Automated Post-Earthquake Building Damage Assessment Using Smart Devices

by

Yunsu Na

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2019

Doctoral Committee:

Professor Sherif El-Tawil, Chair
Associate Professor Eunshin Byon
Assistant Professor Branko Kerkez
Associate Professor Jason P. McCormick

Yunsu Na

yunsu@umich.edu

ORCID ID: 0000-0003-3297-4822

© Yunsu Na 2019

DEDICATION

To my parents

JongKoo Na and Suim Ju

ACKNOWLEDGEMENTS

This dissertation is a journey marked by the culmination of hard work, sacrifice, support, and guidance of a number of individuals who have influenced and helped me over the years. I would like to take this opportunity to express my gratitude toward these individuals.

First, I would like to thank my advisor, Prof. Sherif El-Tawil, for his guidance and support over the past few years. Prof. El-Tawil inspires me to strive for excellence in every aspect of my professional career.

I would like to thank my other committee members, Prof. Jason McCormick, Prof. Branko Kerkez, and Prof. Eunshin Byon, for their advice and support in completing this work.

I would also like to thank Prof. Ahmed Eltawil and Ahmed Ibrahim at UC, Irvine for their excellent collaboration over the few years.

I would like to thank my editors who helped revise my writing: J.D. Jack Fishstrom, Dr. Christine Feak, Shelbi Bolter, and Jessica Freedland.

The next group of people to whom I owe a debt of gratitude are my fellow graduate students and scholars at the University of Michigan with whom I have grown as a researcher and as a person. In particular, I thank the brilliant research team of the Computational Structural Simulation Laboratory (CSSL), Dr. Tung-Yu Wu, Dr. Yuh Shiou Tai. Furthermore, I would like to

acknowledge Steve Donajkowski, Ethan Kennedy, Jan Pantolin, and Bob Spence for all their help in preparing my experiments.

The research presented in this dissertation was supported by the National Science Foundation (NSF) Graduate Student Research Fellowship under Grant No. 1362547 and 1362458.

I would like to thank the many friends I have made along the way who provided much needed stress relief through non-academic pursuits. In particular, Jukyung Kang, Sangyeop Kim, Kwangwoo Jung, Junghyo Lee, Sunghoon Hur, Dowon Park, and all of my soccer team members (FC KSAG).

Finally, I would like to thank my family for their support. My parents, Jongkoo Na and Suim Ju, who always told me that I could accomplish anything and lovingly supported every step of my academic journey. I owe them my deepest gratitude. Their unconditional love and support have been the one constant in my life and for that I am extremely grateful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Conventional practices to evaluate post-earthquake damage to buildings rely on reconnaissance teams deployed to the affected areas to carry out visual inspections of buildings. These inspections are done manually and are therefore time consuming and error prone. Motivated by these drawbacks, this dissertation explores the possibility and means for conducting automated inspections using smart devices, which are ubiquitous in modern communities. The premise is that smart devices can record acceleration data using their onboard sensors. The records can then be double integrated and processed to yield interstory drift ratios (IDRs), which are key indicators of building damage.

The dynamic behavior of a smart device on an underlying surface subjected to seismic motion is studied first. The smart device and its frictional interactions with the underlying surface are represented using a modified friction model. The conditions under which the smart device slides on or sticks to the surface for a given earthquake intensity are investigated. The concept of a 'probability of exceeding the slip limit curve' is introduced to relate the probability of exceeding a given slip limit for a given structure and location.

The presence of sliding motions in an acceleration record can contaminate the recorded data and make it impossible to estimate the motion of the underlying floor from smartphone measurements. To resolve this problem, stick-slip motion identification methods are studied based on two approaches. The first method relies on the theoretical observation that acceleration is constant during sliding. The second method employs two different types of machine learning algorithms to differentiate between sticking and slipping motions. It is shown that the developed techniques can yield reasonably high classification accuracies.

Computation of IDR requires multiple steps, each of which is theoretically investigated and experimentally validated by using a shake table and multiple types of smart devices with different types of protective shells. The needed steps include record synchronization and warping, data fusion, and compensation for errors that are magnified by double integration (needed to compute IDR). The abilities of different types of smart devices to estimate displacement were compared and the error in displacement was shown to have a strong relationship to their mean square amplitude of stationary noise. The proposed IDR estimation process is validated using the results from previously published shake table experiments of a four-story steel frame structure. It is shown that reasonable estimates of IDR can be achieved by using the developed methods.

CHAPTER 1

Introduction

1.1 General

Conventional practices to evaluate post-earthquake damage to buildings are labor intensive and time consuming. After an earthquake occurs, reconnaissance teams are deployed to the affected areas to carry out visual inspections of buildings, as outlined in the guidelines ATC-20 (Applied Technology Council 1989), ATC-20-1 (Applied Technology Council 2005) and ATC-20-2 (Applied Technology Council 1995). These operations require a considerable labor force in the case of large cities where several thousand buildings could be simultaneously damaged after a strong earthquake. Due to the manual and error prone nature of existing practices (Li et al. 2012), there is a clear need for the process to be improved.

This dissertation proposes smartphones as a means to automate damage assessment based on two important premises: 1) smartphones may be located throughout a building on each floor; and 2) sensors embedded in smartphones can measure accelerations, which can then be processed to yield explicit information about building damage. In particular, double integrating accelerometer data will yield floor displacements and, therefore, interstory drift ratios (IDR) along the building height, as shown in Figure 1-1. IDR is commonly used to represent fragility and vulnerability functions because of its direct correlation with building damage, as outlined in FEMA 350 (FEMA 2013). IDR is defined as the difference of displacements of the floors above and below the story of interest normalized by the interstory height. If IDR can be coupled with device location, this data can enable electronic building tagging. Using citizen-owned sensors can overcome the substantial challenge of installing accelerometer sensors manually in buildings and provide a way to monitor buildings using ubiquitous devices.

There have been numerous studies to investigate the usability of smartphones for earthquake engineering in a variety of areas. For example, a “iShake” system was developed using smartphones as seismic sensors to measure and deliver ground motion intensity parameters produced by earthquakes more rapidly and accurately than traditional methods (Ervasti et al. 2011). Smartphone data have also been used to track population movements in the aftermath of earthquakes (Bengtsson et al. 2011). Citizen-owned smartphones can provide early warnings of earthquake activity and a map of peak ground acceleration (Clayton et al. 2012). A global smartphone seismic network, MyShake, has been developed to build a global smartphone seismic network by utilizing the power of crowdsourcing. This study showed that an everyday handheld smartphone can be used as a portable seismometer by monitoring data from the accelerometer to detect earthquakes (Kong et al. 2015). Other research groups have shown that smartphones can be employed in health monitoring systems for bridges (Yu et al. 2015) and buildings (Kong et al. 2018). These efforts are mainly focused on classifying the occurrence of an earthquake or identifying straightforward parameters such as the fundamental frequencies of structures. However, none of the previously mentioned studies have addressed how IDR can be computed from data collected from smartphone accelerometers and used to directly assess seismic damage as is done herein.

Several topics were studied in this dissertation in order to make the proposed vision a reality. A summary of each of these areas is discussed next.

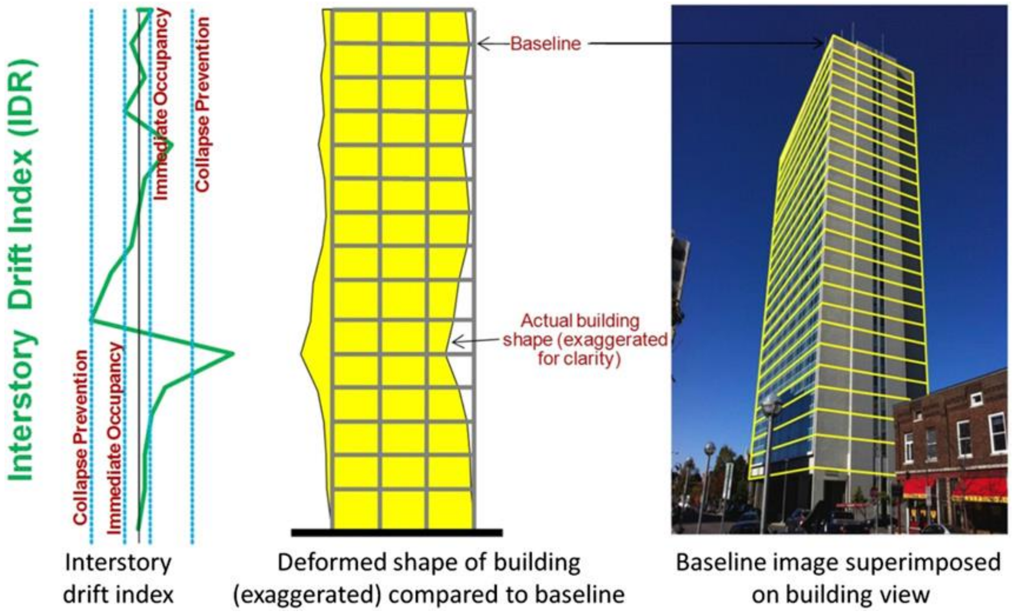


Figure 1-1. Computation of residual IDRs after a seismic event (Li et al. 2012)

1.1.1 Dynamic Behavior of an Unconstrained Smartphone under Seismic Shaking

The first step in using smartphones as earthquake damage sensors is to develop an understanding of how they move during seismic shaking. The intent of this part of the study is to solve the governing equations of motion of a smartphone sitting on a seismically excited base. For the sake of simplicity and without sacrificing accuracy, a smartphone is approximated as a rigid body (or block). The frictional response of an unconstrained block on a moving base can be divided into two main categories of behavior (sticking and sliding) and the block alternates between these two states as the supporting base moves back and forth due to seismic excitation. Because of its importance to accurately computing the motion of the block, “stick-slip” transition and the conditions under which the block is sliding are key concerns in this work.

1.1.2 Stick-Slip Characteristics of a Smartphone under Seismic Shaking

A key problem that hinders the vision for automated damage assessment is sliding motions, which may contaminate the recorded data making it impossible to estimate the motion of the underlying floor from the smartphone’s measurements. In order to detect floor acceleration, a smartphone must move in tandem with the floor. However, smartphones are usually just placed on furniture and not constrained in any way. If a smartphone has an adequate coefficient of friction with the underlying surface (e.g. encased in a rubber protective cover) and the underlying furniture is attached firmly to the floor, the smartphone’s movements will be representative of the floor’s movements. Otherwise, slip occurs. It is therefore necessary to identify the parts of the record during which slip occurs so as to exclude those from a smartphone’s readings of the shaking record.

1.1.3 Computing IDR from Acceleration Measurements

Acceleration measurements from a single smartphone includes substantial noise and may also be contaminated by slip action. Improvements in the reliability of the measured data may be achieved by stacking measurements from multiple phones. It is therefore necessary to develop techniques to reduce the effect of noise in the measured data, detect and eliminate sliding effects, fuse data from multiple sources, and reduce double integration errors. Once these issues are addressed and reliable estimates of the floor motion are obtained, the IDR values can be computed and used for damage assessment, fulfilling the vision of this work.

1.2 Objectives

The goals of this study are to: a) understand the dynamic behavior of an unconstrained smartphone under seismic motions; b) identify stick-slip characteristic of the smartphone; and c) estimate IDR from smartphones' acceleration measurements. There are three specific research objectives as follows:

- Develop tools to compute the dynamic behavior of an unconstrained smartphone under seismic shaking.
- Propose strategies to identify the stick-slip characteristics inherent in the motion of a smartphone on a moving surface.
- Propose methods for computing interstory drift ratios using data computed from multiple smartphones.

1.3 Structure of the dissertation

A brief description of each of the six chapters of this dissertation is provided below.

Chapter 1: Introduction. A general overview of the motivation and objectives of this research is presented. The scope of this work and an introduction to the content to follow is also provided.

Chapter 2: Dynamic Behavior of an Unconstrained Smartphone under Seismic Shaking. The behavior of a smart device on an underlying surface subjected to seismic motion is investigated. The smart device is modeled as a rigid block and its frictional interactions with the underlying base is represented using an existing model, modified for the purposes of this study. After validation of the results using experimental data, the revised model is used to study the sliding potential of smart devices on a surface during strong seismic events. This study identifies under what conditions a smart device (block) will stick to the underlying surface under seismic action. The concept of a probability of exceeding the slip limit curve is introduced and used to relate the probability of exceeding a given slip limit versus first period spectral acceleration for a given structure and location.

Chapter 3: Identify Stick-Slip Characteristics of a Smartphone on a Seismically Excited Surface. A method is presented to identify stick-slip motion from a smartphones' acceleration measurements. The method is based on identifying key characteristics in the acceleration record, specifically, the kinetic coefficient of friction. The noise associated with a smart device's measurement of acceleration is established and noise reduction methods to overcome them are compared. Computational simulation results and experimental data are used to demonstrate the concepts discussed in this Chapter.

Chapter 4: Stick-Slip Classification Based on Machine Learning Techniques. The sliding motion detection method in Chapter 3 is reasonably accurate. However, more accurate methods are needed to increase the effectiveness of the proposed damage evaluation methodology. To that end, machine learning is used to improve the sliding motion detection rate. Two types of machine learning methods are applied: supervised learning and deep learning (DL). To implement the supervised learning algorithms, acceleration data from an unconstrained smartphone undergoing motion with sticking and sliding components are processed to empirically select five features to aid in the classification process. Unlike the supervised learning algorithms, DL does not require prior knowledge beside the input data and automatically tunes the threshold parameters to identify stick-slip motion more accurately. The performance of both methods is contrasted and discussed.

Chapter 5: Stacking multiple device measurements to improve computation of interstory drift ratios. A process for estimation of IDR at each floor level is presented. The computed IDR values are compared to known limits to assess damage levels. The proposed methodology addresses noise reduction, sliding detection, data fusion, double integration errors, and interpolation for missing floors. A shake table test is conducted to showcase and validate the proposed process.

Chapter 6: Summary and Conclusions. Key findings and contributions that can be extracted from this work are summarized. Recommendations for future research related to estimating IDR using smartphones are also included.

1.4 Publications from this dissertation

Chapter 2 consists of work published in one journal paper. The work in Chapter 3 has been submitted for publication. The work in Chapter 4 and Chapter 5 are being prepared for publication and will soon be submitted. Details are listed below:

Na, Y., El-Tawil, S., Ibrahim, A., and Eltawil, A. (2018). "Dynamic behavior of a smartphone on a surface subjected to earthquake motion." *Earthquake Engineering & Structural Dynamics*. **(Chapter 2)**

Na, Y., El-Tawil, S., Ibrahim, A., and Eltawil, A. "Identifying Stick-Slip Characteristics of a Smartphone on a Seismically Excited Surface using On-Board Sensors." *J. of Earthquake Engineering,* Accepted Feb, 2018. **(Chapter 3)**

Na, Y., El-Tawil, S., Ibrahim, A., and Eltawil, A. "Classifying stick-slip motion of a smartphone by applying machine learning algorithms," In preparation. **(Chapter 4)**

Na, Y., El-Tawil, S., Ibrahim, A., and Eltawil, A. "Assessment of Interstory Drift measurements using smartphone sensors." *Journal of Structural Engineering,* Submitted Dec, 2018 **(Chapter 5)**

CHAPTER 2

Dynamic behaviour of a smart device on a surface subjected to earthquake motion

2.1 General

This chapter begins with an overview of the literature relevant to friction models, which are necessary to model the sticking and sliding behavior of smart devices on an underlying moving surface. A friction model is adopted and modified for the purpose of this study. The modified friction model is validated by comparing its behavior to experimentally measured data. After validation, the modified friction model is used to study the sliding potential of smart devices. Sliding spectra of selected records are provided for two-dimensional motion and "probability of exceeding the slip limit" curves are subsequently derived, and their potential application discussed.

2.2 Literature review

The frictional response of an unconstrained block on a moving base can be divided into two main categories of behavior: sticking (pre-sliding) and slipping (sliding) (Hsieh and Pan 2000). The block transitions between sticking and slipping as the supporting base moves back and forth because of seismic excitation. At incipient sliding, in the so-called micro-slip zone, small relative motion occurs between the block and base (Parlitz et al. 2004). The Stribeck (or velocity weakening) effect is characterized by a decrease in the friction force with increasing relative velocity, and frictional lag, where there is hysteresis in the friction versus velocity response.

Once the applied inertial forces exceed the frictional resistance, the block moves beyond the micro-slip zone and starts sliding in earnest. Because of its importance to accurately compute the motion of the block, modeling “stick-slip” transition is a main concern in this work.

Numerous friction models are available in the literature, ranging from simple methods, which ignore key friction phenomena, to sophisticated ones that account for multiple considerations. The Coulomb law of dry friction is widely used in contact problems because of its simplicity. The Coulomb friction force, which acts opposite to the direction of motion of the body with respect to the moving base, is simply equal to the normal force between two bodies multiplied by the kinetic coefficient of friction.

To incorporate the Stribeck effect and the effect of pre-sliding displacements, De Wit et al (1995) proposed the LuGre model, which is an extension of the Dahl model (Dahl 1975). This model is based on a bristle-like interpretation of the frictional interface. However, the LuGre model behaves like a linear spring/damper pair when it is linearized for small velocities, and as such, Parlitz et al. (2004) and Choi et al. (2006) criticized it as incapable of adequately considering pre-sliding hysteresis. Lampaert et al. (2002) and Swevers et al.(2000) attempted to address this problem by incorporating a Maxwell slip model. However, the resulting models are complex and difficult to extend to two dimensions.

To simulate sticking and slipping behavior in contact problems with system damping and stiffness, Karnopp (1985) developed a force-balance friction model for one-dimensional motion. Beyond a predefined velocity window that signals when sliding is occurring, the friction force is described as a function of the sliding velocity. Tan and Rogers (1996) extended Karnopp's one-dimensional model to two dimensions. Their model exhibited “numerical chattering” in the velocity response when the tangential velocity was close to zero. To address this issue, Tariku and Rogers (2000) introduced the concept of a micro-slip region, where the friction force acts opposite to the net external force, rather than the direction of the sliding velocity, which they argued is not reliable in this region. Based on this idea, they presented a new method to alleviate the chattering problem. In addition to a sticking spring, Tariku and Rogers (2001) and Antunes et al. (1990) introduced a sticking damper when a sticking state is detected to eliminate the spurious local vibrations associated with the abrupt change in the friction force direction. However, the sticking damper properties, as well as the sticking spring properties, have no physical meaning.

Several categories of problems in earthquake engineering make use of friction models. Among the largest are base isolation systems and geotechnical engineering problems. Of particular interest to the ideas in this chapter is the sliding behavior of monuments and blocks lying on the ground. For example, using Coulomb's law, Gazetas et al. (2009) introduced the idea of the "sliding potential" of a rigid block resting on horizontal or inclined planes subjected to horizontal and vertical motion. They used the sliding potential as a measure of the capacity of earthquakes to induce damage in sliding systems and introduced one-dimensional sliding spectra of selected ground motion records. Other studies of sliding blocks can be found in Gazetas et al.(2012) and Westermo and Udwadia (1983). Most of the studies in this category of problems have focused on one-dimensional problems and not two, as done herein. Also, to the knowledge of the author, this is the first study to focus on the behavior of smart devices.

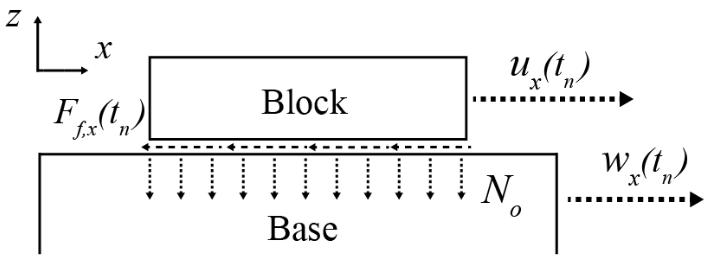
2.1 Friction models for an unconstrained smart device

Figure 2-1a shows the system under consideration, where a block sits on a horizontal base. The base is subjected to some type of motion described by an acceleration time history, $\ddot{w}_x(t_n)$. The block on the base can undergo stick-slip motion without restriction, i.e. it is not constrained in any way. For simplicity, the normal load, N_o, is assumed a constant value, mg, where m is the mass of the block and g is the acceleration due to gravity. This assumption is relaxed later on in the chapter, where variable vertical acceleration is taken into account. It is also assumed that the block and base are rigid and always in contact, i.e, separation because of vertical acceleration does not occur. The equation of motion in the *x* direction is defined as:

$$m\ddot{u}_x(t_n) = F_{f,x}(t_n) \qquad \text{2-1}$$

where $\ddot{u}_x(t_n)$ is the acceleration of the block in the x direction at time, t_n, and $F_{f,x}(t_n)$ is the friction force acting between the base and the block in the x direction. The solution of Equation 2-1 depends on the state of motion (sliding or sticking) and the friction model used.

a)



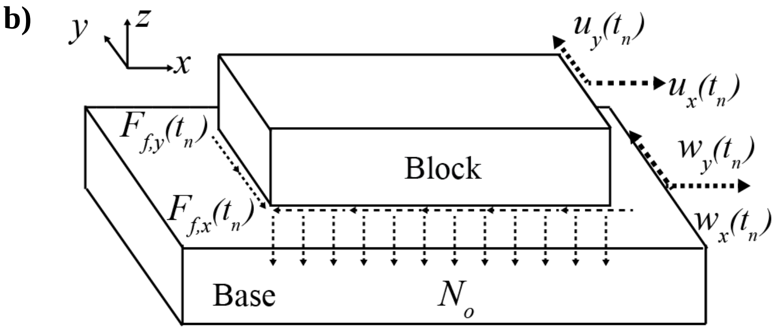


Figure 2-1. Problem setup: a block on a base for a) one-dimensional motion and b) two-dimensional motion

In two dimensions, the equation of motion is presented as follows:

$$\begin{aligned} m\ddot{u}_x(t_n) &= F_{f,x}(t_n) \\ m\ddot{u}_y(t_n) &= F_{f,y}(t_n) \end{aligned} \tag{2-2}$$

where $\ddot{u}_y(t_n)$ is the acceleration of the block in the y direction at time, t_n, and $F_{f,y}(t_n)$ is the friction force acting between the base and the block in the y direction. For simplicity, the relative acceleration, velocity and displacement between the base and the block in x and y directions at time (t_n) are defined as $\ddot{X}(t_n)$, $\ddot{Y}(t_n)$, $\dot{X}(t_n)$, $\dot{Y}(t_n)$, $X(t_n)$ and $Y(t_n)$, respectively.

2.1.1 LuGre Model

De Wit et al (1995) developed the LuGre model by assuming that the friction interface between two surfaces comprises contact between bristles. These bristles provide resistance to motion when there is a relative velocity between the two contact surfaces giving rise to the friction force, which is represented as follows:

$$F_{f,x}(t_n) = -\sigma_0 z(t_n) - \sigma_1 \dot{z}(t_n) - \sigma_2 \dot{X}(t_n) \qquad \text{2-3}$$

$$\dot{z}(t_n) = \dot{X}(t_n) - \frac{\left|\dot{X}(t_n)\right|}{r\left(\dot{X}(t_n)\right)} z(t_n) \qquad \text{2-4}$$

$$r\left(\dot{X}(t_\mathrm{n})\right) = \frac{1}{\sigma_0}\left(F_c + (F_s - F_c)e^{-|\dot{X}(t_n)/\, v_s|^2}\right) \qquad \text{2-5}$$

where $F_{f,x}(t_n)$ is the friction force in the *x* direction, $z(t_n)$ and $\dot{z}(t_n)$ are the average deflection of the bristles and its derivative; σ_0, σ_1, and σ_2 are the stiffness, damping, and viscous friction coefficients, respectively; F_c, F_s and v_s are the Coulomb friction, static friction and Stribeck velocity, respectively; and $r\left(\dot{X}(t_n)\right)$ describes the Stribeck behavior. Note that v_s is a small slip velocity, below which microslip behavior occurs.

To extend the LuGre model to two dimensions, the friction force is split into components parallel to those of the instantaneous sliding velocity unit vector, which consequently results in coupling of the *x* and *y* equations of motion. This force distribution method is based upon the Maximal Dissipation Rate Principle (Sorine 1998) for isotropic situations (Velenis et al. 2005). The two-dimensional equations of motions are presented as follows:

$$\begin{bmatrix} F_{f,x}(t_n) \\ F_{f,y}(t_n) \end{bmatrix} = -\sigma_0 \begin{bmatrix} z_x(t_n) \\ z_y(t_n) \end{bmatrix} - \sigma_1 \begin{bmatrix} \dot{z}_x(t_n) \\ \dot{z}_y(t_n) \end{bmatrix} - \sigma_2 \begin{bmatrix} \dot{X}(t_n) \\ \dot{Y}(t_n) \end{bmatrix} \qquad \text{2-6}$$

$$\begin{bmatrix} \dot{z}_x(t_n) \\ \dot{z}_y(t_n) \end{bmatrix} = \begin{bmatrix} \dot{X}(t_n) \\ \dot{Y}(t_n) \end{bmatrix} - \begin{bmatrix} \dfrac{\left|\dot{X}(t_n)\right|}{r_x\left(\dot{X}(t_n)\right)} z_x(t_n) \\ \dfrac{\left|\dot{Y}(t_n)\right|}{r_y\left(\dot{Y}(t_n)\right)} z_y(t_n) \end{bmatrix} \qquad \text{2-7}$$

$$\begin{bmatrix} r_x\left(\dot{X}(t_n)\right) \\ r_y\left(\dot{Y}(t_n)\right) \end{bmatrix} = \frac{1}{\sigma_0\sqrt{\dot{X}(t_n)^2 + \dot{Y}(t_n)^2}} \begin{bmatrix} \left|\dot{X}(t_n)\right| \\ \left|\dot{Y}(t_n)\right| \end{bmatrix} \left[F_c + (F_s - F_c) \begin{bmatrix} e^{-(|\dot{X}(t_n)|/\, v_s)^2} \\ e^{-(|\dot{Y}(t_n)|/\, v_s)^2} \end{bmatrix} \right] \tag{2-8}$$

where $F_{f,y}(t_n)$ is the friction force component in the *y* direction, $z_y(t_n)$ is the average deflection of the bristles in the *y* direction, $r_y\left(\dot{Y}(t_n)\right)$ describe the Stribeck behavior in the *y* direction. The solution of Equation 2-3 for one dimensional and Equation 2-6 for two-dimensional motion depends on sticking-sliding state as elaborated upon next.

2.1.2 Modified Sticking-Spring-Damper Friction Model (SSDFM)

The friction model used for most of the analyses in this chapter is an extension of the model proposed by Tariku and Rogers (2001) and Antunes et al. (1990). Termed the sticking-spring-damper friction model (SSDFM), it is geared toward application to an unconstrained block (smart device) on a moving base subjected to seismic excitation. Two changes are made: 1) an interpolation technique is introduced to enhance detection of transition points, which occur frequently during seismic motion, and 2) the model is extended to handle vertical accelerations whereas the original model only considered constant vertical acceleration.

When sticking is detected, an imaginary tangential sticking spring with stiffness $K_{f,x}$ (in the *x* direction) and sticking damper with sticking damping coefficient $C_{f,x}$ (in the *x* direction) are inserted between the base and the bottom of the block. Combined, the sticking spring and damper force components represent the total friction force. The model detects sticking from sliding when three conditions are concurrently satisfied: 1) the relative velocity changes its sign; 2) the inertial force is less than the Coulomb static friction force; and 3) the sticking spring-damper force is smaller than or equal to the Coulomb static friction force. When the sticking friction force is greater than the Coulomb static friction force, the sticking state is broken and sliding initiates and becomes governed by the kinetic friction force.

For two-dimensional motion on a base, the sliding friction force in Equation 2-9 is divided into *x* and *y* components, parallel to those of the instantaneous sliding velocity unit vector, as follows:

$$\begin{bmatrix} F_{f,x}(t_n) \\ F_{f,y}(t_n) \end{bmatrix} = -\frac{\mu_k N_o}{\sqrt{\left(\dot{X}(t_n)\right)^2 + \left(\dot{Y}(t_n)\right)^2}} \begin{bmatrix} \dot{X}(t_n) \\ \dot{Y}(t_n) \end{bmatrix} \tag{2-9}$$

where μ_k is kinetic coefficient of friction and $F_{f,y}(t_n)$ is the friction force component in the *y* direction.

The transition from sticking to sliding commences when the magnitude of the external force (in this case inertial) is greater than the Coulomb static friction force. Following Tariku and Rogers (2001), it is assumed that the friction force acts opposite to the external force, rather than opposite to the direction of the sliding velocity in the micro-slip region, until the sliding velocity reaches 0.1 mm/s and has a consistent direction. The components of the friction force are calculated as a function of the acceleration vector as follows:

$$\begin{bmatrix} F_{f,x}(t_n) \\ F_{f,y}(t_n) \end{bmatrix} = \frac{\mu_k N_o}{\sqrt{\ddot{w}_x(t_n)^2 + \ddot{w}_y(t_n)^2}} \begin{bmatrix} \ddot{w}_x(t_n) \\ \ddot{w}_y(t_n) \end{bmatrix} \tag{2-10}$$

The set of equations is solved numerically using the Runge Kutta method.

Transition points are considered to occur when: 1) sliding commences from sticking, 2) the sign of a component of the sliding direction reverses, and 3) sliding ends, leading to sticking. Using a constant time step for numerical simulations (as done in the traditional friction models) makes it difficult to find an accurate transition point. Instead of using the constant time step, an interpolation technique is applied to more accurately compute transition points and prevent overshooting. Therefore, at transition points, the time step is discretized into multiple sub-steps using a cubic spline. Experimentation with various numbers of steps showed that 20 sub-steps produce a reasonable answer as discussed later on in Section 2.4.2.

Figure 2-2 shows how the transition point is found. At transition, the velocity of the base and block are almost the same, and the block stops slipping or changes its slip direction depending on the friction forces at play. As shown in Figure 2-2, the larger time steps used during sliding (designated with stars [☆] in Figure 2-2) causes overshooting behavior. Once overshooting is detected (when the sign of relative velocity reverses), the solution process steps back to the last point before overshooting and restarts the computation with a smaller time step (1/20th of the original step) leading to more accurate detection of the transition point (designated 1st transition point in Figure 2-2). The smaller time steps are shown in Figure 2-2 as diamonds (◊). The larger time step is again reinstituted. As shown in Figure 2-2, the large time-step again misses the transition point (2nd transition point in Figure 2-2). The interpolation process is repeated (shown by triangles [▵] then circles (o) in Figure 2-2) to avoid excessive overshooting.

The SSDFM algorithm, which is based on Tariku and Rogers (2000) and modified as outlined earlier, is applied at each time step t_n where n refers to the present time step and n-1 refers to the previous time as follows:

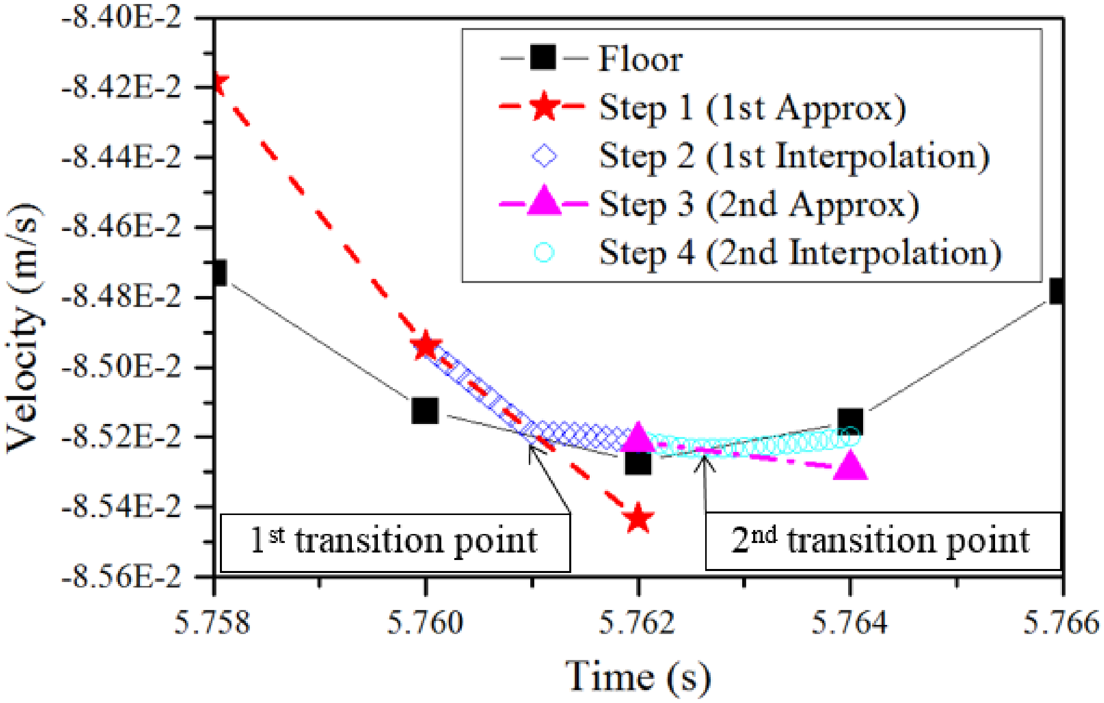


Figure 2-2. Detail of interpolation scheme to accurately identify transition points

1 If the previous state at time t_{n-1} is sliding, then assess sticking by checking to see if the sign of the relative velocity changes. Compute β_x and β_y:

$$\beta_x = \dot{X}(t_n)\cdot\dot{X}(t_{n-1})$$

2-11

$$\beta_y = \dot{Y}(t_n)\cdot\dot{Y}(t_{n-1})$$

1.1 If both β_x and β_y are positive, the system is still sliding. Compute friction forces from Equation 2-9.

1.2 If either one or both of β_x and β_y are zero or negative, compute the inertial force components, $F_{i,x}$ and $F_{i,y}$, acting on the block, and the Coulomb static friction force with static coefficient of friction, μ_s:

$$F_{i,x}(t_n) = m\ddot{w}_x(t_n)$$

$$F_{i,y}(t_n) = m\ddot{w}_y(t_n) \qquad \text{2-12}$$

$$F_s = N_o\mu_s$$

1.2.1 If $\sqrt{\left(F_{i,x}\right)^2 + \left(F_{i,y}\right)^2} > F_s$, then the system is sliding. Compute the friction forces using Equation 2-9.

1.2.2 If $\sqrt{\left(F_{i,x}\right)^2 + \left(F_{i,y}\right)^2} \leq F_s$, then

1.2.2.1 Compute the transition points X_{st} and Y_{st} between $X(t_{n-1})$ and $X(t_n)$ and between $Y(t_{n-1})$ and $Y(t_n)$, respectively by interpolation. Go back one step ($n = n$-1) and restart the computation using Equation 2-9 with 1/20th of the original step until the transition points X_{st} and Y_{st} are reached, defined as the points where the sign of the relative velocity reverses.

1.2.2.2 Insert imaginary springs and dampers between the base and block in both x and y directions. Calculate sticking friction force components $F_{o,x}(t_n)$ and $F_{o,y}(t_n)$:

$$F_{o,x}(t_n) = -K_{f,x}[X(t_n) - X_{st}] - C_{f,x}\dot{X}(t_n)$$

$$\text{2-13}$$

$$F_{o,y}(t_n) = -K_{f,y}[Y(t_n) - Y_{st}] - C_{f,y}\dot{Y}(t_n)$$

1.2.2.3 If $\sqrt{\left(F_{o,x}(t_n)\right)^2 + \left(F_{o,y}(t_n)\right)^2} > F_s$, the system is sliding. In this case, the friction forces are given by Equation 2-9.

1.2.2.4 If $\sqrt{\left(F_{o,x}(t_n)\right)^2 + \left(F_{o,y}(t_n)\right)^2} \leq F_s$, the system is sticking. Compute the sticking friction force by checking the conditions one-step back:

1.2.2.4.1 If $\sqrt{\left(F_{o,x}(t_{n-1})\right)^2 + \left(F_{o,y}(t_{n-1})\right)^2} \geq \sqrt{\left(F_{f,x}(t_{n-1})\right)^2 + \left(F_{f,y}(t_{n-1})\right)^2}$, then the friction force becomes the sticking friction force $F_{f,x}(t_n) = F_{o,x}(t_n)$ and $F_{f,y}(t_n) = F_{o,y}(t_n)$.

1.2.2.4.2 If $\sqrt{\left(F_{o,x}(t_{n-1})\right)^2 + \left(F_{o,y}(t_{n-1})\right)^2} < \sqrt{\left(F_{f,x}(t_{n-1})\right)^2 + \left(F_{f,y}(t_{n-1})\right)^2}$, then go back one step ($n = n$-1). Compute new $F_{o,x}(t_n)$ and $F_{o,y}(t_n)$ from Equation 2-13. Check for the sticking friction force as done in 1.2.2.4.1.

2. If the previous state at time t_{n-1} is sticking, then compare the sticking friction force computed from Equation 2-13 and the Coulomb static friction force F_s

2.1 If $F_s \geq \sqrt{\left(F_{o,x}(t_n)\right)^2 + \left(F_{o,y}(t_n)\right)^2}$, the block continues to stick and the friction forces are computed from Equation 2-13.

2.2 If $F_s < \sqrt{\left(F_{o,x}(t_n)\right)^2 + \left(F_{o,y}(t_n)\right)^2}$, then sliding occurs.

2.2.1 If the relative velocities are greater than 0.1 mm/s, the friction force is given by Equation2-9.

2.2.2 If the relative velocities are less than 0.1 mm/s, the friction force is given by Equation 2-10.

2.1.3 Device on a Table on a Moving Floor

The proposed friction model can be readily applied to the double stacked body problem as shown in Figure 2-3, where it is assumed that the smart device sits on a table or other piece of furniture, which in turn sits on a moving base. The table or furniture is assumed to be a rigid body. The sliding criteria for double stacked bodies are presented as follows:

$$\left|\ddot{u}_g\right| < \mu_1 g\ ,\ \left|\ddot{u}_g + \ddot{u}_1\right| < \mu_2 g \qquad \text{2-14}$$

where u_1 and u_2 are the relative displacements of the table with respect to the base and of the smart device with respect to the table, respectively; μ_1 and μ_2 are the coefficients of friction between the base and the table and between the table and smart device, respectively.

If the coefficient of friction between the table and smart device is higher than that between the base and the table ($\mu_2 > \mu_1$), the smart device does not slide with respect to the table. In other words, the table slides while the smart device sticks to it. The situation is more complex when $\mu_2 < \mu_1$. In this case, the table can slide with respect to the base and the smart device can also slide with respect to the table. Because the mass of the smart device is generally negligible with respect to that of the table, the problem can be uncoupled into two independent parts, where the motion of the table or smart device are computed as a function of the movement of the underlying surface. In either case, the motion can be computed using the algorithm in section 2.3.2.

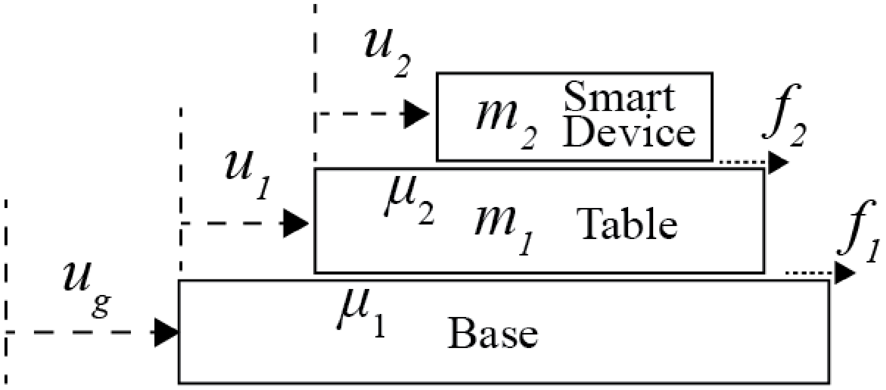


Figure 2-3. Schematic diagram of the double stacked bodies

2.2 Parametric studies

Two key parameters, sticking spring stiffness and number of interpolation points, are studied to show how these parameters affect the simulation results. In the following, it is assumed that m=1 kg, μ_s=0.45, μ_k=0.4, N_o=9.81 N, and $C_{f,x}$=191 N·s/m. The time step of 0.1 millisecond is chosen to be less than one thousandth of the period of the highest frequency, among the natural, excited and normal force frequencies as outlined in Tariku and Rogers (2001). The motion of the surface in the simulations is modeled using cosinusoidal functions with randomly generated amplitude and frequency. The amplitude range is 0.1~3 m/s and the frequency range is 0.2π~3π rad/s in the x and y directions. An error is defined as the deviation of the smallest computed sticking velocity of the block from the velocity of the base at the first transition from slipping to sticking. In the subsequent discussion, errors are normalized by dividing them by the maximum amplitude of the velocity record.

2.2.1 Effect of the Applied Amplitude and Sticking Spring Stiffness

Figure 2-4 shows the effects of the amplitude of the applied base velocity and the sticking spring stiffness of SSDFM on the normalized sticking velocity error. In Figure 2-4, the amplitude of the applied velocity ranges from 0.8 through 2.89 m/s, and the sticking spring stiffness ranges from 1,000 to 80,000 N/m. Figure 2-4 shows that the sticking velocity errors of SSDFM decrease asymptotically as the sticking spring stiffness increases. It appears that increasing the sticking spring stiffness beyond 50,000 N/m does not result in significant reduction in the sticking velocity error. Also, evident from Figure 2-4 is that selecting a larger amplitude for the cosinusoidal function results in a lower normalized sticking velocity error. In particular, the larger amplitude (2.89 m/s) function has a lower normalized sticking velocity than the smaller amplitude (1.30 m/s and 0.80 m/s) functions.

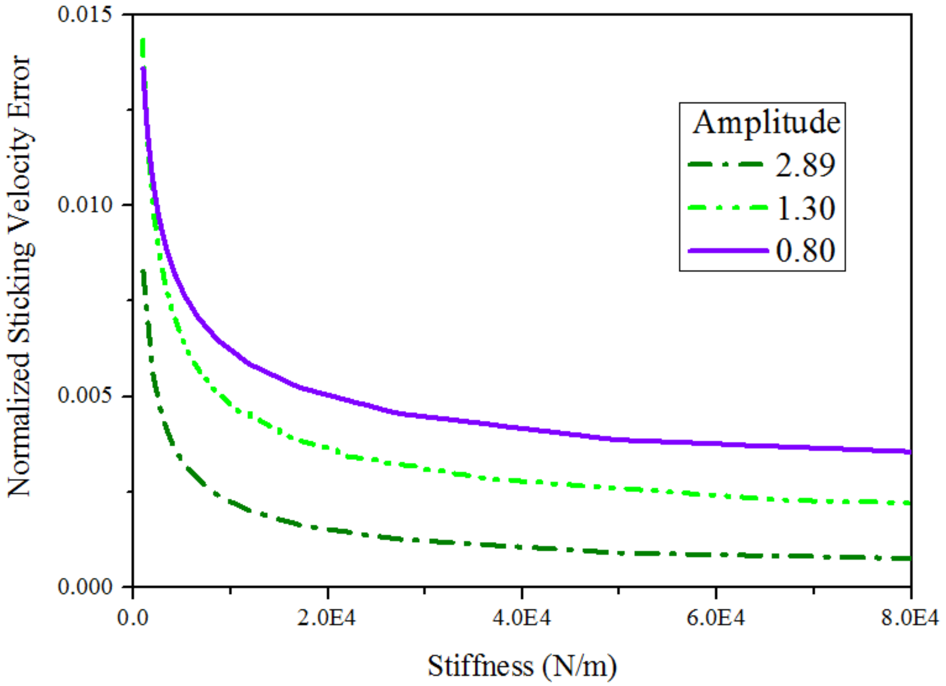


Figure 2-4. Effect of stiffness of spring and amplitude of applied velocity

2.2.2 Effect of the Number of Interpolation Points

Figure 2-5 shows the effect of the number of interpolation points at the transition points on solution accuracy. It is clear that increasing the number of interpolation points results in a lower normalized sticking velocity error, but leads to longer computation time. Figure 2-5 shows that the rate of improvement in reducing the velocity error flattens out considerably at about 20 points, the number adopted in this work.

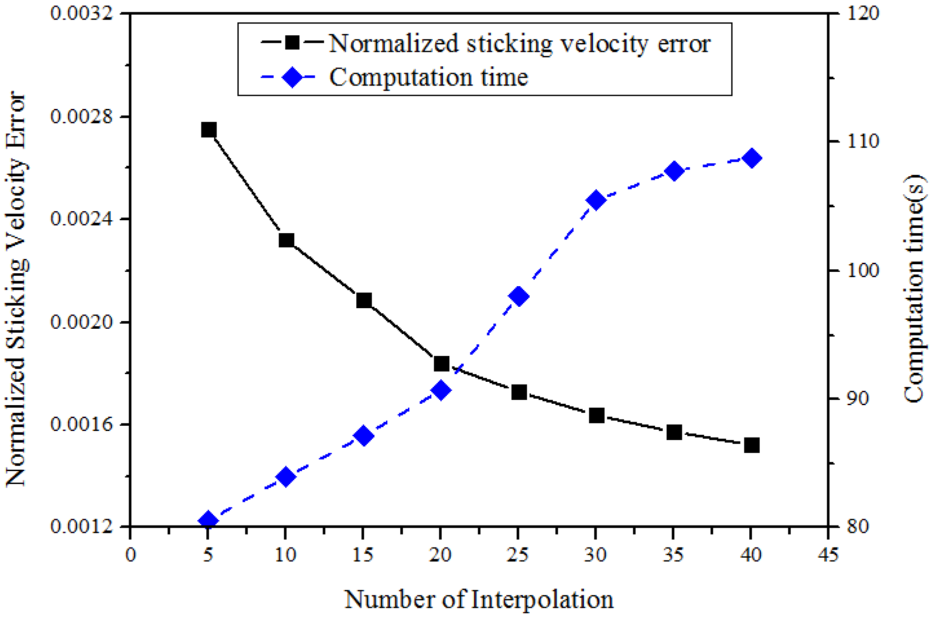


Figure 2-5. Effect of number of interpolation in the proposed model

2.2.3 Statistical Error Analysis

For the purposes of validation, a two-dimensional dynamic system is solved using both the LuGre model and SSDFM, focusing in particular on transition points. To compare the performance of LuGre and SSDFM, the motion of the surface in the simulations is modeled using cosinusoidal functions with randomly generated amplitude and frequency. The amplitude range is 0.1~3 m/s and the frequency range is $0.2\pi \sim 3\pi$ rad/s. The key parameters are m=1 kg, μ_s=0.45, μ_k=0.4 and N_o=9.81 N. For SSDFM, $K_{f,x}$=50,000 N/mm and $C_{f,x}$=191 N·s/m. The time step of 0.1 ms is chosen to be less than one thousandth of the period of the highest frequency, among the natural, excited and normal force frequencies as outlined in Tariku and Rogers (2001). For the LuGre model, σ_0 = 50,000 N/m, $\sigma_1 = \sqrt{\sigma_0}$, σ_2 =0.4 , v_s=0.001, and $F_s = 1.125F_c$ are selected as outlined in (De Wit et al. 1995). Dynamic responses of the system are simulated using the algorithms and solutions discussed in Section 2.3.2. A total of 300 simulation cases for two-dimensional motion are conducted.

The statistical properties of the sticking velocity error appear to follow a Weibull distribution (Shigley and Mitchell 1983), which is characterized by four variables: R, θ, b and e_o. R (reliability) is the cumulative density function complementary to unity. The characteristic value, θ, is the abscissa point corresponding to a reliability of 0.368 or $\ln\ln(1/R)$=0. The shape parameter, b, is the slope of the fitted straight line. e_o is a guaranteed normalized sticking velocity. The Weibull distribution requires that the plot of $\ln\ln(1/R)$ versus ln (Normalized sticking velocity error) forms a straight line as shown in Figure 2-6. Following the approach in Shigley and Mitchell (1983), the values of the three Weibull parameters (e_o, θ, and b) for the two friction models are summarized in Table 2-1. Using these parameters, the mean and the standard deviations of the normalized sticking velocity errors are calculated using the formulas for Weibull distributions when the sticking spring stiffness of 50,000 N/m is used.

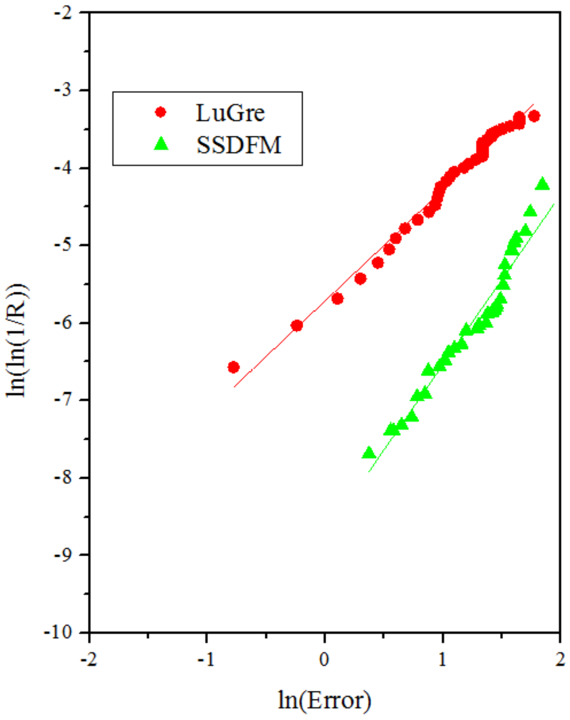


Figure 2-6. Relationship between reliability and error

Table 2-1. Weibull parameters for the normalized sticking velocity error distributions

Model	e_o	θ	b
LuGre	0	3.44E-3	7.77E-1
SSDFM	0	1.98E-3	7.13E-1

Table 2-2 summarizes the statistical parameters for the normalized sticking velocity errors computed from the SSDFM and LuGre. It is clear from Table 2-2 that SSDFM has a smaller mean than the LuGre model (3.98E-3 for LuGre, and 2.02E-3 for SSDFM), smaller standard deviation (5.18E-3 for LuGre, 3.67E-3 for SSDFM), smaller minimum (1.29E-5 for LuGre, and

1.13E-5 for SSDFM) and smaller maximum (7.66E-2 for LuGre, and 5.87E-2 for SSDFM) normalized sticking velocity error values.

Another way of comparing the SSDFM and LuGre is to compute the reliability, which shows the level of consistency expected from a model. The reliability is calculated as shown in Equation 2-15.

$$R(x) = \exp(-[(e - e_o)/(\theta - e_o)]^b) \qquad 2\text{-}15$$

where e is the normalized sticking velocity error. Figure 2-7 shows the normalized sticking velocity error probability curves for the two friction models given by $P(x)$=100$[1 - R(x)]$. The SSDFM gives smaller normalized sticking velocity errors for the same percent probability than the LuGre model. For example, in comparing the two points corresponding to 95 percent confidence on the two curves of Figure 2-7, the maximum expected sticking velocity error of SSDFM (0.010) is smaller than that of LuGre (0.014). The above discussion suggests that the SSDFM is modestly better than the LuGre model, especially in reducing the sticking velocity error at the transition points and is the reason SSDFM is selected for the subsequent studies.

Table 2-2. Statistics of the normalized sticking velocity errors

Model	Mean	Standard Deviation	Minimum	Maximum
LuGre	3.98E-3	5.18E-3	1.29E-5	7.66E-2
SDDFM	2.02E-3	3.67E-3	1.13E-5	5.87E-2

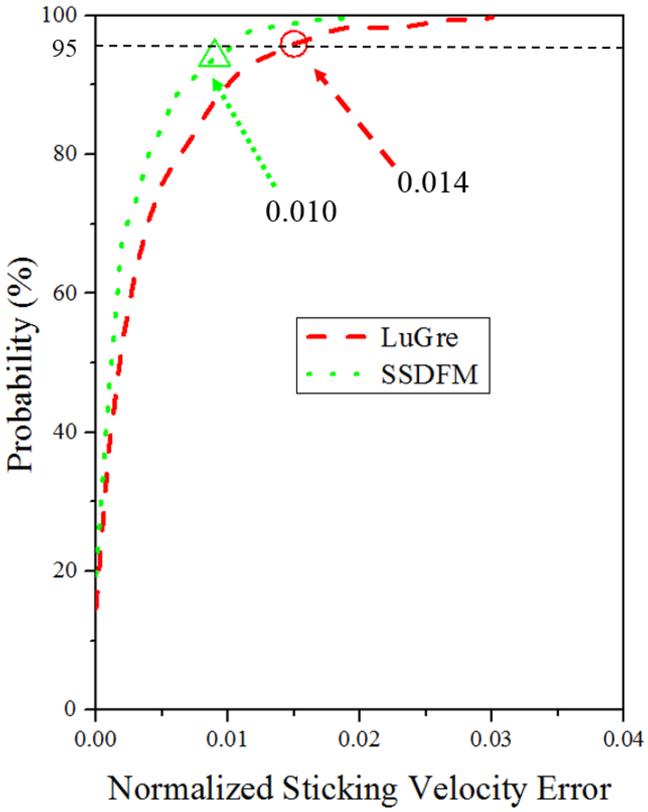


Figure 2-7. Probability of normalized sticking velocity error

2.3 Comparison to experimental results

Shake table experiments are conducted to show that the proposed model can reasonably represent the motion of an actual smart device on a seismically excited surface.

2.3.1 Experimental Setup

The smart device selected is a Samsung Galaxy S7 (model SM-G930). The 'Sensor kinetics pro' Android application by Innoventions, Inc., is used to record the phone's acceleration data. An infrared 3D-motion capture system (Optotrak) is used to capture the true displacement of the smart device and shake table platen. The tracking system can monitor 3D positions of a set of markers with an accuracy of 0.1 mm and resolution of 0.01mm at a distance of about 3 m. Two markers are placed on the platen and another two are placed on the phone to measure the displacement time histories. Corresponding velocities and accelerations are obtained by differentiating the displacement data.

A shake table was custom-built for simulation of ground motions. The shake table consists of a platen driven by a linear actuator and step motor. The step motor is controlled by a step motor driver and an Arduino single-board microcontroller. LabView is used to feed signals to the shake table to simulate earthquake motion. The shake table can only represent one-dimensional motion because of its configuration. Therefore, in this study, it is assumed that, for the sake of simplicity, smartphones are oriented to measure only one-dimensional horizontal motion without vertical motion or lift off. Figure 2-8 shows the overall experimental setup with the smart phone, infrared sensors, shake table, platen, and Optotrak system. The maximum velocity of this shake table is 35cm/s and the maximum stroke is ±30cm. The shake table can produce precise, repetitive motions for a given ground motion so that each smartphone can record multiple rounds of acceleration measurements that are aggregated to reduce the noise as discussed in Chapter 4.

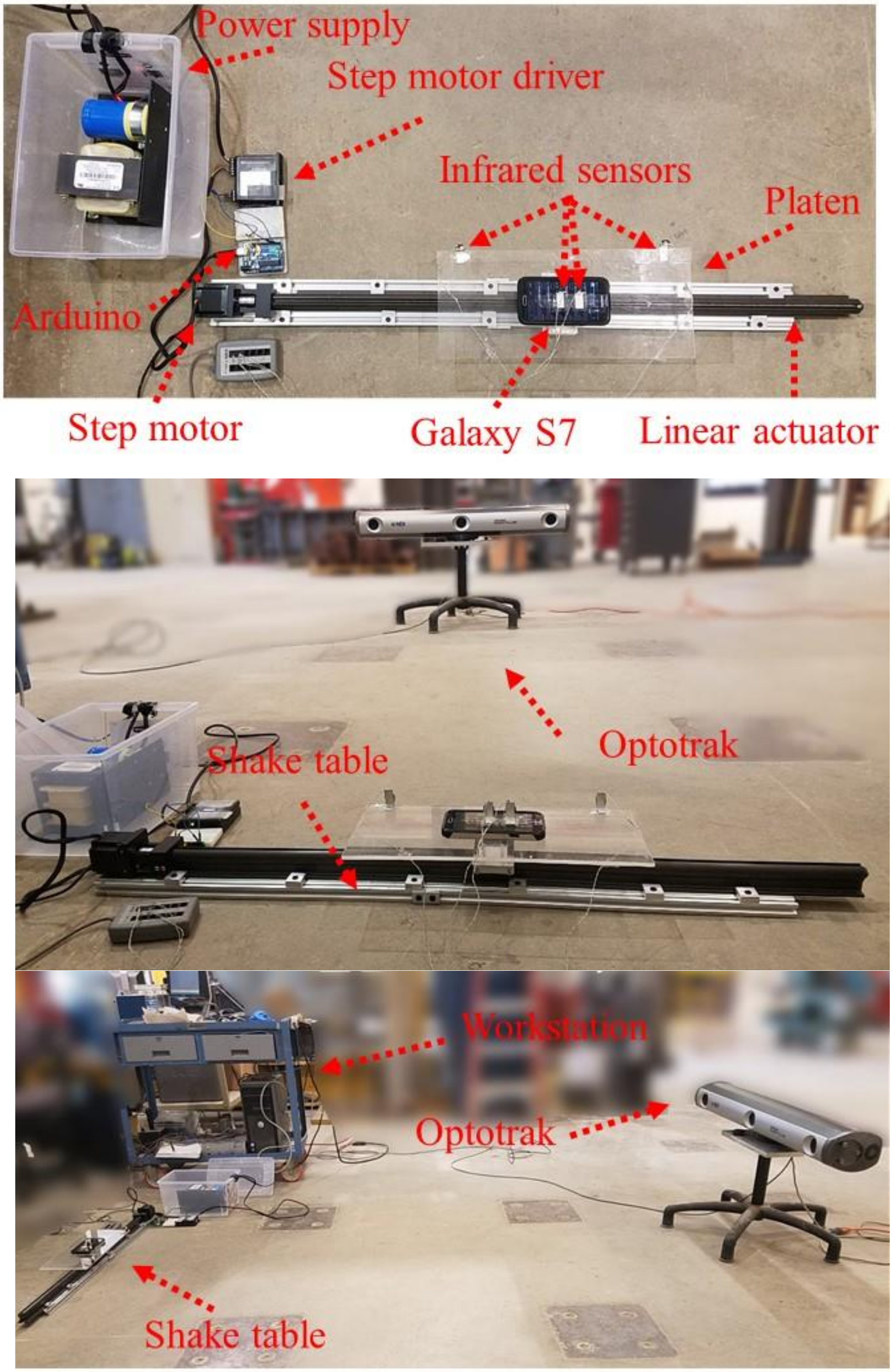


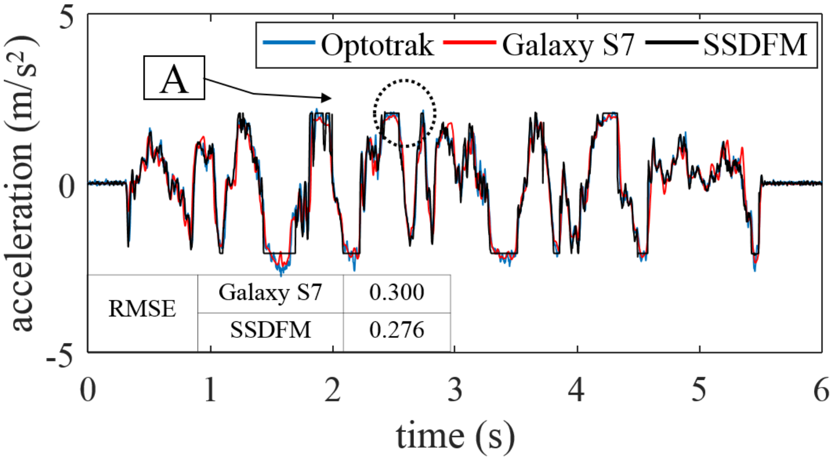
Figure 2-8. Experimental Setup

Two sets of tests are conducted, one with the smart device encased in a polyurethane protective case and another without a case. The phone case type is selected because it has a high coefficient of friction with the platen in order to simulate sticking-dominant response. The bare phone has a much lower coefficient of friction with the platen, which results in sliding-dominant motion. The platen of the shake table is made of acrylic. Together, both sets of tests represent a wide range of possible situations. Using standard procedures by ASTM D1894-14 (ASTM 2008), the measured static coefficient of friction between the shake table platen and polyurethane cover is 0.575 and the dynamic coefficient of friction is 0.384. The bare phone has a static coefficient of friction of 0.315 and a dynamic one of 0.210.

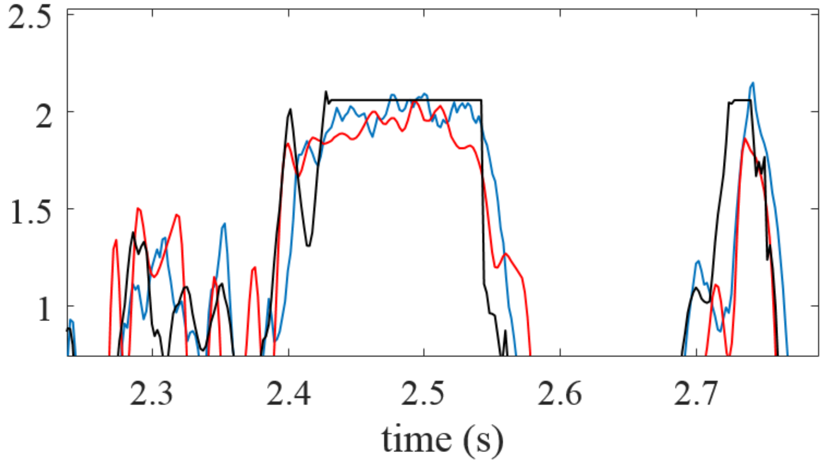
2.3.2 Validation of Proposed Model

Simulation results from the proposed SSDFM are compared with the experimental results (from Optotrak and Galaxy S7) in Figure 2-9a and Figure 2-9c. The input signals represent the strong motion portions of EQ2 and EQ3 in Table 2-3, accounting for 95% of the Arias Intensity. Because of minor surface imperfections in the platen, the measured acceleration during sliding motions is not a perfect plateau as computed in the numerical analysis as shown in Figure 2-9b and Figure 2-9d. Nevertheless the overall measured acceleration responses are quite close to the simulated acceleration responses.

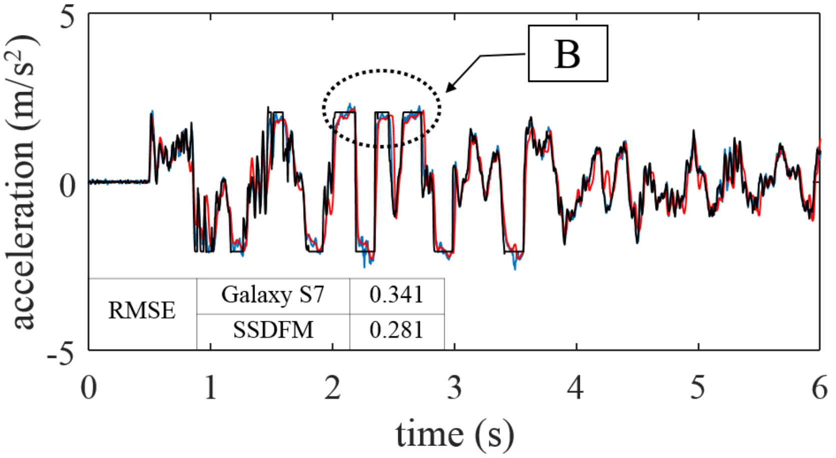
a)


Optotrak
Galaxy S7
SSDFM
A
acceleration (m/s^2)
time (s)
5
0
-5
0
1
2
3
4
5
6
RMSE
Galaxy S7
0.300
SSDFM
0.276

b)


2.5
2
1.5
1
2.3
2.4
2.5
2.6
2.7
time (s)

c)


B
acceleration (m/s^2)
time (s)
5
0
-5
0
1
2
3
4
5
6
RMSE
Galaxy S7
0.341
SSDFM
0.281

d)

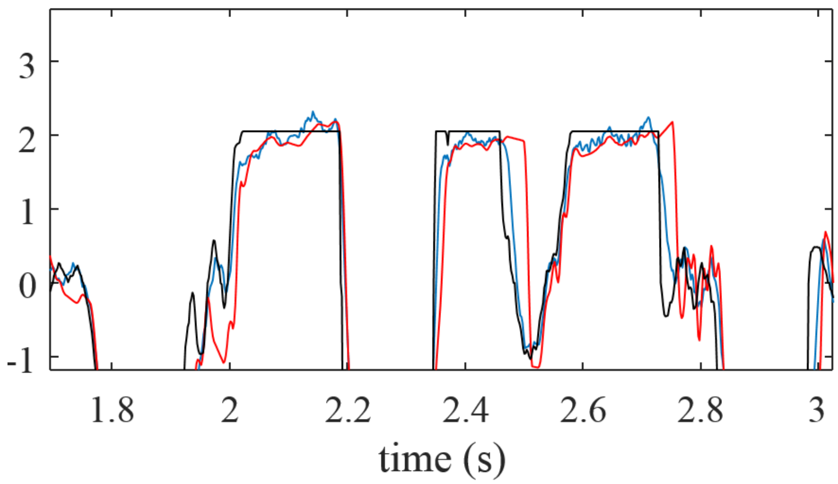


Figure 2-9. Comparison of RMSE of simulated and measured acceleration responses: a) EQ2; b) detail view of A in EQ2; c) EQ3; and d) Detail view of B in EQ3

Table 2-3. Ground motion records used

EQ ID	**PEER-NGA Record Information**		**PGA (g)**			**Arias Intensity (I_A) (m/s)**		
	Name	**Record No.**	***x*-dir**	***y*-dir**	***z*-dir**	***x*-dir**	***y*-dir**	***z*-dir**
EQ1	NORTHR/MUL	953	0.416	0.516	0.327	3.074	4.498	1.350
EQ2	NORTHR/LOS	960	0.410	0.482	0.318	1.913	1.976	0.532
EQ3	DUZCE/BOL	1602	0.728	0.822	0.203	3.724	2.431	0.485
EQ4	HECTOR/HEC	1787	0.266	0.337	0.150	0.831	1.866	0.353
EQ5	IMPVALL/H-DLT	169	0.238	0.351	0.145	2.398	3.290	0.538
EQ6	IMPVALL/H-E	174	0.364	0.380	0.140	3.918	3.227	0.511
EQ7	KOBE/NIS	1111	0.509	0.503	0.371	3.353	2.270	1.325
EQ8	KOBE/SHI	1116	0.243	0.212	0.059	0.827	0.639	0.059
EQ9	KOCAELI/DZC	1158	0.312	0.358	0.229	2.171	2.660	0.863
EQ10	KOCAELI/ARC	1148	0.219	0.150	0.086	0.578	0.434	0.223
EQ11	LANDERS/YER	900	0.245	0.152	0.136	0.231	0.169	0.113

The deviations betweeen the measured acceleration from Optotrak and that from the smart device and between the measured acceleration from Optotrak and the simulated one from SSDFM are calculated by the root mean square error (RMSE) method. For both EQ2 and EQ3 motions, the smart device's measurements have more fluctuations during sliding motions compared to the SSDFM such that its RMSE is slightly higher than that for SSDFM. For EQ2 motion, the RMSE for the smart device is 0.300 compared to 0.276 for SSDFM. For EQ3 motion, the RMSE for the smart device is 0.341 versus 0.2810 for SSDFM.

Comparison between the location and number of sliding points as computed from SSDFM and measured in the experiments is another way to validate the model. The device without protective shell is used in the following discussion because it has a pronounced sliding response. Sliding motion is commonly assumed to occur when the relative velocity between the base and the device is greater than the Stribeck velocity v_s (0.001m/s) (De Wit et al. 1995). Based on the 0.002 second time step used in this study and given the displacement accuracy achievable by the Optotrak system, velocity can be measured to about 0.01m/s.

Figure 2-10 illustrates the effect of the assumed v_s on the accuracy of state detection, i.e. sliding versus sticking. The correct detection rate is the ratio of correctly matched states to the total number of states. The erroneous detection rate represents the remaining points. Figure 2-10 shows that the rate of increase in correct detections flattens out at around 0.012 m/sec, which corresponds to the resolution of the test system. Hence this number is adopted in this work to define the interface between sticking and sliding. Figure 2-11 plots sliding points for EQ2 and EQ3 as computed from the Optotrak and SSDFM data. In Figure 2-11 sliding points are plotted as circles. It is clear that the measured data matches the computed data well. The high correct detection rate (94% for EQ2 and 97% for EQ3) signifies that the SSDFM is reaonably accurate and capable of modeling well the sticking and sliding regimes and the many transitions that occur between them.

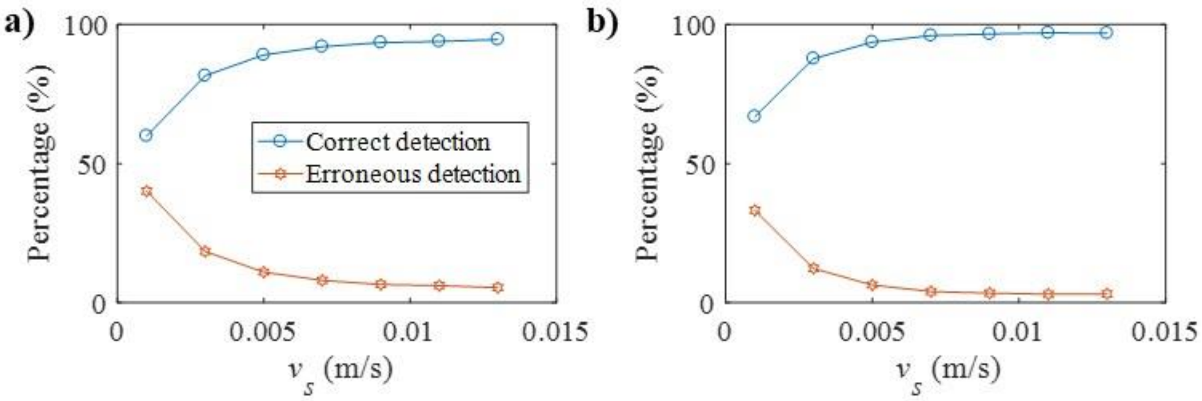
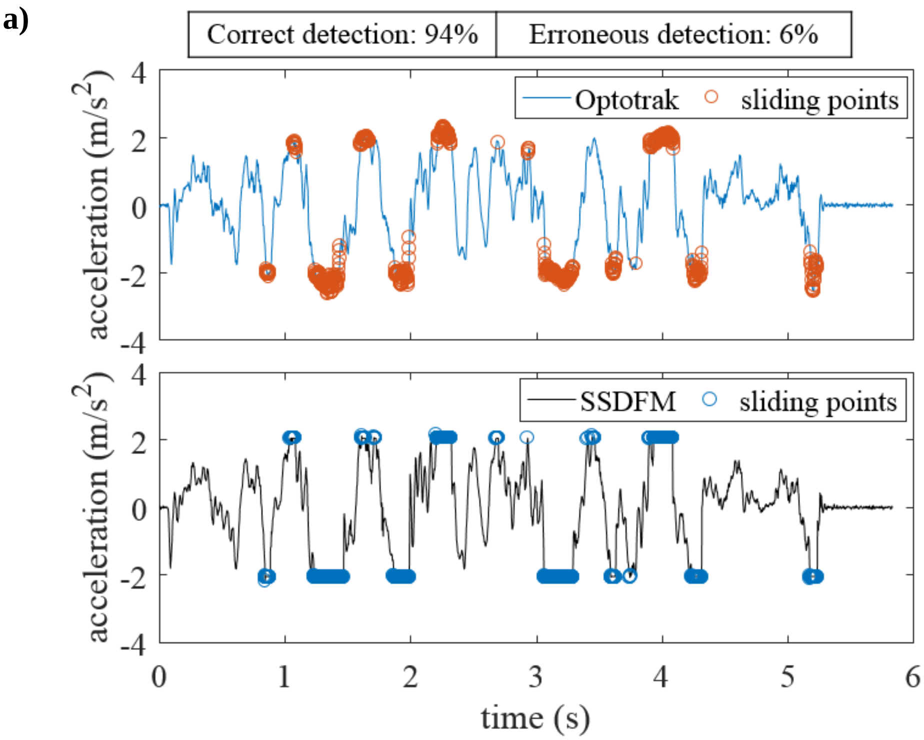
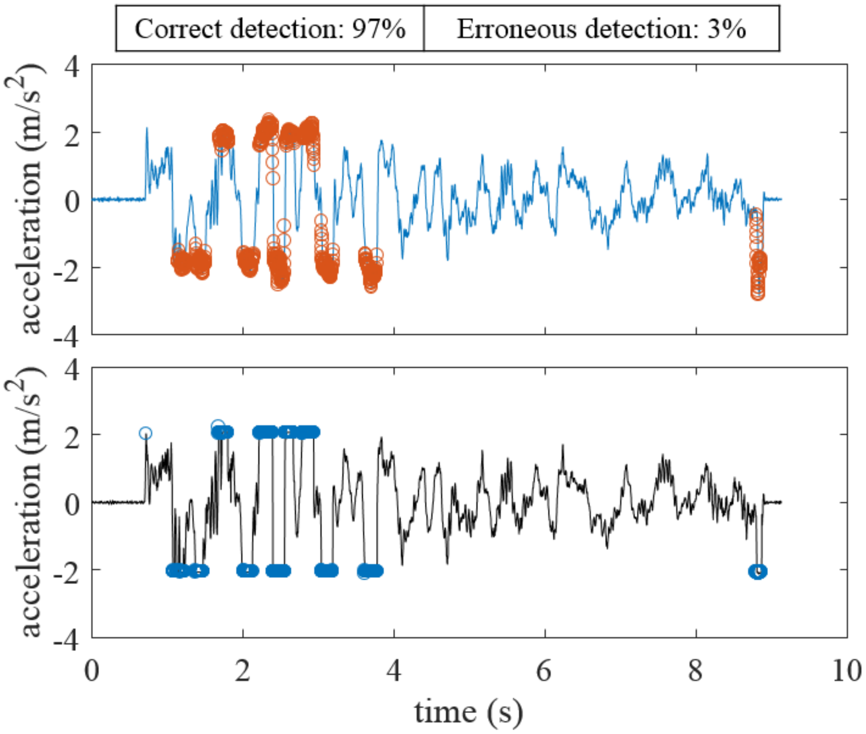


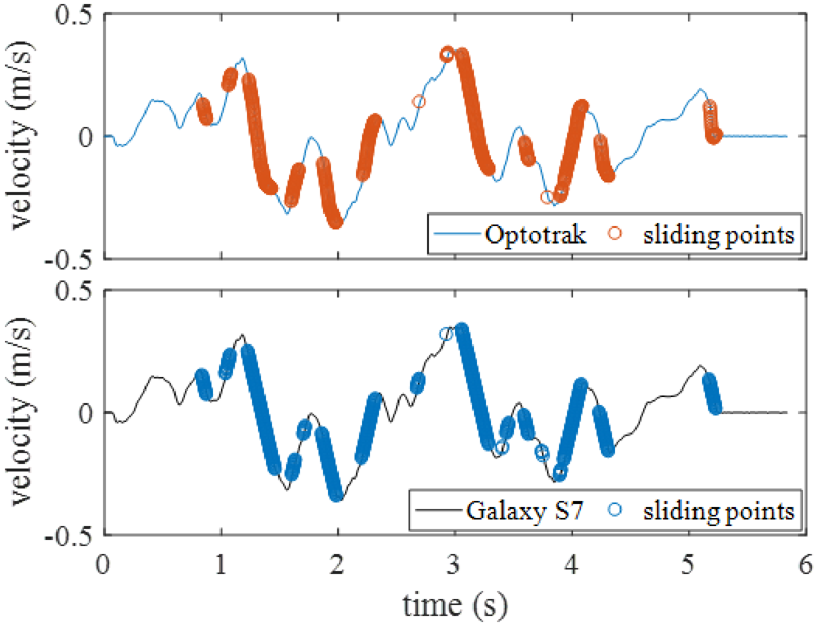
Figure 2-10. The effect of v_s on state detection for a) EQ2 and b) EQ3



b)


Correct detection: 97%
Erroneous detection: 3%
acceleration (m/s^2)
acceleration (m/s^2)
4
2
0
-2
-4
0
2
4
6
8
10
time (s)

c)


velocity (m/s)
velocity (m/s)
0.5
0
-0.5
Optotrak
sliding points
Galaxy S7
sliding points
0
1
2
3
4
5
6
time (s)

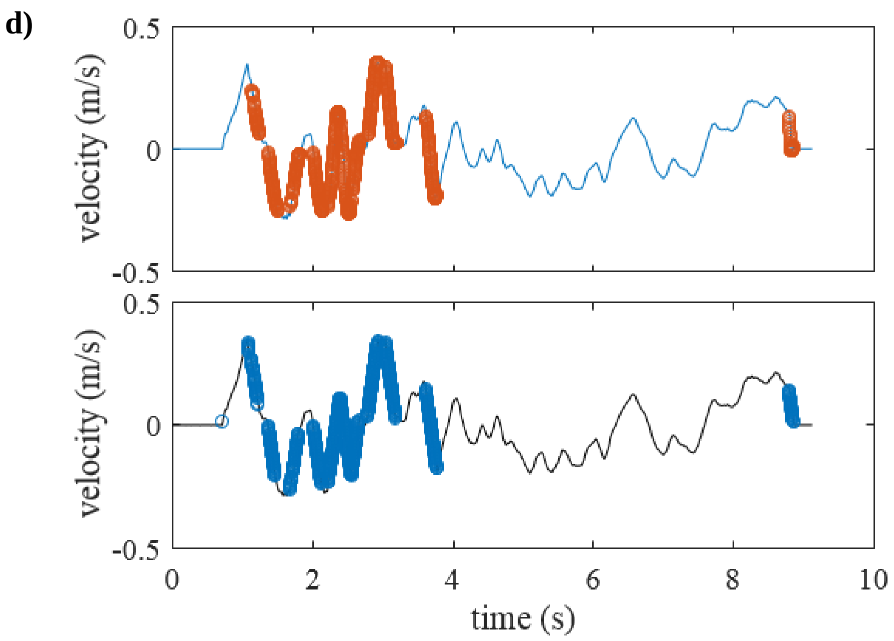


Figure 2-11. Comparison between measured and computed sliding points (v_s of 0.012 m/s): a) acceleration response for EQ2; b) acceleration response for EQ3; c) velocity response for EQ2; and d) velocity response for EQ3

2.4 Dynamic response of smart devices under earthquake ground motion

Using the SSDFM developed earlier, the idea of 'sliding potential' is used to assess the behaviour of a rigid block on a horizontal base subjected to two-dimensional motion and variable vertical acceleration. The intent is to identify under what conditions a smart device (block) will stick to the underlying surface under seismic action. The ground motions used in this study are from the Far-Field ground motion record sets in FEMA P-695 (FEMA 2009).

Table 2-3 lists their key characteristics, including peak ground acceleration (PGA) and the Arias Intensity (I_A) in two horizontal (x and y) and vertical (z) directions. The selected records are typically used by others for assessment of the probability of building collapse under the Maximum Considered Earthquake (MCE) as defined in ASCE/SEI 7-05 (ASCE 2006).

The critical acceleration is defined as the maximum acceleration the block can withstand without slipping. Based on this definition, the critical acceleration at which slip commences is:

$$A_c = |\mu_s g| \tag{2-16}$$

Whenever the magnitude of base acceleration,

$$\ddot{w} = \sqrt{\ddot{w}_x^2 + \ddot{w}_y^2} \tag{2-17}$$

exceeds A_c, sliding of the block initiates with respect to the base as mentioned in Equation 2-12.

2.4.1 The Effect of Horizontal Motion on the Critical Acceleration

Figure 2-12 shows the displacement response of the block in the horizontal plane when the underlying base is subjected to both components simultaneously of EQ1 and EQ2 and when A_c=0.1g. It is clear from Figure 2-12b that the displacement response of the block subjected to the EQ2 ground motion has strong directivity in the x direction. Figure 2-13 shows the acceleration, velocity and displacement responses in the x direction of EQ1 and EQ2 for A_c=0.1g . The x component of EQ1 has a PGA of 0.416g, which is more than four times A_c, yet it produces a peak slip of only 0.060 m, which is less than half the peak displacement of the base (0.131 m). On the other hand, the x component of EQ2, which also has a PGA about four times A_c (PGA =0.410g) has a peak sliding displacement of 0.226 m, which is about twice the peak displacement of the base (0.110 m). The main cause of this dichotomy is the relatively low-frequency content of EQ2 motion compared to EQ1, which allows the slipping action to build up for longer periods during EQ2.

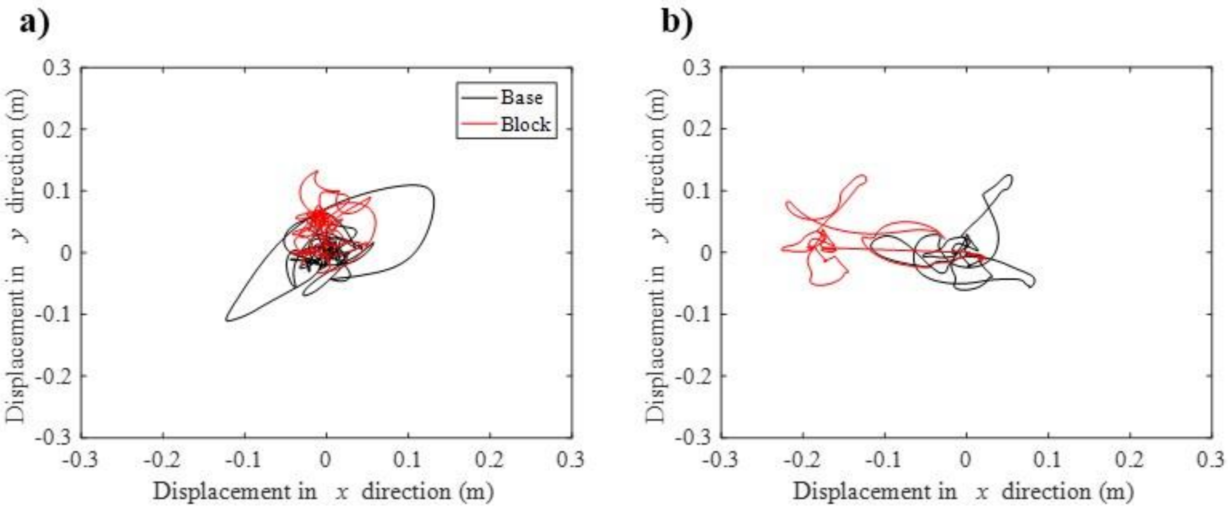
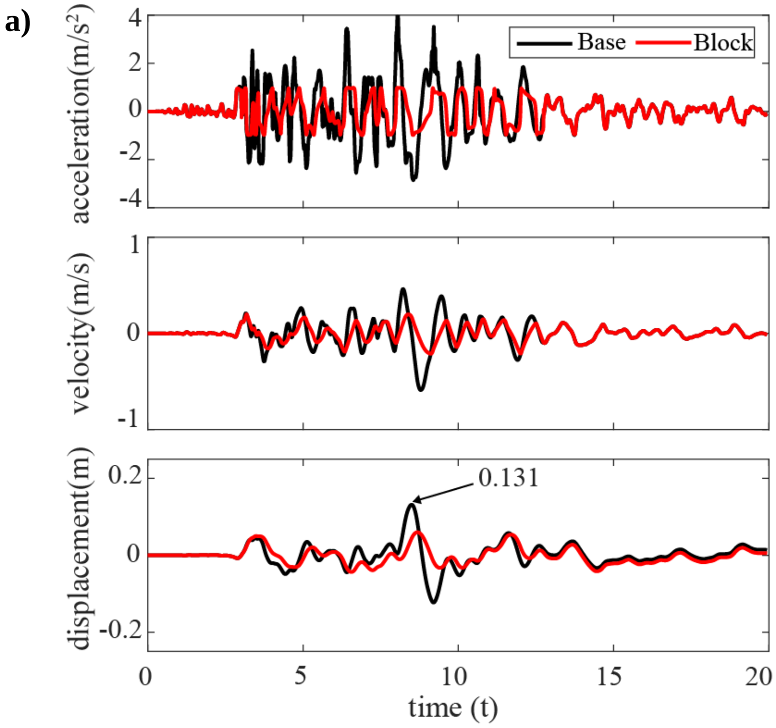


Figure 2-12. Sliding response induced by a) EQ1 and b) EQ2 when *Ac*=0.1*g*



b)

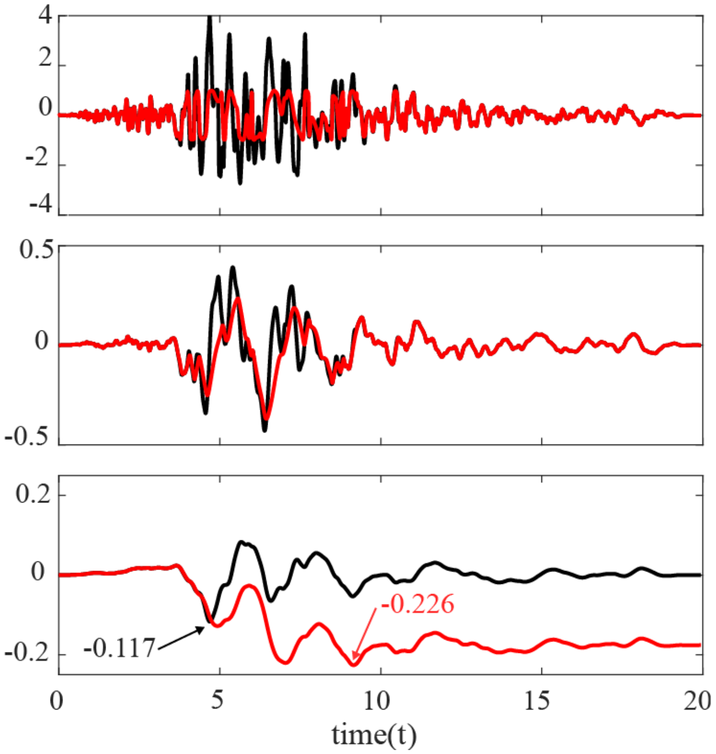


Figure 2-13. Sliding in the response induced by a) EQ1 and b) EQ2 in *x* direction when *Ac*=0.1*g*

Table 2-4 shows a summary of the maximum drift ratios of the block and base for the two horizontal components of each record under horizontal motion. The maximum drift ratio is defined as the largest ratio of the peak displacement of the block to that of the base. Most of maximum drift ratios associated with horizontal motion only are close to unity with some exceptions, e.g. EQ2 has a drift ratio of 1.933 in the x direction because of its high directivity in this particular direction.

Table 2-4. Summary of maximum drift ratio for horizontal only and horizontal and vertical motions

EQ ID	Maximum drift ratio				Difference	
	Horizontal ground motion		**Horizontal and vertical ground motion**			
	x-dir	**y-dir**	**x-dir**	**y-dir**	**x-dir**	**y-dir**
EQ1	0.456	1.197	0.45	1.308	1%	9%
EQ2	1.933	0.995	1.908	0.984	1%	1%
EQ3	0.934	1.383	0.937	1.392	0%	1%
EQ4	0.892	1.290	0.899	1.268	1%	2%
EQ5	1.014	1.014	1.010	0.995	0%	2%
EQ6	0.933	1.070	0.906	1.074	3%	0%
EQ7	0.993	0.912	0.905	0.941	9%	3%
EQ8	0.978	1.065	0.987	1.033	1%	3%
EQ9	0.875	0.988	0.878	0.957	0%	3%
EQ10	0.997	1.003	0.996	1.003	0%	0%
EQ11	1.139	1.015	1.139	1.011	0%	0%

2.4.2 Effect of Vertical Motion

To account for the effect of vertical motion, which typically occurs during earthquakes, the assumption that the normal load is a constant value is relaxed. This is achieved by replacing Equations 2-9, 2-10, and 2-12 by the following three equations, respectively:

$$\begin{bmatrix} F_{f,x}(t_n) \\ F_{f,y}(t_n) \end{bmatrix} = -\frac{\mu_k m(g - \ddot{w}_z(t_n))}{\sqrt{\left(\dot{X}(t_n)\right)^2 + \left(\dot{Y}(t_n)\right)^2}} \begin{bmatrix} \dot{X}(t_n) \\ \dot{Y}(t_n) \end{bmatrix} \qquad \text{2-18}$$

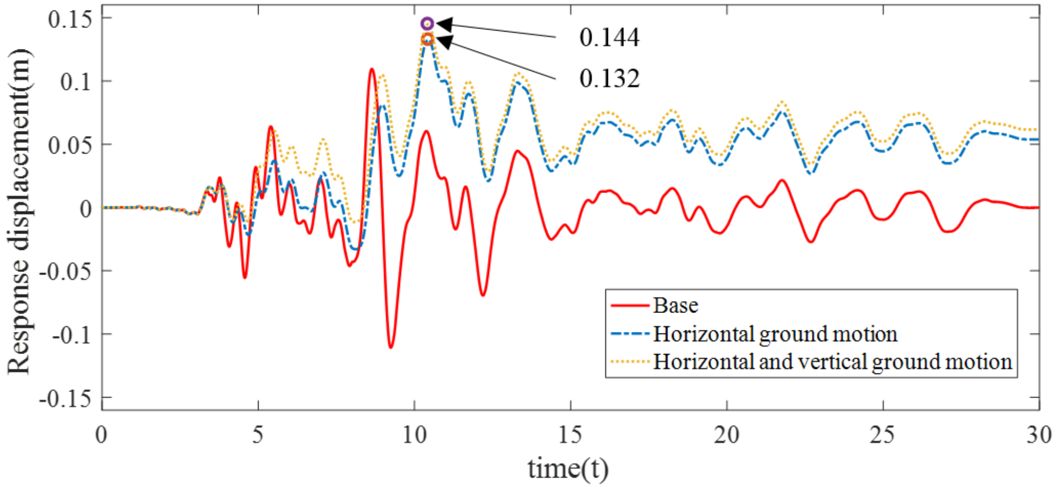
$$\begin{bmatrix} F_{f,x}(t_n) \\ F_{f,y}(t_n) \end{bmatrix} = \frac{\mu_k m\big(g - \ddot{w}_z(t_n)\big)}{\sqrt{\ddot{w}_x(t_n)^2 + \ddot{w}_y(t_n)^2}} \begin{bmatrix} \ddot{w}_x(t_n) \\ \ddot{w}_y(t_n) \end{bmatrix} \qquad \text{2-19}$$

$$F_s = m(g - \ddot{w}_z(t_n))\mu_s \tag{2-20}$$

where $\ddot{w}_z(t_n)$ is the vertical acceleration time history. For the sake of simplicity, it is still assumed that the block remains in contact with the base, i.e. no separation occurs, which is essentially true for all the ground motions consider herein because none of them had a vertical acceleration greater than the gravity acceleration.

The effect of vertical motion on the maximum drift ratio is summarized in Table 2-4. Figure 2-14 shows the displacement record of the two records with largest vertical peak accelerations: EQ1 and EQ7. It is clear from the Figure 2-14 that the effect of vertical motion is small. Figure 2-14 shows the same general trend. This can be seen by comparing the maximum drift ratios computed with and without vertical motion, where the maximum difference is generally within 2%, with the exception of a few cases, e.g. the y component of EQ1 and the x component of EQ7. Others have also observed this effect in sliding systems, e.g. in Gazetas et al. (2012) and Sarma and Scorer (2009).

a)



b)

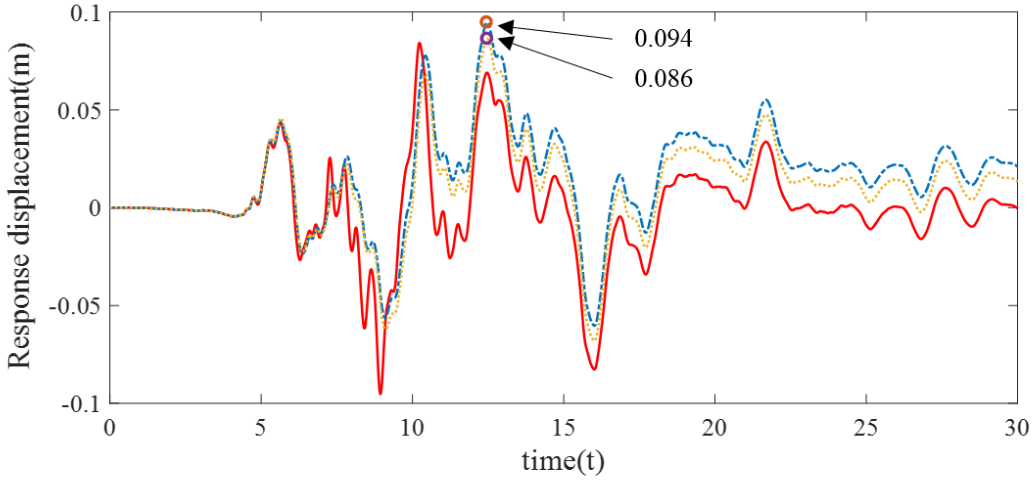


Figure 2-14. Displacement response of a block subjected to the two records with the highest peak vertical acceleration. a) EQ1 in *y* direction and b) EQ7 in *x* direction

2.4.3 Effect of Critical Acceleration

The effect of the critical acceleration is studied by varying A_c from 0.05g to 0.50 g with increments of 0.025. The spectra of the maximum relative velocity between the base and the block of all the horizontal ground motions are plotted in Figure 2-15. It can be seen from Figure 2-15 that the general trend is less slip with increasing A_c, becoming asymptotic with zero especially beyond A_c=0.3g, i.e. the block sticks to the base beyond this point. However, a number of obvious exceptions seem to occur, where the maximum relative velocity actually increases as A_c increases before eventually proceeding in the expected direction. The phenomenon of paradoxical increase of slippage with increasing critical acceleration A_c is labelled as ‘Safe Gulf Paradox (Gazetas et al. 2009).

The sticking critical acceleration is defined as the critical point where the maximum relative velocity is less than 0.001 m/s (the aforementioned Stribeck velocity). The intent of defining the sticking critical acceleration is to define the conditions under which the block sticks, from a practical perspective, to the base for a given earthquake intensity. These sticking A_c values are marked with a circle in Figure 2-15.

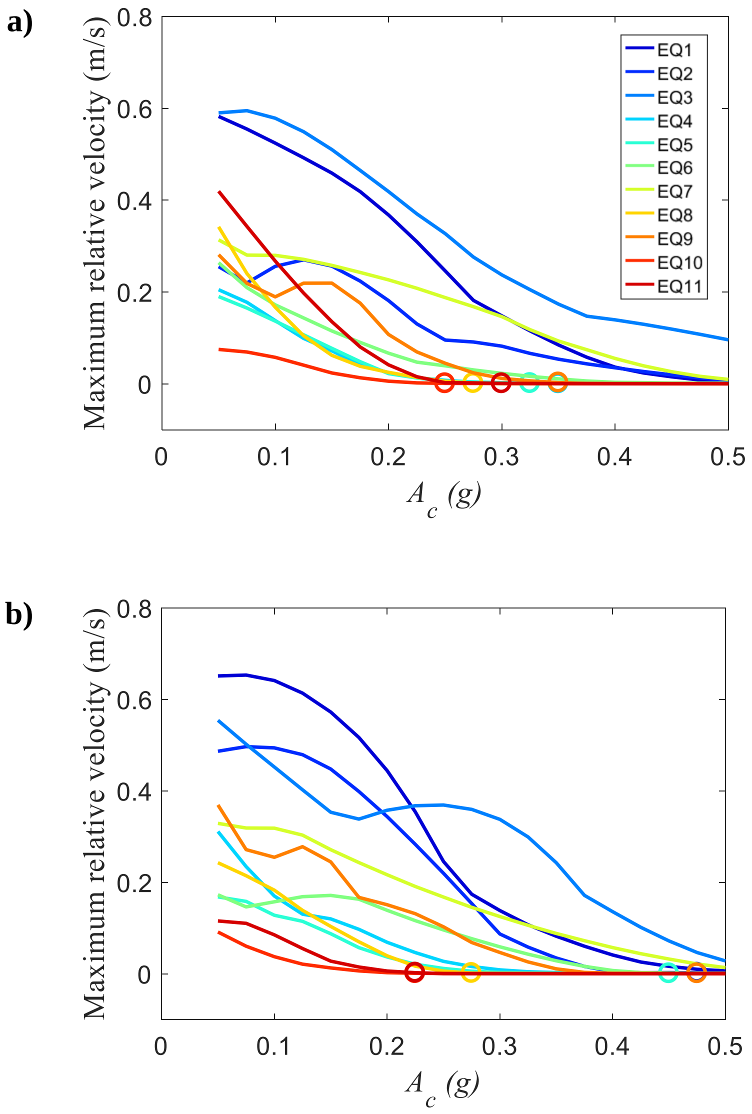


Figure 2-15. Sliding spectra of the block in terms of maximum relative velocity for all the horizontal ground motions with varying A_c. a) in x direction and b) in y direction

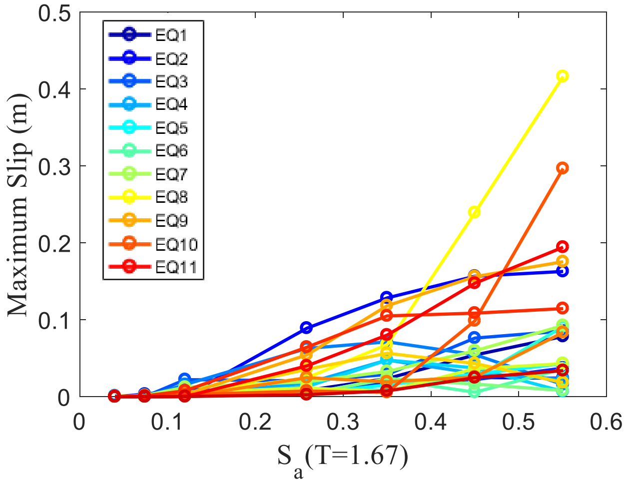
2.5 Probability of exceeding the slip limit

If a sturdy (rigid) desk in a building has a high coefficient of friction and sticks to the floor, e.g. it sits on a carpet that is firmly attached to the floor, its movements during an earthquake are representative of the motion of the underlying floor. The same idea can be extended to a smart device. If the device is in a protective shell made of a rubber-like polyurethane and there is a sufficiently high coefficient of friction between the shell and table to make it stick to the table during motion of the table, then the device's movements are represented of the table's movements, and therefore the underlying floor's movements. If, on the other hand, the coefficient of friction is not high enough or if the demand is too high, slip will likely occur and it becomes more challenging to assess the motion of the underlying floor from the device's readings. It is helpful to know the chance of this occurring.

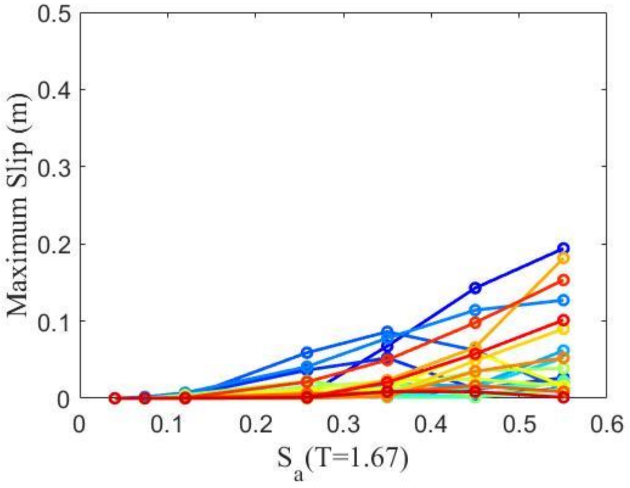
Consider a four-story steel frame building (Wu et al. 2018) with a first mode period of 1.67s with 5% damping located in Seattle with latitude (34.049) and longitude (-118.252). Using hazard curves from the United States Geological Survey (USGS), scale factors are computed by following the scaling method in FEMA P-695 (FEMA 2009) to generate new records that correspond to three hazard levels: 2% in 50 years, 10% in 50 years, and 50% in 50 years. The first period spectral accelerations corresponding to the three hazard levels are $0.55g$, $0.26g$, and $0.07g$, respectively. To enrich the computational space, four additional spectral accelerations, $0.04g$, $0.12g$, $0.35g$, and $0.45g$ (not tied to any specific hazard level) are used in the scaling scheme for a total of 7 scaled records for each earthquake. The scaled records are used to compute four maximum slip curves that correspond to critical acceleration values of A_c= $0.2g$, $0.3g$, $0.4g$, and $0.5g$. The sliding spectra of the block in maximum slip displacement between the base and the block for x and y components of all the horizontal ground motions are plotted in

Figure 2-16. It is can be seen from Figure 2-16 that the general trend is that, aside from cases that follow the ‘Safe Gulf Paradox’ (Gazetas et al. 2009), the maximum slip decreases with increasing A_c.

a)



b)



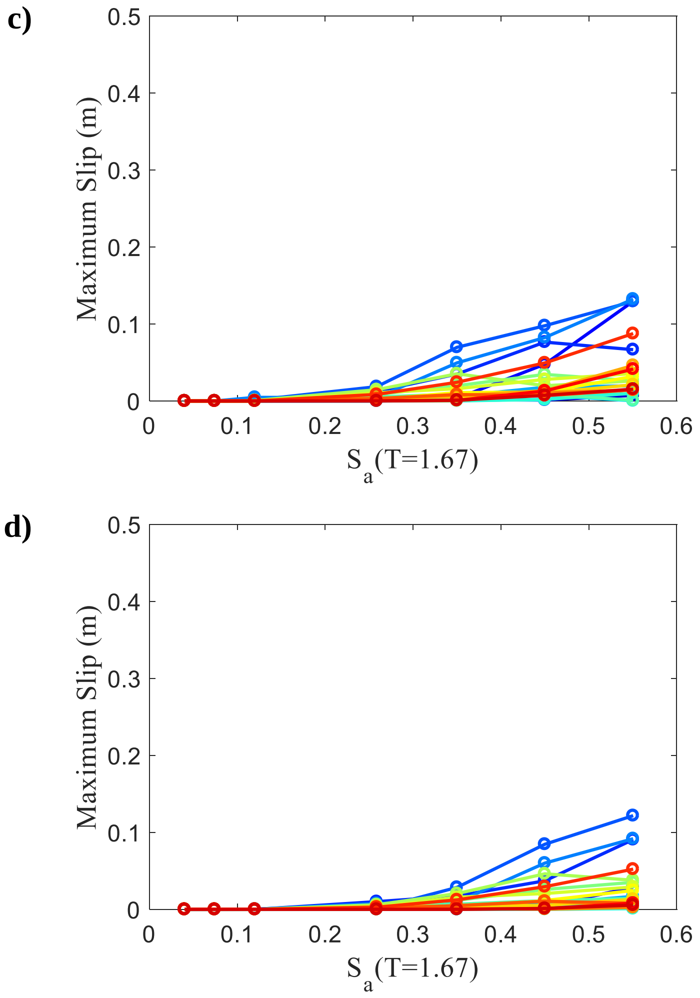


Figure 2-16. Sliding spectra of the block in terms of maximum slip for all the horizontal ground motions with a) A_c=0.2*g*, b) A_c=0.3*g*, c) A_c=0.4*g*, and d) A_c=0.5*g*

Based on the information in Figure 2-16, the idea of a "probability of exceeding the slip limit" (PESL) curve is introduced. The PESL curve is analogous to a fragility function. The slip limit, which is the difference between the absolute values of the peak displacements achieved by the block and underlying base, is selected to be 0.02 m. This value corresponds to 10% of the interstory drift associated with the collapse prevention limit state. The selection of 0.02m is arbitrary and can be tightened or relaxed depending on the accuracy sought. The PESL curve is obtained by using the FEMA P-58 Conditional Probability of Collapse Curve Fit Tool (FEMA P-

58 20012) and relates the probability of slip of an unconstrained block to the first period spectral acceleration. Table 2-5 summarizes the calculated probability of exceeding the slip limit for selected seven spectral accelerations with various critical accelerations. Figure 2-17 shows the fitted curve of probabilities for acceleration response spectrum with various value levels of A_c by using the curve fit tool. For example, for a 50% in 50 years event, there is no chance of exceeding the slip limit for all values of A_c. That grows to 36% for a 2% in 50 years event when A_c= $0.5g$. The data in Table 2-5 and Figure 2-17 suggests that a device with a coefficient of friction of 0.4 will be capable of reading the movement of the underlying surfaces with a maximum slip of less than 0.02m for events with 10% and 50% chances of occurrence in 50 years. This information (and the PESL curve, in general) can be useful for crowdsourcing damage assessment through smart devices after seismic events.

Table 2-5. Probability of exceeding the slip limit for selected spectral accelerations and critical accelerations

Sa(T=1.67s)	A_c=**0.2**g	A_c=**0.3**g	A_c=**0.4**g	A_c=**0.5**g	**Hazard Level**
0.04g	0%	0%	0%	0%	
0.07g	0%	0%	0%	0%	50% in 50 years
0.12g	9%	0%	0%	0%	
0.26g	45%	18%	0%	0%	10% in 50 years
0.35g	82%	36%	18%	18%	
0.45g	91%	55%	18%	18%	
0.55g	100%	64%	5%	36%	2% in 50 years

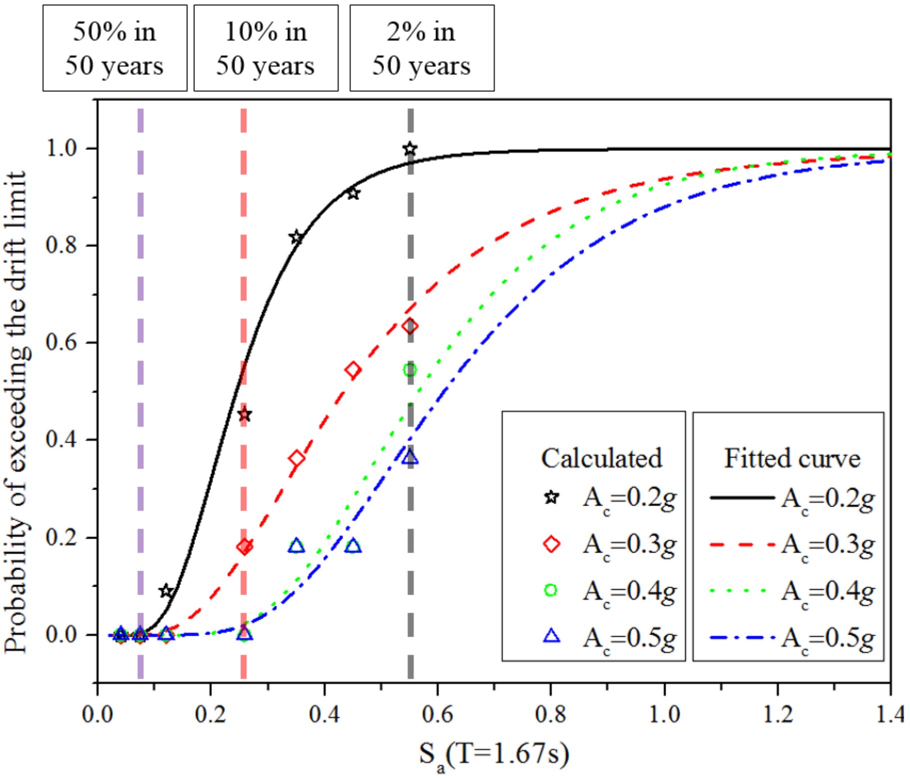


Figure 2-17. Probability of exceeding the slip limit as a function of the acceleration response spectrum for various critical accelerations

2.6 Summary and conclusion

The behavior of a smart device on an underlying surface subjected to seismic motion was investigated in this chapter. The smart device was modeled as a rigid block and its frictional interactions with the underlying base was represented using an existing model that was modified for the purposes of this research. The revised model used an interpolation technique to enhance detection of transition points, which must be accurately detected because of the frequency of their occurrence during seismic motion. The second modification entailed extending the model to handle vertical accelerations. The behavior of a device on (rigid) furniture, which itself is placed on a floor subjected to seismic motion (double stacked problem) was also discussed.

After validation of the model by comparing its behavior to experiments, the sliding/sticking behavior of a smart device on a base subjected to seismic motion was investigated. In particular, the sliding spectra associated with selected ground motions were presented. It was shown that vertical accelerations have a small effect on the sliding behavior of smart devices. The concept of a ‘probability of exceeding the slip limit curve’ was introduced and used to relate the probability of exceeding a given slip limit versus first period spectral acceleration for a given structure and location. Once generalized by taking into account other structures and locations, this information can be of value in future crowd-sourced, post-disaster reconnaissance efforts.

CHAPTER 3

Identifying stick-slip characteristics of a smart device

3.1 General

This chapter provides an analytical error model of an accelerometer in a smart device and the effects of sampling rates on smart devices. Then, a methodology is presented by which a smart device can decide if its motion is representative of the motion of the floor underneath or whether it is tainted by excessive sliding action. This method is based on the estimation of the kinetic coefficient of friction between the smart device and the underlying surface based on wavelet transforms. The noise associated with a smart device’s measurement of acceleration is established and noise reduction methods to overcome them are compared. Computational simulation results and experimental data are used to demonstrate the concepts discussed in this chapter.

3.2 Literature review

Accelerometers in smart devices are imperfect sensors with performance variations among devices of a given model as well as between models. The sensor errors can be classified into two types: deterministic (systematic) and stochastic (random) (Nassar 2003). While major deterministic error sources include bias and scale errors, which can be removed by specific calibration procedures (Park and Gao 2006), stochastic error estimation necessitates a more complex process. As such, stochastic errors are not addressed in this chapter, although they can be reduced in future efforts by using methods such as the zero velocity update method

(Noureldin et al. 2012) as discussed later in Chapter 5. An analytical study of the effects of biases in the accelerometer data and the effects of changing the cut-off frequency in anti-aliasing filters was conducted by Thong et al. (2002).

Sliding motion may contaminate the recorded data making it impossible to estimate the motion of the underlying floor from the device’s measurements. In order to detect floor acceleration, a smart device must move in tandem with the floor. However, smart devices are usually just placed on furniture and not constrained in any way. If a smart device has an adequate coefficient of friction with the underlying surface (e.g. encased in a rubber protective cover) and the underlying furniture is attached firmly to the floor, the device’s movements will be representative of the floor’s movements. Otherwise, slip occurs. It is therefore necessary to identify the parts of the record during which slip occurs so as to exclude that from a device’s readings of the shaking record.

As seen in the simulation results in Figure 2-9, acceleration readings from a sliding device exhibit plateau-like step function. The presence of these plateaus can be used to help distinguish between sliding and sticking motion. Edge or step detection is a fundamental problem in many areas of signal and image processing. Gradient operators, such as Roberts, Prewitt, and Sobel, are classical methods for detecting edges. These operators can be improved by incorporating smoothing procedures to eliminate noise, i.e. the so-called filtered derivative methods, and are widely used due to their simplicity. However, there are trade-offs between the level of smoothing and the variance of the estimated edge location, which can be resolved by adopting a multi-scale analysis through wavelet analysis (Mallat and Hwang 1992). Wavelet analysis has been used in various fields. Sadler and Swami (1999) detected steps using a multiscale wavelet analysis based on the ability of a certain discrete wavelet transform (DWT) to characterize signal steps and edges. Sun and Chang (2002) investigated the use of the wavelet packet transform and a neural network model for damage assessment of civil engineering structures. Zarei and Poshtan (2007) developed a diagnostic method employing Wavelet Packet Analysis for the detection of incipient bearing failures. Sadler et al. (1998) studied detection and estimation of aeroacoustics shock waves generated by supersonic projectiles with wavelet transform. None of these studies, however, and other similar ones in the literature, proposed methods by which to detect sliding regimes as done herein. As will be shown, detecting edges in clean computational data is feasible. However, noisy experimental data presents new challenges.

3.3 Error characteristics of smartphone accelerometers

It is necessary to develop an understanding of the errors and limitation of sampling rate in smart device accelerometers. This section addresses two challenges: 1) an analytical error model for MEMS-based accelerometers that are ubiquitous in smart devices is adopted in an attempt to understand the errors that could potentially contaminate their measurements and 2) an investigation of the sampling rate needed to minimize displacement measurement errors.

3.3.1 Errors in smart device accelerometers

Current commercially available accelerometers have a variety of error sources, which are slightly different depending upon the manufacturing processes employed. The error equation of a conventional mechanical inertial sensor for acceleration measurement along the x-axis ($\tilde{a}_x$) can be expressed as (Titterton et al. 2004):

$$\tilde{a}_x = (1 + S_x + \delta S_x)a_x + M_y a_y + M_z a_z + B_x + \delta B_x + B_v a_x a_y + n_x \tag{3-1}$$

where a_x is the x-acceleration (along the x-axis), a_y and a_z are the accelerations applied perpendicular to the x-axis, S_x is the scale factor error, usually expressed in polynomial form to include non-linear effects, δS_x is residual scale factor error, M_y and M_z are the cross-axis coupling factors, B_x is the measurement zero-offset bias, δB_x is residual bias, B_v is the vibro-pendulous error coefficient, and n_x is the random noise.

Accelerometers based on micro-electro-mechanical (MEMS) technology have insignificant cross-axis coupling factors. Hence, Equation 3-1 can be simplified as follows (Allen et al. 1998):

$$\tilde{a}_x = (1 + S_x + \delta S_x)a_x + B_x + \delta B_x + n_x \qquad 3\text{-}2$$

In this work, deterministic errors are estimated by using the static multi-position method (El-Diasty and Pagiatakis 2008). All biases, scale factor errors and non-orthogonality errors for the axes *x*, *y* and *z* are estimated using measurements from multiple at-rest position configurations (six positions, as discussed next). The idea is to first take an average of Equation 3-2, where residual errors of scale factor (δS_x), residual bias (δB_x), and random noise (n_x) are eliminated because their expected values (means) are zeros.

$$A(\tilde{a}_x) = (1 + S_x)A(a_x) + B_x \qquad 3\text{-}3$$

where $A(\cdot)$ is the average of the data. Consider the at-rest situation where the device is placed along the *x*-axis. Gravity acts on the device in the negative direction (-g) of the *x*-axis when the device is placed pointing along the *x*-axis and vice versa. Equation 3-3 can be extended in the *x*, *y* and *z* directions as Equation 3-4. In this case, the superscript of acceleration as *Xu* in Equation

3-4 is denoted. In the opposite direction, the superscript of acceleration as Xd in Equation 3-5 is denoted.

$$\begin{bmatrix} A(\tilde{a}^{Xu})_x \\ A(\tilde{a}^{Xu})_y \\ A(\tilde{a}^{Xu})_z \end{bmatrix} = \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \right] \begin{bmatrix} -g \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \tag{3-4}$$

$$\begin{bmatrix} A(\tilde{a}^{Xd})_x \\ A(\tilde{a}^{Xd})_y \\ A(\tilde{a}^{Xd})_z \end{bmatrix} = \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \right] \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \tag{3-5}$$

where S_{ij} are the scale factor of i^{th} axis under j^{th}axis as a chosen principal axis. The diagonal elements of S_{ij} represent the scale factors along the three axes, whereas the other elements of S_{ij} are the cross-axis factors. The experiment is repeated in the same way for the *y* and *z* directions. The collection of the above six observation equations provide the following single equation in matrix form:

$$W = EC \tag{3-6}$$

where

$$W = \begin{bmatrix} A(\tilde{a}^{Xu})_x + g & A(\tilde{a}^{Xd})_x - g & A(\tilde{a}^{Yu})_x & A(\tilde{a}^{Yd})_x & A(\tilde{a}^{Zu})_x & A(\tilde{a}^{Zd})_x \\ A(\tilde{a}^{Xu})_y & A(\tilde{a}^{Xd})_y & A(\tilde{a}^{Yu})_y + g & A(\tilde{a}^{Yd})_y - g & A(\tilde{a}^{Zu})_y & A(\tilde{a}^{Zd})_y \\ A(\tilde{a}^{Xu})_z & A(\tilde{a}^{Xd})_z & A(\tilde{a}^{Yu})_z & A(\tilde{a}^{Yd})_z & A(\tilde{a}^{Zu})_z + g & A(\tilde{a}^{Zd})_z - g \end{bmatrix} \tag{3-7}$$

$$E = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} & B_x \\ S_{yx} & S_{yy} & S_{yz} & B_y \\ S_{zx} & S_{zy} & S_{zz} & B_z \end{bmatrix} \tag{3-8}$$

$$C = \begin{bmatrix} -g & g & 0 & 0 & 0 & 0 \\ 0 & 0 & -g & g & 0 & 0 \\ 0 & 0 & 0 & 0 & -g & g \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad 3\text{-}9$$

The calibration parameters are computed by using the Least Square method as follows:

$$\hat{E} = (WPC^T)(CPC^T)^{-1} \quad 3\text{-}10$$

where $P = \sigma_0^2\Sigma^{-1}$, σ_0^2 is the a *priori* variance factor (usually $\sigma_o^2 = 1$) and Σ is the sample variance-covariance matrix comprising the sample variances of the accelerometer average from the six-position test in the diagonal and zeros in the non-diagonal elements.

The result of an experiment with a Samsung Galaxy S5 is shown in Equation 3-11. It is important to note that its selection does not affect the generality of the results presented hereafter. In addition, it should be noted that for low-cost inertial sensors, the bias and scale errors are temperature-dependent as indicated by El-Diasty et al. (2007). These experiments were conducted in an indoor laboratory at 17 - 23 ℃ which has negligible effects on acceleration.

$$E = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} & B_x \\ S_{yx} & S_{yy} & S_{yz} & B_y \\ S_{zx} & S_{zy} & S_{zz} & B_z \end{bmatrix} = \begin{bmatrix} 0.015 & -0.004 & 0.000 & -0.102 \\ -0.05 & 0.017 & 0.007 & 0.166 \\ -0.011 & 0.005 & 0.011 & 0.145 \end{bmatrix} \quad 3\text{-}11$$

This result can be directly applied to Equation 3-2 to obtain the calibrated acceleration, $\hat{a}_x$, which still includes the effects of stochastic errors:

$$\hat{a}_x = (\tilde{a}_x - B_x)/(1 + S_{xx}) \quad 3\text{-}12$$

Table 3-1 shows that the average of standard deviation (σ) of the calibrated acceleration data from six positions of stationary accelerometer is reduced insignificantly in comparison with the measured acceleration data. This implies the MEMS-based accelerometer in smart devices have negligible errors due to bias and scale factor.

Table 3-1. Results of calibration of stationary accelerometer

	***x* direction**	***y* direction**	***z* direction**
$\sigma(\tilde{a}_x)$ **Measured data**	0.0218	0.0214	0.0327
$\boldsymbol{\sigma(\hat{a}_x)}$ **Calibrated data**	0.0215	0.0210	0.0323

3.3.2 The effects of sampling rates on smart devices

Besides the errors in accelerometers described in the previous section, the relatively low sampling rates in current accelerometers can cause additional problems. To understand the limitations of commercially avavilable smart devices, Thong et al. (2002) investigated how errors in position sensed by means of accelerometers vary with time and sampling frequency.

Following the approach in Thong et al. (2002), a study of a Samsung Galaxy S5 is conducted to investigate errors in the measurement of position attributed to noise in accelerometer data. Readings from the device are taken while it is stationary to ensure that any errors computed from its output are due to noise only and not any actual device motion. Errors in the estimated displacement that arise from double integration of noise are used to compute the variation of the root mean square (RMS) error in position with change in the sampling frequency for a fixed integration time, taken as 3.0 seconds. Three cut-off frequencies for the low pass filter are considered: 50, 100, and 200 Hz.

As shown in Figure 3-1, the smaller the cut-off frequency of the filter, the smaller the RMS errors in position with change in the sampling frequency. This can be explained qualitatively by noting that the lower the cut-off frequency, the more significant the correlation between the samples of noise. The results in Figure 3-1 demonstrate that the RMS errors decrease asymptotically as the sampling frequencies increase. It appears that increasing the sampling frequency beyond 1000 Hz for 50, 100 and 200 Hz cut-off frequencies does not result in a significant reduction in the RMS errors. This illustrates that commercially available smart devices that commonly have a maximum sampling rate of 200 Hz are presently not good enough to minimize error in displacement computations associated with noise.

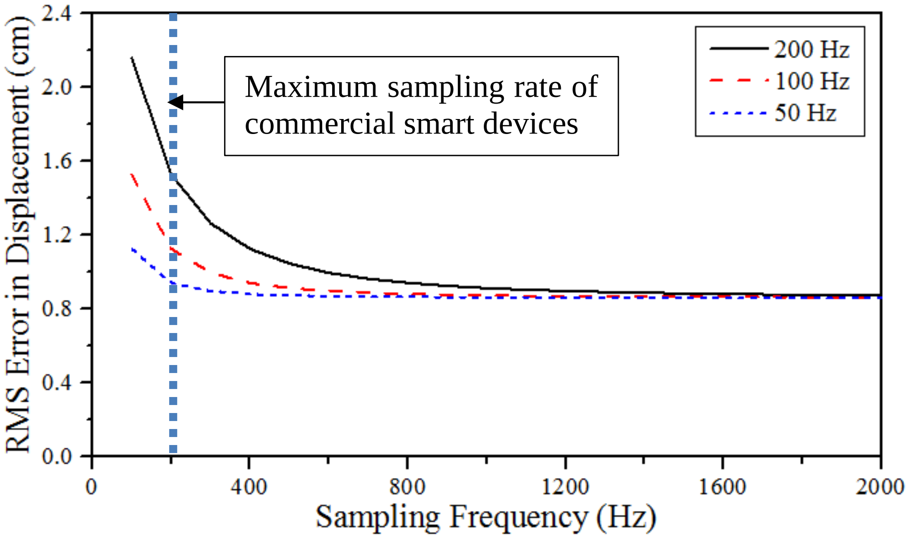


Figure 3-1. RMS position errors as a function of sampling frequency for various filter cut-off frequencies applied to the Samsung Galaxy S5

3.4 Computed behavior of a non-constrained smart device on a moving surface

Assume that a smart device can be adequately approximated as a rigid solid block. The block sits on a rigid surface subjected to transitional motion in one-dimension as shown in Figure 2-1. The equation of motion in the *x* direction is:

$$m\ddot{u}_x(t_n) = F_{f,x}(t_n) \tag{3-13}$$

where m is the mass of the device, $\ddot{u}_x(t_n)$ is the acceleration of the device in the *x* direction at time, t_n, and $F_{f,x}(t_n)$ is the friction force acting between the base and the device in the *x* direction. The solution of Equation 3-13 depends on the state of motion (sliding or sticking).

The Coulomb law of dry friction is widely employed in contact problems because of its simplicity. In this Chapter, we assume that the friction force is of the Coulomb type. The friction force acts opposite to the direction of motion of the device with respect to the moving base. It is further assumed that the weight of the device, N_o, remains constant, i.e. no vertical acceleration occurs, and that the acceleration of gravity, g, is constant. Under these conditions, the Coulomb friction force during sliding is simply equal to the normal load multiplied by a KCOF, μ_k, between the device and the underlying base as follows:

$$F_{f,x}(t_n) = N_o\mu_k = mg\mu_k \tag{3-14}$$

When sliding initiates, the acceleration of the device divided by the gravity acceleration becomes equal to μ_k. This can be seen by substituting Equation 3-14 into Equation 3-13 , as follows:

$$\mu_k = \ddot{u}_x(t_n)/g \tag{3-15}$$

Extending this idea into two-dimension, the KCOF is:

$$\mu_k = \sqrt{\left(\ddot{u}_x(t_n)\right)^2 + \left(\ddot{u}_y(t_n)\right)^2}/g \tag{3-16}$$

where $\ddot{u}_y(t_n)$ is the acceleration of the device in the y direction at time, t_n. In this chapter, only transitional motion in one-dimension is considered for the sake of simplicity. In section 2.3.2, more general cases are discussed, including those that account for the static coefficient of friction (SCOF).

3.5 Computational and experimental simulation of sticking and sliding

Experiments and associated computations are conducted to investigate a smart device’s sticking and sliding behaviour on a shaking surface. The subject of the research in this chapter is an Android smart phone SM-G930 (Samsung Galaxy S7).

The selected device has a smooth aluminum body. A protective cover is also considered in the research for two reasons: 1) such covers are widely used by people to protect their devices and therefore represents a common situation, and 2) the presence of a cover changes the friction properties of the device and hence its interaction with the underlying surface. The smooth body of the device itself promotes a sliding-dominant response. To explore the other extreme in behavior, a protective shell with relatively high coefficient of friction is selected to promote sticking-dominant behavior. Figure 3-2 shows the selected device and its protective cover.

The friction coefficients between the device, with and without its protective shell, and the shaking surface (described in section 2.5.1) is measured using standard procedures (ASTM 2008). The measured KCOF is 0.164 for the device without its shell and the SCOF is 0.216. With the protective shell, the KCOF is 0.273 and the SCOF is 0.362. The SCOF is known to be greater than the KCOF for the same set of interacting surfaces (Sheppard and Tongue 2005, Meriam et al. 2008).

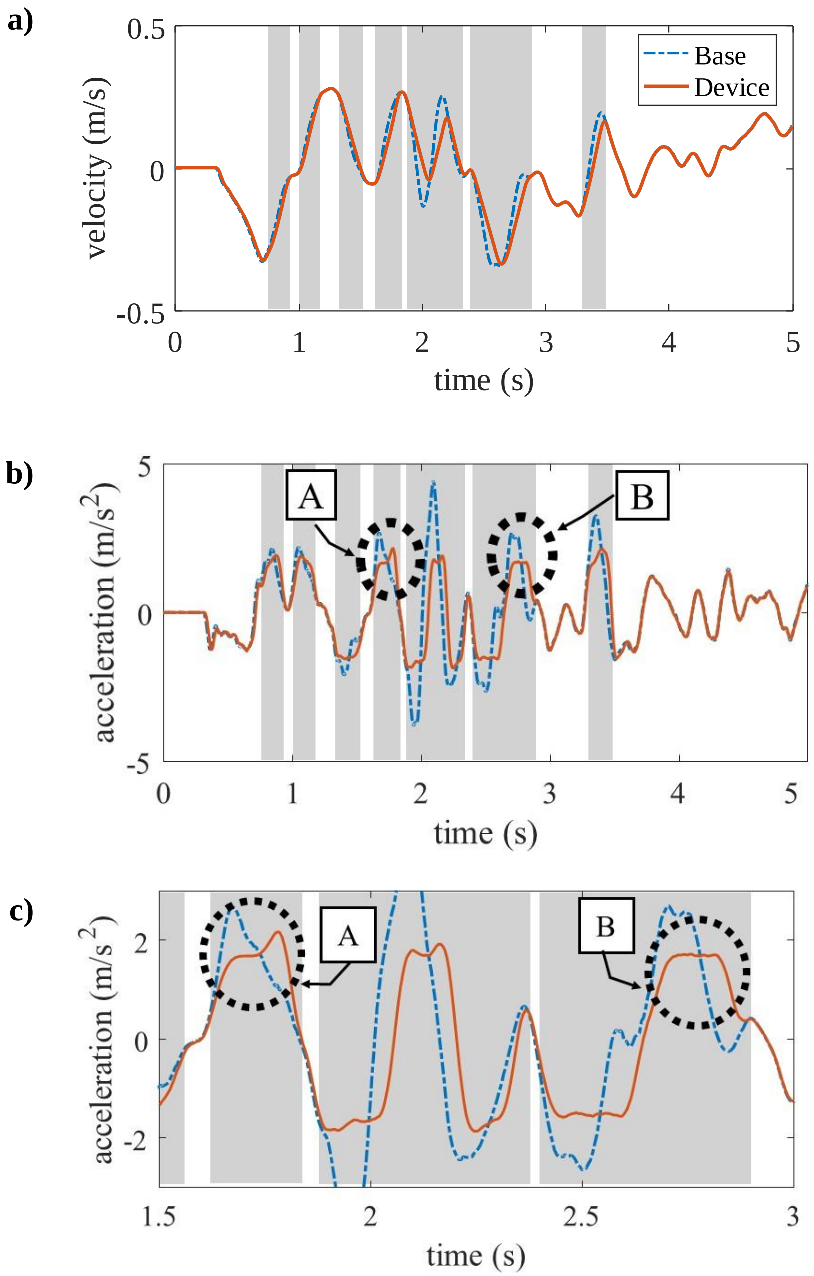
(a) front side (b) back side (c) a protective shell

Figure 3-2. Studied Galaxy S7 and its base

A ‘Sensor Kinetics Pro’ Android application by Innoventions, Inc. is used to record the acceleration data. An infrared 3D-motion capture system (Optotrak) is used to capture the true motion of the smart device. The tracking system can measure the 3D positions of a set of markers with an accuracy of 0.1 mm and resolution of 0.01mm as long as the markers are placed within a 3m radius from the detector. The markers are connected to the recording system via thin wires. The wires are arranged carefully to prevent them from hindering or influencing the motion of the phone. The velocity and acceleration responses are computed by taking the first and second derivatives, respectively, of the displacement measured by the Optotrak system.

The platen is subjected to a given seismic history. Figure 3-3a, b and c show the measured velocity and acceleration responses of the device and platen for a seismic signal. Figure 3-3d, e and f show the behavior as computed using Equation 3-13 through 3-15 wherein the measured SCOF and KCOF are used. Shaded areas represent sliding motion and other areas represent sticking motion in Figure 3-3. Generally, sliding motion is assumed to exist when the relative velocity between the base and the device exceeds the Stribeck velocity (De Wit et al. 1995). This small relative velocity represents the transition limit between static and dynamic behaviors. Given the displacement resolution of the Optotrak system used (± 0.01 mm) and it's 500Hz sampling rate, the Stribeck limit is taken to be 0.01m/s. When sliding occurs, i.e. when the relative velocity between the device and platen exceeds the Stribeck value, the velocity versus time response is linear as shown in Figure 3-3a and d.

In regions C and D in Figure 3-3e, the transition between sliding and sticking is sharp in the computed acceleration data. Singularities (points at which a function does not possess a derivative) can be observed at these transition points. In signal and image processing field, these singularities are called edges and are the basis of commonly used edge detection methods as discussed earlier. In regions A and B in Figure 3-3b, however, transitions between sliding and sticking motions in the experimental acceleration data are rounded. Also, the sliding regime has significant chattering and is not as smooth as the computed data, where the behavior is a plateau with a constant value equal to KCOF. The contrast can be seen in more detail in Figure 3-3c and f. The rounded transitions and uneven sliding response in the experimental data hinder differentiating between sliding and sticking motions and complicate finding the KCOF. Given this difficulty, the remainder of the Chapter is devoted to finding means by which to estimate an appropriate KCOF value from measured data.



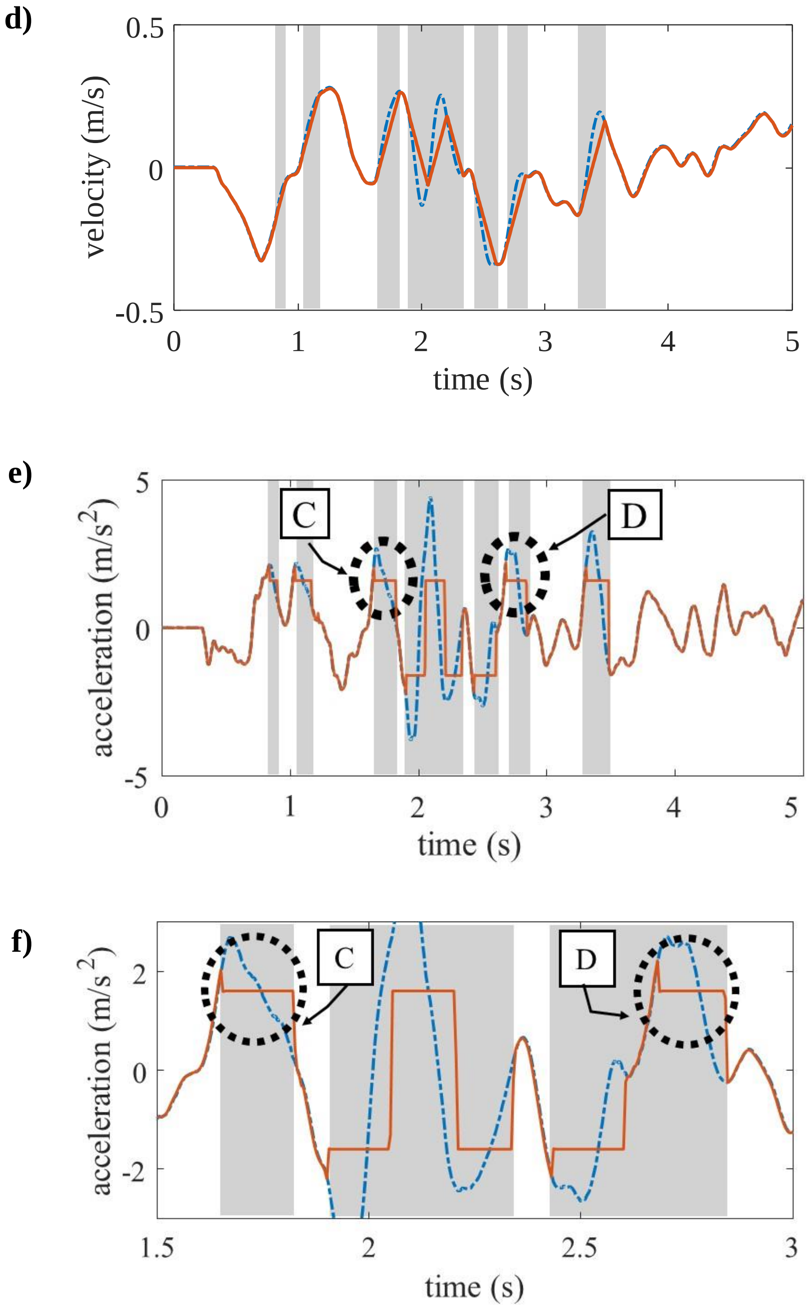


Figure 3-3. Dynamic responses of the device on the moving base: a) Measured velocity; b) Measured acceleration; c) Detailed view of regions A and B; d) Simulated velocity; e) Simulated acceleration; and f) Detailed view of regions C and D. Shaded areas represent sliding motion and other areas represent sticking motion

3.6 Identification of KCOF

As noted earlier, the acceleration during the sliding regime is theoretically constant. It is denoted as the ‘slip acceleration’ and equal to the KCOF multiplied by the gravity acceleration. From Equations 3-15 and 3-16:

$$A_s = \left|\mu_k g\right| \qquad 3\text{-}17$$

This observation is the basis of the KCOF identification methods discussed next.

3.6.1 Wavelet transform method for edge detection

As noted earlier, theory predicts that transitions between sticking and sliding are abrupt, i.e. they are separated by edges. General edge detection methods have four steps in common: 1) smoothing to suppress the noise, without destroying the true edges; 2) enhancement by applying a filter to enhance the quality of the edges; 3) detection of thresholds, and 4) localization to determine the location of the edge. For example, the simplest edges coincide with peaks of the first-derivative of the input signal and “zero-crossings” of the second-derivative of the input signal. To achieve reasonable edge detection results, it is necessary to minimize false detection, e.g. where the algorithm cannot detect true sliding points or falsely detects sliding points and maximize correct detection.

Edge detection is a classic problem in signals and images. A widely used means of estimating edges or step changes in signals is based on the filtered derivative method, which combines smoothing with gradient operators to reduce noise effects. To overcome optimization problems associated with the use of two criteria, localization and signal-to-noise ratio (SNR), wavelet transform (WT) was founded on basis functions formed by dilation and translation of a prototype function (Daubechies 1990). Mallat and Zhong (1992) developed a discrete wavelet transform (DWT) based on smoothed gradient estimation, with smoothing varied over dyadic scale. However, DWT has some limitations (Percival and Walden 2006). First, it requires the length of the dataset to be dyadic (i.e., have a power of 2). Secondly, the output generated by DWT is highly dependent on the origin of the signal being analyzed. A small shift in origin affects the outputs generated, which is known as circular shift. Due to circular shift, it is difficult to align the transformed signals with time. To overcome the above two limitations, a modification of DWT called Maximal Overlap Discrete Wavelet Transform (MODWT) (Percival and Walden 2006) is used herein. MODWT is circular shift invariant and is not limited by the dyadic length constraint. Therefore, the signals are easier to interpret for time series analysis.

Whereas MODWT provides more flexible time-frequency resolution properties, one possible drawback is that the frequency resolution is rather poor in the high-frequency region. It faces some difficulties in discrimination between signals having close high-frequency components, such as noise and edges. Coifman and Wickerhauser introduced the wavelet packet transform (WPT), which uses alternative basis functions that are formed by taking linear combinations of the usual wavelet functions (Coifman and Wickerhauser 1992, Wickerhauser 1996).

The general idea of wavelet decomposition for wavelet transform and wavelet packet transform is applying low-pass (H) and high-pass (G) filters to each intermediate signal, recursively. While WT takes only the low-frequency components, WPT takes both the low-frequency components and the high-frequency components. This rich information with wide range of time-frequency resolution allows discrimination between noise and edges (Yen and Lin 2000). As shown by

Equation 3-18, a wavelet packet is a function with three indices, $W_{j,k}^{i}(t)$, where integer i, j and k are the modulation, scale, and translation parameters, respectively.

$$W_{j,k}^{i}(t) = 2^{j/2} W^{i}\left(2^{j} t - k\right) \quad i = 1,2,\cdots \qquad \text{3-18}$$

The wavelet packet component signal $f_j^i(t)$ can be expressed by a linear combination of wavelet packet function $W_{j,k}^{i}(t)$ as follows:

$$f_j^i(t) = \sum_{k=-\infty}^{\infty} c_{j,k}^{i} W_{j,k}^{i}\,(t) \qquad \text{3-19}$$

where the wavelet packet coefficients $c_{j,k}^{i}$ is

$$c_{j,k}^{i} = \int_{-\infty}^{\infty} f(t) W_{j,k}^{i}(t) dt \qquad \text{3-20}$$

where $f(t)$ is a discrete-time signal.

For higher decomposition levels, the WPT results in good resolution in the frequency domain but poor resolution in the time domain. Nevertheless, WPT has the same limitations as DWT (Percival and Walden 2006). Like MODWT, these drawbacks can be overcome by removing the downsampling step in the WPT as done in the maximal overlap discrete wavelet packet transform (MODWPT) method (Walden and Cristan 1998).

Edges are defined by detecting the abscissa where the wavelet modulus maxima converge at fine scales (Stephane 1999). A wavelet modulus maximum is defined as a point where $\left|f_j^i(t)\right|$ is locally maximum. In the lower scale, it is difficult to determine which belongs to the true edges due to noise. Even though the maxima become less and only the true edges remain when the scale increases, these maxima are located far away from the real edge.

To enhance edges and suppress noise in the signal, the multiscale product of the last component of WPT at each level is applied as Sadler and Swami (1999)

$$P_{j_0 j_1} = \prod_{j=j_0}^{j_1} f_j^{2^j} \quad \text{for } j_0\text{=1, } j_1\text{=1,2,3,4} \qquad \text{3-21}$$

The function $P_{j_0 j_1}$ shows peaks at edges and has relatively small values elsewhere. The idea of a cross-scale correlation was developed by Rosenfeld for edge detection in images (Rosenfeld 1970).

It is important to classify which wavelet coefficients represent the edges or noise in a measured acceleration signal. This depends on the characteristic of the selected wavelet and the properties of a given signal. Daubechies wavelet with one vanishing moment (db 1) (Daubechies 1992) was selected empirically from parametric studies. It is discontinuous and resembles a step function.

The results of the estimated KCOF from the simulations and experiments are compared in the following examples. Two types of earthquake motions are applied to the platen; “DUZCE.BOL” with 1602 record sequence number for sliding-dominant motion and KOBE/NIS with 1111 record sequence number for sticking-dominant motion from FEMA P-695. The selected records have been used by others for assessment of the probability of building collapse under the Maximum Considered Earthquake (MCE) as defined in ASCE/SEI 7-05 (2002). Two versions of KCOF are estimated, one for the interaction between the bottom surface of the smart device and the platen (sliding-dominant motion) and the other between the device with its protective shell and the platen (sticking-dominant motion).

Figure 3-4 and Figure 3-5 depict the results for both sliding- and sticking-dominant responses from simulation and as measured in the experiments, respectively, including the first three levels of normalized MODWPT components (f_1^2, f_2^4, and f_3^8) and multiscale products (P_{12}, P_{13}, and P_{14}) with the range of 0 to 1 values. The red circles in Figure 3-4 are the detected edges using the 0.3 threshold criterion. Because the simulated acceleration response has relatively sharp edges, the resulting MODWPT components and normalized cross-scale products P_{12} and P_{13} in Figure 3-4 depict clean peaks aligned with the edges. Note that the lower scale components, $f_1^{2^1}$, have narrower and sharper peaks at the edges than the higher scale components $f_2^4$and f_3^8.

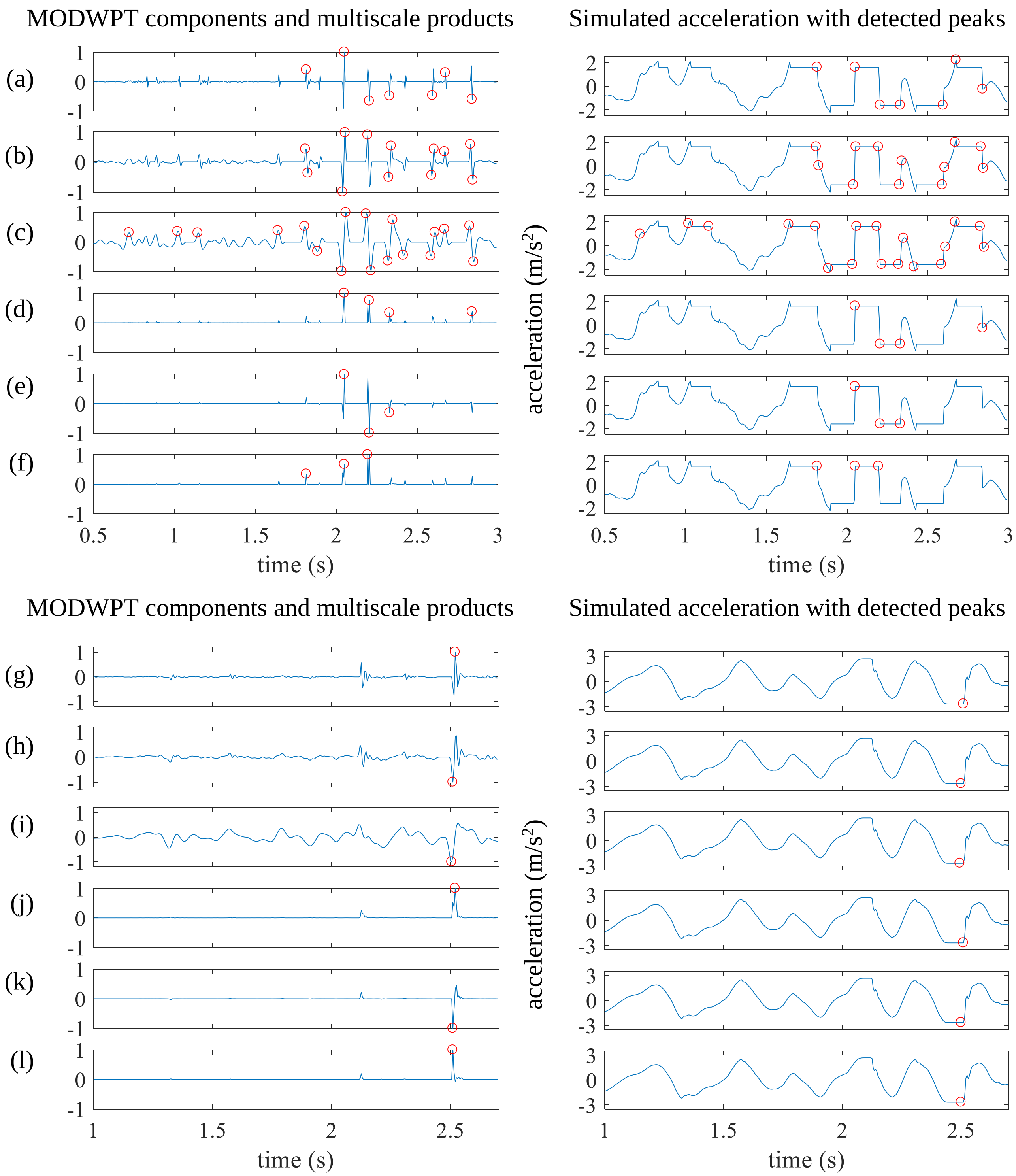


Figure 3-4. Edge detection results (from simulation data) for 1) sliding dominant response (a)-(f) and 2) sticking dominant response (g)-(l): a)-c) and g)-l) are the first three levels of the normalized MODWPT components, f_1^2, f_2^4, and f_3^8 with 'db1' wavelet; d)-f) and (j)-(l) are the normalized multiscale products, P_{12}, P_{13}, and P_{14}, respectively. Circles are detected edges with a threshold of 0.3.

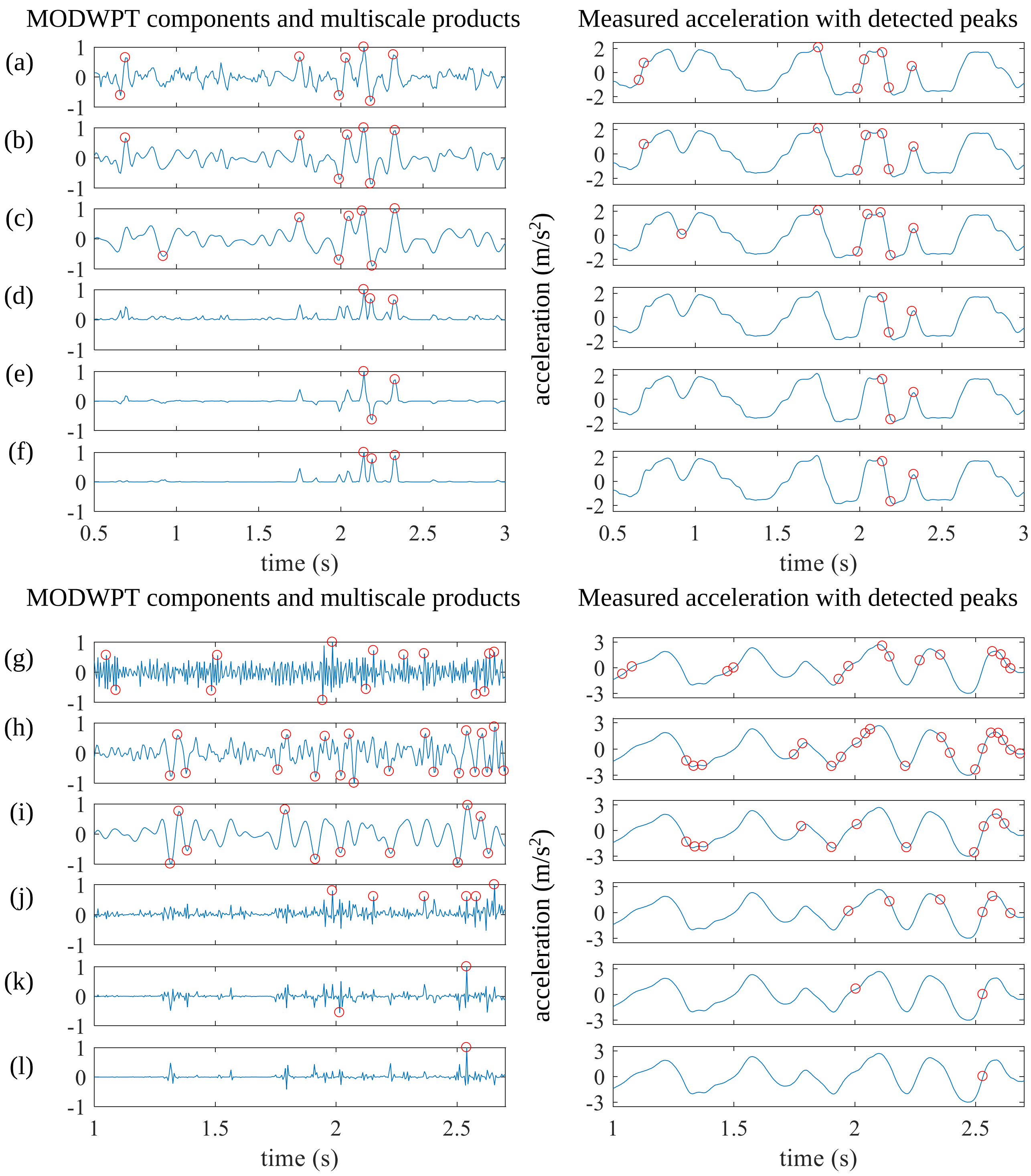
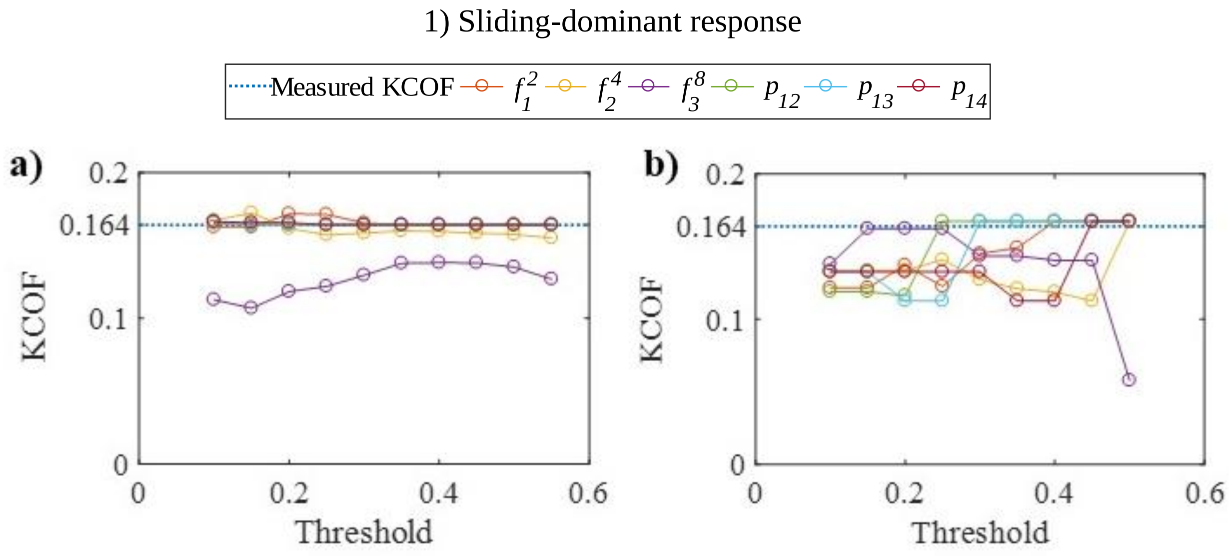


Figure 3-5. Edge detection results (experimental data) for 1) sliding dominant response (a)-(f) and 2) sticking dominant response (g)-(l): a)-c) and g)-l) are the first three levels of normalized MODWPT components, f_1^2, f_2^4, and f_3^8 with ‘db1’ wavelet; d)-f) and j)-l) are the normalized multiscale products, P_{12}, P_{13}, and P_{14}, respectively. Circles are detected edges with a threshold of 0.3

Figure 3-6 plots the relationship between the edge detection peak values and corresponding KCOFs for both simulated and measured acceleration responses. Figure 3-6a and c show that the estimated KCOFs with the low scale of MODWPT component, f_1^2, in particular, match well with the actual KCOFs for the sticking- and sliding-dominant responses, respectively. The estimated KCOF with higher scale MODWPT and multiscale products also match well with the given KCOFs beyond a threshold of 0.3. As such, a threshold value of 0.3 is adopted in this study.

On the other hand, the results from the experiments are quite different and, indeed, much worse. Figure 3-6b and d show that the estimated KCOFs do not match well with the given KCOFs for Daubechies wavelet with one vanishing moment (db 1). Poor edges detection can also be seen in Figure 3-5, where the resulting MODWPT components and normalized cross-scale products are unable to adequately isolate the edges for the given 0.3 threshold. The results suggest that the rounded edges at sliding and noise in the experimental data are highly influential and that an alternative KCOF estimation method is needed.



2) Sticking-dominant response

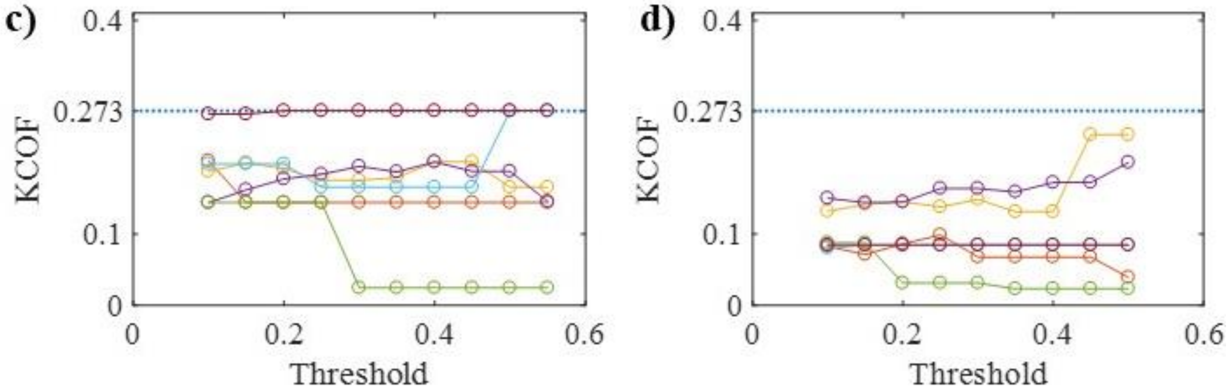


Figure 3-6. Effect of threshold for edge detection on estimated KCOF for both simulated and experimental data: a) simulated response; b) measured response with 'db1' wavelet; c) simulated response; and d) measured response with 'db1' wavelet

3.6.2 Noise reduction

Many noise reduction algorithms tend to distort the original signals. Denoising is generally done by band-pass filters, but these can eliminate non-smooth features of a signal such as the sharp transition to sliding motion in the acceleration response. This is important to preserve in the current application. WT, on the other hand, makes it possible to preserve such features and, at the same time, extract much of the noise. In particular, if the wavelet coefficients that belong to the noise details can be classified, it is possible to remove them and reconstruct the signal, yielding a denoised signal. Therefore, it is essential to identify which wavelet coefficients represent the noise in a measured acceleration signal.

The multiple levels of wavelet decomposition are obtained by applying low-pass and high-pass filters on the sub-sampled output of the low-pass filter, recursively. While the most significant coefficients of the signal are the low-frequency components (approximations), f_k^1 (for k=1, 2, 3, 4), which have the majority of the information of the signal, the high-frequency components (details) are embedded in $f_k^{2^k}$ (for k=1, 2, 3, 4), and are generally related to high-frequency noise components. Using these characteristics of WT, high-frequency noise can be eliminated by removing less relevant frequency components in the sub-band decomposition performed by the WT (El-Sheimy et al. 2004, Kang et al. 2010). In this study, wavelet de-noising is applied to remove part of the high-frequency components (short-term noise).

Experiments for earthquake motions as mentioned earlier are conducted to compare the denoising performance of three different methods: low-pass filter, MODWT, MODWPT. Accelerations recorded by the onboard accelerometers in the Samsung Galaxy S7 and computed from the motion recorded by the Optotrak system are compared in Figure 3-7. The Optotrak data is considered as a reference. The noise in the measured Galaxy S7 acceleration is calculated by the root mean square error (RMSE) deviations from the reference. In both the sliding-dominant and sticking-dominant response cases, MODWPT has the smallest RMSE compared to two other methods. The RMSE is 0.2320 for the sliding-dominant response case and 0.3832 for the sticking-dominant response case, as shown in Figure 3-7. These results suggest that the MODWPT can remove noise while still maintaining the shape of transitions between sticking and sliding or reversals in the direction of sliding.

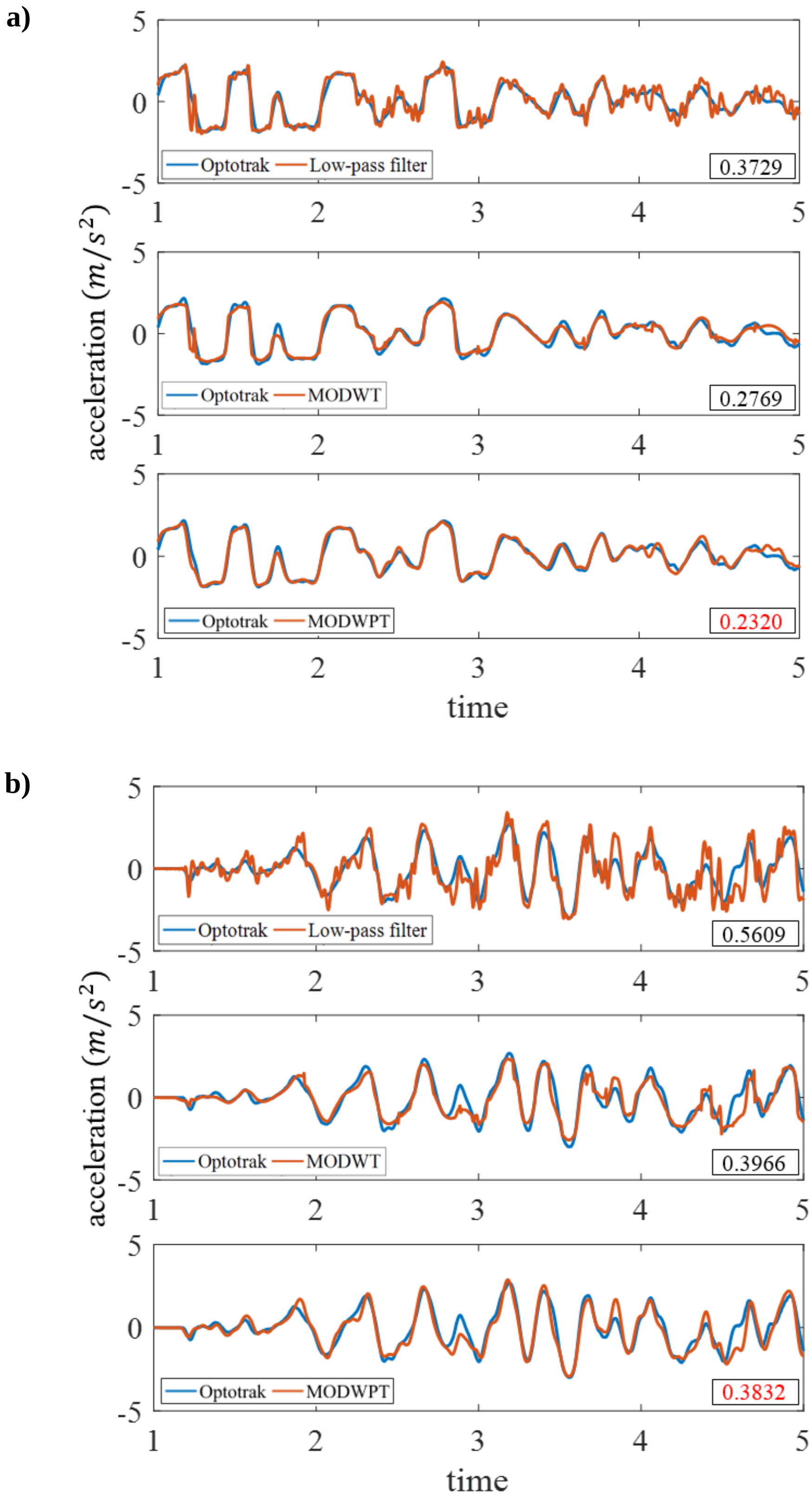


Figure 3-7. Comparison of RMSE for three different noise detection methods for a) sliding-dominant response and b) sticking-dominant response

3.6.3 Proposed KCOF identification method

A method is proposed to estimate KCOF based on sliding motions in the experimental record. Rather than focusing on sharp edges, which are prominent in the computational results, such as region A in Figure 3-3b, the proposed method associates sliding with plateaus, which are clearer in the experimental data, such as region B in Figure 3-3b. It is therefore proposed that during a sliding plateau, the first derivative of acceleration should theoretically be zero. The practical reality is that a margin of error must be accommodated since experimental results have fluctuations associated with imperfections in the sliding surfaces (the previously mentioned chattering).

Based on this idea, four parameters are applied to help isolate KCOF: 1) a minimum acceleration threshold, Th_a, is applied to eliminate the chance for erroneous detection associated with low seismic events, during which it makes no sense to assess seismic damage; 2) a zero threshold, Th_{zero} is applied to the first derivative of acceleration in order to accommodate the experimentally observed chattering in the sliding plateaus; 3) an outlier threshold, $Th_{outlier}$, is applied to eliminate falsely detected points due to fortuitous plateaus. The acceleration values of detected points are sorted by absolute magnitude and points below $Th_{outlier}$ are removed; 4) a minimum distance threshold, Th_d, is applied to eliminate non-consecutive detected sliding points due to chattering or reversed motions. The parameters $\bar{a}$ and $\dot{a}$ are denoted as the denoised acceleration and the first derivative of denoised acceleration, respectively.

As shown in the Figure 3-8, the proposed KCOF identification process is as follows:

STEP 1: Collect the acceleration data from Galaxy S7 smartphone with 200 Hz sampling rate,

STEP 2: Remove noise from the measured acceleration by MODWPT and obtain $\bar{a}$

STEP 3: Take the first derivative of the absolute value of denoised acceleration and obtain $\dot{a}$

STEP 4: Detect the sliding points in the plateau region, which satisfies the following conditions

$$|\bar{a}| > Th_a \text{ and } |\dot{a}| < Th_{zero} \tag{3-22}$$

STEP 5: Remove the outliers when the magnitude of acceleration is less than $Th_{outlier}$

STEP 6: Remove the non-consecutive detected points when the distance between neighbor detected points exceeds Th_d

STEP 7: Take the average of the absolute values of the detected sliding points to estimate KCOF

A survey of the literature shows that KCOF between commercial on-the-shelf smartphones and possible surfaces (such as desks, tables, carpeted floors, and tiled floors) exceeds a value of 0.1 (Friction 2018). This threshold corresponds to a minimum reasonable number that KCOF must exceed. Based on this observation, Th_a is selected as 0.1g.

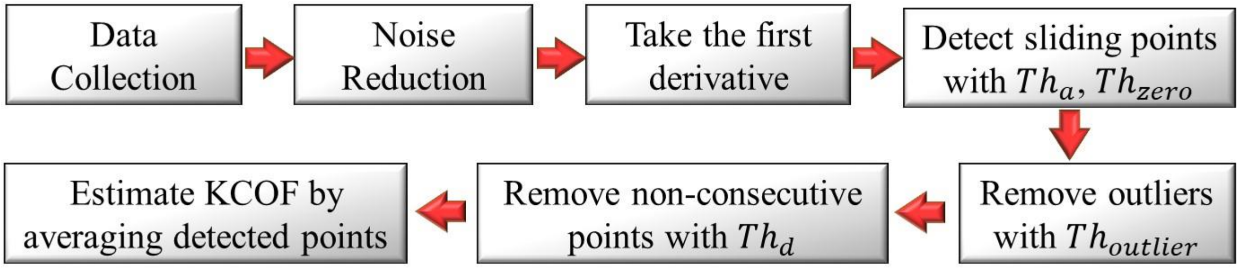
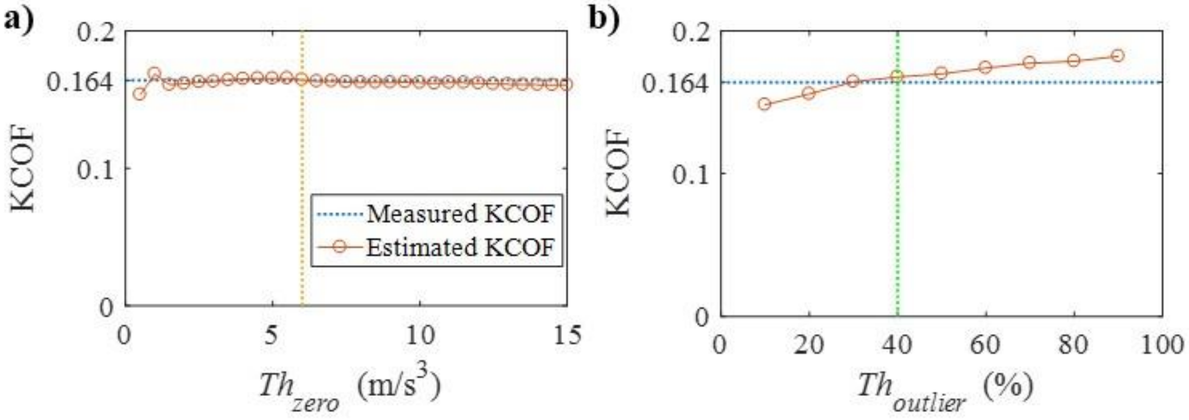


Figure 3-8. Flowchart of the proposed KCOF identification process

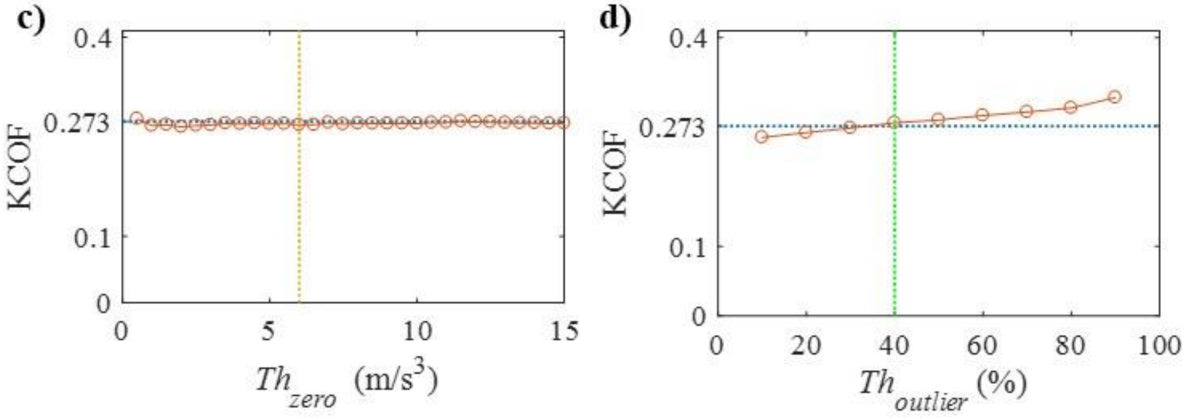
Figure 3-9 illustrates the effect of Th_{zero} and $Th_{outlier}$ on the estimated KCOF for the experimental data outlined earlier in section 3.6.3. As shown in Figure 3-9 a and c, the estimated KCOF becomes asymptotic with the measured KCOF when Th_{zero} is about 6 m/s^3, the number adopted in this work. As shown in Figure 3-9b and d, the estimated KCOFs for both sliding- and sticking-dominant responses become close to the measured KCOFs, respectively, when $Th_{outlier}$ is 40 %, the number adopted in this work.

1) Sliding-dominant response



(for Th_a=1 m/s2, $Th_{outlier}$=40%) **(for Th_{zero}=6 m/s3, Th_a=1 m/s^2)**

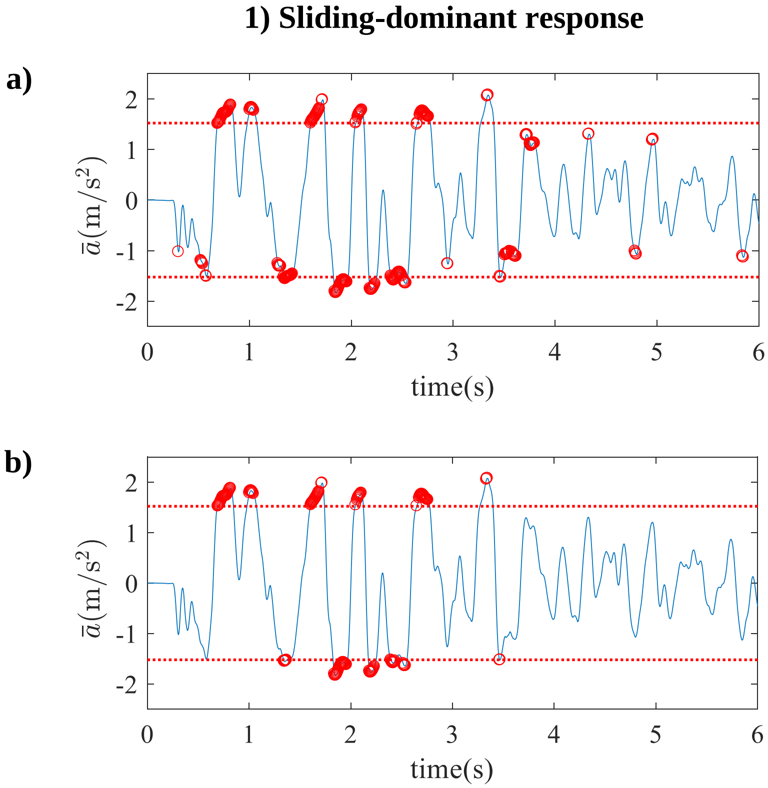
2) Sticking-dominant response



(for Th_a= 1 m/s^2, $Th_{outlier}$=40%) **(for Th_{zero}= 6 m/s^3, Th_a= 1 m/s^2)**

Figure 3-9. Effect of $Th_{outlier}$ and Th_{zero} on detected sliding points: a) and c) are the effect of Th_{zero}; b) and e) are the effect of $Th_{outlier}$

Figure 3-10 illustrates the effect of $Th_{outlier}$ and Th_d on the sliding detection results. The dashed line corresponds to $Th_{outlier}$ = 40%. Non-consecutive detected sliding points are removed when their distance between neighbor detected points exceed Th_d. The parameter Th_d is selected to be 4 time steps, i.e. 0.02 second, which is equal to 50Hz with a 200Hz sampling rate of the smartphone. This is deemed reasonable because it is higher the 0.5-25Hz frequency band commonly observed in strong motion measurements (Trifunac and Todorovska 2001).



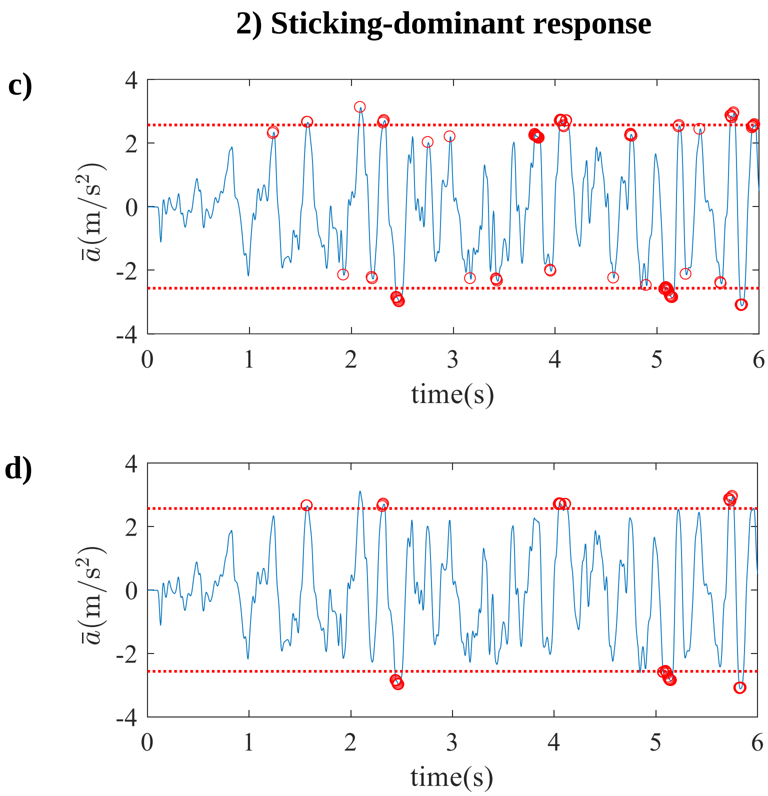
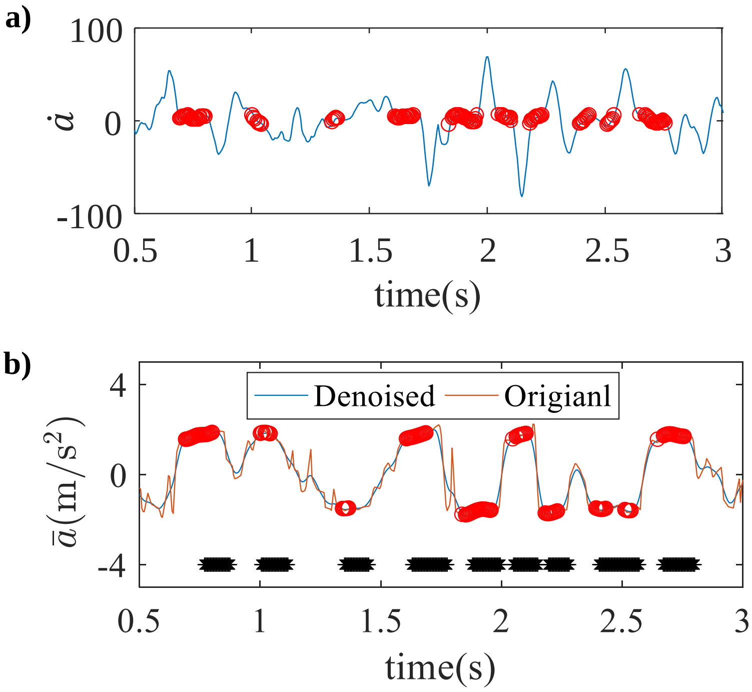


Figure 3-10. The effect of $Th_{outlier}$ of 40% and Th_d of 0.02s (for Th_{zero} = 6 m/s^3, Th_a = 1 m/s^2): a) Detection result at STEP 3; b) Detection result at STEP 5; c) Detection result at STEP 3; and d) Detection result at STEP 5. Dash lines are 40% in detected points

The proposed method is demonstrated using the experimental data outlined earlier in section 3.6.3. Figure 3-11 shows the results of the KCOF estimation process for both sliding- and sticking-dominant responses. The red circles are the detected sliding points on the plateaus as identified by the proposed method. It is clear that they match reasonably well the actual sliding points (black stars in Figure 3-11b and d) determined by the Optotrak measurements.

1) Sliding-dominant response



2) Sticking-dominant response

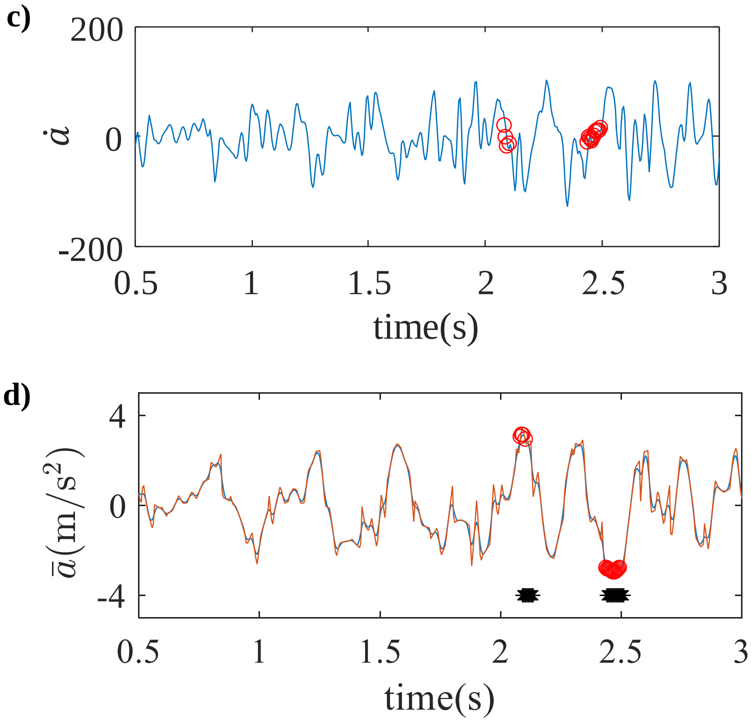


Figure 3-11. Measured response by Galaxy S7 for Th_a=1 m/s^2, Th_{zero}=6 m/s^3, $Th_{outlier}$ of 40% and Th_d of 0.02s: a) First derivative of the acceleration; b) Measured acceleration; c) First derivative of the acceleration; and d) Measured acceleration. Circles are detected plateau-related sliding points. Black stars are actual sliding motions measured by Optotrak

3.7 Validation and discussion

Four different ground motions are used to validate the proposed method. The records are selected from FEMA-P695 (2009). A total of 10 repetitive experiments are conducted and accelerations are measured by the Galaxy S7 device with/without the protective shell for each earthquake. Table 3-2 summarizes the experimental results. The average of the estimated KCOF from the Galaxy S7 are 0.173 for sliding-dominant responses and 0.264 for sticking-dominant responses. These results show that the developed KCOF identification method is in good agreement with the experimentally measured KCOF for sliding- and sticking-dominant responses (3.5% and 2.6% error, respectively). The standard deviations of estimated KCOF values for sliding-dominant responses are slightly higher than those for sticking-dominant responses.

Table 3-2. Results of the estimated KCOFs by Galaxy S7

EQ ID	PEER-NGA Record Information		Sticking-dominant response		Sliding-dominant response	
	Name	**Record No.**	**Mean**	**Std**	**Mean**	**Std**
EQ1	NORTHR/MUL	933	0.181	0.004	0.274	0.016
EQ2	DUZCE/BOL	1602	0.169	0.006	0.272	0.010
EQ3	IMPVALL/H-DLT	169	0.185	0.005	0.269	0.012
EQ4	IMPVALL/H-E	174	0.176	0.003	0.279	0.005

Using the procedure described above, sliding motions can be detected from the acceleration records of smart devices. The idea is that sliding motions occur when the onboard accelerometers measure accelerations that exceed to the slip acceleration in Equation 3-17 with the estimated KCOF. Figure 3-12 shows the results of detected sliding motions for both sliding- and sticking-dominant responses as computed from the Optotrak and Galaxy S7 measurements. Clearly, the detected sliding motions computed from the Galaxy S7 measurements are in close agreement with those from the Optotrak with 93.9% and 95.3% correct detection rate for sliding- and sticking-dominant response, respectively. The correct detection rate is the ratio of correctly matched sliding and sticking states to the number of points in the record.

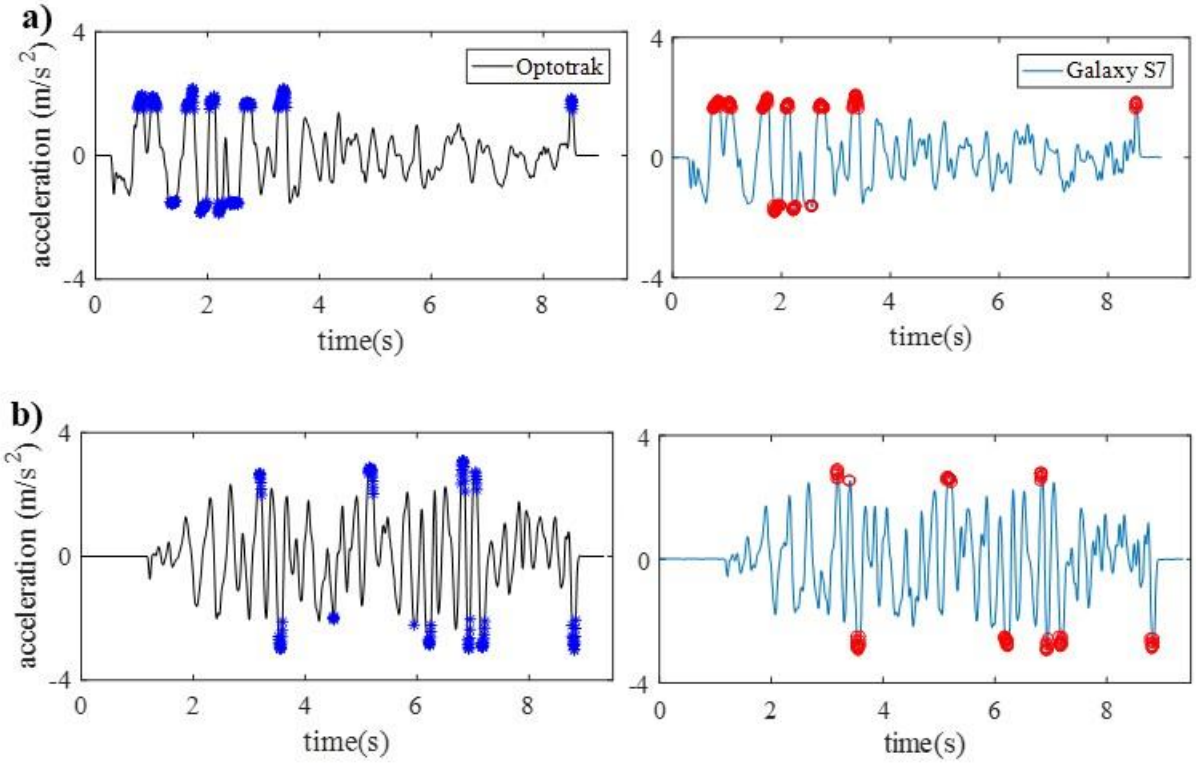


Figure 3-12. Results of the detected sliding motions for a) sliding-dominant response and b) sticking-dominant response. The circles are detected sliding motion points

3.8 Summary and conclusion

To utilize smart devices to characterize building seismic damage in a quantitative manner, two key challenges are highlighted. First, the error equation for MEMS-based accelerometers in smart devices is presented and deterministic errors estimated experimentally by using the static multi-position method. The information presented suggests that MEMS-based accelerometers in smart devices have negligible errors due to bias and scale factor. Also, it is shown that the optimal sampling rate to minimize error for a specific smart device must be in excess of 1000Hz. This means that new generations of smart devices must have a substantially higher sampling rate for their data to be accurate for the proposed application. Second, a sliding detection method is introduced to decide whether the motion of the smart device is representative of the motion of the floor underneath or if it is contaminated by excessive sliding action. In the simulated acceleration response, it was shown that the theoretical sliding motions have step shapes with a clear plateau. However, in the measured acceleration response, the sliding motions were observed to have rounded shapes and chattering, which hinders the ability to differentiate between sliding and sticking motions. To detect sliding motions, the conventional edge detection method based on wavelet transform analysis is reviewed. In addition, noise reduction methods are presented based on the use of wavelet transforms to alleviate noise in the smart device's accelerometer. It is shown that the MODWPT method out performs other wavelet transform methods in noise reduction for measurement of acceleration by the smart device, while still maintaining the shape of transitions between sticking and sliding. A new method for KCOF estimation is proposed, which includes acceleration noise reduction using MODWPT and a plateau template with configurable thresholds to address the occurrence of chattering during sliding. The proposed method is validated using shake table experiments and it was shown that the results of the estimated KCOF are in good agreement with the true KCOFs. The estimated KCOF is used to compute slipping accelerations and corresponding sliding motions. It is shown that a 93.9% correct detection rate can be achieved using the proposed method.

CHAPTER 4

Stick-slip classification based on machine learning techniques

4.1 General

This chapter presents an accurate and robust accelerometer-based stick-slip motion classification framework based on machine learning techniques. To improve upon the stick-slip identification done in Chapter 3, two types of machine learning algorithms (supervised learning and deep learning) are introduced and their classification performance compared. Three classification conditions are considered for the supervised learning algorithm: feature selection, non-linear discriminating analysis and classifier comparison. Three hyperparameters are considered for the deep learning algorithm to find the best performing classification algorithm. Each algorithm is trained and validated with experimental acceleration data from a shaking table test.

4.2 Literature review

To the author's knowledge, no prior studies have examined stick-slip motion classification based on machine learning. Stick-slip motion classification is analogous to Human Active Recognition (HAR) as both use smart phone generated acceleration data to discover meaningful characteristics. For the latter, machine learning algorithms have been successfully used to detect the type of activity being performed from accelerometer readings. The accelerometer data used are typically categorized into separate classes through a classification process. This classification problem is a multidisciplinary research area which shares connections with machine learning, artificial intelligence, machine perception and ubiquitous computing. Because of its wide-ranging capabilities, classification using accelerometers has drawn increasing interest from researchers in a variety of fields.

The majority of movement classification systems with accelerometers have been custom designed for a specific domain of activities. Some of these designs have investigated the use of acceleration signals to analyze and classify different types of the same physical activity (Mantyjarvi et al. 2001). Others have employed acceleration signals for recognizing a wide set of daily physical activities (Najafi et al. 2003). The identification of recurring activities of workers were studied using accelerometer data (Minnen et al. 2005). Other studies have investigated fall detection and prevention for the elderly (Giansanti et al. 2008, Narayanan et al. 2008).

There exists a considerable body of literature on HAR methods that use various types of machine learning algorithms. Some studies have developed the idea of simple heuristic classifiers (Foerster et al. 1999, Najafi et al. 2003, Karantonis et al. 2006). Others studies have investigated more automatic and comprehensive methods from the machine learning literature including decision trees, nearest neighbor and Bayesian networks (Bao and Intille 2004, Maurer et al. 2006, Ermes et al. 2008), support vector machines (SVM) (Bao and Intille 2004), neural networks (Kiani et al. 1997, Mantyjarvi et al. 2001), Gaussian mixture models (GMM) (Allen et al. 2006), and Markov chains (Kern et al. 2003, Minnen et al. 2005, Sung et al. 2005).

Unlike activity recognition methods that require feature vectors as input, deep learning algorithms use the original data directly. In particular, recurrent neural networks (RNN) is suitable for handling time-series data, such as audio and video signals, and natural language. This allows the calculation of feature vectors to be skipped at the time of training and recognition. A method of human activity recognition from raw accelerometer data applying a

RNN was proposed and various architectures and its combination to find the best parameter values were investigate by (Inoue et al. 2018). A generic deep framework for activity recognition based on convolutional and LSTM recurrent units was proposed by (Ordóñez and Roggen 2016). Many of these systems have produced excellent results in classification of specific movements. Given these successful applications, there is scope for the development of similar accelerometer-based systems that can automatically identify and classify stick-slip motion of smart devices under earthquake.

A number of existing studies in the broader literature have examined techniques for identifying structural damage using machine learning algorithms. A statistical feature extraction process was studied to classify features capable of detecting structural damage based on accelerometer data (Figueiredo et al. 2009). An automated crack detection algorithm using image processing was presented to automatically tune the threshold parameters to identify cracks more accurately (Prasanna et al. 2016). A sensor data fusion system based on machine learning classification methods for structural health monitoring (SHM) applications was proposed (Vitola et al. 2017). The residual error of an auto-regressive model of acceleration time series was used for damage detection based on support vector machines (SVM) for SHM (Gui et al. 2017). Convolutional neural networks (CNN) were used to estimate damage severity based on accelerometer data (Abdeljaber et al. 2018). In spite of these studies, there have been no efforts to identify structural damage based on the ideas proposed in this dissertation, i.e. processing accelerometer measurements from free-standing smart devices that are prone to slip-stick motion during seismic events.

4.3 Stick-slip motion classification methodology

The objective of this chapter is to investigate the feasibility of using the accelerometers in smart devices subjected to seismic motions as a means for classifying stick-slip motion based on machine learning techniques. To achieve plausible classification results, three important classification conditions are considered: 1) the consideration of an optimal window size; 2) feature selection; and (3) selection of the best performing classifiers. For classifiers, two types of machine learning algorithm are studied: a supervised learning algorithm and a deep learning algorithm. Supervised learning algorithms require features to be extracted by users on the basis of the characteristics of segmented input data. Figure 4-1 shows the overall research methodology used for the supervised learning framework, which consists of (1) data collection and labeling; (2) data segmentation; (3) feature extraction; (4) discriminant analysis; (5) classification; and (6) validation. Unlike supervised learning algorithms, deep learning algorithms do not required users to extract features. This work is accomplished via the algorithm itself. Figure 4-2 shows the overall research methodology for the deep learning framework, which consists of (1) data collection and labeling; (2) classification; and (3) validation.

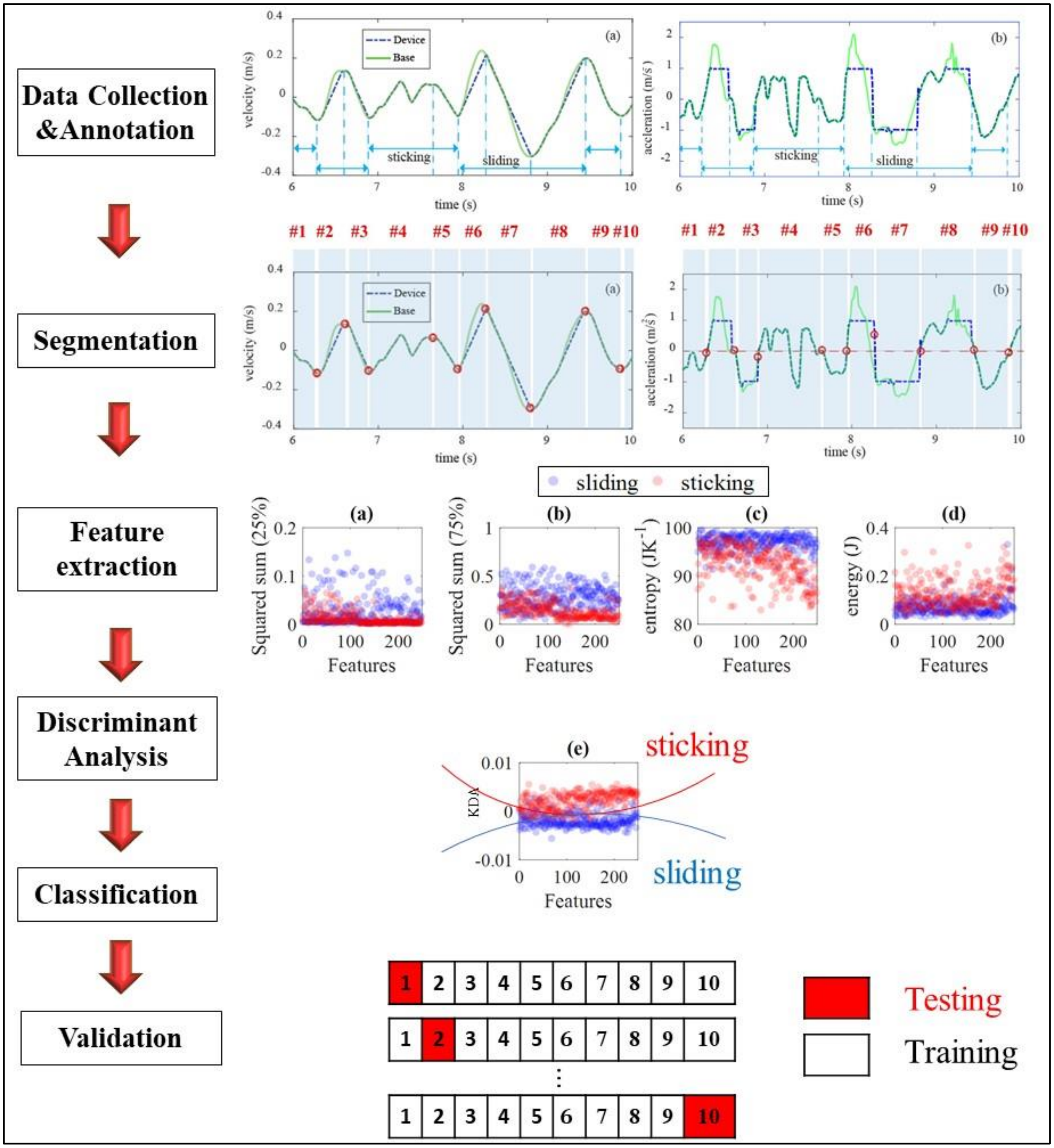


Figure 4-1. Block diagram of the supervised learning algorithm

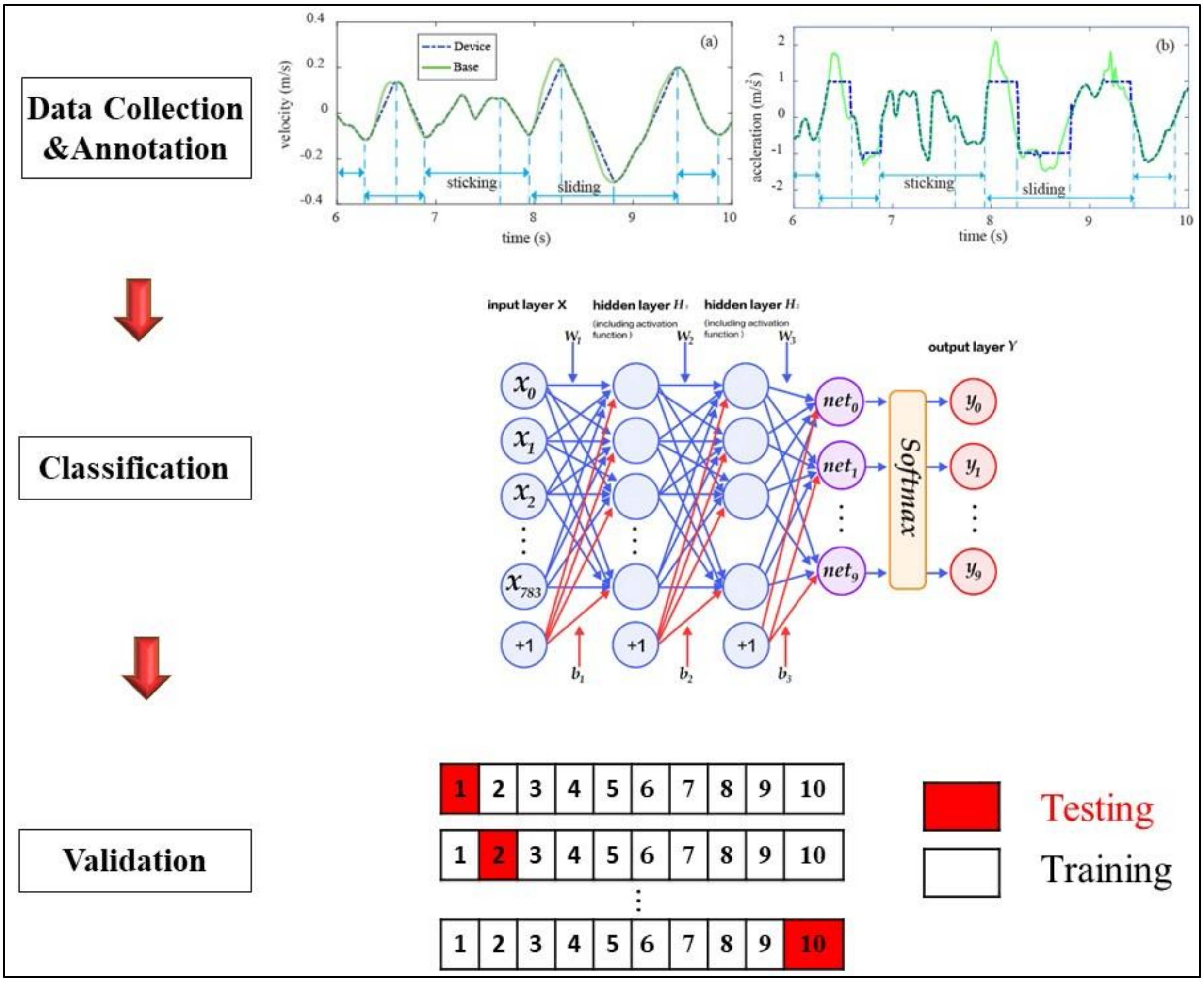


Figure 4-2. Block diagram of the deep learning algorithm

4.3.1 Data Collection

Acceleration data were collected by placing the phone on a shaking table (Figure 2-8). The two motions of interest were sticking and sliding. To simulate earthquake events, six ground motions (EQ1, EQ2, EQ3, EQ5, EQ6, and EQ7) were selected from Table 2-3. The motions were selected because they could lead to substantial sliding action. Five different smartphones were used as shown in Figure 4-3 and summarized in Table 4-1. To collect substantial amount of experimental results for machine learning, each smartphone has two surfaces with and without a protective shell. Each smartphone recorded ten rounds for each ground motion to achieve a large enough dataset to facilitate machine learning. A total of 600 measured acceleration data records were collected.

Figure 4-3. Smartphones used in the test: a) Samsung Galaxy S7; b) Samsung Galaxy S5; c) LG G5; d) Nexus 6P; and e) Nexus 5X

Table 4-1. List of smartphones used in the test

	Sampling rate (Hz)	MS amplitude of noise (mg)	Maximum Range (m/s^2)	Resolution (mm/s^2)
Samsung Galaxy S7	200	106.09	19.6	0.599
Samsung Galaxy S5	100	416.16	19.6	0.599
LG G5	100	268.96	156.9	0.958
Nexus 6P	100	338.56	156.9	0.958
Nexus 5X	100	930.25	39.2	1.200

4.3.2 Annotation Method

The infrared 3D-motion capture system (Optotrak) was used to capture the true displacement of the smart device and shake table platen as mentioned in Section 2.5.1. To annotate sliding and sticking motions for the learning methods, sliding was considered to occur when the relative velocity between the base and the device exceeds the Stribeck velocity (De Wit et al. 1995) as mentioned in Section 2.3.1. The annotation results for stick-slip motions obtained by the Optotrak were quite accurate, as validated by an analysis of video recordings of the motions of the smartphone and underlying surface.

4.3.3 Segmentation Technique

Like any other supervised classification problem, in stick-slip motion classification, the sensor signal is first divided into smaller time segments more commonly known as windows. For each window, features are computed separately and fed to the classification algorithms. This information is then combined to give a motion profile along the entire signal. Unlike human activity, which has some distinct characteristics, such as a certain motion with a certain length or two activities occurring consequently, stick-slip motions are quite random events. This randomness poses a challenge for the application of the conventional window method. To address the issue of randomness, in this study the acceleration measurements are snipped into small segments bounded by the points where the acceleration crosses zero because these points are transitions associated with sliding reversals. This segmentation is denoted as zero crossing (marked with red circles in Figure 4-4c) and compared the classification performance with other window sizes such as 0.25, 0.5, 0.75, 1.00, 2.00, and 3.00 seconds length of window. These windows are defined by a fixed windowing approach that divides the data into single segments with no overlapping data, with a specific length of window size as shown in Figure 4-4a. This fixed windowing approach shows the better performance in the other classification study than the sliding windowing approach, which allows overlap of the data between segments (Jebelli et al. 2018). as shown in Figure 4-4b.

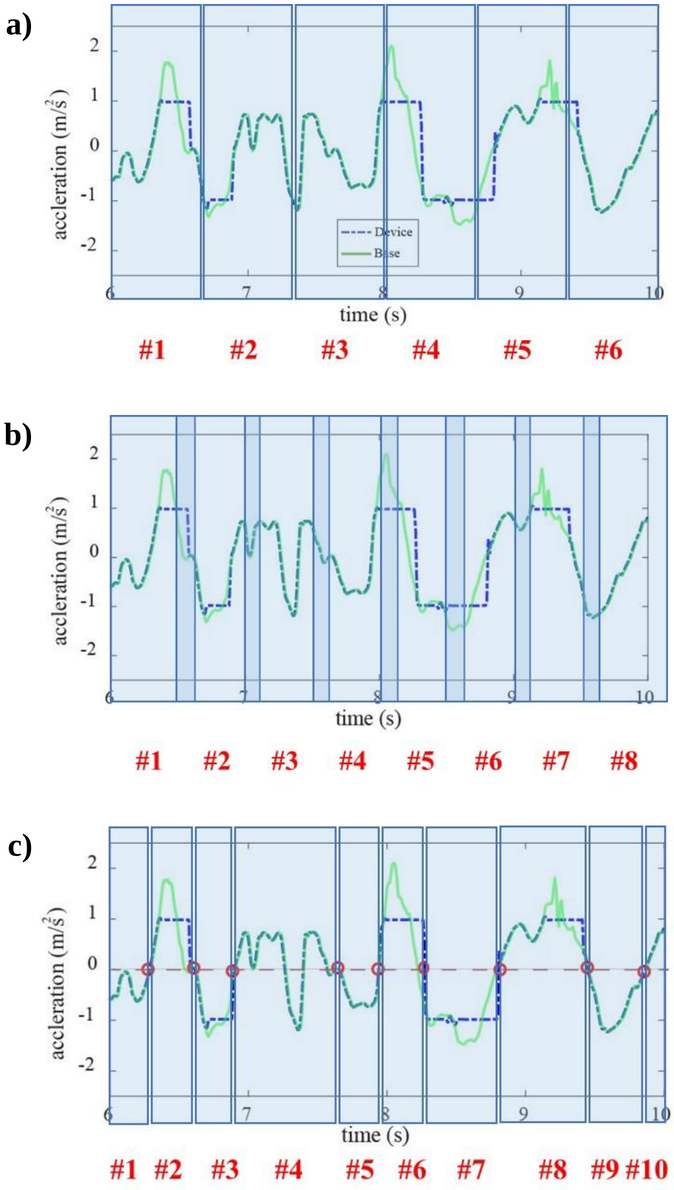


Figure 4-4. Windowing approaches: a) fixed window; b) sliding window; c) zero-crossing window (the window numbers are indicated on each window)

4.3.4 Feature

After labeling the data with specific segment numbers, features representing each of the segments were extracted for motion classification. These features were then fed to a classifier that implements a specific classification algorithm. Features are extracted with respect to both the time and frequency domain because these two approaches are widely used in classification methods. Some studies have derived time-domain features directly from a window of acceleration data. Examples include the mean, median, variance, skewness, and kurtosis. To derive frequency-domain features, the window of sensor data must first be transformed into the frequency domain, normally using a fast Fourier transform (FFT). The output of a FFT typically gives a set of basis coefficients which represent the amplitudes of the frequency components of the signal and the distribution of the signal energy.

Eight different features were selected (6 for time-domain features and 2 for frequency-domain features). This choice was based on their widespread use in accelerometer-based activity recognition studies (Baek et al. 2004, Ravi et al. 2005, Koskimaki et al. 2009). Smartphone accelerations were then integrated once to yield velocity data and the same eight features were selected. The total number of potential features was 16. The time-domain features include the following: (1) mean, an average value of acceleration data over the window; (2) standard deviation of acceleration values in each window; (3) range (difference between maximum and minimum values); (4) skewness (a degree of asymmetry in the distribution of acceleration data); (5) the squared sum of the magnitude of data below 25 cumulative percentage and (6) below 75 cumulative percentage. The features in the frequency domain are (7) energy of wavelet decomposition in the frequency-domain (Laine and Fan 1993) and (8) entropy of the wavelet packet in the frequency-domain (Coifman and Wickerhauser 1992). Energy and entropy features have also been used to identify the states of movement and differentiate actions that have a similar energy level, respectively.

4.3.5 Feature Selection

Characterizing sliding accelerations can be challenging for two reasons. First, the acceleration of sliding motion is highly related to the KCOF between the smartphone and the underlying surface. As a result, the sliding acceleration response would be different for each phone because each smartphone can have a different KCOF. Second, sticking motion is a random movement because it follows random ground motions. Due to this randomness, it is difficult to define a cycle time of a specific motion unlike human activities (Ryu et al. 2018). To achieve effective classification, selecting high discriminative features is important. The high discriminative feature set should have similar attributes between repetitions of either stick or slip motion but should have distinct properties between two different motions.

In order to achieve better classification, in this study, the ReliefF algorithm was used as a feature selection method because it has been widely used for feature selection (Menai et al. 2013). The algorithm iteratively determines k-nearest features of the same and different classes from randomly sampled instances in the training data set. It estimates the importance weight by averaging their contribution (Hall 1999). The parameter k is related to the distance of estimations and can be set heuristically to 10 for most purposes (Robnik-Šikonja and Kononenko 2003). The importance weight of each feature represents the degree of class differentiation. A larger feature weight indicates a more important feature, and the algorithm sets a rank for each feature based on the weight (Hall 1999).

4.3.6 Dimension Reduction and Discriminating Feature Extraction

Due to similarities in time and frequency-domain parameters, discrimination of stick-slip motions was difficult. This difficulty is attributable to the presence of low between class variance in the motion-data that resulted in poor classification accuracy and had to be overcome using a method that could achieve dimension reduction as well as increase the low between-class variance to increase the class separability. Dimension reduction was achieved by means of extracting discriminating features, which works on the idea of maximizing total scatter of the data, while minimizing the variance within classes. The most common techniques for this purpose include linear discriminant analysis (LDA) (Mika et al. 1999) and kernel discriminant analysis (KDA) (Baudat and Anouar 2000).

LDA seeks to find a linear transformation by maximizing the between-class variance and minimizing the within-class variance; as such, it has proved to be a more suitable technique for classification than original features (Li et al. 2003). This method maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. The output of LDA is an optimal linear discriminant function which maps the input into the classification space on which the class identification of the samples is decided. Further details on the LDA are available in Belhumeur et al. (1997).

Although LDA is a major improvement in discriminating features, it is still a linear technique in nature. Because of this, when severe non-linearity is presented, this method is intrinsically poor. A better analytical approach is Kernel Discriminant Analysis (KDA), a non-linear discriminating approach based on kernel techniques that is designed to find non-linear discriminating features. KDA computes the non-linear discriminating basis vectors which has shown good performance in cases where LDA has failed (Li et al. 2003). Details of KDA algorithm are available in Baudat and Anour (2000). The performances of each of these techniques will be discussed later.

4.3.7 Classifiers

Machine learning techniques are used to learn acceleration signal of sticking-slip motions from training data, and then classify types of motions from testing data. To classify stick-slip motions, two types of machine learning algorithms are studied: a supervised learning algorithm and a deep learning algorithm.

For the supervised learning algorithm, two widely used algorithms for HAR are employed and their performance are compared: support vector machine (SVM) and multilayer perception (MLP) artificial-neural networks based on the feed-forward backpropagation algorithm. The SVM classifies data by finding the best hyperplane that separates all data points of one class from those of another class. The best hyperplane for an SVM is the one with the largest margin between the two classes. Here, margin refers to the maximal width of the slab parallel to the hyperplane that has no interior data points. The support vectors are the data points that are closest to the separating hyperplane; these points are on the boundary of the slab. SVM can be improved using kernel function to create non-linear classifiers. In this study, the kernel function used for non-linear classification is the Gaussian radial basis function (Bishop 2006), which has been successfully applied in another activity recognition study (Ravi et al. 2005). Further details of SVM models can be found in Bishop (2006).

The MLP consists of three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. To optimize performance MLP with different number of layers and neurons were tested. The training of MLP classifier was also repeated several times by changing the input order in a random fashion. After several trials, one hidden layer with ten neurons and one output layer with two neurons corresponding to the two classification outputs (sliding and sticking) were selected.

For the deep learning algorithm, a recurrent neural network (RNN) is employed because it is the most suitable algorithm for classification problems with time series data (Hochreiter and Schmidhuber 1997). This method was developed to tackle one important limitation of the MLP architecture, which assumes all inputs and outputs are independent of each other when in fact, they should be considered dependent. To resolve this problem, RNN uses a recurrent connection for every unit. The response of a neuron is fed back to itself with a weight and a unit time delay. This provides a memory of past responses (hidden values) to learn the transitory dynamics of sequential data. For this study, a long short-term memory networks (LSTM) is used (Bishop 2006).

The core components of an LSTM network are a sequence input layer, an LSTM layer, a fully connected layer, a softmax layer, and a classification output layer, as shown in Figure 4-5. The sequence input layer inputs sequence or time series data into the network. The LSTM layer learns the long-term dependencies between the time steps of the sequence data. The fully connected layer multiplies the input by a weight matrix and then adds a bias vector. In this layer, all neurons in a fully connected layer connect to all the neurons in the previous layer. This layer combines all of the features (local information) learned by the previous layers across the data to identify the larger patterns. For classification problems, the last fully connected layer combines the features to classify the signals.

Figure 4-5. Architecture of long short-term memory networks (LSTM) (Demuth and Beale 1994)

In the following, some important details of the RNN with the LSTM algorithm are introduced including a softmax function, a loss function, a stochastic gradient descent, and a gradient clipping method. In addition, some important parameters which cannot be directly estimated from the data are investigated. These types of model parameters are referred to as a hyperparameters (or tuning parameters) in machine learning because there is no analytical formula available to calculate appropriate values (Kuhn and Johnson 2013).

Softmax function: In the softmax layer, the softmax function is a neural transfer function that calculates a layer's output from its net input. For classification problems, the softmax layer and the classification layer must follow the final fully connected layer. The softmax function is also known as the normalized exponential and can be considered a multi-class generalization of the logistic sigmoid function (Bishop 2006). This softmax function is a generalization of the logistic function that squashed a *K*-dimensional vector $\boldsymbol{z}$ of arbitrary real values to a *K*-dimensional vector $\sigma(\boldsymbol{z})$ of real values, for $K \geq 2$, where each entry is in the interval (0,1), and all the entries sum to 1:

$$\sigma: \mathbb{R}^{\mathrm{K}} \rightarrow \left\{ \sigma \in \mathbb{R}^{K} \, | \sigma_i > 0 \ , \sum_{i=1}^{K} \sigma_i = 1 \right\} \tag{4-1}$$

The standard softmax function is given by the standard exponential function on each coordinate, divided by the sum of the exponential function applied to each coordinate, as a normalizing constant.

$$\sigma(\boldsymbol{z})_{\mathrm{j}} = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for j=1,2,…,} K \tag{4-2}$$

Loss function : In the classification layer, the RNN takes the values from the softmax function and assigns each input to one of the K mutually exclusive classes using a loss function (error function), called a cross-entropy function for a 1-of-K coding scheme (Bishop 2006):

$$E = \sum_{i=1}^{N} \sum_{j=1}^{N} t_{ij} \ln y_{ij} \tag{4-3}$$

where N is the number of samples, K is the number of classes, t_{ij} is the indicator that the i th sample belongs to the j th class, and y_{ij} is the output for sample i for class j, which in this case, is the value from the softmax function.

Stochastic gradient descent method: To minimize the loss function, the stochastic gradient descent method is used. The gradient descent method searches the local minimum value in the adjacency by repeating to change w in the negative gradient direction by a very small amount. The weight in the t^{th} time of repetition be w^t,

$$w^{t+1} = w^t - \epsilon \nabla E \tag{4-4}$$

where ϵ is called as a learning rate, and ∇E is the gradient of the loss function. As the learning rate is high, learning becomes faster, but it also vibrates near the local minimum value of the loss function leading errors. To resolve this problem, a method to adjust the learning rate, called adaptive moment estimation (Adam), is employed (Krishnan and Cook 2014). The stochastic gradient descent algorithm evaluates the gradient and updates the parameters using a subset of the training set. This subset is called a mini-batch. Each evaluation of the gradient using the mini-batch is an iteration. At each iteration, the algorithm takes one step towards minimizing the loss function. The full pass of the training algorithm over the entire training set using mini-batches is an epoch. In this study, the 20 mini-batch size is selected.

Gradient clipping method : The LSTM layer is utilized mainly to replace some units of the RNN to solve two problems: 1) an input and output weight conflict, which is the conflict between the input from the previous layer and the recurrent value (Hochreiter and Schmidhuber 1997); and 2) the vanishing and exploding gradient problem, which is the large increase in the norm of the gradient during training due to the explosion of the long-term components and the opposite behavior (Bengio et al. 1994, Pascanu et al. 2013). To resolve these problems, the input/output weight conflict and the vanishing gradient problem are solved by the input/output gate and a constant error carousel (CEC), respectively. The exploding gradient problem is solved by a gradient clipping method. This gradient clipping corrects the L2 norm of the gradient not to exceed the threshold value (Pascanu et al. 2013). Specifically, when c is the threshold value, $\|\nabla E\| \geq \mathrm{c}$ is satisfied and the gradient is modified as follows

$$\nabla E \leftarrow \frac{c}{\|\nabla E\|} \nabla E \qquad 4\text{-}5$$

Table 4-2 summarizes the hyperparameters and corresponding values for the RNN. To achieve the best performing classification, an examination of three other hyperparameters (numbers of layers and units, and gradient clipping parameter) is conducted to the study of motion classification using the RNN.

Table 4-2. Predefined hyperparameters of RNN

Hyperparameters	Values
Activation function of output layer	Softmax
Loss function	Cross entropy
Type of internal layer unit	LSTM
Mini-batch size	20
Initial weights random	Gaussian distribution (μ=0, σ^2=0.01^2)
Initial bias	None
Learning rate adjustment method	Adam
Gradient Threshold Method	L2 norm
Input dimension	2
Output dimension	2

4.4 Classification results

The present study evaluates the performance of the proposed approach for classifying stick-slip motions. The data from the shaking table test were used for motion classification to determine the best combination of features, window sizes, classifiers, and discriminant methods. To assess the classification accuracy, a 10-fold cross-validation was performed using data from smartphones (Kuhn and Johnson 2013, James et al. 2013). From this cross validation, it is possible to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. The classification accuracy was calculated by dividing the number of correctly classified actions by the total number of instances in the data set.

4.4.1 Feature selection and window size on classification accuracy

To assess whether all selected features are robust for motion classification as described in the section 4.3.4. and select optimal features from the potential features of 16, the ReliefF algorithm was used. This algorithm can select highly correlated features among the potential features since it is challenging to avoid selecting superfluous features (Atallah et al. 2011). To resolve this problem, the threshold values were selected by comparing feature importance weights determined through the algorithm (Ryu et al. 2018).

The feature importance weights from each window size and average weight are shown in Figure 4-6. For all window sizes, the importance weights were lower than the averaged importance weights for the first 8 features. To select highly correlated features for classification performance, the top 8 features were selected for the minimum threshold in this study. To examine the claim that the use of more features can produce a better distinction (Hall 1999), the top 8, 12, and 16 features, corresponding to 50%, 75%, and 100% of the total features, respectively, were used.

Figure 4-7 shows the classification accuracy according to the number of selected features and the classifiers when data from all window sizes (each classification accuracy represents the average of the classification accuracy using all window sizes) are used. Two classifiers showed the best performance when using all 16 extracted features. Therefore, all 16 extracted features were used in the following analysis.

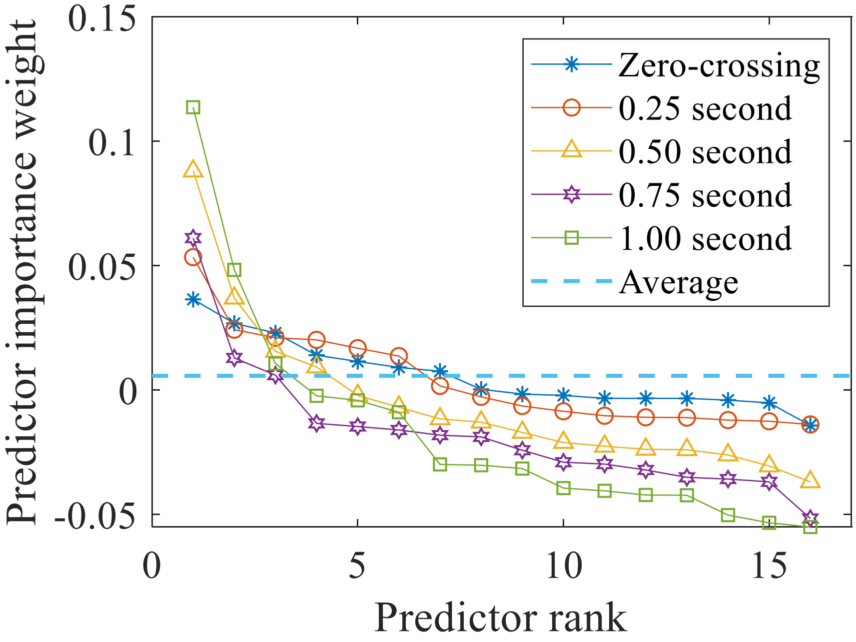


Figure 4-6. Feature importance weight

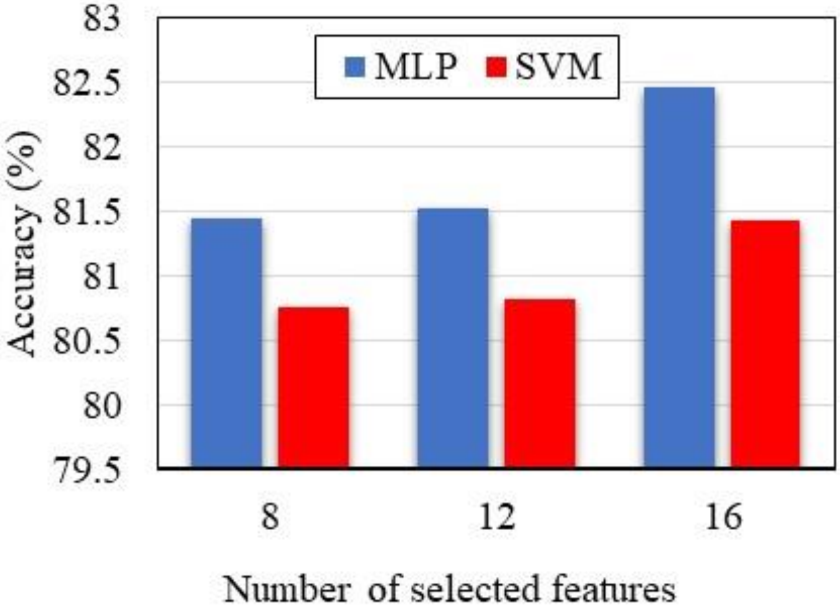


Figure 4-7. Average accuracy of classification with selected features

Figure 4-8 shows the overall accuracy for motion classification with all of the extracted features. The highest accuracy was 86.7% from MPL with zero-crossing window and the lowest accuracy was 76.5% from the SVM with 3-s window size. As can be seen, the use of two classifiers tends to increase classification accuracy with the larger window size until the 1-s window size. When the window size is larger than the 1-s window size, the classification accuracy decreases. This trend is highly related to the average length of the sliding motions, 0.67s. A too large window size is undesirable.

The accuracy with a zero-crossing window was better than other fixed length windows. Unlike other repetitive human activities with relatively consistent length, the randomness of stick-slip motion hinders characterizing each sticking and sliding motions using the fixed length windows. The zero-crossing window can be considered as a dynamic length window model as suggested by Laguna et al. (2011). The dynamic length window model generates different window segment length by actions and achieves better classification accuracy. Using the dynamic length window model, the characteristic of dynamic behavior of stick-slip motions of a non-constrained smartphone can be represented better than with fixed length windows.

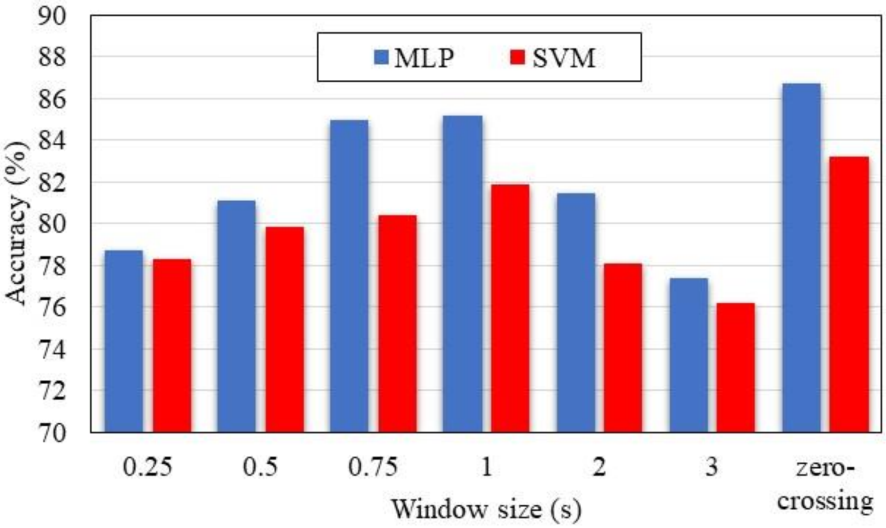


Figure 4-8. Accuracy of classification with different window size using all features

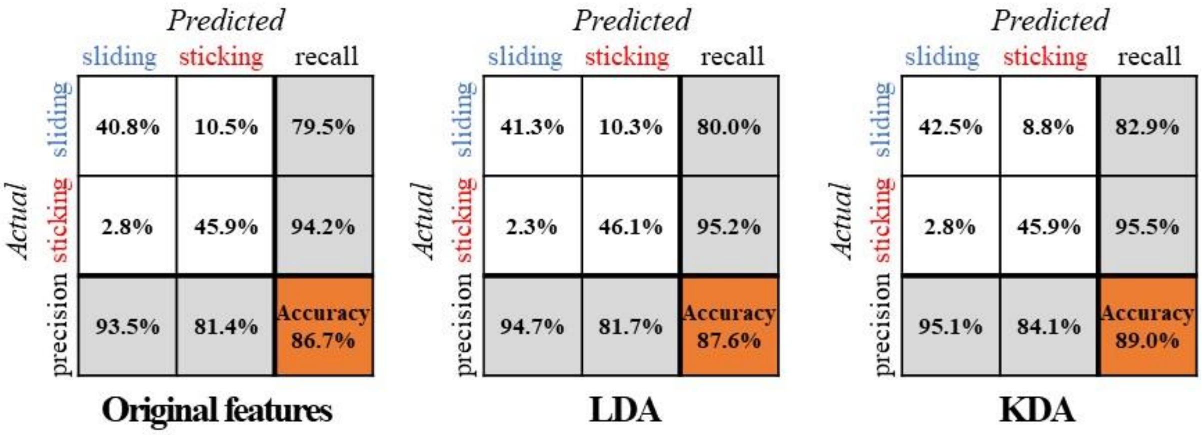
4.4.2 Effect of discriminant analysis on classification accuracy

To demonstrate the effect of dimension reduction methods on classification results, confusion matrices from three features using MLP with a zero-crossing window are presented in Figure 4-9. In the confusion matrix, each row represents actual classes and each column corresponds to predicted classes. Precision is the ratio of the number of correct predictions to the total number of instances classified as positive. Recall is the ratio of the number of correct predictions to the total number of positive instances.

The results show that the classification with features produced by KDA outperformed that with the original features and with features produced by LDA in improving class separation as shown in the Figure 4-9. Features from KDA provided an average sliding motion classification rate of up to 89.0% compared to 87.6% obtained with LDA and 86.7% obtained with the original features. These results imply that the original features are relatively plausible separations and the discriminant analysis results in a modest improvement in classification accuracy.

It is worth noting that the main purpose of this study is to detect the sliding motions from the acceleration measurements in order to remove them to facilitate a more accurate displacement response computation. From this perspective, the most important instance in the confusion matrix is the sliding instances falsely classified as sticking motions. These instances can result in critical errors in the estimated displacements. For these instances, the classification results from the KDA method is 8.8% compared to 10.5% from the original features and 10.3% from the LDA method. This result shows that KDA method is the best method not only in the total classification accuracy, but also in the reduction of falsely detected rates for sliding instances.

Figure 4-10 shows the scatter plot of the sliding and sticking motions. Figures a)-d) are the top four features of the selected 16 features identified using the ReliefF algorithm. The results for LDA and KDA are given in e) and f), respectively. To compare the separability of each feature, the normal distributions of each features are presented with normalizing feature values in terms of its maximum value as shown in Figure 4-11. KDAs show the largest difference mean value, μ, between the two motions compared to other features.



		Predicted		
		sliding	sticking	recall
Actual	sliding	40.8%	10.5%	79.5%
	sticking	2.8%	45.9%	94.2%
	precision	93.5%	81.4%	Accuracy 86.7%

Original features

		Predicted		
		sliding	sticking	recall
Actual	sliding	41.3%	10.3%	80.0%
	sticking	2.3%	46.1%	95.2%
	precision	94.7%	81.7%	Accuracy 87.6%

LDA

		Predicted		
		sliding	sticking	recall
Actual	sliding	42.5%	8.8%	82.9%
	sticking	2.8%	45.9%	95.5%
	precision	95.1%	84.1%	Accuracy 89.0%

KDA

Figure 4-9. Effect of feature reduction by LDA and KDA

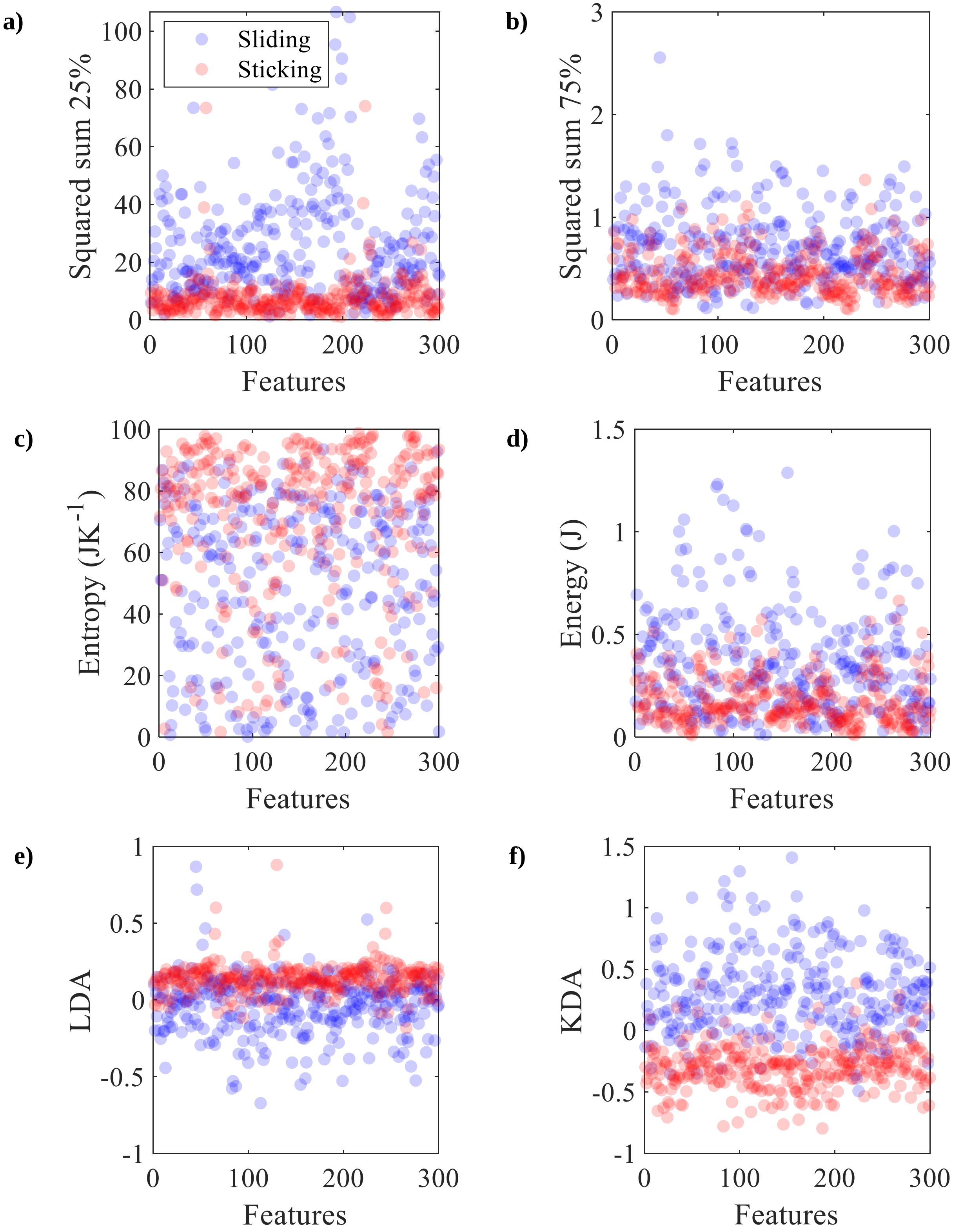


Figure 4-10. Features a)-d) top four most correlated features, e) LDA, and f) KDA

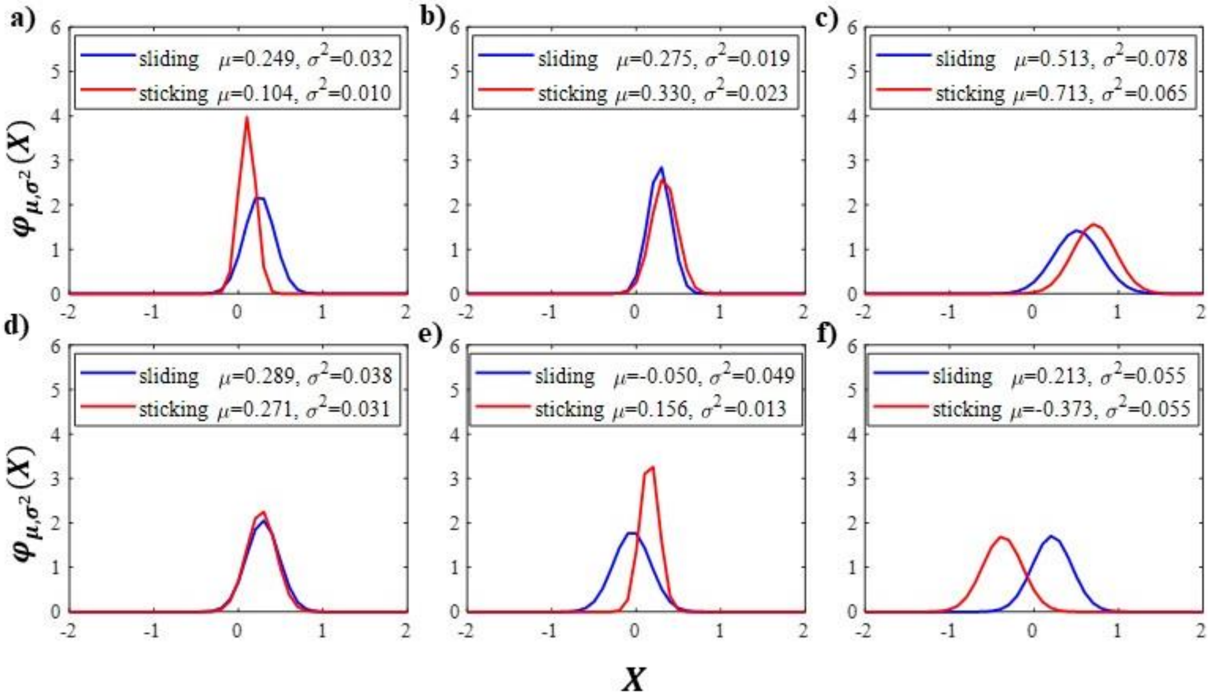


Figure 4-11. Normal distribution of features a)-d) top four most correlated features, e) LDA, and f) KDA

4.4.3 Parametric study of RNN

The evaluation on the correlation of all the hyperparameters is a laborious process due to large number of hyperparameters and lengthy computational time. To deal with these difficulties, the effectiveness of each hyperparameter is determined by varying each hyperparameter based on the empirically obtained best model with the highest classification accuracy. The results for three hyperparameters are presented in Table 4-3. In the best model, the train and test classification rate were 93.45% and 93.86% at maximum, respectively.

Table 4-3. Selected hyperparameters for the best performing classification model

Hyperparameters	Values
Number of internal layers	2
Number of units in one layer	40
Gradient clipping parameter	c=7

To determine the effect of each parameter on classification accuracy and find the best combination for three hyperparameters, only one parameter is changed while holding the other two fixed. The effect of the number of internal layers is presented using 1,2,3, and 4 layers as shown in Figure 4-12. As can be seen in the figure, the classification accuracy is the highest when two layers are used for both training and test data. The classification accuracy increased as the number of layers was increased until two. However, the classification accuracy decreased in the three- and four- layers model. This occurrence of this phenomenon can be explained because of increasing learning difficulty associated with the additional layers and overfitting when the freedom of the model becomes too high. A further weakness associated with additional layers is that the computation time increases in proportion to the increase of internal layers. The computation times were 480s for the single layer, 592s for the two layers, 667s for the three layers, and 842s for the four layers, respectively. In this study, the best model is based on only the highest classification accuracy, but in the future, it is reasonable to consider both computational time and classification accuracy to select the best model. From the above results, a redundant increase in the layers in the model may lead to a degradation in the generalization performance.

The effect of the number of internal layer units is presented using 20, 30, 40, and 50 units as shown in Figure 4-13. As shown in the figure, the classification accuracy is the highest when 40 units are used in train and test data. When more than 40 units are used, the classification accuracy decreases. The reasons for this are the same as those for the number of internal layers. To obtain a high generalization performance, the number of units should not be excessively increased.

The effect of the gradient clipping parameter is presented using 3, 5, 7, and 9 as shown in Figure 4-14. In the figure, it can be seen that the recognition rate is highest in the case of c=7 in train and test data. The lack of improvement may imply that the clipping parameter is not a sensitive parameter or that the gradient explosion was absent in this dataset

The above results show that, in the task of stick-slip classification, a modest difference appeared in the recognition rate for the three selected hyperparameters. The most difference occurred for the number of internal layer units, a difference of 3.21% between the maximum at 40 units and minimum at 20 units classification accuracy.

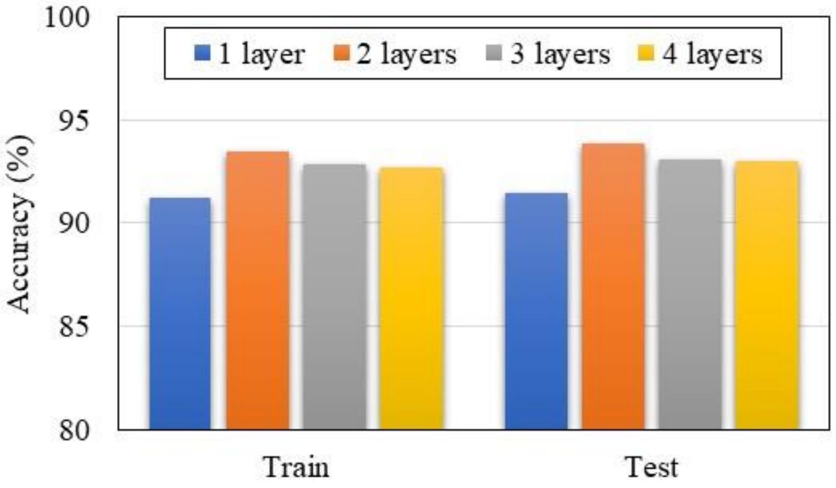


Figure 4-12. Comparison of classification accuracy according to the number of internal layers (50 units, c=7)

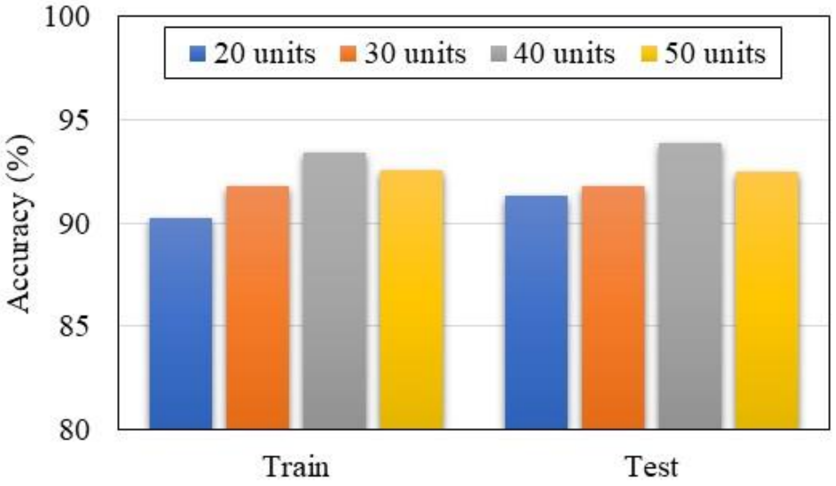


Figure 4-13. Comparison of classification accuracy according to the number of units (2 layers, c=7)

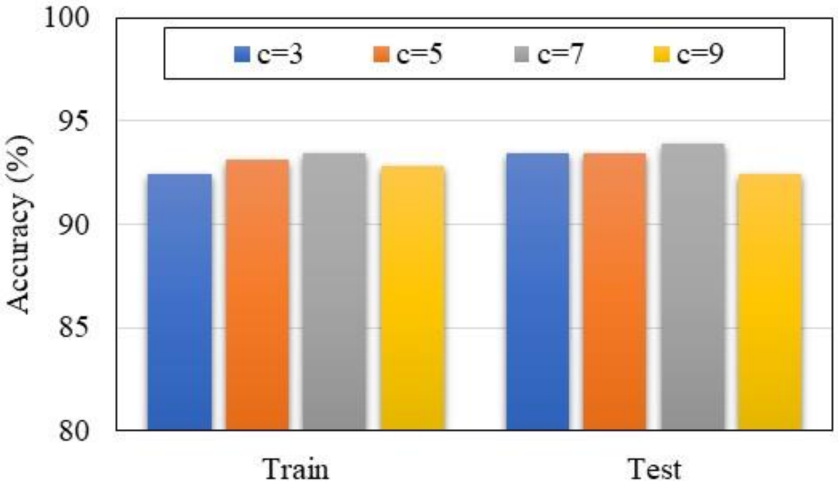


Figure 4-14. Comparison of classification accuracy according to the gradient clipping parameter (2 layers, 50 units)

4.4.4 Comparison classifiers

Compared to other supervised learning algorithms, the deep learning algorithm shows better performance for motion classification as shown in the Figure 4-15. The classification accuracies for each classifier were 93.2% for RNN, 89.0% for MLP, and 86.1% for SVM. Theoretically, in MLP, the input units in a fully connected network have the same connection and would not change the performance of the network by swapping their positions. This lack of invariable connection is not a suitable property for classification on raw data, because it ignores the dependencies between adjacent units. However, the RNN algorithm as a sparse neural network can extract the information in adjacent units.

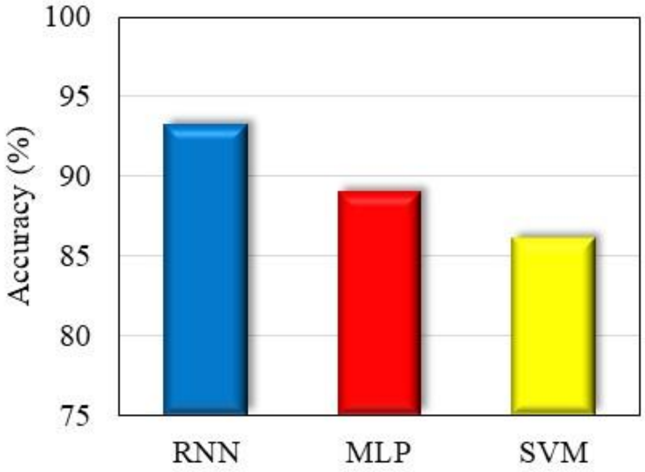


Figure 4-15. Classification Comparison

4.5 Summary and conclusion

This chapter presents an accurate and robust accelerometer-based stick-slip motion classification framework based on the machine learning method. RNN for the deep learning algorithm showed maximum classification with 93.2% compared to two other supervised learning techniques, 89.0% for MLP, and 86.1% for SVM. Three methods are investigated for the supervised learning algorithm: 1) the ReliefF algorithm was used to select highly correlated features among the potential features; 2) the zero-crossing window were selected as the best performing segmentation method for raw acceleration data; and 3) LDA and KDA were used to improve the class separability and showed a modest improvement in classification accuracy. For the deep learning algorithms, detailed examination of a combination of three hyperparameters of the proposed RNN and its performance is considered important for further improvement of the accuracy of this network. Two internal layers, 40 units for each internal layer and c=7 gradient clipping parameter were selected.

CHAPTER 5

Stacking multiple device measurements to improve computation of interstory drift ratios

5.1 General

This chapter addresses means by which to mitigate the unreliability of individual acceleration measurements from smartphones by stacking readings from multiple devices. Some of the causes of unreliability include: 1) smartphones can slide under a given ground motion when the friction between the smartphone and the underlying surface is overcome; and 2) the inherent errors in acceleration measurements make it difficult to accurately compute displacements from a single acceleration record. Successfully stacking acceleration measurements from multiple devices requires noise reduction, sliding detection, and data fusion, methods for which are introduced and discussed. Techniques for accurately double integrating the resulting data and computing inter story drift ratios (IDR) are also introduced. Each of the main ideas presented is assessed and validated by shake table experiments.

5.2 Proposed idr estimation process

A process for IDR estimation from smartphones measurements is proposed in this section. Figure 5-1 shows the 6 steps needed to get from measured accelerations to IDR estimation. These steps are:

STEP 1: Remove noise. The noise associated with measurements of the smartphone is reduced without removing the signals associated with the motion of the underlying surface.

STEP 2: Synchronize measurements. Multiple measurements are synchronized in the same time domain in order combine their values in a meaningful manner.

STEP 3: Remove sliding motions. Smartphone sliding is detected. Only data from devices that stick to the underlying surface can be used for computation of IDR.

STEP 4: Combine filtered measurements. Sticking-only-acceleration responses are aggregated by a data fusion method, Kalman filter (KF), to produce more accurate acceleration responses.

STEP 5: Estimate displacement for each floor. With merged acceleration records, estimated floor displacements are obtained by double integration and zero velocity update (ZUPT) method.

STEP 6: Calculate IDR. The estimated consecutive story displacements produce IDRs that represent building damage after an earthquake.

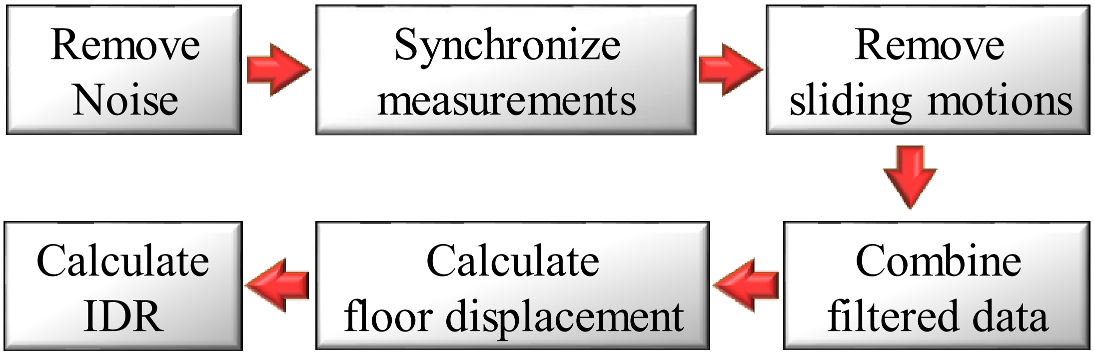


Figure 5-1. Flowchart of IDR estimation

5.2.1 Denoising

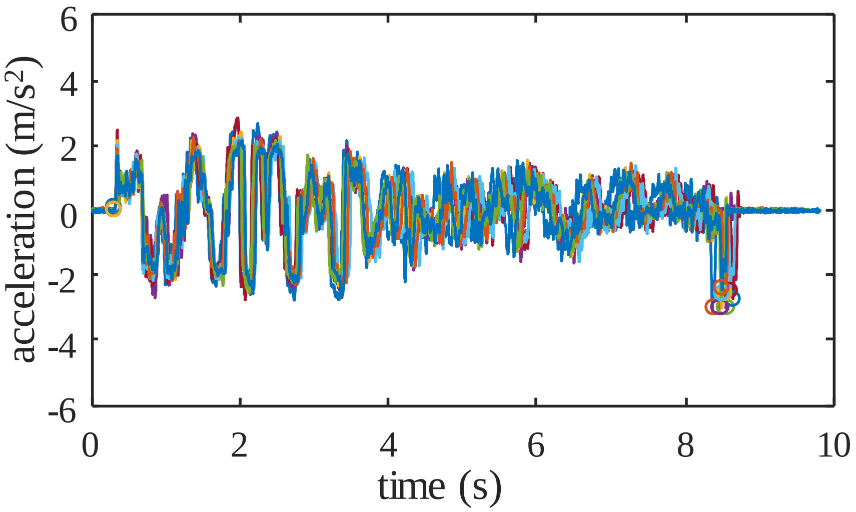
Noise reduction is an essential process in most measurement systems. However, many noise reduction algorithms also tend to degrade the original signals. Denoising is commonly done using band-pass filters for seismic applications (Amini et al. 1987); but that can also detract from the non-smooth features of a signal. Wavelet transform (WT) can overcome this drawback and makes it possible to preserve these features and extract much of the noise. If the wavelet coefficients that characterize the noise can be classified, it is possible to remove the noise and reconstruct a clean signal. In Chapter 3, it was shown that the maximal overlap discrete wavelet packet transform (MODWPT) exhibited superior performance with respect to noise reduction in the measured acceleration records of smartphones that undergo sticking-sliding motion over other filters such as a bandpass filter and the maximal overlap discrete wavelet transform (MODWT). This approach is adopted herein.

5.2.2 Synchronization

Prior to combining multiple measured accelerations into a single estimated acceleration for each floor, it is essential to synchronize them in the time domain. One research group used 25 smartphones at the same location to improve the signal-to-noise ratio by stacking them (Kong et al. 2018). They assumed that recorded noise could be removed by stacking across different measurements because noise from different smartphones is truly random and less related to the building's response. The cross correlation of measured accelerations between phone recordings were calculated by shifting them within 120s windows to find the maximum correlation coefficient. They concluded that stacking more phones does not always lead to improvement in the accuracy of combined records and stacking seven phones provide the best results when the correlation coefficient exceeds 0.6. However, misalignment problems still existed even after applying the cross-correlation method they used.

Figure 5-2a shows the result of stacking multiple acceleration measurements with the best cross correlation. As can be seen, the shaking motions have slightly different time durations. To adjust for these different shaking duration times, measurements are stretched or shortened uniformly for each measurement compared to a benchmark measurement specifically selected to have the highest correlation value among all the measurements. Figure 5-2b shows that this scaled scheme is not enough to adequately synchronize multiple measurements. The poor alignment of multiple acceleration measurements are particularly clear at 1.7-3s. Therefore, it is necessary to explore an alternative technique to enhance the performance of the method used to align multiple measurements.

a)



b)

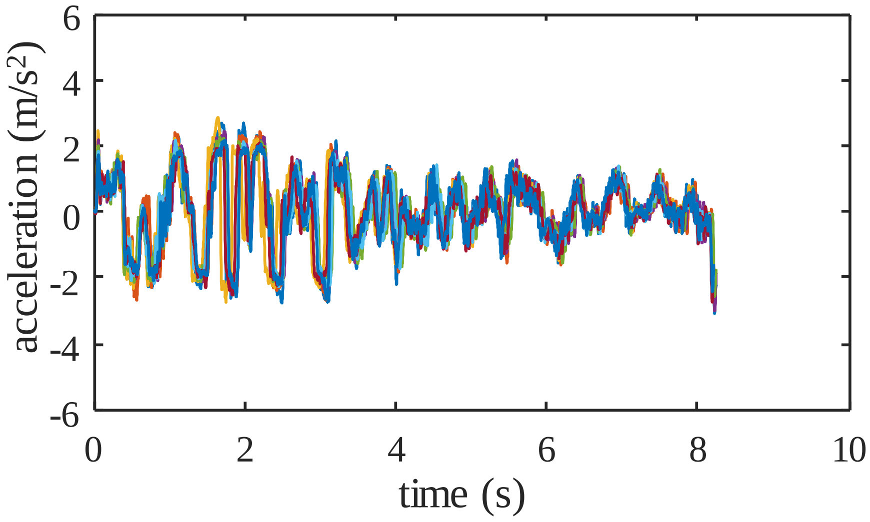


Figure 5-2. Results of stacking multiple acceleration measurements by cross correlation for a) before adjusting the scale and b) after adjusting the scale. Note the poor alignment.

Some researchers have claimed that it is possible that the sensors temporarily freeze or fail to properly collect and store data from fractions of a second to a few seconds and in return, compensate for the missing data points by collecting data at a rate higher than the assigned frequency (Akhavian and Behzadan 2016). In such cases, a preprocessing technique to fill in for missing data points and removing redundant ones can help ensure a continuous and orderly dataset. Imperfect repeatability of simulation in the shake table or technical limitations on the accelerometers could also cause this misalignment problem. Figure 5-3 shows three examples of sampling points from a Galaxy S7 smart device that demonstrate that an exact sampling rate of 200Hz is not achieved.

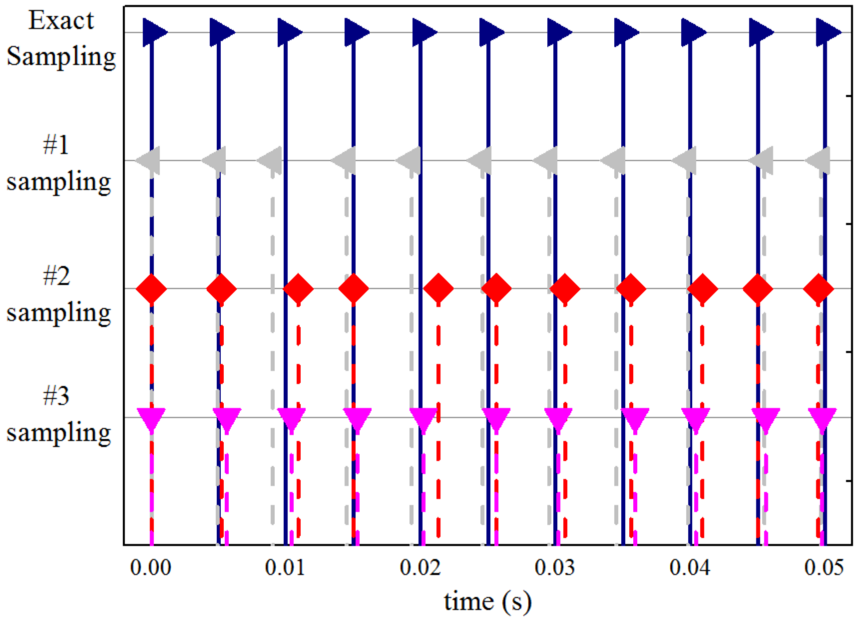


Figure 5-3. Sampling points from Galaxy S7 for three different measurements

Dynamic time warping (DTW) (Bellman and Kalaba 1959, Sakoe and Chiba 1978) can be a possible method to solve misalignment. DTW is a well-known technique to find an optimal alignment between two given sequences (time-dependent). It minimizes the effects of time lagging and warping in the time domain by allowing elastic transformation of time series to detect similar features in different phases. The alignment path built by DTW has certain restrictions and rules as follows: 1) every index from the first sequence must be matched with one or more indices from the other sequence; 2) the first and last index from the first sequence must be matched with the first and last index from the other sequence, respectively; and 3) the mapping of the indices from the first sequence to indices from the other sequence must be monotonically increasing. The distance matrix between two signals is calculated by the

Euclidean distance measure. A dynamic programming algorithm (Bellman and Kalaba 1959) is employed to determine the minimal distance matrix between sequences.

Even though the DTW algorithm is useful to align two signals that are similar in the time axis, the algorithm has a problem when the two sequences differ in the amplitude axis (Keogh and Pazzani 2001). To overcome these drawbacks, the derivative dynamic time warping (DDTW) algorithm is used to yield more accurate alignment results (Keogh and Pazzani 2001). The distance measured between two signals in DDTW is calculated by the square of the difference of the estimated derivatives of the two signals, i.e. it is not Euclidean. However, DDTW still shows poor alignment results with noisy acceleration measurements from smartphones. To resolve this problem, two new simple methods are proposed that add and element-wise multiply two normalized distance matrices calculated in Euclidean space (DTW) and the first derivative of signals (DDTW), respectively. These proposed methods are denoted additive dynamic time warping (ADTW) and multiplicative dynamic time warping (MDTW), respectively.

To compare the alignment performances of these methods, two examples are presented. The first example aligns two artificial signals of the same length, one is a clear sinusoidal waveform and the other is similar to the former but has white Gaussian noise added. In order to evaluate the performance of the methods, the amount of warping, W, defined as the ratio of the length of the warping path, K, to the sum of the length of each signal, n and m, respectively, are compared as follows:

$$W = K/(n+m) \tag{5-1}$$

The more warping there is, the larger the value of W. Figure 5-4 shows warping paths on the distance matrix and alignment results of DTW, DDTW, ADTW and MDTW with two artificial signals. MDTW has the smallest W value compared to the three other methods. Since these signals are the same length except for some perturbations, having a straight diagonal line along the warping path on the distance matrix is an ideal result.

The second example aligns two measured accelerations of different length, one is measured by Optotrak and the other is measured by Samsung Galaxy S7. In this case, the root mean square error (RMSE) of the two accelerations are compared to evaluate the performance of the four methods. It is expected that more accurate alignment will lead to a smaller value of RMSE. Figure 5-5 shows warping paths on the distance matrix and the alignment results of DTW, DDTW, ADTW and MDTW for both Optotrak and Galaxy S7 measurements. Since these signals have different lengths due to an uneven sampling rates, the warping path on the distance matrix is not necessarily a straight line, unlike the previous case. The results indicate that MDTW has the smallest RMSE value compared to the three other methods.

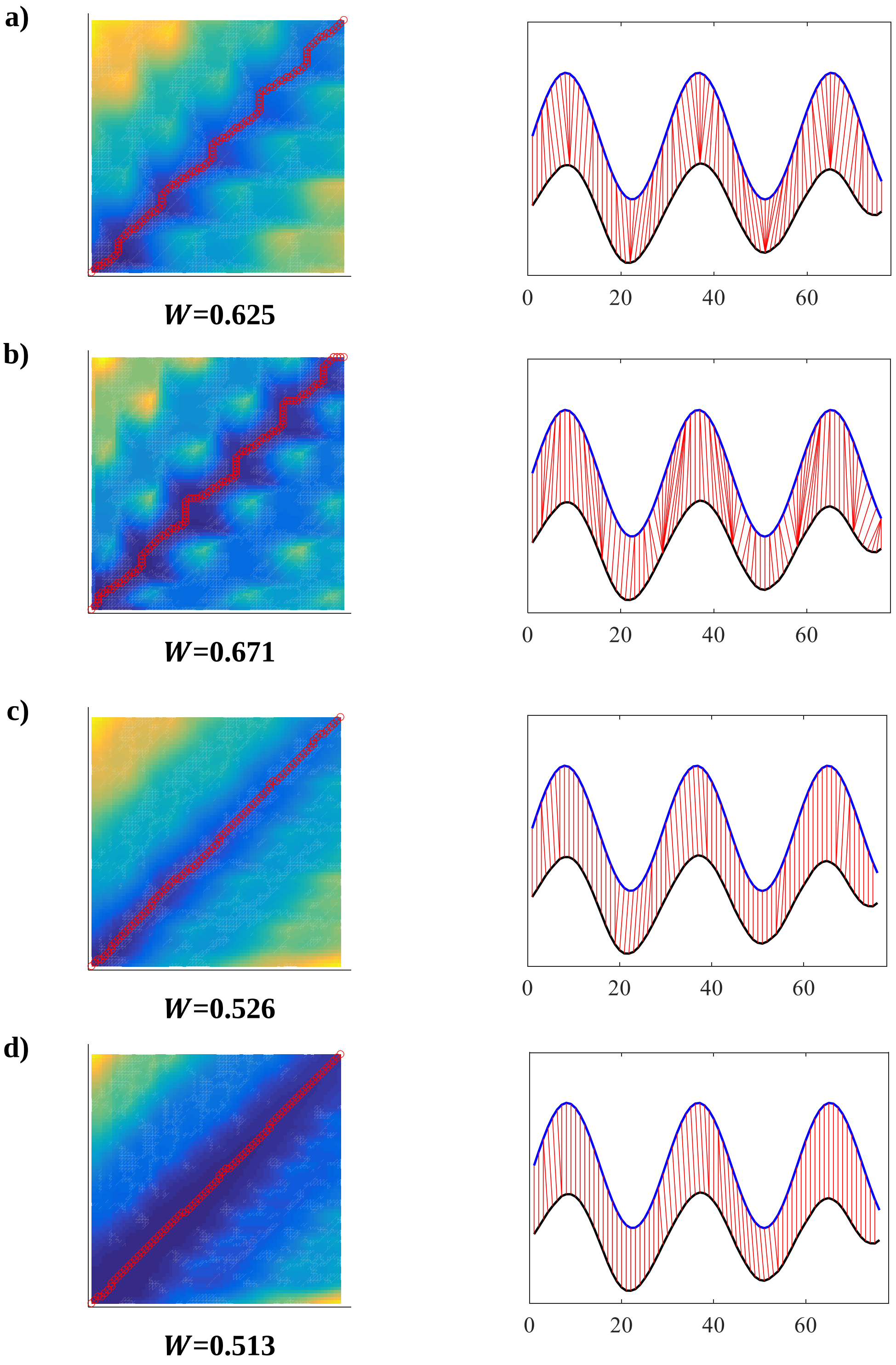


Figure 5-4. Comparison of warping path and alignment for two artificial signals. The lines between two signals in the left columns indicate alignment results for a) DTW, b) DDTW, c) ADTW, and d) MDTW

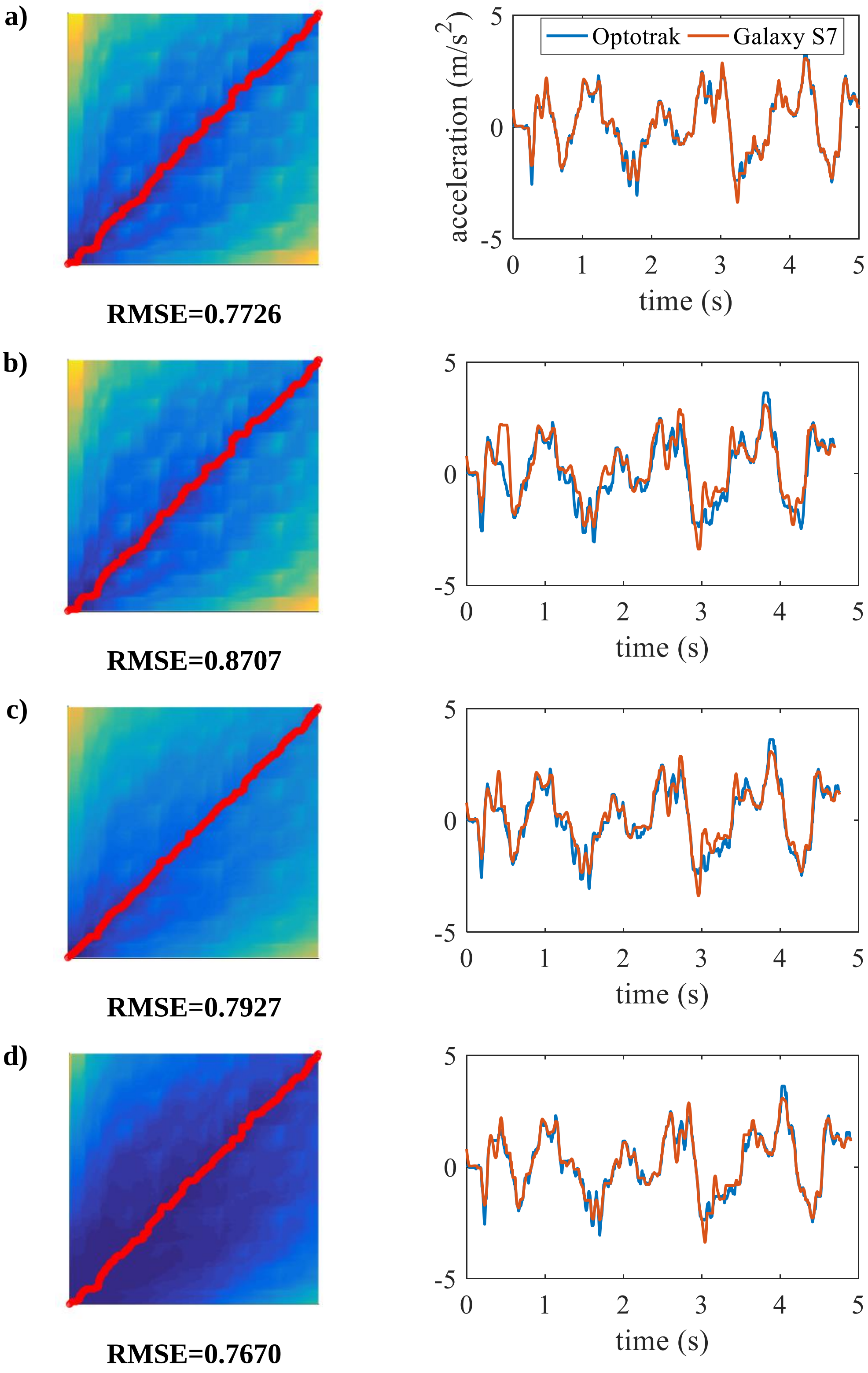


Figure 5-5. Comparison of warping path and alignment for Optotrak measurement and Galaxy S7 measurement for a) DTW, b) DDTW, c) ADTW, and d) MDTW

These two examples show that MDTW outperforms other methods in terms of the warping path for artificial signals and RMSE value for the acceleration measurement. Figure 5-6 shows comparisons of the correlation matrices for five acceleration measurements for the same ground motion without and with MDTW. It is clear that MDTW improves the correlation values between the measurements and is adopted in further work in this chapter.

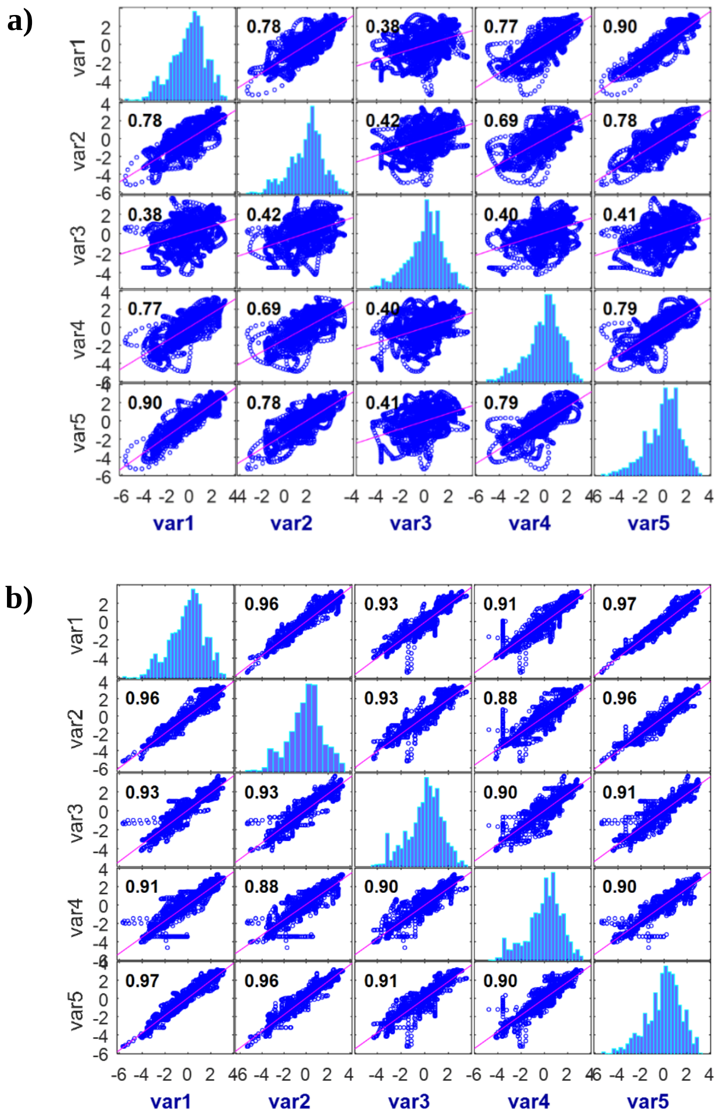


Figure 5-6. Comparison of correlation matrix with multiple acceleration measurements for a) without MDTW and b) with MDTW. Var1-5 indicate the sample number for each acceleration measurement under the same ground motion

5.2.3 Sliding motion detection

Sliding of smartphones on their underlying surface impedes their use for assessing building seismic damage. These sliding motions occur under extreme movement and become prominent when the coefficient of friction between the smartphone and its underlying surface is small. The method presented in Chapter 2 for detecting the occurrence of sliding motions in the recorded acceleration response is adopted in this study. That method requires knowledge of the KCOF. For that, the KCOF estimation method introduced in Chapter 3 is employed.

5.2.4 Interpolation method

It is feasible that some floors may not have good estimates of their movement, for example, due to too few smartphones or if all the smartphones slide during the earthquake in a certain floor. In this case, it is reasonable to dispose of all the acceleration measurements in that floor and indirectly estimate the displacement based on the estimated displacement in adjacent floors. Conventionally, motions of the non-instrumented floors are estimated by an interpolation procedure (De la Llera and Chopra 1995, Naeim 1997). Typically, a piece-wise cubic polynomial interpolation (PWCPI) procedure is used for conventional buildings. It is generally believed that the PWCPI procedure provides reasonable estimates of motions at non-instrumented floors. A convenient implementation of the PWCPI procedure is possible in MATLAB (MathWorks, 2006) with the use of “spline” function, which is not-a-knot spline method. The usefulness of interpolation method will be discussed with experimental results in Section 5.3.7.

5.2.5 Data fusion method

As mentioned in Chapter 3, the inherent errors in acceleration measurements in smartphone can be classified into two types: deterministic (systematic) and stochastic (random) (Nassar 2003).

Although major deterministic error sources (bias and scale factor errors) could be removed by using the static multi-position method (El-Diasty and Pagiatakis 2008), in reality, it is impractical to apply this method to private smartphones before or after an earthquake occurs because it requires the owner's active participation. Scale factor errors are discussed later on in the chapter. Stochastic errors, which are a key concern in this chapter, are addressed by aggregating the results of multiple smartphones. It is assumed that noise recorded by the same smartphone is truly stochastic and that stacking multiple measurements can alleviate that noise as explored later on.

To stack multiple measurements, data fusion is used to estimate IDR for each floor of a building. The advantage of fusing multiple streams of data is that it cancels out the random noise in measurements and produces more consistent and accurate acceleration responses than those provided by measurements from individual smartphones. Kalman filter (KF) is a well-known data fusion method that has beneficial recursive properties, low computational requirements, and is an optimal estimator for one-dimensional linear systems with Gaussian error statistics (Kalman 1960). KF has a prediction step (Equations 5-2 and 5-3) and an update step (Equations 5-4 through 5-6).

Prediction

$$\hat{x}_{t|t-1} = F_t\hat{x}_{t-1|t-1} = 1 \times \hat{x}_{t-1|t-1} = \hat{x}_{t-1|t-1} \tag{5-2}$$

$$P_{t|t-1} = F_tP_{t-1|t-1}F_t^T + Q_t = P_{t-1|t-1} + Q_t \tag{5-3}$$

Update

$$K_t = \left(P_{t|t-1}H_t^T\right)\left(H_tP_{t|t-1}H_t^T + R_t\right)^{-1} \tag{5-4}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t\left(z_t - H_t\hat{x}_{t|t-1}\right) = \hat{x}_{t|t-1}(1 - K_t) + K_tz_t \tag{5-5}$$

$$P_{t|t} = (I - K_tH_t)P_{t|t-1} \tag{5-6}$$

where $\hat{x}_t$ is the estimation of state vector at time t, F_t is the state transition matrix, P_t is the covariance matrix of the state error, Q_t is the covariance matrix of the process noise, H_t is the transformation matrix, R_t is the covariance matrix of the measurement noise, and z_t is the vector of measurements.

In this study, one state value is considered to combine the multiple acceleration measurements. The state of a system at a time *t* evolved from the prior state at time *t*-1 according to the Equation 5-2. Combining only acceleration measurements, it is assumed that the state-transition model is set to 1 and the current state is the same as the previous state in Equation 5-2. For sensor fusion, more than one sensor value in observation vector z_k, which are the current readings of multiple measurements, will be combined. Without any prior knowledge, it is reasonable to assume that all the measured accelerations contribute equally to acceleration estimation, so H_t matrix is a 1×n (*n* is the number of smartphones) matrix of ones. However, each measurement could have a different weight if additional information were available.

The mean square (MS) amplitude of noise can be estimated approximately from the smartphone's stationary response before or after the shaking. This is an approximation of the MS of the amplitude of noise but is a reasonable approach to calculate the weighted values for combining measurements from multiple smartphones since it gives more importance to data with less noise. Using this information, a weighted value, W_i, for each measurement can be calculated inversely proportional to the MS amplitude of noise, n_i, between different smartphones (Brennan 1959).

$$\text{weighted value based on MS amplitude of noise: } W_i = \frac{\frac{1}{n_i}}{\Sigma\frac{1}{n_i}} \qquad 5\text{-}7$$

For example, the MS amplitude of noise is 106.09, 416.16, 268.96, 338.56, and 930.25 mg for Samsung Galaxy S7, Samsung Galaxy S5, LG G5, Nexus 6P, and Nexus 5X, respectively, as shown in Table 4-1. The weighted values are therefore 0.482, 0.123, 0.190, 0.151, and 0.055 for each smartphone.

The diagonal elements of the R_t matrix are the variance of the observation of the noise signal v_t for each measurement, which is assumed to be the variance of the stationary response of each smartphone. Elements of non-diagonal terms of this matrix are assumed as zeros because they represent how much each sensor's noise varies with another's. The size of P_t is a scalar value because it represents the covariance of the single value estimate value $\hat{x}_t$ with itself. The gain K_t is a 1×n matrix, because the single state estimate $\hat{x}_t$ and *n* number of sensor observations in z_t.

While the gain can be extracted from output signals, the covariance of the state error cannot be evaluated without knowledge of the covariance of the process and measurement noise Q_t and R_t (Bulut et al. 2011). The measurement noise represents electronic and random noise characteristics of the sensor. It is calculated from the sensor accuracy, which is represented using standard deviation of measured values from true values during the calibration and summarized in Table 4-1. However, the process noise decides the accuracy and time lag in the estimated value. The higher the value of the covariance of process noise, the higher the gain, giving more weight to noisy measurements and compromising the estimation accuracy. Lower values of the covariance of process noise lead to better estimation accuracy but introduce undesired time lag in the estimated value.

Figure 5-7 shows the relation between the covariance of process noise and RMS error that is calculated between the reference signal, Optotrak measurement, and combined acceleration for four acceleration measurements from Galaxy S7. The plot does not seem like a perfect L-curve as expected for a standard regularization parameter estimation method. Nevertheless, the value of Q_t=10^{-3} and 10^{-1} could be intuitively chosen for KF with a given value of

$$R_t^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \times 10^{-3} \text{ and } R_t^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \times 10^{-1} \qquad 5\text{-}8$$

This result shows that the value of covariance of process noise greater than that of the measurement noise should be selected.

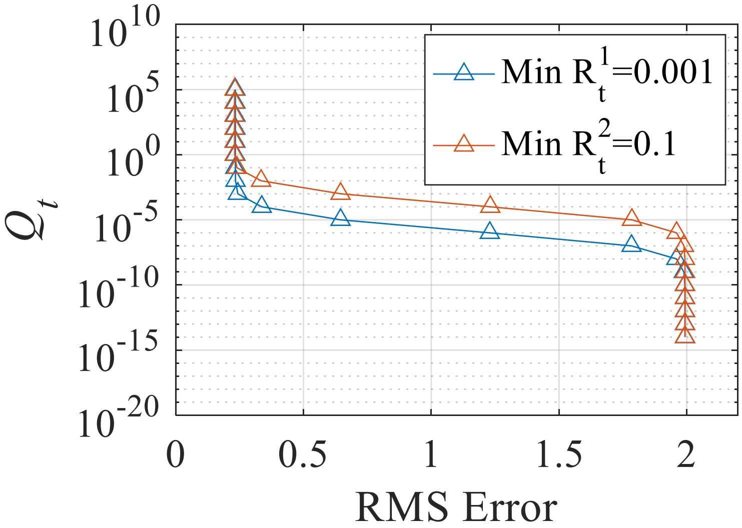


Figure 5-7. Parametric study of the covariance of process noise with four acceleration measurements

5.2.6 Double integration

The double integration error of accelerometer signals is also of importance in this work. Even when appropriate sensor error models and calibration methods are employed, it is still difficult to compute accurate displacements from measured acceleration by double integration. This is due to technical limitations of the accelerometers in smart devices and is analogous to calculating the permanent ground displacements from ground accelerations, which some reputable researchers have concluded impractical (Trifunac and Todorovska 2001). Nevertheless, many researchers have attempted to minimize the error in displacement from double integration (Ribeiro et al. 2003, Park et al. 2005). However, most of studies to date have employed fixed accelerometers (not smartphones) and do not have to contend with the effect of sliding motion of the smart device.

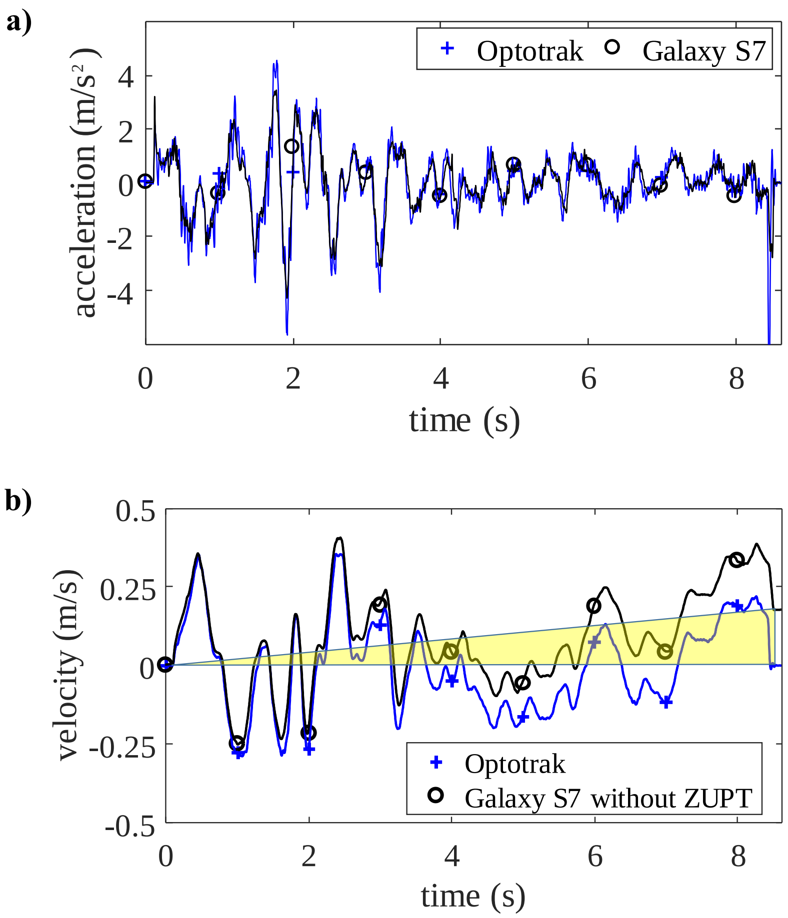
A promising technique to facilitate displacement computation is the zero velocity update (ZUPT) technique (Ibrahim et al. 2018). The method was used for determining the displacement of an in-service bridge from its free vibration acceleration. The key of this idea is that the value of velocity after shaking should be zero, a fact that is used to reduce the accumulated integration errors. Noise cancellation can be achieved by taking advantage of the fact that an earthquake's vibration intensity fades gradually and eventually stops at zero velocity and acceleration. In this case, the measured velocity at the end of shaking reflects the accumulated error in the preceding samples, which can be used to minimize the estimation error. In particular, the ZUPT method compensates for a constant drift error, which is the one of the deterministic errors.

To demonstrate the proposed double integration method with ZUPT, the selected records (FEMA 2009) of ground motions are applied to the shake table platen in Table 5-1; "DUZCE.BOL" (denoted as EQ1 in Table 5-1), KOBE/NIS (EQ2), and NORTHR/MUL (EQ3). Figure 5-8 shows the results of applying the double integration method for EQ1. Five rounds of repetitive acceleration measurements from Galaxy S7 and the reference response from Optotrak are considered in this exercise. As discussed later on, the results must be aggregated to address the stochastic errors associated with the signals.

Table 5-1. Ground motion records used

EQ ID	**PEER-NGA Record Information**		**PGA (g)**	**Arias Intensity (I_A) (m/s)**
	Name	**Record Seq. No.**		
EQ1	DUZCE/BOL	1602	0.728	3.724
EQ2	KOBE/NIS	1111	0.509	3.353
EQ3	NORTHR/LOS	960	0.410	1.913

The signals are first processed using MODWPT to reduce noise. MDTW is then used to align the records and KF is used to fuse the data as shown in Figure 5-8a. The start and end points of shaking are identified, and velocity is calculated from the acceleration record using the cumulative trapezoidal numerical integration method. The terminal velocity after shaking should be zero as theorized in ZUPT. However, as is clear, the terminal velocity of the integrated velocity is not zero because of the acceleration measurement errors (Sekiya et al. 2016). In order to make the terminal velocity equal to zero, the drift component (shaded area in Figure 5-8b) is subtracted out uniformly from the integrated velocity. Clearly, the corrected velocity is in good agreement with the reference velocity response from Optotrak, as shown in Figure 5-8c. Finally, the estimated displacement is obtained by a second integration. Figure 5-8d shows that it is in reasonably good agreement with the Optotrak reference displacement, where the maximum displacement error is 0.015m. These results show that the ZUPT method, as applied to the first integration from acceleration to velocity, effectively removes the constant drift error, which is a key deterministic error. With this error corrected, the second integration (from velocity to displacement) is typically quite accurate. This can be seen in Figure 5-9 where the maximum displacement errors (ME) in displacement estimation compared to the Optotrak measurement are 0.015 m for EQ2, 0.008 m for EQ3, respectively, and the error at peak (EP) in displacement estimation compared to the Optotrak measurements are 0.003 m for EQ2 and 0.006 m for EQ3, respectively.



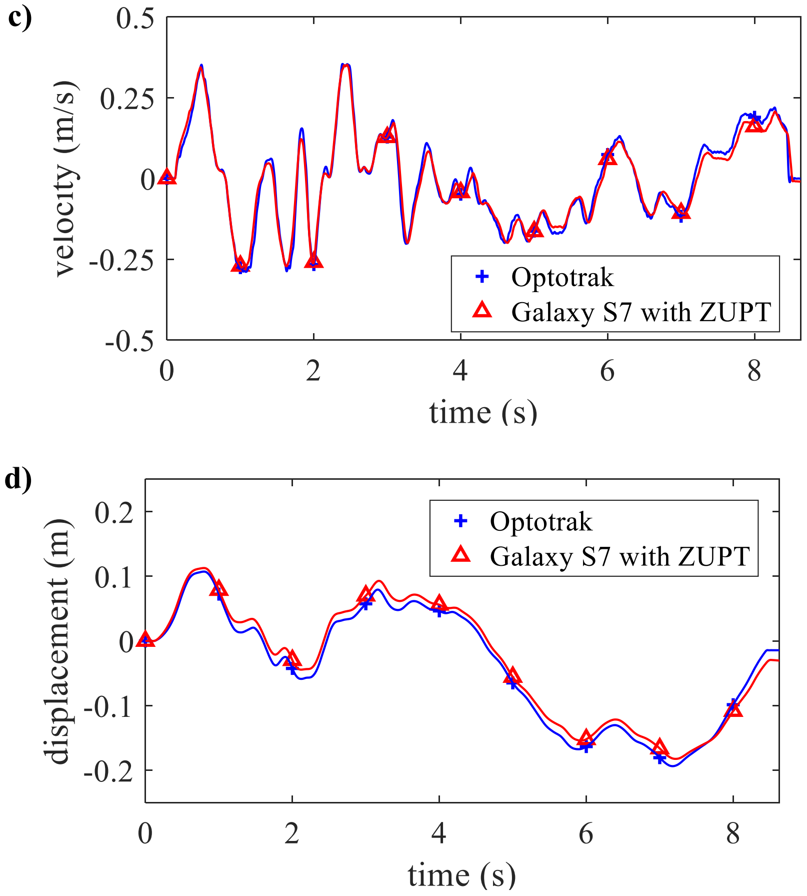


Figure 5-8. Four steps of double integration with five sets of acceleration measurements for EQ1: a) acceleration response; b) velocity response without ZUPT; c) estimated velocity response with ZUPT; and d) estimated displacement response

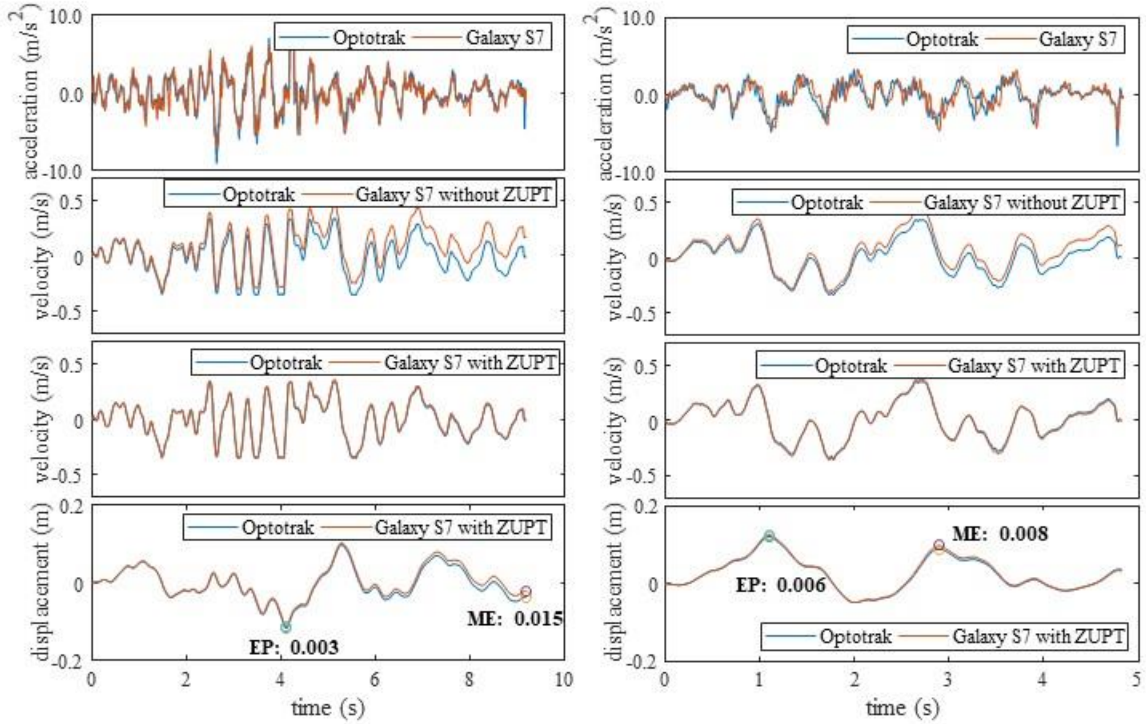


Figure 5-9. Results of double integration with ZUPT method for a) EQ2 and b) EQ3

5.3 Experimental Results

The results from shake table experiments and data from other studies are used to show that the proposed methods can reasonably estimate floor displacements and hence, the IDR of buildings, under seismic events. In the following section, three quantities are compared to show the performance of measured acceleration and displacement estimation: 1) the root mean square error (RMSE) in measured or combined acceleration compared to the Optotrak measurement; 2) the maximum error (ME) in displacement estimation compared to the Optotrak measurement; and 3) error at peak (EP) in displacement estimation compared to the Optotrak measurements.

5.3.1 Experimental setup

Experiments are conducted to simulate a smartphone’s behavior on a customized shaking surface. The overall experimental setup including the smartphone, infrared sensors, shake table, base, and Optotrak system as shown in Figure 2-8.

To study the effect of the different surfaces, the Samsung Galaxy S7 smartphone is used with three types of protective shells as shown in Figure 5-10. The measured KCOF is 0.164 for the device without its shell for Case 1. With the protective shell, the KCOF is 0.273 for Case 2, 0.488 for Case 3, and 0.508 for Case 4.

To study the effect of different types of smartphones in the following section, five different Android smartphones (with various types of onboard accelerometers) are used, as summarized in Table 4-1. It is clear from Table 4-1 that subsequent models of the same type of smartphone have significantly better accelerometers than older versions. For example, the Galaxy S7 has double the sampling frequency (200 Hz versus 100 Hz) and half the amplitude of noise (10.3 mg versus 20.4 mg).

front side

Case 1
(KCOF 0.164)

Case 2
(KCOF 0.273)

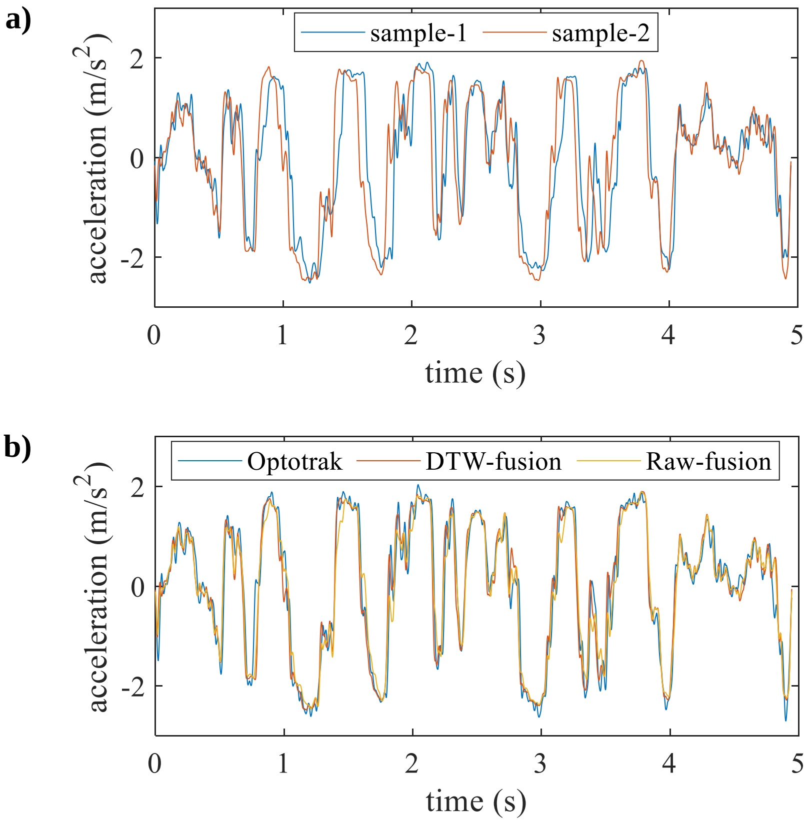
Case 3
(KCOF 0.488)

Case 4
(KCOF 0.508)

Figure 5-10. Configuration of a) Galaxy S7 and b) three types of protective shells

5.3.2 Effect of MDTW on acceleration measurements

To demonstrate the effect of MDTW on acceleration measurements, two sample measurements under the same ground motion are obtained from the Samsung Galaxy S7 as shown in Figure 5-11a. These two samples are fused into a single combined acceleration by KF with MDTW (DTW-fusion) and without MDTW (Raw-fusion). The estimated displacements for both Raw-fusion and DTW-fusion are then obtained by double integration method. The results show that Raw-fusion has a larger error in RMSE, ME, and EP than DTW-fusion (RMSE: 0.375, 0.325, ME: 0.031, 0.013, EP: 0.028, 0.012 for Raw-fusion and DTW-fusion, respectively). It is clear that the DTW method can improve the accuracy of combined measurements and estimated displacement compared to combined measurements without DTW.



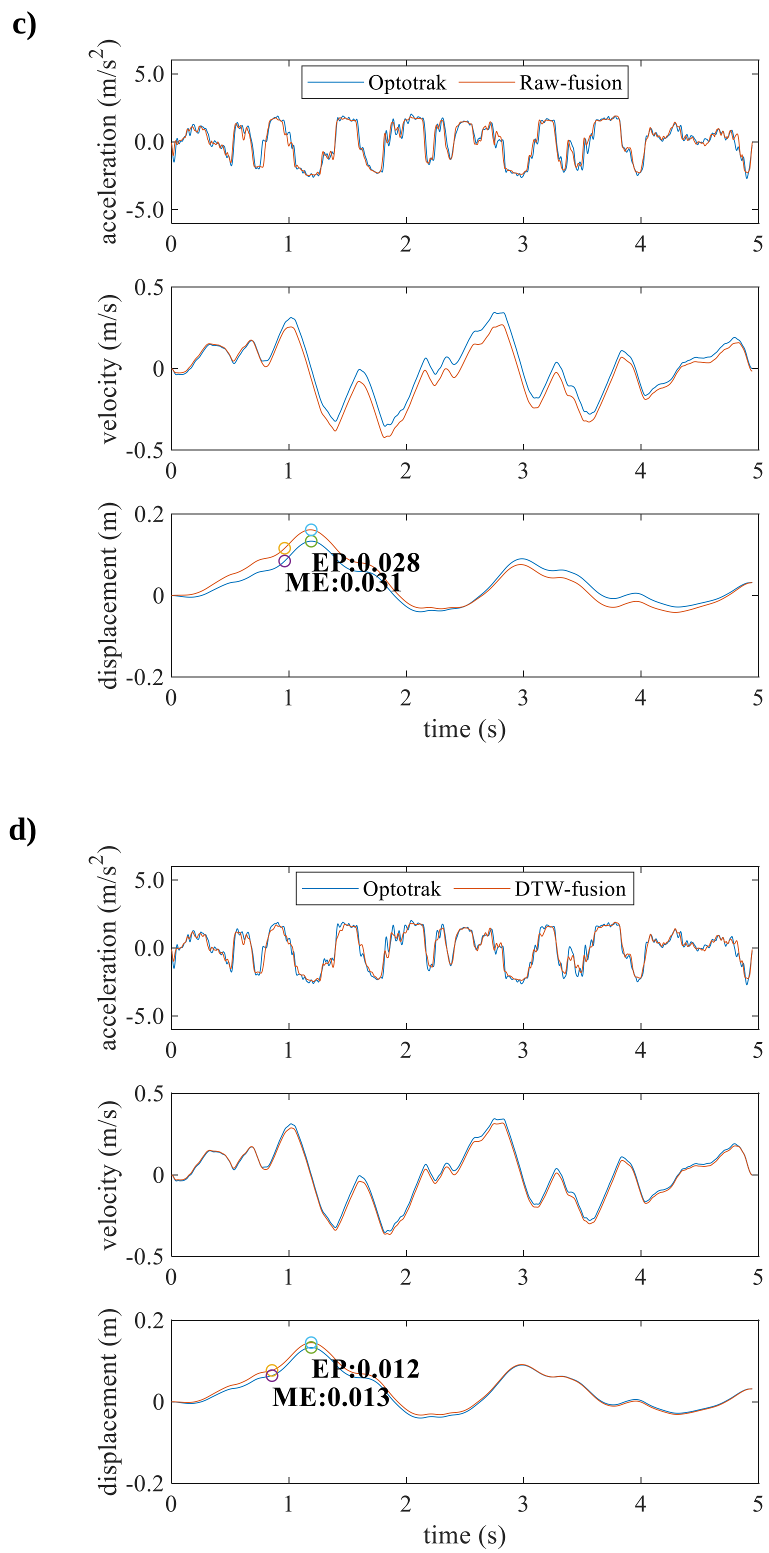


Figure 5-11. Effect of DTW on combined signals: a) Two measurements from Galaxy S7; b) combining two measurements from Galaxy S7; c) Raw-fusion; and d) DTW-fusion

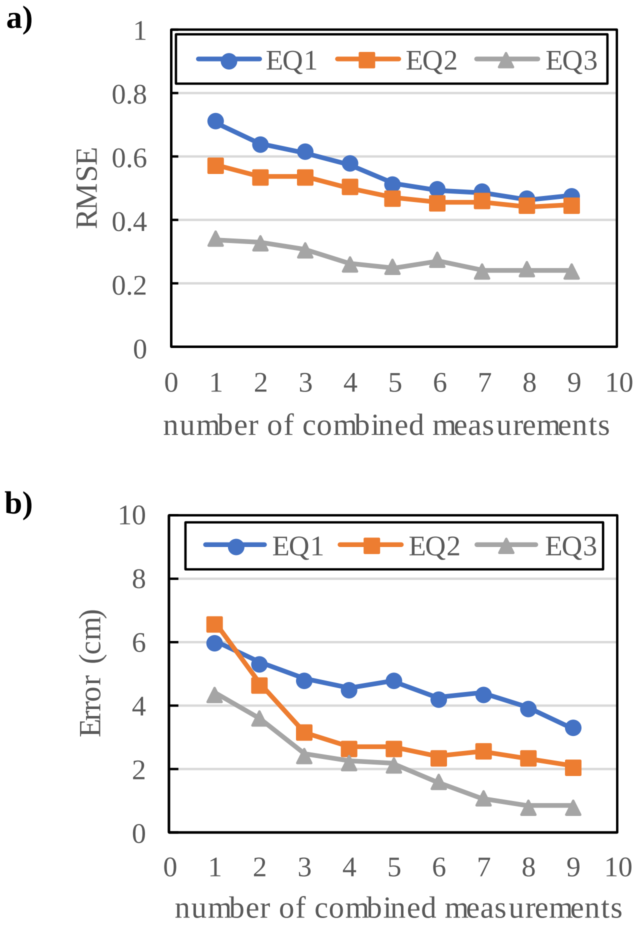
5.3.3 Effect of the number of combined measurements

The effect of the number of combined measurements on improving the accuracy of estimated displacements is investigated. First, it is assumed that all combined measurements are from the same phone to illustrate the improvement in performance without adding uncertainty due to different types of accelerometers being combined. Then, in Section 5.3.5, the performance when combining readings from different types of smartphone accelerometers is investigated.

It is reasonable to increase the number of combined measurements in a specific order to maximize the accuracy of combined measurements, i.e., more reliable measurements are combined first before adding less reliable measurements. In this study, it is assumed that the reliability of measurements corresponds to the correlation coefficient between other measurements in the same device. After obtaining repetitive measurements from Samsung Galaxy S7 under the same ground motion and applying DTW to align multiple measurements, one measurement with the highest average of correlation coefficient between the others is selected as a reference. Based on this reference measurement, the rest of the measurements are sorted by the correlation coefficient value. The combined measurements, with an increasing number of measurements, are obtained by adding the acceleration measurements following the order of the correlation coefficient.

Figure 5-12 shows the results of different numbers of combined measurements. When the number of combined measurements is increased, the RMSE, ME, and EP tend to decrease. This result shows that it is reasonable to combine more than five measurements to acquire accurate combined measurements. However, it is important to note that RMSE, ME, and EP converge asymptotically and tend to be larger as the value of peak ground acceleration (PGA) is larger

(0.728g for EQ1, 0.509g for EQ2, 0.416g for EQ3). This observation led to further investigation of the amplitude dependence of ground motions in acceleration measurement error. Figure 5-13 confirms that the RMSE, ME and EP converge asymptotically for scaled versions of EQ2 with PGA ranging from 0.201g to 0.506g versus the number of combined measurments.



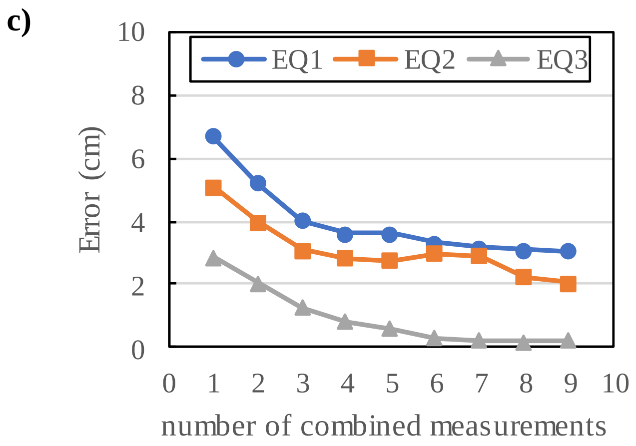
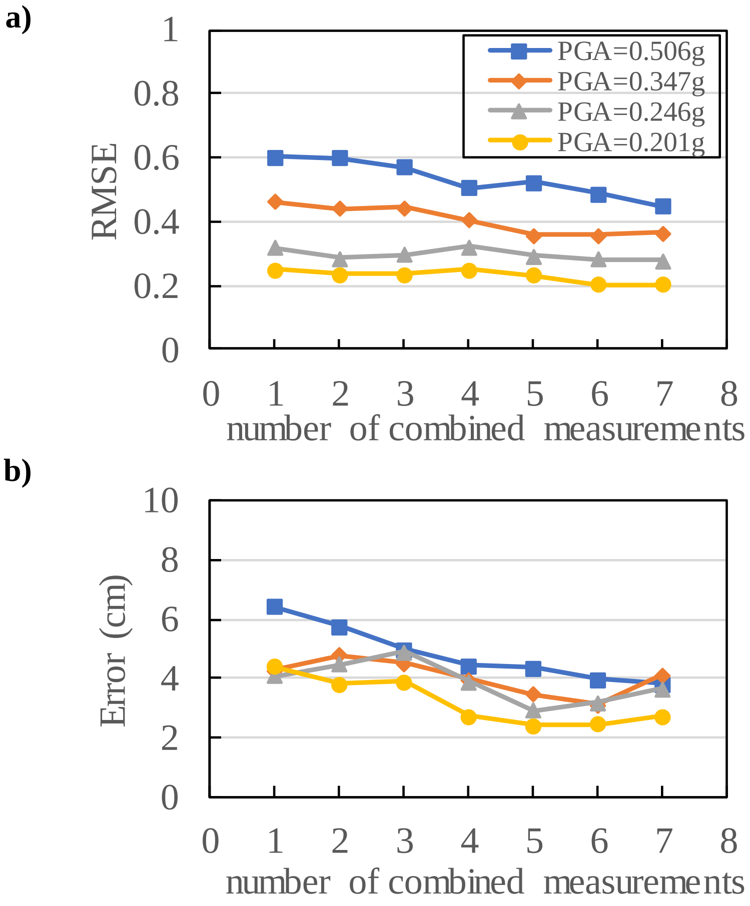


Figure 5-12. Effect of the number of combined measurements a) RMSE in acceleration, b) ME in displacement, and c) EP in displacement



c)

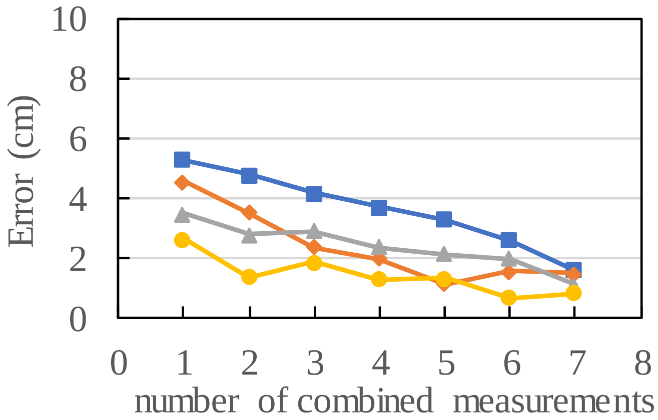


Figure 5-13. Effect of the number of combined measurements for scaled ground motion of EQ:2 a) RMSE in acceleration; b) ME in displacement; and c) EP in displacement

5.3.4 Scale factor errors

To study the effect of varying the amplitude of ground motions, three ground motions are scaled down and used as input to the shake table. PGAs of four scaled ground motions are 0.735g, 0.505g, 0.357g, and 0.292g for EQ1, 0.506g, 0.347g, 0.246g, and 0.201g for EQ2, and 0.405g, 0.319g, 0.233g, and 0.153 for EQ3. Specifically, as shown in Figure 5-14, it is clear that the RMSE, EP, and ME and the amplitude of ground motions are in a proportional relationship. Note that the results presented are after averaging 5 measurments to eleimate stochastic noise as discussed in Section 4.2. These results show that the accelerometer in a smartphone has an amplitude dependent error in acceleration measurement. This dependence is called a scale factor error (Groves 2013) and can, under certain conditions, be removed through a calibration process as mentioned in Section 3.3.1. This is not done in this study because it is impractical to achieve this with private smartphones after a seismic event.

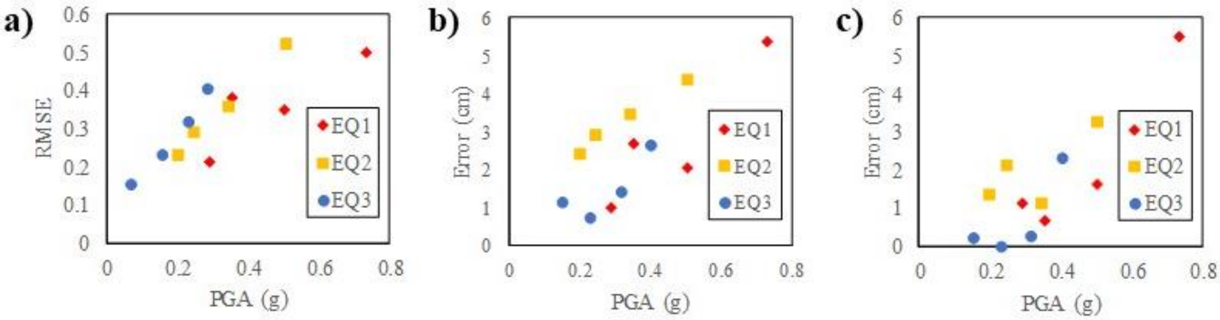


Figure 5-14. Effect of the amplitude of ground motions on combined five measurements: a) RMSE in acceleration; b) ME in displacement; and c) EP in displacement

5.3.5 Results of combining multiple readings from different types of smartphones

The result of combining the readings from the five different types of smartphones employed in this research is investigated. The reading for each smartphone type is assumed to be aggregated from five separate readings for that particular smartphone type for a total of twenty-five readings. Each smartphone has a different type of accelerometer and therefore different ability to measure acceleration accurately. To account for this uncertainty, each measurement has a weighted value that is inversely proportional to the MS amplitude of noise associated with each smartphone in Equation 5-7, as noted earlier in Section 5.2.5.

Figure 5-15 shows the estimated displacement responses and errors in displacement after combining the five smartphones with respect to the Optotrak reference for EQ1. The results for only the Samsung Galaxy S7 are also presented. Figure 5-16 shows the MEs and EPs for the combined data and those from only the Galaxy S7 for three ground motions. This figure shows that combining measurements from multiple smartphones does not always improve displacement estimation. The reason for this is because the data being combined contains poor data sets (from the older and less-capable smartphones), which is overshadowed by the better data set from the Samsung Galaxy S7. Nevertheless, the weight values for each smartphone when multiple measurements are combined produce reasonable results in displacement estimation compared to the best performing smartphone, Galaxy S7, in displacement estimation.

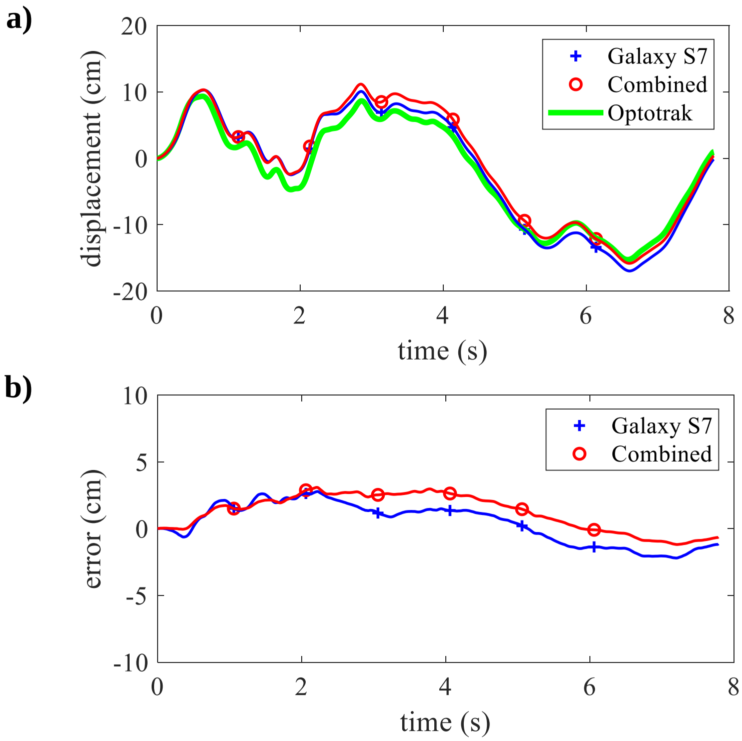


Figure 5-15. Estimated displacements for EQ1: a) estimated displacement (cm) and b) error in displacement (cm)

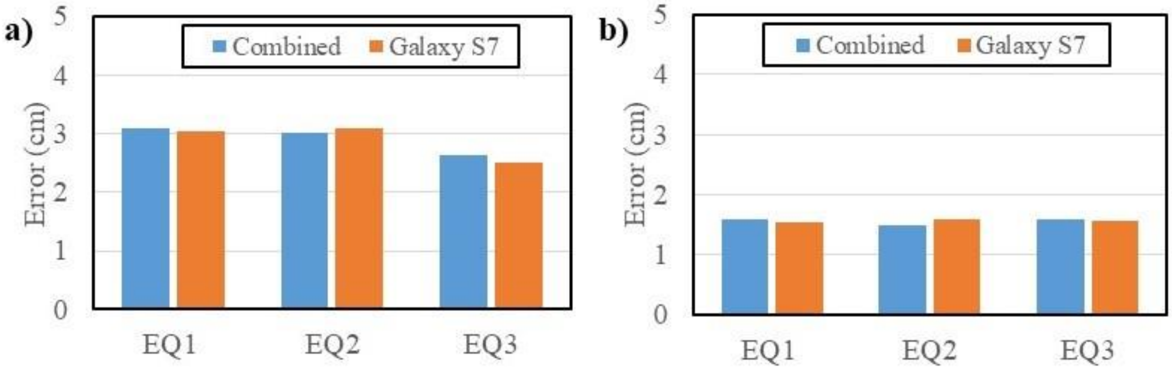


Figure 5-16. Results of displacement estimation for combined five smartphones: a) ME in displacement and b) EP in displacement

5.3.6 Effect of different surface KCOF

Figure 5-17 shows the effect of different surface KCOF in ME and EP in estimated displacements compared to Optotrak measurements. For a given ground motion, the surface with higher KCOF tends to have smaller MEs and EPs than that with a lower value of KCOF. This is because the shell with higher KCOF sticks more effectively to the underlying surface and has a smaller microslip zone.

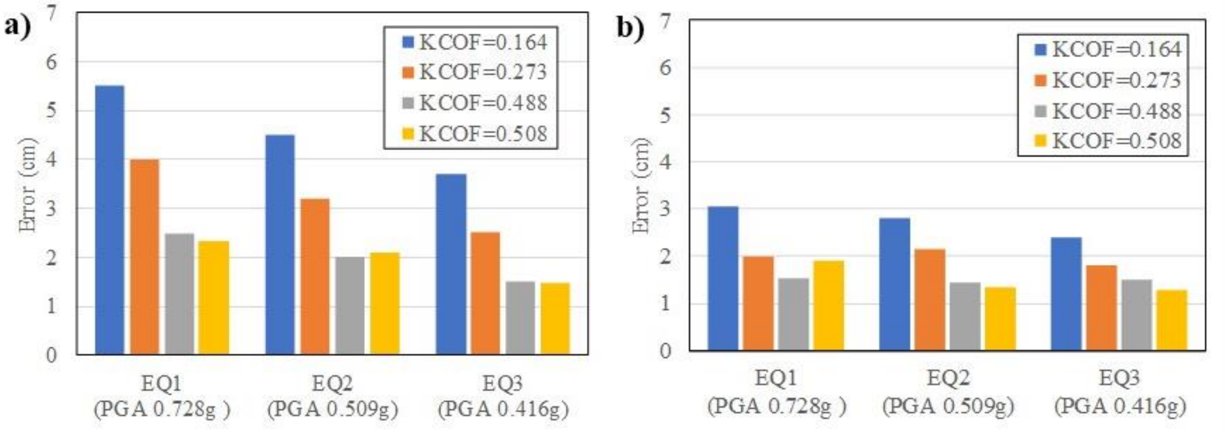


Figure 5-17. Results of displacement estimation for different types of surface: a) ME in displacement and b) EP in displacement

5.3.7 Reconstruction of IDR

To demonstrate the capabilities of the proposed method to compute IDR for a given building on the basis of smartphone data, the results of scaled building experiments conducted by others are used (Lignos 2008) as shown in Figure 5-18. Specifically, the displacement responses of each floor as measured from the building experiments (Lignos 2008) are applied to the shake table in separate experiments. The measurements are recorded using a single smartphone (Samsung

Galaxy S7) for two conditions. The first uses protective shell Case 4 (see Figure 5-10) to minimize sliding motions (denoted EX1). The second uses no protective shell (Case 1 in Figure 5-10) to highlight the effect of sliding motion on displacement estimation (denoted EX2). The displacement records of each floor are estimated following the proposed methodology and IDRs are calculated using the given floor heights of the frame (Lignos 2008).

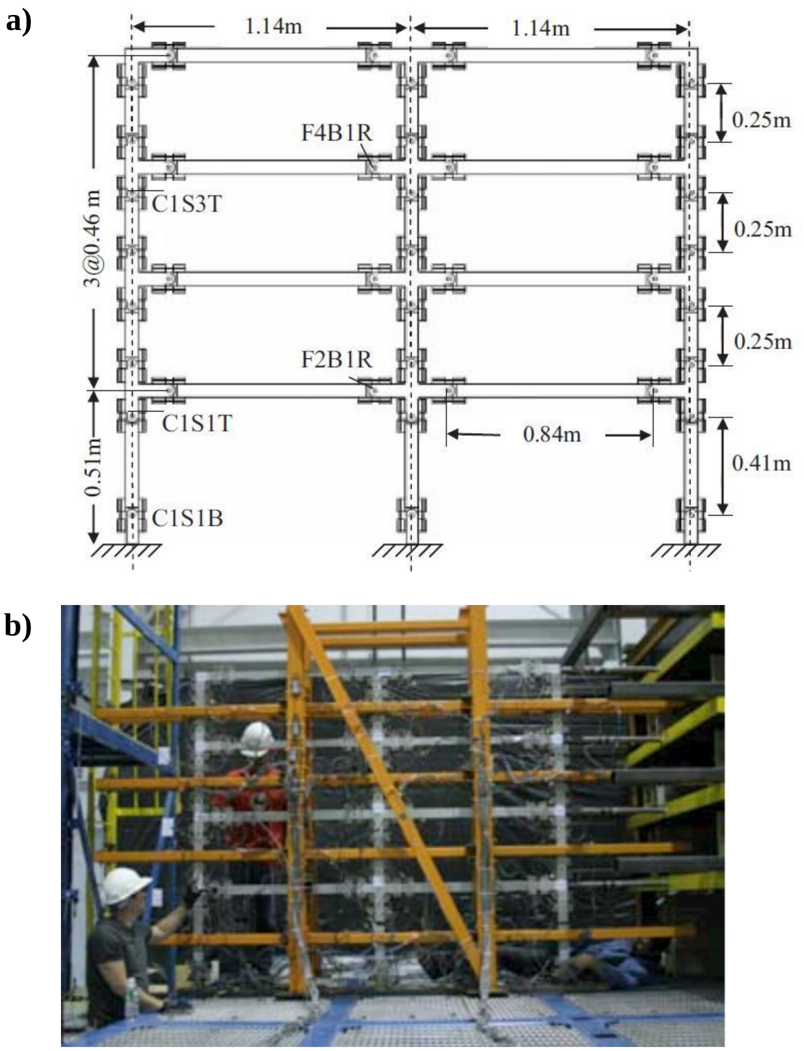


Figure 5-18. Test frame for shaking table tests (Lignos 2008): a) drawing of test frame and b) Test frame for shaking table tests

Figure 5-19 shows the results of the estimated displacements and IDR of each floor for Optotrak and smartphone measurements for Cases 4 and 1, respectively. It is clear that the estimated displacement responses for EX1 match better with the Optotrak measurements compared to that for EX2. The maximum differences in the displacement time histories of each floor are 0.945cm, 0.280cm, 1.175cm, and 0.667cm for EX1, and 3.019cm, 2.590cm, 1.329 cm, and 1.044cm for EX2 for the 1st to 4th floors, respectively.

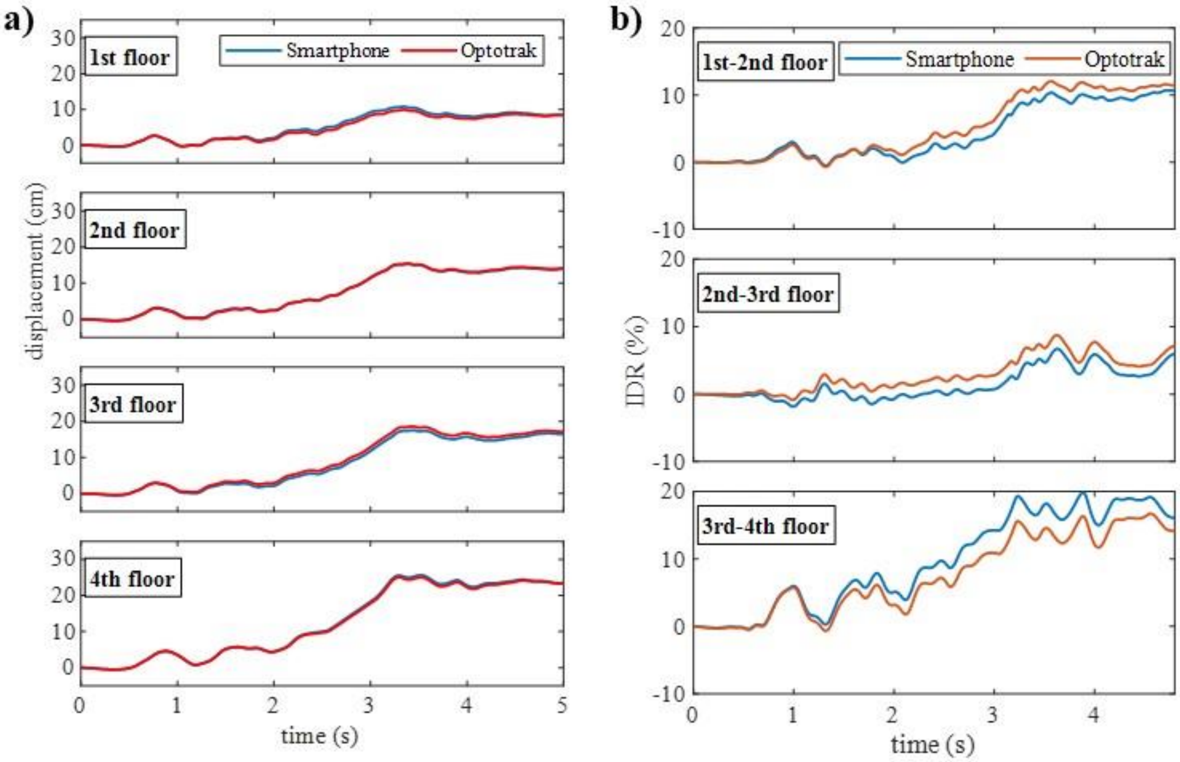


Figure 5-19. Results of estimated displacements and IDRs of each floor for smartphones with Case 4: a) Each floor's displacement (cm) and b) IDR values (%)

The estimated displacements translate into peak IDR differences for EX1 of 3.05%, 2.30%, and 2.81% for 1st-2nd floor, 2nd-3rd floor and 3rd -4th floor, respectively, and 10.31%, 7.86%, and 4.83% for EX2, respectively as shown in Figure 5-20. As expected, these result shows that the smartphone with high KCOF case (EX1) can provide more accurate acceleration measurements and displacement estimations than that with low KCOF (EX2). Most importantly, this study shows the conditions under which smartphone data can be used to estimate actual and residual IDR with reasonable accuracy.

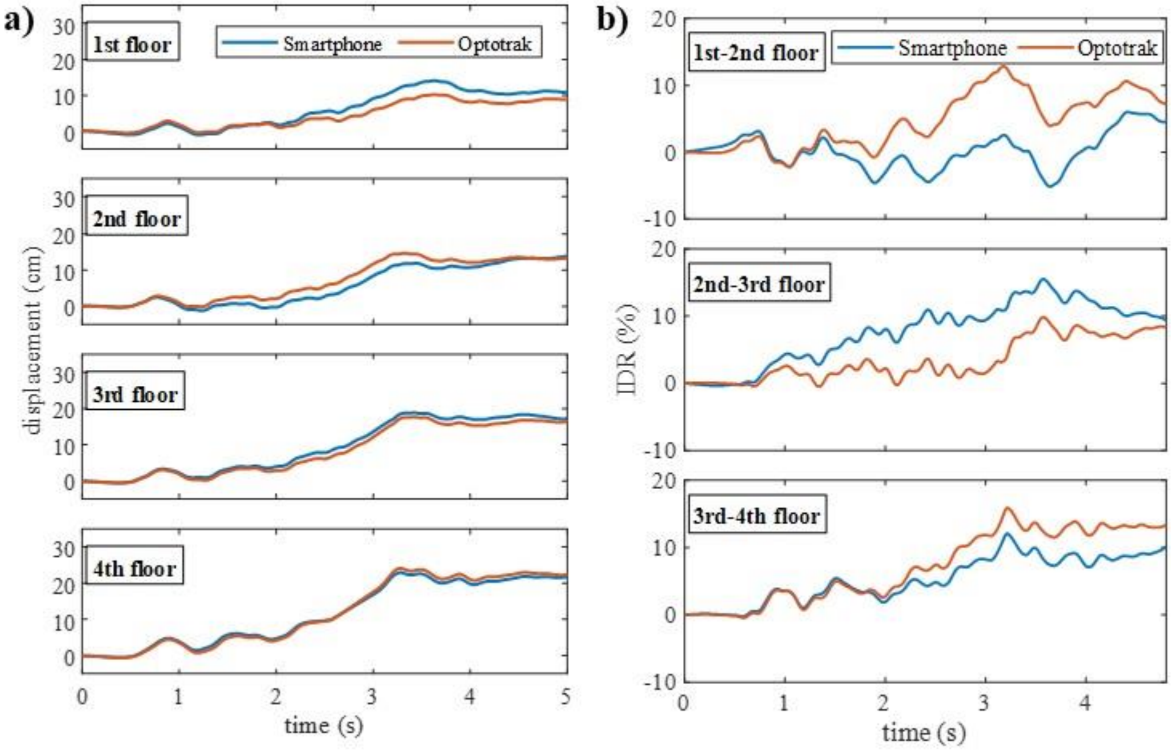


Figure 5-20. Results of estimated displacements and IDRs of each floor for smartphones with Case 1: a) Each floor's displacement (cm) and b) IDR values (%)

Figure 5-21 shows the results of interpolation method (PWCPI) as mentioned in chapter 5.2.3 for the displacement and IDR at unknown floor (third floor). The interpolated displacement at third floor based on the adjacent floors (the second and fourth floor) are in good agreement with the actual displacement from the Optotrak measurement. Using this interpolated displacement, IDR between the 2nd-3rd floor and 3rd-4th floor can be estimated. The interpolated IDR shows also good agreements with the ground truth IDR obtained from the Optotrak. These results confirm that the interpolation method is a robust method for handling missing floor data.

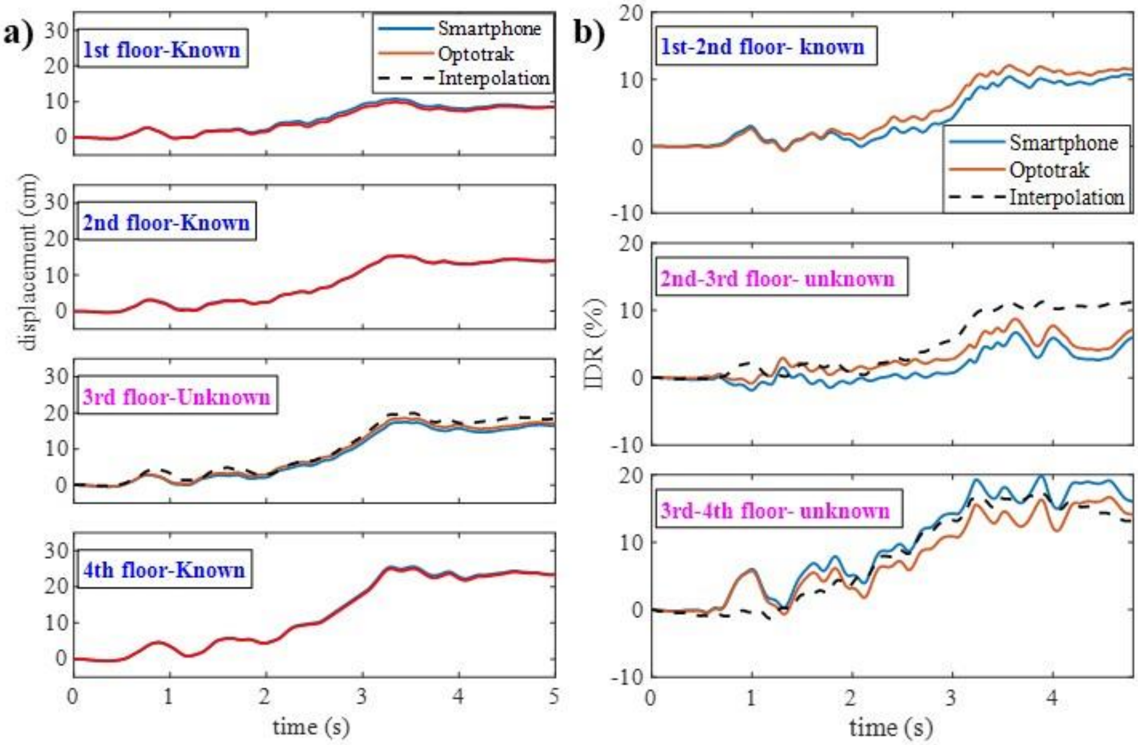


Figure 5-21. Results of interpolation method for the non-instrumented floor at the 3rd floor with Case 1: a) Each floor's displacement (cm) and b) IDR values (%)

5.4 Summary and conclusion

The steps for reconstructing interstory drift ratios from smartphone records was discussed. Estimating IDR in this manner is a critical first step for automating building damage assessment after seismic events. Each step of the proposed process was experimentally investigated, including noise reduction, sliding detection, data fusion, double integration, and calculation of IDR. Different alignment techniques based on the dynamic time warping method for acceleration measurements were compared and the proposed MDTW method was shown to have better performance than the other methods considered. Double integration based on ZUPT method was presented to compensate for errors in velocity estimation.

When data fusion was considered, it was shown that the MDTW method improved the alignment between different measurements and, hence, the accuracy of estimated displacements, compared to methods that did not use MDTW. Experimental results showed that at least five smartphone readings should be aggregated to improve overall accuracy when estimating IDR. The abilities of different types of smartphones to estimate displacement were compared and the error in displacement was shown to have a strong relationship to their MS amplitude of noise while stationary. Another important observation from the experimental data was that the use of protective shells with a high KCOF (high sticking potential) lead to better displacement estimation than those more prone to sliding. The proposed methodology was validated using the results from previously published shake table experiments of a steel frame structure. It is shown that reasonable estimates of IDR can be achieved. Interpolation method shows that robust and reasonable results can be obtained for the displacement and IDR of floors with insufficient data.

A limitation of this research was that multiple smartphone records needed to be aggregated to reduce stochastic errors. If multiple smartphone records are not available, it is difficult to remove stochastic error for single smart device due to randomness of inherent errors of MEMS based accelerometer. Another limitation was the presence of amplitude dependent scale factor errors in the smartphone accelerometers. This dependence may aggravate the displacement estimation because the weight factor that used in this study to combine multiple measurements is based on the noise in stationary. Even though this is out of scope of this research, this is an important finding and it is needed more further investigations to deal with this amplitude dependent scale errors in acceleration measurements.

CHAPTER 6

Research summary and conclusions

6.1 Summary of research

The aim of this dissertation was to propose means by which to automate post-earthquake damage assessment through the use of smart devices that are ubiquitous in most communities. As such, the research explored four inter-related goals: 1) to understand the dynamic behavior of an unconstrained smart device under seismic motions 2) to identify stick-slip characteristic of the smart device on a seismically excited surface; 3) to classify stick-slip motions based on machine learning techniques; and 4) to estimate interstory drift ratios (IDR) from smart devices' acceleration measurements by stacking multiple measurements. Taken together, the studies yielded findings that show great promise for the integration of smart devices in earthquake reconnaissance. A summary of the research conducted in each of these areas is as follows.

6.1.1 Dynamic behavior of an unconstrained smart device under seismic shaking

Chapter 2 presented a study of the dynamic behavior of a smart device placed on the underlying surface of subjected to seismic motion. The smart device was modeled as a rigid block and its frictional interactions with the underlying base was represented using an existing model (sticking-spring-damper friction model), which was modified for the purposes of this research. The first modification used an interpolation technique to enhance detection of transition points, which must be accurately detected because of the frequency of their occurrence during seismic motion. The second modification entailed extending the model to handle vertical accelerations. In this process, the assumption that the normal load is a constant value was relaxed.

To demonstrate the feasibility of the modified friction model, experiments were conducted with a custom-built shaking table. The shaking table consisted of stepper motor and linear actuator and was controlled by Labview and an Arduino microprocessor chipset. The shake table could produce precise, repetitive motions for a given ground motion. Placed on the table were smart devices that recorded multiple rounds of acceleration measurements, which were then combined to reduce the noise. Due to minor surface imperfections, the measured acceleration during sliding motions were not a perfect plateau as computed in the numerical analysis. Nevertheless, the overall response computed by the modified friction model showed good agreement with the acceleration responses measured by an independent and accurate non-contact measurement system.

After validation, the modified friction model was used to investigate the possibility that a device would slide on a flat surface. The aim of this study was to define from a practical perspective the conditions under which the smart device would stick to the surface for a given earthquake intensity. To study the sliding response spectra, a 4-story steel special moment frame (SMF) model designed by NIST was used. The SMF was designed with deep columns and reduced beam sections (RBS) using ASTM A992 steel. Seven scale factors were computed by following the scaling method in FEMA P-695 to generate new records, specifically for three hazard levels: 2%, 10%, and 50% chances of occurrence in 50 years. It was shown that vertical accelerations have a small effect on the sliding behavior of smart devices. The concept of a ‘probability of exceeding the slip limit curve’ was introduced and used to relate the probability of exceeding a given slip limit versus first period spectral acceleration for a given structure and location. This curve suggested that the smart device with a static coefficient of friction of 0.4 and 0.5 can read the movement of the underlying surfaces with a maximum slip of less than 2cm for events with a magnitude corresponding to a 10% and 50% chance of occurrence in 50 years. Once generalized by taking into account other structures and locations, this information could be of value in future crowd-sourced, post-disaster reconnaissance efforts.

6.1.2 Identifying stick-slip characteristics of a smart device

Chapter 3 highlighted two key challenges utilizing smart devices to characterize seismic damage of buildings in a quantitative manner. For one, accelerometers in a smart device produce noisy data and are sampled at a relatively slow rate. For another, sliding can contaminate the acceleration record. To address these issues, first, the error equation for MEMS-based accelerometers in smart devices was presented. The deterministic errors were estimated experimentally by using the static multi-position method. It was shown that the optimal sampling rate to minimize error for a specific smart device must be in excess of 1000Hz to ensure accuracy. Although current devices cannot achieve this sampling rate, it is highly likely that future devcies will because a new generation of high-resolution, low-noise accelerometers, such as Nano-g or nanoelectromechanical system (NEMS) accelerometers, appears poised to enable smart devices.

After development of the error equation, a stick-slip identification method was introduced to determine whether the motion of the smart device is representative of the motion of the floor underneath. This crucial step uses the acceleration measurements, taking into account that acceleration measurements contaminated by excessive sliding action cannot represent the motion of the underlying surface. In a simulated acceleration response, it was shown that the theoretical sliding motions are step shaped with a clear plateau. However, in the measured acceleration response, the sliding motions were observed to have rounded shapes and chattering, which hinders the differentiation of sliding and sticking motions. For the first step of stick-slip identification method, the noise associated with a smart device's measurement of acceleration is established and noise reduction methods are compared. After considering multiple methods, it was shown that the maximal overlap discrete wavelet packet transforms (MODWPT) method out-performs other wavelet transform methods in noise reduction for measurement of acceleration by a smart device, while still maintaining the shape of transitions between sticking and sliding. Then, a kinetic coefficient of friction (KCOF) estimation method was proposed based on the observation that a plateau occurs during sliding. To address the occurrence of chattering during sliding due to imperfections in the frictional surfaces, configurable thresholds were applied. From the shake table experiments, it was shown that the results of the estimated KCOF are in good agreement with the true KCOFs. The estimated KCOF was used to compute slipping accelerations and corresponding sliding motions. It was shown that a 93.9% correct detection rate could be achieved using the proposed method.

6.1.3 Stick-slip classification based on machine learning techniques

Chapter 4 presented an accurate and robust accelerometer-based stick-slip motion classification framework based on two different machine learning methods, supervised learning algorithms and deep learning algorithms. For the former, three methods were investigated: 1) the ReliefF algorithm was used to select highly correlated features among the potential features; 2) the zero-crossing window was selected as the best performing segmentation method for raw acceleration data; and 3) linear discriminant analysis (LDA) and kernel discriminant analysis (KDA) were used to improve the class separability and showed a modest improvement in classification accuracy. For the latter, a detailed examination of a combination of three hyperparameters of the proposed recurrent neural networks (RNN) and its performance were considered important to further improve the accuracy of the network. Two internal layers, 40 units for each internal layer and c=6 gradient clipping parameter were selected. The RNN for the deep learning algorithm showed somewhat better classification accuracy (93.2%) than two other supervised learning techniques, 89.0% for multilayer perception (MLP), and 86.1% for support vector machines (SVM).

6.1.4 Stacking multiple measurements to improve computation of interstory drift ratios.

Chapter 5 proposed a process for reconstructing interstory drift ratios using smart device records by stacking multiple measurements. Each step of the proposed process was experimentally investigated, including noise reduction, sliding detection, data fusion, double integration, and calculation of IDR.

To synchronize multiple measurements, the proposed multiplicative dynamic time warping (MDTW) method, which is based on the dynamic time warping method was shown to have better performance than other methods investigated. To compensate for errors in velocity estimation, double integration based on the zero-velocity update (ZUPT) method and Kalman filter were presented. When data fusion was considered, experimental results showed that at least five smart device readings should be aggregated to improve overall accuracy when estimating IDR. The abilities of different types of smart devices to estimate displacement were compared and the error in displacement was shown to have a strong relationship to their mean square (MS) amplitude of stationary noise. Another important observation from the experimental data was that the use of protective shells with a high KCOF (high sticking potential) led to better displacement estimation than those with a low KCOF. The proposed IDR estimation process was validated using the results from previously published shake table experiments of a four-story steel frame structure. It was shown that reasonable estimates of displacement and IDR of each floor using the proposed IDR estimation method can be achieved. An interpolation method was shown to be capable of obtaining robust and reasonable results for the displacement and IDR of floors with insufficient data.

6.2 Conclusions

Major findings from this research are as follows.

6.2.1 Dynamic behavior of an unconstrained smart device under seismic shaking

- The dynamic behavior of a non-constrained smart device under seismic shaking is simulated with the modified friction model and its results show good agreement with the shaking table experimental data.
- Vertical accelerations have a small effect on the sliding behavior of smart devices for the selected ground motions. They can therefore be ignored in such situations.
- Once generalized by taking into account other structures and locations, the proposed probability of exceeding the slip limit curve could be of value in future crowd-sourced, post-disaster reconnaissance efforts.

6.2.2 Identifying stick-slip characteristics of a smart device

- The optimal sampling rate to minimize error for a specific smart device must be in excess of 1000Hz to ensure accuracy. Although current devices cannot achieve this sampling rate, it is highly likely that future devices will.

- The MODWPT method out performs other wavelet transform methods in noise reduction for measurement of acceleration by a smart device, while still maintaining the shape of transitions between sticking and sliding.
- The estimated KCOF was used to compute slipping accelerations and corresponding sliding motions. It was shown that a 93.9% correct detection rate could be achieved using the proposed method.

6.2.3 Stick-slip classification based on machine learning techniques

- The RNN for the deep learning algorithm showed somewhat better classification accuracy (93.2%) than two other supervised learning techniques, 89.0% accuracy for MLP, and 86.1% for SVM.
- For the RNN algorithms, two internal layers, 40 units for each internal layer and c=7 gradient clipping parameter were shown to give good results.

6.2.4 Stacking multiple measurements to improve computation of interstory drift ratios

- Experimental results showed that at least five smart device readings should be aggregated to improve overall accuracy when estimating IDR.
- The abilities of different types of smart devices to estimate displacement were compared and the error in displacement was shown to have a strong relationship to their mean square (MS) amplitude of stationary noise.
- The use of protective shells with a high KCOF (high sticking potential) lead to better displacement estimation than those with a low KCOF because the shell with higher KCOF sticks more effectively to the underlying surface and has a smaller microslip zone.

- Of most importance, the proposed IDR estimation process can achieve reasonable displacement and IDR of each floor for the shake table experiments of a four-story steel frame structure.
- The interpolation method was shown to be capable of obtaining robust and reasonable results for the displacement and IDR of floors with insufficient data.

6.3 Limitations of the study

It should be borne in mind that this study is limited in two important ways:

- The first is attributable to the study set up. The shake table could represent only one-dimensional motion because of its configuration. Therefore, for the sake of simplicity, in this study it was assumed that the smart devices were oriented to measure only one-dimensional horizontal motion without vertical motion or lift off. Because of the limited experimental setup this is a study of one-dimensional motion only. To simulate more realistic situations, these assumptions need to be relaxed and a fully three-dimensional analysis needs to be conducted.
- Another limitation is the presence of amplitude dependent scale factor errors in the accelerometers of the smart devices, which necessitated stacking of multiple smart device records. In the future, it is expected that fewer readings than those used in this work will be needed as the quality of smart device accelerometers continues to increase.

6.4 Future research

In order to further realize the goal of fast and reliable earthquake damage assessments a number of studies can be done to extend the work presented here. Suggestions for future research can focus on two main areas. The first is to extend the smart device based automated post-earthquake damage assessment system to three-dimensions. The second is to estimate the residual displacement of buildings after earthquake events using an unmanned aerial vehicles (UAV) to improve IDR estimation.

To realize the potential of automated post-earthquake damage assessment, the behavior of the smart device must be measured in three-dimensions. This will require a customized three-dimensional shaking table with a rotary motor to simulate the effects of rotation on a building during an earthquake. This work has the potential to simulate more realistic responses of smart devices during an earthquake and serve as an important step towards completing the automated post-earthquake damage assessment system proposed in this dissertation.

Given the availability of affordable UAV and excellent cameras this technology is a promising choice for estimating the residual displacement of buildings after earthquakes and to enhance the IDR estimation method described herein. The ZUPT method was applied to correct the estimated velocity by the first integration from the measured acceleration. To correct the estimated displacement by the second integration, it is necessary to determine the residual displacement. Using a UAV and the Structure-from-Motion (SfM) technique, 3D models of buildings can be generated, which can be used to improve the estimated displacements and, hence, IDR.

BIBLIOGRAPHY

Abdeljaber, O., Avci, O., Kiranyaz, M.S., Boashash, B., Sodano, H. and Inman, D.J. (2018). "1-D CNNs for structural damage detection: verification on a structural health monitoring benchmark data." Neurocomputing **275**: 1308-1317.

Akhavian, R. and Behzadan, A.H. (2016). "Smartphone-based construction workers' activity recognition and classification." Automation in Construction **71**: 198-209.

Allen, F.R., Ambikairajah, E., Lovell, N.H. and Celler, B.G. (2006). "Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models." Physiological measurement **27**(10): 935.

Allen, J., Kinney, R., Sarsfield, J., Daily, M., Ellis, J., Smith, J., Montague, S., Howe, R., Boser, B. and Horowitz, R. (1998). Integrated micro-electro-mechanical sensor development for inertial applications. IEEE 1998 Position Location and Navigation Symposium (Cat. No. 98CH36153), IEEE.

Amini, A., Trifunac, M. and Nigbor, R. (1987). "A note on the noise amplitudes in some strong motion accelerographs." Soil Dynamics and Earthquake Engineering **6**(3): 180-185.

Antunes, J., Axisa, F., Beaufils, B. and Guilbaud, D. (1990). "Coulomb friction modelling in numericalsimulations of vibration and wear work rate of multispan tube bundles." Journal of Fluids and Structures **4**(3): 287-304.

Applied Technology Council (1989). ATC-20 Procedures for Post Earthquake Safety Evaluation of Buildings. Redwood City, Calif.

Applied Technology Council (1995). ATC 20-2: Addendum to the ATC-20 Postearthquake Building Safety Evaluation Procedures. Redwood City, Calif.

Applied Technology Council (2005). ATC-20-1 Field Manual: Postearthquake Safety Evaluation of Buildings. Redwood City, Calif**:** 12.

ASCE (2002). Minimum design loads for buildings and other structures. Reston, Virginia, American Society of Civil Engineers.

ASCE (2006). Minimum design loads for buildings and other structures, Amer Society of Civil Engineers.

ASTM (2008). D1894-08 Standard Test Method for Static and Kinetic Coefficients of Friction of Plastic Film and Sheeting.

Atallah, L., Lo, B., King, R. and Yang, G.-Z. (2011). "Sensor positioning for activity recognition using wearable accelerometers." IEEE transactions on biomedical circuits and systems **5**(4): 320-329.

Baek, J., Lee, G., Park, W. and Yun, B.-J. (2004). Accelerometer signal processing for user activity detection. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, Springer.

Bao, L. and Intille, S.S. (2004). Activity recognition from user-annotated acceleration data. International Conference on Pervasive Computing, Springer.

Baudat, G. and Anouar, F. (2000). "Generalized discriminant analysis using a kernel approach." Neural computation **12**(10): 2385-2404.

Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J. (1997). "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." IEEE Transactions on Pattern Analysis & Machine Intelligence(7): 711-720.

Bellman, R. and Kalaba, R. (1959). "On adaptive control processes." IRE Transactions on Automatic Control **4**(2): 1-9.

Bengio, Y., Simard, P. and Frasconi, P. (1994). "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks **5**(2): 157-166.

Bengtsson, L., Lu, X., Thorson, A., Garfield, R. and Von Schreeb, J. (2011). "Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: a post-earthquake geospatial study in Haiti." PLoS medicine **8**(8): e1001083.

Bishop, C. (2006). pattern recognition and machine learning. New York, Springer.

Brennan, D.G. (1959). "Linear diversity combining techniques." Proceedings of the IRE **47**(6): 1075-1102.

Bulut, Y., Vines-Cavanaugh, D. and Bernal, D. (2011). Process and measurement noise estimation for Kalman filtering. Structural Dynamics, Volume 3, Springer**:** 375-386.

Choi, J.J., Han, S.I. and Kim, J.S. (2006). "Development of a novel dynamic friction model and precise tracking control using adaptive back-stepping sliding mode controller." Mechatronics **16**(2): 97-104.

Clayton, R.W., Heaton, T., Chandy, M., Krause, A., Kohler, M., Bunn, J., Guy, R., Olson, M., Faulkner, M. and Cheng, M. (2012). "Community seismic network." Annals of Geophysics **54**(6).

Coifman, R.R. and Wickerhauser, M.V. (1992). "Entropy-based algorithms for best basis selection." IEEE Transactions on information theory **38**(2): 713-718.

Dahl, P. (1975). Solid friction damping of spacecraft oscillations. Guidance and Control Conference.

Daubechies, I. (1990). "The wavelet transform, time-frequency localization and signal analysis." IEEE transactions on information theory **36**(5): 961-1005.

Daubechies, I. (1992). Ten lectures on wavelets, Siam.

De la Llera, J. and Chopra, A. (1995). "Evaluating seismic code provisions using strong-motion building records from the 1994 Northridge earthquake." CSMIP report.

De Wit, C.C., Olsson, H., Astrom, K.J. and Lischinsky, P. (1995). "A new model for control of systems with friction." IEEE Transactions on automatic control **40**(3): 419-425.

Demuth, H. and Beale, M. (1994). Neural network toolbox, Mathworks.

El-Diasty, M., El-Rabbany, A. and Pagiatakis, S. (2007). "Temperature variation effects on stochastic characteristics for low-cost MEMS-based inertial sensor error." Measurement Science and Technology **18**(11): 3321.

El-Diasty, M. and Pagiatakis, S. (2008). "Calibration and stochastic modelling of inertial navigation sensor errors." Journal of Global Positioning Systems **7**(2): 170-182.

El-Sheimy, N., Nassar, S. and Noureldin, A. (2004). "Wavelet de-noising for IMU alignment." IEEE Aerospace and Electronic Systems Magazine **19**(10): 32-39.

Ermes, M., Pärkkä, J., Mäntyjärvi, J. and Korhonen, I. (2008). "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions." IEEE transactions on information technology in biomedicine **12**(1): 20-26.

Ervasti, M., Dashti, S., Reilly, J., Bray, J.D., Bayen, A. and Glaser, S. (2011). iShake: mobile phones as seismic sensors--user study findings. Proceedings of the 10th international Conference on Mobile and Ubiquitous Multimedia, ACM.

FEMA (2009). Quantification of building seismic performance factors, Washington, DC.

FEMA (2013). Recommended Seismic Design Criteria for New Steel Moment-frame Buildings: Fema 350, Fema.

Figueiredo, E., Park, G., Figueiras, J., Farrar, C. and Worden, K. (2009). Structural health monitoring algorithm comparisons using standard data sets, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Foerster, F., Smeja, M. and Fahrenberg, J. (1999). "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring." Computers in Human Behavior **15**(5): 571-583.

Gazetas, G., Garini, E., Anastasopoulos, I. and Georgarakos, T. (2009). "Effects of near-fault ground shaking on sliding systems." Journal of geotechnical and geoenvironmental engineering **135**(12): 1906-1921.

Gazetas, G., Garini, E., Berrill, J.B. and Apostolou, M. (2012). "Sliding and overturning potential of Christchurch 2011 earthquake records." Earthquake Engineering & Structural Dynamics **41**(14): 1921-1944.

Giansanti, D., Macellari, V. and Maccioni, G. (2008). "New neural network classifier of fall-risk based on the Mahalanobis distance and kinematic parameters assessed by a wearable device." Physiological measurement **29**(3): N11.

Groves, P.D. (2013). Principles of GNSS, inertial, and multisensor integrated navigation systems, Artech house.

Gui, G., Pan, H., Lin, Z., Li, Y. and Yuan, Z. (2017). "Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection." KSCE Journal of Civil Engineering **21**(2): 523-534.

Hall, M.A. (1999). "Correlation-based feature selection for machine learning."

Hochreiter, S. and Schmidhuber, J. (1997). "Long short-term memory." Neural computation **9**(8): 1735-1780.

Hsieh, C. and Pan, Y.-C. (2000). "Dynamic behavior and modelling of the pre-sliding static friction." Wear **242**(1-2): 1-17.

Ibrahim, A., Eltawil, A., Na, Y. and El-Tawil, S. (2018). "Effect of Sensor Error on the Assessment of Seismic Building Damage." arXiv preprint arXiv:1807.06785.

Inoue, M., Inoue, S. and Nishida, T. (2018). "Deep recurrent neural network for mobile human activity recognition with high throughput." Artificial Life and Robotics **23**(2): 173-185.

Jebelli, H., Hwang, S. and Lee, S. (2018). "EEG-based workers' stress recognition at construction sites." Automation in Construction **93**: 315-324.

Kalman, R.E. (1960). "A new approach to linear filtering and prediction problems." Journal of basic Engineering **82**(1): 35-45.

Kang, C.W., Kang, C.H. and Park, C.G. (2010). "Wavelet Denoising Technique for Improvement of the Low Cost MEMS-GPS Integrated System."

Karantonis, D.M., Narayanan, M.R., Mathie, M., Lovell, N.H. and Celler, B.G. (2006). "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring." IEEE transactions on information technology in biomedicine **10**(1): 156-167.

Karnopp, D. (1985). "Computer simulation of stick-slip friction in mechanical dynamic systems." Journal of dynamic systems, measurement, and control **107**(1): 100-103.

Keogh, E.J. and Pazzani, M.J. (2001). Derivative dynamic time warping. Proceedings of the 2001 SIAM International Conference on Data Mining, SIAM.

Kern, N., Schiele, B., Junker, H., Lukowicz, P. and Tröster, G. (2003). "Wearable sensing to annotate meeting recordings." Personal and Ubiquitous Computing **7**(5): 263-274.

Kiani, K., Snijders, C. and Gelsema, E. (1997). "Computerized analysis of daily life motor activity for ambulatory monitoring." Technology and Health Care **5**(4): 307-318.

Kong, Q., Allen, R.M., Kohler, M.D., Heaton, T.H. and Bunn, J. (2018). "Structural Health Monitoring of Buildings Using Smartphone Sensors." Seismological Research Letters **89**(2A): 594-602.

Kong, Q., Kwony, Y.-W., Schreierz, L., Allen, S., Allen, R. and Strauss, J. (2015). Smartphone-based networks for earthquake detection. 2015 15th International Conference on Innovations for Community Services (I4CS), IEEE.

Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P. and Roning, J. (2009). Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. 2009 17th Mediterranean Conference on Control and Automation, IEEE.

Krishnan, N.C. and Cook, D.J. (2014). "Activity recognition on streaming sensor data." Pervasive and mobile computing **10**: 138-154.

Kuhn, M. and Johnson, K. (2013). Applied predictive modeling, Springer.

Laguna, J.O., Olaya, A.G. and Borrajo, D. (2011). A dynamic sliding window approach for activity recognition. International Conference on User Modeling, Adaptation, and Personalization, Springer.

Laine, A. and Fan, J. (1993). "Texture classification by wavelet packet signatures." IEEE Transactions on pattern analysis and machine intelligence **15**(11): 1186-1191.

Lampaert, V., Swevers, J. and Al-Bender, F. (2002). "Modification of the Leuven integrated friction model structure." IEEE transactions on Automatic Control **47**(4): 683-687.

Li, H., Dong, S., El-Tawil, S. and Kamat, V.R. (2012). "Relative displacement sensing techniques for postevent structural damage assessment." Journal of Structural Engineering **139**(9): 1421-1434.

Li, Y., Gong, S. and Liddell, H. (2003). "Recognising trajectories of facial identities using kernel discriminant analysis." Image and Vision Computing **21**(13-14): 1077-1086.

Lignos, D. (2008). Sidesway collapse of deteriorating structural systems under seismic excitations, Stanford university.

Mallat, S. and Hwang, W.L. (1992). "Singularity detection and processing with wavelets." IEEE transactions on information theory **38**(2): 617-643.

Mallat, S. and Zhong, S. (1992). "Characterization of signals from multiscale edges." IEEE Transactions on Pattern Analysis & Machine Intelligence(7): 710-732.

Mantyjarvi, J., Himberg, J. and Seppanen, T. (2001). Recognizing human motion with multiple acceleration sensors. Systems, Man, and Cybernetics, 2001 IEEE International Conference on, IEEE.

Mantyjarvi, J., Himberg, J. and Seppanen, T. (2001). Recognizing human motion with multiple acceleration sensors. 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236), IEEE.

Maurer, U., Smailagic, A., Siewiorek, D.P. and Deisher, M. (2006). Activity recognition and monitoring using multiple sensors on different body positions. Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on, IEEE.

Menai, M.E.B., Mohder, F.J. and Al-mutairi, F. (2013). "Influence of feature selection on naïve Bayes classifier for recognizing patterns in cardiotocograms." Journal of Medical and Bioengineering **2**(1).

Mika, S., Ratsch, G., Weston, J., Scholkopf, B. and Mullers, K.-R. (1999). Fisher discriminant analysis with kernels. Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468), Ieee.

Minnen, D., Starner, T., Ward, J.A., Lukowicz, P. and Troster, G. (2005). Recognizing and discovering human actions from on-body sensor data. Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, IEEE.

Minnen, D., Starner, T., Ward, J.A., Lukowicz, P. and Troster, G. (2005). Recognizing and discovering human actions from on-body sensor data. 2005 IEEE International Conference on Multimedia and Expo, IEEE.

Naeim, F. (1997). "Performance of extensively instrumented buildings during the January 17, 1994 Northridge earthquake." An Interactive Information System, Report No. 97 **7530**.

Najafi, B., Aminian, K., Paraschiv-Ionescu, A., Loew, F., Bula, C.J. and Robert, P. (2003). "Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly." IEEE Transactions on biomedical Engineering **50**(6): 711-723.

Narayanan, M.R., Scalzi, M.E., Redmond, S.J., Lord, S.R., Celler, B.G. and Lovell, N.H. (2008). A wearable triaxial accelerometry system for longitudinal assessment of falls risk. 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE.

Nassar, S. (2003). Improving the inertial navigation system (INS) error model for INS and INS/DGPS applications, University of Calgary, Department of Geomatics Engineering.

Noureldin, A., Karamat, T.B. and Georgy, J. (2012). Fundamentals of inertial navigation, satellite-based positioning and their integration, Springer Science & Business Media.

Ordóñez, F.J. and Roggen, D. (2016). "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition." Sensors **16**(1): 115.

Park, K.-T., Kim, S.-H., Park, H.-S. and Lee, K.-W. (2005). "The determination of bridge displacement using measured acceleration." Engineering Structures **27**(3): 371-378.

Park, M. and Gao, Y. (2006). "Error analysis and stochastic modeling of low-cost MEMS accelerometer." Journal of Intelligent and Robotic Systems **46**(1): 27-41.

Parlitz, U., Hornstein, A., Engster, D., Al-Bender, F., Lampaert, V., Tjahjowidodo, T., Fassois, S., Rizos, D., Wong, C. and Worden, K. (2004). "Identification of pre-sliding friction dynamics." Chaos: An Interdisciplinary Journal of Nonlinear Science **14**(2): 420-430.

Pascanu, R., Mikolov, T. and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. International Conference on Machine Learning.

Percival, D.B. and Walden, A.T. (2006). Wavelet methods for time series analysis, Cambridge university press.

Prasanna, P., Dana, K.J., Gucunski, N., Basily, B.B., La, H.M., Lim, R.S. and Parvardeh, H. (2016). "Automated crack detection on concrete bridges." IEEE Transactions on Automation Science and Engineering **13**(2): 591-599.

Ravi, N., Dandekar, N., Mysore, P. and Littman, M.L. (2005). Activity recognition from accelerometer data. Aaai.

Ribeiro, J.G., De Castro, J.T. and Freire, J.L. (2003). Using the FFT-DDI method to measure displacements with piezoelectric, resistive and ICP accelerometers. Conference and exposition on structural dynamics, Citeseer.

Robnik-Šikonja, M. and Kononenko, I. (2003). "Theoretical and empirical analysis of ReliefF and RReliefF." Machine learning **53**(1-2): 23-69.

Rosenfeld, A. (1970). "A nonlinear edge detection technique." Proceedings of the IEEE **58**(5): 814-816.

Ryu, J., Seo, J., Jebelli, H. and Lee, S. (2018). "Automated Action Recognition Using an Accelerometer-Embedded Wristband-Type Activity Tracker." Journal of Construction Engineering and Management **145**(1): 04018114.

Sadler, B.M., Pham, T. and Sadler, L.C. (1998). "Optimal and wavelet-based shock wave detection and estimation." The Journal of the Acoustical Society of America **104**(2): 955-963.

Sadler, B.M. and Swami, A. (1998). Analysis of wavelet transform multiscale products for step detection and estimation, ARMY RESEARCH LAB ADELPHI MD.

Sadler, B.M. and Swami, A. (1999). "Analysis of multiscale products for step detection and estimation." IEEE Transactions on Information Theory **45**(3): 1043-1051.

Sakoe, H. and Chiba, S. (1978). "Dynamic programming algorithm optimization for spoken word recognition." IEEE transactions on acoustics, speech, and signal processing **26**(1): 43-49.

Sarma, S. and Scorer, M. (2009). The effect of vertical accelerations on seismic slope stability. Proceedings of the international conference on performance-based design in earthquake geotechnical engineering. Taylor and Francis Group, London.

Sekiya, H., Kimura, K. and Miki, C. (2016). "Technique for Determining Bridge Displacement Response Using MEMS Accelerometers." Sensors (Basel) **16**(2): 257.

Shigley, J.E. and Mitchell, L.D. (1983). "Mechanical engineering design." McGraw-Hill Book Co., 1983: 869.

Sorine, M. (1998). Applications of hysteresis models: Contact friction in tires, muscle contraction. IEEE CDC 98 Workshop.

Stephane, M. (1999). "A wavelet tour of signal processing." The Sparse Way.

Sun, Z. and Chang, C. (2002). "Structural damage assessment based on wavelet packet transform." Journal of structural engineering **128**(10): 1354-1361.

Sung, M., Marci, C. and Pentland, A. (2005). "Wearable feedback systems for rehabilitation." Journal of neuroengineering and rehabilitation **2**(1): 17.

Swevers, J., Al-Bender, F., Ganseman, C.G. and Projogo, T. (2000). "An integrated friction model structure with improved presliding behavior for accurate friction compensation." IEEE Transactions on automatic control **45**(4): 675-686.

Tan, X. and Rogers, R.J. (1996). Dynamic friction modelling in heat exchanger tube simulations. Flow-induced vibration--1996. PVP-Volume 328.

Tariku, F.A. and Rogers, R.J. (2001). "Improved dynamic friction models for simulation of one-dimensional and two-dimensional stick-slip motion." Journal of Tribology **123**(4): 661-669.

Thong, Y., Woolfson, M., Crowe, J., Hayes-Gill, B. and Challis, R. (2002). "Dependence of inertial measurements of distance on accelerometer noise." Measurement Science and Technology **13**(8): 1163.

Titterton, D., Weston, J.L. and Weston, J. (2004). Strapdown inertial navigation technology, IET.

Trifunac, M. and Todorovska, M. (2001). "A note on the useable dynamic range of accelerographs recording translation." Soil dynamics and Earthquake engineering **21**(4): 275-286.

Trifunac, M. and Todorovska, M. (2001). "A note on the useable dynamic range of accelerographs recording translation11." Soil dynamics and Earthquake engineering **21**(4): 275-286.

Velenis, E., Tsiotras, P., Canudas-De-Wit, C. and Sorine, M. (2005). "Dynamic tyre friction models for combined longitudinal and lateral vehicle motion." Vehicle system dynamics **43**(1): 3-29.

Vitola, J., Pozo, F., Tibaduiza, D. and Anaya, M. (2017). "A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications." Sensors **17**(2): 417.

Walden, A.T. and Cristan, A.C. (1998). "The phase–corrected undecimated discrete wavelet packet transform and its application to interpreting the timing of events." Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454**(1976): 2243-2266.

Westermo, B. and Udwadia, F. (1983). "Periodic response of a sliding oscillator system to harmonic excitation." Earthquake engineering & structural dynamics **11**(1): 135-146.

Wickerhauser, M.V. (1996). Adapted wavelet analysis: from theory to software, AK Peters/CRC Press.

Wu, T.-Y., El-Tawil, S. and McCormick, J. (2018). "Highly ductile limits for deep steel columns." Journal of Structural Engineering **144**(4): 04018016.

Yen, G.G. and Lin, K.-C. (2000). "Wavelet packet feature extraction for vibration monitoring." IEEE transactions on industrial electronics **47**(3): 650-667.

Yu, Y., Han, R., Zhao, X., Mao, X., Hu, W., Jiao, D., Li, M. and Ou, J. (2015). "Initial validation of mobile-structural health monitoring method using smartphones." International Journal of Distributed Sensor Networks **11**(2): 274391.

Zarei, J. and Poshtan, J. (2007). "Bearing fault detection using wavelet packet transform of induction motor stator current." Tribology International **40**(5): 763-769.