3D Fall Detection and Gait Parameter Identification

Philip Kent Velbis[1], Diego Aguilar[1], Ho Ting Lam[1], Jessica Ngai[1], Alec Maxwell[2], Charlie Sanchez Vidal[2], Professor Zhaoshuo Jiang[2]

[1]Cañada College 4200 Farm Hill Blvd. Redwood City, CA 94061

[2]School of Engineering, San Francisco State University, San Francisco, CA 94132

Abstract

Falls and injuries are inevitable for seniors for over sixty five. Traditional fall detections are good immediate solutions, but they consist of several drawbacks. The objective of our research is to develop a three-dimensional all fall system that will accurately alert the EMS when a sudden event like a sudden fall happens. In addition to that, our system will take preventative actions to facilitate behavior changes before a fall emerges by studying gait characteristics. The proposed system will consist of accelerometers to monitor the floor vibration. From the collected data, and using post processing techniques listed throughout this paper. Fall severity, fall location, and stride length are able to be detected and quantified. Physical experiments were conducted at San Francisco State University to help develop and verify the algorithm being used. The results from these experiments are presented in this paper.

1.0 Introduction

Falls and sudden illnesses are inevitable events for the senior population. There are many consequences to falling, such as broken bones or head injuries. According to the Centers for Disease Control, one in four people over 65 will suffer from a fall [1]. In 2015, the total medical cost for falls totaled more than $50 billion [2]. The United States is aging at a rapid rate due to higher life expectancy. By 2060, 98 million U.S residents will be over 65 and it is overly concerning, as that will make up about one quarter of the total U.S. population. The recognition and response time of falls is a great predictor for the outcome of the patient. Prolonged time before receiving medical help may result in adverse outcomes related to mortality [3].

While there are already a variety of medical alert systems available, their working principles include drawbacks. For example, medical pendants are devices that are activated after a button is pressed; however, if the person is unconscious, they would not be able to operate it. The smart watch is another device that is commonly used. After it detects a fall, followed by little to no movement within a minute, it will contact emergency medical services. However, it relies on battery power and needs to be recharged every 15 hours. That being said, it can contact emergency services even when the person has not fallen. Both of the medical alert systems mentioned above also require the devices to be worn at all times for them to be effective. Another system involves using surveillance cameras to monitor behavior and detect fall events through image processing, but this is resource intensive and invasive for those being monitored. These working prototypes are only immediate solutions, but they are not suitable for the elders as they can be non-intuitive and invasive for the users.

Similarly studying stride characteristics and gait will allow us to evaluate a patient's walking patterns and encouraging them to adopt evidence-based prevention strategies before a serious fall occurs. Low gait speed has been a greater indication of predicting mortality rates. In addition to that, people who suffer from chronic diseases related to stability or haven't' completely healed from a hip injury is likely to suffer from another fall. In recent studies, evaluating the physical performances such as the gait speed allowed clinicians to identify individuals who are vulnerable of falling. They've investigated the association between low gait speeds and the risk of falling by observing elderly living in a small Norwegian municipality. The mean gait speed was 1.0 m/s and the participants that was below 1.0 m/s indicating increased risk of falling. [4] Other basic gait parameters that are frequently studied are velocity, step length, and step frequency.[5] Current gait analysis techniques are being conducted through wearable devices, walking on devices, and visual aids and tools. Wearable devices have a fairly short battery life and computational visual aid extraction can become expensive making these techniques impractical. On the other hand, if we study the vibration of the floor, we can derive some of the basic human gait characteristics.

The objective of our research is to develop a three-dimensional fall system that will accurately alerts the EMS when a sudden event. like a fall, has occurred. In addition to that, our system will take preventative actions to facilitate behavior changes before a fall emerges. The proposed system will use accelerometers to monitor the floor vibration. From the collected data, and using post processing techniques listed in this paper, fall severity, fall location, and stride length are able to be detected and quantified. Physical experiments were conducted at San Francisco State University to help develop verify the algorithm being used. Results from these experiments are presented in this paper.

2.0 Methodology

Methods of how experimental data was post processed is presented in this section. Data Acquisition, Fast Fourier Transform, windowing, and transfer function are defined, and how these methods are combined to estimate fall severity, identify fall location, and extract stride length are explained.

2.1 Data Acquisition

To proceed with our experiment, we needed used Data Acquisition to input all the necessary parameters for our testing. Data Acquisition is one of the main tasks that our group focused on, and without a proper data acquisition, our data would not have produced any desired results, let alone back-calculating force inputs, and computing a transfer function. In the development process of the codes, the flowchart below allows developers to follow a general idea as to how this series of codes will work as seen in Figure 1.

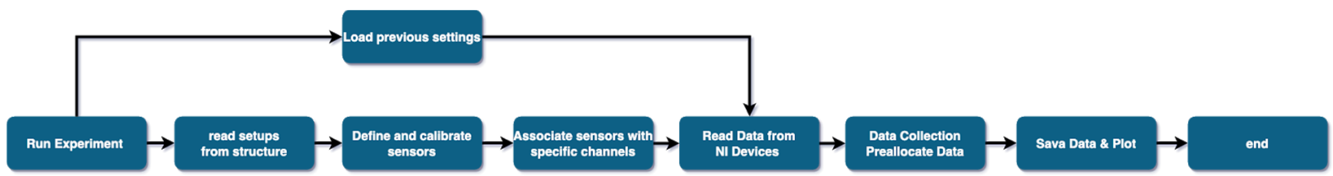


Figure 1: Data Acquisition flowchart.

The two main components of this code development are running the experiments and collecting data, and the primary objectives of this series of codes are:

1. to quantitatively simplify the process of configuring sensors;
2. to reduce any means of unnecessary repetition;
3. to identify channels and associate it with appropriate sensors;
4. to store large quantity of data in a timely fashion;
5. to return any necessary numerical values and assign to particular variables/parameters.

In all the objectives listed above, the most important objective of this code is to allow the computer to efficiently acquire data from and communicate with the DAQ device(s)/ Chassis(es) as listed in the section above.

The procedure of this code is, first, to start running the experiment and a new trial each time with the codes. Then, the codes start to prompt users for inputs, such that it asks for whether to begin the experiment, the experiment types, sensor types, sensor models, sensor series, range, units, sensitivity, bias level, etc. "Run_Experiment" primarily calls other functions while prompting users for inputs to perform certain tasks and collect certain variables and data. The code first asks if user wants to conduct a new trial of experiment, then asks if user(s) wants to use previous setups, or if user wants to conduct another trial using different sensors. After that, "add_new_sensor" will be called to add new sensors to the .json dictionary if the sensor's model and series are not yet registered. Lastly, it prompts user(s) for the type of experiments of either “finite” or “continuous”, calls “setup” for the configuration of the experiment, which then calls the “identify_channel” to do so. All in all, “Run_experiment” is the section of code that retrieves the necessary variables to call other functions and bring all the essential portions of the codes altogether. Based on the user’s preference, the codes will return a series of data as collected within the prompted duration, or until the data collection is over. When the user begins to conduct an experiment using the codes, it will prompt the user(s) for inputs, such that the codes will perform the task accordingly. The codes will not run entirely/smoothly if it runs into an error, even though some entries of sensors’ data will be saved.

2.2 Fast Fourier Transform

We utilized Fast Fourier Transform (FFT) to better understand signals, that we extracted during the data acquisition phase. FFT is a different way to view a time domain signal. The fast Fourier transform is a mathematical algorithm for transforming a signal in the time domain to the frequency domain. Figure 1. below displays Fourier theorem, he proved that any continuous function could be replicated by an infinite sum of sine and cosine waves. Figure 2 shows the output passing the red signal in figure 1 through an FFT. The frequency domain is useful in displaying different forms of frequencies helping us differentiate between noise versus actual forces that are significant to our research.

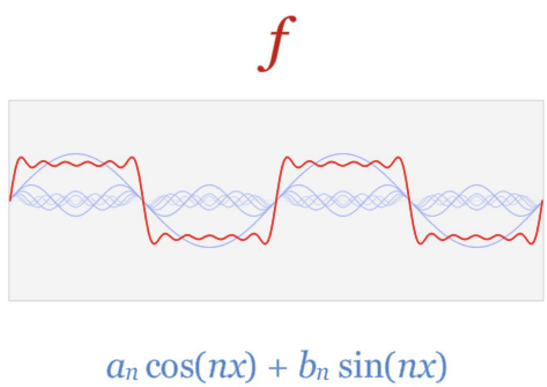
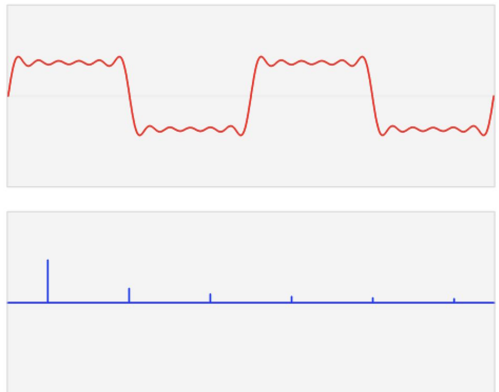


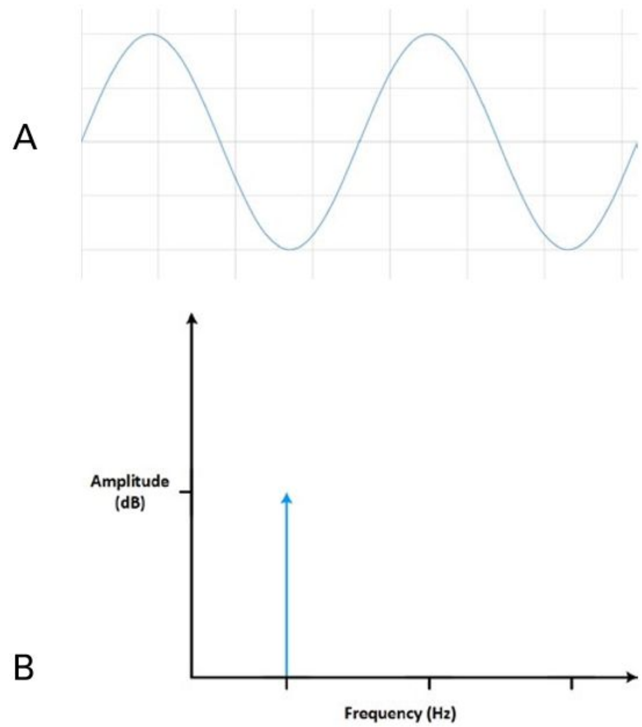
Figure 2: *Blue series represent sinusoidal waves with varying frequency. Series in red is the combination of all of the blue series*



<u>Figure 3:</u> Fourier takes an input function f (in red) in the time domain and converts it into a new function (blue) in the frequency domain.

2.3 Windowing

We use windowing to get a better understanding of a certain signal because of the limitations of FFT. When we use FFT to measure the frequency of a signal, we are comparing the analysis on a finite set of data. The FFT transform assumes the finite data set is a continuous spectrum that is one period of a periodic signal. Both the time domain and frequency domain are circular topologies, so the two endpoints of the time waveform are seen as attached together. When the signal that we measured is periodic and an integer number of periods fill the acquisition time interval, the FFT matches this assumption.



<u>Figure 4:</u> Measuring an integer number of periods (A) gives an ideal FFT (B).

However, if the measured signal is not an integer number of periods, the measured signal may produce a truncated waveform. This waveform will have a different characteristic than the original continuous-time signal, and the finiteness will cause different transition changes into the measured signal making it discontinuous. When the acquisition is not an integer, the endpoints are discontinuous. These discontinuities are present in the FFT as high-frequency components and does not show in the original signal. The spectrum you get by using FFT is not the same spectrum as the original signal, but rather a condensed version. When energy at one frequency leaks into other frequencies, this phenomenon is known as spectral leakage. Fine spectral lines tend to spread out into wider signals.

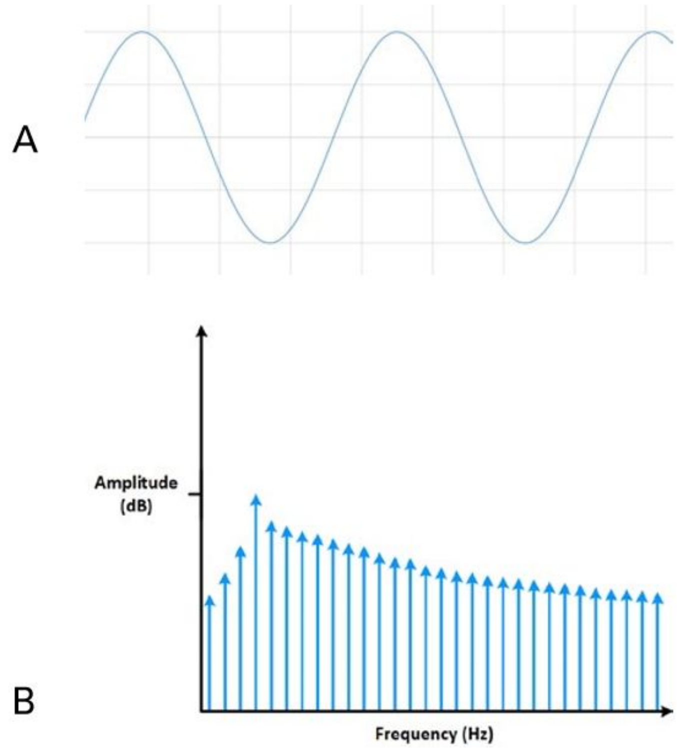
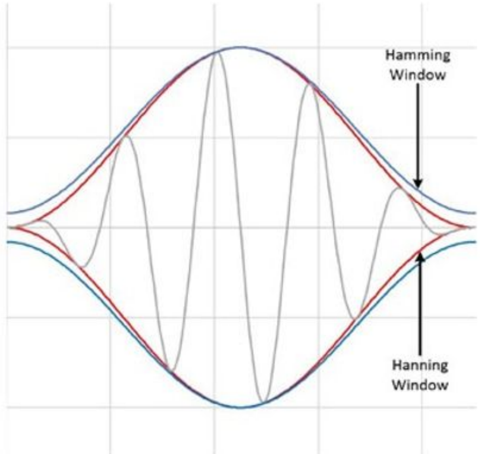
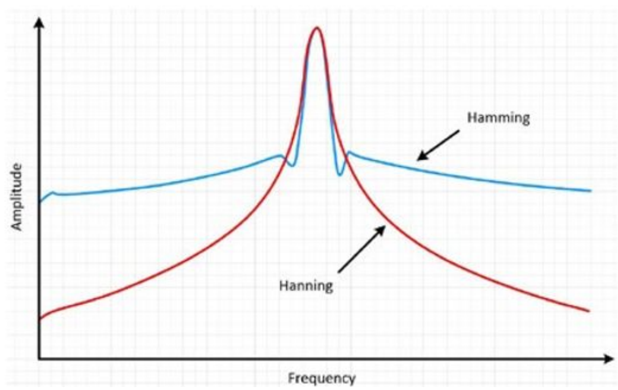


Figure 5: Measuring a non-integer number of periods (A) adds spectral leakage to the FFT (B)

There are a variety of windowing functions are three, but the windowing functions we used are Hamming, Hann and Kaiser-Bessel. The main windowing function we are using is the Kaiser-Bessel window that centers the data in the midpoint. We are applying this to our experiment by centering the data by taking the peak and putting it in the middle which creates a midpoint. This allows us to implement the Kaiser-Bessel to get an accurate results which then we can apply using equation (7). The Hamming and Hann window is used to increase frequency resolution, and reduce spectral leakage. These two windows both have sinusoidal shape, but the Hann window touches zero at both ends eliminating all discontinuities, while the Hamming window doesn't touch zero, but stops at zero meaning that it will have a slight discontinuity.





<u>Figure 6:</u> Hamming and Hann window result in a wide peak, but nice low side lobes.

2.4 Transfer Function

A Transfer Function models a physical system in the frequency domain by algebraically relating the system output to the system input. Since the sensors being used in this research will only measure the floor vibration (output of the system), we need to be able to calculate the input into the system for a given event and estimate the force of that event.

$$TF_{ij} = \frac{Output_{ij}}{Input_i} \qquad (1)$$

Where TF_{ij} represents a portion of the test specimen between a certain location and a certain sensor, $Output_{ij}$ represents the output of the system for a certain sensor, and $Input_i$ represents the input into the system at a particular location. In order to develop a given transfer function the output and input of the system must be represented in the frequency domain, using equation (1) and (2).

$$Output_{ij} = FFT(Acc_{ij}) \tag{2}$$

$$Input_i = FFT(Impact\ Hammer_i) \tag{3}$$

Where Acc_{ij} is the captured response from a given sensor in the time domain, $Impact\ Hammer_i$ is the measured input into the system in the time domain.

2.5 Force Estimation

Before estimating the force of an event, the authors utilize the average transfer function for a specific system. The average transfer function is used to mitigate the "noise" captured when developing individual transfer functions. "Noise" is inherently present in the testing environment, and the authors are assuming that it is random in nature. Therefore, by averaging individual transfer functions random sources of noise will be mitigated and not affect the estimation of forces. Average transfer function is calculated in equation (4).

$$Tf_{Avg_{ij}} = \frac{\sum_{i=1}^{n} Tf_{ij}}{n} \tag{4}$$

The equation $Tf_{Avg_{ij}}$ is the average transfer function where we take the sum of the transfer functions. $\sum_{i=1}^{n}$ is the summation of the calibration trails. The variable Tf_{ij} is the transfer function of a given location on the test specimen and an output sensor. The variable i is the location, j is the sensor, and n is the number of calibration trials used to develop the average transfer function.

$$input_{ij} = \frac{output_j}{Tf_{Avgij}} \tag{5}$$

Equation (5) is used to back calculate the input of a given event. Where the variable $output_j$ is the output of a certain sensor in the frequency domain, and the $input_{ij}$ is the calculated input into the system in the frequency domain. To convert the calculated input from the frequency domain to the time domain the inverse FFT is used, as seen in equation (6).

$$c = corrcoef(input_{ij}) \tag{6}$$

In MATLAB, we use the variable c to calculate the correlation coefficient provided by the force found in the stomp trials. In the stomp trials, we are going to apply the correlation to every single average transfer function that is developed in every single node. Ideally, the correlation will be the highest value when we use the transfer function to identify where exactly a person is stepping. What correlation is doing produces a 3x3 matrix where it compares values to each other. Eventually, we will compare three values giving us a high or low correlation coefficient. A high correlation coefficient means that a sensor receives the most impact acceleration and a low means that it received less. Correlation will give an output of a value between 1 and -1 meaning that if it's closer to 1, the signal is more alike and if it's not close to 1, then it's not going to be the same signal.

2.6 Identify Location

To find the location we will be utilizing a method that is known as Correlated Force Estimates Method. First, a portion of data is obtained based on the maximum amplitude within the data collected from all sensors for an individual trial. This portion contains 50 data points to the left of the maximum, and 500 data points to the right of the maximum. This is due to the method not relying on time and so force estimates will show as peaks across all sensors. The output collected from each sensor is used to estimate the force using the average transfer functions at each location within the test room. After each force is estimated the correlation between each calculated input for a given location is calculated using the Pearson Correlation Method. This method gives us an output of 1 and -1 that tells us which two variables are linearly related. The closer the correlation the closer it is to 1 and if it's -1 it's farther from the correlation. The correlation method used above calculates a correlation coefficient matrix like below:

$$\{L_i\} = \max \begin{bmatrix} 0 & \rho_{xy}(\hat{F}_{i,1}(n), \hat{F}_{i,2}(n)) & \cdots & \rho_{xy}(\hat{F}_{i,1}(n), \hat{F}_{i,j}(n)) \\ & 0 & \cdots & \rho_{xy}(\hat{F}_{i,2}(n), \hat{F}_{i,j}(n)) \\ & & \ddots & \vdots \\ \text{sym.} & & & 0 \end{bmatrix}$$

<u>Figure 7:</u> *Matrix for finding the highest correlation using force estimate [Eq. 5]*

$$L_i = max(max(correlation\ matrix)) \quad (7)$$

The equation L_i takes the maximum value of a matrix in Figure 7 that represents the maximum value of each node calculated in equation (5). Equation (7) is used to calculate the *max(max(correlation matrix)* which calculates the correlation coefficients then it is compared to the location (nodes) of the largest value that tells us the location of the impact. The reason for this approach is to provide redundancy when more sensors are added, and in turn more force estimates. By taking the highest correlation values, error in locating the impact will be reduced. This along with the fact force estimates maintain similar shapes helps increase localization of this robust method.

For our experiment, each input calculated by the force estimate script, a correlation function within MATLAB that will return a 6x6 matrix that corresponds to how well the data matches each sensor. The matrix are the six accelerometers in a square pattern, meaning its rows and columns represent the six accelerometers. Each matrix is then representative of a particular node. The values of each matrix are on a scale from –1 to 1, where –1 is no correlation and 1 is complete correlation. For our research purposes we would like to make sure that each matrix is as close to 1 as possible. This is so we can have a good idea of which node has the most correlation to the input data and therefore identify where the input occurred. With our current working code, we will be able to identify a person's specific gait parameters such as stride length or velocity.

3.0 Data Collection

In the following paragraphs we will talk about the experimental set up and the procedures we performed to collect the necessary data.

3.1 Experiment Setup

We used a room that is 11 ft. by 27 ft. Our room contained some furniture, but it did not move during testing. We used a total of seven sensors: six accelerometers and an impact hammer.

Sensor	Model	Serial Number	Sensitivity
Impact Hammer	PCB 086D05	36630	0.23 $\frac{mV}{lbf}$ ± 15%
Accelerometer 1	PCB 393B31	51836	9.77 $\frac{V}{g}$±5%
Accelerometer 2	PCB 393B31	51835	9.71 $\frac{V}{g}$±5%
Accelerometer 3	PCB 393B31	51820	9.95 $\frac{V}{g}$±5%
Accelerometer 4	PCB 393B31	51815	9.98 $\frac{V}{g}$±5%
Accelerometer 5	PCB 393B31	51819	9.94 $\frac{V}{g}$±5%
Accelerometer 6	PCB 393B31	51814	9.73 $\frac{V}{g}$±5%

Table 1: The list of sensors we used throughout our experiment.

The impact hammer and six accelerometers are all made by PCB Electronics. The purpose of laying these accelerometers is to capture floor vibration at different locations in the test room. At least two accelerometers are needed to identify the location of an event. We are using a 4-slot, USB CompactDAQ-9174 Chassis made by National Instruments. We are using two C Series Sound and Vibration Input Module NI-9234 with 4-Channel per module. Diagram below shows the layout of our room and placements for our accelerometers.

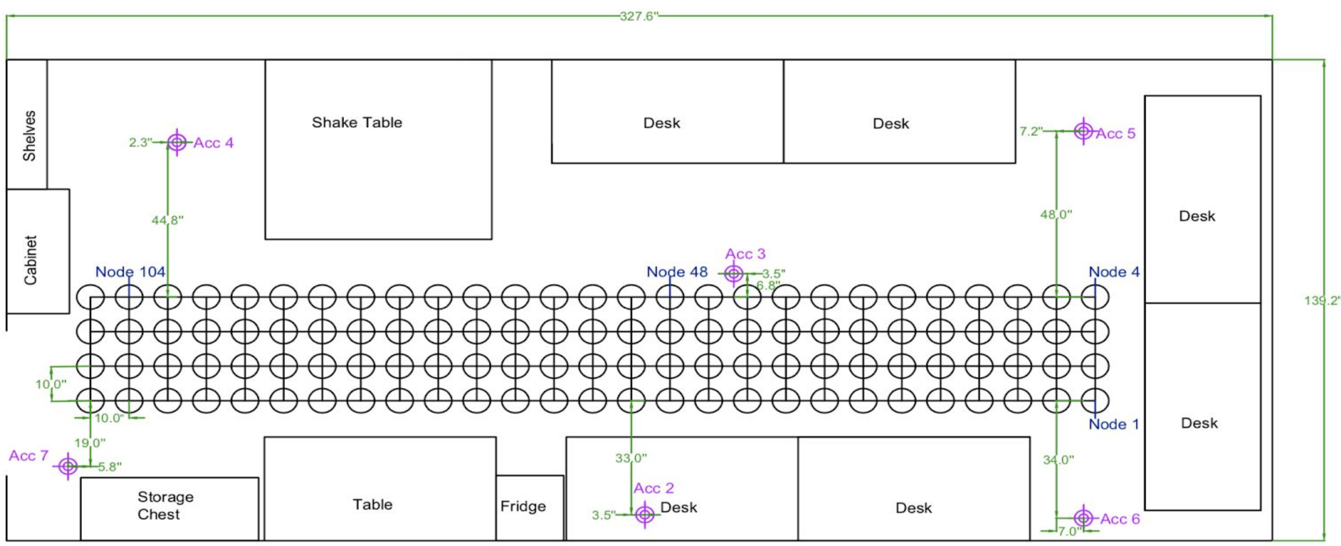


Figure 8: A top down view of the layout of our experiment setup

3.2 Procedure

The layout of grid pattern is a 4 x 27 matrix, which consists of 108 nodes laid in a grid pattern spanning the length of the lab shown in Figure 7. These nodes represent different locations of the floor. In addition to the node layout, the placement of the accelerometers is such that we are able to walk in a straight line and walk randomly while stepping on a node and collect the necessary results while also keeping them at a safe distance. The somewhat random placement of the accelerometers is in part due to the fact that a unique layout will give more unique results and less repetitive results.

In order to develop the transfer functions for each accelerometer in Equation 1. We conducted one type of calibration by using the impact hammer. We hit the impact hammer on each node once per trial, for a total of three runs. When we hit the impact hammer on a node, it records the force in Newtons. This would be the input of our data in equation (3). The accelerometers on the other hand, records the acceleration when the floor vibrates due to the force the impact hammer emits on the floor. This would be the output of our data in equation (2). They send the collected data to the DAQ device to further process the data given in Figure 1. The data we produce will ultimately create the transfer functions that we will us in equation (1). Now that we've developed our transfer functions, we conducted multiple experiments (a total of 791 Runs) such as three variations by hitting the ground using a person's foot. In one trial, we used the toe, another the heel and the final a full foot on the node with a combined trial of 500. We conducted this experiment in order to calculate the average transfer function in equation (4), so then we can use equation (5) to find out how much force each time a person's foot hits the ground. We conducted two more trials: walking randomly while stepping on a node and walking on a straight-line back and forth while stepping on every other node. The data produced by these trials will provide us more force inputs and transfer functions to eventually use the correlation coefficient in equation (6) to find where exactly is person stepping.

When collecting the data, we ensured that the data was as "clean" as we could allow. Meaning we re-did some trials to get data that resemble each other as seen in Figure. An example would be a trial where we hit the node and a person walked by or a chair moved. This would constitute in "noisy" data and a redo would happen. We did this process to hopefully produce good post processing results and data. What we mean by "clean" data is that we ran all trials in a controlled environment, where few uncontrolled variables could cause unintended spikes in data, and that environment was inside the lab.

Calibration Run

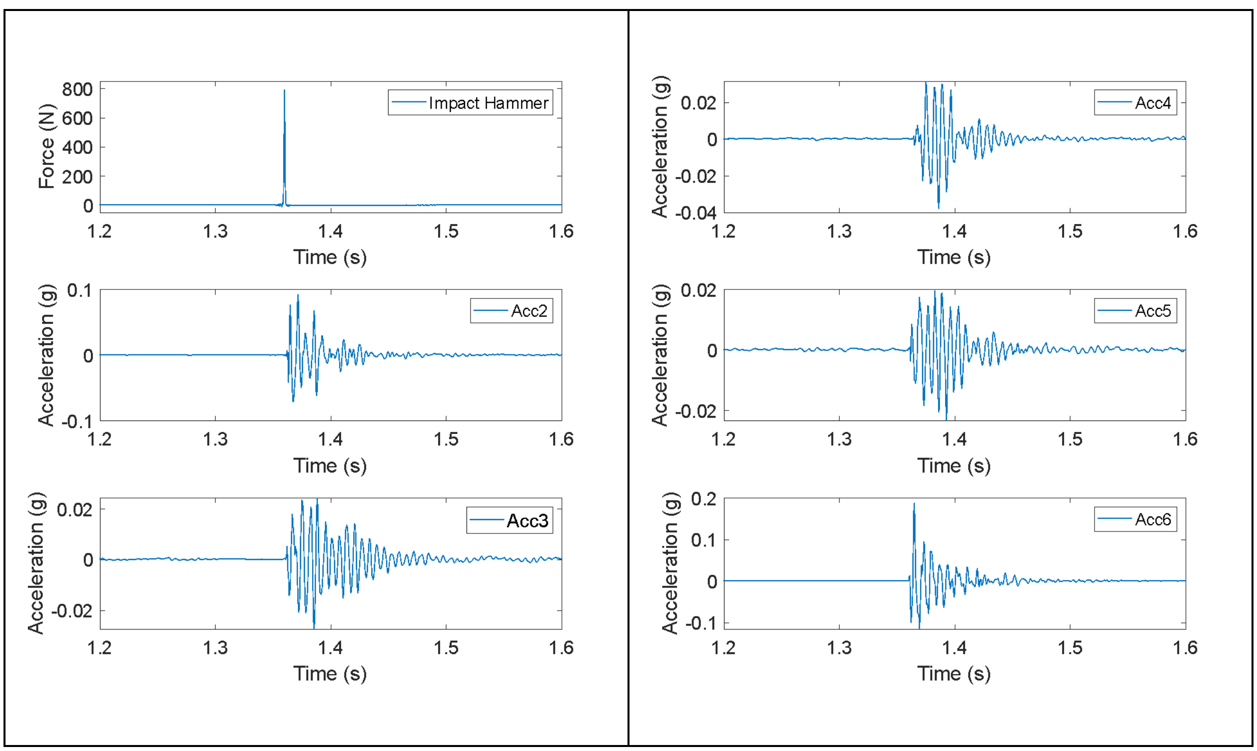


Figure 9: Low noise input and output data from one of our trial runs.

However, the activities outside the lab and in the hallways of the building contribute to the interference of signals. If the interference of the signals is too strong, the particular run would be invalid and thus we would have to rerun the trial. Due to the extremely high sensitivity of the sensors, which allows sensors to pick up signals even beyond the walls and doors, they can detect random noises from outside the lab walls. Such noise could be considered a source of interference, be it a phone dropping, a person walking in the hallway, a person closing a door, or a train passing by. These such vibrations would cause the floor to move just enough for the accelerometers to detect them. The building in which our lab is located, is next to a busy commercial street that has a train line, so there were some variables that caused most of the noise in the data.

Foot trials

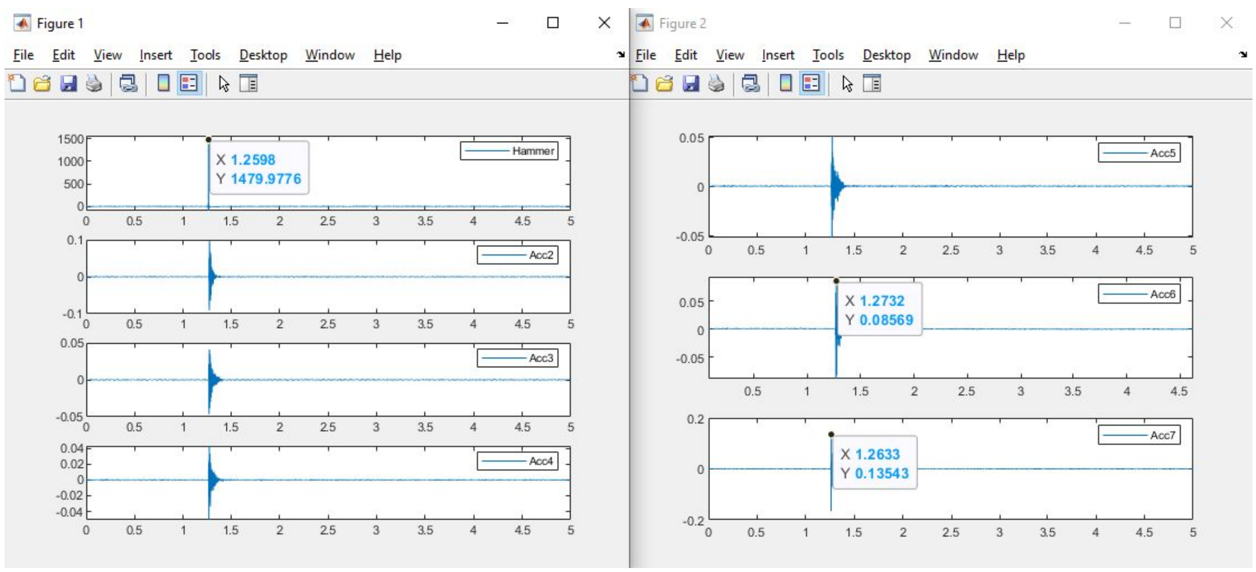


Figure 10: High noise input and output data from one of our trial runs.

3.0 Results

A total of 791 runs has been recorded throughout this experiment. We used 3 trials from the impact hammer and 3 trials from the foot stomps that occurred at node 1 to generalize the entire data set. We discovered that without any windowing function, the statistical measure that associated those transfer functions to the certain nodes, had a fair correlation coefficient. In order to strength that correlation coefficient we decided to apply secondary post processing methods such as hanning, hamming, and kaiser windowing functions.

Trials	383	100.00%
Exact	7	1.83%
Close	22	5.74%
Moderate	135	35.25%
Fail	219	57.18%

Figure 11: Correlation Coefficient of the Force Est. and Transfer Function

We applied the the windowing methods 9 different combinations. Using 50 data plots before the peak and 500 after the peak. The data set tells us that combination 5 had the least variability, however the correlation coefficient was very low. The Kaiser windowing function had higher coefficients and relatively low standard deviations. Proceeding forward we decided to scale down the data points to observe if it provided us with more accuracy.

Trials	32	100.00%
Exact	4	12.50%
Close	13	40.625%
Moderate	14	43.75%
Fail	1	3.125%

Figure 12: Correlation Coefficient of the Force Est. and Transfer Function with portion of the data -50,+50

As a result, we were successful in providing vital information about magnitude of where the fall was occurring.

4.0 Discussion

An area we experienced miscalculations was when it came to the combining of the code, we wrote with the DAQ chassis. Initially, when DAQ chassis is powered up and then immediately ran our code, the data collected would be the supply voltage that the DAQ had used to power itself and not the impact inputs of the sensors. This would return values that we were not expecting and so we would have to restart the data collection process. Another area of miscalculation is the data itself. For our purposes, we ideally wanted the data to be in the center of the length of each experiment, five seconds. We wanted this to be able window a portion of the actual data to better analyze the force of the impact. However, we captured some of the trials of data too early and it resulted in these trials being too close to the beginning to use and were ignored. This is important since for various post-processing techniques, we took 50 data points before and 500 data points after the event as our baseline data. This became problematic for us to use these techniques as some required windowing functions that needed symmetrical data to work properly.

5.0 Conclusion

The development of a sophisticated post-processing algorithm that calculates acceleration amplitudes of the sensors being affected by distance to the impact. This algorithm operates by first calibrating likely fall locations around a node, force and location of an impact. Additionally, sensors do not need to be time-synchronized as frequencies are. This makes the algorithm easier to implement and less costly; computation-wise, to use. Correlation was found to be the best method that gave reasonable results without the aid of windowing. In finding the correlation between sensors, we then extracted the maximum correlation value, and had that represent the node from which it originated. Therefore, the higher the correlation the more likely the fall would take place at the node. This method of the highest correlation value has some promising results, but more experiments are needed to further confirm the effectiveness of the method. The method used to determine the outcome was to evaluate which nodes had the maximum value for each run and compare it with which node the run actually happened at. Then we took the difference between the maximum node and the actual node. If the difference was 0, then the method was exact; if it was 1, then the method was close; if it was between 2 and 8, then the method was off by a larger margin; if it was above 8, then the method did not give the right node. In addition, the use of various post-processing methods became important as to improve the accuracy of the data while keeping the original data. This method was started and the results did get definitely better; the data showed which nodes had the maximum correlation within one node difference. With further extensive research, a compatible method will be chosen for all future experiments to provide accurate and reliable data.

Future Work

Research into modeling human activity from structural vibrations would provide avenues for predicting a condition change of the user, such as the possibility of an oncoming fall. More work into using signal selection, such as manually categorizing more records in the human activity database would provide more insight into the effectiveness of the method. Exploring how to optimize signal preprocessing for better response time and algorithm.

Acknowledgement

In collaboration with San Francisco State University, Cañada College and Skyline College conducted a 10-week summer internship opportunity for community college students. This encourages students to pursue advanced academic work and academia. The summer internship program has proven to be successful in giving students who are underrepresented minority group to achieve higher education and give endless career opportunities in STEM. This summer research internship was supported by the US Department of Education through the Minority Science and Engineering Improvement Program (MSEIP), "Accelerated STEM Pathways through Internships, Research, Engagement, and Support" (ASPIRES), Grant No. P120A150014.

Reference

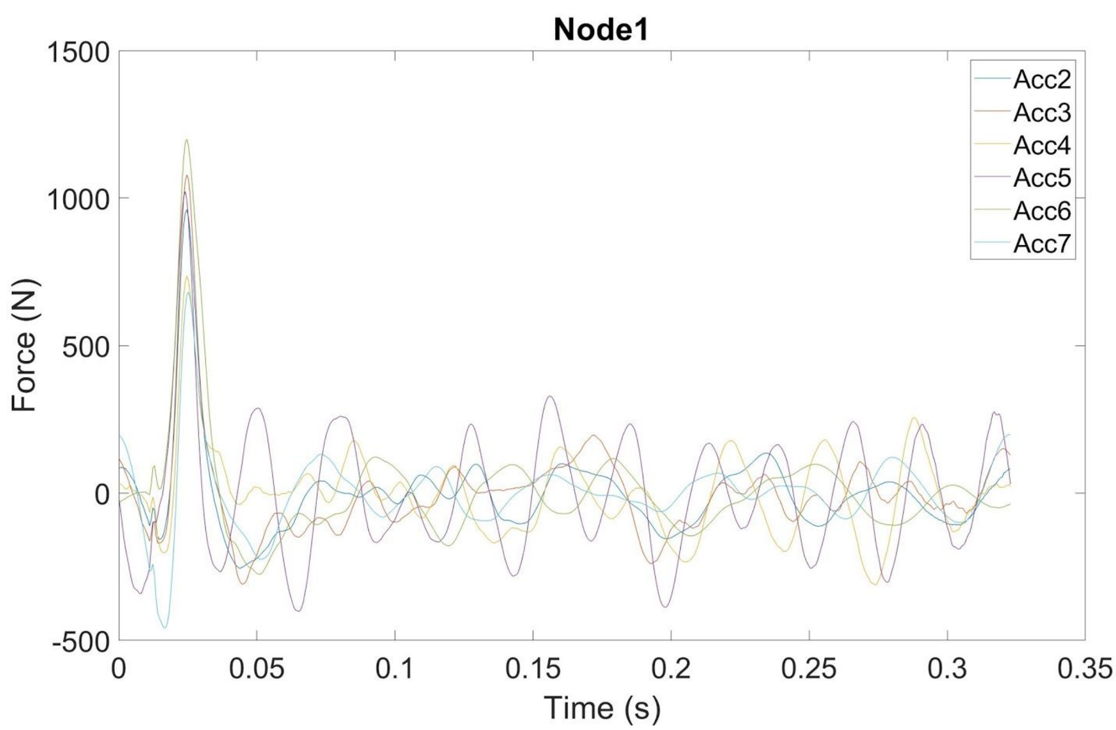
1. Davis, B. T. (2016). Characterization of Human-Induced Vibrations. Retrieved from http://scholarcommons.sc.edu/etd/3770
2. Florence CS., Bergen G., Atherly A., Burns E., Steven J., Drake C. (2018).Medical Costs of Fatal and Nonfatal Falls in Older Adults. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/29512120
3. Kyrdalen IL., Thingstad P., Sandvik L., (2018, September). Associations between gait speed and well-known fall risk factors among community-dwelling older adults. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/30198603
4. Majd A., Siddharth D., Steve K., Robin F. (2003, June). DERIVATION OF BASIC HUMAN GAIT CHARACTERISTICS FROM FLOOR VIBRATIONS. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.618.6014&rep=rep1&type=pdf
5. Patrick P., Declan B., Tomas B., Kathleen B., Bernard S. (2011). Increasing wait times predict increasing mortality for emergency medical admissions. Retrieved from

https://www.researchgate.net/publication/49832900_Increasing_wait_times_predict_increasing_mortality_for_emergency_medical_admissions

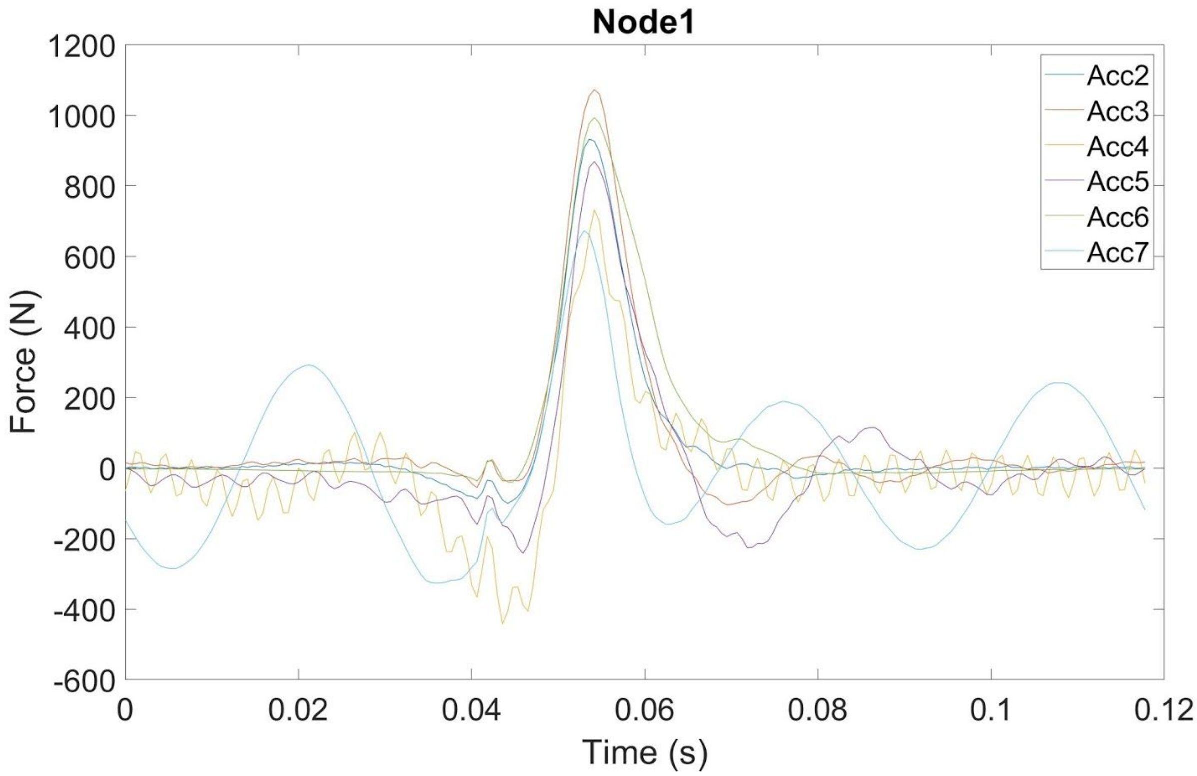
6. Tommy O., Alek K., Kurt O. (n.d.). Basic gait parameters: Reference data for normal subjects, 10-79 years of age. Retrieved from https://pdfs.semanticscholar.org/019d/91af53b19a72030726b2bfae06ddc1686218.pdf
7. (n.d.). Retrieved from https://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html

Appendix

A.



B



Node1
1200
1000
800
600
400
200
0
-200
-400
-600
Force (N)
0
0.02
0.04
0.06
0.08
0.1
0.12
Time (s)
Acc2
Acc3
Acc4
Acc5
Acc6
Acc7