FOREARM SURFACE EMG SIGNALS RECOGNITION AND MUSCULOSKELETAL SYSTEM DYNAMICS IDENTIFICATION USING INTELLIGENT COMPUTATIONAL METHODS

Dissertation

Zur Erlangung des akademischen Grades

Doktoringenieur
(Dr. –Ing.)

Von
D. I. Abdelhafid Zeghbib

geb. am 20.04.63 in Algerien

Genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik
der
OTTO-VON-GUERICKE UNIVERSITÄT MAGDEBURG, DEUSCHLAND

Gutachter:
Univ.-Prof. Dr.-Ing.habil. Dr.h.c. Frank Palis
Univ.-Prof. Dr.-Ing. habil. Heinz-Ulrich Seidel

Promotionskolloquium

am 07.12.2007

To my mother my father my sisters and my brothers

I shall return again. I shall return
To laugh and love and watch with wonder-eyes
At golden noon the forest fires burn
Wafting their blue-black smoke to sapphire skies.
I shall return to loiter by streams
That bathe the brown blades of the bending grasses,
And realise once more my thousand dreams
Of waters rushing down the mountain passes.

by Claude McKay

Life is nothing but a big struggle, but just keep the faith and focus on your goals.
Don't let life beat you or you will be walking around like zombies.
Keep on pushing, keep on trying, life can be whatever you make it to be.
But life can also be a bowl of cherries with whip cream and apple pie.
I say this again; life is what you make of it.
You can achieve or conquer anything it throws at you,
you can't quit or give up, you have got to keep on working,
look higher some way, some how you are going to make it.

By David Cook

Acknowledgements

I am deeply thankful to my advisor, Professor Frank Palis, who initially encouraged me and introduced me to this fascinating field of Intelligent computational Algorithms as methods and biomedical engineering as application. I am also thankful for his constant support during this research-period. I feel honored to have had a chance to work with him in this research for my Doctor Degree and because of him to have reached this degree. His large experience and huge energy of work in research have provided in me a continuous stimulus for my research.

I have the pleasure to thank also the partners of our common project (symbolic controlling of an artificial hand using myo-electrical signals, Prof. Pierre Yves Glorennec, Prof. F. B. Ouezdou and Dr. Patrick Henaff, did in cooperation between *Otto-von-Guericke* University Magdeburg Germany, *Max Planck* Institute Magdeburg Germany, INSA/IRISA university *Rennes-France* and *LIRIS* Laboratory in Versailles University France.

I should also mention thankful to *Max Plank* Institute, in which I have had the possibility to do all my measurements, through the help of my friend Dr. Thomas Schauer and his Engineer student Robert Salbert who helped me and put the laboratory at my availability.

Of course, I am also grateful to all personals of my department for their support, specially Dr. RiemeKasten, Ms. Honymus and Ms. Krieger, they were wonderful people and their support makes also this work possible.

Germany — Abdelhafid Zeghbib

March 27, 2007

Zusammenfassung

Wie erreichen Menchen die beeindruckende Leistung während der Bewegung?

Diese Dissertation Betrachtet das biologische System der Bewegung, sie wird in zwei Teilen dargestellt. Im ersten Teil steht die Erkennung der gemessenen elektromyographischen Signale der Unterarmmuskeln im Fokus, die den Handbewegungen entsprechen. Der zweite Teil beinhaltet die Betrachtung des Muskulosketal-Systems, welches den dynamischen Kniebeugewinkel unter isotonischen Kontraktionen darstellt. Der dabei gemessene Winkel zwischen dem Schenkel und dem Bein zeigt das Ansprechverhalten der elektrischen Stimulation (FES). Dieses Verfahren ist unter Systemidentifikation bekannt.
Das Verstehen der Funktionen der menschlichen Bewegung bildeten in der letzten Dekade einen Schwerpunkt auf dem Gebiet der Neuromuskulären und Skelettären Systeme innerhalb der Biomechanik. Körperbewegungen stellen eine Interaktion zwischen dem neuro-muskulärem Steuersignal und dem muskuloskeletal Dynamiksystem dar. Viele Elemente des neuromuskuloskeletal Systems wirken so aufeinander ein, dass eine reibungslose und koordinierte Bewegung ermöglicht wird. Das skelettartige System, besteht aus Knochen und deren Verbindungen zu den Muskeln (Sehnen), die das muskuloskeletal System unterstützen - sie übertragen die notwendigen Kontraktionen auf das Skelett und realisieren damit die gewünschten Bewegungen. In dieser Dissertation konzentrieren wir uns zum Einen auf die menschlichen neuromusculären Steuersignale sowie deren Klassifizierung und zum Anderen auf die Identifizierung der muskuloskeletal Dynamik. Diese komplexen Systeme erfordern intelligente Modelle, Neuronal Netz und Fuzzy logic, die durch künstliche Intelligenz sich adaptieren.
Informationen, in Form von Nervenimpulsen, figure 1, wandern nach und von unserem

Zentralnervensystem (Gehirn) entlang unseres Rückenmarks und erlauben uns, unsere freiwilligen Bewegungen des Körpers zu koordinieren. Elektrische Impulse des Gehirns, die über die Nervenzellen den Muskeln übermittelt werden, verursachen die Bewegungen (Kontraktionen) dieser Muskeln. Diese Muskeln reagieren, wenn sie die elektrischen Signale des Gehirns empfangen. Diese elektrischen Signale sind über die Muskeln gemessen und sie sie sind als Electromyographische Signale (EMG) erkannt. Die menschliche Bewegung ist ein komplizierter Prozess und kann in die neuronale

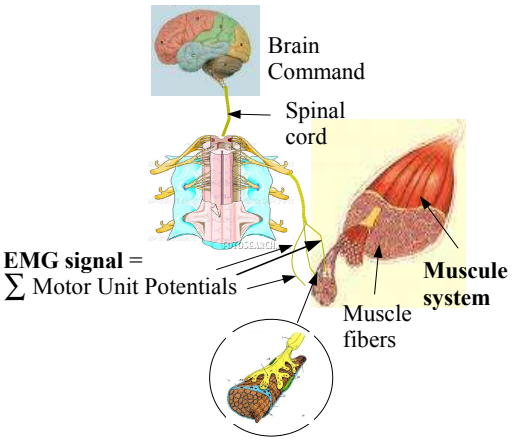

Figure 1: Die Illustration des menschlichen neuromuskulären Signals und des Muskeln Systems als zwei Komponenten für Durchführung der Bewegung

Steuerung, die neuromuskulären Signale und schließlich die Muskelkraft eingeteilt werden. Im Rahmen der vorliegenden Dissertation stehen die zwei letzten Komponente der Bewegung im Fokus:
1) Erkennung der EMG Signale (Klassifizierung) und
2) Die Identifizierung der Muskuloskeletal- dynamischen Belastung.
Das verursachende Verhältnis, zwischen neuromuscular EMG und musculoskeletal Dynamik, wird, in dieser Dissertation, nicht betrachtet. Jedes Teil gilt als allein.
Das erste Ziel ist die Erkennung des EMG, welches die neuromuskulären Signale erkennt und klassifiziert. Dieses neuromuskulär Signal besteht aus einzelnen Muskelaktionspotentialen (MUAPs) von Nerven. Die Summe der Muskelfaser Aktionspotentiale aller Muskelfasern kann mit Hilfe der Elektroden gemessen werden, die auf dem entsprechenden Muskel als Electro-myographsignal (EMG) gesetzt werden. Ein entsprechendes (Online-) Programm zur Erkennung des EMG- Signals ist bereits entwickelt worden.

Das zweite Ziel dieser Dissertation ist, die muskuloskeletal Strukturdynamik zu identifizieren, die für die Bewegung des Körpers verantwortlich ist (Beine und Arme). Solche Bewegungen können mit Functional Electrical Stimulation(FES) produziert werden, sofern die Dynamik zwischen der FES und der sich daraus ableitenden Bewegung bekannt ist. Dieses Studium kann auch für den Ellbogen angewendet werden.
Ein besseres Verständnis dieser zwei Komponenten der Humanbewegungen (movement realisation dynamics), Musculoskeletal Last und Neuromuskulär Rekrutierung, kann körperbehinderten Personen helfen, durch Wiedergewinnung der verlorenen Bewegungsfunktionen eine Verbesserung ihrer Lebensqualität zu erreichen. Sie unterstützt auch den Fortschritt der Rehabilitation und ermöglicht zudem eine bessere Einschätzung und therapeutische Behandlung für körperbehinderte Personen. Diese zwei Komponenten sind in dieser Dissertation separat behandelt worden. Die Entwicklung von technischen Verfahren für die Erforschung des Verhältnisses zwischen beiden Komponenten wird in der vorliegenden Arbeit von großem Interesse sein. Mit Hilfe der ermittelten schwachen freiwilligen Muskelaktivitäten bei Schlaganfallpatienten durch elektromyographische Signale, welche die elektrischen Stimulationen (FES) steuern, kann dieses Verhältnis (der beiden Komponenten) dargestellt werden. Die Steuerung der FES-Signale wird den Patienten helfen, eine korrekte Bewegung der Arme und/oder der Beine durchzuführen. Diese Verfahren helfen das Verhältnis zwischen den mechanischen Bewegungen und den EMG- Eigenschaften zu ergründen.
Die aktuelle Technik erlaubt die technische Annäherung an die Biosignalverarbeitung sowie die Identifizierung solcher komplizierter und dynamischen Systeme wie Muskeln, die als Generator aller Bewegungen des menschlichen Körpers zu betrachten sind.
Fuzzy Logic Systeme und neuronale Netze sind intelligente Methoden, die in dieser Dissertation für die Lösung der Identifizierung und Klassifizierung genutzt worden sind. Sie werden als computerunterstütztes Problem dargestellt, um sie im tagtäglichen komplexen System der Biomedizin anzuwenden.
Diese Signale, die von den Muskeln mittels Oberflächenelektroden gemessen werden, erfordern weitere Berechnungsmethoden für die Erfassung (aquisition), Analyse, Zerlegung (decomposition), und Klassifikation. Der Zweck des ersten Teils ist die Illustrierung der verschiedenen Methodologien und Algorithmen für alle notwendigen Schritte, die für die Finger- und Handbewegungen entsprechend ihren EMG-Signalen zu erkennen sind. Ein Algorithmus für die Klassifikation dieser EMG-Signale konnte bereits in früheren Publikationen vorgestellt werden. Die Klassifikationsergebnisse

dieses vorgeschlagenen Algorithmus werden mit anderen bekanten Berechnungsmethoden verglichen (Fuzzy logic und Neuronale Netz). Diese erste Komponente enthält die Entwicklung aller Verfahren von EMG-Signalen von der Erfassung bis zur Erkennung ihrer entsprechenden Hand- oder Fingerbewegungen mittels Extraktion der relevanten Merkmale und ihrer Klassifikation. Weiterhin soll Patienten mit amputierten Gliedmaßen geholfen werden. Das bedeutet, dass die neuromuskulären Aktivitäten, beispielsweise seiner Unterarmmuskeln, für die Steuerung der Myo-Prothese benutzt werden. Zudem wird die Aktivität seiner Gehirnneurone entsprechend der Neuronen der Unterarmsmuskeln dauerhaft angeregt.
Die zweite Komponente der Realisierung der Humanbewegungen, das Muskuloskeletalsystem, ist von weiterem Interesse. Die Dynamik dieses Systems ist komplex. Daher sollten wir nach einer effektiven Methode suchen, mit der diese komplizierte Dynamik (Motorsystem) modelliert werden kann. Mathematische Modellierungsmethoden (morphologische Modelle) können eine solche Dynamik nicht mit Genauigkeit (fidelity) beschreiben. Als weiterer Beitrag wird ein Hybridalgorythmus vorgeschlagen, um effektiver und schneller eine Lösung (Muskuloskeletalsystem) herbeizuführen. Die Oberschenkelmuskeln werden aufgrund ihrer Grösse betrachtet. Das erleichtert die gewünschten Muskeln zu stimulieren. Des Weiteren ermöglicht die Muskelauswahl die Eindeutigkeit der Stimulation. Anhand eingangs erzeugter elektrischer Impulse und den daraus resultierenden Winkel zwischen dem Knie und dem Oberschenkel können die Modellparameter des "hybrid fuzzy identifier-model" ermittelt werden. Die eistungsfähigkeit des "hybrid fuzzy identifier-model", das eine nicht lineare Input/Output-Dynamik darstellt, hängt von der "fuzzy partition" seines Eingang-Raumes ab. (the initialisation of premise fuzzy sets is an important issue in fuzzy modeling). "Rapid Prototyping" Methode wird durch diesen vorgeschlagenen Algorithmus eingeführt, um die Leistung der Initialisierung der "Fuzzy Sets" durchzuführen. Dieses vorgeschlagenen Hybridalgorithmus besteht aus drei Komponenten: "Rapid Prototyping" Algorithmus, "Gradient Descent" Method, und "Least Squares Estimator". Alle diese drei Teilen sind kombiniert, um diese Modellierungsaufgabe durchzuführen. Des Weiteren ermöglicht die Steuerung der "human knee-joint movements".

Abstract

How do humans achieve the remarkably impressive performance when they move?

The specific aim of this thesis, which considers the biological system "human movement", is presented in two parts. The first part considers the recognition (classification) of measured Electro-myography (EMG) signals of forearm muscles corresponding to hand movements. The second part treats the musculoskeletal system, which is considered by Knee-joint dynamics under non-isometric conditions, in terms of its measured angle between thigh and shank as response for Functional Electrical Stimulation (*FES*) impulses. This procedure is known as system-identification.
Understanding human movement functions is of a great importance in the domain of neuromusculoskeletal systems and biomechanics. Whole-body movement is achieved with help of the interaction between the neuromuscular control signal and musculoskeletal dynamics system. Many elements of the neuromusculoskeletal system interact to enable smooth and coordinated movements. The skeletal system, composed of bones and joint connections with muscles, which complete the musculoskeletal system, apply the necessary driving forces for movement realisation. In this thesis we will focus on human Neuromuscular control signals classification and Musculoskeletal dynamics identification. These complex systems require much knowledge by learning. Hence an improvement of the learning ability, using artificial intelligent methods, is also covered.
The information, in the form of nerve impulses, figure 2, travels to and from our central nervous system (brain) along our spinal cord, allows us to coordinate our voluntary movements of our body. Brain electrical impulses, which are transmitted via nerve cells to the muscles, cause the movement of these muscles. These muscles respond by contracting when the brain's electrical signals reach them. These electrical

signals can be measured over muscles and they are called electromyography (EMG) signals. Generation of human movement is a complex process, involving the following

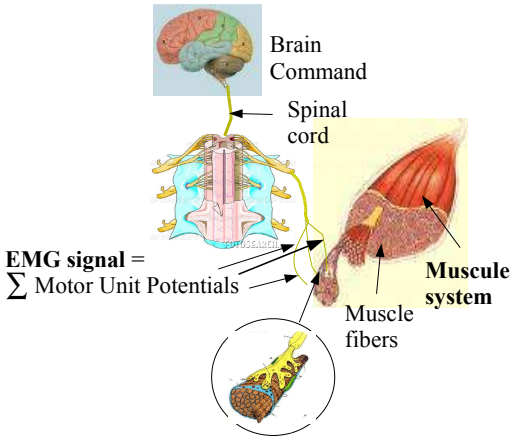

Figure 2: Illustration of human neuromuscular signal and musculoskeletal system as two components of movement realisation

ways: neural command, neuromuscular signals and finally muscle force. This thesis considers the two last components of movement realisation, which are:
1) EMG Neuromuscular recruitment signals recognition (classification), and
2) Musculoskeletal loading dynamics identification.
In this thesis the causal relationships between neuromuscular EMG signals and musculoskeletal dynamics will not be considered. Each part is considered alone.
The first goal, is to recognise and classify the EMG neuromuscular signal. This neuromuscular signal represents the Motor Unit Action Potentials ($MUAPs$) of nerves. The summation of the muscle fiber action potentials from all muscle fibers can be measured with help of electrodes placed on the corresponding muscle as electromyography (EMG) signal. An on-line Algorithm for this part of EMG signals recognition is also developed.
The second goal of this thesis is to identify musculoskeletal structure dynamics, which act as actuators producing the joint torques to drive the body (legs and arms). Such movements can be produced using Functional Electrical Stimulations (FES), if the dynamics between FES and joint torques are known. Although this part of study focuses on walking, using quadriceps muscles, the findings can be generalised to other motor control systems such as elbow joint through biceps and triceps muscles.
A better understanding of these two components of movement realisation dynamics

(musculoskeletal load and neuromuscular recruitment) can help disabled persons in regaining lost function and/or improving their activity of daily living life and for assessing rehabilitation progress. These two components have been studied in this thesis separately. Developing techniques for investigating the relationship between them, in further work, will be of great importance. Such relationship can be illustrated by using the recognition of detected weak voluntary muscle activity, by post-stroke subjects, through electromyography signals (EMG) to control Functional Electrical Stimulation (*FES*) impulses, which will support the patients to accomplish correct leg or arm movements. These techniques help the investigating of the relationship between the mechanics of movement and the characteristics of the EMG signals

The domain of engineers provides efficient technical approaches and tools for biosignals processing and complex dynamic systems identification as muscle, which is the generator of all human body movements. Soft computing includes both neural networks (*NN*) and fuzzy logic (FL) systems represent intelligent approaches, which are used in this thesis for solving the identification and classification problem of such realistic complex systems in biomedical area.

These EMG signals acquired from muscles, through surface electrodes, require advanced computational methods as acquisition, analysis, decomposition, and classification. The purpose of this part is to illustrate the various methodologies and algorithms for all necessary steps used to discriminate the different movements of finger and hand grasps according to their corresponding EMG signals. For the recognition and classification of these EMG signals, a fuzzy-classifier-model algorithm is proposed in this thesis. This classifier-model algorithm, Fuzzy Trimmed Mean Classifier (*FTMC*) uses the trimmed mean method as tool for input space-set initialisation. The results of this algorithm are compared with other known intelligent computational methods. This first part contains the development of all procedures, starting from EMG signals acquisition till the recognition of their corresponding hand/finger movements, using extraction of relevant features and their classification. The main goal of this first part is to help the patient with the amputated hand to keep the neuromuscular activity of forearm muscles, which will be used to manipulate a myoprosthesis, and to keep the virtual neural activity of the brain related also to this activity of forearm's motor unit potentials .

The second component of movement realisation dynamics, which is musculoskeletal dynamics has a great importance. These dynamics are very complex, hence we should

look for an effective method that can model this complex motor system. Mathematical modelling methods-based morphological models cannot describe with fidelity such complex dynamics. For this problem an effective and fast hybrid fuzzy Algorithm for modelling is developed and proposed in this thesis. The quadriceps muscles are used because their dimension, which help to choose the desired muscle to be stimulated. The choice of desired muscle to be stimulated is not possible in case of many small muscles that are located together. The parameters of this hybrid fuzzy identifier-model are obtained using generated Functional Electrical Stimulation (*FES*) impulses as an input set, and the measured knee-joint angle as an output set. The efficiency of this fuzzy identifier-model representing non-linear input-output dynamics depends on the fuzzy partition of its input-space (the initialisation of premise fuzzy sets is an important issue in fuzzy modeling). Hence Rapid Prototyping method is introduced in this proposed algorithm to perform this initialisation of premise fuzzy sets. In this proposed algorithm three techniques: Rapid Prototyping algorithm, Gradient Descent method and Least Squares Estimator are combined as a hybrid algorithm to achieve this modelling task. The main issues of this study, concern the knee-joint dynamics identification, are developed for further control-application of the human knee-joint movements by Functional Electrical Stimulation (*FES*).

Table of Contents

Chapter 1

Introduction

The use of neuromuscular signals and identified Musculoskeletal systems in upper extremity and lower-extremity offer a new generation of assistive technology for both healthy and disabled people. First, they aid disabled persons in regaining lost function or improving their activity of daily living life and for assessing rehabilitation progress. Second, they can be introduced in setting the human machine interface using neuromuscular signals as command signals for the exoskeleton devices.

In chapter 2 it is described first the state of the art concerning myoelectric prostheses, which are used to restore the functionality of an amputated hand. Second, the state of the art of myoelectric-exoskeletons, which are used as human-machine interfaces. These devices can be able to recognise the desired movements of the operator and assist both healthy and disabled people. They are considered also as human movement amplifier. The study of the first part of movement realisation, using electromyograph (EMG) signals to recognise and classify different hand movements, needs first to proceed with signal acquisition. This task is of a great importance. The study of EMG signal acquisition is described in chapter 3. In this chapter, the forearm muscles activity can be read as electro-myographic (EMG) signals via surface-electrodes attached

to the forearm muscles: These EMG signals can be then analysed for further classification tasks. Developing Electro-myograph-hand prosthesis has a problem with the realisation of performed grasping capability, which is still a study subject. This problem of performed grasping manipulation with independent finger movements has been investigated in this first part. The popularity of designing and building hand prosthesis is achieved by a number of universities and research centres that have prosthesis hands named after them. In the past, the hand-prostheses were limited on motion in one degree of freedom and were basically only motorised hooks. Now many research laboratories try to perform voluntary closing and opening hand-prostheses based on EMG neuromuscular signals of forearm muscles. The control of hand-prostheses exist in two categories, the first is conventional body-powered prostheses, which are powered and controlled by gross body movements as mechanical commands, usually of the shoulder. The second, Myoelectrical prostheses, present the best considered way to restore the functionality of an amputated hand. These hand-prostheses, which belong to this second category are divided into two types. The first type exploit EMG signals that are not issued from muscles responsible for corresponding movement, but from any muscles usually biceps. Such type of prosthesis use EMG signals only as switch impulses. The second type of Myoelectrical hand-prostheses are able to recognise the desired movement from EMG neuromuscular signals issued from group of muscles responsible for corresponding movements. The first part of this thesis considers this second type of myoelectric hand-prostheses control. If a machine can understand human movement, it can be used in rehabilitation as a personal trainer that interprets a patient's EMG signals and help to provide a right movement. The most effective

rehabilitation methods employ EMG-machine assisted exercises, to improve the functional capacity and strengthen the affected muscles. In the same way it can also be used to control leg exoskeleton devices, that can be able to support the leg muscles during common movements like getting up from a chair, walking and climbing stairs. It's important to mention that the leg's neuromuscular EMG signals are more easily recognised than those of the forearm's neuromuscular EMG signals.

Extracted features in time-, frequency- and time-frequency-domain will be described and compared, see chapter 4, to find the relevant ones, which are able to discriminate these movement classes in clear separated clusters. Finding the best feature distribution, which has the best discrimination between classes is a crucial step before selecting the classification technique for the specific task of control.

This study goes forward and investigates the recognition capability, which is depending on the number of channels used for collecting EMG signals. This recognition capability increases with the number of measurement channels. In this thesis, it is shown that with only two channels it is possible to recognise and classify hand and also finger flexion movements, which are thumb-, pointer- and middle-finger. Generally, in other published studies, the number of measurement channels are at least four channels. Therefore, it is necessary to take a part of the study for the effect evaluation of the feature space-dimension. If this feature space-dimension increases, its influence on discrimination-accuracy will be positive, but this increasing of space-dimension has a drawback regarding time consuming. For this question we have two solutions: either to take a large feature space-dimension and then to apply space-reduction methods like Principle Component Analysis (PCA), or to consider a small space-dimension, see our publication [74]. This study will be detailed in chapter 5 to

compare between them and to give very important results.

The presentation of these EMG signals in different discriminated clusters corresponding to their movement classes is possible with help of extracted information from their various signal characteristics, particularly in time-frequency domain. This representation will be useful to map out and control hand prostheses or exoskeleton devices, which requires advanced computational methods for acquisition, analysis, decomposition, and classification. Intelligent computational algorithms, described and compared in chapter 6, are those based on neural networks like Multi-Layer Perceptron (MLP), Radial Basis Networks (RBF) and Learning Vector Quantization network (LVQ). The others are based on Fuzzy logic like Fuzzy Subtractive Clustering (FSC) and proposed Fuzzy Trimmed Mean Classification (FTMC) algorithm, see our publications [73] [75]. This proposed intelligent classifier approach, based on trimmed mean clustering and fuzzy logic, will be also compared with above cited intelligent computational methods to be evaluated.

In chapter 7 a general study of some important parameters, which have a great influence on classification performances is considered. These parameters should be optimised to perform the classification accuracy results. Some of these parameters are: pass-band frequency, filter-type, beginning part length of EMG signal, noise base-line reference, and frequency sampling. This study allows us to get a global view about how to choose the values of these parameters in order to get the best classification accuracy.

Models evaluation of musculoskeletal structures is considered in the second part of

this thesis. The FES procedure should stimulate the muscles (quadriceps as application) at the correct time during walking . To perform this task of synchronisation and to assess the stability of walking, a right musculo-skeletal-model is needed for rapid and dynamic adjustments to correct and control the motion of limb segments and consequently the body. A proposed performed musculo-skeletal identifier-model for quadricep muscles properties based on proposed fuzzy-modeling Algorithm see our publications [72] and [16], is described in chapter 8. This model uses Functional Electrical Stimulation (FES) as input set and knee-joint angle as output set. The muscles of legs and arms are enough big, which allows the use of this method for stimulating exactly the desired muscle. Functional electrical stimulation (FES) impulses have been used to activate muscles disabled by spinal cord injuries. Stimulators worn on the leg, which stimulate muscles through electrodes can help to restore or perform muscle activation for walking.

The last chapter, will concern the recapitulation of all this thesis and moreover gives the real applications of this study and the attempted future work.

Chapter 2

State of the art of myoelectrical hand prostheses and exoskeleton devices

2.1 Introduction

The research in "myo-prostheses" is part of a larger trend in human-machine interaction that aims to integrate body, muscle and machine. First goal of this research is focused on the classification (recognition) of myoelectric signals for further control of hand-prostheses. It is expected to advance in this chapter the state of the art about today's most common commercially available myoelectrical-prostheses. Since 1970 a big progression is done in this field. In the beginning they were limited on motion in one degree of freedom and are basically motorized hooks. Now by use of a suitable combination of electronic hardware and software, it is possible to recognise the myoelectric features of at least two different grips in real time, with an accuracy of almost 95 percent. Further researches might eventually include comfortable hand-prostheses that could act, in a nearly lifelike manner and in real time of as many as six different grips. On other side The designs of majority of commercially available electrical hand-prostheses do not provide independent control of fingers and thumbs

but most of them are capable of only simple one degree of freedom grips. Experts in prostheses generally agree that the electrical hand prostheses should respond to the following criteria which are vital to a person's daily activities:

- relatively comparable to the weight of human hand (low weight).
- relatively comparable also to the size of human hand (low size)
- appearance satisfaction for the patient (cosmetic)
- don't be a source of noisy sounds (noiselessness)
- have sufficient autonomy in energy
- have a time of reaction as short as possible, ideally real time.
- reliable

The second application of EMG signals recognition is rehabilitation issue, see section 2.4. There is a growing need for physical rehabilitation and assistance to improve the quality of life for physically disabled peoples. Several exoskeleton devices [51] [65] [18] in robotic systems for rehabilitation have been constructed to perform rehabilitation support systems and many works in this topic are done during the two last years. Some of them [18] exploit EMG signals that are not issued from muscles responsible on corresponding movement, but from biceps. These authors focused their attention on a basic pinching motion between the index finger and the thumb. The amplified, filtered and normalised EMG signal measured from biceps can be used to control the exoskeleton devices. Other works [48] [7] consider the recognition of desired

movement using EMG signals issued from some arm-muscles responsible on corresponding movements. In these cases EMG signals, are simply rectified and filtered then compared to a defined threshold to control the actuators of the exoskeleton.

2.2 Commercially available hand prostheses

2.2.1 The hand *WIME* (Japan)

As an application for the study of artificial hands and the study of upper-limb prostheses; the WH (Waseda Hand) series was started in 1964. The development of the mechanism of hands and their control methods using electromyogram was the main point. Their achievements resulted in the *WIME* hand (Waseda Imasen Myo Electric hand), which has been commercially available since 1978, figure 2.1. The *WIME* HAND is a practical EMG-controlled forearm prosthesis and manufactured by the *Imasen* Engineering Corp. Sufficient field tests were carried out during the developmental period with the cooperation of 30 amputees over a period of three years. The mechanical flip-flop allowed gripping by the fingers and pinching by three fingers using only one motor, which was voluntarily controlled by EMG signals obtained from the arm of the amputee. The pressure sensor attached to the fingers sensed the reaction force of objects, which was then fed back to amputees by electrocutaneous stimulation. The *WIME* HAND has been commercially available since 1978, [6]

2.2.2 The sensor-hand with *SUVA* Technology, Germany

Years ago Winkler and Bierwirth in the Swiss Bellikon at *SUVA* had the idea of feeding measured Ťimpressions of sensesŤ back into the hand-prostheses. For the

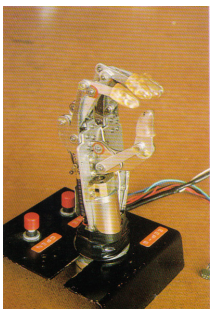

Figure 2.1: Assemly of the *WIME* HAND

first time this hand-prosthesis was equipped with an automatic grip control system, figure 2.2. The *SUVA* sensor is steadily metering the direction and the size of the force, which exist between the thumb and the gripped object. It sends no signal if, in the case of small objects, the tip of the thumb is not touched. In order to recognize a grip nevertheless a second sensor, the finger lever sensor was built into the sensor hand. With this one also these grips are recognizable and governable. The measuring data of the *SUVA* sensor will be read and the amount as well the angle of force is calculated from the measured force components. As long as this measured angle remains under a critical value the object does not slip from the hand, [56]. These devices allow to carry out two types of control:

- 1) Automatic mode of control:

  In this mode the hand is closed with a maximum speed and seizes an object with the weakest grip force (10 N). If the sensor detects a change of position, it automatically makes increase the grip force to its maximum (100 N) to avoid the fall of objects.

- 2) Variable mode of control:

  The speed of opening is determined by the power and the speed of the muscular

signal. The speed of closing is a function of the reduction of the muscular activation, [3].

Figure 2.2: Assemly of the *SUVA* Hand

2.2.3 *RSL* Steeper Prostheses UK

RSL Steeper is the principal service of rehabilitation of the United Kingdom. This company in particular designed the prostheses of 13-year-old Ali Abbas, figure 2.3, who lost both arms in the Iraq war. Doctors plan to fit the teenager with two artificial arms, which will be strapped together and worn somewhat like a rucksack. On his right-hand side, Ali will be fitted with a "myoelectric" control system, a state-of-the-art technology which uses electrodes to pick up nerve signals from existing muscles in the stump. Because Ali's left arm was amputated higher up, at the shoulder, its replacement may not offer the same functionality since there is less muscle to work with. So, a tensing of the upper arm muscle would cause a motorised hand to grip, while relaxing it would release the pincer movement. The greater the tension, the quicker the motor works. His right hand could be wired to his bicep and his motorised wrist to his triceps. For the elbow, a better option might be to use a simple pulley system which Ali would operate simply by shrugging. The electrics depend on a

lithium ion battery, which would be worn in a pouch on the upper body and replaced daily. Mr Cooper, who works for the prosthetics makers *RSL* Steeper says: Cutting with a knife will be difficult but he'll be able to use a fork or spoon almost naturally, comb his hair, type with two fingers, [4].

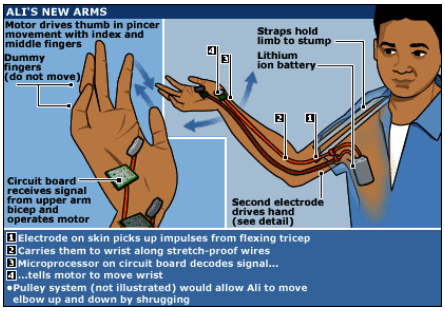

Figure 2.3: *RSL* Steeper Prostheses UK

2.2.4 Conclusion

The development and the improvement of three examples of commercially available electrical hand prostheses in three different countries, like Germany, *UK* and *Japan* are described. The study of these commercially artificial hands began with passive prostheses, but it was possible to develop the active prostheses . The studies had at first aimed only to develop machines to perform motion in one degree of freedom and were basically motorised hooks. Recently the aim has been to develop prostheses that can perform, using EMG signals, more complicated tasks.

2.3 Intelligent hand prostheses in the laboratories

2.3.1 The hand *MARCUS*

Hand *MARCUS* was initially designed like an evolution of *Otto-Bock* hand-prothesis. It consists of three fingers: thumb, index and major, figure 2.4. It has two degrees of freedom and is driven by two separated motors, the first one is responsible for the movements of the inch and second one acting in the movements of the index and the major, which are dependent. The hand moreover is equipped with *Hall-effect-sensors* to obtain information on the position of the fingers and a tactile sensor on the thumb giving information on the force of grasping, [28] [71] [41].

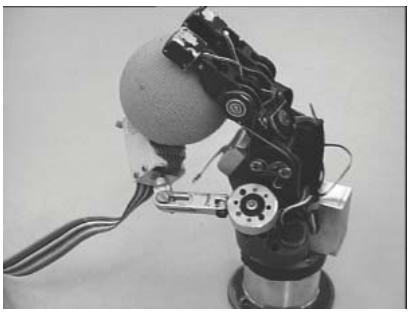

Figure 2.4: *Marcus* Hand

2.3.2 The hand of *KFZ*, Germany

The hand of Karlsruhe is a very light hand, each finger weighs only (20g) which approximates very well the aptitudes of handling of the human hand, by making it possible to move independently all the fingers. It uses for that an original approach; in the place of motors with D.C. current, this hand has 18 fluidic, flexible and miniaturised actuators which order 5 fingers. Each finger contains the actuators responsible for its inflection, and tactile sensors. The metacarpus provides enough space to place there a micro-controller, micro-valves, the source of energy and a micro-pump. An

optional actuators as designed to fold the wrist, figure 2.5.

The flexible fingers are able to grasp objects of various sizes and forms, by distributing the force of contact on a larger surface. Thus, thanks to this self-adaptation, a large variety of objects can be seized without using sensory information. Moreover, the surface of the fingers is soft and the coefficient of friction is increased using a rubbery glove which covers the artificial hand. Because of flexibility of the hand, this one appears more natural than a rigid robot-like hand and the risk of the direct interactions with other people is minimised, [57].

Figure 2.5: The hand of *KFZ*, Germany

2.3.3 *RTR-2*, Rehabilitation Technology Research, Italian

Hand *RTR2* is consisted of thumb and two identical fingers, the major and the index. It has nine degrees of freedom, figure 2.6. RTR2 contains only two motors, for the movements of inflection and extension of the fingers and the thumb, and for the adduction and the abduction of the thumb. The movements are based on a system of transmission per tendon. To improve the grasping operation, the information provided by an artificial sensory system is considered, which react automatically in

Figure 2.6: *RTR2* Hand

case of objects slipping from the hand without any reaction from the user. The hand *RTR2* is equipped with position sensors and a tensiometer on the cable which control the pointer and force sensor on the end of the thumb. All sensors and the two motors are inside the structure of the hand, but its weight remains very light since it is lower than 320g. From other side the grip strength remains insufficient: it is of 16 N whereas that of the commercial prostheses reaches 100 N. The controlling of the prosthesis takes place via a Top level Controlling module (TCM) and a Low level Controlling module (LCM). The TCM uses the myoelectric signals (EMG) to produce a control for the LCM, which regulates the motors after collection of the sensory signals. This hand is a scheme for future Myoelectric hand prosthetic *RTR4* and *Cyber-hand* Prothetic, [14] [12].

2.3.4 Hand-prosthesis of *Hokkaido* university (Japan)

This hand is developed by the Autonomous Systems Engineering Lab of the *Hokkaido* University (Japan), figure 2.7, within the framework of a project to design a prosthesis of hand having the behavior of a natural hand and ordered by EMG signals. It uses an adjustable transmission system in which the course of the cable depends on the size of the load. The fingers move quickly under a light load and slowly with a high couple

of torque under a heavy load. A method of order using tendons was selected to locate the actuators outside the driven elements. The hand has 7 degrees of freedom, for each finger plus the abduction/adduction of the thumb and the pronation/supination of the wrist. It's built of aluminium and each finger weighs 25 G. All the actuators are external of the hand, the total size of the hand is big, which makes its use impossible as a prosthesis, [22].

Figure 2.7: pictures of the prosthetic hand developed at the Autonomous Systems Engineering Lab of the *Hokkaido* University.

2.3.5 Die Hand von *Southampton*, (*UK*)

The *Southampton* philosophy concentrates on devolving the responsibility of grip adjustment from the user to the hand itself. The intelligent hand uses sensors, electronics and microprocessor technology to allow this adaptive device to maintain optimum grip (thereby ensuring that objects do not slip from the hand) under the jurisdiction of a state driven control system (which allows easy control of the prosthesis). The artificial hand of *Southampton*, is in development since several decades. It has been elaborated with the idea to be controlled in a hierarchical way using EMG signals. To grasp objects with a natural hand, the brain must have a multitude of information so as to adjust the grasping operation and to prevent the slipping of the object.

However, with many myo-electric artificial hands, to control the force to be exerted by the hand, the user is asked to use only the visual signal as feedback signal while acting on EMG signals of forearm. In order to cure this insufficiency, the hand of *Southampton* has an intelligent device: the responsibility for the adjustment of the grasping is confided to the hand itself and not to the user. This device uses sensors, electronics and microprocessors to maintain an optimal force of grasping, while the user gives overall orders to open or close using simple signals.

The hand of *Southampton* provides two types of grasp: grasp with precision and grasp with force. The type of grasping adopted is determined by the point of the first contact. If an object touches the palm in first, the grasp with force is applied; if they are the ends of the fingers which enter the first in contact, a grasp with precision is used. The hand is closed until the object is taken in the softest possible way figure 2.8 and 2.9. If a slip occurs, it is detected by acoustic sensors on the level of the ends of the fingers and the grasp is automatically reinforced, [5].

Figure 2.8: First version of the hand of *Southampton*.

Figure 2.9: Second version of the hand of *Southampton*.

2.3.6 Conclusion

There are many articles in the news about the latest developments in intelligent prostheses devices, which are high-tech but expensive prostheses. The use of a microprocessor in the system allows the operator to supervise only the actions of the hand while the microprocessor controls the low level reflexes of grip force and shape. The processor can control more of the functions of the hand itself. The operator gives simple grasping movement (hand closing or opening), which are interpreted as commands and the controller co-ordinates multiple degrees of freedom to shape the hand to maximise the contact area between the hand and the object and so minimise contact forces. If the object slips this is detected and the controller responds. This frees the user to make only the strategic decisions while the functional range is increased. We can mention here that these developed prostheses don't recognise the desired motion of fingers separately, which is our first goal in this research thesis.

2.4 EMG controlled Hand-Exoskeleton devices

The exoskeleton devices are used as an assisting systems for affected people by stroke or other motor diseases or spinal cord injure. Hand-exoskeleton devices are an innovative ideas to reduce physiotherapist intervention and to improve therapy results. In this case the EMG signals are used as self body's neural signals to realise intended hand movements. Several exoskeleton devices in robotic systems for rehabilitation have been constructed to perform rehabilitation support system. Many works in this topic are done during the two last years and some of them will be introduced in the following sections.

2.4.1 *Carnegie Mellon* Exoskeleton (Pittsburgh USA)

The mechanical framework of the exoskeleton consisted of an aluminum anchoring plate mounted to the back of the hand and three aluminum bands, one for each of the finger bones. The aluminum bands are designed to be adjustable for different finger sizes. The flexion of the Proximal Interphalangeal (*PIP*) and Distal Interphalangeal (*DIP*) joints is produced by steel cable running along the front of each finger band and through to the backside of the hand. These cables are pulled by a pneumatic cylinder

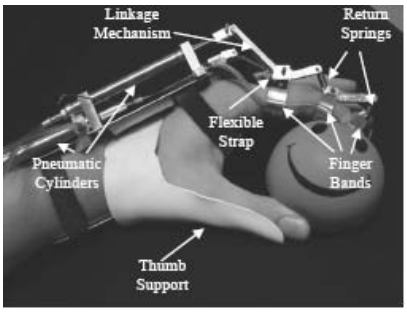

Figure 2.10: *Carnegie Mellon* Exoskeleton, Lab of the Pittsburgh University USA.

acting in compression. The metacarpophalangeal (*MCP*) flexion, on the other hand, is achieved by a linkage mechanism: a floating link is mounted between the finger band closest to the base plate and a second pneumatic actuator, acting in extension (labeled as linkage mechanism), figure 2.10, [18].

2.4.2 Politecnico di Milano Exoskeleton

This exoskeleton-hand, figure 2.11, is composed of a glove, upon which a supporting structure is built, implemented in plastic. The plastic part on the glove is used for two reasons: guiding the fingers of the patient in order to accomplish a natural movement and avoiding that the fingers had to bear an excessive load on their tips. In addition to

Figure 2.11: hand Exoskeleton, Lab of Politecnico di, University Milano Italy.

this, two plastic bended covers are placed upon and under the forearm of the patient and bound together by straps. In order to improve the system stability the upper cover on the forearm is fastened with the plastic structure on the glove by means of a metallic bar. On the upper cover (in the palmar side) two actuators are fastened, that are *Hitec* servos *HS-805BB*. These electric motors can be controlled in position. Two wires are joined to the fingers tips at one end, and rolled up to the pulleys of the servos to the other end. The wires slide through some little plastic pipes and can transmit the maximum force produced by the actuators, about 100 N. One wire is dedicated to the flexion of the thumb, while the other flexes the four fingers at the same time. On the dorsal side, two springs are required to allow the extension movements. In this way, with only two actuated degrees of freedom, the device is able to perform a grasp movement. Finally two potentiometers on the pulleys of the servos are placed in order to record two position signals [48].

2.4.3 Conclusion

Two exoskeleton devices systems for rehabilitation have been described in this section to show the importance of using EMG signals In order to help patients, who had a function disability of their hands, to get normal daily life. Such Exoskeleton devices

controlled by EMG signals can provide a self-performing rehabilitation system that supports the patients to practice the rehabilitation exercise by them self.

2.5 Hand Prostheses Types

The control of hand prosthesis exist in two categories, the first one are conventional body-powered prostheses, which are powered and controlled by gross body movements, usually of the shoulder. The second one are Myoelectrically controlled prostheses, which are considered, at present, as the best way to restore the functionality of an amputated hand. These hand prostheses, which belong to this second category are them self divided also in two types. The first type are those exploit EMG signals that are not issued from muscles responsible on corresponding movement, but from any other muscles usually biceps. Such type of prostheses use EMG signals only as switch impulses. The second type of Myoelectrically hand prostheses are able to recognise the desired movement, on the way as the subject thinks about moving the prosthesis. This case is considered in this thesis.

2.5.1 EMG signals don't correspond to movement's muscles

There is the possibility to use EMG signals issued from any part of our body, for example biceps, as switch (ON-OFF) signal to control hand Prostheses. This type of control is easier, because it doesn't need any data processing but only signal acquisition, rectification and then integration. the rectified signal can be compared to a threshold reference, if it is bigger then an ON command is given otherwise the hand prostheses is not acted.

2.5.2 EMG signals correspond to movement's muscles

The idea in this case is the use of surface electrodes to record the electrical activity of muscle fibres and to send these signals to a computer, which interprets them using convenable algorithms. The algorithms translate the signals into commands that can control a hand prosthesis. The essential task is based on signal acquisition and data processing. Intended hand movement can be interpreted through the EMG signals issued from responsible muscles for such movement. These signals are detected in the region of activated muscles for intended movements. This task, which is considered in this thesis, requires bio-signal processing algorithms for different processing and discrimination stages, which can be resumed in the following steps:

Steps:

- Data Collection through sensing devices
- EMG signal transformation: analysis or Modeling
- Extraction of relevant features, which can discriminate movements
- Classification-models building
- Evaluation and classification

Chapter 3

EMG signal acquisition

3.1 Introduction

The physical phenomena observed generate often analogical signals. The power and the diversity of the realisable transformations by the computers makes desirable the conversion, figure 3.1, of these analogical signals into discrete signals, obtained by measurements with intervals of regular times. This operation of sampling, realised by

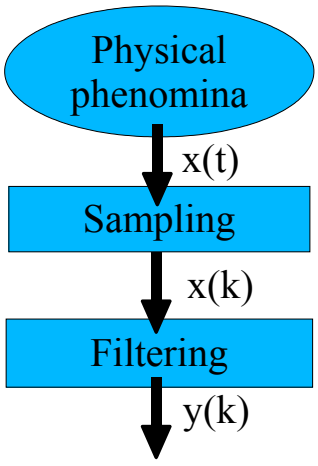

Figure 3.1: The first stage of signal acquisition

the analog-to-digital converters (ADC), can involve a loss of information. It is thus advisable to carry out this operation correctly, in order not to make imperceptible the

information sought in the signal. The exponential increase in computational abilities of computers and the advancement of sensor technology are all factors contributing to the expansion of EMG research with fidelity. EMG signal energy increases with the activation level of a muscle. We observe the variation also in EMG measurements from experiment to experiment so the energy of an EMG signal is a largely qualitative measure. EMG signals can be measured using two electrode types:

- indwelling (fine wire or needle) electrodes, which are inserted directly into the muscle fibres.
- surface electrodes, which are placed on the skin overlying the muscle.

The surface electrodes transduce the motor action potential MAP, converting ionic currents into electrical currents, and the resultant EMG signal can be recorded following appropriate amplification and filtering. However some disturbances may be introduced in measured EMG signal through many ways. First the human body himself is a good antenna, which picks up electrical signal emissions, issued from electrical equipments in the laboratory. Second the cables of measurements are good conductors for power line noise, 50-60 Hz signals. Additionally it's not possible to ignore the effect of artifacts, which are results of cable and electrode movements.
The use of available highly sophisticated devices with help of the advances made in electronics technology had made the acquisition of EMG signal possible with high fidelity and more efficiency. Generally signals are the means (ways) by which information is transmitted, whether we use the vibrations of accelerometers, the electricity of circuitry or electromyograph (EMG) signals of electrods. There are certain fundamental common aspects of signals that are universal. Signal acquisition and processing allow us to understand the systems, which produce these signals. Therefore, the

principles of this operation are extremely important for many aspects of electrical engineering, and can easily be extended to any other study domain.

The size of the Electromyograph (EMG) signal depends on: 1) the thickness of the connective muscle tissue, 2) the quality of the contact between the electrode and the skin, 3) the size of the electrodes and 4) the individual motor unit action potentials. After signal acquisition and processing, it's possible to identify and control the system, which is source of this signal. But the important question explored in digital acquisition is how to sample an analog signal while preserving its full information. The sampling rate for the analog to digital conversion (A/D) must be at least twice as high as the highest frequency or bandwidth of the signal being sampled, according to *Nyquist-Shannon* sampling theorem, [54]. The knowledge of frequency bandwidth, which envelope the most power of EMG signal is necessary to choose the appropriate sampling frequency. With low sampling frequency it's not possible to track with fidelity most rapid changes in the signal, however the high sampling frequency increases the number of samples, which leads to a time consuming. In case of on-line prostheses control, the procedure of signal acquisition, processing till decision control should be short in time. Law sampling rate means time computation consuming, for which the time-delay-phase between human intention of acting and prosthesis response is not acceptable. So it should be found a compromise between them. To avoid another undesirable effect of sampling, it's well to employ an anti-aliasing filter before the signal is sampled, which requires also a knowledge of signal frequency bandwidth of interest in order to perform this task. By definition, the anti-aliasing filter [54] is used to prevent the sampling of frequencies, in the signal, that are higher than the half of sampling frequency. These frequencies will be misrepresented if they are sampled. For

example, a 1 kHZ sampling frequency needs an anti-aliasing filter with a bandwidth of $500Hz$ ($\frac{1}{2}Fs$), the effect of this filter is to avoid the aliasing to a lower frequency for the signals above *500Hz* due to under-sampling of these signals. As consequence to this phenomenon there is the production of a new aliased frequency Fa, which is a mirror of the original signal frequency F about $\frac{1}{2}Fs$.

3.2 Surface EMG signal characteristics

The measurements of EMG signals issued from muscle contractions are realisations of a complex time-variant process that control electrical activation of muscle. They provide an access to physiological processes that cause muscles to generate forces, produce movements, and accomplish functions which allow us to interact with the world around us. It's difficult to discern any distinguishing characteristics of these signals and it's not apparent how to quantify them. With help of two non-invasive EMG surface electrodes placed on forearm muscles it's possible to detect EMG signals, which will be subtracted before amplification. In this differential configuration, the shape and area of surface electrodes and the distance between them are important factors, which affect the characteristics of this measured EMG signal. This signal, issue from a time-variant complex dynamical system figure 3.2, has an amplitude in the range of μV_s or mV_s, it depends on type and/or size of the muscle and its state (level of activation). The usable spectre of this stochastic (random) signal can be limited in the area of 20 *to* $500Hz$, see for deep study about this factor the sections 4.3.1 and 7.2.

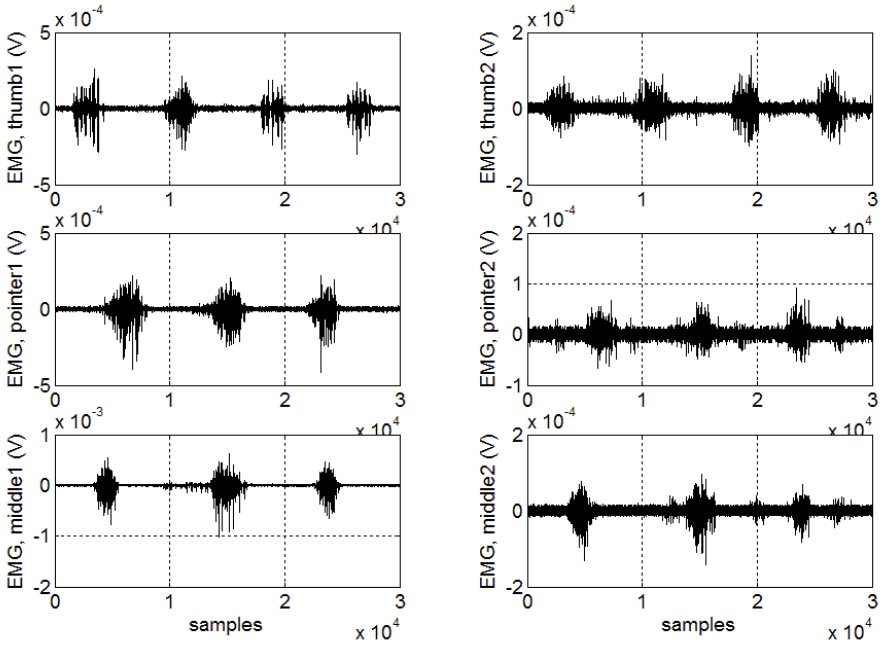

Figure 3.2: Two channels measure of raw EMG signal corresponding to thumb-, pointer- and middle-finger flexion movement. Sampling frequency equal to *4KHz*

3.3 Factors affecting EMG signal measurement

3.3.1 Electrodes

The form and the size of the surface electrodes have an influence on measured EMG signal. For a performed extraction of quantitative information from the EMG signal it is required greater focus on the configuration of the electrodes. The major points to consider are:

- Electrodes material:

  Two types of surface electrodes are known, 1) dry electrodes in direct contact with the skin and 2) gelled electrodes, which contain an electrolytic gel between the skin and the metallic part of the electrode.

- Electrodes technology:

  It's recommended that the electrodes make very good electrical contact with the skin through electrically conductive gels. Therefor, the electrode gels material should be highly conductive and the electrodes adhesive material should have strong adhesive properties to the skin for considerable mechanical stability to avoid movement artifacts.

- Electrodes position:

  The placement of electrode should be along the longitudinal midline of the muscle. The longitudinal axis should be aligned parallel to the length of the muscle fibers.

- Reference electrode placement:

  it's a common reference electrode to the differential amplifier input. The placement of this electrode should be on electrically neutral tissue (the bone).

3.3.2 Amplifier technology

In amplification process of small bioelectric signals generated by the muscles, which are typically in uV, it's necessary to reduce as possible the effect of noisy electrical signals. This task is accomplished through the use of a differential amplifier, figure 3.3, which effectively cancels the ambient electrical noises collected by human body. These ambient electrical noises collected by human body can reach the order of volts. This subtraction operation of differential amplifier eliminates these noises and amplify the small physiological signals. There are several important properties to consider in this differential amplifier:

- High common mode rejection ratio ($CMRR$)
- Very high input impedance: amplifier's input impedance must be higher than the impedance of the electrode-muscle area.
- Using short connecting cables between signal source (electrodes) and Amplifier, otherwise they will collect ambient noises.
- DC signal: they are generated by chemical reaction between skin and electrode.

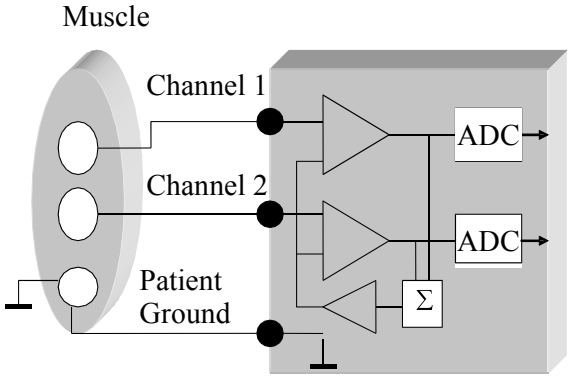

Figure 3.3: Differential Amplifier

3.3.3 Movement artifacts

The motion of the electrode relative to the skin produces motion artifact, which occurs in the range of *0 to 15Hz*. For these reasons, the surface EMG is typically filtered under the range of 20 Hz to eliminate low frequency noise and increase the signal to noise ratio

3.3.4 Disturbances

The acquisition of an accurate EMG signal is very dependent on the noisy electrical environment and the performance of acquisition instruments, [20] [58] [45]. Noises can be described as any aspect of the output signal which is undesirable. Elimination of influences of these noises is not possible, but it's possible to reduce them with examination of circumstance's effects. They are two typical types of disturbances. First are conducted disturbances, which can be caused by electrical disturbances intrinsic to the recording environment and it can be caused also by the nature of the recording devices themselves. Second are radiated disturbances, which can be caused by electromagnetic emissions of environment.

- 1) Conducted Disturbances:

  -Transducer noises (through the cables)

  -Alternative current, generated by fluctuations in impedance between the conductive transducer and the skin (*Duchene* and *Goubel*, 1993).

  -Direct current, caused by differences in the impedance between the skin and the electrode sensor, and from oxidative and reductive chemical reactions taking place in the contact region between the electrode and the conductive gel (*Gerdle et al., 1999*).

- 2) Ambient Disturbances:

  Disturbances from the environment (electromagnetic waves), are noises originate from sources of electromagnetic radiation, such as radio and television transmission, electrical-power wires, light bulbs, fluorescent lamps, ... [45].

3.3.5 Cross talk

Recorded EMG signals are dominated by muscles activities signals, which are close to the electrode. This recorded EMG signal from desired muscle could be mixed "*crosstalk*" with other EMG activities issued from one or more neighboring muscles. Other effect of muscles on recorded EMG signals is the distance (width) between the muscle fibres and electrodes, which increases the spatial low filtering effect. This phenomenon is due to the fact that muscle fibres act as low spatial filter [43].
The above factors illustrate clearly that it is not easy to measure an electro-physiological signals without disturbances. Therefor it is very important to select the right measurement system and the right sensors to maintain optimal electro-physiological data.

3.4 Experimental recording equipment (*Digitimer Neurolog System*)

EMG signals are low in amplitude with respect to other ambient signals on the body surface, hence it is necessary to detect the signal in a differential amplifier configuration in order to reduce noises. The bipolar recording technique is based on bipolar electrode arrangements with a differential amplifier, figure 3.3, which suppresses signals that are common to both electrodes. Correlated signals common to both sites, power sources and electromagnetic devices, are suppressed. The placement of electrodes is required to be on the large face of muscle. The amplification of the two differential inputs should not deviate from each other more than 1/100000, which requires highest common mode rejection (*CMR*) possible. Common mode rejection by around *100dB* is generally sufficient to eliminate such common mode disturbances.

The CMR is expressed in logarithmic form in equation 3.4.1.

$$CMR(dB) = 20 \log \frac{A_0}{A_1} \tag{3.4.1}$$

The balance in the input impedance between electrodes may be an addition source of a substantial effect on generation of noise. such problems can be avoided if short connection cables between electrodes and amplifier are used. The best solution for this problem is to use the new developed electrodes with incorporated amplifiers.

Overview:

In this study the equipment of measurement used is **Digitimer NeuroLog System**, figure 3.4, figure 3.5 and figure 3.6

Modules of this equipement:

- it has four channels.
- amplification's range: *×10 bis ×10000.*
- low band filter (*3, 10 and 30Hz*).
- push button for movement artifact elimination for (*NL824*) module.
- possibility of signal amplification through one or many channels. (4 channel filters).

Notch Filter: this filter is used to remove a particular frequency from a signal and has a frequency response that falls to zero over a narrow range of frequencies (i.e. a *50Hz* notch may block signals from *49.5 to 50.5Hz*). Notch filter is also available

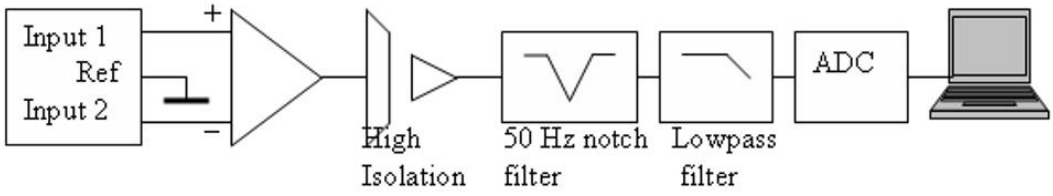

Figure 3.4: *Digitimer Neurolog System.* EMG-Measure-system (Bloc diagramm)

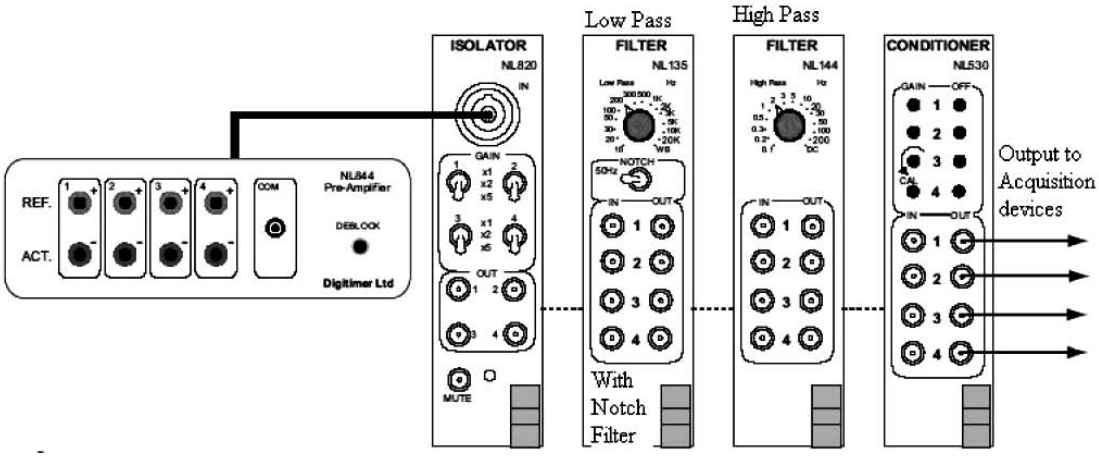

Figure 3.5: Four-Channel Isolated Amplifier System with Filtering and Signal Conditioning

in this instrument. The *NeuroLog* System is a flexible and upgradeable multi-channel recording device for research applications such as electromyography (EMG). The *NL820* is the module at the heart of the isolated amplifier range of components. It is ideal for AC coupled recording applications in the research environment. It has four channel units with independent gain and filtering control of each channel as well as a mute facility. The *NL135* FILTER is a 4 *channels*, second order low-pass, with Notch reject filter module. The filter settings can be selected. Therefor a rotary switch selects the 14 frequency settings giving repeatability over a wide range with *40dB/decade* attenuation above the selected frequency value. The active Notch filter

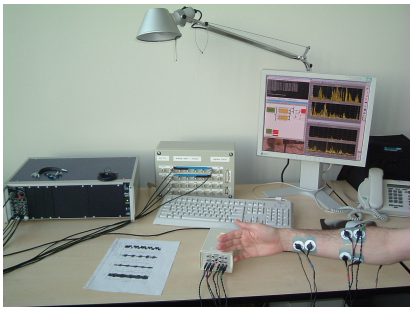

Figure 3.6: Measurements system in Max Plank institute laboratory, Magdeburg, Germany

provides rejection of line frequency *50Hz* interference when switched on. The *NL144* is a 4 *channels*, second order high pass filter which complements the *NL135* when used with the *NL820* isolator, this module provides a compact solution to 4 channel high and low pass filtering. This module designed to give gain and offset set-up controls when interfacing signals to the analog-to-digital converters (*ADCs*) of *PCs*. It contains four channels each with independently adjustable filter settings and front panel gain and offset presets. There is also a master *ADC* offset control to allow unipolar *ADCs* to be used with bipolar signals.

3.5 Functional anatomy of hand and forearm

The Engineers entrusted with the management of the hand-prostheses and hand-exoskeleton must possess a competent knowledge of the functional anatomy and physiology of the hand and forearm, which is a complex biological structure. Necessary also is the ability to correlate the surface topography of muscles, underlying muscle-tendon units, skeleton, joints, and nerves. So here are some information to help to understand the mechanical and anatomical properties of the hand and forearm muscles. The Bones of the Forearm and Hand [1] are presented in figure 3.7. The forearm

contains many muscles [35], a flexor of the elbow (*brachioradialis*), and *pronators* and *supinators* that turn the hand to face upwards or down. In cross-section the forearm can be divided into two fascial compartments. The posterior compartment contains the extensors of the hand, which are supplied by the radial nerve. The anterior compartment contains the flexors, and is mainly supplied by the median nerve.

1. Anterior Compartment
    - Superficial Group
        - Flexors of the hand and wrist
            * Flexor Carpi Radialis
            * Palmaris Longus
            * Flexor Carpi Ulnaris
            * Flexor digitorum superficialis (*sublimis*)
2. Posterior Compartment
    - Extensors of the hand and wrist
        - Extensor Carpi Radialis Longus
        - Extensor Carpi Radialis Brevis
        - Extensor Digitorum (*Communis*)
        - Extensor Digiti Minimi (*Proprius*)
        - Extensor Carpi Ulnaris
        - Abductor Pollicis Longus
        - Extensor Pollicis Brevis
        - Extensor Pollicis Longus

  - Extensor Indicis (*Proprius*)

- Intrinsic forearm muscles
  - Brachioradialis (technically a flexor of the forearm)
  - Supinator
  - Anconeus

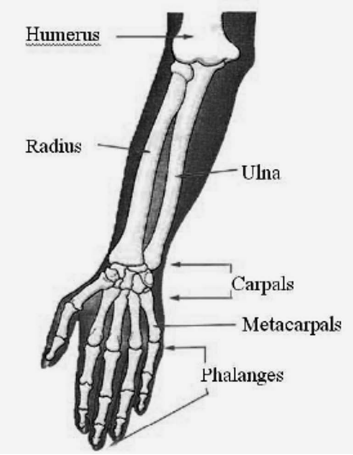

Figure 3.7: The Bones of the Forearm and Hand

In [35] the globally corresponding muscles participating in finger and hand movements are described. The following muscles, *Flexor-digitorum-profundus*, *Flexor-digitorum-superficialis* and *Flexor-polcis*-longus participate in fingers flexion. Fingers extension need globally the activation of *Extensor-digitorum*, *Extensor-indicis* and *Extensor-digiti-minimi*, and thumb extension needs *Extensor-pollicis-longus* and *Extensor-pollicis-brevis*.

Surface electrodes are placed in the manner to cover the large skin surface of these muscles. The locations of electrodes on the subject's arm do not isolate a specified single muscle but collect the EMG activation from all muscles around, even the

muscles of the deep layer, which contribute to this activation signal, although they undergo a space filtering. Some of these important muscles, which contribute in hand and finger motions are presented in figure 3.8 and figure 3.9.

Figure 3.8: Some Forearm muscles for extension movements

Figure 3.9: Some Forearm muscles for flexion movements

3.6 Conclusion

Surface electrodes placed over the muscle can sense electrical potentials produced when a muscle is contracted. This signal detected by the electrodes is called Electro-Myo-Graphic signal (EMG). This signal is recorded and amplified using convenable

instrumentations. It should be recorded with a certain fidelity to assure the transmission of its inside information (without loss of information). This recording fidelity require some fundamental concepts in EMG signal acquisition. Moreover competent knowledge about the functional anatomy of hand and forearm aids to get fidel EMG signals.

Chapter 4

Signal processing and feature extraction

4.1 Introduction

The choice of classification Algorithms begin with finding the features data, which can be available in several forms. These features data must be collected and the question is which features data are needed and can be extracted according to systems and classification problems. Most classification algorithms are highly sensitive to the quality of the data representation. In this chapter we will discuss EMG signal analysis methods, which give relevant features. These features are used as clusters for classes recognition.

There are many sources of high- and low-frequency contamination on EMG signals. For example computers introduce high-frequency noises into acquired signals, especially when the acquisition card is located within the computer chassis. Moreover motion artifacts introduce also low-frequency noises. Generally a 10 till 15 Hz high-pass filter is used to eliminate the movement artifacts and 300 till 500 Hz low-pass filter is used to eliminate the high frequencies. There are recommendations to cut frequencies below 50 Hz for large muscles like for the leg. However there are another muscles

in which low frequencies can have relevant and useful information about correspondent movement type. Thus the intended use of the EMG signals must be considered. More deep study about filtering effect will be given in this chapter, section 4.3.2 and in chapter 7 section 7.2.

4.2 Detection of activation period

Threshold method which compares the level of EMG signal with a given level, is the most intuitive and common computer-based method to detect ON-OFF timing of the muscle activation [47]. The EMG signal is processed in the time-domain. Two first transformations are commonly used as primary tools to analyse the acquired EMG signal, which are the rectification and M-point moving average Filtering, figure 4.1. These primary tools are appropriate and provide useful measurements of the signal amplitude to detect the muscles activation times, which are start and stop phases. The moving average filter is the most common filter for time-domain processing signal,

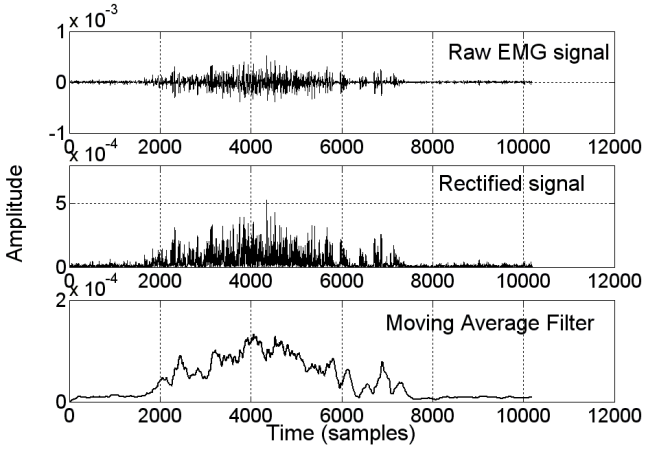

Figure 4.1: Full wave rectified and filtering of raw EMG signal (Hand closing) using moving average filter (window = 50ms)

figure 4.2. It operates by averaging a number of points from the input signal to one point in the output signal, defined by the equation 4.2.1.

$$y(t) = \frac{1}{M} \sum_{k=0}^{M-1} x(t-k) \tag{4.2.1}$$

where

x: is the input signal,

y: is the output filtered signal,

M: the number of points in the average, Fc (cut off frequency) = Fs / M;

Fs (sampling frequency) = 4 kHz;

M (length of Window) = 200 samples (50ms).

Fc = 20 Hz.

In a comparison form, figure 4.3 shows from top to bottom: raw EMG signal, rectified

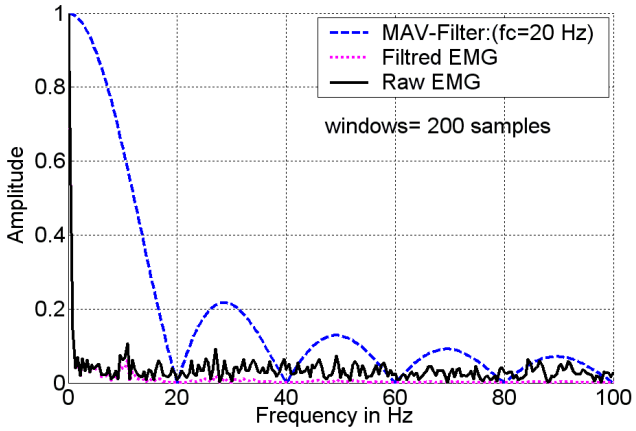

Figure 4.2: frequency response of the moving average filter with window of 200 samples.

EMG signal and then filtered signal. The signal in blod line in bottom represents Average Moving Filter response using 200 averaged samples or a window of 50ms (sampling frequency = *4 kHz*).

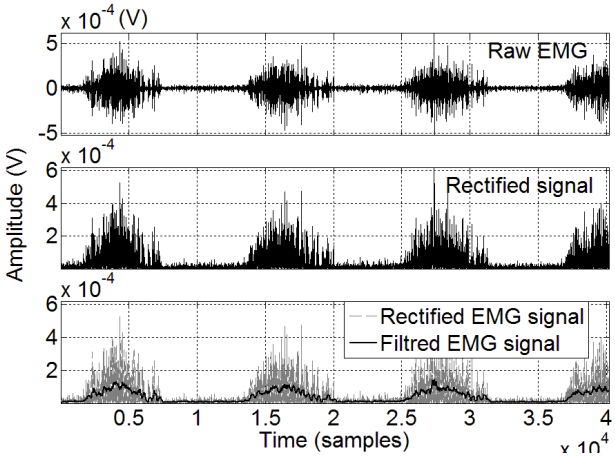

Figure 4.3: From top to bottom:Raw EMG signal, full rectification EMG signal and EMG signal after moving average filter application.

It is possible to detect the beginning of EMG activation after the estimation of the noise amplitude that will be considered as threshold between activation-phase and noise of the signal EMG. Hence Some knowledge about the noise signal is required before the estimation of its level. This threshold level can be defined as a certain amplitude above noise mean value, figure 4.4. The threshold level value can be

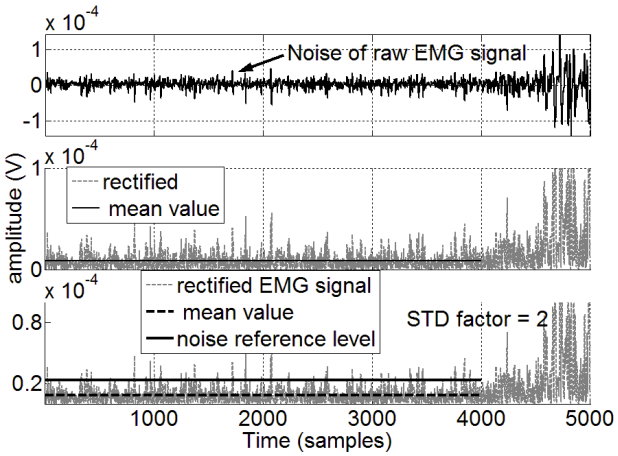

Figure 4.4: Noise threshold value estimation.

considered as a factor of standard deviation value, (dispersion). It is necessary first

to record a certain period of noise signal before EMG signal activation. There are four steps for accomplishing this task in the following way:

- Centering the signal (mean value equal zero).
- Rectifying the signal (absolute value).
- Calculation of mean value.
- Calculation of Standard deviation.

The determination of the ON-OFF timing of the muscle's activation may be found as the intersection between: 1) Moving Average Value (MAV) of EMG signal and 2) noise threshold level. These curves, figure 4.5, are measurements of thumb finger flexion activation. This estimation of noise threshold value is calculated using noise signal

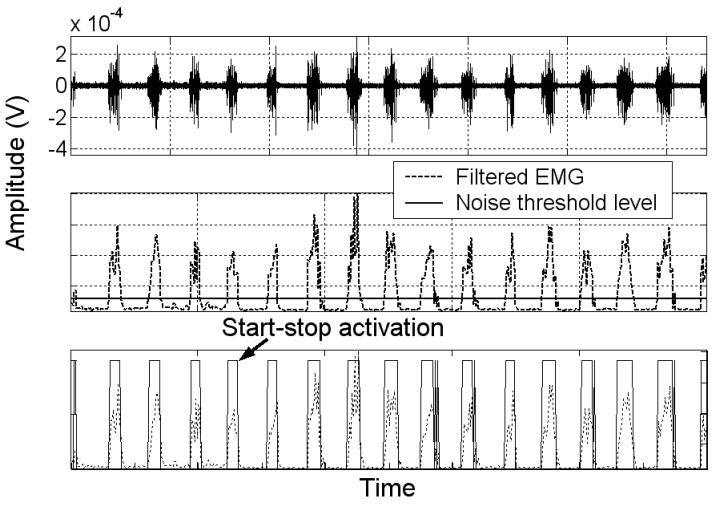

Figure 4.5: Activation periods determination based on estimated noise threshold level.

(no muscle activation) during a period of 2000 samples ($500ms$). Figure 4.5 presents from the top to bottom: 1) the measured raw EMG signal of thumb finger flexion, 2)

filtered EMG signal and noise threshold level in dashed line, 3) the ON-OFF timing activations. This noise threshold level represents the border between noise signal and activation signal. The noise reference level estimation, and consequently initial time-activation of EMG signal is depending on availability of a priori knowledge of the stochastic properties of noise's amplitude.

Concretely this method is not applicable efficiently alone in this way, because there are cases in which the beginning of activation EMG signal presents some oscillations above and under the noise threshold level like in figure 4.6. This phenomenon gives

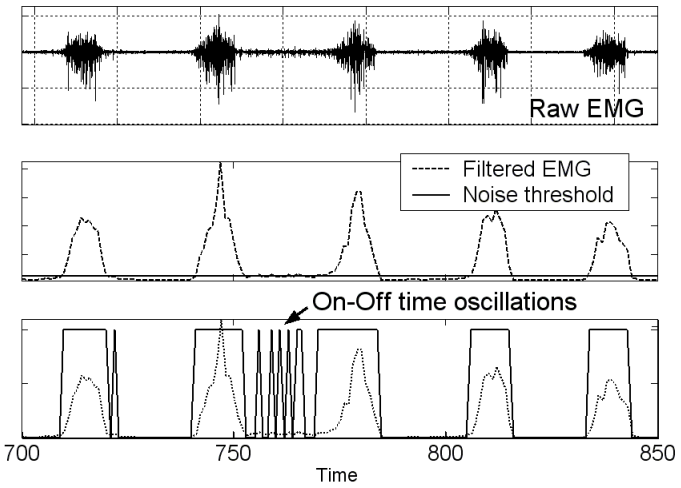

Figure 4.6: Some real oscillations above and under the noise threshold level in the beginning of ON-timing activation.

as consequence many oscillations of ON-OFF timing activations. These oscillations present a big problem for the choice of the right activation beginning (on-time). To resolve such problem it is necessary to develop an algorithm to test the duration of the activation period which is estimated, in our case, to be *320ms*. The period of activation is equal to *320ms* (1280 samples). If the duration of ON-timing, during

the first 128 ms of 320ms falls down to OFF-timing then this ON-timing activation is ignored, otherwise it will be considered as the right instance-time of the beginning of muscle's activation. This method gives the following positive results shown in figure 4.7.

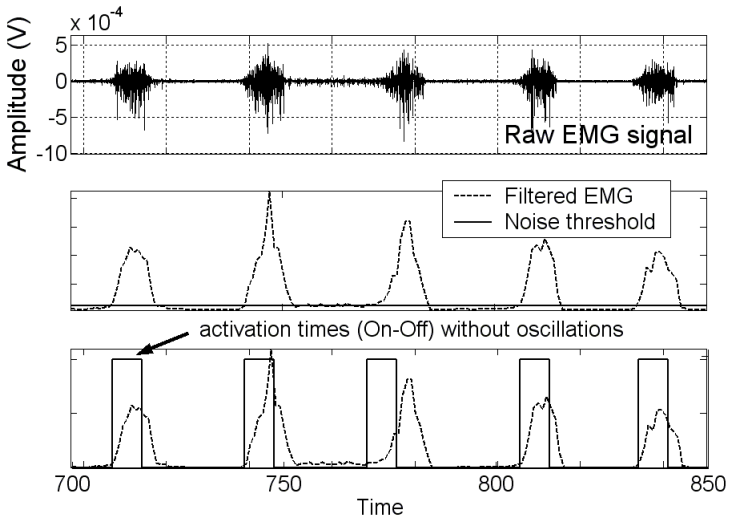

Figure 4.7: Top fig: EMG signal of middle finger flexion. Middle fig: moving average filtering of this signal. Bottom fig: Elimination of oscillations above and under the noise threshold level in the beginning of ON-timing activation and determination of desired activation signal period.

4.2.1 Conclusion

In contrast to commonly used threshold-based estimation methods for detection of activation period ON-OFF timing, the proposed algorithm proves to be reasonably accurate even for low levels of EMG activity. The improved behaviour of this algorithm, with just a modest increase in the computational complexity, can avoid the oscillations of ON-OFF timing. The aim of this Algorithm it was to use the simplest noise threshold-based estimation method, which determine the beginning level

of EMG signal, and to avoid the drawback presented by some oscillations of ON-OFF timing in the beginning of EMG signal's activation.

4.3 Filter design

Signal processing can be defined as signal manipulation for either extracting information or producing an new representation of this signal through analysis. Other motivations of signal processing are the removing of unwanted components corrupting the signal of interest, which is our study in this section and the extracting of useful information, which is the goal of our study in the following sections 4.4.2, 4.4.3 and 4.4.4. This section points on filtering and frequency domain representation of the Signal. The principle function of a filter, figure 4.8, is to filter out the unwanted parts of an input signal. The unwanted frequency parts of the signal as described in section

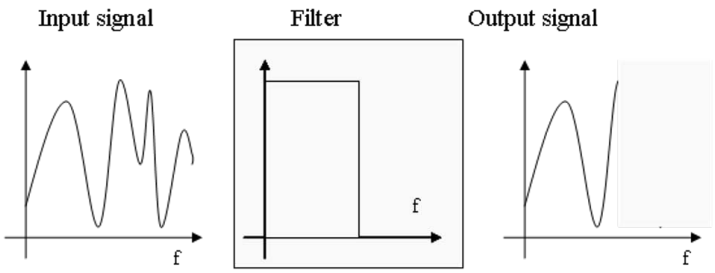

Figure 4.8: Filter effect on signal's specter.

3.3 can not be all eliminated, but only reduced. In many cases, the sequentially localisations of the information carried by the observed signal and the disturbances are generally known *a priori*. The objective is then to build a new signal from the raw signal by exclusion of the disturbances. A digital filter is just a filter that operates on digital signals represented inside a computer. There are plenty of softwares available

for designing digital filters. The effective use of a filter design algorithm requires an understanding of its parameters designing, which requires also some understanding of filter theory [64]. Filtering is a computation which takes one sequence of numbers of input signal and produces a new sequence of numbers of filtered output signal. The digital filter design methods fall into two main categories: 1) Finite Impulse Response (*FIR*) filter design and, 2) Infinite Impulse Response (*IIR*) filter design. Both these two types can be designed with any standard method (*Butterworth*, *Chebeshev*, etc...). Impulse response of a digital filter is the output sequence from the filter when a unit impulse is applied at its input. A unit impulse is a simple input sequence consisting of a single value of 1 at time $t = 0$, followed by zeros at all subsequent sampling instants. For FIR filters, the current output $y(n)$ is calculated solely from the current and previous input values:

$$y(n) = x(n), x(n-1), x(n-2), ... \tag{4.3.1}$$

This type of filter is said also non-recursive, because these filters usually require no feedback. In this case the impulse response of FIR filter is of finite duration.

The *IIR* filters are commonly implemented using a feedback (recursive) structure. The word recursive means "running back", and refers to the fact that previously-calculated output values go back into the calculation of the latest output. The expression for a recursive filter therefore contains not only terms of input values: $x(n), x(n-1), x(n-2)$, . . ., but also terms of output values: $y(n-1), y(n-2)$, ... In this case the impulse response of *IIR* filter is theoretically not of finite duration but continues for ever. The recursive terms or previous output terms feed back energy into the filter input.

Generally, to design a given frequency response characteristic, recursive filter requires

fewer terms to be evaluated by the processor than the equivalent non-recursive filter. The recursive system is specified by two vectors a and b. The coefficients of vector b are convolved with the current and past input samples, while a coefficients are convolved with the past output samples. To calculate [53] output sample $y(n)$, the filter multiplies the current and past input samples $x(n)$, $x(n-1)$, $x(n-2)$, $x(n-3)$, ..., $x(n-k)$ by the set of b coefficients: $b(0)$, $b(1)$, $b(2)$, $b(3)$, ..., $b(k)$; and sums them, then the filter multiplies the past output samples: $y(n-1)$, $y(n-2)$, $y(n-3)$, ..., $y(n-k)$ by the a coefficients: $a(1), a(2)$, $a(3)$, ..., $a(k)$ and sums them, then it combines them to form the output $y(n)$, according to this equation 4.3.2:

$$y(n) = \sum x(n-i)b(i) - \sum y(n-i)a(i) \tag{4.3.2}$$

In *MATLAB* toolbox, this whole process is performed by the filter function: $y = filter(b, a, x)$. This function uses an infinite impulse response (*IIR*) or finite impulse response (*FIR*) filter; where: x is the input signal, y the output signal, and where b and a are the coefficients. The values of these coefficients determine the characteristics of a particular filter. The order of a digital filter can be defined also as the number of previous inputs (stored in the processor's memory) used to calculate the current output. In the case of recursive filters, the definition can be extended to previous input and output values required to compute the current output.

Before to go farther it is preferable to talk about the aliasing problem, which has been described already in section 3.1. To prevent aliasing problem it is more advantageous to filter the continuous-time signal, using analog filters before sampling it, (biomedical applications involve the acquisition of continuous-time signals).

Matlab toolbox has several design algorithms that can be used to create both *IIR* and *FIR* digital filters. The *IIR* filters that can be created in *Matlab* are *Butterworth*,

Chebyshev-1, *Chebyshev-2*, and *elliptic*. The *FIR* filter algorithms in *Matlab* are *equiripple*, *least squares*, and *Kaiser window* types. We should know the parameters of the filter that we are going to design. Some of these parameters are described in following section.

4.3.1 Optimised filter design

Design mode allows us to specify a *FIR* or *IIR* filter by setting design parameters such as filter type, passband/stopband edge frequencies, pass-band and stop-band ripple levels, and stop-band attenuation. We can select from design methods that include Butterworth, *Chebyshev*, Inverse *Chebyshev*, *Elliptic*, *Kaiser*, *Equiripple*, ... The designer should then use different parameters to suggest a filter meeting as many of those specifications as possible. The goal is then to optimise the designed filter to meet desired needs. The *MATLAB* signal processing toolbox contains a number of different functions for designing recursive low-pass, high-pass and band-pass filters. Using *Matlab*, a digital filter is designed with various prototypes: *Chebyshev*, *Butterworth*, and *Elliptic* for *IIR* type. *Equiripple*, *least squares*, and *Kaiser window* are designed for *FIR* type. The optimum filter type is chosen on the basis of implementation complexity and magnitude response. The design specifications of the band-pass filter and the order are given for the following examples. A comparison of these filters is attempted in this section in order to evaluate the advantages and drawbacks of each filter for the same band frequency, which equal to $BP = [30Hz,\ 500Hz]$.

1) Infinite Impulse Response (*IIR*) digital filters:

The Butterworth filter, for Infinite Impulse Response filter design without specified requirements, is often sufficient. More rigorous filter requirements can be met with

Chebyshev and *elliptic* filters, figure 4.9. For each type of *IIR* filter, three order values have been chosen: 2, 4 and 6. These filter responses figure 4.9 designed with

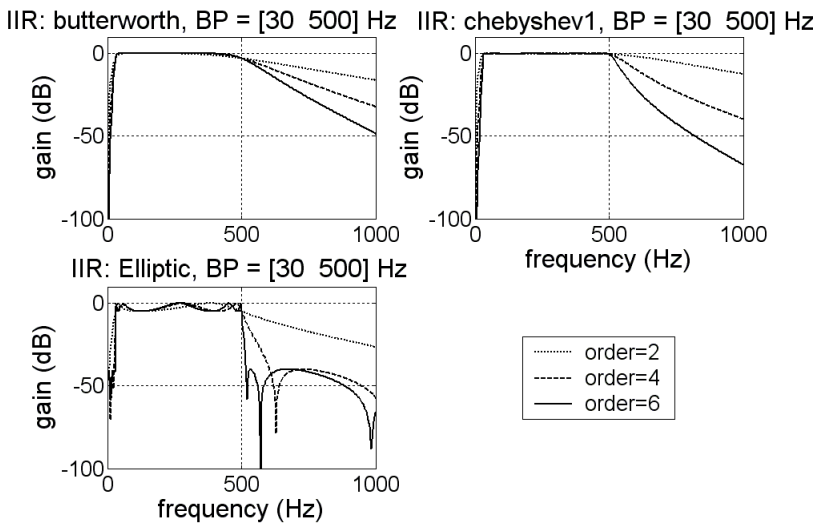

Figure 4.9: Infinite Impulse Response (*IIR*) band-pass filters comparison for orders $n = 2,\ 4\ and\ 6$.

low orders, which are 2, 4 and 6 seem to be good acceptable. We can conclude that the elliptic filter of order 6 presents the best band-pass frequency response, because its pass-band and stop-band cutoff frequencies transition are fast, in comparison with the two others.

2) Finite Impulse Response (*FIR*) digital filters:

FIR filters require a much higher filter order than *IIR* filters to achieve a given almost same level of performance. The *MATLAB* function *fir1(N, Wn, type of window(N))* designs conventional *FIR* filters based on the windowing method. Without explicit specifications, the Hamming window is employed in this design. Other windowing functions can be used by specifying the windowing function as an extra argument of

the function. For example, *Blackman* window, *Hanning* window or *rectangular* window. The *Parks-McClellan* method (called "*Remez*" by *Matlab*) designs FIR filter of order N based on *Parks-McClellan* algorithm and exhibits an *equiripple* behavior in their frequency responses and are sometimes called *equiripple* filters. Filter specifications are given in terms of pass-band and stop-band cutoff frequencies, moreover using also pass-band and stop-band ripples attenuation. Some of FIR filter types are presented in figure 4.10. Note that the frequency response of FIR filter based on *Parks-McClellan* algorithm, presented in figure 4.10 has a high stop-band gain, this is due to the narrow transition band. If the transition band becomes larger we will get lower stop-band gain. There is a trade-off between stop-band gain and transition width. Both IIR and FIR defines a class of digital Filters. IIR filters, which may

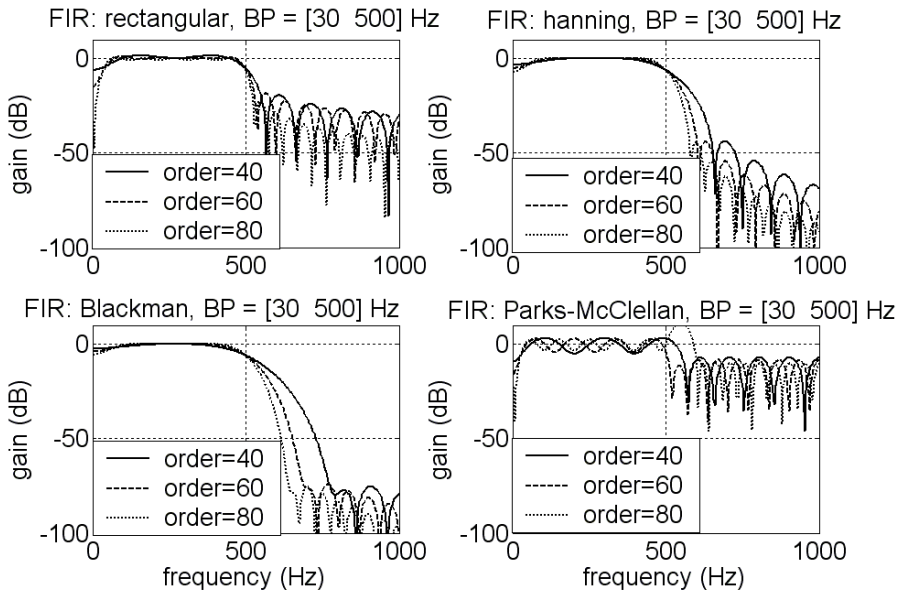

Figure 4.10: Finite Impulse Response (FIR) band-pas filters comparison for orders n=40, 60 and 80.

have both zeros and poles on the z-plane, are not guaranteed to be stable, and they

have nonlinear phase responses. FIR has zeros only on the z-plane, the consequences of this are that FIR filters are always stable, and they have linear phase responses (filter's coefficients are symmetrical). In this case the delay is constant for all frequencies.

A simple design specification for a filter is to remove noise outside a certain band-pass frequency. A more complete specification need some other specific characteristics like pass-band ripple (R_p, in decibels), stop-band attenuation (R_s, in decibels), or transition width (W_p, W_s, in hertz). These specifications should achieve the performance goals with the minimum filter order. Such task can be done using the following *matlab* functions like: *chebyord*, *butterord*, *ellipord*, ...

Filter Specifications in *Matlab* are:

- W_p: Pass-band cutoff frequencies
- W_s: Stop-band cutoff frequencies
- R_p: Pass-band ripple: deviation from maximum gain (dB) in the pass-band
- R_s: Stop-band attenuation: deviation from 0 gain (dB) in the stop-band

In figure 4.11, two examples are presented to compare the required order for each type of filter for almost the same specified characteristics, which are: stop-band cutoff frequency $W_s = [30Hz,\ 500Hz]$, Pass-band cutoff frequencies $W_p = [40Hz, 400Hz]$, $R_p = 2dB$ (ripple in the pass-band), $R_s = 20dB$ (attenuation in the stop-band) for $IIR\ Chebychev$ filter. Concerning FIR filter with $Kaiser$ window, the parameters of vector $devs = [0.29,\ 2,\ 0.29]$, specify the pass-band, ripple and the stop-band attenuation in absolute values and not in decibels. For almost same filter characteristics, FIR with $Kaiserwindow$ required an order $n = 80$, but $IIR\ Chebychev$

filter has required only an order $n = 6$. Note that the number of filter coefficients

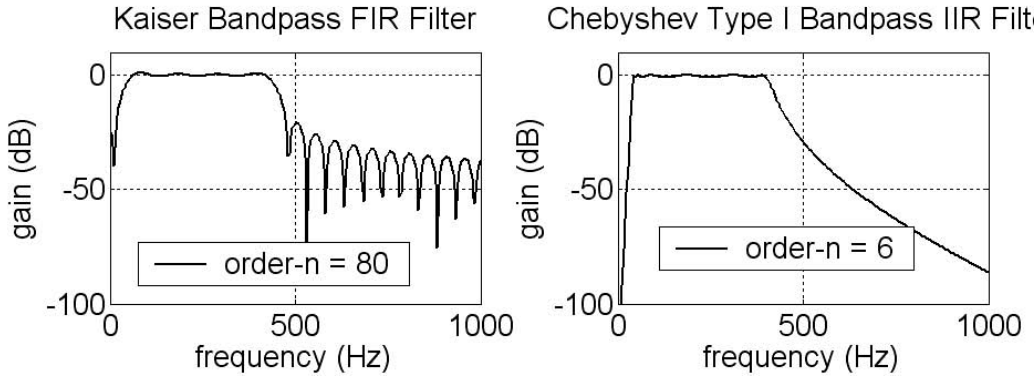

Figure 4.11: Required order for both FIR and *IIR* filters in case of almost same characteristics.

affects significantly on the computational effort needed for the designing of the filter. A large number of filter coefficients requires larger computational time that may not be feasible in certain real-time applications. A general desire in any filter design is that the number of operations (additions and multiplications) needed to compute the filter response is to be as low as possible.

4.3.2 *FIR*-80^{th} and *IIR*-6^{th} order filter responses for different window types

This section deal with the frequency response of an FIR-80^{th} order and IIR-6^{th} order Pass-band filters corresponding to different window types. We will consider different window characteristics in frequency range of $30-200Hz$, and will compare the quality measures and complexity issues related with these two design techniques. For IIR filter the window types used are: *butterworth*, *chebychev-1* and *elliptic*, which are presented in figure 4.12. In figure 4.13 are presented the frequency responses of four 80^{th} order FIR filter corresponding to the following four windows: *rectangular*

window, *hanning* window, *Blackman* and $Parks - McClellan$. The filter order

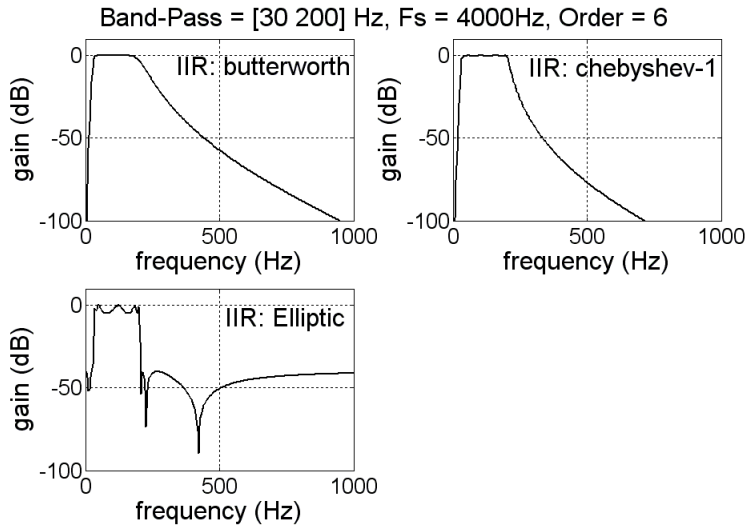

Figure 4.12: Frequency response of IIR-6^{th} order filter for three different windows.

is given by the length of its filter impulse response, which can be considered as its measure of complexity. Although IIR filter design has smaller order (6^{th} order), its frequency response is comparable with that of the 80^{th} order FIR filter. On viewpoint of complexity IIR filter is preferable. In 6^{th} order IIR filter design, the measure of filter quality is good enough with also less complexity.

4.3.3 Order effect of *IIR*-elliptic filter

The investigation of the viability of myoelectric signal recognition by different filtering processes is considered. EMG signal analysis need first the use of low-pass anti-aliasing analog-filter. This anti-aliasing analog-filter has cutoff frequency somewhat above $500Hz$, in this case $900Hz$, and has sampling frequency at almost four times the highest frequency, in this case $4000Hz$. The digitised EMG signal can be then filtered.

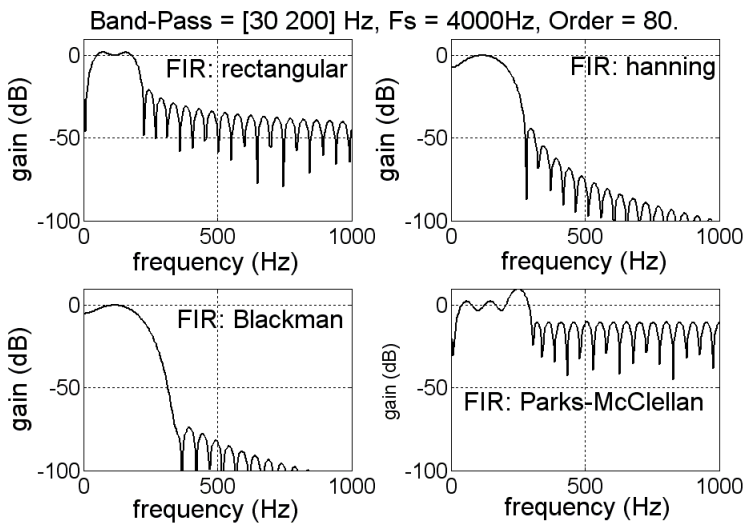

Figure 4.13: Frequency response of FIR-80^{th} order filter for four different windows.

A variety of above IIR-elliptic filter were applied with different orders: 2, 4 *and* 6. EMG signals corresponding to three different finger movements, which are the flexion of thumb-, pointer- and middle-finger are considered. Three different features are extracted from these filtered signals and classified using RBF intelligent classification method. These features are Moments of frequency 'M_n' given in equation 4.3.3 with different values of n (order), $n = 0,\ 1$ *and* 2.

$$M_n(t) = \sum_k \omega_k^n \mid STFT(t,k) \mid, n = 0, 1, 2, 3, ... \tag{4.3.3}$$

where

M_n : is the n^{th} moment of the frequency distribution at time t,

n: order,

ω : frequency.

The following figure 4.14, in case of Elliptic IIR filter and Radial Basis Function (RBF) classification method, presents the classification results of these three finger movements. Radial Basis Function (RBF) neural network architecture is designed

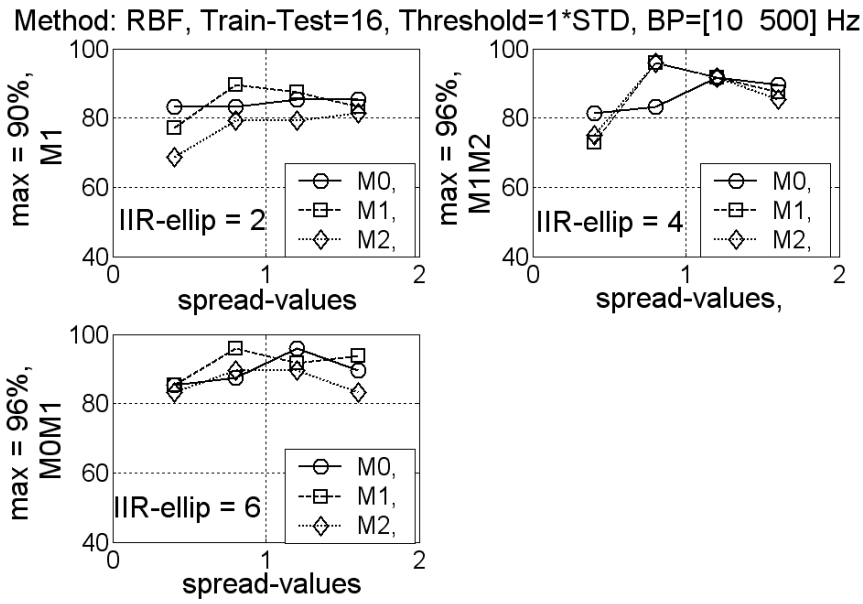

Figure 4.14: Effect of IIR filter orders on EMG signal classification accuracy.

and trained by "*newrb*" *Matlab* function with 16 test sets and 16 training sets. The output layer is linear and the rate of classification is depending on spread values of hidden unit. Hence four values of spread between 0.4 and 1.6 with a step of 0.4 are used. The evaluation of the filter order effect on EMG signal classification accuracy will be clearly presented with these four different spread values and three different features. The pass-band frequency is chosen to be in the range of $10 - 500Hz$. These results show clearly that the increasing of filter order values has a positive effect on EMG signals classification.

4.3.4 Effect of different filter window types

All filters, FIR and IIR, described in section 4.3.1 will be used to compare the discrimination accuracy between them, figure 4.15. The choice of filter type is an important decision for EMG signals recognition. The features and the method of

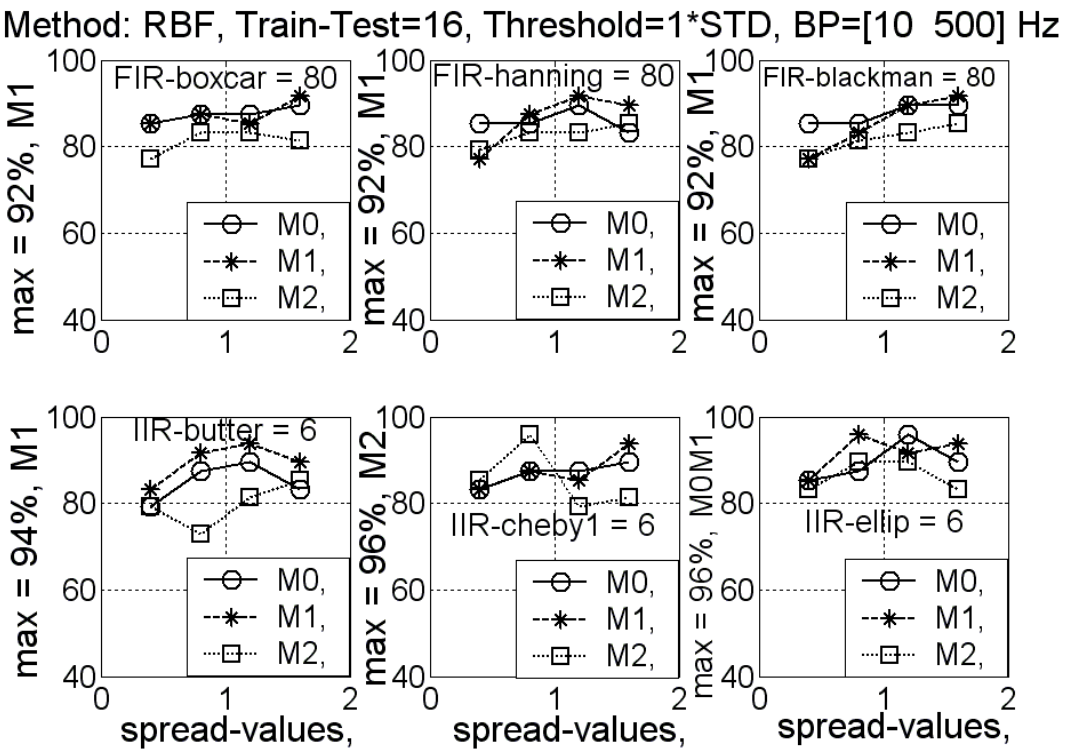

Figure 4.15: Effect of different window types of 6^{th} order IIR filter and 80^{th} order FIR filter, on EMG signal classification accuracy.

classification used in this part are the same with those given in above section 4.3.6, and are illustrated in figure 4.14. The orders of IIR and FIR filters are selected with consideration of the best results found in above study, section 4.3.1, figures 4.9 and 4.10). These all filters are tested on real EMG signal measurements of three different finger movements. RBF classification method is used to evaluate the effect of the window type for both 80^{th} order FIR filter and 6^{th} order IIR filter design.

4.3.5 Pass-band effect of *IIR*-elliptic filter

Filters play a vital role in data acquisition and processing systems to remove unwanted selected frequencies from an incoming EMG signal and minimise artifacts, conducted

disturbances and emitted disturbances. EMG signal offers a great deal of useful information, which is depending on its band frequency. For some features, like moment of second order (*M2*) illustrated in this following example given in figure 4.16, the choice of the Pass-band filter is very important in viewpoint of its information. This

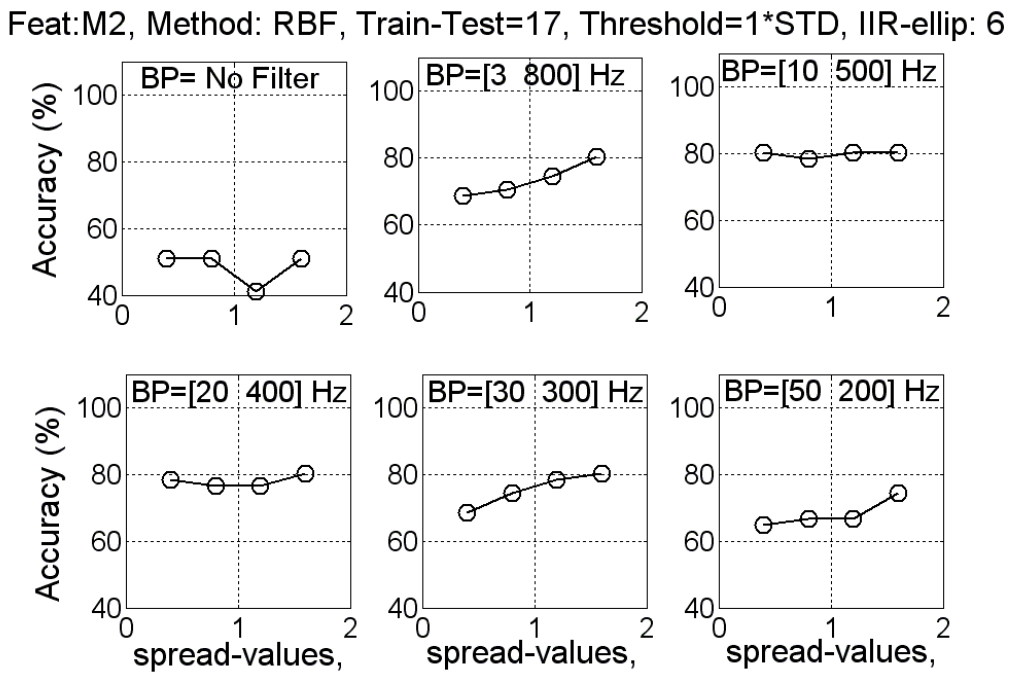

Figure 4.16: Effect of *IIR-elliptic* filter's Pass-band on EMG signal classification.

information depends on filter frequency pass-band. The figure 4.16 shows clearly that the different pass-band widths have an influence on the classification accuracy. As example three finger movements (thumb-, pointer- and middle-finger) are considered. Five pass-bands are chosen, which are: *3-800Hz*, *10-500Hz*, *20-400Hz*, *30-300Hz*, *50-200Hz*. Classification method used here is Radial Basis Function (*RBF*) method, which considers four spread values: 0.4, 0.8, 1.2 and 1.6. The best classification results are obtained with pass-band equal to *10-500Hz*. It is necessary here to give this following important remark. This optimised pass-band found for used feature (*M2*)

can not be generalised for all other features. More details about this observation and its approvement is given in

chapter 7, section 7.2.

4.3.6 Conclusion

The signals in a digital filter are represented by finite and quantised binary values. In this section several filter orders, filter types, windows and filter pass-bands were designed, tested and compared to evaluate their effect on an electro-myogram signal recognition. These systems are generally used to perform a filtering operation. It is important to evaluate the effect of these different filtering methods on the EMG signal recognition to be able to choose the optimal one.

The cost of the filter is determined by its complexity. This complexity can be evaluated on the basis of the following four simple parameters: order, adder operations, multiplier operations and delays, [23]. If only order-parameter is considered, the classification results occurred with RBF classification method, figure 4.15 are almost the same for these both filters(IIR filters, which have $order = 6$, and FIR filters, which have $order = 80$). It is possible here to conclude that IIR filters are less complex and lead to the same accuracy classification results than FIR filters, which are more complex. About the choice of pass-band filter, which have a great importance to transmit a well-defined information and to reject other disturbances, it is for us now not possible to confirm if the pass-band filter *10-500 Hz*, found in above section as best one for the feature $M2$, can be generalised for all other features. It will be shown in section 7 that the described information inside the same signal through different features is not located in the same frequency bands of this signal. Finally six different

features, three of them belong to 6^{th} order IIR filter and three others belong to 80^{th} order FIR filter, are evaluated and compared to get their classification effects. Most of these filters have almost the same results for feature $M2$ and for frequency band in the range of *10-500Hz*.

4.4 Signal analysis and feature extraction

4.4.1 Introduction

Signal representation is very important before to deal with features extraction. There are three known different representations of a signal: 1) time-domain, 2) frequency-domain and 3) time-frequency domain representation.

The first study will be focused on features extraction in time domain, then we use the frequency domain and finally time-frequency domain is considered. EMG signal is a very complex signal. Generally a signal is a carrier of information, which can be represented as a function of variables.

$$Signal = f(x, y,). \tag{4.4.1}$$

Signals, e.g. an Electromyograph (EMG), are signals of complex physical phenomena varying in the time:

$$Signal = f(t; x, y,). \tag{4.4.2}$$

The EMG activity represents the sum of potentials of all active motor unit actions, figure 4.17, under the derivative area of the electrodes. For the analysis of these real and complex EMG signals, special methods of analysis are necessary, which allow the examination of the important information variability in the temporal change of

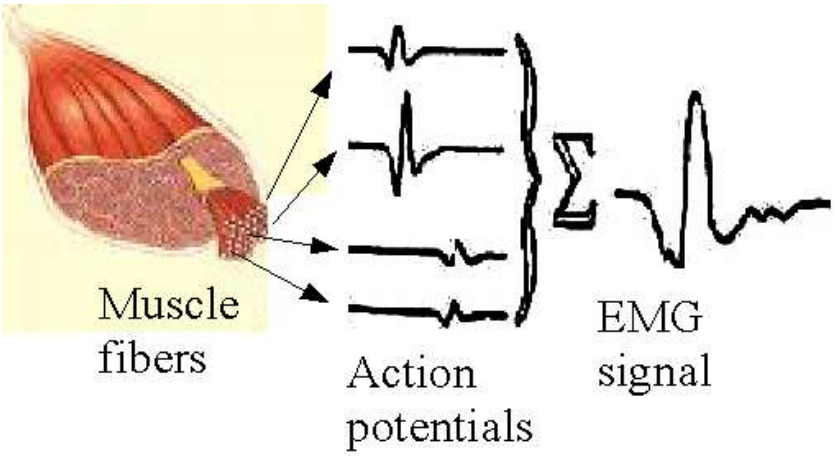

Figure 4.17: Active motor unit actions

a feature "x". Thereby it is possible to build several clusters regarding to clusters information differences. The acceptable features obtained using certain extraction methods from a signal, which belongs to one category, should have the following characteristics:

- they have strong discriminating capabilities
- they are robust and reliable
- they are not time consummating
- they don't have many parameters

This set of attributes is called a signature for the associated signal. This signature can then be used to detect the presence of similar attributes in unknown data. Since the EMG signals are non-stationary these signatures will be extracted using a Time-frequency analysis of the signals, like Short Time Fourier Transform ($STFT$). Briefly, in this chapter we want to answer these two following questions:

1) Which analysis methods are suitable to use for extraction of the useful features? -
2) Which features are the best for the discrimination of different muscle dynamics?

4.4.2 Time domain feature extraction

We want to test the different known forms of EMG signal representations that are more efficient for feature extraction. There are three known representations for each time varying signal:

- Amplitude vs. time representation (2D dimensional space).
- Amplitude vs. frequency representation (2D dimensional space).
- Time-Frequency vs. Amplitude representation (3D dimensional space).

The process begins with reading the EMG signals from two surface electrode channels attached to the test subject's forearm. The transient part (the beginning part) of EMG signal during *400 ms* has been exploited to extract time domain features for the recognition of 4 movement classes. These classes are: 3 finger flexion movements and hand closing. These movements were identified when the signal's envelope crosses the noise threshold level (see section 4.2, which represents the considered noise level reference. The used signal can be extracted from each initial part of signal, which needs to be done synchronously from both channels. Once the required part of the Myoelectric signal is obtained, many time domain features of signal can be extracted like:

- Mean absolute value (MAV),
- Variance (VAR),

- Waveform length (WL),
- Root mean square(RMS).

We take in consideration only one feature, mean absolute value (MAV), as example to show the distribution of these feature instances. Two figures are presented for raw signal, figure 4.18, and filtered signal in *20-250Hz*, figure 4.19. The values are normalised to get mean value equal zero and variance value equal to one. This

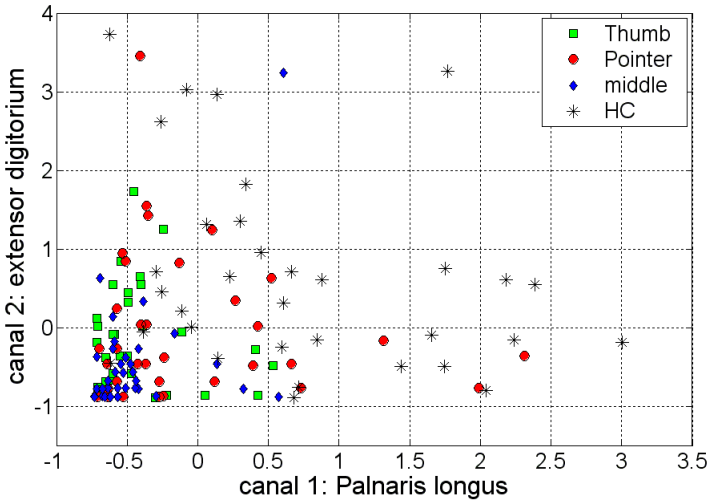

Figure 4.18: Mean absolute value instances distribution with raw EMG signal

example shows us that the features extracted simply, from raw and filtered EMG signal in time domain presentation, without any signal analysis are not relevant. The four classes are not regrouped in discriminated clusters, hence we can't differentiate between them. Temporal approach can not extract important information for the classification of these four difficult gasp types, especially with only two measurement channels.

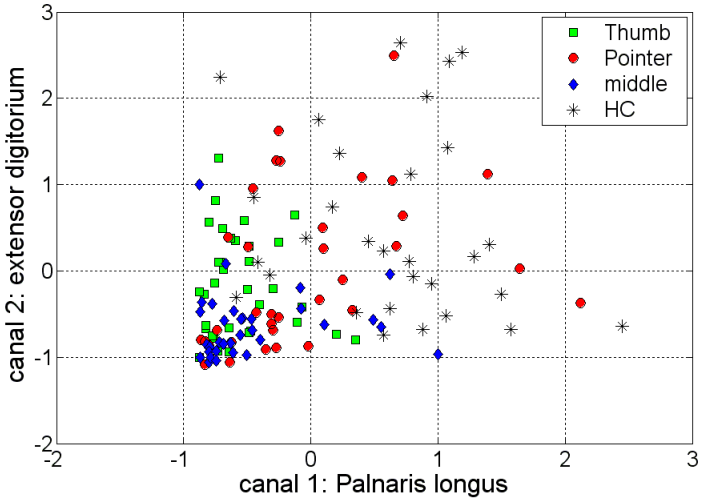

Figure 4.19: Mean Absolute value instances distribution with filtered EMG signal, $20 - 250Hz$.

4.4.3 Frequency domain feature extraction

a) Frequency domain analysis:

The frequency domain is used to extract information contained in EMG signal. The transformation from time domain to frequency domain [46] is achieved through the use of the Fourier transform, which allows us to look at EMG signal energy as a function of frequency. Fourier Transform method is an optimal solution when we assume that there is no frequency change, for each component, over entire time of Analysis. Such analysis does not take in consideration the information on a time localisation of the frequency component of the signal. The Fourier Transform (FT) is defined in equation 4.4.3:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} \tag{4.4.3}$$

$X(f)$ is a complex function of frequency, f, which describes the complex voltages (amplitudes and phases) as a function of frequency, f, of the signal $x(t)$.

The Discrete Fourier Transform (DFT):

If $x(t)$ is time-function limited to a duration of nT samples then DFT, equation 4.4.4, can convert a sampled function of time $x(nT)$ into a sampled function of frequency $X(mF)$. In simple terms we use the DFT to represent digital signal, $x(nT)$, of length nT as a sum of m different sinusoidal waveforms. In these sinusoidal waveforms each sinusoidal function (complex exponential function) $X(mF)$, will have only one single frequency amplitude and phase.

$$X(mF) = \sum_{n} x(nT)e^{-jnm2\pi} \tag{4.4.4}$$

The Fast Fourier Transform (FFT) is a class of algorithm, which deals only with time computation reduction. It allows the computation of the DFT to be performed in O(N log N) rather than O(N^2) computations. The DFT of EMG signal produced with thumb-finger flexion during $400ms$ and sampled at $F_s = 4Khz$, is presented in figure 4.20

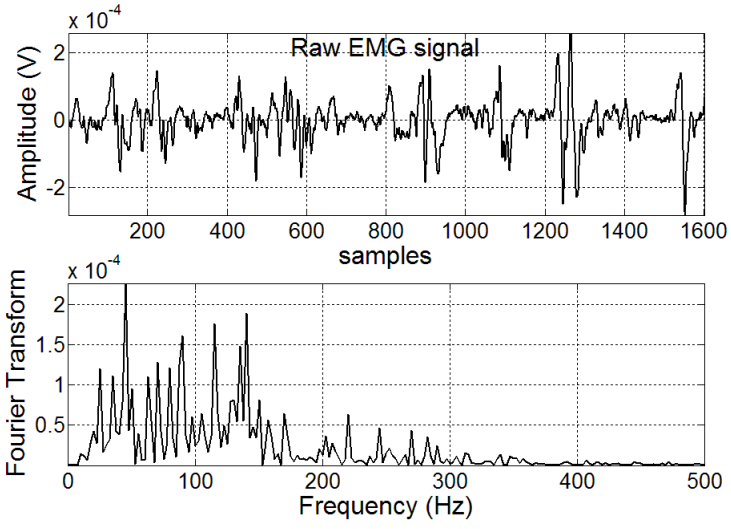

Figure 4.20: Fourier Transform (FT) analysis of raw EMG signal of 400ms length, corresponding to thumb finger flexion movement.

b) sinusoidal harmonic waves

The DFT is used to model our EMG signal as a sum of simple sinusoidal signals. The magnitudes of spectral lines of these simple sinusoidal signals quantify their energy contribution for the global EMG signal. However EMG signals are much more complex than simple sinusoidal functions. For a signal $x(nT)$, which contains nT data samples, the DFT in this case is resulted in nT discrete harmonically related sinusoids. The spectral lines will be occurred at the fundamental frequency that equal to $\frac{F_S}{nT}$. This fundamental frequency can be used to get all decomposed signal frequencies of our original signal $x(nT)$. In the following figure 4.21 we present 8 sinusoidal decomposed signals. These signals are contained in a band of frequency between $2.5Hz$ and $2400Hz$, of EMG signal according to thumb finger flexion recorded during $400ms$ and sampled at F_s equal to $4kHz$. Each signal of figure 4.21 is a combination of 12 ele-

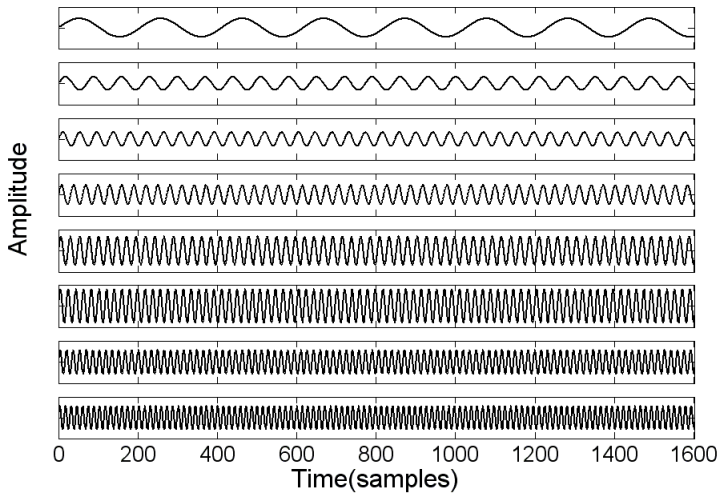

Figure 4.21: Hand closing Raw EMG signal decomposition, in 8 group signals. Each of them is composed of 12 spectral lines issued from DFT analysis method.

mentary sinusoidal decomposed DFT's signals, and its frequency is the average value

of their frequencies, and its amplitude is a sum of their amplitudes. Average frequencies for these 8 signals are: $15Hz$, $45.5HZ$, $75Hz$, $105Hz$, $135Hz$, $165Hz$, $195Hz$ and $225Hz$. The domain frequency covered by each spectral line is equal to $2.5Hz$. We use this representation of the EMG signal, which is frequency-domain analysis to know, through extraction of correspondent features, if there is amelioration in discrimination classes. The changes in the spectrum of EMG signal have been used as an objective measurement of muscle dynamics. The used signal can be extracted from initial part, which needs to be done synchronously from two channels. Once the required initial part of the Myoelectric signal is obtained, many frequency-domain features, which are known in the literature, can be extracted like:

- Median frequency
- Mean frequency

c) Frequency domain feature extraction

1- Median frequency:

We attempt to improve the discrimination capability for our four classes using features related to domain frequency representation. One measure of the frequency content in a signal is the median frequency. It's the measure of the EMG signal frequency that divides the signal into two halves of equal power. The feature samples are presented in two dimensional feature space ($2D$) defined with two measure channels, figure 4.22

2- Mean frequency:

The second measure of the frequency content in a signal is the mean frequency. The

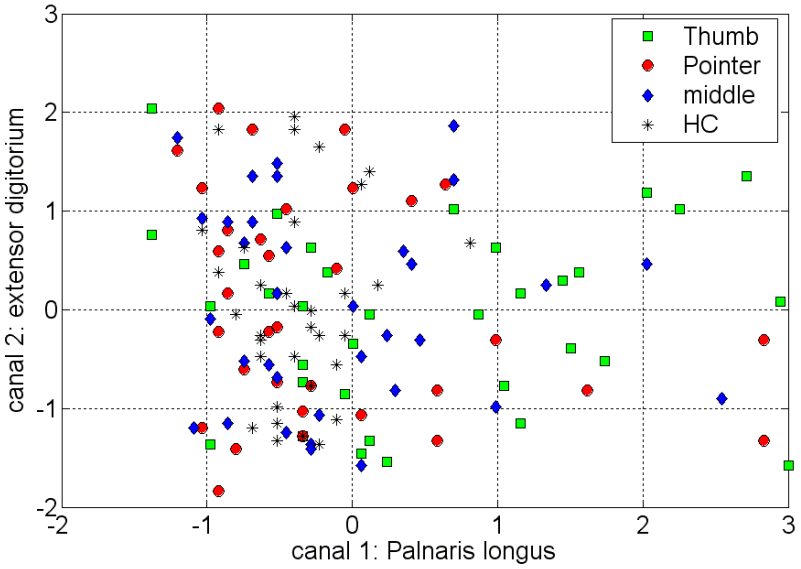

Figure 4.22: Median frequency feature instances distribution corresponding to frequency-domain EMG signal representation.

mean frequency, equation 4.4.5, can be determined from the FFT as:

$$MeanFrequencyFeature = \frac{\sum_i f_i {A_i}^2}{\sum_i {A_i}^2}, \tag{4.4.5}$$

These feature values are presented in two dimensional feature space defined with two measurement channels, figure 4.23. Clusters discrimination with this feature, mean frequency, is more clear than with median frequency. In figure 4.23 it can be clearly distingued the pointer finger class represented with circles. The second class, which is less discriminated is Hand closing that is represented with stars. Between thumb finger cluster and middle finger cluster there is a big interference. This second measure, mean frequency, has more discrimination accuracy than median frequency.

Remark:

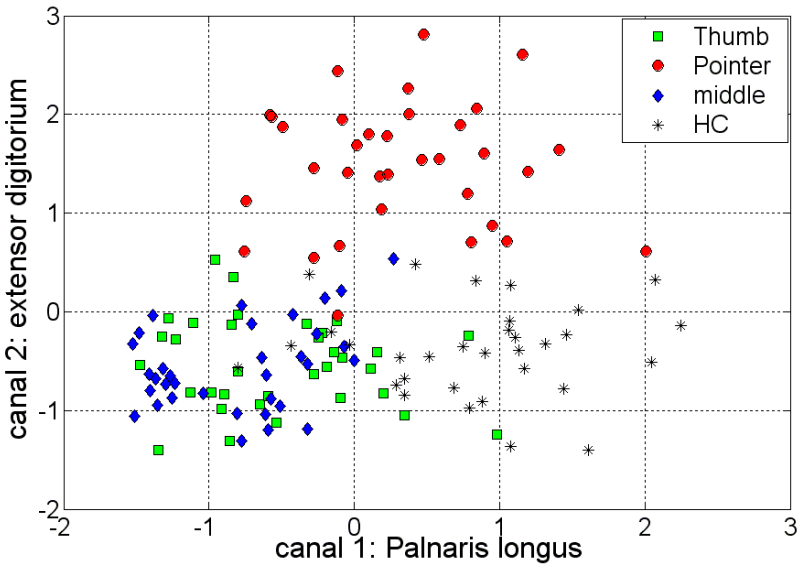

Figure 4.23: Mean frequency feature instances distribution corresponding to frequency-domain EMG signal representation.

A new mean-frequency feature can be defined in the following equation 4.4.6:

$$MeanFrequencyFeature = \frac{\sum_i f_i A_i}{\sum_i A_i}, \tag{4.4.6}$$

where A_i is the FFT amplitude at frequency f_i. The A_i values are not squared like in the above example. In this case we get the following distribution of feature samples, figure 4.24: If we compare this distribution with the above one in figure 4.23, the discrimination accuracy became worse.

3- Norm of power density:

A new feature in the frequency domain is defined, which is the spectrum's norm (*Norm-Spctr*) of the signal EMG. The spectrum's norm can be determined as:

$$\sqrt{\sum_i ({A_i}^2)} \tag{4.4.7}$$

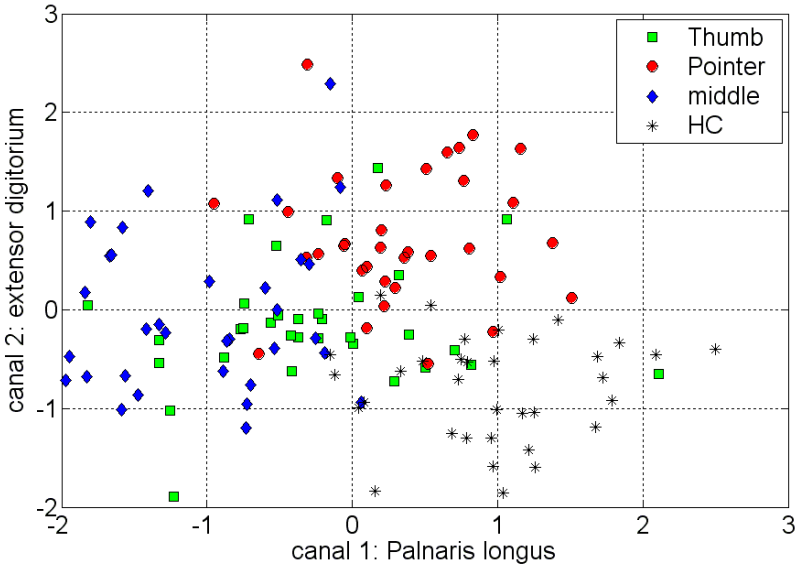

Figure 4.24: Mean frequency feature instances distribution corresponding to frequency-domain EMG signal representation.

where A_i is the Spectrum amplitude at frequency f_i, and the summations are token over all frequencies in the spectrum.

Power spectrum estimates the Power Spectral Density of the signal EMG using $Welch$'s averaged periodogram method, figure 4.25 As it is shown in figure 4.26, our investigations in frequency domain for relevant features have lead to an improved clusters discrimination of our four classes.

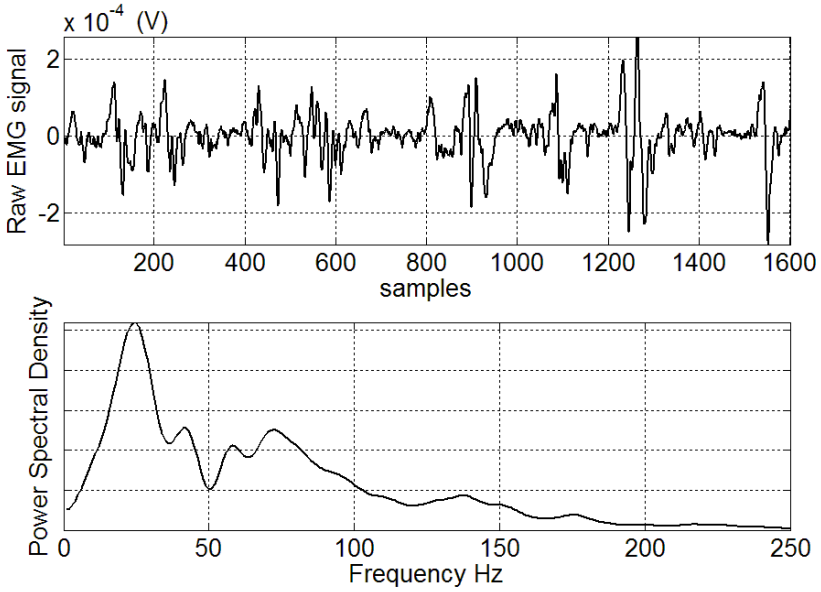

Figure 4.25: Raw EMG signal of 400ms length and its spectral power density.

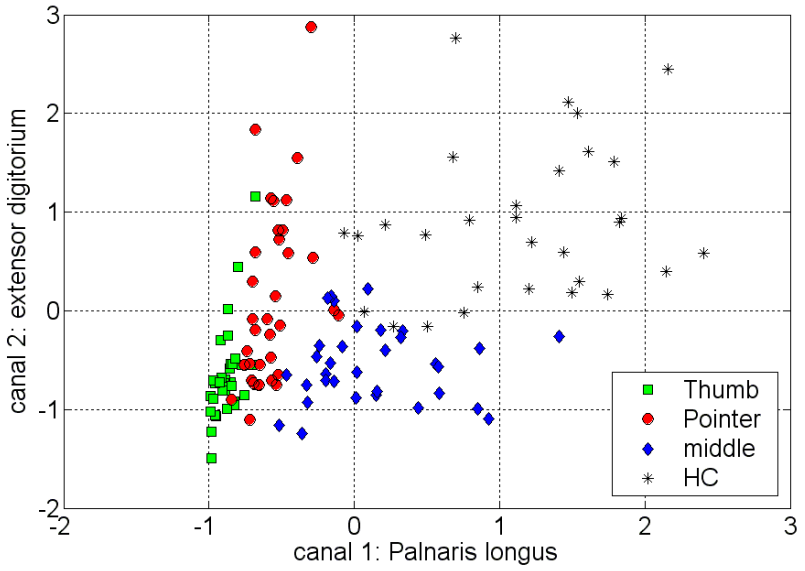

Figure 4.26: Norm of power density feature instances distribution corresponding to frequency-domain EMG signal representation.

4.4.4 Time-frequency domain feature extraction

a) Time-frequency domain analysis

Mono-dimensional signal analysis seems not to be sufficient for extraction of relevant information to characterise signals of complex systems like EMG signals. Therefor we have to consider bi-dimensional, time and frequency, analysis methods. Time-frequency representation combines time domain and frequency domain analysis to get temporal localisations of a signal's spectrum. The Short Time Fourier Transform ($STFT$) considers that the statistical properties of the non-stationary signal are varying in the time. This method of analysis help to extract the information according to the signal time variation. The choice of time-window to track these variations of the signal is of great importance.

This particular Fourier-based analysed method, $STFT$, designs smooth time windows $W_i(t) : i = 1, .., p$, figure 4.27, to chop a given signal into short p pieces and then applying the DFT to each piece. Since we use a signal of nT values length, we have to consider that $p \leq n$. We can't simply shop the signal into short pieces, without smooth functions, because this will cause sharp discontinuities between these sections. Hence the smooth windowing is constructed by multiplying the signal x_{nT} by the time-window $W_i(t)$.

p: number of windows, and $W_i(t)$: short time analysis window

$$X_i(mF) = \sum_n x_i(nT)e^{-jnm2\pi} \tag{4.4.8}$$

The easiest way to be sure that there is continuity between ends of these pieces of signals is to force them to be zero at the extremities, thus their values is necessarily the same. The choice among many existent window functions depends on knowledge

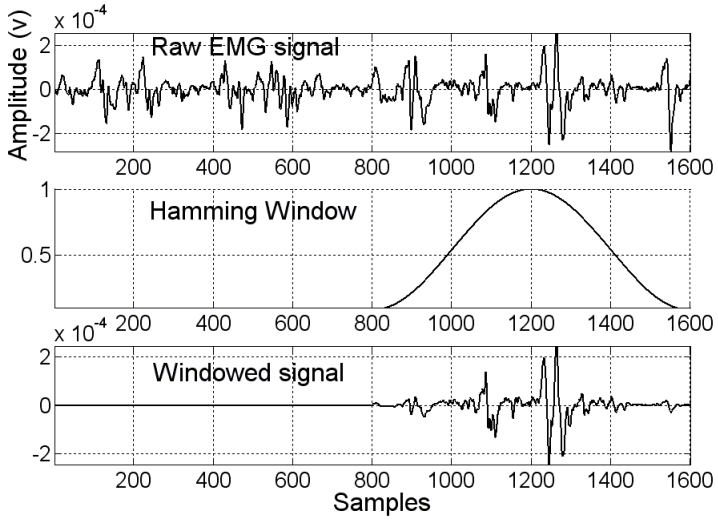

Figure 4.27: An example of how the EMG signal is windowed to create a new signal with smoothed extremities.

of the signal and application. Now it is possible to apply DFT on each time-windowed signal, figure 4.28.

After summation of all Fourier transformed signals $X_i(mF)$; where $i = 1, \ldots, p$; we get as consequence result the Short Time Fourier Transform of our original signal $x(t)$. Thus the $STFT$ considers the signal $x(t)$ as a series of $DFTs$ of time-windowed pieces. It remains some questions about how to choose the time-window length and the rate of time-windows overlapping, which depend on the application.

Now it's possible to identify how the frequency content of the signal evolves over time. An analysed part of Hand closing EMG signal during $400ms$ and for $25ms$ time-window length, using $STFT$ method representation, can be shown in the following figure 4.29. We get the following presentation of muscle activation:

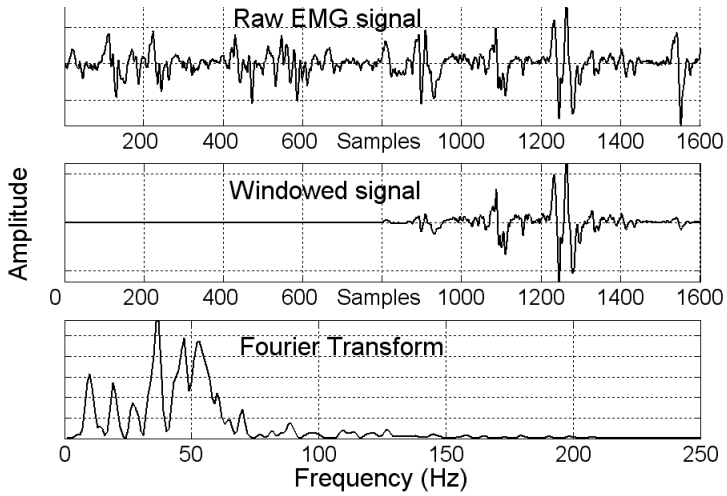

Figure 4.28: Fourier Transform for a windowed EMG signal corresponding to thumb finger flexion.

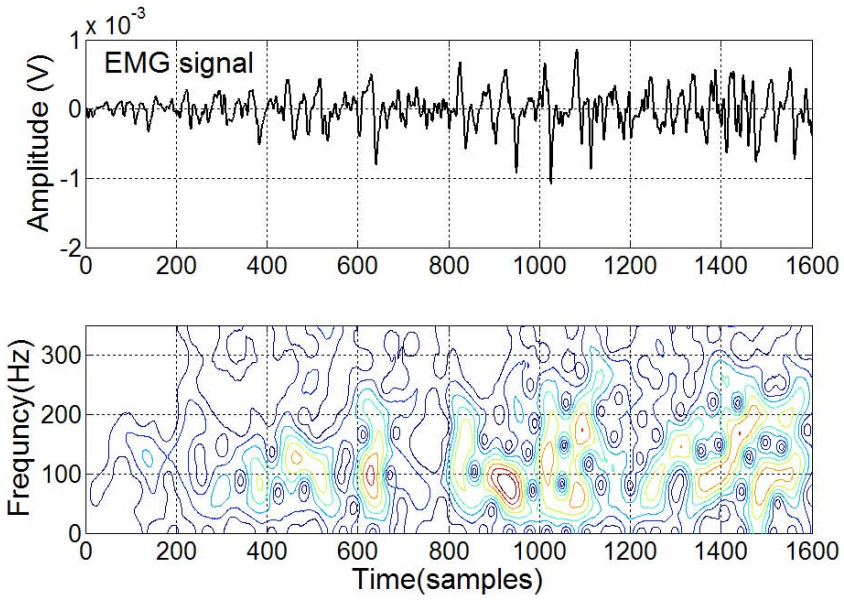

Figure 4.29: $STFT$ analysis of $400ms$ length EMG signal and for $25ms$ time window length, corresponding to hand closing movement using contour presentation.

b) sinusoidal harmonic waves

If the digital signal, $x(nT)$, of length nT is represented, using DFT, (as a sum of m different sinusoidal waveforms), these sinusoidal waveforms have only one single frequency component independently of time. However with $STFT$ transform the amplitudes of these spectral lines are not constant. For the same signal in figure 4.29 the spectral lines are presented in the following figure 4.30. Each signal of figure 4.30

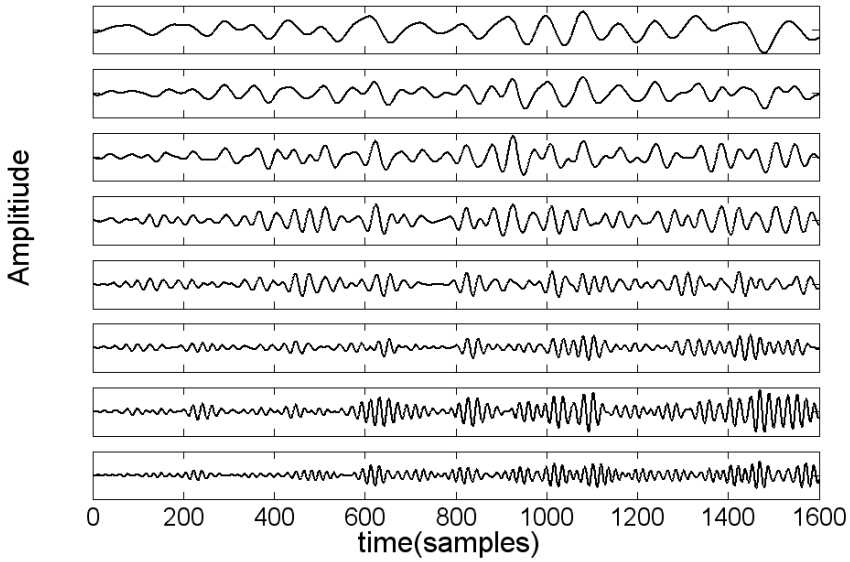

Figure 4.30: Hand closing Raw EMG signal decomposition in 8 group signals. Each of them is composed of 12 spectral lines issued from $STFT$ analysis method. Signal length is equal to $400ms$ and sampling frequency equal to $4KHz$.

is composed of 12 elementary spectral lines obtained from $STFT$'s signal analysis. Middle frequencies for these 8 signals are:

$15Hz$, $45.5HZ$, $75Hz$, $105Hz$, $135Hz$, $165Hz$, $195Hz$ and $225Hz$.

c) Feature extraction

After time-domain and frequency-domain features extraction, we want to improve more the clusters discrimination of the four classes of hand movements described above using time-frequency domain features extraction. EMG signal pre-processing operation using spectrum analysis based on Short-Time Fourier Transform ($STFT$) is applied. This analysis is a form of local Fourier analysis that treats time and frequency simultaneously. It is possible to exploit and to quantify the behaviour of dynamic (non linear and time varying) information present in these EMG signals and to design discrete characteristic vectors. These discrete characteristic vectors can perform some relevant features that, may be, lead to high and accurate classification rates of these different movement classes. The basic spectral parameters, momentary power and momentary frequency, are used as features [26]. The extracted features are: 1) central frequency '$Cent.freq$', 2) standard deviation '$Std.dev$' and 3) moments of frequency 'M_n'. The definition of each feature is given in equations 4.4.9, 4.4.10 and 4.4.11.

$$M_n(t) = \sum_k \omega_k^n \mid STFT(t,k) \mid, n = 0, 1, 2, 3, ... \tag{4.4.9}$$

$$Cent.freq = \frac{M_1}{M_0} \tag{4.4.10}$$

$$Std.dev = \sqrt{\frac{M_2}{M_0} - (\frac{M_1}{M_0})^2} \tag{4.4.11}$$

where: M_n : is the n^{th} moment of the frequency distribution at time t, n: order, and ω : frequency.

With two channels of measurement, 34 raw EMG signals are recorded for each movement class. The four classes, labeled 1, 2, 3 *and* 4, give 136 feature samples. The distribution of all these feature samples in two dimensional (2D) space, $channel_1$

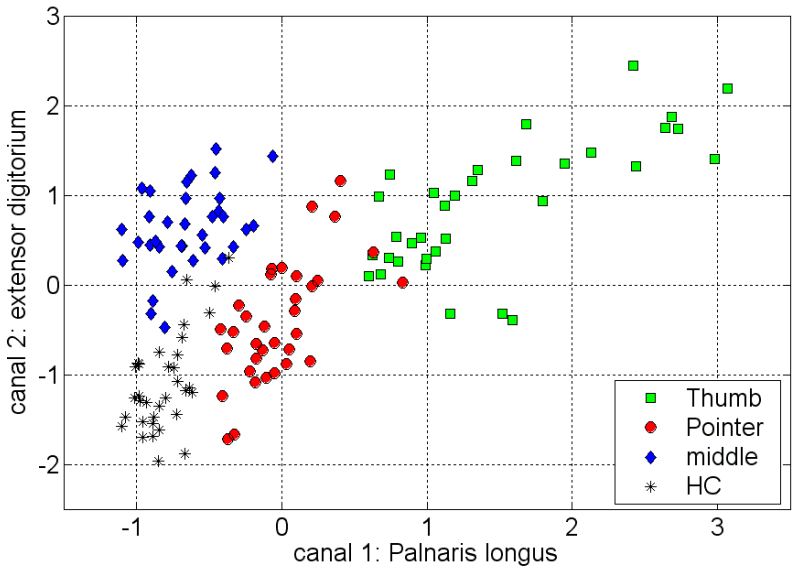

Figure 4.31: Variance-frequency feature distribution.

and $channel_2$, for these three following features: 1) variance frequency, 2) central frequency and 3) second order Moment, are shown respectively in figures 4.31, 4.32 and 4.33. Finally we reach our goal, that to find EMG signal representation, which gives best clusters discrimination between these four movement classes. From these three feature distributions, presented in figures 4.31, 4.32 and 4.33, it's possible to observe, visually, that the groups are better separated than with those extracted from the two precedent EMG signal representations: time-domain and frequency-domain representations.

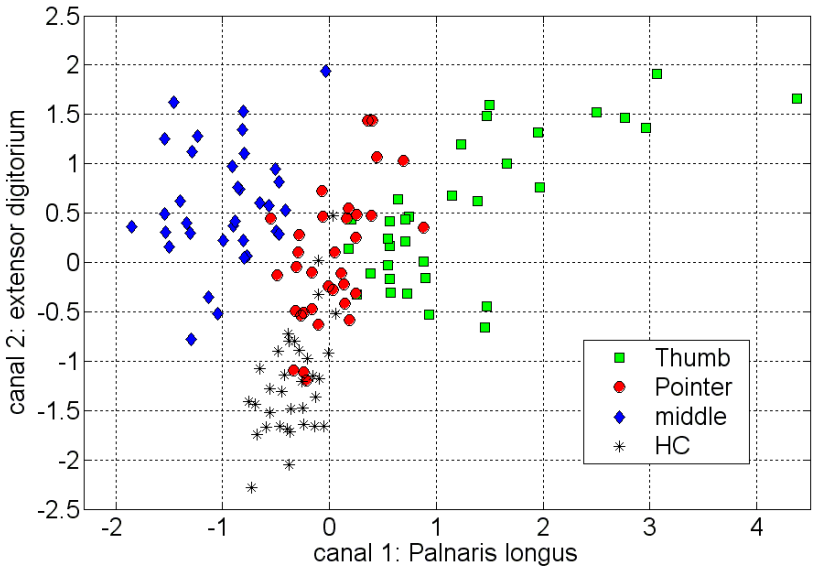

Figure 4.32: Central-frequency feature distribution

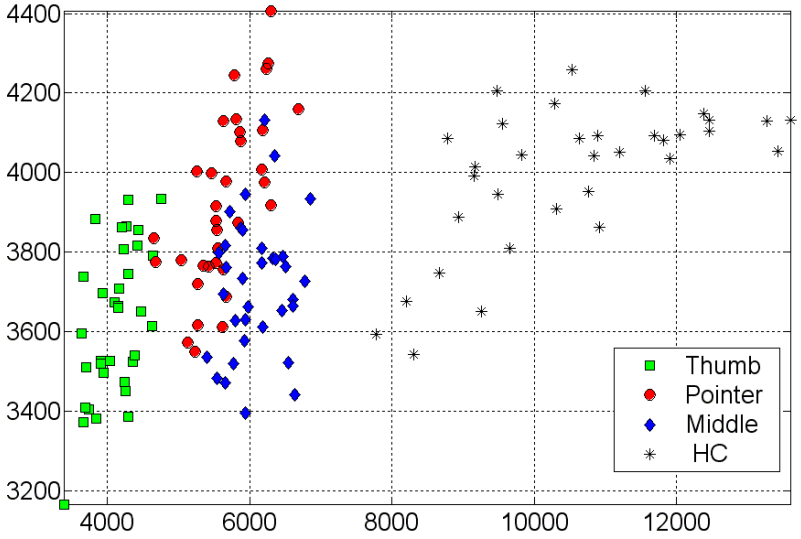

Figure 4.33: feature distribution of second order frequency-moment

.

4.4.5 Conclusion

This section is focused on mechanisms and EMG-signal analysis methods in order to produce discrete characteristics (features), which can recognise different hand and finger movement classes. These different features extracted with help of different analysis methods, are more or less discriminative. In this study three different analysis methods are used: 1) time-domain, 2) frequency-domain and 3) time-frequency-domain. The goal was to extract features corresponding to each analysis method and to compare between their clusters discrimination. Based on only visual distribution of these different feature clusters it was possible to compare the level of discrimination between them and to select the best analysis methods and consequently best features. As results it was found that the features extracted from time-frequency-domain analysis method were more discriminative than for those two other methods.

Chapter 5

Feature input space reduction

5.1 Introduction

Dimension reduction can be used for different purposes, in our case, it is used for classification problem. There are linear reduction methods and non linear reduction methods. Linear methods like Principle Components Analysis, which is used in this chapter, is more interpretable than non linear methods, which can deal with complicated structures. The role of dimensionality reduction is to simplify high-dimensional data sets to retain information, that is important for classes discrimination, and discard that which are irrelevant. Dimension reduction methods present several advantages like:

- possibility to visualise feature data in low-dimensional space.
- produce uncorrelated new features.
- allow building simple modelling or/and classification models.
- reduce space complexity.

The effective way of reducing the time-consuming of calculation is to reduce or to use as small as possible the number of feature vectors. The goal of this chapter is first to present efficient method that can transform a given data set X of dimension m to an alternative data set Y of smaller dimension p. Secondly it will be discussed if it's necessary to use always such methods for solving classification problems or there is another alternative.

5.2 Projection method

PCA is a way of expressing a high dimensional data set in an alternative set of low dimensional data set with high variability, which is used for data visualisation and clustering. In this study a linear dimension reduction technique (PCA), originally introduced by K. Pearson in 1902 see also [33], is investigated. Many problems in information processing involve some form of dimensionality reduction. Popular method of dimensionality reduction, PCA, is an eigenvector method designed to model linear variability in high dimensional data space. It considers the greatest variance, i.e. get a maximum value of the quantity: $\sum_i (x_i - x_{mean})^2$, for the eigenvectors of the data covariance matrix. This reduction is achieved by taking m vectors $X_1,\ X_2\ ,\ldots,\ X_m$ and finding combinations of them to produce principal components: $PC_1,\ PC_2\ ,\ldots,\ PC_p$, which are uncorrelated; where $p < m$. Principle Components (PCs) are ordered so that PC_1 exhibits the greatest amount of the variation, then PC_2 exhibits the second greatest amount of the variation, and so on. Once eigenvectors are found from the covariance matrix, the next step is to order them following the values' order of their eigenvalues, from the highest to lowest. This gives us the principal components in order of significance. Generally it is written in

the literature that this transformation leads to a more discriminated representation of data. In this study it will be shown that for relevant feature vectors this procedure may lead to less discriminated representation of data set distribution. It would be concluded also that there is, some times, a loss of information in using dimensional reduction of data space for relevant features.

5.3 *PCA* illustration

Following is a detailed description of PCA using the covariance method, which provides us the explanation of PCA algorithm at each step. To present this tutorial in details and graphically, an example of two dimensional space is proposed. Two EMG surface electrodes are placed on two muscle groups, palnaris longus $channel_1$ and extensor digitorum $channel_2$. The location of electrodes on the subject's arm is given in figure 5.1. Two feature vectors V_{hc1} and V_{hc2} for zero order moment ($M0$), are

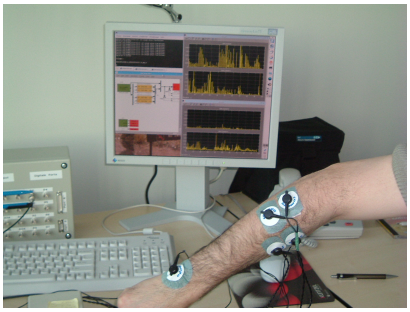

Figure 5.1: surface electrodes posisition for forearm EMG signals measurements

considered. V_{hc1} is feature Vector of 34 variables, 5.3.1, for hand closing of $channel_1$, and V_{hc2} is feature vector of 34 variables, 5.3.2, for hand closing of $channel_2$.

$$V_{hc1}^T = x_1,\ x_2,\ \ldots,\ x_{34}. \tag{5.3.1}$$

$$V_{hc2}^T = y_1,\ y_2,\ \ldots,\ y_{34}. \tag{5.3.2}$$

where T: denotes transpose.

In figure 5.2 some of 34 movements are presented as EMG activation for both channels. The distribution of feature vectors of $M0$ for raw EMG signal in 2D dimensional

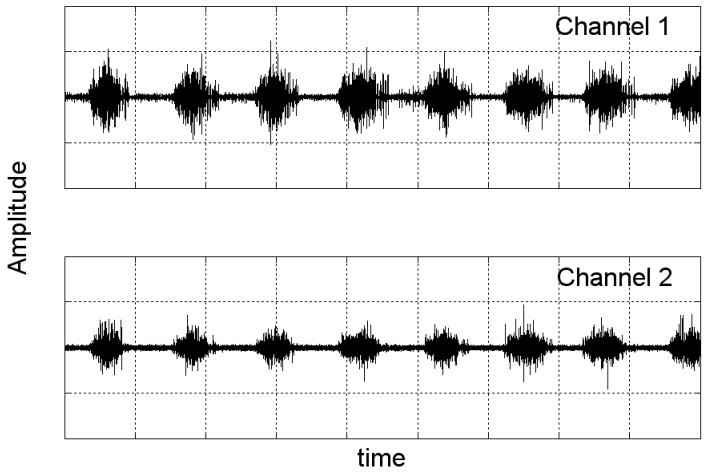

Figure 5.2: some of 34 muscles' activation of hand closing with two channels EMG signals measurements

space is presented in figure 5.3, see chapter 4 section 4.4 about the definition of this feature.

5.3.1 Algorithm's steps illustration

The steps for computing mean values, covariance matrix, eigenvectors and eigenvalues [30] require the use of a computer-based algorithms. These algorithms are available in matrix algebra systems and also in *MatLab*.

Step 1: Normalisation

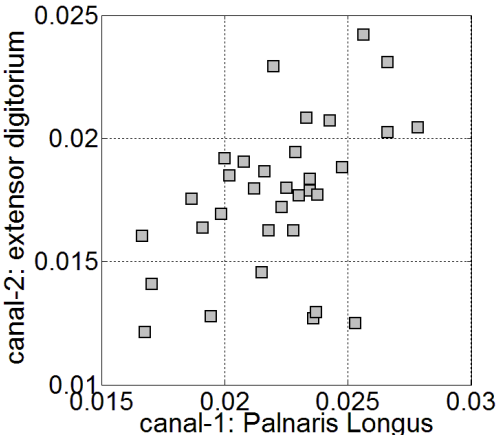

Figure 5.3: Plot of 34 feature samples ($M0$) in 2D space, V_{hc1} and V_{hc2}

Subtraction of mean value, figure 5.4, from each vector, V_{hc1} and V_{hc2}, equation 5.3.3.

$$\begin{aligned} M0_{hc1} &= V_{hc1} - mean(V_{hc1}) \\ M0_{hc2} &= V_{hc2} - mean(V_{hc2}) \end{aligned} \tag{5.3.3}$$

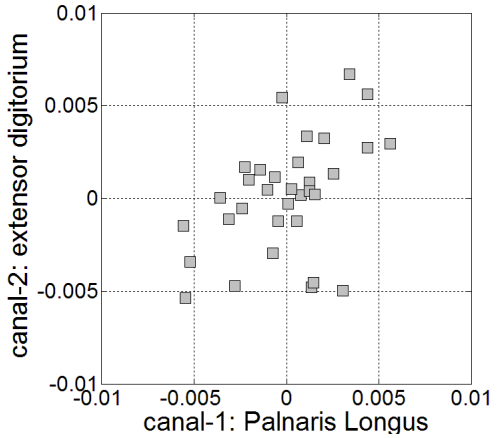

Figure 5.4: Plot of 34 normalised feature samples ($M0$) on two variables V_{hc1} and V_{hc2}

Step2: covariance matrix

The covariance measure [2] of two vectors X and Y, with means of E$\{X\}$ and E$\{Y\}$, describes the behavior of their co-variability, and is given by equation 5.3.4.

$$Cov(X,Y) = E\{[X - E\{X\}][Y - E\{Y\}]\}. \tag{5.3.4}$$

Covariance calculation is the multiplication of differences between the vectors $X = x_1,\ x_2,\ \ldots\ x_n$ and $Y = y_1,\ y_2,\ \ldots\ , y_n$, and their mean values. The value of the covariance is interpreted as follows:

A positive covariance indicates that the two variables tend to move up and down together, however a negative covariance indicates that when one moves higher, the other tends to go lower.

In our example covariance matrix is defined as, equation 5.3.5

$$cov_{matrix}(M0_{hc1}, M0_{hc2}) = \begin{pmatrix} cov(M0_{hc1}, M0_{hc1}) & cov(M0_{hc1}, M0_{hc2}) \\ cov(M0_{hc2}, M0_{hc1}) & cov(M0_{hc2}, M0_{hc2}) \end{pmatrix} \tag{5.3.5}$$

Using Matlab command "*cov*" we get the following covariance matrix values, equation 5.3.6, of the data presented in figure 5.4

$$cov_{matrix}(M0_{hc1}, M0_{hc2}) = 1.0e^{-005} * \begin{pmatrix} 0.7648 & 0.3700 \\ 0.3700 & 0.9944 \end{pmatrix}. \tag{5.3.6}$$

Step 3: calculation of eigenvectors and eigenvalues of covariance matrix

Since covariance matrix is square, it's possible to calculate the eigenvectors and eigenvalues for this matrix. Some properties of eigenvectors are described as :

- only the square matrices have Eigenvectors, and not every square matrix has eigenvectors.
- all the eigenvectors of a matrix are perpendicular (orthogonal) i.e. at right angles to each other, and it doesn't depend on the dimension of this matrix.

- some matrices do not have an eigenvector decomposition, These matrices are defective, or not diagonalizable. A Matrix M is diagonalisable if it is a square matrix and there is an invertible matrix INV such that $INV^{-1}.M.INV$ is a diagonal matrix.

"*pcacov*" is a *matlab* command, which computes the eigenvectors matrix ($Eigenvectors_{matrix}$), equation 5.3.7 and the eigenvalues matrix ($Eigenvalues_{matrix}$), or PC variances (variances of Principle Components), equation 5.3.8. $[Eigenvectors_{matrix}, Eigenvalues_{matrix}] = pcacov(cov_{matrix})$

$$Eigenvectors_{matrix} = \begin{pmatrix} -0.5931 & -0.8051 \\ -0.8051 & 0.5931 \end{pmatrix}. \tag{5.3.7}$$

$$Eigenvalues_{matrix} = 1.0e^{-004} * \begin{pmatrix} 0.1267 & 0 \\ 0 & 0.0492 \end{pmatrix}. \tag{5.3.8}$$

Eigenvectors matrix of dimension 2×2, contains 2 column vectors, each of length 2, which represent the 2 eigenvectors of the covariance matrix cov_{matrix}. Note that both of these eigenvectors ($Eigenvector_1$ and $Eigenvector_2$) have been scaled to a unit i.e. their module equal to 1; $\sqrt{0.59312^2 + 0.80512^2} = 1$. The name "*Eigenvector*" is derived from the German word "eigen", and was first used in this context by Hilbert in 1904, it means "proper" or "own". The Eigenvalues matrix ($Eigenvalues_{matrix}$) takes also the form of an 2×2 diagonal matrix, where the first value ($0.1267 * 1e^{-4}$) is bigger than the second one ($0.0492 * 1e^{-4}$). That means the first principle component ($eigenvector_1$) presents more data variability than the second principle component ($eigenvector_2$).

Mean values and covariance matrix are calculated from the data, however Eigenvectors and eigenvalues are calculated from the covariance matrix. The directions of

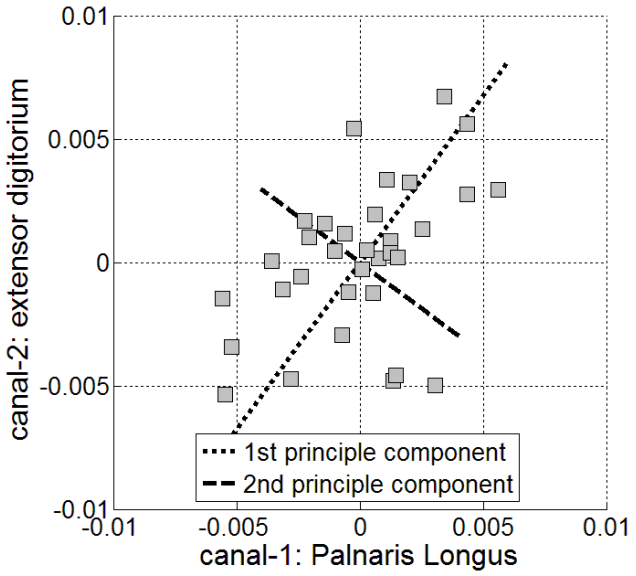

Figure 5.5: Plot of first and second Eigenvectors

eigenvectors are drawn in figure 5.5 as dashed and doted lines. The first eigenvector, which has the largest eigenvalue points to the direction of largest variance of variables, figure 5.5, in doted red line, whereas the second eigenvector, which is orthogonal to the first one points to the direction of less variables variance, figure 5.5, in dashed green line. In this example the first eigenvalue, $Eigenvalue_1 = 0.1267$, in equation 5.3.8 corresponding to the first eigenvector, $Eigenvector_1 = \binom{-0.5931}{-0.8051}$, in equation 5.3.7. While the second eigenvalue, $Eigenvalue_2 = 0.0492$, in equation 5.3.8 corresponding to the second eigenvector, $Eigenvector_2 = \binom{-0.8051}{0.5931}$, in equation 5.3.7. By comparing the values of eigenvalues we can say that the first eigenvector presents more variability. The data could be well approximated with a 1D dimensional, $Eigenvector_1 = \binom{-0.5931}{-0.8051}$, representation. The eigenvalues and eigenvectors found above should satisfy the equation 5.3.9

$$cov_{matrix} \cdot \quad Eigenvectors_{matrix} = Eigenvalues_{matrix} \quad Eigenvectors_{matrix} \tag{5.3.9}$$

Verification:

For the first principle component, the value of the left term of equation 5.3.9 is given in equation 5.3.10, however the value of the right term of equation 5.3.9 is given in equation 5.3.11.

$$\begin{aligned} cov_{matrix} \cdot \quad & Eigenvector_1 = \\ 1.0e^{-005} * \begin{pmatrix} 0.7648 & 0.3700 \\ 0.3700 & 0.9944 \end{pmatrix} \quad & \begin{pmatrix} -0.5931 \\ -0.8051 \end{pmatrix} = \\ & 1.0e^{-004} * \begin{pmatrix} -0.0751 \\ -0.1020 \end{pmatrix} \end{aligned} \tag{5.3.10}$$

$$\begin{aligned} Eigenvalue_1 \quad & \cdot Eigenvector_1 = \\ 1.0e^{-004} * 0.1267 \quad & \begin{pmatrix} -0.5931 \\ -0.8051 \end{pmatrix} = \\ & 1.0e^{-004} * \begin{pmatrix} -0.0751 \\ -0.1020 \end{pmatrix} \end{aligned} \tag{5.3.11}$$

Step 4: Projection of original data onto Eigenvectors

The original data set, $M0_{hc} = [M0_{hc1}, M0_{hc2}]$, is projected onto these two found orthogonal eigenvectors (step 3) ($Eigenvector_1$, and $Eigenvector_2$) figure 5.6. . This

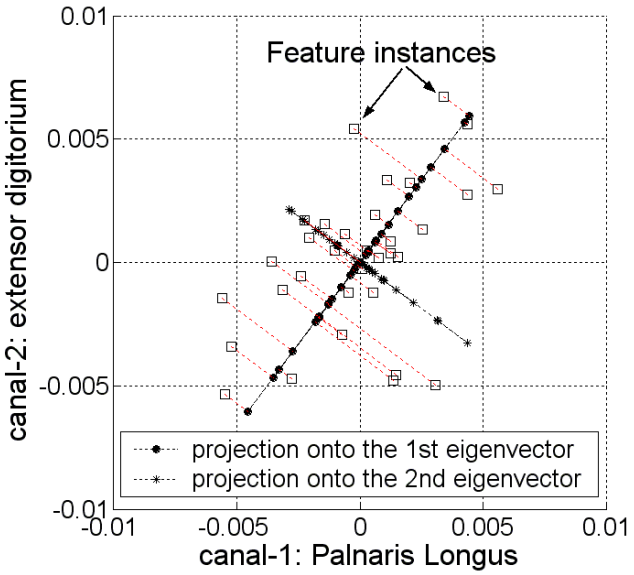

Figure 5.6: projected data samples onto first and second eigenvectors

projection of data samples onto these new axis leads to creation of two principal components PC_1 and PC_2, figure 5.6. We can observe that data samples (variables) of first principal component (PC_1) presented with circles are good distinguishable, however the images of variables of second principal component (PC_2) presented with stars, are less distinguishable.

5.3.2 Graphical determination of eigenvectors

For several directions of these two orthogonal eigenvectors is possible to get after data projection different principal components according to their Eigenvectors' directions. These new reduced data set have different variabilities, more the variability is big less

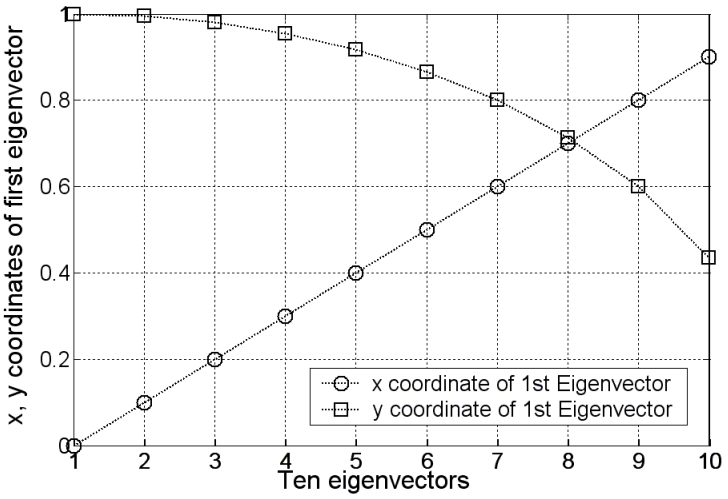

Figure 5.7: The plot of ten (x, y) coordinates corresponding to ten unit eigenvectors

we have loss of information due to the transformation function between original data space and new reduced data space. This variability can be measured with variance function. For ten different directions of unit projection eigenvectors, some values between 0 and 0.9 of x abscise coordinates are given, circles in figure 5.7, and its correspondent y coordinates values are calculated, squares in figure 5.7. Correspondent projected data variables, for these ten unit projection vectors, are also calculated and presented in figure 5.8. Between these ten produced components, through data projection, the component with higher variability (or maximum variance) will be considered as the principle component (PC). The curve that connect all precedent calculated eigenvalues is presented in figure 5.9 . These Eigenvalues are presented as eigenvector coordinates (x, y). The intersection of the maximum variance value of variance curve with these coordinates in vertical line, figure 5.10, gives us the coordinates of the best Eigenvector which is considered as principal component.

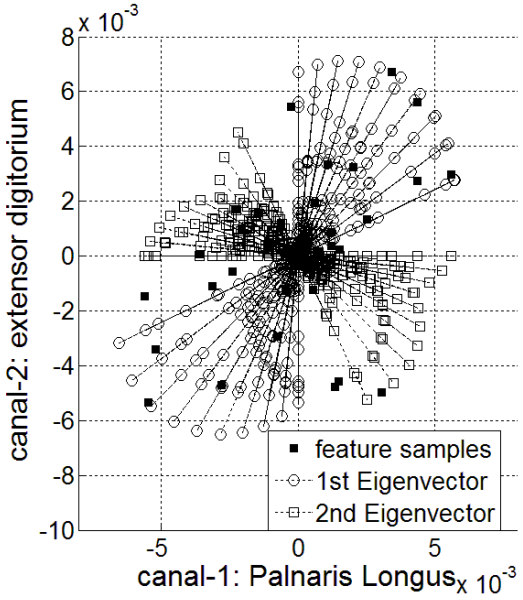

Figure 5.8: Rotation of eigenvector inside 2D data space in ten directions and their corresponding Principal Component vectors

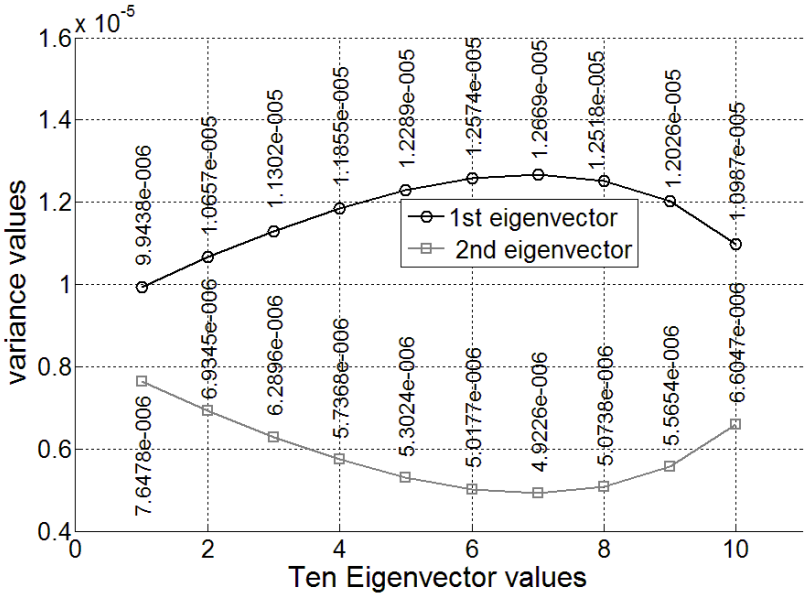

Figure 5.9: The plot of curves of Eigenvalues corresponding to two orthogonal Eigenvectors

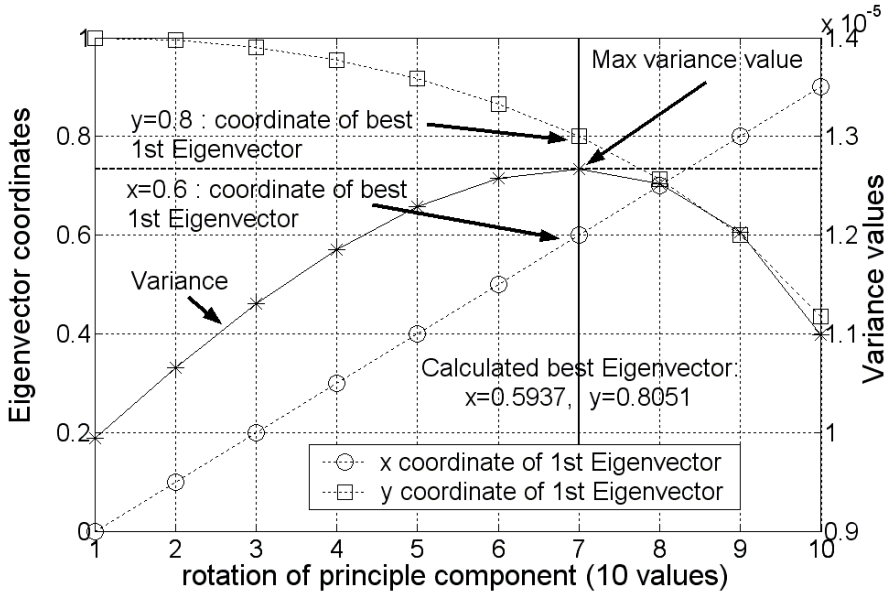

Figure 5.10: Graphical determination of first Eigenvector component

5.4 Efficiency of projection method

Surface EMG signal-based hand movement classification for the control of prostheses is considered. EMG surface electrodes are placed on two muscle groups: palmaris longus and extensor digitorum. From the input feature space, the classifier, see chapter 6, should be able to classify four hand movements: flexion of the thumb, the pointer and the middle finger movements as well as hand closing. The initial part ($400ms$) of each single contraction period is extracted from the raw signal and analysed using Short Time Fourier Transform ($STFT$), see section 4.4.4. This analysis method gives a measure of both time and frequency information. Three relevant features, moment of second order, central frequency and standard deviation see chapter 4 section 4.4.4, are extracted for each channel. Three intelligent computational methods, Radial Basis Function (RBF) networks, Multi-Layer Perceptrons (MLP)

and Learning Vector Quantization (LVQ) networks are used to distinguish between our four different types of hand movements. In first step the classification is applied on each feature separately using the above cited three methods. In the second step all three features are considered together as 6D dimensional feature space. Finally, the Principal Components Analysis (PCA) algorithm is applied to reduce this 6D dimensional feature input space to 2D dimensional input feature space.
PCA is used to reduce the dimensionality of a data set and to retain as much information as possible of the relevant information in the original multi-dimensional input feature space. In case of feature instances of big variability in multi-dimensional input space, the application of PCA algorithm may produce or lead to a loss of information which give as consequence a decrease in classification accuracy. At the end of this study all results will be compared in tables to resume the effect of PCA dimension reduction algorithm on surface EMG signal classification in case of relevant feature vectors.

5.4.1 Problem illustration

The following figure shows us three cases of feature set presentation. The classification task can use these three variants of feature presentation. The problem or the question is to look for a best variant of feature set presentation, figure 5.11, to use for classification. To answer this question, each type of these three feature set presentations for classification is studied alone and in the end we compare them for the choice of the best variant according to their results. The first feature set presentation, for classification, is based on feature selection among n features, $F1$, . . ., Fn, and we test each feature alone. This method allows us to discriminate the best features from

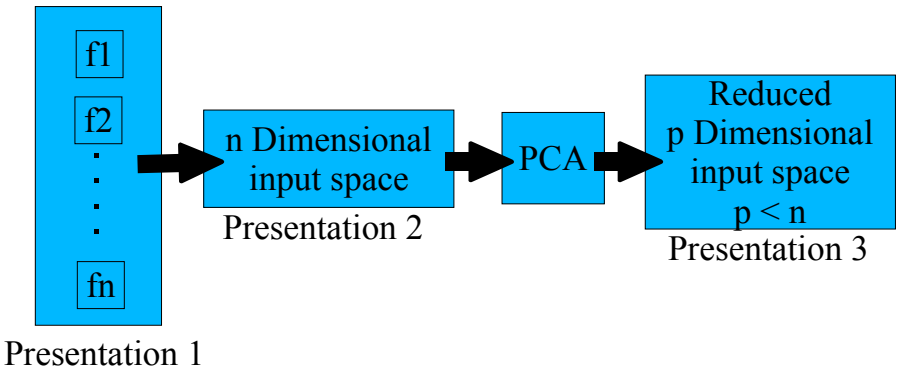

Figure 5.11: three cases of data set features presentation

another. In the second feature set presentation, all our set features are considered together and an n multi dimensional space is builded, which gives us clusters in many dimensional axis. The third and last feature set presentation procedure considers a reduced feature space through linear reduction PCA method. The question is which data set presentation should be used for classification. The answer should take in consideration not only the results but also the complexity of generated classification models.

5.4.2 Features considered separately

1) Learning Vector Quantization classifier model

LVQ networks can classify any set of input vectors, which can be linearly separable or not. The following table presents the value ranges of our four classes used by LVQ model-classifier. The architecture of this network, figure 5.12, resembles to that of unsupervised competitive learning network, except that each output is assigned to a target class and works in two steps. First it uses an unsupervised data clustering method to locate several clusters without using the information about the number

Table 5.1: Class range for each neuron output

classes	Target Output	Output range	Type of Movement
Class1	1	< 1.5	Thumb finger
Class2	2	1.5 - 2.5	Pointer finger
Class3	3	2.5 - 3.5	Middle finger
Class4	4	> 3.5	Hand closing

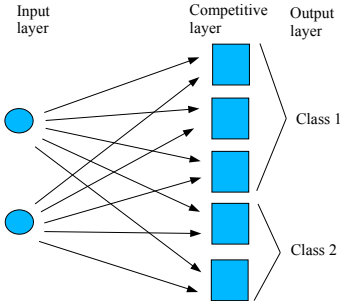

Figure 5.12: Architecture example for LVQ network

of classes. Second it optimises the cluster centres using *a priori* information about the number of classes. In this type of network, LVQ, there is only one neuron affected in the second linear layer for each output class. The number of clusters can be also, *a priori*, specified. This network is able to reduce large data sets to a smaller number of codebook vectors (cluster centres) suitable for data compressing. LVQ was introduced by T. Kohonen [36]. A new version with improved classification performances have been proposed in [38]. Since then, it has been widely used for an extensive bibliography. The "*newlvq*" function in Matlab programm creates a two layer network. This matlab function use the learning function "*learnlv2*" according to the developed learning Algorithm LVQ2 [39]. In the learning quantization method, the weight in the hidden competitive layer are refined and then the input vector is classified to its target class. This layer is partitioned into groups of neurons, each one is associated to a class.

a) Case of feature (M2):

LVQ network used in this work has been first tested for several neuron numbers composed the first competitive layer during 25 train epochs, see figure 5.13, and then

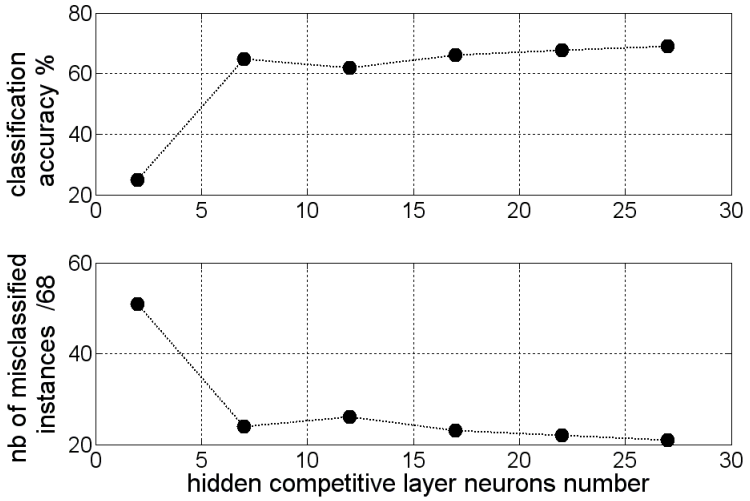

Figure 5.13: Classification accuracy and global number of misclassified instances for different number of neurons in Competitive layer. Number of training epochs = 25. Feature: M2. Method:LVQ

the optimal neurons number found, 27 neurons in this case, in competitive layer is trained again during 100 train epochs. In figure 5.14 the accuracy classification and misclassified instances are presented for each type of movement. The interferences between classes are also presented to know exactly the original class of each misclassified instance. Additionally are presented the global correct classified instances and the global misclassified instances.

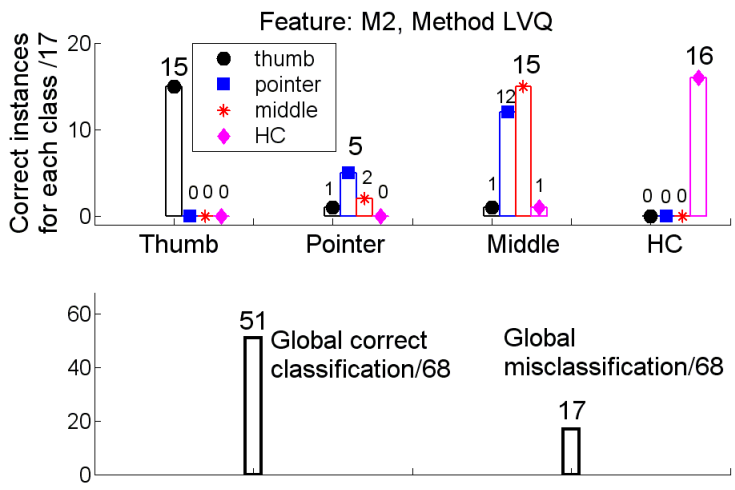

Figure 5.14: Number of misclassified instances for each class with 27 neurons and during 100 epochs. This figure shows even the numerical distribution of misclassified instances in other classes.

b) Case of feature (Fcent):

The same study is applied for the feature $Fcent$. The network LVQ has been tested for several neuron numbers composed the first competitive layer, and an optimal model is found during 25 train epochs, see figure 5.15. The best one of them, 12 neurons in competitive layer, is trained again during 100 epochs. In figure 5.16 the accuracy classification and misclassified instances are presented for each type of movement. The interferences between classes are also presented to know exactly the original class of each misclassified instance. Additionally are presented the global correct classified instances and the global misclassified instances.

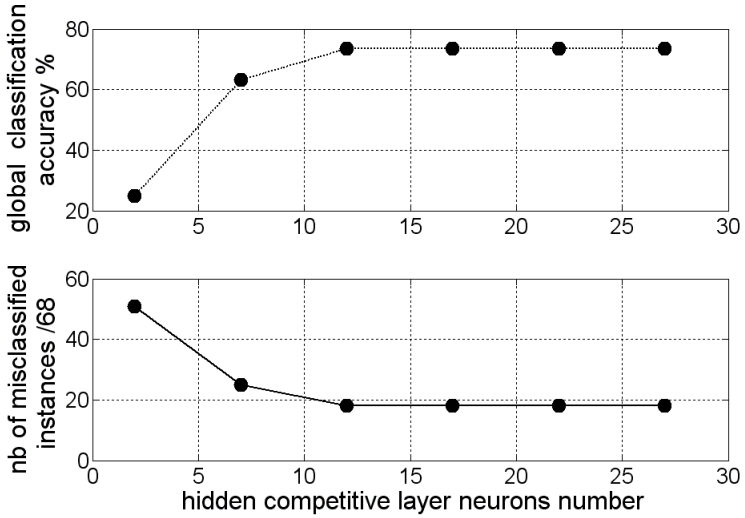

Figure 5.15: Classification accuracy and global number of misclassified instances for different number of neurons in Competitive layer. The number of training epochs = 25. Feature: *Fcent*. Method: *LVQ*

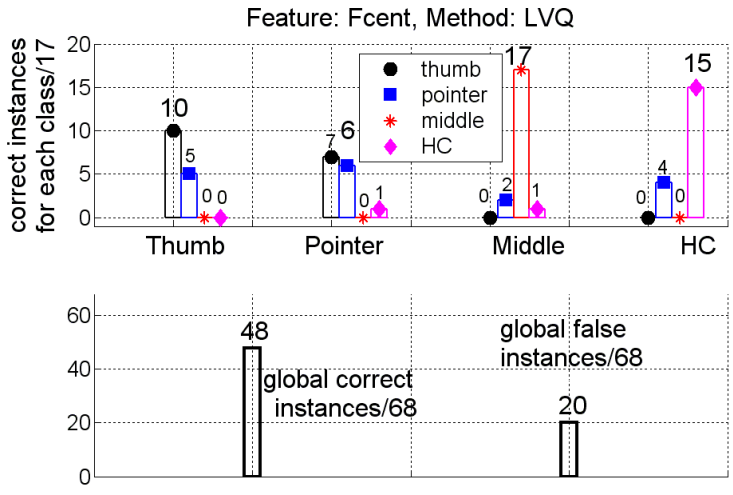

Figure 5.16: Misclassified instances with 12 neurons and during 100 epochs. This figure shows even the numerical distribution of misclassified instances in other classes.

c) Case of feature (Fstd):

The same study is applied for the feature $Fstd$. The network LVQ has been tested for several neuron numbers composed the first competitive layer during 25 train epochs, see figure 5.17. The best one of them, 22 neurons in competitive layer, is trained again during 100 epochs. In figure 5.18 the accuracy classification and misclassified

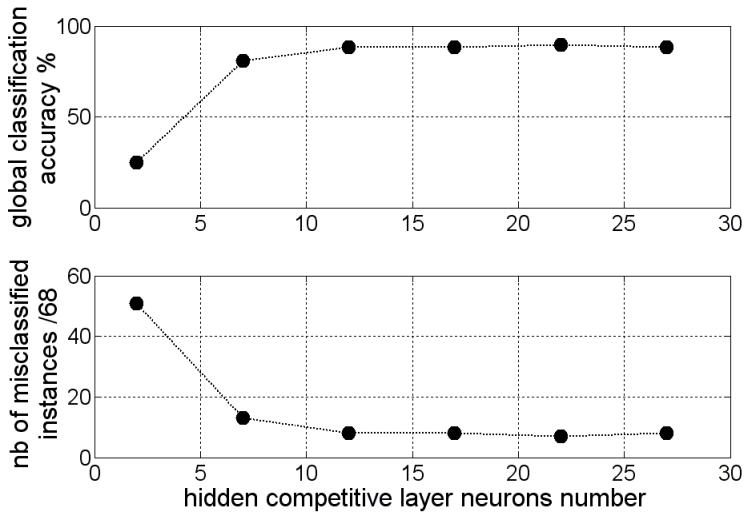

Figure 5.17: Classification accuracy and global number of misclassified instances for different number of neurons in Competitive layer. The number of training epochs = 25. Feature: Fstd. Method: LVQ

instances are presented for each type of movement. The interferences between classes are also presented to know exactly the original class of each misclassified instance. Additionally are presented the global correct classified instances and the global misclassified instances.

We resume all above results about accuracy classification and misclassified instances of the features $M2$, $Fcent$ and $Fstd$ with LVQ classification method in the following two figures 5.19 and 5.20.

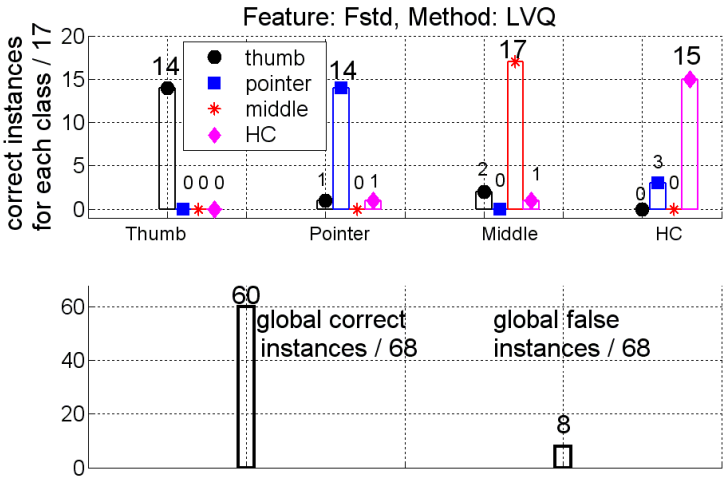

Figure 5.18: Misclassified instances with 22 neurons, and during 100 epochs. This figure shows even the numerical distribution of misclassified instances in other classes.

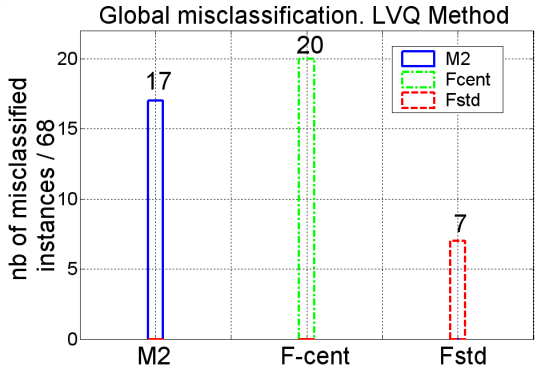

Figure 5.19: Global misclassified instances for three features: $M2$, $Fcent$ and $Fstd$ using LVQ classification method.

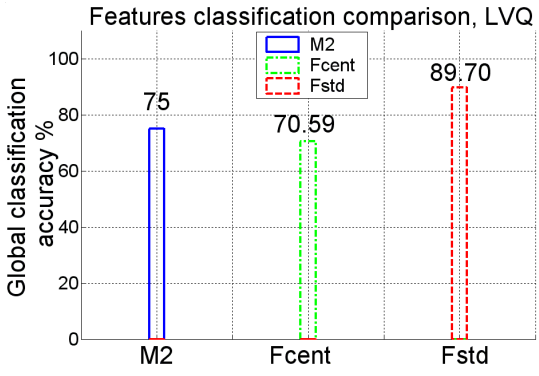

Figure 5.20: Classification accuracy corresponding to three features: $M2$, $Fcent$ and $Fstd$ using LVQ classification method.

2) Multi-Layer Perceptron networks

This network is used in many different applications. Its architecture, figure 5.21, has a large class of network types with many different topologies and training methods. MLP has been widely used for an extensive number of applications. The "*newff*" function in Matlab programm Creates a feed-forward backpropagation network. Feed-forward networks have the following characteristics. Perceptrons are arranged in layers, inputs are presented in the first layer and the last layer producing outputs. The middle layers are called hidden layers and don't have any connections with the external world. Each perceptron in one layer is connected to every perceptron on the next layer, and their information are "fed forward" from one layer to the next. The size of the input and output layers are defined according to the number of system's inputs and outputs. However the number of hidden neurons can be specified according to the needed performances and complexity of the network, for more details see section 6.2. The number of "*tansig*" neurons in the hidden layer is determined based on their

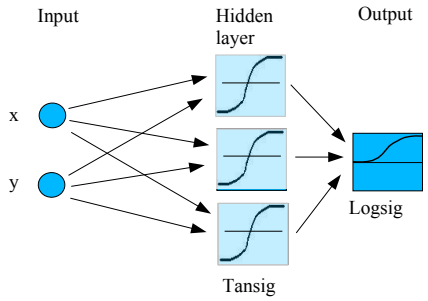

Figure 5.21: Architecture example for MLP network

performance in training process. The one-neuron output-layer log-sigmoid transfer function "*logsig*"is used, which gives an output in the range of 0 to 1. Our output range between 0 and 1 will be divided in four ranges, since we have four classes to be

Table 5.2: Class range for each neuron output

classes	Target Output	Output range	Type of Movement
Class1	0.125	0.00 - 0.25	Thumb finger
Class2	0.375	0.25 - 0.50	Pointer finger
Class3	0.625	0.50 - 0.75	middle finger
Class4	0.875	0.75 - 1.00	Hand closing

identified table 5.2.

The *MLP* network used in this work has been tested for several neuron numbers in the hidden layer during 25 epochs. The optimal one found is trained again during 100 epochs. The accuracy classification and misclassified instances are calculated for each type of movement. Additionally are calculated the average accuracy and the total number of misclassified instances. This same study is applied for each feature for these following three features: $M2$, $Fcent$ and $Fstd$. All results of classification accuracy and misclassified instances, corresponding to each feature, are resumed and presented in the following two figures 5.22 and 5.23.

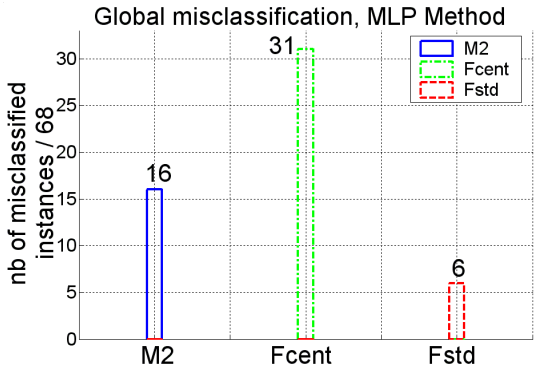

Figure 5.22: Global misclassified instances for three features : $M2$, $Fcent$ and $Fstd$ using MLP classification method.

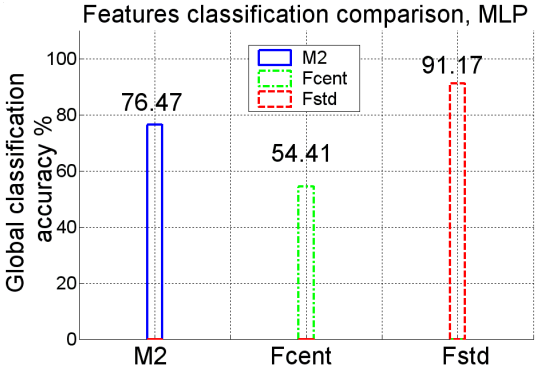

Figure 5.23: Classification accuracy corresponding to three features: $M2$, $Fcent$ and $Fstd$ using MLP classification method

3) Radial Basis Function networks

Compared to the *MLP* network, the radial basis function (*RBF*) network is the next most used network model, see section 6.5.3 for more details. This Network, figure 5.24, is a one hidden layer neural Network with several forms of radial basis activation functions, like bell Gaussian function, which is the basis function chosen for the most commonly used applications. The "*newrb*" function in Matlab programm

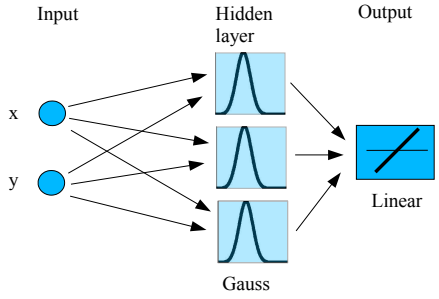

Figure 5.24: Architecture example for *RBF* network

creates a two layer network. The first layer contains neurons of radial basis functions and the second contains neurons of linear functions. This matlab function creates a new neuron for every iteration. The error of the new network is checked, and if it is not low enough the next neuron is added. This procedure is repeated until the error goal is met, or the maximum number of neurons is reached. The output layer is linear and the rate of classification is depending on the spread value of the hidden unit. We give many values between 0.1 and 1 to find the optimal spread value. These networks are trained during 25 epochs for all three features. The optimal spread value found during 25 training iterations is trained again during 100 epochs. The accuracy classification and misclassified instances are calculated for each type of movement. Additionally are calculated the average accuracy and the total number

Table 5.3: Class ranges for one linear neuron output

classes	Target Output	Output range	Type of Movement
Class1	0.125	0.00 - 0.25	Thumb finger
Class2	0.375	0.25 - 0.50	Pointer finger
Class3	0.625	0.50 - 0.75	middle finger
Class4	0.875	0.75 - 1.00	Hand closing

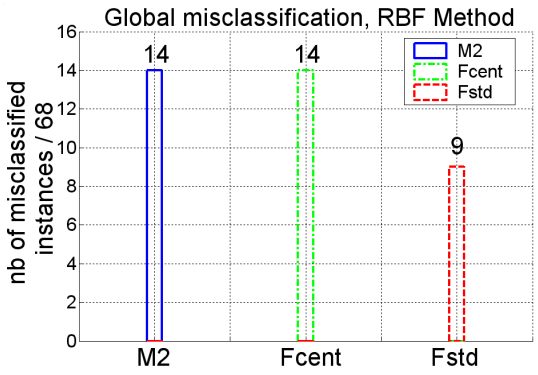

Figure 5.25: Global misclassified instances for three features : $M2, Fcent$ and $Fstd$ using RBF classification method.

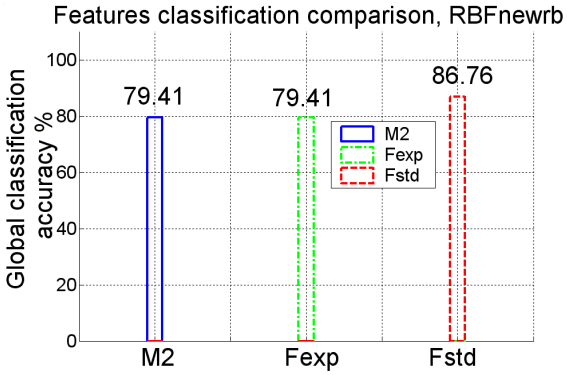

Figure 5.26: Classification accuracy corresponding to three features: $M2, Fcent$ and $Fstd$ using RBF classification method

of misclassified instances. This same study is applied for each feature for our three features: $M2$, $Fcent$ and $Fstd$. All results concerning accuracy classification and misclassified instances, corresponding to each feature are resumed and presented in figures 5.25 and 5.26.

4) Comparison of *LVQ, MLP* and *RBF* methods

Results comparison of accuracy classification and misclassified instances of these three features $M2$, $Fcent$ and $Fstd$ for all three classification methods: LVQ, MLP and RBF, are compared and presented in the following figures 5.27 and 5.28.

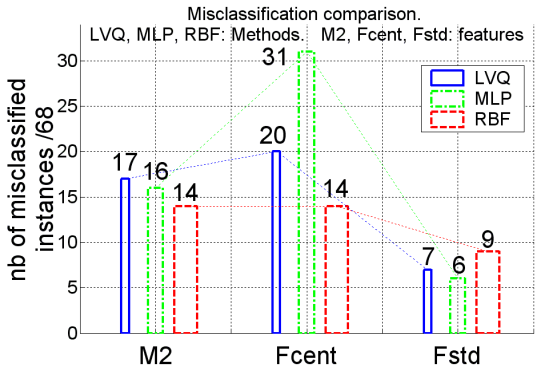

Figure 5.27: Nb of total misclassified instances for three features: $M2$, $Fcent$ and $Fstd$ using LVQ, MLP and RBF classification methods.

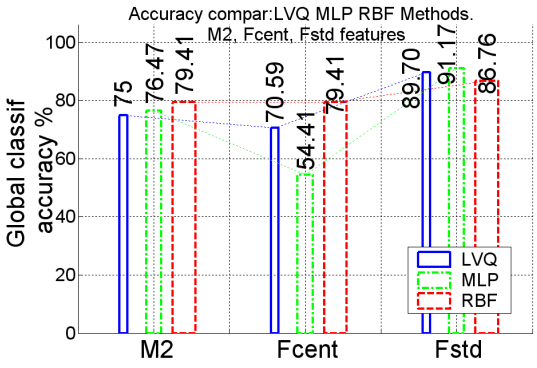

Figure 5.28: Classification accuracy comparison corresponding to three features: $M2$, $Fcent$ and $Fstd$ using LVQ, MLP and RBF classification methods

5.4.3 Features considered together in 6D input space

In the first part, section 5.4.2, the classification was applied on each feature separately using the above cited three methods LVQ, MLP and RBF. In this second part of this chapter our four movements will be classified using all these three features together in one 6D input space, figure 5.29. The discussion about these results will be done in section 5.5.

5.4.4 Feature space reduced in 2D space

Principle Component Analysis (PCA) as a projection method onto a low dimensional subspace aids to filter out the uninteresting variables. It takes in consideration the interactions and correlations between variables. In this third part of study, the input space dimension of the original data features, which is equal to six, will be reduced to two dimensional input space. This procedure allows us to present graphically our feature variables. The three networks, LVQ, MLP and RBF were trained

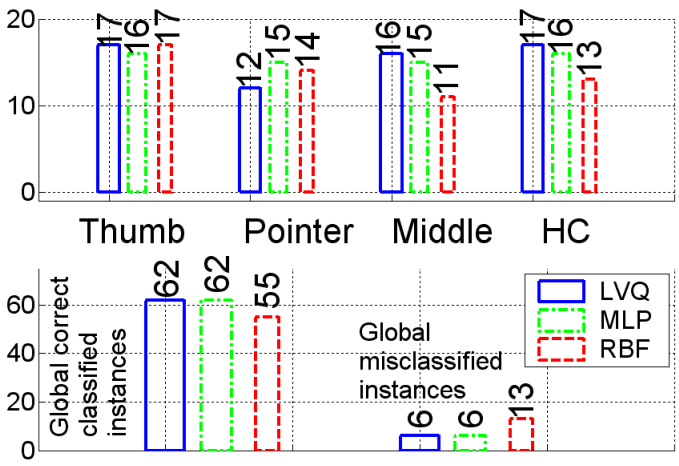

Figure 5.29: Top:Comparison of correct classified instances for each movement corresponding to six features together. Bottom: Comparison of total correct- and mis-classified instances corresponding to six features together. In both figures LVQ, MLP and RBF classification methods are used.

first during 25 epochs for these two obtained feature vectors using linear dimension reduction algorithm PCA. During these 25 training epochs we look for the optimal parameters for each method: spread value, number of hidden neurons and number of neurons in competitive layer for respectively RBF, MLP and LVQ classification methods. We train again our networks with these optimal parameters in 2D feature vectors space during 100 epochs. The correct classified instances for each type of movement and the total correct classified and misclassified instances for all movements are calculated. All these results are presented in figure 5.30.

5.5 Results discussion

First the results of section 5.4.3 in comparison with section 5.4.4 are discussed, see figures 5.31 and 5.32. Second it will be considered the comparison of results between

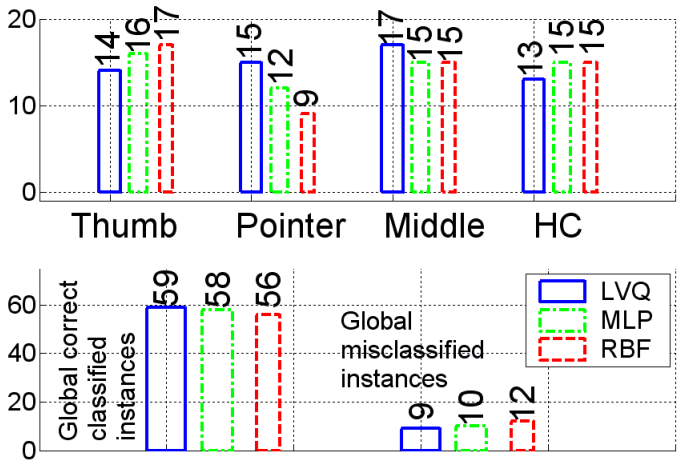

Figure 5.30: Top: Comparison of correct classified instances for each movement corresponding to 2D reduced feature space with PCA for each movement using LVQ, MLP and RBF methods. Bottom: Comparison of total correct- and misclassified instances number corresponding to 2D reduced feature space with PCA: using LVQ, MLP and RBF classification methods.

section 5.4.2, section 5.4.3, and section 5.4.4, see figure 5.33.

a) First discussion:

The global number of misclassified instances, for each classification method, in the case of 6D dimensional feature space, figure 5.31, is just little less than in the case of 2D dimensional feature space, figure 5.32, obtained after application of PCA reduction method. With LVQ classification method there are three misclassified instances more in case of 2D reduced space than in 6D space. With MLP classification method there are 4 misclassified instances more in case of 2D reduced space than in 6D space. Only with RBF method we observe amelioration by one instance in case of 2D reduced space than in 6D space. These detailed results are resumed in table 5.4.

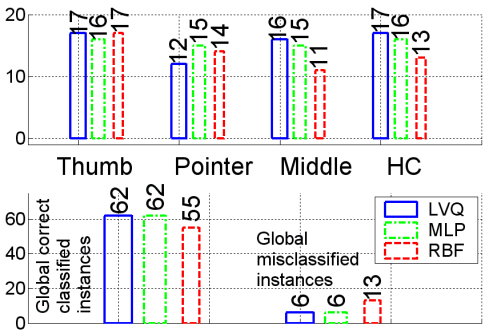

Figure 5.31: Case of section 5.4.3; Top: Comparison of correct classified instances number for each movement corresponding to 6D feature space. Bottom:Global number of misclassified instances corresponding to 6D feature space. Using LVQ, MLP and RBF classification methods for both cases.

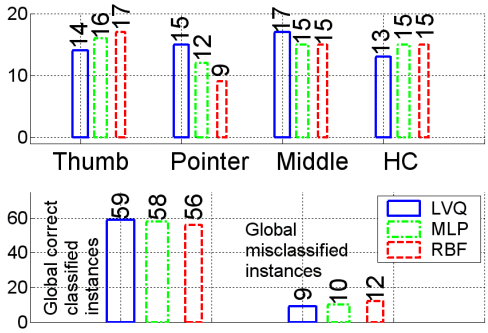

Figure 5.32: Case of section 5.4.4; Top:Comparison of correct classified instances number for each movement corresponding to 2D reduced feature space with PCA. Bottom:Total number of misclassified instances corresponding to 2D reduced feature space of 6D feature space with PCA. Using LVQ, MLP and RBF methods

Table 5.4: global number of misclassified instances comparison between 6D features space and reduced 2D features space for LVQ, MLP and RBF classification methods

Methods	6D space	2D reduced space
LVQ	6	9
MLP	6	10
RBF	13	12

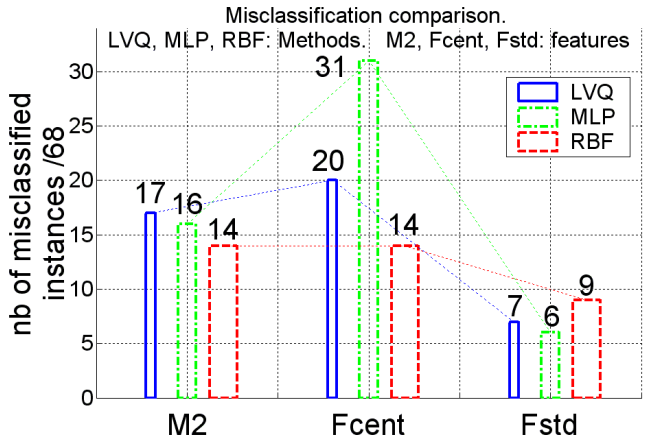

Figure 5.33: Case of section 5.4.2; Global number of misclassified instances for each feature: $M2$, $Fcent$ and $Fstd$ using LVQ, MLP and RBF classification methods.

b) Second discussion:

Now the classification results using LVQ, MLP and RBF methods for each feature separately, which is done in section 5.4.2, is considered. These results, figure 5.33 show us how relevant are these features. The third feature, $Fstd$, alone in 2D dimensional space ($Fstd\ channel_1$ and $Fstd\ channel_2$) gives less average number of misclassified instances with these three methods, which equal to 7,33 than in the case of 2D dimensional reduced features space, which gives an average number of misclassified instances equal to 10,33. In table 5.5 all results found in sections 5.4.2, 5.4.3 and 5.4.4 are resumed and compared.

5.6 Conclusion

Generally the Principal Component Analysis method leads to a more discriminated representation of data. The spatial reduction, in this study, for classification issue

Table 5.5: global number of misclassified instances comparison between 6D features space and reduced 2D features space for LVQ, MLP and RBF classification methods

Methods	6D space	2D reduced space	2D separated three features space
LVQ	6	9	$M2$=17 $Fcent$=20, $Fstd$=7
MLP	6	10	$M2$=16 $Fcent$=31, $Fstd$=6
RBF	13	12	$M2$=14 $Fcent$=14, $Fstd$=9
Average Nb	8,33	10,33	$M2$:15,66 $Fcent$=21,66 $Fstd$=7,33

using PCA, has not lead to a better results. Hence for relevant features, this procedure of space reduction may decrease this discrimination between data groups. In this study the feature $Fstd$ in 2Dimensional feature space gives by it self an average number of misclassified instances, which is equal to 7,33, less than in the case of 2Dimensional reduced space, which gives an average number of misclassified instances equal to 10,33. Therefor in this thesis and for all following studies the features are considered separately.

Chapter 6

Performances of proposed FTMC algorithm

6.1 Introduction

The goal of recognition (classification) procedure is to build a set of models that can correctly predict the right classes for different objects (features). This procedure is performed on the basis of extracted features through builded classification models. Classification models belong to two categories. First are supervised models, like Multi-Layer Perceptron, Radial Basis Networks, and Learning Vector Quantization network. Second are unsupervised models, like Self Organizing Map, Fuzzy Subtractive Clustering and Competitive Layer. There is a large set of neural networks and fuzzy logic methods in the literature addressing classification problems. These methods in case of supervised learning employ optimisation techniques to process the inputs and compare their resulting outputs against the desired outputs. Errors are then calculated, causing the system to adjust its parameters. In case of unsupervised learning, training algorithms attempt to locate clusters in the input data, which approximate the distribution of the data, without *a prior* knowledge. More

details about clustering data in [42]. *Hykin* published a comprehensive foundation for the study of Neural Networks [27], *J. -S. R. Jang*, *C. -T. Sun* and *E. Mizutani*, published a book about a foundation of Neuro-Fuzzy systems, [31]. All these methods have functions parameters and variables to be optimised.

The objective in this chapter is first to describe briefly the known classification procedures, based on Fuzzy logic and Neural Network methods, moreover a proposed Fuzzy Trimmed Mean Classification ($FTMC$) algorithm will be described. Second objective is to compare between these different intelligent computational methods: Multi-Layer Perceptron (MLP), Radial Basis Function Networks (RBF), Learning Vector Quantization network (LVQ), proposed Fuzzy Trimmed Mean Classification ($FTMC$) algorithm, and fuzzy Subtractive Clustering (FSC).

6.2 Neural Network Systems

This section describes briefly the architectures and learning procedures of adaptive neural networks (ANN). Various ANN-based models [9] were applied to identify different systems. There are many functions and variables, which should be identified for each neural network model. Neural networks process information in a similar way the human brain does. It is a network structure consisting of a large number of interconnected processing neurones, through directional links, working in parallel to solve a specific problem. Each neurone has an adaptive interaction with another. The learning methods specify how these interactions should be updated to minimise error measure between the desired and computed output data sets. There are many learning methods to build neural models. Generally an adaptive network is heterogeneous and each neuron has a specific function different from the others. It is possible to classify

these model learning methods and architectures.

6.2.1 Architecture

A simple architecture of an adaptive neural network is expressed in this following mathematical function with parameter p and variables $(x,\ y)$, and presented in figure 6.1.

Mathematical function: $y = f(x,\ p)$

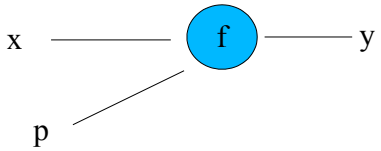

Figure 6.1: Simple perceptron with one output unit.

.

An adaptive network with only one neuron is shown in figure 6.1, which has the function f, where x and y are the input and output sets, and p is a parameter to be optimised.

6.2.2 Example of illustration

We want to build an adaptive network, which can give the result of a simple subtractive function f between two real numbers x_1 *and* x_2.

This function is given as: $y = f(x,\ p)$. defined in $\Re^2 \rightarrow \Re$

The variables are:

$x = [x_1,\ x_2]$, $p = [w_1,\ w_2]$ and y

The subtractive function is defined as:

$y = w_1 \times x_1 + w_2 \times x_2$

Our task is to find the values of w_1 and w_2, which can produce a correct subtraction output result for all inputs.

For this example we test all the values of w_1 and w_2 between -2 and 2. The errors corresponding to each couple of (w_1 and w_2) are calculated for some example values of x_1, and x_2. It is obvious that the values: $w_1 = 1$, and $w_2 =$ -1 are the right values to achieve the function of subtraction in the following operation : $y = x_1 - x_2$, but we will show how proceeds an adaptive neural network to find this solution.

1) Learning procedure to find optimised w_1 and w_2

The question now is how does work the Neural Networks to find, with learning procedure, these two parameter values of w_1 and w_2. The adaptive procedure to find w_1

Table 6.1: Some example values of input and their output results for the function of subtraction f

Input 1 x_1	Input 2 x_2	Output $y = w_1x_1 + w_2x_2$
1	2	-1
12	5	7
-4	6	-10
-2	-5	3

and w_2 should use only some values of input-output sets as training data and then we hope that these found w_1 and w_2 values will give the correct results of subtraction for all other input values of x_1 *and* x_2. In the table 6.1 some train values of x_1 *and* x_2 are given with their corresponding outputs y.

In this classical example it will be shown the simplest intelligent self-learning model

(single perceptron) that can adapt its parameters to achieve (or give)the desired output. The architecture of this neural network, is shown in figure 6.2, .

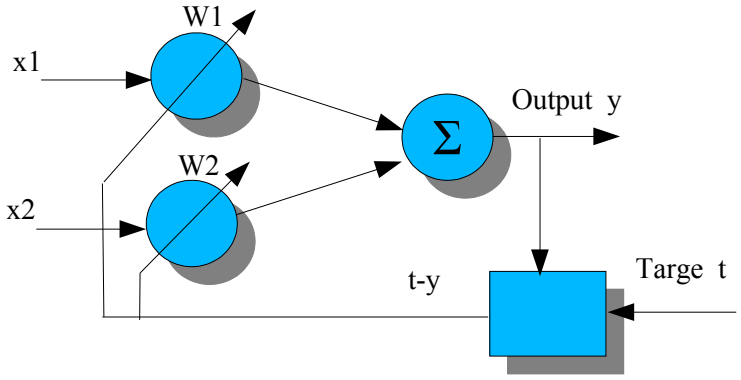

Figure 6.2: Architecture of Simple neural network *NN*.

The output y of a network is a weighted linear combination of inputs x_1 *and* x_2, equation 6.2.1.

$$y = \sum_{i=1,2} w_i x_i \tag{6.2.1}$$

The measure error is given in equation 6.2.2.

$$E = (t - y)^2 \tag{6.2.2}$$

where t design a target output.

2) Derivative-based optimisation method

The objective function E is defined on a 2D dimensional input space w. This input space is given in equation 6.2.3.

$$w = [w_1,\ w_2]^T \tag{6.2.3}$$

The aim is to find the vector $w = w^*$ which minimizes our objective function $E(w)$. The gradient of differentiable function E is a vector of first derivatives of E, denoted

as g in equation 6.2.4.

$$g(w) = [\partial E(w)/\partial w_1,\ \partial E(w)/\partial w_2]^T \tag{6.2.4}$$

The perceptron in figure 6.2 is treated as a single layer perceptron. Starting with a set of initial weights (w_0). Learning algorithm for this single-layer perceptron is achieved in these two following steps until the convergence of weights to the acceptable values is reached. These optimised weights allow the network to give a right target output.

1. select an input vector $x = [x_1,\ x_2]$

2. adapt the weights w_1 and w_2 if the output y is false.

The well known steepest formula that used to adapt the weights w_1 and w_2 is defined in equation 6.2.5.

$$w(t+1) = w(t) - \eta g(w) \tag{6.2.5}$$

where η is the learning rate.

The given training data set has 4 pairs, and the overall error measure is defined as the following squared sum in equation 6.2.6.

$$E = \sum_{p=1,4} (E_p)^2,\ E_p = t_p - y_p \tag{6.2.6}$$

where p: index number of training data pairs, t_p: desired output, and y_p: predicted output.

The derivative of the error with respect to each weight are given in equations 6.2.7, 6.2.8, and 6.2.9.

$$\partial E(w)/\partial w_i = -2 \sum_{p=1,4} (t_p - y_p) x_{ip} \tag{6.2.7}$$

$$\partial E(w)/\partial w_1 = -2 \sum_{p=1,4} (t_p - w_1 x_{1p} - w_2 x_{2p}) x_{1p}. \tag{6.2.8}$$

$$\partial E(w)/\partial w_2 = -2 \sum_{p=1,4} (t_p - w_1 x_{1p} - w_2 x_{2p}) x_{2p}. \tag{6.2.9}$$

If the initial values of w_1 and w_2 are -1.8 and 1.8, the steepest descent minimisation algorithm gives, after 30 iterations with $\eta = 0.01$, the following learning curve convergence, figures 6.3 and 6.4. This example illustrates how the algorithm follows the gradient down, on the error-surface, to reach the optimised values, which are equal to $(w_1 = 1, \ and \ w_2 = -1)$.

This example represents the simplest neural network (NN) model. It provides

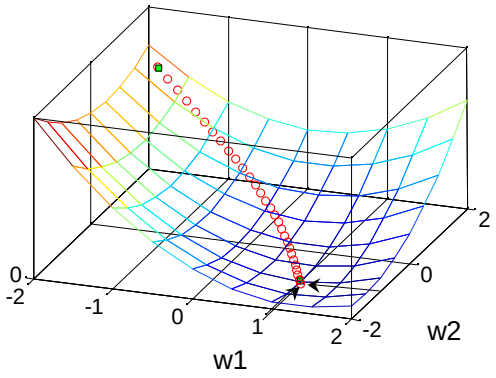

Figure 6.3: Steepest descent convergence (3D).

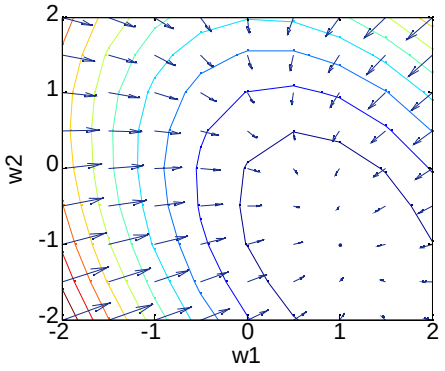

Figure 6.4: Steepest descent convergence (2D).

the ground for other complex NN models like Multi-Layer-Perceptron (MLP). The neurons of backpropagation Multilayer Perceptrons (MLP) are a composite of the weighted sum and a differentiable nonlinear activation function (transfer function).The most used activation functions in back-propagation MLP are:

- Logistic function: $f(x) = \frac{1}{1+e^{-x}}$.
- Hyperbolic tangent function: $f(x) = \frac{1-e^{-x}}{1+e^{-x}}$.
- Identity function: $f(x) = x$.

The gradient vector in this type of *NN* models is calculated in the opposite direction to the flow of the output of each neuron. Once the gradient is obtained , it is possible to apply many methods of derivative-based optimisation, which will be described in the following section 6.2.3.

6.2.3 Gradient-based optimisation methods

This section gives an introduction to nonlinear optimisation techniques using derivative-based method such as gradient-based optimisation algorithm. Optimisation is the process to find the values of a vector Θ that minimises a given function f defined on an *n-dimensional* input space Θ. Many algorithms for minimising $f(\Theta)$ are derived from algorithms, which are used for solving the following equation 6.2.10.

$$g(\Theta) = \partial f / \partial \Theta = 0. \tag{6.2.10}$$

g is the gradient of a differentiable function f: $\Re^n \rightarrow \Re$

It is often useful to use algorithms, which proceed iteratively starting from an approximate trial solution and then will gradually refine the values of parameters of vector Θ until the predetermined precision is reached. The next point Θ_{next}, in iterative descent procedure, is determined by a learning rate (step-down) from the current value Θ_{now} in the direction vector dv, equation 6.2.11

$$\Theta_{next} = \Theta_{now} + \rho dv. \tag{6.2.11}$$

where ρ : represent a learning rate.

dv (**d**escent **v**ector): is a vector of descent direction that moves us closer towards a local minimum Θ^* of our objective function $f : \Re^n \rightarrow \Re$.

Descent methods calculate the term dv in two steps: first step is determination of

the direction vector dv, and in the second step is calculation of learning rate ρ. The values of next vector Θ_{next} should satisfy the inequality in equation 6.2.12.

$$f(\Theta_{now}) > f(\Theta_{now}) + \rho dv. \tag{6.2.12}$$

ρ must be chosen so that we don't take too big or too small value of step, and the value of the step size is allowed to be changed at every iteration.
All descent algorithms look for to reach the minimum point following the line determined by the current value of Θ_{now} and the direction dv. Every descent algorithm has its own method to choose the way of successive directions.

1) Mathematical description

The aim of these methods for a given objective function f is to reduce the value of this function after every iteration. The first derivatives of a differentiable function f at the values of vector parameters Θ represent the gradient vector g defined in equation 6.2.13.

$$g(\Theta) = \partial f(\Theta)/\partial(\Theta) = \nabla f. \tag{6.2.13}$$

Gradient based optimisation strategies search iteratively the minimum of the objective function f, and it proceeds in the three following steps:

- compute the objective function $f(\Theta_{now})$ using the initialisation vector values $\Theta_{initial}$
- compute the vector direction dv and a step width ρ.
- compute of the new point $\Theta_{next} = \Theta_{initial} + \rho dv$; and go to step 1 until to reach the optimum Θ^*

The linear approximation (first order) of the objective function, f, is given in equation 6.2.14.

$$f(\Theta_{now} + \rho\ dv) \approx f(\Theta_{now}) + \rho\ g^T dv. \tag{6.2.14}$$

The condition of error convergence is that "$g^T dv$" must be negative, hence we will have $f(\Theta_{now} + \rho dv) < f(\Theta_{now})$, this condition is described in equation 6.2.12.
Various methods are available to compute descent directions, like: gradient descent, conjugate gradient method, Newton's method and *Levenberg-Marquardt* method. One condition must be hold in use of these methods: the angle between the gradient vector g and descent vector direction dv, equation 6.2.15, must be in the area of 90° and 270°, see figure 6.5. The gradient directions are always perpendicular to the contour curves.

$$\Theta_{next} = \Theta_{now} + \rho\ dv\ g = \Theta_{now} + \rho\ \parallel g \parallel\ \parallel dv \parallel\ \cos\left(angle(g, dv)\right) \tag{6.2.15}$$

The directions from the starting point Θ_{now}, figure 6.5, which shows the surface error

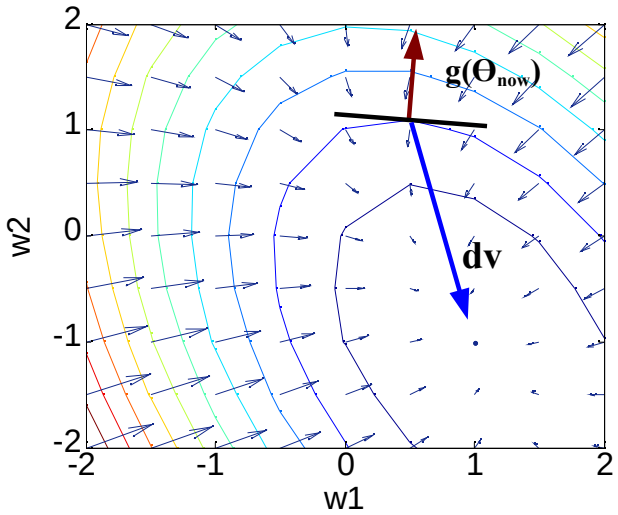

Figure 6.5: possible descent directions from the starting point Θ_{now}.

of our example in section 6.2.2.1, are the all acceptable descent vectors possibilities. If $dv = -g$, (angle$(g,\ dv) = 180°$), this descent is called the steepest descent method, in which dv represents the same direction as the negative gradient direction $(-g)$. The negative gradient direction is not, always, a short way to reach the minimum Θ^*. For this reason many other methods like: conjugate gradient method, Newton's method and *Levenberg-Marquardt*, are developed to direct, quickly as possible, toward the minimum point of objective function f. For more details see [31].

Generally the derivative-based optimisation methods are used to find the nearest local minimum of an objective function, which presupposes that the gradient of this function is computable. It starts from the point Θ_{now} and moves in the direction of the following vector defined in equation, 6.2.16:

$$\| g \| \quad \cos\left(angle(g, dv)\right) \tag{6.2.16}$$

6.3 Neuro-fuzzy systems

The fuzzy inference is a an intelligent computing method based on the concepts of fuzzy set theory, *Lotfi A. Zadeh* [69]. This method has many applications in variety of domains like: control, time series prediction, robotics, systems modelling, and pattern recognitions. The fuzzy rules represent a human knowledge, expressed in natural language, using fuzzy words (in an approximate manner). With help of the composition rules of inference it will be possible to formulate this knowledge of fuzzy reasoning in mathematical relations. This new approach of the analysis of natural physical systems allows us to:

- introduce the knowledge into a model system.

- refine this knowledge by taking in consideration model errors.

The Fuzzy Inference Systems (FIS) are composed of a collection of rules which have the general form:

$$If\ such\ \textbf{Situation}\ then\ such\ \textbf{Conclusion}.$$

A situation is characterized by certain number of expressions of the type x is A, where x is a variable and A a label. It is necessary, for the operational phase, to interpret the qualitative set to the quantitative set, or the subjective set with the objective set. That is carried out primarily by the choice of the expression universe and the membership functions which will define precisely to which degree x is A. There are two various types of *FIS*, which are mainly used:
- $Mamdani$, and
- $Takagi$-$Sugeno$ and $Kang$.

In our thesis, the second type of FIS, $Takagi$-$Sugeno$ and $Kang$ model [67], is used. The main difference between $Mamdani$ and $Sugeno$ is that the $Sugeno$ output membership functions are either linear or constant. This type is based on rules where the antecedent is composed of linguistic variables and the consequence is represented by a linear function of input variables. The most used form for this type of FIS models is shown in the following equation, 6.3.1, in which the consequence constitutes a linear combination of variables implied in the antecedent.

$$if x_1\ is\ A_1\ and\ \ldots .\ x_n\ is\ A_n\ Then\ Y\ = p_1\ x_1\ +\ \ldots\ +\ p_n\ x_n\ +\ p_0 \qquad (6.3.1)$$

where:

x_i: are input variables of the system

Y: is output variable of the system

p_i: real variables

A_i: are parameters of subsets (membership-functions), which define the degree of membership, between 0 and 1, of each point of the input space, corresponding to linguistic labels.

The output of a *Takagi-Sugeno* and *Kang* model using Fuzzy Inference System composed of m rules is formulated as a weighted average of the individual outputs, $y_i : i = 1 \ldots, m$, provided by each rule, equation 6.3.2.

$$\frac{\sum_{i=1,\ldots,m} w_i y_i}{\sum_{i=1,\ldots,m} w_i} \tag{6.3.2}$$

where w_i: is the product of membership degrees of inputs x_i for corresponding membership functions.

A *Sugeno* system is suited for modeling nonlinear physical phenomenon (process systems) by interpolating between multiple linear models, because of the linear dependence of each rule on the input variables of a system.

Fuzzy systems are knowledge-based models constructed from a collection of linguistic IF-THEN rules, which need automatic tuning of their parameters like the center and basis of membership functions . Many researchers in the fuzzy community have considered Neural Networks (NN) to provide for fuzzy models the capacity of automatic parameters self-tune. They aim to conjugate and combine between learning ability of NN and reasoning ability of FIS. In Neural Networks the information about the system consist in the values of inputs and outputs without fixed model structure, which is considered consequently as black-box model [50]. In this case the parameters have no significant physical sense (no interpretability), and no physical description of their corresponding real systems. However, in fuzzy models, available prior knowledge about the system allows the construction of knowledge-based models. With

fuzzy logic method it is possible to introduce or integrate such prior knowledge of the system into the mathematical model. After this introduction about these two approaches, it is necessary to notify that the hybridising between them ($Neuro-Fuzzy$ Systems) to improve the behaviour of their resulting models for either modelling or classification of systems, lead generally to the loss of the physical interpretation of the various FIS' parameters. This notification is given because the notion of the Interpretability of FIS must be considered in such hybrid $Neuro-Fuzzy$ Systems. Otherwise, the resulting optimised models, which are knowledge-based models before optimisation, loss the notion of knowledge and will be uninterpreted. To preserve the notion of model-interpretation in case of *neuro-fuzzy* systems, it is necessary to require some constraints on their optimisation methods.

In the next section, 6.4, hybrid supervised approach, Fuzzy Trimmed Mean Classification ($FTMC$) algorithm is proposed. This approach is mathematical derivative-based optimisation method, without introducing of Neural Networks. Trimmed mean-based rules initialisation gives an interpreted optimised final classifier-model in this approach.

6.3.1 Neuro-fuzzy systems architecture

In the preceding two sections, 6.2 and 6.3, both fuzzy logic systems and neural network systems are described. The hybrid system known as *neuro-fuzzy* system has been became worldwide tool for many researchers in this field and found its birth in the increase of complexity of systems and their identification. Different architectures of *neuro-fuzzy* system have been investigated by number of researchers.

In case of first order *Takagi-Sugeno* and *Kang* system, the fuzzy inference system,

for two inputs x and y and one output z, can be presented as example of four rules. Each input space is presented with two membership functions in the following way: two membership functions A_1 and A_2 for the input space x and two membership functions B_1 and B_2 for the input space y. Consequently we get a system, which is described with four rules $R_i : i = 1, \ldots 4$, in equation 6.3.3. The output level z_i of each rule is weighted by the firing strength w_i of the rule. The firing strength is $w_i \ = T\ (A(x),\ B(y))$. Many operators T are possible, in this case T is AND mathematical operation.

$$
\begin{aligned}
&R_1 : If\ x_i\ is\ A_1\ and\ y_i\ is\ B_1\ then\ z_1 \ = \ p_1\ x_i \ + \ q_1\ y_i \ + \ r_1.\\
&R_2 : If\ x_i\ is\ A_1\ and\ y_i\ is\ B_2\ then\ z_2 \ = \ p_2\ x_i \ + \ q_2\ y_i \ + \ r_2.\\
&R_3 : If\ x_i\ is\ A_2\ and\ y_i\ is\ B_1\ then\ z_3 \ = \ p_3\ x_i \ + \ q_3\ y_i \ + \ r_3.\\
&R_4 : If\ x_i\ is\ A_2\ and\ y_i\ is\ B_2\ then\ z_4 \ = \ p_4\ x_i \ + \ q_4\ y_i \ + \ r_4.
\end{aligned}
\tag{6.3.3}
$$

where $A(.),\ B(.)$ are the membership functions for Inputs x and y. In figure 6.6 the

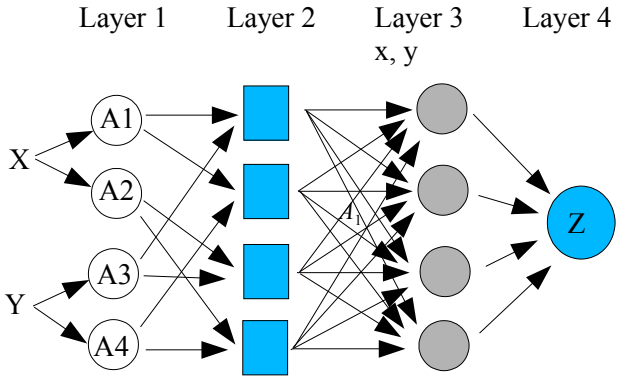

Figure 6.6: General connection structure of FIS for Neuro-Fuzzy model, in case of four rules with linear consequents.

neurons of every layer are similar and have the same specific function. The first one

is adaptive and calculate the membership degrees for different membership functions, which subdivides the corresponding input variable set. The corresponding function for every *gbellmf* (generalised bell-shaped membership function) is given for x input variable set as:

$$output = \mu_{A_i}(x) = \frac{1}{1 + (\frac{x-b}{a})^{2c}} \tag{6.3.4}$$

where $(a,\ b,\ and\ c)$: are parameters set.

The second layer calculate the product of all the incoming signals, each neurone's output represents the firing strength of its rule:

$$output = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y),\ i = 1,\ 2. \tag{6.3.5}$$

In the third layer each neuron has input variables: x, y and θ, θ: is parameters vector. As output it gives the product of normalised firing strength $\overline{w_i}$ and rules outputs z_i as:

$$output = \overline{w_i} z_i = \frac{w_i}{w_1 + w_2 + w_3 + w_4} \cdot z_i. \tag{6.3.6}$$

In case of first order *Takagi-Sugeno* model, the functions $z_i,\ i = 1,\ \ldots,\ 4$, have the following expression:

$$z_i = f(x, y, \theta) = p_i x + q_i y + r_i. \tag{6.3.7}$$

The last layer unit calculates the overall output:

$$output = Z_i = \frac{w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4}{w_1 + w_2 + w_3 + w_4}. \tag{6.3.8}$$

All parameters of figure 6.6 can be initialised using the prior knowledge about the system.

6.3.2 Neuro-fuzzy systems optimisation

The precedent section treated the general *neuro-fuzzy* models architecture. This neural architecture needs both structure and parameters to be designed, which should be able to describe and model the corresponding real system. Optimisation procedure of the builded *neuro-fuzzy* model can be accomplished by the optimisation of both structure and parameters. Hybrid *Neuro-Fuzzy* Systems are under intensive investigations. Such attention to them is mainly provoked by the combination of fuzzy sets and rules initialisation using prior knowledge of system, and also learning algorithms derived from neural networks theory. Fuzzy rules contain information generally in radial basis functions (RBF) parameters, the experts can define membership functions of input set and rules output parameters (consequents) to design fuzzy systems. Neural Networks (NN) have the task to train these parameters for getting the optimised model-parameters. Such *neuro-fuzzy* models have generally parameters divided in two parts. The first one is the input-set parameters, which are called also nonlinear parameters, these parameters describe the Gaussian functions. The second part is the output-set parameters, which are called also consequent parameters, because they describe the rules' consequents.

Model optimisation is the process to find parameters setting and model-archeticture, which lead to the best performances. Usually, this kind of problems is solved by different techniques for model-parameters and model-archeticture. The first part of optimisation is called paramitrical optimisation, it deals with only the model-parameters. Usually, this type of optimisation is divided it self in two other optimisation types: the first type are derivative-based optimisation methods like descent methods and

newton's methods, and the second type are non derivative-based optimisation methods like Genetic Algorithms (GA) and Random Search (RS). The second part of optimisation is called structural optimisation, it deals with only the model-archeticture. Usually this part treats the selection and projection of feature input-vectors, see section 5.4, and also deals with the input space partitioning and clustering.
About the parametrical-optimisation, it is possible to create many hybrid algorithms, which work together to deal with input and output parameters. Many hybrid algorithms are known actually see [68, 32, 49, 40].

6.4 Notion of interpretability

This section describes the proposed supervised classification approach, Fuzzy Trimmed Mean Classification ($FTMC$). It fulfills the transparency (interpretability) of fuzzy system and good accuracy, and allows the cooperation between expert rules and induced rules. An initial fuzzy rules system is generated using the statistic's trimmed mean method [17]. This method provides this algorithm a good initial values of the centers and bases of the membership functions. In this approach, each class is clustered independently from the other classes, and is modeled by the components of gaussian functions. Some of known initialisation methods, such as grid-type partitioning, build a complex and non-interpretable initial models and the optimisation learning steps become computationally demanding and long. Statistic's trimmed mean method can avoid infrequent observations data points and gives an optimal model initialisation, which needs, after that, only a few train-epochs to reach the desired level of performance. Thus it can be considered as a rapid model-identification development tool.

The classical fuzzy rule-based classifier-models consists of an interpretable if-then fuzzy rules, each one describing one of some defined classes. In this section the goal of study is not only to look for the classification accuracy but it focuses on the design of interpretable fuzzy rule based classifier-model. The parameters of this classifier-model after its optimisation should have a physical significance corresponding to the real system. The notion of interpretability in system models based on fuzzy logic is primordial. The interpretability of the parameters in optimised *neuro-fuzzy* models without some constraints is not guaranteed. Hence real effort must be made to keep the transparency in rule parameters. The proposed fuzzy classification algorithm ($FTMC$) to perform the EMG-based finger-movements classification has some advantages like: interpretability, transparency, distinguishable fuzzy sets, coverage and simplicity. It will be compared with other known intelligent computational methods to evaluate its performances.

6.4.1 Input fuzzy sets initialisation

Initial fuzzy rule base, section 6.3, is derived from n available input-output data pairs (X_{nf} , Y_n), the input Matrix $X_{nf} = [X_{ij}]$, where i = *1, ...,* n is number of measured samples and j = *1, ...,* f is number of features. First, proposed supervised $FTMC$ classification method, see our publication [73], extracts initial fuzzy model from data set using statistical trimmed mean method to obtain a set of initial rules. Second, it applies optimisation algorithms to adapt its rules' parameters. Initial fuzzy model can be defined using centres and radius values for different classes. For this task the mean value for each feature vector F_j and each class k is calculated without taking in consideration the outlier samples. $F_j = X_{ij}$ *(* i = *1, ...,* n_k*)* represents the feature

of indices j for the class C_k, where n_k denotes the number of samples for the class C_k. The number of classes is K, where $k = 1, ..., K$. The number of samples for all classes is $n = \sum_{k=1}^{K} n_k$. In short description of this method, each feature vector of matrix (label for each class) in the matrix X_{nf} is ordered from the smallest to largest value, deleting a selected number of samples from each end of the ordered list, and then averaging the remaining values, see example figure 6.7. For this task we have

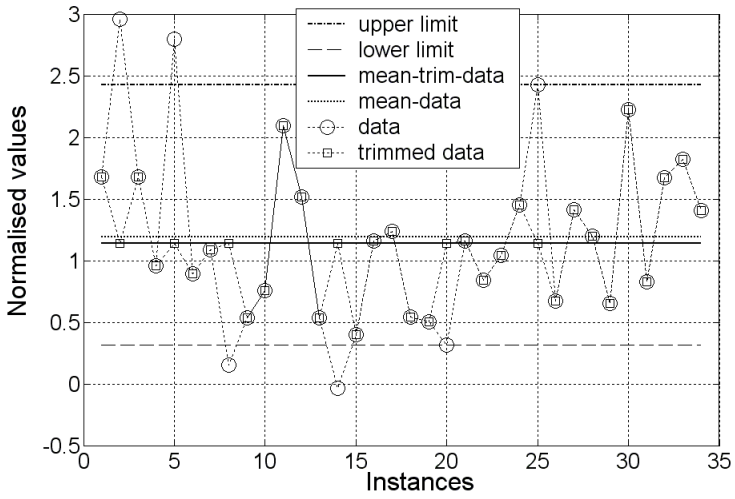

Figure 6.7: Mean of original data and mean of trimmed data.

to choose the trimming percentage β, $1 - \beta$ denotes the percentage of values deleted from each end of the ordered list, in this case β=0.9.

The mean of each feature vector F_j and for each class C_k is given as:

$$V_{jk} = \frac{1}{n_k} \sum_{1=1}^{n_k} X_{ij} \tag{6.4.1}$$

This mean has an extreme sensitivity to each single outlier, hence there is a suspect to take the mean vector V_{ij} as a measure of centres. Outliers are infrequent observations data points, which do not appear to follow the characteristic distribution of the rest

of data. Statisticians have proposed a trimmed mean as solution for this problem. The number of samples that will be deleted from both ends of the ordered label vector is given as:

$$n_{dk} = 2 \times round(n_k \times (1 - \beta)) \tag{6.4.2}$$

round is *matlab* command: it rounds the elements to the nearest integer.
This procedure lets us define the coordinates of centres for the C_k classes in f multi-dimensional space. For f=2, the samples of each class can be delimited with ellipses, which have the above described centres and radius on each axis. In this example,

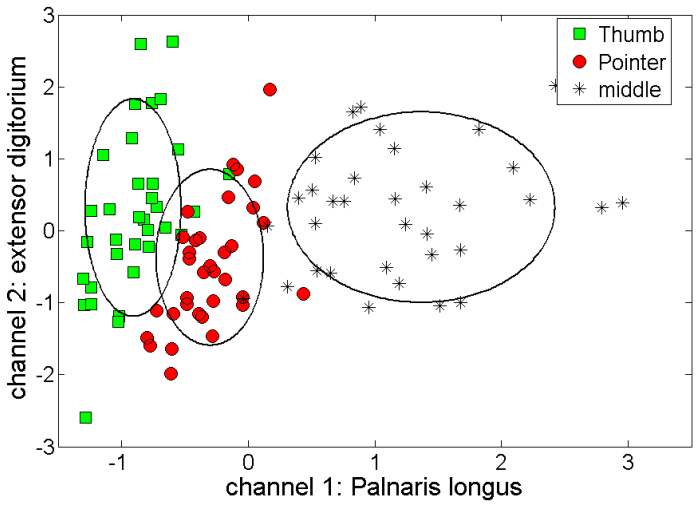

Figure 6.8: Trimmed mean method-based groups delimitation.

figure 6.8, there are three classes corresponding to three different finger movements, which are :thumb, pointer and middle. The parameters of each ellipsoid will be used to generate the initial fuzzy sets. For this task generalised membership function, equation 6.4.19 is chosen. Hence input fuzzy sets initialisation, using trimmed mean-based for these data samples, will have the following partition, figure 6.9, of the input space. If zero order *Takagi-sugeno* model is chosen, then the consequent parameters

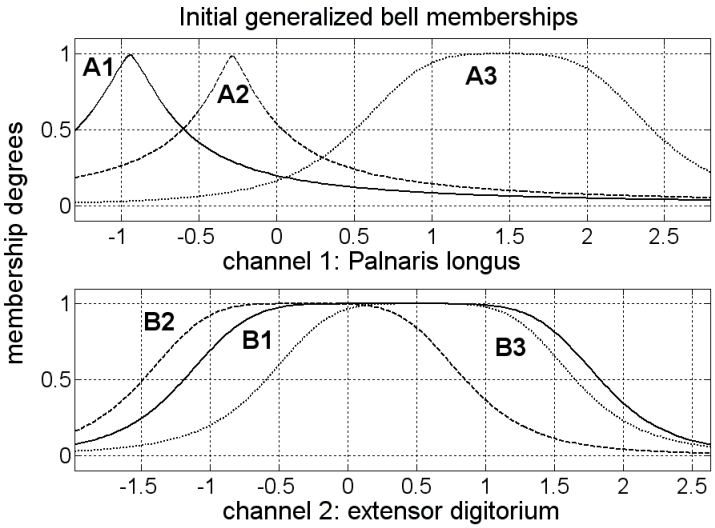

Figure 6.9: Trimmed mean method-based input fuzzy sets initialisation.

of this classifier-model are independent from input variables.

This approach can find a good starting point in proximity of the global minimum, and consequently the learning-time of derivative-based optimisation methods will be shorter, only few epochs of training are required. This procedure avoids also the big changes in membership functions overlapping, which can lead to an inversion of fuzzy sets. The initial input-space partition with this method fulfils many criteria of transparency and semantic properties [24] and [15].

- coverage: the all entry space is covered. That means the model is able to perform an output for all input samples.

- semantic order relation: we have not inversion of fuzzy sets.

6.4.2 Mathematical description

Consider the continuous-time nonlinear system of two inputs $x,\ y$ and one output z of dimension n.

$x^T = x_1\ , x_2,\ \ldots,\ x_n$

$y^T = y_1\ , y_2,\ \ldots,\ y_n$

Each input space is presented with two membership functions in the following way: two membership functions A_1 *and* A_2 for the input space x and two membership functions B_1 *and* B_2 for the input space y. Consequently we get a system, which is described with four rules $R_r : r = 1,\ \ldots\ 4$, equation 6.4.3. The output level z_j of each rule is weighted by the firing strength w_r of the rule. The firing strength is $w_r\ = T\ (A(x),\ B(y))$. Many operators T are available, in this case T is AND mathematical operation.

$$
\begin{aligned}
&R_1 : If\ x_j\ is\ A_1\ and\ y_j\ is\ B_1\ then\ z_1\ =\ p_1\ x_j\ +\ q_1\ y_j\ +\ r_1.\\
&R_2 : If\ x_j\ is\ A_1\ and\ y_j\ is\ B_2\ then\ z_2\ =\ p_2\ x_j\ +\ q_2\ y_j\ +\ r_2.\\
&R_3 : If\ x_j\ is\ A_2\ and\ y_j\ is\ B_1\ then\ z_3\ =\ p_3\ x_j\ +\ q_3\ y_j\ +\ r_3.\\
&R_4 : If\ x_j\ is\ A_2\ and\ y_j\ is\ B_2\ then\ z_4\ =\ p_4\ x_j\ +\ q_4\ y_j\ +\ r_4.
\end{aligned}
\tag{6.4.3}
$$

where $A(.),\ B(.)$: are the membership functions for Inputs x and y. The global output of the system is the weighted average of all rule outputs, computed in equation 6.4.4 as:

$$
Z = \frac{\sum_{r=1,\ldots,4} w_r(x,y) z_j}{\sum_{r=1,\ldots,4} w_r} \tag{6.4.4}
$$

The first order *Takagi-Sugeno* rule architecture is shown in the following diagram, figure 6.10. Where p_i, q_i and r_i: are the linear parameter sets.

The firing strength w_r, $r = 1,\ 2,\ 3,\ 4$ of each rule is given as a T operator function

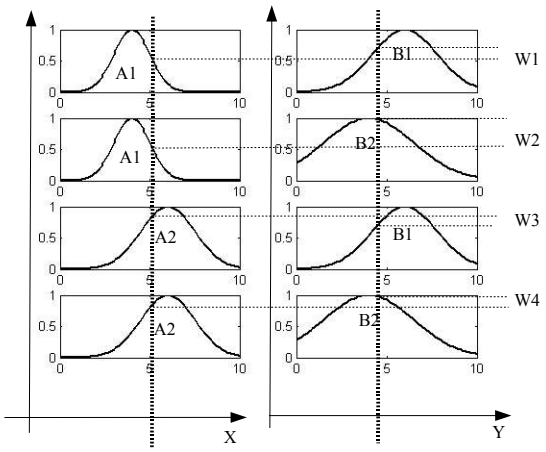

Figure 6.10: Two inputs and four rules *Takagi-Sugeno* fuzzy inference system.

of the membership degrees:

$$w_1 = T[\mu_{A_1(x)},\ \mu_{B_1(y)}].$$

$$w_2 = T[\mu_{A_1(x)},\ \mu_{B_2(y)}].$$

$$w_3 = T[\mu_{A_2(x)},\ \mu_{B_1(y)}].$$

$$w_4 = T[\mu_{A_2(x)},\ \mu_{B_2(y)}].$$

where T: is the Operator function, in our case it presents the AND or product operator, and $\mu_{Ai(x)},\ \mu_{Bi(y)}$: are membership degrees.

Then the overall output is expressed as a linear combination of the consequent parameters, equation 6.4.5:

$$Z = \frac{w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4}{w_1 + w_2 + w_3 + w_4} \tag{6.4.5}$$

or,

$$Z = \overline{w_1} z_1 + \overline{w_2} z_2 + \overline{w_3} z_3 + \overline{w_4} z_4.\ \ with\ \overline{w_r} = \frac{w_r}{\sum_{r=1,4}(w_r)} \tag{6.4.6}$$

6.4.3 Parameters identification

By observing input-output data pairs of nonlinear physical phenomenon it is possible to build a mathematical model with adapted parameters, which can identify our real system. The purposes of system identification can be: approximation, modelling [44] or classification [52].

The following figure shows a schematic diagram of real system identification where outputs Z and Z^* are both system and model outputs respectively. The identified mathematical model should be updated till the acceptable difference measure $D = Z - Z^*$ is reached, figure 6.11. In section 6.4.2 the continuous-time nonlinear system

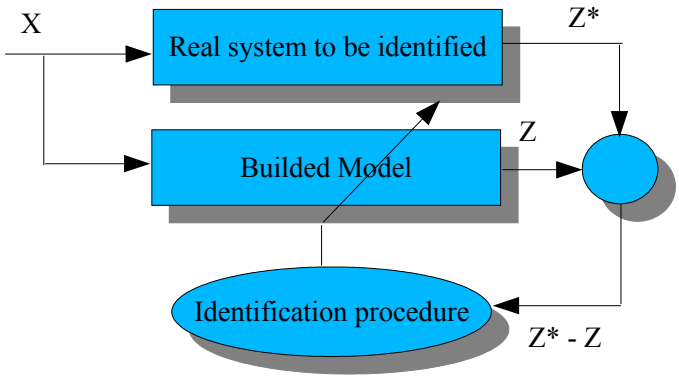

Figure 6.11: Schematic diagram for mathematical model identification

of two inputs x, y and one output Z of dimension n is described in the following mathematical model,6.4.7

$$Z = \overline{w_1} z_1 + \overline{w_2} z_2 + \overline{w_3} z_3 + \overline{w_4} z_4. \tag{6.4.7}$$

The structure of this model is already defined empirically, this structure in fuzzy inference system is determined by an expert who has enough knowledge about the real system. There are many methods, which help us to define the number of *MFs*

and rules, such as clustering methods. Now it remains only the optimisation of model parameters. This model is composed of two parameter sets, first are the nonlinear parameters of the membership functions, A_1, A_2, B_1, and B_2. Second the linear parameters of the consequent rules, p_i, q_i *and* r_i, $i = 1, 2, 3, 4$. To illustrate identification methods for these two types of parameters, this study will be divided in two parts, first identification of linear parameters with consideration that the nonlinear parameters are constant, and second identification of nonlinear parameters with consideration that the linear parameters are constant.

a) Linear parameters identification

The expressions of the functions z_1, z_2, z_3, *and* z_4 in equation 6.4.7 are introduced to get the following equation 6.4.8

$$\begin{aligned} z_j = \overline{w_1}(\ p_1\ x_j\ +\ q_1\ y_j\ +\ r_1) + \overline{w_2}(\ p_2\ x_j\ +\ q_2\ y_j\ +\ r_2) \\ +\overline{w_3}(\ p_3\ x_j\ +\ q_3\ y_j\ +\ r_3) + \overline{w_4}(\ p_4\ x_j\ +\ q_4\ y_j\ +\ r_4). \end{aligned}$$

or,

$$\begin{aligned} z_j = (\overline{w_1}x_j)p_1 + (\overline{w_1}y_j)q_1 + (\overline{w_1})r_1 + (\overline{w_2}x_j)p_2 + (\overline{w_2}y_j)q_2 + (\overline{w_2})r_2 \\ +(\overline{w_3}x_j)p_3 + (\overline{w_3}y_j)q_3 + (\overline{w_3})r_3 + +(\overline{w_4}x_j)p_4 + (\overline{w_4}y_j)q_4 + (\overline{w_4})r_4 \end{aligned} \tag{6.4.8}$$

This last equation 6.4.8 can be formulated in the following new equation 6.4.9

$$\begin{aligned} z_j = f_1(x_j, y_j)p_1 + f_2(x_j, y_j)q_1 + f_3(x_j, y_j)r_1 \\ +f_4(x_j, y_j)p_2 + f_5(x_j, y_j)q_2 + f_6(x_j, y_j)r_2 \\ +f_7(x_j, y_j)p_3 + f_8(x_j, y_j)q_3 + f_9(x_j, y_j)r_3 \\ +f_{10}(x_j, y_j)p_4 + f_{11}(x_j, y_j)q_4 + f_{12}(x_j, y_j)r_4 \end{aligned} \tag{6.4.9}$$

$f_m(x_j,\ y_j)$: are the known functions of our input vector pairs $\{x_j,\ y_j\}$, where m: represents the number of linear parameters.

Finally the output Z in equation 6.4.9 describes an output of a linear model, which is generally given by the linearly parameterised expression, equation 6.4.10

$$Z = f_1(u)\theta_1 + f_2(u)\theta_2 + \ \ldots\ + f_m(u)\theta_m \tag{6.4.10}$$

where $f_1,\ \ldots,\ f_m$: are known functions of $u = [u_1,\ \ldots,\ u_j]^T,\ u_j = \{x_j,\ y_j\}$, and $\theta_1,\ \ldots,\ \theta_m$: are unknown parameters.

In equation 6.4.9 the linear parameters are $p_i,\ q_i\ and\ r_i,\ i = 1,\ 2,\ 3,\ 4$. It is considered that there are n measured data pairs $[(x_j,\ y_j);\ z_j]$ of the real system, where $j = 1,\ \ldots,\ n$. Our identified model is presented in these following n linear equations 6.4.11:

$$\begin{aligned}
z_1 &= f_1(u_1)p_1 + f_2(u_1)q_1 + f_3(u_1)r_1 + \ \ldots\ + f_{10}(u_1)p_4 + f_{11}(u_1)q_4 + f_{12}(u_1)r_4,\\
z_2 &= f_1(u_2)p_1 + f_2(u_2)q_1 + f_3(u_2)r_1 + \ \ldots\ + f_{10}(u_2)p_4 + f_{11}(u_2)q_4 + f_{12}(u_2)r_4,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\\
z_n &= f_1(u_n)p_1 + f_2(u_n)q_1 + f_3(u_n)r_1 + \ \ldots\ + f_{10}(u_n)p_4 + f_{11}(u_n)q_4 + f_{12}(u_n)r_4,
\end{aligned} \tag{6.4.11}$$

In matrix form, the precedent equations can be written as:

$$M\Theta = Z \tag{6.4.12}$$

where M is a matrix of dimension $n\ \times m$:

$$M = \begin{pmatrix} f_1(u_1) & \dots & f_{12}(u_1) \\ . & . & . \\ . & . & . \\ f_1(u_n) & \dots & f_{12}(u_n) \end{pmatrix}$$

Θ is a vector of unknown linear parameters of dimension $1 \times m = 12$ to be identified:

$$\Theta = [p_1,\ q_1,\ r_1,\ p_2,\ q_2,\ r_2,\ p_3,\ q_3,\ r_3,\ p_4,\ q_4,\ r_4]^T$$

Z is an output vector of dimension $1 \times n$:

$$Z = [z_1,\ z_2,\ .,\ .,\ .,\ z_n]^T$$

To identify this unknown nonlinear parameters vector it is necessary that the measured data pairs dimension of the system, should be greater than unknown parameters dimension of its identified model. This condition is verified in this case ($n \geq 12$,). If M is square and nonsingular (its determinant is nonzero), then it is possible to solve the equation: $M\ \Theta = Z$, as:

$$\Theta = M^{-1}Z \tag{6.4.13}$$

Unfortunately there is always, in real systems, a difference between the measured real system response Z^* and the identified mathematical system response. This difference can be described as the error e due to many external factors non identified. Then to represent the real output response, the error among e will be added in the following way:

$$M\ \Theta + e = Z^* \tag{6.4.14}$$

For each measured data pairs $[(x_j,\ y_j),\ z_j]$, $j = 1,\ \dots,\ n$, the error e_j is given as: $e_j = z_j^* - z_j(p_i,\ q_i,\ r_i) = {z_j}^* - \sum_{m=1,12}[M_{jm}^T\ \Theta_m]$, $j = 1...\ n$, $i = 1...\ 4$, and $m = 1, ...,\ 12$. The sum of squared error of the error vector $e(\Theta) = [e_1,\ \dots,\ e_n]^T$ represents the

error in our identified model response. This error is occurred in estimating unknown linear parameters vector Θ, hence the sum of squared error is symbolised as $E(\Theta)$:

$$E(\Theta) = \sum_{j=1,n} ({z_j}^* - \sum_{m=1,12} [M_{jm}^T \ \Theta_m])^2 = e^T \ e. \tag{6.4.15}$$

In matrix form, the precedent equation can be written as:

$$E(\Theta) = (Z^* - M \ \Theta)^T \ (Z^* - M \ \Theta). \tag{6.4.16}$$

The sum squared error, $E(\Theta)$, defined in equation 6.4.15 is depending on the values of the unknown linear parameters of vector Θ. The best parameter values of this vector Θ, which is designed as Θ^* corresponding to the minimum value of sum squared error $E(\Theta)$.

Using Least Squares Estimator (LSE) method [63] it is possible to solve the precedent equation 6.4.16 to reach this optimal solution $\Theta = \Theta^*$, which satisfies the normal equation:

$$M^T \ M \ (\Theta^*) = M^T \ Z. \tag{6.4.17}$$

If $M^T \ M$ is nonsingular, then the optimal solution Θ^* is unique:

$$\Theta^* = [M^T \ Z]^{-1} \ M^T \ Z. \tag{6.4.18}$$

b) Nonlinear parameters identification

The case of generalised bell-shaped membership function for the membership functions, A_1, A_2, B_1 and B_2, figure 6.10, is considered. These functions are defined as:

$$A_{1,2}(x) = \frac{1}{1 + (\frac{x - b_{11,12}}{a_{11,12}})^{2c_{11,12}}} \tag{6.4.19}$$

and

$$B_{1,2}(y) = \frac{1}{1 + (\frac{y - b_{11,12}}{a_{11,12}})^{2c_{11,12}}} \tag{6.4.20}$$

The nonlinear parameters of vector Ω for these four membership functions are given respectively as:

$$\Omega = [\underbrace{a_{11},\ b_{11},\ c_{11}},\ \underbrace{a_{12},\ b_{12},\ c_{12}},\ \underbrace{a_{21},\ b_{21},\ c_{21}},\ \underbrace{a_{22},\ b_{22},\ c_{22}}]^T \tag{6.4.21}$$

In this case these parameters are considered unknown, and the linear parameters vector, $\Theta = [p_1,\ q_1,\ r_1,\ p_2,\ q_2,\ r_2,\ p_3,\ q_3,\ r_3,\ p_4,\ q_4,\ r_4]^T$ is considered known. Ω is a vector of unknown nonlinear parameters of dimension $1 \times m = 12$ (the same dimension as linear parameters). Since the membership functions are continuous and differentiable, the basic learning rule, which is the simple steepest descent method discussed in section 6.2.3 is applied. Like manner as for linear parameters, the error $e(\Omega) = [e_1,\ \ldots,\ e_n]^T$ represents the error in our identified model response. This error is occurred in estimating unknown nonlinear parameters of vector Ω, hence the sum of squared error is symbolised as $E(\Omega)$:

$$E(\Omega) = E([\underbrace{a_{11},\ b_{11},\ c_{11}},\ \underbrace{a_{12},\ b_{12},\ c_{12}},\ \underbrace{a_{21},\ b_{21},\ c_{21}},\ \underbrace{a_{22},\ b_{22},\ c_{22}}]^T) \tag{6.4.22}$$

For each measured data pairs $[(x_j,\ y_j),\ z_j],\ j = 1,\ \ldots,\ n$, the error e_j is given as: $e_j = z_j^* - z_j(a_{st},\ b_{st},\ c_{st}),\ st = 11,\ 12,\ 21\ and\ 22.$

where z_j^*: is the measured output, and $z_j(a_{st},\ b_{st},\ c_{st})$: is the output of the builded model. The gradient-based optimisation method described in section 6.2.3 can be applied to identify each parameter of the vector Ω. Our objective function is $E(\Omega)$. In iterative descent method, like steepest descent, the next values, Ω_{next}, of the vector Ω is determined using a learning rate "step down" and the current point Ω_{now} in the

direction of the gradient g:

$$\Omega_{next} = \Omega_{now} - \rho \boldsymbol{g} \tag{6.4.23}$$

where ρ: represents a learning rate.

$\boldsymbol{g}$: is a gradient vector that moves towards a local minimum Ω^* of our objective function $E : \Re^n \rightarrow \Re$

Parameters identification of vector Ω:

In the equation 6.4.19, the generalised bell-shaped membership function ($gbellmf$) A_1 has three unknown parameters, which are: $a_{11},\ b_{11},\ and\ c_{11}$. The identification of each of them is given as:

$$(a,b,c)_{11,next} = (a,b,c)_{11,now} - \rho \frac{\partial e_j}{\partial (a,b,c)_{11}} \tag{6.4.24}$$

$$\frac{\partial e_j}{\partial (a,b,c)_{11}} = (z_j^* - z_j(a_{st},\ b_{st},\ c_{st})) \frac{\partial z_j(a_{st},\ b_{st},\ c_{st})}{\partial (a,b,c)_{11}} \tag{6.4.25}$$

If the equation 6.4.5 is considered, then we get:

$$\frac{\partial z_r(a_{st},\ b_{st},\ c_{st})}{\partial (a,b,c)_{11}} = \frac{\partial}{\partial (a,b,c)_{11}} \left(\frac{\sum_{r=1}^{r=4} w_r z_r}{\sum_{r=1}^{r=4} w_r} \right), \tag{6.4.26}$$

where r: number of rules.

remark: don't confuse between z_j, which is the measured system output and z_r, which represents the only rules' conclusion.

Further computations of equation 6.4.26 lead to final derivative results of $A_1(x)$ with

respect to parameters $(a,b,c)_{11}$ as:

$$\frac{\partial A_1(x)}{\partial a_{11}} = \frac{2c_{11}}{a_{11}} \cdot (1 - A_1(x)) \cdot A_1(x) \tag{6.4.27}$$

$$\frac{\partial A_1(x)}{\partial b_{11}} = \frac{2c_{11}}{a_{11}} \cdot (\frac{x - b_{11}}{a_{11}})^{2c_{11}-1} \cdot (A_1(x))^2 \tag{6.4.28}$$

$$\frac{\partial A_1(x)}{\partial c_{11}} = -2c_{11} \cdot (\frac{x - b_{11}}{a_{11}})^{2c_{11}-1} \cdot (A_1(x))^2 \tag{6.4.29}$$

In the same way, the final derivative results of $A_2(x)$ with respect to the parameters $(a,b,c)_{12}$ is given as:

$$\frac{\partial A_2(x)}{\partial a_{12}} = \frac{2c_{12}}{a_{12}} \cdot (1 - A_2(x)) \cdot A_2(x) \tag{6.4.30}$$

$$\frac{\partial A_2(x)}{\partial b_{12}} = \frac{2c_{12}}{a_{12}} \cdot (\frac{x - b_{12}}{a_{12}})^{2c_{12}-1} \cdot (A_2(x))^2 \tag{6.4.31}$$

$$\frac{\partial A_2(x)}{\partial c_{12}} = -2c_{12} \cdot (\frac{x - b_{12}}{a_{12}})^{2c_{12}-1} \cdot (A_2(x))^2 \tag{6.4.32}$$

The final derivative results of $B_1(y)$ with respect to the parameters $(a,b,c)_{21}$ is given as:

$$\frac{\partial B_1(y)}{\partial a_{21}} = \frac{2c_{21}}{a_{21}} \cdot (1 - B_1(y)) \cdot B_1(y) \tag{6.4.33}$$

$$\frac{\partial B_1(y)}{\partial b_{21}} = \frac{2c_{21}}{a_{21}} \cdot (\frac{y - b_{21}}{a_{21}})^{2c_{21}-1} \cdot (B_1(y))^2 \tag{6.4.34}$$

$$\frac{\partial B_1(y)}{\partial c_{21}} = -2c_{21} \cdot (\frac{y - b_{21}}{a_{21}})^{2c_{21}-1} \cdot (B_1(y))^2 \tag{6.4.35}$$

At the end the final derivative results of $B_2(y)$ with respect to the parameters $(a,b,c)_{22}$ is given as:

$$\frac{\partial B_2(y)}{\partial a_{22}} = \frac{2c_{22}}{a_{22}} \cdot (1 - B_2(y)) \cdot B_2(y) \tag{6.4.36}$$

$$\frac{\partial B_2(y)}{\partial b_{22}} = \frac{2c_{22}}{a_{22}} \cdot (\frac{y - b_{22}}{a_{22}})^{2c_{22}-1} \cdot (B_2(y))^2 \tag{6.4.37}$$

$$\frac{\partial B_2(y)}{\partial c_{22}} = -2c_{22} \cdot (\frac{y - b_{22}}{a_{22}})^{2c_{22}-1} \cdot (B_2(y))^2 \tag{6.4.38}$$

6.4.4 Complexity and interpretability consideration in both *FSC* and *FTMC* models

Two EMG surface electrodes are placed on two muscle groups, *palnaris longus* ($channel_1$) and *extensor digitorum* ($channel_2$), the locations of electrodes on the subject's arm is given in figure 6.12. From the input feature space, the classifier must be able to classify the three output classes exploiting the information in EMG signal measurements. For each channel the signal was recorded using a single bipolar surface electrode pair. The signal was sampled at a rate of $4Khz$ using A/D board in an *IBM*

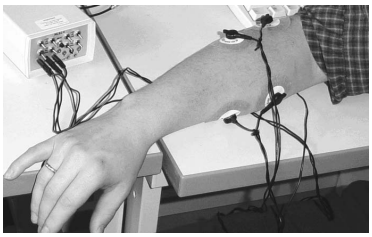

Figure 6.12: Recording of EMG signals using two pairs of surface-electrodes

PC/AT compatible microcomputer. This algorithm is developed with $MATLAB$ 6.5 and is performed in a PC-based off-line process. The human subject was asked to produce a number of continuous movements, 34 single contraction periods are separated from the corresponding sets of continuous movements. Initial *400ms* signal part of each single contraction period is extracted from the raw signal considering defined threshold relative to noise standard-deviation value, see sections 4.2 and 7.3. These extracted signals are analysed using Short Time Fourier Transform ($STFT$). This analysis method gives a measure of both time and frequency information for short signal segments, see section 4.4.4.

Extraction of relevant features, see section 4.4.4.c, needs the use of spectrum analysis based time-frequency domain. Time-frequency analysis based on short-time Fourier

transform ($STFT$), is a form of local Fourier analysis that shows the changes of power spectral density (PSD) of EMG signals during time. This method leads to a better solution to design feature extraction, see section 4.4.4.c. The n^{th} order of frequency moment distribution at time t is defined as:

$$M_n(t) = \sum_k \omega_k^n |STFT(t,k)| \tag{6.4.39}$$

where $M_n(t)$: is the n^{th} moment of the frequency distribution at time t, n: order and ω: frequency.

Using two channels, some EMG training and test data, from filtered raw EMG signal between *30Hz* and *250Hz* are prepared see sections 4.3.1 and 7.2. Three classes labelled 1, 2 and 3 have 51 train-samples and 51 test-samples. The distribution of all samples in 2D dimensional space, $Channel_1$ and $channel_2$, is shown in figure 6.13. Also train and test samples distribution is presented in figure 6.14.

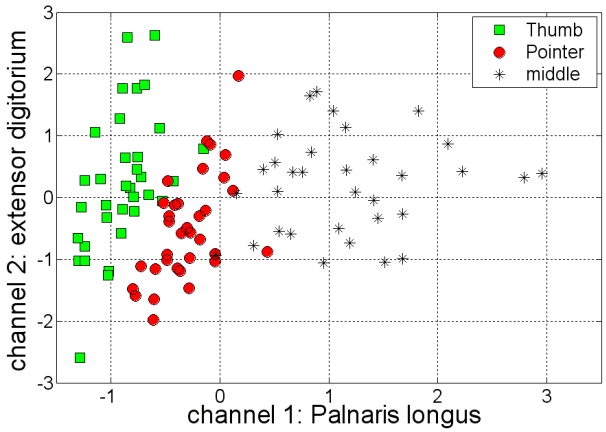

Figure 6.13: Global samples distribution

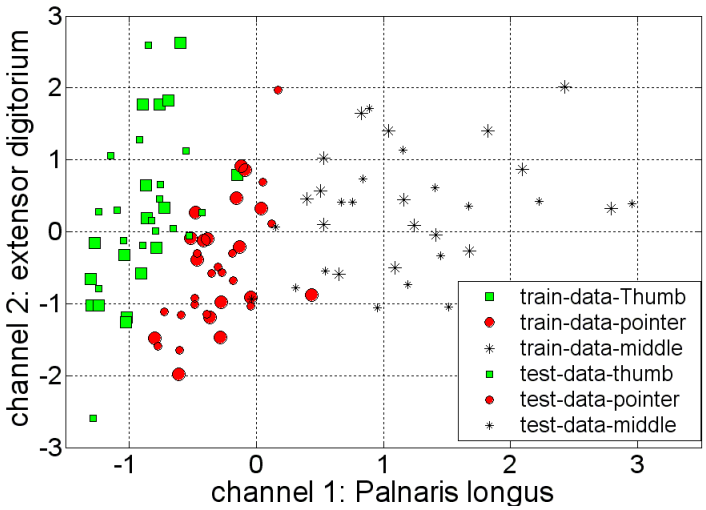

Figure 6.14: Train and test samples distribution.

a) *FTMC* fuzzy classifier-model

The samples of each class for these all train and test data, figure 6.13, can be delimited, according to trimming percentage $\beta = 0.9$. these delimitations are done with ellipses using described Trimmed Mean Inisialisation method (TMI), see section 6.4.1. Derived ellipses for only train samples are presented in figure 6.15. The parameters,

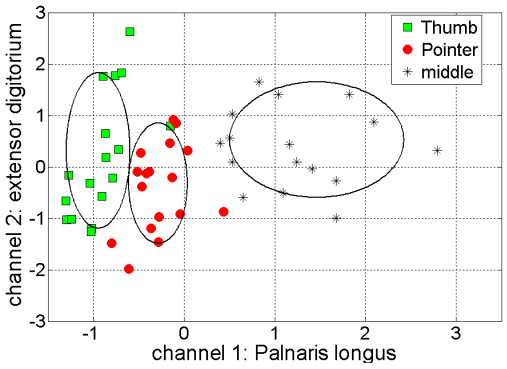

Figure 6.15: Ellipses derived from TMI algorithm for train-samples.

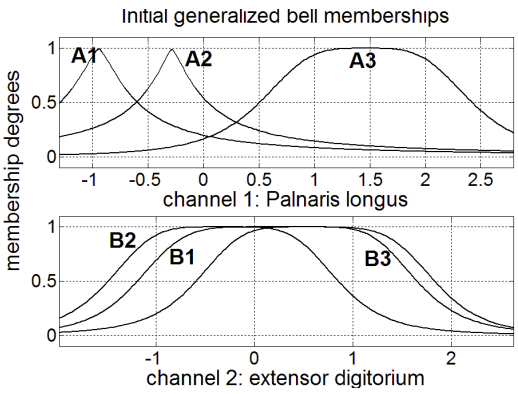

Figure 6.16: Input fuzzy sets initialisation using $FTMC$ algorithm.

centres and radius, of each ellipsoid will be used to generate the initial fuzzy sets. For this task the generalized bell membership function is chosen. Ellipsoids-based input fuzzy sets initialisation for train data, which gives us the partition of our input train-samples space is presented in figure 6.16. Now, after using Trimmed Mean Inisialisation method (TMI), it is possible to build, from data, a compact fuzzy classifier rules with singleton consequents to get zero-order T.S. model. In this method the number of rules to be generated needs to be determined, *a priori*, which are in this case three rules corresponding to our three classes. This builded fuzzy model has three membership functions for each input. The singleton output consequents of this model use the following classification rules, 6.4.40

$$Class_k = \begin{cases} 1 & \text{if} \quad Z_k < 1,5 \\ 2 & \text{if} \quad 1,5 \leq Z_k < 2,5 \\ 3 & \text{if} \quad Z_k \geq 2,5 \end{cases} \tag{6.4.40}$$

The input fuzzy set parameters of the initial model, figure 6.16, are given in table 6.2. This initial model with three rules, which describes 3 classes with singleton con-

Table 6.2: *Gbellmf* membership functions Parameters of initial fuzzy $FTMC$ classifier-model

Input 1	*gbellmf* parameters	Functions
A1	0.351 0.702 -0.946	*gbellmf A1*
A2	0.324 0.647 -0.285	*gbellmf A2*
A3	0.960 1.921 1.463	*gbellmf A3*
Input 2	gbellmf parameters	Functions
B1	1.510 3.020 0.319	*gbellmf B1*
B2	1.169 2.337 -0.316	*gbellmf B2*
B3	1.122 2.244 0.529	*gbellmf B3*

sequents, has average classification accuracy of 80.3922% giving "10" misclassified samples on the test data, figure 6.17. With more detailed study, it is possible, not

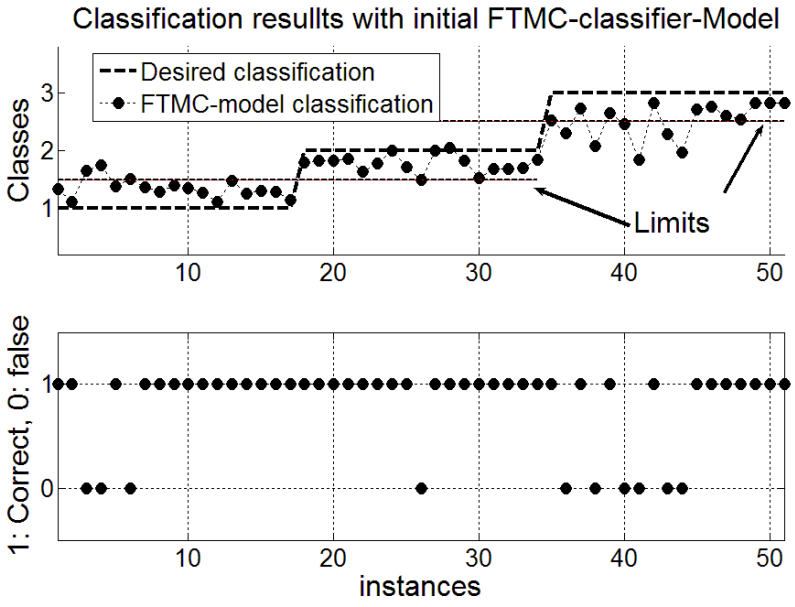

Figure 6.17: Misclassified samples of initial fuzzy $FTMC$ classifier-model

only, to locate the global misclassified samples for all three classes, but also the misclassified samples for each class. For the first class, thumb finger flexion, there are "3" misclassified samples (82.3529%), which are classified as pointer finger movement. About the second class, pointer finger flexion, there is only "1" misclassified sample, or 94.1176% correct, which is classified as thumb finger movement. The third class, middle finger flexion, has "6" misclassified samples, or 64.7059 % correct, which are classified as pointer finger movement.

After generating the initial TMI partitioning of input space using $FTMC$-classifier model, the following adaptation method is applied to perform and increase the results accuracy of this classifier-model. This optimisation is done in two steps: the first one is optimisation of premise parameters (membership functions parameters) using Gradient Descent (GD), see section 6.4.3.b. The second step is the optimisation

of the linear parameters (Consequence parameters) using linear Least Squares Estimator (LSE), see section 6.4.3.a. Initial proposed $FTMC$ classification algorithm, described above has found a good starting point in proximity of the global minimum, which had a classification accuracy equal to 80.3922%. Hence we expect that the adaptation of this initial model will need the application of only a few optimisation epochs and consequently will not have a big effect on the overlap of membership functions and also will not be time-consuming. The optimisation, during only four epochs, of our initial $FTMC$-classifier model, figure 6.16, on both antecedents and consequents parameters using GD and LSE respectively, gives a new repartition of input space. This repartition is presented, with *gbellmf* membership functions, in figure 6.18. These new optimised functions don't present a big difference in compar-

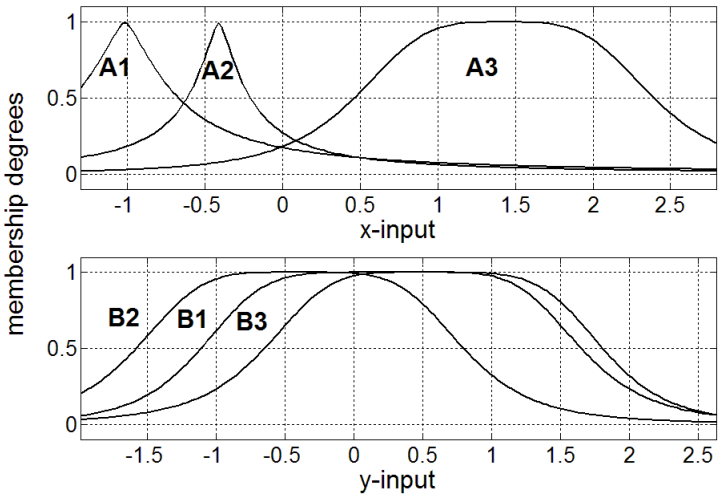

Figure 6.18: *Gebellmf* functions after 4 epochs $FTMC$ optimisation

ison with their initial functions. Hence the Semantic order relation (no inversion of fuzzy sets) see section 6.4.1, is conserved. We obtain consequently the new classification accuracy equal to 88.2353% giving 6 samples of misclassification, they were "10"

before optimisation on the data test see figure, 6.19. It is possible also to locate the

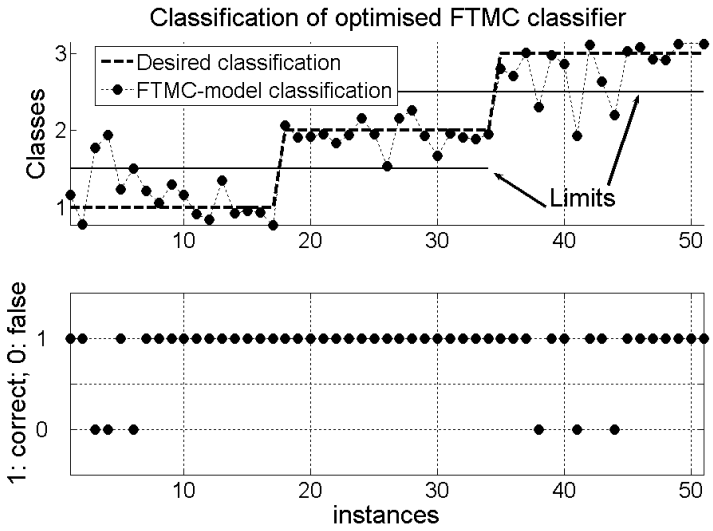

Figure 6.19: Misclassified samples of optimised $FTMC$ classifier-model

global misclassified samples, after optimisation, for all these three classes, and the misclassified samples for each class. For the first class (thumb finger flexion) there are "3" misclassified samples, or 82.3529%, which are classified as pointer finger flexion. About the second class, pointer finger flexion, all samples are classified correctly (100.0% correct). The third class, middle finger flexion, has "3" misclassified samples, or 82.3529 % correct, which are classified as pointer finger movement.

The corresponding rules to the new input fuzzy sets of optimised $FTMC$ classifier-model and their singleton output consequents are given in equation 6.4.41.

$$Class_k = \begin{cases} 1 & \text{if x is } A_1 \text{ and y is } B_1 \text{ then } Z_k = 0.686 < 1,5 \\ 2 & \text{if x is } A_2 \text{ and y is } B_2 \text{ then } 1,5 \leq Z_k = 2.305 < 2,5 \\ 3 & \text{if x is } A_3 \text{ and y is } B_3 \text{ then } Z_k = 3.173 \geq 2,5 \end{cases} \tag{6.4.41}$$

All optimised parameters, nonlinear parameters and linear parameters, belong to the defined domain-space of the real system. There is interpretability and transparency

for the obtained optimised fuzzy $FTMC$ classifier-model. Its parameters values have a clear physical meaning.

b) Fuzzy subtractive clustering (FSC)

The subtractive clustering algorithm is proposed by *Chiu* (1994). It estimates the number of clusters and the cluster centres in a set of data by an iterative procedure. The clusters obtained are used to initialise the fuzzy sets, for $ANFIS$-model. The results, model performances and the notion of interpretability, will be compared with those of our allgorithm. The *Matlab* command $genfis2$, in fuzzy logic toolbox to generate the initial model with subtractive clustering, uses the first order *Takagi-Sugeno* ($T.S.$) model. After $ANFIS$ optimisation method, figure 6.20, the classification accuracies for different epochs: 5, 20 and 50 epochs, are equal to 86.2745% 88,2353% and 90.1961% giving 7, 6 and 5 misclassified samples respectively on the data test. The

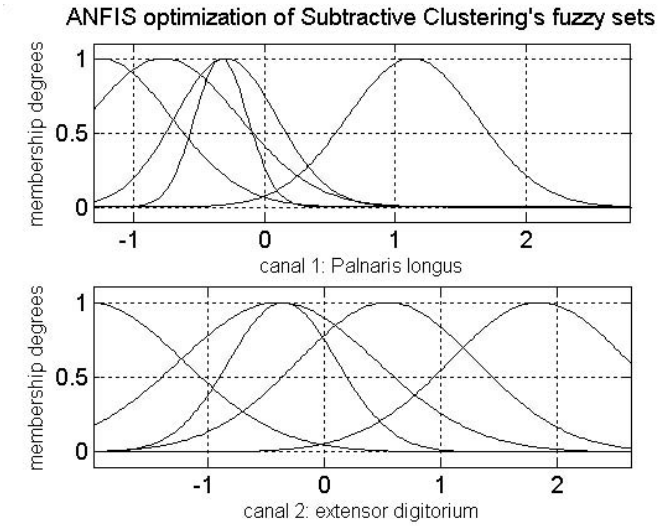

Figure 6.20: $ANFIS$ Optimisation of input fuzzy sets for Subtractive Clustering (FSC) method

obtained fuzzy model with Subtractive clustering method hasn't a physical meaning and is not interpretable, equation 6.4.42. The consequent parameters of the five rules in optimised first order "$T.S.$" model are given in equation 6.4.42.

$$Class_k = \begin{cases} 1 & \text{if x is } A_1 \text{ and y is } B_1 \text{ then } Z_k = -1.521\ 0.6723\ 2.014 \\ 2 & \text{if x is } A_2 \text{ and y is } B_2 \text{ then } Z_k = -1.081\ 0.3386\ -0.215 \\ 3 & \text{if x is } A_3 \text{ and y is } B_3 \text{ then } Z_k = -0.0390\ 0.0102\ 3.072 \\ 4 & \text{if x is } A_3 \text{ and y is } B_3 \text{ then } Z_k = 0.7526\ -0.1067\ 1.76 \\ 5 & \text{if x is } A_3 \text{ and y is } B_3 \text{ then } Z_k = -2.512\ 1.537\ 3.805 \end{cases} \quad (6.4.42)$$

In the following table 6.3, some characteristics of both methods, FSC and $FTMC$, are resumed in comparison form. .

Table 6.3: FSC and $FTMC$ classifier-models characteristics comparison

Mf. Typ.	Our approach *Gbellmf*	Sub. Clustering *Gaussmf*
Nb. Fuzzy sets In-1	3	5
Nb. Fuzzy sets In-2	3	5
Nb Parmtr. In	18	20
Consequent typ.	Singleton	linear
Nb Parmtr. Out	3	15
Nb. of rules	3	5
Interpretability	yes	no
Training(epochs)	4	20
Accuracy(%)	88,2353	88,2353

6.4.5 Conclusion

The proposed fuzzy classification algorithm, to perform the EMG signals-based finger-movements classification, has several advantages that motivate the use of fuzzy systems like: interpretability, transparency, distinguishable fuzzy sets, coverage and simplicity. This algorithm extract fuzzy rules from measured real system data set, which

use trimmed mean measure to avoid infrequent observations data points. It gives an optimal model initialisation, which needs after that only 4 train-epochs to reach the same performance as with FSC, which needs 20 epochs.
A fast and practical method for classification with a simplified fuzzy classifier-model is developed. The structure and parameters, of this proposed fuzzy $FTMC$ classifier-model, are simultaneously optimized. Moreover this model does not loss its interpretability.

6.5 Comparison btween *MLP, RBF, LVQ and FTMC*

Three intelligent computational algorithms will be used to perform the classification of a new considered real system, which is a discrimination of four different hand movements according to their corresponding EMG signals. Intelligent computational algorithms used in this section are those based on neural networks and neuro-fuzzy networks like Multi-Layer Perceptron (MLP), Radial Basis Networks (RBF) and Learning Vector Quantization network (LVQ). The purpose of this section is to illustrate these various intelligent computational algorithms and to compare them with the performance of proposed $FTMC$ fuzzy classifier-algorithm, see our publication [73]. EMG signal pre-processing operation, see section 4.4.4, is performed using spectrum analysis based on Short Time Fourier Transform ($STFT$). With this method it is possible to exploit and to quantify the behaviour of dynamic information presented in EMG signals and to design characteristic (feature) vectors. These vectors can perform a relevant features that lead to a good discrimination of these four different classes of hand movements. The function, which describes the second order moment, (M_2), is used as feature for the further discrimination task. The definition of this

function is given in equation 6.4.39 section 6.4.4.

6.5.1 Proposed *FTMC* classifier-model

a) Initial fuzzy (*FTMC*) classifier-model

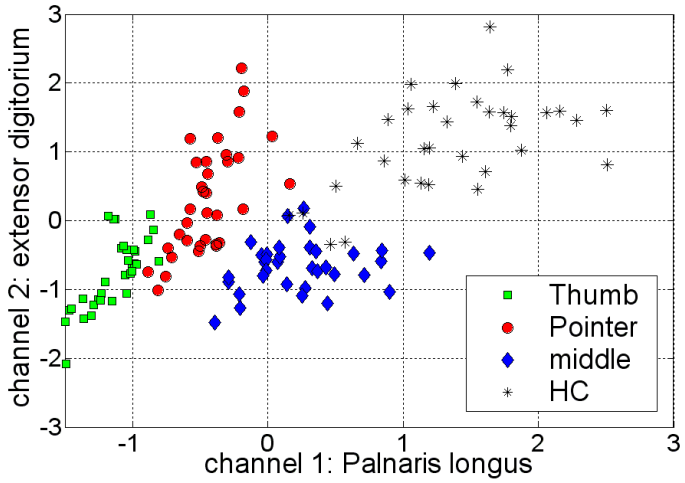

Figure 6.21: Global samples distribution of feature $M2$.

Since all training and test data in the case of Moment of second order feature (M_2) are prepared, figures 6.21 and 6.22, the accuracy of fuzzy $FTMC$ classifier-model according to trimming percentage $\beta = 0.9$ will be performed. The clusters (ellipses) derived from TMC algorithm for the training samples of each class In $2D$ space (two channels), corresponding to four classes: thumb finger flexion, pointer finger flexion, middle finger flexion and hand close movements are presented in figure 6.23. The parameters of each ellipse will be used to generate the initial input-sets of $FTMC$ classifier-model, for this task the generalised bell membership function ($gbellmf$) is chosen. $FTMC$-based input fuzzy sets initialisation for train data gives us the following partition of the input space, figure 6.24. You have obviously observed in this

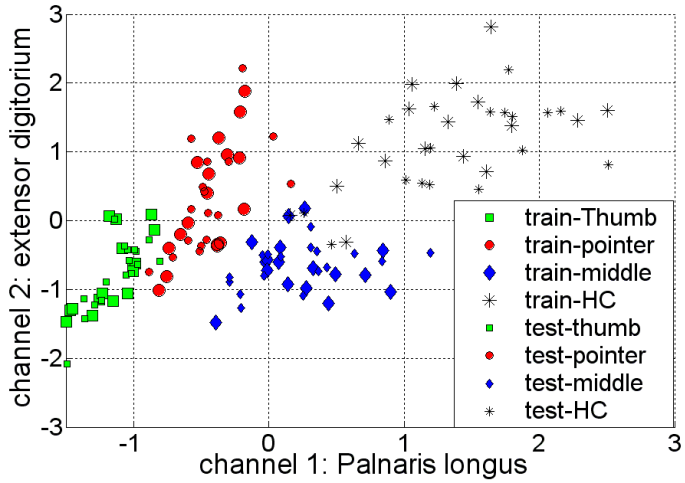

Figure 6.22: Train and test samples distribution of feature $M2$.

figure the existence, in input-2, of redundant sets (similar fuzzy sets). these redundant fuzzy sets can be removed using a similarity measure [61]. This initial model with four rules, which describes four classes with singleton consequents, has average classification accuracy of 79.4118% giving "14" misclassifications on the test data, figure 6.25. It is possible to locate the misclassified samples for each class. For the first class, thumb finger flexion, there are "5" misclassified samples, or 70.5882% correct, which are classified as pointer finger movement. About the second class, pointer finger flexion, there are also "3" misclassified samples, or 82.3529% correct, which are classified as middle finger movement. The third class, middle finger flexion, has only "4" misclassified sample or 76.4706% correct, which is classified as pointer finger movement. the last class, hand closing, has "2" misclassified samples, or 88.2353% correct, which are classified as middle finger movement.

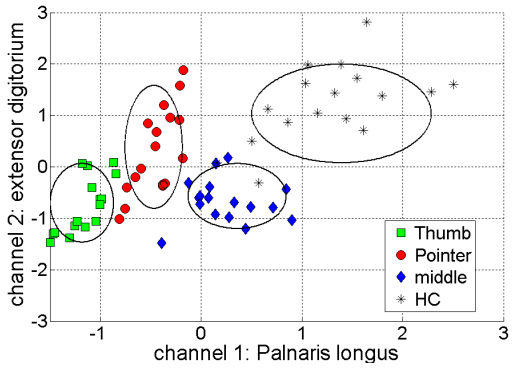

Figure 6.23: Ellipses derived from *TMC* algorithm for train-samples.

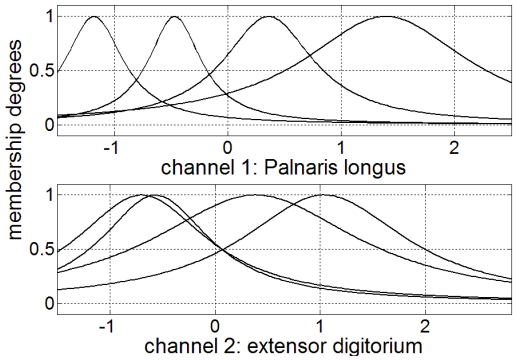

Figure 6.24: Input fuzzy sets initialisation using *FTMC* classifier-model.

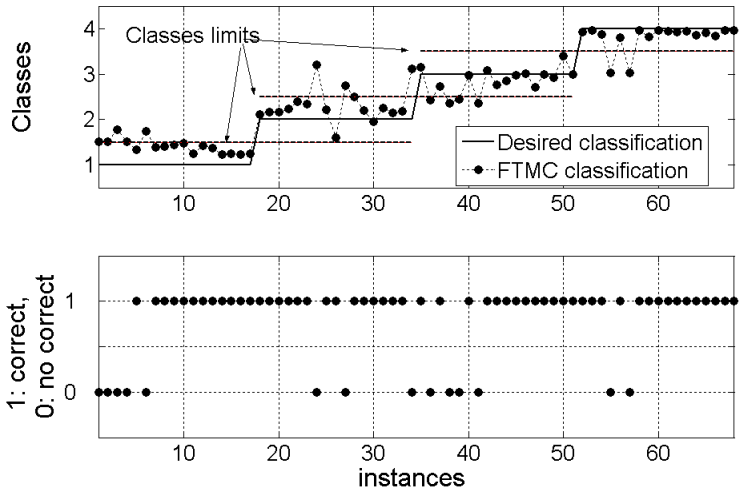

Figure 6.25: Misclassified samples of initial fuzzy *FTMC* classifier-model for test data

b) Optimised *FTMC* classifier-model

Optimisation procedure is applied in two steps. The first step concerns nonlinear parameters using the gradient, see section 6.4.3.b. The second step concerns the optimisation of linear parameters (consequent parameters) with linear Least Squares Estimator (LSE), see section 6.4.3.a. Initial $FTMC$ classification algorithm, described above has found a good starting point in proximity of the global minimum. This initial classifier-model had a classification accuracy equal to 79.4118%. Hence the farther application of a few optimisation epochs will not have a big effect on overlap of the membership functions and will not be time-consuming. The optimisation, during only four epochs, of the initial $FTMC$-classifier model, figure 6.24, on both antecedents and consequents parameters using GD and LSE respectively, gives a new repartition of input space. This repartition is presented in figure 6.26. We obtain

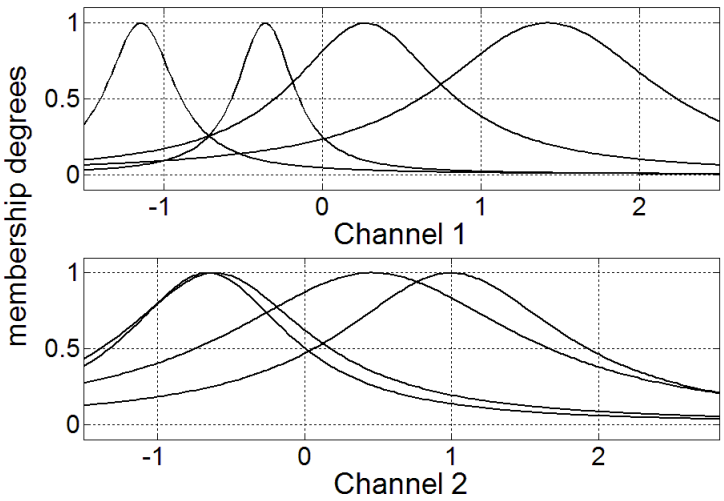

Figure 6.26: New *gebellmf* functions after 4 epochs $FTMC$ optimisation

consequently the new classification accuracy equal to 86.7647% giving 9 samples of misclassification. They were "10" misclassified samples before optimisation, on the

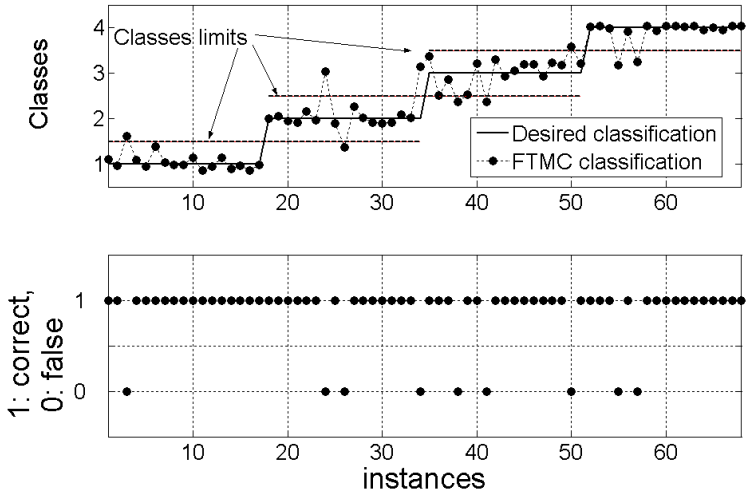

Figure 6.27: Misclassified samples of optimised *FTMC* classifier-model during 4 epochs

data test, figure 6.27. All these results are resumed in the following figure 6.28. After optimisation of these four initial *Gebellmf* functions, the following results are obtained:

For the first class, thumb finger flexion, there is "1" misclassified sample or 94.1176% correct, which is classified as pointer finger movement. About the second class, pointer finger flexion, there is "1" misclassified sample, which is classified as thumb movement, and "2" samples as middle finger flexion ,82.3529% correct. The third class, middle finger flexion, has "3" misclassified samples or 82.3529 % correct, two of them are classified as pointer finger movement, and one as HC. The last class ,hand closing, has "2" misclassified samples or 88.2353% correct, which are classified as middle finger movement. The corresponding rules to the fuzzy sets of optimised *FTMC*

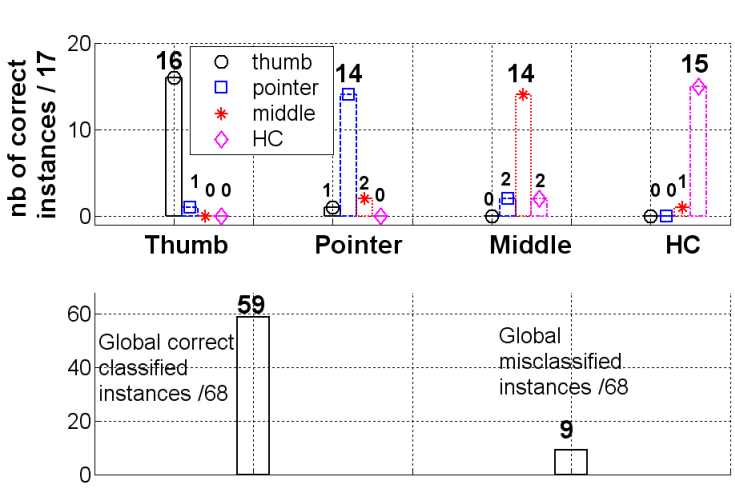

Figure 6.28: *FTMC* classifier-model: Correct- and mis-classified samples for each class and the interferences between them

classifier-model and their singleton output consequents are given in equation 6.5.1.

$$Class_k = \begin{cases} 1 & \text{if x is } A_1 \text{ and y is } B_1 \text{ then } Z_k = 0.4436 < 1,5 \\ 2 & \text{if x is } A_2 \text{ and y is } B_2 \text{ then } 1,5 \leq Z_k = 1.522 < 2,5 \\ 3 & \text{if x is } A_3 \text{ and y is } B_3 \text{ then } 2,5 \leq Z_k = 3.267 < 3,5 \\ 4 & \text{if x is } A_4 \text{ and y is } B_4 \text{ then } Z_k = 4.072 \geq 3,5 \end{cases} \tag{6.5.1}$$

6.5.2 Multi layer perceptron classifier-model

This network is used in many different types of applications, its architecture has a large class of network types with many different topologies and training methods, see section 6.2. The number of neurons in hidden layer is determined based on their performance in training process. For one-neuron output-layer, log sigmoid transfer function *logsig* is used, which gives an output in the range of 0 to 1. The output range between 0 and 1 will be divided in four ranges, since there are four classes to be identified, see Table 6.4. The *MLP* Network is trained, during 100 epochs, with

Table 6.4: One-neuron output-layer defined with log sigmoid transfer function (*logsig*), that is divided in four ranges between 0 and 1

classes	Target Output	Output range	Type of Movement
Class1	0.125	0.00 - 0.25	Thumb finger
Class2	0.375	0.25 - 0.50	Pointer finger
Class3	0.625	0.50 - 0.75	middle finger
Class4	0.875	0.75 - 1.00	Hand closing

Table 6.5: *MLP*-model classification accuracy (class- and average-accuracy), for different hidden layer neuron numbers. (M_2 time-frequency feature)

	Classification accuracy (test data) %				
number of neurons	Thumb	Pointer	Middle	HC	average
10	94.11	70.58	88.23	94.11	86.76
20	94.11	58.82	94.11	88.23	83.82
50	94.11	70.58	82.35	94.11	85.29
	Number of correct classified instances /17				
number of neurons	Thumb	Pointer	Middle	HC	Total
10	16	12	15	16	59/68
20	16	10	16	15	57/68
50	16	12	14	16	58/68

different number of neurons in hidden layer: 10, 20 and 50 neurons. The obtained classification accuracy results are presented in table 6.5, in which the increasing of neurons-number in hidden layer doesn't enhance always the accuracy. More detailed results, figure 6.29, are given for the best obtained *MLP* classifier-model, which has 10 neurons. In the first class, thumb finger flexion, there is "1" misclassified sample, or 94.11% correct, which is classified as pointer finger movement. About the second class, pointer finger flexion, there are "2" misclassified sample, which are classified as

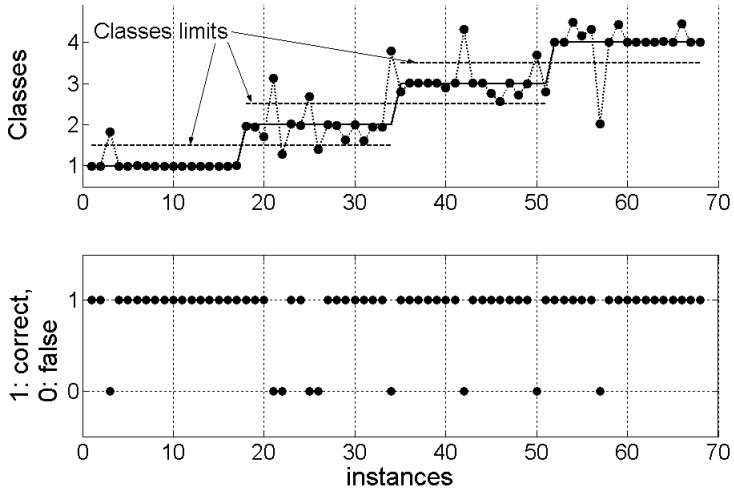

Figure 6.29: Misclassified samples of optimised MLP classifier-model during 100 epochs

thumb movement, and "2" samples as middle finger flexion and "1" sample as "HC" (70.58% correct). The third class, middle finger flexion, has "2" misclassified samples, or 88.23 % correct, which are classified as "HC" movement. The last class, hand closing, has only "1" misclassified sample, or 88.2353% correct, which is classified as pointer finger movement.

6.5.3 Radial Basis Networks classifier-model

The *RBF* Network is a one hidden layer neural Network with several forms of radial basis activation functions, like Gaussian function. The output layer of this type of network is linear. Gaussian functions are monotone and their centre and radius are the parameters of the *RBF*-model.

RBF networks have been applied for many applications including approximation [29], modelling [66] and classification [21], more details see [55] [11]. In classification case the outputs layer correspond to a classes. This method exist in *MATLAB* tool as "*newrb*" command for which the training implementation is the orthogonal least squares (OLS) learning algorithm [13]. The method of creating neurons or centres one at a time (newrb), is used. In each iteration the input vector is applied to create a new neuron, then the error of the new network is checked, if it is not low enough the next neuron is added. This procedure is repeated until the error goal is met, or the maximum number of neurons is reached. So is possible to find the smallest network that can solve the problem within a given error goal. The rate of classification is depending on the hidden unit spread values. Different values for spread parameters between 0.5 and 2.5 are given with a step of 0.2. The aim is to find the optimal value of spread, which is in this case equal to 0.5 and 0.7 see figure 6.30.

One of these two best spread values is chosen. The *RBF*-classifier model is built and tested using $M2$ extracted feature. Classification results are presented in figure 6.31. The global classification accuracy is equal to 83.82 % giving "11" misclassified samples. More detailed results for this best obtained *RBF* classifier-model, with spread value equal to 0.7, are given. In the first class, thumb finger flexion, there is "1" misclassified sample, or 94.11% correct, which is classified as pointer-finger movement.

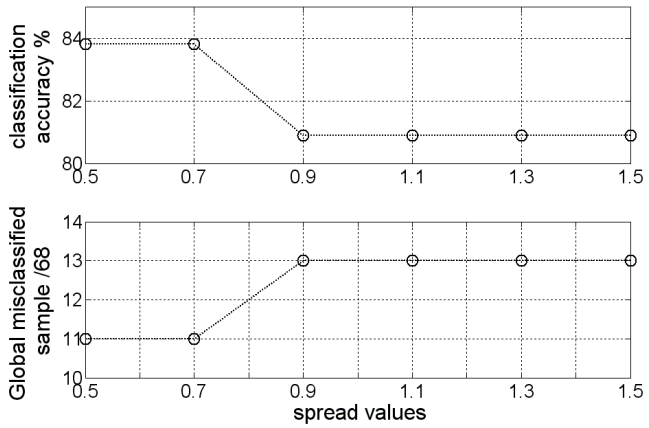

Figure 6.30: Average accuracy according to different spread values of RBF network. (For $M2$ time-frequency domain feature)

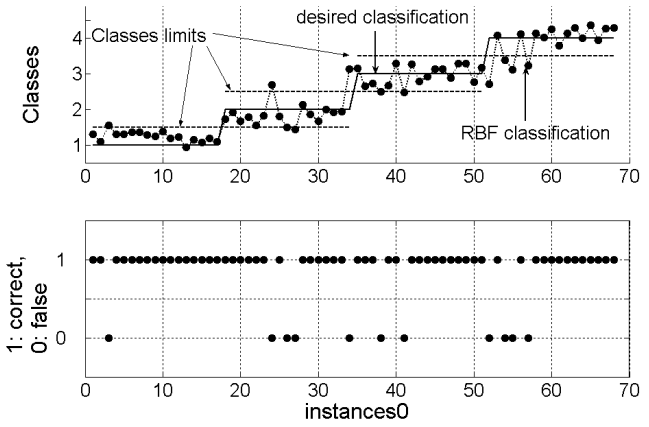

Figure 6.31: Misclassified samples of RBF-classifier-model

In the second class, pointer finger flexion, there are "2" misclassified sample, which are classified as thumb-finger movement, and "2" samples as middle-finger flexion (76.47% correct). The third class, middle-finger flexion, has "2" misclassified samples, or 88.23 % correct, which are classified as pointer-finger movement. The last class, hand closing, has only "4" misclassified samples, or 76.47% correct, which are classified as middle finger movement. All these results are resumed in the following figure 6.32.

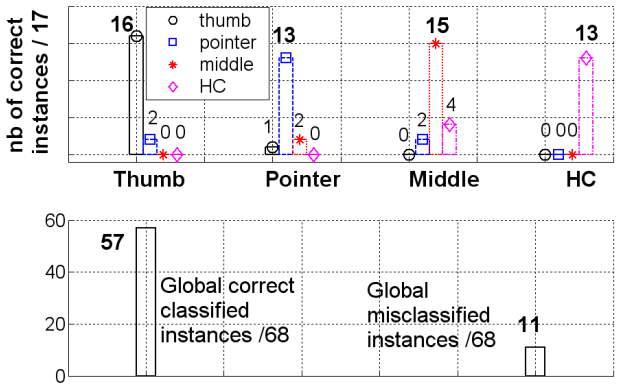

Figure 6.32: *RBF* classifier-model: Correct- and mis-classified samples for each class and the classification-interferences between them

6.5.4 Learning Vector Quantization classifier-model

Learning Vector Quantization [37] [38] networks can classify any set of input vectors like non linearly separable sets of input vectors. Its architecture resembles to that of unsupervised competitive learning network, except that each output is assigned to a target class and works in two steps. In first step it uses an unsupervised data clustering method to locate several clusters. In second step it optimises the cluster

centres. The number of clusters can be specified *a priori* or determined via cluster techniques. It is able to reduce large data sets to a smaller number of codebook vectors (cluster centres) suitable for data compressing. LVQ network used in this work has four neurons in the first competitive layer and one neuron for each class in the second linear layer. After 15 training epochs, the optimal number of neurons in competitive layer found is 12 or 17 neurons, figure 6.33. . The value of 12 neurones

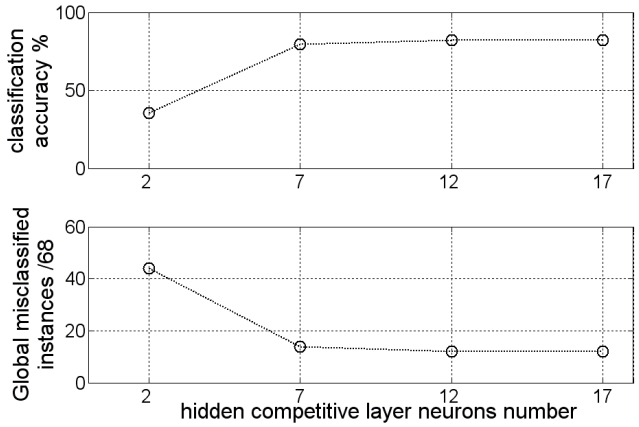

Figure 6.33: LVQ classifier-model: Average accuracy according to competitive neurones number For M_2 time-frequency domain feature

is chosen to built our LVQ classifier-model, then this model is optimised during 100 epochs. This optimisation leads to the following classification results, figure 6.34 . In the first class, thumb-finger flexion, there is no misclassified samples, or 100% correct. In the second class, pointer-finger flexion, there are "5" misclassified samples, which are classified as thumb-finger movement, and "1" sample as middle-finger movement (64.70% correct). The third class, middle-finger flexion, has no misclassified samples (100 % correct). The last class, hand closing, has only "2" misclassified samples, or 88.23% correct, which are classified as middle-finger movement. All these results are resumed in the following figure 6.35.

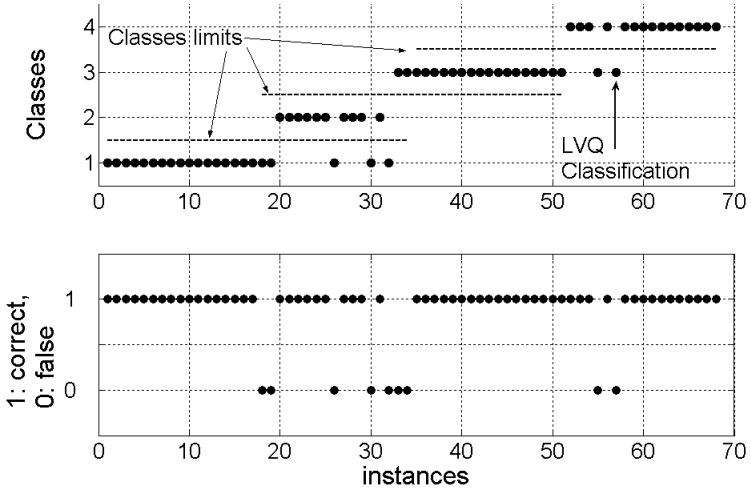

Figure 6.34: Misclassified samples, of optimised *LVQ* classifier-model during 100 epochs

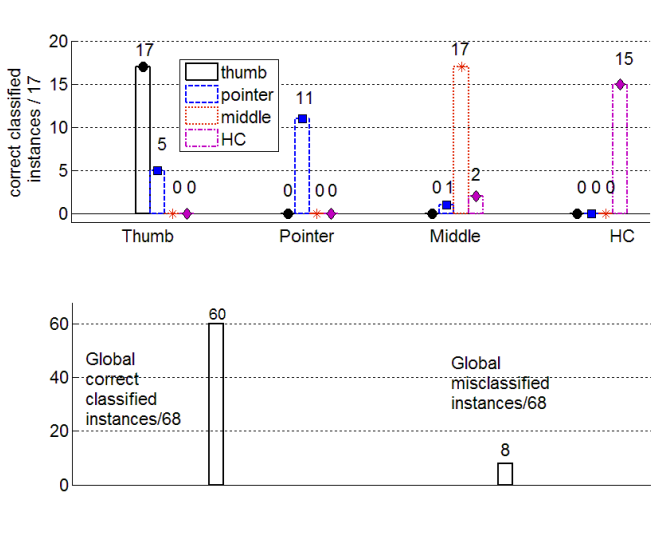

Figure 6.35: *LVQ* classifier-model: Correct- and mis-classified samples for each class and the classification-interferences between them

6.5.5 Classification accuracy comparison

In this section the previous four studied intelligent computational classification methods, MLP, RBF and LVQ, with the proposed supervised $FTMC$ classification algorithm will be compared. In this case the extracted $M2$ time-frequency feature is used. Top figure shows the methods results comparison for each class of hand movement classification. Bottom figure shows methods performance comparison for global classification. Following figure 6.36 resumes all previous results. It's important to remember that this proposed approach has needed only 4 training epochs in comparison with 100 training epochs for other methods.

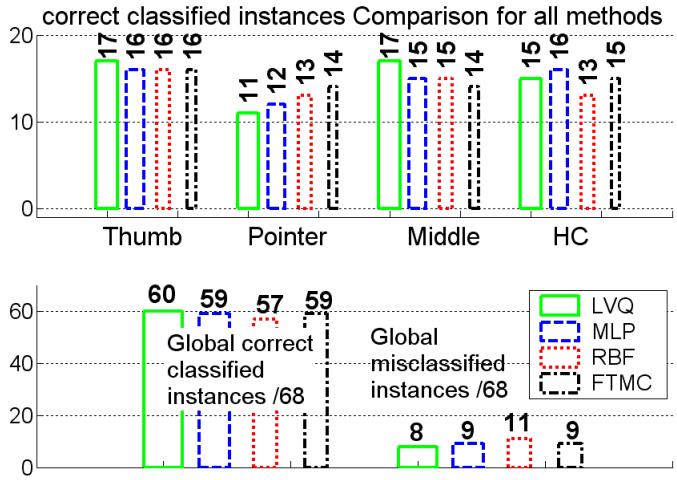

Figure 6.36: Classification accuracy comparison of MLP, RBF, LVQ and proposed $FTMC$ algorithm with extracted $M2$ time-frequency feature.

6.5.6 Conclusion

In case of dynamical complex systems, like forearm EMG signals recognition, intelligent computational methods show their efficiency to deal with such complex systems

and give good classification-models.

The strategy to choose between different classification methods is of great importance. In case of on-line prosthesis control or exoskeleton devices control, the needed time for signal acquisition, then for processing till decision control should be short. Therefor the choice decision of this method to use for classification should considers both time consuming and performance. For these considerations this fuzzy $FTMC$ classifier-algorithm is proposed for such application like surface EMG signals classification. As it is proved in this section 6.5, this algorithm presents acceptable results. Its advantage can be seen in optimisation methods, which are simple and not time consuming, like Gradient Descent (GD) and Least Squared Error (LSE), and also utilisation of simple Trimmed Mean method for determination of initial input fuzzy sets (initial classifier-model structure).

Chapter 7

Influence evaluation of important parameters

7.1 Introduction

After completing the Study of EMG signal processing phases and classification procedures, this first part of thesis will be finalised by some different real application studies. This chapter will consider the influence or the effect of important parameters on classification accuracy. The measurement of surface EMG signals is depending on different factors. Hence the recognition of these signals corresponding to their muscle dynamics has to take in consideration the following factors, which will be studied in details:

a) **Effect of filter frequency band:** different filter frequency bands will be applied to test their effect on classification performances.

b) **Effect of noise threshold level:** different noise threshold levels based on noise's standard deviation will be tested and compared

c) Effect of EMG signal length and sampling frequency: different sampling frequencies will be tested also in combination with different EMG signal lengths.

For an available conclusions, different classification methods are considered, three methods, and also seven different features are used. Taking in consideration the results studied in section 4.4. These features are those extracted from EMG signals time-frequency analysis:

- zero order moment ($M0$).
- first order moment ($M1$).
- second order moment ($M2$).
- Central frequency ($Fcnt$).
- Frequency variance ($Fvar$).
- Frequency standard deviation ($Fstd$).
- Energy of signal (Eng).

Different intelligent computational classification methods are tested to get also more available results. These methods, described in chapter 6 are: Radial Basis Function Networks (RBF), Fuzzy Subtractive Clustering (FSC) and proposed fuzzy trimmed mean classifier-algorithm ($FTMC$).

The reason that incited us to use these different classification methods is to confirm if the influence of each parameter is depending on classification-method or not. If the influence of a parameter changes between these methods, then it's not possible to get a conclusion about the influence of this parameter. Otherwise if this influence of a parameter is the same for all classification methods, in this case it's possible for us to conclude which type of influence has this parameter. Before the beginning of mathematical study, it is preferable, as we believe, to show and to explain these

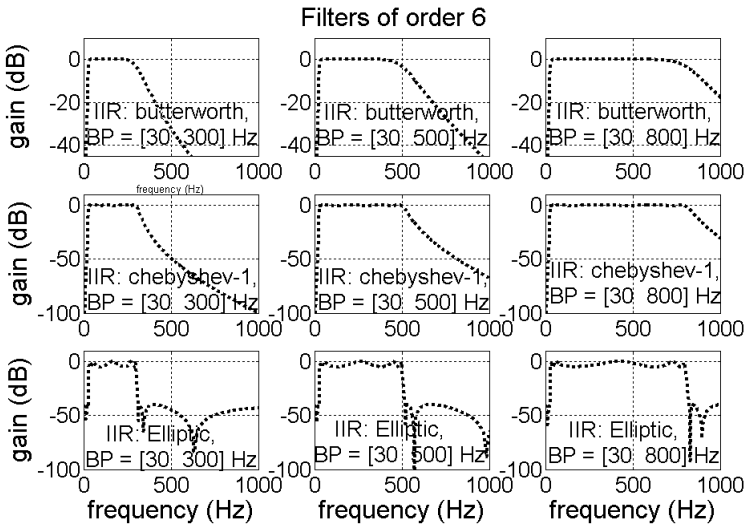

Figure 7.1: *Butterworth-*, *chebyshev-1-* and *Elliptic*-filter frequency bands.

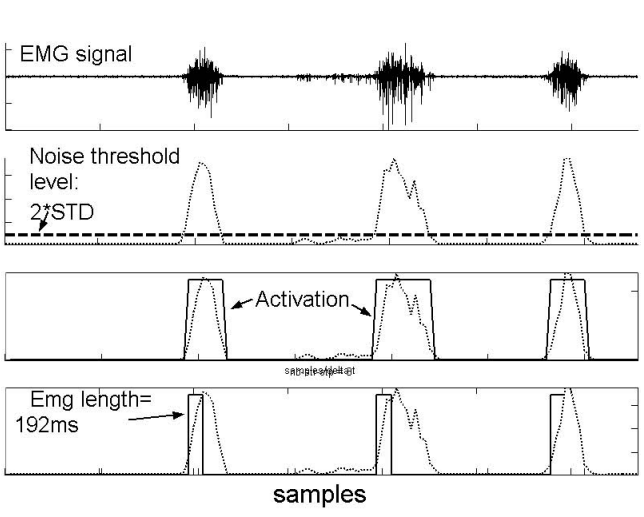

Figure 7.2: Noise threshold level and EMG signal length.

parameter effects using only schematic procedure. Figure 7.1 presents three filter frequency bands using butterworth, *chebyshev-1* and *Elliptic* filters. The two following parameters: noise threshold level and EMG signal length, are presented in figure 7.2

7.2 Frequency Bandpass effect

The study of signal filtering to remove undesired signals, which contribute to drown the important information in the signal, is described in section 4.3.6. The choice of the filter type and its order is based on the study did also in section 4.3.6. The recognition of this signal is depending on its information contents, if this information is kipped inside the signal after filtering, then the signal recognition accuracy increases, otherwise it will be worse. The objective of this section is to show the effect of filtering and to get the tools of the best filtering strategy. In the literature, it is known that the filter frequency band to eliminate undesirable noise frequencies is given in the range of *20Hz* and *500Hz*. Through the study of this section, we will have the possibility to conclude if this pass-band frequency is always useful.

Three different frequency pass-bands, *10-300Hz*, *10-500Hz* and *10-800Hz*, are selected to be tested for all our seven features described above. These seven different features are divided also in three different groups according to their common behaviours regarding these frequency-bands. The first group is composed of the three first features: *M0, M1 and M2*. The second group is composed of the next three features: *Fcnt, Fvar and Fstd*. The last group is presented with only one feature: *Eng*. The performances of each group are calculated for all the three frequency-bands defined above. In this way it will be possible to look for the frequency-band for each feature group, which gives the best classification performance.

7.2.1 Classification performance with *RBF*-based approach

Using *RBF* classification method, figure 7.3, it can be seen that the first feature group (*M0, M1 and M2*) has good global classification performance in the frequency-band of *10-300Hz*. For the second features group (*Fcnt, Fvar and Fstd*) the frequency-band of *10-800Hz* gives the best global classification performance. The global classification performance of third feature group, Eng, seems unsensible to the frequency-band, it gives almost the same performances for these three frequency bands. These results are obtained in the case of four different spread values between 0.4 and 1.6 with a step of 0.4 for these seven features.

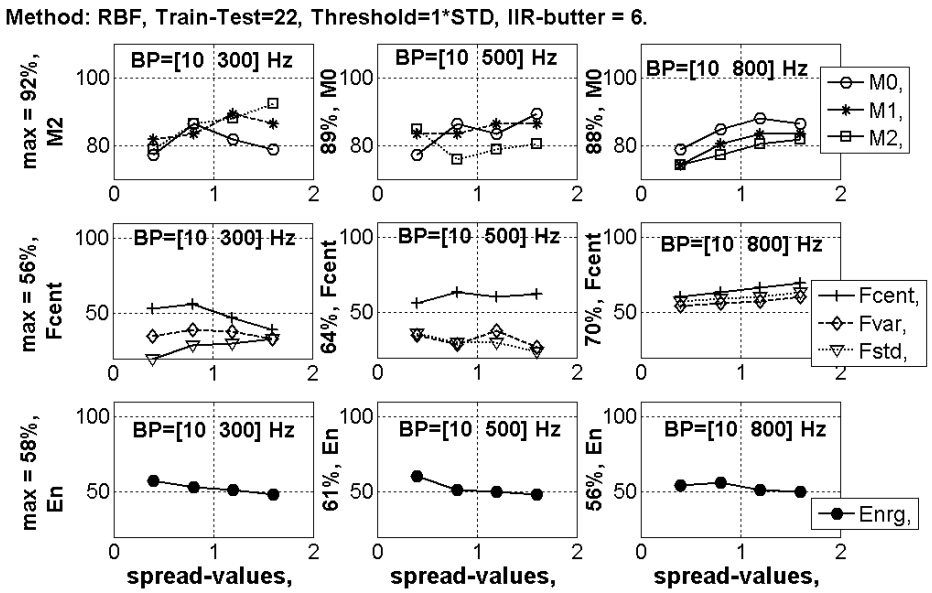

Figure 7.3: Filter frequency Pass-band effect on classification performance using *RBF* classification method for flexion movements of three fingers: Thumb, pointer and middle.

7.2.2 Classification performance with *FSC*-based approach

Like with *RBF* method, *FSC* classification method, figure 7.4, has almost the same results. The first group-feature (*M0, M1 and M2*) has best classification performance in the frequency-band of *10-300Hz*. The best classification performance for the second group-feature (*Fcnt, Fvar and Fstd*) is given by frequency-band of *10-800Hz*. The classification performance of third feature group, *Eng*, seems unsensible to the frequency-band, it gives almost the same performances for these three frequency bands. These results are obtained in the case of four different radius values between 0.2 and 0.8 with a step of 0.2 for these seven features.

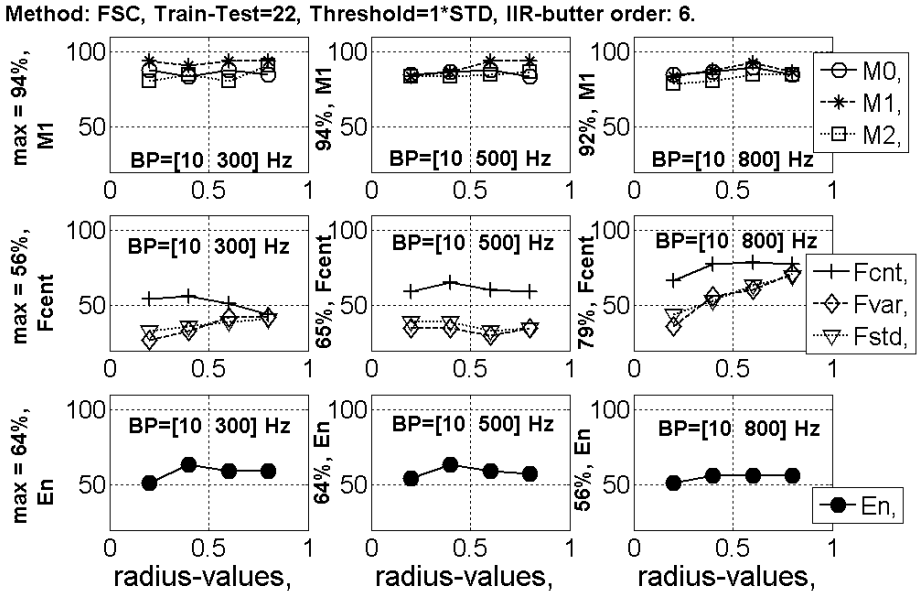

Figure 7.4: Filter frequency Pass-band effect on classification performance using *FSC* classification method for flexion movements of three fingers: Thumb, pointer and middle.

7.2.3 Classification performance with *FTMC* algorithm

Proposed algorithm *FTMC* gives, Like with *RBF* and *FSC* methods, the same results of classification performances, which are shown in figure 7.5. For the first feature group (*M0, M1 and M2*) the best classification performance has been found in the frequency-band of *10-300Hz*. The best classification performance for the second feature group (*Fcnt, Fvar and Fstd*) is given by frequency-band of *10-800Hz*. The classification performance of third feature group, *Eng*, seems unsensible to the frequency-band, it gives almost the same performances for these three frequency bands. These results are obtained in the case of four different trimming percentage, β, values between 0.64 and 0.94 with a step of 0.1 for these seven features.

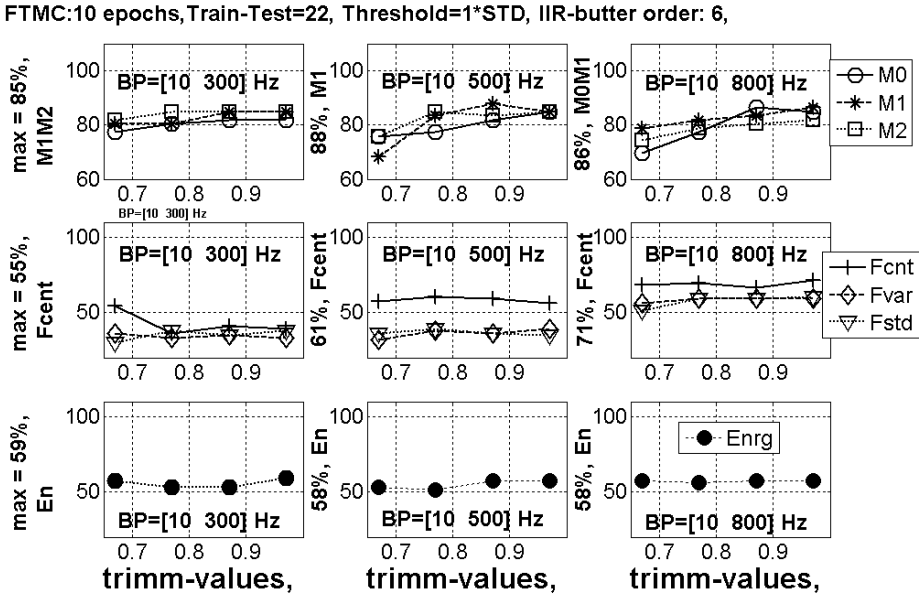

Figure 7.5: Filter frequency Pass-band effect on classification performance using proposed *FTMC* classification algorithm for flexion movements of three fingers: Thumb, pointer and middle.

7.2.4 Conclusion

It is possible for us to conclude now, after these almost same results found with these three different classification methods and seven different features, that the optimal frequency band is depending on the nature of the feature. The study in this section using three different classification methods and seven features gives the optimal frequency band *10-300Hz* for the first group (*M0, M1 and M2*). For the second group *Fcnt, Fvar and Fstd*, optimal frequency band is *10-800Hz*. Finally for the third feature group, *Eng*, classification accuracy it's almost the same for all three frequency bands.

7.3 Threshold level effect

What is the effect of the noise's threshold level on the performance of EMG signal classification (recognition)?. This question had generated intensive discussion between researchers. This parameter is very important because it represents the first step of EMG signal processing, see section 4.2. The beginning time of EMG signal activation is depending on this reference level, therefor this parameter is considered as very sensible factor. In this section three different values of this parameter are tested, which are relative to the value of noise's standard-deviation (*Std*). If the noise's standard-deviation value is given as $Threshold_{std}$, then these values are given as: $0.5Threshold_{std}$, $Threshold_{std}$ and $2Threshold_{std}$.

Our seven different features are divided in three different groups according to their common behaviours regarding these values, see section above. The choice of value of frequency-bands for each feature group is defined using the optimal result found in the preceding section 7.3. These frequency-bands are: *10-300Hz* for feature group

(*M0, M1 and M2*), *10-800Hz* for feature group (*Fcnt, Fvar and Fstd*), and *10-500Hz* for feature *Eng*.

7.3.1 Classification performance with *RBF*-based approach

Using *RBF* classification method, figure 7.6, it can be seen, that for the first features group (*M0, M1 and M2*) filtered in frequency band of *10-300Hz* has good classification performance for the noise's threshold level equal to $2 \cdot Threshold_{std}$. The second

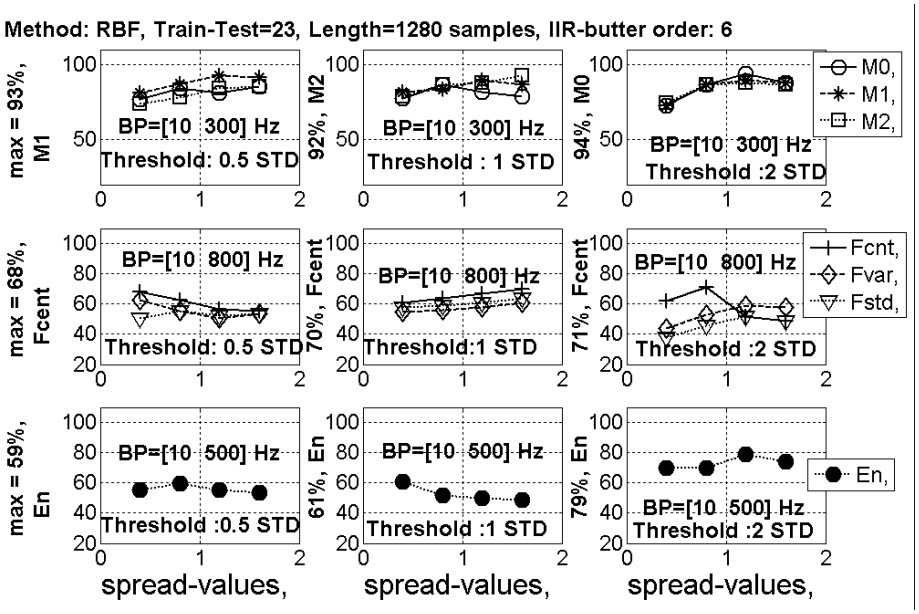

Figure 7.6: Noise threshold level effect on classification performance using *RBF* classification method for flexion movements of three fingers: Thumb, pointer and middle.

feature group (*Fcnt, Fvar and Fstd*) filtered in frequency band of *10-800 Hz* has good classification performance with noise's threshold level equal to $1 \cdot Threshold_{std}$. Finally the classification performance of third feature group, *Eng*, filtered in frequency band of *10-500Hz* get good classification performance for the noise's threshold level equal to $2 \cdot Threshold_{std}$). These results are obtained in the case of four different

spread values of *RBF* membership functions between 0.4 and 1.6 with a step of 0.4.

7.3.2 Classification performance with *FSC*-based approach

This method gives the same results, see figure 7.7, as with *RBF* method described above in section 7.3.1. For the first group (*M0, M1 and M2*) filtered in frequency band

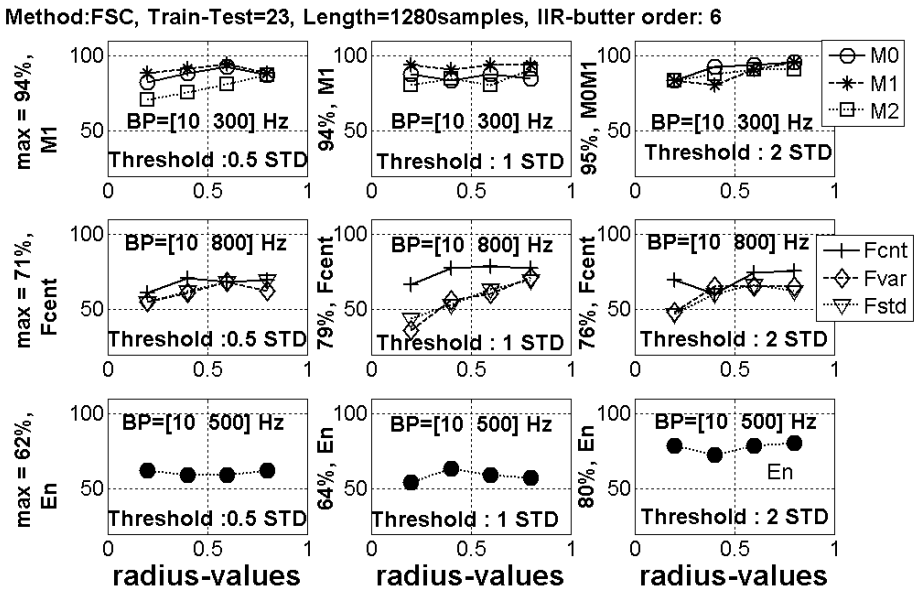

Figure 7.7: Noise threshold level effect on classification performance using *FSC* classification method for flexion movements of three fingers: Thumb, pointer and middle.

of *10-300Hz*, the classification is globally performed for the threshold level equal to $2 \cdot Threshold_{std}$. The second features group: *Fcnt, Fvar and Fstd*, filtered in frequency band of *10-800Hz* has good classification performance for the threshold level equal to $1 \cdot Threshold_{std}$. Finally the classification performance of the third feature group, *Eng*, filtered in frequency band of *10-500Hz* get good classification performance for the threshold level equal to $2 \cdot Threshold_{std}$. These results are obtained in the case of four different radius values of *FSC* clusters between 0.2 and 0.8 with a step of 0.2.

7.3.3 Classification performance with *FTMC* algorithm

With this proposed classification model the results, figure 7.8, are the same with those described above using *RBF* and *FSC* methods. These results are obtained in the case

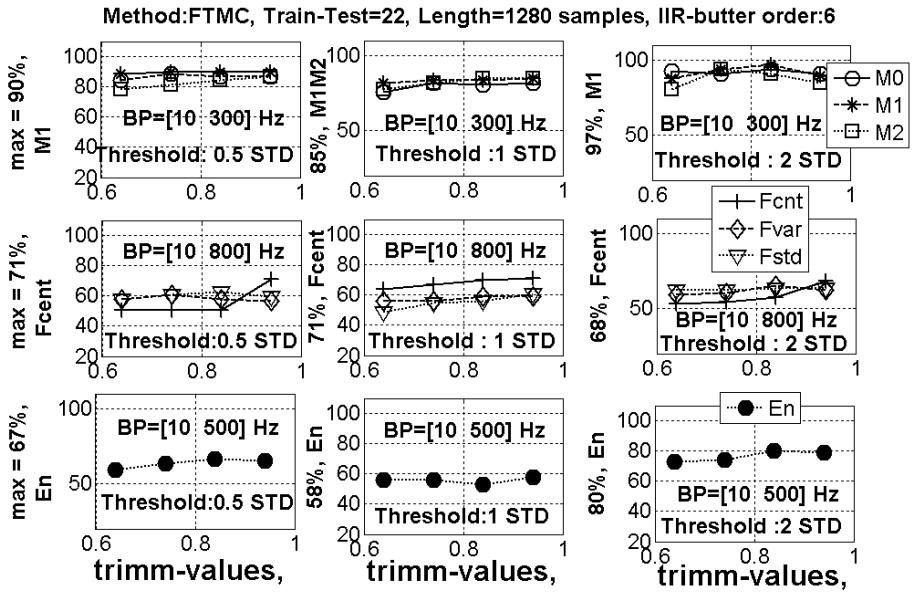

Figure 7.8: Noise threshold level effect on classification performance using proposed *FTMC* classification method for flexion movements of three fingers: Thumb, pointer and middle.

of four different trimming percentage, β, values of *FTMC* ellipses between 0.64 and 0.94 with a step of 0.1.

7.3.4 Conclusion

Noise threshold level parameter is a decisive factor for EMG signals classification. It is difficult to give an optimal value. There are some developed methods, in the literature, which tried to optimise this parameter. The study of this section using three different classification methods and seven features led to the value of noise's

threshold level equal to $2Threshold_{std}$ for the first group (*M0, M1 and M2*) and the third feature group, *Eng*. For the second group *Fcnt, Fvar and Fstd*, both noise threshold values $1 \cdot Threshold_{std}$ and $2 \cdot Threshold_{std}$ are optimal.

7.4 Both signal length and sampling frequency effects

This study, which considers the effect of both EMG signal length and sampling frequency parameters on neuromuscular signals recognition, is of a great importance. It helps to optimise the choice of these parameter values for the best performances and in the same time for the shorter online processing-time.

The considered EMG signal length has a significant importance for the classification performance for the all EMG signal. If this length is of small duration, it will be possible for us to talk about EMG signal prediction. In this case the task becomes prediction and classification. The class of this all EMG activation signal is predicted only from the beginning signal part. Three different values, 512 samples (*128ms*), 1024 samples (*256ms*) and 1536 samples (*384 ms*) are considered. It is known that if the considered signal length part is bigger, then the classification performance is better, but the researchers in this domain are limited this length because of online-processing consideration. We can say, based on our study in this section, that this monotone positive influence of signal length is not always verified if we take additionally in consideration the sampling frequency parameter. Three different Sampling frequency parameters are used, which are: *1kHz, 2kHz and 4kHz*. Two feature groups are considered also, which are: *M0, M1 and M2*, filtered with frequency band of *10-300Hz*, and *Eng* filtered with frequency band of *10-500Hz*. The value of noise baseline,

see above study in section 7.3, is considered equal to two times noise's threshold value ($2Threshold_{std}$). Three classification methods are applied for each feature group, and the results are regrouped together in one figure for each method.

7.4.1 Classification performance with *RBF*-based approach

a) Case of feature-group *M0, M1 and M2*

Using *RBF* classification method it can be seen in figure 7.9, for the first feature-group (*M0, M1 and M2*) filtered in frequency band *10-300Hz* with noise baseline equal to 2 $Threshold_{std}$, that the classification performance does not increase proportionally with the increasing in signal length. If the first frequency sampling value *1kHz* is

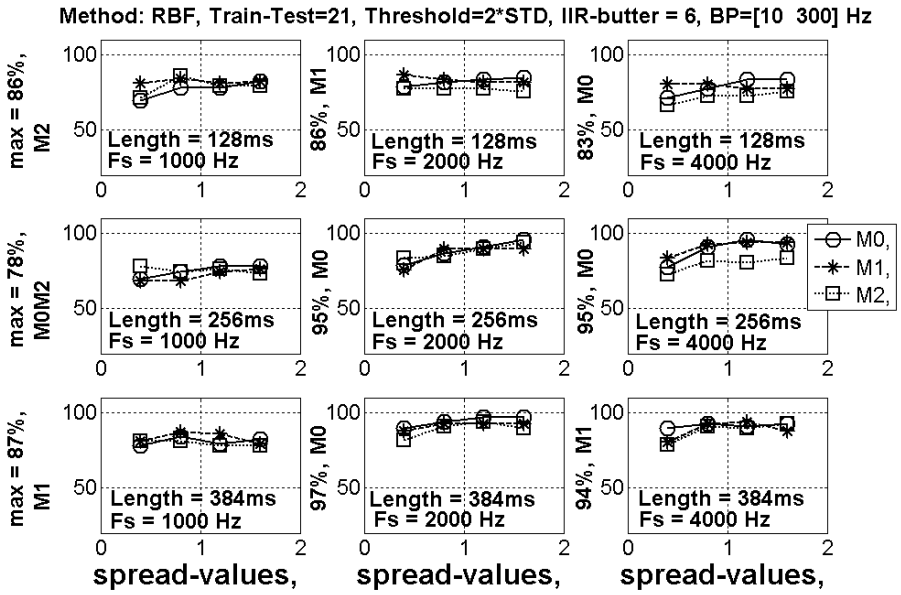

Figure 7.9: Sampling frequency and signal length effect on classification performance in case of *M0, M1 and M2* features using *RBF* classification method.

considered, the classification performance decreases when the signal length increases from the first length 512 samples (*128ms*) to the second length 1024 samples (*256ms*),

and then this performance increases for the third value of 2048 samples (*512ms*). In this case, for *1kHz* sampling frequency, the optimal and the practical value for online-processing is the value of 512 samples (*128ms*). These two parameters, *1kHz* sampling frequency and *128ms* signal length, give almost the same performance if we consider the values of 1536 samples (*384ms*) for signal length and *1kHz* for sampling frequency, which require four times more of processing-time.

For the second and third frequency sampling values, *2kHz* and *4kHz* figure 7.9, the classification performance is proportionally increasing with the signal length. But for both cases, *2kHz* and *4kHz* frequency sampling values, the classification performances are almost the same for signal length of 1024 samples (*256ms*) and 1536 samples (*384ms*). From these results it's possible now to choose the optimal values of these two parameters, which are: *256ms* for signal length and *2kHz* for sampling frequency. These values satisfy both good classification performance and reduced processing-time. The classification performance is equal to 95%, with feature *M0*, in case of *RBF* spread value equal to 1.6.

b) Case of *Eng* feature

For the second feature-group, *Eng*, filtered in frequency band of *10-500Hz*, and with noise baseline equal to 2 $Threshold_{std}$, the classification performance does not increase proportionally with the increasing in signal length. From the results shown in figure 7.10, it's possible to choose the optimal values of these two parameters: *128ms* for signal length and *2kHz* for sampling frequency, which satisfy both good classification performance and reduced processing-time, relatively to all other values of these two parameters. The classification performance is equal 79% max.

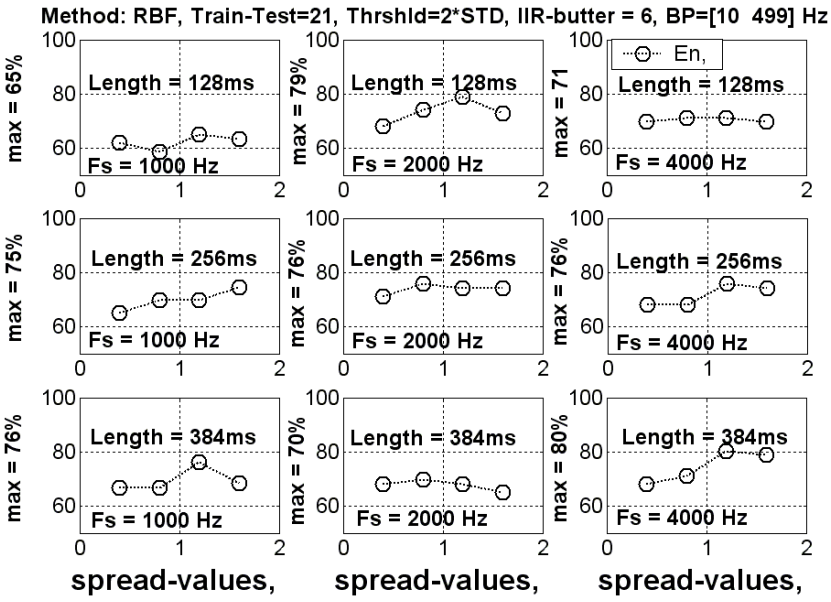

Figure 7.10: Sampling frequency and signal length effect on classification performance in case of *Eng* feature, using *RBF* classification method.

7.4.2 Classification performance with *FSC*-based approach

a) Case of feature-group *M0, M1 and M2*

From the results shown in figure 7.11, and using the same procedure, see above study section 7.4.1.a, it's possible to choose the optimal values of these two parameters, which are: *256ms* for signal length and *2kHz* for sampling frequency. These values satisfy both good classification performances and reduced processing-time. The classification performance is equal to 97% max, with feature *M0*, in case of *FSC* radius value equal to 0.8, and reduced processing-time.

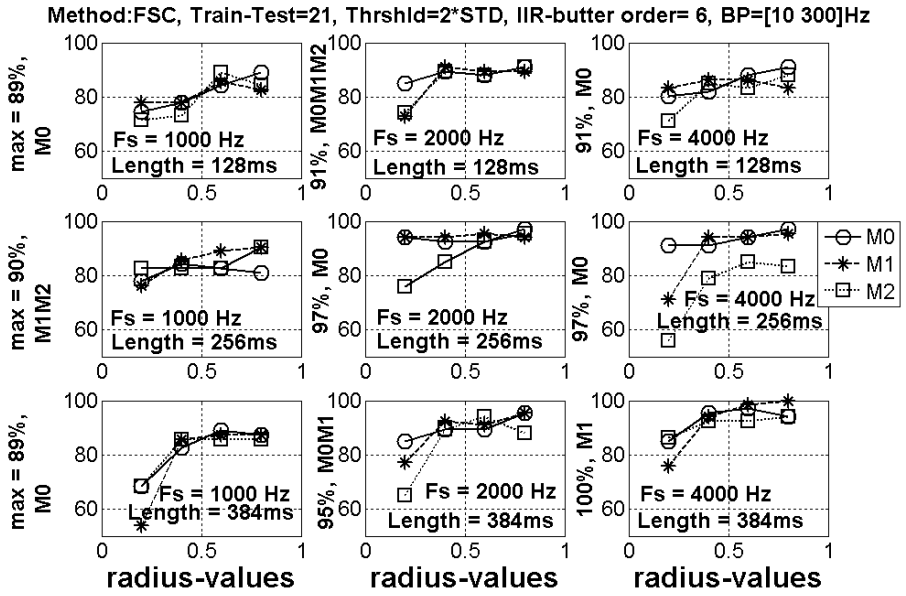

Figure 7.11: Sampling frequency and signal length effect on classification performance in case of frequency moment features using *FSC* classification method.

b) Case of *Eng* feature

Also from the results shown in figure 7.12, and using the same procedure, see above study section 7.4.1.b, it's possible to choose the optimal values of these two parameters, which are: *128ms* for signal length and *2kHz* for sampling frequency. These values satisfy both good classification performances, which is equal to 80% max, in case of *FSC* radius value equal to 0.6, and reduced processing-time.

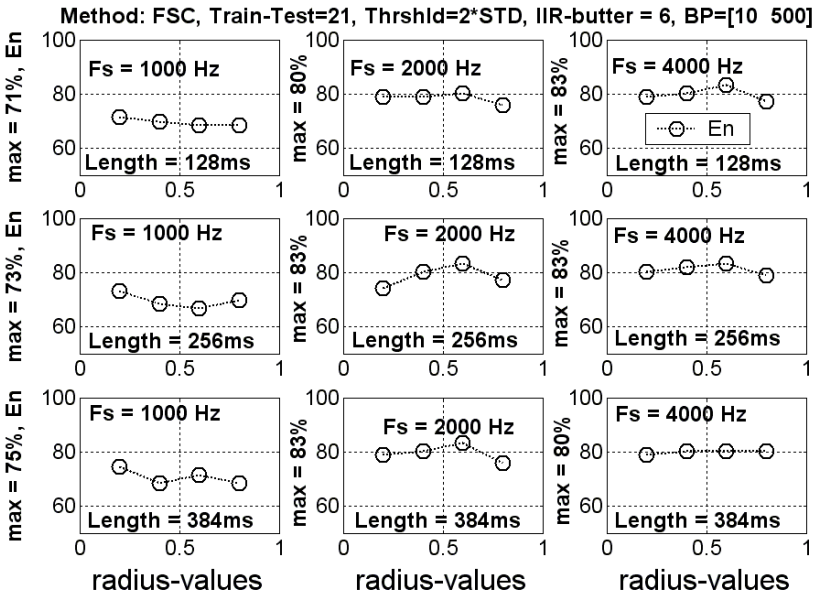

Figure 7.12: Sampling frequency and signal length effect on classification performance in case of *Eng* feature, using *FSC* classification method.

7.4.3 Classification performance with *FTMC* algorithm

a) Case of feature-group *M0, M1 and M2*

From the results shown in figure 7.13, and using the same procedure, see above study section 7.4.1.a, it's possible to choose the optimal values of these two parameters, which are: *256ms* for signal length and *2kHz* for sampling frequency. These values satisfy both good classification performance and reduced processing-time. The classification accuracy is equal to 95%, with feature *M1*, in case of trimming percentage, β, equal to 0.85,

b) Case of *Eng* feature

In the same way, from the results shown in figure 7.14, and using the same procedure, see above study section 7.4.1.b, it's possible to choose the optimal values of these

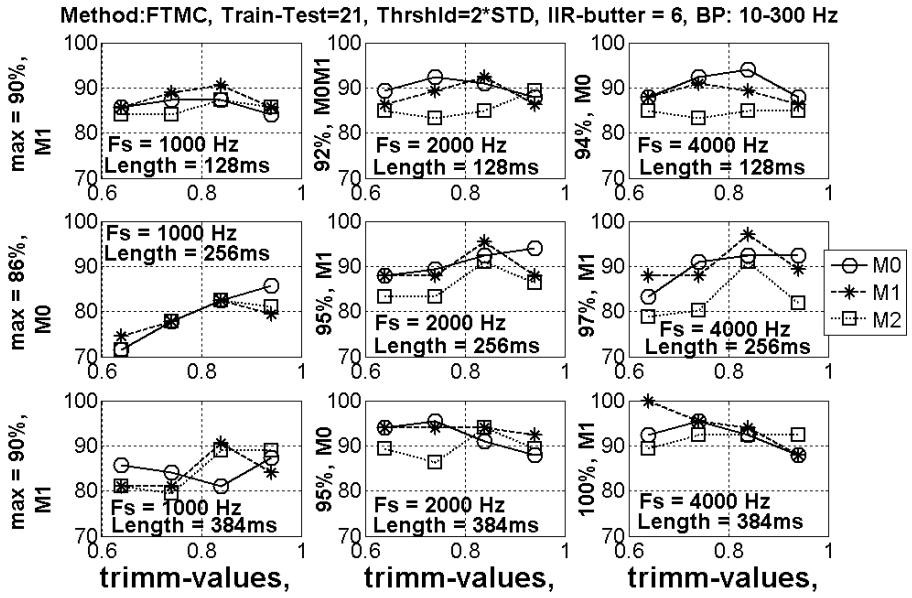

Figure 7.13: Sampling frequency and signal length effect on classification performance in case of frequency moment features using *FTMC* classification method.

two parameters, which are: *128ms* for signal length and *2kHz* for sampling frequency. These values satisfy both good classification performances, which is equal to 82% max, in case of trimming percentage, β equal to 0.94, and reduced processing-time.

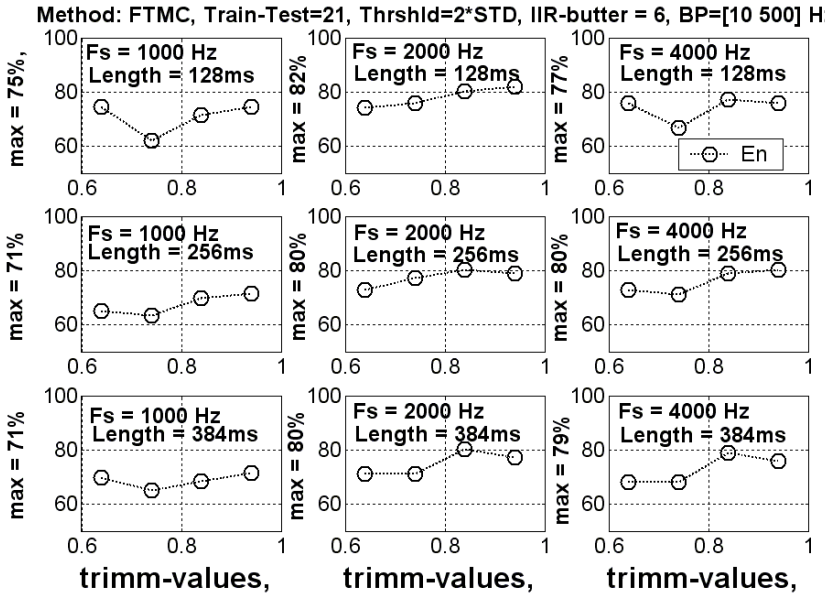

Figure 7.14: Sampling frequency and signal length effect on classification performance in case of *Eng* feature, using *FTMC* classification method.

7.4.4 Conclusion

The choice of EMG signal length and Sampling frequency values are of a great importance for the optimisation of processing-time consummation and for the classification performances. As it has been shown in this study, it's possible with only a part of activation EMG signal length to predict and recognise the all activation signal. Moreover, for some features, a shorter part of activation EMG signal length can predict and classify better than with longer one. Also about frequency sampling, smaller values can give better results. Because the nature of features present different behaviors we should not use the same values of these parameters for all features.

7.5 Conclusion

The objective of this Chapter was focused on the study of important factors that influence the EMG signal representation, the classification accuracy and processing time consuming. For getting able results three different classification methods and seven different features are used. Through this study we could show how these parameters influence EMG signal recognition. The goal was to clarify which parameters are important and how to proceed the choice of the optimal values.

Chapter 8

Musculoskeletal dynamics identification

8.1 Introduction

Whole-body movement is achieved with help of the interaction between the neuromuscular signal control and musculoskeletal dynamics. Measured neuromuscular signals as EMG signals are investigated in the first part of this thesis. It was shown the possibility of hand movements recognition through the classification of these neuromuscular EMG signals. Now, in this second part of thesis, a new study will be initiated. This study concerns the ability of builded musculoskeletal model to capture and identify the highly non-linear dynamics, case of knee-joint dynamics, of human motion as a result of *FES* stimulations. This system has complex nonlinear mechanical properties. The accuracy of computer models depends on experimental data measurement and the power of intelligent computational methods. Our developed fuzzy-identification system from experimental data demonstrates the model's ability to capture the nonlinear time-varying effects observed experimentally in quadriceps

muscles. The examined system consisted of the quadriceps, surface electrical stimulation (FES) signals and a freely swinging shank. The output of the system is the angular position, measured by externally mounted sensors. The gaol is identification model of the active quadriceps dynamics (total quadriceps-shank system) and the *FES*-induced movement (angle prediction). In few words, the task is resumed in building computer model of the musculoskeletal system that can reproduce the musculoskeletal dynamics response of electrical stimulation (*FES*) with good precision.
Dynamical systems are complex systems like in human movement functions and other real systems. They are described by system variables whose values at the next time step cannot be predict with complete certainty. Building models of real dynamical systems needs suitable fidelity to describe and identify the system components and their interactions with the environment. These models should adequately mimic the discrete motor unit structure of quadriceps muscles. Quantitative studies of musculoskeletal dynamics may be divided in two types:

1) Morphological models: the muscle is represented as one single component with viscoelastic properties [8]. The well-controlled phenomenological studies of isolated muscle tissue realised by Hill is known as hill-model. This classical model consists of contractile element (*CE*) and the series viscoelastic element (*VE*). The component (*CE*) has no effect in case of muscle extension, its force is depending on the speed of muscle's flexion.

This type of model has a structure which is related to the anatomy and physiology. The weakness of this approach deal with the structure model parameters, which is unique.

2) Models of adaptable parameters: such models are builded on the basis of

overall system behavior [34] and are derived from measured input-output data sets. These models can describe and identify quantitative behaviors of this real system's dynamics (knee-joint dynamics) through the analysis of the relation between measured experimental input-output data sets. However Process identification based upon conventional mathematical models e.g., linear or nonlinear differential equations, are not well suited for dealing with ill-defined, complex and uncertain systems [25]. This study considers this second type of models. For this task an efficient fuzzy identification model is developed see our publication [72]

8.2 Foundations and Methods

The research in system identification covers a wide applications, system identification is the procedure of making mathematical models of systems starting from experimental data, measurements, and observations. The model of a system is often very important for analysis, control design, simulation and prediction. However for the real complex systems, like Biomedical systems in our life, it's not easy to find always models. In studying musculoskeletal system in humans, researchers must rely on comprehensive mathematical models representing the system of interest for simulating behaviors which would only be experimentally observable. The builded models must be compatible to the signals flow between the input and the output. In the literature there are a number of developed models to help the building of musculoskeletal dynamics. The goal is to use the predictive ability of each model to further our understanding of how these systems work.

Mathematics serves as the language with which we try to understand how nature

works. Many systems are modeled with a continuous time variable t. The continuous systems can be described with differential equations which in its simplest form looks like: $\acute{x}(t) = f(t, x)$. However a physical setting is often reduced to a set of discrete measurements. In discrete-systems these measurements are given at a sequence of specific times. Specific values of the measurements of the system are often called the state of this dynamical system, the state space is the unit interval. When a system is qualified as a linear system, it is possible to use the responses of a small set of inputs to predict the response to any possible input. To see whether a system is linear, we need to test whether it obeys certain rules of linear systems. The two basic tests of linearity are homogeneity and additivity and Shift-invariance. Homogeneity of a system doesn't produce or cause response compression or response expansion. Most of real life problems involve nonlinear systems. However it is possible to approximate a nonlinear system by a linear one. This is called linearisation of nonlinear systems. The main idea of linearisation is to approximate a nonlinear system by a linear one around the equilibrium point. System dynamics can change significantly with a change in the system operating conditions. A system can be represented with multiple local models, each local model is valid for a specific operating region. Local model can be a set of some basis activation functions, as in figure 8.1, in case of two dimensional 9 local models using basis Gaussian activation functions. There are two components to identify a model: the structure and the parameters. The structure is the number of local models, the centres of their activation functions and their widths. The local model's parameters could be the complete set of coefficients for these local models.

Many parameter estimation algorithms are available to identify the parameters of

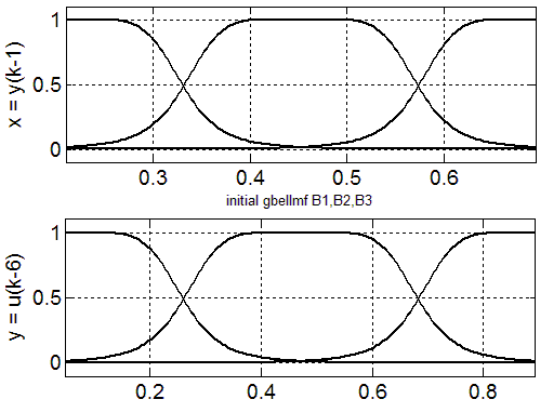

Figure 8.1: Two dimension 9 local models with Gaussian activation functions.

these local models. The models themselves can be black box systems (neural networks) or white box (fuzzy networks) systems, without the need for explicit higher order derivative function forms. This type of models are applicable in a large variety of contexts, including very complex models.

8.3 Experimental setup and procedure

Experimental measurements and procedures are done in Max Planck Institute of Magdeburg, Germany. A short description of this experimental procedure, figure 8.2, is given using the measurement protocol from this Laboratory. Surface stimulation electrodes are placed on the thigh, the quadriceps group of muscles form the major part of the muscles on the front of the thigh. The knee angle θ (system output) between thigh and shank is sampled at 20Hz, and measured using an electrogoniometer. Input signal is the variable pulse width of the mono-phasic current pulses with fixed frequency, 20Hz, and fixed Amplitude. The range of θ angle variation between 90^o

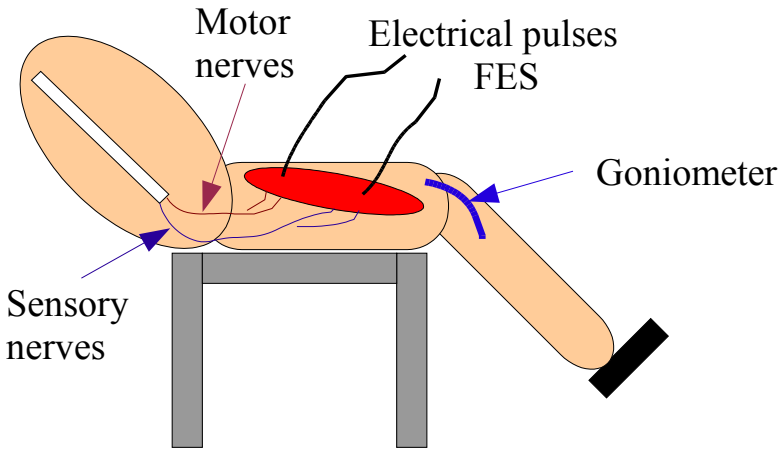

Figure 8.2: Measurement setup.

(Rest position) and 180^o (full extension), is normalised to the range $z = [\ 0,\ 1]$. Many tests are carried out, and classified by the kind of experiments into three experimental sessions. These data are already used for modelling and control using neural networks [60] [59]

1) ***Test-A***: the purpose of this test is to establish a suitable stimulation current **A**mplitude, figure 8.3. The current level I is equal to $40mA$, and stimulation frequency f is equal to $20Hz$. This signal will be used as validation signal.

2) ***Test-SC***: this is an open-loop test signal where the pulse-width possesses specific **S**tochastic **C**haracteristics. The goal is to generate a sequence of input steps which leads to an almost uniform distribution over the interval of the output. We have two measurements of this this signal type, test-SC1 in figure 8.4 and train-SC2 in figure 8.5. The first one will be used as testing signal and the second one will be used as training signal.

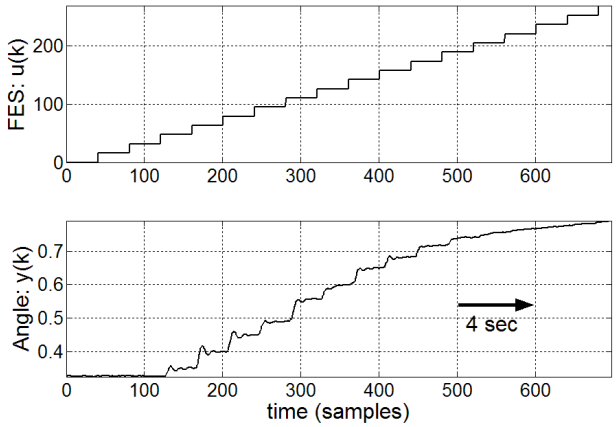

Figure 8.3: Measurement of *Test-A*.

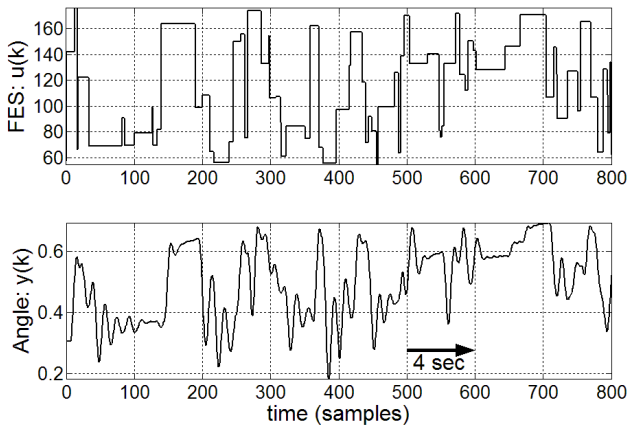

Figure 8.4: Measurement of *Test-SC1* (testing data).

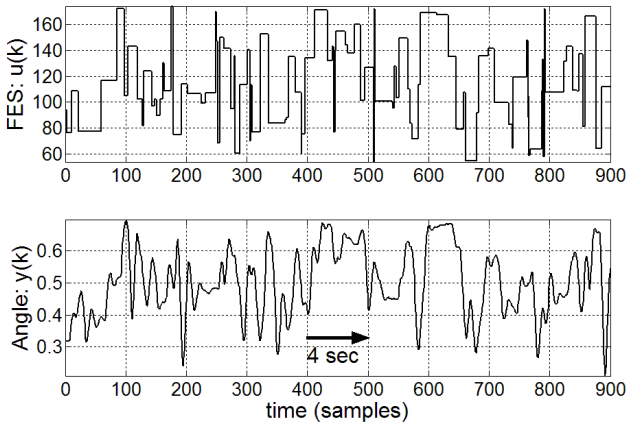

Figure 8.5: Measurement of *Train-SC2* (training data).

3) ***Test-PRBS***: The *PRBS* signal, figure 8.6, is applied around a range of mean stimulation pulsewidth levels. Suitable stimulation mean levels are, $\mu = 0.043$, 0,23, 0.40, 0.59 and 0.82. This signal will be used also as testing signal.

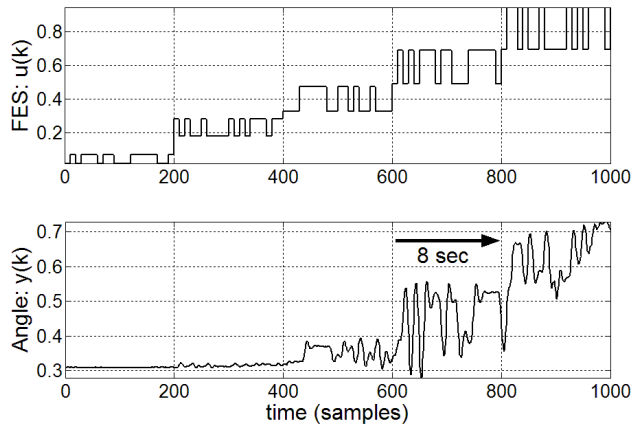

Figure 8.6: Measurement of *Test-PRBS1*.

8.4 Morphological models

In macroscopic models the muscle system is represented as one single component with viscoelastic properties [8]. Hill-Model is one of the earliest most popularly employed muscle model, he attempts to capture the force-length-velocity properties of the muscle in the way to create a mechanical muscle model. This classical model consists of contractile element (CE) and the series viscoelastic element (VE). This type of model has a structure which is related to the anatomy and physiology. The most basic building bloc comprising a muscle model is the ideal spring in serial form with a damper. A linear spring creates a force proportional to its deflection: $F_k = Kx$, where F: is the force, x: the deflexion, and K: the elasticity. An ideal damper creates a force proportional to its viscosity, $F_b = B\dot{x}$, where B: is the viscous damping constant. The response of physiological tissues to a constant force shows both of delay time response and a slower length evolution, which gradually approaches an asymptotic value, figure 8.7. The muscle-joint structure can then be treated as a second (or a

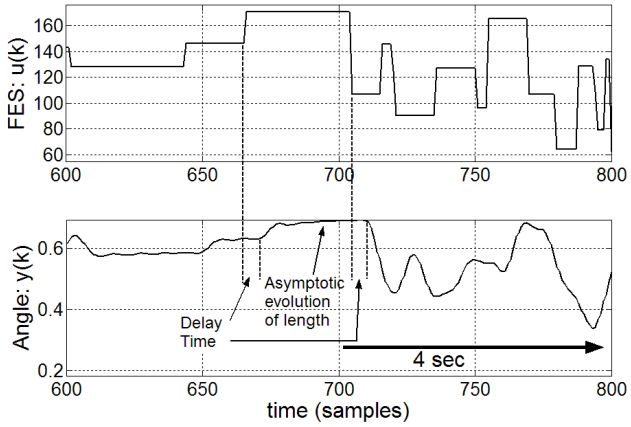

Figure 8.7: Real illustration of both delay-time and asymptotic length evolution of a muscle in response to force excitation.

third) order system. Although a second order linear model can be mathematically represented in several ways,t he basic equation of motion for a second order model, with laplace transfer function is given as:

$$H(s) = a.\frac{\omega_n^2}{s^2 + 2s\xi\omega_n + \omega_n^2} \tag{8.4.1}$$

where a: a constant, ξ:damper factor, and ω_n: proper system pulsation.

In other form this equation can be formulated as:

$$H(s) = \frac{1}{s^2\frac{1}{a\omega^{2n}} + s\frac{2\xi}{a\omega_n} + \frac{1}{a}} \tag{8.4.2}$$

In analogy form with linear spring, the model parameters will take the following values:

The masse: M=$\frac{1}{a\omega^{2n}}$, the viscous element B=$\frac{2\xi}{a\omega_n}$, and the elastic element K=$\frac{1}{a}$.

If the viscous element B is considered constant, this model will be linear system of second order.

This simple model is simulated. The model presents mechanical model of quadriceps muscles, that cause the extension of the knee, as 2 degrees of freedom planar manipulator for the following parameter values as example:

B=4,5[N.s/rad], K=30[N/rad], and M=1kg.

The muscle-model response of an impulse excitation is shown in figure 8.8. Second order linear systems are widely used to represent a variety of dynamic systems and have been used to represent muscle dynamics [70] [10]. This simple and basic modeling approach for predicting stimulated muscle properties provides a valuable basis for the interpretation and comparison of more complex muscle modelling approaches. For more complex second order nonlinear muscle model see [19]

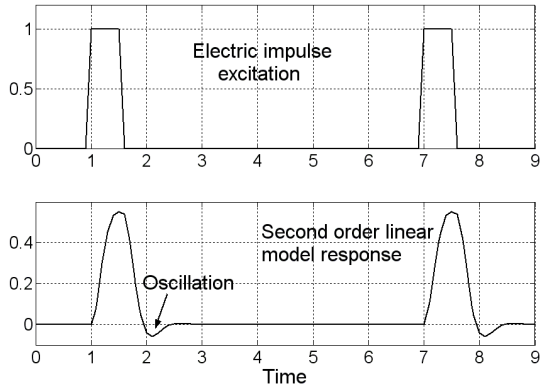

Figure 8.8: Muscle response to force excitation for second order linear model simulated example.

8.5 Proposed hybrid fuzzy modelling algorithm

Methods for data-driven fuzzy modelling and identification are of great efficiency. Many publications focus on hybrid neuro-fuzzy models without taking in consideration the notion of interpretability of the builded model. Such models are black-box models like in neural networks. In this proposed hybrid modelling algorithm two important criteria have been considered. The first one is the problem of model initialisation, which can help us, in case of good initialisation, to start with a model that is close to the optimal needed model. The second one concerns the learning methods, with optimal identification model initialisation we don't need a complex learning methods and the number of learning epochs (time learning) will be hugely reduced. These two criteria has been introduced in this proposed hybrid modelling algorithm. The results using four different testing data, see following setion 8.10, show that this proposed identifier-model is able to identify the real system using only its measured input-output data.

The optimisation of this proposed hybrid algorithm considers the membership functions and the rule consequence coefficients that minimise certain quadratic objective function. Thus the main goal is to minimise the following sum of squared errors E in equation 8.5.1, see for more details section 6.4.3:

$$E = \sum_{n=1,N} (d_n - x_n)^2 \tag{8.5.1}$$

Where d_n: desired output (Knee-joint angle), x_n: the proposed hybrid algorithm output, and N: number of measured training data samples available from the real system.

The Hybrid Algorithm Proposed here has the following functional structure:

1) zero order *Takagi-Sugeno* (*T.S.*) model [67].

2) two inputs as mentioned in section 8.5.

3) each input is partitioned with three gaussian membership functions (*gbellmf*). Three membership functions: A_1, A_2, and A_3 for the first input x, and three membership functions: B_1, B_2, and B_3 for the second input y. This partition lead to a fuzzy model with 9 rules. These rules are described in equation 8.5.2.

$$if\ x\ is\ A_i\ and\ y\ is\ B_j\ then\ z\ is\ r_k. \tag{8.5.2}$$

Where $i,\ j = 1, 2, 3$, and $r = 1, \ldots, 9$ (number of rules).

The degree of membership for each input x and y is defined in equation 8.5.3:

$$\mu_{A,\ B}(x,\ y) = \frac{1}{1 + (\frac{(x,\ y) - b}{a})^{2c}} \tag{8.5.3}$$

Where a, b and c: are the premise parameters that will be updated using the Gradient Descent (*GD*), see section 6.2.3. The firing strength of each rule is given as a product of the membership degrees in equation 8.5.4:

$$w_m = \mu_{A_i}(x) \cdot \mu_{B_j}(y),\ i, j\ =\ 1,\ 2,\ 3,\ and\ m\ =\ 1\ , \ldots,\ 9. \tag{8.5.4}$$

The overall output can be expressed as linear combinations of the consequent parameters. More precisely, the output z can be written in equation 8.5.5:

$$Z = \frac{\sum_{m=1,\ldots,9} w_m * z_m}{\sum_{m=1,\ldots,9} w_i} \tag{8.5.5}$$

8.6 Rules consequent parameters initialisation using RPA

The Rapid Prototyping Algorithm *RPA* [24] reposes on the fact that more a measured point of training data is close to the core of the membership function, more the conclusion of the corresponding rule is close to the desired output associated to this measured point. The core of a fuzzy membership function A is defined by equation 8.6.1:

$$N_A = \{x \in R / \mu_A(x) = 1\} \tag{8.6.1}$$

Example:

The input-output couples $\{(x_i,\ y_i);\ z_i\}$ are considered,

where $i\ = \{\ 1,\ .\ .,\ N\}$, and N: number of samples.

Each rule has only one consequent parameter z_p that represents the image of the input couple $(x_p,\ y_p)$ with maximum inference, where $1\ \leq\ p\ \leq\ N$. With this method it's possible to find 9 prototypes from measured data set for rules consequent parameters. This method gives rapidly the initial solution close to the desired output and prevent the derivative-based optimisation algorithms getting stuck to the local minimum.

8.7 Hybrid algorithm steps

In short word, this approach (hybrid algorithm) optimizes:

a) premise parameters with Gradient Descent (GD), for more details see section 6.4.3.b.

b) consequent parameters with Least Squares Estimators (LSE), for more details see section 6.4.3.a.

Following steps give global description of this algorithm:

1) initialise the 9 rule conclusions using Rapid Prototyping Algorithm (RPA).

2) create the initial model from measured input-output data set and initialised rules.

3) calculate the response Z_{model} of this initial-model.

4) for each measured data pairs $\{(x_i,\ y_i),\ z_i\},\ i = 1,\ \ldots,\ N$, measure the error e_p given as:

$e_i = z_{i_{desired}} - z_{i_{model}}$.

5) epochs = 1 to epoch-max

6) Test if this error E_{model} is acceptable or not:

7) if yes, then END.

8) apply nonlinear parameters optimisation for all membership functions in the following way:

$$\frac{\partial A(x)}{\partial a} = \frac{2c}{a} \cdot (1 - A(x)) \cdot A_1(x) \tag{8.7.1}$$

$$\frac{\partial A(x)}{\partial b} = \frac{2c}{a} \cdot (\frac{x-b}{a})^{2c-1} \cdot (A_1(x))^2 \tag{8.7.2}$$

$$\frac{\partial A(x)}{\partial c} = -2c \cdot (\frac{x-b}{a})^{2c-1} \cdot (A(x))^2 \tag{8.7.3}$$

where A: represents *gbellmf* function.

9) calculate the new membership function parameters:

$$(a,b,c)_{next} = (a,b,c)_{now} - \rho \frac{\partial e_i}{\partial (a,b,c)} \tag{8.7.4}$$

10) apply linear parameters optimisation for all rule consequents (see section 6.4.3.a

11) calculate the new rule consequents

12) go to step 3.

8.8 Methodology

A wide class of nonlinear dynamic processes are *SISO*. In this case, the knee-joint dynamic process has one input u (pulse width) and one output y (angle); the dynamic model can be described by the following formula, using 6 historical (past) inputs: $u(k-1),\ \ldots,\ u(k-6)$, and 4 historical outputs: $y(k-1),\ \ldots,\ y(k-4)$, equation 8.8.1:

$$y(k) = f[u(k-1), u(k-2), ..., u(k-6), y(k-1), y(k-2), ..., y(k-4)]. \tag{8.8.1}$$

Where f is the function to be identified based on measurement input-output data of real system. This function has ten (10) inputs. The number of inputs is big to use fuzzy modelling procedure. For fuzzy model, the number of rules will be hugely big. If all combinations are considered for a fuzzy inference system with 10 inputs, each one with three membership functions, the grid partitioning leads to 3^{10} rules, which is very large for any practical learning methods. Therefor the number of inputs must be reduced, we consider therefor only two inputs. The question now is to define or find these two best inputs between these ten inputs, which can give the best description

of our real system dynamics. The solution is to build models of two inputs through the combination between historical inputs and historical outputs. The number of combinations is equal to $6 \times 4 = 24$. The optimisation of these 24 models will be done during only one epoch, and then the best one undergo further refine optimisation epochs.

All these 24 pair-inputs will be tested in one built model. This method, section 8.5, uses constant consequent parameters with zero-order T.S. model and is based on generating only one fuzzy model (Hybrid Algorithm) that employs all 24 possible data pair inputs. Note that the fuzzy model has nine rules, three generalized bell-shaped membership functions for each input. The training procedure for each pair input is done during only one epoch.

The pair inputs, *y(k-1)* and *u(k-6)*, has been found as the best model, hence we take this pair as input variables for the further study in following section 8.9. Note that the computation time for this identification of the best pair input is done during 2.2660 sec [*AMD Athlon (tm) Processor*]. Thus the best found model of knee-joints dynamics is described by the following formula, equation 8.8.2:

$$y_i(k) \;=\; f[u_i(k-6),\; y_i(k-1)]. \tag{8.8.2}$$

where i is the rank of the set input, $i \;=\; \{1,\; .,\; .,\; .,\; N\}$, and N: number of training data samples.

8.9 Optimisation of selected model

The training data set (*data-sc2*), see figure 8.5 contains 895 points (samples). Each input is partitioned into three fuzzy sets using *gbellmf* membership functions. Following figure 8.9 shows us the resulting initial membership functions. The initial hybrid

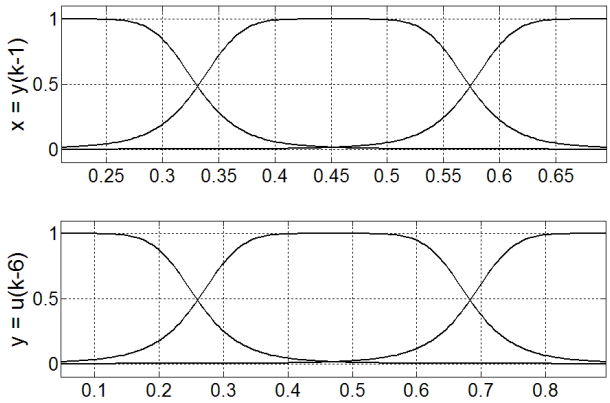

Figure 8.9: Initial membership functions with overlap equal to 0.5, for input1 and input2.

algorithm is built using described rapid prototyping method, see section 8.6, with selected pair inputs *y(k-1)* and *u(k-6)*. Without any optimisation, this initial model is tested using the same input signal for training data (*data-sc2*). The output response of this initial model, is shown in figure 8.10. This output signal of our initial model, using only rapid prototyping algorithm, is close enough to the desired output of real system. Now this initial model is trained during only 10 epochs, with learning rate fixed to 0.003 for all epochs. The final fuzzy sets generated by this optimised hybrid model is illustrated in the following figure 8.11(b). The time needed for the optimisation of this hybrid model during 10 epochs is equal to 1.7660 sec [*AMD Athlon (tm) Processor*]. In this application the consequent parameters are remained in the

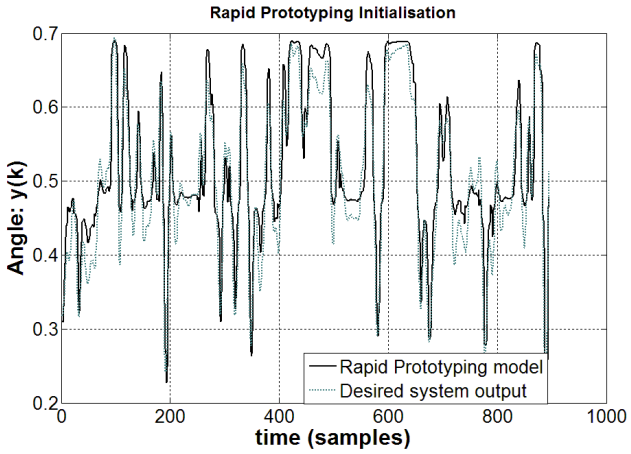

Figure 8.10: Response of initial proposed hybrid model (solid line), and desired system output (dashed line).

envelope of possible values of real system output. The initial values of consequent parameters found with *PRA* algorithm are: $z1,\ z2,\ \ldots\ z9 =$

[0.2692 0.2159 0.3779 0.4466 0.4769 0.4686 0.6567 0.6925 0.6932].

The resulting values after 10 epochs optimisation are:

[0.2238 0.2134 0.1459 0.4327 0.4787 0.5144 0.7142 0.6920 0.7139].

From the viewpoint of interpretability, the consequent parameters have physical interpretation that represent the normalised angle between thigh and shank, figure 8.11(c). The membership functions that partition the two input space have not undergos great changes, 8.11(a).

8.10 Hybrid model validation

Model validation is the heart of the identification problem. The following signals *Test-sc1*, *Test-sc3*, *Test-a* and *Test-prbs* described in section 8.3 will be tested with this optimised hybrid model to evaluate the qualities of the obtained knee-joint dynamics

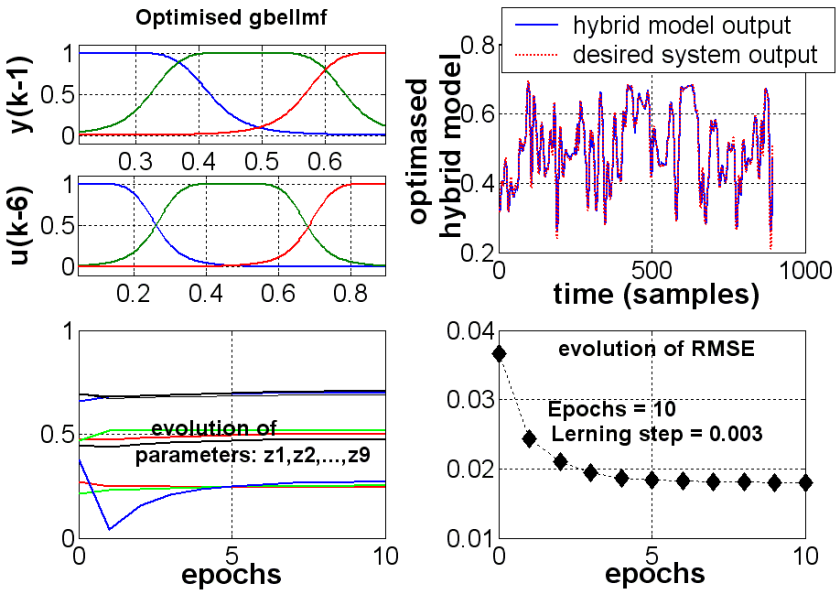

Figure 8.11: a. Optimised *gbellmf*; b. Prediction performance of hybrid Algorithm (solid line), and system output (dashed line); c. Evolution of rules consequents; d. Error between desired system output and hybrid model output after 10 epochs (using train-data set (*test-sc2*).

identifier-model.

The efficiency of the this hybrid fuzzy model, representing non-linear input-output dynamics, depends on the initial fuzzy partition of the input space. The tuning of the premise fuzzy sets and consequent parameters is achieved through three techniques: Rapid Prototyping Algorithm, Gradient Descent and Least squares Estimator.

8.10.1 Signal *Test-SC1*

The identification efficiency of the this hybrid fuzzy model representing non-linear input-output dynamics with data $test - sc1$ is shown in figure 8.12.

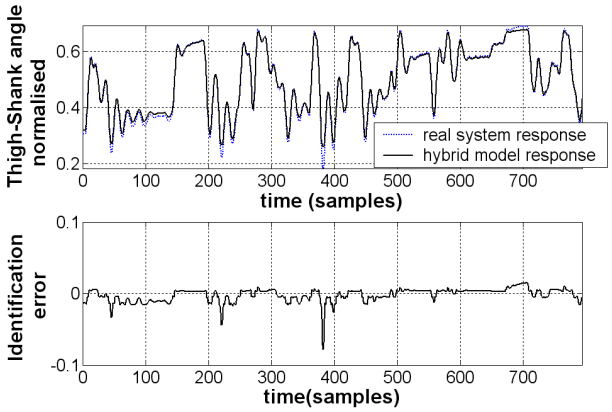

Figure 8.12: Above: dynamic system response with hybrid algorithm (solid line) and system output (dashed line). Below: the identification error of their difference. (for testing data: *test-sc1*).

8.10.2 Signal *Test-A*

The identification efficiency of the this hybrid fuzzy model representing non-linear input-output dynamics with data *test-A* is shown in figure 8.13.

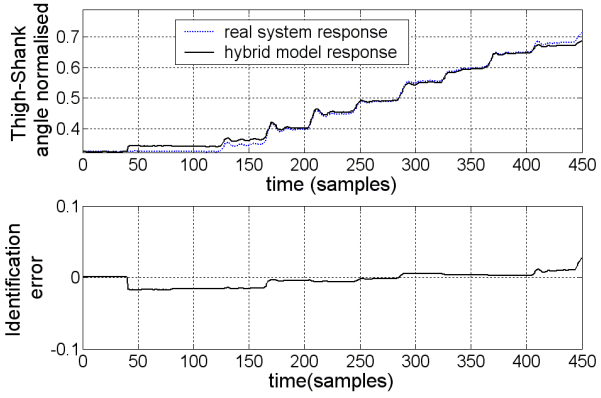

Figure 8.13: Above: dynamic system response with hybrid algorithm (solid line) and system output (dashed line). Below: the error of their difference. (for testing data: *test-A*).

8.10.3 Signal *Test-PRBS*

The identification efficiency of the this hybrid fuzzy model representing non-linear input-output dynamics with data *test-prbs* is shown in figure 8.14.

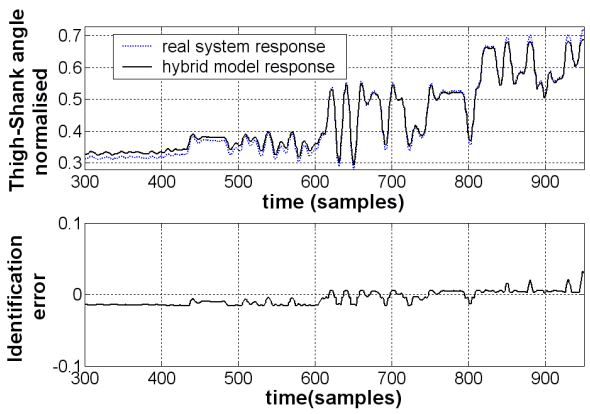

Figure 8.14: Above: dynamic system response with hybrid algorithm (solid line) and system output (dashed line). Below: the error of their difference. (for testing data: *test-prbs*).

8.11 Conclusion

In this Chapter, an effective fuzzy identifier-model based on Hybrid Algorithm is proposed. In this proposed Algorithm the fuzzy rule-conclusion and membership-function parameters can be generated and optimized automatically from the training data.

The proposed fuzzy hybrid algorithm provides a fast and effective method for identification of knee-joint system dynamics (thigh-shank). This method focuses on model simplicity and time-consuming, under a satisfactory modeling accuracy with transparent fuzzy sets. The results obtained demonstrate that this hybrid algorithm approach is effective for identification of nonlinear dynamic process and can be a good

alternative when a mathematical model of the plant is not available. The further improvement in the model structure, (for example: three inputs and adapting learning rate), for this method would be beneficial.

Chapter 9

Conclusions and future works

9.1 Recapitulation

The human neuromusculoskeletal system is considered in this thesis in two parts: 1) Surface electromyogram (EMG) neuromuscular signals recognition for hand movements classification. 2) Musculoskeletal system dynamics identification.

The first goal of this thesis, concerned the classification of surface EMG neuromuscular signals, which can be designed for the control of myo-prostheses and also exoskeleton devices. The surface electrodes transduce the motor unit action potentials MUAPs into resultant electrical EMG signal, which can be recorded following appropriate processing, amplification and filtering procedures.

EMG signals are electrical activities originating in the brain and that are transported via nerve cells to the muscles. These signals cause the contraction of muscles. These measurements of EMG signals issued from muscle contractions are realisations of a complex time-variant process that control electrical activation of muscles. They provide an access to physiological processes, which cause muscles to generate forces, produce movements, and accomplish functions that allow the human to interact with

the world around. In the amplification process of these small bioelectric signals, which are typically in uV, it's necessary to reduce as much as possible the effect of noisy electrical signals. This signal (EMG) should be recorded with a certain fidelity to assure the transmission of its inside information (without loss of information), which should be recognised and classified after that.

In many cases, the sequentially localisations of the information carried by the observed signal and the disturbances are known as *a priori*. The objective is then to build a new signal starting from the raw signal, which preserves important information carried by the signal, and excludes the disturbances. The principle function of a filter then, is to filter out the unwanted parts of an input signal. These unwanted frequency parts cannot be all eliminated, but only reduced. The *MATLAB* signal processing toolbox contains a number of different functions and commands for designing low-pass, high-pass and band-pass filters. Digital filters are designed through various prototypes: *Chebyshev*, *Butterworth*, and *Elliptic* for *IIR* type. *Equiripple*, *least squares*, and *Kaiser window* are designed for *FIR* type. The optimal type filters are chosen on the basis of implementation complexity, magnitude response and efficiency. The design specifications of the band-pass filter and its order are given for many examples. A comparison of these filters was attempted in order to evaluate the advantages and drawbacks of each filter. It was concluded that *IIR* filters are less complex and lead to the same accuracy classification results than *FIR* filters. Filter pass-bands have a great importance to transmit well-defined information, and to reject other disturbances. About the choice of this filter pass-band, it was proved that one fixed pass-band filter can not be generalised, as an optimal frequency band, for all other features. In this thesis three different analysis methods are used:

1) time-domain, 2) frequency-domain and 3) time-frequency-domain. The goal was to extract features corresponding to each analysis method and compare between them. After EMG signal pre-processing operation, using spectrum analysis based on short-time Fourier transform ($STFT$), which is a form of local Fourier analysis that treats time and frequency simultaneously. It was possible to exploit and to quantify the behaviour of dynamic information present in these EMG signals and design characteristic vectors (features). Some of these characteristic vectors could perform a good discrimination of different hand movement classes.

The following task was to reduce the space of the number of extracted feature vectors. PCA is a way of expressing a high dimensional data set in an alternative set of a low dimensional data set with high variability. Such method is also used for data visualisation and clustering. In this study a linear dimension reduction technique (PCA) was investigated. In this thesis it was shown that the feature *Fstd* alone, in 2D feature space, gives by it self an average number of misclassified instances less than in the case of 2D reduced feature space using PCA. Hence we could conclude that the spatial reduction, using PCA, of many features for classification accuracy doesn't lead necessarily to better results.

The classification procedure is performed on the basis of classes' feature distribution. These classification models belong to two categories. First are supervised models, like Multi-Layer Perceptron, Radial Basis Networks, and Learning Vector Quantization network. Second are unsupervised models, like Self Organizing Map, Fuzzy Subtractive Clustering and Competitive Layer. Four intelligent computational algorithms were be used to perform the classification of four different hand movements according to their corresponding EMG signals. Intelligent computational algorithms used in

this thesis are Multi-Layer Perceptron (MLP), Radial Basis Networks (RBF), Learning Vector Quantization network (LVQ) and fuzzy subtractive clustering (FSC). The purpose of this study was to illustrate these various intelligent computational algorithms and to compare them with the performance of our proposed $FTMC$ fuzzy classifier-algorithm. The strategy of choosing between different classification methods is of great importance. In case of on-line prosthesis control, or exoskeleton device control, the needed time for signal acquisition, then for processing and finally decision control, should be short. Hence the choice of classification method should considers both time consuming and performance. For these considerations proposed fuzzy $FTMC$ classifier-algorithm presents acceptable results. Its advantage can be seen in optimisation methods, which are simple and not time consuming, like Gradient Descent (GD) and Least Squared Error (LSE), and also utilisation of simple Trimmed Mean method for determination of initial input fuzzy sets (structural optimisation). In case of dynamical complex systems, like forearm EMG signals analysis, intelligent computational methods are shown, through the comparison between them, their efficiency and ability to deal with these systems and to give good classification-models.

After EMG neuromuscular signals classification, the second part of this thesis, considered the study of musculoskeletal system dynamics identification. Such identification operation should be able to capture the highly non-linear dynamics of musculoskeletal systems (case of knee-joint dynamics) as a result of FES stimulations. Our developed fuzzy-identifier model, from experimental data, demonstrated the model's ability to capture the nonlinear time-varying dynamics observed experimentally from data-measure. The examined real system is consisted of the quadriceps, Surface

Electrical Stimulation signal and a freely swinging shank. The output of this system is the angular position, measured by externally mounted sensors. The gaol was the identification of the dynamic relationship between quadriceps dynamics (total quadriceps-shank system) and the *FES*-induced movement (angle). Surface stimulation electrodes are placed on the quadriceps group of muscles form the major part of the muscles on the front of the thigh. The knee angle θ (system output) between thigh and shank is sampled at *20Hz*, and measured using an electrogoniometer. Input signal is the variable pulse width of the mono-phasic current pulses with fixed frequency, *20Hz*, and fixed Amplitude. The range of θ angle variation between 90 (rest position) and 180 (full extension), is normalised to the range $z = [0, 1]$. Many tests are carried out, and classified by the kind of experiments into four experimental sessions. One measured data is used for this proposed model as training (learning)data, and four other measured data are served for this proposed model as evaluating (testing) data. The proposed hybrid fuzzy model provided a fast and effective algorithm for modeling the knee-joint dynamics system. This algorithm is based on model simplicity and time-computing efficiency, under a satisfactory identification accuracy with transparent fuzzy sets.

9.2 What are the applications of this Thesis

An understanding of EMG neuromuscular signals and musculoskeletal system dynamics can help disabled persons in regaining lost function and improving their activity of daily living life and also for assessing rehabilitation progress. Moreover it enables better assessment and therapeutic operations for them.

The results obtained were very well. With the known Short Time Fourier Transform

($STFT$) and our proposed interpretable fuzzy classifier-model ($FTMC$) and also other known methods, we could reach good accuracy classifications. In case of optimised values of these two parameters, which are: 256 ms for signal length and 2kHz for sampling frequency, the classification accuracy, using our proposed algorithm, could raech 95% see figure 7.13. Moreover, in this thesis, it is shown that with only two channels (one feature vector) it was possible to recognise and classify hand and also finger flexion movements, which are thumb-, pointer- and middle-finger. This discrimination can be increased if the number of channels and feature vectors are increased. This is an important result that will determine the future implementations of hand-prostheses control.

The results of first part of this thesis can be applied to help the patient, with amputated hand, to keep the neuromuscular activity of his forearm muscles for the manipulation of a myo-prosthesis or hand-exoskeleton device. Moreover this recognition of EMG signals help to keep also the virtual neural activity of the brain related to forearm-neuromuscular activity. The devices, myo-prosthesis and hand-exoskeleton, can be considered as human-machine interfaces that can be able to recognise the desired hand and finger movements of the operator and reproduce the same movement as intelligent and real-time human-machine interface.

Functional electrical stimulation (FES) impulses can be used to activate muscles disabled by spinal cord injuries, after stroke, and support the weak voluntary muscle activities. The efficiency of muscle-stimulator worn on the leg or arm of patient and placed over affected muscles need a musculoskeletal system models, which are able to reproduce the knee-joint dynamics or elbow-joint dynamics. Such systems can control

Functional Electrical Stimulation (*FES*) that will support the patients to accomplish a right legs or arms movements. The proposed interpretable fuzzy hybrid identifier-model had a well results, it could reproduce the knee-joint dynamics in estimating of the angle output of four different stochastic *FES* signal inputs: figures 8.12, 8.13, and 8.14

9.3 The goal of this research and future works

The goal of this thesis was to classify the EMG neuromuscular control signals for hand and finger movements and also to identify the musculoskeletal system of knee-joint dynamics using electro stimulation (*FES*), which is done with success.

The future work is attempted to combine the recognition of EMG signals and the models of musculoskeletal system dynamics of elbow-joint and knee-joint movements [62]. Such models allow determination of the set of electrical stimulation that produce the desired movement through their inverse dynamic models. It will be necessary to develop the causal relationship between neuromuscular EMG patterns and musculoskeletal system dynamics.

By post-stroke subjects, there are weak EMG signals, which are extremely small and unable to control the muscles. The correction control of muscle activations with consideration of their musculoskeletal system dynamics would correct their gait pattern so that they match the healthy gait patterns. This is a method of cognitive re-learning to rehabilitate the muscles of paralyzed legs or arms, which cannot be done with electrical stimulation alone.

Finally we can conclude that this thesis covered the study of many intelligent computational tools that are used to analyze and process bio-experimental data. The techniques presented are those that have been most widely and successfully applied to the analysis of physiological systems. Moreover they address issues such as randomness, complexity, dynamic and nonlinearity. In addition, in this thesis it is brought together the most useful methods, and sufficient mathematical details are provided to enable the understanding of these intelligent computational techniques. Thus, this complete and detailed research, in this thesis, will be useful to life-science investigators on several levels to realise several projects, which deal with EMG control signals and musculoskeletal system dynamics, and do further investigations in this area.

Bibliography

[1] http://www.e-learning.uk.com/beauty/Samples/Beauty/lower-arm-bones.htm.

[2] netMBA: http://www.netmba.com/statistics/covariance/.

[3] *(site of otto bock). http://www.fr.ottobock.com/healthcare.*

[4] *(site of rsl steeper), http://news.bbc.co.uk/2/hi/uk_ news/magazine/3146771.stm#graphic*

[5] *(site of southampton hand). http://research.ecs.soton.ac.uk/ projects/sh.html.*

[6] *(site of the university waseda). http://www.humanoid.waseda.ac.jp.*

[7] D. S. Andreasen, S. K. Allen, and D. A. Backus, *Exoskeleton with emg based active assistance for rehabilitation*, proceedings 9th Int. Conf. on Rehabilitation and Robotics, pp:333-336, 28 June-1 July (2005).

[8] Hill AV., *The heat of shortening and the dynamic constants of muscle*, Proc R Soc London B Biol Sci 126, pp:136-195 (1938).

[9] Christopher M. Bishop, *Neural networks for pattern recognition*, Oxford University Press Inc., New York, 1996.

[10] J. Bobet, RB. Stein, and MN. Oguztoreli, *A linear time-varying model of force generation in skeletal muscle*, IEEE Transactions on Biomedical Engineering. Vol:40, p:1000-1006 (1993).

[11] A.G. Bors, *Robust rbf networks, design and applications*, Book Chapter, Eds. R.J. Howlett, L. C. Jain, Physica-Verlag: Heidelberg, pp:125-153 (2001).

[12] M. Carrozza, C. Suppo, F. Sebastiani, B. Massa, F. Vecchi, R. Lazzarini, M. Cutkosky, and P. Dario., *The spring hand: Development of a self adaptive prothesis for restoring natural grasping.*, Autonomous Robots (2003).

[13] S. Chen, C.F.N. Cowan, and P.M. Grant, *Orthogonal least squares learning algorithm for radial basis function networks*, IEEE Trans. on Neural Networks, vol:2, pp: 302-309 (1991).

[14] P. Dario, C. Laschi, M. Carrozza, E. Guglielmelli, G. Teti, B. Massa, M. Zecca, D. Taddeucci, and F. Leoni., *An integrated approach for the design and development of a grasping and manipulation system in humanoid rebotics.*, IROS 2000, IEEE/RSJ Int. Conf. on Int. Rob. and Sys. Japan (2000).

[15] J. Valente de Olevirea, *Semantic constraints for membership function optimisation*, IEEE-Trans. Systems, Man. Cybern., pt. A, vol: 29, pp: 128-138 (Jan, 1999).

[16] M. A. Denai, F. Palis, and A. Zeghbib, *Anfis based modelling and control of nonlinear systems.*, Systems, Man and Cybernetics, 2004 IEEE International Conference, vol:4, pp: 3433-3438 (October 10-13 2004).

[17] J. Devor and R. Peck, *Statistics: the exploration and analysis of data*, 3rd ed New York USA.

[18] M. DiCicco, L. Lucas, and Y. Matsuoka, *Comparison of control strategies for an emg controlled orthotic exoskeleton for the hand*, proceedings, Int. Conf. on Robotics and Automation, May-July (2004).

[19] J. Ding, SA. Binder-Macleod, and AS. Wexler, *Two-step, predictive, isometric force model tested on data from human and rat muscles*, J Appl Physiol. Vol:85, p:2176-2189 (1998).

[20] R. Duff, P. Nolan, M. Rybansky, and M. Malley, *Evolution in impedance at the electrode-skin interface of two types of surface emg electrodes during long-term recordings*, Proceedings, Electrophysiology and Kinesiology. University of Vienna, Austria. (2002).

[21] G. M. Foody, *Supervised image classification by mlp and rbf neural networks with and without an exhaustively defined set of classes*, Taylor & Francis, International Journal of Remote Sensing, Volume 25(15), pp: 3091-3104 (August 10, 2004).

[22] S. Fujii, D. Nishikawa, and H. Yoko., *Development of a prosthetic hand using adaptable control method for human characteristics.*, IOS Press, Amsterdam, vol: 5, pp:360-376 (1998).

[23] Proakis John G. and Manolakis Dimitris G., *Digital signal processing Ű principles, algorithms, and applications*, Prentice Hall, 1996.

[24] P. Y. Glorennec, *Algorithmes d'apprentissage pour systèmes d'inférence floue*, Edition Harmès, France, 1999.

[25] Marian B. Gorzalczany and Adam Gluszek., *Neuro-fuzzy systems for rule-based modelling of dynamic processes*, ESIT 2000, 14-15 September 2000, Aachen, Germany. Vol:18, p:55-87 (2000).

[26] B. Hannaford and S. Lehman, *Shorttime fourier analysis of the electromyogram: Fast movements and constant contraction*, IEEE Transactions on Biomedical Engineering, vol. BME-33, pp: 1173-1181 (Dec. 1986).

[27] S. Haykin, *Neural networks: A Comprehension Fundation*, Prentice Hall, 1998.

[28] O. E. Holland, P. J. Kyberd, R. Tregidgo, P. Bagwell, and P. H. Chappell, *Experiences with a hierarchically controlled myoelectric hand*, Orthopädie Technik, pp: 566-569 (1996).

[29] Park J. and Sandberg J.W., *Universal approximation using radial basis functions network*, Neural Computation, vol: 3, pp: 246-257 (1991).

[30] J. E. Jackson, *A Users' Guide to Principal Components*, Wiley-IEEE, New York, 2003.

[31] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing*, Prentice Hall, 2002.

[32] J.-S.R. Jang, *Anfis: Adaptive-network-based fuzzy inference systems*, IEEE Trans. Systems, Man & Cybernetics 23, pp: 665-685 (1993).

[33] Ian T. Jolliffe, *Principal component analysis*, Springer, New York, 2002.

[34] R. E. Kearney and I. W. Hunter, *System identification of human joint dynamics*, Biomedical Engineering. Vol:18, p:55-87 (1990).

[35] F. Peterson Kendall, E. Kendall McCreary, and P. Geise Provance, *Muskeln: Funktionen und tests*, URBAN and FISCHER, 2001.

[36] T. Kohonen, *Learning vector quantization: Technical report*, Helsinki Univ. of Tech., Otaniemi. (1986).

[37] ———, *Self organization and associate memory*, Springer-Verlag, London, 3rd edition, 1989.

[38] ———, *Improved versions of learning vector quantization*, International Joint Conference on Neural Networks, volume 1, pages 545-550 (1990).

[39] ———, *Self-organizing maps*, Springer Series in Information Sciences, 30, pp. 59Ű144. 2nd ed. New York, 1997.

[40] B. Kosko, *Neural networks and fuzzy systems. a dynamical systems approach to machine intelligence*, Prentice Hall, Englewood Cliffs, NJ, 1992.

[41] P. J. Kyberd, O. E. Holland, P. H. Chappel, S. Smith, R. Tregidgoi, P. J. Bagwell, and M. Snaith., *Marcus: A two degree of freedom hand prosthesis with hierarchical grip control.*, IEEE Trans Rehab Eng, vol: 3(1), pp: 70-76 (1995).

[42] Kaufman Leonard and Rousseeuw Peter J., *Finding groups in data. an introduction to cluster analysis*, Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990.

[43] L. H. Lindstrom and R. I. Magnusson, *Interpretation of Myoelectric Power Spectra: A Model and its Applications*, Proceedings of the IEEE: The Physiome and Byond, vol:65/5, pp:653-662. May (1997).

[44] Lennart Ljung and Torkel Glad, *Modeling of dynamic systems*, Prentice Hall, Series In Information And System Sciences, 1994.

[45] Carlo J. De Luca, *Surface Electromyography: Detection and Recording*, by DelSys Incorporated. (2002).

[46] Richard G. Lyons, *Understanding digital signal processing*, Prentice Hall PTR, second edition March 15, 2004.

[47] S. Micera, G. Vannozzi, A. M. Sabatini, and P. Dario, *Improving detection of muscle activation intervals*, IEEE Engineering in Medicine and Biology Magazine, vol: 20(6), pp:38-46. (2001).

[48] M. Mulas, M. Folgheraiter, and G. Gini, *An emg-controlled exoskeleton for hand rehabilitation*, proceedings 9th Int. Conf. on Rehabilitation and Robotics, pp:371-374, 28 June, 1 July (2005).

[49] D. Nauck and R. Kruse, *Designing neuro–fuzzy systems through backpropagation*, W. Pedrycz, Ed., Fuzzy Modelling: Paradigms and Practice, pp: 203-228, Kluwer Boston (1996).

[50] O. Nerrand, P. Roussel-Ragot, D. Urbani, L. Personnaz, and G. Dreyfus, *Training recurrent neural networks: why and how? an illustration in process modeling*, IEEE Trans. on Neural Networks vol: 5, pp: 178-184 (1994).

[51] B. Nisim and S. Kenny, *Myoelectric Hand Orthosis*, Journal of Prosthetics and Orthotics, vol:2(2), pp:149-154. (1990).

[52] Richard o. Duda and Peter E. Hart, *Pattern classification and scene analysis*, (1995).

[53] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck, *Discrete-time signal processing*, Prentice Hall, USA, 1998.

[54] Sophocles J. Orfanidis, *Introduction to Signal Processing*, 1995.

[55] Mark J. L. Orr, *Introduction to radial basis function networks*, Centre for cognitive Science, University of Edinburgh, Scotland (1996).

[56] G. Puchhammer, *The tactile slip sensor: Integration of a miniaturized sensory device on an myoelectric hand*, Rehabilitation and Home Health Care, pp:07-11, in Orthopädie-Technik Quarterly, English, edition I. (2000).

[57] M. Reischl, R. Mikut, C. Pylatiuk, and S. Schulz, *Control strategies for hand prostheses using myoelectric patterns*, 9th Zittau Fuzzy Colloquium, Zittau, Germany. (2001).

[58] M. P. Robinson, D. Bozec, and C. A. Marshman, *Healthcare engineering and electromagnetic compatibility*, Proc. I Mech E Conf. on Healthcare Engineering: Latest Developments and Applications, London. pp 53-61, 25-26 November. (2003).

[59] T. Schauer and K. J. Hunt, *Nonlinear predictive control of knee-joint angle using fes*, Proceedings of the 5th Annual Conference of the International Functional Electrical Stimulation Society (IFESS 2000), pp. 425-428, Aalborg, Denmark (2000-06-18).

[60] T. Schauer, K. J. Hunt, M. H. Fraser, W. Stewart, and F. Previdi, *Identification of a biomechanical system using neural networks*, IFAC Workshop on Adaptation and Learning in Control and Signal Processing 2001,S. Bittanti,pp. Cernobbio-Como, Italy (2001-08-29).

[61] M. Setnes, R. Babuska, U.Kaymak, and H. R. van Nauta Lemke, *Similarity measures in fuzzy rule base simplification*, IEEE Trans. Syst., Man, Cybern. B, vol: 28(3) pp: 376-386 (1998).

[62] Qi Shao and Thomas S. Buchanan, *Estimation of corrective changes in muscle activation patterns for post-stroke patients*, CBER Biomechanics Research Symposium. (2005).

[63] Von Jun Shao, *Mathematical statistics*, Springer, 2003.

[64] B. A. Shenoi, *Introduction to digital signal processing and filter design*, Wiley and sons, 2005.

[65] M. Slack and D. Berbrayer, *A myoelectrically controlled wrist-hand orthosis for brachial plexus injury: A case study*, Journal of Prosthetics and Orthotics, vol.4(3), pp.171-174 (1992).

[66] Y. Song and P. Wang, *A predictive model based on rbf neural network*, Intelligent Systems and Control, 23-25/08, 446-098, Honolulu, Hawaii, USA (2004).

[67] T. Takagi and M. Sugeno, *Fuzzy identification of systems and its applications to modeling and control*, IEEE Trans. Systems, Man, and Cybernetics, vol: 15(1), pp: 116-132 (1985).

[68] N. Tschichold-Gürman, *RuleNet a New Knowledge-Based Artificial Neural Network Model with Application Examples in Robotics*, Ph.D. thesis, ETH Zürich, 1996.

[69] Lotfi A. Zadeh, *Outline of a new approach to the analysis of complex systems and decision processes.*, IEEE Transactions on Systems, Man, and Cybernetics, vol:3(1), pp: 28-44 (January 1973).

[70] G.I. Zahalak and S.P. Ma, *Muscle activation and contraction: constitutive relations based directly on cross-bridge kinetics*, Biomechanical Engineering, PubMed Abstract. Vol:112, pp: 52-62 (1990).

[71] M. Zecca., *On the development of a cybernetic prosthetic hand*, Ph.D. thesis, Scuola Superiore Sant'Anna, mars 2003.

[72] A. Zeghbib and F. Palis, *Effective fuzzy identification method to modeling the input-output behavior of the knee-joint dynamics*, Rencontres Francophones sur la Logique Floue et ses Applications. ISBN:2-85428-624-3, Réf.:624, pp: 191-198 (2003).

[73] A. Zeghbib, F. Palis, and F. B. Ouezdou, *Emg-based finger movement classification using transparent fuzzy system.*, proceedings of the 4th conference of the European Society for fuzzy logic and technology, EUSFLAT, Technical Univ. of Catalonia, pp: 816-821, 7-9 September (2005).

[74] A. Zeghbib, F. Palis, R. C. Salbert, and T. Schauer., *Dimension reduction effect on emg signal identification using mlp, rbf and lvq methods in case of relevant features.*, Automed 6 Workshop. Rostock-Warnemunde, Germany (24-25 March 2006).

[75] A. Zeghbib, Frank Palis, and F. B. Ouezdou., *Emg feature evaluation using transparent fuzzy system for hand and finger movements identification.*, In International Federation for Medical and Biological Engineering (Hrsg.): 3rd European medical and biological Engineering conference, EMBEC-05. IFMBE European conference on biomedical engineering (Prague Czech Republic November/20-25). - proceedings. Zagreb : IFMBE, 2169, IFMBE Proceedings 11 (2005).