# DATA-GLOVE-BASED HAND GESTURE RECOGNITION SYSTEM USING FLEX SENSORS AND AN IMU SENSOR

**ONG JING HAO**

**UNIVERSITI SAINS MALAYSIA**

**2017**

# DATA-GLOVE-BASED HAND GESTURE RECOGNITION SYSTEM USING FLEX SENSORS AND AN IMU SENSOR

**by**

**ONG JING HAO**

**Thesis submitted in partial fulfilment of the**
**requirements for the degree of**
**Bachelor of Engineering (Electronic Engineering)**

**UNIVERSITI SAINS MALAYSIA**

**2017**

# ACKNOWLEDGEMENT

This dissertation is dedicated to everyone in the field of hand gesture recognition who embarks on the journey of expanding the collection of knowledge and transcendent passion for hand gesture recognition.

First and foremost, I am very grateful to the School of Electrical and Electronic Engineering of Universiti Sains Malaysia for providing me the opportunity to undertake and complete this interesting project. The resources and facilities provided by the school are important in accomplishing the project.

Besides that, I wish to express my sincere gratitude to my respected thesis advisor and project supervisor, Dr. Patrick Goh Kuan Lye, who was my backbone and guide throughout the project. His invaluable supports, insightful advices, and encouragements are important for me to bring out a successful project.

In addition, I wish to thank my respected project examiner, En. Mohd Nazri Mahmud for his valuable comments and suggestions. Besides that, I would like to express my gratitude to the officers and staffs of the School of Electrical and Electronic Engineering of Universiti Sains Malaysia who provide their support and cooperation to me in completing the project.

Last but not least, I offer my best regards and blessing to my friends and beloved family who support me in any aspect during the completion of this project. Their supports are greatly appreciated and helpful to me in completing the project.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS AND ABBREVIATION

| | |
|---|---|
| ADC | Analogue-to-Digital Converter |
| ANN | Artificial Neural Network |
| ASL | American Sign Language |
| DTW | Dynamic Time Warping |
| EMG | Electromyography |
| GPIO | General Purpose Input/ Output |
| GUI | Graphical User Interface |
| HCI | Human-Computer Interface |
| HMM | Hidden Markov Model |
| IMU | Inertial Measurement Unit |
| IP | Internet Protocol |
| I2C | Inter-Integrated Circuit |
| k-NN | k-Nearest Neighbor |
| MEMS | Microelectromechanical System |
| OS | Operating System |
| SPI | Serial Peripheral Interface |
| SVM | Support Vector Machines |
| UART | Universal Asynchronous Receiver/Transmitter |

# SISTEM PENGECAMAN GERAKAN ISYARAT TANGAN BERDASARKAN SARUNG TANGAN DENGAN MENGGUNAKAN PENDERIA LENTUR DAN IMU

## ABSTRAK

Dengan pengembangan teknologi komputer yang pesat, Interaksi Manusia-Komputer (HCI) perlu menjadi lebih berkesan. Gerakan isyarat tangan lebih semula jadi berbanding dengan gerakan yang berkaitan dengan peranti tradisional seperti papan kekunci, tetikus, dan lain-lain. Tesis ini mencadangkan sistem pengecaman gerakan isyarat tangan berdasarkan sarung tangan dengan menggunakan penderia lentur dan IMU yang boleh mengenali gerak isyarat tangan statik dan dinamik untuk membantu manusia berinteraksi dengan komputer secara lebih semula jadi. Sistem ini mengenali gerakan isyarat tangan berdasarkan maklumat yang diperolehi oleh penderia lentur dan IMU. Sudut lenturan jari diukur dengan menggunakan penderia lentur manakala kecondongan tangan dikesan dengan menggunakan penderia IMU. Kemudian, data yang diperoleh diproses di dalam modul Raspberry Pi. Algoritma penapis Complementary digunakan untuk menggabung data dari akselerometer dan giroskop yang terdapat di dalam IMU untuk mendapatkan pengukuran yang tepat. Sistem ini adalah berdasarkan k-jiran terdekat (k-NN) algoritma pengelasan, dinamik masa meleding (DTW) dan Euclidean algoritma metrik jauh. Sistem yang dicadangkan telah diuji dengan mengenali 38 statik dan 12 dinamik gerakan isyarat tangan. Maksud gerakan isyarat tangan dipaparkan pada GUI yang dicipta dalam Raspberry Pi dengan menggunakan pemprograman Tkinter Python. Ketepatan 98.97% dicapai bagi sistem ini dalam pengecaman 50 gerakan isyarat tangan sekiranya tiada gangguan daripada pengguna.

# DATA-GLOVE-BASED HAND GESTURE RECOGNITION SYSTEM USING FLEX SENSORS AND AN IMU SENSOR

## ABSTRACT

With the great expansion of computer technology, Human-Computer Interaction (HCI) is required to be more effective. Hand gestures are more natural compared with actions associated with the traditional devices such as the keyboard, mouse, etc. This thesis proposes a wearable data gloved-based hand gesture recognition system that is able to recognize static hand and dynamic gestures to allow human interacts with the computer in more natural manner. This system recognizes the hand gestures based on the information captured by the flex sensors and an IMU sensor. The fingers' bending angles are measured by using the flex sensors while the pitch and roll of the hand are detected by using the IMU sensors. The acquired data is then processed in Raspberry Pi board. The Complementary filter is used to fuse the data from the accelerometer data and gyroscope packed in IMU sensor to obtain an accurate measurement. This system is based on k-Nearest Neighbors (k-NN) classifier algorithm, Dynamic Time Warping (DTW) and Euclidean distance metric algorithms. The proposed system was tested in recognizing 38 static and 12 dynamic hand gestures. The meaning of the hand gesture is displayed on GUI created in Raspberry Pi by using Python's Tkinter programming. An accuracy of 98.97 % is achieved by this system in recognizing 12 dynamic and 38 static hand gestures without the user's noise.

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Human-Computer Interaction (HCI) has expanded steadily and rapidly for three decades. Nowadays, HCI has changed from user interface to gesture interface [1]. Physical gesture as intuitive expression can greatly ease the interaction process and allow humans to communicate with computers more naturally [10]. Hence, the gesture recognition has grabbed attention from many professionals from other disciplines [2]. The hand gesture is the most animated, communicative, and most often used in communication between human compared with other body parts for gesturing [3]. Hence, it is best suited for Human-Computer Interaction.

The hand gesture may be static or dynamic. Static hand gestures, also called hand posture are fixed with respect to time while dynamic hand gestures involve the movement of the hand with respect to time [2]. Much researches have been done in the field of hand gesture recognition. The hand gesture recognition systems can be broadly classified into glove-based and vision-based. In data-glove-based systems, the hand gestures are transferred to the computer using a data glove worn on the hand. In vision-based systems, user's hand is tracked by a camera. The hand gesture's image captured by a camera is then recognized by computer based on the feature extraction.

This research is intended to develop a data-glove-based hand recognition system by using flex sensors and an IMU sensor to recognize the static and dynamic hand gestures. Typically, flex sensor is used to measure the joint angle of fingers while IMU sensor is used in detecting the pitch, roll or yaw of hand. The combination of these two

types of sensors allows the system to capture more features of the hand gesture performed. Based on the extracted features, the system is able to recognize the hand gestures, translate them into text and display on a GUI application.

## 1.2 Problem Statement

There are at least 360 million deaf-mute people around the world [6]. Due to learning disabilities, they often convey with the general public by using sign languages. However, the majority of healthy people do not understand the sign language, creating a gap between these patients and the public, as well as difficulties in their daily life living. Besides that, worldwide sign language is not a universal language. Each country has its own variation of sign language like American Sign Language (ASL), British Sign Language (BSL), Vietnamese Sign Language (VSL), Chinese Sign Language (CSL), Thai Sign Language (TSL), and others [7]. There is a challenge for deaf-mute people communicate with others. Recently, researches has developed the sign language translator to translate the sign languages to text or speech, ease the communication of deaf-mute people with others in order to improve the quality of his or her life. However, the sign language translator still need be improved. It requires an effective Human-Computer Interaction (HCI) to translate the user's expression perfectly. Hence, a lot of research works in hand gesture recognition have been done to improve the Human-Computer Interaction.

Table 1.1 shows some of the previous related works in hand gesture recognition with different classification algorithm and input device. By summarizing these research works, hand gesture recognition can be developed into two directions, by using data-glove-based technique with the sensors mounted on a glove or the vision-based technique

with a camera. The vision-based methods much relies on the assumption of high contrast stationary backgrounds and ambient light conditions may affect the hand gesture recognition accuracy [8]. Besides that, the classification algorithms used in vision-based recognition systems like HMM and ANN algorithms are much complicated than algorithms used in data-glove-based technique [9]. Hence, in recent times, the data-glove-based technique is researched as to the demand of the virtual reality and animation industry.

Table 1.1: Difference approaches in hand gestures recognition

| Classification Algorithm | Static / Dynamic hand gesture | Number of Hand Gesture | Accuracy | Input device | Remark |
|---|---|---|---|---|---|
| k-NN Classifier [3] | Static | 31 | 98.54% | Data glove with 6 flex sensors | Number 0 to 5 and alphabet A to Z self-defined hand gesture |
| k-NN Classifier [4] | Static | 10 | 98.82% | Data glove with an IMU sensor and 4 accelerometers | 10 self-defined hand gestures |
| Template Matching [7] | Static and Dynamic | 29 | 90.34% | Data glove with flex sensors and accelerometer | 29 Vietnamese Sign alphabets |
| Bayes Linear Classifier [12] | Dynamic | 19 | 95% | 3-axis accelerometer and | 19 mobile operation instructions |

| | | | | | |
|---|---|---|---|---|---|
| | | | | electromyography (EMG) sensor | |
| Support Vector Machine (SVM) [15] | Static | 26 | 94.23% | Camera | 26 Indian Sign Language |
| Hidden Markov Models (HMM) [16] | Dynamic | 10 | 91.9% | Camera | Numbers 0-9 |
| Artificial Neural Network (ANN) [17] | Static | 36 | 94.32% | Mobile Video Camera | ASL number 0 to 9 and ASL alphabet A to Z |
| Decision-Tree (DT) [18] | Static | 24 | 95.8% | Leap Motion | 24 Chinese Sign Language |

For data glove technique, there is various kind of sensors can be implemented on the glove like flex sensor, accelerometer sensor, gyroscope sensor, capacitive touch sensor, electromyography (EMG) sensor, and IMU sensor [2] [3] [4] [7] [12]. The accuracy of results by using EMG sensor is less than satisfactory because EMG signal is different for every person since each person has different physical conditions such as the subcutaneous fat quantity and skin impedance [13]. The gyroscope data is reliable only in the short term, as it starts to drift on the long term because of the integration over time, the measurement has the tendency to drift, not returning to zero when the system backs to its original position. The accelerometer data is reliable only in the long term. Since accelerometer measures all forces that are working on the object more than just the gravity vector, small unwanted motion and force working on the object will disturb its

measurement completely [14]. However, the accelerometer data and gyroscope data can be fused by using sensor fusion algorithm to obtain an accurate measurement. The sensors fusion algorithm used commonly are Kalman filter and Complementary filter [14].

By reviewing the previous works listed in Table 1.1, the number of hand gesture recognized by the data-glove-based hand gesture recognition system should be increased. To increase the number of hand gestures, the combination of various types of sensors should be implemented so the system can extract more features of hand gesture. Moreover, the accuracy of systems still can be improved. The hand gestures with high similarity in features should be avoided to improve the result's accuracy of the hand gesture recognition system.

## 1.3 Objectives

The main objectives of this research are:

1. To develop a data-glove-based hand gesture recognition system that is able to recognize static and dynamic hand gestures by using flex sensors and an IMU sensor with Raspberry Pi board.
2. To increase the number of hand gestures to be recognized by system compared to previous related works.
3. To improve the hand gesture recognition rate compared to previous related works.

## 1.4 Scope of Research

A data-glove-based hand gesture recognition system that is able to recognize static and dynamic hand gestures will be developed. The system is made of a Raspberry Pi board, an ADC module, and seven flex sensors and an IMU sensor mounted on nitrile

glove. Software used in system consists of sensors fusion algorithm based on Complementary filter and classification algorithm based k-Nearest Neighbors (k-NN) and distance metric algorithms based on Dynamic Time Warping (DTW) and Euclidean distance. Seven flex sensors are connected in voltage divider configuration to be used in tracing the finger bending angle. Since Raspberry Pi cannot direct read the analogue data, the analogue data from the flex sensors should be converted to digital data using an ADC module before fed into Raspberry Pi board. IMU sensor will be used in detecting the pitch and roll of hand. The Complementary filter will be used to fuse the data from accelerometer and gyroscope packed in IMU to obtain an accurate measurement. The acquired data will be processed in Raspberry Pi board. The system will be able to identify the hand gestures based on these data with the k-NN classifier algorithm. Finally, the hand gesture will be translated into text and displayed on GUI created in Raspberry Pi by using Python's Tkinter programming.

## 1.5 Report Outline

This thesis consists of five main chapters which explain the full details and specification from the beginning to the conclusion of this project.

Chapter 1 describes the introduction of this project. The project background, problem statement, project objectives, research scopes are described clearly in this chapter. The ideas, concepts, and reasons to conduct this project are stated. The objectives and research scopes ease the reader in understanding the core of this project.

Chapter 2 describes the literature review. It contains general concepts, previous related works, and the description of hardware considered to be used in this project.

Chapter 3 explains the methodology of this project. The hardware and software used to implement the project are described in detail. The assembly of hardware, software development, and algorithms used in the system are discussed.

Chapter 4 explains the results and discussion of the project. The prototype of the system and the hand gestures defined to be recognized by the system are shown. The accuracy of the system is evaluated and discussed. The method to display the output of the system is also described.

Chapter 5 describes the conclusion and future works. It summarizes the overall project work and achievement. The future work to solve the limitations of this project and the improvement that can be done are covered in this chapter.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Overview

This section begins with the review of existing approaches and application of the hand gesture recognition. Different methods have been proposed in acquiring data for the hand recognition gestures system. The hand gesture recognition system can be used to perform several applications, such sign language translation, virtual reality, gaming, application controlling or other applications. The MEMS-based IMU sensor is considered to be used in the project. However, the data provided by the low-cost MEMS-based IMU is affected by the high noise level and time-varying biases. This can be resolved by applying the sensors fusing algorithm to filter out the noise and drift in order to obtain accurate measurement [24]. Different sensors fusion algorithms are described in this section. The classification algorithm is used to determine the gesture from the acquired information. The recognition rate of the hand gesture recognition system is affected by the classification algorithm [25]. Hence, the review of different classifier algorithms is done to select the best-suited classification algorithm to be used in this project. Different classification algorithm approached in the previous related research works will be discussed. To quantize the similarity of the data from user input and database, distance metric algorithms should be used and will be discussed. Lastly, the hardware considered to be used in the project is described.

## 2.2 Review of Existing Hand Gesture Recognition

Hand gesture recognition has become an increasingly attractive research subject. Researchers have proposed a variety of ways to achieve the hand gesture recognition so far and use it in different applications such as sign language translator, home application controller, mobile self-portrait controller, and projector controller [4] [12] [21] [22] [23] [27]. By summarizing the previous related work, the hand gesture recognition can be broadly categorized into data-glove-based and vision-based techniques.

The data-glove-based technique is having the sensors being mounted on a glove to collect the information about the position, motion, shape, and orientation of hand. There are various kind of sensors can be implemented on the data glove, such as flex sensor, accelerometer sensor, gyroscope sensor, contact sensor, inertial sensor, and electromyography (EMG) sensor [2] [3] [4] [7] [12] [27]. This technique has advantages of high accuracy and fast reaction speed. It can get information about the fingers movement accurately as well as the tiling of hand. Besides that, it can perform hand gesture recognition in high speed due to fewer input data and it does not need to have pre-processing of input data which must be done in vision-based technique. However, wearing of glove all the time is required when using the data-glove-based recognition system [12]. This technique could only be used in laboratory until a major breakthrough in the field of material sciences to make the data glove become handy and light [5].

The vision-based technique uses the cameras include stereo cameras and single camera to collect gesture image and then identify the gesture through processing and analyzing the image. The significant advantage of the vision-based technique is the communication between human and computer need no other intermediate media [19]. The vision-based technique is also able to track the user's hand gesture effectively with no noise from the user. However, this technique relies much on the assumption of high

contrast stationary backgrounds and ambient light conditions which may affect the hand gesture recognition accuracy [2]. In addition, it has a slower dynamic response and comparatively huge data collections and processing compared to data-glove-based technique. It is much complex to implement the vision-based technique compared to data-glove-based technique [6].

## 2.3 Sensors Fusing Algorithm Used in IMU Sensor

The Inertial Measurement Unit (IMU) sensor is an electronic device used to measure accelerations, rotation rates and possibly earth magnetic field with the use of 3-axis accelerometer, 3-axis gyroscope, and sometimes 3-axis magnetometer to measure the attitude or orientation of an object. For recent years, high precision inertial sensors are too expensive. Hence, the low-cost and light-weight Microelectromechanical System (MEMS) inertial sensor is adopted for a wider range of application [24]. However, the data provided by low-cost MEMS-based IMU sensor is affected by high noise level and time varying biases [14]. The integration of gyroscope of IMU sensor measurement errors will lead to an accumulating error in the calculated orientation. It cannot provide an absolute measurement of the orientation. Besides, the accelerometer of IMU sensor is always subject to high levels of noise and susceptible to external acceleration interference. Figure 2.1 and Figure 2.2 show the dynamic response of accelerometer and gyroscope respectively [28]. The accelerometer shows a good dynamic response to the low-frequency motions but the diverge in the high frequency while gyroscope has better fidelity in the high frequency but poor quality in the low-frequency range. To resolve the problems, the sensors fusion algorithm of IMU is applied to obtain a smooth and bias-free estimation of orientation. The most commonly used sensors fusion algorithms are Kalman filter and Complementary filter. Kalman filter is an iterative filter, which is

efficient but high complexity in computation. The Complementary filter is a relative easy algorithm, which only requires light computation and easy to implement [24].

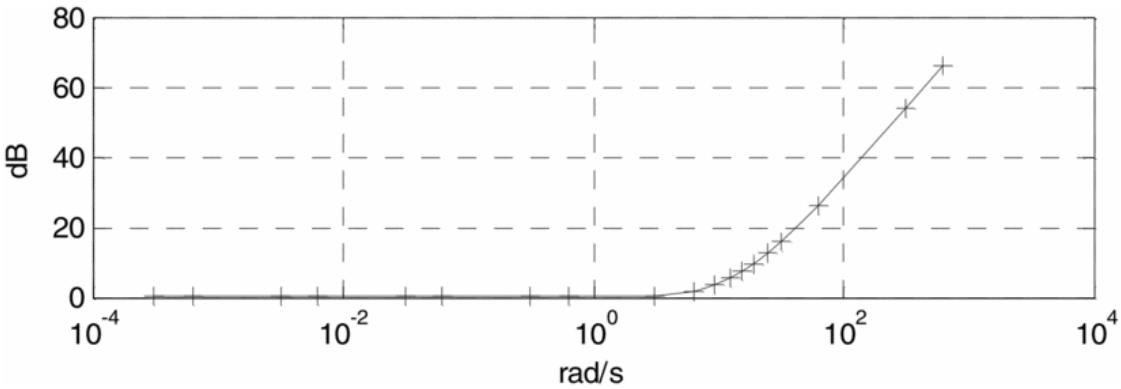


Figure 2.1: Dynamic response of accelerometer [28]

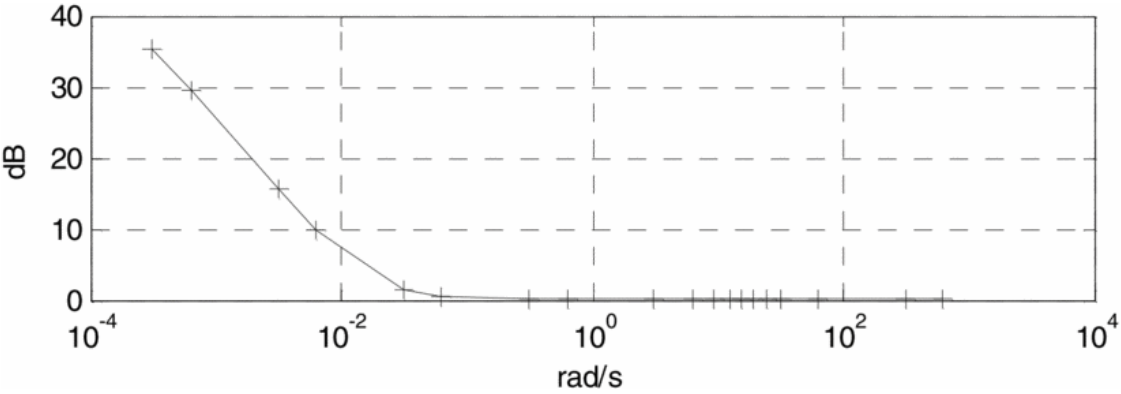


Figure 2.2: Dynamic response of gyroscope [28]

### 2.3.1 Kalman Filter

Kalman filter was a data fusion algorithm first published by R.E. Kalman in 1960. Due to its advances in digital computing, Kalman filter has been the subject of extensive research in the area of the robotic system, human motion control, and navigation [29]. The Kalman filter is a linear quadratic estimation. It is an iterative algorithm which relies on a series of measurements observed over time.

For data fusion process of the IMU sensor, the Kalman filter takes the noise into account through covariance matrices and updates the matrices at each time interval. The

Kalman filter estimates the state of the system based on the previous and current states. The precision of the data can be significantly improved after filtering by using Kalman filter. The key of Kalman filter is to find the weighted average [24]. The standard Kalman filter expressions are shown in Equation 2.1 and 2.2.

$$x_k = F_{x_{k-1}} + B_{u_k} + w_k \tag{2.1}$$

$$z_k = H_{x_k} + v_k \tag{2.2}$$

Where:

$x_k$ is the system state matrix at time $k$,

$F$ is the state transition matrix which is applied to the previous state $x_{k-1}$,

$B$ is the control matrix,

$u_k$ is the control input,

$w_k$ is the process noise,

$v_k$ is the measurement noise,

$z_k$ is the measured output,

$H$ is the measurement matrix used to map the true state space into the observed space [24].

The implementation of Kalman filter includes two main steps: predict and update processes. In predict process, the filter first estimate the current state and the error covariance matrix at time $k$. Next, the priori error covariance matrix based on the previous error covariance matrix is estimated. In the update process, the filter first computes the difference between the measurement $z_k$ and the priori state $\hat{x}_{k|k-1}$. The $\hat{x}_{k|k-1}$ is the previous estimated state based on the previous state and the estimated state before [24].

### 2.3.2 Complementary Filter

The accelerometer and gyroscope have their reliable frequency ranges for measurement as shown in Figure 2.1 and 2.2. Hence, an idea is proposed by compensating the performance of both sensors in the whole range. The Complementary filter is implemented to extract the best attitude of both sensors. As shown in Figure 2.3, the Complementary filter extracts the accelerometer's data at low frequency by using a low pass filter and the gyroscope's data at high frequency by using high pass filter, then fuses them to obtain an accurate tilt angle.

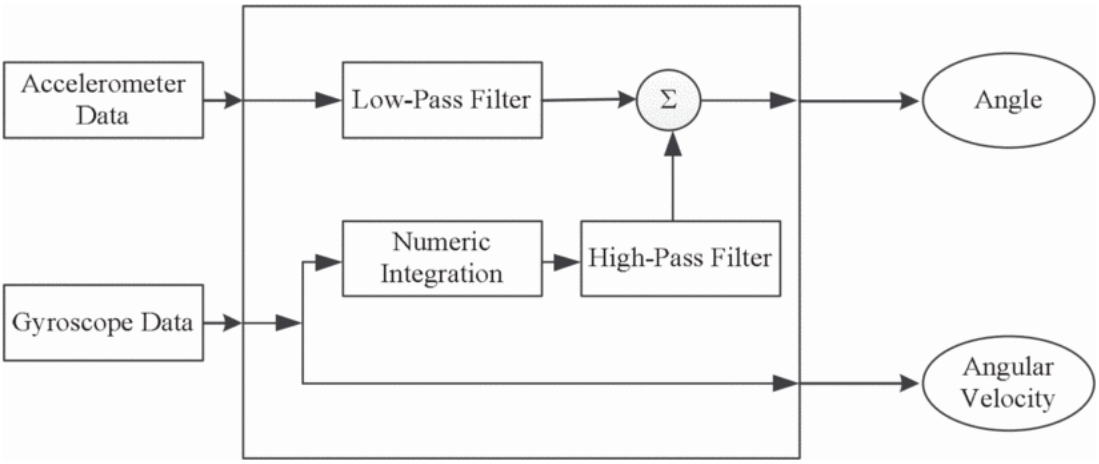


Figure 2.3: Complementary filter algorithm used in IMU sensor [24]

Commonly, the formula of Complementary filter used for IMU to obtain accurate tilt measurement is expressed as Equation 2.4. The filter coefficient, $c$ is determined by Equation 2.5.

$$\theta_n = c\,(\theta_{n-1} + \Omega_n\,dt) + (1-c)\emptyset_n \tag{2.4}$$

Where:

$\theta_n$ is the $n^{th}$ estimate of angle,

$\Omega_n$ is the $n^{th}$ gyroscope data,

$\emptyset_n$ is the rotation angle calculated from $n^{th}$ accelerometer data,

$dt$ is the sampling period,

$c$ is the filter coefficient [30].

$$c = \frac{\tau}{\tau + dt} \tag{2.5}$$

Where $\tau$ is the time constant of filter [30].

## 2.4 Classification Algorithm Used in Hand Gesture Recognition

There are different algorithms can be used to recognize the hand gesture such as HMM, Decision Tree, SVM, and k-NN classifier algorithms [3] [4] [15] [16] [18]. These algorithms will be discussed in this section.

### 2.4.1 Hidden Markov Model (HMM) Algorithm

Hidden Markov Model (HMM) algorithm is a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved states [31]. In an ordinary Markov model, the state is directly visible to the observer. The state transition probabilities are the only parameters for the Markov model. In a hidden Markov model, the state is not visible but its output is dependent on the state. The output generated by HMM provides the information about the sequence of the state [25].

An HMM consists of a set of N states: $S = \{S_0, S_1, S_2, \dots, S_n\}$ with the state transition probability matrix, A and the output probability matrix, B. It also included a set of M distinct observation symbols: $O = \{O_0, O_1, O_2, \dots, O_m\}$. The observation symbol corresponds to the physical output of the system being modeled. The transition

probability matrix, A={ $a_{ij}$ } is an $N\ x\ N$ matrix for the state transition probability distribution where $a_{ij}$ is the probability of the transition from state $S_i$ to state $S_j$. The output probability matrix, B = { $b_j$(x) } is an $N\ x\ M$ matrix for the observation symbol probability distributions where $b_j$(x) is the probability of emitting $O_x$ at time t in state $S_j$. A complete set of parameters of an HMM can be compactly expressed as λ = (A, B, π) [20].

Basically, there are three problems that must be solved for HMM to be useful in real application such as hand gesture recognition: evaluation, decoding, and learning. Given the observation sequence, O and a model, λ = (A, B, π), to efficiently evaluate the probability of given observation sequence, P( O | λ), the Forward-Backward algorithm is applied. To determine an optimal state sequence that best explains the observation, the Viterbi algorithm is used. Lastly, the Baum- Welch algorithm is used to adjust the model parameters, λ to maximize the probability, P( O | λ) [25].

### 2.4.2 Decision Tree Algorithm

A decision tree is a hierarchical tree structure consisting of a root node, branch nodes, and leaf nodes for classification. As shown in Figure 2.4, in the decision tree, the input sample data are initially put in the root node. The input sample data is then split into leaf node based on a series of rules about the attributes of classes in branch nodes, where each leaf node denotes a class. The input sample data will be allocated in one of the leaf nodes and the system will determine the attributes of the input sample based on the leaf node. Decision trees are simple to understand and interpret. Their ability for diverse information fusion is well suited for pattern classification with multiple features. They also take advantage of their sequential structure of branches so that the searching range

between classes for classification can be reduced rapidly. Decision trees are robust for good performance with large data in a short time [32], which is the significant advantage to realize real-time classification systems.

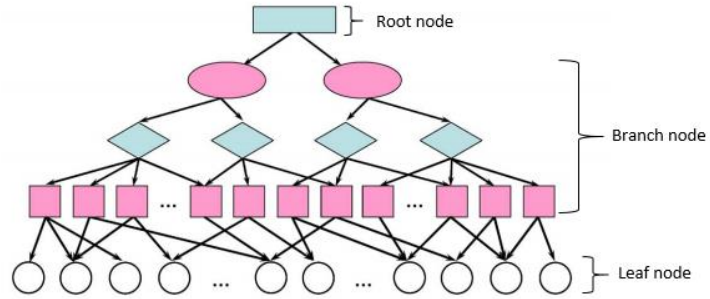


Figure 2.4: Structure of decision tree [33]

### 2.4.3 Support Vector Machines (SVM) Algorithm

Support Vector Machines (SVM) is a linear model algorithm. The idea of the algorithm is to find linear separation boundary between the class instances which correctly separates the instances into the classes. This boundary must have maximal margin distance to the closest training sample instances is the maximal one. Figure 2.5 shows an example of the Support Vector Machines algorithm used to classifies the sample [34]. From the graph in Figure 2.5, the boundary $H_3$ can be observed that it doesn't separate the two classes. Boundary $H_1$ does but there is a small margin. Hence, boundary $H_2$ is chosen which separates the instances with the maximum margin. Support Vector Machines can also work in non-linear space. This is achieved by transforming instances space into a new space and use the non-linear mapping between them.

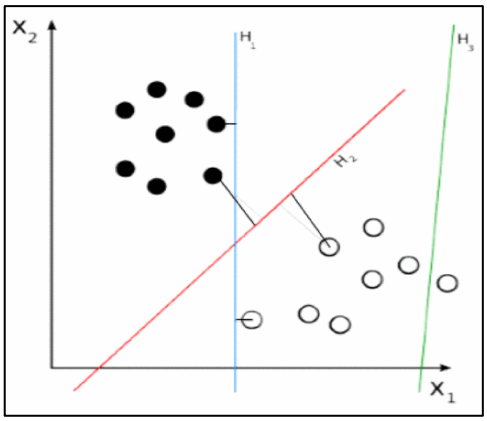


Figure 2.5: Support Vector Machines algorithm used to classifier the sample [34]

### 2.4.4 k-Nearest Neighbors (k-NN) Classifier Algorithm

The k-Nearest Neighbors (k-NN) classifier algorithm is a non-parametric method that can manage classification and regression problems. It is one of the simplest of machine learning algorithms due to its simplicity computational [24]. It is a density based classifier that classifies new unknown data based on its k-nearest known classes. The value of k is the number of the nearest neighbors involving in voting the new unknown data into its class. Figure 2.6 shows the k-Nearest Neighbour classifier algorithm used to determine the class of new unknown data [25]. Suppose there are two classes of elements, blue squares, and red triangles. A new element, the green circle is needed to put into one of these classes. The nearest neighbors and vote on what the newest element is. Let include 1 voting neighbor, the value of k is set to 1, the new example is classified as Class 1. However, the new example would be put in Class 2 if the value of k is set to 3. Two of three nearest neighbors vote for the new example to be put into Class 2.

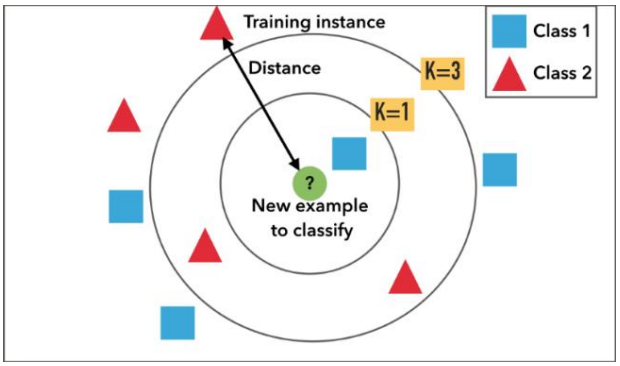


Figure 2.6: Example of k-Nearest Neighbour classifier algorithm used to determine the class of new unknown data [35]

In order determine the nearest neighbors among the training instance in a dataset, distance metrics plays a very critical role. The common distance metric to be used with K-NN classifier algorithm is Euclidean distance [36].

## 2.5 Distance Metric Algorithm Used in Hand Gesture Recognition

Distance metric algorithm is used to calculate the similarity between two instances, quantize their similarity into a distance value. Euclidean distance and Dynamic Time Warping (DTW) algorithms are discussed in this section.

### 2.5.1 Euclidean Distance Algorithm

Euclidean distance is used to calculate straight-line length between two instances which have the same dimensions. In Cartesian coordinates, if $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$ are two points in Euclidean n-space, then the distance between x and y is given by Pythagorean formula shown in Equation 2.6.

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \quad (2.6)$$

### 2.5.2 Dynamic Time Warping (DTW) Algorithm

Dynamic Time Warping (DTW) algorithm is a robust algorithm to find an optimal alignment between two given time dependent sequences that have differences in length or speed [26]. Assume there are two discrete warping sequence x and y having different sequence size of n and m: $x = \{x_0 , x_1 , x_2 , \dots , x_N\}$ and : $y = \{y_0 , y_1 , y_2 , \dots , y_M\}$. To align both sequences, a $N\ x\ M$ matrix is established and filled up with the distance metric, $d$ corresponds to each cell $(n, m)$, where $d(n, m) = (x_n - y_m)^2$. Then, the DTW algorithm finds a series of contiguous cells in which the smallest cumulative distances, D from the beginning cell $(1,1)$ are stored by using recursive the law as shown in Equation 2.7. The cumulative distances at the last cell $(N, M)$, $D(N, M)$ denoted the distance metric that measures the similarity between the sequences.

$$D(n, m) = d(n, m) + \min\{D(n-1, m) + \{D(n, m-1) + \{D(n-1, m-1)\} \quad (2.7)$$

## 2.6 Description of Hardware

In this section, the hardware that considered to be used in this project will be described. The Raspberry Pi board, GY-521 IMU sensor, Spectra Symbol 2.2 and 4.5 inches' flex sensors, and MCP 3008 ADC module are considered to be implemented to this project. The Raspberry Pi is used as the microcontroller of the system. The flex sensors will be used to detect the bending angle of finger whereas the IMU sensor will be used to detect the tilt angle of the hand. The ADC module will be used to convert the analogue reading of flex sensors to digital data that is readable by Raspberry Pi.

### 2.6.1 Hardware Description of Raspberry Pi

Raspberry Pi is considered to be used as a microcontroller in hand gesture recognition system. The Raspberry Pi is a small, powerful, and education-oriented computer board developed in the United Kingdom by the Raspberry Pi Foundation. The Raspberry Pi can be used as a standard PC by connecting with a monitor, a standard keyboard, and mouse. The Raspberry Pi hardware has evolved through several versions. To date, Raspberry Pi has various of models which comprise of Raspberry Pi 1 Model A, A+, B, Raspberry Pi 2 Model B, Raspberry Pi 3 Model B, Computer Model and Raspberry Pi Zero.

Raspberry Pi 1 Model B is considered to be used in this project. The Raspberry Pi 1 Model B has 512MB of RAM and support interfaces Integrated-Circuit (I2C), Serial Peripheral Interface(SPI), and Universal Asynchronous Receiver/Transmitter (UART). Raspberry Pi has a Broadcom BCM2835 system on chip, two USB ports, a SD card slot, 26 pins GPIO header exclude the external header, RCA Video Out port and 100 Mbps Ethernet port. The 26 pins GPIO header includes the pins of I2C, SPI, and UART.

The Raspberry Pi 1 Model B contains a single 26 pins expansion header labeled as ‘P1’ that contains seventeen GPIO pins, two 3.3 VDC power pins, two 5 VDC power pins, and four Ground pins. As shown in Figure 2.7, the Raspberry Pi’s P1 Header contains 2 pins for I2C, 2 pins for UART, and 5 pins for SPI. The GPIO pin numbering shown in Figure 2.7 is the Broadcom GPIO pin numbers. The five pins for SPI_0 communication of Raspberry Pi are pin header of P1_19, P1_21, P1_23, P1_24, and P1_26. The two pins for I2C_1 communication of Raspberry Pi are pin header of P1_3 and P1_5. The description of each SPI_0 pins is described in Table 2.1 while the description of each I2C_1 pins is described in Table 2.2.

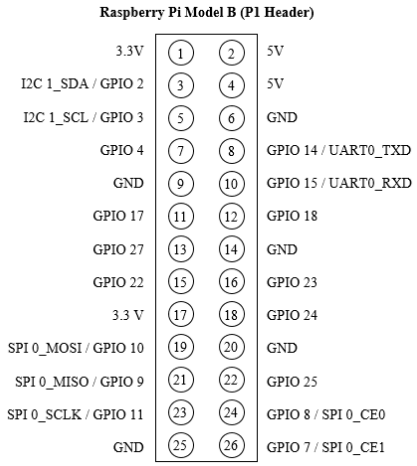


Figure 2.7: Pinout Diagram of P1 Header on Raspberry Pi 1 Model B

Table 2.1: Pin Description of SPI pins of Raspberry Pi

| Pin Header | Pin Name | Pin Description |
|---|---|---|
| P1_19 | SPI_MOSI | Master Output Slave Input (MOSI) pin is used in transfering data from Raspberry Pi that acts as master device to the slave device. |
| P1_21 | SPI_MISO | Master Input Slave Output (MISO) pin is used in transfering data from slave device to Raspberry Pi that acts as master device. |
| P1_23 | SPI_SCLK | Serial Clock (SCLK) pin is used in sending the clock signal generated from Raspberry Pi to synchronize all data transfers over the SPI bus. |
| P1_24 | SPI_CE0 | Chip Enable (CE) pin is often called Chip select. CE pin is used to enable the slave device to communicate with Raspberry Pi. SPI_0 supports 3 Chip selects but there are only 2 pins ( CE0 and CE1) available on header. These pins function is to select which slave device can communicate with Raspberry Pi. |
| P1_26 | SPI_CE1 | |

Table 2.2 : Pin Description of I2C pins of Raspberry Pi

| Pin Header | Pin Name | Pin Description |
|---|---|---|
| P1_3 | I2C_SDA | Serial Data Line (SDA) pin is used for both Raspberry Pi and slave to transfer data over the I2C bus. |
| P1_5 | I2C_SCL | Serial Clock Line (SCL) pin is used for synchronizing all data transfers over the I2C bus. |

### 2.6.2 Hardware Description of GY-521 IMU Sensor

GY-521 Inertial Measurement Unit (IMU) sensor as shown in Figure 2.8 is a 6 axis IMU sensor which consists of a MPU-6050 MotionTracking chip and a low-dropout or (LDO) regulator. It supports the operating voltage from 3VDC to 5VDC. The MPU-6050 chip combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor (DMP) in a small package. The gyroscope in MPU-6050 can provide the digital output of triple-axis angular rate with a user-programmable full-scale range of ±250, ±500, ±1000, and ±2000°/sec. The accelerometer in MPU-6050 can provide a digital output of triple-axis acceleration with a programmable full-scale range of ±2g, ±4g, ±8g, and ±16g. The DMP in MPU-6050 is used in 3D Motion Processing and gesture recognition. The MPU-6050 chip only supports I2C interfacing. The pin description of GY-521 module is shown in Table 2.3.

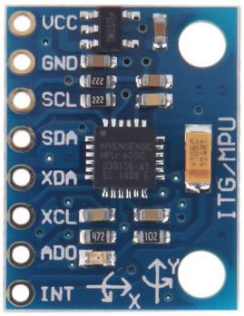


Figure 2.8: GY-521 IMU sensor

Table 2.3: Pin Description of GY-521 IMU sensor

| Pin Name | Pin Description |
|---|---|
| SCL | Serial Clock Line (SCL) pin is used in synchronizing all data transfers over the I2C bus. |
| SDA | Serial Data Line (SDA) pin is used in transfering data over the I2C bus. |

| | |
|---|---|
| XDA | Auxiliary Serial Data Line (XDA) pin is the pin in the auxiliary I2C bus and used in trasfering data over the I2C bus. Normal I2C bus has it's own I2C controller to be a master on a second I2C bus. It uses the pins XDA and XCL for the second I2C-bus. |
| XCL | Auxiliary Serial Clock Line (XDA) pin is the pin in the auxiliary I2C bus and used in synchronizing all data transfers over the I2C bus. |
| AD0 | Address (AD0) pin is usd in selection of I2C slave address. The I2C slave address is 0x68 when AD0 pin is LOW (0V). The I2C slave address is 0x69 when AD0 pin is HIGH (3.3V). |
| INT | Interrupt (INT) pin is used in transferring interrupt signal. The raw values from the accelerometer and gyroscope can be programmed to be placed in the 1024byte First In First Out (FIFO) buffer and read by certain microcontroller. GY-521 module will signal the microcontroller using the interrupt signal so the microcontroller knows that there is data in the FIFO buffer waiting to be read. |

The MPU-6050 chip of GY-521 sensor has a number of registers which have different functionality. Table 2.4 shows the function of some registers in MPU-6050 with their hexadecimal address.

Table 2.4: Function and hexadecimal address of some registers in MPU-6050

| Register Name | Address (Hex) | Function |
|---|---|---|
| ACCEL_XOUT_H | 3B | High byte of accelerometer X-axis data. |
| ACCEL_XOUT_L | 3C | Low byte of accelerometer X-axis data. |
| ACCEL_YOUT_H | 3D | High byte of accelerometer Y-axis data. |
| ACCEL_YOUT_L | 3E | Low byte of accelerometer Y-axis data. |

| | | |
|---|---|---|
| ACCEL_ZOUT_H | 3F | High byte of accelerometer Z-axis data. |
| ACCEL_ZOUT_L | 40 | Low byte of accelerometer Z-axis data. |
| GYRO_XOUT_H | 43 | High byte of the X-axis gyroscope output. |
| GYRO_XOUT_L | 44 | Low byte of the X-axis gyroscope output. |
| GYRO_YOUT_H | 45 | High byte of the Y-axis gyroscope output. |
| GYRO_YOUT_L | 46 | Low byte of the X-axis gyroscope output. |
| GYRO_ZOUT_H | 47 | High byte of the Z-axis gyroscope output. |
| GYRO_ZOUT_L | 48 | Low byte of the X-axis gyroscope output. |
| PWR_MGMT_1 | 6B | Power Management register that contain 7 bits and each bit has specific function.<br><br>Bit [7]: Reset the internal registers and restore default setting when set.<br><br>Bit [6]: The chip is set to sleep mode when set.<br><br>Bit [5]: The chip the chip will cycle between sleep and taking a sample at a rate<br><br>Bit [4]: When set, the gyro drive and PLL circuitry are enabled, but the sense paths are disabled.<br><br>Bit [3]: Temperature sensor is disabled when set.<br><br>Bit [2:0]: Selection of clock source<br><br>00000000 bytes written will wake MPU-6050 up as it starts in sleep mode and enable the measurement. |