



University of
Salford
MANCHESTER

MACHINE LEARNING AND AUTONOMOUS SYSTEM FOR HUMAN GAIT ANALYSIS BASED ON WALK SPEED

Abdulhakim Elkurdi

School of Computing, Science and Engineering

University of Salford, Manchester, UK

Submitted in Partial Fulfilment of the Requirement of the
Degree of Doctor of Philosophy, Jan 2020

TABLE OF CONTENTS

TABLE OF CONTENTS.....	i
LIST OF FIGURES	v
LIST OF TABLES.....	ix
LIST OF ABBREVIATIONS.....	xi
ACKNOWLEDGEMENT	xiii
ABSTRACT.....	xiv
1. INTRODUCTION AND MOTIVATION.....	1
1.1 Introduction	1
1.2 Research Motivation	2
1.3 Research Scope	5
1.4 Research Question.....	6
1.5 Research Aim and Objectives	6
1.6 Proposed Methods	7
1.7 Research Contribution.....	8
1.8 Thesis Structure.....	8
2. THE EFFECTS OF WALKING SPEED ON GAIT PARAMETERS.....	10
2.1 Introduction	10
2.2 Gait Parameters Changes due to Walking Speeds	10
2.2.1 Timed Get Up and Go (TUG).....	11
2.2.2 Timed 25-foot walk (T25FW)	12
2.2.3 Multiple Sclerosis Walking Scale (Rating Scales)	13
2.3 Residential Care for Elderly People	14
2.4 Sensor Technologies	16
2.4.1 Non-vision-based Tracking System.....	16
2.4.2 Vision Based Tracking System.....	16
2.4.3 Microsoft Kinect Sensor Based Depth Data	17

2.4.4	Gait Analysis using Microsoft Kinect Sensor.....	19
2.5	Summary	20
3.	LITERATURE REVIEW	22
3.1	Introduction	22
3.2	Skeletal Data Collection.....	23
3.3	Smoothing and Filtering Data Techniques.....	24
3.3.1	Moving Average Filter.....	24
3.3.2	Exponential Moving Average Filters.....	25
3.3.3	Median Filter.....	25
3.3.4	Savitzky–Golay Filter	25
3.3.5	Local Regression Filter	26
3.3.6	Kalman Filter	27
3.4	Gait Cycle Determination.....	28
3.5	Gait Features Extraction.....	29
3.5.1	Spatiotemporal Gait Analysis	30
3.5.2	Kinematic Gait Parameters	31
3.6	Gait Features Reduction	32
3.6.1	Principal Component Analysis	32
3.7	Data Classification Techniques.....	33
3.7.1	Decision Tree	34
3.7.2	Support Vector Machine Approach	35
3.7.3	k-Nearest Neighbour Approach	36
3.7.4	Discriminant Analysis Classification.....	38
3.8	Cross validation technique	39
3.8.1	Holdout Method	39
3.8.2	Leave-P-Out Cross Validation.....	40
3.8.3	K-Fold Cross Validation	40

3.9	System performance evaluation	41
3.9.1	Receiver Operating Characteristics.....	42
3.9.2	Confusion Matrix	43
3.10	Summary	45
4.	RESEARCH METHODS AND PROPOSED TECHNIQUES.....	47
4.1	Introduction	47
4.2	Instruments and Data Acquisition	47
4.2.1	Kinect Camera View	48
4.3	Framework and Proposed Approaches.....	50
4.3.1	Skeletal Data Smoothing and Filtering	51
4.3.2	Gait Cycle Determination	55
4.3.3	Features Extraction	55
4.3.4	Gait Features Reduction.....	64
4.3.5	Gait Data Classification	67
4.3.6	K-Fold Cross Validation	74
4.3.7	System Evaluation	75
4.4	Summary	78
5.	EXPERIMENTAL RESULTS OF IMPLEMENTED SYSTEM AND DISCUSSION ..	80
5.1	Introduction	80
5.2	The Proposed System AM/CE	80
5.3	Experiment 1- Skeleton Data Smoothing Techniques	81
5.3.1	Original data.....	81
5.3.2	Filtered data	82
5.4	Experiment 2- Kinect camera validation.....	89
5.4.1	Forward walk to the Kinect camera view	89
5.4.2	Parallel walk to the Kinect camera view.....	90
5.5	Experiment 3- Kinect for timed walk test	91

5.6	Experiment 4- Gait features extraction	92
5.6.1	Spatiotemporal gait analysis	93
5.6.2	Proposed method (Amplitude Modulation)	95
5.7	Experiment 5- AM&FM Gait Features Extraction	98
5.8	Experiment 6- Gait Features Classification	99
5.8.1	AM/CE System Implementation.....	100
5.9	AM/CE System Performance	106
5.10	Summary	107
6.	CONCLUSIONS AND FUTURE WORK.....	110
6.1	Research Aim and Scope.....	110
6.2	Research Objectives Definition.....	111
6.3	Contributions of the Research	113
6.4	Research Limitations.....	113
6.5	Future work	114
6.6	Summary	115
	REFERENCES	116
	APPENDICES	133
	Appendix-A.....	133
	Appendix -B	136
	Appendix C	138
	Appendix D	143
	PUBLICATIONS.....	165

LIST OF FIGURES

Figure 1.1: UK Population of the Elderly from 1974 - 2014 (Source: UK National Archives).
2

Figure 1.2: The UK Expenditure on Health Care for People over 65 (Source: UK Public Spending)3

Figure 1.3: Quality of Life VS Cost of Care (Source: MacIntosh et al., 2014).....4

Figure 2.1: Timed-Up & Go Test 12

Figure 2.2: Timed 25-Foot Walking 13

Figure 2.3: The Main Components of Microsoft Kinect 17

Figure 2.4: Human Body Joints: (a) Skeleton of Human Body (b) Hierarchal Human Body Joints 18

Figure 3.1: One Complete Gait Cycle that is Determined between 2 Strike Heels for the Same Foot, and Generally a Composite of 2 Phases; the Stance Phase and Swing Phase 60/ 40% of a Gait Cycle Period, Respectively.29

Figure 3.2: The Vertical Displacement of the Spine-Base during a Walk Process. [Source: Kinesiology Scientific Basis of Human Motion, 12th Edition by Hamilton, Luttgens & Weimar (1991)].....29

Figure 3.3: Decision tree construction34

Figure 3.4: Possible Hyperplane in SVM Algorithm35

Figure 3.5: The Principle Work of k-NN Approach37

Figure 3.6: Holdout Method for Single-Training/Testing Sets.....39

Figure 3.7: Leave One Out Cross Validation.....40

Figure 3.8: k-Fold Cross Validation41

Figure 3.9: Three ROC Curves that Correspond (Right Side) to Overlapping Distributions (Left Side) (Source: Schwartz, 2012)43

Figure 3.10: Confusion Matrix for Two Classes with Some Measures44

Figure 4.1: Kinect v2 camera view (a) horizontal view (b) vertical view48

Figure 4.2: Horizontal Angle View of Kinect Camera v249

Figure 4.3: Vertical Angle View of Kinect Camera v250

Figure 4.4: General Block Diagram of a Human Gait Analysis with Six Stages50

Figure 4.5: The Relationship between Distance, Velocity and Acceleration58

Figure 4.6: Displacement-Time Graph58

Figure 4.7: AM-Modified Gait Length Signal Generation59

Figure 4.8: Amplitude Modulation Signal in Time Domain.....	59
Figure 4.9: AM-Signal Spectrum Contains Three Components.....	60
Figure 4.10: Frequency Modulation Signal in Time Domain.....	62
Figure 4.11: Spectrum Range in FM-NB Modulated Signal.....	64
Figure 4.12: The (2 1 3) Convolutional Encoder with Code Rate 1/2.....	69
Figure 4.13: The State Diagram of the 1/2 Rate Convolutional Encoder.....	70
Figure 4.14: Trellis Diagram Example to Encode Input Bits [1001].....	71
Figure 4.15: Path Metric using Threshold T1.....	72
Figure 4.16: Path Metric using Threshold T2.....	73
Figure 4.17: Path Metric using Threshold T3.....	73
Figure 4.18: K-Fold Cross-Validation Method.....	74
Figure 5.1: Right Ankle Movement Signal, which is Affected by Noise (Spikes), the Signal Shows Strike Heel and Toe-Off on the Curve that Corresponds with the Concave Down and Concave Up, Respectively.....	82
Figure 5.2: Filtering and Smoothing of Right Ankle Movement Signal using Six Different Filter Techniques; Original Data (black line), SG Filter (pink line), RLOESS Filter (blue line), MA Filter (green line), Median Filter (purple line), Kalman Filter (blue line) and Exponential MA (red dot line).....	83
Figure 5.3: The Output Curve of the Savitsky-Golay Filter, Input Curve is a Black Line, Output Curve is a Red Line.....	83
Figure 5.4: The Output Curve of Local Regression Filter (RLOESS), Input Curve is a Black Line, Output Curve is a Red Line.	83
Figure 5.5: The Output Curve of Moving Average Filter (MA) for Right Ankle Movement, Input Curve is a Black Line, Output Curve is a Red Line.	84
Figure 5.6: The Output Curve of Median Filter for Right Ankle Movement, Input Curve is a Black Line, Output Curve is a Red Line.....	84
Figure 5.7: The Output Curve of Kalman Filter for Right Ankle Movement, Input Curve is a Black Line, Output Curve is a Red Line.....	84
Figure 5.8: The Output Curve of Exponential MA Filter for Right Ankle Movement, Input Curve is a Black Line, Output Curve is a Red Line.....	85
Figure 5.9: The Non-Linear Regression (a) Original Data - Filtered Data Relation in Savitzky-Golay Filter (black dots). While, the Curve of the 1 th Degree Polynomial Model is Plotted (blue line) (b) is the Residuals between Two Curves.	86

Figure 5.10: The Non-Linear Regression (a) Original Data - Filtered Data Relation in Median Filter (black dots). While, (blue line) is a 1th Degree Polynomial Model, (b) is the Residuals between Two Curves.....86

Figure 5.11: The Non-Linear Regression (a) Original Data- Filtered Data Relation in RLOESS Filter (black dots). While, (blue line) is a 1th Degree Polynomial Model, (b) is the Residuals between Two Curves.....86

Figure 5.12: The Non-Linear Regression (a) Original Data - Filtered Data Relation in Savitzky-Golay Filter (black dots). While, the Curve of the 9th Degree Polynomial Model is Plotted (blue line) (b) is the Residuals between Two Curves.87

Figure 5.13: The Non-Linear Regression (a) Original Data - Filtered Data Relation in Median Filter (black dots). While, (blue line) is a 9th Degree Polynomial Model, (b) is the Residuals between Two Curves.....87

Figure 5.14: The Non-Linear Regression (a) Original Data - Filtered Data Relation in RLOESS Filter (black dots). While, (blue line) is a 9th Degree Polynomial Model, (b) is the Residuals between Two Curves.....87

Figure 5.15: Timed Walk Test in Real Time using Kinect Camera for 3 meters; Skeleton Data collection During Walk Test.....92

Figure 5.16: Three Walk Speeds Categorisation (a) Group One ($S \leq 0.55$), (b) Group Two ($0.55 < S < 1$), and (c) Group Three ($S \geq 1 \text{ meter/sec}$).93

Figure 5.17: Gait Length Signal Generation from the Distance between (a) Right Ankle (dashed line) and Left Ankle (solid line) During Walk, (b) Generated Gait Signal in Absolute Value and (c) Without Absolute Value.94

Figure 5.18: The Determination of a Complete Gait Cycle for Extracting Step Length (SL), Stride Length (STL), Double Support Time (DS), Swing Time (SW), Stance Time (ST) and Gait Cycle Time (SW+ST).94

Figure 5.19: Gait Length Signal Conversion into AM Domain.....95

Figure 5.20: Sampling Frequency Adjustment for Reference Signal as: $f_s =$ (a) 90 Hz, (b) 100 Hz and (c) 110 Hz.....96

Figure 5.21: The AM-Modified Gait Signal in the Time Domain Changes its Features According to the Walk Speed.96

Figure 5.22: AM-Modified Gait Signal in the Frequency Domain Changes its Features According to the Walk Speed.97

Figure 5.23: A 5-Fold Cross Validation for Dataset Containing 120 Samples.102

Figure 5.24: A 10-Fold Cross Validation for Dataset Containing 120 Samples. 102

Figure 5.25: Error Rate Calculation of CE in 3-Classes of Gait Speed Classification when 5-Fold CV; (a) to (e) Represent Folds from 1 to 5. 103

Figure 5.26: Error Rate Calculation of CE in 3-Classes of Gait Speed Classification when 10-Fold CV; (a) to (j) Represent Folds from 1 to 10. 104

LIST OF TABLES

Table 3-1: The Main Measures that are Commonly Used in Model Evaluation using the Confusion Matrix.....	44
Table 4-1: The Confusion Matrix for Three Classes	76
Table 5-1: The Simple Correlation Coefficient for Filtered Data to Original Data.....	85
Table 5-2: Polynomial (9) Curve Fitting to Three Types of Filtered Data; Savitzky-Golay, Local Regression and Median Filters	88
Table 5-3: The Length and Width of Gait Step Data are Provided from Kinect and ART in Forward Walk to Kinect View, when Partitions are Labelled by 0.45 m and 0.17 m.	90
Table 5-4: The Length and Width of Gait Step Data are Provided from Kinect and ART in Forward Walk to Kinect View, when the Partitions are Labelled by 0.55 m and 0.20 m.	90
Table 5-5: The Length and Width of the Gait Step Data are Provided from Kinect and ART in Parallel Walk to Kinect View, when Partitions are Labelled by 0.45 m and 0.17 m.	90
Table 5-6: The Length and Width of Step Data are Provided from Kinect and ART in Parallel Walk to Kinect View, when Partitions are Labelled by 0.55 m and 0.20 m.....	91
Table 5-7: Timed Walk Test for Five Trials using Two Methods; Stop Watch and Kinect Camera	92
Table 5-8: The Evolution of Data Gait Classification using Spatiotemporal Analysis for Gait Feature Extraction.....	95
Table 5-9: The Evolution of Data Gait Classification using the AM Technique for Gait Feature Extraction.....	97
Table 5-10: The Evolution of Data Classification using AM Tech for Gait Feature Extraction	98
Table 5-11: The Evolution of Data Classification using FM Tech for Gait Feature Extraction	99
Table 5-12: Dataset Reduction Based on Eigenvalues and Captured Variance of Total Data	101
Table 5-13: The Seven Gait Features from the AM-Modified Gait Signal are Reduced into Two Vectors using PCA.....	101
Table 5-14: Confusion Matrix for Three Classes in Case of 5-Fold CV	104
Table 5-15: Confusion Matrix for Three Classes in Case of 10-Fold CV	104
Table 5-16: CE Classifier Performance in Case of 5-Fold Cross Validation (k equals 5)	105

Table 5-17: CE Classifier Performance in 10-Fold Cross Validation (k equals 10) 105

Table 5-18: Supervised Classifiers Performance in both Cases of 5-Fold & 10-Fold Cross Validation..... 105

Table 5-19: AM/CE System Performance in Classification of Gait Pattern Changes..... 107

LIST OF ABBREVIATIONS

ADL	Activities of Daily Living
AM	Amplitude Modulation
AS	Amplitude Spectrum
CCD	Charged Coupled Device
CE	Convolutional Encoder
CMOS	Complementary Metal Oxide Semiconductor
CV	Cross Validation
DEL	Departmental Expenditure Limit
DT	Decision Tree
EDGE	Evaluation Database to Guide Effectiveness
EDSS	Expanded Disability Status Scale
EEG	Electroencephalography
EMG	Electromyography
ENoLL	European Network of Living Labs
EOG	Electrooculography
FM-NB	Frequency Modulation-Narrow Band
FN	False Negative
FP	False Positive
HD	Hamming Distance
GARS	Gait Abnormality Rating Scale
GIS	Geographic Information Systems

GSM	Global System Mobile
ICT	Information and Communication Technology
k-NN	k-Nearest Neighbour
MEMS	Microelectromechanical Systems
MS	Multiple Sclerosis
MS Kinect	Microsoft Kinect
OpenNI	Open Natural Interaction
OMs	Outcome Measures
PCA	Principle Component Analysis
RAI-MH	Resident Assessment Instrument-Mental
RGB	Red, Green, Blue
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristics
SDK	Software Development kit
SMSW	Short Maximum Speed Walk
SVM	Support Vector Machine
T-25FW	Timed 25 Foot Walk test
TP	True Positive
TN	True Negative
TUG	Timed Up and Go test
UN's	United Nationals

ACKNOWLEDGEMENT

I would like to thank my supervisor Professor Samia Neft-Meziani for the continuous support, her motivation, and knowledge during my PhD study. Professor Nefti-Meziani's office was always open for valuable advice and mentor for my PhD study.

Besides my supervisor, my sincere thanks also go to Dr. Tauseef Gulrez, Dr. Ipek Caliskanelli for their help and support. Dr Tauseef has provided me helpful advices, when he was my co-supervisor in the first year of my study. Dr. Caliskanelli has supported me for papers publication in second and third years. Especial thank to my co-supervisor in third year of my study Dr. Majeed Soufian for his guiding me in technical issues, mentoring me in research and giving me a deeper advice that I need for carrying out my PhD research.

I am very grateful to my wife, my sons who supported me in every possible way to see the completion of this work. Also, I would like to thank my brothers and sisters for their support and encouragement, and all others who have helped me throughout the research.

Abdulahkim Elkurdi.

ABSTRACT

The quality of life and cost of care for elderly people varies dramatically between those living independently and those receiving acute or long-term care, which takes place at home, in residential care or in hospital. The common aim of national health service providers is to keep elderly people safe at their own homes for as long as possible to promote independent living, increase their quality of life and reduce hospital costs. Hence, the application of autonomous sensing systems to enhance everyday life of such population will be valuable and has been considered here.

Recently, Microsoft Kinect v2 has been used for gait analysis systems, to perform data classification of gait pattern changes based on walking speeds. This system enables the tracking without the need of any markers. Moreover, the Kinect camera is considered a low-cost device, and is quick to install, even in an unprepared environment. However, the primary challenge of such a device is that it provides a low data rate which leads to a decrease in the quality of extracted features, compared to other Motion Capture Systems (MoCap). Furthermore, in the data classification stage, the performance of classification is greatly affected by the boundary between different classes which is called decision boundary. This raises other questions such as: how to weight the features from the class labels, and which kind of similarity metric can be used.

To improve the quality of features, the Amplitude Modulation (AM) and Convolutional Encoder (CE) can play a major role in detection and in ranking the gait pattern changes based on walking speed. For this purpose, the collected data is mapped into a higher frequency spectrum using the AM domain. Consequently, the “AM-modified gait signal” is produced to improve the quality of extracted gait features, by increasing the level of the frequency sampling rate.

In this research, the main novelty is the combination of Amplitude Modulation (AM) and Convolutional Encoder (CE) techniques in one system (AM/CE) in order to understand and identify the walking speed effects on gait parameters. The former is proposed to extract new gait features without the need to determine the gait cycle phases, while the latter is developed to classify gait data based on walking speeds. Therefore, the performance of the CE technique is improved efficiently in gait data classification by weighting the bit positions in

Hamming Distance (HD) length, which leads to an increase in the accuracy of measurement of the similarity metric.

1. INTRODUCTION AND MOTIVATION

This chapter presents the motivation, the main aim, objectives and the contributions of the research. It also briefly reviews the most relevant aspects of the study.

1.1 Introduction

Human gait is a clinical terminology which refers to locomotion performance during the walking process. Since there is a significant correlation between gait disorder and some cognitive diseases (Choi, Park, Lee, Yoo, Kim, Jang & Oh, 2019), gait analysis becomes a highly recommended approach for use in the discrimination between normal and abnormal gaits (Jarchi, Lee, Tamjidi, Mirzaei & Sanei, 2018). This analysis has been exploited widely in different applications, for instance, in rehabilitation treatments (Steultjens, Dekker, Van Baar, Oostendorp & Bijlsma, 2000), sports analysis (Wahab & Bakar, 2011), biomedical engineering (Gabel, Gilad-Bachrach, Renshaw & Schuster, 2012), and other medical purposes (Ries, Echternach, Nof & Gagnon Blodgett, 2009).

Recently, a 3D skeleton for the lower body limbs was retrieved using the Microsoft Kinect sensors. The widespread use of Kinect sensors is due to its capability of providing the joints' skeleton data without the need for markers to be placed on the body (Andersson & Araujo, 2015). Moreover, it is inexpensive and easy to install even in unequipped facilities. This make it an efficient approach for gait analysis by tracking elderly people in their homes, which might improve the quality of life for such residents (Soufian, Nefti- Meziani and Drake, 2020), (Malekmohamadi et al 2018). However, one of the key challenges that exists in these approaches is the necessity of gait cycle identification, i.e. detection of a gait cycle phases is required as a first task in gait analysis. Furthermore, the use of the Kinect camera in data collection might affect the measurements accuracy as it is considered a low data rate device.

In this research, many approaches have been examined (Elkurdi, Soufian & NeftiMeziani, 2018) for gait analysis of human walking behaviour to assess the level of the abnormal changes that could exist in gait pattern based on gait speed changes. Among these, the proposed AM/CE approach reduces the need to detect gait cycle phases, since during gait features extraction, it deals with the whole data collected on the AM domain.

The enhancement of extracted gait feature quality is important because the efficiency of extracted features can affect the accuracy of data classification. For this purpose, the combination of AM and CE techniques in one system is performed. First, the use of AM aims to convert a gait length signal into AM domain (Elkurdi, Caliskanelli and Nefti-Meziani, 2018), which enables the signals to be modified and analysed on the higher frequency's spectrum. This leads to increased accuracy of measurements especially during gait feature extraction. In the data classification stage, the CE technique is proposed for determining three decision classes by calculating the similarity matrices based on Hamming Distance (HD), where a low quantity of HD means a high similarity in class prediction.

1.2 Research Motivation

The rapid growth of the ageing population is caused by a reduction of fertility (Gavrilov and Heuveline, 2003). Notably, the population is ageing (over 60 years old) in most of the world's regions and this is anticipated to increase in the next decades. According to the World Data Bank (2011-2014), the ratio of old-age dependency (65 years old or over) to working-age (less than 65 years old) is growing, especially within developed countries. Figure 1.1 illustrates the proportion of elderly people in the UK, starting from mid-1974 until 2014 for three age categories. It is noticeable that the number of people over the age of 85 increased by almost four percent in 2014.

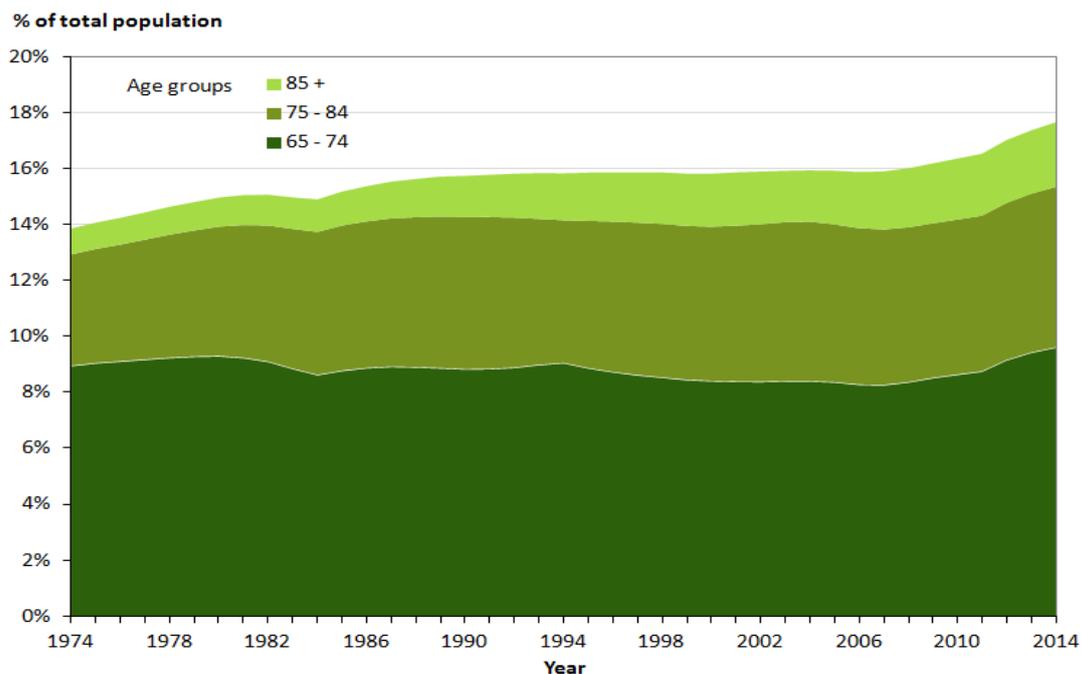


Figure 1.1: UK Population of the Elderly from 1974 - 2014 (Source: UK National Archives).

For all ages, there is a tangible rise in the population over the last four decades. For instance, in mid-1974, thirteen percent of the population was above 75, whereas by the end of 2014 this had increased to almost sixteen percent. The increase in the ageing population of the UK indicates that the human living period has become longer than ever before, with this achievement resulting in pride. However, with longevity, significant challenges have emerged due to sickness and health issues being prolonged (Brown, 2015). Therefore, of foremost importance is the provision of support for people during these health-related years. The consequences of longevity are reflected in the UK's expenditure on healthcare.

According to a recent report by the UK government, the largest number of hospital care users are the elderly, with 62% of total bed days in 2014/2015 (Humphries e al., 2016). Figure 1.2 shows an increase in healthcare costs for people over 65 from 2006 to 2016. At the same time, a demand for housing is also likely to increase due to the anticipated rise of the elderly population. However, existing housing standards may not always meet this change in needs, which could lead to a critical situation. This will put more pressure on hospital unless there are changes that can be adapted to the new housing needs. Such housing can assist individuals to remain living at home, which has the greatest potential to lead to cost savings. Consequently, their quality of life and end of life functioning will be improved. The support can potentially be applied in two different ways: by improving the quality of life or by providing high quality medical support

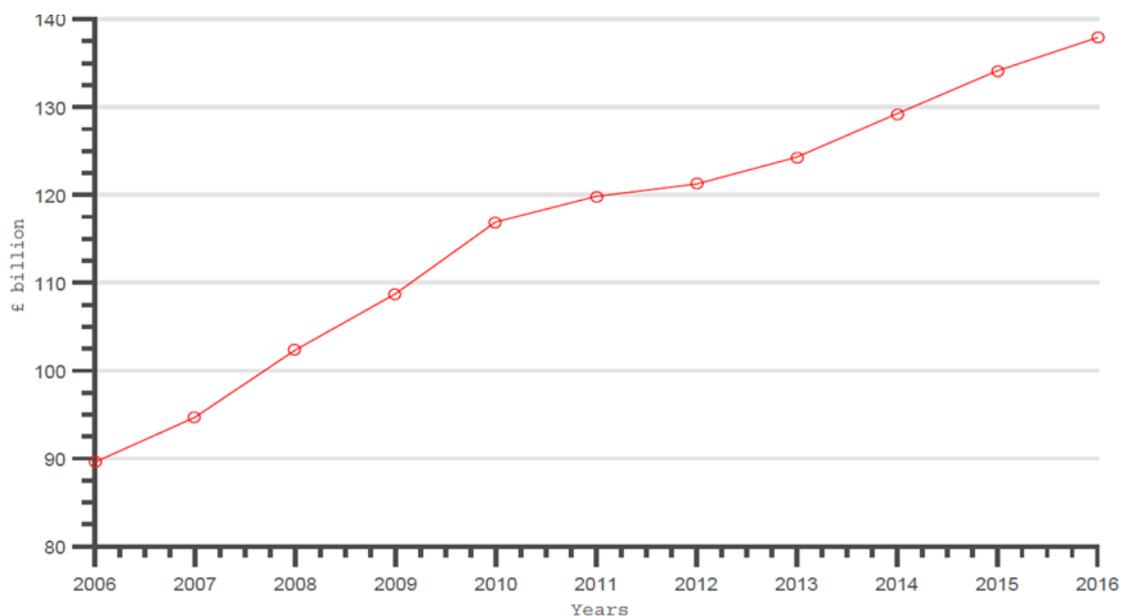


Figure 1.2: The UK Expenditure on Health Care for People over 65 (Source: UK Public Spending)

Figure 1.3 (MacIntosh et al., 2014) below, shows a correlation between the quality of life versus the cost of care. On the one hand, quality of life can be improved when elderly people are healthy, under disease prevention and living independently in their own homes. Whilst, quality of life is decreased if they are receiving acute care in hospitals. The cost of care is the highest when acute care is being received, whereas it is low for those who are healthy under disease prevention and living independently in their own homes. The life quality and cost of care varies dramatically between living independently and receiving acute care, which takes place in either a home or residential care. The target is to keep people safe in their own homes for as long as possible to promote independent living, increase their quality of life and reduce hospital costs.

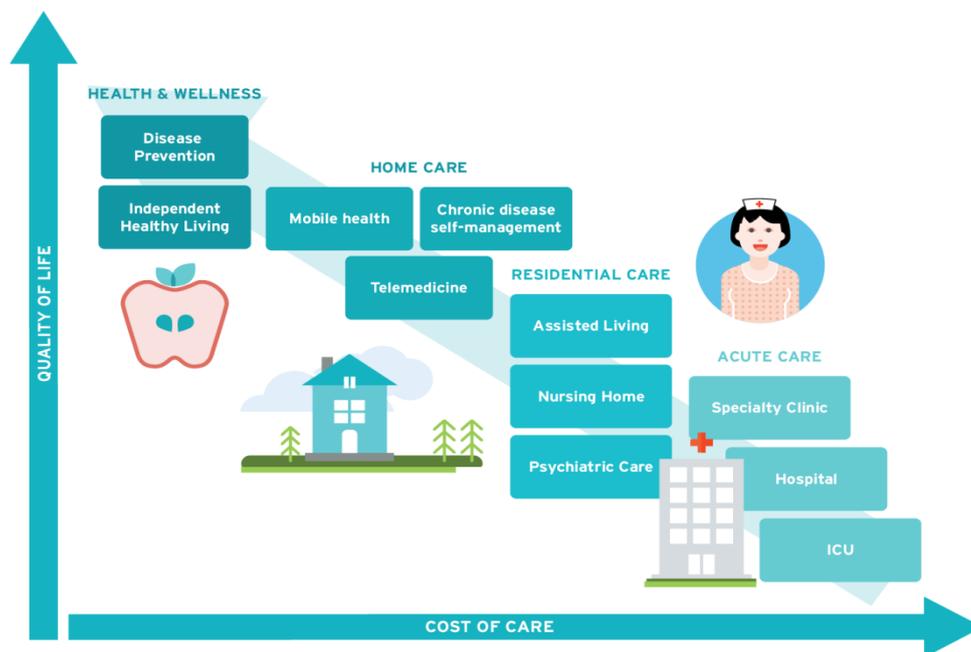


Figure 1.3: Quality of Life VS Cost of Care (Source: MacIntosh et al., 2014).

The expenditure of healthcare and an improvement in the quality of life for older people, along with the longevity challenges, are driving developments in the assistive living field, where technologies, smart applications and machine learning are involved. From this perspective, Ambient Assisted Living (AAL) is defined as “the use of information and communication technologies (ICT) in a person’s daily living and working environment to enable them to stay active longer, remain socially connected and live independently into old age” (www.aal-europe.eu) (Monekosso et al., 2015).

Furthermore, low-cost, technology-based solutions, including ambient living and remote healthcare management systems, are under development to tackle some of the aspects

of ageing. Such solutions aim to improve the quality of life and safety, while reducing treatment costs, as well as aiding healthcare systems to enable people to live safely in their own home for as long as possible. The most effective and indicative way to assess the quality of life and wellbeing is to track a subject's activities, for example through the monitoring of several walk patterns in their own environment. This insight into their daily activities (Malekmohamadi et al., 2018), could help to detect frailty (Bravo et al., 2018), neurological disorders (McGough et al., 2013) and even Mild Cognitive Impairment (Soufian et al., 2020). The subject's activity and walking speed are of paramount importance in such examinations. A person's characteristics such as gait feature changes may also be monitored using a variety of sensors. Many studies have been conducted for gait tracking to extract gait features. One study by Liu, et al., (2009) focused on the lower trunk inclination being estimated gait phase detection. The authors used acceleration data collected during the mid-stance of gait. Another study used an accelerometer for gait tracking (Takeda, et al., 2009). The authors extracted the component of gravitational acceleration from acceleration data recorded during walking to calculate joint position. Chapter 2 provides further information on additional influential research in the fields of gait analysis for residential and clinical use.

This thesis has been motivated by some global issues including the ageing population (Al-Yaman, 2004), diseases that are related to the elderly, a shortage of professional nurses, and the high financial expenditures on health care for patients and elderly care. These phenomena will be more complicated in the future (Zeitler et al., 2012, Soufian et al., 2020). However, autonomous robotic sensing systems have been adopted by researchers and clinicians to improve the quality of life for patients and the elderly, with some approaches being considered as a low cost, reliable and robust solution. Hence, this thesis proposes a low-cost, tech-based solution to automate gait features extraction and classification, that can be deployed in residential environments.

1.3 Research Scope

The use of walking speeds as a measure in gait analysis system, for elderly people in their homes is the main concern of this research. For improvement, the accuracy and reliability of the measurements recorded using a gait analysis system are supported by different automated techniques, learning algorithms and vision systems, which can provide rich information of the gait movement. This system can be used to improve the quality of life for elderly people, thus enabling them to live in their home safely and independently for as long as possible. In Timed

Walk Test (TWT), the measure of gait speed is highly recommended for evaluating the gait pattern changes. To address existing concerns, the proposed system involves multiple stages for collecting, smoothing, extracting and reducing the gait data, to be used in data classification based on walking speed.

This study focuses on enabling elderly people, who live in a residential setting, to be monitored and tracked without the use of any attached devices to their bodies or video data to respect their privacy. The MS Kinect sensor as an efficient tool can satisfy these requirements and hence improve the quality of life by reducing the cost of healthcare and helping such environments to meet some needs of elderly people. Unfortunately, the Kinect camera provides a low data rate which leads to inaccuracy in measurements. However, the use of Amplitude Modulation and Convolutional Encoder techniques can play a major role in gait analysis, by improving the quality of extracted gait features, which leads to the enhancement of data classification accuracy.

1.4 Research Question

To achieve the aim and objectives of the study, this research is primarily concerned with the development of a gait analysis technique using Kinect camera for the extraction of the most representative gait parameters that can efficiently represent changes in walk pattern. Furthermore, these extracted features will be used to improve the accuracy of data classification based on walking speeds. The research question is:

“Can the automated gait analysis technique be used to improve the effectiveness of distinguishing gait pattern changes and ranking them based on walk speed, in particular when a low-cost, low data rate sensing device is used to collect the data?”.

1.5 Research Aim and Objectives

The aim of this research is to develop an autonomous gait analysis system to detect and rank the changes in gait pattern based on walk speed, by avoiding gait cycle phases determination in case of using low data rate device. Moreover, the literature showed a widespread belief of a relation between the changes in gait parameters and walking at speeds that range from slow to fast (Fukuchi, Fukuchi & Duarte, 2019). Hence, this research attempts to enhance the performance of a gait analysis system, using a Kinect camera by improving the

accuracy of gait data classification based on walk speed. To address this aim, the following research objectives will be covered:

- To develop a low-cost and affordable system to track and collect the 3D skeleton data for the participant's legs during walk process.
- To select the efficient smoothing data approach among a set of filtering techniques for reducing the level of noise from the collected data using Kinect camera.
- To validate the effectiveness of Kinect camera compared to high data rate camera systems.
- To develop an autonomous system that can perform a timed walk test for investigating that, the changes in walk speed might be detected efficiently by increasing the quality of extracted gait data.
- To map the baseband frequency of gait length signal into passband frequency using Amplitude Modulation (AM) technique for extracting new gait features without requirement for gait cycle determination.
- To use Principle Component Analysis (PCA) technique to calculate the most representative vectors of gait features to reduce the dimensionality of data matrix.
- To improve the accuracy of gait data classification in different cases of walk speeds using Convolutional Encoder (CE) technique.

1.6 Proposed Methods

In this research, a combination of both the Amplitude Modulation AM and Convolutional Encoder CE techniques is proposed, to automatically distinguish among the changes of gait pattern based on walking speed using the skeletal data of lower limb movement. However, a smoothing and filtering process is required due to the noise level in collected data when using a Kinect camera. An appropriate filter is chosen based on the fast response and time delay. Gait cycle determination is commonly used as a first task in human gait analysis (Kharb et al., 2010). Though, the AM approach is used for the extraction of gait features without the need to determine a gait cycle. Kinematic and spatiotemporal gait features are employed to extract the major of gait features. In addition, the performance of Amplitude Modulation (AM) technique in gait feature extraction is evaluated compared to traditional methods. In the dimensionality reduction of gait features, a Principal Component Analysis

(PCA) is used to reduce the matrix dimension of extracted features, where feature vectors are selected based on eigenvalues that offer a more representative pattern. During the classification stage, several supervised classifiers are used to classify the gait pattern changes. Moreover, use of the Convolutional Encoder (CE) technique is proposed, to classify the extracted gait features according to their changes in the gait pattern, then its accuracy is compared to a set of supervised classifiers. Finally, different evaluation metrics are applied to evaluate the accuracy of the whole proposed system.

1.7 Research Contribution

The major contribution of this research is to explore autonomous gait tracking techniques that can be effectively used for distinguishing gait pattern changes based on walk speed. The specific expected contributions will include:

- Identification of the best smoothing technique among six filters, for the reduction of the noise from skeletal positional data that is collected by a Kinect camera, by using correlation coefficients and fitting curve approaches for input and output data of filters.
- Investigate the effectiveness of extracting gait features using Amplitude Modulation technique comparing to Frequency Modulation technique in case of a low-data rate of sensing device.
- Introduce a new technique called Amplitude Modulation for extracting new gait features without need to determine gait cycle.
- Building a full system of autonomous gait analysis called (AM/CE) for detecting and ranking the gait pattern changes based on gait speed.

1.8 Thesis Structure

This thesis contains six chapters. Chapter 1 has presented the main topics that launched the study by introduction and motivation. Followed by the research problem and scope, addressing the main aim and objectives. Finally, proposed methods that will contribute in solving some problems and study structures.

- **Chapter 2: Effects of Walking Speed on Gait Parameters**

Chapter 2 reviews the walking speed effects on gait parameters. Contents; a general description of elderly care in a residential environment for increasing the quality of life, where gait analysis is being adopted using sensing technologies and smart approaches.

- **Chapter 3: Literature Review**

The purpose of this chapter is to review the most relevant subjects for this research by exploring previous studies. This review also involves the use of a Kinect camera in gait analysis. The main topics include; 3D positional skeleton data collection, several techniques that were used in smoothing and filtering the skeleton data, a summary of gait cycle detection, a comprehensive guide to the extraction of gait features (i.e. kinematic and spatiotemporal gait features), reduction of gait features matrix, and gait pattern classification based on gait speed changes.

- **Chapter 4: Implemented Methods and Proposed approaches**

Chapter 4 demonstrates the methods and approaches that will be used for obtaining the results, with the proposed methods be derived mathematically in steps and explained through examples.

- **Chapter 5: Results and Discussion**

This chapter describes the aim of the experiment and the collation of results, which are illustrated in figures and tables. An analysis of the results is explained at the end of each experiment separately.

- **Chapter 6: Conclusion and Recommendations**

Chapter 6 concludes with a summary of the findings, then offers research recommendations and issues with potential future work, that could be followed up in future research projects.

2. THE EFFECTS OF WALKING SPEED ON GAIT PARAMETERS

This chapter explains the effects of walking speed on gait parameters during feature extraction. In addition, gait analysis use in certain environments are reviewed, including residential living and a clinical walk test for the elderly using a Kinect camera.

2.1 Introduction

Gait speed is a common measurement across several gait assessment tests, which is used to describe and rank walking ability. For instance, a Timed Walk Test (TWT) is a clinical approach used to assess gait performance, where the walk speed is an essential measure. Although such tests are considered as subjective assessments, the use of these approaches with supporting technological tools could assist in making decisions. Especially for the elderly who prefer to live independently. Moreover, Ambient Assisted Living (AAL) is designed to improve the quality of life in such environments. Particularly when a residential setting does not have a medical license. Consequently, the cost of healthcare can be reduced, and the quality of life could be improved. In addition, an objective assessment can be provided by introducing technologies which may aid clinicians to make decisions, rather than relying on the use of a subjective assessment. Therefore, this study proposes to automate a gait analysis system that can be used efficiently during the detection and classification of gait pattern changes using a Microsoft Kinect V2, which may contribute to the improvement of the quality of life for the elderly, who prefer to live in their own home.

2.2 Gait Parameters Changes due to Walking Speeds

Human gait analysis is an attractive field of study for many researchers and clinicians at present, especially with the use of a vision tracking system. This analysis involves measurement, compression, description, classification and assessment of the changes in gait pattern (Ghoussayni, Stevens, Durham, & Ewins, 2004). It is commonly used to detect gait phases, extract the kinematic and spatiotemporal gait parameters, and classify the gait data.

Notably, walking speed is a fundamental evaluative tool in gait assessment (Robertson, Parsons, Sidtis, Hanlon Inman, Robertson, Hall & Price, 2006). In fact, the biomechanical

variables are correlated to the changes in gait speed such as kinematics gait features, kinetics gait features, spatiotemporal gait features, muscle activity and ground reaction forces (GRF) amongst others. In this context, numerous studies have been explored, which explain the effects of walking speed on gait patterns in different fields of gait analysis. For example, (Jordan, Challis & Newell, 2007) clarified the fluctuation of gait cycle parameters (interval & length of step & stride gait) due to walking speeds. While, (Ardestani, Ferrigno, Moazen & Wimmer, 2016) investigated changes in cadence and stride length because of gait speed changes from slow to fast by using joint movement from the lower extremity. Furthermore, spatiotemporal gait features showed speed-dependency for distinguishing between healthy and unhealthy people, who live with bilateral vestibulopathy (BVP) (McCrum, Lucieer, Van De Berg, Willems, Fornos, Guinand & Meijer, 2018). It was explained that the most significant differences were at slower walking speeds with temporal and sagittal plane spatial gaits, while frontal plane spatial gait variability was demonstrated at faster walking speeds.

Clinically, a walk speed is used as a measurement for gait assessment in many approaches of Timed Walk Tests. For example, Behrens, Pfüller, Mansow-Model, Otte, Paul & Brandt (2014) assessed 22 patients with MS disease, nine of whom were males. A Kinect sensor was placed 2m in front of the patients to measure gait speed. The author was then able to determine an acceptable correlation between the gait speeds measured using a Kinect and the clinical measurements. The tracking of the lower body joint movements was conducted using skeletal data. Furthermore, Galna, Barry, Jackson, Mhiripiri, Olivier & Rochester (2014) tested 9 people with Parkinson's disease, three of whom were male. The authors measured the up/down displacement of the knees within the movement timing and spatial displacement using skeletal data.

2.2.1 Timed Get Up and Go (TUG)

TUG is a timed test commonly used to measure an elderly person's ability to turn around 180° (Podsiadlo & Richardson, 1991), where functional mobility, walking balance and the ability to stand to sit and sit to stand are also considered. In this assessment, a person is instructed to stand up, walk forward 3 meters, turn around 180°, walk back and sit on the chair (Dubois, Bihl & Bresciani, 2017) see Figure 2.1 below for more details.

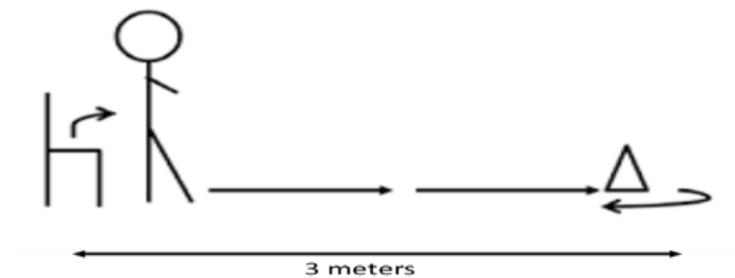


Figure 2.1: Timed-Up & Go Test

If the participant can complete this task in 11 to 20 seconds, this can be considered normal (Shumway & Woollacott, 2007). However, Shumway-Cook, Brauer & Woollacott (2012) suggest that more than 30 seconds to complete the TUG test could mean that the participant may be at a fall risk. While, Straudi, Martinuzzi, Pavarelli, Charabati, Benedetti, Foti & Basaglia (2014) used TUG for 10 subjects with MS disease to assess their mobility. Moreover, further research was carried out by Vernon, Paterson, Bower, McGinley, Miller, Pua & Clark (2015) who compared data that was measured by a Kinect to clinical test measurements. The Up and Go test (TUG) was used as a clinical timed test. Thirty participants were recruited aged between 15 to 68 years, 21 of whom were male. In this case, the Kinect showed excellent association with the TUG's clinical test.

2.2.2 Timed 25-foot walk (T25FW)

The T25-FW test is managed for walking speed assessment (Hubbard, Wetter, Sutton, Pilutti & Motl, 2016). In this test, participants are instructed to walk as fast as possible, but safely. This is done over a 25-foot carpeted surface (Fischer, Rudick, Cutter, Reingold & National MS Society Clinical Outcomes Assessment Task Force, 1999). The time taken is recorded for participants over two trials. The average completion time over two completed trials is the score for the T25-FW, and this average can be converted into walking speed (Kieseier & Pozzilli, 2012). Management of the T25-FW test covers trials 1&2 and the recording of the report form. In trial one, the subject should be instructed to walk from the starting line, and to finish the task of safely walking 25 feet as fast as possible. The departure time is recorded from the starting line till arrival at the finish line, which are considered as begin time and stop time respectively, as shown in Figure 2.2 below. In trial two, the subject must repeat the same instructions as trial one when he/she reaches the second line. To complete the record form, completion time of the two tasks together (trials 1 and 2) can only be recorded as a successfully completed task.

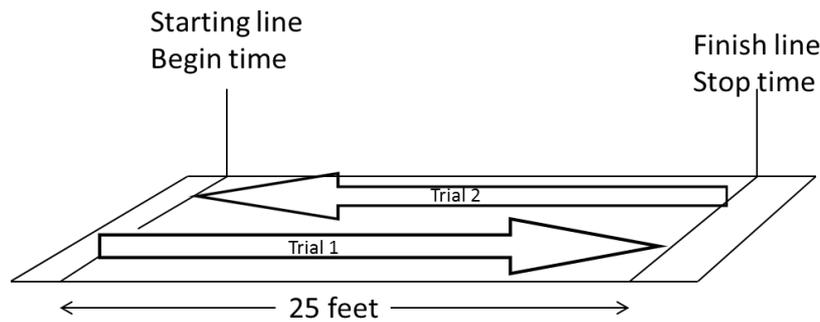


Figure 2.2: Timed 25-Foot Walking

Several methods have been introduced to determine a clinically significant change in this measurement. For instance, a range of values has been developed for the T25-FW to assess changes in walk-time. Examples include changes of gait features that occur during an exacerbation of multiple sclerosis. Ries, Echternach, Nof & Gagnon Blodgett (2009) suggest that an increase in test score may indicate a significant gait feature change. Moreover, the T25-FW has been considered as a reliable test, especially in longitudinal studies with different physicians assessing patients (Freedman, Patry, Grand'Maison, Myles, Paty & Selchen, 2004). In the same context, Clark, Vernon, Mentiplay, Miller, McGinley, Pua & Bower (2015) compared Kinect data to clinical test measurement. A 10m walking test was used as a clinical assessment. Thirty participants were recruited, nine of whom were females, to measure step length and gait speed by using one Kinect sensor to provide the skeletal data.

2.2.3 Multiple Sclerosis Walking Scale (Rating Scales)

Gait velocity is described as an important primary tool in Multiple Sclerosis (MS) clinical assessment by the National Multiple Sclerosis Society's Clinical Outcomes Measurement Task Force (Fischer et al., 1999) & Robertson, Parsons, Sidtis, Hanlon Inman, Robertson, Hall & Price (2006). One standard scale for MS is the Expanded Disability Status Scale (EDSS). The use of this scale is to classify disability levels (for further details see Figure A.1 in appendix A), which have a numerical range from 0 to 10 for patients with MS (Kurtzke, 1983). While, the Hauser Ambulation Index (AI) is another scale for patients with MS to assess ambulation-related disability (Hauser, Dawson, Lehigh, Beal, Kevy, Propper & Weiner, 1983). However, the AI scale provides outcomes in walking speed assessment for patients at a more reliable level than the EDSS (Schwid, Goodman, Mattson, Mihai, Donohoe, Petrie & McDermott, 1997).

Another clinical scale is the 12-item MS walking scale (MSWS-12) which is a self-report for individuals who have Multiple Sclerosis (Allen, Diane, Bennett, Brandfass, Pittsburg, Stratford, Widener & Flint, 2011). The MSWS-12 is highly recommended for patients with Multiple Sclerosis to assess their walk ability with five levels: 1 means no disability, while 5 means extreme disability (see Table A.1 in appendix A).

The Multiple Sclerosis Evaluation Database to Guide Effectiveness (EDGE) task force outlines 63 outcome measurements (OMs) (Potter et al., 2014), (see Table A.2 in appendix A). These OMs cover the list of recommended tests for patients with MS including T25-FW, TUG, and MSWS-12. These tests and scales use the Timed Walk Test and the ‘ability of walk’ as a primary tool for achieving the assessment. For this purpose, the use of a marker-less based vision system in walk assessment is adopted for this research. Particularly for use in a home, or even a clinical setting, as this may enable the automation of gait feature extraction, which is required for improvement in measurement accuracy.

2.3 Residential Care for Elderly People

Assisted living facilities and nursing homes provide services such as personal care and medical amenities. However, residential homes (or assisted living facilities) are not licensed to give nursing care. Typically, an assisted living facility is a place where elderly people live, and where they receive help in the activities of daily living (Helal & Abdulrazak, 2006).

Assisted living facilities do not include licensed nurses nor do they have any connection with nurses. These settings are considered as non-medical services. Even if nurses are available amongst the staff of an assisted living facility, the nurse cannot carry out the duties of nursing, such as administering oxygen, insulin, or other clinical jobs. Assisted living facilities are organized by the state Department of Social Services (Zimmerman, Scott, Park, Hall, Wetherby, Gruber-Baldini & Morgan, 2003), (Zimmerman & Sloane, 2007). Whilst, nursing homes are organized by the Department of Health. However, the population of these settings continues to grow and includes residents with dementia diseases, which can have a demonstrable impact on assisted living facilities. A study conducted by the National Academy on an Aging Society (2000), reports that almost 4 million USA residents aged 65 years or over, have Alzheimer’s disease. Furthermore, it states that this number is predicted to triple by 2050. Additionally, another study reported that the amount of people who live in a residential setting with mental diseases such as dementia had reached at least half of the total population (United States General Accounting Office, 1997).

In other words, it is possible to surmise that some assisted living communities have become places for people who live with dementia or Alzheimer's disease. According to a study that conducted a visit to 22 random assisted living places, research showed that two out of the three residents living there had Alzheimer's disease (Rosenblatt, Samus, Steele, Baker, Harper, Brandt & Lyketsos, 2004). In the same context, another study recorded that out of every three residents, one had a cognitive impairment, ranked between moderate to severe (Hawes, Phillips, Rose, Holan & Sherman, 2003). Recently, some studies have also reported that diseases such as MS and Parkinson's, which are associated with ageing, may decrease an older person's independence of their daily living needs. Moreover, the ageing population is increasingly becoming a larger part of the population. Subsequently, home care approaches will not be sustainable. However, sensing technologies have been introduced to develop "Assistive Environments" that aid the elderly and enhance their quality of life, with safe and independent environments.

Consequently, there is a need to address the problem from both a societal and economic standpoint. As support can be driven into either increasing the quality of life or providing high quality medical support. For instance, Fried, Cwikel, Ring & Galinsky (1990) designed the "Extra-Laboratory Gait Assessment Method" or ELGAM, which was designed to assess gait in the home or in an outpatient setting, with the gait speed measured by stop-watch. However, these clinical tests are considered as semi-subjective assessments, as they are carried out by specialists who assess a patient's gait through observation of their walking.

One such support system is Ambient Assisted Living (AAL), which has been adopted to play a major role in the assistance of elderly people within a low-cost environment with continuous social communication. Ambient Assisted Living is an area where sensors can be involved to create an intelligent environment for ageing or cognitively impaired patients, thus enabling them to stay independent, safe and active for longer in their preferred environment (Monekosso, Florez-Revuelta & Remagnino, 2015). A walking analysis is explored in such environments specifically to detect falling (Lombardi, Ferri, Rescio, Grassi & Malcovati, 2009). This can contribute to an increase in the quality of life and a cost reduction for public health systems (Kleinberger, Becker, Ras, Holzinger & Müller, 2007). Research in the AAL community has covered a large range of studies. Notably, most research has been conducted in the area of human activity recognition and behaviour comprehension, with the objective of detecting activities within an environment. Furthermore, the recognition and detection of events is an important topic in AAL solutions. One example is fall detection, where wearable

sensors have been widely used for detecting falls, but one limitation is that they must always be worn. More recently, researchers have included optical sensors in assisted living environments. However, the challenges of the optical sensor include clutter, obstruction and other noises. Moreover, statistics show that the largest number of falls has been recorded in the bathroom, a location where privacy concerns are highest. To tackle this issue, Infra-Red IR sensors are used for privacy-aware techniques.

2.4 Sensor Technologies

Gait speed assessment has been experienced widely using different kinds of sensors. Recently, vision-based tracking with marker-less systems has been involved in research that aims to track human movement data in real-time. Whilst, different sensor technologies have been used in the tracking of human movement. Examples include inertial, marker and RGB/IR sensors.

2.4.1 Non-vision-based Tracking System

In non-visual based systems, sensors are attached to parts of the body to collect data on position and velocity (Zhou & Hu, 2008). These sensors can be classified as inertial, acoustic, magnetic sensing, mechanical and RF sensing. The advantages and limitations of these sensors are dependent on the sensor type. Limitations include frequent battery operations and replacements, along with modality-specific, measurement-specific and circumstance-specific issues (Zhou & Hu, 2004). For example, as part of inertial sensors, accelerometer sensors (also known as inside-out systems) are employed on an object to sense an external source (e.g. the earth's gravitational field as reference) to provide information in 3D. However, the limitation of this system is its dependence on an external source. In addition, an accelerometer suffers from 'drift problems' during measurement. For example, when the accelerometer estimates location or velocity, a drift might take place due to sensor noise or offsets. Consequently, additional correction is required throughout the tracking process (Bouten, Koekkoek, Verduin, Kodde & Janssen, 1997). Furthermore, resolution and signal bandwidth are normally limited by the interface circuitry (Bouten et al., 1997).

2.4.2 Vision Based Tracking System

The vision system-based tracking system can be categorised into two parts: vision-based tracking, with and without markers. In the former part, optical sensors (cameras) are used

to track and monitor human movements, which are captured by placing markers (identifiers) on the parts of the human body involved. Examples include a Vicon camera. As skeletal human movement is highly articulated, rotations and twists require full 3D movement tracking (Bray, 2001). This system has been used successfully in biomedical fields (Delahunt, Monaghan & Caulfield, 2007), where it enables the participant to move continuously in and out of the camera view. Consequently, this leads to consistent and reliable tracking of the human body. However, one major limitation of vision sensors with markers is that they cannot be used outside laboratory environments. This leads to difficulty in preparation and installation; therefore, such systems are unsuitable for unstructured clinical and residential settings. In addition, this kind of system suffers from obstruction and ghost problems due to interference (Sullivan, Eriksson, Carlsson & Liebowitz, 2002).

In the second category, which is vision-based tracking with marker-less systems, computer vision algorithms are employed with an inexpensive camera to estimate the position data of human gait (Andersson & Araujo, 2015). Use of these sensing technologies adds another dimension to traditional RGB cameras, which is an RGB-Depth device, such as the Microsoft Kinect, thus providing a robust solution to infer 3D scene information regarding human gait analysis through continuous-projection onto a screen. In later sections, more details will be provided on the principal work of marker-less vision-based tracking systems.

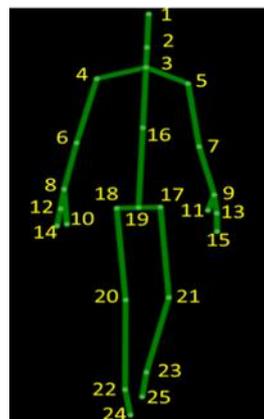
2.4.3 Microsoft Kinect Sensor Based Depth Data

Basically, Microsoft Kinect is a set of devices that work together to make it a powerful product (Lachat, Macher, Mittet, Landes & Grussenmeyer, 2015). The Kinect contains a colour camera, depth sensor (IR camera & IR projector) and four microphones, as can be seen in Figure 2.3 below. The Kinect can be used for motion sensing and tracking, as well as capturing and interpreting full-body movements. Consequently, researchers quickly realized that the Kinect could be used for purposes other than games (Zhang, 2012).

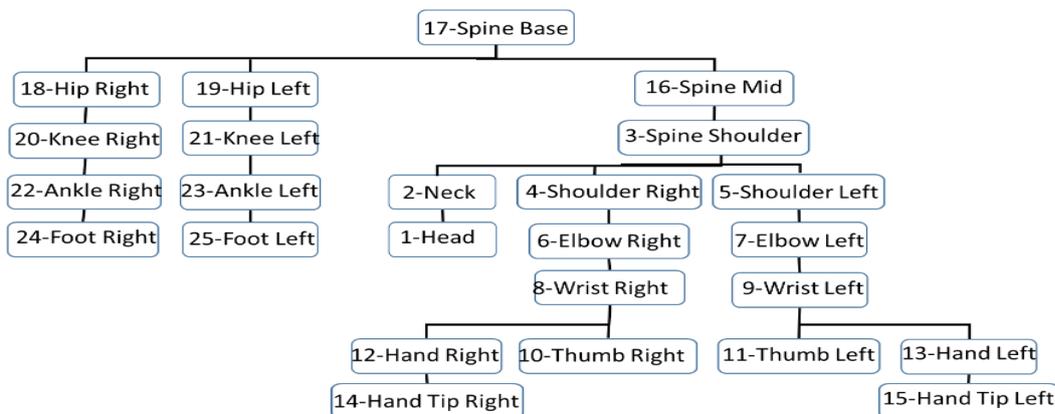


Figure 2.3: The Main Components of Microsoft Kinect

Due to the Kinect's depth sensor and low cost, it can be used in different fields. One example is the detection and tracking of skeleton joints. The depth camera sensors make it possible to obtain the depth data of an object's parts in a scene. The information is provided by the RGB-D camera, where each pixel has information on the estimated distance from the RGB-D camera to a certain point on the object. The pixels' location can be formed in frames (30 frames per second). The position information from the RGB-D sensor allows the tracking of several human body joints (25 joints with Kinect V2) in 3D as shown in Figure 2.4 below. The skeleton joint positions can be acquired by the MS Kinect using the MS Kinect SDK, which is a middleware framework. In recent years, depth cameras have commonly been used as sensors to capture depth images in real time (frame rates). In this section, some of the studies that have involved the capturing of depth information using a Microsoft Kinect in their applications will be reviewed.



(a)



(b)

Figure 2.4: Human Body Joints: (a) Skeleton of Human Body (b) Hierarchal Human Body Joints

One example of application is provided by Aitpayev & Gaber (2012) who used MS Kinect for collision objects (human body parts) in augmented reality. The author developed the new Kinect SDK to improve the accuracy of skeleton motion recognition. In the same year, Tong, Xu & Yan (2012), proposed skeleton animation motion data based on the Kinect. The joint position data was captured first, followed by a joint rotation calculation to enhance the 3D model. The author designed a low-cost system for 3D motion, but jittering was observed in the implementation process. Furthermore, Zeng, Liu, Meng, Bai & Jia (2012) presented a technique of motion capture and reconstruction using depth information from the Kinect. This method was based on a model of the human body reconstructed using 3D motion. The authors recorded high accuracy of human motion and posture with low latency in the system process. Although noise and errors are usually generated during the process of data acquisition, this issue is solvable. For instance, Ma, Xu & Liu (2011) collected 3D positions of human joints in real time using the MS Kinect, and then rotation matrices were computed for those joints. The authors obtained a target skeleton for animation of 3D characters. Significantly, the noise and errors in joint position were removed.

Notably, using multiple Kinects instead of a single camera increases performance and reliability. For example, Tong, Xu & Yan (2012) designed an algorithm for scanning a 3D human body using low cost devices such as the Kinect. The authors were able to deal with the interference phenomenon and obstructions using multiple Kinects. However, the reconstruction model quality was still poor, due to the low accuracy of the depth data captured. Specifically, skeleton tracking techniques were studied using different technologies that were either marker based or without marker data vision. For instance, the following studies adopted sensing technology for skeleton tracking using both methods (markers/marker-less). One study was carried out by Fern'ndez-Baena & Lligadas (2012) who conducted a comparison of the MS Kinect with another visual motion capture device (Vicon) for the validation of human body joint movements (upper/lower) for rehabilitation treatments. In this case, it was demonstrated that the accuracy of the Kinect was less than that of the Vicon. However, the Kinect is still beneficial as it is marker less, portable and has a low-cost.

2.4.4 Gait Analysis using Microsoft Kinect Sensor

The use of a Kinect camera as a gait analysis instrument has emerged in the last few years as an attractive tool in marker-less vision-based tracking systems. Moreover, recent

studies have suggested that the Microsoft Kinect can be exploited to estimate spatiotemporal gait parameters (Clark, Bower, Mentiplay, Paterson & Pua, 2013), and gait kinematics features (Gabel, Gilad-Bachrach, Renshaw & Schuster, 2012).

Gait analysis has been explored widely by researchers and clinicians, with the focus on gait feature extraction, which can explain gait pattern changes. One study by Clark et al., (2013) used skeletal data to assess the gait step time, gait speed, gait step length, gait stride time, gait stride length and velocity of foot swing for twenty-one participants. More accuracy for stride length, step length and gait speed were observed than for the other parameters. Another study by Mentiplay, Perraton, Bower, Pua, McGaw, Heywood & Clark (2015) used 3D skeleton data for thirty participants. They placed the Kinect in front of the subjects, to test gait velocity, speed variability, length/width of step gait, swing leg duration, and displacement of the pelvis. The flexion for ankle/hip, and flexion/adduction for the ankle were also measured. Furthermore, Xu, McGorry, Chou, Lin & Chang (2015) recruited twenty participants, half were male, to assess gait cycle parameters such as stride time, step time, swing time, stance time and double limb support time. The joint angles of the hip and knee during a gait cycle were also measured. Notably, the measurements of step time, stride time, and step width were more accurate when compared to the measurements of the Kinematic parameters. In addition, Auvinet, Multon & Meunier (2015) computed the largest distance between the knees to estimate heel strike events. Eleven participants were recruited using depth data provided by the Kinect, which was placed 2m in front of the subjects.

Many studies have attempted to exploit the MS Kinect in gait analysis, particularly in clinical tests. For example, Pfister, West, Bronner & Noah (2014) positioned a Kinect sensor on the left of the participants at 45° to the treadmill. This was carried out on twenty participants, to measure the peak angular displacement for hip and flexion / extension of knee and stride time, using skeletal data. The measurements of the Kinect for the knee were more accurate than for the hip. Although this study recorded outcomes that were insufficiently in agreement with the clinical tests, recent research has recorded acceptable results in comparison with clinical timed walking tests.

2.5 Summary

The ageing population is increasing more than ever before, and it is expected to rise in the future, therefore the demand for elderly housing care will be higher. One significant

challenge is whether existing housing facilities will still meet the changing needs of the residents. Moreover, studies have reported that many of the residents who live in residential care suffer from dementia diseases. However, support can be reached by increasing the cost of healthcare or by improving the quality of life. In other words, there is a trade-off between acute care cost and the quality of life. The latter can be adopted and improved by enabling the elderly to live in their own homes independently and safely for as long as possible, this could dramatically decrease the cost of healthcare.

AAL is defined as a solution for improved quality of life, it involves technologies, smart systems and automated approaches being incorporated into assisted living environments. In this context, the use of a gait analysis system is highly recommended as a tool for walk assessment, especially based on the evidence that supports a correlation between walking speeds and changes in gait parameters. This means that a gait analysis system can contribute to the improvement of the elderly's quality of life. In addition, an MS Kinect camera can be an efficient tool in assisted living environments, as it provides various benefits such as being marker-less, low-cost, quick to install even in an unequipped environment and can be used as a privacy aware system.

3. LITERATURE REVIEW

This chapter reviews one of the most state-of-the-art devices available for exploration into gait analysis – a Kinect camera, which focuses on skeletal data for gait spatiotemporal analysis. In addition, it identifies and describes the approaches and algorithms used for the extraction and classification of gait features.

3.1 Introduction

A vision-based tracking system without markers plays a crucial role in gait analysis. This low-cost system provides vital and rich information. Moreover, the tracking of system-based 3D skeletal data can be achieved without the attachment of any kind of devices to the body, and it is quick to install even in a non-equipped place. Hence, its overall contribution is a reduction in the cost and complicity of the measurements and analysis. The processing of skeletal data for gait analysis is applied in several stages to build a complete system that can collect, smooth, extract and classify gait features. The purpose of this system is to distinguish among the gait pattern changes due to walking speeds.

Gait feature extraction plays a major role in the next stages of the system, as the quality of the extracted features can affect performance at the data classification stage. Several techniques are conducted for gait feature extraction, with the challenge being how to address the extracted features that can enhance classification accuracy. For this purpose, the extraction approaches can be defined as efficient, robust and reliable if they can improve the performance of the classification stage. In addition, the dimension of the gait features matrix can be reduced to improve gait feature quality. Using the data reduction technique, the vector features that have a high percentage of variance will be selected. For example, a Principal Component Analysis (PCA) is commonly used in feature reduction solutions, based on eigenvalues and eigenvectors, to determine the high representative feature vectors. During the classification stage, several techniques are employed in gait pattern classification. To test the unseen data, a Cross-Validation (CV) approach is commonly used with supervised classifiers, where a k-fold CV is highly recommended as it guarantees that each point of data can be used as a training and testing set without overlapping.

This chapter is organised into several sections to describe the main stages of the gait analysis system. First, skeletal data collection is presented in section 3.2. Then, section 3.3 provides a brief overview of the pre-processing techniques for smoothing and filtering noisy data from skeletal data. After that, section 3.4 gives an overview of human gait cycle determination including the main sub-phases that form one complete gait cycle. While, gait feature extraction and reduction are shown in sections 3.5 and 3.6 respectively. Then, data classification of the gait pattern changes is detailed in section 3.7, and section 3.8 details the most suitable evaluation metrics for classifier performance, while cross validation techniques are provided in section 3.9. Finally, this chapter is summarised in section 3.10.

3.2 Skeletal Data Collection

MS Kinect provides several data types that can be employed in a gait analysis system by tracking human movement in a 3D space. The RGB-D information and 3D skeleton data are broad categorizations of the Kinect's sensor data (Han, Reily, Hoff, & Zhang, 2017). In this section, 3D skeleton data will mainly be reviewed, because the proposed system is built to investigate changes in walk speed that can affect gait parameters using this data. Specifically, for spatiotemporal and kinematics gait parameters.

Significantly, 3D skeleton data can be collected using a Kinect camera, to extract gait features for classification of gait pattern changes based on walking speeds. For example, one study recruited twenty healthy participants to walk on the treadmill, while both systems (Kinect sensor and motion capture system) tracked the kinematics data of body joints. The aim was to extract knee and hip joint angles, where the frame error rate of the heel strike estimation was 0.18 and 0.3 for the right and left leg, respectively. Whilst, the average toe off frame errors estimation was 2.25 and 2.61 across three different walking speeds of 0.85, 1.07, and 1.30 m/s (Xu, McGorry, Chou, Lin & Chang, 2015). In a similar context, 21 healthy subjects were instructed to walk at maximum speed as much as possible and safe, then the spatiotemporal gait parameters were extracted from collected data using both a Kinect and gold standard device. The results showed that there was a high correlation of ($ICC \geq 0.888$) in the case of one Kinect, however the correlation rate improved when multi-Kinects were used (Geerse, Coolen & Roerdink, 2015).

Furthermore, an accuracy evaluation of 3D Kinect data in spatiotemporal gait analysis was investigated by (Dolatabadi, Taati & Mihailidis, 2016), where the GAITRite system was included as a comparable tool and gold standard, under three walking conditions, including the

usual, dual task and fast walk. The results of the agreement showed at 95% Bland-Altman limits for the Kinect as a valid tool in spatiotemporal gait analysis during three conditions. An excellent correlation ($ICC_{2,1} = 0.98$), and strong reliability were exhibited among the walking types ($ICC_{3,1} > 0.73$).

3.3 Smoothing and Filtering Data Techniques

The skeletal positional data, that is tracked and collected by a Kinect camera during a human walk, is known as a dynamic problem. The captured information is always described as noisy data due to numerous conditions including obstruction by other objects; people or furniture, other body parts known as self-occlusion, and the movement of a joint outside the captured area. In most of these cases, the marker-less skeleton tracking system is still able to track the position of the joints' movements. However, even though the corrupted data might be improved with higher quality sensors, it is not possible to eliminate all noise. Hence, in practice, an error rate reduction or a smoothing data approach is often desirable. In this section, a list of the different filter techniques, that are commonly used for smoothing skeleton data and reducing noise and errors from the original data, will be included.

3.3.1 Moving Average Filter

This filter is the most popular smoothing data technique, it is a simple low pass structure, that acts to meet the engineers' requirements for solving unwanted components of collected data. This technique uses a certain sized window width that is shifted instantaneously over the data from one instant point to the next. The average value of points in that window is instead of each data point. In addition, this technique can be applied as multi-filters to run all at once and increase the smoothness rate. For instance, Cai, Wu, Xiang, Zhong, He, Shi & Xu (2012) used a cascaded average moving filter to remove the baseline wander from knee joint vibration signals. The desired signal was sampled at 2 kHz, and then it was converted into digital form. The experiment demonstrated an efficient smoothness of data at the output of the cascaded moving average filter when compared to the original data. Whilst, Meanwhile, Putz-Leszczynska & Granacki (2014) showed a smoothed signal of the right foot compared to the original signal, using a moving average filter that was tuned to a window length equal to 5 with all weights equal. In the context of noise and latency, a moving average filter suffers from latency at its output due to the filtering data process. One case study by Casiez & Vogel (2012) compared a moving average filter and a proposed algorithm called "one-euro filter" for filtering

efficiency with less delay. The result showed that the effectiveness in lag reduction was 25% more than with the proposed filter.

3.3.2 Exponential Moving Average Filters

An EMA filter is a kind of moving average approach that acts based on the exponential weighted moving average of the most current data elements. However, the equal weighted moving average is applied by a simple moving average filter. Hence, this filter is known as an exponentially weighted moving filter. In other words, an exponential window is designed to exponentially decrease weights over time. Whereas, the weight values in a simple moving average filter are designed to be equal. There are many studies that have used this technique for smoothing 3D skeleton data, collected using a Kinect camera. One study by Adjeisah, Yang & Li (2015) applied an exponential filter to smooth skeleton joint position data. This research produced a graph with the tracking results being smoothed with less latency. However, the response of a smoothed curve still had slow tracking especially when the curve was turned up.

3.3.3 Median Filter

This filter is also known as a moving Median filter; where the output is calculated from the median of the points inside the window, with the window size equal to $2N + 1$ (Qiu, 1994). A Median filter is commonly used for the elimination of running spike noises (Harres, 2013). In the context of a latency problem, it suffers from a time delay regarding its response to input data, where the lag is proportional to the length of N . However, the filter effectiveness in eliminating spike noise depends on the length of N , which should exceed the spike noise peaks (Liu & Shibata, 2008). In other words, the latency of the Median filter is correlated directly to the size of N . Another limitation was reported by Feuerstein, Parker & Boutelle (2009), who established that the Median filter has slow computation due to the process of ascending or descending order. While, for image processing, Verma, Singh & Thoke (2015) examined the effectiveness of maintaining edge, and a signal to noise ratio, among a set of filters, with the results confirming an efficient performance with a Median filter in SNR improvement but less performance in preserving edge.

3.3.4 Savitzky–Golay Filter

This filter is also called a smoothing polynomial filter or a least square smoothing filter (Duong & Choi, 2013). It fits a polynomial to neighbour data point j for each input x_j in a

least-squares sense and uses the value of the polynomial at time n as the filter output. A polynomial of order K is defined to be less than the window size (Isnanto, 2011). Several studies have applied this technique for filtering the noisy data of body joints that have been collected from different sensors. That is to say, the GS filter works within the window size similarly to the function of a moving average filter. In addition, the high order of the polynomial set is involved for the least squares fitting (Romo-Cárdenas, Avilés-Rodríguez, Sánchez-López, Cosío-León, Luque, Gómez-Gutiérrez & Navarro-Cota, 2018). For each data point j , the least squares fit a polynomial within the size of the window $(2L+1)$ points.

The effectiveness of the SG filter in data smoothness for different collected data was studied by Ojaniemi (2016) who used a SG filter for processing noisy data collected from both an inertial and visible sensor. The research compared knee angle data that had been computed from different sensors, and the results showed the filtered data in almost similar shapes. Meanwhile, in the context of data classification, Ľupa, Procházka, Vyšata, Schätz, Mareš, Vališ & Mařík (2015) employed an SG filter for smoothing joint position data to detect the gait pattern changes for patients with PD, with the results revealing an achievement accuracy of 97.2%.

In addition, the efficiency of an SG filter in the improvement of action recognition, was demonstrated by Mendez et al., (2017) who proved that the accuracy of action recognition was enhanced by more than 15% when filtering approaches such as the Savitzky-Galoy and Kalman were included with feature reduction techniques as part of their proposed system. The effectiveness of the SG filter in the smoothing performance, while maintaining the shape of an original signal, was examined through a comparison with the other filters; including the moving average and local regression. Notably, the results showed that the SG filter was the best, especially when the polynomial degree was nine (Bassey, Whalley & Sallis, 2014).

3.3.5 Local Regression Filter

This method is based on linear least squares for fitting curves to the original data, it is also known as a locally weighted scatter plot smoothing technique. A local regression filter can be divided into two kinds: (LOWESS) which is a linear polynomial, whilst (LOESS) is a quadratic polynomial. In addition, this technique can be active and resistant in fitting to the outlier's values due to the robust LOWESS and LOESS techniques (Bassey, Whalley & Sallis, 2014).

In this smoothing technique, the smoothed point is processed within a certain span known as neighbour data (Cleveland & Grosse, 1996). For this, it is also called weighted due to the regression surface that is determined for the data points that represent the span (Nurunnabi, West, & Belton, 2013). Consequently, a low degree polynomial is commonly chosen for fitting data points locally in the neighbourhood. Thus, the weight is non-zero when the data point is inside the local neighbourhood, while it is zero when the data point is far from the current data (Garimella, 2017). In addition, a robust local regression approach can be used for solving the outlier's data (Helmreich, 2016). A robust weight local regression (RLOWESS and RLOESS procedures) is assigned a low weight value to resist outlier data problems.

3.3.6 Kalman Filter

This technique has a different principle rule, which is the most popular approach for optimal estimation algorithms. The Kalman filter follows two procedures to perform the estimation process; the first is a prediction process, where the current state variables produced within the process noises are considered (uncertainties). The second is called a correlation process, where the weighted average is used to update the estimation, when measurement noise has been observed.

For the state estimation model in a dynamic system, the Kalman filter uses measured data to estimate the states and covariance matrix over time progress. Notably, many studies have tried to use the Kalman filter to reduce the level of error from collected data using a dynamic system. For example, one study by Loumponias, Vretos, Daras & Tsaklidis (2016) documented the ability of a Kalman filter to obviously reduce noise from skeletal tracking signals for hand positions in vertical displacement. In addition, a Tobit Kalman filter was used, which showed as smoother in the comparison result. Also, Berti, Salmerón & Benimeli (2012) exploited the Kalman filter for tracking robotic arms with Kinect sensors, with the figures' results illustrating de-noised data and in-depth coordinates. Furthermore, one study employed a Kalman filter for the data fusion of joint positions, that were collected using a multi-Kinect approach (Moon, Park, Ko & Suh, 2016).

Generally, smoothing data approaches are often required to increase the signal to noise ratio. Particularly for 3D skeleton data that is collected from a Kinect camera, as it suffers from inaccurate capture data due to the low data rate, compared to other motion capture systems. However, there are several filtering data methods that can reduce the error rate in collected

data. For this, differentiation among the smoothing algorithms can be based on a set of requirements, including low latency and a fast response between the input and output data of a filter. In addition, the smoothing process could guarantee the shape of the original data because the most vital information is located in the amplitude of the signal.

3.4 Gait Cycle Determination

A walking pattern is a gait, which is a repeated event with several consecutive phases (Papageorgiou, Chalvatzaki, Tzafestas & Maragos, 2014). A gait cycle can be defined as the distance between two consecutive strike heels of the same leg. One cycle involves two main phases; a stance phase that is limited to between the strike heel and toe-off. In addition to a swing phase, which is started from a toe-off to the next strike heel of the same leg.

The determination of a gait cycle is the first task for most researchers interested in human gait analysis (Ahmed, Polash Paul & Gavrilova, 2015), human gait recognition, human gait modelling, and gait image representation (Wang, Kurillo, Ofli & Bajcsy, 2015). Where, the most important features, with a strong relevance to gait pattern, can be extracted within a complete gait cycle, during the human walk process (Jiang, Wang, Zhang & Sun, 2014). Therefore, extracted gait features from one gait cycle duration can define the walk assessment. One complete gait cycle is also called a gait stride, which is composed of a set number of sub-phases including; Initial Contact (IC), Loading Response (LR), Mid-Stance (MS), Terminal Stance (TS), Pre-Swing (PSW), Initial Swing (ISW), Mid Swing (MSW) and Terminal Swing (TSW), as can be seen in Figure 3.1 below.

Various studies have focused on gait cycles such as the one carried out by Ahmed, Polash, Paul & Gavrilova (2015) who detected a gait cycle by calculating the changes of distance between the left and right feet during a walk trial, where participants were instructed to walk in front of a Kinect camera. The results showed the state of the feet when they were farthest apart (maximum horizontal distance), when the IC sub-phase happened, while the minimum horizontal distance corresponded to the feet when they were closest to each other. Another study that used a similar approach to determine a complete gait cycle was by Dikovski, Madjarov & Gjorgjevikj (2014) who applied a Euclidean distance equation to calculate the distances between the skeleton joints in 3D. For this purpose, the gait cycle was defined as a period between three consecutive local minima, which ensured the period between the two strike heels of the same foot. This study started with gait cycle detection, which is considered as a repetitive action, and then some features were extracted within a gait cycle period

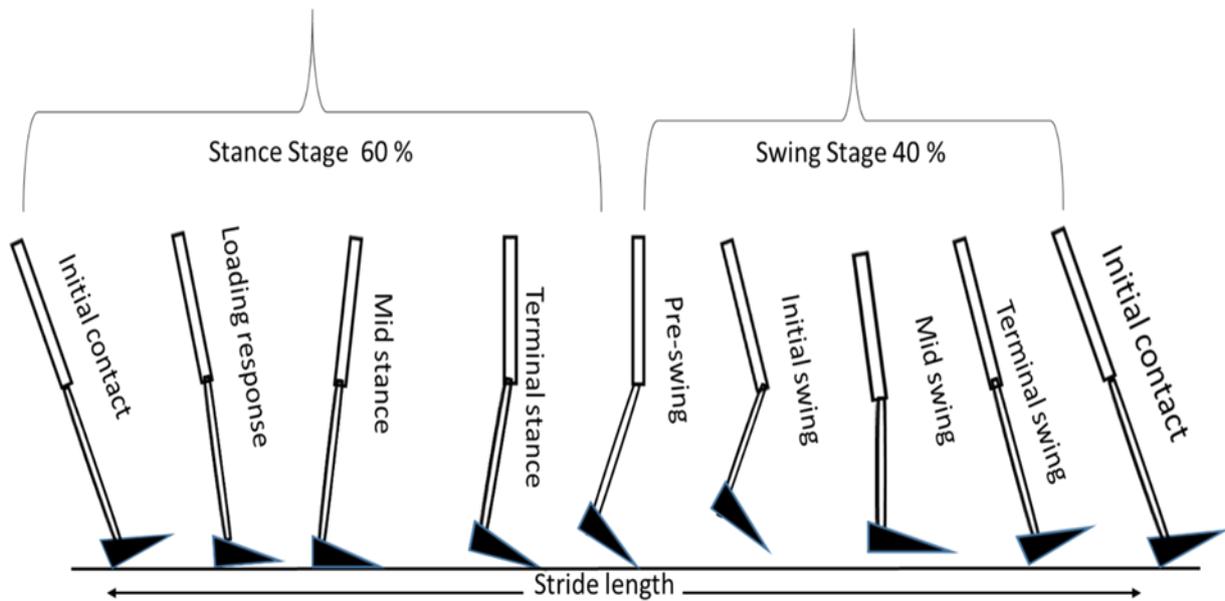


Figure 3.1: One Complete Gait Cycle that is Determined between 2 Strike Heels for the Same Foot, and Generally a Composite of 2 Phases; the Stance Phase and Swing Phase 60/ 40% of a Gait Cycle Period, Respectively.

However, other researchers have used different approaches, for instance Wang (2015) used a zero-velocity crossing approach using signal analysis on a frequency spectrum to detect the segmentation of a human gait cycle. A gait cycle can be detected from the vertical displacement of the spine-base (in y-axis) during a walk (Kale, 2015), where the maximum points represent the single-support and the minimum points represent double-support of gait events, as illustrated in Figure 3.2.

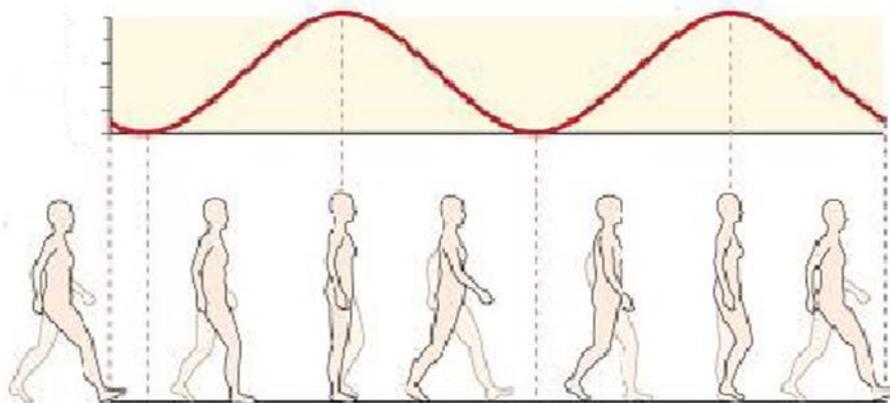


Figure 3.2: The Vertical Displacement of the Spine-Base during a Walk Process. [Source: Kinesiology Scientific Basis of Human Motion, 12th Edition by Hamilton, Luttgens & Weimar (1991)]

3.5 Gait Features Extraction

The feature extraction process can be described as “*the extraction of significant features from a background of irrelevant detail*” (Selfridge, 1955). It can be considered as a powerful

case for dimensionality reduction (Kumar & Bhatia, 2014). In gait analysis systems, feature extraction is involved as one of the most important stages, where the dataset is transformed into a set of features to describe the interest values and create a desired pattern. Usually, not all features are valuable for showing a significant change in the general pattern. For this reason, a principal component analysis is chosen to calculate the eigenvalues and select the feature vectors that can be highly representative in data classification (Guyon & Elisseeff, 2003).

A Kinect camera is exploited in a gait analysis system by collecting joint movement data to extract the feature interests and understand the changes in general gait pattern. Notably, recent studies have suggested that Microsoft Kinect can be used to estimate spatiotemporal gait parameters (Clark et al., 2013), and gait kinematics (Gabel, Gilad-Bachrach, Renshaw & Schuster, 2012).

3.5.1 Spatiotemporal Gait Analysis

Spatiotemporal gait features explore gait speed, rhythm, stride length, step length, and step width (Bonnyaud, Jansen, Salvia, Bouzahouene, Omelina, Moiseev & Jan, 2015), as well as single and double limb support time (Kim & Son, 2014), and gait cycle time (Clark, Bower, Mentiplay, Paterson & Pua, 2013). These features can be addressed from lower body limb data during the walk process, where both the stance and swing stages are detected.

In the context of spatiotemporal data, a study by (Auvinet, Multon & Meunier, 2015) calculated some gait features using a spatiotemporal gait analysis, that included gait step length, stride length and speed by dividing the gait stride length over the number of Kinect frames. These attributes were applied to three different classifiers to investigate the biometric recognition. Significantly, the results' accuracy increased when spatiotemporal gait information was combined with static attributes, obtained from a Euclidean distance. Furthermore, Babak & Sallis (2014) determined both gait cycle stages (stance and swing) automatically by tracking the ankle joint displacement over the z-axis. Consequently, some spatiotemporal gait attributes were determined, including the step length, stance phase time, and gait rhythm within a full gait cycle. Additionally, a Kinect camera can be used as multiple cameras to increase the area of view, which provides richer information. For instance, Geerse & Roerdink (2015) used multiple Kinect cameras to increase measured volume data. As a result, more accurate assessment was achieved in the measurements of spatiotemporal gait parameters compared to other studies that only used one sensor. For the purpose of this study

four sensors are used, one of which is positioned 4m from the interesting scene, while the others are within 2.5m.

3.5.2 Kinematic Gait Parameters

Kinematics is the branch of study for describing the motion of objects. This branch of study is based on describing the quantities of the position, velocity, and acceleration of several parts of the object or human skeleton (Teodorescu, 2007). In addition, these quantities can be extracted from the angle data of a body's joints, which is called angular kinematic parameters.

For example, Pfister, West, Bronner & Noah (2014) used skeleton data for measuring the angular kinematics of hip and knee flexion and extension. Similarly, Nguyen, Huynh & Meunier (2016) employed skeleton data to extract several joint angles; namely left/ right hip angles, left/ right knee angles, left/ right ankle angles, and two feet angles, these features were then used to distinguish between normal and abnormal gait. While, Jiang, Wang, Zhang & Sun (2014) calculated four joint angles (left/ right hips and left/ right knees) as a dynamic feature. In addition, the static features were measured from some lengths and heights of body parts (length of right thigh, length of right calf, length of right arm, body height, and ratio of thigh length to body height). These features were used for gait recognition based on Kinect skeleton data, with the correct classification rate (CCR) as 82% after both kinds of features were applied together. Meanwhile, another study conducted a comparison between using a Kinect camera and a Vicon system, where the maximum deviation for the angles of the elbow, shoulder, and hip in both systems were measured. The results were 12.6, 10, and 14.2, respectively. However, the minimum deviation between the two systems for the elbow, shoulder and hip were 7.1, 8, and 2.7, respectively (Phommahavong, Haas, Krüger-Ziolek, Möller & Kretschmer, 2015).

Additionally, Mentiplay, Perraton, Bower, Pua, McGaw, Heywood & Clark (2015) used skeletal data to test the walk velocity, variability of speed, length/width of gait step, swing duration, and the displacement of the pelvis. Also, the flexion of the ankle/hip, and the flexion/adduction of the ankle were recorded. Another study was carried out by Xu, McGorry, Chou, Lin & Chang (2015) to assess gait cycle parameters such as stride, step, swing, stance and double limb support time. The joint angles of the hip and knee during a gait cycle were also measured. Notably, the measurements of step time, stride time, and step width were more accurate compared to the measurements of kinematic parameters. Moreover, Auvinet, Multon & Meunier (2015) computed the largest distance between the knees to estimate heel strike events, utilizing depth data collected by a Kinect camera. However, the feature extraction stage

is very important because the performance of the next stages depend on the quality of the data that is extracted.

3.6 Gait Features Reduction

Exploration into gait feature extraction usually leads to the creation of a large matrix dimensionality. Therefore, a reduction technique is required for a dimensional decrease of the feature matrix to minimise the processing cost of redundancy data. In the context of data reduction, both linear and nonlinear techniques are explored to decrease data dimensionality.

In supervised learning, the jeopardy of an overfitting problem can occur frequently, when the number of vectors (features) have exceeded the number of samples in a dataset matrix (Kung, 2014). Therefore, feature reduction techniques play a major role in the improvement of data classification accuracy. As they reduce the number of feature vectors by considering only the most representative ones, which can increase the performance of a learning algorithm (Law, 2006). In more detail, not all features can positively contribute to the training phase, because part of them may just be 'noise' and this may degrade the learning process. Consequently, data dimensionality becomes an additional cost, due to processing, computation and complexity during data analysis (Raudys & Jain, 1991). However, the main disadvantage of a data reduction technique is that some relevant information might be discarded during the feature reduction process (Law, 2006).

A data reduction solution includes the use of linear and nonlinear data reduction techniques. One example of the former technique is a Principle Component Analysis (PCA), which can rotate the variable data from its original axis into a principle component axis by calculating the maximum variation among data vectors (Maaten, Postma & Herik, 2009), (Campbell & Atchley, 1981).

3.6.1 Principal Component Analysis

A Principal Component Analysis (PCA) is commonly used in data reduction to decrease the number of feature vectors. This technique can be helpful among raw data that have a strong correlation, which means higher redundancy. Generally, a PCA can be considered as a method that transforms several initial input features into non-correlated features (i.e. Principal Components PCs). For instance, Milovanović & Popović (2012) carried out a study where the PCs were calculated as 58% and 30% of total variance for the first PC1 and the second PC2, respectively. Where, both PC1 & PC2 were used in data classification among healthy people

and chronic stroke patients. Pearson's correlations were conducted for hip, knee and ankle movement data. The results showed that the hip and knee had a low degree of correlation ($r = 0.09-0.22$), while a moderate degree of correlation was present in the case of the ankle.

In the same context of using PCA for data reduction, Phinyomark, Petri, Ibáñez-Marcelo, Osis & Ferber (2018) identified a significant correlation in the first few PCs, when they were used in an investigation into the effects of three conditions of shoe midsole hardness (soft, medium and hard) using kinematic data analysis during the running test. A further example by Nigg, Baltich, Maurer & Federolf (2012) showed that the first 35 Principal Components (PCs) reached a variance in the dataset by 95.6%. However, the classification accuracies among the three conditions were explained as 99.5, 95.6 and 86.0% in the following groups of hard/soft, hard/medium and medium/soft shoe midsole hardness.

As a classification solution, the studies show the accuracies as between 80% to 100% regarding classification of gait pattern changes into different classes (Eskofier, Federolf, Kugler & Nigg, 2013) (Maurer, Federolf, von Tscherner, Stirling & Nigg, 2012), (Nigg, Baltich, Maurer & Federolf, 2012). In these analyses, the high and medium order of the PCs are eliminated, while the low order of the PCs (i.e. the first few PCs) are retained. The PCA approach is also used in data dimensionality reduction for gait disorder discrimination. For instance, Slijepcevic, Zeppelzauer, Gorgas, Schwab, Schüller, Baca & Horsak (2017) recorded the best trade-off among data reduction and improvement of the classification performance, with the results reaching 98% of the variance in data. In a similar context, kinematic angles and kinetic features were analysed over the stance phase of a gait cycle. In this study, Foch & Milner (2014), considered the first three PCs that represented a variance from 93.3 to 99.4% in five-biomechanical waveforms including frontal plane trunk, frontal plane pelvis, frontal plane hip, transverse plane knee angles and frontal plane knee movement.

3.7 Data Classification Techniques

Machine learning plays a major role in data classification, it is divided into two broad categories based on how to learn from data to distinguish data pattern; they are known as supervised and unsupervised classifiers. The former is trained from the labelled input data (training set) to predict unlabelled data (testing set), this is known as supervised learning. While the latter discovers hidden patterns in unlabelled data and is known as an unsupervised learning approach. This section will include only the supervised learning approaches, that are used for classifying gait pattern changes based on walking speeds.

The data classification stage is the last block of the proposed system, which aims to distinguish among 3-classes by using a subset of gait feature data. This set is split into training and testing data groups, based on k-fold cross validation, where k is equal to 5 and 10 as the most common denominators used in classification solutions. However, supervised learning approaches contain many different methods, the most common of which are the; Decision Tree (DT), linear / nonlinear Support Vector Machine (SVM), k-Nearest Neighbour (k-NN) and Discriminant classifiers which will be examined in more detail in the following sections.

3.7.1 Decision Tree

The Decision Tree (DT) approach is widely used in data mining analysis for interpreting and visualizing useful information. The principle work of DT is based on multiple covariates to predict the target features in classification solutions. This task is performed by using a subset of data as a training set to initiate a decision tree model, while the rest of the data is used to decide on a suitable tree path. The algorithm of DT reaches a decision by following the path that consists of branches (rules) and leaves (outcomes), while the decision nodes represent the relevant features (attributes). The classification procedure starts from the base of the tree (roots) and processes the features at the nodes for testing and reaching the outcomes. Figure 3.3 below illustrates the main components of a DT, which consists of a logic map between parent and child nodes.

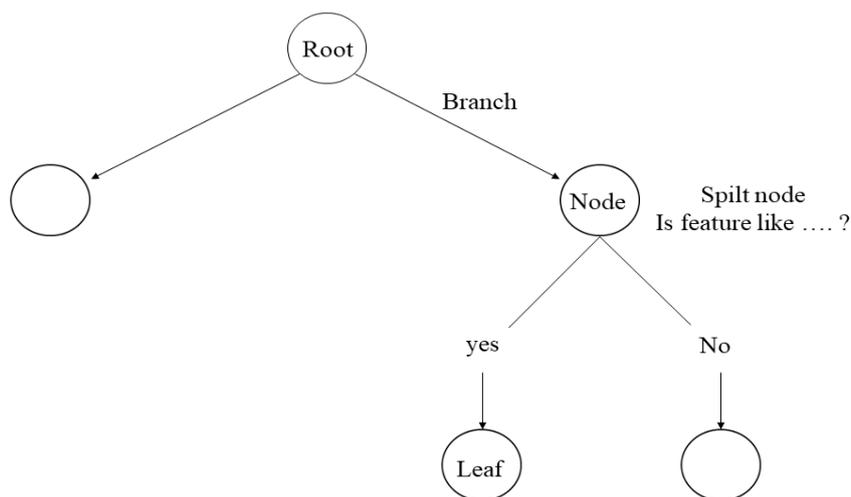


Figure 3.3: Decision tree construction

In gait phases detection, the classification is based on walking speeds, while the DT technique is successfully developed to identify and detect the gait cycle phases (Guo & Jiang, 2015). A C4.5 Decision Tree algorithm has been implemented by (Thongsook,

Nunthawarasilp, Kraypet, Lim & Ruangpayoongsak, 2019) which is trained from a collected dataset for testing data at different walking speeds, then the results are verified by being compared to neural network approaches. Moreover, in a similar context focused on walking conditions, (Farah, Baddour & Lemaire, 2019) have developed a logistic model decision tree algorithm to train and test a dataset that includes knee angle, thigh velocity and acceleration, which was collected using different surfaces (flat, down & up-slopes, right & left cross-slopes) and various walking speeds (1.33, 0.8, 0.6, 0.4 m/s). The results showed that the implemented algorithm had a high accuracy in gait phase detection.

3.7.2 Support Vector Machine Approach

This approach is also known as a support vector network in machine learning and it is categorised as a supervised learning algorithm (Cortes & Vapnik, 1995). The Support Vector Machine (SVM) is efficient in non-linear classification, in addition it performs the classification of linear data domains. In the case of the destruction of the data, linear segregation is inefficient, so the Kernel SVM will convert data from input space into feature space (Berwick, 2003), which provides n-dimensional space with a higher value of n. The SVM algorithm is developed to determine an efficient hyperplane that can identify, as possible as data points that belong to relevant classes (Tong & Koller, 2001). Moreover, the choice of a possible hyperplane that maximizes the distance between data points (maximum margin) will improve the classification accuracy, as can be seen in Figure 3.4 below.

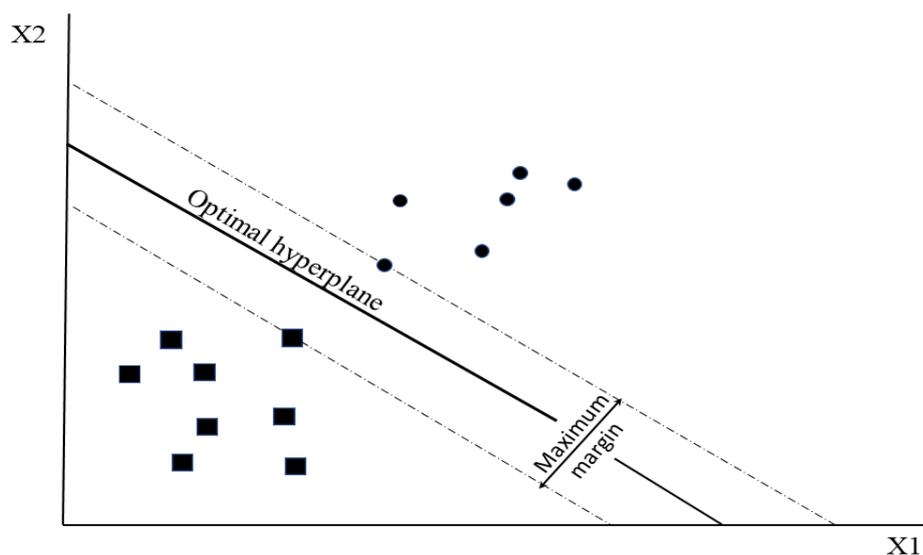


Figure 3.4: Possible Hyperplane in SVM Algorithm

In data classification based on walking speeds, the SVM approach is involved to identify the changes in gait pattern. For instance, (Begg, Palaniswami & Owen, 2005) applied SVM automatically to distinguish between young and old gait by extracting features from a dataset collected from participants, who were instructed to walk on a treadmill at a self-selected walking speed, the results showed that a better classification performance was obtained by SVM over the NN in young/old gait pattern recognition. However, for different walking conditions the SVM was employed by (Lau, Tong & Zhu, 2008) to distinguish among the kinematic data that was collected from five different walk styles including: stair ascent/descent, level ground, and down/up slopes. The SVM performance in gait data classification surpassed the other approaches when compared, including those of the Artificial Neural Network (ANN), Radial Basis Function network (RBF) and the Bayesian Belief Network (BBN).

In gait data analysis, Andersson & de Araújo (2015) exploited the SVM approach in person identification. In this study, a model based on gait analysis used a Kinect camera for full body recognition, which resulted in the effectiveness of the SVM being better than the MLP. While, in human motion detection, computer vision using a Kinect camera is widely explored for clinical assessment. For instance, Leightley, Yap, Coulson, Barnouin & McPhee (2015) exhibited the effectiveness of the SVM in standardized tests that included; jumping, a timed up and go test and a symmetrical assessment. The results were recorded as 85.53, 62.89, 82.42, 79.64 and 71.11% for the Support Vector Machines (SVM), Random Forests (RF), Artificial Neural Networks (ANN) and the Gaussian Restricted Boltzmann Machines (GRBM), respectively.

3.7.3 k-Nearest Neighbour Approach

This approach is a supervised learning algorithm, which employs the whole training dataset for prediction purposes, based on the similarity between a new sample and the training dataset. However, a proposed method introduced by Dramé, Mougine, & Diallo (2014) for clustering a training set based on the value of k and the weight of each cluster, can be performed without the need to use all the points in a training set.

The k -NN approach involves the decision of which class can be assigned to a new sample, with the value of k directly determining the number of data points responsible for voting in a class's assignment. For example, the new sample is 'C', and k is three, which means that the three nearest neighbours can vote to find a class for the new element 'C' as can be seen in Figure 3.5 below. In the region of $k=3$, there are three nearest neighbours, which are near to

'C', this means two votes for 'B' and one vote for 'A'. In this case, the class of element 'C' is going to be 'B', this is very simply how the algorithm k-nearest neighbour works.

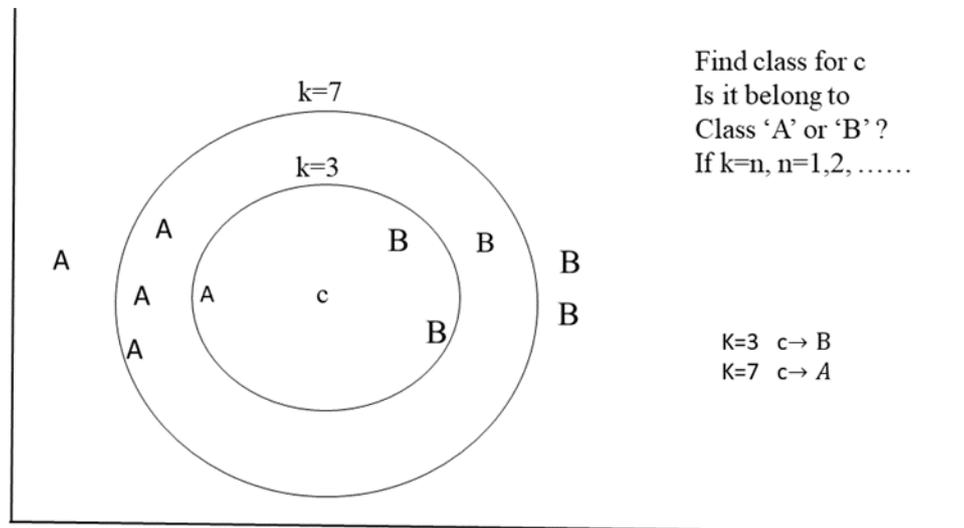


Figure 3.5: The Principle Work of k-NN Approach

In a special case when $k=1$, the feature space representation requires the need to find the nearest neighbour of the element that will define the class, with each training vector defining a region in this feature space. However, if k is chosen as a large value, then the results may be affected by noisy or error data. Research into the improvement of classification accuracy has been widely explored for k-NN classification. For example, Suguna & Thanushkodi (2010) proposed a new approach to increase the effectiveness of k-NN by involving a genetic algorithm to reduce the cost of complexity. In gait pattern classification, the k-NN has been exploited to solve data classification problems. For example, Nieto-Hidalgo et al., (2018) proposed a method that was based on gait spatiotemporal parameters for distinguishing between normal and abnormal gait. They used a smart device for collecting data, while the k-NN approach was used to classify the obtained data, with the results showing that the best accuracy of classification is 95%. In another study, a gait analysis for person identification was used along with the k-NN approach to address the aim of the study. The results showed that both the KNN and SVM approaches yielded the same performance, while the MLP had less accuracy (Andersson & de Araujo, 2015). Similarly, an automatic gait analysis was proposed using a Kinect camera. The study aimed to determine a gait cycle and detect the differences in gait activities. Several machine-learning algorithms were used; k-NN,

Decision Tree, Random Forest, SVM, MLP, and MLPEP approaches, the accuracy performance was recorded as 98.6%, 95.1%, 98.6%, 98.3%, 98.3% and 98.4 %, respectively.

3.7.4 Discriminant Analysis Classification

This technique was introduced in 1963 by Ronald Fisher. The main task of discriminant analysis is to separate two or more groups g_i of observations or events, and then it is used for the classification of new data. In Linear Discriminant Analysis (LDA), the assumption is that the covariance matrix is equal for all classes, but the mean is variable. While the mean and covariance matrix vary for each class, in the case of a Quadratic Discriminant Analysis QDA. LDA calculates the discriminant scores for the observations, to decide which classes belong to them (i.e. Yes or No). The classifier estimates the prediction of a single variable ($X = x$) as:

$$\hat{\delta}_k(x) = x \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2 \hat{\sigma}^2} + \log(\hat{\pi}_k) \quad (3.10)$$

Where $\hat{\delta}_k(x)$ is the estimated discriminant score, $\hat{\sigma}^2$ is the weighted average of the sample variances, $\hat{\mu}_k$ is the average of all the training observations for each kth class, and $\hat{\pi}_k$ is the prior probability of observations that belong to the kth class. The alternative approach of LDA is a Quadratic Discriminant Analysis QDA, which assumes that each class has a different covariance matrix, in mathematical terms, the observation from the kth class is of the distribution $X \sim N(\mu_k, \Sigma_k)$, where Σ_k is a covariance matrix for the kth class. Consequently, the observation is assigned to the class based on the largest of $\hat{\delta}_k(x)$ see (3.11).

$$\hat{\delta}_k(x) = -\frac{1}{2} x^T \Sigma_k^{-1} x + x^T \Sigma_k^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \Sigma_k^{-1} \hat{\mu}_k - \frac{1}{2} \log |\Sigma_k| + \log(\hat{\pi}_k) \quad (3.11)$$

Machine learning algorithms are used widely for solving data classification problems. With the different methods of data classification providing an opportunity to choose the best approach that can be efficient for gait pattern classification. The choice can be addressed through the evaluation of both system performance and cost. For example, important factors include complicity, computational time, availability (versatility) and classification accuracy, which are usually dependent on the application and dataset itself. On the one hand, the DT approach is easy to generate and understand, since it is based on logical rule and non-parametric training data, with less computing time and the capacity problem is avoided. However, one significant limitation of the DT approach is the large amount of time required for training the large data set, as well as the overfitting problem and the fact that the accuracy can be affected

by the selected feature. While, the SVM approach has high accuracy in classification and prediction solutions, and it is good at dealing with high dimensional data. However, the limitations of this approach due to performance improvement, result in the computational cost being intensive, with the parameter tuning also becoming time-consuming (e.g. the kernel parameters and the hyperplane adjustment). On the other hand, the k-NN approach is a simple algorithm for nonlinear data, with no assumptions for data processing and it is useful for classification or regression (versatility). However, this approach suffers from some limitations including the computational cost as it becomes expensive because of the storage of all the training data. Therefore, enough memory is required, and the processing time is consumed with big data for the prediction stages.

3.8 Cross validation technique

The purpose of the CV technique is to assess the effectiveness of the model that is built for machine learning, particularly for understanding the overfitting problem. This leads to the determination of model parameters (hyper-parameters) that can reduce error, when the model is used for testing unseen data. In other words, it emphasises that the model pattern is based on the correct data, and not on the noise.

3.8.1 Holdout Method

This method is the simplest in cross validation techniques, it separates the data set into two parts randomly, which are called either a training or testing subset. The former is used to fit the function of a model (learner), while the latter is used to predict the output for unseen data, they are also known as single training and testing sets, see Figure 3.6 below. The evaluation of a Holdout method cannot be stable if the data division was made randomly, and this significant difference can be related to the points of data that either belong to the training or the testing subset. Although a Holdout method saves computational cost, its single train-and-test trial may lead to an error rate estimation due to the unfortunate split.

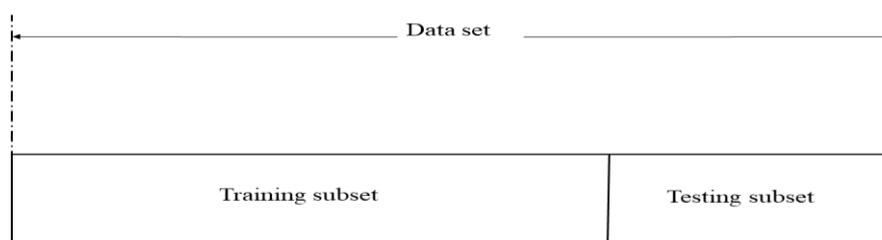


Figure 3.6: Holdout Method for Single-Training/Testing Sets

3.8.2 Leave-P-Out Cross Validation

This method excludes a part of the original data called ‘P’, and the rest of the data is training data. For example, if the original data has data size ‘N’, then the N-P is called a training data set, while the P points of data are called the validation data set. This process is repeated by shifting ‘P’ points over all the original data without an overlap occurring between the P set and N-P set, and then the average of errors is calculated for all trials. The special case of this method is when ‘P’ is equal to one, which is called Leave One-Out Cross Validation LOO CV, as can be seen in Figure 3.7.

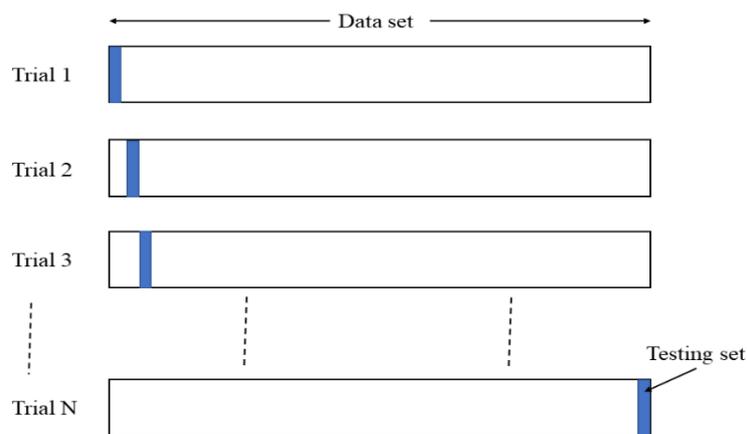


Figure 3.7: Leave One Out Cross Validation

In the LOO CV method, the dataset size N is split into the training subset as N-1, and the rest of the data as a testing subset. In each trial, the procedure will be repeated by shifting the testing set until it reaches the last data point of the test dataset.

3.8.3 K-Fold Cross Validation

In the k-fold Cross Validation (k-fold CV), the original dataset is randomly categorised into k parts (folds) that are equal sizes of the data points. A single part is used for testing the model, and the rest of the k-1 parts are used for training data to fit the model. As shown in Figure 3.8, each data point can be used in a testing set one time, with k-1 times for the same data point represented in a training set without overlapping occurrence. This leads to an interchange of the data points between the training and testing processes, which increases the effectiveness of this method. The CV process is then repeated in k trials. In each trial, exactly one-fold is used as the validation data, and then the k results are obtained. For a single

estimation, the average (or otherwise combination) can be conducted for the k results to achieve total model effectiveness.

In machine learning, the main task of a Cross Validation (CV) technique is to assess the skill of the learner model on unseen data for prediction. This is carried out by partitioning the original dataset into a training subset to fit the model, and a test subset for model evaluation. There are several methods for this task; including the holdout method, leave one-out and k-fold cross validation.

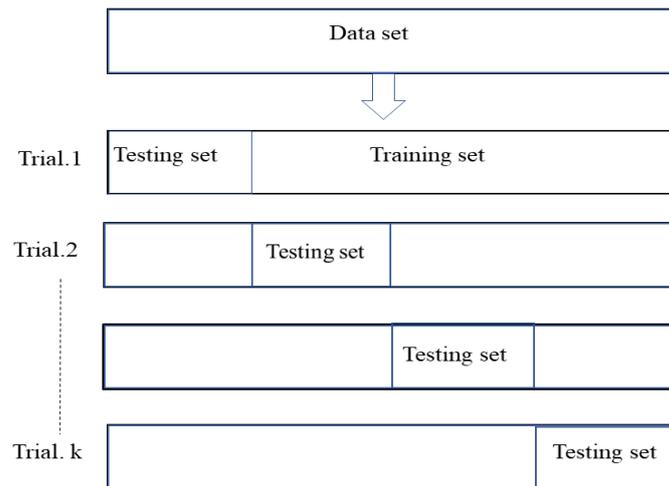


Figure 3.8: k-Fold Cross Validation

However, the k-fold CV provides some features that do not exist with the other methods, as it provides ample data for training processes and leaves ample data for model testing with less computational cost compared to the LOO CV. The main advantages of the k-fold CV are that all the data points are exploited in both the training and testing processes, with each data point represented in the model testing only once. This can significantly reduce bias as most of the data is used for fitting, which decreases the variance because most of the data can be represented in a testing set. In practical terms, this method showed acceptable results for $K = 5$ or 10 , but nothing is fixed, and it may take any value.

3.9 System performance evaluation

The evaluation and comparison of system performance methods is paramount to understanding the effectiveness of a model. For data classification problems, the evaluation metrics can highlight effort, accuracy and difficulties, as well as analysing pitfalls among the different models. Furthermore, the results of the classification evaluation can assist in the

improvement of the system performance, by adjusting and modifying the main parameters of the classifier (back adjustment).

The most commonly used evaluative metrics for the performance of data classification are; Sensitivity or True Positive Rate (TPR), Specificity or True Negative Rate (TNR), Precision or Positive Predictive Value (PPV), Accuracy, Receiver Operating Characteristics (ROC) curve and the F-measure. In more detail, the True Positive (TP) represents that the abnormal is predicted correctly, while if the normal is correctly predicted this is called the True Negative (TN). Meanwhile, the False Positive (FP) represents that the abnormal is predicted incorrectly and the False Negative (FN) is when the normal is predicted incorrectly.

3.9.1 Receiver Operating Characteristics

Receiver Operating Characteristic (ROC) curves play a major role in the evaluation of unseen data prediction, by detecting the true state from a dataset and comparing the different test results for the same case (Kumar & Indrayan, 2011). The evaluation of system performance is a matter that concerns both researchers and clinicians. For this purpose, the target is not only to confirm that the positive/negative data are defined, but also to rule this out for the negative/positive data to be measurable if they are incorrectly predicted. The demand of the ROC curve is represented by its ability to illustrate the relationship between the sensitivity and specificity for the predictor. In more detail, the ROC curve refers to the plot of True Positive TP rate (sensitivity) on the Y-axis, and the False Positive FP rate ($1 - \text{specificity}$) on the X-axis (Schwartz, 2012).

The area under the curve (AUC) is made from a combination of sensitivity and specificity measures, to assess the performance of a prediction result. Significantly, when the AUC equals one this means the test performance is perfect in differentiation between the positive and negative cases (Hajian-Tilaki, 2013). In other words, both the sensitivity and specificity scales can reach one, if both the FP and FN rates are zero. This arises only when the distribution of positive and negative are not in overlap (Swets, 1979), which only happens in an ideal case, as can be seen in Figure 3.9. The effectiveness of the AUC was investigated by Morrison (2005), where data was efficiently discriminated when the AUC gave approximately one, while there was less ability in discrimination when the AUC was around 0.5. In the same context, Morrison, Coughlin, Shine, Coull & Rex (2003) used the ROC curves to distinguish amongst healthy and unhealthy people.

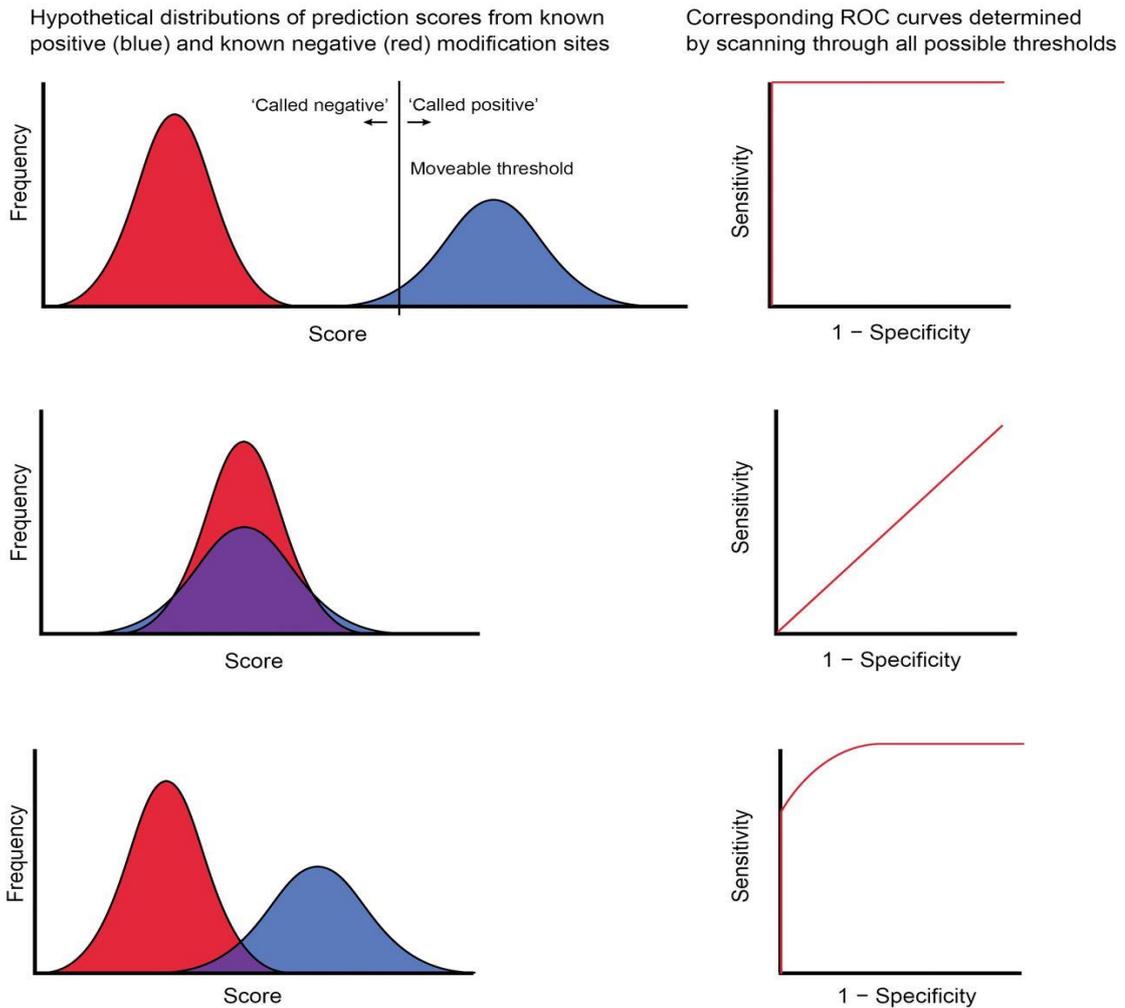


Figure 3.9: Three ROC Curves that Correspond (Right Side) to Overlapping Distributions (Left Side) (Source: Schwartz, 2012)

3.9.2 Confusion Matrix

The confusion matrix is exploited in machine learning to describe the performance of the classification model. This evaluation metric is also called an error matrix, which can illustrate whether the classified data has predicted correctly or incorrectly. It provides visualisation for two or more classes of predicted data, to assess the classifier’s effectiveness in discrimination among classes. Although several evaluation metrics can be used through the confusion matrix, the chosen measures depend on the needs. However, for model performance comparison, the accuracy, F-measure, sensitivity and specificity measures are most commonly adopted in the evaluation of a classification model.

The confusion matrix consists of rows and columns in classes, the former is for actual classes while the latter is for predicted classes, as shown in Figure 3.10 below. The model can

be considered as perfect if the matrix contents show only the values of a true positive and true negative (diagonal values). Although, this score is not likely to exist, an alternative is when the matrix shows the lowest error values, for instance false positive (type I error), false negative (type II error) or error overall.

	Predictive Positive	Predictive Negative	F-measure = $\frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$
Actual Positive	True Positive TP	False Negative FN	Sensitivity $\frac{TP}{TP + FN}$
Actual Negative	False Positive FP	True Negative TN	Specificity $\frac{TN}{TN + FP}$
	Positive predictive value $\frac{TP}{TP + FP}$	Negative predictive value $\frac{TN}{TN + FN}$	Accuracy = $\frac{TP + TN}{TP + TN + FP + FN}$

Figure 3.10: Confusion Matrix for Two Classes with Some Measures

In the above figure, the confusion matrix shows a set of measures that play a major role in the evaluation of a classification model, based on the elements of the matrix; Positive True (PT), Negative True (NT), False Positive (FP) and False Negative (FN). The most important elements that represent the diagonal of the matrix and the high accuracy of a classifier depend directly on the diagonal values. The evaluation metrics are illustrated in Table 3.2 below.

Table 3-1: The Main Measures that are Commonly Used in Model Evaluation using the Confusion Matrix.

True Positive Rate (Sensitivity or recall)	$TP / (TP + FN)$
True Negative Rate (Specificity or selectivity)	$TN / (TN + FP)$
Positive Predicted Value (Precision)	$TP / (TP + FP)$
False Positive Rate (Fall-out)	$FP / (TN + FP)$
False Negative Rate (Miss Rate)	$FN / (TP + FN)$
Accuracy	$(TP + TN) / (TP + FP + TN + FN)$
F-measure	$\frac{2 \times \text{precision} \cdot \text{Recall}}{\text{precision} + \text{Recall}}$

The confusion matrix and the ROC curve are effective approaches in observing the required properties of several classifiers. These evaluative metrics can be used in the determination of the gains and limitations of the classification performance. The capability of these metrics is not only in the rate evaluation of the ability of the classifier to predict correctly, but also in the evaluation of the incorrect prediction rate, which is very important criteria for

distinguishing among the data classification approaches by calculating their sensitivity, specificity, precision and other.

3.10 Summary

A gait analysis system can be implemented using a Kinect camera, by including several stages; collection, smoothing, extraction, reduction and classification of the relevant data to detect and rank gait pattern changes. For this purpose, efficient approaches will be involved to contribute in the improvement of measurement accuracy. This research explores how to enhance the quality of data during its processing, via the three stages of a gait analysis system (smoothing, extraction and classification of 3D skeleton data) to improve the performance of the whole system.

In the literature review, the main limitation can be identified is the inaccurate measurement in data collection, when a Kinect camera was used comparing to other motion capture systems, due to noisy and low data rates. However, the noise and error rate can be reduced by using a smoothing and filtering technique. For this purpose, the filter will be chosen using criteria which retains the original shape of the data, because most of the gait information is located in the amplitude of the gait signal. The first criterion is time delay, where the timing of the filtering process (i.e. the time between the input and output data of the filter) needs to be as short as possible. The second criterion is the fast response of the output filter to its input data, as the information is located in the amplitude of a gait signal, especially in the concave-up and concave-down, which corresponds to the gait stride length. A filtering approach will be considered if the filtered data contains gait information fast enough so that it can retain the original shape of the data.

In the gait feature extraction stage, the low data rate of a Kinect camera can directly affect the measurement accuracy, subsequently the quality of the extracted data will become affected. For these reasons, the mapping of the collected data into an AM domain is proposed to extract a new gait feature. An AM domain can improve the quality of extracted feature data by increasing the sampling frequency rate of the modulated gait length signal.

In the gait data classification stage, most of the classifier techniques work based on boundary decisions that can classify data points into relevant classes. Therefore, the classification accuracy will increase if there is enough separation between classes (i.e. clear cut or discontinued data). An efficient classifier can predict classes correctly for data points that

are located on the boundaries of classes (i.e. gradual data), where the boundary decision is difficult to be defined. For this purpose, the CE technique is proposed as a data classification solution, as it uses a different approach to calculate the similarity rate among data points by measuring the Hamming Distance (HD) between the codewords. Where the codewords mean the gait features have been converted into binary format rather than a decimal format. This increases the accuracy of the similarity rate among the codewords because the measuring is based on each bit within a codeword length. In addition, the position of the bit that differs from the threshold (i.e. $HD = 1$), will be considered and weighted based on the order of the bits' position within HD length. Subsequently, the error rate can be calculated by summation of the elements of the HD (i.e. numbers of 1's with its weight). If the error rate is low this means the similarity is high and the possibility of the data point belonging to the corresponding class will be high.

4. RESEARCH METHODS AND PROPOSED TECHNIQUES

This chapter focuses on the methods that have been used in the acquisition and pre-processing of 3D skeleton data, followed by a gait cycle detection. In addition, it explains the proposed mathematical approaches that have been used in the gait feature extraction and classification. Finally, it describes the evaluative metrics that are used to assess the system performance.

4.1 Introduction

The main goal of this research is to effectively extract gait features during walking tests using a Kinect camera. The obtained data is usually considered as noisy. Consequently, the pre-processing stage is required to decrease the level of error and smooth the dataset without affecting the features' information. In the feature extraction stage, an Amplitude Modulation (AM) technique is proposed to modify the gait step signal for extracting new gait features which would be more clearly represented. For this purpose, the seven gait features can be extracted using a modified gait signal, while the Principle Component Analysis (PCA) algorithm is chosen to reduce data dimensionality in order to produce the most representative gait features.

In the data classification stage, a Convolutional Encoder (CE) technique is proposed to classify the extracted gait features. Consequently, the performance of the CE can be evaluated by comparing its classification accuracy to various commonly used classifiers. For this purpose, a set number of supervised classifiers, including the SVM, k-NN, DT and discriminant classifiers are employed to classify the obtained data, based on walking speeds. In addition, a set number of evaluative metrics, including sensitivity, specificity, precision, accuracy, F-measure and AOR curve approaches are applied to evaluate the performance of each approach.

4.2 Instruments and Data Acquisition

Microsoft Kinect is a camera that combines several sensors for providing information, such as RGB colour and Depth information (Raposo, Barreto & Nunes, 2013), with with a data rate that is equal to 30 frames per second (Otte et al, 2016). Its depth sensor consists of two

infrared (IR) sensors - a projector that emits a dot pattern into the scene and a detector for detecting the dots from the scene. 3D skeletal positional data is used in this study, as it can guarantee the participants are not subject to a breach of privacy, which is a concern that has priority for many researchers (Henry, Krainin, Herbst, Ren & Fox, 2012).

To obtain the 3D skeletal joint information, the participants are directed to walk forwards towards the Kinect camera. Then, the Microsoft Kinect and the Software Development Kit (SDK) are used to track the skeletal joints movement in real time (Shingade & Ghotkar, 2014). The Microsoft SDK is a middleware framework supported by the Kinect camera - twenty-five joint positions can be tracked using MS Kinect SDK. The depth information corresponds with the estimation of the distance between the Kinect camera and each pixel located in the scene (Nguyen, Izadi & Lovell, 2012). Since this research is concerned with human gait analysis, which is directly conducted from the lower body joints data, the Kinect camera is used to track the movement of several body joints in three dimensions (3D) provided as a data rate of 30 fps. For this purpose, the seven lower joints are mentored and tracked, including spine-base, left/right hips, left/right knees, and left/right ankle joints.

The skeletal positional data of lower body joints were collected to generate the gait step length signal for a total of 40 participants. The collected data was processed under three conditions of walking speeds (low, normal and fast). A MATLAB code was developed to process a dataset for gait feature extraction and classification (see Appendix D).

4.2.1 Kinect Camera View

The location of the camera during a walk test is important in order to maintain the object in the camera view. The angle view of the Kinect v2 is specified as 60° and 70° for vertical and horizontal views, respectively (as shown in Figure 4.1 below). This leads to a calculation of the height position of the Kinect from the ground, and its location from the scene, which can guarantee that at least two gait cycles will be captured during the walk trial.

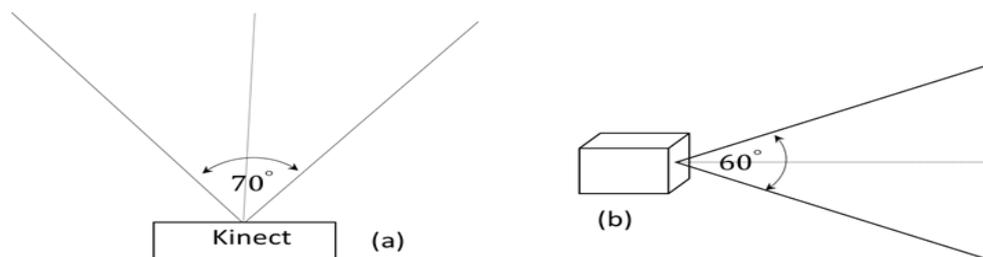


Figure 4.1: Kinect v2 camera view (a) horizontal view (b) vertical view

The camera accuracy for tracking the object's position will be increased whenever the object is closer to the camera. However, the number of gait cycles that can be captured will increase whenever the subject walks further away from the camera. These calculations are used to guarantee a closer walk to the camera and allow several gait cycles to be captured per individual, without distortions that might be caused by subjects moving in or out of the sensor's field of vision. The height distance of the camera from the ground is chosen according to the placement of the individual as close to the camera as is possible. Equations from (4.1) to (4.3) are used to calculate the skeletal captured area of the camera by using the angled views of the Kinect.

$$\sin(\theta) = \text{opposite} / \text{hypotenuse} \quad (4.1)$$

$$\cos(\theta) = \text{adjacent} / \text{hypotenuse} \quad (4.2)$$

$$\tan(\theta) = \text{opposite} / \text{adjacent} \quad (4.3)$$

In the horizontal view of the Kinect, the distance of the camera view is denoted as 'v' (as shown in Figure 4.2 below). This distance can be calculated by halving the horizontal angle of the camera view to 35°, which produces the right-angled triangle.

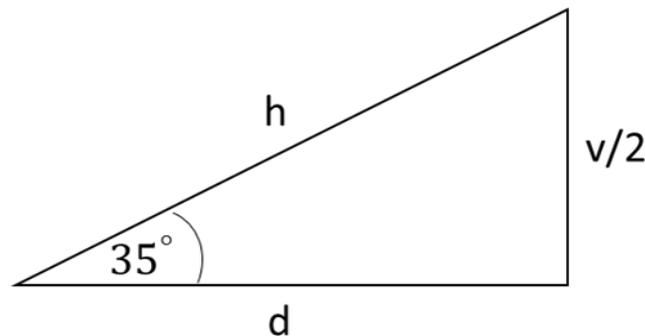


Figure 4.2: Horizontal Angle View of Kinect Camera v2

By substituting the values of the triangle lengths in (4.3), the distance of the camera's view (v) is calculated as: $\tan(35) = 0.5v/d$, where (v) can be written as $1.4(d)$, and (d) can be chosen as any value that guarantees at least one gait cycle is captured. For example, if (d) equals 2 meters, then the captured skeletal distance (v) will be equal to 2.8 meters.

In case the walk line is in front of the camera, the vertical view of the Kinect v2 (60°) can be halved to obtain a right-angled triangle, as can be seen in Figure 4.3 below. The camera's field of vision (v) is calculated from the Equation (4.3) which yields $(1.15 \times d)$. For the different

values of (d) as 1, 2, 3 & 4 m, then (v) will be equal to 1.15, 2.30, 3.45 & 4.6 m. This shows that the best position of the camera height from the ground is around one meter, which guarantees that the whole body is in the area captured by the camera.

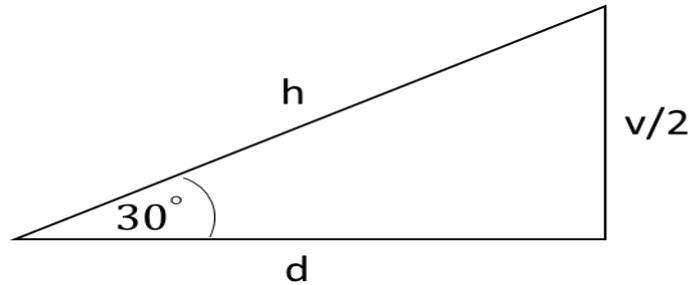


Figure 4.3: Vertical Angle View of Kinect Camera v2

For this experiment, the participants were recruited from a university campus. All subjects were students aged between 22 and 48 years. Each of the subjects who agreed to participate in the experiment provided gender, height and weight information. Everyone generated about 70 to 350 frames and completed between 2 and 7 gait cycles per walk, dependent on the walk speed.

4.3 Framework and Proposed Approaches

The purpose of the proposed method is to develop an automated gait analysis, using skeletal positional data of leg joints, for the extraction and classification of gait pattern changes. The six stages are involved, namely: skeletal data collection, pre-processing, data conversion, extraction of gait features & reduction, gait data classification, and system evaluation (see Figure 4.4 below).

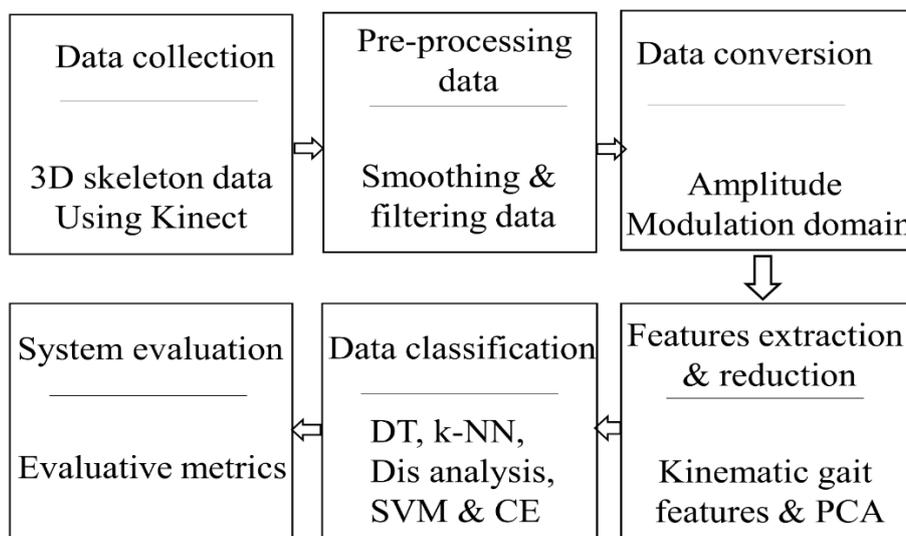


Figure 4.4: General Block Diagram of a Human Gait Analysis with Six Stages

In the first stage, 3D skeleton data is collected for the lower body joints using a Kinect camera v2. Skeleton data is usually considered as inaccurate due to its noise and error rate. For this purpose, six different filter techniques are involved for the pre-processing of data, which is applied in the second stage. An efficient filter is used that can perform data smoothing and filtering without information distortion of the original data - the filtering process is evaluated based on time delay and fast response between input/output data of filter. In the third stage, gait cycle determination is calculated using the maximum distance between ankles, which represents the gait step length. The fourth stage is feature extraction using spatiotemporal analysis. Moreover, an AM technique is proposed for the extraction of gait features that can detect the changes in gait parameters more efficiently, followed by feature reduction (using a PCA algorithm). The gait data classification solution is performed in the fifth stage using various supervised classifiers.

In addition, a CE technique is proposed to classify the changes in gait pattern. Finally, several approaches of evaluative metrics are used in the system evaluation stage. In this research, MATLAB v2018a (MathWorks Inc., Natick, Massachusetts, United States) is used for the process, extraction and classification of the skeleton data collected by an MS Kinect v2.

4.3.1 Skeletal Data Smoothing and Filtering

Skeleton data that is provided by a Kinect camera typically contains noise during the tracking process (Wang, Kurillo, Ofli & Bajcsy, 2015). This can be related to different causes such as the disparity of the camera's data rate to object speed, occlusion of body parts, and light conditions (Edwards & Green, 2014). To reduce as much noise as possible from the skeletal data, a set number of filter techniques were used for filtering and smoothing the collected data of skeletal positional joints. However, the filtering process causes latency, and this may be increased when the data is aggressively smoothed. For this reason, the study considers that the discrimination between filters will be based on time delay, the fast response between input & output data, the data smoothness degree, and the fact that the maintenance of the shape of original data is not distorted in order to guarantee the gait information.

In terms of the concept of a trade-off between latency and smoothness, the parameters of filters are tuned to meet the requirements, where the tuning process of the filters' parameters is continued until the noise (spikes) is removed or reduced as much as possible, while the delay between the input and output of the filter is observed. This task can be achieved without causing

distortion in the original data. Then, a filtering method is carefully chosen that best matches the specific needs, such as the lowest delay, highest smoothness, or the fastest response that can track the original data, especially when the curve is turned up or down (concave up and concave down). For example, the ankle joint position data represents the most important information that indicates the beginning and ending of a gait cycle. This information corresponds to the edges on the curve when it is turned up or down and depends on the direction of the joint movement. The application of a different kind of filter that has different degrees of smoothness and latency will be considered in the discrimination between them. In this research, several filters are used for the smoothing and reduction of noise, including the Average Moving, Savitzky-Golay, RLOESS, Median filter, Kalman filter and Exponential Average Moving filters.

- **Moving Average Filtering**

This technique is a simple Low Pass FIR (Finite Impulse Response) filter commonly used for smoothing a row of sampled data. The input samples will be divided by N-samples and averaged to produce a single output point. The first N-sample is processed and then shifted forward by excluding the first element of the array and including the next element - over all elements of an array, the moving average filter can be mathematically expressed as (4.4) (Koswatta & Karmakar, 2010):

$$y(i) = \frac{1}{N} \sum_{j=0}^{N-1} x(i + j) \quad (4.4)$$

Where $y(i)$ is the smoothed signal, $x(i)$ is the input signal, i is order of data point, and N is all neighbouring data in the average within the window.

- **Savitzky-Golay Filter**

The principle work of an SG filter is to fit a polynomial order to the subset of data. The data points are replaced by an unweighted average value that is calculated from its neighbouring points (Shajeesh, Kumar, Pravena & Soman, 2012). Then, the window will be shifted from one sample to fit a polynomial and evaluate the central location (Schafer, 2011). This will be repeated over all samples of data. The coefficients of a polynomial for a sequence of samples are calculated as (4.5) (Savitzky & Golay, 1964):

$$P(n) = \sum_{k=0}^N C_k n^k \quad (4.5)$$

Where $p(n)$ is the approximate value that corresponds to the n^{th} data sample in the window ($-M \leq n \leq M$), M is the points number of data (i.e. $2M+1$ is window size including smoothing sample), N is the polynomial order, and C_k is the coefficient of the polynomial. For example, the output (smoothed data) when the interval centre is at ($n = 0$), then the output is $y(0) = P(0) = C_0$. The value of the output at the next sample is obtained by shifting the window to be located at the new centre of the window ($n + 1$) and finding a new polynomial every time.

- **Local regression filtering**

This approach determines a regression surface to fit locally with the parametric functions for the independent variable space based on the weighted least-squares (Cleveland & Grosse 1991). Suppose y_i and $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$, $i = 1, 2, \dots, n$ are the dependent and independent variable measurements, respectively; consequently, the regression functions model can be written as (4.6):

$$y_i = g(x_i) + \varepsilon_i \quad (4.6)$$

Where ε_i is the normal distribution, and $g(x_i)$ is a smooth function of x_i , the local regression uses a local neighbourhood $N(x)$ to estimate x_i in the x space, where the smoothness is increased whenever the local neighbourhood is larger. Generally, locally quadratic (non-linear) fitting performs better in a regression surface compared to a linear fitting (Cleveland & Devlin, 1988). However, the tri-cube weight function is commonly used and defined as (4.7) (Wettayaprasit, Laosen & Chevakidagarn, 2007).

$$w_i(x) = \left(1 - \left|\frac{x-x_i}{d(x)}\right|^3\right)^3 \quad (4.7)$$

Where w_i is the regression weight for data point i , x_i is the nearest neighbours of x as predictor value, and the $d(x)$ is the maximum distance of x_i to x within the selected window. Points that are calculated with 0 weights will be ignored because they are classified as outlier's points. Finally, the estimates of the parameters in (4.6) are the values of the parameters that minimize the coefficients from each local neighbourhood and are used to estimate the fitted values at x_i . Then the ordered pairs give the fitted regression line for the whole dataset (Nurunnabi, West & Belton, 2013).

$$\sum_{i=1}^n w_i(x)(y_i - g(x_i))^2 \quad (4.8)$$

- **Exponential smoothing filter**

This technique uses the exponential window function for the smoothing time series data, while the moving average filter acts like it is weighted equally of observation. In the exponential moving average, the weighting factor of the previous inputs decreases exponentially (Hansun, 2016). This technique is used commonly as a low-pass filter to suppress high frequency components (Casiez, Roussel & Vogel, 2012). The input sequence is often represented by x_n , and the output of the filter is commonly written as y_n , where the simplest form of exponential smoothing is given by the formula (4.9) (Casiez, G et al., 2012).

$$y[n] = \alpha x[n] + (1 - \alpha)y[n - 1] \quad (4.9)$$

Where $y[n]$ is the filter output, $y[n - 1]$ is the previous filter output, and $x[n]$ is the input sequence, and α is ranged between 0 and 1; where $\alpha=1$ means no filtering takes place. The previous output can be repeated and in use will yield the general formula as (4.10) (Goot, Mahalab & Cohen, 2005):

$$y[n] = \alpha \sum_{k=0}^n (1 - \alpha)^{n-k} x[k] + (1 - \alpha)^n y_0 \quad (4.10)$$

Where the weighted sum of all sequence data is decreased exponentially.

- **Median filter**

In this case, a set number of reading points is processed as a discrete time for a certain size of window as $(2N + 1)$ where N is any positive integer (Coyle, Gabbouj & Lin, 1991). The window points are sorted by its values and then the median value is calculated as the filter output for each position of window. The output of the median filter is given as (Qiu, 1994):

$$y(n) = \text{median} [x(n - N), x(n - N + 1), \dots, x(n + N)] \quad (4.11)$$

Where $x(n)$ and $y(n)$ are the input and the output data of the filter, respectively and considering that the window is centered on the n^{th} point.

- **Kalman filter**

The standard algorithm of this filter is described as a dynamical system that contains two main equations: the former is a state equation and the latter is a measurement equation, see (4.12) & (4.13) (Moon, Park, Ko & Suh, 2016).

$$X_t = F_t x_{t-1} + B_t u_t + W_t \quad W_t = N(0, Q_t) \quad (4.12)$$

Where X_t is the state vector, u_t is the control vector, F_t is the state transition matrix, x_{t-1} is the prior state, B_t is the control input matrix, and W_t is the state noise, which is assumed as a normal distribution that is given as a zero mean and the covariance matrix Q_t .

$$z_t = H_t x_t + V_t \quad V_t = N(0, R_t) \quad (4.13)$$

Where z_t is the measurement vector, H_t is the transformation matrix, and V_t is the measurement noise, which is assumed as a normal distribution, that is given as a zero mean and the covariance matrix R_t .

4.3.2 Gait Cycle Determination

A gait cycle is defined as the duration that occurs between two consecutive strike heels of the same foot. Generally, it involves two main phases: the stance stage which takes 60% of the gait cycle period, whilst 40% is taken by the swing stage (Peterkova & Stremy, 2015). The gait cycle can be determined by calculating the horizontal distance between ankles (in z-direction) during a walk (Ahmed et al., 2015), where one gait cycle involves three maximal values, and the maximum distance represents the gait step length, see (4.14).

$$gait\ cycle(k) = \sum_{k=1}^N |R_{ankle(k)} - O_{ankle(k)}| \quad (4.14)$$

Where N is the size of the ankles' data in the z-axis and k is the moment frame that was provided by the Kinect camera in 30 frames per second as data rate, R_{ankle} and O_{ankle} are the positional data movement of the reference ankle and opposite ankle, respectively. The determination of gait cycle is important for the extraction of gait features within one cycle of gait (see Figure B.1 in Appendix B).

4.3.3 Features Extraction

A gait assessment can be performed by extracting some features that can be representative in gait pattern changes, which might indicate a gait disorder that is related to some diseases. The extracted gait features are conducted using several methods, such as spatiotemporal gait analysis and kinematic gait features.

4.3.3.1 Spatiotemporal Gait Analysis

A gait assessment can be evaluated by using a spatiotemporal gait analysis which involves three parts: spatial gait parameters (step length, step width, and stride length), temporal gait parameters (gait cycle time, stance and swing stage time), and spatiotemporal

gait parameters (gait cadence and speed) (Tunca, Pehlivan, Ak, Arnrich, Salur & Ersoy, 2017), (Andersson & de Araújo, 2015). These gait parameters can be calculated as:

- **Gait step length**

Gait step is defined as the maximum distance between two different consecutive strike heels, where the walk process is based on these steps (see 4.15).

$$\text{Step Length} = \max \Delta_x (\text{Refheel}_i - \text{otherheel}_i) \quad (4.15)$$

- **Gait stride length**

The stride length of a gait is also known as a gait cycle length, which involves two consecutive steps of gait. In other words, it is the distance that is limited between two strike heels of the same leg (see 4.16).

$$\text{Stride Length} = 2 \times \text{step length} \quad (4.16)$$

- **Gait cycle time**

The time that is taken to cover one complete gait cycle is called gait cycle time. This can be calculated by subtracting the time at the strike heel from the time of the next strike heel of the same leg (see 4.17), where the Kinect provides 30 frames per second.

$$\text{gait cycle time} = TSH_{i+1} - TSH_i \quad (4.17)$$

Where TSH_{i+1} , TSH_i represent the time at next strike heel and time at current strike heel, respectively.

- **Stance stage time**

This is the time that is taken for the stance stage of a gait cycle. It can be calculated by subtracting the time at the toe-off from the time at the strike heel of the same leg (Mariani, Rouhani, Crevoisier & Aminian, 2013) (see 4.18).

$$\text{Stance Time} = TTO_i - TSH_i \quad (4.18)$$

Where, TTO_i and TSH_i are the time at the toe-off and the time at the strike heel of the same leg.

- **Swing stage time**

The time that is taken for the swing stage of a gait cycle can be calculated by subtracting the time at the next strike heel from the time at the toe-off of the same leg (Moon, McGinnis, Seagers, Motl, Sheth, Wright, ... & Sosnoff, 2017), (see 4.19).

$$\text{Swing Time} = TSH_{i+1} - TTO_i \quad (4.19)$$

- **Double support time**

A gait cycle involves two double support phases (as can be seen Figure B.1 in Appendix B). This double support time can be calculated by accounting for the time that is taken from the strike heel of one leg (R-leg) till the toe-off of the other leg (L-leg); this is added to the time that is taken from the strike heel of the (L-leg) till the toe-off of the (R-leg) (see 4.20).

$$\text{Double support time} = (TTO_i|_R - TSH_i|_L) + (TTO_i|_L - TSH_i|_R) \quad (4.20)$$

Where $TSH_i|_R$ is the time at the strike heel of the right leg, while $TTO_i|_R$ is the time at the toe-off of the right leg. $TSH_i|_L$ means the time of the strike heel of the left foot. $TTO_i|_L$ represents the time at the toe-off of the left leg.

- **Gait cadence**

This quantity could be calculated as the number of gait steps that are accounted for per minute, (see 4.21).

$$\text{Cadence} = \sum \text{gait steps} |_{\text{time(one min)}} \quad (4.21)$$

- **Gait speed**

This is calculated by dividing the stride length over the time of the gait cycle, (see 4.22).

$$\text{gait speed} = \frac{\text{Stride length}}{\text{gait cycle time}} \quad (4.22)$$

Where the gait cycle time (Yoo, Hwang & Nixon, 2005) is defined as (4.23):

$$\text{gait cycle time(sec)} = \text{gait period(frames)} / \text{frame rate(frames / sec)} \quad (2.23)$$

4.3.3.2 Linear Kinematic Gait Features

In linear gait kinematics, the position, velocity and acceleration of the lower body joints are calculated by using the relations between them as vector quantities, as can be seen in Figure 4.5 below. The changes in these quantities (final value from initial value) with respect to the time change (Δt) can be defined as the line slope (s) of joint movement over the time.

$$s = \frac{y_2 - y_1}{x_2 - x_1} \quad (4.24)$$

Where S means the slope of the displacement curve. While, y_2 , & x_2 are final values, and y_1 , & x_1 are the initial values.

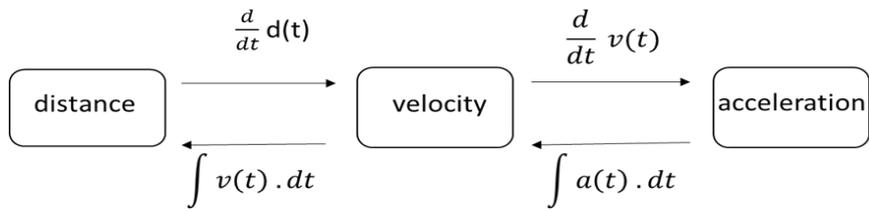


Figure 4.5: The Relationship between Distance, Velocity and Acceleration

In this part, the swing stage is analyzed by using the gait features of displacement, velocity and acceleration, because the foot movement data can be richer than at the stance stage. The time graph of displacement illustrates the relation between velocity and acceleration particularly in the curve line pattern, as can be seen in Figure 4.6. The change of the location is shown as a curve line, which is related to change in velocity. However, the straight-line movement means the velocity is constant over time.

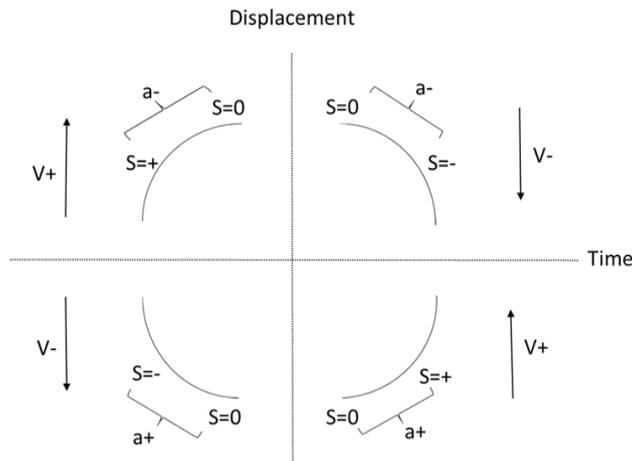


Figure 4.6: Displacement-Time Graph

The velocity sign shows the direction movement - whenever the subject moves to the right, the velocity sign will be positive and vice versa. The slope S is positive when the final value is greater than the initial value and vice versa. The signs of the slope can specify the sign of acceleration - when the direction of movement goes from the small slope into the high value, then the acceleration will be positive and vice versa.

4.3.3.3 AM-modified Gait Signal Features

The Amplitude Modulation (AM) technique maps the baseband signal into the passband domain to generate an AM-modified signal on the higher frequency spectrum (Freeman, 2005). The AM technique modulates the amplitude of a reference signal $x_r(t)$ according to the variation of the amplitude of a gait length signal $g(t)$. The produced

signal is called an AM-modified signal $M_{AM}(t)$ as can be seen in Figure 4.7. The $M_{AM}(t)$ is obtained by multiplying the $g(t)$ to the reference signal $x_r(t)$, see (4.25).

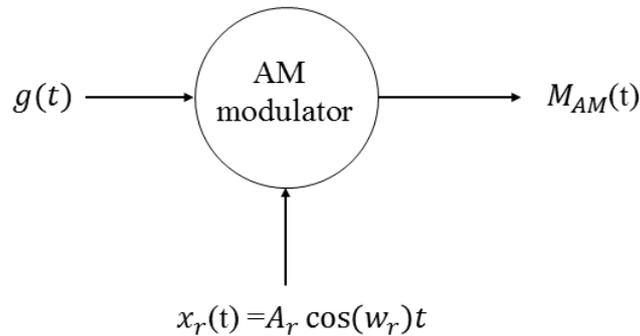


Figure 4.7: AM-Modified Gait Length Signal Generation

$$M_{AM}(t) = A_r \cos(1 + D \cdot g(t))t \quad (4.25)$$

Where $M_{AM}(t)$ is a modified signal for $g(t)$ which is the gait walk signal, A_r and A_g are amplitudes of the reference and gait signal, respectively. The relation between these amplitudes can be formed as the modification depth and it is expressed as:

$$D = \frac{A_g}{A_r} \quad (4.26)$$

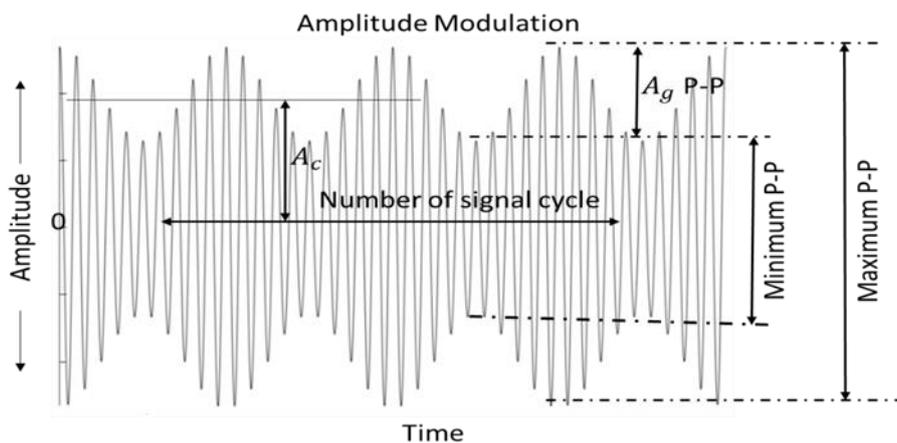


Figure 4.8: Amplitude Modulation Signal in Time Domain

Figure 4.8 illustrates that $A_g = \frac{1}{4}(\max_{p-p} - \min_{p-p})$, and $A_r = A_g + \frac{1}{2} \min(P_P)$. Finally, modification depth in (4.26) can be rewritten as:

$$D = \frac{(\max p_p) - (\min P_P)}{(\max P_P) + (\min p_p)} \times 100 \quad (\%) \quad (4.27)$$

The spectrum of the modification signal equation can be extracted from the general equation (4.25). This leads to a representation of the modification signal in three signals components as:

$$M_{AM}(t) = A_r \cos w_r t + A_r \cos w_r t (D \cdot g(t)) \quad (4.28)$$

Where the $g(t)$ is the gait signal and can be written as $g(t) = A_g \cos w_g t$ for substituting in the equation (4.28) to obtain the following equation:

$$M_{AM}(t) = A_r \cos w_r t + A_r \cos w_r t \times D \cdot \cos w_g t \quad (4.29)$$

The second term of the above equation can be analyzed geometrically to extract the fundamental components of the AM-signal:

$$\cos(A)\cos(B) = \frac{1}{2} \cos(A+B) + \frac{1}{2} \cos(A-B) \quad (4.30)$$

$$M_{AM}(t) = A_r \cos w_r t + \frac{A_r D}{2} \cos(w_r + w_g) t + \frac{A_r D}{2} \cos(w_r - w_g) t \quad (4.31)$$

Finally, the spectrum of the AM-signal is in the simplest form represented by three terms, each one contains a sinusoidal signal with the maximum value of amplitude and is located at a different frequency on the spectrum range. Where, the angular frequencies w_r and w_g can be simplified into f_c and f_g respectively, as can be seen in Figure 4.9 below.

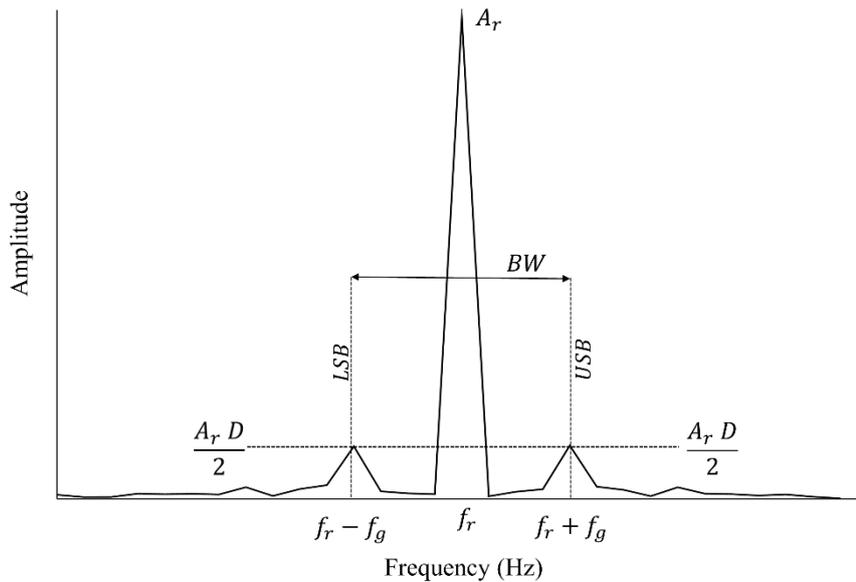


Figure 4.9: AM-Signal Spectrum Contains Three Components

The first term of the equation (4.31) is called a reference signal $x_r(t) = A_r \cos w_r t$. The parameters of this signal (i.e. A_r and f_r) are chosen as fixed values (i.e. $A_r = 1$ and $f_r = 25$ Hz) for all the gait walk tests. However, f_r is chosen as the fourth sampling frequency, and consequently the Nyquist frequency is $(\frac{f_s}{2})$.

The seven gait features are extracted from the spectrum of the AM-modified gait signal $M_{AM}(f)$. These features are listed as: upper side band frequency (f_{USB}), lower side band frequency (f_{LSB}), bandwidth (BW), modulation index (D), modulation efficiency (eff), side-lobes level (sll), and total amplitude level (Ta) which are calculated as the following:

$$f_{USB} = f_r + f_g \quad (4.32)$$

Where, f_{USB} is the frequency of the upper side band spectrum, f_r , f_g are the frequencies of reference signal and gait signal, respectively.

$$f_{LSB} = f_r - f_g \quad (4.33)$$

Where, f_{LSB} is the frequency of the lower side band component.

$$BW = (f_r + f_g) - (f_r - f_g) \quad (4.34)$$

Where, BW is the bandwidth of the AM-modified gait signal.

$$D = \frac{A_g}{A_r} \quad (4.35)$$

Where, D is the modulation index of the AM-modified gait signal.

$$eff = \frac{D^2}{2+D^2} \quad (4.36)$$

Where, eff is the modulation efficiency of the AM-modified gait signal.

$$sll = \frac{A_r \times D}{\sqrt{2}} \quad (4.37)$$

Where, sll is the side-lobes level of upper side band and lower side band components.

$$Ta = \frac{A_r}{\sqrt{2}}(1 + D) \quad (4.38)$$

Where, Ta is the total amplitude level of AM-modified gait signal.

4.3.3.4 FM-modified Gait Signal Features

The FM-modified signal $M_{FM}(t)$ is generated by using a Frequency Modulation technique that changes the frequency of the reference signal f_c according to the variation of the amplitude of a gait walk signal $g(t)$; then mathematically, the instantaneous frequency f_i of $M_{FM}(t)$ (Maini, 2011) is represented as:

$$f_i = f_r(1 + k_f A_g \cos w_g t) \quad (4.39)$$

Where k_f is the constant proportionality, f_r is the reference signal frequency, and A_g/ w_g are the amplitude and angular frequency of the gait signal, respectively. From the above equation, the f_i can reach the maximum value when $\cos w_g t$ equals positive one, and its minimum value when $\cos w_g t$ equals negative one. These yield $f_{max} = f_r(1 + k_f A_g)$, $f_{min} = f_r(1 - k_f A_g)$, respectively. Consequently, the frequency deviation Δf can be calculated from these changes of instantaneous frequency and compared to the original frequency of the reference signal as $(f_{max} - f_r)$ or $(f_r - f_{min})$.

$$\Delta f = k_f A_g \quad , \text{measured in } \left(\frac{\text{Hz}}{v} \times v\right) \quad (4.40)$$

Where Δf is the frequency deviation which is defined as the quantity of frequency that has changed from the frequency of the reference signal as can be seen in Figure 4.10 below.

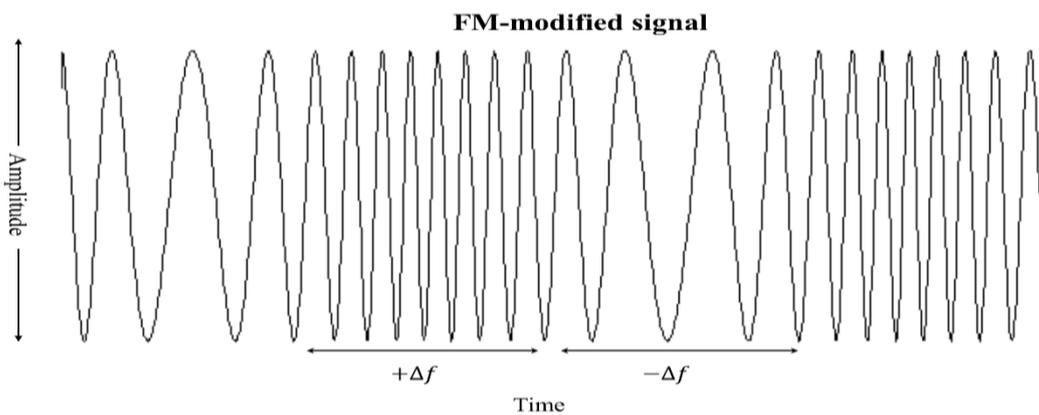


Figure 4.10: Frequency Modulation Signal in Time Domain

In general, the Frequency Modulation expression explains the relation of the information signal amplitude to the reference signal frequency (Neustein, 2011), (see 4.41).

$$M_{FM}(t) = A_r \cos(w_r t + B g(t)) \quad (4.41)$$

Where, $g(t)$ is the gait signal and A_r, w_r are the amplitude and angular frequency of the reference signal, respectively. B is the modulation index (Faruque, 2017) and can be represented as (4.42).

$$B = \frac{\Delta f}{f_g} \quad (4.42)$$

If the modulation index is $B < 1$, then the modulation is called the Frequency Modulation Narrow Band (FM-NB), and when $B \geq 1$, the modulation is called the Frequency Modulation Wide Band (FM-WB). In this research, the FM-NB technique will be chosen because it is similar to the AM technique and they have the same number of spectral components; this increases the measurement accuracy when they are in compression for the data gait analysis, especially in the frequency domain representation.

In the frequency domain, the spectrum of the FM-modified signal $M_{FM}(f)$ contents more than one side of the spectral components in the case of FM_WB. However, the modulation index B is less than one with (FM-NB). For this, assume that $B \approx 0$, and then the Equation (4.41) can be rewritten as:

$$\cos(a + b) = \cos a \cos b - \sin a \sin b \quad (4.43)$$

$$M_{FM}(t) = A_r [\cos (w_r t) \cos (B g(t)) - \sin(w_r t) \sin(B g(t))] \quad (4.44)$$

As $B \approx 0$, then $\sin (B g(t)) \approx B g(t)$, and $\cos(B g(t)) = 1$. Therefore, the above equation will be simplified as:

$$M_{FM}(t) = A_r [\cos w_r t - (\sin w_r t \times B g(t))] \quad (4.45)$$

If the gait length signal is $g(t) = \sin w_g t$, then the above equation can be rewritten as:

$$M_{FM}(t) = A_r \cos w_r t - A_r (\sin w_r t \times B \sin w_g t) \quad (4.46)$$

$$M_{FM}(t) = A_r \cos w_r t + \frac{A_r B}{2} \cos(w_r - w_g)t - \frac{A_r B}{2} \cos (w_r + w_g)t \quad (4.47)$$

The above equation is the simplest form of the FM-NB signal which contains three sinusoidal signals and each one can be represented as a spectral component with peak amplitude and frequency location on the spectrum range as illustrated in Figure 4.11 below.

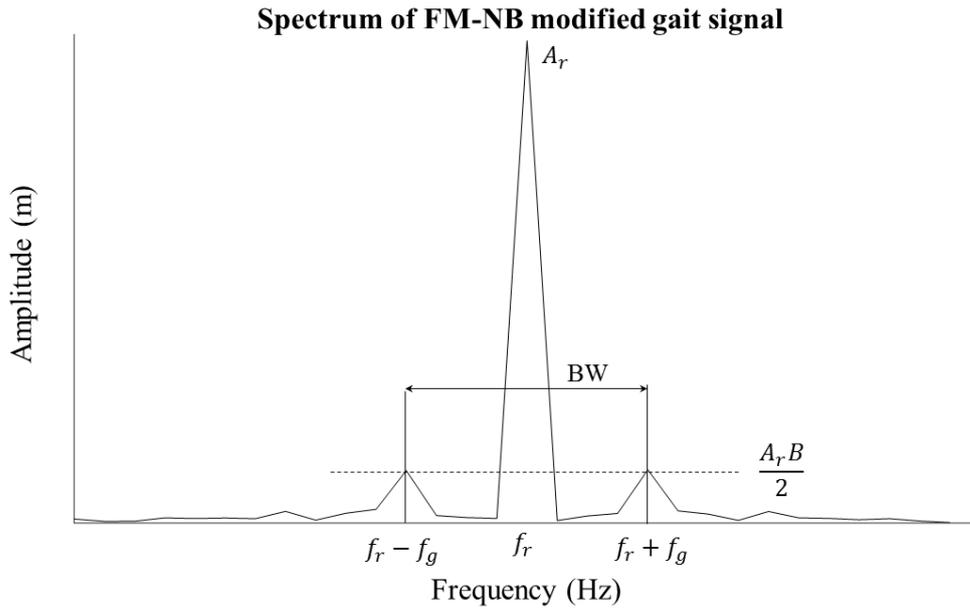


Figure 4.11: Spectrum Range in FM-NB Modulated Signal

The bandwidth BW of the FM-modulated gait signal can be calculated by subtracting the frequency of the LSB from the frequency of the USB component as (4.48).

$$BW = (f_r + f_g) - (f_r - f_g) \quad (4.48)$$

$$BW = 2f_g \quad (4.49)$$

4.3.4 Gait Features Reduction

In this section, principal component analysis (PCA) technique is used to reduce the dimension of the dataset matrix. Before using the PCA approach it is common to use the features scaling method (normalization).

4.3.4.1 Features scaling methods

The features can be rescaled using several methods and the scaling may alter distribution of the original data to improve the quality of the results. This is required when the dataset has variation between the features in magnitudes and ranges. To reduce this impact, the scaling can present several features to be at the same level as the magnitudes.

- **Standardization**

The data points are replaced by the z-score (Gamble, Ravela & McGarigal, 2008) using the standardization, (as seen in 4.50). This is called the features redistribution, with the mean equal to zero and the standard deviation equal to one.

$$x' = \frac{x - \bar{x}}{\sigma} \quad (4.50)$$

Where x' is the z-score, x is the original data point. While, \bar{x} , σ are the mean and standard deviation, respectively.

- **Mean Normalization**

In this scale, the values are distributed from -1 to +1 (as seen in 4.51), with the mean equal to zero.

$$x' = \frac{x - \bar{x}}{\max(x) - \min(x)} \quad (4.51)$$

The algorithms that re-centre the features vectors at zero coordinate such as the PCA approach, the standardisation and mean normalization can be efficient in rescaling several features on the same range of the magnitude.

4.3.4.2 Principal Components Analysis

The Principal Components Analysis (PCA) is commonly used in the dimensionality reduction of dataset to extract the main representative features (Yuan, 2016). The PCA can be considered as an efficient solution when the correlation between data vectors is strong, which means high redundancy. The main task in this section is to calculate the PCs that can be used instead of the initial features - the following steps illustrate the calculation of the PCs

- **Dataset re-centre**

By subtracting the mean from the corresponding data vector to re-centre the data set, the new mean will be at zero for the adjusted data X (Milovanovic & Popovic, 2012) as:

$$X = \begin{pmatrix} k_1 & \dots & k_M \\ \dots & \dots & \dots \\ k_N & \dots & k_{NM} \end{pmatrix}_{N \times M} \quad \bar{X} = (\text{zero})_{M \times 1} \quad (4.52)$$

Where N is the total number of point's data, M is the number of variables (features). The mean of adjusted data \bar{X} is centred at zero.

- **Covariance matrix**

The variance-covariance matrix C (Konstorum, 2018) can be calculated as:

$$C = \frac{1}{N-1} (X - \bar{X}')'(X - \bar{X}') \quad (4.53)$$

Where, the \bar{X} is equal to zero, then the matrix C can be simplified into:

$$C = \frac{1}{N-1} X'X \quad (4.54)$$

The diagonal elements of the matrix C represent the variance of the attributes, and the off-diagonal elements of the covariance.

- **Eigenvalue and eigenvector**

Eigenvalue λ_i determines the radius of the ellipse, so the peak of the eigenvalue represents the longer radius of the ellipse that is formed from the data distribution.

$$\det (C - \lambda I) = 0$$

$$\det \begin{pmatrix} c_1 - \lambda & c_m \\ c_2 & c_{mm} - \lambda \end{pmatrix} = 0 \quad (4.55)$$

Where, I is the identity matrix, λ represents the eigenvalues, which can be solved as a quadratic equation.

The eigenvector e_i can be calculated by multiplying the covariance matrix C to each eigenvector i as:

$$C e_i = \lambda_i e_i$$

$$\text{If } i = 1 \quad \begin{pmatrix} c_1 & c_m \\ c_2 & c_{mm} \end{pmatrix} \begin{pmatrix} e_{11} \\ e_{12} \end{pmatrix} = \lambda_1 \begin{pmatrix} e_{11} \\ e_{12} \end{pmatrix} \quad (4.56)$$

$$\text{when } i = 2 \quad \begin{pmatrix} c_1 & c_m \\ c_2 & c_{mm} \end{pmatrix} \begin{pmatrix} e_{21} \\ e_{22} \end{pmatrix} = \lambda_2 \begin{pmatrix} e_{21} \\ e_{22} \end{pmatrix} \quad (4.57)$$

The first principal component $PC1$ comprises the most variance of data that can be captured by a linear combination of the attributes; while the second principal $PC2$ component comprises the most variance of data that can be captured by a linear combination of the attributes after the first principal component.

$$PC1 = \begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix} \quad PC2 = \begin{bmatrix} e_{21} \\ e_{22} \end{bmatrix} \quad PCi = \begin{bmatrix} e_{i1} \\ e_{ii} \end{bmatrix} \quad (4.58)$$

4.3.5 Gait Data Classification

The terminology of classification in machine learning is supervised learning implementation using a training set to learn the system with labelled observations available; whilst unsupervised learning acts are based on the measurements of distance similarity or dissimilarity between the observations to calculate the relevant group, which is also called a clustering method (Duda et al., 2001) (Guerra et al., 2011). However, this research focusses only on supervised learning algorithms.

4.3.5.1 Supervised classification

Different supervised classification algorithms are used in this study. In addition, the performance of these algorithms is compared to a proposed classification method known as a Convolutional Encoder (CE). The purpose of the assessment is to determine which algorithms are most effective for data classification of changes in gait pattern.

In this technique, the data vector will be predicted at each instance of (i) as a vector is denoted by ($x^{(i)}, l^{(i)}$), with $i \in (1 \dots N)$, where $x^{(i)}$ is represented as the value of the variable or feature and $l^{(i)}$ represents the order of the labels, where labels $\in (1, \dots, R_o)$ are of the classes l . The main task of the classifier is to create a model automatically for the training data stage that is based on a data subset. Throughout the testing data stage, the model must understand the new instances that are unknown labels using only the variable values. Assuming that x is a new instance, then supervised classification will build a deduced function of γ (Guerra et al., 2011) as:

$$\gamma: x \rightarrow \{1, \dots, R_o\} \quad (4.59)$$

Consequently, the deduced function predicts where the unseen data belongs to and the appropriate membership class in the testing stage. Cortes & Vapnik (1995) confirm that the training data stage learns the algorithm to estimate unobserved sample classes in a suitable pattern. Therefore, several kinds of supervised classifiers are used in this research to classify and detect the changes in gait pattern. Specifically, SVM, k-NN, DT and discriminant classifiers are all involved in the data classification stag.

4.3.5.2 Convolutional Encoder

This CE technique acts based on a binary format (i.e. logic 0 & logic 1) as input to encode data at the output as a codeword. For this purpose, all features will be converted into a binary system first.

- **Decimal to Binary Conversion**

To achieve a decimal feature to binary conversion, let assume the decimal feature is $XX.YY$ where XX is an integer part and YY is a fraction part, the equivalent binary number will be computed in two steps.

For the integer part of a decimal number, the binary number will be obtained by dividing the XX by 2 and this process will be repeated for the quotients till, where one quotient equals zero as follows:

$$\frac{XX}{2} : \text{if the quotient} = X, \text{and the remainder} = 1$$

$$\frac{X}{2} : \text{if the quotient} = 2, \text{and the remainder} = 1$$

$$\frac{2}{2} : \text{the quotient will} = 1, \text{and remainder} = 0$$

$$\frac{1}{2} : \text{the quotient will} = 0, \text{and remainder} = 1$$

Therefore, $(XX)_{10} = (1011)_2$ is generated from the remainders' results.

For the fraction part of a decimal number, the binary number will be obtained by multiplying the $0.YY$ by 2 and this process will be repeated, so that the first one result equals "1" as follows:

$$0.YY \times 2 : \text{if the result} = 0.YX, \text{ the integer part is } 0$$

$$0.YX \times 2 : \text{if the result} = 0.XY, \text{ the integer part is } 0$$

$$0.XY \times 2 : \text{if the result} = 1.YX, \text{ the integer part is } 1$$

$$0.YX \times 2 : \text{if the result} = 1.00, \text{ the integer part is } 1$$

Therefore, $(0.YY)_{10} = (0011)_2$ is generated from the remainders' results.

Finally, both parts are combined $(XX.YY)_{10} = (1011.0011)_2$.

The Convolutional Encoder is commonly used with a Viterbi decoder in communication links to detect and correct the bits that are received in error. The maximum likelihood decoding is used as a metric to calculate the survival path (Liu, 2004). This technique is proposed to be used for gait pattern classification that is based on Hamming Distance.

A Convolutional Encoder (CE) is specified by three parameters $(n\ k\ L)$, where k and n are the numbers of input and output bits, respectively. L is the number of the memory registers (s_i), which is initially set to zero (Liu, 2004). The encoder includes two of the modulo-2 adders which produce two bits at the output of the CE for each bit at the input, which means the code rate $\left(\frac{n}{k}\right)$ equals $(\frac{1}{2})$, as can be shown in Figure 4.12 below.

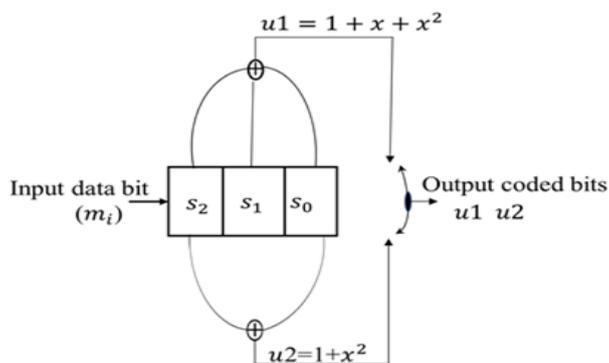


Figure 4.12: The (2 1 3) Convolutional Encoder with Code Rate 1/2

- **Polynomial vector generation**

The polynomial vector is generated for each branch of the encoder, one vector for each modulo-2 adder. The i^{th} element in each vector is “1” if the i^{th} stage in the shift register is connected to the corresponding modulo-2 adder, and otherwise “0” as in this case; $G_1 = (1,1,1)$, $G_2 = (1,0,1)$ for the upper and lower branches, respectively. Consequently, the polynomial vectors are generated as (4.60) and (4.61).

$$G_1 = 1 + x + x^2 \tag{4-60}$$

$$G_2 = 1 + x^2 \tag{4-61}$$

An input bit m_i is applied into the leftmost register. The generator polynomials and the existing values in the remaining registers will form the encoder outputs. All register values shift to the right (m_1 moves to m_0 , m_0 moves to m_{-1}), the encoder continues shifting until all

registers have returned to zero. Therefore, the output sequences are calculated as: u_1, u_2 symbol.

$$u_1^i = m_i \cdot G_1 \tag{4.62}$$

$$u_2^{i+1} = m_i \cdot G_2 \tag{4.63}$$

Where i is the order of the input and output bits. Finally, the output codewords of the encoder are formed as:

$$U = u_1^1 u_2^2, u_1^3 u_2^4, \dots \dots u_1^{N-1} u_2^N \tag{4.64}$$

Where N is the length of the output codeword. This means that each bit at the encoder input will generate two bits at the encoder output.

The extracted gait features are converted into binary format to agree that the encoder is based on “0”, “1” bits. The CE produces four possible outputs (00, 10, 01 and 11 states) dependent on the input bit. Figure 4.13 illustrates that each state (i.e. square box) has just two possible outputs: dashed line when input bit is “1” and solid line if the input bit is “0”. Meanwhile, the output of the CE is represented on the path of the transition as symbol (two bits together) (Liu, 2004). To understand the principle work of the CE, a Trellis diagram is included in the next step.

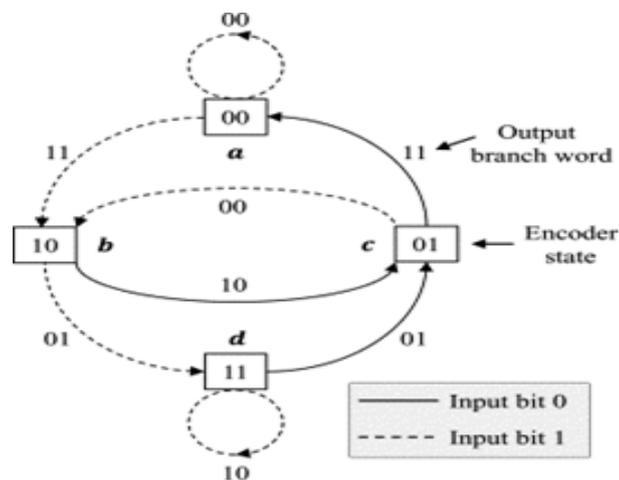


Figure 4.13: The State Diagram of the 1/2 Rate Convolutional Encoder

- **Trellis Diagram Representation**

The Trellis diagram represents the principal work of the CE technique, with the following example demonstrating the encoded data process for input bits as [1 0 0 1]. The

transition between states has just two possible paths which depend on the input bit - if it is “0” the transition will be via a solid line, otherwise it is “1”, which makes the transition via a dashed line; the same structures are repeated for all input of data bits. The path of the transitions represents the encoded bits at the output of the CE; while the labelled bits on the branches of the complete path forms are the output codewords (as illustrated in Figure.4.14 below) highlighted in red.

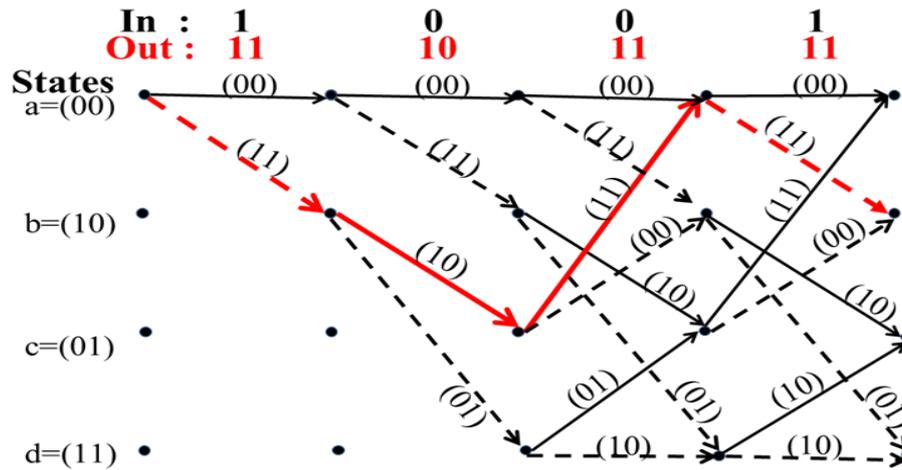


Figure 4.14: Trellis Diagram Example to Encode Input Bits [1001]

All extracted gait features are encoded by generating the codewords of the path as explained in the previous Trellis diagram example. The classification decision is dependent on the threshold of the CE encoder, which is also built using the same procedures as the Trellis diagram. Moreover, the classification decision is calculated using the Hamming Distance (HD) for the codewords of the gait features and threshold. For instance - if HD equals “1” this means that the bits differ, so in this instance the position of the bits will be considered (see 4.65); otherwise “0” means the bits are similar which yields the formulae of Error equals zero.

$$Error = \sum_{p=1}^N HD_p \times 10^{-p} \quad (4.65)$$

Where HD is the Hamming distance, and p means the order of the sequence bits of the codewords. The HD method is used to calculate the *Error level* for each feature from the threshold. The smallest error means a high similarity to that class of walk speed, otherwise the high error means low similarity and then low probability of belonging to that class.

- **Similarity Metric**

In this section, the Hamming Distance (HD) and the position (p) of the bit that differs from the threshold are combined as (HD_p) to calculate the error rate using the Equation (4.65).

The low rate of error means the decision is highly likely to be a similarity decision (i.e. it belongs to a class), while the high rate of error means the decision is highly likely to not be a similarity decision (i.e. it does not belong to a class).

To understand how the similarity metric works, the obtained codeword C1 from the previous section (i.e. C1: 11 10 11 11) could be known as the tested feature C1, with the similarity decision dependent on the principal work of the Viterbi Decoder (VD). The C1 is evaluated from three thresholds known as T1: 11 10 11 00, T2: 11 10 00 10 and T3: 11 01 01 00 instantaneously, (as can be seen in Figures 4.15, 4.16 and 4.17). First, the Hamming Distance is calculated for C1 from each threshold (see below HD1, HD2 and HD3), whilst considering that the order of the bit's position has a HD equal to one. Finally, the error equation (4.66) for a class decision is used, where the lowest error rate for the tested feature C1 means that the C1 belongs to the that class.

HD1:00 00 00 11, HD2:00 00 11 01, HD3: 00 11 10 11.

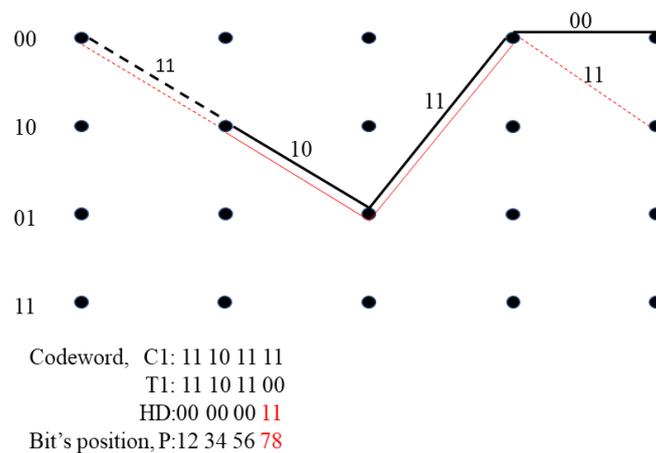


Figure 4.15: Path Metric using Threshold T1

$$Error1 = \sum_{p=1}^N HD1_p \times 10^{-p} \quad , N=8. \quad (4.66)$$

$$Error1 = (0 \times 10^{-1}) + (0 \times 10^{-2}) + (0 \times 10^{-3}) + (0 \times 10^{-4}) + (0 \times 10^{-5}) + (0 \times 10^{-6}) + (1 \times 10^{-7}) + (1 \times 10^{-8})$$

$$Error1 = 1.1 \times 10^{-7}$$

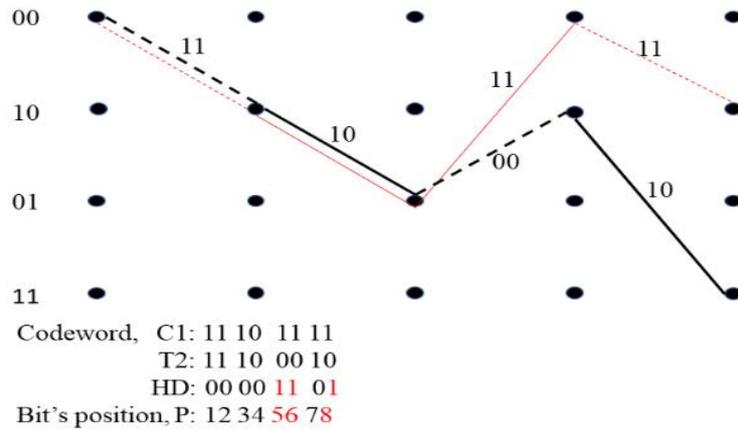


Figure 4.16: Path Metric using Threshold T2

$$Error2 = (0 \times 10^{-1}) + (0 \times 10^{-2}) + (0 \times 10^{-3}) + (0 \times 10^{-4}) + (1 \times 10^{-5}) + (1 \times 10^{-6}) + (0 \times 10^{-7}) + (1 \times 10^{-8})$$

$$Error2 = 1.1 \times 10^{-5}$$

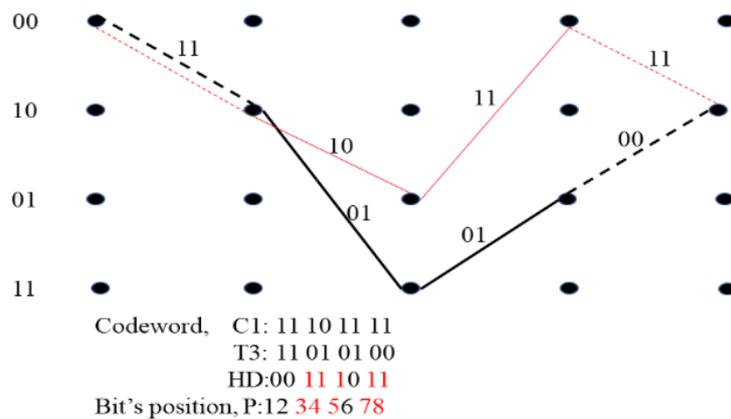


Figure 4.17: Path Metric using Threshold T3

$$Error3 = (0 \times 10^{-1}) + (0 \times 10^{-2}) + (1 \times 10^{-3}) + (1 \times 10^{-4}) + (1 \times 10^{-5}) + (0 \times 10^{-6}) + (1 \times 10^{-7}) + (1 \times 10^{-8})$$

$$Error3 = 1.1 \times 10^{-3}$$

The Error1 is the lowest value when compared to Error2 and Error3, which means the sample C1 is high likely to belong to class one.

4.3.6 K-Fold Cross Validation

the available data D is partitioned into k parts, which are called folds as $d=1, 2, \dots, k$, where all folds have the same size. This makes the k -folds involved in all samples equal to the original dataset D . The learning algorithm is trained by using $(D - d_i)$ which is known as a training set, while one-fold (d_i) is expected to be the testing set and this process is repeated k times for $t \in (1, 2, \dots, k)$ as shown in Figure 4.18.

In other words, the dataset D is managed as training and testing methods. Meanwhile, in each fold $d= 1, 2, \dots, k$, the model is fitted with the training set to be used in the prediction of the observation's responses for the testing set. The estimation prediction error is calculated from MSE_i for each fold d_i (Hastie et al., 2013):

$$MSE_i = (y_i - \hat{y}_i)^2 \tag{4.67}$$

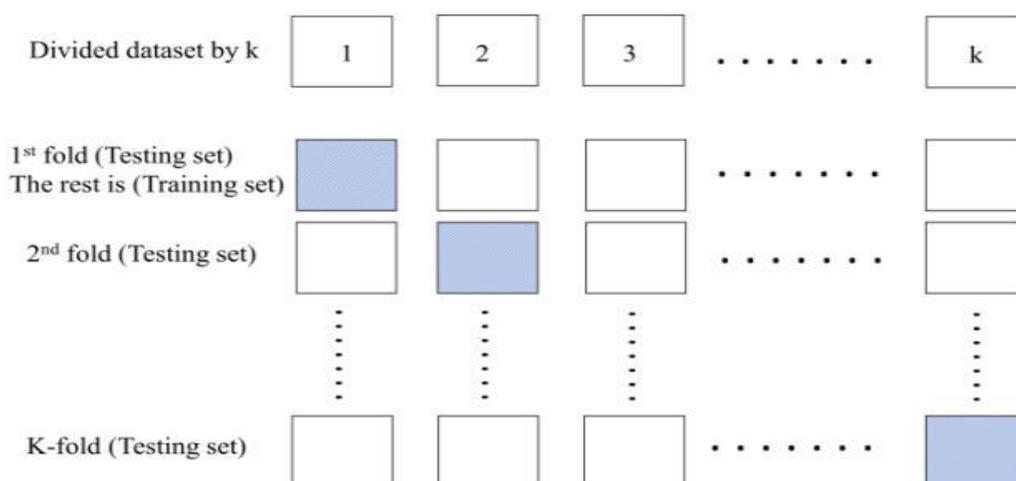


Figure 4.18: K-Fold Cross-Validation Method

Where MSE is the mean square error and y and \hat{y} are the training observation and prediction estimation, respectively. This gives the cross-validation error as an average of the estimation prediction errors for all folds (Kohavi, 1995):

$$CV_k = \frac{1}{k} \sum_{i=1}^k MSE_i \tag{4.68}$$

The main advantage of the k -fold CV is that it avoids the randomness of splitting the training/testing sets, where all the datasets can be used in the training/testing sets with no overlapping (Hastie et al., 2013). Typically, the use of $k = 5$ or 10 is commonly performed as

a rule and used as empirical evidence. Hence, the trade-off is alternated between the computational costs and the bias-corrections of the k-fold CV (Fushiki, 2011).

4.3.7 System Evaluation

4.3.7.1 Goodness-of-Fit Statistics

- **Sum of Squares Due to Error**

The SSE is a statistic measurement for total deviation from the fit to the response values. It is also known as the summed square of residuals and it can be defined as:

$$SSE = \sum_{i=1}^n \{w_i(y_i - f_i)^2\} \quad (4.69)$$

Where y_i and f_i are the observed data and the predicted value from the fit, respectively; while w_i is the data point weight, usually $w_i = 1$. The closest value of the SSE to zero, indicates that the model behaves in a small random error, and the fit model is efficient in prediction.

- **R-Square**

The R-square is a statistic measurement that evaluates the fitting performance by showing the data variation. In other words, the R-square is the square of the correlation between the response values and the predicted response values. This measurement uses the values of between zero and one, and a greater rate of variation can be obtained when the R-square indicates as closer to one - this can be defined as:

$$R_{square} = 1 - \frac{[\sum_{i=1}^n \{w_i(y_i - f_i)^2\}]}{\sum_{i=1}^n \{w_i(y_i - y_{av})^2\}} = 1 - \frac{SSE}{SST} \quad (4.70)$$

Where f_i and y_{av} are the predicted values from the fit, and the average value of the observed data; while SSE and SST are the sum of squares due to error and the total sum of squares, respectively.

- **Root Mean Squared Error**

The RMSE is a statistical measurement for the fit standard error and the standard error of regression. It is an estimate of the standard deviation of the random components in the data, and is defined as:

$$MSER = \sqrt{MSE} \quad (4.71)$$

Where MSE is the mean square error or the residual mean square. The prediction can be useful when the MSE value indicates closer to zero. The MSE can be calculated as:

$$MSE = SSE/v \quad (4.72)$$

Where v is the residual degree of freedom, which is calculated by subtracting the fitted coefficients m from the response values n numbers as ($v = n - m$).

4.3.7.2 Classification Evaluation

The performance of the classifier can be evaluated in many ways. In supervised learning with three possible classes, the first task is to calculate four ratios including: True Positives TP, False Positives FP, True Negatives TN and False Negatives FN. These quantities are essential for evaluating the classifiers' performance. A Positive True PT can be extracted from the diagonal of the confusion matrix (as can be seen in Table 4-2). However, the rest of the ratios are calculated for each class as the following:

FN for a class is the sum of values in the corresponding row except for the (TP).

FP for a class is the sum of values in the corresponding column except for the (TP).

TN for a class is the sum of values from the confusion matrix except for a column and row of a class.

Table 4-1: The Confusion Matrix for Three Classes

		Predicted		
		A	B	C
Actual	A	TPa	Eab	Eac
	B	Eba	TPb	Ebc
	C	Eca	Ecb	TPc

The elements of confusion matrix constitute vital information to compute the most metrics that are effective in the understanding of classification performance (see Table B.1 in Appendix B. for more details). Some of the evaluative metrics of data classification can be calculated as the following:

- **Classification Accuracy**

The accuracy of classification is an important scale to evaluate the ability of the classifier for predicting the class correctly. This can be calculated directly from the confusion matrix by dividing the summation of the values of the diagonal matrix over the summation of the values of all elements.

$$Accuracy = \frac{(TPa+TPb+TPc)}{(TPa+Eab+Eac+TPb+Eba+Ebc+TPc+Eca+Ecb)} \quad (4.73)$$

- **Precision**

This measure indicates the level of a model's exactness, where a high level of a classifier's precision means a perfect classifier. This scale can be calculated for each class as:

$$Precision A = \frac{TPa}{(TPa+Eba+Eca)}$$

$$Precision B = \frac{TPb}{(TPb+Eab+Ecb)} \quad (4.74)$$

$$Precision C = \frac{TPc}{(TPc+Eac+Ebc)}$$

- **Sensitivity**

A sensitivity measurement is the rate of true positive or the recall. It is used to measure the rate of actual positives that are predicted correctly. The following equations are used for the sensitivity measurement of each class:

$$Sensitivity A = \frac{TPa}{(TPa+Eab+Eac)}$$

$$Sensitivity B = \frac{TPb}{(TPb+Eba+Ebc)} \quad (4.75)$$

$$Sensitivity C = \frac{TPc}{(TPc+Eca+Ecb)}$$

- **Specificity**

A specificity measurement is the rate of true negative. It is used to measure the rate of actual negatives that are predicted correctly. The following equations are the specificity measures for each class:

$$Specificity A = \frac{(TPb+Ebc+Ecb+TPc)}{(TPb+Ebc+Eba+Ecb+TPc+Eca)}$$

$$\text{Specificity } B = \frac{(TPa+Eac+Eca+TPc)}{(TPa+Eac+Eab+Eca+TPc+Ecb)} \quad (4.76)$$

$$\text{Specificity } C = \frac{(TPa+Eab+Eba+TPb)}{(TPa+Eab+Eac+Eba+TPb+Ebc)}$$

- **F-measure**

This measurement balances between precision and sensitivity rates; when either rate is zero quantity then the measure will equal zero. However, high performance is when the data balances this measure.

$$F_Measure = \frac{2 \times \text{sensitivity} \times \text{precision}}{\text{sensitivity} + \text{precision}} \quad (4.77)$$

- **ROC Curve**

The ROC curve and the AUC are used to assess the data classification approaches, where the ROC plot is based on a True Positive Rate (sensitivity) and a False Positive Rate (1-specificity), in which a large area under the curve indicates that the classifier is more efficient.

4.4 Summary

Gait pattern changes can be addressed using several algorithms when dealing with gait data collected during a walk test using a Kinect camera. Both algorithms and machine learning are involved to estimate and understand the behaviour of the gait features and define some conditions.

The Microsoft Kinect v2 is adopted to track and collect the relevant data because it can meet some needs of elderly people who prefer to live independently, including markers-less, quick installation and no video data. However, such a device has a low data rate which leads to a decrease in the accuracy of the measurements - thus the quality of the extracted features could be affected. Furthermore, in the data classification stage, the classification performance is greatly affected by the boundary between different classes, which is called a ‘decision boundary’. This raises other questions, such as ‘how to weigh the features from the class labels’ and ‘which kind of similarity metric is suitable for the dataset’.

Subsequently, although the skeleton data is collected using a low data rate device in this research, the collected data is mapped into a higher frequency spectrum using an AM technique. This increases the sampling frequency rate of an AM-modified gait signal.

Consequently, the extracted gait features from the AM-modified signal become more representative of the gait pattern changes. In addition, a CE technique is proposed to be used as a classifier for gait pattern changes. The principal work of the CE is based on the Hamming Distance HD to calculate a similarity rate among gait features from the class threshold. However, the similarity decision does not only consider the number of bits that differ (i.e. HD=1), but also the positions of the bits that differ from the threshold, as this can increase the classification accuracy.

5. EXPERIMENTAL RESULTS OF IMPLEMENTED SYSTEM AND DISCUSSION

The focus of this chapter is on a comparison between the six smoothing data approaches, the gait feature extraction approaches, and the classification solutions for gait pattern changes. The experimental results are provided and followed by a relevant discussion of the findings.

5.1 Introduction

The main task is to use the 3D skeleton data of lower body joints that was collected during a walk test, by using a Kinect camera for the extraction of the most representative gait features, which were then classified based on gait speed. To achieve these requirements, the methods and algorithms previously discussed in chapter four, will be involved. In addition, a set number of walk trials were conducted to address the main objectives of the research. MATLAB programming language was used to analyse the collected skeleton data and run the gait analysis system through methods and algorithms that can yield the results, thus enabling an understanding of the gait pattern changes via some representative features and machine learner approaches.

A definition of the proposed system AM/CE is introduced in section 5.2, while the rest of the sections are presented according to the experiments that have been conducted. Section 5.3 explains the selection of smoothing data techniques to smooth and reduce the level of error from the collected skeleton data, while the Kinect validation to gold standard system is explained in section 5.4. The use of Kinect for timed walk tests in real time is shown in section 5.5. Whilst gait feature extraction using AM/Spatiotemporal analysis and AM/FM techniques were explained in sections 5.6 and 5.7 respectively. In section 5.8 and 5.9, the implementation and performance of AM/CE system were introduced, respectively. Finally, a summary of this chapter is given in section 5.10.

5.2 The Proposed System AM/CE

The implementation of an AM/CE system plays a significant part in fulfilling the research objectives, where six stage are involved (see Figure 1. In appendix C) for addressing

the performance improvement in gait speed classification. The proposed system combines two main parts; the former, AM technique and is used to extract new gait features that can be efficient and representative in gait pattern changes. While, the latter, CE technique, which uses the extracted gait features for gait speed classification. This system can perform tasks without the need to determine the gait cycle phases, thus reducing the computational cost. Moreover, the limitation of low data rate of Kinect data is solved by transforming the gait length signals into a higher frequency spectrum. This has been achieved by using Amplitude Modulation (AM) which modifies the gait signal to be in the AM domain.

The research aim has been addressed through the improvement of the classification accuracy for gait pattern changes, where the low data rate of collected data was the main challenge. This improvement was investigated by comparing the classification performance of several supervised classifiers to the AM/CE system. The obtained results show that the proposed method is more efficient than a set number of supervised classifiers in gait speed classification.

5.3 Experiment 1- Skeleton Data Smoothing Techniques

The objective of this experiment was to determine a smoothing approach that would be efficient in reducing the level of noise from 3D skeleton data, whilst guaranteeing that the original data was preserved.

To achieve this task, a set number of smoothing approaches were evaluated based on two conditions; the time delay and fast response between the input and output data of the filter. Commonly used filters in skeleton data smoothing such as the Moving Average (MA), Exponential Moving (EMA), Savitzky-Golay (GS), Local Regression (LR), Median (M), and Kalman filters were used to filter and smooth the collected data. However, although an increase in smoothing performance can be obtained by the adjustment of the filters' parameters, this could also lead to latency.

5.3.1 Original data

The positional data movement of the right ankle was selected to be smoothed using several filter techniques. The raw data of the right ankle was considered as noisy because the right ankle was occluded by the left leg of the body during the walk process. Figure 5.1 below illustrates the original data signal of the right ankle that contains some noise (spikes). The task

was to reduce the spikes (noise) where possible without distortion of the gait information, that is contented on the displacement curve of the original data, with the main information being the movement measured between the two strike heels known as the gait cycle period or gait stride length.

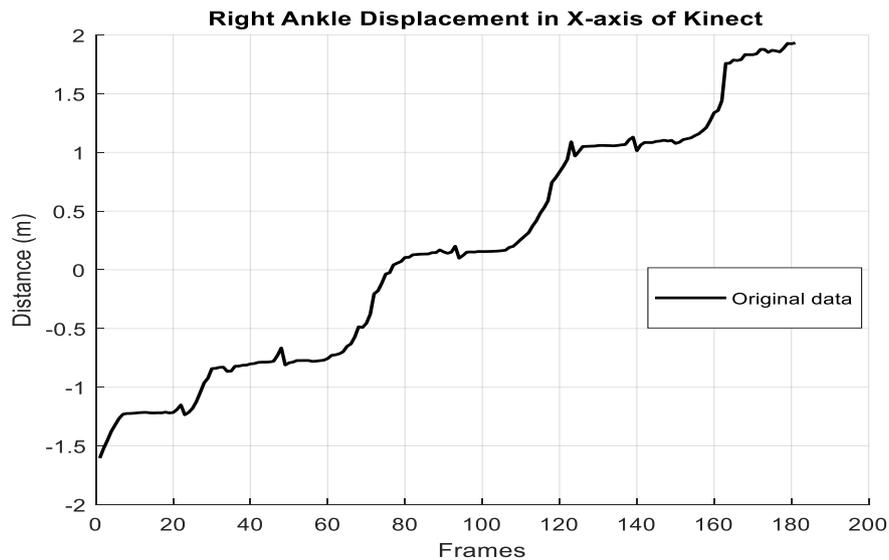


Figure 5.1: Right Ankle Movement Signal, which is Affected by Noise (Spikes), the Signal Shows Strike Heel and Toe-Off on the Curve that Corresponds with the Concave Down and Concave Up, Respectively

5.3.2 Filtered data

Several kinds of filters were applied to reduce the noise and smooth the collected data as can be seen in Figure 5.2. When comparing the smoothness performance among the output of the smoothing techniques, a trade-off between the smoothness and latency was considered. Moreover, the curve's parts that represented the displacement changes, which indicate the starting and ending of gait cycle, could be smoothed where possible without distortion of the original data. This is particularly true for the filtered data curve, where the concave up and concave down represent the toe-off and strike heel information, respectively. Firstly, all filters were tuned to remove or reduce the spike noise from the original signal of joint movement, then all the adjusted filters' parameters were kept at the same values to be used in the second step of the assessment. Notably, the processes of spike elimination for each filter affects the latency at different levels according to the type of filtering approach used, as can be seen in Figures 5.3-5.8.

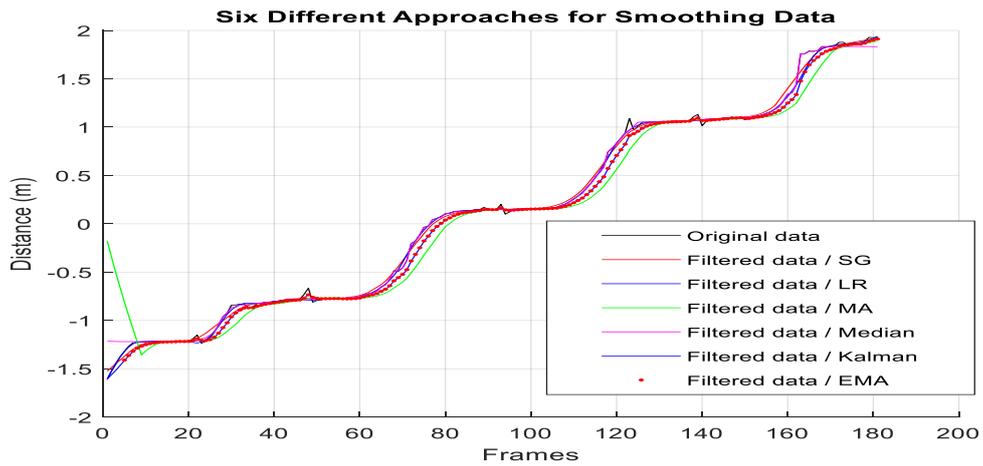


Figure 5.2: Filtering and Smoothing of Right Ankle Movement Signal using Six Different Filter Techniques; Original Data (black line), SG Filter (pink line), RLOESS Filter (blue line), MA Filter (green line), Median Filter (purple line), Kalman Filter (blue line) and Exponential MA (red dot line)

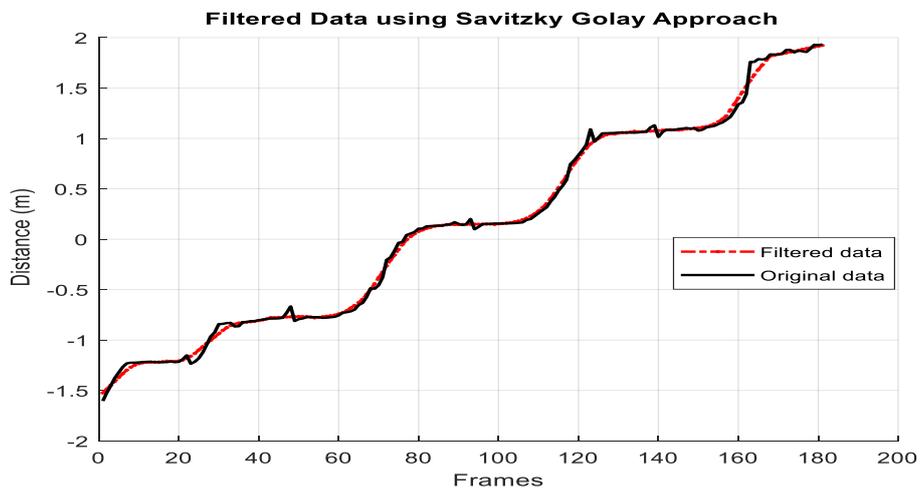


Figure 5.3: The Output Curve of the Savitzky-Golay Filter, Input Curve is a Black Line, Output Curve is a Red Line.

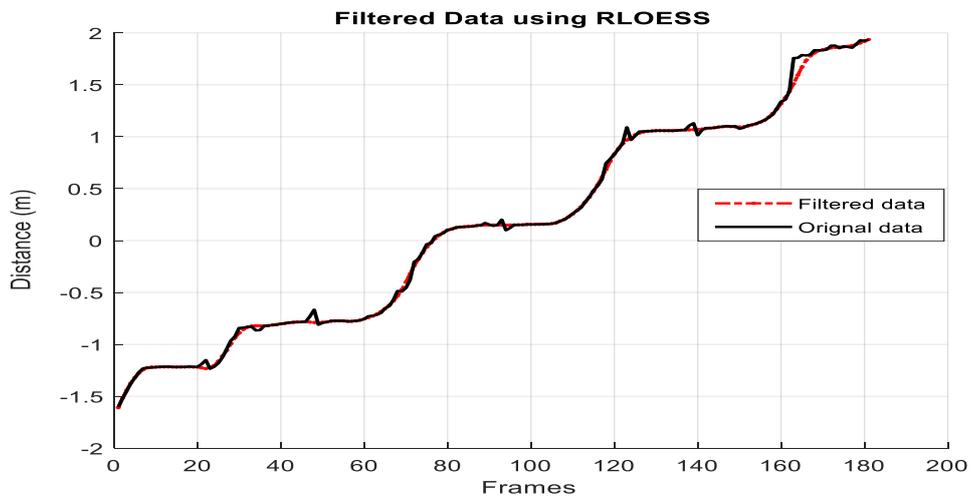


Figure 5.4: The Output Curve of Local Regression Filter (RLOESS), Input Curve is a Black Line, Output Curve is a Red Line.

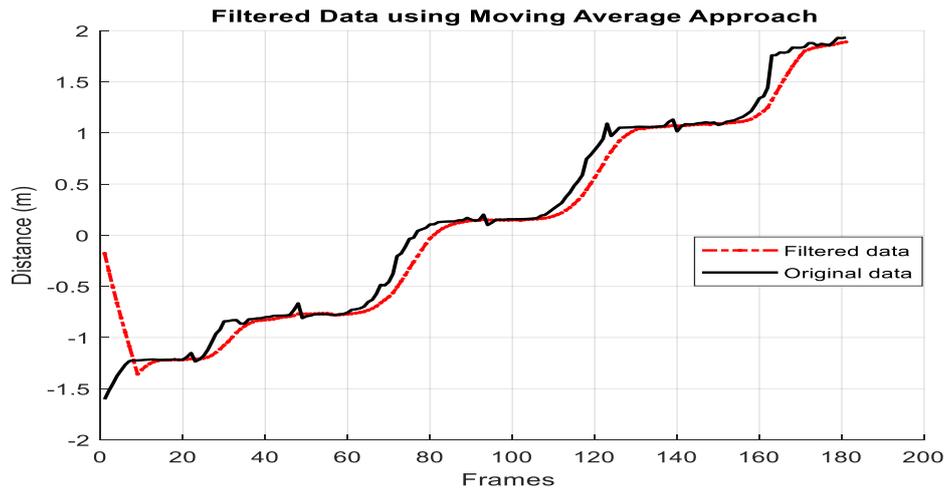


Figure 5.5: The Output Curve of Moving Average Filter (MA) for Right Ankle Movement, Input Curve is a Black Line, Output Curve is a Red Line.

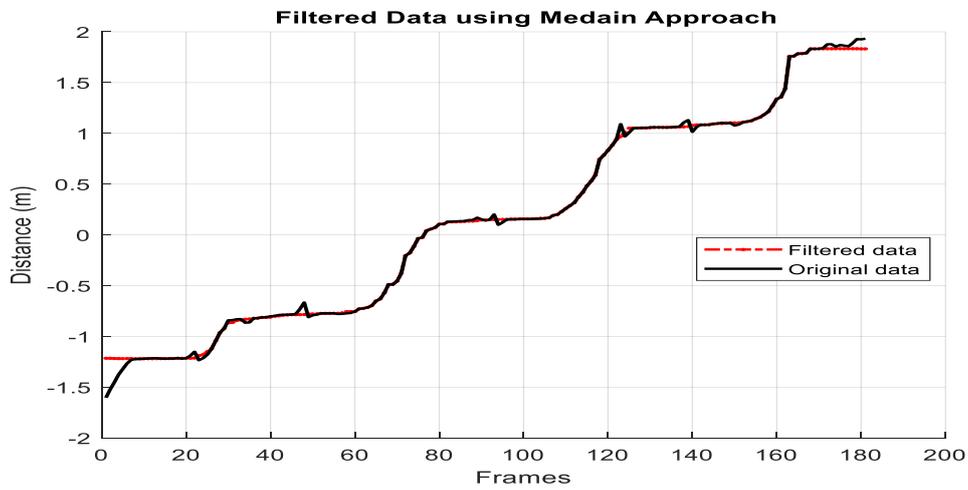


Figure 5.6: The Output Curve of Median Filter for Right Ankle Movement, Input Curve is a Black Line, Output Curve is a Red Line.

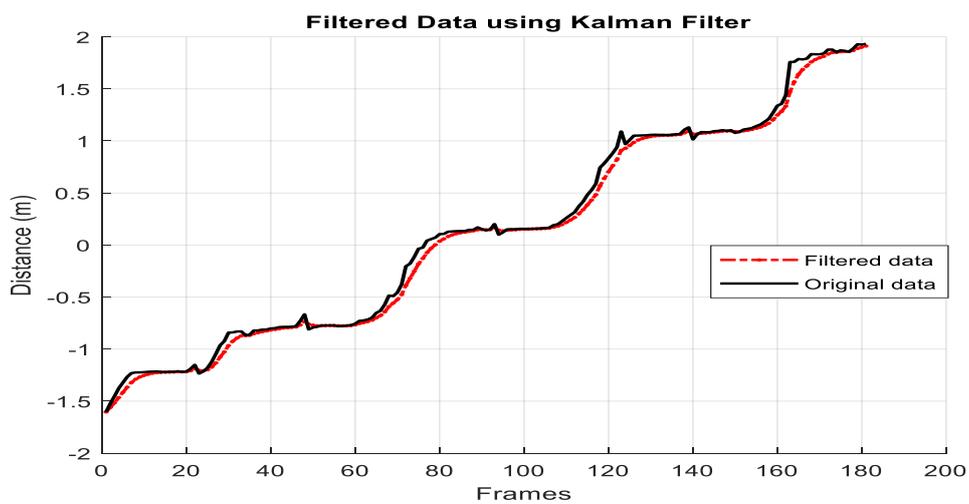


Figure 5.7: The Output Curve of Kalman Filter for Right Ankle Movement, Input Curve is a Black Line, Output Curve is a Red Line.

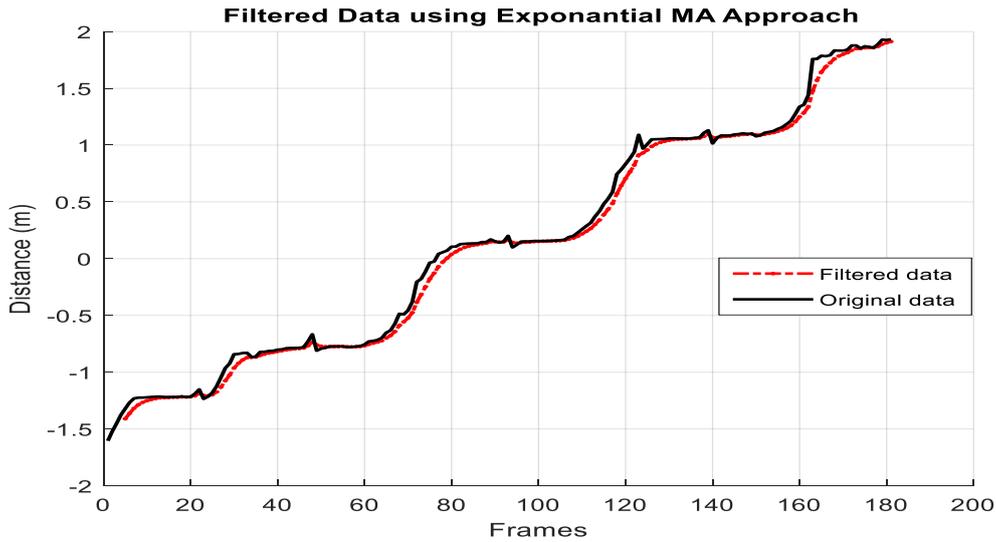


Figure 5.8: The Output Curve of Exponential MA Filter for Right Ankle Movement, Input Curve is a Black Line, Output Curve is a Red Line.

The simple correlation coefficient (r) was calculated for the original data (input signal) and filtered data (output signal). The obtained results show different measurements according to the filtering process used. Table 5.1 lists the values of correlation (r) for each filter.

Table 5-1: The Simple Correlation Coefficient for Filtered Data to Original Data

The Simple Correlation Coefficient (r) between Original Data and Filtered Data					
Savitsky-Golay Filter	Local Regression Filter	Moving Average Filter	Median Filter	Kalman Filter	Exponential MA Filter
0.9993	0.9995	0.9803	0.9989	0.9988	0.9988

From the above table, it is possible to see that there are three filtering techniques (RLOESS, SG and Median filters) which have a lower latency compared to the other filtering approaches. Secondly, the fast rate of the filter's response was considered when evaluating the three filters, which have a low latency from the original data curve. To discriminate between the performance of the Savitzky-Golay, Median and RLOESS filters in the fast response of the output to the input of the filter, a curve fitting was used by applying a polynomial with 1 & 9 degrees that was built-in MATLAB. Figures 5.9, 5.10 and 5.11 illustrate the relation between the filtered data to the original data for the SG, M and LR filters, when the relation was fitted by a first order polynomial model, respectively. While, Figures 5.12, 5.13 and 5.14 illustrate the relation of the filtered data to the original data for the SG, M and LR filters, when the relation was fitted by a ninth order polynomial model, respectively.

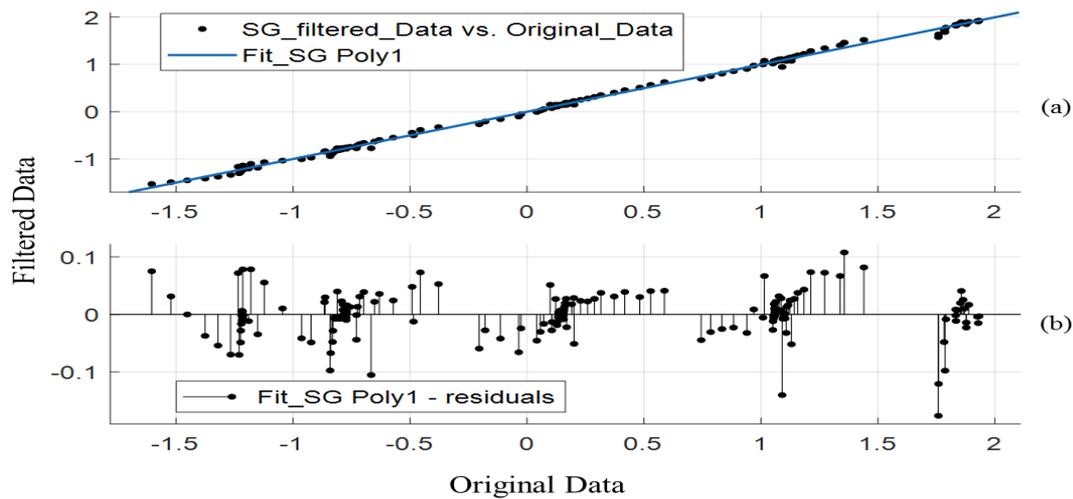


Figure 5.9: The Non-Linear Regression (a) Original Data - Filtered Data Relation in Savitzky-Golay Filter (black dots). While, the Curve of the 1th Degree Polynomial Model is Plotted (blue line) (b) is the Residuals between Two Curves.

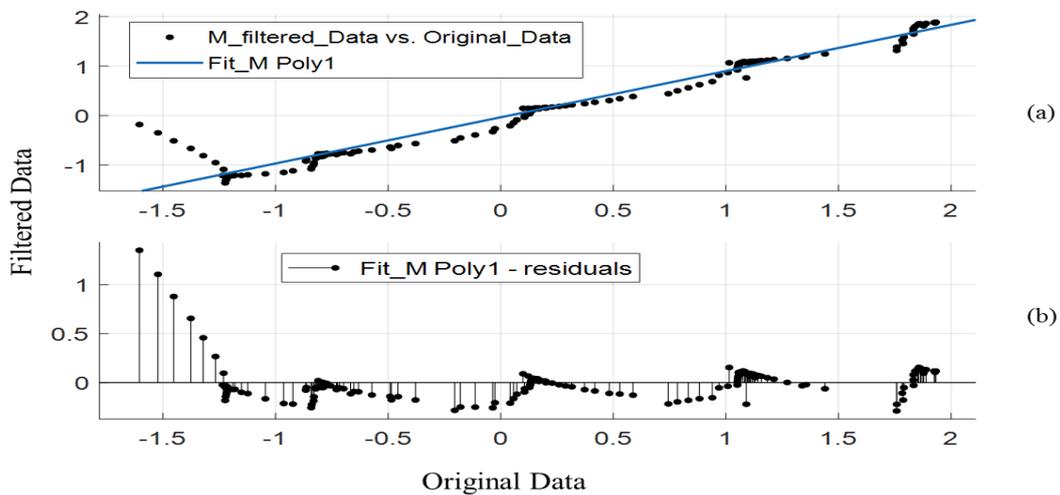


Figure 5.10: The Non-Linear Regression (a) Original Data - Filtered Data Relation in Median Filter (black dots). While, (blue line) is a 1th Degree Polynomial Model, (b) is the Residuals between Two Curves.

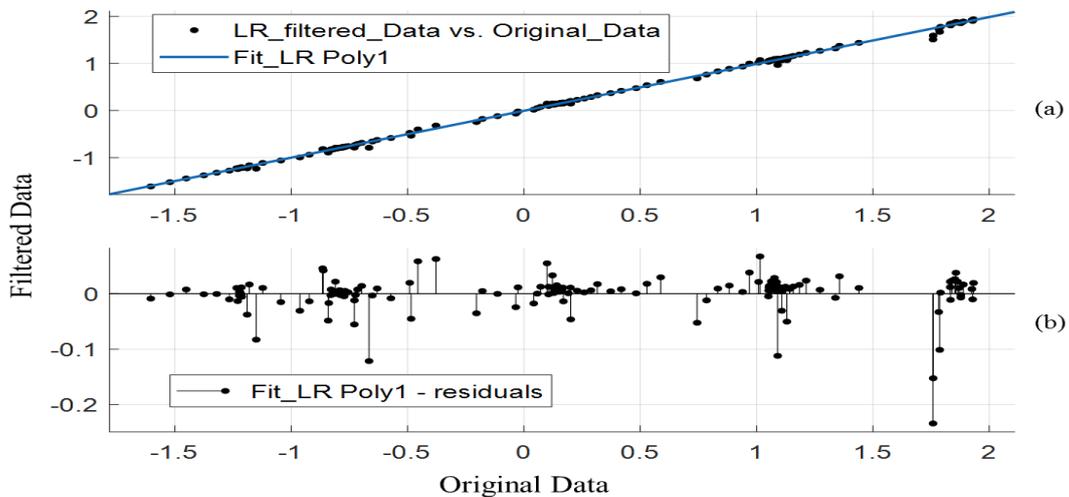


Figure 5.11: The Non-Linear Regression (a) Original Data- Filtered Data Relation in RLOESS Filter (black dots). While, (blue line) is a 1th Degree Polynomial Model, (b) is the Residuals between Two Curves.

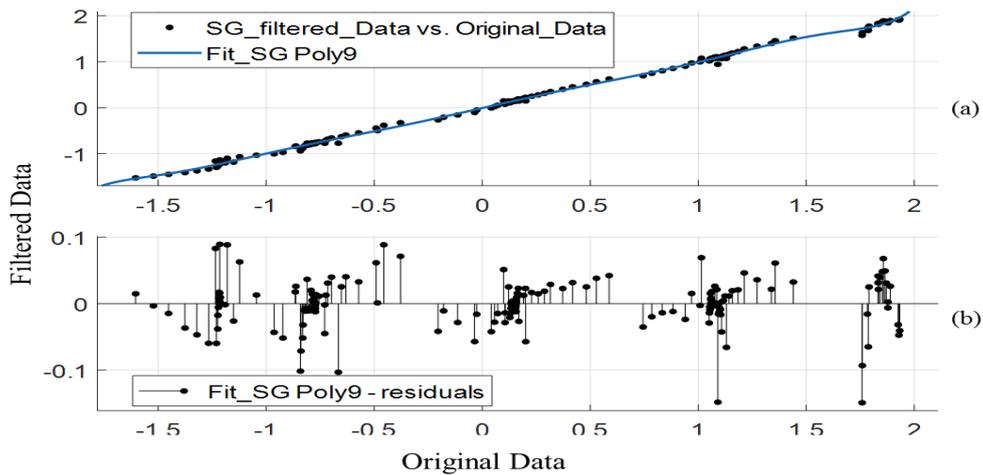


Figure 5.12: The Non-Linear Regression (a) Original Data - Filtered Data Relation in Savitzky-Golay Filter (black dots). While, the Curve of the 9th Degree Polynomial Model is Plotted (blue line) (b) is the Residuals between Two Curves.

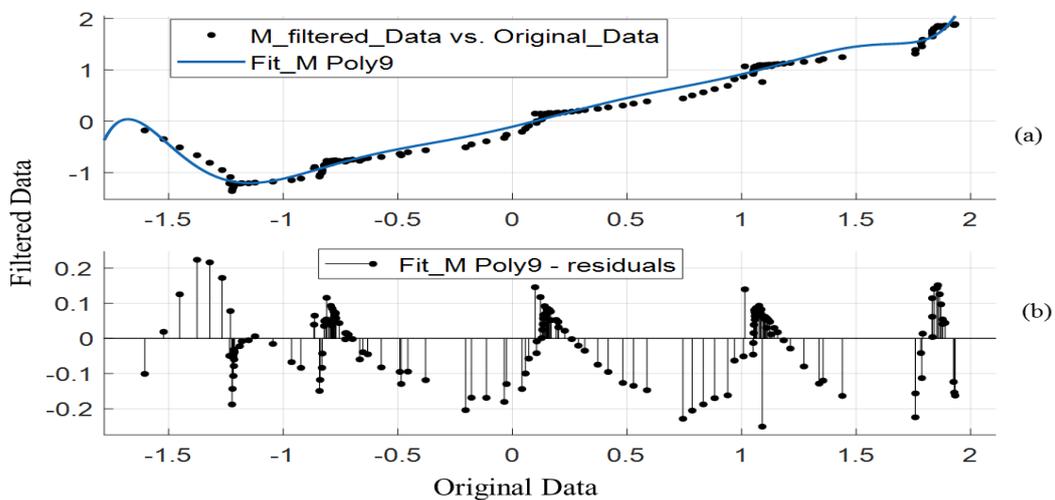


Figure 5.13: The Non-Linear Regression (a) Original Data - Filtered Data Relation in Median Filter (black dots). While, (blue line) is a 9th Degree Polynomial Model, (b) is the Residuals between Two Curves.

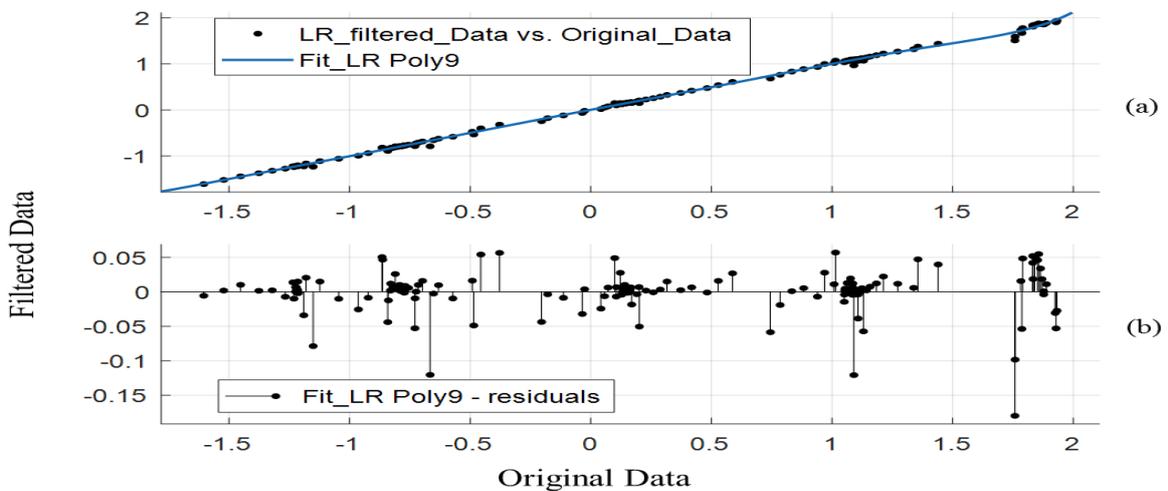


Figure 5.14: The Non-Linear Regression (a) Original Data - Filtered Data Relation in RLOESS Filter (black dots). While, (blue line) is a 9th Degree Polynomial Model, (b) is the Residuals between Two Curves.

In these tests, the filtered data versus the original data curves were fitted to several degrees in a polynomial model to calculate the SSE, R-square and residual measures (see Table 5.2). These metrics could be efficient for assessing which filter can smooth data in fast response to the input data, without destroying the original data.

Table 5-2: Polynomial (9) Curve Fitting to Three Types of Filtered Data; Savitzky-Golay, Local Regression and Median Filters

Approach		SSE	R-square	Max Residual	Min Residual
LR	Poly 1	0.1844	0.999	0.0673	-0.234
	Poly 9	0.1553	0.9992	0.0570	-0.1797
SG	Poly 1	0.2745	0.9985	0.1078	-0.1764
	Poly 9	0.2367	0.9987	0.0891	-0.1478
M	Poly 1	6.527	0.9611	1.353	-0.2902
	Poly 9	1.601	0.9905	0.2238	-0.2506

- **Differentiation of the best smoothing approach among a set of filtering techniques for 3D skeletal positional data, based on less time delay and fast response**

The above experiment was performed to reduce spike noise as possible as. Noisy data was smoothed using different kinds of filters separately. This joint data was selected because of its occlusion noise caused by other body’s parts during a walk in parallel with the camera’s x-axis. The filters were tuned to be able to reduce the noise and smooth the data. During the filtering process, a time delay issue was emerged between the filter input and its response on the output. In Table 5.1 above, the largest latency was observed with the AM filter where the correlation was $r = 0.9803$, followed by both the EAM and Kalman filter with $r = 0.9988$. In addition, the moving average filter cannot track data at the beginning of a curve, due to its principle work based on the averaging of data within the window. Although other filters showed an acceptable delay on the output, some of them still suffered from different issues such as poor smoothness on the curve turn up (slow response) as was the case for the median filter. Moreover, the output of the median filter showed very poor tracking data at the beginning and end of the curve.

Generally, the RLOESS and SG filters showed the best smoothness degree at the output with acceptable latency in processing. However, although the SG filter showed acceptable smoothness, its response was a little slow at the beginning and end of the curve. While, the RLOESS filter showed faster tracking than the curve response of the SG filter especially at the beginning of the curve, which might result in information distortion.

In the curve fitting approach, the relation of the filtered data to original data was fitted to a polynomial model with different degrees to assess the performance of the RLOESS, SG and Median filters for their fast response effectiveness to the input data. As outlined in Table 5.2, the lowest value of the SSE metric (0.1553) and the highest value of the R-square (0.9992) were obtained with the RLOESS filter in both cases of the polynomial model degrees. However, the Median filter showed the worst result for the SSE (6.527) and R-square (0.9611) metrics in fast response to input data when compared with the LR & SG filters. Moreover, the lowest range of residual metric was recorded with the LR filter (0.2367) then (0.2369) with the SG filter, while the highest was (0.4744) with the Median filter when the degree of the polynomial was 9. The overall results showed that the RLOESS filter was the most efficient approach in smoothing data for skeletal position data.

5.4 Experiment 2- Kinect camera validation

The objective of this experiment was to validate a Kinect camera's accuracy in spatial gait parameters (step length and step width of gait). An ART motion analysis system was used as a gold standard system which provided 60 frames per second using eight cameras. This test had two parts according to the direction of the walk line.

5.4.1 Forward walk to the Kinect camera view

A participant was instructed to walk three times on the walk line, which was partitioned and labelled by 0.45 m (actual step length) and step width by 0.17 m (actual step width). The direction of the walk line was forward to the Kinect camera. The subject was instructed to repeat the same instructions in a previous trial except for a difference in the actual gait step length and gait step width which were replaced by (0.55 m) and (0.20 m), respectively. The line walk direction was in the same direction as the (x) coordinate system for the ART motion analysis system. Both systems (ART and Kinect) were run to track and record the joint movements of the left and right legs during the walking test. Tables 5.3 & 5.4 below illustrate the error level from the actual values of the step length and width in both systems.

Table 5-3: The Length and Width of Gait Step Data are Provided from Kinect and ART in Forward Walk to Kinect View, when Partitions are Labelled by 0.45 m and 0.17 m.

Gait Features	Actual Values (0.45m for step length and 0.17m for step width)			
	Kinect v2 data		ART data	
	Measurement Values (m)	Error from Actual Value (m)	Measurement Values (m)	Error from Actual Value (m)
Step Length Average (m)	0.4360	0.014	0.4395505	0.0104
Step Width Average (m)	0.2082	0.0382	0.2065008	0.0365

Table 5-4: The Length and Width of Gait Step Data are Provided from Kinect and ART in Forward Walk to Kinect View, when the Partitions are Labelled by 0.55 m and 0.20 m.

Gait Features	Actual Values (0.55m for step length and 0.20m for step width)			
	Kinect v2 data		ART data	
	Measurement Values (m)	Error from Actual Value (m)	Measurement Values (m)	Error from Actual Value (m)
Step Length Average (m)	0.5336	0.0164	0.54003795	0.0099
Step Width Average (m)	0.1862	0.0138	0.18890878	0.0111

5.4.2 Parallel walk to the Kinect camera view

In this trial, the same instructions from the test in the previous section were followed, but the direction of the walk line changed. The Kinect camera was positioned 2 m from the walk line, where the walk direction was set up to be parallel with the Kinect’s x-axis. This walk direction made the y-axis of the ART system the same direction as the walk line. Tables 5.5 & 5.6 illustrate the error level from the actual values of step length and width in both systems, when the walk line direction is parallel with the Kinect camera.

Table 5-5: The Length and Width of the Gait Step Data are Provided from Kinect and ART in Parallel Walk to Kinect View, when Partitions are Labelled by 0.45 m and 0.17 m.

Gait Features	Actual Values (0.45m for step length and 0.17m for step width)			
	Kinect v2 data		ART data	
	Measurement Values (m)	Error from Actual Value (m)	Measurement Values (m)	Error from Actual Value (m)
Step Length Average (m)	0.4348	0.0152	0.4393106	0.0107
Step Width Average (m)	0.1876	0.0176	0.178000	0.008

Table 5-6: The Length and Width of Step Data are Provided from Kinect and ART in Parallel Walk to Kinect View, when Partitions are Labelled by 0.55 m and 0.20 m.

Gait Features	Actual Values (0.55m for step length and 0.20m for step width)			
	Kinect v2 data		ART data	
	Measurement Values (m)	Error from Actual Value (m)	Measurement Values (m)	Error from Actual Value (m)
Step Length Average (m)	0.5292	0.0208	0.5428283	0.007
Step Width Average (m)	0.1857	0.0143	0.206381	0.006

- **Validation of the Kinect camera compared to a high data rate camera system in measurements of gait step length and gait step width.**

The Kinect camera has been validated with the motion capture system (ART) for tracking left and right ankles during the walk process. This test was conducted by four trials. In the first trial, the walk line direction was towards the Kinect camera, which was divided into 0.45 m and 0.17 m as step length, step width, respectively. Both systems tracked the ankles displacement at the same time. The errors from the actual values were as 0.014 m, 0.0104 m, 0.0382 m and 0.0365 m for step length by Kinect, step length by ART, step width by Kinect and step width by ART, respectively. The ART's result was better than the Kinect's result by 0.36 cm and 0.17 cm for step length and step width measurements, respectively.

In the second test, the same procedure as the previous experiment was used, although the actual value changed to 0.55 m, 0.20 m as step length, step width, respectively. The results of the step length and step width by Kinect were less accuracy than the ART measurements by 0.65 cm and 0.27 cm, respectively.

In the third test, the results of both step length and step width by the ART were better by 0.45 cm and 0.96 cm, respectively. Moreover, the last trial records that the results of Kinect measurement are inaccurate when compared to the results of ART system by 1.38 cm and 0.83 cm for gait step length and gait step width, respectively.

5.5 Experiment 3- Kinect for timed walk test

The objective of this test was to investigate the effectiveness of the Kinect camera for a timed walk test in real time. For this purpose, the participant was instructed to walk in front of the Kinect at a random speed. Figure 5.15 below illustrates the timed walk test for 3 meters. This trial was repeated many times to obtain different walk speeds. In each trial, the

participant's speed and the number of gait steps were calculated using two methods at the same time.

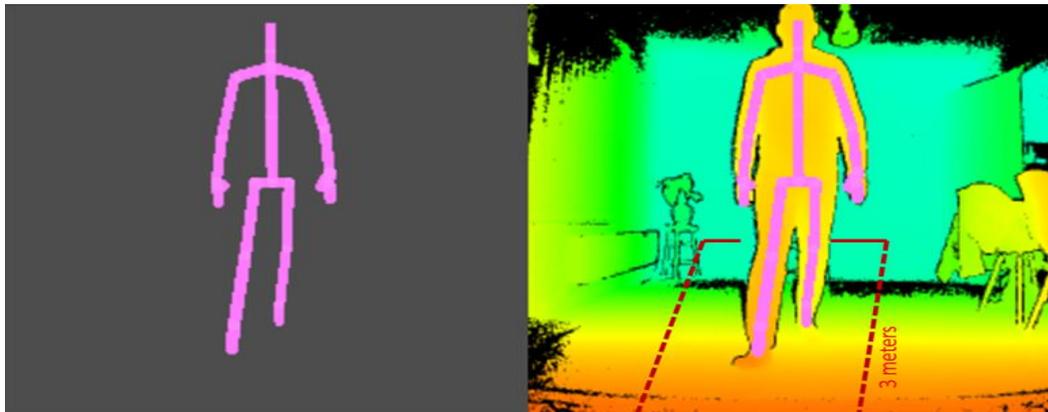


Figure 5.15: Timed Walk Test in Real Time using Kinect Camera for 3 meters; Skeleton Data collection During Walk Test.

Firstly, a stop watch was used to measure the time that was required to cross the distance of 3 meters. Secondly, a JAVA library was introduced by (Barmpoutis, 2013) for Human Body Reconstruction in Real-time. This library has been developed to calculate the participant's speed and the number of gait steps over the same distance. Table 5.7 below shows the error percentage of the Kinect estimation from the actual measurements for both the gait speed and the number of gait steps.

Table 5-7: Timed Walk Test for Five Trials using Two Methods; Stop Watch and Kinect Camera

Manual Measurement (Stop Watch)		Automated Measurement (Kinect)		Error (%)	
Walk Speed (m/s)	Gait Steps (#)	Walk Speed (m/s)	Gait Steps (#)	Speed	Steps
0.495	7	0.402	6	18.8	14.3
0.613	6	0.377	4	38.5	33.3
0.465	7	0.411	7	11.6	0
0.520	6	0.575	6	10.6	0
0.722	6	0.640	5	11.4	16.7

The percentage error (%) was calculated from $[100 \times (actual\ value - estimated\ value) / actual\ value]$, where the Kinect's measurements represented the estimated value.

The results of this test demonstrated that the average error between the two approaches was 18.2 and 12.9 % in measurements of gait speed and steps, respectively

5.6 Experiment 4- Gait features extraction

The objective of this test was to investigate an improvement in the quality of the extracted gait features by using an AM technique to modify the gait length signal. To address

this aim, the gait features were extracted from two methods; a spatiotemporal gait analysis and the proposed Amplitude Modulation (AM) technique.

The experiment involved 35 subjects who were directed to walk forward towards the Kinect camera. The subjects performed different walk speeds randomly. By using MATLAB, the three groups of walk speeds were categorised as: Group One when the walk speed was less than or equal to 0.55, Group Two when the walk speed was more than 0.55 and less than 1 and Group Three when the walk speed was more than or equal to 1 meter/sec. The walk speed was calculated from the displacement of the spine-base joint in the z-axis (as shown in Figure 5.16).

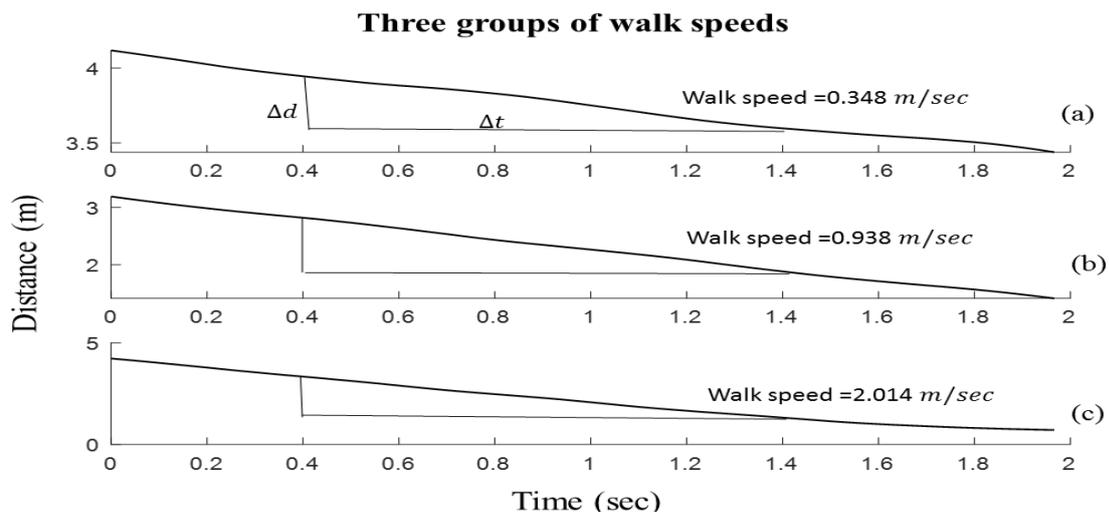


Figure 5.16: Three Walk Speeds Categorisation (a) Group One ($S \leq 0.55$), (b) Group Two ($0.55 < S < 1$), and (c) Group Three ($S \geq 1$ meter/sec).

5.6.1 Spatiotemporal gait analysis

In this method, the gait features were extracted from the gait step length signal, that was calculated from the distance difference between ankles during the walk test. For this purpose, the displacement of both the left and right ankles was measured in the z-axis towards the Kinect camera. Figure 5.17 illustrates the generation of gait length signal, where the peak values represent the gait step lengths, when the ankles are farthest from each other. While, the minimum values of gait length signal occurred when the ankles were closest to each other.

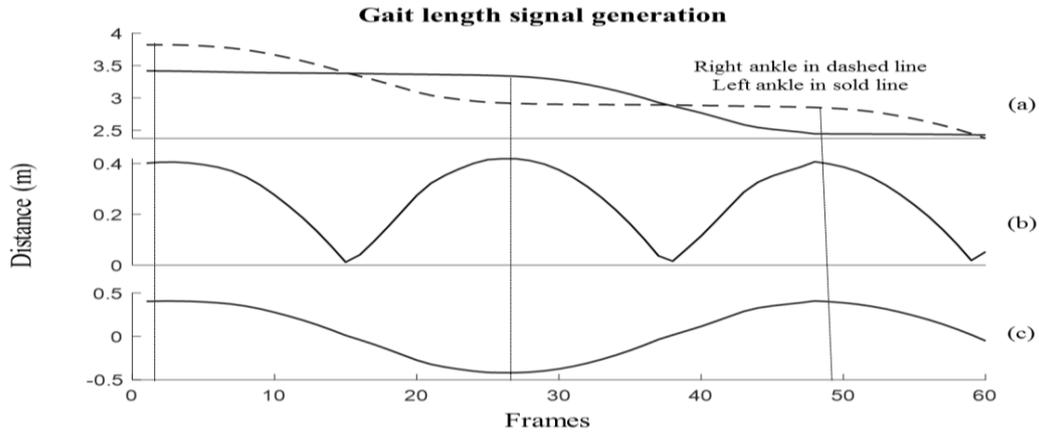


Figure 5.17: Gait Length Signal Generation from the Distance between (a) Right Ankle (dashed line) and Left Ankle (solid line) During Walk, (b) Generated Gait Signal in Absolute Value and (c) Without Absolute Value.

An algorithm was developed to measure the spatiotemporal gait parameters of step length, stride length, stance stages of right and left feet, swing stages of right and left feet, gait cycle, double support stage and gait cadence (as can be seen in Figure 5.18).

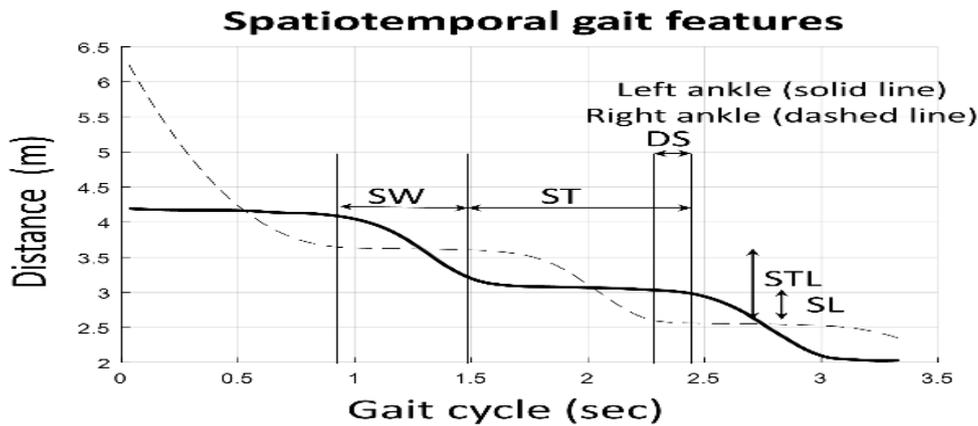


Figure 5.18: The Determination of a Complete Gait Cycle for Extracting Step Length (SL), Stride Length (STL), Double Support Time (DS), Swing Time (SW), Stance Time (ST) and Gait Cycle Time (SW+ST).

The spatiotemporal gait features were applied to a set number of classifiers to categorise the gait data into three classes, according to the walk speed changes. Table 5.8 lists the evaluation of each classifier using several metrics, including sensitivity, specificity, precision, accuracy, F-measure and area under curve (AUC). The available data was formed as a matrix of 105×8 where the rows represented the samples and the features were located on the columns of the dataset matrix. This dataset is grouped into three classes based on the walk speed and each class contains 35 samples.

Table 5-8: The Evolution of Data Gait Classification using Spatiotemporal Analysis for Gait Feature Extraction

Approach	Sensitivity	Specificity	Precision	Accuracy (%)	F-Measure (%)	AUC
Decision Tree	0.886	0.943	0.885	88.6	88.5	0.887
k-NN	0.857	0.929	0.860	85.7	85.9	0.893
L-SVM	0.895	0.947	0.903	89.5	89.9	0.953
Q-SVM	0.905	0.952	0.910	90.5	90.7	0.980
L-Discriminant	0.895	0.948	0.909	89.5	90.2	0.977
Q-Discriminant	0.910	0.955	0.911	91.0	90.8	0.980

5.6.2 Proposed method (Amplitude Modulation)

In this experiment, the Amplitude Modulation (AM) technique was used to extract new gait features. First, the gait length signal was transformed into the AM domain. Figure 5.19 below shows the use of the AM technique to obtain the modified gait length signal $x_{AM}(t)$ by multiplying the gait length signal $x_g(t)$ to the reference signal $x_c(t)$.

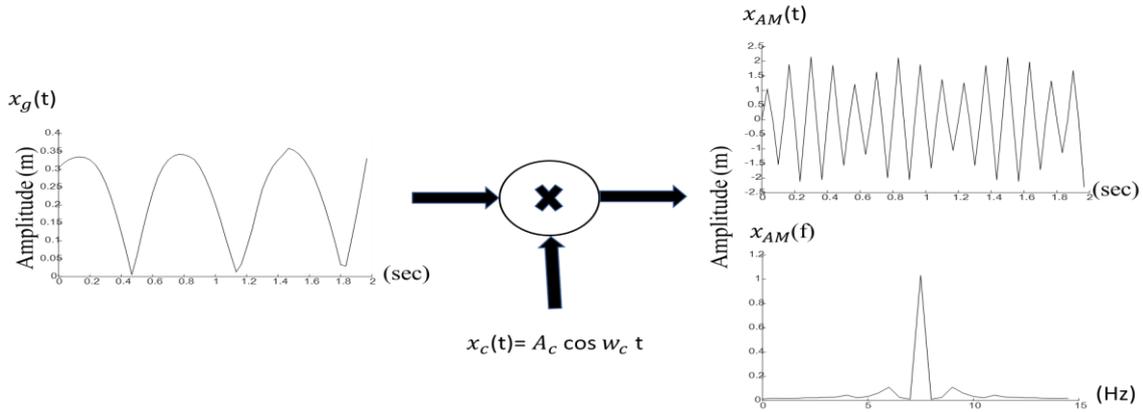


Figure 5.19: Gait Length Signal Conversion into AM Domain

The parameters of the reference signal were chosen with the amplitude of $A_c = 1$ as a normalized value and the frequency as $f_c = 25 \text{ Hz}$, which is a quarter of the sampling frequency (100 Hz). This sampling frequency value was adjusted until the spectrum graph of the reference signal showed the amplitude as equal to one. As can be seen in Figure 5.20, where several values of sampling frequencies ($f_s=90, 100, 120 \text{ Hz}$) were adjusted to represent the reference signal in the frequency domain with an amplitude equal to one. The below figure demonstrates that when the sampling frequency is at 100 Hz this makes the amplitude of the reference signal equal to one, without destroying the shape of the spectral signal.

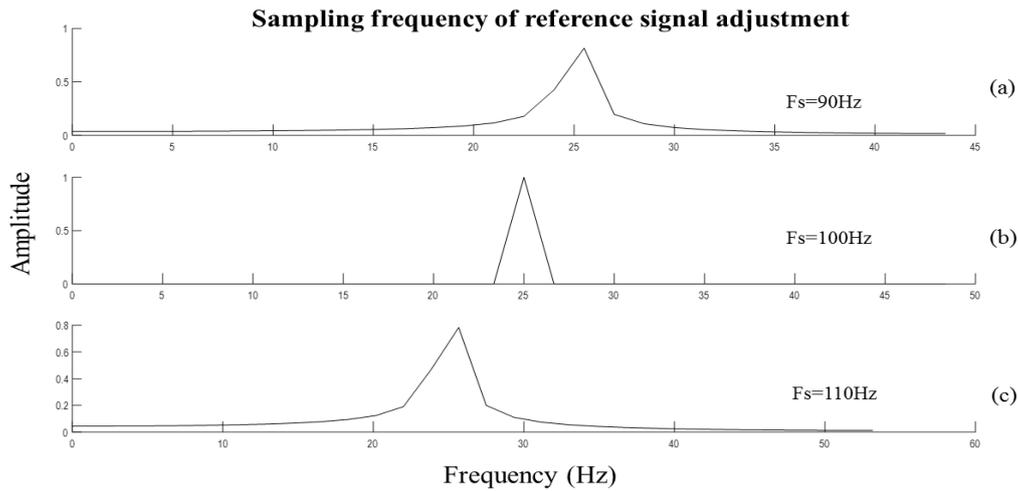


Figure 5.20: Sampling Frequency Adjustment for Reference Signal as: $f_s =$ (a) 90 Hz, (b) 100 Hz and (c) 110 Hz.

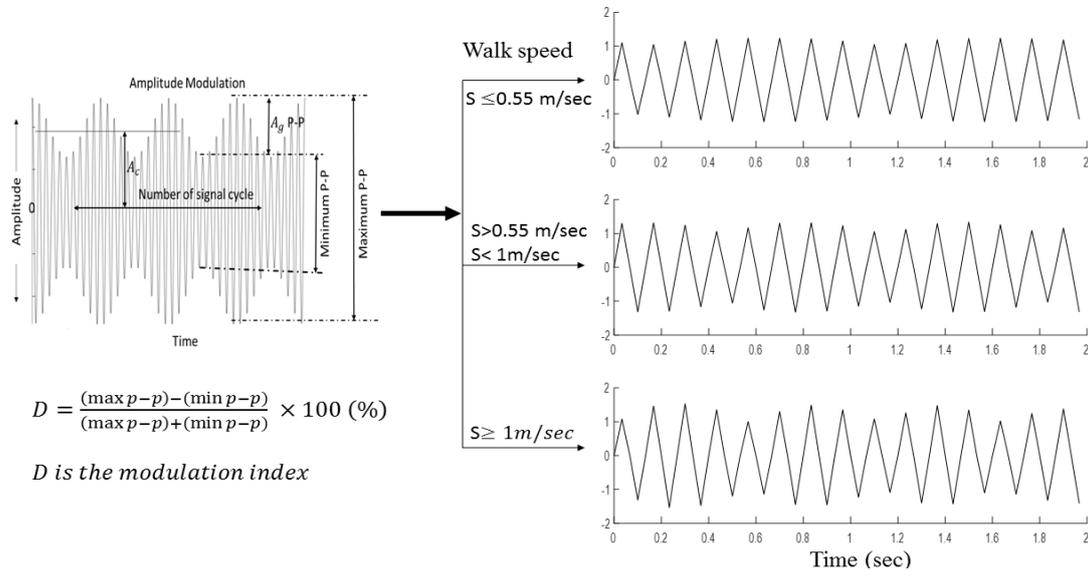


Figure 5.21: The AM-Modified Gait Signal in the Time Domain Changes its Features According to the Walk Speed.

This modified signal can be represented in the time domain as shown in Figure 5.21, and in the frequency domain, as shown in Figure 5.22, for three groups of walk speeds. The parameters of the modified gait signal were extracted to follow the gait pattern changes due to the walk speed changes. To assess the quality of the extracted gait features from the modified gait length signal using the AM technique, a set number of classifiers were applied to classify the gait data into three classes, according to the walk speed changes. Table 5.9 lists the evaluation of each classifier using several metrics including; sensitivity, specificity, precision, accuracy, F-measure and the area under curve (AUC). These evaluative metrics were used for both cases (gait length signal before modification and after modification using the AM technique), to investigate which method could improve the accuracy of gait data classification.

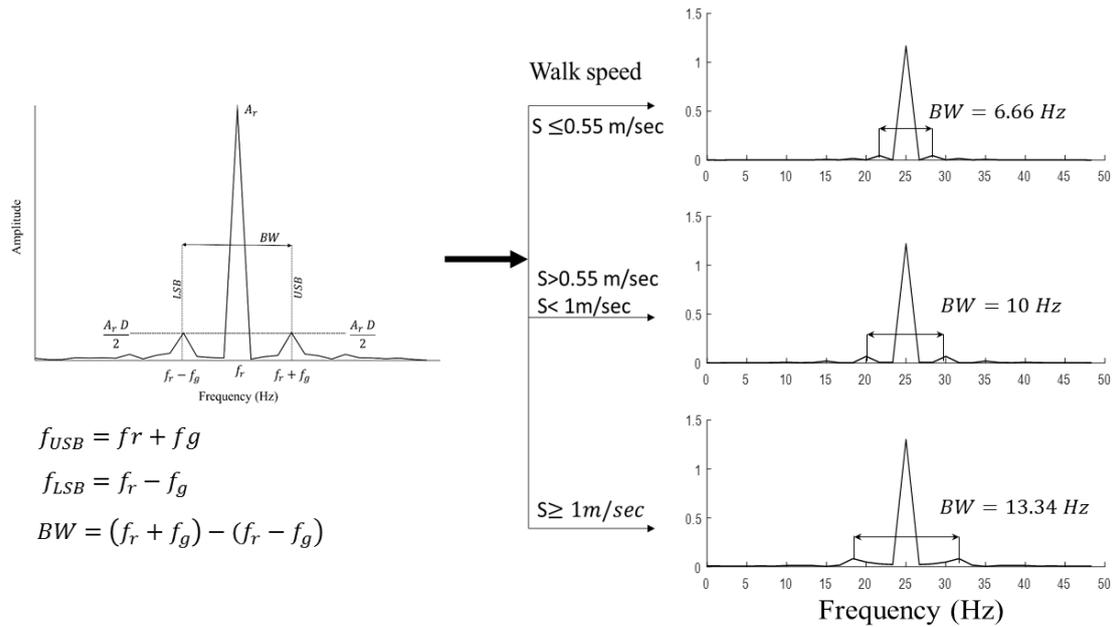


Figure 5.22: AM-Modified Gait Signal in the Frequency Domain Changes its Features According to the Walk Speed.

Table 5-9: The Evolution of Data Gait Classification using the AM Technique for Gait Feature Extraction

Approach	Sensitivity	Specificity	Precision	Accuracy (%)	F-Measure (%)	AUC
Decision Tree	0.810	0.905	0.809	81.0	80.9	0.890
k-NN	0.895	0.948	0.897	89.5	89.6	0.920
L-SVM	0.933	0.967	0.934	93.3	93.4	0.953
Q-SVM	0.933	0.967	0.934	93.3	93.4	0.980
L-Discriminant	0.838	0.919	0.845	83.8	84.2	0.910
Q-Discriminant	0.857	0.929	0.868	85.7	86.3	0.923

• **Compression of AM and Spatiotemporal Approaches in Gait features Extraction**

The sensitivity, specificity, precision, accuracy, F-measure, and AUC evaluative metrics were all investigated for various classifiers used in this test. It is noticeable that the Quadric Discriminant (QD) classifier achieved the best results in gait speed classification for spatiotemporal gait approaches with an accuracy rate of 91% compared to the other classifiers. In addition, the sensitivity and specificity metrics have demonstrated the effectiveness of the QD at a high rate of true positive detection (0.91), and true negative detection (0.955), respectively. Notably, the highest classification accuracy reached (93.3%) with the Quadric SVM and Linear SVM approaches when the AM technique was used for gait feature extraction. In this case, the sensitivity, specificity and precision reached (0.933), (0.967) and (0.934), respectively. In the case of spatiotemporal gait approach, the highest measure of the AUC (0.980) obtained for the Quadric SVM and Quadric Discriminant classifiers. However, that

level of AUC measure was only reached in the case of the AM technique with the Quadric SVM, while the F-measure attained a value of 93.4%, which was the best amongst all classifiers for both. In summary, this was achieved with the Linear and Quadric SVMs, when the AM technique was used for gait feature.

5.7 Experiment 5- AM&FM Gait Features Extraction

- **AM-Modified Gait Length Signal**

First, the gait length signal $g(t)$ was generated by calculating the distance between the ankles during the walk test, and then $g(t)$ was converted into the AM domain. The AM-modified gait length signal $M_{AM}(t)$ was used to extract three gait features, including the bandwidth (in Hz), modulation index (ratio) and the amplitude level of both the upper and lower side bands (in meter). These AM modified gait signal parameters were extracted for all walking speed groups. A set of supervised classifiers were applied for these features to classify three classes based on walk speed. Table 5.10 below illustrates an evaluation of the performance of each classifier.

Table 5-10: The Evolution of Data Classification using AM Tech for Gait Feature Extraction

Approach	Sensitivity	Specificity	Precision	Accuracy (%)	F-Measure (%)	AUC
Decision Tree	0.848	0.924	0.846	84.8	84.69	0.910
k-NN	0.838	0.919	0.838	83.81	83.82	0.840
L-SVM	0.810	0.905	0.825	81.0	81.73	0.930
Q-SVM	0.914	0.957	0.921	91.4	91.75	0.970
L-Discriminant	0.829	0.914	0.839	82.9	83.35	0.897
Q-Discriminant	0.924	0.962	0.924	92.4	92.38	0.950

- **FM-Modified Gait Length Signal**

In this part, the gait length signal was converted into the FM-NB domain to extract three gait features including the Bandwidth (in Hz), frequency deviation (in Hz) and the amplitude level of both side bands (in meters) from the FM-modified gait length signal $M_{FM}(t)$. Table 5.11 illustrates the performance of a set number of the classifiers for walk speed classification.

Table 5-11: The Evolution of Data Classification using FM Tech for Gait Feature Extraction

Approach	Sensitivity	Specificity	Precision	Accuracy (%)	F-Measure (%)	AUC
Decision Tree	0.8000	0.9000	0.7954	80.0	79.77	0.877
k-NN	0.8762	0.9381	0.8765	87.62	87.64	0.910
L-SVM	0.8476	0.9238	0.8453	84.80	84.65	0.953
Q-SVM	0.8190	0.9095	0.8184	81.90	81.87	0.953
L-Discriminant	0.8095	0.9048	0.8071	81.00	80.83	0.920
Q-Discriminant	0.8667	0.9333	0.8678	86.70	86.73	0.950

- **Compression of AM and FM Techniques in Gait Features Extraction**

The results show that the AM technique was more efficient than the FM technique for improving the quality of extracted gait feature, when the classification data based on walking speeds was evaluated using several evaluation metrics. The highest classification accuracy and F-measure reached 92.4% and 0.9238, respectively. This was obtained from the Q-discriminant classifier when the AM-modified gait signal was used for gait feature extraction. In addition, the second highest accuracy in data classification was also achieved when the AM technique was used, which reached 91.4% with the Q-SVM classifier. In the FM case, the highest classification accuracy and F-measure were 87.6% and 87.64% with the k-NN approach, respectively. While the AUC results were approximately between 0.91 and 0.95. However, the best value of the AUC was 0.97, when performed with the Q-SVM classifier with the AM used for modifying the gait length signal to be in the AM domain. The precision metric records were just 0.8765 as the highest value in the FM case, whilst it reached 0.924 with the AM technique. All supervised learning approaches showed more improvement in the sensitivity and specificity results that when the AM was used rather than the FM, but with one exception, when using (k-NN) that showed better results in both the sensitivity and specificity with the FM.

5.8 Experiment 6- Gait Features Classification

The objective of this test was to investigate the effectiveness of the AM/EC system for classification of gait pattern changes. To address this aim, the experiment involved 40 subjects who were instructed to walk forwards towards the Kinect camera at random walk speeds, this was repeated many times to increase the walk speed range. Three groups of walk speeds were specified as less than or equal to 0.6, greater than 0.6 and less than 1.1, and greater than 1.1 meters per second. Each group contained 40 samples, overall three classes of 120 samples.

5.8.1 AM/CE System Implementation

The AM/CE system was implemented to process the collected data in six stages. The first two stages were the collection and smoothing of gait data, as demonstrated in the previous experiments. The rest of the stages included gait feature extraction using the AM technique, gait feature reduction using the PCA approach, gait feature classification using the CE technique and finally the evaluation of the system performance, as follows:

- **Gait Feature Extraction using the AM Technique**

A gait length signal was generated for each sample by calculating the distance between the ankles during the walking test. The AM technique was used to convert the gait length signal from the baseband into the passband (AM domain), consequently an AM-modified gait length signal was obtained for 120 samples to extract the new gait features. Seven gait features were extracted for each sample including the upper side band frequency, lower side band frequency, bandwidth, modulation index, modulation efficiency, amplitude level of side bands, and the total amplitude level of the AM-modified gait signals. The dataset was formed as a matrix with 120 samples (rows) and 7 features (columns).

- **Gait Feature Reduction using the PCA Approach**

A rescale approach known as mean normalisation was used to normalize the seven vectors features. The PCA technique was then applied to reduce the dimension of dataset matrix, where the eigenvalues were calculated to indicate the eigenvectors that could capture the highest percentage of data variation. Consequently, the first and second principle components of PC1 & PC2 were selected, as they explained 90.57 % of total data variance, as detailed in Table 5.12 below.

As shown in Table 5.13 below, seven gait features were reduced into two vectors using the PCA approach for a subset of the data, (the full dataset can be seen in Appendix C table 2).

Table 5-12: Dataset Reduction Based on Eigenvalues and Captured Variance of Total Data

Variable	PC1	PC2	PC3	PC4	PC5	PC6
x1	-0.187	0.672	0.074	-0.083	-0.0299	-0.0048
x2	-0.187	0.672	0.074	-0.083	-0.0299	-0.0048
x3	-0.472	-0.068	-0.187	0.1038	0.5698	0.6343
x4	-0.465	-0.078	0.0949	0.6898	-0.5392	0.0453
x5	-0.472	-0.100	-0.106	0.0528	0.4062	-0.7668
x6	-0.088	-0.039	-0.922	-0.1833	-0.3273	-0.0039
x7	-0.511	-0.272	0.2898	-0.6807	-0.3326	0.0868
Eigenvalues	$\lambda_1=0.1835$	$\lambda_2=0.0216$	$\lambda_3=0.0172$	$\lambda_4=0.0024$	$\lambda_5=0.0014$	$\lambda_6=0.00034$
Total variance	81.036	9.539	7.59	1.06	0.618	0.0015
$\left(\frac{\lambda_i}{\sum \lambda}\right) \%$						

Table 5-13: The Seven Gait Features from the AM-Modified Gait Signal are Reduced into Two Vectors using PCA

Seven columns of gait features from AM-Modified gait signal							PCA	
Lower-fre	Upper-fre	Bandwidth	Mod index	Mod efficiency	Sides level	Total level	PC1	PC2
21.6667	28.3333	6.6667	0.0703	0.0025	0.0747	0.9777	-1.0462	0.0237
21.6667	28.3333	6.6667	0.1009	0.0051	0.1151	1.0648	-0.9600	0.0630
20.0000	300000	10.0000	0.0969	0.0047	0.1012	0.9729	-0.9599	0.0629
15.0000	35.0000	20.0000	0.0056	0.0000	0.0064	1.0213	-0.9586	0.0626
15.0000	35.0000	20.0000	0.0056	0.0000	0.0064	1.0213	-0.9274	0.0518
15.0000	35.0000	20.0000	0.0056	0.0000	0.0064	1.0213	-0.8954	0.0242
21.6667	28.3333	6.6667	0.1258	0.0079	0.1420	1.0654	-0.8941	0.0241
21.6667	28.3333	6.6667	0.0958	0.0046	0.0973	0.9455	-0.6306	0.0195
21.6667	28.3333	6.6667	0.0825	0.0034	0.0816	0.9157	-0.6100	0.0165
21.6667	28.3333	6.6667	0.0958	0.0046	0.0973	0.9455	-0.5752	0.0125
13.3333	36.6667	23.3333	0.0055	0.0000	0.0059	0.9538	-0.5438	0.0115

- **k-Fold Cross Validation**

The k-fold Cross Validation approach was used to assess the system performance in unseen data classification. First, the data was divided into five folds (k=5), and each fold was used once as a testing set, while the rest were used as training sets, (as shown in Figure 5.23). Secondly, the data was divided into ten folds (k=10), and each fold was used once as a testing set, while the rest were used as training sets, (see Figure 5.24).

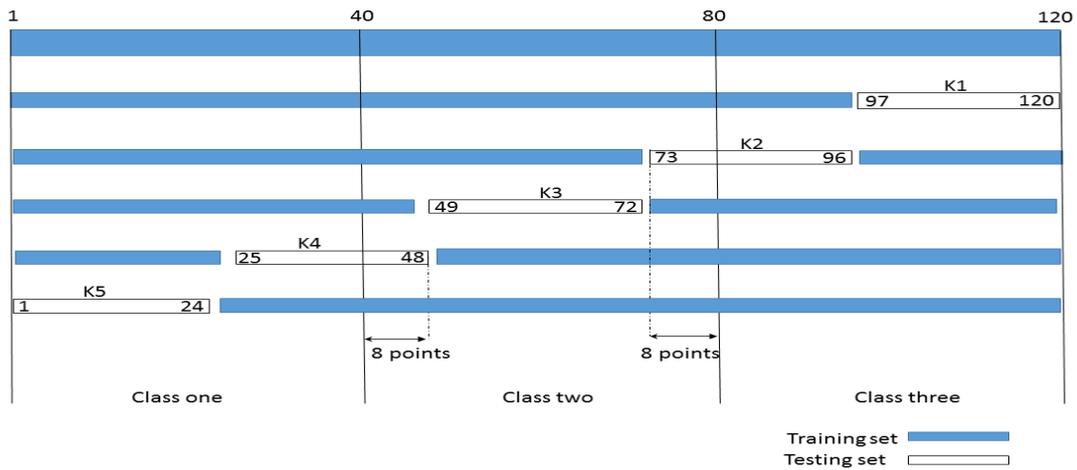


Figure 5.23: A 5-Fold Cross Validation for Dataset Containing 120 Samples.

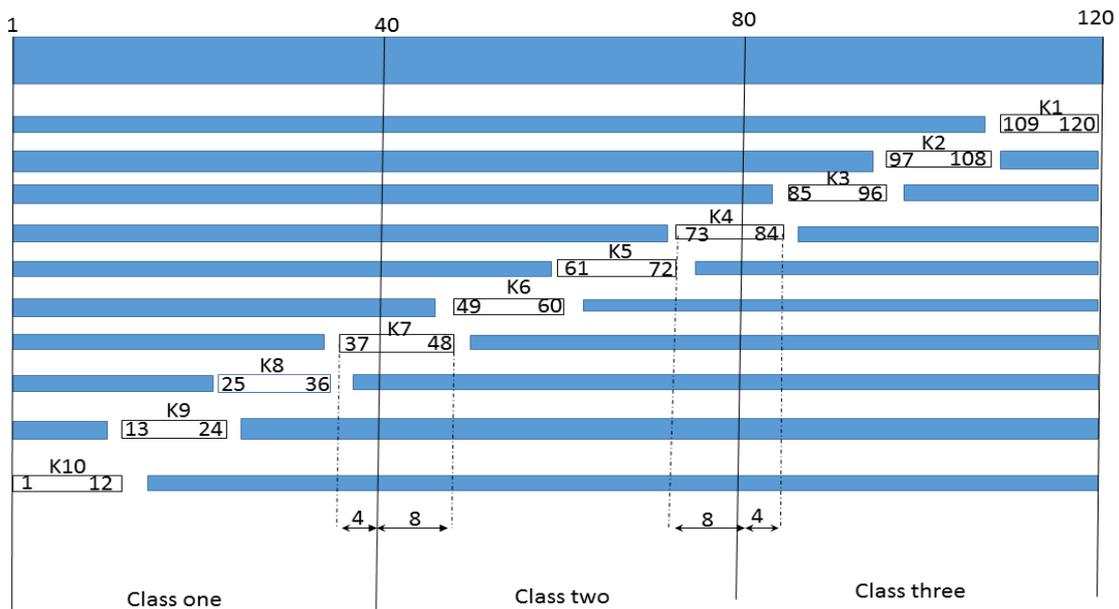


Figure 5.24: A 10-Fold Cross Validation for Dataset Containing 120 Samples.

- **Unseen Data Classification using the CE Technique**

In the unseen gait data classification, the CE technique was applied for the first and second principal components (PC1 & PC2) that represented 90.57 % of data variation (as shown in Table 5.12). The classification was made by using the CV approach in both cases of $k=5$ and $k=10$ to assess the performance of CE in the unseen data classification. The similarity metric for the classes was made based on the HD measure and the order of the bit's position, where a low value of HD between the codeword and threshold meant a low error rate, which led to the similarity decision being highly considered for that class. Figures 5.25 & 5.26

illustrate the error rate calculation of the CE for three classes of gait speed classification with both cases; $k=5$ & $k=10$, respectively.

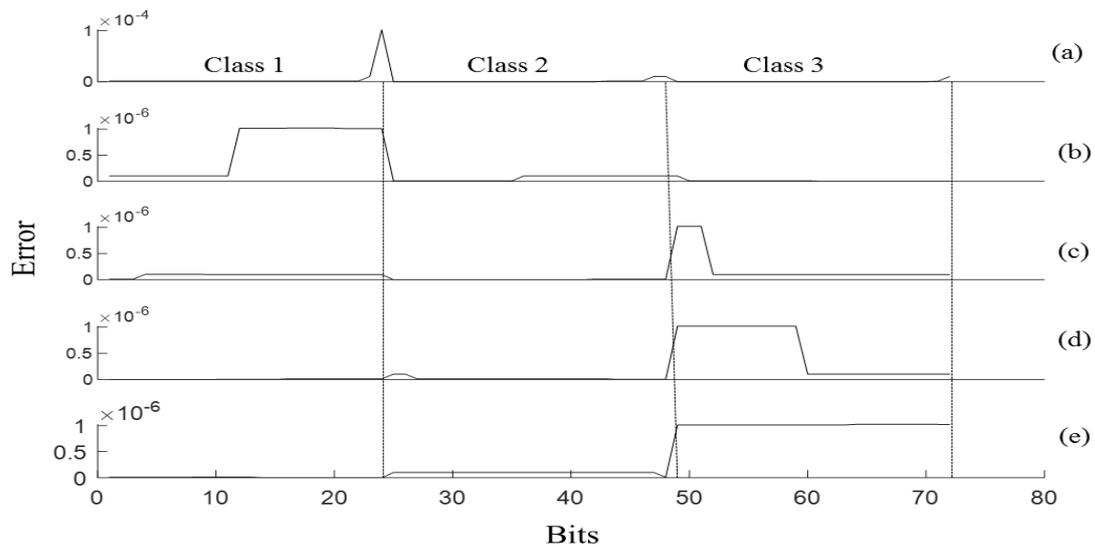


Figure 5.25: Error Rate Calculation of CE in 3-Classes of Gait Speed Classification when 5-Fold CV; (a) to (e) Represent Folds from 1 to 5.

In the case of the 5-fold CV, the dataset contained 120 samples (codewords) which were divided into five-folds, and each fold contained 24 samples (codewords). For the 10-fold case, 12 samples (codewords) were included. The similarity metric was considered as one minus error rate, where the error rate was calculated from the Equation (4.61).

The error rate was defined using the HD measure between each codeword within the fold and thresholds of class one, class two and class three. Hence, the lowest value of the HD among the three calculations (i.e. HD1, HD2 or HD3) would be highly similar to that class. For instance, the lowest error rate in Figure 5.25 (a), (c) and (e) were correlated to classes 3, 2 and 1, respectively. Whilst, in case of the 10-fold CV, Figure 5.26 shows that the lowest value of error rate can be obtained when [(a),(b) & (c)], [(d), (e), (f) & (g)] and [(h), (i) & (j)] correlated to classes 3, 2 and 1, respectively.

For the system evaluation, the ability of the CE technique for classification of the gait speed data are listed in Tables 5.14 and 5.15, for both 5-fold and 10-fold CVs, respectively. These Tables illustrate the number of classes that were correctly and incorrectly predicted over three classes of walk speeds.

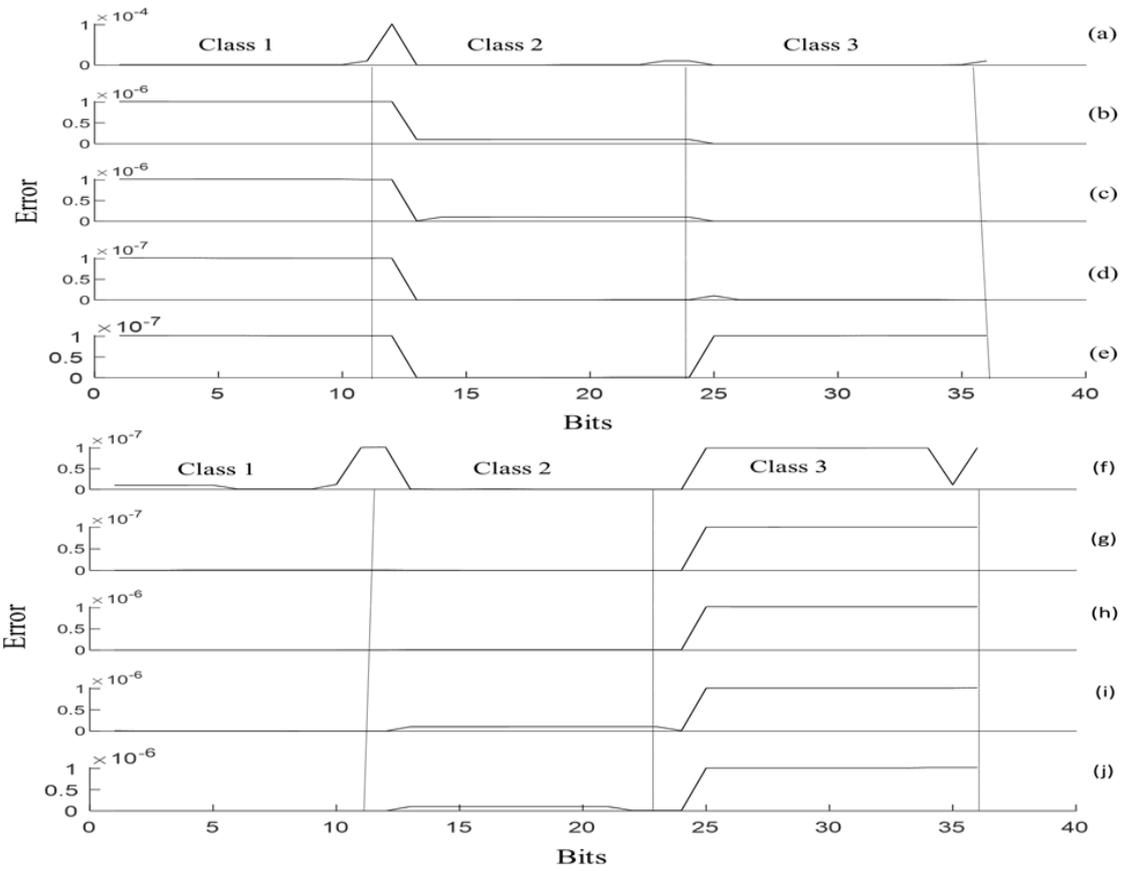


Figure 5.26: Error Rate Calculation of CE in 3-Classes of Gait Speed Classification when 10-Fold CV; (a) to (j) Represent Folds from 1 to 10.

Table 5-14: Confusion Matrix for Three Classes in Case of 5-Fold CV

		Predicted Classes		
		a	b	c
Actual Classes	a	40	0	0
	b	2	37	1
	c	0	1	39

Table 5-15: Confusion Matrix for Three Classes in Case of 10-Fold CV

		Predicted Classes		
		a	b	c
Actual Classes	a	38	2	0
	b	0	40	0
	c	0	0	40

Evaluation of the CE technique for gait speed classification was assessed using several evaluation metrics, including sensitivity, specificity, precision, accuracy and the F-measure. Tables 5.16 and 5.17 show the results of CE system evolution in unseen gait data classification for the 5-fold and 10-fold CVs respectively.

Table 5-16: CE Classifier Performance in Case of 5-Fold Cross Validation (k equals 5)

Approach	Sensitivity	Specificity	Precision	Accuracy (%)	F-Measure (%)
CE k-fold = 5	0.9667	0.9833	0.9670	96.67	96.68

Table 5-17: CE Classifier Performance in 10-Fold Cross Validation (k equals 10)

Approach	Sensitivity	Specificity	Precision	Accuracy (%)	F-Measure (%)
CE k-fold = 10	0.9833	0.9917	0.9841	98.33	98.37

Table 5-18: Supervised Classifiers Performance in both Cases of 5-Fold & 10-Fold Cross Validation

Approach		Sensitivity	Specificity	Precision	Accuracy (%)	F-Measure (%)
Decision Tree	k = 5	0.9167	0.9583	0.9166	91.7	91.66
	k=10	0.9167	0.9583	0.9175	91.7	91.71
k-NN	k = 5	0.9667	0.9833	0.9667	96.7	96.67
	k=10	0.9667	0.9833	0.9667	96.7	96.67
L-SVM	k = 5	0.9250	0.9625	0.9345	92.5	92.97
	k=10	0.9417	0.9708	0.9472	94.2	94.44
Q-SVM	k = 5	0.9222	0.9589	0.9233	91.8	92.28
	k=10	0.9333	0.9667	0.9380	93.3	93.57
L-discriminant	k = 5	0.9417	0.9708	0.9504	94.2	94.60
	k=10	0.9500	0.9750	0.9565	95.0	95.32
Q-discriminant	k = 5	0.8167	0.9083	0.8234	81.67	82.00
	k=10	0.8167	0.9083	0.8234	81.67	82.00

Moreover, the performance of the CE technique for classification of the gait pattern changes was compared to a set of supervised classifiers. These classifiers were applied to the same dataset (seven vectors of gait features) to classify the data into three classes according to the

type of walk speed. Table 5.18 illustrates the performance of a set number of supervised classifiers including DT, k-NN, L-SVM, Q-SVM, L-discriminant and Q-discriminant. For this purpose, the same metrics were used to assess the performance of the classifiers in both cases of k-fold (k=5 and k=10), as it shown in Figures C.2 & C.3 in Appendix C.

- **Performance of the Convolutional Encoder in Gait Data Classification Compared to Set Number of Supervised Classifiers in Case of 10-Fold Cross Validation**

The performance of the CE technique for the classification of gait pattern changes was conducted using several evaluation metrics and then compared to the performance of a set number of classifiers. In this comparison, the k-fold CV approach was used to assess the unseen data in both cases of the 5-fold and 10-fold. The highest accuracy of classification and F-measure were obtained with the CE of 10-fold CV as 98.33% and 98.37%, respectively. For 5-fold CV however the highest classification accuracy for supervised classifications was 96.7% due to the k-NN, while the F-measure still showed the highest achievement with the CE technique (96.68 %). Generally, the classification accuracy was improved when the 10-fold was applied. The three highest improvements were 1.7%, 1.66% and 1.5% for the L-SVM, CE and Q-SVM, respectively. At the same time, sensitivity, specificity and precision all reached their peak values with the CE technique as 0.9833, 0.9917 and 0.9841, respectively. Whilst, their worst rates were obtained with the Q-Discriminant supervised classifier as 0.8167, 0.9083 and 0.8234, respectively.

5.9 AM/CE System Performance

The performance of the AM/CE system was assessed using several evaluation metrics as described in the previous sections. The results showed an acceptable rate of classification accuracy in gait pattern changes. For further explanation, the outcomes of the AM/CE system were compared to the other studies, as outlined in Table 3.1. The comparison involved the studies that explored spatiotemporal gait analysis for classification of gait pattern changes using 3D skeleton data from a Kinect camera. Table 5.19 below shows the AM/CE performance in gait data classification compared to a set of previous studies. The compression is based on three conditions; 1) gait cycle determination which means that the gait features extraction was obtained within the period of gait cycle, 2) data used which is related to one of

three kinds of gait features analysis including Spatiotemporal gait features, Kinematic gait features or Kinetic gait features, 3) accuracy of gait data classification.

Table 5-19: AM/CE System Performance in Classification of Gait Pattern Changes

Approaches	Gait Cycle Detection	Data Used	Classification Accuracy (%)
(Bei et al., 2018)	Needed	Skeletal positional & angular data	94.2
(Tupa et al., 2015)	Needed	Static segment length + Skeletal positional data	97.2
(Vektor, P., 2018)	Needed	Skeletal positional data	98.33
Proposed approach	Not needed	Skeletal positional data	98.33

In the above table, it is noticeable that, the proposed approach achieved the highest rate of the classification accuracy compared to the other methods. Vektor, 2018 reached the same achievement, but its method still needed the gait cycle determination for extracting the relevant gait features. Gait cycle detection is a difficult task to be defined correctly (Zeni, Richards & Higginson, 2008), where the error level in gait cycle detection will be cumulative quantity in gait features exaction (Khan & Badii, 2019). In addition, gait cycle requires extra approaches to detect it, which might lead to an increased cost of the computational process. The rest of the methods demonstrated high level of classification but are still less than the achievement of the AM/CE system. What makes the proposed approach unique and robust is that it was based on one kind of data, and the gait features have been extracted without the need to detect the gait cycle period. This approach achieves the same or better results with less computational cost.

5.10 Summary

The gait analysis system was implemented based on 3D skeletal data of lower body joints using a Kinect camera. This analysis system is called the AM/CE approach, which aims to improve the classification accuracy of gait pattern changes, where the dataset was collected from a low data rate device. To address the aim of the AM/CE system, six-stages were involved, and a set of algorithms and machine learning approaches were proposed and developed to increase the quality of the extracted gait features to be more efficient and representative parameters in gait analysis system.

This chapter has involved several experiments to explain the purpose of the main stages of the proposed system practically. Moreover, the performance of the AM/CE system has been assessed by comparing its outcomes to the results of the other approaches.

In **experiment one**, the discriminations among six filters were highlighted based on two criteria; time delay and the fast response of the output to the input of data. The results showed that the RLOESS filter was the most efficient tool in the filtering process, where the correlation of the coefficient and R-square between its input and output of data was the highest. In addition, the SSE and residual metrics were the lowest with the RLOESS filter. This guaranteed that the original data was still preserved after the filtering process.

In **experiment two**, a comparison between the Kinect and ART cameras for the effectiveness of the spatial gait measurements was conducted and compared to actual values. The outcomes showed that the Kinect was less accurate than the Marker-ART system, due to the low data rate of the Kinect.

In **experiment three**, an investigation into the effectiveness of the Kinect in a Timed Walk Test at home was performed. However, the results showed a level of error in the Kinect estimation from the actual values for both the gait speed and the number of gait steps.

In **experiment four**, the quality of the gait feature improvement was investigated by using the AM technique. For this purpose, a gait signal was mapped into the AM domain to extract new gait features. The results showed an improvement in the accuracy of gait data classification when the AM-modified gait signal was used for gait feature extraction. This improvement was investigated by comparing the traditional approach of spatiotemporal gait analysis and applying a set number of supervised learner approaches, where the highest improvement reached (2.5%) over the traditional method.

In **experiment five**, the effectiveness of the AM technique in gait feature extraction was investigated by comparing it to the FM-NB technique. The extracted gait features from both techniques were evaluated based on the gait pattern changes. For this purpose, a set number of classifiers were used to classify the gait pattern changes based on gait speed, with the results demonstrating that 3.78% higher classification accuracy was achieved in the case of the AM in comparison with the FM.

In **experiment six**, the performance of the AM/CE system was evaluated for the classification of gait pattern changes based on the gait speed. The result showed that the

proposed method achieved the correct classes prediction and higher by 1.63% over a set number of supervised learning approaches.

6. CONCLUSIONS AND FUTURE WORK

This chapter describes the research aim by showing the relevant objectives for each stage of the implemented system, as well as the research contribution, limitations and finally recommendations for future work which could prove beneficial for readers.

To conclude, I recall from earlier the **research question** of this thesis. Specifically:

“Can the automated gait analysis technique be used to improve the effectiveness of distinguishing gait pattern changes and ranking them based on walk speed, in particular when a low-cost, low data rate sensing device is used to collect the data?”

Amplitude Modulation and Convolutional Encoder approaches can play a major role in improving the performance of gait speed analysis, especially when the data collection is performed using a Kinect camera, which shows usually low measurement accuracy compared to other motion capture systems due to its low data rate. The proposed method is called an AM/CE system which has been developed to smooth the collected data of 3D positional lower body joints during movement and to extract/reduce the relevant gait features to be used in the classification of gait pattern changes based on walk speed.

The enhancement of extracted features quality was performed using the AM technique to map the original data into an AM domain. The main advantage of the modified gait signal is that it can be represented by sampling frequency more than an unmodified gait signal. This leads to an increase in the accuracy of measurements when the gait analysis is performed for gait feature extraction. Moreover, the combination of AM & CE techniques into one proposed system has been investigated as to its effectiveness in gait data classification based on the changes of walk speed.

6.1 Research Aim and Scope

The main aim is to develop a reliable model that can enhance the quality of extracted gait features in order to be efficiently analysed and effectively classified, based on the changes in walk speed. To address this aim, a gait analysis system was constructed and implemented, in six stages, in order to collect the 3D skeleton data using a Kinect camera. The implemented

system might be used to contribute to an improvement in the quality of life for elderly people, who prefer to live independently, by detecting abnormality in gait speed.

6.2 Research Objectives Definition

The proposed system is consisted of several stages to achieve the aim of this research. Therefore, at each stage, the data was processed to investigate a set of objectives. As follows:

- **To differentiate the best smoothing approach among a set of filtering techniques for 3D skeletal positional data, based on less time delay and fast response.**

A set number of filters were assessed based on two criterions. Firstly, less time delay and secondly the fast response of the output to the input data of the filter. These criterions seek the time cost of processing that can improve the measurement accuracy. Moreover, the fast response of the filter output that can retain the original data, especially at the concave up and down of the data curve.

The correlation coefficient between the original and filtered data was calculated to determine the efficient filters for smoothing the data with less time delay, whereas the curve fitting was used to evaluate the filters that showed a high level of agreement between their input and output data. In the curve fitting approach, several metrics were used, including SSE, R-square and residual. The fast response of the filter showed a small value in residual and SSE measures and a high value for R-square metric. The measurements are performed for original and filtered data to fit the polynomial curve with several degrees.

- **To validate the effectiveness of a Kinect camera compared to a high data rate camera system.**

The RTA system is used as a gold standard system to validate the Kinect camera v2, this system involves eight cameras that collect data from the markers in the 3D space with a data rate of 60 frames per second. The participant was instructed to walk in front of the Kinect, with the walk line placed in an area that could be captured by both systems; ART and Kinect. This line of walk was apportioned into five sections that represented the actual length of gait steps on the ground. Then a comparison between both systems was performed by calculating the error values from the actual lengths of gait steps which were labelled on the ground.

- **To develop an autonomous system that can perform a timed walk test for investigating that the changes in walk speed might be detected efficiently by increasing the quality of extracted gait feature.**

A timed walk test is commonly used for gait assessment, which basically aims to measure the gait speed. In this test, the participants were directed to walk about 3 meters with a random speed, and the time is recorded using stop watch, when the participants started to walk till the end line is reached. During the test time, the Kinect was prepared to track the participant's movements in z-axis, and Java for Kinect library is developed to calculate the gait speed and number of gait steps in real time. Finally, the Kinect's estimation has been compared to the actual measure for gait speed and number of gait steps over 3 meters.

- **To introduce a new technique for extracting new gait features without the requirement of gait cycle determination.**

The Amplitude Modulation (AM) technique is used for mapping gait length signals from its baseband into the passband spectrum. The AM domain modifies the gait signal to be more efficient for representative features in gait pattern changes. Consequently, a comparison was performed between the modified and unmodified gait signals to investigate the extracted features that could be efficiently representative in gait speed classification. The investigation was carried out by using three types of data; including spatiotemporal gait data and the linear velocity of ankle data during a walk test.

- **To use data reduction technique for reducing the matrix dimension of extracted features.**

The Principal Component Analysis (PCA) was used to reduce the dimensionality of the gait feature matrix. The most representative features were adopted whilst the other feature vectors were ignored. The eigenvalues were calculated to determine which eigenvector PCs would be considered. Subsequently, the feature matrix was reduced to include just two vectors representing the largest axis width of the variance data.

- **To improve the accuracy of gait data classification in different walk speeds using Convolutional Encoder (CE) technique.**

The CE technique is proposed to classify the changes in gait pattern based on walking speeds, with the collected data provided by a Kinect camera during a walk test. In Cross Validation, the dataset was divided into 10 folds for testing unseen data to predict the appropriate class

among three possible classes. Then the performance of (CE) technique in gait data classification was evaluated and compared to the set number of supervisor classifiers, where the accuracy of data classification reached the highest in case of CE technique.

6.3 Contributions of the Research

In short, the significant contributions of this thesis can be summarised based on the stages of the implemented system as follows:

- **In smoothing and filtering data stage**

The 3D skeleton data of the lower body's joints has been smoothed using six different approaches. The best smoothing technique is identified by using correlation coefficients to calculate the time delay and by fitting curve approaches to a defined level of response for filtered data to the input data of each filter.

- **In gait features extraction stage**

Investigation of the effectiveness of Amplitude Modulation technique in quality improvement of extracted gait features compared to spatiotemporal gait analysis in the case of a low-data rate of sensing device; where the gait cycle determination was avoided during the features extraction process in case of the Amplitude Modulation.

- **In gait data classification stage**

Combining AM and CE in a full system of gait analysis called (AM/CE) to detect and rank the gait pattern changes based on gait speed; where the convolutional encoder is used as a classifier, by developing its similarity decision to be based on weighting the bits' position of Hamming Distance length.

6.4 Research Limitations

- **The adjustment of sampling frequency rate**

The proposed system (AM/CE) needs the adjustment or tuning to define the value of sampling frequency that satisfies the symmetrical shape of the reference signal spectrum.

- **Number of participants**

The large size of samples data can improve the level of measurements accuracy. This study recruited 40 participants which can be considered a relatively small rate compared to other studies.

- **Differences in the heights and weights of the participants**

the dataset is analysed based on gait speed without taking into consideration differences in the height and weight of the participants, which could result in variation, as tall legs can be faster than those that are shorter.

- **Numbers of classes**

the numbers of classes in the classification stage can be considered as another limitation - a wider range of ranking could be more valuable in the explanation of gait pattern changes.

- **Collecting data in long term**

A lack of long term collected data monitoring limits investigation into the changes of walk patterns in the system.

6.5 Future work

- **Using the AM/CE system for patient data**

The results presented the effectiveness of the AM/CE system in the extraction and classification of the gait features, with the AM/CE system demonstrating promising use for the ranking of disease progress such as Multiple Sclerosis (MS). This can be achieved through the investigation of the clinical scale known as the EDSS scale. This scale has ten levels of walking ability starting from zero, which is normal walk, and ending with ten, which is death due to MS (see Table 1 in Appendix A). However, the focus will be on the ranges from zero to 4.5 of the EDSS scale, where the person can walk without any aids.

- **Kinematic angles of joint data**

Angular skeleton data of the lower body plays a vital role in gait analysis for discrimination between normal and abnormal gait. This kind of data can be used for the extraction of gait features which may prove valuable information when responding to the changes in gait pattern.

- **Using feature selection approach**

Feature selection algorithms can be used for reducing the dimensionality of extracted gait features. This can be performed by replacing the use of feature reduction algorithms. A comparison between both techniques can be obtained by changing the classification accuracy.

- **Using unsupervised classifiers**

A data classification solution that splits the dataset into both a training and testing subset of data to predict the classes by labelling the samples, is called supervised. However, an unsupervised classifier clusters data into groups based on distance measurements without the need to label the samples. Algorithms include the k-mean cluster.

- **Non-Kinect camera**

A vision system with a higher data rate might be used for the proposed system, as a Kinect camera does suffer from some limitations that affect the measurement accuracy, due to the low data rate and noisy data generated during the tracking of gait movement. Consequently, the noise filtering process increases the cost of computation and processing time.

6.6 Summary

This chapter has discussed the main body of the research that has been carried out, as well as work that could be done in the future, in the following sections: objectives, contributions, limitations and future work. It has also reviewed the main achievements by describing the approaches that have been analysed to obtain data to address the research aim.

REFERENCES

- Adjeisah, M., Yang, Y., & Li, L. (2015, August). Joint Filtering: Enhancing gesture and mouse movement in Microsoft Kinect application. In *Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015 12th International Conference on (pp. 2528-2532). IEEE.
- Ahmed, F., Polash Paul, P., & Gavrilova, M. L. (2015). Kinect-based gait recognition using sequences of the most relevant joint relative angles. *Journal of WSCG*, 23(2)
- Aitpayev, K., & Gaber, J. (2012, October). Collision Avatar (CA): Adding collision objects for human body in augmented reality using Kinect. In *Application of Information and Communication Technologies (AICT)*, 2012 6th International Conference on (pp. 1-4). IEEE.
- Allen, Diane D., Susan E. Bennett, Kathleen Brandfass, P. A. Pittsburg, N. J. Stratford, Gail L. Widener, and M. I. Flint. "Task Force Members Kirsten Potter, PT, DPT, MS, NCS--Chair Northwestern University Chicago, IL."
- Al-Yaman, F. (2004). Population division department of economic and social affairs United Nations MORTPAK for windows version 4.0. *Journal of Population Research*, 4.
- Andersson, V. O., & de Araújo, R. M. (2015, January). Person Identification Using Anthropometric and Gait Data from Kinect Sensor. In *AAAI* (pp. 425-431).
- Ardestani, M. M., Ferrigno, C., Moazen, M., & Wimmer, M. A. (2016). From normal to fast walking: impact of cadence and stride length on lower extremity joint moments. *Gait & posture*, 46, 118-125.
- Auvinet, E., Multon, F., & Meunier, J. (2015). New lower-limb gait asymmetry indices based on a depth camera. *Sensors*, 15(3), 4605-4623.
- Auvinet, E., Multon, F., Aubin, C. E., Meunier, J., & Raison, M. (2015). Detection of gait cycles in treadmill walking using a Kinect. *Gait & posture*, 41(2), 722-725.
- Bassey, E., Whalley, J., & Sallis, P. (2014, July). An evaluation of smoothing filters for gas sensor signal cleaning. In *Proceedings of the Fourth International Conference on Advanced Communications and Computation*, Paris, France(pp. 20-24).
- Babak Taati PhD, P., & Alex Mihailidis PhD, P. (2014). Vision-based approach for long-term mobility monitoring: Single case study following total hip replacement. *Journal of rehabilitation research and development*, 51(7), 1165.

- Barmpoutis, A. (2013). Tensor body: Real-time reconstruction of the human body and avatar synthesis from RGB-D. *IEEE transactions on cybernetics*, 43(5), 1347-1356.
- Begg, R. K., Palaniswami, M., & Owen, B. (2005). Support vector machines for automated gait classification. *IEEE transactions on Biomedical Engineering*, 52(5), 828-838.
- Behrens, J., Pfüller, C., Mansow-Model, S., Otte, K., Paul, F., & Brandt, A. U. (2014). Using perceptive computing in multiple sclerosis-the Short Maximum Speed Walk test. *Journal of neuroengineering and rehabilitation*, 11(1), 89.
- Bei, S., Zhen, Z., Xing, Z., Taocheng, L., & Qin, L. (2018). Movement disorder detection via adaptively fused gait analysis based on Kinect sensors. *IEEE Sensors Journal*, 18(17), 7305-7314.
- Berti, E. M., Salmerón, A. J. S., & Benimeli, F. (2012). Kalman filter for tracking robotic arms using low cost 3D vision systems. In *The Fifth International Conference on Advances in Computer-Human Interactions* (pp. 236-240).
- Berwick, R. (2003). *An Idiot's guide to Support vector machines (SVMs)*. Retrieved on October, 21, 2011.
- Bonnyaud, C., Pradon, D., Vuillerme, N., Bensmail, D., & Roche, N. (2015). Spatiotemporal and kinematic parameters relating to oriented gait and turn performance in patients with chronic stroke. *PloS one*, 10(6), e0129821.
- Bouten, C. V., Koekkoek, K. T., Verduin, M., Kodde, R., & Janssen, J. D. (1997). A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE transactions on biomedical engineering*, 44(3), 136-147.
- Bravo, J., Hervás, R., Fontecha, J., & González, I. (2018). m-Health: Lessons Learned by m-Experiences. *Sensors*, 18(5), 1569.
- Bray, J. (2001). *Markerless based human motion capture: a survey*. Department Systems Engineering Brunel University.
- Brown, G. C. (2015). Living too long: The current focus of medical research on increasing the quantity, rather than the quality, of life is damaging our health and harming the economy. *EMBO reports*, 16(2), 137-141.
- Campbell, N. A., & Atchley, W. R. (1981). The geometry of canonical variate analysis. *Systematic Biology*, 30(3), 268-280.

- Cai, S., Wu, Y., Xiang, N., Zhong, Z., He, J., Shi, L., & Xu, F. (2012, August). Detrending knee joint vibration signals with a cascade moving average filter. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE* (pp. 4357-4360). IEEE.
- Casiez, G., Roussel, N., & Vogel, D. (2012, May). 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2527-2530). ACM.
- Choi, J., Park, J., Lee, B. I., Yoo, S., Kim, H., Jang, W., ... & Oh, E. (2019). The Correlation between Cognition Screening Scores and Gait Status from Three-Dimensional Gait Analysis. *Journal of Clinical Neurology*, 15(2), 152-158.
- Clark, R. A., Bower, K. J., Mentiplay, B. F., Paterson, K., & Pua, Y. H. (2013). Concurrent validity of the Microsoft Kinect for assessment of spatiotemporal gait variables. *Journal of biomechanics*, 46(15), 2722-2725.
- Clark, R. A., Vernon, S., Mentiplay, B. F., Miller, K. J., McGinley, J. L., Pua, Y. H., ... & Bower, K. J. (2015). Instrumenting gait assessment using the Kinect in people living with stroke: reliability and association with balance tests. *Journal of neuroengineering and rehabilitation*, 12(1), 15.
- Cleveland, W. S., & Grosse, E. (1991). Computational methods for local regression. *Statistics and Computing*, 1(1), 47-62.
- Cleveland, W. S., & Loader, C. (1996). Smoothing by local regression: Principles and methods. In *Statistical theory and computational aspects of smoothing* (pp. 10-49). Physica-Verlag HD.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.
- Coyle, E. J., Gabbouj, M., & Lin, J. H. (1991, June). From median filters to optimal stack filtering. In *Circuits and Systems, 1991., IEEE International Symposium on* (pp. 9-12). IEEE.
- Delahunt, E., Monaghan, K., & Caulfield, B. (2007). Ankle function during hopping in subjects with functional instability of the ankle joint. *Scandinavian journal of medicine & science in sports*, 17(6), 641-648.
- Dikovski, B., Madjarov, G., & Gjorgjevikj, D. (2014, May). Evaluation of different feature sets for gait recognition using skeletal data from Kinect. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on* (pp. 1304-1308). IEEE.

- Dolatabadi, E., Taati, B., & Mihailidis, A. (2016). Concurrent validity of the Microsoft Kinect for Windows v2 for measuring spatiotemporal gait parameters. *Medical engineering & physics*, 38(9), 952-958.
- Dramé, K., Mougin, F., & Diallo, G. (2014, October). A k-nearest neighbor based method for improving large scale biomedical document indexing. In 6th International Symposium on Semantic Mining in Biomedicine (SMBM) (pp. 19-26).
- Dubois, A., Bihl, T., & Bresciani, J. P. (2017). Automating the Timed Up and Go Test Using a Depth Camera. *Sensors*, 18(1), 14.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. 2nd. Edition. New York, 55.
- Duong, S., & Choi, M. H. (2013). Interactive Full-Body Motion Capture Using Infrared Sensor Network. *International Journal of Computer Graphics & Animation*, 3(4), 41.
- Edwards, M., & Green, R. (2014, November). Low-latency filtering of kinect skeleton data for video game control. In Proceedings of the 29th International Conference on Image and Vision Computing New Zealand (pp. 190-195).ACM.
- Elkurdi, A., Caliskanelli, I., & Nefti-Meziani, S. (2018, March). Using Amplitude Modulation for Extracting Gait Features. In ICT4AWE (pp. 161-168).
- Elkurdi, A., Soufian, M., & Nefti-Meziani, S. (2018). Gait speeds classifications by supervised modulation based machinelearning using Kinect camera. *Journal of Medical Research and Innovations*, 2(4), 1-6.
- Eskofier, B. M., Federolf, P., Kugler, P. F., & Nigg, B. M. (2013). Marker-based classification of young–elderly gait pattern differences via direct PCA feature extraction and SVMs. *Computer Methods in Biomechanics and Biomedical Engineering*, 16(4), 435-442.
- Farah, J. D., Baddour, N., & Lemaire, E. D. (2019). Design, development, and evaluation of a local sensor-based gait phase recognition system using a logistic model decision tree for orthosis-control. *Journal of neuroengineering and rehabilitation*, 16(1), 22.
- Faruque, S. (2017). *Radio frequency modulation made Easy*. Springer.
- Fern'ndez-Baena, A., Susín, A., & Lligadas, X. (2012, September). Biomechanical validation of upper-body and lower-body joint movements of kinect motion capture data for rehabilitation treatments. In Intelligent networking and collaborative systems (INCoS), 2012 4th international conference on (pp. 656-661). IEEE.

- Feuerstein, D., Parker, K. H., & Boutelle, M. G. (2009). Practical methods for noise removal: applications to spikes, nonstationary quasi-periodic noise, and baseline drift. *Analytical chemistry*, 81(12), 4987-4994.
- Fischer, J. S., Rudick, R. A., Cutter, G. R., Reingold, S. C., & National MS Society Clinical Outcomes Assessment Task Force. (1999). The Multiple Sclerosis Functional Composite measure (MSFC): an integrated approach to MS clinical outcome assessment. *Multiple Sclerosis Journal*, 5(4), 244-250.
- Foch, E., & Milner, C. E. (2014). The influence of iliotibial band syndrome history on running biomechanics examined via principal components analysis. *Journal of biomechanics*, 47(1), 81-86.
- Freedman, M. S., Patry, D. G., Grand'Maison, F., Myles, M. L., Paty, D. W., & Selchen, D. H. (2004). Treatment optimization in multiple sclerosis. *Canadian journal of neurological sciences*, 31(2), 157-168.
- Freeman, R. L. (2005). *Fundamentals of telecommunications*(Vol. 92). John Wiley & Sons.
- Fried, A. V., Cwikel, J., Ring, H., & Galinsky, D. (1990). ELGAM—Extra-laboratory gait assessment method: Identification of risk factors for falls among the elderly at home. *International disability studies*, 12(4), 161-164.
- Fukuchi, C. A., Fukuchi, R. K., & Duarte, M. (2019). Effects of walking speed on gait biomechanics in healthy participants: a systematic review and meta-analysis. *Systematic Reviews*, 8(1), 153.
- Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing*, 21(2), 137-146.
- Gabel, M., Gilad-Bachrach, R., Renshaw, E., & Schuster, A. (2012, August). Full body gait analysis with Kinect. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE* (pp. 1964-1967). IEEE.
- Galna, B., Barry, G., Jackson, D., Mhiripiri, D., Olivier, P., & Rochester, L. (2014). Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease. *Gait & posture*, 39(4), 1062-1068.
- Gamble, L., Ravela, S., & McGarigal, K. (2008). Multi-scale features for identifying individuals in large biological databases: an application of pattern recognition technology to the marbled salamander *Ambystoma opacum*. *Journal of Applied Ecology*, 45(1), 170-180.

- Garimella, R. V. (2017). A Simple Introduction to Moving Least Squares and Local Regression Estimation (No. LA-UR-17-24975). Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
- Gavrilov, L. A., & Heuveline, P. (2003). Aging of population. *The encyclopedia of population*, 1, 32-37.
- Geerse, D. J., Coolen, B. H., & Roerdink, M. (2015). Kinematic validation of a multi-Kinect v2 instrumented 10-meter walkway for quantitative gait assessments. *PloS one*, 10(10), e0139913.
- Ghoussayni, S., Stevens, C., Durham, S., & Ewins, D. (2004). Assessment and validation of a simple automated method for the detection of gait events and intervals. *Gait & Posture*, 20(3), 266-272.
- Goot, R. E., Mahalab, U., & Cohen, R. (2005). Nonlinear exponential smoothing (NLES) algorithm for noise filtering and edge preservation. *HAIT Journal of Science and Engineering*, 2, 2-10.
- Guerra, L., McGarry, L. M., Robles, V., Bielza, C., Larranaga, P., & Yuste, R. (2011). Comparison between supervised and unsupervised classifications of neuronal cell types: a case study. *Developmental neurobiology*, 71(1), 71-82.
- Guo, Q., & Jiang, D. (2015). Method for walking gait identification in a lower extremity exoskeleton based on C4. 5 decision tree algorithm. *International Journal of Advanced Robotic Systems*, 12(4), 30.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar), 1157-1182.
- Hajian-Tilaki, K. (2013). Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. *Caspian journal of internal medicine*, 4(2), 627.
- Han, F., Reily, B., Hoff, W., & Zhang, H. (2017). Space-time representation of people based on 3D skeletal data: A review. *Computer Vision and Image Understanding*, 158, 85-105.
- Hansun, S. E. N. G. (2016). A New Approach of Brown's Double Exponential Smoothing Method in Time Series Analysis. *Balkan Journal of Electrical & Computer Engineering*, 4(2), 75-78.
- Hastie, T., James, G., Tibshirani, R., & Witten, D. (2013). An introduction to statistical learning with applications in R.
- Harres, D. (2013). Median filter: An efficient way to remove impulse noise. *EDN Network*

- Hauser, S. L., Dawson, D. M., Leirich, J. R., Beal, M. F., Kevy, S. V., Propper, R. D., ... & Weiner, H. L. (1983). Intensive immunosuppression in progressive multiple sclerosis: a randomized, three-arm study of high-dose intravenous cyclophosphamide, plasma exchange, and ACTH. *New England Journal of Medicine*, 308(4), 173-180.
- Hawes, C., Phillips, C. D., Rose, M., Holan, S., & Sherman, M. (2003). A national survey of assisted living facilities. *The Gerontologist*, 43(6), 875-882.
- Helal, A., & Abdulrazak, B. (2006). TeCaRob: Tele-care using telepresence and robotic technology for assisting people with special needs. *International Journal of ARM*, 7(3), 46-53.
- Helmreich, J. E. (2016). Regression Modeling Strategies with Applications to Linear Models, Logistic and Ordinal Regression and Survival Analysis. *Journal of Statistical Software*, 70(b02).
- Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*, 31(5), 647-663.
- Humphries, R., Thorlby, R., Holder, H., Hall, P., & Charles, A. (2016). *Social care for older people*. London: The King's Fund and Nuffield Trust.
- Hubbard, E. A., Wetter, N. C., Sutton, B. P., Pilutti, L. A., & Motl, R. W. (2016). Diffusion tensor imaging of the corticospinal tract and walking performance in multiple sclerosis. *Journal of the neurological sciences*, 363, 225-231.
- Ismail, S. I., Osman, E., Sulaiman, N., & Adnan, R. (2016). Comparison between Marker-less Kinect-based and Conventional 2D Motion Analysis System on Vertical Jump Kinematic Properties Measured from Sagittal View. In *Proceedings of the 10th International Symposium on Computer Science in Sports (ISCSS)* (pp. 11-17). Springer, Cham.
- Isnanto, R. (2011). Comparison on several smoothing methods in nonparametric regression. *Jurnal Sistem Komputer*, 1(1), 41-48.
- Jarchi, D., Pope, J., Lee, T. K., Tamjidi, L., Mirzaei, A., & Sanei, S. (2018). A review on accelerometry-based gait analysis and emerging clinical applications. *IEEE reviews in biomedical engineering*, 11, 177-194.
- Jiang, S., Wang, Y., Zhang, Y., & Sun, J. (2014, November). Real time gait recognition system based on kinect skeleton feature. In *Asian Conference on Computer Vision* (pp. 46-57). Springer, Cham.

- Jordan, K., Challis, J. H., & Newell, K. M. (2007). Walking speed influences on gait cycle variability. *Gait & posture*, 26(1), 128-134.
- Kale Moyano, R. (2015). Human gait characterization using Kinect.
- Khan, W., & Badii, A. (2019). Pathological Gait Abnormality Detection and Segmentation by Processing the Hip Joints Motion Data to Support Mobile Gait Rehabilitation. *Research in C Medical & Engineering Sciences*, 7(03).
- Kharb, A., Saini, V., Jain, Y. K., & Dhiman, S. (2011). A review of gait cycle and its parameters. *IJCEM International Journal of Computational Engineering & Management*, 13, 78-83.
- Kieseier, B. C., & Pozzilli, C. (2012). Assessing walking disability in multiple sclerosis. *Multiple Sclerosis Journal*, 18(7), 914-924.
- Kleinberger, T., Becker, M., Ras, E., Holzinger, A., & Müller, P. (2007, July). Ambient intelligence in assisted living: enable elderly people to handle future interfaces. In *International conference on universal access in human-computer interaction* (pp. 103-112). Springer, Berlin, Heidelberg.
- Kim, C. J., & Son, S. M. (2014). Comparison of spatiotemporal gait parameters between children with normal development and children with diplegic cerebral palsy. *Journal of physical therapy science*, 26(9), 1317-1319.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Paper presented at the Ijcai.
- Konstorum, A., Vidal, E., Jekel, N., & Laubenbacher, R. (2018). Comparative Analysis of Linear and Nonlinear Dimension Reduction Techniques on Mass Cytometry Data. *bioRxiv*, 273862.
- Koswatta, R., & Karmakar, N. C. (2010, December). Moving average filtering technique for signal processing in digital section of UWB chipless RFID reader. In *Microwave Conference Proceedings (APMC), 2010 Asia-Pacific* (pp. 1304-1307). IEEE.
- Kumar, G., & Bhatia, P. K. (2014, February). A detailed review of feature extraction in image processing systems. In *Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference on* (pp. 5-12). IEEE.
- Kumar, R., & Indrayan, A. (2011). Receiver operating characteristic (ROC) curve for medical researchers. *Indian pediatrics*, 48(4), 277-287.
- Kung, S. Y. (2014). *Kernel Methods and Machine Learning*. United Kingdom: Cambridge University.

- Kurtzke, J. F. (1983). Rating neurologic impairment in multiple sclerosis: an expanded disability status scale (EDSS). *Neurology*, 33(11), 1444-1444.
- Lachat, E., Macher, H., Mittet, M. A., Landes, T., & Grussenmeyer, P. (2015). First experiences with Kinect v2 sensor for close range 3D modelling. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5), 93.
- Lau, H. Y., Tong, K. Y., & Zhu, H. (2008). Support vector machine for classification of walking conditions using miniature kinematic sensors. *Medical & biological engineering & computing*, 46(6), 563-573.
- Law, H. C. (2006). Clustering, dimensionality reduction, and side information. Retrieved from
- Leigh Hollands, K., Hollands, M. A., Zietz, D., Miles Wing, A., Wright, C., & Van Vliet, P. (2010). Kinematics of turning 180 during the timed up and go in stroke survivors with and without falls history. *Neurorehabilitation and Neural Repair*, 24(4), 358-367.
- Leightley, D., Yap, M. H., Coulson, J., Barnouin, Y., & McPhee, J. S. (2015, December). Benchmarking human motion analysis using kinect one: An open source dataset. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific* (pp. 1-7). IEEE.
- Lim, D., Kim, C., Jung, H., Jung, D., & Chun, K. J. (2015). Use of the Microsoft Kinect system to characterize balance ability during balance training. *Clinical interventions in aging*, 10, 1077.
- Liu, E. (2004). Convolutional coding & Viterbi algorithm. *IEEE S*, 11-16.
- Liu, T., Inoue, Y., & Shibata, K. (2009). Development of a wearable sensor system for quantitative gait analysis. *Measurement*, 42(7), 978-988.
- Liu, Y., Liu, C., & Wang, D. (2008). A 1D time-varying median filter for seismic random, spike-like noise elimination. *Geophysics*, 74(1), V17-V24.
- Lombardi, A., Ferri, M., Rescio, G., Grassi, M., & Malcovati, P. (2009, October). Wearable wireless accelerometer with embedded fall-detection logic for multi-sensor ambient assisted living applications. In *Sensors, 2009 IEEE* (pp. 1967-1970). IEEE.
- Loumponias, K., Vretos, N., Daras, P., & Tsaklidis, G. (2016). Using kalman filter and tobit kalman filter in order to improve the motion recorded by kinect sensor ii. In *Proceedings of the 29th Panhellenic Statistics Conference* (pp. 000-000).

- Ma, M., Xu, F., & Liu, Y. (2011, December). Animation of 3D characters from single depth camera. In 3D Imaging (IC3D), 2011 International Conference on (pp. 1-4). IEEE.
- Maaten, L. V. d., Postma, E., & Herik, H. V. d. (2009). Dimensionality reduction: A comparative review. Retrieved from
- MacIntosh, E., Rajakulendran, N., & Salah, H. (2014). Transforming Health: Towards Decentralized and Connected Care. MaRS Market Insights.
- Maini, A. K., & Agrawal, V. (2011). *Satellite technology: principles and applications*. John Wiley & Sons.
- Malekmohamadi, H., Moemeni, A., Orun, A., & Kumar, J. (2018). Low-Cost Automatic Ambient Assisted Living System.
- Mariani, B., Rouhani, H., Crevoisier, X., & Aminian, K. (2013). Quantitative estimation of foot-flat and stance phase of gait using foot-worn inertial sensors. *Gait & posture*, 37(2), 229-234.
- Maurer, C., Federolf, P., von Tscharner, V., Stirling, L., & Nigg, B. M. (2012). Discrimination of gender-, speed-, and shoe-dependent movement patterns in runners using full-body kinematics. *Gait & posture*, 36(1), 40-45.
- McCrum, C., Lucieer, F., Van De Berg, R., Willems, P., Fornos, A. P., Guinand, N., ... & Meijer, K. (2018). Is faster always better? The walking speed-dependency of gait variability in bilateral vestibulopathy. *BioRxiv*, 413955.
- McGough, E. L., Logsdon, R. G., Kelly, V. E., & Teri, L. (2013). Functional mobility limitations and falls in assisted living residents with dementia: physical performance assessment and quantitative gait analysis. *Journal of geriatric physical therapy*, 36(2), 78-86.
- Mendez, C. G. M., Mendez, S. H., Solis, A. L., Figueroa, H. V. R., & Hernandez, A. M. (2017, November). The Effects of Using a Noise Filter and Feature Selection in Action Recognition: An Empirical Study. In *Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, 2017 International Conference on (pp. 43-48). IEEE.
- Mentiplay, B. F., Perraton, L. G., Bower, K. J., Pua, Y. H., McGaw, R., Heywood, S., & Clark, R. A. (2015). Gait assessment using the Microsoft Xbox One Kinect: Concurrent validity and inter-day reliability of spatiotemporal and kinematic variables. *Journal of biomechanics*, 48(10), 2166-2170.
- Milovanović, I., & Popović, D. B. (2012). Principal component analysis of gait kinematics data in acute and chronic stroke patients. *Computational and mathematical methods in medicine*, 2012.

- Monekosso, D. N., Florez-Revuelta, F., & Remagnino, P. (2015). Guest editorial special issue on ambient-assisted living: Sensors, methods, and applications. *IEEE Transactions on Human-Machine Systems*, 45(5), 545-549.
- Moon, S., Park, Y., Ko, D. W., & Suh, I. H. (2016). Multiple kinect sensor fusion for human skeleton tracking using Kalman filtering. *International Journal of Advanced Robotic Systems*, 13(2), 65.
- Moon, Y., McGinnis, R. S., Seagers, K., Motl, R. W., Sheth, N., Wright Jr, J. A., ... & Sosnoff, J. J. (2017). Monitoring gait in multiple sclerosis with novel wearable motion sensors. *PLoS One*, 12(2), e0171346.
- Moons, T., Van Gool, L. and Vergauwen, M., 2009. 3D reconstruction from multiple images, Part 1: Principles. Now Publishers Inc.
- Morrison, A. M. (2005). Receiver Operating Characteristic (ROC) Curve Analysis of Antecedent Rainfall and the Alewife/Mystic River Receiving Waters: Massachusetts Water Resources Authority, Environmental Quality Department
- Morrison, A. M., Coughlin, K., Shine, J. P., Coull, B. A., & Rex, A. C. (2003). Receiver operating characteristic curve analysis of beach water quality indicator variables. *Applied and environmental microbiology*, 69(11), 6405-6411.
- Mulder, A. (1994). Human movement tracking technology. Simon Fraser University School of Kinesiology Technical Report, 94-1.
- Neustein, A. (Ed.). (2011). *Springer Briefs in Electrical and Computer Engineering: Speech Technology*. Springer.
- Nguyen, C. V., Izadi, S., & Lovell, D. (2012, October). Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, 2012 Second International Conference on (pp. 524-530). IEEE.
- Nguyen, T. N., Huynh, H. H., & Meunier, J. (2016). Skeleton-based abnormal gait detection. *Sensors*, 16(11), 1792.
- Nieto-Hidalgo, M., Ferrández-Pastor, F. J., Valdivieso-Sarabia, R. J., Mora-Pascual, J., & García-Chamizo, J. M. (2018). Gait Analysis Using Computer Vision Based on Cloud Platform and Mobile Device. *Mobile Information Systems*, 2018.
- Nigg, B. M., Baltich, J., Maurer, C., & Federolf, P. (2012). Shoe midsole hardness, sex and age effects on lower extremity kinematics during running. *Journal of biomechanics*, 45(9), 1692-1697.

- Nurunnabi, A., West, G., & Belton, D. (2013). Robust locally weighted regression for ground surface extraction in mobile laser scanning 3D data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 1, 217-222.
- Otte, K., Kayser, B., Mansow-Model, S., Verrel, J., Paul, F., Brandt, A. U., & Schmitz-Hübsch, T. (2016). Accuracy and reliability of the kinect version 2 for clinical measurement of motor function. *PloS one*, 11(11), e0166532.
- Papageorgiou, X. S., Chalvatzaki, G., Tzafestas, C. S., & Maragos, P. (2014, May). Hidden markov modeling of human normal gait using laser range finder for a mobility assistance robot. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (pp. 482-487). IEEE.
- Peterkova, A., & Stremy, M. (2015, September). Obtaining the gait parameters from Kinect sensor for the person identification. In *Intelligent Engineering Systems (INES), 2015 IEEE 19th International Conference on* (pp. 337-340). IEEE.
- Pfister, A., West, A. M., Bronner, S., & Noah, J. A. (2014). Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis. *Journal of medical engineering & technology*, 38(5), 274-280.
- Phinyomark, A., Petri, G., Ibáñez-Marcelo, E., Osis, S. T., & Ferber, R. (2018). Analysis of big data in gait biomechanics: current trends and future directions. *Journal of medical and biological engineering*, 38(2), 244-260.
- Phommahavong, S., Haas, D., Yu, J., Krüger-Ziolek, S., Möller, K., & Kretschmer, J. (2015). Evaluating the microsoft kinect skeleton joint tracking as a tool for home-based physiotherapy. *Current Directions in Biomedical Engineering*, 1(1), 184-187.
- Podsiadlo, D., & Richardson, S. (1991). The timed "Up & Go": a test of basic functional mobility for frail elderly persons. *Journal of the American geriatrics Society*, 39(2), 142-148.
- Potter, K., Cohen, E. T., Allen, D. D., Bennett, S. E., Brandfass, K. G., Widener, G. L., & Yorke, A. M. (2014). Outcome measures for individuals with multiple sclerosis: recommendations from the American Physical Therapy Association Neurology Section Task Force. *Physical therapy*, 94(5), 593-608.
- Putz-Leschczynska, J., & Granacki, M. (2014, October). Gait biometrics with a Microsoft Kinect sensor. In *Security Technology (ICCST), 2014 International Carnahan Conference on* (pp. 1-5). IEEE.

- Qiu, G. (1994). Functional optimization properties of median filtering. *IEEE Signal Processing Letters*, 1(4), 64-65.
- Raposo, C., Barreto, J. P., & Nunes, U. (2013, June). Fast and accurate calibration of a kinect sensor. In *2013 International Conference on 3D Vision (3DV)* (pp. 342-349). IEEE.
- Raudys, S. J., & Jain, A. K. (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence*, 13(3), 252-264.
- Ries, J. D., Echternach, J. L., Nof, L., & Gagnon Blodgett, M. (2009). Test-retest reliability and minimal detectable change scores for the timed “up & go” test, the six-minute walk test, and gait speed in people with Alzheimer disease. *Physical therapy*, 89(6), 569-579.
- Robertson, K. R., Parsons, T. D., Sidtis, J. J., Hanlon Inman, T., Robertson, W. T., Hall, C. D., & Price, R. W. (2006). Timed Gait test: normative data for the assessment of the AIDS dementia complex. *Journal of clinical and experimental neuropsychology*, 28(7), 1053-1064.
- Romo-Cárdenas, G., Avilés-Rodríguez, G. J., Sánchez-López, J. D. D., Cosío-León, M., Luque, P. A., Gómez-Gutiérrez, C. M., ... & Navarro-Cota, C. X. (2018). Nyquist-Shannon theorem application for Savitzky-Golay smoothing window size parameter determination in bio-optical signals. *Results in Physics*, 11, 17-22.
- Rosenblatt, A., Samus, Q. M., Steele, C. D., Baker, A. S., Harper, M. G., Brandt, J., ... & Lyketsos, C. G. (2004). The Maryland Assisted Living Study: Prevalence, recognition, and treatment of dementia and other psychiatric disorders in the assisted living population of central Maryland. *Journal of the American Geriatrics Society*, 52(10), 1618-1625.
- Savitzky, A., & Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8), 1627-1639.
- Selfridge, O. G. (1955, March). Pattern recognition and modern computers. In *Proceedings of the March 1-3, 1955, western joint computer conference* (pp. 91-93). ACM.
- Schafer, R. W. (2011, January). On the frequency-domain properties of Savitzky-Golay filters. In *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE)*, 2011 IEEE (pp. 54-59). IEEE.
- Schwartz, D. (2012). Prediction of lysine post-translational modifications using bioinformatic tools. *Essays in biochemistry*, 52, 165-177.

- Schwid, S. R., Goodman, A. D., Mattson, D. H., Mihai, C., Donohoe, K. M., Petrie, M. D., ... & McDermott, M. P. (1997). The measurement of ambulatory impairment in multiple sclerosis. *Neurology*, 49(5), 1419-1424.
- Shajeesh, K. U., Kumar, S., Pravena, D., & Soman, K. P. (2012). Speech enhancement based on Savitzky-Golay smoothing filter. *International Journal of Computer Applications*, 57(21).
- Shingade, A., & Ghotkar, A. (2014). Animation of 3D human model using markerless motion capture applied to sports. arXiv preprint arXiv:1402.2363.
- Shumway-Cook, A., & Woollacott, M. H. (2007). *Motor control: translating research into clinical practice*. Lippincott Williams & Wilkins.
- Shumway-Cook, A., Brauer, S., & Woollacott, M. (2012). Timed-up & go test. *UB Physical Therapy*.
- Slijepcevic, D., Zeppelzauer, M., Gorgas, A. M., Schwab, C., Schüller, M., Baca, A., ... & Horsak, B. (2017). Automatic Classification of Functional Gait Disorders. *IEEE Journal of Biomedical and Health Informatics*.
- Soufian M., S. Nefti- Meziani and J. Drake (2020). Toward Kinecting Cognition by Behaviour Recognition based Deep Learning and Big Data, accepted and under publication in *Journal of Universal Access in the Information Society*, Special issue: ICTs for cognitively impaired older adults, Springer, 2020.
- Stueltjens, M. P. M., Dekker, J., Van Baar, M. E., Oostendorp, R. A. B., & Bijlsma, J. W. J. (2000). Range of joint motion and disability in patients with osteoarthritis of the knee or hip. *Rheumatology*, 39(9), 955-961.
- Straudi, S., Martinuzzi, C., Pavarelli, C., Charabati, A. S., Benedetti, M. G., Foti, C., ... & Basaglia, N. (2014). A task-oriented circuit training in multiple sclerosis: a feasibility study. *BMC neurology*, 14(1), 124.
- Suguna, N., & Thanushkodi, K. (2010). An improved k-nearest neighbor classification using genetic algorithm. *International Journal of Computer Science Issues*, 7(2), 18-21.
- Sullivan, J., Eriksson, M., Carlsson, S., & Liebowitz, D. (2002). Automating multiview tracking and reconstruction of human motion. In *European Conference on Computer Vision*.
- Swets, J. A. (1979). ROC analysis applied to the evaluation of medical imaging techniques. *Invest. Radiol.* v14 i2, 09-121.

- Takeda, R., Tadano, S., Todoh, M., Morikawa, M., Nakayasu, M., & Yoshinari, S. (2009). Gait analysis using gravitational acceleration measured by wearable sensors. *Journal of biomechanics*, 42(3), 223-233.
- Takeda, R., Tadano, S., Natorigawa, A., Todoh, M., & Yoshinari, S. (2009). Gait posture estimation using wearable acceleration and gyro sensors. *Journal of biomechanics*, 42(15), 2486-2494.
- Teodorescu, P. P. (2007). *Mechanical Systems, Classical Models: Volume 1: Particle Mechanics*. Springer Science & Business Media.
- Thongsook, A., Nunthawarasilp, T., Kraypet, P., Lim, J., & Ruangpayoongsak, N. (2019, January). C4. 5 Decision Tree against Neural Network on Gait Phase Recognition for Lower Limb Exoskeleton. In *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)* (pp. 69-72). IEEE.
- Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov), 45-66.
- Tong, X., Xu, P., & Yan, X. (2012, October). Research on skeleton animation motion data based on kinect. In *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on* (Vol. 2, pp. 347-350). IEEE.
- Tunca, C., Pehlivan, N., Ak, N., Arnrich, B., Salur, G., & Ersoy, C. (2017). Inertial sensor-based robust gait analysis in non-hospital settings for neurological disorders. *Sensors*, 17(4), 825.
- Ťupa, O., Procházka, A., Vyšata, O., Schätz, M., Mareš, J., Vališ, M., & Mařík, V. (2015). Motion tracking and gait feature estimation for recognising Parkinson's disease using MS Kinect. *Biomedical engineering online*, 14(1), 97.
- Vektor, P. (2018). Kinect-Based Human Gait Recognition Using Locally Linear Embedded and Support Vector Machine. *Jurnal Kejuruteraan*, 30(2), 235-247.
- Verma, K., Singh, B. K., & Thoke, A. S. (2015). An enhancement in adaptive median filter for edge preservation. *Procedia Computer Science*, 48, 29-36.
- Vernon, S., Paterson, K., Bower, K., McGinley, J., Miller, K., Pua, Y. H., & Clark, R. A. (2015). Quantifying individual components of the timed up and go using the kinect in people living with stroke. *Neurorehabilitation and neural repair*, 29(1), 48-53.
- Wahab, Y., & Bakar, N. A. (2011, June). Gait analysis measurement for sport application based on ultrasonic system. In *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on* (pp. 20-24). IEEE.

- Wang, Q., Kurillo, G., Ofli, F., & Bajcsy, R. (2015). Remote health coaching system and human motion data analysis for physical therapy with microsoft kinect. arXiv preprint arXiv:1512.06492.
- Weber, W. E. (1836). *Mechanik der menschlichen Gehwerkzeuge: eine anatomisch-physiologische Untersuchung*. Dieterich.
- Wettayaprasit, W., Laosen, N., & Chevakidagarn, S. (2007, September). Data filtering technique for neural networks forecasting. In *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization* (pp. 225-230). World Scientific and Engineering Academy and Society (WSEAS).
- Xu, X., McGorry, R. W., Chou, L. S., Lin, J. H., & Chang, C. C. (2015). Accuracy of the Microsoft Kinect™ for measuring gait parameters during treadmill walking. *Gait & posture*, 42(2), 145-151.
- Yoo, J. H., Hwang, D., & Nixon, M. S. (2005, September). Gender classification in human gait using support vector machine. In *International Conference on Advanced Concepts for Intelligent Vision Systems* (pp. 138-145). Springer, Berlin, Heidelberg
- Yuan, C., Sun, X., & Lv, R. (2016). Fingerprint liveness detection based on multi-scale LPQ and PCA. *China Communications*, 13(7), 60-65.
- Zeng, M., Liu, Z., Meng, Q., Bai, Z., & Jia, H. (2012, October). Motion capture and reconstruction based on depth information using Kinect. In *Image and Signal Processing (CISP), 2012 5th International Congress on* (pp. 1381-1385). IEEE.
- Zeni Jr, J. A., Richards, J. G., & Higginson, J. S. (2008). Two simple methods for determining gait events during treadmill and overground walking using kinematic data. *Gait & posture*, 27(4), 710-714.
- Zeitler, E., Buys, L., Aird, R., & Miller, E. (2012). Mobility and active ageing in suburban environments: Findings from in-depth interviews and person-based GPS tracking. *Current gerontology and geriatrics research*, 2012.
- Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2), 4-10.
- Zhou, H., & Hu, H. (2008). Human motion tracking for rehabilitation—A survey. *Biomedical Signal Processing and Control*, 3(1), 1-18.
- Zhou, H., & Hu, H. (2004). *A survey-human movement tracking and stroke rehabilitation*. University of Essex, Colchester United Kingdom.

- Zhu, R., & Zhou, Z. (2004). A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. *IEEE Transactions on Neural systems and rehabilitation engineering*, 12(2), 295-302.
- Zimmerman, S., Scott, A. C., Park, N. S., Hall, S. A., Wetherby, M. M., Gruber-Baldini, A. L., & Morgan, L. A. (2003). Social engagement and its relationship to service provision in residential care and assisted living. *Social Work Research*, 27(1), 6-18.
- Zimmerman, S., & Sloane, P. D. (2007). Definition and classification of assisted living. *The Gerontologist*, 47(suppl_1), 33-39.

APPENDICES

Appendix-A

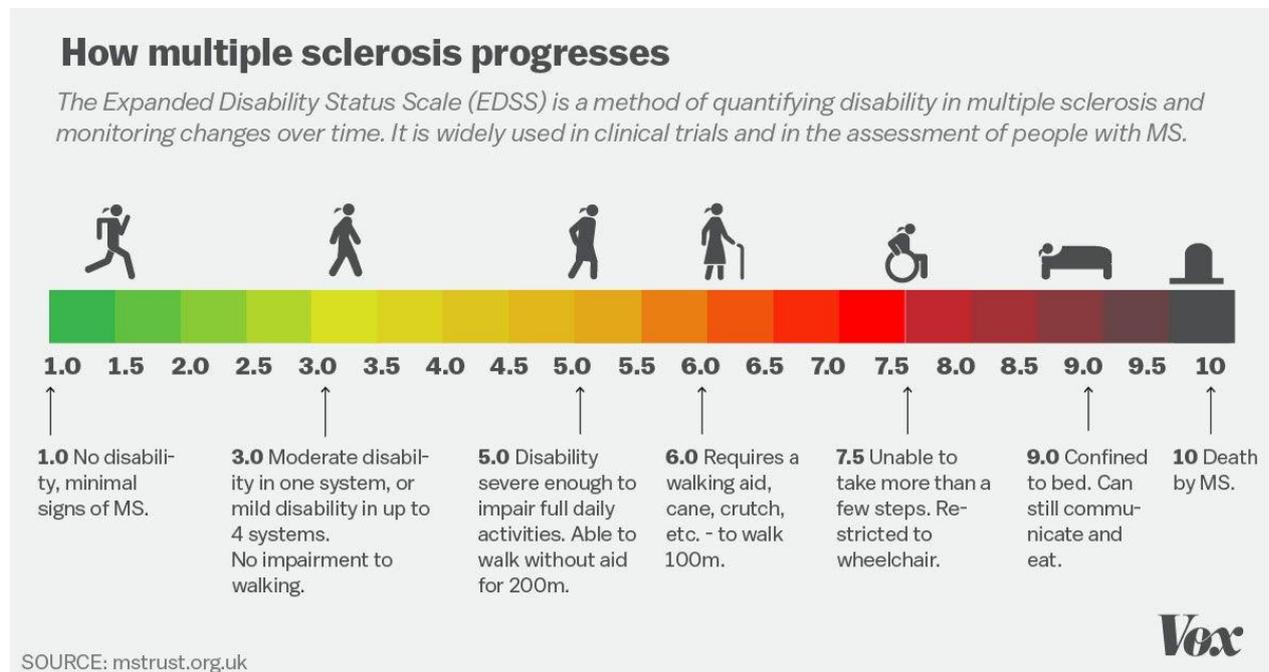


Figure A.1: Extension Disability Status Scale (EDSS) illustrates the Progression to Disability due to Multiple Sclerosis (MS) disease.

Table A.1: A 12-item Multiple Sclerosis MS Walking Scale (MSWS-12).

In the past 2 weeks, how much has your MS	Not at all	A little	moderately	Quite a bit	extremely
1. Limited your ability to walk?	1	2	3	4	5
2. Limited your ability to run?	1	2	3	4	5
3. Limited your ability to climb up and down stairs?	1	2	3	4	5
4. Made standing when doing things more difficult?	1	2	3	4	5
5. Limited your balance when standing or walking?	1	2	3	4	5
6. Limited how far you are able to walk?	1	2	3	4	5
7. Increased the effort needed for you to walk?	1	2	3	4	5
8. Made it necessary for you to use support when walking indoors (e.g., holding on to furniture, using a stick, etc.)?	1	2	3	4	5
9. Made it necessary for you to use support when walking outdoors (e.g., using a stick, a frame, etc.)?	1	2	3	4	5
10. Slowed down your walking?	1	2	3	4	5
11. Affected how smoothly you walk?	1	2	3	4	5
12. Made you concentrate on your	1	2	3	4	5

(1-5) levels, 1 means no disability, 5 means extremely disability

Table A.2: Outcomes Measures OMs for Patients with MS

Multiple Sclerosis Outcome Measures Summary Recommendations	Acute Care	In- patient Rehab	Home Health	Skilled Nursing Facility	Out- patient	EDSS 0.0 - 3.5	EDSS 4.0 - 5.5	EDSS 6.0 - 7.5	EDSS 8.0 - 9.5	Entry-level Education: Required	Entry-level Education: Exposure	Use in Research	Psychometric Testing Needed
Motion Sensitivity Test	2	2	2	2	2	2	2	2	1				x
Movement Ability Measure	2	2	2	2	2	2	2	2	2				x
Multi-Component Fatigue Scale	1	1	1	1	1	1	1	1	1				
Multiple Sclerosis Functional Composite	1	3	3	3	4	3	3	3	3		x	x	
Multiple Sclerosis Impact Scale (MSIS-29)	4	4	4	4	4	4	4	4	4		x	x	
Multiple Sclerosis International Quality of Life Question	3	3	3	3	3	3	3	3	3				
Multiple Sclerosis Quality of Life (MS-QOL 54)	3	4	4	4	4	4	4	4	4	x		x	
Multiple Sclerosis Quality of Life Inventory	1	1	1	1	3	3	3	3	3			x	
Multiple Sclerosis Spasticity Scale (MSSS - 88)	2	2	2	2	2	2	2	2	2				x
Neuropathic Pain Scale	2	2	2	2	2	2	2	2	2				x
Nottingham Sensory Assessment	1	1	1	1	1	1	1	1	1				x
Patient-specific Functional Scale	1	2	2	2	2	2	2	2	2				x
Physiologic Cost Index	1	1	1	1	1	1	1	1	1				x
Rivermead Assessment of Somatosensory Performance	1	1	1	1	1	1	1	1	1				x
Rivermead Mobility Index	3	3	3	3	3	3	3	3	3	x		x	
SARA (Ataxia Rating Scale)	2	2	2	2	2	2	2	2	2				x
Scripps Neurological Rating Scale	1	1	1	1	1	1	1	1	1				
Semmes Weinstein Monofilament	2	2	2	2	2	2	2	2	2				x
Short Form Health Survey of the Medical Outcome Study (SF-36)	1	1	3	1	3	3	3	3	1		x	x	
Static Standing Balance Test	2	2	2	2	2	2	2	2	1	x			x
Tardieu Spasticity Scale	2	2	2	2	2	2	2	2	2				x
Timed 25 foot walk	4	4	4	4	4	4	4	4	1	x		x	
Timed Up & Go (TUG) w/ Cognitive & Manual	4	4	4	4	4	4	4	4	1	x		x	
Tinetti Falls Efficacy Scale	2	2	2	2	2	2	2	2	1				x
Tinetti Performance Oriented Mobility Assessment	2	2	2	2	2	2	2	2	1				x
Trunk Control Test	2	2	2	2	1	1	1	1	2				x
Trunk Impairment Scale	3	3	3	3	1	1	3	3	3	x		x	x
Visual Analog Scale (for Fatigue)	3	3	3	3	3	3	3	3	3	x		x	

Continues Table A.2.

Multiple Sclerosis Outcome Measures Summary Recommendations	Acute Care	In- patient Rehab	Home Health	Skilled Nursing Facility	Out- patient	EDSS 0.0 - 3.5	EDSS 4.0 - 5.5	EDSS 6.0 - 7.5	EDSS 8.0 - 9.5	Entry-level Education: Required	Entry-level Education: Exposure	Use in Research	Psychometric Testing Needed
12 Minute Walk / Run	1	2	1	2	2	2	2	1	1		x		x
12-Item MS Walking Scale	4	4	4	4	4	4	4	4	1	x		x	
2 Minute Walk Test	2	2	2	2	2	2	2	2	1	x			x
5-Time Sit to Stand	2	2	2	2	2	2	2	2	1				x
6 Minute Walk Test	3	4	1	3	4	4	4	3	1	x		x	
9-Hole Peg Test	4	4	4	4	4	4	4	4	3	x		x	x
Activities-specific Balance Confidence Scale	3	3	3	3	3	3	3	3	1	x		x	
Balance Evaluation Systems Test (BESTest)	2	2	2	2	2	2	2	2	1				x
Berg Balance Scale	4	4	4	4	4	4	4	4	1	x		x	
Biothesiometer	2	2	2	2	2	2	2	2	2				x
Box & Blocks Test	3	3	3	3	3	3	3	3	3		x	x	
Brief Fatigue Index/Inventory	1	1	1	1	1	1	1	1	1				
Canadian Occupational Performance Measure	2	2	2	2	2	2	2	2	2				x
Clinical Test of Sensory Interaction in Balance	2	2	2	2	2	2	2	2	1				x
Disease Steps	3	3	3	3	3	3	3	3	3		x		
Dizziness Handicap Inventory	1	3	2	2	4	4	4	4	2	x		x	
Dynamic Gait Index	3	3	3	3	3	3	3	1	1	x		x	
Expanded Disability Status Scale & Kurtzke Functional Systems Scale	1	1	1	1	1	1	1	1	1				
Fatigue Descriptive Scale	2	2	2	2	2	2	2	2	2				x
Fatigue Scale for Motor and Cognitive Functions	3	3	3	3	3	3	3	3	3	x		x	
Four Square Step Test	2	3	2	2	3	2	3	3	1		x		
Fullerton Advanced Balance Scale	2	2	2	2	2	2	2	1	1				x
Function in Sitting Test	2	2	2	2	1	1	1	1	2				x
Functional Assessment of MS	1	3	3	3	3	3	3	3	3		x	x	
Functional Gait Assessment	2	2	2	2	2	2	2	1	1				x
Functional Independence Measure	1	3	1	3	1	3	3	3	3	x		x	
Functional Reach	3	3	3	2	3	3	3	1	1	x		x	
Goal Attainment Scale	1	3	3	3	3	3	3	3	3			x	
Guy's Neurological Disability Scale	3	3	3	3	3	3	3	3	3			x	
Hauser Ambulation Index	3	3	3	3	3	3	3	3	3				
High Level Mobility Assessment Tool (HiMat)	1	2	1	1	2	2	2	1	1				x
Maximal Inspiratory/Expiratory Pressure	3	3	3	3	3	3	3	3	3		x	x	
Maximal Oxygen Uptake: VO2 max and VO2 peak	1	1	1	1	3	3	3	2	1		x	x	
Modified Ashworth Scale of Spasticity	2	2	2	2	2	2	2	2	2				x
Modified Fatigue Impact Scale	3	3	3	3	3	3	3	3	1		x	x	

Appendix -B

Figure B.1: Gait cycle description for left and right legs (Tunca, Pehlivan, Ak, Arnrich, Salur & Ersoy, 2017)

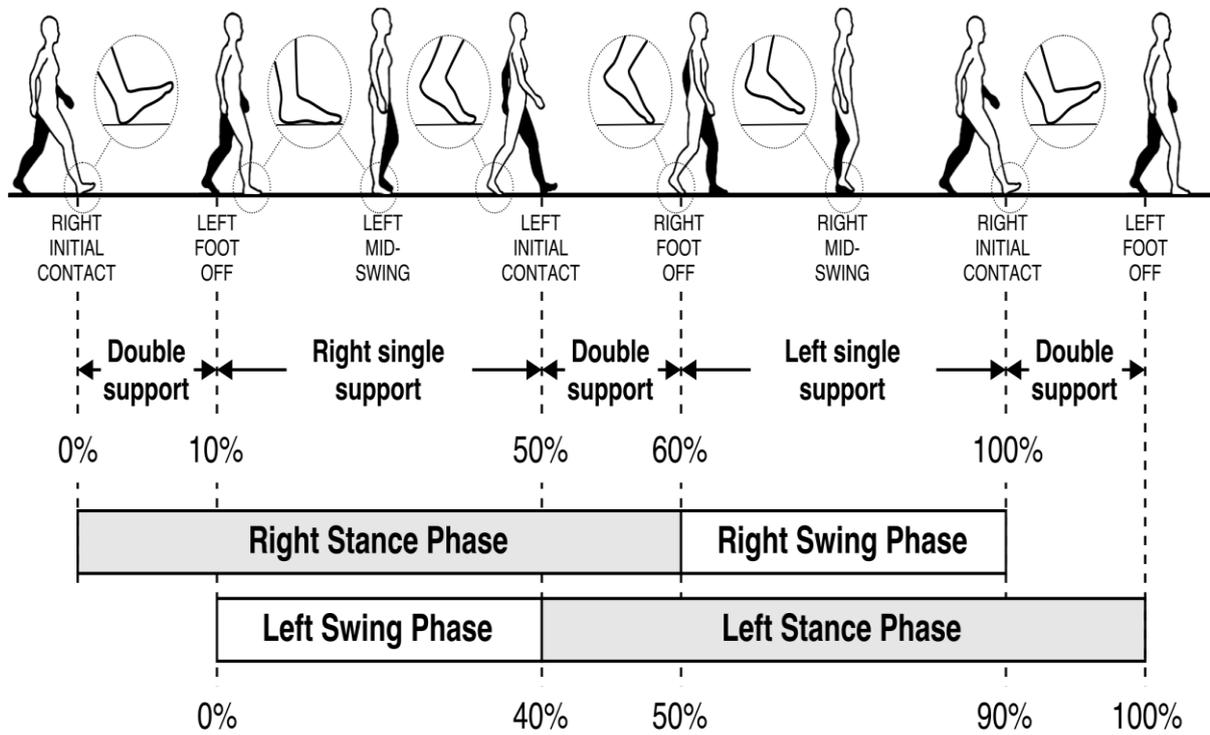


Table B.1: Confusion Matrix for the Evaluative Metrics of Classification Performance
 (https://en.wikipedia.org/wiki/Confusion_matrix)

		True condition			
		Condition positive	Condition negative	Prevalence= $\frac{\Sigma \text{Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC)= $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ total population}}$
Predicted condition	Predicted condition positive	True positive , <u>Power</u>	False positive , <u>Type I error</u>	<u>Positive predictive value</u> (PPV), <u>Precision</u> = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ predicted condition positive}}$	<u>False discovery rate</u> (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	False negative , <u>Type II error</u>	True negative	<u>False omission rate</u> (FOR) = $\frac{\Sigma \text{ false negative}}{\Sigma \text{ predicted condition negative}}$	<u>Negative predictive value</u> (NPV) = $\frac{\Sigma \text{ true negative}}{\Sigma \text{ predicted condition negative}}$
<u>True positive rate</u> (TPR), <u>Recall</u> , <u>Sensitivity</u> , probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ condition positive}}$		<u>False positive rate</u> (FPR), <u>Fall-out</u> , probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ condition negative}}$	<u>Positive likelihood ratio</u> (LR+)= $\frac{\text{TPR}}{\text{FPR}}$	<u>Diagnostic odds ratio</u> (DOR) = $\frac{\text{LR+}}{\text{LR-}}$	<u>F1 score</u> = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$
<u>False negative rate</u> (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ condition positive}}$		<u>Specificity</u> , <u>True negative rate</u> (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ condition negative}}$	<u>Negative likelihood ratio</u> (LR-)= $\frac{\text{FNR}}{\text{TNR}}$		

Appendix C

Figure C.1: The Main Stages of the Proposed System AM/CE for Classification of Gait Pattern Changes

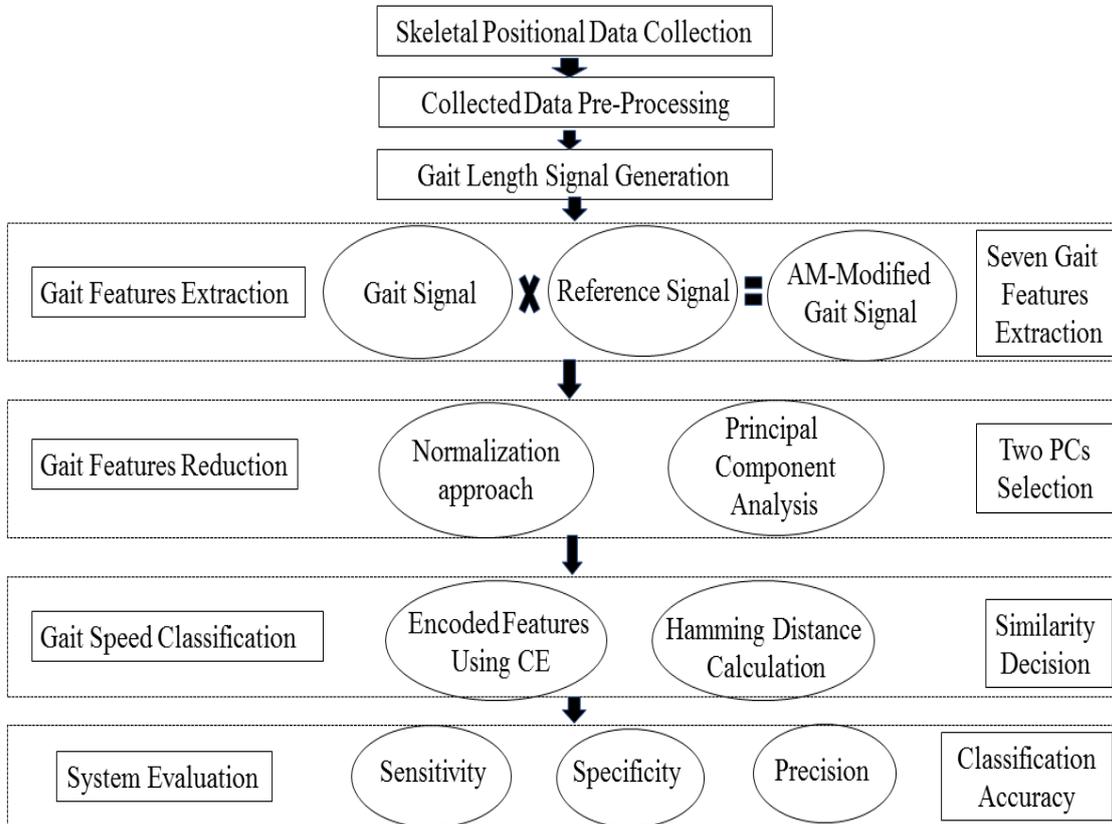


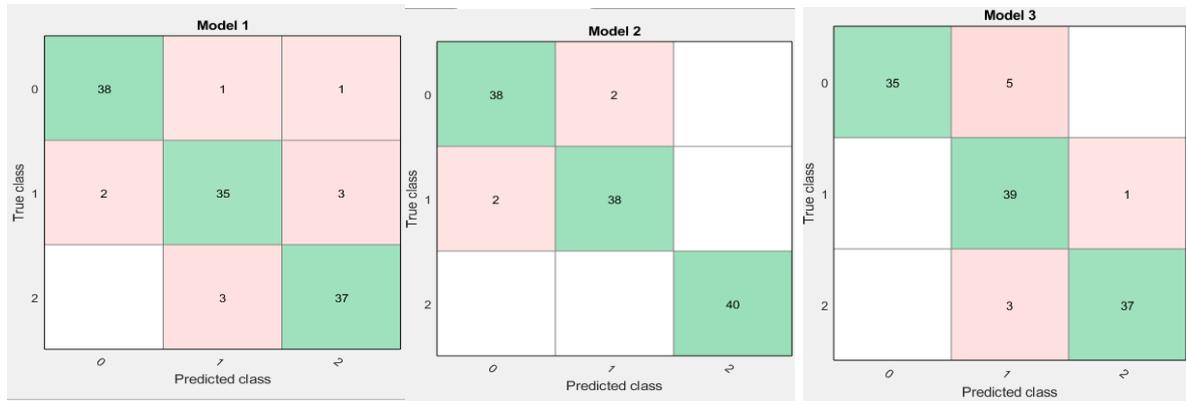
Table C.1: The Seven Gait Features from the AM-Modified Gait Signal are Reduced into Two Vectors using PCA

Seven columns of gait features from AM-Modified gait signal							PCA	
Lower fre	Upper fre	Bandwidth	Mod index	Mod eff	Sides level	Total level	PC1	PC2
21.6667	28.3333	6.6667	0.0703	0.0025	0.0747	0.9777	-1.0462	0.0237
21.6667	28.3333	6.6667	0.1009	0.0051	0.1151	1.0648	-0.9600	0.0630
20.0000	30.0000	10.0000	0.0969	0.0047	0.1012	0.9729	-0.9599	0.0629
15.0000	35.0000	20.0000	0.0056	0.0000	0.0064	1.0213	-0.9586	0.0626
15.0000	35.0000	20.0000	0.0056	0.0000	0.0064	1.0213	-0.9274	0.0518
15.0000	35.0000	20.0000	0.0056	0.0000	0.0064	1.0213	-0.8954	0.0242
21.6667	28.3333	6.6667	0.1258	0.0079	0.1420	1.0654	-0.8941	0.0241
21.6667	28.3333	6.6667	0.0958	0.0046	0.0973	0.9455	-0.6306	0.0195
21.6667	28.3333	6.6667	0.0825	0.0034	0.0816	0.9157	-0.6100	0.0165
21.6667	28.3333	6.6667	0.0958	0.0046	0.0973	0.9455	-0.5752	0.0125
13.3333	36.6667	23.3333	0.0055	0.0000	0.0059	0.9538	-0.5438	0.0115

3.3333	46.6667	43.3333	0.0006	0.0000	0.0007	0.9920	-0.5372	0.0058
13.3333	36.6667	23.3333	0.0055	0.0000	0.0059	0.9538	-0.5147	0.0003
21.6667	28.3333	6.6667	0.1559	0.0120	0.1738	1.0681	-0.4648	-0.0060
21.6667	28.3333	6.6667	0.1197	0.0071	0.1271	0.9998	-0.4378	-0.0095
21.6667	28.3333	6.6667	0.0920	0.0042	0.0986	0.9954	-0.4332	-0.0116
20.0000	30.0000	10.0000	0.0524	0.0014	0.0555	0.9651	-0.4114	-0.0129
21.6667	28.3333	6.6667	0.0847	0.0036	0.0935	1.0228	-0.3787	-0.0212
21.6667	28.3333	6.6667	0.1022	0.0052	0.1072	0.9796	-0.3604	-0.0287
21.6667	28.3333	6.6667	0.0783	0.0031	0.0845	0.9965	-0.3418	-0.0333
21.6667	28.3333	6.6667	0.0827	0.0034	0.0905	1.0119	-0.3406	-0.0343
21.6667	28.3333	6.6667	0.0744	0.0028	0.0797	0.9866	-0.3382	-0.0364
21.6667	28.3333	6.6667	0.1181	0.0069	0.1275	1.0160	-0.2933	-0.0399
21.6667	28.3333	6.6667	0.1283	0.0082	0.1392	1.0260	-0.2788	-0.0512
21.6667	28.3333	6.6667	0.1358	0.0091	0.1515	1.0587	-0.2542	-0.0559
20.0000	30.0000	10.0000	0.0743	0.0028	0.0787	0.9762	-0.2529	-0.0567
21.6667	28.3333	6.6667	0.0963	0.0046	0.0989	0.9563	-0.2451	-0.0610
21.6667	28.3333	6.6667	0.0592	0.0017	0.0579	0.8950	-0.2379	-0.0621
21.6667	28.3333	6.6667	0.1533	0.0116	0.1698	1.0596	-0.2059	-0.0651
21.6667	28.3333	6.6667	0.1177	0.0069	0.1271	1.0167	-0.1989	-0.0653
21.6667	28.3333	6.6667	0.1533	0.0116	0.1698	1.0596	-0.1477	0.0462
21.6667	28.3333	6.6667	0.0614	0.0019	0.0600	0.8951	-0.0394	0.0390
21.6667	28.3333	6.6667	0.0503	0.0013	0.0490	0.8865	-0.0365	0.0389
21.6667	28.3333	6.6667	0.0614	0.0019	0.0600	0.8951	0.0806	0.3840
13.3333	36.6667	23.3333	0.0060	0.0000	0.0066	0.9713	0.1739	0.3479
21.6667	28.3333	6.6667	0.0847	0.0036	0.0935	1.0228	0.1947	0.3455
21.6667	28.3333	6.6667	0.1022	0.0052	0.1072	0.9796	0.2937	0.4565
21.6667	28.3333	6.6667	0.0923	0.0042	0.0992	0.9988	0.4464	0.4413
21.6667	28.3333	6.6667	0.0847	0.0036	0.0935	1.0228	0.4470	0.4408
21.6667	28.3333	6.6667	0.0621	0.0019	0.0692	1.0211	0.6825	1.1329
21.6667	28.3333	6.6667	0.0960	0.0046	0.1063	1.0313	-0.3816	-0.0564
21.6667	28.3333	6.6667	0.0960	0.0046	0.1063	1.0313	-0.3053	-0.0682
20.0000	30.0000	10.0000	0.0799	0.0032	0.0884	1.0217	-0.2424	0.0465
20.0000	30.0000	10.0000	0.0941	0.0044	0.1017	1.0054	-0.2330	0.0449
20.0000	30.0000	10.0000	0.1018	0.0052	0.1146	1.0514	-0.2081	0.0358
20.0000	30.0000	10.0000	0.1203	0.0072	0.1316	1.0304	-0.1984	0.0345
20.0000	30.0000	10.0000	0.0900	0.0040	0.0990	1.0212	-0.1967	0.0345
20.0000	30.0000	10.0000	0.0957	0.0046	0.1040	1.0120	-0.1749	0.0201
20.0000	30.0000	10.0000	0.1215	0.0073	0.1394	1.0820	-0.1548	0.0173
20.0000	30.0000	10.0000	0.0985	0.0048	0.1131	1.0710	-0.1466	0.0125
20.0000	30.0000	10.0000	0.1191	0.0070	0.1358	1.0735	-0.1298	0.0025
20.0000	30.0000	10.0000	0.1394	0.0096	0.1584	1.0799	-0.1154	-0.0037
20.0000	30.0000	10.0000	0.1379	0.0094	0.1626	1.1207	-0.1034	-0.0077
20.0000	30.0000	10.0000	0.1320	0.0086	0.1512	1.0857	-0.0977	-0.0089
18.3333	31.6667	13.3333	0.0988	0.0049	0.1135	1.0711	-0.0862	-0.0101
20.0000	30.0000	10.0000	0.1037	0.0053	0.1201	1.0823	-0.0211	-0.0166
20.0000	30.0000	10.0000	0.1221	0.0074	0.1353	1.0453	0.0268	-0.0216
20.0000	30.0000	10.0000	0.1317	0.0086	0.1542	1.1091	0.0341	-0.0233
20.0000	30.0000	10.0000	0.1376	0.0094	0.1603	1.1059	0.0418	-0.0237
20.0000	30.0000	10.0000	0.1371	0.0093	0.1651	1.1440	0.0484	-0.0266
20.0000	30.0000	10.0000	0.1422	0.0100	0.1649	1.1032	0.0570	-0.0290
20.0000	30.0000	10.0000	0.1317	0.0086	0.1498	1.0771	0.0633	-0.0299
20.0000	30.0000	10.0000	0.1017	0.0051	0.1173	1.0770	0.0643	-0.0302
20.0000	30.0000	10.0000	0.1317	0.0086	0.1498	1.0771	0.0813	-0.0340
20.0000	30.0000	10.0000	0.1334	0.0088	0.1590	1.1300	0.0835	-0.0349
20.0000	30.0000	10.0000	0.1219	0.0074	0.1428	1.1042	0.0929	-0.0373
20.0000	30.0000	10.0000	0.1334	0.0088	0.1590	1.1300	0.1072	-0.0417
20.0000	30.0000	10.0000	0.0996	0.0049	0.1165	1.0916	0.1620	-0.0535
20.0000	30.0000	10.0000	0.1191	0.0070	0.1393	1.1012	0.1642	-0.0553
20.0000	30.0000	10.0000	0.1207	0.0072	0.1367	1.0673	0.1681	-0.0569
20.0000	30.0000	10.0000	0.0909	0.0041	0.1058	1.0820	0.1770	-0.0598
20.0000	30.0000	10.0000	0.1004	0.0050	0.1199	1.1135	0.1971	-0.0646
18.3333	31.6667	13.3333	0.0918	0.0042	0.1099	1.1126	0.2021	-0.0681
20.0000	30.0000	10.0000	0.1486	0.0109	0.1735	1.1146	0.2259	-0.0696
20.0000	30.0000	10.0000	0.1248	0.0077	0.1403	1.0615	0.2330	-0.0712
20.0000	30.0000	10.0000	0.1240	0.0076	0.1409	1.0721	0.2453	-0.0778
20.0000	30.0000	10.0000	0.1119	0.0062	0.1298	1.0880	0.2692	-0.0876
20.0000	30.0000	10.0000	0.1241	0.0076	0.1376	1.0464	0.2877	-0.0882

20.0000	30.0000	10.0000	0.1259	0.0079	0.1448	1.0864	0.4037	0.0188
20.0000	30.0000	10.0000	0.1498	0.0111	0.1827	1.1655	0.4517	-0.0052
20.0000	30.0000	10.0000	0.1293	0.0083	0.1558	1.1404	-0.4858	-0.1336
18.3333	31.6667	13.3333	0.1019	0.0052	0.1216	1.1145	-0.4858	-0.1336
18.3333	31.6667	13.3333	0.1474	0.0107	0.1813	1.1739	-0.1158	-0.0479
21.6667	28.3333	6.6667	0.0445	0.0010	0.0545	1.1115	-0.1051	-0.0519
21.6667	28.3333	6.6667	0.0445	0.0010	0.0545	1.1115	-0.0793	-0.0541
18.3333	31.6667	13.3333	0.0954	0.0045	0.1155	1.1275	-0.0577	-0.0568
18.3333	31.6667	13.3333	0.1109	0.0061	0.1344	1.1369	-0.0112	-0.0630
20.0000	30.0000	10.0000	0.1213	0.0073	0.1476	1.1470	0.0267	-0.0744
18.3333	31.6667	13.3333	0.1595	0.0126	0.1965	1.1821	0.0322	-0.0771
18.3333	31.6667	13.3333	0.1412	0.0099	0.1686	1.1358	0.0966	0.0372
18.3333	31.6667	13.3333	0.1412	0.0099	0.1686	1.1358	0.1046	0.0303
18.3333	31.6667	13.3333	0.1628	0.0131	0.2100	1.2398	0.1046	0.0303
18.3333	31.6667	13.3333	0.1727	0.0147	0.2204	1.2319	0.1596	0.0238
18.3333	31.6667	13.3333	0.1525	0.0115	0.1922	1.2060	0.1827	0.0172
18.3333	31.6667	13.3333	0.1157	0.0066	0.1363	1.1082	0.1895	0.0113
18.3333	31.6667	13.3333	0.1157	0.0066	0.1363	1.1082	0.1911	0.0108
18.3333	31.6667	13.3333	0.1157	0.0066	0.1363	1.1082	0.2135	0.0015
18.3333	31.6667	13.3333	0.1221	0.0074	0.1693	1.3069	0.2337	-0.0160
18.3333	31.6667	13.3333	0.1221	0.0074	0.1693	1.3069	0.2570	-0.0204
18.3333	31.6667	13.3333	0.1221	0.0074	0.1693	1.3069	0.2746	-0.0215
18.3333	31.6667	13.3333	0.1231	0.0075	0.1552	1.1889	0.2908	-0.0280
18.3333	31.6667	13.3333	0.1261	0.0079	0.1608	1.2043	0.3026	-0.0344
18.3333	31.6667	13.3333	0.1231	0.0075	0.1552	1.1889	0.3523	-0.0358
18.3333	31.6667	13.3333	0.0989	0.0049	0.1196	1.1283	0.3902	-0.0471
18.3333	31.6667	13.3333	0.1645	0.0134	0.2079	1.2154	0.3902	-0.0471
18.3333	31.6667	13.3333	0.1645	0.0134	0.2079	1.2154	0.4343	-0.0527
18.3333	31.6667	13.3333	0.1445	0.0103	0.1882	1.2406	0.4700	-0.0597
18.3333	31.6667	13.3333	0.1900	0.0177	0.2520	1.2906	0.5037	-0.0628
18.3333	31.6667	13.3333	0.1900	0.0177	0.2520	1.2906	0.5089	-0.0673
20.0000	30.0000	10.0000	0.1107	0.0061	0.1353	1.1468	0.5138	-0.0715
20.0000	30.0000	10.0000	0.1067	0.0057	0.1317	1.1551	0.5229	-0.0724
20.0000	30.0000	10.0000	0.0948	0.0045	0.1134	1.1131	0.5798	-0.0818
18.3333	31.6667	13.3333	0.1516	0.0114	0.1904	1.2004	0.6266	-0.0925
18.3333	31.6667	13.3333	0.1516	0.0114	0.1904	1.2004	0.6485	-0.0955
18.3333	31.6667	13.3333	0.1516	0.0114	0.1904	1.2004	0.7125	-0.1421
18.3333	31.6667	13.3333	0.1285	0.0082	0.1600	1.1779	0.7286	-0.1433
18.3333	31.6667	13.3333	0.1371	0.0093	0.1625	1.1251	0.7561	-0.1605
18.3333	31.6667	13.3333	0.1285	0.0082	0.1600	1.1779	0.8233	-0.1641
20.0000	30.0000	10.0000	0.1557	0.0120	0.2153	1.3241	1.0199	-0.1720
20.0000	30.0000	10.0000	0.1613	0.0128	0.2034	1.2108	1.0383	-0.1880

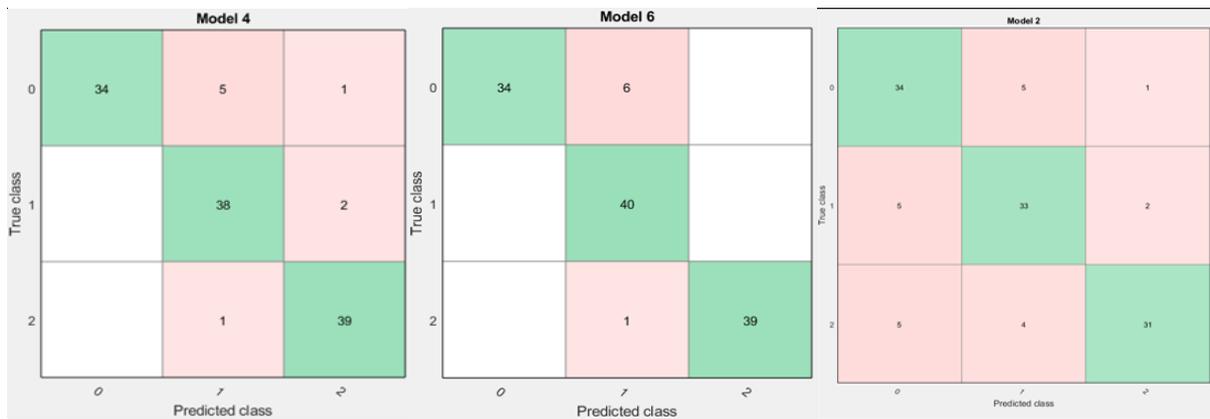
Figure C.2: Confusion Matrix for Several Supervised Classifiers at 5-Fold Cross Validation, (a) DT, (b) k-NN, (c) L-SVM, (d) Q-SVM, (e) L-D, (f) Q-D classifiers.



(a)

(b)

(c)

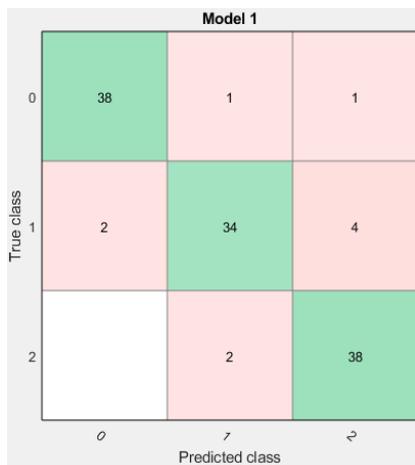


(d)

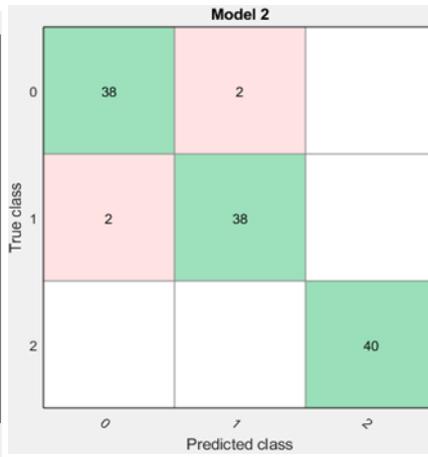
(e)

(f)

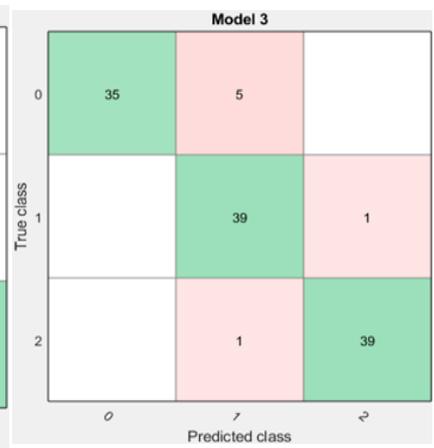
Figure C.3: Confusion Matrix for Several Supervised Classifiers at 10-Fold Cross Validation, (a) DT, (b) k-NN, (c) L-SVM, (d) Q-SVM, (e) L-D, (f) Q-D classifiers.



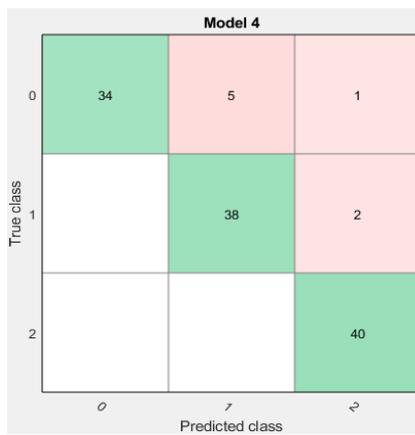
(a)



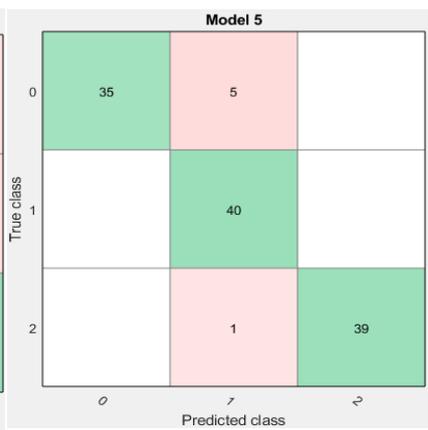
(b)



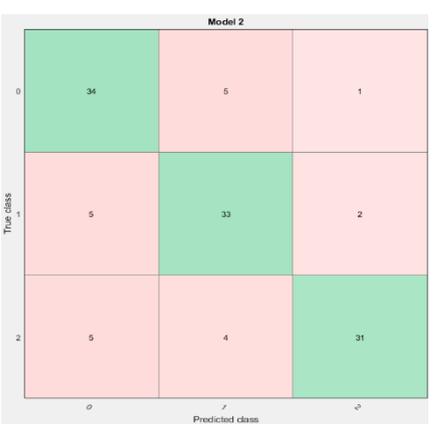
(c)



(d)



(e)



(f)

Appendix D

D-1 MATLAB code for gait features extraction using spatiotemporal gait analysis to calculate gait step length, stride gait length and gait cadence.

```
clc; clear all;close all;
```

```
%%%%%%%%%%%% Data Import %%%%%%%%%%
```

```
TP1S1=load('p1_slow1_towards.txt');TP1S2=load('p1_slow2_towards.txt');
TP1S3=load('p1_slow3_towards.txt');TP1F1=load('p1_fast1_towards.txt');TP1F2=load('p1_fast2_towards.txt');
TP1F3=load('p1_fast3_towards.txt');TP1N1=load('p1_normal1_towards.txt'); TP1N2=load('p1_normal2_towards.txt');
TP1N3=load('p1_normal3_towards.txt'); TP2S1=load('p2_slow1_towards.txt'); TP2S2=load('p2_slow2_towards.txt');
TP2S3=load('p2_slow3_towards.txt');TP2S4=load('p2_slow4_towards.txt');TP2F1=load('p2_fast1_towards.txt');TP2F2=load('p2_fast2_towards.txt');
TP2F3=load('p2_fast3_towards.txt');TP2N1=load('p2_normal1_towards.txt');
TP2N2=load('p2_normal2_towards.txt'); TP2N3 =load('p2_normal3_towards.txt'); TP3S1=load('p3_slow1_towards.txt');
TP3S2=load('p3_slow2_towards.txt'); TP3S3=load('p3_slow3_towards.txt'); TP3S4=load('p3_slow4_towards.txt');
TP3F1=load('p3_fast1_towards.txt'); TP3F2=load('p3_fast2_towards.txt'); TP3F3=load('p3_fast3_towards.txt');
TP3N1=load('p3_normal1_towards.txt'); TP3N2=load('p3_normal2_towards.txt'); TP3N3 =load('p3_normal3_towards.txt');
TP4S1=load('p4_slow1_towards.txt'); TP4S2=load('p4_slow2_towards.txt');
TP4S3=load('p4_slow3_towards.txt');TP4F1=load('p4_fast1_towards.txt'); TP4F2=load('p4_fast2_towards.txt');
TP4F3=load('p4_fast3_towards.txt');TP4N1=load('p4_normal1_towards.txt'); TP4N2=load('p4_normal2_towards.txt');
TP4N3=load('p4_normal3_towards.txt');TP5S1=load('p5_slow1_towards80h.txt');TP5S2=load('p5_slow2_towards80h.txt');
TP5S3=load('p5_slow3_towards80h.txt');TP5S4=load('p5_slow4_towards80h.txt');TP5F1=load('p5_fast1_towards80h.txt');
TP5F2=load('p5_fast2_towards80h.txt');TP5F3=load('p5_fast3_towards80h.txt');TP5N1=load('p5_normal1_towards80h.txt');
TP5N2=load('p5_normal2_towards80h.txt');TP5N3=load('p5_normal3_towards80h.txt');
TP6S1=load('p6_slow1_towards.txt');TP6S2=load('p6_slow2_towards.txt'); TP6F2=load('p6_fast2_towards.txt');
TP6S3=load('p6_slow3_towards.txt');TP6F1=load('p6_fast1_towards.txt'); TP6N1=load('p6_normal1_towards.txt');TP7S1=load('p7_slow1_towards.txt');
TP6N2=load('p6_normal2_towards.txt'); TP6N3=load('p6_normal3_towards.txt');TP7S2=load('p7_slow2_towards.txt');
TP7S3=load('p7_slow3_towards.txt');TP7F1=load('p7_fast1_towards.txt');
TP7F2=load('p7_fast2_towards.txt'); TP7F3=load('p7_fast3_towards.txt');TP7N1=load('p7_normal1_towards.txt');
TP7N2=load('p7_normal2_towards.txt');TP7N3=load('p7_normal3_towards.txt');TP8S1=load('P8_slow1.txt');TP8S2=load('P8_slow2.txt');
TP8S3=load('P8_slow3.txt');TP8N1=load('P8_normal1.txt');TP8N2=load('P8_normal2.txt');TP8N3=load('P8_normal3.txt');
TP8F1=load('P8_fast1.txt');TP8F2=load('P8_fast2.txt');TP8F3=load('P8_fast3.txt');TP9S1=load('P9_slow1.txt');
TP9S2=load('P9_slow2.txt');TP9S3=load('P9_slow3.txt');TP9N1=load('P9_slow1.txt');TP9N2=load('P9_slow2.txt');TP9N3=load('P9_slow3.txt');
TP9F1=load('P9_slow1.txt');TP9F2=load('P9_slow2.txt');TP9F3=load('P9_slow3.txt');TP10S1=load('P10_slow1.txt');
TP10S2=load('P10_slow2.txt');TP10S3=load('P10_slow3.txt');TP10N1=load('P10_normal1.txt');TP10N2=load('P10_normal2.txt');
TP10N3=load('P10_normal3.txt');TP10F1=load('P10_fast1.txt');TP10F2=load('P10_fast2.txt');TP10F3=load('P10_fast3.txt');
TP11S1=load('P11_slow1.txt');TP11S2=load('P11_slow2.txt');TP11S3=load('P11_slow3.txt');TP11N1=load('P11_normal1.txt');
TP11N2=load('P11_normal2.txt');TP11N3=load('P11_normal3.txt');TP11F1=load('P11_fast1.txt');
TP11F2=load('P11_fast2.txt');TP11F3=load('P11_fast3.txt');TP12S1=load('P12_slow1.txt');TP12S2=load('P12_slow2.txt');
TP12S3=load('P12_slow3.txt');TP12N1=load('P12_normal1.txt');TP12N2=load('P12_normal2.txt');TP12N3=load('P12_normal3.txt');
TP12F1=load('P12_fast1.txt');TP12F2=load('P12_fast2.txt');TP12F3=load('P12_fast3.txt');TP13S1=load('P13_slow1.txt');
TP13S2=load('P13_slow2.txt');TP13S3=load('P13_slow3.txt');TP13N1=load('P13_normal1.txt');TP13N2=load('P13_normal2.txt');
TP13N3=load('P13_normal3.txt');TP13F1=load('P13_fast1.txt');TP13F2=load('P13_fast2.txt');TP13F3=load('P13_fast3.txt');
TP14S1=load('P14_slow1.txt');TP14S2=load('P14_slow2.txt');TP14S3=load('P14_slow3.txt');TP14N1=load('P14_normal1.txt');
TP14N2=load('P14_normal2.txt');TP14N3=load('P14_normal3.txt');TP14F1=load('P14_fast1.txt');TP14F2=load('P14_fast2.txt');
TP14F3=load('P14_fast3.txt');
```

```
%%%%%%%%%%%% 3 different walking SPEEDs ON FRONT VIEW %%%%%%%%%%
```

```
% Spine base tracking in z-axis
```

```
T1S1 = TP1S1(40:99,3); T1S2 = TP1S2(40:99,3); T1S3 = TP1S3(40:99,3);
T2S1= TP2S1(40:99,3); T2S2 = TP2S2(40:99,3); T2S3 = TP2S3(40:99,3);
T3S1= TP3S1(40:99,3); T3S2 = TP3S2(40:99,3);T3S3 = TP3S3(40:99,3);
T4S1 = TP4S1(40:99,3); T4S2 = TP4S2(40:99,3); T4S3 = TP4S3(40:99,3);
T5S1 =TP5S1(40:99,3); T5S2 = TP5S2(40:99,3); T5S3 = TP5S3(40:99,3);
T6S1 = TP6S1(40:99,3); T6S2 = TP6S2(40:99,3); T6S3 = TP6S3(40:99,3);
T7S1 = TP7S1(40:99,3); T7S2 = TP7S2(40:99,3); T7S3 = TP7S3(40:99,3);
```

T8S1 = TP8S1(40:99,3); T8S2 = TP8S2(40:99,3); T8S3 = TP8S3(40:99,3);
 T9S1= TP9S1(40:99,3); T9S2 = TP9S2(40:99,3); T9S3 = TP9S3(40:99,3);
 T10S1= TP10S1(40:99,3); T10S2 = TP10S2(40:99,3); T10S3 = TP10S3(40:99,3);
 T11S1 = TP11S1(40:99,3); T11S2 = TP11S2(40:99,3); T11S3 = TP11S3(40:99,3);
 T12S1 =TP12S1(40:99,3); T12S2 = TP12S2(40:99,3); T12S3 = TP12S3(40:99,3);
 T13S1 = TP13S1(40:99,3); T13S2 = TP13S2(40:99,3); T13S3 = TP13S1(40:99,3);
 T14S1 = TP14S1(80:139,3); T14S2 = TP14S2(40:99,3); T14S3 = TP14S1(80:139,3);
 T15S1 = TP1N1(40:99,3); T15S2 = TP1N2(40:99,3); T15S3 = TP1N3(40:99,3);
 T16S1= TP2N1(40:99,3); T16S2 = TP2N2(40:99,3); T16S3 = TP2N3(40:99,3);
 T17S1= TP3N1(35:94,3); T17S2 = TP3N2(40:99,3); T17S3 = TP3N3(40:99,3);
 T18S1 = TP4N1(40:99,3); T18S2 = TP4N2(40:99,3); T18S3 = TP4N3(40:99,3);
 T19S1 =TP5N1(30:89,3); T19S2 = TP5N2(9:68,3); T19S3 = TP5N3(12:71,3);
 T20S1 = TP6N1(40:99,3); T20S2 = TP6N2(29:88,3); T20S3 = TP6N1(40:99,3);
 T21S1 = TP7N1(40:99,3); T21S2 = TP7N2(40:99,3); T21S3 = TP7N1(40:99,3);
 T22S1 = TP8N1(40:99,3); T22S2 = TP8N2(40:99,3); T22S3 = TP8N3(40:99,3);
 T23S1= TP9N1(40:99,3); T23S2 = TP9N2(40:99,3); T23S3 = TP9N3(40:99,3);
 T24S1= TP10N1(40:99,3); T24S2 = TP10N2(40:99,3); T24S3 = TP10N3(40:99,3);
 T25S1 = TP11N1(40:99,3); T25S2 = TP11N2(40:99,3); T25S3 = TP11N3(40:99,3);
 T26S1 =TP12N1(40:99,3); T26S2 = TP12N2(40:99,3); T26S3 = TP12N3(40:99,3);
 T27S1 = TP13N1(40:99,3); T27S2 = TP13N2(50:109,3); T27S3 = TP13N1(40:99,3);
 T28S1 = TP14N1(65:124,3); T28S2 = TP14N1(65:124,3); T28S3 = TP14N1(65:124,3);
 T29S1 = TP1F1(1:60,3); T29S2 = TP1F2(1:60,3); T29S3 = TP1F2(1:60,3);
 T30S1= TP2F1(1:60,3); T30S2 = TP2F2(1:60,3); T30S3 = TP2F3(1:60,3);
 T31S1= TP3F1(1:60,3); T31S2 = TP3F2(1:60,3); T31S3= TP3F2(1:60,3);
 T32S1 = TP4F1(1:60,3); T32S2 = TP4F2(1:60,3); T32S3 = TP4F3(1:60,3);
 T33S1 =TP5F1(1:60,3); T33S2 = TP5F1(1:60,3); T33S3 = TP5F1(1:60,3);
 T34S1 = TP6F1(1:60,3); T34S2 = TP6F1(1:60,3); T34S3 = TP6F1(1:60,3);
 T35S1 = TP7F1(1:60,3); T35S2 = TP7F2(1:60,3); T35S3 = TP7F1(1:60,3);
 T36S1 = TP8F1(1:60,3); T36S2 = TP8F2(1:60,3); T36S3 = TP8F2(1:60,3);
 T37S1= TP9F1(11:70,3); T37S2 = TP9F2(1:60,3); T37S3 = TP9F3(1:60,3);
 T38S1= TP10F1(1:60,3); T38S2 = TP10F2(1:60,3); T38S3= TP10F2(1:60,3);
 T39S1 = TP11F1(1:60,3); T39S2 = TP11F2(1:60,3); T39S3 = TP11F3(1:60,3);
 T40S1 =TP12F1(1:60,3); T40S2 = TP12F1(1:60,3); T40S3 = TP12F1(1:60,3);
 T41S1 = TP13F3(40:99,3); T41S2 = TP13F2(13:72,3); T41S3 = TP13F3(40:99,3);
 T42S1 = TP14F1(1:60,3); T42S2 = TP14F2(20:79,3); T42S3 = TP14F1(1:60,3);

%%%%%%%%%%%%Smoothing Data using LR filter %%%%%%%%%%

window=11;

TSS(:,1)=smooth(T1S1>window/length(T1S1),'rloess');TSS(:,2)=smooth(T1S2>window/length(T1S2),'rloess');
 TSS(:,3)=smooth(T1S3>window/length(T1S3),'rloess');TSS(:,4)=smooth(T2S1>window/length(T2S1),'rloess');
 TSS(:,5)=smooth(T2S2>window/length(T2S2),'rloess');TSS(:,6)=smooth(T2S3>window/length(T2S3),'rloess');
 TSS(:,7)=smooth(T3S1>window/length(T3S1),'rloess');TSS(:,8)=smooth(T3S2>window/length(T3S2),'rloess');
 TSS(:,9)=smooth(T3S3>window/length(T3S3),'rloess');TSS(:,10)=smooth(T4S1>window/length(T4S1),'rloess');
 TSS(:,11)=smooth(T4S2>window/length(T4S2),'rloess');TSS(:,12)=smooth(T4S3>window/length(T4S3),'rloess');
 TSS(:,13)=smooth(T5S1>window/length(T5S1),'rloess');TSS(:,14)=smooth(T5S2>window/length(T5S2),'rloess');
 TSS(:,15)=smooth(T5S3>window/length(T5S3),'rloess');TSS(:,16)=smooth(T6S1>window/length(T6S1),'rloess');
 TSS(:,17)=smooth(T6S2>window/length(T6S2),'rloess');TSS(:,18)=smooth(T6S3>window/length(T6S3),'rloess');
 TSS(:,19)=smooth(T7S1>window/length(T7S1),'rloess');TSS(:,20)=smooth(T7S2>window/length(T7S2),'rloess');
 TSS(:,21)=smooth(T7S3>window/length(T7S3),'rloess');TSS(:,22)=smooth(T8S1>window/length(T8S1),'rloess');
 TSS(:,23)=smooth(T8S2>window/length(T8S2),'rloess');TSS(:,24)=smooth(T8S3>window/length(T8S3),'rloess');
 TSS(:,25)=smooth(T9S1>window/length(T9S1),'rloess');TSS(:,26)=smooth(T9S2>window/length(T9S2),'rloess');

TSS(:,27)=smooth(T9S3>window/length(T9S3),'rloess');TSS(:,28)=smooth(T10S1>window/length(T10S1),'rloess');
TSS(:,29)=smooth(T10S2>window/length(T10S2),'rloess');TSS(:,30)=smooth(T10S3>window/length(T10S3),'rloess');
TSS(:,31)=smooth(T11S1>window/length(T11S1),'rloess');TSS(:,32)=smooth(T11S2>window/length(T11S2),'rloess');
TSS(:,33)=smooth(T11S3>window/length(T11S3),'rloess');TSS(:,34)=smooth(T12S1>window/length(T12S1),'rloess');
TSS(:,35)=smooth(T12S2>window/length(T12S2),'rloess');TSS(:,36)=smooth(T12S3>window/length(T12S3),'rloess');
TSS(:,37)=smooth(T13S1>window/length(T13S1),'rloess');TSS(:,38)=smooth(T13S2>window/length(T13S2),'rloess');
TSS(:,39)=smooth(T13S3>window/length(T13S3),'rloess');TSS(:,40)=smooth(T14S1>window/length(T14S1),'rloess');
TSS(:,41)=smooth(T14S2>window/length(T14S2),'rloess');TSS(:,42)=smooth(T14S3>window/length(T14S3),'rloess');

TSS(:,43)=smooth(T15S1>window/length(T15S1),'rloess');TSS(:,44)=smooth(T15S2>window/length(T15S2),'rloess');
TSS(:,45)=smooth(T15S3>window/length(T15S3),'rloess');TSS(:,46)=smooth(T16S1>window/length(T16S1),'rloess');
TSS(:,47)=smooth(T16S2>window/length(T16S2),'rloess');TSS(:,48)=smooth(T16S3>window/length(T16S3),'rloess');
TSS(:,49)=smooth(T17S1>window/length(T17S1),'rloess');TSS(:,50)=smooth(T17S2>window/length(T17S2),'rloess');
TSS(:,51)=smooth(T17S3>window/length(T17S3),'rloess');TSS(:,52)=smooth(T18S1>window/length(T18S1),'rloess');
TSS(:,53)=smooth(T18S2>window/length(T18S2),'rloess');TSS(:,54)=smooth(T18S3>window/length(T18S3),'rloess');
TSS(:,55)=smooth(T19S1>window/length(T19S1),'rloess');TSS(:,56)=smooth(T19S2>window/length(T19S2),'rloess');
TSS(:,57)=smooth(T19S3>window/length(T19S3),'rloess');TSS(:,58)=smooth(T20S1>window/length(T20S1),'rloess');
TSS(:,59)=smooth(T20S2>window/length(T20S2),'rloess');TSS(:,60)=smooth(T20S3>window/length(T20S3),'rloess');
TSS(:,61)=smooth(T21S1>window/length(T21S1),'rloess');TSS(:,62)=smooth(T21S2>window/length(T21S2),'rloess');
TSS(:,63)=smooth(T21S3>window/length(T21S3),'rloess');TSS(:,64)=smooth(T22S1>window/length(T22S1),'rloess');
TSS(:,65)=smooth(T22S2>window/length(T22S2),'rloess');TSS(:,66)=smooth(T22S3>window/length(T22S3),'rloess');
TSS(:,67)=smooth(T23S1>window/length(T23S1),'rloess');TSS(:,68)=smooth(T23S2>window/length(T23S2),'rloess');
TSS(:,69)=smooth(T23S3>window/length(T23S3),'rloess');TSS(:,70)=smooth(T24S1>window/length(T24S1),'rloess');
TSS(:,71)=smooth(T24S2>window/length(T24S2),'rloess');TSS(:,72)=smooth(T24S3>window/length(T24S3),'rloess');
TSS(:,73)=smooth(T25S1>window/length(T25S1),'rloess');TSS(:,74)=smooth(T25S2>window/length(T25S2),'rloess');
TSS(:,75)=smooth(T25S3>window/length(T25S3),'rloess');TSS(:,76)=smooth(T26S1>window/length(T26S1),'rloess');
TSS(:,77)=smooth(T26S2>window/length(T26S2),'rloess');TSS(:,78)=smooth(T26S3>window/length(T26S3),'rloess');
TSS(:,79)=smooth(T27S1>window/length(T27S1),'rloess');TSS(:,80)=smooth(T27S2>window/length(T27S2),'rloess');
TSS(:,81)=smooth(T27S3>window/length(T27S3),'rloess');TSS(:,82)=smooth(T28S1>window/length(T28S1),'rloess');
TSS(:,83)=smooth(T28S2>window/length(T28S2),'rloess');TSS(:,84)=smooth(T28S3>window/length(T28S3),'rloess');
TSS(:,85)=smooth(T29S1>window/length(T29S1),'rloess');TSS(:,86)=smooth(T29S2>window/length(T29S2),'rloess');
TSS(:,87)=smooth(T29S3>window/length(T29S3),'rloess');TSS(:,88)=smooth(T30S1>window/length(T30S1),'rloess');
TSS(:,89)=smooth(T30S2>window/length(T30S2),'rloess');TSS(:,90)=smooth(T30S3>window/length(T30S3),'rloess');
TSS(:,91)=smooth(T31S1>window/length(T31S1),'rloess');TSS(:,92)=smooth(T31S2>window/length(T31S2),'rloess');
TSS(:,93)=smooth(T31S3>window/length(T31S3),'rloess');TSS(:,94)=smooth(T32S1>window/length(T32S1),'rloess');
TSS(:,95)=smooth(T32S2>window/length(T32S2),'rloess');TSS(:,96)=smooth(T32S3>window/length(T32S3),'rloess');
TSS(:,97)=smooth(T33S1>window/length(T33S1),'rloess');TSS(:,98)=smooth(T33S2>window/length(T33S2),'rloess');
TSS(:,99)=smooth(T33S3>window/length(T33S3),'rloess');TSS(:,100)=smooth(T34S1>window/length(T34S1),'rloess');
TSS(:,101)=smooth(T34S2>window/length(T34S2),'rloess');TSS(:,102)=smooth(T34S3>window/length(T34S3),'rloess');
TSS(:,103)=smooth(T35S1>window/length(T35S1),'rloess');TSS(:,104)=smooth(T35S2>window/length(T35S2),'rloess');
TSS(:,105)=smooth(T35S3>window/length(T35S3),'rloess');TSS(:,106)=smooth(T36S1>window/length(T36S1),'rloess');
TSS(:,107)=smooth(T36S2>window/length(T36S2),'rloess');TSS(:,108)=smooth(T36S3>window/length(T36S3),'rloess');
TSS(:,109)=smooth(T37S1>window/length(T37S1),'rloess');TSS(:,110)=smooth(T37S2>window/length(T37S2),'rloess');
TSS(:,111)=smooth(T37S3>window/length(T37S3),'rloess');TSS(:,112)=smooth(T38S1>window/length(T38S1),'rloess');
TSS(:,113)=smooth(T38S2>window/length(T38S2),'rloess');TSS(:,114)=smooth(T38S3>window/length(T38S3),'rloess');
TSS(:,115)=smooth(T39S1>window/length(T39S1),'rloess');TSS(:,116)=smooth(T39S2>window/length(T39S2),'rloess');
TSS(:,117)=smooth(T39S3>window/length(T39S3),'rloess');TSS(:,118)=smooth(T40S1>window/length(T40S1),'rloess');
TSS(:,119)=smooth(T40S2>window/length(T40S2),'rloess');TSS(:,120)=smooth(T40S3>window/length(T40S3),'rloess');
TSS(:,121)=smooth(T41S1>window/length(T41S1),'rloess');TSS(:,122)=smooth(T41S2>window/length(T41S2),'rloess');
TSS(:,123)=smooth(T41S3>window/length(T41S3),'rloess');TSS(:,124)=smooth(T42S1>window/length(T42S1),'rloess');
TSS(:,125)=smooth(T42S2>window/length(T42S2),'rloess');TSS(:,126)=smooth(T42S3>window/length(T42S3),'rloess');

%%%%%%%%%%%%%% Left Ankle tracking

L1S1 = TP1S1(40:99,12); L1S2 = TP1S2(40:99,12); L1S3 = TP1S3(40:99,12);
L2S1= TP2S1(40:99,12); L2S2 = TP2S2(40:99,12); L2S3 = TP2S3(40:99,12);
L3S1= TP3S1(40:99,12); L3S2 = TP3S2(40:99,12); L3S3 = TP3S3(40:99,12);
L4S1 = TP4S1(40:99,12); L4S2 = TP4S2(40:99,12); L4S3 = TP4S3(40:99,12);
L5S1 = TP5S1(40:99,12); L5S2 = TP5S2(40:99,12); L5S3 = TP5S3(25:84,12);
L6S1 = TP6S1(40:99,12); L6S2 = TP6S2(40:99,12); L6S3 = TP6S3(40:99,12);
L7S1 = TP7S1(20:79,12); L7S2 = TP7S2(40:99,12); L7S3 = TP7S3(20:79,12);
L8S1 = TP8S1(35:94,12); L8S2 = TP8S2(40:99,12); L8S3 = TP8S3(40:99,12);
L9S1= TP9S1(40:99,12); L9S2 = TP9S2(40:99,12); L9S3 = TP9S3(40:99,12);

L10S1= TP10S1(40:99,12); L10S2 = TP10S2(40:99,12); L10S3 = TP10S3(40:99,12);
 L11S1 = TP11S1(40:99,12); L11S2 = TP11S2(40:99,12); L11S3 = TP11S3(40:99,12);
 L12S1 = TP12S1(40:99,12); L12S2 = TP12S2(40:99,12); L12S3 = TP12S3(40:99,12);
 L13S1 = TP13S1(40:99,12); L13S2 = TP13S2(30:89,12); L13S3 = TP13S1(40:99,12);
 L14S1 = TP14S1(100:159,12); L14S2 = TP14S2(40:99,12); L14S3 = TP14S1(100:159,12);
 L15S1 = TP1N1(30:89,12); L15S2 = TP1N2(40:99,12); L15S3 = TP1N3(40:99,12);
 L16S1 = TP2N1(40:99,12); L16S2 = TP2N2(40:99,12); L16S3 = TP2N3(40:99,12);
 L17S1 = TP3N1(25:84,12); L17S2 = TP3N2(40:99,12); L17S3 = TP3N3(40:99,12);
 L18S1 = TP4N1(40:99,12); L18S2 = TP4N2(40:99,12); L18S3 = TP4N3(40:99,12);
 L19S1 = TP5N1(40:99,12); L19S2 = TP5N2(9:68,12); L19S3 = TP5N3(12:71,12);
 L20S1 = TP6N1(40:99,12); L20S2 = TP6N2(29:88,12); L20S3 = TP6N1(40:99,12);
 L21S1 = TP7N1(40:99,12); L21S2 = TP7N2(30:89,12); L21S3 = TP7N1(40:99,12);
 L22S1 = TP8N1(40:99,12); L22S2 = TP8N2(40:99,12); L22S3 = TP8N3(40:99,12);
 L23S1 = TP9N1(180:239,12); L23S2 = TP9N2(40:99,12); L23S3 = TP9N3(40:99,12);
 L24S1 = TP10N1(40:99,12); L24S2 = TP10N2(40:99,12); L24S3 = TP10N3(40:99,12);
 L25S1 = TP11N1(40:99,12); L25S2 = TP11N2(10:69,12); L25S3 = TP11N3(40:99,12);
 L26S1 = TP12N1(40:99,12); L26S2 = TP12N2(40:99,12); L26S3 = TP12N3(40:99,12);
 L27S1 = TP13N1(40:99,12); L27S2 = TP13N2(40:99,12); L27S3 = TP13N1(40:99,12);
 L28S1 = TP14N1(60:119,12); L28S2 = TP14N2(100:159,12); L28S3 = TP14N1(60:119,12);
 L29S1 = TP1F1(1:60,12); L29S2 = TP1F2(1:60,12); L29S3 = TP1F2(1:60,12); %87
 L30S1 = TP2F1(1:60,12); L30S2 = TP2F2(1:60,12); L30S3 = TP2F3(1:60,12); %90
 L31S1 = TP3F1(1:60,12); L31S2 = TP3F2(1:60,12); L31S3 = TP3F2(1:60,12); %93
 L32S1 = TP4F1(1:60,12); L32S2 = TP4F2(1:60,12); L32S3 = TP4F3(1:60,12); %96
 L33S1 = TP5F1(1:60,12); L33S2 = TP5F1(1:60,12); L33S3 = TP5F1(1:60,12); %99
 L34S1 = TP6F1(1:60,12); L34S2 = TP6F1(1:60,12); L34S3 = TP6F1(1:60,12); %102
 L35S1 = TP7F1(1:60,12); L35S2 = TP7F2(1:60,12); L35S3 = TP7F1(1:60,12); %105
 L36S1 = TP8F1(1:60,12); L36S2 = TP8F2(1:60,12); L36S3 = TP8F2(1:60,12); %108
 L37S1 = TP9F3(188:247,12); L37S2 = TP9F2(40:99,12); L37S3 = TP9F3(20:79,12); %111
 L38S1 = TP10F1(1:60,12); L38S2 = TP10F2(1:60,12); L38S3 = TP10F2(1:60,12); %114
 L39S1 = TP11F1(1:60,12); L39S2 = TP11F2(1:60,12); L39S3 = TP11F3(1:60,12); %117
 L40S1 = TP12F1(1:60,12); L40S2 = TP12F1(1:60,12); L40S3 = TP12F1(1:60,12); %120
 L41S1 = TP13F3(45:104,12); L41S2 = TP13F2(1:60,12); L41S3 = TP13F3(45:104,12); %123
 L42S1 = TP14F1(1:60,12); L42S2 = TP14F2(1:60,12); L42S3 = TP14F1(1:60,12); %126

%%%%%%%%%%%% Smoothing Data of Left Ankle movement

window=11;
 LSS(:,1)=smooth(L1S1>window/length(L1S1),'rloess');LSS(:,2)=smooth(L1S2>window/length(L1S2),'rloess');
 LSS(:,3)=smooth(L1S3>window/length(L1S3),'rloess');LSS(:,4)=smooth(L2S1>window/length(L2S1),'rloess');
 LSS(:,5)=smooth(L2S2>window/length(L2S2),'rloess');LSS(:,6)=smooth(L2S3>window/length(L2S3),'rloess');
 LSS(:,7)=smooth(L3S1>window/length(L3S1),'rloess');LSS(:,8)=smooth(L3S2>window/length(L3S2),'rloess');
 LSS(:,9)=smooth(L3S3>window/length(L3S3),'rloess');LSS(:,10)=smooth(L4S1>window/length(L4S1),'rloess');
 LSS(:,11)=smooth(L4S2>window/length(L4S2),'rloess');LSS(:,12)=smooth(L4S3>window/length(L4S3),'rloess');
 LSS(:,13)=smooth(L5S1>window/length(L5S1),'rloess');LSS(:,14)=smooth(L5S2>window/length(L5S2),'rloess');
 LSS(:,15)=smooth(L5S3>window/length(L5S3),'rloess');LSS(:,16)=smooth(L6S1>window/length(L6S1),'rloess');
 LSS(:,17)=smooth(L6S2>window/length(L6S2),'rloess');LSS(:,18)=smooth(L6S3>window/length(L6S3),'rloess');
 LSS(:,19)=smooth(L7S1>window/length(L7S1),'rloess');LSS(:,20)=smooth(L7S2>window/length(L7S2),'rloess');
 LSS(:,21)=smooth(L7S3>window/length(L7S3),'rloess');LSS(:,22)=smooth(L8S1>window/length(L8S1),'rloess');
 LSS(:,23)=smooth(L8S2>window/length(L8S2),'rloess');LSS(:,24)=smooth(L8S3>window/length(L8S3),'rloess');
 LSS(:,25)=smooth(L9S1>window/length(L9S1),'rloess');LSS(:,26)=smooth(L9S2>window/length(L9S2),'rloess');
 LSS(:,27)=smooth(L9S3>window/length(L9S3),'rloess');LSS(:,28)=smooth(L10S1>window/length(L10S1),'rloess');
 LSS(:,29)=smooth(L10S2>window/length(L10S2),'rloess');LSS(:,30)=smooth(L10S3>window/length(L10S3),'rloess');
 LSS(:,31)=smooth(L11S1>window/length(L11S1),'rloess');LSS(:,32)=smooth(L11S2>window/length(L11S2),'rloess');
 LSS(:,33)=smooth(L11S3>window/length(L11S3),'rloess');LSS(:,34)=smooth(L12S1>window/length(L12S1),'rloess');
 LSS(:,35)=smooth(L12S2>window/length(L12S2),'rloess');LSS(:,36)=smooth(L12S3>window/length(L12S3),'rloess');
 LSS(:,37)=smooth(L13S1>window/length(L13S1),'rloess');LSS(:,38)=smooth(L13S2>window/length(L13S2),'rloess');
 LSS(:,39)=smooth(L13S3>window/length(L13S3),'rloess');LSS(:,40)=smooth(L14S1>window/length(L14S1),'rloess');
 LSS(:,41)=smooth(L14S2>window/length(L14S2),'rloess');LSS(:,42)=smooth(L14S3>window/length(L14S3),'rloess');
 LSS(:,43)=smooth(L15S1>window/length(L15S1),'rloess');LSS(:,44)=smooth(L15S2>window/length(L15S2),'rloess');
 LSS(:,45)=smooth(L15S3>window/length(L15S3),'rloess');LSS(:,46)=smooth(L16S1>window/length(L16S1),'rloess');
 LSS(:,47)=smooth(L16S2>window/length(L16S2),'rloess');LSS(:,48)=smooth(L16S3>window/length(L16S3),'rloess');
 LSS(:,49)=smooth(L17S1>window/length(L17S1),'rloess');LSS(:,50)=smooth(L17S2>window/length(L17S2),'rloess');

```

LSS(:,51)=smooth(L17S3>window/length(L17S3),'rloess');LSS(:,52)=smooth(L18S1>window/length(L18S1),'rloess');
LSS(:,53)=smooth(L18S2>window/length(L18S2),'rloess');LSS(:,54)=smooth(L18S3>window/length(L18S3),'rloess');
LSS(:,55)=smooth(L19S1>window/length(L19S1),'rloess');LSS(:,56)=smooth(L19S2>window/length(L19S2),'rloess');
LSS(:,57)=smooth(L19S3>window/length(L19S3),'rloess');LSS(:,58)=smooth(L20S1>window/length(L20S1),'rloess');
LSS(:,59)=smooth(L20S2>window/length(L20S2),'rloess');LSS(:,60)=smooth(L20S3>window/length(L20S3),'rloess');
LSS(:,61)=smooth(L21S1>window/length(L21S1),'rloess');LSS(:,62)=smooth(L21S2>window/length(L21S2),'rloess');
LSS(:,63)=smooth(L21S3>window/length(L21S3),'rloess');LSS(:,64)=smooth(L22S1>window/length(L22S1),'rloess');
LSS(:,65)=smooth(L22S2>window/length(L22S2),'rloess');LSS(:,66)=smooth(L22S3>window/length(L22S3),'rloess');
LSS(:,67)=smooth(L23S1>window/length(L23S1),'rloess');LSS(:,68)=smooth(L23S2>window/length(L23S2),'rloess');
LSS(:,69)=smooth(L23S3>window/length(L23S3),'rloess');LSS(:,70)=smooth(L24S1>window/length(L24S1),'rloess');
LSS(:,71)=smooth(L24S2>window/length(L24S2),'rloess');LSS(:,72)=smooth(L24S3>window/length(L24S3),'rloess');
LSS(:,73)=smooth(L25S1>window/length(L25S1),'rloess');LSS(:,74)=smooth(L25S2>window/length(L25S2),'rloess');
LSS(:,75)=smooth(L25S3>window/length(L25S3),'rloess');LSS(:,76)=smooth(L26S1>window/length(L26S1),'rloess');
LSS(:,77)=smooth(L26S2>window/length(L26S2),'rloess');LSS(:,78)=smooth(L26S3>window/length(L26S3),'rloess');
LSS(:,79)=smooth(L27S1>window/length(L27S1),'rloess');LSS(:,80)=smooth(L27S2>window/length(L27S2),'rloess');
LSS(:,81)=smooth(L27S3>window/length(L27S3),'rloess');LSS(:,82)=smooth(L28S1>window/length(L28S1),'rloess');
LSS(:,83)=smooth(L28S2>window/length(L28S2),'rloess');LSS(:,84)=smooth(L28S3>window/length(L28S3),'rloess');
LSS(:,85)=smooth(L29S1>window/length(L29S1),'rloess');LSS(:,86)=smooth(L29S2>window/length(L29S2),'rloess');
LSS(:,87)=smooth(L29S3>window/length(L29S3),'rloess');LSS(:,88)=smooth(L30S1>window/length(L30S1),'rloess');
LSS(:,89)=smooth(L30S2>window/length(L30S2),'rloess');LSS(:,90)=smooth(L30S3>window/length(L30S3),'rloess');
LSS(:,91)=smooth(L31S1>window/length(L31S1),'rloess');LSS(:,92)=smooth(L31S2>window/length(L31S2),'rloess');
LSS(:,93)=smooth(L31S3>window/length(L31S3),'rloess');LSS(:,94)=smooth(L32S1>window/length(L32S1),'rloess');
LSS(:,95)=smooth(L32S2>window/length(L32S2),'rloess');LSS(:,96)=smooth(L32S3>window/length(L32S3),'rloess');
LSS(:,97)=smooth(L33S1>window/length(L33S1),'rloess');LSS(:,98)=smooth(L33S2>window/length(L33S2),'rloess');
LSS(:,99)=smooth(L33S3>window/length(L33S3),'rloess');LSS(:,100)=smooth(L34S1>window/length(L34S1),'rloess');
LSS(:,101)=smooth(L34S2>window/length(L34S2),'rloess');LSS(:,102)=smooth(L34S3>window/length(L34S3),'rloess');
LSS(:,103)=smooth(L35S1>window/length(L35S1),'rloess');LSS(:,104)=smooth(L35S2>window/length(L35S2),'rloess');
LSS(:,105)=smooth(L35S3>window/length(L35S3),'rloess');LSS(:,106)=smooth(L36S1>window/length(L36S1),'rloess');
LSS(:,107)=smooth(L36S2>window/length(L36S2),'rloess');LSS(:,108)=smooth(L36S3>window/length(L36S3),'rloess');
LSS(:,109)=smooth(L37S1>window/length(L37S1),'rloess');LSS(:,110)=smooth(L37S2>window/length(L37S2),'rloess');
LSS(:,111)=smooth(L37S3>window/length(L37S3),'rloess');LSS(:,112)=smooth(L38S1>window/length(L38S1),'rloess');
LSS(:,113)=smooth(L38S2>window/length(L38S2),'rloess');LSS(:,114)=smooth(L38S3>window/length(L38S3),'rloess');
LSS(:,115)=smooth(L39S1>window/length(L39S1),'rloess');LSS(:,116)=smooth(L39S2>window/length(L39S2),'rloess');
LSS(:,117)=smooth(L39S3>window/length(L39S3),'rloess');LSS(:,118)=smooth(L40S1>window/length(L40S1),'rloess');
LSS(:,119)=smooth(L40S2>window/length(L40S2),'rloess');LSS(:,120)=smooth(L40S3>window/length(L40S3),'rloess');
LSS(:,121)=smooth(L41S1>window/length(L41S1),'rloess');LSS(:,122)=smooth(L41S2>window/length(L41S2),'rloess');
LSS(:,123)=smooth(L41S3>window/length(L41S3),'rloess');LSS(:,124)=smooth(L42S1>window/length(L42S1),'rloess');
LSS(:,125)=smooth(L42S2>window/length(L42S2),'rloess');LSS(:,126)=smooth(L42S3>window/length(L42S3),'rloess');

```

%%%%%%%%%% Right Ankle tracking on z-axis

```

R1S1 = TP1S1(40:99,21); R1S2 = TP1S2(40:99,21); R1S3 = TP1S3(40:99,21);
R2S1 = TP2S1(40:99,21); R2S2 = TP2S2(40:99,21); R2S3 = TP2S3(40:99,21);
R3S1 = TP3S1(40:99,21); R3S2 = TP3S2(40:99,21); R3S3 = TP3S3(40:99,21);
R4S1 = TP4S1(40:99,21); R4S2 = TP4S2(40:99,21); R4S3 = TP4S3(40:99,21);
R5S1 = TP5S1(40:99,21); R5S2 = TP5S2(40:99,21); R5S3 = TP5S3(25:84,21);
R6S1 = TP6S1(40:99,21); R6S2 = TP6S2(40:99,21); R6S3 = TP6S3(40:99,21);
R7S1 = TP7S1(20:79,21); R7S2 = TP7S2(40:99,21); R7S3 = TP7S3(20:79,21);
R8S1 = TP8S1(35:94,21); R8S2 = TP8S2(40:99,21); R8S3 = TP8S3(40:99,21);
R9S1 = TP9S1(40:99,21); R9S2 = TP9S2(40:99,21); R9S3 = TP9S3(40:99,21);
R10S1 = TP10S1(40:99,21); R10S2 = TP10S2(40:99,21); R10S3 = TP10S3(40:99,21);
R11S1 = TP11S1(40:99,21); R11S2 = TP11S2(40:99,21); R11S3 = TP11S3(40:99,21);
R12S1 = TP12S1(40:99,21); R12S2 = TP12S2(40:99,21); R12S3 = TP12S3(40:99,21);
R13S1 = TP13S1(40:99,21); R13S2 = TP13S2(30:89,21); R13S3 = TP13S3(40:99,21);
R14S1 = TP14S1(100:159,21); R14S2 = TP14S2(40:99,21); R14S3 = TP14S3(100:159,21);
R15S1 = TP1N1(30:89,21); R15S2 = TP1N2(40:99,21); R15S3 = TP1N3(40:99,21);
R16S1 = TP2N1(40:99,21); R16S2 = TP2N2(40:99,21); R16S3 = TP2N3(40:99,21);
R17S1 = TP3N1(25:84,21); R17S2 = TP3N2(40:99,21); R17S3 = TP3N3(40:99,21);
R18S1 = TP4N1(40:99,21); R18S2 = TP4N2(40:99,21); R18S3 = TP4N3(40:99,21);
R19S1 = TP5N1(40:99,21); R19S2 = TP5N2(9:68,21); R19S3 = TP5N3(12:71,21);
R20S1 = TP6N1(40:99,21); R20S2 = TP6N2(29:88,21); R20S3 = TP6N3(40:99,21);
R21S1 = TP7N1(40:99,21); R21S2 = TP7N2(30:89,21); R21S3 = TP7N3(40:99,21);
R22S1 = TP8N1(40:99,21); R22S2 = TP8N2(40:99,21); R22S3 = TP8N3(40:99,21);
R23S1 = TP9N1(180:239,21); R23S2 = TP9N2(40:99,21); R23S3 = TP9N3(40:99,21);

```

R24S1= TP10N1(40:99,21); R24S2 = TP10N2(40:99,21); R24S3 = TP10N3(40:99,21);
R25S1 = TP11N1(40:99,21); R25S2 = TP11N2(10:69,21); R25S3 = TP11N3(40:99,21);
R26S1 = TP12N1(40:99,21); R26S2 = TP12N2(40:99,21); R26S3 = TP12N3(40:99,21);
R27S1 = TP13N1(40:99,21); R27S2 = TP13N2(40:99,21); R27S3 = TP13N1(40:99,21);
R28S1 = TP14N1(60:119,21); R28S2 = TP14N2(100:159,21); R28S3 = TP14N1(60:119,21);
R29S1 = TP1F1(1:60,21); R29S2 = TP1F2(1:60,21); R29S3 = TP1F2(1:60,21);
R30S1 = TP2F1(1:60,21); R30S2 = TP2F2(1:60,21); R30S3 = TP2F3(1:60,21);
R31S1 = TP3F1(1:60,21); R31S2 = TP3F2(1:60,21); R31S3 = TP3F2(1:60,21);
R32S1 = TP4F1(1:60,21); R32S2 = TP4F2(1:60,21); R32S3 = TP4F3(1:60,21);
R33S1 = TP5F1(1:60,21); R33S2 = TP5F1(1:60,21); R33S3 = TP5F1(1:60,21);
R34S1 = TP6F1(1:60,21); R34S2 = TP6F1(1:60,21); R34S3 = TP6F1(1:60,21);
R35S1 = TP7F1(1:60,21); R35S2 = TP7F2(1:60,21); R35S3 = TP7F1(1:60,21);
R36S1 = TP8F1(1:60,21); R36S2 = TP8F2(1:60,21); R36S3 = TP8F2(1:60,21);
R37S1 = TP9F3(188:247,21); R37S2 = TP9F2(40:99,21); R37S3 = TP9F3(20:79,21);
R38S1 = TP10F1(1:60,21); R38S2 = TP10F2(1:60,21); R38S3 = TP10F2(1:60,21);
R39S1 = TP11F1(1:60,21); R39S2 = TP11F2(1:60,21); R39S3 = TP11F3(1:60,21);
R40S1 = TP12F1(1:60,21); R40S2 = TP12F1(1:60,21); R40S3 = TP12F1(1:60,21);
R41S1 = TP13F3(45:104,21); R41S2 = TP13F2(1:60,21); R41S3 = TP13F3(45:104,21);
R42S1 = TP14F1(1:60,21); R42S2 = TP14F2(1:60,21); R42S3 = TP14F1(1:60,21);

%%%%%%%%%%%% Smoothing Data of Rightt Ankle movement
window=11;

RSS(:,1)=smooth(R1S1,window/length(R1S1),'rloess');RSS(:,2)=smooth(R1S2,window/length(R1S2),'rloess');
RSS(:,3)=smooth(R1S3,window/length(R1S3),'rloess');RSS(:,4)=smooth(R2S1,window/length(R2S1),'rloess');
RSS(:,5)=smooth(R2S2,window/length(R2S2),'rloess');RSS(:,6)=smooth(R2S3,window/length(R2S3),'rloess');
RSS(:,7)=smooth(R3S1,window/length(R3S1),'rloess');RSS(:,8)=smooth(R3S2,window/length(R3S2),'rloess');
RSS(:,9)=smooth(R3S3,window/length(R3S3),'rloess');RSS(:,10)=smooth(R4S1,window/length(R4S1),'rloess');
RSS(:,11)=smooth(R4S2,window/length(R4S2),'rloess');RSS(:,12)=smooth(R4S3,window/length(R4S3),'rloess');
RSS(:,13)=smooth(R5S1,window/length(R5S1),'rloess');RSS(:,14)=smooth(R5S2,window/length(R5S2),'rloess');
RSS(:,15)=smooth(R5S3,window/length(R5S3),'rloess');RSS(:,16)=smooth(R6S1,window/length(R6S1),'rloess');
RSS(:,17)=smooth(R6S2,window/length(R6S2),'rloess');RSS(:,18)=smooth(R6S3,window/length(R6S3),'rloess');
RSS(:,19)=smooth(R7S1,window/length(R7S1),'rloess');RSS(:,20)=smooth(R7S2,window/length(R7S2),'rloess');
RSS(:,21)=smooth(R7S3,window/length(R7S3),'rloess');RSS(:,22)=smooth(R8S1,window/length(R8S1),'rloess');
RSS(:,23)=smooth(R8S2,window/length(R8S2),'rloess');RSS(:,24)=smooth(R8S3,window/length(R8S3),'rloess');
RSS(:,25)=smooth(R9S1,window/length(R9S1),'rloess');RSS(:,26)=smooth(R9S2,window/length(R9S2),'rloess');
RSS(:,27)=smooth(R9S3,window/length(R9S3),'rloess');RSS(:,28)=smooth(R10S1,window/length(R10S1),'rloess');
RSS(:,29)=smooth(R10S2,window/length(R10S2),'rloess');RSS(:,30)=smooth(R10S3,window/length(R10S3),'rloess');
RSS(:,31)=smooth(R11S1,window/length(R11S1),'rloess');RSS(:,32)=smooth(R11S2,window/length(R11S2),'rloess');
RSS(:,33)=smooth(R11S3,window/length(R11S3),'rloess');RSS(:,34)=smooth(R12S1,window/length(R12S1),'rloess');
RSS(:,35)=smooth(R12S2,window/length(R12S2),'rloess');RSS(:,36)=smooth(R12S3,window/length(R12S3),'rloess');
RSS(:,37)=smooth(R13S1,window/length(R13S1),'rloess');RSS(:,38)=smooth(R13S2,window/length(R13S2),'rloess');
RSS(:,39)=smooth(R13S3,window/length(R13S3),'rloess');RSS(:,40)=smooth(R14S1,window/length(R14S1),'rloess');
RSS(:,41)=smooth(R14S2,window/length(R14S2),'rloess');RSS(:,42)=smooth(R14S3,window/length(R14S3),'rloess');
RSS(:,43)=smooth(R15S1,window/length(R15S1),'rloess');RSS(:,44)=smooth(R15S2,window/length(R15S2),'rloess');
RSS(:,45)=smooth(R15S3,window/length(R15S3),'rloess');RSS(:,46)=smooth(R16S1,window/length(R16S1),'rloess');
RSS(:,47)=smooth(R16S2,window/length(R16S2),'rloess');RSS(:,48)=smooth(R16S3,window/length(R16S3),'rloess');
RSS(:,49)=smooth(R17S1,window/length(R17S1),'rloess');RSS(:,50)=smooth(R17S2,window/length(R17S2),'rloess');
RSS(:,51)=smooth(R17S3,window/length(R17S3),'rloess');RSS(:,52)=smooth(R18S1,window/length(R18S1),'rloess');
RSS(:,53)=smooth(R18S2,window/length(R18S2),'rloess');RSS(:,54)=smooth(R18S3,window/length(R18S3),'rloess');
RSS(:,55)=smooth(R19S1,window/length(R19S1),'rloess');RSS(:,56)=smooth(R19S2,window/length(R19S2),'rloess');
RSS(:,57)=smooth(R19S3,window/length(R19S3),'rloess');RSS(:,58)=smooth(R20S1,window/length(R20S1),'rloess');
RSS(:,59)=smooth(R20S2,window/length(R20S2),'rloess');RSS(:,60)=smooth(R20S3,window/length(R20S3),'rloess');
RSS(:,61)=smooth(R21S1,window/length(R21S1),'rloess');RSS(:,62)=smooth(R21S2,window/length(R21S2),'rloess');
RSS(:,63)=smooth(R21S3,window/length(R21S3),'rloess');RSS(:,64)=smooth(R22S1,window/length(R22S1),'rloess');
RSS(:,65)=smooth(R22S2,window/length(R22S2),'rloess');RSS(:,66)=smooth(R22S3,window/length(R22S3),'rloess');
RSS(:,67)=smooth(R23S1,window/length(R23S1),'rloess');RSS(:,68)=smooth(R23S2,window/length(R23S2),'rloess');
RSS(:,69)=smooth(R23S3,window/length(R23S3),'rloess');RSS(:,70)=smooth(R24S1,window/length(R24S1),'rloess');
RSS(:,71)=smooth(R24S2,window/length(R24S2),'rloess');RSS(:,72)=smooth(R24S3,window/length(R24S3),'rloess');
RSS(:,73)=smooth(R25S1,window/length(R25S1),'rloess');RSS(:,74)=smooth(R25S2,window/length(R25S2),'rloess');
RSS(:,75)=smooth(R25S3,window/length(R25S3),'rloess');RSS(:,76)=smooth(R26S1,window/length(R26S1),'rloess');
RSS(:,77)=smooth(R26S2,window/length(R26S2),'rloess');RSS(:,78)=smooth(R26S3,window/length(R26S3),'rloess');
RSS(:,79)=smooth(R27S1,window/length(R27S1),'rloess');RSS(:,80)=smooth(R27S2,window/length(R27S2),'rloess');
RSS(:,81)=smooth(R27S3,window/length(R27S3),'rloess');RSS(:,82)=smooth(R28S1,window/length(R28S1),'rloess');

```

RSS(:,83)=smooth(R28S2>window/length(R28S2),'rloess');RSS(:,84)=smooth(R28S3>window/length(R28S3),'rloess');

RSS(:,85)=smooth(R29S1>window/length(R29S1),'rloess');RSS(:,86)=smooth(R29S2>window/length(R29S2),'rloess');
RSS(:,87)=smooth(R29S3>window/length(R29S3),'rloess');RSS(:,88)=smooth(R30S1>window/length(R30S1),'rloess');
RSS(:,89)=smooth(R30S2>window/length(R30S2),'rloess');RSS(:,90)=smooth(R30S3>window/length(R30S3),'rloess');
RSS(:,91)=smooth(R31S1>window/length(R31S1),'rloess');RSS(:,92)=smooth(R31S2>window/length(R31S2),'rloess');
RSS(:,93)=smooth(R31S3>window/length(R31S3),'rloess');RSS(:,94)=smooth(R32S1>window/length(R32S1),'rloess');
RSS(:,95)=smooth(R32S2>window/length(R32S2),'rloess');RSS(:,96)=smooth(R32S3>window/length(R32S3),'rloess');
RSS(:,97)=smooth(R33S1>window/length(R33S1),'rloess');RSS(:,98)=smooth(R33S2>window/length(R33S2),'rloess');
RSS(:,99)=smooth(R33S3>window/length(R33S3),'rloess');RSS(:,100)=smooth(R34S1>window/length(R34S1),'rloess');
RSS(:,101)=smooth(R34S2>window/length(R34S2),'rloess');RSS(:,102)=smooth(R34S3>window/length(R34S3),'rloess');
RSS(:,103)=smooth(R35S1>window/length(R35S1),'rloess');RSS(:,104)=smooth(R35S2>window/length(R35S2),'rloess');
RSS(:,105)=smooth(R35S3>window/length(R35S3),'rloess');RSS(:,106)=smooth(R36S1>window/length(R36S1),'rloess');
RSS(:,107)=smooth(R36S2>window/length(R36S2),'rloess');RSS(:,108)=smooth(R36S3>window/length(R36S3),'rloess');
RSS(:,109)=smooth(R37S1>window/length(R37S1),'rloess');RSS(:,110)=smooth(R37S2>window/length(R37S2),'rloess');
RSS(:,111)=smooth(R37S3>window/length(R37S3),'rloess');RSS(:,112)=smooth(R38S1>window/length(R38S1),'rloess');
RSS(:,113)=smooth(R38S2>window/length(R38S2),'rloess');RSS(:,114)=smooth(R38S3>window/length(R38S3),'rloess');
RSS(:,115)=smooth(R39S1>window/length(R39S1),'rloess');RSS(:,116)=smooth(R39S2>window/length(R39S2),'rloess');
RSS(:,117)=smooth(R39S3>window/length(R39S3),'rloess');RSS(:,118)=smooth(R40S1>window/length(R40S1),'rloess');
RSS(:,119)=smooth(R40S2>window/length(R40S2),'rloess');RSS(:,120)=smooth(R40S3>window/length(R40S3),'rloess');
RSS(:,121)=smooth(R41S1>window/length(R41S1),'rloess');RSS(:,122)=smooth(R41S2>window/length(R41S2),'rloess');
RSS(:,123)=smooth(R41S3>window/length(R41S3),'rloess');RSS(:,124)=smooth(R42S1>window/length(R42S1),'rloess');
RSS(:,125)=smooth(R42S2>window/length(R42S2),'rloess');RSS(:,126)=smooth(R42S3>window/length(R42S3),'rloess');

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% GAIT LENGTH GENERATION

```

```

for i=1:length(LSS) % No of participants (columns)
    for k=1:length(LSS(:,1)) % No of data points (Rows)

        gait(k,i)=abs(RSS(k,i)-LSS(k,i)); % gait step length signal in positive values
        gaitleng(k,i)=RSS(k,i)-LSS(k,i); % gait length signal in positive & negative values
    end
    step(i)=max(gait(:,i));
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% grouping of the gait signals g(t) according to speed (slow mid fast)

```

```

lo=0;mi=0;hi=0;
for i=1:length(TSS); % No of samples (trials [3]* participants[42])

    Dis(i)= abs(TSS(10,i)-TSS(end-5,i));
    speed(i)=Dis(i)/(length(TSS(10:end-5,i))/30); % (length(TSS(:,i)))
    if (speed(i) <= 0.55)
        lo=lo+1;
        low(lo)=i;
        spelo(lo)=speed(i);
        gaitlo(:,lo)=gait(:,i);
        gaitlenglo(:,lo)=gaitleng(:,i);
        LSS1(:,lo)=LSS(:,i);
        RSS1(:,lo)=RSS(:,i);
    elseif (speed(i) > 0.55 && speed(i) < 1)
        mi=mi+1;
        mid(mi)=i;
        spemi(mi)=speed(i);
        gaitmi(:,mi)=gait(:,i);
        gaitlengmi(:,mi)=gaitleng(:,i);
        LSS2(:,mi)=LSS(:,i);
        RSS2(:,mi)=RSS(:,i);
    elseif (speed(i) >= 1)
        hi=hi+1;
        high(hi)=i;
        spehi(hi)=speed(i);
        gaithi(:,hi)=gait(:,i);
    end
end

```

```

        gaitlenghi(:,hi)=gaitleng(:,i);
        LSS3(:,hi)=LSS(:,i);
        RSS3(:,hi)=RSS(:,i);
    end
end
Dia=[lo mi hi]; Di=min(Dia); % Di means the minimum size among [slow,normal,fast] speeds

%%%%%%%%%%%% calculation of gait step length & cadence
for i=1:length(gaitlo(1,:))
    a1=0;a2=0;a3=0;

    for k=2:length(gaitlo(:,1))-1

        if(gaitlo(k,i)>gaitlo(k+1,i)&&gaitlo(k,i)>gaitlo(k-1,i))
            a1=a1+1; %%% number of steps for low speed
            apeakL1(a1,i)=gaitlo(k,i); %%% peak values for low speed group
            numL(a1,i)=k; %%% value of frames (K)for low speed
        end
        if(gaitmi(k,i)>gaitmi(k+1,i)&& gaitmi(k,i)>gaitmi(k-1,i))
            a2=a2+1; %%% number of steps for meddil speed
            apeakM1(a2,i)=gaitmi(k,i);
            numM(a2,i)=k; %%% value of frames (K)for meddil speed
        end
        if(gaithi(k,i)>gaithi(k+1,i)&& gaithi(k,i)>gaithi(k-1,i))
            a3=a3+1; %%% number of steps for high speed
            apeakH1(a3,i)=gaithi(k,i);
            numH(a3,i)=k; %%% value of frames (K)for high speed
        end
    end
end

apeakL=sum(apeakL1);
apeakM=sum(apeakM1);
apeakH=sum(apeakH1);

%%%%%%%%%%%% To Eliminate Zeros from the data matrix
for i=1:length(gaitlo(1,:))
    a11=0;a22=0;a33=0;
    for k=1:length(apeakL1(:,1))

        if(apeakL1(k,i)>0)
            a11=a11+1;
        end
    end
    for k=1:length(apeakM1(:,1))
        if(apeakM1(k,i)>0)
            a22=a22+1;
        end
    end
    for k=1:length(apeakL1(:,1))
        if(apeakH1(k,i)>0)
            a33=a33+1;
        end
    end
    NL(i)=a11;
    NM(i)=a22;
    NH(i)=a33;
end

% ape(ape == 0) = NaN;
for i=1:Di
    a_stepL(i)=apeakL(i)/NL(i);

```

```

a_stepM(i)=apeakM(i)/NM(i);
a_stepH(i)=apeakH(i)/NH(i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gait Stride length calculation
a_stridL=a_stepL*2;
a_stridM=a_stepM*2;
a_stridH=a_stepH*2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gait cadence calculation
fsm=30;
for i=1:Di

    if(NL(i)==1)
        secL(i)=1;%(35/fsm);
    elseif (NL(i)==2)
        secL(i)=2;%(numL(2,i)-numL(1,i))/fsm);
    elseif (NL(i)==3)
        secL(i)=3;%(numL(3,i)-numL(1,i))/fsm);
    end

    if(NM(i)==1)
        secM(i)=1;%(35/fsm);
    elseif (NM(i)==2)
        secM(i)=2;%(numM(2,i)-numM(1,i))/fsm);
    elseif (NM(i)==3)
        secM(i)=3;%(numM(3,i)-numM(1,i))/fsm);
    elseif (NM(i)==4)
        secM(i)=4;%(numM(4,i)-numM(1,i))/fsm);
    end

    if(NH(i)==1)
        secH(i)=1;%(35/fsm);
    elseif (NH(i)==2)
        secH(i)=2;%(numH(2,i)-numH(1,i))/fsm);
    elseif (NH(i)==3)
        secH(i)=3;%(numH(3,i)-numH(1,i))/fsm);
    elseif (NH(i)==4)
        secH(i)=4;%(numH(4,i)-numH(1,i))/fsm);
    end

    a_cadecL(i)=secL(i)*(60/2); % divission by 2 becasue the number of frames is 60 which means 2 sec
    a_cadecM(i)=secM(i)*(60/2);
    a_cadecH(i)=secH(i)*(60/2);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

D-2 MATLAB code for gait features extraction using spatiotemporal gait analysis to calculate time of stance and swing stages, and time of double support legs.

```

%% to avoid the repetition, the import data, filtering data, gait signal generation are shown in D-1, so will start from
%% grouping data into 3 categories based on type of walk speed

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% grouping of the gait signals g(t) according to speed (slow mid fast)

lo=0;mi=0;hi=0;
for i=1:length(TSS); % No of samples (trials [3]* participants[42])

```

```

Dis(i)= abs (TSS(10,i)-TSS(end-5,i));
speed(i)=Dis(i)/(length (TSS(10:end-5,i)))/30);           %(length(TSS(:,i))
if (speed(i) <= 0.55)
    lo=lo+1;
    low(lo)=i;
    spelo(lo)=speed(i);
    gaitlo(:,lo)=gait(:,i);
    gaitlenglo(:,lo)=gaitleng(:,i);
    LSS1(:,lo)=LSS(:,i);
    RSS1(:,lo)=RSS(:,i);
elseif (speed(i) > 0.55 && speed(i) < 1)
    mi=mi+1;
    mid(mi)=i;
    spemi(mi)=speed(i);
    gaitmi(:,mi)=gait(:,i);
    gaitlengmi(:,mi)=gaitleng(:,i);
    LSS2(:,mi)=LSS(:,i);
    RSS2(:,mi)=RSS(:,i);
elseif (speed(i) >= 1)
    hi=hi+1;
    high(hi)=i;
    spehi(hi)=speed(i);
    gaithi(:,hi)=gait(:,i);
    gaitlenghi(:,hi)=gaitleng(:,i);
    LSS3(:,hi)=LSS(:,i);
    RSS3(:,hi)=RSS(:,i);
end
end
Dia=[lo mi hi]; Di=min(Dia); % Di means the minimum size among [slow, normal, fast] speeds

%%%%%%%%%%To calculate the value of frame (k) @ peak value of gait signal

for i=1:Di
    sd1=0;sd2=0;sd3=0;
    for k=2:length(gaitlenglo(:,1))-1
        if(gaitlenglo(k,i)>gaitlenglo(k+1,i)&& gaitlenglo(k,i)>gaitlenglo(k-1,i))
            sd1=sd1+1;
            s1(sd1,i)=k;
        end
    end

    for k=1:length(gaitlengmi(:,1))-1
        if(gaitlengmi(k,i)>gaitlengmi(k+1,i)&& gaitlengmi(k,i)>gaitlengmi(k-1,i))
            sd2=sd2+1;
            s2(sd2,i)=k;
        end
    end

    for k=2:length(gaitlenghi(:,1))-1
        if(gaitlenghi(k,i)>gaitlenghi(k+1,i)&& gaitlenghi(k,i)>gaitlenghi(k-1,i))
            sd3=sd3+1;
            s3(sd3,i)=k;
        end
    end
end

%%%%%%%%% determine the Two values of peak points of gait length (case of Pos & Neg signal)
for i=1:Di
    if(s1(1,i)==0 && s1(2,i)==0)
        s1(1,i)=2;
        s1(2,i)=(length(gaitlenglo(:,i)))-1;
    end
end

```

```

elseif(s1(1,i)> (length(gaitlenglo(:,i)))/2 && s1(2,i)== 0)
    s1(2,i)=s1(1,i);
    s1(1,i)=2;

    elseif(s1(1,i)<(length(gaitlenglo(:,i)))/2 && s1(1,i)> 0 && s1(2,i)== 0)
    s1(2,i)=(length(gaitlenglo(:,i)))-1;

    elseif(s1(2,i)<(length(gaitlenglo(:,i)))/3 && s1(1,i)> 0 )
    s1(2,i)=(length(gaitlenglo(:,i)))-1;

end

end

%%%%%%%%%%%% stance stage time calculation

for i=1:Di
    cycLR(:,i)=abs(diff(RSS1(:,i)));
    cycLL(:,i)=abs(diff(LSS1(:,i)));
    cycMR(:,i)=abs(diff(RSS2(:,i)));
    cycML(:,i)=abs(diff(LSS2(:,i)));
    cycHR(:,i)=abs(diff(RSS3(:,i)));
    cycHL(:,i)=abs(diff(LSS3(:,i)));

    DsupL(:,i)=abs(diff(gaitlenglo(:,i)));
    DsupM(:,i)=abs(diff(gaitlengmi(:,i)));
    DsupH(:,i)=abs(diff(gaitlenghi(:,i)));
end

for i=1:Di
    LR=0;LL=0;MR=0;ML=0;HR=0;HL=0;LR1=0;LL1=0;MR1=0;ML1=0;HR1=0;HL1=0;
    for k=1:length(cycLR(:,1))
        %%%%%%%%% Swing & Stance in low speed
        if(cycLR(k,i)<=0.0099 && k>=s1(1,i) && k<=s1(2,i))
            LR=LR+1;
            stancLR(LR,i)=k;
            elseif (cycLR(k,i)>0.0099 && k>=s1(1,i) && k<=s1(2,i))
            LR1=LR1+1;
            swingLR(LR1,i)=k;
            end

        if(cycLL(k,i)<=0.0099 && k>=s1(1,i) && k<=s1(2,i))
            LL=LL+1;
            stancLL(LL,i)=k;
            elseif (cycLL(k,i)>0.0099 && k>=s1(1,i) && k<=s1(2,i))
            LL1=LL1+1;
            swingLL(LL1,i)=k;
            end

        %%%%%%%%% Swing & Stance in mid speed
        if(cycMR(k,i)<=0.0099 && k >=s2(1,i) && k <=s2(2,i))
            MR=MR+1;
            stancMR(MR,i)=k;
            elseif (cycMR(k,i)>0.0099 && k>=s2(1,i) && k<=s2(2,i))
            MR1=MR1+1;
            swingMR(MR1,i)=k;
            end

        if(cycML(k,i)<=0.0099 && k>=s2(1,i) && k<=s2(2,i))
            ML=ML+1;
            stancML(ML,i)=k;
            elseif (cycML(k,i)>0.0099 && k>=s2(1,i) && k<=s2(2,i))
            ML1=ML1+1;

```

```

swingML(ML1,i)=k;
end
%%%%%%%%%%%%% Swing & Stance in high speed
if(cycHR(k,i)<=0.0099 && k >=s3(1,i) && k<=s3(2,i))
HR=HR+1;
stancHR(HR,i)=k;
elseif (cycHR(k,i)>0.0099 && k >=s3(1,i) && k<=s3(2,i))
HR1=HR+1;
swingHR(HR1,i)=k;
end

if(cycHL(k,i)<=0.0099 && k >=s3(1,i) && k<=s3(2,i))
HL=HL+1;
stancHL(HL,i)=k;
elseif (cycHL(k,i)>0.0099 && k >=s3(1,i) && k<=s3(2,i))
HL1=HL+1;
swingHL(HL1,i)=k;
end
end
nustLR(i)=LR;
nuswLR(i)=LR1;
nustLL(i)=LL;
nuswLL(i)=LL1;
nustMR(i)=MR;
nuswMR(i)=MR1;
nustML(i)=ML;
nuswML(i)=ML1;
nustHR(i)=HR;
nuswHR(i)=HR1;
nustHL(i)=HL;
nuswHL(i)=HL1;

end

%%%%%%%%%%%%% Double support time calculation

for i=1:Di
ll=0;mm=0;hh=0;
for k=1:length(DsupL(:,1))
if(DsupL(k,i)<=0.009 && k>=s1(1,i) && k<=s1(2,i) && gaitlenglo(k,i)<0)
ll=ll+1;
DsL(ll,i)=k;
end

if(DsupM(k,i)<=0.0097 && k>=s2(1,i) && k<=s2(2,i)&& gaitlengmi(k,i)<0)
mm=mm+1;
DsM(mm,i)=k;
end

if(DsupH(k,i)<=0.0097 && k>=s3(1,i) && k<=s3(2,i)&& gaitlenghi(k,i)<0)
hh=hh+1;
DsH(hh,i)=k;
end
end
end
DS1(i)=ll;
DS2(i)=mm;
DS3(i)=hh;
end

%%%%%%%%%%%%%

```

D-3 MATLAB code for extracting gait features using AM tech and classifying the features using CE tech.

%% the importing data, smoothing data are shown in D-1, so will start with %% grouping data into 3 categories including slow, normal and fast walk %% speed

%% specify 3 groups of gait signal $g(t)$ according to speed s $(s \leq 0.6), (0.6 < s < 1.1), (s > 1.1)$

```
lo=0;mi=0;hi=0;
for i=1:length(TSS); % No of samples (trials [3]* participants[42])

    Dis(i)= abs(TSS(10,i)-TSS(end-5,i));
    speed(i)=Dis(i)/(length(TSS(10:end-5,i))/30); % (length(TSS(:,i))
    if (speed(i) <= 0.60)
        lo=lo+1;
        low(lo)=i;
        spelo(lo)=speed(i);
        gaitlo(:,lo)=gait(:,i);
    elseif (speed(i) > 0.60 && speed(i) < 1.1)
        mi=mi+1;
        mid(mi)=i;
        spemi(mi)=speed(i);
        gaitmi(:,mi)=gait(:,i);
    elseif (speed(i) >= 1.1)
        hi=hi+1;
        high(hi)=i;
        spehi(hi)=speed(i);
        gaithi(:,hi)=gait(:,i);
    end
end

Dia=[lo mi hi]; Di=min(Dia); % Di means the minimum size among [slow,normal,fast] speeds
```

%% Modified gait signal using AM modulation

```
Ac=1 ;ka=.17;fc=25;
N= length(gaitlo(:,1)); % N11 = ceil(N/2);
fs=100; ts=1/fs;
f = [0 : N-1] *fs/N; % frequency in Hz
t=[0 : N-1]/ fs;
xc=Ac*sin(2*pi*fc*t); % reference signal
ftxc=abs(fft(xc)/(N/2)); % spectrum of reference signal
for i=1:Di;
    am1(:,i)=xc.*(1+ka.*gaitlo(:,i));
    am2(:,i)=xc.*(1+ka.*gaitmi(:,i));
    am3(:,i)=xc.*(1+ka.*gaithi(:,i));
    ft1(:,i)=abs(fft(am1(:,i))/(N/2));
    ft2(:,i)=abs(fft(am2(:,i))/(N/2));
    ft3(:,i)=abs(fft(am3(:,i))/(N/2));
end
%% To calculate the frequency @ maximum amplitude (carrier freq)
for i=1 : N/2; % the size of carrier signal
    if (ftxc(i)==max(ftxc(1:N/2)))
FCm=i*(fs/(N))-(fs/N); % the minus means to make one point shift back because the f&t started from 0 rather 1
    end
end
```

%% To plot gait features of AM-modified gait in time/frequency domain for 3 all samples

```
%% figure (5); plot (f(1:N/2), ftxc (1:N/2))
% for v=1:40;
% ft11=ft1(:,v);ft22=ft2(:,v);ft33=ft3(:,v);
```

```

% figure (v);
% plot(f(1:N/2),ft1(1:N/2));
% title('fre_modulation in slow walk');
% axis tight
% figure (v+Di);
% plot(f(1:N/2),ft2(1:N/2));
% title('fre_modulation in normal walk');
% axis tight
% figure (v+Di+Di);
% plot(f(1:N/2),ft33(1:N/2));
% title('fre_modulation in fast walk');
% axis tight
% end
% figure (3);
% v=7; ft11=ft1(:,v);ft22=ft2(:,v);ft33=ft3(:,v);
% plot(f(1:N/2),ft33(1:N/2));
% % % % determining the frequencies and peaks amplitude on the spectrum % % % range.

di=length(ft1)/2; % Half frequencies axis which equals to (fs/2)
for k= 1 : Di % Di means the smallest size among three groups (slow, % normal fast speed)
    fl=0; fm=0; fh=0; ll=1;
    for i=2 : di
        if (ft1(i,k)>ft1(i-1,k) && ft1(i,k)>ft1(i+1,k))
            fl=fl+1;
            FL(fl,k)=i*(fs/N)-(fs/N); % to convert the bins into the frequencies
            Aml(fl,k)=ft1(i,k); % to find the magnitude @ above frequencies
        end
        if (ft2(i,k)>ft2(i-1,k) && ft2(i,k)>ft2(i+1,k))
            fm=fm+1;
            FM(fm,k)=i*(fs/N)-(fs/N);
            Amm(fm,k)=ft2(i,k);
        end
        if (ft3(i,k)>ft3(i-1,k) && ft3(i,k)>ft3(i+1,k))
            fh=fh+1;
            FH(fh,k)=i*(fs/N)-(fs/N);
            Amh(fh,k)=ft3(i,k);
        end
    end
end
% % % % % % % % % % (2)Reorder the frequencies that be centred on fc= 8Hz
% % % % % % % % % % (3) Reorder the amplitude that be centred on maximum value
%
Mn=min([length(Aml(:,1)) length(Amm(:,1)) length(Amh(:,1))]);
% the minimum size among three arraies of spectral
for i=1:Di % Di means the minimum size of data [slow,normal,fast] speeds
    for k=1:Mn
        if (FL(k,i) ==FCm)
            Lcf(i)=FL(k,i); % Freq @ centre spectral for low speed
            Llf(i)=FL(k-1,i); % Freq @ lower spectral for low speed
            Luf(i)=FL(k+1,i); % Freq @ upper spectral for low speed
            Lca(i)=Aml(k,i); % Amplitude of centre spectral for low speed
            Lla(i)=Aml(k-1,i)*(2/ka); % Amplitude level of spectral for low
            Lua(i)=Aml(k+1,i)*(2/ka); % Amplitude of upper spectral for low speed
            BWL(i)=Luf(i)-Llf(i); % BW of modified gait signal for low speed
            ModL(i)=(Lla(i)+Lua(i))/2/Lca(i); % Modification index
            EffL(i)=(ModL(i))^2/(2+(ModL(i))^2); % modification Efficiency
            PsL(i)= (Lla(i)+Lua(i)/2)*ka/(2.828); % Amplitude level of side loops
            % RatioL(i)= (PsL(i))/(( Lca(i))); % Amplitude level ratio of side loops to main loop
            PL(i)=(Lca(i)*0.707)*(1+(ModL(i)/2)); % Total Amplitude level of %modified signal
        end
    end
    if (FM(k,i) ==FCm)
        Mcf(i)=FM(k,i); % Freq @ centre spectral for normal speed
        Mlf(i)=FM(k-1,i); % Freq @ lower spectral for normal speed
        Muf(i)=FM(k+1,i); % Freq @ upper spectral for normal speed
        Mca(i)=Amm(k,i); % Amplitude of centre spectral for normal speed
        Mla(i)=Amm(k-1,i)*(2/ka); % Amplitude of lower spectral for normal speed
    end
end

```

```

Mua(i)=Amm(k+1,i)*(2/ka); % Amplitude of upper spectral for normal speed
BWM(i)=Muf(i)-Mlf(i); % Bandwidth of modified gait signal for normal speed
ModM(i)=(Mla(i)+Mua(i))/2/Mca(i); % Modification index
EffM(i)=(ModM(i)^2/(2+(ModM(i))^2)); % Efficiency of modulat
PsM(i)=(Mla(i)+Mua(i)/2)*ka/(2.828); %Amplitude level of sides
%RatioM(i)=(PsM(i))/((Mca(i))^2/2); % Amplitude level ratio of side-loops %to main loop
PtM(i)=(Mca(i)*0.707)*(1+(ModM(i)/2));% Total Amplitude level of modified %signal
end

if (FH(k,i) ==FCm)
    Hcf(i)=FH(k,i); % Freq @ centre spectral for fast speed
    Hlf(i)=FH(k-1,i); % Freq @ lower spectral for fast speed
    Huf(i)=FH(k+1,i); % Freq @ upper spectral for fast speed
    Hca(i)=Amh(k,i); % Amplitude of centre spectral for fast speed
    Hla(i)=Amh(k-1,i)*(2/ka);% Amplitude of lower spectral for fast speed
    Hua(i)=Amh(k+1,i)*(2/ka); % Amplitude of upper spectral for fast speed
    BWH(i)=Huf(i)-Hlf(i); % Bandwidth of modified gait signal for fast speed
    ModH(i)=(Hla(i)+Hua(i))/2/Hca(i); % Modification index
    EffH(i)=(ModH(i)^2/(2+(ModH(i))^2)); % Efficiency of modification
    PsH(i)=(Hla(i)+Hua(i)/2)*ka/(2.828); % Amplitude level of side loops
    %RatioH(i)=(PsH(i))/((Hca(i))^2/2); % the ratio of side-loops to main loop
    PtH(i)=(Hca(i)*0.707)*(1+(ModH(i)/2));% Total Amplitude level of modified signal
end
end

%
% % % % % % The 7 extracted features are converted into binary system
% % % % % % (1) upper side band frequency % % % % % % to ensure that all values of fg
be in fraction
D1 = sort( Luf); % FREQ gait features of class one
D2 = sort( Muf); % FREQ gait features of class two
D3 = sort( Huf);
% % % % % % (2) Bandwidth of modi=fied gait signal
D4 = sort( BWL); % Bandwidth features of class one
D5 = sort( BWM); % Bandwidth gait features of class two
D6 = sort( BWH);
% % % % % % (3) Modification index of modified gait signal
D7 = sort( ModL); % modulation index features of class one
D8 = sort( ModM); % modulation index gait features of class two
D9 = sort( ModH);
% % % % % % (4) Modification efficiency of modified gait signal
D10 = sort( EffL); % modulation efficiency features of class one
D11 = sort( EffM); % modulation efficiency gait features of class two
D12 = sort( EffH);
% % % % % % (5) power of sided-loops
D13 = sort( PsL); % power of sided-loops features of class one
D14 = sort( PsM); % power of sided-loops features of class two
D15 = sort( PsH); % power of sided-loops features of class three
% % % % % % (6) power Ratio
% D16 = sort( RatioL); % power Ratio features of class one
% D17 = sort( RatioM); % power Ratio features of class two
% D18 = sort( RatioH); % power Ratio features of class three
% % % % % % % % % % Lower side band frequency
D16 = sort( Llf); % FREQ gait features of class one
D17 = sort( Mlf); % FREQ gait features of class two
D18 = sort( Hlf);
% % % % % % (7) Total power
D19 = sort( PtL); % Total power features of class one
D20 = sort( PtM); % Total power features of class two
D21 = sort( PtH); % Total power features of class three

L1=length(D1);L2=length(D2);L3=length(D3); % the length of each class
N0=L1+L2+L3; % the length of all classes together (120 samples)
% % % % % % % % % % Getting the seven features as each columns contains 3 classes

for i=1:L1

```



```

if(i~=KK && i~=1)
TrainT(i,:)=Dset([1:end-window*i,end-window*(i-1)+1:end]);
end
if(i==1)
TrainT(i,:)=Dset([1:2*window,2*window+1:end-window]);

elseif(i==KK)
TrainT(i,:)=Dset([1+window:3*window,3*window+1:end]);
end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% claculating m1, m2, m3 from Training set.

o1=1;o2=1;o3=1;
for i=1:KK
for k=1:window1;
if (k==mc1)
MC1(o1)= TrainT(i,k);
o1=o1+1;
elseif (k==mc2)
MC2(o2)=TrainT(i,k);
o2=o2+1;
elseif (k==mc3)
MC3(o3)=TrainT(i,k);
o3=o3+1;
end
end
end
% m1=0.01;m2=0.038;m3=0.08;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Testing set generating
for i=1:KK

TesT(i,:)=Dset(window1+1-f:end-f) ;

f=f+window;

end

TesT01=TesT'; % Test set data in column form
% TesT01= sort(TesT0,'descend');
%
for i=1:KK;
TesT1(:,i)=abs(TesT01(:,i)-MC1(i));
TesT2(:,i)=abs(TesT01(:,i)-MC2(i));
TesT3(:,i)=abs(TesT01(:,i)-MC3(i));

end

test1_1(1:window,1)=TesT1(:,1);
test1_1(1+window:2*window,1)=TesT2(:,1);
test1_1(1+(2*window):3*window,1)=TesT3(:,1);
test1_1=test1_1-min(min(test1_1(:,:)));

test1_2(1:window,1)=TesT1(:,2);
test1_2(1+window:2*window,1)=TesT2(:,2);
test1_2(1+(2*window):3*window,1)=TesT3(:,2);
test1_2=test1_2-min(min(test1_2(:,:)));

test1_3(1:window,1)=TesT1(:,3);
test1_3(1+window:2*window,1)=TesT2(:,3);
test1_3(1+(2*window):3*window,1)=TesT3(:,3);
test1_3=test1_3-min(min(test1_3(:,:)));

test1_4(1:window,1)=TesT1(:,4);
test1_4(1+window:2*window,1)=TesT2(:,4);
test1_4(1+(2*window):3*window,1)=TesT3(:,4);

```



```

end

for h=1:2:2*num;
if (cod3_1(i,h)==0 && cod3_1(i,h+1) == 0);
tes3(i,g2)=40;
elseif (cod3_1(i,h)==1 && cod3_1(i,h+1) == 0);
tes3(i,g2)=30;
elseif (cod3_1(i,h)==0 && cod3_1(i,h+1) == 1);
tes3(i,g2)=20;
else (cod3_1(i,h)==1 && cod3_1(i,h+1) == 1);
tes3(i,g2)=10;

end
g2=g2+1;
end

for h=1:2:2*num;

if (cod4_1(i,h)==0 && cod4_1(i,h+1) == 0);
tes4(i,g3)=40;
elseif (cod4_1(i,h)==1 && cod4_1(i,h+1) == 0);
tes4(i,g3)=30;
elseif (cod4_1(i,h)==0 && cod4_1(i,h+1) == 1);
tes4(i,g3)=20;
else (cod4_1(i,h)==1 && cod4_1(i,h+1) == 1);
tes4(i,g3)=10;

end
g3=g3+1;
end

for h=1:2:2*num;

if (cod5_1(i,h)==0 && cod5_1(i,h+1) == 0);
tes5(i,g4)=40;
elseif (cod5_1(i,h)==1 && cod5_1(i,h+1) == 0);
tes5(i,g4)=30;
elseif (cod5_1(i,h)==0 && cod5_1(i,h+1) == 1);
tes5(i,g4)=20;
else (cod5_1(i,h)==1 && cod5_1(i,h+1) == 1);
tes5(i,g4)=10;

end
g4=g4+1;
end
end
% % % % % % % % % % % % % % % % %
% n=0;
% for nn=1:KK;
for i=1:length(tes1(:,1)) % 72

for k=1 : num % 16

if ( tes1(i,k)==40)
metr1_1(i,k)=0;
elseif ( tes1(i ,k)==30)
metr1_1(i,k)=.01;
elseif ( tes1(i,k)==20)
metr1_1(i,k)=.01;
elseif ( tes1(i,k)==10)
metr1_1(i,k)=.1;

end

if ( tes2(i,k)==40)
metr1_2(i,k)=0;

```

```

elseif ( tes2(i,k)==30)
    metr1_2(i,k)=.01;
elseif ( tes2(i,k)==20)
    metr1_2(i,k)=.01;
elseif ( tes2(i,k)==10)
    metr1_2(i,k)=.1;

end

if (tes3(i,k)==40)
    metr1_3(i,k)=0;
elseif ( tes3(i,k)==30)
    metr1_3(i,k)=.01;
elseif ( tes3(i,k)==20)
    metr1_3(i,k)=.01;
elseif ( tes3(i,k)==10)
    metr1_3(i,k)=.1;

end

if (tes4(i,k)==40)
    metr1_4(i,k)=0;
elseif ( tes4(i,k)==30)
    metr1_4(i,k)=.01;
elseif ( tes4(i,k)==20)
    metr1_4(i,k)=.01;
elseif ( tes4(i,k)==10)
    metr1_4(i,k)=.1;

end

if (tes5(i,k)==40)
    metr1_5(i,k)=0;
elseif ( tes5(i,k)==30)
    metr1_5(i,k)=.01;
elseif ( tes5(i,k)==20)
    metr1_5(i,k)=.01;
elseif ( tes5(i,k)==10)
    metr1_5(i,k)=.1;

end
end

end
% metr1_1(1+n:window+n,:)=metr1_1(:,:);
% metr1_2(1+n:window+n,:)=metr1_m(:,:);
% metr1_3(1+n:window+n,:)=metr1_h(:,:);
% n=n+window;
% end
% % figure (1),plot(metr1_1(1:21,1));figure (2),plot(metr1_1(22:42,1));figure (3),plot(metr1_1(43:63,1));
% % figure (4),plot(metr1_1(64:84,1));
% % figure (5),plot(metr1_1(85:105,1:16));
% % %
% % % % % % % % % % % % % % % % Path Metric M
% % % % % % % % % % % % % % % % FOR First Prancipal Components (PC1)
for i=1:length(metr1_1(:,1)) % 72

MER1=0;MER2=0;MER3=0;MER4=0;MER5=0;
    for k=1:num;
        ME1(i,k)= metr1_1(i,k)*10^(-k);
        ME2(i,k)= metr1_2(i,k)*10^(-k);
        ME3(i,k)= metr1_3(i,k)*10^(-k);
        ME4(i,k)= metr1_4(i,k)*10^(-k);
        ME5(i,k)= metr1_5(i,k)*10^(-k);

    end
    d1(i)=sum(ME1(i,:));

```

```

d2(i)=sum(ME2(i,:));
d3(i)=sum(ME3(i,:));
d4(i)=sum(ME4(i,:));
d5(i)=sum(ME5(i,:));

end

% figure
%(1);subplot(511);plot(d1);subplot(512);plot(d2);subplot(513);plot(d3);subplot(514);plot(d4);subplot(515);plot(d5);
%figure (1);plot(d1);hold on;plot(d2);hold on;plot(d3);hold on;plot(d4);hold on;plot(d5);
j1=0;j2=0;j3=0;j4=0;j5=0;j6=0;j7=0;j8=0;j9=0;j10=0;j11=0;
j12=0;j13=0;j14=0;j15=0;j16=0;j17=0;j18=0;j19=0;j20=0;j21=0;
for i=1:window
if (d1(2*window+i) < d1(i)&& d1(2*window+i) < d1(i+window)) %TPc=j3;
j3=j3+1;          % TPC(j3)=i;
end
if ( d3(window+i) < d3(i) && d3(window+i) < d3(2*window+i)) %TPb= j2;
j2=j2+1;
%   TPB(j2)=i;
end
if(d5(i) < d5(window+i) && d5(i) < d5(2*window+i)) %TPa= j1;
j1=j1+1;
%   TPA(j1)=i;
end

end

for i=1:window
if (d1(2*window+i) > d1(i+window)) %Ecb=j14;
j14=j14+1;
end

end

for i=1:num
if(d2(2*window+i+8)<d2(i+8)&& d2(2*window+i+8)<d2(window+i+8))% TPC=j6;
j6=j6+1;
end

if(d4(i)<d4(window+i)&& d4(i)<d4(2*window+i)) % TPa=j4;
j4=j4+1;
end

end

for i=1:num/2
if(d2(window+i)<d2(i) && d2(window+i)<d2(2*window+i)) % TPb=j5;
j5=j5+1;
end
if(d4(window+i+16)<d4(i+16) && d4(window+i+16)<d4(2*window+i+16))
% TPb=j7;
j7=j7+1;
end
end
tpc=j3+j6;
tpb=j2+j5+j7;
tpa=j1+j4;
for i=1:num
if(d4(window+i) < d4(i)) %Eab= j8;
j8=j8+1;
end
if (d2(window+i+8)< d2(2*window+i+8)) % Ecb=j9;
j9=j9+1;
end
end
eab=j8;
ecb=j9+j14;

for i=1:num/2

```

```

    if (d4(i+num) < d4(window+i+num))    % Eba= j10;
        j10=j10+1;
    end
    if(d2(2*window+i) < d2(window+i))    % Ebc= j11;
        j11=j11+1;
    end
end
eba=j10;
ebc=j11;

for i=1:num
    if (d4(i) > d4(2*window+i))    % Eac= j12;
        j12=j12+1;
    end
    if(d2(2*window+i+8) > d2(i+8))    % Eca= j13;
        j13=j13+1;
    end
end
eac=j12;
eca=j13;
% j4 ; %Eba= j5;% Ebc= j6; % Ecb= j9; % Eca= j8;
% j7;
% % % tpa=35;tpb=32;tpc=23;eab=0;eac=0;eba=3;ebc=0;eca=0;ecb=12;
% tpa=j1;eab= j2;eac= j3; eba= j5; tpb= j4;ebc= j6;eca= j8;ecb= j9;tpc= j7;
% MERTRICF=[tpa eab eac; eba tpb ebc; eca ecb tpc];
% % tpa=tpa-j2-j3; tpb=tpb-j5-j6; tpc=tpc-j8-j9;
%
accuracyF=(tpa+tpb+tpc)/(tpa+tpb+tpc+eab+eac+eba+ebc+eca+ecb);
precisiFA=tpa/(tpa+eba+eca);
precisiFB=tpb/(tpb+eab+ecb);
precisiFC=tpc/(tpc+eac+ebc);
Av_pre=(precisiFA+precisiFB+precisiFC)/3;
sensitiviFA=tpa/(tpa+eab+eac);
sensitiviFB=tpb/(tpb+eba+ebc);
sensitiviFC=tpc/(tpc+eca+ecb);
Av_sen=(sensitiviFA+sensitiviFB+sensitiviFC)/3;
specificiFA=(tpb+ebc+ecb+tpc)/(tpb+ebc+ecb+tpc+eba+eca);
specificiFB=(tpa+eac+eca+tpc)/(tpa+eac+eca+tpc+eab+ecb);
specificiFC=(tpa+eab+eba+tpb)/(tpa+eab+eba+tpb+eac+ebc);
Av_speci=(specificiFA+specificiFB+specificiFC)/3;
F_mea=2*(Av_sen*Av_pre/(Av_sen+Av_pre));
%
%
% precisiFA;
% precisiFB ;
% precisiFC ;
% sensitiviFA ;
% sensitiviFB ;
% sensitiviFC ;
% specificiFA ;
% specificiFB ;
% specificiFC ;
Av_sen
Av_speci
Av_pre
accuracyF
F_mea
Z=[tpa eab eac; eba tpb ebc; eca ecb tpc]
% % %
% % %
% % % % % Errs19(i,:) = bitxor(codes19(1,:), codes19(i,:));
% % %
% % %
% % %
% % %
% % %

```

PUBLICATIONS

Elkurdi, A., Caliskanelli, I., & Nefti-Meziani, S. (2018, March). Using Amplitude Modulation for Extracting Gait Features. In *ICT4AWE* (pp. 161-168).

Elkurdi, A., Caliskanelli, I., & Nefti-Meziani, S. (2018, August). Amplitude Modulation and Convolutional Encoder Techniques for Gait Speed Classification. In 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR) (pp. 544-549). IEEE.

Elkurdi, A., Soufian, M., & Nefti-Meziani, S. (2018). Gait speeds classifications by supervised modulation-based machine-learning using Kinect camera. *Med Res Innov*, 2018. Volume 2(4): 1-6.

Elkurdi, A., & Nefti-Meziani, S. (2018). Several Modulation Techniques for Gait Features Extraction using Kinect Camera. In *DeSE2018 V7-1-1*.

Elkurdi, A., Soufian, M., & Nefti-Meziani, S. (2018). Amplitude Spectrum for Gait Features Extraction and Classification. In *IRES-TheIIER Conference and Member's Meeting* (in press).