shoe with acceleration tracking system [3]

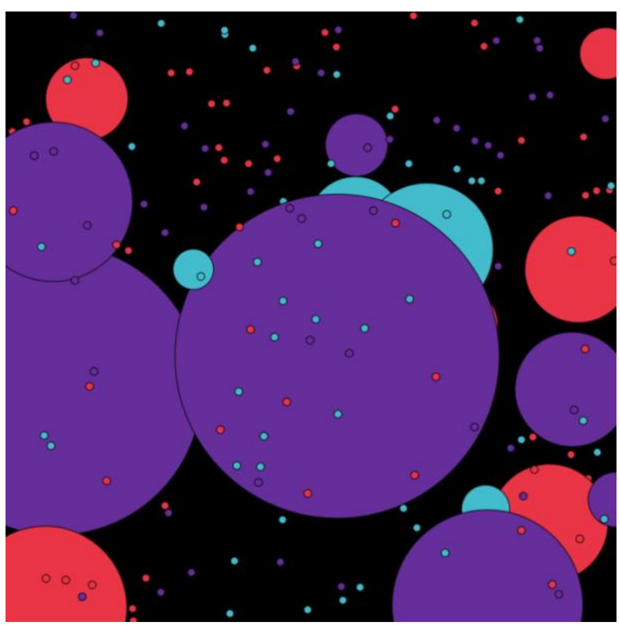

Fig. 3. Art piece generated from the acceleration data of a forward roll acrobatic skill

After Fosse's work over the summer, he teamed up with Evan Mitchell to continue work in this field of research. Evan has a strong music background and has spent many years playing the piano, along with a collection of other instruments. We wanted to come up with a technical engineering project that would encompass both Fosse's interest in circus art and Evan's interest in music. In the end, we decided on a project we call the Piano Pole.

The Piano Pole is a modified version of the Chinese pole circus apparatus, a vertical pole on which performers climb, that creates music and a visualization based on a circus performer's movement on the pole. The vertical position of the performer's feet and hands along the pole produce different musical notes that correspond to the keys of a piano. In this way, the performer is able to create a piece of music based on the positions of their hands and feet on the Chinese pole. In addition, a visualization would be created based on the positions of the hands and feet in the other two dimensions, resulting in an art piece that represents a bird's eye view of the performer's movement around the pole.

The Chinese pole apparatus is shown in Fig. 4 and a concept image of the Piano Pole system is shown in Fig. 5.

Fig. 4. Image of the Chinese pole circus apparatus [4]

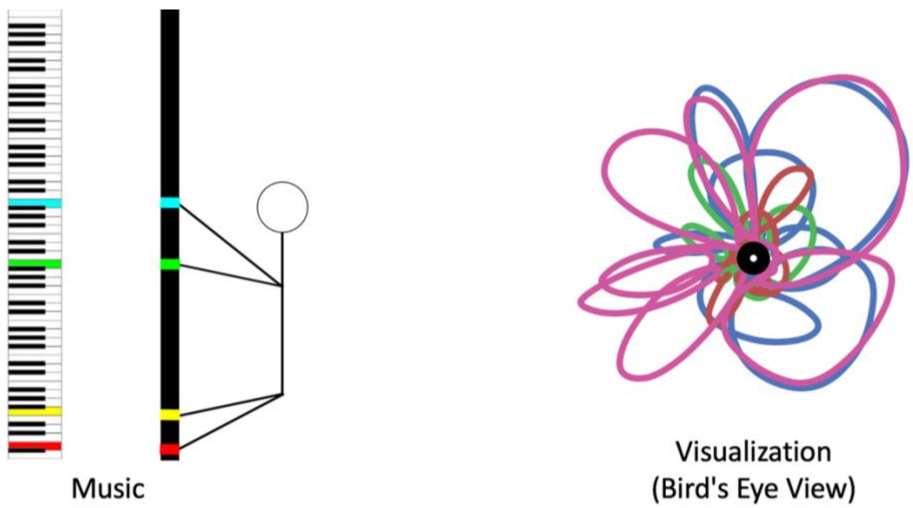

Fig. 5. Concept image of the Piano Pole system

The Piano Pole system relies heavily on tracking the position of the human body. Position tracking, also referred to as dead reckoning, is largely a saturated field, since much research has been done on the topic. However, accurate position tracking systems are typically optimized for basic human movements such as walking, jogging, or running, while the research on more complex movements such as dancing or acrobatics is less developed [5] [6] [7] [8]. Some work has been done to track dance movements; this type of position tracking is most commonly accomplished using an external camera or depth sensor, such as the Microsoft Kinect sensor.

For the Piano Pole project, we narrowed down our position tracking options on the Chinese pole apparatus to an external RGB camera, a Kinect sensor, and an inertial measurement unit (IMU) sensor. After a thorough analysis, it was determined that the IMU sensors would be the best method to track the human movement for a few reasons. Firstly, while the RGB camera can detect the full body of a human, it does not track depth very well, and three-dimensional tracking would be inaccurate or require multiple cameras. Secondly, the Kinect sensor has very accurate depth tracking and can track up to 25 joints, but the range of the Kinect sensor is limited. The Kinect can only measure 2.9m in the vertical direction and 2.8m in the horizontal direction, while the Chinese pole apparatus stands at 5.5m tall. To detect this full range, two Kinect sensors would need to be used, with one of the sensors suspended 3m off the ground. This setup was not only deemed impractical, but it would also cause the audience's view of the performer to be obstructed. The IMU sensors, then, provide the best solution because they do not obstruct the audience's view of the performer, since the sensors would be attached to the performer's hands and feet. Additionally, an IMU sensor can track 3-axis linear and rotational acceleration, making it capable of tracking position in three dimensions, and it can detect position on the full 5.5m range of the Chinese pole.

The BNO085 IMUs being utilized for this project contain three different sensors: an accelerometer, a gyroscope, and a magnetometer. The accelerometer works by detecting the acceleration, or the change in velocity, of the sensor in the x, y, and z directions. The sensor operates similarly to the image depicted in Fig. 6. One can imagine a floating ball that rests in the middle of the sensor. As the sensor accelerates in one direction, the floating ball is pressed up against the wall of the opposite direction and induces an electrical current. The strength of this current then corresponds to the magnitude of acceleration.

The gyroscope operates by detecting the angular acceleration caused by rotation of the sensor. An image of a gyroscope is shown in Fig. 7. Finally, the magnetometer operates by detecting the sensor's orientation with respect to the Earth's magnetic poles and is shown in Fig. 8.

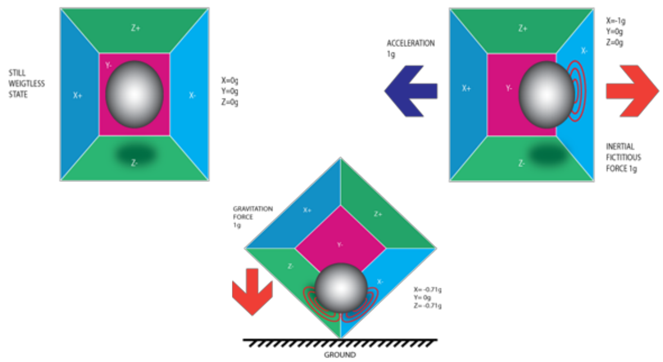

Fig. 6. Illustration of how an accelerometer operates [9]

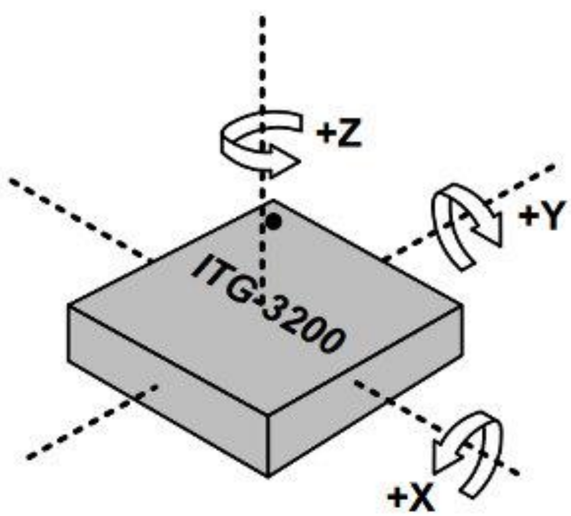

Fig. 7. Illustration of a gyroscope [10]

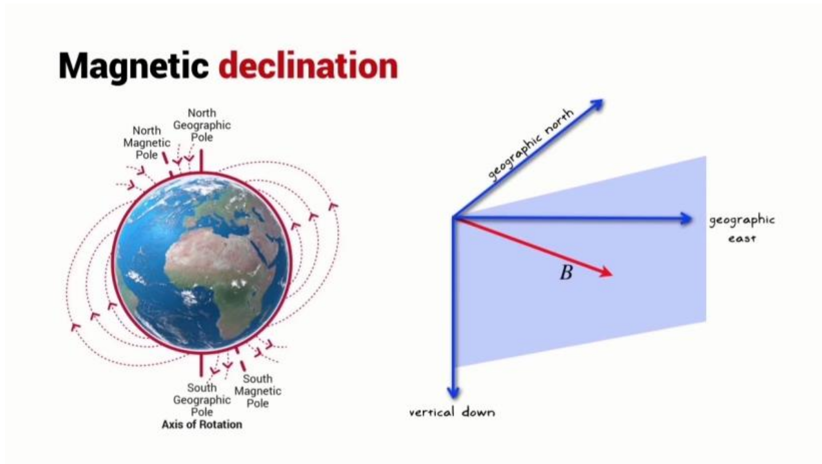

Fig. 8. Illustration of how a magnetometer operates [11]

Utilizing the IMU sensors for position tracking, however, presents many challenges for the accuracy of the system. Since the IMU sensor can only detect linear and angular acceleration and magnetic fields, an algorithm is needed to translate these values into meaningful position data. In short, a fusion algorithm is first applied to determine absolute linear acceleration values, removing acceleration due to gravity, and then those acceleration values are integrated to obtain velocity and integrated again to obtain position, as shown in (1), where $\tilde{x}(t)$ represents the position, $\tilde{v}(t)$ represents the velocity, and $\tilde{a}(t)$ represents the acceleration.

$$\iint \tilde{a}(t)\,dt = \int \tilde{v}(t)\,dt = \tilde{x}(t) \quad (1)$$

During the integration, however, small errors and gaps in the acceleration data are amplified, resulting in much larger errors in the final position data. In our tests, we found that this error was amplified by as much as two orders of magnitude from acceleration to position. The graph in Fig. 9 demonstrates this error propagation. In this test, the IMU is at rest, but the calculated position data shows that the IMU moved hundreds of meters.

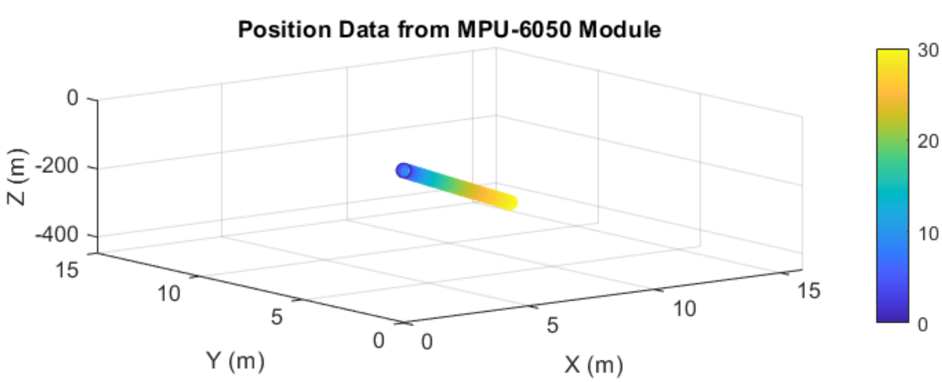

Fig. 9. 3D graph of position from IMU test

This is a problem known as drift. When the errors in the position propagate over time, the calculated position can "drift" away from the true position of the sensor. This drift error is most prominent when there are sharp changes in speed or direction, causing sudden large acceleration values for only a short time. Fig. 10 presents a diagram demonstrating this drift error, where the green line represents the IMU's calculated position and the red line represents the IMU's actual position.

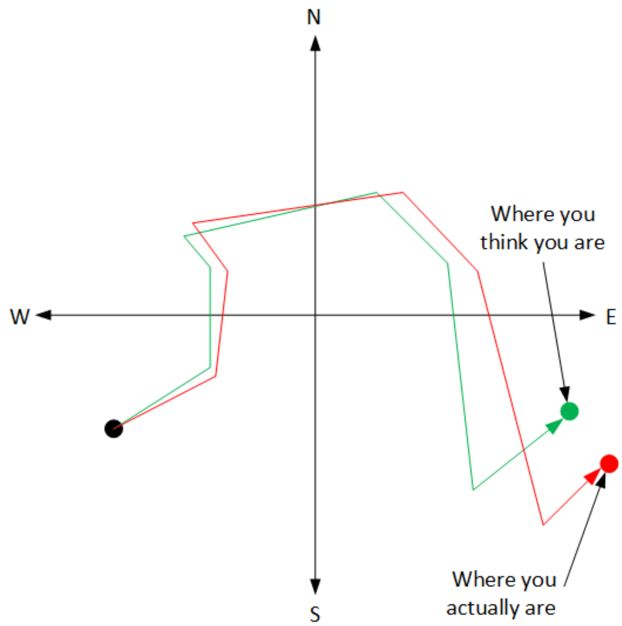

Fig. 10. Graph that demonstrates drift error [12]

Since a typical circus act is around three minutes in length, the IMU sensors would need to be accurate, to a certain degree, for the full duration of the act. Therefore, to combat drift, we developed the idea of attaching radio-frequency identification (RFID) tags to the Chinese pole, where the exact position of each tag is known. An RFID reader can then be attached along with the IMUs to each of the performer's hands and feet so that each time one of the performer's hands or feet approach an RFID tag, the corresponding IMU's position is able to be recalibrated to the tag's known position. Considering all these factors for the Piano Pole system, the following project objective was developed.

II. Project Objective

A. Background Information

The objective of this project was to create a modified version of the Chinese pole circus apparatus, a vertical pole on which performers climb, in order to artistically musicalize and visualize a circus performer's movement in real time. Wireless, wearable inertial measurement units allow for tracking the position of the performer's hands and feet. The vertical height of the performer is used to play a corresponding pitch on a musical scale, while the position with respect to the other two dimensions is used to produce a visualization. Radio-frequency identification tags were added to the pole to improve the accuracy of the IMU position tracking by providing anchor points with which to recalibrate an IMU's position in space. This helps to reduce the effect of drift, the result of small inaccuracies in the acceleration data collected by the IMUs, which are compounded when integrating over time to determine velocity and position. This project allows an audience to experience the movement of a circus performer from a new perspective. In addition to the choreography, the audience is left with a unique musical composition and visual art pieces they can remember. More importantly, the audience can experience the movement from a perspective that cannot be experienced without the use of this technology.

With these objectives in mind, two key problem statements arise. First, this project aims to explore how to provide the audience with musical and visual perspectives of a circus performer's movement. Secondly, in order to realize our artistic vision that accurately reflects a performer’s movement, this project attempts to reduce drift in IMU position tracking to provide increased accuracy.

B. Customer Requirements

The following five customer requirements were then determined to accurately match the goal of the project:

1. The system should produce a musical composition with a range of two octaves of piano notes that corresponds to the performer's live vertical position on the Chinese Pole.
2. The system should produce a visualization based on a bird's eye view of the performer's live position on the Chinese pole.
3. The system hardware should be able to be easily attached to and removed from the Chinese Pole.
4. The system should be safe for a performer to use (given proper training) and should not add to the risk of the Chinese pole.
5. The system should have low cost to allow for widespread use.

III. Proposed Solution

A. Trades Leading to Proposed Solution

To arrive at our solution of the Piano Pole system, we first brainstormed about what to include in the system. The general ideas that we came up with are shown in the concept fan in Fig. 11. The ideas of creating music and art pieces that respond to the performer’s movement appear in the concept fan, along with the results of brainstorming what kind of movement data should be collected and utilized in the music and art. We considered using acceleration, velocity, and position, but we ultimately decided to utilize the performer’s position because we believed this data would be easier for a general audience to understand.

After determining that it would be best to use position tracking to develop the music and visualizations, we brainstormed what the best methods would be to achieve these goals. These ideas are shown in Fig. 12 with the concept table. From the concept table, we can see the three different methods of position tracking being considered: namely the Microsoft Kinect, IMU, and RBG camera. We also included different ideas that we discussed for how to musicalize and visualize the movement. We considered the idea that the performer's vertical position on the pole could be mapped to a pitch on a continuous spectrum rather than discrete piano notes. Additionally, we discussed creating a full three-dimensional visualization of the performer's position around the pole. In the end, we decided to utilize a two-dimensional bird's eye visualization because we believed this would be simpler for a general audience to understand and it would provide a unique perspective, since an audience cannot easily view a performance from above.

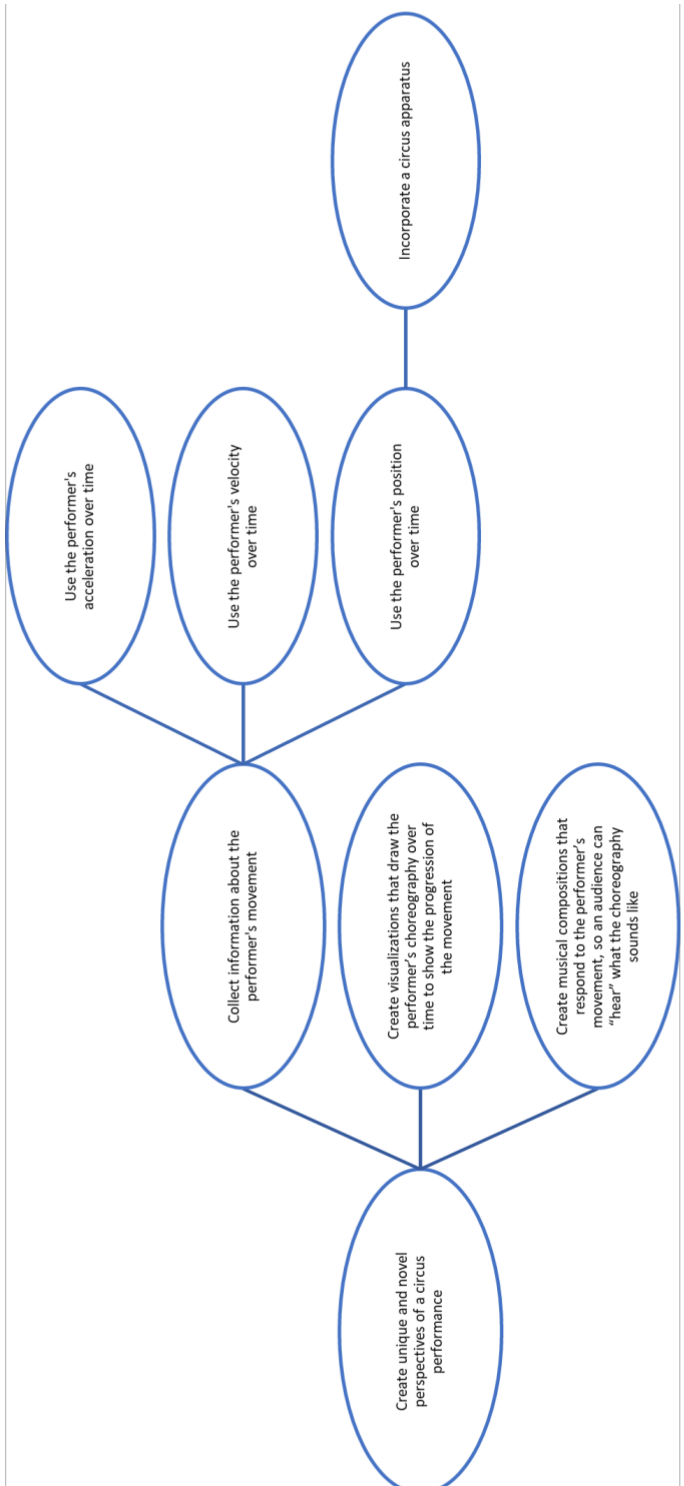

Fig. 11. Concept fan diagram for brainstormed ideas for the Piano Pole system.

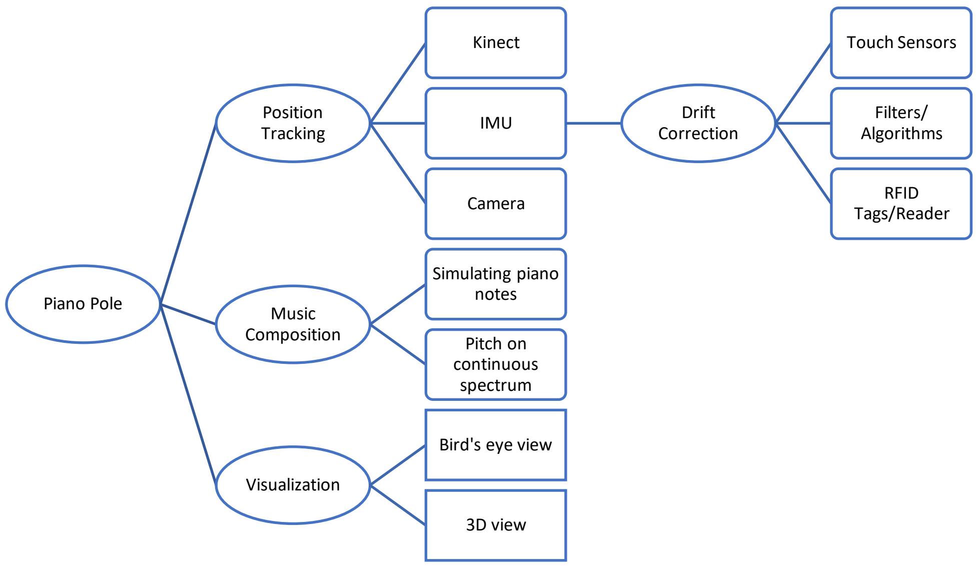

Fig. 12. Concept table diagram for brainstormed methods to achieve the Piano Pole system

The Piano Pole system is mainly comprised of a position tracking system for the human body. With the position data, it is possible to musicalize and visualize the movement of a circus performer. Since tracking position is the main function of the system, it was imperative that the correct type of method was used for this specific scenario. We determined that there were five distinct features that the position tracking system needed to possess including: high accuracy, sufficient range, low cost, concealment from the audience, and low interference with the performer. High accuracy is needed to determine the position of the performer's hands and feet on the pole and to produce the correct piano notes and visualizations. Low cost is needed to comply with the limits of departmental funds. Concealment from the audience involves disrupting the audience's viewing experience as little as possible with the system's hardware components. Finally, a low level of interference with the performer is needed so the performer does not feel unsafe using the system.

The weights of these specifications relative to each other are given in the Pairwise Comparison Matrix in Table I. The rightmost column of Table I details the resulting weights for each specification. It was determined that concealment from the audience was the most important factor, with a weight of 0.32 out of 1.0. Other crucial factors to note were the cost and range, which hold weights of 0.24 and 0.20, respectively. These weights were then used in the Decision Matrix in Table II to determine which position tracking method would be optimal for the Piano Pole system. In this matrix, the Kinect performs the best in accuracy and ties with the RGB Camera in low interference with the performer. The IMU had a much lower accuracy score, but it was strong in the categories of range, cost, and ability to be concealed. Combining the scores in each category with the predetermined weights of each category, the IMU sensor was determined to be the best position tracking method for the Piano Pole system due to its high range, lost cost, and high concealment from the audience in comparison to the Microsoft Kinect and the RGB camera. This result can be shown in the Decision Matrix with the IMU receiving the highest final score of 0.46

out of 1.0, in comparison to the Kinetic and RGB Camera which received scores of 0.25 and 0.29, respectively.

Table I. Pairwise Comparison Matrix

	Accuracy	Range	Cost	Conceal	Interference	Geometric Mean	Weights
Accuracy	1	1	0.5	0.25	3	0.82	0.15
Range	1	1	0.5	1	3	1.08	0.20
Cost	2	2	1	0.5	2	1.32	0.24
Conceal	4	1	2	1	2	1.74	0.32
Interference	0.3	0.3	0.5	0.5	1	0.49	0.09

Table II. Decision Matrix

		Kinect	IMU	Camera
Accuracy	0.15	0.54	0.22	0.24
Range	0.20	0.29	0.45	0.26
Cost	0.24	0.17	0.49	0.34
Ability to be concealed	0.32	0.11	0.62	0.27
Low interference with performer	0.09	0.37	0.26	0.37
Score		0.25	**0.46**	0.29

B. Technical Requirements

Utilizing the customer requirements stated in the Project Objective section, corresponding engineering requirements were developed along with justifications for each of these requirements. These requirements and justifications are shown in the Requirements Analysis Framework in Table III.

Finally, the House of Quality, given in Table IV, was developed to describe the correlation between the customer and engineering requirements in addition to the correlation between each individual engineering requirement. Looking at the relationship between the customer requirements and the engineering requirements, a strong correlation is notated with a solid black dot (•) in the center grid of the table. Additionally, the direction of improvement, shown right above the engineering requirements on the table, describes the goal of each requirement. For example, the down arrow (▼) for cost means the system aims to reduce the cost whereas an up arrow (▲) for position accuracy means the system aims to increases accuracy. When observing the requirements for the Piano Pole, there is a strong correlation between the latency and position accuracy (engineering requirements) with the musical composition and visualizations (customer requirements). This makes sense as low latency and higher position accuracy produce music and art pieces that more closely align with the performer's movement. Another crucial correlation to note is the one between protrusion and safety. There is a strong correlation for this relationship because less protrusion greatly increases the safety of the Piano Pole system.

Looking at the relationship between the individual engineering requirements, a positive correlation (+) denotes that the direction of improvement for one requirement improves the direction for the other requirement whereas a negative correlation (-) reduces the direction of improvement for the other requirement. When comparing the Piano Pole requirements, the most significant correlations are the negative correlations between latency and position accuracy with cost. The lower the latency and the higher the position tracking, the greater the cost of the system. This along with the other correlations and weights are summarized in the House of Quality in Table IV.

Finally, the relative weights in the House of Quality are determined based on the customer importance and relationship to the functional (engineering) requirements. The relative weights in the leftmost column are proportional to their customer importance. For example, the musical composition and the visualization both possess an equal importance of 7 out of 10. The relative weights adjust these numbers to a percentage scale based on the importance ratings of the other requirements. The relative weights in the bottommost row are determined from the importance of each customer requirement and their relationship to the functional requirements. For example, the functional requirement of cost has strong correlations to all the customer requirements and therefore has a large relative weight of 30%. This along with the other relative weights are summarized in the House of Quality in Table IV.

Table III. Requirements Analysis Framework

Customer Requirements	**Engineering Requirements**	**Justification**
1, 2, 5	The system should function in near-real time, with a latency of under 1 second. Therefore, wireless communication and algorithmic complexity must support low latency.	In order to produce a musical composition and a visualization for a live performance, the latency of the system should be limited. This allows the audience to associate the system outputs with the performer's live movements.
1, 2, 5	The position tracking system should function with an accuracy of 1 foot from the true position for a duration of 30 seconds.	In order for the system to cover 2 octaves (or 15 piano notes), each note should cover a range of 1 foot for a total range of 15 feet. The Chinese pole is 18 feet tall. However, the performer rarely uses the top and bottom portions of the pole, so 1 foot at the top and 2 feet at the bottom will be not musicalized. Using a 15-foot range, the position tracking system, therefore, requires an accuracy of 1 foot, so the performer's position corresponds to the correct piano note.
3, 4	Average setup time for the system hardware should not exceed 20 minutes.	This requirement ensures the system can be easily attached to and removed from the Chinese pole.
1-5	Production cost should not exceed $500.	This requirement is based on departmental funds.
4, 5	The system hardware should not protrude more than 1 centimeter from the Chinese pole.	This requirement ensures the safety of the performer. Any protrusion larger than this specification would interfere with the performer's natural movement on the pole.
1. The system should produce a musical composition with a range of 2 octaves of piano notes that corresponds to the performer's live vertical position on the Chinese pole. 2. The system should produce a visualization based on a bird's eye view of the performer's live position on the Chinese pole. 3. The system hardware should be able to be easily attached to and removed from the Chinese pole. 4. The system should be safe for a performer to use (given proper training) and not add to the risk of the Chinese pole. 5. The system should have low cost.		

Table IV. House of Quality

Correlations	
Positive	+
Negative	-
No Correlation	

Relationships		**Weight**
Strong	●	9
Medium	○	3
Weak	▽	1

Direction of Improvement	
Maximize	▲
Target	□
Minimize	▼

-	-			
	-	+		
	-		-	

Relative Weight	**Customer Importance**	**Customer Requirements**	Latency (<1 s)	Position Accuracy (<1ft., 30 s)	Setup Time (<20 min.)	Cost (<$500)	Protrusion (<1cm)
		Engineering Requirements					
		Direction of Improvement	▼	▲	▼	▼	▼
22%	7	Musical Composition	●	●		●	
22%	7	Visualization	●	●		●	
12%	4	Easy installation and removal			●	▽	
25%	8	Safety			▽	▽	●
19%	6	Cost	▽	●		●	○
		Importance Rating Sum (Importance x Relationship)	413	563	138	600	281
		Relative Weight	21%	28%	7%	30%	14%

C. System Description

An overview of the Piano Pole system functionality is shown in Fig. 13 and Table V below, which includes the inputs and outputs to the system in addition to a general description of the system functionality.

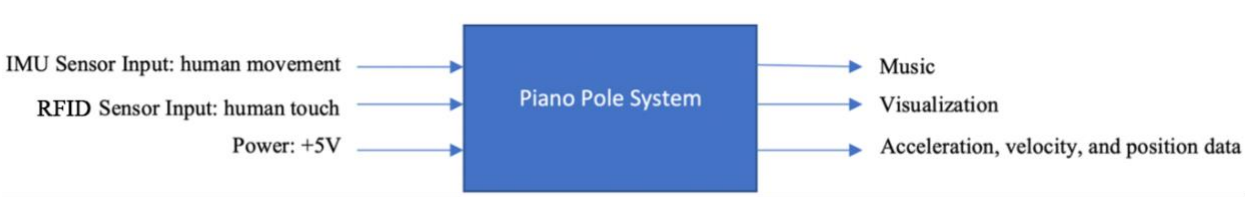

Fig. 13. Level 0 Piano Pole System Functionality

Table V. Level 0 Piano Pole System Functionality

Module	Piano Pole System
Inputs	• Human movement (IMU) • Human touch (RFID sensor input) • Power (+5V)
Outputs	• Music • Visualization • Acceleration, velocity, and position data
Functionality	The system should produce musical notes corresponding to the vertical height of the performer. The system should also produce a visualization depicting a bird's eye view of the performer's movements over time. Additionally, the performer's acceleration, velocity, and position should be output for further use.

Performing functional decomposition to identify smaller units of functionality that make up the overall system, Fig. 14 depicts how these units interact. Tables VI-X then describe the Piano Pole system in terms of the functionality of these first-level blocks.

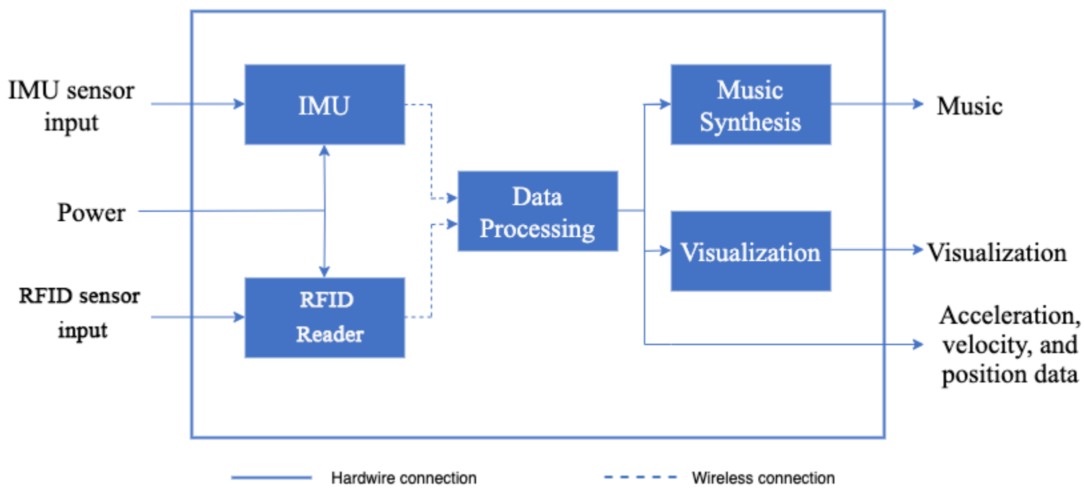

Fig. 14. Level 1 Piano Pole System Design

Table VI. Level 1 Inertial Measurement Unit Functionality

Module	Inertial Measurement Unit (IMU)
Inputs	• Human movement • Power (+5V)
Outputs	• Linear acceleration vector • Rotation vector
Functionality	Measure linear acceleration, angular acceleration, and the direction and strength of the surrounding magnetic field. Process these measurements and output a linear acceleration vector (with gravitational acceleration removed) and a rotation vector describing the sensor's orientation.

Table VII. Level 1 RFID Reader Hardware Functionality

Module	RFID Reader Hardware
Inputs	• Human touch • RFID receiver signal • Power (+5V)
Outputs	• Code indicating which RFID tag was touched and which IMU was closest
Functionality	Measure proximity of RFID reader to RFID tags on the pole and trigger RFID reader when tags are within close range. Communicate a code indicating which tag was read and which IMU was the closest to the sensor.

Table VIII. Level 1 Data Processing Functionality

Module	Data Processing
Inputs	• IMU linear acceleration vector • IMU rotation vector • Touch sensor data
Outputs	• Acceleration, velocity, and position data
Functionality	Determine velocity and position data in three dimensions based on the IMU and touch sensor data. The position should have an accuracy of one foot from the true position for a duration of thirty seconds.

Table IX. Level 1 Music Synthesis Functionality

Module	Music Synthesis
Inputs	• Acceleration, velocity, and position data
Outputs	• Musical composition
Functionality	Play corresponding piano notes when the performer enters the note's defined vertical region.

Table X. Level 1 Visualization Functionality

Module	Visualization
Inputs	• Acceleration, velocity, and position data
Outputs	• Visualization
Functionality	Create a visual art piece that demonstrates the position of the sensor from a bird's eye view of the Chinese pole.

Fig. 15 provides a flowchart describing the detailed algorithm for the Data Processing block. The program begins by initializing the Kalman filters for each remote system attached to the performer's body. All three dimensions of the state variables of acceleration, velocity, and position are set to 0, with the initial variance set to 20. The next step is determining if there are any IMU or RFID readings available. Once a reading is found, the program determines whether this data is coming from the IMU or the RFID reader. If the data is from the IMU, the program receives the linear acceleration and rotation vectors from the sensor and transmits the data wirelessly via XBee. Next, the rotation vectors are used to orient the linear acceleration vector appropriately. If the data is from the RFID reader, the program receives the RFID tag information and transmits the data wirelessly via XBee. In either case, the Kalman filter predicts the next state based on the state variables determined in the last update and the time in between readings. For example, the previous velocity is multiplied by the difference in time to predict a new position. In addition, the Kalman filter calculates an uncertainty associated with its prediction, which is based on the assumption of Gaussian distributions. Next, the acceleration (from the IMU) or position (from the RFID reader) measurement is averaged with the prediction, with each weighted by its estimated uncertainty. Thus, acceleration measurements are given an estimated variance of 0.5, while RFID measurements are given a much lower uncertainty of 0.001. The result is used to update the state variables of acceleration, velocity, and position. The updated position is displayed to the audience as a circle in the matching location. If a note's vertical region has been entered, a piano note is played. Finally, after these steps the program checks for available readings again.

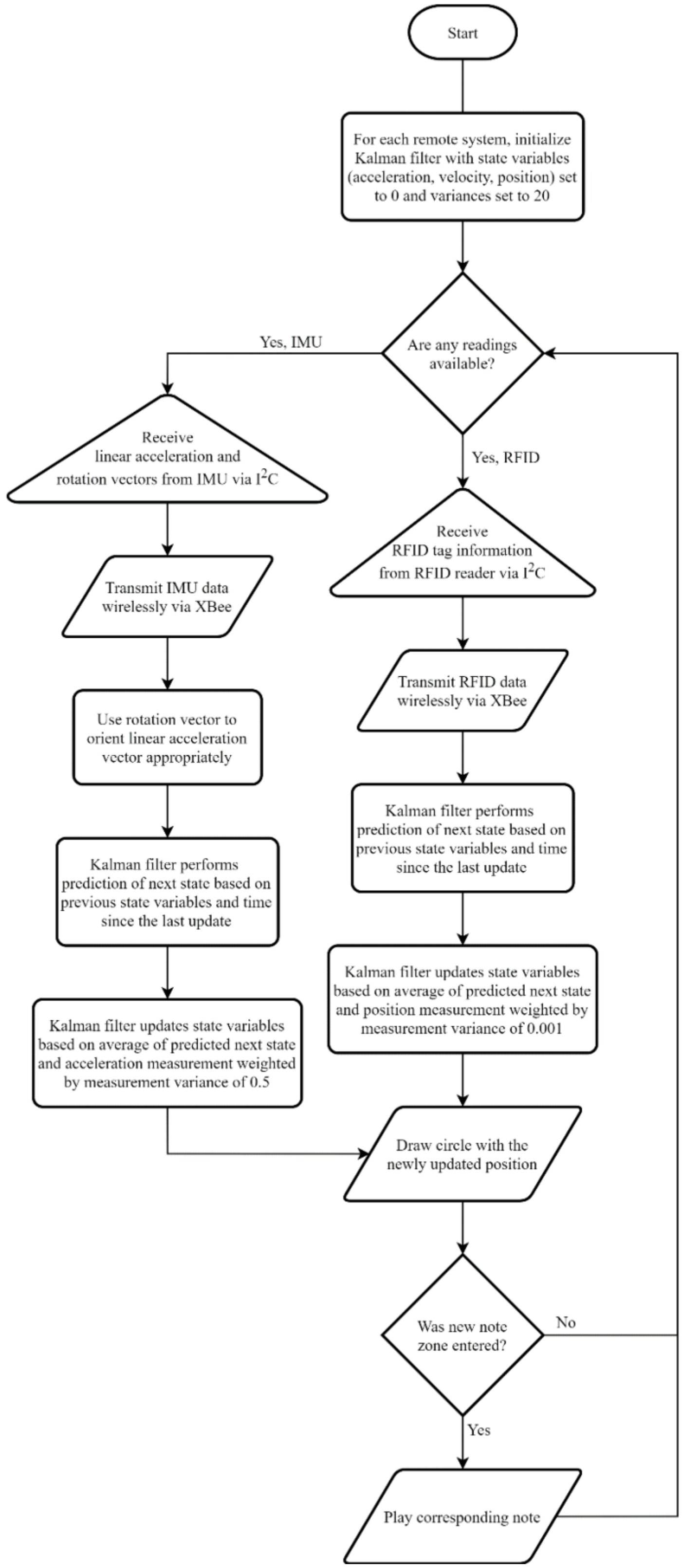

Fig. 15. Flowchart describing the algorithm performed by the Data Processing block

D. Standards and Constraints

The Piano Pole system makes use of a variety of standards and protocols for data transmission. Utilizing these pre-existing standards creates a more efficient and universal system. The standards and protocols used in the Piano Pole system include:

- IEEE 802.15.4 (Wireless Communication protocol) [13]
  - The IEEE 802.15.4 protocol is a standard for wireless communication issued by IEEE that the XBee modules used in this system adhere to. The protocol defines a wireless personal area network (WPAN) which focuses on low-cost, low-power, and low-speed communication between devices. Three possible frequency bands are proposed for operation, 868 MHz, 915 MHz, and 2450 MHz, and the standard supports a 10-meter range with a transfer rate of 250 kb/s.
- NXP UM10204 (I^2C protocol) [14]
  - The BNO085 IMU and the RFID reader use the Fast-mode I^2C protocol to communicate with the Arduino. The I^2C protocol, defined by the NXP UM10204 standard, involves serial communication over two wires, serial data (SDA) and serial clock (SCL). The Arduino acts as the master node, which supplies the clock signal and initiates communication by sending a START signal followed by the address of the slave node it wants to communicate with. The slave device then responds with an active low acknowledged (ACK) bit and begins to transmit big-endian messages, each beginning with a START signal and concluding with a STOP signal. The Fast-mode I^2C protocol supports speeds of up to 400 kb/s.
- EPC UHF Gen2 Air Interface Protocol (RFID protocol) [15]
  - The RFID readers and antennas used in this system follow the "Gen2" interface protocol, which defines the physical and logical requirements for an RFID system of interrogators and passive tags, operating in the 860 MHz – 960 MHz range.
- USB 2.0 (Universal Serial Bus 2.0 protocol) [16]
  - The microcontroller communicates with a computer via the Universal Serial Bus (USB) 2.0 protocol. This protocol uses two wires for power and a twisted pair for data. USB 2.0 enables speeds of 480 Mb/s and specifies a minimum of 5V and 100mA, allowing devices to receive power as well as data over a USB connection.

The Piano Pole system is a complex and powerful system, but with these advantages also come limitations. The system is meant to operate in very specific conditions, primarily designed to function in a theater venue. This environment allows for more control, higher accuracy, and better results from the system. In summary, the Piano Pole system is designed to be limited to the following constraints:

- The system is designed to function in a controlled environment inside a theater. It is not designed to work outside where the equipment is susceptible to significant natural damage.
- The system is meant to be used by a performer who is skilled in the Chinese pole apparatus and is also properly trained on the Piano Pole system.
- The RFID tag stickers are meant to be attached to a Chinese pole with a standard height of 18ft (5.5m). The system also assumes the Chinese pole has already been properly erected and secured tightly to the floor using floor bolts.

- The position tracking will maintain an accuracy of within one foot from the true position, given that the performer makes contact with a band of RFID tags at least every 30 seconds.

E. Design Impact

The idea of using external sensors to address IMU position tracking inaccuracies had not been studied as well as algorithmic solutions for complex, unpredictable movement; previous research focused on algorithmically determining when recognizable events such as footsteps occurred. Therefore, our solution of using touch sensors to periodically recalibrate the position of each IMU is a novel one that we plan to share with the engineering community. We have begun to do so already through our public GitHub repository [13]. In addition, we presented our project and findings at the National Conference on Undergraduate Research (NCUR) 2021 @Home, as well as at LMU.

The musical composition and visualization are works of art that allow audiences to view a Chinese Pole act from a new perspective that would not be possible without the technology we are using. Additionally, the music and visualizations capture the essence of live performance. The position of movement on the pole will differ slightly each time it is executed, which means outputs are always unique. These sounds and images retain the live aspect of circus performance, which give more meaning to the music heard or the art drawn on the canvas. This system creates a unique perspective for anyone experiencing this art. Furthermore, it provides a benefit to society on a global level because it gives people the opportunity to rethink the way they view circus arts, and in turn rethink the way they view life.

Additionally, the position tracking system could be utilized for training or teaching applications. Students could execute different acrobatics skills and the coaches would be able to review the position data graph to determine how the performers hands and feet were moving over time during those skills. The coach can then provide more detailed feedback on how to improve the student's technique. In these ways, the Piano Pole system provides a benefit to society.

With regard to its environmental impact, the Piano Pole system is meant to be reused many times and is built to last. It is not designed as a one-time-use system, and while some components may degrade over time, a user will only need to replace that specific components instead of the entire system. Furthermore, the system can last a long time while still being relevant to society. Even with the hardware remaining the same, there are many ways to alter the software elements of the system so that the music and visualizations are unique in each performance.

Finally, with regard to economic impact, this system is novel in both the engineering and circus arts industries. Therefore, the system has the potential to attract a larger audience to circus arts shows, which will increase ticket sales for circus companies. The companies can then re-invest these funds to push the art form to a new level.

IV. Electrical Design

A. Wiring Diagrams

Fig. 16 and 17 display the wiring diagrams for the remote (transmitting) systems attached to the gloves and the shoes of the performer. The system for the gloves is made up of a SparkFun Qwiic Pro Micro microcontroller connected to a BNO085 IMU and an RFID reader by Qwiic cables, which make use of the I^2C protocol. The Qwiic Pro Micro is also connected to a SparkFun XBee 3 Thing Plus by a UART connection. Finally, each Qwiic Pro Micro is powered by a 110mAh lithium polymer (LiPo) battery connected via JST PH connector to a breakout board. The system for the shoes is identical, but with a LilyPad XBee in place of the XBee 3 Thing Plus.

Figure 18 shows the receiving end of the system, which was connected via USB to a laptop computer. This system consists of an XBee 3 Module connected to a SparkFun XBee Explorer Dongle.

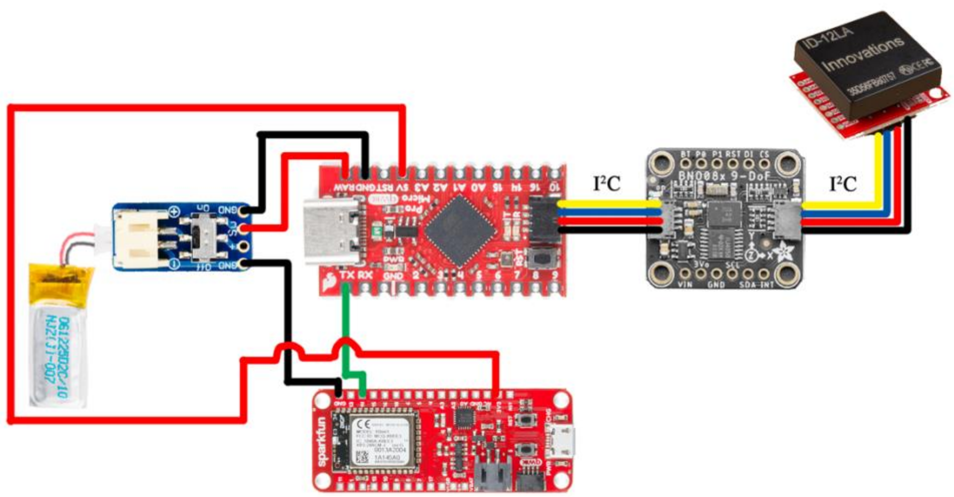

Fig. 16. Wiring diagram of remote system attached to gloves

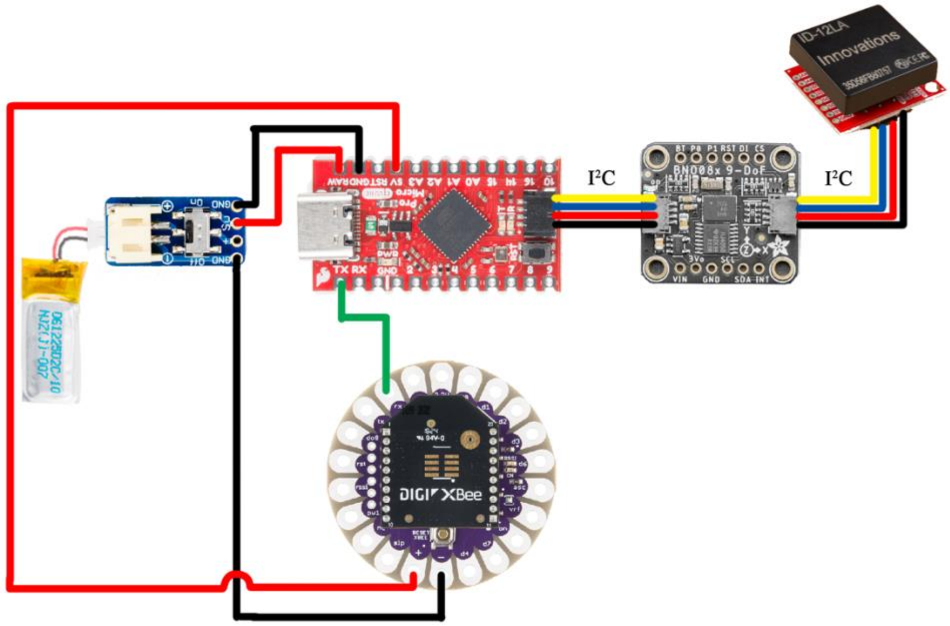

Fig. 17. Wiring diagram of remote system attached to shoes

Fig. 18. Diagram of receiving system connected to computer

B. Bill of Materials and Cost Estimates

The materials used in this project and their associated costs are presented in Table XI.

Table XI. Bill of Materials

Component Name	Part Number	Cost per Item	Quantity	Total Cost
9-DoF IMU	BNO085	$19.95	4	$79.80
SparkFun XBee 3 Thing Plus	WRL-15454	$49.95	2	$99.90
SparkFun Qwiic Cable Kit	KIT-15081	$11.94	1	$11.94
RFID Reader	ID-12LA	$29.95	4	$119.80
SparkFun RFID Qwiic Reader	SEN-15191	$19.95	4	$79.80
125kHz RFID Soft Paper Stickers	B07PCJKPZZ	$7.98	3	$23.94
SparkFun Qwiic Pro Micro	DEV-15795	$19.95	4	$79.80
LilyPad XBee (Fosse)	DEV-1292	$15.95	1	$15.95*
LilyPad XBee	DEV-1292	$15.95	1	$15.95
110mAh E-Textiles Battery (2C Discharge)	PRT-13112	$6.95	6	$41.70
XBee 3 Module	WRL-15126	$17.95	3	$53.85*
SparkFun XBee Explorer Dongle	WRL-11697	$25.95	1	$25.95*
Compression Gloves	-	$16.99	1 pair	$16.99
Acrobatics Shoes	-	$31.95	1 pair	$31.95
50mm Qwiic Cable	PRT-14426	$0.95	6	$5.70
JST PH Right Angle Connector	PRT-08612	$0.95	8	$7.60
Switched JST PH Right Angle Breakout Board		$2.99	4	$11.96
JST PH 4-Pin to Male Header Cable	PRT-14425	$1.50	2	$3.00
5V DC-DC Step Up Power Module Voltage Boost Converter (5 pack)		$7.98	1	$7.98*
Tax + Shipping	-	-	-	$34.11
			Total Hardware Cost:	$767.67
			Total Cost of Materials Purchased:	**$663.94**

*Items already available to us

C. Mechanical Drawings

The final construction of the shoes and gloves for the Piano Pole system are shown in Fig. 19-22 and the mock Chinese Pole made out of PVC pipes is shown in Fig. 23.

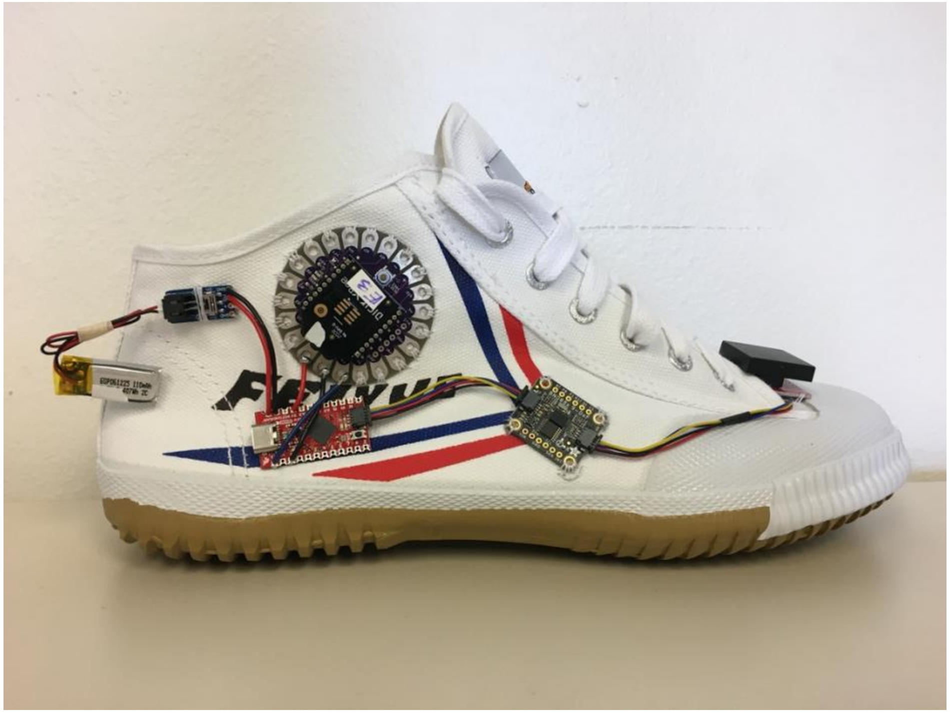

Fig. 19. Right shoe of Piano Pole system

Fig. 20. Left shoe of Piano Pole system

Fig. 21. Left glove of Piano Pole system

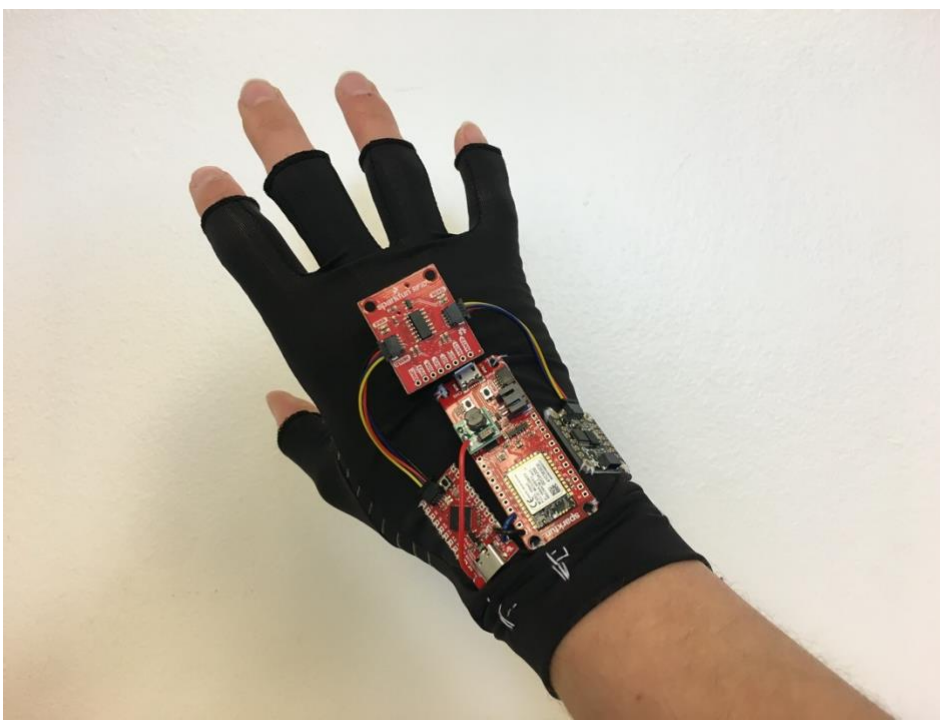

Fig. 22. Right glove of Piano Pole system

Fig. 23. Mock Chinese Pole constructed out of PVC pipes with RFID tag bands attached

D. System Design

After being wired as shown in Fig. 16 and 17, the transmitting systems were sewn to gloves and acrobatic shoes for the performer to wear, as demonstrated in Fig. 19-22. The receiving system shown in Fig. 18 was plugged into a laptop computer.

Since a Chinese pole was not available to us, a mock Chinese pole was constructed using polyvinyl chloride (PVC) pipe with a similar diameter and height. Two bands of four RFID sticker tags were placed around the pole at 1-meter intervals, as shown in Fig. 23. The PVC Chinese pole was then placed upright next to a planter, used to mimic climbing the pole.

With this setup, the entire Piano Pole system is able to be tested. A performer wears the gloves and shoes with the remote hardware attached. The receiver hardware is connected to a computer running the position tracking Python script. The performer begins by touching each RFID reader to an RFID tag on the pole in order to initialize its position. The performer then proceeds to perform by moving around the base of the pole and climbing next to it (or climbing the Chinese pole itself if one were available). The associated visualization and musicalization should then be played from the laptop.

V. Experimental Test and Demonstration

A. Setup Procedure

Before performing the experimental tests, the Piano Pole system needs to be properly set up. For the hardware in the Piano Pole system, the performer must put on all four remote systems including the two gloves and two shoes. Batteries need to be inserted into the system and switched to the "ON" position. The IMU's orientation should be tared before beginning any test. RFID tags must be installed at 1-m heights along the mock Chinese Pole. A receiver XBee should be connected a computer.

To install the necessary software for the Piano Pole system, first clone the project's GitHub repository [17] to your local computer by running

`git clone --recursive https://github.com/fosselb/piano-pole`.

This repository includes the Arduino script for the Qwiic Pro Micro microcontrollers and the Python script for tracking the position of the performer and producing visualizations and music. In addition, we include our modified BNO085 Arduino library [18], based on the SparkFun BNO080 IMU library [19]. Finally, we make use of piano samples obtained from MIDI.js Soundfonts [20].

Next, with the Arduino IDE [21] installed, install the SparkFun Qwiic RFID Reader Arduino Library [22] from the Library Manager. Then, follow this SparkFun setup guide [23] to install the Arduino add-on necessary for interfacing with the Qwiic Pro Micro.

Finally, a number of Python packages must be installed for proper operation of *piano_pole.py*: the plotting library Matplotlib [24], NumPy [25] for matrix support, the playsound [26] library for playing piano samples, Processing Python [27] for graphics, the Kalman filter library FilterPy [28], SciPy [29] for additional mathematics, and the serial communication library pySerial [30]. All of these packages can be installed with PyPI [31] using `pip install <package>`.

Once the installation process is completed, and with hardware set up according to Fig. 19-22 in the Mechanical Drawings section, upload the *Qwiic_Pro_Micro.ino* Arduino script to all four Qwiic Pro Micro microcontrollers. Disconnect the microcontrollers from the computer and plug in the batteries with all four IMUs oriented along the same axes. Put on the gloves and shoes and prepare to begin the performance. When ready, with the receiver XBee plugged into the computer, run *piano_pole.py*.

The Python script can be run with the serial port of the receiver XBee using the command

`python piano_pole.py -p <port>`.

A sample output reflecting recorded data can be produced by running

`python piano_pole.py -t -i logs/log7.txt`.

An additional flag -o allows for the output of readings to a .txt file.

For further information about the available flags and their operations, please see `python piano_pole.py -h`:

```
$ python piano_pole.py -h
usage: piano_pole.py [-h] [-p PORT | -i INPUT] [-o OUTPUT] [-t] [-r]

Visualize and musicalize a Piano Pole performance.

optional arguments:
  -h, --help            show this help message and exit
  -p PORT, --port PORT  serial port to listen to (default COM8)
  -i INPUT, --input INPUT
                        file to read from
  -o OUTPUT, --output OUTPUT
                        file to write to
  -t, --time            play back in real time
  -r, --recording       play beep at start for recording purposes
```

Additionally, the five XBee modules in the system need to be configured properly using the XCTU software [32] as described in Table XII.

Table XII. Configuration settings for the XBee modules in the system

Configuration Setting	Coordinator XBee	XBee Thing Plus 1 (Right glove)	XBee Thing Plus 2 (Left glove)	End Device XBee 3 (Right shoe)	End Device XBee 4 (Left shoe)
Network PAN ID	F077	F077	F077	F077	F077
Node Identifier (NI)	Coor_XBee	End_XBee1	End_XBee2	End_XBee3	End_XBee4
Device Role	Coordinator	End Device	End Device	End Device	End Device
16-bit Source Address (MY)	0x0000	0x0001	0x0002	0x0003	0x0004
Destination Address High (DH)	0x0000	0x0000	0x0000	0x0000	0x0000
Destination Address Low (DL)	0xFFFF	0x0000	0x0000	0x0000	0x0000
API Enable	API Mode without Escape	Transparent Mode	Transparent Mode	Transparent Mode	Transparent Mode

UART Baud Rate	115200	9600	9600	9600	9600

B. Benchmark Tests

i. Unit Testing

The following testing procedures were modified from the Fall Final Report due to changes in design of the project. Specifically, the RFID system replaced the touch sensor hardware originally proposed, so the touch sensor unit tests were replaced with the appropriate RFID unit tests. Each of the results described in this section corresponds to a detailed test given in Appendix C.

The Arduino Serial Communication Unit Test results are shown in Fig. 24. This test requires that the BNO085 IMU be connected via I^2C to the Arduino Qwiic Pro Micro board. The Pro Micro should be connected via USB cable to a computer with the Arduino IDE installed. This test is designed to verify the proper functioning of both the IMU and the Arduino.

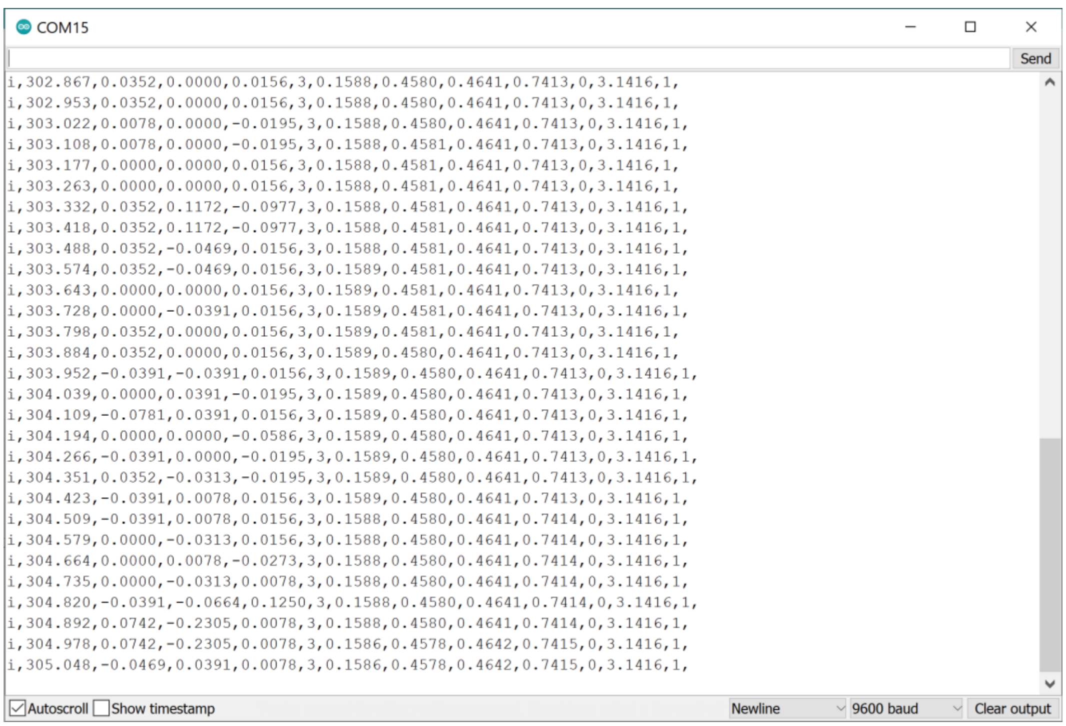

Fig. 24. Serial Monitor demonstrating serial communication between IMU and Arduino

The RFID Sensor Hardware Unit Test results are shown in Fig. 25 and 26. Fig. 25 shows a screen shot of the video test. The full video can be found in the 'Test Results' folder or on YouTube [33]. This test requires that the RFID sensor hardware be connected to the Arduino Qwiic Pro Micro, which is in turn connected via USB cable to a computer with the Arduino IDE installed. This test is designed to verify that a unique digital output is produced when each RFID rag band is touched.

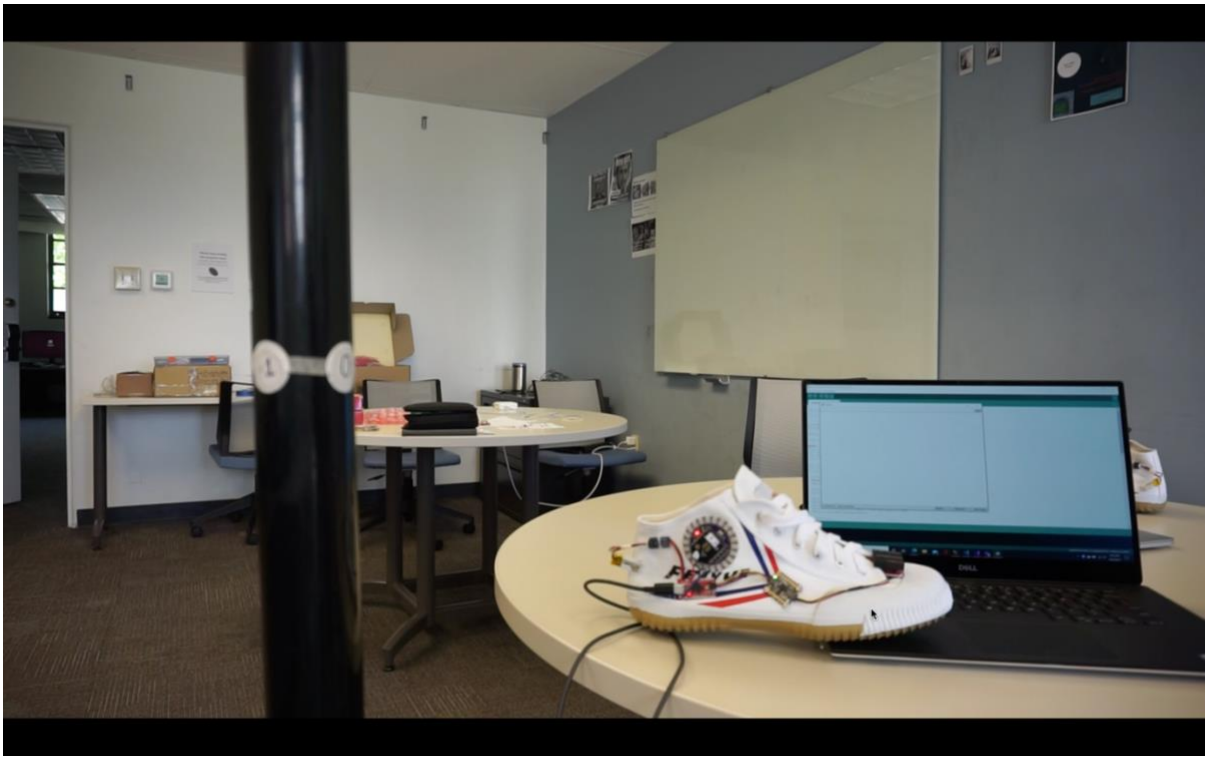

Fig. 25. Screenshot of RFID Sensor Hardware Unit Test video [33]

Fig. 26. Serial Monitor demonstrating all eight unique RFID tags on the mock pole

The Musical and Visual Elements Unit Test result is shown in Fig. 27, which depicts a screenshot of the video test. The full video can be found in the 'Test Results' folder or on YouTube [34]. This test requires a computer with the Processing Python [27] library installed. The test is designed to verify the musical and visual outputs of the Piano Pole system.

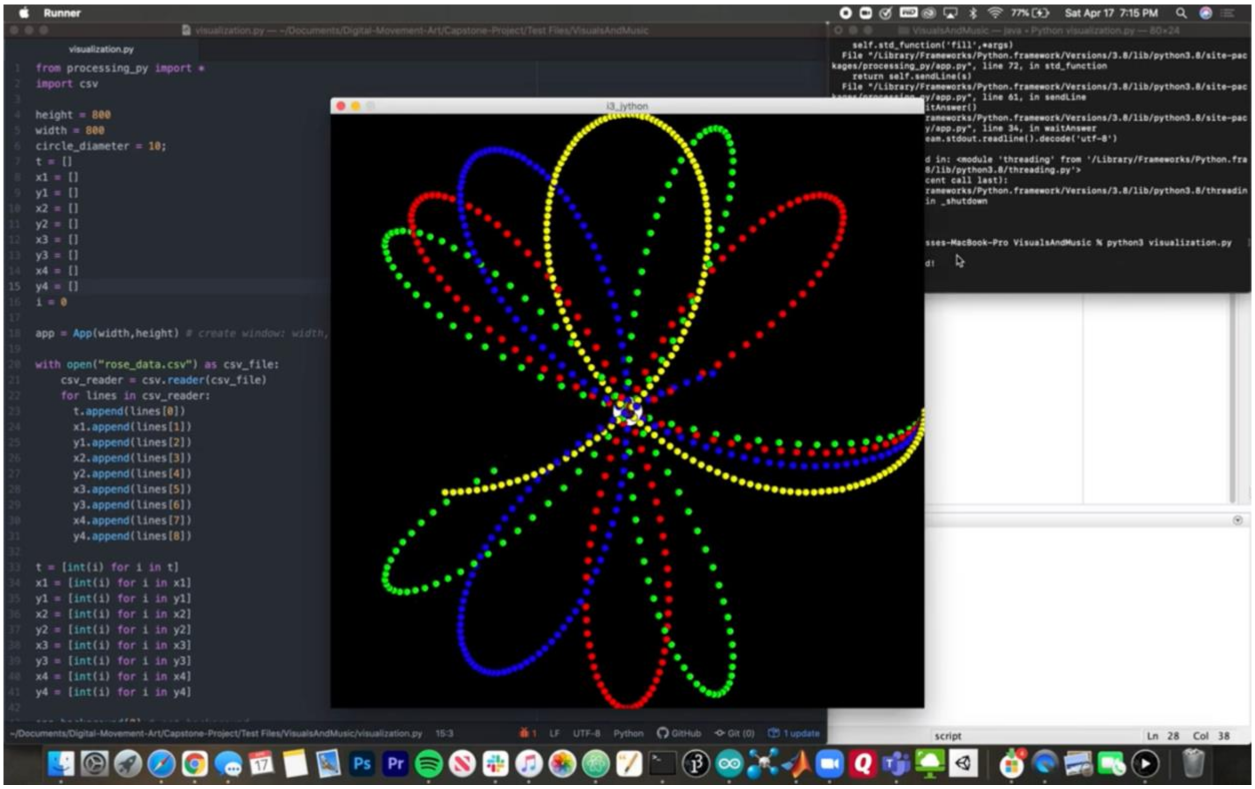

Fig. 27. Screenshot of Musical and Visual Elements Test video [34]

ii. Integration Testing

The IMU Wireless Communication Integration Test result is shown in Fig. 28, which depicts screenshots of the 'log2.txt' file from Test 2 of the Piano Pole Integration Test. In the log file, the 'i' represents a packet of IMU data and the 'r' represents a packet of RFID data. This test requires a coordinator XBee module connected to a computer with the Python program set up. In addition, the test requires that the four remote systems (two gloves and two shoes) are powered on. Finally, all the XBee modules need to be configured according to Table XII. This test is designed to verify that the IMU and RFID sensor data is properly sent and received wirelessly via the XBee modules.

```
<2>
0.1602,3,-0.8843,-0.1874,0.4208,0.0760,3,0.0520,4,
i,63.280,-0.0313,0.1836,0.2891,3,-0.8843,-0.1874,0.4208,0.0760

<1>
73,3,0.0569,4,
i,90.386,0.2930,0.2539,-0.0391,3,0.0682,-0.5576,0.6018,0.5677,3,0.0569,4,
i,90.456,-0.0078,0.3164

<4>
-0.3603,-0.8841,0.2812,3,0.0801,3,
i,70.261,0.0234,-0.0078,0.0586,3,-0.0969,-0.3615,-0.8836,0.2813,3,0.0801,3,
i

<3>
0.0469,0.0898,3,0.5258,0.1715,-0.2933,0.7798,3,0.0708,3,
i,71.933,0.0469,-0.0469,0.0898,3,0.5258,0.1715,-0.2933,0
```

(a)

```
<1>
5251,0.3247,3,0.0630,3,
r,99.129,38,
i,99.181,-0.0703,0.0469,-0.0898,3,0.1570,-0.7708,0.5251,0.3247,3,0.0630,3,
```

(b)

```
<2>
07.681,-0.4844,-0.2500,1.5859,3,-0.7101,0.1168,-0.1615,0.6754,3,0.0803,4,
r,107.676,38,
i,107.769,-1.0039,0.2070
```

(c)

```
<3>
40,3,
i,189.737,-0.8789,0.0977,0.4063,3,0.1828,0.2136,0.6689,0.6881,3,0.0632,3,
r,189.793,227,
i,189.824,-0.878
```

(d)

```
<4>
,0.1448,0.1482,0.7122,0.6707,3,0.1348,4,
r,86.020,227,
i,86.042,0.9766,-1.0000,-0.8086,3,0.1448,0.1482,0.7122,0.
```

(e)

Fig. 28. Screenshots of log2.txt file showing IMU and RFID data properly received from all four remote systems. (a) IMU data received from systems 1-4 (b) RFID data received from system 1 (c) RFID data received from system 2 (d) RFID data received from system 3 (e) RFID data received from system 4

The Position Tracking Accuracy Integration Test results are shown in Fig. 29 and 30. Fig. 29 depicts a screenshot of the video test and Fig. 30 shows the position graphs of all the IMUs. The full video can be found in the 'Test Results' folder or on YouTube [35]. These tests require that the IMU position tracking and RFID sensor systems be fully set up as described in the Setup Procedure section. This test is designed to verify that the accuracy of the position tracking system alone.

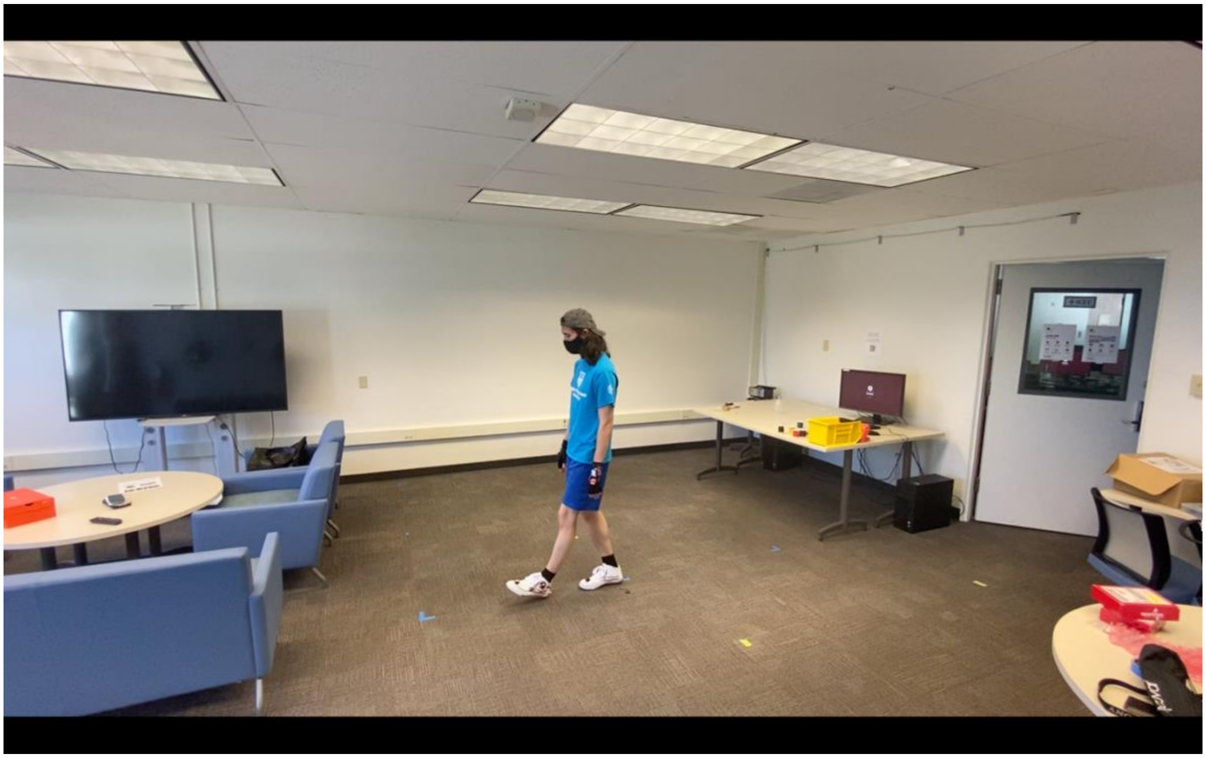

Fig. 29. Screenshot of Position Tracking Accuracy Test video [35]

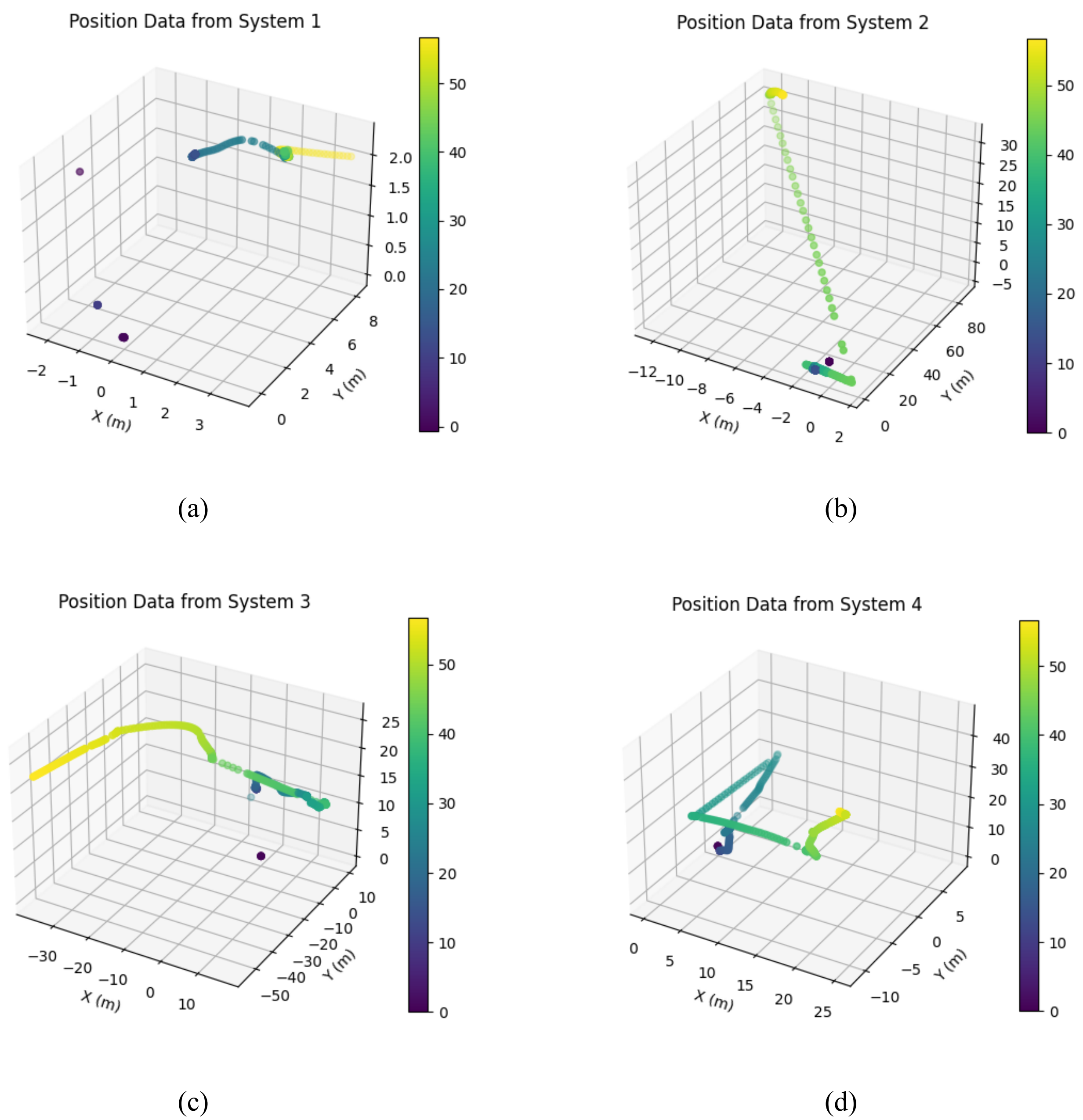

Fig. 30. 3-D plots of the position data from each IMU sensor over time

The Position Tracking Recalibration Integration Test result is shown in Fig. 31, which depicts a screenshot of the Serial Monitor and the corresponding position graph of the data. These tests require that the IMU position tracking and RFID sensor systems be fully set up as described in the Setup Procedure section. This test is designed to verify that the position tracking system recalibrates the position of the performer when it comes in contact with an RFID tag.

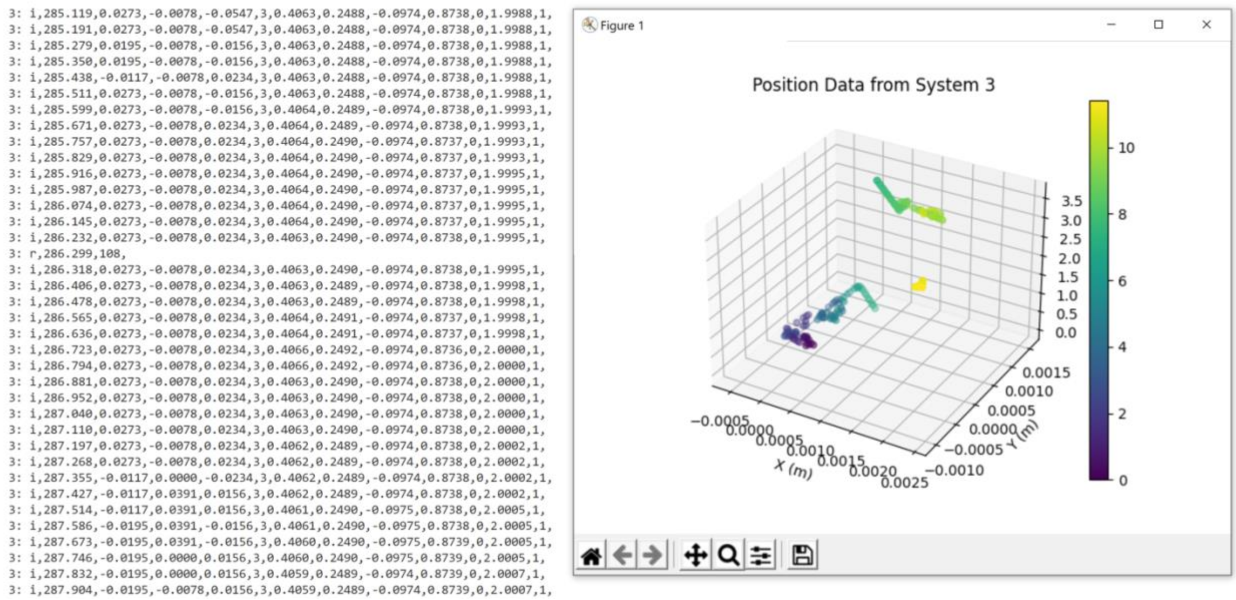

Fig. 31. Serial Monitor demonstrating an RFID tag was scanned and the position graph showing how the position was updated accordingly

The Combined Position Tracking Accuracy and Recalibration Integration Test results are shown in Fig. 32 and Fig. 33. The video test used to verify this benchmark was also a part of the Piano Pole Integration Test. It is referred to as Test 1 in the Piano Pole Integration Test, so please refer to this test for the full video. These tests require that the IMU position tracking and RFID sensor systems be fully set up as described in the Setup Procedure section. This test is designed to verify the combination of the position tracking accuracy and the recalibration.

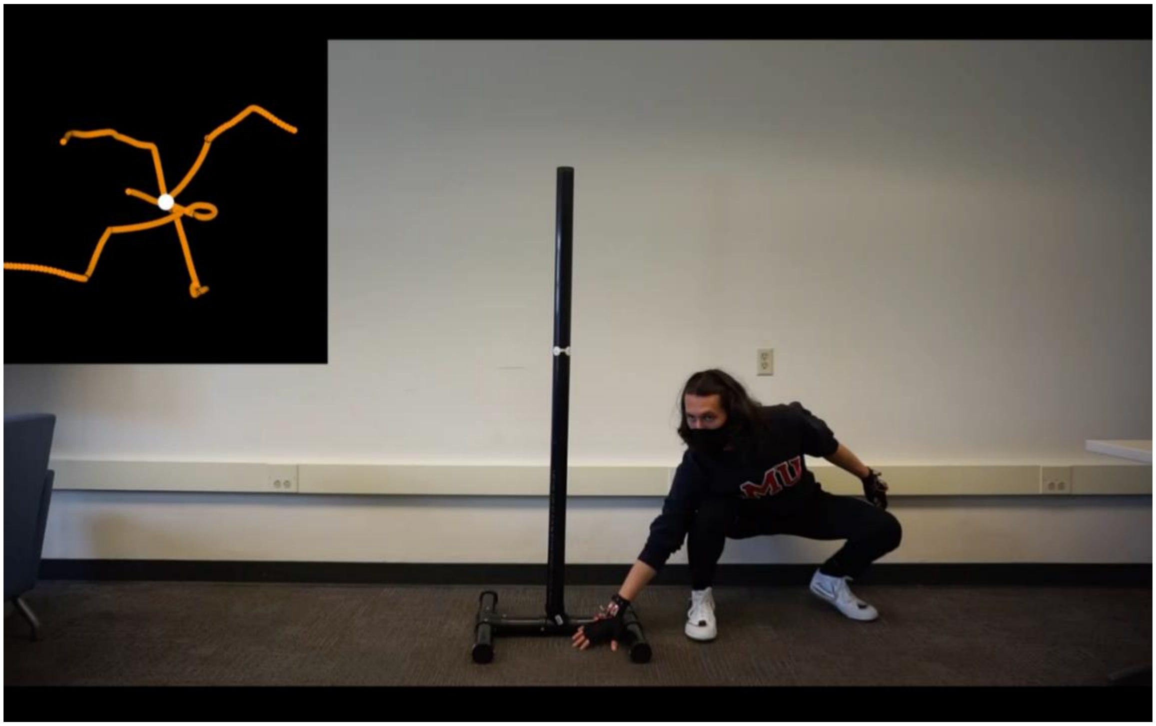

Fig. 32. Screenshot of Combined Position Tracking Accuracy and Recalibration Integration Test

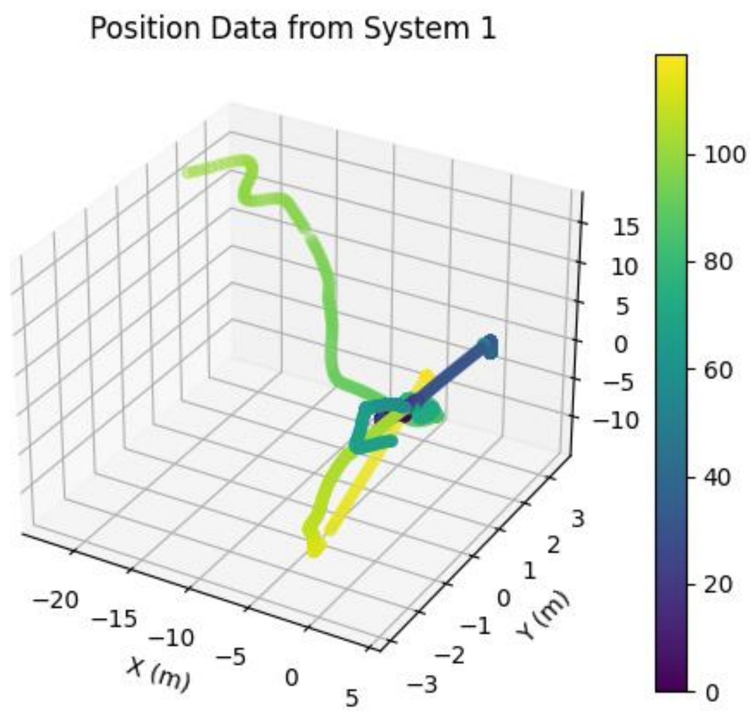

Fig. 33. Position data graph from Combined Position Tracking Accuracy and Recalibration Integration Test

The integration tests results for verifying system performance are described in the following tests. The Assembly and Disassembly Integration Test result is shown in Table XIII. This test requires that the four remote systems are already sewn into the fabric of the shoes and gloves. The test is designed to verify that the system can be installed in at most twenty minutes.

Table XIII. Assembly and Disassembly Test Results

Assembly Time	Time (min)
Initial assembly time - Attaching RFID stickers, putting on shoes and gloves	15
Recurring assembly/disassembly time - Putting on/taking off shoes and gloves	5

The Touch Sensor Protrusion Integration Test result is shown in Fig. 34, which shows a picture of the hardware on the Chinese pole. This test requires that the RFID tags be installed on the Chinese pole. The test is designed to verify that the system does not protrude more than one centimeter from the pole.

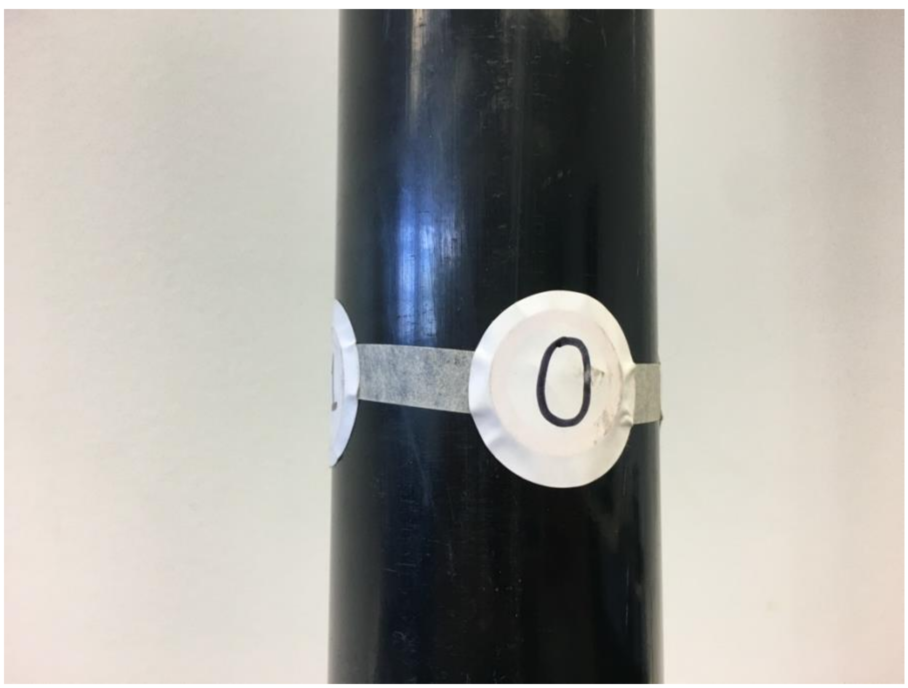

Fig. 34. Image demonstrating protrusion amount of RFID tags on the pole

The System Latency Integration Test result is shown in Fig. 35 and Table XIV. Fig. 35 shows a screenshot of the test result video. The full video can be found in the 'Test Results' folder or on YouTube [36]. This test requires that the entire Piano Pole system be installed. The test is designed to verify that system latency is at most one second. The latency measurement includes the time it takes to collect, transmit, process, visualize, and musicalize one data point.

Fig. 35. Screenshot of System Latency Integration Test video [36]

Table XIV. System Latency Test Results

System Latency Test	
System latency time (seconds)	0.67

Finally, the Piano Pole Integration Test result is shown through a series of seven video tests. The full video of these tests can be found in the 'Test Results' folder or on YouTube [37]. This test requires that the entire Piano Pole system be installed as described in the Setup Procedure section. This test is designed to verify the overall functionality of the Piano Pole system, with visual and musical effects corresponding to the performer's movements. Screenshots of the video tests are shown in Fig. 36-38.

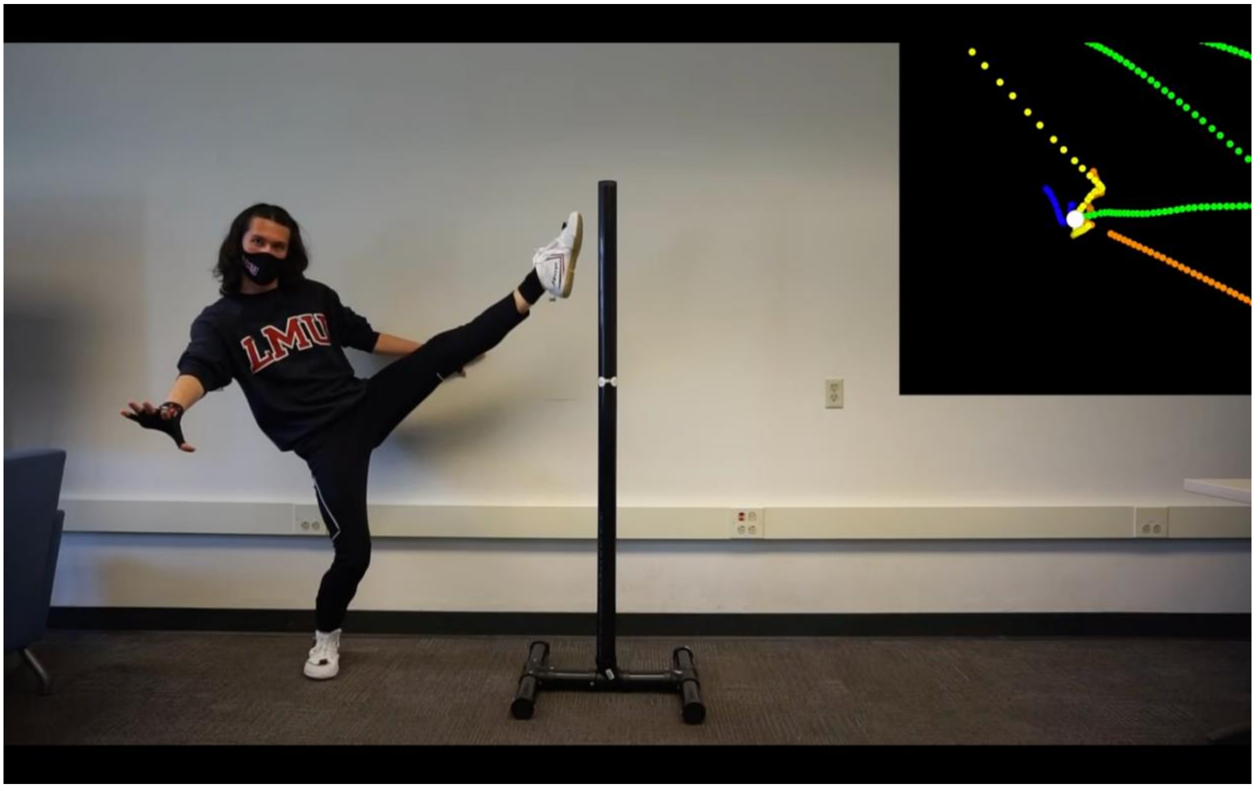

Fig. 36. Screenshot of Piano Pole Integration Test 5

Fig. 37. Screenshot of Piano Pole Integration Test 6

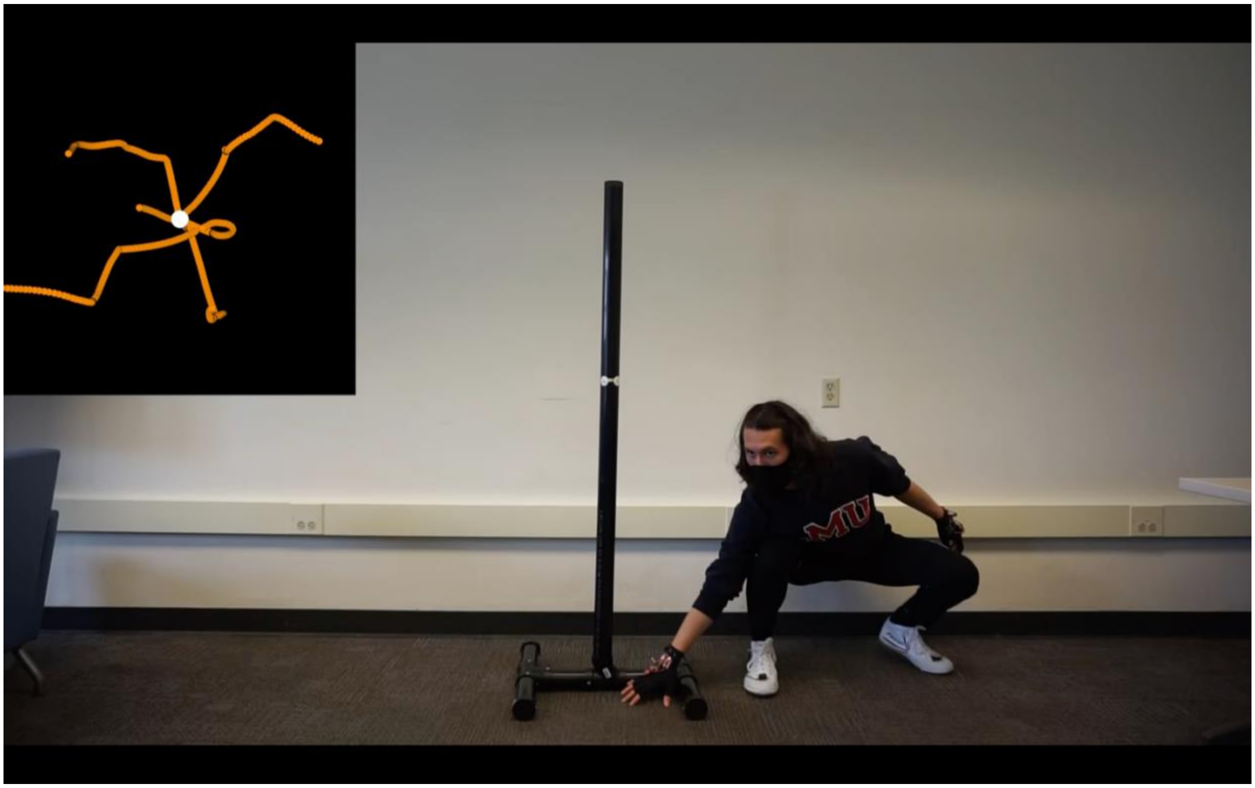

Fig. 38. Screenshot of Piano Pole Integration Test 1

C. Customer Requirements Met

Table XV describes testing results which validate each customer requirement. The first and second customer requirements, which discuss the visualization and musicalization of the performer's movement, were shown to be met by the above integration tests. However, the accuracy of the position tracking used to create these visualizations and music was lower than desired. The third customer requirement, stipulating ease of setup, is verified by the results in Table XIII. The safety customer requirement is verified by the test results demonstrated in Fig. 34. Finally the fifth customer requirement, discussing the cost of the system, was not met. The final cost given in table XI is significantly greater than $500. A future method for bringing this cost down is discussed in the Future Research section of this report.

Table XV. Description of how each customer requirement will be demonstrated as complete

Customer Requirements	**Results to demonstrate completion of requirement**
1. The system should produce a musical composition that corresponds to the performer's live vertical position on the Chinese Pole.	A video recording of the integration testing procedures will show the performer touching the Chinese pole and the corresponding musical notes will be played through the computer speakers and captured in the video recording. Additionally, an mp3 file of the music generated will be produced for each test.
2. The system should produce a visualization based on a bird's eye view of the performer's live position on the Chinese Pole.	The final visualizations generated from the Processing sketch will be saved for each test. Additionally, each data point being visualized will also be saved. As more data points are collected over time, an animation of the art pieces will show the art being created.
3. The system hardware should be able to be easily attached to and removed from the Chinese Pole.	The set up and take down time of the hardware system will be recorded.
4. The system should be safe for a performer to use (given proper training) and not add to the risk of the Chinese Pole.	The protrusion of the hardware on the Chinese pole will be measured and recorded.
5. The system should have low cost.	A final budget report of all the materials purchased will be demonstrated.

D. Data Analytics

Overall, the Piano Pole system was found to be very successful at meeting the established requirements. IMU and RFID readings were successfully transmitted wirelessly via XBee to a laptop computer, where it was received and processed properly. The results were visualized and musicalized successfully, according to the x-, y-, and z-components of the positions of the performer's hands and feet. The overall system latency was determined to be consistently below one second. The pole was made safe for the performer by adding only RFID tag stickers. While the cost was higher than desired, better use of the SparkFun XBee 3 Thing Pluses, as described in Future Research, would bring the cost down below $500.

However, the only other failure of the Piano Pole system is a major one. This is a failure to meet the accuracy requirement of less than one foot of error for thirty seconds. Unfortunately, despite the implementation of a Kalman filter to assist in reducing drift, IMU position tracking led to wildly variable results. This can be seen especially well in Fig. 30, which displays the results of the Position Tracking Accuracy Test, in which Fosse walked in a 3mx3m square. Remote system 4 does the best during this test, with the estimated position within 10m of Fosse's position and a clear square pattern visible. However, system 2 demonstrates massive drift, with the estimated position ending almost 90m from Fosse's actual position.

Some methods discovered to improve the system's performance in terms of accuracy included moving slowly and smoothly, so as to eliminate sudden jumps in acceleration, along with holding as still as possible when done with a movement. This holding still allows the IMU to stabilize and recognize that it is not being moved. Finally, touching RFID tags more often also helped to improve the position tracking accuracy significantly, since the locations of the RFID tags are known.

Although the final system is not as accurate as we had hoped, it does produce visualizations and musicalizations that are recognizable to the audience as representing the performer's movement.

VI. Ethics Considerations

Building and designing apparatuses for live performance involves inherent risk. This can range from risk of minor injury, such as a sprained ankle, to more severe consequences, including death. However, the risk can never be eliminated, and it is therefore ethical practice to ensure the safety of performers by aiming to minimize the risk of using these apparatuses whenever possible. While the performers bear a significant responsibility to operate with caution when interacting with these apparatuses, it is equally important for the engineers to design these apparatuses with the safety and welfare of the performers in mind. Additionally, it is crucial that the engineers ensure proper training of their products so the performers and technical staff can operate the technology safely and efficiently.

The IEEE Code of Ethics discusses these ethical issues in Section I, which includes the provision "to hold paramount the safety, health, and welfare of the public" [38]. Accordingly, an engineer has an ethical obligation when designing and producing a circus apparatus to consider the safety of the circus performer who will be using it. The Code's stipulation "to disclose promptly factors that might endanger the public" is also important when designing a circus arts apparatus [38]. If the engineer does not properly disclose all necessary safety information to technical staff and performers, then the performer's safety (and possibly their life) is not fully in their control. Failing to disclose all possible hazards and faults of the apparatus is therefore unethical behavior on the part of the engineer.

In this capstone project, we uphold these ethical standards to create a product that is both entertaining to watch and safe to use. We accomplished this by first designing the hardware attached to the Chinese pole such that it minimized the interference with the performer's movement. This safety consideration was integrated into the engineering requirement of the project when stating the system hardware should not protrude more than one centimeter from the pole. This requirement was achieved as the RFID sticker tags are the only piece of hardware attached to the pole. We also securely attached the IMUs and RFID readers to the shoes and gloves worn by the performer. This hardware does not contain any sharp edges or any other exposable material that could harm the performer in a significant way. Finally, while the risk of the system was minimized to the best of our ability, complete safety cannot be guaranteed. Therefore, in addition to writing technical documentation on the safety and risk of the apparatus, we also included a safety guide meant to be read by the performer and coaches. This safety guide highlights the important safety concerns and risks, so the performers do not need to sift through long technical documentation to find pertinent safety information.

As mentioned in the Design Impact section, the Piano Pole position tracking system could be utilized for future training or teaching applications. Coaches can use the position data to provide more detailed feedback on how to improve the student's technique, which can help to prevent future injuries. In these ways, the Piano Pole system provides a benefit to society by creating safer coaching techniques. With regard to environmental concerns, the Piano Pole system is meant to be reused many times and is not designed as a one-time-use system. A user has the ability to replace individual components as they begin to degrade over time, which eliminates the need to replace the entire system when one component fails. Finally, the lithium polymer batteries used in the system are rechargeable and therefore create less electronic waste (e-waste) compared to one-time use batteries.

VII. Contribution to ABET Program, LMU Values, and Interdisciplinary Efforts

This capstone project contributes to the LMU core value of "educating the whole person" [39]. This interdisciplinary project, combining electrical engineering and circus arts, expands preconceived notions about how electrical engineering can be applied in the real world. This project works in educating both the technical and artistic capabilities of the team members. Finally, this project will inspire audiences (potentially future students) to think beyond traditional applications of engineering and create their own interdisciplinary work. The project also demonstrates the "creative confidence" described in LMU core value 5 [39]. In developing and implementing our solution to the creative problem of finding new ways for an audience to experience a circus performance, we combined imagination and intellectual curiosity. We developed and enhanced our technical skills, working with new and different devices, such as IMUs, and algorithms, like the Kalman filter. Finally, we fulfilled LMU core value 3 [39] by both committing to lifelong learning ourselves and inspiring others to lifelong learning, even in the context of circus arts. With our Piano Pole system, we inspire the audience to learn about position tracking technologies, as well as different methods of visualization and musicalization.

Through our project, we fulfill each of the ABET student outcomes [40]. Student outcome (1) involves the solution of complex engineering problems by applying principles of engineering, science, and mathematics. In performing IMU position tracking, we are combining physics principles of velocity and acceleration with the calculus topic of numerical integration. The rotation of acceleration vectors uses linear algebra knowledge of matrices applied to quaternion rotation. In addition, in trying to reduce drift, we apply a number of engineering and mathematical principles, such as the use of a Kalman filter. Student outcome (2) stipulates an ability to apply engineering design to meet needs with consideration of safety, global, cultural, social, environmental, and economic factors. We have demonstrated our success in this regard in the Design Impact section. Outcome (3) entails an ability to communicate effectively, which we have done in both our proposal and final presentations, as well as in this report. Outcome (4) describes an ability to recognize ethical and professional responsibilities. We discuss our fulfillment of this outcome in the Ethics Considerations section of this report. Outcome (5) discusses the ability to function effectively on a team, which is a major component of our capstone project. Fosse and Evan have proven their effectiveness as teammates, working collaboratively when necessary to solve problems, while also dividing work between them to handle individually. Outcome (6) defines an ability to conduct appropriate experimentation, analyze data, and use engineering judgement. We have defined the procedures and demonstrated the results for experimentation in the Experimental Test and Demonstration section. We also discuss our data analysis in the Data Analytics section. Student outcome (7) deals with the ability to acquire and apply new knowledge. Accordingly, we have performed research regarding IMUs and position tracking, radio-frequency identification systems, and numerous variants of the Kalman filter, none of which had been encountered by either teammate previously during class instruction.

VIII. Conclusion

The objective of this capstone project was to create a modified version of the Chinese pole circus apparatus in order to artistically musicalize and visualize a circus performer's movement in real time. Utilizing IMU sensors on the performer's hands and feet, the vertical height of the performer was successfully used play corresponding pitches on a musical scale, while the position with respect to the other two dimensions was used to produce a bird's-eye-view visualization. RFID tags were added to the pole to improve the accuracy of the IMU position tracking by providing anchor points with which to recalibrate an IMU's position in space and work to reduce the effect of drift. A Kalman filter was used to further reduce the effects of drift by filtering out noise and allowing the combination of IMU acceleration readings with RFID position readings. The Piano Pole system was tested using a combination of video recordings and data collection. The final music and art pieces produced from the tests were shared with the community. The project thus successfully fulfilled its objective of allowing an audience to experience the movement of a circus performer from a new perspective. The Piano Pole system allows us to rethink the way we view circus arts, and in turn it allows us to rethink the way we view the larger world.

IX. Future Research

While this project accomplished quite a bit, there is still plenty of room for future research.

First, we were unable to make full use of the SparkFun XBee 3 Thing Plus boards. These boards incorporate microcontrollers as well as XBee 3 devices. Thus, all the Qwiic Pro Micro boards can be replaced by Thing Plus boards, eliminating the need for external XBees. Additionally, the Thing Plus boards incorporate JST connectors, so the battery breakout boards would also be unnecessary. The difficulty with switching from the current setup of Qwiic Pro Micro boards to XBee 3 Thing Pluses is that the Thing Plus boards cannot run Arduino code, only MicroPython. This presents a challenge because there is no public MicroPython library for the BNO085 IMU. Thus, in order to use the existing Arduino library with the Thing Plus boards, the entire library would need to be converted to MicroPython. While this was unfeasible for this project, given the time constraints, this could be done in the future to further simplify the setups installed on the gloves and shoes.

Second, a custom sensor fusion algorithm could be produced to integrate the outputs of all four IMUs more closely. While this endeavor is a very challenging one and was outside the scope of this project, a custom fusion algorithm could use the combined raw accelerometer, gyroscope, and magnetometer readings of all four IMUs to better predict the actual linear and angular accelerations experienced. Currently, the project relies on the sensor fusion algorithms built into the BNO085 IMUs individually, but additional accuracy could be obtained by using the raw data from all four.

Finally, future testing will hopefully include data from a real Chinese pole performance, rather than the mock Chinese pole we had to use. We would love to perform a real Chinese pole trial as soon as this becomes a possibility.

X. References

[1] L. Trubat, "E-TRACES," *Behance*, 06-Nov-2014. [Online]. Available: https://www.behance.net/gallery/21108721/E-TRACES. [Accessed: 01-Dec-2020].

[2] L. Trubat, "E-TRACES, memories of dance," *Vimeo*, 06-Oct-2014. [Online]. Available: https://vimeo.com/108109673. [Accessed: 01-Dec-2020].

[3] M. Susman, Makenna Susman Photography, 18-Aug-2020.

[4] S. McFarlin, Scott McFarlin Photography, 27-April-2018. https://scottmcfarlin.com/.

[5] Y. Chen, C. Fu, W.S.W. Leung, and L. Shi, "Drift-Free and Self-Aligned IMU-Based Human Gait Tracking System With Augmented Precision and Robustness," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4671-4678, Jul. 2020, doi: 10.1109/LRA.2020.3002203

[6] G. Zizzo and L. Ren, "Position Tracking During Human Walking Using an Integrated Wearable Sensing System," *Sensors*, vol. 17, no. 12, p. 2866, 2017, doi: 10.3390/s17122866

[7] Q. Yuan and I. M. Chen, "Localization and velocity tracking of human via 3 IMU sensors," *Sensors and Actuators A: Physical*, vol. 212, pp. 25-33, Jun. 2014, doi: 10.1016/j.sna.2014.03.004

[8] S. Ikizoglu, K. Sahin, A. Atas, E. Kara, and T. Cakar, "IMU Acceleration Drift Compensation for Position Tracking in Ambulatory Gait Analysis," *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics*, vol. 1, pp. 582-589, 2017, doi: 10.5220/0006422905820589

[9] "Inertial Measurement Unit (IMU) Use," *Inertial Measurement Unit (IMU) Use - ESE205 Wiki*. [Online]. Available: https://classes.engineering.wustl.edu/ese205/core/index.php?title=Inertial_Measurement_Unit_%28IMU%29_Use. [Accessed: 12-Dec-2020].

[10] *Gyroscope*. [Online]. Available: https://learn.sparkfun.com/tutorials/gyroscope/all. [Accessed: 12-Dec-2020].

[11] P. Corke, "How magnetometers work," *Robot Academy*, 30-Jul-2018. [Online]. Available: https://robotacademy.net.au/lesson/how-magnetometers-work/. [Accessed: 12-Dec-2020].

[12] B. Moyer, "Dead Reckoning Error," *Inside The Iot*. 19-Oct-2018 [Online]. Available at: https://www.insidetheiot.com/accels-and-gyros-better-together-with-gps/dead-reckoning-error/ [Accessed 6 December 2020].

[13] "IEEE 802.15.4-2020 - IEEE Standard for Low-Rate Wireless Networks," IEEE. [Online]. https://standards.ieee.org/standard/802_15_4-2020.html. [Accessed: 5-Dec-2020].

[14] "UM10204: I^{2}C Bus Specification and User Manual," NXP Semiconductors. [Online]. https://www.nxp.com/docs/en/user-guide/UM10204.pdf. [Accessed: 5-Dec-2020].
[15] "EPC UHF Gen2 Air Interface Protocol," GS1. [Online]. https://www.gs1.org/standards/epc-rfid/uhf-air-interface-protocol. [Accessed: 5-Dec-2020].
[16] "USB 2.0 Specification," USB. [Online]. https://usb.org/document-library/usb-20-specification. [Accessed: 5-Dec-2020]
[17] F. Lin-Bianco and E. Mitchell, "piano-pole". [Online]. https://github.com/fosselb/piano-pole.
[18] E. Mitchell, "SparkFun_BNO080_Arduino_Library". [Online]. https://github.com/evanmmitchell/SparkFun_BNO080_Arduino_Library.
[19] "SparkFun_BNO080_Arduino_Library," SparkFun. [Online]. https://github.com/sparkfun/SparkFun_BNO080_Arduino_Library. [Accessed: 1-Dec-2020].
[20] "MIDI.js Soundfonts". [Online] https://github.com/gleitz/midi-js-soundfonts. [Accessed: 20-Apr-2021].
[21] "Arduino Software Downloads," Arduino. [Online]. https://www.arduino.cc/en/software. [Accessed: 20-Apr-2021].
[22] "SparkFun Qwiic RFID Reader Arduino Library," SparkFun. [Online]. https://github.com/sparkfun/SparkFun_Qwiic_RFID_Arduino_Library. [Accessed: 20-Apr-2021].
[23] "Qwiic Pro Micro USB-C (ATmega32U4) Hookup Guide," SparkFun. [Online]. https://learn.sparkfun.com/tutorials/qwiic-pro-micro-usb-c-atmega32u4-hookup-guide#setting-up-arduino. [Accessed: 20-Apr-2021].
[24] "Matplotlib". [Online]. https://pypi.org/project/matplotlib/. [Accessed: 20-Apr-2021].
[25] "NumPy". [Online]. https://pypi.org/project/numpy/. [Accessed: 20-Apr-2021].
[26] "playsound". [Online]. https://pypi.org/project/playsound/. [Accessed: 20-Apr-2021].
[27] "Processing Python". [Online]. https://pypi.org/project/processing-py/. [Accessed: 20-Apr-2021].
[28] "FilterPy". [Online]. https://pypi.org/project/filterpy/. [Accessed: 20-Apr-2021].
[29] "SciPy". [Online]. https://pypi.org/project/scipy/. [Accessed: 20-Apr-2021].
[30] "pySerial". [Online]. https://pypi.org/project/pyserial/. [Accessed: 20-Apr-2021].
[31] "PyPI". [Online]. https://pypi.org/. [Accessed: 20-Apr-2021].
[32] "XCTU," Digi. [Online]. https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu. [Accessed: 20-Apr-2021].
[33] F. Lin-Bianco. "Piano Pole RFID Sensor Hardware Unit Test". [Online Video]. https://www.youtube.com/watch?v=h5cg_xR2Yhw. [Accessed: 30-Apr-2021].
[34] F. Lin-Bianco. "Musical and Visual Elements Unit Test". [Online Video]. https://www.youtube.com/watch?v=QhYEqthtW6g. [Accessed: 30-Apr-2021].
[35] F. Lin-Bianco. "Position Tracking Accuracy Integration Test". [Online Video]. https://www.youtube.com/watch?v=9_jNilTD3cc. [Accessed: 30-Apr-2021].
[36] F. Lin-Bianco. "Piano Pole Latency Test". [Online Video]. https://www.youtube.com/watch?v=D8aBzsL3FZE. [Accessed: 30-Apr-2021].
[37] F. Lin-Bianco. "Piano Pole Integration Tests". [Online Video]. https://www.youtube.com/watch?v=PUX-DIjDF1w. [Accessed: 30-Apr-2021].

[38] "IEEE Code of Ethics," IEEE. [Online]. https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 14-Apr-2021].
[39] "Mission, Vision, & Core Values," LMU. [Online]. https://cba.lmu.edu/about/mission/. [Accessed: 30-Apr-2021].
[40] "Criteria for Accrediting Engineering Programs," ABET. [Online]. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/. [Accessed: 30-Nov-2020].

XI. Appendices

A. Teammate Roles and Responsibilities
B. Project Code
C. Testing Procedures

Appendix A. Teammate Roles and Responsibilities

In the efforts to realize the goals of the Piano Pole system, Evan took the lead on the BNO085 IMU and developing the IMU position tracking algorithm. With regard to the BNO085, this involved reading low-level documentation in order to modify the SparkFun BNO080 IMU Library to add support for calibration and taring. Work on the position tracking algorithm included reading a textbook to understand Kalman filtering and the implementation of a Kalman filter. Evan also contributed the musical component of the system.

Meanwhile, Fosse took the lead on XBee communication and RFID readers and tags. Fosse tested several microcontroller boards, eventually settling on the Qwiic Pro Micro. Additionally, Fosse worked to design and implement the visual elements. He also took the lead on the final assembly of the system, performing the necessary soldering and sewing.

Finally, both team members worked to test the system. Fosse wore the assembled gloves and shoes and performed while Evan managed the recording of the final tests.

Appendix B. Project Code

I. Arduino Program (Qwiic_Pro_Micro.ino)

```
/*
Qwiic_Pro_Micro.ino
Author: Fosse Lin-Bianco and Evan Mitchell
Purpose: Read IMU and RFID data; transmit both via XBee.
*/

#include <SparkFun_BNO085_Arduino_Library.h>
#include <SparkFun_Qwiic_Rfid.h>

#define XBee Serial1

float time;

BNO085 imu;
float linAccelX, linAccelY, linAccelZ;
float quatReal, quatI, quatJ, quatK, quatRadianAccuracy;
byte linAccelAccuracy, quatAccuracy, stability;
String imuString;

Qwiic_Rfid rfid(0x7D);
byte rfidTag;
String rfidString;

void setup() {
  Serial.begin(9600);
  XBee.begin(9600);

  Wire.begin();
  Wire.setClock(400000);  //Increase I2C data rate to 400kHz

  imu.begin();
  imu.enableLinearAccelerometer(5000);  //Send data updates at 200Hz
  imu.enableRotationVector(5000);       //Send data updates at 200Hz
  imu.enableStabilityClassifier(5000);  //Send data updates at 200Hz
  imu.tareAllAxes(TARE_ROTATION_VECTOR);

  rfid.begin();

  imuString =
        "t,linAccelX,linAccelY,linAccelZ,linAccelAccuracy,quatI,quatJ,quatK,quatReal,quatAccuracy,
        quatRadianAccuracy,stabilityClassification,";
  Serial.println(imuString);
}

void loop() {
  if (imu.dataAvailable()) {
    time = millis() / 1000.0;

    linAccelX = imu.getLinAccelX();
    linAccelY = imu.getLinAccelY();
    linAccelZ = imu.getLinAccelZ();
    linAccelAccuracy = imu.getLinAccelAccuracy();

    quatI = imu.getQuatI();
    quatJ = imu.getQuatJ();
    quatK = imu.getQuatK();
    quatReal = imu.getQuatReal();
```

```
    quatAccuracy = imu.getQuatAccuracy();
    quatRadianAccuracy = imu.getQuatRadianAccuracy();

    stability = imu.getStabilityClassification();


    imuString =
      "i,"

      + String(time, 3) + ","

      + String(linAccelX, 4) + ","
      + String(linAccelY, 4) + ","
      + String(linAccelZ, 4) + ","
      + String(linAccelAccuracy) + ","

      + String(quatI, 4) + ","
      + String(quatJ, 4) + ","
      + String(quatK, 4) + ","
      + String(quatReal, 4) + ","
      + String(quatAccuracy) + ","
      + String(quatRadianAccuracy, 4) + ","

      + String(stability) + ",";

    Serial.println(imuString);
    XBee.println(imuString);
  }

  rfidTag = (byte) rfid.getTag().toInt();   //Extract final byte of tag
  if (rfidTag != 0) {
    time = millis() / 1000.0 - rfid.getPrecReqTime();

    rfidString =
      "r,"
      + String(time, 3) + ","
      + String(rfidTag) + ",";

    Serial.println(rfidString);
    XBee.println(rfidString);
  }
}
```

II. Arduino IMU Library (from SparkFun_BNO085_Arduino_Library.cpp)

```
//Sends the command to tare along all axes
void BNO085::tareAllAxes(uint8_t basisVector)
{
	sendTareCommand(TARE_ALL, basisVector);
}

//Sends the command to tare along the Z axis
void BNO085::tareZAxis(uint8_t basisVector)
{
	sendTareCommand(TARE_Z, basisVector);
}

//This tells the BNO085 to tare
//See page 45 of reference manual and Tare Function Document 1000-4045
void BNO085::sendTareCommand(uint8_t axes, uint8_t basisVector)
{
```

```
        /*shtpData[3] = 0; //P0 - 0x00 - Subcommand: Tare Now
        shtpData[4] = 0; //P1 - Bitmap of axes to tare
        shtpData[5] = 0; //P2 - Rotation Vector to use as basis for tare
        shtpData[6] = 0; //P3 - Reserved
        shtpData[7] = 0; //P4 - Reserved
        shtpData[8] = 0; //P5 - Reserved
        shtpData[9] = 0; //P6 - Reserved
        shtpData[10] = 0; //P7 - Reserved
        shtpData[11] = 0; //P8 - Reserved*/

        for (uint8_t x = 3; x < 12; x++) //Clear this section of the shtpData array
                shtpData[x] = 0;

        shtpData[4] = axes;
        shtpData[5] = basisVector;

        //Using this shtpData packet, send a command
        sendCommand(COMMAND_TARE);
}

//This tells the BNO085 to persist the results of the last tare to flash
void BNO085::persistTare()
{
        /*shtpData[3] = 0; //P0 - 0x01 - Subcommand: Persist Tare
                shtpData[4] = 0; //P1 - Reserved
                shtpData[5] = 0; //P2 - Reserved
                shtpData[6] = 0; //P3 - Reserved
                shtpData[7] = 0; //P4 - Reserved
                shtpData[8] = 0; //P5 - Reserved
                shtpData[9] = 0; //P6 - Reserved
                shtpData[10] = 0; //P7 - Reserved
                shtpData[11] = 0; //P8 - Reserved*/

        for (uint8_t x = 3; x < 12; x++) //Clear this section of the shtpData array
                        shtpData[x] = 0;

        shtpData[3] = 0x01;

        //Using this shtpData packet, send a command
        sendCommand(COMMAND_TARE);
}
```

III. Arduino IMU Library Example (Example18-Tare.ino)

```
/*
  Using the BNO085 IMU
  By: Evan Mitchell
  SparkFun Electronics
  Date: December 1st, 2020
  License: This code is public domain but you buy me a beer if you use this and we meet someday (Beerware license).
  Feel like supporting our work? Buy a board from SparkFun!
  https://www.sparkfun.com/products/14586
  This example shows how to tare the sensor. See document 1000-4045.
  It takes about 1ms at 400kHz I2C to read a record from the sensor, but we are polling the sensor continually
  between updates from the sensor. Use the interrupt pin on the BNO085 breakout to avoid polling.
  Hardware Connections:
  Attach the Qwiic Shield to your Arduino/Photon/ESP32 or other
  Plug the sensor onto the shield
  Serial.print it out at 115200 baud to serial monitor.
*/
```

```
#include <Wire.h>

#include "SparkFun_BNO085_Arduino_Library.h"
BNO085 myIMU;
bool tared = false;

void setup()
{
  Serial.begin(115200);
  Serial.println();
  Serial.println("BNO085 Read Example");

  Wire.begin();

  myIMU.begin();

  Wire.setClock(400000); //Increase I2C data rate to 400kHz

  //Enable Rotation Vector output
  myIMU.enableRotationVector(50000); //Send data update every 50ms

  Serial.println(F("Press 't' to tare"));
}

void loop()
{
  if (myIMU.dataAvailable())
  {
    float quatI = myIMU.getQuatI();
    float quatJ = myIMU.getQuatJ();
    float quatK = myIMU.getQuatK();
    float quatReal = myIMU.getQuatReal();
    float quatRadianAccuracy = myIMU.getQuatRadianAccuracy();

    Serial.print(quatI, 2);
    Serial.print(F(","));
    Serial.print(quatJ, 2);
    Serial.print(F(","));
    Serial.print(quatK, 2);
    Serial.print(F(","));
    Serial.print(quatReal, 2);
    Serial.print(F(","));
    Serial.print(quatRadianAccuracy, 2);

    Serial.println();
  }

  if (Serial.available())
  {
    byte incoming = Serial.read();
    if (incoming == 't')
    {
      myIMU.tareAllAxes(TARE_ROTATION_VECTOR); //Tares the rotation vector along all axes
      // myIMU.tareZAxis(TARE_ROTATION_VECTOR); //Tares the rotation vector along the Z-axis

      tared = true;
      Serial.println("Tared. Press 'p' to persist");
      delay(1000);
    }
    else if (incoming == 'p' && tared)
    {
      myIMU.persistTare(); //Persists the results of the last tare to flash

      Serial.println("Saved");
```

```
      delay(1000);
    }
  }
}
```

IV. Python Program (piano_pole.py)

```
# piano_pole.py
# Author: Fosse Lin-Bianco and Evan Mitchell
# Purpose: Visualize and musicalize a Piano Pole performance.
# Usage: `piano_pole.py [-h] [-p PORT | -i INPUT] [-o OUTPUT] [-t] [-r]`
# For sample output: `python piano_pole.py -t -i logs/log7.txt`

import argparse
from contextlib import redirect_stderr, redirect_stdout
from time import time, sleep
import sys

import matplotlib.pyplot as plt
import numpy as np
from playsound import playsound
import processing_py as processing
from filterpy.common import Q_discrete_white_noise
from filterpy.kalman import KalmanFilter
from scipy.linalg import block_diag
from scipy.spatial.transform import Rotation
from serial import Serial, SerialException


HEIGHTS = {244: 1, 38: 1, 227: 1, 204: 1, 178: 2, 181: 2, 190: 2, 16: 2, 108: 100}  # last byte of tag id: height (m)
COLORS = [(255, 0, 0), (255, 127, 0), (255, 255, 0), (0, 255, 0), (0, 0, 255), (75, 0, 130), (148, 0, 211)]
NOTES = ["C3", "D3", "E3", "F3", "G3", "A3", "B3", "C4", "D4", "E4", "F4", "G4", "A4", "B4", "C5"]


class File_Reader:
    def __init__(self, filename):
        with open(filename, "r") as file:
            contents = file.read()
            self.messages = iter(contents.split("<")[1:])

    def read(self):
        address, message = next(self.messages).split(">\n")
        address = int(address)
        message = message.removesuffix("\n\n")
        if message == "E":
            message = None
        return address, message


class XBee_Reader:
    def __init__(self, port):
        self.XBee = Serial(port, 115200)

    def read(self):
        START_BYTE = 0x7E

        byte_num = None
        length = None
        checksum = 0

        while True:
```

```
            if not self.XBee.in_waiting:
                continue

            current = self.XBee.read()
            current = int.from_bytes(current, byteorder="big")

            if current == START_BYTE:
                message = ""
                byte_num = 0
            elif byte_num == None:
                continue

            if byte_num > 2:
                checksum += current

            if byte_num == 2:
                length = (previous << 8) | current
                length += 3  # Include start and length bytes but NOT checksum byte
            elif byte_num == 5:
                address = (previous << 8) | current
            elif byte_num >= 8 and byte_num < length:
                message += chr(current)
            elif byte_num == length:
                message = message.replace("\r", "")
                if (checksum & 0xFF) != 0xFF:
                    message = None
                return address, message

            previous = current
            byte_num += 1


def parse_args():
    parser = argparse.ArgumentParser(description="Visualize and musicalize a Piano Pole performance.")
    exclusive = parser.add_mutually_exclusive_group()
    exclusive.add_argument("-p", "--port", default="COM8", help="serial port to listen to (default COM8)")
    exclusive.add_argument("-i", "--input", help="file to read from")
    parser.add_argument("-o", "--output", help="file to write to")
    parser.add_argument("-t", "--time", help="play back in real time", action="store_true")
    parser.add_argument("-r", "--recording", help="play beep at start for recording purposes", action="store_true")
    return parser.parse_args()

def get_input(args):
    if args.input == None:
        try:
            return XBee_Reader(args.port)
        except SerialException:
            sys.exit("Port '" + args.port + "' not available")
    else:
        try:
            return File_Reader(args.input)
        except FileNotFoundError:
            sys.exit("Input file '" + args.input + "' not found")

def get_output(args):
    if args.output == None:
        return None
    try:
        return open(args.output, "w")
    except:
        sys.exit("Unable to create output file '" + args.output + "'")

def get_next_message(input, output):
    address, message = input.read()
```

```
    if output:
        message_str = message if message != None else "E"
        output.write("<" + str(address) + ">\n" + message_str + "\n\n")
    return address, message

existing = dict()
def get_next_line(input, output):
    line = None
    for address, lines in existing.items():
        if len(lines) > 1:
            line = lines.pop(0)
            break
    while line == None:
        message = None
        while message == None:
            address, message = get_next_message(input, output)
            if message == None:
                existing[address] = []
        lines = message.split("\n")

        try:
            beginning = existing[address].pop(0)
        except (KeyError, IndexError):
            if not (lines[0].startswith("i") or lines[0].startswith("r")):
                lines.pop(0)
        else:
            lines[0] = beginning + lines[0]

        if len(lines) > 0:
            if len(lines) > 1:
                line = lines.pop(0)
            existing[address] = lines
    return address, line

play_times = {}
def get_next_reading(input, output):
    address, line = get_next_line(input, output)
    # print(str(address) + ": " + line)
    data = line.strip(",").split(",")
    reading = {"address": address, "type": data.pop(0)}
    data = np.array([float(datum) for datum in data])
    if reading["type"] == "i":
        t_k = data[0]
        acc_k = data[1:5]
        quat_k = data[5:11]
        stability_k = data[11]
        accr_k = Rotation.from_quat(quat_k[0:4]).apply(acc_k[0:3])
        reading["data"] = [t_k, accr_k, acc_k, quat_k, stability_k]
    elif reading["type"] == "r":
        t_k = data[0]
        tag = data[1]
        height = HEIGHTS[tag]
        reading["data"] = [t_k, height]

    if args.time:
        try:
            diff = play_times[address]
        except KeyError:
            play_times[address] = time() - t_k
        else:
            sleep_time = t_k + diff - time()
            if sleep_time > 0:
                sleep(sleep_time)
```

```
    return reading

def visualize(visualization, color=0, x=None, y=None):
    width = visualization.width
    height = visualization.height
    xyscale = 200
    performer_diameter = 40
    pole_diameter = 100

    if x != None and y != None:
        visualization.fill(*COLORS[color % len(COLORS)])
        visualization.ellipse(width / 2 + x * xyscale, height / 2 + y * xyscale, performer_diameter, performer_diameter)
    visualization.fill(255)
    visualization.ellipse(width / 2, height / 2, pole_diameter, pole_diameter)
    visualization.redraw()

def musicalize(height):
    playsound("piano_samples/" + NOTES[height] + ".mp3", block=False)


if __name__ == "__main__":
    args = parse_args()
    input = get_input(args)
    output = get_output(args)

    with redirect_stderr(None), redirect_stdout(None):
        visualization = processing.App(2000, 2000)
    visualization.background(0)
    visualize(visualization)

    if args.recording:
        sleep(5)
        playsound("piano_samples/beep.mp3", block=False)

    kalman = {}
    t = {}
    pos = {}
    vel = {}
    acc = {}

    try:
        while True:
            reading = None
            while reading == None:
                try:
                    reading = get_next_reading(input, output)
                except (KeyboardInterrupt, StopIteration):
                    raise
                except:
                    pass
            addr = reading["address"]
            type = reading["type"]

            if type == "i":
                t_k, accr_k, acc_k, quat_k, stability_k = reading["data"]
            elif type == "r":
                t_k, height = reading["data"]

            if addr not in kalman.keys():
                kalman[addr] = KalmanFilter(dim_x=9, dim_z=3)
                kalman[addr].P = np.diag([0, 0, 20, 0, 0, 20, 0, 0, 20])

                t[addr] = np.array([t_k])
                pos[addr] = np.zeros((1, 3))
```

```
            vel[addr] = np.zeros((1, 3))
            acc[addr] = np.zeros((1, 3))
            continue

        t[addr] = np.append(t[addr], t_k)
        dt = t[addr][-1] - t[addr][-2]

        F_block = np.array([[1, dt, dt**2/2], [0, 1, dt], [0, 0, 1]])
        if type == "i":
            kalman[addr].R = np.diag([0.5, 0.5, 0.5])
            H_block = np.array([0, 0, 1])
            if stability_k != 4:
                F_block = np.array([[1, 0, 0], [0, 0, 0], [0, 0, 0]])
        elif type == "r":
            kalman[addr].R = np.diag([0.001, 0.001, 0.001])
            H_block = np.array([1, 0, 0])

        kalman[addr].H = block_diag(H_block, H_block, H_block)
        kalman[addr].F = block_diag(F_block, F_block, F_block)
        Q_block = Q_discrete_white_noise(dim=3, dt=dt, var=1)
        kalman[addr].Q = block_diag(Q_block, Q_block, Q_block)

        kalman[addr].predict()
        if type == "i":
            kalman[addr].update(accr_k)
        elif type == "r":
            kalman[addr].update(np.array([0, 0, height]))

        pos_k, vel_k, acc_k = kalman[addr].x.reshape((3, 3), order="F")
        pos[addr] = np.vstack((pos[addr], pos_k))
        vel[addr] = np.vstack((vel[addr], vel_k))
        acc[addr] = np.vstack((acc[addr], acc_k))

        visualize(visualization, addr, pos_k[0], pos_k[1])
        for threshold in range(15):
            # if np.sqrt(pos[addr][-1][0]**2 + pos[addr][-1][1]**2) < 0.5:
            if pos[addr][-2][2] < threshold / 3 and pos[addr][-1][2] >= threshold / 3 and threshold < 14:
                musicalize(threshold + 1)
                break
            elif pos[addr][-2][2] > threshold / 3 and pos[addr][-1][2] <= threshold / 3:
                musicalize(threshold)
                break

except (KeyboardInterrupt, StopIteration):
    for addr in kalman.keys():
        fig = plt.figure()
        # acc_ax = fig.add_subplot(3, 1, 1, projection="3d")
        # acc_ax.set_title("Acceleration Data from System " + str(addr))
        # acc_ax.set_xlabel("X (m/s^2)")
        # acc_ax.set_ylabel("Y (m/s^2)")
        # acc_ax.set_zlabel("Z (m/s^2)")
        # acc_sc = acc_ax.scatter(acc[addr][:,0], acc[addr][:,1], acc[addr][:,2], c=t[addr]-t[addr][0])
        # fig.colorbar(acc_sc, ax=acc_ax)
        # vel_ax = fig.add_subplot(3, 1, 2, projection="3d")
        # vel_ax.set_title("Velocity Data from System " + str(addr))
        # vel_ax.set_xlabel("X (m/s)")
        # vel_ax.set_ylabel("Y (m/s)")
        # vel_ax.set_zlabel("Z (m/s)")
        # vel_sc = vel_ax.scatter(vel[addr][:,0], vel[addr][:,1], vel[addr][:,2], c=t[addr]-t[addr][0])
        # fig.colorbar(vel_sc, ax=vel_ax)
        pos_ax = fig.add_subplot(1, 1, 1, projection="3d")        # pos_ax = fig.add_subplot(3, 1, 3, projection="3d")
        pos_ax.set_title("Position Data from System " + str(addr))
        pos_ax.set_xlabel("X (m)")
        pos_ax.set_ylabel("Y (m)")
```

```
        pos_ax.set_zlabel("Z (m)")
        pos_sc = pos_ax.scatter(pos[addr][:, 0], pos[addr][:, 1], pos[addr][:, 2], c=t[addr] - t[addr][0])
        fig.colorbar(pos_sc, ax=pos_ax)
        plt.show()

finally:
    if output:
        output.close()

    try:
        while True:
            pass
    except:
        visualization.exit()
```

Appendix C. Testing Procedures

I. Unit Testing

Unit testing is a method of testing in which the system is decomposed into distinct units of functionality, each of which are tested individually. Stubs are used to simulate the outputs of other units, allowing each unit to be tested independently of the others. Unit tests are designed to catch problems occurring at the lowest levels of functionality.

A. IMU Position Tracking Unit Tests

Table I. Arduino Serial Communication Unit Test

System to be Tested	Arduino Serial Communication
Test Setup	Testing Equipment: - Arduino IDE - Arduino Qwiic Pro Micro - IMU Data to Collect: - Raw IMU data
Test Plan	1. Run the Arduino program and check that raw IMU data is being printed to the Serial Monitor.
Analysis of Test Results	The test will be successful if raw IMU data is printed to the Serial Monitor.

B. Touch Sensor Unit Tests

Table II. RFID Sensor Hardware Unit Test

System to be Tested	Touch Sensor Hardware
Test Setup	Testing Equipment: - RFID sensor - RFID tag - Arduino IDE - Arduino Qwiic Pro Micro Data to Collect: - Digital outputs on Arduino Serial Monitor
Test Plan	1. Press the RFID tag to the sensor. 2. Check that a corresponding digital output is displayed to the Serial Monitor when the RFID sensor comes in contact with the tag. The digital output should be unique and independent from the other RFID tags.
Analysis of Test Results	The test will be successful if a unique digital output is displayed for each tag scanned.

C. Musical and Visual Elements Unit Test

Table III. Musical and Visual Elements Unit Test

System to be Tested	Musical and Visual Elements
Test Setup	Testing Equipment: - Python Data to Collect: - Music Composition - Visualization
Test Plan	1. Create a stub to simulate position data. 2. Check that the vertical height (Y-axis) of the position data produces a musical note that corresponds to correct note on a piano. 3. Check that the visualization produces a drawing of the X-axis and Z-axis (or a bird's eye view of the performer's movement on the pole)
Analysis of Test Results	The test will be successful if the musical and visual elements match as described in the project objective.

II. Integration Testing

Integration testing involves the verification of larger functional blocks, each of which is made up of several units. These tests are designed to catch problems occurring at higher levels of the system architecture. Bottom-up integration testing is used, meaning that smaller blocks of the system are tested first, building up to the testing of larger blocks.

A. Wireless Communication Integration Tests

Table IV. IMU Wireless Communication Integration Test

System to be Tested	IMU Wireless Communication
Test Setup	Testing Equipment: - XBee modules - XCTU - IMU - RFID reader + tag - Arduino IDE - Arduino Qwiic Pro Micro Data to Collect: - Raw IMU and RFID data from wireless connection
Test Plan	1. Run the Arduino program and check that raw IMU data is being printed to the Serial Monitor.
Analysis of Test Results	The test will be successful if raw IMU data is successfully received wirelessly.

B. IMU Position Tracking Integration Tests

Table V. Position Tracking Accuracy Integration Test

System to be Tested	Position Tracking Accuracy
Test Setup	Testing Equipment: - Video Camera - IMU Position Tracking System Data to Collect: - Final position data values
Test Plan	1. Mark out a ten-foot line in one-foot intervals using tape. 2. Record a 30-second video of the IMU starting from a standstill, moving ten feet along the line, and stopping at the end. 3. Compare the video to the calculated position data to check that the error remains under one foot for the entire test. 4. Repeat steps 2 and 3 starting from a standstill, moving ten feet along the line, moving back ten feet along the line, and stopping at the starting point. 5. Repeat steps 1-3 with a ten-foot square or other shape as desired.
Analysis of Test Results	The test will be successful if the system is able to track the position of the IMU sensor within 1 foot of error for 30 seconds.

Table VI. Position Tracking Recalibration Integration Test

System to be Tested	Position Tracking Recalibration/Correction
Test Setup	Testing Equipment: - IMU Position Tracking System - RFID Sensor System Data to Collect: - Final position data values
Test Plan	1. Tap the RFID tag to the reader on the shoe or glove 2. Demonstrate that the position data updates to the location of the RFID tag that was pressed. 3. Repeat procedure for the remaining touch sensor bands demonstrating each band updates the position for its unique location.
Analysis of Test Results	The test will be successful if the system recalibrates the position data based on the unique touch sensor that was pressed.

Table VII. Combined Position Tracking Accuracy and Recalibration Integration Test

System to be Tested	Combined Position Tracking Accuracy and Recalibration
Test Setup	Testing Equipment: - Video Camera - IMU Position Tracking System - RFID Sensor System

	Data to Collect: - Final position data values
Test Plan	1. With all four IMU units attached to a person, record a video of the person will slowly climb the Chinese pole making sure that each foot and hand presses a touch sensor band every 30 seconds. 2. Compare the video to the calculated position data to check that the error remains under one foot for the entire test.
Analysis of Test Results	The test will be successful if the system is able to track the position of the IMU sensor within 1 foot of error for 2 minutes, pressing a touch sensor every 30 seconds.

C. System Performance Integration Tests

Table VIII. Assembly and Disassembly Integration Test

System to be Tested	Assembly and Disassembly
Test Setup	Testing Equipment: - IMU Position Tracking System - RFID Tags - Timer Data to Collect: - Assembly and disassembly times
Test Plan	1. Start a timer. 2. Install the IMU position tracking and touch sensor systems. Record the assembly time. 3. Uninstall the IMU position tracking and touch sensor systems. Record the disassembly time.
Analysis of Test Results	The test will be successful if the system is able to be assembled in under 20 minutes and disassembled in under 20 minutes.

Table IX. Touch Sensor Protrusion Integration Test

System to be Tested	Touch Sensor Protrusion
Test Setup	Testing Equipment: - RFID tags on mock Chinese pole - Ruler Data to Collect: - Touch sensor protrusion
Test Plan	1. Measure the distance that the touch sensor protrudes from the Chinese pole.
Analysis of Test Results	The test will be successful if the touch sensor does not protrude more than 1 cm from the Chinese pole.

Table X. System Latency Integration Test

System to be Tested	System Latency
Test Setup	Testing Equipment: - Entire Piano Pole System - Video Camera Data to Collect: - Total system latency
Test Plan	1. Set up 2 distinct sounds. The first one to play when the RFID tag is scanned and the second one to play when the software is finished processing the data. 2. Record a video of the RFID tag touching the reader and wait for both sounds to be produced 3. Determine the time between the two sounds using an editing software
Analysis of Test Results	The test will be successful if the system is able to operate in real time with a latency under 1 second.

D. Piano Pole Integration Test

Table XI. Piano Pole Integration Test

System to be Tested	Piano Pole Integration
Test Setup	Testing Equipment: - Entire Piano Pole System - Video Camera Data to Collect: - Piano Pole performance
Test Plan	1. Record a video of a performer performing on the Piano Pole. 2. Check that the visualization displays as expected. 3. Check that the music plays as expected.
Analysis of Test Results	The test will be successful if the system is able to produce a visualization and music that reflect the movements of the performer.