Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage.
http://researchspace.auckland.ac.nz/feedback

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form and Deposit Licence.
Optimization of Electrode Placement in Electromyographic Prosthetic Control

By Scott Walbran

Supervised by:
Associate Professor Iain A. Anderson, Dr. Emilio P. Calius and Dr. G. Reg Dunlop

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at The University of Auckland

The Biomimetics Laboratory
Auckland Bioengineering Institute
The University of Auckland
New Zealand

February 2013
Abstract

Exciting new approaches have led to a great increase in the ability of prosthetic devices to accurately replicate the functionality of the human hand. Advanced methods of neural interfacing exist, however they are expensive and not commonly available. Electromyographic (EMG) pattern recognition techniques exist to allow determination of user intention, however they rely on prior knowledge of muscle groups, use a symmetrical array to place electrodes or a simple reduction algorithm, and often require large amounts of data to accurately classify user intention.

The aim of this project was to provide improved accuracy of myoelectric prosthetic classification system, without inhibiting its ability to operate in real-time. In order to be able to investigate the affect of various parameters, rich data sets were acquired from the forearm of 10 subjects, through creation of a silicone armband with embedded electrodes. Current methods have had success using a statistical cluster analysis measure on a non-amputee subject, with an eye on future applications to real world prosthetic wearers. This thesis attempts to improve prosthetic control accuracy through the attainment of three goals.

First, a real-time capable preprocessing step was performed. A method has been presented to extract action potentials from forearm surface EMG signals, and to have this extracted action potential used in a matched filtering approach. The algorithm was able to identify action potential signals, with average positive predictive value (number of true positives as a percent of total positive results) of 97%. Total processing time involved for this method was less than 10% the length of a signal segment, proving that it is able to be used in real-time applications. This increased the signal-to-noise ratio by, on average, 2.2 dB for data from 10 subjects. Further analysis however showed that this technique was not helpful in improving accuracy in prosthetic control, and due to success with other methods it was not pursued further.

The second goal was to provide a means to choose optimal data sets from a larger, irregularly spaced grid of data. This was realized through an interpolation scheme followed by a heuristic data set replacement algorithm. When tested on a reduced number of data sets, this algorithm performed to within 99.9% of an exhaustive algorithm, while taking $1 \times 10^{-5}$ of the time.
Finally, these methods were applied to forearm surface EMG signals in order to compare the performance of the optimized sites to symmetrically placed sites, and two previously used location selection algorithms. For the data processed off-line, this showed a great increase in classification accuracy from 85% to approximately 95%.
Acknowledgements

There are many people and groups who have been important in making this work happen. Firstly, I would like to extend thanks to The University of Auckland for my University of Auckland Doctoral Scholarship. My thanks also go to the ABI for hosting me, their support and assistance in all administrative and technical needs and for providing a fantastic environment and brilliant colleagues. Also, the Biomimetics Laboratory where I carried out the majority of my research has been a wonderful place to work.

To my principle motivators: Iain Anderson, Emilio Calius and Reg Dunlop. For ensuring I keep on track (at least, most of the time), for understanding when things haven’t been going so well and for your enthusiastic encouragement when they have been. Special thanks to Reg for the invaluable experience you have been able to add.

Several institutions have provided me with equipment and knowledge it would have been impossible to complete this thesis without. My thanks go out to the Auckland Artificial Limb Centre and especially to John Brookes for your insights into myoelectric prostheses, and to UnEmap and David Budgett for the ability to acquire all the data required and the countless hours in helping me set up.

My colleagues in the Biomimetics Laboratory: Ben O’Brien, Todd Gisby and Thomas McKay. Your ability to tolerate my incessant chatter and odd sense of humour is worthy of a medal. Though, I won’t be getting you one. My thanks go out to you for being brilliant researchers, assisting with work when required, letting me assist you with yours when it looked interesting enough and for all the late nights you have put into making our Laboratory the success that it has become.

The many exchange students, summer students, final year project students and other postgraduate students who have shared in the hard times and the good; David Van Berkel, Ben Lynch, Anita McKenzie, Michaela Herzer, Gabriel Loh, Paul Bomke, Andreas Rick, Andrew Lo, Casey Jowers, Samuel Schlatter, Tony Tse, Tessa Paris, Anne Kikker, Antoni Harbuz, Mahdieh Nejati and Jae Jae Kim. Tom Hale for your interesting talks about cars when nobody else would listen. Pete Blythe, Sharif Malak and Michael Byrne for your technical expertise. Peter Hunter and Bruce Smaill for taking care of this wonderful Institute and bringing it to where it is today. Poul Nielsen for the advice you have provided over the years. Our IT team: Weni Fernandez, Peter Schmeideskamp,
Andrew Cantell, Tiong Lim and Matt Wilson. Our admin team: Mary Grigor, Kate Palmer, Carmen Balanon, Maria Fung, Suman Nath, Nirosha Heart, Mohini Singh and Lyn Vu.

To Anita McKenzie for all your support, particularly when things didn’t go well. To my family for making me the person I am today and your unending support during this time.

Enfin, à Camille Regnery pour avoir toujours été là pour moi. Je n’aurais pas pu terminer sans toi, merci beaucoup.
# Glossary of terms

**ABI**  
*Acronym, Auckland Bioengineering Institute.*

** ACTION POTENTIAL**  
The basic unit of bioelectric signalling.

**ALS**  
*Acronym, Amyotrophic Lateral Sclerosis.* A degenerative motor neuron disease that eventually leads to loss of motor control.

**ANTERIOR**  
A reference to anatomical position. Considered to be the “front” of the body.

**CLASSIFIER**  
An algorithm that takes the output(s) of a feature extraction algorithm and converts these into predetermined groups.

**CMRR**  
*Acronym, Common Mode Rejection Ratio.* The ability of an instrumentation amplifier to reject signals that are common to both input terminals.

**COVARIANCE**  
The variation of one data set with another. A positive covariance means that one data set will, in general, increase with the second data set, while a negative covariance means that one data set will, in general, decrease with the second data set.

**DISTAL**  
A reference to anatomical position. In general, refers to a position furthest from the body.

**DNI**  
*Acronym, Direct Neural Interfacing.* A process where information is taken directly from the nervous system.

**DOF**  
*Acronym, Degrees of freedom.*

**EAP**  
*Acronym, ElectroActive Polymer.* Polymers that exhibit a change in shape and/or size due to an electric field.

**ECG**  
*Acronym, ElectroCardioGraphy.* Bioelectrical signals recorded due to cardiac muscle groups firing.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEG</td>
<td>Acronym, ElectroEncephaloGraphy. Bioelectrical signals recorded due to muscle groups firing.</td>
</tr>
<tr>
<td>EMG</td>
<td>Acronym, ElectroMyoGraphy. Bioelectrical signals recorded due to skeletal muscle groups firing.</td>
</tr>
<tr>
<td>Euclidean Distance</td>
<td>“Ordinary” distance between two points. The distance between two points we would consider in everyday life.</td>
</tr>
<tr>
<td>Fuzzy C-Means Clustering</td>
<td>A form of cluster analysis where each point has a certain degree of belonging to a cluster, rather than belonging solely to one cluster. In c-means clustering, the centre of each cluster is taken as the mean of all points, weighted by their degree of belonging to that cluster.</td>
</tr>
<tr>
<td>HCI</td>
<td>Acronym, Human Computer Interfacing. Any form of providing a means for a person to interact with a computer.</td>
</tr>
<tr>
<td>HIR</td>
<td>Acronym, Human Intention Recognition. The ability to detect the intention of a user.</td>
</tr>
<tr>
<td>LDA</td>
<td>Acronym, Linear Discriminant Analysis. A pattern recognition technique that involves finding linear combinations of features to separate data into classes.</td>
</tr>
<tr>
<td>Mahalanobis Distance</td>
<td>A distance measure between a cluster of data points and another data point. In a general sense, the Mahalanobis distance can be thought of as the distance between the centre of the cluster of data and the point, divided by the standard deviation of the cluster in the direction of the point.</td>
</tr>
<tr>
<td>MLP</td>
<td>Acronym, MultiLayer Perceptron. A form of neural network.</td>
</tr>
<tr>
<td>MUAP</td>
<td>Acronym, Motor Unit Action Potential. The action potential as caused by muscle cells depolarizing.</td>
</tr>
</tbody>
</table>
| MV           | Acronym, Majority Voting. An algorithm whereby the output for a given number of data points is assumed to be the most commonly
occuring value within that set of points.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYOELECTRIC</td>
<td>A prosthetic device that uses electromyographic signals as a control source from the body.</td>
</tr>
<tr>
<td>NN</td>
<td>Acronym, Neural Network or Artificial Neural Network. A learning algorithm that can adapt the way it processes its inputs in order to better match the required outputs.</td>
</tr>
<tr>
<td>OEPEPC</td>
<td>Acronym, Optimization of Electrode Placement in Electromyographic Prosthetic Control. The name of this thesis.</td>
</tr>
<tr>
<td>PCA</td>
<td>Acronym, Principal Component Analysis. A useful tool in cluster analysis, involving reduction of large numbers of data sets down into data sets that best describe the underlying variation of the data sets.</td>
</tr>
<tr>
<td>POSTERIOR</td>
<td>A reference to anatomical position. Considered to be the “back” of the body.</td>
</tr>
<tr>
<td>PROSTHETIC</td>
<td>An artificial device used to augment the human body.</td>
</tr>
<tr>
<td>PROXIMAL</td>
<td>A reference to anatomical position. In general, refers to a position closest to the body.</td>
</tr>
<tr>
<td>sEMG</td>
<td>Acronym, surface EMG. EMG signals as recorded on the surface of the skin, rather than within the muscles.</td>
</tr>
<tr>
<td>UNEMAP</td>
<td>Hardware developed at the University of Auckland for the amplification, filtering and measurement of biopotential signals.</td>
</tr>
</tbody>
</table>
# Table of Contents

## Contents

### 1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION ................................................................. 1

1.2 HUMAN INTENTION RECOGNITION AND HUMAN COMPUTER INTERFACING (HCI) ................................................................. 2

1.2.1 Electroconductive (EMG) Human Computer Interface ........................................ 3

1.2.2 Electroencephalographic Human Computer Interfacing ......................................... 4

1.2.3 Superconducting quantum interference devices .................................................. 5

1.2.4 Direct neural interfacing .................................................................................. 6

1.2.5 Summary of signal acquisition techniques, and motivation for using Electromyography.. 10

1.3 EMG BASED PROSTHETIC CONTROL .......................................................... 12

1.3.1 Data acquisition ......................................................................................... 13

1.3.2 Electrode locations .................................................................................... 15

1.3.3 Signal preprocessing .................................................................................. 15

1.3.4 Feature extraction ...................................................................................... 16

1.3.5 Classification .............................................................................................. 17

1.4 RESEARCH OBJECTIVES ............................................................................... 20

1.5 THESIS OUTLINE AND CONTRIBUTIONS .................................................. 22

1.6 SUMMARY ................................................................................................... 23

### 2 LITERATURE REVIEW

2.1 PREVIOUS sEMG APPLICATIONS TO HCI....................................................... 25

2.1.1 Majority voting .......................................................................................... 26

2.2 HAND MOVEMENT RECOGNITION .............................................................. 29

2.3 ELECTRODE LOCATIONS ............................................................................ 30

2.4 SIGNAL PREPROCESSING AND DECOMPOSITION ........................................ 31

2.5 FEATURE EXTRACTION TECHNIQUES .......................................................... 38

2.6 CLASSIFICATION METHODS .......................................................................... 43

2.6.1 Neural networks ......................................................................................... 46

2.6.2 Fuzzy approach ......................................................................................... 48

2.6.3 Fuzzy neural classification ......................................................................... 48

2.7 DATA REDUCTION ALGORITHMS .................................................................. 50

2.8 SUMMARY ................................................................................................... 53

### 3 ACTION POTENTIAL EXTRACTION THROUGH ITERATIVE FILTERING

SUMMARY ............................................................................................................. 55

3.1 INTRODUCTION ............................................................................................ 55

3.2 DATA ACQUISITION ..................................................................................... 57

3.3 TEMPLATE EXTRACTION .............................................................................. 60

3.3.1 Methods .................................................................................................... 64

3.3.2 Results and Discussion .............................................................................. 65

3.4 APPLICATION TO SIMULATED DATA ................................................................ 68

3.4.1 Methods .................................................................................................... 68

3.4.2 Results and Discussion .............................................................................. 69

3.5 APPLICATION TO REAL DATA ......................................................................... 70

3.5.1 Methods .................................................................................................... 70

3.5.2 Results and Discussion .............................................................................. 71

3.6 APPLICATION TO PROSTHETIC CONTROL .................................................. 74

3.6.1 Methods .................................................................................................... 74

3.6.2 Results and Discussion .............................................................................. 75

3.7 CONCLUSIONS ............................................................................................. 80

### 4 OPTIMIZATION OF ELECTRODE PLACEMENT FOR ELECTROMYOGRAPHIC PROSTHETIC CONTROL ........ 83
SUMMARY ........................................................................................................................................... 83
4.1 INTRODUCTION .......................................................................................................................... 84
4.2 FEATURE EXTRACTION .................................................................................................................. 87
4.3 FEATURE RANKING OPTIMIZATION ............................................................................................ 88
4.3.1. Methods ...................................................................................................................................... 88
4.3.2. Results ........................................................................................................................................ 89
4.3.3. Discussion ................................................................................................................................. 93
4.4 MEAN-DIFFERENCE RANKED OPTIMIZATION ............................................................................ 94
4.4.1. Methods ...................................................................................................................................... 94
4.4.2. Results ........................................................................................................................................ 95
4.4.3. Discussion ................................................................................................................................. 96
4.5 MAHALANOBISS DISTANCE MAXIMIZATION OPTIMIZATION ..................................................... 96
4.5.1. Methods ...................................................................................................................................... 96
4.5.2. Results ........................................................................................................................................ 111
4.5.3. Discussion ................................................................................................................................. 119
4.6 CONCLUSIONS ............................................................................................................................ 121
5 VALIDATION AND TESTING .......................................................................................................... 123
SUMMARY ........................................................................................................................................... 123
5.1 METHODS ...................................................................................................................................... 124
5.1.1. Selection of sites ......................................................................................................................... 124
5.1.2. Grasp classification ..................................................................................................................... 125
5.2 RESULTS ......................................................................................................................................... 126
5.3 DISCUSSION .................................................................................................................................... 127
5.4 CONCLUSIONS .............................................................................................................................. 128
6 CONCLUSIONS ............................................................................................................................... 129
6.1 IMPACT AND CONTRIBUTIONS ................................................................................................... 130
6.2 FUTURE WORK ............................................................................................................................... 131
6.3 SUMMARY ..................................................................................................................................... 133
6.4 PUBLICATIONS ............................................................................................................................... 134
6.4.1. Conference articles ..................................................................................................................... 134
7 APPENDICES ................................................................................................................................. 137
7.1 APPENDIX 1 – MATLAB CODE ................................................................................................... 137
7.1.1. Main.m....................................................................................................................................... 137
7.1.2. loadAndProcess.m ..................................................................................................................... 139
7.1.3. matchedFilter.m ......................................................................................................................... 141
7.1.4. read_dud_index.m ...................................................................................................................... 141
7.1.5. remove_duds.m .......................................................................................................................... 142
7.1.6. index_UnEmap.m ....................................................................................................................... 142
7.1.7. tidy_electrodes.m ....................................................................................................................... 143
7.1.8. filter_extract.m ......................................................................................................................... 143
7.1.9. get_actives.m ............................................................................................................................ 143
7.1.10. features.m .............................................................................................................................. 144
7.1.11. findends.m .............................................................................................................................. 144
7.1.12. porridge.m ............................................................................................................................... 145
7.1.13. placeholder.m .......................................................................................................................... 146
7.1.14. Remove_NaNs.m ..................................................................................................................... 146
7.1.15. Mahalanobis_sphere.m ........................................................................................................... 147
7.1.16. ProjectDirections.m ................................................................................................................ 148
7.1.17. Mahalanobis_distance_sphere ................................................................................................. 149
7.1.18. Mahalanobis_distance.m ......................................................................................................... 149
7.1.19. createXYZ.m ........................................................................................................................... 149
7.1.20. DAV.m ...................................................................................................................................... 150
7.1.21. IAV.m ....................................................................................................................................... 150
7.1.22. Zerox.m .................................................................................................................................... 150
7.1.23. AccuracyCheck.m .................................................................................................................... 151
List of figures

Figure 1: The EEG process — the electrical signals generated in the brain are able to pass through the skull, where they are picked up by electrodes. This signal is then amplified and filtered before being sampled and acquired on a computer. ................................................................. 7

Figure 2: Schematic of the EMG prosthetic control process. (1) Data are acquired, (2) features are extracted in order to reduce dimensionality, (3) data are classified into intended hand motions, (4) control signals are sent to the prosthetic. .................................................................................. 13

Figure 3: (a) Unipolar electrode configuration (b) Bipolar electrode configuration .................................................. 14

Figure 4: “Pinch” grasp ........................................................................................................................................... 18

Figure 5: “Spherical” grasp. ...................................................................................................................................... 19

Figure 6: “Cylindrical” grasp. ...................................................................................................................................... 19

Figure 7: The wooden hand of Captain Danjou .................................................................................................. 25

Figure 8: Majority voting using m = 2. The output of the majority voting is shown in red. As can be seen, this reduces the variability of the data set. For point A, though the raw data point is assigned a value of ‘1’, the points either side are all assigned a value of ‘0’, and hence the output for this point is given as ‘0’. Likewise for point B, there are 2 values assigned to ‘0’, and 3 assigned to ‘1’; consequently the output for point B is taken to be ‘1’ .......................................................................................................................... 30

Figure 9: Example locations of a symmetric array of data on the posterior surface of the forearm. The locations are marked with red crosses. This would be for an array of 14 electrodes (assuming 7 additional electrodes would be placed on the part of the arm that cannot be seen). ........................................... 32

Figure 10: A simple example of using a SFS algorithm. The objective function for this example is to maximise the sum of the numbers. From the initial set of numbers, 10 is selected as the highest number. From the remaining set of numbers, the potential sums are: 15, 17, 19 and 19. Hence, 9 is selected as it would result in the highest sum. .................................................................................................................. 33

Figure 11: Matched filtering a signal with itself, followed by low pass filtering (LPF) the absolute value of the response. ................................................................................................................................. 59

Figure 12: The printed forearm replica (white, on left) and the Dimension Elite 3D printer (on right) .................. 61

Figure 13: Silicon armband with embedded electrodes. ......................................................................................... 62

Figure 14: Anatomical features used to reference the silicone armband to. ............................................................ 63

Figure 15: Raw EMG data used to extract action potential and the associated spectrum. ................................. 64

Figure 16: Action potential detected after a single iteration ............................................................................... 65

Figure 17: Flow chart of the extraction algorithm. .................................................................................................. 66

Figure 18: Action potential detected after the tenth iteration ................................................................................. 67

Figure 19: Original data set, and location of detected action potentials. ............................................................... 67

Figure 20: Simulated data ......................................................................................................................................... 69

Figure 21: The ideal impulse response for the simulated action potential. ............................................................ 71

Figure 22: Response of system in Figure 21 to a series of randomly spaced, random amplitude impulses .......... 72

Figure 23: Response of system in Figure 18 to a series of randomly spaced, random amplitude impulses, with added Gaussian noise .................................................................................................................................................. 72

Figure 24: Simulated action potential shown in black, overlaid with the action potential extracted from the simulated data set in Figure 22. ........................................................................................................... 73

Figure 25: Action potential detected after the first iteration, and the 5th iteration ................................................. 74

Figure 26: Accuracy and processing time versus number of iterations. Error bars indicate ± 1 standard deviation .......................................................................................................................................................... 74

Figure 27: Affect of running the extraction algorithm on Gaussian noise without the presence of the simulated action potential .................................................................................................................................................. 76

Figure 28: Sample of data before filtering via iterative AP extraction ........................................................................ 77

Figure 29: Data set from Figure 28 after matched filtering. Notice the decrease in amplitude for the periods of rest .............................................................................................................................................. 78

Figure 30: Action potential extracted from data set in Figure 28 ............................................................................. 78

Figure 31: Results for accuracy of classifier on data that has been preprocessed (red) and has not been preprocessed (black) .............................................................................................................................................. 81

Figure 32: The surface of the arm is “unrolled” mapping the Cartesian coordinates onto cylindrical coordinates .................................................................................................................................................. 86

Figure 33: 0, z coordinates of the channels with data available for both grasps, and those channels where data are only available for one grasp .................................................................................................................................................. 90
Figure 34: Optimal channels positions found using radial derivatives for discriminating between pinch grasp and relaxing. ................................................................................................................. 90
Figure 35: Optimal channel positions found using longitudinal derivatives for discriminating between pinch grasp and relaxing. ................................................................................................................. 91
Figure 36: Optimal channel positions found using radial derivatives for discriminating between cylinder grasp and relaxing. ................................................................................................................. 91
Figure 37: Optimal channel positions found using longitudinal derivatives for discriminating between cylinder grasp and relaxing. ................................................................................................................. 92
Figure 38: Location of useful sites found using the feature ranking optimization method, and the region lacking in data for the cylindrical grasp ................................................................................................................. 93
Figure 39: Qualitative view of the difference in means for each of IAV and DAV, both circumferential and longitudinal. The relative sizes of the markers indicate the strength of the difference in means. The region at approximately (-0.5,80) shows a large marker for longitudinal IAV, indicating that this region has a large difference in means (between the two grasp types) using the longitudinal IAV feature. Conversely, areas such as (1,150) do not have large markers at all, and therefore had very little difference in means between the two grasp types. ................................................................................................................. 95
Figure 40: A grid of 9 electrodes, represented by asterisks ................................................................................................................. 98
Figure 41: Hypothetical recordings for two grasp types. Grasp type 1 in (A) shown in green has x,y recordings of (3,3), with the total vector shown in the solid green line. Grasp type 2 in (A) shown in blue has x,y recordings of (4,-1), with the total vector shown ................................................................................................................. 99
Figure 42: The two resultant vectors for each hypothetical grasp ................................................................................................................. 99
Figure 43: Projecting two vectors (solid blue and solid green) onto two directions given by dashed lines. In (A), the vectors are projected onto a direction in the middle of the two original vectors – notice that the projections of the two vectors are both ................................................................................................................. 101
Figure 44: Projecting the green vector onto the blue vector. Note that this result in two vectors that are relatively easy to discriminate between, and also relatively easy to discriminate from background noise. ................................................................................................................. 102
Figure 45: Simulated data for four hypothetical grasp types that illustrates good class separability between the red and blue clusters, and poor separability between the black and cyan clusters ................................................................................................................. 103
Figure 46: Choosing which of the two green points is closer to the blue data cloud. In (A), an ellipse has been fitted where the data are assumed not to vary off axis at all. In this case it is difficult to algorithmically determine which green point is the better fit. In (B) however, an off axis ellipse has been fitted, showing that it is much easier to determine which green point is the better fit. ................................................................................................................. 106
Figure 47: Flow chart of the initial data set reduction algorithm. SNR is calculated based on the ratio of amplitude during active periods to inactive periods. While this required an initial manual tagging of these regions, it was assumed that each data set from an interpolated grid would have active and inactive periods at approximately the same time. M is the starting number of channels, and N is the final number of channels. ................................................................................................................. 109
Figure 48: Intercluster Mahalanobis distance measure vs. accuracy of a neural network using the same data. The red line represents a linear fit, $R^2 = 0.6452$. Average time taken for the Mahalanobis calculations was 4.7 milliseconds, while for the neural network was 30.9 seconds. ................................................................................................................. 112
Figure 49: Final Mahalanobis distance reached and time taken for increasing grid density, ± 1 standard error. ................................................................................................................. 113
Figure 50: Final Mahalanobis distance reached for increasing the number of electrodes to optimize for, ± 1 standard error. Note the increasing variability for n>6 and the decrease in score for n>14. In studies in the literature, these numbers of electrodes are not found, so this result does not infringe on the use of the methods presented for a real-time system. ................................................................................................................. 114
Figure 51: The effect of the initial “culling” step on the performance of the algorithm. ................................................................................................................. 115
Figure 52: Signal-to-noise ratio of the data set that would be left out in order to reduce the number of data sets to a given number. SNR measurements have been given as a % of maximum SNR in order to non-dimensionalise. Notice that to be left with approximately 40 data sets, a culling threshold of approximately 35% maximum SNR would be required. ................................................................................................................. 116
Figure 53: Plot of the number of calculations required as a function of M and N. ................................................................................................................. 117
Figure 54: All possible Mahalanobis distances based on a smaller data subset of 36 data sets, with the optimization algorithm’s performance marked by the red asterisk. ................................................................................................................. 118
Figure 55: A demonstration of how a smaller grid density can provide a better estimation of the optimal sites. The black grid represents a 3x3 grid, while the blue grid represents a 2x2 grid, and the red X represents
WHERE THE TRUE OPTIMAL SITE IS. IN THIS CASE, THE 2x2 GRID WOULD GIVE A BETTER ESTIMATION OF WHERE THE TRUE OPTIMAL SITE IS, AS IT INTERSECTS THE BEST SITE.

Figure 56: Accuracies using the algorithm from Chapter 4, symmetrically placed sites, and sites chosen using SFS and PCA algorithms, ± 1 standard deviation.

Figure 57: Performance of sites chosen by SFS algorithm (black) and new algorithm (blue) with P values of a paired t-test (plotted on a log scale) shown in green.
“Allow me to introduce myself. My name is Optimization of Electrode Placement in Electromyographic Prosthetic Control. I am a thesis by Scott Walbran submitted in partial fulfilment of the requirements of Doctor of Philosophy at the University of Auckland.”

OEPEPC, 22/10/2010

Recent trends in smart materials together with new and detailed models of biological function are opening new horizons for rehabilitating body functions with assistive devices. Lightweight actuators that have performance comparable to human muscles are becoming more readily available. The Biomimetics Laboratory at the Auckland Bioengineering Institute (ABI) is currently working on methods for accurate real-time control of electro-active polymers (EAPs).

Proposed applications include wearable devices for augmenting human movement, new kinds of prostheses, and many other uses of active dimension and shape control. One ABI programme is to develop externally mounted devices to help rehabilitate and supplement natural muscles. A current collaborative project run in conjunction with Industrial Research Limited (IRL) involves using actuators to assist in hand grasping as a mechanical strength augmenting glove or gauntlet. Clearly any such device would be required to provide some means of determining user intentions – for example if a user wants to pick up a pencil, they need to have an efficient method to communicate this to the device. This project therefore aims to investigate various means of extracting information from a potential user.
1.1 Background and motivation

The human hand is a marvel of evolution. Along with its prehensile thumb, it has allowed us to manipulate our surroundings with both precision and strength; a feat technology is just beginning to almost replicate. This technology presents an exciting opportunity to provide advanced prosthetic devices for amputees.

The major challenges associated with development of such a device are:

- Coupling of appropriate actuators to intended motions;

- Accurate touch and pressure sensitivity in order to determine the relationship state between actuator and object;

- Acquisition and interpretation of human intention to provide the actuation pattern for the artificial hand;

- Providing feedback to the user.

One of the goals of the ABI’s Biomimetics Laboratory is to create a high degrees of freedom prosthetic hand that can replicate the key functions of a human hand. A challenge associated with such a device is to provide an intuitive means to control it, with very little user training time. This thesis was inspired by the idea of providing more direct, intuitive ways to control devices. Current powered prosthetics use electromyographic (EMG) signals for control, and more direct means of extracting the intention of the brain involve direct nervous innervations (DNI) and electroencephalographic (EEG) deconvolution [1-5]. Other possible means for determining user intention include use of super conducting quantum interference devices and eye movement tracking [6-8].

Prosthetic devices allow those with amputations to manipulate surroundings, giving back some degree of mobility to such people. Most common prosthetic devices currently are a simple hook, using a cable from the shoulder to open and close [9]. This requires that the
shoulder move relative to the arm in order to operate the prosthesis, a motion not customarily associated with grasping. Myoelectric hands (i.e. prosthetic hands that use EMG as a control source) are available; however they generally operate on a single degree of freedom (such as opening and closing the hand), are bulky and difficult to use [10-13].

Current Myoelectric prosthetic devices take forearm EMG signals to toggle the device on or off (i.e. open or close the hand). To control a higher degrees of freedom device, clearly a more complex signal would be required. The current choices for human interfaces are:

- Direct Neural Interfacing – very accurate, can be done at any amputation level, but requires surgery and is invasive;
- Electroencephalography – difficult to get a useable signal, but non invasive and relies solely on brain signal;
- Electromyography – not as accurate as DNI, but provides useful information. Non invasive, however signals may not be present if too much of the arm has been amputated.

**1.2 Human Intention Recognition and Human Computer Interfacing (HCI)**

Several possible options exist for acquiring signals from the human body. Most of these techniques involve acquiring information either directly or indirectly from the nervous system. All are based on the detection of electrical potentials, caused by signalling through nerves and muscle.

This arises from the electrical activity of a nerve cell, which can in turn be passed into a muscle cell if it is desired to have that cell contract. The inside of a cell will usually have a resting potential of approximately -70mV, with respect to the extracellular space. This
arises due to the different ion concentrations within and without the cell, as well as the cell membrane being semi-permeable. The four main ions responsible for setting the resting potential of the cell are potassium, sodium, calcium and chloride. Potassium and chloride concentrations are higher inside than outside the cell, while the concentration of sodium and calcium ions are higher outside the cell. The concentration difference is due to ion pumps and exchangers, which actively move ions across the cell membrane. The permeability of the membrane to a particular ion can change according to how many channels within the membrane are open. At rest, the cell membrane is highly permeable to potassium, but not very permeable to other ions. This results in the resting membrane potential being set mainly by potassium, though it is slightly dependent on the other ions mentioned [14].

Activation signals are passed down the nervous system via action potentials. If the cell membrane potential reaches a certain threshold value, voltage gated sodium channels open, effectively increasing the membrane permeability to sodium. This causes the membrane potential to rapidly increase, in a process known as depolarization (as sodium ion concentration is much higher on the outside of the membrane than the inside). When this potential increase occurs in the axon (the projections of nerve cells that transmit information around the body) it causes the adjacent cell membrane to reach the threshold value, causing depolarization in this area of the membrane also. In this way, the action potential is passed down the length of a nerve cell. The velocity of propagation can vary, and is generally in the order of 1-50ms$^{-1}$ [14].

There are several ways of measuring this:

1. Electromyography (EMG) – sensors are either placed intramuscular (iEMG) or on the surface of the skin (sEMG) to detect electrical activity caused by muscles.

2. Electroencephalography (EEG) – sensors are placed on the surface of the head in order to detect electrical activity caused by nerves within the brain.

3. Superconducting quantum interference devices – a large device that measures magnetic disturbances caused by the electrical activity in the brain.
4. Direct neural interfacing – sensors are placed directly within the brain in order to directly measure the activity of the brain.

1.2.1. Electromyographic (EMG) Human Computer Interface

Electromyography measures the electrical potential developed in muscles. When an action potential reaches the neuromuscular junction, it enters the muscle fibres of that motor unit. This causes an action potential within the motor unit, known as the motor unit action potential (MUAP). The MUAP causes the motor unit to depolarize from its normal resting potential (with respect to the extracellular space) to approximately +30mV, resulting in contraction of the motor unit. The sum of all of the action potentials can be detected on the skin surface by surface electromyography (sEMG), or within the muscle by intramuscular electromyography (iEMG) [14].

Electromyography has been used for a number of years as a control signal. Some of its uses are:

- **Myoelectric hands** – artificial hands where the thumb can open and close by small motors. The thumb opens and closes in response to EMG signals taken from the forearm. Myoelectric hands are used today, however they are relatively expensive, heavy and cumbersome compared to other forms of artificial hands [15-17];

- **Assistive devices** – EMG signals have been used for assistive devices in order to aid people who have movement impairments – these include devices to induce movement in elderly persons with movement impairments, prosthetic limb replacements and exoskeletal systems [18-21];

- **Wheelchair control** – in patients with C4 or C5 level spinal cord injury (approximately in the middle-base of the neck), EMG from the shoulders has been used as an input to a wheelchair [22-24];

- **Detecting preterm births using uterine EMG signals** [25];

- **Interfacing with robots** [26-28];
Chapter 1 Introduction

- Virtual reality interfaces [29, 30];
- Speech recognition [31].

The useful bandwidth of EMG signals is within approximately 20-500Hz [8]. This presents a problem in that some of the usable signal will be in the mains frequency range (50 Hz in New Zealand), and is hence more likely to be obscured by mains noise. EMG is considered one of the fastest electrophysiological signals from the body – though no information has been found on data transfer rates [32]. Neurons relating to movement tend to have large diameters and heavy myelination – allowing for fast information flow (in the order of 50 ms⁻¹) between the brain and muscles, and hence offering the potential for greatly reduced reaction times [14].

The advantages of EMG are:
- Surface EMG is non-invasive;
- Information transfer rates are relatively fast;
- Quick to set up – only minor skin preparation is required before application of the EMG electrodes.

While the disadvantages are:
- Information must be taken from a peripheral location, so any condition that causes information not to reach the periphery will cause the interface mechanism to fail.

1.2.2. Electroencephalographic Human Computer Interfacing

Electroencephalography (EEG) is a technique that was first introduced in 1929. It is similar to surface electromyography, in that it detects electrical potential on the skin surface. The basic EEG process is illustrated in Figure 1.
Section 1.2 Human Intention Recognition and Human Computer Interfacing (HCI)

While EMG detects the potential developed in muscle cells, EEG electrodes are placed on the head and detect electrical potential caused by neurons in the brain firing [4]. As such, EEG has the potential to detect thought patterns of the brain. In practice, however, humans have very little active control over their EEG signals. This is due to two factors [2, 3, 33]:

EEG only picks up information on the surface of the brain. The brain is a folded structure, consisting of raised surfaces of the brain along with depressions or fissures. As such, much of the information in the brain cannot be detected;

The size of the electrodes used compared to the area that would need to be targeted in such a place as the motor cortex is quite large. As such, the specificity of electrodes is quite low, and the potential detected on the surface of the head is the sum of the potentials from many parts of the brain.

Conventional EEG recordings use a silver/silver chloride electrode in conjunction with an electrolytic paste on the skin surface. The electrolytic paste anchors the electrodes to the skin surface, as well as providing a conducting medium. As the electrodes need to be placed directly onto the skin, the scalp needs to be cleaned – involving hair removal, and abrading the skin (this can be an uncomfortable process for the patient). Skin abrasion improves the conductivity of the skin, hence allowing for a clearer EEG signal. Recent developments have indicated that it may be possible to detect EEG signals without need of skin preparation [14, 33].
Studies have indicated that humans can learn to control certain components of their EEG; showing the potential for this to be used as a new communication channel. Note that this requires no neuromuscular communication, indicating it could be useful for patients with conditions such as amyotrophic lateral sclerosis (ALS); where the patient is essentially “locked in” their own body [3].

Several parts of the EEG have been shown to be useful as control signals – such as the $\mu$ and $\beta$ rhythms [34], and the p300 event-related potential. $\mu$ rhythms have a frequency of approximately 7-11 Hz (which is within the $\alpha$ band of the EEG). They are seen over the sensorimotor cortex, and are strongly suppressed during movement of the contralateral limb. $\beta$ rhythms have a frequency of approximately 12-30 Hz. $\beta$ rhythms of varying frequency indicate that a subject is anxious, busy or active. The p300 event-related potential is a positive deflection in the parietal lobe of the EEG recording approximately 300ms after a stimulus is provided to the subject. It is thought that this could be used to determine whether or not a stimulus (such as a word displayed on screen) is desirable to the user of an HCI [1, 35].

One particular part of the EEG that humans have been shown to have control over is the slow cortical potential (SCP). The SCP describes a part of the EEG that has a slow, event related current shift. The slow cortical potential however has a long time constant, and as such information can only be transferred at a limited rate – current research suggest a maximum of 35 bits/minute [3, 4, 36, 37]. For comparison, a keyboard-based interface gives data transfer rates between user and computer in excess of 1 kilobit per minute [5].

The advantages of EEG are:

- It is relatively non-invasive – electrodes must be applied to the head, but the skin is not broken;
- Information is taken directly from the brain, and hence corresponds more directly to the users intention.

While the disadvantages are:

- It is a non-specific method for measuring brain activity;
• Noise levels are generally relatively high, though there has been investigations into methods for improving EEG signal quality [38, 39];

• Data transfer rates are relatively low;

• EEG requires preparation of the scalp – skin abrasion, and applying electrode gel.

1.2.3. Superconducting quantum interference devices

Superconducting quantum interference devices (SQUIDs) are another way of detecting the electrical activity in the brain. A SQUID remotely detects the magnetic disturbance caused by the electrical disturbance when neurons fire (magnetoencephalograms – MEG) [33].

The advantages of SQUIDs are:

• They are non-invasive;

• Information is taken directly from the brain.

While the disadvantages are:

• They require magnetically shielded environments;

• The detectors require cryogenic cooling – high temperature superconductors operate at 77K [7];

• The equipment required is bulky and not at all portable.

Because of the limitations placed on this technology, it was not investigated any further.

1.2.4. Direct neural interfacing

Another possible way of interfacing is by directly instrumenting the brain. This can be achieved through the use of neurotrophic electrodes that use trophic factors to encourage the growth of neural tissue into hollow electrode tips. The neural tissue grows through both ends of the electrode, holding it firmly in place [36, 40, 41].
The biggest disadvantage of direct neural interfacing is that it is extremely invasive. It requires sub-cranial implantation of the electrode devices, and as such poses risks (such as infection and bio-incompatibility) to the patient, and the implantation is not easy. The benefit to this method however is that greater selectivity can be achieved. Rather than using components of the EEG (such as SCP) as a channel for control, the firing of individual neurons can be used [36, 40, 41].

Kennedy et al [40] presented a method where a patient was able to control movement of a cursor across a screen, by increasing or decreasing the firing rate of motor neurons. The patient showed improved performance with repetition of the same task – indicating that the brain is able to learn to control this new communication pathway. For the first few months of operation, neural signals would fire when the patient moved specific muscles – first the mouth and tongue, then the eye and eyebrow. Approximately 5 months after implantation, the patient required no movement in order to activate the signals. This indicates that the patient had learned to control some neural signals in order to interact with the cursor. The patient in this paper was able to write three letters within 72 seconds.

The advantages of direct neural interfacing are:

• Information is taken directly from the brain;

• Specific areas of the brain can be targeted.

While the disadvantages are:

• It is a very invasive method of interfacing, requiring direct implantation into the brain.

1.2.5. Summary of signal acquisition techniques, and motivation for using Electromyography

Several possible methods for acquiring signals from the body have been presented. Electromyography detects the electrical potential on the surface of the skin, caused by activation of muscles underneath the skin. It has advantages in that it is non-invasive, comparatively fast, and easy to use, while the main disadvantage lies in its inability to
detect signals in cases such as total body paralysis. Electroencephalography detects the electrical potential on the surface of the skin, caused by activity in the brain. EEG as an interface is non-invasive, and takes information directly from the brain, however it is limited by slow transfer speeds, and lack of specificity. SQUID non-invasively measures the magnetic disturbances caused by the brain, however it is currently limited by its large size, cryogenic operation, and magnetic shielding. Direct neural interfacing takes information directly from electrodes implanted into the brain, however is very invasive, and would not be available for use “off the shelf”.

Electromyography was chosen as a control signal for a number of reasons:

a. Surface EMG is non invasive.

A non invasive method of interfacing with the body is preferable for the following reasons:

- Not breaking the skin greatly reduces the risk of infection, and installation is more comfortable;
- No surgery is required;
- It would potentially be a lot easier to use “off the shelf” than anything that requires implantation;

b. EMG signals relate to the activation of muscle.

Depolarization of the muscle motor units is responsible for both muscle contraction and EMG signals. This means that there is a direct relationship between the contraction of muscle and EMG signals. The implication of this is that muscle contraction can essentially be used as a control source for an external device such as a computer. This acts to “cut out the middle man” such as a keyboard or mouse, and uses muscle signals directly as an input.

c. Speed
EMG is one of the fastest electrophysiological signals. Compared to a signal such as respitrace (i.e. strain measurements of the chest as the subject breathes), EMG signals change much more rapidly – with the upper limit of the EMG band being at approximately 500Hz. This allows information to be extracted faster, hence allowing for quicker changing control over any external device.

d. An initial experiment conducted at the beginning of this thesis used a simple electromyographic amplification circuit to control a bending actuator. This demonstrated very effectively the ease with which EMG could be used as a control signal for a single degree of freedom. The need for a means to deconvolve the signals from multiple channels in order to control multiple degrees of freedom was then identified as the required next step.

1.3 EMG based prosthetic control

Clinically available prostheses tend to use amplitude or rate encoding with mode control. Since the early work by Hudgins et al. [42], attempts at multifunctional prosthetic control tend to (though do not always) focus on pattern recognition systems. The process to myoelectric control involves a series of steps to move from the EMG signal to the control of a prosthetic device. Figure 2 depicts this process, described as follows [43]:

1. Data acquisition and conditioning: Electrodes are applied to the surface of the skin, where signals are acquired. The signal is then amplified, filtered and sampled.

2. Feature extraction: Data are segmented into time windows, and various features are extracted from each time window. These features are chosen to best represent the underlying data, while reducing the complexity of the data set.

3. Classification: The representative features are passed through some form of classifier that maps the input (features) to the output (desired hand motion)
4. Prosthetic control: The control signals from the classifier are fed to the prosthesis, and the prosthesis acts to carry out the desired motion. This will generally involve local feedback in order to prevent damage to an object being grasped.

Figure 2: Schematic of the EMG prosthetic control process. (1) Data are acquired, (2) features are extracted in order to reduce dimensionality, (3) data are classified into intended hand motions, (4) control signals are sent to the prosthesis.

1.3.1. Data acquisition

In order to improve the signal quality, the skin at the acquisition site is generally prepared by removal of the hair, followed by scrubbing of the skin using a preparing gel in order to remove dead skin cells. Electrodes are then applied to the skin in order to record the surface potential. While this is commonly used in a laboratory, it is not a process that would be easy for a prosthesis user to undergo.

In order to reduce noise, a ground electrode is generally placed at a location that has very little potential change – a bony prominence such as the elbow is usually preferred. Voltages are measured as a potential difference; hence the surface EMG signals require two additional electrode locations: one as the fixed “reference” state, and the second as the site where the “signal” is recorded. The reference electrode can be in the same place as the ground electrode; this is referred to as “unipolar” recordings. In this configuration,
if multiple channels are being recorded, they all reference to the same voltage point and hence have a relationship to each other, i.e. differential voltages can be determined.

In contrast, the reference electrode may also be applied to a location close to the signal electrode, recording the difference between two close points – in essence an electric field. This configuration is referred to as “bipolar” recording. This is shown in Figure 3 (b). If two unipolar recordings are taken in close proximity, a bipolar recording can be simulated by taking the difference in the two unipolar recordings.

Figure 3: (a) Unipolar electrode configuration (b) Bipolar electrode configuration.
1.3.2. Electrode locations

Symmetrically placed surface electrodes have been shown to perform as well as intramuscular electrodes when classifying signals into grasp types [44]. Targeting the electrode location to specific muscles does not seem to affect classifier performance [45], however optimally selecting locations appears to offer some improvement [44, 46].

The algorithms used to select the optimal locations however are based on a “greedy” approach – whenever a choice is available, the choice that appears “best” is taken without considering its affect on future choices [47]. Greedy algorithms often perform well, and in a reasonable time frame, however generally a solution closer to optimal can be obtained using a different algorithm [47].

1.3.3. Signal preprocessing

Prior to myoelectric signal feature extraction and classification, recent attempts have been made to preprocess the myoelectric signal using a wavelet approach in order to improve the accuracy of the classification system [48-51]. Several previous methods have been able to accurately pick out action potentials from within extraneuronal and electromyographic signals [52-59].

These methods however are reasonably complex, and are computationally intensive making them currently unsuitable to be included in a real-time system; hence it is desired to produce a relatively simple algorithm that can be applied in addition to a matched filter to improve signal to noise ratio, with the intention of using this as a preprocessing step to improve accuracy in signal classification. Matched filtering [60] (also known as cross correlation) is relatively simple to implement in Matlab, and is effective at picking out a given signal from within an unknown signal – in essence if an action potential could be extracted from within a segment, it could be used as a template to filter the signal and provide a way to quickly reduce the part of the signal that is not “action potential like”. 
1.3.4. Feature extraction

A method common to many EMG based HCIs is the extraction of various features from the data set to be used as input to a classification system. Carefully chosen feature extraction methods can result in a decrease in the amount of data to be handled, hence a decrease in computation required and an increase in speed, without adversely affecting the performance of the classifier. There is ongoing research into reducing the features used in classification by selecting optimal feature sets [61].

Feature extraction is a commonly used method in signal processing. It allows for a reduction in dimensionality often without loss of information. Previous EMG feature extraction techniques fall under two categories: structural and phenomenological (empirical) approaches [43]. Structural approaches are based on physical and physiological models, whereas phenomenological approaches are based on the performance and robustness of the feature. Each feature extraction technique will also belong to the time domain, frequency domain, or time-scale (time-frequency domain).

Structural approaches need to consider the origin of the electromyographic signal. The signal is formed by the summation of several motor unit action potentials (MUAPs) passed through a filter comprised of the skin and connective tissue between the forearm surface and the muscle. Basmajian and Deluca [62] modelled the electromyograms as

\[ s(t) = \sum_j \text{MUAPT}_j(t) + n(t) = \sum_j \sum_i k_j f\left(\frac{t - \theta_{ij}}{\alpha_j}\right) + n(t) \]  \hspace{1cm} (1)

Where MUAPT is a MUAP train, modelled by \( \text{MUAPT}_j(t) = \sum_i k_j f\left(\frac{t - \theta_{ij}}{\alpha_j}\right) \), \( j \) is a specific motor unit, \( n(t) \) represents noise, \( f(t) \) is the shape of the action potential function, \( k_j \) represents an amplitude factor associated with a given action potential, \( \theta_{ij} \) refers to the occurrence time of each MUAP, and \( \alpha_j \) is a temporal scaling factor. It is clear from this equation that the signal as observed on the skin surface would consist of multiple overlapping action potentials at various amplitudes based on the time delays in equation
1.3. EMG based prosthetic control

The surface electromyograms has been previously modelled using the generation of filtered white Gaussian noise passed through a linear filter [63-66].

Features that have been previously used in literature that this thesis will utilise are:

- Mean absolute value used by Hudgins et al. [42].
- Low pass filtering rectified data used by Vuskovic and Du [67, 68].
- The number of times the signal crosses the x-axis in a given time window used by Hudgins et al. [42].

1.3.5. Classification

When multiple data sets correspond to different grasp types, it can be assumed that there will be different “clusters” of data – with each cluster corresponding to each grasp type. The features extracted from the EMG data must be classified into distinctive classes – these classes should correspond to meaningful movements of the hand. The approach presented in this thesis uses grasp types of the hand as the desired outputs of the classifier. A classifier in this sense is defined as an algorithm that takes the output from the feature extraction and converts them into commands or signals for the desired hand movement. In order to best represent the bulk movement of the hand, its functions have been separated into the three main grasps:

- **Pinch** – shown in Figure 4. The thumb and forefinger are brought into opposition. This is used for grasping objects such as keys, paper and combs.

- **Spherical** – shown in Figure 5. The fingers curl around an object; thumb and forefinger are not brought into opposition. Used in grasping spherical objects such as a tennis ball.

- **Cylindrical** – shown in Figure 6. The fingers again curl around an object; however they are kept in-line. Thumb and forefinger are not brought into opposition. Used in grasping objects such as a glass or drink bottle.
The grasp types were chosen as they are able to represent the majority of the motions of the hand [69]. If needed, they can then be subdivided – for example spherical can be separated into small spherical (for a tennis ball) or large spherical (for a grapefruit).

Figure 4: "Pinch" grasp.
Section 1.3 EMG based prosthetic control

Figure 5: "Spherical" grasp.

Figure 6: "Cylindrical" grasp.
1.3.5.1. Covariance and the covariance matrix

The classification method used in this thesis involves a distance measure called the Mahalanobis distance. Before the Mahalanobis distance can be understood, the concept of covariance must first be explained. The variance of a data set is a measure of the spread of the data from its mean. The covariance between two data sets is a measure of how much one variable changes as the other changes. As an example, two variables that would exhibit a very high covariance could be the height of a person, and the length of their trousers. Conversely, two variables that would exhibit a very low covariance could be the height of a person, and the street number they live at. The covariance matrix for a number of variables is simply a matrix whose $i,j^{th}$ entry is simply the covariance between variable $i$ and variable $j$.

1.3.5.2. Mahalanobis distance

The Mahalanobis distance takes into account the covariance – that is it does not assume each dimension is independent of the other. Because the Mahalanobis distance makes use of the covariance, it requires statistical information i.e. at least one of the data sets must contain more than one data point. The Mahalanobis distance in essence gives an idea of the distance between a data point and a cluster, taking into account of the standard deviation of the cluster in the direction of the data point.

1.4 Research Objectives

Clearly there is a need to provide accurate control to prosthetic devices through recognition of the intention of the user. While there are more direct ways of obtaining information from the brain, EMG has the potential to provide a “plug and play” solution due to the relative simplicity of setting up such a system. Currently there is an established method for taking EMG signals and classifying them into the desired movements of the
hand, however the effect of parameters prior to this classification has only been investigated more recently.

The overall goal of this thesis has therefore been to improve the accuracy of myoelectric prosthetic control. In order to do this, two parameters were investigated – preprocessing data to remove noise, and the location of electrodes. These parameters must be able to be implementable in real time, and use a realistic number of EMG channels. As highlighted in Sections 1.3.2 and 1.3.3, there has been previously some work investigating the effect of preprocessing and optimizing electrode locations, however they have not been fully investigated. As a result they have been selected as the most likely means by which an improvement in accuracy can be provided. Further details on these parameters are as follows:

1. Signal preprocessing:
   a. Improvement of signal-to-noise ratio (SNR) in surface EMG recordings via detection of characteristic signals.

2. Electrode locations:
   a. A method to optimize electrode placement for electromyographic control of prostheses.
   b. Determination and application of a means to find optimal data sets from a larger data set with given, non-regular, geometric properties.

In particular, it was desired to determine:

1. If a real time adaptive filter could be used to improve SNR, and whether this would lead to an improvement in prosthetic control accuracy.

2. If previous location optimization algorithms are a valid means to determine which sites on the forearm are best for use in prosthetic control.
Chapter 1 Introduction

3. If further reductions in the number of electrodes are possible without compromising accuracy.

As the location of electrodes would not need to be determined while the prosthetic is in use and hence will not impact real time execution of pattern recognition control.

One of the desired end user points of this project is to investigate the coupling of wearable devices to the human body. This would involve developing an interface between human muscle movements and the controlling software and hardware of the gauntlet. The principle behind this would be to detect signals from the body – such as muscle movements in the forearm, and to use these signals in the control software for the gauntlet. This “human computer interface” could then be used to assist in injury rehabilitation – where patients have limited ability to control their muscles due to nerve injury or muscle degradation. In this case, a series of actuators could be developed to provide the force that the body itself cannot – enabling the patient to still use and redevelop the existing muscle/nerves.

The application of this project then has the potential to extend into control of other strength augmenting devices – such as heavy lifting machinery. It could also be used to provide a more advanced interface with computers – by knowing more precisely the intended movement of a human being.

1.5 Thesis outline and contributions

This thesis provides an insight into several areas of electromyographic prosthesis control.

- The iterative application of an action potential detection algorithm for surface EMG, combined with matched filtering allows for an increase in SNR prior to any attempt at prosthesis control.
The development of a dataset reduction algorithm has provided a means to select, in a timely manner, a close to optimal subset of data recordings for use in a co-dependent classification.

This algorithm has been used to investigate electromyographic electrode placement; it is envisaged this could be used in future studies that use electromyographic recordings to investigate signal origins.

An investigation into the current method for electrode placement via symmetrical location of electrodes, compared to a placement based on the developed algorithm and two previous site selection algorithms.

1.6 Summary

Prosthetic hands are becoming more and more capable of accurately replicating the functionality of the human hand. These advances in complexity require a more sophisticated human interface than few degrees of freedom currently provided. Electromyographic control is the most common and promising method for providing a human interface. However current commercially available prosthetics have yet to introduce an increase in complexity. Current research has shown that various techniques show promise in providing more advanced control; as is detailed in Chapter 2. There are however, several areas that could potentially be improved upon. The most likely means for providing an improvement were determined to be the effect of preprocessing data prior to classification, and determining an improved method for selecting locations of data acquisition.
“It would seem that throughout our history, humans have become adept at finding more efficient ways to maim each other. With modern technology, our ability to do so is unparalleled, while fortunately restrained; yet, with modern technology can we not also seek to move in the reverse direction?”

OEPEPC, 10/01/2011

Evidence of prosthetic devices can be seen in items such as bronze, crown-like dental implants from the late Roman age [70]. In 1853, Captain Jean Danjou of the French Foreign Legion lost his hand in a rifle misfire, and wore a wooden prosthetic until the day he died at the Battle of Camarón, Mexico in 1863. With just 62 soldiers and 3 officers, Captain Danjou attempted to hold against a combined force of 2000 Mexican soldiers. Fighting down to the last five soldiers, each with a single bullet left, his troop charged with their bayonets. The opposing commander, in admiration of their charge, ordered a cease-fire and allowed the survivors to form an honour guard for Captain Danjou’s body. Such was the impact of this battle that the prosthetic hand of Captain Danjou (shown here in Figure 7) is taken out of display and paraded on the 30th of April each year – Camarone day [71].

Figure 7: The wooden hand of Captain Danjou
Earlier devices seem to be aesthetic in nature (such as Captain Danjou’s arm) and while there are several such devices available today, there is now a greater focus on restoring some lost function to the body. A prosthetic hook was first patented by Ray Trautman in 1919 [72], giving some functionality back to those who have lost hands. In a more modern setting, there are several commercially available prosthetic hands, arms, legs and implantable devices.

To be successful, a functional prosthetic requires not only some means of restoring functionality, but a means for feedback to the user and a method for attaching it to the body and recognizing the intention of its wearer. It has been identified that multifunctional prosthesis control can only exist when control of one function of the prosthetic does not affect control of any other – that is to say a movement such as abduction/adduction of the wrist can be performed entirely independently of all other movements – such as opposition of the thumb [73]. Consequently, it is important to accurately determine the user’s intention.

2.1 Previous sEMG applications to HCI

Myoelectric hands were first developed by the Central Prosthetic Research Institute, Moscow of the former USSR, in 1964. Common myoelectric hands use a single or double EMG electrode pair to control opening and closing of a single degree of freedom actuator [74]. More recently, there has been increased interest in applying pattern recognition and cluster analysis techniques in order to allow for the detection of many different hand movements [73, 75-84], an idea popularized by Hudgins et al [42].

Previous works have shown the ability to determine preset grasps using 4 well placed electrodes [68, 69, 73, 85, 86], object specific grasps [87], individual finger movements determined using 32 targeted electrode sites [84], or extraction of neural drives from surface EMG signals [88]. These methods however have all suffered drawbacks in that either expert knowledge is required to correctly place electrodes, or very large numbers of EMG data sets are required to provide high accuracy.
Grasp recognition has been performed using fuzzy logic techniques – fuzzy c-means data clustering and predictive fuzzy adaptive resonance theory (ARTMAP). Whereas traditional logic will have a binary classification system – i.e. an input either belongs to a particular output (given the value ‘1’) or does not belong (given the value ‘0’) – fuzzy logic allows the inputs to have a certain “degree” of belonging – for example an input may belong to output 1 to a degree of 0.7, and to output 2 to a degree of 0.3. So, with fuzzy c-means clustering, a data point will have a degree of belonging to each cluster dependent on its distance to the mean of the points in each cluster. Likewise, fuzzy ARTMAP uses multiple ART neural networks to assign a given input a degree of belonging to each of the output types [68, 73].

Individual finger movements were classified using a principal component transform passed through a multilayer feed-forward artificial neural network. Common to all of these techniques was the arbitrary number and location of electrodes chosen – electrodes were placed in order to target specific muscle groups that would be used when a particular grasp/movement was being performed. While this allowed for accuracies of greater than 90% classification, there was the possibility that redundant information was being collected, or that even greater accuracies could be obtained using the same number of electrodes were their position optimized. Placement has also required expert knowledge, and in the case of patients with limb deficiencies the related muscles may not be where they are expected.

Pattern recognition based control has been the favoured form of HCI, and has four main modules – see Figure 2 for a graphical overview of this process [43]:

1. Data segmentation initially segments the data into small parts, and preprocesses the data prior to feature extraction, with the goal to improve accuracy and response time of the HCI.

2. Feature extraction uses preselected algorithms to extract features from the data. The goal is to reduce the size of the data set to improve efficiency of classification. It is important to select effective features.
3. Classification involves recognition of patterns and classifies them into predetermined categories.

4. Control algorithms create the signals for a prosthetic device from the outputs of the classifier.

In order to be useful, myoelectric control needs to provide intuitive and dextrous control. A 2007 review of myoelectric control systems [43] identified 3 major aspects of controllability:

- Accuracy of classification;
- Intuitive use of control;
- Response time.

Accuracy is the most important part of a prosthetic control, and can be improved through methods such as extracting more information from muscle states, creating a more powerful controller to use this information, increasing the number of muscles that are used in data collection and developing a feature set with rich information.

Data should be segmented into time slots in order to facilitate feature extraction. A study by Hargrove et al. [89] showed that a controller delay between 150-250ms produced optimal results.

Segmentation can be performed using either adjacent windows or overlapping windows:

- Adjacent: processing delay while the signals are acquired for the appropriately sized window and delay for the classification. While the data are being collected, classification of the previous data window takes place and there is the possibility of downtime for the processor;
- Overlapping: the idle time of processor is used to generate more outputs. However, this can possibly lead to redundancy in outputs.
Englehart & Hudgins [81] highlight that continuous segmentation on a steady state signal can allow for reduction in segment length to 32ms without much decrease in accuracy. Continuous segmentation makes use of both transient and steady-state data, and produces classification very quickly. This study took data from four sites equally spaced around the forearm, followed by feature extraction using windows that may be up to 256ms in length, and pattern recognition using linear discriminant analysis (LDA) – a method that finds a linear combination of features to characterize multiple classes of input signals. The feature sets extracted were zero crossings, waveform length, slope sign changes and mean absolute value. While the feature extraction and pattern recognition steps used were in line with prior art, the location of data acquisition sites assumed that symmetrically placed electrodes would perform as well as other sites.

Farina & Merletti [66] demonstrated that overlapped segments do not significantly improve accuracy of spectral features, they merely increase processing time. The same paper also showed that segment lengths of less than 125ms leads to high variance and bias in the frequency domain.

Myoelectric signals have a transient state consisting of bursts from fibres, and a steady state resulting in a constantly maintained contraction from a muscle. Hudgins et al. [42] were the first to consider transient signals that come with the onset of contraction. However, contractions should occur from rest to be used in a transient classifier, this impedes tasks utilizing multiple degrees of freedom; therefore steady-state signals have been commonly used for real-time control.

2.1.1. Majority voting

Majority voting is a technique that has been used in previous studies in order to “smooth” data. Majority voting [81] is illustrated in Figure 8. This process includes the previous $m$ samples and the next $m$ samples for decision making after classification. The decision was given by the class with the greatest number of occurrences within the $2m+1$ data points given by this window. Majority voting was used when a dense stream of class decisions was input – a form of low pass filtering in the post-processing Section. This would effectively diminish the effect of decreasing the segment size. In the same paper [81], it
was shown that with a segment length of 32ms and an acceptable processing delay of 128ms, the maximum accuracy could be obtained, resulting in a very responsive system.

![Diagram showing majority voting using m = 2. The output of the majority voting is shown in red. As can be seen, this reduces the variability of the data set. For point A, though the raw data point is assigned a value of ‘1’, the points either side are all assigned a value of ‘0’, and hence the output for this point is given as ‘0’. Likewise for point B, there are 2 values assigned to ‘0’, and 3 assigned to ‘1’; consequently the output for point B is taken to be ‘1’.

2.2 Hand movement recognition

The human hand has 22 degrees of freedom (DoF) – three flexion/extension and one abduction/adduction per finger, two flexion/extension and one abduction/adduction for the thumb and three rotational at the wrist [90]. Traditional Myoelectric prosthetics controlled a single degree of freedom [91]. Clearly, it would be an improvement to allow higher degree of freedom control over a prosthetic. However, control over each individual
DoF may be unnecessary and/or difficult to achieve. Also, many of the degrees of freedom provided by the hand are impossible or difficult to individually elicit, such as flexion/extension of each joint of the third (ring) finger. It should also be noted that control over abduction/adduction of the fingers is almost entirely intrinsic to the hand, making post-amputation electromyographic abduction/adduction detection difficult or impossible [90].

Obviously, there will be a trade off between the number of DoF that a myoelectric prosthesis can detect, and the accuracy/computational power required. A common approach to high-DoF detection has been pattern recognition of grasp types [67-69, 85, 86], though more complex pattern recognition has been attempted reasonably successfully [84].

### 2.3 Electrode locations

It has been previously identified that there will be different results for different patients, and that it may be necessary to choose electrode locations for amputees that are different to non-amputee subjects [43, 69]. Ferguson and Dunlop [69] identified that repeatedly locating electrodes at the same spot was essential as small changes in position can lead to large changes in output.

Previous approaches to electromyographic control have placed electrodes above the appropriate muscle groups [68, 69, 73, 92], or by placing electrodes around the forearm [81-83] with one study [84] applying 32 electrodes covering the majority of the forearm.

Hargrove et al. (2007) [44] performed a study comparing surface and intramuscular myoelectric signals. Data were recorded simultaneously from both surface EMG sites and intramuscular sites, while 10 different hand/wrist movements were performed. The sEMG signals were recorded using a linear array with 16 electrodes wrapped around the forearm, with the amplification system being configured to make differential measurements between adjacent sites, resulting in 15 channels of signals. This study found that there
Chapter 2 Literature review

was no significant difference between classifying using surface signals or intramuscular techniques.

The same study [44] also considered the effect of using a smaller subset of sEMG channels. Two methods were chosen to reduce the data to a smaller subset: symmetrical reduction and brute force reduction. For the symmetrical method electrodes were chosen as being equally spaced around the circumference, while for the brute force method every combination of channels possible was tested and those that provided the highest accuracy were chosen to be the optimal subset. An example of the locations of a symmetric array can be seen in Figure 9.

![Figure 9: Example locations of a symmetric array of data on the posterior surface of the forearm. The locations are marked with red crosses. This would be for an array of 14 electrodes (assuming 7 additional electrodes would be placed on the part of the arm that can not be seen).](image)

Adding additional channels of sEMG data improved classification accuracy, however a maximum accuracy is quickly reached beyond which adding additional channels is not beneficial. Optimal sites also reach this maximum quicker than the symmetric sites, though as the number of channels included in the optimisation was increased beyond 10, the accuracy decreased slightly [44].

Optimal channels were not the same between all subjects, however it was also noted that there was some variation in the manner of the application of the surface electrodes. There
were general regions that could provide better discrimination. While this study provided an investigation into both the difference between optimal and symmetrical surface sites, and surface sites compared to intramuscular sites, the data were collected from a single strip around the forearm. Ideally a study into optimal sites would consider a larger area over which the data were taken [44].

A high density EMG study was performed by Daley et al. [46] by taking data from up to 64 unipolar electrode sites for both normally limbed and transradial amputee subjects. Sequential forward selection (SFS, demonstrated in Figure 10) was used in order to determine if the accuracy could be improved for normal limbed and transradial amputees by optimally placing electrodes. Subjects were given a brief period of time to familiarize and practice (with the amputee subjects being given slightly longer). In general classification accuracies were greater for normal limbed subjects than for amputee subjects, though the classification was based on more classes of data for normal subjects than for amputee subjects. It was also noted that normal limbed subjects will perform the expected movements and have proprioceptive feedback on a day to day basis while amputee subjects would not.

5, 7, 9, 9, 10

⇒ Select 10

10 5, 7, 9, 9

⇒ Select 9

10, 9 5, 7, 9

Figure 10: A simple example of using a SFS algorithm. The objective function for this example is to maximise the sum of the numbers. From the initial set of numbers, 10 is selected as the highest number. From the remaining set of numbers, the potential sums are: 15, 17, 19 and 19. Hence, 9 is selected as it would result in the highest sum.
The SFS algorithm is useful as it is quick, easy to implement, and depending on the objective function can produce a very close to optimal result. However, as it does not allow for data to be removed once it has been included, there is the potential that channels that interact very well with each other would not be included simply because they do not interact well with the current optimal solution. Given that there is a large amount of cross talk in surface EMG (i.e. one channel will often have overlapping data from another channel), it is likely an improved algorithm could be developed [46].

The optimally selected sites chosen by the SFS algorithm in general outperformed the full set of sites, with the cause being cited as the "curse of dimensionality" - i.e. the higher the dimensionality of a data set, the more difficult it is to generalize from it. If a higher dimensional data set is to be used, more training data would be required. The exception to this was for the amputee subjects where two subjects showed an increase and two showed a decrease in accuracy [46].

While this provided a study into the effects of optimal electrode sites, these sites were not compared to the same number of sites placed without previous knowledge of the muscular activity. It also used a simple SFS method for finding optimal sites. While this has allowed for more useful sites to be selected, this algorithm only considers adding a single channel at a time, and hence does not take into account possible effects caused by interactions between several channels [46].

Farrell and Weir [45] state that symmetrically placed sEMG electrodes have the advantage in that they would simplify socket fabrication. They also state that relatively little work has been done towards either the effect of electrode targeting or implantation. Signals for this study were collected using a 16 channel sEMG system, and fine wire electrodes for the intramuscular recordings. Four electrode configurations were investigated: targeted surface, untargeted surface, targeted intramuscular and untargeted intramuscular.

Targeted surfaces sites were based on locations that produced the maximum EMG signal during test movements of each muscle. Channel subsets were chosen using a SFS method. This is a very computationally efficient means of determining channel subsets, however
may not take into consideration the affect that different combinations of channels would have when interacting with each other. Ideally a channel selection algorithm would allow for some amount of inter-channel interaction without resorting to a brute force method [45].

The study did not however compare the accuracies of the subsets chosen using this forward selection algorithm with the accuracies of data taken from the same number of sites placed symmetrically around the arm. While the untargeted sites were placed symmetrically, once they were passed through this forward selection algorithm, the resulting sites would not be symmetrical [45].

The results of this study by Farrell and Weir show again show that increasing the number of channels used in the classifier results in an increase in accuracy, and again that this will tend to result in diminishing returns - i.e. the added benefit of each new channel decreases as more channels were added. Consistent with previous works [44, 46] the optimal number of locations was found to be approximately 4 before very small benefits were added. It was suggested that at least four electrodes are used for a clinical device [45].

It was also shown that there was very little difference between targeted and untargeted sites, however intramuscular electrodes gave lower classification accuracies than surface electrodes when less than four electrodes were used. The reason for this is sited as being that intramuscular electrodes were much more specific than surface electrodes, so when a single electrode is removed, it is possible to remove the only source of recording for a particular muscle [45].

The study concluded that the choice of electrode should be driven by clinical factors rather than by classification accuracy. While it was shown that targeted sites do not outperform symmetrically placed sites, data were only collected from 16 surface sites. As such, it can not be concluded that data from other locations on the arm could not provide better information for a classification system. Given that sEMG signals tend to pick up data from multiple muscles and that such data are filtered through the skin, it is highly likely that more appropriate signals would not be found directly above the expected muscle [45].
Zhou et al. [93] presented a method for reinnervating muscles using residual nerve endings from non-functional muscles in amputees, called targeted muscle reinnervation (TMR), which has been continued by Kuiken et al. [94]. This allowed for activity that was previously not available to be amplified by the remaining muscle, and picked up by an EMG sensor or sensors. A high density EMG study was performed using 79-128 electrodes on the subjects after the residual nerve endings had been reinnervated. This study also reduced the data to a smaller subset using a SFS approach [93].

Results of this study were very promising, indicating that the TMR successfully allowed previously unavailable information to be picked up. The SFS sites showed that only small decreases in accuracy were found when using in the order of 5-9 electrodes, indicating that fewer channels of optimally placed electrodes could perform nearly as well as a large, high density array. No mention was made however of comparing these results to the results of taking data from a regular grid of reduced size, similar to symmetrically placing electrodes around a forearm [93].

Barniv et al. [29] used EMG signals to anticipate head movement for virtual-environment applications - essentially determining the movement using EMG signals which precede motion, and hence decreasing latency by up to 70ms. Their initial analysis used 32 electrodes rather than trying to target specific regions. This essentially allowed a given spatial resolution to be obtained. A principal component analysis (PCA) revealed that electrodes located above main muscle groups involved in head motion have the most relevant information.

Subsequently 8 pairs of electrodes were chosen to target these muscle groups for the study. This PCA process would allow for quick selection of the optimal channels, however may also not take into consideration the interaction between various channels. However, the results of the PCA algorithm were also not compared to other potential algorithms, and the mapping back from principal components to actual channels may also fail to take into account potential inter-channel interactions. A suitable data reduction algorithm could potentially eliminate both the quick selection issue, and the channel-
channel interaction issue, however the results of such an algorithm would need to be compared to a PCA and SFS algorithm [29].

Kanitz et al. [95] used a genetic algorithm to investigate reducing both the feature set (see Section 2.5) and number of electrode locations required. Sixteen channels of data were recorded while subjects performed twelve different finger movements. This genetic algorithm showed that the number of channels could be reduced to between 8 and 11 without greatly affecting the classification accuracies. No comparison was made however with other subsets of sites. It was also noted that the genetic algorithm optimization, while effective, is too time consuming and that other methods could be initially used to reduce the problem size. Ideally the study could also be furthered to a high-density study such as in [46, 93], however this would necessarily further increase the problem size.

More recently, Young et al.[96] investigated the effect of the size and orientation of electrodes to "shift" - movement of the electrodes during operation. It was noted that larger electrodes would cause a given shift to have less of an impact relative to the electrode pickup, hence possibly reducing the effect of the electrode shift. Data were taken from five electrode locations: non-shifted sites, 1 and 2cm shifts parallel to the muscle fibres, and 1 and 2cm shifts perpendicular to the muscle fibres.

Their results found that "transversely" orientated bipolar electrode pairs (i.e. circumferentially around the arm/perpendicular to the length of the arm) were more sensitive to electrode shift than longitudinally oriented electrodes. Perpendicular shifts also showed higher classification errors than parallel shifts. It was also found that using both longitudinally and transversely orientated electrodes gave a significant decrease in error rate [96].

A continuation of this study investigated the inter-electrode distance [96, 97]. Four sites were used for data collection, evenly spaced around the forearm. Each site had two unipolar electrodes, such that 8 channels of differential EMG data were recorded: 4 longitudinal and 4 transverse. Data were collected from pairs of electrodes with three different interelectrode distances: 2, 3 or 4 cm. Reduced numbers of electrodes were also investigated by determining the error rate for every possible combination of channels. For
a given number of channels to be used as an input, the combination with the lowest weighted error was used [96].

Results of this study show that larger interelectrode distances perform significantly better when electrode locations are shifted when compared with electrodes that started closer together. It was also shown that increasing the number of channels available lead to an decrease in error rates, up to approximately six channels. The optimal combination of sites always included at least one longitudinal and one transverse channel. Ideally this study could be further extended to a high-density study using a large array of unipolar electrodes, however clearly the method of selecting reduced subsets of channels would need to be improved so as to provide a time feasible solution [96].

### 2.4 Signal preprocessing and decomposition

A common definition for signal-to-noise ratio (SNR) is the ratio of RMS signal amplitude during an “active” period to signal amplitude during and “inactive” period. This simply assumes that the signal occurs during the active period, while anything that occurs during the inactive period must be noise. This definition however relies on regions of activity and inactivity being identified – almost impossible using an automated system.

Image processing often uses an alternative definition – the mean of a signal divided by its standard deviation, however when a signal has zero, or close to zero, mean, this definition becomes less meaningful [98]. As EMG signals will often have zero or close to zero mean, this definition is not particularly useful. When large numbers of electrodes are used, periods of activity and inactivity are likely to be in very similar temporal locations – so in these circumstances the SNR for a large number of data sets can be calculated with little manual intervention.

Hargrove et al. [99] introduced a method where data were preprocessed using PCA. The dimensionality of the data were increased using this method, and hence a sequential forward selection algorithm was employed in order to choose the 30 dimensions that gave
the best accuracy of a classifier. The single channel that produced the best result was chosen for a one channel classifier, then each of the remaining channels tested in combination with the one channel case, and the channel that produced the highest classification was chosen for the two channel case, and so on. This PCA method was shown to significantly reduce error rates for both amputee and non amputee subjects.

Kim et al. [100] used an energy based local maxima approach to action potential extraction. The energy of each candidate action potential was calculated as the sum of squared values around the maxima. A histogram of the energy for each candidate was then used to create a threshold energy level - candidates with energy above this level were used to create a model template, as the median value at each sampled point in time. The template was then compared to each candidate, and the total error for each candidate calculated. A further histogram of the total error was created, and from this a threshold for determining which candidates were true action potentials.

Data were recorded from three microelectrode recordings, located in the globus pallidus internus, globus pallidus externus and subthalamus nucleus - regions located within the brain. This technique could potentially be applied to surface electromyographic recordings that tend to have much lower SNR due to filtering through the tissue and significant cross talk.

Work has also been investigated into decomposing EMG data into its constituent motor unit action potentials [52, 54, 56, 57, 101-104]. The aim in these studies was to find the firing times of individual motor units. It is identified that signals from a given motor unit will evolve over time, and also different motor units may have similar shapes - making them difficult to discriminate between.

An "Integrated Processing and Understanding of Signals" framework was used in [57] to improve upon an initial round of analysis of precision decomposition (PD). Their work focused on resolving pulse superpositions through a suprasegmental analysis - the results of a segmental analysis were used to estimate the probability of each MUAP train in every segment of data. The probabilities were then used in a "utility" maximisation process i.e. a total weighted value of MUAP timings was maximised. Their results
showed improvements from 90% to 95% accuracy, indicating a reasonable degree of success when this method is implemented.

However it was also stated that progress using the PD technique has "been slow" as analysing one minute of EMG data can often take several hours. Manual editing was often also necessary in order to achieve accuracy rates above 95% - though this has potentially been solved using this suprasegmental analysis. In general, many channels of data were also used to decompose the data – ideally a clinical prosthetic device would use approximately 4-8 channels. As this process requires large amounts of time and potentially manual editing, it would not be appropriate for a real time system.

DeLuca et al. investigated decomposing sEMG signals in 2006 [101]. The data processing by LeFever and DeLuca [54] took 12 minutes to process 1 second of data. Deluca et al. 2006 managed to get processing time to be eightfold the acquisition time. A small array of 4 sEMG electrodes in a 3.6mm grid was used, with differential signals being taken between adjacent channels. The configuration and dimensions were heuristically based. Skin preparation and conductive gel were not required.

It was also identified that in addition to the challenges raised in iEMG there is significant "background clutter" - arising from large numbers of MUs also being detected. The smaller amplitude signals were assumed to be not decomposable, and were considered to be part of the noise. sEMG MUAPs also have more similar shapes and amplitudes than in iEMG. There is also increased amounts of superposition as the connective tissue tends to act as a spatial filter that will increase the duration of APs.

This study utilised a framework called “Integrated Processing and Understanding of Signals” (IPUS), and was developed as "PDIII" - the next system after "PDII" used previously by [57, 58]. PDIII consisted of a bandpass Infinite impulse response (IIR) filter, MAP decomposition, identification of trains that have been "split off" from another train using "trellis traversal", and a final step that takes into account potential complex superpositions that would make the resulting data very different from any other Section of data. Two sets of data were taken in order to determine the accuracy of PDIII: one from sEMG sites described above that was processed using PDIII, and one from iEMG sites.
that was processed using PDII. The decomposition showed an approximate 97% match in AP decomposition for the MUAPs that were able to be detected with the sEMG signals.

Chang et al [52] have provided a method whereby large amounts of "aliasing" -i.e. action potentials were incorrectly identified as having occurred due to the firing of a similar shaped action potential - can be rejected. Their work makes use of a PD-IPUS framework, with the addition of two new rules to reject aliasing. The first rule accounts for "main lobe to main lobe" aliasing, whereby two action potentials have a main lobe at the same point in time. The second rule accounts for "main lobe to side lobe" aliasing, whereby the main lobe of one action potential occurs at the same point in time as the side lobe of another action potential. When this system was applied to a surface EMG recording, precision decomposition accuracy improved from 75% to 83%, with careful manual inspection showing that the errors due to aliasing had been reduced by 85%.

sEMG signals were further investigated using an automated decomposition by Nawab et al in 2010 [56]. The bulk of the algorithm was the same as that reported by DeLuca et al. in 2006 [101]. The template matching method used by Chang et al [52] was then used to identify potential regions where the signal exhibited a correlation with one of the templates from the PD-IPUS step. The resulting MUAPTs were required to meet two criteria: the first that the residual energy (i.e. the difference between the signal and the identified MUAPTs) be relatively low, and the inter-pulse time between of a given MUAPT not be greater than a given threshold. Processing time took on the order of 3 seconds per second of data.

The accuracy of their system was tested by reconstructing the original signal - by taking the firing times of each MUAP and the shape of each MUAP, and finally adding gaussian noise with the variance of the noise set to be the variance of the residual signal (i.e. the original signal minus the reconstructed, noiseless signal). This reconstructed signal was then decomposed by the same method as the original signal. Given that the constituent signals for the reconstructed signals were known precisely, it was possible to identify where the decomposition had an had not correctly identified the original signal. Results
from this reconstruct-decompose method for testing accuracy showed that accuracies ranged between 77% and 97%, with an average accuracy of 92.5%.

Marateb et al. [55] attempted to provide a means to deal with some of the challenges arising in EMG signal decomposition for a single channel iEMG signal. In particular, they looked at superposition/similar shaped APs being identified prior to assigning each AP to a given "cluster" of APs, data-dependent detection of outliers, using firing time information to reduce the possibility of mixed clusters of APs, merging duplicate clusters, and estimating statistics for each cluster separately. By using these techniques, they were able to obtain accuracies above 85% using a completely automated algorithm. Total processing time for a 10 second signal was reported as 19 ± 16 seconds – insufficient for use within a real time system, especially if that system is also required to perform other tasks such as would be required for prosthetic control.

Florestal et al. [103] reported a method wherein 6-8 channels of EMG signals were decomposed into activity from up to 25 motor units. The algorithm has two phases: clustering, wherein recurring MUAPs within each channel were identified and assigned to a given motor unit, and an identification stage wherein matched filtering and superposition resolution techniques were used to identify discharges of the MUAP within the signal. 20 seconds worth of data for three subjects were decomposed. Comparison of the results from this automatic algorithm to expert manual decomposition showed a 95% correlation.

The algorithm was cited as being “fast”, and indeed processing time for the clustering step was on average 15.3 seconds, while for the identification step was 73.8 seconds. However, this still leaves the total processing time approximately 5 times longer than the data length, making it unsuitable for real time use.

A convolution kernel compensation decomposition method was developed by Holobar and Zazula [105, 106], with the accuracy of this system applied to high density sEMG signals being investigated by Marateb et al. in 2011 [102]. In [102] iEMG was simultaneously recorded while an array of 128 sEMG signals were also recorded. To be considered to have correctly identified a “discharge”, the result from the iEMG and
sEMG had to agree to within 0.5ms of each other. Results of this method gave impressive accuracies in the order of 92% when decomposing 644 MUAP trains within the high density sEMG data.

While this method of decomposition gave very impressive results, the required number of signals and data processing time would likely limit the ability of this system to be used in a real time portable device.

Parsaei and Stashuk [59] have investigated eight methods to validate MUAPs using shape information. The methods were based on determining whether or not the shape within a given MUAP were consistent with each other – if not they are assumed to have been classified as invalid. These were tested on simulated and real data, with accuracies of 73.9% and 80.4% respectively for each form of data.

It was however identified that when used on simulated data with very similar shaped action potentials, the accuracy dropped to 20% - indicating that shape information alone may not be sufficient. The authors recommended the use of firing time information in addition to shape information. Processing time for the fastest algorithm was in the order of 70ms per MUAP for 30s of data. This indicates that – subject to the number of MUAPS present – this MUAPT validation method could be used in a real time system.

The majority of these methods have accuracies that would allow for an effective template for use in a matched filter to be produced, however are too slow to currently be implemented in real time. While the study in [59] could potentially be implemented in real time, the accuracy of this system was not as high as that of others. A method could potentially be developed that focuses solely on the creation of a template signal that would both have a high accuracy, and be real time implementable.

2.5 Feature extraction techniques
The initial step in EMG classification is the conditioning of the signal. Increasing signal quality prior to classification is performed through methods such as skin preparation and amplification at the site of signal acquisition. Time domain features are generally favoured due to their relative computational simplicity.

Many different features have been used, though the most commonly used have come from the work of Hudgins et al. [42]: mean absolute value (MAV), root mean square (RMS), MAV slope, zero crossings (ZC), slope sign changes (SSC), waveform length (WL) [49, 66, 107]. An additional feature that has been explored by Vuskovic and Du [67, 68] was to low-pass filter rectified data.

Frequency domain features are generally used to study muscle fatigue and infer changes in motor unit recruitment. The frequency spectrum of an EMG signal is generally influenced by the firing rate of a recruited motor unit and the morphology of an AP as it propagates along a muscle fibre. Power spectral density (PSD) can be calculated as the Fourier transform of the autocorrelation of a signal. The mean and median of a PSD provide information about the change of a signal over time. A periodogram can be used to estimate the PSD of a signal, using the square of its Fourier transform divided by its length. The myoelectric signal is generally considered non-stationary, though for low levels of maximum voluntary contraction (20-30%), and short duration (20-40s), it can be considered wide-sense stationary, i.e. the population mean and covariance do not change with time [108-110].

Frequency domain analysis causes time information to be lost; meaning the temporal location of an event will be lost. Various methods of time-frequency analysis give some amount of temporal and spectral information. Short time Fourier transforms (STFT) provide time and frequency information; however their resolution is limited by the window size. Wavelet transforms (WT) offer an alternative to STFT that reveal information in different frequencies, as defined by its “scale” [111, 112]. The Wavelet packet transform (WPT) is another time-frequency feature extraction method similar to WT, with additional filters used. PCA was used by Englehart et al. [81, 83] to reduce dimensionality prior to classification via time domain features, STFT, WT and WPT.
PCA was again used by Chu et al. [80] in conjunction with a self-organized feature map (SOFM). In this study, EMG data were recorded from four sites; targeted at the extensor digitorum, extensor carpi radialis, Palmaris longus and flexor carpi ulnaris. Wavelet packet transform features were then extracted, followed by feature reduction using PCA. The PCA reduced data sets were then transformed using SOFM; this gives the new feature space high class separability. Finally, the data were classified using a multilayer neural network. This method achieved error rates of approximately 3%, slightly greater than for the same method but without PCA. The SOFM method presented gave a significant improvement on the previous methods; however the data acquisition and classifier methods used were similar to what has been presented previously.

It has been identified by Englehart et al. [82] that steady-state data are classified more accurately than transient data. This study identified that the performance of a classifier is affected more by choice of feature set rather than the classifier used. Data were taken from four channels, placed on the lateral, medial, posterior and anterior surfaces of the forearm, and processed using time domain or time-frequency domain features, followed by classification by LDA or multilayer perceptron (MLP) neural networks. Results showed that PCA-reduced wavelet packet transforms features using LDA outperformed other techniques (time domain, STFT and WT features classified using LDA or MLP); however the location of electrodes appears to simply be targeted at acquiring data from sites spaced around the forearm.

The study by Ferguson & Dunlop [69] presented a method that uses high common mode rejection ratio instrumentation amplifiers and feeds the inverted common mode signal from each electrode pair onto the arm; this very effectively cancels out unwanted signals. STFT, autoregressive (AR) model coefficients and wavelet features were compared, followed by pattern recognition using either a neural network or Mahalanobis distance based statistical model. The Mahalanobis distance is a distance measure that normalizes the distance between a point and a cluster of data, taking into account the variability of the data cluster in the direction of the single point. Data were taken from four sites, above extensor muscles in the forearm, corresponding to thumb movement (x2), index and middle finger movement, and little finger movement.
This study identified that repeated correct location of electrodes was important, and that application of conductive gel can cause slipping of the electrodes. It also identified that while these electrode sites are suitable for able-bodied subjects, amputees may have different anatomy and hence require different electrode locations. There was, however, no investigation into the affect of changing these locations, or into where data should be acquired from for an amputee.

Various model coefficients have been used as a feature set. Huang et al. [113] and Chan and Englehart [76] have used Autoregressive model coefficients. Autoregressive models take the form of:

\[ x_t = c + \sum_{i=1}^{p} a_i x_{t-i} + n_t \]  

Where \( c \) is a constant, \( p \) is the order of the model, \( n \) is white noise, and \( a_i \) are the model coefficients. These coefficients are then used as the feature sets. While these methods have achieved error rates between 3\% and 6.75\%, the data acquisition step simply took data at four equally spaced sites around the wrist.

### 2.6 Classification methods

Following extraction of features, data must be classified into classes corresponding to the grasp types to be determined. Ajiboye and Weir [73] highlighted that multifunction prosthesis control can only exist when control of one function does not affect control of any another. There must be a balance between choosing sites the user can easily elicit and those that will most accurately correspond to the intended movements of the user. In addition, signals corresponding to the same grasp may change due to factors such as sweat, muscle fatigue and electrode placement.

It has been shown through EEG-EMG studies that there is an alteration in cortico-muscular coupling related to fatigue [114]; the effect of long-term effects over a course of
21 days on pattern matching algorithms has also been investigated [115]. This study showed that state-of-the-art (in 2010) algorithms tended to have degrade in accuracy over time. Ideally, classifiers must also be able to cope with these additional variations. It should be noted that algorithms that perform well on one subject may not perform very well on others. Classification generally occurs through pattern recognition methods, though Qizhu et al. [116] and Zhang et al. [28] have presented a method for determination of muscle contraction onset, and Chan and Englehart [76] presented a probabilistic Gaussian mixture model based approach.

The study by Qizhu et al. [116] considered the effect of an action potential as it propagates along a muscle fibre, and the effect this has on determining the onset of contraction. This study used a single threshold decision method, two threshold decision method and maximum value detection algorithm to detect contraction onset. Data were taken from the biceps brachii; as such it would likely need further investigation before it could be incorporated into a prosthetic device. The study by Zhang et al. [28] took data from left and right calves, and left and right shoulders, again not ideal for prosthetic control. Data were then processed using an adaptive threshold method based on the power of the EMG signal in order to determine contraction onset.

The study by Chan and Englehart [76] used the same data as the study in [81], from four equally spaced sites around the forearm. RMS and autoregressive coefficient features were extracted using 256ms overlapping windows. Classification then occurred using a Gaussian mixture model, where signals are classified according to whichever model has the highest probability of generating the given signal. This has the advantage that it can obtain high accuracies (≈90%) with a relatively simple method. However, as with many other studies presented, the location that data were acquired from was simply evenly spaced locations around the forearm.

Hidden Markov models (HMMs) were used in [77, 117]. The Markov property refers to the memoryless property of a stochastic process. A process is said to be a Markov process if the future state of the process depends only on the current state of the process, rather than on previous states or how the current state was obtained. A hidden Markov model is
then a model of this process, with unobserved states. The study by Chan and Englehart [77] used the HMM as a means of determining the transition from a current limb motion to the future limb motion.

Many of these techniques (such as self organizing feature maps and fuzzy classification) have also been used for detection of epileptiform events from EEG signals (i.e. detecting parts of an EEG signal that could mean the patient has or will develop epilepsy), a problem that has parallels to the classification of hand movement from EMG signals [118-122].

2.6.1. Neural networks

Neural networks have long been used as a classification method. They allow simple artificial intelligence to be used for learning tasks. Multi-layer perceptrons (MLPs) are a common neural network, consisting of multiple layers of neurons, with each neuron in a layer connected to all neurons in the surrounding layers. The neurons in this case act as weighting functions for the various inputs given to them; multiplying each of the inputs by the neuron’s associated weight before passing it to the next layer of neurons. The network is trained through back propagation, where these weights can be altered such that the network more accurately recreates the desired outputs for the training inputs.

MLPs have previously been applied to electromyographic patterns in several studies [42, 92, 123], however LDA and radial basis function (RBF) have been considered as an alternate to MLP networks in several studies [75, 82, 124, 125]. In addition, [43] refers to time-delayed artificial neural networks used in [126]. These networks were fed raw EMG rather than features, and were able to accurately classify any nonlinear input/output combination. However, the time delay is in the order of a second, making them inappropriate for real-time use. Unsupervised algorithms have also been used for pattern recognition [79].

2.6.2. Fuzzy approach
Ajiboye and Weir [73] presented a heuristic fuzzy logic approach to signal classification. Their study involved selecting myoelectric control sites within the geometry of a standard prosthetic socket. The sites were easily elicited by the user and created the best signals for use of EMG classification.

The motions chosen to elicit were:

- Wrist extension;
- Ulnar deviation;
- Finger flexion;
- Wrist flexion.

The muscle groups chosen for myoelectric pickup were:

- Extensor digitorum;
- Extensor carpi ulnaris;
- Flexor digitorum superficialis;
- Flexor carpi ulnaris.

Sites were located based on the medial and lateral humeral epicondyles.

This study gave possibly the most realistic application of a myoelectric control system published to date; the application of such a system to limb deficient subjects is the key outcome for myoelectric control. For these subjects only three independent surfaces on the forearm could be obtained. Trials covered the subjects’ full range of motion. The study used a mirroring technique on the contralateral limb for the limb deficient subjects when performing tests in order to visually aid their function and covered the full range of motion with the contralateral limb [73].

The electrode sites for two limb-deficient subjects were determined by palpation while the subjects performed the appropriate contractions. This was again an attempt to
determine electrode location based on where active muscles are; however this approach did not take into consideration the affect that the skin and connective tissue may have on the signals.

The fuzzy classifier can be summarised as follows:

1. Input membership functions that convert numerical inputs to linguistic variables – these linguistic variables corresponded to OFF, LOW, MED or HIGH, associated with the degree of activation.

2. Interference rule to perform pattern classification, returning linguistic outputs and associated degrees of truth – i.e. each output gave the OFF, LOW, MED and HIGH variables a degree of truth.

3. Output membership to defuzzify the degree of truth outputs and convert to a numerical value.

4. A fuzzy c-means clustering attempts to group data together in order to minimize variance within a cluster while maximizing variance between data in different clusters.

The resulting classification assigns a certain degree of membership for an input data set to each output class, corresponding to how much the classifier considers the input data to belong to each output. Another study by Khushaba et al. [127] used fuzzy LDA for classification, and found that the results they obtained outperformed uncorrelated LDA slightly, and both outperformed PCA.

2.6.3. Fuzzy neural classification

Vuskovic and Du [68] used a fuzzy ARTMAP approach to signal classification. Their previous work [86] used Mahalanobis minimization to classify grasp types where the clusters are elliptically shaped (cylindrical, spherical, precision (pinch), lateral (key) and hook), however identified that for more general cluster shapes stronger methods would be needed. They also identified that back propagation perceptrons were an attractive method;
however the success rates for such networks was 1-5% above the success rate of the Mahalanobis classifier and the training of back propagation networks could be unpredictable and did not always converge. Back propagation networks also must be structured before they can be used (number of hidden layers and neurons in each layer must be predetermined), and networks are not adaptive. Fuzzy neural classifiers have been used in several other studies [78, 128, 129].

ARTMAPs are a form of neural network (NN) that consist of two Adaptive Resonance Theory (ART) classifiers, which are also a form of NN. ARTs are an unsupervised learning system that assigns inputs into outputs based on their similarity to that output group. The vigilance parameter controls the learning of the network; a high vigilance parameter will allow for less dissimilarity between inputs and the target output, resulting in more specific detailed memories, while a low vigilance parameter will result in less detailed memories, however will obviously require less output groups to classify all possible inputs. Fuzzy ARTs extend ARTs through the use of fuzzy logic and complement coding; Fuzzy ARTMAP is a supervised learning network, combining two fuzzy ARTs with one taking the input data and the other taking the output data. The two ARTs are combined through the output unit controlling the vigilance parameter of the first unit.

Fuzzy ARTMAPS were modified for use in EMG classification [68]. The algorithm was efficient and robust with respect to sensitivity, size and ordering of test sets. The study in [68] using simplified fuzzy ARTMAP (SFAM) based on Euclidean distance had an increase in success rate by 1%; however the number of output neurons required for classification increased considerably. This was put down to the elliptic nature of the clusters. SFAM based on Mahalanobis distance further reduced training time, and improved success rate by between 8 and 17%. While this study applied a novel classification method, the electrodes were again located above targeted muscles (Extensor Pollicis group, Extensor Communis Digitorum x2, and Extensor Carpi Ulnaris), and the feature extraction method used was “one of the simplest”: squaring the signal, followed by a moving average FIR filter and a hamming window function of size 300ms.
2.7 Data reduction algorithms

Section 2.3 highlighted the need to investigate the placement of electrodes. In order to perform this investigation, it is necessary to collect a large amount of data; however dealing with large numbers of data sets could lead to long processing. Hence, a review of algorithms available to reduce input data dimensions is necessary. A graph theoretic approach to data pruning was presented in [130]. It was identified that training NNs to be accurate requires large amounts of training data, which is time consuming, hence the need to prune data sets.

A 2003 investigation into sample selection for RBF neural network classification [131] identified that neural network classification cost is high, and increases are proportional to the number of training samples. Also, large numbers of input samples ensure high classification accuracy. The goal of the study was to decrease the size of the training set, in a similar way to what is required for this thesis. Several methods were referred to for improvement of NN classification accuracy including adding input samples from regions not yet covered in training set [132], and adding more input samples from regions that the classifier could not classify correctly [133].

Removing redundant input samples can be achieved through methods such as measuring the similarity or distance between samples. An example of this is active learning where the classifier selects training input samples. For labelled samples, this starts with 0 training samples, with samples selected according to quantitative criteria. Samples continue to be added until it is well trained. For unlabeled samples, the learner selects samples and requests labels prior to learning.

Feature and sample selection have previously been carried out as independent processes, i.e. treated as two separate problems. Both of these processes essentially reduce the size of the data set. NN sensitivity analysis is a measure of output sensitivity to changes in inputs and weights, and can be used in NN design, e.g. choosing number of hidden neurons, connecting weight selection and network pruning [133]. This can be done using either a stochastic analysis or partial derivative analysis. Stochastic analysis finds a
relevant feature subset at one pass prior to network training, while partial derivatives find it iteratively during back propagation training.

Zurada et al. [134] extended sensitivity analysis to include feature reduction, while Ng et al. [135] proposed an input feature selection method to select relevant features from the original data set. The sensitivity analysis however assumes that input data sets are independent of each other, and as such it has a limited usefulness.

### 2.8 Summary

Prosthetic devices have been in use for centuries – the wooden prosthetic worn by the heroic Captain Danjou 150 years ago is still paraded once per year; however their use for anything more than aesthetics, in particular the creation of myoelectric prostheses, has only occurred within the last few decades. While initial myoelectric devices provided a single DoF actuation, more advanced prosthetics are attempting to replicate the full functionality of the human hand.

There have been some previous attempts to preprocess EMG signals prior to classification that have shown promise. In addition, there has been ongoing work into identifying motor unit action potentials and decomposing the surface EMG into its constituent MUAPs. While these methods have shown great success, the processing time required is beyond what would be employable in a real time system as required for prosthetic control.

The approach to electrode placement has generally been to place them above the muscles most likely to elicit useable signals during hand function, or to place an array of radially symmetric sensors around the forearm; while the symmetrically placed sensors have shown to perform as well as anatomically targeted sensors, few studies have looked at placing sensors in other locations. One study [46] has performed a high density study of forearm EMG signals, though the data set reduction algorithm used to choose optimal sites may not be ideal.
A review of data set reduction algorithms has been presented. From this brief overview, a suitable method for the reduction of large numbers of EMG datasets to a smaller number of real-time measurable data sets requires additional investigation, and the development of a new algorithm for “large non-independent data set optimization” is required. While there were previous attempts at optimizing electrode placement, they may optimize to a local maximum rather than a global maximum. One of the goals of this thesis is therefore to provide an improved heuristic algorithm, in order to determine if these previous algorithms perform well enough or if a more sophisticated algorithm is required.
3

Action potential extraction through iterative filtering

“A mass of data sufficient to accurately classify hand movements may be far more than necessary, however what better place to start?”

OEPEPC, 22/01/2011

Summary

It is hypothesised that a matched filter could be used to improve accuracy in EMG prosthetic control. As a matched filter requires a template signal to be used, this Chapter deals with the extraction of an “Action Potential” with the aim being to provide a filter that can adapt to the dominant activity within a subject’s data. As the template would contain a single shape of action potential, this method assumes that the dominant activity comes from a single MU, or several MUs with similar shaped action potentials. This Chapter also aims to determine if this single action potential assumption is sufficient or not to improve accuracy in a prosthetic control system.

EMG decomposition has been used now for many decades [52, 54, 56-59, 101, 136]. While these methods have exhibited reasonable degrees of success, they tend to be time consuming and relatively complex due simply to the nature of EMG signals – generally consisting of multiple overlapping motor units of different shapes. This Chapter aims to provide a method for determining a first-order “action potential” estimation, in order to determine if it is possible to improve prosthetic control accuracy using a less complicated method that assumes a single shape of action potential.

As a significant amount of effort in decomposition is given to resolving complex superpositions of action potentials, the method designed here aims to avoid this by simply
determining the onset of periods of EMG activity. By taking the data from the onset of “AP” activity, and averaging all of these points within a data set, an estimate of a true action potential can be obtained. As the aim of the AP extraction step is to provide a template action potential, the measure of success for this algorithm will be:

- The ability to correctly identify some number of “action potentials”.

- The ability not to incorrectly identify background conditions as an action potential.

- The ability to improve the signal-to-noise ratio of a surface EMG signal in real time. This shall be measured by having an overall processing time less than 10% of the signal length.

As the algorithm is not attempting to decompose superpositions into constituent action potentials, it would not be appropriate to measure its success based on its ability to correctly identify every action potential.

This estimate is then used in a matched filtering step – essentially picking out regions of the signal that are similar to the action potential while diminishing areas that are not. While this is a method that has been commonly used, it is extended here to make an iterative “detect and filter” method for determining an action potential template.

This new method of iteratively extracting and filtering is applied in this Chapter to simulated data in order to show that it is able to pick out the onset of true EMG activity without prior knowledge of the signal. This is compared to the results when decomposing the EMG signal into its constituent MUAPs. It is then applied to data taken from 10 subjects to show that in addition to providing a good estimate to the overall action potential shape within a signal, it is also able to improve the signal to noise ratio in these signals significantly.

Finally, to test the effect of this system on a classification system, further data from 10 subjects was used as the input for a classification system to automatically detect which of three grasp types the subject was performing. This was tested using data that had been preprocessed using this matched filtering technique, and comparing it to data that had not had this preprocessing performed.
3.1 Introduction

Specific muscle groups control various hand movements. While many of these are intrinsic to the hand, muscles within the forearm control the majority of the bulk movement of the hand [137]. These bulk movements can therefore be determined using electromyographic signals taken from the forearm. However, the skin and connective tissue acts as a spatial filter, and there are person-to-person variations in signals observed on the skin, so the exact location of ideal signals cannot be predetermined. In order to determine the optimal locations to measure signals on the surface of the forearm, a method has been developed to acquire and process data from as many sites on the forearm as possible while bulk hand movements are carried out, and process it using an iterative action potential detection, followed by matched filtering.

Noise arises in electrophysiological recordings from a variety of sources. Intrinsic to the body are other electrophysiological signals occurring remote from the site of detection, e.g. deep muscle tissue electric fields, and electrocardiographic signals, along with superposition of other EMG signals that have passed through significant amounts of tissue filtering. Extrinsic sources of noise include mains hum, thermal noise and shot noise (i.e. electrical signals are quantized, so in small signals such as EMG, there is a stronger likelihood that a given electrical “packet” will follow its statistically expected behaviour. The end result of this is that the total signal may have small amounts of noise corresponding to one of these packets not behaving as expected).

APs are characterised by a rapid upstroke, followed by a downstroke with a slight undershoot. These signals relate to the muscle contraction commanded by the brain. Hence, from EMG recordings, there is the potential to infer the intention of the brain [14].

If the response of the muscle cells to chemical stimulation \(h(t)\) can be extracted from a noisy signal, it can then be used to infer the commands from the brain \(\delta(t)\)[138].

Previous works have applied matched filtering to extracellular neuronal recordings. This technique uses an energy-based action potential detection algorithm and thresholding to detect the temporal position of action potentials [53, 100].
In order to distil APs from noisy EMG data we have combined an action potential upstroke detection algorithm with the use of matched filtering. Matched filtering is a technique that cross correlates a known “template” signal with an unknown signal (i.e., it convolves a time-reversed “template” signal with the unknown signal). As convolution in the time domain is the equivalent of multiplication in the frequency domain, this process can be thought of as amplifying those frequencies that are present in the template signal, while attenuating those frequencies not present. In this way, the unknown signal gets “matched” to the known signal.

A Matlab routine was written to identify likely action potentials based on a characteristic upstroke in electromyographic data collected from a human forearm. This action potential was then used as a template for matched filtering of the EMG data, and this filtered data set was used to extract a new improved template based on the same action potential detection algorithm.

This Chapter addresses signal processing techniques to be used in the initial stage of myoelectric control. Mathematical justification for the techniques used is presented in the equations that follow. While these may be difficult to follow, an attempt to graphically present the effect of this signal processing is presented throughout the Chapter.

In order to understand how matched filtering works, an explanation along with a mathematical framework is presented here. As described earlier, matched filtering can be used to detect the presence of a known (template) signal $h(t)$ within an unknown signal $s(t)$. It is obtained by correlating the time reversed template signal $h(-t)$ with the unknown signal; the equivalent of convolving the unknown signal with a time-reversed version of the template. The affect of matched filtering a signal with itself is shown in Figure 11.
The ideal matched filter response to a single impulse response is \( f(t) = h(t) \otimes h(-t) \), i.e. the template signal being matched filtered with itself. Now, if \( u(t) = |f(t)| \), then \( v(t) = u(t)_{LPF} \) is the ideal detected matched filter output to detect the impulse \( \delta(t) \).

Given that the effect of convolution in the time domain is to perform multiplication in the frequency domain, i.e. \( s(t) \otimes h(-t) \leftrightarrow S(\omega)H(-\omega) \). Hence, we would expect the output of the filtered signal to also peak at the same frequency as the unfiltered signal, but roll off at twice the rate of the original signal, due to the multiplication of two identical frequency profiles.

Given a noise free signal that is made of multiple occurrences of an impulse response, the signal can be modelled as:
where $A_i$ is the amplitude, and $t_i$ is the time shift; to demonstrate this for a simulation, both are generated using a Gaussian random number generator, and $k$ is the number of known signals to be included.

From this signal, the series of original impulses $V_k(t)$ shall be detected as follows:

$$ f_k(t) = s(t) \otimes h(-t), $$

$$ f(t) = \sum_{i=1}^{k} A_i h(t - t_i) \otimes h(-t) $$

$$ f(t) = \sum_{i=1}^{k} A_i f(t - t_i) $$

$$ u_k(t) = \sum_{i=1}^{k} A_i |f(t - t_i)| $$

$$ v_k(t) = \left( \sum_{i=1}^{k} A_i |h(t - t_i) \otimes h(-t)| \right)_{LPF}. $$

### 3.2 Data Acquisition

In order to obtain data from a series of points on the forearm, a 3D model of a subject’s forearm was taken using a Faro Laser Scan Arm (Faro, USA) and a replica created using a Dimension Elite 3D printer (see Figure 12). Dow Corning flowable sealant 734 was then used to create a silicone mould around the replica forearm, with electrodes
embedded in the silicone so that they would be held against the surface of the subject’s arm when the sleeve was worn. The armband is shown in Figure 13.

Figure 12: The printed forearm replica (white, on left) and the Dimension elite 3D printer (on right)
The Faro Laser Scan arm was then used to measure the x,y,z co-ordinates of the electrodes. These were referenced to several pronounced structures on the forearm (namely, the ulna styloid process and head of the radius distally and the Olecranon proximally). These locations are demonstrated in Figure 14. The x and y coordinates were used to refer to distance in plane through the arm, while z was used to refer to distance from the Olecranon – i.e. the large bony prominence of the elbow. The electrodes were then indexed to channels in a system developed in the University of Auckland for filtering, amplification, measurement and processing of biopotential signals (UnEmap), to ensure the recorded data would be matched to the correct position on the forearm.
The following procedures were approved by the University of Auckland’s Human Participant Ethics committee, reference number 7932. A copy of the approval, along with the consent form and participant information sheet can be found in appendix 3. The subjects’ skin was prepared by first removing the hair, and then scrubbed using Nuprep skin preparing gel (D.O Weaver & Co.). A total of 128 surface electrodes were then connected to the UnEmap system. The data recording settings were as follows:

Sampling rate: 5KHz
Anti-aliasing filter: 1KHz
Gain: 88

Unipolar data were recorded from each of ten subjects’ forearms, from 128 approximately evenly spaced electrodes. Electrodes were Ag/AgCl, approximately 10mm in diameter. For details on the ethical approval and subjects, please see appendices 3 and 4 respectively. The subject was asked to perform each of the grasps “pinch”, “cylinder” and “sphere”. Subjects were asked to perform the grasps at a moderate level of intensity. Contractions were isometric, and lasted approximately 10 seconds. Each grasp type was repeated, to provide two sets of data. Subjects consisted of 9 males and 1 female, between the ages of 20 and 30.
Chapter 3 Action potential extraction through iterative filtering

The measured voltages were relative to the Olecranon. Before signal processing was applied, the channels with useful data were identified. To this end, plots of the potential against time for each channel were produced, and those channels found to vary across extremes of range (rail-to-rail) were discarded. Generally, this was in the order of 10 channels per data set.

### 3.3 Template extraction

#### 3.3.1 Methods

A sample EMG data set is shown in Figure 15. In order to perform matched filtering on such a data set, a template signal was required. In human biopotential signalling the most dominant repeated signal is the action potential. Thus, in order to have a template signal, it is necessary to extract a representation of an AP from a relevant data set. The surface EMG however consists of multiple APs superimposed and filtered through the connective tissue before it reaches the surface of the skin; thus any AP extracted from a surface EMG data set should be considered as the summation of several individual APs with various time delays.

![Figure 15: Raw EMG data used to extract action potential and the associated spectrum.](image)

The simplest way to extract an action potential was visual inspection to time-average multiple signals that were believed to have an action potential. The result of manually tagging all “action potentials” and averaging them can be seen in Figure 16.
The ‘extracted’ action potential was then used as the template for the first matched filtering pass. By filtering the original EMG data, a better estimate of the action potential locations was obtained. The estimate was then refined by once again manually tagging these action potentials wherever one was found, and averaging all of them. This process can be summarized as follows:

1. Using EMG data, first filter for mains frequency noise and DC offset,
2. Mark all occurrences of action potentials,
3. Average all action potentials that are detected, from the initial point where it was marked, for the next 12ms. As the duration of an action potential is in the order of 2-3ms [14], recording the data for 12ms ensures that the entire action potential is recorded,
4. Perform matched filtering using the averaged “action potential” found in step 3,
5. Repeat steps 2-4 10 times.

However, this method relies on manual detection of APs, a process which is subjective and time consuming. To eliminate this step, an algorithm was developed to automatically detect the rapid upstroke that is characteristic of action potentials. This algorithm is illustrated in Figure 17. This algorithm is repeated until the algorithm converges, or a predefined number of iterations has been exceeded. Convergence is assumed to occur when the sum of residuals between the previous action potential and the new action potential falls below $1 \times 10^{-3}$.
Chapter 3 Action potential extraction through iterative filtering

Figure 17: Flow chart of the extraction algorithm.

Calculate the derivative with respect to time by backwards differencing

For each time point, determine if the time derivative exceeds the threshold – the threshold is calculated as 20% of the maximum derivative value

If derivative exceeds threshold, ensure derivative is positive for next 0.4 milliseconds

Match the peak values of all spikes in time, and average all spikes that are detected

Perform matched filter using the averaged “action potential” found in the previous step and use this filtered signal as the starting signal for the next iteration
3.3.2. Results and Discussion

Ten iterations\(^1\) of filtering produced the action potential shown in Figure 18.

![Figure 18: Action potential detected after the tenth iteration](image)

The action potential in Figure 16 was extracted from the data set that was given in Figure 15. The locations of all of the action potentials are shown overlaid with the original data set (Figure 19).

The initial extracted action potential estimate (Figure 16) exhibited the upstroke used to determine action potential locations. This signal exhibited a duration in the order of 5 milliseconds.

![Figure 19: Original data set, and location of detected action potentials.](image)

\(^1\) 10 iterations were used in order to provide a timely demonstration of the system.
Iterative application of a template extraction algorithm in combination with matched filtering has the potential to greatly enhance the SNR of sEMG signals. A new upstroke detection algorithm has been used to find an approximation to action potential temporal location. Similar to previous works [53, 100], this has then been extended to provide an aggregate action potential for use as a template in matched filtering. In addition, with the original data set run through a matched filter using this template, the original action potential detection algorithm has been applied iteratively to converge on a much cleaner action potential template. The algorithm can be applied to multiple data sets and continue until a specified SNR is achieved.

Normally, an undershoot would be expected in an action potential, and is referred to as hyperpolarisation. However, as the signal used in this study started with a normalized amplitude of -1, and ended at 0, it was slightly different to a conventional action potential – this would be expected to have an ending value approximately the same as the start value [14].

The final converged AP signal also featured an upstroke, and had an undershoot. The duration of this converged signal appeared to be approximately 2-3ms, corresponding well with the expected duration of an action potential. This signal levelled out to approximately zero much faster than for a single iteration. Also, while it still had a disparity between start and end amplitudes, the hyperpolarisation was much better developed than in the first iteration.

### 3.4 Application to simulated data

#### 3.4.1 Methods

To demonstrate the matched filtering process, a sample 2\textsuperscript{nd} order system representing the action potential of a nerve signal was modelled by:

\[ h(t) = e^{-1\times10^3t} \sin(4\times10^3t), 0 < t < 5 \]  \hspace{1cm} (9)
Where \( h(t) \) is the action potential, and \( t \) is time. In this equation, the \( 10^3 \) means that we are working in milliseconds rather than seconds, and the ‘4’ is the equivalent of our angular frequency – giving a frequency of \( \frac{2}{\pi} \), i.e. an approximate period of 1.5 milliseconds. This gives us a decaying sinusoid, with a rapid upstroke, followed by hyperpolarisation and a return to the original state.

Equation (3) was used to create an unknown signal containing multiple occurrences of the impulse response given in (9). Each data point was set to have a 5% chance to be the origin of an action potential. To generate a more realistic signal, Gaussian noise was added to the signal, with an RMS value equal to the RMS value of the signal, based on methods found in [43, 53, 56, 63-66, 100, 112]. An example of the generated data can be seen in Figure 20.

![Simulated data](image)

Figure 20: Simulated data

The action potential extraction algorithm from Section 3.3 was then run on this simulated data. In order to determine a measure of “accuracy” for the algorithm, the number of false positives and true positives were calculated. As the measure of success of the algorithm is based on its ability to reject false positives while being able to accurately determine true positives, it was not deemed appropriate to consider the number of false negatives. The
most appropriate measure of accuracy was deemed to be the “Positive Predictive Value” given in equation (10):

\[
PPV = \frac{TP}{TP + FP}
\]  

(10)

Where PPV is the Positive Predictive Value, TP is the number of true positives and FP is the number of false positives. Given that the accuracies reported tend to be a product of two such quantities, the square of the PPV, multiplied by 100 was used as the final accuracy measure.

The algorithm was originally run on the data described above, with each data point having a 5% chance of being the origin of an action potential. It was found however that it was difficult to accurately determine if a positive result was a true positive or a false positive, simply due to the number of action potentials in the original signal. In these cases the algorithm certainly converged to what qualitatively appeared to be close to the original simulated action potential, however no meaningful quantitative measure of performance was able to be obtained to compare to previous works.

The simulation algorithm was modified to give a 0.5% chance of each data point being the origin of an action potential. The RMS value of the noise added to the signal was kept constant as the RMS value of the signal when generated with a 5% chance of having and action potential at each point. The simulation algorithm was run 20 times in order to provide different signals for the extraction algorithm to be tested on. For each simulated signal, the extraction algorithm was run 40 times, each time increasing the maximum number of iterations allowed, starting from a single iteration.

As a final test of the extraction algorithm, it was run on a data set consisting of Gaussian noise, without the presence of the simulated signal. This was intended to ensure that the extraction algorithm was in fact extracting the simulated signal due to its presence, rather than extracting the simulated signal due to over rigidity of the extraction constraints – e.g. if the constraints placed on the AP detection algorithm included the upstroke, duration and downstroke, then all extracted signals would look very similar.

3.4.2. Results and Discussion
Five milliseconds of the impulse response (9) are shown in Figure 21 for a 100kHz sampling rate. The simulated signal generated by multiple occurrences of (9) is shown in Figure 22. The effect of adding Gaussian noise (mean = 0V, RMS amplitude = 0.8V) to this signal is shown in Figure 23.

Figure 21: The ideal impulse response for the simulated action potential.
Chapter 3 Action potential extraction through iterative filtering

Figure 22: Response of system in Figure 21 to a series of randomly spaced, random amplitude impulses.

Figure 23: Response of system in Figure 18 to a series of randomly spaced, random amplitude impulses, with added Gaussian noise.
The results of applying the extraction algorithm to noiseless data are shown in Figure 24, with the original simulated action potential shown in black, and the extracted action potential shown in red. This figure shows the rapid upstroke that is being selected for is present, followed by the downstroke and oscillations as present in the original signal. Although there is a difference in the roll-off rate of the oscillations, the extracted action potential is very similar to the original signal, i.e. the extraction algorithm has performed very well.

The affect of the iterative application of template extraction and matched filtering is shown in Figure 25. The action potential detected after the first round of template extraction is shown overlaid with the action potential detected after the 5th iteration. The signal after the first iteration is a very poor estimate of the original simulated action potential, however after the 5th iteration it shows a very good correlation with the signal seen in Figure 24. This shows that while the original estimate was very poor and noisy, the iterative improvement has allowed for a much improved estimate of the true signal.

Figure 24: Simulated action potential shown in black, overlaid with the action potential extracted from the simulated data set in Figure 22.
Chapter 3 Action potential extraction through iterative filtering

Figure 25: Action potential detected after the first iteration, and the 5th iteration

Figure 26: Accuracy and processing time versus number of iterations. Error bars indicate ± 1 standard deviation
Figure 26 shows the result for accuracy and time taken when increasing the maximum number of iterations allowed. This clearly shows that accuracies obtainable by this algorithm are in excess of 95%. There is a roll off at 5 iterations, after which allowing additional iterations does not improve the accuracy at all. The accuracy using this method converged to (on average) 97%. The time taken to complete the entire algorithm, using 5 iterations maximum was on average 333ms. The length of the data segment was approximately 8s, meaning that the algorithm performs within the desired 10% (i.e. 800ms) of the total time allowable. Given that EMG decomposition studies in the literature [55-59, 101, 136] report accuracies between 75 and 95%, the algorithm here has performed very well, however it must be remembered that it is focused only on providing a template AP signal in order to perform matched filtering. This method then has the potential to be useful in applications such as field monitoring of electrophysiological signals in particularly noisy environments.

Figure 27 shows the output of the extraction algorithm performed on the Gaussian noise. This figure clearly shows that the algorithm detects the rapid upstroke (although it has a much greater gradient than the action potentials) as is expected, however, any other characteristics of the simulated signal are absent. From this, it is concluded that the algorithm is clearly able to extract the desired template from a signal comprised of that signal superimposed with various time delays and amplitudes, in addition to additive noise.
Chapter 3 Action potential extraction through iterative filtering

3.5 Application to real data

3.5.1. Methods

To demonstrate the efficacy of the matched filtering algorithm it was used on raw data from sEMG recordings taken on a human forearm. The SNR of a data set expressed in dB is:

\[
\text{SNR} = 20 \log_{10} \left( \frac{A_{\text{signal}}}{A_{\text{noise}}} \right)
\]

Figure 27: Affect of running the extraction algorithm on Gaussian noise without the presence of the simulated action potential.

The SNR was estimated by taking \( A_{\text{signal}} \) to be the amplitude during periods where the muscle activation signal was occurring, and \( A_{\text{noise}} \) to be the amplitude during periods where no muscle activation signal was apparent. As it was difficult to determine the true amplitude, a root mean squared (RMS) value was taken across all values during these periods, ensuring that the periods were of the same duration.
The algorithm was run on data from each of the 10 subjects, with areas of activity and non-activity being manually marked.

### 3.5.2. Results and Discussion

A sample channel of data are shown in Figure 28, with the data after filtering shown in Figure 29. The action potential extracted from this data set is shown in Figure 29. Notice that the signal level during the periods of “rest” have been markedly diminished, while the periods of activity have remained relatively untouched.

**Figure 28:** Sample of data before filtering via iterative AP extraction
Figure 29: Data set from Figure 28 after matched filtering. Notice the decrease in amplitude for the periods of rest.

Figure 30: Action potential extracted from data set in Figure 28
The average SNR of the 10 signals before filtering was calculated as 8.3 dB, with a standard deviation of 2.4 dB, while the average SNR after filtering was calculated as 10.5 dB with a standard deviation of 3.1 dB. The average increase in SNR was 2.2 dB. Processing time was on average 6.1% the duration of the signal, again within the specified 10%.

The application of a matched filtering technique to EMG data has shown an improvement in signal quality. While previous work had applied a similar technique to extracellular neural recordings, the work presented here shows that the same approach can be applied to noisier signals coming from surface EMG recordings.

The increase in SNR by the application of the extracted template allows faster extraction of clean APs. This has implications for control of robotic devices where signals could be taken from multiple adjacent sites in order to provide control signals to multi-degree-of-freedom prosthetic devices.

3.6 Application to prosthetic control

3.6.1. Methods

While action potential extraction algorithm has shown great potential for increasing the SNR in EMG signals, it remains to be determined if this increase in SNR will improve the accuracy for a prosthetic control system. To this end, the previously collected data were further processed in order to determine the accuracy of a classification system both with, and without, preprocessing the data by matched filtering with the detected burst onset signal.

Subsets of the data were used as inputs to a neural network, using 1 channel of data, up to and including 10 channels of data. The subsets were chosen to be symmetrically positioned around the proximal aspect of the forearm, as in [44, 46].

The neural network was created using the Matlab newff function, and consisted of 1 input layer with 3xn neurons (corresponding to each of the 3 features from the n input channels)— see Section 2.6.1 for an explanation of neural networks, 1 hidden layer with 12
neurons, and 1 output layer with 3 neurons using Levenberg-Marquardt\(^1\) training; mean squared error performance and random data division.

The first data set was used as training data, while the second data set was treated as unknown data to be classified. An associated output vector was created to correspond to the training data, with each of the three output neurons corresponding to each of the grasp types to be classified for. In order to account for the Neural Network to potentially train poorly, the training and accuracy calculations were repeated 5 times, with the greatest accuracy being selected to represent a well trained network.

The output from the NN algorithm was then compared to the desired output. The number of times the algorithm correctly or incorrectly classified the grasps within a data set was recorded. The classification accuracy was calculated as the proportion of correct classifications out of all possible classifications in all data sets.

This entire process was repeated for each subject, with and without the matched filter preprocessing step, and for one up to ten electrode sites.

3.6.2. Results and Discussion

The results for the neural network accuracies are shown in Figure 31. This shows very clearly that the data that has been matched filtered does not outperform the data that was not. In fact, the reverse is true – the data without preprocessing performs better than the data with the preprocessing. While this means that the system as it currently stands is not suitable for use in prosthetic control, future work will aim to incorporate an initial action potential detection scheme as the preliminary step to a more sophisticated preprocessing mechanism.

Given that previous attempts to preprocess signals using wavelet approaches [48-51] have successfully increased classification accuracies, the algorithm presented here is clearly not appropriate for use in its current form. It is hypothesised that important information has been discarded by assuming that a single action potential shape can be representative of the entire EMG activity.

\(^1\) Levene-Marquardt is a numerical algorithm that provides a solution to minimization problems. For neural networks, it can be used to update the weight and bias values.
Suggested future work would be to allow for different “shapes” of action potentials – most likely the contribution from different motor units. Using a single action potential as a template has been assumed to be insufficient, however as more shapes of action potentials are allowed for, this could potentially be computationally expensive. Nevertheless, this could merit further investigation, and could then possibly be used to provide a more appropriate template, or series of templates for use in a matched filter.

Figure 31: Results for accuracy of classifier on data that has been preprocessed (red) and has not been preprocessed (black)

3.7 Conclusions

As a first step to processing EMG signals, an action potential extraction algorithm has been developed. Initial application of this to EMG data showed a good approximation to an action potential could be extracted. An additional algorithm was developed using matched filtering that would detect the presence of a template signal within an unknown signal.

These algorithms were then combined and tested on a simulated data set consisting of multiple overlapping modelled action potentials. This test simulation showed that the
algorithm was accurately able to recreate the original action potential used to create the sample signal, along with accurately picking out the location and amplitudes of the original signals. Processing time was found to be less than 5% of the time of a signal segment. This extraction algorithm was then applied to real data from 10 subjects, showing improvements in SNR from, on average, 8.3 dB to 10.5 dB.

Finally, data from between one and ten symmetrically placed sites from ten subjects were preprocessed using this action potential extraction and matched filtered. These data, in addition to the same data without the matched filtering step, were passed through a classification system in order to determine if the preprocessing was able to improve accuracy. The results from this showed that the data without the preprocessing outperformed the data with the matched filtering step.

As will be seen in Chapter 5, a method has also been developed that does allow for an increase in accuracy without a computationally expensive algorithm. As a consequence, while still useful for detecting the onset of EMG firings, this preprocessing technique will not be further used. Finally, this work has shown that the use of a single Action Potential as a template for matched filtering is not sufficient for use in electromyographic prosthetic control. While the increase in SNR demonstrated in Section 3.6 indicates that it merits further exploration, it is recommended that the AP extraction algorithm described here be used as a first step for a more complicated de-noising method. The most promising method found in the literature in terms of processing time, accuracy and amount of data channels required was presented by Florestal et al. [103], and is hence suggested as the most likely to be a suitable candidate.
Optimization of electrode placement for electromyographic prosthetic control

“How to select the best information? How do we even begin to define best?”

OEPEPC 10/03/2011

Summary

It is hypothesised that there are regions on the surface of the forearm that will provide better signals to control prosthetic devices than other regions. In order to test this hypothesis, a pilot study was conducted using data from a single subject (the author). The pilot study data were taken for two grasp types: pinch and cylindrical, and used two methods to give an overall idea of the location of activity on the subject’s forearm: feature ranking optimization and mean-difference ranked optimization.

Feature ranking optimization involved extracting two features (see Section 2.5 for details on feature extraction) from the data sets, and ranking each data site by the mean value of the feature across all time. As this method showed that there were several prominent regions on the forearm when each grasp type was performed, this method was developed into the mean-difference ranking method.

Mean-difference ranking optimization involved first interpolating the data onto a regular grid, followed by calculation of derivatives to simulate bipolar recordings, and extraction of features. The mean values for each grasp type were calculated across time, and the mean values for the two grasp types were subtracted in order to determine if there were areas of the forearm that were active for one grasp type and not the other, or vice versa.
The final approach to determining optimal electrode sites involved maximising an inter-grasp type Mahalanobis distance measure. This method continued to use data interpolated onto a regular grid, with derivatives calculated and features extracted. While more sophisticated classification methods exist than a Mahalanobis method, a slightly modified Mahalanobis measure is used in this Chapter that requires a single calculation in order to obtain a measure of performance.

It is hypothesised that this measure can be used to predict the performance of a more sophisticated classifier, such as a neural network or linear discriminant analysis. Testing this hypothesis using a neural network showed a reasonable relationship between the Mahalanobis measure and the accuracy of the network, while taking in the order of 1x10^{-5} the amount of time. It was hence used as the objective function in an optimization algorithm.

### 4.1 Introduction

In order to usefully control a multi-degree of freedom prosthetic device, it is necessary to provide adequate control signals. Previous studies have shown the ability to detect grasp types [68, 69, 85, 86] and finger movements [84] using electromyography; however the approach to electrode placement in these studies was to target specific muscle groups based on the underlying anatomy. As discussed in Chapter 2 this method is not ideal as the electromyograph recorded on the skin surface consists of signals from multiple muscles with various time delays and spatial filtering through the skin and connective tissue. Because of this, the optimal location to record from the targeted muscle may not be directly above that muscle.

Other studies have also used symmetrically placed arrays of sites [81-83], and attempted to optimise locations using a PCA [29] or SFS algorithms [44, 46]. The investigation into optimising sites showed that some improvement was possible over the symmetrically placed sites, however the SFS method would easily converge to a local maximum.

In this Chapter a method is presented to determine the optimal sites from which to record electromyographs, in order to discriminate between three basic grasp types; pinch,
cylindrical and spherical. Unipolar data were recorded from 128 sites on the subject’s forearm via a custom made silicone arm band (Figure 13), as described in Section 3.2. The location of the recording sites was measured in x-y-z space using a Faro Laser Scan Arm. The two initial methods discriminated only between pinch and cylindrical grasps, while the final method (see Section 4.5) attempted to discriminate between all three grasps, and could easily be modified to account for additional hand movements.

The recordings were first processed via mains (50 Hz) bandstop filtering. Spatial derivatives were then calculated in order to simulate bipolar recordings (using the Matlab function gradient). Bipolar recordings are preferable as preamplified signals have reduced noise (by performing amplification at the data collection site, the signal is less affected by noise introduced post measurement).

For the initial (feature ranking optimization) method, features were not considered between grasp types – the aim was simply to determine if there were regions for each grasp type that showed large amounts of electrical activity. Initial results from this method showed promise, and it was therefore decided to continue and determine if there were areas that would be useful in discriminating between grasp types.

The mean-ranking optimization method allowed for comparisons to be made between grasp types by subtracting the mean of one data set from another. Again this method showed promise, and it was hence decided to investigate a method whereby the affect of multiple features could be incorporated simultaneously, along with different grasp types.

The geometry is approximately cylindrical so to more easily represent the data, the Cartesian x,y,z coordinates were converted to θ,r,z cylindrical coordinates. The variable for θ is then multiplied by the average r value in order to bring θ and z into the same units. Finally θ and z were taken to represent a 2 dimensional Cartesian coordinate system on the skin surface (Figure 32). In this way the electrode sites were “unrolled” onto a 2D plane with θ and z being the new coordinate system.
The third method used a Mahalanobis distance measure. To test the theory of Mahalanobis distance being able to act as a predictor for neural network accuracy, data were taken from 128 sites on the forearms of 10 different subjects, while three different grasp types were performed. At this point an additional feature and an additional grasp type were used in order to ensure the system would be robust for future developments. Random selections of sites were chosen in order to give varying potential performances of a classification system. These then had a measure of accuracy calculated using both a neural network system and a Mahalanobis measure, and a linear fit was made to the two sets of data. This showed that while not perfect, the Mahalanobis measure was a reasonable predictor of neural network performance, while taking several orders of magnitude less time to calculate than the true performance of the neural network. For this reason it was selected as the performance metric to be used in the final algorithm.

This final algorithm used a data channel replacement method – an initial selection of channels was chosen, and the Mahalanobis distance measure calculated. As the interpolation scheme provided a regular grid of data, derivatives used to represent bipolar electrodes could be calculated in both the longitudinal and circumferential directions. This could then be converted into a magnitude and direction. Data for each grasp type would have individual directions – however using a single bipolar electrode setup, data can only be recorded in a single direction. As the recordings for each grasp type will be different depending on this direction – hence the direction that derivatives were calculated...
was also considered. In order to reduce the computational power required to consider directions, the total number considered was limited to three.

The remaining channels were then presented one at a time, and allowed to “replace” each of the current channels. If any of the combinations of channels were found to have a superior distance measure, they were then considered to be the new “optimal” combination of channels. This process was repeated until each channel had been presented.

The algorithm was run on a smaller data set in order to be able to compare it to an exhaustive algorithm that checked every possible combination of sites. Results from this test when run on the data from each of 10 subjects demonstrated that it was able to select a combination of channels that was within 0.1% of the optimal result, and to quite often pick the true optimal solution. It was able to do so in 3 seconds, compared to approximately 36 hours taken by the exhaustive check.

4.2 Feature extraction

As most pattern recognition systems extract features from EMG data, several methods reported in the literature were selected to do this. For the purpose of the first two methods, two feature sets were extracted. These features are integral of absolute value (IAV) and differential absolute value (DAV), as defined below.

To extract the features, non-overlapping windows of 60 data points were used. These were taken from simulated bipolar signals rather than the raw unipolar data - i.e. gradients of the data were taken in "longitudinal" and "circumferential" directions, and features were extracted from this gradient data. 60 data points were used as this gave a time window of 12ms (60 x 0.2ms) – combined with a majority voting method (as explained in Section 2.1.1) of 10 data points, this will give a total of 120ms (12 x 10) delay, within the acceptable time delay specified in [43]. The 60 data points were then passed through the following algorithms:
IAV – The absolute value of each point within the window is calculated, followed by
calculation of the integral of these absolute values within the 60 points. This was based on
the mean absolute value used by Hudgins et al. [42].

DAV – The maximum absolute value data point within the 60 sample window. This
feature is then low pass filtered (using 1 data point for each window). The filter was a 3rd
order digital butterworth low pass filter with a cut off frequency of 0.1 times half the
sampling rate - in this case the sampling rate is 5000/60 = 83.3 (as the filter is operating
on 1 data point in 60), so the cut off frequency is 4.17 Hz. Finally a numerical derivative
with respect to time of the low pass filtered data is calculated. This was based on the
work of Vuskovic and Du [67, 68] – their approach was to low pass filter rectified data –
this was extended by calculating a temporal derivative and substituting the rectified value
for a maximum value within the window.

4.3 Feature ranking optimization

4.3.1. Methods

The initial approach to optimization involved calculating derivatives either in the
proximal-distal (longitudinal) direction, or around the arm in the anterior-
posterior/lateral-medial (circumferential) direction [139], then ordering the data sets
based on their IAV (feature 1) and DAV (feature 2) features. The data sets with the
highest mean values for each feature were taken as being the optimal sites. This initial
study investigated the difference between cylindrical grasp and resting, along with pinch
grasp and resting. Data were taken from the forearm of a single subject. At this stage, no
comparisons were made between grasp types. Approximately 15 seconds of data were
recorded for each of the Cylindrical and Pinch grasp types.

Any channels where data could not be recorded (due most likely to poor connections to
the skin) were discarded for each grasp. An algorithm was created to map the remaining
sites to their nearest neighbours within ± 5° – generally corresponding to between 1mm
and 5mm, and derivatives were calculated using a 2-step numerical approximation. This
could potentially lead to the nearest neighbour being very far away from the original
point. In this feature ranking optimization, longitudinal derivatives were considered separately from radial derivatives, along with IAV being considered independently from DAV.

The process can be summarised as follows:

- Poor data channels were compensated for by finding “neighbours”
- Bipolar recordings were simulated by taking the gradient between two points
- IAV and DAV features were extracted, and the mean of these values calculated for each data channel
- Each feature was sorted, and those channels with the largest mean feature were taken to be optimal.

4.3.2. Results

Positions of the channels where useable data were recorded can be seen in Figure 33. This shows that a large amount of data are available for nearly all positions on the arm, with the exception of an area at approximately (0.5,80) that lacks information for the cylindrical grasp.

The results of the feature ranking optimization are shown in Figure 34 through Figure 37, and are shown on the subject’s forearm in Figure 38. These initial results highlighted several regions of interest. The first region at (-3,30) is a useful site for both grasp types with derivatives taken in either direction. There is also a region at approximately (-0.5,190) that is useful for both grasp types and both derivative directions. There are also several regions in the positive θ Section that are useful for one grasp type, but not the other. These can provide useful information for discriminating between grasps.
Figure 33: $0, z$ coordinates of the channels with data available for both grasps, and those channels where data are only available for one grasp.

Figure 34: Optimal channels positions found using radial derivatives for discriminating between pinch grasp and relaxing.
Figure 35: Optimal channel positions found using longitudinal derivatives for discriminating between pinch grasp and relaxing.

Figure 36: Optimal channel positions found using radial derivatives for discriminating between cylinder grasp and relaxing.
Figure 37: Optimal channel positions found using longitudinal derivatives for discriminating between cylinder grasp and relaxing.
Figure 38: Location of useful sites found using the feature ranking optimization method, and the region lacking in data for the cylindrical grasp

4.3.3. Discussion

The region at (-3,30) corresponds to a position in the extensor compartment of the forearm – shown in Figure 38 [137]. The extensors are an important part of forming a grasp, as it is required that a user’s fingers extend (open) before grasping an object. As such, it is expected that extensor muscles would be useful in the classification of grasp types.

The region at approximately (-0.5,190) corresponds to the posterior-lateral Section of the wrist, where the extensor carpi ulnaris runs [137]. Again, the extensors play a large role in forming the pre-grasp state. The regions in the extensor compartment that discriminate between grasps are likely useful for the same reasons, however without comparisons between grasp types, no further conclusions can be drawn.
4.4 Mean-difference ranked optimization

4.4.1. Methods

There were several issues with the feature ranking approach that needed to be accounted for. Most notably, there was no way to compare between different grasp types at those locations where data were unable to be collected – either due to a poor connection or the electrode not contacting the subject very well. These locations would vary from trial to trial, depending on which electrodes did not make contact. This mean-difference ranked approach attempted to take this into account [140]. At this stage, again, only data from a single subject was considered.

Due to the experimental setup, some of the channels did not make good contact to the skin surface. As a result, the data from these sites was not useful and contained primarily saturated outputs from mains pickup. These channels were removed from the active data set; however this left some disparity between different data recordings when different electrode sites were unable to contact the skin. In an attempt to compensate for this, an interpolation scheme was devised to translate the data onto a regular grid with a similar number of electrode sites as the original recordings.

It was found that performing the interpolation directly on the $\theta$-$z$ data caused variations in $\theta$ to have little effect on the interpolated data. To compensate for this, the $\theta$ data was converted from radians into millimetres by multiplying with $r$. This meant that the by sets of data were in the same scale, and variations in either dimension would have a similar effect on the final interpolated data.

This interpolation scheme would be susceptible to edge effects, i.e. the interpolated data at the edge of the interpolated region may be slightly inaccurate due to a lack of data at the edge. As the data are periodic in $\theta$, the interpolation scheme compensated for this in $\theta$ by duplicating the data twice with $\theta=\theta \pm 2\pi$ (or $\Theta \pm 2\pi r$ once converted to angular displacement). This would negate any edge effects at the boundary where the arm surface was unrolled.
This also allowed comparisons to be made directly between grasp types, regardless of channels with unusable data. Spatial gradients were again calculated using a 2-step numerical approximation (i.e. the derivative at a point between $x_n$ and $x_{n+1}$ is calculated as $\frac{f(x_{n+1})-f(x_n)}{x_{n+1}-x_n}$). The mean and standard deviation for each gradient point in the interpolation grid was calculated, for IAV and DAV features. These were then used to calculate a difference in means between the two different grasp types, and a total standard deviation. The optimal sites were taken to be those with the highest difference in means.

### 4.4.2. Results

Figure 39 plots the “useful” locations superimposed onto the locations of the electrodes. In this figure, the size of the markers represents the relative size of the difference in means. This figure shows that there are regions around the wrist that are useful in discriminating between grasps, along with several regions towards the elbow that are useful, particularly in the extensor compartment.

![Figure 39: Qualitative view of the difference in means for each of IAV and DAV, both circumferential and longitudinal. The relative sizes of the markers indicate the strength of the difference in means. The region at approximately (-0.5,80) shows a large marker for longitudinal IAV, indicating that this region has a large difference in means (between the two grasp types) using the longitudinal IAV feature. Conversely, areas such as (1,150) do not have large markers at all, and therefore had very little difference in means between the two grasp types.](image-url)
4.4.3. Discussion

As with the feature ranking optimization approach (described in Section 4.3), there are important regions in the extensor compartment that are most likely responsible for forming the pre-grasp. The regions around the wrist correspond to the abductor pollicis longus and extensor pollicis brevis, among others. These are two of the muscles located in the forearm that control the thumb. As the thumb is the major component in grasping, it is to be expected that electrodes located in this region would be useful in classification.

4.5 Mahalanobis distance maximization optimization

4.5.1. Methods

The mean-difference ranked approach was still limited in that it considered gradients in only longitudinal and radial directions, with each derivative direction being considered separately, and considered each feature independently rather than a combination of the features being used in classification. It also did not allow for covariance between data sets, and only considered two different grasp types. This final approach took derivative direction into account by calculating magnitude and direction of derivatives, and projecting these derivatives onto the directions that each grasp type was maximum in. The justification for this is contained in Section 4.5.1.1. Features were taken into account simultaneously through use of a Mahalanobis classifier (see Section 1.3.5 for definition) that accepted multiple feature inputs. This also allowed for changes in covariance to be considered along with optimization for multiple grasp types.

Data were collected for 10 subjects. The original two feature extraction methods were used, in addition to Zero crossings (ZC) (again as used by Hudgins et al. [42]) i.e. – The number of times the data crosses the x-axis within the window. This was based on its use in the study by Hudgins et al. [42]. Data were also recorded for three grasp types – pinch, cylindrical and spherical.
Consideration of direction of derivatives

The data collected using UnEmap has allowed for the creation of an electrical field map over the forearm. Each grasp type will have its own unique field map, with a different electrical field at each point. However, a real-time system will only be able to record data in one direction at each point, hence a method is required for projecting this 2 dimensional field onto a single direction.

The aim of this Section is to find a single direction at each electrode site that will enable good discrimination between each grasp type, and also allow good discrimination of between each grasp type and background noise. In this way, the number of bipolar electrode pairs needed for each site will be only one – oriented in the direction given by this Section.

Given a vector for the derivative in polar co-ordinates (i.e. the magnitude and direction of the derivative), it is possible to calculate the derivative in any arbitrary direction. If the effect of looking at \( m \) directions for each of \( n \) channels is to be investigated, there will be \( m^n \) possible combinations for each set of \( n \) channels. Hence, to be able to process the data in a timely manner, the number of directions the derivative is taken in must be constrained, with the exception of the case where \( m \) and \( n \) are very small.

In order to constrain the directions data are calculated in, the length of the data in the direction of each of the maximum derivatives was given by the scalar projection:

\[
b \cdot \hat{a} = |b| \cos \theta
\]  

(11)

Where \( \mathbf{b} \) is the vector being projected, \(|b|\) is the magnitude of the vector \( \mathbf{b} \), \( \hat{a} \) is a unit vector representing the direction being projected onto, and \( \theta \) is the angle between the two directions. This follows directly from the trigonometric formula \( \cos \theta = \frac{\text{opposite}}{\text{adjacent}} \), where \text{opposite} represents the vector projection, and \text{adjacent} represents the original vector.
Chapter 4 Optimization of electrode placement for electromyographic prosthetic control

The reasons for taking data in these directions is as follows:

Ideally, the data should be maximally represented – i.e. recordings should be taken in the direction of maximum gradient. However:

- Only a single direction can be measured
- Therefore only one grasp can have data measured in the direction of maximum gradient

Now consider a hypothetical example where recordings are taken in a regularly spaced grid (shown in Figure 40). If recordings are taken for two different grasp types, derivatives can be calculated in the x and y directions (i.e. the direction of the grid of electrodes), and from this, a magnitude and direction for each grasp type can be calculated (shown in Figure 41). These resultant vectors for each grasp type are shown in Figure 42.

![Figure 40: A grid of 9 electrodes, represented by asterisks](image)
Section 4.5 Mahalanobis distance maximization optimization

Figure 41: Hypothetical recordings for two grasp types. Grasp type 1 in (a) shown in green has x,y recordings of (3,3), with the total vector shown in the solid green line. Grasp type 2 in (a) shown in blue has x,y recordings of (4,-1), with the total vector shown.

Now, consider taking recordings in a direction inside of the angle formed by the two directions found by calculating x,y derivatives for each grasp type, and converting to polar coordinates. This corresponds to a vector with \( \theta_1 < \phi < \theta_2 \). For two vectors, this is illustrated in Figure 43.
In this scenario, both vectors magnitudes will slightly decrease, and become more similar to each other. This will result in two vectors that are easy to distinguish from background noise (as their relative magnitude is still quite high), but are difficult to distinguish from each other.

Now, consider taking recordings in a direction outside the angle formed by the two directions found. This corresponds to a vector with $\theta_1 > \phi < \theta_2$. For two vectors, this is illustrated in Figure 43 (b)

In this scenario, the blue vector has slightly decreased in magnitude, while the green vector has greatly decreased in magnitude. This results in two vectors that are easy to distinguish from each other, however one vector that is difficult to distinguish from background noise.
Finally, consider taking recordings in either of the directions formed by the three directions found. For two vectors, this is illustrated in Figure 44. In this scenario, the magnitude of the green vector is different enough from the magnitude of the blue vector to make it easy to discriminate between them, while not reducing the magnitude of either vector to the point where they would be difficult to distinguish from background noise.

The same argument can also be applied to directions of additional grasp types – in this case three in total – pinch, cylindrical and spherical. While it would be preferable to factor in optimization of direction, doing so would increase the size of the problem by an order of magnitude. By making some reasonable assumptions (as outlined above), it is possible to reduce the number of directions to be optimized to three, i.e. the direction each grasp was maximum in. If every possible combination of directions is to be calculated, the total number of calculations required (for each combination of channels) would be:

\[ \text{NumberDirections}^\text{NumberChannels} \]

For example, when using 3 directions and 5 channels, the number of calculations required is \( 3^5 = 243 \), while if 30 directions were to be used, the number of calculations required would be \( 30^5 = 24,300,000 \).
4.5.1.2. Cluster separability

The ideal case after feature extraction in order to make a more accurate classifier would be to have high separability between clusters corresponding to different grasp types. Figure 45 shows four hypothetical grasp types (represented by blue, red, black and cyan) in two dimensions, representing a single feature extracted for 2 electrode sites. From this it is straightforward to differentiate points that belong to the red cluster or blue cluster, indicating that they have good separability. Comparing this to the cyan and black clusters however, it can be seen that these overlap and are difficult to differentiate. Ideally, data sets similar to red and blue are desired rather than the cyan and black sets.
4.5.1.3. Distance measure classifications

Cluster analysis techniques have involved a distance measure calculation – most commonly a Euclidean distance, however recognition of co-dependence between channels of data lead into a Mahalanobis distance representation. Training data sets are applied to the distance classifier which then assigns each data point to a hyperspace cluster. Any new “unknown” data points then have their distance measure calculated and are assigned to the cluster that they are closest to by the chosen performance metric [43, 68, 69].

Euclidean distance

The Euclidean distance of two points in hyperspace is taken as the length of the line segment connecting the two points. As such, the Euclidean distance \( (E_d) \) between two points \( (P_1 \text{ and } P_2) \) is given by the following formula:
Chapter 4 Optimization of electrode placement for electromyographic prosthetic control

\[ E_d = \sqrt{\sum_{i=1}^{n}(P_i - \bar{P}_{2i})^2} \]

while the Euclidean distance between two clusters (C₁ and C₂) with means in dimension \( i \) of \( \overline{C_{1i}} \) and \( \overline{C_{2i}} \), is given by the following formula:

\[ E_d = \sqrt{\sum_{i=1}^{n}(\overline{C_{1i}} - \overline{C_{2i}})^2} \]  

which is simply the formula from equation (12) with the points P₁ and P₂ replaced by the mean of the clusters C₁ and C₂

**Covariance and the covariance matrix**

While the covariance matrix and Mahalanobis distance have been introduced in Chapter 1, for the sake of clarity a more in depth description is provided here. Further explanation can be found in [141]. The variance of a data set is a measure of the spread of the data from its mean. It will be seen shortly that the variance of a data set is in fact the covariance of the data set with respect to itself. The covariance between two data sets \( x \) and \( y \) is given as:

\[ Cov(x, y) = \sum_{i=1}^{n}[(x_i - \mu_x)(y_i - \mu_y)] \]  

Where \( n \) is the number of data points and the means of the data sets \( x \) and \( y \) are \( \mu_x \) and \( \mu_y \) respectively. Setting \( y=x \) gives:

\[ Cov(x, x) = \sum_{i=1}^{n}[(x_i - \mu_x)(x_i - \mu_x)] = \sum_{i=1}^{n}(x_i - \mu_x)^2 \]  

This is the definition of variance i.e. the covariance of a data set with respect to itself is its variance.

If we now assume that \( X \) is a vector \([x_1,x_2,x_3...x_n]\) where each \( x_i \) is one random variable, then the covariance matrix, \( S \), is defined as follows:
Section 4.5 Mahalanobis distance maximization optimization

\[ S(i, j) = \text{Cov}(x_i, x_j) = \sum_{i=1}^{n} \left[ (x_i - \mu_j)(x_j - \mu_i) \right] \]  

(16)

It can be seen then that the covariance matrix is simply a matrix whose \( i,j^{th} \) is the covariance between variable \( x_i \) and \( x_j \).

**Mahalanobis distance**

The Mahalanobis distance normalizes to take into account the covariance of the data set – that is, it no longer assumes each dimension is independent of the other. As will be seen shortly, if each dimension is independent, i.e. the covariance is zero, the Mahalanobis distance becomes equal to the Euclidean distance. Because the Mahalanobis distance makes use of the covariance, it requires statistical information i.e. at least one of the data sets must contain more than one data point. The Mahalanobis distance \( D_m \) between a data point \( P \) and a cluster of data \( C \) is given by the following formula:

\[ D_m = \sqrt{(p - \mu)^T S^{-1} (p - \mu)} \]  

(17)

Where \( p \) is a vector containing the coordinates of the point \( P \), \( \mu \) is a vector containing the means of the data cluster \( C \) and \( S \) is the covariance matrix of the cluster \( C \). Note that if there is no covariance between dimensions, \( S \) becomes a diagonal matrix and equation (17) becomes:

\[ D_m = \sqrt{\frac{(p - \mu)^T (p - \mu)}{\sigma^2}} = \sqrt{\sum_{i=1}^{n} \frac{(p_i - \mu_i)^2}{\sigma_i^2}} \]  

(18)

Where \( \sigma \) is the standard deviation of the data set.

Mahalanobis classifiers have been used successfully in previous studies [69, 85, 86], with final results giving success rates approximately 75-80%. Figure 46 shows a two dimensional example where taking covariance into account can be useful.
Figure 46: Choosing which of the two green points is closer to the blue data cloud. In (a), an ellipse has been fitted where the data are assumed not to vary off axis at all. In this case it is difficult to algorithmically determine which green point is the better fit. In (b) however, an off axis ellipse has been fitted, showing that it is much easier to determine which green point is the better fit.

4.5.1.4. Mahalanobis distance as a predictor of neural network performance

It was hypothesized that a Mahalanobis distance measure could act as an indicator of a classification system's accuracy. Generally each classification performed using a neural network will be relatively quick, however each calculation of accuracy would require training the network – which can be quite time consuming if done multiple times.

In order to test this hypothesis, 50 calculations of neural network accuracy were performed, along with 50 calculations of Mahalanobis distance, using an Intel Core 2 6600 PC. It was found that the 50 Neural network calculations took on average 39.0 seconds each, with a standard deviation of 13.82 seconds and the 50 Mahalanobis distance calculations took 7.4 milliseconds, with a standard deviation of 3.45 milliseconds. Mean and standard deviations for each subject are available in appendix 4. This indicates that a Mahalanobis distance based algorithm could be up to 5,000 times quicker than a neural network based algorithm.

As an inter-cluster Mahalanobis distance can be obtained with a single calculation, this could provide a much faster means to test the "accuracy" of a system, rather than training a classifier and testing each data point in turn. In order to test if this was feasible, the data from each subject was tested as follows:
• A random selection of five data channels was chosen, 50 times for each subject. Five data channels were taken as this represented the “asymptote” found in previous studies where additional input channels provided little additional improvement in accuracy [44, 46].

• The data from these channels was used as the input for a neural network classifier.

• The neural network was trained on the first data set for each subject.

• The accuracy of the Neural network classifier was calculated using the second data set for each subject as unknown data to be classified, and taking the total number of correct calculations as a percent of the total number of calculations to be the accuracy.

• A Mahalanobis distance measure was then calculated for this set of data. This was calculated as the cube root of the product of the Mahalanobis distances between each of the three clusters that data had been collected for.

• Data points that had a NaN or complex Mahalanobis distance were discarded as these were likely to correspond to a poorly conditioned covariance matrix.

The neural networks were created and trained in the same manner as in Section 3.6

4.5.1.5. Optimization algorithm

To implement a Mahalanobis distance maximization routine, an algorithm was developed wherein the spatial derivatives for each point were calculated using the Matlab gradient function. The magnitude of the derivatives was then used in determining optimal electrode pair sites, while the direction of the maximum derivative was used in calculating the orientation of the electrode pairs. The optimal electrode sites were taken as those that would create the greatest Mahalanobis distance between the data clusters.

In its simplest form, the problem can be expressed as:

\[ \text{1 The product was taken in order to account for particularly small inter-cluster Mahalanobis distances, as this would result in a similarly small product. The cube root was taken in order to preserve units.} \]
Given $m$ sets of data for two groups, determine which $n$ sets of data perform the best according to an objective function $f(x(n))$.

In this case, $m$ corresponds to the number of grid points in the interpolated data, $n$ corresponds to the number of electrode sites to be optimised for, and $f(x(n))$ — the objective function — corresponds to maximisation of the Mahalanobis distance $D_m(n)$. This is calculated based on the $3n$ dimensional space - as three features are being used for each channel.

Recall from section 4.5.1.3 that a Mahalanobis distance between a cluster of data and a point is defined as:

$$D_m = \sqrt{(p - \mu)^T S^{-1} (p - \mu)}$$ (19)

Where $p$ is a vector containing the coordinates of the point $P$, $\mu$ is a vector containing the means of the data cluster $C$ and $S$ is the covariance matrix of the cluster $C$. For our case, we have two clusters that we want to calculate a distance between. To compensate for this, $p$ is assumed to be the mean of the data points in cluster 1, and $\mu$ to be the mean of the data points in cluster 2. If we are optimising for $n = 5$ for example, and then $p$ and $\mu$ will be 15 point vectors - with each point corresponding to the mean with respect to time for 1 feature for 1 channel. Finally the covariance matrix can be modified to take into account both clusters according to:

$$S = \frac{n_1 S_1 + n_2 S_2}{n_1 + n_2 - 2}$$ (20)

Where $n_i$ corresponds to the size of data set $i$ and $S_i$ corresponds to the covariance matrix of data set $i$.

The initial algorithm devised to solve this issue without exhaustively checking every possible combination is described in Figure 47.
Figure 47: Flow chart of the initial data set reduction algorithm. SNR is calculated based on the ratio of amplitude during active periods to inactive periods. While this required an initial manual tagging of these regions, it was assumed that each data set from an interpolated grid would have active and inactive periods at approximately the same time. $m$ is the starting number of channels, and $n$ is the final number of channels.
It should be noted that this algorithm has the potential to reach a “local” maximum where no other channel added to it would produce an increased Mahalanobis distance, though it may not be the true optimal solution due to the co-dependence taken into account by Mahalanobis measures.

This algorithm also takes into account the direction that the spatial derivative is projected onto (see section 4.5.1.1) by calculating the spatial derivatives for each channel in each of the three directions given by the three grasps. Again, this does not consider all possible directions for derivatives, however it greatly reduces the size of the problem while still considering a reasonable amount of direction information. As the goal of this thesis was to investigate the location of the electrodes rather than the direction in which they should be placed, this was considered sufficient at this stage.

### 4.5.1.6. Algorithm investigations

The effect of the following was investigated:

- The interpolation grid density \((m)\);

  This investigation involved running the algorithm repeatedly, increasing the density of the interpolation grid each time. The time taken and final Mahalanobis distance reached for each run was stored. The electrode grid density \(m\) is taken to mean here the number of points in each direction. For example, \(m\) of 10 would mean a 10x10 grid, giving a total of 100 spatial locations.

- The number of electrodes to optimize for \((n)\);

  This investigation involved running the algorithm repeatedly, with an increase in the number of channels to optimize for. With \(n = 1\) to 30. This was repeated 20 times in order to account for the variability of the algorithm.

- The number of data sets left by the culling algorithm;

  This investigation involved running the algorithm repeatedly, removing one data set at a time, with those with the lowest SNR as the input to the culling algorithm removed first. The time taken and final Mahalanobis distance
reached for each run was stored. The number of data sets could then be compared with the culling threshold required to leave only those data sets.

- Size of data set vs. computations required using this algorithm as opposed to an exhaustive algorithm;

This investigation involved calculating the number of objective function calls that were theoretically required to complete the developed algorithm as opposed to exhaustively checking every possible combination, as a function of both the number of input data sets and the desired number of output data sets.

- The accuracy of the developed algorithm;

This investigation involved using a smaller subset of data, and calculating the $D_m$ for every possible combination, then comparing this with the result given by the developed algorithm.

## 4.5.2. Results

### 4.5.2.1. Mahalanobis distance as a predictor of neural network performance

The results of the investigation into Mahalanobis distance as a predictor of neural network performance are shown in Figure 48. A linear fit has been made to these data, resulting in an $R^2$ value of 0.6452, indicating a reasonable fit to the data. While there are data points that have a high accuracy and low Mahalanobis distance, there are no data points that have a high Mahalanobis but a low accuracy.

Due to this relationship, and the time taken to train and use a neural network, Mahalanobis distance measures were used as an indicator of data suitability rather than the performance of a neural network. However, as there is some amount of variability between the linear model and actual data, final checks should be made using a neural network.
4.5.2.2. Algorithm investigations

Due to the possibility of the algorithm reaching a different result based on the starting combination of channels that was selected using the rand function, investigations for this Section were run 10 times to account for the variability of the algorithm.

The results of increasing the grid density can be seen in Figure 49. This figure shows that as the grid density increases, both the maximum Mahalanobis distance reached (corresponding to the optimal channels) and time taken increase. However, there are fluctuations in the Mahalanobis distance, implying that increasing the grid density does not always lead to an increase in performance.
Figure 49: Final Mahalanobis distance reached and time taken for increasing grid density, ± 1 standard error.

The results of increasing the number of electrodes to optimize for are shown in Figure 50. As expected, a higher Mahalanobis distance can be reached by using an increased number of electrodes; however there are several interesting features:

1. Beyond 6 electrode sites the standard error increases dramatically,
2. There is a slight oscillation as \( n \) increases,
3. The Mahalanobis distance reached dramatically drops off for \( n > 14 \).

---

1 Where standard error is defined as standard deviation divided by the square root of the sample size.
Figure 50: Final Mahalanobis distance reached for increasing the number of electrodes to optimize for, ± 1 standard error. Note the increasing variability for n>6 and the decrease in score for n>14. In studies in the literature, these numbers of electrodes are not found, so this result does not infringe on the use of the methods presented for a real-time system.

Figure 51 shows the affect of adding more channels with lower SNR to the algorithm. In order to reduce the computational time required, this investigation was performed on a 10x10 interpolation grid. It can be seen from this figure that increasing the number of data sets, in general, increases linearly with the time taken to reach a solution as is expected from equation (22). This figure also clearly shows an increase in the Mahalanobis distance as the number of input channels is increased (with those channels with highest SNR being added first), until a plateau at approximately 21 channels. This will be due to nearly all of the “optimal” channels being found within the top 21 channels when ranked by SNR.
Figure 51: The effect of the initial "culling" step on the performance of the algorithm.

The corresponding SNR of the data sets left out is shown in Figure 52. This shows an approximately linear relationship between number of data sets and SNR, and that to reduce the number of channels by half while keeping those channels with the highest SNR requires the threshold to be set at 35% of the maximum SNR.
Figure 52: Signal-to-noise ratio of the data set that would be left out in order to reduce the number of data sets to a given number. SNR measurements have been given as a % of maximum SNR in order to non-dimensionalise. Notice that to be left with approximately 40 data sets, a culling threshold of approximately 35% maximum SNR would be required.

Figure 53 shows the results of the investigation into the number of Mahalanobis distance calculations required for the optimization algorithm as opposed to exhaustively checking every possible combination. The number of calculations required for each method is calculated as follows:

1. **Exhaustive algorithm:**

   The number of calculations is taken as the number of possible ways to choose $n$ data sets from $m$, often written as $\binom{m}{n}$ or $^mC_n$. The calculation is given as

   $\text{TotalCalculations} = \frac{m!}{n!(m-n)!}$  \hspace{1cm} (21)

2. **Optimization algorithm:**

   The number of calculations required consists of 1 for the initial combination of channels, and another $n$ calculations for every additional channel that is checked, resulting in $m \times n$ calculations, giving the total number of calculations as:
Section 4.5 Mahalanobis distance maximization optimization

\[
\text{TotalCalculations} = m \times n + 1
\]  

\[(22)\]

Note that for low numbers of \( m \) and \( n \) (i.e. \( m < 5 \) or \( n < 2 \)), exhaustively checking every combination will be more efficient than the algorithm developed. However, as \( m \) and \( n \) increase the cost for exhaustively checking greatly increases while in the algorithm presented here the cost does not increase nearly as much. For example, consider \( m = 30 \) and \( n = 5 \). In this case, an exhaustive method would require 142,506 calculations, as opposed to 151 for the algorithm developed.

An exhaustive algorithm (i.e. an algorithm that checks every possible combination of channels) was performed on the data from a 6x6 interpolation grid. The number of sites to be optimized for was chosen to be 4 in order to limit the total combinations of channels to 58,905, and hence reduce the amount of time taken to calculate this. This was repeated for data from each of 10 subjects. The results of this investigation for a single subject are shown in Figure 54, with the Mahalanobis distances ranked by value. This figure shows that the greatest Mahalanobis distance found is approximately 6.1, and that the majority of channel combinations have a Mahalanobis distance falling between 2 and 4.5. The exhaustive checking method took approximately 36 hours to complete on a Dell Precision PWS390. The optimization algorithm developed in this Chapter produced a Mahalanobis distance of 5.9669 (corresponding to the 8th best ranked solution out of
58,905 found with the exhaustive algorithm) after approximately 3 seconds, on the same machine. Comparing this exhaustive algorithm to the performance of the optimization algorithm, it can clearly be seen that the new algorithm allows a very close to optimal data set to be found in greatly reduced time.

![Mahalanobis distance graph](image)

**Figure 54**: All possible Mahalanobis distances based on a smaller data subset of 36 data sets, with the optimization algorithm’s performance marked by the red asterisk.

When run on the data for all 10 subjects, the Mahalanobis distance reached was ranked, on average was found to be the 8th best of 58,905 with a standard deviation of 9.61. For four of the subjects the algorithm found the best ranked site. This is a good indication that while the algorithm does not always select the truly optimal solution, it is able to come very close.
4.5.3. Discussion

4.5.3.1. Algorithm investigations

The positive relationship between Mahalanobis distance and grid density in Figure 49 is as expected. Increasing the grid density makes it more likely that there will be a grid point at or near the true “optimal” site. As data can only be recorded from a limited number of sites, it may not be recorded at the “optimal” site; however the interpolation scheme as described in Section 4.4 attempts to account for this. It should be noted, however, that increasing the number of interpolation points does not increase the amount of data available; it merely re-samples the data at different positions.

To illustrate this, consider the problem of finding one optimal site using either a 2x2 grid or a 3x3 grid, as seen in Figure 55. This shows how a smaller number of grid points may give a closer approximation to the true optimal site.

![Figure 55: A demonstration of how a smaller grid density can provide a better estimation of the optimal sites. The black grid represents a 3x3 grid, while the blue grid represents a 2x2 grid, and the red X represents where the true optimal site is. In this case, the 2x2 grid would give a better estimation of where the true optimal site is, as it intersects the best site.](image)

As the increasing grid density results in a quadratically increasing number of data points to process, the grid density was limited to a size of 30x30 for the grid density investigation. The maximum Mahalanobis distance was found at a grid density of 29x29, so this was used in all future investigations.

For small numbers of electrodes, the positive relationship between Mahalanobis distance and \( n \) is expected as increasing the available data would lead to an increase in accuracy (Figure 50). Put another way, the greater the number of electrodes available, the more
accurate the classification will be. However, the cost of collecting and processing data in real time becomes infeasible as \( n \) is increased due to data processing costs.

The increase in standard error as \( n \) increases would be expected as the number of possible combinations increases. This means that the likelihood of reaching a local maximum would increase, causing the algorithm to converge to a different result. If the number of channels to be optimized were to exceed 6, an alteration to the algorithm would be required in order to reduce this variability and provide a higher likelihood of achieving the true optimum. The most likely alteration would be to take 2 channels for each step to be passed into the algorithm. This would increase the number of combinations checked at each step, hence increasing the accuracy at the cost of increased computational time.

The oscillations as \( n \) increases are not expected, nor can they be explained currently. The drop off in Mahalanobis distance above \( n=14 \) is most likely due to issues with the estimation of the covariance matrix as the number of data sets increases. When the number of data sets is large with respect to the sample size, traditional estimates of the covariance matrix will tend to be very unstable [142]. To compensate for this, further work would need to either take more data points, or use a modified approach to covariance matrix estimation such as the shrinkage estimate presented in [142].

Previous optimization algorithms have assumed no co-dependence or covariance between channels of data. As the culling algorithm deals with a single channel of data at a time, it can not possibly consider covariance. Clearly, this can be assumed to an extent as is seen in Figure 51; however the decrease in Mahalanobis distance as the number of data channels is reduced beyond 40 shows that the covariance becomes very important at this point. As a result, the initial culling algorithm was performed with the threshold set to 35\% of the maximum SNR.

As expected, the exhaustive algorithm does reach a higher Mahalanobis distance than the optimization algorithm; however the resulting 8\textsuperscript{th} best possible result tested on a reduced grid size (of 6x6) indicates that the algorithm performs suitably well.
A method has been developed to optimize the electrode placement in electromyographic HCI control, that is able to substantially reduce calculation time when compared to a brute force approach. Initial approaches attempted to find those signals that had the greatest signal amplitude, and subsequently the greatest difference in signal amplitude between grasp types. The final method used an inter-cluster Mahalanobis distance maximisation approach to optimize for classification of three different cluster types. This required the development of an algorithm to determine which combination of data sets would perform best according to some objective function – in this case, maximising the cube root of the product of three Mahalanobis distances.

The performance of this algorithm was validated against an exhaustive method, on a smaller subset of 36 data sets. The algorithm on average converged to the 8th best result of 58,905, indicating it performs with 99.9% accuracy, while taking only 3 seconds. This shows that a very close to optimal solution can be achieved using this algorithm, while performing very economically on current computers.
5

Validation and testing

“And now I require the assistance of a volunteer from the audience”

OEPEPC 10/03/2011

Summary

In Chapter 4 electrode placement was optimized to better differentiate between three different grasps. The performance of the Mahalanobis distance maximisation algorithm was evaluated against an exhaustive algorithm using a reduced data subset. It would find a combination that was very close to optimal, although not necessarily the best combination of data sets.

While the accuracy of the algorithm can be measured by its ability to find the optimal sites based on Mahalanobis distance classification, the ability of the system to improve electromyographic grasp recognition should be measured by the improvement in overall accuracy of a classification system.

This Chapter aims to compare the accuracy of a classification system where the data are taken from the optimal sites found in Chapter 4 with the accuracy of a classification system where the data are taken from other sites. Four sets of data sites were used in order to demonstrate this:

- Optimal sites selected by the Mahalanobis distance maximisation algorithm
- Symmetrically placed sites
- Sites selected using sequential forward selection
- Sites selected using a PCA reduction algorithm
5.1 Methods

5.1.1. Selection of sites

The data used to calculate the accuracy of a classifier using different sites was the same as that used in Chapter 3. Recalling that for each data subject, and for each grasp type, two data sets were taken, the first data set for each grasp type was then used as training data, while the second set was used as unknown data to be classified. Symmetric sites were chosen as those sites that were 25% along the length of the forearm. The first channel was chosen to be approximately over the radius. The angle of each channel of data from the centre of the arm was then calculated based on the total number of channels, and the remaining channels selected as those that most closely matched the angle required.

Each of the optimization algorithms (Mahalanobis, SFS and PCA) was run on the data, with the number of sites to optimize for starting at 1, and increasing up to 10. This process was repeated for each of the 10 subjects. The SFS and PCA algorithms were implemented as follows:

5.1.1.1. SFS algorithm

1. Each site was tested in turn to determine which single site had the highest Mahalanobis distance product (as described in Chapter 4). This was chosen as the optimal site.

2. Each of the remaining sites was tested in turn, together with the single optimal site, in order to determine which of the remaining sites has the highest Mahalanobis distance product.

3. This process was then repeated until the desired number of channels had been reached.

5.1.1.2. PCA algorithm

1. A principal component analysis decomposition was performed on the data
2. The \( n \) most significant principal components were selected, where \( n \) is the number of channels to optimize for.

3. The similarity of each of the original channels to these principal components is calculated as follows:

\[
S(i) = \sum_{j=1}^{n} \lambda_j \frac{D_i}{P_j}
\]

Where \( S(i) \) is the similarity of the data \( D \) in channel \( i \) to the data projected onto the projected data \( P \) in principle direction \( j \), and \( \lambda \) is the eigenvalue corresponding to that principal direction.

5.1.2. Grasp classification

For all three channel selection protocols, the data from these selections was then run through a feedforward back propagation neural network (NN). The neural network was created using the Matlab \textit{newff} function, and consisted of 1 input layer with \( 3 \times n \) neurons (corresponding to each of the 3 features from the \( n \) input channels) – see Section 2.6.1 for an explanation of neural networks, 1 hidden layer with 12 neurons, and 1 output layer with 3 neurons using Levenberg-Marquardt\(^1\) training; mean squared error performance and random data division.

The first data set was used as training data, while the second data set was treated as unknown data to be classified. An associated output vector (with three values) was created to correspond to the training data, with each of the three output neurons corresponding to one of the grasp types to be classified for. In order to account for the Neural Network to potentially train poorly, the training and accuracy calculations were repeated 5 times, with the greatest accuracy being selected to represent a well trained network. A majority vote with 10 values was used to smooth the output data.

The output from the NN algorithm was then compared to the desired output. The number of times the algorithm correctly or incorrectly classified the grasps within a data set was recorded. The classification accuracy was calculated as the proportion of correct

\(^1\) Levene-Marquardt is a numerical algorithm that provides a solution to minimization problems. For neural networks, it can be used to update the weight and bias values.
classifications out of all possible classifications in all data sets. This was repeated for each of the three different sets of sites and the results from each set of sites were compared.

## 5.2 Results

The results for each selection of electrodes for all 10 subjects can be seen in Figure 56. This clearly shows that while the SFS algorithm slightly outperforms the symmetrically placed sites, and the PCA algorithm appears to offer no improvement, the algorithm developed in this thesis offers a considerable improvement not only over the symmetrically placed sites but over previous optimization algorithms also. Figure 57 shows the accuracies of the new algorithm, the SFS algorithm and the p-values of a paired t-test between the two.

![Figure 56: Accuracies using the algorithm from Chapter 4, symmetrically placed sites, and sites chosen using SFS and PCA algorithms, ± 1 standard deviation](image-url)
5.3 Discussion

The results in Figure 56 clearly show that for the data available, the optimization has outperformed those chosen by SFS, which have in turn (on average) outperformed the symmetrically or PCA chosen sites. The accuracy increases from approximately 85% using symmetrically chosen sites to 88% using SFS chosen sites and 95% for optimal sites when considering values in the “asymptotic” region, i.e. ≥4 electrodes.

This shows an accuracy of approximately 77% using 1 electrode for all selection methods, increasing as more channels are added. The standard deviation tends to decrease as more channels are added for the new algorithm, however the standard deviation of approximately 5 when only a single channel is used indicates that there is a reasonable amount of inter-subject variability.

Recall from Chapter 4 that the optimization algorithm had the potential to converge to different sets of sites, though each set of sites had a similar Mahalanobis distance. This
variation in classification accuracy indicates that the sets of sites with similar Mahalanobis distances may have very different accuracies when used in classification, as was noted in Section 4.5.2. Due to this potential to have a varying accuracy, the output sets of sites from Chapter 4 with high Mahalanobis distances were passed through a classification routine in order to determine which set of sites will perform best for classification. By having the 5 or 6 (approximately) combinations of channels that are most likely to give the best performance, the amount of time taken to calculate the accuracies using a neural network can be substantially reduced, while still obtaining close to optimal solutions.

The p-values for 1 and 2 optimized sites show that there is no evidence for a difference in means between sites chosen by the SFS algorithm and the new algorithm. For 3 and 6 sites there is evidence and strong evidence respectively, while for all other numbers of sites there is extremely strong evidence of a difference in means. This indicates that the sites chosen using the new algorithm can offer improved accuracy in myoelectric prosthetic control.

5.4 Conclusions

The optimal electrode sites have been validated using against symmetrically placed electrodes, and electrodes with locations chosen by SFS and PCA algorithms. By choosing the correct locations for data collection, classification accuracy was increased from 85-88% to approximately 95%. The off-line validation confirmed that collection of data at optimal sites can provide a considerable increase in classification accuracy using a NN classifier.
Conclusions

“Preprocessing, data reduction, electrode placement and validation. My work here is done.”

The goal of this research was to advance the state of the art in prosthetic hands, with a focus on their control and recognizing the user’s intention. Chapters 1 and 2 identified the mechanisms required to provide control over an advanced prosthetic device using electromyographic signals. This showed that there was room to improve classification accuracy via investigating the effect of electrode position on classification and the effect of preprocessing the EMG signals. This problem was split into three phases:

- Acquisition of a rich data set from the forearm. This required signal preprocessing to increase the SNR due to the large amount of noise introduced to the signal during the signal acquisition process.
- Development of an algorithm to re-sample unevenly distributed data onto a regular grid, and select the optimal data sets from this grid, and application of the optimization algorithm to the preprocessed data.
- Validation of the chosen sites by investigation of the performance of those sites in comparison to anatomically chosen sites, along with an investigation into the effectiveness of two different classifiers using these optimized sites.

Chapter 3 described the collection of a rich EMG data set from the forearm and then introduced an iterative action potential detection algorithm that produced a representative action potential as observed on the surface of the skin. While the detected action potential
exhibited many of the characteristics expected from a “text-book” action potential, it
differed in that its final resting value was greater than its starting value. This
representative action potential was then used as the input for a matched filtering method;
results of application of this method to ten data sets showed increases in SNR of 2.2dB on
average, however did not lead to an improvement in accuracy when applied as a
preprocessing step in classifying EMG data.

An algorithm to reduce data sets was presented in Chapter 4. This algorithm used an
interpolation scheme to resample irregular data onto a regular grid. Following
interpolation, data sets with the lowest SNR were excluded. The algorithm used a
heuristic approach to increasing the Mahalanobis distance between data clusters by
processing different data sets.

An investigation into the grid density on the final Mahalanobis distance revealed that
increasing the density, in general, led to an increase in maximum distance between
clusters, and noted that fluctuations can occur due to the proximity of the re-sampled
points to the actual optimal positions. Due to the physical constraints imposed by the
hardware available, interpolation was limited to a 29x29 grid. An investigation of a
reduced number of data sets showed that the algorithm was able to consistently reach the
8th best combination (out of 58,905) of sites within 3 seconds, as opposed to an
exhaustive method for selecting channels that would take 1 ½ days.

Application of this algorithm to data from 10 subjects allowed optimal sites to be selected
for each subject. A comparison of the performance of these sites as opposed to
symmetrically placed sites, or sites chosen by a PCA algorithm or a SFS algorithm was
presented in Chapter 5. This investigation showed that the accuracy of a NN classifier
could be increased from 85% for symmetrically based sites to 95% using optimized sites

### 6.1 Impact and contributions

In the existing literature, placement of electrodes relied on a symmetric array, or a simple
data reduction algorithm. While the first approach is useful in that it is simple and easy to
replicate, there is the potential that improvements in control accuracy could be made by optimally selecting data acquisition sites. To this end, the second method used a simple data reduction algorithm, however this algorithm may select a set of channels that is not as close to optimal as those that more sophisticated algorithms could produce. To improve upon this, a quantitative approach to electrode placement is presented in this thesis. The contributions of this thesis to the field of prosthetic control are summarised as follows;

1. Iterative action potential detection and matched filtering:

   This demonstrated that it improved SNR quickly, and can be applied to future electromyographic studies where it is desired to reduce the noise content in a signal prior to analysis, with processing time being less than 10% the total data length.

2. Data set reduction algorithm:

   The algorithm developed for optimization of electrode sites (through Mahalanobis distance maximisation) can be applied to any study where a method is required to select a smaller data subset when the interaction between channels needs to be taken into account. Total processing time for this method was in the order of $1 \times 10^{-5}$ the time taken for an exhaustive check. Depending on the nature of the dataset in question, an appropriate performance metric would need to be specified that would then replace the Mahalanobis distance as the objective function in this algorithm.

3. Electrode location:

   Electrode placement provides a means to determine if the accuracy of a classifier could be improved should the location of data acquisition be changed. This also serves to provide a mechanism to validate the choice of electrode sites for future studies, as well as a means to choose electrode sites for prosthetic devices in commercial applications.

6.2 Future work
Chapter 6 Conclusions

While this thesis offers new insights into the placement of electromyographic electrodes and the processing of electromyographic signals, there are clearly further steps to be taken to allow this to be used in commercially available prosthetics.

Most notably, the application of this methodology to a large number of subjects, ideally including amputees, would begin to allow information to be collected about the general population. This investigation may reveal that there are several likely sites that should be considered for each individual, or that each person requires their own unique sites.

The action potential extraction algorithm could potentially be further developed to improve accuracy in prosthetic control. As mentioned in Chapter 3, the extraction method assumed a single shape in order to reduce computational complexity. If additional shapes could be accounted for, it is possible that it may assist in preprocessing EMG signals prior to classification.

The electrodes used in this thesis were commercially available Ag/AgCl electrodes, and the amplification performed using a relatively simple circuit followed by a first order active band-pass filter. An investigation into the effect of different electrodes and a more advanced amplification circuit could lead to further improvements into classification accuracy.

A future study should be conducted into the effect of a filter using multiple action potentials in the template rather than the single action potential used here. As the results presented in Chapter 3 showed an improvement in SNR, it is likely that an improved template extraction method could be used to increase accuracy.

The method presented in this thesis used a Mahalanobis distance based measure of accuracy; this was due to the reduced time it would take to calculate when compared to a neural network accuracy. It was shown that there is a relationship between inter-cluster Mahalanobis distance and the accuracy of a neural network, this relationship was not perfect. While this has been successful in previous studies, it is generally accepted that more sophisticated classifiers can provide higher accuracies. Future work would incorporate these more sophisticated classification mechanisms, however as each
calculation of accuracy requires a training period, the increased computational time required in calculating the accuracy would necessarily have to be considered.

A further use of the work presented in this thesis would be to investigate subsets of each of the grasp types. For example, spherical grasp could be further split into large and small spherical, cylindrical grasp could be split into large and small cylindrical, and pinch grasps could be split to represent objects such as keys or wallets. This would allow application of the work presented in this thesis to finer motor movements of the hand.

As mentioned in Chapter 4, the direction that the derivatives should be taken in from the recorded data were assumed to be a direction of steepest gradient. A future study should test the validity of this assumption and examine the effect of taking derivatives in other directions, most likely the projection of derivatives onto other directions at regular angular displacements.

A real-time prosthetic system will benefit by the use of this technique, if key electrode locations on the forearm are identified prior to fitting. Future development of this system will provide a method to target these regions with finer electrode arrays, already placed in a regular grid. This can then be applied quickly and easily to each patient to identify with greater resolution where electrodes should be placed.

In addition, given that a large amount of forearm EMG data are now available for three different grasp types, an investigation into the time dependency (i.e.) the spread of activity from one point to another over time) of electrical activation could now be attempted. This would provide insight into the spread of electrical activity from its source, and provide useful information for modelling of the musculature of the forearm.

6.3 Summary

A method has been developed to acquire, process and reduce a large number of forearm electromyographic data sets. This method is broken down as follows:
• A method for improving SNR of acquired signals based on an iterative action potential detection algorithm improved signal quality prior to signal processing.

• This method has been applied as a preprocessing step in EMG classification, however it has been shown that this did not improve accuracy.

• Interpolation of irregularly spaced data onto a regular grid allowed comparisons to be made between data sets where one or more channels were missing data.

• An algorithm was developed to enable timely reduction of the number of data sets to be used without exhaustively checking every combination of channels.

• These techniques were applied to determine the optimal electrode locations in order to discriminate between spherical, cylindrical and pinch grasps.

• The effectiveness of these optimal sites was validated using by comparison to existing site selection schemes. For a group of 10 subjects, recognition accuracy was improved from approximately 85-88% using previous site selection algorithms to approximately 95% using the algorithm developed in this thesis.

Through the ability to provide accurate and intuitive control, it is envisaged that these techniques will enable “plug and play” type prosthetics available off-the-shelf in the near future.

### 6.4 Publications

Articles for which I am the primary author:

6.4.1. Conference articles
Section 6.4 Publications


Articles in progress:


%Main.m: main file
%Author Scott Walbran
%Date: 23/1/09
004
005 %Clear workspace
006 clear
007 clc
008 close all
009
010 %Set the interpolation grid density
011 points=10;
012
013 %Check if a file exist for the correct number of points. If not, create the
014 %files, if so, load that file.
015 fid=fopen(sprintf('data%i.mat',points));
016 if fid<0
017     loadAndProcess('pinch',points)
018     loadAndProcess('sphere',points)
019     loadAndProcess('cyl',points)
020     %Clear variables to save memory
021     clear stuff
022     data_1_f=data_1_final;
023     clear input
024     clear data_1
025     clear data_2
026     clear totalmd
027     clear ts
028     save(sprintf('data%i',points))
029 else
030     load(sprintf('data%i',points))
031 end
032 %Create alternate data variables
033 data_1_f=data_1_final;
034 data_2_f=data_2_final;
035 data_3_f=data_3_final;
036
037 %o is used when multiple runs are required in order to obtain statistical
038 %information
039 for o = 1:1
040     disp(['o= ' num2str(o)])
041     %Calculate which channels have SNR above the "factor"
042     factor=0.2;
043     usefult=(abs((mean(magnitude_deriv_all_2)-}
mean(magnitude_deriv_all_1,3))./sqrt(std(magnitude_deriv_all_2,0,3).^2+std(magnitude_deriv_all_1,0,3).^2))>(factor)).*abs((mean(magnitude_deriv_all_2,3)-mean(magnitude_deriv_all_1,3))./sqrt(std(magnitude_deriv_all_2,0,3).^2+std(magnitude_deriv_all_1,0,3).^2)));  
0048  
0049 \text{\textit{m} is used to test the affect of changing things such as the culling factor, grid density, or number of channels to be optimized for}}  
0050 \text{\textit{m} = 5:5}  
0051 for m = 5:5  
0052 disp(['m= ' num2str(m)]);  
0053 \text{%Close all files that are currently open}}  
0054 fclose('all');  
0055 \text{%Clear the variables corresponding to direction}}  
0056 clear direct1  
0057 clear direct2  
0058 clear direct3  
0059 \text{%Redimension the data for directions}}  
0060 for i = 1:size(activefeature1_2,1)  
0061 direct1((i-1)*points+1:i*points,:,1)=squeeze(direction1_final(i,:,:));  
0062 direct2((i-1)*points+1:i*points,:,1)=squeeze(direction2_final(i,:,:));  
0063 direct3((i-1)*points+1:i*points,:,1)=squeeze(direction3_final(i,:,:));  
0064 end  
0065 \text{%Store the directiong variables for use later}}  
0066 direct1_orig=direct1;  
0067 direct2_orig=direct2;  
0068 direct3_orig=direct3;  
0069 \text{%Cull the data based on the original SNRs}}  
0070 for i = 1:points  
0071 for j = 1:points  
0072 if usefuls(i,j)==0  
0073 data_1_final((i-1)*points+j,1,1)=NaN;  
0074 direct1((i-1)*points+j,1,1)=NaN;  
0075 data_2_final((i-1)*points+j,1,1)=NaN;  
0076 direct2((i-1)*points+j,1,1)=NaN;  
0077 data_3_final((i-1)*points+j,1,1)=NaN;  
0078 direct3((i-1)*points+j,1,1)=NaN;  
0079 end  
0080 end  
0081 \text{%Remove the NaNs from the data sets, and store where they were in}}  
0082 [data_1_final,data_2_final,data_3_final,index]=remove_NaNs(data_1_final,data_2_final,data_3_final);  
0083 [direct1,direct2,direct3,index]=remove_NaNs(direct1,direct2,direct3);  
0084 \text{%Calculate the means of the directions}}  
0085 direct1=mean(direct1,2);  
0086 direct2=mean(direct2,2);  
0087 direct3=mean(direct3,2);  
0088 \text{%Store all of the data channels in 1 variable}}  
0089 data=[data_1_final data_2_final data_3_final];  
0090 ns=[size(data_1_final,2) size(data_2_final,2) size(data_3_final,2)];  
0091 tic  
0092 \text{%Calculate which channels are best}}  
0093 [final_comb,chans,comball,m_ds]=mahalanobis_sphere(data,ns,direct1,direct2,direct3,m);  
0094 \text{%Set the number of channels that were optimized for}}  
0095 n=5;  
0096 finalchans=chans;
%Add in the NaNs in order to map back to the original channels using index
index=add_NaNs(index,size(data_1_final,1));
%Redimension the channels variable
for i = 1:length(chans)
    chans(i)=index(chans(i));
    final_chans(i,1)=mod(chans(i)-1,points-1)+1;
    final_chans(i,2)=floor((chans(i)-1)/(points-1))+1;
end
[X,Y,Z,th_use,z_use]=createXYZ(x,y,z,points,useables);
for i = 1:length(chans)
    XYZ(i,1)=X(final_chans(i,1),final_chans(i,2));
    XYZ(i,2)=Y(final_chans(i,1),final_chans(i,2));
    XYZ(i,3)=Z(final_chans(i,1),final_chans(i,2));
end
XYZ(:,3)=XYZ(:,3)+min(z);
Z=Z+min(z);
hold off
surf(X,Y,Z,ones(size(Z)),'facecolor','interp','edgecolor','none');colormap(bone);axis equal;lightangle(30,30)
xlabel('x (mm)')
ylabel('y (mm)')
zlabel('z (mm)')
hold on
axis equal
i=5;
plot3(XYZ(:,1),XYZ(:,2),XYZ(:,3),'k*');
plot3(x,y,z,'r*');
combination=final_comb[43];
directions_final=[mean(direct1_orig')' mean(direct2_orig')' mean(direct3_orig')'];

for i = 1:5
    eval(sprintf('final_direct(i)=directions_final(chans(i),%i);',1+str2num(combination(i))));
end
lengthconst=10;
zeds=lengthconst*ones(1,length(final_direct));
thetas=lengthconst*tan(final_direct);
lengths=sqrt(zeds.^2+thetas.^2);
thetas=lengthconst*thetas./lengths;
zeds=lengthconst*zeds./lengths;
figure;plot(th_use,z_use,'*');xlabel('\theta*r (mm)');ylabel('z (mm)');[sitesth,sitesr,sitesz]=cart2pol(XYZ(:,1),XYZ(:,2),XYZ(:,3));
hold on
sitesth=sitesth*mean(sitesr);
plot(sitesth,sitesz,'k*');
for i = 1:5
    plot([sitesth(i)-thetas(i) sitesth(i)+thetas(i)],[sitesz(i)-zeds(i) sitesz(i)+zeds(i)],'k');
end

7.1.2. loadAndProcess.m

function loadAndProcess(grasp,points)
%loadAndProcess.m: load all the files for a given grasp type and process
%author: Scott Walbran
%date: 23/1/09
%get files that are in the directory
list=dir;

initialize j to 1. This will be used to count which directory we are up to
j=1;
for i = 1:size(list,1)
if (strfind(list(i).name,grasp)&isempty(strfind(list(i).name,'.')))
    stuff(j)=i;
end
%Initialize the variable containing the Magnitude of the derivatives
magnitude_deriv_all_1=zeros(points,points,2);

%Loop through each folder
for a = 1:length(stuff)
    %Load the general data files
    load Electrode_index.txt
    load ElectrodeData_pointsOnly.txt
    load matched_signal.txt

    %Matched filter the data
    matched_signal=matched_signal/sqrt(matchedFilter(matched_signal,matched_signal));

    %Read the "input.txt" file in
    input=dlmread(sprintf('%s/%s',Directory_name,'input.txt'),'	');
    timesfid=fopen(sprintf('%s/active_times_index.mat',Directory_name));
    if timesfid<0
        [active_times_index]=get_actives(input(:,keeps));
        save(sprintf('%s/active_times_index',Directory_name),'

    [keeps,Dud_Electrode_index]=read_dud_index(input,Directory_name);

    Electrode_index = remove_duds(Electrode_index,Dud_Electrode_index);

    UnEmap_electrode=index_UnEmap(Electrode_index);

    [Non_Dud,useables]=tidy_electrodes(ElectrodeData_pointsOnly,Electrode_index);

    [input] = filter_extract(input,matched_signal);

    active_times_index=dlmread(sprintf('%s/%s',Directory_name,'active_times_index.txt'),'	');
    if timesfid<0
        [active_times_index]=get_actives(input(:,keeps));
        save(sprintf('%s/active_times_index',Directory_name),'

    [active_times_index]=active_times_index/60;

    %Find nearest neighbours
    [positive_horiz,negative_horiz,positive_vert,negative_vert] = positioning(Non_Dud,useables);

    %Get x,y,z coordinates
    for i = 1:size(ElectrodeData_pointsOnly,1)
        z(i) = ElectrodeData_pointsOnly(i,2);
        x(i) = (ElectrodeData_pointsOnly(i,1)+438.0.12*z(i));
        y(i) = (ElectrodeData_pointsOnly(i,3)-324.0.16*z(i));
    end

    % Process the data to get features out
    fid=fopen(sprintf('%s/feature1.mat',Directory_name));
    [TH,Z,t,feature1,feature2,feature3,feature4,direction,magnitude_deriv]=features(input,x,y,

    %Save the variables of interest to disk
    save(sprintf('%s/feature1',Directory_name),'

    save(sprintf('%s/feature2',Directory_name),'

    save(sprintf('%s/feature3',Directory_name),'

    save(sprintf('%s/feature4',Directory_name),'

    save(sprintf('%s/magnitude_deriv%i',Directory_name,points),'

    save(sprintf('%s/direction',Directory_name),'

    save(sprintf('%s/t',Directory_name),'

    %Remove any part of the data that is not during an "active" grasping period
### 7.1.3. matchedFilter.m

```matlab
function [y] = matchedFilter(x,signal)
%matchedFilter.m: Matched filter x using signal as a template
%Author: Scott Walbran
%Date: 23/1/09

% Time reverse the signal
for i = 1:length(signal)
    temp(i)=signal(length(signal)+1-i);
end
signal=temp;

% Convolve the signals
for j = 1:size(x,2)
    y(:,j) = conv(x(:,j),signal);
end
```

### 7.1.4. read_dud_index.m

```matlab
%Store the extracted features in a single variable "data_1"
for i = 1:size(activefeature1,1)
    data_1((i-1)*points+1:i*points,:,1)=squeeze(activefeature1(i,:,:));
    data_1((i-1)*points+1:i*points,:,2)=squeeze(activefeature2(i,:,:));
    data_1((i-1)*points+1:i*points,:,3)=squeeze(activefeature3(i,:,:));
end

%Calculate the size of the data file, and save to disk
ns(1)=size(activefeature1,3);
load Directory_name2

% Concatenate the data files for each different directory
if a == 1
    direction1_final=direction;
data_1_final = data_1;
else
    data_1_final=[data_1_final data_1];
direction1_final=shiftdim(direction1_final,1);
direction1_final=[direction1_final shiftdim(direction,1)];
direction1_final=shiftdim(direction1_final,2);
end
end
end

data_1=data_1_final;
return
```
function [keeps,Dud_Electrode_index]=read_dud_index(input,Directory_name)

%red_dud_index.m: either ask the user to identify the dud channels, or read
%the pre-existing data file

%Author: Scott Walbran
%Date: 23/1/09

%attempt to open the dud electrode index
fid=fopen(sprintf('%s/Dud_Electrode_index.txt',Directory_name));

%if the file could not be opened, create it for writing, and ask the user
%which channels contain useful data (by checking through each channel 1 at
%a time)
if fid<0
    fid=fopen(sprintf('%s/Dud_Electrode_index.txt',Directory_name), 'wt');
    j=1;
    k=1;
    for i = 2:129
        plot(input(:,i))
        h=gcf;
        set(h,'Position',[0 0 1000 1000]);
        drawnow
        check=questdlg('Is this useful?', 'Usefulness check', 'Yes', 'No', 'Yes & keep', 'Yes');
        if strcmp(check, 'Yes')
        elseif strcmp(check, 'Yes & keep')
            keeps(k)=i;
            k=k+1;
        else
            fprintf(fid, '%d
',i);
            Dud_Electrode_index(j)=i;
            j=j+1;
        end
    end
    fclose(fid);
    disp(['The total number of dud channels is ' num2str(j)])
else
    fclose(fid);
end

%if the file could be opened, read the data straight from it
Dud_Electrode_index=dlmread(sprintf('%s/Dud_Electrode_index.txt'), '	');
load(sprintf('%s/keeps',Directory_name));
save(sprintf('%s/keeps.mat',Directory_name), 'keeps')

return

7.1.5. remove_duds.m

function final = remove_duds(index,duds)
%remove_duds.m
%Author: Scott Walbran
%Date: 27/01/09

% h=waitbar(0,'Removing dud electrodes');
%Remove all 'Dud' electrodes from the index variable
for i = 1:size(index,1)
    for j = 1:length(duds)
        if index(i,2)==duds(j)
            index(i,1)=0;
        end
    end
end
% waitbar(i/size(index,1),h);
end
% close(h)

final = index;
return

7.1.6. index_UnEmap.m

function UnEmap_electrode=index_UnEmap(Electrode_index)
%index_UnEmap.m: map between the unemap electrodes and the armband
%electrodes
%Author: Scott Walbran
%Date: 23/1/09
UnEmap_electrode = zeros(174,1);
for i = 1:size(Electrode_index,1)
    if Electrode_index(i,1)~=0
        UnEmap_electrode(Electrode_index(i,1)) = Electrode_index(i,2);
    end
end
return

function [Non_Dud,useables]=tidy_electrodes(ElectrodeData_pointsOnly,Electrode_index)
%tidy_electrodes.m: remove any channels that have no data, and create an
%index for them
%Author: Scott Walbran
%Date: 23/1/09
j = 1;
for i = 1:size(Electrode_index,1)
    if Electrode_index(i,1)>0
        Non_Dud(j,:) = ElectrodeData_pointsOnly(Electrode_index(i,1),:);
        useables(j)=Electrode_index(i,1);
        j = j+1;
    end
end
return

function [active_times_index]=get_actives(data)
%Date: 17/08/10
%Author: Scott Walbran
%This function plots the first figure from 'input' and prompts the user to
%enter where the grasp was occurring
% for i = 1:size(data,2)
%     figure
%     plot(sum(data'))
%     drawnow
active_times_index=inputdlg('Please enter vector: ');
active_times_index=cell2struct(active_times_index,'active_times_index',1);
active_times_index=str2num(active_times_index.active_times_index);
return
7.1.10. features.m

```matlab
function [TH,Z,t,feature1,feature2,feature3,feature4,Direction,magnitude_deriv]=features(input,x,y,z,matched_signal,useables,UnEmap_electrode,points)

%features.m: interpolate onto a regular grid, and extract the required features
%Author: Scott Walbran
%Date: 24/02/09

%Set the number of data points to use for the time window
const=60;

%Create a regular grid to interpolate over
[theta,r,z]=cart2pol(x,y,z);
theta=theta*mean(r);
[ptop,pbot]=findends(theta,z,r);
THs=linspace(-3*pi,3*pi,3*points)*mean(r);
Zs=linspace(0.96*min(z),0.96*max(z),points);
[TH,Z]=meshgrid(THs,Zs);
for i = 1:length(useables)
    th_use(i)=theta(useables(i));
    z_use(i)=z(useables(i));
end

%Duplicate the data +/- 2pi
th_use=[th_use-2*pi*mean(r) th_use th_use+2*pi*mean(r)];
z_use=[z_use z_use z_use];

%Initialize the features to zero. Only 3 features are being extracted at this stage, so "feature4" should end up untouched
feature1=zeros(points,points,floor((size(input,1)-const)/const)+1);
feature2=zeros(points,points,floor((size(input,1)-const)/const)+1);
feature3=zeros(points,points,floor((size(input,1)-const)/const)+1);
feature4=zeros(points,points,floor((size(input,1)-const)/const)+1);
t=zeros(1,floor((size(input,1)-const)/const)+1);
Direction=zeros(points,points,floor((size(input,1)-const)/const)+1);

%Loop through the data in steps of "const"
for j = 1:const:size(input,1)-const
    %Interpolate on to a regular grid using "porridge" then calculate the magnitude and direction of derivatives
    [th_d,z_d]=porridge(input(j:j+60,:),th_use,z_use,TH,Z,matched_signal,useables,UnEmap_electrode,points);
    magnitude_deriv(:,:,,(j-1)/const+1)=sqrt(th_d(:,:,1).^2+z_d(:,:,1).^2);
    Direction(:,:,,(j-1)/const+1)=mean(atan2(th_d,z_d),3);
    magnitude=sqrt(th_d.^2+z_d.^2);

    %Extract each feature
    feature1(:,:,,(j-1)/const+1)=IAV(magnitude);
    feature2(:,:,,(j-1)/const+1)=DAV(magnitude);
    feature3(:,:,,(j-1)/const+1)=zerox(atan2(th_d,z_d));
    feature4(:,:,,(j-1)/const+1)=0;
    t((j-1)/const+1)=input(j,1);
end

%Set up a low pass filter
[B,A]=butter(3,0.1,'low');

%In the penultimate step for feature 2, low pass filter the data
for i = 1:size(feature2,1)
    for j = 1:size(feature2,2)
        feature2(i,j,:)=filter(B,A,feature2(i,j,:));
    end
end

%Calculate 1-step derivatives for feature2, and ensure it is the same size as the other features
feature2=feature2(:,:,2:size(feature2,3))-feature2(:,:,1:size(feature2,3)-1);
feature2(:,:,size(feature2,3)+1)=feature2(:,:,size(feature2,3));

return
```

7.1.11. findends.m
function [Ptop,Pbot]=findends(theta,z,rs)
%findends.m: find a polynomial fit for the bottom and top of the data set
%Author: Scott Walbran
%Date: 25/03/10

%==========================================================================
%find top of arm
derp=z(2:length(z))-z(1:length(z)-1);
j=1;
for i = 1:length(z)-1
    if derp(i)<0
        v1(j)=theta(i);
        v2(j)=z(i);
        j=j+1;
    end
end
v1(7:8)=v1(8:9);
v2(7:8)=v2(8:9);
v2=v2(1:9);
v1(10)=theta(174);
v2(10)=z(174);
v1=[v1-2*pi*mean(rs) v1 v1+2*pi*mean(rs)];
v2=[v2 v2 v2];
%==========================================================================
Ptop=polyfit(v1,v2,10);

%==========================================================================
%find bottom of arm
derp=z(2:length(z))-z(1:length(z)-1);
j=1;
for i = 1:length(z)-1
    if derp(i)<0
        v1(j)=theta(i+1);
        v2(j)=z(i+1);
        j=j+1;
    end
end
v1=[theta(1) v1(1:12)];
v2=[z(1) v2(1:12)];
v1=[v1-2*pi*mean(rs) v1 v1+2*pi*mean(rs)];
v2=[v2 v2 v2];
%==========================================================================
Pbot=polyfit(v1,v2,20);

return

7.1.12. porridge.m
7.1.13. placeholder.m

```matlab
function [out]=placeholder(input,a_t_index,t)
% Placeholder.m removes any area of data
% Author: Scott Walbran
% Date: 26/07/09

k = 1;
% loop through each "active" time
for i = 1:size(a_t_index,1)
  % check where in the "time" (t) vector this interval falls.
  for j = 1:length(t)
    if (a_t_index(i,1)<j)&&(j<a_t_index(i,2))
      % If the jth time is in "active" range, add it to the output
      % tensor
      out(:,:,k)=input(:,:,j);
      k=k+1;
    end
  end
end
return
```

7.1.14. Remove_NaNs.m

```matlab
function [data_1,data_2,data_3,index]=remove_NaNs(data_1,data_2,data_3)
% Author: Scott Walbran
% Date: 08/06/2010
% removes_NaNs.m: removes any rows of data_1, data_2 and data_3 that start with NaN

i=1;
j=1;
while i<=size(data_1,1)
  if isnan(data_1(i,1,1))
    data_1(i,:,:)=[];
    data_2(i,:,:)=[];
    data_3(i,:,:)=[];
    index(j)=i+j-1;
    j=j+1;
  else
    i=i+1;
  end
end
while i<=size(data_2,1)
  if isnan(data_2(i,1,1))
    data_1(i,:,:)=[];
    data_2(i,:,:)=[];
    data_3(i,:,:)=[];
    index(j)=i+j-1;
    j=j+1;
  else
    i=i+1;
  end
end
while i<=size(data_3,1)
  if isnan(data_3(i,1,1))
    data_1(i,:,:)=[];
    data_2(i,:,:)=[];
    data_3(i,:,:)=[];
    index(j)=i+j-1;
    j=j+1;
  else
    i=i+1;
  end
end
return
```
7.1.15. **Mahalanobis_sphere.m**

```matlab
function [final_comb,chans,allcombs,m_ds]=mahalanobis_sphere(data,ns,direction1,direction2,direction3,n)
  %Author: Scott Walbran
  %Date:28/05/10
  % mahalanobis_sphere.m: calculate from "data" which "n" channels are the best for
  % maximizing accuracy in PCA/cluster analysis system by maximizing the
  % mahalanobis distance between each cluster.

  RandStream.setDefaultStream(RandStream('mt19937ar','seed',sum(100*clock)));
  count=0;
  m_ds=0;
  m_d=0;

  %Pick a random selection of channels to be used as our starting seed
  chans=ceil(size(data,1)*rand(n,1));

  %Store the data for those channels in pots
  for i = 1:n
    pots((i-1)*3+1:i*3,:)=[data(chans(i),:,1);data(chans(i),:,2);data(chans(i),:,3)];
  end

  %m_d_old is used to compare which m_d is better
  m_d_old=0;

  %initialize the combination refering to which direction to be used
  for f = 1:n
    combination(f)='0';
  end

  %Loop through each binary combination possible for the number of channels
  for k=1:n
    temp_comb=combination;
    for bin=0:2
      temp_comb(k)=num2str(bin);
      %based on the current binary combination, replace the potentials
      [pots_temp]=projectdirections(pots,ns,direction1,direction2,direction3,temp_comb,chans);
      [count,m_d_new]=mahalanobis_distance_sphere(pots_temp,ns,count);
    end
    %compare direction 1 with direction 2 and direction 3 and calculate the correct combination to store
    if m_d_new>m_d_old
      m_d_old=m_d_new;
      combination=temp_comb;
    end
  end

  %Loop to test each channel in turn
  for j = 1:size(data,1)-1
    if min(min(abs(chans-(j+1))))
      %Define the data to be used for the current optimization as the %data(channels), where channels is the current list of optimal channels
      for i = 1:n
        pots((i-1)*3+1:i*3,:)=[data(chans(i),:,1);data(chans(i),:,2);data(chans(i),:,3)];
      end
      %Replace the current potentials with the potentials in the correct %direction
      [pots]=projectdirections(pots,ns,direction1,direction2,direction3,combination,chans);
      %calculate the mahalanobis distance for the current optimal channels,
      %store it in the n+1th entry of m_d for use in later if statement
      cl=clock;
      [count,m_d(n+1)]=mahalanobis_distance_sphere(pots,ns,count);
      cl=clock-
    end
  end

  %Loop to replace each entry in "current" channels with the next entry
```

---

**Section 7.1 Appendix 1 – Matlab code**

147
% from data set, and calculate the mahalanobis distance
for i = j+1:j+n
    % k represents each direction (0th, 1st or 2nd)
    for k = 0:2
        temp_chans=chans;
        % Replace the "i"th entry with the next channel to be tested
        temp_chans(i-j)=j+1;
        temp_combination=combination;
        temp_combination(n-(i-j)+1)=num2str(k);
        % pots stores the data for analysis from the current channels to be
determined
        for l = 1:n
            pots((l-1)*3+1:l*3,:)=[data(temp_chans(l),:,1);data(temp_chans(l),:,2);data(temp_chans(l),:,3)];
        end
        % Replace the current potentials with the potentials in the correct
        % direction
        [pots]=projectdirections(pots,ns,direction1,direction2,direction3,temp_channels);
    end
end
for k = 0:2
    temp_chans=chans;
    % Replace the "i"th entry with the next channel to be tested
    temp_chans(i-j)=j+1;
    temp_combination=combination;
    temp_combination(n-(i-j)+1)=num2str(k);
    % pots stores the data for analysis from the current channels to be
determined
    for l = 1:n
        pots((l-1)*3+1:l*3,:)=[data(temp_chans(l),:,1);data(temp_chans(l),:,2);data(temp_chans(l),:,3)];
    end
    % Replace the current potentials with the potentials in the correct
    % direction
    [pots]=projectdirections(pots,ns,direction1,direction2,direction3,temp_channels);
end
% Find the maximum mahalanobis distance of the current samples
[wicked,index]=max(m_d);
% % Calculate the mahalanobis distance of those clusters
switch k
    case 0
        count,m_d(i-j)=mahalanobis_distance_sphere(pots,ns,count);
    case 1
        end
    case 2
        m_d(n)=m_d(i-j);
end
% Find the maximum mahalanobis distance of the current samples
[end
1004 [wicked,index]=max(m_d);
1005 m_d(n)=wicked;
1006 % if the maximum mahalanobis is found in n+1, leave the channels alone,
1007 if index>n+1
1008     chans(index)=j+1;
1009     combination(n-(index)+1)=sprintf('%i',bestDirection(index)-1);
1010 end
1011 end
1012 end
1013 final_comb([42]=combination;
1017 disp(['final count of Mahalanobis_distance calls: ' num2str(count)])
1018 return

7.1.16. Projectdirections.m

function
[outpots]=projectdirections(pots,ns,direction1,direction2,direction3,binary,chans)
% projectdirections.m: switch case based on which direction is to be used
% Author: Scott Walbran
% Date: 25/03/11
for l = 1:size(pots,1)/3
    switch str2num(binary(size(pots,1)/3-1+1));
    case 0
        outpots((l-1)*3+1:l*3,1:ns(1))=pots((l-1)*3+1:l*3,1:ns(1));
        outpots((l-1)*3+1:l*3,ns(1)+1:ns(1)+ns(2))=pots((l-1)*3+1:l*3,ns(1)+1:ns(1)+ns(2))*cos(direction2(chans(l))-direction1(chans(l)));
        outpots((l-1)*3+1:l*3,ns(1)+ns(2)+1:sum(ns))=pots((l-1)*3+1:l*3,ns(1)+ns(2)+1:sum(ns))=
148
1) *3+1:1*3, ns(1)+ns(2)+1:sum(ns)) *cos(direction3(chans(l)) - direction1(chans(l)));  
0013 case 1  
0014     outputs((l-1)*3+1:ns(1))=pots((l-1)*3+1:1*3,ns(1)) *cos(direction2(chans(l)) - direction1(chans(l)));  
0015     outputs((l-1)*3+1:1*3,ns(1)+ns(1)+ns(2))=pots((l-1)*3+1:1*3,ns(1)+1:ns(1)+ns(2));  
0016     outputs((l-1)*3+1:1*3,ns(1)+ns(2)+1:sum(ns))=pots((l-1)*3+1:1*3,ns(1)+ns(2)+1:sum(ns)) *cos(direction3(chans(l)) - direction2(chans(l)));  
0017 case 2  
0018     outputs((l-1)*3+1:1*3,ns(1)+ns(1))=pots((l-1)*3+1:1*3,ns(1)+ns(1)) *cos(direction3(chans(l)) - direction1(chans(l)));  
0019     outputs((l-1)*3+1:1*3,ns(1)+ns(1)+ns(2))=pots((l-1)*3+1:1*3,ns(1)+ns(1)+ns(2)) *cos(direction3(chans(l)) - direction2(chans(l)));  
0020     outputs((l-1)*3+1:1*3,ns(1)+ns(2)+1:sum(ns))=pots((l-1)*3+1:1*3,ns(1)+ns(2)+1:sum(ns));  
0021 end  
0022 end  
0023 return  

7.1.17. Mahalanobis_distance_sphere

0001 function [count,m_d]=mahalanobis_distance_sphere(in,ns,count)  
0002 %mahalanobis_distance_sphere.m: calculates the mahalanobis distances  
0003 %between the 3 input clusters and then returns a geometric average  
0004 %Author: Scott Walbran  
0005 %Date: 07/02/11  
0006  
0007 in1=in(:,1:ns(1)+ns(2)); in2=[in(:,1:ns(1)) in(:,ns(1)+ns(2)+1:sum(ns))]; in3=in(:,ns(1)+1:sum(ns));  
0008 ns1=[ns(1) ns(2)]; ns2=[ns(1) ns(3)]; ns3=[ns(2) ns(3)];  
0009  
0010 [count,m_d1]=mahalanobis_distance(in1,ns1,count);  
0011 [count,m_d2]=mahalanobis_distance(in2,ns2,count);  
0012 [count,m_d3]=mahalanobis_distance(in3,ns3,count);  
0013  
0014 m_d=(m_d1*m_d2*m_d3)^(1/3);  
0015 return  

7.1.18. Mahalanobis_distance.m

0001 function [count,m_d]=mahalanobis_distance(in,ns,count)  
0002 %mahalanobis_distance.m: calculates the mahalanobis distance between 2  
0003 %clusters present in “in”  
0004 %Author: Scott Walbran  
0005 %Date: 26/09/09  
0006  
0007 %Calculate S1 and S2, the covariance matrices  
0008 S1=cov(in(:,1:ns(1))');  
0009 S2=cov(in(:,ns(1)+ns(2)+1:sum(ns))');  
0010 x=mean(in(:,1:ns(1))');  
0011 mu=mean(in(:,ns(1)+ns(2)+1:sum(ns))');  
0012 S=(ns(1)*S1+ns(2)*S2)/(sum(ns)-2);  
0013 if (rcond(S)<1e-10)  
0014     m_d=-1;  
0015 else  
0016     m_d=sqrt((x-mu)*inv(S)*(x-mu)');  
0017 end  
0018 count=count+1;  
0019 return  

7.1.19. createXYZ.m

0001 function [X,Y,Z,th_use,z_use]=createXYZ(x,y,z,points,useables)  
0002 %CreateXYZ.m: create a 3D mesh for use in plotting  
0003 %Author: Scott Walbran  
0004 %Date: 29/05/10
function out = DAV(in)
%DAV.m: calculates the DAV feature
%Author: Scott Walbran
%Date: 23/01/09
in = abs(in);
out = max(in,[],3);
return

function out = IAV(in)
%IAV.m: take a signal, and calculates the IAV(integral of absolute value)
%Author: Scott Walbran
%Date: 24/01/09
in = abs(in);
out=sum(in,3);
return

function out = zerox(input)
%zerox.m: take a signal, and calculates the number of zero crossings
%Author: Scott Walbran
%Date: 24/01/09
out=zeros(size(input,1),size(input,2));
for j = 1:size(input,1)
    for k = 1:size(input,2)
        for i = 1:size(input,3)-1
            if sign(input(j,k,i))~=sign(input(j,k,i+1))
                out(j,k)=out(j,k)+1;
            end
        end
    end
end
return
%AccuracyCheck.m
%Author: Scott Walbran
% Date: 21-2-11
 This m file checks the accuracy of a randomly selected set of channels based on the UnEmap data collected against the accuracy of channels selected by my optimization algorithm. It will check using a classifier function, which I intend to have a Mahalanobis distance classifier and a NN classifier. Where available, it will also check the optimal sites as determined by an exhaustive checking method.
clear
clc
close all
%load the data file as already previously processed.
load data29
factor=0.0;
usefuls=((abs((mean(magnitude_deriv_all_2,3)-mean(magnitude_deriv_all_1,3))./sqrt(std(magnitude_deriv_all_2,0,3).^2+std(magnitude_deriv_all_1,0,3).^2))>(factor)).*abs((mean(magnitude_deriv_all_2,3)-mean(magnitude_deriv_all_1,3))./sqrt(std(magnitude_deriv_all_2,0,3).^2+std(magnitude_deriv_all_1,0,3).^2)));
for i = 1:size(activefeature1_2,1)
direct1((i-1)*points+1:i*points,:,1)=squeeze(direction1_final(i,:,:));
direct2((i-1)*points+1:i*points,:,1)=squeeze(direction2_final(i,:,:));
direct3((i-1)*points+1:i*points,:,1)=squeeze(direction3_final(i,:,:));
end
direct1_orig=direct1;
direct2_orig=direct2;
direct3_orig=direct3;
for i = 1:points
for j = 1:points
if usefuls(i,j)==0
    data_1_final((i-1)*points+j,1,1)=NaN;
    direct1((i-1)*points+j,1)=NaN;
    data_2_final((i-1)*points+j,1,1)=NaN;
    direct2((i-1)*points+j,1)=NaN;
    data_3_final((i-1)*points+j,1,1)=NaN;
    direct3((i-1)*points+j,1)=NaN;
end
end
data_3=data_3_final;
data_2=data_2_final;
data_1=data_1_final;

%Determine the set of channels to use, and the directions the data will be recorded in
chansSelected=[99 69 130 129 44
  659 564 187 158 639
  158 483 558 187 455
  157 132 652 605 187
  187 639 640 564 633
  187 682 158 508 605
  187 639 633 640 564
  44 40 691 185 41
  44 483 185 186 455 ];

n=5;

%Determine the set of channels to use, and the directions the data will be recorded in
chansSelected=[99 69 130 129 44
  659 564 187 158 639
  158 483 558 187 455
  157 132 652 605 187
  187 639 640 564 633
  187 682 158 508 605
  187 639 633 640 564
  44 40 691 185 41
  44 483 185 186 455 ];
combinationSelected=['02202'; '20200'; '02110'; '00101'; '00122'; '00022'; '01022'; '11000'; '00220'; '11000'];

chansAnat=[136 118 234 397 581];

combinationAnat='11111';

for i = 1:size(chansSelected,1)

    [AccuracySelected(i)]=MahalanobisAccuracy(data,chansSelected(i,:),combinationSelected(i,:),ns,direct1,direct2,direct3);

    AccuracySelectedNN(i)=NNAccuracy(data,chansSelected(i,:),combinationSelected(i,:),ns,direct1,direct2,direct3);

end

[AccuracyAnat]=MahalanobisAccuracy(data,chansAnat,combinationAnat,ns,direct1,direct2,direct3);

AccuracyAnatNN=NNAccuracy(data,chansAnat,combinationAnat,ns,direct1,direct2,direct3);

for i = 1:1

    chansRand=ceil(size(data,1)*rand(n,1));

    combinationRand=dec2bin(floor((2^n-1)*rand),n);

    [AccuracyRand(i)]=MahalanobisAccuracy(data,chansRand,combinationRand,ns,direct1,direct2,direct3);

    [AccuracyRandNN(i)]=NNAccuracy(data,chansRand,combinationRand,ns,direct1,direct2,direct3);

end

7.1.24. MahalanobisAccuracy.m

function [AccuracyOut]=MahalanobisAccuracy(data,chans,combination,ns,direction1,direction2,direction3)

% Takes the data contained in chans and in the direction given by combination, uses 10% of it to create a classifier and finds the accuracy of the classifier using the remaining 90% of the data

AccuracyOut=5;

n=length(chans);

% Split the data into the two different grasp types, and then separate them into Classifier training data and checking data

for i = 1:n

    pots((i-1)*3+1:i*3,:)=[data(chans(i),:,1);data(chans(i),:,2);data(chans(i),:,3)];

end

[pots]=projectdirections(pots,ns,direction1,direction2,direction3,combination,chans);

output=zeros(1,size(dataChecker,2));
Section 7.1 Appendix 1 – Matlab code

```matlab
S1 = cov(data1Classifier');
S2 = cov(data2Classifier');
S3 = cov(data3Classifier');
for i = 1:size(dataChecker,2)
output(i) = MahalanobisClassifier(dataChecker(:,i), data1Classifier, data2Classifier, data3Classifier, S1, S2, S3);
end
idealOutputs = [ones(1, size(data1Checker,2)) 1+ones(1, size(data2Checker,2)) 2+ones(1, size(data3Checker,2))];
AccuracyOut = (sum(idealOutputs == output)) * 100 / length(output);
return
```

7.1.25. **NNAccuracy.m**

```matlab
function [AccuracyOut] = NNAccuracy(data, chans, combination, ns, direction1, direction2, direction3)
%MahalanobisClassifier.m
%Author: Scott Walbran
%Date: 21-2-11
% Takes the data contained in chans and in the direction given by
% combination, uses 10% of it to create a classifier and finds the accuracy
% of the classifier using the remaining 90% of the data
n = length(chans);
for i = 1:n
    pots((i-1)*3+1:i*3,:) = [data(chans(i),:,:); data(chans(i),:,:); data(chans(i),:,:)];
end
const = 0.3;
data1 = pots(:,1:ns(1));
data2 = pots(:,ns(1)+1:ns(1)+ns(2));
data3 = pots(:,ns(1)+ns(2)+1:sum(ns));
data1Classifier = data1(:,1:round(size(data1,2)*const),:);
data2Classifier = data2(:,1:round(size(data2,2)*const),:);
data3Classifier = data3(:,1:round(size(data3,2)*const),:);
data1Checker = data1(:,1:size(data1,2),:);
data2Checker = data2(:,1:size(data2,2),:);
data3Checker = data3(:,1:size(data3,2),:);
et = newff([data1Classifier data2Classifier data3Classifier], [ones(1, size(data1Classifier,2)) 1+ones(1, size(data2Classifier,2)), 2+ones(1, size(data3Classifier,2))], 10);
et = train(net, [data1Classifier data2Classifier data3Classifier], [ones(1, size(data1Classifier,2)) 1+ones(1, size(data2Classifier,2)), 2+ones(1, size(data3Classifier,2))], 10);
dataChecker = [data1Checker data2Checker data3Checker];
output = zeros(1, size(dataChecker,2));
for i = 1:size(dataChecker,2)
    output(i) = NNClassifier(dataChecker(:,i), net);
end
idealOutputs = [ones(1, size(data1Checker,2)) 1+ones(1, size(data2Checker,2)) 2+ones(1, size(data3Checker,2))];
AccuracyOut = (sum(idealOutputs == output)) * 100 / length(output);
return
```

7.1.26. **NNClassifier.m**
function [ClusterOut]=NNClassifier(point,net)
%MahalanobisClassifier.m
%Author: Scott Walbran
%Date: 21-2-11
% Takes the data from point and decides if it belongs to either of the
% Clusters passed in
ClusterOut=round(sim(net,point));
return
This code was used to acquire, process and classify data from five differential EMG circuits for real time accuracy determination.
7.3 Appendix 3 – Ethical approval

Office of the Vice-Chancellor
Research Integrity Unit

The University of Auckland
Private Bag 92019
Auckland, New Zealand
Level 10, 49 Symonds Street
Telephone: 64 9 373 7599
Extension: 87830 / 83761
Facsimile: 64 9 373 7432

UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS
ETHICS COMMITTEE

20-Apr-2012

MEMORANDUM TO:

Assoc Prof Iain Anderson
Engineering Science

Re: Application for Ethics Approval (Our Ref. 7932)

The Committee considered your application for ethics approval for your project titled Optimization of electrode placement in electromyographic control of prostheses on 20-Apr-2012.

Ethics approval was given for a period of three years.

The expiry date for this approval is 20-Apr-2015.

If the project changes significantly you are required to resubmit a new application to the Committee for further
consideration.

In order that an up-to-date record can be maintained, you are requested to notify the Committee once your project is completed.

The Chair and the members of the Committee would be happy to discuss general matters relating to ethics approvals if you wish to do so. Contact should be made through the UAHPEC secretary at humanethics@auckland.ac.nz in the first instance.

All communication with the UAHPEC regarding this application should include this reference number: 7932.

(This is a computer generated letter. No signature required.)

Secretary
University of Auckland Human Participants Ethics Committee

c.c. Head of Department / School, Engineering Science
Mr Scott Walbran
Prof Peter Hunter

Additional information:
1. Should you need to make any changes to the project, write to the Committee giving full details including revised documentation.

2. Should you require an extension, write to the Committee before the expiry date giving full details along with revised documentation. An extension can be granted for up to three years, after which time you must make a new application.

3. At the end of three years, or if the project is completed before the expiry, you are requested to advise the Committee.

4. Do not forget to fill in the 'approval wording' on the Participant Information Sheets and Consent Forms, giving the dates of approval and the reference number, before you send them out to your participants.

5. Send a copy of this approval letter to the Manager - Funding Processes, Research Office if you have obtained funding other than from UniServices. For UniServices contract, send a copy of the approval letter to: Contract Manager, UniServices.

6. Please note that the Committee may from time to time conduct audits of approved projects to ensure that the research has been carried out according to the
approval that was given.
CONSENT FORM
Participant
THIS FORM WILL BE HELD FOR A PERIOD OF 6 YEARS

Project title: Optimization of Electrode Placement in Electromyographic control of prostheses
Names of Researchers: Scott Walbran (Phd Student, Auckland Bioengineering Institute), Associate Professor Iain Anderson

I have read the Participant Information Sheet, have understood the nature of the research and why I have been selected. I have had the opportunity to ask questions and have them answered to my satisfaction.

- I agree to take part in this research.
- I understand that my data will be coded in order to best preserve my confidentiality and that any information I provide that is published will be done in such a way as to not identify me as the source.
- I understand that I am free to withdraw participation at any time, and to withdraw any data traceable to me up to the 1st of July 2012.
- I understand that data will be kept for 6 years, after which they will be destroyed.

I DO/DO NOT (Please circle one) wish to be provided with a summary of the research findings to the following email address:

________________________________________

Name ________________________________

Signature _____________________________ Date __________________

APPROVED BY THE UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE ON 20-Apr-2012 FOR (3) YEARS REFERENCE NUMBER 7932
PARTICIPANT INFORMATION SHEET

Participant

Project title: Optimization of Electrode Placement in Electromyographic control of prostheses
Names of Researchers: Scott Walbran (Phd Student, Auckland Bioengineering Institute), Associate Professor Iain Anderson

Researcher introduction

Scott Walbran is a PhD student with the Auckland Bioengineering Institute. Associate Professor Anderson is a Principal Investigator with the Auckland Bioengineering Institute, and an Associate Professor with the department of Engineering Science.

Project description and invitation

The aim of this project is to determine optimal sites to place electrodes on a human forearm in order to control a prosthetic device. These sites will then be evaluated to determine just how “optimal” they are compare to other sites. We invite able limbed people to take part in this study. Unfortunately due to the nature of the study, it is required that participants have healthy arms. Some of the equipment used in this study however is very specialized, and the ideal candidate would have a medium sized male arm. Participation is entirely voluntary, and you may withdraw at any stage of the study.

Project Procedures

The project will take place in four steps. The first step will involve preparation of the skin in order to record electromyographic (EMG) signals. Each remaining step will involve recording of varying number of EMG signals from the forearm, and signal processing of those signals.

The first step in the procedure will be to prepare the skin on the forearm in order to produce cleaner signals. This procedure will be as follows:
1. Shaving the hair off the arm.
2. Scrubbing the skin using an abrasive paste.
3. Cleaning the skin with isopropyl alcohol to remove any residual abrasive paste.

*Warning!* This part of the procedure – in particular step 2 – can cause discomfort. If you find you would not like to proceed at this point, please let the researcher know immediately.

This step in the procedure will take approximately 5 minutes.

There is also the possibility of an allergic reaction to the abrasive paste or the adhesive used in attaching the electrodes for EMG recording.

During the recording stage, there is a very small chance of equipment malfunction. This could potentially result in current being passed back to the participant who would feel a mild electrical shock.

The second step involves taking a single channel EMG recording. A single patch containing three electrodes will be placed onto the forearm, which will in turn have a small circuit attached to it. This circuit will be connected to a desktop PC which will record and process data. Once the circuit is attached, the recording software will be switched on, and you will be asked to squeeze an object as tightly as you can. This object has force transducers in it – this allows all future recordings to be made based against your “maximum voluntary contraction” (MVC). You will also be asked to squeeze at approximately 25%, 50% and 75% as tightly as you can. The data will then be processed for use in the next step.

The total time for this step will be approximately 10 minutes – though most of this time will be in data processing.

The third step involves taking EMG signals from the forearm using a silicone armband with 128 embedded electrodes. This armband will be strapped to your forearm, and you will be asked to grasp three objects of different shapes, one after another. Each object will be grasped for approximately 20 seconds in total, over the course of 1 minute. While each object is being grasped, data will be recorded from all of the electrode sites. This data are then passed into an optimization programme that will determine which of the 128 locations is best to use in the final stage.

The total time for this step will be approximately 2 hours – 10 minutes for the data recording, and 110 minutes for the optimization programme to run. During this time, if you would like, you will be taken on a tour of the Auckland Bioengineering Institute, and asked to join the researchers for a cup of coffee.

The final step involves taking EMG recordings from 3-5 sites, in two different locations. These locations are determined in the first instance based on the muscles in the forearm, and in the second instance based on the results from the third step. For each set of locations, you will be prompted (by computer code) to grasp one of the three objects from step 3. After an initial period where the computer “learns” which object you are grasping based on the EMG signals it receives, the computer will then start guessing which object you are grasping, while still prompting you to grasp different objects. The computer will then compare how many times it correctly guessed the object it asked you to grasp.

This step will take approximately 20 minutes.

The total time involvement will be approximately 4 hours over 1 day.
Data storage/retention/destruction/future use

The collected data will be retained for 6 years beyond the duration of the study. This will allow for any future study to use the same data for corroboration. The data will be stored on magnetic disk for this time. Data will be destroyed at the end of the six years by erasure from the disk. The data in this study will be used for presentation in the PhD thesis of Scott Walbran, and for several academic articles and presentations. Should you wish a copy of the data pertaining to you, please request this from Scott (details below) who will make an electronic copy for you, or fill out the Section in the Consent Form.

Right to Withdraw from Participation

Participants have the right to withdraw from participation at any time. You are also provided the opportunity to withdraw your data up to 1st of July 2012.

Anonymity and Confidentiality

All personal details will be kept confidential. Data will be coded rather than having your name directly associated with it – in this way the data cannot be directly identified as yours. If the information you provide is reported/published, this will be done in a way that does not identify you as its source.

Contact Details and Approval Wording

Researcher contact details:
Scott Walbran, Auckland Bioengineering Institute, The University of Auckland, Private Bag 92019, Auckland 1142. Telephone 09 373-7599 extn. 88363. Email s.walbran@auckland.ac.nz

Supervisor contact details:
Associate Professor Iain Anderson, Auckland Bioengineering Institute, The University of Auckland, Private Bag 92019, Auckland 1142. Telephone 09 373-7599 extn. 82465. Email i.anderson@auckland.ac.nz

HOD details:
Professor Peter Hunter, Auckland Bioengineering Institute, The University of Auckland, Private Bag 92019, Auckland 1142. Telephone 09 373-7599 extn. 88395. Email p.hunter@auckland.ac.nz

Chair contact details: “For any queries regarding ethical concerns you may contact the Chair, The University of Auckland Human Participants Ethics Committee, The University of Auckland, Research Office, Private Bag 92019, Auckland 1142. Telephone 09 373-7599 extn. 87830/83761. Email: humanethics@auckland.ac.nz.”
APPROVED BY THE UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE ON 20 April 2012 for (3) years, Reference Number 7932
### 7.4 Appendix 4 – Subject information

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject Gender</th>
<th>Subject Age</th>
<th>Subject health</th>
<th>Time taken to complete NN classification (± 1 SD) (seconds)</th>
<th>Time taken to calculate MD (± 1 SD) (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL1</td>
<td>Male</td>
<td>22</td>
<td>Healthy</td>
<td>31.14 ± 3.38</td>
<td>0.0079 ± 0.0021</td>
</tr>
<tr>
<td>HL2</td>
<td>Male</td>
<td>27</td>
<td>Healthy</td>
<td>25.77±4.84</td>
<td>0.0038±0.00057</td>
</tr>
<tr>
<td>HL3</td>
<td>Male</td>
<td>22</td>
<td>Healthy</td>
<td>37.14±14.04</td>
<td>0.0051±0.00091</td>
</tr>
<tr>
<td>HL4</td>
<td>Male</td>
<td>21</td>
<td>Healthy</td>
<td>41.46±25.68</td>
<td>0.0053±0.000724</td>
</tr>
<tr>
<td>HL5</td>
<td>Female</td>
<td>23</td>
<td>Healthy</td>
<td>31.15±7.44</td>
<td>0.0043±0.00073</td>
</tr>
<tr>
<td>HL6</td>
<td>Male</td>
<td>21</td>
<td>Healthy</td>
<td>35.94±10.11</td>
<td>0.019±0.035</td>
</tr>
<tr>
<td>HL7</td>
<td>Male</td>
<td>23</td>
<td>Healthy</td>
<td>39.21±25.34</td>
<td>0.0042±0.00062</td>
</tr>
<tr>
<td>HL8</td>
<td>Male</td>
<td>27</td>
<td>Healthy</td>
<td>58.89±44.97</td>
<td>0.0041±0.00053</td>
</tr>
<tr>
<td>HL9</td>
<td>Male</td>
<td>30</td>
<td>Healthy</td>
<td>39.99±26.5</td>
<td>0.0054±0.0014</td>
</tr>
<tr>
<td>HL10</td>
<td>Male</td>
<td>30</td>
<td>Healthy</td>
<td>49.69±18.11</td>
<td>0.0146±0.0072</td>
</tr>
</tbody>
</table>
References


J.-U. Chu, I. Moon, S.-K. Kim, and M.-S. Mun, "Control of a multifunction myoelectric hand using a real-time EMG pattern recognition," presented at...


