

國立清華大學

碩士論文

題目：穿戴式裝置上之九軸感測器的姿態辨識

Gesture Recognition with Wearable 9-axis Sensors

系所別：電機工程學系碩士班 組別：系統組

學號姓名：103061628 劉芳廷 (Fang-Ting Liu)

指導教授：馬席彬 博士 (Prof. Hsi-Pin Ma)

中華民國一〇五年九月

摘要

姿態辨識近年來在機算機科學領域裡面是一個很熱門的話題，他的目標是能夠透過數學演算法來表達人類的姿態，它可以應用在許多不同技術上，例如：行動電話上的應用、穿戴式的無線裝備上、運動上的偵測、電視遊戲或是與藝術的結合。

在這篇論文中，我們將會在手腕內側配戴9軸感測器（包誇加速度計、陀螺儀及磁力計）來紀錄8種動作的訊號並放到電腦裡去計算與分析，使用論文所提出的演算法加以辨識受測者的動作。我們使用的是機器學習的流程來建立的辨識系統。除了使用機器學習的方式來分類所要辨識的動作，我們還開發了一個使用閾值來判斷動作的方法，這個方法較為簡單與直觀，然而並非所有的動作都可以透過如此簡單的方式來偵測，所以我們最終是以機器學習的分類方法來建立我們的系統，這兩種方法的驗證與比較也將會呈現在實驗結果的章節裡。

為了達到較高的辨識準確度，我們使用機器學習的分類流程，並且更進一步做特徵的選取與萃取，我們使用的方法為主成分分析法（principal component analysis）在加上線性判別分析（linear discriminant analysis）來萃取出較明確的特徵，主成分分析法與線性判別分析的優點為可以減少資料的維度並且減少後面分類的訓練時間，也能盡可能的保留原始資料的最大資訊量，我們也開發了一個方法來建立一個適合後面分類器的特徵矩陣以達到較好的表現。最後我們使用的分類器是支持向量機（support vector machine），這個分類器可以使辨識的精準度更高、計算時間較少、也可以支援高維度的數據，在實驗中，我們把動作分成8類，20個人一人做每個動作5次為實驗數據，在使用者相依的狀況下可以得到99.63%的準確度，而在使用者無關的狀況下我們收集了12個人的測試資料並得到88.43%的準確度。

致謝

首先謝謝我們的指導教授馬席彬老師，在研究所這兩年的期間對我們的耐心指導，讓原本不懂得如何做研究的我們能有所成長，並懂得研究的精神，也謝謝老師在研究過程當中給我們許多的建議和想法，這兩年真的學到了很多，不僅是專業研究方面，還有表達能力也增長了不少。

謝謝實驗室的同學們：彥~勳~、阿昌、李泓毅、阿光、米嘉、庭瑜、盧彥丞，在我忙碌的時候還不時的關照著我給我幫助，讓我可以順利的完成研究所的學業，謝謝你們隨時隨地帶來的歡樂，讓這兩年的回憶增添了許多色彩。

謝謝學長姊：曼蒂、易廷、冠廷、維漢，分享了很多經驗給我，在研究上也給予我許多幫助及想法。

謝謝學妹詠婷，在我的研究上給予我許多的幫忙，幫我減輕了許多研究上的壓力與工作量，讓我的研究能夠順利完成，希望你之後這一年也可以研究順利，順利畢業。

謝謝好友們：君庭、小淨、大頭、哥、陳昱廷、肥蛋、Zn+、范綱，在這兩年給予我研究上的幫助，常常幫我排遣壓力、解決心事，還有一直抓著我血拼，有你們真好，讓我大學畢業後的生活還是跟以前一樣快樂。

最後要謝謝我的爸爸媽媽，在我遇到瓶頸時總是支持著我，給與我建議與想法，也時時聽我分享生活上的鎖事，謝謝你們從小到大對我的付出與關懷，讓我擁有足夠的資源學習到現在。

Gesture Recognition with Wearable 9-axis Sensors

Fang-Ting Liu

Department of Electrical Engineering,
National Tsing Hua University, Hsinchu, Taiwan

September, 2016



Abstract

Gesture recognition is a topic in computer science with the goal of describing human gestures through mathematical algorithms in recent year. In the field of hand gesture recognition, it apply in many kinds of technologies such as mobile phone applications, wearable wireless devices, sports detection, video game or art combination.

In this thesis, we will record signals of eight kinds of hand movements into computer using wearable wireless device with nine axis sensor (including accelerometer, gyroscope and magnetometer) worn on the wrist, then recognized gestures using the algorithms being described later. We built a system of recognition with machine learning classification process.

Besides classification process, we also developed a thresholding method to easily detect movements. In the thresholding method, for each movement, we defined threshold value for each kind of data and filtered the movements data with threshold combined with detection windows. However, not all the movements can be detected by this easy and less calculation method so that we finally used a machine learning process to solve problems. The analyzing of the two method will be introduced later.

In order to achieve higher recognition accuracy, we used machine learning process in the system and did feature extraction to get well distinguished features. We used principal component analysis (PCA) and linear discriminant analysis (LDA) to extract features. The advantages of PCA and LDA are reducing dimensions of data while preserving as much of the class discriminatory information as possible and reducing the training time of classification. Last, with support vector machine (SVM), we can recognize movement with higher accuracy with less computation time, and it also support data with high dimension. We can model even non-linear relations with more precise classification due to SVM kernels. In our experiment, we can get the accuracy of recognition at 99.63% for 8 classes with 20 subjects data for 5

times each in user-dependent case, and 12 subjects testing data for user-independent case with recognition rate at 88.43%.



Contents

Abstract	i
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Main Contributions	3
1.4 Organization	4
2 Overview of Gesture Recognition Technologies	5
2.1 Different Sensing Technologies in Gesture Recognition	5
2.1.1 Inertial Sensing Technology	6
2.1.2 Image Based Technology	6
2.1.3 Inertial Sensing Integrate with Image Based Technology	7
2.1.4 Glove Sensing Technology	8
2.1.5 Optical Sensing Technology	8
2.1.6 Acoustic Sensing Technology	9
2.1.7 Radio frequency Sensing Technology	9
2.1.8 Summary of Different Technologies	10
2.2 Gesture Recognition Algorithms of Inertial Sensing Technology	12
2.2.1 Characteristics of Features	12
2.2.2 Detection Methods	13
2.3 Comparison of Different Classifier using Inertial Sensing Technology	16

3	Proposed System and Algorithms	21
3.1	Overview	21
3.2	Architecture of System	21
3.3	Introduction to Inertial Body Sensor (Nine Axis Sensor)	23
3.3.1	MPU9250	23
3.3.2	Signals of Nine-Axis Sensor	25
3.4	Definition of Movements to be Recognized	30
3.5	Thresholding Method with Scanning Window	32
3.6	Dimension Reduction and Feature Extraction	36
3.6.1	Dimension Reduction using Principal Component Analysis Algorithm	37
3.6.2	Feature Extraction using Linear Discriminant Analysis Algorithm	42
3.6.3	Comparison of PCA and LDA Algorithm	46
3.7	Data Classification	46
3.7.1	Theory	47
3.7.2	Feature Vectors for Support Vector Machine	48
3.7.3	System Classification using SVM Classifier	49
4	Implementation Results	53
4.1	Experiment	53
4.1.1	Experimental Environments	53
4.1.2	Experiment Procedure	55
4.2	Analysis of Received 9-Axis Signals	56
4.3	Result Analysis	59
4.3.1	Thresholding Method Analysis	59
4.3.2	SVM Analysis in User-dependent/User-independent Case	60
4.3.3	Comparison of Thresholding and SVM	64
4.3.4	Comparison between Our Work and Some Related Studies	66
5	Conclusions and Future Works	71
5.1	Conclusions	71
5.2	Future Works	72

List of Tables

2.1	Comparison of different sensing technologies.	11
2.2	Literature comparison.	18
2.3	Method comparison.	19
3.1	Specification of MPU9250.	24
3.2	Specification of each sensors in MPU-9250.	24
4.1	Experimental environments.	53
4.2	Thresholding result for the 2 movements - Up and Left.	59
4.3	Accuracy under different combination of C and γ	60
4.4	Confusion matrix in user-dependent case based on the method of PCA+LDA+SVM with kernel model when widow size is 50.	62
4.5	Confusion matrix in user-independent case based on the method of PCA+LDA+SVM with kernel model when widow size is 50.	63
4.6	Comparison of different sample rate.	63
4.7	Comparison of different window size in sample rate of 50 Hz.	64
4.8	Confusion matrix in user-dependent case based on the method of PCA+LDA+SVM with kernel model when widow size is 100.	65
4.9	Accuracy comparison of the 2 method.	66
4.10	Comparison of or work with related studies.	68
4.11	Comparisons to the related works using SVM classifier in user-dependent case.	69



List of Figures

2.1	Block diagram of the CHG technique using smart devices.	7
2.2	An example of $k - NN$ with $k = 3$ and $k = 11$. In the figure, when $k = 3$, it calculate the nearest 3 objects and vote the amount of each class. The new sample (in star sign) is classified to class A. When $k = 11$, it calculate the nearest 11 objects and vote the amount of each class. The new sample is also classified to class A.	14
2.3	An example of SVM with 2 classes. The goal of SVM is to find a plane (optimal separating hyperplane) which can separate the 2 classes (white dots and black dots) with the margin being as large as it can be, so that we can distinguish the 2 classes clearly. Support vectors are the data points that lie closest to the optimal separating hyperplane. Thus, H1 and H2 are support hyperplanes with support vectors on them.	15
3.1	Flow chart of the gesture recognition system.	22
3.2	Block digram of MPU-9250.	25
3.3	Sensory node implementation.	26
3.4	Flow chart for accessing the sampled data of gyroscope and accelerometer. . .	26
3.5	Flow chart for accessing the sampled data of magnetometer.	27
3.6	The structure of a MEMS accelerometer.	28
3.7	Velocity and acceleration vectors present in the Coriolis Effect.	29
3.8	Implementation of a MEMS gyroscope.	30
3.9	Lorentz force affect on the charge q in a velocity v under the influence of a magnetic field B	31

3.10	Hall effect on a semiconductor.	31
3.11	Movements definition. (a. Up, b. Down, c. Left, d. Right, e. Counter clockwise circle, f. Clockwise circle, g. Wrist turn right, h. Wrist turn left) . . .	33
3.12	Threshold flow chart.	34
3.13	The acceleration waveform of x-axes, y-axes and z-axes with minimum and maximum value.	35
3.14	The acceleration waveform of x-axes, y-axes and z-axes with interval divided by 4. The up-interval is between 0 and the maximum value, similarly, the down-interval is between 0 and the minimum value. ($Q1 = \text{interval} * \frac{1}{4}$, $Q2 = \text{interval} * \frac{2}{4}$, $Q3 = \text{interval} * \frac{3}{4}$)	36
3.15	Thresholdind method integrated with scanning windows. The translucent area is condition passing the threshold, and we have marked the sequence of scanning windows.	37
3.16	Steep diagram.	41
3.17	Comparison of PCA and LDA.	46
3.18	A dataset with scanning window which window size is 50.	49
3.19	Kernel machine mapping the original data, which cannot be separated linearly, into a higher-dimensional space to find a hyperplane. (a. High-dimensional data b. Higher-dimensional data)	50
3.20	Different situation in different γ . a. $\gamma = 1$, b. $\gamma = 10$, c. $\gamma = 100$	51
3.21	A schematic diagram of margin violation.	52
4.1	Directions and placement of the sensor.	54
4.2	Schematic diagram of the range of movement.	55
4.3	Orientation of axes of sensitivity and polarity of rotation for the nine-axis sensor.	56
4.4	The accelerometer data generated by certain movement sequentially.	57
4.5	The gyroscope data generated by certain movement sequentially.	58
4.6	The magnetometer data generated by certain movement.	58
4.7	Comparison of accuracy under different C	61
4.8	Line chart for accuracy comparing of different window size.	65

4.9	Accuracy comparison of the 2 method. (a)Thresholding method. (b)SVM. . .	66
-----	--	----





Chapter 1

Introduction

1.1 Background

Gesture recognition can be used to understand and classify meaningful movements by a humans hands, arms and face through mathematical algorithms in computer science. It has become one of the hottest fields of topic because of great significance in designing artificially intelligent human-computer interfaces for various applications such as sign language and medical rehabilitation. Literature shows a large amount of works using vision sensors or inertial body sensors for gesture recognition of human body movements.

In this thesis, we use inertial body sensor (nine axis sensor) to measure hand movements. There are many recognition and measure systems with such sensor appear in the literature. For example, a human body motion capture system using wireless inertial sensors was presented in [1]. In [2], an accelerometer-based digital pen for handwritten digit and gesture recognition using trajectory recognition algorithm. In [3], a customizable wearable body sensor system was introduced for medical monitoring and physical rehabilitation.

For gesture recognition, some use Euler angles to define and measure movement, but now, most of people use machine learning task — supervised learning, or more complicated, unsupervised learning to do that. Compare to Euler angles, supervised learning algorithm not only can get high accuracy, but also analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for

the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a reasonable way. In the thesis, we will introduce a supervised algorithm - support vector machine (SVM) about how it works and its applications.

1.2 Motivation

Living in a city with fast pace and busy life, most of people are concentrated in their work and are lack of time to relax themselves. Being under a lot of pressure and not relaxing ourselves appropriately will easily lead people to a bad mood and poor quality of life, even more, overstrained and unhealthy body. Being in a bad mood and being anxious are common hidden problems of modern people, and there are more and more people suffered from emotion disorder, such as depression and anxiety. Although the medical technology is developed very well and existing many consultative center to help people solving psychological problems, people still need to solve the core issue - practicing how to relax and adjust mood by ourselves.

Many research has been published for applications and studies for gesture recognition field. In [4], they presented a continuous hand gestures (CHG) recognition technique, which extended from dynamic timing wrapping (DTW), being capable of continuous recognition of hand gestures. They develop a prototype system for gesture recognition using a smartphone as an input smart device. The proposed system can also be used on any smart device which has the accelerometer and gyroscope sensors and interacts with a machine. In [5], it presents a hand gesture recognition system using magic ring (a ring-shape wearable device with accelerometer on it) instead of data glove or camera to translate the predefined hand gestures into Japanese and to accomplish remote transmission of the information/command between user groups. The k-nearest neighbors (kNN) method is adopted to the pattern recognition according to the bimanual sensing data. In [6], this paper presents the design and implementation of a system of accelerometer-based hand gesture recognition. As the recognition of hand gestures is a pattern classification problem, two techniques based on artificial neural networks are explored: multilayer perceptron and support vector machine. For the research mentioned above, the accuracy are still not high enough and the system were not suitable for

our future applications, we want to improve the accuracy and combine the system with more robust feature extraction method and classifier.

The accuracy is the main topic in gesture recognition system, thus, we propose a wearable gesture recognition (motion detection) system which can recognize hand movement and applies in various field. We had a robust feature extraction method and used support vector machine (SVM) which can get higher accuracy to complete our system. By the system, it can solve the accuracy issue.

With our system, it can bring a lot of interest not only in mobile phone game applications but also in interactive game in our life. In the future, in addition to applications in interactive things, we can use the gesture recognition algorithm in life, with the motion data, we can study the movements of a person and determined his/hers behavior of daily life, furthermore, we can detect people's changing of mood and give them some alert to let them know they are overstrained by his/hers behavior.

1.3 Main Contributions



Main contributions of this thesis are listed as follow:

- Data analysis
 - Analyze the 9-axis data in detail.
 - Defined the values of threshold equals to $\frac{1}{2}$ of the maximum/minimum value of signal and zero to make the system distinguish gestures easily with less computation time.
- System design
 - Build a system with machine learning task – supervised learning.
- Result analysis
 - Improve the accuracy to 99.63% in 8 classes with parameters: $\gamma = 0.03125$ and $C = 0.25$.

- Analyze accuracy of the results and parameters of support sector machine (SVM) which need to be found out for making the result acceptable.

1.4 Organization

We have made a brief introduction to this thesis in Chapter 1. In Chapter 2, we will introduce what gesture recognition is, sensor information and the classification research. Data structure and implemented algorithms will be mentioned in Chapter 3. In Chapter 4, we will introduce the experiment of recognition, analyze the data and show the result of our system. In last Chapter, future work and conclusion of the thesis will be presented.



Chapter 2

Overview of Gesture Recognition Technologies

In this chapter, we will introduce applications and characteristics of gesture recognition field, and consideration for choosing the recognition algorithm.

2.1 Different Sensing Technologies in Gesture Recognition

Hand gesture is a high practical value body language that its specific meaning is established through our language center by the palm and finger position and shape. Generally speaking, the gesture consists of static and dynamic hand gestures. As its name suggests, static hand gestures refer to the individual shape of the hand, and the dynamic hand gestures are composed of a series of hand movements [7]. Gesture recognition is a hot topic in research recently. The recognition system can be implemented by many kinds of methods. For example, in [8], it present a real time system for hand gesture recognition on the basis of detection of some meaningful shape based features like orientation, center of mass (centroid), status of fingers, thumb in terms of raised or folded fingers of hand and their respective location in image. In [9], it presents a system for quick and effective recognition of gestures of hand body language, based on data from a specialized glove equipped with ten sensors. In [10], it proposes a kinect-based calling gesture recognition scenario for taking order service of an elderly care robot. The proposed scenarios are designed mainly for helping non

expert users like elderly to call service robot for their service request. Many gesture tracking and recognition technologies have been proposed. In general, these current gesture tracking technologies derive pose estimates from electrical measurements received from mechanical, magnetic, acoustic, inertial, optical, radio or microwave sensors [11] [12] [13]. When applied to gesture recognition, most of these technologies can be used alone with good results. Next, we will introduce various sensing technologies in gesture recognition field.

2.1.1 Inertial Sensing Technology

Recent advances in smart devices have sustained them as a better alternative for the design of humanmachine interaction (HMI), because they are equipped with accelerometer sensor, gyroscope sensor, and an advanced operating system. It is called sensor-based gesture recognition (SGR) techniques. A SGR technique consists of a hand gesture action. A gesture recognition technique is said to be a continuous hand gesture recognition technique, if the start and end points of gestures are automatically detected. In [4], they presented a continuous hand gestures (CHG) recognition technique using three-axis accelerometer and gyroscope sensors in a smart device, Figure 2.1 illustrates the block diagram of CHG technique. In data acquisition step, a smart device collects the sensory data from the accelerometer and gyroscope sensors. The collected sensory data, then work as input of the data preprocessing step where the raw sensory data are compressed and coded. They create a gesture database for recognizing the gestures in gesture database creation step. The gesture spotting process automatically identifies the start and end points of each gesture. The coded sensory data finally process to the gesture reorganization step. They used DTW in the recognition step and got recognition rate of 93.56% in their experiment with 12 classes.

2.1.2 Image Based Technology

The recognition of human gesture in video data can be used for automatic video monitoring, public safety can be guaranteed through observing the human gesture of videos. There are various human gestures, 2-dimensional image features are generally used to depict the human gestures. Human motion involves many parts, the contour of the human body is a very

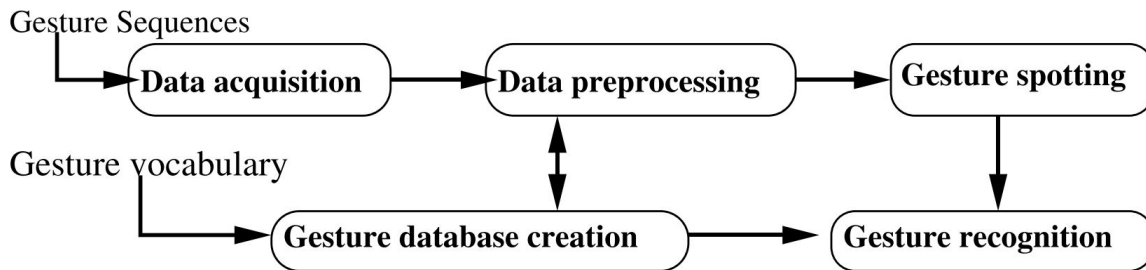


Figure 2.1: Block diagram of the CHG technique using smart devices [4].

important feature in human gesture recognition. A human action can be recognized through the body contour, the gestures of other body parts cannot be observed.

In [14], this study used KNN, Bayes, LDA and SVM classifiers to recognize two human gestures (walk and bend). This study extracted the distances between the contour and center of human body as the feature sets, 5-fold cross-validation indicates that the performances of SVM and LDA are better than KNN and Bayes with recognition rate of 99.28% and 98.91% respectively.

Another image based technology is using Kinect. In the field of view of Kinect, maximum 6 users can be detected and two of them can be tracked with skeleton joints details. Kinect also can track users both in standing and sitting mode in real time. With these advantages of Kinect sensor, skeleton based gesture recognition attracted attentions of developers all over the world. In [10], it proposes a Kinect-based calling gesture recognition scenario for taking order service of an elderly care robot. There are two modes of our calling gesture recognition: Skeleton based gesture recognition and Octree based gesture recognition.

2.1.3 Inertial Sensing Integrate with Image Based Technology

Since an inertial sensor is better at tracking rapid movements, while a vision sensor is more stable and accurate for tracking slow movements, a novel adaptive algorithm has been developed to adjust measurement noise covariance according to the measured accelerations and the angular rotation rates.

For real-time gesture tracking, inertial sensors suffer from the zero-drift problem while vision sensors have poor performance for resolving fast motions due to motion blur and oc-

clusions. Hence, neither of them is perfect for gesture tracking alone. Hybrid gesture tracking base on vision and inertial sensor fusion offers not only fast motion tracking and good stability, but also robust performance over occlusions [15].

In [16], the algorithm has been developed to track the real-time position and orientation of a μ IMU+Camera (μ IC) system by fusing data from a MEMS-based inertial sensor and a vision sensor. The inputs to this system are the 3D accelerations and the angular rates from the μ IMU and pose estimation from the vision sensor. They have built our motion tracking system by using a single low resolution camera commonly used for internet communications (i.e., a webcam) and a MEMS-based inertial sensor to capture the motion. In order to address the synchronization problem due to the different sensor data acquisition frequencies, they have developed an algorithm for calculating the optimum filter length for a moving average filter. It not only helps to remove the high frequency noise, but also propagates the inertial data to solve the data synchronization problem. The experimental results also prove that the reconstructed ten Arabic numerals can be recognized with DTW at an accuracy of 92.3%.

2.1.4 Glove Sensing Technology

The final goal of the gesture analysis is to extract characteristic features describing the movement of a hand and fingers. Gloves equipped with sensors like accelerometers, gyroscopes, etc., seem to be ideally suited for this task, as the data recorded by this kind of sensors is directly related to the hand movements.

In [9], they recognized 22 hand body language gestures using the glove with 10 sensors in it. The data for gestures recognition were collected with the DG5 VHand glove device. The features are record by flexion sensors in five fingers, a 3-axis accelerometer and a gyroscope. The best classifier achieved the sensitivity equal to 98.32%.

2.1.5 Optical Sensing Technology

The optical sensing technology, it often used to recognize various hand language (or sign language). The input is image recording by camera and the lightning condition, color and 3D depth map which transferring by image are used to detect and track the hand shape. In [17],

they recognized the 26 American Sign Language (ASL) alphabet from hand images. Using lightning conditions, angle and the distance between the hand and the camera are the main factors while collecting images for sign language data. An average 93.23% recognition rate was achieved by using fuzzy-c-means algorithm. In [18], they used color and 3D depth map to detect and track the hand, then combined features of location, orientation and velocity. 26 ASL alphabet and 10 Arabic numerals were recognized with a 98.33% recognition rate when 720 video samples were used for training and 360 video sequences for testing.

2.1.6 Acoustic Sensing Technology

The sounds produced during any activity carry information about what is occurring. In many pattern recognition situations, sound can be recorded conveniently, and can supplement or replace visual sources of information. The sounds generated by a writing instrument provide a rich and under-utilized source of information for pattern recognition. Using features based on peak count and relative amplitude in the gated audio power signal can classify different patterns.

In [19], they investigate the acoustic emissions produced by a human writer, analyzing the sounds that are generated by the friction between the writing instrument and surface. Recognized characters by identifying sound patterns generated from a written instrument and recognition rate of over 70% (alphabet) and 90% (26 words) were achieved.

2.1.7 Radio frequency Sensing Technology

RFID tags are used for supply chain management and automatic identification of objects [20] [21]. They also allow us to augment physical objects and the environment with digital information [22] and to monitor indoor human activities [23] or to detect people's interactions with RFID-tagged objects [24]. RFID tags have also been deployed as landmarks on the floor or within the environment to support navigation of mobile robots equipped with RFID readers [25] [26]. Moreover, many location-sensing techniques have been developed that use RFID technology. In [27], the motion patterns of passive RFID tags and hand gesture were tracked by using multiple hypothesis tracking and subtag count information with an accuracy

of 93%.

2.1.8 Summary of Different Technologies

Mechanical sensors provide accurate pose estimates and have a low latency, but their mobility is low and they usually occupy a large volume of space. Magnetic sensors are also accurate for pose estimation, have a low latency, and good mobility [28]. But the problem is that they are vulnerable to distortions from conductive objects in the environment, and the signal attenuates quickly with an increase in distance between the magnet and the sensor. Acoustic sensors are small in size, lightweight and have good mobility, but their accuracy is affected by background ambient noise and, atmospheric effects. They also require a fairly unobstructed line-of-sight between the emitters and the receivers. MEMS-based inertial sensors are lightweight, good for fast motion tracking, and can cover a large sensing range, but they lack long term stability due to the problem of severe zero drift. Optical sensors are very accurate and have no accumulated errors, but their ability to resolve fast movements is poor due to motion blur. They also suffer from line-of-sight limitations. For radio and microwave sensing, they can cover a large tracking range and are very mobile, but their precision is low [11]. When applied to gesture recognition, most of these technologies can be used alone with good results, as summarized in Table 2.1.

From the summary, the accuracy of each sensing technology is close to each other, but the method and the using equipments is very different. Besides optimizing for an efficient algorithm, it is always desirable to use fewer, cheaper, smaller sized and simpler sensors to achieve the same goal. In this thesis, we chose inertial sensing method to apply on our system due to the light weight, conveniently equipment and high accuracy. In section 2.3, we will also compare studies using inertial sensing technology with different detection method.

Table 2.1: Comparison of different sensing technologies.

Technologies	Literature	Algorithm	Patterns	Features	Classes	Accuracy
Inertial sensing	Gupta et al. [4]	DTW	Geometry	acc, gyro	12	93.56%
Inertial sensing (digital pen)	Wang et al. [29]	PNN FNN	Geometry	extract from acc	8	98.75% 96.25%
		PNN	Digits		10	98%
Image Based	Huan et al. [14]	KNN	Walk and bend	image	2	92.86%
		Bayes				83.98%
		LDA				98.91%
		SVM				99.28%
IMU + Image sensing	Zhou et al. [16]	ARP and DTW	Digits	imag, acc, gyro	10	92.30%
Glove sensing	Plawiak et al. [9]	SVM	Hand body language	acc, roll, pitch	22	98.32%
		KNN				97.36%
		PNN				97.23%
Optical sensing (monocular camera)	Amin et al. [17]	Fuzzy-c-mean clustering	ASL finger alphabets (a-z)	image	26	93.23%
Optical sensing (stereo camera)	Elmezain et al. [18]	HMM	Alphabets and digits	image -> location, orientation, velocity	36	98.33%
Acoustic sensing	Seniuk et al. [19]	Template matching	Alphabets Words	Audio	26	70% 90%
Radio frequency sensing	Asadzadeh et al. [27]	Multiple hypothesis approach	Geometry	RFID antenna readings	16	93%
Inertial sensing and magnetic sensing	Choi et al. [30]	DTW	Alphabets and digits	acc, velocity, position	36	95.70%

2.2 Gesture Recognition Algorithms of Inertial Sensing Technology

2.2.1 Characteristics of Features

From the Table 2.1 above, we can see that the features for their system are different from each sensing technology. In image based technology, it use 2-dimensional images which record from cameras as features. In glove sensing technology, the features are flexion of fingers, acceleration and angular acceleration of hand. In optical sensing technology, it not only use images but also transfer the images to get the lightning condition, color, 3D depth map as features. In acoustic sensing technology, it use the signals of sounds produced during writing as features. In radio frequency sensing, the features are the RFID tag. It has various kind and method of choosing features in different technology. Here we will introduce the features for inertial sensing technology.

In inertial sensing technology, most of studies use the sensor with at least a 3-axis accelerometer and a 3-axis gyroscope. With the sensor, signals of x-, y-, z-axis acceleration and angular acceleration are received. The signals generated from the sensor have great relevance with continuous hand movements, the acceleration and angular acceleration would change over time. It is clear that they are the most significant and obvious signals which can be taken as the features for gesture recognition technology. Most of studies directly use acceleration or angular acceleration as features like in [5], [6], and [31].

Except the raw data from inertial sensor, some other studies will use features which generated from accelerometer, gyroscope and magnetometer. For example, mean value, STD and variation of acceleration or angular acceleration signals, correlation between signals of each axes, energy of the movements, mean absolute deviation (MAD) of signals, and interquartile range (IQR) represents the dispersion of the data and eliminates the influence of outliers in the data when different classes have similar mean values, etc [32] [33].

2.2.2 Detection Methods

There are many mathematical methods used for gesture recognition (movement detection) in inertial sensing technology. However, their effectiveness is almost gauged by their accuracy. We will introduce some methods below. For each method, a brief description is followed by a simple example and pros & cons of the method. For people who are unfamiliar with the method can obtain some initial information, for people who are familiar with it can go through the introduction quickly and focus on the comparison of each method. Besides, we should know that accuracy of each method can only be an reference to the comparison of methods. It is difficult to compare accuracy directly between studies because there are many factors can influence accuracy, such as type of extracting features, normalization method and dimension reduction method.

Nearest Neighbors

In pattern recognition, the k-nearest neighbor (k-NN) algorithm is one of the simplest classification algorithms. When a new feature vector needs to be classified, the algorithm computes distance (usually Euclidean or Mahalanobis) to each feature vector in the training data set. The training vectors are then ranked according to their distance to the new sample, and the k (where k is a positive interger, and $k \geq 1$) nearest training vectors (neighbors) are used to classify the new feature vector by a majority vote of its neighbors, with the sample being assigned to the class most common among its k nearest neighbor. An example is shown in Figure 2.2.

There are two things that we have to notice before calculating distance. First, it is necessary to normalizing the scale of different data features (e.g. normalizing each feature to $[0,1]$). Because all features will contribute equally to the distance calculation. Second, since the algorithm weighs all features equally even though some features may not be relevant, dimension reduction is important and necessary. For instance, if the version of kNN is the 1-nearest neighbor, the rule is that the new feature vector will be assigned to the same class as the nearest vector in the training data set.

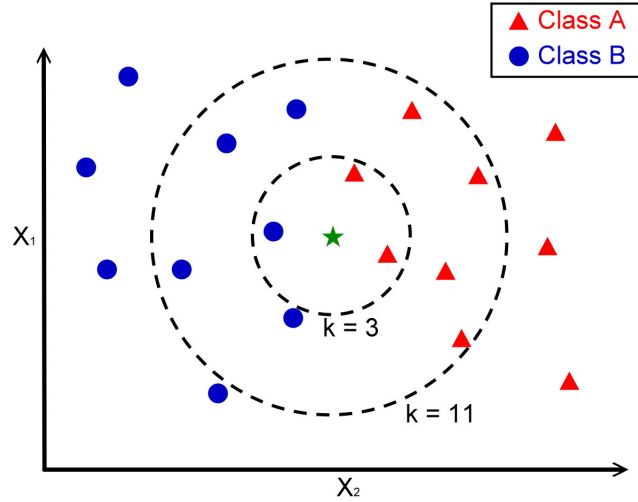


Figure 2.2: An example of k -NN with $k = 3$ and $k = 11$. In the figure, when $k = 3$, it calculate the nearest 3 objects and vote the amount of each class. The new sample (in star sign) is classified to class A. When $k = 11$, it calculate the nearest 11 objects and vote the amount of each class. The new sample is also classified to class A.

Dynamic Time Wrapping

Dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in speed. For example, similarities in speaking could be detected using DTW, even if one person was speaking faster than the other. DTW has been applied to temporal sequences of video, audio, and graphics data indeed, any data which can be turned into a linear sequence can be analyzed with DTW.

In general, DTW is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restrictions. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in time series classification. Although DTW measures a distance-like quantity between two given sequences, it doesn't guarantee the triangle inequality to hold.

The advantage of using DTW is that reliable time alignment between reference and test patterns is obtained. The disadvantage of using DTW is the heavy computational burden required to find the optimal time alignment path [34].

Support Vector Machine

Support vector machines (SVMs) are a method of generating hyperplanes in n -dimensional space (where n is the number of input features) that separate feature vectors of different classes. The principal difference between the two is the criterion used to calculate these hyperplanes. SVMs are a maximum margin classifier: it creates the hyperplane so that the distance (margin) between the hyperplane and the closest feature vectors on each side is maximized. An example is given in Figure 2.3.

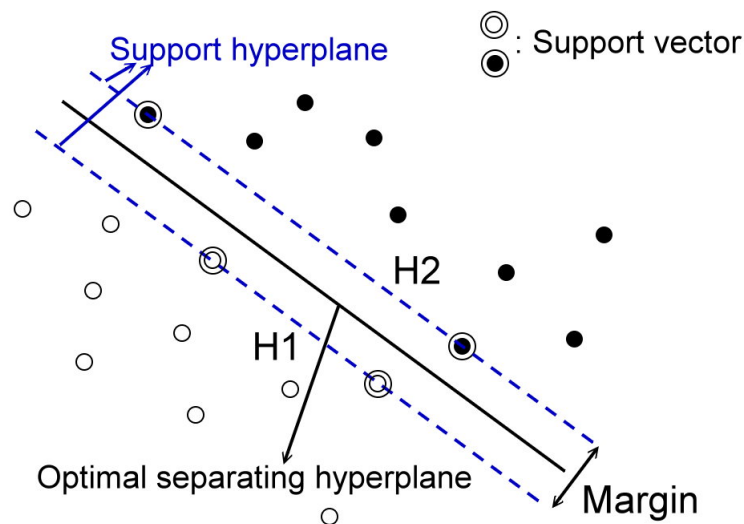


Figure 2.3: An example of SVM with 2 classes. The goal of SVM is to find a plane (optimal separating hyperplane) which can separate the 2 classes (white dots and black dots) with the margin being as large as it can be, so that we can distinguish the 2 classes clearly. Support vectors are the data points that lie closest to the optimal separating hyperplane. Thus, H1 and H2 are support hyperplanes with support vectors on them.

The advantages of the basic SVMs are transparent and it is easy to determine the contribution of each input feature. However, they are a linear classifier. In order to avoid limitation of linearity, researchers commonly expand SVMs by kernels. Kernels can be explained that the training data is transformed into a higher dimensional space and a new hyperplane is generated in this space. In the new transformed space, we will find out that the new hyperplane is linear.

But it may be nonlinear in the original feature space and become a nonlinear classifier. The good performance and nonlinearity of SVMs has led to frequent use in gesture recognition.

2.3 Comparison of Different Classifier using Inertial Sensing Technology

There are many applications for gesture recognition using different classifying methods. Here we list some paper using the algorithms mentioned above and summarized in Table 2.2.

In [31], the gesture recognition system operates primarily on data from a single 3-axis accelerometer and comprises two main stages: a training stage and a testing stage. For training, the system employs dynamic time warping as well as affinity propagation to create exemplars for each gesture while for testing, the system projects all candidate traces and also the unknown trace onto the same lower dimensional subspace for recognition. A dictionary of 18 gestures is defined and a database of over 3700 traces is created from seven subjects on which the system is tested and evaluated. They got average accuracy of 94.6% in 18 classes.

In [4], they presented a Continuous Hand Gestures (CHG) recognition technique, which extended from DTW, being capable of continuous recognition of hand gestures. They develop a prototype system for gesture recognition using a smartphone as an input smart device. The proposed system can also be used on any smart device which has the accelerometer and gyroscope sensors and interacts with a machine. The input features are signals from accelerometer and gyroscope, and they have 12 classes of gestures to be recognized. They got an average accuracy of 93.56% in their experiment which was tested by ten participants and each gesture was acquired from each participant around 100 times.

In [35], they present a novel experimental setup using accelerometer and gyroscope sensors embedded on a single board along with a distance-based pattern recognition algorithm is presented for accurately identifying basic movements for possible application in gaming using a mobile platform. They had 5 different recognized patterns and used signals from accelerometer as features, and got a accuracy of 77.4% in using kNN classifier.

In [5], it presents a hand gesture recognition system using Magic Ring (a ring-shape wear-

able device with accelerometer on it) instead of data glove or camera to translate the pre-defined hand gestures into Japanese and to accomplish remote transmission of the information/command between user groups. The hand gestures in 10 classes are defined based on simplified Japanese sign language. The KNN method is adopted to the pattern recognition according to the bimanual sensing data. The recognition rate of the gesture of the user during the experiment is also measured. It is probability that users gesture is the same as result of recognition. They practice gesture for 5 minutes, before using a system. The recognition rate was improved by 7.8% from 77.4% of using only one hand to 85.2% of using both hands.

In [33], This paper presents the development of a user-specific hand Gesture Recognition System (GRS) based on the information of a single tri-axial accelerometer to recognize 7 different dynamic gestures for natural Human Machine Interaction (HMI). The aim of this paper is to analyze and compare different computational methods for feature extraction, dimensionality reduction, and vector classification in order to select the most suitable combination of signal processing stages that meets the performance requirements for a single-chip, wearable GRS system. These requirements are lag-free response, , low size, and low power consumption while keeping high recognition accuracy. Experimental results show that the overall achievable accuracy can be up to 98% for Artificial Neural Network (ANN) and Extreme Learning Machine (ELM) predictors, and 99% for Support Vector Machines (SVM).

In [6], this paper presents the design and implementation of a system of accelerometer-based hand gesture recognition. As the recognition of hand gestures is a pattern classification problem, two techniques based on artificial neural networks are explored: multilayer perceptron and support vector machine. A gestural vocabulary of 8 types of gestures was used, which was also used by other similar works in order to compare results. The experimental data base is performed by 8 users, for 7 days and 10 repetitions per day. In the user-independent case, the variability is bounded to a particular user. An average recognition rate of 97.31% was obtained by SVM with linear kernel.

The classifying method comparison is summarized in Table 2.3. From the comparison of the pros and cons, we chose SVM classifier in our system.

Table 2.2: Literature comparison.

	Akl et al. [31]	Gupta et al. [4]	Aravind Kailas [35]	Kuroki et al. [5]	Marques et al. [33]	Ducloux et al. [6]
Same patterns	U, D, R, L, C, C-C	U, D, R, L, C, C-C	R	U, D, R, L, C	U, D, R, L, C, C-C	U, D, R, L, C, C-C
Different patterns	triangle, 90 corner, "ε", "m" "3", "w"	slanting (4 direction), "s" (2 direction)	F, B, F-B, B-F	turn R, turn D, F-B D-U-D	ON/OFF	square, 90 corner
Features	acc	acc, gyro	acc, gyro, euler	acc	gen by acc	acc
Classes	12	12	5	10	7	8
Experiment samples	7 subjects (30 times)	10 subjects (100 times)	20 subjects	2 subjects (5 minutes)	60 subjects / per gesture	8 subjects (10 times)
DTW	94.6% (ave)	93.56% (ave)	-	-	-	-
kNN	-	-	80% (ave)	77.40%	-	-
SVM	-	-	-	-	98%~99%	97.31%

* Acronym Definition:

U: UP

D: Down

R: Right

L: Left

C: Clockwise circle

C-C: Counter clockwise circle

F: Forward

B: Backward

Table 2.3: Method comparison.

	<i>DTW</i>	<i>kNN</i>	<i>SVM</i>
Features	acc & gyro	acc & gyro	acc & gyro
Accuracy	Intermediate	Low	High
Sensitivity	Intermediate	Low	High
Specificity	Low	Intermediate	High
Advantage	Less computation time	Easy to use and understand Low complexity	High accuracy Less computation time, Support high-dimensional data



Chapter 3

Proposed System and Algorithms

3.1 Overview

We proposed a system which can detect hand movement with an nine-axis sensor and machine learning methods. In this chapter, we will talk about our system composed with dimension reduction method, how we select our features and classification method. First, in section 3.2, we will introduce data structure of the system with a flow chart and discuss about why we choose the methods. The definition of movements we want to recognize and some criterion will introduce in section 3.4. In section 3.5, a thresholding method composed of threshold definition and scanning windows will be introduced. Dimension reduction and feature extraction will make the result better, we will explain it in section 3.6. Before classification, we have to build a set of feature vectors. In addition to feature extraction, we also use a window to select a set of features due to the continuous of the movement, this will be introduced in section 3.7. A classification theory and how we use it will also be presented in the last section.

3.2 Architecture of System

The system consist of the methods mentioned above. We will not add thresholding method in the system now, the reason will be shown in section 4.3.1. The flow chart of system is shown in Figure 3.1.

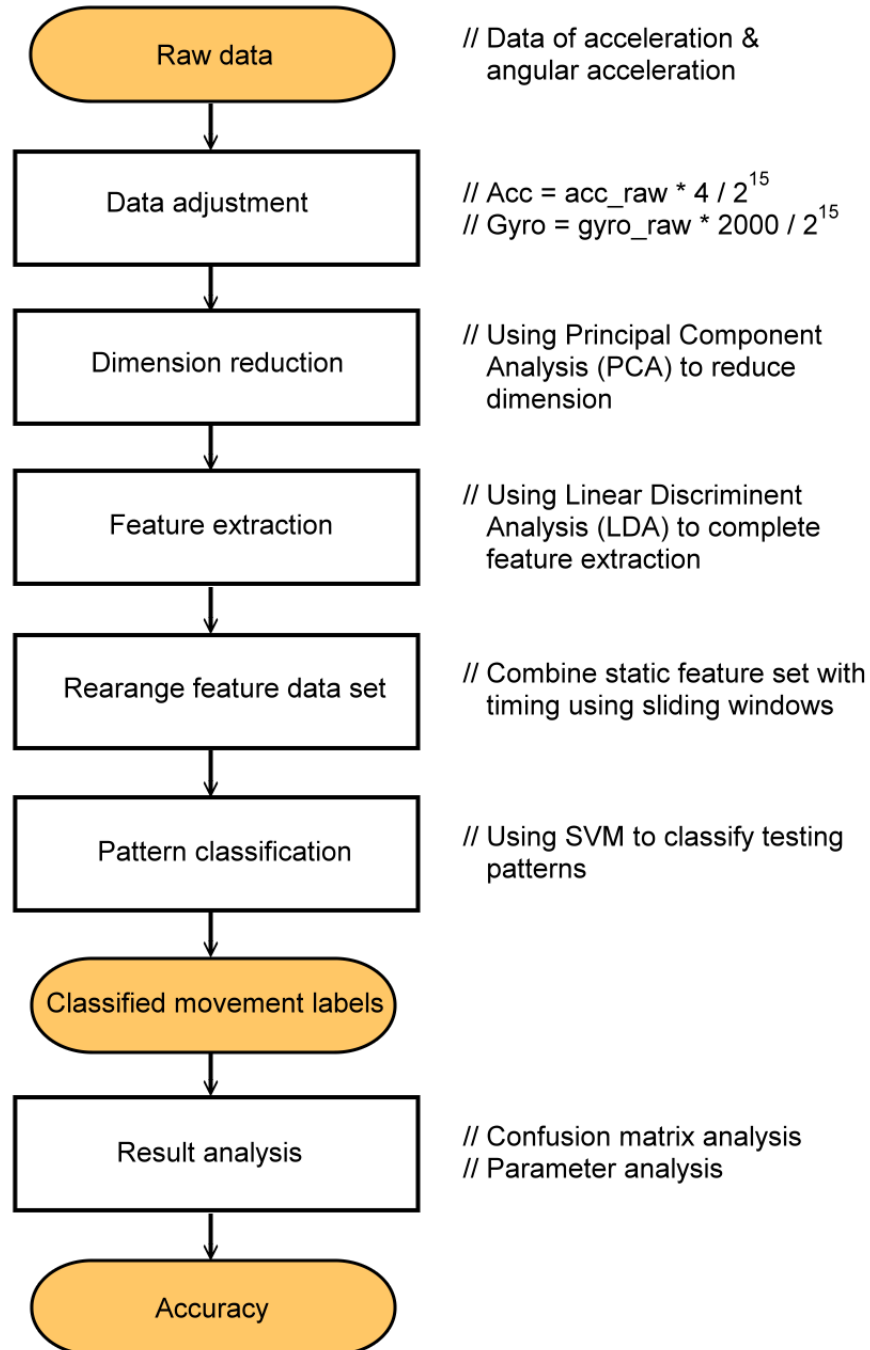


Figure 3.1: Flow chart of the gesture recognition system.

As in the flow chart, first, we will record raw data of nine-axis and only keep data of accelerometer and gyroscope (six-axis left). Reason of not using the data of magnetometer is that its data changes irregularly and has many noise. After we get raw data, we will adjust them to correct unit. The unit of acceleration is in g/s and the unit of angular acceleration is in $degree/s^2$. Next, we will do dimension reduction with method of Principal Component Analysis (PCA), this will reduce dimension from six to four to reduce training time in classification step. After dimension reduction, feature extraction will be done next. Feature extraction has the characteristics of segmenting features that is benefit for classification, and then use these features to describe or represent the shape and color of a particular object. Here we use Linear Discriminant Analysis (LDA) to extract features and we will make a detail description in section 3.6. In addition to feature extraction, we integrate scan window to build a feature set of continuous movement to rearrange the feature vectors for classification. Last, we will use Support Vector Machine (SVM) to classify the testing data. A class label is marked before each feature vector as the input of SVM. The output of classifier is a predicted label followed by a target label. We will take the predicted label to do parameter analysis and confusion matrix analysis to get the best accuracy.

3.3 Introduction to Inertial Body Sensor (Nine Axis Sensor)

In this section, we will introduce what sensor we use and its signals.

3.3.1 MPU9250

In our experiment, we use the sensor system designed by our laboratory which have ECG/Respiration sensor, EEG/EMG sensor and motion tracking sensor (nine-axis sensor) in it. Here we only use the nine-axis sensor - MPU9250. The MPU9250 is a nine-axis gesture recognition device that combines a three-axis gyroscope, three-axis accelerometer, three-axis magnetometer all in a small 3x3x1mm package [36].

Specification of MPU9250

The nine-axis sensor used in the thesis is MPU9250, and it's specification is listed in Table 3.1.

Table 3.1: Specification of MPU9250.

Gesture Recognition Sensor	
Model	MPU9250 by InvenSense
Type of Sensors	Accelerometer, Gyroscope, Magnetometer
Power Consumption	3.7 mA
Number of Channel	9
Sampling Rate	50 Hz

The block diagram is shown in Figure 3.2, and the appearance of the sensor is shown in Figure 3.3 below.

As a programmable SoC, the specification of each sensor for our application is listed in Table 3.2.

Table 3.2: Specification of each sensors in MPU-9250.

	Gyroscope	Accelerometer	Magnetometer
Full-Scale Range	± 2000 °/s	± 4 g	± 4800 μ T
ADC Word Length	16 bits	16 bits	16 bits
Sensitivity Scale Factor	0.061 (°/s)/LSB	0.12 mg/LSB	0.15 μ T/LSB

Operation

To access data of gyroscope and accelerometer, local registers are written and read. But magnetometer (AK8963) is a slave of the I2C bus, we have to access the data via I2C slave controller. The flow charts for accessing the sampled data are shown in Figure 3.4 and 3.5. There are two ways for accessing the registers, I2C and SPI. We used the SPI as the communication interface.

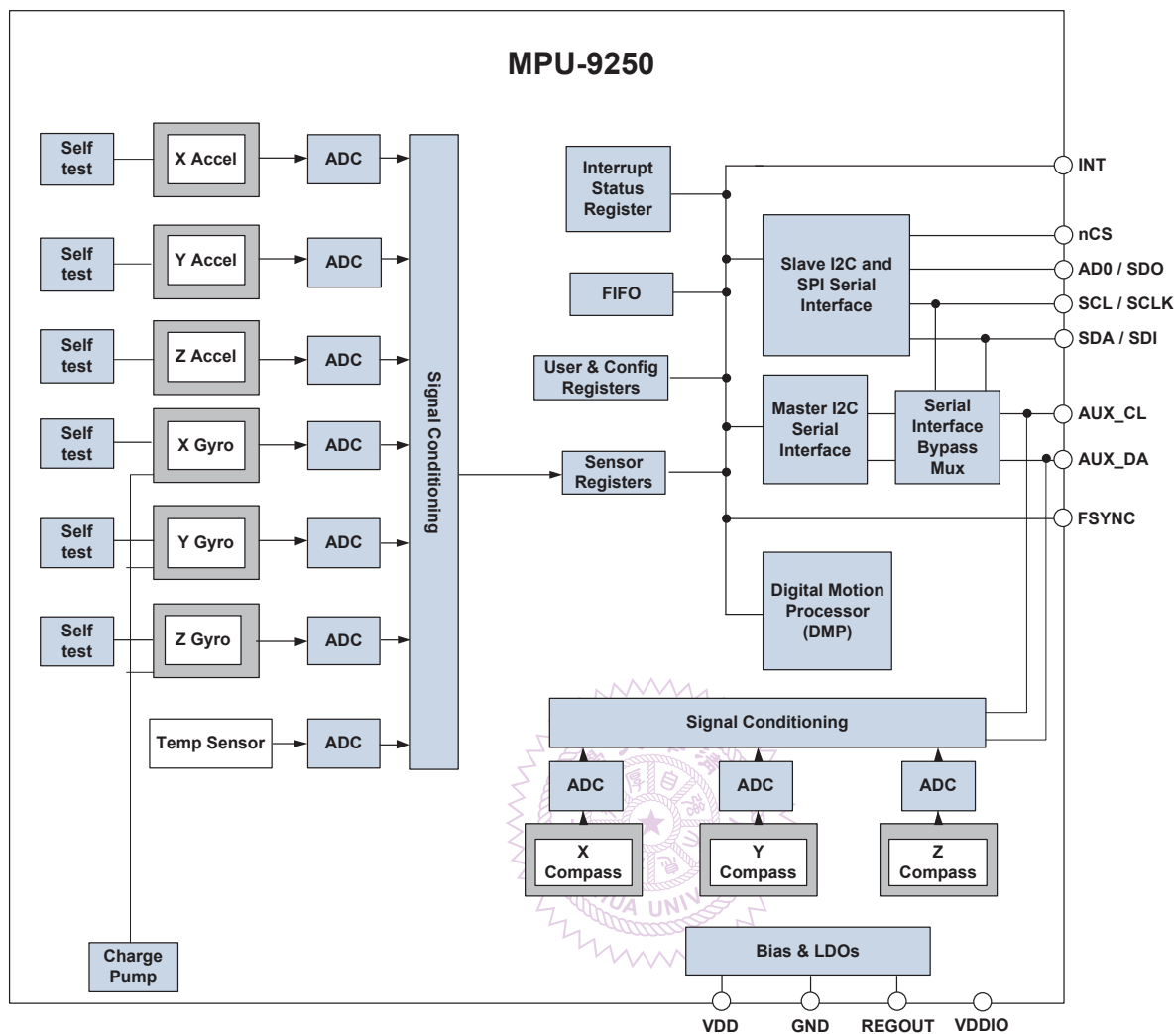


Figure 3.2: Block diagram of MPU-9250 [36].

3.3.2 Signals of Nine-Axis Sensor

Gesture recognition is a key technology in synthetic environments, robotics, and other applications that require real-time information about the motion of a human [37]. There are many ways to track one's motion. One of them is using inertial and magnetic sensors to detect the movement of the body. The sensors are accelerometer, gyroscope and magnetometer [37] [38].

Unlike sensing the physiological electrical activity on the body we have mentioned priorly, the electrical signals triggered by the microelectromechanical systems (MEMS).

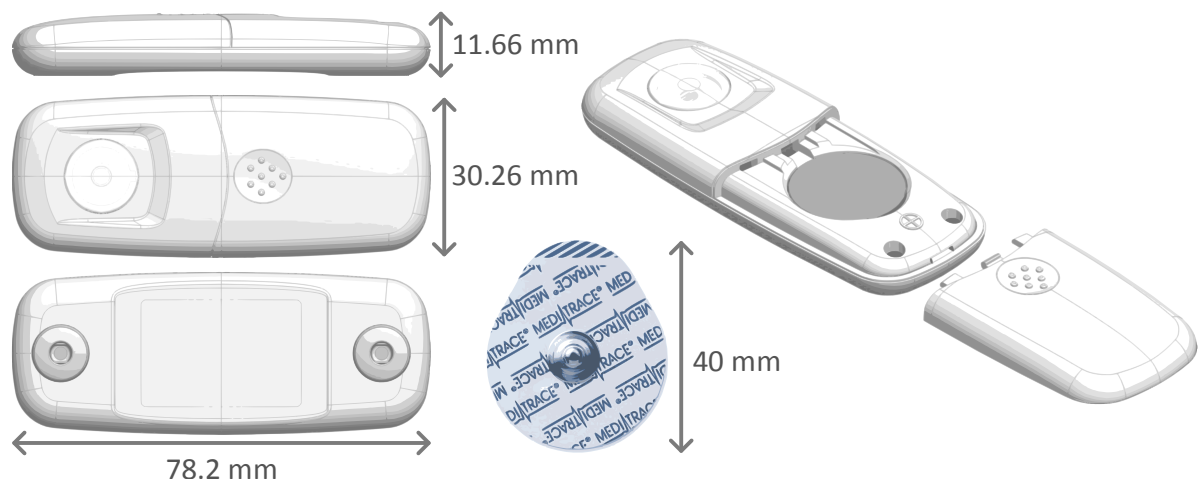


Figure 3.3: Sensory node implementation.

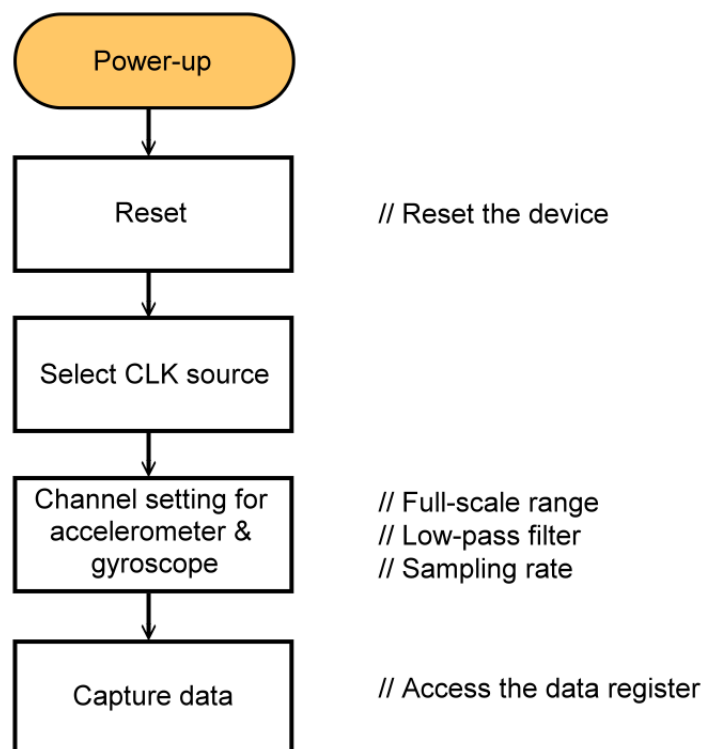


Figure 3.4: Flow chart for accessing the sampled data of gyroscope and accelerometer.

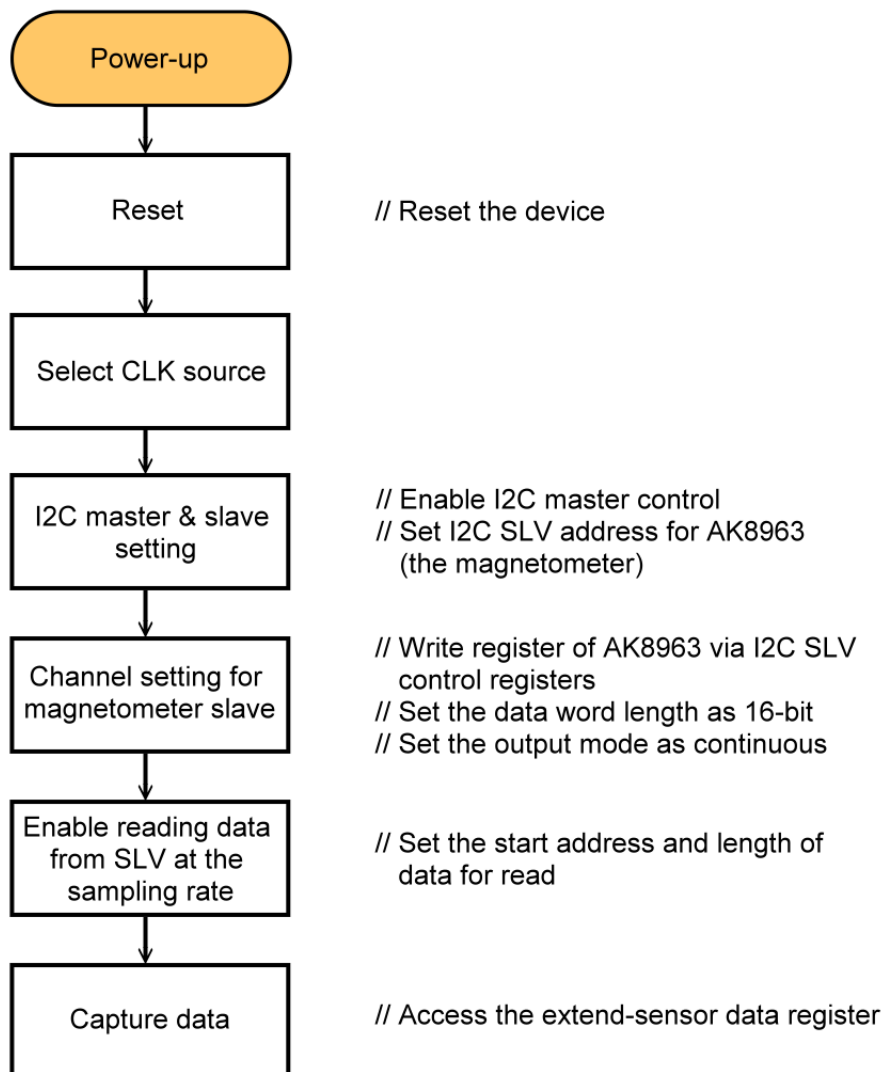


Figure 3.5: Flow chart for accessing the sampled data of magnetometer.

Accelerometer

An accelerometer is a device that measures proper acceleration ("g-force"). Proper acceleration is not the same as coordinate acceleration (rate of change of velocity). For example, an accelerometer at rest on the surface of the Earth will measure an acceleration $g = 9.81 \text{ m/s}^2$ straight upwards. By contrast, accelerometers in free fall orbiting and accelerating due to the gravity of Earth will measure zero [39].

The core element of a typical MEMS accelerometer is a moving beam structure composed of two sets of fingers: one set is fixed to a solid ground plane on a substrate; the other set

is attached to a known mass mounted on springs that can move in response to an applied acceleration. This applied acceleration (Figure 3.6) changes the capacitance between the fixed and moving beam fingers [40].

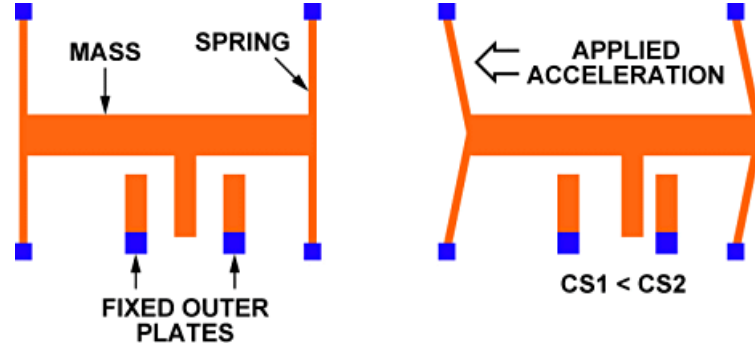


Figure 3.6: The structure of a MEMS accelerometer.

Gyroscopes

A gyroscope is a device used primarily for navigation and measurement of angular velocity. 3-axis gyroscopes detect the angular velocity of three oriental rotation.

MEMS gyroscope is implemented by vibration structure standardised as Coriolis vibratory gyroscope (CVG) which is based on Coriolis force application.

Coriolis Effect occurs when an object moves in a linear direction inside a rotating reference frame. Refer to the Figure 3.7, a mass m has velocity v_r to move straight line towards the outer edge (red arrow) on a spinning plate in angular speed of Ω . It will be affected by Coriolis Effect. One part of the effect is from the tangential velocity v_t (blue arrow) which has a magnitude of Ωr (r : radius at the point). Since

$$r = v_r t, \quad (3.1)$$

the tangential velocity

$$v_t = \Omega \times v_r t, \quad (3.2)$$

then the mass m has an acceleration which equals to $\Omega \times v_r$.

Another portion of Coriolis acceleration comes from the green acceleration vectors. It is from the force which is to maintain the mass moving toward the radius line. Then we can see

the red arrow change the direction at different time. This acceleration is equal to $\Omega \times \mathbf{v}_r$.

Adding the two acceleration, we will get $2\Omega \times \mathbf{v}_r$ which exhibit a force of $2m\Omega \times \mathbf{v}_r$ on the mass caused by rotation system. The force on the mass will create a reactionary force in the opposite direction, for a force of

$$\mathbf{F}_c = -2m\Omega \times \mathbf{v}_r. \quad (3.3)$$

This is so-called Coriolis force.

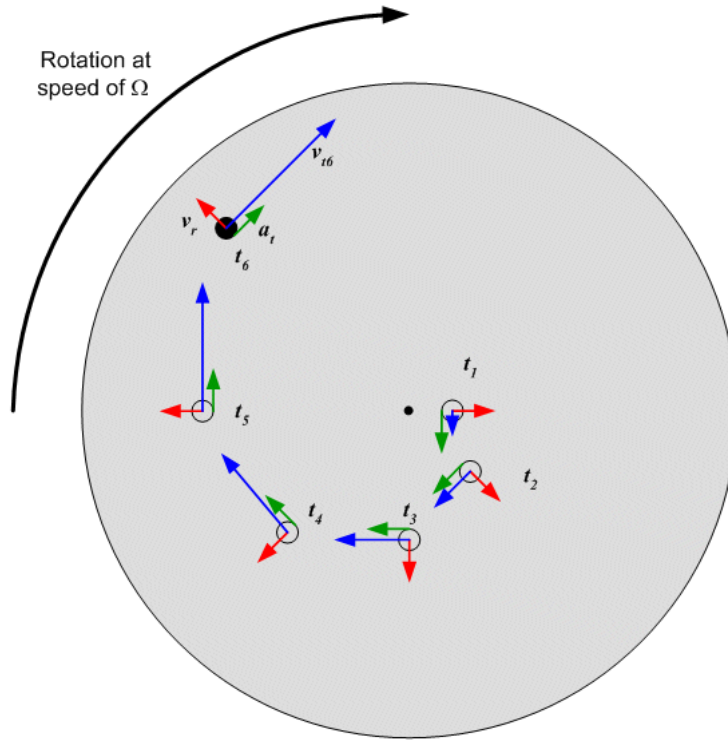


Figure 3.7: Velocity and acceleration vectors present in the Coriolis Effect [41].

A implementation of a MEMS gyroscope is shown in Figure 3.8. The tiny mass, m , will vibrate back and forth, which will cause orthogonal Coriolis forces of the same frequency [41].

Magnetometer

A magnetometer is a scientific instrument used to measure the strength and/or direction of the magnetic field. In electronic field, it also called an electronic compass. One of the approaches for MEMS magnetometers is a Hall effect sensor. The Hall effect is when electrons

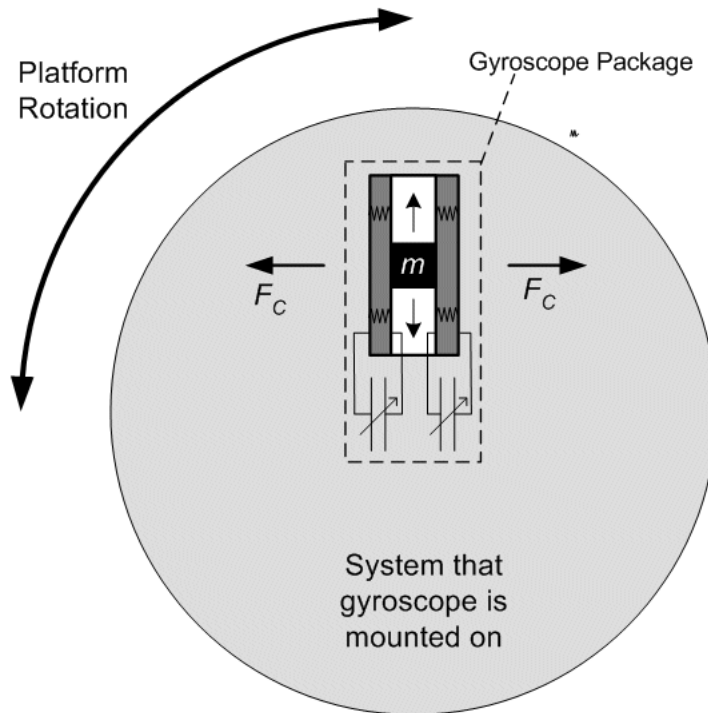


Figure 3.8: Implementation of a MEMS gyroscope.

(or holes) move in a conducting plate that is immersed in a magnetic field, they experience a Lorentz force (Figure 3.9) then bias to one side reducing a voltage called Hall voltage as shown in Figure 3.10.

3.4 Definition of Movements to be Recognized

Before starting the system, we have to define the movements we want to classify first. The range of classes for movements in most of studies is between 2 classes to 10 classes. Here we choose 8 movements described in Figure 3.11 to be classified. In the figure, a, b, c and d are the easiest movements and are the substrate of many movements. E and f are counter clockwise circle and clockwise circle respectively, which are the movements people most likely to try in many applications. G and h are wrist turning left and right which are inspired by the game of driving. From the schematic diagram, the dot is the start and end point of the movement. The direction of arrow strokes is the direction of all movements. The arrow without dot is the return movement direction.

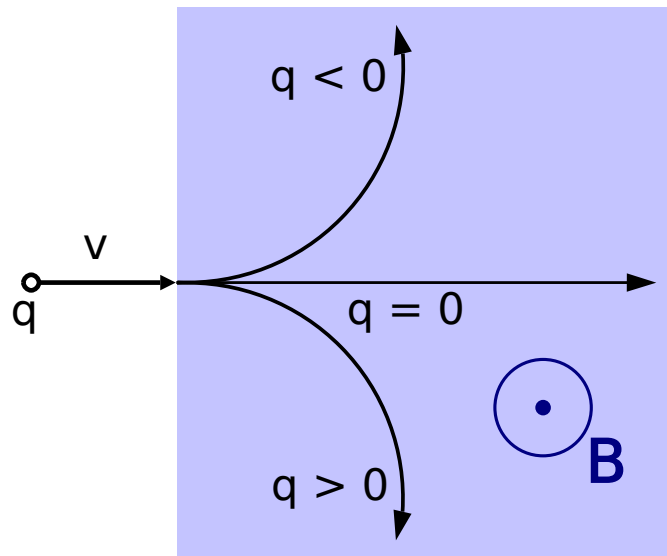


Figure 3.9: Lorentz force affect on the charge q in a velocity v under the influence of a magnetic field B [42].

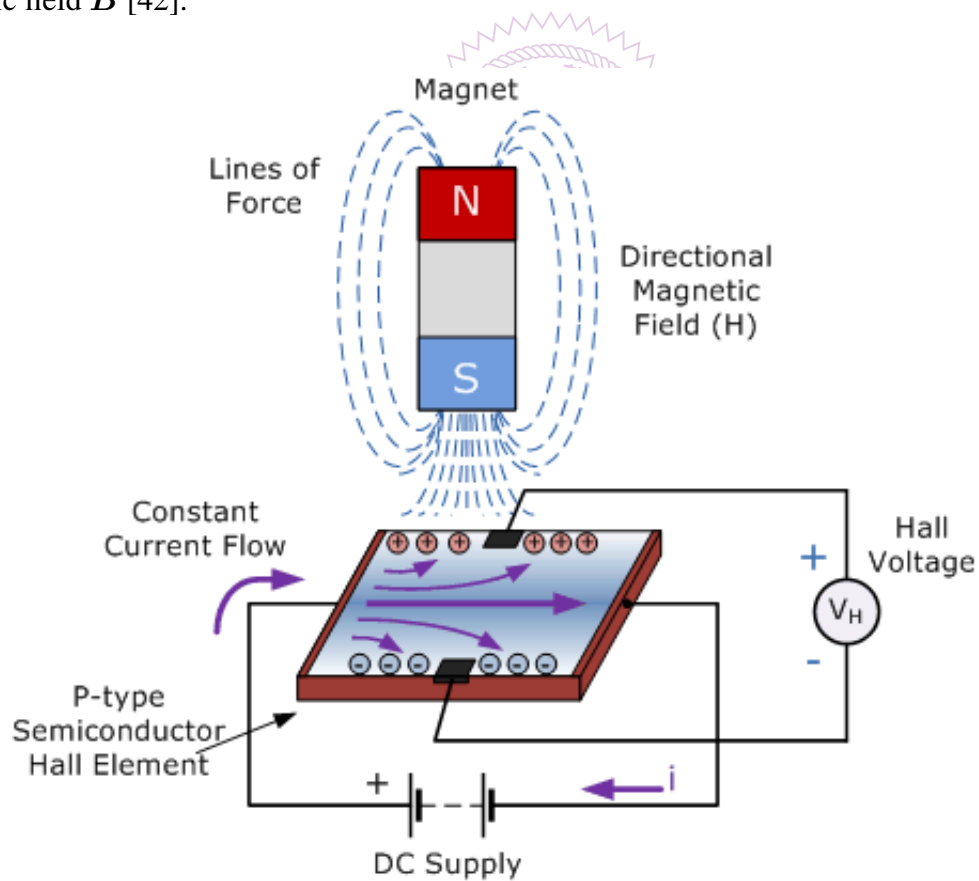


Figure 3.10: Hall effect on a semiconductor [43].

Expect the movement definition, here we list some specification of the movements for the system. Users have to follow the specification to apply this system to achieve the best performance.

- Attach the sensor on left wrist, along the direction of blood vessel.
- Initial movement calibration:
 - Put left hand straight forward with fist and as high as shoulder.
 - The wrist face downward.
- Start point of movements:
 - The start point of every movement is the initial movement of calibration.
- End point of movements:
 - The end point of movements is 45 degrees to up, down, left or right.
- Range of movements:
 - The range of acceleration is between $-2g$ to $2g$, the minimum acceleration should exceed over $0.3785g$.

3.5 Thresholding Method with Scanning Window

Thresholding is an easy method to detect movements, but relatively, it also has many limitations. We will introduce the method and explain why we do not use it. The thresholding flow chart in in Figure 3.12.

Detecting method

Here we will introduce the method step by step with the 3-axis acceleration example of "UP" movement. Step 1 to step 5 are calibration stage. Step 6 to step 7 are prediction stage.

































Basic	Acceptable variant	Basic	Acceptable variant
a. 	  	e. 	  
b. 	  	f. 	  
c. 	  	g. 	  
d. 	  	h. 	  

Figure 3.11: Movements definition. (a. Up, b. Down, c. Left, d. Right, e. Counter clockwise circle, f. Clockwise circle, g. Wrist turn right, h. Wrist turn left)

1. Calculate the maximum and minimum value for each data (each movement).

In Figure 3.13, we take acceleration in x-axes for example, the red dotted line is the maximum and the minimum value.

2. Subtract zero from the maximum (or minimum) value as an interval.
3. Divide the interval from by 2 to by 10, and take each value as threshold to detect the testing data. Figure 3.14 is an example of dividing waveform interval by 4.
4. Take Q_2 of up-interval (Choice of up-interval or down-interval is depended on waveform, here we choose up-interval for acc_x) as threshold value (we have got best result in taking Q_2 as threshold value).
5. Set window size and window-scan interval for scanning-detection.

We make a cross-validation to find the most appropriate window size/window-scan interval which ranges from 20 to 100. The rule to set window-scan interval is that only one window meets the pass condition ($\frac{\text{Number of threshold-passed samples}}{\text{window size}} > 0.65$)

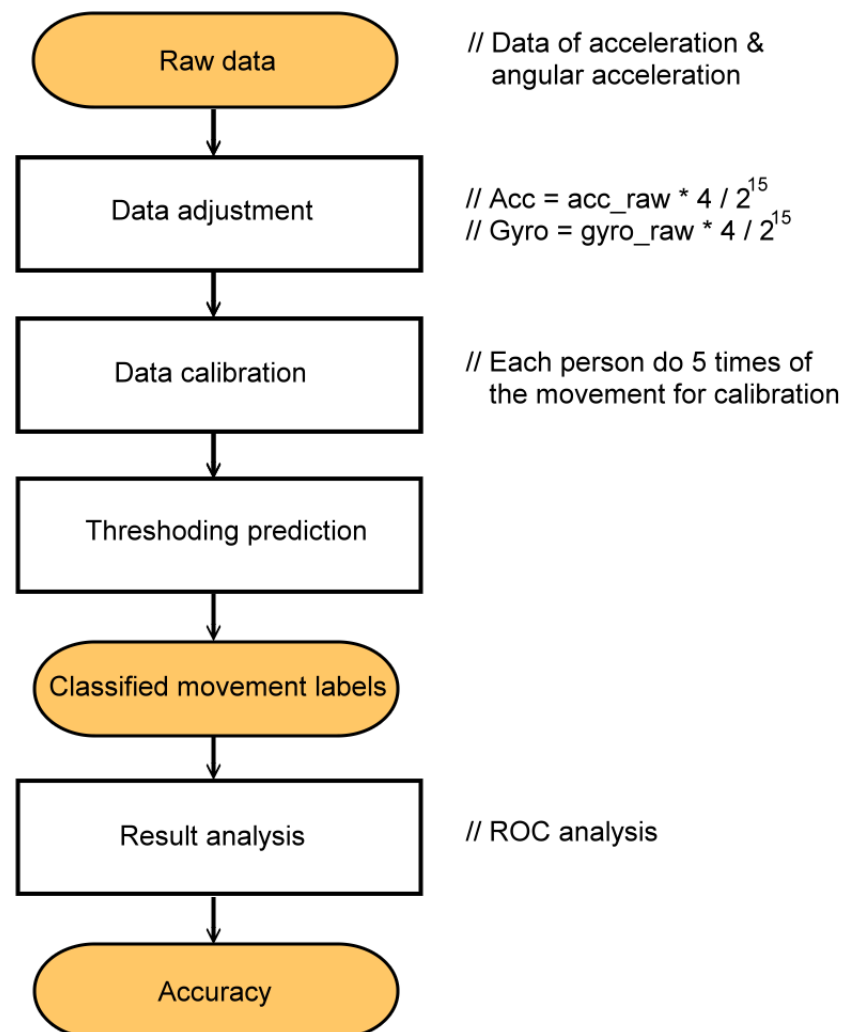


Figure 3.12: Threshold flow chart.

), and window size should be bigger than translucent area in calibration data. The set window and window-scan interval are used for detecting movements. An example is shown in Figure 3.15.

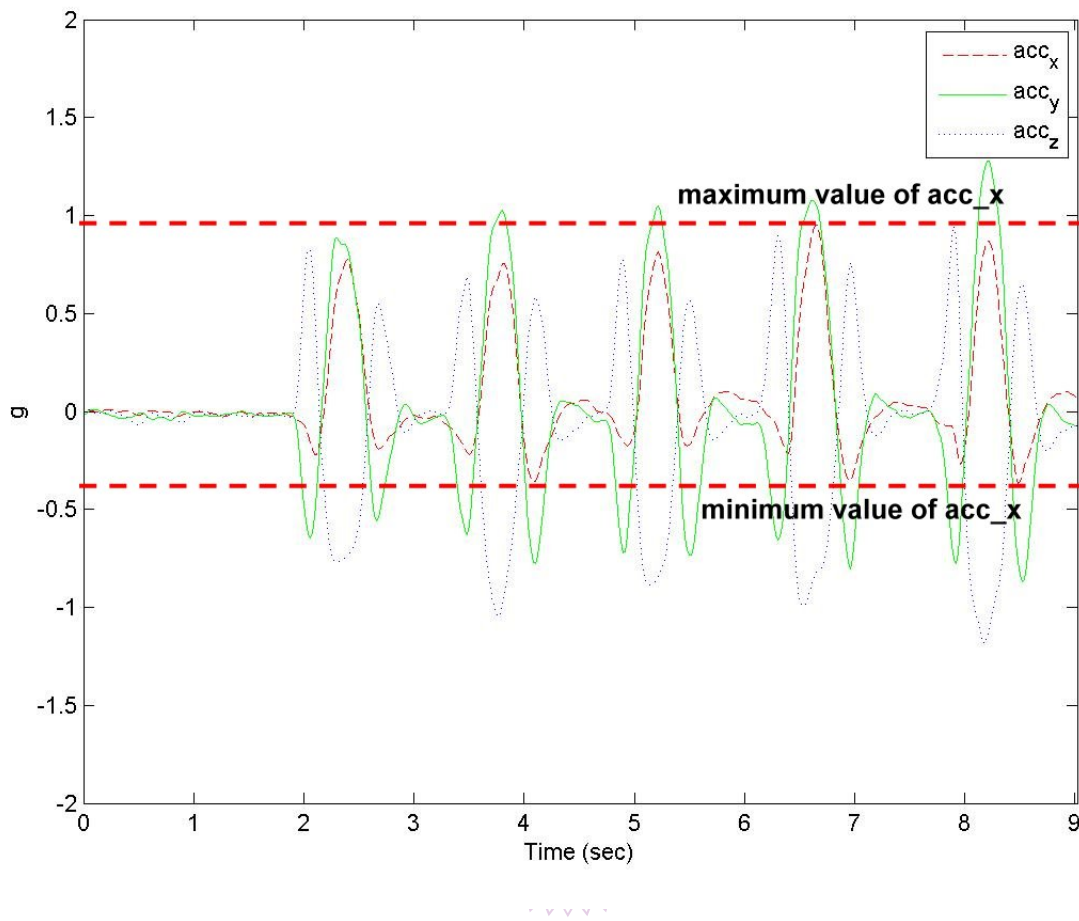


Figure 3.13: The acceleration waveform of x-axes, y-axes and z-axes with minimum and maximum value.

6. Use the defined threshold value, window size and window-scan interval to detect testing movements.
7. The data passes threshold marked as **DETECT**, in contrast, the data does not pass threshold marked as **NON**. The movement-detected conditions is shown below:

$$\frac{\text{Number of threshold-passed samples (DETECT)}}{\text{window size (NON)}} > 0.65. \quad (3.4)$$

After passing the condition, it will output a detected-movement label to let users know which movement has been done.

This is one of the algorithms of our system, the determination of threshold value and

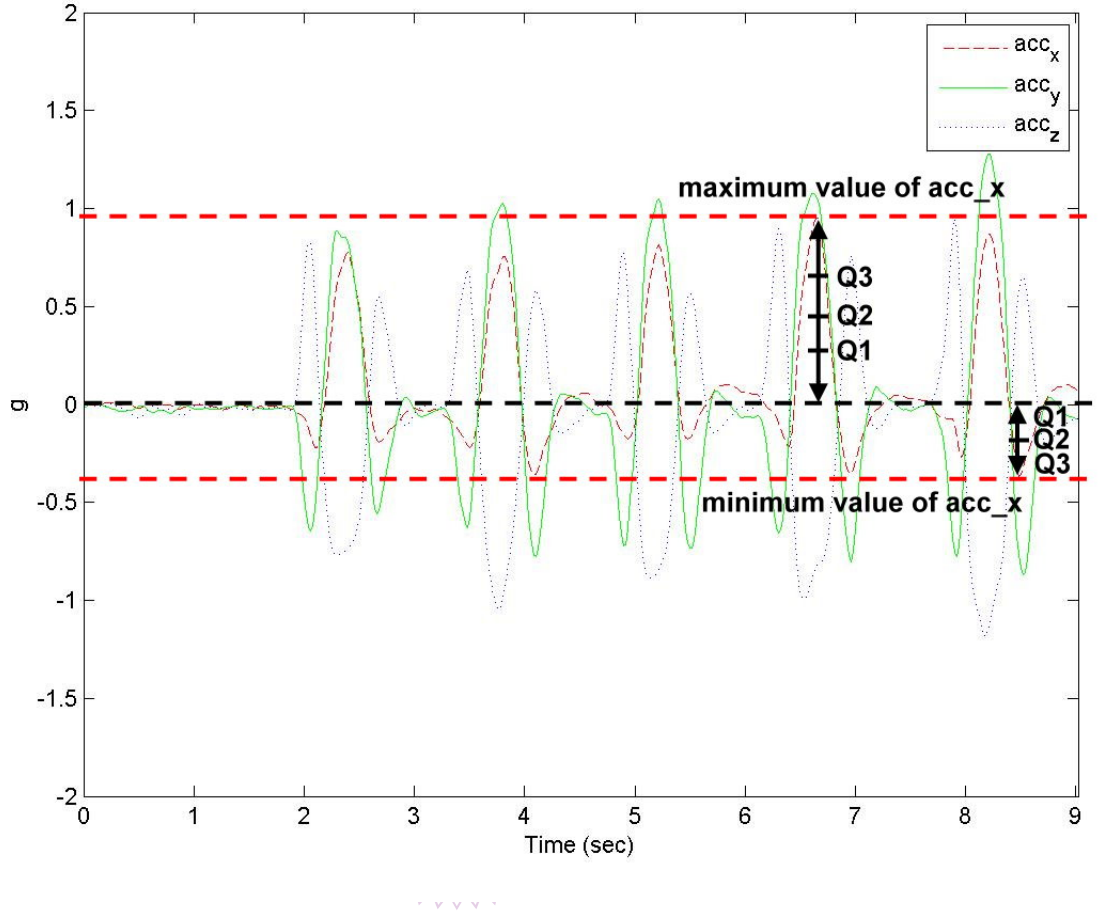


Figure 3.14: The acceleration waveform of x-axes, y-axes and z-axes with interval divided by 4. The up-interval is between 0 and the maximum value, similarly, the down-interval is between 0 and the minimum value. ($Q1 = \text{interval} * \frac{1}{4}$, $Q2 = \text{interval} * \frac{2}{4}$, $Q3 = \text{interval} * \frac{3}{4}$)

window size is independent. We defined Q2 of every signal as threshold value due to having the best performance. After decided the threshold value, we would then choose the most suitable window size and scan interval from the training set.

3.6 Dimension Reduction and Feature Extraction

In this section, we will introduce a dimension reduction method and a feature extraction method to reduce training time and get more obvious features.

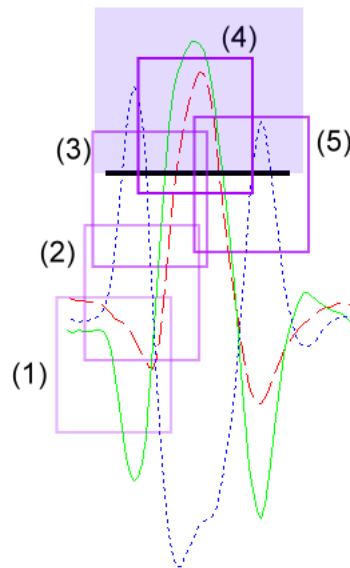


Figure 3.15: Thresholding method integrated with scanning windows. The translucent area is condition passing the threshold, and we have marked the sequence of scanning windows.

3.6.1 Dimension Reduction using Principal Component Analysis Algorithm

By definition, dimension reduction is to find a low-dimensional data set which can still describe the original data set in high dimension and does not lose any characteristics of the original data. It is the process of reducing the number of random variables under consideration [44], via obtaining a set of **uncorrelated** principal variables. In our system, we use PCA to reduce dimension of data. PCA is an analyzing and simplifying data collection methods, commonly used to reduce dimensions of data set, while keeping characteristics with the greatest contribution to the variance. PCA does eigen decomposition of covariance matrix to get the main components (eigen vectors) and their weights (eigen values). Then, we will get the optimal low-dimensional data by removing the components which correspond to the smallest eigen values which will lose minimal information. The important characteristics of PCA is list below [45]:

- Representative

- Retain the original variable information.
- Independence
 - No overlap between the main component.
- Compactness
 - Few components take place of the original numerous variable.

Basic Theory

PCA is an orthogonal linear transformation which converts the data to a new coordinate system. In the new data set, data with the largest variance is projected to the first axis (called the first principal component), data with the second largest variance is projected to the second axis (the second component), and so on. It represents the largest component (variable) in original data with minimum variable information.

- Principles:
 - New variables are linear combination of original variables.
 - Retain the maximum amount of variation between the original variables.

Below are 7 steps for performing PCA. Here we use 6 dimensions (3-axis acceleration and 3-axis angular acceleration) for example:

1. Taking the whole dataset ignoring the class labels

Suppose each class has $\frac{M}{2}$ samples of data, and because we do not need class labels for the PCA analysis, we merge the samples for 2 classes into one $6 \times M$ matrix. Below is the form of our dataset:

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{1,5} & x_{1,6} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & x_{2,5} & x_{2,6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{M,1} & x_{M,2} & x_{M,3} & x_{M,4} & x_{M,5} & x_{M,6} \end{pmatrix}_{M \times N}, N = 6. \quad (3.5)$$

2. Computing the 6-dimensional mean vector and standard deviation

The mean vector is composed of the mean of each column. The matrix form is shown below:

$$\boldsymbol{\mu} = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{pmatrix}^T, \quad (3.6)$$

where

$$m_{N,1} = \frac{(x_{1,N} + x_{2,N} + x_{3,N} + \cdots + x_{M,N})}{M}, N = 1, 2, 3, 4, 5, 6, \quad (3.7)$$

The standard deviation vector is formed as:

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix}^T, \quad (3.8)$$

and it is computed by the following equation:

$$\sigma_N = \sqrt{\frac{1}{M} [(x_{1,N} - m_N)^2 + (x_{2,N} - m_N)^2 + \cdots + (x_{M,N} - m_N)^2]}, \quad (3.9)$$

where $N = 1, 2, 3, 4, 5, 6$ and m_N is the mean value of each column.

3. Standardize the dataset matrix

After getting the mean vector and standard deviation vector, we can standardize our dataset. First, we can get a mean-centered data matrix C by subtracting mean vector from the dataset:

$$C = X - \mu. \quad (3.10)$$

Then, the standardized matrix S is as follow:

$$S = \frac{C}{\sigma}. \quad (3.11)$$

4. Compute the covariance matrix using the standardized matrix

$$\text{Cov} = \frac{S^T S}{M - 1}. \quad (3.12)$$

5. Calculate the eigenvalues and eigenvectors for the covariance matrix

$$\Sigma v = \lambda v, \quad (3.13)$$

where

Σ = Covariance matrix

v = Eigen vector

λ = Eigen value

6. Sorting the eigenvectors by decreasing eigenvalues and choosing dimension 4 as the output dimension.

In general, there are 3 ways to choose how many dimensions we want to keep. The first one is to keep the eigenvector whose related eigenvalue is greater than 1. This method is simple and convenient, but it is easy to overestimate or underestimate, the error is also bigger than the others. For example, there are 2 eigenvalues, one is 1.02 and the other is 0.98. The first component will be kept but the second will be abandoned, but the fact is that the difference between two components is negligible. The second method is deciding by steep diagram. It paints the eigenvalues in the y-axis and the

number of elements in x-axis, then abandoned the components which is beyond right side of the turning point. For example, in Figure 3.16, point A is the turning point of the eigenvalues, so we will abandon the point beyond right side of A.

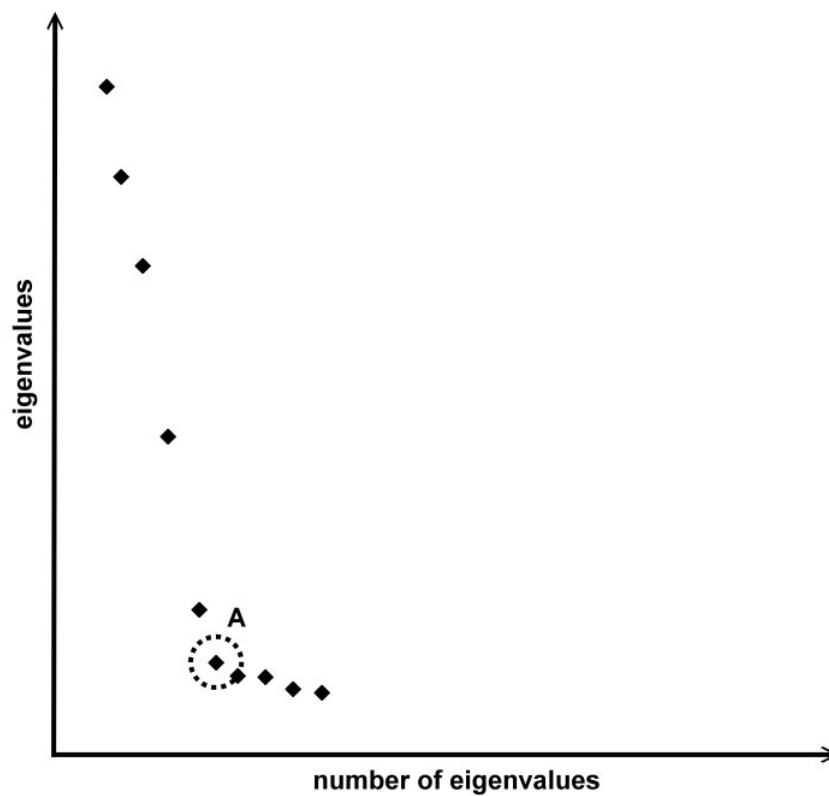


Figure 3.16: Steep diagram.

The third method is choosing the eigenvectors with cumulative proportion of related eigenvalues being more than 85%.

$$\frac{\sum_{j=1}^m \lambda_j}{\sum_{j=1}^p \lambda_j} \geq 0.85. \quad (3.14)$$

where m = number of components we want to keep, p = total number of dimensions

In our system, we use the method similar to the third method. In the experiment, we can get dimension-4 or dimension-5 depending on the property of data, but we set 4 as our stationary dimension because that we got dimension as 4 over 95% when the number of data increase. As the number of data increase, the occurrence probability of dimension-4 is getting higher. Also, in order to meet the condition of classifier, the dimension of each data must be the same. From above, that is the reason we chose 4 as the dimension after original data being reduced.

7. Transforming the samples onto the new subspace

$$\mathbf{Y} = \mathbf{S} * \mathbf{v}_{new}, \quad (3.15)$$

where \mathbf{Y} is dataset with new dimension, and \mathbf{v}_{new} is new eigenvectors after computing principal components analysis.

Advantages

- Eliminate related impacts of assessments metrics. Because the PCA analysis transforms the raw data into principal components which are independent of each other. The higher the correlation between indicators, the better performance PCA analysis will get.
- In analyzing the problem, we can abandon a part of the components, just take the components with greater variance to represent the original dataset, thereby it can reduce the computational effort.

3.6.2 Feature Extraction using Linear Discriminant Analysis Algorithm

Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning

and generalization steps. The purpose of feature extraction is allowing us to take advantage of these features to discriminate the type of object. Feature extraction involves reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power, also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. We use LDA as a feature extraction method, the theory will be described in detail below.

Basic Theory [46]

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order to avoid overfitting and also reduce computational costs.

Ronald A. Fisher formulated the Linear Discriminant in 1936 (The Use of Multiple Measurements in Taxonomic Problems) [47], and it also has some practical uses as classifier. The original Linear discriminant was described for a 2-class problem, and it was then later generalized as Multiple Discriminant Analysis by C. R. Rao in 1948 (The utilization of multiple measurements in problems of biological classification) [48].

The general LDA approach is very similar to PCA, but in addition to finding the component axes that maximize the variance of our data (PCA), we are additionally interested in the axes that maximize the separation between multiple classes (LDA). Here we are going to introduce the steps for LDA, and we take 4-dimensions, 50 samples for each 3 classes (total 150 samples) as example:

1. Reading in the dataset

The form of dataset and labels are as follow:

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ \vdots & \vdots & \vdots & \vdots \\ x_{150,1} & x_{150,2} & x_{150,3} & x_{150,4} \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 3 \end{pmatrix}. \quad (3.16)$$

2. Computing the d-dimensional mean vectors

In this step, we will compute the mean vectors \mathbf{m}_i , ($i = 1, 2, 3$) of the 3 different classes:

$$\mathbf{m}_i = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix}^T, i = 1, 2, 3, \quad (3.17)$$

where $\mu_j, j = 1, 2, 3, 4$, is the mean value of each column.

3. Computing the Scatter Matrices

Here we will compute the two 4x4-dimensional matrices: The within-class and the between-class scatter matrix.

Within-class scatter matrix \mathbf{S}_W :

$$\mathbf{S}_W = \sum_{i=1}^c \mathbf{S}_i, \quad (3.18)$$

where

$$\mathbf{S}_i = \sum_{\mathbf{x} \in D_i}^n (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T, \text{ is the scatter matrix for every class}$$

and \mathbf{m}_i is the mean vector:

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i}^n \mathbf{x}_k. \quad (3.19)$$

Between-class scatter matrix \mathbf{S}_B :

$$\mathbf{S}_B = \sum_{i=1}^c N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T, \quad (3.20)$$

where

\mathbf{m} is the overall mean, and \mathbf{m}_i and N_i are the sample mean and sizes of the respective classes.

4. Solving the generalized eigenvalue problem for the matrix $\mathbf{S}_W^{-1} \mathbf{S}_B$

In this step, we will solve the generalized eigenvalue for the matrix $\mathbf{S}_W^{-1} \mathbf{S}_B$ to obtain the linear discriminants.

$$\mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{v} = \lambda \mathbf{v}, \quad (3.21)$$

where

\mathbf{v} = Eigen vector
 λ = Eigen value

5. Selecting linear discriminants for the new feature subspace

Here we sort the eigenvectors by decreasing eigenvalues and choosing the eigenvectors with cumulative proportion of related eigenvalues being more than 85% like the step in PCA, and we will get a matrix \mathbf{W} composed of chosen eigenvectors with dimension 3.

6. Transforming the samples onto the new subspace

In the last step, we use the 4x3-dimensional matrix \mathbf{W} (4 is the dimension of the data after computing PCA, 3 is the new dimension number we got in previous step) that we just computed to transform our samples onto the new subspace via the equation

$$\mathbf{Y} = \mathbf{X} * \mathbf{W}, \quad (3.22)$$

where \mathbf{X} is the original dataset, and \mathbf{Y} is the transformed dataset with dimension k in the new subspace.

Advantages

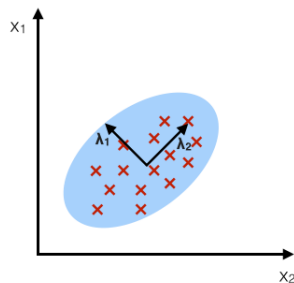
- It is a supervised dimension reduction method with labels. Comparing to PCA, it is more clearly reflecting differences between samples.

3.6.3 Comparison of PCA and LDA Algorithm

PCA and LDA is quite similar to each other, the most significant difference is that LDA has the ability to separate features but PCA cannot. Both LDA and PCA are linear transformation techniques: LDA is a supervised whereas PCA is unsupervised. PCA ignores class labels. The purpose of two method is that PCA finds the axes with maximum variance for the whole data set where LDA tries to find the axes for best class separability. In practice, often a LDA and a PCA is used together to do feature extraction.

PCA:

component axes that maximize the variance



LDA:

maximizing the component axes for class-separation

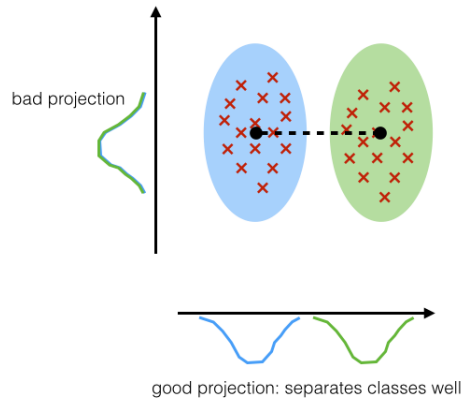


Figure 3.17: Comparison of PCA and LDA [46].

3.7 Data Classification

In this section, we will introduce classification theory in machine learning field and what classifier we used in our system. The experiment using this classifier and result will be discussed in Chapter 4.

3.7.1 Theory

In machine learning and statistics, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is known.

Theoretical results in machine learning mainly deal with a type of inductive learning called supervised learning. In supervised learning, an algorithm is given samples that are labeled in some useful way. For example, the samples might be descriptions of mushrooms, and the labels could be whether or not the mushrooms are edible. The algorithm takes these previously labeled samples and uses them to induce a classifier. This classifier is a function that assigns labels to samples including the samples that have never been previously seen by the algorithm. The goal of the supervised learning algorithm is to optimize some measure of performance such as minimizing the number of mistakes made on new samples.

The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science. Because training sets are finite and the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. The biasvariance decomposition is one way to quantify generalization error.

For the best performance in the context of generalization, the complexity of the hypothesis should match the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model has underfit the data. If the complexity of the model is increased in response, then the training error decreases. But if the hypothesis is too complex, then the model is subject to overfitting and generalization will be poorer [49].

In addition to performance bounds, computational learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

3.7.2 Feature Vectors for Support Vector Machine

Before doing classification, we have to rearrange the feature-extracted dataset into the format which is suitable for classification. The pattern of movements is related to timing, only one sample cannot represent a movement, so we have to use a set of samples to represent our time-variant feature for each movement. The sample rate of the sensor is 50Hz, so we open a window with size 50 and scan the dataset with interval 1. An example of 3-dimension (dataset after feature extraction) is shown in Figure 3.18. In this example, the window starts from the first sample, scans the dataset with interval = 1, and has total 5 windows for the dataset.

After having scanning windows, we unfold each window to arrange it into a feature vector with dimension = 150 (3-dimension x 50-samples) and column = 1. Then, mark each feature vector with its class label in first column of the vector (An example of the feature vectors is formed below).

$$\begin{array}{cccccccc}
 1 & 1 : v_{1,1}^1 & 2 : v_{1,2}^1 & 3 : v_{1,3}^1 & 4 : v_{1,4}^1 & \cdots & 149 : v_{1,149}^1 & 150 : v_{1,150}^1 \\
 1 & 1 : v_{2,1}^1 & 2 : v_{2,2}^1 & 3 : v_{2,3}^1 & 4 : v_{2,4}^1 & \cdots & 149 : v_{2,149}^1 & 150 : v_{2,150}^1 \\
 1 & 1 : v_{3,1}^1 & 2 : v_{3,2}^1 & 3 : v_{3,3}^1 & 4 : v_{3,4}^1 & \cdots & 149 : v_{3,149}^1 & 150 : v_{3,150}^1 \\
 & & & & & & & \vdots \\
 2 & 1 : v_{1,1}^2 & 2 : v_{1,2}^2 & 3 : v_{1,3}^2 & 4 : v_{1,4}^2 & \cdots & 149 : v_{1,149}^2 & 150 : v_{1,150}^2 \\
 2 & 1 : v_{2,1}^2 & 2 : v_{2,2}^2 & 3 : v_{2,3}^2 & 4 : v_{2,4}^2 & \cdots & 149 : v_{2,149}^2 & 150 : v_{2,150}^2 \\
 & & & & & & & \vdots \\
 & & & & & & & \vdots \\
 9 & 1 : v_{1,1}^9 & 2 : v_{1,2}^9 & 3 : v_{1,3}^9 & 4 : v_{1,4}^9 & \cdots & 149 : v_{1,149}^9 & 150 : v_{1,150}^9 \\
 & & & & & & & \vdots
 \end{array}$$

No matter training data or testing data, both of them need to arrange in the form mentioned above to fit the classifier input format. In next subsection, we will introduce the classification theory and SVM classifier.

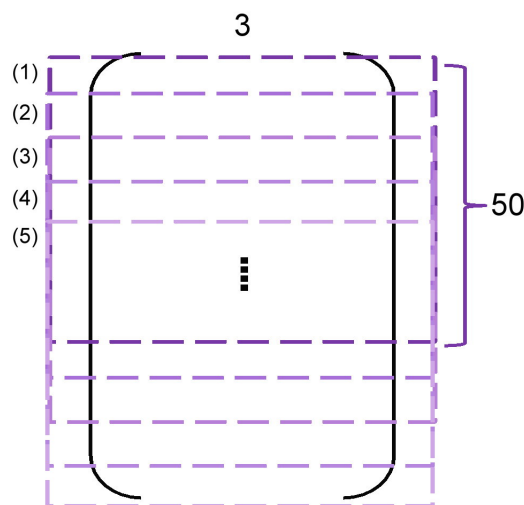


Figure 3.18: A dataset with scanning window which window size is 50.

3.7.3 System Classification using SVM Classifier

The theory of SVM classifier has been introduced in section 2.2. Here we use a library from Chih-Chung Chang and Chih-Jen Lin [50]. LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification. There are also providing functions of adjusting parameters to let users get a more suitable model to their experiment.

In our system, we use SVM with radial basis function kernel (RBF kernel), a non-linear SVM called Gaussian SVM. The purpose of SVM is to find a hyperplane to separate the two or more different sets. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin the lower the generalization error of the classifier. For the non-linear problem, it use a kernel function mapping the original high-dimensional space into a higher-dimensional space to make the separation much more easier. In SVM, a RBF kernel function is selected to suit the problem by easily computing dot products in terms of the variables in the original space. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot

product with a vector in that space is constant. Figure 3.19 is a schematic diagram for kernel machine.

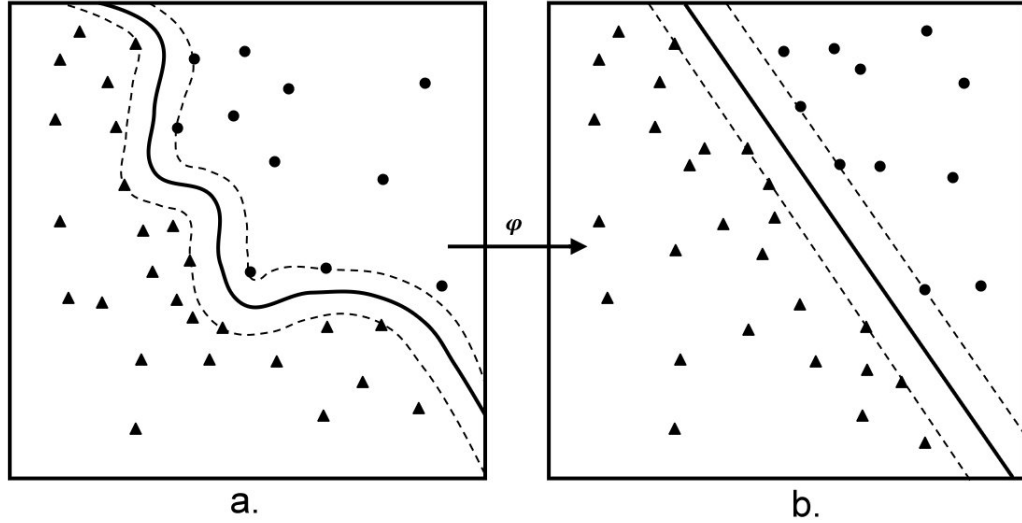


Figure 3.19: Kernel machine mapping the original data, which cannot be separated linearly, into a higher-dimensional space to find a hyperplane. (a. High-dimensional data b. Higher-dimensional data)

Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the "kernel trick". The kernel function of Gaussian SVM is:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$

γ is the parameter we have to carefully select, too big or too small will appear overfit situation in classification. Below Figure 3.20 is an example of schematic diagrams in different value of γ . In the figure, we can see that if the γ is too large (like in c.), it causes overfitting.

Another parameter we have to analyse is c . There may be some noise in the dataset, it will affect the performance of classification, so we may like to give up on some noisy samples. This

happens to the analyzing of c . c means cost, a trade-off of large margin and noise tolerance (margin violation). Figure 3.21 is a schematic diagram of margin size and margin violation.

- Value of c :
 - Large c : want less margin violation
 - Small c : want large margin

The range of γ and c in most studies is between 2^{-10} to 2^{10} . In next chapter, we will analyze both γ and c from 2^{-5} to 2^5 to get a better accuracy of classification.

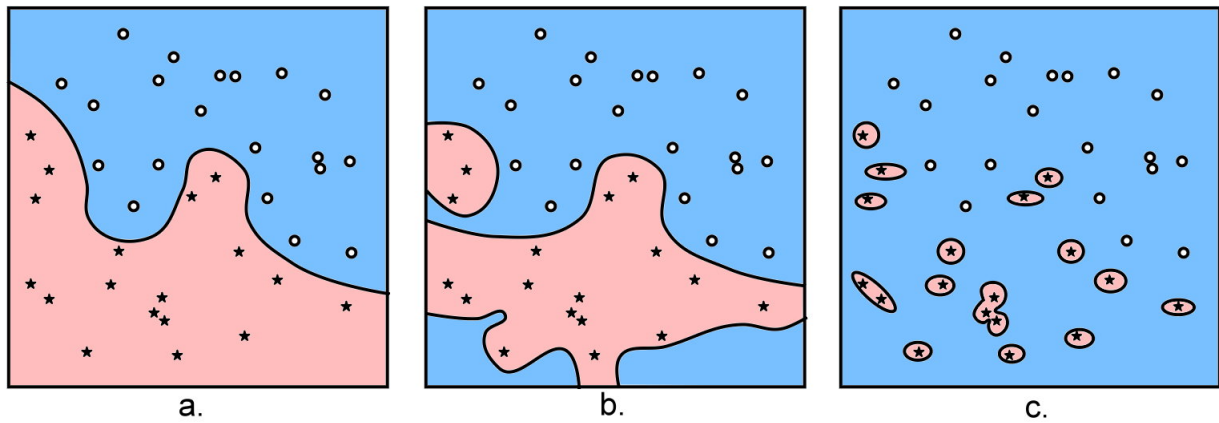


Figure 3.20: Different situation in different γ . a. $\gamma = 1$, b. $\gamma = 10$, c. $\gamma = 100$

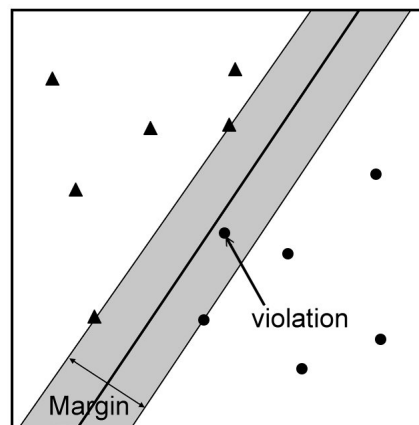


Figure 3.21: A schematic diagram of margin violation.

Chapter 4

Implementation Results

4.1 Experiment

4.1.1 Experimental Environments

To verify and analyze our system, we will use some device and tools. They will be listed below.

Table 4.1: Experimental environments.

Experiment item	Environment
Data receiving	Laboratory sensor (MPU9250)
SVM analyzing	LIBSVM [50]
Analysis of result	C++ (in Linux)

Placement of Wearable 9-axis Sensor

To receive data of every movement, first we attach the sensor on the left wrist, along the direction of blood vessel. A schematic diagram of placement and three-axis direction is shown in Figure 4.1.

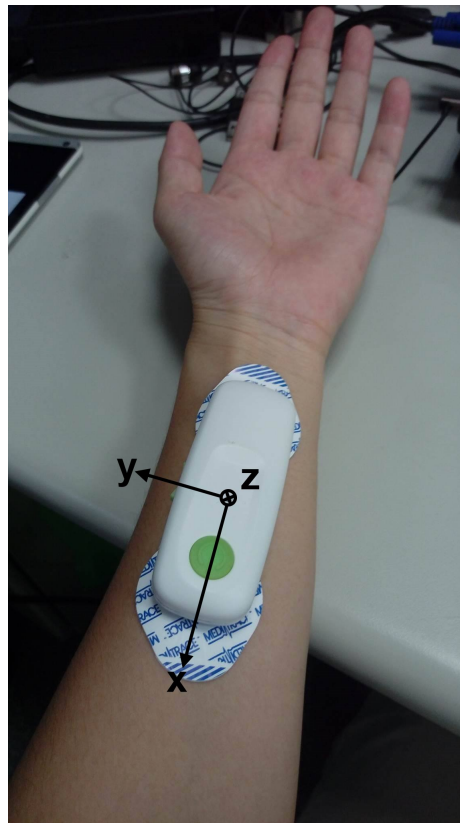


Figure 4.1: Directions and placement of the sensor.

Rules of Experiment

The rules is list below, and a schematic diagram is shown in Figure 4.2.

- Initial movement calibration:
 - Put left hand straight forward with fist and as high as shoulder.
 - The wrist face downward.
- The start point of every movement is the initial movement of calibration.
- The end point of movements is about 45 degree to the up and 45 degree to the left.
- The range of acceleration is between -2g to 2g, the minimum acceleration should exceed over 0.3785g.

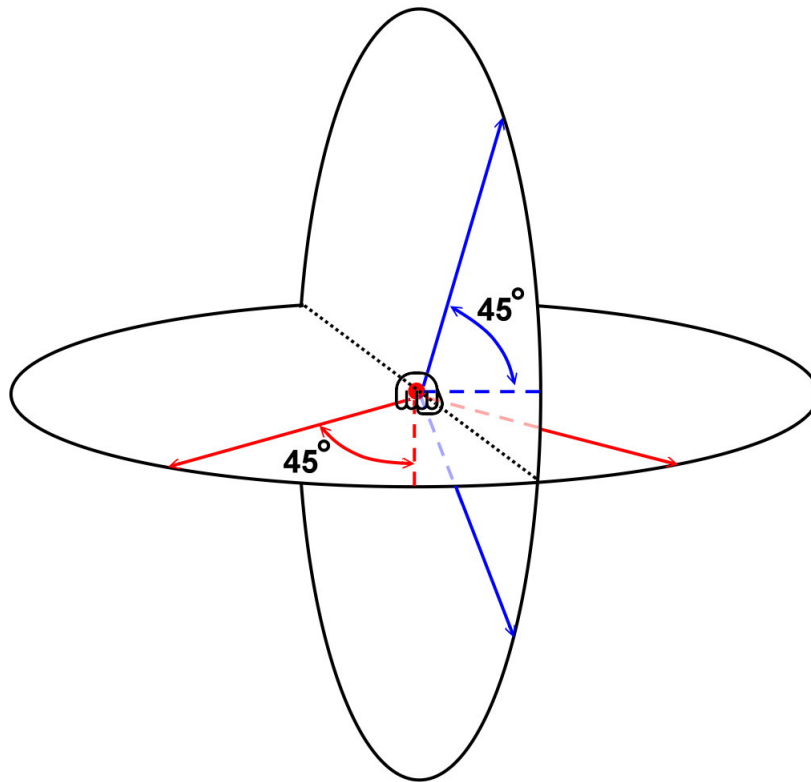


Figure 4.2: Schematic diagram of the range of movement.

4.1.2 Experiment Procedure

- Attach the sensor on left wrist, along the direction of blood vessel.
- 20 subjects data as experiment objects.
- Each person do 5 times per gesture.
- Execute feature extraction to generate feature vectors for classification.
 - Feature dimension: 150
 - Training samples: over 75000
 - Testing samples: 4100
- Use cross-validation to find best parameters (γ & C) for SVM training.
 - Split training data into n groups, each group is in equal size.

- Do cross-validation of the n groups to avoid over-fitting in SVM.
- Increase the 2 parameters (γ , C) exponentially. For example: $C = 2^{-10}, C = 2^{-9}, \dots, C = 2^{10}$.
- Find the best combination of γ and C .
- Use LIBSVM to classify.
- Analyzing accuracy of the output label of LIBSVM.

4.2 Analysis of Received 9-Axis Signals

Before implement our system, we analyze the received 9-axis signals first. The MPU-9250 contains three types of 3-axis sensors including accelerometer, gyroscope and magnetometer. Figure 4.3 shows the orientation of axes of each sensor. We observe the alteration of signals on each axis by certain movements.

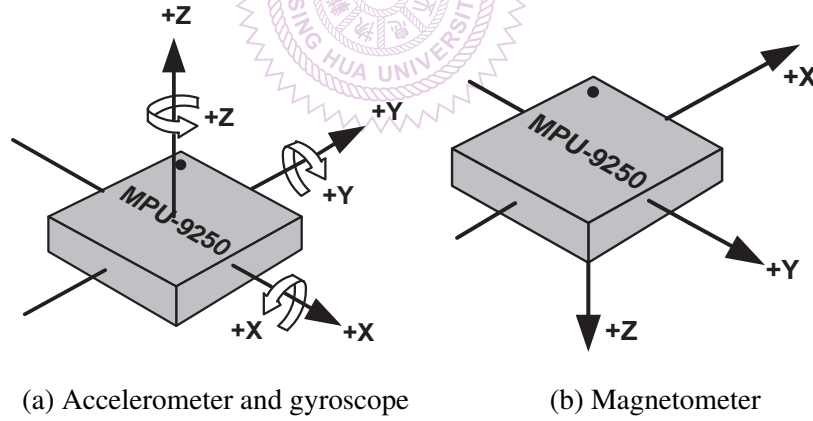


Figure 4.3: Orientation of axes of sensitivity and polarity of rotation for the nine-axis sensor [36].

Accelerometer

Accelerometer measures the g-force of object. We sway the sensor among each axis by the sequence of $x \rightarrow y \rightarrow z$. The result is shown in Figure 4.4. The orientations of top to bottom ones are x, y and z respectively.

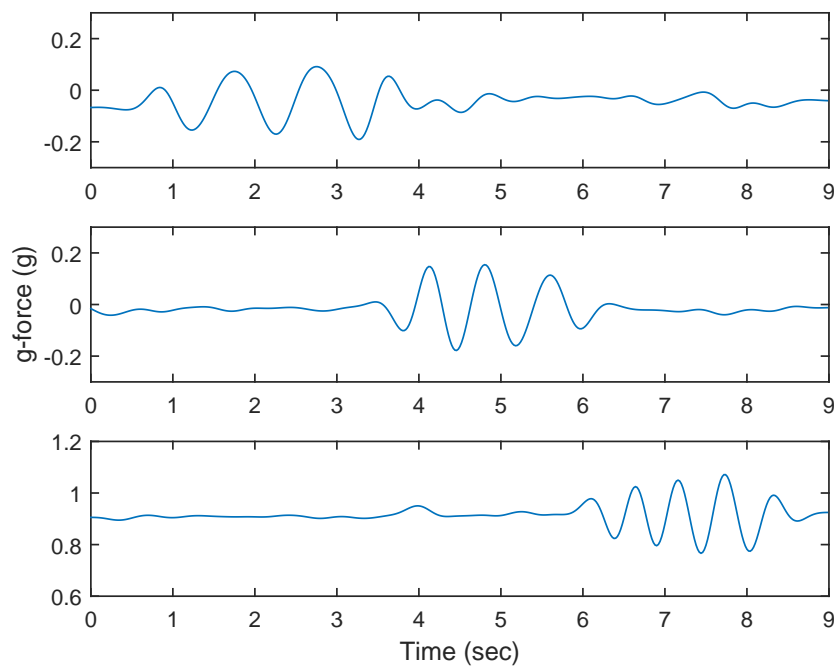


Figure 4.4: The accelerometer data generated by certain movement sequentially.

Gyroscope

Gyroscope measures the angular velocity. We rotate the sensor by the normal vector as the orientation of y, then x and z axis. The result is shown in Figure 4.5. The orientations of top to bottom ones are x, y and z respectively.

Magnetometer

Magnetometer is affected by the magnetic field. We rotate the sensor in clockwise with a certain angle, then rotate in counterclockwise back to the initial state. Under the affect of geomagnetic, the amplitude of each axis alters sequentially then back to the initial state. The result is shown in Figure 4.6. The imbalance of the z-axis data is caused by not-enough horizontal placement.

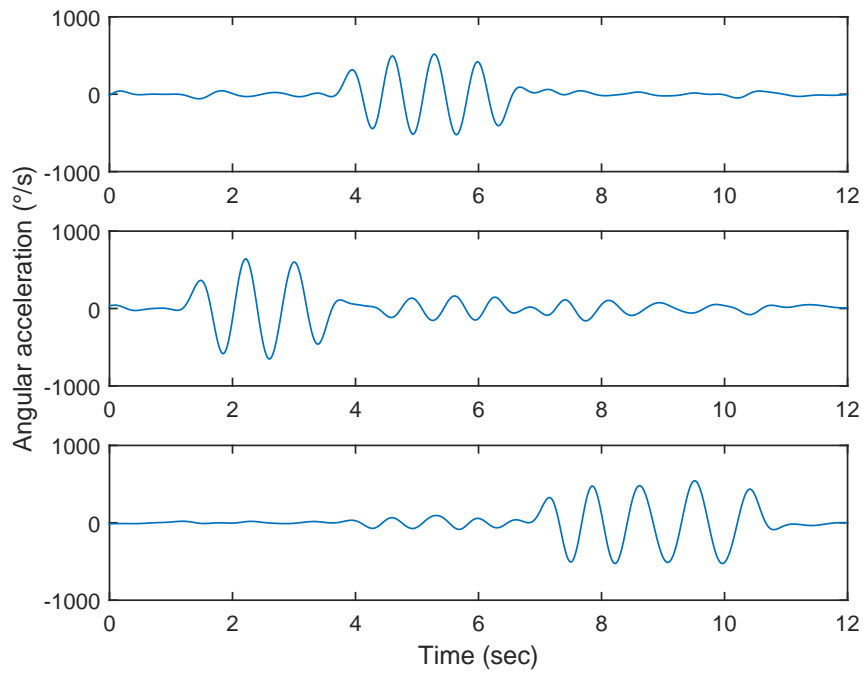


Figure 4.5: The gyroscope data generated by certain movement sequentially.

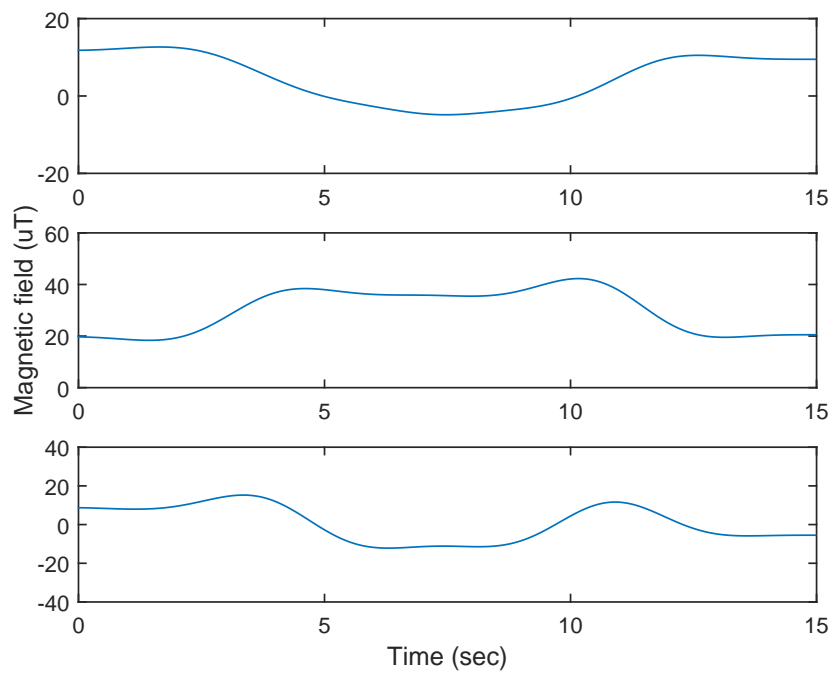


Figure 4.6: The magnetometer data generated by certain movement.

4.3 Result Analysis

In this section, we will present the result and analyzing of experiment. It can separate in 2 case: user-dependent case and user-independent case. User-dependent case is that the user of testing data may be included in training data set, user-independent case is that neither the user of testing data is included in training data set.

4.3.1 Thresholding Method Analysis

In the thresholding method, it is an user-dependent case that we take Q2 of calibration data of each user as threshold value. Window size and window-scan interval for each movement are listed in Table 4.2.

Table 4.2: Thresholding result for the 2 movements - Up and Left.

Movement	Window size	Window-scan interval	Accuracy
Up	25	20	81.54%
Left	45	30	88.33%

The thresholding method is easy to determine which movement has been done, but there are some limitations:

- Human factors is complicated and fluctuating, some data will easily not pass threshold value even if the movements are done right.
- Some movements will have the same passing-threshold condition. It will easily be detected as wrong movement.
- It is very difficult to predict accurately using thresholding method when number of feature is bigger than 2.

Due to the limitations above, we can only detect 2 movements in our experiment and did not get a high accuracy.

4.3.2 SVM Analysis in User-dependent/User-independent Case

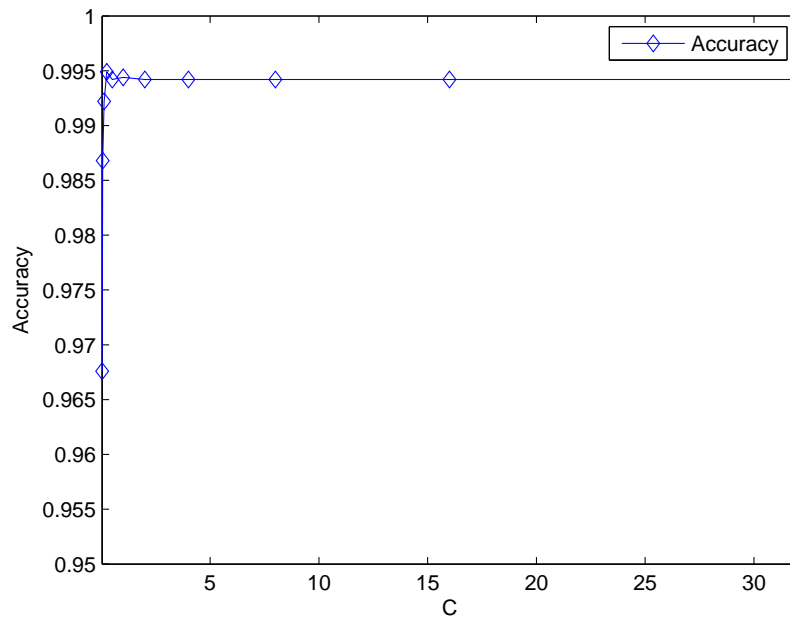
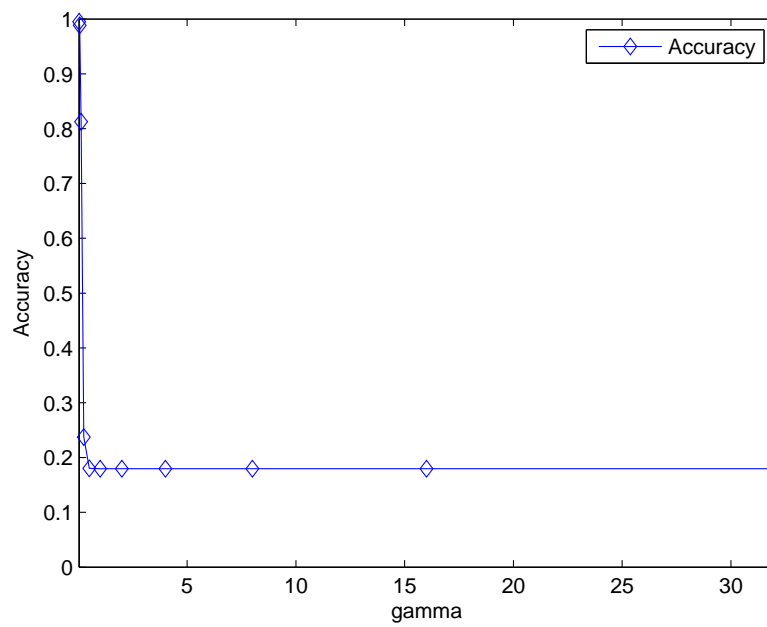
Before using LIBSVM, we use cross-validation technique to estimate the accuracy of each parameter combination in the specified range of training data, and get the best combination of 2 parameters - γ and C , which is 0.03125 and 0.25 respectively to predict our testing data. In Table 4.3a and Figure 4.7a show the changes of accuracy at fixed γ and various C . Similarly, Table 4.3b and Figure 4.7b show the changes of accuracy at fixed C and various γ .

Table 4.3: Accuracy under different combination of C and γ .

(a) $\gamma = 0.03125$		(b) $C = 0.25$	
C	Accuracy	γ	Accuracy
0.03125	0.9676	0.03125	0.9949
0.0625	0.9868	0.0625	0.9886
0.125	0.9922	0.125	0.8127
0.25	0.9949	0.25	0.2373
0.5	0.9942	0.5	0.1803
1	0.9944	1	0.1796
2	0.9942	2	0.1796
4	0.9942	4	0.1796
8	0.9942	8	0.1796
16	0.9942	16	0.1796
32	0.9942	32	0.1796

User-dependent Case

After classification, a predict label of each feature vector will be shown in the output file. Comparing the target label (which the movement should be) and predict label, we can get a confusion matrix for user-dependent case as in Table 4.4. It is a confusion matrix of our system whose features are signals of accelerometer and gyroscope with scanning window size of 50 and sample rate of 50 Hz. With the confusion matrix below, we got the accuracy of

Figure 4.7: Comparison of accuracy under different C (a) $\gamma = 0.03125$ (b) $C = 0.25$

99.49%.

Table 4.4: Confusion matrix in user-dependent case based on the method of PCA+LDA+SVM with kernel model when widow size is 50.

Predict \ Target	Up	Down	Left	Right	Circle	Circle (counter)	Turn L	Turn R
Up	452	0	0	0	0	0	0	0
Down	0	391	0	0	0	3	0	0
Left	0	0	487	0	7	0	0	0
Right	0	0	0	466	4	7	0	0
Circle	0	0	0	0	679	0	0	0
Circle (counter)	0	0	0	0	0	748	0	0
Turn L	0	0	0	0	0	0	452	0
Turn R	0	0	0	0	0	0	0	409

From the table, we can see that there are wrong classification in drawing circle but not in turn left and turn right. The reason is that circle can be composed of up, down, left and right, and the strength of these 4 direction may not be distributed evenly when done by different people, so in the period of greater strength may have wrong classification.

User-independent Case

Here shows the result of user-independent case which the testing user is not included in the training subjects. Table 4.5 is the confusion matrix of user-independent case for 12 subject (5 times for each movement) testing data, we got the accuracy of 88.43% with scanning window size of 50 and sample rate of 50 Hz.

From the table above, turn left and turn right have the highest distinguishment, they have the smallest number of wrong classification. While the wrong distinguishment of up, left and right is due to inertia action of people or the unclear signal of off-line analysis.

Comparing to user-dependent case, the accuracy would be relatively low and it is also the most realistic situation in real life. Even if we have set the movement specification, different

Table 4.5: Confusion matrix in user-independent case based on the method of PCA+LDA+SVM with kernel model when widow size is 50.

Predict \ Target	Up	Down	Left	Right	Circle	Circle (counter)	Turn L	Turn R
Up	370	3	6	4	14	14	1	0
Down	0	348	1	0	32	8	0	2
Left	20	0	385	36	5	0	0	1
Right	2	0	36	322	0	0	0	0
Circle	2	0	0	0	382	0	0	0
Circle (counter)	0	0	0	0	1	384	0	0
Turn L	51	2	5	0	56	20	282	29
Turn R	1	0	0	0	4	3	0	282

strength of every direction, inertia action of different people or doing accidentally wrong may cause the decreasing of accuracy. In the future, it is possible to improve the situation of misjudgment by correcting the standard coordinates or increasing the number of feature types.

Different Parameters Analysis

In the previous studies, sample rate are different from each study. We wonder that if sample rate will influence the accuracy or not, so we implement down-sampling to analyze the accuracy in different sample rate(γ , C and window size are 0.03125, 0.25, 50 respectively in each case). The result of sample rate comparison is shown in Table 4.6. It shows that we can get a higher accuracy in higher sample rate.

Table 4.6: Comparison of different sample rate.

Sample rate	50Hz	40Hz	30Hz	20Hz	10Hz
Accuracy	99.49%	99.35%	99.11%	98.29%	96.62%

Next, we analyze the effect of window size mentioned in section 3.7.2. Originally, we

use sensor sample rate - 50 hz as window size, here we will adjust the window size from 10 to 100 to see how the accuracy changes. The result is listed in Table 4.7 and the trend of accuracy is in Figure 4.8. From the figure, it shows that the bigger the size of window, the higher the accuracy is, and the accuracy does not change significantly between 60 to 100. We got the highest accuracy at 99.63% when window size equalled to 100. Table 4.8 shows the confusion matrix of the system when window size was 100. Comparing to Table 4.4, the number of wrong classification decreased when window size adjusted to 100. 99.63% accuracy was the best result we could get in analysis. There are still some chance to improve accuracy if commencing on adding more features and global axes calibration in the future.

Table 4.7: Comparison of different window size in sample rate of 50 Hz.

Window size	Accuracy
10	81.89%
20	91.83%
30	97.30%
40	98.99%
50	99.49%
60	99.52%
70	99.59%
80	99.60%
90	99.62%
100	99.63%

4.3.3 Comparison of Thresholding and SVM

In thresholding method, we got an average accuracy of 84.94% for only 2 classes, and in SVM, we got accuracy of 99.39% in 8 classes. In Table 4.9 and Figure 4.9, it is the accuracy comparison of the two method. We can see that SVM performed better than thresholding. Comparing to the studies and our experiment using SVM, thresholding did not have a good performance, so that is the reason why we chose SVM rather than thresholding method for

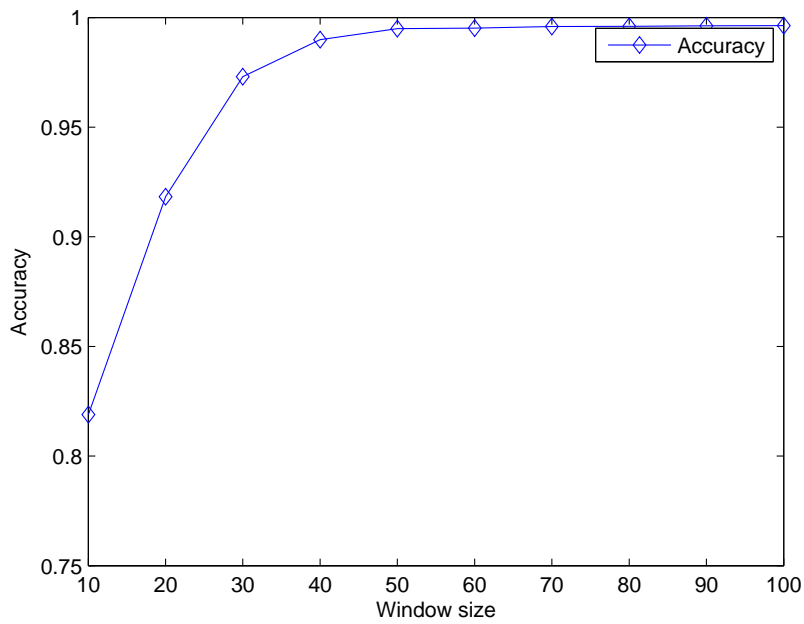


Figure 4.8: Line chart for accuracy comparing of different window size.

Table 4.8: Confusion matrix in user-dependent case based on the method of PCA+LDA+SVM with kernel model when window size is 100.

Predict \ Target	Up	Down	Left	Right	Circle	Circle (counter)	Turn L	Turn R
Up	452	0	0	0	0	0	0	0
Down	0	391	0	0	0	2	0	0
Left	0	0	487	0	5	0	0	0
Right	0	0	0	466	3	5	0	0
Circle	0	0	0	0	682	0	0	0
Circle (counter)	0	0	0	0	0	751	0	0
Turn L	0	0	0	0	0	0	452	0
Turn R	0	0	0	0	0	0	0	409

our system.

Table 4.9: Accuracy comparison of the 2 method.

Method	Classes	Average Accuracy
Threshoding	2	84.94%
SVM	8	99.63%

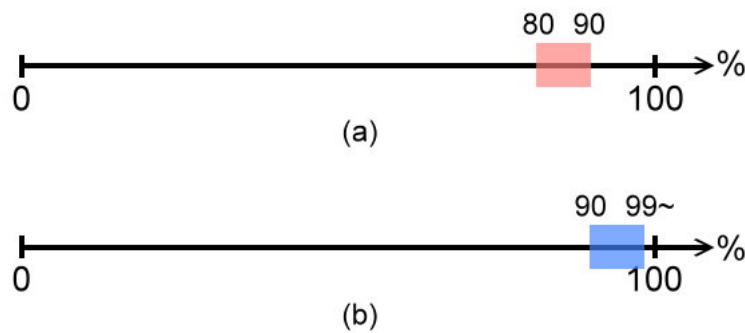


Figure 4.9: Accuracy comparison of the 2 method. (a)Thresholding method. (b)SVM.

4.3.4 Comparison between Our Work and Some Related Studies

There are some inertial sensing studies using different classifiers and Table 4.10 shows the comparison result of our work and other papers. In the table, it lists the same recognized patterns and different patterns of each studies, all of them are geometry patterns. The 6 patterns: Up, down, left, right, clockwise circle and counter clockwise circle are the most common gestures in every studies. The factors for influencing accuracy may be:

1. Different features chosen.
2. The amount of experiment samples.
3. The proportion between training data and testing data.
4. Definition of different recognized patterns.
5. Class number.
6. System algorithms for every step of the flow.

In our work, the patterns included the 6 patterns mentioned above, joined turn right and turn left additionally. The features were chosen the raw data from the sensor and combined with change of timing, and the classes number is 8. The higher accuracy of our work is credited to our method of feature generation which combined with sliding windows, the method of feature extraction and the integration of SVM classifier. The extra patterns of turning left and right may also be an element of accuracy improving.

In Table 4.11, it list the comparison of our work and the studies using SVM classifiers which is the same with us. The class number is similar, but in feature choosing part, we additionally included signals of gyroscope and combination with windows. The more feature being chosen, the higher the accuracy is. The more different identification the patterns are, the easier the system can recognize. We can see that the accuracy of our system is higher than others for sure, it concludes that we have improved the performance of the gesture recognition system.

From the table, the accuracy of our work is higher than others, the key reason is that comparing to others, we not only used acceleration as features, but also added signals of gyroscope to feature set, and also combined features with sliding window to adapt dynamic movements. In our system, we used only one 9-axis sensor and got prediction accuracy of 99.63% in separating 8 classes movements, it proves that our system may be better than others. Different patterns and sample rate may also be an impact on accuracy. Sample rate of our system is 50 hz, it is lower than Marqus *et al.* [33] and higher than Ducloux *et al.* [6]. We have mentioned before that accuracy would be higher when sample rate increased, but when the sample rate is high enough and arrives to a certain level, the accuracy would not change significantly. Due to this reason, we can say that sample rate of 50hz is enough and better for gesture recognition. Overall, although the training time can still be improved, the different features and sample rate of our system lead to a higher accuracy than the others.

Table 4.10: Comparison of our work with related studies.

	Akl et al. [31]	Gupta et al. [4]	Aravind Kailas [35]	Kuroki et al. [5]
Same patterns	U, D, R, L, C, C-C	U, D, R, L, C, C-C	R	U, D, R, L,C
Different patterns	triangle, 90 corner, "ε", "m", "3", "w"	slanting (4 direction), "s" (2 direction)	F, B, F-B, B-F	turn R, turn D, F-B, D-U-D
Features	acc	acc, gyro	acc, gyro, euler angles	acc
Classes	12	12	5	10
Experiment samples	7 subjects (30 times each)	10 subjects (100 times each)	20 subjects	2 subjects (5 minutes)
DTW	94.6% (ave)	93.56% (ave)	-	-
kNN	-	-	80% (ave)	77.40%
SVM	-	-	-	-
	Marques et al. [33]	Ducloux et al. [6]	Our work	
Same patterns	U, D, R, L, C, C-C	U, D, R, L, C, C-C	U, D, R, L, C, C-C	
Different patterns	ON/OFF	square, 90 corner	turn R, turn L	
Features	generated by acc	acc	acc & gyro with windows	
Classes	7	8	8	
Experiment samples	60 subjects samples / per gesture	8 subjects (10 times each)	User-dependent	User-independent
			20 subjects (5 times each)	12 subjects (5 times each)
DTW	-	-	-	-
kNN	-	-	-	-
SVM	98%~99%	97.31%	99.63%	88.43%

Table 4.11: Comparisons to the related works using SVM classifier in user-dependent case.

	Our work	Marques et al. [33]	Ducoux et al. [6]
Classifier	SVM		
Class	8	7	8
Same patterns	up,down, left, right, clockwise circle, counter-clockwise circle		
Different patterns	turn left, turn right	ON/OFF	square, 90 corner
Features	acc & gyro with widows	acc	acc
Accuracy	99.63%	98%~99%	97.31%



Chapter 5

Conclusions and Future Works

5.1 Conclusions

Gesture recognition is an potential and interesting field which has been studied for years. It now has many applications in activity of daily livings (ADL). In this thesis, we have the goal of using the gesture recognition technology to make people's life be filled with joy, so we introduced a system of gesture recognition which can recognize geometrical movements and used only one inertial body sensor to make it conveniently to use in interactive activity. In the beginning of the thesis, we introduced 9-axis signals and how we would use them to develop the system. Choosing signals of accelerometer and gyroscope as features and combined with sliding windows made the system more robust. Then, feature extraction method of PCA integrated with LDA, which promoted the system's performance, was presented. The developed thresholding method and machine learning system were also analyzed and chose machine learning process (PCA+LDA+SVM) as final method in our system. Except analyzing parameters to get higher accuracy, we also made analysis of different window size and sample rate to see if it would help the increase of accuracy.

In analyzing the parameters with SVM classifier, we got the best accuracy of 99.63% at $\gamma = 0.03125$, $C = 0.25$, window size in 100 and sample rate in 50hz. It is higher than the other studies we mentioned in 4.3.4. The reason of not using thresholding method was also mentioned before, even if it took less time in calculation. By the SVM classifier and the

particular feature extraction method integrated with PCA and LDA, our system became more solid. The purpose of our system is not only achieve higher accuracy in gesture recognition field, but also can be applied in different field such as art, gaming or wisdom appliances in the future.

5.2 Future Works

The system is aimed at accurately distinguishing different gestures generated by 9-axis signals. The ideal result is presented above. There are still some functions of the system can be improved in the future. The training time of the classifier can be reduced to apply in real-time operation of human-machine interface. Dynamic Euler angles which generated by fusion-sensor can be add into the feature set to achieve much higher accuracy. The 8 defined movement to be recognized is all follow the provided range and direction of the rule mentioned in section 4.1.1, but due to human cognition, people will see the same movement in different direction as the equivalent. Therefore, the orientation calibration of the 9-axis signals is an important thing to be performed. With wireless functionality of the sensor, we can implant our system in smartphone applications.

Bibliography

- [1] Z. Zhang, Z. Wu, J. Chen, and J.-K. Wu, "Ubiquitous human body motion capture using micro-sensors," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, March 2009, pp. 1–5.
- [2] J. S. Wang and F. C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 2998–3007, July 2012.
- [3] M. Zhang and A. A. Sawchuk, "A customizable framework of body area sensor network for rehabilitation," in *2009 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Nov 2009, pp. 1–6.
- [4] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, "A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6425–6432, Aug 2016.
- [5] K. Kuroki, Y. Zhou, Z. Cheng, Z. Lu, Y. Zhou, and L. Jing, "A remote conversation support system for deaf-mute persons based on bimanual gestures recognition using finger-worn devices," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, March 2015, pp. 574–578.
- [6] J. Ducloux, P. Colla, P. Petrashin, W. Lancioni, and L. Toledo, "Accelerometer-based hand gesture recognition system for interaction in digital tv," in *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, May 2014, pp. 1537–1542.

- [7] L. Chen, F. Wang, H. Deng, and K. Ji, "A survey on hand gesture recognition," in *Computer Sciences and Applications (CSA), 2013 International Conference on*, Dec 2013, pp. 313–316.
- [8] M. Panwar, "Hand gesture recognition based on shape parameters," in *2012 International Conference on Computing, Communication and Applications*, Feb 2012, pp. 1–6.
- [9] P. Pawiak, T. Sonicki, M. Niedwiecki, Z. Tabor, and K. Rzecki, "Hand body language gesture recognition based on signals from specialized glove and machine learning algorithms," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1104–1113, June 2016.
- [10] X. Zhao, A. M. Naguib, and S. Lee, "Kinect based calling gesture recognition for taking order service of elderly care robot," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Aug 2014, pp. 525–530.
- [11] D. H. Shin and W.-S. Jang, "Utilization of ubiquitous computing for construction {AR} technology," *Automation in Construction*, vol. 18, no. 8, pp. 1063 – 1069, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0926580509000922>
- [12] G. Welch and E. Foxlin, "Motion tracking: no silver bullet, but a respectable arsenal," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 24–38, Nov 2002.
- [13] S. Zhou, Q. Shan, F. Fei, W. J. Li, C. P. Kwong, P. C. K. Wu, B. Meng, C. K. H. Chan, and J. Y. J. Liou, "Gesture recognition for interactive controllers using mems motion sensors," in *Nano/Micro Engineered and Molecular Systems, 2009. NEMS 2009. 4th IEEE International Conference on*, Jan 2009, pp. 935–940.
- [14] L. Huan and R. Bo, "Human gesture recognition based on image sequences," in *Control Conference (CCC), 2015 34th Chinese*, July 2015, pp. 8388–8392.
- [15] Y. Tao, H. Hu, and H. Zhou, "Integration of vision and inertial sensors for 3d arm motion tracking in home-based rehabilitation," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 607–624, 2007. [Online]. Available: <http://ijr.sagepub.com/content/26/6/607.abstract>

- [16] S. Zhou, F. Fei, G. Zhang, J. D. Mai, Y. Liu, J. Y. J. Liou, and W. J. Li, "2d human gesture tracking and recognition by the fusion of mems inertial and vision sensors," *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1160–1170, April 2014.
- [17] M. A. Amin and H. Yan, "Sign language finger alphabet recognition from gabor-pca representation of hand gestures," in *2007 International Conference on Machine Learning and Cybernetics*, vol. 4, Aug 2007, pp. 2218–2223.
- [18] M. Elmezain, A. Al-Hamadi, S. S. Pathan, and B. Michaelis, "Spatio-temporal feature extraction-based hand gesture recognition for isolated american sign language and arabic numbers," in *Image and Signal Processing and Analysis, 2009. ISPA 2009. Proceedings of 6th International Symposium on*, Sept 2009, pp. 254–259.
- [19] A. Seniuk and D. Blostein, "Pen acoustic emissions for text and gesture recognition," in *2009 10th International Conference on Document Analysis and Recognition*, July 2009, pp. 872–876.
- [20] R. Angeles, "Rfid technologies: Supply-chain applications and implementation issues," *Information Systems Management*, vol. 22, no. 1, pp. 51–65, 2005. [Online]. Available: <http://dx.doi.org/10.1201/1078/44912.22.1.20051201/85739.7>
- [21] G. Borriello, W. Brunette, M. Hall, C. Hartung, and C. Tangney, *Reminding About Tagged Objects Using Passive RFIDs*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 36–53. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30119-6_3
- [22] L. Vlaming, J. Smit, and T. Isenberg, "Presenting using two-handed interaction in open space," in *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, Oct 2008, pp. 29–32.
- [23] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall, "Recognizing daily activities with rfid-based sensors," in *Proceedings of the 11th International Conference on Ubiquitous Computing*, ser. UbiComp '09. New York, NY, USA: ACM, 2009, pp. 51–60. [Online]. Available: <http://doi.acm.org/10.1145/1620545.1620553>

- [24] K. P. Fishkin, B. Jiang, M. Philipose, and S. Roy, *I Sense a Disturbance in the Force: Unobtrusive Detection of Interactions with RFID-tagged Objects*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 268–282. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30119-6_16
- [25] O. Kubitz, M. O. Berger, M. Perlick, and R. Dumoulin, “Application of radio frequency identification devices to support navigation of autonomous mobile robots,” in *Vehicular Technology Conference, 1997, IEEE 47th*, vol. 1, May 1997, pp. 126–130 vol.1.
- [26] M. Atif, L. Kulik, and E. Tanin, “Autonomous navigation of mobile agents using rfid-enabled space partitions.”
- [27] P. Asadzadeh, L. Kulik, and E. Tanin, “Gesture recognition using rfid technology,” *Personal and Ubiquitous Computing*, vol. 16, no. 3, pp. 225–234, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s00779-011-0395-z>
- [28] C. Hu, M. Li, S. Song, W. Yang, R. Zhang, and M. Q. H. Meng, “A cubic 3-axis magnetic sensor array for wirelessly tracking magnet position and orientation,” *IEEE Sensors Journal*, vol. 10, no. 5, pp. 903–913, May 2010.
- [29] J. S. Wang and F. C. Chuang, “An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 2998–3007, July 2012.
- [30] S. D. Choi and S. Y. Lee, “3d stroke reconstruction and cursive script recognition with magnetometer-aided inertial measurement unit,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 661–669, May 2012.
- [31] A. Akl, C. Feng, and S. Valaee, “A novel accelerometer-based gesture recognition system,” *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6197–6205, Dec 2011.
- [32] J. S. Wang and F. C. Chuang, “An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 2998–3007, July 2012.

- [33] G. Marqus and K. Basterretxea, "Efficient algorithms for accelerometer-based wearable hand gesture recognition systems," in *Embedded and Ubiquitous Computing (EUC), 2015 IEEE 13th International Conference on*, Oct 2015, pp. 132–139.
- [34] M. Brown and L. Rabiner, "Dynamic time warping for isolated word recognition based on ordered graph searching techniques," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82.*, vol. 7, May 1982, pp. 1255–1258.
- [35] A. Kailas, "Basic human motion tracking using a pair of gyro + accelerometer mems devices," in *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, Oct 2012, pp. 298–302.
- [36] "MPU-9250 Product Specification," InvenSense, Datasheet, Jan. 2014, rev. 1.0. [Online]. Available: <http://store.invensense.com/datasheets/invensense/MPU9250REV1.0.pdf>.
- [37] X. Yun and E. Bachmann, "Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1216–1227, Dec. 2006.
- [38] R. Zhu and Z. Zhou, "A Real-Time Articulated Human Motion Tracking Using Tri-Axis Inertial/Magnetic Sensors Package," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 12, no. 2, pp. 295–302, Jun. 2004.
- [39] "Accelerometer," Jul. 2015, page Version ID: 671645806. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Accelerometer&oldid=671645806>.
- [40] R. O'Reilly, K. Harney, and A. Khenkin, "Sonic Nirvana: MEMS Accelerometers as Acoustic Pickups in Musical Instruments," Jun. 2009. [Online]. Available: <http://www.sensorsmag.com/sensors/acceleration-vibration/sonic-nirvana-mems-accelerometers-acoustic-pickups-musical-i-5852>.
- [41] P. Prendergast and B. Kropf, "How to Use Programmable Analog to Measure MEMS Gyroscopes," Jan. 2007. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1274559.

- [42] “Lorentz force,” Jul. 2015, page Version ID: 671382070. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Lorentz_force&oldid=671382070.
- [43] W. Storr, “Hall Effect Sensor and How Magnets Make It Works.” [Online]. Available: <http://www.electronics-tutorials.ws/electromagnetism/hall-effect.html>.
- [44] I. Fodor, “A survey of dimension reduction techniques,” Tech. Rep., 2002.
- [45] H. T. Hung, “Principal component analysis, pca,” [Online; accessed 04-August-2016]. [Online]. Available: [http://120.118.226.200/member/hunght/i\(iem\)/principalcomponentanalysis.pdf](http://120.118.226.200/member/hunght/i(iem)/principalcomponentanalysis.pdf)
- [46] “Linear discriminant analysis,” Mar 2014, [Online; accessed 05-August-2016]. [Online]. Available: http://sebastianraschka.com/articles/2014_python_lda.html
- [47] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, p. 179188, 1936.
- [48] C. R. Rao, “The utilization of multiple measurements in problems of biological classification,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 10, no. 2, pp. 159–203, 1948. [Online]. Available: <http://www.jstor.org/stable/2983775>
- [49] E. Alpaydin, *Introduction to machine learning*. MIT Press, 2010.
- [50] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.