THE UNIVERSITY OF NEW SOUTH WALES

SCHOOL OF ELECTRICAL ENGINEERING
AND TELECOMMUNICATIONS

Head Movement in Task Analysis

by

Robert William Makepeace

Thesis submitted as a requirement for the degree

Bachelor of Engineering (Electrical Engineering)

Submitted: November 1, 2014
Supervisor: A/Prof Julien Epps

Student ID: z3331578
Topic ID: JE34

Abstract

This thesis investigates the potential for analysing head movement in automated task analysis. The nature of head movement has been characterised for some daily-life tasks, in employment, educational and recreational environments. This analysis shaped the machine learning techniques, which predict the type and intensity of activities given training data. A practical and inexpensive hardware prototype has been developed and tested to track head movement. Due to a lack of a dataset, an experiment has conducted to explore distinguishing everyday human activities, and also different levels of mental load and transitions between tasks. The system has a classification accuracy of 84% on detecting activities using the k-Nearest Neighbour classifier. This novel technology has potential applications in dynamic computer human interaction and in conjunction with other sensors.

Acknowledgements

I would like to express my appreciation for everyone who has helped me throughout the thesis project. It has been a enjoyable journey and I have learnt a large amount along the way.

I sincerely thank my supervisor, Julien, who has provided great guidance and advice throughout the process.

I wish to acknowledge the UNSW Coop Scholarship which enabled me to focus my attention on my studies throughout my degree. They also provided invaluable industrial training which has shaped my skills as an engineer.

Special thanks goes to the generosity and cooperation of the experimental participants. Without their assistance, collection of the dataset would not have been possible.

Finally, thanks to the support of my family and friends and their assistance and support in the project.

Abbreviations

AD	Allan Deviation
ADC	Analog to Digital Converter
AVAR	Allan Variance
ANN	Artificial Neural Network
EEG	Electroencephalography
EVA	Exposure Variation Analysis
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HMD	Head Mounted Display
HMM	Hidden Markov Model
I2C	Inter-Integrated Circuit (digital protocol for communication)
IR	Infrared
IMU	Inertial Movement Unit
kNN	k-Nearest Neighbours
LDA	Linear Discriminant Analysis
MEMS	Micro Electro-Mechanical Systems
ROC	Receiver Operator Characteristic
SCL	Serial Clock Line
SDA	Serial Data Line
SVM	Support Vector Machine

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

With the development of wearable computing, the human-computer interface is rapidly changing to meet the demands of consumers. Advances such as the Google Glass, have broken down societal perceptions and boundaries allowing for more intimate interactions of computers. The goal of wearable computing is for a more natural and dynamic relationship to improve the utility of devices in everyday life.

Fitness tracking wristbands have gained popularity, allowing users to log their exercise and sleeping patterns. This trend will continue to seamlessly embed computers within people's lives with more natural interaction. The capacity to understand the nature of what the user is currently occupied with will shape the way the computer behaves. This passive intelligence is enabled by automated task analysis.

Task analysis is the study and classification of human activity. Automated task analysis is the training of computers to passively recognise different activities and adapt their behaviour and interaction with humans. This has huge potential applications in wearable computing and in the broader human machine interface. Through better understanding of what the user is currently doing, the computer can dynamically adapt their behaviour.

Current techniques being investigated for this application include eye movement/blinking, and audio/visual analysis. While these techniques are somewhat successful in their detection, they lack practicality for usage in everyday life. A desirable system should be unobtrusive for the subject, and have the capacity for real time analysis.

Machine learning methods find patterns in large datasets in order to make predictions about future outcomes. These techniques enable a computer to find the underlying patterns through a process of training with the desired outcomes. This method of analysing complicated phenomena is powerful and is able to achieve accurate predictions.

1.2 Motivation

Popular devices such as the Occulus Rift and Space Glasses utilise head tracking to position their user in an augmented reality space. The tracking is seamless in capturing all the movement, otherwise the user would get easily unsettled. With these recent developments in head tracking, head movement has been proposed as a novel indicator of what are a person's current activities. Today's sensors, such as those found in a modern smart phone, are capable of precisely tracking a person's head movement, both in terms of orientation and position.

This thesis explores the ability to use head movement as the information source for task analysis. Head movement, through deliberate or subconscious actions provides insight into deeper activities within the brain. Unlike many of the alternate techniques, it is less invasive and most applicable to common usage.

Modern lifestyles are often sedentary, particularly in work and recreation activities. Having a stationary subject increases the difficulty of a task classification problem. The characteristic factors defining activities are motor movement, communication, cognitive load and perceptual load. From these categories, a model of human focus can be constructed.

1.3 Potential Applications

The range of potential applications include monitoring of human performance and activity. For example, on-line training courses are one application where the content delivery can be changed depending on the user's response to the material. Traditional teachers naturally use feedback from students to pace their lessons and find areas where the students are struggling. Enabling computers to use this technique personalises learning and can achieve better educational outcomes.

This technology could be used to monitor employees in high-stress/performance reliant industries such as air-traffic controllers and call centre operators. Not withstanding the obvious ethical issues, the technology could be used to schedule breaks, or allocate resources to optimise the performance of the employees.

Relevant content delivery, with software such as *Google Now*, attempts to provide information to the user depending on what they are doing. Currently the software relies on smart phone data such as GPS position, time of day and internet browsing history. Coupling this system with activity recognition, would improve the accuracy of this information. This has large potential in wearable computing and tele-assistance delivering the right information as it is needed. Other usages could be applicable in gaming, virtual reality and home automation.

To enable these applications, information about the nature and intensity of activities is required. Other useful information includes the transitions between activities and the mood of the subject (restless, tired, happy, etc). The majority of interesting applications are in the employment, education and recreation fields which mostly consist of sedentary activities.

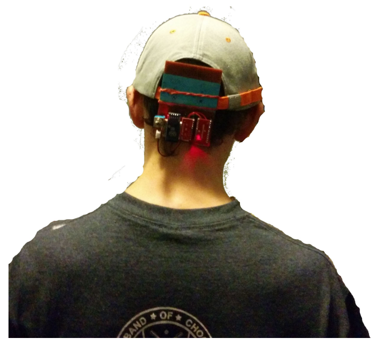


Figure 1.1: Device Prototype

1.4 Problem Definition

This thesis explores what information about a user's activity can be extracted from their head movement. The problem involves training a computer to distinguish between aspects of an activity, developing a simple hardware prototype to measure head movement and forming a basic dataset with an experiment on twenty people.

The specific sub-problems are distinguishing between basic daily activities, determining cognitive load levels and determining transitions between activities. While doing these classifications with a controlled dataset, effort has been made to consider extensions to real world situations and real time processing.

1.5 Objectives

This thesis aims to:

- Understand the nature of head movement in the type and intensity of basic daily tasks
- Explore the potential of machine learning to train and automatically recognise human activities
- Design and execute an experiment to make a dataset

1.6 Report Structure

The report is structured into the following chapters:

Chapter 2 details the literature review of the research area. In particular it analyses the related work of task analysis and head tracking. Machine Learning algorithms are explored with reference to the feature extraction and classifier training.

Chapter 3 outlines the methods, experimental design and preliminary experimental results. The design and characterisation of the recording device are discussed with the broader software system architecture. The implementation of the experiment is explained as well as how the classification results will be assessed.

Chapter 4 covers the results and discussion of the experiment and system performance. System Limitations and avenues for improvement are explored as well.

Chapter 2

Literature Review

2.1 Overview

This chapter outlines the Literature Review for the thesis. The review covered three main areas of investigation: current techniques in task analysis, existing head movement research and the machine learning algorithms which process and analyse the data signals.

2.2 Definitions

Task Analysis - The study of the nature of human tasks. Automated task analysis processes sensor data to attempt to predict information about the nature of human activities.
Head Tracking - Capturing the movement and position of the head and neck. The idea is to isolate the movement of just the head compared to the external body movement.
Machine Learning - The analysis of data to find statistical patterns to predict class outcomes. The process involves pre-processing the data, extracting the distinctive features and mapping these features using a classifier.
Cognitive Load - Cognitive load is an aspect of mental application in required when undertaking a task. It relates to the working memory of the brain.
Perceptual Load - Perceptual load relates to the amount of visual information to process.
Human Activity Taxonomy - The study of human activity and tasks. This involves

defining and labelling different activities, and understanding the relationships of smaller tasks to the bigger purpose.

Feature Vector - a vector of numerical attributes which describe the signal within a time segment.

2.3 Human Activity Taxonomy

2.3.1 Typical Daily Life Breakdown

Before looking at how human activities are analysed, the statistics of average daily activities should be considered. Human activity taxonomy attempts to quantify the complicated and diverse nature of daily tasks. The American Time Use study, details the time spent on various activities through user annotation [27]. With a broad demographic of thousands of participants in the survey categories, the survey breaks down activities by time of day, profession, age and other factors. Another study, "How Australians Use Their Time" by the ABS breaks down the broad types of activities by their function (E.g. working) [44]. The function/purpose of activities is often much more complicated in determining than the basic task. Table 2.1 summarises the results for the average person (Note the numbers vary significantly between genders, ages and lifestyles).

Table 2.1 shows the figures for these typical activities. It is interesting to note these are divided up on functional or purpose based activities, say for example a conversation could be counted as employment, childcare, social or purchasing. To get these broader categories the machine needs to learn the context of the activity, a much harder challenge. More relevant to this experiment are the fundamental building block actions which make up more complex tasks. The basic subtasks include reading, writing, conversation, observing, moving/physical activity.

Most common activities are a combination of these tasks, and are often complicated with layers and phases of tasks with a broader purpose. To successfully build up these tasks, the subtasks need to be identified, and also the transitions between them. Machines can

Table 2.1: Breakdown of Use of Time Activities [44]. Note the large proportion of time taken up by Employment/Education and Recreation activities, most of which are sedentary.

Activity	**Time Taken (hrs per day)**
Personal Care (Sleeping, Eating, Hygiene)	10:57
Employment/Education	3:57
Domestic	2:15
Childcare	0:41
Purchasing	0:48
Social and Volunteering	1:02
Recreation Leisure	4:13
Other	0:07

then build up information with temporal relationships and historical trends to establish the context of a subtask to identify the broader task. One model used to distinguish human activity uses four features - motor, communicative cognitive load and perceptual load. At the simplest level these factors can be rated as low or high.

2.4 Physical Task Analysis

Automatic Task Analysis is the analysis of human activities using automated sensors and computers. The purpose is to collect information about what activities the subject is doing and the intensity of the activity. This has large potential for computers having dynamic interaction with humans [20][45]. Physical activities can be broadly classified by the amount of physical movement - sedentary, light, moderate and vigorous [55]. The majority of existing task analysis involves tracking and analysing physical movement on the higher end of the spectrum of movement. The existing research can be broken into two categories: analysis involving either external or wearable sensors.

2.4.1 Sensors

External sensors are those such as microphones, cameras [4], or fixed mounted distance sensors such as infrared (IR) [64]. External sensors are common in tele-monitoring applications, such as Maunder (2013) who attempted to classify tasks through audio signals [40]. This technique detects activities in a static context of a house or an office, which is advantageous in the simplicity of a fixed device, but disadvantageous in being inflexible to the subject's life. Thus, it is very application specific, with aged care monitoring being the most common.

Wearable sensors are advantageous in their flexibility and practicality. Lara (2013) describes the common sensors such as accelerometers, heart rate, electroencephalography (EEG), light and temperature [35]. These wearble sensors are more complicated with small unit profiles, the need for battery power and for data communication. Moreover, the wearable nature introduces a significant amount of noise. The benefits of wearable sensors are in their flexibility, allowing freedom for the user's movements.

2.4.2 Accelerometry

Accelerometry based task analysis utilises sensors to capture the motion of parts of the human body. This has been used to measure human gait[53], position [9] and activities such as running [1]. Accelerometry is advantageous in providing accurate movement information, in a much easier way than camera or IR systems[9]. The user can utilise the task recognition system in any setting as it can easily be integrated into the clothing or accessories. The popular uptake of accelerometry based systems is due to their simplicity, low cost and non-invasiveness.

These systems are very dependent on sensor position, usually requiring reference sensors to isolate individual motions. The literature suggests the hip as the best position, as it is closest to centre of gravity [39]. Some new accelerometers are even being fabricated solely for this field [7]. Accelerometer based systems usually detect gestures, with techniques such as template matching, which compare current data to long term behaviour [42]. Other features utilised are temporal or frequency domain features, the goal being finding distinctive signatures of activities. Detecting the boundaries of events is challenging due to overlap, fuzziness and non-repeatability of actions. Complex algorithms are often required to deal with sensor movement and change [54].

2.4.3 Processing

The analysis can either be statistical features or specific gestures[41]. Statistical features indirectly show longer term patterns over a sampled window. Their advantage is finding the common patterns in the tasks, allowing for some variability between subjects and between repeated events. Conversely, gesture detection is a direct observation of an event, by looking for periodic signals or specific patterns. For example, walking could be detected by the gait signals (gesture - periodic waveform) or indirect statistics of the event (statistical - average mean of acceleration) [32].

The majority of systems can distinguish between about 5-15 activities with reasonable accuracy. Recent efforts have been made to understand more complicated tasks with the temporal links between activities and trying to understand the intent or purpose. To achieve this, dynamic systems have been developed to use multi-modal signals [11]. The extra information allows sensors to accommodate the strengths and weaknesses of each other[45].

The widespread adoption of smart-phones with numerous onboard sensors has initiated a new field of task analysis using much larger samples of data. Hamm (2013) utilises these sensors with delayed self annotation to get some interesting trends [28]. Longer datasets allow patterns to appear in user behaviour and diversions to be detected. The scalability of these studies also gives interesting results on the spread of activities over a much larger demographic.

The focus for task analysis is shifting from the "easy" activities with large movements to more challenging activities. A large majority of people spend significant time mostly stationary (at a desk, on the couch, etc.), making traditional task recognition schemes ineffective. New sources of data from sensors are needed to investigate recognising these tasks. Activity data for these low periods of movement are important in ergonomics and healthy lifestyles [55]. Exposure Variation Analysis (EVA) is used to measure the time and frequency of activity intensity, and could benefit from more detailed activity breakdowns.

2.5 Head Movement

2.5.1 Applications

Most Head Tracking applications are fairly recent, due to the rapid decrease in cost of equipment. Head Mounted Displays (HMD) such as the Occulus Rift[59] and Space Glasses[23] utilise head tracking for virtual reality applications. In this way, they are able to position their user inside a virtual world and their head movement allows exploration. These displays sometimes utilise an IMU to capture rotational movement and avoid long term drift in the sensors. The algorithms used in the head tracking, have high sample rates and smooth error correction to provide a seamless experience to the user.

Head and neck movement is a subject of analysis in terms of biomedical injuries. Grip (2008) explores the development of a biometric model to assist in the treatment of neck pain [24]. They utilised a helical model which allowed for rotation around a movable axis. Some medical research such as Giansanti (2003) [22] have applied IMU sensors to model head movements. However, they struggled to fit and reconstruct the movements from the models due to sensor inaccuracies. Ideally, the sensors are placed as close as possible to the centre of gravity of the limb to capture the true movement. Biomedical literature suggests that the centre of mass for the head/neck system is at eye level towards the back of the skull [3]. Figure 2.1 shows the typical head movements, with average movement [3]: lateral bend ±50 °C; extension 80 °C; flexion 60 °C; rotation ±85 °C.

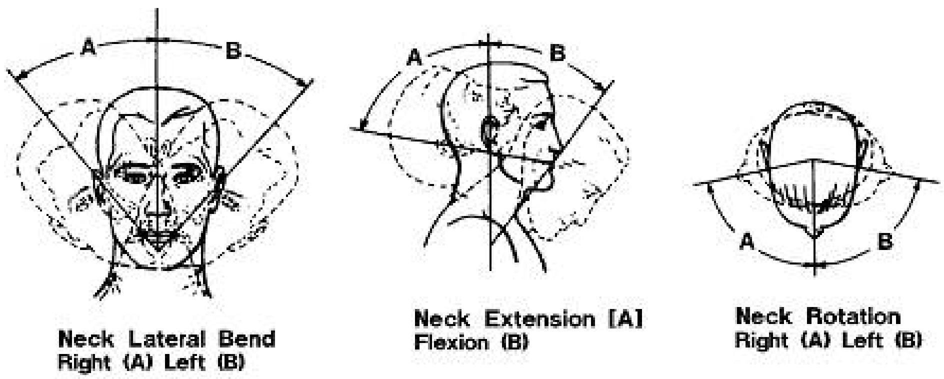


Figure 2.1: Types of Head and Neck Movement [3]

In paraplegic patients, wheelchairs can be configured to be controlled through head movement [61]. In emotion detection, head movement is often used in conjunction with other facial features [18]. Head based accelerometry has also featured in research on concussion injuries in sport [17]. These many and varied applications reflect the versatility and precision of inertial sensors in measuring head movement and social acceptance of the usage of such devices.

2.5.2 Task Analysis With Head Movement

The early work of task analysis involved tracking body parts to distinguish activities. Madabhushi (2000) proposed that tracking the position of the head would be a more distinctive signal than other body parts [37]. These early systems used multiple cameras to track the head position, compared to the rest of the body to distinguish activities.

Tan et al achieved an accuracy of 97% in distinguishing between ten activities using a combination of hidden Markov models and k-Nearest Neighbour [57]. By using a temporal classifier, the images could build a broader picture of the movement, than isolated moments. The system consisted of a series of states, which required assumptions about the movement.

With systems such as Usabinga (2007), the systems are inherently restricted by the need for fixed cameras and large amounts of processing ability [58]. Head movement provides some useful signals which would be best utilised in a multi-modal system. It is potentially suitable to combine head movement with audio and visual systems which log sound and vision from a first person view. For example, head tracking could be added to Kern's (2003) meeting annotation system to provide supplementary information about the wearer's attention and activity. [34].

2.6 Cognitive Load Measurement

2.6.1 Cognitive Load Background

The complementary aspect of human activities to physical movements is the cognitive behaviour. Measuring cognitive load indicates the mental load, which can be used to feedback performance for many dynamic applications. Current measuring methods include EEG [62] and eye blinking [13]. Unfortunately neither of these systems is practical, being intrusive and hard to set up. Cognitive load is important in understanding the intensity of an activity and gaining insight into user performance in a particular activity.

Human gestures have been shown to help offload mental loads, especially when explaining or solving problems. [33]. During human conversations, people naturally pick up on head gestures to gain non-verbal information, such as emotion, condition and thought processes [18]. Most gestures are unconscious movements and patterns that can be used to convey complicated information. During periods of high cognitive load, people can offload mental concentration and thought through subtle movements of their head. Tracking these head movements may be a more practical measurement technique for cognitive load.

2.6.2 Experiment Design

Significant efforts have been made into developing a reliable means of measuring the mental load of an individual while undertaking a activity. The challenge lies in separating the target mental activity from anything else in the subject's brain. For each individual their capacity will be different, requiring calibration for each individual. Also there is a lack of a ground truth of the actual mental load, making training and evaluation systems difficult.

The dual task approach requires a background task to be conducted simultaneously with a foreground activity. The difficulty of the background tasks, (for example counting different frequency tones) is changed to vary the cognitive load. This is used when the load is wanted for a set task, but can lead to mixed results when the participant focuses on only one of the tasks. Other methods involve changing the difficulty of an individual task with cognitive tasks such as mathematical tasks [63] or perceptual tasks such as complex navigation.

To compensate for the various levels of cognitive load in individuals, there are various methods to assess the actual load. Chen [14] suggests the most accurate method is self reporting, but completion time and eye behaviour are other potential indicators. An aspect of task analysis is measuring cognitive load.

2.7 Machine Learning

Machine Learning is the process of teaching a computer to identify patterns to classify pieces of data. The machine is trained with a dataset with the desired classification. From this information the computer works out the underlying trends to make predictions on new data. Figure 2.2 shows the system architecture.

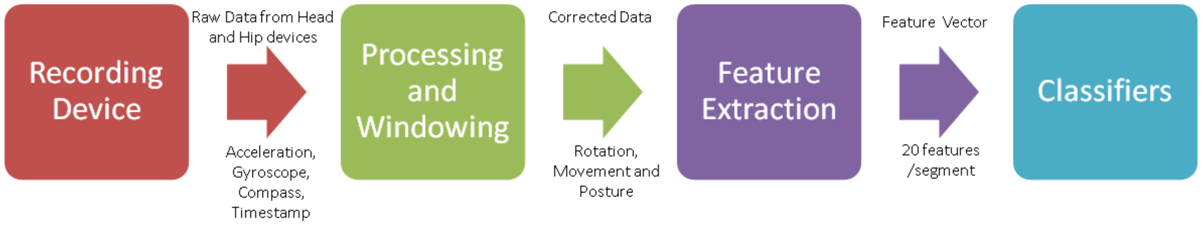


Figure 2.2: System Architecture - The boxes are blocks of the system, the arrows are the signals interfacing each block. Note most of the work is in the machine learning and software of the system.

2.7.1 Pre-Processing

Once the raw data is collected, some processing is required to improve the quality of data [60]. The signals are synchronised if there are multiple sensors, as is often the case with reference sensors. Filtering removes the noise of the signals (especially for gyroscopes) leaving the fundamental motion. Error correction is conducted using the gravity and compass directions in IMUs, to give feedback to the system. More information about accelerometry is discussed in Section 3.1.1.

Inertial signals are quite noisy and need processing to be useful. Firstly MEMS devices sometimes give a spurious value that is saturated at the extreme range of values. This is obviously an outlier and should not skew the results. Hence a non-linear filter has been implemented to extract the median of consecutive samples, removing any of the outliers [39]. Secondly a simple complementary filter is often used to remove the Gaussian noise component.

Sensor Fusion is the process of combining the IMU input signals to improve the accuracy of position and orientation. The major challenge with object tracking is getting an accurate orientation of the object within space. From this, the gravity component can be removed and the position can be computed. Figure 2.3 shows the process of using the signals to compute the desired outcomes in object tracking.

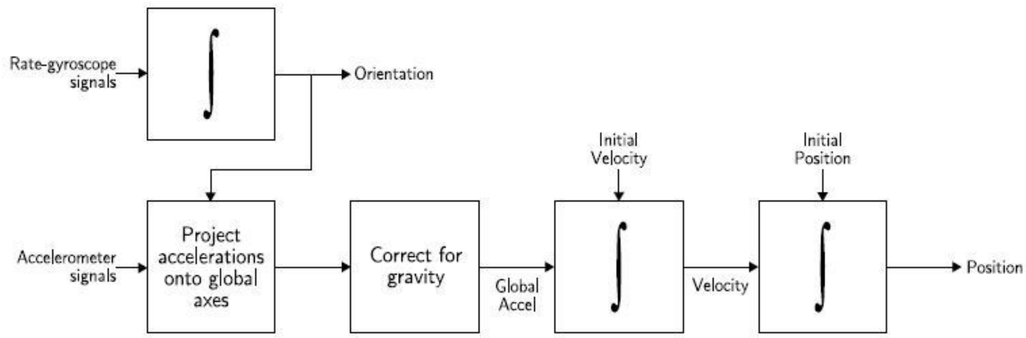


Figure 2.3: Outline of Navigation System [60] - Note the integrations cause errors to build up rapidly.

There are three algorithms for this computation beyond raw gyroscope:

- Complementary Filter - cleans the signal before filtering
- Kalman Filters - predictive filter [21]
- Mahony Method - control loop on drift error [38]

The Mahony method is advantageous because utilises a control loop to correct the orientation by comparing the direction of gravity from the accelerometer with the expected direction from the computed orientation (Equation 2.1). The control loop to reduce the error can be tuned for speed vs accuracy in correcting for the drift. This results in the best system performance. The Kalman filter is popular, but makes significant assumptions about the nature of movement, not relevant in this application:

$$\begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix} [k] = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix} [k-1] + \begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} \Delta t + K_p(a_g \times \theta_g) \tag{2.1}$$

where:

θ: current orientation

k: sample number

t: time

g: gyroscope value

a_g: direction of gravity given by accelerometer

θ_g: direction of gravity given by current orientation

There are a few different numerical ways to represent orientations in space - euler angles, rotation matrices and quaternion. The quaternion number system has been chosen to represent rotations in space due to its computational efficiency. Figure 2.4 and Equation 2.2 shows the representation consisting of a direction e and a rotation θ (Refer to Appendix B for more details and justification of this coordinate system).

$$\left(\cos(\tfrac{\theta}{2}), e_x \sin(\tfrac{\theta}{2}), e_y \sin(\tfrac{\theta}{2}), e_z \sin(\tfrac{\theta}{2})\right) \tag{2.2}$$

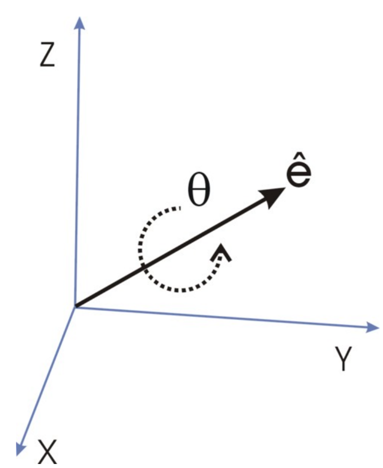


Figure 2.4: Coordinate System - where any object's orientation in space can be described by a direction and a rotation around this direction

2.7.2 Feature Extraction

Feature Extraction is the process of finding the characteristic elements of the signals to distinguish the outcomes. Designing features is an iterative process of processing data to see which has any determining power in the distributions of outcomes.

The samples are divided into windows of data which are analysed. Choosing the size of this window depends on the nature of the signals being analysed. Traditional body worn inertial sensors have looked for periodic signals such as those in gait [32]. The challenge in windowing is in choosing the size of the windows; too large and events get attenuated and blurred, too small and it is not captured. The other influencing factor for window size is whether the activities are temporally static, that is they are similar over time. For some activities this can be true (e.g. reading), but for others it can be very dynamic (e.g. talking). To best enable the capturing of events, multiple sized windows are often utilised, on the scale of 500ms to 5s [31]. To avoid events being split over a consecutive windows, sliding windows will be used where there is some overlap between the frames.

Table 2.2 shows the main types of features used in the literature. The most relevant features are the basic statistics of the signals, such as mean and variance. The patterns are not defined in an individual window, but spread over a series of samples. Frequency information is informative in the nature of the movements and can be utilised in template matching for long term patterns of movement. Wavelet analysis of breaking systems into sub-bands, isolates subtle features that may overlap [56].

Table 2.2: Types of Features

Type	**Technique**	**Description**	**Complexity**
Heuristic	Orientation	raw data	Easy
Time	Statistics	mean, variance, etc	Easy
Time	Gestures	nodding/shaking	Medium
Frequency	Spectrum Centroid and Slope	local spectral shape	Easy
Frequency	Template Matching	similar spectrums	Hard
Time-Frequency	Wavelet Analysis	break into subbands	Hard

Feature Selection is the process of selecting which ones will be used in classification. Similar systems have feature vectors of lengths 10-20. The size of this vector depends on the amount of overlap in the feature space and the number of classes being distinguished. To select these features, there are two core approaches: visual analysis [46] of distributions and numerical analysis [47]. The visual inspection involves plotting histograms or box plots of the feature distributions for each class. Then judgements are made about whether the particular feature is adding knowledge in the class selection. The advantage of this method is the human ability to spot patterns and get a deeper understanding of the phenomenon at play. The alternative is numerical analysis which evaluates features against a metric of information gain or classifier accuracy improvement. The advantage of this method is finding the subset of features that work well together to separate the classes in the feature space for easier classification. A core principle here, is to validate the meaning of the features, that is, understanding the underlying physical phenomena (e.g. walking has a higher acceleration mean than activities at a desk).

2.7.3 Machine Learning (Classifier Training)

The purpose of the machine learning is to map between the features and the class of activities. The complexity of this mapping depends on the input data and the numbers of classes to be isolated. The field is rather experimental, with a large array of techniques, that need to be matched to the data provided. A training set is used to train the classifier by making a model of the system. This model then predicts the outcome of new unseen data.

The techniques can be separated into two broad categories of supervised and unsupervised learning [16]. The separation is based on whether the training sets have been annotated with the desired outcome of the classification. Supervised classifiers have the training outcome marked, for much easier and accurate results. In some cases, unsupervised classifiers are used with no outcome attached, to see the clustering of results or when the data size is large. For this application, supervised is the most useful as the experiment is a controlled setting where the training set can be simply annotated with the outcome.

Table 2.3 outlines the common types of classifiers used in the activity recognition space. In selecting the method that will be used, there are a few factors to consider - distribution of the dataset, number of attributes/inputs, number of classes/outcomes, desired accuracy and complexity. The choice of classifier is very dependent on the nature of the dataset, and in turn, the features selected. The accuracy values quoted in the table are presented to show the variability in results depending on the experimental configuration and the complexity of the classifier.

One technique is the k-Nearest Neighbour (kNN). This technique maps a training point into the multi-dimensional feature space. The k closest neighbours in terms of Euclidean distance are tallied up, with the majority predicting the class of the new point (as shown pictorially in Figure 2.5). In this way the classifier can overcome noise and outliers by choosing a larger value of k. To allow Euclidean distances to be calculated the feature vector needs to be normalised. As a method, the kNN is computationally heavy as it has to compute distances to many points in the training set [49].

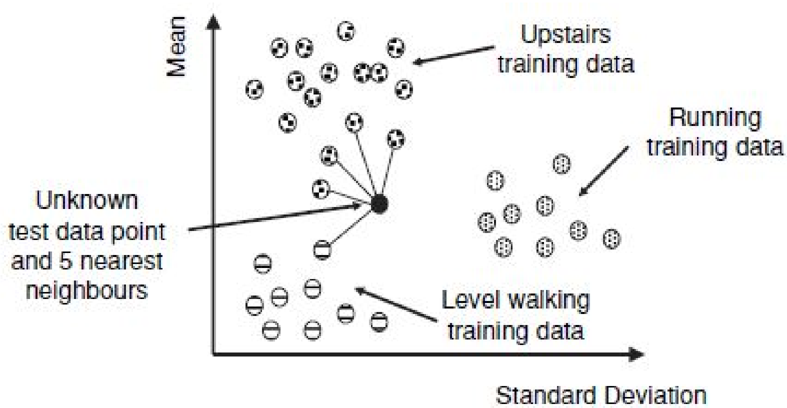


Figure 2.5: kNN - Simple 2D feature space (mean, standard deviation) with three outcomes (walking upstairs, running and level walking) Source:[49]

Support Vector Machines (SVM) are fundamentally a clustering method. The training data forms a multi-dimensional feature space, where each feature is its own axis and has been normalis ed. Boundaries are created between clusters of the data using the bisectors of individual points of data. Once the model is generated, the data is simply analysed to see which region it is in to map it to its class. Therefore, the model will be relatively simple, with no need for memory of the training data. This simple technique works efficiently in well clustered data, with minimal overlap or noise [10].

Table 2.3: Types of Classifiers

Type	**Technique**	**Description**	**Complexity**	**Accuracy**	**References**
Rules	Threshold	simple rule based on one variable	Low		
Trees	Hierarchical	user determined tree of rules	Low	86%;83%	[46];[19]
Trees	Decision Trees (eg J48)	computer determined tree of rules	Low	84%;82%;53%;60%	[10];[46];[51];[19]
Clustering	k-Nearest Neighbour (kNN)	which cluster is it closest to	Medium	83%;50%;90%	[10];[51];[47]
Clustering	Support Vector Machine (SVM)	boundaries between clusters	Medium	63%	[51]
Probability	Naive Bayes Model	conditional gaussian probability	Medium	52%;64%;67%	[10];[51];[36]
Probability	Gaussian Mixture Model (GNN)	Bayes with more complicated probability functions	Medium		
Non-linear	Artificial Neural Network (ANN)	optimise complex mapping from inputs to outputs	High	82%;93%;87%	[46];[47];[19]
Non-linear	Elman Networks	Neural Networks with state memory	High		
Rules	Fuzzy Logic	approximate rules	High		
Temporal	Markov Models	time and state based model	High	47%	[36]
Combined	Many techniques	combine techniques from above	High	90%	[36]
Unsupervised	Many techniques	self organising feature maps	High		

Gaussian Mixture Models (GMM) are a probability based approach. Unlike Naive Bayesian, which assume a simple Gaussian distribution, mixture models include a combination of distributions. They are useful when the outcomes have common features which overlap in the feature space. Probability can be used to determine the more likely outcome from the combination of features. The weighting of components can be modified to achieve the best accuracy [8].

Techniques such as Neural Networks are quite efficient at classifying, but are very complex to design and compute [47]. The benefit of the additional effort for marginal improvement of results is questionable [35]. Surprisingly, the simplest techniques where a rule (or tree of rules) divides attributes are often more effective. The issues with these techniques arise as the situation generalises and the numbe ‘1r of outcomes increases.

Temporal characteristics can be included in the system with long term features or in the classifiers themselves. Some classifiers such as Hidden Markov Models [36], have recent memory of states, allowing for better classification of events which remain static for longer periods. The complexity of these methods often grows too large to handle the current and past information about the system. Depending on the system, this may be a useful consideration for future work.

2.8 Summary

This chapter explored the breadth of techniques used in task recognition on body dynamics. These methods will be extended to using head movement to measure more sedentary activities and cognitive performance. Using intertial movement sensors to measure the head movement, overcomes many of the barriers to use in everyday life. The machine learning concepts will be implemented in the analysis of head movement described in the next chapter.

Chapter 3

Methodology

This chapter outlines the methodology used in constructing, conducting and analysing the results of the experiments. The prototype design and testing are presented. The overall system structure, and experiment design are explained. Pilot Experiments which determined the experimental design are discussed.

3.1 Measurement Device

3.1.1 Theoretical Background

With the onset of smart-phones, Micro Electro-Mechanical System (MEMS) sensors have become widespread in their application and affordable in terms of their costs. The precision of their data has increased, leading to a wide variety of implementations in health, gaming and wearable computing.

To measure movement completely, an Inertial Movement Unit (IMU) is often used (such as in Figure 3.1). Three axis accelerometers can measure linear acceleration and gravity, which can be used to compute the orientation of the sensor with respect to the earth. However, adding a three axis gyroscope allows much better precision by capturing the angular velocity when the unit is rotating on its axis. This unit is very sensitive to errors and can often drift in the output value, so a magnetometer is occasionally added to use a compass to reorientate the device with respect to the magnetic field of the earth.

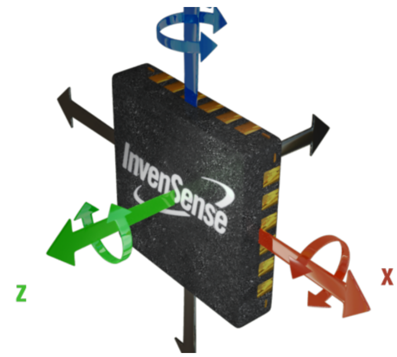


Figure 3.1: Accelerometer and Gyroscope Axes

Understanding the operation of the devices assists in positioning the unit on the body and selecting which features to analyse. Fundamentally, MEMS devices consist of a small mass which is deflected by external accelerations. To achieve the small scale, piezo-resistive materials are used, which have the property of changing the electrical resistance when they are bent or strained. Using piezo-resistive springs, the deformation is converted into an analogue electrical signal for processing. Accelerometers look at the linear deformation of the spring to find the acceleration of the object in that axis. Gyroscopes utilise the rotations of a vibrating mass again with piezo-ceramic materials. The compass or magnetometer utilises magnetic effects such as the Hall effect or Lorentz force to translate into electrical signals.

The signals will capture the following information in their sensors ([39]):

1. Head movement
2. External and body movement
3. Movement from loose attachment of sensors
4. Sensor noise (and magnetic interface for the compass)

The goal of the measurement device is to isolate the first item from the others. Body movement (eg walking) and external movement (eg driving) can be separated by placing a second unit on the hip of the person. The hip is commonly used [39][43] as the optimal position to place the sensor as it is the closest to the centre of mass and is easy to attach to the belt. The sensors are attached firmly to the body to ensure the devices stay in the same position relative to the body. The head sensor is integrated into a cap, and the hip sensor features a sturdy clip. During the experiments the devices are positioned consistently between participants. Finally to counter the noise, multiple channels are combined using Sensor Fusion and features focus on properties which do not drift over time. The extent of the errors are discussed in Section 3.2.2 in more detail.

3.1.2 Measurement Device Design and Construction

A device was designed and constructed to capture head movement using an Inertial Movement Unit (IMU). Figures 3.2 and 3.3 show the prototype device which consists of an IMU, an Arduino microcontroller, bluetooth communications and a battery. The device is able to record linear acceleration, angular velocity and magnetic field strength at a rate of 20Hz. The IMU was chosen for its combination of all three sensors and Analog to Digital Converter (ADC) into the one chip. The Arduino platform was selected for its ease of use and for availability of existing libraries (Details about the Arduino Code are in Appendix C). Bluetooth communications were adopted for low power short range communications with laptops and smart phones. Finally a 400mAhr Lithium Ion polymer is included for a long battery life of about ten hours. The components of this device could easily be designed into a smaller package, such as a pair of sunglasses. More technical details and design decisions in the prototype construction are discussed in Appendix A.

3.2 Sensor Characterisation

As this is a novel area, limited work has been completed in modelling head movement using IMUs. The first step in selecting the feature characteristics of a signal is understanding the behaviour of the signals. Preliminary observations based on the raw signals will influence the pre-processing and sensor fusion methods.

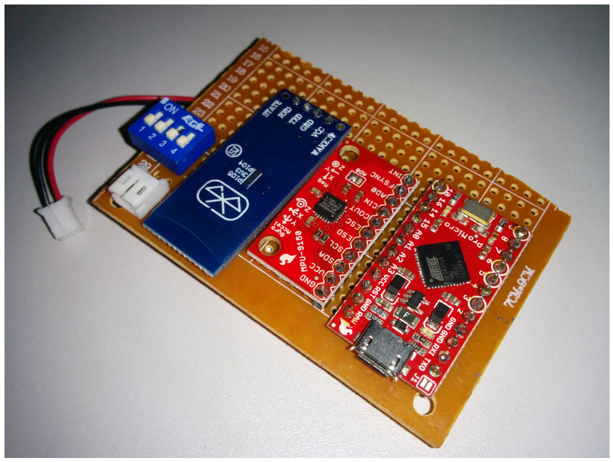


Figure 3.2: Device Prototype (From Left to Right: Bluetooth, IMU, Microcontroller. Battery underneath). Note the practicality of the small profile and wireless communication. Dimensions: 70mm × 40mm × 10mm.

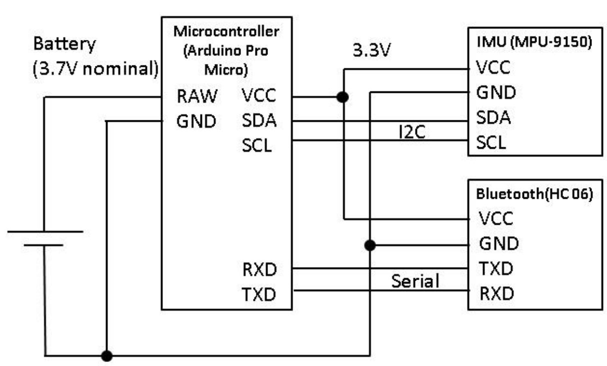


Figure 3.3: Device Schematic

3.2.1 Range of Movement

Firstly the extent of head movement was tested to validate the range and rate of the measurements. As an experiment the head was moved through the extremities and at maximum speed to validate the sensor's ability to capture all data. Figure 3.4 shows the raw data for the maximal possible head movements. Analysis of the figure suggests the fastest possible movement is about 2.5Hz, well below the sampling rate of the system. Moreover, most signals stay within the resolution ranges (except for the maximal values of the y axis of the gyroscope). This validates the hardware prototype in its capacity to capture head movement.

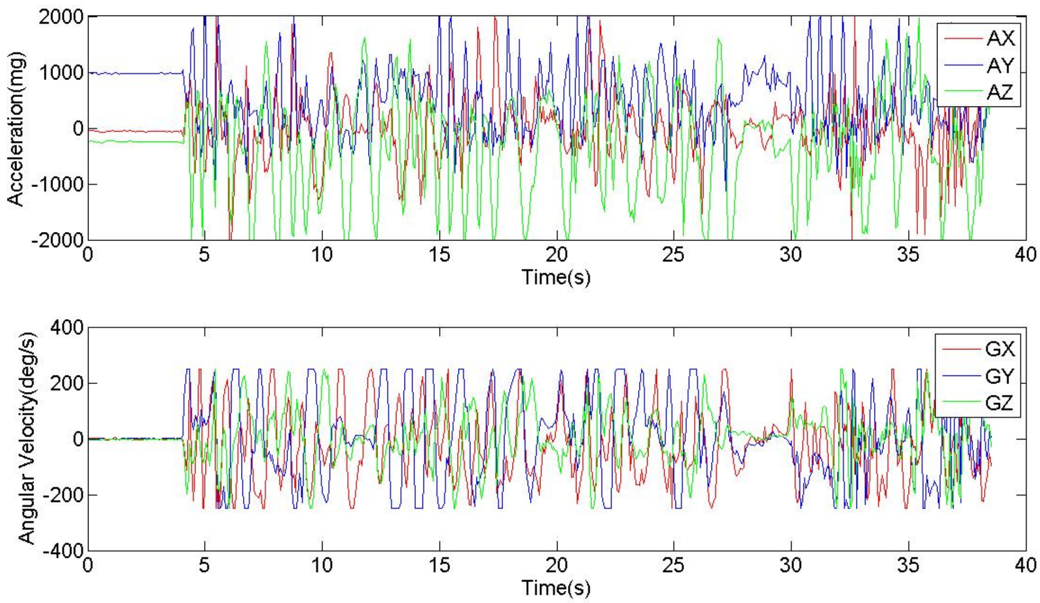


Figure 3.4: Range Of Head Movements - Top: Acceleration, Bottom: Gyroscope - This test involved the greatest extent of head movement possible and seeing if any channels saturated. GY just saturated on the maximal value, but this extent of movement is unlikely in normal movement.

3.2.2 Noise Characterisation

The nature of the inaccuracies in the measurement MEMS sensors is briefly considered with four main sources: ([60]):

1. Bias - This is the output of the sensor when no rotation occurs. For gyroscopes a constant bias, ε, steadily grows over time; $\theta_g(t) = \varepsilon \times t$ With accelerometers, the error in position grows in quadrature over time due to the double integral to obtain position; $\theta_a(t) = \varepsilon \times \frac{t^2}{2}$ Note that if the bias is unstable, a second order random walk will be generated.

2. White Noise - this produces a random walk which grows with the square root of time. Gyroscopes: $\sigma_\theta(t) = \sigma\sqrt{\delta t \times t}$ Accelerometers: $\sigma_\theta(t) = \sigma \times t^{\frac{3}{2}} \times \sqrt{\frac{\delta t}{3}}$

3. Temperature Effects - onboard temperature measurement to compensate

4. Calibration - errors in the scaling and linear properties

For the MEMS gyroscope being used, the bias and the random walk due to noise are the most important errors. As the angular orientation is the integral of the errors, the output error can grow rapidly. That is why the gyroscopes are often corrected with feedback mechanisms from other sensors. Also the features of head movement should be concerned with short term movement, rather than the absolute position over time to avoid the influence of errors.

Noise sources can be estimated using the Allan Variance. This technique is a means of isolating and estimating the error sources by averaging signals over varying amounts of time. The experiment involves leaving the device recording in a still environment for about one hour and then analysing the results. Figure 3.5 shows the log-log plot of the Allan Deviation over a wide range of time windows. The different patterns of curves indicate a different source of error, with m being the gradient of the slope.

The calculation for Allan Deviation is shown in Equation 3.1, where the variance of windows are computed for different windows of length t (where $\bar{a}_i(t)$ is the average value

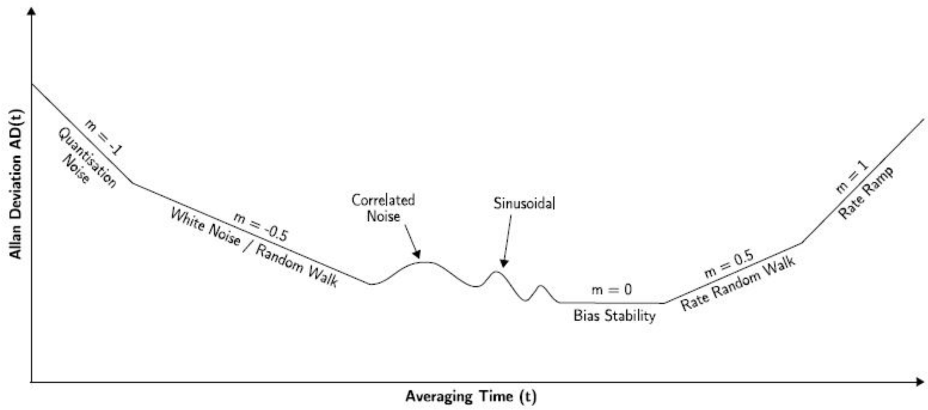


Figure 3.5: Allan Deviation Log-Log Plot - The m values show sections of the plot with that gradient which corresponds to particular noise features [2][60]

for each window). Section 3.2 details the results of this experiment using this method.

$$AD(t) = \sqrt{AVAR(t)} = \sqrt{\frac{1}{2(N-1)} \sum_i (\bar{a}_{i+1}(t) - \bar{a}_i(t))^2} \tag{3.1}$$

To characterise the devices, the noise sources are quantified. Before quantification, the offset bias of the gyroscopes has been corrected on-board in the microcontroller. Figure 3.6 shows the Allan Deviation for the Gyroscope Axes. For both plots the linear slope on the log-log plot corresponds to angle random walk, where the random walk per second is the value at a window of one second. The flat section of the plot is the bias instability quantified by the minimum value of the curve. Table 3.1 highlights the results of estimating the noise parameters of the sensors. It is observed that the gyroscope and compass are more vulnerable to drift than the accelerometer.

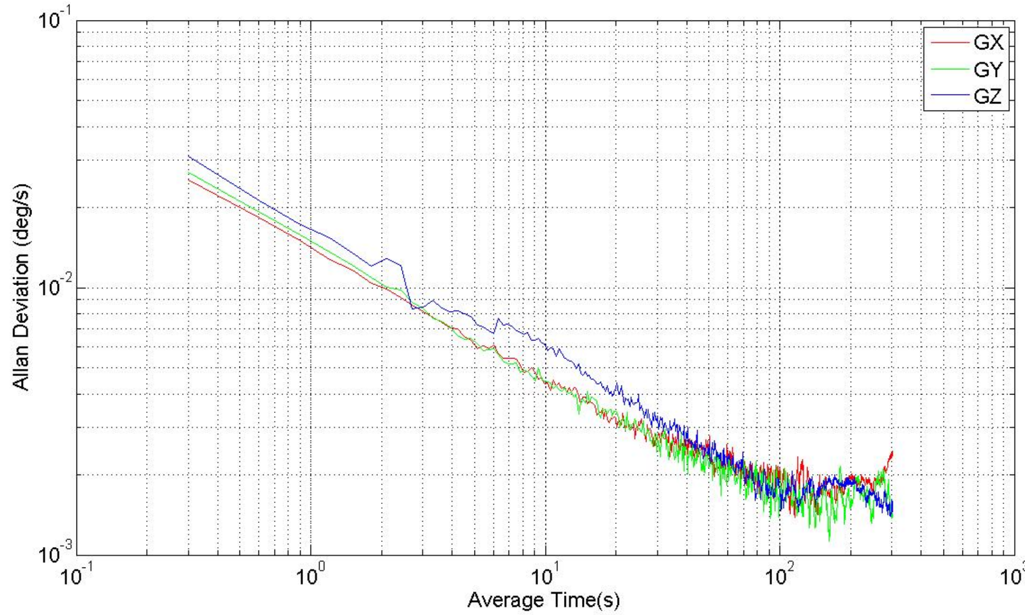


Figure 3.6: Allan Deviation for the Gyroscope - A technique of measuring the errors in the measurements. The left hand slope relates to the angle random walk, and the flat right hand section details the bias instability.

Table 3.1: Noise Estimates from Allan Deviation - Note that the errors are much larger for the gyroscope than the accelerometer. Thus gyroscopes can only be used on a shorter time frame.

Sensor	Axis	Bias Instability	Angle Random Walk
Accelerometer	X Axis	$0.0096m/s^2$	$0.00006m/s^2/\sqrt{s}$
Accelerometer	Y Axis	$0.0005m/s^2$	$0.00004m/s^2/\sqrt{s}$
Accelerometer	Z Axis	$0.008m/s^2$	$0.0002m/s^2/\sqrt{s}$
Gyroscope	X Axis	$0.08\deg/s$	$0.005\deg/\sqrt{s}$
Gyroscope	Y Axis	$0.06\deg/s$	$0.002\deg/\sqrt{s}$
Gyroscope	Z Axis	$0.05\deg/s$	$0.005\deg/\sqrt{s}$
Compass	X Axis	$3.6uT$	$0.15uT/\sqrt{s}$
Compass	Y Axis	$3.6uT$	$0.1uT/\sqrt{s}$
Compass	Z Axis	$3.6uT$	$0.7uT/\sqrt{s}$

3.2.3 Long Term Frequency Profile

The next phase was to see the long term characteristics, particularly the frequency profile. The device was worn for a long period of about one hour (The raw data is shown in Figure 3.7). The data was then processed to filter the signal and remove the gravity component. Figure 3.8 shows the power spectrum information of the long term signals. For the accelerometers most of the signal power is in the low frequency and is spread evenly amongst the horizontal plane of movement. The vertical acceleration (Blue line in Figure 3.8) is smaller, as typically the head stays mostly at a steady height, just moving sideways. The gyroscope signals are spread across a wider range of frequencies, noting there is some noise in the low frequency. The blue signal, representing neck rotations as in Figure 2.1, has slightly more power than other rotations.

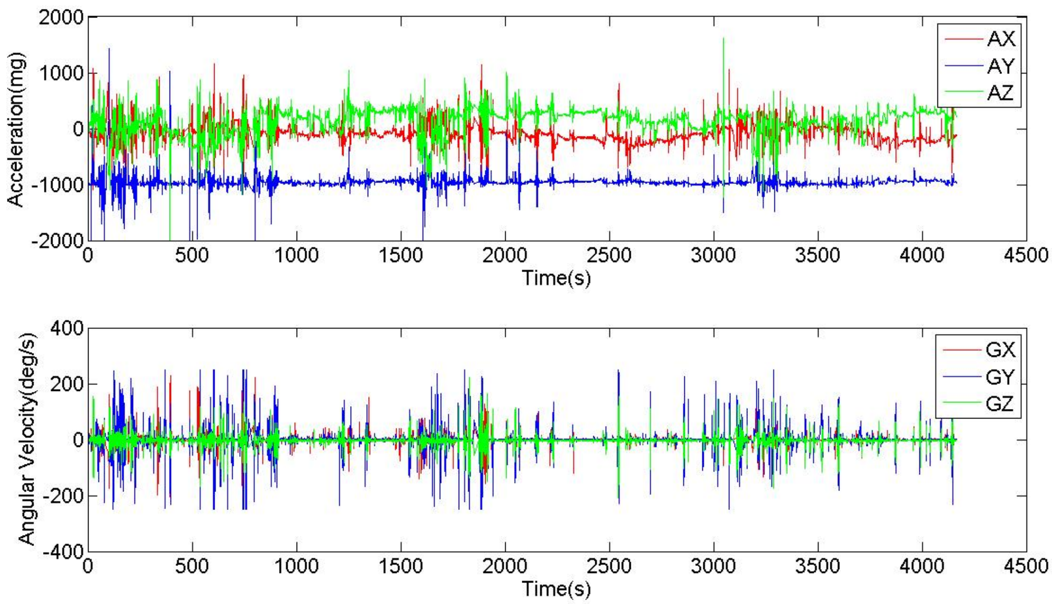


Figure 3.7: Long Term Plot - Raw Signals - Top: Accelerometer, Bottom: Gyroscope. Head Movement of a subject over one hour of daily life.

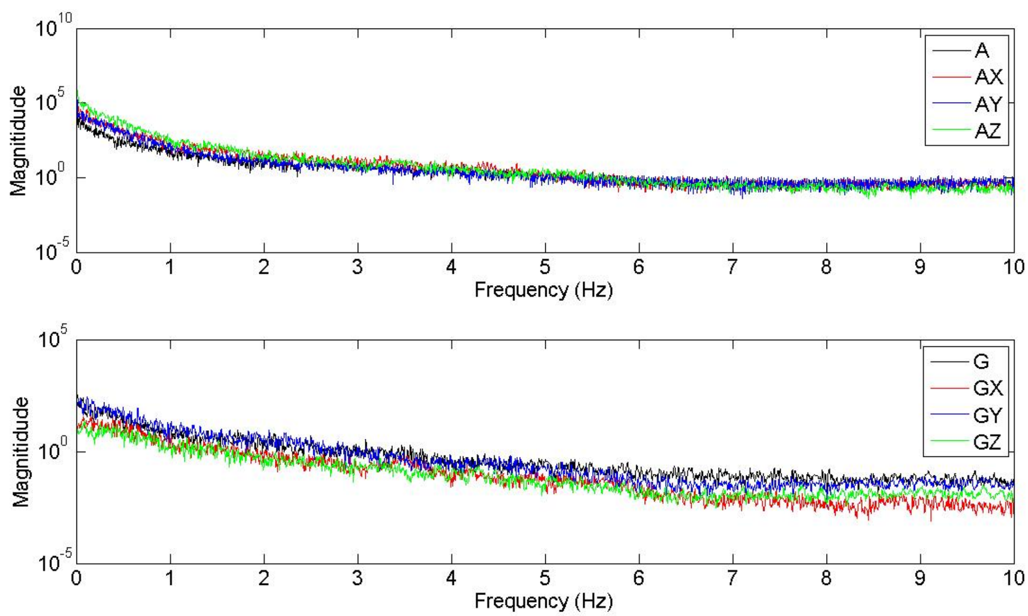


Figure 3.8: Long Term Plot - Power Spectrum - Top: Accelerometer, Bottom: Gyroscope. Power Spectrum of daily life. The vertical acceleration (blue) has the lowest energy per axis and the neck rotation (blue) has the highest energy for the rotations.

Note most of the energy is under 1Hz, which will used later in the window size selection

3.3 System Architecture

The overall system records and processes head movement to make predictions. The following section details the specifics of each section as displayed in Figure 3.9. More details about the programming code written can be found in Appendix C (Arduino) and Appendix D (Matlab).

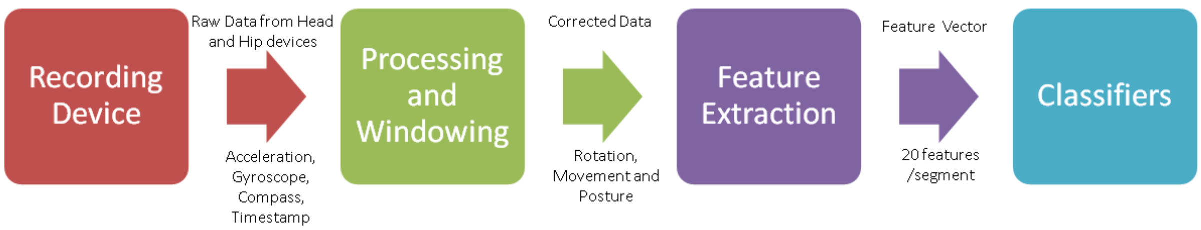


Figure 3.9: System Architecture

3.3.1 Data Collection

The data from each device is transmitted over Bluetooth to the USB Bluetooth serial converter on the laptop. The data from each device consists of the nine 16-bit data channels and a time-stamp. The devices are configured to relay data at a rate of 20Hz, well above the level needed for this analysis. Within Matlab the values are logged to a variable file.

With two devices (head and hip), the devices individually transmit data back to the laptop. They are synchronised at the start of each stage by getting the person to stand and sit. Both devices have their timestamp corrected to be the time from this event. The scene camera can also be used to annotate the synchronisation if issues occur.

3.3.2 Data Pre-Processing

From the raw signals, the data is processed to make it useful for analysis. The first step is to scale the 16 bit numbers to their true range: Accelerometer - $\pm 2g$; Gyroscope -

$\pm 250 \deg /sec$; Compass - 2000 μ T.

The filtering is in two stages with a median filter removing extreme outliers (that often happen with accelerometers) and a simple digital filter to smooth out the noise [39]. A set of three samples was used for the median, as the saturating outliers only occurred for one sample.[39]. A digital filter was used to remove the noise discussed in Section 3.2.2. The digital filter works by weighting the new sample value against the previous total, as shown in the equation below, where x is input signal and y is the output signal.

$$y_k = (1 - \alpha) y_{k-1} + \alpha x_k \tag{3.2}$$

A value of 0.1 for α which acts as a low pass filter the signal. This value was chosen after balancing removal of noise and resolution in movements.

To enable orientation correction in the sensor fusion of the next step, the gyroscope data has its mean removed. This has an inherent assumption that the net head rotational position is not far from the starting location. For the experiment where most of the activities are centred on the computer, this assumption is acceptable. The accelerometer and magnetic directions are needed for direction, so they will be zero-meaned after the sensor fusion.

3.3.3 Sensor Fusion

Sensor Fusion is the process of computing the relative orientation of the device with respect to the earth. The system uses gyroscopic rotation compensated by gravity and magnetic directions due to drift errors. After analysis of methods discussed in Section 2.7.1, the Mahony algorithm was chosen which simply uses a feedback loop with integral and proportional error terms. From the orientation the actual velocities and positions can be computed.

To improve the computational efficiency of the method quaternion number systems were used to model the rotation in space. An experiment was conducted to quantify the improvement compared to the basic method of gyroscope integration. Figures 3.10, 3.11 and 3.12 show the improvement of sensor fusion across a series of ninety degree rotations. The gravity correction of the Mahony algorithm, gets the magnitude of the rotations more accurate, and is more valid over time.

3.3.4 Windowing

The next stage of the system was to segment the data and extract statistical properties from it. The concept of windowing allows small events, such as yawning to be separated from the bulk of the data. By using enough data segments, the isolated events do not affect the overall classification. Without any periodic properties in the data, the size of windows depend on the size of movement events. Experiments are conducted below to decide the ideal size of window and amount of overlap. Initial estimates from the power spectrum in Figure 3.8 suggest most energy is in movements less than 1Hz, so the window will be close to that size. In case of events being split over two windows, an overlapping factor will be used to capture events on the edge in the next window. Each window forms a segment of the feature vector used in classification.

3.3.5 Feature Extraction

Types of Features

The processing of the data cleans up the signal, allowing easier analysis. All of these signals will be analysed to choose the ones which provide deterministic power over the classification data. The processing signals take a few different forms as seen in Figure 2.3 from the previous chapter. The important signal forms are the rotational position, rotational velocity, linear acceleration and linear jerk (derivative of acceleration).

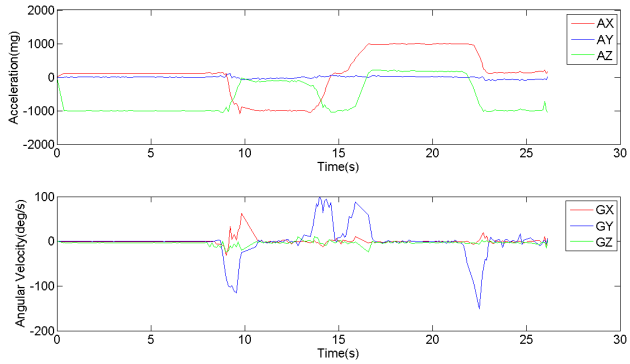


Figure 3.10: Sensor Fusion - Raw Data. Device is rotated through a series of ninty degree rotations

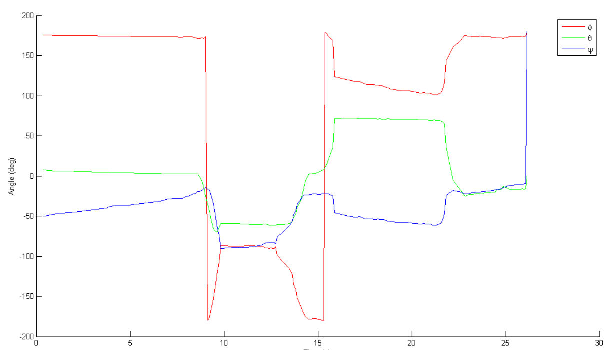


Figure 3.11: Sensor Fusion - No Correction. Euler Angles show the rotations, but not well.

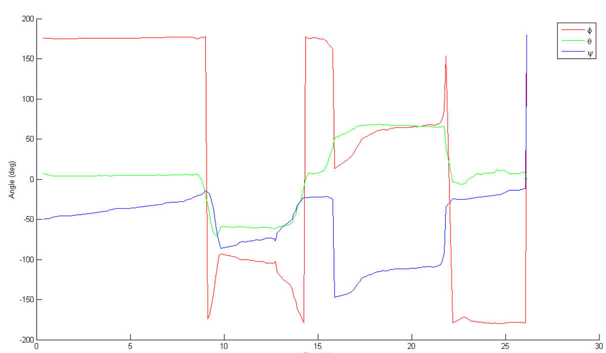


Figure 3.12: Sensor Fusion - Gravity Correction. Much better performance, as magnitude of shifts are closer to the ninety degree rotations

From these signals, defining features quantify the signal's behaviour within a window. Statistical features are more powerful in providing numerical quantities to compare various events. Features range from distribution statistics such as variance to correlations between data streams to heuristic features. Many features are built up within a window, to create a feature vector for that time segment. As there is minimal periodic content, frequency based approaches are less advantageous due to the computational complexity. Power Spectrums, Fourier Transforms and Subbands have been calculated, but provided little advantage over time based statistics.

Table 3.2 lists the common time based statistical features and their respective formulas (Recall time statistics are one type of features from Table 2.2 in Chapter 2). The input signals are corrected rotational velocity, linear acceleration and orientation. For simplicity, a single window size is adopted and optimised for classifier performance. These parameters of the signal, window and feature are iteratively processed to determine an effective basis for the classifier. Introduction of temporal features could be included by doing statistics on the features to include previous data in the current window or for the segments to cover longer time periods.

Feature Selection

The selection process for choosing features consists of two stages: numerical analysis in Weka [25] and visual analysis of class distributions.

Numeric feature selection can be done by analysing the distribution of feature values compared to the outcome on a training set. This is to be conducted in Weka where it processes feature vectors to optimise the subset of vectors [25]. The desired features of the subset are discernibility of the feature in predicting the class and the redundancy compared to the rest of the subset. Weka evaluates the features using a correlation algorithm of the feature values and class outcomes [26]. To find the best subset it utilises a greedy hill-climbing search method which assesses if added nodes improve the subset. This method returns the subset of features, best for that subset. This method was used

Table 3.2: Some Common Statistical Features And Their Formulas

Type	Description	Equation	References
Mean	average value	$\bar{x} = \frac{1}{N}\sum_{k=1}^{N} x(k)$	[46]
Standard Deviation	spread	$\sigma_x = \frac{\sum_{k=1}^{N}(x(k)-\bar{x})}{N-1}$	[51]
Skewness	asymmetry of distribution	$s_x = \frac{1}{N\sigma_x^3}\sum_{k=1}^{N}(x(k)-\bar{x})^3$	[9]
Kurtosis	clustering in centre of distribution	$k_x = \frac{1}{N\sigma_x^4}\sum_{k=1}^{N}(x(k)-\bar{x})^4$	[9]
Percentiles	points on distribution	$Q_{x\%} = \frac{x}{100}(n+1)th$ sample	[46]
Interquartile Range	spread	$iqr_x = Q_{75\%} - Q_{25\%}$	[46]
Correlation	relationship between data	$corr(X,Y) = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y}$	[31]

to narrow down the range of features from roughly 100 to about 20.

The next stage involves plotting histograms of the distributions for each class for each feature. Visual Analysis can assess patterns, to see if the feature is viable. The desired properties are a Gaussian distribution, consistency in trend between subjects and difference between the classes. Scatter plots of two features is another tool to visualise if the features are useful. The Gaussian distributions are required for the Bayesian classifier and Z-score normalisation of features. The normalisation is needed to cover the classifiers which use Euclidean distance between vectors.

3.3.6 Classifiers

Classifiers are the algorithms which map feature vectors to predicted classes after receiving some training data. The seven relevant methods are shown in Table 3.3 (Refer to Table 2.3 in Chapter 2 for a list of types of classifiers). These methods were selected through analysis of common successful methods in literature and simplistic testing on initial datasets. A range of methods was selected which take different approaches and are appropriate for the classification problem construction. The majority of the analysis was done on the Weka Machine Learning platform [25]. To validate their methods and understand how these algorithms work, they were implemented in Matlab. The results were compared to check the operation of each method. While most methods were not replicated exactly, the majority were very close.

ZeroR is the modal class method. It is calculated by evaluating the most common class in all the training set:

$$c_j = mode(c_i) \forall i \tag{3.3}$$

where:

c_j: class of instance j

i: instances in test set

Table 3.3: Machine Learning Classifiers Methods - Note the complexity of the methods increases further down the table. The desired method has the best performance, without being too computationally intensive.

Classifier	Description	Reference
ZeroR	Most common class	
OneR	Rules on the most deterministic attribute	[30]
Naive Bayes	Bayesian Probabilities	[16]
J48	Decision Tree	[50]
Support Vector Machine (SVM)	Make hyperplanes to separate clusters of data	[48]
k-Nearest Neighbours (kNN)	Match it to training data	[5]

OneR is a one attribute algorithm classifying by dividing instances into blocks. The instances are quantised into intervals of values. For each value interval in each attribute, the number of instances are tallied. The modal class is estimated for each interval to form the hypothesis. Each attribute is evaluated using these hypotheses to find the attribute with the best performance.

$$c_{a,v} = c|maximise(count(a,v,c))\forall(a,v) \quad (3.4)$$

$$a_{best} = maximise(\frac{|\sum_i c_i == c_{a,v}|}{|c_i|})\forall(a) \quad (3.5)$$

where:

a: attribute/feature

v: quantised value interval

c: class

a_{best}: chosen attribute

Naive Bayes is method of fitting test instances to the most likely class distributions for all the features. Assuming Gaussian distributions for all the classes, the mean and standard deviation properties are extracted from each feature for each class. New instances are assessed as the highest probability of the product of the new value fitting that class's distribution:

$$maximise|p(c,j) = \prod_{a} p(a,c|v_j) \tag{3.6}$$

where:
$p(c,j)$: probability of instance j belonging to class c
$p(a,c|v_j)$: probability of value j for attribute a belonging to class c

The **J48** method is a decision tree which uses entropy selection for each node, which is a feature. It operates recursively, choosing the feature with the best information gain to further segment the data down the tree. The information gain is selected as the gain in entropy, that is differences in this feature cause significant gains in the output. The data is segmented in two subsets at each level and the process is repeated until the leaves of the tree are sufficiently small.

$$\text{Information Gain}(a) \quad = \quad \text{Entropy}(S) - \sum_{v} \frac{|S_v|}{|S|}\text{Entropy}(s_v) \tag{3.7}$$

where:
S: is the set of output classses
S_v: is the subset of output classes separated by the values of attribute a

The **Support Vector Machine** (SVM) attempts to separate the feature space into hyper-spaces and assign them to a class. The algorithm tries to find the hyperplane of dimension p-1 (where the feature space is dimension p), with the greatest distance between classes. The algorithm acts to define clustering into region of classes. As the model only requires the boundaries and not the training data, it can run efficiently, but requires well spaced data. The core idea is expressed below:

$$maximise(d(i,j)) = \frac{2}{|w|}|w.x - b = \pm 1 \tag{3.8}$$

where:

$w.x - b = \pm 1$: equation for hyperplanes

d(i,j): distance between points in feature space

$|w|$: normal vector to hyperplane

i: one class index

j: another class index

The **k-Nearest Neighbour** method chooses the class by finding nodes with the minimum distance in the feature space. The test instance is assigned an output class matching to the most common class of its neighbours. For one neighbour:

$$c_i = c_j|minimum(d(i,j)) \tag{3.9}$$

where:

i: test index

j: training index

ZeroR and oneR served as performance benchmarks for the sophisticated classifiers. Within each method, there are often parameters which need to be tuned to optimise performance and ensure no over-fitting. Some methods are computationally heavy, so preliminary analysis may focus study on a few in particular. For a real-time classifier, an optimised method would be required to allow quick processing while not sacrificing performance. Over-fitting is a potential problem, when the learning characterises the noise in the signals rather than the signals themselves.

3.4 Classification Analysis (Metrics)

Data structure and classification assessment methods are important in ensuring valid results. The way data is arranged between developing, training and testing the models is significant in acheieving the best performance, whilst not over-fitting to the specific dataset. It is imperative that it is structured in a way that verification of the system is done by data independent of the system design.

3.4.1 Data Organisation

There are three main groups of data in the experiment: training, deployment and evaluation data. Evaluation data is used for verification once the system is complete and is not analysed prior to this. Typically about 10% of the overall data is used for this purpose. In this case, two people were the evaluation data, and not used in the development of models. Moreover, a small proportion of each participant's data was randomly removed to provide evaluation data within the person's data set.

The rest of the data is used for the development of the system. Mostly cross-validation is used to ensure the model is able to best showcase the information within the dataset. This uses a single dataset to train and test the model by cycling through different combinations of training and testing roles. This deployment data usually consists of 90%, with ten folds of cross validation. Ten folds means there are ten rotations with different combinations of training and test data. Stratified Cross Validation can be achieved by balancing the amount of data in each class and from each participant. When comparing statistics across experiments it is more equitable to have a level playing field with balanced data, however the final result should use as much data as possible. For data rich training, cross validation may over-fit the results, and percentage splits will be used instead. This reflects an application based approach that uses a small amount of calibration data to classify.

The deployment and evaluation results are presented for the different classification algorithms. It is likely the evaluation results will be somewhat worse than the deployment ones, but it may not be statistically significant. After this run with all the participants, there is also subject variability classifications with averages of each participant's results when trained from their own data. In the pilot combinations a variety of testing arrangements are explored to see the nature of subject variability. This should provide insight into whether these patterns are consistent across individuals and how much calibration data is needed per person.

In some preliminary experiments it was noted the ordering of tasks affected the results, especially features which accrued errors over time. To compensate for this factor, the tasks have been randomised in order for each participant.

3.4.2 Classifier Metrics

Metrics are numbers used to interpret the results of classifications. These statistics are used to analyse the performance beyond a simple accuracy. There are important features not captured in accuracy, basically about the distribution of classifications results and an application's desired properties.

Table 3.4 shows the different metrics. The kappa statistic is used for unbalanced datasets, to give a measure of the level of accuracy above guessing. The confusion matrix is the most useful metric showing the distribution of predictions and actual classes, visualising the properties of the classifier. The majority of the other metrics are computed for each class, on a within the class or not within the class basis. They provide insight into class specific features and distributions (eg false positives, etc).

Receiver Operator Characteristic (ROC) curves are a plot of the true positive rate against the false positive rate. The area under the curve is the probability of a randomly chosen instance of class A ranking higher than a randomly chosen instance of class B [29]. Thus the metric quantifies the performance of a classifier in a two class problem in a threshold independent manner, making it a more relevant measure than accuracy. The ROC Area can be extended to multiple classes by finding the ROC Area for each class against all

Table 3.4: Metrics - The ones used most frequently in this study are Accuracy, ROC Area, and the Confusion Matrix

Metric	**Description**	**Formula**
Accuracy	Proportion of Correct Classes	$\frac{Correct}{instances}$
Kappa Statistic	Accuracy beyond guessing	$\frac{accuracy - guessing}{1 - guessing}$
Confusion Matrix	Distribution of actual and predicted classes	$\mathbf{C} = \sum_{p=predict} \sum_{a=actual} \lvert i_{pa} \rvert$
Recall/True Postive Rate	Accuracy within predicted class	$\frac{ClassCorrect}{ClassPredictions}$
False Postive Rate	Inaccuracy Rate	$\frac{ClassIncorrect}{OtherClassPredictions}$
Precision	Accuracy with actual Class	$\frac{ClassCorrect}{ClassActual}$
F-Measure	Overall Measure	$\frac{2 \times Precision \times Recall}{Precision + Recall}$
ROC Area Under Curve	Probability ranking	$\int_{-\infty}^{\infty} TPR(T)FPR'(T)dT$
Friedman Test	Order, treatments,etc	

other classes, then taking the weighted average. A value of 0.5 is random guesses, with 1 being perfect discrimination.

The literature varies markedly in the metrics used to assess classifier performance. For this study the Accuracy, Confusion Matrix and weighted ROC Area are the core measures. For comparisons with similar studies other metrics such as f-measures, false postive/false negative rates are also considered. The majority of this information is expressed in the confusion matrix and the distribution of predicted outcomes against the ground truth.

3.5 Dataset Design And Implementation

Due to the lack of an existing dataset, an experiment was designed and conducted. The following sections detail the design and format of the process. The project received Ethics Approval by the university panel as part of a larger series of experiments by the research group. More details about the Ethics approval are in Appendix F. Moreover, Appendix G documents the risk assessment which was important in experiment design in ensuring zero harm to the participants.

The experiment aims to explore the potential of head tracking in a range of issues surrounding task analysis. The experiment was structured in a staggered manner for each problem to see what range of applications is possible.

3.5.1 Task Design

The experimental design was created through analysis of the literature, and iterations of testing and analysis. The sub-problems of task analysis were extracted and tasks designed to explore particular variables.

The sample size of participants was chosen at twenty, large enough to have decent statistical validation, but manageable logistically. The demographic of participants, being mostly volunteer undergraduate engineers is not ideal, but should not skew the numerical results much.

As seen in the literature, the main experiment consisted of directed activities within a fixed time frame. This is to simplify the experiment and later analysis. Most activities in a recreation, employment or education context consist of a few basic subtasks. Daily life consists of activities which have layers, actions and purposes. To simplify things, the core tasks were isolated within the framework of a desk environment. The activities with respect to the mental characteristics are shown in Table 3.5. Note the ranking is a subjective assessment by the author. The tasks were chosen to ensure a range of these factors.

Arithmetic task was chosen to simulate cognitive load, as it is a more effective construction than dual task approaches where a background task is changed in difficulty.

Table 3.5: Activity Tasks - Taxonomy Ranking - These ranking were subjectively assigned by the author. Note the activities try to cover a range of these factors.

Task(L=Low/H=High)	**Cognitive**	**Motor**	**Communicative**	**Perceptual**
Writing	L	H	H	L
Reading	H	L	L	L
Walking	L	H	L	L
Watching	L	L	L	L
Rubik's Cube	L	H	L	H
Conversation	L	L	H	L
Arithmetic	H	L	L	L

3.5.2 Experiment Protocol

The experiment was conducted on twenty volunteers, with each participant being fitted with the two sensors, one on the head and another on the hip. The hip sensor acts as a reference device to extract true head movement. The participants were instructed through four different tasks. The majority of the tasks were done in front of a computer and a desk. This was deliberate in trying to simulate the standard sedentary environment for people. See Figure 3.13 for two photos of a participant.

Figure 3.13: Experiment Setup - Participant at computer with head device on cap and hip device on belt

Task 1 was a calibration task to get the participants comfortable with the user interface and devices. They were instructed to move through a range of head movements and a quick walk around the room. The task acted to distract the participants from the fact their head movement was being monitored. Awareness of what is being recorded causes consciousness of actions often leading to less natural movements.

Task 2 involved working through six different activities for two minutes each (randomised in order). There was a buzzer directing transitions to next activity. To ensure smooth transitions, the participants did a practice run beforehand to become aware of the activities and the ordering. The activities include:

1. Watching - video about solar cars
2. Reading - reading the Australian constitution on computer
3. Writing - writing anything on a sheet of paper
4. Rubik's Cube - attempt to solve a Rubik's Cube

5. Walking - walking around the room

6. Talking - discussion about wearable computing as part of survey.

.

Note, a seventh activity, arithmetic, was also added to analysis using data from Task 3. Walking was chosen as the only non desk based activity, and is deliberately included for the easiest classification. Writing, Rubik's Cube and talking all have distinctive patterns of head movement, and breakdown many common sedentary activities. To test the ability of the system, three hard classes are included of reading, watching and arithmetic, all of which are similar in only having relatively small amounts of movements.

Task 3 consisted of completing arithmetic tasks on the computer. Three levels were chosen from seven levels created by Chen [14]. Level 1 consisted of adding a one digit number to a two digit number with no carry (eg 14 + 3). Level 4 contained the addition of two digit numbers with one carry (eg 56 + 25). Finally Level 7 offered many carries with the addition of two three digit numbers (eg 528 + 693). To keep the subjects focused still on the screen and not on the keyboard, a number pad was made part of the Matlab Graphical User Interface (GUI). To compound the cognitive load the subjects had to enter the most significant number first, forcing special techniques to complete the problems. To validate the load, the following metrics were recorded for the questions: score, average time and a self reported scale. The self reported scale rated the difficulty of each question from 1 being very easy to 10 being very difficult. Additionally they were quizzed to explain their strategies on attempting the harder questions, to qualitatively understand how they cognitively completed the task.

Task 4 was an open ended recording of the participants to attempt to simulate their typical real world behaviour. The purpose of this task is to attempt to use the stored data to predict activities in a real world environment. The participants were recorded for 10 minutes and their activities annotated. They were asked where possible to include some activities within the trained set and some outside (eg phone, internet browsing, homework, etc)

To complement the numerical data, the participants were surveyed on a range of topics around the context of the research to validate the assumptions and predictions of technology. Throughout the experiment they were recorded using the webcam of the computer and the experimenter made observational notes about each subject.

3.6 Pilot Testing

A number of experiments were conducted to understand the underlying patterns and refine the system design. The list of variables to optimise were:

1. Amount of data per subject
2. Window size (size, overlap)
3. Feature selection
4. Classifier algorithm parameters
5. Handling of subject variability

It is assumed that these variables are independent in terms of performance. Whilst this is not absolutely correct, it is an acceptable approximation when tuning variables locally on a small scale. These experiments were conducted on a preliminary dataset consisting of four persons.

3.6.1 Learning Curve

The amount of data needed to train models dictates the structure of the experiment. The performance of classifiers generally improves with more training data, and then flattens out as it reaches critical size. An experiment was conducted to observe the performance of classifiers with different amounts of training data to see how much was needed. The learning curve is a plot of classifier performance against size of training data.

Figure 3.14 shows the learning curve for a preliminary data set of five activities. The results depend on the classifier, with the simplistic classifiers (ZeroR, SMO and Bayes) not improving with more training data for each model. Conversely, the more complicated systems of IB1 and J48 improved with more data, as they found the general patterns and became more resistant to noise. After using approximately 40% of the training data, the improvement was minimal suggesting a baseline of 120 seconds of data per class was needed to make a good judgements.

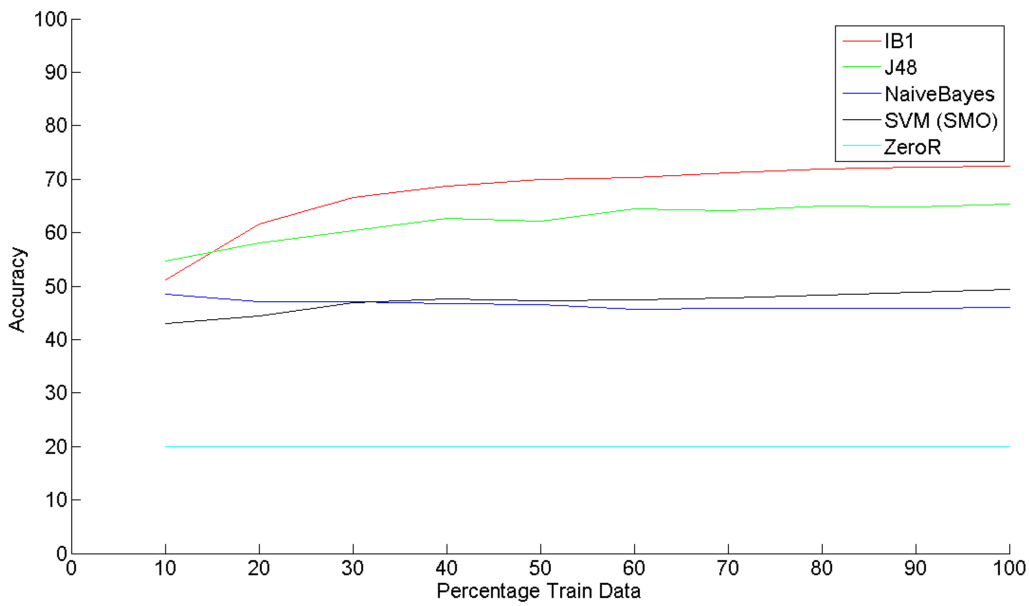


Figure 3.14: Learning Curve - Accuracy of Classifiers against percentage of data used to train the model. Forty Percent is the critical amount of training data.

3.6.2 Windowing

Window Size

The number of samples in the window was optimised for best classifier performance. The experiment involved fixing the number of time segments in the feature vector and varying the number of samples in each segment. This test was conducted on a preliminary dataset with a subset of five activities, a simple feature vector and utilised the k-Nearest Neighbour and J48 Decision Tree Algorithms. The system performance was evaluated using

Accuracy and ROC Area. The best performance was around ten samples per segment which corresponds to about 500ms.

Window Overlap

As the movement events are distributed randomly with respect to the windows, often one event can be split over two different windows. Hence overlapping windows are utilised to cover this circumstance as the event on the edge will be in the overlapped version. The most common value of overlap from the literature is 50%, meaning half the old window is reused in the next one.

To optimise the overlap, the best window size of 10 samples from the previous experiment was taken and the overlap was varied. The amount of overlap affect classifiers in different ways, particularly those which use neighbours as training points. Overlapping somewhat compromises the independence of cross validation as a test and training point can share data. However, the advantage of capturing more events outweighs this. The common conclusion is there is a region of overlap with better performance than the 0% case. To compromise a middle value of 50% was used.

This experiment suggests that all these variable factors are not truly independent, with everything changing on a very complex range of factors. The idea of these experiments is to find a good starting point for these parameters which will be tuned with the real data. When the overlap factor was included, the optimal window size increased to 20 samples or one second.

3.6.3 Feature Selection and Optimisation

Inclusion of Magnetic Features

The magnetometer's primary role in the system is in the sensor fusion algorithm. The data stream provides information on the absolute orientation of the person, as opposed to the relative one given by gyroscopes. This has the potential to provide information about which activity the person was completing due to different positions at a desk. In a real world system, this would not be a valid practice as orientations change with scenery. To see the effect of this inclusion of this data stream an experiment was conducted with the same four-person, five-activity dataset.

The results suggested the contrary conclusion that performance increased from Accuracy 61.0% and ROC 0.756 to 62.2% and ROC 0.879 once the magnetometer was removed from the feature vector. The reason for the reduction is that the compass is a noisy signal and this noise harmed the classification algorithm. Selection of features does need some critical evaluation because a feature could either be an artefact of the test environment (eg desk orientation) or a versatile feature which is repeatable.

Clustering

One tool in validating feature selection is the clustering of the data points in feature vectors. Clustering tries to break the data down into various groups using Gaussian distributions. Where Bayesian classifiers build models for each class to predict the likelihood of an instance being in that class, clustering builds up models for groups of instances. It builds up the groups to find the most likely groupings of data. The clustering can be compared to the classes to evaluate the performance of the features in segregating data for easy classification.

Clustering was tested using the Weka toolbox. Weka uses the expectation maximisation clustering algorithm, a form of hard clustering. The method works by building up probabilities around instances, and extending clustering groups. The most likely groups survive, and cover distinct regions of the feature space. The clustering can be controlled by inputting the desired number of clusters.

The Contingency matrix in Figure 3.6 shows the distribution of the clusters as columns compared to the ideal classes as the rows. It can be seen that cluster 1 relates to the high rotation of Rubik and writing, clusters 2-4 cover the high movmement of walking and the others cover the other classes. These results suggest that the clustering loosely matches the classes, but not fully requiring sophisticated classifiers. The clustering results also suggest the spread of instances within each class across many clusters, indicating some dynamic behaviour within activities.

Table 3.6: Contingency Matrix - Distribution of clusters and actual classes. Diagonal entries would suggest each class is in distinct clusters. However, the walking class covers the majority of the feature space leading to occupancy of a few classes. The other clusters show some division between the classes.

Class/Cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Read	0	17	0	0	261	998	2505
Rubik	241	32	0	0	230	1957	1321
Walk	0	1311	1051	1287	132	0	0
Watch	0	42	0	0	199	563	2977
Write	966	89	0	4	202	1488	1032
Talk	7	409	10	24	1053	798	1480
Arithmetic	0	48	1	0	130	2322	1280

Long Term Features

Some experimentation was conducted with long term features and template matching. Features either can characterise individual movement events on the scale of one second or longer term patterns. Long term features capture information about how the movement changes over time, or how they stay the same. Template matching usually occurs on the power spectrum, where spectral patterns of activities are characterised, and instances are compared to these templates. One simplistic comparison involved the power spectrum centroid and slope. During experimentation, this feature successfully discriminated walking from the other activities, but provided little value in the other activities. The core movements of the other activities are too similar when taking the longer term view.

An alternative scheme for long term features is the utilisation of many different sized windows, to capture statistical features over different time frames. This recent history assumes the activity stays constant for reasonable periods of time. During experiments with windows up to one minute, these longer windows provided no real benefit. The statistics get blurred over time, becoming similar for different windows as they feature common head movement events.

Where long term features become important is in building more complicated models of activities. Temporal relationships between subtasks, may require long term features, and help build up the context of events. These functions could work in conjunction with state based classifiers such as Markov Chains which use the previous instance as the starting point for classification. In this study's case, simplicity was preferred when no significant gain could be achieved.

Inclusion of Hip Device

The importance of the hip reference device is an interesting side note. The device is common practice in wearable inertial sensors, and isolates limb movements from the whole body. As will be seen in Section 4.1, the hip features appear a few times, as they provided strong discriminating power over the walking activity. They are also indicative of upper body movements such as the shoulders and arms in the writing, arithmetic and Rubik problems. The vibrations of these movements are observed in the hip device as small fluctuations.

In this experiment it was also used to record the posture of participants. Through comparing differential changes in sensor orientation, posture shifts were recorded and included in the classification. The differential signal is important as it counters changes in position of the sensor. The Flexion (Figure 2.1) head position was the most useful, as it indicated focus on the table or the computer.

3.6.4 Classification Tests

Classifier Selection and Tuning

Some preliminary investigation was completed in assessing classifier performance between the alternate methods. This analysis narrowed down the range of potential methods. Preliminary tuning was also conducted to see what was most promising. A smaller selection of methods was required to reduce the processing time.

For the J48 classifier, the main parameter in the decision tree is the minimum number of nodes on each leaf. To avoid over-fitting, this parameter was set to 20 instances, allowing a tree of maximal depth of approximately seven levels. For k-Nearest Neighbour, the k parameter was tuned for performance. While a value of one gave the best performance, a value of ten was chosen to provide some versatility and consistency.

Meta Classifiers

Within the set of methods outlined above, the classifier structure can be adjusted in many ways to improve performance.

Two Layered Classifier Structure

Given the gap between the size of the window (500ms) and a likely application demand (30s), there is room for a second layer of classifier. One can also assume that the actual activity is unlikely to be changing much within a thirty second window, and if it is the transition would be detected. This second layer takes the predicted class output for each window and guesses the task over the longer period. For example the modal task is a simple method of achieving this.

Table 3.7 shows an example of this process with a rolling 5 segment window. This system takes a majority vote of the time outputs for the current window and previous ones. The advantage of this is in utilising temporal stability in the activity to remove outliers.

Table 3.7: Two Layered Classifier - Majority Voting Example. In this example a majority vote is conducted on a rolling window of the last five time segments. The accuracy is generally improved, except during transitions

Time (s)	**Actual**	**Predicted**	**Majority**
0	Reading	Reading	Reading
0.5	Reading	Reading	Reading
1	Reading	Reading	Reading
1.5	Reading	Reading	Reading
2	Reading	Watching	Reading
2.5	Reading	Reading	Reading
3	Reading	Reading	Reading
3.5	Reading	Talking	Reading
4	Reading	Reading	Reading
4.5	Reading	Walking	Reading
5	Watching	Watching	Reading
5.5	Watching	Watching	Watching
6	Watching	Talking	Watching
6.5	Watching	Watching	Watching
7	Watching	Watching	Watching
7.5	Watching	Watching	Watching
8	Watching	Walking	Watching
8.5	Watching	Watching	Watching
9	Watching	Watching	Watching
9.5	Watching	Watching	Watching
Accuracy		75%	95%

However, this system assumes minimal transitions, with delayed recognition of activity switches.

The problem with adopting this method is that it is very application-specific in its assumptions, so it was considered further in this study. However, it should be kept in mind that an accuracy of about 70% is sufficient to get very accurate results in most applications with temporal patterns.

Cost Sensitive Classifiers

For multilevel classifications where the classes are degrees of an underlying phenomenon, some mislabelling of classes is worse than others. A cost can be attached to the classifiers to account for the distance away of classes. This is commonly conducted with a cost matrix so that the cost is minimised reducing occurrences of extreme mislabelling. The consequence of this is more middle classifications and less on the extremities. However, for most applications this is desired behaviour, so adjustments are made when the load changes outside of a normal range.

In Weka this is the *MetaCost* and can be used in conjunction with any other classifier. The following tables show an experiment to see the distributions with different cost matrices. Essentially a cost is applied to a result being in each element of the confusion matrix. Table 3.8 shows equi-weight cost where opposite classes have an equal penalty to adjacent classes. Table 3.9 shows less opposite predicted classes, but more adjacent classes than correct states. Table 3.10 shows a further optimised state with minimal opposite states.

Table 3.8: Cost Matrix 1 - Effect on Confusion Matrix - Accuracy 89.1143%, ROC 0.931

Cost	L1	L4	L7	**Confusion**	L1	L4	L7
L1	0	4	4	L1	502	23	54
L4	4	0	4	L4	30	856	53
L7	4	4	0	L7	61	58	926

Table 3.9: Cost Matrix 2 - Effect on Confusion Matrix - Accuracy 87.6707%, ROC 0.918

Cost	L1	L4	L7	**Confusion**	L1	L4	L7
L1	0	2	6	L1	465	77	37
L4	2	0	2	L4	23	870	46
L7	6	2	0	L7	40	93	912

Table 3.10: Cost Matrix 3 - Effect on Confusion Matrix - Accuracy 84.1982%, ROC 0.9

Cost	L1	L4	L7	**Confusion**	L1	L4	L7
L1	0	1	10	L1	393	175	11
L4	1	0	1	L4	22	875	42
L7	10	1	0	L7	22	133	890

Voting

Another technique is voting classifiers. This utilises many classifiers and gets them to collaborate to achieve the best result possible. Many predictions are made for the same feature vectors, and collaborating on results can improve performance. The bagging methods makes many models of different combinations of the training set. Randomisation varies the way the trees constructed, not the data itself. Boosting iteratively improves runs, by focusing on failed instances. Stacking runs many classifiers on a base level and then a secondary classifier on the original outputs.

Table 3.11 shows the performance results of these voting classification methods. While all methods improved the results over the baseline, none had significant performance gain worth the extra computational complexity, high variability in the models and risk of over-fitting.

Table 3.11: Voting Classifiers - While these techniques improve nominal classifier performance, they are disadvantageous in their complexity and over-fitting

Method	**Accuracy**	**ROC Area**
Baseline (J48)	89.6606%	0.935
Bagging (Bagging)	91.3383%	0.988
Randomisation (Randomised Forest)	92.2747%	0.989
Boosting (AdaBoost1)	93.3281%	0.992
Stacking (Stacking)	90.1288%	0.933

The following section explores the classification of subsets of the main classification problems. The main results will be discussed in Section 4.2

Task Property Problem

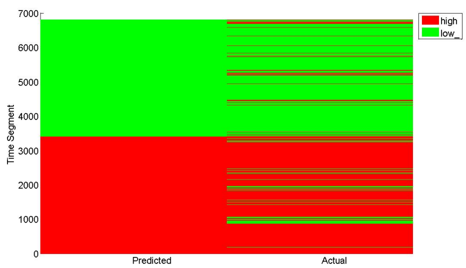
An alternate way of classifying the tasks is through the features discussed in Table 3.5. Each property can be analysed as a simple two class problem between low and high in the property. To investigate if this is a viable approach, quick analysis was done using the subsets of the activities where the other three features were kept constant. Inherently this is a limited approach as the noise from the completion of activities does not affect each class evenly. That is to say, that some of the classification is of the difference in task, not difference in level of the feature.

The results are shown in Table 3.12 and Figure 3.15. From the results, the two class

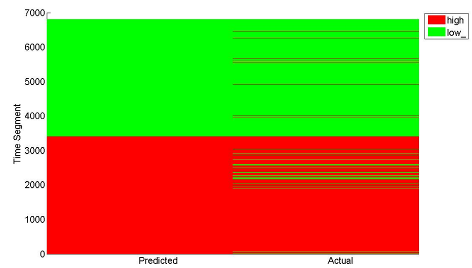
Table 3.12: Cost Matrix 3 - Effect on Confusion Matrix - Accuracy 84.1982%, ROC 0.9

Feature	**Low Activity**	**High Activity**	**Accuracy**	**ROC Area**
Cognitive	Watching	Arithmetic	90.7%	0.907
Motor	Watching	Writing	97.3%	0.970
Communicative	Watching	Talking	83.3%	0.833
Perceptual	Writing	Rubik	90.4%	0.904

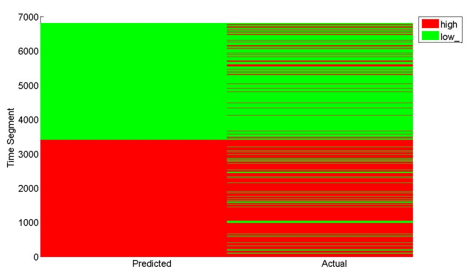
problems are successful in distinguishing the categories. These accuracies are similar



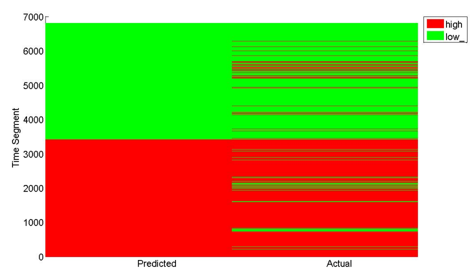
(a)



(b)



(c)



(d)

Figure 3.15: Task Property Confusion Plot - (a) Cognitive (b) Motor (c) Communicative (d) Perceptual. Note that the accuracy is not statistically significantly above discriminating between any two activities. More classes are required for better analysis.

to any two activity classes classification. That confirms that measuring the underlying factors is a more difficult problem to measure. It needs a larger number of different classes to assess if it is viable. From the results it is clear that motor activities are the easiest to distinguish, while communicative is the hardest.

Activity Problem

As classes are added to the activity problem, the performance of the system decreases. Physical task analysis can recognise an activity compared to no activity, as physical movements have distinctive patterns which are predominately unique to the activity. The problem with the stationary activities, is the movements being analysed are common to a lot of activities. As activities are added, the overlap increases and the performance drops off rapidly.

Within the experiment, this effect was established with grouping of activities that are quite similar. The relatively low movement activities of watching, reading and arithmetic are going to be hard to distinguish. Adding more activities in the relatively higher levels of activities should give greater discernibility as they will have movements that can be characterised.

The overlap of each class can be seen in the detailed class statistics and confusion matrix. Walking is very distinctive and is the class with the highest performance, a true positive rate of 0.963, well above the average of 0.756. The next three best were writing (0.84 TP), arithmetic (0.811 TP) and the Rubik's cube (0.757 TP). These moderate level movements, can be justified by the difference in their movement, the side to side movement of writing, the vibrations and visualisations of Rubik's cube and the mouse movement of arithmetic, while on the computer. In a three class classification of these three activities, the performance increased significantly to an accuracy of 88.7%. This suggests the dynamic behavior of these activities, with periods of movement, and periods of passivity, such as while thinking about the next action.

The lowest performing classifiers are the low relative movement ones of reading (0.638 TP), watching (0.669 TP) and talking (0.614 TP). Reading and watching have the greatest paired misclassification due to the similarity of the activities. The two class problem of reading and watching, has an improved accuracy of 81%, but the majority of this is the removal of false positives from low motion windows of other activities. Talking is an interesting class, as it has a large spread within all the feature vectors. As an action, talking can have many sudden movements, nodding, laughing, moving to face who is talking. However, it can have low motion as well. The poor classification results reflect the stretched profiles which get misclassified as most of the other classes. Interestingly, in a six class classification problem (talking removed), the performance increases significantly, beyond improvement of reduction in classes. The new classification accuracy is up near 85%. The five class problem (without watching or talking) has accuracy up to 90%.

Cognitive Load Problem

The cognitive load is a continuous property that ranges from low levels to high levels. The number of levels of measurement is related to the predictive capacity and the application. Too many levels can cause a loss of determinative ability as the trends overlap too much and instances are mislabelled.

A quick comparison between the two class problem (low/high) and the three class problem (low/medium/high) was made to see if the extra predictive power was feasible given the dataset. Unlike the other problems, these classes have a high degree of overlap within the trends, making the classification problem much harder. The three class problem had an accuracy of 69%, but the majority of the errors were in adjacent errors (one level off) not opposite errors (two levels off). The two class problem had higher results with an accuracy of 85% and better ROC values. The improvement is equivalent to an improvement in guessing due to having less classes. The one negative consequence of a two class problem is a high penalty for misclassification as all errors are opposite errors. The benefits of the three class method are understood and will be used in the major analysis.

3.6.5 Subject Variability

Classifier Results

A good feature vector and classification algorithm is adaptable to new individuals using the system. Subject variability arises from the different ways people carry out activities. The first approach is to find the underlying features which do not change between people. The other approach is to find models which once trained with the person's behaviour, can then easily adapt and predict future behaviour for that person.

Table 3.13 shows the results of a simple classification. A preliminary dataset of four people doing five activities was taken. Each person had their data as the testing dataset with both J48 and IB1 (nearest neighbour) classifiers. The experiment varied the training set from only other people's dataset and only their dataset. For the latter case, cross-validation is needed as the same dataset is training and testing the model.

The table shows the performance of the classifier with Accuracy and ROC area metrics. The results demonstrate the subject variability, with very poor performance with everyone's else training data. The results are barely above guessing. Within the self training, the unreliable results suggest that the consistency of activities varies between subjects.

Table 3.13: Subject Variability

Participant	**Only Self Training**		**Only Other Training**	
	Accuracy	**ROC Area**	**Accuracy**	**ROC Area**
1	58.4%	0.761	24.0%	0.521
2	42.4%	0.692	22.4%	0.515
3	70.4%	0.801	17.6%	0.507
4	83.2%	0.920	21.6%	0.515

Feature Consistency

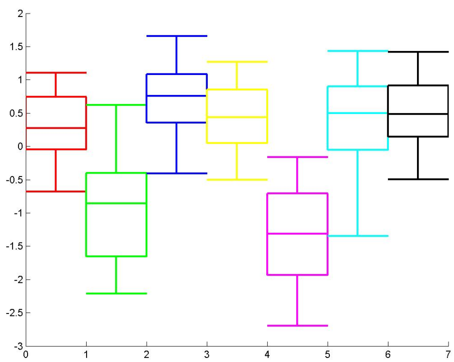
Another aspect of subject variability is whether the same features have similar trends for each person. The variability can be either in the magnitudes and gaps of the trends or the trends themselves. Using the preliminary dataset of four persons, an experiment was conducted to see which features are resistant to change between people, while also looking for deterministic ability for the classes.

The figures below show the overall distributions of the 10% percentile linear acceleration in a window (Figure 3.16 (a) shows the overall distribution and Figures 3.16 (b) and (c) show two individual trends). The feature shows the orientation of the head, with writing and Rubik's cube focused on the desk, and the other activities on the computer. The individual trends have narrow distributions, as the head stays in the one spot to complete the task. As seen been (b) and (c), these positions can vary between people, especially for the talking activity. While the magnitudes vary, the nominal order between activities stays the same, allowing for group classification.

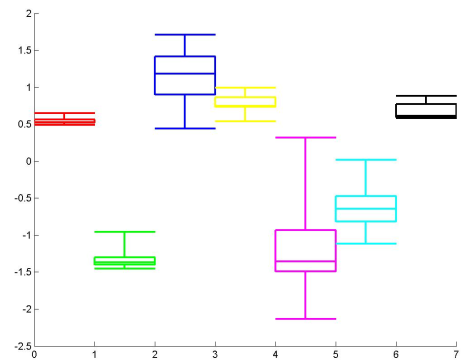
Subject Normalisation

Bayesian classifier techniques attempt to fit a Gaussian distribution to each feature distribution for each class. The model is trained to find the parameters (mean, variance) of each distribution. The classifier makes predictions by finding the highest probability of a new instance belonging to the feature distributions of each class. A core assumption of this technique is the independence of each of the feature parameters, but still works reasonably when this assumption is clearly broken. This fairly simplistic method works effectively on a dataset containing a single person, as their distributions are approximately Gaussian.

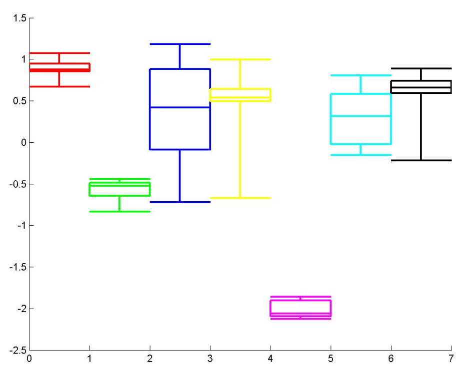
An experiment got an average of 70% accuracy on the individual datasets and 35% accuracy on the combined dataset. The huge difference is mainly due to the non Gaussian behaviour of the distributions. Each individual has slightly different profiles with respect to the magnitude of feature peaks, for example different walking speeds. When combined



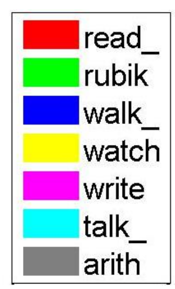
(a)



(b)



(c)



(d)

Figure 3.16: Activity Task Normalised Feature Distribution Comparison - Linear Acceleration X Axis 10% Percentile. (a) Group distribution (b) Participant 2 (c) Participant 19 (d) Legend of Class Colours. Note the similarities and differences in features, the group has much wider spread of values.

this leaves a distribution with many peaks. To combat this effect, subject normalisation was attempted where each individuals feature's were pre-normalised using a Z Score (mean and variance) algorithm. This attempts to keep the trends but to scale them to the same positions. After this normalisation, the whole dataset is normalised again to standardise the weightings for distance classifiers. This resulted in an improvement of the whole dataset accuracy for Bayesian methods of up to 50%. The difference to the individual scores is reflected in changes in trends of the classes.

3.7 Summary

A significant amount of experimentation was conducted with the pilot testing. The system design was shaped by two core motivations of system performance and simplicity. Given the large amount of data for the small set of outcomes, simpler features and classifiers are the most effective for this study. The next chapter details the final system design in the features selected and the classification results.

Chapter 4

Evaluation

This chapter details the results of the classification problem and presents a discussion of their significance. The results for each problem are presented, followed by discussion and comparison to other studies. The results of the survey are also presented, with reference to the social context of the technology.

4.1 Feature Selection

4.1.1 Final Selection

The feature selection methods from Section 3.3.5, have been implemented for three different classification problems: activities, cognitive load and transitions. From the deployment data, feature vectors of length 100 were fed into the feature selector algorithm for each participant. The algorithm chose the best twenty features as a subset. The most frequent features across the participants have been chosen for the final subset.

Table 4.1 shows the selected features for each classification problem. The number of hip features reflects the power of that feature in discriminating walking in the activity class. It is also a reflection of how still the subject is in their chair. Vibrations of their back, neck and head movement are felt in the hip device.

Table 4.1: Selected Features for each classification problem.

Activity	Cognitive Load	Transition
Acceleration Jerk Mean	Zeroed Head Acceleration Mean	Zeroed Head Acceleration p90%
Gyroscope Jerk Mean	Zeroed Hip Acceleration p90%	Zeroed Head Acceleration p10%
Zeroed Head Acceleration p90%	Zeroed Hip Acceleration p10%	Zeroed Head Acceleration Mean
Zeroed Hip Acceleration p90%	Zeroed Hip Acceleration Mean	Zeroed Hip Acceleration p90%
Zeroed Hip Acceleration p10%	Zeroed Hip Gyroscope p10%	Zeroed Hip Acceleration p10%
Hip Acceleration p90%	Zeroed Hip Gyroscope Mean	Zeroed Hip Acceleration Mean
Hip Acceleration p10%	Head Gyroscope p90%	Zeroed Hip Gyroscope p90%
Hip Acceleration Mean	Hip Acceleration p90%	Zeroed Hip Gyroscope p10%
Hip Gyroscope p10%	Hip Acceleration p10%	Zeroed Hip Gyroscope Mean
Hip Gyroscope Mean	Hip Acceleration Mean	Head Acceleration p90%
Posture X Axis p90%	Hip Gyroscope Gyroscope p90%	Hip Acceleration p90%
Posture X Axis p10%	Hip Gyroscope Gyroscope p10%	Hip Acceleration Mean
Acceleration X Axis Mean	Posture X Axis p90%	Posture X Axis p90%
Acceleration X Axis p90%	Acceleration X Axis p90%	Posture Y Axis p90%
Acceleration X Axis p10%	Acceleration X Axis p10%	Posture Y Axis p10%
Acceleration Y Axis Mean	Acceleration Y Axis p90%	Acceleration Y Axis p90%
Acceleration Y Axis p90%	Acceleration Z Axis Mean	
Acceleration Y Axis p10%	Acceleration Z Axis p10%	
Acceleration Z Axis Mean	Acceleration Z Axis p90%	
Acceleration Z Axis p90%		
Acceleration Z Axis p10%		

It is interesting to observe the high prevalence of very simple statistics in the final set (compared to the original selection, refer to Table 3.2). The reasons for the prevalence of percentiles and means are two-fold, in having nice Gaussian distributions and being in the range of processed signals. For the short windows, the simplest statistics are often the most informative, with the others not appearing in the optimal subsets. As seen in the Table, the range of signals, removes the need for complicated features. The desired information can be extracted through processing the signal into a new form and then extracting simple features. This results in a simpler system and a better understanding of the underlying behaviour.

4.1.2 Physical Meaning of Features

For the activities, these features describe different aspects of the movement. The hip movements are used for separating the walking activity from the rest. Posture is used to distinguish whether the head is directed towards the table or the computer screen. The other head signals (such as the jerk) capture the small distinct movements related to each activity. The axis based features are used due to the different directions of movement, with writing, reading and Rubik's providing movement in different directions related to actions of the task.

The cognitive load features have more significant overlap in their distributions. The dominance of the zeroed signals reflect analysis of the importance of scale of movements in their relative contexts for individuals, for example this trend appears in the scatter plot of two features for Participant 1 in Figure 4.1. The challenge in this problem is effect the different frequency of movements related to entering the arithmetic answers for each cognitive level. This may result in higher results than expected for classification. The lower loads had a more active participant, responding to the new questions and entering the response. Higher loads have the unconscious movements related to thinking about the problem. It is hard to assess a general trend, but within one subject, there are patterns which can separate the amount of load. It should be noted, that many of these features are common to activity problem, which implies the nature of activity will affect the cognitive prediction. This is not desirable, but cognitive classification is a challenging problem.

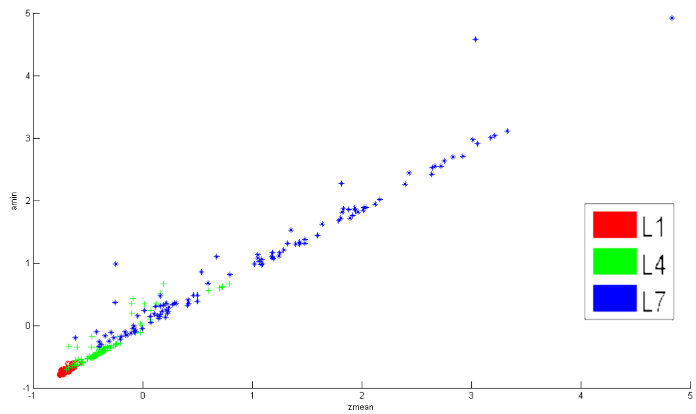


Figure 4.1: Cognitive Load Feature Scatter - Participant 1 - This shows a scatter plot of the distribution of classes for two features: Acceleration Z Axis Mean and Zeroed Hip Gyroscope p10%. As load increases the relative movement increases, supporting the offloading hypothesis. The trend disappears for the group dataset due to subject variability.

The transition features attempt to capture the sudden shifts in motion from the directed changes to activities. These events are characterised by posture shifts and head movements reflected in the features selected. Distinguishing these movements is difficult as the non-transition state has a wide range of activities. Again, the zeroed features dominate, reflecting the importance of locally high movement in distinguishing a transition. In this way these zeroed signals are long term features.

Subject variability is core issue in feature selection. The features were chosen through finding the ideal subsets for each individual. The majority of features were similar for activities and transitions suggesting similar trends. The main difference was in the orientation of the head and the intensity of the movements. In general most of the trends were similar, but the overlapping nature of the dataset, requires significant personal training data to get any meaningful results.

4.2 Classification Results

The results in this section are structured into five areas. The three main identification problems are the activities, cognitive load and transition detection. Subject variability was analysed, to see how much calibration is needed to adapt to the individual. Finally, the success of the problem is explored in a pseudo real world situation. For each section, the results are in two parts, the deployment results and the evaluation dataset results

4.2.1 Activities

Table 4.2: Activity Dataset

Property	**Value**
Persons	21
Classes	7
Time/per class	2 minutes
Deployment - Set	19 Persons
Evaluation - Set	2 Persons
Evaluation - Personal	10 seconds/class/person
Features	See Table 4.1
Test Arrangement	Percentage Split (Train/Test) 10%
Balancing	Class and Participant

This section describes the results for the seven-class activity problem. Table 4.2 outlines the nature of the dataset used in the classification. The numerical results are displayed in Table 4.3. Figure 4.2, is a graphical representation of the classification results for the different classifiers. This representation is a visual confusion matrix showing the distribution of classes. The top row is the ground truth of colour, with the x axis being time segments, and different colours being different colours. Each subsequent row is a method of classification. Colours lining up represent successful classification, as can be seen in the blue with the walking activity. This representation provides insight into the distribution, for example the misidentification of reading (red) and watching (yellow).

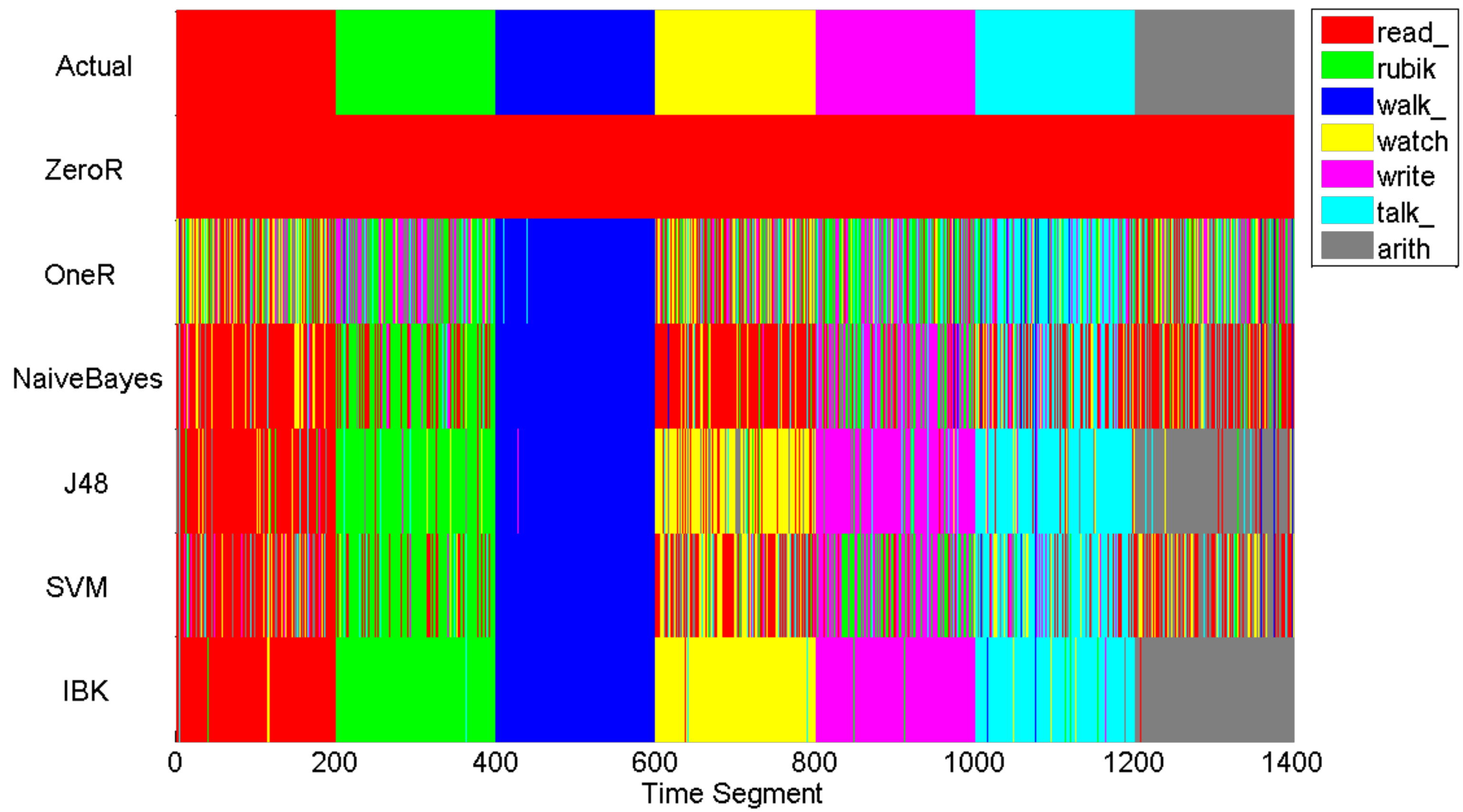


Figure 4.2: Activity Group Deployment Results - Graphical Representation of Classification Results. This is a visual representation of a confusion matrix with the classes represented by different colours. The X Axis is time segment index, Top row is actual class, other rows are for the classifiers. Note the success of walking classification in the common blue column. Also note the misidentification of reading (red) and watching (yellow)

Table 4.3: Activity Deployment - Comparison of Classifier Methods

Method/Accuracy and ROC Area	Deploy Group	Deploy Indiv	Eval Group	Eval Indiv
ZeroR	14.3% 0.500	14.3% 0.500	14.3% 0.500	13.9% 0.500
OneR	40.7% 0.577	69.6% 0.827	55.8% 0.763	79.1% 0.849
NaiveBayes	51.7% 0.815	93.5% 0.978	60.1% 0.907	89.4% 0.990
J48	69.5% 0.876	96.4% 0.997	69.1% 0.917	92.7% 0.993
SVM	62.7% 0.810	89.7% 0.983	67.3% 0.806	87.5% 0.981
IBK	84.1% 0.978	89.4% 0.954	82.4% 0.963	89.5% 0.949

Table 4.4: Activity Group Deployment - Best Method IBK - Detailed Accuracy By Class And Confusion Matrix

Class	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**A/P**	**A**	**B**	**C**	**D**	**E**	**F**	**G**
A	0.867	0.057	0.718	0.867	0.785	0.996	**B**	2807	19	3	219	0	63	127
B	0.938	0.018	0.898	0.938	0.917	0.996	**C**	133	3046	2	0	56	5	5
C	0.970	0.008	0.954	0.970	0.962	0.998	**C**	7	13	3285	20	6	61	4
D	0.705	0.041	0.744	0.705	0.724	0.951	**D**	507	0	12	2301	4	128	312
E	0.902	0.005	0.967	0.902	0.933	0.994	**E**	0	232	5	6	2914	75	0
F	0.653	0.022	0.83	0.653	0.731	0.957	**F**	297	86	67	395	26	2107	247
G	0.849	0.036	0.798	0.849	0.823	0.974	**G**	155	0	61	170	1	101	2745
Av	0.841	0.027	0.844	0.841	0.839	0.978								

4.2.2 Cognitive Load

This section outlines the results for the cognitive load measurement with the arithmetic task. There are three levels of cognitive load low - L1, medium - L4, and high - L7.

Cognitive Load Validation

To validate the experiment design in creating different levels of cognitive load in the participants, three different metrics were recorded. Each metric has its own weaknesses and limitations. The results are shown in Table 4.5. It is clear the self reported difficulty increased for each level beyond the standard deviation. The correctness percentage dropped and the time taken increased. This validates the subjects had different levels of cognitive load for the same task.

Table 4.5: Cognitive Load Validation Metrics - Average Values and Standard Deviations

Level	**Self Report(1-10)**	**Correct(%)**	**Average Time (sec)**
L1	1.42 ± 0.60	0.99 ± 0.02	3.46 ± 0.50
L4	3.86 ± 1.41	0.97 ± 0.04	8.34 ± 2.24
L7	6.62 ± 1.23	0.85 ± 0.17	18.94 ± 6.04

Deployment and Evaluation Results

The experimental configuration is shown in Table 4.6. The results are displayed in Table 4.7. Figure 4.3 shows the results for all the classifiers on the group deployment dataset.

Only the Nearest Neighbour (IBK) and J48 decision trees are effective in this classification. Recall the limitation of different completion times for entering answers, which explains part of the classification success. It is interesting the observe the amount of opposite misidentification, that is low load classified as high and visa versa.

Table 4.6: Cognitive Dataset

Property	**Value**
Persons	21
Classes	3
Time/per class	dependent on subject
Deployment	19 Persons
Evaluation - Set	2 Persons
Evaluation - Personal	10 seconds/class/person
Features	See Table 4.1
Test Policy	Percentage Split (Train/Test) 10%
Balancing	Class and Participant

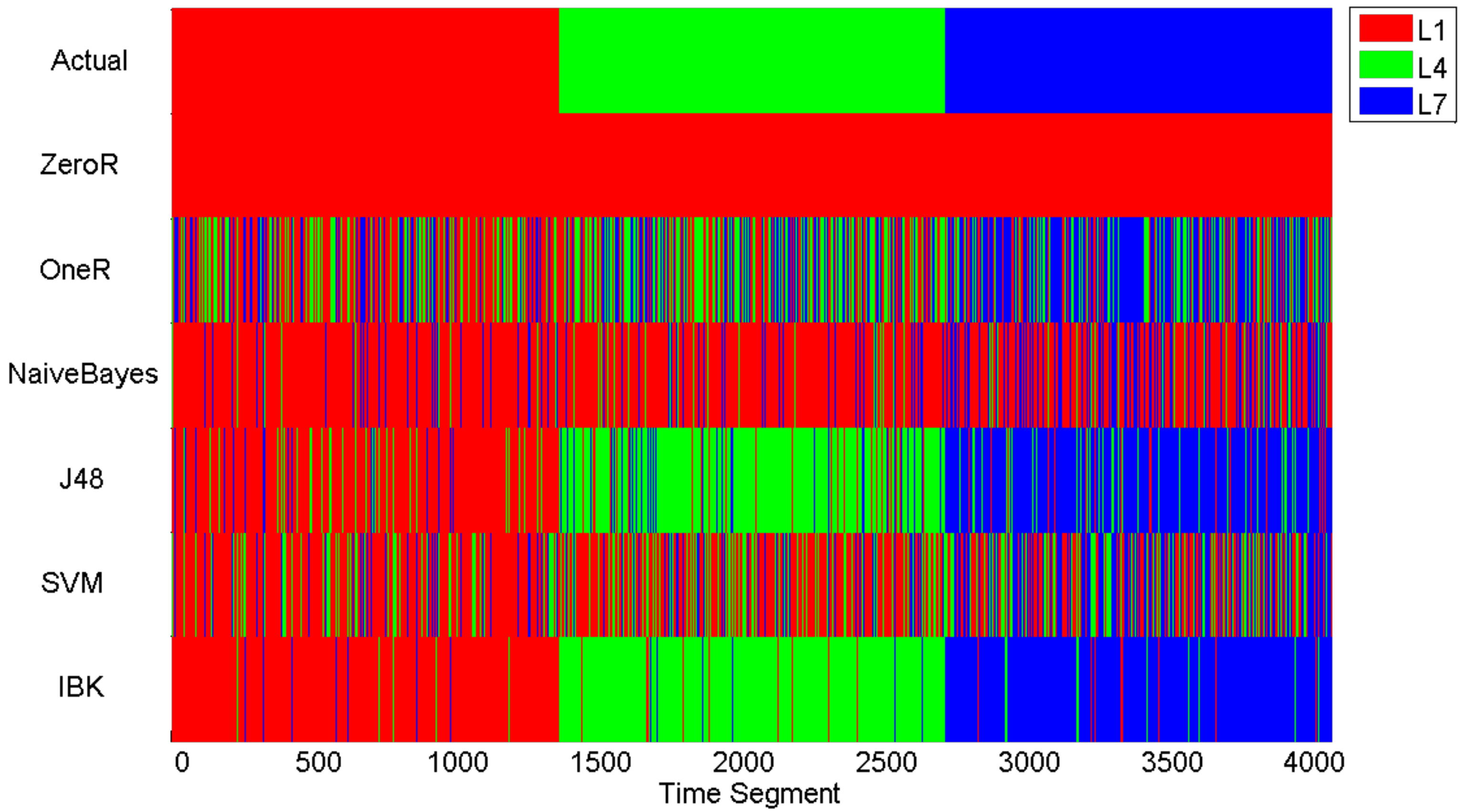


Figure 4.3: Cognitive Load Group Results (Cross Validation) - Graphical Representation of Classification Results. X Axis is time segment index, Top row is actual class, other rows are the other methods. Note J48 and IBK are the only successful classifiers

Table 4.7: Cognitive Load - Comparison of Classifier Methods across Deployment and Evaluation; Group and Individual

Method/Accuracy and ROC Area	**Deploy Group**	**Deploy Indiv**	**Eval Group**	**Eval Indiv**
ZeroR	33.3% 0.500	33.1% 0.500	33.2% 0.500	33.2% 0.500
OneR	48.3% 0.620	70.6% 0.806	50.9% 0.639	62.9% 0.724
NaiveBayes	40.7% 0.623	82.7% 0.942	43.4% 0.712	83.4% 0.951
J48	58.8% 0.757	76.5% 0.850	56.8% 0.780	71.4% 0.827
SVM	43.8% 0.579	81.7% 0.894	42.8% 0.560	85.7% 0.944
IBK	74.5% 0.902	83.9% 0.901	66.6% 0.842	82.1% 0.869

Table 4.8: Cognitive Load Group Deployment - Best Method IBK - Detailed Accuracy By Class And Confusion Matrix

Class	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**A/P**	**A**	**B**	**C**
A	0.862	0.190	0.698	0.862	0.771	0.902	**A**	1059	127	43
B	0.697	0.145	0.706	0.697	0.701	0.869	**B**	292	844	75
C	0.676	0.048	0.873	0.676	0.762	0.934	**C**	166	225	814
Av	0.745	0.128	0.759	0.745	0.745	0.902				

4.2.3 Transition Detection

The experimental configuration for the detection detection is shown in Table 4.9. The numerical results are displayed in Table 4.10. Figure 4.4 show the graphical representations of the transition classification. Clearly the skewed proportions of the training data have affected the performance of the model.

Table 4.9: Transition Dataset

Property	**Value**
Persons	21
Classes	2
Time/per class	transition: 5×10 seconds, non: 5×110 seconds
Deployment	19 Persons
Evaluation - Set	2 Persons
Evaluation - Personal	10 seconds/class/person
Features	See Table 4.1
Test Policy	Cross Validation
Balancing	Participant only

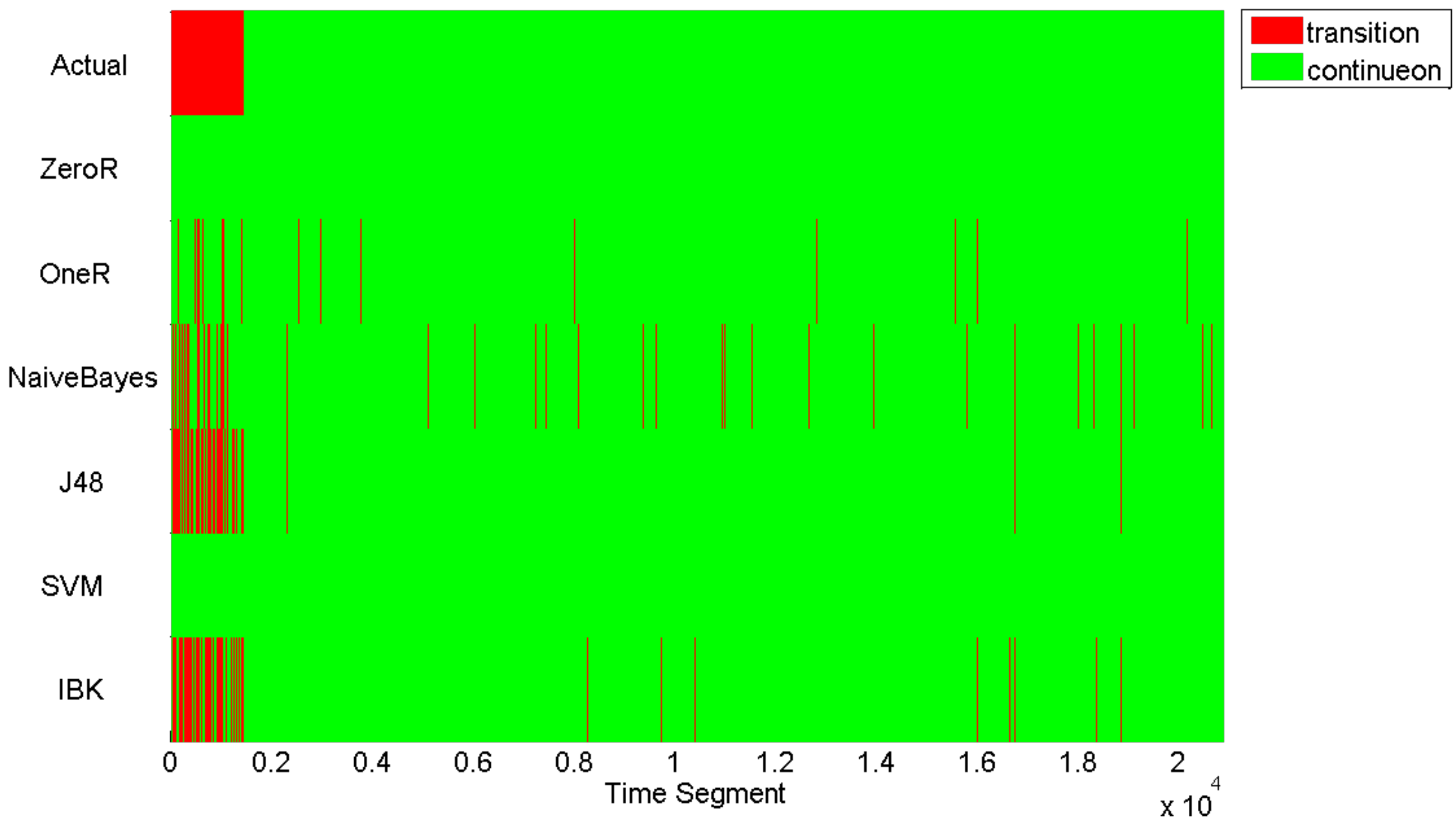


Figure 4.4: Transition Detection Group Deployment Results - Graphical Representation of Classification Results. Note the skewed arrangement of the dataset leads to more false positives.

Table 4.10: Transition Detection - Comparison of Classifier Methods

Method/Accuracy and ROC Area	**Deploy Group**	**Deploy Indiv**	**Eval Group**	**Eval Indiv**
ZeroR	93.1% 0.500	93.1% 0.500	91.7% 0.500	91.7% 0.500
OneR	92.9% 0.539	97.0% 0.830	96.1% 0.818	97.9% 0.893
NaiveBayes	92.0% 0.753	83.9% 0.834	53.7% 0.816	67.5% 0.835
J48	97.4% 0.935	97.4% 0.888	97.9% 0.910	97.4% 0.927
SVM	93.1% 0.500	94.9% 0.652	93.4% 0.612	98.1% 0.981
IBK	97.1% 0.974	97.2% 0.973	99.1% 0.959	95.2% 0.715

Table 4.11: Transition Detection Group Deployment - Best Method IBK - Detailed Accuracy By Class And Confusion Matrix

Class	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**A/P**	**A**	**B**
A	0.623	0.003	0.937	0.623	0.748	0.974	**A**	897	543
B	0.997	0.377	0.973	0.997	0.985	0.974	**B**	60	19380
Av	0.971	0.351	0.970	0.971	0.968	0.974			

4.2.4 Real World

Experimentation on the real world data is difficult as there is no ground truth. Notes taken during the experiment were used to evaluate the closest class being conducted. Figure 4.5 shows the raw data for Participant 3 during their 10 minutes of free time. Figure 4.6 shows the classification results for activity, transition, and cognitive load. On first impressions, the results aren't very good, but it is worth considering the reasons behind the poor performance. Firstly, for simplicity, a common feature subset (activity from Table 4.1) was used for all three classification problems. The reading task was actually on a mobile phone positioned on the desk, hence the misidentification as Rubik's Cube. The main issue was that with variety of stimuli, all the activities became more dynamic.

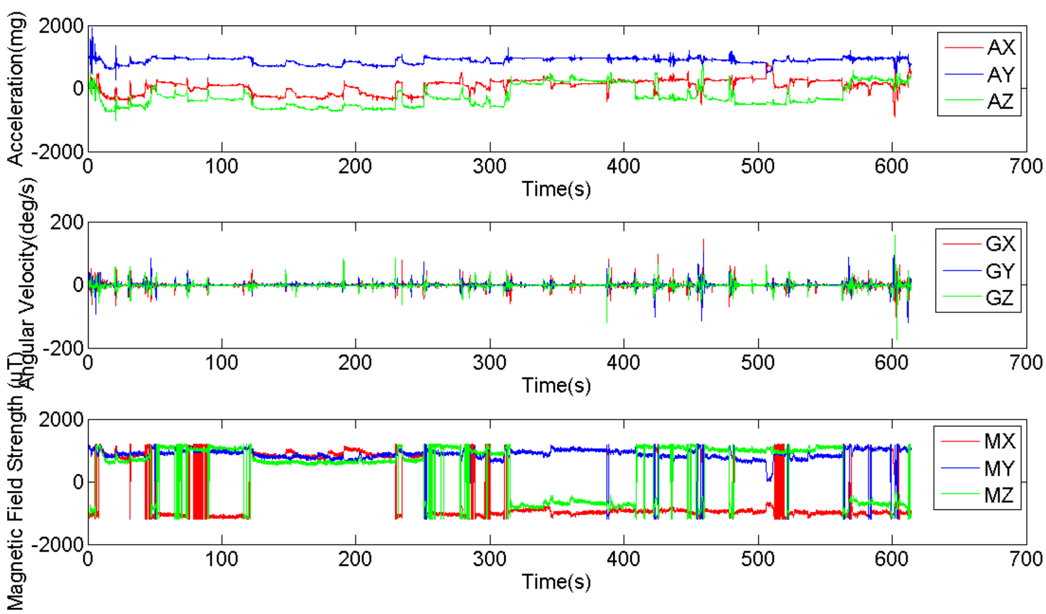


Figure 4.5: Real World Raw Data - Participant 3.

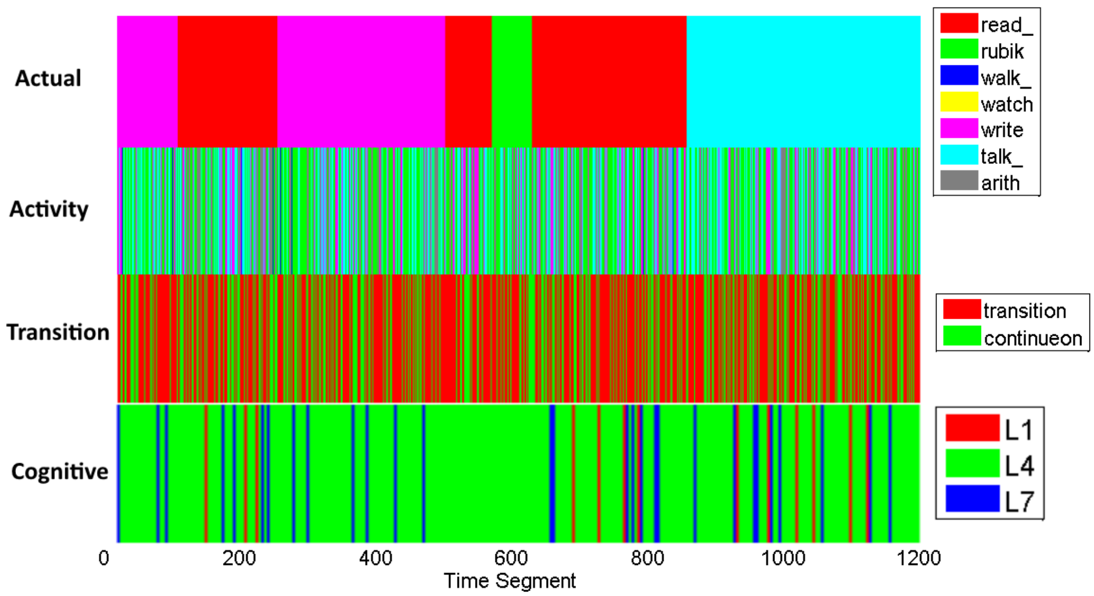


Figure 4.6: Real World Raw Data - Participant 3 - The actual data shows the closest task chosen by the author. The second row shows the activities predicted by the model, third row shows the transition state and bottom row shows the cognitive load levels

4.3 Discussion

4.3.1 Activities

The seven-class activity problem results are promising for the novel technique of head movement. The deployment results achieved accuracies of 84% across the combined dataset and an average of 89% on individual participant datasets (IBK). The evaluation dataset had no noticeable change in performance, with some improvign slightly and some decreasing. The results are dependent on the experimental conditions and classification structure, making it hard to make comparisons to other studies in the literature.

As discussed in the previous chapter, the classifier is more successful with some activities than others. The walking task was the easiest to isolate, as it involves large distinctive movements. Activities with moderate relative movement (arithmetic, writing, Rubik's Cube) were successful as they had unique characteristics to be distinguished. More activities of similar movement levels could potentially be added, provided they are distinctive enough. The lower relative movement activities are much harder to discriminate as the movements are too similar. The mislabelling between reading and watching are evident in the Confusion Matrix, and could be explained by the similarity of the tasks themselves. The system performance would be improved by combining these tasks.

Talking was the surprising class, with the worst performance due to a broad scope of movements overlapping with every other class. Most of the tasks were temporally static movements, with similar behaviour across time windows. Some activities such as arithmetic and Rubik's Cube had some dynamic behaviour with different phases of passive thinking (still) and active (entering number and moving respectively). This meant there were some windows of these activities that were mislabelled as the low relative movement categories of reading and watching. Talking activity however, was very dynamic with periods of nodding, listening, turning and laughing which all had very different movements.

The limitations of experimental design are in the construction of tasks. Due to time constraints each participant only repeated each activity twice. As the first instance was the practice one, the data had to be removed as there were too many interruptions of the participant not knowing what they were doing or missing the transition. Thus the analysis was only done on one sitting, and the consistency and repeatability of recognition could not be tested. When a person is repeating an activity they may have different posture, energy levels, attention or environmental factors which may influence their movements. This means the testing and training data for the classification are not truly independent of each other, but should still be useful in predicting the future activities. The real world dataset provides some opportunity for this, but it does not contain all the activities. The other problems were the tasks were specifically selected to be different, rather than what is potentially useful for applications. The laboratory setting removed all distractions, making the signals mostly free of non-activity-related movements.

4.3.2 Cognitive Load

The Cognitive Load was the hardest classification problem of all the ones presented. The hypothesis of detecting unconscious movements, is challenging in picking up small movements beyond the noise of the signals. Qualitatively, there is evidence to support offloading mental capacity through small gestures such as nodding and leaning to the side. The group performance of 75% accuracy (IBK) is promising, with detection of increasing movements, despite the large overlap in the feature space. The problem is these trends are very individualistic and broader trends across the whole dataset are limited. This is reflected in the increased individual performance of 84% (IBK).

While the levels of cognitive load have been validated, the results are easily corrupted by the activity-specific actions. In the preliminary dataset, the participants were using a keyboard to enter the arithmetic results, which resulted in heads moving from the keyboard to the screen with each question. As the frequency of completed questions decreased with increasing load, the results were skewed by this large motion. Isolating the motions caused by the cognitive load is challenging as these movements are on a much smaller scale than general movements (stretching, yawning, etc).

4.3.3 Transitions

The transitions in this database were prompted transitions. The results need to be viewed in this context, as adaptation to the real world may not be as successful. While the subject had practised the order, many transitions were turbulent, as their current activity was abruptly ended, they were prompted by the computer for the next activity, and then had to locate the resources required for the next task. The duration of most of these transitions was between five and ten seconds. However, as most tasks focused on the computer, the posture and position remained mostly static. During the real world dataset, each transition usually entailed a shift in posture and/or head position. The subjects were usually shifting between more stimuli such as mobile phones, paper and the computer.

Another problem for the transition problem is the numerous false positives (up to 38% in Table 4.10), as many common movements, such as stretching and turning the head quickly are indistinguishable from a deliberate change of activities. Any reasonable system would use an integrated task and transition detection as the output of each one informs the other.

4.3.4 Real World

The real world dataset, has a huge variety of activities and movements. The complexity of a real world application becomes quickly apparent. Annotating the load and activities for these tasks is open to interpretation, depending on what is desired.

The results in Figure 4.6, show mixed results. The activity results are low, due to differences in the nature and execution of the activities. For example the reading task was done on a phone resting on the desk. The mislabelling as Rubik's is reasonable given the posture and amount of movement present. The Rubik's Cube and talking were predicted better. Improving system performance in these environments could be achieved with more noise processing, layered classifiers and more environment independent features.

The transitions resulted in a large number of false positives. The actual transitions were marked with most frequent transition predictions, suggesting some success in the system. More training data and integration with activity detection is needed for a feasible system.

Cognitive Load training data was used to characterise the mental performance during the open ended task. Unlike the arithmetic task, no validation can be conducted on the real world activity to truly assess the underlying loads. Again the associated actions of an activity skewed the underlying unconscious movements. However, as the participant was actively engaging in content, Rubik's Cube, talking, etc, the middle level for the majority of the time seems reasonable.

Some test subjects had transitions of the order of a few seconds as they were flicking between their phone and another activity. This multitasking is beyond the scope of the system as it is layered with complicated activities. Any reasonable system would require numerous other input data streams such as screen content, eye movement and brain activity. A promising sign in the majority of transitions involved a shift of posture, usually indicative of a shift of resources (paper, computer, phone, etc). This reaped useful results for analysis.

4.3.5 Subject Variability

From these results, it is worth considering the viability of a common model against a personalised one. While both approaches are on the extremes, there are potential benefits from each scheme. A common model allows for the broader trends to be discovered underneath the noise. The larger dataset can be trained more, utilise more trends to see the underlying phenomenon. However, this does not account for any personal traits and can have poor performance. A purely personalised model will be closely adapted to an individual, requiring more calibration, but potentially better performance. The caveat is the risk of over-fitting on the smaller dataset, and lacking consistency between repetitions. The ideal system would use a combination of the two, with a form of mapping between the individual and the broader dataset.

In the activity results, when the classification had one participant's dataset, the performance was generally 10% better in terms of accuracy. This is caused by the different way people conduct activities in their posture, movement intensity and speed. For the individual case the distributions were close to a normal Gaussian, whereas the combined results had many peaks. Thus, the Bayesian methods were much more effective on a personal classification than combined. Normalising each person's features prior to normalising the entire set improved the Bayesian performance significantly, but not completely.

4.3.6 Comparison With Literature Results

Earlier, Table 2.3 summarised the typical classification results by the classifier methods used. Table 4.12 highlights the detailed results from the studies most closely related to this experiment. The majority explore classifying basic daily activities and locomotion modes. As each experiment is quite differently conducted it is difficult to do a direct comparison and evaluate the best method. However looking over the range of accuracies, the results of this study lie within the range of other studies. This study had more participants than most of the others, allowing better analysis of subject variability. However, the study had a much smaller number of activity classes due to the difficulty of classification problem. The broad range of techniques and accuracies reflect the complicated nature of machine learning, generally the more sophisticated methods are needed over basic Bayesian and Rules based systems. This reflects the overlapping nature of the feature vectors. It is also worth noting that a greater number of activities being discriminated is achieved in the other studies through more sensor signals being available and more distinct physical activities being chosen.

One similar study (not in Table 4.12) was the Opportunity Challenge or recognising physical daily life activities [12]. There were five physical locomotion modes and seventeen gestures or activities. The participants wore 19 different inertial sensors on all parts of body, (except the head). The participant was instructed though a range of activities mostly in a kitchen arrangement. The most successful algorithm was the Nearest Neighbour method for the locomotion problem achieving f-measures of 0.83 to 0.9, compared to f-measures of 0.7 to 0.9 in the results above. The study also considered the problem of a null class, where a large amount of data does not belong in any class. This essentially looks at transitions between tasks, something usually ignored by Human Activity Recognition literature. This improved the f-measures by about 0.05. With the gesture detection the results dropped when excluding the null class to 0.48-0.58, as many borderline decisions were mislabelled. The ROC Area averaged across the participants was 0.8 for the locomotion and 0.85 for the gestures. The opportunity has similar results to the ones presented above (ROC Area between 0.7 and 0.9), but lacks the null class. Inclusion

Table 4.12: Literature Comparison Results for Similar Studies. These are similar daily life task analysis studies. Generally they utilise a large number of inertial sensors on the body to distinguish more activities. The wide range of accuracies and methods reflect the differences in experimental design and feature selection.

Author	**Subjects**	**Activities**	**Sensors**	**Method**	**Accuracy**
Bao (2004) [10]	20	20:Locomotion/Daily Life	five IMU	Decision Tree	84%
				kNN	83%
				Bayes	52%
Ravi (2005) [51]	2	8:Locomotion/Daily Life	one IMU	Bayes	64%
				SVM	63%
				Decision Tree	57%
				kNN	50%
Allen (2006) [6]	6	8:Posture	one IMU	GMM	91%
				Hierachical	71%
Prittikangas (2006) [47]	13	17:Daily Life	three IMU	Neural Network	93%
				kNN	90%
Riboni (2011) [52]	N/A	10: Locomotion/Daily Life	IMU + smart phone	COSAR	93%
Cheng (2010) [15]	N/A	11: Upper Body Activities	electrodes	LDA	77%
Parkka (2006) [46]	16	8: Locomotion/Daily Life	two IMU	Decision Tree	86%
				Hierarchical	82%
				Neural Network	82%

of the null class would reduce system performance somewhat.

In terms of the other classification problems, there are less studies to compare against. A study by Chen (2011) which compares four cognitive load measurements is the most similar study as it uses the same metric for creating cognitive load [14]. For the same three cognitive load level problem, eye activity gave an accuracy of 69%, whereas self reporting - 85%, completion time - 69% and completion accuracy - 51%. Other studies using Electroencephalography (EEG) have much better discrimination, even with seven levels, but are more invasive and computationally intensive [62]. The high results of the cognitive load should be treated with some scepticism, as there were some limitations to the experiment with the frequency of computer activity being characterised rather than the cognitive load. For the indirect metrics (self reporting, precision and performance time), similar results have been achieved.

4.3.7 Real Time Processing Pathway

Given the success of results, it is now possible to explore how this would be implemented in more realistic systems. The major obstacle for applications is getting the predictions quickly to allow dynamic response while the person is still completing the activity. The current system takes about two to five minutes to process the information. The room for improvement is mainly in the selection of machine algorithms and the size of the training data.

The most successful technique was the k-Nearest Neighbour, however the complexity increases as the test point is compared to each training point; $C_{nearestneighbour} \propto N_{test} \times N_{train}$ (where C is complexity and N is the size of the dataset). A large training set which gets good results, requires a large amount of data storage and processing. Instead more appropriate methods would be the Naive Bayesian or Support Vector which builds up a simple model that can be quickly evaluated against new data. The difference is in computational complexity, with Bayesian being linear instead of squared: $C_{Bayes} \propto N_{test} + N_{train}$. Some degree of accuracy will be lost, but it is a compromise that is necessary to be made.

The other area for improvement is reducing the size of the training set. A device will need to be calibrated with a few five second segments of each class or activity. Some general data could be used in the system as well. The cross validation patterns of the experimental analysis above is not necessary. This analysis was used to create the best model possible with the data available, while for applications, separate training and testing data could be utilised. Some dynamic unsupervised learning which builds these models over time could also improve the performance. The unsupervised learning is where the ground truth class is not provided, and the computer tries to work this out itself.

Other processing methods can be utilised to optimise performance. Some of the orientation calculations can be shifted to the IMU device, with its onward conversions of the data to other formats (E.g. Euler Angles). Some of the processing can be shifted to the Arduino Microcontroller. The features can be reduced by feature dimensionality reduction which combines information into fewer dimensions. There may be potential in reducing the sampling rate, as the signal is oversampled for the movements being made.

Given these potential improvements it is possible to process a time segment within 100ms of the completion of the time segment. A loss of accuracy of roughly 5% to 10% is expected, but can be compensated by looking at outcomes over a series of windows. As most application time frames are longer than the windows, a method such as the modal outcome of the last five windows could be adopted.

A basic proof of concept real time system was constructed as part of a demonstration. The system used real time collection of the data into Matlab, from the bluetooth module. The system collected one window worth of data, then conducted processing and feature extraction within Matlab. The classification was done with the Naive Bayesian classifier, which has the model built at start up allowing quick evaluation in real time. The system was able to process the window within 400ms of finishing the time segment. The size of this delay was mostly independent of window size due to Matlab's vector processing capacity. It is worth noting the majority of the delay is in Matlab's handling of the communications rather than the processing.

4.3.8 Application Pathway

Even with the Real World stage of the experiment, the analysis is still a very controlled setting. Applying the technology to a broader context will involve more detailed system design. The main difference between the controlled setting and an application is the noise within the signal. The real world has many other sources of stimulus which create head movements that can interfere with the machine learning. Especially for the cognitive load, it is likely the other movements may cloud the selection.

The range of activities provide some interesting insights into extending this problem to a broader context The mood and temperament of subjects can have a large impact of their behaviour, with a range in the ability to focus on activities. The consistency of these patterns is non-linear, making it difficult to predict those outputs, but they deeply influence the activity outcomes. Boredom and tiredness can also shape a person's behaviour throughout an activity causing dynamic trends. The movements can be affected by the subject's posture, with some positions propping the head on an arm limiting the range of movements. Despite these complications, the results are promising, with discrimination possible. Temporal relationships can be exploited in bigger set-ups as actions can be viewed with their interdependencies on each other. In this way, trends can be built up to view the sub tasks as part of a purpose driven activity with many parts.

More classes of activities could be potentially added provided they are unique. It is necessary for these to have unique characteristics, usually with larger movements. Some potential activities that could be added would be cooking, cleaning, sleeping, While they are more complicated with many phases, they provide distinctive patterns that could be established over time. Other classes would have trouble providing discrimination such as typing, phone conversation and internet browsing. These tasks are too similar to existing tasks.

As the classification will be more challenging, the scope for more information to assist the prediction can come from two avenues. More data sensors such as heart rate, eye movement, sound, cameras, EEG and other sensors could provide insight to activities. While head movement is beneficial for many activities, it is a quite indirect approach, which can get confused by movements. Combining with other sensors covers the weakness of all systems to provide a more complete picture. The other area for potential improvement is collection of large amounts of data over time for the wearer and others. Activity patterns over broader demographics can inform predictions for the individual, especially when the desired outputs are higher level and purpose driven activities. The learning over time allows models to improve and account for shifts in consistency of the subject's performance of a task. Also change in environment may affect how tasks are conducted with factors such as ergonomics, lighting and temperature.

It is envisioned that the device would be incorporated into a set of glasses or clothing. The device would then communicate with a smartphone which can collaborate with the other data sources such as location and share the information through the internet. The smart phone could also be used as the reference IMU device as the chip on the constructed experimental device is equivalent to the ones in modern smart phones. A nice property would be a generic framework in which each user could program the desired predicted classes by training the model with some sample data of themselves. This may be possible provided the tasks are not too similar or too many.

4.3.9 Personal Factors

The demographic factors were found not to impact the system. While age and gender affects the extent of movements possible, that is their flexibility, it does not affect the way they carry out basic activities. The age of the participants was an average of 21.52 ± 4.20. The volunteers had an average height of 173.29 ± 12.23 cm. The experiment consisted of 8 females and 13 males. Subject variability is a more complex phenomenon than just the age, gender and height. The demographic was fairly narrow, of young adults, mostly engineering students. While this should have no effect on their completion of activities, it does skew the survey results somewhat.

4.3.10 Survey Results

During the experiment, a survey was conducted on a range of relevant topics within the context of the technology and the industry. The survey results provide insight in the potential benefits and issues with the particular development. The most interesting statistic is the amount of sedentary time people spend per day (work, education, leisure, but not sleeping) at 9.28 ± 2.43 hours/day. Traditional task analysis done on physical activities is a much smaller component of modern lifestyles. Task recognition for the sedentary tasks has market potential, but requires significant further development.

Wearable Computing

During the study, the participants were surveyed on their thoughts on wearable computing and the breakdown of their average days. Note that as the majority of participants are university students in the young demographic (18-25), the results may be quite skewed and not represent the general public. Given the bias in these results, the results offer insight into some common preliminary thoughts into wearable computing.

Most participants do not own any wearable devices yet, but were open to the concept given it is something that is useful that can fit seamlessly into everyday activities. The bulky devices used in this experiment would not be suitable for everyday use, due to their size and visual presence.

In terms of head tracking, some participants would be happy to have it used in applications, but struggled to see an immediate usage yet. This suggests applications would focus around commercial settings rather than personal leisure ones.

Each participant was asked to rank four factors influencing wearable computing design out of ten. The factors are fashion, comfort, practicality and privacy (Table 4.13). These factors show, that amongst the tech-savvy target market for these devices, they are looking for technology that is efficient for their lives beyond the often gimicky smart watches and fitness bands. Privacy was surprisingly important for the generation which happily signs away data to online services such as Facebook and Google. Moreover, it shows a hesitation to adoption in not knowing what the data can tell about the person and whether it will be used maliciously in commercial advertising, surveillance or personal security. The lack of importance of fashion is more reflective of the participant demographic which were mainly engineers.

Table 4.13: Wearable Computing Factors - Surveyed Participants were most worried about privacy of data collected and the functionality of the device.

Factor	**Importance (1-10)**
Comfort	8.21 ± 0.93
Privacy of Data	9.10 ± 1.34
Fashion	6.79 ± 2.60
Functionality	8.71 ± 1.34

Daily Activities Breakdown

The participants were also asked about the breakdown of their daily activities. These results provide an interesting comparison to the broad studies found in Chapter 2. Again, it is noted that as the demographic is focused mainly on students, they would have a very different lifestyle from the broader population, varying timetables, less domestic duties and different employment and leisure activities.

Nevertheless, the subjects were asked to quantify the amount of time that they are stationary in their daily lives, be it at a desk, during transport or leisure. The average time spent was about nine hours a day, a significant proportion of their waking hours. The employment/education and leisure periods were predominately stationary rendering traditional accelerometer based recognition ineffectual.

The purpose based activity breakdown is compared in Table 4.14. It shows similar patterns, but differences due to the demographic. Also the different means of reporting probably lead to less accurate results in the survey as people are not very good at remembering what they did with accuracy with respect to mundane activities.

Table 4.14: Breakdown of Use of Time Activities [44] Study and Survey Comparison

Activity	**ABS Study (hrs/day)**[44]	**Experimental Survey (hrs/day)**
Personal Care (sleeping, eating, hygiene)	10.95	8.92 ± 1.62
Employment/Education	3.95	8.98 ± 3.46
Domestic/Childcare	2.93	1.04 ± 0.93
Shopping	0.80	0.26 ± 0.68
Social	1.03	2.47 ± 2.00
Recreation Leisure	4.22	2.31 ± 1.47
Other	0.12	0

In addition to this information, they were also quizzed about the lower level tasks that make up the purpose driven activities. These tasks were designed to mostly line up with the ones in the experiment. "Screen time" was recorded as well, as it is often a composite activity between computers, phones and TVs. Figures 4.7 and 4.8 show the average breakdowns of subtasks within these categories of activities. Note the overlap of these activities with the ones tested in experiment. It is interesting to observe the similarity of employment/education and recreation breakdowns, opening the possibility for the system usage in both contexts.

The results show a varied mix between individuals, possibly reflecting the hard nature of remembering how much of these things were actually completed. Often they were taken as guesses, and the fields overlap so it is hard to quantify.

Figure 4.7: Employment Subtasks Survey Results - Note the overlap of these activities and the ones tested in the experiment.

Finally they were asked about their ability to multitask and typical transitions between tasks. This was compared to the real world activity, where they were observed to see how often task transitions occurred. With modern distractions of electronic devices and social media, many participants were prone to multitask or flick between things quickly without much focus. Task transition times varied with some having purpose to move to the next activity or others with thinking time choosing the next activity before transitioning to it.

4.4 Summary

Head tracking classification of daily human activities has been successfully implemented with promising results on a experimental database. The activity classification is the most successful with accuracies as high as 90%. Other interesting problems about transitions,

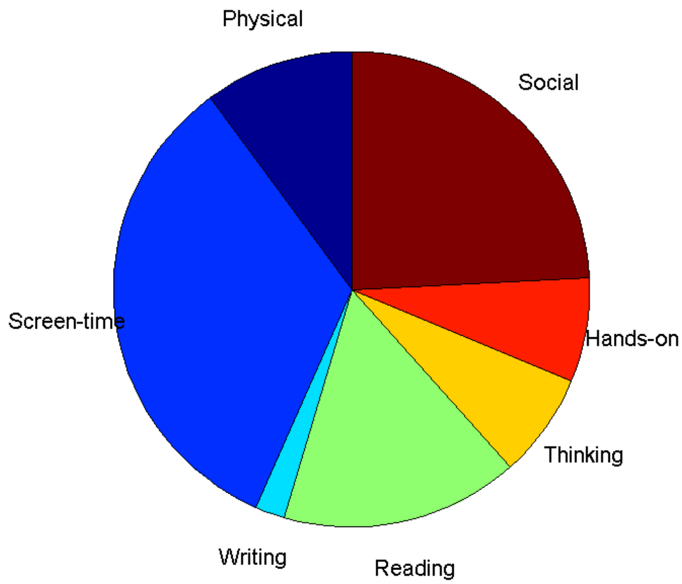


Figure 4.8: Recreation Subtasks Survey Results - Note the overlap of these activities and the ones tested in the experiment. Employment and Recreation had similar breakdowns, opening the possibility for usage in both settings.

cognitive load and real world adaptation have been investigated. The system performs well against similar literature studies, but these comparisons are difficult due to significant differences in the systems. Future improvements have been discussed with reference to real time processing and adapting to real world conditions. The survey results from the experimental participants are discussed exploring the context and potential market for such technology.

Chapter 5

Conclusion

This project has investigated the novel idea of using inertial movement systems in automated task analysis. Chapter 1 explored the societal context of wearable computing and speculated on the future areas of development. Chapter 2 detailed the literature review of existing accelerometer and task analysis research in applying the technology to head movement. The system and experimental design were discussed in Chapter 3 with some preliminary results. Chapter 4 covered the results and discussion about the performance of the system.

5.1 Thesis Contribution

This thesis aimed to assess the feasibility of the technology in activity recognition, having many advantageous properties compared to existing systems. A few different areas have been explored to see the potential in recognising transitions, mental load and the activities themselves. The outcomes of the work include:

- Characterisation of head movement using accelerometry
- Identification of the areas in which the sensors are useful in separating contextual user information
- Analysis and selection of the best features for classification and understanding of the physical meaning behind the features.

- Evaluation of a few classifier algorithms
- Creation of a general dataset for activity recognition from head movement
- Proposal of a pathway towards practical environments and real time processing

This thesis has achieved the desired objectives of the project. The potential of the novel technology has been explored significantly and numerous interesting results and conclusions have been presented. As task analysis with head movement had not been explored previously in the literature, this work has applied existing methods to the new application with success.

5.2 Future Work

Due to the promising results outlined in Chapter 4, there are potential avenues for further research and application of the work. The greatest benefit of the system is in discriminating activities with some small head movement. More work can be developed in the range of activities that can be detected by a system.

The classification algorithms could be improved by incorporating more sensors which can enable more information to be extracted about the activities being conducted. Temporal processing of the relationships could be explored in terms of long term features or state based classifications. A more complicated system architecture combining transition and activity detection would be beneficial as both systems feed back on each other. Subject variability can be improved through personal calibration of the device and possibly by mapping signals into templates to adapt the general model to the individual.

As discussed in Chapter 4, real-time processing is an area for improvement, currently limited by the classification algorithms. The best algorithm in terms of results/complexity appears to be a Bayesian method, which could provide reasonable results within half a second of the event happening. In terms of utilising the device in a more practical environment more signal processing is required to reduce background noise and tailor selection to the desired application.

The device itself could be improved and modified into a smaller package that could fit inside eye-wear or some other wearable device. As discussed in Chapter 3, the second device on the hip is not strictly needed, and could be replaced by a person's smart phone or some other accelerometer on the body.

Task Analysis in wearable computing is a growing field, with rapid development of applications and technology. Commercial devices (not necessarily with inertial systems) for tracking human activities are expected to be available within the next five years. Many of applications are still unknown but there is potential for this technology to be useful in many aspects of human daily life.

Bibliography

[1] Comparison of Feature Classification Algorithm for Activity Recognition Based on Accelerometer and Heart Rate Data.

[2] IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros. *IEEE Std 952-1997*, 1998.

[3] United States. National Aeronautics and Space Administration. *Man-Systems Integration Standards*. Number v. 3 in NASA-STD. National Aeronautics and Space Administration, 1995.

[4] JK Aggarwal and Michael S Ryoo. Human Activity Analysis: A rReview. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[5] David W Aha, Dennis Kibler, and Marc K Albert. Instance-Based Learning Algorithms. *Machine learning*, 6(1):37–66, 1991.

[6] Felicity R Allen, Eliathamby Ambikairajah, Nigel H Lovell, and Branko G Celler. Classification of a Known Sequence of Motions and Postures from Accelerometry Data Using Adapted Gaussian Mixture Models. *Physiological Measurement*, 27(10):935, 2006.

[7] Ranjith Amarasinghe, Dzung Viet Dao, and Susumu Sugiyama. Developing a Wearable System with MEMS Accelerometer for Real-Time Activity Monitoring. *IEEJ Transactions on Sensors and Micromachines*, 129:142–147, 2009.

[8] Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga. Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and

Sports Applications: A Survey. In *Architecture of Computing Systems (ARCS), 2010 23rd International Conference on*, pages 1–10. VDE, 2010.

[9] Jonghun Baek, Geehyuk Lee, Wonbae Park, and Byoung-Ju Yun. Accelerometer Signal Processing for User Activity Detection. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 610–617. Springer, 2004.

[10] Ling Bao and Stephen S Intille. Activity Recognition from User-Annotated Acceleration Data. In *Pervasive computing*, pages 1–17. Springer, 2004.

[11] Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. *A Methodology to use Unknown New Sensors for Activity Recognition by Leveraging Sporadic Interactions with Primitive Sensors and Behavioral Assumptions.* Eidgenössische Technische Hochschule Zürich, D-ITET, Institut für Elektronik, 2010.

[12] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Trster, Jos del R. Milln, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, 2013.

[13] Siyuan Chen and Julien Epps. Blinking: Toward Wearable Computing that Understands your Current Task. *IEEE pervasive computing*, 12(3):56–65, 2013.

[14] Siyuan Chen, Julien Epps, and Fang Chen. A comparison of four methods for cognitive load measurement. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, pages 76–79. ACM, 2011.

[15] Jingyuan Cheng, Oliver Amft, and Paul Lukowicz. Active Capacitive Sensing: Exploring a New Wearable Sensing Modality for Activity Recognition. In *Pervasive Computing*, pages 319–336. Springer, 2010.

[16] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Pattern Classification and Scene Analysis: Pattern Classification. Wiley, 2001.

[17] Stefan M Duma, Sarah J Manoogian, William R Bussone, P Gunnar Brolinson, Mike W Goforth, Jesse J Donnenwerth, Richard M Greenwald, Jeffrey J Chu, and

Joseph J Crisco. Analysis of Real-Time Head Accelerations in Collegiate Football Players. *Clinical Journal of Sport Medicine*, 15(1):3–8, 2005.

[18] Rana El Kaliouby and Peter Robinson. Real-Time Inference of Complex Mental States from Facial Expressions and Head Gestures. In *Real-time vision for human-computer interaction*, pages 181–200. Springer, 2005.

[19] Miikka Ermes, Juha Parkka, Jani Mantyjarvi, and Ilkka Korhonen. Detection of Daily Activities and Sports with Wearable Sensors in Controlled and Uncontrolled Conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20–26, 2008.

[20] Stephen H Fairclough. Fundamentals of Physiological Computing. *Interacting with computers*, 21(1):133–145, 2009.

[21] Eric Foxlin. Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pages 185–194. IEEE, 1996.

[22] Daniele Giansanti, Velio Macellari, Giovanni Maccioni, and Aurelio Cappozzo. Is it Feasible to Reconstruct Body Segment 3-D Position and Orientation using Accelerometric Data? *IEEE Transactions on Biomedical Engineering*, 50(4):476–483, 2003.

[23] Space Glasses. Meta: Augmented Reality Glasses with Head Tracking. `https://www.spaceglasses.com/technology`, Accessed: 2014-06-01.

[24] Helena Grip. Biomechanical Assessment of Head and Neck Movements in Neck Pain Using 3D Movement Analysis. *Straa lningsvetenskaper*, 2008.

[25] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[26] Mark A Hall. *Correlation-Based Feature Selection for Machine Learning*. PhD thesis, The University of Waikato, 1999.

[27] Daniel S Hamermesh, Harley Frazis, and Jay Stewart. Data Watch: The American Time Use Survey. *The Journal of Economic Perspectives*, 19(1):221–232, 2005.

[28] Jihun Hamm, Benjamin Stone, Mikhail Belkin, and Simon Dennis. Automatic Annotation of Daily Activity from Smartphone-Based Multisensory Streams. In *Mobile Computing, Applications, and Services*, pages 328–342. Springer, 2013.

[29] James A Hanley and Barbara J McNeil. The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143(1):29–36, 1982.

[30] Robert C Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993.

[31] Tâm Huynh and Bernt Schiele. Analyzing Features for Activity Recognition. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies*, pages 159–163. ACM, 2005.

[32] Ronny Kurniawan Ibrahim. *Novel Gait Models and Features for Gait Patterns Classification*. PhD thesis, The University of New South Wales, 2011.

[33] Azadeh Jamalian, Valeria Giardino, and Barbara Tversky. Gestures For Thinking. In *Proceedings of the Cognitive Science Society Meetings*, 2013.

[34] Nicky Kern, Bernt Schiele, Holger Junker, Paul Lukowicz, and Gerhard Tröster. Wearable Sensing to Annotate Meeting Recordings. *Personal and Ubiquitous Computing*, 7(5):263–274, 2003.

[35] Oscar D Lara and Miguel A Labrador. A Survey on Human Activity Recognition using Wearable Sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209, 2013.

[36] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A Hybrid Discriminative/Generative Approach for Modeling Human Activities. In *IJCAI*, volume 5, pages 766–772, 2005.

[37] Anant Madabhushi and JK Aggarwal. Using Head Movement to Recognize Activity. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 698–701. IEEE, 2000.

[38] Sebastian OH Madgwick, Andrew JL Harrison, Paul M Sharkey, Ravi Vaidyanathan, and William S Harwin. Measuring Motion with Kinematically Redundant Accelerometer Arrays: Theory, Simulation and Implementation. *Mechatronics*, 23(5):518–529, 2013.

[39] Merryn Mathie. *Monitoring and interpreting human movement patterns using a triaxial accelerometer*. PhD thesis, The University of New South Wales, 2003.

[40] David Maunder, Julien Epps, Eliathamby Ambikairajah, and Branko Celler. Robust Sounds of Activities of Daily Living Classification in Two-Channel Audio-Based Telemonitoring. *International journal of telemedicine and applications*, 2013:3, 2013.

[41] Sushmita Mitra and Tinku Acharya. Gesture Recognition: A Survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, 2007.

[42] Long-Van Nguyen-Dinh, Daniel Roggen, Alberto Calatroni, and Gerhard Troster. Improving Online Gesture Recognition with Template Matching Methods in Accelerometer Data. In *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*, pages 831–836. IEEE, 2012.

[43] Norbert Noury, A Galay, J Pasquier, and M Ballussaud. Preliminary Investigation into the Use of Autonomous Fall Detectors. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 2828–2831. IEEE, 2008.

[44] Australian Bureau of Statistics (ABS). How Australians Use Their Time, 2006.

[45] Maja Pantic, Alex Pentland, Anton Nijholt, and Thomas S Huang. Human Computing and Machine Understanding of Human Behavior: a Survey. In *Artifical Intelligence for Human Computing*, pages 47–71. Springer, 2007.

[46] Juha Parkka, Miikka Ermes, Panu Korpipaa, Jani Mantyjarvi, Johannes Peltola, and Ilkka Korhonen. Activity Classification Using Realistic Data from Wearable Sensors. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):119–128, 2006.

[47] Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. Feature Selection and Activity Recognition from Wearable Sensors. In *Ubiquitous Computing Systems*, pages 516–527. Springer, 2006.

[48] John C Platt. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In *Advances in kernel methods*, pages 185–208. MIT Press, 1999.

[49] Stephen J Preece, John Y Goulermas, Laurence PJ Kenney, Dave Howard, Kenneth Meijer, and Robin Crompton. Activity Identification Using Body-Mounted Sensors a Review of Classification Techniques. *Physiological Measurement*, 30(4):R1, 2009.

[50] John Ross Quinlan. *C4. 5: Programs for Machine Learning*, volume 1. Morgan Kaufmann, 1993.

[51] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. Activity Recognition from Accelerometer Data. In *AAAI*, volume 5, pages 1541–1546, 2005.

[52] Daniele Riboni and Claudio Bettini. COSAR: Hybrid Reasoning for Context-Aware Activity Recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[53] DA Rodríguez-Silva, F Gil-Castiñeira, FJ González-Castaño, RJ Duro, F Lopez-Pena, and J Vales-Alonso. Human Motion Tracking and Gait Analysis: A Brief Review of Current Sensing Systems and Integration with Intelligent Environments. In *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2008. VECIMS 2008. IEEE Conference on*, pages 166–171. IEEE, 2008.

[54] Hesam Sagha, Alberto Calatroni, Jose del R Millan, Daniel Roggen, Gerhard Troster, and Ricardo Chavarriaga. Robust Activity Recognition Combining Anomaly Detection and Classifier Retraining. In *Body Sensor Networks (BSN), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[55] Leon Straker, Amity Campbell, Svend Erik Mathiassen, Rebecca Anne Abbott, Sharon Parry, and Paul Davey. Capturing the Pattern of Physical Activity and Sedentary Behavior: Exposure Variation Analysis of Accelerometer Data. *Journal of Pysical Activity & Health*, 11(3):614–625, 2014.

[56] Sofia Suvorova, Tharshan Vaithianathan, and Terry Caelli. Action Trajectory Reconstruction from Inertial Sensor Measurements. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pages 989–994. IEEE, 2012.

[57] Henry CC Tan and Liyanage C De Silva. Human Activity Recognition by Head Movement Using Elman Network and Neuro-Markovian Hybrids. In *Proceedings of Image and Vision Computing New Zealand*, pages 320–326, 2003.

[58] Jorge Usabiaga, George Bebis, Ali Erol, Mircea Nicolescu, and Monica Nicolescu. Recognizing Simple Human Actions Using 3D Head Movement. *Computational Intelligence*, 23(4):484–496, 2007.

[59] Occulus VR. Occulus Rift: Augmented Reality Headwear. `http://www.oculusvr.com/rift/`, Accessed: 2014-06-01.

[60] Oliver J Woodman. An Introduction to Inertial Navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14:15, 2007.

[61] Mitchell Yuwono, AM Ardi Handojoseno, and HT Nguyen. Optimization of Head Movement Recognition Using Augmented Radial Basis Function Neural Network. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 2776–2779. IEEE, 2011.

[62] Pega Zarjam, Julien Epps, and Fang Chen. Characterizing Working Memory Load Using EEG Delta Activity. In *Proceedings of the 19th European Signal Processing Conference EUSIPCO*, pages 1554–1558, 2011.

[63] Pega Zarjam, Julien Epps, and Nigel H Lovell. Characterizing Mental Load in an Arithmetic Task Using Entropy-Based Features. In *Information Science, Signal Pro-*

cessing and their Applications (ISSPA), 2012 11th International Conference on, pages 199–204. IEEE, 2012.

[64] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):39–58, 2009.

Appendix A - Device Construction

The development of the prototype is briefly detailed in this section. As the thesis is not directly concerned with the device, a quick prototype (see Figure 3.2) was developed relying heavily on sensor breakout boards and existing libraries. In a practical system the components could easily be designed into a much smaller package such as the Space Glasses [23], which have equivalent head tracking hardware. For details of the costing and parts (roughly $80/unit), refer to Appendix E.

The ideal system for detecting head movement had the following requirements:

- High resolution in detecting different types of head movement
- Non intrusive system potentially applicable for everyday usage
- Capacity for real time processing
- Simple and cost effective for widespread use
- Distinguish head movement from the rest of the body

Sensor Selection

As discussed in Chapter 2, an IMU device has been chosen with accelerometers, gyroscopes and magnetometers. The factors affecting the sensor choice included data rate, data resolution, data interface, power usage, number of axes and the size/weight of the device. There were many small and inexpensive sensors that were capable of the above requirements. The MPU-9150 by Invensense was selected for the availability of on-line libraries, (see Appendix C for details about the Arduino code) and its capability for on-board processing. The device is very similar to the IMUs found in many modern smart phones. The device has “nine degrees of freedom” which means it has three axes of accelerometer, gyroscope and magnetometer. The device has a built-in ADC converter, and digital communications with the I2C protocol giving adequate signal resolution and range for this application. The device is inexpensive and has a very small footprint as seen in Figure 3.2.

Microcontroller

The Arduino platform was chosen for its inbuilt libraries and the low cost, as the focus of the thesis is not in the hardware design. The Pro Micro board was chosen for its small profile (Figure 3.2), low power consumption and its plug and play capacity with various digital protocols. Refer to Appendix D for information about the code and libraries.

Wireless Communications

To transfer the data between the device and the processing computer, there are a few communication options in wired, unwired or on-board. To make the system as practical and non-invasive as possible, wireless communication is preferred. For the application, some of the potential protocols include Wifi, RF and Bluetooth. Bluetooth was chosen for its low power usage, small scale (see Figure 3.2) and flexibility in connection, potentially with smart phones for later implementations. With a bluetooth USB dongle on the computer, a range of 30m was easily achieved. Figure 5.1 demonstrates a smart phone receiving data from the device using a simple Bluetooth terminal application.

Battery

To make the system completely wireless, a battery was added. Instead of bulky conventional alkaline batteries, a higher energy density Lithium Ion Polymer battery was chosen which has 400mAhr at 3.7V in a small profile ($5 \times 25 \times 35$ mm). This should give the device roughly a ten hour life, nearly spanning a whole day of activities.

Figure 5.1: Smartphone Bluetooth Screenshot - Each line is a packet of data with a timestamp and the nine unscaled sensor values

Appendix B - Representation Of Signals

The signals from the IMU device are 16 bit raw unscaled values for each axis of the sensor. A system of coordinates is required to model the rotation in space. These coordinates are needed when removing the gravitational component of signals. The most common coordinate systems are rotation matrices, euler angles and quaternions. For the purposes of this section (ϕ, θ, ψ), are defined as the rotations around the (x, y, z) axes. The raw signals are output as rotations around each of the axes (x, y, z).

Rotation matrices are used to multiply a Cartesian vector to alter the direction while maintaining the magnitude. The equations below show the matrix for rotating a vector around one axis (Equation 5.1), with the product being the total rotation (Equation 5.2).

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} R_z = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.1)$$

$$R = R_x R_y R_z \quad (5.2)$$

Euler angles represent a series of three rotations and are commonly referred to as yaw, pitch and roll. They are dependent on the order of rotation. However, they suffer from gimbal lock, in special cases where the orientation is not correct due to singularities.

Quaternions are a mathematical construct with one real number and three complex parts (Figure 2.4 and Equation 5.3). The orientation is expressed with a direction (e_x, e_y, e_z) and a rotation around direction θ. Unlike Euler angles they do not suffer from gimbal lock. All rotations can be computed using simple matrix operations, except that the order matters. The components of a quaternion are:

$$\left(\cos(\tfrac{\theta}{2}), e_x \sin(\tfrac{\theta}{2}), e_y \sin(\tfrac{\theta}{2}), e_z \sin(\tfrac{\theta}{2})\right) \tag{5.3}$$

Quaternions have been used for this project, as they are the most efficient in quantifying a rotation in terms of memory and multiplications required.

Appendix C - Arduino Code

As discussed in Chapter 3, the prototype was developed on the Arduino platform in order to leverage existing libraries and code bases. The micro-controller code consists of four levels:

- the application software - prints data to bluetooth
- the sensor library - hardware specific implementations
- the I2C protocol library - reads and writes to registers using I2C protocol
- the Arduino codebase

Application Software The application layer was modified by the author from existing code by Jeff Rowberg (`https://github.com/jrowberg`). The file calls the functions from the MPU-9150 library and prints the information over the USB or bluetooth communications channels. The application software file is available at: (`https://www.dropbox.com/s/6fgxjy24ni8ih3e/Accel.ino`).

Sensor Library The sensor library was developed by Jeff Rowberg for the sister sensor MPU-6050. The code is publicly available at: (`https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050)`. The code was slightly modified by Sparkfun for usage with the MPU-9150 and is publicly available at: (`https://github.com/sparkfun/MPU-9150_Breakout/tree/master/firmware`).

I2C Library The library for I2C protocol was again developed by Jeff Rowberg and is available at: (`https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/I2Cdev`).

Arduino The code was developed in Arduino 1.0.5 which is available at: (`http://arduino.cc/en/Main/Software`).

Appendix D - Matlab Code

The Matlab Code implemented is broken into the following categories. Please contact the author directly for copies of the code:

1. **Input Processing** - reads in data from bluetooth and saves to a database. Real-time plotting of data allows for immediate observation of data.

2. **Database** - the data is stored in Matlab variable files, which contains the 9 channels of data and timestamps for each unit.

3. **Filtering** - The signals are median and complementary digital filtered to remove the noise of signals.

4. **Sensor Fusion** - The orientation is computed using the gyroscope readings corrected by the accelerometer and compass.

5. **Tracking** - Using the corrected orientation, the head tracking outputs (eg. linear acceleration) are calculated.

6. **Segmentation** - The data is windowed in smaller chunks for feature processing. These segments may be of variable length to capture different motion events. Event detection may be utilised to define the windows.

7. **Class Annotation** - The data is annotated with an outcome for each segment. Video analysis side-by-side with the raw data allows the outcomes to be assigned to each window.

8. **Feature Selection** - For each segment a variety of windows is computed and are plotted in a histogram for each class outcome. Visual analysis of each histogram indicates which should be used.

9. **Feature Vector** - A feature vector is generated for each segment, normalizing the numerical values and including (if needed) temporal features. The output of these vectors is fed into the classifiers.

10. **Normalisation and Balancing** - Z Score normalisation to scale feature values to the same distributions for use in the distance between different features. There are also methods implemented for sub-sampling and randomising the number of data instances to be balanced between classes and participants.

11. **Classifier Model** - Some classifier methods are implemented in Matlab to validate the Weka results. These methods include ZeroR, oneR, J48, Naive Bayes and Nearest Neighbour. The model is generated from the training data and takes different forms depending on the type of classifier.

12. **Classifier Training** - This module controls the dataset, feeding some datasets to train the classifier and others to test the classifier. This will use cross validation to process which data is used for training and which is used for testing.

13. **Classifier Evaluation** - The purpose of this code is to output the metrics required to assess the performance of the classifier. Graphic visualisations have been developed to help communicate the results.

Appendix E - Bill Of Materials

Table 5.1: Bill of Materials

Item	Part Number	Manufacturer	Cost	Source
IMU	MPU-9150 breakout board	Invensense/ Sparkfun	\$35	`https://www.sparkfun.com/products/11486`
Microcontroller	Arduino Pro Micro 3v3	Sparkfun	\$20	`https://www.sparkfun.com/products/12587`
Bluetooth Module	HC06	CREATE UNSW	\$10	`http://www.createunsw.com.au/partSale.php`
Battery	Polymer Lithium Ion Battery - 400mAh	Sparkfun	\$7	`https://www.sparkfun.com/products/10718`
Battery Charger	LiPo Charger Basic - Micro-USB	Sparkfun	\$8	`https://www.sparkfun.com/products/10718`
Veriboard	prototyping board	CREATE UNSW	\$2	`http://www.createunsw.com.au/partSale.php`
Total			**\$82**	

Appendix F - Ethics Approval

THE UNIVERSITY OF
NEW SOUTH WALES

HUMAN RESEARCH ETHICS ADVISORY
PANEL 'H'
SCIENCE & ENGINEERING

24 April 2014

A/Prof. Julien Epps
School of Electrical Engineering and Telecommunications

Re: **Automatic task analysis for wearable computing**

Reference Number: 08/2014/23

Dear A/Prof Epps

At its meeting of 15th April the Human Research Ethics Advisory Panel 'H' reviewed the protocol and requested clarification of the following issue:

- Please clarify the nature and value of the incentive in the PISC form and poster.

Please respond by email or letter.

Yours sincerely

Professor Mike Regan
Panel Convenor
Human Research Ethics Advisory Panel 'H'

Cc: Head of School
Prof Eliathamby Ambikairajah
School of Electrical Engineering and Telecommunications

* http://www.nhmrc.gov.au

SYDNEY 2052 AUSTRALIA
Telephone: +61 (2) 9385 5774
Facsimile: +61 (2) 9385 6040
Email: m.regan@unsw.edu.au

Ethics Compliance The following measures were taken to ensure compliance with the Ethics Approval.

- The participants were rewarded with a movie voucher for their time, usually about one hour
- The participants signed a consent form at the beginning of the experiment after receiving a briefing about what they were doing.
- Video/Sound recording for annotation purposes were only conducted if the participant consented
- The data stored only contained their participant number (none of the data can be personally identified)
- All the activities were not physically or emotionally demanding. The participants were given detailed explanations of what they were required to do and given practice runs to familiarise themselves with the tasks.

Appendix G - Risk Assessment

HS017

HS Risk management form

For additional information refer to HS329 Risk Management Procedure

Faculty/Division: Faculty of Engineering			School/Unit: School of Electrical Engineering & Telecommunications	
Document number: HS_ENG_001	Initial Issue date: 06/04/14	Current version: 1	Current Version Issue date: 06/04/14	Next review date: 01/01/15

Risk management name	**Honours Thesis – Experiments and General Work**

Form completed by	Robert Makepeace (Honours Students)	*Signature*	*Date*
Responsible supervisor/ authorising officer	A/Prof Julien Epps (Supervisor)	*Signature*	*Date*

Identify the activity and the location of the activity

Description of activity	**1) Experiment of research participants: People will be asked to undertake some activities that occur in their daily life with some sensors attached on/near their bodies** **2) Student will undertake construction of hardware sensors.**
Description of location	N/A

Identify who may be at risk from the activity:

This may include fellow workers, visitors, contractors and the public. The types of people may affect the risk controls needed and the location may affect the number of people at risk

Persons at risk	Student, Research Participants
How they were consulted on the risk	Briefed before activity commenced

List legislation, standards, codes of practice, manufacturer's guidance etc used to determine control measures necessary

Work Health and Safety Act 2011
Work Health and Safety Regulation 2011

Identify hazards and control the risks.
1. An activity may be divided into tasks. For each task identify the hazards and associated risks. Also list the possible scenarios which could sooner or later cause harm.
2. Determine controls necessary based on legislation, codes of practice, Australian standards, manufacturer's instructions etc.
3. List existing risk controls and any additional controls that need to be implemented
4. Rate the risk once all controls are in place using the matrix in HS329 Risk Management Procedure

SHADED GREY AREAS
If you need to determine whether it's reasonably practicable to implement a control, based on the risk complete the shaded grey columns

Feel free to resize the boxes to suit your situation/the amount of text you need to use

Task/ Scenario	*Hazard*	*Associated harm*	*Existing controls*	*Any additional controls required?*	*Risk Rating*			*Cost of controls (in terms of time, effort, money)*	*Is this reasonably practicable* Y/N
					C	*L*	*R*		
Ergonomics	Bad Posture	Back/Neck Injuries	• Workstation setup following ergonomic guidelines • Regular breaks (involving going for a quick walk)	N/A	*2*	*D*	*L*		
Construction	Soldering Hand Tools	Hand Injuries/burns Injuries	• Appropriate personal protective equipment is worn (PPE) • Soldering is conducted in a well ventilated area	N/A	*3*	*D*	*M*		
Experiment	Movement	Personal Injuries	• Participants will be instructed about tasks before commencing • Participants will be able to leave experiment at any time if they feel uncomfortable • Tasks will be designed to avoid physical injuries	N/A	*1*	*E*	*L*		
			•						

List emergency procedures and controls
List emergency controls for how to deal with fires, spills or exposure to hazardous substances and/or emergency shutdown procedures

Standard university emergency procedures

Implementation

Additional control measures needed:	Resources required	Responsible person	Date of implementation
N/A			

REVIEW

Scheduled review date:			
Are all control measures in place?			
Are controls eliminating or minimising the risk?			
Are there any new problems with the risk?			
Review by: (name)			
Review date:			

Acknowledgement of Understanding

All persons performing these tasks must sign that they have read and understood the risk management (as described in HS329 Risk Management Procedure).

Note: for activities which are low risk or include a large group of people (e.g. open days, BBQ's, student classes etc), only the persons undertaking the key activities need to sign below. For all others involved in such activities, the information can be covered by other methods including for example a safety briefing, induction, and/or safety information sheet (ensure the method of communicating this information is specified here)

Risk management name and version number: **I have read and understand this risk management form**

Name	Signature	Date