CONCUSSION BALANCE AND POSTURAL STABILITY ASSESSMENT SYSTEM USING KINETIC DATA ANALYSIS

by

Lingyi Liu

B.Sc., Xi’an Jiaotong University, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES (Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2014

© Lingyi Liu, 2014
Abstract

In current scientific literature, there are numerous approaches that clinicians can use to assess the static postural stability of patients. Among them, the Balance Error Scoring System is a notable method with merits such as cost-effectiveness and portability. Traditional measurement of errors made by patients in BESS test experiment relies on the manual inspection of sophisticated clinicians to the whole experiment process. A new avenue of detecting errors with wireless sensor network and signal processing technique can eliminate the instability from subjective evaluation in traditional method. This thesis present a reliable analytical system that can provide accurate evaluation on errors in BESS test of patient with concussion to assist clinicians to investigate their standing postural stability.

In this research, the kinetic signal data is collected by wearable WSN equipment consisting of seven sensors embedded with accelerometer and gyroscope fixed on body of patients while they are completing BESS experiment. We use experimental data of 30 subjects to train back-propagation neural network and test the performance of neural network with testing data set. In this procedure, statistical technique such as principal component analysis and independent component analysis are applied in the step of signal pre-processing. Meanwhile, feature extraction is an alternative pre-processing technique for kinetic signal and the feature data serves as input data to train the neural network. With regard to target training data, the standard error information are acquired from the analysis of a group of researchers on video of the conducted experiment and we present them with Gaussian curve signal indicating the possibility of the error event. By testing the neural network, the technique of feature extraction in combination with
back-propagation neural network is confirmed to account for the most optimal assessment of the postural error in BESS test.

Furthermore, we can confirm the type of each detected error from six possible types of postural errors with neural network classification technique. Each type of error is corresponding to a certain unstable posture according to “BESS Protocol”.

Ultimately, the presented error detecting system is convinced to supply reliable evaluation of the static postural stability of patients with concussion problem.
Preface

This thesis is an original intellectual product of the author Lingyi Liu. None of the text of the dissertation is taken directly from previously published or collaborative articles. The experiment for acquiring the primary BESS test data, presented in Chapter 2, was conducted in the Sensorimotor Physiology Lab at the University of British Columbia (UBC) using method approved by UBC’s Clinical Research Ethics Board (CREB; H11-02306) and led by Harrison Brown. I have helped with resolving the problem of charging the modules and with the synchronization of sensors. The test of Shimmer WSN system in 2.2.2 is conducted by myself. The raw data in database was analyzed and was prepared for inclusion in this thesis by myself. The two error-detection approaches in chapter 3 and chapter 4 together with the classification of errors and the evaluation of each sensor’s contribution are designed and evaluated by myself independently. Edmond Cretu and Jean-Sébastien Blouin were supervisory authors involved in concept formation, analysis advisory and manuscript editing.
Table of Contents

Abstract .................................................................................................................................................. ii

Preface ................................................................................................................................................ iv

Table of Contents .................................................................................................................................. v

List of Tables ......................................................................................................................................... ix

List of Figures ....................................................................................................................................... x

Acknowledgements .............................................................................................................................. xiv

Dedication ................................................................................................................................................ xv

Chapter 1: Introduction .......................................................................................................................... 1
  1.1 Motivation and Challenges ........................................................................................................... 1
  1.2 Background ................................................................................................................................ 2
    1.2.1 Balance Error Scoring System ............................................................................................. 2
    1.2.2 Shimmer Sensor Equipment ............................................................................................... 2
    1.2.3 Pattern Recognition and Classification .............................................................................. 5
  1.3 Research Problem and Approach ................................................................................................. 6
    1.3.1 Detection of the Error and Classification of the Error ....................................................... 6
    1.3.2 Error Detection System Design ........................................................................................... 8
    1.3.3 Classification System Design .............................................................................................. 13
  1.4 Discussion of Related Work .......................................................................................................... 13
  1.5 Thesis Organization ....................................................................................................................... 19

Chapter 2: Data Collection ..................................................................................................................... 21
  2.1 Introduction .................................................................................................................................. 21
  2.2 Experimental Equipment ............................................................................................................. 21
Chapter 2: System Implementation

2.2.1 Experimental System Architecture .......................................................... 22
2.2.2 Test of the System with the Heart Rate Measurement.................................. 23

2.3 BESS Test Experiment ..................................................................................... 25
2.3.1 Subjects and Protocol................................................................................... 25
2.3.2 Wearable Equipment...................................................................................... 26
2.3.3 Error Activities in Experiment...................................................................... 26
2.3.4 Data Collection ............................................................................................. 27
2.3.5 Evaluation of Raters...................................................................................... 28

2.4 Summary .......................................................................................................... 30

Chapter 3: Error Detection .................................................................................... 32

3.1 Introduction....................................................................................................... 32
3.2 Problems and Proposed Approach.................................................................... 32

3.3 Pre-processing of the Original Signal .............................................................. 33
3.3.1 Representation of the Collected Kinetic Data.............................................. 33
3.3.2 Centralization............................................................................................... 34
3.3.3 Extraction of Valid Data .............................................................................. 35
3.3.4 Signal Compression...................................................................................... 37

3.4 Principal Component Analysis and Independent Component Analysis for Original Signal ............................................................................................................. 38
3.4.1 Application of PCA ..................................................................................... 39
3.4.2 Dimensionality Reduction .......................................................................... 40
3.4.3 Data Distribution after Implementing PCA .................................................. 42
3.4.4 Definition of ICA ....................................................................................... 42
Chapter 3: Error Signal Generation and Classification

3.4.5 Application of ICA ................................................................. 44

3.5 Generating Standard Error Signal .............................................. 46
  3.5.1 Extract Valid Standard Error Signal Data .................................. 47
  3.5.2 Digital Pulse Signal Representation ......................................... 47
  3.5.3 Comparison with Sensor Signal and Observed Deviations .......... 49
  3.5.4 Digital Window Signal Representation ..................................... 59
  3.5.5 Gaussian Curve Signal Representation .................................... 61
  3.5.6 Combination of Error Record from Different Raters .................. 63
  3.5.7 Summary .............................................................................. 68

3.6 Neural Network Classifier .......................................................... 68
  3.6.1 Neural Network Architecture .................................................. 69
  3.6.2 Testing Result and Accuracy Analysis ...................................... 70
  3.6.3 Summary .............................................................................. 76

3.7 Summary .................................................................................... 76

Chapter 4: Error Detection with Feature Extraction ......................... 78

4.1 Introduction .................................................................................. 78

4.2 Problems and Proposed Approach ............................................. 78

4.3 Feature Extraction ....................................................................... 79
  4.3.1 Calculation of Features ......................................................... 80
  4.3.2 Sliding Window Length and Hop Length ................................. 81
  4.3.3 Feature 1 - Mean Value ....................................................... 82
  4.3.4 Feature 2 - Standard Deviation ............................................. 83
  4.3.5 Feature 3 - Crest factor ....................................................... 83
Chapter 4: Image Processing................................. 21
4.2 Pre-Processing............................................. 22
4.2.1 Image Segmentation................................. 22
4.2.2 Image Segmentation.................................. 23
4.3 Neural Network Classifier with Feature Extraction................................. 24
4.3.1 Neural Network Architecture.......................... 24
4.3.2 Testing Result and Accuracy Analysis................. 25
4.4 Summary.................................................. 25
Chapter 5: Classification of Errors............................. 26
5.1 Introduction.............................................. 26
5.2 Problems and Proposed Approach.......................... 26
5.2.1 Application of Neural Network........................ 26
5.2.2 Target Output Signal.................................. 27
5.2.3 Input Signal........................................... 27
5.3 Classification and Analysis................................. 28
5.3.1 Neural Network Architecture.......................... 28
5.3.2 Testing Result and Accuracy Analysis................. 29
5.4 Each Sensor’s Contribution................................. 30
5.4.1 Evaluation Approach................................... 30
5.4.2 Validation............................................... 30
5.5 Summary.................................................. 30
Chapter 6: Conclusion and Future Work.......................... 31
6.1 Conclusion................................................ 31
6.2 Future Work............................................... 31
Bibliography.................................................... 33
Appendices...................................................... 34
List of Tables

Table 2.1 Structure of all kinetic data collected through the BESS test experiment .................... 28
Table 3.1 Relationship between the selected principal signal components and the percentage of energy they contains ........................................................................................................ 41
Table 3.2 Training information of $NN_1, NN_2$ and $NN_3$ ................................................................ 69
Table 4.1 Training information about neural networks $NN_4$ and $NN_5$ .................................................. 85
Table 5.1 Detailed information about the neural network $NN_6$ and $NN_7$ .............................................. 92
Table 5.2 Contribution of each sensor ........................................................................................................... 99
List of Figures

Figure 1.1 Conceptual structure of the error detecting system .......................................................... 10
Figure 2.1 Architecture of the experimental system ............................................................................. 23
Figure 2.2 Average value of the acceleration data on three-axis collected by one shimmer unit placed on left side of the subject’s chest ......................................................................................... 24
Figure 2.3 Position of each sensor deployed on human body .................................................................. 26
Figure 2.4 Evaluations from four raters of trial 2 for subject 9 .............................................................. 29
Figure 2.5 Evaluations from four raters of trial 6 for subject 13 ............................................................ 30
Figure 2.6 Evaluations from four raters of trial 6 for subject 5 ............................................................ 30
Figure 3.1 Structure of the sensor data matrix ....................................................................................... 34
Figure 3.2 Structure of the valid sensor data matrix ............................................................................. 36
Figure 3.3 (a). Data distribution of signal P(31)-to-P(33), the horizontal axis stands for P(31) and the vertical axis stands for P(33); (b). Data distribution of signal P(39)-to-P(42), the horizontal axis stands for P(39) and the vertical axis stands for P(42) ......................................................................................... 42
Figure 3.4 Process of independent component analysis ................................................................. 44
Figure 3.5 Evaluation of raters and experimental sensor signal for subject 01 trial 2 .................. 48
Figure 3.6 Observed deviation between raters’ evaluation signal and forehead sensor signal from subject 23 trial 2 ............................................................................................................ 50
Figure 3.7 Observed deviation reflects in signals collected from seven different sensors in Subject 05 Trial 3: (a). forehead sensor; (b). chest sensor; (c). waist sensor; (d). left-wrist sensor; (e). right-wrist sensor; (f). left-shin sensor; (g). right-shin sensor ......................................................................................................................... 54
Figure 3.8 Raters’ evaluation signal and forehead sensor signal after alignment from subject 23 trial 2

Figure 3.9 Raters’ evaluation signal and signal collected from seven sensors after alignment from subject 05 trial 3

Figure 3.10 Digital window signal representation for the evaluation of the rater and corresponding sensor signal, subject 01 trial 2.

Figure 3.11 Gaussian representation for the target output signal for subject 03 trial 6

Figure 3.12 Bell curve for gaussian function when (a).\(\sigma=2\); (b).\(\sigma=5\); (c).\(\sigma=10\)

Figure 3.13 Format of the error signal vector for the evaluation of one rater

Figure 3.14 Gaussian curve summation representing the target output signal for subject 9 Trial 2

Figure 3.15 Gaussian curve summation representing the target output signal for subject no.9 trial no.3

Figure 3.16 Gaussian curve maximum value representing the target output signal for subject 9 trial 2

Figure 3.17 Gaussian curve maximum value representing the target output signal for subject 9 trial 3

Figure 3.18 Relationship between elements of \(G_3, E_1, E_2, E_3\) and \(E_4\)

Figure 3.19 Architecture of \(NN_1\) and \(NN_2\)

Figure 3.20 Architecture of \(NN_3\)

Figure 3.21 Actual output signal and target output signal of \(NN_1\) (neural network with \(G_1\) as the target output)
Figure 3.22 Four possible relationship between actual output signal and target output signal of testing result of every neural network.................................................................73

Figure 3.23 Actual output signal and target output signal of NN_2 (neural network with G_2 as the target output)..........................................................................................................................74

Figure 3.24 Actual output signal and target output signal of testing simulation for NN_3: (a). error information from Rater 1; (b). error information from Rater 2; (c). error information from Rater 3; (d). error information from Rater 4..........................................................................................................................76

Figure 4.1 Process of building error-detecting system with neural network and feature extraction ........................................................................................................................................79

Figure 4.2 Relationship between raw data matrix and three feature matrixes and the process of feature extraction based on sliding window..................................................................................................................82

Figure 4.3 Architecture of neural networks NN_4 and NN_5..................................................................................................................86

Figure 4.4 Actual output signal and target output signal of testing simulation for NN_4............87

Figure 4.5 Actual output signal and target output signal of testing simulation for NN_5...........88

Figure 5.1 Format of the output matrix TC..................................................................................91

Figure 5.2 Architecture of neural network NN_6 and NN_7 ......................................................93

Figure 5.3 The actual output signal and target output signal of testing simulation for NN_6 for each type of error: (a). “moving the hands off of the iliac crests”; (b). “opening the eyes”; (c). “step stumble or fall”; (d). “abduction or flexion of the hip beyond 30°”; (e). “lifting the forefoot or heel off of the testing surface”; (f). “remaining out of the proper testing position for greater than 5 seconds”........................................................................................................................................95

Figure 5.4 Actual output signal and target output signal of testing simulation for NN_7 for each type of error: (a). error “moving the hands off of the iliac crests”; (b). error “opening the eyes”;
(c). error “step stumble or fall”; (d). error “abduction or flexion of the hip beyond 30°”; (e). error “lifting the forefoot or heel off of the testing surface”; (f). error “remaining out of the proper testing position for greater than 5 seconds” ................................................................. 96

Figure 5.5 Accuracy percentage of six types of errors in $NN_6$ and $NN_7$ .................................................. 97

Figure 5.6 Actual output and the target output signal of the testing simulation for $NN_8$.............. 101
Acknowledgements

I would like to express the deepest appreciation to the help and supervision of Dr. Edmond Cretu for guiding me all through my program. His academic guidance and support have encouraged me during my study in this school. Without his inspiration, I cannot finish this work. I would also like to acknowledge and thank for the assistance from Dr. Jean-Sébastien Blouin and Harrison Brown in data acquisition experiment from which I have benefited a lot. Without your valuable research work, this project is impossible.

I would also like to offer my sincere gratitude to my lab mate Amy Wei You and my friend Yize Xu for all the discussions and helpful suggestions about my work.

Outside of a formal school setting, I would like to thank my friends Xun zhou, Meng Wang, and Aoyu Yan for genuine support in my hard times. Their help when I was injured and have to walk relying on crutches helped me overcome the darkness.

Ultimately, I owe my special thanks to my parents who have been supporting me for my studies and in life, and I hope I can reward your hard work with my own in the future.
Dedication

Dedicate to my friends and family
Chapter 1: Introduction

1.1 Motivation and Challenges

The field of intelligent systems has been developing extremely fast, and its application in the medical field is exponentially growing in recent years. Wearable sensor systems can definitely not be ignored when we talk about the estimation of postural balance for medical purposes, or the so-called Balance Error Scoring System (BESS) test, about which we will discuss in detail in the following chapters. Despite that this classic approach is cost-effective and convenient to be conducted, it requires a well-trained specialist and a large amount of labor cost. Despite the training of the physiotherapist or medical rater, the quantitative error assessment in static postural stability is still subjective and shows variability from one examiner to another. Therefore, we propose a new objective approach to complement the BESS test, based on collected sensor data from modules attached to the patient. Our aim is to build an auto-diagnosis system that can support the rater to make back-to-play decisions for the patients who get injured and suffer concussion effects. Important questions, among the challenges, are: How are we going to design this system? What will be the most important functioning components of the system? What kind of information about the patients do we need to collect and process, in order to help the rater to make correct decisions? What kind of hardware equipment do we need to collect information? How can we extract the information we really need and through what kind of processing can we make use of these information? How are we going to make the right decision from the information we collect? Is there any possibility to simplify the hardware equipment we use in this experiment and how can we do that? All these problems will be discussed in this thesis, during the development of our system.
1.2 Background

1.2.1 Balance Error Scoring System

The Balance Error Scoring System is called “BESS” for short. It is a standard which is used to quantitatively evaluate how the static postural stability is affected when the patient got mild head injuries. When a sophisticated and expensive postural stability assessment tool is not available, BESS is a good option since it is a cost-effective and portable method. By using this clinic balance evaluation test procedure, some useful information can be collected to help the clinicians decide if a sport player, for instance, may return playing and training following a mild head injury. It only takes about 10 minutes to implement the BESS procedure, in almost any environment. A set of BESS tests for a subject consists of six distinct types of trials. The whole BESS test protocol is described in Appendix A.

1.2.2 Shimmer Sensor Equipment

The word “Shimmer” stands for “Sensing Health with Intelligence Modularity, Mobility and Experimental Reusability” (Shimmer, Realtime Technologies Ltd., Dublin, Ireland) [1], which is a low-consumption wireless sensor platform, consisting of wearable wireless sensing modules that can record and transmit various kinds of data with certain RF protocols (IEEE 802.15.4 or Bluetooth). The product line has been focusing on wearable sensor technology and solutions for research and education applications, as well as for commercial ones.
1.2.2.1 Application

Shimmer platform is designed to approach the wireless sensor network problems in both research and industrial areas. In research and education, the Shimmer wireless sensing platform provides strong and sufficient sensing capabilities useful for many body sensor network projects. The company provides a range of both equipment and software interface layers, enabling the integration of the system in larger projects.

1.2.2.2 Construction

The platform integrates hardware equipment with supportive software applications to implement various functions of the platform.

The hardware part of the platform mainly consists of several Shimmer units – each of them consists of a central board with a core set of functions (e.g. wireless communication) and sensors, while providing connectors for expanded functionality. Through expansion, we can add various specific sensors into a given Shimmer unit. For example, there are expansion modules for ECG, EMG, GSR and other specific measurements. In the project for this thesis, I use the module with kinematic sensors expansion. This kind of Shimmer unit includes 3D accelerometers, gyroscopes and magnetometers. We have used only the acceleration and angular data for the algorithms of this project.

The software part of the platform provides a variety of approaches for the user to control the whole system. On the most basic level, the Shimmer firmware provides users with a convenient environment to use the platform. On a higher level, Shimmer has various supporting software
development tools and also some instruments drivers, such as “LabVIEW Instrument Driver Library and LabVIEW Development Library,” “Shimmer MATLAB Instrument Driver” and “Shimmer Android Instrument Driver”. In this project, we use “LabVIEW Instrument Driver Library and LabVIEW Development Library” to build a LabVIEW application to take in charge of the data collection and storage. In addition, Shimmer also provides additional software to realize the function of the platform. For example, there are applications supporting the display, storage and Bluetooth streaming of the data, and also applications helping to calibrate the Shimmer unit. In this project, we use some of the enabling software to ensure that the system works well. There are as well additional software features, enabling the user to embed more specific tasks into the developed applications.

1.2.2.3 Features and Advantages

Firstly, Shimmer has a configurable and flexible architecture. With different sensor modules, the platform can be programmed to meet various applications, with different requirements for data acquisition and processing. Secondly, Shimmer reduces the application development time and cost with a reliable and robust technology, that can be rapidly integrated into the overall system solution. Unlike propriety and closed sensor platform, Shimmer provides a complete control over the raw data captured, for interpretations and analyses specific to the application. A Shimmer unit to be attached to the body has a small size and convenient shape, causing minimal inconveniences as wearable module. As for one of the most challenging problem in Wireless Sensor Networks, the energy consumption, Shimmer is also outstanding by its low power requirements. At last, Shimmer platform provides an open and wide development environment, including instrument driver, enabling software, and advanced features.
1.2.2.4 Provide Solutions

Shimmer platform provides sensing solutions for wearable technology. With the whole set of equipment and the supporting software, user are able to capture a wide range of data, such as kinematic data, ECG, EMG signals, etc. Shimmer units specifications, including sensing abilities and characteristics are presented in [1]. In this project, we have used both the hardware and some of the software tools provided by the developers.

1.2.3 Pattern Recognition and Classification

The main problem to be solved consists in identifying, from the measured data channels, those relevant signal patterns associated with errors flagged by the specialists judging a BESS procedure. Pattern recognition algorithms are useful in this sense, for they can look for a certain level of matching between input values and a database of identified patterns. Classification is an example of pattern recognition. The purpose of classification is to assign each given input to one of the classes in a given set. However, pattern recognition is a more general concept that encompasses other types of output as well, including regression (which assigns an output value to each input), sequence labeling (which classify every element of a sequence of values), and parsing. In this project, we use a classification method to identify the postural balance errors.

In machine learning and statistics, supervised classification provides a solution based on a set of training data containing observations and their category information – after a training stage using the already labeled data, the system is able to classify new inputs in the relevant categories. There are various classification algorithms. Among the mostly used classifiers, one can mention neural networks (multilayer perceptron), k-nearest neighbors, Gaussian mixture model, Gaussian, naive
Bayes, decision trees, Radial Basis Functions (RBF) classifiers, and support vector machines (SVMs). We have used a neural network for the signal processing of the signals collected by the wireless sensor network. Neural network is a classic approach for solving the machine learning problems. The chosen architecture of the neural network is a multi-layer perceptron with three neuron layers. This is a common layer structure in a neural network structure, with an input layer, hidden layer and output layer. These three layers are connected in a chain. More specifically, the input of the successive layer is the weighted sum of the outputs of previous layer, plus a constant bias. Given a known training set containing both input signals and the desired corresponding outputs, we can train the neural network for thousands of times so that it can interconnect the neurons between different layers, with the proper weights. The process of training includes adjusting the weights and the constant bias for each neuron, so that it follows an error minimization tuning algorithm (backpropagation rule); in the end, the output is adjusted so that the difference between the neural network output and the target output is minimized. Eventually, a well-trained neural network will be able to produce an input-output nonlinear mapping that implements the classification problem to be solved, namely to detect and recognize the error patterns in the input signal space.

1.3 Research Problem and Approach

1.3.1 Detection of the Error and Classification of the Error

As we have talked before, the BESS score is a widely used and cost-effective approach to evaluate whether the athlete can be back to play after injured. Despite many advantages over other methods, the Balance Error Scoring System can still be improved.
Traditionally, the BESS test has to be supervised by an experienced rater or researcher who can clearly memorize the six categories of errors and need to recognize them during each balance trial of the subject. It is a task with a high pressure and responsibility, especially when there are many subjects who need to take the test. It is time-consuming and is definitely not effective in terms of labor.

In addition, there is another difficulty in recognizing the errors made by the subject during the test trials. It has to do with the subjectivity in decision-making by the specialist, which add a certain degree of uncertainty and variability to the error evaluation. Since there is not a clear standard to recognize each error, the result can vary even for the same subject under the same experimental condition, let alone the results recorded by different raters. Recognizing errors manually is a choice between two options (have error or not), while classifying errors is a choice between six possible categories (six types of errors). Because the reliability and accuracy of the error detection cannot be ensured, the diagnostic made by raters according to their evaluation of the quantitative errors might be biased by subjectivity, and not necessarily be an accurate and consistent result. The errors evaluation have a great influence on the final decision of whether to allow the athletes to go back to play. In order to get a more precise assessment, we will need to reduce the variability and subjectivity of the BESS tests by an objective analysis of postural balance. The test postures specified by the BESS standard will remain the same, but the evaluation of the specialist (physiotherapist or rater) will be complemented by analyzing signals collected from the subjects with the help of attached wireless body sensor modules. Ideally, the diagnostic results should not be influenced by subjective factors or any environmental
conditions. The tests performed on the same subject, in the same recovery state, should be reliable and give consistent results in several repeated trials.

Considering the aspects above, we propose an approach towards an auto-diagnosis system. This system can collect the postural balance information regarding the subject, during the BESS testing procedures, with the help of a wearable wireless sensor network. The multi-channel collected data contains the raw information correlated to the body position and dynamics. We need to develop a post-processing detecting system, using pattern recognition technology, to identify when the subject makes any error of those six categories. The system should also be able to classify the detected errors into the six categories defined by BESS test standard.

1.3.2 Error Detection System Design

The auto-diagnosis system architecture contains therefore two main parts: the wearable wireless sensor network equipment (the Shimmer equipment) and the detecting and classifying software subsystem. In this thesis, we focus on the post-processing software part, using the collected raw data for the error detection and classification. This part conducts the function of signal processing or data analysis through pattern recognition and machine learning technology. The information collected through the BESS test experiment conducting with the shimmer wireless sensor network equipment can be analyzed and then we come into a conclusion through the analysis.

Therefore, the main research problem in this project can be summarized as the design of an error detection and classification system, based on kinetic body sensor data, able to complement the
evaluation of the specialists. The conceptual structure of the error detecting system is shown in Figure 1.1.
Figure 1.1 Conceptual structure of the error detecting system
Firstly, we need to transfer the data collected from the Shimmer equipment into a format appropriate for future signal analysis and processing. As discussed before, the Shimmer units we have used in this project are expanded with kinetic modules that include 3D accelerometers and gyroscopes. These sensor units are attached with non-intrusive straps to different parts of the body and can record the movement information of the subjects during experiments. The acquired multichannel signal data needs to be aligned along the time axis, and mapped to synchronized time series. The time series representation will also allow the reduction of the dimensionality of the input signal space for the neural network, for instance by feature extractions. The details will be discussed in the following chapters.

Secondly, after acquiring the sensor data and mapping the raw data into properly aligned time series, filtering out noise components, it is necessary to identify the specific signal patterns associated with postural balance errors. In order to identify these patterns correlated to balancing errors, the errors identified by specialists, on the duration of the same tests, need to be also time-aligned with the sensor signals. In the present case, four distinct specialists have diagnosed the same patients, based on the visual recording of their tests. The binary error signals already show the variability mentioned due to the subjectivity of the observers. In order to simplify matters, a unique analog signal is generated from the four binary error signals, using a Gaussian envelope over the signal mixture, as detailed later in the thesis. This error signal is taken as a reference output for the training and validation of the neural networks response. In the next step, several neural networks are trained and the one offering the highest accuracy is chosen.
Thirdly, after transforming the data into a reduced order form (pre-processing step), the next stage consists in developing a decision making system to analyze the input data and provide an error decision; compared to the error identification feedback offered by the medical specialists, based on the visual analysis of the patients, this time the error signal is based on the input signals provided by the sensors attached to the body, so the information is complementary and objective. A machine learning approach, based on a three-layer neural network architecture, has been chosen for this problem. The neural networks use the sensor inputs and the error signal as reference. 80% of the collected signals have been used for training the networks (supervised training, with a reference signal), tuning the interconnection weights through a classic back-propagation algorithm, while the remaining 20% were used for the validation of the results.

After training several neural networks, we have compared their performance, and in the end have selected the one giving the best matching with the error signal generated from the specialists decisions. In the following chapters, we will discuss the details about the comparison result and the analysis of the results.

We have thus obtained an effective error detecting system that can generate, based on the input sensor data, a time error signal that identifies, for every BESS test experimental trial, when the subject has made each individual error. The errors information and the exact time they have been made are a valuable reference for the rater, in helping him to make a wise decision regarding the patient recovery status, and if he/she is ready to return to play and training.
In this error detecting system, one needs to point out that we have two different methods to complement the classic BESS system. Each of them obeys the basic designing procedure already outlined. However, the detailed implementation of some of the steps might be different. We will compare the performances of these two methods and choose in the end the one that provides the best results (best matching with the decisions of the specialists).

1.3.3 Classification System Design

A further added feature to such a system is to not only identify the moment when a postural balance error occurs during the BESS procedure, but to also categorize it. The BESS standard mentions six distinct classes of errors, as described in Appendix 1. The error classifier was implemented as well using a neural network architecture. As the error signals from the raters did not include any information regarding the type of error they have identified at a certain moment in time, a more delicate training procedure was necessary, based on firstly adding the error category in the reference signal (by the examination of the visual information).

1.4 Discussion of Related Work

There have been several research works on the topic of how to detect whether the subject has recovered from cerebral concussion. Cavanaugh has proposed in [2] that approximate entropy (ApEn), a regularity statistic from non-linear dynamics, can reflect the changes in postural control of the athletes with normal postural stability after cerebral concussion. In that paper, the author collected a series of center of pressure (COP) data through the Sensory Organization Test (SOT) [2] under certain conditions, with the Smart Balance Master System [4]. The analysis of the amplitude of COP displacements has shown that the oscillations of the COP (quantified as
ApEn) of the quiet standing athletes after concussion is related to their postural control abilities. The author hypothesized that the return of the ApEn value to the pre-injury level indicated the complete recovery of postural control after injury.

In addition, Kevin M. Guskiewicz has investigated in [5] the postural stability and the neuropsychological deficits in athletes after they experience concussion. He conducted a series of experiments including the measurements of the postural stability of the subjects using SOT and BESS, as well as some neuropsychological tests. The result has revealed that the subjects presented postural stability deficits after injuries, as revealed by both SOT and BESS, especially on the first day after injury. Nevertheless, the neuropsychological measurements did not demonstrate that the loss of consciousness and amnesia are associated with slowed recovery on measures of postural stability or neurocognitive functioning. The research indeed supports the opinion that BESS can reflect the postural stability of the subjects after concussion. However, as for the limitations, the BESS test here is conducted through the traditional approach involving much labor work of the clinicians. The novel approach in this thesis will improve this situation.

Moreover, the wearable wireless sensor network (WSN) system has been widely used before to detect specific human body activities, with many reported research works in this field. Clemens Lombriser and his colleagues described their working on human body activity recognition in [6]. They used the hardware SensorButton [6], a miniaturized wireless sensor platform, to collect the body activity information from the subject. The sensor nodes, including accelerometer, light sensor and microphone, can be wore by the subject, and communicate with the sensor platform. Then the data was segmented using a sliding window and they compute the features of the data
in this window. Finally, they use the *Weka Machine Learning Toolkit* [7] to train the k-Nearest Neighbours [8] classifier. Their research has used a different wearable WSN system and a different recognition algorithm compared to us. However, their idea of a reduced dimensionality space through a feature computation first was an inspiration for our research.

There are some researches working on the human body activity recognition through biomedical signal processing. The EEG [9] and ECG [11] signals are for instance collected from the subject while he is doing pre-designed set of activities using wearable wireless sensor network equipment like Shimmer; through signal processing and analysis researchers were able to recognize what kind of activity the subject was doing. In this thesis we use the 3D accelerometer 3D gyroscope sensors from the Shimmer units to collect acceleration signal and angular velocity signals during the BESS experiments.

Jonghun Baek and his colleagues have proposed in [13] an improved method for probing the user’s activities, by using only one accelerometer. The wearable computer equipment is used especially for monitoring elder people, for instance sending an alarm message to the elder user’s family member when the user falls down and is likely to be injured. Related researches are reported by Randell and Muller [14], Farringdon and colleagues [15], and Schmidt and colleagues [16]. To summarize their work, most of the time they used one-axis or two-axis accelerometers to conduct the experiment, extract the average value, root mean square and other integrated feature values, used afterwards as input signals to a neural network for classification of activities. Jonghun Baek has calculated a histogram of acceleration data collected by a two-axis accelerometer, discovering that the shape of the histogram for certain activities has a certain
distinct pattern. He selected the mean value, standard deviation, skewness, kurtosis and eccentricity as signal features, and then used a multi-layer perceptron classifier to estimate the possible activity of the user. The experimental equipment limits the accuracy of this research. Comparing to the Shimmer sensor network used in our research, only one two-axis accelerometer is not likely to collect enough activity information.

There are several researches working on accelerometer and/or angular rate signal processing for various medical applications. In [17], Przemyslaw investigates the influence of the accelerometer pre-processing on the human activity recognition result. The researcher conducted the experiment using a Shimmer platform to acquire accelerometer data while subjects are doing a list of activities. The feature-based classifier calculated features for overlapping time frames and used them for distinguishing between activities. The features are calculated for both time and the frequency domains. The time-domain features are: mean value, standard deviation, the kurtosis [18] and the crest factor [19]. Our research is inspired by their choice of features. They have used a neural network classifier with back-propagation algorithm for tuning [20], and a K-NN classifier [8]. The pre-processing investigated in their paper includes a study on signal filtering and the number of accelerometers choice. The results indicate that the choice for both the filtering band and for the number of accelerometers has influences on the accuracy of the activity recognition, especially if it is necessary to avoid lower classification accuracies.

In [14], Cliff Randell and Henk Muller have performed a research on context awareness by analyzing accelerometer data. The word context means in their work a certain human body activity to be recognized. They have tried to minimize the number of the sensors in the
experiment, so they have only used one two-axis accelerometer. They extracted four features from only two data source, the X-axes and Y-axes data of the accelerometer, and used a clustering algorithm [21] as the classifier. The time-window length used to calculate the feature is 2 seconds. Nevertheless, in this research, the accuracy cannot be guaranteed because the number of the source of the signals is not enough to provide sufficient information. And the author has claimed that the features discussed are specific to each individual person. In that case, one person’s running activity can provide the same result as another person’s walking activity. The accuracy is obviously not ensured.

In [24], Jong Gwan Lim and his colleagues worked on motion segmentation by pre-processing the raw acceleration data by a median filter and a low-pass filter (FIR 10th, 5 Hz) and selected the absolute value of the acceleration, the absolute value of first derivatives of the acceleration and the absolute value of second derivatives of the acceleration to generate the feature vector with Principal Component Analysis (PCA) [25] and Fisher’s Linear Discriminant Analysis (FLD) [27]. They have chosen to use Multi-Layer Perceptron (MLP) and Radial Basis Function networks as classifiers. In terms of time delay, MLP had the best performance.

The idea of feature extraction based classification for activity recognition has been also widely discussed in the scientific literature. In [28], Tam Huynh and Bernt Schiele have carefully investigated the influence of the choice of the features and the window length over which the feature is calculated. They pointed out that there are some widely used features usually extracted for the acceleration signal processing: mean value [29] [30] [31] [32] [37], entropy [29], standard deviation and variance [30] [31] [34] [37], discrete FFT coefficients [32], peaks in raw data [33],
correlation between axis [29] [37], powers of wavelet coefficients [36] or energy [29] [37]. They used the k-means clustering [38] algorithm to classify the activities, and evaluated each feature’s performance separately to make a comparison. The features with a higher cluster differentiation were shown to also perform better in recognition. Many combinations of features and window length values were investigated – the conclusion of that research work was that the computed features, as well as the choice for the time-window length did have clear effects on the accuracy of the results, but there was not a single feature or a single time-window length that would perform the best for all the activities. From the discussions above, we can conclude that the choice of features and the time-window length are principal aspects to be individualized for our research.

There are also many researchers working on acceleration signal processing for applications other than human body activity recognition. For instance, in [40] Andrew Wixted and his colleagues worked on the estimation of the energy expenditure of the athletes through tri-axial acceleration signal processing. They used the spectrum of the acquired data to estimate the energy, and to compare it with the standard biochemical estimation method. In [41], I.J. Jang and W.B. Park used signal processing of the acceleration data to recognize different gestures on handheld devices. The signal processing part includes low-pass filtering, thresholding and comparing, high-pass filtering, boundary filtering and debouncing. During recognition, the system compared the signal collected with standard signal patterns from a template database for each type of gestures – a specific gesture is recognized based on a template-matching criterion. Moreover, Matt Van Wieringen has developed a real-time signal processing algorithm to process the
acceleration data on wearable patient monitoring devices in [42]. A Wii system was used instead of precise sensor devices, so the accuracy of the results could not be guaranteed.

The most significant related research work is [43] from Harrison Brown; he conducted a series of BESS test experiments using Shimmer equipment to collect postural balance related data. The work of this thesis is a more systematic continuation of his approach. Harrison has also developed a simple empirical error detector for the BESS test. The detector only sets a threshold to the raw measured acceleration data and counts all the moments when the acceleration value is higher than the threshold value as the moments of error. The value of the threshold is tuned empirically, based on an error minimization criterion. Harrison adjusted the threshold while comparing the error results with a reference error signal from the judgments of several medical specialists and set a convenient threshold value. One of the main issues is that the threshold can vary from patient to patient, defeating the purpose of a generic algorithm. The research in this thesis brings an improvement for the error detection system. The proposed method in this thesis is more systematic and reliable, and we can further achieve even an error classification, according to the BESS standard. The motivation for the error classification problem is related to the fact that different sensors might be responsible for the identification of specific categories of errors, and the overall algorithm can be further simplified in the future.

1.5 Thesis Organization

There are six chapters in total in this thesis. In Chapter 1, we have provided an overview of the project. It also presents the motivation and challenges, the background knowledge and related work, and general technical information. We have also introduced here our research problem and
the approach used during this research. In Chapter 2, we provide the details of the BESS test experimental system and the way the experiments were conducted, along with the explanation of the original data collected by the Shimmer units. Chapter 3 presents the general pre-processing of the raw sensor data. The acquired raw data is firstly properly formatted and then compressed using two compression approaches, principal component analysis (PCA) and independent component analysis (ICA). In this chapter, we also discuss a very important aspect of our project, namely the generation of the standard/reference error signal. The error information provided by the raters in our experiment acts as the standard error signal, used as reference output for the training and validation of the designed neural networks. The error signals from the four specialists are only binary, flagging the occurrence of an error at a certain time moment. We will combine the binary signals and generate a global continuous error signal to be used. So in this chapter we discuss several ways of generating this error signal. The last section of Chapter 3 presents an implemented first type of error detecting system, which does not involve any feature extraction. Chapter 4 analyzes a feature-based error detecting system, from design to the evaluation of its performance. In Chapter 5, we discuss about an enhanced system, capable to classify the errors in different categories. We will also evaluate the contribution of each Shimmer sensor node to the output signals, aiming to decrease the number of the sensor nodes used in the BESS test, if possible. Chapter 6 concludes the research results and discusses about potential future directions to improve the system. For instance, we can increase the accuracy of the system by accumulating more and more standard experimental data to train the neural network. We can also develop a better interface between sensor nodes and mobile equipment such as cellphones, in order to increase its accessibility.
Chapter 2: Data Collection

2.1 Introduction

From the presentation in Chapter 1, we know that the suite of BESS tests, simultaneously with collecting sensor data from the seven Shimmer units placed on the patients’ body, is the first step in our research. Postural balance data, simultaneous with video recording of the BESS tests, were collected from a set of seven patients. A group of four raters have analyzed the video recordings and provided their individual error evaluation results, in a manner similar with the standard method of evaluating patients. This process has generated the database that has been used in this work, for building our classifier able to recognize the errors and even their type. The details of the error recognition method will be discussed in the following few chapters. In this chapter, we will discuss the experimental equipment we use in the BESS test experiments and the way the experiments were conducted. A more detailed discussion about the equipment, the system architecture of the Shimmer units wireless sensor network and a test experiment, using this set of equipment to measure the heart rate of the patients, will be included in the discussion. Regarding the tests, we will discuss the status of the subjects and the experiment protocol. How did the equipment support the experiment? Which postures are regarded as error? How do we collect data and what the meaning of the raw data? And at last, how did the raters provide the standard error data for the experiment.

2.2 Experimental Equipment

In our research, we used a wearable sensor network consisting of seven Shimmer units to collect data from the subjects. As we discussed in the previous chapter, each Shimmer unit contains data
transmitting modes allowing the user to transmit acquired data to a controlling computer, which acts as a master controller of the whole system through [44] wireless transmitting protocols (Bluetooth or IEEE 802.11). In our research, we have chosen the Bluetooth protocol for transmitting data between the Shimmer modules and the computer or Android tablet. The master controller is in charge with the starting and ending of the data collection activity, as well as the selection of the activated sensors in each Shimmer sensor node unit. As presented in Chapter 1, each Shimmer module contains several kinds of sensors. We only need some of these sensors active instead of all of them, depending on the measurement context. So only the sensors we choose will be activated. A supporting software environment, running on the master controller, provides a user-friendly interface for the users to control the whole process of data acquisition.

2.2.1 Experimental System Architecture

Figure 2.1 illustrates the architecture of the experimental system, which consists of two main components, WSN and master controller. The forehead sensor unit is the master node of the seven-sensor WSN. All other sensors in the network transmit data to the master node and receive command signals from it. The forehead sensor is responsible for the communication with controller, receiving commands from controller then sending it to all other sensors and transmitting data back to controller. The master controller provides an interface to the user to control the operation of the whole system. The controller runs a LabVIEW application with a Shimmer interface library, which allows user to send controlling commands to WSN, such as starting and stopping data streaming, as well as activating individual sensors. The controller also takes charge of storing the data and post-analysis of the data, in order to get the desired processing.
2.2.2 Test of the System with the Heart Rate Measurement

After we build the whole experimental system, we tested the performance of the system on a separate case study, namely the measurement of the human heart rate using the inertial sensors present in the Shimmer units (mechanocardiography). In this test, we used only three Shimmer units to collect the heart beat rate of the subject.

The three Shimmer units are fixed on the left chest of the subject with medical adhesive bandage. The accelerometer mode and the gyroscope mode are activated to measure the movement happening around the left chest, which can probably represent the movement of the heartbeat.
The periodic rhythms in the recorded signals are correlated with the heartbeats, so this case study has helped in testing the proper configuration and use of a Shimmer network using a lower number of sensing nodes. The acceleration and angular rate data acquired during the experiment make possible to extract the heart beat signals.

Three subjects were involved in this experiment and each of them took three trials for three different postures. For each subject, we used the system to measure his heart rate for three times, while the subject was standing stably, sitting down and lying down on the pad. The signal in Figure 2.2 is the average value of the acceleration data on three-axis collected by one of the three sensors. Figure 2.2 shows that the rhythmical fluctuation reflects the heartbeat of the subject. It can be concluded that the Shimmer system can sense kinetic movement within a narrow range and the system works well enough to get the data we need.

![Figure 2.2 Average value of the acceleration data on three-axis collected by one shimmer unit placed on left side of the subject’s chest](image)
2.3 BESS Test Experiment

The aim of the experiment is to collect the movement data from the subject. A group of 30 subjects were selected and each of them was asked to wear the Shimmer sensor nodes. Then they performed seven trials of experiments according to the BESS test protocol, while the entire procedure was being video recorded. Signals collected from the seven sensor nodes were transmitted to the master computer and saved for further post-processing. Four raters have reviewed the videos of the tests and they marked down every moment when the subject made an error, according to the BESS test protocol. The idea of shooting videos ensured that there is no changeable factor from the environment so that the four raters can review exactly the same experiment trials. The error record provided by the raters is regarded as the standard error information to conduct the error detection. All the experimental data is provided by Harrison Brown who conducted set of experiments in Physiology Lab supervised by Professor Jean-Sebastien Blouin in the University of British Columbia.

2.3.1 Subjects and Protocol

The aim of the system is to help the rater diagnose whether the athletes have recovered from the concussion they have suffered and if they are able to get back to a normal training and playing scheduling after their sport injury. So the subjects we choose for this experiment should have similar physical condition as the athletes. The subjects are 15 females and 15 males, healthy students and athletes from the school sports team, aging from 18 to 29 (25.4 ± 4.2).
2.3.2 Wearable Equipment

The wearable body sensor network consists of seven sensor nodes (Shimmer units) fixed with straps in seven distinct positions on the subjects’ body. As shown in Figure 2.3, the sensor nodes are fixed on the forehead, the chest, the waist, the left wrist, the right wrist, the left shin and the right shin.

![Figure 2.3 Position of each sensor deployed on human body](image)

2.3.3 Error Activities in Experiment

According to the BESS test protocol, there are altogether six categories of activities regarded as errors during the static balance tests. They are: moving the hands off of the iliac crests, opening the eyes, step stumble or fall, abduction or flexion of the hip beyond 30°, lifting the forefoot or heel off of the testing surface, remaining out of the proper testing position for greater than 5 seconds.
Every time when the subject does any of the activities listed above while he was assumed to maintain his static balance, the moment this activity happens will be labeled as an error. Sometimes several errors happen at the same moment - for example, the subject may stumble or fall, together with moving his hands of his iliac crests. In that case, the raters recording the standard error information will just count that event as one single error.

2.3.4 Data Collection

In this experiment, we used seven sensor node units to collect data - each sensor node unit has two activated sensors, a 3D accelerometer and a 3D gyroscope. So each of them provided three scalar signal channels. Altogether, we have 7 sensor nodes (Shimmer units), each having 2 active sensors (1 accelerometer and 1 gyroscope in each shimmer unit), multiplied by 3 data channels per active sensor (3-axis acceleration signal: $A_x$, $A_y$ and $A_z$ or 3-axis angular velocity signal: $\omega_x$, $\omega_y$ and $\omega_z$), namely 42 channels of acquired signals for each trial.

There are 30 subjects involved in this series of experiments, with each subject taking 7 trials of experiments. For each subject, the first trial is considered as calibration trial, used for calibrating and verifying the equipment. The other six trials follow the standard BESS test protocol. Each trial lasts for 20 seconds, during which the subject is required to perform certain static postures and to try to keep a stable status. The sample rate is 102.4 Hz, so there are 2048 sampling points in each trial. Table 2.1 indicates the structure of all the data collected through the BESS test experiment.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Subject 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Subject 2–29</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Subject 30</td>
</tr>
<tr>
<td>Trials (7/Subject)</td>
<td>Trials</td>
<td>Trials</td>
<td></td>
</tr>
<tr>
<td>20 Seconds Each</td>
<td>......</td>
<td>......</td>
<td></td>
</tr>
<tr>
<td>T0 T1 T2 T3 T4 T5 T6</td>
<td>Trials</td>
<td>T0 T1 T2 T3 T4 T5 T6</td>
<td></td>
</tr>
<tr>
<td>1 Sensor 1</td>
<td>$A_x$</td>
<td>Signal</td>
<td>$A_x$</td>
</tr>
<tr>
<td></td>
<td>$A_y$</td>
<td>......</td>
<td>$A_y$</td>
</tr>
<tr>
<td></td>
<td>$A_z$</td>
<td>......</td>
<td>$A_z$</td>
</tr>
<tr>
<td>Sensor 2</td>
<td>$\omega_x$</td>
<td>Signal</td>
<td>$\omega_x$</td>
</tr>
<tr>
<td></td>
<td>$\omega_y$</td>
<td>......</td>
<td>$\omega_y$</td>
</tr>
<tr>
<td></td>
<td>$\omega_z$</td>
<td>......</td>
<td>$\omega_z$</td>
</tr>
<tr>
<td>2~6 Sensor</td>
<td>Signal Channels</td>
<td>Signal</td>
<td>Signal .......</td>
</tr>
<tr>
<td></td>
<td>......</td>
<td>......</td>
<td></td>
</tr>
<tr>
<td>7 Sensor 1</td>
<td>$A_x$</td>
<td>Signal</td>
<td>$A_x$</td>
</tr>
<tr>
<td></td>
<td>$A_y$</td>
<td>......</td>
<td>$A_y$</td>
</tr>
<tr>
<td></td>
<td>$A_z$</td>
<td>......</td>
<td>$A_z$</td>
</tr>
<tr>
<td>Sensor 2</td>
<td>$\omega_x$</td>
<td>Signal</td>
<td>$\omega_x$</td>
</tr>
<tr>
<td></td>
<td>$\omega_y$</td>
<td>......</td>
<td>$\omega_y$</td>
</tr>
<tr>
<td></td>
<td>$\omega_z$</td>
<td>......</td>
<td>$\omega_z$</td>
</tr>
</tbody>
</table>

Table 2.1 Structure of all kinetic data collected through the BESS test experiment

2.3.5 Evaluation of Raters

We have explained the so far the structure of data set collected from the experiment. These signals contain the movement information of each subject, with the assumption that they will
offer relevant information to correlate with errors identification. To investigate the error
information hidden in the acquired signals through machine learning methods, we need a
reference signal to indicate when errors occur during the trials. We have used in this sense the
independent evaluations of four raters. More precisely, videos of all performed experiments were
recorded and then four specialists were asked to provide error flagging for all thirty subjects.
Already this independent evaluation has shown the degree of subjectivism in error evaluation,
and the reliability of each individual rater is actually not equal to each other [43]. In Harrison’s
research, the reliabilities of evaluations from different raters have been investigated. While in our
research, we will use a combination of their independent assessments to decrease the subjective
effects of the raters. Furthermore, to serve as a reference error signal, the error evaluations from
the raters have to be specially processed, to generate a combined signal used as a standard error
signal. Figure 2.4, Figure 2.5 and Figure 2.6 show some example signals to demonstrate that
the evaluations from four raters are close to each other. X-axis stands for time and Y-axis stands
for the evaluation of error. The data point where y value equals “1” indicates that an error
happens at this moment according to rater’s evaluation.

Figure 2.4 Evaluations from four raters of trial 2 for subject 9
2.4 Summary

In this chapter, we discussed the Shimmer equipment we used in our experiment, the components of the whole experimental system and how each component collaborate with others through transmitting data within the whole wireless sensor network. We have indicated the positioning of the seven sensors on the human body and the network architecture of the system. A test experiment was performed with the system to monitor the heart rate of the subjects, and the results showed that this set of system works well with the kinetic data collection. At last, we mainly discussed about the whole BESS test experiments including the subjects involved in our projects, the experiment protocol, the wearable equipment, what was counted as an error in the experiment, the format of the data we collected from the equipment and the standard error
information from the evaluation of the raters. The discussion in this chapter was focused on the various aspects of the BESS test experiment.
Chapter 3: Error Detection

3.1 Introduction
In previous chapters, we have discussed the motivation, the research approach, the experimental equipment, the process of the BESS test experiment and also the data we collected from the experiment. From this chapter to Chapter 5, we will propose two approaches to detect errors by analyzing the signals collected from the wearable sensor network and also the classification of the errors. In this chapter, we will discuss the first error detection approach, detection with neural network along with principal component analysis (PCA) and independent component analysis (ICA). The details will be discussed in the following sections.

3.2 Problems and Proposed Approach
In this chapter, we focus on building the error detection system with neural network. Kinetic signals collected from the BESS test experiments act as the input signal of the neural network. The target output signal is a proper representation of the raters’ evaluation. Namely, the target output signals are some different transformations of the reference error record data provided by the raters. We train several different neural networks with each target output signal respectively. These neural networks have the same input training signal but different output training signals. After analyzing the accuracy of the neural network system with the test data sets, we are able to choose a neural network with the best performance to be our error detecting system. And the representation of the reference error information used to train this neural network is the best choice for representing the reference error signal.
3.3 Pre-processing of the Original Signal

Proper pre-processing procedure is necessary to eliminate noises and to extract the most valuable information from raw data collected from the experiments. In our project, we applied two-step pre-processing, centralization and valid signal data extracting.

3.3.1 Representation of the Collected Kinetic Data

According to the discussion in Chapter 2, we have 42 channels of signals and 210 experimental trials altogether and each trial lasts for 20 seconds. We use a 42-row matrix to represent the whole data set. Figure 3.1 clearly demonstrates the structure of the sensor signal matrix. The rows in matrix represent the 42 channels of signals from seven shimmer units and the columns represent all the experimental trials for 30 subjects with the order shown in figure below. The matrix is denoted by $L_1$. We choose this representation because it helps to simplify the computation in following data analysis procedure.
stands for a 2048-by-1 vector containing data collected from a certain experiment trial.

Figure 3.1 Structure of the sensor data matrix

3.3.2 Centralization

The first step in preprocessing is the centralization of data. This process is a preparation for a following step principal component analysis (PCA). For each vector $L_1(i)$, $i = 1, 2, 3, \ldots, 42$, we subtract the mean value of all the elements in each $L_1(i)$ from itself to form a new vector $L_2(i)$. 

34
Consequently, a new matrix $L_2$ consists of vectors $L_2(i)$, when $i = 1, 2, 3, \ldots, 42$. The Equation (3.1) presents the relationship between $L_1$ and $L_2$.

$$L_2(i) = L_1(i) - \frac{1}{430080} \sum_{j=1}^{430080} L(i, j) \quad (i = 1, 2, 3, \ldots, 42)$$ (3.1)

### 3.3.3 Extraction of Valid Data

In the experiment of this project, there are two types of invalid signal data that need to be eliminated, the calibration trial signal data and the unsuccessful trial signal data. We have learnt from previous chapters that there are seven trials for each subject. However, according to the standard BESS test protocol, only six trials with different postures will be taken by each subject. So what happens to the other trial? Actually, besides six required trials of BESS standard, there is also a calibration trial, which aimed to ensure that there is no erroneous offset causing by the system. These calibration trials cannot be counted as valid experimental trials and their measurements will not be regarded as useful movement information of subjects. Thus they must be excluded from the raw measurement data set. Another type of invalid data is unsuccessful trial data. In these unsuccessful trials, the evaluation of the raters didn’t provide exact time information of errors. These don’t include trials in which subject made ten errors. Most of the situation happens when some subject made too many errors within a very short period of time. Under this circumstance, it was usually too hard for the raters to recognize those errors very precisely. So they put a special mark in their error record report to claim that in this trial the subject has made too many errors that cannot be accurately recognized by themselves. In this case, we have to give up these trials because important standard error information is missing.
After eliminating these two types of invalid trial data, the residual is used to train the neural network of the error detection system. The new valid signal is denoted by $L_3$. Figure 3.2 indicates the calibration trial data that need to be eliminated and the format of the kept valid data.

Figure 3.2 Structure of the valid sensor data matrix

stand for a 2048-by-1 vector containing sensor data collected from a valid experiment trial.

stand for a 2048-by-1 vector containing sensor data collected from a calibration experiment trial.
3.3.4 Signal Compression

In the experiment, each trial lasts for 20 seconds and the sampling rate is 102.4Hz. So there are 2048 sampling points in the signal of each trial. However, when we train the neural network in following section, the sampled signal will serve as input signal and the evaluation of raters will act as target output signal. In previous chapter, we have explained that the raters recorded the error in denomination of 0.1 second. So there are 200 data points in the valuation of rater for each trial. For each trial, there are 2048 sampling points in the input signal but 200 data points in the corresponding target signal while training. But in order to train the neural network, we need to ensure that the dimensions of input signal and target signal are the same. So we have to properly modify the input signal to keep consistency of their dimensions. Before this modification, we need to adjust the number of sampled signal data points for each trial. We keep the first 2000 sampled signal data points in each trial and discard the last 48 data points. This may cause distortion of the final result. So later in chapter 4, we will propose a more reasonable approach to deal with this problem. But in this approach, this potential distortion is inevitable.

Then we calculate the average value of every 10 data points in the input signal and assemble these 200 calculating results to form a new input signal. This modification process successfully avoids the loss of information in the input signal and meanwhile provides a new input signal with the same number of data points with target output signal. This new input signal is denoted by $L_4$. Equation (3.2) and (3.3) expresses the relationship between elements in and $L_4$.

$$l_4(i, j) = \frac{1}{10} \sum_{m=1}^{10} l_3(i, k + m)$$

(3.2)
When,

\[ k = 2048 \times (j/200) + 10 \times (j/200 - 1) \]  

(3.3)

3.4 Principal Component Analysis and Independent Component Analysis for Original Signal

Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are efficient approaches to implement many forms of analysis, including the signal processing and data analysis in our project. PCA is to lower the dimensions of the complex data set to reveal the hidden and simplified structure underlying it [25]. Linear techniques like PCA are simpler and easier to implement than more recent methods considering non-linear transforms [51]. This process can reduce the huge data set we acquired from the series of experiments to a lower dimension so as to lessen the difficulties of the following neural network training task. In our research, we analyze data acquired from seven sensors to detect the possible error made by subjects. While, there might be some redundant information in the whole data set. So in order to eliminate the negative effect of the useless information and also simplify the following analyzing process, we need to separate the useful information with the trashy data. Moreover, ICA can separate the data set into components that are independent to each other. This process can help us to select the most valuable components in the whole data set and implement the following neural networking training with less dimensionality of input signals while avoiding the missing of the information hidden in the original data set.
3.4.1 Application of PCA

PCA statistically transform a set of observation data of possibly correlated variables into uncorrelated variables known as principal components. The number of principal components is equal to the number of variables in original data set. The first principal component accounts for as much of the variability as possible and each succeeding component in turn accounts for as much variability as possible. In other words, the principal components acquired from PCA are ordered by the contribution they made to form the original data space. Since we would like to reduce the dimensionality of the original data set, only some principal components with highest variable possibility are kept, which contain the most valuable information in the data set [25].

The principle of PCA and the detailed statistical procedure are demonstrated in [26]. In our project, the task of implementing PCA aims at dimensionality reduction. According to [26], the principal components are based in a new coordinate system and this transformation from the original data set $X$ to the new set of principal components $T$ is done by the transforming matrix $W$. This transformation is signified in the Equation (3.4).

$$T = XW$$

(3.4)

$W$ is a matrix whose columns are the eigenvectors of $X^T X$. Each principal component we got from this transformation is related to an eigenvector of $X^T X$ and has a corresponding eigenvalue. This eigenvalue reflects how much variability the principal components accounts for, in other words, the order of all the principal components [26]. The eigenvalue of each component is
attached in Appendix B. Thus, we can keep the principal components with higher eigenvalue to reduce the dimensionality.

### 3.4.2 Dimensionality Reduction

Another important question is how many principal components we should keep to balance the number of principal components with the loss of the information of the original data set. We need to select certain number of principal components in order to reduce the dimension of the data set without losing important valuable information.

According to the physical meaning of eigenvalue and eigenvector [45], the corresponding eigenvalue of this component is the energy of each principal component. Therefore, the higher variable possibility a single principal component accounts for, the higher energy it contains. Thus, the group of kept principal components contains certain percentage of energy out of energy contained in all the principal components, according to the number of kept principal components. It is a sum of percentage of energy contained in each kept principal component. The matrix representing all principal components is denoted by $P$. $p(i)$ stands for each principal component and $E_{p(i)}$ stands for energy contained in each principal component. The number of kept principal components is denoted by $k$ and the percentage of the total signal energy contained in all kept $p(i)$ out of total signal energy contained in all principal components is denoted by $R$. The Equation (3.5) exhibits the relationship between them.
Table 3.1 shows the relationship between the kept principal components and the percentage of energy they contain.

<table>
<thead>
<tr>
<th>The Range of Kept Principal Components</th>
<th>Number of Kept Principal Components ($k$)</th>
<th>The percentage of the total signal energy of kept $P(i)$ out of the total signal energy of all principal components ($R$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P (42)–P (30)</td>
<td>13</td>
<td>81.0397142923050 %</td>
</tr>
<tr>
<td>P (42)–P (29)</td>
<td>14</td>
<td>82.8749140914548 %</td>
</tr>
<tr>
<td>P (42)–P (28)</td>
<td>15</td>
<td>84.5973639598833 %</td>
</tr>
<tr>
<td>P (42)–P (27)</td>
<td>16</td>
<td>86.1025804018423 %</td>
</tr>
<tr>
<td>P (42)–P (26)</td>
<td>17</td>
<td>87.4493260182842 %</td>
</tr>
</tbody>
</table>

\[ R = \frac{\sum_{i=42-k}^{42} E_{p(i)}}{\sum_{i=1}^{42} E_{p(i)}} \times 100\% \]  

3.5

Table 3.1 Relationship between the selected principal signal components and the percentage of energy they contains

To decide how many principle components we should keep after PCA, there is a heuristics [52] considering the sum of eigenvalues of kept principle components. The rule is that little variance is gained by retaining additional eigenvalues. In this computation, 13 principle components need to be kept according to the heuristics above. So more than 13 principle components need to be kept. In order to keep a balance between the less principal components and the less loss of the information of original data set. If we choose to keep 80% signal energy of the total signal energy after PCA, 13 principal signal components $p(j)$, ($j=30$, 31, ..., 42) will be kept to form the new data set to represent the original data set. Meanwhile, the choice of keeping more than 80% of the signal energy satisfies the requirement of the heuristics. This new data set is denoted by $L_5$.  

41
3.4.3 Data Distribution after Implementing PCA

To illustrate that after PCA, the selected 13 channels of signal are uncorrelated to each other. We plot the 2D distribution graphs of any two channels from these selected channels that contain the most information. **Figure 3.3** represents the data distribution of signal P(31)-to-P(33) and P(39)-to-P(42).

![Data distribution graphs](image)

**Figure 3.3** (a). Data distribution of signal P(31)-to-P(33), the horizontal axis stands for P(31) and the vertical axis stands for P(33); (b). Data distribution of signal P(39)-to-P(42), the horizontal axis stands for P(39) and the vertical axis stands for P(42)

3.4.4 Definition of ICA

Although we have got uncorrelated principal components after PCA, they are still not strictly independent to each other [47]. The procedure of independent component analysis is to separate the signals we acquired from PCA into independent signal components to prepare for the following neural network training procedure.
Independent component analysis (ICA) is a computational approach to separate observed linear mixtures of several source signals. Here, we assume that source signals are non-Gaussian and independent from each other. For example, in Equation (3.6), we observed $x_1, \ldots, x_n$ which are $n$ mixtures of $n$ independent components $s_1, \ldots, s_n$. Here, we assume that each mixture $x_j$ as well as each independent component $s_k$ is a random variable.

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \ldots + a_{jn}s_n, \text{ for all } j.$$  \hspace{1cm} (3.6)

It is convenient if we use the vector to represent observed mixtures and independent components in Equation (3.7). And $A$ is the matrix with elements of parameters $a_{j1}, \ldots, a_{jn}$.

$$x = As$$ \hspace{1cm} (3.7)

We denote the columns of matrix $A$ by $a_j$. Equation (3.7) can also be written as Equation (3.8).

$$x = \sum_{i=1}^{n} a_is_i$$ \hspace{1cm} (3.8)

Equation (3.8) indicates how the observed data is constituted by mixture of components $s_i$. And we estimate the mixing matrix $A$ by assuming that the components $s_i$ are statistically independent and these independent components must have non-Gaussian distributions. After that, we can simply obtain the independent components by computing the inverse of matrix $A$ denoted by $W$ and independent components are expressed in Equation (3.9).
ICA is to obtain independent components and non-Gaussianity can reflect independence [47].

The kurtosis and negentropy offers us two important measures of non-Gaussianity. In this project we choose negentropy [47] as the measure of non-Gaussianity.

3.4.5 Application of ICA

Figure 3.4 represents the process of Independent Component Analysis.

The preprocessing of ICA consists of three steps, centering, whitening and further processing. Centering original signal matrix \( X \) is subtracting its mean vector \( m = E\{X\} \) to make \( X \) a zero-mean variable. This implies that the obtained independent components matrix \( S \) is also zero-mean.

The process of whitening [47] is that after centering, we transform the original vector \( X \) linearly to form a new white vector \( \tilde{X} \). The components of \( \tilde{X} \) are uncorrelated and their variances equal
unity. In other words, as shown in Equation (3.10), the covariance matrix of $\tilde{X}$ equals identity matrix.

$$E\{\tilde{x}\tilde{x}^T\} = I$$  \hspace{1cm} (3.10)

Further processing [47] means that some application-dependent preprocessing steps which can crucially affect the success of the ICA.

We perform FastICA algorithm to process the data set. First we need to discuss FastICA for one single unit. The basic form of this algorithm is as follows:

1. Choose an initial (e.g. random) weight vector $w$;
2. Let $w^* = E\{xg(w^T x)\} - E\{g'(w^T x)\}w$;
3. Let $w = w^*/\|w^*\|$;
4. If not converged, go back to 2.

The derivation of the algorithm is explained in [47].

In our research, we have 13 channels of signal after PCA. To perform FastICA on multiple signal units, we need to ensure that different weight vectors do not converge to the same value by decorrelating the outputs $w_1^T x, ..., w_n^T x$ after every iteration. We choose a deflation scheme based on a Gram-Schmidt-like decorrelation [47]. With this method, we estimate the independent components one by one. After we obtaining $p$ independent components and $p$ weight vectors $w_1, ..., w_p$, we estimate the $p+1$ weight vector $w_{p+1}$ through one-unit FastICA algorithm and after
every iteration step, subtract from \( w_{p+1} \) the "projections" \( w_{p+1}^T w_j \), \( j = 1, \ldots, p \) of the previously estimated \( p \) vectors, and then renormalize \( w_{p+1} \):

1. Let \( w_{p+1} = w_{p+1} - \sum_{j=1}^{p} w_{p+1}^T w_j \);

2. Let \( w_{p+1} = w_{p+1} / \sqrt{w_{p+1}^T w_{p+1}} \).

We implement the whole process of PCA and ICA with Matlab toolbox (Hyvarinen, A. 1998. *FastICA MATLAB toolbox. Helsinki Univ. of Technology*)[50], since it has strong computational ability that fits our needs. After ICA, we obtain 13 independent signal components that contain the most information of the original data set. These 13 channels of signals are the input signals for the neural network we are going to train in this chapter. The matrix of these signals is denoted by \( L_6 \).

### 3.5 Generating Standard Error Signal

As discussed in chapter 2, we need standard error signal to serve as the standard target signal to train the neural network. The evaluations of four sophisticated raters provide us the standard error information. The record of the raters contains just the moment at which each error happens. The basic error signal represents the error record of the raters in a clear and straight way. To better represent the error, we need an alternate representation of the error record of the raters.
3.5.1 Extract Valid Standard Error Signal Data

In previous section 3.3.3, we have extracted valid sensor signal data. Similarly, in order to keep consistent format of input signal and target output signal used to train the neural network, extraction of valid standard error signal data is also necessary. Thus, the calibration trial data and the unsuccessful trial data are eliminated and the residual should be used to train the neural network in the error detection system.

3.5.2 Digital Pulse Signal Representation

In this section, we will use the digital pulse signal to represent the happening moments of each error evaluated by raters. Figure 3.5 exhibits user interface of the software developed to conduct the data analysis process and to display the evaluation of raters with certain representation as target output signal for neural network training and the sensor signal collected from shimmer WSN equipment in experiments. The signals are displayed trial by trial and the software allows user to choose the displayed trial and which sensor the sensor signal comes from. In this figure, we choose trial 2 from the experiments for subject No.1. The top plot shows the evaluation of four raters represented by digital pulse signal, each color representing evaluation of one rater. The bottom plot shows the corresponding forehead sensor signal. The sum of digital pulse target signal from four raters is denoted by $T_1$. 


Nevertheless, there are some limitations about this representation. In each trial, there are 2048 time slots but usually only less than 10 errors happen during this period of time. If each error is represented by just one 1-value time slot, and all other more than two thousands time slots are 0-value, the total percentage of 1-value in the signal will be too small. When we are training the neural network with these digital signals, these discrete 1-value signals can be very easily ignored. The main reason is that the purpose of neural network training is to find out the hidden relationship between the signal collected by shimmer and the evaluation by the raters. The reality is that the event of making an error doesn’t happen just in a flash. The whole process of making an error lasts for a period of time. And this period of time is reflected in WSN movement signals but not reflected accurately in the discrete 1-value signal representation. So it is not appropriate to use this representation.
To illustrate the discussion above, we use this digital signal as the target output signal to train the neural network and investigate its performance. Its detection is not precise enough comparing to the approach using other representation for reference error information. The details will be discussed in Appendix C.

Therefore, it is obviously not reasonable to represent each error with just one time slot of “1” value signal. The error lasts for about 1 second.

3.5.3 Comparison with Sensor Signal and Observed Deviations

The software tool we developed can visually display the signal collected from the shimmer equipment and the evaluation signal from the raters. In most of the trials, with the software, we observed there is a small deviation between these two signals. Figure 3.6 represents some typical examples of this situation. There is an obvious fluctuation in the sensor signal and it is probably an error. On timeline, each error recorded by the rater is a little bit slower than that reflected in the sensor signals.
Figure 3.6 Observed deviation between raters’ evaluation signal and forehead sensor signal from subject 23 trial 2

This deviation does not only appear in forehead sensor signal. Figure 3.7 shows the observed deviation between evaluation from raters and signals from seven sensors.
Figure 3.7 Observed deviation reflects in signals collected from seven different sensors in subject 05 trial 3: (a). forehead sensor; (b). chest sensor; (c). waist sensor; (d). left-wrist sensor; (e). right-wrist sensor; (f). left-shin sensor; (g). right-shin sensor

This situation happens probably because of practical problem. Detailed speaking, the starting point for the raters to watch the video of the experiments was manually controlled and perhaps the timer for raters has started but the video begins to play a little bit later. So there might be some small gap between these two error records.

To solve this problem, we need to align these two types of signals with a reasonable time deviation. By observation, we obtain that the average deviation between these two types of signals is 0.7 second. So we move every trial of the evaluation of the raters leftward for 7 time slots and supply the 7 slots from the right edge with “0” value.
Figure 3.8 and Figure 3.9 indicate the aligned signal of the example above. The observed deviation between raters’ evaluation signal and sensor signal has been eliminated.

Figure 3.8 Raters’ evaluation signal and forehead sensor signal after alignment from subject 23 trial 2
After alignment, the deviation reflected in signals collected from seven different sensors in trial 3 of subject No.5 has been eliminated.

3.5.4 Digital Window Signal Representation

Based on the discussion above, we can improve the standard error signal by substituting the discrete 1-value points in the previous digital signal with 1-value windows of fixed length. This representation can reflect the real time interval for each error. We still take trial 2 of the first subject as example. Figure 3.10 shows this type of representation and the corresponding sensor...
signal. The sum of window digital target output signals of four raters is denoted by $T_2$. The window length is 10 and the final result is the summation of four raters’ evaluation.

![Figure 3.10 Digital window signal representation for the evaluation of the rater and corresponding sensor signal, subject 01 trial 2.](image)

To illustrate the discussion above, we use this digital signal as the target output signal to train the neural network and investigate its performance. Its detection is not precise enough, compared to the approach using other representations, for reference error information. The details will be discussed in Appendix C.
3.5.5 Gaussian Curve Signal Representation

The time point of each error provided by the raters is the center of the time interval for the observed error. To reflect the possibility of an uncertainty when deciding the precise moment when the observed error occurred, we choose a Gaussian curve to represent each error event. This representation can not only reflect the time interval of the error but also the probability of the error. Figure 3.11 represents an example of Gaussian representation for the target output signal.

![Figure 3.11 Gaussian representation for the target output signal for subject 03 trail 6](image)

To demonstrate that the Gaussian curve can reflect the possibility of each error, we take the trial shown in Figure 3.11 for example. In Figure 3.11 a rater claims that an error happens at the moment \( t_1 \). So \( t_1 \) is the time point at which an error is most likely to happen. The possibility that
an error happens at the moment $t_2$ is less than that at $t_1$. In reality, $t_1$ and $t_2$ are representing the same error. In other words, the performance of the subject at $t_1$ shows stronger possibility an error movement while that at $t_2$ is less possible. Actually $t_2$ is the moment when the subject is adjusting his body balance back to the normal position.

So it is reasonable to say that this Gaussian curve representation reflects how the performance of subject at certain moment is likely to be an error movement.

There are still some parameters of the Gaussian curve that need to be discussed. The height of the Gaussian curve is normalized to 1. The width of the Gaussian curve reflects the length of time interval uncertainty for each error. Equation (3.11) is the original Gaussian function. Parameter $\mu$ is the mean of the bell curve for Gaussian function and reflects the coordinate position of the curve’s maximum point called median point. So $\mu$ is decided by the time point at which each error happens. Another parameter $\sigma$ is the variance of the function and reflects the width of the bell curve. So $\sigma$ reflects how long each error lasts, or the uncertainty in its exact identification moment.

\[
f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.11}
\]

The value of $\mu$ can be confirmed easily. In order to find the proper value for $\sigma$, we need more explorations. Figure 3.12 demonstrates the bell curve when $\sigma$ equals three different values. When $\sigma=5$, the mean width of the bell curve is approximately 15. It means that the error lasts for 1.5 seconds and it’s reasonable. So we choose $\sigma=5$ for Gaussian function curve to represent each error.
3.5.6 Combination of Error Record from Different Raters

We have talked about three options to express isolated error in the standard error signal. No matter which expression we choose for the isolated error, the evaluations from four raters need to be expressed with proper combination. So regardless of the expression choice for the isolated error, for each rater we will generate an error signal vector to express his evaluation for all the trials of all the subjects. The format of the signal vector is in Figure 3.13.
After we generate four error signal vectors for each rater’s evaluation respectively, we have three options to format the real standard error signal.

The first two options, simple summation and selecting maximum value, are to use a combination of these four vectors to generate a new standard error vector.

For the first option, in simple summation, the value of every new time slots in the new standard error vector is the summation of the values in corresponding time slots of four error vectors. This summation target output signal is denoted by $G_1$ and the four error vectors from four raters are denoted by $E_1$, $E_2$, $E_3$ and $E_4$, respectively. Their relationship is shown in equation (3.12).

$$G_1 = E_1 + E_2 + E_3 + E_4$$  \hspace{1cm} (3.12)

**Figure 3.14** and **Figure 3.15** illustrate the Gaussian curve summation combination result for trial 2 and trial 3 of subject No.9.
Figure 3.14 Gaussian curve summation representing the target output signal for subject 9 trial 2

Figure 3.15 Gaussian curve summation representing the target output signal for subject 9 trial 3
For the second option, by selecting the maximum value, the value of every new time slots in the new standard error vector is the maximum of the values in corresponding time slots of four error vectors. This maximum value target output signal is denoted by $G_2$. The relationship between elements of $G_2$, $E_1$, $E_2$, $E_3$ and $E_4$ is shown in equation (3.13).

$$g(i) = \max[e_1(i), e_2(i), e_3(i), e_4(i)]$$

(3.13)

**Figure 3.16** and **Figure 3.17** show the Gaussian curve maximum value result for trial 2 and trial 3 of subject No.9.
The third option is to construct a new matrix of four rows. Each row is the error signal vector from each of the four raters. This matrix serves as the target output signal for training the neural network. Compared with the combination of four vectors, this method can decrease the information loss of the combining process by keeping all the original evaluation of the raters. But after all, the function of our detecting system is to provide the user with just one prediction result instead of four possible results. Considering this, if we use this four-row matrix to be the target output to train the neural network, the prediction result we get will still be four results other than one when we use the system to detect the errors in practical application. This will cause confusion regarding which of these four results is the most accurate prediction we want, or how to decide the weights of each outcome in the final result. This vectorial value target output signal is denoted by $G_3$. The relationship between elements of $G_3$, $E_1$, $E_2$, $E_3$, and $E_4$ is shown in Figure 3.18.
3.5.7 Summary

The main task of our project is to train the neural network with proper input signals and target output signals. Since the target output signals are based on information provided by the raters, we need a proper method to represent the evaluation of the raters with signal data. We have several choices of representation of these signals and choosing the one with the best performance is a vital factor for the accuracy of our detecting system.

3.6 Neural Network Classifier

In this section, we discuss the neural network classifier. There are three different neural networks to be discussed, the main difference between them residing in the target output signal used to train them. These three neural networks are trained with the same input signal, which is denoted by $L_6$ in previous section. The target output signals used to train these three neural networks are $G_1$, $G_2$ and $G_3$, respectively. And the neural networks are denoted by $NN_1$, $NN_2$ and $NN_3$ respectively.
3.6.1 Neural Network Architecture

Now we will discuss the details of these three neural networks and analyze their detection efficiency reflected by the testing result of the neural networks. **Table 3.2** shows some information about the neural networks.

<table>
<thead>
<tr>
<th>Neural Network</th>
<th>$NN_1$</th>
<th>$NN_2$</th>
<th>$NN_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Type</td>
<td>Feed-forward Back-propagation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Function</td>
<td>Levenberg-Marquardt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaption Learning Function</td>
<td>Gradient descent momentum weight and bias learning function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Function</td>
<td>Mean Squared Error Performance Function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Layers</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of neurons in hidden layer</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transfer function</td>
<td>Hyperbolic tangent sigmoid transfer function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training epochs</td>
<td>1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input signals</td>
<td>$L_6$ (13-row, 31800-column)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target output signals</td>
<td>$G_1$ (1-row)</td>
<td>$G_2$ (1-row)</td>
<td>$G_3$ (4-row)</td>
</tr>
</tbody>
</table>

**Table 3.2** Training information of $NN_1$, $NN_2$ and $NN_3$

**Figure 3.19** illustrates the architecture of $NN_1$ and $NN_2$.

**Figure 3.19** Architecture of $NN_1$ and $NN_2

**Figure 3.20** illustrates the architecture of $NN_3$. 
3.6.2 Testing Result and Accuracy Analysis

While training each neural network, we randomly selected 80% of all the trial data to be the training data and other 20% to be testing data. After training them with 1000 epochs, we tested the classification performance of them. Figure 3.21, Figure 3.22 and Figure 3.23 demonstrate the difference between actual output signal and target output signal in their testing result - the less different they are, the more accurate detection the corresponding neural network can provide.

1) Neural network with $G_1$ and $G_2$ as the target output

From the Figure 3.21 and Figure 3.22, we can see that both $NN_1$ and $NN_2$ have acceptable performance while we complete simulation with testing data set. The actual output of them can partially reflect their own target output we expect respectively.
We propose a method to analyze the accuracy of the neural network error detecting system. The accuracy of each neural network detecting system is calculated according to the comparison between the actual output signal and the target output signal of each network’s testing result. We initially use two different thresholds \( TH_1 \) and \( TH_2 \), in actual output signal and target output signal, respectively, to locate the errors. These two thresholds are adjustable. Different values of thresholds result in different final accuracy results, while they have no effect on the detection result. So the thresholds can be adjusted to get the best final accuracy result and will be used in the future application to count the numbers of errors. Then, we set a window with length \( w \). And for every error in target output signal, we place a window around the error, letting the error be at the center of the window. The value of \( w \) should be set to satisfy that the window of adjacent
errors wouldn’t overlap to each other. And we map all the windows onto the timeline of actual output signal. Then, when we compare actual output errors with target output errors with windows, there are two possible relationships between them, illustrated in A and B in Figure 3.22. Figure 3.22. (A) shows the situation when the error in the actual output is located within the window of error in target output. We wrote a script program to find the number of the existences of this situation and noted this number as C. In this process, after we find some errors in actual output satisfying our requirement, we label them as visited. Figure 3.22 (B) shows the situation when in the actual output there is no error located within the window of error in target output. The number of the existences of this situation is noted by E.

Afterwards, we check the unvisited errors in actual output and place a window around each unvisited error, letting the error be at the center of the window. And we map all the windows back to the timeline of target output signal. Then, when we compare target output errors with actual output errors with windows, there are two possible relationships between them, shown in C and D in Figure 3.22. Figure 3.22. (C) shows the situation when there is an error in the target output located within the window of error in actual output. The script program we wrote is used to find the number of the existences of situation C and noted this number as CR. Figure 3.22 (D) shows the situation when there is no error in the target output located within the window of error in actual output. The number of the existences of situation D is noted by ER.
Figure 3.22 Four possible relationship between actual output signal and target output signal of testing result of every neural network

Situation A and D means that the error claimed by raters is detected by our system and for the error detected by system there are corresponding error claimed by raters. These two situations are successful detections. Vice versa, Situation B and C are unsuccessful detections. Thus, the accuracy percentage (AP) of detecting system is shown in Equation 3.14.

\[
AP = \frac{(CR+C)}{CR+C+ER+E} \times 100\%
\]  

(3.14)
With the AP calculation approach above, we can work out the accuracy percentage of $NN_1$:

$$\text{AP (} NN_1 \text{)} = 60.73\%$$

![Graph](image1.png)

**Figure 3.23** Actual output signal and target output signal of $NN_2$ (neural network with $G_2$ as the target output)

With the AP calculation approach above, we can work out the accuracy percentage of $NN_2$:

$$\text{AP (} NN_2 \text{)} = 70.99\%$$

However, in [43], Harrison Brown proposed an error-detecting approach with accuracy of 80%. So we need an improved method, which we will discuss in the following chapter.

2) **Neural network with $G_3$ as the target output**
There are four channels of output in $NN_3$ and each of them stands for error information from a rater.

To illustrate the previous observation that, using $G_3$ as target output, the detection is not accurate enough, we investigated its testing performance and found that there are big differences between actual output and target output. Many errors recorded by raters are not recognized by the neural network system. Another factor is that the final prediction result we require is one single error possibility detection result other than four channels of signals. The task of combining them to form a new detection signal may involve more unstable tuning factors. So we will not consider it as our final choice for the detecting system.
3.6.3 Summary

After analyzing the detection accuracy of three neural networks, we can come into a conclusion that $NN_1$ and $NN_2$ can provide acceptable detection result for errors but still need improvements. However, $NN_3$ is not an appropriate option for error detecting system. In the next chapter, we will discuss another approach of training neural network as an improvement of $NN_1$ and $NN_2$.

3.7 Summary

In this chapter, we have discussed how to preprocess the signals we collected through experimental BESS test with filtering, principal component analysis and independent component analysis. Moreover, we talked about several methods to generate the standard error signal, which is the target output signal used to train the neural network of the detecting system, with different
representation for the evaluation of the raters. Then we build some neural networks with these preprocessed input signals and target output signals. By analyzing the performance of the neural network and the accuracy of the detection result we obtained from the neural network with testing simulation, we come into a conclusion that this method still need to be improved for better accuracy.
Chapter 4: Error Detection with Feature Extraction

4.1 Introduction
In chapter 3, we have discussed the first approach to build the error detecting system. However, the performance is not good enough, so we need to explore other approaches. In this chapter, we will discuss a solution for building a classification neural network with feature extraction.

4.2 Problems and Proposed Approach
In this chapter, we will still build a neural network to perform the classification. The difference with the previous approach is that the input data we use to train the neural network is not the preprocessed original signal. We extract some statistical features from the raw data and detect errors with the neural network classification based on these features. These features formed new input signals to train the neural network. And the target output data is selected from the standard error signals we discussed in the previous chapter. We select $G_1$ and $G_2$ to be the candidate standard error signal in this chapter. Figure 4.1 expresses the detailed process of the approach we will discuss in this chapter.
4.3 Feature Extraction

Feature extraction has been used in human activity recognition based on recorded acceleration signals. In [A5], the author chose mean, energy, variance, standard deviation, fluctuation, mean gradient, absolute gradient and mean crossing rate to be the features. In [A9], Przemyslaw uses features in both time-domain and frequency-domain, mean value, standard deviation, kurtosis,
crest factor and correlation coefficients in time-domain as well as the acceleration energy and signal entropy in frequency domain. In [A16], Jonghun has chosen mean value, standard variation, skewness, kurtosis and eccentricity as features for classification.

The vital factor for the accuracy of the detecting system is the choice of features extracted from the original signal. There are wide range of features as listed above. Considering the characteristics of this project, when subject made errors in the experiment, the movement signal we collected should have more fluctuations than usual. Therefore, the features we select should be able to reflect the fluctuation and changes in the signal. After investigating some literatures using different features of kinetic data to recognize human body activities, mean value, standard deviation and crest factor are chosen as features for our research.

4.3.1 Calculation of Features

To calculate the feature values of the original signal, first we need to set a sliding window with a proper hop length for the calculation. As the window slides over the signal, we calculate three feature values for each window. The values of each feature compose a new feature value matrix. The information of these features will be used to train the neural network as input data. Here, the original data matrix is denoted by $S$. The mean value feature matrix is denoted by $M$. The standard deviation feature matrix is denoted by $D$ and crest factor feature matrix is denoted by $C$. We use this representation in the explanatory formulas and subsequent figures.
4.3.2 Sliding Window Length and Hop Length

Since the extracted feature matrixes will act as input data to train the neural network, the number of data points in input signal matrix, namely the number of total columns in input signal matrix, should be the same with that in target output signal matrix. As we have discussed before, there are 200 columns of data in output signal for each trial and we compress the size of input signal matrix with calculation of average value in previous chapter. Here the calculation of features based on a sliding window will solve this problem. The window length and hop length can decide the number of data points in acquired input signal matrix. The length of sliding window is denoted by \( w \) and the hop length is denoted by \( h \). Because there are 2048 sampling data points in the raw data for each trial, the values of \( w \) and \( h \) should satisfy equation (4.1).

\[
w + h \times (200 - 1) = 2048
\]  

(4.1)

By observation, we confirm the reasonable values of two parameters as shown in equation (4.2) and (4.3).

\[
w = 58
\]  

(4.2)

\[
h = 10
\]  

(4.3)

Figure 4.2 explains the relationship between the raw data matrix and the feature matrix. Figure 4.2 indicates how features are extracted from the raw data for a trial. As the sliding window slides over the raw data signal hop by hop, we calculate the value of features within window to generate the feature matrixes.
Figure 4.2 Relationship between raw data matrix and three feature matrixes and the process of feature extraction based on sliding window

4.3.3 Feature 1 - Mean Value

Mean value feature is the average value of all the data in certain window. The varying-mean value of the signal reflects how stable the subject were in an experimental trial. Higher value means higher acceleration and angular rate of the sensor attached to certain position on the body.
of the subject. It implies the change of subject’s posture, which indicates a possible error. The mean value matrix should be calculated within the sliding window. Elements in the mean value feature matrix $M$ should be calculated in equation (4.4).

\[
M(i, k) = \frac{1}{w} \sum_{j=1}^{w} S(i, (k - 1) \times h + j) 
\]  

(4.4)

**4.3.4 Feature 2 - Standard Deviation**

The standard deviation of the data set in sliding window is the square root of its variance. It indicates the fluctuation of the signal. A high value standard deviation implies that data points in this data set are distributed over a wide range of values.. A low value standard deviation implies that data points in this data set are close to the mean value. When a subject continually keeps the required posture, the movement signal tends to be smooth. When the subject makes an error, the movement signal collected through shimmer units tend to show some larger changes in a short period of time, for the subject tries to fastly rebalance. Thus, the fluctuation of the signal can properly reflect whether the subject makes an error. Elements in standard deviation feature matrix $D$ should be calculated as in equation (4.5).

\[
D(i, k) = \sqrt{\left[ \frac{1}{w-1} \sum_{j=1}^{w} (D(i, (k - 1) \times h + j) - M(i, k))^2 \right]}
\]  

(4.4)

**4.3.5 Feature 3 - Crest factor**

Crest factor is a measurement for the ratio of peak values in a waveform. In our project, this feature represents how many peak values there are in a certain period of signal. Higher crest
factor for a certain period of signal indicates that the subject has made many sharp posture changes. The more changes there are, the higher the probability that an error could happen. Thus the value of crest factor reflects the possibility of error. Similarly to the first two features, we calculate crest factor values for a sliding window; these crest factor values form a new crest factor feature matrix, represented by C. Elements in crest factor feature matrix should be calculated as in equation (4.6).

\[
C(i,k) = \frac{\max\{S(i,(k-1)\times h + 1), S(i,(k-1)\times h + 2), \ldots, S(i,(k-1)\times h + w)\}}{\sqrt{\frac{1}{w} \sum_{j=1}^{w} [S(i,(k-1)\times h + j)]^2}}
\]  

(4.6)

4.4 Neural Network Classifier with Feature Extraction

We have discussed \(NN_1\) and \(NN_2\) in previous chapter and we will improve them with different input training data in this section. The input matrix we use is the combination of three feature matrixes, which is denoted by \(I\). The relationship between input matrix and feature matrixes \(M\), \(D\) and \(C\) is signified in equation (4.7).

\[
I = \begin{bmatrix} M \\ D \\ C \end{bmatrix}
\]

(4.7)

The target output signal we use in this section is the same target signal we have used to train \(NN_1\) and \(NN_2\). They are Gaussian summation expression error signal and Gaussian maximum
value expression error signal, namely, $G_1$ and $G_2$. The two neural networks are denoted by $NN_4$ and $NN_5$ respectively.

### 4.4.1 Neural Network Architecture

Now we will discuss the details of two neural networks and analyze their detection efficiency reflected by the testing result of the neural networks. **Table 4.1** explains some information about the neural networks.

<table>
<thead>
<tr>
<th>Neural Network</th>
<th>$NN_4$</th>
<th>$NN_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Type</td>
<td>Feed-forward Back-propagation</td>
<td>Levenberg-Marquardt</td>
</tr>
<tr>
<td>Training Function</td>
<td>Gradient descent momentum weight and bias learning function</td>
<td></td>
</tr>
<tr>
<td>Adaption Learning Function</td>
<td>Mean Squared Error Performance Function</td>
<td></td>
</tr>
<tr>
<td>Number of Layers</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Number of neurons in hidden layer</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Transfer function</td>
<td>Hyperbolic tangent sigmoid transfer function</td>
<td></td>
</tr>
<tr>
<td>Training epochs</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>Input signals</td>
<td>$I$ (126-row, 31800-column)</td>
<td></td>
</tr>
<tr>
<td>Target output signals</td>
<td>$G_1$ (1-row)</td>
<td>$G_1$ (1-row)</td>
</tr>
</tbody>
</table>

**Table 4.1** Training information about neural networks $NN_4$ and $NN_5$

**Figure 4.3** illustrates the architecture of $NN_4$ and $NN_5$. 
4.4.2 Testing Result and Accuracy Analysis

While training each neural network, we still randomly selected 80% of all the trial data to be the training data and other 20% to be testing data. Figure 4.4 and Figure 4.5 represent the testing result reflected by the difference between actual output signal and target output signal. The less different they are, the more accurate the error detection process is.
With the AP calculation approach above, we can work out the accuracy percentage of $\text{NN}_4$:

$$\text{AP (NN}_4) = 90.14\%$$
With the AP calculation approach above, we can work out the accuracy percentage of $NN_5$:

$$AP \ (NN_5) = 77.81\%$$

After comparing the testing result of two neural networks, it is obviously that $NN_4$ can predict the error more accurately than $NN_5$ and of course more accurately than $NN_1$, $NN_2$ and $NN_3$. Thus, we choose $NN_4$ to be our error detecting system, because, among all the four neural networks we built, it provides us with the most accurate detection of error from the sensor signal collected while subjects are required to complete BESS test experiments.
4.5 Summary

In this chapter, feature extraction is applied to improve the detecting system. We train the neural network with the statistical features data of raw data as input signal. The statistical features we selected should properly reflect the characteristics of the changes of raw data. We selected mean value, standard deviation and crest factor as features. We choose two types of Gaussian expression of reference error signals to be target output signal to train the neural network. Through analysis, the neural network trained with feature matrix as input signal and Gaussian summation expression error signal as target output signal is the most accurate error detecting system.
Chapter 5: Classification of Errors

5.1 Introduction
In chapter 4, we have built an error detecting system. In this chapter, we will go further to classify these errors into classes. This part of the system will not only predict each error but also predict the category for each error.

5.2 Problems and Proposed Approach

5.2.1 Application of Neural Network
The classifier we use is still a back-propagation neural network. The input data we use to train the neural network is the feature matrix we used to train the neural network in chapter 4, and the target output signals are error category information manually collected through watching the videos of the BESS test experiments.

5.2.2 Target Output Signal
In this chapter, for evaluation of each rater, we will separate the errors into six categories according to the BESS standard. The reference error information we used in previous chapter is provided by four raters. However, only one researcher worked on error classification. The researcher was asked to review the experiments videos and meanwhile, for each rater’s evaluation, respectively, label the type of every error. There are altogether six error categories listed below according to the BESS standard.
- Moving the hands off of the iliac crests
- Opening the eyes
- Step stumble or fall
- Abduction or flexion of the hip beyond 30°
- Lifting the forefoot or heel off of the testing surface
- Remaining out of the proper testing position for greater than 5 seconds

The output signal is represented by four six-row matrices. In each matrix, each row representing one of six error categories. The six-row data matrices share the same matrix structure, which is denoted by $TC$. Shown from Figure 5.1, in each row, we use Gaussian curves to express each category of errors in all trials. The order of trials is the same as in input data.

![Figure 5.1 Format of the output matrix TC](image)

There still remains the issue of how the evaluations of four raters are combined together. Similarly to the previous chapter, there are Gaussian summation signal expression that is denoted by $TC_1$ and Gaussian maximum signal expression, denoted by $TC_2$. 

91
5.2.3 Input Signal

The input signal we use to train the neural network in this section is the same as that in previous chapter. Because the error detection of $NN_4$ is the most accurate one, we use its input signal, the feature matrix $I$ in error classification.

5.3 Classification and Analysis

5.3.1 Neural Network Architecture

Table 5.1 shows the features of the neural network.

<table>
<thead>
<tr>
<th>Neural Network</th>
<th>$NN_6$</th>
<th>$NN_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Type</td>
<td>Feed-forward Back-propagation</td>
<td></td>
</tr>
<tr>
<td>Training Function</td>
<td>Levenberg-Marquardt</td>
<td></td>
</tr>
<tr>
<td>Adaption Learning Function</td>
<td>Gradient descent momentum weight and bias learning function</td>
<td></td>
</tr>
<tr>
<td>Performance Function</td>
<td>Mean Squared Error Performance Function</td>
<td></td>
</tr>
<tr>
<td>Number of Layers</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Number of neurons in hidden layer</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Transfer function</td>
<td>Hyperbolic tangent sigmoid transfer function</td>
<td></td>
</tr>
<tr>
<td>Training epochs</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>Input signals</td>
<td>$I$ (126-row, 31800-column)</td>
<td></td>
</tr>
<tr>
<td>Target output signals</td>
<td>$TC_1$ (6-row)</td>
<td>$TC_2$ (6-row)</td>
</tr>
</tbody>
</table>

Table 5.1 Detailed information about the neural network $NN_6$ and $NN_7$

The neural network with $TC_1$ as output data is denoted by $NN_6$ and the neural network with $TC_2$ as output data is denoted by $NN_7$. Figure 5.2 shows their architecture.
5.3.2 Testing Result and Accuracy Analysis

While training each neural network, we still randomly selected 80% of all the trial data to be the training data and other 20% to be testing data. Figure 5.3 and Figure 5.4 represent the testing result for six categories of errors respectively. The classification accuracy is reflected by difference between actual output signal and target output signal. The less different they are, the more accurate detection the corresponding neural network can provide.

1) The testing result of neural network with $T_{C_1}$ as target output signal
Figure 5.3 The actual output signal and target output signal of testing simulation for \( NN_6 \) for each type of error: (a) “moving the hands off of the iliac crests”; (b) “opening the eyes”; (c) “step stumble or fall”; (d) “abduction or flexion of the hip beyond 30°”; (e) “lifting the forefoot or heel off of the testing surface”; (f) “remaining out of the proper testing position for greater than 5 seconds”

With the AP calculation approach above, we can work out the accuracy percentage for each type of error of \( NN_6 \):

\[
\text{AP (} NN_6 \text{)} = [66.67 \%, 100\%, 100\%, 91.53 \%, 86.67 \%, 41.94 \%]
\]

2) The testing result of neural network with \( TC_2 \) as target output signal
Figure 5.4 Actual output signal and target output signal of testing simulation for $NN_7$ for each type of error:
(a). error “moving the hands off of the iliac crests”; (b). error “opening the eyes”; (c). error “step stumble or fall”; (d). error “abduction or flexion of the hip beyond 30°”; (e). error “lifting the forefoot or heel off of the testing surface”; (f). error “remaining out of the proper testing position for greater than 5 seconds”

With the AP calculation approach above, we can work out the accuracy percentage for each type of error of $NN_7$:

$$AP\ (NN_7) = [100\%, 100\%, 97.14\%, 100\%, 84.38\%, 36.67\%]$$

After comparison, we can come into a conclusion that $NN_7$ can classify the errors made by subject with higher accuracy than $NN_6$. So $NN_7$ is an efficient error classification system for Balance Error Scoring System. The comparison between accuracy of $NN_6$ and $NN_7$ is shown in

Figure 5.5. The average AP for $NN_6$ (81.135%) is higher than that of $NN_7$ (86.365%).
5.4 Each Sensor’s Contribution

In this section, we will try to evaluate the contribution of each sensor involved in the experiment, in order to reduce the number of sensors we need to conduct the experiment. We only keep those sensors making dominant contributions and abandon those with less contributions, in order to make the experimental system more cost-effective.

5.4.1 Evaluation Approach

According to the neural network we have built, we can extract the weights of the 2-layer neural network to calculate each input channel’s contribution to the output. The weights reflect the importance of input data from a certain channel. These input signals are collected from certain
sensors. So it is not difficult to estimate the importance of each sensor. Here, only the most accurate neural network \( NN_4 \) will be used.

First, we need to extract the weights from neural networks. \( IW \) stands for Input Layer Weights Matrix and \( LW \) stands for Hidden Layer Weights Matrix.

Second, we multiply \( IW \) by \( LW \), and the result represents the contribution of each row in input feature matrix \( I \) into the output signal. It is a 126-column vector denoted by \( W \) and each column represents the contribution from each input channel respectively. Equation (5.1) illustrates the relationship between \( IW \), \( LW \) and \( W \).

\[
W = IW \times LW
\]  
(5.1)

Thirdly, the 126 input channels consist of 3 types of features calculated from the 42 channels original input signals. So the corresponding 126 weights are based on different kinds of input features. For each feature, we calculate the percentage of each weight in all the 42 weights. And the result is three 42-column vectors, which are denoted by \( W_{p1} \), \( W_{p2} \) and \( W_{p3} \). Equation (5.2) demonstrates the relationship between the elements of \( W \), \( W_{p1} \), \( W_{p2} \) and \( W_{p3} \).

\[
w_{p1}(i) = \frac{w(i)}{\sum_{i=1}^{42} w(i)} \times 100\% , \quad w_{p2}(i) = \frac{w(i)}{\sum_{i=43}^{84} w(i)} \times 100\% , \quad w_{p3}(i) = \frac{w(i)}{\sum_{i=85}^{126} w(i)} \times 100\%
\]  
(5.2)
Fourthly, for 42 weight-percentages in $W_{p1}, W_{p2}$ or $W_{p3}$, every 6 weights are related to one sensor. So for each feature we separate 42 weights into 7 groups and each group represents one of 7 sensors. Then we sum up 6 weights in each group to form three new matrix denoted by $W'_{p1}, W'_{p2}$ and $W'_{p3}$. The relationship between them is shown in Equation (5.3).

$$w'_{p1}(i) = \sum_{j=(i-1)\times7+1}^{j=7i} w_{p1}(j), \quad w'_{p2}(i) = \sum_{j=(i-1)\times7+1}^{j=7i} w_{p2}(j), \quad w'_{p3}(i) = \sum_{j=(i-1)\times7+1}^{j=7i} w_{p3}(j)$$ (5.3)

At last, $W'_{p1}, W'_{p2}$ and $W'_{p3}$ contain contribution percentages from seven sensors based on 3 features. Because the contribution is shown by percentage other than absolute value, there is no problem about different levels of impact from 3 features. Thus, for each sensor, we add the 3 contributions base on 3 features. The result is a 7-column vector denoted by $W_s$ representing the relative contribution every sensor unit made. The relationship between them is shown in Equation (5.4).

$$W_s = W'_{p1} + W'_{p2} + W'_{p3}$$ (5.4)

Table 5.2 shows the contribution of each sensor.

<table>
<thead>
<tr>
<th>$W_s$</th>
<th>$w_s(1)$</th>
<th>$w_s(2)$</th>
<th>$w_s(3)$</th>
<th>$w_s(4)$</th>
<th>$w_s(5)$</th>
<th>$w_s(6)$</th>
<th>$w_s(7)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>27.76</td>
<td>12.6</td>
<td>27.76</td>
<td>21.36</td>
<td>38.07</td>
<td>38.16</td>
<td>14.76</td>
</tr>
<tr>
<td>Sensor</td>
<td>forehead</td>
<td>chest</td>
<td>waist</td>
<td>left wrist</td>
<td>right wrist</td>
<td>left shin</td>
<td>right shin</td>
</tr>
</tbody>
</table>

Table 5.2 Contribution of each sensor
From Table 5.2, we can see that \( w_s(1), \ w_s(3), \ w_s(4), \ w_s(5) \) and \( w_s(6) \) are much larger than \( w_s(2) \) and \( w_s(7) \). Thus the corresponding sensors that are highlighted make the most contribution among all the sensors. So we only keep forehead sensor, waist sensor, left wrist sensor, right wrist sensor and left shin sensor. However, will the signals collected by these 5 sensors be enough to detect errors with high accuracy? We still need to confirm it.

5.4.2 Validation

In this section, we make some changes on feature matrix input \( I \) to form a new input data set \( I' \). In \( I' \), the input channels related to chest sensor and right shin sensor are excluded and only the rest are kept. The relationship between feature matrixes and \( I' \) is shown in Equation (5.5).

\[
I' = \begin{bmatrix}
M(:,1:6) \\
M(:,13:36) \\
D(:,1:6) \\
D(:,13:36) \\
C(:,1:6) \\
C(:,13:36)
\end{bmatrix} \tag{5.5}
\]

Then, we use \( I' \) to be the input signal and \( G_2 \) to be the target output signal to train a new neural network \( NN_8 \). Here we only discuss the testing performance other than the details of \( NN_8 \). Figure 5.5 signifies the actual output and the target output of the testing result. From the figure we can see that the actual output can provide good error detection. In conclusion, we can only use five sensors to collect kinetic signal in BESS test experiment and still get good error detection results with the neural network system we have built. In summary, it is more cost-
effective to use only five sensors on forehead, waist, left wrist, right wrist and left shin to complement the BESS test experiment. And then use data collected by these five sensors to train the neural network. The neural network error detecting system can still detect most of the errors made by the subject.

![Figure 5.6 Actual output and the target output signal of the testing simulation for $NN_8$](image)

### 5.5 Summary

In this chapter, we discuss the classification of the error with neural network and the evaluation of each sensor’s contribution to reduce the number of sensors used in the system, while ensuring a good error detection accuracy. The neural network in this chapter is trained with a feature matrix containing three features of original input data, and two kinds of Gaussian error curves as target output signal. The evaluation of the sensors’ contribution is based on the weights of the hidden and output layers of the neural network we trained in chapter 4 for the error detecting
system, since the weights of the neural network reflect the importance of input data on certain channels. According to the testing data set, the performance of error classification neural network is good enough to classify six types of error in BESS test experiments. After two sensors with relatively less contribution are excluded from the error detecting system, the neural network trained with less input data has a good performance as well.
Chapter 6: Conclusion and Future Work

6.1 Conclusion

In conclusion, the error detecting system for BESS test experiment developed in this thesis is accurate enough to provide researchers, as well as other users, with reliable evaluation on the balance status of the subjects or patients. It is a more cost-effective and objective approach to detect the errors made by subjects or patients when they complete the BESS test experiment, which is a standard protocol to help users to diagnose whether the patient has recovered from concussions caused by sports injuries.

6.2 Future Work

In future research work, the system still needs some improvements. Firstly, more experimental data, especially experimental data from people of different ages and health status, need to be accumulated and be used to train the neural network, in order to improve the error detecting accuracy of the system. This will help the system to perform well for different patients or subjects. Secondly, the master controller in this system is a personal computer - we can try to substitute it with mobile devices, so that the software will run on a portable platform. This will deeply enhance the application of the system. It can be used not only in research organization or clinical diagnostic but also in casual outdoor activities, with its convenience and flexibility. The most important remaining problem remains is that whether the computing ability of the mobile devices is strong enough to deal with the error detecting tasks. Last but not least, the sensor we used in this project is a shimmer wireless system, usually used in research, but not small and light enough to be used in clinical diagnostic. So it is important to develop a more advanced
sensor unit, with advantages in terms of size and weight, in order to improve its wearable advantage.
Bibliography


Social-Informatics and Telecommunications Engineering).


Appendices

Appendix A  Balance Error Scoring System Protocol

The Balance Error Scoring System provides a portable, cost-effective, and objective method of assessing static postural stability. In the absence of expensive, sophisticated postural stability assessment tools, the BESS can be used to assess the effects of mild head injury on static postural stability. Information obtained from this clinical balance tool can be used to assist clinicians in making return to play decisions following mild head injury.

The BESS can be performed in nearly any environment and takes approximately 10 minutes to conduct.

A.1  Materials

1) Testing surfaces
   -two testing surfaces are need to complete the BESS test: floor/ground and foam pad.

   1a) Floor/Ground: Any level surface is appropriate.

   1b) Foam Pad (Power Systems Airex Balance Pad 81000)

      Dimensions:  Length: 10"
                    Width:  10"
                    Height: 2.5"

      The purpose of the foam pad is to create an unstable surface and a more challenging balance task, which varies by body weight. It has been hypothesized
that as body weight increases the foam will deform to a greater degree around the foot. The heavier the person the more the foam will deform. As the foam deforms around the foot, there is an increase in support on the lateral surfaces of the foot. The increased contact area between the foot and foam has also been theorized to increase the tactile sense of the foot, also helping to increase postural stability. The increase in tactile sense will cause additional sensory information to be sent to the CNS. As the brain processes this information it can make better decisions when responding to the unstable foam surface.

2) Stop watch
   -necessary for timing the subjects during the 6, twenty second trials

3) An assistant to act as a spotter
   -the spotter is necessary to assist the subject should they become unstable and begin to fall. The spotter’s attention is especially important during the foam surface.

4) BESS Testing Protocol
   -these instructions should be read to the subject during administration of the BESS

5) BESS Score Card
(the Testing Protocol and a sample Score Card are located at the end of this document)

A.2 BESS Test Administration

1) Before administering the BESS, the following materials should be present:
   - foam pad
   - stop watch
   - spotter
   - BESS Testing Protocol
   - BESS Score Card

2) Before testing, instruct the individual to remove shoes and any ankle taping if necessary. Socks may be worn if desired.

3) Read the instructions to the subject as they are written in the BESS Testing Protocol.

4) Record errors on the BESS Score Card as they are described below.

A.3 Scoring the BESS

Each of the twenty-second trials is scored by counting the errors, or deviations from the proper stance, accumulated by the subject. The examiner will begin counting errors only after the individual has assumed the proper testing position.

Errors: An error is credited to the subject when any of the following occur:

◆ moving the hands off of the iliac crests
◆ opening the eyes
◆ step stumble or fall
◆ abduction or flexion of the hip beyond 30°
◆ lifting the forefoot or heel off of the testing surface
◆ remaining out of the proper testing position for greater than 5 seconds

-The maximum total number of errors for any single condition is 10.

<table>
<thead>
<tr>
<th>Normal Scores for Each Possible Testing Surface</th>
<th>6.1.2 Firm</th>
<th>6.1.3 Foam</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.4 Double Leg</td>
<td>6.1.5</td>
<td>.0</td>
</tr>
<tr>
<td>6.1.7 Single Leg</td>
<td>6.1.82</td>
<td>.45 ±</td>
</tr>
<tr>
<td>6.1.10 Tandem</td>
<td>6.1.11</td>
<td>.9</td>
</tr>
<tr>
<td>6.1.13 Surface Total</td>
<td>6.1.14</td>
<td>3.</td>
</tr>
<tr>
<td>6.1.16 BESS Total</td>
<td>6.1.15</td>
<td>8.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum Number of Errors Possible for Each Testing Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firm Surface</td>
</tr>
<tr>
<td>Foam Surface</td>
</tr>
<tr>
<td>Double Leg Stance</td>
</tr>
<tr>
<td>Single Leg Stance</td>
</tr>
<tr>
<td>Tandem Stance</td>
</tr>
<tr>
<td>Surface Total</td>
</tr>
</tbody>
</table>

-if a subject commits multiple errors simultaneously, only one error is recorded. For example, if an individual steps or stumbles, opens their eyes, and removes their hands from their hips simultaneously, then they are credited with only one error.

-subjects that are unable to maintain the testing procedure for a minimum of five seconds are assigned the highest possible score, ten, for that testing condition.

A.1 Testing Positions

1) Firm/Ground
Double leg stance: Standing on a firm surface with feet side by side (touching), hands on the hips and eyes closed

Single leg stance: Standing on a firm surface on the non-dominant foot (defined below), the hip is flexed to approximately 30° and knee flexed to approximately 45°. Hands are on the hips and eyes closed.

Non-Dominant Leg: The non-dominant leg is defined as the opposite leg of the preferred kicking leg

Tandem Stance: Standing heel to toe on a firm surface with the non-dominant foot (defined above) in the back. Heel of the dominant foot should be touching the toe of the non-dominant foot. Hands are on the hips and their eyes are closed.
2) Foam

Double leg stance: Standing on a foam surface with feet side by side (touching), with hands on the hips and eyes closed

Single leg stance: Standing on a foam surface on the non-dominant foot (defined below), with hip flexed to approximately 30° and knee flexed to approximately 45°. Hands are on the hips and eyes closed.

Non-Dominant Leg: The non-dominant leg is defined as the leg opposite of the preferred kicking leg

Tandem Stance: Standing heel to toe on a foam surface with the non-dominant foot (defined above) in the back. Heel of the dominant foot should be touching the toe of the non-dominant foot. Hands are on the hips and their eyes are closed

WARNING: Trained personnel should always be present when administering the BESS protocol. Improper use of the foam could result in injury to the test subject.
## Balance Error Scoring System (BESS)

### Balance Error Scoring System – Types of Errors

1. Hands lifted off iliac crest  
2. Opening eyes  
3. Step, stumble, or fall  
4. Moving hip into > 30 degrees abduction  
5. Lifting forefoot or heel  
6. Remaining out of test position > 5 sec  

The BESS is calculated by adding one error point for each error during the 6 20-second tests.

### SCORE CARD:

<table>
<thead>
<tr>
<th></th>
<th>FIRM Surface</th>
<th>FOAM Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double Leg Stance (feet together)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single Leg Stance (non-dominant foot)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tandem Stance (non-dominant foot in back)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Scores:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BESS TOTAL:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which foot was tested: (Left/ Right)  
(i.e. which is the non-dominant foot)
## Appendix B  The Energy of Principal Components

The Energy of each principal component of signal acquired from principal component analysis:

<table>
<thead>
<tr>
<th>P(42)</th>
<th>0.00622659055678219</th>
<th>P(35)</th>
<th>0.00229188932661905</th>
<th>P(28)</th>
<th>0.000837992472707775</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(41)</td>
<td>0.0054433535415688</td>
<td>P(34)</td>
<td>0.00198824081358293</td>
<td>P(27)</td>
<td>0.000732305811203982</td>
</tr>
<tr>
<td>P(40)</td>
<td>0.00467868807314094</td>
<td>P(33)</td>
<td>0.00149517799824669</td>
<td>P(26)</td>
<td>0.000655207858246806</td>
</tr>
<tr>
<td>P(39)</td>
<td>0.00438656112705764</td>
<td>P(32)</td>
<td>0.00122875172788803</td>
<td>P(25)</td>
<td>0.00052970062026888</td>
</tr>
<tr>
<td>P(38)</td>
<td>0.00396459814262314</td>
<td>P(31)</td>
<td>0.0010925592752573</td>
<td>P(24)</td>
<td>0.000482669143649068</td>
</tr>
<tr>
<td>P(37)</td>
<td>0.00290200552842161</td>
<td>P(30)</td>
<td>0.00101818698123522</td>
<td>P(23)</td>
<td>0.000450378386435929</td>
</tr>
<tr>
<td>P(36)</td>
<td>0.00271019120865509</td>
<td>P(29)</td>
<td>0.000892846663227128</td>
<td>P(22)</td>
<td>0.000409267772854650</td>
</tr>
<tr>
<td>P(21)</td>
<td>0.000382562814063087</td>
<td>P(14)</td>
<td>0.000247695561392088</td>
<td>P(7)</td>
<td>0.000141217566516839</td>
</tr>
<tr>
<td>P(20)</td>
<td>0.00035037396195670</td>
<td>P(13)</td>
<td>0.00021248171752561</td>
<td>P(6)</td>
<td>0.000116512063982094</td>
</tr>
<tr>
<td>P(19)</td>
<td>0.000333852350039922</td>
<td>P(12)</td>
<td>0.000192222670849765</td>
<td>P(5)</td>
<td>0.000106235159477420</td>
</tr>
<tr>
<td>P(18)</td>
<td>0.000309452890245678</td>
<td>P(11)</td>
<td>0.000182792040964499</td>
<td>P(4)</td>
<td>9.89354456925667e-05</td>
</tr>
<tr>
<td>P(17)</td>
<td>0.000298552420903016</td>
<td>P(10)</td>
<td>0.000165673443595660</td>
<td>P(3)</td>
<td>8.81615286966330e-05</td>
</tr>
<tr>
<td>P(16)</td>
<td>0.000285494066241477</td>
<td>P(9)</td>
<td>0.000160837835986920</td>
<td>P(2)</td>
<td>7.91972396879334e-05</td>
</tr>
</tbody>
</table>
Appendix C Testing result and Analysis of Neural Network with Digital Pulse Signal Representation and Digital Window Signal Representation as Target Signal

C.1 Neural Network Architecture

Now we will discuss the details of two neural networks and analyze their detection efficiency that is reflected by the testing result of the neural networks. Table below explains some information about the neural networks $NN_9$ and $NN_{10}$.

<table>
<thead>
<tr>
<th>Neural Network</th>
<th>$NN_9$</th>
<th>$NN_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Type</td>
<td>Feed-forward Back-propagation</td>
<td></td>
</tr>
<tr>
<td>Training Function</td>
<td>Levenberg-Marquardt</td>
<td></td>
</tr>
<tr>
<td>Adaption Learning Function</td>
<td>Gradient descent momentum weight and bias learning function</td>
<td></td>
</tr>
<tr>
<td>Performance Function</td>
<td>Mean Squared Error Performance Function</td>
<td></td>
</tr>
<tr>
<td>Number of Layers</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Number of neurons in hidden layer</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Transfer function</td>
<td>Hyperbolic tangent sigmoid transfer function</td>
<td></td>
</tr>
<tr>
<td>Training epochs</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>Input signals</td>
<td>$L_6$ (13-row, 31800-column)</td>
<td></td>
</tr>
<tr>
<td>Target output signals</td>
<td>$T_1$ (1-row)</td>
<td>$T_2$ (1-row)</td>
</tr>
</tbody>
</table>

Figure below illustrates the architecture of $NN_9$ and $NN_{10}$. 
C.2 Testing Result and Analysis

While training each neural network, we still randomly selected 80% of all the trial data to be the training data and other 20% to be testing data. Figures below represent the testing result reflected by difference between actual output signal and target output signal. The less different they are, the more accurate detection the corresponding neural network can provide.
With the AP calculation approach above, we can work out the accuracy percentage of $NN_9$:

$$AP\ (NN_9) = 58.68\%$$

![Graph showing actual output and target output for $NN_9$.]

After comparing the testing result of two neural networks, it is obviously that other neural networks especially $NN_4$ can predict the error more accurately than $NN_9$ and $NN_{10}$. Thus, we choose $NN_4$ to be our error detecting system because it provides us with the most accurate detection of error from the sensor signal collected while subjects are required to complete BESS.
test experiments. And it is clearly demonstrated that the error detecting ability $NN_o$ and $NN_{10}$ are not so optimal.

Therefore, it is obviously not reasonable to represent target signal with $T_1$ or $T_2$. 