Time Synchronization of Smart Wearable Devices For Recording Seismic Activity

Alex Furlanic[1], Philip Thomas[1], Panfilo Armas[1], Rene Parra[1], Zhaoshuo Jiang[2], Jackie Lok[2]
[1]Cañada College, Redwood City, CA 94061 / [2]School of Engineering, San Francisco State University, San Francisco, CA 94132

Abstract

Earthquakes can cause massive destruction and take many lives in a matter of seconds. Efforts are being made to advance our ability to detect earthquakes in order to understand their behavior and minimize casualties. However, that task has not been easy. Large amounts of earthquake records are needed to help us better understand the characteristics of earthquakes, and there are currently very limited sources for obtaining such data. For instance, there are only six functional high-fidelity seismic monitoring stations in the entire state of California, a state with a high risk of earthquakes. Smart wearable devices, with recent advancements of technologies in the sensors they are equipped with and access to the internet, are becoming popular, giving them the potential to detect earthquakes by measuring and recording seismic data. Among many challenges in achieving this goal, time synchronization is a critical factor in ensuring that accurate measurements are recorded. Unsynchronized sensors will result in inaccurate and unreliable data. Methods have been developed for successfully synchronizing wireless devices. This research focuses on developing an appropriate time synchronization procedure suited for smart wearable devices. We test the reliability and accuracy of wearable device sensors compared to traditional high-fidelity sensors. This is done with a variety of excitation tests and test durations. Data is collected and synced using the procedure that we have developed. This research is the first step towards developing a system for detecting earthquakes using smart wearable devices.

I. Introduction

Earthquakes have been detected and recorded using traditional seismic stations that are dug into the ground at various locations all over the world. These seismic stations help seismologists collect and analyze seismic data in order to help understand the behavior of earthquakes. The biggest issue with these seismic stations is that because of their high cost, there is currently a shortage in making them. A potentially more cost-effective addition to seismic sensing is using smart wearable devices such as smartwatches. Smartwatches, just like smartphones, are becoming popular in the general public as well as the tech world. They are equipped with several different types of sensors that have given them the ability to track and record a wide range of information from the wearer, such as the wearer's movements, daily steps, calories burned, and heart-rate. The array of wireless sensors within mobile devices and the extent to which they can be used is a key research topic[1].

Currently there is research being done using the accelerometer sensors within smart devices to help with structural health monitoring of buildings (SHM). In a study by Columbia University they have tested and utilized accelerometers to measure structural vibrations from earthquakes so that they can rapidly assess structural damage and diagnose post-earthquake events[2]. The issue with these devices is that there is still a need to develop a proper method for these sensors to distinguish white noise, such as the vibration of a person's body, from excitations of different intensities in earthquakes. Since we are trying to use smart wearable devices to detect these excitations, we are figuring out a way to create a working network within these sensors so that we can collect accurate data simultaneously from each sensor.

Wireless sensor networks (WSNs) have become increasingly important in recent decades with the connectivity of the internet age. These sensors have a wide variety of applications as has been previously mentioned. However, there are several problems inherent to the use of WSNs. One of the most important has to do with the accurate telling of time by any given sensor. Most devices contain individual clocks, which are just timers that use a crystal oscillator to keep time. Because each sensor has their own individual clock, there is potential for a phenomenon called clock drift. Clock drift refers to the fact that not all clocks have the exact same frequencies as each other. In other words, they do not count time at the same rate, and as time progresses the clocks of two different sensors will drift apart from each other. The phenomenon can be caused by environmental factors such as temperature. Other problems related to time include delays from software and also message loss. It is possible that messages being sent by the sensors can simply not make it to the desired location. All of these problems relating to time result in data being unreadable and meaningless. The solution to these problems is termed time synchronization and it is clear why such synchronization is an important feature in WSNs.

Time synchronization also allows movement, location, and proximity detection. All these sensor networks' goals can be achieved by a process that is formed by four steps: 1) send time, 2) access time, 3) propagation time, and 4) receive time. Send time is when the collected information from either wired or wireless sensors is sent to the master node of the system. Access time is the time it takes the master node to retrieve data from the connected sensor. Propagation time is also referred to as propagation delay because it is classified as the amount of time it takes for the information signal to travel from the sender to the receiver. And last, receive time is the time it takes for the master node, to receive and graph the data received from the sensor, either wired or wireless. The sum of all these times is called the offset between the two nodes. If the offset can be measured then time synchronization can be achieved.

Wireless time synchronization is becoming very popular in today's research topics to the extent that we can now see it being used for many different purposes, such as determining location, proximity, mobility, and energy efficiency, among others. In all sensor networks, their exact location is not known until time synchronization is used. The proximity between sensor nodes is determined when stamped messages are transmitted between the nodes. Energy efficiency is also achieved using time synchronization because it allows the nodes to sleep when not in use and awaken when ready to receive signals. Also, to determine the speed of a moving node, there needs to be common timing between a large number of nodes[3].

II. Methodology

In our research we are focusing on wireless sensors networks. There exist two types of networks, peer-to-peer and master-slave. The peer-to-peer network refers to a network where all sensors are connected with each other. On the other hand we have the master-slave network, which consists of one sensor serving as the reference for the rest of the sensors. The master-slave network can consist of one master sensor and one slave sensor or one master sensor and multiple slave sensors. Both types of wireless sensor networks are currently the most used communication solution in process control and monitoring applications in industrial environments.

This paper introduces a proposed solution for collecting more accurate data from wireless devices. A time synchronization procedure has been developed that can be used by smart wearable devices, which will potentially allow them to detect seismic patterns and collect this data effectively for post-analysis. We test the reliability and accuracy of the sensors in wireless devices versus traditional wired sensors and we test the effectiveness of the time synchronization procedure that we have developed, which is done in post-processing. Time synchronization is an important step to further our research in collecting accurate data from wireless devices because it will help collect more useful data during earthquakes so that we can better understand seismic activity.

We introduce the factors of time synchronization and the process of our procedure, as well as how crucial this step is to collecting data from multiple sensors. We elaborate on Shimmers, the wireless devices that we use in our research which are equipped with various sensors that can also be found in smart wearable devices. We use MATLAB software to control these Shimmers so that we can implement our time synchronization procedure into their data collection process. MATLAB also allows us to use multiple Shimmers simultaneously so that we can create a network of sensors and synchronize their time. We use a shake table to run three different types of tests and collect data from Shimmers.

A. Data Collection and Synchronization

For this research, we use wireless sensors called Shimmers as opposed to smart wearable devices in testing for several reasons. These devices already provide a framework to connect with a PC, allowing us to focus on data collection and synchronization rather than programming another app or framework. We have access to four Shimmer devices and can connect any number of these with a PC over a Bluetooth wireless network. Also, Shimmers contain many of the same sensors that smart wearable devices have and therefore the methods used to synchronize time on these devices should be applicable to smart wearable devices. We refer to Shimmers as sensors, however each Shimmer is actually a wireless device with multiple embedded sensors. Notable sensors that are embedded include the accelerometer, gyroscope, and magnetometer. These three types of sensors are commonly used for recording human motion and activity. Each of these sensors gather data in three coordinate directions (xyz) and therefore each Shimmer device has nine degrees of freedom. The gyroscope and the magnetometer are used in conjunction to measure accurately the orientation of the sensor. The accelerometer measures the proper acceleration experienced by a body, and is often used to record seismic activity among other applications like machine vibrations and airbag release mechanisms. When the Shimmers are attached to a body, in the simplest case of a single-degree-of-freedom structure, they measure the response of the body to the input excitation that is applied to the structure. We physically orient all Shimmers to record accelerometer data on the x-coordinate axis for ease and consistency of data collection.

Our method of time synchronization is similar to a master-slave network. All four Shimmers are wirelessly connected to the PC and the PC records the data that the sensors read (we are concerned with the accelerometer values). The Shimmer applies a timestamp to each instance that a sample is taken. We decide the rate at which Shimmers record samples. Unix time can be enabled for the Shimmers, which is a key part of our method of time synchronization. Unix time is defined as the number of seconds that have elapsed since Thursday, 1 January 1970, and is therefore a rather large number. The computer measures Unix time to the millisecond, which is necessary for our application. It provides us a global time and common reference time, which is recorded as a Unix timestamp for several Shimmer timestamp instances during any given test. In order to synchronize the data, we can then compare the Unix timestamps and Shimmer's local timestamps of all Shimmers with just one Shimmer. Therefore, it may seem like that one Shimmer is the master node, however the Shimmers communicate with the PC and not with each other. Though the PC itself is not a sensor gathering acceleration data, it is useful to describe the system as a master-slave scheme. Our method of time synchronization is implemented after the data has been collected (during post-processing), and therefore differs from common methods where time is synchronized and clocks are corrected within the sensors themselves. Our method reduces stress on the sensors, so it may be a more viable option for smart wearable devices which are limited by processing power, memory, and energy capacity[4].

B. MATLAB and Synchronization Procedure

We use MATLAB for this research because we have access to the software, we have experience using it, and it can be used for commanding Shimmers. MATLAB is also a powerful tool for gathering and assessing data. We start off by downloading the most recent version of Shimmer's MATLAB instrument driver, the Shimmer MATLAB Instrument Driver v2.6. In addition to the driver, it also provides the user with a few MATLAB example functions for controlling the Shimmers. One such example function plots and writes data from a single Shimmer for a specified amount of time and saves the data into a specified data file. This data includes the low noise accelerometer, the gyroscope, and the magnetometer readings in the xyz directions, as well as the Shimmer's local timestamp, the Unix timestamp seen by the PC, and the battery voltage sensor readings. We modify the code for this function so that it connects to all four of our Shimmers rather than just one. We also add the synchronization procedure that we developed to the end of the MATLAB function. First, the function connects each Shimmer to a single PC via Bluetooth. It then assigns calibration values to each Shimmer. Next, this same function assigns sampling rates to each Shimmer. It then begins to collect data from all four Shimmers for a specified duration, and records that data collected from each Shimmer into a specified data file on the PC. Once it has reached the specified duration, it stops collecting data. It then disconnects the PC from all four Shimmers. The function then begins the synchronization procedure that we developed. It first reads all four data files from the current run. It uses that data to create two arrays for each

Shimmer: one with all of the Unix timestamps that the PC recorded and one with all that Shimmer's local timestamps that are paired with a Unix timestamp. It then uses those two arrays to create an array of offsets between the first Shimmer and the other three Shimmers. It does this by calculating the time offset between the first Shimmer and the other three Shimmers for each timestamp. This is done by subtracting the difference in the Shimmer's local timestamps from the difference in the Unix timestamps. For example, the equation to calculate how much behind Shimmer 2 is from Shimmer 1 is as follows, where u represents Unix timestamp, and t represents the local timestamp:

$$\text{Offset} = (u_2 - u_1) - (t_2 - t_1)$$

It then takes the average of each time offset array to find the estimated time offset for each individual sensor. It then adds the time offset to the Shimmer timestamps for every Shimmer (except the first Shimmer). Finally, it writes the new timestamps into a new data file for all of the Shimmers (except the first Shimmer). The function then displays a graph of the unsynchronized data from the low noise accelerometer in the x direction and a graph of the synchronized data from the low noise accelerometer in the x direction, and then ends. The graphs allow us to visualize how well the data was synchronized and the data files allow us to numerically determine how well the data was synchronized.

III. Testing

A. Procedure

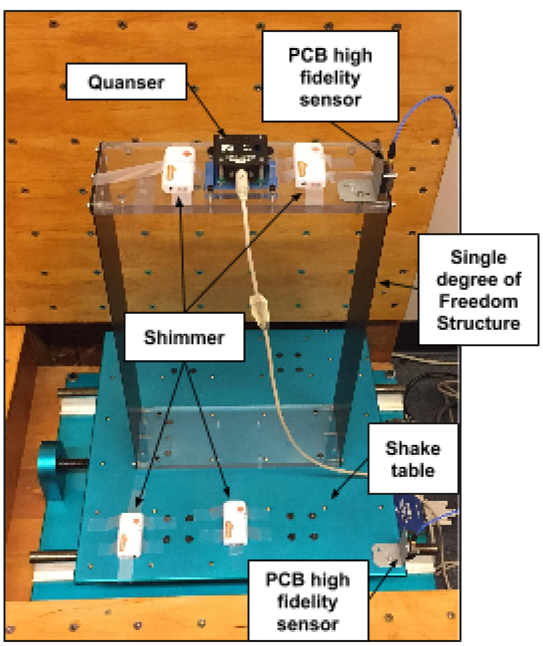

Figure 1: Shows an image of the Single Degree of Freedom Structure on a shake table with four Shimmer Devices, a Quanser Accelerometer Module, and a PCB high fidelity sensor.

We test our time synchronization procedure by applying it to the data that we collect from our Shimmer sensors and validate that they are all synchronized together. To do this, we attach them to a single degree of freedom (SDOF) structure that is fixed on a shake table, which can be seen in Figure 1. We run several tests using the structure and the table. By simulating certain earthquakes and vibrations onto the SDOF structure, we are able to analyze the data each sensor collected so that we can compare them during our post-analysis. The test simulations used are: two earthquakes, three sine waves with different frequencies, a sine sweep, and three free-vibrations with a displacement of four inches. We also embedded two high fidelity sensors onto the structure, which can be seen in Figure 1. Collecting data from both the Shimmer sensors and the high fidelity sensors allows us to compare the results and determine the accuracy and reliability of the Shimmer sensors.

B. Vibration Tests

1. Free Vibration

One of the tests we conduct is a free vibration test. We remove our SDOF structure from the shake table and put it on a static platform. Once we do this, we then displace the top of the SDOF structure four inches from its resting position. We then release it, allowing it to vibrate freely while the wireless sensors collect data. The collection of data helps us determine the effectiveness of the time synchronization procedure and determine the accuracy of that synchronized data by comparing it to the data collected by the high-fidelity accelerometers.

2. Sine Waves

For this part of the testing we attach the SDOF structure atop the shake table and run a sine wave in the shake table with different frequencies. We run one sine wave with a frequency of 1 Hz, one with 3.9 Hz, and one with 10 Hz. In total, we run three tests, each with different frequencies to determine whether our time synchronization procedure works to synchronize data from these different excitations. Again, we are able to see the data each sensor collects and compare the data from the wireless sensors both synchronized and unsynchronized.

3. Sine Sweep

Another type of test that we conduct is the sine sweep, which involves a sine wave with an increasing frequency at which the sinusoidal vibration travels. During the sine sweep tests, the frequency is allowed to increase from 0 Hz to 10 Hz. The reason why these tests are conducted is because we want to find out if the collected data can synchronize and also because we want to test the reliability of the sensors.

4. Earthquakes

Finally, we run a simulation of two different earthquakes on the shake table: one that occurred in Kobe, Japan in 1995, which had a magnitude of 6.9 on the Richter scale, and one that occurred in Northridge, CA in 1994 which had a magnitude of 4.0 on the Richter scale. We are able to use modeled representations of these earthquakes that are scaled down for our equipment to safely simulate, since the actual representation of these recorded earthquakes exceeds the limits of our shake table. Once we collect the data from these simulated earthquakes, we are able to compare and see the difference between the unsynchronized data and the synced data collected by the Shimmers.

IV. Results and Analysis

After each test, data collected by each Shimmer sensor is saved into a data file. The accelerometer data from those data files is then displayed in two graphs: one displaying the data *before* synchronization and one displaying the data *after* synchronization. Figures 2-11 display those graphs. The x-axes represent time in seconds and the y-axes represent acceleration in meters per second squared. The blue and purple lines refer to the accelerometer data collected from the two Shimmers fixed on top of the SDOF structure (upper Shimmers) and the orange and green lines refer to the accelerometer data collected from the two Shimmers fixed on the shake table (lower Shimmers). Refer to Figure 1 for a better visualization of the setup. We analyze the effectiveness of our synchronization procedure both visually, in Figures 2-11, and quantitatively, in Figure 12. Figure 13 is a comparison made between the accelerometer data collected from a Shimmer device and from a high-fidelity sensor.

Before conducting tests using the shake table, we first conduct a free vibration test with the SDOF structure fixed on a static platform. We begin recording data on all of the attached sensors. The top of the structure is displaced by four inches and then released (after about 5 seconds) so that it may vibrate freely. This test is done three times. Figures 2-3 show the results from one of those tests. Because it is a free vibration test, we expect the lower Shimmers (green and orange) to show no response and the upper Shimmers (blue and purple) to show a sine wave that decreases in amplitude over time.

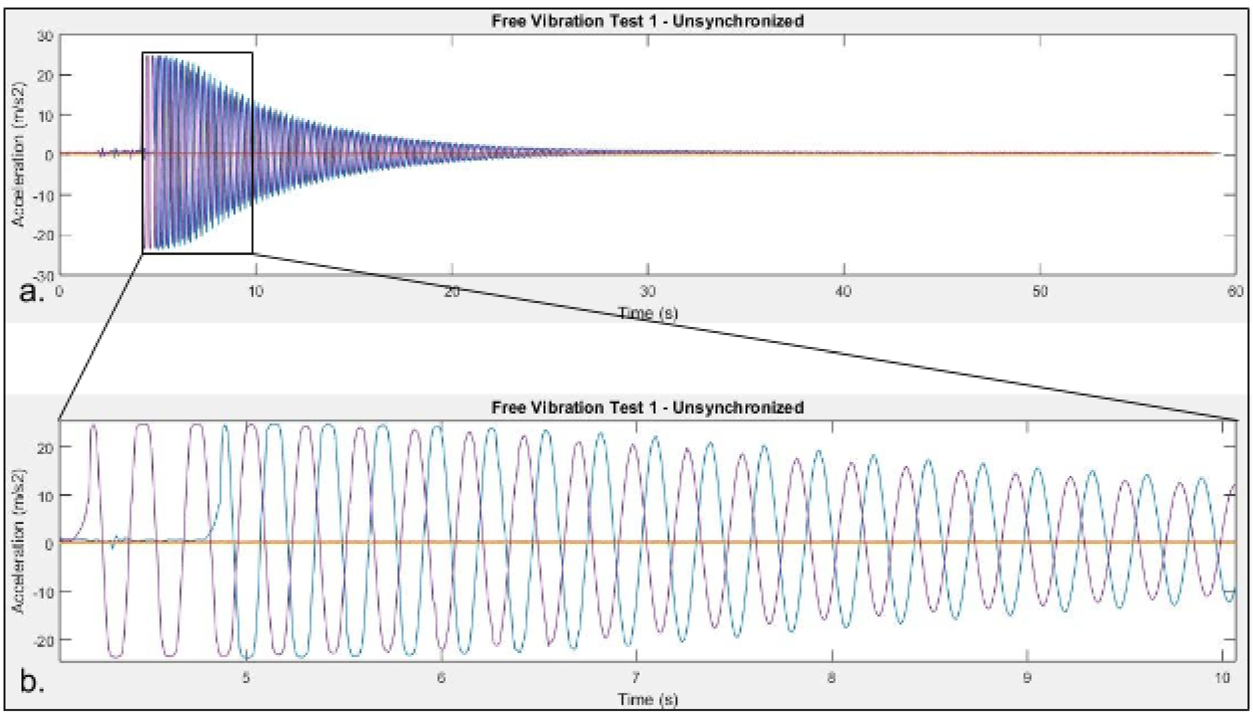

Figure 2: Shows (a.) a graph of the the unsynchronized data from one of the three free vibration tests that were conducted and (b.) a zoomed in section of that graph.

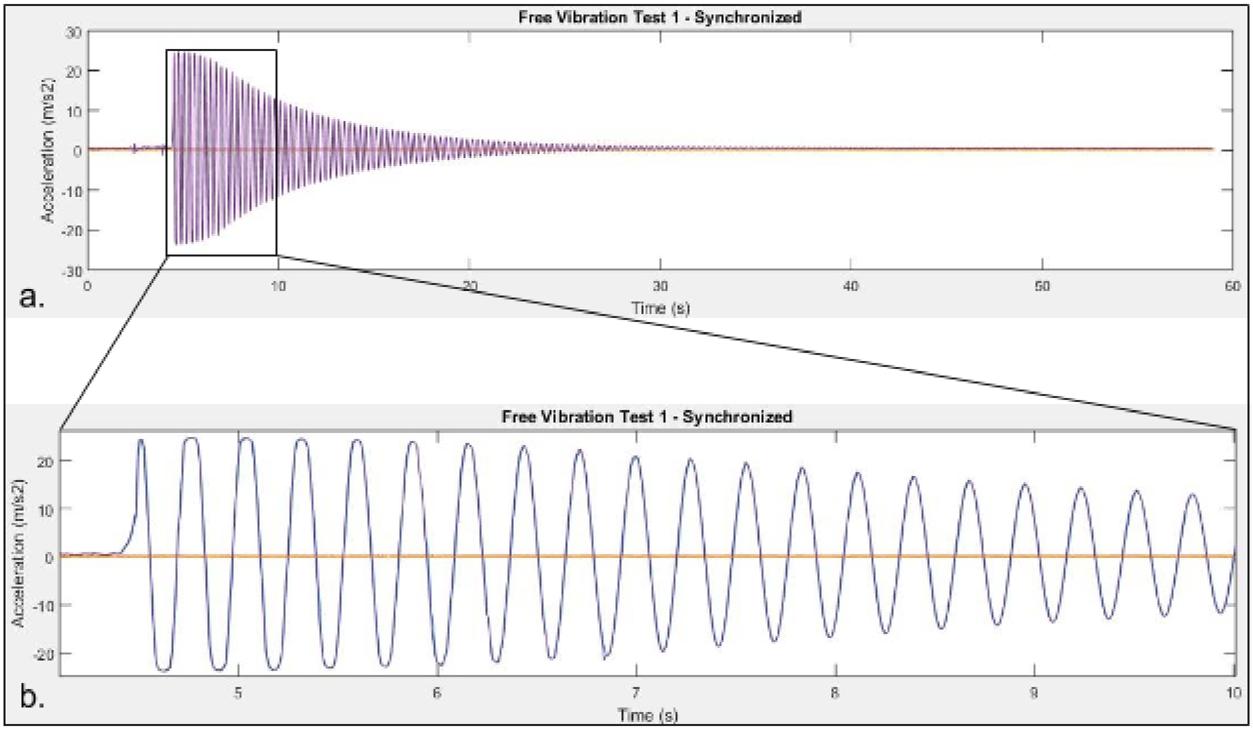

Figure 3: Shows (a.) a graph of the the synchronized data from one of the three free vibration tests that were conducted and (b.) a zoomed in section of that graph.

In both figures, the results show what we predicted they would do. However, Figure 2 shows that the two waves from the upper Shimmers are not aligned, proving that there is an offset in the timestamps. This is shown more clearly in Figure 2b. At first glance this misalignment looks small, that the two waves are only half a wavelength apart, which corresponds to 125 ms. However, they are actually 2.6 wavelengths apart, that is 670.5 ms apart, as is determined by numerical analysis in Figure 12. The data in Figure 3b is the data that has been processed by our time synchronization procedure in MATLAB. Figure 3b shows little to no offset in the two waves. The numerical comparison in Figure 12 proves the offset once synchronized to be only 2 ms compared to 670.5 ms pre-synchronization. This data analysis shows that the time synchronization procedure that we developed is effective in aligning the collected data. The same is demonstrated by the other tests .

Figures 4-5 show the results from one of the three sine wave tests that were conducted. We expect to see a sign wave from the lower Shimmers, and a response wave of the structure from the upper Shimmers. Once the shake table stops moving (after about 27 seconds), we expect to see a zero output from the lower Shimmers and a sine wave decreasing in amplitude over time from the upper Shimmers.

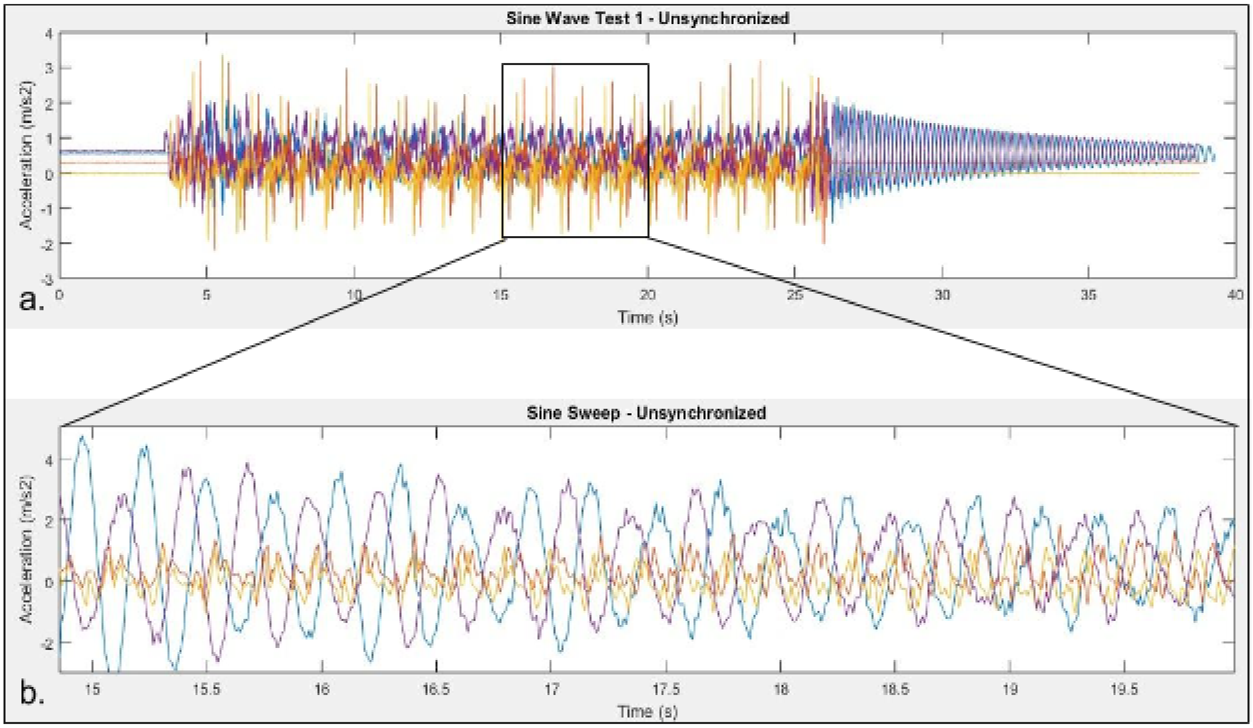

Figure 4: Shows (a.) a graph of the the unsynchronized data from one of the sine wave tests that were conducted and (b.) a zoomed in section of that graph.

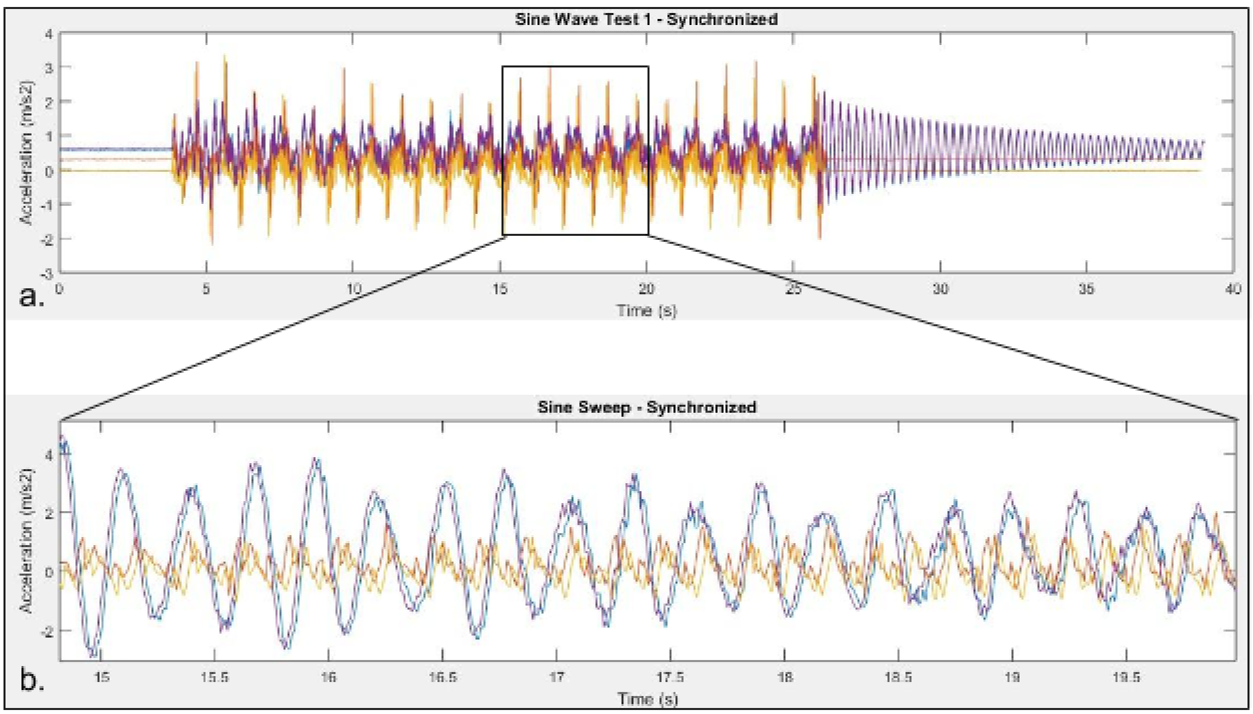

Figure 5: Shows (a.) a graph of the the synchronized data from one of the sine wave tests that were conducted and (b.) a zoomed in section of that graph.

In both figures, the same prediction we have occurs. However, Figure 4 shows that the four waves are not aligned, proving that there is an offset in the timestamps. Numerically analyzing the data on the graph we see there is an offset of 690.5 ms. This is shown more clearly in Figure 4b. Figure 5b shows little to no offset in the four waves. The data in Figure 5 is the data that has been processed by our time synchronization procedure. The data from this test trial shows that the time synchronization procedure that we developed is effective in aligning the collected data.

Figures 6-7 show the results of the sine sweep test that we conducted. In the data, we expect to see a sine wave increasing in frequency over time from the Shimmers embedded on the SDOF structure, and a response wave of the structure from the Shimmers on the shake table. As we predicted we can see in figure 6a that the response wave increased over time. In figure 6b the offset of the timestamps between the shimmers is noticeable. Numerically the offset between them is 660.0 ms. Once we put the unsynchronized data through the time synchronization procedure, we then see in figure 7 that the offsets are corrected and the synchronized data is easily visible. Although there is still a small offset of 16 ms, this data is still more useful to us than the unsynchronized data.

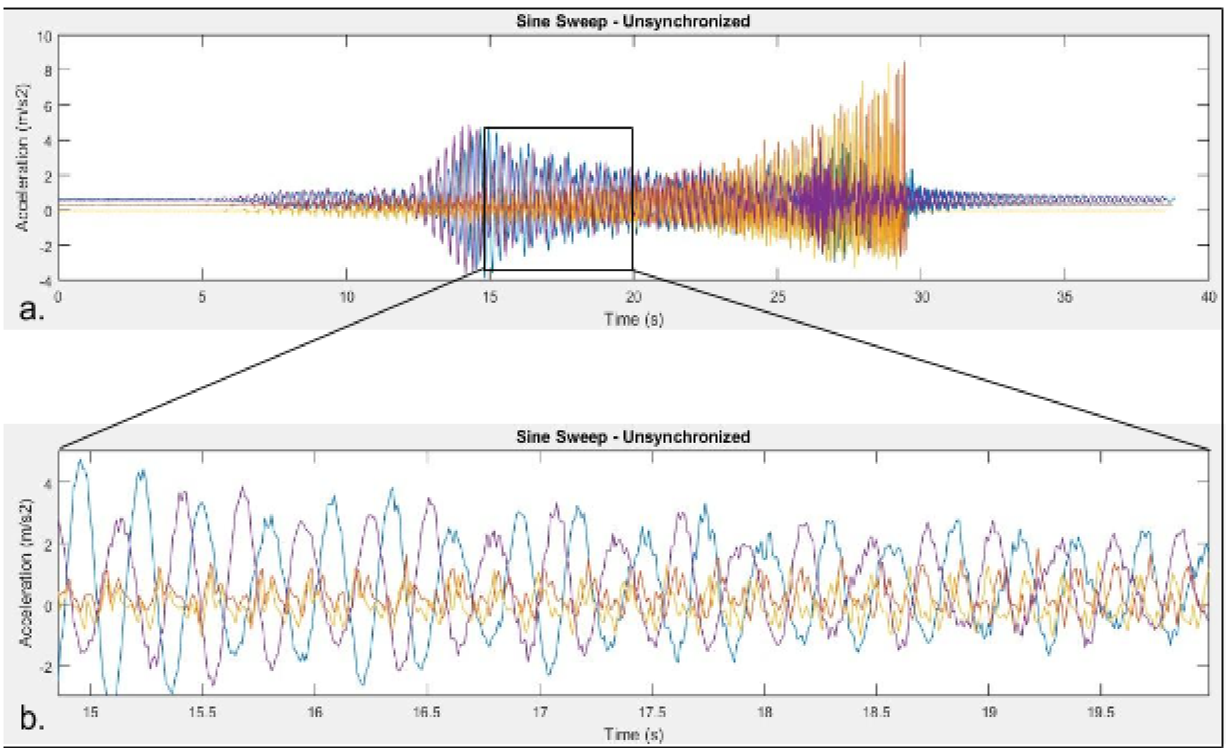

Figure 6: Shows (a.) a graph of the the unsynchronized data from the sine sweep test that was conducted and (b.) a zoomed in section of that graph.

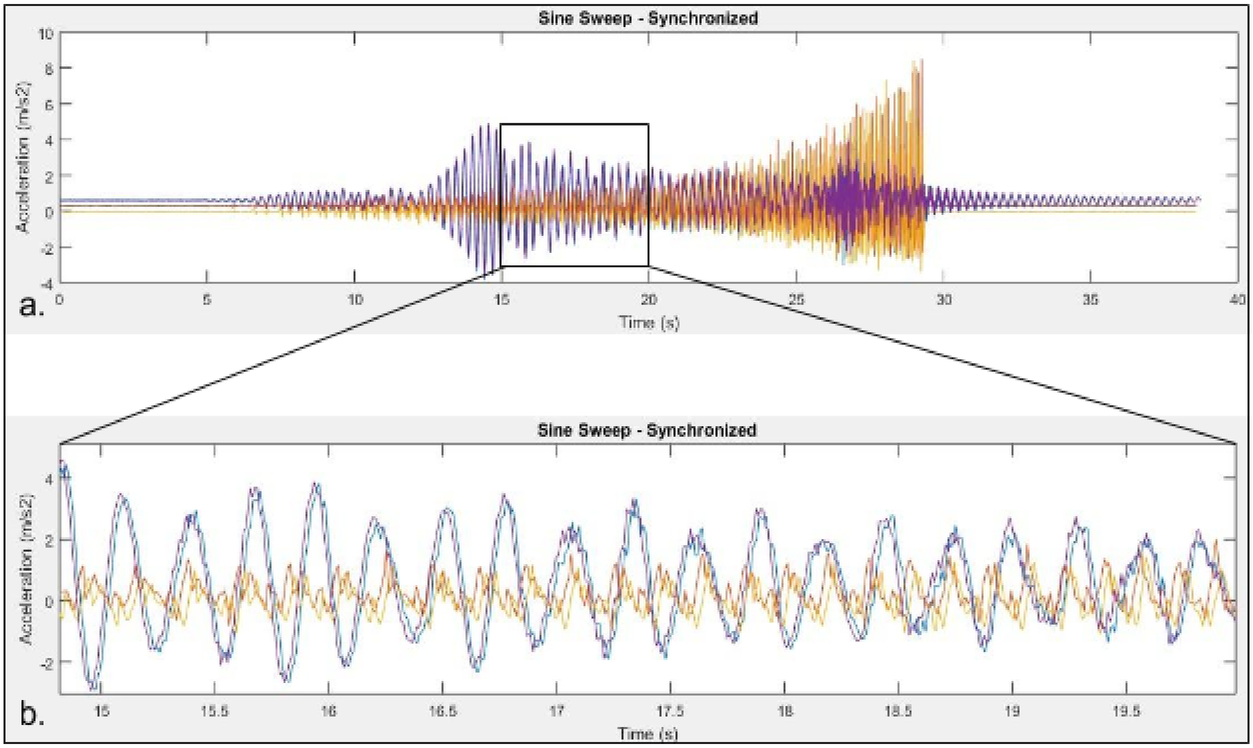

Figure 7: Shows (a.) a graph of the the synchronized data from the sine sweep test that was conducted and (b.) a zoomed in section of that graph.

In both figures, we get about what we expected. However, Figure 6 shows that the four waves are not aligned, proving that there is an offset in the timestamps. This can be seen more clearly in Figure 6b. Figure 7b shows little to no offset in the four waves. The data in Figure 7 has been processed by our time synchronization procedure. The data from this test trial shows that the time synchronization procedure that we developed is effective in aligning the collected data.

Figures 8-9 show the results of a test we ran which models a magnitude 6.9 earthquake that occurred in Kobe, Japan in 1995. We expect to see an unpredictable wave excitation with a sudden increase and then a more gradual decrease in amplitude. We still expect the upper Shimmers to be in synch with each other and the lower Shimmers to be in synch with each other.

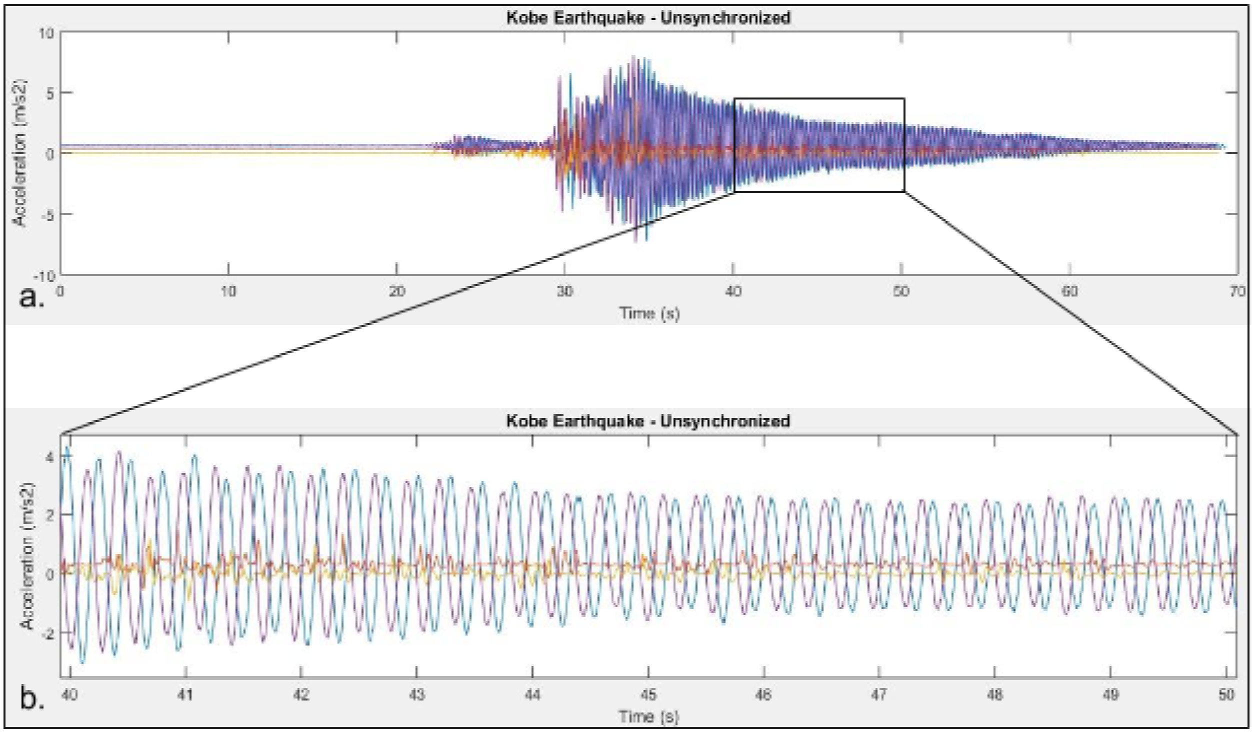

Figure 8: Shows (a.) a graph of the the unsynchronized data from the Kobe earthquake simulation that was conducted and (b.) a zoomed in section of that graph.

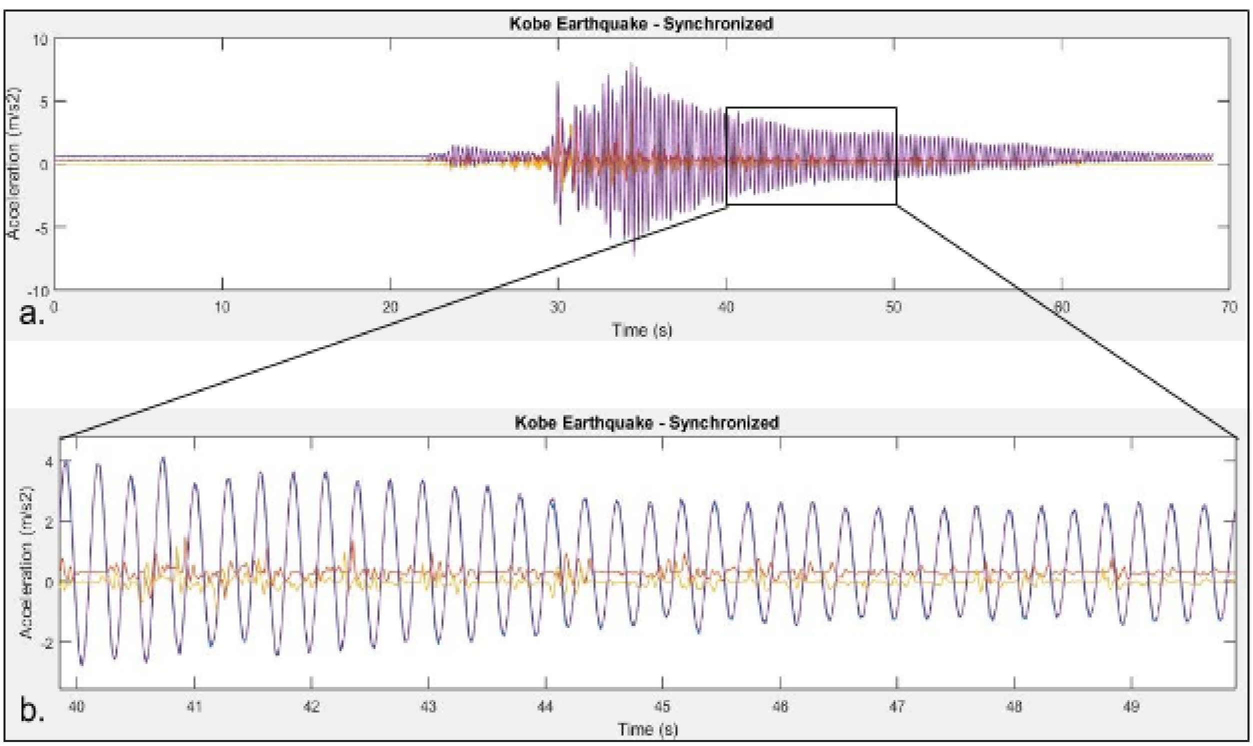

Figure 9: Shows (a.) a graph of the the synchronized data from the Kobe earthquake simulation that was conducted and (b.) a zoomed in section of that graph.

The figures above show that our unsynchronized Shimmers collect data at different time intervals, making it difficult to read the data they collect. Figure 8b shows that there is an offset between two sensors, the value of which is 655 ms. This makes it difficult to use the data from earthquake patterns that may be recorded using smart wearable devices. Figure 9 and 12 show that when all the Shimmers are synchronized together by our method the data collected is only offset by 5.7 ms.

Figures 10-11 below show the results of a simulation that models a 4.0 earthquake that occurred in Northridge, CA. As we've seen in the previous figures for the Japan earthquake, we can also see the unsynchronized data is unreadable so we want to run the procedure in order to synchronize all the data. Once we run the procedure we then see in Figure 11 that the data sensors collected are synchronized and that it validates that all the sensors are collecting the same data simultaneously.

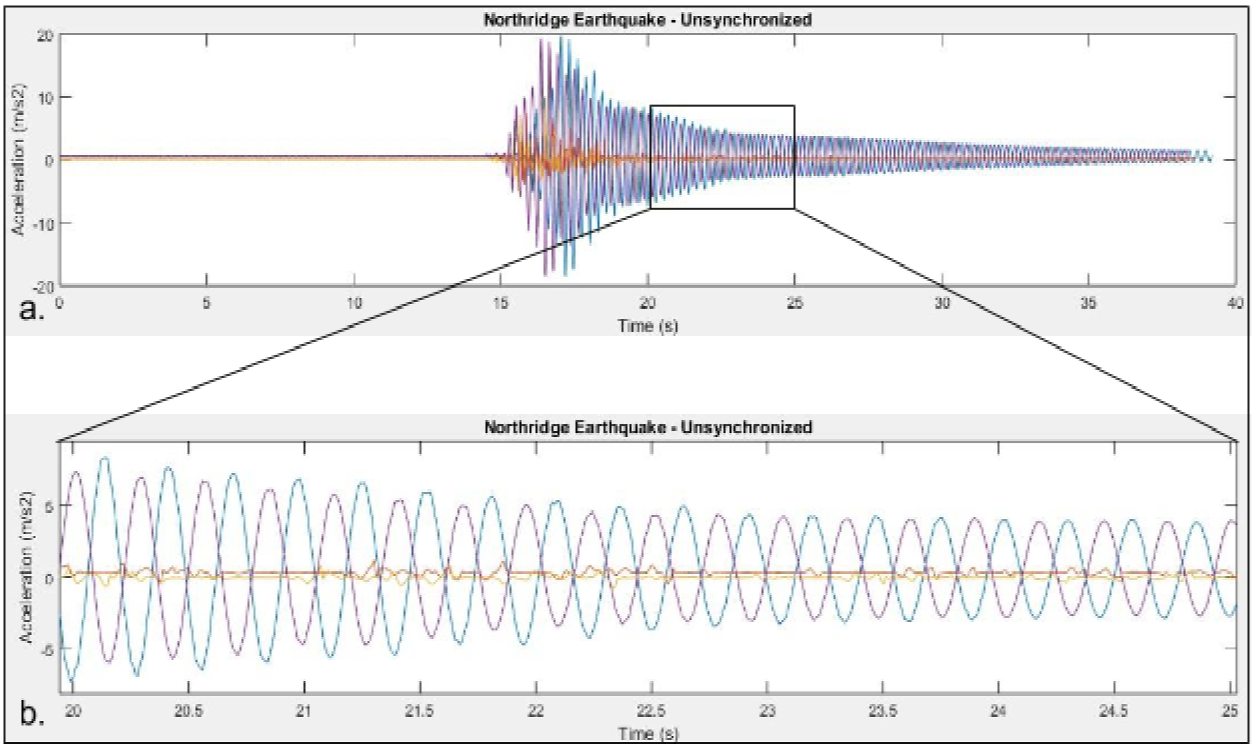

Figure 10: Shows (a.) a graph of the the unsynchronized data from the Northridge earthquake simulation that was conducted and (b.) a zoomed in section of that graph.

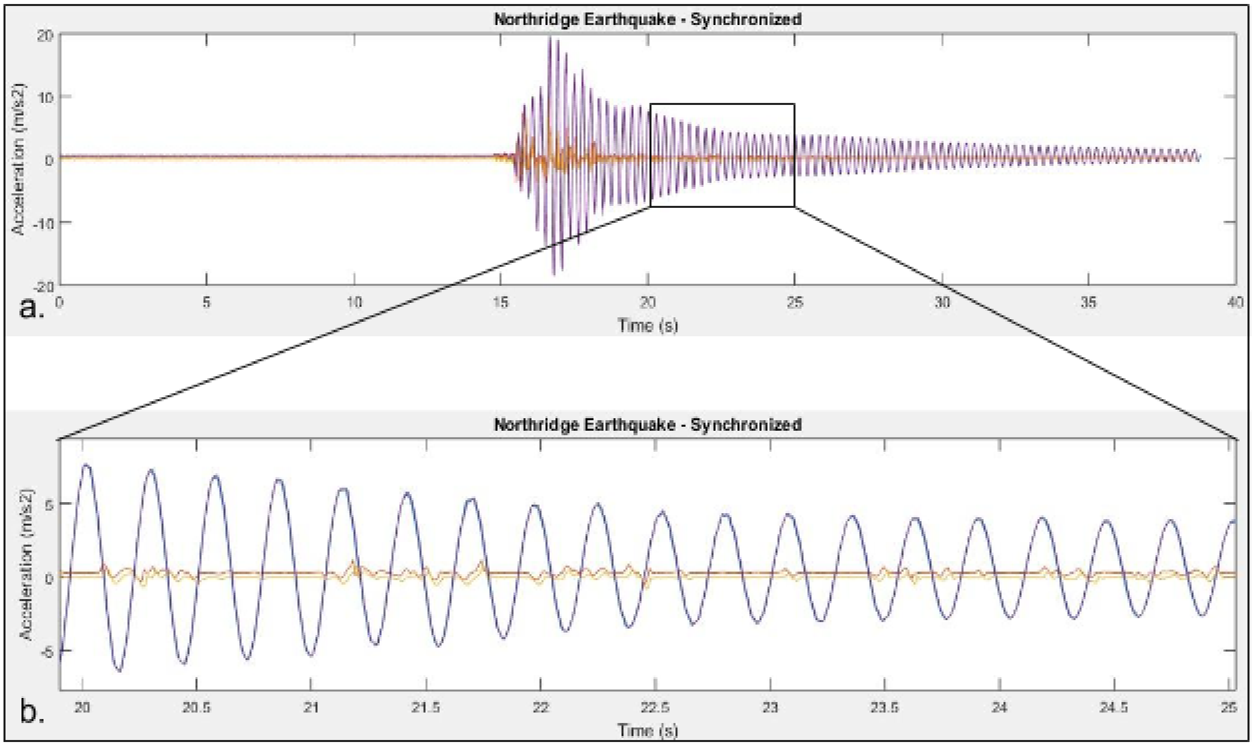

Figure 11: Shows (a.) a graph of the the synchronized data from the Northridge earthquake simulation that was conducted and (b.) a zoomed in section of that graph.

Figure 12 shows the numerical offsets for the unsynchronized and synchronized data for each test. We had a total of four Shimmer sensors available, but the table includes comparison of the two on top of structure. The table of numerical offsets includes values (in milliseconds) that were calculated from the peaks of the graphs. Peaks were analyzed because they are easy to compare and special attention was given to the beginning of the data sets to ensure that the correct peaks were chosen.

Test	Unsynced Offsets (ms)	Synced Offsets (ms)	Percent Decrease in Offset (%)
Free Vibration 1	670.5	2.0	99.7
Free Vibration 2	730.0	9.0	98.8
Free Vibration 3	691.0	22.5	96.7
Sine Wave 1	690.5	5.5	99.2
Sine Wave 2	746.0	15.5	97.9
Sine Wave 3	625.0	19.0	97.0
Sine Sweep	660.0	16.0	97.6
Northridge Earthquake	690.0	1.9	99.7
Kobe Earthquake	655.0	5.7	99.1
Maximum	746.0	19.0	
Minimum	625.0	1.9	
Average	677.8	10.6	98.4

Figure 12: Numerical comparison of the synced and unsynced data. This table shows how far apart the top two shimmers are from each other.

The tests that we have conducted demonstrate the effectiveness of the time synchronization procedure that we have developed and the reliability of the sensors used in wireless devices. We can witness the effectiveness of this time synchronization procedure visually by analyzing the graphs in Figures 2-11. When comparing the graphs, it can be seen that the waves are aligned in the synchronized versions compared with the unsynchronized versions because the offsets have been dramatically reduced. This occurs in all of the test trials. We can also see the effectiveness of this procedure numerically. Before synchronization, the time offsets between each Shimmer are on average 677.8 ms with a minimum offset of 625 ms. After synchronization, the average time offset is 10.6 ms with a minimum offset of 1.9 ms. The percent decrease in offset due to our synchronization procedure is on average 98.4%.

Figure 13 below compares the acceleration data collected from the Shimmer to the data from the high-fidelity sensor in a free vibration test in order to determine it's accuracy and reliability.

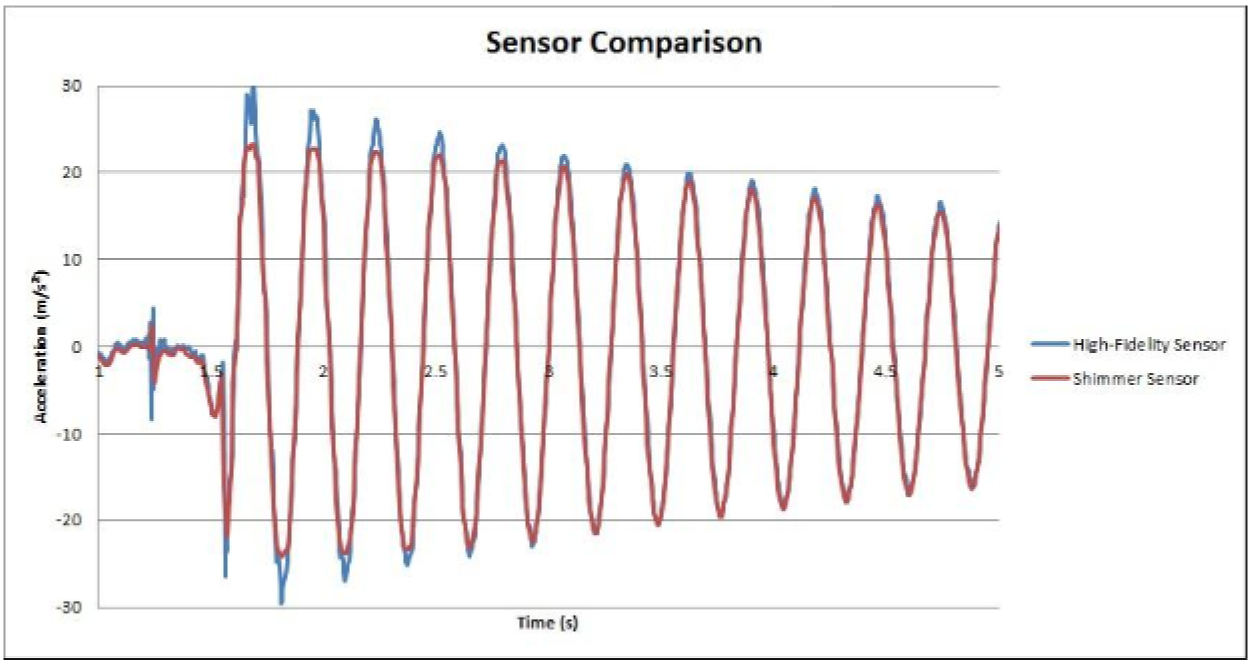

Figure 13: Compares the data collected from the Shimmer to the High-Fidelity sensor to determine the accuracy of sensors in wireless devices.

Figure 13 shows that the data gathered by the Shimmer sensors and high-fidelity sensors match, thus verifying the reliability of the sensors in the Shimmer devices. This, in turn, verifies the reliability of the sensors in smart wearable devices. Although the amplitudes for the beginning of the free vibration test are slightly off with a percent difference of 20%, the amplitudes of the Shimmer sensors become more accurate with time and have a percent difference of 10%.

V. Conclusion

We have successfully developed a procedure for synchronizing wireless devices using MATLAB. We are able to run four Shimmer sensors simultaneously and collect data from each that is then synchronized. The results verify that the sensors are able to collect and read the same vibration frequencies simultaneously as well as sync the data well. We validate this by analyzing synchronized and unsynchronized visually as well as numerically. It is also shown that sensors are reliable enough to collect almost identical data when they are synchronized.

As we progress in our research, we want to apply this synchronization method to smart wearable devices like smartwatches. It is important for us to use this method on all the sensors so that we can determine the limits of the sensors inside the devices when collecting high frequency data. Now that we are able to verify the accurate collection of these wireless sensors we can further test the sensors within the smart wearable devices. This will then aid our ability to use the devices to distinguish the difference between white noise, such as a person's movements, and the seismic data from the ground during an earthquake. With this research we can potentially apply our methodology for applications that will further our understanding of monitoring earthquakes and seismic activity. We hope to see these applications in seismic monitoring, structural analysis of buildings and infrastructure, and potentially even early seismic warnings.

Acknowledgments

We would like to acknowledge Dr. Zhaoshuo Jiang, Assistant Professor of Civil Engineering at San Francisco State University, for his guidance in our learning and research. We acknowledge student mentors Jackie Lok and Alec Maxwell for helping us research and conduct experiments. We also acknowledge Dr. Amelito Enriquez for his work to organize the research program and his guidance for our group. These have contributed to the success of this research and their time and expertise is greatly appreciated.

This project was partly supported by the US Department of Education through Minority Science and Engineering Improvement Program (MSEIP, Award No. P120A15014), and the Hispanic-Serving Institution Science, Technology, Engineering, and Mathematics (HSI STEM) Program, Award No. P031C110159.

Bibliography

1. Ali, Shaukat. "Sensors and Mobile Phones: Evolution and State-of-the-Art."*Pakistan Journal of Science* (2014): n. pag. *ResearchGate*. Web. 5 July 2016.

2. Feng, M., Fukuda, Y., Mizuta, M., & Ozer, E. (2015). Citizen Sensors for SHM: Use of Accelerometer Data from Smartphones. *Sensors, 15*(2), 2980-2998. doi:10.3390/s150202980

3. Roche, Michael. "Time Synchronization in Wireless Networks." *Time Synchronization in Wireless Networks*. Washington University in St. Louis, n.d. Web. 03 Aug. 2016. <http://www.cs.wustl.edu/~jain/cse574-06/ftp/time_sync/index.html>.

4. Sundararaman, Bharath, Ugo Buy, and Ajay D. Kshemkalyani. "Clock Synchronization for Wireless Sensor Networks: A Survey." *Ad Hoc Networks* 3.3 (2005): 281-323. Web.