NTNU
Norwegian University of
Science and Technology

Myoelectric signal features for upper limb prostheses

Per Ferdinand Bach

Master of Science in Engineering Cybernetics

Submission date:	June 2009
Supervisor:	Øyvind Stavdahl, ITK
Co-supervisor:	Anders Fougner, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

Myoelectric signal features for upper limb prostheses

For the purpose of controlling upper limb prostheses, a common input is the myoelectric signal measured on the surface of the remaining parts of the limb. Several research teams have recently tried to find the signal features with the best performance for controlling the prosthesis (for example Boostani & Moradi, 2003). In a recent term project, our student has tried to implement and evaluate these features.
In this thesis, the student will optimize the techniques formerly used, include wavelet-based techniques, and find the best combination of the signal features. The algorithms will be tested on an existing data set containing myoelectric signals and tracked positions/angles.

This project consists of the following steps:

1. Do a brief literature study on feature extraction and techniques for combining features in a systematic manner.

2. Study the wavelet transform. Give an overview of its properties and how it operates.

3. Augment the feature set with the techniques mentioned in point 2 above, and briefly compare their individual performance to those of the existing feature set.

4. Identify an optimal feature set and evaluate the performance of this feature set. Discuss the implications of your results in relation to a real-world prosthesis controller.

Assignment given: 12. January 2009
Supervisor: Øyvind Stavdahl, ITK

Preface

This thesis is submitted in fulfilment of the degree Master of Science at the Norwegian University of Science and Technology (NTNU), in the Department of Engineering Cybernetics.

As usual in a thesis, problems occur and remain unanswered. This leaves a whole new set of topics to be researched in the future. Prostheses are an important tool for amputees in their daily life, and it is my hope that the future research will be successful, and that my study is a contribution.

I would like to take this opportunity to thank the following people:

Øyvind Stavdahl for being my excellent and inspiring supervisor.

Anders Fougner for being my incredibly helpful advisor.

Parsa Rahmanpour, a very helpful and supporting friend.

Cathrine Bach, my utmost kind and helpful sister, for proofreading this thesis.

Trondheim, June 2009

Per Ferdinand Bach

Contents

List of Figures

List of Tables

Nomenclature

AAC	Average amplitude change
AAV	Averaged absolute value
ANN	Artificial Neural Network
AR	Auto-regressive coefficients
BSS	Backward sequential selection
CC	Cepstrum/cepstral coefficients
CORR	Correlation Coefficient
CST	Cosine Similarity Transform
CWT	Continuous Wavelet Transform
DCT	Discrete Cosinus Transform
DoF	Degrees of Freedom
DWT	Discrete Wavelet Transform
EMG	Electromyography
EWC	Energy of Wavelet Coefficients
EWPC	Energy of Wavelet Packet Coefficients
FSS	Forward sequential selection
FT	Fourier Transform
HIST	Histogram
LDF	Linear discriminant function
LF	Linear mapping function
LTI	Linear time-invariant (filter)
MLP	Multi-Layer Perceptron
MRA	Multiresolution Analysis/Approximation
MUAP	Motor unit action potential
MYOP	Myopulse percentage rate

NN	Neural network
NT	Number of turns
RMSE	Root mean square error
ROM	Range of motion
STFT	Short-time Fourier Transform
SVM	Support vector machines
VAR	Variance
WAMP	Wilson amplitude
WAVE	Wavelength
WFT	Windowed Fourier Transform
WPT	Wavelet Packet Transform
WT	Wavelet Transform
ZC	Zero-crossings

Abstract

In the last couple of years The Institute of Cybernetics at NTNU, Norway, has based its research on the SVEN work carried out in Sweden in the late 1970's. The SVEN hand was an on/off-controlled upper limb prosthesis based on electromyographic (EMG) signals. This master thesis is a part of the renewed and continuing research.

This study will try to identify signal features that are beneficial in a proportional control of a multi-function upper limb prosthesis. The intent is to identify a set of signal features that could be implemented in a practical proportional control system to enhance the movement functions of the prosthesis such that it more closely mimic the movements of a normal upper limb.

The data set used in this paper consist of EMG signals and VICON angle data recorded by Fougner (2007). A short explanation will be given on how to acquire such data.

A brief introduction on feature selection defines the properties of a wrapper and filter approach in search for a feature subset. Wavelets properties are explained and two wavelet techniques are used in order to obtain more information from the EMG signal in addition to existing features. From this, we search for a subset of features that will let us use a mapping function that estimates a correct motion with respect to the features fed to it.

The Cosine Similarity Transform (CST) and the Correlation coefficient (CORR) will in addition to RMSE be investigated in order to find an optimal performance indicator. With a good and reliable indicator we may find a suitable subset.

EWC-WAVE were found to be the best subset according to both CST and RMSE. Based upon the information obtained from each performance indicator, it is suggested that CST should be carried out as a measure of accuracy on how to map data in the future.

There are still unsolved problems. Some of the angles we tried to estimate with a neural network suffered and produced non-informative data. This indicate that one should add more hidden nodes to a neural network when more features are used as input.

We have obtained indications that we do need to combine feature subsets in order to obtain higher accuracy of the estimated signal.

It is proposed that a post-processing technique should be developed and used subsequent to the pattern recognition methods in order to achieve a signal that better reflects the estimation and may be used as a control signal for a prosthesis.

Hopefully will these findings help improve future work to achieve an enhanced proportional control for a real prosthesis.

1 Introduction

In the last couple of years The Institute of Cybernetics at NTNU, Norway, has based its research on the SVEN work carried out on Sweden in the late 1970's. The SVEN hand was an on/off-controlled upper limb prosthesis based on electromyographic (EMG) signals. This master thesis is a part of the renewed and continuing research.

The previous work by Fougner (2007), Rahmanpour (2008) and Bach (2008) will be used as a basis. Fougner recorded EMG signals and VICON angle data from three subjects. In addition, three mapping functions were implemented in Matlab. Rahmanpour implemented another mapping function but it will not be used in this study because of a long calculation time. Feature extractions from Bach will be used in addition to a few new features.

The first part of this thesis will give a brief overview on how the data used in this thesis is acquired.

A great effort will be put in a literature study to gain knowledge on wavelets and how to find a suitable subset of a feature set.

Further, wavelet signal features together with other features (Boostani, R. and Moradi, M. H., 2003) will be tested to achieve a first hand knowledge on what set of features could be useful for a proportional[1] control of a powered upper limb prosthesis.

The features will be applied on EMG signals and will be used to feed pattern recognition methods. These mapping functions will use the VICON angle recordings as referance to estimate angles.

[1] With Proportional control the prosthesis functions are controlled not only by the identification of a motion, but also based on how much or how fast the user want to execute the command.

2 Previous work

All data used in this thesis has been previously recorded, and a more thorough explanation is given in (Fougner, 2007). The reader is therefore adviced to acquire a copy of it in order to be able to reproduce the setup.

The first part will briefly explain what kind of equipment was used, how data was collected and what the data contains. The last part will in a short form present the different classifiers used in this thesis.

2.1 Equipment

2.1.1 EMG sampling equipment

EMG was sampled with a portable multichannel box connected to the VICON system. Eight myoelectrodes were used. Ground point was placed between the two signal electrodes, and the myoelectrode contained a built-in 20x preamplifier. An additional signal ground point electrode was connected to the multi-channel box for zero voltage reference level. The multi-channel box was set to 4000x amplification for all EMG channels (Bach, 2008).

2.1.2 VICON motion measurement

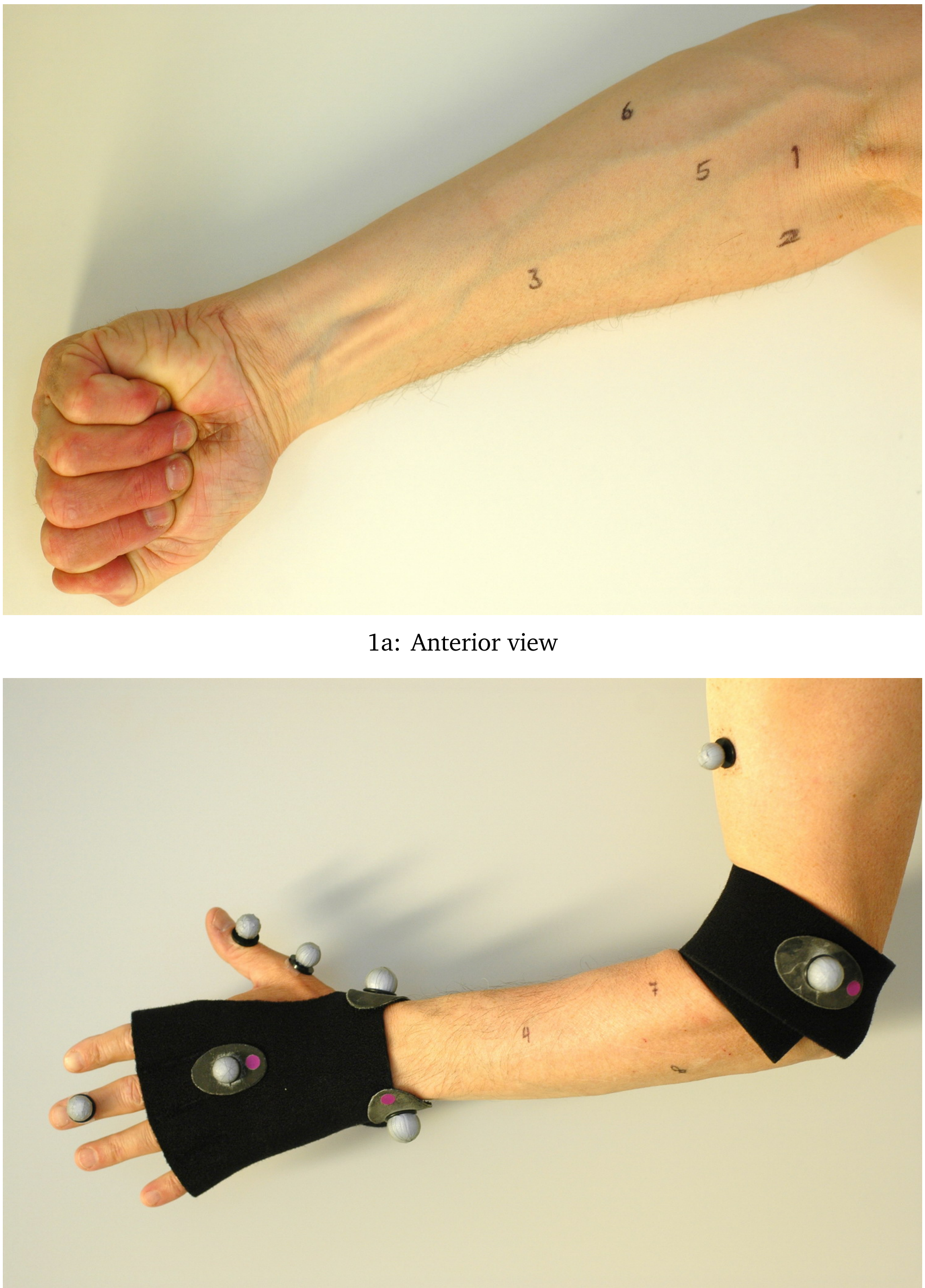
The VICON motion capture system is a video based system to record motion. Markers reflects light emitted by diodes placed around the lens of the camera.

The following information is based on (Fougner, 2007).

A total of 6 cameras were used and all directed towards the centre of the room. The system was always calibrated on the day of recording. The camera setup was not optimal in the sense of marker visibility because the equipment is usually used for gait analysis. VICON Workstation v4.6 was used, and the system recorded at 60Hz. The marker set can be seen in figure. 1b. Note that an additional marker was placed on the acromion (shoulder) as reference.

2.2 EMG electrode placement

Eight electrodes were used to distinguish the eight movements and table 1 list the electrode site placement. See Appendix B.3 for figures of the muscles and their names. Electrodes should be placed correctly and at the same place every time for each subject so that the classifier adapt to the correct signals.



1a: Anterior view

1b: Posterior view, with markers

Figure 1: Electrode site placement (Fougner, 2007)

Electrode site	Muscle(s)
1	Pronator teres
2	Supinator
3	Flexor digitorum superficialis/sublimis
4	Extensor digitorum
5	Flexor carpi radialis
6	Flexor carpi ulnaris
7	Extensor carpi radialis brevis & longus
8	Extensor carpi ulnaris

Table 1: Electrode site placement (Fougner, 2007)

2.3 Contents of data sets

The data set consist of EMG signals and VICON angle data. The VICON angle data was recorded at 60Hz, while the EMG signal was sampled at 1500Hz to be sure all relevant data was recorded[2].

The data is divided into 3 sets, where two (1 and 2) was recorded on the same day. The reason to record the sets on two different days was to check that the pattern recognition method would still be useful when new markers and electrodes have been placed and that the skin conditions could be a bit different. Table 2 show what data set combination was used.

Training	Validation	Testing
1	2	3

Table 2: Data set combination

Signal bandwidth According to Fougner (2007) it is found that the bandwidth of the wrist to be approximately 10-12Hz, containing $\approx 75\%$ of the signal. Fingers are slightly faster, and might have a bit larger bandwidth, but normal prostheses are not able to move that fast. Based on Nyquist theorem a sample frequency of 20Hz was then chosen for pattern recognition and prosthesis control signals. The features were calculated at periods of 1/20 second to get the correct frequency, making the resulting estimated angles 20Hz.

[2] Based on the Nyquist-theorem one would need at least 1kHz sampling frequency because EMG signals normally have a bandwidth of 500Hz.

2.3.1 Movements

Movements are illustrated in Fig. 2, all but finger flexion/extension movement are shown.

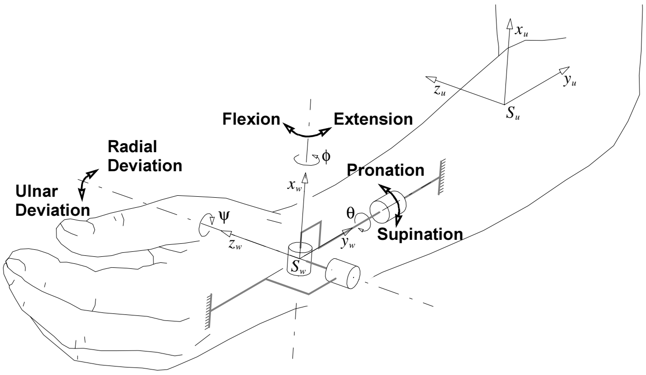


Figure 2: Clinical angles of the wrist (Stavdahl, 2002)

The data set contain both simple and combined movements of four different angles. These movements have been selected because the are common in simple movements in activities of daily living (ADL)[3]. Table 3 show the selected movements.

#S	Simple movements
S1	Finger flexion/extension
S2	Wrist flexion/extension
S2	Pronation/supination
S3	Radial/ulnar deviation

#C	Combined movements
C1	Finger flexion/extension and wrist flexion/extension
C2	Finger flexion/extension and pronation/supination
C3	Finger flexion/extension and radial/ulnar deviation
C4	Wrist flexion/extension and pronation/supination
C5	Wrist flexion/extension and radial/ulnar deviation
C6	Pronation/supination and radial/ulnar deviation

Table 3: Movements (Fougner, 2007)

[3] Activities of daily living includes regular tasks like slicing bread, eating, sweeping etc.

2.4 Test subjects

Three volunteers were used in the research of acquiring the EMG and VICON data. Subjects are held anonymous. To get a more generally valid result both genders were present, all had different age and body build. No known neuromuscular diseases (Fougner, 2007).

Signals were recorded from the non-dominant hand, because for most unilateral amputees the remaining hand will become dominant. Even though one subject was left-handed, the recordings were taken on the left arm.

2.5 Angle calculations

From the VICON data one needs to calculate the clinical angles that will be used as reference angles in the pattern recognition system. The VICON data only gives vectors of points for each marker in the setup. Based on the information in these vectors, new vectors defining the clinical angles can be computed.

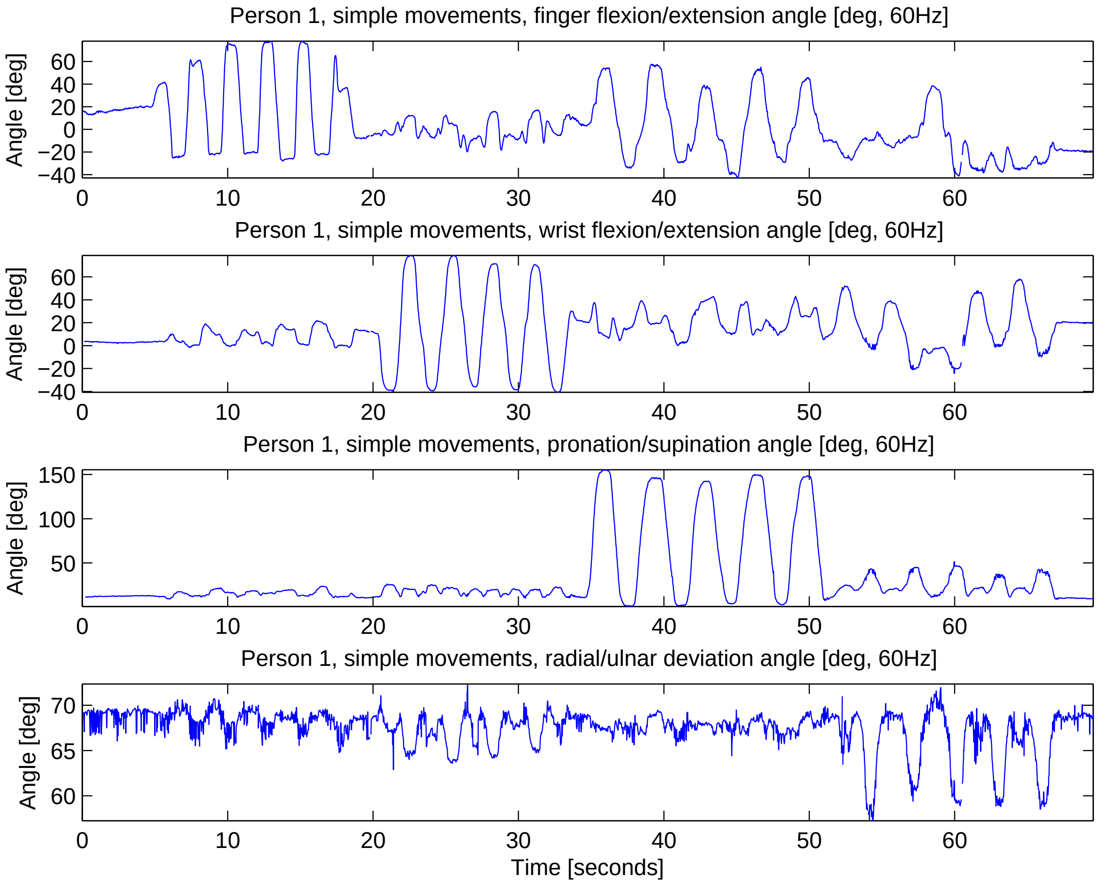


Figure 3: Angles, simple movements, person 1 (Fougner, 2007)

Figure 3 show a graphical view of the four different angles. It will be used as training sets for pattern recognition methods and as reference for pattern recognition on validation and test sets. Not all markers were visible all the time, and this results in broken graphs. These timeslots has been removed by Fougner (2007). To get a detailed description on how to calculate the vectors and angles, reader is adviced to look up Fougner (2007).

2.5.1 Range of motion

Table 4 show a simple relation between the four angles and their range of motion (ROM).

Angle	ROM [deg]
Finger flexion/extension	(-40,80)
Wrist flexion/extension	(-40,80)
Pronation/supination	(0,170)
Radial/ulnar deviation	(55,75)

Table 4: Range of motion for each angle

2.6 Pattern recognition methods

The objective of pattern recognition is to estimate the correct angle with a feature vector based on some known knowledge acquired through training of a classifier. Such a classifier could be in a representation of a discriminant function, neural network or some other type of function.

In Fougner (2007) amongst other a discriminant function and a neural network was implemented. The discriminant function and the neural network will also be used in his thesis.

The discriminant function is a simple Bayesian classifier but we use the function value directly instead of a threshold function. These estimated angles can be used directly as control output signal for a proportional control. The discriminant function is renamed to a mapping function because of this (Fougner, 2007).

2.6.1 Problem description

Given a data set containing both EMG signals and angle data, we want to minimize the error $e_j = \theta_j - \hat{\theta}_j$. The difference between the actually measured angles and the estimated one will be our error e_j. $\hat{\theta}_j$ is thought of as a control signal for a prosthesis. Minimizing the error is usually done with least-squares estimation.

2.6.2 Linear mapping function

The SVEN control system was based on a LDF (1) as a classifier , and this type of classifier will also be used in this paper.

$$g_j(X) = W_j{}^T X + \omega_{0j} \tag{1}$$

where

$$X = [x_1, .., x_i]^T \ , \ W_j = [\omega_{1j}, .., \omega_{ij}]^T \tag{2}$$

and where i is the electrode site number, j the movement number, x_i the feature vector taken from the EMG signal from electrode site i, w_{ij} is corresponding weighting factor for electrode site i, movement j and w_{0j} is a constant term.

By using least-squares estimation (3) we will try and find the best representation of W_j and ω_{0j}.

$$g_j(X) \ : \ X \rightarrow \hat{\theta_j} \underset{\hat{\theta_j}}{min}(\theta_j - \hat{\theta_j})^2 \tag{3}$$

$$\hat{\theta_j} = f(g_j(X)) \tag{4}$$

Using least square estimation one will find the best values of W_j and ω_{0j}. Further details can be found in (Fougner, 2007). Matlab implementation of the function `firstOrderEstimation.m` can be found in Appendix A.17.

2.6.3 Multilayer perceptron network

An artificial neural network (ANN) is a network made up of interconnecting artificial neurons. Such neurons mimic the properties of a biological neuron, and is often called nodes. It is used to solve artificial intelligence problems. A multilayer perceptron (MLP) network is a feedforward ANN model and maps input data to appropriate output data. It is commonly used as a pattern recognition method on bioelectric signals. A simple MLP network consist of three layers of nodes named input, hidden and output.

The input layer contains as many nodes as input signals, in our case 8. The hidden nodes represent most of the processing, and the result of the estimation depends on the size of this layer. The output layer has 4 nodes in our case, since we want to estimate 4 different angles. In Matlab, Neural network toolbox was used together with the function `tansig` based on formula (5). The MLP network was trained using least-squares estimation and back-propagation (Fougner, 2007; Bach, 2008).

The function `neuralNetwork.m` can be found in Appendix Appendix A.18.

$$tansig(x) = \frac{2}{(1 + e^{-2x}) - 1} \tag{5}$$

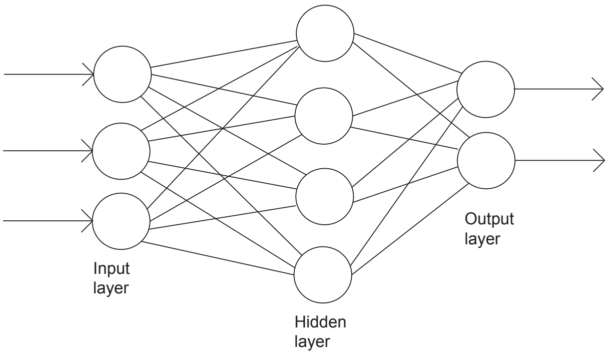


Figure 4: Example of MLP network with 3 inputs, 2 outputs and 4 nodes in the hidden layer(Fougner, 2007)

It is important to choose a training set that contains a large variety of movements, in order for the network to recognize different movements.

3 Theory

3.1 Feature

A feature is a measurable individual property from what is observed. Gathering discriminating identifications are important as input for algorithms in pattern recognition problems (Wikipedia, The Free Encyclopedia, 2009*f*).

3.2 Feature extraction

This section is based on Bach (2008). Research has been carried out within this field to detect and classify muscle and nerve activities specific to different motion patterns with respect to the hand. This will lead to a more useful hand prosthesis that naturally act as a real arm based on the signals sent from the brain or spinal cord. There are a number of signal features that is possible to extract from a myoelectric signal (MES), but only a few have been tested; Estimation of the amplitude, rate of change of the MES and time-domain features such as zero crossings or mean absolute value are examples of such features.

3.3 Feature selection

Feature selection may go under names such as feature reduction, feature subset reduction or feature weighting. If one has a classification or mapping problem, features are specific details that may help to separate classes or generate an estimation. Because of this it is desirable to find what features are irrelevant and may be eliminated, and how to combine the remaining features to get an optimal result (Avrim L. Blum and Pat Langley, 1997). This will reduce the dimensionality of a pattern recognition problem leading to possibly less complex problems to solve. By selecting or weighting features, it may be easier to find what kind of properties are relevant to a task and acquire a better knowledge of the collected data. Feature selection has been investigated for decades, and most researchers have concentrated the search around linear regression (Robi Polikar, 2006).

By removing irrelevant and redundant features, the performance of a selection model will increase. In some cases the model is quite robust against noise, but the computation time may drop considerably when this is removed.

The biggest problem is to find an optimal solution to a problem. It often requires an exhaustive search approach including all possible feature combinations. If there are many features, and the data set is long, it may be impractical. With supervised learning problem, realised with a MLP, this may be difficult because a method such as ANN starts with random initial conditions, and it is not certain that the computed result, represents the true possibility for that specific feature. In such a problem, a feature should be calculated multiple times and arranged accordingly.

A minimization of a pattern recognition problem requires a model that will specify the approach and measure for a dimensionality reduction. There are four approaches according to the literature for a feature domain reduction. They can be categorized as follows:

Embedded

Ranking

Wrapper

Filter

The embedded method may be a specific technique realized in the specific model. Embedded method has not been mentioned much in the literature, and will therefore not be investigated further here. Ranking is a statistical method that apply a form of score for each individual feature. It can be seen as a subtechnique for both the filter and the wrapper method (Eugeniusz Gatnar, 2006). The filter and wrapper methods are traditionally the methods discussed in the literature. The two methods differ in the way that the latter uses the classifier itself as a function to evaluate the performance, while the filter method filters out undesirable features based on a criterion like mean square error (MSE) ahead of classification.

3.3.1 Wrapper

A wrapper method uses the classification results directly to evaluate and select features. This approach guarantees good results for training data, but may not give very good results on a test data set. The reason for this is the tendency to overfit the training data. If MLP or support vector machines (SVM) are used as classifiers, this method will be computationally difficult because of the extensive training of such classifiers (Jun Yang and Yue-Peng Li, 2006). A wrapper method can be realized with an induction algorithm as a black box used to find an optimal set of features. Another technique may be achieved with a continuous optimizer. Any heuristic search algorithm can be used (D. Wettschereck and D. W. Aha, 1995). As an example; based upon the result from a classification of training data, features are excluded.

3.3.2 Filter

The filter method is independent from the classifier because it uses measures not dependent on the classifier. Such a measure can be for instance correlation or MSE (Jun Yang and Yue-Peng Li, 2006). The filter method uses a search algorithm to search through the space of possible features to find an optimal subset. The idea is to filter out unwanted features ahead of the induction algorithm, known as the classifier. This can be achieved with a separate process that evaluates features based on a simple condition: All features are weighted and the k best features should be combined in subsets and fed to a classifier (Avrim L. Blum and Pat Langley, 1997). A potential problem with this process is that it may give high relevance to highly correlated features and this may result in redundant features not being removed (Yijun Sun and Dapeng Wu, 2008).

According to Yijun Sun and Dapeng Wu (2008) filter methods often perform worse than wrapper because it uses a weak form of feature selection. Given a criterion function, feature selection is reduced to a search problem.

One can make assumptions to the evaluation function, by assuming monotonicity, the fact that increasing the subsets dimensionality can only increase the performance. But this assumption is not valid for many induction algorithms used today (Robi Polikar, 2006).

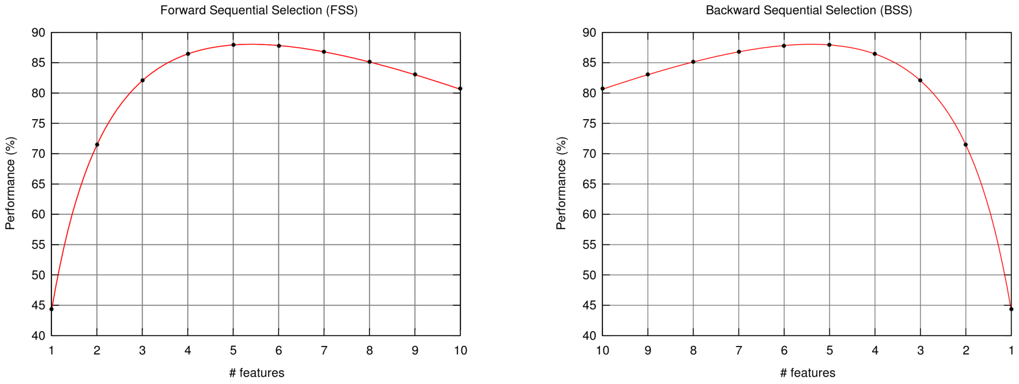
3.3.3 Search approaches

Search algorithms can be applied when it is desirable to find a path amongst data. If one is to develop a wrapper or filter method to reduce the complexity of a pattern recognition problem, an algorithm should be appliecd. Many search algorithms use a greedy algorithm that finds and optimal solution at each stage, in search for a global optimal solution. It is clear that in order to find a global optimal solution, or combination, all possible paths must be tried. This can be extremely time consuming and in some cases impossible.

According to D. Wettschereck and D. W. Aha (1995) there are three main categories of search algorithms: i) Exponential: branch or exhaustive (the running time is upper bounded by $O(2^d)$ where d indicates number of features/the dimension.), ii) Randomized: genetic and simulated annealing search, iii) Sequential: polynomial complexity, add or subtract features (Hill-climbing strategy) (the running time is upper bounded by $O(d^2)$).
One should take into account the amount of data and the time it may take to achieve a result after calculation. It is not always practically possible to try all possible combinations. When dealing with a problem indicating a large dimension, one would clearly not want an exponential approach. Today, many popular search approaches use greedy hill climbing[4]. In a feature reduction problem, it will iteratively evaluate a candidate subset of features, then modify the subset and evaluate if the new subset is an improvement of the old (Wikipedia, The Free Encyclopedia, 2009*g*).

[4] A greedy algorithm find a local optimum at each stage trying to find a global optimum. Hill-climbing is a greedy graph search algorithm, that will make a locally optimal choice in each stage, hoping for a global optimal solution.

Forward sequential selection (FSS) and backward sequential selection (BSS) are two heuristic search approaches. FSS starts with zero features and based upon a trial-and-error concept it will add a feature in the search for a subset. It can only guarantee a sub-optimal solution. BSS tries to eliminate feature by feature from a complete set of features. A typical approach is to rank the feature on some evaluation criterion, and then combine the best features in a FSS or BSS manner.



5a: Forward sequential selection

5b: Backward sequential selection

Figure 5: Possible realizations of sequential selection, forward and backward. A stopping criterion should be applied to stop the computation when a given performance is achieved.

Table 14 in Appendix B.1 show different algorithms that are common in the literature for feature reduction problems. Many of the algorithms use a form of sequential selection approach.

3.4 Diving into wavelets

In this text the theory behind wavelets will be concentrated around continuous wavelet theory, even though a discrete wavelet will be implemented.

There is not much difference in how the discrete or continuous wavelet work other than that implementation is usually achieved discrete.

The following will only give a brief technical and mathematical explanation of the concept of wavelet theory. For more thorough mathematical evidence, the reader is adviced to consult Stephan Mallat’s book *A wavelet tour of signal processing*.

This section is organized in three parts. THe first part introduces the concept behind *Fourier Transform*, *Convolution* and *Windowed Fourier Transform*.

The second part will give a brief information about *Frequency Filtering*, *Time-frequency localization* and the *Heisenberg problem* before diving into *Wavelets and its properties* and *types of Wavelets*. The third part will introduce the *Wavelet Transform* and *Wavelet Packet Transform*.

3.4.1 History and Concept

In 1807 Jean Baptiste Joseph Fourier (1768-1830), whilst trying to solve the heat equation, discovered that any periodic function can be represented as sum of simple oscillating functions. This was called Fourier series.

Due to the properties of sine and cosine it is possible to recover the amount of each wave in the sum by an integral. It is often desirable to use Euler's formula (6) to write Fourier series in terms of the basic waves. This complex valued Fourier coefficients contain both amplitude (size) and phase (angle) of the wave.

$$e^{2\pi i\theta} = cos2\pi\theta + isin2\pi\theta \tag{6}$$

This discovery had a profound impact in mathematical analysis, physics and engineering, but it took over one and a half century to understand the convergence of Fourier series and complete the theory of Fourier integrals. Wavelet was somehow first introduced by Haar in 1909. Throughout the 20th century many thought of a way to decompose or transform a signal into pieces, to be able to locate frequencies with respect to time. But it was first around 1980 that the use, understanding and theory around it really escalated (Dana Mackenzie, 2001; Mallat, Stéphane, 1999).

3.4.2 The Fourier Transform (FT)

The FT is a mathematical approach to convert a time-domain function (signal) to the frequency domain to identify frequencies present in a signal. Time aspect of the signal is then lost and one will not be able to reconstruct the signal.

A simple Fourier integral, or Fourier series, measures how much oscillations there are of a frequency ω in a signal f. It is a continuous function of ω (Wolfram Mathworld, 2009*b*). Indeed they have certain drawbacks. For instance, they do not operate efficiently when a signal is not periodically synchronized. Though, the FT allows the signal to have continuous variations. This transform analyses the frequency contents of a signal. The FT of $f \in L^2(\mathbb{R})$[5]:

[5] $L^2(\mathbb{R})$: Finite energy functions $\int |f(t)|^2 dt < +\infty$

$$\hat{f}(w) = \int_{-\infty}^{+\infty} f(t)e^{iwt}dt \tag{7}$$

Since sinusoidal waves are eigenvectors of the differentiation operator, the Fourier transform can therefore give indications on the regularity of a signal [6] (F. Chaplais, 1998).

One of the most important concepts of Fourier theory is convolution. A mathematical property of the FT makes it convenient to perform calculations by using convolution.

The concept of convolution Basically a convolution, or folding, is an integral that expresses the amount of overlap of one function g as it is shifted over another function f. It therefore "blends" one function with another. (Wolfram Mathworld, 2009*a*)

From the time domain point of view, any linear time-invariant (LTI) filter, or system, can be characterized entirely by a single function, an impulse response. In other words a LTI filter is equivalent to a convolution between the input and the filter's impulse response (Wikipedia, The Free Encyclopedia, 2009*i*). The convolution theorem is the most important property of the FT because it express the fact that sinusoidal waves $e^{i\omega t}$ are eigenvalues of convolution operators (F. Chaplais, 1998).

If a (oscillating) function is divided into small time segments created with a window function the FT would provide the frequency components in time. This is known as Short-time Fourier Transform (STFT) (or Windowed Fourier Transform (WFT)), but this would be a low resolution solution.

3.4.3 Windowed Fourier Transform (WFT/STFT)

In 1946 the physicist Gabor defined an elementary time-frequency localization known as atoms, to decompose a signal and therefore achieve better information about local frequencies in a signal. This approach is closely related to the way a human's ear is sensitive to sound (Mallat, Stéphane, 1999).

By multiplying with a function that is non-zero for only a short period of time, one can determine the frequency and phase of a local section of a signal. This window function is applied before performing the regular FT to obtain the frequencies in the region of the window.

[6] Regularity of a signal means it can be locally approximated by a polynomial. See Lipschitz regularity in literature for more information. The FT analyses the global regularity of a function, while Wavelet Transform (WT) makes it possible to analyse the pointwise regularity of a function (F. Chaplais, 1998).

$$\textbf{STFT}\{x(t)\} = X(\tau, \omega) = \int_{-\infty}^{+\infty} f(t)w(t-\tau)e^{iwt}dt \tag{8}$$

where $x(t)$ is the signal to be transformed and $w(t)$ is a window function like Hann or Gaussian centered around zero. The window function defines if there will be good frequency resolution or good time resolution. From this one will get a two-dimensional representation of a one-dimensional signal (Wikipedia, The Free Encyclopedia, 2009*j*). See Fig. 28 in Appendix B.4.

This approach is widely used today for time-frequency localization. In audio engineering one can visualize frequency representation in an audio sample and locate specific noise and its frequency to be able to remove it.

3.4.4 Frequency filtering

Time invariance means that if input $f(t)$ is delayed by τ, the output will also be delayed by τ.

$$g(t) = Lf(t) \Rightarrow g(t-\tau) = Lf(t-\tau) \tag{9}$$

The following is mainly based on and cited from (Mallat, Stéphane, 1999).

In FT the sinusoidal waves $e^{i\omega t}$ are eigenvectors of LTI convolution operators. Such an operator L is only specified by the eigenvalues $\hat{h}(\omega)$:

$$\forall \omega \in \mathbb{R}, \quad Le^{i\omega t} = \hat{h}(\omega)e^{i\omega t} \tag{10}$$

A signal f is defined as a sum of sinusoidal eigenvectors:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{f}(\omega)e^{i\omega t}d\omega \tag{11}$$

if f has finite energy, the FT gives the amplitude $\hat{f}(\omega)$ of each wave (frequency component) $e^{i\omega t}$. This integral measures the amount of oscillations for a frequency ω in f.

$$\hat{f}(\omega) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} f(t)e^{i\omega t}dt \tag{12}$$

Applying a LTI operator L to (11) and inserting the eigenvector expression (10) gives:

$$Lf(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{f}(\omega)\hat{h}(\omega)e^{i\omega t}d\omega \tag{13}$$

This may be written on a short form as a convolution:

$$g = f \star h \tag{14}$$

In (13) L amplifies each sinusoidal component $e^{i\omega t}$ of f by $\hat{h}(\omega)$. This is a convolution and is often called *frequency filtering* of f. $\hat{h}(\omega)$ acts as a transfer function of the filter. This is nice as long as we are not interested in *time-variant* information. Fourier coefficients are obtained in (12) by correlating f with $e^{i\omega t}$. Since $e^{i\omega t}$ has no compact support, in other words is locally defined, $\hat{f}(\omega)$ depends on the values $f(t)\,\forall t \in \mathbb{R}$. This results in a global "mix" of information and makes it difficult to analyse local parts of f from $\hat{f}$.

The fact that the sinusoidal waves ($e^{i\omega t}$) are eigenvalues of the convolution operator tells us what frequency components are present in the signal that was convoluted while $\hat{f}(\omega)$ defines the amount of a frequency.

3.4.5 Time-frequency localization

To clearly identify a signal event from a short time interval is thought of as a time localization and the ability to clearly identity signal components concentrated at particular frequencies are thought of as frequency localization. As an example one can see the FT as a function of a sum of sinusoidal waves. The waves are localized in the frequency, but not in time. The reason for this is that the waves are periodic, hence infinite of length. This means that the Fourier elements $b_\omega(t) = e^{i\omega t}$ has poor time localization abilities. Therefore, in order to represent the frequency behaviour of a signal in time, it should be analysed by functions that are localized excellent both in time and frequency. This time-frequency localization is unfortunately limited. One such limitation comes from the uncertainty theorem of Heisenberg (Amara Graps, 2004; Phil Schniter, 2005; F. Chaplais, 1998; Mallat, Stéphane, 1999).

3.4.6 The uncertainty principle of Heisenberg

This principle states that there has to be a balance between the time and frequency resolution. It tells that the energy spread of a function and its FT cannot be simultaneously arbitrarily small. The time-frequency plane may be sliced into small rectangles called atoms [7](F. Chaplais, 1998; Mallat, Stéphane, 1999).

[7] It is not possible to measure both position and velocity at the same time for an object, likewise with a signals frequency and its exact time occurrence. Cited from Dana Mackenzie (2001): "*In musical terms, the trade-off means that any signal with a short duration must have a complicated frequency spectrum made of a rich variety of sine waves, whereas any signal made from a simple combination of a few sine waves must have a complicated appearance in the time domain. Thus, we can't expect to reproduce the sound of a drum with an orchestra of tuning forks.*"

The following mathematical explanation is mainly cited from (Mallat, Stéphane, 1999).

Consider a family of time-frequency atoms $\{\phi_\gamma\}_{\gamma\in\Gamma}$ (γ may be a multi-index parameter). $\phi_\gamma \in \mathbf{L^2}(\mathbb{R})$ and $\|\phi_\gamma\| = 1$. The linear time- transform of $f \in \mathbf{L^2}(\mathbb{R})$ is defined by

$$Tf(\gamma) = \int_{-\infty}^{+\infty} f(t)\phi_\gamma^*(t)dt = \langle f, \phi_\gamma\rangle \tag{15}$$

$\langle f, \phi_\gamma\rangle$ contains a slice of information represented in the time-frequency plane (t, ω) by a region whose location and width depends on the time-frequency spread of ϕ_γ. Since:

$$\|\phi_\gamma\| = \int_{-\infty}^{+\infty} |\phi_\gamma(t)|^2 dt = 1 \tag{16}$$

We interpret $|\phi(t)|^2$ as a probability distribution centered at

$$u_\gamma = \int_{-\infty}^{+\infty} t|\phi_\gamma(t)|^2 dt = 1 \tag{17}$$

The spread around u_γ is measured by the variance

$$\sigma_t(\gamma) = \int_{-\infty}^{+\infty} (t - u_\gamma)^2|\phi_\gamma(t)|^2 dt \tag{18}$$

Since $\int_{-\infty}^{+\infty} |\hat{\phi}_\gamma(\omega)|^2 d\omega = 2\pi\|\phi_\gamma\|^2$ (see Plancherel formula in literature). The center frequency $\hat{\phi}_\gamma$ is defined by

$$\xi_\gamma = \frac{1}{2\pi}\int_{-\infty}^{+\infty} (\omega)|\phi_\gamma(\omega)|^2 d\omega \tag{19}$$

and the spread around ξ_γ is

$$\sigma_\omega^2(\gamma) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} (\omega - \xi_\gamma)^2|\hat{\phi}_\gamma(\omega)|^2 d\omega \tag{20}$$

The time-frequency resolution of ψ_γ is represented in the time-frequency plane (t, ω) by a Heisenberg box centered at (u_γ, ξ_γ). The width along time and frequency is $\sigma_t(\gamma)$ and $\sigma_\omega(\gamma)$ respectively. See Fig. 6.

The Heisenberg inequality states:

$$\sigma_t{}^2\sigma_\omega{}^2 \geq \frac{1}{4}. \tag{21}$$

where σ_t is the time variance of a function $f \in L^2(\mathbb{R})$, and σ_ω is the Fourier variance. $\sigma_t\sigma_\omega$ can be seen as a time-bandwidth product, and is invariant to time or frequency scaling.

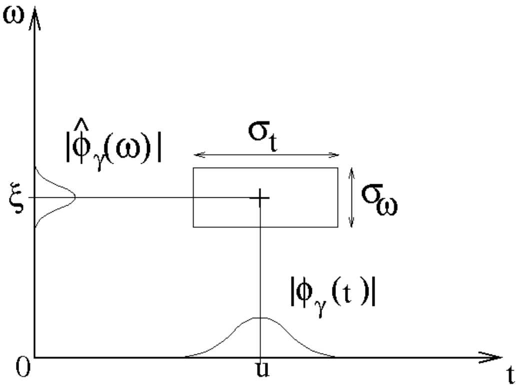


Figure 6: Heisenberg box representing an atom ϕ_γ (F. Chaplais, 1998)

The Heisenberg uncertainty theorem proves that the area of the rectangle is lower bounded by $1/2$. The time-frequency plane must be split into rectangles with area at least $1/2$ in order to create atoms, because no function can be perfectly well concentrated around a single point. From this it is impossible not to have a trade-off between time and frequency resolution (Phil Schniter, 2005).

From this, waveforms [8] can only be scaled and positioned by a ratio limited by the Heisenberg uncertainty. Atoms are time windows defined by a translation in time and frequency. An atom will therefore contain the total energy of a function in the neighbourhood of u over an interval with a size σ_t, and the FT will give the total energy localized near the frequency ω over an interval with a size σ_ω In a time-frequency plane (t, ω) the energy of these windows are seen as time-frequency boxes (Also known as Heisenberg rectangles).

3.4.7 Introduction to wavelets

A wavelet is a mathematical function best thought of as an extension to the FT. While in Fourier Analysis one tries to fit sines and cosines to a signal to generate a set of coefficients, in Wavelet Analysis on the other hand, one tries to fit a basis function called a mother wavelet to a signal. This has a different property in that a wavelet has compact support[9], while sines and cosines have not.

With wavelets, different frequency components and time slots can be scaled giving better insight into the harmonics and base frequencies of a signal. One important aspect of the wavelet is that is may accurately deconstruct and recon-

[8] Waveform: Shape and form of a signal.

[9] Compact support is a property that limits the range of integration because a function $\psi(x) = 0$ if $|x| > M$ for some M.

struct finite, non-periodic and/or non-stationary signals (Wikipedia, The Free Encyclopedia, 2009*k*).

Wavelets have had a profound impact in many engineering and physics applications. Amongst others: seismic geophysics, optics, quantum mechanics, multifractal analysis, ECG analysis and signal and image processing. Wavelets have been successfully applied to digital signal processing and digital image processing. Especially in compression, resizing, edge detection and texture analysis of images and video. JPEG-2000, an image compression format uses wavelet. Both the video formats REDCODE RAW and Dirac, which is an open source contribution from BBC, also employs wavelet compression instead of the usual discrete cosine transforms(DCT)[10] (Wikipedia, The Free Encyclopedia, 2009*k*).

In this paper we will look into two types of wavelet properties, the Wavelet Transform(WT) and the Wavelet Packet Transform(WPT).

3.4.8 Wavelets

The following is based on Mallat, Stéphane (1999).

To be able to analyse a signal with many different components, it is necessary to use time-frequency atoms with different time supports. Time-frequency atoms are waveforms that are well present in a signal.

A wavelet transform decomposes signals over scaled and translated wavelets. In addition, a wavelet is a function $\psi \in L^2(\mathbb{R})$ with a zero average:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{22}$$

The function is normalized $||\psi|| = 1$, and centered in the neighbourhood of $t = 0$. This function, ψ, is called the mother wavelet. A family of waveforms (23) (child wavelets) or atoms, are obtained by dilating ψ by s and positioned by u.

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-u}{s}) \tag{23}$$

The wavelet transform of f with a scale s at time u is found by correlating f with a wavelet atom [11]:

$$Wf(u,s) = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{s}}\psi^*(\frac{t-u}{s})dt \tag{24}$$

This again, can be written as a convolution product:

[10] Discrete cosine transform is a Fourier related transform, expressing a signal with a sum of cosines. It is common in audio and picture compression like MP3 and JPEG.

[11] $\langle f, \psi_{u,s} \rangle$ is called an inner product.

$$f \star \bar{\psi}_s(u) \tag{25}$$

where

$$\bar{\psi}_s(t) = \frac{1}{\sqrt{s}}\psi^*(\frac{-t}{s}) \tag{26}$$

These atoms remains normalized $||\psi_{u,s}|| = 1$.

A large s correlates with low-frequency components of a signal, while small s correlates with high-frequency components. The factor $(\frac{1}{\sqrt{s}})$ is used to preserve the energy. This is called the *continuous wavelet transform* (CWT) (D.K. Kumar,N.D. Pah & A. Bradley, 2003).

In a linear time-frequency transform the signal and such waveforms are correlated.

Scaling function The wavelet is defined by its mother wavelet and it can be seen as a bandpass filter. By scaling the wavelet the bandwidth is halved as seen in Fig. 7 and the daughter/child wavelets are created. This would require an infinite number of levels in order to cover the whole spectrum. A father wavelet (scaling function) called ϕ, is an auxiliary function used to avoid this numerical complexity. The scaling function basically filters the lowest level of the transform to ensure that the whole spectrum is covered (C. Valens, 2004).

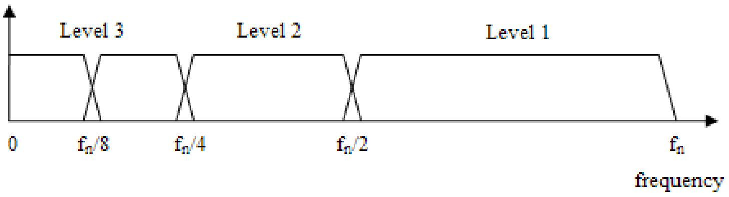


Figure 7: A mother wavelet scaled multiple times. The lowest level of the frequency spectrum should not be covered by a bandpass filter but instead a low-pass filter (scaling function) as indicated in this figure.

Orthogonal wavelets When the associated wavelet transform is orthogonal, the wavelet is called an orthogonal wavelet.

A bi-orthogonal wavelet is a wavelet where the associated wavelet transform is invertible but not necessarily orthogonal. Designing bi-orthogonal wavelets allows more degrees of freedoms than orthogonal wavelets. One additional degree of freedom is the possibility to construct symmetric wavelet functions Mallat, Stéphane (1999).

While bi-orthogonal wavelets and scaling functions are characterized by perfect reconstruction filter banks, orthogonal wavelets and scaling functions are characterized by a pair of conjugate mirror filters (F. Chaplais, 1998).

3.4.9 Multiresolution Analysis (MRA)

The concept of MRA is to approximate a signal at finer and finer resolutions. A MRA compute the approximation of signals at various resolutions using orthogonal projections.(F. Chaplais, 1998)

3.4.10 Vanishing moments

A vanishing moment limits a wavelets ability to represent information in a signal. The order of a wavelet often describes the number of vanishing points. Higher order wavelet transforms usually result in better signal approximations because the scaling function can represent more complex signals accurately (F. Chaplais, 1998; F. Qiao & R. Milam, 2005; Jani Huhtanen, 2005).

3.4.11 Conjugate Mirror Filters

The approximation of a function at a resolution 2^{-j} is defined as an orthogonal projection on a space $\mathbf{V}_j \subset \mathbf{L^2}(\mathbb{R})$. A multiresolution approximation is characterized by a scaling function ϕ that generate an orthogonal basis of each space $\mathbf{V}_j$. A scaling function is specified by a discrete filter called a *conjugate mirror filter.* This is important in discrete signal processing, and it makes it possible to decompose discrete signals in separate frequency bands with filter banks. See 3.4.14.

3.4.12 Types of wavelets

Wavelets can be divided into continuous and discrete wavelets, where continuous can be further divided into classes of real and complex valued. In the following each discrete wavelet will be briefly explained.

(a) Discrete wavelets

Type
Coiflet
Daubechies
Haar
Symmlet

(b) Continuous Real and complex wavelets

Type (Real)	Type (Complex)
Beta	Mexican Hat
Hermitian	Morlet
Mexican Hat	Modified Morlet
Shannon	Shannon

Table 5: Different types of wavelets

Haar The simplest form of a wavelet proposed in 1909. It can be defined by the scaling filter ϕ and is obtained with a multiresolution of piecewise constant functions. It is a compact, orthogonal and symmetric wavelet but it is discontinuous. Daubechies wavelet with order $p = 1$ is the same (Bruce Hawkins, 1999; Mallat, Stéphane, 1999).

Daubechies A family of orthogonal wavelets defining discrete Wavelet Transform (DWT). It is a continuous, compact, orthogonal and asymmetric wavelet. Further, it is characterized by a maximal number of vanishing moments for a given support. The wavelet is not defined by a scaling and a wavelet function, but it is usually computed with finite impulse response conjugate mirror filters. However, for each wavelet type there is a scaling function which generates an orthogonal multiresolution analysis. Daubechies have $N/2-1$ vanishing moments, and is one of the most used wavelets (Wikipedia, The Free Encyclopedia, 2009*d*; Bruce Hawkins, 1999; Mallat, Stéphane, 1999).

Coiflet Coiflet is a discrete wavelet designed to be more symmetric than Daubechies. It is constructed using a scaling function and a wavelet function. The scaling function has $N/3 - 1$ vanishing moments while the wavelet function has $N/3$ tf(Wikipedia, The Free Encyclopedia, 2009*b*; Mallat, Stéphane, 1999).

Symmlets Symmlets are compact, orthogonal and continuous wavelets, but they are only nearly symmetric. The father wavelet, ϕ, is used only for the largest scale decompositions while the mother wavelet, ψ, is used for all the finer scales (Bruce Hawkins, 1999). Complex conjugate mirror filters with compact support and linear phase can be constructed, but produce complex wavelet coefficients. Both coefficients are redundant when the signal is real (Wikipedia, The Free Encyclopedia, 2009*k*; Bruce Hawkins, 1999; Mallat, Stéphane, 1999).

3.4.13 Wavelet frame

The following is mainly cited from Mallat, Stéphane (1999):

Frames are a stable, possibly redundant, representation of signals (F. Chaplais, 1998). It is a family of vectors $\{\phi_n\}_{n\in\Gamma}$ that characterizes any signal f from its inner products $\{\langle f, \phi_n\rangle\}_{n\in\Gamma}$.

Consider the real continuous wavelet transform:

$$Wf(u,s) = \langle f, \psi_{u,s}\rangle \tag{27}$$

To construct a wavelet frame one needs to cover the time-frequency plane with the Heisenberg boxes from the corresponding discrete wavelet family. The energy in time of a wavelet $\psi_{u,s}$ is centered at u over a domain proportional to s. Dealing only with positive frequencies, the Fourier transform $\hat{\psi}_{u,s}$ has a support centered at a frequency η/s with a spread proportional to $1/s$. To get a full cover we sample s along an exponential sequence $\{a^j\}_{j\in\mathbb{Z}}$, with a sufficiently small dilation step $a > 1$. The time translation u is sampled uniformly at intervals proportional to the scale a^j. See Fig. 8.

$$\psi_{j,n}(t) = \frac{1}{\sqrt{a^j}}\psi(\frac{t - nu_0a^j}{a^j}) \tag{28}$$

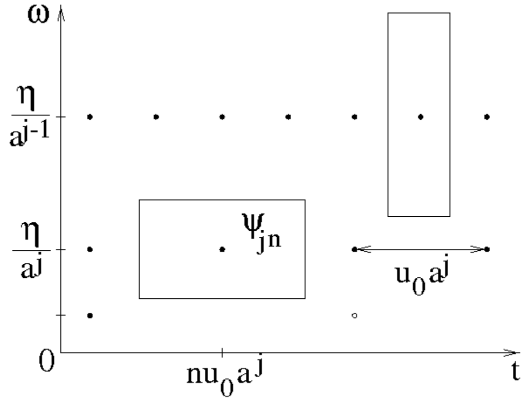


Figure 8: Heisenberg box of a wavelet $\psi_{j,n}$ scaled by $s = a^j$ (F. Chaplais, 1998)

See Appendix B.4 for more detailed figures of Heisenberg boxes and time-frequency tiling.

3.4.14 Filter banks

Because a mother wavelet can be realized as a bandpass filter and the father wavelet/scaling function as a low-pass filter, they can together form what is known as *filter banks*. Filter banks are arrays of filters that separates a signal into components as single frequency sub-band of the original signal. Filter banks are often constructed so the sub-bands can be recombined to recover the original signal. This filter process is thought to isolate different, or often specific frequency components of a signal. (Wikipedia, The Free Encyclopedia, 2009*h*; C. Valens, 2004).

Consider Fig. 10:

$x[n]$ is the signal we want to investigate. $g[n]$ is a low pass filter with an impulse response g corresponding to a scaling function ϕ. $h[n]$ is a band-/high pass filter with an impulse response h corresponding to a wavelet function ψ.

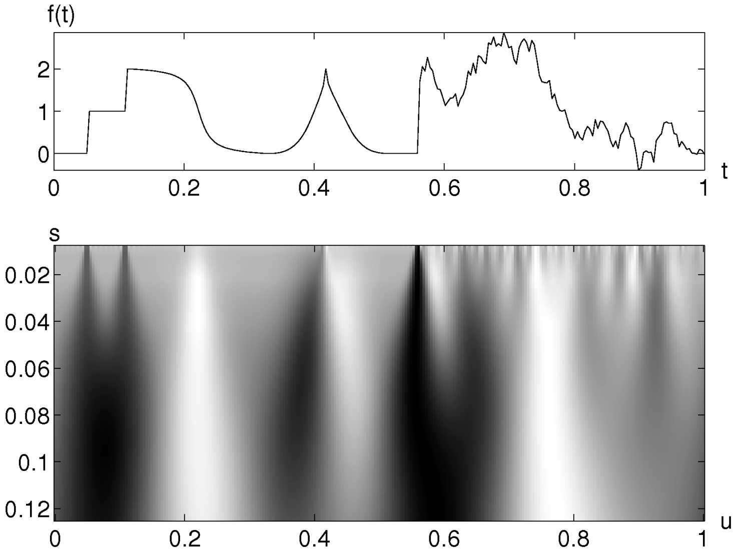


Figure 9: WT $Wf(u,s)$ calculated with $\psi = -\theta'$ where θ is a Gaussian for the signal above. The position parameter u and the scale s vary respectively along the horizontal and vertical axes. Black, grey and white points correspond respectively to positive, zero and negative wavelet coefficients. Singularities create large amplitude coefficients in their cone of influence F. Chaplais (1998).

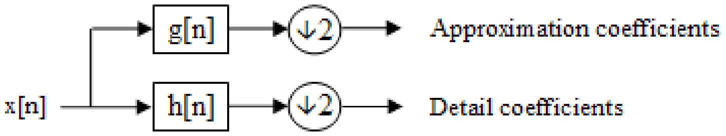
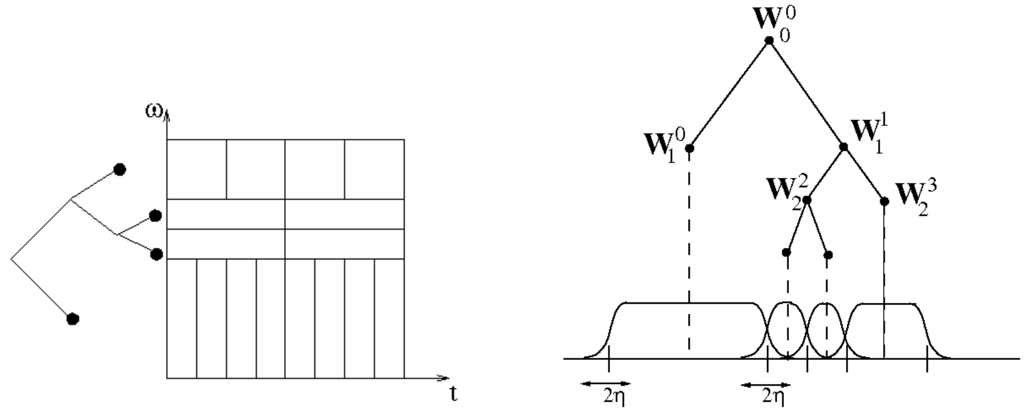


Figure 10: Detail and approximation coefficients from a WT realized as a filter bank (Wikipedia, The Free Encyclopedia, 2009*e*).

Because half the frequency range has been removed, the filter outputs are downsampled. Each decomposition halves the time resolution, but the frequency resolution has doubled. (As can be seen as an example in figure 11b) This is because each filter output has half the frequency band but the filter output characterises only half the signal.(Wikipedia, The Free Encyclopedia, 2009*e*)

The filter and the input signal is convoluted using a lifting-scheme[12].

[12] A lifting-scheme is a technique used to improve wavelet properties. A lifting is an elementary modification of perfect reconstruction filters. The lifting-scheme of Sweldens does not rely on Fourier transform. Because of this it can construct wavelets over non-translation invariant domains like a surface.(F. Chaplais, 1998)



11a: Forming Heisenberg boxes

11b: Wavelet packet tree

Figure 11: On the left, a wavelet packet tree that is represented with Heisenberg boxes. To the right a wavelet packet binary tree, where $\mathbf{W}_\mathbf{j}^\mathbf{p}$ denotes the pth subspace occurring at level j (F. Chaplais, 1998).

3.4.15 Wavelet Transform

With Fourier analysis we get a single coefficient for each sine and cosine. But because the wavelet has compact support we get a series of coefficients that vary with time, and it will accommodate local changes in a signal.

As a short non-technical way of describing the way a wavelet decomposes a signal, consider figure 12 and the following description:

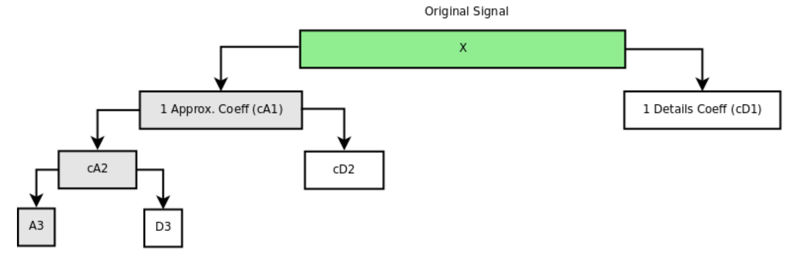


Figure 12: Decomposition of a signal x, resulting in approximation and details coefficients.

A mother and a father wavelet (filter bank) is applied to the original signal. From this, two coefficients evolves. A first detail for the high frequencies and a first approximation for the lower frequencies as a remainder. Next the time-scale is doubled and the wavelets (becoming child wavelets in this case a new filter bank) are dilated and fitted to the first approximation. This then produces a second detail and a second approximation. This process is repeated until the

number of desired levels is reached.
The details corresponds to an average time-scale that is doubled at each level. They are completely independent from each other, in other words orthogonal (Derek Goring, 2006). A realization is seen in figure 13.

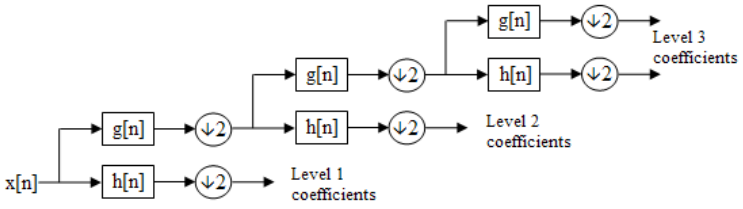


Figure 13: Filter banks decompose a signal into detail and approximation coefficients (F. Chaplais, 1998)

3.4.16 Wavelet Packet Transform

Wavelet packet were introduced by Coifman, Meyer and Wickerhauser. They wanted to generalize the link between MRA and wavelets.

It is a form to re-express a function in terms of wavelet basis $\{\psi_{jk}(f)\}$. This amounts to decomposing the function space L^2 into a direct sum of orthogonal subspaces $\{W_j\}$ Wolfram Mathworld (2009*c*). See Fig. 11b and Fig. 14 (Recall V_j from 3.4.11, in this figure $j = 3$).

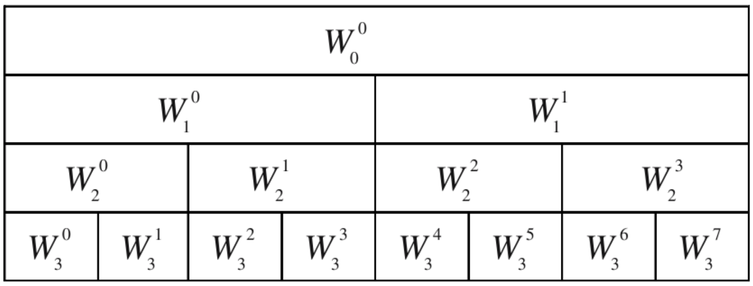

<table>
<tr><td colspan="8">W_0^0</td></tr>
<tr><td colspan="4">W_1^0</td><td colspan="4">W_1^1</td></tr>
<tr><td colspan="2">W_2^0</td><td colspan="2">W_2^1</td><td colspan="2">W_2^2</td><td colspan="2">W_2^3</td></tr>
<tr><td>W_3^0</td><td>W_3^1</td><td>W_3^2</td><td>W_3^3</td><td>W_3^4</td><td>W_3^5</td><td>W_3^6</td><td>W_3^7</td></tr>
</table>

Figure 14: Tree structure of wavelet packet decomposition. $\mathbf{W_j^p}$ denotes the pth subspace occurring at level j G. Wang, Z. Wang, W. Chen and J. Zhuang (2006)

The wavelet packet decomposition can be achieved by sending the signal through multiple filters. This leads to that both the detail and approximation coefficients are decomposed, in oppose to the WT where only the approximation coefficients are further filtered through a filter bank. While WT produces $n + 1$ sets of coefficients, the WPT produces 2^n sets of coefficients, but there are no redundancy because of a downsampling process (Wikipedia, The Free Encyclopedia, 2009*l*). A WPT can be seen as a tree (See Fig.14). The root is the original signal. The next branch of the tree is a single step of a regular WT, and so on. At the bottom one will get a number of coefficients which are called the packets. The height of the tree is $n + 1$, where n is number of levels to decompose the signal. An example is seen in figure 15.

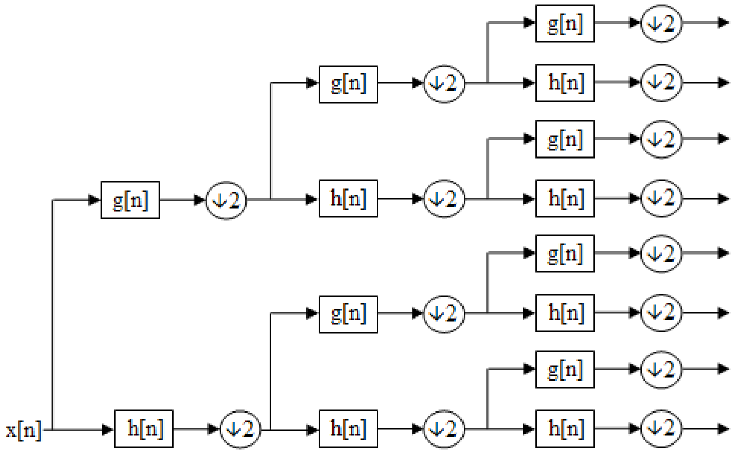


Figure 15: Decomposition of Detail and Approximation coefficients from a wavelet packet transform (Wikipedia, The Free Encyclopedia, 2009*e*)

Today, WFT/STFT is still widely used as a time-frequency representation, but WT and WPT may often give better and more precise information. However, this comes at a cost because they are more computationally expensive.

The reader is adviced to look up Seeley, Stephens, Tate (2007); J. V. Basmajian & C. J. De Luca (1985); Muzumdar (2004) if interested in getting an in-depth knowledge about the nervous system, muscles, electromyography and signal processing. The forthcoming will only scrape the surface in order to get an understanding of the information we want to acquire and analyse in order to develop a proportional prosthesis.

3.5 The nervous system

The nervous system consists of the brain and spinal cord (CNS) and nerves and ganglias (PNS) This system sense, generate and transmit electrical impulses that resolves reactions in the body. Information is sent through this network from the brain or the spinal cord about a motion, and response based on sensor units in the body may regulate the action performed.

The information that is picked up in the muscles is of a key issue when it is desirable to connect computerized equipment onto human extremities. Such equipment can be thought of as a prosthesis. (Bach, 2008)

3.6 Muscles

From the spinal cord nerve fibres connect to muscles to give and receive information regarding contraction in the muscle. A single nerve cell connect to different muscle fibres forming motor units. One nerve cell can connect from 10 to up to 1000 fibres depending on the muscle, but a muscle fibre is connected to only one nerve. As an example, to move a limb, an impulse travels to the specific motor units that must be triggered to generate a desired motion. More motor units will be recruited, if necessary, based on information from sensor receptors in the muscle and joints, and the activity of the active motor units. (Bach, 2008)

3.7 Electromyography

This section is cited from Bach (2008) which is based on (Seeley, Stephens, Tate, 2007; J. V. Basmajian & C. J. De Luca, 1985; Muzumdar, 2004; Todd Farrell, Richard F. ff. Weir, 2008). Electromyography (EMG) is a technique used to evaluate and record the activation signal of muscles by using needle electrodes in order to find muscle weakness caused by neurological disorders. Such technique is also used for prosthesis control, but the signal is then recorded using skin electrodes, and signals from healthy muscles are used. This is possible because cells naturally contain an electrochemical potential.

A summation of the activity (action potentials) from all the fibres in a motor unit is called a motor unit action potential (MUAP).

Adding up all the active motor neurons makes up the externally observed signal seen on an Electromyograph. These electrical impulses is what forms a myoelectrical signal (MES).

3.8 Signal processing

This section is shortened and cited from Bach (2008) which is based on (J. V. Basmajian & C. J. De Luca, 1985; Muzumdar, 2004; Todd Farrell, Richard F. ff. Weir, 2008). A MES have frequencies ranging between 1-500Hz. The voltage ranges from approximately 10μV to 1mV. The signal is usually sampled at a frequency of 1000-1500Hz to be sure that all information is acquired. This is based on the Nyquist theorem [13]. This MES can be recorded using electrodes

[13] Nyquist theorem states that one need to sample a signal at least twice the highest frequency occurring in the signal.

placed on the skin surface near a muscle. Three electrodes are usually needed where two are placed so there will be a voltage difference between them when a MES propagates, and the other one is placed in a neutral area to sense ground [14].

It is difficult to record these signals because they are weak and associated with noise.

3.9 Prosthesis control

This section is shortened and cited from Bach (2008) which is based on (Muzumdar, 2004; Todd Farrell, Richard F. ff. Weir, 2008). Earlier prosthesis were primarily a simple equipment with few (mostly one) degrees of freedom (DoF) to help the user in life, such as a hook, but it also acted as a purely cosmetic enhancement.

It is desirable to create upper limb prosthesis that act like the phantom arm but one of the main problems in making this possible is to be able to create a correct mapping of MES while getting consistently good and trustworthy signals from the electrodes. Muscles used to actuate wrist and fingers are mostly located in the forearm. In cases where some parts of the forearm remain intact, it is possible to control an advanced prosthesis. Reading the signals of the agonist and antagonist muscles enables the possibility to map signals to the desired movement. However, there are issues related to control thumbs correctly as many of the muscles to control this limb are located in the palm of the hand.

[14] We want to sense the ground on the body to remove the common mode noise made by electrical surroundings. Using a differential amplifier one can remove this common mode signal that is mixed up with the MES signal on the other two electrodes, and thereby only retrieve the voltage difference between them.

4 Aim and objectives

Aim This study will try to identify signal features that are beneficial in a proportional control of a multi-function upper limb prosthesis. The intent is to identify a set of signal features that could be implemented in a practical proportional control system to enhance the movement functions of the prosthesis such that it more closely mimic the movements of a normal upper limb.

Objectives The study will attempt to identify a subset of features that contain useful information in order to estimate angles, and evaluate them based on measured angles. The feature sets will be identified and tested by

1. Implementing algorithms based on wavelets in order to identify those signal features that would be useful for the above purpose
2. Combining the features with features found in an earlier study to optimise (search) for a suitable feature subset
3. Testing for statistical significance

The algorithms are explained in section 5 and implementation is made in Matlab. A short comparison of the newly added features against the previously acquired ones will be discussed to gain knowledge of important aspects like individual features.

Two pattern recognition methods will be used to evaluate the features (see section 6). 10 nodes will be used for the neural networks (NN) in this thesis because it gave the best results in earlier research by Fougner (2007) and Rahmanpour (2008).

5 Overview and features

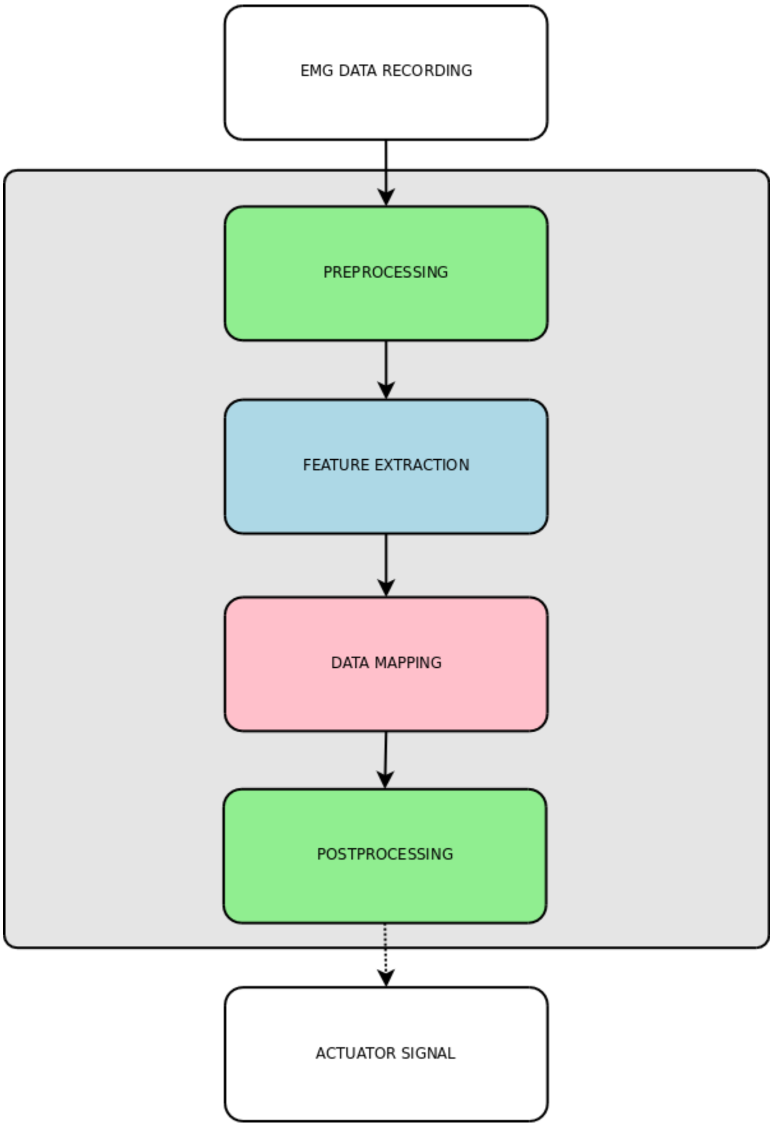


Figure 16: A simple flow diagram showing key elements of a prosthesis control system from EMG acquisition to actuator signal. The four elements inside the rectangle, must be found and dealt with before a prosthesis control may be acquired.

5.1 Building a system

The muscle activity that is recorded by a myoelectrode should be handled with some preprocessing techniques before the chosen features are extracted from the EMG signal. After this, the feature data will be weighted in a mapping function before running through some post-processing, enabling the estimated signal ready to determine the action of the actuator in a prosthesis. Four key elements in figure 16 illustrates how such a system may be achieved.

Preprocessing The preprocessing technique should filter the EMG signal to remove noise without loosing valuable information. This is achieved i.a. by a high-pass filter removing zero frequency.

Feature extraction Extracting features that makes up a perfect subset that best inform a mapping function about a desired motion from a user. Further, it is a good idea to downsample the signal to an appropriate size.

Data mapping This element is made up of possibly a NN that is trained on some data. This must be found together with a subset that successfully achieve a good pattern recognition infront of an implementation in a prosthesis control.

Post-processing This final element is quite important. Even though a NN may filter a signal, it is not necessarily a perfect signal for a prosthesis control. Filtering the signal and making it smooth will be much more useful in constructing an actuator control signal.

In our case, we have previously recorded data sets that enables us to investigate what features and mapping function to use in a system like this. Most of the pre-processing is allready taken care of, but it may be refined. We need to add new features and try to find a suitable subset that will contain so much information that a mapping function will successfully create an estimate of the motion the user (in our case the test subject) attempts to perform. In the search for this, the optimal mapping function will be found.

The mapping function must be trained enough (given it is a NN) so it will give an accurate result of the features that are combined in a subset. If it does not, a feature combination could possibly perform poorly even though it may be the best subset of all. To evaluate the result of an estimation a good and reliable measure of the performance must be at hand. To be able to show how well a combination perform, it is necessary to carry out a form of post-processing of the estimated signal, removing any potential noise. This however, will often have a downside in that it will make the performance of a controller slow but accurate.

5.2 EMG signal features

The following information is based on (Boostani, R. and Moradi, M. H., 2003; Fougner, 2007)

Features can be divided into different categories regarding what information they may obtain. Qualitative features are indicative only of relative sizes or magnitudes, rather than their numerical values which quantitative features will give.

Below follows a short explanation of each feature that is used in this thesis.

5.2.1 Qualitative EMG signal features

Averaged absolute value (AAV) This feature is the most common EMG feature today used in prosthesis control. Usually it is calculated on a filtered EMG signal. In this paper, AAC will be computed twice, directly on the unfiltered signal and on a filtered (processed) signal (Fougner, 2007) .

$$AAV = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{29}$$

x_i is the ith sample. N is the number of samples in each segment.

5.2.2 Quantitative EMG signal features

Average amplitude change (AAC) Mean value between the difference between two consecutive samples of a signal (Fougner, 2007).

$$AAC = \frac{1}{N}\sum_{i=1}^{N} |x_{i+1} - x_i| \tag{30}$$

x_i is the ith sample. N is the number of samples in each segment.

The Matlab function `EMGaac.m` is found in Appendix A.1.

Auto-regressive coefficients (AR) Signal samples are estimated by linear combination of their previous samples. EMG spectrum changes with muscle contraction state, results in a change in AR coefficients. Order of the model has been defined by various experimental and theoretical approaches. Model of order P=4 has been shown by Graupe and Cine (Boostani, R. and Moradi, M. H., 2003) to be suitable for EMG signals. Levinson-Durbin algorithm was used.(Rice University, 2000)

The Matlab function `EMGar.m` is found in Appendix A.3, and the Levinson function `myLevinson.m` is found in Appendix A.15

Cepstrum/cepstral coefficients (CC) Cepstrum is defined to be the FT of the decibel spectrum as if it were a signal. It can be seen as the rate of change in the different spectrum bands. Quefrency is the independent variable of a cepstral graph. It is a measure of time, but not in the sense of a signal in the time domain (Wikipedia, The Free Encyclopedia, 2008). These coefficients are a very powerful tool for speech applications. It has been shown that it is a feature very suitable for motion classification. The built-in Matlab function (rceps) has been used.

The Matlab function `EMGcc.m` is found in Appendix A.4.

Energy of wavelet coefficients (EWC) The energy of a wavelet coefficient tells us how much of the signal we have kept after a WT. Using WT one can extract information (certain frequencies in time) from a signal-band that is suitable for a special purpose.

Boostani & Moradi decomposed the EMG signal into nine scales with wavelet transform. The signal energy was found as components of the feature vector. A bi-orthogonal mother wavelet was used because it has similarities with the action potential (Fougner, 2007).

The Matlab function `EMGewc.m` is found in Appendix A.6.

Energy of wavelet packet coefficients (EWPC) While in WT one get a fixed grid on the information the coefficients contain from a signal. With the Packet Transform, this grid can be altered as desired in order to focus on more specific frequencies in a signal. Using WPT instead, Boostani & Moradi obtained the best result. It is more complex, and calculation time is longer.

The Matlab function `EMGewpc.m` is found in Appendix A.7.

A more thorough explanation of the implementation of the wavelets will be given in section 6.2.

Histogram (HIST) This feature determines the number of signal samples in different amplitude levels in a time segment. Introduced by Zardoshti. Extension of ZC and WAMP, by comparing a single threshold to the EMG signal. The built-in Matlab function was used, with number of levels equal to Boostani & Moradis work, $n = 9$. Though we do not know how these 9 levels were selected.

The Matlab function `EMGhist.m` is found in Appendix A.5.

Myopulse percentage rate (MYOP) The myopulse output is defined as 1 (one) when the absolute value of a signal is above a threshold. Zero otherwise. The myopulse percentage is then defined to be the average value of the myopulse output.(Fougner, 2007).

$$MYOP = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{31}$$

where $f(x_i)$ is defined in 37.

x_i is the ith sample. N is the number of samples in each segment. Threshold was set to 0.15 in this study.

The Matlab function `EMGmyop.m` is found in Appendix A.8.

Number of turns (NT) The number of times the slope of a signal changes sign. Closely related to ZC (Fougner, 2007). The Matlab function `EMGnt.m` is found in Appendix A.9.

$$NT = \sum_{i=1}^{N} sgn(-(x_{i+1} - x_i)(x_{i+2} - x_{i+1})) \tag{32}$$

where the $sgn(x)$ is defined in 40.

x_i is the ith sample. N is the number of samples in each segment.

The Matlab function `EMGnt.m` is found in Appendix A.9.

Processed EMG A processed EMG signal feature was also used. This feature is the averaged amplitude value (AAV) applied on a filtered EMG signal. It was calculated and filtered in Fougner (2007). The EMG signal was first high-pass filtered to get the average to zero (cut-off frequency of 1Hz (33)), then rectified. It was then low-pass filtered (33) with cut-off frequency of 10Hz since it made a smooth signal and good representation of the amplitude. A special non-linear dead-zone filter was used to assure that the integrator's output only varied if the input was larger than the width of the dead-zone. More detailed explanation is given in (Fougner, 2007).

$$H_{hp}(s) = \frac{\frac{1}{2\pi}s}{\frac{1}{2\pi}s + 1} = \frac{s}{s + 2\pi} \tag{33}$$

$$H_{lp}(s) = \frac{1}{\frac{1}{2\pi 10}s + 1} = \frac{20}{s + 20\pi} \tag{34}$$

Variance (VAR) The variance is a measure of the signal power. It is one of the most used EMG signal feature today. With this function one can use thresholds to control selected functions on a prosthesis (Boostani, R. and Moradi, M. H., 2003).

$$VAR = \frac{1}{N-1}\sum_{i=1}^{N} {x_i}^2 \tag{35}$$

x_i is the ith sample. N is the number of samples in each segment.

The Matlab function `EMGvar.m` is found in Appendix A.10.

Wilson amplitude (WAMP) A count for how many times a change of amplitude between two consecutive samples is exceeding a threshold value, typically 50μV, occur. This feature is an indicator of firing MUAP's, that is an indicator of the contraction level (Boostani, R. and Moradi, M. H., 2003).

$$WAMP = \sum_{i=1}^{N} f(|x_i - x_{i-1}|) \tag{36}$$

where

$$f(x) = \begin{cases} 1 & \text{if x > threshold,} \\ 0 & \text{otherwise.} \end{cases} \tag{37}$$

x_i is the ith sample. N is the number of samples in each segment.

The Matlab function `EMGwamp.m` is found in Appendix A.11.

Wavelength (WAVE) The wavelength feature estimate the length of a waveform in a segment (Boostani, R. and Moradi, M. H., 2003).

$$WAVE = \sum_{i=n-N+1}^{n} |\Delta x_i| \tag{38}$$

where Δx_i is defined as $\Delta x_i = x_i - x_{i-1}$.

x_i is the ith sample. N is the number of samples in each segment. Threshold was set to 0.15 in this study.

The Matlab function `EMGwave.m` is found in Appendix A.12.

Zero-crossings (ZC) Zero-crossings count the number of times the EMG-signal crosses the zero amplitude level (Boostani, R. and Moradi, M. H., 2003).

$$ZC = \sum_{i=1}^{N} sgn(-x_i x_{i+1}) \tag{39}$$

where

$$sgn(x) = \begin{cases} 1 & \text{if x > threshold,} \\ 0 & \text{otherwise.} \end{cases} \tag{40}$$

x_i is the ith sample. N is the number of samples in each segment. Threshold is set to 0, makes it independent of amplitude changes.

The Matlab function `EMGzc.m` is found in Appendix A.13.

It is also possible to take zero-crossings and average amplitude value of the wavelet coefficients as well as finding AR coefficients of the wavelet coefficients. However, this is not explored in this thesis.

5.3 EMG filtering

We want the average of the EMG signal to be zero. To do this we need to remove the zero frequency. A simple high-pass filter (41) has been used to remove this distortion. If more than just the zero frequency is removed, the signal could loose important information, and a given feature would not perform as good as it should. Obviously it is not possible to remove zero frequency, but a cut-off frequency $f = 0.25Hz$ should hopefully be enough. The gain was set to $K = 0.57$. The features contained more valuable information with this filter properties.

$$H(s) = \frac{\frac{K}{2\pi f}s}{\frac{1}{2\pi f}s + 1} = \frac{Ks}{s + 2\pi f} \tag{41}$$

Realization of the filter can be found as the function `hpfilter.m` in Appendix A.14.

5.4 Selected features and calculation time

Table 6 below lists the extracted features (in total 16) with the associated calculation time. It is a measure on the time it takes to calculate a feature for all (eight) electrodes and a window frame equal to the downsampling. The abbreviation for each feature used in MATLAB and throughout the paper is shown in the left column.

Abbrev.	Feature	Calc. time (s)
AAV	Average Amplitude Value	0,0403
AAC	Average Amplitude Change	0,0422
AR	AR coefficients	0,4002
CC	Cepstrum coefficients	0,0561
EWC	Energy of wavelet coefficients	1,4572
EWCL	Energy of wavelet coefficients loss	1,4572
EWPC	Energy of wavelet packet coefficients	25,1284
EWPCL	Energy of wavelet packet coefficients loss	25,1284
HIST	Histogram	0,5815
MYOP	Myopulse percentage rate	0,0425
NT	Number of turns	0,0577
PROC	processed EMG with AAV	NA
VAR	Variance	0,0456
WAMP	Wilson Amplitude	0,0506
WAVE	Wavelength	0,0546
ZC	Zero-crossings	0,0516

Table 6: Features and calculation time

6 Implementation

6.1 Feature Selection

Not all features may give useful information when they act alone. Likewise, it is not always certain that a feature subset will perform better overall than one of the single features in the subset. Even though, it is more likely that a set of feature combinations will give better classification results than just a single feature. Therefore, it is desirable to try different combinations of features and feature subsets in this case.

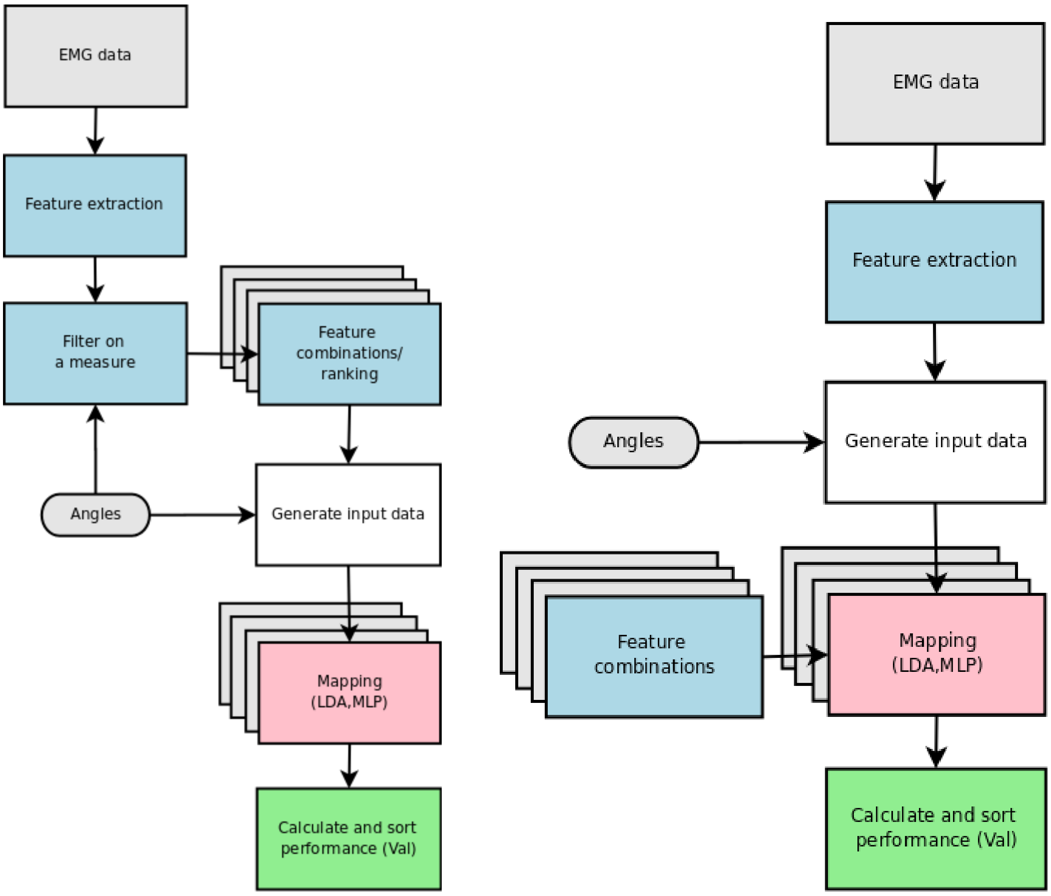
6.1.1 Subset selection approach

Having considered some of the academic papers on wrappers it is clear that this method would not be suitable, because it uses the classifiers results to evaluate and select features. In this case it would mean that all feature combinations were to be computed and then arranged accordingly, or used a sequential method adding or subtracting features giving an extremely complex and computational nightmare for a computer running a MLP. In addition, according to Robi Polikar (2006) one cannot use this method for time-series based features such as ECG because features are strongly correlated.

The filter method on the other hand could possibly be used because it does not evaluate and select features based on the classification method. This approach may be implemented in MATLAB with the built-in function `sequentialFS` as either FSS or BSS. It would require input values such as EMG feature and angle data, and a function for evaluation criterion. Such a function could be (as mentioned earlier in section 3.3.2) MSE or for that matter correlation. The output could be a list of feature combinations. This approach is depicted in Fig. 17a

Sending in EMG features and angle data and try to find a relation would be difficult. Given 8 EMG recordings (one for each muscle) and 4 measured angles it would be difficult to find an evaluation function to rank the features individually. It would quite possibly result in a favouring of features with amplitude alternating properties like the course of a given angle if one was to choose MSE or correlation as a criterion. Also, it would be difficult to find a meaningful relation when each feature contain 8 or more values for each time step. (Some features give out an array of coefficients for each muscle (AR, HIST, EWC), in such a case, each coefficient would have to be interpreted as a single feature) .

One could directly map given muscles (agonist/antagonist) to given angle; this approach could give meaningful information to filter on. However the filter approach was deemed inadequate.



17a: Ideal Filter Approach

17b: Constructed Approach

Figure 17: Fig. 17a show a theoretical filter approach on how to rank features and construct subsets to be calculated in a mapping function. Fig. 17a was implemented. Generate input data constructs an input for the mapping process where angles and feature data is fed to it. The feature combinations boxes indicates that there are many combinations to be tested, while the mapping box indicates that it is to be calculated multiple times with multiple mapping functions (LF, NN)

Given all 16 calculated features, a total number of $2^{16} = 65536$ combinations are possible. Because we try to adapt EMG signal to angle based motion, we involve the use of MLP. Since this approach is a computationally extensive, a first thought of running through all combinations looked impossible.

Previous attempts on using ANN with 10 nodes on the given data sets, indicated a calculation time of about 5 minutes a feature.[15]

[15] Based on using a *Core2Duo E4300* processor $(2x2.4GHz)$ with $4GB$ of RAM.

$$\frac{2^{16} \times 5[minutes]}{60[minutes] \times 24[hours]} \approx 227[days]. \tag{42}$$

In addition it is often pays back to try an ANN multiple times because it uses random initial starting values on the nodes. Also, one should incorporate some overhead in case something goes wrong (Unpredicted computer shutdown).

The solution was to skip the feature weighting using wrapper or filter approach and instead, use the obtained information from earlier work about a given feature. Multiple features were then manually combined in order to test subsets with features that could complement each other. A list of 100 subsets were chosen ranging from 1 to 5 combinations. The first 16 combinations were the single features. Based partly on the result in Bach (2008) and partly on new tests on improved feature calculations and wavelet features, different combinations were picked. Some were randomly chosen while others were specifically manufactured because they either had very good properties or performed best in Bach (2008). Some feature combinations were also constructed as a kind of sequential adding of specific features. The list over tested combinations can be found in Appendix B.5.

The feature subsets were implemented as a file with binary values indicating what features to use. Each subset were fed to a mapping function.

The following approach was chosen because of the limited time to compute results. It may very well not turn out to be the best approach, but it may give good indications on what to focus on in future work.

1. Select a subset
2. Construct input data
3. Classify/Map $\forall \{LF, MLP(,...)\}$
4. Calculate performance criterion on validation data
5. Rank features/subsets

6.2 New features

Energy of wavelet coefficients *"A wavelet is a mathematical function used to divide a given function or continuous-time signal into different frequency components and study each component with a resolution that matches its scale."* (Wikipedia, The Free Encyclopedia, 2009*k*)

In this thesis the energy of wavelet coefficients was implemented in Matlab with the `wavedec` function which is a multilevel 1-D wavelet decomposition. It computes the wavelet packet tree for a given discrete wavelet and number of levels. The result was then fed to the `wenergy` function which returned the percentage of energy corresponding to the approximation coefficient and details coefficients. The $n-1$ details coefficients were then used as input arguments as one feature, and the single approximation coefficient as another.

Energy of wavelet packet coefficients Using wavelet packet transform instead, Boostani & Moradi obtained superior result. It is more complex, and calculation time is longer. This can also be seen in table 6.

In our case it was implemented in Matlab with the `wpdec` function which is a wavelet packet decomposition 1-D. It returns a wavelet packet tree corresponding to the WPD at a given level. It was then fed to the `wenergy` function which returned a vector with the percentage of energy corresponding to the terminal nodes of the tree.

6.2.1 Choice of wavelet

Boostani & Moradi used a bi-orthogonal wavelet in 9 scales because of a dyadic relation of the decomposition to the number of samples in each step. They used a rather large time window, 200ms.

Numerous research have deployed wavelet before, and experience from some were taken into account when deciding which wavelet family to use.

J. Kilby & H. G. Hosseini (2004) tested *haar, bi-orthogonal, daubechies, coiflet* and *symmlet* wavelet families. They indicated that *daubechies*, $db5$, could be most suitable for analysing EMG signals, but both *symmlet* and *coiflet* should be evaluated when choosing a wavelet for a project. K. Englehart, B. Hudgins, P.A. Parker, M. Stevenson (1999) used *Coiflet*, $coif4$, with WT and *symmlet*, $sym5$, with WPT. D.K. Kumar,N.D. Pah & A. Bradley (2003) tried three different wavelets (*haar, daubechies* and *symmlet*). $sym4$ and $sym5$ was found to give best results. Martha Flanders (2002) employed *Daubechies*, $db2$, in her studies.

Five different wavelet families were tested to find a most useful wavelet family for this problem. The two features EWC and EWPC were implemented with the Matlab functions as described earlier. **Symmlet of order 4 with 5 decomposition scales** was chosen because it evidently preserved most information from an EMG signal, leading the mapping functions to find a good estimat of the desired motions. Different scales was used in the approach, but the all wavelets performed best at 5 scales. `haar, bior2.2, db5, coif4` all performed relatively well, but it was seen that they either did not manage to express the fluctuations in the angle estimate as well, or they followed too slowly. This was especially clear when looking at large angle variations, like in pronation/supination.

6.3 Filtering and smoothing

Averaging a signal over a time $T = t_j - t_i$, is a digital way of smoothing a signal. The larger the time window is, the smoother a signal will become. The drawback of this, will be that the signals amplitude might get suppressed, because fast changes in the signal will average out because of their short appearance. Often it is convenient to low-pass filter a signal in order to get the smooth variations of a signal. Moving average is the operation where the time window is shifted along the signal in order to obtain a time-varying average. (J. V. Basmajian & C. J. De Luca, 1985)
The Matlab implementation of the smoothing function `smoothing.m` can be found in appendix Appendix A.16.

6.4 Performance indicators

From previous work it was though that RMSE would give a fairly reasonable value on how well a feature performed. It might give quite a good indication, but the value it self does not tell anything about how well the measured signal follows the original, wanted, signal.

6.4.1 Root mean square error (RMSE)

The RMSE is a measure of accuracy, like the difference between values from an estimation and the actually observed values from what is being estimated. Given a parameter θ and an estimator $\hat{\theta}$, the RMSE is defined as

$$RMSE(\hat{\theta}) = \sqrt{MSE(\hat{\theta})} = \sqrt{E((\hat{\theta} - \theta)^2)} \tag{43}$$

Where E is the expected value operator. It was realized in MATLAB by applying the square root on the built-in function `mse`.

6.4.2 Correlation Coefficient (CORR)

The correlation coefficient is a statistical measure that indicates the strength and direction of a linear relationship between two random variables Wikipedia, The Free Encyclopedia (2009*c*).

The mathematical interpretation of correlation is

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_x \sigma_y} = \frac{E((X - \mu_X)E(Y - \mu_Y))}{\sigma_x \sigma_y} \tag{44}$$

Where E is the expected value operator, μ is the expected value and σ is the standard deviation. The correlation coefficient is a value between -1 and

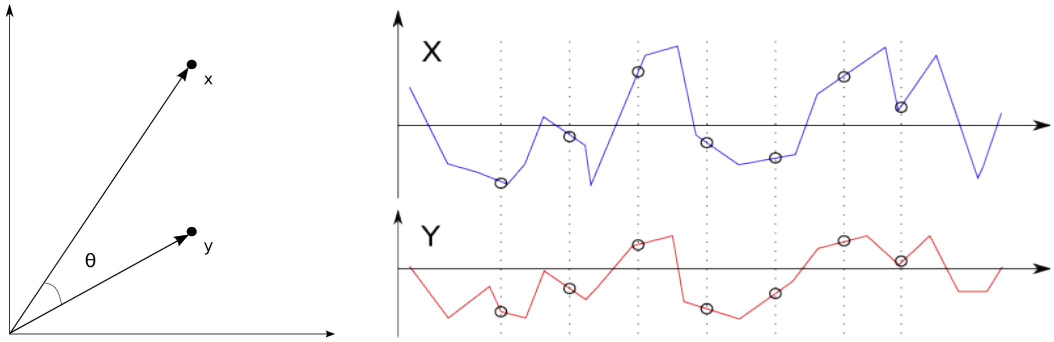
1, where extremities indicates a complete negative or positive linear relationship, respectively, and zero indicates no linear relationship or co-movement. The built-in MATLAB function `corrcoef` was used. This indicator will be denoted as CORR from now on.

6.4.3 The Cosine Similarity Transform (CST)

To see if the shape of an estimated signal is equal to a reference the cosine similarity transform may come in hand. We are not concerned with the amplitude of the signal, just the direction.

$$cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}||\mathbf{y}|} \tag{45}$$

θ is independent of $|\mathrm{x}|$ and $|\mathrm{y}|$ If the angle between two vector samples decreases, the cosine angle approaches 1 (E. Garcia, 2006). The resulting cosine angle will have a range between -1 and 1.



18a: Angle calculation

18b: Samples from two signals

Figure 18: A measure to find if two signals are similar even though their magnitude is different.

6.4.4 Coherence

Coherence in signal processing is a statistic that gives the relation between two signals. It is commonly used to estimate the power for each frequency in a signal.

$$C_{xy} = \frac{|G_{xy}|^2}{G_{xx}G_{yy}} \tag{46}$$

where G_{xy} is the cross-spectral density between x and y. G_{xx} and G_{yy} represents the auto-spectral density of x and y respectively. $|G|$ is the power of the spectral density.

The values of coherence lies in the range $0 \leq C_{xy} \leq 1$, indicating 1 for MATLAB function `mscohere`. The problem lies in that instead of obtain a single coefficient defining the performance, one get an array of coefficients (Wikipedia, The Free Encyclopedia, 2009a). This is difficult to interpret together with Correlation and RMSE, and was therefore not used.

One may also use the Power spectrum density to find the energy of each frequencies representing each signal. These spectrums may give an indication about how equal the signals are.

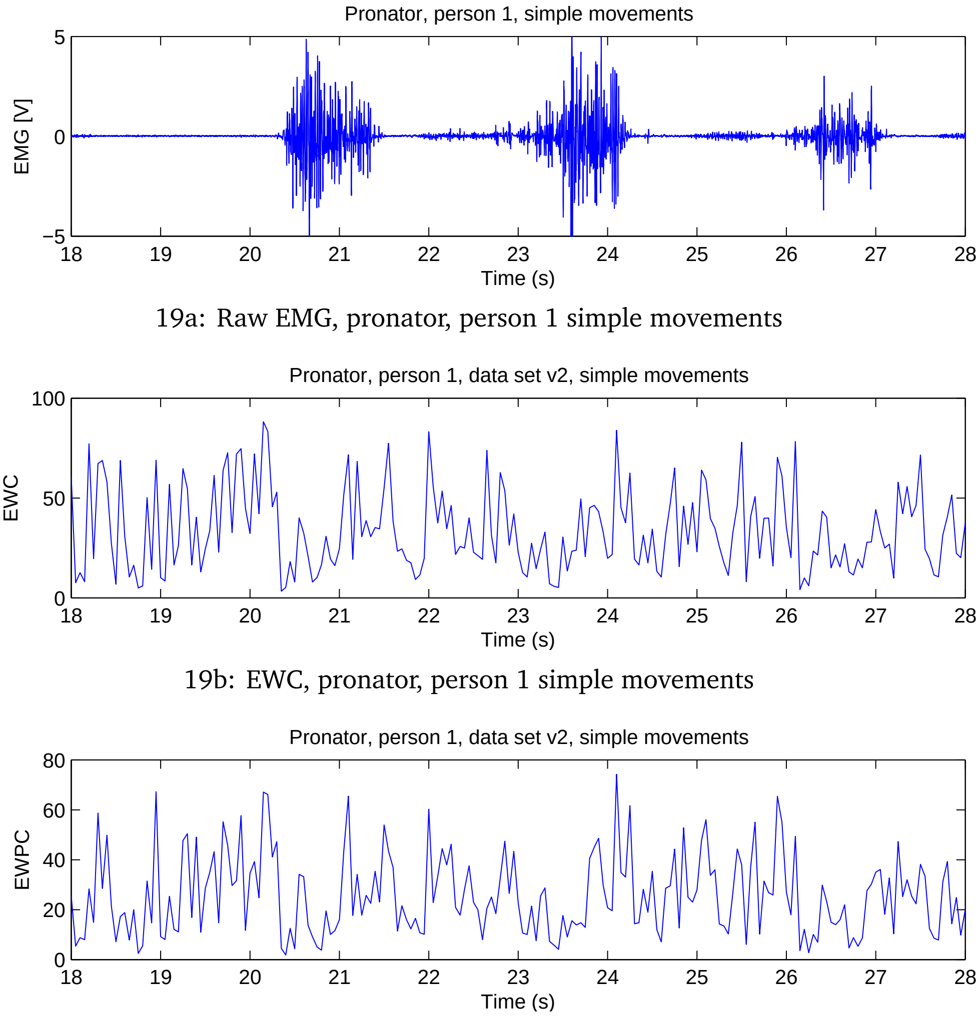
6.5 MATLAB implementation

A great effort has been put in trying to make the MATLAB program easy to use. All variables are placed in a single file to be edited. The specific MATLAB files have been categorised, and an extensive folder hierarchy has been created to achieve a better structure of the data that are calculated. This setup will make it easy to add new subjects, classifiers and features in the future.

7 Results and observations

7.1 EMG features

A 10 seconds sample segment shown below is taken from person 1 showing channel 1 which is feeding pronator activity. The plots show features at 20Hz. They seem quite noisy and should be filtered if possible before using them further.



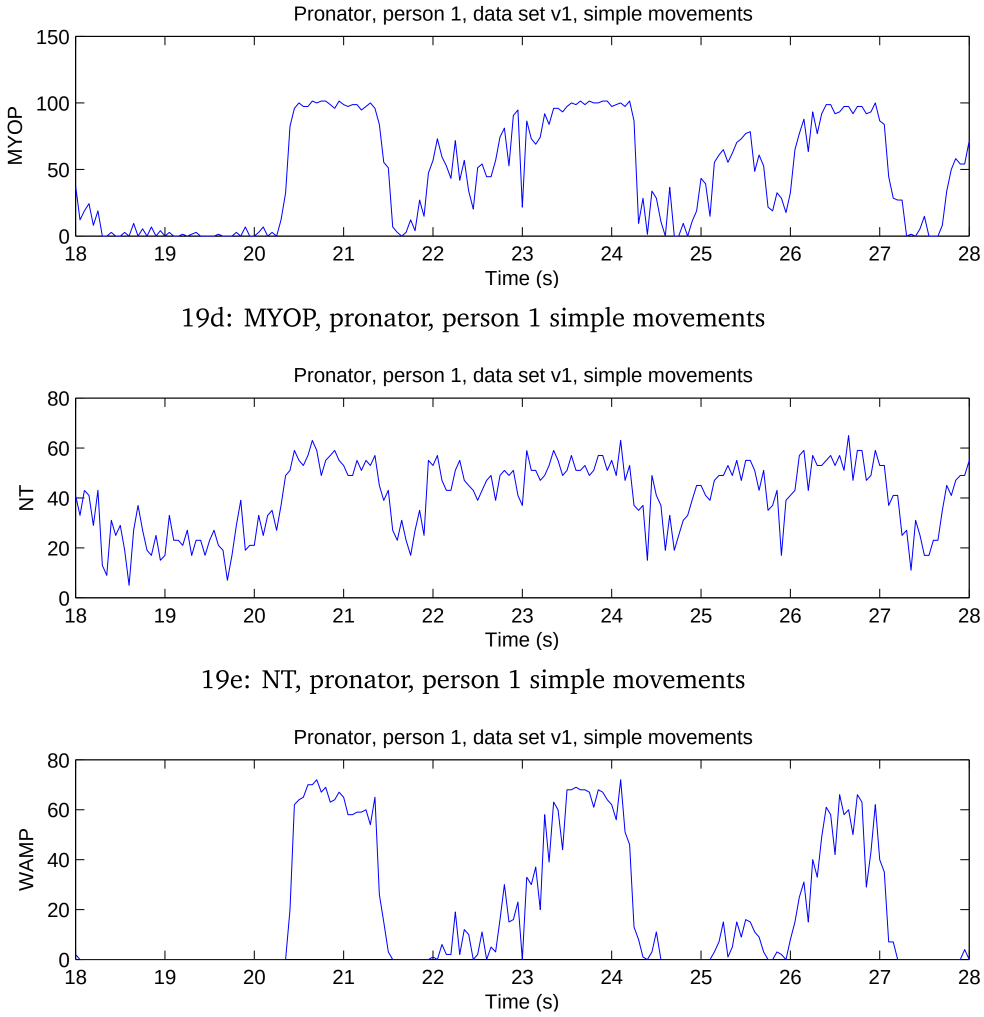
19a: Raw EMG, pronator, person 1 simple movements

19b: EWC, pronator, person 1 simple movements

19c: EWPC, pronator, person 1 simple movements

Figure 19: EMG features, channel 1(Pronator), person 1

The computation algorithm for NT (see fig. 19e) has been altered from Bach (2008) where it did not give any useful information. WAMP (see fig. 19f) and MYOP (see fig. 19d) did not achieve any good results because a wrong thresh-



19d: MYOP, pronator, person 1 simple movements

19e: NT, pronator, person 1 simple movements

19f: WAMP, pronator, person 1 simple movements

Figure 19: EMG features, channel 1(Pronator), person 1

old was set. This has now been corrected. EWC (see Fig. 19b) and EWPC (see Fig. 19c) are depicted here as one function. It is a summation of each detail coefficients from the computation. The coefficients tell what percentage of energy is kept in the signal after the wavelet decomposition. EWCL and EWPCL are not shown, but they show the percentage of energy lost after the decomposition.

7.2 Evaluating the results

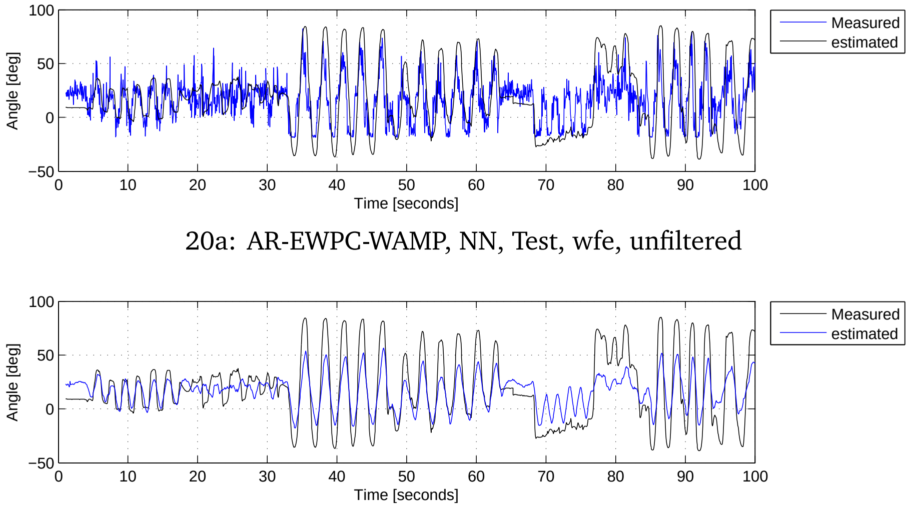
In order to evaluate the results, a closer look at the angle wrist flexion/extension will be carried out. This angle performed best and was valid for both classifiers.

Due to the large data material after such a calculation, it is important to have a reliable and efficient measure that tells which feature is best. Because earlier tests indicated that RMSE did not achieve better results when the signal was filtered, two other methods were considered. These methods, CORR and CST, are also calculated and evaluated together with RMSE.

In this problem we have data from 3 subjects from 3 individual recordings. The combinations should be ranked according to the performance on the data set that is used as validation for the classifier. Given that we only have 3 subjects it is difficult to find statistical evidence that one subset perform better than another. Even though, the results depicted in the tables below are average of the subjects performance.

7.3 Filtering

The observed estimated signal is very noisy, and it is often difficult to see how well the signal actually reflects the true motion it attempts to estimate. This problem may very well reduce the performance indicated by a measure such as RMSE or correlation. Below (see Fig. 20) are two plots showing the difference after the signal has been estimated by the LF. A smoothing function that averages the signal from the last n (in this case $n = 15$) samples was added. This approach will damp the amplitude on the estimated signal and in addition phase shift the signal.



20a: AR-EWPC-WAMP, NN, Test, wfe, unfiltered

20b: AR-EWPC-WAMP, NN, Test, wfe, filtered

Figure 20: A more readable plot is available when one filters the signal.(showing subset 59 and test results from subject 1.)

Type	RMSE	CORR	CST
unfiltered	20.4	0.58	0.72
filtered	16.4	0.81	0.84

Table 7: Based on the performance for subset 59 depicted in Fig. 20 the values for RMSE, CORR and CST are shown for filtered and unfiltered signal respectively. Note that RMSE is in % of ROM.

Lower % of ROM is good for the RMSE while the values for CORR and CST should be as close as possible to 1. It was first anticipated that RMSE did not vary much if the data was filtered past of the mapping procedure. This seems incorrect, but the value may still not provide a good measure for how good the estimated signal follow what is being estimated. Fig. 21 and Tab. 8 show values for an estimation that is faulty. The succeeding plots will all be of a filtered estimated signal.

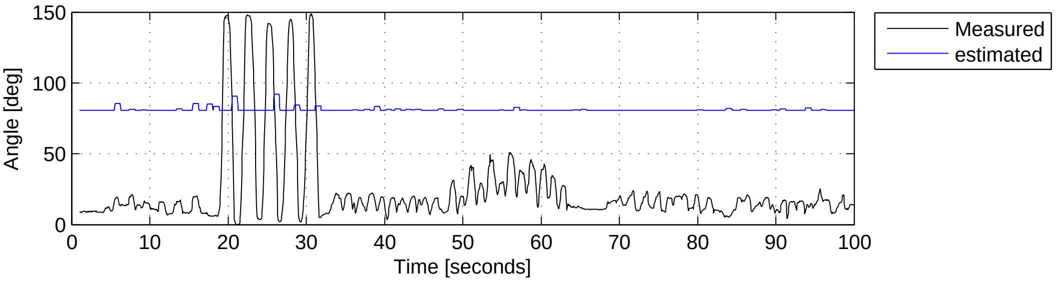


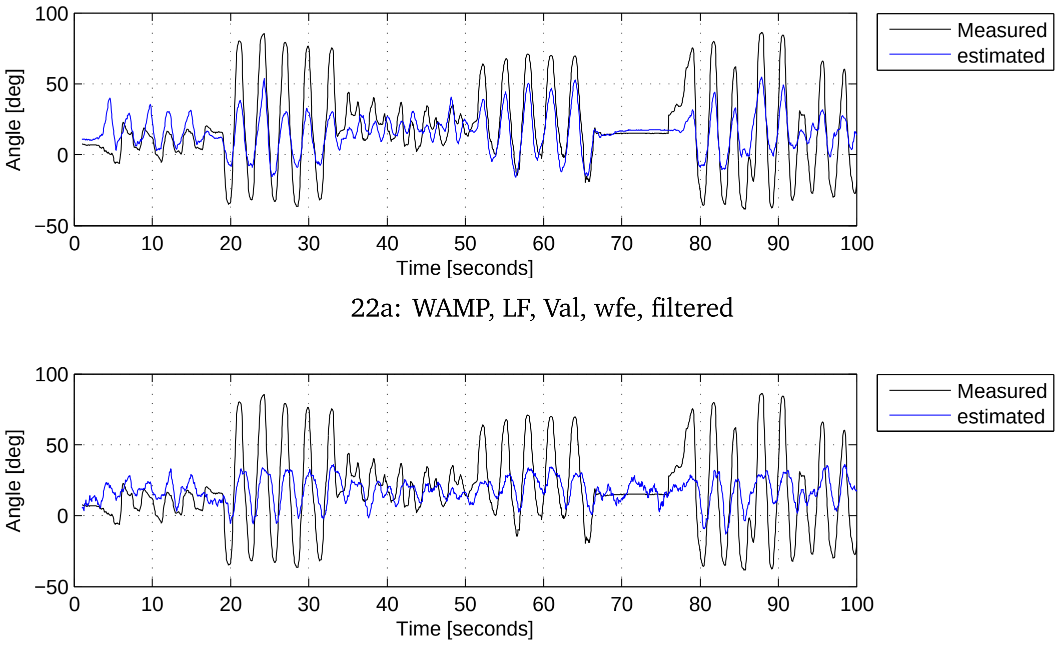
Figure 21: AR-EWPC-WAMP subset performance from NN on angle pronation/supination. (Subject 1 and test data used.)

Type	RMSE	CORR	CST
filtered	36.96	0.10	0.64

Table 8: Based on the performance for subset 59 depicted in Fig.21 the values for RMSE, CORR and CST are shown. Note that RMSE is in % of ROM.

7.4 Wavelets as features

It can be seen in table 9 that the wavelet features were not among the best when it comes to single features. Instead, two of the wavelet features performed among the worst. Overall, the features did not perform as expected relative to results indicated when the wavelet family was selected. Expanding the window (currently of 75 samples) it is applied to on a signal might increase the features relevance. Noticing that this is for the linear mapping function, both EWC and EWPC did follow the variations in the signal, but the magnitude was not followed that well. This can be seen in seen in Fig. 22b. In comparison, the best feature is depicted in Fig. 22a. Note that the wavelet performs considerably better when it is applied to a NN. This can be seen in Fig. 23.



22a: WAMP, LF, Val, wfe, filtered

22b: EWPC, NN, Test, wfe, filtered

Figure 22: EWPC and WAMP performance depicted to show how good each feature is on an angle basis.

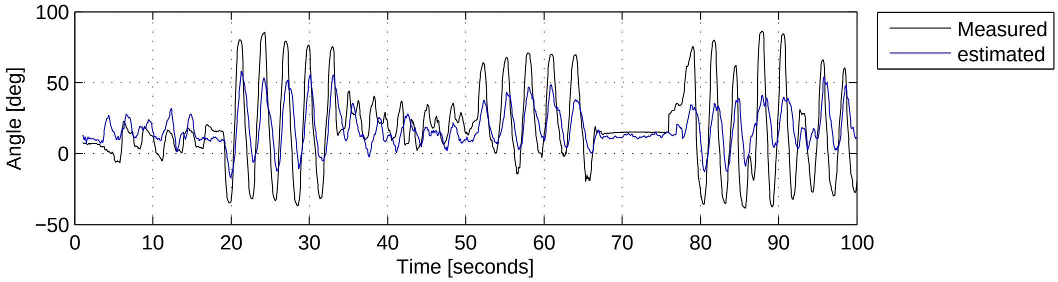


Figure 23: EWC performance from NN on angle wrist flexion/extension. (Subject 1 and validation data used.)

LF Wrist flexion/extension				
Num	Feature	RMSE	CORR	CST
14	wamp	15.06	0.83	0.86
1	aac	15.09	0.81	0.86
15	wave	15.10	0.81	0.86
12	proc	16.74	0.69	0.81
2	aav	17.59	0.69	0.79
10	myop	17.90	0.68	0.79
7	ewpc	18.26	0.68	0.77
3	ar	18.50	0.63	0.76
5	ewc	18.67	0.67	0.76
4	cc	19.76	0.63	0.75
16	zc	20.28	0.50	0.70
9	hist	20.52	0.57	0.70
11	nt	20.68	0.54	0.69
8	ewpcl	20.89	0.47	0.68
6	ewcl	21.07	0.46	0.68
13	var	21.80	0.33	0.65

Table 9: Features ranked (Ranking criteria is RMSE (% of ROM) on validation data). Correlation value changes considerably between best and worst feature.

NN Wrist flexion/extension				
Num	Feature	RMSE	CORR	CST
15	wave	13.50	0.86	0.90
1	aac	13.58	0.86	0.89
14	wamp	14.05	0.86	0.89
12	proc	14.77	0.77	0.86
2	aav	15.21	0.80	0.86
10	myop	15.62	0.79	0.85
3	ar	16.44	0.74	0.82
13	var	16.63	0.75	0.82
5	ewc	16.80	0.75	0.82
7	ewpc	17.28	0.73	0.80
4	cc	18.37	0.72	0.78
11	nt	18.54	0.71	0.78
16	zc	19.37	0.59	0.74
9	hist	19.97	0.61	0.72
8	ewpcl	20.25	0.53	0.71
6	ewcl	20.44	0.52	0.70

Table 10: Features ranked (Ranking criteria is RMSE (% of ROM) on validation data). Correlation value changes considerably between best and worst feature.

7.5 Sub-optimal subspace

To reduce the diversity we keep the observations and results to the NN and focus on wrist flexion/extension and finger flexion/extension due to the fact that the other two angle estimates were faulty for all feature combinations.

Neural Network The NN results are averaged over the 5 best runs from a total of 50 runs for each combinations.

According to RMSE the best subset among those tested where the EWC-WAVE combinations. The validation and test estimation is shown in Fig. 24 and is a plot for subject 1. The values indicating the performance for all the measures are depicted in Tab. 12. It can be seen that all measures indicate very high estimation accuracy.

(a) RMSE, NN, Validation

Feature	RMSE of ROM [%]	
Num	WFE	FFE
80	12.99	23.58
42	13.00	23.91
35	13.01	24.13
6	20.44	25.51

(b) RMSE, NN, Test

Feature	RMSE of ROM [%]	
Num	WFE	FFE
80	17.06	30.58
42	19.26	31.30
35	17.85	30.50
6	22.64	31.11

(c) CORR, NN, Validation

Feature	Correlation	
Num	WFE	FFE
66	0.863	0.523
15	0.861	0.525
1	0.859	0.510
6	0.516	0.229

(d) CORR, NN, Test

Feature	Correlation	
Num	WFE	FFE
66	0.559	0.286
15	0.577	0.288
1	0.622	0.241
6	0.305	0.051

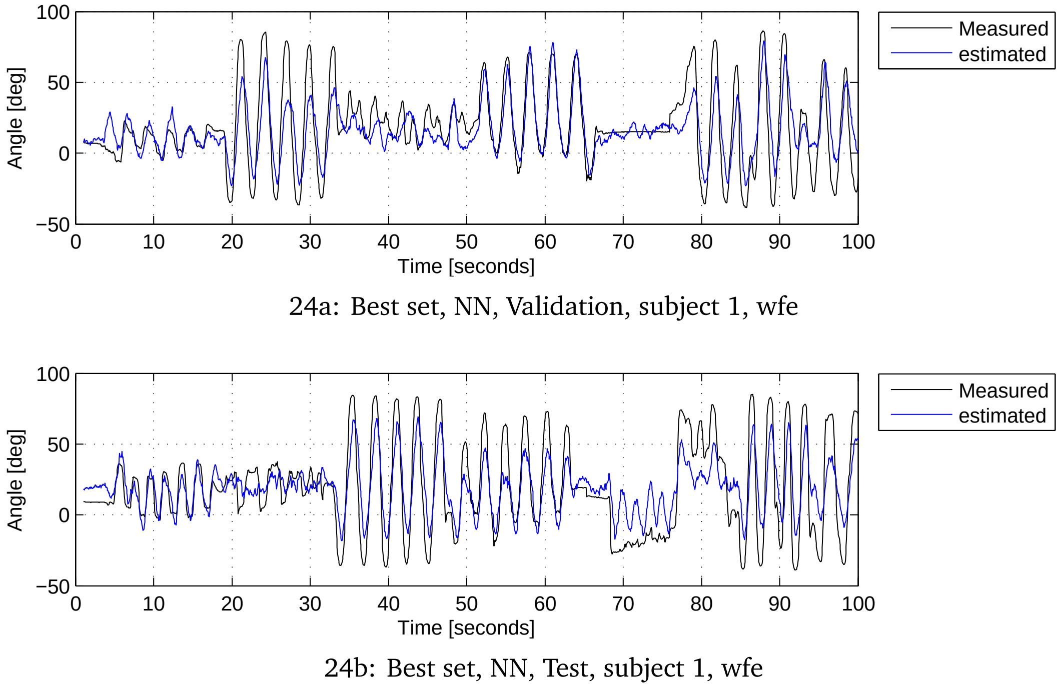
(e) CST, NN, Validation

Feature	CST	
Num	WFE	FFE
80	0.900	0.558
35	0.900	0.567
57	0.899	0.537
6	0.703	0.387

(f) CST, NN, Test

Feature	CST	
Num	WFE	FFE
80	0.822	0.316
35	0.821	0.304
57	0.814	0.270
6	0.655	0.208

Table 11: Tables showing the 3 best features calculated by NN. The values are sorted on best validation data 10(a), 10(c) and 10(e). Test data shown in 10(b), 10(d) and 10(f). Record no. 4 on each table is showing the worst feature combination as comparison.



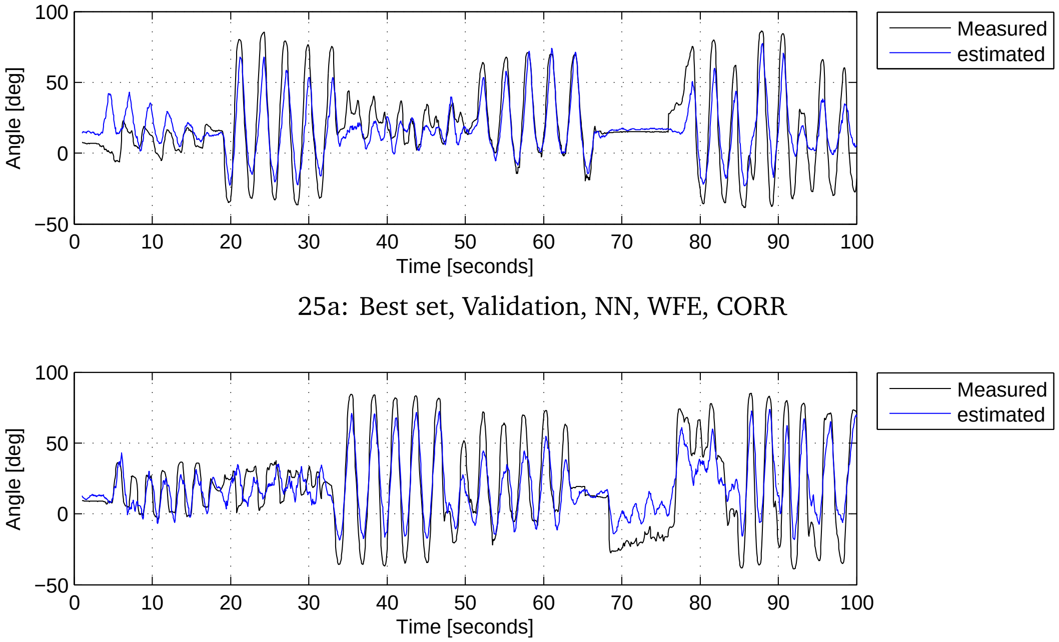
24a: Best set, NN, Validation, subject 1, wfe

24b: Best set, NN, Test, subject 1, wfe

Figure 24: Best subset according to RMSE/CST using NN on angle wrist flexion/extension. (Validation and test data shown for subject 1.)

Wrist flexion/extension			
Type	RMSE	CORR	CST
Val	13.31	0.86	0.89
Test	15.85	0.81	0.84

Table 12: Performance values for the EWC-WAVE combination, thought to best based on RMSE/CST measurement.



25a: Best set, Validation, NN, WFE, CORR

25b: Best set, Test, NN, WFE, CORR

Figure 25: Best subset according to CORR using NN on angle wrist flexion/extension. (Validation and test data shown for subject 1.)

Wrist flexion/extension			
Feature	RMSE	CORR	CST
Val	13.37	0.87	0.89
Test	14.19	0.85	0.88

Table 13: Performance values for the WAMP-WAVE combination thought to best based on CORR measurement.

Larger plots for wrist flexion/extension showing how it adapts to combined movements are depicted in Appendix B.10 and Appendix B.11, while plots for the finger flexion/extension are depicted in Appendix B.8 and Appendix B.9. Note that the plots do not show what kind of movements are performed but some of the combined movements (in a data set) are listed in Tab. 3.

Three values, one for each performance indicator, indicating which subset is best, is depicted in Tab. 11.

8 Discussion

Since three performance indicators were used in this thesis as a measure to find an optimal subset, different combinations were found to be optimal in the sense of being the best among those tested. Both RMSE and CST indicated that *EWC-WAVE* performed best while CORR suggests that *WAMP-WAVE* was the optimal combination. These findings are based on the fact that we use NN to map data.

Performance indicator Now, the interesting part is what these performance indicators suggest as the second and third best combinations. RMSE and CST suggest *EWC-EWCL-WAVE* to be respetively second and third best combination, but note that this also includes the combinations that were found to be best. Besides, all indicators define EWCL to be the worst (in other words, it is not a informative fature) feature and clearly this poor feature reduces the performance instead of improving it. This feature should eventually be removed from a search for a suitable subset (It may be that it will perform better with other features, but it is not likely). CORR on the other hand measures *WAVE* and *AAC* as single features to be better than most subsets. This is surprising. If one take a closer look at the estimated values vs. the measured, one may see differences (Fig. 25 and Appendix B.12). It is a feature with good properties, since it e.g. reflects stationarity. Notable around 65-78s in validation figures. But the fact that both RMSE and CST both suggest the same subset should be attached importance to. Because we are more familiar with RMSE becaus it presumably gives an indication of ROM, the first though will indicate that it is the measure to use. The drawback of RMSE is its inability to discriminate between significant and non-significant estimations. For example, an RMSE measure of 15% (of ROM) indicates a very good estimation but at 20% the signal is weak at representing the amplitude of the signal. CST and CORR provide much more informative results. Their values change from approximately 0.9 to 0.6 and 0.9 to 0.3, respectively. This corresponds more closely to expectation, and is possibly a better way to rank features. That being said, CST is a good measure indicating whether two signals are approximately equal, except in size. The size, in this case the amplitude, may perhaps be altered with a non-linear dead-zone filter approach. It is more important that the signal is similar in shape than in size. Squaring each term, like RMSE does, result in weighting outliers such that they become dominant. If an estimated signal fluctuates around the mean value of the signal but does not map correctly, or follow large amplitudes which only occur a few times in the signal, the RMSE value becomes surprisingly small. Thus, is less reliable for identifying sub-optimal combinations. I therefore indicate that the use of CST or CORR to measure performance a superior result to that of RMSE.

Optimal subset Based on what is observed and defined in the preceding chapter, an optimal subset will most certainly contain multiple features. An interesting observation is that among the eight best subsets (See Appendix B.15), *EWC* appear in four combinations while *WAVE* appear 5 times. This indicates that these features contain important information. As table 10 illustrates, EWC does not emphasize its important properties alone. The optimal subset was found to be *EWC-WAVE* as mentioned in the beginning, based on the fact that two out of three indicators suggested it. But this is definitely not a globally optimal solution, because only approximatley 0.15% of the feature combinations have been tested.

One important parameter that should be mentioned is the computation time it takes to attain a feature. Table 6 show this, and *EWC* has a very long calculation time (without mentioning *EWCL*) with respect to the others. A prosthesis should give a fairly real-time feedback on what is being performed. One may weight features also in terms of calculation time, but because the calculation capacity for a computer increases, this may not be of a significant importance.

The fact that *AAC*, *EWC*, *WAVE*, and *ZC* (Can be seen in Appendix B.14 and Appendix B.15) are used to combine the best subsets, tells us that these features are individual, important and quite possibly complement each other. That being said, it is not known if other features will perform excellent given the right subset.

16 features have been used as a basis for feature combinations. Some of these features may be removed or refined if one look closer at the results. Monotonicity has not been identified here, even though the best subsets for LF was those that had most combinations of features.

Filter approach When the assignment was given, it was thought that one should use a filter approach in order to find an optimal subspace of features. Quite some time were used to sketch possible solutions, but eventually it was found inadequate due to the that fact that it was difficult to find a function to measure a features, relevance on without actually mapping the features first, and then use RMSE. But the fact is, it would in a way become a wrapper method. But we could have possibly altered this approach. None of the sketched alternatives were feasible, but there might be others. For instance, one could have used the LF as a filter technique because it has a short calculation time. Trying all combinations or use a sequential approach, and given a good measure on performance, the feature combinations could be ranked according to the performance on the validation set, and the best subsets could be fed to train a NN. However, we loose a classifier and become dependent on the performance of the NN. Even this would only lead to a sub-optimal solution since it is not given that LF will find what is the best subset trained with a NN. If the fact that we have a weak form of monotonicity for LF (The more features the better), the ranking from this will lead to extremely complex combinations to calculate for the NN.

Clearly, trying to find an optimal subset is difficult, but the preceding has proved that the performance increases with more than just one feature. There are at least two factors we need to take into account when analysing the results. Not only did we investigate few combinations but we received faulty data (with respect to NN) for the calculation of two angles; pronation/supination and radial/ulnar deviation. There are two possible outcomes from this. A wrong set-up with the input to the NN, or the fact that the choice of 10 hidden nodes were wrong and possibly led to weighting of just the first two angles. If one is to have a subset consisting of many features, it may help improving a subset by adding more nodes to the hidden layer in a NN. This however, will increase complexity and the calculation time will increase. This may demand more memory in a computer that calculate these feature combinations.

Motions We are measuring four angles, but not all angles share the same importance for a prosthesis user in the first place. If one is to implement a practical proportional control system for a prosthesis, a review over what is important is in its place.

The radial/ulnar deviation has the smallest range of motion. It is clearly useful when one tries to execute a task of precision. When trying to pick up an item, this angle has an advantage, but it is only a fine motoric precision angle as opposed to the shoulder or the elbow joint. Even though, at small angles, it is still an incredibly useful angle, but hardly important. One should still not dismiss it entirely. A realisation of this angle need not have a full range of motions, thus making it easier to implement with the results of today.

The wrist flexion/extension is an angle that has performed quite good in this thesis. The reason for this may be that the muscles involved in this motion are easy to access with EMG electrodes because they are superficial muscles and quite large. It is an important angle making it possible to support one self, by adjusting the palm of the hand, for instance, when trying to get up of a chair.

Finger flexion/extension did not achieve very distinct results, but the motions are adapted without big amplitude significance. It is an important angle, because it is used to grip and point.

The pronation/supination angle did not achieve any good results, and this was disappointing. Since this motion has a large ROM it is an indication that the angle is useful for a prosthesis user. For instance may be used to turn a key in a lock, pour a glass of milk and to turn a paper on a desk. The estimation of this angle worked much better in a previous work, (Bach, 2008), and this performance drop may have been affected by the lack of hidden nodes in NN.

That being said, it is not necessary to develop a true cyborg prosthesis. Given what is available today for a prosthesis user is extremely simple but possibly difficult to adapt to in the beginning, we do not need to implement all angles at once. This approach may in first place double the number of DoF in a prosthesis today. This will indicate the need for two DoF. Based on the information acquired in this thesis, the wrist flexion/extension and the finger flexion/extension seem feasible within a few years. It would of course be convenient with the angle pronation/supination, but given the result available as of today this is not possible.

Improvements of the signal From what is indicated in section 7.3 we have found that filtering is necessary. A filtering process should also be implemented in future work after the features have been extracted from the EMG signal, even though the NN should work as a low-pass filter. Since the NN is of a non-linear type and if the input signal is noisy, the weighting may be affected by this. If one filters the signal in front of the feature extraction, the filter coefficients will dominate the feature rather than the specific feature that lies in the signal.

Since the amplitude of the signal often is of a low resolution, a non-linear form of scaling might be helpful. Even though, one should not underestimate the significance of actually haveing a visual feedback when a prosthesis is implemented for a user, this may very well elliminate that problem. But it may involve another problem, a movement would possibly be thought of as heavy without having any load on it. This is not what we want.

Feedback from the user will be important. The first closed-loop regulation will be the visual feedback from what is being observed, and what is intended. An important aspect considering this is the bandwidth the existing human interface have. A desirable connection would be that a prosthetic control would operate in the range of the human's ability to sample sensoric and visual information. This would hopefully let the user obtain a control over the motion.

If the user senses a feeling that he actually controls the prosthesis, it may not be that important to have 100% classification accuracy. An adaptive control system would perhaps be a meaningful approach. This however, is for now bounded by the fact that we train a NN ahead of implementation in a prosthesis.

We are able to estimate fairly well what is meant to be executed from a users perspective. Even though not all angles give sufficient results yet, an ongoing approach towards this problem may in the future be realized as a prosthesis with more DoF's than a prosthesis user have today in addition to be able to use the "correct" muscles formerly used for that action.

Ethical dilemma Binding the nervous system and a computer brings up ethical questions. Trying to achieve a link between the mental functioning of a human and a machine may not be executed without thinking of consequences. Will a prosthesis user be enhanced by such a device if the controller actually give better performance than that of a real arm? May the artificial hand become even better than a real hand in the future? May we, because of this, be able to strengthen the human limbs by replacing our congenital extremities? For now, these questions may not be of a key issue or seem relevant. But even though our goal is to help people with disabilities, the approach could find new ways of interest in the future.

9 Conclusion

An introduction to wavelets was given, and one can see that a wavelet implementation as a feature extraction algorithm clearly finds important information inherent in a signal. It was expected to be even more significant, and it might be limited due to the fact that we applied it on a rather small window.

EWC-WAVE were found to be the best subset according to both CST and RMSE. Based upon the information obtained from each performance indicator, it is suggested that CST should be carried out as a measure of accuracy on how to map data in the future.

We have indicated the important parameters in a optimal subset. Even though not all combinations were tested, we have gained knowledge of the concept on finding a optimal subset, and its problematic implementation to our system. Reducing the feature set dimensionality, by cancelling among other *EWCL* and *EWPCL*, will considerably ease the complexity of the problem.

Finding an even better subset in the future, where one perhaps uses CST as a performance indicator, may hopefully end up in a useful prosthesis controller where at least two or three of these angles that we have concentrated our research around may be feasible.

Just by adding a simple smoothing function the signal looks much better and becomes more readable. However, this was not the task but it made it easier to evaluate the noisy estimated signals. A filter designed to remove the noise will possibly reduce the suppressed amplitude of the angles.

Overall, it looks as if training a NN will make it possible to adapt a control for a prosthesis based on the EMG signal formerly used to control the real hand, though, the main problem is to achieve a subset, or find a feature, that is able to reflect the true amplitude, which is for now, mostly suppressed.

It is important to keep in mind that the visual feedback the user of a prosthesis obtain, will possibly help the user to correct what is not mapped efficiently.

10 Suggestions for future work

This section explains interesting topics for future work in order for making a real prosthesis. It focus on improvements found in this thesis, Bach (2008) and Fougner (2007).

More hidden nodes More hidden nodes of a NN would possibly gain a feature subsets performance because there will be more neurons to connect and weight the input better from a range of features, to achieve better angle estimations on all angles at once.

Filter input signal To filter, low-pass, a feature would remove noise that might increase the performance of a neural network. It has been thought that neural network would remove noise, but it seems like its weightings does not do a good job. In theory the neural network should act as a filter, and remove the unwanted noise. But for now, there are too much spikes on the output signal.

Post-processing of the signal A more thorough work should be put around the concept of post-processing. Different filter characteristics should be analytically evaluated. Different filters should possibly be applied to the angles.

Filtering of EMG signals Some sort of filtering should be done with the EMG signals. Research should be carried out to find a good way to filter EMG signals without destroying the information lying within the signal. It is important that distinct information in the signal that is extracted as a feature remain intact. Some sort of filtering should be appended on the features as well. The estimated angles seem quite noisy, but they follow the reference quite good. The signal is useless to feed an actuator. One should find a good way to filter this, possibly a low-pass filter that smooths the angle changes.

Implementation of new features The frequency mean feature should be examined. The wavelet features could possibly perform better if a larger time-window is used, giving larger scale possibilities. Considering that Boostani and Moradi's work indicated that the wavelet feature outperformed the others should leave a question mark behind both their and our approach.

Feature subspace Further deploy a subset approach that will consider new subset combinations. Ideally, try all possible combinations by using another approach like the one mention in the discussion.

Calculation time Calculation time to find the different features should be taken into account when trying to find the optimal feature set. A feature subset with long computation time is not directly interesting.

Acquire more data Three test subjects seems to be a bit small data set. To be sure the features will compute good results, more subjects should be used. It has been proposed that 7-11 subjects could give accurate results, and we would not be dependent on luck.

Improvements of pattern recognition methods More pattern recognition methods should be implemented and tested. Many different methods exist, and they might give different results. The best of all the tested should be used. It could be interesting to see how good a polynomial discriminant function would classify compared to the linear discriminant function and neural network. Anders Fougner tested a quadratic discriminant function, but it gave no good results.

Visualisation Some sort of visualisation of the prosthetic hand should be implemented. This will make it easier to try, test and validate the different pattern recognition and proportional control methods, before implementation on a real prosthesis.

Genetic programming Genetic programming should be tested as a machine learning technique. It is an evolutionary algorithm-based methodology inspired by biological evolution. This could improve the classification of features.

Reconsider the sample rate It is not certain that the EMG and VICON data was recorded as what was first thought. If a higher samplings frequency were used, it may give us additional information especially regarding the wavelet features. In addition it is not given that the best resolution is 20Hz even though a prosthesis is not able to preform movements as fast as 10Hz. It could be an idea to investigate if a higher frequency may give better results.

Time delay from EMG to motion There is a natural delay from muscle force (and EMG signals) to motion. Muscle force affects acceleration, and from acceleration to position there will be a 180 degrees phase shift causing a small delay. This is a knowledge that could be exploited in the pattern recognition procedures (Fougner, 2007)

Improvements in angle calculation More accurate angles of the upper limb can be calculated with the use of rotation matrices, based on defined coordinate systems for ulna and the palm. The rotation matrix from ulna to the palm would actually contain all the three important angles for wrist flexion/extension, radial/ulnar deviation and pronation/supination (Stavdahl (2002), on p.24). These will be extracted from the rotation matrix using inverse kinematics. This method may give more accurate angles than the method used in this thesis. The finger flexion/extension may also be calculated in a similar way, but the method used in this thesis seems good enough for the purpose. (Fougner, 2007)

Choice of proportional control method If we manage to design pattern recognition methods that produce a usable output for proportional control, we should try, and test, several different methods for proportional control. Control of position, velocity, force and mechanical impedance are some possibilities, and there are other existing methods that could be interesting for the control of an upper-limb prosthesis.

11 Bibliography

Aha, D. W. and Bankert, R. L. (1995), 'A comparative evaluation of sequential feature selection algorithms', pp. 1–7.

Amara Graps (2004), *An introduction to wavelets*. [Accessed 23. may 2009].
URL: *http://www.amara.com/IEEEwave/IW_wave_vs_four.html*

Avrim L. Blum and Pat Langley (1997), 'Selection of relevant features and examples in machine learning', **97**, 245–271.

Bach, P. (2008), 'Pattern recognition and estimation for the control of a myoelectric prosthesis'.

Blum, A. L. and Langley, P. (1997), 'Selection of relevant features and examples in machine learning', *Artif. Intell.* **97**(1-2), 245–271.

Bock, O. (2008), *Arm Prosthesis*, Otto Bock. [Accessed 29. sept 2008].
URL: *http://www.ottobock.com/cps/rde/xchg/ob_com_en/hs.xsl/14183.html*

Boostani, R. and Moradi, M. H. (2003), 'Evaluation of the forearm emg signal features for the control of a prosthetic hand'.

Bruce Hawkins (1999), *Wavelets*. [Accessed 23. may 2009].
URL: *http://www.science.smith.edu/departments/Physics/fstaff/bhawkins/wavlets/sigma_xi.htm*

C. Valens (2004), *A Really Friendly Guide To Wavlets*. [Accessed 31. may 2009].
URL: *http://pagesperso-orange.fr/polyvalens/clemens/wavelets/wavelets.html*

D. Joshi, K. Kandpal and S. Anand (2008), 'Feature evaluation to reduce false triggering in threshold based emg prosthetic hand'. Biomed 2008, Preceedings 21, 769-772.

D. Wettschereck and D. W. Aha (1995), 'Weighting features'.
URL: *http://citeseer.ist.psu.edu/193130.html*

Dana Mackenzie (2001), 'Wavelets: Seeing the forest and the trees'.
URL: *http://www.beyonddiscovery.org/content/view.txt.asp?a=1952*

David A. Rosenbaum (1990), *Human Motor Control*, Academic Press.

Derek Goring (2006), *Orthogonal Wavelet Decomposition*. [Accessed 23. april 2009].
URL: *http://www.tideman.co.nz/Salalah/OrthWaveDecomp.html*

D.K. Kumar,N.D. Pah & A. Bradley (2003), *Wavelet Analysis of Surface Electromyography to Determine Muscle Fatigue*, IEEE.

E. Garcia (2006), *Cosine Similarity and Term Weight Tutorial*. [Accessed 02. june 2009].
URL: *http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html*

Eugeniusz Gatnar (2006), 'A wrapper feature selection method for combined tree-based classifiers'.
URL: *http://www.springerlink.com/content/r176556m765911r1/*

F. Chaplais (1998), 'A wavelet tour of signal processing by stephane mallat'.
URL: *http://cas.ensmp.fr/~chaplais/wavetour_presentation/*

F. Qiao & R. Milam (2005), *Moments And Vanishing Wavelet Moments*. [Accessed 23. may 2009].
URL: *http://cnx.rice.edu/content/m11156/latest/?format=pdf*

Fougner, A. (2007), Proportional myoelectric control of a multifunction upper-limb prosthesis, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

G. Wang, Z. Wang, W. Chen and J. Zhuang (2006), 'Classification of surface emg signals using optimal wavelet packet method based on daviesbouldin criterion'.
URL: *http://www.springerlink.com/content/78282k5ntg2k07p2/*

Gray, H. (1918), *Anatomy of the Human Body*, 20th edn, Lea & Febiger, Philadelphia, US.
URL: *http://www.bartleby.com/107/*

Ian Kaplan (2002), *The Wavelet Packet Transform*. [Accessed 9. may 2009].
URL: *http://www.bearcave.com/misl/misl_tech/wavelets/packet/index.html*

J. Kilby & H. G. Hosseini (2004), 'Wavelet analysis of surface electromyohraphy signals'.

J. V. Basmajian & C. J. De Luca (1985), *Muscles Alive Their Functions Revealed by Electromyography 5th Edition*, Williams & Wilkins.

Jani Huhtanen (2005), *wavelets - vanishing moments - what are they?* [Accessed 23. may 2009].
URL: *http://dsprelated.com/showmessage/46547/1.php*

John G.H., Kohavi R., Pfleger K. (1994), 'Irrelevant features and subset selection problem'.

John W. Kimball (2008), *Kimball's Biology Pages*. [Accessed 02. oct 2008].
URL: *http://nmrc.bu.edu/tutorials/motor_units/index.html*

Jun Yang and Yue-Peng Li (2006), 'Orthogonal relief algorithm for feature selection', **4113/2006**, 227–234.
URL: *http://www.springerlink.com/content/r667p07q2561rl11/*

K. Englehart, B. Hudgins, P.A. Parker, M. Stevenson (1999), 'Classification of the myoelectric signal using time-frequency based representations'. Medical Engineering & Physics 21 (1999) 431-438.

Kira, Kenji and Rendell, Larry A. (1992), 'A practical approach to feature selection', pp. 249–256.
URL: *http://portal.acm.org/citation.cfm?id=142034*

L. Brechet,MF. Lucas, C. Doncarli & D. Farina (2007), *Compression of Biomedical Signals With Mother Wavelet Optimization and Best-Basis Wavelet Packet Selection*, IEEE.

Luscinia (2007), *Bioacoustics for field recordings*. [Accessed 11. dec 2008].
URL: *http://luscinia.sourceforge.net/page26/page35/page35.html*

Mahyar Zardoshti-Kermani, Bruce C. Wheeler (1995), 'Emg feature evaluation for movement control of uper extremity prostheses'. IEEE Transaction on Rehabilitation engineering, Vol 3, No. 4.

Mallat, Stéphane (1999), *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*, Academic Press.
URL: *http://dx.doi.org/wavetour*

Martha Flanders (2002), *Choosing a wavelet for single-trial EMG*, Elsevier.

Midtgaard, T. (2006), Advanced myoelectric control of a multifunction upper-limb prosthesis, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Muzumdar, A. (2004), *Powered Upper Limbs Prostheses Control, Implementation and Clinical Application*, Springer.

NeuroMuscular Research Center (2002), *Motor Unit Presentation*. [Accessed 02. oct 2008].
URL: *http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/M/Muscles.html*

Parker, P. A., Englehart, K. B. and Hudgins, B. S. (2004), Control of powered upper limb prostheses, *in* 'Electromyography: Physiology, Engineering, and Noninvasive Applications', Institute for Electrical and Electronics Engineers, Inc., New Brunswick, Canada, chapter 18, pp. 453–475.

Phil Schniter (2005), *Time-Frequency Uncertainty Principle*. [Accessed 23. may 2009].
URL: *http://cnx.org/content/m10416/2.18/?format=pdf*

Rahmanpour, H. (2008), 'Pattern recognition and estimation for the control of a myoelectric prosthesis'.

Rice University (2000), *The Unknown Vocoder*. [Accessed 31. oct 2008].
URL: *http://www.owlnet.rice.edu/~elec532/PROJECTS00/vocode/*

Robi Polikar (2006), 'Pattern recognition'.
URL: *http://users.rowan.edu/~polikar/RESEARCH/PUBLICATIONS/wiley06.pdf*

Seeley, Stephens, Tate (2007), *Essentials of Anatomy & Physiology Sixth edition*, McGraw-Hill.

Stavdahl, Ø. (2002), Three-dimensional rotation statistics and parameter estimation, PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Todd Farrell, Richard F. ff. Weir (2008), 'A comparison of the effects of electrode implantation and targeting on pattern classification accuracy for prosthesis control'.

Wikipedia, The Free Encyclopedia (2008), *Cepstrum*. [Accessed 11. dec 2008].
URL: *http://en.wikipedia.org/wiki/Cepstrum*

Wikipedia, The Free Encyclopedia (2009*a*), *Coherence*. [Accessed 29. may 2009].
URL: *http://en.wikipedia.org/wiki/Coherence*

Wikipedia, The Free Encyclopedia (2009*b*), *Coiflet*. [Accessed 22. may 2009].
URL: *http://en.wikipedia.org/wiki/Coiflet*

Wikipedia, The Free Encyclopedia (2009*c*), *Correlation*. [Accessed 29. may 2009].
URL: *http://en.wikipedia.org/wiki/Correlation_coefficient*

Wikipedia, The Free Encyclopedia (2009*d*), *Daubechies Wavelet*. [Accessed 22. may 2009].
URL: *http://en.wikipedia.org/wiki/Daubechies_wavelet*

Wikipedia, The Free Encyclopedia (2009*e*), *Discrete wavelet transform*. [Accessed 09. may 2009].
URL: *http://en.wikipedia.org/wiki/Discret_wavelet_transform*

Wikipedia, The Free Encyclopedia (2009*f*), *Feature (pattern recognition)*. [Accessed 20. may 2009].
URL: *http://en.wikipedia.org/wiki/Feature_(pattern_recognition)*

Wikipedia, The Free Encyclopedia (2009*g*), *Feature selection*. [Accessed 20. may 2009].
URL: *http://en.wikipedia.org/wiki/Feature_selection*

Wikipedia, The Free Encyclopedia (2009*h*), *Filter bank*. [Accessed 20. may 2009].
URL: *http://en.wikipedia.org/wiki/Filter_bank*

Wikipedia, The Free Encyclopedia (2009*i*), *LTI system theory*. [Accessed 20. may 2009].
URL: *http://en.wikipedia.org/wiki/Linear_time-invariant*

Wikipedia, The Free Encyclopedia (2009*j*), *Short-time fourier transform*. [Accessed 29. may 2009].
URL: *http://en.wikipedia.org/wiki/Short_Time_Fourier_Transform*

Wikipedia, The Free Encyclopedia (2009*k*), *Wavelet*. [Accessed 20. may 2009].
URL: *http://en.wikipedia.org/wiki/Wavelet*

Wikipedia, The Free Encyclopedia (2009*l*), *Wavelet Packets*. [Accessed 22. may 2009].
URL: *http://en.wikipedia.org/wiki/Wavelet_packets*

Wolfram Mathworld (2009*a*), *Convolution*. [Accessed 20. may 2009].
URL: *http://mathworld.wolfram.com/Convolution.html*

Wolfram Mathworld (2009*b*), *Fourier Transform*. [Accessed 9. may 2009].
URL: *http://mathworld.wolfram.com/FourierTransform.html*

Wolfram Mathworld (2009*c*), *Wavelet Packets and Wavelet Packet Transforms*. [Accessed 9. may 2009].
URL: *http://documents.wolfram.com/applications/wavelet/FundamentalsofWavelets/1.5.6.html*

Yijun Sun and Dapeng Wu (2008), 'A relief based feature extraction algorithm'.
URL: *http://www.siam.org/proceedings/datamining/2008/dm08_17_Sun.pdf*

Appendix A Source code from Matlab

Comment: All my programs are tested in `Matlab R2008a` and require the `Neural Network Toolbox` and `Wavelet Toolbox`.

Appendix A.1 `EMGaac.m`

```
%% Calculate average amplitude change
% EMGaac(X,samples,scale,file)
% X:        Matrix containing the data,currenly assumed to contain 8 rows
%           of data. One for each electrode.
% samples:  # samples with new frequency.
% scale:    # times to scale down the signal.
% file:     name of the file, EMG_aac_file.mat
%
% AAC = (1/N)*Sum( |x_i+1 - x_i|)
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2008/12/10 18:00:00 $

function AAC = EMGaac(X,samples,scale,file)
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            i1 = i+1;
            N =(k*scale)-1;
            Xaac = X(t,i:N);
            Xaac1 = X(t,i1:N+1); %x_i+1
            AAC(t,k) = (1/N)*sum(abs(Xaac1-Xaac));
        end
    end
filename = ['../data/EMG20Hz/EMGaac/EMG_aac_' file '.mat'];
save(filename,'AAC');
```

Appendix A.2 EMGaav.m

```
%% Calculate average amplitude value
% EMGaav(X,samples,scale,file)
% X:          Matrix containing the data,currenly assumed
   to contain 8 rows
%             of data. One for each electrode.
% samples:    # samples with new frequency.
% scale:      # times to scale down the signal.
% file:       name of the file, EMG_aav_file.mat
%
% AAV = (1/N)*Sum(x_i)
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2008/12/10 18:00:00 $

function AAV = EMGaav(X,samples,scale,file)
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            i1 = i+1;
            N =(k*scale)-1;
            Xaav = X(t,i:N);
            AAV(t,k) = (1/N)*sum(abs(Xaav));
        end
    end
    filename = ['../data/EMG20Hz/EMGaav/EMG_aav_' file '.
       mat'];
    save(filename,'AAV');
```

Appendix A.3 EMGar.m

```
%% Calculate AR coefficients
% EMGar(X,samples,scale,file)
% X:          Matrix containing the data,currenly assumed to contain 8 rows
%             of data. One for each electrode.
% samples:    # samples with new frequency.
% scale:      # times to scale down the signal.
% file:       name of the file, EMG_ar_file.mat
%
% Using model order p=4 to find the coefficients.
% Returns the coefficients, currently 3.
%
% function myLevinson.m used.
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2008/12/10 18:00:00 $

function [AR1 AR2 AR3] = EMGar(X,samples,scale,file)
    p = 4;
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            N =(k*scale)-1;
            %Xar = levinson(X(t,:),P);
            [Xar,msep,K] = myLevinson(X(t,i:N),p);
            for j=1:p-1
             eval(['AR' num2str(j) '(' num2str(t) ',' num2str(k) ')␣=␣abs(Xar(' num2str(j) '));']);
            end
        end
    end
    for j=1:p-1
        filename = ['../data/EMG20Hz/EMGar' num2str(j) '/EMG_ar' num2str(j) '_' file '.mat'];
        save(filename, ['AR' num2str(j)]);
    end
```

Appendix A.4 `EMGcc.m`

```
%% Calculate cepstral coefficients
% EMGcc(X,samples,scale,file)
% X:          Matrix containing the data,currenly assumed to contain 8 rows
%             of data. One for each electrode.
% samples:    # samples with new frequency.
% scale:      # times to scale down the signal.
% file:       name of the file, EMG_cc_file.mat
%
% CC = sum(rceps(X));
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2008/12/10 18:00:00 $

function CC = EMGcc(X,samples,scale,file)
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            Xcc = rceps(X(t,i));
            CC(t,k) = sum(Xcc);
        end
    end
filename = ['../data/EMG20Hz/EMGcc/EMG_cc_' file '.mat'];
save(filename,'CC');
```

Appendix A.5 `EMGhist.m`

```
%% Calculate histogram
% EMGhist(X,samples,scale,file)
% X:         Matrix containing the data,currenly assumed to contain 8 rows
%            of data. One for each electrode.
% samples:   # samples with new frequency.
% scale:     # times to scale down the signal.
% file:      name of the file, EMG_hist_file.mat
%
% Returns the buckets.
%
% HIST = internal command.
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2008/12/10 18:00:00 $

function [HIST1 HIST2 HIST3 HIST4 HIST5 HIST6 HIST7 HIST8 HIST9] = EMGhist(X,samples,scale,file)
    n = 9;
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            N =(k*scale)-1;
            Xext = X(t,i:N);
            Xhist = hist(Xext,n);
            for j=1:n
                eval(['HIST' num2str(j) '(' num2str(t) ',' num2str(k) ')␣=␣abs(Xhist(' num2str(j) '));']);
            end
        end
    end
    for j=1:n
        filename = ['../data/EMG20Hz/EMGhist' num2str(j) '/EMG_hist' num2str(j) '_' file '.mat'];
        save(filename, ['HIST' num2str(j)]);
    end
```

Appendix A.6 `EMGmyop.m`

```
%% Find energy of wavelet coefficients
```

```
% EMGewc(X,samples,scale)
% X:          Matrix containing the data,currenly
%                 assumed to contain 8 rows
%             of data. One for each electrode.
% samples:    # samples with new frequency.
% scale:      # times to scale down the signal.
%
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.5 $Date: 2009/03/12 16:00:00 $

function EWC = EMGewc(X,samples,scale)
    global wname nMax;
    EWC = zeros([8,length(samples),nMax],'double');
    for k=1:samples
        i = (k-1)*scale+1;
        N =(k*scale);
        for t=1:8;
            [C,L] = wavedec(X(t,i:N),nMax,wname);
            [Ea,Ed] = wenergy(C,L);
            for h=1:length(Ed)
               EWC(t,k,h) = Ed(h);
            end
        end
    end
```

Appendix A.7 EMGewc.m

```
%% Find energy of wavelet coefficients
% EMGewpc(X,samples,scale)
% X:          Matrix containing the data,currenly
%                 assumed to contain 8 rows
%             of data. One for each electrode.
% samples:  # samples with new frequency.
% scale:    # times to scale down the signal.
%
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.7 $Date: 2009/03/27 16:00:00 $

function EWPC = EMGewpc(X,samples,scale)
    global wname nMax;
    EWPC = zeros([8,length(samples),nMax],'double');
    for k=1:samples
        i = (k-1)*scale+1;
        N =(k*scale);
        for t=1:8;
            [T] = wpdec(X(t,i:N),nMax,wname);
            [E] = wenergy(T);
            for h=2:length(E)
                EWPC(t,k,(h-1)) = E(h);
            end
        end
    end
```

Appendix A.8 EMGewpc.m

```
%% Calculate myopulse percentage rate
% EMGmyop(X,samples,scale,file)
% X:          Matrix containing the data,
%             currenly assumed to contain 8 rows
%             of data. One for each electrode.
% samples:    # samples with new frequency.
% scale:      # times to scale down the signal.
%
% MYOP = (1/N)*Sum f(x_i)
% threshold = 50mjuV?.
% Threshold depends on the gain of the electrode etc.
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.5 $Date: 2009/01/21 13:00:00 $

function MYOP = EMGmyop(X,samples,scale)
    thresh = 0.9; %0.15 %50e-06;
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            N =(k*scale);
            Xa = abs(X(t,i:N));
            Xmyop = length(find(Xa>thresh));
            MYOP(t,k) = (1/(N-i))*sum(Xmyop)*100;
        end
    end
```

Appendix A.9 EMGnt.m

```
%% Calculate number of turns
% EMGnt(X,samples,scale,file)
% X:          Matrix containing the data,currenly
%             assumed to contain 8 rows
%             of data. One for each electrode.
% samples:    # samples with new frequency.
% scale:      # times to scale down the signal.
%
% NT = Sum(sgn(x_i-x_i+1)(x_i+2-x_i+1))
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.5 $Date: 2009/02/05 17:00:00 $

function NT = EMGnt(X,samples,scale)
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            i1 = i+1;
            i2 = i+2;
            N =(k*scale)-2;
            N1 =(k*scale)-1;
            N2 =(k*scale);

            Xdiff1 = (X(t,i1:N1)-X(t,i:N));
            Xdiff2 = (X(t,i2:N2)-X(t,i1:N1));
            Xdiff = Xdiff1.*Xdiff2;
            NT(t,k) = sum(sign(Xdiff(1:scale-2)));
        end
    end
```

Appendix A.10 EMGvar.m

```
%% Calculate variance
% EMGvar(X,samples,scale,file)
% X:         Matrix containing the data,currenly assumed to contain 8 rows
%            of data. One for each electrode.
% samples:  # samples with new frequency.
% scale:    # times to scale down the signal.
% file:     name of the file, EMG_var_file.mat
%
% VAR = (1/(N-1))*Sum((x_i)^2)
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2008/12/10 18:00:00 $

function VAR = EMGvar(X,samples,scale,file)
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
          % i1 = i+1;
            N =(k*scale)-1;
            Xvar = X(t,i:N);
            VAR(t,k) = (1/(N-1))*sum(Xvar.^2);
        end
    end
filename = ['../data/EMG20Hz/EMGvar/EMG_var_' file '.mat'];
save(filename,'VAR');
```

Appendix A.11 EMGwamp.m

```
%% Calculate Wilson amplitude
% EMGwamp(X,samples,scale,file)
% X:          Matrix containing the data,
%             currenly assumed to contain 8 rows
%             of data. One for each electrode.
% samples:   # samples with new frequency.
% scale:     # times to scale down the signal.
%
% WAMP = Sum( |x_i - x_i+1|)
% threshold = 50mjuV?.
% Threshold depends on the gain of the electrode etc.
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.5 $Date: 2009/01/21 13:00:00 $

function WAMP = EMGwamp(X,samples,scale)
    thresh = 0.9; %0.35;%4e+03;
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            i1 = i+1;
            N =(k*scale)-1;
            Xwamp = X(t,i:N);
            Xwamp1 = X(t,i1:N+1); %x_i+1
            Xthresh = abs(Xwamp1-Xwamp);
            Xw = find(Xthresh>thresh);
            WAMP(t,k) = length(Xw);
        end
    end
```

Appendix A.12 `EMGwave.m`

```
%% Calculate wavelength
% EMGwave(X,samples,scale,file)
% X:          Matrix containing the data,currenly assumed to contain 8 rows
%             of data. One for each electrode.
% samples:    # samples with new frequency.
% scale:      # times to scale down the signal.
% file:       name of the file, EMG_wave_file.mat
%
% WAVE = Sum f(|x_i - x_i-1|)
% thresh = 50e-06;
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2008/12/10 18:00:00 $

function WAVE = EMGwave(X,samples,scale,file)
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+2;
            i1 = i+1;
            N =(k*scale)-1;
            Xwl = X(t,i:N);
            Xwl1 = X(t,i1:N+1); %x_i-1
            Xwave = abs(Xwl-Xwl1);
            WAVE(t,k) = sum(Xwave);
        end
    end
filename = ['../data/EMG20Hz/EMGwave/EMG_wave_' file '.mat'];
save(filename,'WAVE');
```

Appendix A.13 `EMGzc.m`

```
%% Calculate zero-crossings
% EMGzc(X,samples,scale,file)
% X:          Matrix containing the data,currenly assumed to contain 8 rows
%             of data. One for each electrode.
% samples:    # samples with new frequency.
% scale:      # times to scale down the signal.
% file:       name of the file, EMG_zc_file.mat
%
% ZC = Sum( sgn(-(x_i-x_i+1)))
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2008/12/10 18:00:00 $

function ZC = EMGzc(X,samples,scale,file)
    for k=1:samples
        for t=1:8;
            i = (k-1)*scale+1;
            Xs = sign(X(t,i:k*scale));
            Xdiff = diff(Xs)/2;
            ZC(t,k) = sum(abs(Xdiff));
        end
    end
filename = ['../data/EMG20Hz/EMGzc/EMG_zc_' file '.mat'];
save(filename,'ZC');
```

Appendix A.14 `hpfilter.m`

```
function Y = hpfilter(X,K,F)
%% High-pass filter
% INPUT OPTIONS
% X:     signal array to filter.
% K:     Gain.
% F:     Cut-off frequency.
% High-pass filter to get average to zero.
% This function uses the filtfilt() internal command of matlab.
%
% Per Ferdinand Bach, perferdi@stud.ntnu.no
% $Revision: 1.0 $Date: 2009/01/19 12:00:00 $

%Tuning variables
%K = 0.57;  %Gain.
%F = 0.25;  %Frequency.

w = 2*pi*F; % omega -> rad/s
s = tf('s'); % Create transfer function.
H = (K*s) / (s + w); % Filter structure.
[num,den] = tfdata(H); %we want the numerater and denumerater separated.
num = num{1};
den = den{1};

for a=1:size(X,1)
    eval('Y(a,:)␣=␣filtfilt(den,num,X(a,:));');
    Y(a) = (Y(a));
end
```

Appendix A.15 `myLevinson.m`

```
function [A, MSEP, K] = myLevinson(x,P)
% Input : x-- The input time series x
%                                P-- The order of LP
% Output:
%                               A -- The A parameters;
%                               MSE -- The MSE in each recursion;
%                               K      -- The reflection coefficients in each recursion;
% Downloaded from http://www.owlnet.rice.edu/~elec532/PROJECTS00/vocode/

N=length(x);

R=xcorr(x);
R=R(N:2*N-1);   % shift it to make it R(0)-R(N-1);

% To compute the p-order prediction, we only need R(0)-R(p);


Apar=zeros(P,P);          % To store the A parameters in each recursion;
MSEp=zeros(P);                        % To store the MSE(Rou(f,p)) in each recursion;
Kpar=zeros(P);            % To store the PARCOR coef in each recursion;

%compute the initializations
Apar(1,1)=-R(2)/R(1);
MSE(1)=R(1)-R(2)^2/R(1);
Kpar(1)=Apar(1,1);
%loop to compute the parameters in 2--> P orders;
for p=1:P-1
    Deltap=R(p+2);
    for m=1:p
        Deltap=Deltap+Apar(p,m)*R(p+2-m);
    end
    Kpar(p+1)=-Deltap/MSE(p);
    for m=1:p
        Apar(p+1,m)=Apar(p,m)+Kpar(p+1)*Apar(p,p+1-m);
    end
    Apar(p+1,p+1)=Kpar(p+1);
    MSE(p+1)=MSE(p)*(1-Kpar(p+1)^2);
end

A=Apar;             % A P*P matrix;
MSEP=MSE; % 1*P vector;
K=Kpar;  % 1*P vector;

% [A1,E1,K1]=levinson(R,P); % Here for debuging;
return;
```

Appendix A.16 `smoothing.m`

```
function yout = smoothing(yin,N)

% SMOOTH.M: Smooths vector data.
%                         YOUT=SMOOTH(YIN,N) smooths the data in YIN using a running mean
%                         over N previous successive point.

if nargin<2, error('Not␣enough␣input␣arguments!'), end

[rows,cols] = size(yin);
if min(rows,cols)~=1, error('Y␣data␣must␣be␣a␣vector!'), end
if length(N)~=1, error('N␣must␣be␣a␣scalar!'), end


yin = (yin(:))';
l = length(yin);

yout = zeros(1,l);
yout(1) = yin(1);
for i=2:N
    yout(i) = mean([yout(i-1) yin(i)]);
end

for j=N:l
    yout(j) = mean(yin((j-N+1):j));
end

if size(yout)~=[rows,cols], yout = yout'; end
```

Appendix A.17 firstOrderEstimation.m

```
function [F_e,Fval_e,Ftest_e] = firstOrderEstimation(Tdata,Tval,Ttest,N,Xdata,Fdata,Xvaldata,Fvaldata,Xtestdata,Ftestdata)
% FIRSTORDERESTIMATION  Calculates first-order parameters W to estimate the
% function F from signals in Xdata using least-squares estimation.
% There are N signals and T time steps.
%
% For every time step we have
% F_e(X) = X'*W+ w0
%
% where
% X = (x1,x2,...,xN), W = (w1,w2,...,wN), w0 is a scalar treshold value
%
% We rewrite it as
% F_e(X) = X'*W+ W_0
%
% where
% X = (1,x1,x2,...,xN), W = (w0,w1,w2,...,wN)
%
% W is found such that we minimize V = 0.5*sum( (F_e(X) - F(X))^2 )
% where X and f(X) are known vectors.
%
% F_e, Fval_e and Ftest_e are then found using W.
%
% Finally, the function returns F_e, which is the estimate of F,
% Fval_e, which is the estimate of Fval, and Ftest_e, which is
% the estimate of Ftest.
%
% See also SECONDORDERESTIMATION, NEURALNETWORK, MAKEDATA

% Anders Fougner, anderfo@stud.ntnu.no
% $Revision: 2.0 $Date: 2007/05/20 22:13:00 $

%% Estimation 1st order
```

```
% Performing least-squares estimation for each of N signals

% Adjust X
X = [ones(1,Tdata); Xdata]; % add 'ones' for calculating W_0 parameters

%Calculate the optimal W_big containing W, W_0
W = X'\Fdata';

% Calculate F_e(X), the estimate of F(x)
F_e = W'*X;

%% Simulate using validation data
% Adjust Xval
Xval = [ones(1,Tval); Xvaldata]; % add 'ones' for calculating W_0 parameters
% Calculate Fval_e(X), the estimate of Fval(x)
Fval_e = W'*Xval;

%% Simulate using test data
% Adjust Xtest
Xtest = [ones(1,Ttest); Xtestdata]; % add 'ones' for calculating W_0 parameters
% Calculate Fval_e(X), the estimate of Fval(x)
Ftest_e = W'*Xtest;
```

Appendix A.18 `neuralNetwork.m`

```
function [F_e,Fval_e,Ftest_e] = neuralNetwork(Tdata,Tval,Ttest,Nx,Nf,Xdata,Fdata,Xvaldata,Fvaldata,TfChoice,HiddenNodes,Xtestdata,Ftestdata)
% FIRSTORDERESTIMATION  Uses neural network theory to find a relation
% between a signal matrix X (for N signals and T time steps) and a function
% matrix F.
% Then it generates an estimate F_e of the function matrix F, using the
% neural net.
%
% Finally, the function returns F_e, which is the estimate of F, and
% according results for validation and test sets.
%
%
% See also MAKEDATA, FIRSTORDERESTIMATION, SECONDORDERESTIMATION

% Anders Fougner, anderfo@stud.ntnu.no
% $Revision: 2.0 $Date: 2007/05/20 22:13:00 $


% "Normalization/standardization" of output data
[Fdata2,FdataS] = mapminmax(Fdata);
[Fvaldata2,FvaldataS] = mapminmax(Fvaldata);
[Ftestdata2,FtestdataS] = mapminmax(Ftestdata);

% Choice of transfer function in the nodes/synapses of the NN
Tfs(1:HiddenNodes,1:Nf) = {TfChoice}; % tansig is default
disp(['Using␣' num2str(HiddenNodes) '␣nodes␣in␣the␣hidden␣layer.']);

%% Adjust X to 2nd degree
% X = Xdata;
% for k=Nx:-1:1
%     X = [X;(Xdata(1:k,1:Tdata).*Xdata(Nx-k+1:Nx,1:Tdata))];
% end
```

```
%% Adjust Xval in the same way as X was adjusted
% Xval = Xvaldata;
% for k=Nx:-1:1
%     Xval = [Xval;(Xvaldata(1:k,1:Tval).*Xvaldata(Nx-k+1:Nx,1:Tval))];
% end
%% Adjust Xtest in the same way as X was adjusted
% Xtest = Xtestdata;
% for k=Nx:-1:1
%     Xtest = [Xtest;(Xtestdata(1:k,1:Ttest).*Xtestdata(Nx-k+1:Nx,1:Ttest))];
% end

% Generate a neural network
net = newff(minmax(Xdata),[HiddenNodes Nf],Tfs);

% Make validation and testing structures
VV.P = Xvaldata;
VV.T = Fvaldata2;
TV.P = Xtestdata;
TV.T = Ftestdata2;

% Train the neural network, but do not print error messages
net.trainParam.show = NaN;
[net,tr]=train(net,Xdata,Fdata2,[],[],VV,TV);

% Simulate Xdata and Xvaldata in the NN to estimate F and Fval
F_e = sim(net,Xdata);
Fval_e = sim(net,Xvaldata);
Ftest_e = sim(net,Xtestdata);

% "Denormalization/unitization" of output data
F_e = mapminmax('reverse',F_e,FdataS);
Fval_e = mapminmax('reverse',Fval_e,FvaldataS);
Ftest_e = mapminmax('reverse',Ftest_e,FtestdataS);
```

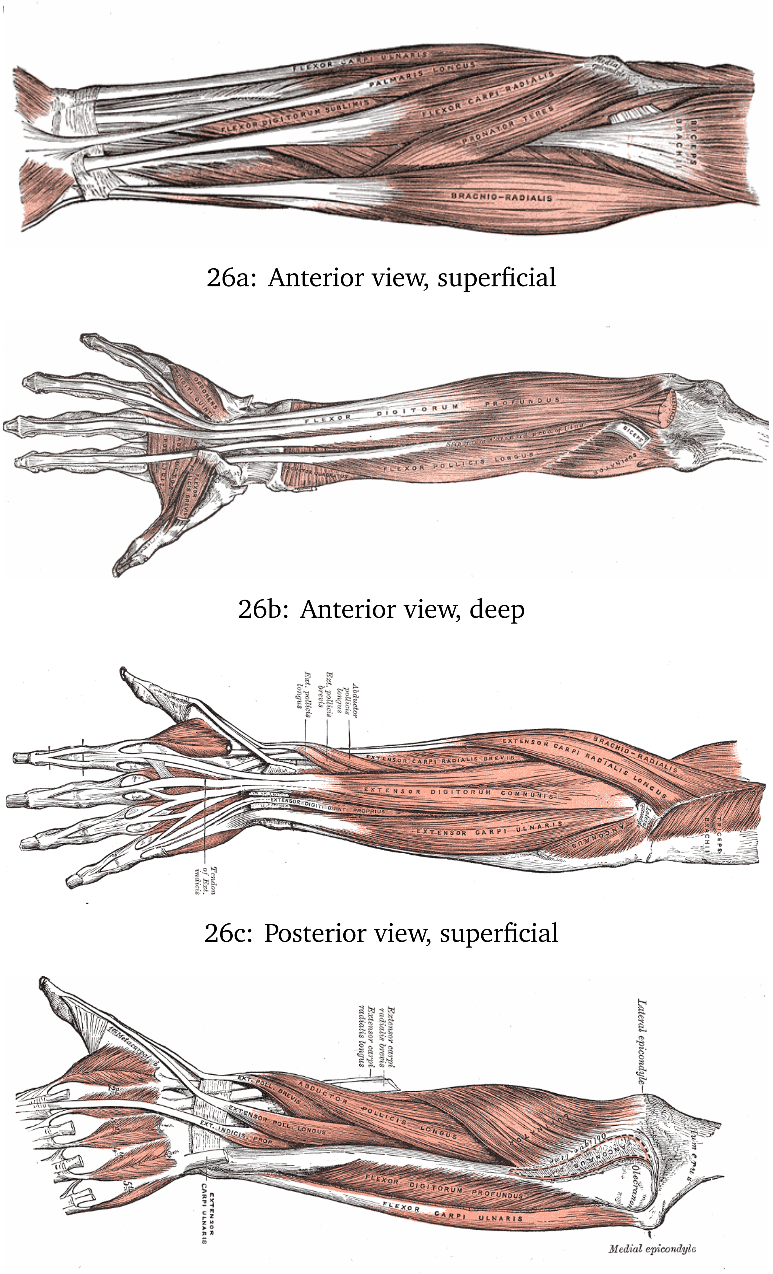
Appendix B Tables and Figures

Appendix B.1 Selection Algorithms

Type	Definition
RELIEF	Feature weighting algorithm, simple and effective implementation of a convex optimization problem will avoid any exhaustive or heuristic search. (Yijun Sun and Dapeng Wu, 2008)
FOCUS	Exhaustive search, stops when the smallest sufficient subset is found. (Kira, Kenji and Rendell, Larry A. , 1992)
ID3	Heuristic search, a kind of sequential forward search (incrementally selects the best feature with most information gain). Builds a decision tree. (Kira, Kenji and Rendell, Larry A. , 1992)
OBLIVION	Backward elimination. Starts with all features and removes the one that leads to the greatest improvement in the set.(Blum and Langley, 1997)
BEAM	A Framework for sequential selection/elimination. (Aha and Bankert, 1995)
PRESET	Filter algorithm that uses theory of *rough sets* to heuristically rank the features. (Robi Polikar, 2006)

Table 14: Different search algorithms.

Appendix B.2 Forearm muscles



26a: Anterior view, superficial

26b: Anterior view, deep

26c: Posterior view, superficial

26d: Posterior view, deep

Figure 26: Forearm muscles, illustrations from (Gray, 1918)

Appendix B.3 Forearm muscles table

Movement	Muscles in prioritized order
Finger flexion	Flexor digitorum superficialis/sublimis, Opponens pollicis[2], Flexor pollicis brevis[2], Abductor pollicis brevis[2]
Finger extension	Extensor digitorum, Flexor carpi radialis, Flexor carpi ulnaris
Pronation	Pronator teres, Pronator quadratus[1]
Supination	Supinator
Wrist flexion	Flexor carpi ulnaris, Flexor carpi radialis, Palmaris longus
Wrist extension	Extensor carpi radialis brevis & longus[3], Extensor carpi ulnaris, Extensor digitorum
Radial deviation	Extensor carpi radialis brevis & longus[3], Flexor carpi radialis
Ulnar deviation	Flexor carpi ulnaris, Extensor carpi ulnaris

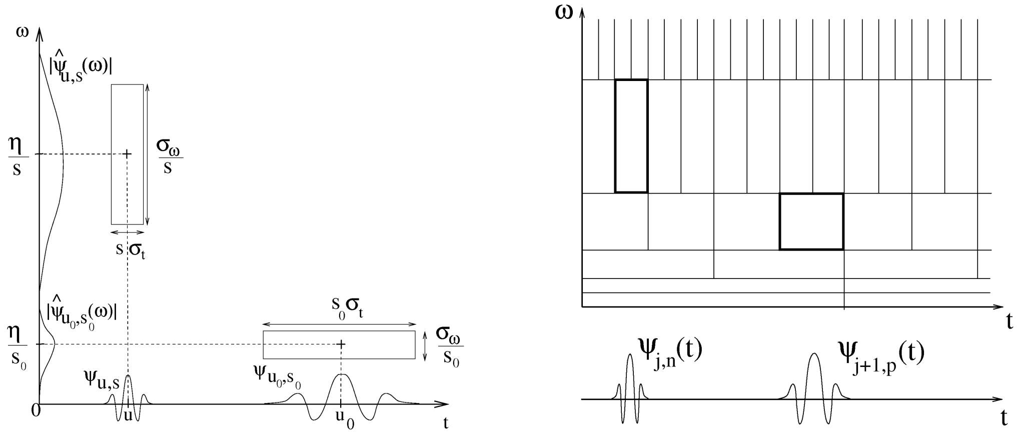
Table 15: Relevant muscles for specific movements (Fougner, 2007)

[1] These muscles probably lay too deep to be measured with EMG signals

[2] These muscles are placed inside the hand and are irrelevant in amputees

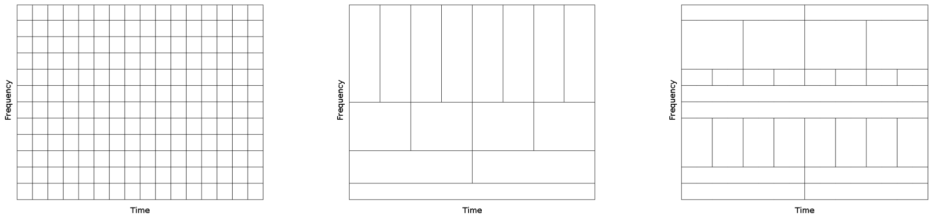
[3] Extensor carpi radialis brevis & longus are usually considered as one single muscle, since they are not distinguished easily when placing the electrodes (Fougner, 2007)

Appendix B.4 Heisenberg boxes and time-frequency tiling of STFT,WT and WPT



27a: Heisenberg boxes of two wavelets. 27b: Tilig of the time-frequency plane

Figure 27: Heisenberg Boxes. Smaller scales decrease the time spread, but increases the frequency support.(F. Chaplais, 1998)



28a: STFT resolution 28b: WP resolution 28c: WPT resolution

Figure 28: Time-frequency tiling of a Short Time Fourier, Wavelet and Wavelet Packet Transform. The tiling of STFT and WT is fixed, while WPT tiling can be adopted to suit a particular application (K. Englehart, B. Hudgins, P.A. Parker, M. Stevenson, 1999)

Appendix B.5 Feature subsets

Num	Feature Name	Num	Feature Name
17	ar-cc-proc-wamp-wave	59	ar-ewpc-wamp
18	ar-cc-ewc-nt-proc-wamp-wave	60	ar-ewpc-myop
19	cc-nt-var	61	ar-ewpc-var
20	ewc-proc-wave	62	ar-proc-wave
21	ewc-wamp	63	ewpcl-hist-myop-nt-proc
22	cc-ewcl-hist-myop	64	ar-ewcl-ewpcl
23	ewpc-myop-var-wave	65	var-wamp
24	aac-aav	66	wamp-wave
25	aac-ewc-ewpc-myop	67	wave-zc
26	aav-ewc-wamp	68	var-zc
27	aav-ar-cc-ewc	69	wamp-zc
28	ewc-ewpcl	70	myop-var
29	ewc-ewcl-ewpc-ewpcl	71	myop-zc
30	cc-ewc-nt	72	hist-proc
31	ewpc-hist-var	73	proc-wamp
32	ewpc-var	74	proc-wave
33	proc-zc	75	hist-zc
34	ewcl-ewpcl-myop-zc	76	hist-nt-zc
35	ewc-ewcl-wave	77	hist-var
36	ewc-var-wamp	78	ewpcl-proc
37	ewpc-ewpcl-proc	79	ewc-proc
38	myop-proc-var	80	ewc-wave
39	cc-ewpcl-zc	81	ewc-var
40	cc-proc-var	82	ewc-hist
41	ar-nt-var	83	ewcl-hist
42	aac-aav-ar	84	ewcl-ewpcl
43	cc-ewc-ewcl	85	myop-nt
44	ewpc-ewpcl-hist	86	cc-proc
45	myop-nt-proc	87	cc-ewcl-nt
46	var-wamp-wave	88	cc-wamp
47	myop-wamp-zc	89	cc-wave
48	cc-ewc	90	aav-ewpcl
49	ar-ewc	91	aac-ewpc
50	ar-ewcl	92	ar-cc-ewc
51	cc-ewcl	93	cc-myop-proc
52	cc-ewpc	94	ewpc-hist-zc
53	ar-ewpc	95	ewc-hist-zc
54	ewpc-myop-nt	96	ar-hist
55	ewpcl-myop-var	97	ar-wave
56	ewc-zc	98	ewpcl-wamp
57	aac-ewc-zc	99	aav-cc-hist
58	ar-ewpc-wave	100	aac-cc-hist

Table 16: Feature combinations

Appendix B.6 LF RMSE Unfiltered

(a) RMSE, LF, Val, Unfiltered

Feature	RMSE of ROM [%]			
Num	WFE	PS	FFE	RUD
18	15.52	18.25	23.64	13.15
17	15.84	18.52	23.63	13.28
59	16.17	19.27	24.38	13.25
13	22.79	29.01	25.12	15.77

(b) RMSE, LF, Test, Unfiltered

Feature	RMSE of ROM [%]			
Num	WFE	PS	FFE	RUD
18	22.08	32.52	47.38	16.21
17	22.32	32.86	47.37	16.06
59	21.49	28.50	35.09	16.54
13	125.16	142.44	366.09	49.24

Table 17: Table showing the 3 best features calculated by LF without post filtering. Sorted on best validation data. Validation data shown in 16(a), and Test data shown in 16(b)

Appendix B.7 NN RMSE Unfiltered

(a) RMSE, NN, Val, Unfiltered

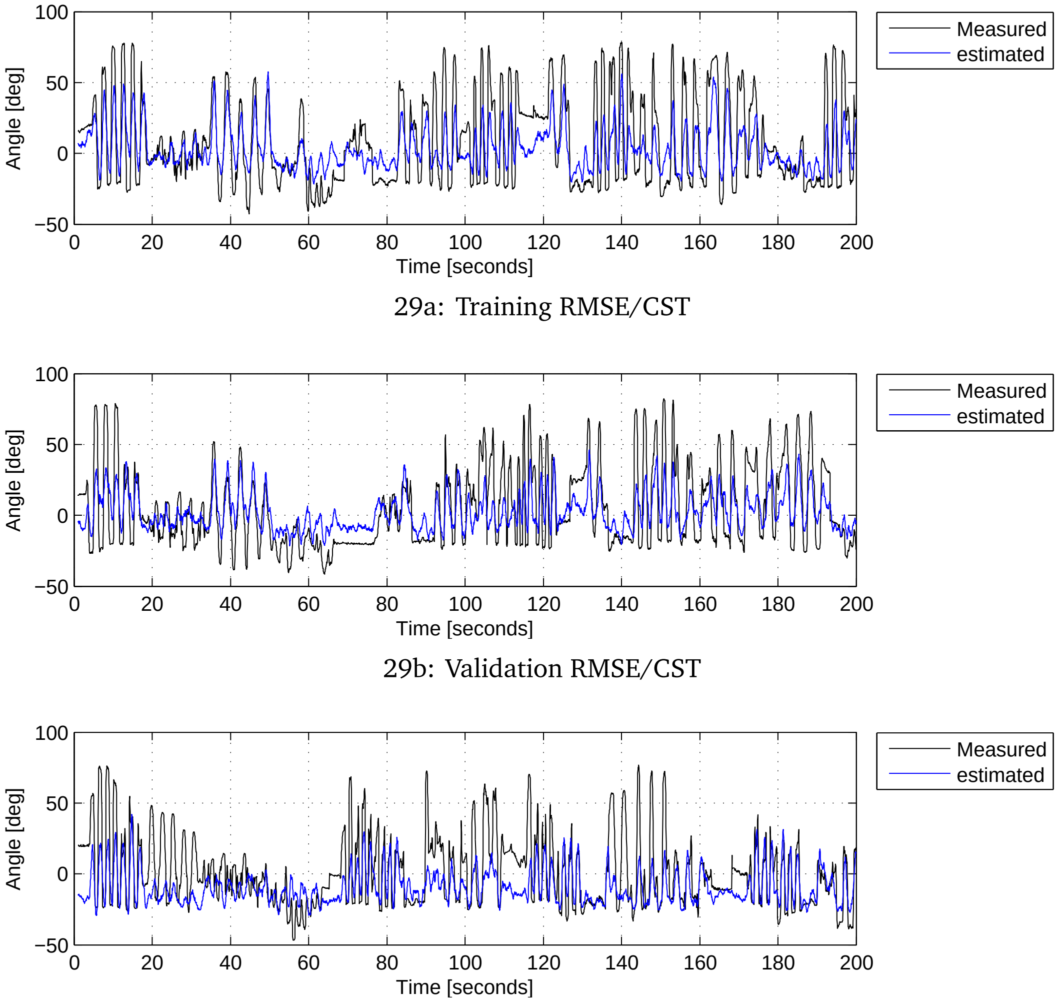
Feature	RMSE of ROM [%]	
Num	WFE	FFE
17	13.19	26.41
62	13.23	25.62
18	13.25	25.18
9	20.77	26.62

(b) RMSE, NN, Test, Unfiltered

Feature	RMSE of ROM [%]	
Num	WFE	FFE
17	21.55	33.82
62	21.06	34.20
18	21.49	34.43
9	22.23	32.84

Table 18: Table showing the 3 best features calculated by LF without post filtering. Sorted on best validation data. Validation data shown in 16(a), and Test data shown in 16(b)

Appendix B.8 Best set NN FFE RMSE



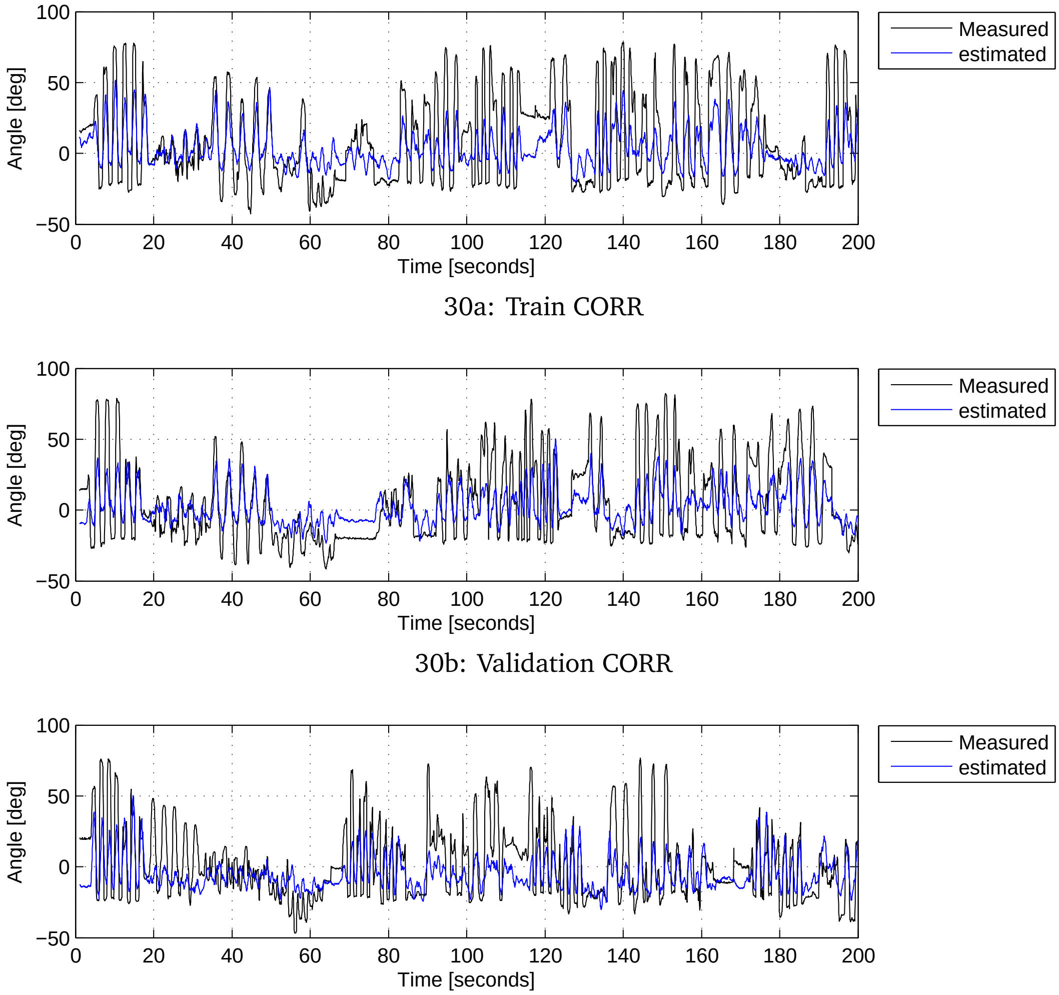
29a: Training RMSE/CST

29b: Validation RMSE/CST

29c: Testing RMSE/CST

Figure 29: Best subset according to RMSE (showing subset 80 for subject 1.)

Appendix B.9 `Best set NN FFE CORR`



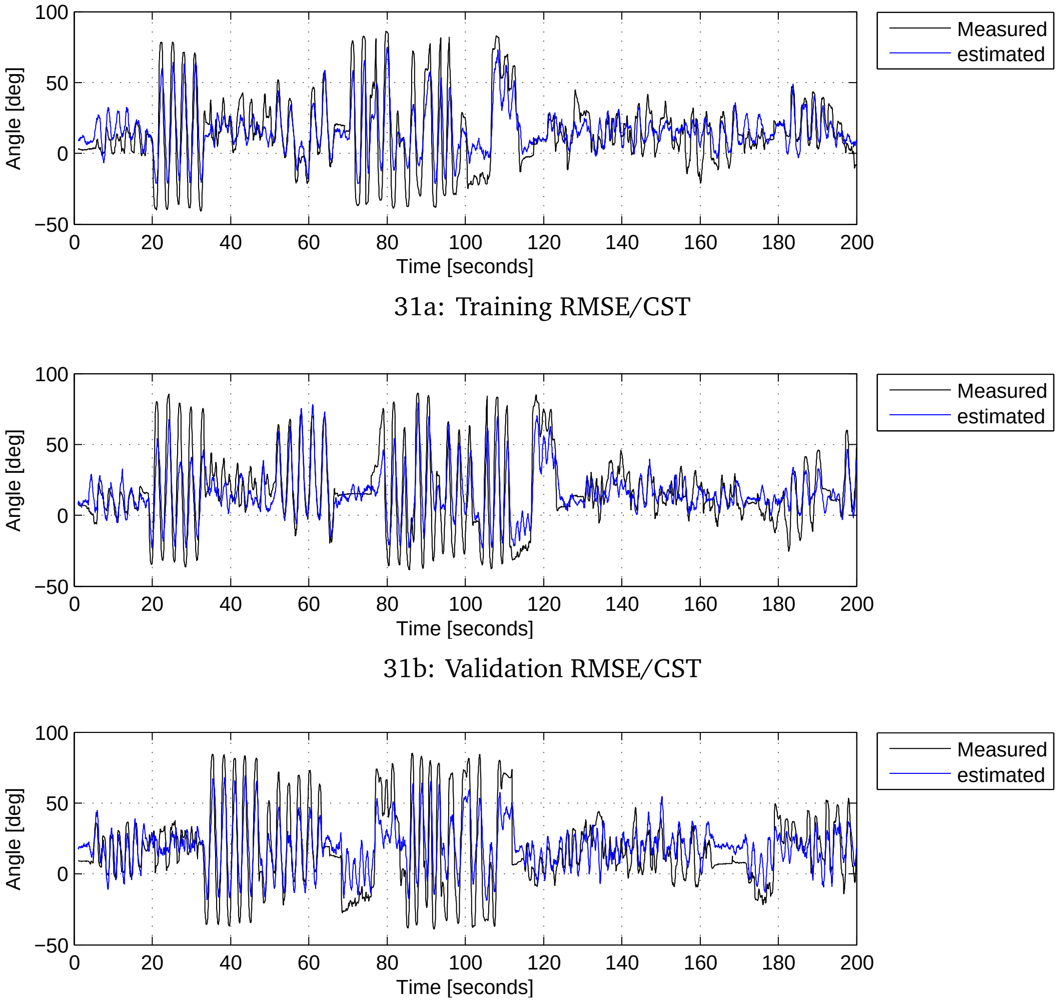
30a: Train CORR

30b: Validation CORR

30c: Testing CORR

Figure 30: Best subsets according to CORR (showing subset 66 for subject 1.)

Appendix B.10 Best set NN WFE RMSE



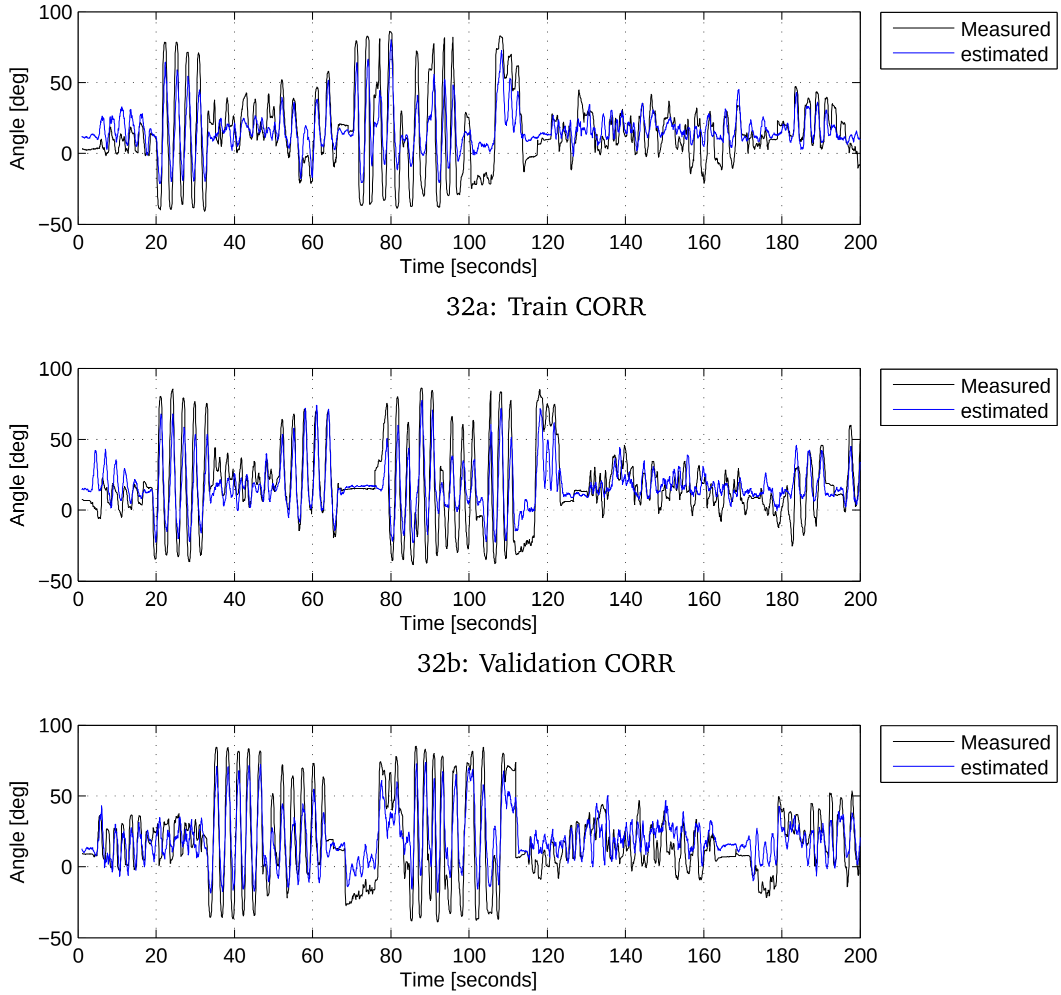
31a: Training RMSE/CST

31b: Validation RMSE/CST

31c: Testing RMSE/CST

Figure 31: Best subset according to RMSE (showing subset 80 for subject 1.)

Appendix B.11 `Best set NN WFE CORR`



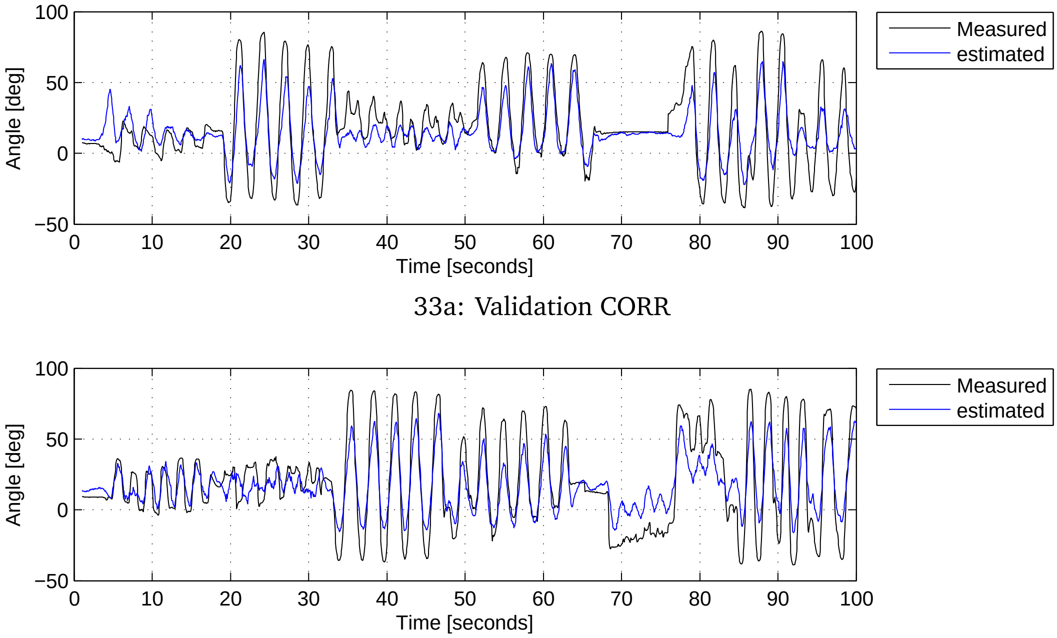
32a: Train CORR

32b: Validation CORR

32c: Testing CORR

Figure 32: Best subsets according to CORR (showing subset 66 for subject 1.)

Appendix B.12 `AAC NN WFE CORR`



33a: Validation CORR

33b: Testing CORR

Figure 33: AAC second best subset according to CORR (subject 1.)

Appendix B.13 Best Subsets WFE RMSE

Feature		CST	
Num	Name	WFE	FFE
80	ewc-wave	12.99	23.59
42	aac-aav-ar	13.00	24.13
35	ewc-ewcl-wave	13.01	23.91
57	aac-ewc-zc	13.01	25.74
97	ar-wave	13.04	25.14
67	wave-zc	13.13	23.97
24	aac-aav	13.13	25.03
21	ewc-wamp	13.16	24.85
66	wamp-wave	13.22	24.63
18	ar-cc-ewc-nt-proc-wamp-wave	13.23	23.71
23	ewpc-myop-var-wave	13.28	23.18
58	ar-ewpc-wave	13.29	24.20
17	ar-cc-proc-wamp-wave	13.31	25.10
47	myop-wamp-zc	13.31	25.94
25	aac-ewc-ewpc-myop	13.34	25.23
26	aav-ewc-wamp	13.37	24.67
62	ar-proc-wave	13.37	24.33
36	ewc-var-wamp	13.39	26.22
91	aac-ewpc	13.42	24.25
46	var-wamp-wave	13.47	25.25
27	aav-ar-cc-ewc	13.48	25.20
20	ewc-proc-wave	13.49	24.15
15	wave	13.50	24.90
89	cc-wave	13.53	25.38
69	wamp-zc	13.55	23.84
59	ar-ewpc-wamp	13.58	24.23
1	aac	13.58	25.35
100	aac-cc-hist	13.60	24.42
63	ewpcl-hist-myop-nt-proc	13.83	23.96
60	ar-ewpc-myop	13.87	25.56
79	ewc-proc	13.89	25.02
98	ewpcl-wamp	13.94	25.58
34	ewcl-ewpcl-myop-zc	13.94	25.14
65	var-wamp	13.96	25.70
41	ar-nt-var	13.99	26.14
88	cc-wamp	14.01	25.93
14	wamp	14.05	24.37
71	myop-zc	14.07	24.59
45	myop-nt-proc	14.09	25.34
37	ewpc-ewpcl-proc	14.10	24.52

Table 19: Listing the 40 best subsets using RMSE as a performance indicator.

Appendix B.14 `Best Subsets WFE CORR`

Feature		CST	
Num	Name	WFE	FFE
66	wamp-wave	0.863	0.523
15	wave	0.861	0.525
1	aac	0.859	0.510
80	ewc-wave	0.858	0.529
14	wamp	0.858	0.497
89	cc-wave	0.857	0.527
35	ewc-ewcl-wave	0.857	0.531
21	ewc-wamp	0.856	0.509
67	wave-zc	0.855	0.529
57	aac-ewc-zc	0.853	0.525
91	aac-ewpc	0.853	0.493
46	var-wamp-wave	0.851	0.509
100	aac-cc-hist	0.851	0.529
98	ewpcl-wamp	0.850	0.490
36	ewc-var-wamp	0.850	0.496
47	myop-wamp-zc	0.849	0.525
88	cc-wamp	0.849	0.503
69	wamp-zc	0.848	0.502
25	aac-ewc-ewpc-myop	0.848	0.516
97	ar-wave	0.848	0.532
42	aac-aav-ar	0.847	0.529
26	aav-ewc-wamp	0.847	0.510
23	ewpc-myop-var-wave	0.846	0.509
58	ar-ewpc-wave	0.846	0.505
24	aac-aav	0.846	0.522
59	ar-ewpc-wamp	0.845	0.486
65	var-wamp	0.844	0.474
27	aav-ar-cc-ewc	0.838	0.487
34	ewcl-ewpcl-myop-zc	0.836	0.469
71	myop-zc	0.834	0.479
54	ewpc-myop-nt	0.827	0.464
18	ar-cc-ewc-nt-proc-wamp-wave	0.827	0.528
60	ar-ewpc-myop	0.824	0.472
17	ar-cc-proc-wamp-wave	0.822	0.526
62	ar-proc-wave	0.820	0.517
22	cc-ewcl-hist-myop	0.819	0.467
30	cc-ewc-nt	0.819	0.456
20	ewc-proc-wave	0.818	0.509
41	ar-nt-var	0.818	0.455
48	cc-ewc	0.817	0.451

Table 20: Listing the 40 best subsets using CORR as a performance indicator.

Appendix B.15 Best Subsets WFE CST

Feature		CST	
Num	Name	WFE	FFE
80	ewc-wave	0.900	0.558
35	ewc-ewcl-wave	0.900	0.567
57	aac-ewc-zc	0.899	0.537
67	wave-zc	0.899	0.550
42	aac-aav-ar	0.899	0.549
66	wamp-wave	0.899	0.540
21	ewc-wamp	0.898	0.538
97	ar-wave	0.897	0.552
15	wave	0.896	0.549
24	aac-aav	0.896	0.545
25	aac-ewc-ewpc-myop	0.895	0.544
91	aac-ewpc	0.895	0.545
58	ar-ewpc-wave	0.895	0.534
1	aac	0.894	0.533
47	myop-wamp-zc	0.894	0.452
26	aav-ewc-wamp	0.894	0.541
89	cc-wave	0.894	0.533
23	ewpc-myop-var-wave	0.893	0.553
36	ewc-var-wamp	0.893	0.539
46	var-wamp-wave	0.892	0.542
100	aac-cc-hist	0.892	0.539
69	wamp-zc	0.891	0.530
18	ar-cc-ewc-nt-proc-wamp-wave	0.890	0.516
59	ar-ewpc-wamp	0.890	0.524
27	aav-ar-cc-ewc	0.889	0.531
17	ar-cc-proc-wamp-wave	0.888	0.536
62	ar-proc-wave	0.887	0.552
14	wamp	0.886	0.525
20	ewc-proc-wave	0.886	0.550
98	ewpcl-wamp	0.886	0.492
65	var-wamp	0.885	0.525
88	cc-wamp	0.884	0.519
34	ewcl-ewpcl-myop-zc	0.883	0.512
71	myop-zc	0.881	0.517
60	ar-ewpc-myop	0.880	0.507
63	ewpcl-hist-myop-nt-proc	0.880	0.551
41	ar-nt-var	0.878	0.520
54	ewpc-myop-nt	0.877	0.512
79	ewc-proc	0.877	0.541
74	proc-wave	0.873	0.542

Table 21: Listing the 40 best subsets using CST as a performance indicator.

Appendix C DVD

Here is a short explanation of what is put on the DVD.

C.1 Report Contains the project as a PDF file, and a subfolder with LaTeX source code.

C.2 References Contains most of the references as PDF files, and the BibTeX file `bibliography.bib`.

C.3 Matlab Contains all files from Matlab work.

Plots are available in pdf and png format in the subfolder named `/gfx/`.