NTNU
Innovation and Creativity

Proportional Myoelectric Control of a Multifunction Upper-limb Prosthesis

Anders Lyngvi Fougner

Master of Science in Engineering Cybernetics
Submission date: June 2007
Supervisor: Tor Engebret Onshus, ITK
Co-supervisor: Øyvind Stavdahl, ITK
Peter J. Kyberd, University of New Brunswick, Canada

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

The possibilities of controlling a multifunction upper-limb prosthesis by means of EMG signals and basic Bayesian pattern recognition were demonstrated decades ago, but due to technological limitations the results never became available for prothesis users.
Most of today's research focuses on more complex classifiers with an ON/OFF-style output, and no records indicate that these systems yield an outcome that is functionally superior to the simple Bayesian approach. The SVEN methods have recently been revived in a cooperative efford by NTNU and UNB, Canada. The purpose of this project is to further develop the results from a previous term project to a practical proportional control algorithm for a multifunction prosthesis. The rationale for this study is the hypothesis that a "simple" and smooth (proportional) control function will be easier for the Central Nervous System to adapt to, and thus provide increased functionality for the user.

1. Give an overview of existing methods and algorithms for EMG-based quantitative estimation of mechanical parameters like force, velocity, position etc. Special emphasis should be given to methods that are insensitive to amplitude variations, which is a fundamental problem associated with surface EMG signals.

2. Establish a protocol for data acquisition, preferably using ITK's motion lab, suitable for collecting training, testing and validation data for a proportional, multifunctional myoprocessor. The protocol should include implementation of any specific software needed.

3. Select a set of relevant pattern recognition methods, and adapt these to the problem at hand as needed. Evaluate and compare their performance as proportional, multifunction myoprocessors by means of real biomedical signals. Visualize the main results.

Assignment given: 08. January 2007
Supervisor: Tor Engebret Onshus, 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 such a thesis, problems occur and remain unanswered. This leaves a whole new set of topics to be researched in the future. Since prostheses are important tools for amputees in their daily life, it is my personal hope that the future research will be successful, and that my study is a contribution.

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

Øyvind Stavdahl for being my excellent and enthusiastic advisor at NTNU

Peter Kyberd for being my helpful co-advisor at University of New Brunswick, Canada

Tor Onshus for being my helpful supervisor at NTNU

Kristian and Reidun, my understanding parents, for always asking me the simple questions I need to be asked, to understand what I am really doing

My patient test subjects for making this study possible

My helpful friends, colleagues and discussion partners Håvard Torpe, David Karnå, Peter Näsholm, Morten Engen, Jan Egil Wagnild, Audun Sølvberg, Lars Vråle and others

Trondheim, Norway, 15 June 2007

Anders Fougner

Contents

List of Figures

List of Tables

Nomenclature

AAC	Average amplitude change
AAV	Averaged absolute value
ADL	Activity of daily living
ANN	Artificial Neural Network
EMG	Electromyography, the recording of the extracellular field potentials produced by muscles
LF	Linear mapping function
MLP	Multi-layer Perceptron
MYOP	Myopulse percentage rate
NT	Number of turns
QF	Quadratic mapping function
RMS	Root mean square, a common statistical measure of the magnitude of a varying quantity
RMSE	Root mean square error, a common statistical measure of estimation error
ROM	Range of motion
sEMG	Surface electromyography, surface recording of the extracellular field potentials produced by muscles
VAR	Variance
WAMP	Willison amplitude
WL	Wavelength
ZC	Zero-crossings

Abstract

This study is a part of a renew and continuation of the SVEN work done in Sweden in the later 1970's. The SVEN hand was an on/off-controlled upper-limb prosthesis based on measured electromyographic (EMG) signals. Recently the SVEN methods have been revived in a cooperation by NTNU and UNB, Canada.

The aim of this study is to further develop a practical porportional control system for a multifunction upper-limb prosthesis. This is based on a hypothesis that a simple and smooth proportional control system will be easier for the central nervous system to adapt to, compared to existing systems, and will thus provide increased functionality for the user.

A protocol has been developed for the recording of EMG signals and VICON motion measurements in a laboratory. Suitable data sets have been recorded from three test subjects, and signal processing and three pattern recognition methods have been applied on these data sets to generate estimates of clinical angles. The pattern recognition methods tested were linear (LF) and quadratic (QF) mapping functions and multi-layer perceptron (MLP) network. The performance of these methods has been evaluated, compared and visualized. More testing is needed to find the best method, and the MLP network can be improved in several ways.

To achieve better angle estimates that can be used for proportional control of prostheses, we wanted to use EMG signal features that are insensitive to amplitude changes due to variations in skin conductance. Qualitative and quantitative EMG signal features are described with this property as an important concern. The zero-crossings (ZC) feature has been tested as one of these, also in combination with the averaged absolute value (AAV). Although ZC did not always perform superior to AAV, it is likely that other features and combinations of these should be tested. Inclusion of other properties from the prosthesis, like elbow angle or measured pressure from the arm on the prosthesis, can also be included to improve the estimates.

We now have a large, suitable data set from the laboratory, which can be used for further work on pattern recognition and multifunction proportional control of prostheses. There are also other applications for the methods developed.

The final step will hopefully be implementation in a real prosthesis.

1 Introduction

This study is a part of a renew and continuation of the SVEN work done in Sweden in the later 1970's. The SVEN hand was an on/off-controlled upper-limb prosthesis based on measured electromyographic (EMG) signals.

Midtgaard (2006) did a modern reconstruction of the SVEN project and established a good starting point for further studies. The beginning of my work on this topic (Fougner, 2006) introduced the use of a motion analysis system (VICON) and applied pattern recognition methods on EMG signals to estimate the relevant angles of the upper limb. The final step will be to use this estimate and generate suitable control signals for proportional control of a prosthesis, but first several improvements in the estimation procedure will be needed. This study is a part of that.

To achieve better angle estimates that can be used for proportional control of prostheses, we want to develop methods that are insensitive to amplitude changes due to variations in skin conductance. Qualitative and quantitative EMG signal features are described with this property as an important concern.

A great effort has been made to record suitable data sets in the laboratory, and the developed signal processing and pattern recognition methods from Fougner (2006) have been applied on these data sets. Their performance has been evaluated, compared and visualized.

2 Theory

2.1 Electromyography

Electromyography (EMG) is a medical technique for recording and evaluating extracellular electrical potentials generated by muscles. A detailed description of this technique used in prosthesis control can be found in Muzumdar et al. (2004).

EMG is most often used for differentiating primary muscle conditions from muscle weakness caused by neurological disorders. It is normally recorded with a bipolar needle electrode inserted through the skin into the muscle of interest, and it is displayed on an oscilloscope while doing muscle contractions (MedlinePlus Medical Encyclopedia, 2007).

EMG is also used for prosthesis control. For this use, the needle electrodes are too painful og not suitable, thus skin-electrodes are used instead. They are less accurate but are still considered as a good measure of muscle activity or muscle force. The technique is sometimes referred to as surface EMG (sEMG), but in this thesis it will be called EMG.

The conductance of the skin-electrode interface is dependent on the placement of electrodes, the preparation of the skin and the amount of sweat. This affects the amplitude of the EMG signal and is a common problem in surface EMG measurements. It may be helpful to investigate if alternative EMG signal features, that might be insensitive to these amplitude variations, can be exploited for proportional prosthesis control. See Section 2.5.

2.2 The SVEN control system

This section is cited from Fougner (2006) and is mainly based on Midtgaard (2006).

The SVEN control system was developed in the 1970s by a team of scientists at Chalmers University of Technology in Göteborg, Sweden. It was designed for controlling a prosthetic hand with three degrees of freedom. These were finger flexion/extension, wrist flexion/extension and pronation/supination. The control system was electronic and utilised EMG signals measured by surface electrodes on the forearm stump of amputees. It used simple Bayesian pattern recognition techniques.

This control method lets the amputees move the muscles in their stump as they would do to move a natural hand. This is based on phantom hand perception, which is a phenomenon where the amputee can feel the movement and presence of the lost limb. This method is suitable because it requires minimal training time.

The original SVEN hand provided only on/off control for the three degrees of freedom. This included six movements (finger flexion/extension, wrist flexion/extension and pronation/supination), and the SVEN report (Almström, 1977) describes the results of an experiment with six electrodes. The samples showed distinct patterns of EMG signals for the six movements. Based on this a classification procedure was made to figure out which movement to perform.

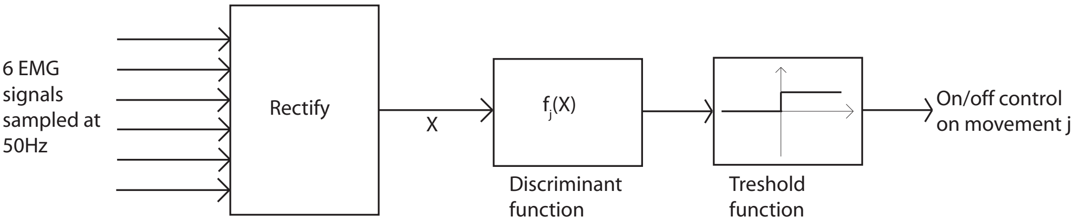


Figure 1: The SVEN control system

The EMG signals were sampled at 50Hz and rectified before the classification procedure.

2.2.1 The classification procedure

The classification in the SVEN control system was based on the following linear discriminant function calculated from the six collected EMG signals.

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

where

$$\begin{aligned} X &= [x_1, \ldots, x_i, \ldots, x_6]^T \\ W_j &= [w_{1j}, \ldots, w_{ij}, \ldots, w_{6j}]^T \end{aligned} \tag{2}$$

and

i : electrode site number

j : movement number

x_i : rectified and filtered EMG signal from electrode site i

w_{ij} : corresponding weighting factor for electrode site i, movement j

w_{0j} : constant term

For on/off-control, the resulting values of $f_j(X)$ decided whether to perform movement j or not, and the movements were classified in one of two populations. If $f_j(X) > 0$ the movement was performed and was said to be in the first population, and if $f_j(X) < 0$ the movement was not performed and was said to be in the second population. This also made it possible to perform more than one movement simultaneously.

The weighting factors w_{ij} and the constant term w_{0j} (which can be considered as the threshold level) had to be calculated to minimize the expected loss of misclassification. The calculations were derived from Bayes' criterion of optimality and based on statistics of experiments. In these calculations the covariance matrices of the six EMG signals were included, but to simplify the discriminant function it was assumed that the covariance matrices of the signals were equal for all the movements in both populations. More about this in Section 2.3. Another assumption was that the populations are normally distributed.

The covariance matrix and the weighting factors were calculated as follows

$$\Sigma = \begin{bmatrix} var(X_1) & \dots & cov(X_1, X_6) \\ \vdots & \ddots & \vdots \\ cov(X_6, X_1) & \dots & var(X_6) \end{bmatrix} \tag{3}$$

μ_{ij} : mean values for population i, electrode site j

$$M_i = [\mu_{i1}, \dots, \mu_{i6}]^T \tag{4}$$

$$W = \Sigma^{-1}(M_1 + M_2) \tag{5}$$

$$w_0 = -\frac{1}{2}(M_1 + M_2)^T \Sigma^{-1}(M_1 + M_2) \tag{6}$$

These equations are derived in Midtgaard (2006, on pp. 20–22).

2.3 The human nervous system

This section is cited from Fougner (2006).

The movement of a normal human upper limb is controlled through the peripheral nervous system. In the case of arm movements, the signals are sent from the motor cortex down the spinal cord, through a spinal nerve to the nerve plexus in the shoulder. From the plexus, the radial, median and ulnar nerves lead the signals to the forearm and hand muscles. Each of these nerves consists of thousands of fibres that can carry individual signals.

The motoric part of these nerves control several muscles in the forearm and the palm, so that the hand and fingers move. For the main movements of the forearm, the most important muscles are described in Table 1 on page 15.

The nerves do not only control the positions/angles of the various parts of the forearm, they control also motion speed and muscle force.

The peripheral nervous system also has a sensoric part. This includes proprioception (the sense of the relative position of neighbouring parts of the body, such as muscles, joints and tendons) and exteroception (sensors in the skin that can feel touch/pressure, temperature and pain) which are both important for the control of the upper limb. The proprioceptors are placed inside the limb, around muscles and tendons, while the exteroceptors lay in the skin or close to the skin. The other senses (vision, taste, smell, hearing and balance) by organs in the head can also play their role in the control of the upper limb.

In all, the human nervous system makes an accurate, reliable and robust control.

2.4 Comparison

This section is cited from Fougner (2006).

A fully functional human upper limb naturally has more DoFs (degrees of freedom) than the SVEN control system. The fingers can be moved individually and do fast and complex movements, and the whole system is very flexible. It is also stronger, more reliable, natural-looking and easier to carry around. The fingers can feel the temperature or the structure of a surface. In addition, the hand has its own properties; the surface may be soft and warm and the fingerprint is unique. To reconstruct all these qualities in a prosthesis is extremely difficult and may be impossible. Anyway, it is not obvious that we need all these qualities to manage the daily life, and an improved control system may be satisfying as a substitute after an amputation. So far, the main goal is not necessarily to reproduce a perfect upper limb, but to make a tool which can be useful for an amputee in daily life.

The SVEN control system had some advantages compared to other prosthesis control systems, because it was adapted to the amputees. In contrast to most other existing systems, the SVEN system allowed the amputees to move the muscles in their stump as they would do to move a natural hand.

The on/off control was one of the obvious drawbacks of the SVEN control system. Because of this, one of the main targets in this project is to introduce proportional control. Instead of comparing a function value with a treshold to turn motors on and off, the function value can be used as a control signal.

Another drawback was the reliability of the SVEN system. While a healthy limb has 100% reliability, the SVEN control system had up to 24% wrong classification on some movements (Almström and Herberts, 1980). One of the reasons was probably that the assumptions made in the SVEN study were too simplified, to make it possible to calculate with the technology of the 1970's. For example, the assumption that the covariance matrices are identical for all movements, is probably not true. Midtgaard (2006) showed that the covariance matrices of each population are not identical, and inside each population they are probably also different. Due to this, we want to try more complicated pattern recognition methods to generate a better control signal.

Using the SVEN control system on an upper-limb prosthesis, a normal problem was that the wrist flexes when you want to flex the fingers. This happened because the two movements use essentially the same muscles. In a healthy upper limb, this is solved by stabilizing the wrist with antagonists (extensor carpi ulnaris, extensor carpi radialis longus & brevis) (Dahl and Rinvik, 1999). Equivalent problems and solutions also exist on several other movements, and in a prosthesis this may need special attention. In our control system, we will try to place electrodes on the relevant opposing muscles to be able to distinguish the relevant movements. We also need effective pattern recognition methods.

Possibly, our proportional control may introduce some sort of feedback through the amputee's eyes. If there is an error due the pattern recognition, and the prosthesis moves wrong, the amputee may detect this visually in an early stage and autocorrect it - consciously or unconsciously.

2.5 EMG-based estimation

2.5.1 Qualitative EMG signal features

Averaged absolute value (AAV) The averaged amplitude of the EMG signal is the most common feature used in prosthesis control. A simplified formula is

$$\text{AAV} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{7}$$

where N represents the number of samples in the segment and i is the sample number within the segment (Boostani and Moradi, 2003). This calculation can be done directly on the raw EMG signal or (more usual) on a filtered EMG signal. Usually some signal processing will be performed.

The main problem of this feature is that it will be dependent on amplitude changes due to sweat and varying skin conductance. The calculation time is short and the implementation is easy, but some effort is needed to optimize the signal processing. See Section 5.3.1 and 9.

2.5.2 Quantitative EMG signal features

Variance (VAR) The variance of the signal may be calculated for each time segment. This is a common statistical feature and also represents the signal power. The formula is

$$\text{VAR} = \frac{1}{N-1}\sum_{i=1}^{N} x_i^2 \tag{8}$$

where N represents the number of samples in the segment and i is the sample number within the segment (Boostani and Moradi, 2003).

The calculation time is very short and the implementation is easy, but this feature has the same problems as AAV.

Willison amplitude (WAMP) The Willison amplitude is counted every time the change of amplitude between two samples exceed a treshold value (Philipson, 1987), typically $50\,\mu$V. This can be expressed mathematically (Boostani and Moradi, 2003)

$$\mathrm{WAMP} = \sum_{i=1}^{N} \mathrm{tresh}(|x_{i+1} - x_i|) \tag{9}$$

where

$$\mathrm{tresh}(x) = \begin{cases} 1 & \text{if} \quad x > \text{treshold} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

This feature has a short calculation time and is easily implemented. We do not know if will have the same problems as AAV.

Zero-crossings (ZC) Zero-crossings count the number of times the EMG signal crosses the zero amplitude level (Boostani and Moradi, 2003). Mathematically it is calculated as

$$\mathrm{ZC} = \sum_{i=1}^{N} \mathrm{tresh}(-x_i x_{i+1}) \tag{11}$$

where the tresh function is defined in (10).

It is important to high-pass filter the signal before calculating the ZC feature, so that offset changes do not affect the result. This could be a higher-order filter with 1Hz cut-off frequency, to keep intact as much of the signal as possible.

The treshold value is introduced to avoid low voltage fluctuations around zero being counted as zero-crossings (Philipson, 1987). If it is set to zero, the ZC feature will be independent of amplitude changes due to varying skin conductance, and that is a good property.

In this study we may try ZC as a single feature, but also combined with AAV as a complementing feature.

Number of turns (NT) Number of turns count the number of times the slope of the EMG signal changes sign (Philipson, 1987). Mathematically it is calculated as

$$\mathrm{NT} = \sum_{i=1}^{N} \mathrm{sgn}(-(x_{i+1} - x_i)(x_{i+2} - x_{i+1})) \tag{12}$$

where

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if} \quad x > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

This feature is closely related to ZC. It has a short calculation time and is easily implemented.

Myopulse percentage rate (MYOP) The myopulse output (Philipson, 1987) is defined as 1 when the absolute value of the EMG signal exceeds a treshold value, and as 0 otherwise. The myopulse percentage rate is then the average value of the myopulse output. Mathematically it is calculated as

$$\mathrm{MYOP} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{tresh}(x_i) \tag{14}$$

where the tresh function is defined in (10).

This feature will be dependent on amplitude changes due to sweat and varying skin conductance . The implementation is easy and the calculation time will be short.

Average amplitude change (AAC) This feature is calculated as the mean value of the difference between two consecutive samples of the EMG signal (Boostani and Moradi, 2003):

$$\mathrm{AAC} = \frac{1}{N}\sum_{i=1}^{N} |x_{i+1} - x_i| \tag{15}$$

This feature is also easily implemented and calculated.

2.5.3 Other qualitative EMG signal features

Boostani and Moradi (2003) have described several other EMG signal features, but none of them were described clear enough to be reproduced and explained well in this study. The feature **Energy of wavelet packet coefficients** was the most promising, so it would be interesting to try this feature in the future (possibly combined with other features) if it can be reproduced. More time is needed to test these features. All the following features are found in the study of Boostani and Moradi (2003) and their performance in classifying EMG signals was investigated. For all features, the calculation time was evaluated and compared to the maximum tolerable time consumption for real-time calculations with reasonable hardware.

Wavelength (WL) This feature was partially described with a formula, but their formula is not usable without a better description. A short explanation is that this feature estimates the length of a waveform in a segment. This feature performed quite well.

Mean frequency / Median frequency Mean frequency is partially described by referring to Park and Meek (1993), but they are mixing up the expression *mean frequency* with *median frequency* and it is not known exactly how this was calculated. No formulas are given.

Histogram This is a count of signal samples in different amplitude levels within a time segment. It can be described as an extension of the ZC and WAMP features. Boostani and Moradi (2003) used nine amplitude levels and we do not know how these levels were selected.

Auto-regressive coefficients A fourth-order autoregressive (AR) model was used. Signal samples are estimated by linear combinations of previous samples, and the AR coefficients will change with the muscle force. The AR coefficients can also be extracted from the third- or fourth-order cumulant of the signal in a time segment, and doing this will determine the relations among samples in higher orders and will also contain phase information on the signal. This is not described in detail. The calculations are quite time-consuming.

Cepstrum/cepstral coefficients Cepstrum is explained as finding the spectrum of the spectrum (the fourier transform of the fourier transform) of a signal, as if the spectrum was a normal signal. This feature is widely used for representing voice and music, but it can be used also on EMG signals. Often the cepstrum is found by doing the *inverse* fourier transform of the fourier transform of the signal, but this is not the original definition. It was never explained in detail how the coefficients were found in this case, but the feature performed very well in classification of EMG signals and the calculation time was short.

Energy of wavelet coefficients The EMG signals were decomposed by wavelet transform into nine scales and the signal energy was found as components of the feature vector. A biorthogonal mother wavelet was used because it has similarities with the action potential of muscles. There were 2^9 samples in each time segment of 200ms and thus the number of scales was chosen to be nine. No further description is given, so to reconstruct this feature exactly is impossible. Finding the wavelet transform is however a common method for classifying EMG signals, and it should be investigated also for proportional control.

Energy of wavelet packet coefficients Instead of wavelet transform it is possible to use wavelet packet transform. This feature will be more complex, but it actually had the best performance of all features in the study of Boostani and Moradi (2003). The calculation time was the longest of all features. It is not completely described how to calculate, but Matlab has some built-in algorithms in the Wavelet Toolbox that might be used in future work on this topic.

ZC of wavelet coefficients This is a simple combination of other features; using zero-crossings on the wavelet coefficients. This feature performed well in classification of EMG signals, but it was too sensitive to noise and the calculation time was long.

AAV and ZC of wavelet coefficients This feature will have twice as many components as the previous feature.

AR coefficients of wavelet coefficients These coefficients can be calculated by the third- or fourth-order cumulant of wavelet coefficients. Both are sensitive to noise and require quite time-consuming calculations.

3 Aim of the study

The aim of this study is to further develop a practical proportional control for a multifunction upper-limb prosthesis.

A protocol will be established for recording of electromyographic (EMG) signals and VICON motion measurements in a laboratory, and the protocol will be used for recording suitable data sets for training, validation and testing of pattern recognition methods. See Section 4.

Several methods and algorithms are described for EMG-based quantitative estimation (see Section 2.5). They will be evaluated, and special emphasis will be given to EMG features that are independent of amplitude variations due to skin conductance, which is a common problem in EMG-based estimation.

Three pattern recognition methods (see Section 5.2) will be evaluated and compared in producing proportional control signals for an upper-limb prosthesis.

4 Protocol for data aquisition

This section describes:

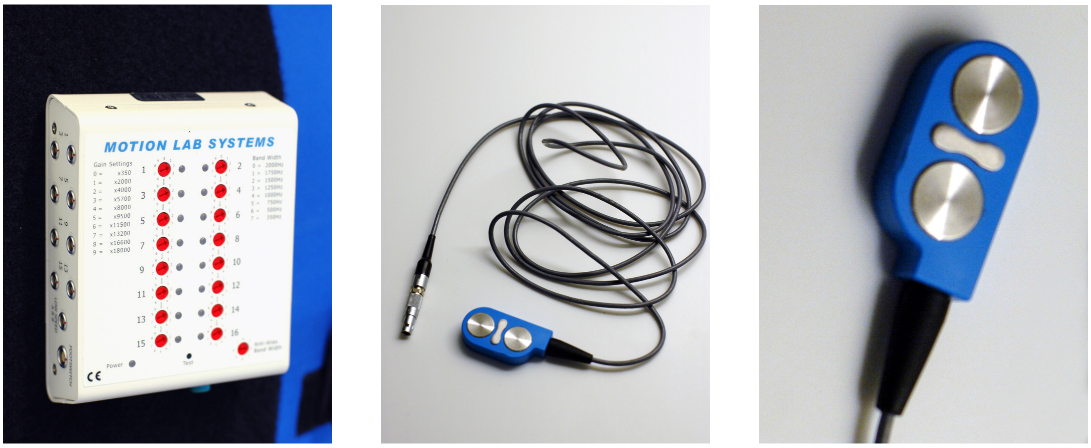
- The equipment used for recording EMG signals and positions of the upper limb
- The placement of electrode sites and markers
- Guidelines for selection of test persons
- A specification of movements to be recorded to form data sets used for pattern recognition and proportional control of upper-limb prostheses

4.1 Technical specifications

4.1.1 EMG sampling equipment

EMG was sampled using a portable multi-channel box (Fig. 2a) connected to the VICON system. The eight electrodes used (Fig. 2b–2c) had a signal ground point between the contact points, and a built-in 20x pre-amplifier. In addition an extra 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.

EMG signals normally have a bandwidth of 500Hz. VICON did the EMG sampling at 1.5kHz, which is a common sampling frequency for surface EMG signals and according to the Nyquist criterion 1kHz would be enough.



2a: EMG multi-channel box

2b: Electrode

2c: Electrode

Figure 2: Equipment

4.1.2 VICON equipment for motion measurements

VICON is a motion measurement system made for gait analysis, biomechanical research, sports motion analysis and animal science. The system can compute two-dimensional positions from one camera or three-dimensional coordinates from 2-16 cameras. The cameras have two or four megapixles and record digital photos in grayscale, usually 60 photos per second (60 Hz recording).

Six cameras were used in this study; four cameras under the ceiling in the corners of the room and two cameras placed high on the end walls of the room. They were all directed towards the center of the room and the recording system was always calibrated on the day of recording. See Fig. 3.

Since the laboratory is normally used for gait analysis, all cameras were placed high, which is not optimal for the marker visibility in our study. Anyway, it was decided not to move the cameras as long as no problems occurred concerning the marker visibility.

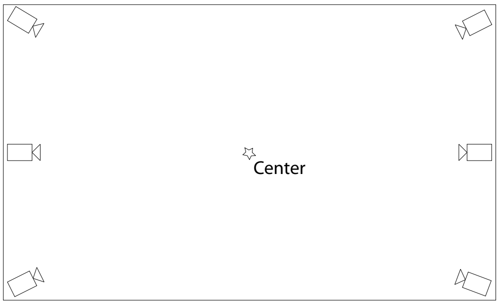


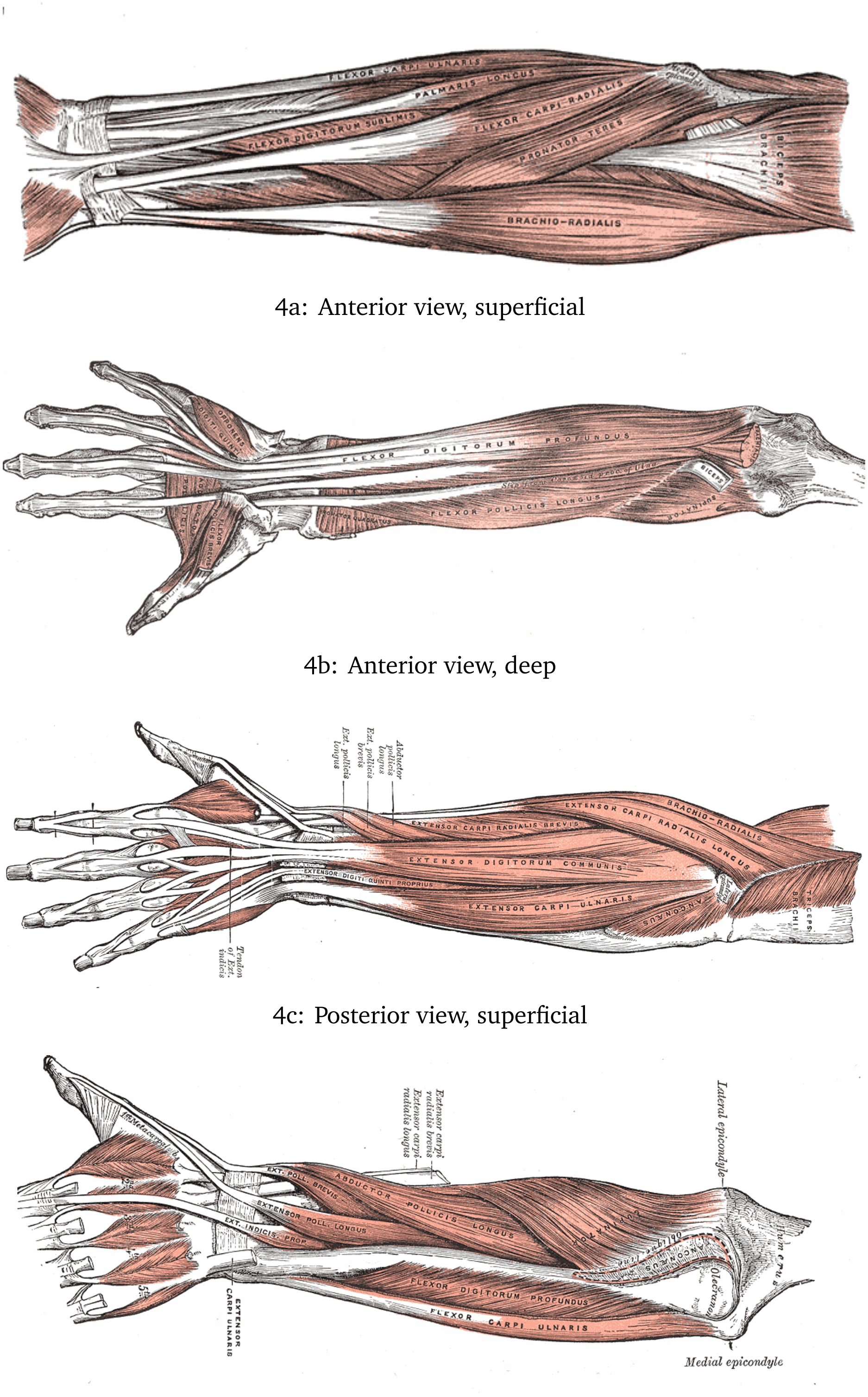
Figure 3: VICON camera placement

VICON Workstation v.4.6 was used. The motion measurements were 60 Hz. They were however downsampled to 20 Hz before being used as input to pattern recognition, see Section 5.3.2.

4.2 EMG electrode site placement

When selecting the placement of EMG electrode sites, the main focus should be to place them as close as possible to the relevant muscles. Table 1 was made to select the best placements of the electrodes (Midtgaard, 2006; Stavdahl, 2002; Perotto, 1994; Gray, 1918; Boostani and Moradi, 2003). See illustration of the forearm muscles in Fig. 4.

The movements of the wrist are illustrated in Fig. 5. The finger flexion/extension movement is not illustrated here but is assumed to be known.



4a: Anterior view, superficial

4b: Anterior view, deep

4c: Posterior view, superficial

4d: Posterior view, deep

Figure 4: Forearm muscles, illustrations from Gray (1918)

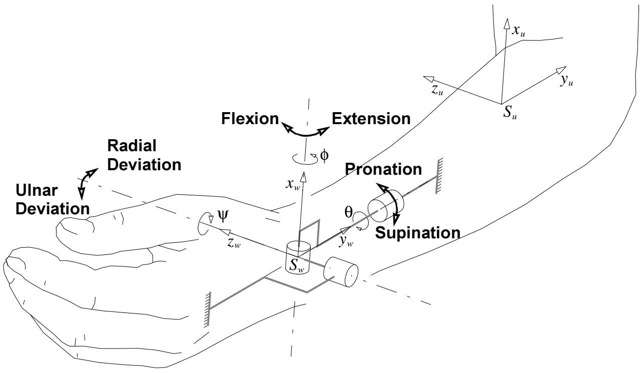


Figure 5: Clinical angles of the wrist (Stavdahl, 2002)

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 1: Relevant muscles for specific movements

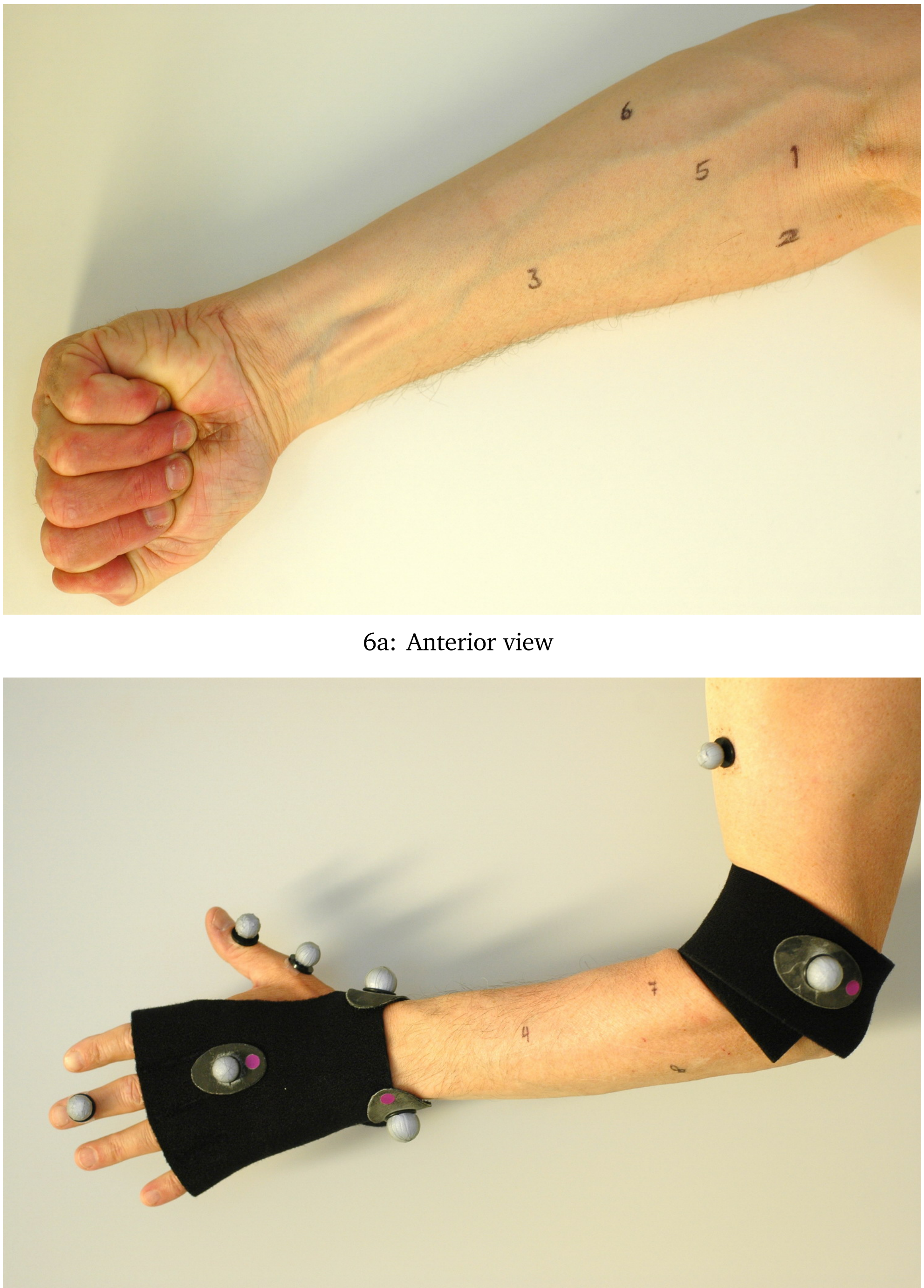
In total, this table indicates use of at least eight different electrodes on the forearm, to distinguish the eight movements. This leads to the EMG electrode site placement of Table 2.

An important concern is that for each person the electrodes should be placed correctly and at the same place every time, so that the classification procedure is adapted to the correct signals.

[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 (Perotto, 1994)



6a: Anterior view

6b: Posterior view, with markers

Figure 6: Electrode site placement

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 2: Electrode site placement

4.3 VICON marker set placement

For calculation of relevant lines and angles of the upper limb, to record the movements of Table 1, the marker set placement of Table 3 is sufficient. Note that the markers A and LMB were recorded but not used in this study, however they might be useful in future work on the recorded data sets.

Abbreviation	Description
A	Acromion
LMB	Lateral (side of) Musculus Biceps
LE	Lateral Epicondyle
ME	Medial Epicondyle
RS	Radial Styloid Process
US	Ulnar Styloid Process
MCP(1,3)	$(1^{st}, 3^{rd})$ Metacarpophalangeal joint
DIP(1,3)	$(1^{st}, 3^{rd})$ Distal interphalangeal joint

Table 3: Marker set placement abbreviations

See the marker set in Fig. 8.

4.4 Test subjects

EMG and VICON position measurements were done on three volunteers. These persons are held anonymous. To have a more generally valid result, at least one person of each gender was chosen, with different age and body build and without any known neuromuscular diseases. Here are descriptions of the three test persons chosen:

Person 1 Male, 26 years, very athletic, elite cross-country skier, well-trained forearm muscles, little subcutaneous fat. Right-handed.

Person 2 Male, 57 years, athletic, regularly exercising, moderately trained forearm muscles. Left-handed.

Person 3 Female, 55 years, non-athletic, normal build. Right-handed.

We wanted to record signals from the non-dominant hand, because for most unilateral amputees the remaining hand will become dominant. In the study of Kestner (2006), only 2 of 40 individuals with unilateral limb loss who lost their dominant hand did not change hand dominance to their unaffected side.

One of the test persons appeared to be left-handed. However, it was preferred to use the same side for all test persons, thus all data were recorded from the left arm in this study.

4.5 Choice of movements

We want to choose a set of movements that we can record in the laboratory and use for training, testing and validation of different pattern recognition techniques. The system should be able to recognize and perform simple movements, combinations of these, and activities of daily living (ADLs). Thus, the movement sets should include these types of movements, and originally a set containing ADLs (slicing bread, eating, cutting paper, stirring, sweeping, hanging clothes on a clothesline) was selected.

However, the ADLs proved to be non-suitable for this purpose. The reason was that ADLs were selected such that they are movements that require both hands (since single-hand movements will normally be done with the healthy hand, not the prosthesis), and such that the healthy hand does the main movement while the prosthesis does the support (if the amputee is unilateral). This resulted in very small movements that were very difficult to record with VICON, although the EMG signals and the muscle force used for support (which we do not record) was significant. To find a relation between these EMG signals and very small movements was not a good idea, and another choice of movements was done.

All possible combinations of two simple movements were selected. This includes movements following each other, like first pronating and then doing finger flexion/extension, and movements done simultaneously, like wrist flexion while doing pronation or supination. These combinations span a space which we want to cover as good as possible. See Fig. 7 for an example. Note that this space is not always a rectangular space, for example the radial/ulnar deviation is very restricted when doing maximum wrist flexion or extension. a and b refers to doing wrist flexion and two different amounts of pronation at the same time, while c and d refers to doing first wrist flexion and then pronation.

See Section 6.3 on page 31 for example plots of combined movements recorded in the laboratory.

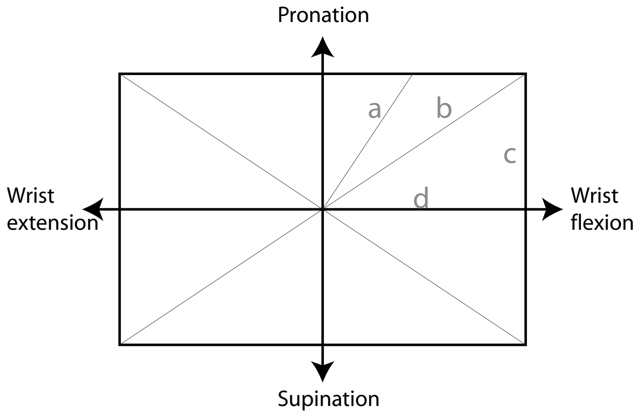


Figure 7: Example of combined movements

A set of movements was selected, see Table 4. These movements should be recorded at least three times for each subject, so that we have sets for training, testing and validation. We record all these movements every time, but we will be able to select a subset later on if the sequence is too long or if we want to see if a subset performs better than the total sequence.

S#	Simple movements
S1	Finger flexion/extension
S2	Wrist flexion/extension
S3	Pronation/supination
S4	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 4: Movements

4.6 Training, testing and validation sets

For each test subject, three different data sets were recorded:

Set 1 & 2 Recorded at the same day, with the same electrodes and markers

Set 3 Recorded at another day, with new electrodes and markers

Training - Validation - Testing
1 - 2 - 3
1 - 3 - 2
2 - 1 - 3
2 - 3 - 1
3 - 1 - 2
3 - 2 - 1

Table 5: Data set combinations

The purpose of recording set 3 at another day, was to see if a trained pattern recognition method will still be useful when new electrodes and markers have been placed and the skin conditions may be a little different.

These three data sets may be combined in six ways as training, testing and validation sets, see Table 5. We want to try all these combinations.

5 Method description and implementation

5.1 Angle calculations

The clinical angles of the upper limb were calculated based on the recorded marker positions. Several vectors were calculated, see Table 6 and Fig. 8. The clinical angles were calculated from these vectors, as presented in Table 6.

Vector	Markers
Fvec1	MCP1, DIP1
Fvec3	MCP3, DIP3
WristAxis	RS, US
ElbowAxis	ME, LE
Radius	RS, LE
Ulna	US, ME
Hand	US, MCP3
Angle	Vectors
Finger flexion/extension	Fvec1, Fvec3
Wrist flexion/extension	Hand, Ulna
Pronation/supination	WristAxis, ElbowAxis
Radial/ulnar deviation	Hand, WristAxis

Table 6: Vectors and angles of the upper limb

A special problem occured in the case of finger flexion/extension and wrist flexion/extension, because resulting angles were rectified (the angle was found as the smallest angle of two vectors, thus it will be positive). This was solved using the following method:

1. project the vectors into a plane orthogonal to the WristAxis vector
2. calculate the angle of the new vectors (this will always be positive)
3. calculate the cross product of the new vectors (we can use the name Xprod for this cross product)
4. calculate the dot product of the vectors Xprod and WristAxis
5. check the sign of the result, and use this sign for the angle

The new angles are now in the area [-180,180] degrees instead of [0,180] degrees. See angle calculation results in Section 6.2

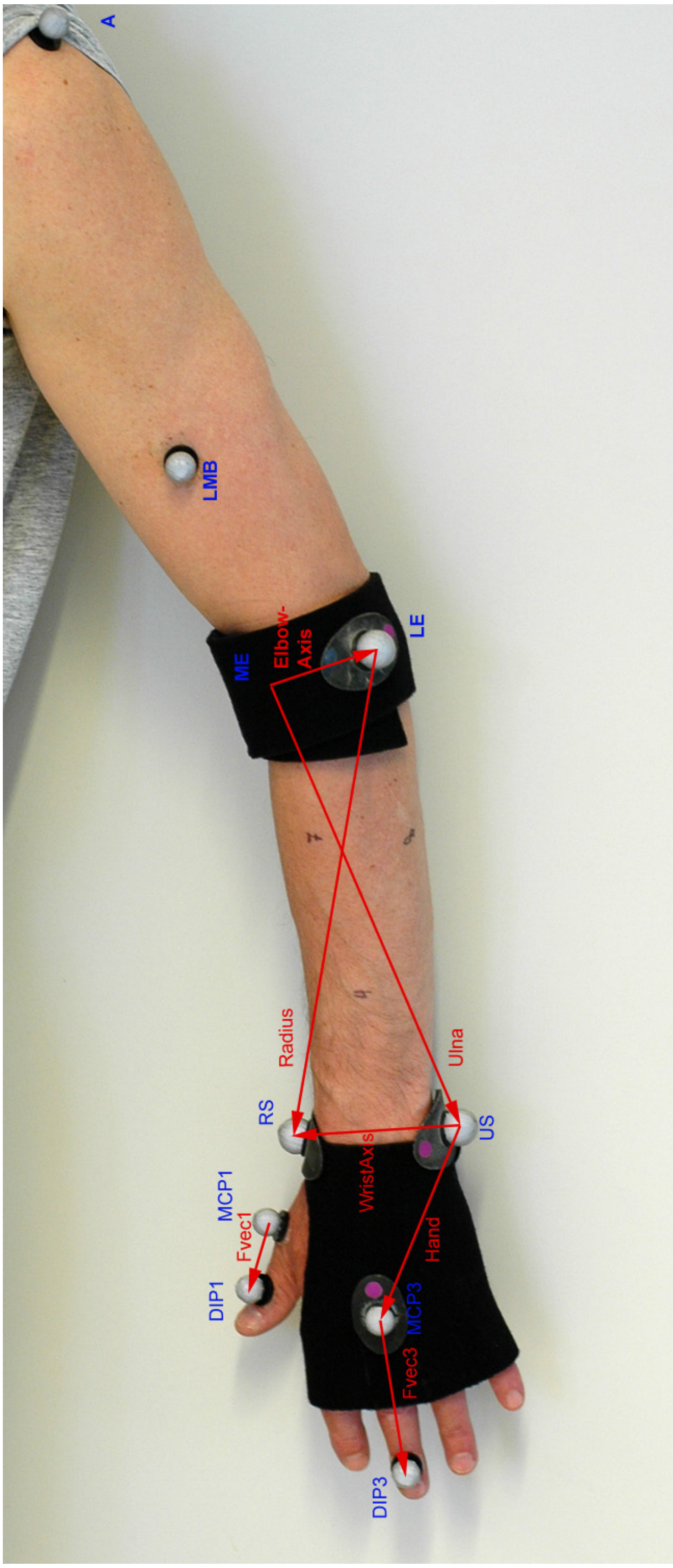


Figure 8: Vectors and angles of the upper limb (note that the arm is pronated)

5.2 Pattern recognition methods

This section is cited from Fougner (2006).

The most simple and intuitive way to introduce proportional control of the upper-limb prosthesis, is to use estimated angles directly as control output signal. I have investigated three methods for this estimation process.

From now on the discriminant function is renamed to a mapping function, since we are talking about proportional control rather than on/off-control. It is still a simple Bayesian classifier, but we use the function value directly instead of a treshold function.

5.2.1 Problem description

In the calculation of input data for the pattern recognition methods, the VICON system is essential. This system gives reference values θ_j representing the angles of the natural limb on which we are measuring the EMG signals. The measured θ_j are then ideal values, while the pattern recognition methods calculates estimated values $\hat{\theta}_j$.

The goal of the pattern recognition methods will be to minimize the error $e_j = \theta_j - \hat{\theta}_j$. This process will be done on a training data set, and then the calculated function parameters or relations may be used to find estimated angles $\hat{\theta}_j$ as control signal for a prosthesis. The minimizing process can be done in several ways, but the conventional method is least-squares estimation.

5.2.2 Linear mapping function

$$f_j(X) = W_j^T X + w_{0j} \tag{16}$$

We may use the linear mapping function (LF) (16) as in (Midtgaard, 2006).

$$\begin{aligned} f_j(X) &: X \rightarrow \hat{\theta}_j | \min_{\hat{\theta}_j} (\theta_j - \hat{\theta}_j)^2 & (17) \\ \hat{\theta}_j &= g(f_j(X)) & (18) \end{aligned}$$

We use least-squares estimation (17) to find the best values of W_j and w_{0j}. This is done with the Matlab function `secondOrderEstimation.m` in Appendix A.1, which returns the values $\hat{\theta}_j$.

5.2.3 Quadratic mapping function

$$f_j(X) = X^T W_{1j} X + W_{2j}^T X + w_{0j} \tag{19}$$

An alternative is the quadratic mapping function (QF) (19). Since X is a measured EMG signal, the rest of the minimizing to find the best values of W_{1j},

W_{2j} and w_{0j} is analog to the linear case. This is done with the Matlab function `secondOrderEstimation.m` in Appendix A.2, which returns the values $\hat{\theta}_j$.

5.2.4 Multi-layer perceptron network

This section is based on Bronzino et al. (2005) and is not a complete description of MLP networks, but a short introduction to the concept.

A Multi-layer perceptron (MLP) network is a special version of an Artificial Neural Network (ANN) and is a pattern recognition method commonly used on bioelectric signals. It can be constructed by built-in functions of the Neural Network Toolbox in Matlab.

A simple MLP network consists of three layers of nodes (also called neurons) and is a simplified model of how the network in a human brain works. See example in Figure 9.

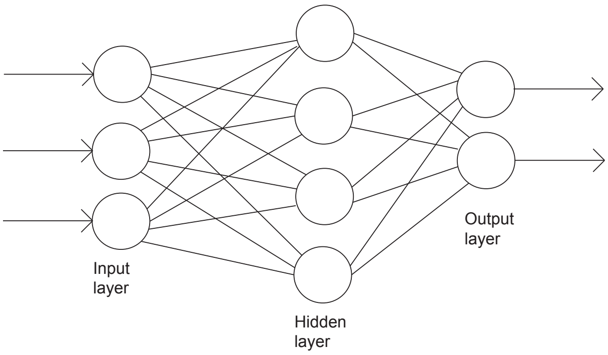


Figure 9: Example of MLP network with 3 inputs, 2 outputs and 4 nodes in the hidden layer

The first layer is called *input layer* and has as many nodes as input signals. When we have eight EMG electrode sites, we want eight nodes in the input layer. The middle layer is called *hidden layer* and most of the processing occurs in this layer. The size of this layer is important for the result of the estimation. The third layer, *output layer*, defines the output signals. If we want to estimate four angles, this layer will have four nodes.

Each node in the MLP network does a summation of all the inputs and a bias value, and a transfer function to generate the output. See Fig. 10. In Matlab's Neural Network Toolbox, a useful transfer function is `tansig`, based on the formula of (20). It is mathematically equivalent to `tanh`, with some numerical differences, but it is faster to calculate.

$$\mathrm{tansig}(x) = \frac{2}{(1+e^{-2x})-1} \tag{20}$$

The MLP network in this project is trained using least-squares estimation and back-propagation. In this process, the derivative of the transfer function is used,

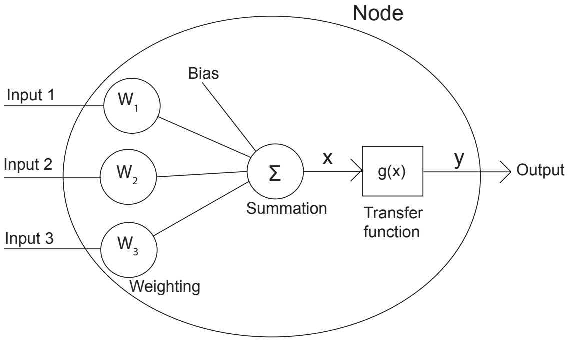


Figure 10: Example of a single node in a MLP network

and that is the reason why a smooth transfer function is chosen. The derivative of the $tansig$ function is a very simple calculation and makes the training process fast. When the MLP network is made with $tansig$ transfer functions in every node, and the number of nodes is equal to the number of parameters in one of the mapping functions, the MLP network and the mapping function actually gives the same result.

The MLP network may be a good alternative to the mapping functions. Of the three methods mentioned, the MLP network can be implemented faster and is also easier to improve or extend, by adjusting a few parameters, but it is trained slower than the mapping functions. Since the training is done off line, not in real time inside a prosthesis, the training time is not that important and the MLP network may be useful.

The implementation of the MLP network is done with the Matlab function `neuralNetwork.m` in Appendix A.3, which returns the values $\hat{\theta}_j$.

It is important to choose the training data such that it contains a large variety of movements. Then the network is able to recognize many different moves. See Section 4.5.

5.3 Choice of EMG feature sets

Originally it was a wish to try several different EMG feature sets and compare them to find the best combination of features. However, after a time-consuming work in the laboratory, only three EMG feature sets were selected:

1. Averaged absolute value (AAV), a filtered EMG signal, the filtering procedure is described in Section 5.3.1

2. Zero-crossings (ZC) without treshold, on a high-pass filtered EMG signal

3. Combination of AAV and ZC

AAV was selected because this is the traditional and most used feature set. ZC without treshold was selected because this is is completely independent of amplitude changes due to varying skin conductance; as long as the EMG signal is first high-pass filtered so that the average is zero. Independency of amplitude changes is interesting, thus two completely different sets were chosen, and also the combination of these, to see how this affects the result.

5.3.1 EMG filtering procedure

The purpose of this filtering procedure is to make a smooth representation of the signal's amplitude.

The EMG signal is sampled at 1500 Hz. We use a high-pass filter (21) with a cut-off frequency of 1 Hz to remove the bias and rectify the signal. A low-pass filter (22) is used to remove the highest frequencies. After some testing, a low-pass filter with a cut-off frequency of 10 Hz was chosen because this made a smooth signal and a good representation of the amplitude of the EMG signal.

$$H_{high-pass} = \frac{\frac{1}{2\pi}s}{\frac{1}{2\pi}s+1} = \frac{s}{s+2\pi} \tag{21}$$

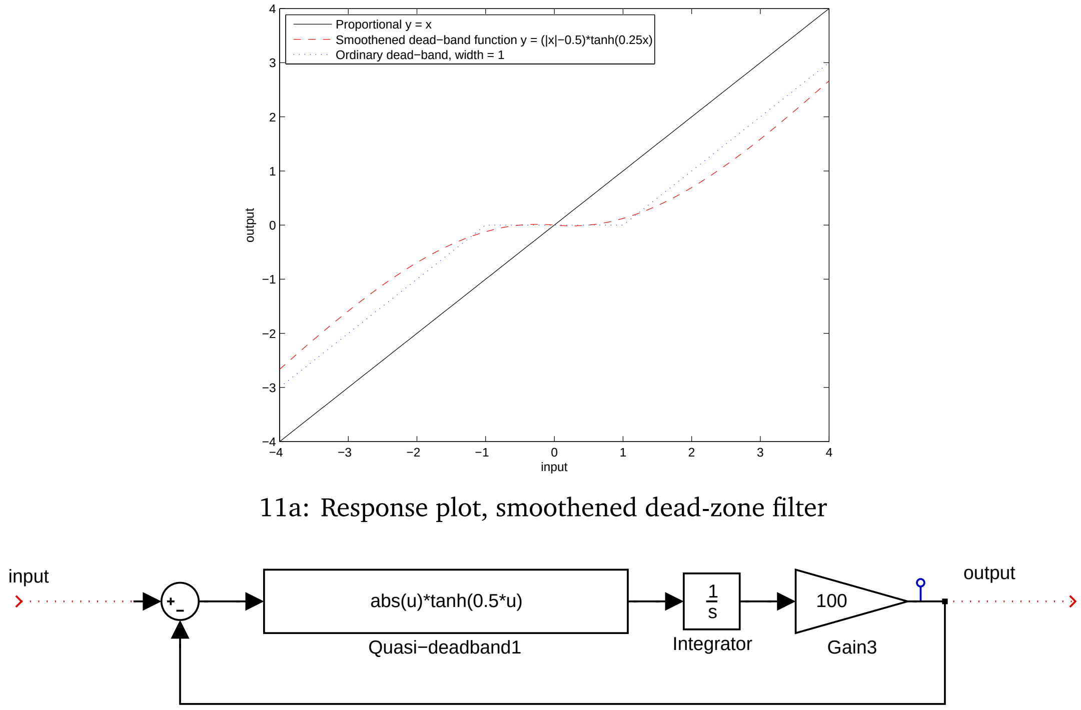
$$H_{low-pass} = \frac{1}{\frac{1}{2\pi 10}s+1} = \frac{20\pi}{s+20\pi} \tag{22}$$

In the end of the signal processing part, a special non-linear dead-zone filter (Fig. 11) was used, based on the $tanh$ function. The dead-zone assures that the integrator's output does vary only when the difference between the input and the output is larger than the width of the dead-zone. Small changes in the input will not affect the output, while larger changes will affect like they should. This can however introduce some problems when the amplitude for some reason is smaller than normal, for example due to increased skin conductance or bad contact with the electrode. This is discussed in Section 7 and 9.

Note also that the dead-band function in Fig. 11a has a small overshoot, which is not very evident but it can be seen when zooming into the plot. The overshoot is not optimal and should be removed. It may be possible to make a smooth dead-zone function by using a higher-order function and combine it with a linear function that has the same function value and slope in the meeting point. This is a topic for future work, see Section 9.

The amplifier ($K_i = 200$) was adjusted such that the integrator is fast enough to follow the changes of the input.

Our system is now quite similar to the SVEN control system, but with some significant improvements (Fig. 12).



11a: Response plot, smoothened dead-zone filter

11b: Simulink diagram containing the filter

Figure 11: Non-linear dead-zone filter

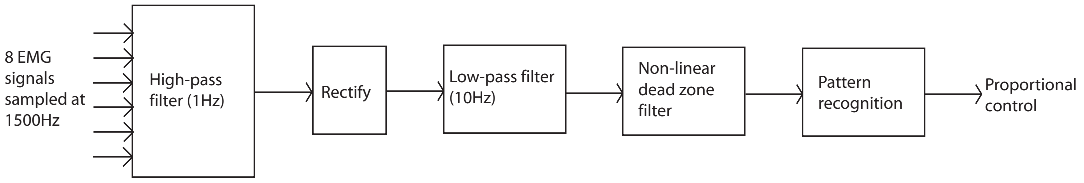


Figure 12: Proportional control system based on EMG signals

5.3.2 Signal preparation for pattern recognition

Mann et al. (1989) found the bandwidth of the wrist to be $\approx$ 10–12 Hz, containing $\approx 75\%$ of the signal. The finger flexion/extension might have a slightly larger bandwidth since fingers are is faster than the wrist, but normal protheses are not able to do the motions that fast. Because of this and the Nyquist criterion, a sample frequency of 20 Hz was chosen as a starting point for the pattern recognition and prosthesis control signals.

Thus, for doing effective estimation, the VICON position measurements (60Hz) and the processed EMG signals (1500 Hz) were all averaged to 20 Hz before being used as input to the pattern recognition algorithms. This allowed a choice more nodes in the MLP networks (see Section 5.2.4) without tying up all memory on the computer during the estimation procedure. The zero-crossing values were calculated at periods of 1/20 second to get the same frequency, and the resulting estimated angles were 20 Hz.

In short parts of the data sets, not all markers were visible. These periods were always removed from the data set before using the set in pattern recognition.

5.4 Proportional control

As mentioned, the estimated angles may be used directly as control signal for a prosthesis. This, however, will introduce noise and unwanted movements when the amputee tries to relax and let the prosthesis stand still, because the estimates will always contain some noise. To reduce this problem, we may implement low-pass filtering and a dead-zone element on the control signal.

To be able to use the θ_j values from VICON, the angles must be chosen such that we get the best measure of the degrees of freedom that we want to control. This is proportional control of position. It is not trivial that this solution is the best, it could be just as intuitive to control velocity, torque or even mechanical impedance (Abul-Haj and Hogan, 1990). To make it simple and easier to verify how good the pattern recognition methods works as control signals, we will try simple control of positions/angles first.

6 Results and observations

6.1 EMG processing

The EMG signal processing is described in Section 5.3.1. The results of this process is presented in Fig. 13. Note that this is a 10 seconds segment example from a data set of approximately 600 seconds.

The zero-crossings plot shows a noisy 20 Hz signal which should have been filtered in some way before being used as input to the estimation procedures. This is a topic for future work, see Section 9. In this case, the treshold value was chosen to be zero, to make it independent of skin conductance.

The signals from pronator and supinator seemed to be very weak, for all three test persons, and especially the supinator signal was weak. It is expected, because supinator is laying under brachioradialis and because supinator is relatively small.

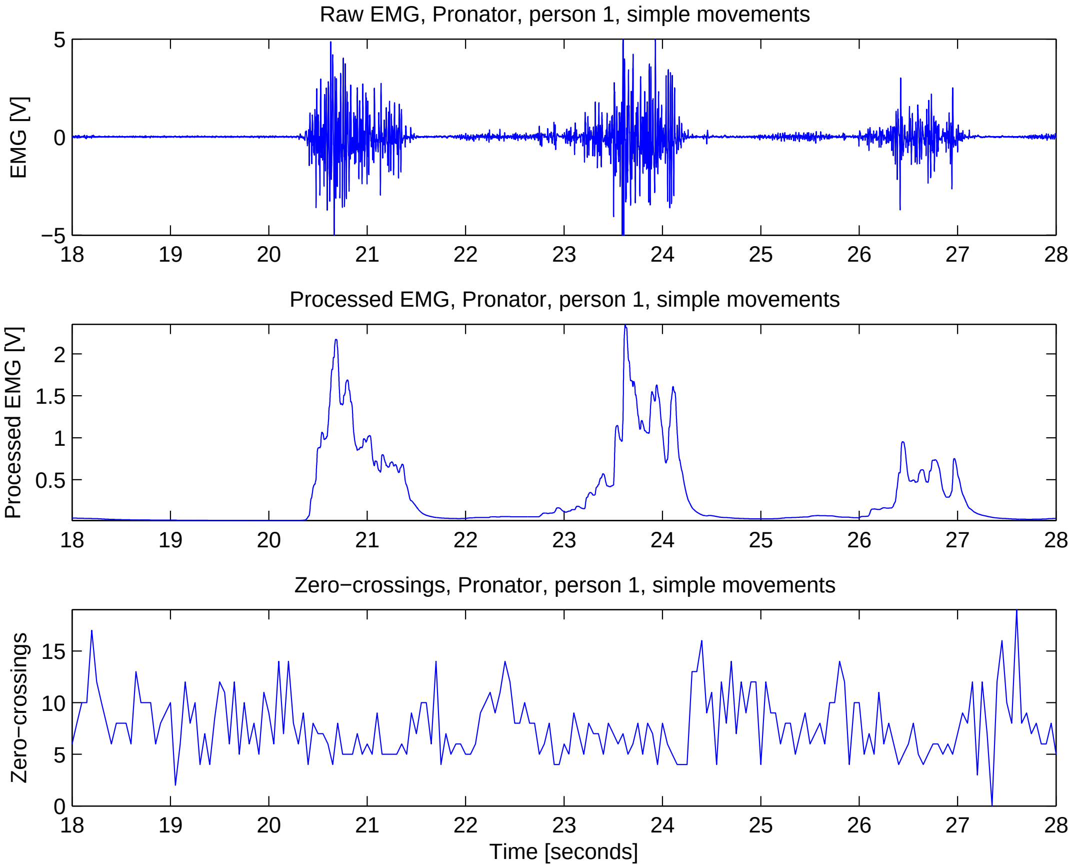


Figure 13: EMG processing example, channel 1 (Pronator), person 1

6.2 Angle calculation

The angle calculation procedure is described in Section 5.1. An example plot of calculated angles is presented in Fig. 14. This is a segment of 50-60 seconds from data set 1 for person 1, and it contains the simple movements (S1-S4 from Table 4).

These angles will be used in training sets for pattern recognition methods and as reference for pattern recognition performed on validation sets and test sets (see Section 4.6).

The radial/ulnar deviation angle contains some noise, though not more than 3-4 degrees, while the other angles are very smooth. The reason for the noise is unknown. All in all the angle calculations seem to work properly. In the future, inverse kinematics may make more accurate angles (see Section 9), but the method used is probably good enough as input to the estimation procedures.

We had some trouble with the finger flexion/extension angle, because the thumb markers (MCP1, DIP1) were placed too close to each other and were sometimes interchanged in the results. A solution to this was to do new recordings until the result was correct, but an easier option for future studies might be to place the DIP1 marker a bit further out on the thumb.

Some of the angles have a broken graph after 60 seconds, and this is due to a couple of markers that were not visible at that time. See Section 5.3.2.

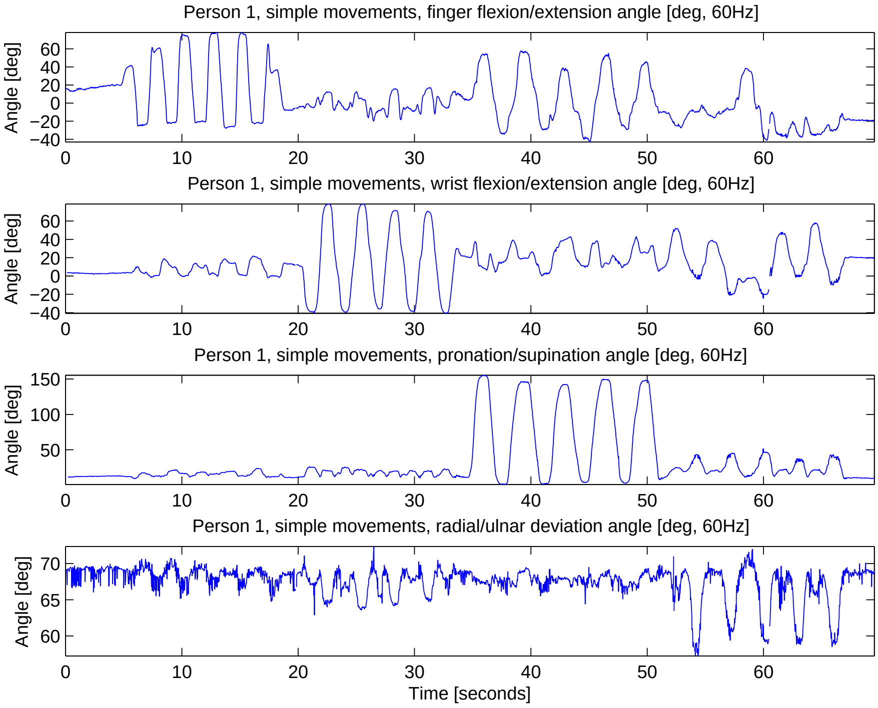


Figure 14: Angle calculation example, simple movements, person 1

6.3 Combined movements

Fig. 15 shows the combined movements (as described in section 4.5) of the complete data set 1 for person 1. The purpose of this plot is to check if our data sets contained all possible combinations of two simple movements. This example data set seems to be a good set of the possible combined movements, although combinations of more than two simple movements are not checked.

Note that the radial/ulnar deviation is usually restricted by the other movements, so the spanned space will not be rectangular. The results containing radial/ulnar deviation are also affected by noise (see Fig. 14).

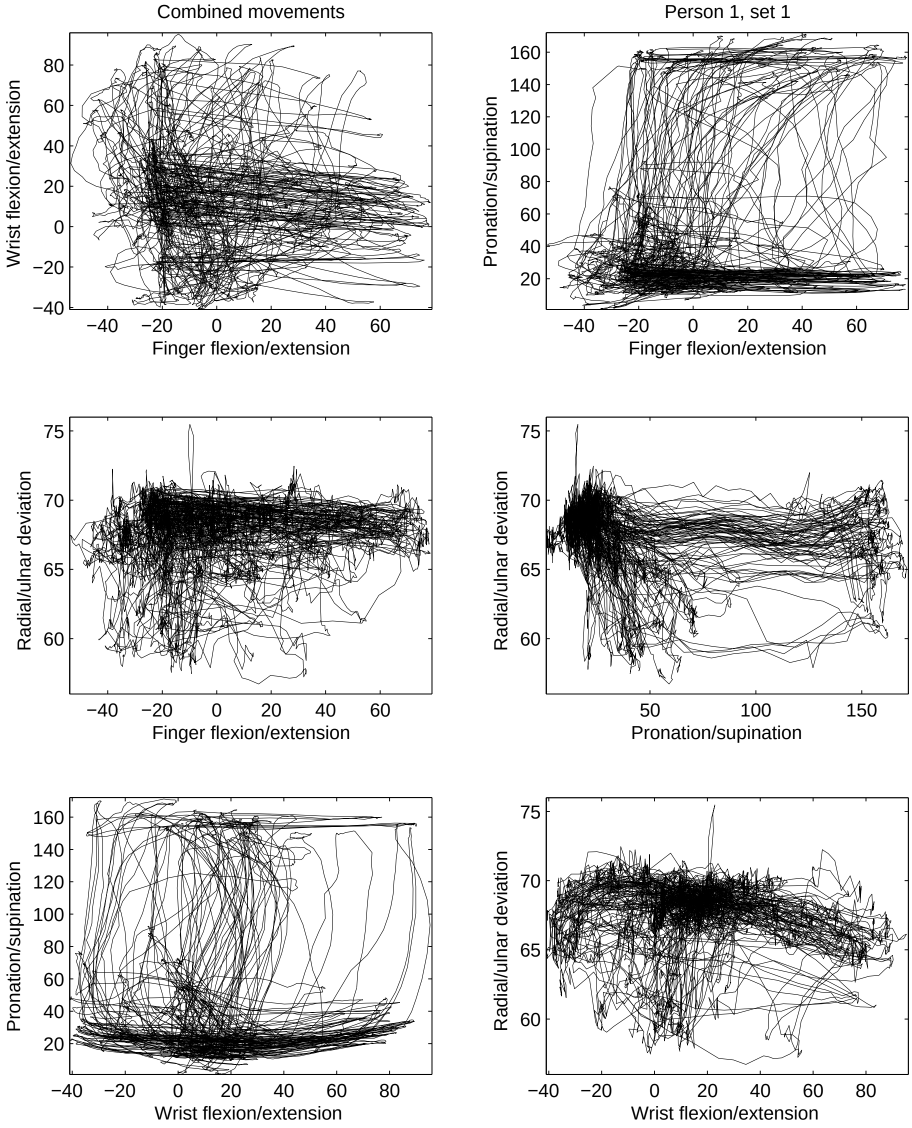
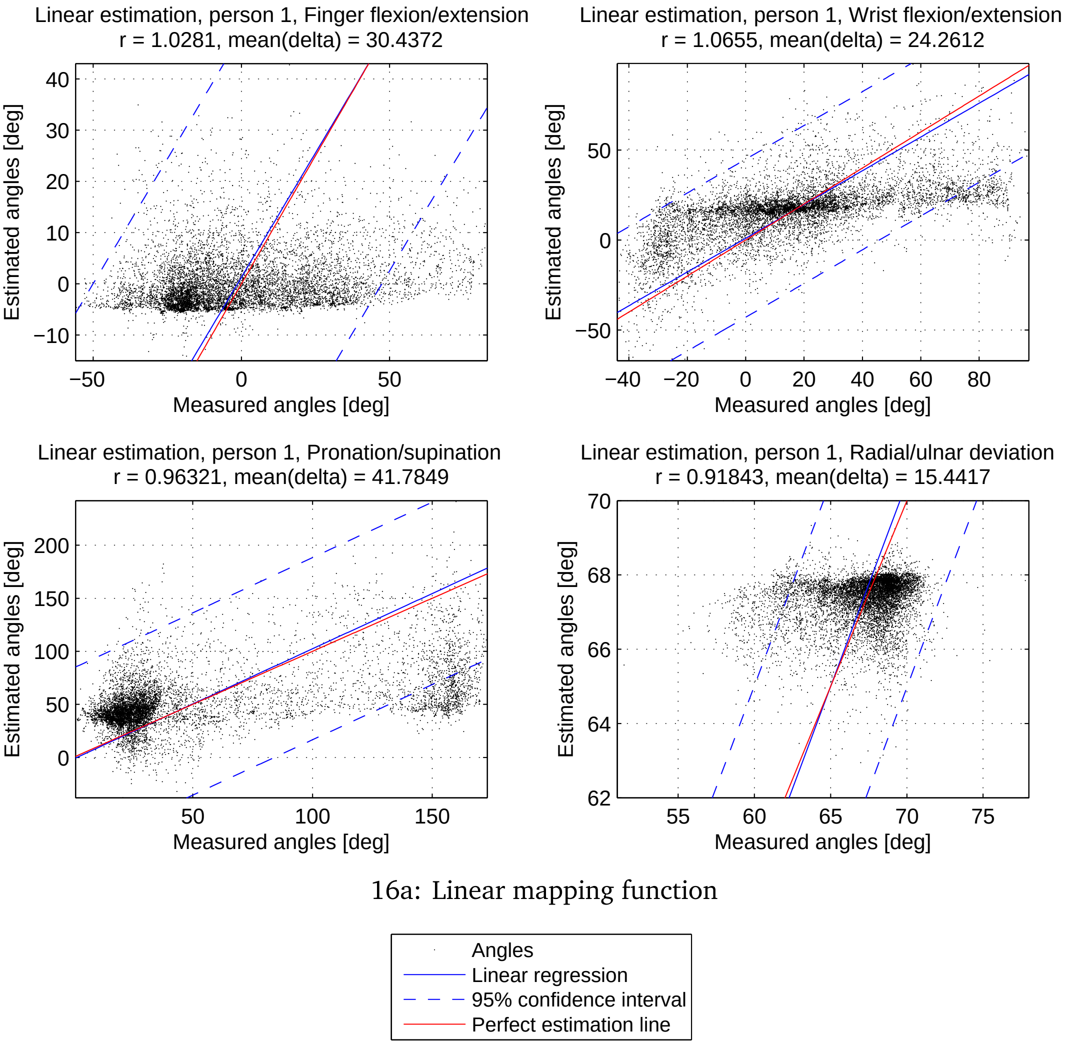


Figure 15: Combined movements, person 1

6.4 Pattern recognition

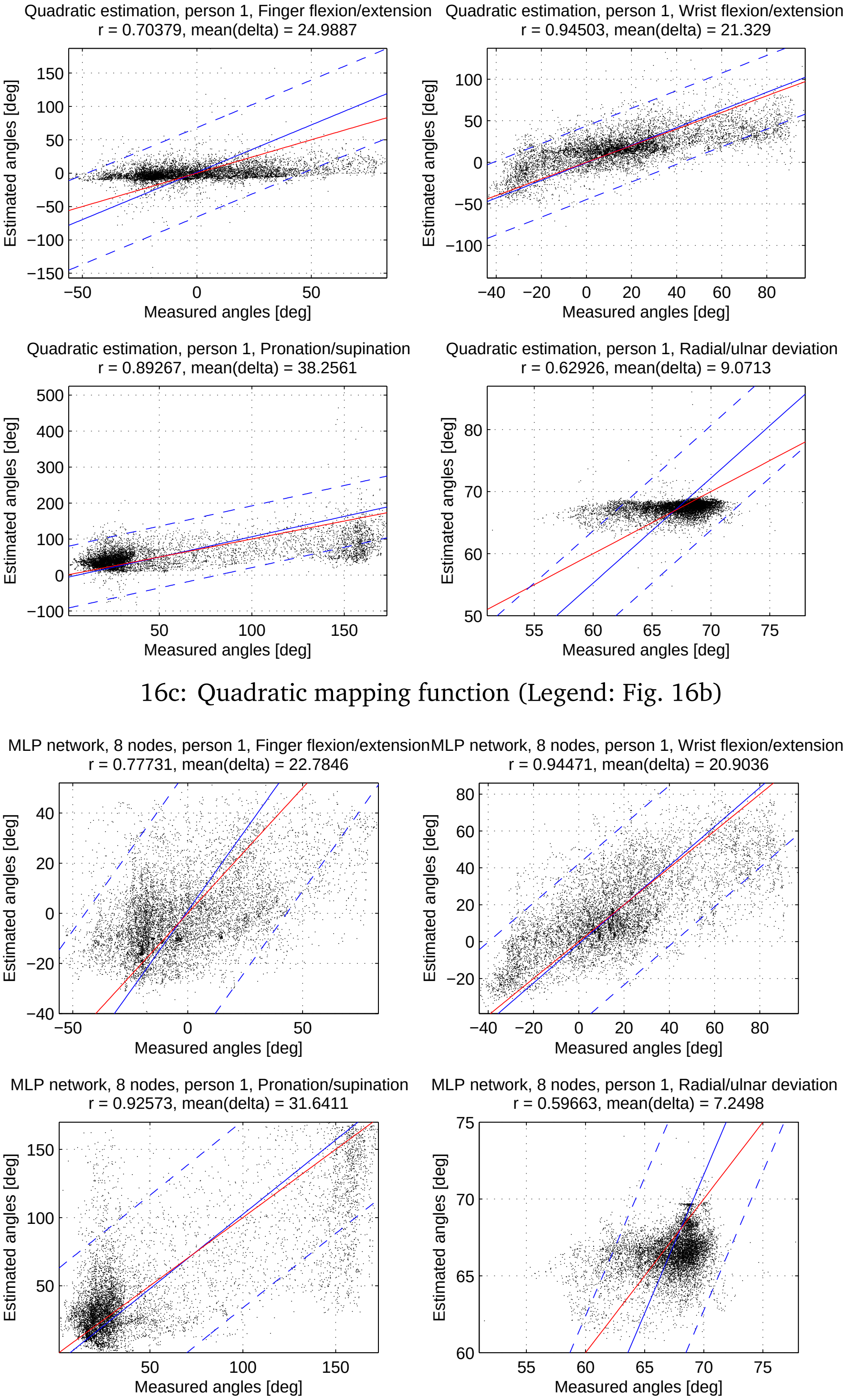
Measured angles (calculation in Section 5.1) vs. estimated angles (using all three methods described in Section 5.2) are presented in Fig. 16 on pp. 32–33. Accompanying linear regression lines and 95% confidence intervals are presented in the plots, and a line for perfect estimation (estimated angles = measured angles) is also plotted for reference. The r values indicate the slope of the regression lines, while the $\mathrm{mean}(delta)$ values indicate the mean distance from the regression lines to the 95% confidence intervals, as a measure of the spread.



16a: Linear mapping function

16b: Legend

Figure 16: Estimated vs. measured angles, validation set, person 1



16c: Quadratic mapping function (Legend: Fig. 16b)

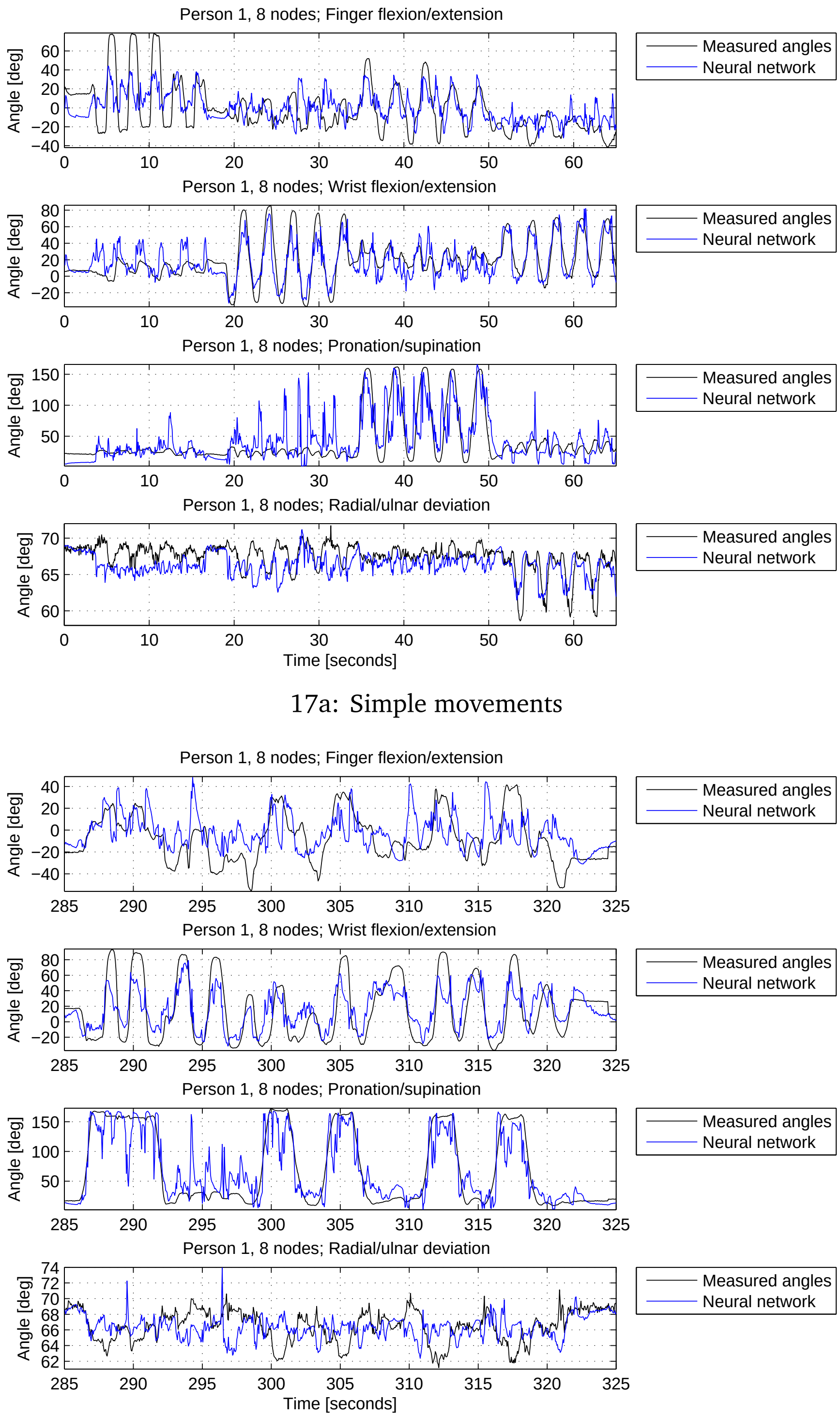
16d: MLP network, 8 nodes (Legend: Fig. 16b)

Figure 16: Estimated vs measured angles, validation set, person 1

Example plots of the estimation are presented in Fig. 17. These are segments of 50-60 seconds from the validation set; one segment of simple movements and one segment of combined movements, to show the variations. Both are for person 1, using an MLP network of 8 nodes. The training set and the validation set were in this case made at the same day. The results for person 2 and 3 were slightly worse than for person 1.

Two more example plots of the estimation are presented in Fig. 18. They compare the estimation results for the training set, the validation set and the test set. In all cases only the wrist flexion/extension angle is represented, and the results are for person 1, using an MLP network of 8 nodes. The training set and the validation set were made at the same day while the test set is from another day (set 3). The plots are from the same periods of all sets, but they contain slightly different movements since the movement order was different in the three sets.

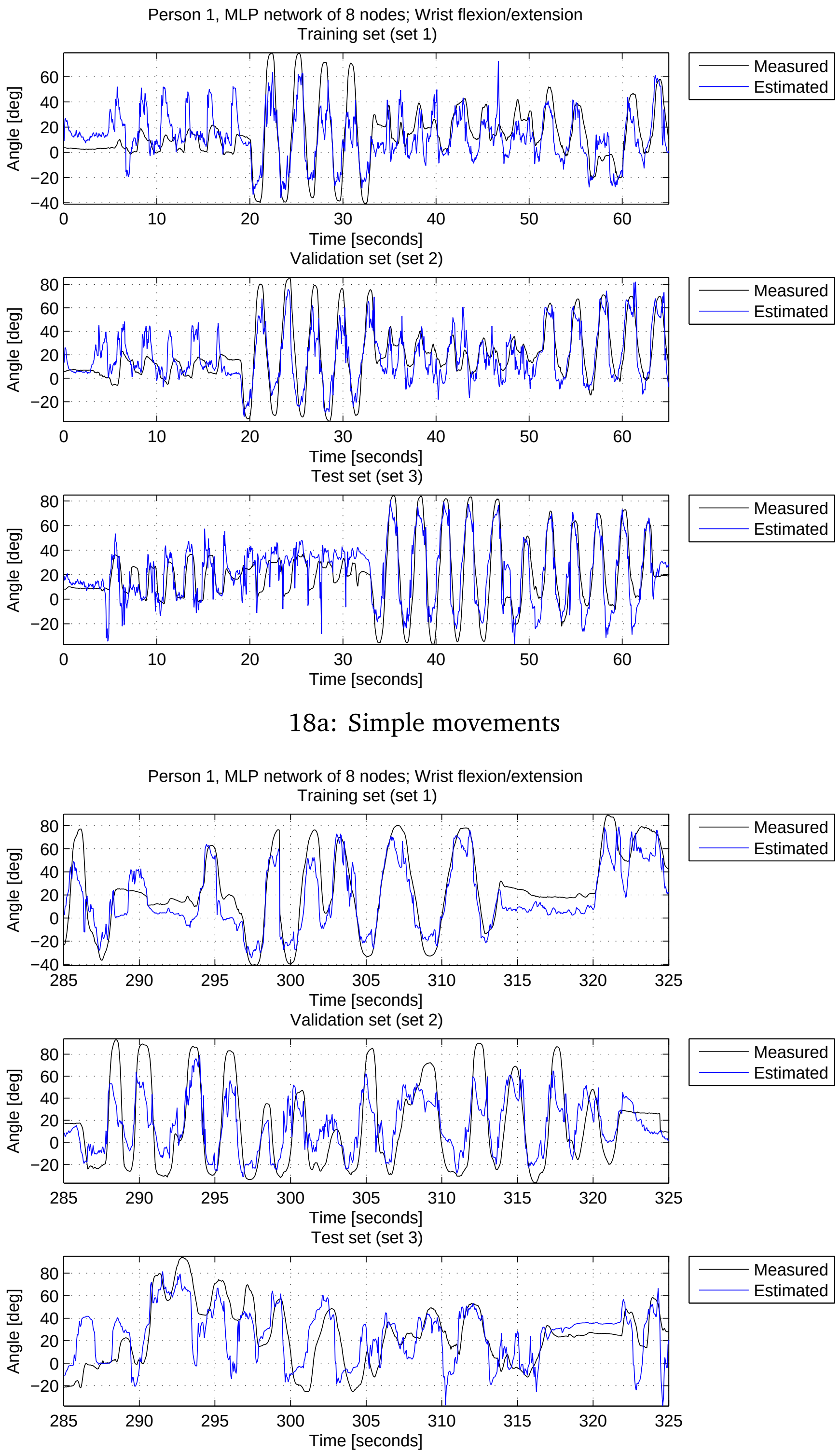
Note the errors in the estimate of pronation/supination angle after for example 20-30 seconds and 290-300 seconds. Errors occur also when the angle is close to the maximum range of motion (≈ 180 degrees). All these errors are probably the same as already seen in Fig. 16d.



17a: Simple movements

17b: Combined movements

Figure 17: Estimated and measured angles, validation set, trained MLP network of 8 nodes, person 1



18a: Simple movements

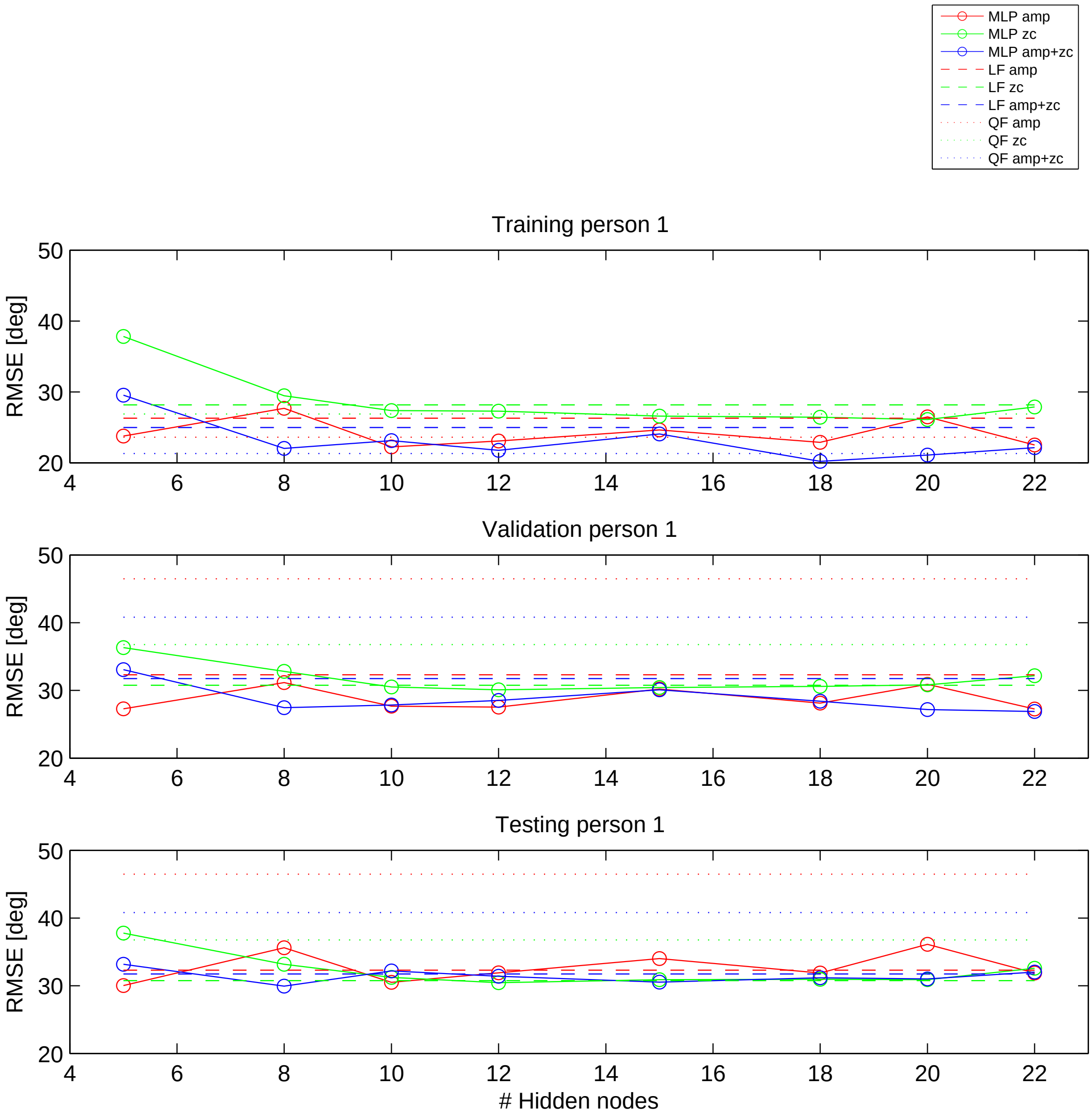
18b: Combined movements

Figure 18: Comparison plot for training-validation-test sets (set 1-2-3), wrist flexion/extension, MLP network of 8 nodes, person 1)

6.4.1 RMS error

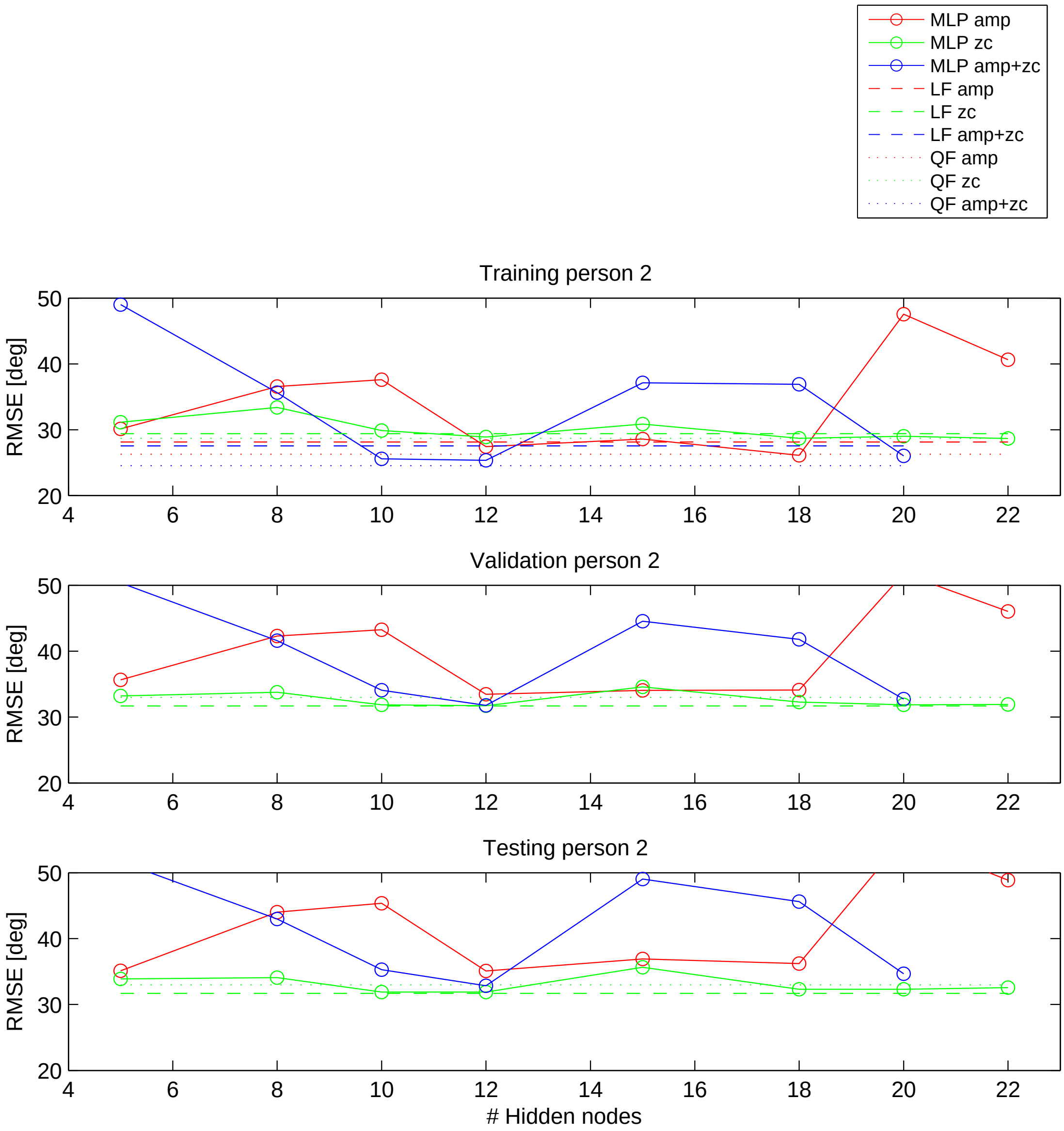
The root mean square (RMS) error values for all persons are presented in Fig. 19 on pp. 37–39. For the MLP network estimation, this value represents the average of all six possible combinations of training, validation and test sets.

The terms amp, amp+zc and zc in these plots refer to the different EMG feature sets described in Section 5.3.



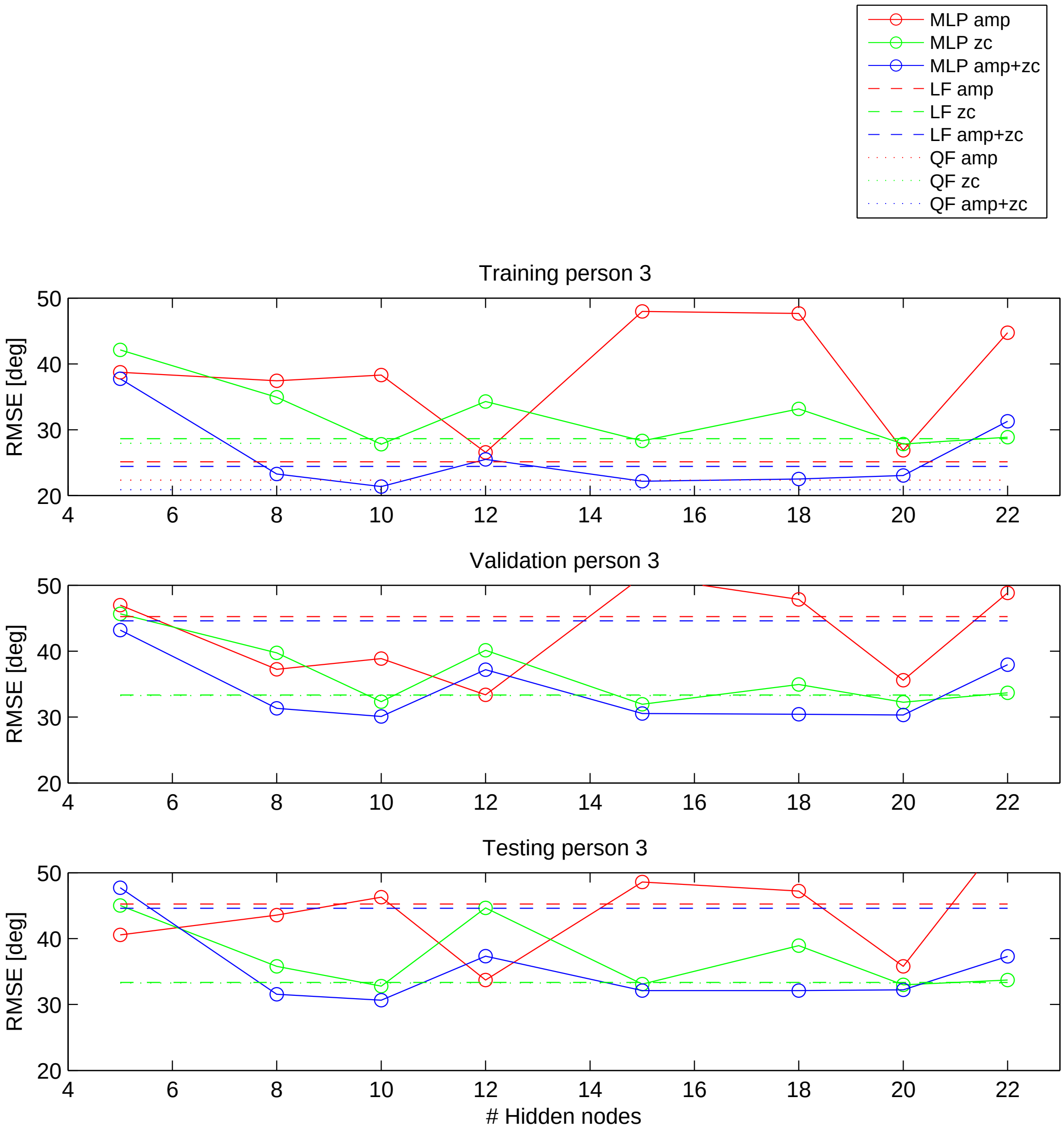
19a: Person 1

Figure 19: RMS error (average of all six training processes for each number of nodes in the MLP network)



19b: Person 2

Figure 19: RMS error (average of all six training processes for each number of nodes in the MLP network)



19c: Person 3

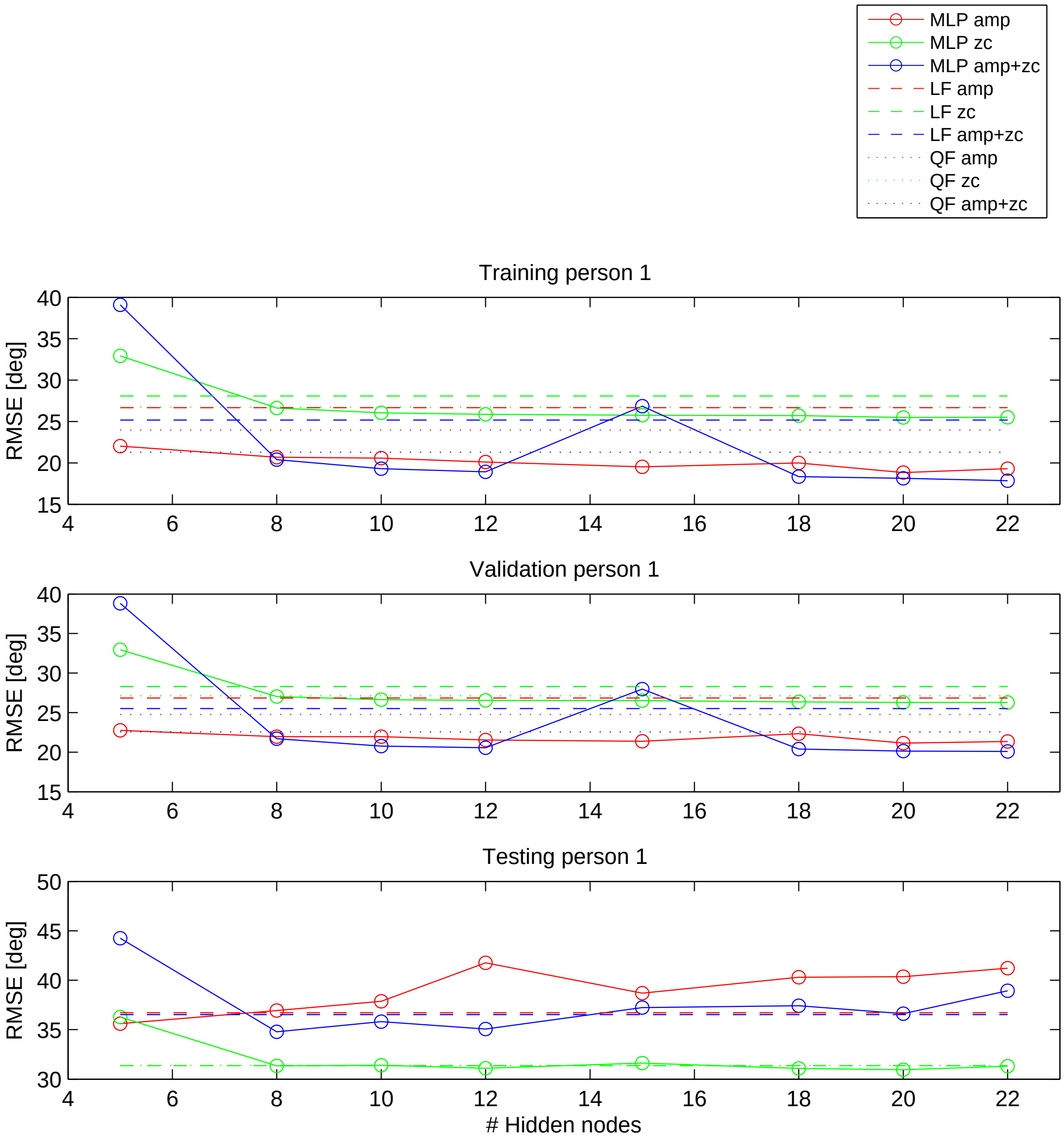
Figure 19: RMS error (average of all six training processes for each number of nodes in the MLP network)

Because the RMS error of these plots (Fig. 19) was quite high, especially for the validation and test sets for all persons, new plots were made by restricting the training and validation sets to be from day 1 (set 1 and 2) and the test set from day 2 (set 3). See table 5. This results in only two combinations.

The purpose of these plots was to check if set 3 resulted in stopping the training procedure too early, because set 3 was too different from set 1 and 2. The `train` method of Matlab's Neural Network Toolbox stops training when the error for the validation set starts to increase. If the validation set is too different from the training set, the training will stop already after the initialization, and the resulting angle estimates will only be the initial values.

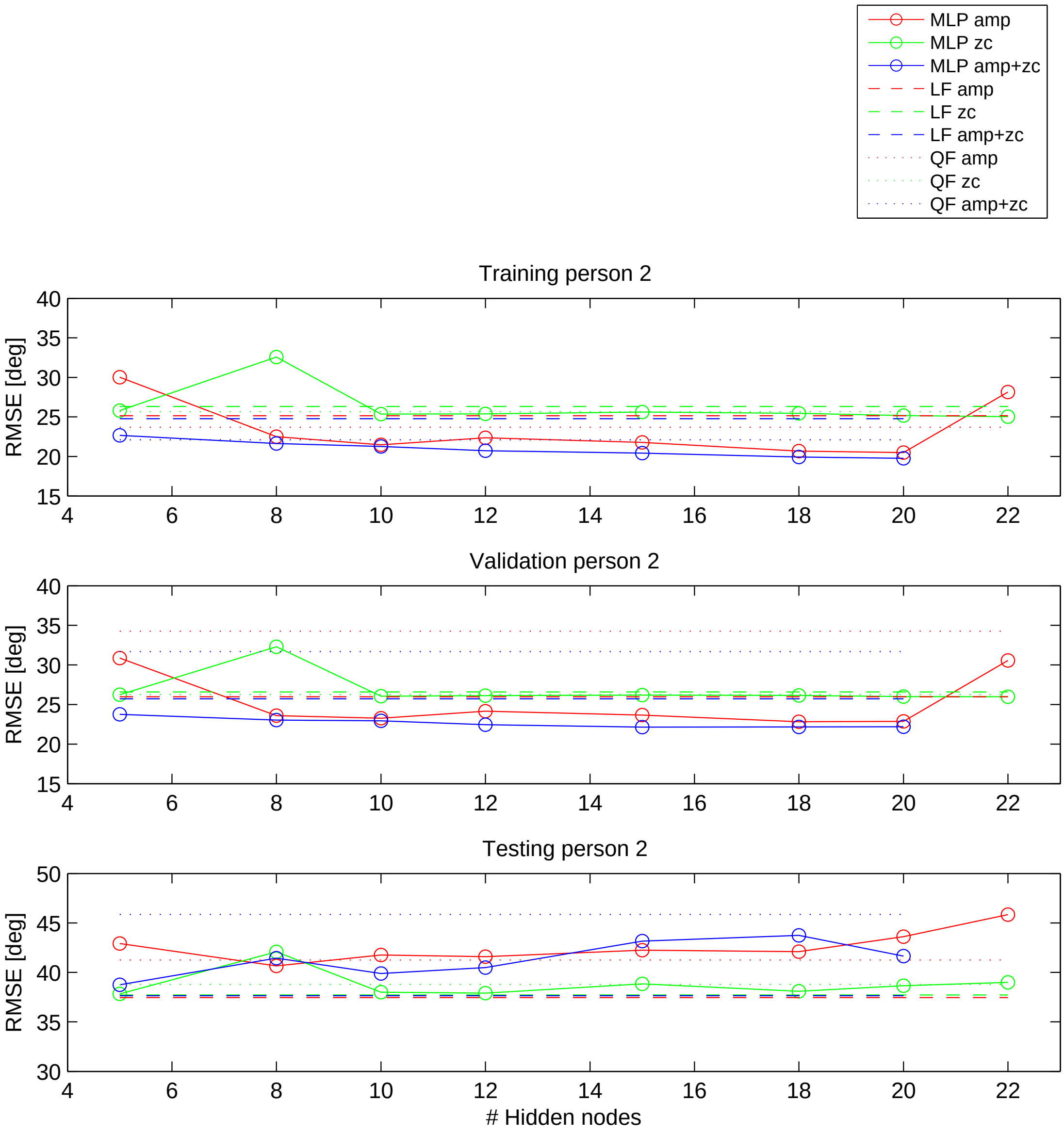
As we see in Fig. 20 on pp. 41–43, this may have happened in some cases, because the RMS error of these combinations is lower than the mean RMS error of all combinations. Compare especially Fig. 19b and 20b, which show the same person with completely different results.

Note that the angle axes in the plot for person 3 are different from the other two plots, because the results were quite different.



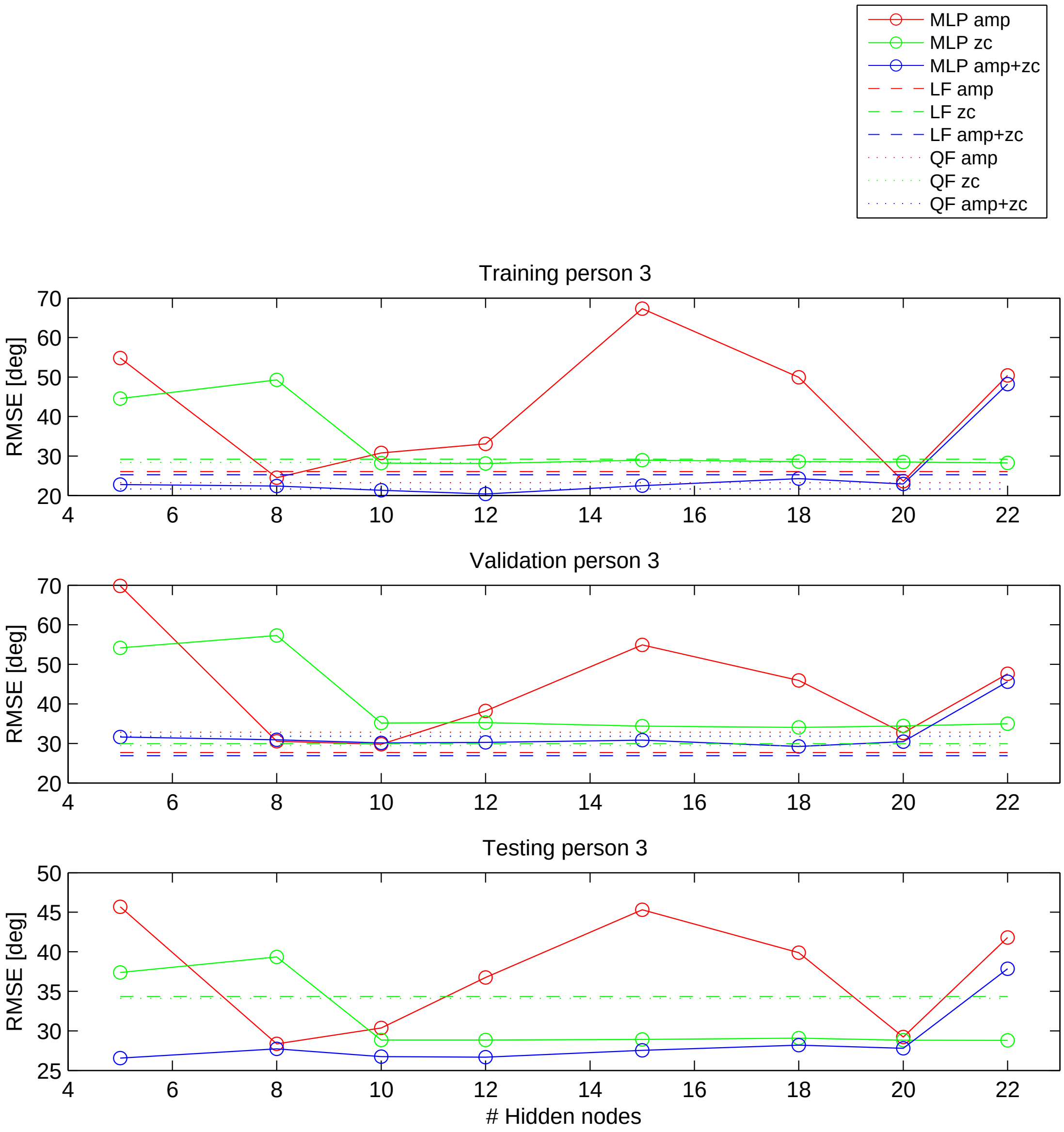
20a: Person 1

Figure 20: RMS error (average of training combinations 123 and 213)



20b: Person 2

Figure 20: RMS error (average of training combinations 123 and 213)



20c: Person 3

Figure 20: RMS error (average of training combinations 123 and 213)

So far, the RMS error plots have only been calculated as a mean value for all four angles. Since the angles have very different range of motion (ROM), for pronation/supination ≈ 0–170 degrees and for radial/ulnar deviation ≈ 55–75 degrees (for person 1), it was decided to make a new example plot where the RMS error is calculated separately for each angle. Fig. 21 shows the specific RMS error for each angle. This is for person 1, the EMG feature set was only the AAV (see Section 5.3) and the MLP network had 8 nodes.

The training and validation sets are from day 1 (set 1 and 2) and the test set from day 2 (set 3), in the same way as in Fig. 20a.

Table 7–8 show the RMS error in percentage of the motion range, for the validation set and the test set. These are approximate values and are based on the data from person 1 trained with an MLP network of 8 nodes. Note that the values are measured in Matlab, not in this small figure. For a closer look and similar plots for the other test persons, see Appendix B.

Angle	ROM [deg]	RMSE [deg]	RMSE [% of ROM]
Finger flexion/extension	$(-40, 80)$	23	19
Wrist flexion/extension	$(-40, 80)$	19	16
Pronation/supination	$(0, 170)$	32	19
Radial/ulnar deviation	$(55, 75)$	2.5	13

Table 7: RMS error, person 1, validation set

Angle	ROM [deg]	RMSE [deg]	RMSE [% of ROM]
Finger flexion/extension	$(-40, 80)$	24	20
Wrist flexion/extension	$(-40, 80)$	27	23
Pronation/supination	$(0, 170)$	63	37
Radial/ulnar deviation	$(55, 75)$	5	25

Table 8: RMS error, person 1, test set

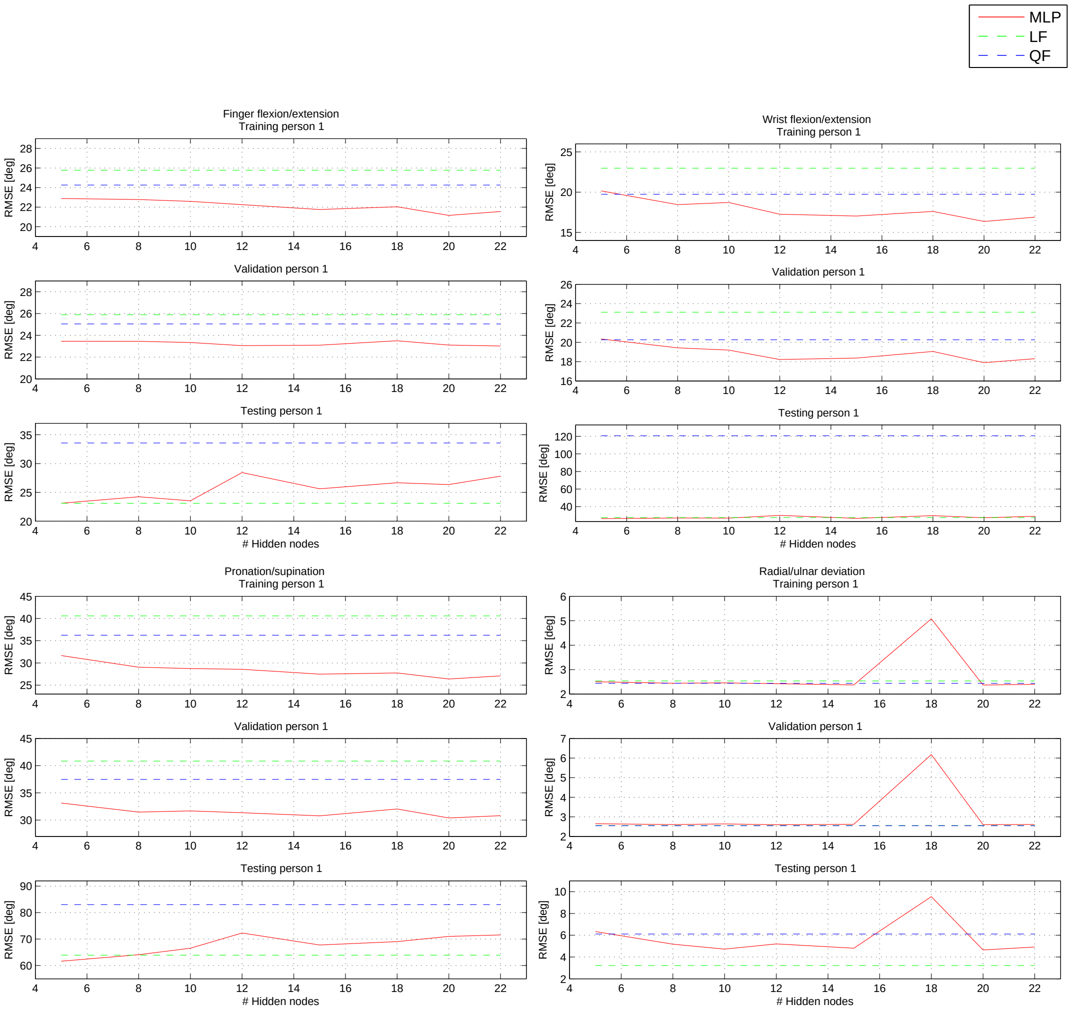


Figure 21: RMS error, separate angles, person 1 (average of training combinations 123 and 213)

7 Discussion

The comparison of estimated angles and measured angles, in Fig. 17–18, are promising. Although there are some spikes and errors, the estimated angles follow the measured angles quite good, and it seems like some kind of low-pass filtering of the estimated angles could be a solution to minimize these errors. See Section 9 for notes about this filtering.

The RMS error, for pattern recognition methods trained on six different combinations of data sets (Fig. 19 on pp. 37–39), is not satisfactory low. An RMS error of $\approx$ 30–35 degrees (for person 1) is too much compared to the range of motion ($\approx$ 0–170 degrees for pronation/supination and $\approx$ 55–75 degrees for radial/ulnar deviation, see Fig. 14 for an example). A possible reason is that the MLP network training procedure terminates extremely early if the validation data is too different from the training data, and this seems to happen when the validation data is recorded in another day than the training data. The RMS error plots in Fig. 20 on pp. 41–43, which are restricted to training and validation data from the same day and test data from another day, perform better for the validation data and just slightly worse for the test data. The RMS error is $\approx$ 20–25 degrees for validation data and 35 degrees for test data (for person 1); this is much, but it should be compared to the range of motion.

The RMS error was plotted separately for each estimated angle, as in Fig. 21. It is now possible to calculate the RMS error compared to the range of motion, and the example in Table 8 shows that the RMS error for the test set varies from $\approx$ $20\,\%$ for finger flexion/extension to $\approx$ $37\,\%$ for pronation/supination. It is not satisfactory, but we do not know exactly how these errors will be experienced by the user of a real prosthesis. It remains to see if the amputee will adapt to the errors, in a conscious or unconscious manner.

The EMG processing needs to be optimized. The amplitude of the data sets seems to vary, especially from one day to another, and this is probably due to varying electrode contact and skin conductance. It is a problem when we use the averaged absolute value (AAV) EMG feature set, because the non-linear deadband filter (see Section 5.3.1) is amplitude dependent. For data set 3 for person 3, almost no signals came through the filtering, and it affected the training of the MLP network. This might be solved with better EMG filtering procedures and by choosing other EMG feature sets.

The linear regression line in Fig. 16 on pp. 32–33 should be close to $r \approx 1$, but it is not always close. In the worst case we have $r \approx 0.6$ for radial/ulnar deviation. Since it appears in all the pattern recognition methods this may be a result of differing EMG amplitudes in the validation set compared to the training set. This is a another good example of why we should try to include EMG signal features that are insensitive to amplitude changes due to varying skin conductance.

Since the results for the zero-crossing (ZC) feature are not much worse and sometimes better than the AAV feature, it seems like the zero-crossing feature contains useful information for the pattern recognition. Thus, the MLP networks also performs better when the AAV feature is combined with the ZC feature as input. This is in fact promising - it indicates that the use of other features, preferably insensitive to conductance changes of the skin, may be used. There exist many other possible EMG signal features that can be combined and optimized for this use. 3,4 or even 10 different EMG signal features might be combined as input, and if they are chosen in the right way they may complement each other. A combined input will possibly contain more information than just one of these features. The extra information can result in much better pattern recognition results. See Section 9.

All pattern recognition methods used in this thesis are easily adapted to more complex inputs than an EMG signal feature set. More features may be used; for example the elbow angle or pressure/force measurements inside the prosthesis. Both of these features contain information about noise caused by elbow motion or forces applied on the prosthesis. As long as these features contain information that complement the EMG signals, they may be useful in the pattern recognition. This is another topic for future work, see Section 9.

The ZC feature tried (see Section 2.5.2 and Fig. 13 on page 29) contains noise which seems to run through the pattern recognition and result in a noisy output signal. Low-pass filtering the ZC feature may help on this.

In Fig. 16, which shows the estimated angles versus the measured angles for a validation set, it is interesting to see the lines of the 95% confidence intervals - they actually show that the linear mapping function (LF) has a wider spread of estimation than the MLP network, although the MLP network has larger root mean square (RMS) error (see Fig. 20a). The quadratic mapping function (QF) performs in the middle between these. However, the RMS error plots (Fig 20 and 21) show clearly that the QF have problems on the test set. The error for QF is sometimes low and sometimes more than the double of the other methods, especially for the test set. The test set is the most important, since it was recorded on an other day than the other sets, and a real prosthesis should function properly also the day after training. The QF may be too adapted on the training set and will thus perform bad on other sets; an over-trained pattern recognition method. Fig. 16a indicates that the LF may be under-trained, and this is not surprising since the relationship between EMG signals and upper-limb angles is not a linear relationship. Anyway, the LF performs well in some cases, and Fig. 21 shows that the RMS errors for LF and MLP network are close.

In Fig. 16d we see that the estimates of pronation/supination are sometimes far away from the real angles, and this appears usually when the measured angle is close to its maximum or minimum. A possible reason is that when you are close to a joint's minimum/maximum range of motion, the antagonists will be activated, and this might cause the estimate to be smaller than it should be. We do not know how this error will be experienced by a prosthesis user, but we would like to find a method for reducing this problem.

A possibility to get less noise and better MLP network outputs, is to do some kind of averaging of the output from several MLP networks (of equal or different sizes) and use this as a common output. This can be the mean value of all outputs, the mean value when extreme output values are neglected, or the optimum value can be found using other statistical methods. This should be tested, see Section 9.

In this study, only 2-6 MLP networks of each size were tried, and this was not enough to find the optimum number of nodes. We should try to train many MLP networks of each size and find the mean RMS error for each size. If we do this for enough networks, we may find a clear trend and choose the optimum number of nodes with the least RMS error.

In some special cases (for example in Fig. 20a, see the MLP network of 15 nodes with combined AAV and ZC features, and in several cases in Fig. 20c), the results for the training set are much worse than for MLP networks of other sizes. In these cases the MLP network performs worse also on the validation set and the test set. We do not know why this happens, but we know that MLP networks are never identical because of varying initial values, and some averaging or other statistics on several trained MLP networks could be a solution. See suggestions for future work in Section 9.

According to the example in Fig. 15, our data sets contained a large variety of combined movements. When this is used in training of pattern recognition methods, they will likely be able to recognize both simple and combined movements.

8 Conclusion

A protocol has been developed for the recording of EMG signals and VICON motion measurements in a laboratory. This protocol contains guidelines to assure that recorded data sets are usable in the further work on pattern recognition and proportional prosthesis control.

Three months have been spent in the laboratory to record suitable data sets for this study. Further studies remain on these data sets, but some research was done to check if other EMG signal features than the averaged absolute value (AAV) can be usable as input for pattern recognition. The zero-crossings (ZC) feature was tested as one of these, mainly because it is insensitive to amplitude changes due to varying skin conductance. It was also tested in combination with AAV feature. Although ZC did not always perform superior to AAV, it is likely that other features should be tested and tried in combinations. Other properties measured in the prosthesis, like elbow angle or pressure from the arm on the prosthesis, can also be included.

The results of the estimation methods are not yet convincing. The RMS error is 20–$37\,\%$ of the range of motion, for four angles of the example test set (Fig. 21). Other plots indicate that the high RMS errors may be due to noise and spikes on the output signal, which may quite easily be removed through some filtering procedure or other limitations on the control signal of a prosthesis. We do not yet know how this would affect the RMS error and how it will be experienced by the amputee when using this control signal on a prosthesis.

The EMG signal processing to generate the AAV feature needs optimalization to make it less sensitive to skin conductance variations. The MLP networks need more testing to be able to find the optimum number of nodes in the hidden layer. The estimated angles need processing and filtering to be usable as a proportional control signal for a prosthesis.

We now have a large, suitable data set from the laboratory, which can be used for further work on pattern recognition and multifunction proportional control of prostheses. See Section 9.

The final step will hopefully be implementation of proportional control in a real prosthesis.

9 Suggestions for future work

An extensive work was done to record suitable data sets in the laboratory, and a deep analysis and research may still be done on these. This section explains the interesting topics for future work on this data set and the possibilities for making a real prosthesis.

Filtering of EMG signals The averaged absolute value (AAV) and zero-crossings (ZC) are features that are filtered or should be filtered before being used as pattern recognition input.

The non-linear filtering procedure for AAV needs is sensitive to amplitude variations due to varying skin contact and skin conductance. This may be improved by dynamically updating the width of the dead-band function according to the amplitude.

The dead-band also has a small overshoot which should be removed. It may be possible to make another smooth dead-zone function by using a higher-order function and combining it with a linear function that has the same function value and slope in the meeting point.

The ZC feature contains noise and should be filtered. The treshold value for this feature (see Section 2.5.2) would also need to be optimized; it is not obvious that the treshold value should be zero.

Testing other EMG signal features The methods used in this project uses the averaged absolute value (AAV) and zero-crossings (ZC) feature of the EMG signal as input. Other features are described but not yet tested for proportional control, see Section 2.5. Among the features tried for on/off control (classification) in Boostani and Moradi (2003), the best result was from a feature which was not completely described, but with some research work it might be reconstructed and tried on proportional control or combined with other features. As for AAV and ZC, all the described features will probably need some filtering/smoothening before being used as pattern recognition input.

Risdal (2006) describes a technique for selection of feature sets. Start by choosing the feature that performs best on its own, and add the next best features one by one until the performance curves flatten out. This may be a suitable method for choosing the best feature set. The performance curves can be based on RMS error calculated for multiple MLP networks of equal or different sizes.

Additional input data The upper limb has several properties that can be measured; not only the muscle force measured through EMG. The elbow angle is easy to measure, and pressure sensors inside the prosthesis can also easily be used as input to the pattern recognition methods. The elbow angle affects which muscles are used for each movement, so it contains useful information for the estimation. Pressure sensors will feel the muscle contractions in another way than EMG electrodes, and this information may be a good supplement. The pattern recognition methods from this study are already suitable for more input data; they do not need any change to take advantage.

The technique of measuring noise and artifacts to improve the estimation, reminds of the multi-channel recursive adaptive matching pursuit (MC-RAMP) algorithm for electrocardiogram (ECG) measuring, described by Husøy et al. (2002). A similar technique may be improve the EMG measuring.

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.

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.

Improvements of pattern recognition methods More pattern recognition methods should be tried and tested. There are many different methods that may give different results, so all of these should be tried and the best of these should be tested more.

The number of nodes in the MLP network may be optimized. We need to do many trainings on different MLP networks of each size, until we see a clear trend on what number of nodes is optimal.

Different filtering procedures on the control signals, like low-pass filtering, may improve the control and should be tested and evaluated.

Recurrent or dynamic MLP networks may be used. Feedback connections in the MLP network will introduce states that are rememberes in the nodes; thus the network will be able to behave like a dynamical system and may for example converge to a low-pass filtered output if this is optimal. The training procedures for recurrent MLP networks are not as simple as for normal MLP networks but would be explored.

Statistical methods for multiple MLP networks A possible method to get better and more reliable outputs from the MLP network, is to use more than one network. Several MLP networks can be trained, either on the same training set or on different training sets from the same person, and the outputs of these network can be averaged or applied to other statistical methods. For example, 18 MLP networks can be trained, distributed on three different training sets and different number of nodes. The most extreme outputs (for example the two highest and two lowest) can be neglected when calculating the mean output. The output may contain less noise (if the noise was random) and give a more accurate estimate.

Subspace estimation Techniques exist for subspace estimation; methods that estimate the order and structure of a system. In this study, this would be the order and structure of the upper limb, from nerve signals to positions/angles of the wrist and fingers. This estimate might be a help in choosing the optimal pattern recognition method. See Schneider and Willsky (2000).

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.

Visualization It would be nice to be able to test the different control methods on a virtual hand, for example in Matlab, or to have some other good visualization of the result. This would make it easier to try, test and validate the different pattern recognition methods and proportional control methods, before the implementation in a real prosthesis.

Realization It must be developed a procedure for training the pattern recognition methods also on amputees, because they do not have a hand on which we can measure the angles with VICON. Some kind of visualization may be required. An idea is to visualize a virtual hand doing a set of movements, while the amputee tries to follow it.

Finally, the system can be implemented in a real prosthesis.

Alternative applications The methods developed in this study can be used also on other problems; not only upper-limb prostheses. They can be used on other joints like the elbow and the knee. We have applied methods for a man-machine interface which have several possible applications in robotic telemanipulation.

10 Bibliography

Abul-Haj, C. J. and Hogan, N. (1990), ‘Functional assessment of control systems for cybernetic elbow prostheses - part i: Description of the technique’, *IEEE Transactions on Biomedical Engineering* **37**, 1025–1036.

Almström, C. (1977), An electronic control system for a prosthetic hand with six degrees of freedom, Technical Report 1, Chalmers University of Technology, Göteborg, Sweden.

Almström, C. and Herberts, P. (1980), ‘Myoelectric control of multifunctional hand prosthesis - a clinical assessment’, *Läkartidningen (Swedish medical journal)* **77**, 768–770.

Boostani, R. and Moradi, M. H. (2003), ‘Evaluation of the forearm emg signal features for the control of a prosthetic hand’, *Physiological Measurement* **24**, 309–319.

Bronzino, J., Enderle, J. and Blanchard, S. (2005), *Introduction to Biomedical Engineering*, 2$^{\text{nd}}$ edn, Elsevier Academic Press, California, US.

Dahl, H. A. and Rinvik, E. (1999), *Menneskets funksjonelle anatomi (Functional anatomy of humans) - in Norwegian*, 2$^{\text{nd}}$ edn, Cappelen Akademisk forlag, Oslo, Norway, pp. 453–498.

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

Fougner, A. (2006), *Proportional myoelectric control of a multifunction upper-limb prosthesis*, Norwegian University of Science and Technology, Trondheim, Norway. Term project, Norwegian University of Science and Technology, Trondheim, Norway.

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

Gunnerød, J. (2005), *FES-basert regulering av ledd ved hjelp av kamerabasert posisjonsmåling (FES-based joint control using camera-based position measurements) - in Norwegian*, Norwegian University of Science and Technology, Trondheim, Norway.

Hudgins, B., Parker, P. A. and Scott, R. N. (1993), 'A new strategy for multifunction myoelectric control', *IEEE Transactions on Biomedical Engineering* **40**, 82–94.

Husøy, J. H., Eilevstjønn, J., Eftestøl, T., Aase, S. O., Myklebust, H. and Steen, P. A. (2002), 'Removal of cardiopulmonary resuscitation artifacts from human ECG using an efficient matching pursuit-like algorithm', *IEEE Transactions on Biomedical Engineering* **49**, 1287–1298.

Kalman, R. E. (1960), 'A new approach to linear filtering and prediction problems', *Transactions of the ASME–Journal of Basic Engineering* **82**(Series D), 35–45.

Kestner, S. (2006), 'Defining the relationship between prosthetic wrist function and its use in performing work tasks and activities of daily living', *Journal of Prosthetics and Orthotics* **18**(3), 80–86.

Light, C. M., Chappell, P. H., Hudgins, B. and Englehart, K. (2002), 'Intelligent multifunction myoelectric control of hand prosthesis', *Journal of Medical Engineering & Technology* **26**, 139–146.

Mann, K. E., Werner, F. W. and Palmer, A. K. (1989), 'Frequency spectrum analysis of wrist motion for activities of daily living', *Journal of Orthopaedic Research* **7**, 304–306.

MedlinePlus Medical Encyclopedia (2007), *Electromyography*, Medline. [Accessed 30 May 2007].
URL: *http://www.nlm.nih.gov/medlineplus/ency/article/003929.htm*

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

Miyano, H. and Sadoyama, T. (1979), 'Theoretical analysis of surface emg in voluntary isometric contraction', *European Journal of Applied Physiology and Occupational Physiology* **40**, 155–164.

Muzumdar, A., Lovely, D. F. and et al (2004), *Powered Upper Limb Prostheses. Control, Implementation and Clinical Application.*, Springer-Verlag, Oslo, Norway, pp. 17–33.

Nazarpour, K., Sharafat, A. R. and Firoozabadi, S. M. (2005*a*), 'A novel feature extraction scheme for myoelectric signals classification using higher order statistics', *Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering* pp. v–viii.

Nazarpour, K., Sharafat, A. R. and Firoozabadi, S. M. (2005*b*), 'Surface emg signal classification using a selective mix of higher order statistics', *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference* pp. 4208–4211.

Oetiker, T. (2007), *The Not So Short Introduction to $\LaTeX2_\varepsilon$*, GNU Free Software Foundation.

Park, E. and Meek, S. (1993), 'Fatigue compensation of the electromyographic signal for prosthetic control and force estimation', *IEEE Transactions on Biomedical Engineering* **40**, 1019–1023.

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.

Perotto, A. O. (1994), *Anatomical guide for the electromyographer: The limbs and trunk*, 3rd edn, Thomas Books, Springfield, US.

Philipson, L. (1987), The electromyographic signal used for control of upper extremity prostheses and for quantification of motor blockade during epidural anaesthesia, PhD thesis, Linköping University, Linköping, Sweden.

Risdal, M. (2006), Impedance plethysmyography for ventilation and pulse monitoring during cardiopulmonary resuscitation, PhD thesis, University of Stavanger, Trondheim, Norway.

Schneider, M. K. and Willsky, A. S. (2000), 'Krylov subspace estimation', *SIAM Journal on Scientific Computing* **22**(5), 1840–1864.

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

VICON Tracker v1.0 User's Guide (2004), Vicon Motion Systems Ltd., Oxford, UK.

Wikipedia, The Free Encyclopedia (2007), *Electromyography*, Wikipedia. [Accessed 30 May 2007].
URL: *http://en.wikipedia.org/wiki/Electromyography*

Wright, V. (2006), 'Measurement of functional outcome with individuals who use upper extremity prosthetic devices: Current and future directions', *Journal of Prosthetics and Orthotics* **18**(2), 46–56.

Appendix A Source code from Matlab

Comment: All my programs are tested in `Matlab R2006a` and require the `Neural Network Toolbox`.

Appendix A.1 `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.2 secondOrderEstimation.m

```
function [F_e,Fval_e,Ftest_e] = secondOrderEstimation(Tdata,Tval,Ttest,N,Xdata,Fdata,Xvaldata,Fvaldata,Xtestdata,Ftestdata)
% FIRSTORDERESTIMATION  Calculates second-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'*W2*X + W1'*X + w0
%
% where
% X = (x1,x2,...,xN), W1 = (w1,w2,...,wN), w0 is a scalar treshold value
% and
%        |w11  ...  w1N|
% W2 =   |...  ...  ...|
%        |wN1  ...  wNN|
%
%
% We rewrite it as
%
% F_e(X) = W*X
%
% where
% X = (  1;   x1:xN;   x1x1:xNxN;   x1x2:x(N-1)xN;   x1x3:x(N-2)xN;  ...  )
% W = (w0,    w1:wN,   w11:wNN,     w12:w(N-1)N,     w13:w(N-2)N     ...  )
%
% W should be 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 NEURALNETWORK, MAKEDATA, FIRSTORDERESTIMATION

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

%% Estimation 2nd order
% Performing least-squares estimation for each of N signals

% Adjust X
X = [ones(1,Tdata)]; % add 'ones' for calculating W_0 parameters
X = [X; Xdata];
% The for-loop goes on like this:
% X = [X;(Xdata(1:N,1:T).*Xdata(1:N,1:T))];
% X = [X;(Xdata(1:N-1,1:T).*Xdata(2:N,1:T))];
% X = [X;(Xdata(1:N-2,1:T).*Xdata(3:N,1:T))];
% ...
% X = [X;(Xdata(1:1,1:T).*Xdata(N:N,1:T))];
for k=N:-1:1
    X = [X;(Xdata(1:k,1:Tdata).*Xdata(N-k+1:N,1:Tdata))];
end

%Calculate the optimal W
W = X'\Fdata';

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

%% Simulate using validation data
% Adjust Xval in the same way as X was adjusted
Xval = [ones(1,Tval)]; % add 'ones' for calculating W_0 parameters
Xval = [Xval; Xvaldata];
for k=N:-1:1
    Xval = [Xval;(Xvaldata(1:k,1:Tval).*Xvaldata(N-k+1:N,1:Tval))];
end

```

```
% Calculate Fval_e(X), the estimate of Fval(X)
Fval_e = W'*Xval;

%% Simulate using test data
% Adjust Xtest in the same way as X was adjusted
Xtest = [ones(1,Ttest)]; % add 'ones' for calculating W_0
    parameters
Xtest = [Xtest; Xtestdata];
for k=N:-1:1
    Xtest = [Xtest;(Xtestdata(1:k,1:Ttest).*Xtestdata(N-k
       +1:N,1:Ttest))];
end

% Calculate Ftest_e(X), the estimate of Ftest(X)
Ftest_e = W'*Xtest;
```

Appendix A.3 `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 DVD

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

B.1 Report Contains the Master's thesis as a PDF file, and a subfolder with LaTeX source code.

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

B.3 LaboratoryWork Contains all files from the laboratory, divided in five parts. The subfolders are arranged cronologically:

B.3.1 RecordingProcess Contains the marker set definition used for autolabels in VICON.

B.3.2 C3DtoMatlab Contains methods for importing data from VICON C3D files to a Matlab `.mat` file, it contains also `alldata.mat` with all recorded data from the laboratory. The C3D files are thus not included.

B.3.3 GenerateInputs Contains methods for calculating vectors and angles from the 3D coordinates, a file with the calculated vectors and angles included, and methods for plotting them.

B.3.4 EMGprocessing Contains methods and SimuLink diagrams for processing EMG signals, files with all the processed signals and methods for plotting the results.

B.3.5 PrepareForPatternRecognition Contains methods for calculating the zero-crossing feature and preparing all data sets for the pattern recognition. It contains also prepared data files for all test persons.

B.3.6 PatternRecognition Contains methods for pattern recognition and the results of the different methods applied. It contains also the results of the pattern recognition and methods for plotting the results.

All plots are available in pdf, eps and png format in subfolders with the name `/gfx/`.

Regarding the data sets: Most of the file names are self-explaining, for example containing the code of the test person, which recording set (**v1**, **v2** or **v3**), which training-validation-test order (**123**, **312** etc) and/or how many nodes were used in the MLP network.