

**UNSUPERVISED LEARNING: K-MEANS APPROACH IN CLASSIFYING  
HUMAN WALKING PATH**

**MUHAMMAD SYAHIR BIN SHAIDAN**

**A report submitted in partial fulfillment of the requirements for the degree of  
Mechatronics Engineering**



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA


**Faculty of Electrical Engineering**

**UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

**2016**

“I hereby declare that I have read through this report entitle “Unsupervised Learning: K-Means Approach in Classifying Human Walking Path” and found that it has comply the partial fulfilment for awarding the degree of Bachelor of Mechatronics Engineering”

Signature

  
.....

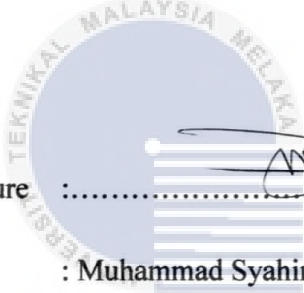

Supervisor's Name : Miss Nur Maisarah Binti Mohd Sobran

Date

اونيومر سیتی تیکنیکل ملیسيا ملاک : 23/6/2016

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

I declare that this report entitle “Unsupervised Learning: K-Means Approach in Classifying Human Walking Path” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

   
Signature : .....  
Name : Muhammad Syahir Bin Shaidan  
Date : 23/6/2016  
اونيورسيتي تيكنيكل ماليسيا ملاك  
UNIVERSITI TEKNIKAL MALAYSIA MELAKA



## ACKNOWLEDGEMENT

Upon the completion of this report, there are many works involved that need the help of various people. They have guided me thoroughly upon understanding the concept of the project, the essence of knowledge provided will not be at waste. Particularly, I would like to express my biggest thanks to my main project supervisor, Miss Nur Maisarah Binti Mohd Sobran of her guidance for the project, suggesting ideas, help in understanding project purpose and concept and thoroughly guide me upon this whole project. Without her guidance, the project might unable to be implementing well and may not be able to succeed.

I am also very thankful to UTeM administration especially FKE administration for the funding of material of the project, helping much in implementing the hardware part of the project.

Besides that, thanks to my fellow course mates that help giving friendly advice, suggestions, for me to complete the system well and help me on understanding some of the confusing part of the project.

## ABSTRACT

Along the decade, researchers have proposed some various methods to represents the human walking behaviors in either individual or in crowd environment. Human walking behaviors can be divided into several viewpoints such as route selection, navigation, and path finding. Although human walking behavior is unpredictable and has dynamic characteristics, these viewpoints are the most dominant behavior that human always considered when they are walking. Furthermore, many researchers have widen their scope of investigation as they are more focusing in human walking behaviors under panic situations that usually in crowded conditions due to natural disaster or terrorist attacks. In this project, a method to analyze human walking data is proposed and human walking data will be analysed by classifying the path taken by human while walking whether right, left of a forward path. However in order to collect human walking data, an experiment is set up which involves five different subjects with different gender, height and weight considered. In the experiment, the subject will wear a wearable device containing inertial measurement unit around their waist and walks along a created path. Inertial measurement unit is used to determine the orientation and position of human while they are walking. The data considered from inertial measurement unit is Yaw, Pitch, and Roll data. The human walking data will be classified by using unsupervised learning method because human walking data are unclassified data and the results of the analysis are cannot be predicted. K-means clustering method is used to classified human walking activity. The number of cluster will be determined by using K-means clustering. After that, the performance measurement of K-means clustering is carried out to evaluate the performance of K-means clustering. Silhouette Coefficient method is used for this purpose. The validity of number of cluster for clustering human walking activity is determined. As conclusion, the number of cluster that is suitable to classified human walking data is three.

## ABSTRAK

Sepanjang dekad ini, penyelidik telah mencadangkan beberapa pelbagai kaedah untuk mewakili tingkah laku berjalan kaki manusia sama ada individu atau dalam persekitaran orang ramai. Tingkah laku yang berjalan kaki manusia boleh dibahagikan kepada beberapa pandangan seperti pemilihan laluan, navigasi, dan laluan dapatan. Walaupun tingkah laku berjalan manusia adalah tidak menentu dan mempunyai ciri-ciri dinamik, pandangan ini adalah tingkah laku yang paling dominan manusia yang sentiasa dipertimbangkan apabila mereka berjalan. Tambahan pula, ramai penyelidik telah memperluaskan skop siasatan kerana mereka lebih memberi tumpuan dalam tingkah laku berjalan kaki manusia di bawah keadaan panik yang biasanya dalam keadaan sesak disebabkan oleh bencana alam atau serangan pengganas. Dalam projek ini, kaedah untuk menklasifikasikan data manusia berjalan telah dicadangkan dan data berjalan kaki manusia akan dianalisis dengan mengklasifikasikan jalan yang diambil oleh manusia ketika berjalan sama ada kanan, kiri jalan ke hadapan. Walau bagaimanapun dalam usaha untuk mengumpul data berjalan kaki manusia, eksperimen ditubuhkan yang melibatkan lima mata subjek yang berbeza dengan jantina berbeza, ketinggian dan berat badan dipertimbangkan. Dalam eksperimen ini, subjek menggunakan peranti yang mengandungi unit pengukuran inersia di pinggang mereka dan berjalan di sepanjang jalan yang diwujudkan. Unit pengukuran inersia digunakan untuk menentukan orientasi dan kedudukan manusia sementara mereka berjalan. Data yang berjalan kaki manusia akan diklasifikasikan dengan menggunakan kaedah pembelajaran tanpa pengawasan kerana data berjalan manusia adalah data tidak dikelaskan dan keputusan analisis tidak dapat diramalkan. cara kaedah pengelompokan K digunakan untuk aktiviti berjalan kaki manusia terperingkat. Bilangan kluster akan ditentukan dengan menggunakan cara berkelompok K. Selepas itu, pengukuran prestasi K-cara kelompok dijalankan untuk menilai prestasi kelompok K. Kaedah Pekali Bayang digunakan untuk tujuan ini. Kesahihan beberapa kelompok untuk kelompok aktiviti berjalan kaki manusia ditentukan. Kesimpulannya, bilangan kelompok yang sesuai untuk terperingkat data berjalan manusia adalah tiga.

## TABLE OF CONTENT

<b>CHAPTER</b>	<b>TITLE</b>	<b>PAGE</b>
	<b>ACKNOWLEDGEMENT</b>	<b>i</b>
	<b>ABSTRACT</b>	<b>ii</b>
	<b>TABLE OF CONTENTS</b>	<b>iv</b>
	<b>LIST OF TABLE</b>	<b>viii</b>
	<b>LIST OF FIGURES</b>	<b>ix</b>
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Introduction	1
	1.2 Project Research Background	1
	1.3 Motivation	2
	1.4 Problem Statement	4
	1.5 Objectives	4
	1.6 Scope	5
<b>2</b>	<b>PROJECT BACKGROUND &amp; LITERATURE REVIEW</b>	
	2.1 Introduction	6
	2.2 Project Background	6
	2.2.1 Supervised Learning	6
	2.2.2 Unsupervised learning	9
	2.3 Differences between Supervised Learning and Unsupervised Learning	11
	2.4 Previous Research on Human Walking Path Analysis	12
	2.4.1 An Approach for Extraction for Extraction of Human Walking Path in Intelligent Space – Hiromu	12



	Kobayashi, Hideki Hashimoto and Mihoko Niitsuma [10]	
	2.4.2 Multilinear Decomposition of Human Walking Paths – Chris A. Ramirez, Mario Castelan and Gustavo Arechavaleta [11]	14
	2.4.3 A Genetic Fuzzy System to Model Pedestrian Walking Path In A Built Environment – P.L.Chee, N. Saeid, C. Douglas, N. Mojdeh [14]	18
	2.4.4 Human-Observation-Based Extraction of Path Patterns for Mobile Robot Navigation – Takeshi Sasaki, Drazen Brscic and Hideki Mashimoto [15]	21
	2.4.5 Abstracting People’s Trajectories for Social Robots to Proactively Approach Customers – Takayuki Kanda, Dylan F. Glass, Masahiro Siomi and Narihiro Hagita [16]	24
	2.5 Summary	27
	2.6 Method to cluster human walking data	29
<b>3</b>	<b>METHODOLOGY</b>	
	3.1 Project methodology to achieve first objective	31
	3.1.1 Hardware selection	32
	3.1.1.1 Arduino Nano	32
	3.1.1.2 XBee® Zigbee RF module	32
	3.1.1.3 Inertial Measurement Unit (IMU)	33
	3.1.2 Early calibration of MPU6050	33
	3.1.3 Built-in Arduino code to Determine Euler Angle and Yaw, Pitch and Roll of a path	34
	3.1.4 Data collection of human walking activity	34
	3.1.5 Construction of walking path for human walking activity	35
	3.2 Project methodology to achieve second objective	36

3.2.1	Method selection	36
3.2.1.1	K-means clustering approach on human walking analysis	36
3.2.1.2	Algorithm for K-means clustering	39
3.2.1.3	Application of K-means clustering in MATLAB	40
3.2.2	Analysis on clustering of human walking data	41
3.3	Methodology for achieving third objective	41
3.3.1	Silhouette Coefficient method in selecting number of cluster	42
3.3.2	Analysis on performance of K-means clustering of human walking data	42
<b>4</b>	<b>RESULT AND DISCUSSION</b>	
4.1	Analysis of data collection on human walking data	43
4.1.1	Analysis of Yaw, Pitch and Roll data	43
4.1.1.1	Analysis of Yaw data	43
4.1.1.2	Analysis of Pitch data	45
4.1.1.3	Analysis of roll data	46
4.2	Analysis on K-means clustering of human walking data	47
4.2.1	Analysis on K-means clustering when $k = 2$	47
4.2.2	Analysis on K-means clustering when $k = 3$	48
4.2.3	Analysis on K-means clustering when $k = 4$	50
4.2.4	Analysis on K-means clustering when $k = 5$	51
4.2.5	Analysis on K-means clustering when $k = 6$	52
4.3	Analysis on performance of K-means clustering of human walking data	53
4.3.1	Analysis on Silhouette Coefficient value in selecting number of $k$	53
4.3.1.1	Analysis for $k = 2$ with average Silhouette value	53

4.3.1.2 Analysis for $k = 3$ with average Silhouette value	55
4.3.1.3 Analysis for $k = 4$ with average Silhouette value	56
4.3.1.4 Analysis for $k = 5$ with average Silhouette value	57
4.3.1.5 Analysis for $k = 6$ with average Silhouette value	58
4.3.2 Conclusion from the analysis	59
<b>5 CONCLUSION AND RECOMMENDATIONS</b>	
5.1 Conclusion	60
5.2 Recommendation	60
<b>REFERENCES</b>	<b>62</b>
<b>APPENDICES</b>	<b>64</b>



## LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Difference between Supervised and Unsupervised.	11
2.2	Comparison of clustering method used and performance measurement.	27
2.3	Differences between K-means clustering and Hierarchical Clustering [18]	29



## LIST OF FIGURES

FIGURE	TITLE	PAGE
1.1	Steps in Knowledge Discovering of Database [4]	3
2.1	Structure of Neural Networks [4].	7
2.2	A good separation of data [7]	8
2.3	Examples of decision tree concept [8]	8
2.4	Two types of hierarchical clustering representation [9]	10
2.5	Examples of K-means Clustering [9].	10
2.6	DBSCAN clustering [9].	11
2.7	Intelligent Space concept [10].	12
2.8	Ultrasound receivers which are installed to ceiling and ultrasound transmitter which is mounted to student card strap [10]	13
2.9	Experimental environment with obtained path [10].	13
2.10	Experimental result [10]	14
2.11	Mode n-matrices and flattening of tensor [14].	15
2.12	The routes of database [14]	15
2.13	Four types of subspace analysis [14]	16
2.14	The (x, y)-coordinates clustering [14].	17
2.15	Clustering of routes [15]	17
2.16	Different shapes of routes based on subject's intentions [14]	18
2.17	Spatial representation of environment within human's field of view [15]	19
2.18	Architecture of steering fuzzy model [15]	19
2.19	Result of Experiment 1 [15]	20
2.2	Result of Experiment 2 [15]	21
2.21	Result of Experiment 3 [15]	21

2.22	iSpace configuration [16].	22
2.23	Experimental environment for extraction of human walking [16]	22
2.24	Human walking path obtained [16].	23
2.25	Results of clustering [16].	23
2.26	Placement of SICK LMS-200 laser range finders [17].	24
2.27	<i>Style</i> category [17]	25
2.28	State chain model of global behavior [17]	26
2.29	Patterns of global behavior [17].	27
3.1	Overall methodology of the project	30
3.2	Basic operation of the hardware	31
3.3	Arduino Nano board	32
3.4	Xbee RF module	32
3.5	Inertial measurement unit	33
3.6	Yaw, Pitch, and Roll angles of human body.	35
3.7	Hardware to collect human walking data	35
3.8	Path for human walking activity	36
3.9	Flowchart of K-means clustering process	37
3.1	Basic K-means algorithm code	39
3.11	K-means algorithm code for MATLAB	40
4.1	Graph of Yaw data of each subject	44
4.2	Graph of Pitch data from each subject	45
4.3	Graph of Roll data of each subject	46
4.4	Graph of human walking data when k=2	48
4.5	Graph of human walking data when k=3	49
4.6	Graph of human walking data when k = 4	50
4.7	Graph of human walking data when k = 5	51
4.8	Graphs of human walking data when k = 6	53
4.9	Graphs of Silhouette plot for k = 2	54
4.1	Graphs of Silhouette for k = 3	55
4.11	Graphs of Silhouette for k = 4	56
4.12	Graphs of Silhouette for k = 5	58



## CHAPTER 1

### INTRODUCTION

#### 1.1 Introduction

The main goal of this project is to develop a method to classify an unclassified human walking data by using unsupervised learning and to developed experimental setups to get the human walking data. This chapter will cover the research background, motivation, problem statement, objectives and scope of the research. A further explanation also will be included here to provide understanding for the user about this project.

#### 1.2 Project Research Background

Along the decade, researchers have proposed some various methods to represents the human walking behaviors in either individual or in crowd environment. Human walking behaviors can be divided into several viewpoints such as route selection, navigation, and path finding [11]. Although human walking behavior is unpredictable and has dynamic characteristics, these viewpoints are the most dominant behavior that human always considered when they are walking [6]. Furthermore, many researchers have widen their scope of investigation as they are more focusing in human walking behaviors under panic situations that usually in crowded conditions due to natural disaster or terrorist attacks [7-9]. Zheng et al. [7], has proposed seven methods for crowd evacuations that also includes the important aspects for human walking behaviors which cover heterogeneity, scale of modeling, condition, space, and time steps.



In this project, human walking data will be analysed by classifying the path taken by human while walking whether right, left or a forward path. The human walking data will be classified by using unsupervised learning method because human walking data are unclassified data and the results of the analysis are cannot be predicted. Therefore, unsupervised learning method is very suitable as it is used to clustering unclassified data and the results of clustering are varies according to the users. Moreover, it is used for large number of dataset as human walking data consist of a large number of data.

### 1.3 Motivation

Knowledge Discovery in Database can be defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. It is an interactive and iterative process involving several steps such as selection, pre-processing, transformation, data mining and interpretation of data. Knowledge in data is very important as the knowledge will help to identify and interpret the data which can be used later in the future.

As reported by CIO Magazine's cover story in May 1998 [1], about 13 million customers will contact the customer service call centre of Bank of America in order to listen to the marketing advertisement. Therefore, Bank of America conclude that the customer like to be informed of new product or services. So in this case, knowledge discovery in database is important as they help to identify the type of marketing approach for certain customer based on their profile.

Besides that, nearly 3000 cases per year involving brain tumors among children have been reported in United States. Among all the cases, nearly half of them are considered fatal. According to Director of brain tumor research at Children's Memorial Hospital in Chicago; Eric Bremer [2], they have set a goal to create a database of gene expression of the patients, in order to give them a better treatment. Therefore, they used software of data mining which is Clementine data mining software; developed by SPSS, Inc. to classify the types of tumor into 12 or so cluster types.

In November 2002, former President of United States of America, President Bill Clinton spoke at Democratic Leadership Council [3], mentioned that after the event of 11 September 2001, the Federal Bureau of Investigation (FBI) has received a great amount of data considering five terrorist related to the incident. As stated from the data, one of the terrorist possessed 30 credit cards with some combined balances and stay in the country for almost 2 years. Moreover, President Bill Clinton concluded that further investigations are needed to gain some information for the data.

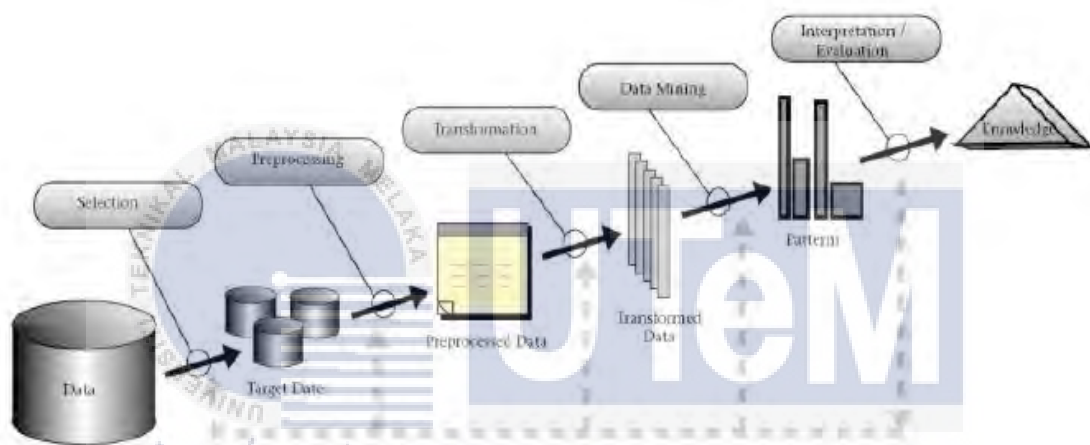


Figure 1.1: Steps in Knowledge Discovering of Database [4].

## 1.4 Problem Statements

In achieving this project, there are several problems that have been encountered. First of all, the problems face is to select the most suitable clustering method that need to be used to classify the human walking data. The method used need to accurately classify the human walking data as many data are involved and the data have different range of value. Therefore, unsuitable clustering method will result inaccurately classification of data.

Next, the experimental setup to collecting the human walking data is needs to take serious part in succeeding the project. The experimental setup has able to collect the position and orientation of human while walking. Moreover, every aspect for the experiments such as the sensor used and the walking path have to be considered. The sensor need to detect and track the position which means it has to accurately identify the position and orientation of human. Therefore, a single sensor cannot achieve this accurate data and sensor fusion process has to take place. The sensor fusion also can correct the deficiencies of the single sensor.

The height of the human must also be considered in running this project. The sensor will be placed on the human body at their waist in order to examine the positon and orientation of human. Different height of human will result the different height position of waist of a person. If the person is taller, this means that the angle of elevation will be different from a short person as higher position will has a bigger angle of elevation.

## 1.5 Objectives

The objectives of this project are:

- 1) To develop an experimental setup to collect human walking data.
- 2) To develop a method to classify an unclassified human walking data by using unsupervised learning.
- 3) To evaluate the performance of the k-means clustering by using Silhouette Coefficient method.

## 1.6 Scope

The scope below should be emphasized in order to achieve the objectives of the project.

- 1) The sensors have to be able to identify the position and orientation of human whenever in static or walking motions. This means that the sensors have gyroscopic features and can compute data from X-Y-Z plane.
- 2) The method must be able to classify human walking data as there are many data produced during the experiment.
- 3) The performance of k-means clustering will be discussed in term of number of suitable k clusters.



## CHAPTER 2

### PROJECT BACKGROUND & LITERATURE REVIEW

#### 2.1 Introduction

In this chapter, the review of related previous research project will be discussed. The information obtained will become some useful source that can be used as references in order to make this project successful. Other than that, the obtained information will be synthesized to some literature review by integrating the information to evaluate them. Therefore, this chapter will discuss about the literature review.

#### 2.2 Project Background

Machine learning technique is very popular technique nowadays as it is used to train computer to explore and study the algorithm that can be learn and interpret some data. Machine learning will gives computer the ability to learn and improving performance without being programmed [5]. There are two types of machine learning; supervised learning and unsupervised learning. These two types have different functions and different characteristics which can be applied according to certain condition.

##### 2.2.1 Supervised Learning

In supervised learning, the training data will includes both the input and the desired result. In other words, the correct results or target are known first and the input to the model are given during the learning process. The supervised learning will find a way to build a model

that can predict the response value of dataset. This method is popular because it gives fast and accurate results. Moreover, supervised learning are able to generalize the data which means that it is able to give reasonable outputs even new inputs are given without knowing a proper target. There are two categories of algorithm in supervised learning which are classification and regression. Classification is used to assign membership of group for data samples. There are two ways to assign new values to given class. First by using crisp classification method; which the classifier will return the input label. Second by using probabilistic classification method; which the classifier will return the input probabilities to the belonging class.

Moreover, there are three tools of supervised learning; neural networks, support vector machine, and decision tree [6]. Neural Networks refer to Artificial Neural Network (ANN) that consists of a set of interconnected simple processing units (neurons or nodes) which combine to output a signal to solve a certain problem based on the input signals it received [7]. The neural network also inspired by human brain which contains highly amount of connected neurons. Figure below shows the graphical structure of neural network.

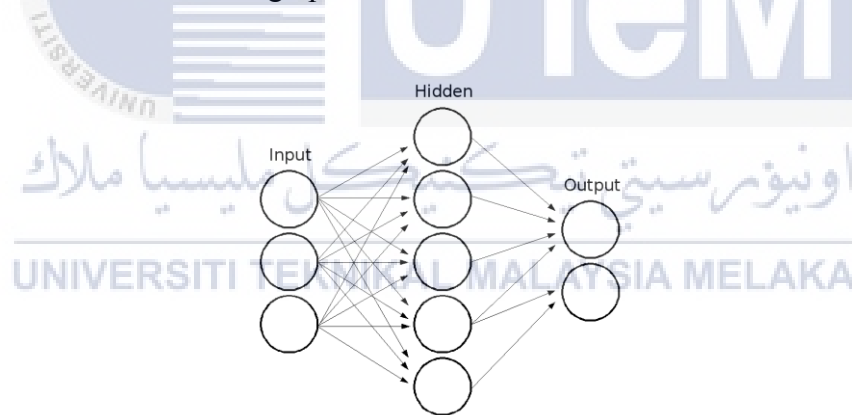


Figure 2.1: Structure of Neural Networks [4].

A Neural Network is usually contains three layers which is an input layer, a hidden layer, and an output layer. The main property of neural networks is that it has adjustable gains that are slowly adjusted through iterations influenced by the input-output patterns.

Support vector machine (SVM) is a method used for classification, regression and outlier detection. It means that the data will be analyzed and the pattern of data will be recognized by mapping vectors of input feature to higher dimensional space [7]. This method

uses subset of training to make decision; therefore it is also a memory efficient. A standard support vector machine is non-probabilistic linear classifier that categorizes the new input. A good separation is achieved if the hyperplane which has largest distance is nearer to training point to minimize the generalization error [7]. Figure 2.2 shows the good separation of data by using support vector machine method.

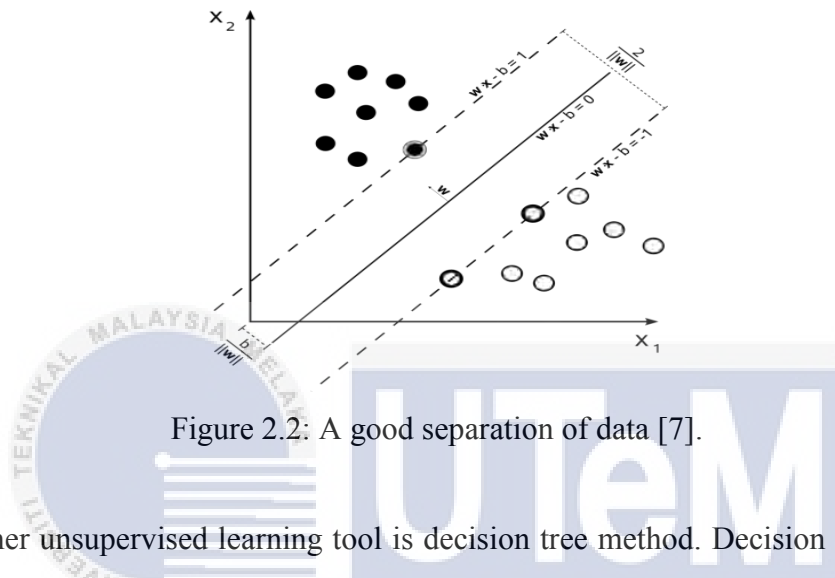


Figure 2.2: A good separation of data [7].

Another unsupervised learning tool is decision tree method. Decision tree method is a tree-like model used to approximating discrete-valued function that is robust to noisy data and able to learn functions. This method is very popular as it has been successfully applied to a broad range of tasks from learning to diagnose medical cases. As an example, decision tree learning has been applied to classify medical patients by their disease and equipment malfunction by their cause which generally the task is to classify the data into a set of possible categories. Figure 2.3 shows the example of decision tree method.

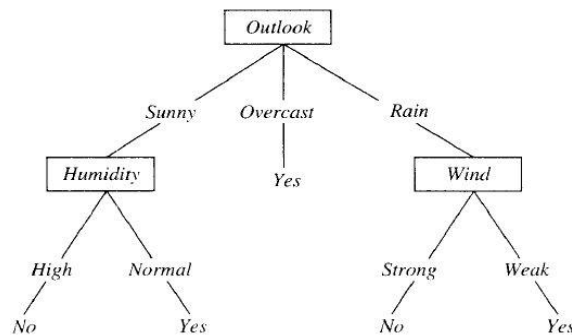


Figure 2.3: Examples of decision tree concept [8]

### 2.2.2 Unsupervised Learning

In unsupervised learning, the training data is not provided with the predicted results during the training. In other words, the results of data of unsupervised learning will vary according to user need. This means that there will no incorrect results when unsupervised learning method is used. This method is different from supervised learning method as it is not allow for prediction of result compared to supervised data which the result for the training is provided and can be predicted [6]. Moreover, unsupervised learning can be used to cluster the input data according to classes based on their statistical properties only. Even though this method does not require labels for clustering, the labelling can be carried out even if they are only available for small number of objects in desired class.

Clustering can be defined as a process of dividing a set of data or objects into a set of sub-classes, called clusters. There are two types of clustering in unsupervised learning; hierarchical clustering and partitional clustering [6]. A hierarchical clustering is a set of nested sub-classes or clusters that are organized as a tree which each cluster in the tree is the combination of its sub-clusters, and the root is cluster containing all objects [9].

In hierarchical clustering, it will always finds successive cluster by previously refers to actualized cluster. Hierarchical clustering is generated based on two basic approaches; agglomerative and divisive. The difference between agglomerative hierarchical clustering technique and divisive hierarchical clustering technique is that the agglomerative will start with points of each individual element in separate cluster merging with closest cluster pairs while divisive start with single cluster which then be splatted until clusters of individual points are remain. Figure 2.4 shows the graphical representation of hierarchical clustering.



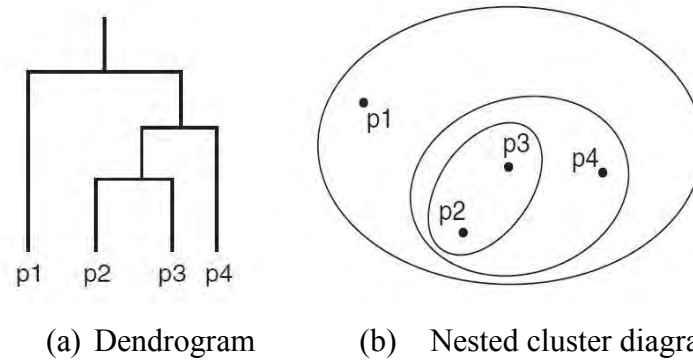


Figure 2.4: Two types of hierarchical clustering representation [9].

Partitional clustering is referring to a subset which contains each division of set of data objects that transformed into non-overlapping clusters. This clustering method usually determines all clusters at ones. There are two approaches in generating partitional clustering; K-means clustering and DBSCAN clustering [9]. K-means clustering is a clustering method which separates the data into number of  $k$  clusters, based on their characteristics. Each cluster will be represented by the centroid; center of cluster and each nearest point to the centroid will be assigned in that particular cluster. The main purpose of K-means clustering method is to minimize the distance between cluster centroids with corresponding data. Figure 2.5 below shows the examples of k-means clustering with four number of iteration.

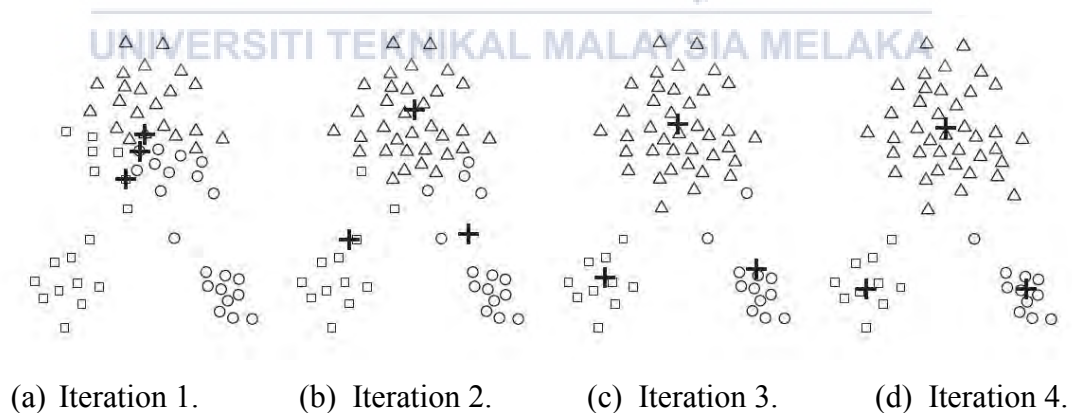


Figure 2.5: Examples of K-means Clustering [9].

DBSCAN or density-based clustering algorithm is a partitional clustering that automatically determines the number of cluster of data. However, this clustering method is

considered as incomplete clustering because it only considered points in high and medium density only as data while points in low-density are considered as noise and omitted.

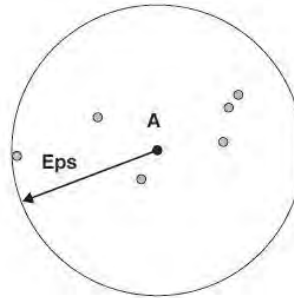


Figure 2.6: DBSCAN clustering [9].

### 2.3 Differences between Supervised Learning and Unsupervised Learning

Table 2.1: Difference between Supervised and Unsupervised.

Supervised Learning	Unsupervised Learning
1. Training data includes both the input and the desired result during the training.	1. Training data is not provided with the predicted results during the training.
2. Find a way to build a model that can predict the response value of dataset.	2. Not allow for prediction of result.
3. Able to generalize the data even new inputs are given without knowing a proper target	3. Used to cluster the input data according to classes based on their statistical properties only.
4. Tools: <ul style="list-style-type: none"> <li>• Neural Network</li> <li>• Support Vector Machine</li> <li>• Decision Tree</li> </ul>	4. Tools: <ul style="list-style-type: none"> <li>• Agglomerative hierarchical clustering</li> <li>• Divisive hierarchical clustering</li> <li>• K-means clustering</li> <li>• DBSCAN</li> </ul>

## 2.4 Previous Research on Human Walking Path Analysis

### 2.4.1 An Approach for Extraction for Extraction of Human Walking Path in Intelligent Space – Hiromu Kobayashi, Hideki Hashimoto and Mihoko Niitsuma [10].

This journal explains on the method to extract human walking path by using Distributed Intelligent Network Device (DIND) and method to classify human walking path data. Distributed Intelligent Network Device is a fundamental of Intelligence Space (iSpace) which consists of sensors, processors and network devices. iSpace can perceive and understand events in whole space by communicating with each DIND. Figure 2.7 shows the concept of Intelligent Space.

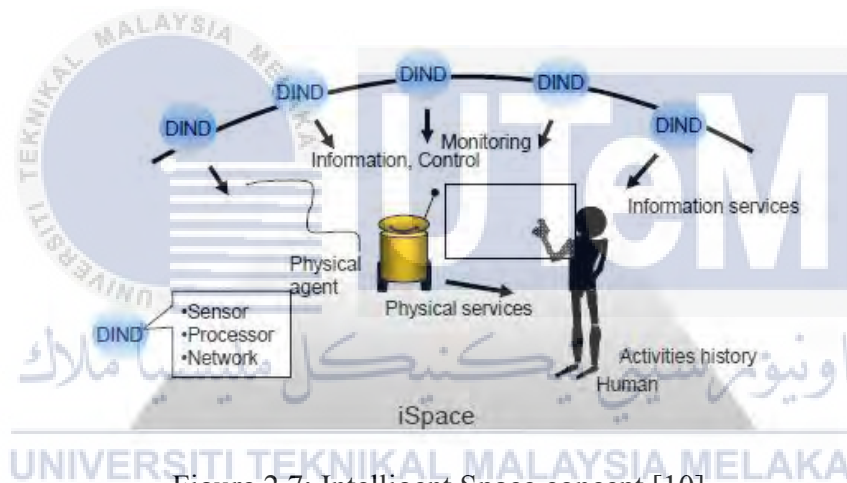


Figure 2.7: Intelligent Space concept [10].

During experiment, the sensor used is ultrasonic positioning system; which is divided into ultrasound receiver and ultrasound transmitter. During experiment, the subject will wear the transmitter which is mounted into student card strap, while the receiver is installed on the ceiling. The transmitter will regard the position of the subject by sending the signal to the receiver. 110 walking paths are obtained with consists of 5,481 points. Figure 2.8 shows the method on collecting human walking data and Figure 2.9 shows the path of subject resulting from the experiment.

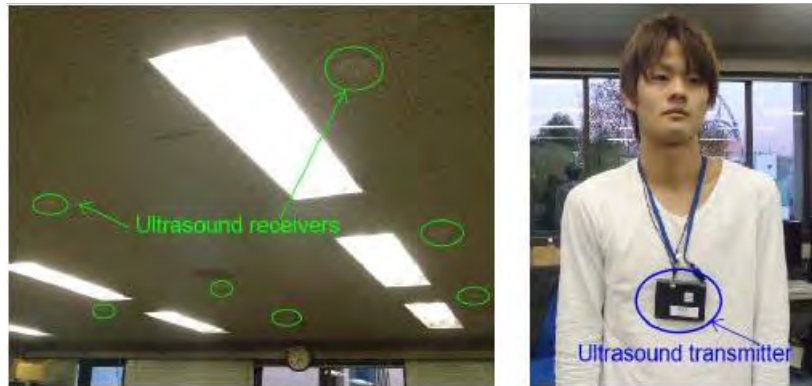


Figure 2.8: Ultrasound receivers which are installed to ceiling and ultrasound transmitter which is mounted to student card strap [10].

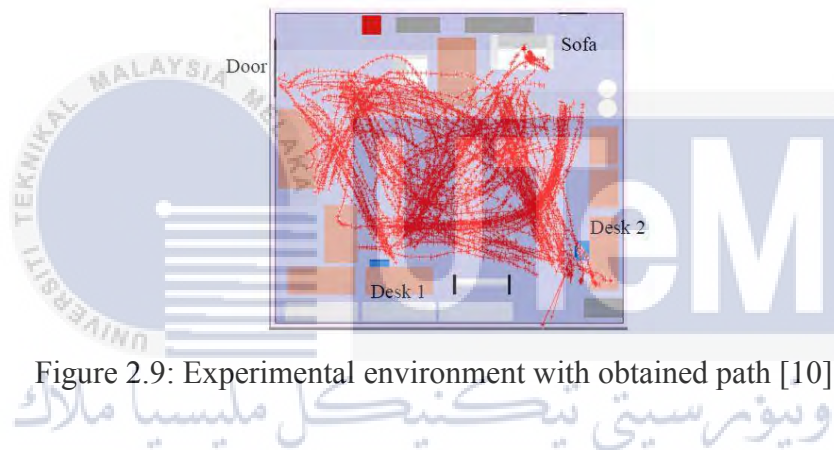


Figure 2.9: Experimental environment with obtained path [10].

As the human walking path data is not robust as is effected by human's stopped state, data smoothing method is used. In this paper, researchers used Generalized Cross-Validation (GCV) [11] based on empirical Bayes [12] approach in order to choose trade off parameter  $\lambda$ . After that, the similarity is calculated to set an appropriate threshold to the dataset and to gain same information about the trajectories beforehand. In this process, Angular Metrics for Shape Similarity (AMSS) [13] is applied to compare similarities between two trajectories. During AMSS, Dynamic Programming is applied to maximize the total similarities.

The next is clustering the data. In this process, a hierarchical clustering method is used because the number of trajectories does not have to be specified. In hierarchical clustering, the medoid; trajectory with average minimum distance of same cluster is extracted and then clarify as representative trajectory in each cluster.

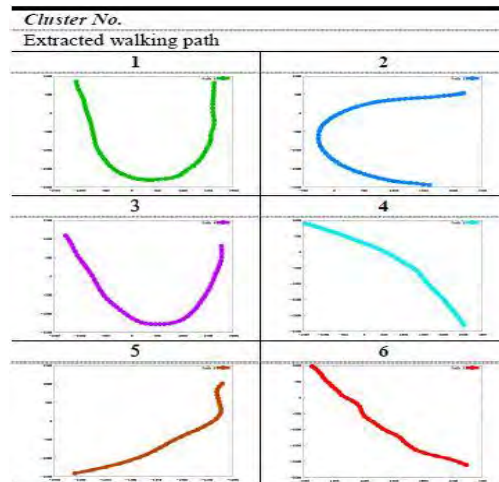


Figure 2.10: Experimental result [10].

There are total six paths are obtained after data extraction which represented the six clusters. Each cluster has different colors to differ each other. Figure 2.10 above shows the entire six paths resulting from the clustering.

#### 2.4.2 Multilinear Decomposition of Human Walking Paths – Chris A. Ramirez, Mario Castelan and Gustavo Arechavaleta [14].

This paper aims to construct a consistent tensor model to study highly redundant human walking paths. Tensors are geometric entities that are introduced into mathematics and physics in order to widen the range of scalars, vectors and matrices. In this paper, tensors are used to generalize and widen the analytic capacity of linear algebra. Moreover, tensors (A, B,...) are commonly organized into multi-dimensional arrays of numerical values as they have relationship with vector spaces. A mode  $n$ -matrix (A, B,...) is produced when the tensor is flattened. Figure 2.11 indicates the mode  $n$ -matrix and flattening of tensor.

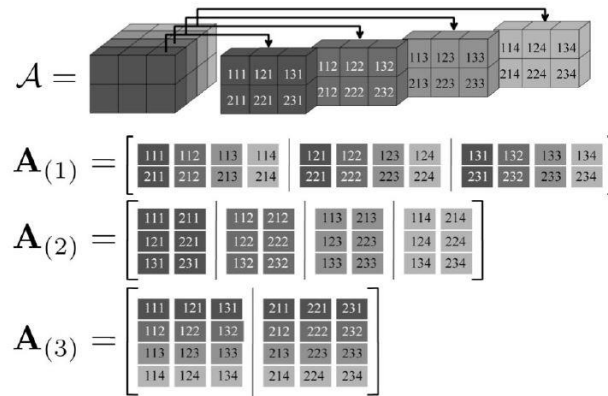


Figure 2.11: Mode  $n$ -matrices and flattening of tensor [14].

Other than that, Singular Value Decomposition (SVD) is applied in tensor flattening process. Singular Value Decomposition will widen the capabilities of Principal Components Analysis (PCA) in order to study the interaction between the vector spaces of data. The analysis of the research is an original database of human walking experiment. The experiment is performed by five subjects with total path is 14 kilometers. During the experiment, the subjects were asked to walk through a distance doorway where the initial and final positions of the subjects were greater than zero.

However, the database is considered as redundant data because the data consist of multiple trajectories linked with pair of configuration ( $x$ ,  $y$ , and  $\theta$ ) performed by each subjects. Roughly the database is consists of 900 examples, with 60 routes, performed three times each by 5 subjects. All the trajectories were transformed into routes by some procedures. Figure 2.12 shows the trajectories that have been transformed into routes.

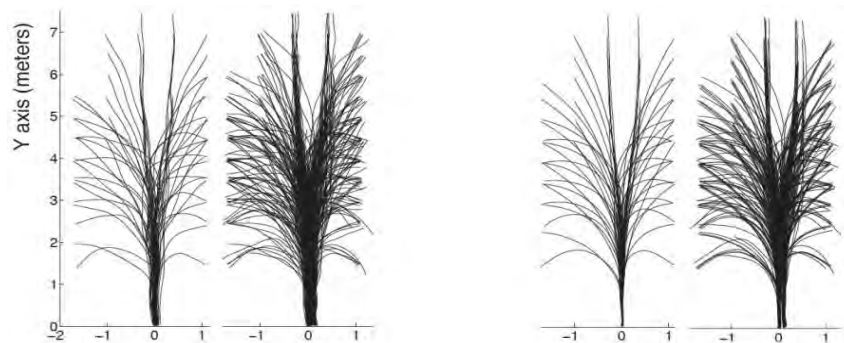


Figure 2.12: The routes of database [14].



The analysis of the experiments is carried out based on three steps. Firstly, the method on how the different characteristics of tensor are anticipated onto each subspace is shown. Secondly, the shape of axes and planning strategies in database are illustrated. Lastly, the way the multilinear be used to predict the shape of trajectory of particular subject is described. In the first step which is subspace analysis, the analysis is divided into four subspace analyses which are subspace of  $(x, y)$ -coordinates, subspace of routes, subspace of subjects and subspace of repetitions. Figure 2.13 indicates all the subspace analysis.

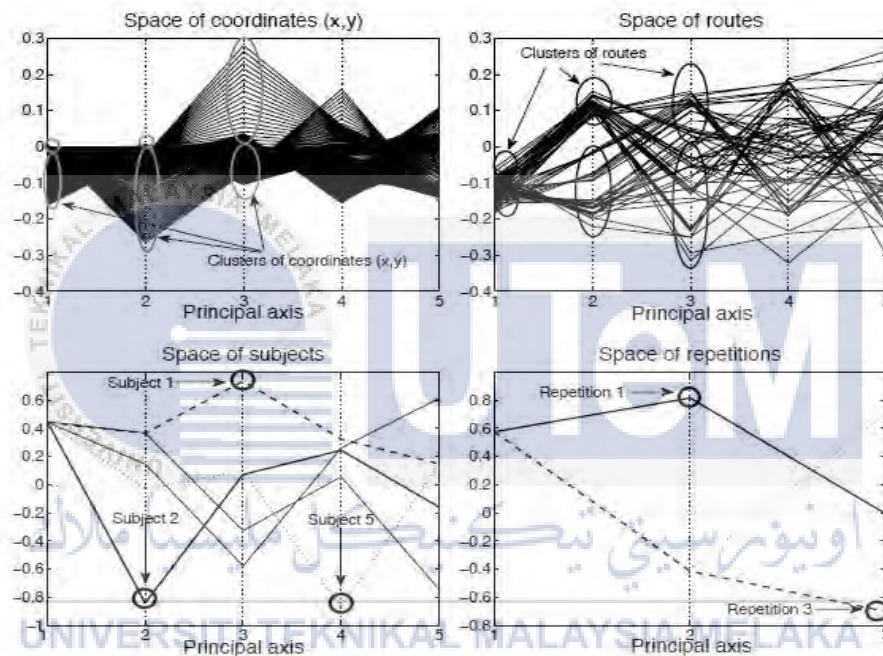


Figure 2.13: Four types of subspace analysis [14].

In subspace of  $(x, y)$ -coordinates, points of  $(x, y)$  are clustered into three clustering class. 200 vectors of latent variables represent 100  $x$ -coordinates and 100  $y$ -coordinates combined forming different paths of database. Both coordinates  $(x, y)$  must exist in the same cluster otherwise a path will not appear in the plot. Figure 2.14 shows the clustering of  $(x, y)$ -coordinates.

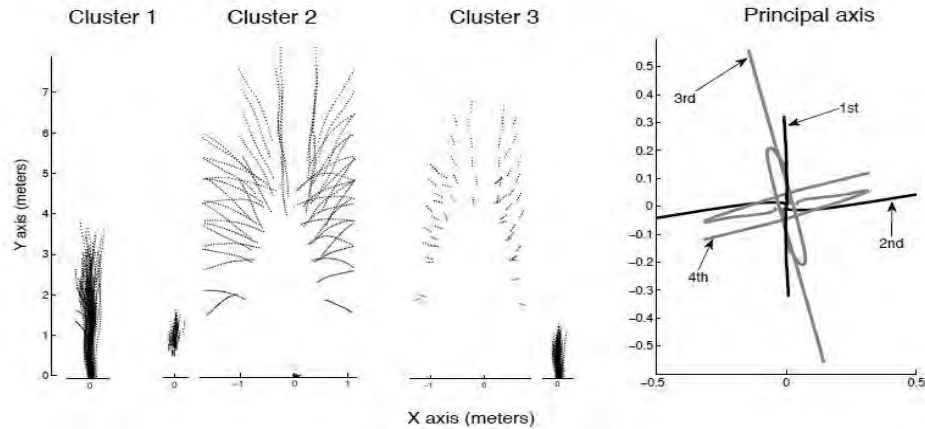


Figure 2.14: The  $(x, y)$ -coordinates clustering [14].

As results from connection between points of  $(x, y)$ -coordinates, routes are formed. In this part, the latent variables are clustered by separating them below and above  $-0.1$ . The changes in route shapes over the  $x$  and  $y$  axes and also the curvature can be inspected through observation from the previous graph. Figure below points to the clustering of routes.

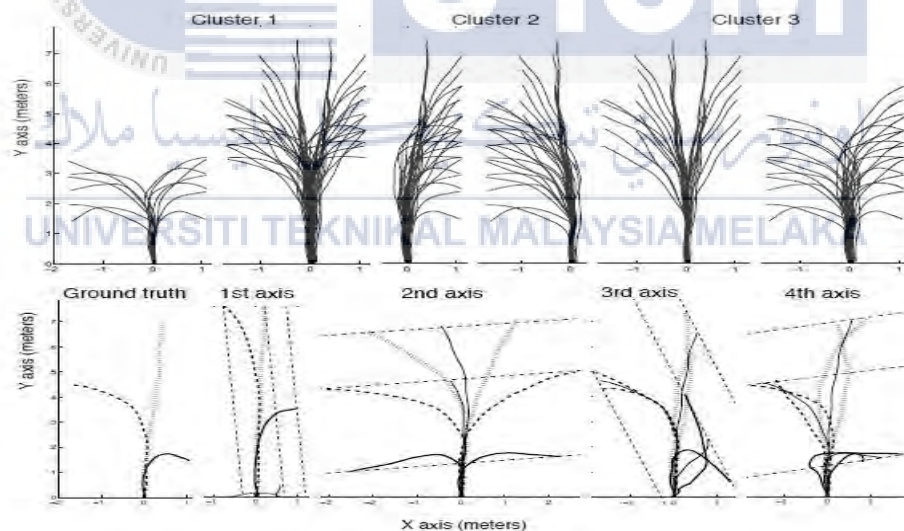


Figure 2.15: Clustering of routes [15].

In this part, the behavior of the subject is observed. During walking, the subject is told to cross an initial door. After that, it is all up to the subject's intention whether he wants to walk further to the right or to the left from his starting point. The subject's intention plays an important part in determining the shape of the path. For a subspace of repetition, the subjects are suggested



to perform a route in two to three repetitions in order to clarify the performances of the routes. Figure 2.16 shows the shape of routes based on subject's intention.

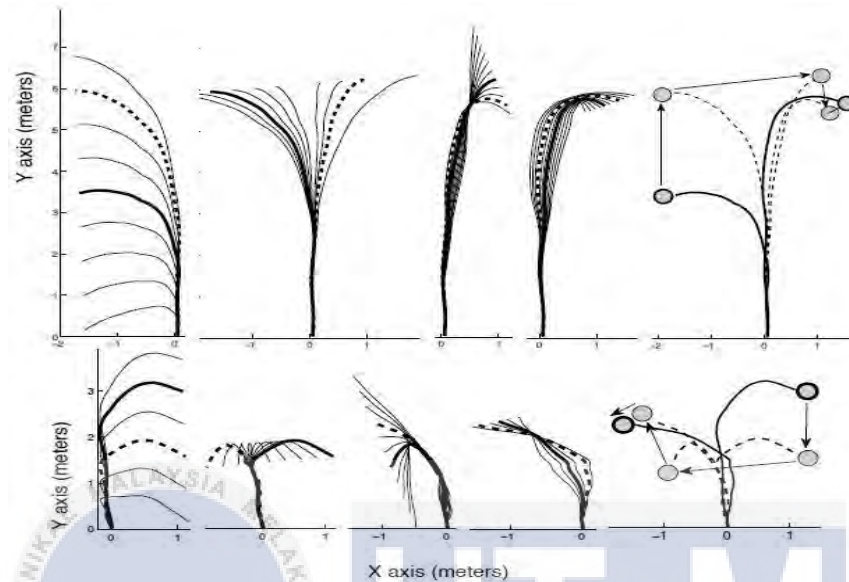


Figure 2.16: Different shapes of routes based on subject's intentions [14].

### 2.4.3 A Genetic Fuzzy System to Model Pedestrian Walking Path In A Built Environment – P.L.Chee, N. Saeid, C. Douglas, N. Mojdeh [15]

This paper focuses the relationship between human walking trajectories and the environments. The relationship between the pedestrian and surrounding environment is cryptic and ambiguity as a system is proposed for modelling and simulation the pedestrian's walking trajectory dealing with the environment stimuli. Moreover, the perception of human to the environment is indeed obscure, and subjective to human's intentions and characteristics. In order to deal with obscurity and uncertainty of human perception in walking, a fuzzy logic method is used in this study. Fuzzy logic has certain advantages when it comes to the ability to imitate human perception compared to other methods as fuzzy logic can deal with vague issues. Furthermore, fuzzy logic is able to produce smooth transitions outputs for human walking trajectories.

When a genetic fuzzy logic system (GFS) is proposed, the genetic algorithm (GA) will be considered as genetic algorithm is used to optimize the parameter of membership function of fuzzy logic. The fuzzy logic design is divided into two parts which are the steering behavior modelling and simulation. In modelling, the discretion of floor is done to represents environment. A radial-based discretion method is used to terrain of human walking environment. Figure 2.17 shows the spatial representation of the environment.

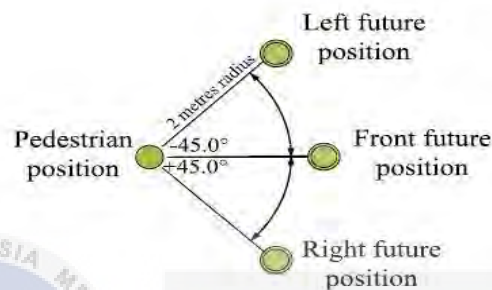


Figure 2.17: Spatial representation of environment within human's field of view [15].

To define possible human position of the next step, three radical positions is located within two meters from human current position with 45 degrees angle to the right and left. All the effects induced by environment to these positions are represented by input fuzzy sets. The inputs are compute by using social fore model (SFM) to represent them in scalar quantities. Then, the scalar quantities are fuzzified by using six membership functions. Figure 2.18 shows the fuzzy rules of scalar quantities.

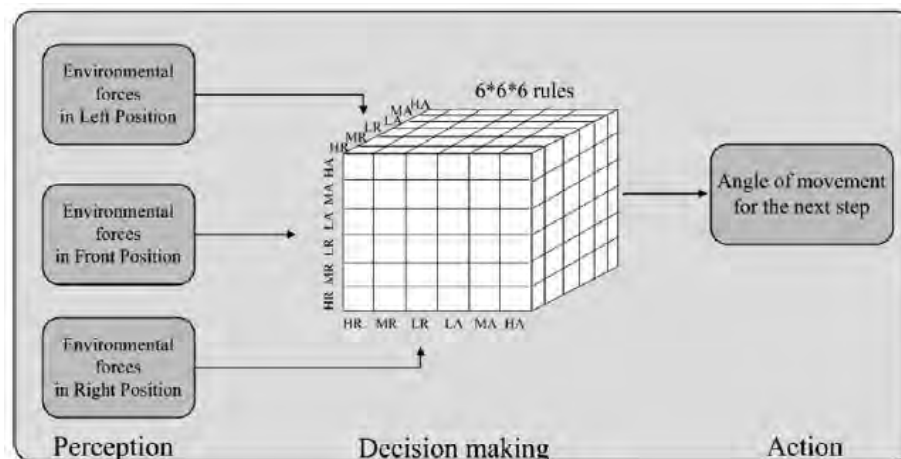


Figure 2.18: Architecture of steering fuzzy model [15].

In data collection, a motion capture device namely OptiTrack™ system is used to collect the data for model validation. Experiment 1 is conducted in a rectangular space with 7 meters in length and 2 meter in width. An obstacle with 70cm in length and 60cm in width is also included into the rectangular space. 25 subjects are considered to walked several times on the rectangular space from a specific point to the destination. Position and orientation angles of the subjects are extracted from the experiment with a total of 74 samples. After that,  $n$ -fold cross validation technique is used to get a more accurate validation of the data. Figure 2.19 shows the results of Experiment 1.

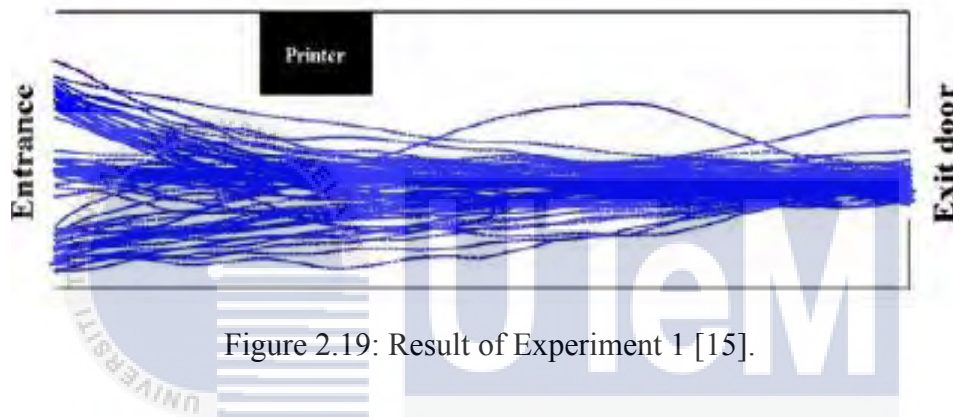


Figure 2.19: Result of Experiment 1 [15].

Next, two other experiments are carried to evaluate the performance of validation. Both experiments used the same method as Experiment 1, but the number of obstacles are differ from each other. For Experiment 2, the environment consists of a hexagonal-shaped environment with two obstacles while in Experiment 3, the number of obstacles is reduced to one. A total of 30 data samples are collected for training purpose and a total of 100 and 125 new data samples are for performance evaluation. The reverse cross validation technique was also applied similar to Experiment 1. Figure 2.20 and Figure 2.21 show the results of both experiment.

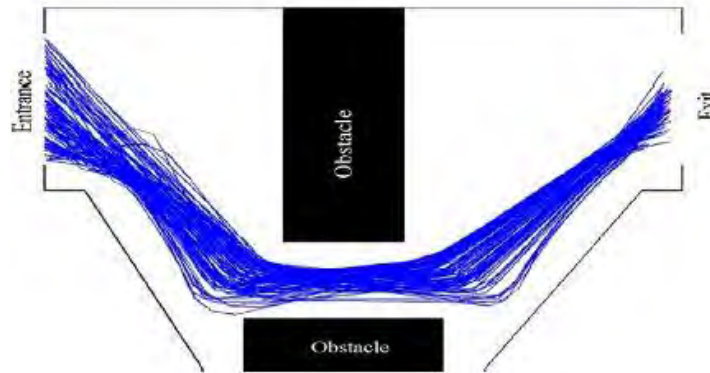


Figure 2.20: Result of Experiment 2 [15].

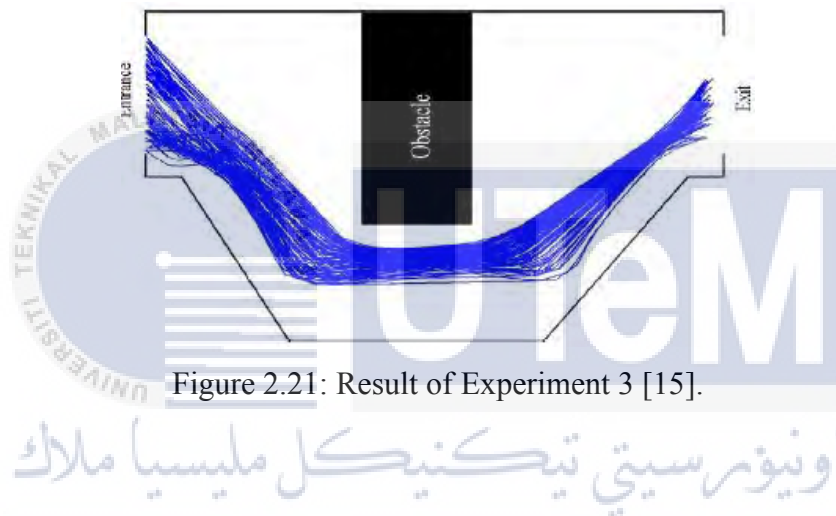


Figure 2.21: Result of Experiment 3 [15].

#### 2.4.4 Human-Observation-Based Extraction of Path Patterns for Mobile Robot Navigation – Takeshi Sasaki, Drazen Brscic and Hideki Mashimoto [16]

This paper proposed the extraction of human walking path method for application of mobile robot navigation method. In this paper, the extraction of human walking path is made through observation method. The human walking behavior are observed and the important points are extracted from the observation before a topological map are build. Intelligent Space (iSpace) configuration is utilized in order to observe human walking behavior as it is difficult to use onboard sensors as their capabilities are restricted.

iSpace is a space with consists of multiple sensors and actuators that are distributed and connected to each other. iSpace contains sensor processing intelligence which reduce the

network load and sensor devices. The networked sensors is called as distributed intelligent network device (DIND) which consists of sensors, processors, and communication device. In this paper, DIND consists of ten charge-coupled device cameras which are connected in pairs to computer with two video capture boards and acts as human tracking system. Moreover, each pair of camera in DIND will display 3-D position of objects by using stereo vision. Figure 2.22 shows the configuration of iSpace implementation and Figure 2.23 shows the experimental environment for extraction of human walking.

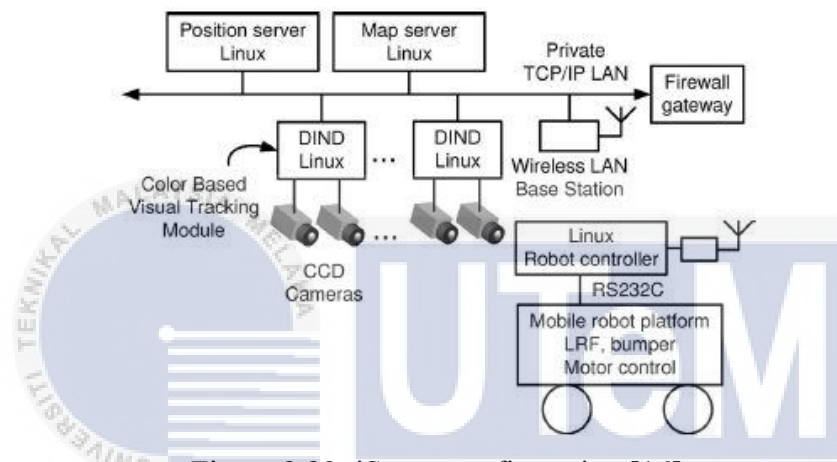


Figure 2.22: iSpace configuration [16].

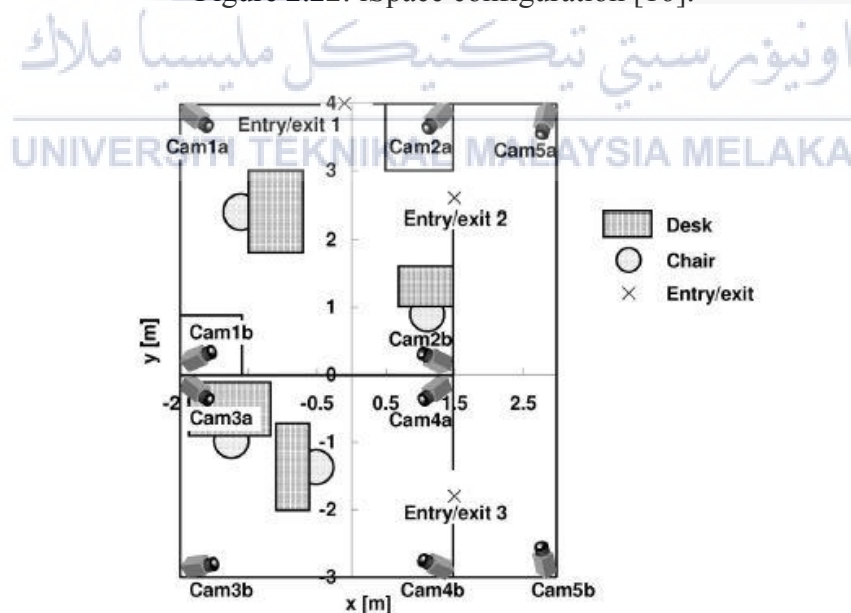


Figure 2.23: Experimental environment for extraction of human walking [16].





### 2.4.5 Abstracting People's Trajectories for Social Robots to Proactively Approach Customers – Takayuki Kanda, Dylan F. Glass, Masahiro Siomi and Narihiro Hagita [17].

In this paper, a series of abstraction techniques for people's trajectories and service framework for using this techniques in social robot are presented. People's trajectories will then used in servicing robots as the robots will approach customers only by using their target local behaviour. As sensing capabilities of robots are limited, a "network robot system" approach is used in this paper. "Network robot system" is backed by universal sensor network that observes and deciphers people's information.

The abstraction technique for people trajectories consists of three techniques; local behavior, space used, and global behavior. These three technique is connected to each other. Local behavior refers to human basic motion such as walking and running, use of space can be define by observation through local behavior, and finally global behavior is perception of people's behavior structure. In order to collect position and local behavior of people, SICK LMS-200 laser range finders are used to measure human walking motion. The experiments are carried out far a week at the entrance of Universal Studio Japan which is a major theme park. The laser range finders are mounted at a height of 85, and a particle-filtering technique is used to track people's trajectories. Figure 2.26 shows the placement of laser range finders

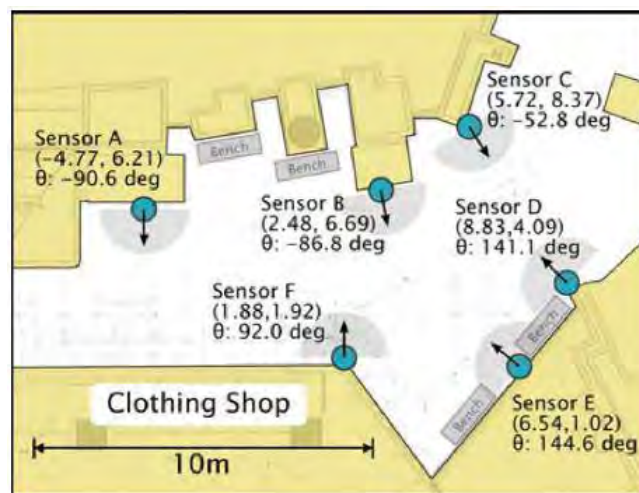


Figure 2.26: Placement of SICK LMS-200 laser range finders [17].

There are two dataset that are analyzed, first a dataset contains 100 trajectories with length of 5s is considered in this analysis. The dataset is taken during morning where only a few people passing through the tracking area. For second dataset, it is taken during evening as the places become crowded and many people passing through the area. After that, the datasets are classified based on their velocity, direction, and shape by using support vector machine (SVM). However, there are several features that are considered by SVM before classifying the datasets such as the end point of normal trajectory, size of space, angle of trajectory, and velocity.

There are a total of 32 features and 4 categories considered by SVM by using Gaussian radial basis function (RBF) kernel. The four categories are style, speed, short-term style and short-term speed. Each category consist of 200 learning samples containing 2-or 5-s trajectory segment and labeled with different number of trajectories. The segments are selected by considering the suitability with each category's concept. Furthermore, the sytem category is tested by using one of cross validation method, the leaving one out method. Figure 2.28 shows the example of a category.

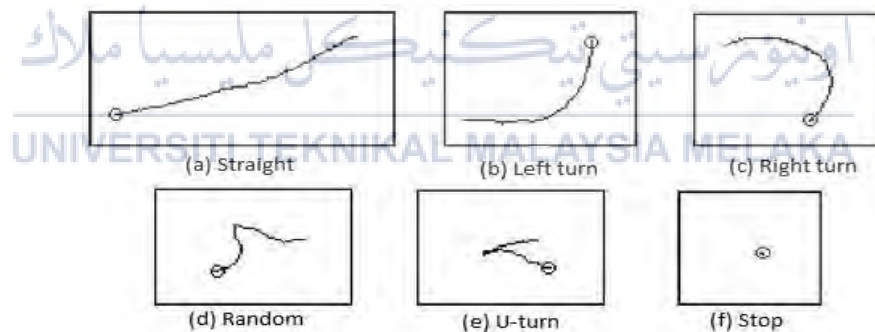


Figure 2.27 : *Style* category [17]

In mapping or space usage technique, the analysis is to recognize the usage of space with changed over time. In this analysis, ISODATA clustering method is used to clustered the space usage during experiment. The environment space are partitioned into a 25cm grid with 2360 grid elements as the time is categorized into 1 hour segments. 40 spatial partitions and four temporal partitions are used to improve the understanding of phenomena occurred in the environment.



For global behavior, the behavior of people visiting the shopping mall is analyzed by using three steps; state chain models, distance between trajectories, and clustering and visualization. In state chain models, the space is partitioned into 4 partitions. Therefore, the trajectories of people are classified based on how long they stayed in the partitioned space and then they are represented in sequences. Figure 2.28 shows the state chain model for global behavior.

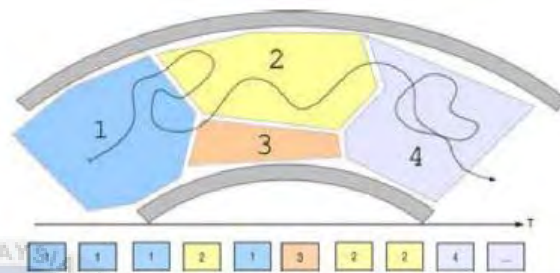


Figure 2.28: State chain model of global behavior [17].

In distance between trajectories, the comparison between trajectories are analyzed. The distance between two state chain models are calculated by using dial pulse (DP) matching method which the space is partitioned into 25cm grid with 2360 grid elements again. DP matching is a suitable method to minimize the number of parameters by keeping the process in simple way. Lastly for clustering and visualisation, the trajectories are clustered with k-means method to determine distinctive visiting pattern. The space is separated into 50 partitions with k-means clustering method. The space partition is similar in size to each other as k-means method provide spatial division which divides polygonal space similarly. Figure 2.30 indicates the global behavior at  $k = 6$ .

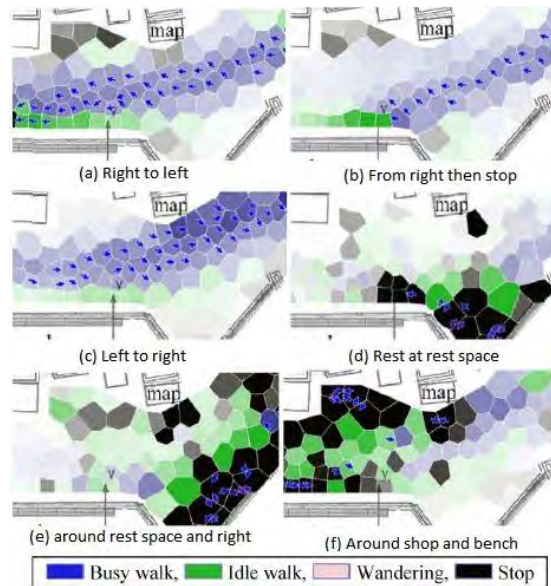


Figure 2.29: Patterns of global behavior [17].

## 2.5 Summary

Table 2.2 show the comparison on the method used for clustering and the performance measurement of the method.

Table 2.2: Comparison of clustering method used and performance measurement.

Title	Clustering method used		Performance measurement
	Unsupervised learning	Supervised learning	
An Approach for Extraction for Extraction of Human Walking Path in Intelligent Space [10]	1) Hierarchical clustering method	-	1) Generalized Cross-Validation (GCV)
Multilinear Decomposition of Human Walking Paths [14]	1) Singular Value Decomposition (SVD)	-	1) Tensor structure improvement

A Genetic Fuzzy System to Model Pedestrian Walking Path In A Built Environment [15]	-	1) Fuzzy logic system model. 2) Genetic algorithm.	1) Reverse cross validation technique.
Human-Observation-Based Extraction of Path Patterns for Mobile Robot Navigation [16]	1) Hierarchical clustering method. 2) K-means clustering method.	-	1) Longest common subsequence (LCSS) method. 2) Dynamic programming algorithm.
Abstracting People's Trajectories for Social Robots to Proactively Approach Customers [17]	1) ISODATA clustering method. 2) K-means clustering method.	3) Support vector machine . 4) Gaussian radial basis function (RBF) kernel.	1) Cross validation method

## 2.6 Method to cluster human walking data

Table 2.3: Differences between K-means clustering and Hierarchical Clustering [18]

Properties	K-Means Clustering	Hierarchical Clustering
<b>Definition</b>	Generates specific number of disassociate flat clustering.	Construct hierarchy of clustering.
<b>Clustering Criteria</b>	Well-suited to generate big clustering.	Use distance matrix. Well-suited for small clustering.
<b>Performance</b>	Increases as number of cluster increases.	Less compared to K-means Clustering Algorithm.
<b>Sensitivity</b>	Very sensitive to noise in dataset.	Less sensitive to noise in dataset.
<b>Cluster</b>	Number of cluster, K.	Number of cluster, K is not required.
<b>Execution Time</b>	Increases time of execution.	Better time of execution.
<b>Quality</b>	Less quality.	More quality.
<b>Data Set</b>	Suitable for large dataset.	Suitable for small dataset.

Based on Table 2.2, K-means clustering method is selected to analyze human walking activity as K-means clustering is suitable for clustering large dataset as human walking data consists of large amount of angular data. K-means clustering also is a simple method which it is easy to be understood and a robust method. Moreover, in this method, the number of cluster, K for clustering can be decided according to users. Even though the quality of K-means clustering is less than hierarchical clustering method, but the method has a better performance than hierarchical clustering which its performance increases as number of cluster, K increases.

## CHAPTER 3

### METHODOLOGY

In this chapter, the methodology to achieve the objectives of the project is discussed. This chapter covers about the method used to collect human walking activity and the method to classify the human walking data. Figure below shows the overall methodology to achieve the objectives of the project.

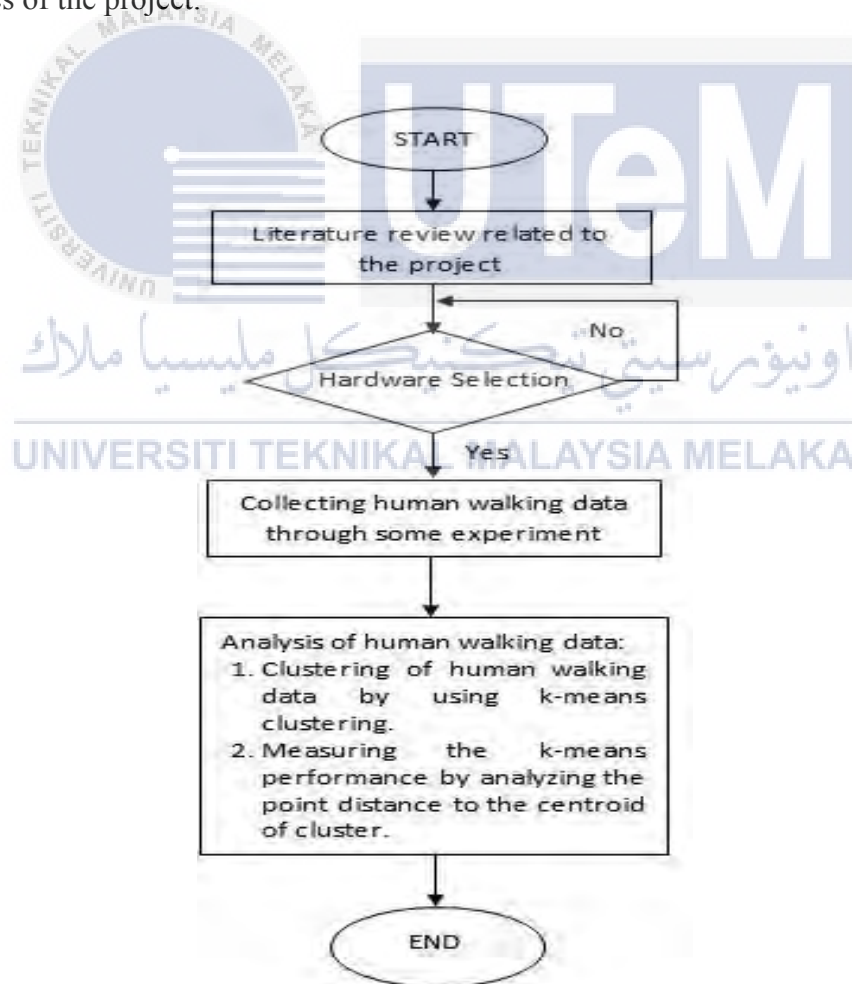


Figure 3.1: Overall methodology of the project.

### 3.1 Project methodology to achieve first objective

In order to collect human walking data, a hardware with consists of several components is created. The hardware will be worn by the subjects and they are asked to walk to a created path. This part will discuss about the selection of component that will be implemented in the hardware. Before going further to the selection of hardware, the basic operation of the hardware will be discussed first. Figure 3.1 indicates the basic operation of the hardware in order to collect data from human walking activity.

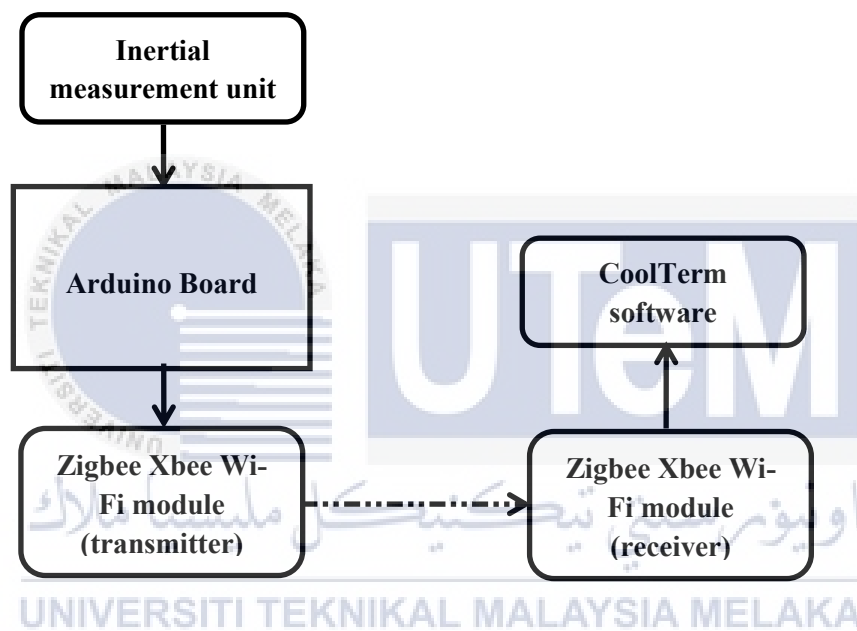


Figure 3.2: Basic operation of the hardware

As shown in Figure 3.2 above, the Inertial Measurement Unit (IMU) is connected to the Arduino Board. The IMU will determine the position and orientation of the human while they are walking. The data from the IMU will be sending to Arduino board and the Arduino will converts the data to a signal that will be transmitted via Zigbee Xbee Wi-Fi module. The other Zigbee Xbee Wi-Fi module that acts as receiver will receives the signal from the transmitter and interfaces the data to the CoolTerm software.

### 3.1.1 Hardware Selection

#### 3.1.1.1 Arduino Nano

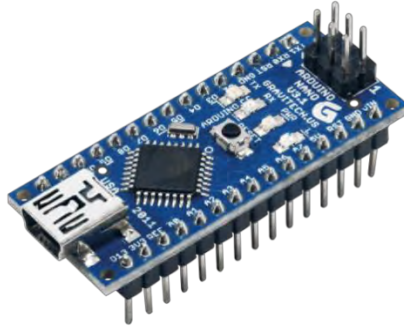


Figure 3.3: Arduino Nano board

Arduino is a microcontroller based-kits that is an open-source prototyping platform based on easy-to-use hardware and software. Arduino Nano is different from other types of Arduino board as it lacks a DC power jack and operates with Mini-B USB cable. Furthermore, Arduino Nano is smaller than other Arduino boards which make it to be used in a simple hardware. Arduino Nano can be programmed by using C language with Arduino Software.

#### 3.1.1.2 XBee® Zigbee RF module



Figure 3.4: Xbee RF module

An Xbee radio frequency (RF) module is a data transfer communication module, which transfer data from one point to another via over-the-air communication. This module can also be operated by connection it to the Arduino board. The Arduino board does not contain data

transfer ability to transfer data to another device. Therefore, an Xbee module is used in order to perform this action. Figure 3.4 shows the picture of Xbee RF module

### 3.1.1.3 Inertial Measurement Unit (IMU)



Figure 3.5: Inertial measurement unit

The IMU is a self-contained device that measure and collect the angular velocity and linear acceleration data with the help of sensors such as 3D gyroscopes and 3D accelerometers. The IMU contained many group of gyroscopes and accelerometers which decides the angular acceleration in X-Y-Z axes of an object. The IMU will determine the position and orientation of the human while walking. The inertial measurement unit that is used for the hardware is Invensens MPU6050.

### 3.1.2 Early calibration of MPU6050

MPU6050 has total six degrees of freedom. Before it is implemented to some circuit, MPU6050 must going through some calibration process first to ensure the accuracy of MPU6050. It is also to understand the way MPU6050 functions. The program for calibration is shown in Appendix B.



### 3.1.3 Built-in Arduino code to Determine Euler Angle and Yaw, Pitch and Roll of a path

In this experiment, the values of Euler angle and Yaw, Pitch and Roll have to be determined. Euler angles are used to describe the orientation of rigid body, while Yaw, Pitch and Roll will describe the rotational of the body. In order to get Euler angle, OUTPUT\_READABLE\_EULER function is defined. Euler angle value is calculated from quaternions values that come from inertial measurement unit which can be shown by using OUTPUT\_READABLE\_QUATERNION function in Arduino.

For Yaw, Pitch and Roll values, to make inertial measurement unit to display their value, function of OUTPUT\_READABLE\_YAWPITCHROLL is defined. Once the function is defined, Serial.print() function is called to show the values on Arduino serial monitor. All the values shown in radian, so to convert them to degrees, formula  $\Theta = (180 \cdot \text{angle in radian}) / \text{Pi}$  had been used.

### 3.1.4 Data collection of human walking activity

To obtain data of human walking activity, the hardware will be worn by the subjects and the subjects will walk along a created path that is provided. In this part, the subjects are told to walk in their normal walking pace without hesitation. There are five subjects involved in doing this experiment. The subjects consist of two males and three females with each of them have different height and weight. Each subject will repeat the experiment five times to clarify the data obtained. The data will be recorded in term of Quaternions, Euler angle, Yaw, Pitch and Raw value. However, only Yaw, Pitch and Raw values are considered for analysis. This is because human also have Yaw, Pitch and Raw angles along the joint of the body. Eventhough the Yaw, Pitch, and Raw angles of human are take account in every joint, the angles are only study at center of gravity which is the waist of human body for this experiment. Figure 3.5 show the Yaw, Pitch and Roll angles at waist of human body.



Figure 3.6: Yaw, Pitch, and Roll angles of human body.

The hardware is placed into a small pouch in order to attach it to human waist. The pouch is attached to human waist because human's waist movement and position are fixed while walking unless the human change his direction. Figure 4.1 shows the hardware setup to gather human walking data.



Figure 3.7: Hardware to collect human walking data.

### 3.1.5 Construction of walking path for human walking activity

In order to perform the analysis of human walking analysis, a specific path which contains left turning and right turning is designed for subject to walk along the path. The path has total length of 80 metres. Figure 3.5 shows the design of the path.

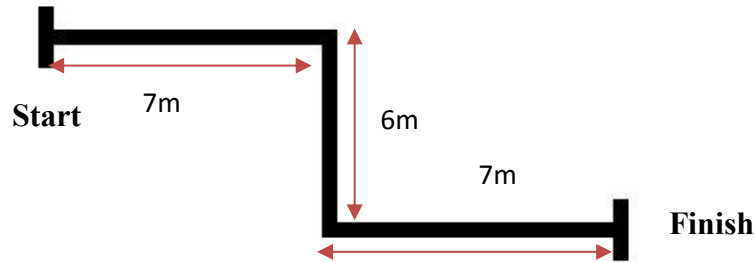


Figure 3.8: Path for human walking activity

### 3.2 Project methodology to achieve second objective

After the first objective is achieved, the second objective will take part. In the second objective, the data from human walking activity will be analyzed. During the analysis, the human walking data will be clustered according to types of path.

#### 3.2.1 Method selection

This part will discuss about the selection of clustering method and the results for method selection.

##### 3.2.1.1 K-means clustering approach on human walking analysis

K-means clustering method is a simple, an unsupervised learning algorithm that is widely used in data mining. It is used to clustering a large dataset and most commonly used to solve problem related to unclassified data. This clustering method is proposed by Mac Queen in 1967 which it is belongs to partitioning clustering method [20]. The method is so simple as it is consists of two phases; first is to select cluster  $k$  centers randomly and seconds is to bring the data to closest center points or centroids. Then, the grouping of each points is done by comparing the distance between the data and the centroids.

The distances between the data and the centroids can be determined by using Euclidean distance formula. When all points are belonging to the same cluster, the

recomputation of centroid is made and the formation of new centroids as there will be comprehension of new points to the cluster. Therefore, there will might be a new formation of new cluster also to improve the distance between data and centroid points. Finally, as there are no movement of centroids after a number of trials, a solution will be reached. Figure 3.6 shows process of K-means clustering.

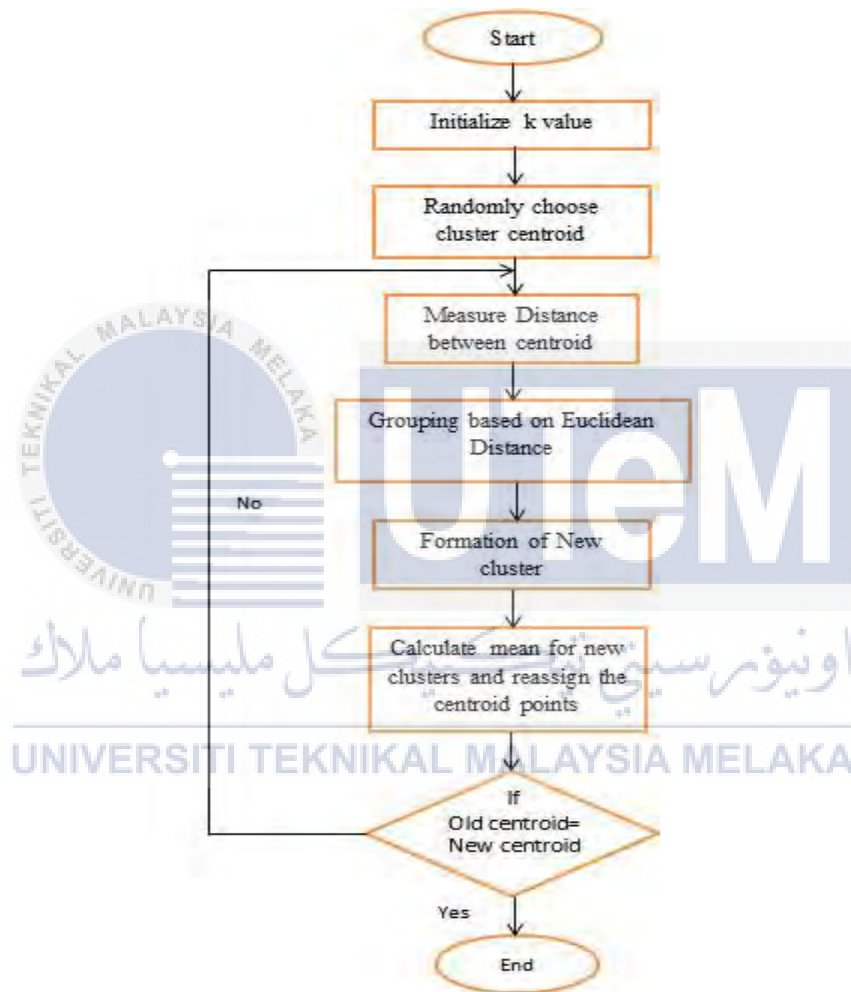


Figure 3.9: Flowchart of K-means clustering process

The formula of Euclidean distance:

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.1)$$

Formula to determine centroid:

$$\operatorname{argmin} d(x_i, c_j) \quad (3.2)$$

Formula for recalculation of centroid within the cluster:

$$c_j(a) = \frac{1}{n_j} \sum_{x_i \rightarrow c_j} x_i(a) \quad \text{for } a=1 \dots d \quad (3.3)$$

However, in order to measure the quality of k-means clustering, sum of squared error (SSE) is used. For SSE, the error for each data point is calculated which means the Euclidean distance of each data point to the closest cluster centroid is calculated first, then the total sum of squared error is computed. By comparing the total sum of squared error of each cluster, the cluster with smallest squared error is selected because it means the centroids are better representation of points in each cluster.

For sum of squared error (SSE), equation (3.2) is used

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} d(c_i, x)^2 \quad (3.4)$$

where

$K$  = number of clusters

$x_i$  = mean of all points

$c_j$  = new cluster center

$n_j$  = Number of object in  $j^{\text{th}}$  cluster

$C_i$  = The  $i^{\text{th}}$  cluster

### 3.2.1.2 Algorithm for K-means clustering

K-means formally is described in an algorithm code. The algorithm code is easy to understand as it simplify the process of K-means in a simple way. Figure 3.10 shows the basic algorithm code for k-means clustering.

**Input:**

- Number of desired cluster,  $k$
- Database  $D = \{d_1, d_2, \dots, d_n\}$  //contain  $n$  data items

**Output:**

- A set of  $k$  clusters.

**Steps:**

- 1) Randomly select  $k$  data object as initial centroids from dataset  $D$ ;
- 2) Repeat;
- 3) Calculate Euclidean distance of data object from all  $k$  centroids and assign data object to nearest cluster;
- 4) Recalculate the cluster centroid;
- 5) Until no changing in centroid movement.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.10: Basic K-means algorithm code [4].

Before K-means algorithm merges to local minimum, the distance and centriods were calculated a number of times or represented by integers  $t$  which is known as number of k-means iterations. The number of iteration done will effects the results of the clustering as the more suitable cluster centers are selected.

### 3.2.1.3 Application of K-means clustering in MATLAB

In order to perform the analysis, K-means clustering method is done by using MATLAB software. In MATLAB software, K-means clustering is performed by defining kmeans command with several algorithm. Figure 3.11 shows the MATLAB commands for k-means clustering.

```
load datafile.csv
rng(1); % For reproducibility

tic; % Start stopwatch timer
[idx,C,sumd] =
kmeans(datafile,3,'Distance','cityblock','Display','iter','Replicates',10);
toc % Terminate stopwatch timer

ptsymb = {'bs','r^+','md','go','c+','b:','c--'};
% plot 3d k means result
for i = 1:5
    clust = (idx == i);
    plot3(datafile(clust,1),datafile(clust,2),datafile(clust,3),ptsymb{i});
    hold on
end
plot3(C(:,1),C(:,2),C(:,3),'k*');
hold off;
xlabel('ROLL'); ylabel('PITCH'); zlabel('YAW');
view(-137,10);
grid on;
title('Human walking data with K-means where K=5');
hold on;
```

Figure 3.11: K-means algorithm code for MATLAB

Firstly, the database was saved in .csv (comma delimited) file. The database in .csv file only consists of data values only without labeling for the classes. This is because the MATLAB software can only read the data value only from .csv file. Next, the size of database was determined to analyze which dimension the database is belongs to. After that,  $[idx,C,sumd] = kmeans(datafile,3, 'Distance','cityblock','Display','final','Replicates',7)$  was called to perform k-means clustering which returns the within-cluster sums of point-to-centroid distances in k-by-1 vector sumd. The value 3 in the function indicates the number of k-cluster, 'Distance' is distance measure of centroid clusters, 'cityblock' is the available distance measures which the Euclidean distance formula is used. Moreover, 'Display'

arguments was used to display the level of output which the '*final*' is one of the display option which displays the results of final iteration. '*Replicates*' argument was used to repeat the clustering in number of times by using initial cluster centroid and value 7 shows the number of repetition that will be done. Lastly the clustering was plotted into a 3-D graph.

### 3.2.2 Analysis on clustering of human walking data

This analysis was done to analyse the clustering of human walking data. Since the path for all walking analysis are the same, the walking data were analyzed by considering the data from inertial measurement sensor, MPU6050. The data that were considered from MPU6050 are Quaternions, Euler angle, and Yaw, Pitch and Roll values. However, only Yaw, Pitch, and Roll values are been clustered. All the data values were taken from each subject. A total of 15 data values were used in clustering process.

Then, all the data values were going through clustering process where k-means clustering algorithm was used. The number of clustering of each data was determined according to the visualisation by observing compatibility of each data points to the centroid. Further explanations of the analysis will be discussed in the next chapter.

### 3.3 Methodology for achieving third objective

In this section, the performance of K-means clustering is evaluated. The performance of K-means clustering is evaluated to avoid finding noise patterns in the clustering data. Moreover, the performance evaluation is important as it will compare the two clustering data and determined whether the suitable number of k cluster. In this experiment, Silhouette Coefficient method is used to measure the performance of the k-means clustering algorithm.



### 3.3.1 Silhouette Coefficient method in selecting number of cluster

Silhouette Coefficient is a method that is used to study the suitable number of  $k$  cluster by measuring the of similarity of each point in same cluster compared to the each point of adjacent cluster. The closeness of the points is measured by separation distance between the clusters. The measure has a range from -1 to +1. If the coefficient shows a high value, it indicates that the objects if well classified into the cluster while small value indicates the object is poorly classified into the cluster. The Silhouette index can be defined by using Equation (3.5)

$$s = \frac{b - a}{\max(a, b)} \quad (3.5)$$

Where

$a$  is the mean distance between a point to all point in same cluster.

$b$  is the mean distance between a point to all point in adjacent cluster.

As the silhouette value is near to +1, it indicates that the point is well classified into the cluster and very far away from the adjacent clusters. Meanwhile, if the silhouette value is equal to 0, it shows that the point is located near the border of two clusters which make it considered as overlapping clusters. Lastly, when the silhouette value is near to -1, it indicates that the point is poorly classified as the point is been clustered into a wrong cluster.

### 3.3.2 Analysis on performance of K-means clustering of human walking data

In order to analyze the performance of K-means clustering, the Silhouette Coefficient method was used. The Silhouette coefficient will determined the suitable number of  $k$  cluster of a database. In this analysis, the number of cluster will be selected from 1 to 6 and the Silhouette Coefficient values will be compared. To determined the suitable number of cluster for human walking data.

## CHAPTER 4

### RESULT AND DISCUSSION

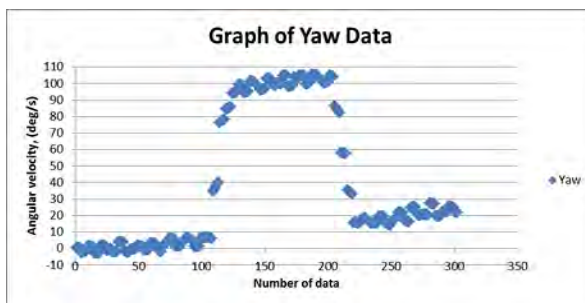
The results collected based on the experiments done in completing and achieving the objectives are discuss here. Experiments are done in order to complete the objectives of this projects

#### 4.1 Analysis of data collection on human walking data

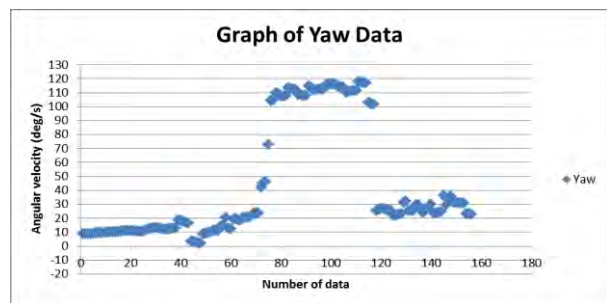
This analysis is done to analyse the data that are representing human walking activity. The data are taken from the hardware which a total of 5 subjects are asked to walk along a created path. In this analysis, Yaw, Pitch, and Roll values are selected as Yaw, Pitch, and Roll values represents the rotation of body in X-Y-Z axes.

##### 4.1.1 Analysis of Yaw, Pitch and Roll data

###### 4.1.1.1 Analysis of Yaw data



(a) Subject 1



(b) Subject 2

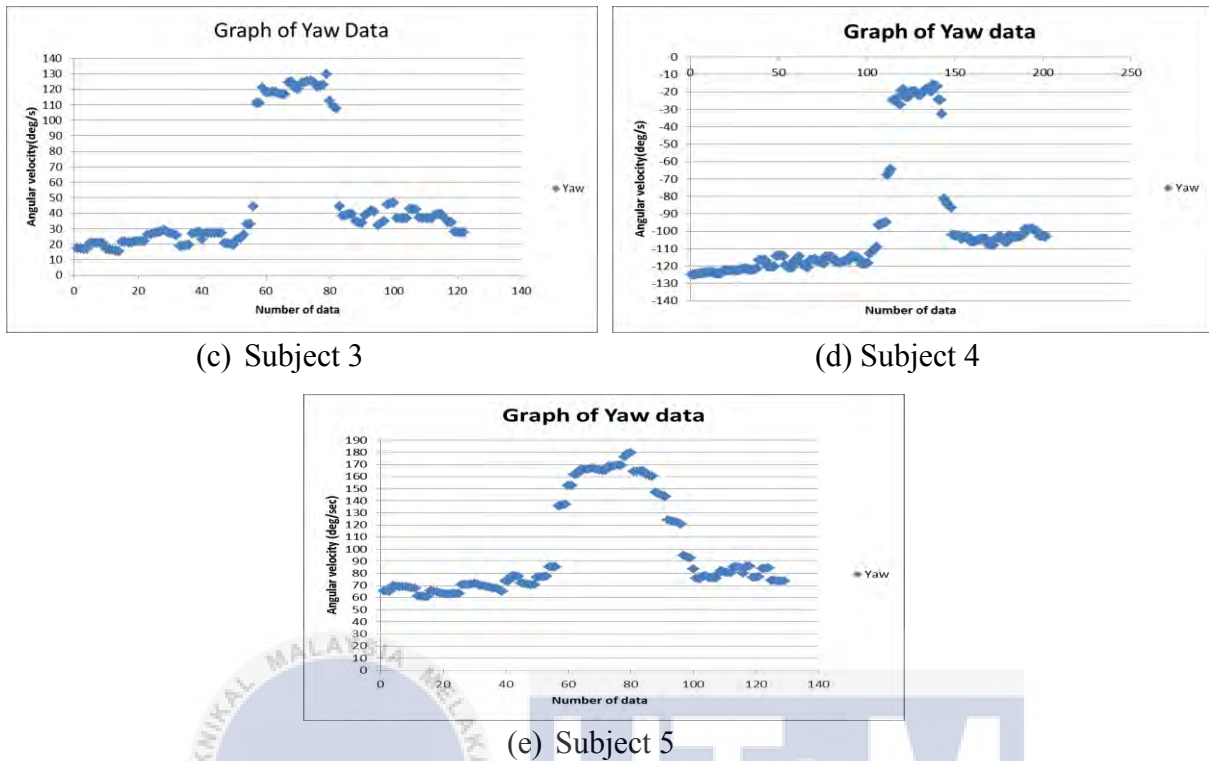


Figure 4.1: Graph of Yaw data of each subject.

Figure 4.2 shows the graph of Yaw, Pitch, and Roll data of each subject. As shown in the figure, the starting section of all graphs have constant values because during that time, subjects are walking in a straight forward path. Each subject has different starting values from each other as the height of waist of each subject is different. However as shown in Figure 4.1(d), all the values are in negative values because the subject is asked to wear the hardware in upside position. When the subjects walk along the straight path, there is no rotation in Y-axis of the body present during the motion. In the second section of the graphs, the angular velocity of data has drastically increased from the first section. The increasing of angular velocities of data shows that the inertial measurement unit, MPU6050 has detected clockwise rotation at Yaw-axis of the body yet a change in direction. At this point, the subjects are turning right while they are walking on a straight forward path. Next, the angular velocities of data are decreasing drastically in the third section. The decreasing of angular velocity indicates the negative yaw or anticlockwise rotation at Yaw-axis of the body. This shows that the subjects are turning left while they are walking. However, there are some skid away data existed in the

graphs as shown in Figure 4.3(b). That data are existed because of there are some distortions or noises occurs while during the experiment.

#### 4.1.1.2 Analysis of Pitch data

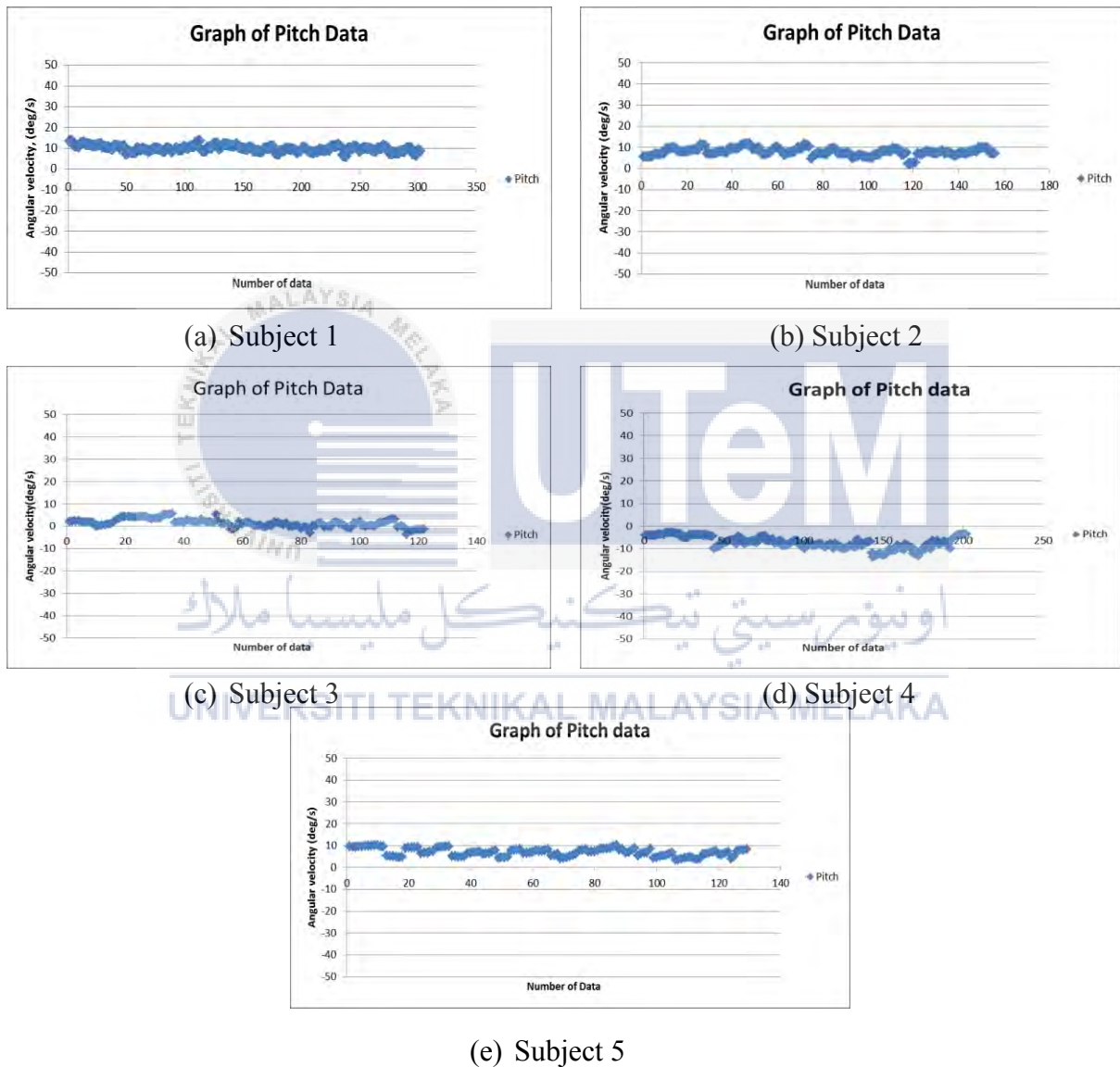


Figure 4.2: Graph of Pitch data from each subject

Figure 4.4 shows the graph of Pitch data from each subjects. As shown in the graph, there are no huge changes in values of Pitch data. The Pitch data remains in the range from -

20 degrees per second to 20 degrees per second. Pitch data is considered when any moment acting on a pitch axis or x-axis of a body. Pitch axis can be explained by taking aircraft principal axes theory. In aerodynamics, pitch moves the nose of aircraft up and down. However in human walking activity, the pitching moment is occurred but in a small range as there are no rotation on pitch axis while human are walking. Therefore, the values of pitch data remain constant on each body.

#### 4.1.1.3 Analysis of Roll data

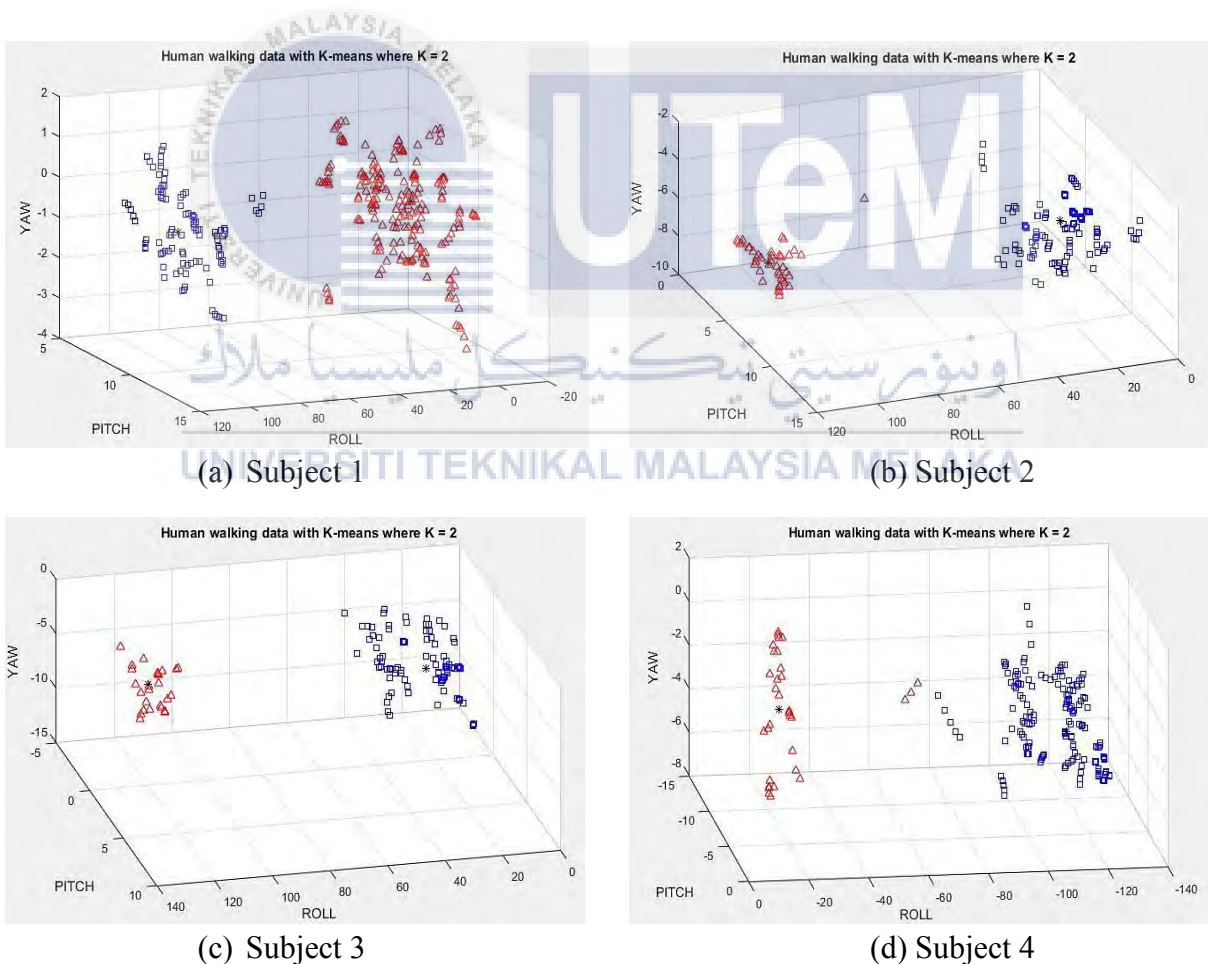


Figure 4.3: Graph of Roll data of each subject

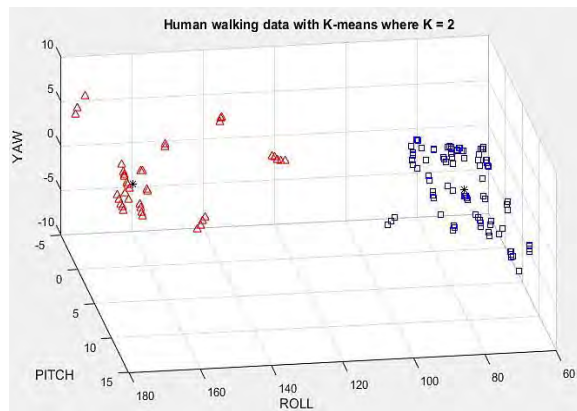
Figure 4.5 shows the graph of Roll data of each subject that are collected from human walking analysis. As shown in the graph, there are no huge changes in values of Roll data which remains in range of -10 degrees per second to 10 degrees per second. Roll data is considered when any moment acting on a roll axis or z-axis of a body. Roll moment is occurred but in small ranges as there are no rotation on roll axis while human are walking. The small changes are caused by subject's walking posture which are different from each other.

## 4.2 Analysis on K-means clustering of human walking data

### 4.2.1 Analysis on K-means clustering when $k = 2$





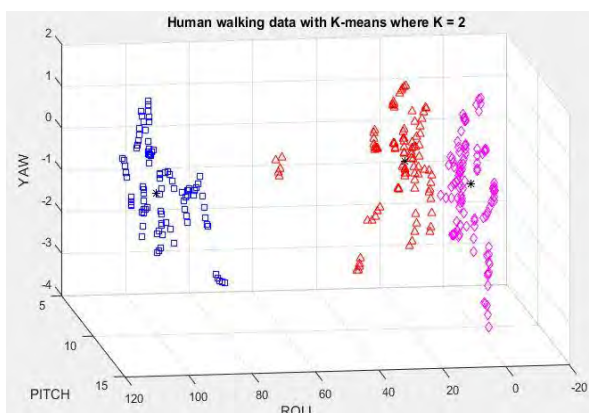


(e) Subject 5

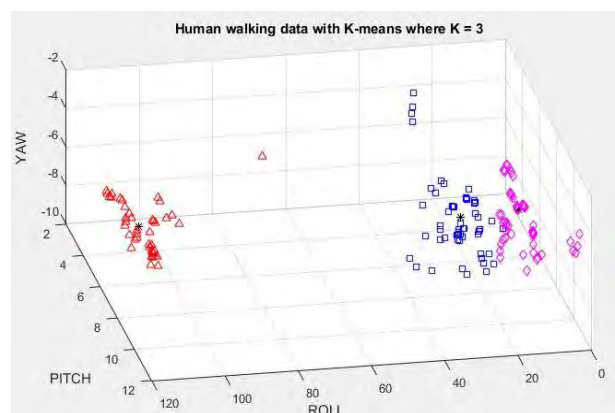
Figure 4.4: Graph of human walking data when  $k=2$ 

Figure 4.4 shows the graph of k-means clustering of human walking data of all subjects. As shown in figure, there are two clusters exist in the graph. This is because the number of cluster,  $k$  was selected to be two. The results of the selection has made two cluster centroids to be randomly choose. Therefore, all the closest points to the cluster centroid are assigned to the existing clusters. However, there are some points that are poorly classified as they are very far from the cluster centroid yet still been clustered as in Figure 4.4(d). The three points are supposedly belong to the right cluster as they are closer to the right centroid. In Figure 4.4(e), there are many points that are located far away from the cluster centroid but they are belong to the cluster. This was an example to indicate a bad cluster.

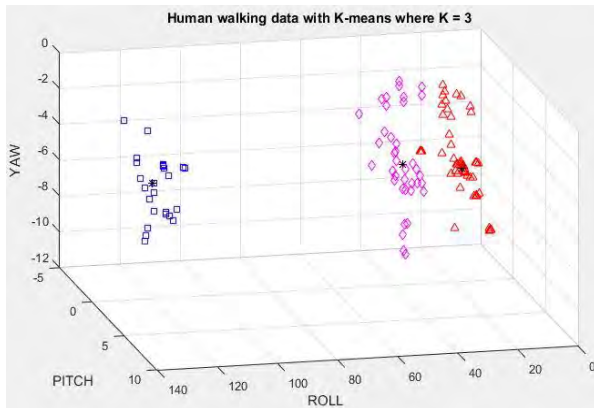
#### 4.2.2 Analysis on K-means clustering when $k = 3$



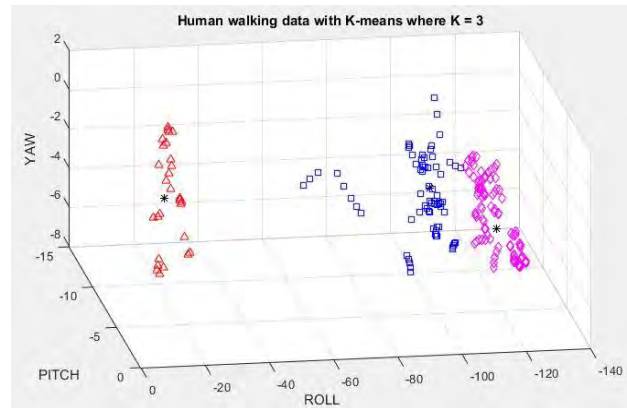
(a) Subject 1



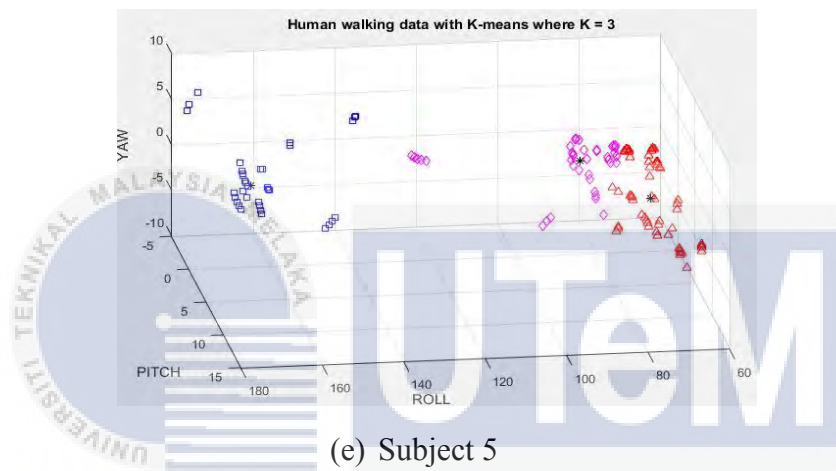
(b) Subject 2



(c) Subject 3



(d) Subject 4



(e) Subject 5

Figure 4.5: Graph of human walking data when  $k=3$

Figure 4.5 shows the graph of human walking data when number of cluster,  $k$  is equal to three. As shown in the figure, there are 3 clusters exist in the graph. . This is because the number of cluster,  $k$  was selected to be three. The results of the selection has made three cluster centroids to be randomly choose. Therefore, all the closest points to the cluster centroid are assigned to the existing clusters. The distance of points to the cluster centroid are much closer now compared to Figure 4.4. Based on Figure 4.4(d), there are three points that are located far away from the centroid of the cluster. However, as shown in Figure 4.5(d), after the number of  $k$  cluster is changed into three, the three points are assigned to the new cluster which the distance from the centroid is much closer. This shows a good example of good clusering.



### 4.2.3 Analysis on K-means clustering when $k = 4$

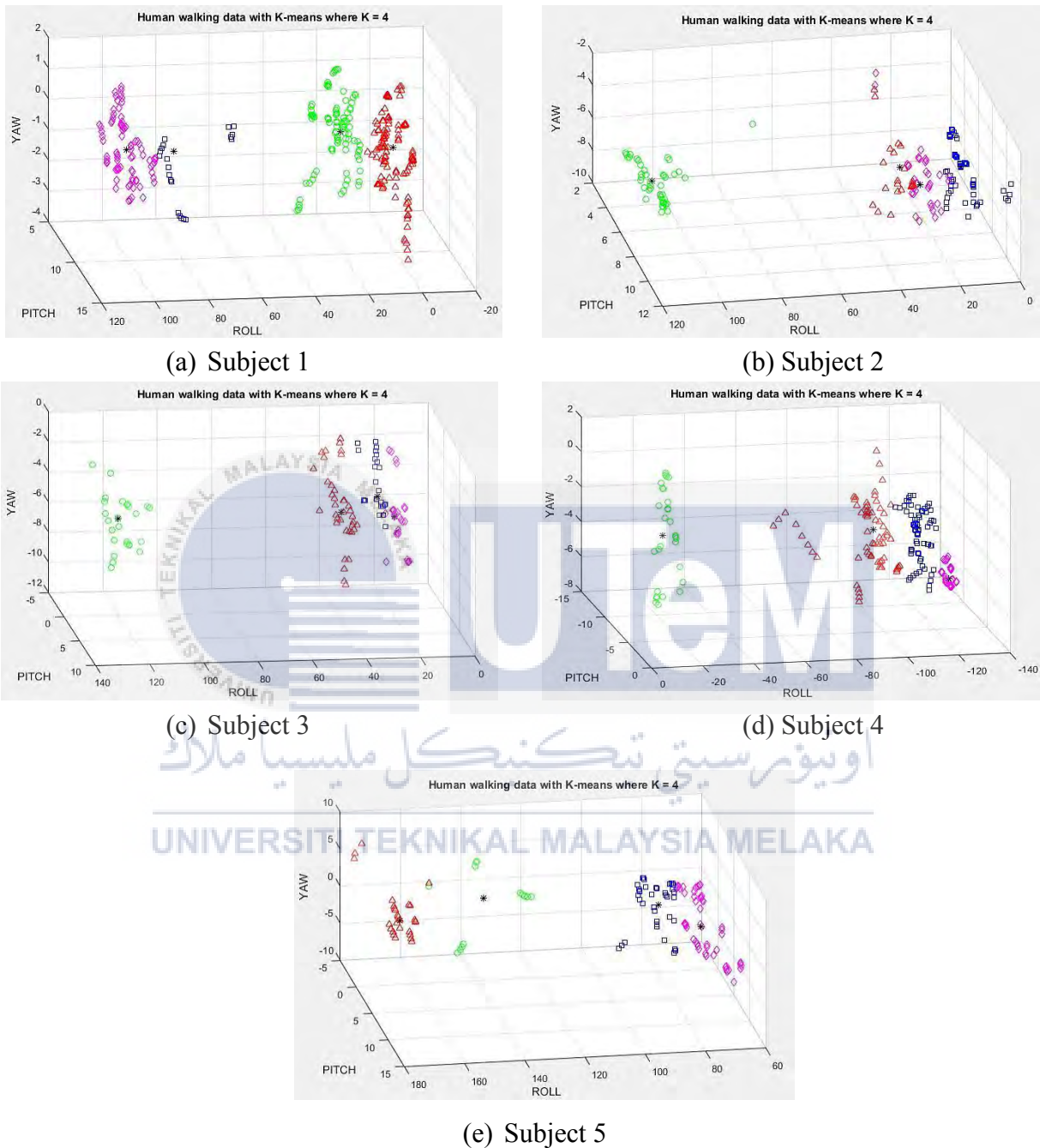


Figure 4.6: Graph of human walking data when  $k = 4$

Figure 4.5 shows the graph of human walking data when number of cluster,  $k$  is equal to four. As shown in the figure, there are 4 clusters exist in the graph. This is because the number of cluster,  $k$  was selected to be four. The results of the selection has made four cluster

centroids to be randomly choose. Therefore, all the closest points to the cluster centroid are assigned to the existing clusters. In Figure 4.6(b), there are a point that are located far away from the green cluster while in Figure 4.6(e), there are no points that are located near the centroid of green cluster but the all the points are classified into green cluster.

#### 4.2.4 Analysis on K-means clustering when $k = 5$

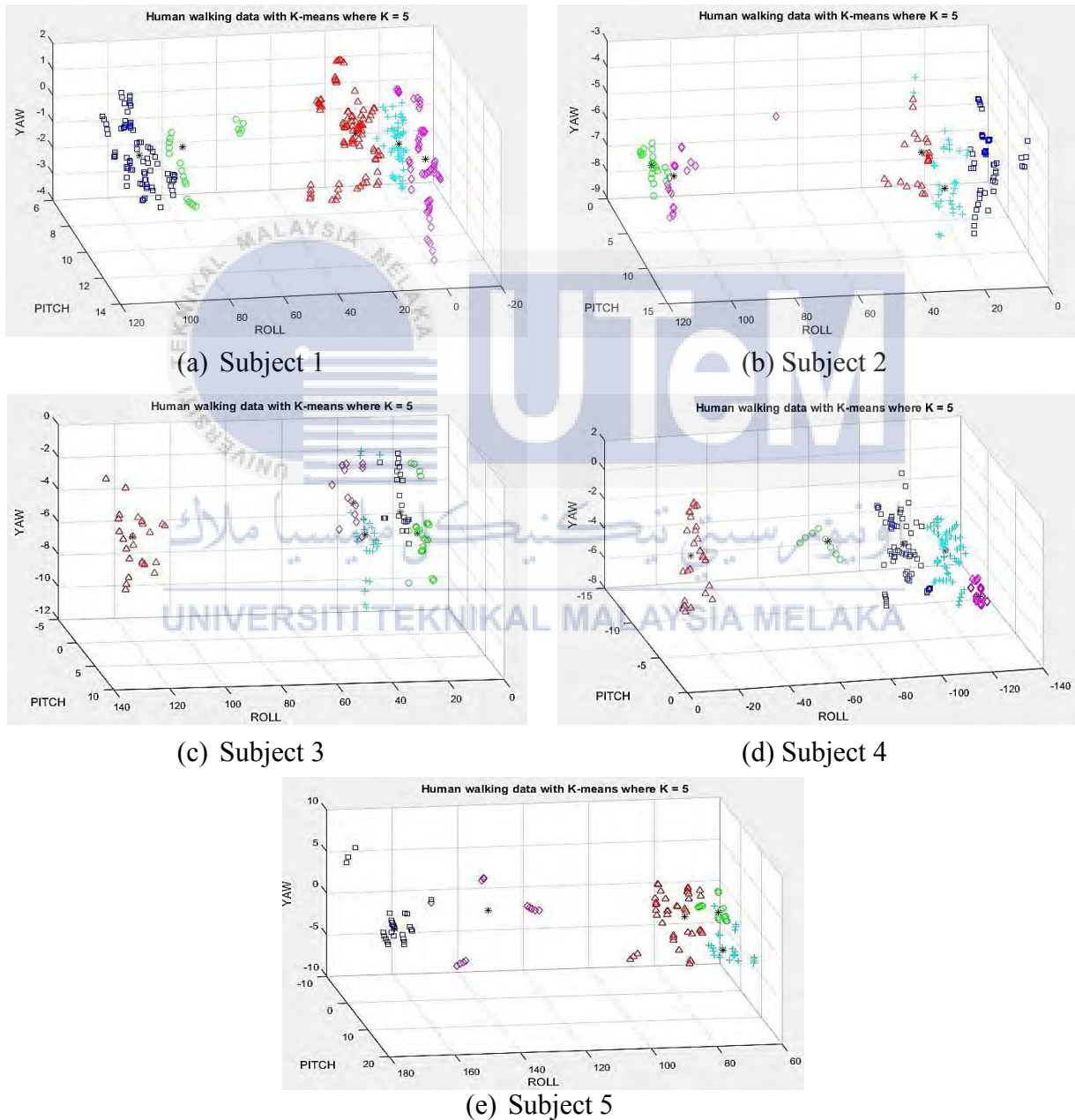
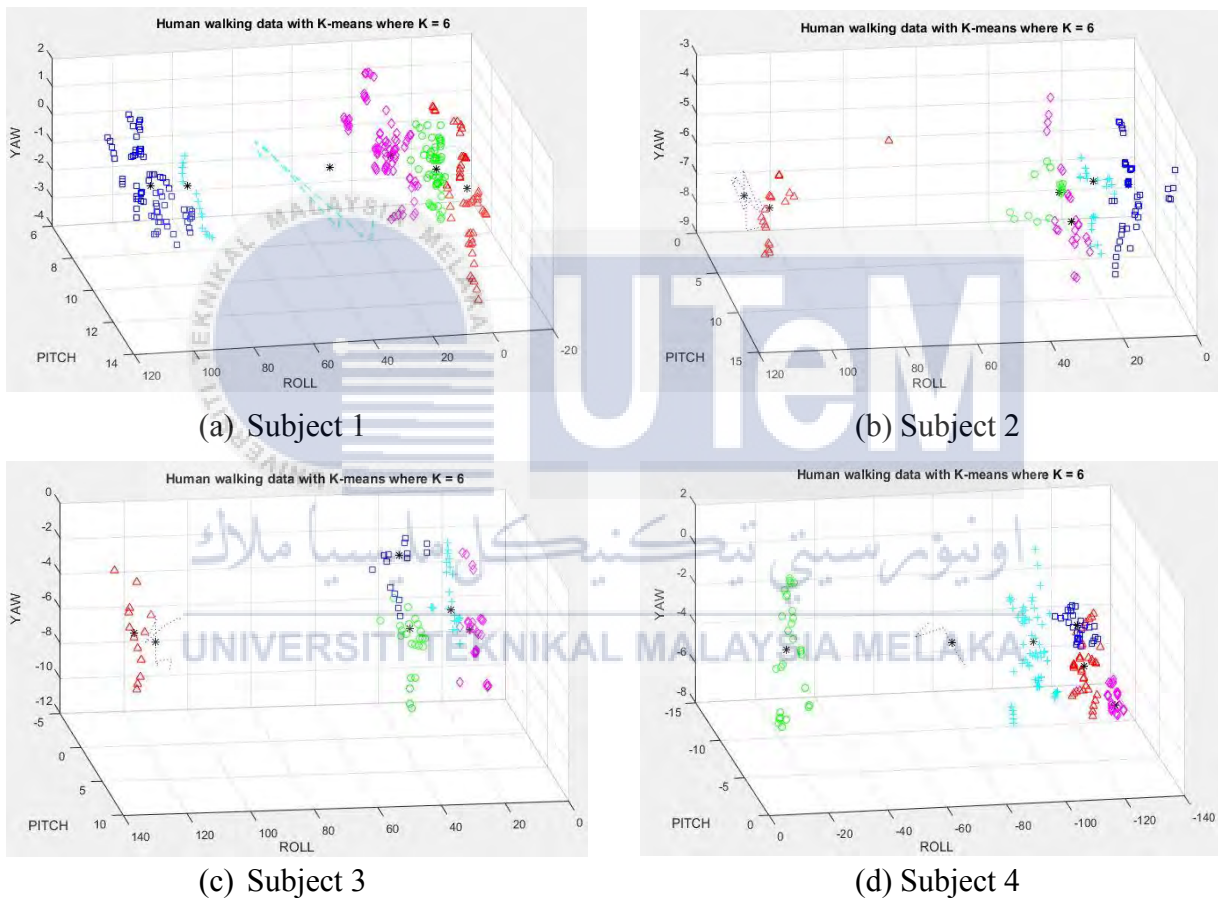
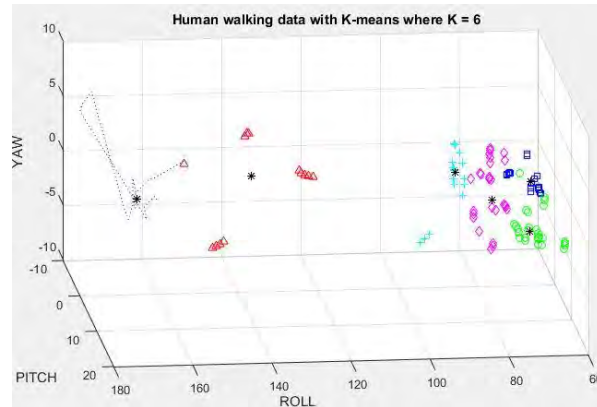


Figure 4.7: Graph of human walking data when  $k = 5$

Figure 4.7 shows the graph of human walking data when number of cluster is 5. In Figure 4.7(e), all the points in the pink cluster is located far away from the pink cluster centroid. This is because the collapsing between the two clustering distance for the points make the points are classified into the wrong cluster. Moreover, other points in other clusters have mixed with each other and the accurate clustering is not achieved.

#### 4.2.5 Analysis on K-means clustering when $k = 6$





(e) Subject 5

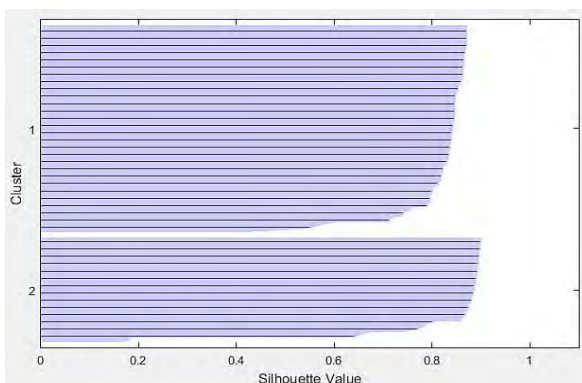
Figure 4.8: Graphs of human walking data when  $k = 6$ 

Figure 4.8 shows the graphs of human walking data when number of cluster is 6. In Figure 4.8(e), all the points in the red cluster is located far away from the red cluster centroid. This is because the collapsing between the two clustering distance for the points make the points are classified into the wrong cluster. Moreover, other points in other clusters have mixed with each other and the accurate clustering is not achieved.

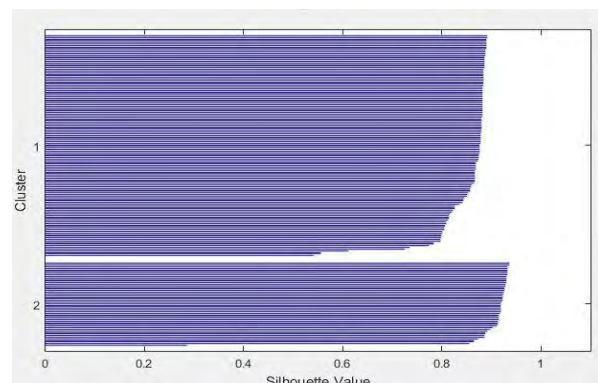
### 4.3 Analysis on performance of K-means clustering of human walking data

#### 4.3.1 Analysis on Silhouette Coefficient value in selecting number of $k$

##### 4.3.1.1 Analysis for $k = 2$ with average Silhouette value

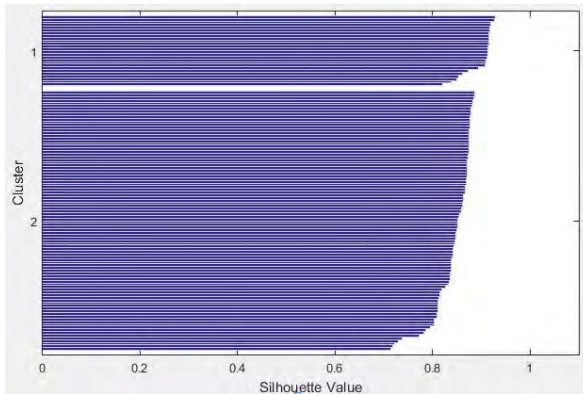


(a) Subject 1 with average 0.8227

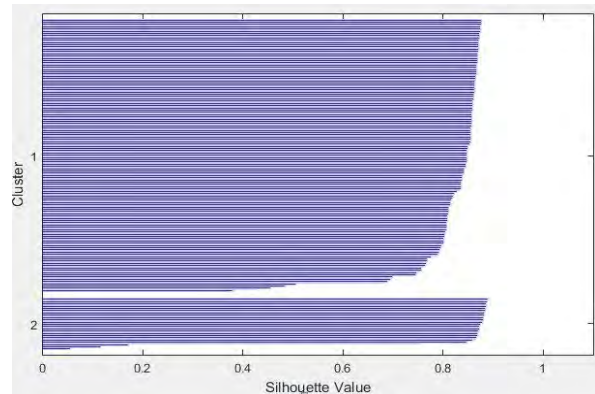


(b) Subject 2 with average 0.8062

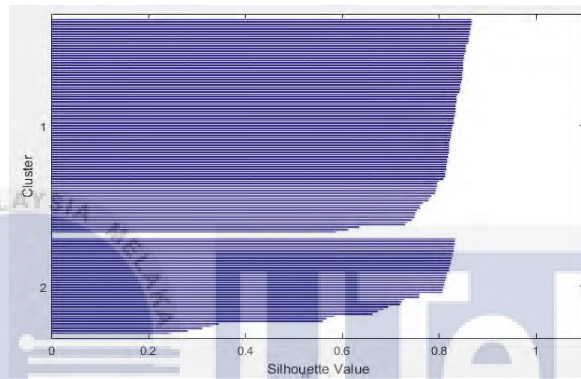




(c) Subject 3 with average 0.8554



(d) Subject 4 with average 0.8190



(e) Subject 5 with average 0.7839

Figure 4.9: Graphs of Silhouette plot for  $k = 2$ 

Figure 4.9 shows the Silhouette plot of k-means clustering for number of cluster,  $k$  equals to 2. From the silhouette plot, all the most point in first and second cluster have a large silhouette which is greater than 0.8, indicates that the clusters are separated from each other. The first cluster for each subject have a large thickness referring that a large number of data is clustered in. However for subject 3, the second clustered is thicker than the first cluster because the value of the data is all negative as the subject is asked to wear the hardware upside down. The highest average of Silhouette value is 0.8554 which is the data of second subject.

#### 4.3.1.2 Analysis for $k = 3$ with average Silhouette value

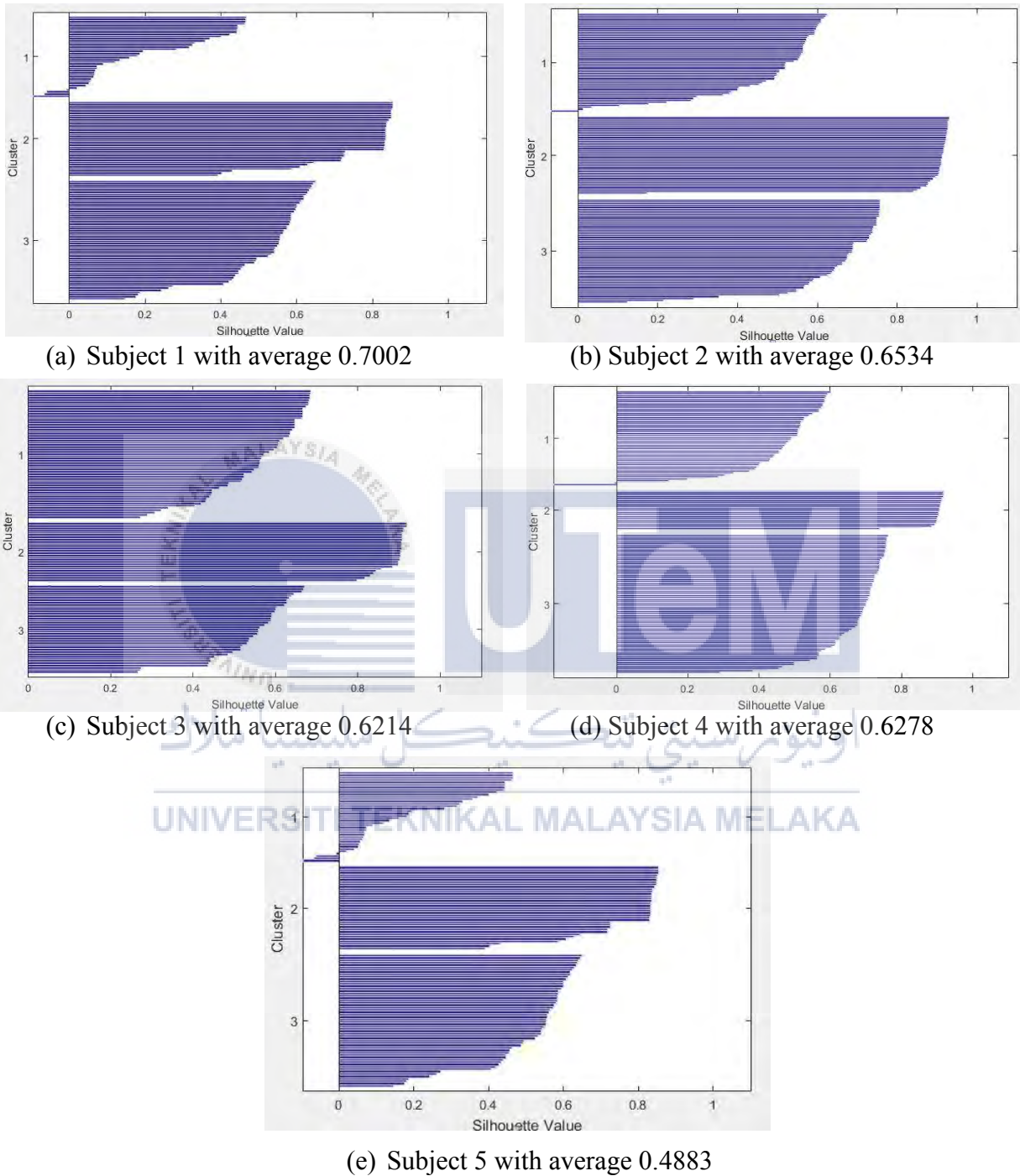


Figure 4.10: Graphs of Silhouette for  $k = 3$

Figure 4.10 shows the Silhouette plot of  $k$ -means clustering for number of cluster,  $k$  equals to 3. From the graphs, the second cluster have a large silhouette value which are mostly

greater than 0.8, indicates that a good clustering take place. However, the third cluster contains points with low silhouette values, and also have a few points with negative values which indicates a bad clustering or the points are classified into wrong clusters.

#### 4.3.1.3 Analysis for $k = 4$ with average Silhouette value

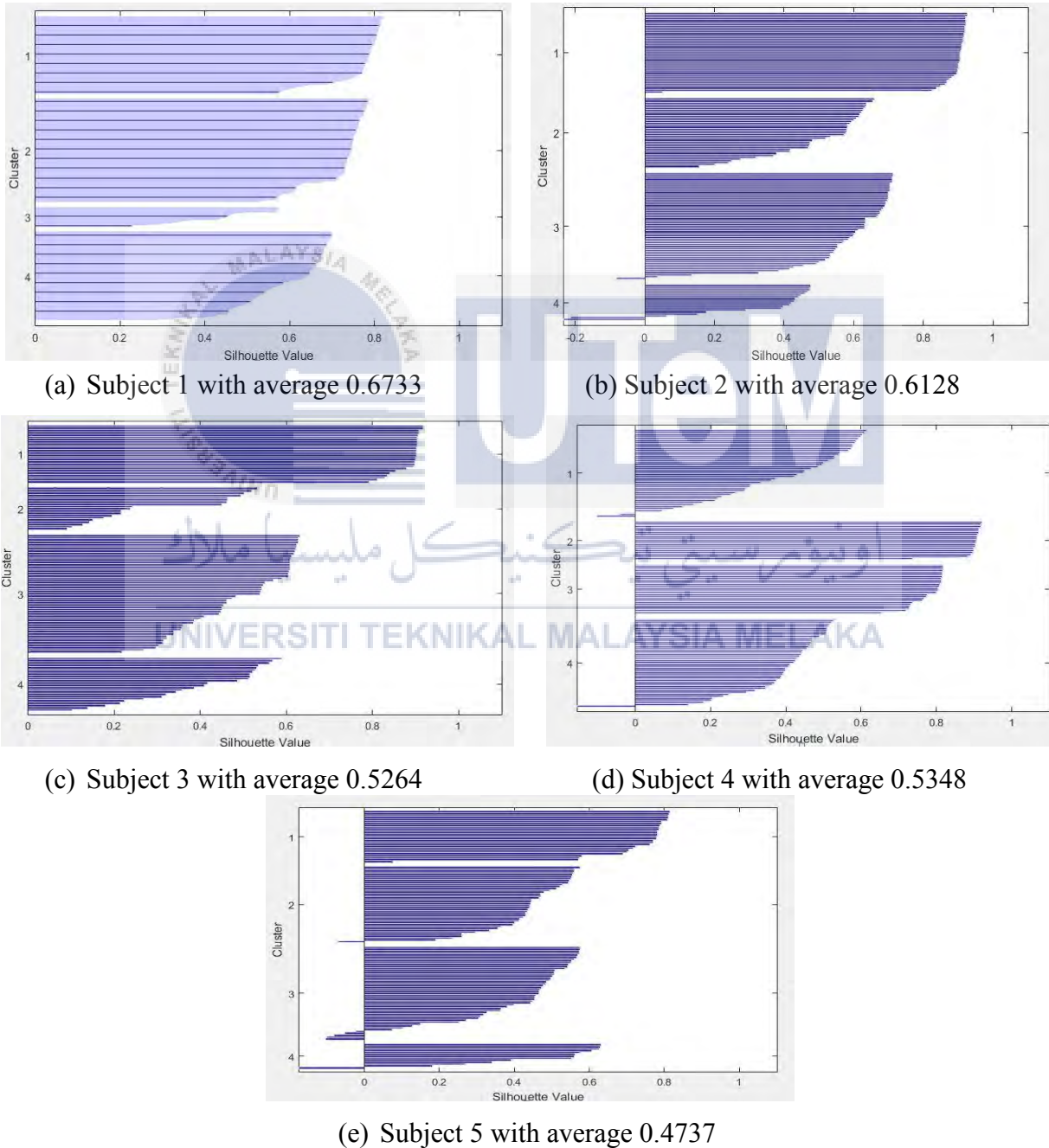
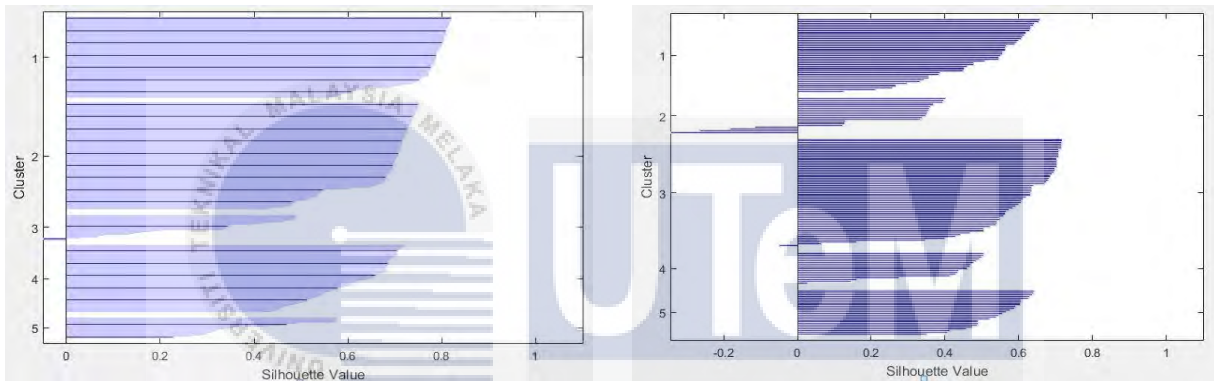


Figure 4.11: Graphs of Silhouette for  $k = 4$

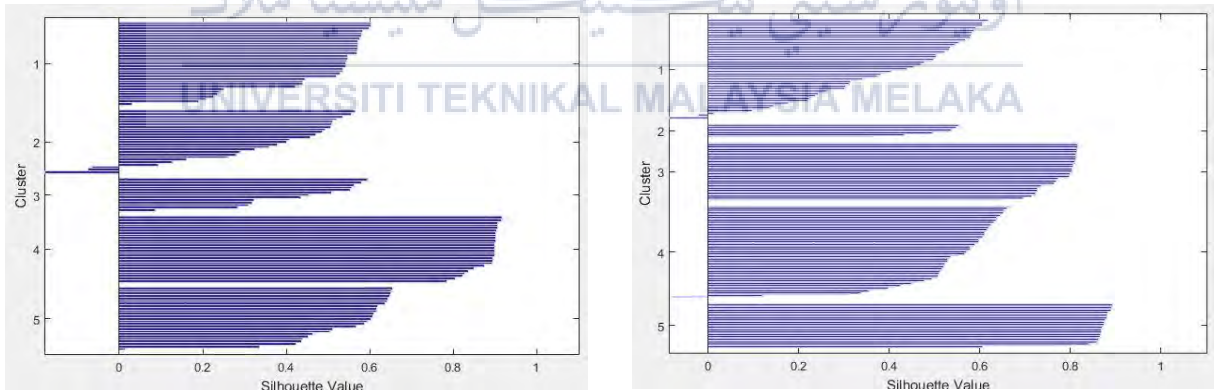
Figure 4.11 shows the Silhouette plot of k-means clustering for number of cluster,  $k$  equals to 4. In this graphs, it is shown that the first cluster, for Figure 4.11(b)(d) and (e), some of the points contain with low silhouette value indicates that the points are poorly classified. The negative silhouette value can also be seen in the third cluster and fourth cluster. Therefore, 4 cluster is not suitable in clustering human walking data.

#### 4.3.1.4 Analysis for $k = 5$ with average Silhouette value



(a) Subject 1 with average 0.6332

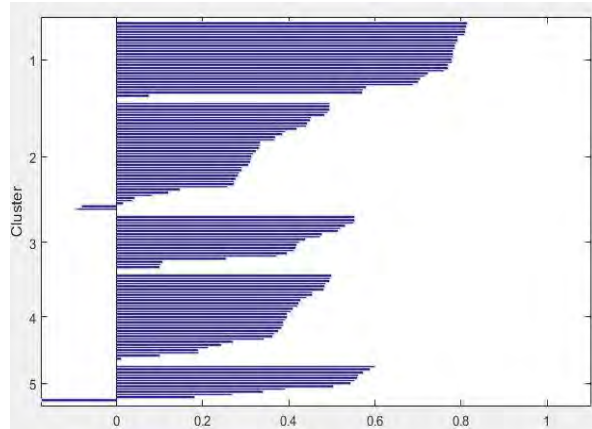
(b) Subject 2 with average 0.4905



(c) Subject 3 with average 0.5402

(d) Subject 4 with average 0.5769



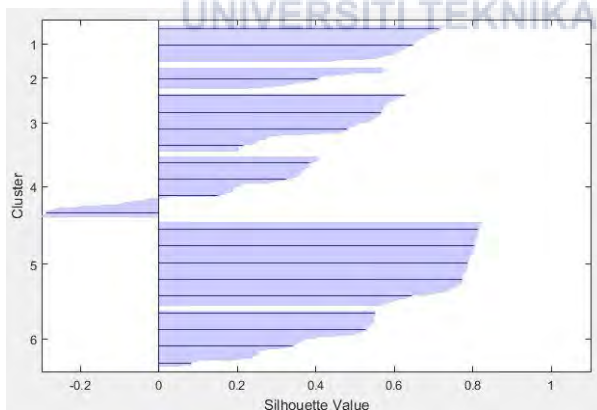


(e) Subject 5 with average 0.4308

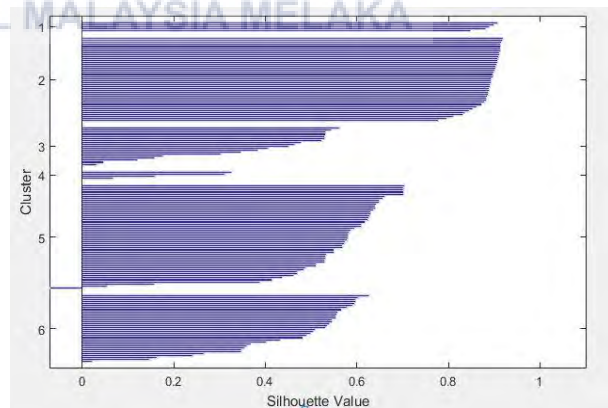
Figure 4.12: Graphs of Silhouette for  $k = 5$ 

Figure 4.11 shows the Silhouette plot of k-means clustering for number of cluster,  $k$  equals to 5. In this figure, it shows that all the graphs for each subject containing negative silhouette value within a cluster. As shown, Figure 4.12(b) has the lowest silhouette value which is below -0.2 for the second cluster. This can be concluded that when number of cluster is 5, many points are assigned into the wrong cluster.

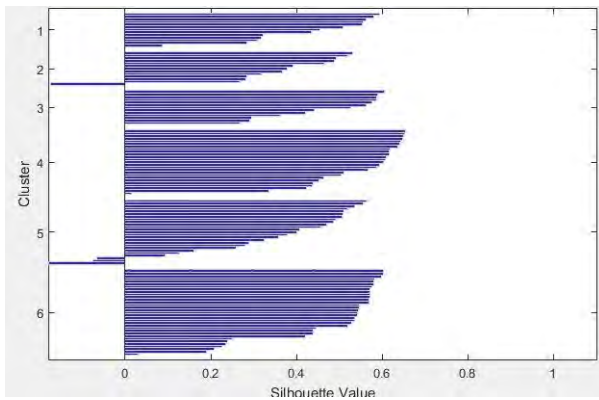
#### 4.3.1.5 Analysis for $k = 6$ with average Silhouette value



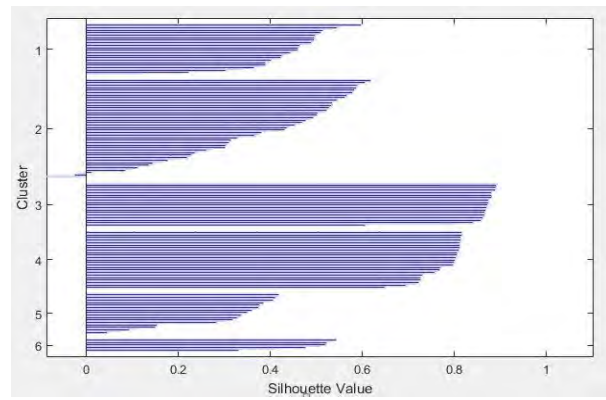
(a) Subject 1 with average 0.4839



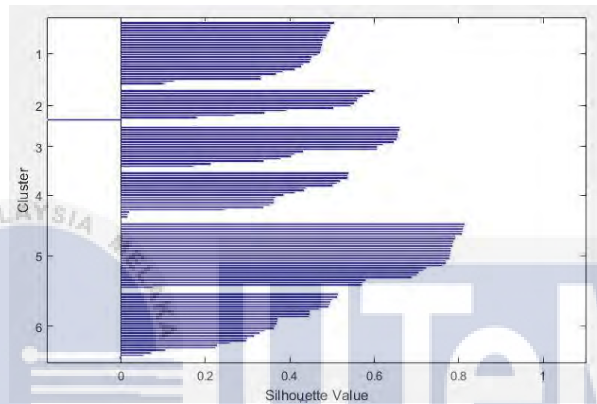
(b) Subject 2 with average 0.6015



(c) Subject 3 with average 0.4401



(d) Subject 4 with average 0.5258



(e) Subject 5 with average 0.4721

Figure 4.13: Graphs of silhouette for  $k = 6$ 

Figure 4.11 shows the Silhouette plot of k-means clustering for number of cluster,  $k$  equals to 6. In this figure, the negative silhouette value are occurred in every graph of the subjects. This indicates that the points on some clusters are not goodly classified and do not belong to the cluster of any means.

### 4.3.2 Conclusion from the analysis

By comparing silhouette graph to the k-means clustering graph, it can be concluded that the suitable number of cluster for clustering human walking data is 3. This is because in silhouette graph, the thickness of each cluster is same as the points are distributed equally to each cluster. Eventhough there are negative silhouette values when  $k=3$ , the negative value are caused by the noises of the human walking data.

## CHAPTER 5

### CONCLUSION AND RECOMMENDATION

#### 5.1 Conclusion

There are three objectives of this project that can be summarized. Firstly is to develop an experimental setup to collect human walking data. In order to collect data from human walking, an experiment is set-up by implementing a hardware to human and asked them to walk on a specific path. From the results obtained from the experiment, a clustering method is carried out to classify human walking data in order to achieve the second objective of this project which is to develop a method to classify an unclassified human walking data by using unsupervised learning. The unsupervised learning method is k-means clustering method. An analysis is carried out to observe the k-means algorithm method which will be then determined the suitable number of cluster. Besides that, performance evaluation method is carried out to evaluate the performance on k-means clustering based on number of k cluster. the method used is Silhouette Coefficient method. The result obtained from the performance measurement shows that the suitable number of k cluster for clustering human walking data is 3.

Based on the results gained, in future research, the k-means clustering will be improved by adding fuzzy logic algorithm. This is because k-means algorithm have many limitation and disadvantages in clustering the dataset. By adding fuzzy logic algorithm, the limitation can be widen and the disadvantages will be minimized in order to get a better result.

## 5.2 Limitation of the project

The limitation of this project is k-means cannot be used if the dataset has natural cluster that are non-spherical because k-means only works on 2 dimensional data. Next, the results of k-means may be vary as different results will yield if the algorithm was run a several times. Therefore, the results obtained is not accurate aand lack of consistency.



## REFERENCES

- [1] A. . Fallis, *Discovering knowledge in data*, vol. 53, no. 9. 2013.
- [2] M. Computing, “Comparative Study of Various Clustering Techniques,” vol. 3, no. 10, pp. 497–504, 2014.
- [3] R. Harikumar, B. Vinoth, and G. Karthick, “Performance Analysis for Quality Measures Using K means Clustering and EM Models in Segmentation of Medical Images,” *Int. J. Soft Comput. Eng.*, vol. 1, no. 6, pp. 74–80, 2012.
- [4] P.-N. Tan, M. Steinbach, and V. Kumar, “Chap 8 : Cluster Analysis: Basic Concepts and Algorithms,” *Introd. to Data Min.*, p. Chapter 8, 2005.
- [5] T. Sasaki, D. Bršćić, and H. Hashimoto, “Human-observation-based extraction of path patterns for mobile robot navigation,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1401–1410, 2010.
- [6] C. A. Ramirez, M. Castelan, and G. Arechavaleta, “Multilinear decomposition of human walking paths,” *2010 10th IEEE-RAS Int. Conf. Humanoid Robot. Humanoids 2010*, pp. 492–497, 2010.
- [7] M. Nasir, C. P. Lim, S. Nahavandi, and D. Creighton, “A genetic fuzzy system to model pedestrian walking path in a built environment,” *Simul. Model. Pract. Theory*, vol. 45, pp. 18–34, 2014.
- [8] H. Kobayashi, H. Hashimoto, and M. Niitsuma, “An approach for extraction of human walking path in Intelligent Space,” *4th Int. Conf. Hum. Syst. Interact. HSI 2011*, pp. 281–286, 2011.
- [9] T. Kanda, D. F. Glas, M. Shiomi, and N. Hagita, “Abstracting peoples trajectories for social robots to proactively approach customers,” *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1382–1396, 2009.
- [10] T. M. Mitchell, *Machine Learning*, no. 1. 1997.
- [11] Y.-X. Meng, “The practice on using machine learning for network anomaly intrusion detection,” *2011 Int. Conf. Mach. Learn. Cybern.*, vol. 2, pp. 576–581, 2011.

- [12] A. J. Stimpson and M. L. Cummings, "Assessing intervention timing in computer-based education using machine learning algorithms," *IEEE Access*, vol. 2, pp. 78–87, 2014.
- [13] P. Dixit and G. I. Prajapati, "Machine learning in bioinformatics: A novel approach for DNA sequencing," *Int. Conf. Adv. Comput. Commun. Technol. ACCT*, vol. 2015-April, pp. 41–47, 2015.
- [14] M. Schlesinger and V. Hlavác, "Supervised and unsupervised learning.," *Artif. Intell.*, no. April, 2011.



## APPENDICES

### APPENDIX A: Program for calibration of MPU6050

```
// Arduino sketch that returns calibration offsets for MPU6050
// Version 1.1 (31th January 2014)
// Done by Luis Ródenas <luisrodenaslorda@gmail.com>
// Based on the I2Cdev library and previous work by Jeff Rowberg <jeff@rowberg.net>
// Updates (of the library) should (hopefully) always be available at
// https://github.com/jrowberg/i2cdevlib

// I2Cdev and MPU6050 must be installed as libraries
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

////////////////////// CONFIGURATION ////////////////////////
//Change this 3 variables if you want to fine tune the sketch to your needs.
int buffersize=1000; //Amount of readings used to average, make it higher to get more
precision but sketch will be slower (default:1000)
int acel_deadzone=8; //Accelerometer error allowed, make it lower to get more precision,
but sketch may not converge (default:8)
int giro_deadzone=1; //Giro error allowed, make it lower to get more precision, but
sketch may not converge (default:1)

// default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for InvenSense evaluation board)
// AD0 high = 0x69
//MPU6050 accelgyro;
MPU6050 accelgyro(0x68); // <-- use for AD0 high
```

```

int16_t ax, ay, az,gx, gy, gz;

int mean_ax,mean_ay,mean_az,mean_gx,mean_gy,mean_gz,state=0;
int ax_offset,ay_offset,az_offset,gx_offset,gy_offset,gz_offset;

////////////////////////////////// SETUP ////////////////////////////////////
void setup() {
  // join I2C bus (I2Cdev library doesn't do this automatically)
  Wire.begin();
  // COMMENT NEXT LINE IF YOU ARE USING ARDUINO DUE
  TWBR = 24; // 400kHz I2C clock (200kHz if CPU is 8MHz). Leonardo measured
  250kHz.

  // initialize serial communication
  Serial.begin(115200);

  // initialize device
  accelgyro.initialize();

  // wait for ready
  while (Serial.available() && Serial.read()); // empty buffer
  while (!Serial.available()){
    Serial.println(F("Send any character to start sketch.\n"));
    delay(1500);
  }
  while (Serial.available() && Serial.read()); // empty buffer again

  // start message
  Serial.println("\nMPU6050 Calibration Sketch");
  delay(2000);
  Serial.println("\nYour MPU6050 should be placed in horizontal position, with package
  letters facing up. \nDon't touch it until you see a finish message.\n");
  delay(3000);

```



```

// verify connection
Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" :
"MPU6050 connection failed");
delay(1000);
// reset offsets
accelgyro.setXAccelOffset(0);
accelgyro.setYAccelOffset(0);
accelgyro.setZAccelOffset(0);
accelgyro.setXGyroOffset(0);
accelgyro.setYGyroOffset(0);
accelgyro.setZGyroOffset(0);
}

```

```

//////////////////// LOOP //////////////////////////////////////

```

```

void loop() {
  if (state==0){
    Serial.println("\nReading sensors for first time...");
    meansensors();
    state++;
    delay(1000);
  }

```

```

  if (state==1) {
    Serial.println("\nCalculating offsets...");
    calibration();
    state++;
    delay(1000);
  }

```

```

  if (state==2) {
    meansensors();
    Serial.println("\nFINISHED!");
    Serial.print("\nSensor readings with offsets:\t");
    Serial.print(mean_ax);

```

```

Serial.print("\t");
Serial.print(mean_ay);
Serial.print("\t");
Serial.print(mean_az);
Serial.print("\t");
Serial.print(mean_gx);
Serial.print("\t");
Serial.print(mean_gy);
Serial.print("\t");
Serial.println(mean_gz);
Serial.print("Your offsets:\t");
Serial.print(ax_offset);
Serial.print("\t");
Serial.print(ay_offset);
Serial.print("\t");
Serial.print(az_offset);
Serial.print("\t");
Serial.print(gx_offset);
Serial.print("\t");
Serial.print(gy_offset);
Serial.print("\t");
Serial.println(gz_offset);
Serial.println("\nData is printed as: acelX acelY acelZ giroX giroY giroZ");
Serial.println("Check that your sensor readings are close to 0 0 16384 0 0 0");
Serial.println("If calibration was succesful write down your offsets so you can set them
in your projects using something similar to mpu.setXAccelOffset(youroffset)");
while (1);
}
}

////////////////////////////////////// FUNCTIONS ////////////////////////////////////////
void meansensors(){
long i=0,buff_ax=0,buff_ay=0,buff_az=0,buff_gx=0,buff_gy=0,buff_gz=0;

```

```

while (i<(buffersize+101)){
    // read raw accel/gyro measurements from device
    accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

    if (i>100 && i<=(buffersize+100)){ //First 100 measures are discarded
        buff_ax=buff_ax+ax;
        buff_ay=buff_ay+ay;
        buff_az=buff_az+az;
        buff_gx=buff_gx+gx;
        buff_gy=buff_gy+gy;
        buff_gz=buff_gz+gz;
    }
    if (i==(buffersize+100)){
        mean_ax=buff_ax/buffersize;
        mean_ay=buff_ay/buffersize;
        mean_az=buff_az/buffersize;
        mean_gx=buff_gx/buffersize;
        mean_gy=buff_gy/buffersize;
        mean_gz=buff_gz/buffersize;
    }
    i++;
    delay(2); //Needed so we don't get repeated measures
}
}

```

```

void calibration(){
    ax_offset=-mean_ax/8;
    ay_offset=-mean_ay/8;
    az_offset=(16384-mean_az)/8;

    gx_offset=-mean_gx/4;
    gy_offset=-mean_gy/4;
    gz_offset=-mean_gz/4;
    while (1){

```

```

int ready=0;
accelgyro.setXAccelOffset(ax_offset);
accelgyro.setYAccelOffset(ay_offset);
accelgyro.setZAccelOffset(az_offset);

accelgyro.setXGyroOffset(gx_offset);
accelgyro.setYGyroOffset(gy_offset);
accelgyro.setZGyroOffset(gz_offset);

meansensors();
Serial.println("...");

if (abs(mean_ax)<=acel_deadzone) ready++;
else ax_offset=ax_offset-mean_ax/acel_deadzone;

if (abs(mean_ay)<=acel_deadzone) ready++;
else ay_offset=ay_offset-mean_ay/acel_deadzone;

if (abs(16384-mean_az)<=acel_deadzone) ready++;
else az_offset=az_offset+(16384-mean_az)/acel_deadzone;

if (abs(mean_gx)<=giro_deadzone) ready++;
else gx_offset=gx_offset-mean_gx/(giro_deadzone+1);

if (abs(mean_gy)<=giro_deadzone) ready++;
else gy_offset=gy_offset-mean_gy/(giro_deadzone+1);

if (abs(mean_gz)<=giro_deadzone) ready++;
else gz_offset=gz_offset-mean_gz/(giro_deadzone+1);

if (ready==6) break;
}
}

```

## APPENDIX B: Program for collecting human walking data

```

// I2Cdev and MPU6050 must be installed as libraries, or else the .cpp/.h files
// for both classes must be in the include path of your project
#include "I2Cdev.h"

#include "MPU6050_6Axis_MotionApps20.h"
// #include "MPU6050.h" // not necessary if using MotionApps include file

// Arduino Wire library is required if I2Cdev I2CDEV_ARDUINO_WIRE implementation
// is used in I2Cdev.h
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif

// class default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for SparkFun breakout and InvenSense evaluation board)
// AD0 high = 0x69
MPU6050 mpu;

```

```

//MPU6050 mpu(0x69); // <-- use for AD0 high

// uncomment "OUTPUT_READABLE_QUATERNION" if you want to see the actual
// quaternion components in a [w, x, y, z] format (not best for parsing
// on a remote host such as Processing or something though)
#define OUTPUT_READABLE_QUATERNION

// uncomment "OUTPUT_READABLE_EULER" if you want to see Euler angles
// (in degrees) calculated from the quaternions coming from the FIFO.
// Note that Euler angles suffer from gimbal lock (for more info, see
// http://en.wikipedia.org/wiki/Gimbal_lock)
#define OUTPUT_READABLE_EULER

// uncomment "OUTPUT_READABLE_YAWPITCHROLL" if you want to see the yaw/
// pitch/roll angles (in degrees) calculated from the quaternions coming
// from the FIFO. Note this also requires gravity vector calculations.
// Also note that yaw/pitch/roll angles suffer from gimbal lock (for
// more info, see: http://en.wikipedia.org/wiki/Gimbal_lock)
#define OUTPUT_READABLE_YAWPITCHROLL

// uncomment "OUTPUT_READABLE_REALACCEL" if you want to see acceleration
// components with gravity removed. This acceleration reference frame is
// not compensated for orientation, so +X is always +X according to the
// sensor, just without the effects of gravity. If you want acceleration
// compensated for orientation, us OUTPUT_READABLE_WORLDACCEL instead.
#define OUTPUT_READABLE_REALACCEL

// uncomment "OUTPUT_READABLE_WORLDACCEL" if you want to see acceleration
// components with gravity removed and adjusted for the world frame of
// reference (yaw is relative to initial orientation, since no magnetometer
// is present in this case). Could be quite handy in some cases.
#define OUTPUT_READABLE_WORLDACCEL

```

```

#define LED_PIN 13 // (Arduino is 13)
bool blinkState = false;

// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 = success, !0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
Quaternion q; // [w, x, y, z] quaternion container
VectorInt16 aa; // [x, y, z] accel sensor measurements
VectorInt16 aaReal; // [x, y, z] gravity-free accel sensor measurements
VectorInt16 aaWorld; // [x, y, z] world-frame accel sensor measurements
VectorFloat gravity; // [x, y, z] gravity vector
float euler[3]; // [psi, theta, phi] Euler angle container
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector

// packet structure for InvenSense teapot demo
uint8_t teapotPacket[14] = {'$', 0x02, 0,0, 0,0, 0,0, 0,0, 0x00, 0x00, '\r', '\n' };
static char outstr[15];

//
=====

// === INTERRUPT DETECTION ROUTINE ===
//
=====

volatile bool mpuInterrupt = false; // indicates whether MPU interrupt pin has gone high
void dmpDataReady() {
    mpuInterrupt = true;
}

```

```

//
=====
// ===          INITIAL SETUP          ===
//
=====

void setup() {

    // join I2C bus (I2Cdev library doesn't do this automatically)
    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
        Wire.begin();
        TWBR = 24; // 400kHz I2C clock (200kHz if CPU is 8MHz)
    #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
        Fastwire::setup(400, true);
    #endif

    // initialize serial communication
    // (115200 chosen because it is required for Teapot Demo output, but it's
    // really up to you depending on your project)
    Serial.begin(9600);
    while (!Serial); // wait for Leonardo enumeration, others continue immediately

    // NOTE: 8MHz or slower host processors, like the Teensy @ 3.3v or Arduino
    // Pro Mini running at 3.3v, cannot handle this baud rate reliably due to
    // the baud timing being too misaligned with processor ticks. You must use
    // 38400 or slower in these cases, or use some kind of external separate
    // crystal solution for the UART timer.

    // initialize device

```



```

Serial.println(F("Initializing I2C devices..."));
mpu.initialize();

// verify connection
Serial.println(F("Testing device connections..."));
Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") :
F("MPU6050 connection failed"));

// wait for ready
/* Serial.println(F("\nSend any character to begin DMP programming and demo: "));
while (Serial.available() && Serial.read()); // empty buffer
while (!Serial.available()); // wait for data
while (Serial.available() && Serial.read()); // empty buffer again
*/
// load and configure the DMP
Serial.println(F("Initializing DMP..."));
devStatus = mpu.dmpInitialize();

// supply your own gyro offsets here, scaled for min sensitivity
mpu.setXGyroOffset(24);
mpu.setYGyroOffset(16);
mpu.setZGyroOffset(9);
mpu.setZAccelOffset(1897);

// make sure it worked (returns 0 if so)
if (devStatus == 0) {
  // turn on the DMP, now that it's ready
  Serial.println(F("Enabling DMP..."));
  mpu.setDMPEnabled(true);

  // enable Arduino interrupt detection
  Serial.println(F("Enabling interrupt detection (Arduino external interrupt 0)..."));
  attachInterrupt(0, dmpDataReady, RISING);
  mpuIntStatus = mpu.getIntStatus();

```

```

// set our DMP Ready flag so the main loop() function knows it's okay to use it
Serial.println(F("DMP ready! Waiting for first interrupt..."));
dmpReady = true;

// get expected DMP packet size for later comparison
packetSize = mpu.dmpGetFIFOpacketSize();
} else {
  // ERROR!
  // 1 = initial memory load failed
  // 2 = DMP configuration updates failed
  // (if it's going to break, usually the code will be 1)
  Serial.print(F("DMP Initialization failed (code "));
  Serial.print(devStatus);
  Serial.println(F(""));
}

// configure LED for output
pinMode(LED_PIN, OUTPUT);
}

//
=====
// ==          MAIN PROGRAM LOOP          ==
//
=====

void loop(){

  // if programming failed, don't try to do anything
  if (!dmpReady) return;

  // wait for MPU interrupt or extra packet(s) available
  while (!mpuInterrupt && fifoCount < packetSize) {

```

```

// other program behavior stuff here
// .
// .
// .
// if you are really paranoid you can frequently test in between other
// stuff to see if mpuInterrupt is true, and if so, "break;" from the
// while() loop to immediately process the MPU data
// .
// .
// .
}

// reset interrupt flag and get INT_STATUS byte
mpuInterrupt = false;
mpuIntStatus = mpu.getIntStatus();

// get current FIFO count
fifoCount = mpu.getFIFOCount();

// check for overflow (this should never happen unless our code is too inefficient)
if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
  // reset so we can continue cleanly
  mpu.resetFIFO();
  //Serial.println(F("FIFO overflow!"));

// otherwise, check for DMP data ready interrupt (this should happen frequently)
} else if (mpuIntStatus & 0x02) {
  // wait for correct available data length, should be a VERY short wait
  while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

  // read a packet from FIFO
  mpu.getFIFOBytes(fifoBuffer, packetSize);

  // track FIFO count here in case there is > 1 packet available

```

```

// (this lets us immediately read more without waiting for an interrupt)
fifoCount -= packetSize;

#ifdef OUTPUT_READABLE_QUATERNION
    // display quaternion values in easy matrix form: w x y z
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    Serial.begin(9600);
    Serial.print("quat\t");
    Serial.print(q.w);
    Serial.print("\t");
    Serial.print(q.x);
    Serial.print("\t");
    Serial.print(q.y);
    Serial.print("\t");
    Serial.println(q.z);
#endif

#ifdef OUTPUT_READABLE_EULER
    // display Euler angles in degrees
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetEuler(euler, &q);
    Serial.print("euler\t");
    Serial.print(euler[0] * 180/M_PI);
    Serial.print("\t");
    Serial.print(euler[1] * 180/M_PI);
    Serial.print("\t");
    Serial.println(euler[2] * 180/M_PI);
#endif

#ifdef OUTPUT_READABLE_YAWPITCHROLL
    // display Euler angles in degrees
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);

```

```

Serial.print("ypr\t");
Serial.print(ypr[0] * 180/M_PI);
Serial.print("\t");
Serial.print(ypr[1] * 180/M_PI);
Serial.print("\t");
Serial.println(ypr[2] * 180/M_PI);
#endif

#ifdef OUTPUT_READABLE_REALACCEL
// display real acceleration, adjusted to remove gravity
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetAccel(&aa, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);
mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
Serial.print("areal\t");
Serial.print(aaReal.x);
Serial.print("\t");
Serial.print(aaReal.y);
Serial.print("\t");
Serial.println(aaReal.z);
#endif

#ifdef OUTPUT_READABLE_WORLDACCEL
// display initial world-frame acceleration, adjusted to remove gravity
// and rotated based on known orientation from quaternion
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetAccel(&aa, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);
mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
mpu.dmpGetLinearAccelInWorld(&aaWorld, &aaReal, &q);
Serial.print("aworld\t");
Serial.print(aaWorld.x);
Serial.print("\t");
Serial.print(aaWorld.y);

```

```

Serial.print("\t");
Serial.println(aaWorld.z);
#endif

// blink LED to indicate activity
blinkState = !blinkState;
digitalWrite(LED_PIN, blinkState);
//delay(100);
}
}

```



### APPENDIX C: Silhouette Coefficient MATLAB code

```

rng default % For reproducibility
load datafile.csv
size(datafile)

[idx3,cent3,sumdist] =
kmeans(datafile,3,'Distance','cityblock','Display','iter','Replicates',5);
figure
[silh3,h] = silhouette(datafile,idx3,'cityblock');
h = gca;

```

```
h.Children.EdgeColor = [.8 .8 1];  
xlabel 'Silhouette Value'  
ylabel 'Cluster'  
mean(silh3)
```

