UNIVERSITY OF CALIFORNIA,
IRVINE

Event Detection and Estimation Using Distributed Population Owned Sensors

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical Engineering and Computer Science

by

Ahmed Mokhtar Nagy Ibrahim

Dissertation Committee:
Professor Ahmed Eltawil, Chair
Professor Ender Ayanoglu
Professor A. Lee Swindlehurst

2019

DEDICATION

To the memory of my Father,

my beloved Mom,

my dear Brother,

my beloved Wife,

and my little Daughter

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor Ahmed Eltawil, for the continuous support and guidance that he provided during my research at UC Irvine. I would also acknowledge the patience and perseverance that he showed towards this research. He was always thoughtful and attentive, which helped me a lot to face the research challenges, and he was always encouraging to explore new ideas and methods.

I would also like to thank Professor Ender Ayanoglu for participating in my qualification exam and defense committee, I also would like to mention that what I have learned from his graduate digital communications course has built the mathematical foundation that I have used to develop most of the theoretical formulation in my research.

Besides, I would like to thank Professor A. Lee Swindlehurst for dedicating some of his time to participate in my qualification exam and defense committee.

With special thanks to Professor Sherif El-Tawil and Dr. Yunsu Na, for the last five years of fruitful collaboration, between UCI and the University of Michigan.

I would like to acknowledge the support of the National Science Foundation (NSF) for funding this work under Award Numbers CMMI:1362547, CMMI:1362458 and OAC:1638186.

I would also thank my colleagues and friends, especially Dr. Ahmed Khorshid who was a real brother and supporter. I also would like to thank Mohamed Nada, Dr. Mohamed Othman, Islam Rabie, Mohamed Fouda, Sergey Shaboyan, Marwen Zorgui, Tarek Mealy, Ahmed Farghali and other friends in southern California who are considered our big family. Besides, I would like to acknowledge the help of my colleague Ahmed Alzughaibi, specifically in the implementation of a practical application of this research.

In addition, I would like to thank my parents for their extreme support during my research. My father, may God bless his soul, had encouraged me so much to pursue PhD studies, and my mother who is continuously supporting me in all aspects of my life. I would like to thank my brother as well who was a real motivator and helper.

Finally, my special thanks go to my wife, Amira Youness, for her continuous and constant support. Her perseverance has much motivated me to continue my studies and to pursue this degree. I would also like to thank my little angel, Mariam who was a true bless during this journey.

CURRICULUM VITAE

Ahmed Mokhtar Nagy Ibrahim

EDUCATION

Ph.D. in Electrical Engineering and Computer Science **2019**
University of California, Irvine *Irvine, California*

M.Sc. in Microelectronics System Design **2014**
Nile University *Giza, Egypt*

B.Sc. in Electronics and Electrical Communication **2008**
Ain Shams University *Cairo, Egypt*

RESEARCH EXPERIENCE

Graduate Research Assistant **2015–2019**
University of California, Irvine *Irvine, California*

INDUSTRY EXPERIENCE

Storage System and Controller Architecture PhD Intern **Summer 2018**
Western Digital *Irvine, California*

Computer Hardware Developer Intern **Summer 2016**
Bloomberg *Manhattan, New York*

Staff Digital Design Engineer **2008 – 2014**
Si-Ware Systems *Cairo, Egypt*

SELECTED PUBLICATIONS

- **A. Ibrahim**, A. Eltawil, Y. Na and S. El-Tawil, "Accuracy Limits of Embedded Smart Device Accelerometer Sensors," submitted to IEEE Transactions on Instrumentation and Measurement, 2019.
- **A. Ibrahim**, A. Eltawil, Y. Na and S. El-Tawil, "A Machine Learning Approach for Structural Health Monitoring Using Noisy Datasets," submitted to IEEE Transactions on Automation Science and Engineering, 2019.
- Y. Na, S. El-Tawil, **A. Ibrahim** and A. Eltawil. Automated Assessment of Building Damage from Seismic Events using Smartphones," submitted to Journal of Structural Engineering, 2019.

- **A. Ibrahim**, A. Eltawil, Y. Na and S. El-Tawil, "Effect of Sensor Error on the Assessment of Seismic Building Damage," in IEEE Transactions on Instrumentation and Measurement, 2019. doi: 10.1109/TIM.2019.2896371

- A. Alzughaibi, **A. Ibrahim**, A. Eltawil, Y. Na and S. El-Tawil. "Post-Disaster Structural Health Monitoring System Using Personal Mobile-Phones" Wireless Sensors and Sensor Networks (WiSNet), 2019 IEEE Topical Conference on. IEEE, 2019.

- Y. Na, S. El-Tawil, **A. Ibrahim**, and A. Eltawil. "Dynamic behavior of a smart device on a surface subjected to earthquake motion." Earthquake Engineering Structural Dynamics 47, no. 9 (2018): 1905-1920.

- Y. Na, S. El-Tawil, **A. Ibrahim** and A. Eltawil. The Feasibility of Using Smart Devices for Quantifying Seismic Damage to Buildings. Structures Congress 2017.

PATENTS

- A. Elmallah, A. Elshennawy, A. Shaban, B. George, M. Elmala, A. Ismail, M. Sakr, **A. Mokhtar**, A. Elsayed, "Interface for MEMS inertial sensors", U.S. Patent 9,013,233, issued April 21, 2015.

- A. Elsayed, A. Elmallah, A. Elshennawy, A. Shaban, B. George, M. Elmala, A. Ismail, M. Sakr, **A. Mokhtar**, "Interface for MEMS inertial sensors", U.S. Patent 8,508,290, issued August 13, 2013.

- **A. Mokhtar**, A. Elmallah, A. Elshennawy, A. Shaban, B. George, M. Elmala, A. Ismail, M. Sakr, A. Elsayed, "Interface for MEMS inertial sensors", U.S. Patent 8,476,970, issued July 2, 2013.

PEER REVIEW ACTIVITIES

- IEEE Transactions on Instrumentation and Measurement

- Journal of King Saud University - Computer and Information Sciences

ABSTRACT OF THE DISSERTATION

Event Detection and Estimation Using Distributed Population Owned Sensors

By

Ahmed Mokhtar Nagy Ibrahim

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California, Irvine, 2019

Professor Ahmed Eltawil, Chair

Smart phones are an indispensable tool in modern day-to-day life. Their widespread use has spawned numerous applications targeting diverse domains such as bio-medical, environment sensing and infrastructure monitoring. In such applications, the accuracy of the sensors at the core of the system is still questionable, since these devices are not originally designed for high accuracy sensing purposes. In this thesis, we investigate the accuracy limits of one of the commonly used sensors, namely, a smart phone accelerometer. As a use case, we focus on utilizing smart phone accelerometers in structural health monitoring (SHM). Using the already deployed network of distributed citizen-owned sensors is considered a cheap alternative to standalone sensors. These devices can capture floors vibration during disasters, and consequently compute the instantaneous displacement of each floor. Hence, damage indicators defined by government standards such as peak relative displacement can be estimated. In this work, we study the displacement estimation accuracy and propose a zero velocity update (ZUPT) method for noise cancellation. Theoretical derivation and experimental validation are presented and we discuss the impact of sensor error on the achieved building classification accuracy. Moreover, in spite of the presence of sensor error, SHM systems can be resilient by adopting machine learning. Several algorithms such as support vector machine (SVM), K-nearest neighbor (KNN) and convolutional neural network (CNN) are adopted and compared. Techniques for addressing noise levels are proposed and

the results are compared to regular noise cancellation techniques such as filtering.

Finally, since most previous work focused on modelling the sensor chip error itself, we study other sources of error such as sampling time uncertainty which is introduced by the device operating system (OS). That type of error can be considered a major contributor to the overall error, specially for sufficiently large signals. Hence, we propose a novel smart device accelerometer error model that includes the traditional additive noise as well as sampling time uncertainty errors. The model is validated experimentally using shake table experiments, and maximum likely-hood estimation (MLE) is used to estimate the model parameters. Moreover, we derive the Cramer-Rao lower bound (CRLB) of acceleration estimation based on the proposed model.

Chapter 1

Introduction

1.1 Motivation

With the advent of the smart device era, numerous applications (*apps*) have emerged that monitor a wide range of phenomena, ranging from human activities and vitals to environment sensing and infrastructure monitoring. The driving factor behind these innovative apps is the availability of sensors such as inertial (accelerometers and gyroscopes), pressure, temperature, humidity, light and proximity sensors.

Several apps utilize smart phone tri-axial accelerometers in bio-medical applications to detect or monitor diseases. In [4], a smart phone-based app that is used to assess upper limb tremor in Parkinson's disease (PD) patients has been developed. That app assists physicians remotely to evaluate the patient's condition. Depending on a smart phone in such situations outperforms any other biomedical device in terms of cost, since no special hardware is needed. In [5], atrial fibrillation (AFib) detection has been proposed. A smart phone is placed on the chest of the patient and performs a noninvasive recording. No external sensors are needed yielding a very low cost approach. Using this method, it is possible to detect paroxysmal

(or silent) AFib. In [6], regular activities are detected and recorded as a main indicator of health. Data is captured by smartphones inertial sensors, then an online classifier differentiates between five common activities: cycling, running, sitting, standing, and walking. Monitoring these activities can effectively prevent and manage chronic diseases. Moreover, in [7] an approach for the monitoring of elders and people with neurological pathologies (e.g., Alzheimer) has been proposed. Multi-sensor approaches are used to recognize critical events, such as falls or prolonged inactivity, that can be used to alert caregivers. Algorithms have been developed and applied on data provided by inertial sensors embedded in the user device. Other than alerts, the information provided by the proposed system can be used to track the evolution of diseases, and to track rehabilitation progress. In [8], smart watch accelerometers have been used to capture workout data which can assist users and physicians in monitoring health status and rehabilitation tasks. The proposed framework MiLift, automatically performs exercise identification, in addition to start/stop detection to remove user burdens of manual data entry.

On the other hand, smart phone apps also target environment and public safety and structural health monitoring (SHM) monitoring. For instance, in [9], a method to compute the gravity orientation using a smart phone accelerometer was presented. In [10], a crowd sensing application to estimate road conditions (CRATER) was presented. The app uses accelerometer data to detect speed bumps and potholes, and reports the locations to a cloud-based application engine.

In fact, smart devices are not originally designed for high accuracy sensing purposes, as a result, the system performance is questionable. In this thesis, we investigate the accuracy limits of a smart device accelerometer, and its application in SHM which is explained in section 1.2.

1.2 SHM

Infrastructure monitoring systems that use a smart device in the sensing mechanism have been studied recently. Smart phones offer a cheap attractive alternative to standalone sensors, since the phones network is already deployed. Moreover, the system is upgraded automatically since users are regularly purchasing new phones which are equipped with higher accuracy sensors. As mentioned, we focus on SHM application as a use case of smart phone accelerometer. Related work in this field is summarized below.

1.2.1 Related Work

In [11], community seismic network (CSN), an earthquake detection system was developed using phone accelerometers.CSN is a dense open seismic network based on cheap standalone micro electro mechanical system (MEMS) sensors owned by citizens as well as smart devices. The measurements are sent to a cloud server for processing. It was shown that CSN is capable of generating a seismic map, i.e. map that displays peak acceleration, and that is generated shortly after the event. The system is able to capture the s-wave in addition to the much harder to detect p-wave. CSN was built to replace the United States Geological Survey (USGS) earlier system "Did You Feel It" (DYFI)? which basically asks people if they have felt shaking and how strong the shaking was. People are required to grade the shaking in a scale from 1 to 10.

In general, USGS already has its own sensing network, however CSN was intended to add several measuring points and consequently to increase the seismic map resolution. General earthquake parameters such as the epicenter is easily detected using the first few readings. One of the main advantages CSN is the cheap deployment costs since smart phones are already deployed and the used standalone sensors cost is about $100, which is much cheaper

compared to the multi-thousand dollar seismic-grade accelerometers. As an enhancement to CSN, in [12] decentralized event detection was adopted followed by server level detection. Single device detection was achieved by using a machine learning (ML) algorithm. The researchers derived the relation between error rate of a single device and the overall system accuracy.

Another smart phone based application, named iShake was designed by a team at UC Berkeley[13]. iShake uses inertial sensors in smart phone to measure earthquake vibration. The main goal of the project is also to gather more quantitative data, than DYFI. iShake is not focusing on early warning, but it estimates some earthquake parameters, such as Arias intensity. Further improvement to iShake has been achieved in the successive project; MyShake [14]. This system successfully provided a 20 seconds warning for a magnitude 5 (M5) earthquake. It was also shown that mobile phone sensors can detect earthquakes of magnitude 5 or higher. MyShake adopted machine learning algorithms for detection rather than simple threshold mechanism used earlier in iShake. Machine learning role is to detect earthquake shaking when it happens and to distinguish earthquake vibration from normal activities.

In [15, 16], a mobile app was developed that utilizes phones internal accelerometers to sense earthquake vibration and assess building damage status. The main goal was not to estimate earthquake parameters, but to infer particular information about a given building such as its damage level. The developed app estimates the interstory drift ratio (IDR) which is a key damage indicator as illustrated in section 1.2.2. The system uses a central Amazon web services (AWS) server connected to several phones located in a given building. At the advent of a disaster, phones automatically record acceleration which is sent to the server. Then, the server in turn uses the crowd sourced acceleration data to calculate damage indicators. Finally, the building are tagged according to its damage severity and displayed on an online map.

1.2.2 Damage Indicator

Relative displacement of floors within a given building is used to calculate the IDRs for the building as explained in chapter 2. Government federal agencies and civil engineering societies such as Federal Emergency Management Agency (FEMA) and American Society of Civil Engineers (ASCE) release documents that relate IDR values to building damage level. For instance, in [2, 17, 18], two main critical thresholds of relative floor displacement are defined, such that the building can be classified into one of three categories: immediate occupancy (IO), life safety (LS) or collapse prevention (CP), which indicate that the building is either safe, needs further inspection or unsafe respectively. Hence, during an earthquake, the instantaneous relative floor displacement is a key indicator of the structure state [3]. In this work, we study the effect of sensor error on IDR calculation, and consequently its impact on classification accuracy is also presented.

1.2.3 Machine Learning in SHM

Machine learning algorithms have been used recently to identify structural damage. For instance, damage identification method intended for steel moment-frame buildings was presented in [19]. The method was based on artificial neural networks and modal variables. In [20], SHM was achieved using a neuro-wavelet technique. Damage was detected based on the continuous wavelet transform (CWT). According to [21], it is expensive to increase the number of the measurement points, hence, one way to address that issue is to predict the unmeasured mode shape data based on a limited number of measured data. Two-stage neural network was adopted, the first is used to predict the unmeasured mode shape and the second was used to localize and estimate the damage severity. Machine learning techniques specially deep learning has shown robustness and resilience in noisy environments such as speech recognition applications. In this work, we apply several machine learning classifiers

to detect damage level and investigate their resilience against noise.

1.2.4 Reliability of Measurements

One argument against adopting a smart phone based system in SHM is the chance of having unreliable measurements. Readings are generally accompanied by two main sources of error: 1) phone related error due to the internal sensor chip or the device operating system (OS) and 2) the fact that the devices are not fixed to the vibrating structure, for instance if the phone is left on a horizontal surface during shaking, then there is a chance of sliding. In this thesis we study the former source of error, and we propose several techniques for minimizing its impact. With respect to the later, in [22] the response of a non fixed smart device placed on a moving plane was studied. As expected, there is a chance of sliding that depends on the friction coefficient between the phone and the underlying surface. Detecting sliding interval was achieved by detecting sharp transitions of acceleration accompanying the change of state. Since the sliding interval can be detected, it can be labeled as corrupted data and ignored.

For both sources of error, since nowadays it is legitimate to assume several smart phones are located in each floor, any interval of missing or corrupted data in one phone can be recovered from another one located in the same floor since they are subjected to the same excitation signal. Moreover, by using averaging or more complex estimation technique accuracy can be enhanced significantly.

1.3 Large Signal Errors

In literature, inertial sensors main stochastic error contributor is assumed to be random additive noise [23, 24]. That assumption is acceptable for small signals or as long as two conditions are satisfied: 1) the clock driving the sensor is stable with low jitter and 2) output

samples are read properly, i.e. no samples are dropped or read twice. However, for a smart device accelerometer (and other sensors as well) those conditions are not necessarily satisfied due to the OS timing uncertainty. In [25], it was shown that Android has some limitations to be considered as a real time OS, and the execution is expected to have some uncertainty that strongly depends on the coding style.

Sampling time jitter was studied in [26]. Error was modelled as additive white noise and sampling time uncertainty. Unlike additive noise, the error due to sampling time jitter is signal dependent. It directly depends on the signal first derivative, which is proportional to the signal amplitude and sensitive to sharp transitions. In other words, a signal of small amplitude or slow variation is not much affected by sampling jitter and vice versa. Since the error is signal dependent and function of its first derivative, hence deriving the optimal estimator is almost impossible. Hence, the researchers in the same paper proposed maximum likely-hood estimation (MLE) to estimate the model parameters and it outperformed LS method. In [27], sampling time uncertainty in multi-channel analog to digital converters (ADCs) was studied. An iterative MLE method was proposed and its accuracy was very close to the best possible accuracy determined by the Cramer-Rao lower bound (CRLB) which was also derived. We adopt similar methodology to model the accelerometer error for large signal. An error model that takes into account sampling time uncertainty is proposed, MLE is used to estimate the model parameters and the CRLB is derived. The proposed model is validated experimentally using shake table experiments as explained in details in chapter 4.

1.4 Thesis Contributions

In this thesis we study the feasibility of using a smart device accelerometer as a core sensing device in structural health monitoring applications. We focus on accelerometer error mod-

elling for small and large signals. We also develop noise cancellation techniques in addition to applying machine learning for classification even in the presence of errors. The main contributions of this thesis can be summarized as follows:

1. Adopting the simple accelerometer error model that includes only additive noise error in SHM to classify buildings according to damage severity, and we focus on:
   (a) Studying how acceleration error affects displacement estimation, which in turn affects the accuracy of building damage classification.
   (b) Deriving a zero velocity update (ZUPT) technique which is applied to minimize displacement estimation error; a theoretical derivation is presented and validated by shake table experiments. .
   (c) Comparing the performance of a number of commercially available sensors including the accelerometer chip used in most of smart devices; Invensense MPU6500 [28].
2. To develop resilient SHM system and mitigate sensor error, we apply machine learning and compare between several algorithms. In particular we:
   (a) Create a dataset of buildings response to historical earthquakes and label the response of each floor as according to its damage severity.
   (b) Demonstrate the effect of sensor noise on the classification accuracy.
   (c) Present an approach to train classifiers using both ideal and noisy measurements to enhance accuracy.
   (d) Propose classification using convolutional neural network (CNN) that uses the raw input instead of extracted features.
3. For large excitation signals, the simple additive noise error model is shown not to be accurate, and we propose a more general error model that takes into account sampling

time uncertainty introduced by smart devices operating systems. Hence, the main contributions are:

(a) Proposing a novel and experimentally validated error model for smart device accelerometer that includes sampling time uncertainty.

(b) Proposing a MLE technique to estimate the model parameters.

(c) Showing that linear estimation accuracy is close to the best achievable estimation accuracy determined by the CRLB.

1.5 Thesis Organization

The rest of the thesis is organized as follows. In chapter 2, accelerometer simple error model is presented and applied to the SHM use case. Damage detection mechanism is presented and the effect of sensor error is investigated. Error cancellation using ZUPT is derived and sensor selection methodology is illustrated. Chapter 3 discusses the application of machine learning in SHM to mitigate the effect of sensor error. Several machine learning algorithms are compared and we propose a CNN architecture that uses raw input rather than handcrafted features. Besides, noise mitigation using machine learning is compared to conventional high pass filter (HPF) techniques. In chapter 4, we propose a large signal error model that takes into account sampling time jitter caused by the device operating system. Finally, the conclusions and future work are presented in chapter 5.

Chapter 2

Small Signal Accelerometer Error Model: A Case Study

Smart devices are equipped with accelerometers which are capable of capturing vibration. Hence, systems that depend on vibration to assess damages using smart phone based system has been studied recently [29, 14, 30]. In this chapter, we focus on SHM as a use case for smart device accelerometer, including error modeling of the sensor itself, and reflecting how this error affects the damage assessment accuracy.

2.1 Problem Statement

Monitoring the structural health of buildings during and after natural disasters, such as earthquakes provides the public and policy makers with a clear view of the state of critical infrastructure that affects the safety and well being of the population. Previous research on building damage assessment generally falls into one of two main categories: remote sensing techniques and sensor-based technology. In the former, optical images are captured using

spacecraft or aircraft, where before and after image comparisons are performed to assess the damage. This technique is effective in detecting partial to complete collapse of buildings, however it cannot reliably detect incipient collapse because the resolution is too low [31]. On the other hand, sensor based technology uses an internet of things (IoT) network of pre-installed sensors to capture the movement of a building during an event, enabling distributed, accurate and instantaneous monitoring of structures In [32], examples of using sensors such as vibration, temperature, tilt and accelerometer sensors, for SHM targeting bridges are presented, where sensors are interconnected in a wireless sensor network (WSN) configuration to detect damage.

One of the well known methodologies used for damage detection is tracking the shift in the natural frequency of a given structure, which can be measured by an accelerometer. For instance, long term monitoring of Gabbia tower in Mantua showed effectiveness of detecting non reversible structural damages based on frequency shifts [33]. In [34], low cost MEMS accelerometers were used to estimate the fundamental normal mode frequency of steel buildings. A dynamic acceleration measurement system with an accelerometer integrated semipassive radio-frequency identification (RFID) tag was presented in [35], where the network was used to remotely monitor the natural frequency of the structure. In [36], a low-cost distributed system for SHM based on MEMS accelerometers was developed. To enhance the accuracy and reduce the effect of noise and drifts, redundant accelerometer chips were used and online averaging as well as on-board filtering features were available. Data transmission after processing was achieved using a WiFi transceiver. The proposed system focused on real-time monitoring of building dynamic behaviour represented by normal modes of building oscillation. Another similar wireless sensor node was demonstrated in [37]. The main node components were a low-cost accelerometer chip, microcontroller, wireless module and a storage card. The measured parameters are mainly natural frequencies, damping ratios and last floor modal displacement of a tower. In [15, 16], structural health monitoring using a network of smart phones was presented. A mobile *app* was developed that utilizes the

phones internal accelerometers to sense disaster vibration. Synchronization between devices was addressed using network time protocol (NTP).

The work presented in this chapter is based on the fact that other than natural frequency, the IDR is an indicative feature that can be used for SHM as mentioned in [34]. Relative displacement of floors within a given building is used to calculate the IDRs for the building using (2.1). Documents released by the United States federal agencies and civil engineering societies such as FEMA and ASCE relate IDR values to building damage level. Basically, these documents define two main critical thresholds of relative floor displacement of a given building, such that the building can be classified into one of three categories: IO, LS or CP, which indicate that the building is either safe, needs further inspection or unsafe respectively. In other words, measuring the instantaneous relative floor displacement of a given building during an earthquake event is a good indicator of the structure state [2, 3].

$$IDR = \frac{Displacement_{Floor} - Displacement_{NextFloor}}{FloorHeight} \tag{2.1}$$

The cost of an accelerometer depends on several parameters such as dynamic range, linearity, bandwidth, output data rate, output noise and output type, i.e. analog or digital. Based on those specifications, cost ranges from a few dollars to a few thousand dollars. According to [38], sensor output noise is a major contributor to displacement measurement error, which is accentuated by double integration required to calculate displacement from acceleration.

While other technologies such as global positioning system (GPS) are widely used for localization and position estimation, accuracy becomes a major limiting factor in their suitability for structural health monitoring. As will be discussed later, to be useful, IDR values need to be estimated with an accuracy that is within a few centimeters from ground truth. This degree of accuracy is not possible using GPS alone, unless high-end GPS receiver is used, which is much more expensive than a standard GPS receiver [39, 40]. Furthermore, GPS

signals are not available indoors, which mandates outdoor installation for the sensing devices. Another approach to estimate position is to use vision based displacement estimation techniques as mentioned in [41]. Although this approach does not suffer from error accumulation, it faces other challenges such as, measurement error due to heat haze and ground motion, in addition to errors due to dim lighting and optical noise.

Therefore, in this chapter, we focused on studying the limits of using accelerometers to estimate structural displacement for a number of reasons: 1) earthquake event time is relatively short (∼20-30 seconds) which results in bounded accumulated error that can be quantified, 2) accelerometers can work indoors which is not the case of GPS, and 3) accelerometers are not affected by ambient light conditions as compared to cameras.

As mentioned earlier, accelerometer inherent noise is one of the main challenges in displacement estimation. However, noise cancellation can be achieved if some of the disaster signal characteristics are known, such as the fact that a disaster vibration intensity fades gradually and eventually stops at zero velocity and acceleration. In this case, the measured velocity at the end of shaking (EOS) reflects the accumulated error in the preceding samples, which can be used to minimize the estimation error. This technique is known as ZUPT [42, 43].

2.2 Noise Cancellation

An earthquake signal is characterized by stopping at zero acceleration and zero velocity. The EOS instant can be detected when the absolute acceleration is below a certain threshold δ within a specified window of time W as illustrated in figure 2.1. The selection of W is arbitrary, whereas δ is dependent on the sensor noise. If the sensor noise standard deviation (STD) is σ, then we believe selecting $\delta = 3\sigma$ is a reasonable assumption, which indicates that the noise is below that threshold most of the time, In this region, the sensor has true

zero velocity. Any non-zero velocity measured at this time is due to the sensor noise, and is correlated with the noise at shaking time. As mentioned before, using such characteristic in noise cancellation is known in the literature as ZUPT [42, 43].

ZUPT has been used in inertial navigation systems, specifically pedestrian ones [44]. In such systems, navigation devices are mounted on a pedestrian's foot, which is known to be stationary on the ground once every step. The goal of applying ZUPT in that case is to reset the velocity and prevent further error accumulation, which in turn reduces the error in upcoming velocity samples and consequently reduces the error in displacement estimation as well. However, in this work, we are only concerned in correcting displacement estimation for the time window **prior** to the EOS instant, since that window contains the peak relative displacement which reflects the damage state.

2.2.1 Noise cancellation using ZUPT

Accelerometer Model

We use a linear model for accelerometer which has a scale factor and a constant bias. In addition, the model parameter values drift with temperature as mentioned in [23]. In our application, the measurement time window is short and temperature can be assumed to be constant throughout the measurement. Hence, the temperature dependent bias is estimated right before -or after- each measurement using a narrow band low pass filter to be subtracted from the readings and hence, the temperature effect on bias is minimized. On the other hand, the scale factor error in acceleration results in the same scale error in displacement. Usually, scale factor variation versus temperature is constrained by the manufacturer to be small and reported in the device datasheet. Therefore, the error in displacement due to scale factor temperature dependence is negligible compared to other sources of error such as double integration of additive noise.

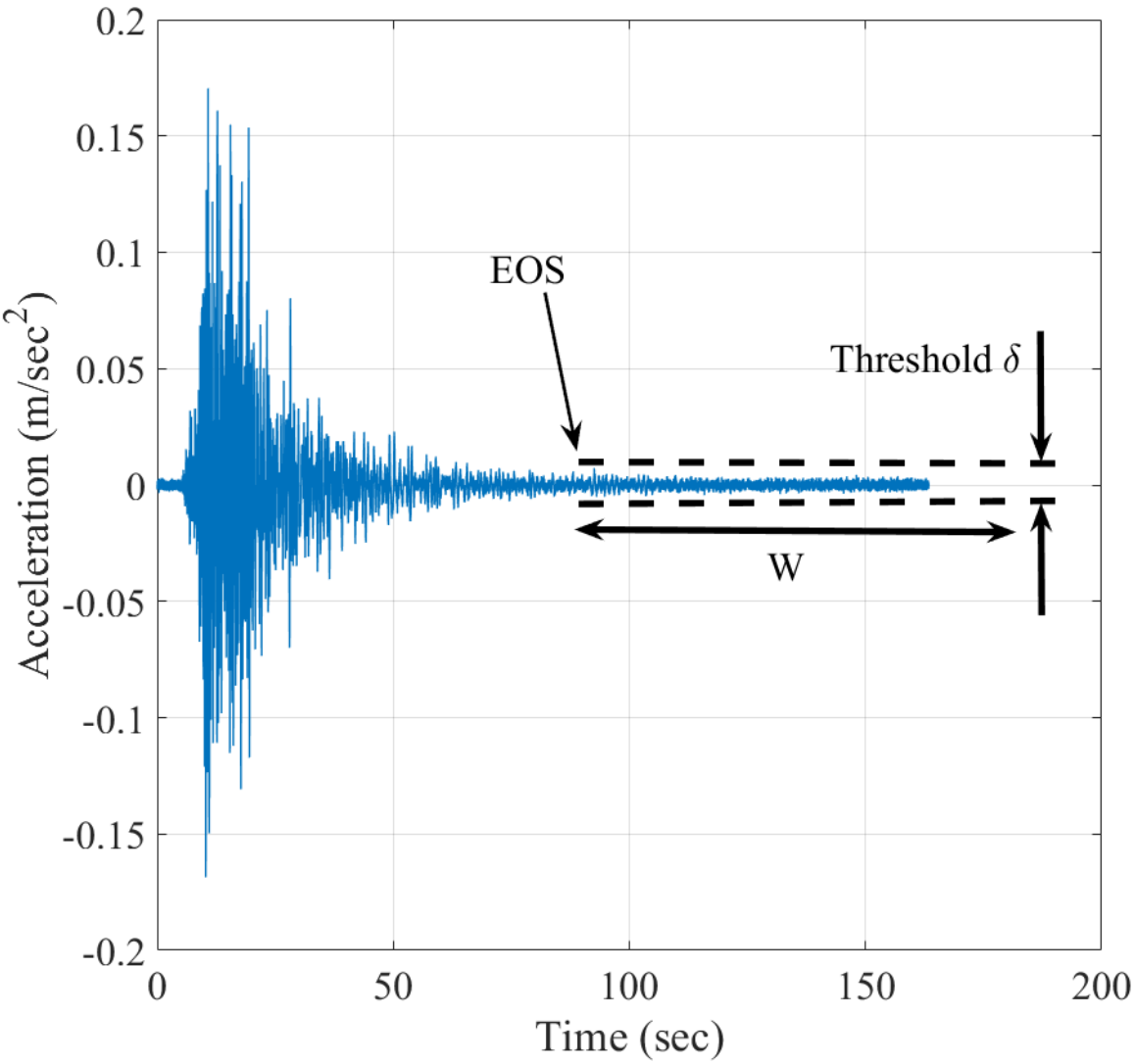


Figure 2.1: EOS instant detection.

Misalignment

In this work, we are concerned with measuring floors' horizontal displacement. Hence, we assume that accelerometers will be oriented to measure only horizontal motion, i.e. gravity will not affect the reading. However, to account for miss-orientation, we consider a constant bias $g \times sin(\theta)$ is added to the measurement and a scale factor $cos(\theta)$ is multiplied by the measurement, where g is the gravitational constant and θ is the misalignment angle, i.e. ideally $\theta = 0$. Assuming linear motion, θ is constant throughout the measurement, hence $g \times sin(\theta)$ is constant and be removed with the sensor temperature varying bias as explained in section 2.2.1. With respect to the scale factor error, assuming θ is small enough, then similar to the temperature effect on scale factor, the scale factor error due to misalignment is negligible compared to other sources of error. More information about orientation and misalignment is found in [24]. However, in case of curvilinear motion, θ is not constant, and removing the gravity component in this case is more complex and can be addressed by using techniques described in [45]. Curvilinear motion is out of the scope of this work and will be investigated in future work.

ZUPT Algorithm

Using rectangular numerical integration method [46], true velocity is expressed by (2.2), where $a_{true}[k]$ is the ground truth k^{th} horizontal acceleration sample and Δt is the sampling time. True displacement is obtained by (2.3) which can be simplified as shown in appendix B to be calculated by (2.4), and the vector form is expressed by (2.5), where A_{true} is the true acceleration vector and $P = [i - 1/2, i - 3/2, ...1/2]^T$.

$$v_{true}[i] = \sum_{k=0}^{i-1} a_{true}[k]\ \Delta t \tag{2.2}$$

$$\begin{aligned} s_{true}[i] =& s_{true}[i-1] + v_{true}[i-1]\Delta t \\ &+ \frac{1}{2} a_{true}[i-1]\Delta t^2 \end{aligned} \tag{2.3}$$

$$= \sum_{k=0}^{i-1} (i-k-\frac{1}{2}) a_{true}[k]\ \Delta t^2 \tag{2.4}$$

$$= P^T A_{true}[0:i-1]\Delta t^2 \tag{2.5}$$

Measurement noise $z[k]$ is considered additive with zero mean, since constant bias is estimated by long term averaging and then subtracted from the measurement [38]. Hence, measured acceleration is expressed by (2.6). Consequently, measured displacement is shown by (2.7). As a result, displacement error is shown by (2.8), where Z is the noise vector.

$$a[i] = a_{true}[i] + z[i] \tag{2.6}$$

$$\begin{aligned} s[i] =& \sum_{k=0}^{i-1} (i-k-\frac{1}{2}) a[k]\ \Delta t^2 \\ =& \sum_{k=0}^{i-1} (i-k-\frac{1}{2}) (a_{true}[k] + z[k])\ \Delta t^2 \\ =& s_{true} + \sum_{k=0}^{i-1} (i-k-\frac{1}{2}) z[k]\ \Delta t^2 \end{aligned} \tag{2.7}$$

$$\begin{aligned} e[i] =& \sum_{k=0}^{i-1} (i-k-\frac{1}{2}) z[k]\Delta t^2 \\ =& P^T Z[0:i-1]\Delta t^2 \end{aligned} \tag{2.8}$$

Let the shaking window length be n samples, where the shaking window is defined as the

time window starting from the beginning of earthquake shaking to the end of shaking (EOS) instant. $v[n]$ is the measured velocity at the EOS instant, which is equal to the accumulated noise since the true velocity at that instant is zero. As shown by (2.9), ZUPT is applied at any given sample i, by multiplying $v(n)$ by coefficient c_i and then adding the result to the displacement measurement.

Note that in traditional inertial systems, ZUPT is triggered when zero velocity is detected to cancel noise at this time instant in order to decrease the error in the next velocity/displacement samples. However, in our work we focus on using the zero velocity information to minimize the error in the *preceding samples* to the zero velocity moment rather than the next ones. We use the fact that measuring the velocity at the zero velocity instant reflects the accumulation of noise of all the acceleration samples that precede the zero velocity instant. Hence, the displacement error at any time instant is correlated to the velocity measured at the EOS instant.

Thus, the modified displacement error is calculated by (2.10), where Q is an $n \times 1$ vector of ones.

$$s_{ZUPT}[i] = P^T A[0:i-1]\Delta t^2 + c_i v[n]\Delta t \tag{2.9}$$

$$\begin{aligned} e_{ZUPT}[i] &= P^T Z[0:i-1]\Delta t^2 + c_i \sum_{k=0}^{n-1} z[k]\Delta t^2 \\ &= P^T Z[0:i-1]\Delta t^2 + c_i Q^T Z[0:n-1]\Delta t^2 \end{aligned} \tag{2.10}$$

Let $\sigma_S^2[i]$ and $\sigma_{S\_ZUPT}^2[i]$ be the mean squared error in displacement at sample i without and

with applying ZUPT respectively as shown by (2.11) and (2.12).

$$\begin{aligned}\sigma_S^2[i] = E[e^2] =& P^T E[Z[0:i-1]Z[0:i-1]^T]P\Delta t^4 \\ =& P^T R_{ii} P \Delta t^4 \end{aligned} \tag{2.11}$$

$$\begin{aligned}\sigma_{S\_ZUPT}^2[i] =& E[e_{ZUPT}^2] \\ =& P^T E[Z[0:i-1]Z[0:i-1]^T]P\Delta t^4 \\ &+ c_i^2 Q^T E[Z[0:n-1]Z[0:n-1]^T]Q\Delta t^4 \\ &+ 2c_i P^T E[Z[0:i-1]Z[0:n-1]^T]Q\Delta t^4 \\ =& P^T R_{ii} P\Delta t^4 + c_i^2 Q^T R_{nn} Q \Delta t^4 \\ &+ 2c_i P^T R_{in} Q \Delta t^4 \end{aligned} \tag{2.12}$$

where $E[.]$ denotes the expectation operator and the noise covariance R_{in} and R_{nn} are given by (2.13) and (2.14) respectively. The value of c_i is calculated such that $\sigma_{S\_ZUPT}^2$ is minimized as shown by (2.15) and (2.16).

$$R_{in} = \begin{bmatrix} r_0 & r_1 & \dots & r_{i-1} & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{i-2} & \dots & r_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{i-1} & r_{i-2} & \dots & r_0 & \dots & r_{n-i} \end{bmatrix} \tag{2.13}$$

$$R_{nn} = \begin{bmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{bmatrix} \tag{2.14}$$

$$\frac{\partial \sigma^2_{S\_ZUPT}}{\partial c_i} = 0 \tag{2.15}$$

$$0 = 2c_i Q^T R_{nn} Q + 2P^T R_{in} Q$$

$$\Rightarrow c_i = -\frac{P^T R_{in} Q}{Q^T R_{nn} Q} \tag{2.16}$$

Assuming that the noise can be modeled as a stationary process as will be illustrated in section 2.3.1, then as shown in appendix A, for noise processes that are characterized by having decaying covariance coefficients, i.e. $r_0 >> r_1 >> ...$ $RQ \approx Q\eta$, where η is calculated by (2.17). Hence, equation (2.16) can be simplified as shown by (2.18). The resulting mean squared error $\sigma^2_{S\_ZUPT}$ is expressed by (2.19).

$$\eta = \sum_{k=-\infty}^{\infty} r_k \tag{2.17}$$

$$c_i = -\frac{P^T Q[0 : i-1]\eta}{Q^T Q \eta} \approx -\frac{i^2}{2n} \tag{2.18}$$

$$\sigma^2_{S\_ZUPT}[i] = (P^T R_{ii} P + \frac{i^4}{4n^2} Q^T R_{nn} Q - \frac{i^2}{n} P^T R_{in} Q)\Delta t^4 \tag{2.19}$$

It is clear that the resulting mean squared error is a function of R which depends on the noise characteristics. For example, in case of white noise, $R = \sigma^2 I$ and $\eta = \sigma^2$, where I is the identity matrix and σ^2 is the noise variance. By substituting in (2.11), the mean squared error without applying ZUPT is calculated by (2.20) and can be simplified by (2.21) for sufficiently large i.

$$\begin{aligned}
\sigma_S^2[i]\Big|_{white} &= \sigma^2 P^T\ I\ P\Delta t^4 \\
&= \sigma^2 P^T P\Delta t^4 \\
&= \sigma^2\Delta t^4 \sum_{k=0}^{i-1}(k+\frac{1}{2})^2 \\
&= \sigma^2\Delta t^4 \sum_{k=0}^{i-1} k^2+k+\frac{1}{4} \\
&= \sigma^2\Delta t^4(\frac{(i-1)(i)(2i-1)}{6}+\frac{(i-1)i}{2}+\frac{i}{4}) \qquad (2.20) \\
&\approx \sigma^2\Delta t^4(\frac{i^3}{3}) \qquad (2.21)
\end{aligned}$$

Similarly, by substituting in (2.19), the mean squared error with applying ZUPT is calculated by (2.22), where G is an $i \times (n-i)$ matrix of zeros. For sufficiently large i, (2.22) can be simplified as shown by (2.23).

$$\begin{aligned}
\sigma_{S_ZUPT}^2[i]\Big|_{white} &= \sigma^2(P^T\ I\ P+\frac{i^4}{4n^2}Q^T\ I\ Q-\frac{i^2}{n}P^T[I\ G]Q)\Delta t^4 \\
&= \sigma^2(P^T P+\frac{i^4}{4n^2}Q^T\ Q-\frac{i^2}{n}P^T Q[0:i-1])\Delta t^4 \\
&= \sigma^2(\sum_{k=0}^{i-1}(k+\frac{1}{2})^2+\frac{i^4}{4n^2}\sum_{k=0}^{n-1}1-\frac{i^2}{n}\sum_{k=0}^{i-1}(k+\frac{1}{2}))\Delta t^4 \\
&= \sigma^2\Delta t^4(\frac{(i-1)(i)(2i-1)}{6}+\frac{(i-1)i}{2} \\
&\quad +\frac{i}{4}+\frac{i^4}{4n}-\frac{i^2}{n}(\frac{(i-1)i}{2}+\frac{i}{2})) \qquad (2.22) \\
&\approx \sigma^2\Delta t^4(\frac{i^3}{3}+\frac{i^4}{4n}-\frac{i^4}{2n}) \\
&= \sigma^2\Delta t^4(\frac{i^3}{3}-\frac{i^4}{4n}) \qquad (2.23)
\end{aligned}$$

At the EOS instant, i.e., at $i = n$, the mean squared error without and with applying ZUPT are expressed by (2.24) and (2.25) respectively. Comparing both equations, it is concluded that using ZUPT reduces the mean squared error by 75% at the EOS instant, i.e. σ_{S_ZUPT} is smaller than σ_S by 50%. For the rest of the chapter we will refer to σ_{S_ZUPT} as σ_S.

$$\sigma_S^2\Big|_{white\ \&\ EOS} = \sigma^2 \Delta t^4 (\frac{n^3}{3}) \tag{2.24}$$

$$\sigma_{S\_ZUPT}^2\Big|_{white\ \&\ EOS} = \sigma^2 \Delta t^4 (\frac{n^3}{12}) \tag{2.25}$$

2.2.2 Experimental Validation

In order to validate the developed algorithm, shake table experiments have been performed. We have used different amplitudes of sinusoidal, frequency sweep and random vibration profiles for a duration of 20 seconds. The sensing device is a smart phone that captures acceleration using its internal accelerometer and transmits the data to a PC. The phone internal accelerometer chip is Invensense MPU6500 which is a 6-axis inertial module that contains 3-axis accelerometer and 3-axis gyroscope sensors, and is widely used in commercial devices [28]. Figure 2.2 shows the experimental setup. MPU6500 chip noise is measured and characterized such that a complete noise model is developed that includes not only white noise, but also bias instability (BI) and rate random walk (RRW), which is described in details in section 2.3.1. That noise model is used to apply ZUPT method.

The motion starts and ends by zero velocity to mimic a seismic event. For each experiment, a vibration profile is selected. Then the whole experiment is repeated for 20 times to generate repeated acceleration measurements for the same vibration profile. Without applying ZUPT, for each trial the displacement is calculated by double summation of acceleration. At each

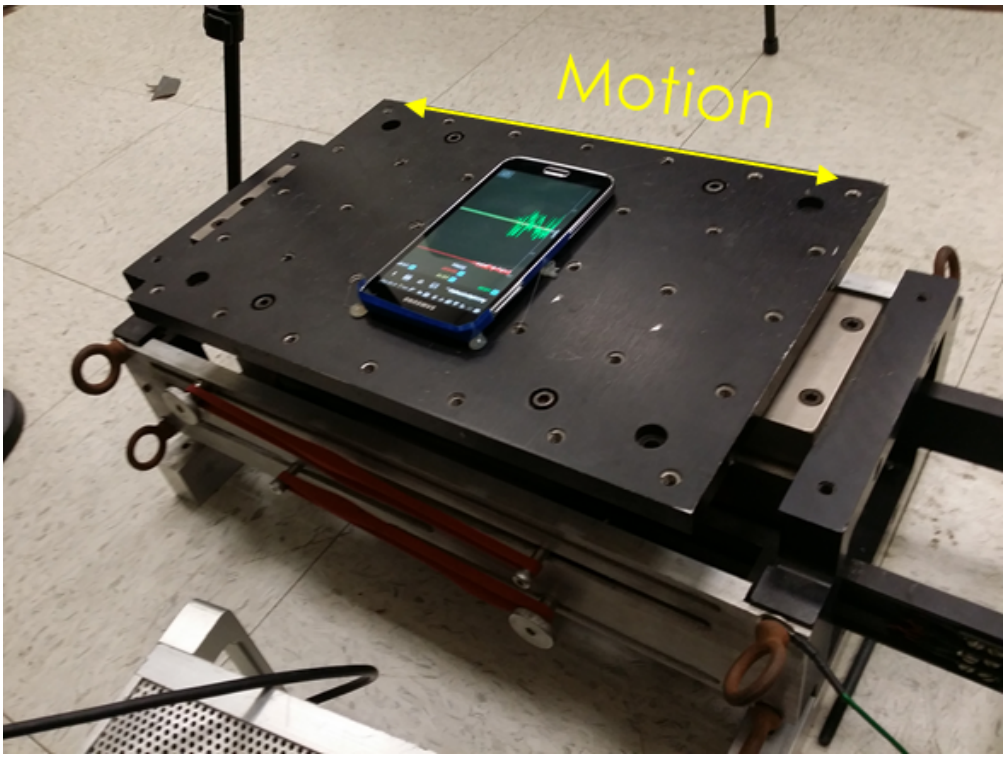


Figure 2.2: Shake table experiment setup.

time sample, the error is calculated by evaluating the standard deviation of the calculated displacements for all the trials of the same experiment. Then, ZUPT is applied to get better displacement estimate from acceleration, and similarly, the error is calculated by evaluating the standard deviation of the calculated displacements. Table 2.1 summarizes the experimental results and reports the error of each experiment at $t = 20$ *sec*, and figure 2.3 shows the STD of theoretical and measured error in displacement σ_S versus time. It is clear that the measured error follows the theoretical one with and without applying the ZUPT algorithm. It is worth noting that the white noise approximation leads to pessimistic results in terms of error reduction performance. As shown in figure 2.3, when using the full noise model, the noise reduction exceeds the expected 50% reduction achievable by only white noise assumption mentioned in section 2.2.1. It is also worth noting that there is a slight difference between the measured and the theoretical model results, the measured error is slightly larger. This is attributed to other sources of error such as time jitter and nonlinearity that have not been included in the model.

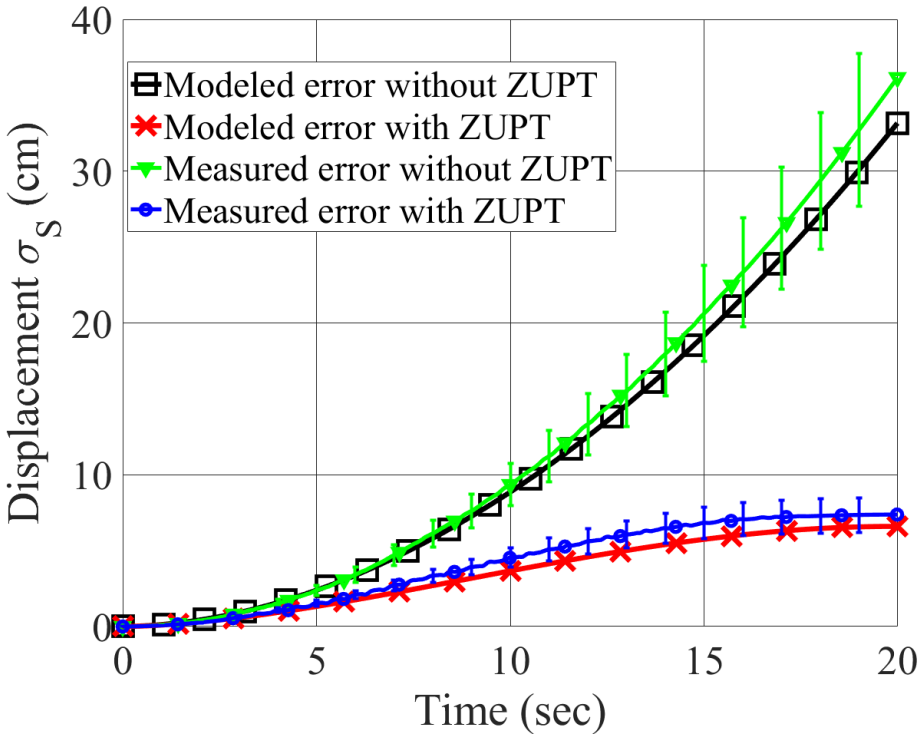


Figure 2.3: Experimental and theoretical displacement error with and without using ZUPT. The error bars represent 99% confidence interval.

Table 2.1: Summary of experimental results. The displacement error is measured at the end of the experiment i.e. at $t = 20$ *sec*.

Experiment	**Amplitude** (m/sec^2)	**Frequency** (Hz)	**Error Without ZUPT** (cm)	**Error With ZUPT** (cm)	**Reduction (%)**
Sine	2	3	32	7	77
Sine	2	4	24.3	8.2	82
Sine	2	5	27.1	7.1	74
Sine	2	7	46.1	5.7	87
Sweep	1	1-30	25	6.6	73
Sweep	2	1-30	42.7	9.3	78
Random	2	-	26.5	9.1	65.4

Table 2.2: Relation between IDR and building state for steel moment frame buildings [2, 3].

IDR %	**Building State**	**Tag**
$< 0.7\%$	Immediate occupancy (IO)	Green
$0.7\% - 5\%$	Life safety (LS)	Yellow
$> 5\%$	Collapse prevention (CP)	Red

2.3 Building Classification

According to government documents, buildings are classified according to their damage state as IO, LS or CP buildings. For instance, table 2.2 lists the IDR limits for steel moment frame buildings which are stated in [2, 3], and the corresponding physical tag used to signal the buildings' post-event condition. Hence, a building's performance can be assessed by comparing its peak IDR to the predefined thresholds. Knowing the floor height, which is 4m in typical US construction, thresholds in IDR corresponds to certain thresholds in relative floor displacement that we denote by $d0$ and $d1$.

Let the true displacement of the two floors be denoted as S_1 and S_2, then the relative displacement D is expressed by equation (2.26). Since each displacement measurement has its own error, then the measured relative displacement D_e is calculated by (2.27), where e_1 and e_2 are the measurement error for S_1 and S_2 respectively.

$$D = S_2 - S_1 \tag{2.26}$$

$$D_e = S_2 + e_2 - (S_1 + e_1) \tag{2.27}$$

$$D_e = S_2 - S_1 + (e_2 - e_1) \tag{2.28}$$

let $X = e_2 - e_1$ then

$$D_e = D + X \tag{2.29}$$

and knowing that the errors in both measurements are not correlated, then the mean squared error in relative displacement measurement is expressed by (2.30). If identical sensors are used, then $\sigma_{S2}^2 = \sigma_{S1}^2 = \sigma_S^2$ and (2.30) reduces to (2.31).

$$\sigma_X^2 = \sigma_{S2}^2 + \sigma_{S1}^2 \tag{2.30}$$

$$= 2\sigma_S^2 \tag{2.31}$$

To evaluate classification accuracy, let B and B_{true} be the building's estimated and true states respectively. Equation (2.29) shows the measured relative displacement of two consecutive floors. The accuracy of the true classification of a building is obtained by evaluating the conditional probability $P(B|B_{true})$ as shown by (2.32).

$$P(B|B_{true}) = \frac{P(B \cap B_{true})}{P(B_{true})} \tag{2.32}$$

where $P(B \cap B_{true})$ and $P(B)$ are expressed by (2.33) and (2.34).

$$P(B \cap B_{true}) = \int \int f_{X,D}(x,d)\ \mathbf{d}d\ \mathbf{d}x \tag{2.33}$$

$$P(B_{true}) = \int f_D(d)\ \mathbf{d}d \tag{2.34}$$

where $f_{X,D}(x,d)$ is the joint probability density function (PDF) of X and D, and $f_D(d)$ is the marginal PDF of D. The integral in (2.33) is done over the area shown in figure 2.4. Besides, limits of the integral in (2.34) is given by table 2.3.

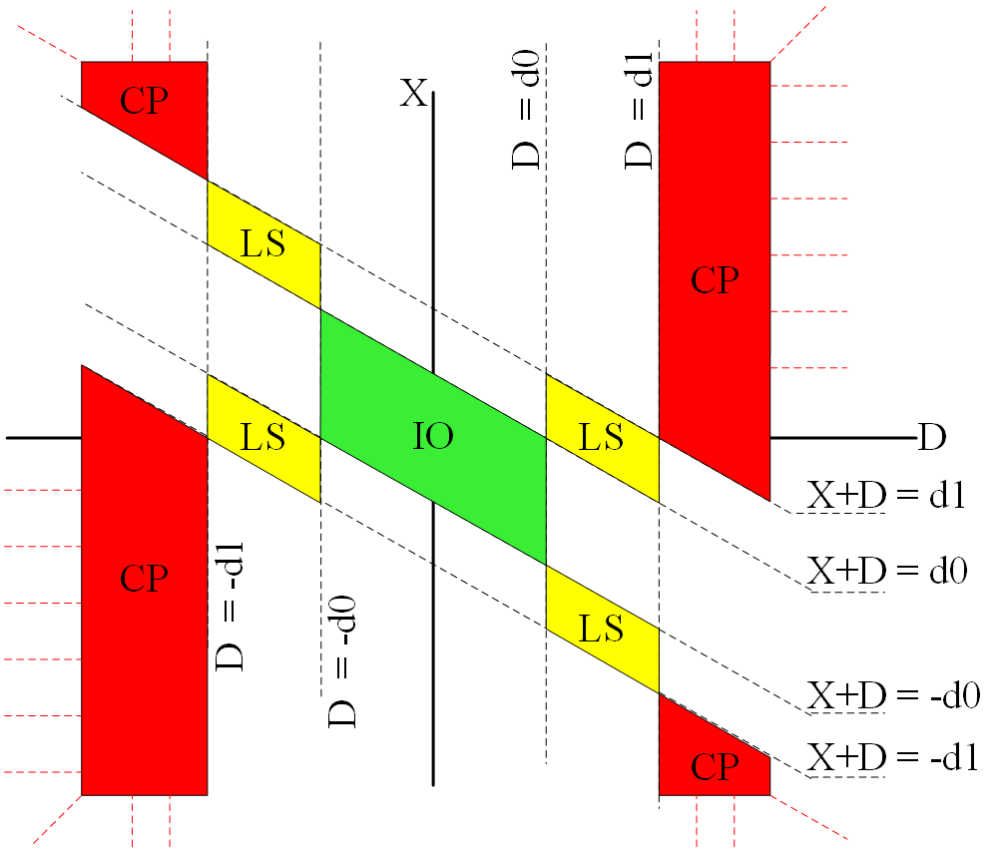


Figure 2.4: Integration regions for different $P(B|B_{true})$.

Table 2.3: Integration interval of equation (2.34) for different $P(B|B_{true})$.

$P(B\|B_{true})$	Integration Region
$P(B\|IO)$	$\|D\| < d_0$
$P(B\|LS)$	$d_0 < \|D\| < d_1$
$P(B\|CP)$	$\|D\| > d_1$

The measurement error only depends on the accelerometer itself and its inherent sources of noise, which is not related to the excitation signal. Hence, noise distribution is considered independent of IDR distribution. As a result, the joint PDF of X and D is expressed by (2.35).

$$f_{X,D}(x,d) = f_X(x) f_D(d) \tag{2.35}$$

where $f_X(x)$ is the marginal PDF of X. We will illustrate below how $f_X(x)$ and $f_D(d)$ can be modeled.

2.3.1 Modeling

Sensor Noise

Acceleration measured by an accelerometer sensor is contaminated by several sources of noise. Referring to [38], noise can be modeled as: constant bias, angle random walk (ARW) (or velocity random walk), BI, and RRW (or acceleration random walk), where each of these is considered an independent Gaussian noise source with certain power spectral density. Since we are only concerned with relatively short durations, higher order noise sources such as drift rate ramp (DRR) are ignored and removed with the constant bias.

According to [47], different noise sources can be modeled as white Gaussian noise shaped with a shaping finite impulse response (FIR) function $H_f(z)$. Since the input of the FIR filter is white Gaussian noise, i.e., wide sense stationary (WSS) noise process, then the generated noise is also WSS.

Case Study: For instance, we consider the Invensense MPU6500, which is a 6-axis inertial module that contains 3-axis accelerometer and 3-axis gyroscope. MPU6500 is widely used in

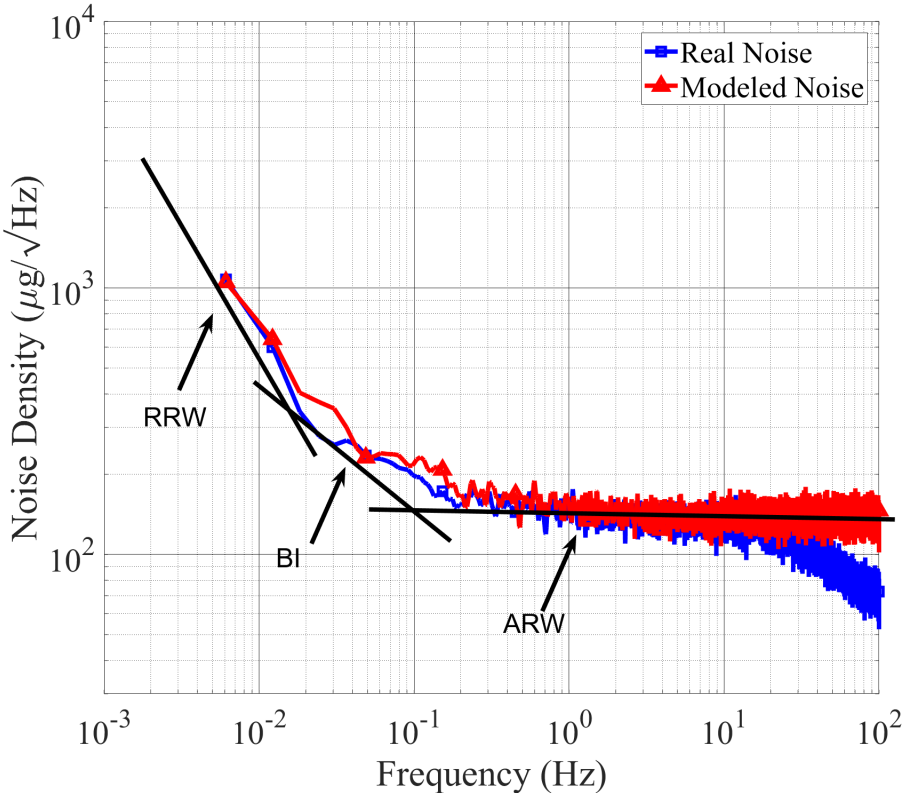


Figure 2.5: PSD of accelerometer real and modeled noise. Real noise is recorded from the sensor x-axis.

commercial devices such as smartphones. To characterize the noise profile of the sensor, the output of the chip was recorded for 12 hours without motion. Using methods described in [48], noise can be modeled as ARW, BI and RRW, and the overall noise covariance matrix R is calculated. Figure 2.5 shows the real (measured) and modeled noise density of MPU6500 accelerometer sensor. It is clear that the noise model matches the real one at low frequency, whereas there is some discrepancy at high frequency, which is due to the fact that the sensor has a low pass filter in the output. However, since in our application the data is double integrated, the low frequency content is the main contributer to the displacement error. Therefore, the discrepancy at high frequency is irrelevant.

In order to provide a reference for comparison, we selected a number of sensors with different noise characteristics as summarized in table 2.4. Figure 2.6 shows the noise spectral density

Table 2.4: Sensors noise characteristics according to their data sheets (except for MPU6500, noise is measured and characterized).

Sensor	**Noise Density** ($\mu g/\sqrt{Hz}$)	**In-run Bias Stability** (μg)	**Detailed PSD** ($\mu g/\sqrt{Hz}$)
MPU6500 (measured)	-	-	700@0.01Hz 200@0.1Hz 150@10Hz
MTI-100 [49]	60	15	-
AXO215 [50]	15	3	-
Mistras1030 [51]	-	-	0.09@10Hz 0.03@100Hz
KB12VD [52]	-	-	0.3@0.1Hz 0.06@1Hz 0.03@10Hz

of the selected sensors based on their data sheets. From the figure, it is clear that some of the sensors noise can be approximated as white noise such as MTI100 and AXO215 sensors, whereas for other sensors higher order noise sources as BI and ARW should be considered.

IDR Distribution

Simulation of building response is conducted in order to model the IDR distribution as a result of earthquake excitation. We consider four and eight-story buildings designed by NIST [53] in Seattle to be representative of steel frame buildings. The buildings have 42.7 m x 30.5 m plans as shown in figure 2.7a. Three-bay perimeter steel special moment frames (SMFs) on each side of the building are used for the lateral load resisting system. The SMFs are designed with reduced beam sections (RBS). With respect to the type of soil, we consider site class D which includes mixtures of dense clays, silts, and sands, which is the most common site class throughout the United States [54]. The seismic design category is D_{max}, i.e., structures are expected to suffer from considerable rotational loads during strong earthquakes [55]. As shown in figure 2.7b, finite element models of the SMFs are created using HyperMesh [56] and analyzed using the commercial code LS-DYNA [57]. The steel is ASTM-A992 and its

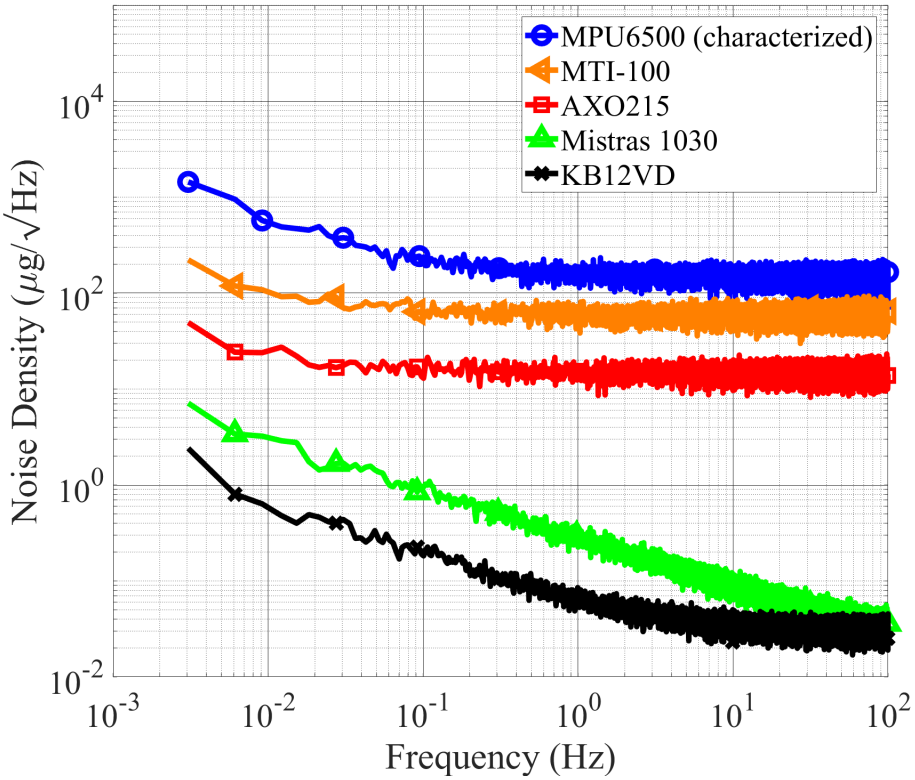
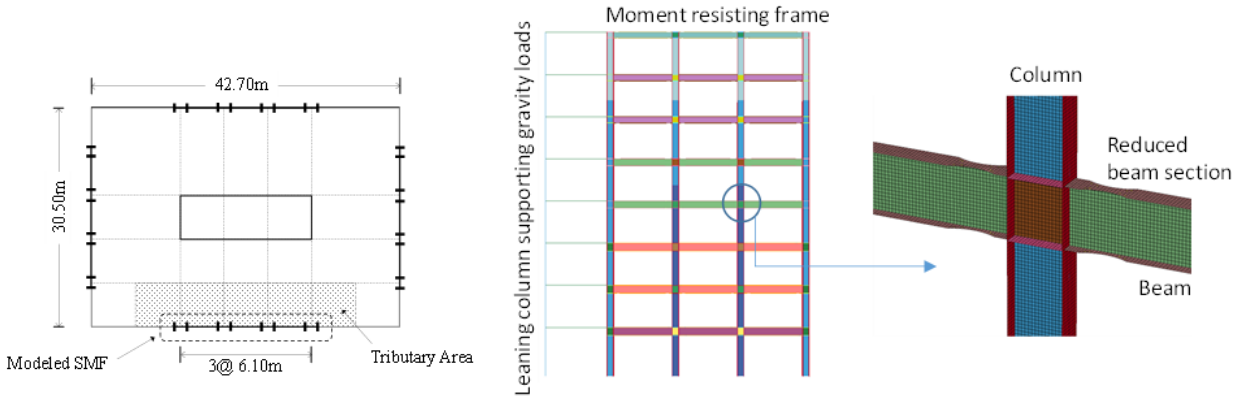


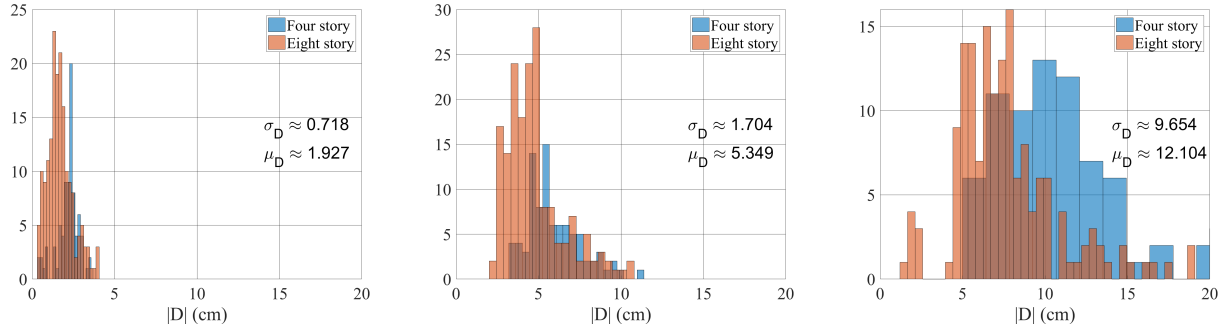
Figure 2.6: Noise density of several accelerometers based on their data sheets (except MPU6500, the x-axis is measured and modeled).



(a) Building plan.

(b) SMF model

Figure 2.7: Building plan and finite element modeling details of perimeter frame.



(a) 50% in 50 years hazard level (b) 10% in 50 years hazard level (c) 2% in 50 years hazard level

Figure 2.8: Histogram of peak relative displacement of four and eight-story moment frame steel buildings.

engineering stress-strain properties are converted into true stress-strain data then assigned to the finite elements as done in [58]. Gravity loads from the tributary area shown in figure 2.7a are directly applied to the frame and the remainder of the gravity loads are applied to a leaning column connected to the SMF by truss members. Mass weighted damping of 2.5% is assumed at the first mode period of the SMFs. Additional modeling details can be found in [58].

The distributions of peak relative displacement are computed for three seismic hazard levels: 2% probability of exceedance in 50 years, 10% in 50 years, and 50% in 50 years. Eleven seismic records are selected from the Far-Field ground motion record set in FEMA [59] and scaled to the three specified hazard levels at the first period of each building, resulting in 33 records for each building. The first period spectral accelerations corresponding to the three hazard levels are 0.55g, 0.26g, and 0.07g for the four-story building and 0.41g, 0.17g, and 0.04g for the eight-story building, respectively. Each building is then subjected to the scaled seismic records for each hazard level and the peak relative displacement is computed. The histogram of peak relative displacement is shown in figures 2.8a through 2.8c. The distribution can be approximated as Gaussian with mean μ_D and variance σ_D^2 that depend on the hazard level, with slight variation depending on the building type.

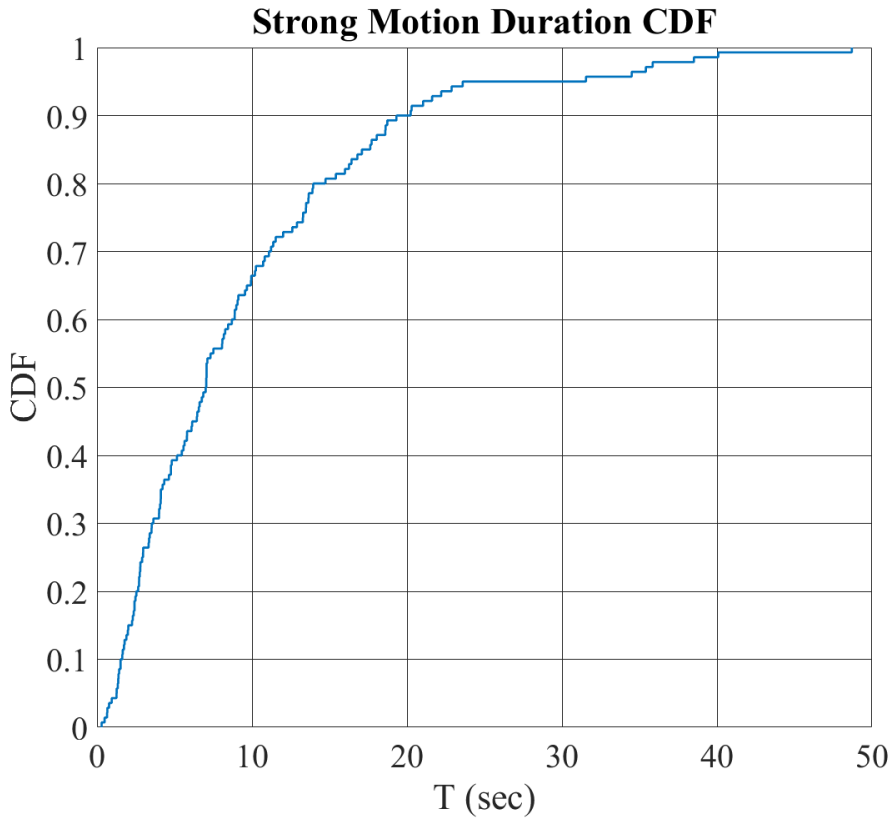


Figure 2.9: CDF of strong motion duration of 140 horizontal components of earthquake ground motion recorded in California reported in [1].

Earthquake Strong Motion Duration

Damage prone buildings will suffer damage during the strong shaking part of the seismic event. As mentioned in [1], there are several definitions for the strong motion duration, which is calculated based on acceleration magnitude or cumulative energy obtained by integrating squared acceleration. In [1], strong motion duration of 140 earthquake records were evaluated, and figure 2.9 shows the cumulative density function (CDF) of strong motion duration.

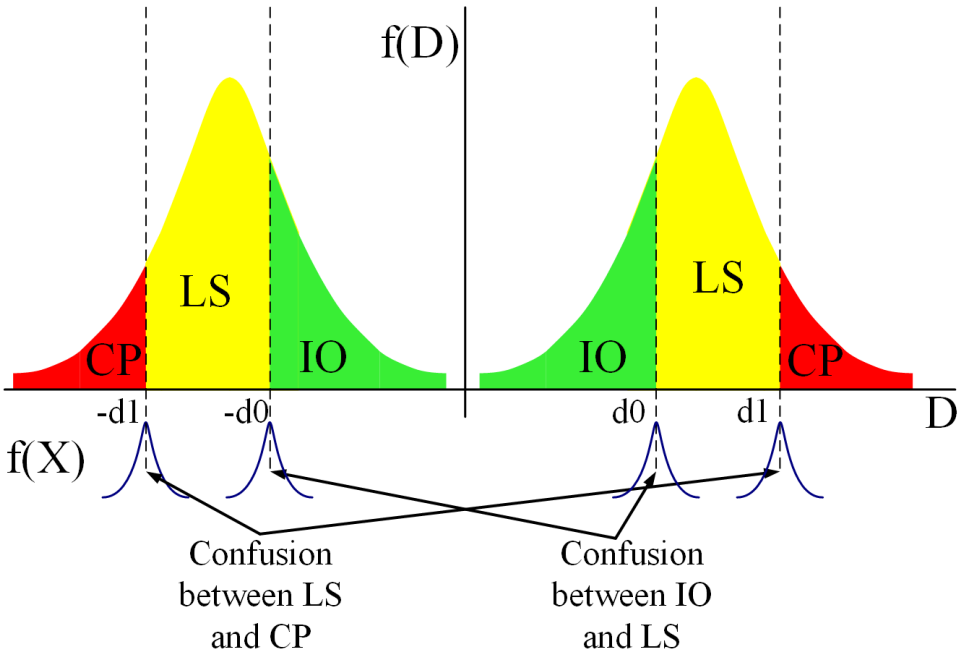


Figure 2.10: Sketch of peak relative displacement Gaussian distribution.

2.3.2 Probability of Classification Error

As mentioned in section 2.3.1, relative displacement measurement error can be modeled as zero mean Gaussian of variance σ_X^2, and relative displacement distribution can be modeled as Gaussian of mean μ_D and variance σ_D^2 which varies according to the hazard level. As a result, substituting into (2.35), the joint probability distribution $f_{X,D}(x,d)$ can be expressed by (2.36).

$$f_{X,D}(x,d) = \mathcal{N}(0,\sigma_X^2)\mathcal{N}(\mu_D,\sigma_D^2) \tag{2.36}$$

where $\mathcal{N}(\mu,\sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. Figure 2.10 shows a sketch of Gaussian peak relative displacement distribution. Classification boundaries are highlighted, where error is expected to occur.

Substituting (2.36) in (2.32) and (2.33), conditional probabilities can be calculated. For instance, for IO buildings, the probability of correct classification is defined as $P(IO|IO)$, whereas probability of error is defined as $P(\overline{IO}|IO)$, i.e. $P(LS \cup CP|IO)$. Similarly, for LS buildings, the probability of correct classification is defined as $P(LS|LS)$, whereas the probability of error is defined as $P(IO \cup CP|LS)$, and with respect to CP buildings, the probability of correct classification is defined as $P(CP|CP)$, whereas the probability of error is defined as $P(IO \cup LS|CP)$. The probability of error of the system is calculated by (2.37), and will be used later in section 2.4 for sensor selection.

$$\begin{aligned} p_e =& (P(LS|IO) + P(CP|IO))P(IO) + \\ & (P(IO|LS) + P(CP|LS))P(LS) + \\ & (P(IO|CP) + P(LS|CP))P(CP) \end{aligned} \tag{2.37}$$

2.4 Sensor Selection

In section 2.3 we showed that the probability of classification error is a function of displacement measurement accuracy, hazard level and strong motion duration. In this section, we demonstrate how a sensor can be selected based on the acceptable probability of error p_e which is calculated by (2.37). The probability of classification error is calculated for each of the sensors mentioned in section 2.3.1.

Figure 2.11 shows the probability of error in buildings classification as a function of strong motion duration in case of 50% in 50 years hazard level. As expected, it is clear that the high accuracy seismic sensors such as Mistras1030 and KB12VD have very small probability of error, and the probability of error increases as sensor accuracy decreases. In the same

figure, we also compare between the simple white noise model, and the more complex model that takes into account other noise components as explained in section 2.3.1. It is worth noting that for high accuracy sensors, using white noise model results in negligible probability of error which is not plotted in the figure. Hence, only the more complex noise model is plotted for the two high accuracy sensors Mistras 1030 and KB12VD. However, for MTI-100 and AXO215, using only the simple white noise model results in probability of error slightly smaller but comparable to the complex model. With respect to MPU6500, there is a larger discrepancy between white noise model and complex noise model results. Intuitively, that result was expected, as mentioned in section 2.3.1 by comparing the noise density curves shown earlier in figure 2.6, it is clear that only MTI-100 and AXO215 noise can be approximated as flat white noise. Similarly, figures 2.12 and 2.13 show the probability of error in case of 10% in 50 years and 2% in 50 years hazard levels respectively. Depending on the acceptable probability of error, the curves presented in figures 2.11 to 2.13 can be used to evaluate the maximum accepted noise density, hence an appropriate sensor can be selected.

2.5 Conclusion

Monitoring structural health of buildings during and after natural disasters is crucial, and directly impacts public safety. Buildings can be added to an IoT network by deploying inertial sensors in civil infrastructure, which facilitates post disaster identification of structurally unsound buildings. In this work, we illustrated how accelerometer sensors can be employed to identify buildings damage state. We presented a theoretical derivation of a ZUPT algorithm that is used to increase displacement measurement accuracy, and consequently increase buildings classification accuracy. The developed algorithm has been validated experimentally using shake table experiments. We investigated the effect of sensors inherent noise on the

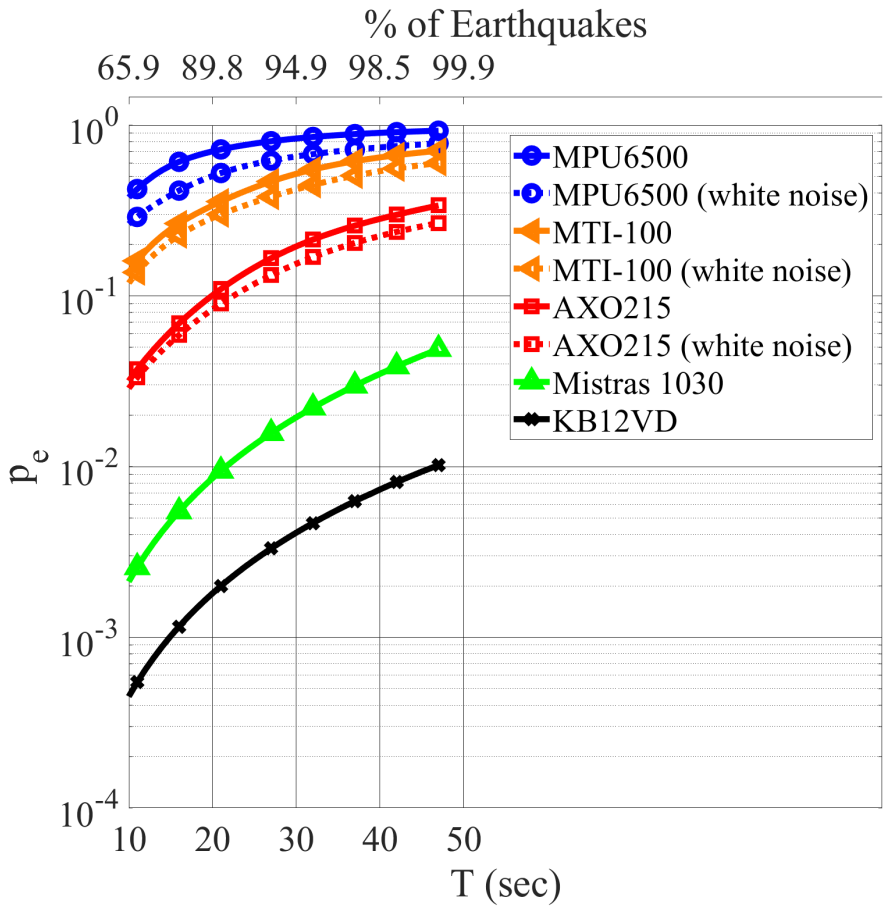


Figure 2.11: Probability of error in classification p_e versus strong motion duration time T for several sensors, for 50% in 50 years hazard level. Sensors noise is modeled according to their data sheets except for MPU6500 we used the noise model mentioned in section 2.3.1.

overall building classification accuracy. The probability of error was calculated as a function of sensor noise density, earthquake duration time and IDR distribution.

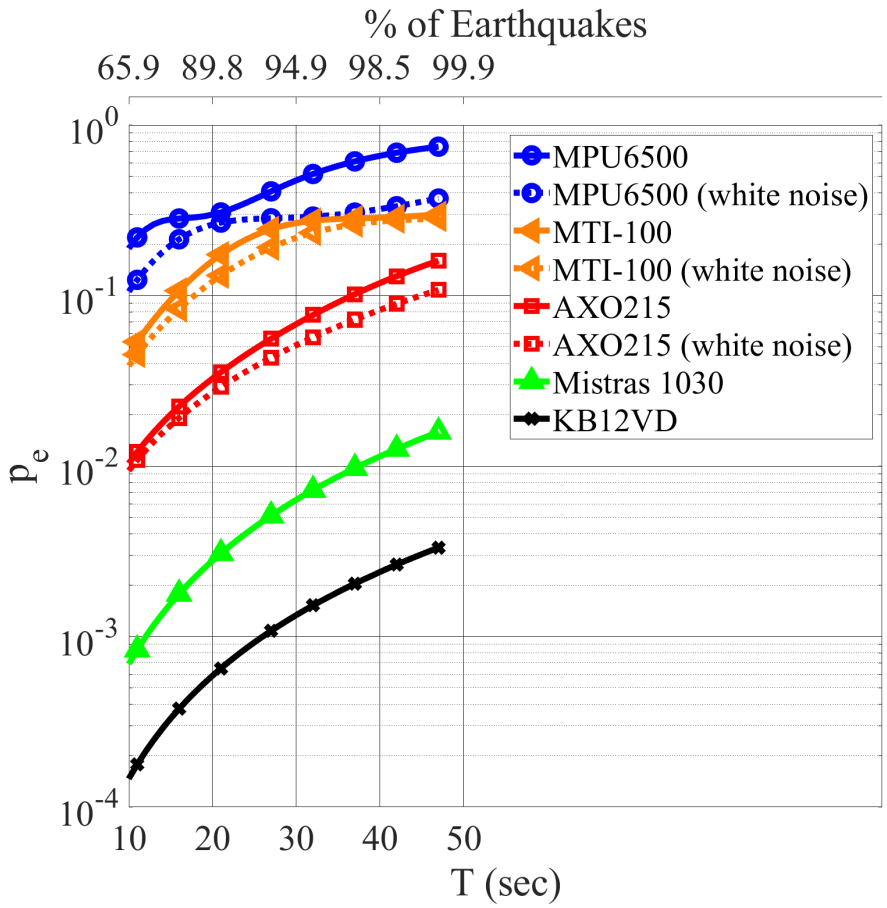


Figure 2.12: Probability of error in classification p_e versus strong motion duration time T for several sensors, for 10% in 50 years hazard level. Sensors noise is modeled according to their data sheets except for MPU6500 we used the noise model mentioned in section 2.3.1.

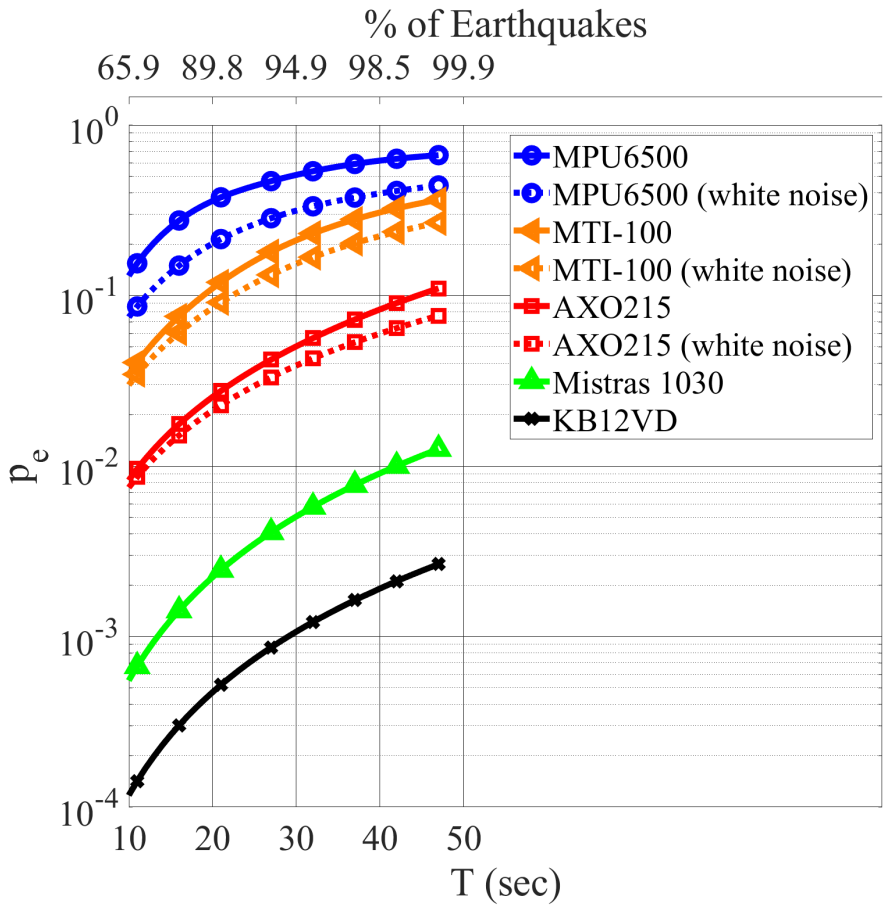


Figure 2.13: Probability of error in classification p_e versus strong motion duration time T for several sensors, for 2% in 50 years hazard level. Sensors noise is modeled according to their data sheets except for MPU6500 we used the noise model mentioned in section 2.3.1.

Chapter 3

Using Machine Learning for Sensor Error Mitigation

In the aftermath of a natural event such as an earthquake, rapid and accurate information from SHM systems enables first responders and reconnaissance teams to address the most vulnerable structures first, and consequently avoid loss of lives and property. Automated monitoring is achieved by using an IoT network of pre-installed sensors to capture the movement of a building during an event, enabling distributed, accurate and instantaneous monitoring of structures [60].

Measuring instantaneous relative displacement of floors within a given building is used to calculate the IDRs for each floor of the building, where IDR is the ratio of the relative floor displacement to the floor height. Documents released by government agencies and civil engineering societies such as FEMA and ASCE relate IDR values to building damage level. These documents define two main critical thresholds of relative floor displacement of a given building, such that the building can be classified into one of three categories: IO, LS or CP, which indicate that the building is either safe, needs further inspection or unsafe respectively

[2, 3].

3.1 Previous Work

In the last two decades, using machine learning algorithms to identify structural damage has been studied extensively. For instance, in [61], a Bayesian probabilistic approach is presented for smart structures monitoring based on the pattern matching approach, neural networks are used for matching the damage patterns for the purpose of detecting damage locations and estimating their severity. Damage was defined as the reduction of inter-story stiffness which induced changes in modal parameters. To add the effect of noise, white noise is added to the simulated acceleration time history of the structure subjected to wide band excitation. Similarly, in [62] a bi-level damage detection algorithm that utilizes dynamic responses of the structure was presented. The approach utilized a neural network as a classifier. Anomaly index (SAI) was proposed to express the amount of changes in the shape of frequency response functions (FRFs) or strain frequency response function (SFRF). SAI was calculated by using the acceleration. Damage was detected from the magnitude of the SAI value, and the location of the damage was identified using the pattern matching achieved by the neural network. In addition, noise was added to prepare a large number of training sets and to examine how noise affects the damage detection results.

A damage identification method intended for steel moment-frame buildings was presented in [19]. The method was based on artificial neural networks and modal variables. The inputs of the network were the mode frequencies and the outputs was the stiffness. After an earthquake, damage index at each story was determined by comparing the stiffness to the initial one which is evaluated in the beginning, i.e. without damage. A simplified finite element model was used to generate the data needed to train the nets. The method was simulated on a 5-story office building under conditions as close as possible to reality. In

addition, white noise was added to simulate the effect of sensor noise.

The selection of the internal nodes activation functions of a neural network was addressed in [63], IASC-ASCE benchmark structure was used, and it was concluded that the best performance was achieved by training neural networks using modal parameters rather than Ritz vectors. Measurement noise is already included in part of the benchmark dataset, specifically case study 2, 4 and 5. In [20], SHM was achieved using a neuro-wavelet technique. Damage was detected based on the continuous wavelet transform (CWT). The input of the system was the mode shapes of structure elements. With respect to noise, Gaussian noise was added and the neural network was trained with noisy and noise free data. We use a similar approach, however our system utilizes acceleration traces rather than deflection, since accelerometers are cheaper and already found in most smart devices nowadays. According to [21], it is expensive to increase the number of the measurement points, hence, one way to address that issue is to predict the unmeasured mode shape data based on a limited number of measured data. Two-stage neural network was adopted, the first is used to predict the unmeasured mode shape and the second was used to localize and estimate the damage severity. In [64], combining neural networks with genetic algorithm for damage detection was presented. A neural network was used to identify the structural main frequencies and mode shapes, then a genetic algorithm was used to identify the damage areas.

In [65], unsupervised learning approach for structural health monitoring self-organizing map (SOM) which is a class of neural networks that is used to create a two-dimensional representation of highly dimensional data. SOM's training process relies only on the internal properties between the inputs and does not require input-output samples and hence a non-parametric damage detection algorithm was proposed, i.e. the algorithm does not depend on system parameters such as modal frequency and employ statistical approaches such as machine learning on raw acceleration traces and extract damage indicators.

In [66], an automated crack detection algorithm using image processing was presented. The

algorithm used machine learning to automatically tune the threshold parameters to identify cracks more accurately. In [67], statistical procedures addressing the feature extraction process were used to identify features capable of detecting structural damage based on accelerometer data. Similarly, [68] focused on automatic fault diagnosis for rotating machinery using machine learning, and recommended support vector machine (SVM) and K-nearest neighbor (KNN) algorithms for classification. In [69], several variants of KNN were proposed for SHM. Furthermore, in [70], SVM was used for SHM, where the residual error (RE) of an auto-regressive (AR) model of acceleration time series was used for damage detection. Moreover, in the same paper the authors used optimization algorithms to find the best parameter values of the SVM. In [71], an online version of SVM for structural health monitoring was proposed. The input to the SVM are the statistical features of the acceleration, velocity and displacement. Calculating velocity and displacement is done by numerical integration of acceleration, and low frequency noise is cancelled by a HPF. In [72, 73], CNN were used to estimate damage severity based on accelerometer data. CNN was used as a better alternative to other algorithms that depend on hand-crafted features. In addition, the same concept was applied to wireless sensors networks in [74]. The damage detection method operates directly on the raw vibration signals without filtering or pre-processing. A decentralized damage detection algorithm was proposed based on 1D CNN and it involved training an individual 1D CNN for each wireless node.

Some of the prior work assumed the use of high quality accelerometers to generate acceleration traces. However, a key point to note is that recently there has been a shift from using sparse, high quality accelerometers, to using a larger number of distributed lower quality, cheaper sensors. For example, the feasibility of using smart phones to quantify seismic damage to buildings has been studied in [75, 16]. In addition, in [22], the planar response of a smart device subjected to friction forces induced by an underlying moving plane due to an earthquake motion has been studied. In [29], built-in smart phones accelerometers were used to estimate buildings displacement during shaking. Since these accelerometers

Table 3.1: Relation between IDR and building state for steel moment frame buildings [2, 3].

IDR %	**Building State**	**Tag**
$< 0.7\%$	Immediate occupancy (IO)	Green
$0.7\% - 5\%$	Life safety (LS)	Yellow
$> 5\%$	Collapse prevention (CP)	Red

are low quality consumer grade ones, noise cancellation was achieved by using HPF. While filtering does improve system performance by reducing integrative noise, it comes at a risk of removing significant information from the signal as well.

Most of the prior work used change in modal parameters to detect damage. In this work, we use IDRs as a damage indicator. We created a finite element model of a 4 and 8-story buildings, which is excited by historic earthquake traces and the response of each floor is simulated. Based on the IDRs, each floor is labeled by its damage level. We use the generated dataset to train several machine learning classifiers. We also focus on the effect of noisy measurements on classification accuracy. We present alternative methods to mitigate the effect of noise, without resorting to HPF.

3.2 Dataset Creation

According to government documents, buildings are classified per their damage state as IO, LS or CP buildings. For instance, table 3.1 lists the IDR limits for steel moment frame buildings which are stated in [2, 3], and the corresponding physical tag used to identify the buildings' post-event condition. Hence, a building's status can be assessed by comparing its peak IDR to the predefined thresholds. Knowing the floor height, thresholds in IDR correspond to thresholds in relative floor displacement.

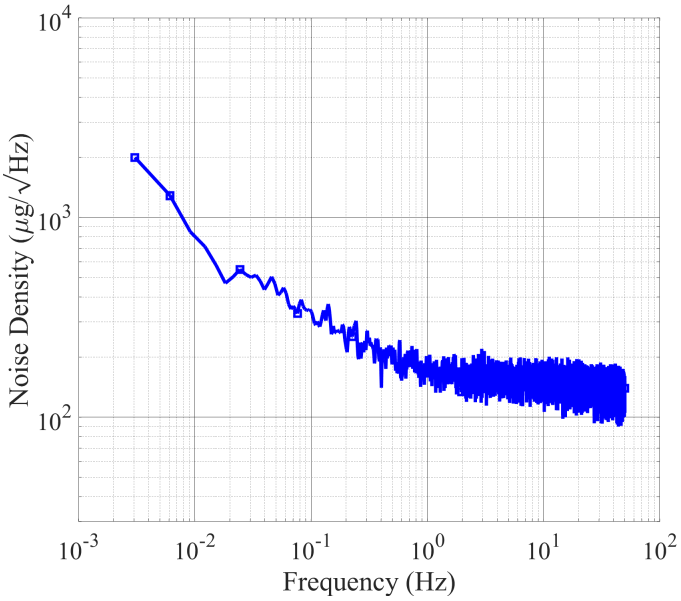


Figure 3.1: Noise density of MPU6500 accelerometer.

3.2.1 Buildings Simulation

Simulation of building response is conducted in order to generate training samples. We consider four and eight-story buildings designed by NIST [53] in Seattle to be representative of steel frame buildings. Three-bay perimeter steel special moment frames (SMFs) on each side of the building are used for the lateral load resisting system as shown earlier in chapter 2 in figures 2.7a and 2.7b. The SMFs are designed with reduced beam sections (RBS). With respect to the type of soil, we consider site class D which includes mixtures of dense clays, silts, and sands, which is the most common site class throughout the United States [54]. Finite element models of the SMFs are created using HyperMesh [56] and analyzed using the commercial code LS-DYNA [57]. The steel is ASTM-A992 and its engineering stress-strain properties are converted into true stress-strain data then assigned to the finite elements. More details about the simulation setup are found in [58, 30].

Based on the resulting IDRs, and using table 3.1, a dataset is created that includes relative floor acceleration labeled with damage level, i.e. either IO, LS or CP. In order to model real

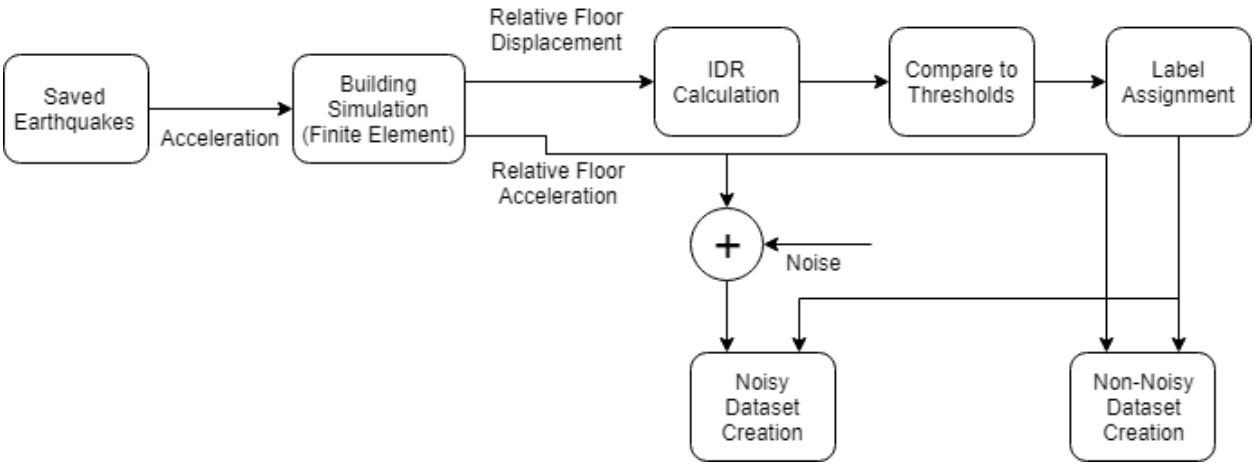


Figure 3.2: Datasets creation block diagram.

cheap sensors, we have recorded the output noise of MPU6500 accelerometer chip [28] which is an off-the-shelf inertial sensor chip used in consumer devices such as smart phones. Figure 3.1 shows the power spectral density of the MPU6500 sensor. According to [38, 30] we have assumed an additive noise model and added the recorded noise to the simulated acceleration data. Hence, two datasets are created: 1)Non-noisy dataset and 2)Noisy dataset. For each dataset, 70% of the data is used for training and 30% is used for validation.

Figure 3.2 summarizes the proposed dataset creation technique.

3.3 Classification Algorithms

3.3.1 Features Selection

Typically, acceleration data time series is represented by its discriminative features which are more convenient as inputs to classifiers. In [70], the following features were extracted from the RE of AR model: maximum, minimum, difference of maximum and minimum, mean, mean of absolute, standard deviation, skewness, kurtosis, root mean squared, form factor, crest factor, kurtosis factor, pulse factor, margin factor, upper and lower control limits.

The formulas needed to calculate each of them are mentioned in [70]. These features are assumed to be relevant to damage information. However, since in this work we are concerned with additive noise that contaminates the measured acceleration. Hence, we propose three different methodologies to mitigate the effect of additive noise: 1) Identification of noise sensitive features, 2)Using all the features while training using noisy and non-noisy datasets and 3)Using raw noisy and non-noisy datasets and allow the machine learning algorithm to identify the discriminative features that are immune to noise. The results of the proposed methods are presented in section 3.4.

3.3.2 Classification Algorithms

Since the dataset was created by comparing the peak relative displacement to certain standard thresholds, then the ordinary technique to estimate the damage label is to calculate relative displacement from relative acceleration, i.e. by double integration, and then compare the maximum value to the threshold. Ideally, without adding any noise to acceleration, that technique achieves 100% accuracy, which can be demonstrated by applying it to the non-noisy dataset. However, the same technique is not expected to achieve high accuracy when applied to the noisy dataset, since double integration of noise introduces large errors, that severely reduce classification accuracy.

Machine learning classifiers have significant potential in processing noisy data. Several machine learning classifiers have been recommended for SHM in literature. For instance, in [69], several variants of KNN were proposed. Besides, SVM was used in [70] for damage detection where several optimization algorithms were used to find the best parameters of the SVM, and it was concluded grid search outperforms other searching algorithms.

As mentioned in [72, 74], CNN is an appropriate method to capture damage patterns in the response of buildings to vibration. In this work, we propose a CNN to classify buildings

according to the level of damage based on relative floor acceleration. In addition, for the sake of comparison, we also use the the grid search SVM algorithm described in [70], KNN algorithms and double integration (DI) with HPF noise cancellation which is illustrated later.

Proposed CNN Architecture Based on earthquake strong motion durations reported in [1], we assumed the maximum monitoring duration is 50 seconds, which indicates that the number of time samples is 5000 at 100Hz sampling rate. The CNN consists of 7 layers that are described in details in table 3.2. We also use one-hot encoding to the output, i.e. the three classes are represented by three output nodes. The network uses two convolutional and maxpooling layers.

Table 3.2: CNN network architecture.

Layer	**Type**	**Parameters**	**Activation**
0	input	size = 5000	-
1	Conv1D	Kernel size=50 Filters=4	relu
2	Maxpooling	Pool size=20	-
3	Conv1D	Kernel size=10 Filters=10	relu
4	Maxpooling	Pool size=2	-
5	Flatten	-	-
6	Dense	Size=2	tanh
7	Dense	Size=3	tanh

Noise Cancellation Using HPF For the sake of comparison, we illustrate how HPF is used to minimize the effect of noise on classification using the ordinary DI method. As mentioned in [30], acceleration is contaminated with additive white noise. Evaluating displacement from acceleration requires double integration which amplifies the low frequency noise significantly. Hence, using a HPF reduces the effect of noise in the DI method if a proper cut-off frequency is selected. We denote this method as DI+HPF. The higher the cut-off frequency, the more noise cancellation is achieved. However, increasing the cut-off

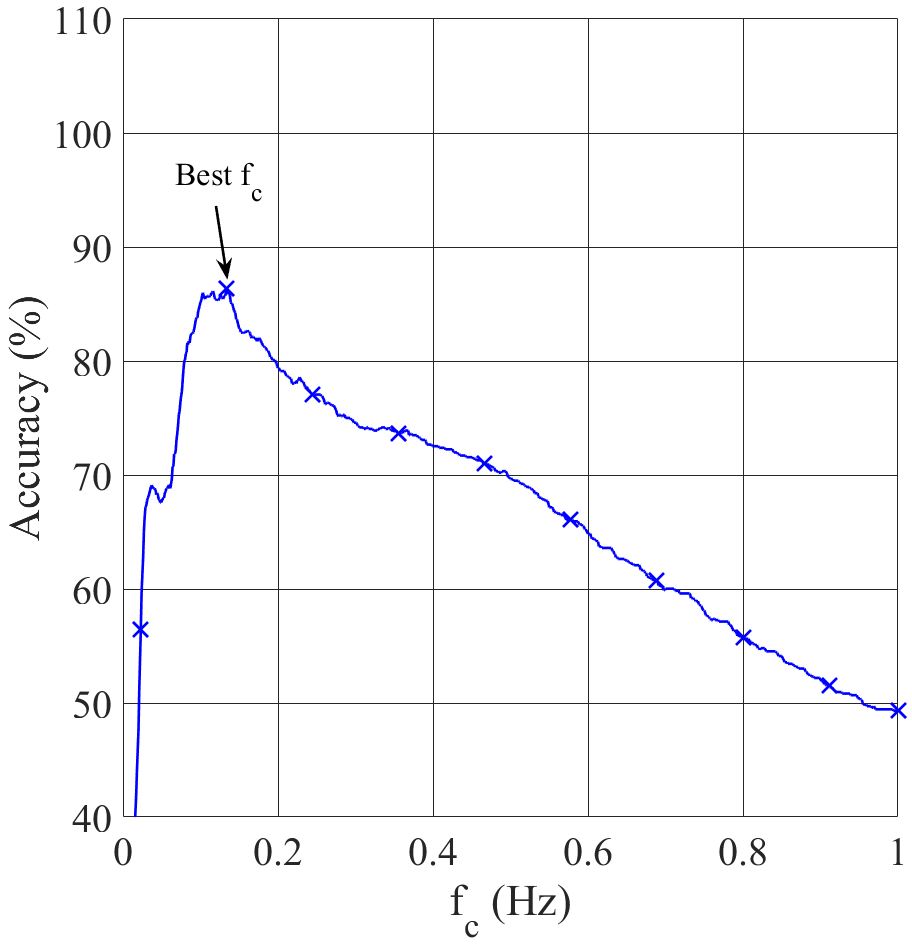
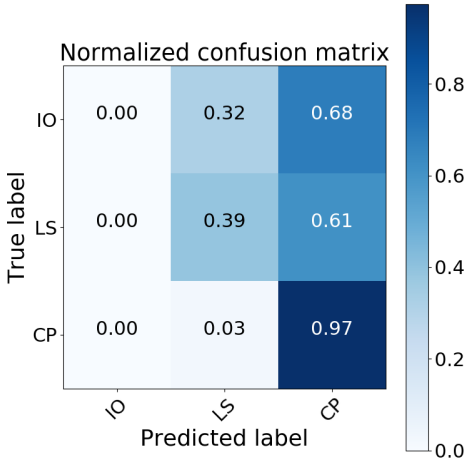


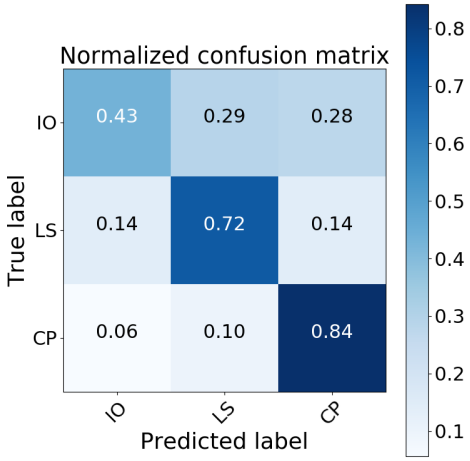
Figure 3.3: Effect of changing HPF cut-off frequency on classification accuracy.

frequency affects the signal and removes its low frequency content, which in turn introduces an error.

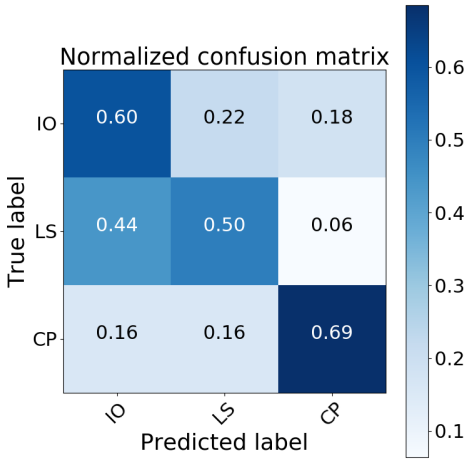
Therefore, accuracy is expected to increase by increasing the cut-off frequency due to noise cancellation, and then decrease when removing the low frequency content of the signal becomes the dominant source of error. Figure 3.3 shows the effect of changing the HPF cut-off frequency on the classification accuracy.



(a) DI



(b) SVM



(c) KNN

Figure 3.4: Confusion matrices in the presence of noise without applying any noise cancellation technique.

3.4 Results and Discussion

As expected, applying DI on the noisy dataset results in low classification accuracy due to the accumulated error and is demonstrated by the confusion matrix shown in figure 3.4a. In addition, training machine learning algorithms with the non-noisy dataset and validating using the noisy one results in poor performance as illustrated by figures 3.4b and 3.4c. Below we discuss three different techniques to enhance the classification accuracy in the presence of noise.

3.4.1 Removing Noise Sensitive Features

As mentioned in [76], noise in measurements is hard to model and is therefore challenging to separate signal from noise. Specifically, for classification-oriented problems, such as activity recognition, appropriate features are derived from raw sensor data, which is then hand-crafted and fed into a classifier for training. This process is time consuming and requires extensive experiments to generalize well, besides being data dependent, i.e. features selection must be repeated if the dataset is changed.

Hence, we apply the same approach of hand crafted features, and each of the features mentioned in section 3.3.1 is studied in the absence and presence of noise to identify those most sensitive to noise. The study includes the distribution of the data represented by its histogram. We focus on two main things in the histogram: 1) the effect of additive noise on the feature distribution, which is indicated if there is a significant change in the histogram of the feature with and without noise and 2) the ability of the feature to discriminate between classes, and this is indicated by comparing the histograms of the feature across classes. This methodology is applied to the structural health monitoring dataset created in section 3.2 and the two most affected features by adding noise are pulse factor and margin factor which

are calculated by (3.1) and (3.2) respectively, where N is the window size [70]. Histograms of these features are shown in figures 3.5 and 3.6 respectively. By removing these features from the features list, better immunity to noise is achieved without much loss in classification accuracy since these features are slightly discriminative as shown by figures 3.7 and 3.8. It is clear that these two features are not well differentiating between IO and LS classes, while marginally useful to identify CP class.

$$PulseFactor = \frac{max(RE) - min(RE)}{1/N \sum_{i=1}^{N} |RE_i|} \tag{3.1}$$

$$MarginFactor = \frac{max(RE) - min(RE)}{[1/N \sum_{i=1}^{N} \sqrt{|RE_i|}]^2} \tag{3.2}$$

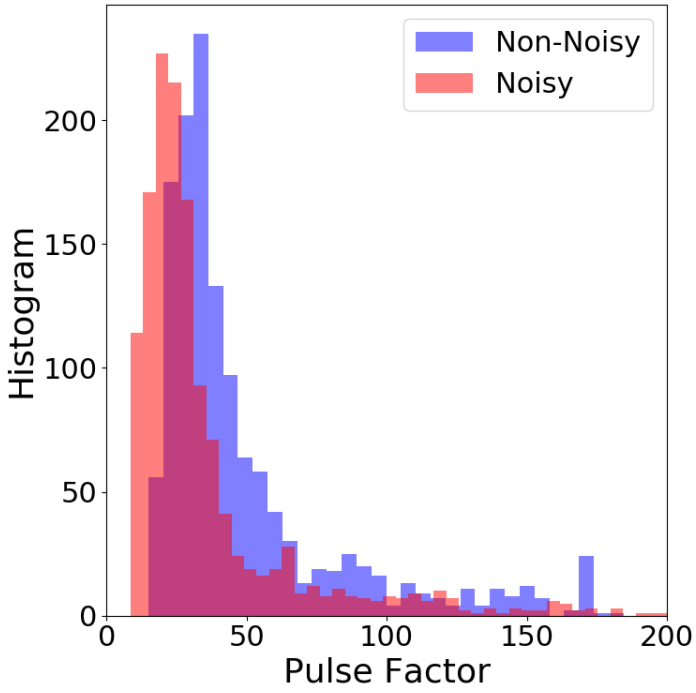


Figure 3.5: Histogram of the pulse factor feature.

SVM and KNN are trained using the non-noisy dataset after removing the noise sensitive

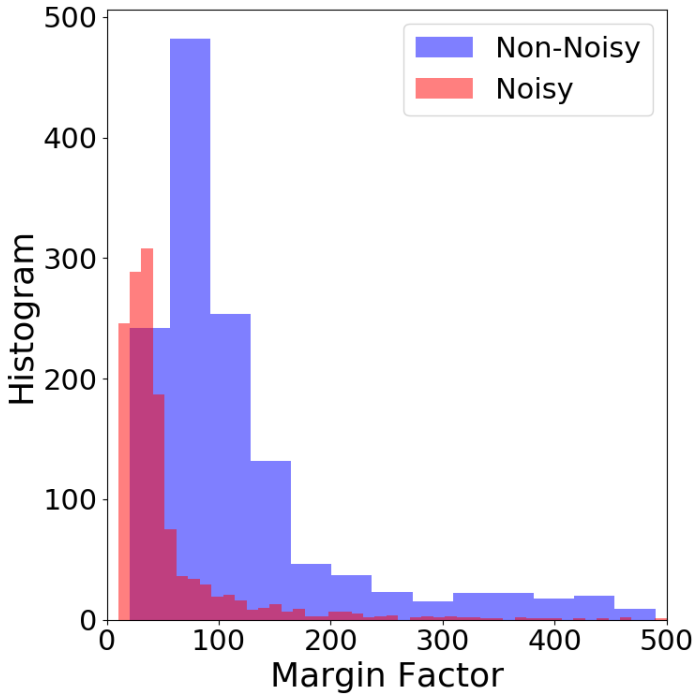


Figure 3.6: Histogram of the margin factor feature.

features, then validated by the noisy one. This technique results in performance enhancement as shown by figures 3.9 and 3.10 respectively. The performance enhancement is clear by comparing the figures to figures 3.4b and 3.4c.

3.4.2 Training Using Both Datasets

Another approach is to train the classifier using both noisy and non-noisy training datasets, and validate using the noisy validation dataset. In this case, the classifier automatically ignores the noise sensitive features while keeping the discriminative ones. The validation confusion matrices of SVM and KNN are shown in figures 3.11 and 3.12 respectively. It is clear that training using both datasets results in better performance when compared to the base case, however manually removing the noise sensitive features results in better accuracy.

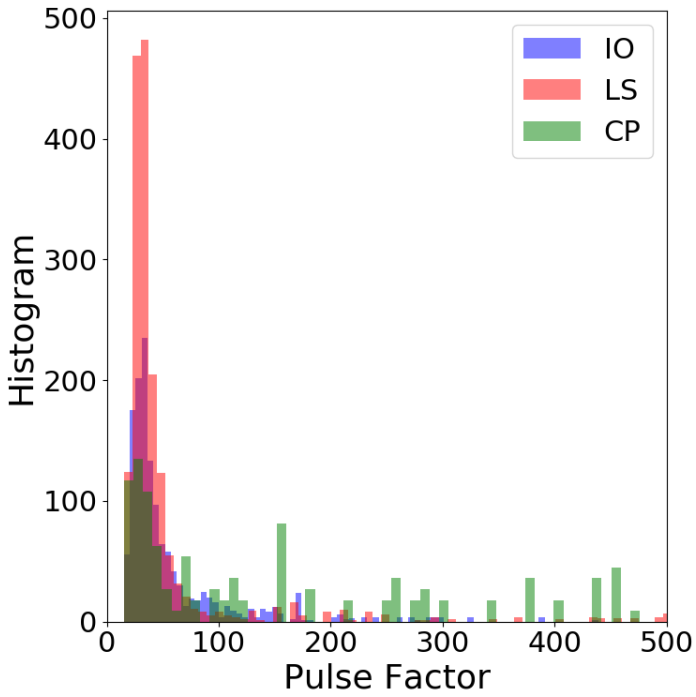


Figure 3.7: Histogram of the pulse factor feature across the three classes.

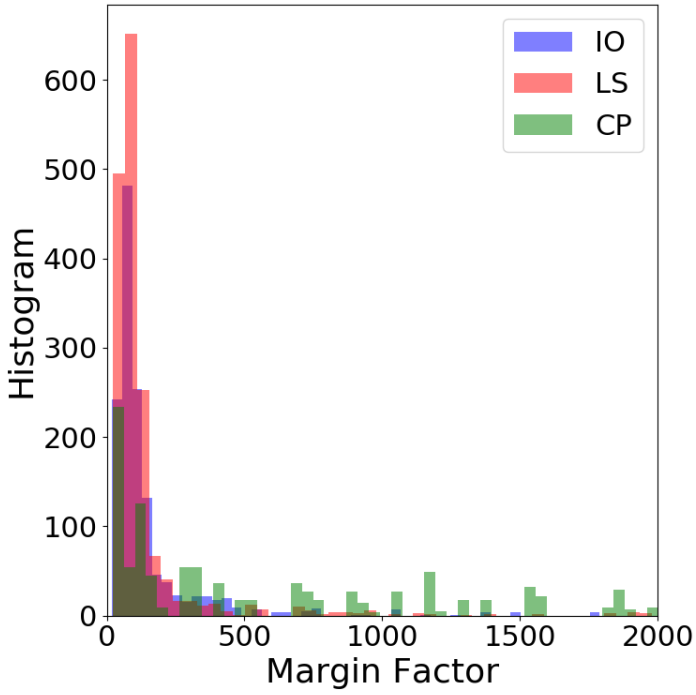


Figure 3.8: Histogram of the margin factor feature across the three classes.

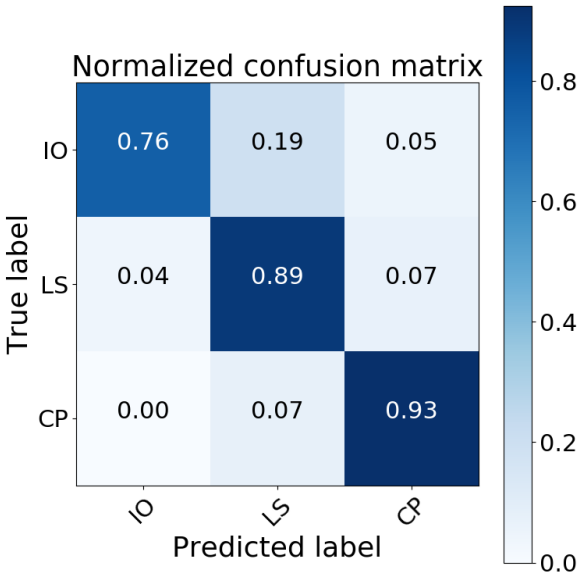


Figure 3.9: Confusion matrix of SVM after removing the sensitive noise features. Algorithm is trained using the non-noisy training dataset and validated using the noisy validation one.

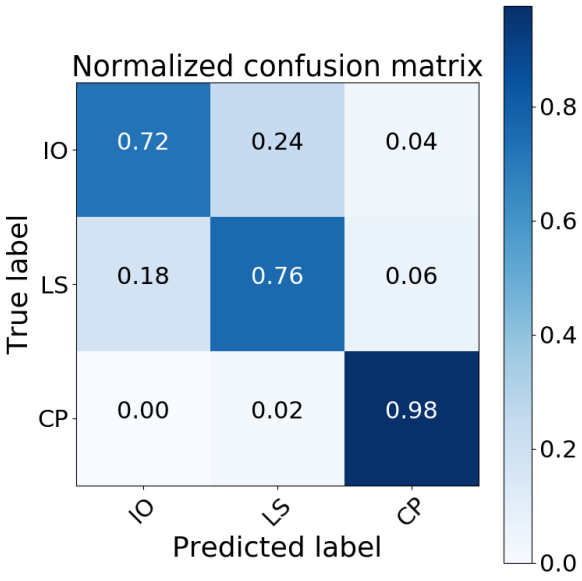


Figure 3.10: Confusion matrix of KNN after removing the sensitive noise features. Algorithm is trained using the non-noisy training dataset and validated using the noisy validation one.

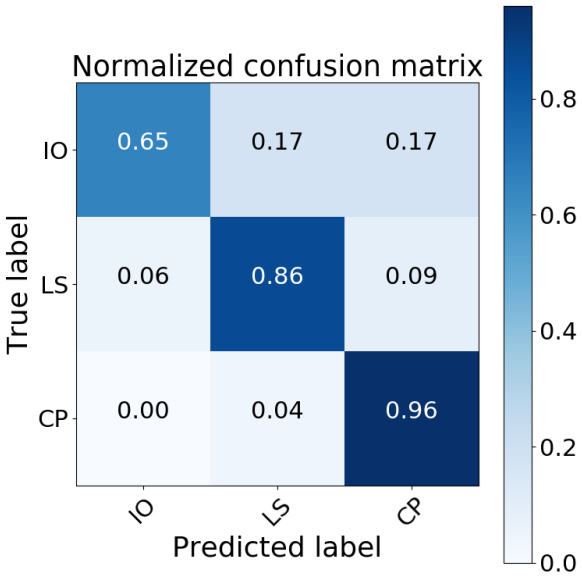


Figure 3.11: Confusion matrix of SVM trained using the noisy and non-noisy training datasets and validated using the noisy validation one.

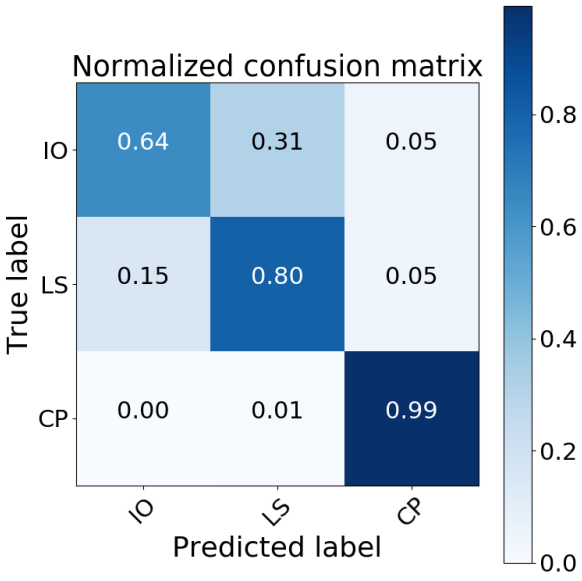


Figure 3.12: Confusion matrix of KNN when trained using the noisy and non-noisy training datasets and validated using the noisy validation one.

3.4.3 Proposed CNN

The CNN network described in section 3.3.2 is trained using both datasets similar to the approach mentioned in section 3.4.2. However, instead of training using features, the raw input data is used. The CNN network is assumed to be capable of identifying the discriminative features automatically. As shown in figure 3.13, the proposed algorithm outperforms the other techniques mentioned earlier.

3.4.4 HPF Noise Cancellation

As mentioned in section 3.3.2, HPF can be used for noise cancellation. Then, regular double integration is performed on the filtered data to calculate displacement which is directly used for classification. Figure 3.14 shows the confusion matrix after applying this technique, when using the optimal cut-off frequency.

3.4.5 Effect of Noise Level on Classification Accuracy

It is clear that the proposed CNN method outperforms other methods. We demonstrate how changing the noise power affects the classification accuracy in the proposed CNN and the traditional DI+HPF techniques by multiplying the additive noise by a scale factor as shown in figure 3.15. It is clear that the accuracy of the DI+HPF technique is less than but comparable to the proposed CNN technique at low noise levels. However, at high noise levels, the proposed CNN method results in better classification accuracy and is more resilient.

The CNN method achieves high accuracy regardless of filtering the training data. As shown in figure 3.16, high pass filtering the data prior to training results in almost the same accuracy regardless of the high pass filter cut-off frequency, which outperforms the DI+HPF method.

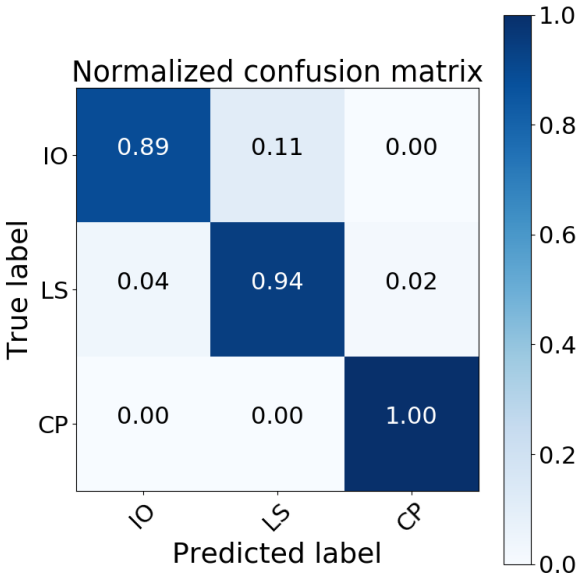


Figure 3.13: Confusion matrix of the proposed CNN when trained using the noisy and non-noisy training datasets and validated using the noisy validation one.

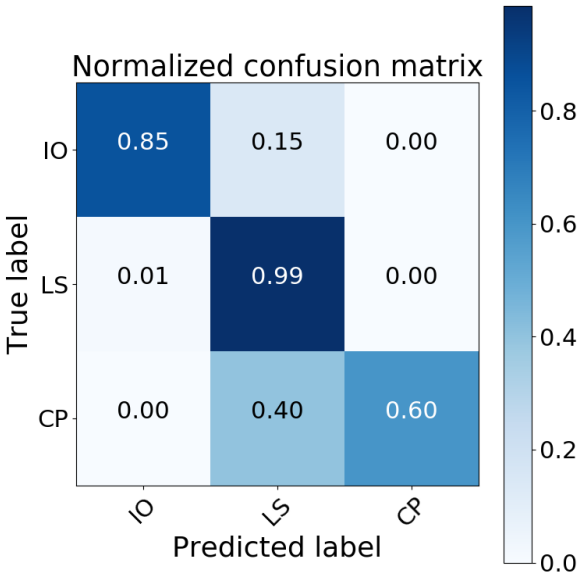


Figure 3.14: Confusion matrix of DI method after applying HPF to remove noise.

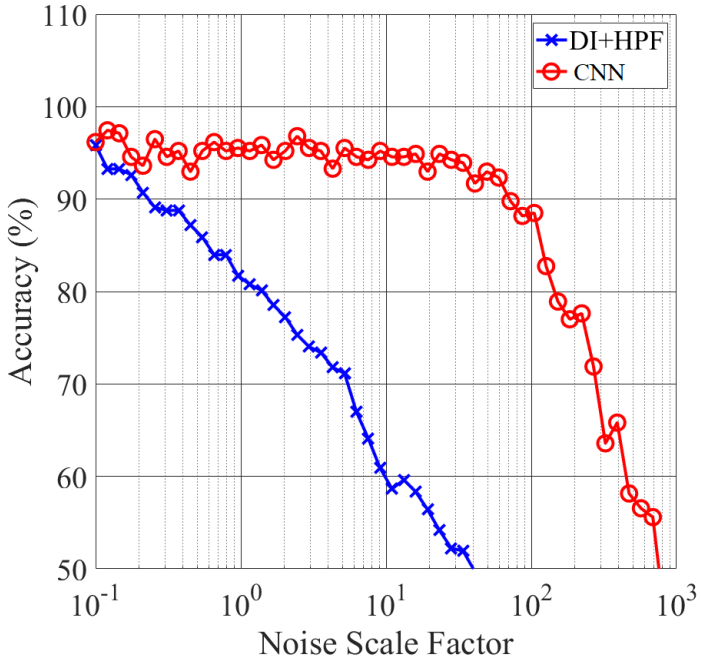


Figure 3.15: Classification accuracy versus noise scale factor for the HPF and the proposed CNN techniques.

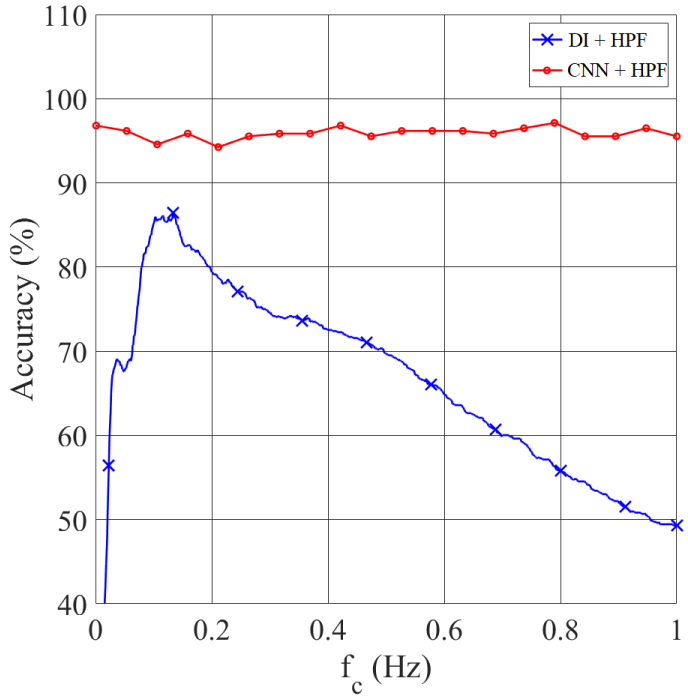


Figure 3.16: Classification accuracy versus high pass filter cut-off frequency with and without using the proposed CNN method.

3.5 Conclusion

Monitoring structural health of buildings during and after natural disasters is crucial, and directly impacts public safety. Buildings can be added to an IoT network by deploying inertial sensors in civil infrastructure, which facilitates post disaster identification of structurally unsound buildings. In this work, we illustrated how machine learning techniques can be employed to identify buildings damage state based on accelerometer readings in the presence of noise. We demonstrate how to address the effect of noise by using machine learning methods such as SVM and KNN, and that removing the noise sensitive features enhances the classification accuracy. In addition, we proposed a CNN method that is applied directly on raw data instead of the extracted features which outperformed KNN, SVM and traditional HPF noise cancellation methods.

Chapter 4

Large Signal Smart Device Accelerometer Error

Smart device sensors are designed for a consumer electronics market, not as a specialized sensing system, thus it is natural to expect that the specifications are lower than standalone sensors. In chapter 2, we presented an error model that is assumes small excitation signal. However, for large signals other errors arise which are significantly higher than additive noise as will be explained later.

Inertial sensors error is modelled in previous work [23, 24] as: 1)Scale factor error, 2) constant bias, 3) non-linearity and 4) additive random noise. While the first three error types are deterministic and can be compensated by calibration, the random noise cannot be calibrated and can only be characterized as mentioned in [30, 77]. However, using the above traditional error model for a smart device accelerometer (or other sensors as well) does not capture the OS timing uncertainty. In [25], it was shown that Android has some limitations to be considered as a real time OS, and consequently when a mobile app polls the internal accelerometer chip every constant time period, the execution is expected to have some un-

certainty that strongly depends on the coding style. Hence, the main contributions of this work can be summarized as follows:

4.1 Smart Device Accelerometer Proposed Noise Model

4.1.1 Notation

Throughout this chapter, vectors are denoted by boldface lowercase letters, e.g. $\mathbf{x}$, and matrices by boldface uppercase letters, e.g. $\mathbf{A}$. We use $(\cdot)^T$ for transpose of a vector or a matrix. Vector elements are denoted by lowercase normal font letters with subscript that indicates the element index, e.g. x_i represents the ith element in vector $\mathbf{x}$. In addition, the ith column of the matrix $\mathbf{A}$ is given by $\mathbf{A}[:, i]$. We also use $\mathbb{E}[\cdot]$ as an expectation operator. The PDF of single variable y_i and a vector $\mathbf{y}$ are denoted as $p(y_i; \mathbf{x})$ and $p(\mathbf{y}; \mathbf{x})$ respectively, where $\mathbf{x}$ is a nonrandom parameter vector. Identity matrix of K elements is denoted by I_K. Finaly, $diag(\mathbf{x})$ is a diagonal matrix whose diagonal entries correspond to the elements in the vector $\mathbf{x}$:

$$diag(\mathbf{x})[i, j] = \begin{cases} x_i & i = j \\ 0 & i \neq j \end{cases}.$$

4.1.2 Model

Most prior work modeled the non-deterministic inertial sensor error as additive noise and characterised it using either power spectral density or Allan variance techniques as mentioned in [30, 77]. Sampling time uncertainty was neglected due to the fact that using a stable clock and proper sampling, results in negligible sampling time jitter error. While this is true in a

general case, it is not true in smartphone systems, since reading acceleration data using a smart device is done by polling the sensor periodically using an OS timer. As mentioned in section 1, smart device OS such as Android has limitations to act as a real time OS. Hence, it is expected to have sampling time uncertainty while polling the sensors. Therefore, we propose an error model for smart device accelerometer that includes additive noise as well as sampling time jitter error.

Let the measured acceleration at any sampling time i be calculated by (4.1).

$$a_i = \alpha x(iT + z_i) + w_i \tag{4.1}$$

where $x(t)$, T, z_i and w_i are the true acceleration, sampling period, sampling time jitter and additive noise at time sample i respectively, and α is the scale factor that is ideally equal to one. Without loss of generality, we use normalized sampling frequency, i.e. let $T = 1$. Error due to sampling time jitter was introduced in [26] by using the first order terms of Taylor series expansion of (4.1) and ignoring higher order terms. However, we found that this approach leads to erroneous values at low values of $|\dot{x}|$, as will be illustrated later in this chapter.

Taylor expansion of (4.1) is given by (4.2), where $\frac{d^n}{dt^n}(.)$ is the nth derivative operator. Both noise sources z and w are modeled as zero mean Gaussian noise with variance σ_z^2 and σ_w^2 respectively [26]. As mentioned in [27], nth derivative is approximately evaluated by (4.3), where $sinc^{(n)}$ is the nth derivative of the cardinal sine function defined by (4.4), and K is the number of time samples. For simplicity, let $\frac{dx}{dt}$, and $\frac{d^2x}{dt^2}$ be denoted as $\dot{x}$ and $\ddot{x}$ respectively.

$$a_i = \alpha x_i + w_i + \alpha \sum_{n=1}^{\infty} \frac{1}{n!} \frac{d^n x(t)}{dt^n}|_{t=i} z_i^n \tag{4.2}$$

$$\frac{d^n x(t)}{dt^n}|_{t=i} = \sum_{k=0}^{K-1} x_k sinc^{(n)}(i-k) \tag{4.3}$$

$$sinc(t) = \frac{sin\pi t}{\pi t} \tag{4.4}$$

Based on the negligible terms of (4.2) we investigate the following two models:

1. **M1** First order: neglect $n > 1$ terms in (4.2).

2. **M2** Second order: neglect $n > 2$ terms in (4.2).

M1: In this model, higher order terms ($n > 1$) of (4.2) are neglected, hence measured acceleration is expressed by (4.5), and the vector form is given by (4.6), where $\mathbf{Z} = diag(\mathbf{z})$. The joint PDF of $\mathbf{a}$ is given by (4.7), where σ_i^2 is the error variance at sample i expressed by (4.8). The log likely-hood (LL) function is given by (4.9).

$$a_i = \alpha x_i + w_i + \alpha \dot{x}_i z_i \tag{4.5}$$

$$\mathbf{a} = \alpha \mathbf{x} + \mathbf{w} + \alpha \mathbf{Z} \dot{\mathbf{x}} \tag{4.6}$$

$$p(\mathbf{a}; \mathbf{x}, \dot{\mathbf{x}}, \sigma_w, \sigma_z)\Big|_{M1} = \prod_{i=0}^{K-1} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(a_i - \alpha x_i)^2}{2\sigma_i^2}\right) \tag{4.7}$$

$$\sigma_i^2 = \alpha^2 \dot{x}_i^2 \sigma_z^2 + \sigma_w^2 \tag{4.8}$$

$$LL(\mathbf{a};\mathbf{x},\dot{\mathbf{x}},\sigma_w,\sigma_z)\Big|_{M1} = log(p(\mathbf{a};\mathbf{x},\dot{\mathbf{x}},\sigma_w,\sigma_z)) = -\frac{1}{2}\sum_{i=0}^{K-1}\left(log(2\pi\sigma_i^2)+\frac{(a_i-\alpha x_i)^2}{\sigma_i^2}\right) \tag{4.9}$$

M2: In this model, higher order terms ($n > 2$) of (4.2) are neglected, hence measured acceleration is expressed by (4.10), and the vector form is given by (4.11). The joint PDF of $\mathbf{a}$ is given by (4.12). $p(a_i; x_i, \dot{x}_i, \ddot{x}_i, \sigma_w, \sigma_z)$ is calculated by (4.13) where $*$ is the convolution operator, $\mathcal{N}(0,\sigma_w^2)$ is a zero mean normal distribution with variance σ_w^2 and y_i is a random variable defined by (4.14). Since the PDF of z is $\mathcal{N}(0,\sigma_z^2)$ and y is a function of z then $p(y_i; \dot{x}_i, \ddot{x}_i, \sigma_z)$ is calculated numerically as described in [78]. The LL function of **M2** is given by (4.15).

$$a_i = \alpha x_i + w_i + \alpha\dot{x}_i z + \alpha\frac{\ddot{x}_i}{2}z_i^2 \tag{4.10}$$

$$\mathbf{a} = \alpha\mathbf{x} + \mathbf{w} + \alpha\mathbf{Z}\dot{\mathbf{x}} + \frac{1}{2}\alpha\mathbf{Z}\mathbf{Z}\ddot{\mathbf{x}} \tag{4.11}$$

$$p(\mathbf{a};\mathbf{x},\dot{\mathbf{x}},\sigma_w,\sigma_z) = \prod_{i=0}^{K-1} p(a_i; x_i, \dot{x}_i, \ddot{x}_i, \sigma_w, \sigma_z) \tag{4.12}$$

$$p(a_i; x_i, \dot{x}_i, \ddot{x}_i, \sigma_w, \sigma_z) = \mathcal{N}(0,\sigma_w^2) * p(y_i; \dot{x}_i, \ddot{x}_i, \sigma_z) \tag{4.13}$$

$$y_i = \alpha\dot{x}_i z_i + \alpha\frac{\ddot{x}_i}{2}z_i^2 \tag{4.14}$$

$$LL(\mathbf{a};\mathbf{x},\dot{\mathbf{x}},\sigma_w,\sigma_z)\Big|_{M2} = \sum_{i=0}^{K-1} log(p(a_i; x_i, \dot{x}_i, \ddot{x}_i, \sigma_w, \sigma_z)) \tag{4.15}$$

As mentioned in [26], MLE is applied to estimate the model parameters, i.e. σ_w and σ_z.

However, since σ_w can be estimated while the device is at rest as described in [30, 77], hence, we assume σ_w is known.

For M1 and M2 models, σ_z is estimated using (4.16). Without assuming negligible additive noise, i.e. $\sigma_w \neq 0$, equation (4.16) has no closed form solution, hence we used a numerical approach as explained in section 4.1.3.

$$\hat{\sigma}_z = \underset{\sigma_z}{\operatorname{argmax}}\, LL(\mathbf{a}; \mathbf{x}, \dot{\mathbf{x}}, \sigma_w, \sigma_z) \tag{4.16}$$

4.1.3 Simulation Results

In this section, we evaluate the estimation accuracy of the models proposed in section 4.1.2. A sinusoidal signal is used as the ground truth acceleration $x(t)$ whose normalized frequency is 0.04 and amplitude $10^4 mm/sec^2$. According to [30, 28], $\sigma_w = 10 mm/sec^2$ is typical for smartphone accelerometers. Let σ_z be 0.2, which is the parameter to be estimated.

The simulation is repeated 1000 times to calculate the estimation accuracy of the proposed models. Figure 4.1 compares between the exact root mean squared (RMS) error, M1 and M2 RMS errors. It is clear that M1 and M2 match the exact error RMS at high values of $|\dot{x}|$, however at low values, M1 underestimates the error.

Though off the shelf sensors have a calibrated scale factor α, still vendors report scale factor error even at nominal temperature. Hence, we sweep over a range of α values to demonstrate the resilience of the **M1** and **M2** to scale factor errors.

Equation (4.16) is solved numerically by sweeping on a range of values to get the $\sigma_z$value that maximizes the LL function. Figure 4.2 shows σ_z estimation for **M1** and **M2** in case a slight scale factor error is introduced. It is clear that **M2** is less sensitive to scale factor errors than **M1**. This is due to the fact that **M2** is much more accurate at the low $|\dot{x}|$ values

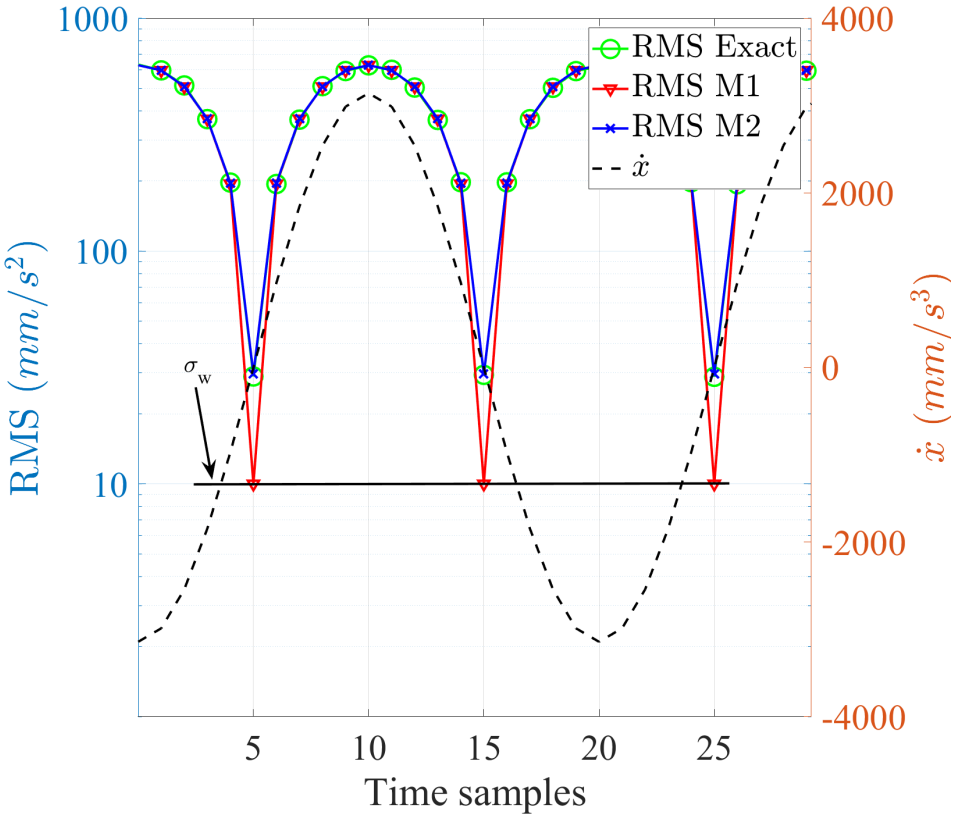


Figure 4.1: Exact, M1 and M2 RMS (left axis). True first derivative of acceleration signal $\dot{x}$ (right axis).

as mentioned earlier, and these points are the maximum values of x, i.e. the points that have the maximum error in case there is a scale factor error.

4.2 Experiments

In order to validate the proposed models, shake table experiments were performed using four different smart phones: SMG900T, SMT3350 and two different SMG920V devices. A seismic grade accelerometer PCB Piezotroincs Model393C [79] is used as a reference to measure the ground truth acceleration.

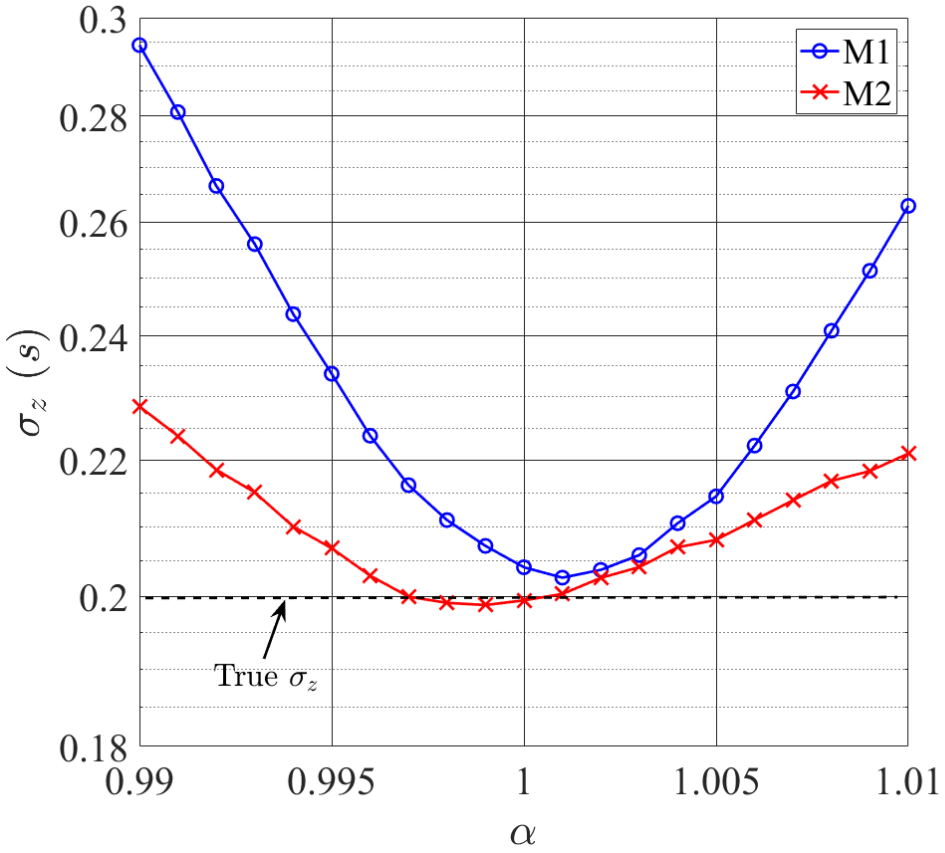


Figure 4.2: M1 and M2 estimation bias due to scale factor error.

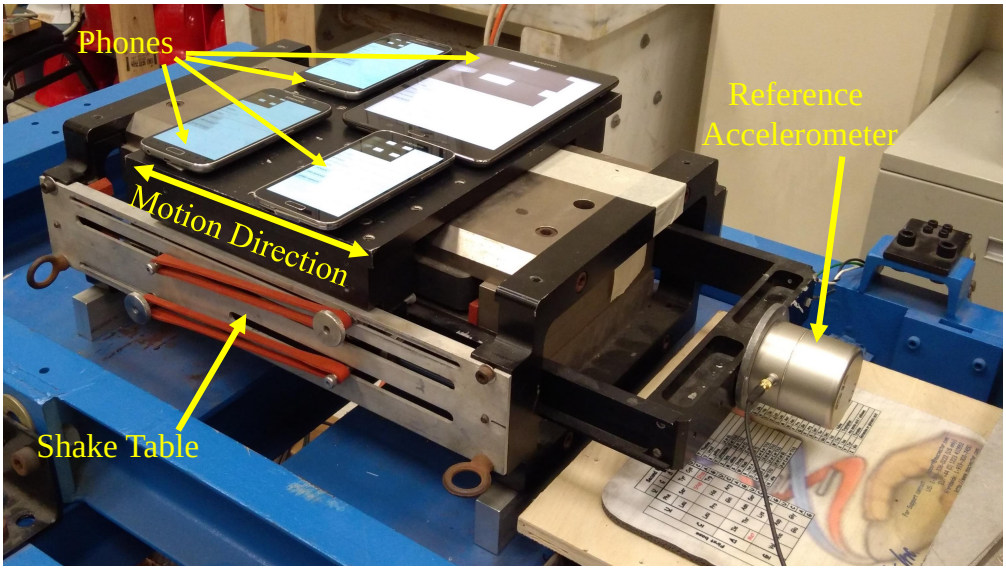


Figure 4.3: Shake table experiment setup.

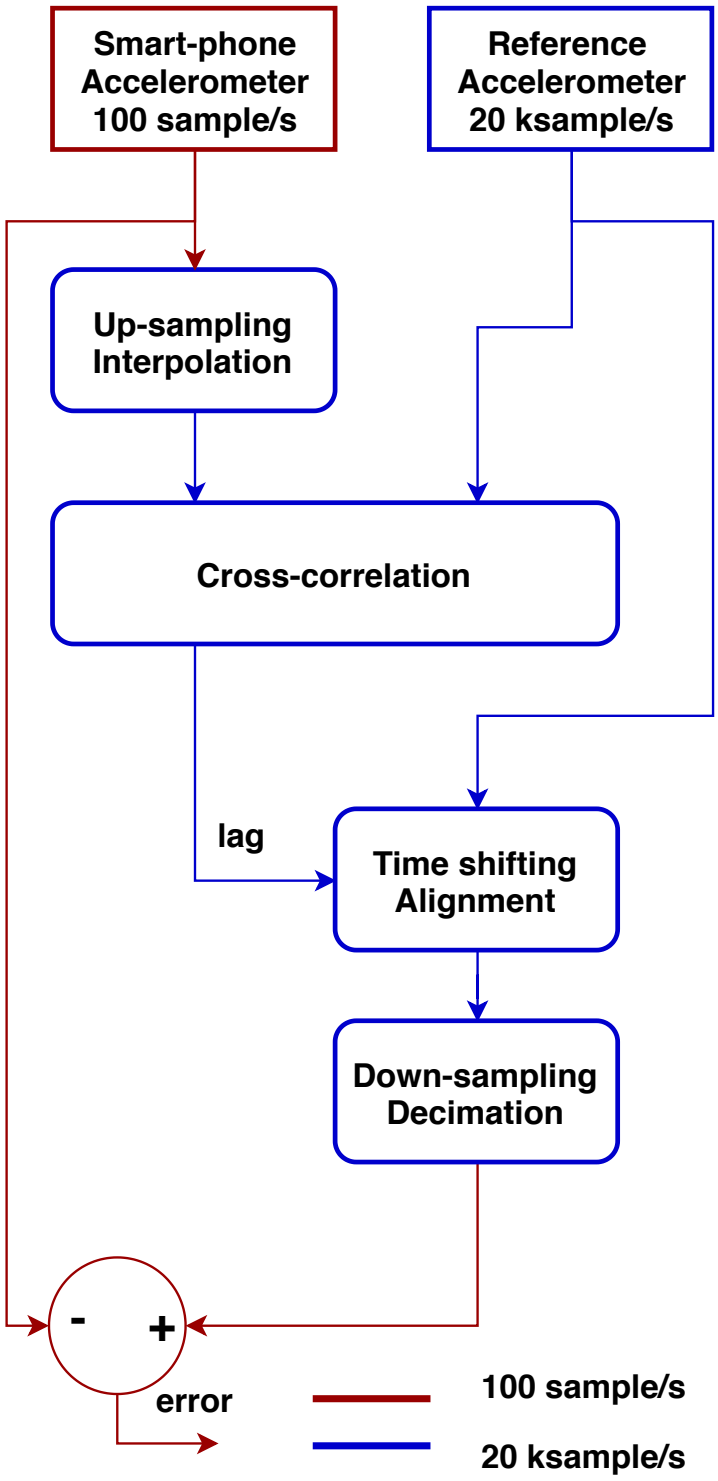


Figure 4.4: Smart phone accelerometer error calculation flow chart. Phone acceleration is up-sampled to match the reference sampling rate. Cross correlation is used to align the phone data with the reference one. Reference is down-sampled to the phone nominal rate after alignment. Error is calculated by subtraction.

Table 4.1: Summary of experimental results. Five experiments are performed to estimate the proposed model **M1** parameter σ_z for four different phones. Each experiment is a sinusoidal wave with certain frequency.

	σ_{z1} (s)			
	SMG900T	**SMG920V**	**SMG920V (2nd)**	**SMT3350**
Sine 4Hz	0.213	0.21	0.141	0.243
Sine 5Hz	0.3	0.25	0.1160	0.3
Sine 6Hz	0.239	0.187	0.12	0.272
Sine 7Hz	0.156	0.216	0.131	0.235
Sine 8Hz	0.214	0.208	0.131	0.235
Average	0.2292	0.2166	0.1248	0.2714

Table 4.2: Summary of experimental results. Five experiments are performed to estimate the proposed model **M2** parameter σ_z for four different phones. Each experiment is a sinusoidal wave with certain frequency.

	σ_{z2} (s)			
	SMG900T	**SMG920V**	**SMG920V (2nd)**	**SMT3350**
Sine 4Hz	0.187	0.185	0.131	0.208
Sine 5Hz	0.178	0.174	0.108	0.208
Sine 6Hz	0.179	0.171	0.116	0.212
Sine 7Hz	0.132	0.18	0.126	0.201
Sine 8Hz	0.163	0.167	0.109	0.2
Average	0.1689	0.1755	0.1184	0.2059

Five experiments are performed, where in each one the table is excited for 25 seconds by a sinusoidal signal of one of the following frequencies: 4, 5, 6, 7 and 8 Hz. The four smart devices are mounted to the shake table to avoid sliding. Figure 4.3 shows the experiment setup.

Figure 4.4 shows the procedure followed to calculate the error of each smartphone. A nominal sampling rate of 100 $samples/s$ is selected for the smart phones accelerometers, whereas the reference is sampling at 20 $ksample/s$. First, the smart phone data is up-sampled by cubic interpolation to match the reference rate. Second, cross-correlation is used to estimate the time lag that maximizes the cross-correlation. The reference is shifted by the estimated lag

and then down-sampled to the nominal phone sampling rate. Finally, the error is calculated by subtraction. The procedure is repeated for each of the four phones in the five experiments.

We used the proposed MLE **M1** and **M2** to estimate σ_z for each phone, and the results are summarized in tables 4.1 and 4.2 respectively. We denote **M1** and **M2** estimation results as σ_{z1} and σ_{z2} respectively. Based on the results, it is clear that that the estimated σ_z values are not much varying across the experiments for a given device, which validates the assumption that this parameter is not signal dependent.

It is worth noting that the models described in section 4.1 have two use cases:

1. Characterization: where a known vibration signal is used to estimate σ_z
2. Error estimation: where the model parameters σ_z and σ_w are known, i.e. already estimated in the characterization phase, and then the error of measuring an unknown signal is to be estimated using the models

Hence, for all the experiments, we compare the measurements to the following three techniques:

1. $M1\_\sigma_{z1}$: σ_z is characterized by **M1** and error is estimated by **M1**
2. $M2\_\sigma_{z2}$: σ_z is characterized by **M2** and error is estimated by **M2**
3. $M1\_\sigma_{z2}$: σ_z is characterized by **M2** and error is estimated by **M1**

For the four phones, figures 4.5-4.8 compare the measured error represented by the time average root mean squared (TRMS) to $M1\_\sigma_{z1}$, $M2\_\sigma_{z2}$ and $M1\_\sigma_{z2}$ models. We also added the traditional additive noise model to the figures which does not take into account sampling time jitter. It is clear that the closest models to the measurements are $M2\_\sigma_{z2}$ $M1\_\sigma_{z2}$, which indicates the effectiveness of using **M2** in the characterization phase. However, if σ_z

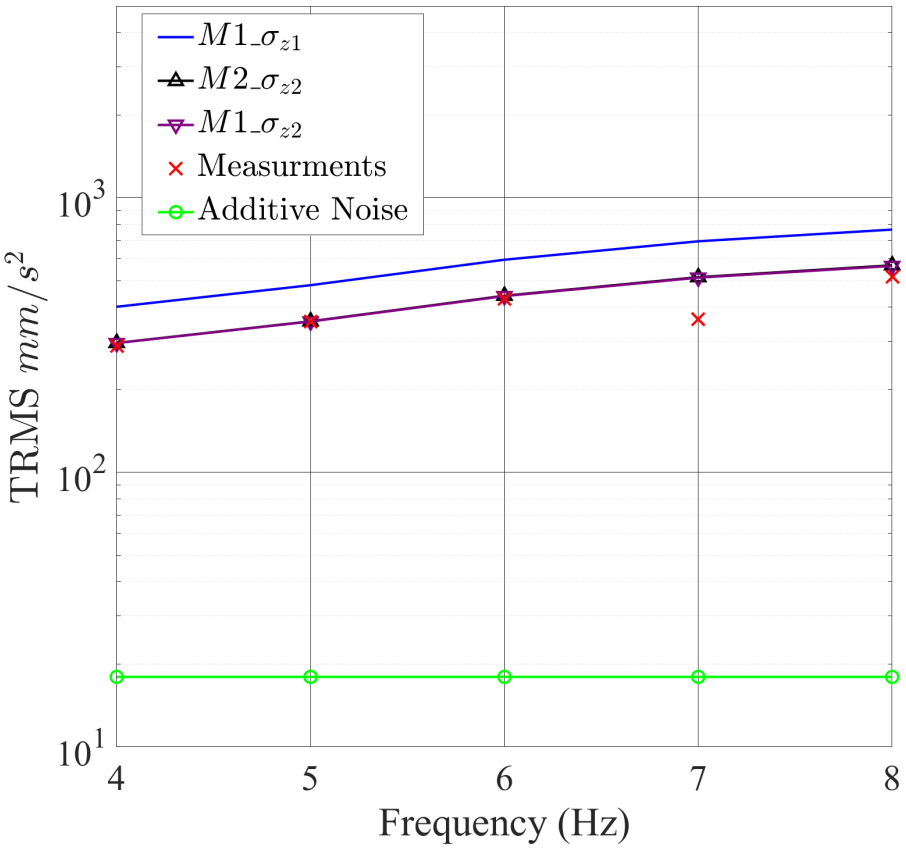


Figure 4.5: TRMS error versus the frequency of shaking for $M1\_sigma_{z1}$, $M2\_sigma_{z2}$ and $M1\_sigma_{z2}$ compared to measurments of SMG900T.

is properly estimated or given, then estimating the error using **M1** and **M2** result in almost the same values. Since, **M1** is a less complex model, we believe $M1\_\sigma_{z2}$ is the best technique.

4.3 Acceleration Estimation

In this section we derive the CRLB of acceleration estimation based on the proposed model. We assume that since **M1** and **M2** are almost equivalent in the TRMS, then we evaluate the CRLB of **M1**. In addition, since we are concerned about the best achievable accuracy, let α be equal to one. Equation (4.1) can be written as (4.17). The vector form is expressed by (4.18), where **D** is a matrix whose rows are defined by (4.19).

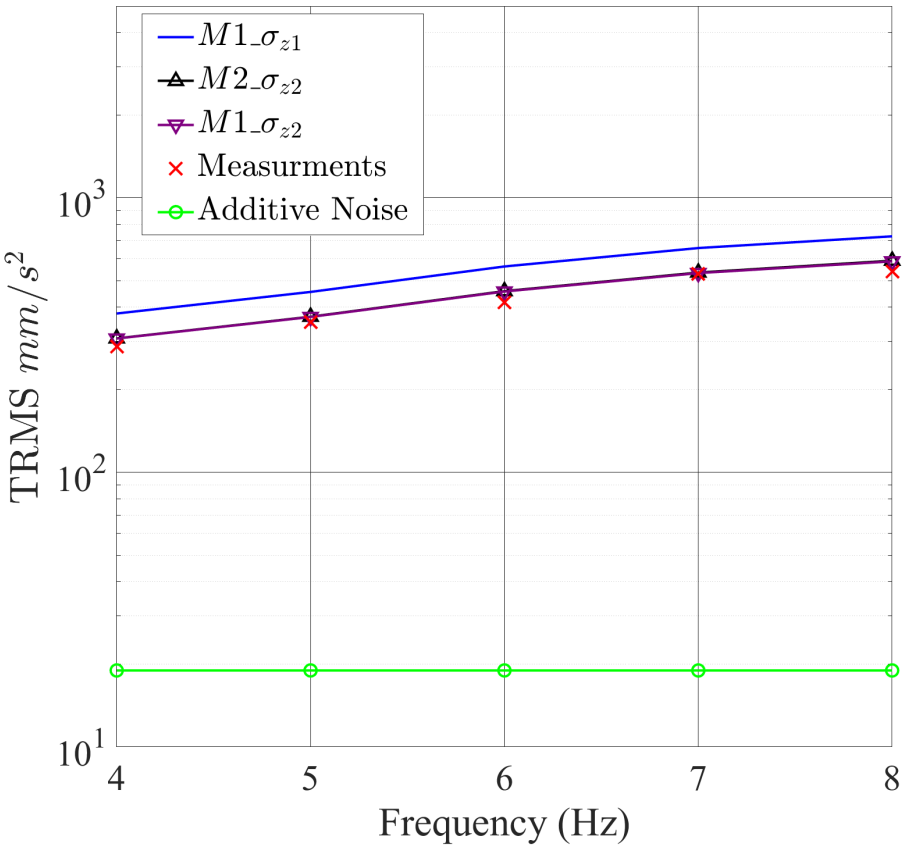


Figure 4.6: TRMS error versus the frequency of shaking for $M1\_sigma_{z1}$, $M2\_sigma_{z2}$ and $M1\_sigma_{z2}$ compared to measurments of SMG920V.

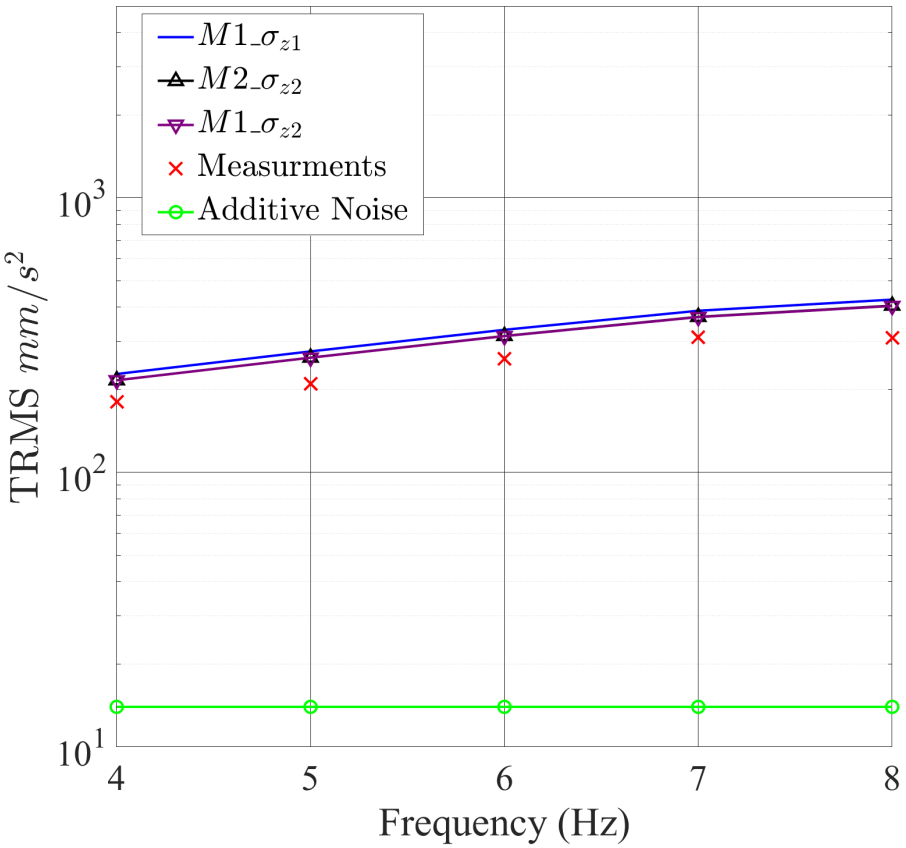


Figure 4.7: TRMS error versus the frequency of shaking for $M1\_sigma_{z1}$, $M2\_sigma_{z2}$ and $M1\_sigma_{z2}$ compared to measurments of the second SMG920V.

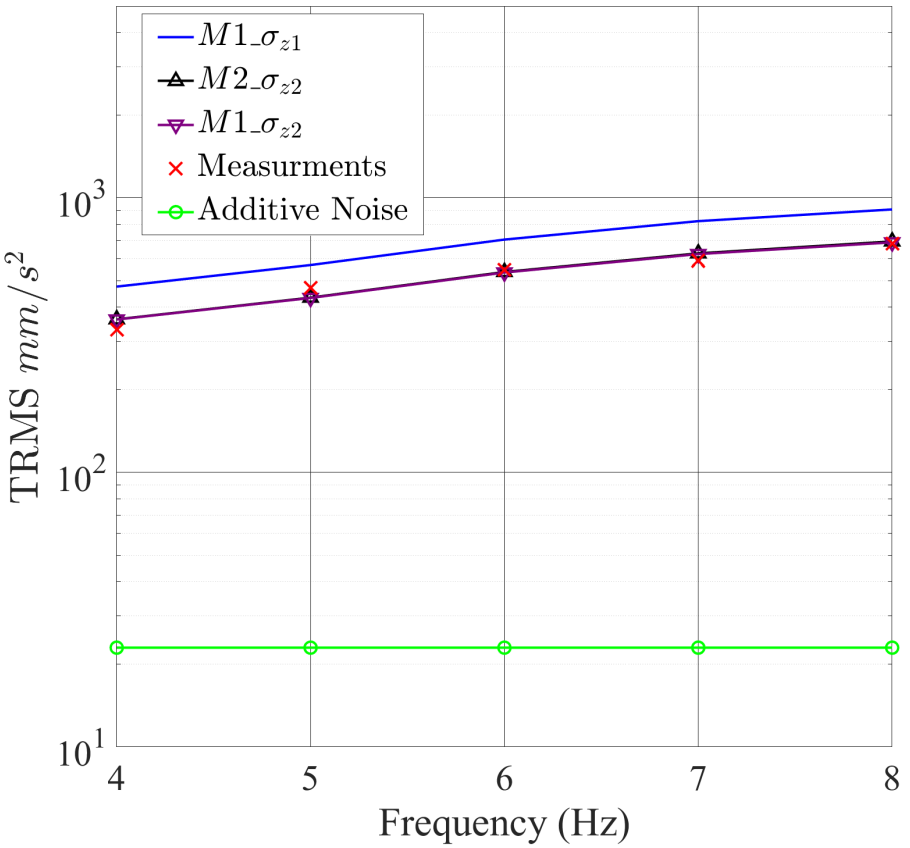


Figure 4.8: TRMS error versus the frequency of shaking for $M1\_sigma_{z1}$, $M2\_sigma_{z2}$ and $M1\_sigma_{z2}$ compared to measurments of SMT3350.

$$a_i = x_i + w_i + \sum_{k=0}^{K-1} x_k sinc^{(1)}(i-k) z_i \tag{4.17}$$

$$\mathbf{a} = \mathbf{x} + \mathbf{w} + \mathbf{ZDx} \tag{4.18}$$

Hence, as described in [27], each element of the Fisher information matrix (FIM) is evaluated by (4.20), where the covariance matrix of $\mathbf{a}$ is given by (4.21) and its derivative with respect to x_i is given by (4.22). Then, according to the CRLB theorem, the minimum mean squared error (MSE) obtained by an unbiased estimator is given by (4.23).

$$\mathbf{D}[i,:] = \begin{bmatrix} sinc^{(1)}(-i) \\ sinc^{(1)}(1-i) \\ sinc^{(1)}(2-i) \\ \vdots \\ sinc^{(1)}((K-1)-i) \end{bmatrix}^T \tag{4.19}$$

$$\mathbf{FIM}[i,j] = \frac{1}{2} tr\{ \frac{\partial \mathbf{\Sigma_a}}{\partial x_i} \mathbf{\Sigma_a}^{-1} \frac{\partial \mathbf{\Sigma_a}}{\partial x_j} \mathbf{\Sigma_a}^{-1} \} + \mathbf{\Sigma_a}^{-1} \tag{4.20}$$

$$\mathbf{\Sigma_a} = \sigma_z^2 (\mathbf{D x x}^T \mathbf{D}^T) + \sigma_w^2 I_K \tag{4.21}$$

$$\frac{\partial \mathbf{\Sigma_a}}{\partial x_i} = 2\sigma_z^2 diag\{ \mathbf{D}[:,i] \odot (\mathbf{Dx}) \} \tag{4.22}$$

$$MSE\Big|_{min} = \mathbb{E}[(\hat{\mathbf{x}} - \mathbf{x})^T (\hat{\mathbf{x}} - \mathbf{x})] = tr\{\mathbf{FIM}^{-1}\} \tag{4.23}$$

The CRLB is evaluated numerically assuming ground truth acceleration is a sinusoidal signal of amplitude $10^4 mm/s^2$ and normalized frequency of 0.04 Hz, $\sigma_w = 10 mm/sec^2$ and different σ_z values. Figure 4.9 compares the CRLB to the linear estimation accuracy. It is worth noting that a linear estimator as mentioned in [27], corresponds to $\hat{\mathbf{x}} = \mathbf{a}$, i.e. $MSE|_{Linear} = tr\{\mathbf{\Sigma_a}\}$

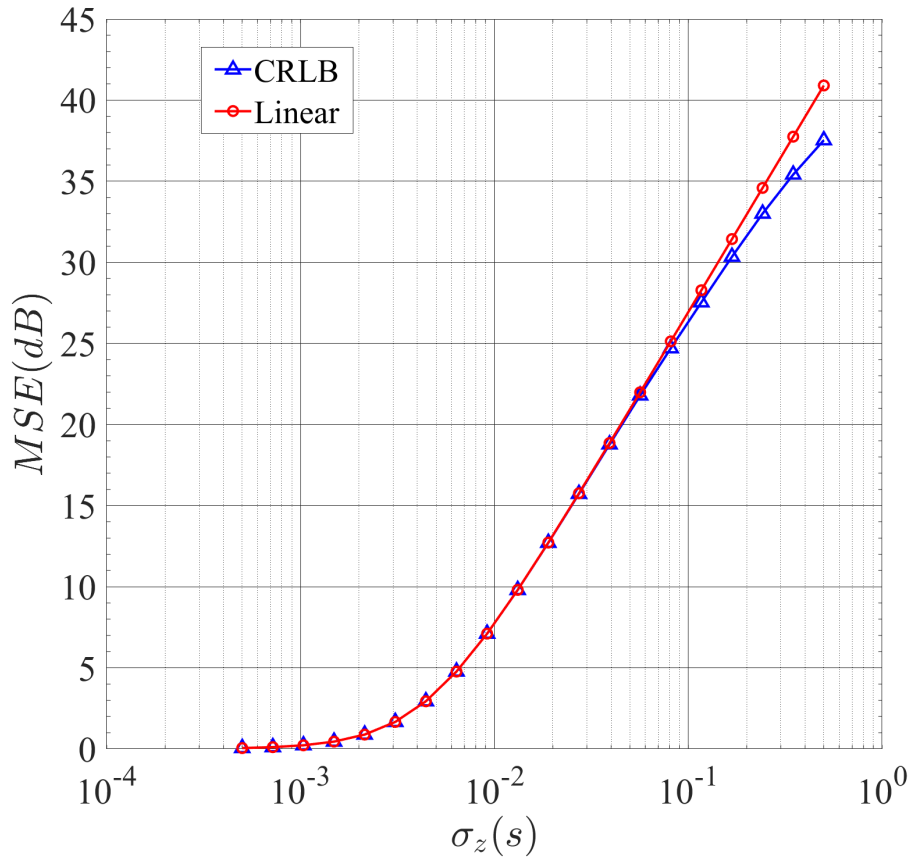


Figure 4.9: MSE of linear estimation of acceleration is plotted versus σ_z and compared to the CRLB,knowing that the true acceleration signal is a sinusoidal signal of $10^4 mm/s^2$ amplitude and 0.04Hz normalized frequency. The zero-dB level corresponds to the additive noise.

in case no over sampling is assumed.

As mentioned in section 4.2, σ_z is approximately about 0.2, hence, it is clear from figure 4.9 that linear estimator accuracy is very close to the minimum theoretical bound. However, for higher jitter, other estimation techniques such as maximum likelihood estimation [27] can be adopted.

4.4 Conclusion

Recently, smart devices have become a necessity in today's world. Since they are equipped with multiple sensors that measure most of the surrounding physical quantities. Lots of apps emerged that target biomedical, environment and infrastructure applications. In this work, we have investigated the accuracy limits of a smart device accelerometer which is considered one of the most used sensors in such applications. We proposed a novel error model that takes into account software sampling time error in addition to the additive noise model developed in previous work. The proposed model has been validated experimentally on four smart devices by several shake table experiments. It was shown that the proposed model estimates error TRMS that is much closer to the real measured error compared to the additive noise model which is about an order of magnitude less. In addition, with respect to acceleration estimation accuracy based on the proposed model, it was shown that the linear estimators are close to the CRLB at the current sampling time jitter value. Hence, using a less computationally expensive linear estimator will not affect much the achieved accuracy.

Chapter 5

Conclusion and Future Work

Recently, smart devices are equipped with multiple sensors that measure most of the surrounding physical quantities. For instance, a smart phone has a tri-axial accelerometer, a tri-axial gyroscope and a tri-axial magnetometer, which measure linear acceleration, rotational velocity and magnetic field respectively. In addition to a pressure, humidity, light and proximity sensors. Moreover, recent devices are equipped with sensors that measure human vitals such as electrocardiogram (ECG). Lots of apps emerged that target biomedical, environment and infrastructure applications.

In this work, we have investigated the accuracy limits of a smart device accelerometer which is considered one of the most used sensors in several applications. We have selected SHM as a use case of smart device accelerometer since it is crucial, and directly impacts public safety. Buildings can be continuously monitored by using the already available smart devices within these buildings. For noise cancellation, we have derived a ZUPT which benefit of the fact that an earthquake vibration eventually stops. Shake table experiments have been performed to validate the algorithm.

We have also illustrated how machine learning techniques can be employed to identify build-

ings damage state based on accelerometer readings in the presence of noise. We have illustrated how to address the effect of noise by adopting machine learning classifiers such as SVM and KNN. We have also shown how classification accuracy increases by removing the noise sensitive features. In addition, a CNN method as been proposed where raw data is used directly as an input instead of the extracted hand-crafted features. The proposed method has outperformed KNN, SVM and traditional HPF noise cancellation methods.

For large signal excitation, we have proposed a novel error model that considers software timing issues in addition to the additive noise model. Four smart devices have been experimented using a shake table to validate the model. Using several shaking frequencies, it was shown that the proposed model estimates error TRMS that is much closer to the real measured error, specially compared to the additive noise model which underestimates the error significantly. In spite of the complexity of the proposed model, we have shown that a linear estimator results in an accuracy that is very close to the CRLB at the given jitter values.

5.1 Future Work

Recently, using smart devices such as smart phones has become a daily habit for the majority of the population. People depend on smart phones in their daily activities such as recording notes, checking emails, and even playing games. In addition, people have moved from using a standalone GPS device into using navigation *apps*, which are more convenient and continuously updated with traffic information. For user security and privacy purposes, apps are not by default having access to location, and the user is required to grant that access for each app [80]. However, since there are sensors inside smart devices that require no permissions to access such as accelerometer, gyroscope, pressure and magnetometer, several attempts have been made to gather unauthorized information from smart phones by accessing these

sensors. For instance, in [81] it was shown that the MEMS gyroscopes found on modern smart phones are sufficiently sensitive to measure acoustic signals surrounding the phone. Though the captured signals contain only very low-frequency information, however using signal processing and machine learning, that was sufficient to identify speaker information and parse speech. Even if the user has never granted the access to an app to access the microphone, however using gyroscope, it is possible to identify speaker gender and part of his speech.

On the other hand, in [82], a 'zero-permission' approach that only used motion sensors on the device to infer the text typed was presented. It was shown, on average, that nearly 30 percent of typed bigrams from unseen words can be identified, using a very small volume of training data. Moreover, in [83], key press inference attacks on handheld touchpads by using smartwatch motion sensors as a side-channel were investigated. It was validated experimentally that inference using smartwatches is comparable with similar attacks using smartphone motion sensors. In addition, combining of both smartwatch and smartphone motion sensors results in even better inference accuracy. According to [84], location inference using acoustic side-channels has been studied. The attack uses acoustic information in a close environment which is used to generate a location fingerprint. In other words, the attack depends on the pattern of acoustic reflections of human voice at that specific location.

In this preliminary study, we use smart device pressure sensor as a side-channel to infer the device location with 'zero-permission'. Atmospheric pressure value changes according to the altitude at any given location. Hence, since several locations have the same altitude, naturally they will have very close atmospheric pressure values. As a result, mathematically though pressure can be defined as a function of location, the opposite is not true. However, let's focus on a regular route between two locations at separated by a few-minute driving distance. Due to the irregular structure of the terrain, the altitude in several cities is not expected to be constant, however each road has its own altitude profile. For instance, that

phenomenon encouraged researchers to create bicycling friendly maps for navigation that disfavor steep slopes in San Fransisco [85]. What we propose is to use the altitude profile of each road as a signature that identifies that road. At any given time according to the traffic condition, most of the cars passing through certain road will have almost the same average speed, which implies almost the same altitude patterns. Therefore, if passengers of some cars have granted an access to certain app to get location, it can also track the altitude change using the pressure sensor. Then, an app in another car that has only access to pressure can use pattern matching to infer the location.

As mentioned earlier, most popular smart devices operating systems such as Android and iOS do not require permissions to access pressure sensor. Even if that changed in the future, if specifically the user is asked to grant a specific application the access to pressure sensor data, the link between pressure and location inference is not clear and might seem impossible. Hence, in this work we focus on illustrating how pressure sensor can be used as a side channel to infer the device outdoor location.

To validate our concept, we use data collected by a smart device mounted in one of University of California, Irvine (UCI) transportation busses. The smart phone app records pressure data along with the location. The data is used to train a long short-term memory (LSTM) [86] neural network model to be used for pattern matching. LSTM has been used recently in several applications that deal with time-series data [87, 88, 89, 90]. In addition, it was shown that LSTM outperforms traditional approaches based on sequence mining or hidden Markov modeling [91].

5.1.1 Dataset Creation

In order to validate the proposed concept, a dataset of location sequences accompanied by the corresponding pressure sequences is required. We coordinated with UCI transportation

(UCI Anteater Express) to mount smart phone devices in one of their busses to collect data. The reason behind selecting these busses is the fact that a transportation bus follows fixed defined route repeatedly. For instance, the bus that we selected repeats its lap about 28 times a day, which creates several measurements for the same location sequence. Since the dataset is recorded throughout a day, it takes into account different traffic conditions. In other words, the pressure pattern of certain road stretches if the vehicle is moving slowly due to traffic and vice versa. This problem has been addressed in speech recognition field.

We developed an Android app that continuously records readings of pressure sensor along with the location updates. Each measurement is stored with its time stamp so pressure data can be aligned with the corresponding location later. The time stamp function provided by Android is `currentTimeMillis`, which returns time in milliseconds since midnight of January 1, 1970 [92].

Figure 5.1 shows the recorded pressure, latitude and longitude of several bus laps captured by the device GPS sensor. The data is normalized by removing the bias and adjusting the scale as shown by (5.1), where $X_{normalized}$ is the normalized version of an arbitrary vector X. It is clear from the figure that each segment of latitude and longitude has a corresponding pressure pattern.

Pressure is recorded at 5Hz rate, and location is updated at 1Hz rate and both are resampled at 0.1Hz. As mentioned in [93], time series data is converted to dataset samples by windowing. Hence, the recorded data is divided into 10-minute 75% overlapping windows, where each window consists of 60 time samples. A single window is considered one sample of the dataset with pressure sequence as an input and location sequence as an output. Figure 5.2 summarizes the dataset generation process. Then, the dataset is shuffled and divided into three sections, 65% for training, 25% for validation and 10% for testing.

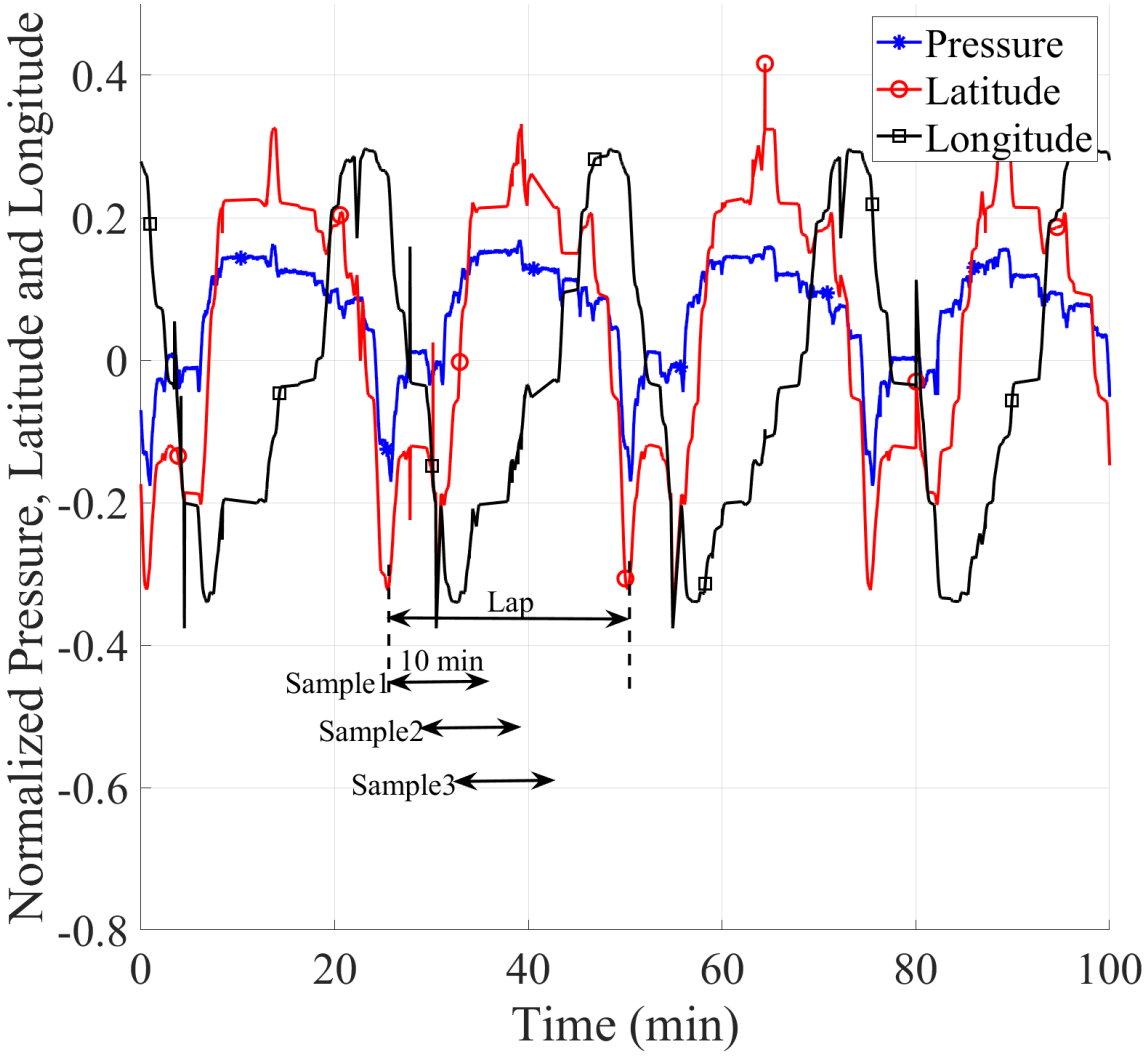


Figure 5.1: Normalized pressure, latitude and longitude measurements of several bus laps.

$$X_{normalized} = \frac{X - mean(X)}{|max(X) - min(X)|} \tag{5.1}$$

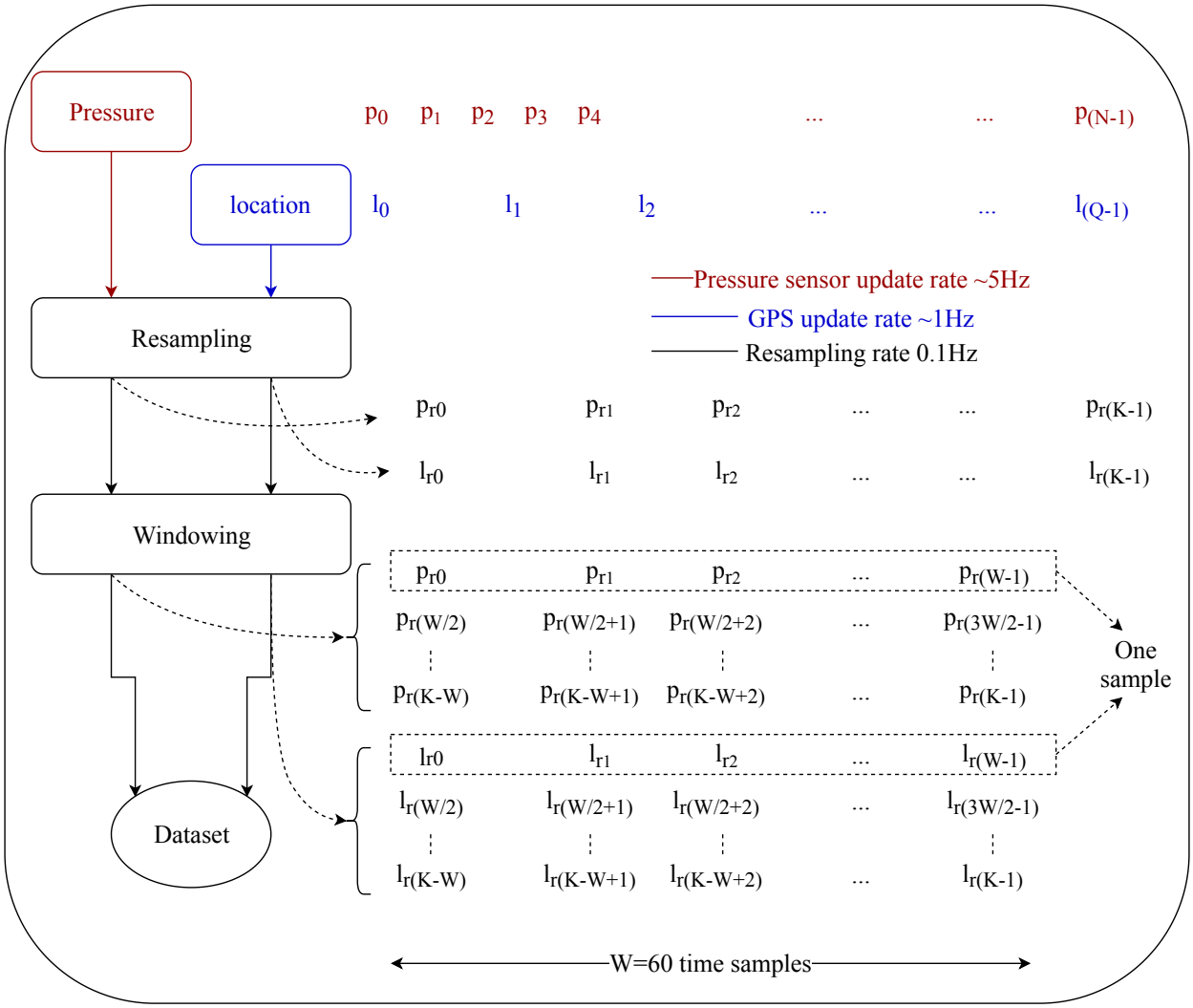


Figure 5.2: Dataset creation flow chart. Overlapping in the figure is 50% just for illustration.

5.1.2 LSTM Architecture

As mentioned earlier, LSTM is one of the recently used techniques for pattern matching which was shown to achieve better performance than traditional methods. The architecture of the LSTM network is described in table 5.1 and was implemented using Keras Python library. The input layer consists of 60 units since the input data is a 60-sample pressure window. The first two layers are LSTM layers of size of 16 units, and each layer outputs the entire sequence, hence the output size is 60x16. LSTM layer connects keeps the time dependency of the input data and infer temporal relations between time steps. The next layer is a dropout one where its rate is rate equal to 0.4, i.e. randomly 40% of the outputs of the previous layer are dropped while training, which is mainly used to avoid over fitting and forces the network to depend on several features. Then another stage of LSTM followed by a dropout layer are used. Finally, the output layer is an LSTM of size 2 and also returning the full sequence, i.e. the output size is 60x2. Recall that the output is the location sequence, which consists of a 60-sample window of latitude and longitude, i.e. the output size is 60x2.

Table 5.1: LSTM network architecture.

Layer	**Type**	**Parameters**	**Activation**
0	input	size = 60	-
1	LSTM	units=16 return_sequences=True	relu
2	LSTM	units=16 return_sequences=True	relu
3	Dropout	rate=0.4	-
4	LSTM	units=16 return_sequences=True	relu
5	Dropout	rate=0.4	-
6	LSTM	units=2 return_sequences=True	linear

5.1.3 Results and Discussion

The network described in section 5.1.2 is trained by the training section of the dataset. Figure 5.3 shows the mean absolute error for both training and validation datasets versus epochs. To demonstrate the efficacy of the proposed method, few data points of the test dataset are shown in figure 5.4. The ground truth path is shown by the dashed line, whereas GPS and pressure inferred locations are annotated. It is clear that using zero permissions and depending on pressure traces only the device location can be inferred with fair accuracy. Also, it is worth noting that the GPS location itself was noisy since the used devices did not have access to the network and only used GPS for localization, which resulted in lower accuracy and sometimes loss of GPS signal which in turn introduced sudden jumps in the location.

It is clear that location inference with zero-permission depending on pressure sensor as a side channel is possible. LSTM is shown to be an efficient sequence matching technique to infer location sequence from pressure pattern. This preliminary result opens several research areas related to augmenting several smart device sensors to infer information or to enhance measurements accuracy.

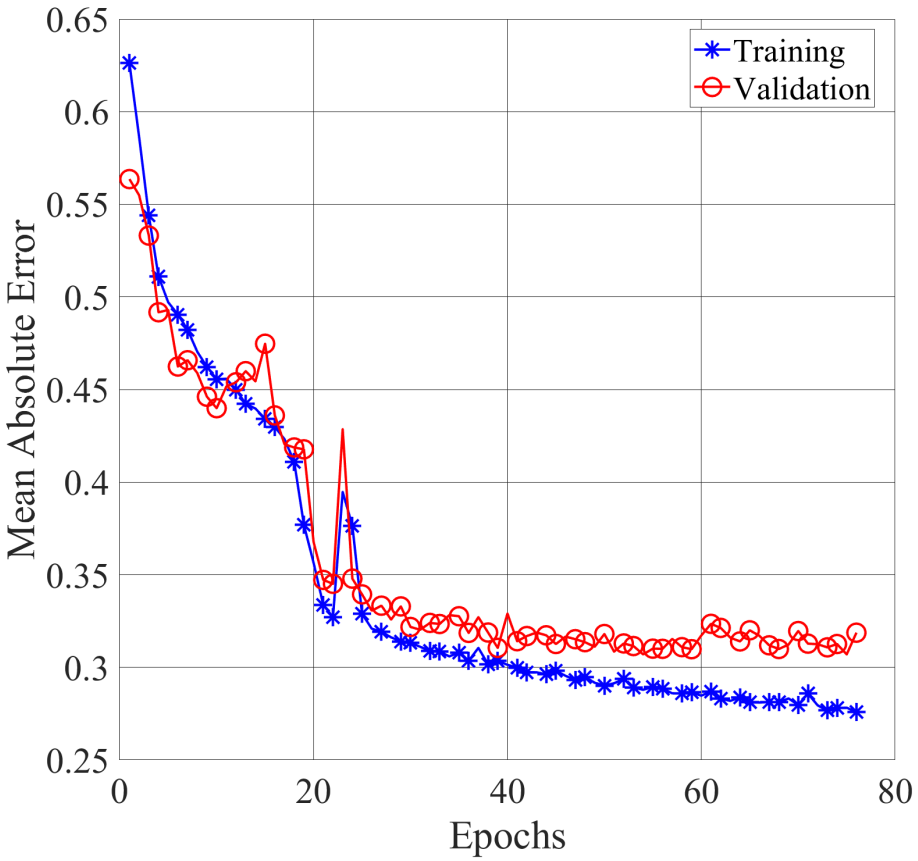


Figure 5.3: Mean absolute error for training and validation datasets versus epochs.

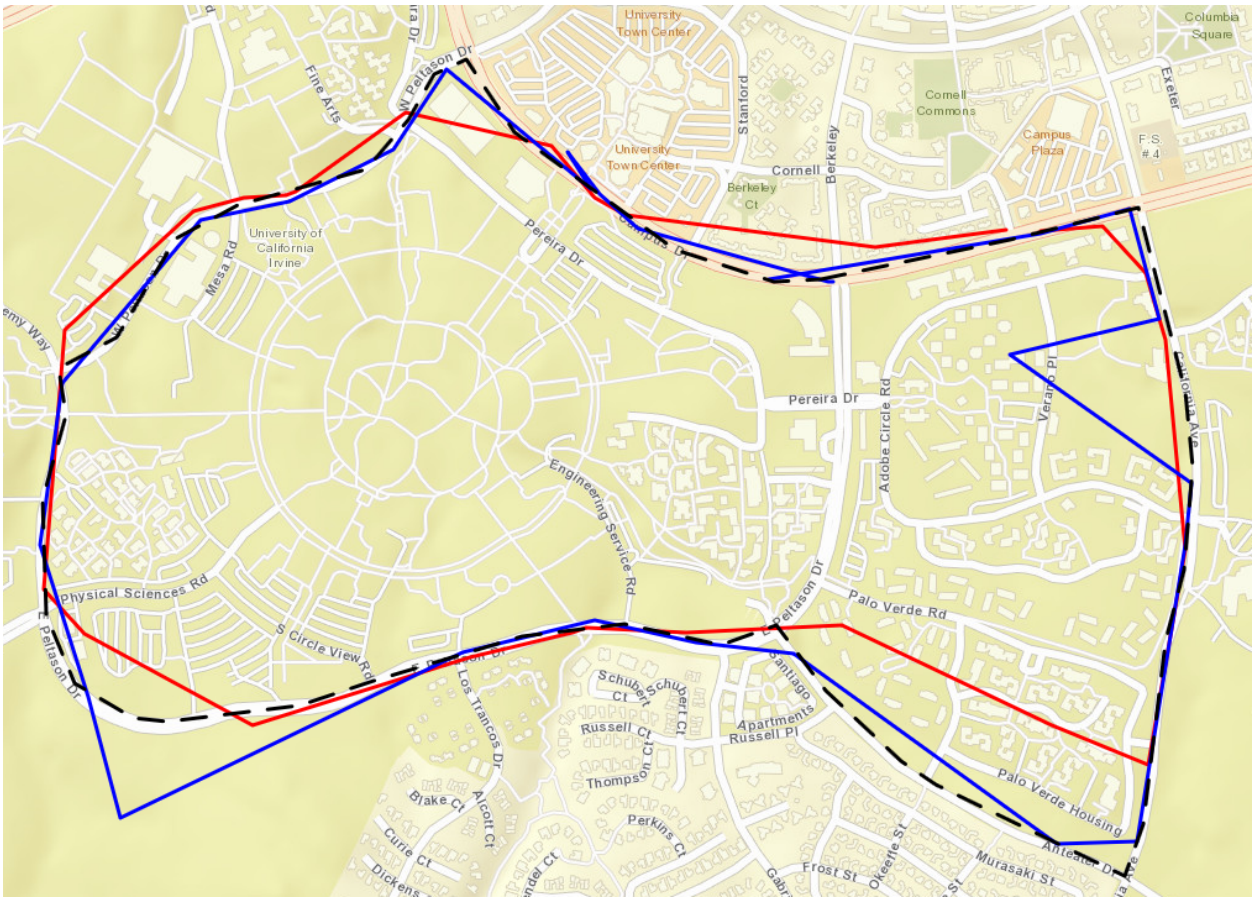


Figure 5.4: The ground truth bus route (dashed line) is plotted on UCI map. In addition, GPS (blue line) and pressure inferred locations (red line) are annotated.

Bibliography

[1] Erik H Vanmarcke and Shih-Sheng P Lai. Strong-motion duration and rms amplitude of earthquake records. *Bulletin of the seismological Society of America*, 70(4):1293–1307, 1980.

[2] Building Seismic Safety Council. Nehrp guidelines for the seismic rehabilitation of buildings. *FEMA-273, Federal Emergency Management Agency, Washington, DC*, pages 2–12, 1997.

[3] American Society of Civil Engineers. *Seismic rehabilitation of existing buildings*, volume 41, pages 12–13. ASCE Publications, 2007.

[4] N. Kostikis, D. Hristu-Varsakelis, M. Arnaoutoglou, and C. Kotsavasiloglou. A smartphone-based tool for assessing parkinsonian hand tremor. *IEEE Journal of Biomedical and Health Informatics*, 19(6):1835–1842, Nov 2015.

[5] O. Lahdenoja, T. Hurnanen, Z. Iftikhar, S. Nieminen, T. Knuutila, A. Saraste, T. Kiviniemi, T. Vasankari, J. Airaksinen, M. Pnkl, and T. Koivisto. Atrial fibrillation detection via accelerometer and gyroscope of a smartphone. *IEEE Journal of Biomedical and Health Informatics*, 22(1):108–118, Jan 2018.

[6] P. Li, Y. Wang, Y. Tian, T. Zhou, and J. Li. An automatic user-adapted physical activity classification method using smartphones. *IEEE Transactions on Biomedical Engineering*, 64(3):706–714, March 2017.

[7] B. And, S. Baglio, C. O. Lombardo, and V. Marletta. A multisensor data-fusion approach for adl and fall classification. *IEEE Transactions on Instrumentation and Measurement*, 65(9):1960–1967, Sep. 2016.

[8] C. Shen, B. Ho, and M. Srivastava. Milift: Efficient smartwatch-based workout tracking using automatic segmentation. *IEEE Transactions on Mobile Computing*, 17(7):1609–1622, July 2018.

[9] M. Gil-Martn, R. San-Segundo, S. L. Lutfi, and A. Coucheiro-Limeres. Estimating gravity component from accelerometers. *IEEE Instrumentation Measurement Magazine*, 22(1):48–53, Feb 2019.

[10] F. Kalim, J. P. Jeong, and M. U. Ilyas. Crater: A crowd sensing application to estimate road conditions. *IEEE Access*, 4:8317–8326, 2016.

[11] Matthew Faulkner, Michael Olson, Rishi Chandy, Jonathan Krause, K Mani Chandy, and Andreas Krause. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 13–24. IEEE, 2011.

[12] Matthew Faulkner, Robert W Clayton, Thomas Heaton, K Mani Chandy, Monica D Kohler, Julian Bunn, Richard Guy, Annie Liu, Michael Olson, Ming-Hei Cheng, et al. Community sense and response systems: Your phone as quake detector. *Communications of the ACM*, 57(7):66–75, 2014.

[13] Jack Reilly, Shideh Dashti, Mari Ervasti, Jonathan D Bray, Steven D Glaser, and Alexandre M Bayen. Mobile phones as seismologic sensors: Automating data extraction for the ishake system. *IEEE Transactions on Automation Science and Engineering*, 10(2):242–251, 2013.

[14] Qingkai Kong, Richard M Allen, Louis Schreier, and Young-Woo Kwon. Myshake: A smartphone seismic network for earthquake early warning and beyond. *Science advances*, 2(2):e1501055, 2016.

[15] Ahmed Ali M Alzughaibi. Post-disaster structural health assessment system using personal mobile-phones. Master's thesis, UC Irvine, 2018.

[16] A. A. Alzughaibi, A. M. Ibrahim, A. M. Eltawil, Y. Na, and S. El-Tawil. Post-disaster structural health monitoring system using personal mobile-phones. In *2019 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, pages 1–4, Jan 2019.

[17] Venture, SAC Joint and Guidelines Development Committee and others. *Recommended seismic design criteria for new steel moment-frame buildings*. Federal Emergency Management Agency, 2000.

[18] American Society of Civil Engineers and United States. Federal Emergency Management Agency. *Global topics report on the prestandard and commentary for the seismic rehabilitation of buildings*. The Agency, 2000.

[19] María P González and José L Zapico. Seismic damage identification in buildings using neural networks and modal data. *Computers & structures*, 86(3-5):416–426, 2008.

[20] Magdalena Rucka and Krzysztof Wilde. Neuro-wavelet damage detection technique in beam, plate and shell structures with experimental validation. *Journal of Theoretical and Applied Mechanics*, 48:579–604, 2010.

[21] LD Goh, N Bakhary, AA Rahman, and BH Ahmad. Prediction of unmeasured mode shape using artificial neural network for damage detection. *Jurnal Teknologi*, 61(1), 2012.

[22] Yunsu Na, Sherif El-Tawil, Ahmed Ibrahim, and Ahmed Eltawil. Dynamic behavior of a smart device on a surface subjected to earthquake motion. *Earthquake Engineering Structural Dynamics*, 2018.

[23] Wei Tech Ang, Si Yi Khoo, Pradeep K Khosla, and Cameron N Riviere. Physical model of a mems accelerometer for low-g motion tracking applications. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1345–1351. IEEE, 2004.

[24] Gong-Xu Liu, Ling-Feng Shi, Jian-Hui Xun, Sen Chen, Lei Zhao, and Yi-Fan Shi. An orientation estimation algorithm based on multi-source information fusion. *Measurement Science and Technology*, 29(11):115101, 2018.

[25] Cláudio Maia, Luis Miguel Nogueira, and Luis Miguel Pinho. Evaluating android os for embedded real-time systems. In *6th international workshop on operating systems platforms for embedded real-time applications*, pages 63–70, 2010.

[26] Gerd Vandersteen and Rik Pintelon. Maximum likelihood estimator for jitter noise models [hf sampling scopes]. *IEEE Transactions on Instrumentation and Measurement*, 49(6):1282–1284, 2000.

[27] H. Araghi, M. A. Akhaee, and A. Amini. Joint compensation of jitter noise and time-shift errors in multichannel sampling system. *IEEE Transactions on Instrumentation and Measurement*, pages 1–10, 2018.

[28] MPU6500 datasheet. Accessed on: February 12th, 2018.

[29] Qingkai Kong, Richard M Allen, Monica D Kohler, Thomas H Heaton, and Julian Bunn. Structural health monitoring of buildings using smartphone sensors. *Seismological Research Letters*, 89(2A):594–602, 2018.

[30] A. Ibrahim, A. Eltawil, Y. Na, and S. El-Tawil. Effect of sensor error on the assessment of seismic building damage. *IEEE Transactions on Instrumentation and Measurement*, pages 1–12, 2019.

[31] Sudha Radhika, Yukio Tamura, and Masahiro Matsui. Application of remote sensing images for natural disaster mitigation using wavelet based pattern recognition analysis. In *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International*, pages 84–87. IEEE, 2016.

[32] T. Harms, S. Sedigh, and F. Bastianini. Structural health monitoring of bridges using wireless sensor networks. *IEEE Instrumentation Measurement Magazine*, 13(6):14–18, December 2010.

[33] C. Gentile, M. Guidobaldi, and A. Saisi. Structural health monitoring of a historic masonry tower. In *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings*, pages 168–173, July 2015.

[34] R. Yin, Y. Wu, and T. Hsu. Application of the low-cost mems-type seismometer for structural health monitoring: A pre-study. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–5, May 2016.

[35] D. Jayawardana, S. Kharkovsky, R. Liyanapathirana, and X. Zhu. Measurement system with accelerometer integrated rfid tag for infrastructure health monitoring. *IEEE Transactions on Instrumentation and Measurement*, 65(5):1163–1171, May 2016.

[36] A. Girolami, D. Brunelli, and L. Benini. Low-cost and distributed health monitoring system for critical buildings. In *2017 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, pages 1–6, July 2017.

[37] S. Valenti, M. Conti, P. Pierleoni, L. Zappelli, A. Belli, F. Gara, S. Carbonari, and M. Regni. A low cost wireless sensor node for building monitoring. In *2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, pages 1–6, June 2018.

[38] Oliver J Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14:15, 2007.

[39] Clotaire Michel, Krisztina Kelevitz, Nicolas Houlié, Benjamin Edwards, Panagiotis Psimoulis, Zhenzhong Su, John Clinton, and Domenico Giardini. The potential of high-rate gps for strong ground motion assessment. *Bulletin of the Seismological Society of America*, 107(4):1849–1859, 2017.

[40] Official U.S. government information about the Global Positioning System. GPS Accuracy. Accessed on: May 11th, 2018.

[41] Dongming Feng and Maria Q Feng. Identification of structural stiffness and excitation forces in time domain using noncontact vision-based displacement measurement. *Journal of Sound and Vibration*, 406:15–28, 2017.

[42] Aboelmagd Noureldin, Tashfeen B Karamat, and Jacques Georgy. *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Science & Business Media, 2012.

[43] Zhelong Wang, Hongyu Zhao, Sen Qiu, and Qin Gao. Stance-phase detection for zupt-aided foot-mounted pedestrian navigation system. *IEEE/ASME Transactions on Mechatronics*, 20(6):3170–3181, 2015.

[44] Khairi Abdulrahim, Terry Moore, Chris Hide, and Chris Hill. Understanding the performance of zero velocity updates in mems-based pedestrian navigation. *International Journal of Advancements in Technology*, 5(1):53–60, 2014.

[45] Jingchang Pan, Xicheng Qian, and Bo Zhang. Design and implementation of application based on smartphone sensors. In *Proceedings of the 2017 International Conference on Telecommunications and Communication Engineering*, pages 97–102. ACM, 2017.

[46] YK Thong, MS Woolfson, JA Crowe, BR Hayes-Gill, and DA Jones. Numerical double integration of acceleration measurements in noise. *Measurement*, 36(1):73–92, 2004.

[47] N. J. Kasdin. Discrete simulation of colored noise and stochastic processes and 1/f alpha; power law noise generation. *Proceedings of the IEEE*, 83(5):802–827, May 1995.

[48] Haiying Hou. *Modeling inertial sensors errors using Allan variance*. Library and Archives Canada= Bibliothèque et Archives Canada, 2005.

[49] MTI100 datasheet. Accessed on: February 12th, 2018.

[50] AXO215 datasheet. Accessed on: February 12th, 2018.

[51] Mistras model 1030 datasheet. Accessed on: February 12th, 2018.

[52] KB12VD datasheet. Accessed on: February 12th, 2018.

[53] Venture, NEHRP Consultants Joint. Evaluation of the fema p-695 methodology for quantification of building seismic performance factors. *US Department of Commerce, Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD*, pages 20899–8600, 2010.

[54] Dominic Kelly. Seismic site classification for structural engineers. *Structure*, 21, 2006.

[55] Cem Topkaya and Ahmet Kuşyılmaz. Seismic performance factors for steel eccentrically braced frames. In *8th Hellenic National Conference on Steel Structures*, pages 2–4, 2014.

[56] Altair Engineering Inc. Hypermesh.

[57] Livermore Software Technology Corp. Ls-dyna.

[58] T. Y. Wu, S. El-Tawil, and J. McCormick. Behavior of steel moment frames with deep column sections under seismic loading. In *16th World Conference on Earthquake,16WCEE, Santiago Chile, January 9th to 13th*, 2017.

[59] P FEMA. 695. quantification of building seismic performance factors. *Federal Emergency Management Agency*, 2009.

[60] Laura M Rodríguez Peralta and Eduardo Ismael-Hernández. A proposal to estimate seismic risk on buildings using wsn. In *Science and Information Conference (SAI), 2015*, pages 1170–1177. IEEE, 2015.

[61] Ka-Veng Yuen and Heung-Fai Lam. On the complexity of artificial neural networks for smart structures monitoring. *Engineering Structures*, 28(7):977–984, 2006.

[62] Jungwhee Lee and Sungkon Kim. Structural damage detection in the frequency domain using neural networks. *Journal of Intelligent Material Systems and Structures*, 18(8):785–792, 2007.

[63] Heung Fai Lam and Ching Tai Ng. The selection of pattern features for structural damage detection using an extended bayesian ann algorithm. *Engineering Structures*, 30(10):2762–2770, 2008.

[64] Michele Betti, Luca Facchini, and Paolo Biagini. Damage detection on a three-storey steel frame using artificial neural networks and genetic algorithms. *Meccanica*, 50(3):875–886, 2015.

[65] Onur Avci and Osama Abdeljaber. Self-organizing maps for structural damage detection: a novel unsupervised vibration-based algorithm. *Journal of Performance of Constructed Facilities*, 30(3):04015043, 2015.

[66] Prateek Prasanna, Kristin J Dana, Nenad Gucunski, Basily B Basily, Hung Manh La, Ronny Salim Lim, and Hooman Parvardeh. Automated crack detection on concrete bridges. *IEEE Trans. Automation Science and Engineering*, 13(2):591–599, 2016.

[67] Eloi Figueiredo, Gyuhae Park, Joaquim Figueiras, Charles Farrar, and Keith Worden. Structural health monitoring algorithm comparisons using standard data sets. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2009.

[68] Francisco de A Boldt, Thomas W Rauber, and FMV Ao. Feature extraction and selection for automatic fault diagnosis of rotating machinery. *X Encontro Nacional de Inteligencia Artificial e Computacional (ENIAC).-Fortaleza, Ceara*, pages 213–220, 2013.

[69] Jaime Vitola, Francesc Pozo, Diego A Tibaduiza, and Maribel Anaya. A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications. *Sensors*, 17(2):417, 2017.

[70] Guoqing Gui, Hong Pan, Zhibin Lin, Yonghua Li, and Zhijun Yuan. Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE Journal of Civil Engineering*, 21(2):523–534, 2017.

[71] X. Li, W. Yu, and S. Villegas. Structural health monitoring of building structures with online data mining methods. *IEEE Systems Journal*, 10(3):1291–1300, Sep. 2016.

[72] Osama Abdeljaber, Onur Avci, Mustafa Serkan Kiranyaz, Boualem Boashash, Henry Sodano, and Daniel J Inman. 1-d cnns for structural damage detection: Verification on a structural health monitoring benchmark data. *Neurocomputing*, 275:1308–1317, 2018.

[73] Osama Abdeljaber, Onur Avci, Serkan Kiranyaz, Moncef Gabbouj, and Daniel J Inman. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388:154–170, 2017.

[74] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, and Daniel J Inman. Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks. *Journal of Sound and Vibration*, 424:158–172, 2018.

[75] Yunsu Na, Sherif El-Tawil, Ahmed Ibrahim, and Ahmed Eltawil. The feasibility of using smart devices for quantifying seismic damage to buildings. In *Structures Congress 2017*, pages 145–154, 2017.

[76] S. Yao, Y. Zhao, A. Zhang, S. Hu, H. Shao, C. Zhang, L. Su, and T. Abdelzaher. Deep learning for the internet of things. *Computer*, 51(5):32–41, May 2018.

[77] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on instrumentation and measurement*, 57(1):140–149, 2008.

[78] Sheldon Ross. *A First Course in Probability 8th Edition*, chapter 5, page 219. Pearson, 2009.

[79] Piezotronics model 393c datasheet. Accessed on: March 11th, 2019.

[80] Android Documentation. App manifest overview. Accessed on: June 26th, 2019.

[81] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. Gyrophone: Recognizing speech from gyroscope signals. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 1053–1067, 2014.

[82] D. Hodges and O. Buckley. Reconstructing what you said: Text inference using smartphone motion. *IEEE Transactions on Mobile Computing*, 18(4):947–959, April 2019.

[83] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic. Side-channel inference attacks on mobile keypads using smartwatches. *IEEE Transactions on Mobile Computing*, 17(9):2180–2194, Sep. 2018.

[84] Amit Kumar Sikder, Giuseppe Petracca, Hidayet Aksu, Trent Jaeger, and A Selcuk Uluagac. A survey on sensor-based threats to internet-of-things (iot) devices and applications. *arXiv preprint arXiv:1802.02041*, 2018.

[85] Jeffrey Hood, Elizabeth Sall, and Billy Charlton. A gps-based bicycle route choice model for san francisco, california. *Transportation letters*, 3(1):63–75, 2011.

[86] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[87] T. Ergen and S. S. Kozat. Online training of lstm networks in distributed systems for variable length data sequences. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10):5159–5165, Oct 2018.

[88] R. Zazo, P. Sankar Nidadavolu, N. Chen, J. Gonzalez-Rodriguez, and N. Dehak. Age estimation in short speech utterances based on lstm recurrent neural networks. *IEEE Access*, 6:22524–22530, 2018.

[89] J. Zhao, H. Qu, J. Zhao, and D. Jiang. Towards traffic matrix prediction with lstm recurrent neural networks. *Electronics Letters*, 54(9):566–568, 2018.

[90] S. Dai, L. Li, and Z. Li. Modeling vehicle interactions via modified lstm models for trajectory prediction. *IEEE Access*, 7:38287–38296, 2019.

[91] Kundan Krishna, Deepali Jain, Sanket V Mehta, and Sunav Choudhary. An lstm based system for prediction of human activities with durations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):147, 2018.

[92] Android Documentation. System. Accessed on: June 26th, 2019.

[93] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209, 2012.

Appendix A

Stationary Process Covariance Matrix Row Summation

In this appendix, we derive an approximate summation formula of rows of a covariance matrix of a stationary random process that is characterized by having decaying covariance coefficients, i.e. $r_0 >> r_1 >> ....$ As mentioned in section 2.2.1, stationary process covariance matrix is expressed by (2.14). Hence, the summation of covariance matrix rows is calculated using (A.1). Let the summation of row k be denoted as η_k as shown by (A.2).

$$RQ = \begin{bmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_k & r_{k-1} & \dots & r_{n-k-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=0}^{n-1} r_j \\ r_1 + \sum_{j=0}^{n-2} r_j \\ \vdots \\ \sum_{j=1}^{k} r_j + \sum_{j=0}^{n-k-1} r_j \\ \vdots \\ \sum_{j=0}^{n-1} r_j \end{bmatrix} \tag{A.1}$$

$$\eta_k = \sum_{j=1}^{k} r_j + \sum_{j=0}^{n-k-1} r_j \tag{A.2}$$

If the stationary noise process is characterized by having decaying covariance coefficients values, i.e. $r_0 >> r_1 >> r_2 >> ...$, then the maximum value at any row is r_0, and the values decay and can be neglected compared to r_0, r_1 and the first few coefficients. Hence, approximately η_k for rows away from the boundaries is given by (A.3), and because of the covariance symmetry, η_k can be expressed by (A.4) which does not depend on k. Hence, by neglecting the error at the boundary rows, RQ is given by (A.5).

$$\eta_k \approx \eta = \sum_{j=1}^{\infty} r_j + \sum_{j=0}^{\infty} r_j \tag{A.3}$$

$$= \sum_{j=-1}^{-\infty} r_j + \sum_{j=0}^{\infty} r_j$$

$$= \sum_{j=-\infty}^{\infty} r_j \tag{A.4}$$

$$RQ = \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{n-1} \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \eta = Q\eta \tag{A.5}$$

Appendix B

Displacement Calculation

In this appendix, we derive the discrete time displacement as a function of acceleration, assuming that acceleration is considered constant in the duration between any two successive time samples. Discrete time velocity is calculated by (B.1), whereas displacement is calculated by (B.2). Equations (B.3) and (B.4) are the Z-transforms of (B.1) and (B.2) respectively. Hence, $S(z)$ is obtained by (B.5).

$$v[i] = \sum_{k=0}^{i-1} a[k]\Delta t \tag{B.1}$$

$$s[i] = s[i-1] + v[i-1]\Delta t + \frac{1}{2}a[i-1]\Delta t^2 \tag{B.2}$$

$$V(z) = \frac{z^{-1}}{1-z^{-1}}A(z)\Delta t \tag{B.3}$$

$$S(z) = z^{-1}S(z) + z^{-1}V(z)\Delta t + \frac{1}{2}z^{-1}A(z)\Delta t^2 \tag{B.4}$$

$$= z^{-1}S(z) + z^{-1}\frac{z^{-1}}{1-z^{-1}}A(z)\Delta t^2 + \frac{1}{2}z^{-1}A(z)\Delta t^2$$

$$= \frac{1}{1-z^{-1}}(z^{-1}\frac{z^{-1}}{1-z^{-1}}A(z)\Delta t^2 + \frac{1}{2}z^{-1}A(z)\Delta t^2)$$

$$= \frac{z^{-2}}{(1-z^{-1})^2}A(z)\Delta t^2 + \frac{1}{2}\frac{z^{-1}}{(1-z^{-1})}A(z)\Delta t^2 \tag{B.5}$$

By using the inverse Z-transform, $s[i]$ is expressed by (B.6) knowing that $a[i]$, $v[i]$ and $s[i]$ are zeros $\forall i < 0$.

$$s[i] = \sum_{k=0}^{i-1}\sum_{j=0}^{k-1} a[j]\Delta t^2 + \frac{1}{2}\sum_{k=0}^{i-1} a[k]\Delta t^2$$

$$= \sum_{k=0}^{i-1}(i-k-1)a[k]\Delta t^2 + \frac{1}{2}\sum_{k=0}^{i-1} a[k]\Delta t^2$$

$$= \sum_{k=0}^{i-1}(i-k-\frac{1}{2})a[k]\Delta t^2 \tag{B.6}$$