Fall Detection and Mitigation

Alexander Pugh, Andrew Page, Nicholas Gabrisko
University of Illinois at Urbana-Champaign
April 2019

Abstract

This paper analyzes the motion of a mannequin falling, such that the motion is similar to that of a human falling forwards. We utilized three separate sensors attached to the mid-chest, right waist, and right ankle that each detect the accelerations, rotation rate, and magnetic field alignment as the mannequin falls. The goal was to look at the data collected during a fall and pick out portions of the data that could serve as signals that a person was in the early stages of falling. The results showed that analyzing the magnitude of the acceleration was sufficient in noticing when the mannequin would begin falling due to the magnitude beginning to increase as the mannequin accelerated downwards due to gravity. However, this signal could be present in non-falling motions that people perform daily such as sitting down or bending over to pick something up. It then became a priority to be able to differentiate between what was a fall and what was a normal motion in order to prevent fall detectors from reaching the wrong conclusion.

Introduction

On average, an elderly person dies every 19 minutes due to fall-related injuries.[1] Current fall recognition technology attempts to detect a fall and call emergency responders for help. However, these devices are unreliable in that they may give false positives and negatives. For instance, a movement such as sitting down might trigger the alarm, or it might not even register that an actual fall had happened. A reliable system to detect falls in real time and either trigger a prevention system or call for help automatically would save thousands of lives a year.

Many detection devices determine a fall has happened once the victim has already hit the ground. A common, and simplest, example of a device like this is a button necklace that the victim wears at all times.[2] If the victim falls, it detects them hitting the ground and then staying still, and it alerts authorities. Another example is to use less direct methods of detection, such as monitoring Wi-Fi or water usage in homes, and alert authorities when there is a significant change in a person's routine.[3] For both of these very common designs, the idea is the same: after the device believes that a fall has occurred, it will send an alert to whatever protection system has been included. This concept is helpful for rescuing people from potentially lethal situations after they happen, but more can be done.

The goal for this project is to gather and interpret data that can be used to make a fall detector that serves as minimally intrusive into a user's life. This device would search for indicators of a fall while it's happening. Instead of looking for a large upward spike in acceleration or complete stillness for a long period of time, it would search for a steady downward motion determined by the rate of change of the acceleration, rotation rate and orientation with respect to the local magnetic field.

Equipment

The microcontroller that we used to communicate with all of the sensors and devices was an Arduino Mega 2560. It operated at a voltage of 5V which was supplied by an external battery pack. The controller itself had a clock speed of 16 MHz. It communicated with three LSM9DS1 9-Degree of Freedom sensors that calculated the acceleration, rotation rate, and magnetic field alignment. The LSM9DS1 supplies data at 16 bits and has a variety of *g* ranges, ±2/±4/±6/±8/±16 *g,* but ours was operated at ±2 *g* as the accelerations that we would be calculating would be on a smaller scale. The sensor offers a ±4/±8/±12/±16 gauss magnetic full scale, as well as a ±245/±500/±2000 degrees per second (dps) angular rate scale[4]. The sensors were connected to the microcontroller through the SCL and SDA pins and communicated via an I2C connection. We utilized an 8 GB micro-SD card to hold the data that were gathered.

Due to our use of three units of the LSM9DS1, there was the issue of the microcontroller being unable to communicate with three separate sensors that all used the same I2C address. To fix this we utilized TCA9548A I2C multiplexer to allow our device to collect data from each of the three LSM9DS1 devices. In order to control and view what was occuring on our device, we utilized a 16x2 LCD that had its display contrast controlled by a potentiometer and the content of the display controlled by a 4x3 keypad.

All of this was installed on a printed circuit board (PCB) that we encased in a 3D-printed case for protection.

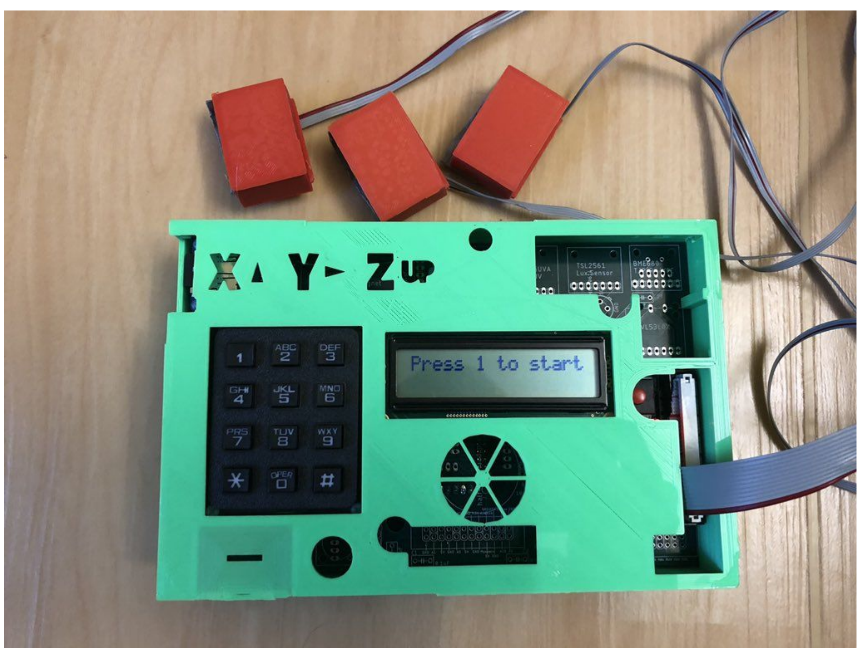

Figure 1: Measurement device used during experiment.

The mannequin used in the experiments had a height of six feet, with mass 12.25 kg and was acquired from the Krannert Center prop shop, which stores older and unused props. A photo of the mannequin with the data acquisition device attached is in Figure 2.

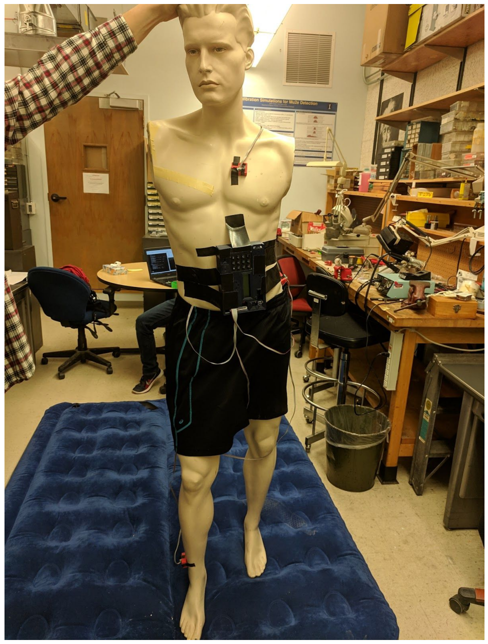

Figure 2: Test mannequin with accelerometers attached.

Data Acquisition

We utilized the Arduino developer design environment to create a program capable of pulling data from three accelerometers and and subsequently writing the data to an 8GB SD card. When creating the program, we wanted to make something that allowed us to begin and stop trials as well as create separate trials with ease. To achieve this, we created code that began a trial with the press of the '1' key of a 9-key keypad and ended that trial with the press of the '2' key. In order to avoid any confusion regarding what the program was doing at any time, we defined certain modes within the code. 'Mode 0' is the initial boot-up stage, where the program scans the device to ensure that there are three LSM9DS1's connected, and an SD card inserted into the card holder. When the '1' key is pressed, the device enters 'Mode 1' and begins writing the accelerometer, gyroscope, and magnetometer data to the SD card at a rate of roughly 30 HZ. 'Mode 1' is active until the '2' key is pressed and the mode is switched to 'Mode 2', which halts the writing of data to the SD card. A subsequent push of the '1' key to enter 'Mode 1' after being in 'Mode 2' increments the file number and opens up a new data file, which is how we were able to perform multiple trials and separate the data between trials. There is also a 'Mode 3' which displays the current and voltage of the battery powering the device.

To increase the ease of use, we wanted to ensure that we knew what the program was doing at any moment that it was on. Using the LCD screen, we would print out the action that the code was performing. For example, when in ‘Mode 0’ the LCD screen reads “Press to Start”, when in ‘Mode 1’ it reads “Collecting to data0.csv” when it is writing to ‘data0’. When ‘Mode 2’ is active the LCD displays “Paused”.

Experiment

When performing test falls, we attached the three accelerometers to the front side of the test mannequin. One accelerometer was attached to the upper chest at a height of 57 inches, one was on the right waist at a height of 44 inches, and one was attached to the right ankle at height of 7 inches. The accelerometers were secured in cases such that there was no movement of the sensors relative to the mannequin to ensure that the sensors’ values mirrored the values of the mannequin. We chose the positioning of the accelerometers to gather data from the upper, middle, and lower portions of the body so that the data gathered could be used to analyze the behavior during a fall at the most extreme positions of the body relative to the center of mass, as well as analyze how the center of mass behaves. We believed that the chest and waist accelerometers would provide the most useful results, as the path that the upper and middle parts of the body travel during a fall is larger than that of an ankle.

In order to obtain visuals for each trial, we would record a video that spanned a time shortly before the mannequin would fall and finished after the mannequin came to rest. To perform a test fall, we oriented the mannequin with its back facing an inflated air mattress to prevent damage to the mannequin or surroundings. We set it backwards due to the offset in the feet, as a forward fall would cause the mannequin to not fall directly to the center of the air mattress as desired. When oriented backwards we were more successful in allowing the mannequin to fall in a path that simulated a person falling directly forward.

To initiate the experiment, we stood the mannequin at the edge of the air mattress and tilted slightly such that the mannequin would begin to fall as soon as we stopped supporting it. To begin a data trial, we would say the name of the trial on video and press the ‘1’ key to initiate the writing of data to the SD card. The initial state of the mannequin is shown in Figure 3. After initializing a trial run, the mannequin would be allowed to fall “naturally”, without any external force applied from the group. After being allowed to go through the falling motion (Fig. 4), the mannequin would then impact the air mattress (Fig. 5). After impact, the mannequin would then be allowed to let its kinetic energy dissipate on the following bounces. When the mannequin reaches rest again we would press ‘2’ and end the trial.

We additionally attached our device to one of the group members and had them perform common actions in order to compare the data to that of the mannequin falling. The sensors were attached in the same places as they were during the mannequin trials in order to contrast data directly from each body part. We wanted to analyze motions that could possibly be mistaken for a fall strictly by looking at the data, so the group member performed actions including sitting up and down, bending over, and going up stairs. The experimental procedures were otherwise the exact same as the mannequin trials.

Figure 3: Test Mannequin in initial state immediately before fall begins.

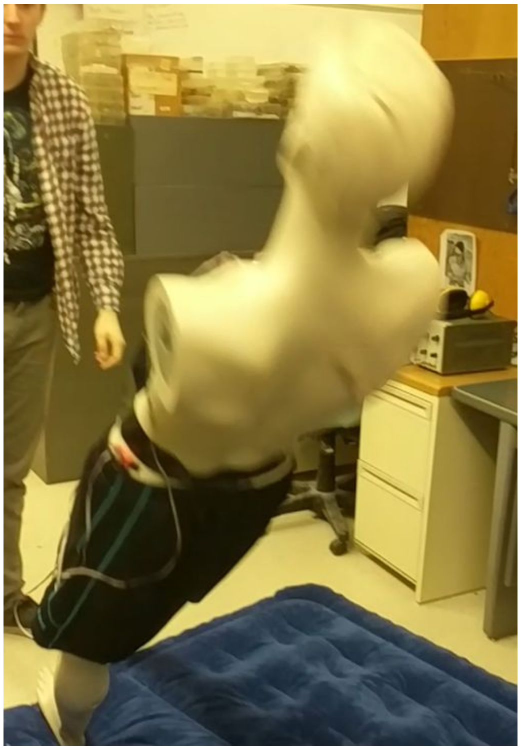

Figure 4: Mannequin in falling motion.

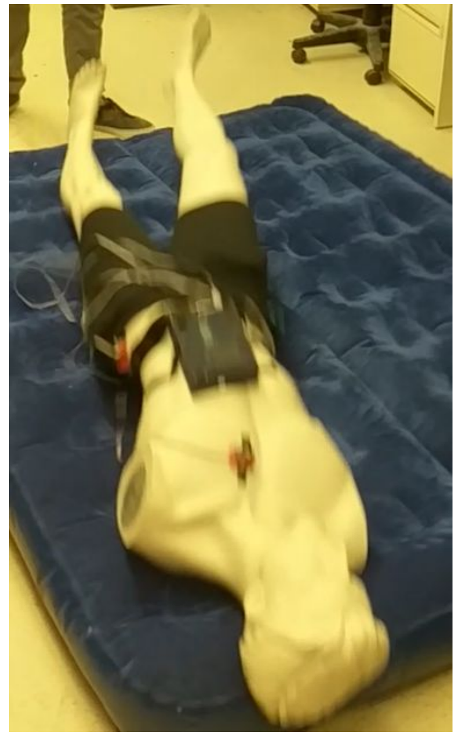

Figure 5: Mannequin first impacts the air mattress.

Offline Analysis

To visualize the results, the '.csv' files that held the LSM9DS2 data were loaded into arrays in Spyder Python. We stored the values for the 'x', 'y', and 'z' acceleration, rotation rate, and magnetic field alignment in arrays that held the values for each time increment that the sensors data was written to the SD card. To create the time value array, we utilized the values that the Arduino on-board clock would give us and subtract the initial time value from each of the data arrays in order to have the data trial begin at $t = 0$.

When graphing the data, we defined a function 'plotData2', which created each array previously mentioned as well as an array that holds the values for the magnitude of the acceleration, $a = \sqrt{a_x^2 + a_y^2 + a_z^2}$. We then create twelve subplots that can plot the acceleration, rotation rate, magnetic field alignment, and magnitude of acceleration versus the time array. To speed up the process, we used a 'for' loop to perform this process for each file that we store in the working directory and create the graphs for each trial fall that we performed.

Beyond 'plotData2', we created numerous functions to plot different aspects of the data, such as the magnitude of the gyroscope data, $\omega = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$, which will help us analyze the rotation of the body during falls, or the magnitude of the accelerometer data after performing various spatial transformations on it and comparing it with a simple physical model.

The goal of the offline analysis is to characterize a fall before the victim hits the ground, based on data gathered in previous tests. To do this, the data were converted into a more regular and usable form using coordinate geometry and compared to a model, where the model is based on the approximate angle of the mannequin from the vertical at any point in time. A function was created to accumulate the difference between a small portion of the data and its corresponding points of a predictive model based off of a plank rotating around a fixed pivot, which will be explored more in depth in the next section.

Results and Analysis

The sensors gather data of three different values: acceleration, rotation rate, and magnetic field for orientation. Generally, the acceleration yields the most relevant and useful information for fall detection. The gyroscope is useful for determining the changing orientation of the sensors over time, which is important for data processing. The magnetometer data can be used to determine orientation, however it can be easily skewed from any magnetic devices nearby. For most of the data analysis, the acceleration will be used to analyze and detect falls, and the rotation rate will be integrated to determine angular orientation.

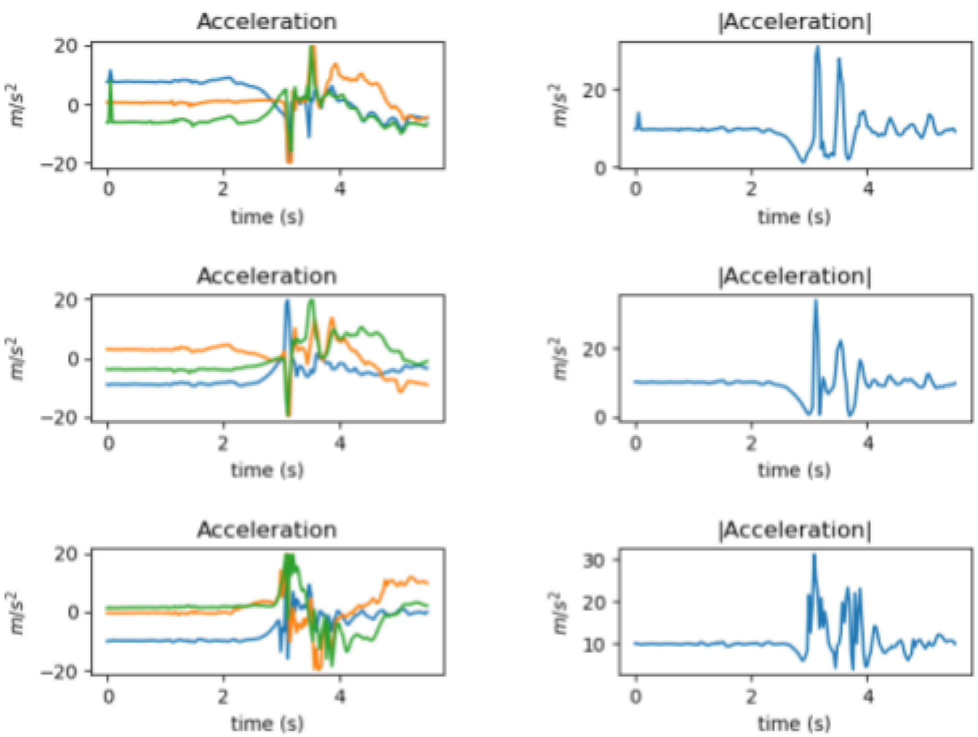

Figure 6: Acceleration and Magnitudes

Figure 6 is a visual example of accelerometer data from a fall. The left column of plots depicts the *x*-, *y*-, and *z*- axis accelerations (blue, orange, and green respectively, as will be used for the remainder of this paper) for each accelerometer. The right column shows the magnitude of the acceleration for the given accelerometer, $a = \sqrt{a_x{}^2 + a_y{}^2 + a_z{}^2}$ where the subscripts indicate that respective coordinate of acceleration in the rest frame of the accelerometer. The top row is the data gathered by the chest accelerometer, the second row is the left hip, and the third is from the ankle.

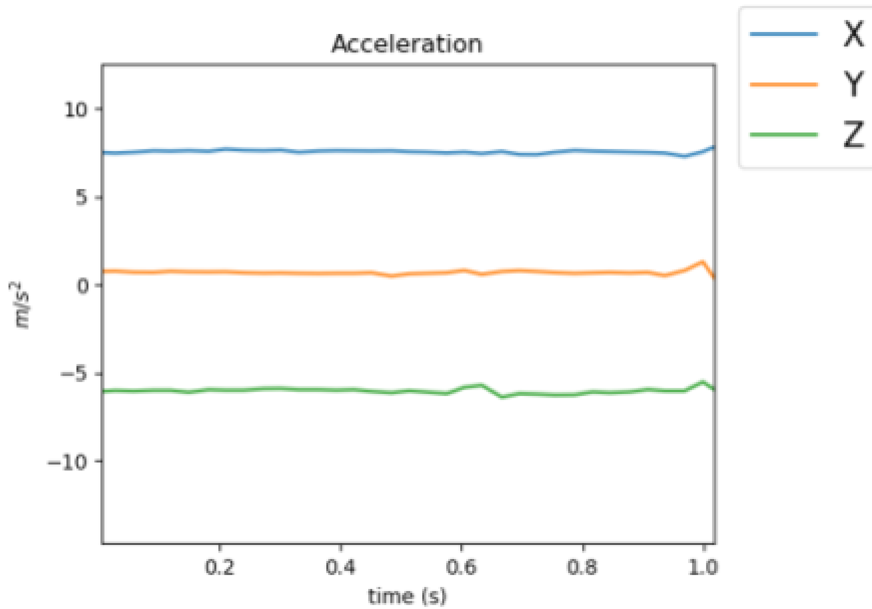

Figure 7: Chest Acceleration before push

This data shows strange results upon first inspection. Figure 7 is an image of the acceleration measured by the chest accelerometer, before the fall occurs. This plot shows that the measured acceleration in the x- direction is approximately 7.5 m/s^2 , while the measured z-direction acceleration is approximately -6 m/s^2 . The only acceleration that is present at that moment is due to the gravitational field, which is approximately -9.8 m/s^2 in the z- direction. If the accelerometer were reading the accelerations as one would expect, the x- acceleration component should read zero, while the z- component should read -*g*. This occurs because the accelerometer isn't perfectly aligned with the coordinate frame in the lab. The chest accelerometer coordinate axes will be referred to as $x_b, y_b,$ and z_b , where the *b* stands for the body frame coordinates. The lab coordinate axes will be defined as *x,y,* and *z.* Note that x_b, y_b and z_b are measured in a left-handed coordinate frame.

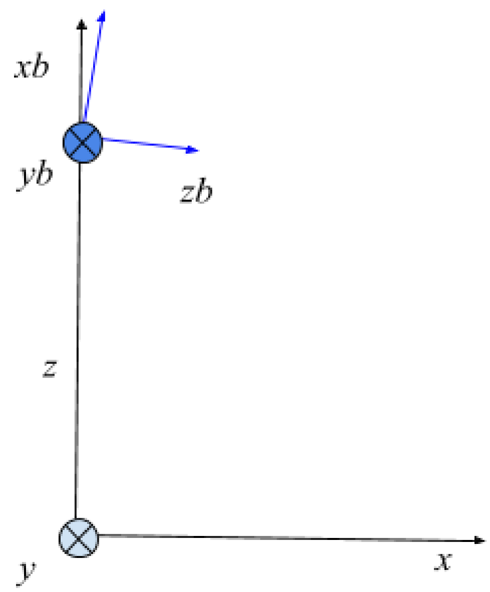

Figure 8: Coordinate Diagram

Figure 8 shows that, from the experimental setup, initial values of x_b roughly correspond to the z- direction in the lab frame, and that of z_b roughly correspond to the x- direction of the lab frame. They are roughly equivalent because the accelerometer is also tilted by some angle about y_b. Figure 7 shows that the acceleration due to gravity is measured partially in x_b and in z_b, when it should be completely measured in x_b (because x_b corresponds to the lab frame's z).To correct this, the accelerometer coordinate frame can be rotated about y_b by an angle specified by the calibration data taken early in the run.

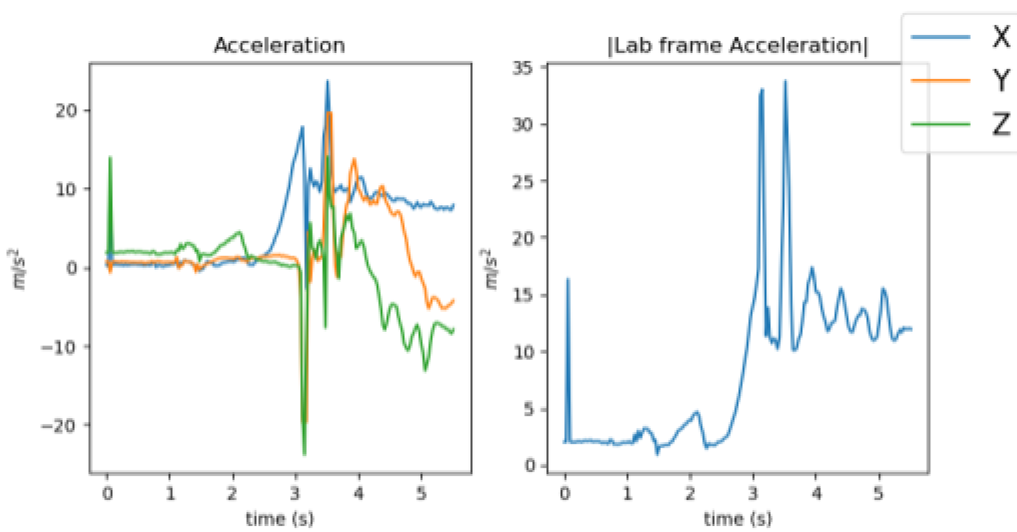

Figure 9: Corrected Acceleration and Magnitude

Figure 9 shows a corrected fall, depicting the accelerations in the lab frame. The acceleration of gravity is measured as -*g* in the $-x_b$ direction and zero in the other directions, which implies that it is standing still in the Earth's gravitational field, vertically oriented. The acceleration magnitude properly increases with time during the fall, occurring between 2 and 3 seconds. The data can now be analyzed.

When analyzing a fall, it is useful to divide it into four sections. First, there is a two second interval in the beginning of each run where the acceleration is relatively constant, which is the time from starting our program to beginning the fall. This was depicted in Figure 7. This segment is used to calibrate the sensors orientation, looking for the precise angle to rotate about y_b to align with the lab frame.

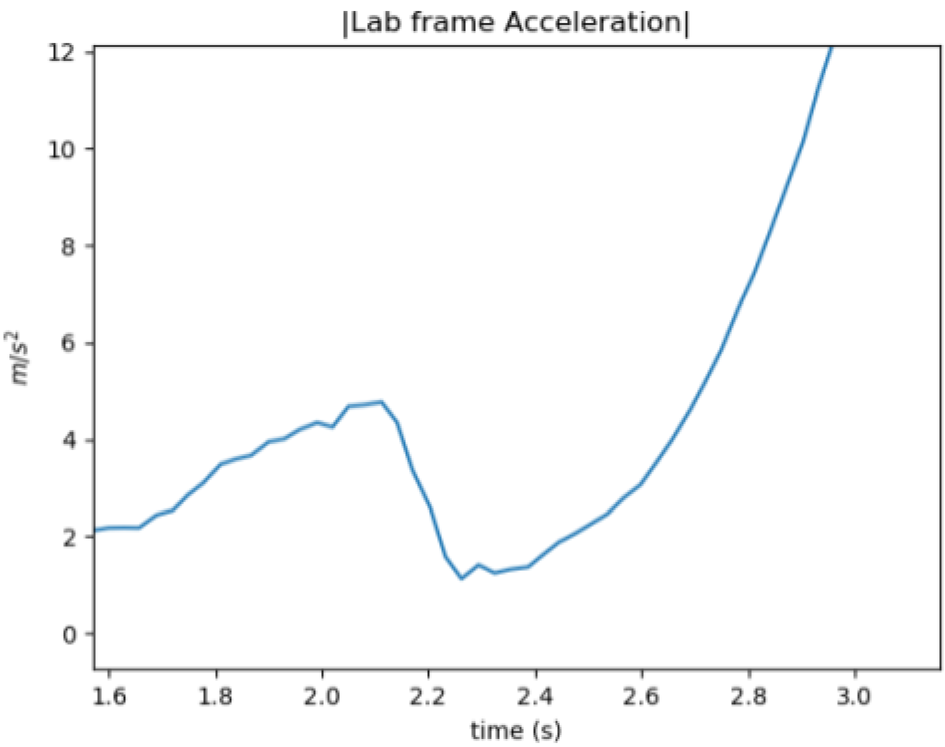

Figure 10: Chest Acceleration Magnitude vs. Time

Figure 10 shows the second section, which is from approximately 1.6 seconds to 3 seconds, when the fall takes place. From 1.6 to 2.0 seconds, there is a slight increase in acceleration, which corresponds to lightly pushing the mannequin to initiate the fall. From 2.0 to 2.2 seconds, the mannequin is tilting over and beginning to accelerate downwards. Starting after 2.2 seconds, the acceleration increases with time, with the curve being convex. The convexity of the curve means that the time-rate of change in the acceleration is increasing. This general curve can be seen in all three magnitude graphs and is a useful signal to alert that a person may be falling.

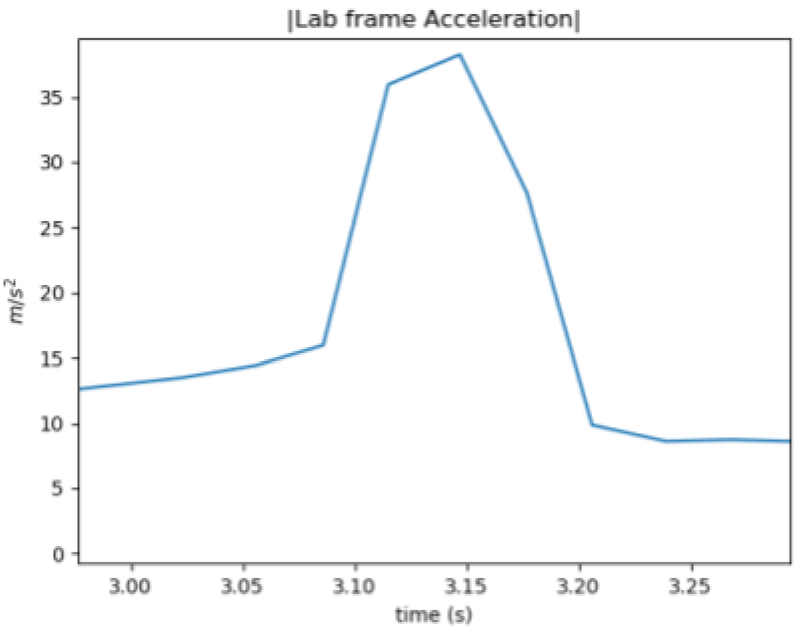

Figure 11: Acceleration Spike

The third region occurs for a brief time. Figure 11 shows a spike in the magnitude of acceleration from the chest sensor at 3.1 seconds, where it reaches a peak of 36.1 m/s^2 . This region, from 3.10 to 3.20 seconds, represents the mannequin impacting the mattress and bouncing upwards. However, one must note that this data is that of a rigid mannequin impacting a surface that allows it to bounce. The fall of a human onto a carpeted or hardwood floor may have a different magnitude of measured acceleration changes, but the large spike due to the impact will still be present. We believe that this acceleration spike is a key feature of the falling process, as reaching a magnitude of 36 m/s^2 in an interval of .2 seconds is behavior that does not occur in everyday life.

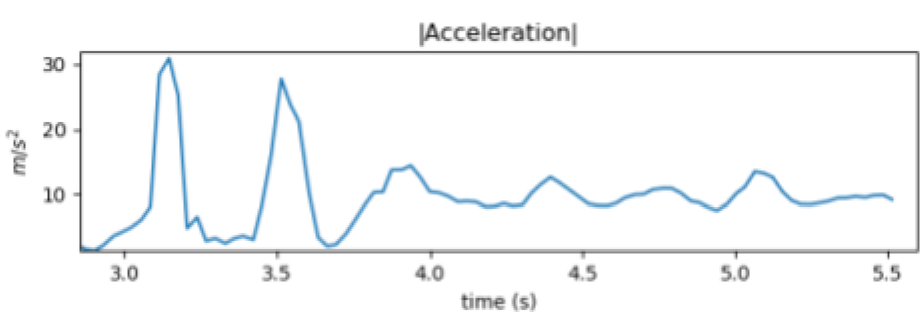

Figure 12: Magnitude of Acceleration After Impact

The fourth section occurs after the first large spike. Figure 12 depicts this period, as the magnitude of the chest acceleration. Since our test subject bounced after each run, there were several spikes after the peak with diminishing magnitude as the total energy of the mannequin dissipated.

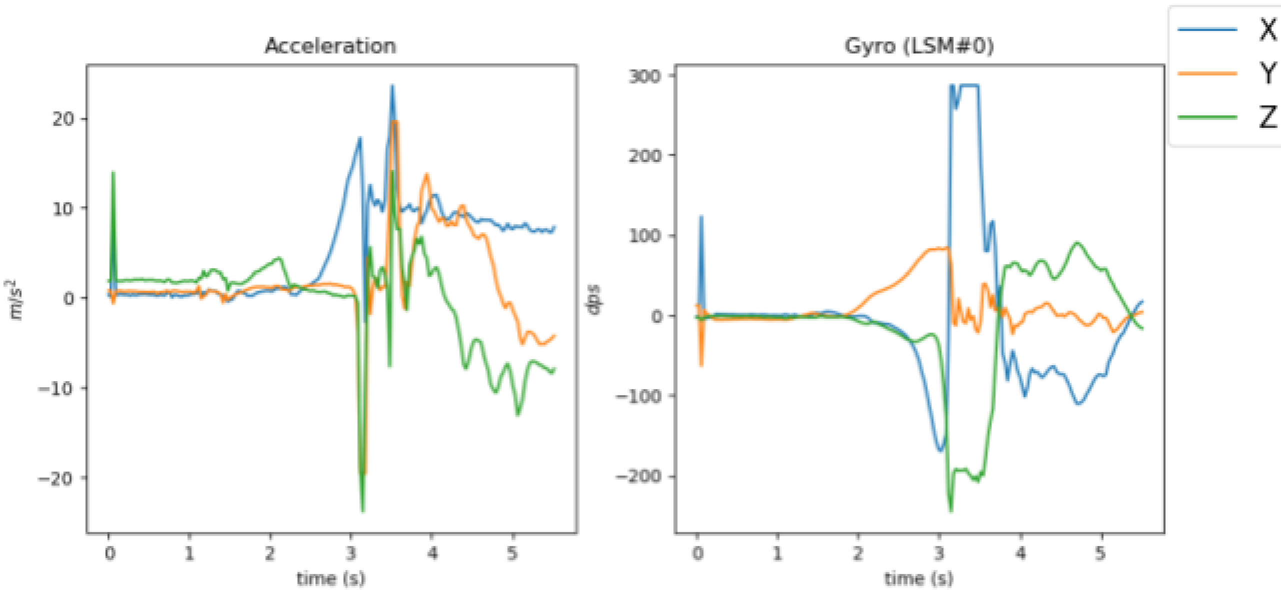

Figure 13: Acceleration and Rotation Rate of the Chest

The other problem with this region is that the mannequin is not in a pose that orients itself symmetrically; its legs were offset, meaning that it it could violently rotate its body after the first bounce. This can be viewed in both the acceleration graphs and the gyroscope graphs, as in Figure 13. It depicts the acceleration and angular velocity, where the *x*-, *y*-, and *z*- components are in blue, orange, and green respectively. Corresponding to the spike in acceleration, there is a sharp and sustained increase in the rotation rate, which indicates that the mannequin spun rapidly. This can be seen in footage of the experiments. During the fall, the rotation rate can be difficult to analyze due to the randomness of each experiment. There is a chance that the mannequin twirls after impact causing cases where the rotation rate peaks the sensor and flatlines the graph, as shown with the blue line in the rotation rate speed in Figure 13.

These previous plots show the usefulness of accelerometer and gyroscope data. The magnetometer, however, is less useful for fall detection. It gives rather chaotic results that are difficult to characterize, and has the potential to give many false-positive fall warnings from strong magnetic fields that can be found in permanent magnets. The field strength when within one foot of a sensor given off by such magnets can be comparable to Earth's magnetic field.[5]

Magnetometer data can be potentially useful when it comes to sorting out accelerations that produce overly chaotic data, such as walking up stairs, bending over, or sitting down. In each graph the top row corresponds to the chest, the second to the hip, and the third to the ankle sensors, and the usual color conventions.

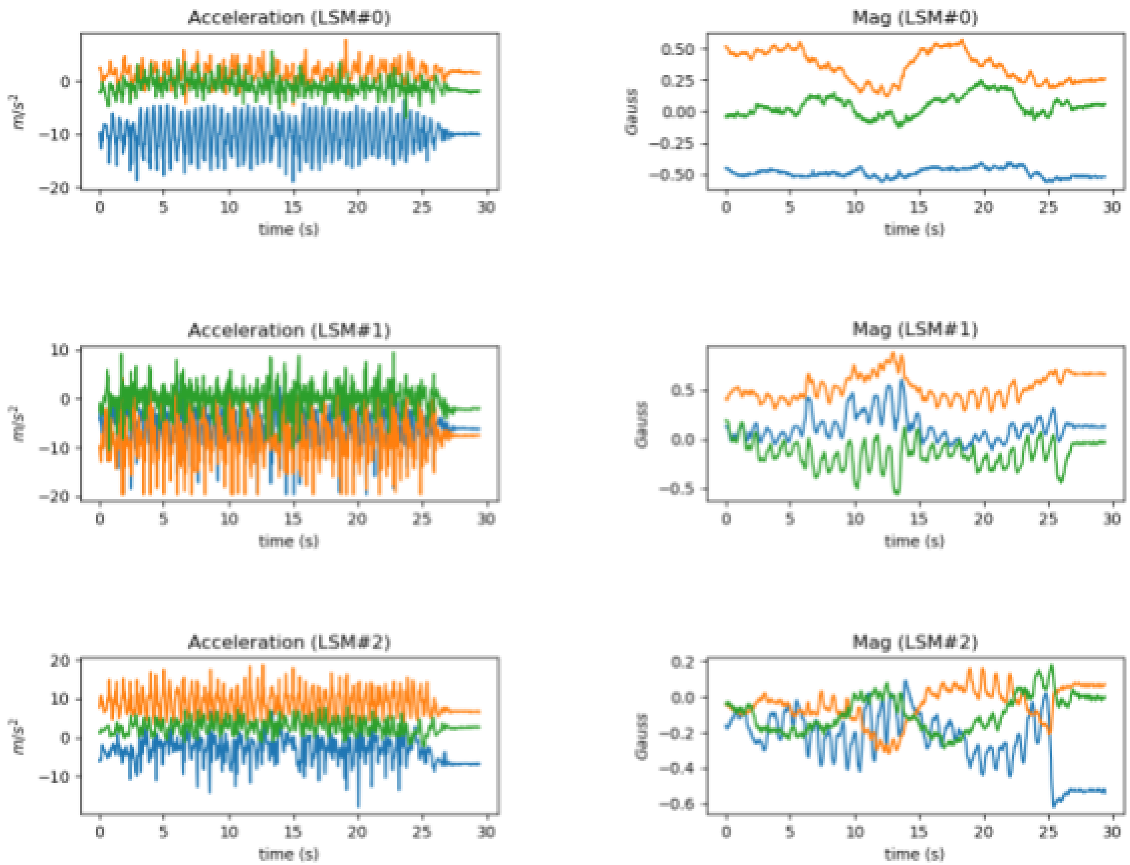

Figure 14: Walking Up Stairs

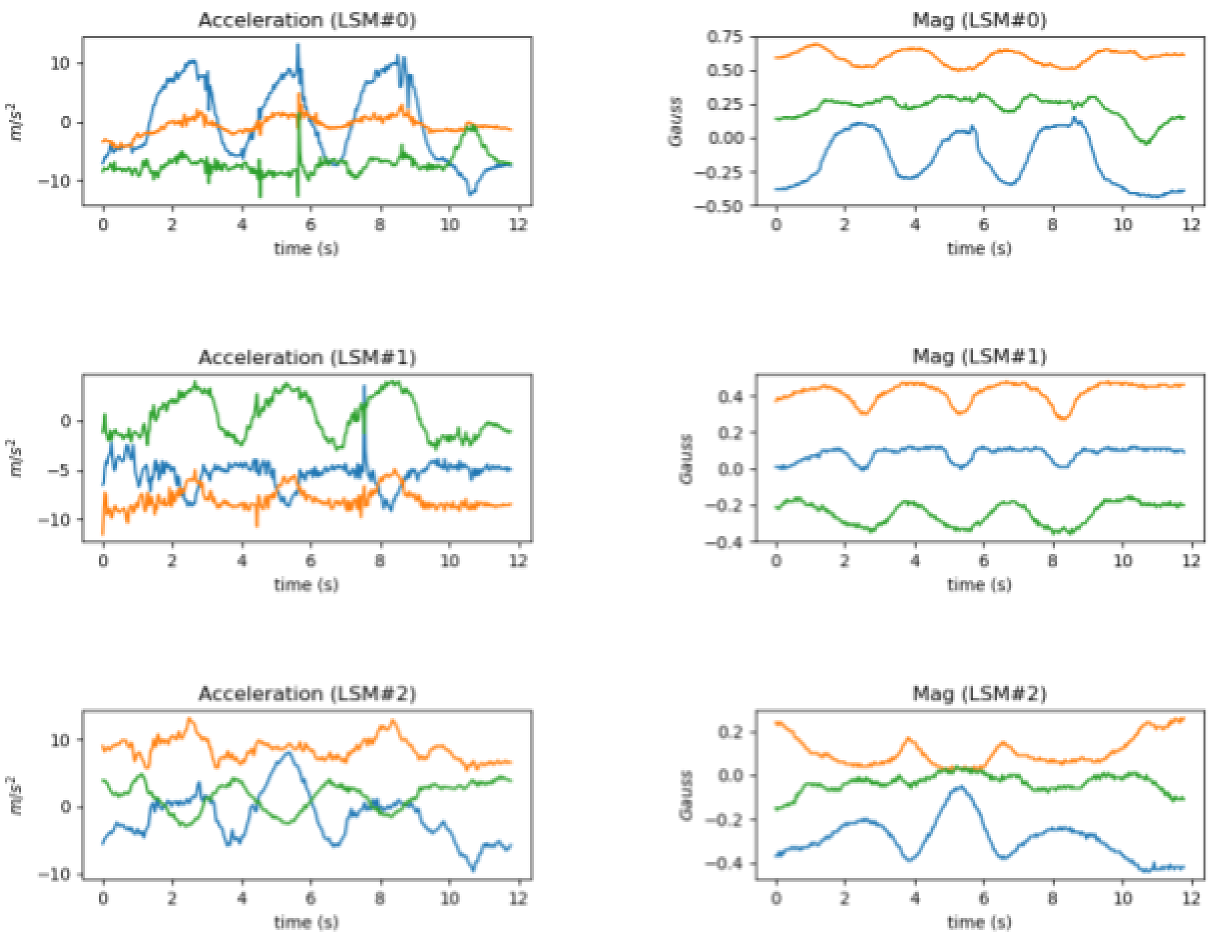

Figure 15: Bending Down and Standing Repeatedly

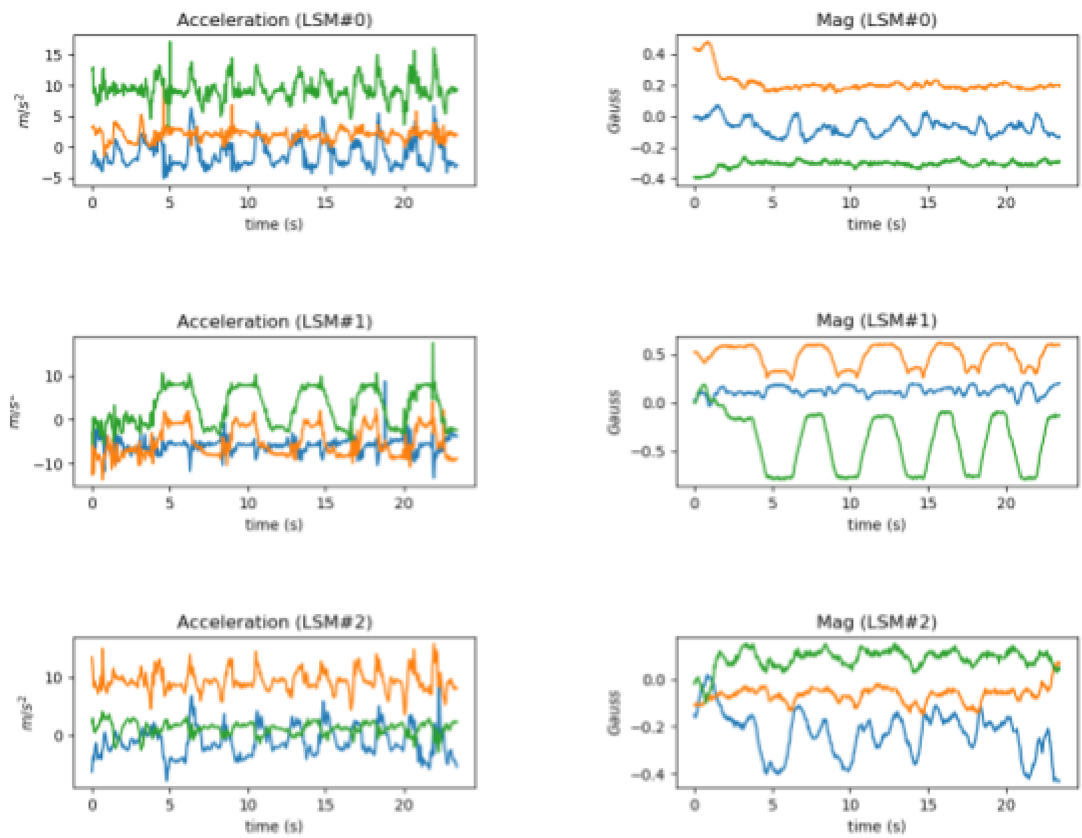

Figure 16: Sitting Down and Standing Repeatedly

Figure 14 specifically shows how walking up the stairs produces acceleration that is hard to manage, with lots of changes occurring each moment. The magnetometer provides data that is much less chaotic and easier to analyze. Figure 15 shows instances of a person bending over, and the magnetometer data . Figure 16 depicts someone sitting down several times, each cycle being one instance of that. In that figure, the second sensor, placed at the hip, gives a great clean set of data because it was placed with its *z*- axis facing up. This means it was consistently aligned with the Earth's magnetic field, so it gave visually satisfying results.

Despite its potential usefulness, though, our project shows that the data from the magnetometer doesn't pose any significant advantages over acceleration. Acceleration and gyroscope data are fully sufficient to detect falls, so magnetometer data was not used in offline analysis.

Discussion

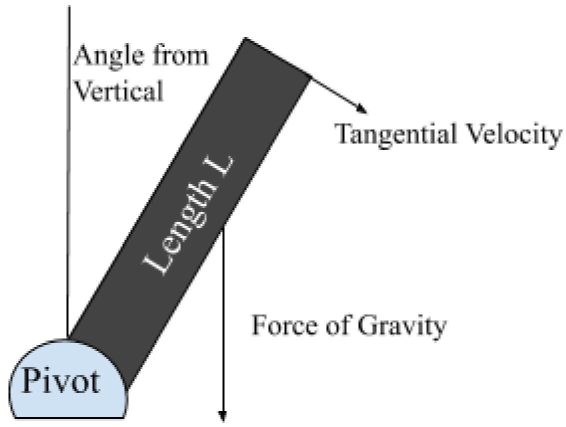

Figure 17: Model

To understand the behavior of a falling person, a useful beginning point is using a simple model. One can imagine a plank attached to the ground at the bottom end, released from a vertical position, as shown in Figure 17. The plank has M, length L, and uniform linear mass density $\rho \equiv \frac{dm}{dr} = \frac{M}{L}$. Its moment of inertia about the pivot point is given by

$$I \equiv \int_0^L r^2 dm = \int_0^L \rho r^2 dr = \int_0^L \frac{M}{L} r^2 dr = \frac{1}{3} ML^2 .$$

The net external torque on the plank acts at the center of mass, located at $r = \frac{L}{2}$. The net external torque is $\Gamma = \frac{mgLsin(\theta)}{2}$ where θ is the angle of the plank with respect to the the vector representing the force of gravity.

This plank is undergoing planar rotation with a known torque, so $\frac{dL}{dt} = \Gamma = I\alpha$ where α is the angular acceleration of the plank and $\boldsymbol{L}$ is its angular momentum (Taylor, 373). Solving for angular acceleration α, one can show that $\alpha = \frac{\Gamma}{I} = \frac{3gsin(\theta)}{2L}$.

Since the plank is rigid, the tangential velocity at any point *r* along the rod can be written as $|v| = \omega r$ (Taylor, 373), implying that $\frac{dv}{dt} = a = \alpha r$. Using this formula, for any point *r* along the rod, $a = r \cdot \frac{3gsin(\theta)}{2L}$, giving the magnitude of the acceleration. This value will vary with the angle from the vertical, increasing as the plank approaches the ground.

The setup of the experiment resembles the assumptions of this model rather closely. The important points to the model are that the plank is attached to a pivot on the ground and that it is relatively uniform. Inspecting the videos, the mannequin's feet are in contact with the floor up until the last moment before contact with the ground, so the pivot is a valid assertion. The mannequin has a mostly uniform density throughout, making the second assumption reasonable.

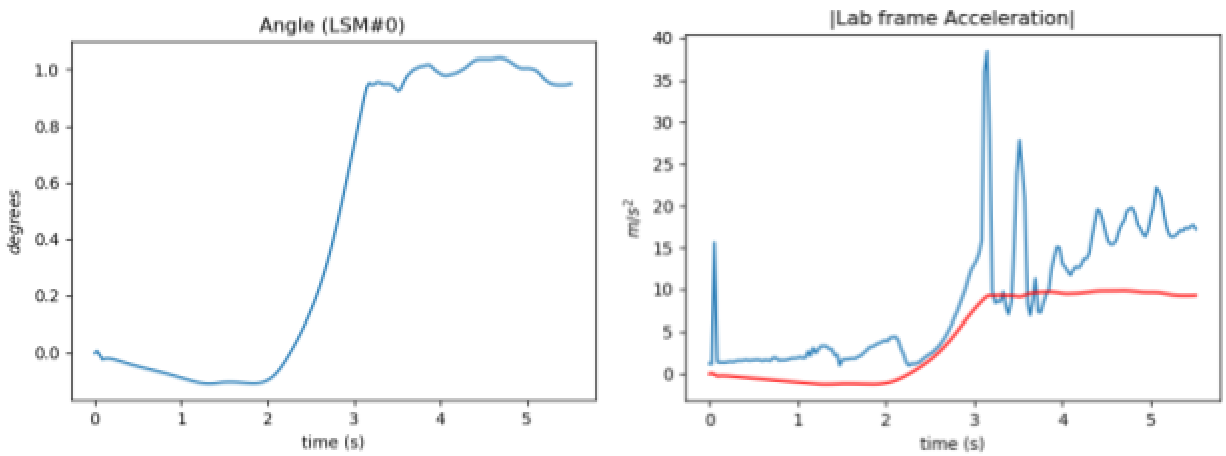

Figure 18: Measured Acceleration and Orientation Against Model

Figure 18 shows the integrated gyroscope readings, approximating the angle from the vertical as used in the model. Using the formula for the magnitude of acceleration a above, a was calculated and plotted for time steps corresponding to the data. The calculated a is shown in the plot to the right, and it can be seen that during the fall, the calculated acceleration value is a good approximation to the data. This can be used to measure how similar incoming data is to a real fall, where the angle is calculated in real time and the fall pattern is assembled and compared to what actual data is flowing in.

This is a rather crude model for a fall, but it can closely resemble the motions of a person who may be walking forward and have their feet get caught on something and trip, while their body rotates around the point where their feet are. In most cases however, the body will not move rigidly and there will be more unpredictable movements that complicate calculations.

Conclusions

One of the largest issues with data interpretation and fall detection is deciding the thresholds that determine a fall. If they are too high, there might be false negatives, meaning a fall went undetected. If they are too low, then there is a great chance that there will be a false positive, meaning something like sitting down will trigger the system. No alert system will ever be perfect, but optimizing the threshold parameters will be crucial to the overall success of fall detection projects.

One of the key findings is that the acceleration is most useful when looking at its magnitude, not its individual components. This is because the accelerometers have set axes that they measure their acceleration from. If there is code attempting to detect a fall based on the components, knowing the orientation of the sensor at every given time is crucial. A reliable way of doing this would be to place the accelerometers in a consistent, known orientation and location. In real situations, a person would have to put on a belt or a special garment to attach these sensors to their body, but this does not guarantee a standard orientation. However, if one inspects the magnitude of the acceleration, the dependence on orientation mostly vanishes, which is why the magnitude is so widely used in this project.

An interesting point is that a simple fall detector can be simple and low-profile, in principle. Each sensor's magnitude of acceleration has similar characteristics, only with slightly differing amplitudes. This means that only one is absolutely necessary to detect a fall, more are extra information and more burden to carry. This is valuable, because making a user wear one accelerometer will be much easier than wearing three or more, and it will be significantly less costly.

The creation of the device and tests have shown that using the Arduino Mega 2560 board is not necessary for this project. The purely necessary components are a battery, one LSM9DS1, a SD card reader (which in itself may be unnecessary, see below) and a microcontroller. To help with the size of the device, a smaller board than the Arduino Mega may be used, such as the Arduino Nano or Feather.

Future Direction

There are two broad goals for the future: gathering more data on human subjects, and creating a prototype detector device. The prototype detector device would be simple, as outlined above. It would be as small as possible while gathering as much data as it can, all while minimizing its power consumption. More precise engineering of the circuit to improve computational and electrical efficiency would be necessary to ensure it is working at all times.

Gathering data on human subjects will enhance the depth of the project greatly. With the current mannequin model, only one type of fall is being detected, and it's a rather artificial one at that. Real falls involve a person rolling, or their legs crumpling, or them hitting something on the way down, and no situation is ever as controlled as it is in the laboratory. Gathering that data and performing more advanced signal processing on it will allow a more generally usable and functional device for future fall victims, potentially saving many lives.

References

1. Fall Prevention Facts. Retrieved from https://www.ncoa.org/news/resources-for-reporters/get-the-facts/falls-prevention-facts/.
2. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6068511/
3. New Advances and Challenges of Fall Detection Systems: A Survey
4. Adafruit LSM9DS1 Info Page. Retrieved from https://learn.adafruit.com/adafruit-lsm9ds1-accelerometer-plus-gyro-plus-magnetometer-9-dof-breakout/overview
5. Taylor, J. R. (2005). *Classical mechanics*. Sausalito, CA: University Science Books.
6. The K&J Magnetic Field Calculator. (n.d.). Retrieved from https://www.kjmagnetics.com/fieldcalculator.asp