Fall Detection Algorithms Using Accelerometers, Gyroscopes and a Barometric Pressure Sensor

by

Magnus Michael Marquez Musngi

B.A.Sc., Simon Fraser University, 2015

B.S., De La Salle University – Manila, 2009

Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Applied Science

in the

School of Mechatronic Systems Engineering

Faculty of Applied Sciences

SIMON FRASER UNIVERSITY

Spring 2018

Approval

Name:	**Magnus Michael M. Musngi**
Degree:	**Master of Applied Science**
Title:	**Fall Detection Algorithms Using Accelerometers, Gyroscopes, and a Barometric Pressure Sensor**
Examining Committee:	**Chair:** **Dr. Behraad Bahreyni** Associate Professor
	Dr. Edward J. Park Senior Supervisor Professor
	Dr. Stephen Robinovitch Supervisor Professor
	Dr. Siamak Arzanpour Internal Examiner Associate Professor
Date Defended/Approved:	April 19, 2018

Ethics Statement

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

a. human research ethics approval from the Simon Fraser University Office of Research Ethics

or

b. advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

c. as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016

Abstract

Falls commonly occur in older adults and could result in long-lies when no one is around to assist, which could result to additional emotional and physical consequences. The use of inertial sensors allows a portable and unobtrusive way to detect motion, enabling the automatic detection of falls when used with a fall detection algorithm. The wrist and trunk are two locations that are favorable for fall detection as the former provides a convenient location for the user, while the latter provides a good location for capturing the body's general motion. The objective of this thesis is to further improve the performance of a wrist-mounted and a trunk-mounted threshold-based fall detection algorithm using inertial sensors comprised of tri-axial accelerometer, tri-axial gyroscope, and a barometric pressure sensor. The algorithms were tested using a comprehensive set of laboratory-simulated falls, activities of daily living (ADL), and near-falls.

In the first study, a wrist-based fall detection algorithm for a commercially available smartwatch was proposed. The algorithm used forearm angle to filter the forearm's downward vertical orientation that could be associated to a non-fall event's post-activity position. Additionally, to deal with disturbance in barometric pressure data during dynamic motion, barometric pressure was used selectively in a Kalman filter. The algorithm gave 100% sensitivity, 97.2% ADL specificity, and 97.1% non-fall (i.e. including both ADLs and near-falls) specificity.

In the second study, the addition of either difference in altitude or average vertical velocity to a trunk-based algorithm that uses vertical velocity + vertical acceleration + trunk-angle (base algorithm) was investigated. The experimental results show that adding either difference in altitude or average vertical velocity was able to increase the algorithm's non-fall specificity from 91.8% to 98.0% and 99.6%, respectively.

Keywords: Smartwatch-based fall detection; trunk-based fall detection; selective use of pressure; forearm angle; average vertical velocity; Kalman filter

Dedication

This thesis is dedicated to my family, friends, and to older adults who could potentially benefit from this work.

Acknowledgements

I would like to thank: Dr. Edward Park my senior supervisor and Bigmotion Technologies Inc., my industry sponsor, for trusting me with this research, for guidance and advice, and for all the help and support; Dr. Stephen Robinovitch my supervisor, for access to Injury Prevention and Mobility Laboratory's (IPML) fall lab; Drs. Omar Aziz and Shaghayegh Zihajehzadeh, my colleagues in the Biomechatronic Systems Laboratory (BSL) for advice and assistance on the experiments, and the use of Kalman filters, respectively; Chantel Galang, Brigitte Poitvin and Dr. Andreas Ejupi of IPML for their contribution to the design of the experiments and all their assistance in conducting it; Ginelle Nazareth (previously with BSL) for her previous work during her co-op that were helpful to this research; my other colleagues in BSL (current and previous) and friends who also helped and contributed in this research and in this journey; to all the participants who volunteered in our lengthy and exhausting experiments; and to our university SFU and this country through which I was able to have this opportunity to pursue a Master's degree.

Special thanks to my family, especially my parents for all the help, support and sacrifice throughout this journey.

Above all, I would like to thank God for guiding me in this journey and in my life, for the strength, for His unfailing love, and for all His blessings. To Him be the glory forever.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ADL	Activities of Daily Living
AMU	Altitude Measurement Update
Ave.V.Vel.	Average Vertical Velocity
D.Alt.	Difference in Altitude
DVO	Downward Vertical Orientation
FP	False Positive
FN	False Negative
HPFEA	High-Pass Filtered External Acceleration
IMU	Inertial Measurement Unit
IPML	Injury Prevention and Mobility Lab
LTC	Long-Term Care
MEMS	Microelectromechanical Systems
ROC	Receiver Operating Characteristic
SMA	Signal Magnitude Area
SUP	Selective Use of Pressure
TN	True Negative
TP	True Positive
VPV	Vertical Position and Velocity
V.Acc.	Vertical Acceleration
V.Vel	Vertical Velocity
ZVU	Zero Velocity Update

Glossary

Base Algorithm	Reference trunk-based fall detection algorithm in Study 2 which uses vertical velocity, vertical acceleration, and trunk-angle as parameters.
Study 1	Smartwatch-based fall detection study.
Study 2	Trunk-based fall detection study.

Chapter 1.

Introduction

1.1. Need to Automatically Detect Falls

Falls are a major health concern to older adults. Falls can cause physical injury, negative mental health outcomes, reduced quality of life, and even mortality [1, 2]. In Canada, falls account for 85% of injury-related hospitalization in seniors each year [2], and 1 out of 5 older adults reported that they suffered from a serious fall-related injury [3]. When a person is unable to get up after a fall and is alone, it could result in a long-lie (i.e. remaining on the ground for more than an hour [4]). Long-lies could cause emotional trauma and physical ailments such as hypothermia, dehydration, bronchopneumonia, and pressure sores [5]. In a study by Tinetti *et al*. of community-dwelling seniors of at least the age of 72, 47% were unable to get up after a fall even without sustained serious injuries [6]. Having the capability to automatically detect falls will allow automatic calling for assistance, and hence could prevent long-lies.

1.2. Detecting Falls Using MEMS (Micro-Electro-Mechanical Systems) Inertial Sensors

During a fall, there are unique observable characteristics through which a fall could be distinguished from a non-fall event. Being able to detect these characteristics allow the automatic detection of falls. In the past, different methods for fall detection, such as, video-based, acoustic-based, and body-mounted MEMS inertial sensor-based methods had been studied. Among them, an inertial sensor-based method, which detects a fall via analysis of body's kinematics, provides a portable and less expensive way of automatically detecting falls [7]. The miniature size of MEMS inertial sensors allows for its unobtrusive use [8, 9, 10, 11, 12]. Additionally, inertial sensor-based methods do not raise privacy concerns compared to video-based methods which analyzes images [13].

Thresholding parameters that are calculated using inertial sensors is one of the main fall detection methods employed, and has been an active research area during the

last decade. In addition to monitoring multiple parameters (e.g. acceleration, velocity, posture, etc.) to asses different phases of a fall [14, 15, 16, 17], various methods for estimating parameters were also proposed in the past (e.g. in [16, 18, 19, 20, 21]). Most of the studies used accelerometers [22, 17, 16, 14] and others used a combination of accelerometers and gyroscopes [19, 18, 23] or a barometric pressure sensor [15].

1.3. Favorable Sensor Locations: Wrist and Trunk

The wrist is a convenient location for a fall detector to be worn at as it similar to wearing a watch and does not require frequent removal during activities such as using the washroom or changing clothes [23]. Recently, programmable smartwatches that are equipped with MEMS inertial sensors, barometric pressure sensor, user interface, and wireless communications capabilities have become available to the consumer market. These capabilities make the smartwatches an available platform for wrist-based fall detection. Additionally, with the programmability of these smartwatches, other health related apps and functionalities that could benefit the older adult population can also be added to them.

However, detecting falls using wrist-worn sensors is a challenging task due to the hand's diverse functionalities, and the arm's dynamics of motion and articulation. Consequently, the wrist moves frequently and could easily produce big or abrupt movements during some ADLs. Additionally, it could only give a limited amount of information about the body's general motion. Using only an accelerometer, Degen *et al*. implemented a wrist-based fall detection algorithm employing norm of acceleration and two forms of vertical velocity estimate as parameters, and were only able to detect 65% of all the fall trials [16]. Using the same type of sensor, Kangas *et al.* compared the performance of multiple algorithms of different complexities using sensors at different locations, and reported the maximum sensitivity of 71% (at 100% specificity) for the wrist, whereas for head and waist, the sensitivity was about 98% and 97%, respectively [14]. Using accelerometers and gyroscopes, Casilari *et al.* [24] used sensors on the thigh in addition to the wrist sensors and reported the best combination of sensitivity and specificity of 96.7% and 98.3%, respectively, among the algorithms tested. However, when using only sensors on the wrist, sensitivity and specificity was reduced to 93.3% and 93.3%, respectively. Using the same type of sensors, Hsieh *et al.* reported an

average sensitivity and specificity of 95% and 97%, respectively [23]. However, sensors on both hands were required, which is impractical.

Although not as convenient as the wrist location, the trunk is a more ideal location for fall detection since it contains a major part of the body's total mass, where its motion represents "whole body" movements [25, 8]. Additionally, it does not produce big and quick motions as easily like the wrist during ADLs.

Despite the high or even perfect sensitivities and specificities of trunk-based fall detection algorithms (e.g. in [22, 14, 17]), a study by Bagala *et al.* [26] showed that their performance were substantially lower (best was 83% sensitivity at 97% specificity, while worst was 100% sensitivity at 11% specificity) when tested using accelerometer data from frail older adults. Additionally, even if high performance of fall algorithms is already achieved, further improving them (e.g. by having parameters with better discriminative capacities) will further increase their robustness as it will make them more capable of handling variations in fall and non-fall kinematics. Furthermore, fall detection algorithms were seldom tested using near-falls [18, 27], which also commonly occurs in older adults [28, 29], and could potentially cause false alarms due to the associated abrupt movements [18].

Another convenient form for a fall detector is through a pendant since it is similar to wearing a necklace. Although the pendant will be resting on or will be close to the trunk, having it suspended through a cord will make its motion more complex compared to a directly mounted sensor. This complexity could limit the amount of useful kinematic information for fall detection. Additionally there is a risk of strangulation from the fall detector's cord, as there had been cases of strangulation from medical alert necklaces [30, 31].

1.4. Improving the Accuracy of Fall Detection Algorithms

The performance of an algorithm heavily relies on the collective discriminative capacity of the algorithm's monitored parameters. A way to improve the performance of an algorithm is to develop new parameters (or improve existing ones) such that they are capable of addressing the inadequacy of current set of parameters. However, the types of parameters that could be developed (including its reliability) are limited by the types of

motion that could be estimated using the set of sensors employed [19, 16]. A recent development in threshold-based fall detection algorithms showed that a barometric pressure sensor can be used as an alternative for altitude information and was able to improve the accuracy of an accelerometer-based fall detection algorithm for the trunk [15]. Additionally, fusing barometric pressure data with accelerometer and gyroscope data allows an accurate and drift free estimate of vertical velocity and altitude [32].

1.5. Thesis Objectives

1.5.1. Overall Objective

The overall objective of this thesis is to improve the performance and to develop novel methods for wrist-based and trunk-based fall detection algorithms for wearable fall detectors equipped with multiple MEMS inertial sensors.

1.5.2. Specific Objectives

- To develop an accurate wrist-based fall detector using a smartwatch equipped with MEMS accelerometers, gyroscopes, and a barometric pressure sensor
- To improve the performance of a trunk-based fall detector using a wearable sensor equipped with MEMS accelerometers, gyroscopes, and a barometric pressure sensor

1.6. Scope and Delimitation

The scope and delimitation of this thesis were determined based on the research interests of the sponsoring company, Bigmotion Technologies Inc., in consideration to the requirements of a Master of Applied Science thesis, and were as follows:

- The fall detection algorithms will be developed for wearable inertial sensors employing MEMS accelerometers, gyroscopes, and barometers, mounted on the wrist and trunk.

- The wrist-based fall detection algorithm will be specifically designed for a commercially available smartwatch that will be provided by the sponsoring company.

- Each of the fall detection algorithms will only use a single wearable device as source of sensor data (for user convenience).

- Between the two major categories for inertial sensor-based fall detection (i.e. threshold-based and machine learning-based), only the threshold-based method will be considered.

- Only sensor data gathered from laboratory-based simulated trials from young adults will be used in the study (vs. real-world falls or simulated laboratory-based falls from older adults).

- The fall detection algorithms will be designed in a way that they could be implemented online on a device (for real-time fall detection and to conserve power resources by not having to transmit all sensor data).

- The fall detection algorithms will be evaluated offline using the sensor data gathered from the experiments (vs. running the algorithm online while performing the laboratory-based trials).

1.7. Organization of Thesis

Chapter 2 provides the review of the literature that will serve as the foundation and motivation for the methods that will be used for the fall detection algorithms, which will be presented in Studies 1 and 2 (i.e. in Chapters 4 and 5, respectively).

Chapter 3 presents the methods that were used for gathering the sensor data that were used to develop (in part) and evaluate the fall detection algorithms that will be presented in Studies 1 and 2.

Chapter 4 presents the first study which proposes a novel wrist-based fall detection algorithm for a commercially available smartwatch equipped with tri-axial accelerometer, tri-axial gyroscope, and a barometric pressure sensor. This chapter is an expanded version of a journal manuscript for the study and hence follows its format.

Chapter 5 presents the second study which evaluates the effect of adding average vertical velocity and difference in altitude parameters to a trunk-based fall detection algorithm (base algorithm) that uses vertical velocity, vertical acceleration, and trunk angle. This chapter is an expanded version of a conference manuscript for the study and hence follows its format.

Chapter 6 concludes this thesis by providing a summary, the main contributions of this thesis, and recommendations for future work.

Chapter 2.

Literature Review

2.1. Detecting Falls Using MEMS Inertial Sensors

During a fall, there are unique motion characteristics through which a fall could be distinguished from a non-fall event [33]. Body mounted MEMS inertial sensor systems, together with motion measurement techniques allow a portable and unobtrusive way of quantifying body's motion [8, 9, 10, 11, 12]. Together with a fall detection algorithm (which detects the unique motion characteristics or signatures of a fall), a fall can be automatically detected. Additionally, having the necessary communications infrastructure allows the automatic calling for assistance.

Threshold-based and machine learning-based algorithms are the two main categories of fall detection algorithms. In threshold-based fall detection, a fall is detected based on the satisfaction of predefined threshold values of different monitored parameters (e.g. acceleration, velocity, posture, etc.) according to a predefined logic structure. These parameters, thresholds and structure (including timing relationships) are selected and/or developed based on (and to capture) the kinematic and signal characteristics that are dominant to a fall event. On the other hand, in machine learning-based fall detection, a model that defines the relationship between the selected set of features (equivalent of parameters in threshold-based method) and the two classifications, fall and non-fall, are used to differentiate between the two events. This model is developed via automated means using machine learning algorithms using labeled (i.e. whether fall or non-fall) training data.

Threshold-based fall detection algorithm offers the following strengths:

1. Since the algorithm development, starting from parameter selection, to structure design, threshold selection, and tuning, is done via analysis of the body kinematics, parameter’s signals and algorithm’s performance, insight is gained during the development process.

2. Quick adjustments to the algorithm’s threshold and structure is possible.

3. Since the algorithm's structure and its mapping to the actual kinematics are known:

    a. Strengths and weaknesses of the algorithm for specific motion characteristics could be analyzed.

    b. Algorithm tuning could be guided by the knowledge of its effect to the detection of the monitored kinematic and signal characteristics.

    c. If there is new information that could help optimize the algorithm, like user-specific motion characteristics (i.e. for personalized tuning), the algorithm could be adjusted. This attribute may also be applicable for tuning the algorithm during transition from use of laboratory-based fall trials to real-world fall data while there is only a few real-world fall data available at the moment.

4. Contribution of each parameter to the algorithm's performance could be analyzed.

5. The knowledge gained from the algorithm's development process could be used in developing a machine learning-based algorithm (e.g. in terms of which parameters are effective).

6. It has a low computational cost [34].

2.2. Threshold-Based Fall Detection Algorithms

Threshold-based fall detection has been an active research area during the last decade. In addition to monitoring multiple parameters (e.g. acceleration, velocity, posture, etc.) to asses different phases of a fall [14, 15, 16, 17], various methods for estimating parameters were also proposed in the literature (e.g. in [16, 18, 19, 20, 21]).

2.2.1. Measuring Algorithm's Performance

Sensitivity and specificity are the metrics used to describe a fall detection algorithm's classification performance. Sensitivity measures how well the algorithm

correctly classifies falls (positive cases) and is calculated using (2.1), where true positives (TP) are the fall events that were correctly classified as falls, while false negatives (FN) are fall events that were incorrectly classified as non-falls. On the other hand, specificity measures how well the algorithm correctly classifies non-fall events (negative cases) and are calculated using (2.2), where true negatives (TN) are non-fall events that were correctly classified as non-falls, while false positives (FP) are non-fall events that were incorrectly classified as falls. The values for these two metrics will depend on how differentiable fall and non-fall events are based on the collective discriminative capacity of the parameters used, together with their assigned thresholds and the algorithm's structure. An ideal algorithm will have 100% sensitivity and 100% specificity; however, if fall and non-fall events cannot be completely differentiated using the selected set of parameters and the algorithm's structure, there will be a trade-off between sensitivity and specificity.

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \tag{2.1}$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \tag{2.2}$$

Accuracy is a metric that is sometimes used in fall detection [15] to describe the algorithm's overall capacity to correctly classify either of the two events (2.3). However, compared to accuracy, sensitivity and specificity provides specific information about the algorithm's classification performance for each of the activity type, which is important since there is a trade-off between the two metrics.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \times 100\% \tag{2.3}$$

Sensitivity and specificity are also used to measure the classification performance of the individual parameters of an algorithm (for a given threshold).

2.2.2. Accelerometer-Based Fall Detection Algorithms

Using a tri-axial accelerometer, Bourke *et al.* [22] evaluated two simple algorithms for both trunk and thigh using only the norm of accelerometer data as a

parameter. One used an upper fall threshold to detect the impact phase of a fall, while the other used a lower fall threshold to detect the acceleration of the trunk during the descent phase of the fall. For the trunk-located sensor, the former resulted to 100% sensitivity and 100% specificity, while the latter resulted to 100% sensitivity and 91.25% specificity.

Kangas *et al.* [14] compared three algorithms of different complexities for the waist, head, and wrist, wherein at least two or more phases of fall were used for each algorithm. The first algorithm used both impact and posture, the second used both start of fall, impact, and posture, and the third algorithm used start of fall, velocity, impact, and posture. The first algorithm tested all of total sum vector (acceleration norm), dynamic sum vector, sliding sum vector, or vertical acceleration for the impact, while the second and third algorithm only used acceleration norm. For all the algorithms, lying posture was detected by directly thresholding the low-pass filtered acceleration in the vertical axis. For vertical velocity, integration of acceleration norm was only done starting from the detection of pit during start of fall until impact. The best algorithm for the waist-based sensor was the impact + posture algorithm which gave 97% sensitivity and 100% specificity using either total sum vector or vertical acceleration for impact.

Bourke *et al.* [17] evaluated 21 novel and existing algorithms of varying degrees of complexity using a waist-mounted sensor. Combinations of velocity, impact, and posture algorithms were tested, wherein 4 types of impact were used, and for the impact and posture combination, both initialized and uninitialized (as used in [35]) posture were used. Vertical velocity was estimated using the method proposed in [20]. Results show that algorithms that used at least both velocity and posture gave 100% sensitivity and specificity. The algorithms were also tested using continuous unscripted ADLs from older adult volunteers, where the combination of velocity, impact, and posture gave the lowest false positive rate of 0.6 false positives per day among the algorithms (using either of the methods for impact).

For a wrist-based sensor, Degen *et al.* [16] proposed an algorithm which used the acceleration norm and two forms of vertical velocity estimate and was only able to detect 65% of all the fall trials.

For the same aforementioned study of Kangas *et al.* [14], when the algorithms were tested for the wrist (with the removal of posture), the best method was the use of impact using vertical acceleration, but only gave 73% sensitivity at 100% specificity. Kangas *et al.* mentioned that the wrist does not appear to be a suitable location for fall detection.

2.2.3. Gyroscope-Based Fall Detection Algorithm

Bourke *et al.* [36] proposed a two-stage algorithm using a bi-axial gyroscope. The first stage required the satisfaction of norm of roll and pitch angular velocities (i.e. resultant angular velocity), while the second stage required both the norms of integrals (i.e. change in trunk angle) and derivatives (i.e. angular acceleration) of roll and pitch angular velocities. The second stage was used to deal with the effect of overlap between fall and non-fall peaks when using only resultant angular velocity. The addition of the second stage increased the specificity of the algorithm from 97.5% to 100% at 100% sensitivity.

2.2.4. Accelerometer and Gyroscope-Based Fall Detection Algorithms

Fusing accelerometer with gyroscope data provides a more accurate estimate of the sensor's gravity vector, and hence, a better estimate of the kinematic components of acceleration (and consequently vertical acceleration [18]) [1] and also posture. Accuracy in estimating vertical acceleration is crucial in reliable estimation of vertical velocity since the integration of any error from the former will result to drift in vertical velocity. Without gyroscope, low-pass filtering is used to separate the gravitational and kinematic components, of the accelerometer signal [21, 15, 35, 14], which is unreliable during dynamic activities[2] [19]. Other methods for estimating vertical velocity directly integrates

[1] The gravitational and kinematic components, are linearly combined in the accelerometer data [47]. Estimating the vertical acceleration requires, first, estimating the gravitational vector, to separate the gravitational and kinematic components of acceleration, and second, getting the vertical component of kinematic acceleration, whose direction is parallel to the direction of gravity. Gravitational vector is estimated by relying on the accelerometer data during quasi-static kinematic conditions, and on the gyroscope data during dynamic kinematic conditions [19, 18].

[2] Since posture is used to detect the lying position after the impact of a fall (in trunk-based fall detection algorithms), wherein the body's movement is expected to be minimal [18], improved accuracy in posture estimation may not have a significant effect to the algorithm's performance.

the difference between the accelerometer norm and 9.81 m/s^2, which does not remove the horizontal components of acceleration [16, 20, 17].

Bourke *et al.* and Lee *et al.* [19, 18] proposed an algorithm using only vertical velocity for pre-impact fall detection with the use of both accelerometers and gyroscopes. The difference between the algorithms was with how the gravitational acceleration was estimated and also with the drift reduction method. Bourke *et al.* [19] estimated gravitational acceleration by low-pass filtering the accelerometer data during quasi-static activities, while strapdown integration of gyroscope data was implemented during dynamic activities. The sensitivity and specificity of the algorithm was both 100%[3].

Lee *et al.* [18] estimated the gravitational acceleration by fusing accelerometer and gyroscope data using a tilt Kalman filter that was discussed in [37]. In addition to ADLs, the algorithm was also tested using near-falls, which are also of common occurrence [28, 29]. When differentiating only against ADLs (i.e. fall vs ADL), algorithm's sensitivity and specificity was 97.4% and 99.4%, respectively, while when differentiating against both ADLs and near-falls (i.e. fall vs. non-fall), it decreased to 95.2% and 97.6%, respectively, which shows the additional challenge in correctly classifying such events[4]. The algorithm was also compared to a peak acceleration-based algorithm (using norm of kinematic acceleration) and results show that vertical velocity was better than the latter when differentiating against near-falls. Sensitivity and specificity for fall vs. ADL was 98.7% and 99.4%, respectively, while for fall vs. non-fall (i.e. including both ADLs and near-falls) it was 84.0%, and 85.5%, respectively.

For wrist-based fall detection, Hsieh *et al.* [23] proposed an algorithm using accelerometers and gyroscopes on both wrists. The accelerometers were used to detect the fall's impact using acceleration norm and its standard deviation; and the body's stationary state after impact using signal magnitude area. At the same time, the gyroscopes were used to capture the arm's swing or turning using the norm of gyroscope data (but excluding the wrist rotation). The algorithm was tested including

[3] Upon comparing the vertical velocity profiles to an optical motion capture camera, it was concluded that the method's accuracy makes it a suitable replacement for optical motion capture camera for measuring vertical velocity profiles at home [19].

[4] Compared to Bourke et al.'s aforementioned study, Lee et al. included stand-to-sit fall trials which has a lower starting fall height compared to other fall types, which could explain the algorithm's lower accuracy [18].

activities that could potentially cause false alarms such as clapping, arm and wrist rotation, jumping, and lying down. The sensitivity and specificity and specificity of the algorithm was 95.0% and 96.7%, respectively.

2.2.5. Accelerometer and Barometric Pressure Sensor-Based Fall Detection Algorithm

Bianchi *et al.* [15] proposed an algorithm that used accelerometer and barometric pressure sensor data from trunk-mounted sensors, where the pressure data were used as an alternative to altitude data. The parameters used were kinematic acceleration's norm and signal magnitude area (SMA), posture (uninitialized) and normalized differential pressure (as an alternative for change in altitude). Kinematic acceleration was obtained by subtracting to the accelerometer data its low-pass filtered version (i.e. its gravitational acceleration). To allow detection of falls with low impact (e.g. slow falls), an alternative path was provided in the algorithm. The algorithm was compared to simpler algorithms using only accelerometer, and was tested on both indoor and outdoor environments, and using simulated free-living ADLs. For the indoor experiment, using barometric pressure sensor provided a sensitivity and specificity of 97.5% and 96.5%, respectively, whereas using only accelerometer it was 75.0% and 96.5%, respectively.

2.2.6. Algorithm Performance Using Data from Frail Older Adults

Bagala *et al.* [26] evaluated 13 existing threshold-based fall detection algorithms for a waist-based accelerometer using real-world fall data from older adults with supra-nuclear palsy (total of 29 sets of data). The results show that the accuracy were substantially lower compared to the ones from simulated laboratory trials. Factors contributing to this difference in performance include the tuning of thresholds using participants with different age, mass, clinical history and diseases during the simulated laboratory trials. Additionally, some fall phases that were observed in simulated laboratory trials were not detectable during real-world falls. Among the algorithms, Bourke *et al.*'s velocity + impact + posture [17] provided the best trade-off between sensitivity and specificity, which was at 83% and 97%, respectively.

2.3. Improving Algorithm's Performance

The accuracy of a threshold-based fall detection algorithm heavily relies on the parameters' collective discriminative capacity. The goal in selecting parameters is that they must be able to maximally differentiate between fall and non-fall events by utilizing the two events' kinematic and signal differences. Although a single parameter may not be capable of completely differentiating between the two events, the complementary effect of using multiple parameters that are monitoring different characteristics and phases of the fall makes the algorithm perform better as a whole.

A way to improve the algorithm's performance is by finding parameters with better discriminative capacity through developing new ones or through improving or modifying existing ones. Developing new parameters could be done by finding new kinematic and/or signal differences between fall and non-fall events (and a reliable method to estimate them), or by modifying existing parameter/s to further amplify such signal differences. Improving the reliability of a parameter's currently unreliable estimation method could also potentially improve its discriminative capacity.

Additionally, even if the algorithm has already achieved very high or 100% sensitivity and specificity, further improving the discriminative capacity of its parameters could potentially make the algorithm more robust in dealing with the variabilities in fall and non-fall kinematics.

Bourke *et al.*'s [17] vertical velocity + impact + posture algorithm produced 100% sensitivity and specificity during simulated laboratory-based falls, and also had the best trade-off between sensitivity (83%) and specificity (97%) when tested using real-world falls [26]. However, the sole use of accelerometers for estimating vertical velocity has inherent limitations in accuracy [19, 16]. Using both accelerometers and gyroscopes allow a better estimation of gravitational acceleration, and hence, a better estimate of kinematic acceleration, and consequently, vertical velocity [19, 18]. The addition of barometric pressure as an alternative for altitude improved the sensitivity and specificity of an accelerometer-based algorithm by Bianchi *et al.* from 75% and 91.5% to 97.5% and 96.5% [15], respectively.

Lee *et al.* [18] showed that peak acceleration's performance became lower than that of vertical velocity when falls were also differentiated from near-falls in addition to

ADLs. Such performance shows the limitations of not having near-falls in an experiment protocol especially that such events are commonly occurring [28, 29] and could be challenging to properly classify due to the associated abrupt movement [18].

Among the aforementioned methods to estimate vertical velocity, the best method was with the use of accelerometers and gyroscopes, which although is better than using accelerometer alone, still has limitations in terms of drift. Zihajehzadeh *et al.* [32] fused accelerometer, gyroscope, and barometric pressure sensor data using a cascaded Kalman filter and was able to provide an accurate and drift-free estimate of vertical velocity and altitude. With all the aforementioned explanations, an easily seeable method that could potentially result to an improved performance would be an algorithm that uses vertical velocity, impact, change in altitude, and posture, by employing accelerometers, gyroscopes and a barometric pressure sensor, wherein the data from the three sensors are fused using a cascaded Kalman filter. The research started with this method and proceeded with the development of better parameters to further improve the algorithm's performance. It involved a rigorous iterative process of observing, modifying, developing and tuning of parameters and other aspects of the algorithm using the data gathered during the study.

2.3.1. Other Tools for Analyzing Algorithm's Performance

Being able to analyze the performance of the individual parameters of the algorithm allows a more detailed understanding of the algorithm's discriminative capacity and at the same time are important tools towards improving its performance.

Receiver Operating Characteristic Curves

A common way to compare the classification performance of individual parameters is through comparing their receiver operating characteristic (ROC) curves (see Figure 2.1) [18]. An ROC curve illustrates the trade-off between sensitivity and specificity as the parameter's threshold is swept across the entire distribution of the parameter's maximum-magnitude peaks (positive for parameters that are positive-valued during falls and vice-versa) from all the trials of the data set. Each point in the ROC curve represents the corresponding sensitivity (y-axis) and 1-specificity (x-axis) for a specific value of threshold [38]. The closer the curve could reach the upper left corner (x = 1, y = 0) the better the parameter is in differentiating between falls and non-fall events

[38], and a curve that reaches such corner could completely differentiate between the two events .

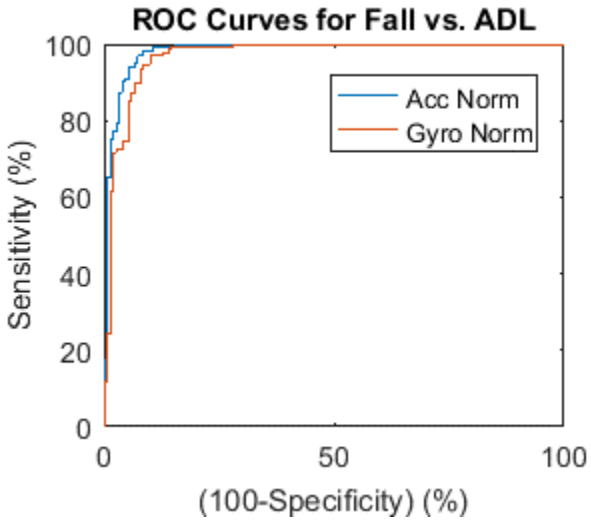

Figure 2.1. Example of plot comparing the ROC curves of accelerometer norm (Acc Norm) and gyroscope norm (Gyro norm) parameters

Comparing the areas under the curve (AUC) is a way to compare the discriminative capacity of the parameters [38]. However, in terms of contribution to a fall detection algorithm, only one point from a parameter's ROC curve matters, which is the sensitivity and specificity corresponding to the threshold used in the algorithm. In fall detection research, examples of thresholds that had been used correspond to maximum specificity at 100% sensitivity (intersection of curve and 100% sensitivity or minimum of fall peaks) [22, 36, 17], maximum sensitivity at 100% specificity (intersection of curve and 0% 1-specificity or maximum of non-fall peaks) [39, 19, 20, 14], and an optimum trade-off between the two measurements [18]. Tuning the thresholds for 100% sensitivity makes sure that no falls will be missed; however, its disadvantage is that if there will be frequent false alarms[5], the alarms may end up being ignored by the responder [7]. On the other hand, if the algorithm is tuned for 100% specificity, although there will be no false alarms, it could cause falls to be undetected, which could result to loss of confidence in the fall detector even with a single missed emergency [7].

[5] Since daily events are dominated by non-fall events, a deviation from 100% specificity could translate to more false positives compared to the false negatives that will result from the same amount of deviation from 100% sensitivity. However, it should be noted that the probability of events being misclassified also depends on the types of activities that are conducted (where abruptness of movement is a big factor to false alarms) and on the manner the fall occurs.

Boxplots

An ROC curve could only compare the overall performance of parameters. As each parameter could be monitoring different characteristics and phases of falls, each could have specific strengths and weaknesses for the different activity types within the data set. Generating boxplots of parameter's maximum-magnitude peaks according to trial types (see Figure 2.2) allow seeing which specific trial-type the parameter is good or bad at in classifying, and also tells how separated the distribution of peaks of falls are from that of non-fall events [22, 36, 18]. Additionally, comparing the sets of boxplots of all the parameters allow seeing how each parameter contributes to the performance of the whole algorithm. Furthermore, the superimposition of thresholds on the boxplots shows which specific activity type will be affected when threshold is adjusted.

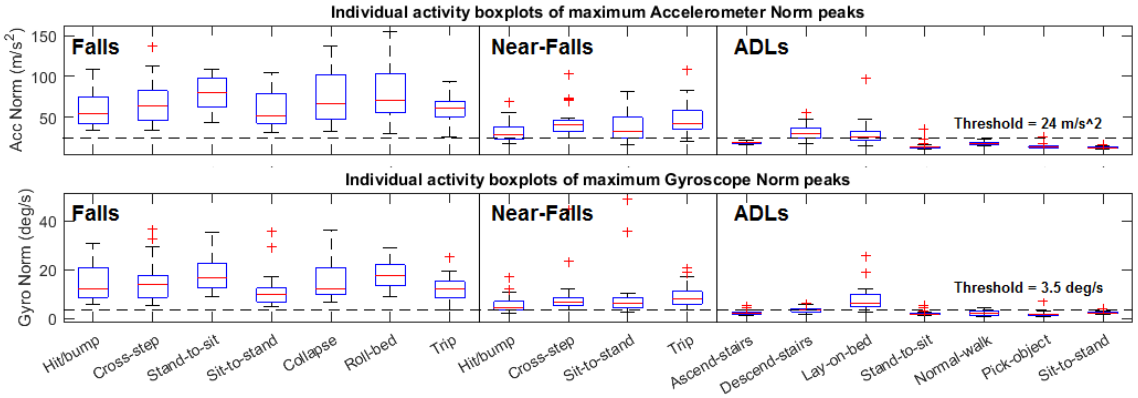

Figure 2.2. Example boxplots of individual activity types for accelerometer norm (above) and gyroscope norm (below) from simulated laboratory-based trials using body-mounted accelerometers and gyroscopes

Chapter 3.

Experimental Data Collection

3.1. Experiment Protocol

Experiment data from 15 young and healthy volunteers (age 23.00 ±3.98 years) were gathered for the study. The experiment protocol consisted of simulated falls, near-falls, and ADLs conducted in a laboratory setting (Experiment 1), and continuous scripted free-living ADL trial in a prepared environment (Experiment 2). All the experiments conducted for this study were a joint experiment together with researchers from Injury Prevention and Mobility Lab (IPML) at Simon Fraser University, Burnaby, British Columbia.

Although real-world fall data from older adults is the most ideal dataset for validating fall detection algorithms, gathering sufficient amount of such data for this thesis is practically impossible to carry out within the timeline of a MASc project. This is because real-world falls are challenging to record due to their low incidence and there are limitations in recording periods [40][6][7].

3.1.1. Simulated Laboratory-Based Trials (Experiment 1)

The simulated laboratory-based trials consisted of 7 types of falls, 9 types of activities of daily living (ADL), and 4 types of near-falls (see Table 3.1) wherein each trial is repeated twice by each participant. For each trial, the participants were asked to stand normally for 15 seconds before the event to allow settling of initial signal processing transients for some parameters and to give room for possible need for parameter initializations[8]. After the event, participants were also asked to stay in their last position

[6] For example, recording 100 falls requires recording of 100,000 days of physical activity [40].

[7] Recently, a consortium (i.e. FARSEEING) was able to build a database of sensor data from real-world falls of older adults [40]. The sensors used were only attached either on the lower back or the thigh. Additionally, barometric pressure sensors were not employed.

[8] Since the experiments were conducted before the algorithms were completely developed, and also for its possible use in future research, a full 15 seconds was captured to make sure that sufficient room is available in case it will be needed.

in a relaxed manner for another 15 seconds to allow for post-fall signal analysis in the algorithm. Except for roll-out-of-bed, the fall trials were those that commonly occur in Long-Term-Care (LTC) facility based on video evidence [41]. For these trials, the participants were shown example videos of real-world falls from older adults[9] and were guided by a research assistant from IPML (see the footnote[10]). For roll-out-of-bed trials, the participants were simply instructed to roll from the bed onto the landing surface. Additionally, participants were not given instructions regarding direction of fall to allow variability in fall kinematics [18] and to help falls occur as natural as possible. The ADLs include activities that could potentially cause false alarms for both wrist-mounted and trunk-mounted sensors (e.g. washing hands, stand-to-sit, and lying on bed). Near-falls are successful recovery attempts from a possible fall, that also commonly occur in older adults [28, 29], and could potentially cause false alarms due to the associated abrupt movement [18]. Among the fall types, collapse, roll-out-of-bed, and sit-to-stand trials were not included as near-fall types because (a) it is almost impossible to recover stability from such activities, and, (b) such activities do not lead to near–fall events. Additionally, since wash-hands and reach-object trials does not involve movement of the trunk, both trials were excluded from the trunk-based fall detection study (i.e. Study 2).

[9] Example of sit-to-stand video may not be part of this.

[10] The research assistant was involved in video analyses of real-world falls that were captured from long-term care facilities, as described in [41].

Table 3.1: Types of trials conducted during simulated laboratory-based falls, near-falls, and ADLs experiment (Experiment 1)

Falls	Near-Falls	ADLs
1. Trip 2. Hit/Bump 3. Loss of consciousness/collapse (collapse) 4. Incorrect transfer due to gait variability misstep/cross-step (cross-step) 5. Incorrect transfer while descending from standing to sitting (stand-to-sit) 6. Incorrect transfer while rising from sitting to standing (sit-to-stand) 7. Roll out of bed (roll-bed)	1. Trip 2. Hit/Bump 3. Incorrect transfer due to gait variability misstep/cross-step (cross-step) 4. Incorrect transfer while rising from sitting to standing (sit-to-stand)	1. Normal walk 2. Rising from sitting to standing (sit-to-stand) 3. Descending from standing to sitting (stand-to-sit) 4. Reach and pick an object from the ground (pick-object) 5. Reach for an object above head height (reach-object) 6. Washing hands (wash-hands) 7. Descending from standing to lying on bed (lay-on-bed) 8. Ascending stairs (ascend-stairs) 9. Descending stairs (descend-stairs)

All of the fall and near-fall trials were performed on top of a 30 cm thickness gymnasium mattress[11] with a layer of 13 cm high density ethylene vinyl acetate foam placed on top to make the surface stiff enough to allow for stable standing and walking, but soft enough to reduce impact forces to a safe level[12]. The participants, were asked to wear a helmet and wrist-guards. The roll-out-of-bed fall trials were performed using a massage table, where two 30 cm gymnasium mattresses stacked on top of each other are placed adjacent to the table to act as a landing surface and to mimic the height of a normal bed. During ADL trials, the trials were conducted on top of a solid wooden

[11] The mattresses were sitting on top of a solid wooden platform that is separated from the ground through casters/rollers (and is also attached to linear motors that allows controlling the platform to move in 2D, but the motor was deactivated). Due to the casters/rollers, during a few of the fall trials (seemingly the ones with very strong impact), there were some horizontal movement in the platform during the impact phase of the fall.

[12] Even with the addition of the dense foam, the impact during walking and especially during a fall were still expected to be affected by the softness of the surface. Additionally, it could also affect the gait of the participant.

platform (i.e. the platform were the mattresses were placed onto). The reach-object and pick-object trials were conducted using the dominant hand.

With issues with the sensor data as they were transmitted from the smartwatches to the tablet (as will be discussed in Section 3.2.1), only the data sets from the last twelve participants were used for wrist-based fall detection study (i.e. Study 1). Likewise, for the second study, only the data sets from the same participants were used.

3.1.2. Continuous Scripted Free-Living ADLs (Experiment 2)

Continuous-scripted free-living ADLs consist of continuous sequence of scripted normal activities in a prepared environment. It includes: sitting, lying down, washing hands, drinking, picking/getting an object from floor/cupboard, plugging an object to the wall outlet, turning on and off lights, opening and closing doors, walking, pausing after walking, and ascending and descending stairs[13]. The activities were done continuously in three different rooms where one was located in on upper level of the building. The trial lasts for approximately 15 minutes for each participant. The complete sequence of activities are enumerated in Table 3.2.

[13] Experiment 2 was designed by researchers from Injury Prevention and Mobility Laboratory.

Table 3.2. Sequence of events that were performed in the continuous free-living ADL experiment (Experiment 2)

Sequence	Location	Activities
1	Room1a (IPML's platform room)	1. Sit on chair for 2 min. 2. Walk to room 1b
2	Room1b (IPML's BOB room) [14]	1. Turn on lights 2. Sit at desk for 2 min., where after the 1st minute, ask the participant to pretend to drink 3. Take the plug from the table and plug it into the wall outlet close to doorway 4. Turn off lights 5. Open door and exit room 6. Walk to room 2
3	Room 2 (Kinesiology lounge)	1. Enter room 2. Turn on lights[15] 3. Sit down on sofa for 2 min. 4. Walk toward cupboard 5. Open cupboard (above head height), reach for an item and bring item to table 6. Sit on chair at the table for 2 min. 7. Return item to cupboard 8. Turn off light 9. Open door and exit room 10. Walk down the corridor, then upstairs (need to open doors to get in and out of the stairs), then walk down the corridor towards room 3
4	Room 3 (copy room upstairs)	1. Enter room 2. Turn on lights 3. Lie down on sofa for 2 min. 4. Walk toward sink and wash hands 5. Sit on sofa chair for 2 min. 6. Have the participant reach down and pick up an item while sitting 7. Turn off light 8. Open door and exit room 9. Walk straight down corridor, pause at the end for 30 seconds, then walk down the stairs and return to room 1a
5	Room 1a (platform room)	1. Enter room 2. Sit on chair

[14] Note that there were no doors that need to be opened between rooms 1a and 1b.

[15] There were some instances where the room was being used, and hence, the lights were already turned on.

The experiment protocol was approved by the Research Ethics Committee at Simon Fraser University (reference number: 2012s0233) and all participants provided informed written consent.

3.2. Sensors and Other Equipment Used

The sensors used in the experiments were a combination of sensors used by the author of this thesis and IPML researchers. The smartwatch's wireless control, and all the synchronization methods described in this section of this chapter were developed by the author.

3.2.1. LG Watch Urbane Smartwatch

For the first study, data from the tri-axial accelerometers (custom range ±8 g, 100 Hz), tri-axial gyroscopes (±2000 deg/sec, 100 Hz), and barometric pressure sensors (50 to 110 kPa, 25 Hz) of three LG Watch Urbane (LG Electronics Inc.) Android smartwatches (see Figure 3.1) were gathered from the experiments. Two of the watches were mounted on both wrists of the participant (see Figure 3.2) and one was mounted on the wall at a known height. Only the data from non-dominant[16] wrist were used for analysis in the first study (except for reach-object and pick-object trials where data from the dominant wrist were used). The smartwatch on the wall was for gathering a reference barometric pressure (will be used for potential future research).

[16] Information regarding which hand is dominant was determined prior to conducting the trials for each participant.

Figure 3.1. LG Watch Urbane smartwatch with custom app in foreground

Figure 3.2. Participant standing on mattress and equipped with smartwatch, Xsens sensors, and optical motion capture camera markers

For synchronization purposes and to prevent the recording of extra movements (due to manually starting of data logging in the watches), a custom Android tablet app was developed to be able to control data logging wirelessly (where pressing a single start/stop button in the custom app starts/stops the logging, respectively, in all of the smartwatches). Data was initially sent in 1-second chunks from the smartwatches to the

tablet for convenience for the first few participants[17], but due to issues with unsent chunks and, in rare cases, improperly ordered arrival of chunks, the data were just eventually stored in the watches[18] for the succeeding participants.

3.2.2. Xsens MTw Sensors

For the second study, data from tri-axial accelerometers (±160 m/s^2), tri-axial gyroscopes (±2000 deg/s) and barometric pressure sensors (300-1100 mBar) of nine Xsens MTw (wireless) sensors (Xsens Technologies B.V.) (see Figure 3.3) were also gathered from the experiments. Six of the Xsens sensors were mounted on the sternum, lower-back, left and right wrists, and left and right thighs (see Figure 3.2), while the remaining three sensors were mounted on fixed locations in the environment. Only the data from the lower back sensors were used for analysis in the second study[19]. All data from the Xsens sensors were wirelessly sent to a laptop using a USB-connected Xsens transceiver.

Figure 3.3. Xsens MTw sensor

For synchronization purposes, the Xsens transceiver was interfaced to the Android tablet through an IOIO OTG development board (SparkFun Electronics) (see Figure 3.4). The triggering of data logging in the tablet through the custom app also

[17] In compressed binary format.

[18] Codes related to sensor data compression and data logging inside the smartwatch were developed by other researchers and software developers.

[19] Between the two sensor locations on the trunk, the lower-back and sternum, the trunk was selected since it will be a more convenient location for real-life use since the sensors could be directly clipped on the belt or pants

sends a TTL signal to the Xsens transceiver which then triggers the Xsens sensors' data logging[20].

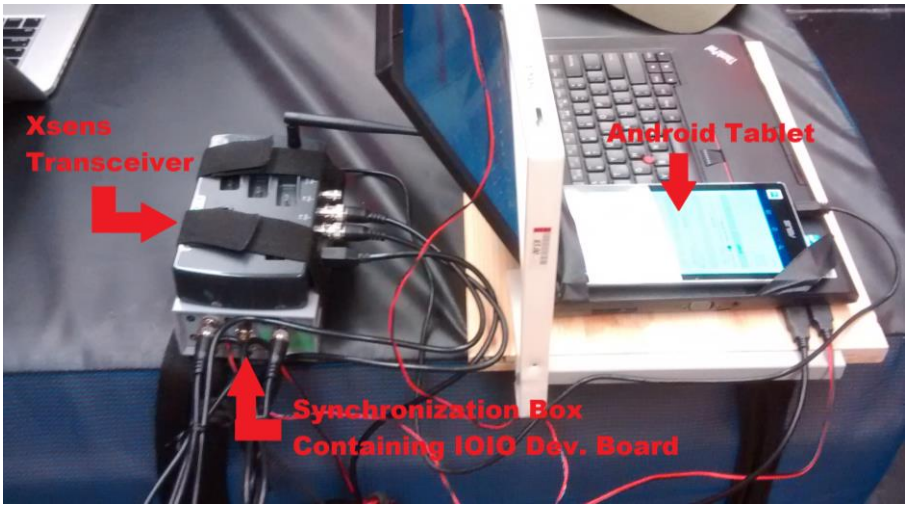

Figure 3.4. Synchronization of Xsens transceiver to the Android Tablet via the IOIO development board

Due to corrupted Xsens sensor data for some of the trials, out of the 168 falls, 168 ADLs, and 96 near-falls for all the 12 participants, only 166, 150, and 95, respectively were available for data analysis.

3.2.3. Optical Motion Capture Cameras/System

Motion of body locations where the sensors were mounted and other locations were also recorded using Eagle optical motion cameras (at 100Hz sampling rate). This data provides an accurate motion reference for validating IMU sensor estimates. The logging of data from the cameras were triggered manually using EVaRT motion capture software (Motion Analysis Corp.), but for synchronization purposes for use in data analysis, the TTL output coming from the Android tablet (via the IOIO board) was also recorded through the EVaRT software via the analog input of its data acquisition board. Since processing data from all the trials could potentially take several months, they were kept for future research.

[20] The TTL signal is sent only once all of the watches already gave feedback to the tablet when they already started/stopped logging.

3.2.4. Video Cameras

Video cameras were used in the experiments to help in mapping specific characteristics observed from the recorded sensor signal and estimated parameters to the participant's motion during data analysis. Two cameras, Allied Vision Prosilica GS (Allied Vision Technologies, GmbH) and a Sony Action Cam (Sony Corporation), were fitted with synchronization capabilities and were used for trials conducted inside (Prosilica GS) and outside (Sony Action Cam) the laboratory[21]. A regular digital video camera (Sony Handycam) was also used for all the trials. Figure 3.5 shows the video recording, data logging control, and synchronization setup used for trials conducted outside the laboratory and during the free-living ADL experiment.

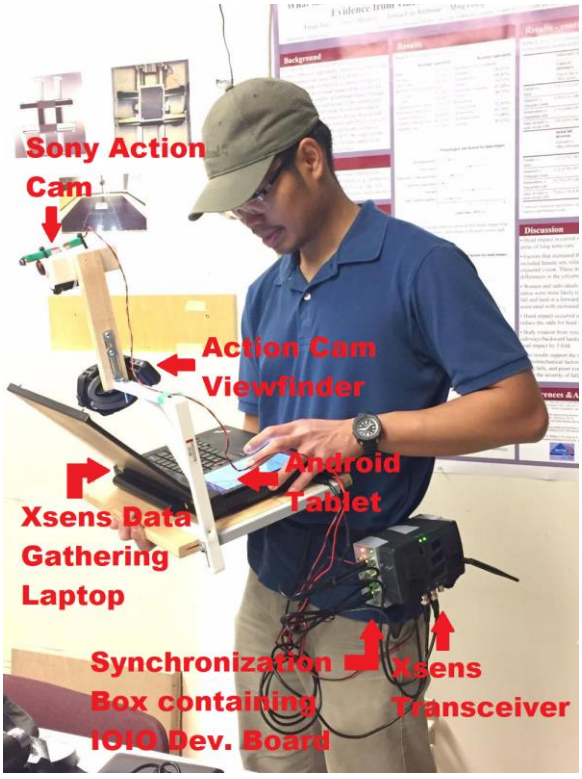

Figure 3.5. Setup for video recording, data logging control, and synchronization for trials conducted outside the laboratory and during the free-living ADL experiment

[21] More information about the synchronization methods developed or employed for the Sony Action Cam and Prosilica GS, respectively could be found in Appendix A. Since the recorded video together with the parameter and sensor signals were already found to be sufficient during data analyses, the synchronization information from both the Sony Action Cam and Prosilica GS cameras were not used.

3.2.5. Synchronization

As mentioned in the previous subsections, the Xsens sensors, motion capture cameras, Prosilica GS camera, and Sony Action Cam camera were synchronized to the smartwatches by interfacing them to an Android tablet via an IOIO development board and by using a custom tablet app.

3.2.6. Post-Processing

Shell scripts were developed to automate the process in part for converting and renaming of the smartwatch and Xsens sensor data files in preparation for data analysis. All data were analyzed using MATLAB R2016b.

Chapter 4.

Study 1: Smartwatch-Based Fall Detection Algorithm Using Accelerometers, Gyroscopes and a Barometric Pressure Sensor

4.1. Abstract

This chapter (Study 1) proposes a threshold-based fall detection algorithm for a wrist worn commercially available smartwatch equipped with tri-axial accelerometer, tri-axial gyroscope and a barometric pressure sensor. The algorithm uses vertical velocity, vertical acceleration, difference in altitude, average vertical velocity, descent signal magnitude area (SMA), post-impact SMA, and forearm angle as parameters. Forearm angle was used to filter the downward vertical orientation of the forearm that could be associated to a non-fall event's post-activity positon. Additionally, to deal with strong disturbances in pressure data from the smartwatch, especially during abrupt movements of the arm, pressure data were used selectively for altitude estimation in a Kalman filter.

The algorithm was tested using simulated laboratory-based trials described in Chapter 3, consisting of 168 falls, 216 activities of daily living (ADL), and 96 near-falls recorded from 12 participants. Results showed that analyzing the forearm angle along with a selective use of pressure data substantially improved the specificity of our fall detection algorithm: 97.2% specificity (from 90.7%) for ADLs, and at least 97.1% specificity (from 78.9%) for non-falls (i.e. including both ADLs and near-falls). Further testing using simulated continuous sequence of free-living ADLs performed by each participant resulted to a total of 3 misclassified non-fall events.

4.2. Introduction

Convenience of use is important for user compliance [42]. The wrist is a convenient location for a fall detector to be worn at as it similar to wearing a watch and does not require frequent removal during activities such as using the washroom or changing clothes [23, 43]. Recently, programmable smartwatches that are equipped with MEMS inertial sensors, barometric pressure sensor, user interface, and wireless

communications capabilities have become available to the consumer market. These capabilities make the smartwatches an available platform for wrist-based fall detection. Additionally, with the programmability of these smartwatches, other health related apps and functionalities that could benefit the older adult population can also be added to them.

However, detecting falls using wrist-worn sensors is a challenging task due to the hand's diverse functionalities, and the arm's dynamics of motion and articulation. Consequently, the wrist could easily produce big or abrupt movements during some ADLs. Additionally, it could only give a limited amount of information about the body's general motion. Using only accelerometers, Degen *et al.* [16] implemented a wrist-based fall detection algorithm employing norm of acceleration and two forms of vertical velocity estimate as parameters, and were only able to detect 65% of all the fall trials. Using the same type of sensor, Kangas *et al.* [14] compared the performance of multiple algorithms of different complexities using sensors at different locations, and reported the maximum sensitivity of 71% (at 100% specificity) for the wrist, whereas for head and waist, the sensitivity was about 98% and 97%, respectively. Using accelerometers and gyroscopes, Casilari *et al.* [24] used sensors on the thigh in addition to the wrist sensors and reported the best combination of sensitivity and specificity of 96.7% and 98.3%, respectively, among the algorithms tested. However, when using only sensors on the wrist, sensitivity and specificity was reduced to 93.3% and 93.3%, respectively. Using the same type of sensors, Hsieh *et al.* [23] reported an average sensitivity and specificity of 95% and 97%, respectively. However, sensors on both hands were required, which is impractical.

In addition to the decision-making structure, accuracy in threshold-based fall detection depends on the collective capacity of all the selected parameters to differentiate between fall and non-fall events. The goal when selecting parameters is to maximize the kinematic and signal differences between the two event types, and at the same time, parameters need to be estimated in a reliable way. The limited accuracy of an existing algorithm could be improved by developing new parameters (or by improving existing ones) that could contribute in addressing the existing algorithm's inadequacy.

From previous wrist-based fall detection studies, only accelerometers [16, 14] or a combination of accelerometers and gyroscopes [23] were used. Fusing barometric

pressure sensor with accelerometer and gyroscope is a robust way for drift-free estimation of altitude as well as vertical velocity [32]. In addition, from a study by Bianchi *et al.* [15], adding change in altitude as one of the parameters in a waist-based algorithm increased the algorithm's accuracy from 85% to 97%. On the other hand, barometric pressure data from a waterproof smartwatch could become inaccurate and unreliable during dynamic motion of the arm. This is mainly due to the sealed waterproof enclosure of the smartwatch. Thus, for a smartwatch-based fall detection algorithm, pressure data should be selectively used.

Forearm angle is a parameter that could potentially improve the discriminative capacity of a wrist-based fall detection algorithm. Although the forearm could exhibit abrupt movements during non-fall events, its post-activity equilibrium orientation is a downward vertical orientation (DVO), unless it is resting on a structure or sustained effort is given to hold the forearm in a different orientation. On the other hand, right after the impact of a fall, the forearm is expected to be outside the downward vertical orientation unless sustained effort is made to straighten the whole arm downward starting from the descent, which could be difficult especially during impact. This expected quasi-complementary post-activity orientation of the forearm between a fall and non-fall event can potentially be utilized to differentiate between the two events.

This chapter proposes a threshold-based fall detection algorithm using a commercially available smartwatch equipped with tri-axial accelerometer, tri-axial gyroscope, and a barometric pressure sensor. The monitored parameters include vertical velocity, vertical acceleration, difference in altitude, average vertical velocity, descent signal magnitude area (SMA), post-impact SMA, and forearm angle. Forearm angle was used to filter the downward vertical orientation of the forearm that can be associated to a non-fall event's post-activity positon. To deal with disturbance in pressure data of the smartwatch, pressure data were used selectively.

The novelty of the proposed wrist-based fall detection algorithm in this chapter lies in: a) the fusing of tri-axial accelerometer, tri-axial gyroscope, and barometric pressure sensor data of a commercially available smartwatch using a cascaded Kalman filter to obtain vertical velocity and changes in altitude estimates; b) selective use of barometric pressure data to deal with pressure disturbances due to sealed enclosure of the smartwatch; and differentiating between fall and non-fall events based on c) quasi-

complementary post-activity orientation of the forearm between the two events using forearm angle, and on d) the difference in wrist movement variation between the two events using SMA.

4.3. Methodology

4.3.1. Parameters

Vertical Acceleration

During the impact phase of a fall, a sudden spike in the wrist's vertical acceleration is generated as the forearm from its maximum downward velocity (at the end of descent) is suddenly put to a halt. *Vertical acceleration (* ${}^{I}a_z[k]$*)* was used to detect this acceleration spike during impact, whose occurrence also serves as a time reference for subsequently monitored parameters. It is the 3rd component of the gravity-compensated acceleration (${}^{I}\boldsymbol{a}[k]$) in the inertial reference frame (I) at time step k. ${}^{I}\boldsymbol{a}[k]$ was calculated using

$$ {}^{I}\boldsymbol{a}[k] = {}^{I}_{S}\boldsymbol{R}[k]\left(\boldsymbol{y}_A[k] - g\ {}^{S}\boldsymbol{x}_1[k]\right) \tag{4.1} $$

where ${}^{I}_{S}\boldsymbol{R}$ is the tilt angles-based rotation matrix that aligns the z-axis of the sensor reference frame (S) to the inertial reference frame, $\boldsymbol{y}_A[k]$ are the accelerometer signals, g is the magnitude of gravitational acceleration, and ${}^{S}\boldsymbol{x}_1[k]$ is the normalized gravity vector in the sensor reference frame. ${}^{I}_{S}\boldsymbol{R}$ and ${}^{S}\boldsymbol{x}_1[k]$ were estimated by fusing accelerometer and gyroscope data using a tilt Kalman filter by Lee et *al.* [37, 32]. Without gyroscope, accelerometer data are typically low-pass filtered to estimate the gravitational vector, which is unreliable during dynamic activities [19]. Accurate estimation of vertical acceleration is important for accurate calculation of vertical velocity.

Vertical Velocity and Difference in Altitude

During the descent phase of a fall, the downward velocity of the wrist is expected to have a higher magnitude compared to ADLs. *Vertical velocity (* ${}^{I}v_z[k]$*)* was used to detect this quick downward movement of the wrist and was estimated using the vertical position and velocity Kalman filter (VPV Kalman filter) by Zihajehzadeh *et al.* [32]. The

inputs to the VPV Kalman filter are the tilt Kalman filter's outputs and the barometric pressure data (see Figure 4.1). The cascading of the tilt Kalman filter and the VPV Kalman filter allows an accurate and drift-free estimate of vertical position and velocity [32][22].

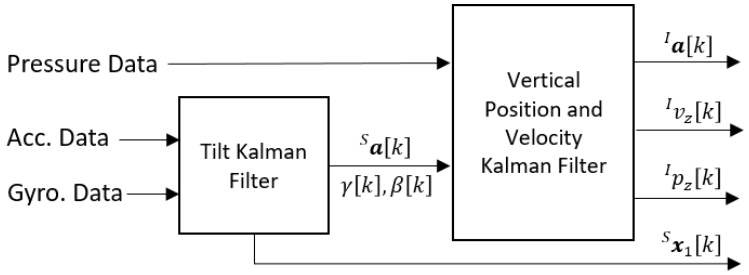

Figure 4.1. The tilt and vertical position and velocity (VPV) Kalman filters that were used to estimate the variables used in calculating the algorithm's parameters

In the same phase of a fall, the wrist is also expected to undergo a substantial change in elevation from a higher position to a lower position, before descent phase and right after impact phase, respectively. *Difference in altitude (*$\Delta h[k]$*)* was used to capture this elevation difference of the wrist and was estimated as

$$\Delta h[k] = {}^{I}p_z[l] - {}^{I}p_z[l - m + n] \tag{4.2}$$

$$l = k - (k\%n) \tag{4.3}$$

where ${}^{I}p_z[l]$ is the estimated vertical position using the VPV Kalman filter at time-step l (4.3), m is the number of time-steps within a 4-second window (i.e. $4\ sec \times 100\ Hz\ sampling\ rate$), and n is the number of time steps within the pressure data's 0.5-second rolling-average window (i.e. $0.5\ sec \times 100\ Hz\ sampling\ rate$) [32].

During preliminary analyses, it was found that dynamic motion of the wrist causes strong disturbances in the pressure data of the smartwatch, which makes it unreliable for the altitude measurement update (AMU) step of the VPV Kalman filter. In this chapter, the rolling average pressure [32], and consequently AMU, was only used whenever the difference between the maximum and minimum pressure within a 0.5-

[22] A general explanation about the mechanism behind the tilt and VPV Kalman filters used in this study could be found in Appendix B.

second window was lower than or equal to the pressure stability threshold of 11 Pascals[23] for two consecutive windows. This reliability criterion-based use of pressure data is referred to as selective use of pressure (SUP) in this chapter. Additionally, due to the gaps in reliable pressure data, ${}^{I}p_z$ in (4.2) will be the last ${}^{I}p_z$ with AMU before the time steps $[l]$ and $[l-m+n]$, respectively.

In between AMU steps (at 2 Hz) of the VPV Kalman filter and in the absence of reliable pressure, zero velocity update (ZVU) technique [44] was used to reduce the drift in vertical velocity/altitude estimation. ZVU was activated when the variance of a 0.5-second window of external acceleration norm was less than 0.05 $(m/s^2)^2$ (i.e. when wrist motion was almost static) [24]. This method for checking static conditions provided a more sensitive and consistent ZVU activations compared to [32] for wrist-based devices. Additionally, not directly thresholding the norm of acceleration prevented ZVU activations during free-fall-like descending motions of the wrist. Furthermore, when the condition for ZVU was not met, ${}^{I}\boldsymbol{a}[k]$ was high-pass filtered (using a 2nd order elliptical filter with 0.25 Hz cut-off frequency, 0.01 dB pass-band ripple, and 100 dB stop-band attenuation). This filter removes the DC component of the external acceleration (including the sensor's time varying bias) that could result in drift in vertical velocity. Figure 4.2 shows the block diagram of the algorithm used for vertical velocity/altitude calculation.

[23] This threshold was determined by analyzing the experimental data from the barometric pressure sensor.

[24] This statistic and threshold for checking static conditions were previously developed by Ginelle Nazareth who was previously with Biomechatronic Systems Laboratory.

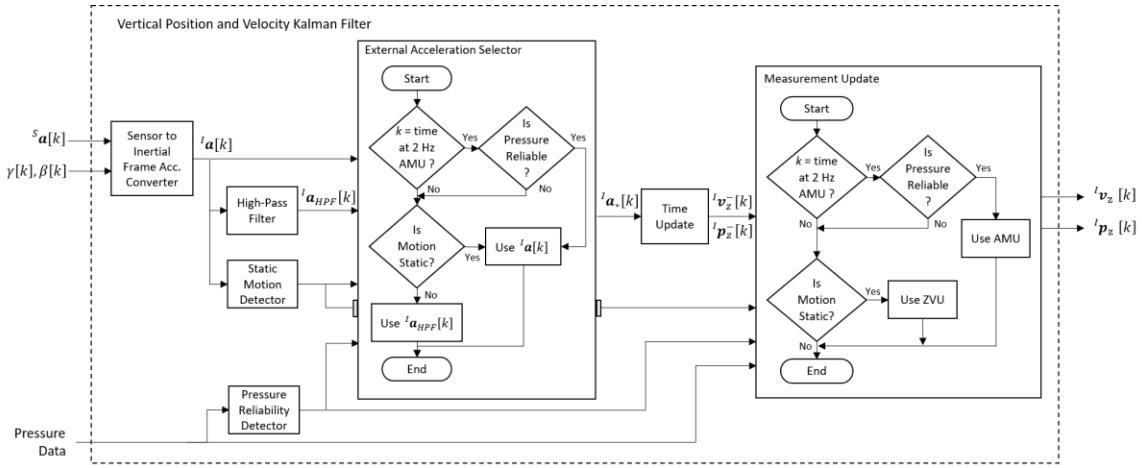

Figure 4.2. Block diagram of vertical position and velocity (VPV) Kalman filter in Study 1 which emphasizes the drift reduction logic

Average Vertical Velocity

During attempts to recover balance during near-falls, and also in some ADLs (e.g. drying hands after washing), although the associated abrupt motions could produce a significant amount of negative vertical velocity, their duration is short compared to the longer (i.e. starting from descent to impact) and generally stronger downward vertical velocity during falls. By taking the average of vertical velocity from both events, it is expected that the resulting signal profile from such abrupt non-fall events will be more attenuated compared to that in falls. *Average Vertical Velocity* was used to capture such longer and generally stronger negative vertical velocity profile during falls. It was calculated by taking the mean of the vertical velocity within a 1-second window.

Forearm Angle

Similarly, although non-fall events may exhibit big or abrupt hand movements, it is expected that the forearm's post-activity equilibrium position is the downward vertical orientation (DVO). On the other hand, due to the sustained effort required to straighten the whole arm downward starting from descent (which could be difficult especially during impact), it is expected that the forearm will be away from this orientation right after impact. *Forearm Angle (*$FA[k]$*)* was used to detect the arm being away from the downward vertical orientation right after a fall's impact. It was the 1-second mean of the estimated angle between the forearm's length and direction of gravity (see Figure 4.3a), and was estimate using

$$FA[k] = \frac{1}{F_s} \sum_{l=k-F_s+1}^{k} \cos^{-1}\left({}^{S}x_{1\_x}[l] \right) \frac{180}{\pi} \tag{4.5}$$

where ${}^{S}\boldsymbol{x}_1[l]$ is the normalized gravity vector in the sensor frame and ${}^{S}x_{1\_x}[l]$ is its x-component. The accelerometer's x-axis was approximately parallel to the forearm and was directed towards the elbow when the smartwatch was on the right wrist (see Figure 4.3b). When on the left wrist, ${}^{S}x_{1\_x}[l]$ was multiplied to -1. Forearm angle was measured every 1-second interval[25]. The range used for downward vertical orientation in this study was between 0 to 25 degrees, and the forearm angle needed to be outside this range for three measurements within the 5.5 seconds monitoring period. Since ${}^{S}\boldsymbol{x}_1[l]$ was estimated by fusing accelerometer and gyroscope data using a tilt Kalman filter [37, 32], it does not require the wrist to be static for proper estimates. The forearm angle measurements are independent of the arm's axis of rotation.

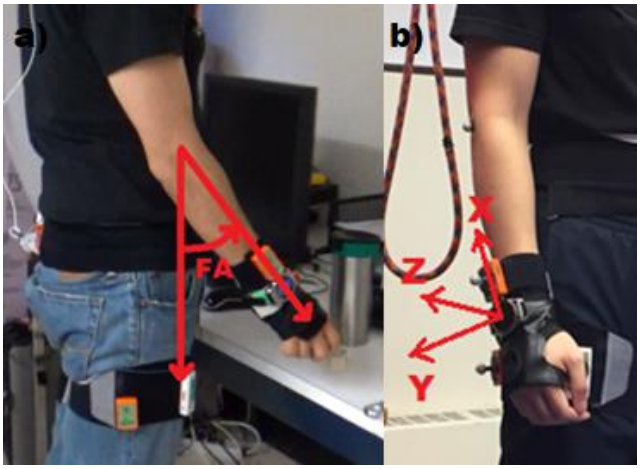

Figure 4.3. Illustration of (a) Forearm Angle measurement and (b) smartwatch's sensor co-ordinate frame

Signal Magnitude Area

Moments after the impact of a fall, it is assumed that the wrist movement will be minimal (or static), especially for frail older adults. *Signal magnitude area (SMA)* was used to detect this minimal movement after impact (post-impact SMA). Additionally, to help differentiate falls from kinematically similar but more controlled non-fall events such

[25] With measurements conducted every 1-second interval, the forearm angle represents the average angle within the last second.

as lying-on-bed, SMA was also used to detect the potentially greater amount of variation in wrist movement during the descent phase of a fall (descent SMA). It was calculated using

$$SMA[k] = \frac{1}{F_s} \sum_{l=k-F_s+1}^{k} \left(\left| {}^{I}a_x[l] \right| + \left| {}^{I}a_y[l] \right| + \left| {}^{I}a_z[l] \right| \right) \tag{4.4}$$

where F_s is the sampling frequency, and ${}^{I}a_x[l]$, ${}^{I}a_y[l]$, and ${}^{I}a_z[l]$ are the x, y, and z components of gravity-compensated acceleration in inertial reference frame (${}^{I}\boldsymbol{a}[l]$), respectively. SMA was calculated continuously for every time step for descent SMA. For post-fall SMA, it was calculated at every 1-second interval.

Apart from vertical velocity, vertical acceleration and difference in altitude, other parameters were determined through an iterative process of testing and modifying existing parameters in literature, and in developing new ones based on analyzing kinematic and signal differences between fall and non-fall events using experiment data. The process was continued until an acceptable algorithm performance was achieved.

4.3.2. Algorithm

Two algorithms, Algorithms 1 and 2, are presented in this section. Algorithm 1 is a simpler version of Algorithm 2 and serves as a reference for performance comparison. Algorithm 2 is the main proposed algorithm. Both algorithms were tested with and without the selective use of pressure (SUP).

Figure 4.4 illustrates Algorithm 1 which used all of the aforementioned parameters except for forearm angle. The algorithm first monitored the vertical velocity to detect descent, and once it reached its threshold, the vertical acceleration was subsequently monitored for 0.8 seconds to detect impact. Detection of impact was the starting point for simultaneous monitoring of average vertical velocity, descent SMA, difference in altitude, and post-impact SMA. When all the aforementioned parameters reached their pre-set threshold value within their monitoring period, the event was considered as a fall.

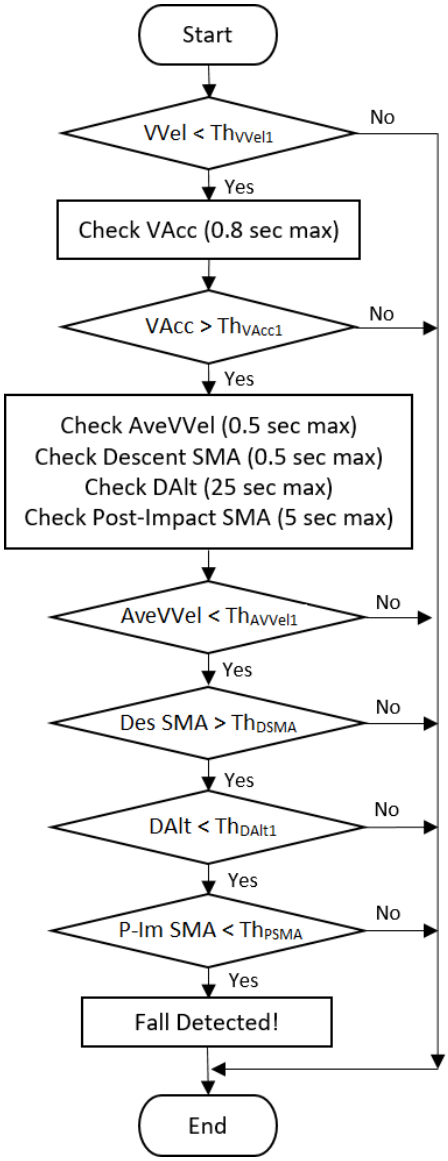

Figure 4.4. Flowchart of proposed Algorithm 1 of Study 1

Figure 4.5 shows Algorithm 2 which was further divided into two paths. The first path incorporates post-descent forearm monitoring in addition to Algorithm 1's parameters (except for average vertical velocity). The second path used the same parameters as Algorithm 1, but excluded the difference in altitude. Additionally, the thresholds for vertical acceleration, vertical velocity, and average vertical velocity of the second path were set to higher values compared to Algorithm 1 to serve as alternative restrictions in place of forearm angle and difference in altitude. The second path was included for the rare cases of falls where the forearm remains in a downward vertical position after impact (e.g. when breaking a fall). It also served as a backup path for falls

with sufficiently strong downward vertical motion of wrist for which the pressure data did not stabilized within its monitoring period.

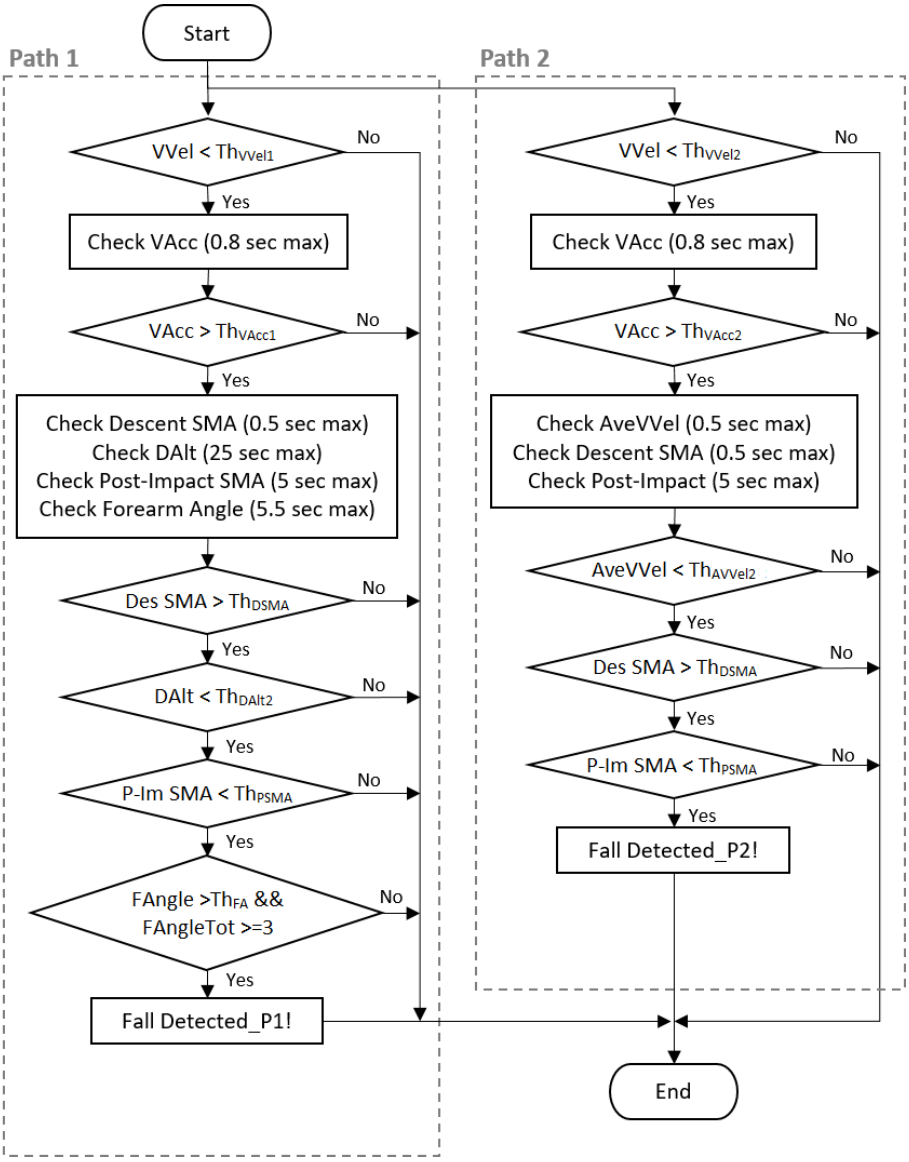

Figure 4.5. Flowchart of proposed Algorithm 2 of Study 1

The thresholds of Algorithm 1 were based on each parameter's minimum-magnitude peak from all the fall trials together with the associated timing requirements using the experimental data and were tuned to achieve 100% sensitivity (when possible) with the highest possible specificity[26]. For post-impact SMA, the threshold was based on

[26] Note that the tuning of threshold and monitoring period for some parameters is in part an iterative process.

SMA's maximum magnitude right after impact, but leniency was given to allow minimal movement of the wrist. For Algorithm 2, thresholds of path 1 were the same as that in Algorithm 1 except for difference in threshold, which during tuning, excluded falls that cannot satisfy the forearm angle (which made its threshold slightly higher). The stricter thresholds of parameters in path 2 allow at least all falls that could not pass path 1, and as much as possible all other fall trials without adding new false positives (if possible). Table 4.1 shows the thresholds used in Algorithms 1 and 2.

Table 4.1. Thresholds of each parameter for Algorithms 1 and 2 of Study 1

Parameter	Algorithm 1	Algorithm 2, Path 1	Algorithm 2, Path 2
Vertical Velocity (m/s)	-0.65	-0.65	-1.5
Vertical Acceleration (m/s^2)	10	10	20
Average Vertical Velocity (m/s)	-0.06	n/a	-0.45
Descent SMA (m/s^2)	10	10	10
Difference in Altitude (m)	-0.18	-0.25	n/a
Post-Impact SMA (m/s^2)	2	2	2
Forearm Angle (degrees)	n/a	25	n/a

The corresponding monitoring periods of parameters for both algorithms are shown in Table 4.2. The sequence and monitoring periods of all the parameters were based on the relative time-occurrence of their fall-associated signal attributes and the signal's properties using the experiment data. All these timing requirements serve as additional fall signatures in addition to the parameter thresholds. Based on our observations, some amount of time could be required (amount of time varies) for pressure to stabilize after the disturbances caused by dynamic movements[27]. Thus, monitoring period for difference in altitude was given a maximum of 25 s, or a maximum of 10 available reliable pressure averages after impact, whichever comes first.

[27] There was one fall trial in the experiment dataset where pressure data did not stabilized within the remaining length of data after fall.

Table 4.2. Monitoring periods of each parameter for Algorithms 1 and 2 of Study 1

Parameter	Monitoring Period	Algo. 1	Algo. 2 Path 1	Algo. 2 Path 2
Vertical Acceleration	0.8 sec. after descent	Yes	Yes	Yes
Average Vertical Velocity	0.5 sec. after impact	Yes	No	Yes
Descent SMA	0.5 sec. after impact	Yes	Yes	Yes
Difference in Altitude	Max. 10 reliable pressure data points or 25 sec. after impact (whichever comes first)	Yes	Yes	No
Post-Impact SMA	5 sec. after impact	Yes	Yes	Yes
Forearm Angle	5.5 sec. after impact	No	Yes	No

An iteration of the algorithm (for both algorithms) was initiated at every time step (i.e. at 100 Hz) to prevent the algorithm from missing the time-instance of any fall event. To save memory resources, only the important contexts, such as the parameter's timing references and logic conditions were stored for each algorithm iteration.

With this iteration frequency and the signal characteristics of parameters during a fall, consecutive fall detections might occur for a single fall event. To prevent such multiple detections, a time difference of 1.5 s was required before a subsequent detection was considered as a new fall event.

4.4. Results and Discussion

4.4.1. Parameter Estimation with and without SUP

Shown in Figure 4.6 are example plots of the parameters used in both algorithms with SUP, using data from one of the collapse fall trials. From standing normally, the participant collapsed and remained on the ground. In Figure 4.6a, the spike in vertical acceleration indicates the wrist's impact to the ground during the fall, and the preceding trough indicates the descent that occurred before it. In Figure 4.6b, SMA indicates the amount of movement of the wrist during the entire fall event. During a fall, the pressure profile is supposed to resemble a sigmoid function due to the sudden change in height,

however, in Figure 4.6c, there is instead a strong disturbance in the pressure signal[28] whose extent of magnitude is far more than the actual amount of descent. Similar to vertical acceleration, vertical velocity in Figure 4.6d also indicates the descent and impact during the fall. The strong downward slope represents the descent (which occurs at the same time as the trough in vertical acceleration), and its sudden return to zero represents the sudden halt in the wrist's vertical motion upon impact (right after the wrist reaches its maximum downward velocity).

[28] Although with different sensitivity to movement, disturbance in pressure sensor data were also noticed in other smartwatches with sealed enclosure (e.g. LG G Watch R and Huawei Watch 2).

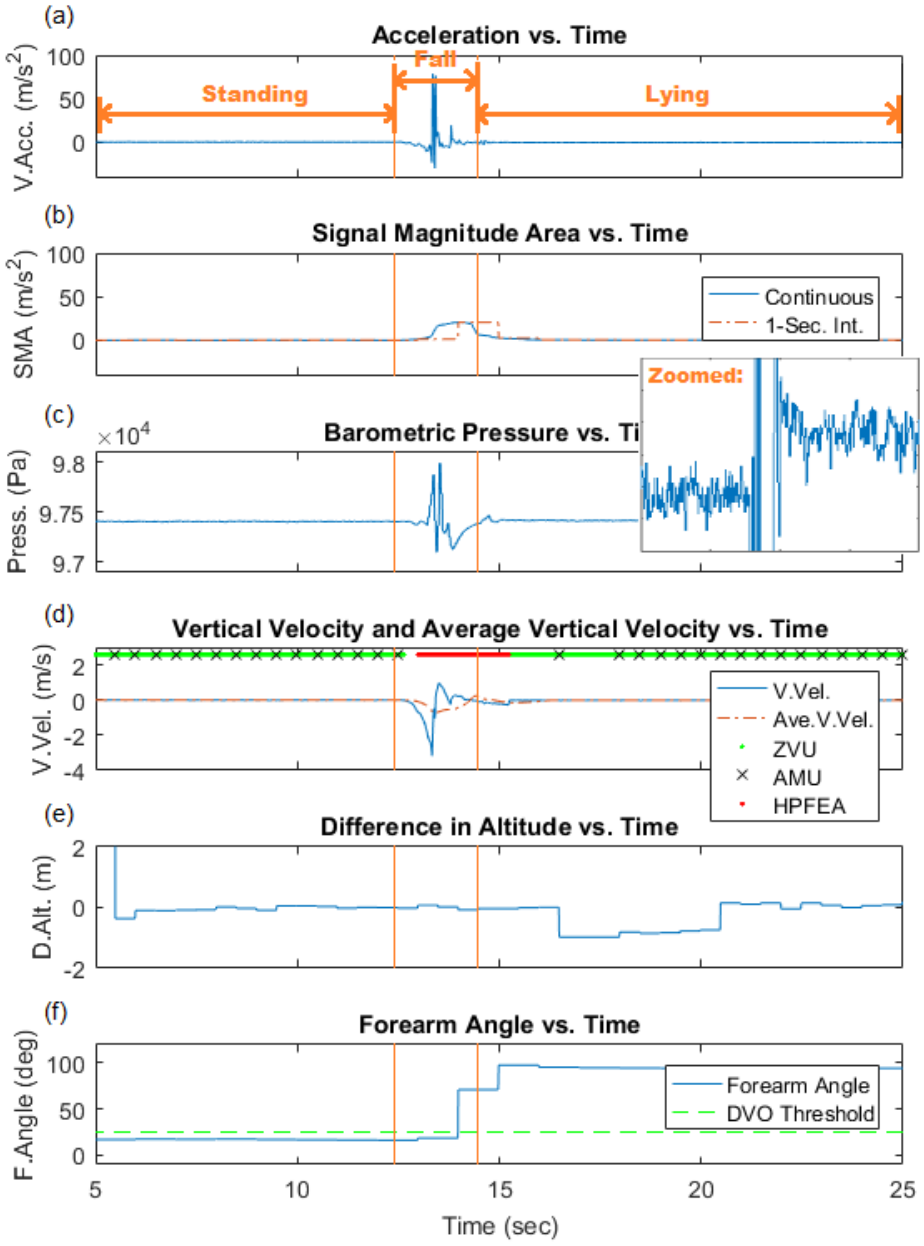

Figure 4.6. Plots of parameters for a collapse trial in Study 1 with selective use of pressure (SUP). (a) is vertical acceleration, (b) is SMA, (c) is barometric pressure, (d) is vertical velocity and average vertical velocity, (e) is difference in altitude, and (f) is forearm angle.

Shown also in the vertical velocity plot (Figure 4.6d) are the indicators for AMU, ZVU, and use of high-pass filtered version of external acceleration (HPFEA) in the VPV Kalman filter. When motion was static and there was no pressure disturbance while the participant was standing, ZVU and AMU, respectively, were both used in the VPV Kalman filter. As both the pressure disturbance reached its limit and motion was no longer considered static, HPFEA was used in the VPV Kalman filter to reduce the drift in

vertical velocity. After the fall, when motion became static and pressure satisfied its stability criteria, ZVU and AMU were resumed, respectively.

Even with pressure disturbance, information indicating the elevation change during the fall was still available in difference in altitude (Figure 4.6e) and appeared as a rectangular notch once pressure regained its stability. The notch's 4-second width was caused by the parameter's window-width. For fall detection purposes, being able to detect this change even with the delay is already sufficient. Figure 4.6f also shows that the forearm angle was within the DVO while the person was standing and then goes out of the range after the participant fell.

For the same collapse trial, when SUP was not used, the distortion in both vertical velocity (Figure 4.7d) and difference in altitude (Figure 4.7e) was large, and the fall profile's signature in both parameters were no longer identifiable. Vertical velocity and difference in altitude were forced in both directions depending on the magnitude and direction of pressure's disturbance.[29]

[29] Of the three smartwatches that were used in the experiments, one of them almost didn't had pressure disturbance, wherein for the trials where it had during abrupt movements, it was very minimal. Consequently, the difference with and without SUP was also minimal for difference in altitude and vertical velocity. This smartwatch was only used on one of the twelve participants for the non-dominant wrist and the reason for the more stable pressure data cannot be explained.

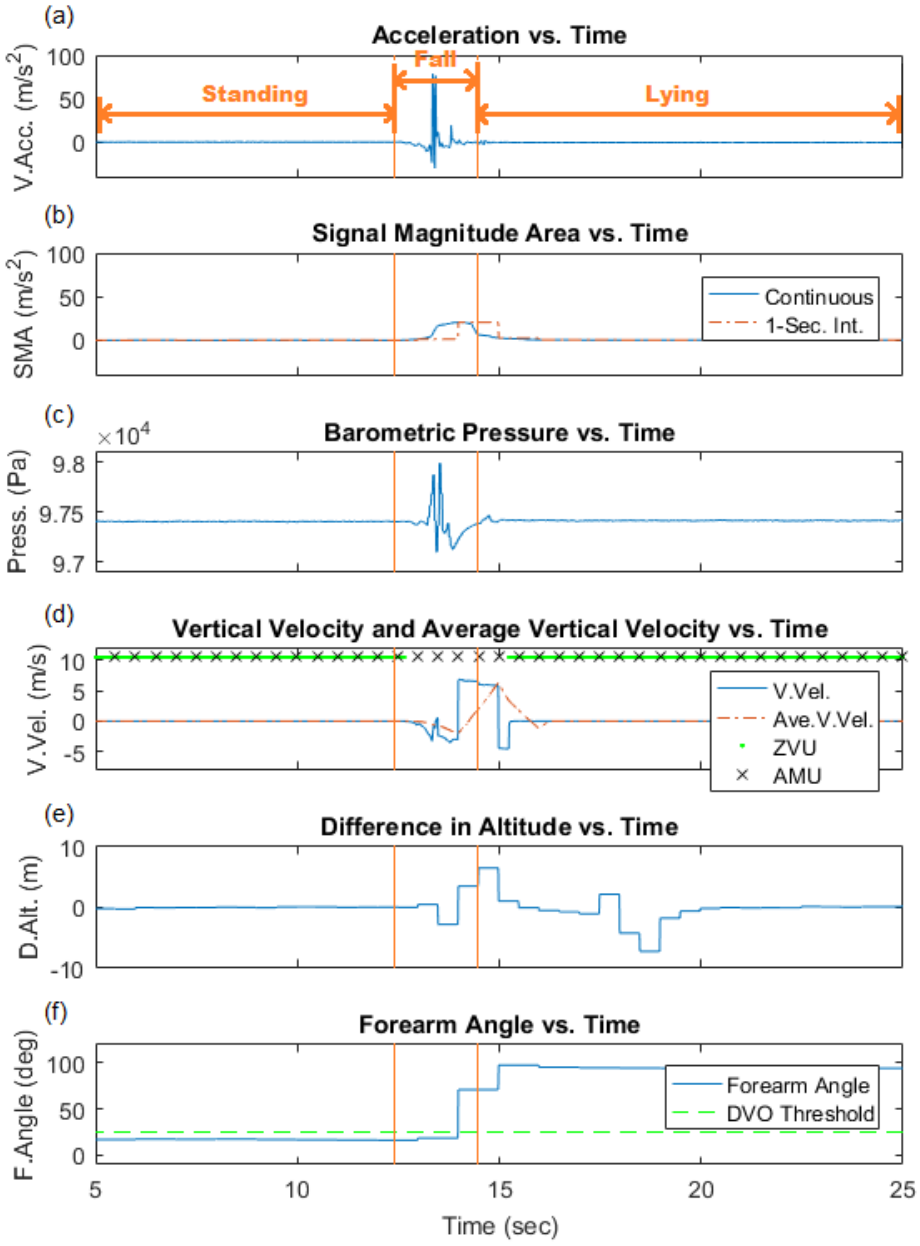

Figure 4.7. Plots of parameters for the same collapse trial for Study 1 but without selective use of pressure (SUP). (a) is vertical acceleration, (b) is SMA, (c) is barometric pressure, (d) is vertical velocity and average vertical velocity, (e) is difference in altitude, and (f) is forearm angle.

The maximum and minimum peaks for vertical velocity and difference in altitude with and without SUP are shown through boxlplots in Figures 4.8 and 4.9, respectively. Vertical velocity's maximum and minimum increased from 5.12 m/s to 71.46 m/s (Figures 4.8a and 4.8b), and decreased from -6.16 m/s to -87.85 m/s (Figures 4.8c and 4.8d), respectively. Difference in altitude's increased from 3.84 m to 74.45 m (Figures

4.9a and 4.9b), and decreased from -3.98 m to -75.91 m (Figures 4.9c and 4.9d), respectively. Among the general activity types, for difference in altitude, it was falls that had the biggest spread in magnitude without SUP despite ADLs having the biggest elevation change due to ascending and descending stairs trials.

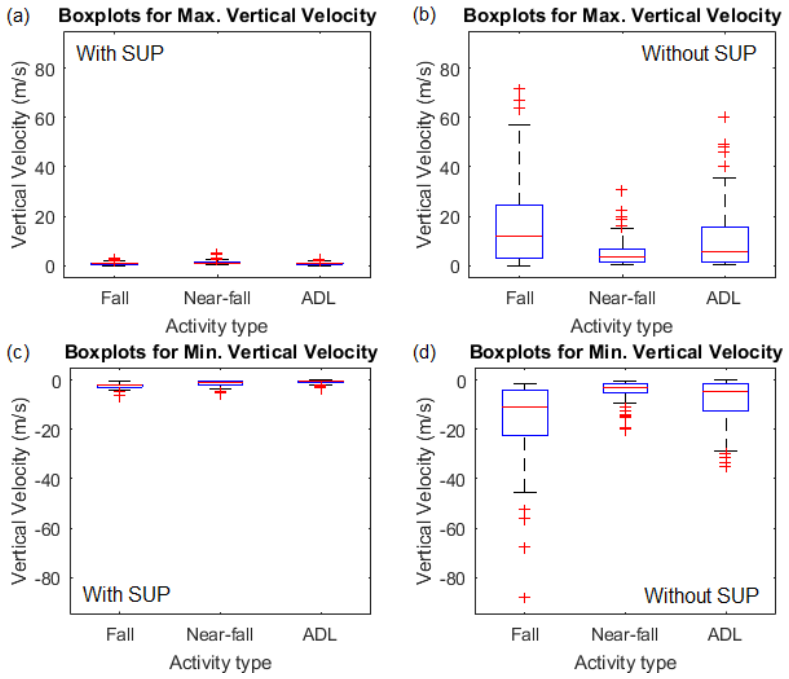

Figure 4.8. Peaks of vertical velocity for all the trials in Study 1 for (a) & (b) maximum vertical velocity with and without selective use of pressure (SUP), respectively, and (c) & (d) minimum vertical velocity with and without SUP, respectively

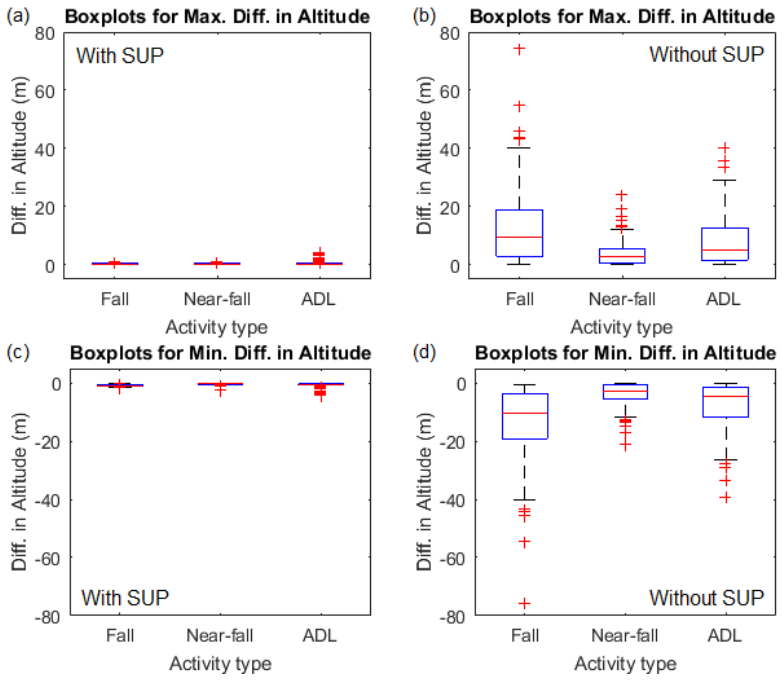

Figure 4.9. Peaks of difference in altitude for all the trials in Study 1 for (a) & (b) maximum difference in altitude with and without selective use of pressure (SUP), respectively, and (c) & (d) minimum difference in altitude with and without SUP, repectively

4.4.2. Algorithm Performance Using Simulated Laboratory-Based Trials

With Selective Use of Pressure

Shown in Tables 4.3 and 4.4 are the experimental results from simulated laboratory based trials. For Algorithms 1 and 2 with SUP, sensitivity was 98.8% and 100%, respectively; ADL specificity was 94.0% and 97.2%, respectively; and non-fall specificity was 88.8% and 97.1% respectively.

Table 4.3. Experimental Results. Sensitivity and specificity of each algorithm during simulated laboratory-based trials (Experiment 1) of Study 1

Algorithm	Sensitivity (%)	Fall vs. ADL Specificity (%)	Fall vs. Non-Fall Specificity (%)
Algorithm 1 w/ SUP	98.8	94.0	88.8
Algorithm 1 w/o SUP	97.6	90.7	78.9
Algorithm 2 w/ SUP	100.0	97.2	97.1
Algorithm 2 w/o SUP	99.4	92.6	89.1

Table 4.4. Experimental results. Misclassified activities of each algorithm during simulated laboratory-based trials of Study 1

	Algorithm 1		Algorithm 2	
	w/ SUP	**w/o SUP**	**w/ SUP**	**w/o SUP**
False Negatives				
Trip	0	0	0	0
Hit/Bump	0	0	0	0
Collapse	0	0	0	0
Cross-step	0	0	0	0
Stand-to-sit	0	0	0	0
Sit-to-stand	1	2	0	1
Roll-bed	1	2	0	0
(Total)	(2)	(4)	(0)	(1)
False Positives (ADLs)				
Normal walk	0	0	0	0
Sit-to-stand	0	0	0	0
Stand-to-sit	0	0	0	0
Pick-object	0	0	0	0
Reach-object	3	0	2	0
Wash-hands	7	14	2	9
Lay-on-bed	2	4	2	5
Ascend-stairs	0	1	0	1
Descend-stairs	1	1	0	1
(Total)	(13)	(20)	(6)	(16)
False Positives (Near-Falls)				
Trip	9	15	1	7
Hit/Bump	1	7	0	4
Cross-step	7	13	0	3
Sit-to-stand	5	8	2	4
(Total)	(22)	(46)	(3)	(18)

For sensitivity, in Algorithm 1, even though the thresholds were already set for 100% sensitivity, two falls were still classified as a non-fall event. A sit-to-stand trial was misclassified due to the absence of reliable pressure within the 25-second monitoring period after impact detection. The other, a roll-bed trial, was misclassified due to an unusual early time-occurrence of dip in difference altitude profile which caused it to be undetected. For Algorithm 2, the two aforementioned activities were properly classified as they had sufficient vertical velocity, vertical acceleration, and average vertical velocity to pass through the second path which did not require difference in altitude. Thus, having this additional path in Algorithm 2 increased the algorithm's reliability by providing an alternative pathway without requiring difference in altitude but guarded against false positives by having higher threshold values for its parameters.

In terms of ADL specificity, Algorithm 1 resulted in 13 misclassified non-fall events. These false positives include 2 lying-on-bed, 7 washing-hands, 3 reaching-object, and 1 descending stairs event. Lying-on-bed trials were detected as falls due to the hand's descent and impact as the hand supports the weight of the body before laying on the bed. For washing hands, most were misclassified due to the abrupt movement of the wrist when shaking hands to dry. This abrupt movement could have been filtered by average vertical velocity, but since it was tuned for maximum sensitivity, its effectiveness was limited. For reaching-object, trials were misclassified due to the hand's descent (after reaching) and the sudden halt in motion at the end of descent (when arms fully extend). Algorithm 2 resulted in only 6 false positive events. These false positives include 2 lying-on-bed, 2 washing hands, and 2 reaching-object trials. Less number of false positives in Algorithm 2 was the result of forearm angle requirement in its first path. Additionally, the downward vertical motion of the aforementioned activities were not strong or dominantly negative enough to satisfy vertical velocity, vertical acceleration, and average vertical velocity thresholds. Therefore they were not able to pass through the second path either.

With the addition of near-falls, there were 22 additional misclassified non-fall events in Algorithm 1, whereas for Algorithm 2, there were only 3 of them. Although near-falls are non-fall events, they could cause sudden movement in the wrist (satisfying all the thresholds in algorithm 1) as the person tries to recover balance and hence could be misclassified as falls. For Algorithm 2, the 19 near-falls that were not misclassified were filtered by the forearm angle requirement in path 1 since the arm returns back to the downward vertical orientation after the event. The 19 near-falls also did not have sufficiently strong vertical velocity, vertical acceleration, or sufficiently dominantly negative vertical velocity profile to pass through path 2 either. On the other hand, all of the three misclassified near-fall trials passed through the second path which would have not been misclassified if the path did not exist. This additional misclassification is the disadvantage of having additional paths in the algorithm. However, this second path results in better sensitivity and significantly better overall specificity compared to Algorithm 1.

For algorithm 2, out of the168 total number of falls, 163 (97.0%) and 134 (79.8%) were able to pass through either path 1 or 2, respectively, and 129 (76.8%) were able to pass through both paths.

For additional comparison purposes, the wrist-based fall detection algorithm by Hsieh et al. [23], which used accelerometers and gyroscopes on both wrists and with simpler parameters, was also tested (using only one of the wrists)[30]. Its sensitivity, ADL specificity and non-fall specificity was 58.3%, 99.5%, and 99.0%, respectively. A possible reason for the algorithm's very low sensitivity is that together with the algorithm's thresholds being tuned for high specificity, the collective discriminative capacity of the parameters is very limited.

Without Selective Use of Pressure

Excluding SUP for AMU decreased the sensitivity of Algorithms 1 and 2 from 98.8% and 100%, to 97.6% and 99.4%, respectively. The ADL specificity of Algorithms 1 and 2 decreased from 94.0% and 97.2%, to 90.7% and 92.6%, respectively. When near-falls were included, the decrease was more evident from 88.8% and 97.1%, to 78.9% and 89.1%, respectively.

The pressure with strong disturbance produced strong magnitudes of distortion in both vertical velocity and difference in altitude such that portions of both parameters were pushed beyond or pulled away from their thresholds[31] (compare Figures 4.6 and 4.7). Whether additional activities will be misclassified depends on the pressure disturbance's profile and timing relative to vertical velocity and difference in altitude's profile during the fall, which is unpredictable. The consequence of having a lot more false positives will make the fall detector ending up being ignored [7], but the worse is that any type of fall could end up being undetected with unpredictability, which can have dangerous consequences.

For both Algorithms 1 and 2, even without the use of SUP, ADLs such as normal-walk, sit-to-stand, and stand-to-sit did not get new false positives (see Table 4.4) as they

---

[30] With some uncertain information about the algorithm from the aforementioned study's paper, best effort was done in implementing it using the paper's algorithm description, sensor sampling rate, and thresholds; and existing methods in the literature.

[31] The pressure data is used to correct the altitude estimates during the altitude measurement update (AMU) in the VPV Kalman filter. When this data is used during the AMU, it is assumed that it is correct. Any errors in pressure data will result to errors in vertical position and velocity estimates whose magnitude and direction depends on the profile of the error in pressure data. For the case of the strong pressure disturbance that was observed in this study, its oscillatory characteristics could strongly force both vertical velocity and difference in altitude to any of positive or negative directions.

were already at least filtered completely by SMA, or for most of them by vertical acceleration (see Figures 4.10e and 4.10b). For Algorithm 2, the forearm angle requirement was also able to filter such activities.

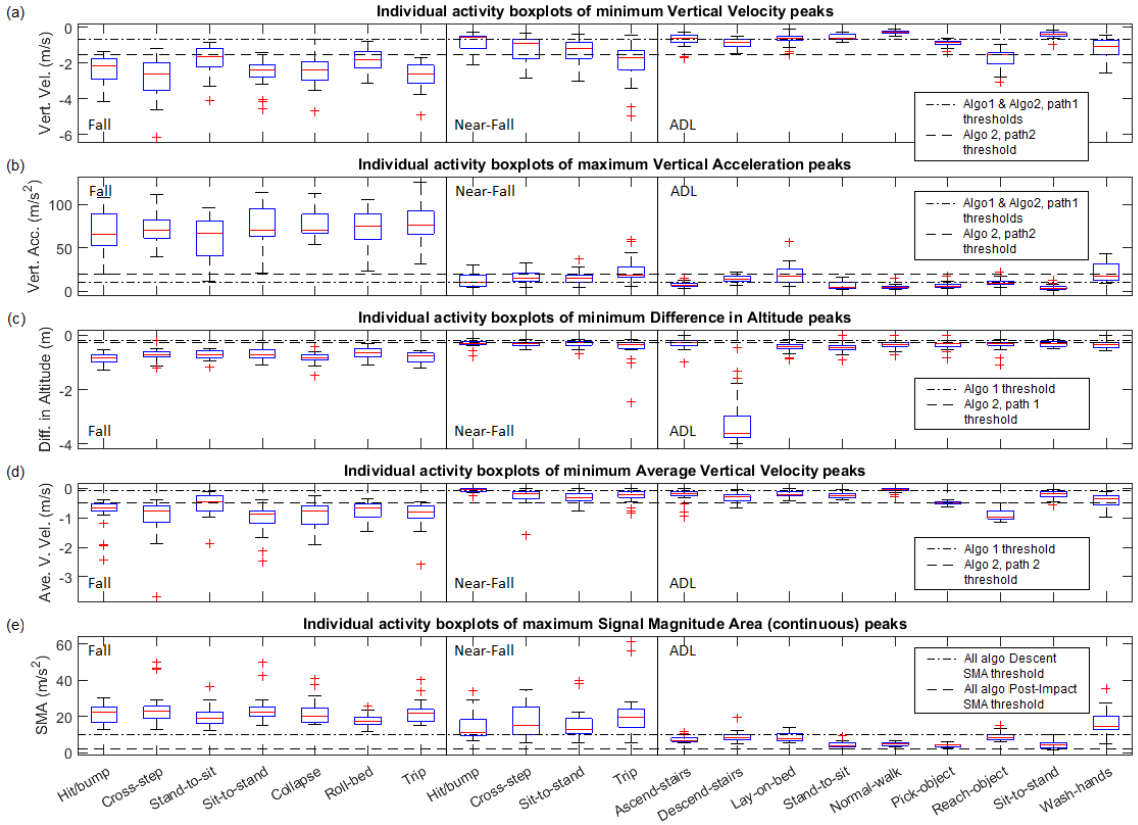

Figure 4.10. Boxplots of peaks of individual activities in Study 1 for (a) minimum vertical velocity, (b) maximum vertical acceleration, (c) minimum difference in altitude, (d) minimum average vertical velocity, and (e) maximum SMA. Continuous SMA is the SMA that is calculated at every time step (used in descent SMA).[32]

Algorithm Performance at Different Accelerometer Ranges and Data Rates

The smartwatch used in this chapter used a custom firmware to be able to use the accelerometer at its ±8 g range instead of the lower ±2 g or ±4 g range for smartwatches. For additional comparison purposes, Algorithm 2 with SUP was also tested with a simulated ±4 g and ± 2 g range by clipping the accelerometer data. Additionally, for potential power consumption reduction, the algorithm was also tested at a simulated lower data rate of 50 Hz for the accelerometer and gyroscope data (including the Kalman filter) for all the aforementioned ranges including ±8 g. The same

[32] Minimum peak means the negative peak with the strongest negative value.

thresholds were used for the different variations of the algorithm. The results are shown in Table 4.5[33].

Table 4.5. Sensitivities and specificities of Algorithm 2 at different accelerometer ranges and sampling rates in Study 1 during simulated laboratory-based trials

Data Rate (Hz)	Accelerometer Range (g)	Sensitivity (%)	Fall vs. ADL Specificity (%)	Fall vs. Non-Fall Specificity (%)
100	±8	100.0	97.6	97.4
100	±4	100.0	97.6	96.6
100	±2	98.2	98.2	98.5
50	±8	99.4	97.6	97.0
50	±4	99.4	97.0	95.8
50	±2	96.4	98.8	98.5

Sensitivities at both ±8 g and ±4 g were the same for the same data rate, and became slightly lower at ±2 g, especially at 50 Hz. For both ADL and non-fall specificities, from ± 8 g to ±4 g, they were either slightly lower or the same, but from either ±8 g or ±4 g range to ±2 g range, the specificities became slightly higher. In terms of data rate, a noticeable difference was a slight decrease in sensitivity at 50 Hz for the same accelerometer range. The misclassified activities for each accelerometer range and data rate for Algorithm 2 are shown in Appendix C.

4.4.3. Parameter Performance Using Simulated Laboratory-Based Trials

Figure 4.11 shows the ROC curves of the fall parameters which shows their overall discriminative capacities when differentiating falls from ADLs (Figure 4.11a), from near-falls (Figure 4.11b), and from all non-falls (Figure 4.11c) for all the trials[34]. Since none of the ROC curves were able to touch the upper left corner of the plot (i.e. at 100% sensitivity and 100% specificity), no individual parameter was able to completely

[33] Please note that since the analyses for pick-object and reach-object trials for Algorithms 1 and 2 were only recently conducted as a response to a manuscript revision for a journal paper, the results in Table 4.7 does not reflect analysis from such trials.

[34] It should be noted that Figure 4.11 and Table 4.8 were based solely on peaks of parameters (from the whole dataset) and did not take into consideration timing relationships between them. In addition, the threshold used for difference in altitude parameter of Algorithm 2 did not include trials that did not pass the forearm angle requirement, which resulted in a slightly higher magnitude threshold compared to Algorithm 1.

separate falls from any of the non-fall events (i.e. either ADLs or near-falls). In Table 4.6, for falls vs. ADLs, at 100% sensitivity, the parameter with highest maximum specificity was continuous SMA (86.1%), followed by vertical acceleration (65.7%), vertical velocity (50.9%), average vertical velocity (17.6%), and difference in altitude (6.9%). When comparing against near-falls, at 100% sensitivity, highest maximum specificity was still continuous SMA (38.5%), followed by vertical acceleration and average vertical velocity (35.4%), vertical velocity (24.0%), and difference in altitude (8.3%). All the parameters except for average vertical velocity and difference in altitude have a substantially smaller specificity when differentiating falls from near-falls. Comparing average vertical velocity to vertical velocity, although both are forms of velocity, the former had better specificity which could be due to its filtering effect on abrupt but short duration wrist movements during near-falls. When falls were compared against both ADLs and near-falls (i.e. non-falls), at 100% sensitivity, the parameter with the highest maximum specificity was continuous SMA (71.5%), followed by vertical acceleration (56.4%), vertical velocity (42.6%), average vertical velocity (23.1%), and difference in altitude (7.4%). Despite the low specificity of the parameters individually, especially when differentiating falls from near-falls, their collective discriminative capacity (with the inclusion of the forearm angle) together with their timing requirements and algorithm structure, allowed Algorithm 2 to have a substantially better ADL specificity and non-fall specificity of 97.2% and 97.1%, respectively.

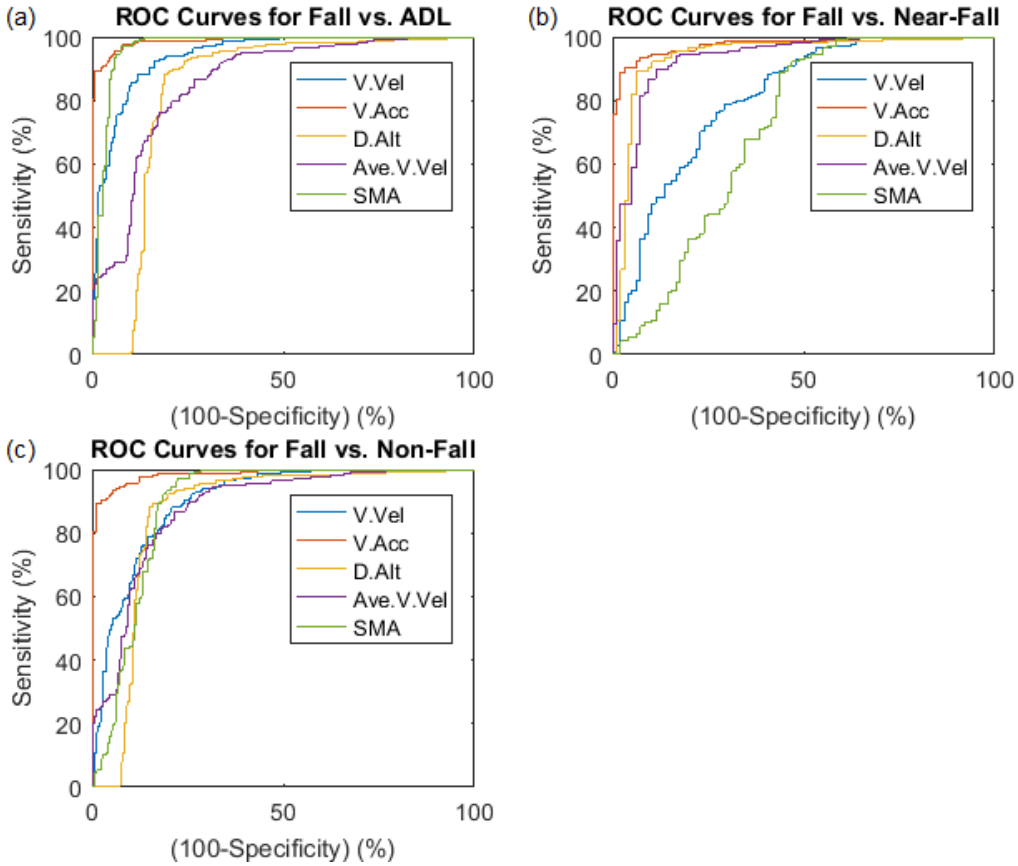

Figure 4.11. ROC curves of all the parameters of Study 1 for (a) fall vs. ADL, (b) fall vs. near-fall, and (c) fall vs. non-fall (i.e. including both ADL and near-fall). The SMA presented is the SMA that is calculated at every time step.

Table 4.6. Experimental results. Maximum specificity of parameters at 100% sensitivity (for all trials) for Study 1

Parameter	Fall vs. ADL (%)	Fall vs. Near-Fall (%)	Fall vs. Non-Fall (%)
Vertical Acceleration	65.7	35.4	56.4
Vertical Velocity	50.9	24.0	42.6
Difference in Altitude	6.9	8.3	7.4
Average Vertical Velocity	17.6	35.4	23.1
Continuous SMA	86.1	38.5	71.5

4.4.4. Algorithm Performance using Continuous Scripted Free-Living ADLs

Using continuous scripted free-living ADL trials, Algorithms 1 and 2 with SUP resulted to 4 and 3 false positives, respectively (see Table 4.7). The four false positives of Algorithm 1 occurred while participants were sitting either on a chair/couch (3 events),

and while lying on a couch. Alarms were triggered during two of the sitting on a chair/couch events as the non-dominant hand was used to support the body weight during the transition, and thus, together with the participant's body's descent, there was sufficient downward motion by the wrist followed by impact and rest. For the remaining sitting on chair/couch event, the algorithm was initially triggered during a prior event of washing hands, which satisfied all the parameters except for difference in altitude. Difference in altitude was only satisfied after walking towards a chair and then sitting down. This satisfaction by a subsequent event was caused by the 25-second monitoring period for difference in altitude, which although allows sufficient time for pressure stabilization after a fall's impact, it also allowed capturing of any sufficient elevation change within the period. This length of monitoring period results to a very loose timing relationship requirement between impact and descent for difference in altitude.

Table 4.7. Experimental results. Misclassified activities of each algorithm during continuous scripted free-living ADL trials in Study 1

Algorithm	False Positives
Algorithm 1 with SUP	4
Algorithm 1 without SUP	8
Algorithm 2 with SUP	3
Algorithm 2 without SUP	7

In the lying on couch event, alarm was triggered not during the transition from standing to lying, but when the participant moved his hand above the head, while already lying, thus satisfying downward motion and impact. But the difference in altitude requirement during such event was satisfied by a small amplitude noise when the hand was already at rest after the event rather than from the actual descent, which was enabled by the very small difference in altitude's threshold of -0.18m.

Algorithm 2 had the same misclassified events as algorithm 1 except without the lying on couch event. The three sitting on chair/couch events were still misclassified despite requiring forearm angle, since all ended with the forearm laid either on the armrest or lap. The lying on couch event didn't caused a false alarm due to the slightly higher threshold of difference in altitude for Algorithm 2.

Without the use of SUP, the number of false positives increased to 8 and 7 for Algorithms 1 and 2, respectively.

4.4.5. Limitations of Study

Having only five fall trials not satisfying the forearm angle requirements of path 1 (although all of them successfully passed through path 2), might not be sufficient evidence to conclude that all falls where the forearm will end in the vertical orientation will be able to satisfy the higher vertical velocity, vertical acceleration, and average vertical velocity thresholds of the second path. However, the observations regarding filtering of the vertical resting position of the forearm are still valid and its benefits could be clearly seen, especially with the reduction of false positives, and for having an alternative path that does not require stability in pressure after the fall.

Due to the higher succeeding steps where a person might land or extent their hands onto during a fall while ascending stairs, it could result to a substantially smaller amount of descent (and consequently, vertical velocity and impact will be smaller) compared to falling on the ground. In such case, both paths of Algorithm 2 might not be able to detect the fall. Other fall scenarios that the algorithm could misclassify include: falling while picking an object from the ground; and falling while the hand is holding onto a fixed structure.

The amount of time to detect a fall after impact will depend on the how quick the post-impact parameters could detect their fall-associated monitored characteristics. For Algorithm 2's second path, due to post-impact SMA, detection time will depend on how early the minimum amount of wrist movement is exhibited and detected, which is given a maximum of 5 seconds (monitoring period). On the other hand, for Algorithm 2's first path, in addition to the maximum of 5 and 5.5 seconds to detect post-impact SMA and forearm angle, respectively, the longest possible detection time will depend on how quickly a difference in altitude is detected after impact (which varied by device in our experiment) which is given a maximum of 25 seconds.

Similar to other studies, fall trials were simulated under controlled laboratory conditions by young and healthy individuals, who fell on soft gymnasium mats. There are inevitable discrepancies between the falling patterns observed in our trials, and those of older adults who are the target for our fall monitoring technology [26, 45, 46]. However, a strength of our study is the selection of falls based on the causes and activities reported to be the most common among older adults in long-term care [41]. Additionally, by not

giving instructions regarding fall direction allows variability in fall kinematics [18]. Furthermore, in order to minimize the effect of surface stiffness on falling behavior, the top 13 cm layer of the gymnasium mats consisted of high-density ethylene vinyl acetate foam. This provided the composite structure with a stiffness high enough to allow for stable standing and walking, but soft enough to reduce impact forces to a safe level.

4.5. Conclusion

This chapter proposed a threshold-based fall detection algorithm for a commercially available smartwatch equipped with tri-axial accelerometer, tri-axial gyroscope, and a barometric pressure sensor. The novelty of the proposed algorithm lies in the use of forearm angle to further improve the discriminative capacity between falls and non-fall events. Additionally, selective use of barometric pressure (SUP) data was proposed in order to remove barometric pressure disturbances during dynamic arm motion. In addition to forearm angle, the proposed algorithm employs vertical acceleration, vertical velocity, difference in altitude, average vertical velocity, descent signal magnitude area (SMA), and post-impact SMA as parameters. To improve the algorithm's performance, two parallel paths were proposed where the second path detects the falls that do not satisfy the forearm angle requirement. In addition, the second path provides a backup path for falls with strong downward motion but whose pressure data does not stabilize after impact.

Experimental results show that without the proposed SUP, the number of false positives significantly increased due to the drastic effects of distortion in both vertical velocity and difference in altitude. Results also show that addition of the forearm angle requirement in the algorithm can significantly reduce the number of false positives especially for near-fall events. Based on the laboratory-based simulated falls, the proposed algorithm has 100% sensitivity, 97.2% ADL specificity, and 97.1% non-fall (i.e. including both ADLs and near-falls) specificity. Further testing using continuous scripted free-living ADL trials, resulted to 3 false positive events.

Chapter 5.

Study 2: Use of Average Vertical Velocity and Difference in Altitude for Improving Automatic Fall Detection from Trunk Based Inertial and Barometric Pressure Measurements

5.1. Abstract

This chapter (Study 2) evaluates the potential use of either difference in altitude or average vertical velocity in further improving the performance of a trunk-based fall detection algorithm that uses vertical velocity, vertical acceleration, and trunk-angle as parameters (i.e. base algorithm). Difference in altitude and average vertical velocity were used to detect the trunk's substantial change in elevation, and its longer timespan of decreasing vertical velocity, respectively, during the descent phase of a fall. Vertical velocity, difference in altitude and average vertical velocity were estimated by fusing tri-axial accelerometer, tri-axial gyroscope, and barometric pressure sensor data using a Kalman filter.

The algorithm was tested using a comprehensive set of simulated laboratory-based trials described in Chapter 3, consisting of 166 falls, 150 activities of daily living (ADL), and 95 near-falls recorded from 12 participants. Results show that the addition of at least difference in altitude or average vertical velocity was able to increase the algorithm's non-fall (i.e. including both ADLs and near-falls) specificity from 91.8% to 98.0% and 99.6%, respectively.

5.2. Introduction

Although not as convenient as the wrist location, the trunk is a more ideal location for fall detection since it contains a major part of the body's total mass, where its motion represents "whole body" movements [25, 8]. Additionally, it does not produce big and quick motions as easily like the wrist during activities of daily living.

Despite the high or even perfect sensitivities and specificities of trunk-based fall detection algorithms (e.g. in [22, 14, 17]), a study by Bagala *et al.* [26] showed that their

performance were limited when tested using accelerometer data from frail older adults. Additionally, the sole use of accelerometer for estimating vertical velocity has inherent limitations for accuracy [19, 16]. Among the evaluated algorithms, Bourke *et al.*'s [17] vertical velocity + impact + posture algorithm gave the best trade-off between sensitivity and specificity.

More recent studies have shown that the accuracy of fall detection systems can be improved by combining accelerometers and barometric pressure sensors. For example, equipped with a waist-mounted barometric pressure sensor, Bianchi *et al.* used difference in barometric pressure as an alternative parameter for detecting changes in altitude and was able to improve an accelerometer-based algorithm's sensitivity from 75.0% to 97.5% at 96.5% specificity for both [15]. Fusing barometer data to accelerometer and gyroscope data allows for an accurate and drift-free estimate of vertical velocity and altitude [32].

In addition to the change in altitude, the duration and magnitude of the trunk's downward vertical velocity during the descent phase of a fall is another important characteristic. For example, during recovery attempts in near-fall events, although the associated abrupt motions [18] could produce sufficiently strong negative vertical velocities, their duration remains short. On the other hand, the span of the increased negative vertical velocity during falls (i.e. from start of descent until impact) is generally longer than non-falls. Thus, by taking the average of the vertical velocity profiles of both events could potentially result to a more attenuated signal for non-falls compared to falls, and hence could help distinguish between the two events.

Therefore, the objective of chapter is to evaluate the addition of average vertical velocity and difference in altitude to the vertical velocity, vertical acceleration and trunk angle parameters for developing an improved trunk-based fall detection algorithm using triaxial accelerometer and gyroscope, and a barometric pressure sensor.

The novelty in this study for a waist-based fall detection algorithm lies in the use of average vertical velocity to differentiate between falls and non-fall events.

5.3. Methodology

5.3.1. Parameters

Vertical Acceleration

During the impact phase of a fall, a sudden spike in vertical acceleration is generated as the trunk from its maximum downward velocity (at the end of descent) is suddenly put to a halt. *Vertical acceleration (* ${}^{I}a_z[k]$*)* was used to detect this acceleration spike during impact, whose occurrence also serves as a time reference for subsequently monitored parameters. Similar to Study 1, it is the 3rd component of the gravity-compensated acceleration (${}^{I}\boldsymbol{a}[k]$) in the inertial reference frame (I) at time step k, where ${}^{I}\boldsymbol{a}[k]$ was calculated using (4.1)[35].

Vertical Velocity and Difference in Altitude

During the descent phase of a fall, the downward velocity of the trunk is expected to have a higher magnitude compared to ADLs [33]. *Vertical velocity (* ${}^{I}v_z[k]$*)* was used to detect this quick downward movement of the trunk, and similar to Study 1, it was estimated using the vertical position and velocity Kalman filter (VPV Kalman filter) by Zihajehzadeh *et al.* [32] which allows a drift-free estimate of vertical velocity and altitude. The inputs to the VPV Kalman filter are the tilt Kalman filter's outputs and the barometric pressure data[36] (see Figure 4.1).

In the same phase of a fall, the trunk is also expected to undergo a substantial change in elevation from a higher position to a lower position, before descent phase and right after impact phase, respectively. *Difference in altitude (*$\Delta h[k]$*)* was used to capture this elevation difference of the trunk and was estimated as:

$$\Delta h[k] = \left({}^{I}p_{z,LPF}[k] - {}^{I}p_{z,LPF}[k-w+1] \right) \tag{5.3}$$

[35] Although vertical acceleration parameter in Studies 1 and 2 used the same equation (i.e. 4.1), as mentioned in Chapter 3, the two studies used different sensor data.

[36] A general explanation about the mechanism behind the tilt and VPV Kalman filters used in this study could be found in Appendix B.

where ${}^{I}\boldsymbol{p}_{z,LPF}[k]$ is the low-pass filtered (using 2nd order Butterworth filter at cut-off frequency = 50Hz) vertical position estimate (${}^{I}p_{z}[k]$) using the VPV Kalman filter [32], and w is the number of time steps within a 4-second window (i.e. $4\ sec \times 100\ Hz\ sampling\ frequency$).

It was noticed that when the sensor became sealed between the participant's lower back and mattress, such as during impact in backward falls, there was a strong disturbance in pressure data. This disturbance makes the pressure data unreliable for the altitude measurement update (AMU) step of the VPV Kalman filter. As such, the rolling average pressure [32] was only used in AMU whenever the difference between the maximum and minimum pressure within a 0.5-second window was lower than or equal to the pressure stability threshold of 25 Pascals[37].

In between AMU steps of the VPV Kalman filter and in the absence of reliable pressure, zero velocity update (ZVU) technique [44] was used to reduce the drift in vertical velocity/altitude estimation. ZVU was activated when the variance of a 0.5-second window of external acceleration norm was less than 0.05 $(m/s^2)^2$. With the pressure disturbance occurring only during instances when the pressure gets sealed (and hence, it occurs a lot less frequent compared to the smartwatch), high-pass filtering of ${}^{I}\boldsymbol{a}[k]$ was not used. Figure 5.1 shows the block diagram of the algorithm used for vertical velocity/altitude calculation.

[37] Compared to Study 1, the threshold only needs to be satisfied in a single window instead of two windows.

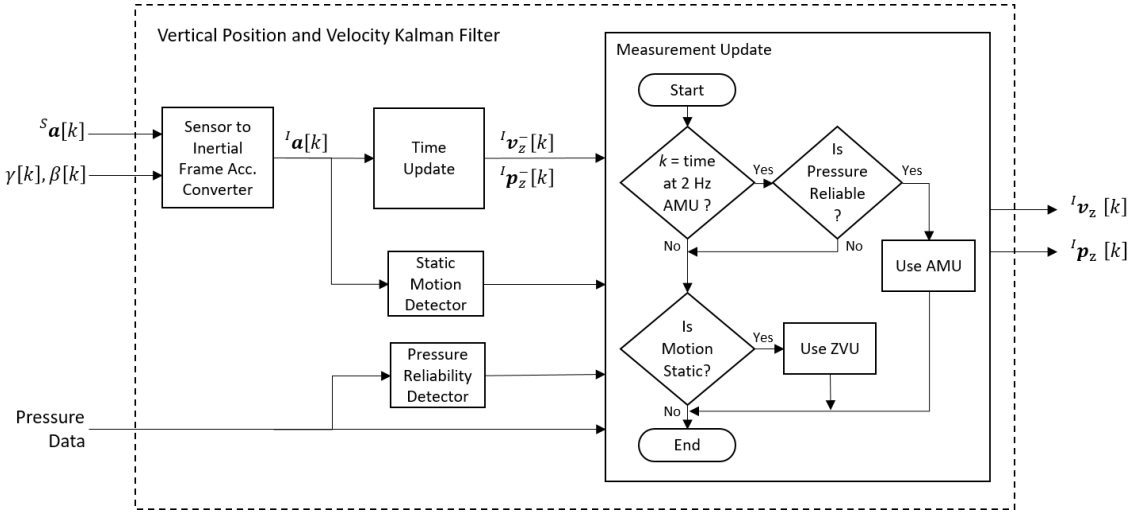

Figure 5.1. Block diagram of vertical position and velocity (VPV) Kalman filter in Study 2 which emphasizes the drift reduction logic

Average Vertical Velocity

During attempts to recover balance in near-falls (and possibly in ADLs that involve a quick downward motion), the associated abrupt movements can produce a high negative vertical velocity. The duration of such vertical velocity is relatively shorter compared to the duration of longer (i.e. from descent to impact) and generally stronger downward vertical velocity during falls. By taking the average of vertical velocity from both events, it is expected that the resulting signal profile from such near-fall event will be more attenuated compared to the resulting signal profile from fall events. *Average Vertical Velocity* was used to capture such longer and generally stronger negative vertical velocity profile during falls. It was calculated by taking the mean of the vertical velocity within a 1-second window.

Trunk Angle

After the impact of a fall, the body is assumed to be in a lying posture. *Trunk Angle* $TA[k]$ was used to detect this posture by measuring the angle of the trunk with respect to the direction of gravity. It was estimated as:

$$TA[k] = \frac{1}{F_S} \sum_{l=k-F_S+1}^{k} cos^{-1}\left({}^{S}x_{1\_x}[l]\right) \frac{180}{\pi} \tag{5.2}$$

where ${}^{S}\boldsymbol{x}_1[l]$ is the normalized gravity vector in the sensor frame [37, 32] (x-axis is the sensor's axis that is most parallel to the body's longitudinal axis and is pointed towards the head) and ${}^{S}x_{1\_x}[l]$ is its x-component. Trunk angle was measured every 1-second interval, and should be above the 20-degree threshold within the 3-second monitoring period. An initialized trunk angle was not used since an accidental improper initialization during actual use could cause incorrect estimates which may result in undetected falls.

5.3.2. Algorithm

Four algorithms were evaluated in this chapter, where each is a variation of the combination of addition of difference in altitude and average vertical velocity to the base algorithm (vertical velocity + vertical acceleration + trunk angle) and are shown in Table 5.1.

Table 5.1. Algorithms to be compared in Study 2

	Base Parameters			Additional Parameters	
Algorithm	**Vertical Velocity**	**Vertical Acceleration**	**Trunk Angle**	**Difference in Altitude**	**Average Vertical Velocity**
Algorithm 1	Yes	Yes	Yes	No	No
Algorithm 2	Yes	Yes	Yes	Yes	No
Algorithm 3	Yes	Yes	Yes	No	Yes
Algorithm 4	Yes	Yes	Yes	Yes	Yes

Algorithm 4 (see Figure 52.) started with the monitoring of vertical velocity to detect descent, and once it reached its threshold, the vertical acceleration was monitored for 0.8 seconds to detect impact. The detection of impact was the starting point for simultaneous monitoring of difference in altitude, average vertical velocity, and trunk angle. When all the aforementioned parameters satisfied their pre-set threshold values within their monitoring period, the event was considered as a fall.

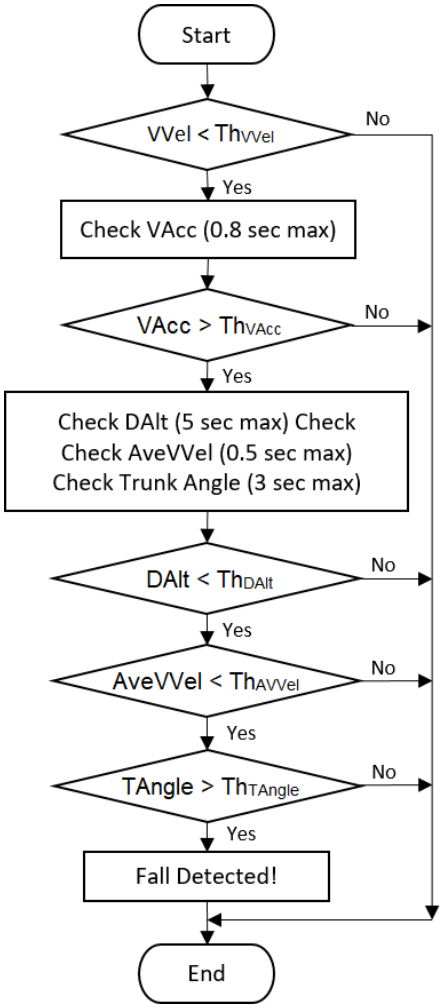

Figure 5.2. Flowchart of proposed Algorithm 4 in Study 2

Algorithms 1 to 3 were implemented the same way as Algorithm 4 except for the corresponding exclusion/s of difference in altitude or/and average vertical velocity. All the parameters' threshold were tuned for 100% sensitivity with the maximum possible specificity based on the minimum peak-magnitudes of parameters and the relative timing relationships between parameters from all the fall trials. The monitoring periods serve as fall signatures in addition to the parameter thresholds. The thresholds and monitoring periods are shown in Tables 5.2 and 5.3, respectively, and were the same for all the algorithms.

Table 5.2. Threshold of each parameter of Algorithms 1 to 4 of Study 2

Parameter	Threshold
Vertical Velocity (m/s)	-1.2
Vertical Acceleration (m/s^2)	11.6
Difference in Altitude (m)	-0.31
Average Vertical Velocity (m/s)	-0.4
Trunk Angle (degrees)	20

Table 5.3. Monitoring period of each parameter for Algorithms 1 to 4 of Study 2

Parameter	Monitoring Period
Vertical Acceleration	0.8 sec. after descent
Difference in Altitude	Max. 10 reliable pressure data points or 5 sec. after impact (whichever comes first)
Average Vertical Velocity	0.5 sec. after impact
Trunk Angle	3 sec. after impact

An iteration of the algorithm was initiated at every time step (i.e. at 100 Hz) to prevent the algorithm from missing the time-instance of any fall event. To save memory resources, only the important contexts, such as the parameter's timing references and logic conditions were stored for each algorithm iteration.

With this iteration frequency and the signal characteristics of parameters during a fall, consecutive fall detections might occur for a single fall event. To prevent such multiple detections, a time difference of 1.5 s was required before a subsequent detection was considered as a new fall event.

5.4. Results

5.4.1. Algorithm Performance

Tables 5.4 and 5.5 show the algorithms' performance. The sensitivity, ADL specificity, and non-fall specificity of the base algorithm was 100%, 97.3%, and 91.8%, respectively. With the addition of difference in altitude it was 100%, 97.3%, and 98.0%, respectively, while the addition of average vertical velocity further improved the performance to 100%, 99.3%, and 99.6%, respectively. The addition of both average vertical velocity and difference in altitude gave 100%, 99.3%, and 99.6%, respectively, which was the same performance as adding only average vertical velocity.

Table 5.4. Experimental Results. Sensitivity and specificity of each algorithm during simulated laboratory-based trials (Experiment 1) of Study 2

Algorithm	Sensitivity (%)	Fall vs. ADL Specificity (%)	Fall vs. Non-Fall Specificity (%)
Algorithm 1	100.0%	97.3%	91.8%
Algorithm 2	100.0%	97.3%	98.0%
Algorithm 3	100.0%	99.3%	99.6%
Algorithm 4	100.0%	99.3%	99.6%

Table 5.5. Experimental results. Misclassified activities of each algorithm during simulated laboratory-based trials for Study 2

	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
False Negatives				
Trip	0	0	0	0
Hit/Bump	0	0	0	0
Collapse	0	0	0	0
Cross-step	0	0	0	0
Stand-to-sit	0	0	0	0
Sit-to-stand	0	0	0	0
Roll-bed	0	0	0	0
(Total)	(0)	(0)	(0)	(0)
False Positives (ADLs)				
Normal walk	0	0	0	0
Sit-to-stand	0	0	0	0
Stand-to-sit	0	0	0	0
Pick object	0	0	0	0
Lay-on-bed	4	4	1	1
Ascend-stairs	0	0	0	0
Descend-stairs	0	0	0	0
(Total)	(4)	(4)	(1)	(1)
False Positives (Near-Falls)				
Trip	7	1	0	0
Hit/Bump	0	0	0	0
Cross-step	5	0	0	0
Sit-to-stand	4	0	0	0
(Total)	(16)	(1)	(0)	(0)

With just the base algorithm, there were 4 ADL false positives and 16 near-fall false positives. All the 4 ADL trials were from lay-on-bed trials and were misclassified due to their kinematic similarities to a fall, which include the downward movement, the impact, and the laying of back to the mattress. The 16 misclassified near-fall trials were from all of the near-fall types. The abrupt movements during the attempt to recover

balance during a near-fall caused velocity and vertical acceleration to satisfy their thresholds (see Figures 5.4a and 5.4b)[38]. The use of trunk angle could have contributed more in filtering such activities, but its low threshold (a trade-off for not having initialization) had limited its discriminative capacity.

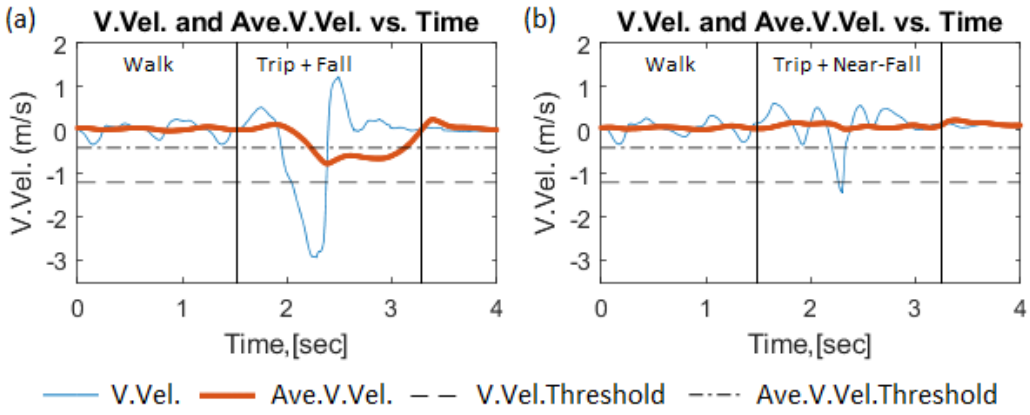

Figure 5.3. Plots of vertical velocity (V.Vel.) and average vertical velocity (Ave.V.Vel.) from (a) fall and (b) near-fall trip trial

The addition of difference in altitude was able to reduce 15 of the remaining near-fall trials since near-falls mostly involve only minimal amount of descent. On the other hand, the addition of average vertical velocity was able to filter all of the near-fall trials since it was able to relatively attenuate the shorter duration of near-fall's abrupt movement compared to falls (see Figure 5.3a and 5.3b). In Figure 5.4b, it can be seen that average vertical velocity was completely capable of differentiating between fall and near-fall peaks with a clear gap. For ADLs, only the addition of average vertical velocity was able to reduce the ADL false positive trials – it was able to filter 3 out of the 4 lay-on-bed trials. Adding both difference in altitude and average vertical velocity gave the same result as adding only average vertical velocity when differentiating falls either from ADLs or near-falls.

[38] This amount of false positives show the base algorithm's limitation that which would not be known without the inclusion of near-fall trials.

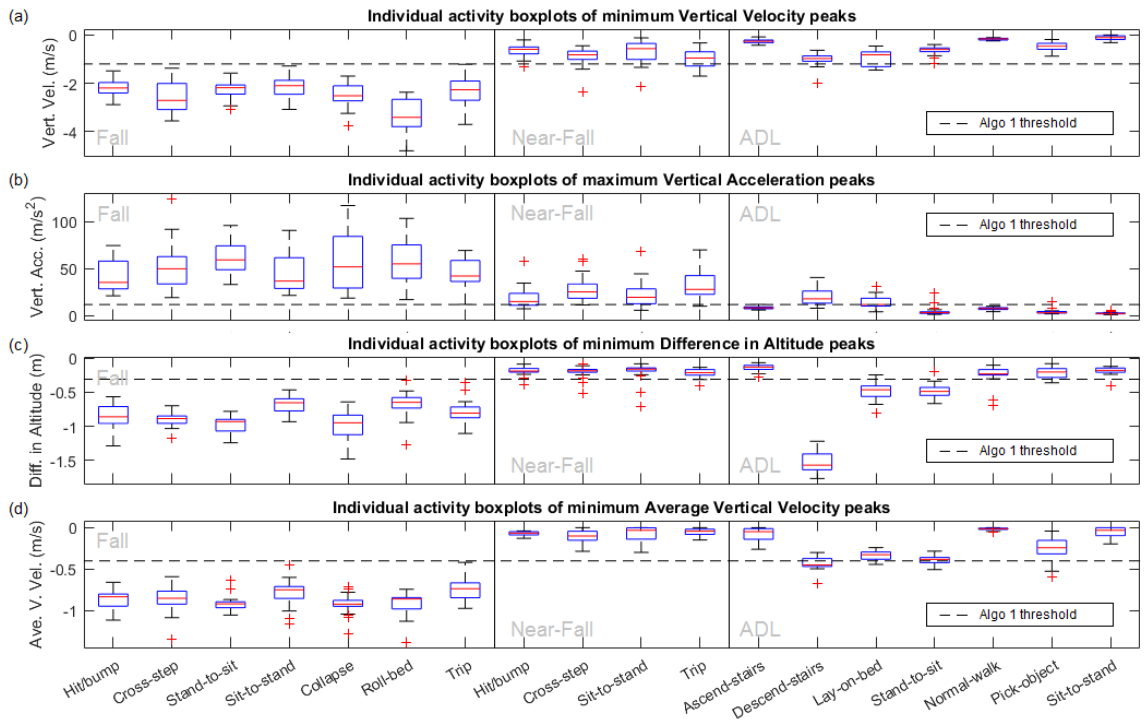

Figure 5.4. Boxplots of peaks of individual activities in Study 2 for (a) minimum vertical velocity, (b) maximum vertical acceleration, (c) minimum difference in altitude, and (d) minimum average vertical velocity[39]

5.4.2. Parameter Performance

Figure 5.6 shows the ROC curves of all the parameters for fall vs. ADL (Figure 5.5a), fall vs. near-fall (Figure 5.5b), and fall vs. non-fall (Figure 5.5c). No parameter was able to completely differentiate falls from any of the non-fall (i.e. either ADLs or near-falls) groups except for average vertical velocity for near-falls. Table 5.6 shows the corresponding maximum specificity of each parameter at 100% sensitivity. For fall vs. ADL, vertical velocity had the highest specificity (94.0%), followed by average vertical velocity (87.3%), vertical acceleration (82.0%), and difference in altitude (57.3%). Compared to other parameters, difference in altitude had a low specificity and was mostly due to ADLs that involve substantial amount of descent such as lay-on-bed, stand-to-sit, and most especially descending-stairs (see Figure 5.4c).

[39] Minimum peak means the negative peak with the strongest negative value.

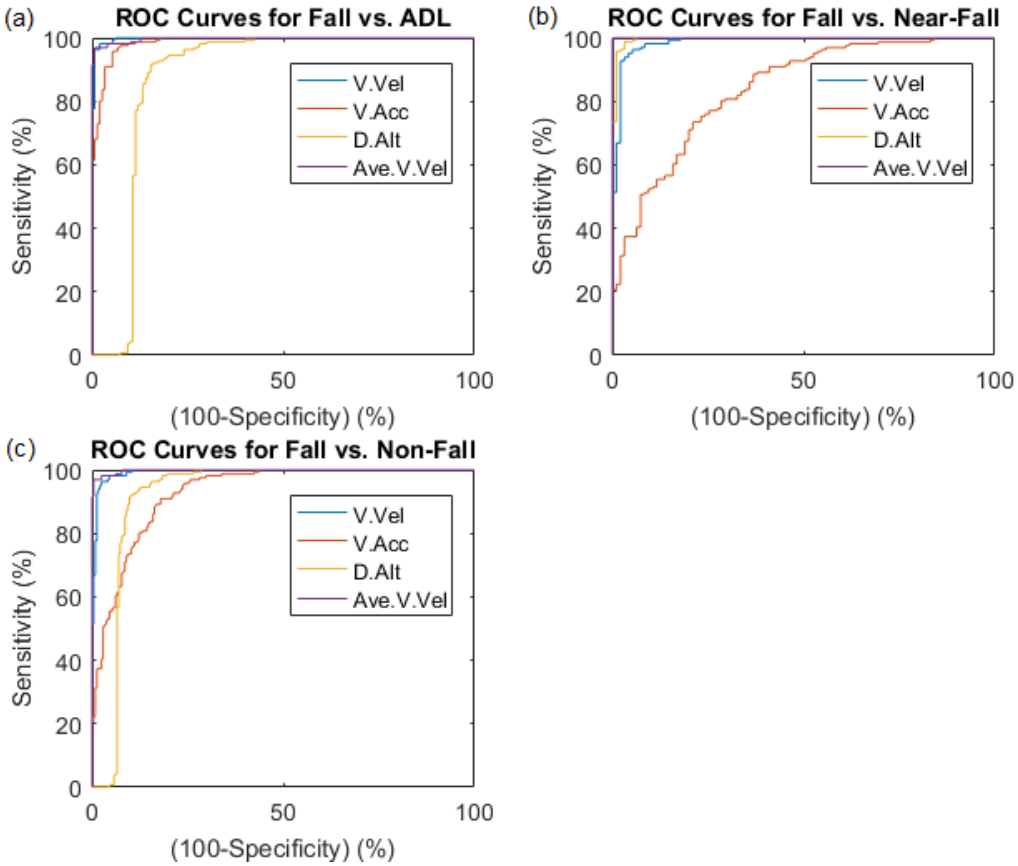

Figure 5.5. ROC curves of all the parameters of Study 2 for (a) fall vs. ADL, (b) fall vs. near-fall, and (c) fall vs. non-fall

Table 5.6. Experimental results. Maximum specificity of parameters at 100% sensitivity (for all trials) for Study 2

Parameter	Fall vs. ADL (%)	Fall vs. Near-Fall (%)	Fall vs. Non-Fall (%)
Vertical Velocity	94.0	82.1	89.4
Vertical Acceleration	82.0	15.8	56.3
Difference in Altitude	57.3	93.7	71.4
Average Vertical Velocity	87.3	100.0	92.2

When differentiating between falls and near-falls, at 100% sensitivity, average vertical velocity had the highest specificity (100.0%), followed by difference in altitude (93.7%), vertical velocity (82.1%), and vertical acceleration (15.8%). As mentioned previously, average vertical velocity was able to completely differentiate between fall and near-fall events with a clear gap between their peaks (see Figure 5.4d). Also, for difference in altitude, although it had a poor performance in ADLs, it was much better in

near-falls since they only mostly involve minimal descent of the trunk. Opposite to vertical velocity when differentiating against ADLs, average vertical velocity's performance was higher when differentiating against near-falls. Although both parameters are forms of vertical velocity, the latter had a substantially better performance against near-fall events since averaging was able to relatively attenuate the shorter duration of abrupt movements during such events compared to falls. Compared to other parameters, vertical acceleration had a very low specificity when comparing against near falls which could be due to the associated abrupt movements or impacts during attempts to recover balance. When differentiating from both ADLs and near-falls (non-falls), at 100% sensitivity, average vertical velocity had the highest specificity (92.2%), followed by vertical velocity (89.4%), difference in altitude (71.4%), and vertical acceleration (56.3%).

5.4.3. Limitations of Study

Similar to other studies, trials were simulated using young and healthy volunteers in a laboratory setting and it is expected that the captured motion could be different from that of an older adult in a real-world scenario [46, 45]. Hence, further study using sufficient real-world data from the older adult population is needed in order to accurately evaluate the proposed algorithm.

On the other hand, compared to existing studies, the simulated falls in this study were based on commonly occurring falls in long-term-care (LTC) facilities based on video evidence [41]. Also, similar to Lee et al.'s study, the participants were not given instruction regarding the direction of fall, and hence will allow variability in fall kinematics [18]. Additionally, near-falls were also included in this study, making our simulated trials more comprehensive than the existing literature.

5.5. Conclusion

This chapter evaluated the effect of the addition of either difference in altitude or average vertical velocity to the performance of a vertical velocity + vertical acceleration + trunk angle fall detection algorithm (base algorithm) using tri-axial accelerometers, tri-axial gyroscope, and a barometric pressure sensor mounted on the trunk. The algorithms were tested using a comprehensive set of 166 falls, 95 near-falls, and 150

ADLs from 12 participants. Experimental results show that adding either difference in altitude or average vertical velocity was able to increase the algorithm's non-fall specificity from 91.8% to 98.0% and 99.6%, respectively.

Chapter 6.

Conclusion

6.1. Summary and Conclusion

This thesis presented the work done in improving the performance of threshold-based fall detection algorithms for the wrist and trunk using tri-axial accelerometer, tri-axial gyroscope and a barometric pressure sensor.

Chapter 1 introduced the work that was presented in this thesis by providing the problem that it was aiming to solve and the objectives. It explained the need to detect falls, where an inertial sensor-based wrist-mounted and trunk-mounted fall detectors are two desirable solutions. However their existing performance is limited. Therefore, the objective was to improve the performance the two fall detection algorithms.

Chapter 2 provided the review of the literature which served as the foundation and motivation for the methods that were used in the first and second studies (i.e. Chapters 4 and 5, respectively). It focused on the previous methods employed in trunk-based and wrist-based fall detection, and potential solutions in improving the algorithm's performance.

Chapter 3 presented the methods used for data gathering, wherein the recorded data were used for evaluating the algorithms presented in Chapters 4 and 5. The experiment protocol consisted of a comprehensive set of simulated laboratory-based trials of falls ADLs, and near-falls (Experiment 1), and a continuous scripted free-living ADL trial in a prepared environment (Experiment 2), all recorded from 12 participants. The chapter also discussed the sensors and other equipment used in the experiments.

Chapter 4 presented the first study which proposed a wrist-based fall detection algorithm for a commercially available smartwatch equipped with a tri-axial accelerometer, tri-axial gyroscope, and a barometric pressure. The algorithm used vertical velocity, vertical acceleration, difference in altitude, average vertical velocity, descent signal magnitude area (SMA), post-impact SMA, and forearm angle as parameters. Forearm angle was used to filter the downward vertical orientation of the

forearm that could be associated to a non-fall event's post-activity positon. Additionally, to deal with strong disturbances in pressure data from the smartwatch, especially during abrupt movements of the arm, pressure data were used selectively for altitude estimation in a cascaded Kalman filter. Two parallel paths were employed where the second path detects the falls that do not satisfy the forearm angle requirement.

The algorithm was tested using simulated laboratory-based trials consisting of 168 falls, 216 ADLs and 96 near-falls recorded from 12 participants and gave 100% sensitivity, 97.2% ADL specificity, and 97.1% non-fall (i.e. including falls and non-falls) specificity. Further testing using a scripted continuous sequence of free-living ADLs resulted to 3 misclassified ADLs.

Chapter 5 presented the second study which evaluated the effect of adding average vertical velocity and difference in altitude parameters to a trunk-based fall detection algorithm (base algorithm) that uses vertical velocity, vertical acceleration, and trunk angle. Difference in altitude and average vertical velocity were used to detect the trunk's substantial change in elevation, and its longer timespan of decreasing vertical velocity, respectively, during the descent phase of a fall. Both parameters and average vertical velocity were estimated by fusing barometric pressure data to accelerometer and gyroscope data using a cascaded Kalman filter. The algorithms were tested using data from simulated laboratory trials of 166 falls, 150 ADLs, and 95 near-falls recorded from 12 participants. Results show that at 100% sensitivity, the addition of difference in altitude and average vertical velocity was able to increase the base algorithm's non-fall specificity (i.e. including both ADLs and near-falls) from 91.8% to 98.0% and 99.6%, respectively.

6.2. Thesis Contributions

6.2.1. For a Wrist-Based Fall Detector

- A novel accurate threshold-based fall detection algorithm for a commercially available smartwatch equipped with MEMS tri-axial accelerometer, tri-axial gyroscope, and a barometric pressure sensor. The algorithm employed two parallel paths and used vertical velocity, vertical acceleration, difference in altitude, average vertical velocity,

descent signal magnitude area (SMA), post-impact SMA, and forearm angle as parameters.

- Use of forearm angle to help differentiate between fall and non-fall events based on quasi-complementary post-activity orientation of the forearm between the two events.

- Use of cascaded Kalman filters to estimate vertical velocity, difference in altitude, and average vertical velocity parameters.

- Selective use of pressure data in the vertical position and velocity Kalman filter to prevent the negative effects of strong disturbance in barometric pressure data to the vertical velocity and difference in altitude estimates.

6.2.2. For a Trunk-Based Fall Detector

- A more accurate threshold-based fall detection algorithm for the trunk through the addition of average vertical velocity to a base algorithm that employs vertical velocity, vertical acceleration, and trunk-angle as parameters. The algorithm used MEMS tri-axial accelerometer, tri-axial gyroscope and a barometric pressure sensor.

- Use of cascaded Kalman filters to estimate vertical velocity, difference in altitude, and average vertical velocity parameters.

- Use of average vertical velocity to help differentiate between a fall and non-fall event based on the width and magnitude of negative vertical velocity profile between the two events. Additionally, average vertical velocity could completely differentiate between fall and non-fall events.

6.3. Future Direction

The author suggests the following for future direction:

1. Collection of real-world data from older adults to validate the proposed algorithms.

2. Use of other context (e.g. location information using environment sensors) to further help in differentiating between fall and non-fall events when in a controlled environment.

3. Development of new parameters to further improve accuracy, reliability, and efficiency of fall detection algorithms.

4. Development of methods to prevent vertical velocity drift without the use of barometric pressure sensor in order to save power resources.

5. Incorporation of the parameters used in this study to a machine learning-based fall detection algorithm and to compare its performance to the algorithms that were presented in this study.

References

[1] N. D. A. Boye, E. M. M. Van Lieshout, E. F. Van Beeck, K. A. Hartholt, T. J. M. Van der Cammen and P. Patka, "The impact of falls in the elderly," *Trauma,* vol. 15, no. 1, pp. 29-35, 2012.

[2] Public Health Agency of Canada, "Senior's Falls in Canada: Second Report," Public Health Agency of Canada, Ottawa, 2014.

[3] C. Pearson, J. St-Arnaud and L. Geran, "Understanding seniors' risk of falling and their perception of risk," *Health at a Glance,* pp. 1-11, October 2014.

[4] D. Wild, U. S. L. Nayak and B. Isaacs, "How dangerous are falls in old people at home?," *British Medical Journal,* vol. 282, pp. 266-268, 1981.

[5] A. C. Reece and J. M. Simpson, "Preparing Older People to Cope after a Fall," *Physiotherapy,* vol. 82, no. 4, pp. 227-235, 1996.

[6] M. E. Tinetti, W.-L. Liu and E. B. Claus, "Predictors and Prognosis of Inability to Get Up After Falls Among Elderly Persons," *Journal of the American Medical Association,* vol. 269, no. 1, pp. 65-70, 1993.

[7] K. Doughty, R. Lewis and A. McIntosh, "The design of a practical and reliable fall detector for community and institutional telecare," *Journal of Telemedicine and Telecare,* vol. 6, no. 1, pp. 150-154, 2000.

[8] A. Godfrey, R. Conway, D. Meagher and G. OLaighin, "Direct measurement of human movement by accelerometry," *Medical Engineering & Physics,* vol. 30, pp. 1364-1386, 2008.

[9] U. Jensen, P. Blank, P. Kugler and B. M. Eskofier, "Unobtrusive and Energy-Efficient Swimming Exercise Tracking Using On-Node Processing," *IEEE Sensors Journal,* vol. 16, no. 10, pp. 3972-3980, 2016.

[10] C. Strohrmann, R. Labruyère, C. N. Gerber, H. J. Van Hedel, B. Arnrich and G. Tröster, "Monitoring motor capacity changes of children during rehabilitation using body-worn sensors," *Journal of Neuroengineering and Rehabilitation,* vol. 10, pp. 83-98, 2013.

[11] M. Tierney, A. Fraser and N. Kennedy, "Users' Experience of Physical Activity Monitoring Technology in Rheumatoid Arthritis," *Musculoskeletal Care,* vol. 11, pp. 83-92, 2013.

[12] V. Huppert, J. Paulus, U. Paulsen, Burkart, B. Wullich and B. M. Eskofier, "Quantification of Nighttime Micturition With an Ambulatory Sensor-Based System," *IEEE Journal of Biomedical and Health Informatics,* vol. 20, no. 3, pp. 865-872, 2016.

[13] N. Noury, A. Fleury, P. Rumeau, B. A. K, G. OLaighin, V. Rialle and J. E. Lundy, "Fall detection - Principles and Methods," in *Proceedings of the 29th Annual International Conference of the IEEE EMBS*, Lyon, France, 2007.

[14] M. Kangas, A. Konttila, P. Lindgren, I. Winblad and T. Jamsa, "Comparision of low-complexity fall detection algorithms for body attached accelerometers," *Gait & Posture,* vol. 28, pp. 285-291, 2008.

[15] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti and N. H. Lovell, "Barometric Pressure and Triaxial Accelerometry-Based Falls Event Detection," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 18, no. 6, pp. 619-627, 2010.

[16] T. Degen, H. Jaeckel, M. Rufer and S. Wyss, "SPEEDY: A Fall Detector in a Wrist Watch," in *Seventh IEEE Internation Symposium on Wearable Computers, 2003. Proceedings.*, White Plains, 2003.

[17] A. K. Bourke, P. van de Ven, M. Gamble, R. O'Connor, K. Murphy, E. Bogan, E. McQuade, P. Finucane, G. OLaighin and J. Nelson, "Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and

conitnuous scripted activities," *Journal of Biomechanics,* vol. 43, pp. 3051-3057, 2010.

[18] J. K. Lee, S. N. Robinovitch and E. J. Park, "Inertial Sensing-Based Pre-Impact Detection of Falls Involving Near-Fall Scenarios," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 23, no. 2, pp. 258-266, 2015.

[19] A. K. Bourke, K. J. O'Donovan and G. OLaighin, "The identification of vertical velocity profiles using an inertial sensor to investigate pre-impact detection of falls," *Medical Engineering & Physics,* vol. 30, pp. 937-946, 2008.

[20] A. K. Bourke, K. J. O'Donovan, J. Nelson and G. M. OLaighin, "Fall-detectioin through vertical velocity thresholding using a tri-axial accelerometer characterized using an optical motion-capture system," in *IEEE Engineering in Medicine and Biology Society*, Vancouver, 2008.

[21] A. K. Bourke, K. ODonovan, A. Clifford, G. OLaighin and J. Nelson, "Optimum gravity vector and vertical acceleration estimation using a tri-axial accelerometer for falls and normal activities," in *IEEE Engineering in Medicine and Biology Society*, Boston, 2011.

[22] A. K. Bourke, J. V. O'Brien and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & Posture,* vol. 26, pp. 194-199, 2007.

[23] S.-L. Hsieh, C.-C. Chen, S.-H. Wu and T.-W. Yue, "A wrist -worn fall detection system using accelerometers and gyroscopes," in *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, Miami, 2014.

[24] E. Casilari and M. A. Oviedo-Jimenez, "Automatic Fall Detection System Based on the Combined Use of a Smartphone and a Smartwatch," *PLoS ONE,* vol. 10, no. 11, p. e0140929, 2015.

[25] C. V. C. Bouten, K. T. M. Koekkoek, M. Verduin, R. Kodde and J. J. D, "A Triaxial Accelerometer and Portable Data Processing Unit for the Assessment of Daily

Physical Activity," *IEEE Transactions on Biomedical Engineering,* vol. 44, no. 3, pp. 136-147, 1997.

[26] F. Bagala, C. Becker, A. Capello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra and J. Klenk, "Evaluation of Accelerometer-Based Fall Detection Algorithms on Real-World Falls," *PLoS ONE,* vol. 7, no. 5, p. e37062, 2012.

[27] O. Aziz, M. Musngi, E. J. Park, G. Mori and S. N. Robinovith, "A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials," *Medical & Biological Engineering & Computing,* vol. 55, no. 1, pp. 45-55, 2017.

[28] J. M. Srygley, T. Herman, N. Giladi and J. M. Hausdorff, "Self-Report of Missteps in Older Adults: A Valid Proxy of Fall Risk?," *Archives of Physical Medicine and Rehabilitation,* vol. 90, no. 5, pp. 786-792, 2009.

[29] C. M. Arnold and R. Faulkner, "The history of falls and the association of the timed up and go test to falls and near-falls in older adults with hip osteoarthritis," *BMC Geriatrics,* vol. 7, no. 1, pp. 17-25, 2007.

[30] mklinkenberg, "Alberta health officials to replace medical alert devices that lead to strangulation death," *Edmonton Journal,* 17 February 2015.

[31] N. Golgowski, "Elderly Woman Strangled By Medical Alert Necklace After Fall," *The Huffington Post,* 03 03 2016.

[32] S. Zihajehzadeh, T. J. Lee, J. K. Lee, R. Hoskinson and E. J. Park, "Integration of MEMS Inertial and Pressure Sensors for Vertical Trajectory Determination," *IEEE Transactions on Intrumentation and Measurement,* vol. 64, no. 3, pp. 804-814, 2015.

[33] G. Wu, "Distinguishing fall activities from normal activities by velocity characteristics," *Journal of Biomechanics,* vol. 33, pp. 1497-1500, 2000.

[34] R. Igual, C. Medrano and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomedical Engineering OnLine,* vol. 12, pp. 66-89, 2013.

[35] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell and B. G. Celler, "Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring," *IEEE Transactions on Information Technology in Biomedicine,* vol. 10, no. 1, pp. 156-167, 2006.

[36] A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical Engineering & Physics,* vol. 30, pp. 84-90, 2008.

[37] J. K. Lee, E. J. Park and S. N. Robinovitch, "Estimation of Attitude and External Acceleration Using Inertial Sensor Measurement During Various Dynamic Conditions," *IEEE Transactions on Instrumentation and Measurement,* vol. 61, no. 8, 2012.

[38] D. J. Goodenough, "ROC Theory: Limitations and Applications," *IEEE Transactions on Nuclear Science,* Vols. NS-29, no. 4, pp. 1359-1367, 1982.

[39] M. Kangas, A. Konttila, I. Winblad and T. Jamsa, "Determination of simple thresholds for accelerometry-based parameters for fall detection," in *29th Annual International Conference of the IEEE EMBS*, Lyon, France, 2007.

[40] J. Klenk, L. Schwickert, L. Palmerini, S. Mellone, A. Bourke, E. A. F. Ihlen, N. Kerse, K. Hauer, M. Pijnappels, M. Synofzik, K. Srulijes, W. Maetzler, J. L. Helbostad, W. Zijlstra and K. Aminian, "The FARSEEING real-world fall repository: a large-scale collaborative database to collect and share sensor signals from real-world falls," *European Review fo Aging and Physical Activity,* vol. 13, no. 8, 2016.

[41] S. N. Robinovitch, F. Feldman, Y. Yang, R. Schonnop, P. M. Leung, T. Sarraf, J. Sims-Gould and M. Loughin, "Video capture of the circumstances of falls in elderly people residing in long-term care: an observational study," *Lancet,* vol. 381, pp. 47-54, 2013.

[42] M. J. Mathie, A. C. F. Coster, N. H. Lovell and B. G. Celler, "Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological Measurement,* vol. 25, pp. R1-R20, 2004.

[43] S. Chaudhuri, L. Kneale, T. Le, E. Phelan, D. Rosenberg, H. Thompson and G. Demiris, "Older Adult's Perception of Fall Detection Devices," *Journal of Applied Gerontology,* vol. 36, no. 8, pp. 915-930, 2017.

[44] E. Foxlin, "Pedestrian Tracking with Shoe-Mounted Inertial Sensors," *IEEE Computer Graphics and Applications,* vol. 25, no. 6, pp. 38-46, 2005.

[45] M. Kangas, I. Vikman, L. Nyberg, R. Korpelainen, J. Lindblom and T. Jamsa, "Comparision of real-life accidental falls in older people with experimental falls in middle-aged test subjects," *Gait & Posture,* vol. 35, pp. 500-505, 2012.

[46] J. Klenk, C. Becker, F. Lieken, S. Nicolai, W. Maetzler, W. Alt, W. Zijlstra, J. M. Hausdorff, R. C. van Lummel, L. Chiari and U. Lindemann, "Comparison of acceleration signals of simulated and real-world backward falls," *Medical Engineering & Physics,* vol. 33, pp. 368-373, 2011.

[47] M. Mathie, *Monitoring and Interpreting Human Movement Patterns Using a Triaxial Accelerometer,* The University of New South Wales, 2003.

Appendix A.

Synchronization Methods Developed and Employed for Sony Action Cam and Prosilica GS, Respectively

The Sony Action Cam was used for free-living ADL trials which required following the participant, and also for other trials conducted outside the laboratory. An L.E.D. was mounted in front of the camera (Figure 6.1a) for event marking and was visible at the top of the video's frame (Figure 6.1b). The L.E.D. was interfaced to the Android tablet via the IOIO development board and was blinked (Figure 6.1c) during start/stop of sensor recording and during specific events of the trial (event marking). Event marking was controlled through a button press in the custom Android tablet app which at the same time logs the corresponding timestamp in a text file during the button press.

Figure 6.1. Sony ActionCam setup, wherein a) a L.E.D. is mounted on top of camera for synchronization; b) the L.E.D. is slightly seen on top portion of video frame; and c) the L.E.D. is blinked to mark an event

The Prosilica GS camera was equipped with a TTL interface and was synchronized to the Android tablet via the IOIO development board for triggering the start/stop of recording using TTL signals. It was also interfaced to a MATLAB software (The MathWorks Inc.) running on a laptop computer (through an Ethernet cable) for logging the video files and for initializing the camera. Due to the camera's size, weight, and power requirements, it was only used in the laboratory, where it could be mounted onto a tripod.

Appendix B.

Brief Explanation of the Kalman Filters Used in Studies 1 and 2 [37, 32]

Two Kalman filters, a tilt Kalman filter and a vertical position and velocity (VPV) Kalman filter, were cascaded and were used in Studies 1 and 2 for estimating the parameters[40]. The tilt Kalman filter fuses accelerometer and gyroscope data to estimate the sensors' tilt angles (i.e. roll and pitch) which was then used to estimate the gravity vector, and consequently, the kinematic acceleration. To estimate the tilt angles, the tilt Kalman filter uses the acceleration data during static conditions since its output during such conditions is dominated by the gravitational acceleration. However, during dynamic conditions, the accelerometer outputs the sum of gravitational acceleration and kinematic acceleration, wherein the two cannot be distinguished from each other. During such conditions, the Kalman filter gives more weight to the gyroscope data, which through strapdown integration also provides angle estimates. The greater the magnitude of the kinematic acceleration, the more weight is given to the estimates from the gyroscope data.

On the other hand, the vertical position and velocity Kalman filter fuses the outputs of the tilt Kalman filter, including the estimated kinematic acceleration, with the barometric pressure data to have an accurate and drift free estimate of vertical position and velocity. The tilt angles were used to convert the kinematic acceleration from the sensor frame to the inertial frame to get the vertical component of acceleration. Vertical acceleration is integrated to estimate vertical velocity and the latter is also integrated to estimate altitude. Since any errors in estimating vertical acceleration and vertical velocity gets propagated to vertical velocity and altitude, respectively, it causes boundless drift to both estimates (especially for altitude). To prevent altitude drift, the VPV Kalman filter uses barometric pressure data (average filtered) to periodically correct the altitude estimates, which consequently also prevents drift in vertical velocity. Zero velocity

[40] For a complete and detailed explanation of the Kalman fitlers used in the study, please refer to references [37] for the tilt Kalman filter and [32] for the cascading of a tilt and a VPV Kalman filter (however, it should be noted that the VPV Kalman filters used in in Studies 1 and 2 is a modified version this).

update was also used in the VPV Kalman filter which sets the vertical velocity to zero during static conditions. Since for Study 1, there could be strong disturbance in pressure data during dynamic movements due to the sealed enclosure of the smartwatch, pressure data was selectively used (SUP). Additionally, since the duration of pressure disturbance could be long (depending on the person's activity and the watch's response), high-pass filtering of external acceleration was conducted whenever there is no reliable pressure and motion is not static to help prevent drift in vertical velocity.

Appendix C.

Misclassified Activities of Algorithm 2 (Study 1) at Different Accelerometer Ranges and Sampling Rates

Table 6.1. Experimental results. Misclassified activities of Algorithm 2 at different accelerometer ranges and sampling rates during simulated laboratory-based trials of Study 1[41]

	100 Hz			50 Hz		
	±8 g	**±4 g**	**±2 g**	**±8 g**	**±4 g**	**±2 g**
False Negatives						
Trip	0	0	0	0	0	1
Hit/Bump	0	0	0	0	0	0
Collapse	0	0	0	1	1	1
Cross-step	0	0	0	0	0	1
Stand-to-sit	0	0	3	0	0	2
Sit-to-stand	0	0	0	0	0	0
Roll-bed	0	0	0	0	0	1
(Total)	(0)	(0)	(3)	(1)	(1)	(6)
False Positives (ADLs)						
Normal walk	0	0	0	0	0	0
Sit-to-stand	0	0	0	0	0	0
Stand-to-sit	0	0	0	0	0	0
Wash-hands	2	2	2	3	3	1
Lay-on-bed	2	2	1	1	2	1
Ascend-stairs	0	0	0	0	0	0
Descend-stairs	0	0	0	0	0	0
(Total)	(4)	(4)	(3)	(4)	(5)	(2)
False Positives (Near-Falls)						
Trip	1	3	0	2	4	1
Hit/Bump	0	0	0	0	0	0
Cross-step	0	0	0	0	0	0
Sit-to-stand	2	2	1	2	2	1
(Total)	(3)	(5)	(1)	(4)	(6)	(2)

From ±8g to ±4g the ADL false positives and false negatives are at least very similar (within their respective sampling rates). For ADL false positives, a possible reason is that for these activity types, the range of acceleration is mostly below the ±4g

[41] Note that since the analyses for pick-object and reach-object trials for Algorithms 1 and 2 were only recently conducted as a response to a manuscript revision, the results in Table 6.1 does not reflect analysis from such trials types.

range, as could be seen in Fig. 4.10b (although the plots only represents the vertical component). For the false negatives, although the fall activities' vertical acceleration peak could go beyond ±4g (see Fig. 4.10b), possible reasons for the same algorithm performance would include: (a) during the descent phase, for the decreasing vertical velocity (see Fig. 4.6d), what is only integrated is the trough in vertical acceleration before impact (see Fig. 4.6a) which could only reach a minimum of -1g; and (b) during the impact phase of a fall, since the highest magnitude of threshold for vertical acceleration is only 20 m/s^2 (see Table 4.1), vertical acceleration's requirement will already be satisfied even before the acceleration gets clipped at 4g[42]. The increase in near-fall false positives with the reduction of accelerometer range from ±8g to ±4g cannot be explained. For the reduction of sampling rate from 100 Hz to 50 Hz (at same accelerometer ranges of ±8g and ±4g), a possible reason for the addition of a false negative is that since fall is a very quick event, the reduction of sampling rate could had resulted to the missing of critical sensor information during the fall.

From ±8g and ±4g to ±2g, a possible reason for the increase in false negatives is that, with the smaller acceleration range, falls that could only pass the second path of Algorithm 2 may not had been detected due to the path's 20 m/s^2 threshold[43]. Similar to ±8g and ±4g, the additional increase in false negatives with the reduction of sampling rate could also be caused by missing critical sensor information during the fall.

[42] Since it is the positive spike in vertical acceleration that pushes vertical velocity back to zero after impact (i.e. right after maximum negative vertical velocity; see Figs 4.6d and 4.6a), having the acceleration clipped at ±4g is expected to affect this returning of vertical velocity to zero. However, if the person's wrist movement is static after the fall, zero velocity update will be able to push vertical velocity back to zero.

[43] Note that since path 1's vertical acceleration threshold is only 10 m/s^2 (see Table 4.1), vertical acceleration requirement for falls that could pass through this path are not expected to be affected by the acceleration clipping at ±2g.