Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea

Subject-independent modeling of sEMG signals for the motion of a single robot joint through GMM modelization

Advisor: Prof. Enrico Pagello
Co-Advisor: Dr. Stefano Michieletto

Student: Francesca Stival

21 09 2015
Academic Year 2014/2015

Abstract

The interaction with robotic devices by means of physiological human signals has become of great interest in the last years because of its capability of catching human intention of movement and translating it into similar movement performed by robotic devices, i.e. humanoids. In this way, the robot can emulate or simulate living biological organisms and using patterns coming from human muscular-skeleton structure makes the behavior of humanoid autonomous robots more natural, robust and efficient.

This thesis evaluates the use of a probabilistic model, namely Gaussian Mixture Model (GMM), trained through Electromyography (EMG) signals to estimate the bending angle of a single human joint. Electromyography measures the electrical activity in muscles as a product of contraction. Usually, Surface Electromyography (sEMG) signals are preferred because of their capability of extracting information in a less invasive way. Our goal is to create a general model based on the data coming from different subjects. The aim is to enable every person using the model to control a robot without any additional train phase or in a very short one. Tests on the whole system are performed on new, unseen data collected from several people. The goodness of the estimated data is evaluated by means of Normalized Mean Square Error (NMSE) and correlation coefficient.

In this thesis, there are several parameters needed to be handle in order to create a very robust model, like the number of Gaussian components, the channels used for building the model and the size of the training set. As a first step, we performed an offline evaluation of the signals, so the time has been used as input to build a stationary model. The achieved results show that our framework can work well in both accuracy and computational time, even with data collected from different subjects.

More intensive tests has been performed by using also a noisier dataset. At the beginning, the results were very poor, so it has been used an algorithm called Dynamic Time Warping (DTW), this method can adapt the input signals to a reference one. The use of the DTW algorithm allows us to considerably improved the initial results and to correctly follow the desired trajectory.

After proving the goodness of the model offline, it has been considered the possibility of trying it online by using Daubechies wavelet function *db2* applied to EMG signals. This configuration, has been proved effectively on the single-subject case in a previous work. Even in this case, we improved the original framework by adding a phase of normalization and smoothing of the

signal after the construction of the wavelet. Our solution is able to generalize the approach to the subject-independent case by significantly improving the obtained results even in several real tests.

Finally, has been developed a C++ software capable of interfacing with a real robot by using Robot Operating System (ROS) modules. The whole procedure has been tested on a humanoid and a manipulator robot by remapping the human motion to the robotic platform in order to verify the proper execution of the original movement. An Aldebaran NAO robot and a Comau Smart5 SiX has been used for testing purposes.

Acronyms

AIC Akaike Information Criterion

BIC Bayesian Information Criterion

CWT Continuous Wavelet Transform

DoF Degree of Freedom

DoFs Degrees of Freedom

DWT Discrete Wavelet Transform

EEG Electroencephalogram

EM Expectation-Maximization

EMG Electromyography

sEMG Surface Electromyography

FFT Fast Fourier Transform

GMM Gaussian Mixture Model

IGMM Incremental Gaussian Mixture Model

GMR Gaussian Mixture Regression

HMM Hidden Markov Model

HMM Hill Muscle Model

HPR Hierarchical Projected Regression

IAV Integral Absolute Value

MAV Mean Average Value

MoG Mixture of Gaussians

LOO Leave One Out

MU Motor Unit

NMSE Normalized Mean Square Error

NN Neural Networks

RLfD Robot Learning from Demonstration

RMS Rooted Mean Square

ROS Robot Operating System

sEMG Surface Electromyography

YAML YAML Ain't Markup Language

DTW Dynamic Time Warping

GoF Goodness of Fit

COW Correlation Optimized Warping

Wavelet Transform Wavelet Transform

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Goal of the Thesis

The interaction with robotic devices by means of physiological human signals has become of great interest in the last years because of the capability of catching human intention of movement and translate it in a coherent movement performed by a robotic platform. If we think at the intention of movement as a cognitive process used by human to accomplish an actual motion, therefore this information is likely shared by many subjects. Several studies have been done about models built on a single subject [28] [45], for example estimating forces by tuning the model parameters to fit the motion of a particular person (subject-specific). On the other hand, executing a particular task intuitively leads to some constraints that could be extracted by looking to different interpretations of this task to obtain a subject-independent model. The few attempts in literature showed the possibility of creating a multiuser interface able to adapt to novel users (subject-independent) [47]. This thesis evaluates the use of a widely spread learning technique like Gaussian Mixture Model (GMM) [11] [48] trained through Surface Electromyography (sEMG) [5] signals coming from human subjects to actuate the knee and the wrist joint of different robot devices. Both stationary and non-stationary models have been built. The former explicitly used time as input data and processed information offline, while in the latter, we applied Wavelet Transform to use the model online. The goodness of the proposed framework has been tested by means of data collected from three different dataset involving various joints.

1.2 State of the Art

During the last years the interest in comprehending the physiological bases of human and animal movement has increased from a biomechanical and neurophysiological point of view. Up to now it is not possible to connect the mechanism which occurs at the neurophysiological level with the one that occurs at the musculoskeletal level in the human being. Some international projects have tried to study how intellective activity performs the motor functions in an adaptive way [1]. Despite the difficulties of the problem, the capability of catching human intention of movement is very relevant as well as its translation in an analogous movement performed by a robot, usually humanoids. In this way, the robot can emulate or simulate biological organisms. Using patterns coming directly from human musculoskeletal structure makes the humanoid behavior more natural, robust and efficient. These techniques can find several practical applications in order to achieve robust and intuitive human-machine interfaces [55], such as the creation of wearable devices like prosthesis or exoskeletons.

More than others, Electromyography-driven musculoskeletal modeling has been used to develop intuitive Human-Machine Interfaces (HMI) for the proportional and simultaneous control of multiple Degrees of Freedom (DOFs) in orthotic and prosthetic devices. Electromyography (EMG) measures electrical activity in response to a nerve stimulation of the muscle, this stimulation generates electrical activity in the muscle, which in turn causes contraction. Particularly relevant are the sEMG , they give less accurate results than the classical EMG, but have the advantage of being utilized in a simple and not invasive way. This is ideal in the prosthesis or exoskeleton control. In literature, several different ways to extract structural features from the information coming from EMG signals has been proposed by extracting structural characteristics from a single sEMG channel, but also simultaneously from different channels to build a cross-channel pattern. Different methods has been used for the extraction of the single feature, depending on the domain:

- Time domain (linear envelope [76], mean absolute value, root mean square [19], zero crossings, slope sign changes, waveform length [34], wave complexity [70], log-detector [32], histogram [77]);
- Frequency domain (power spectral moments [39], power spectral density [26], spectral magnitude averages [24], short time Fourier transform, median frequency [7], cepstrum [49], short time Thompson transform [24]);
- Time and frequency domain (wavelet packet transform [25], discrete

wavelet transform [61]).

For the simultaneous extraction of features from multiple channels for the determination of time-invariant features representing the principles of muscle coordination has been realized two techniques:

- Feature projection, transform a multi-channel input in a subspace of smaller dimension (NMF [56], PCA [6], ICA, fuzzy clustering [3], LDA, orthogonal fuzzy neighborhood discriminant analysis [38], self organizing feature maps);
- Spatial filtering, decorrelate the channels (common spatio-spectral pattern [33], multiresolution muscle synergy analysis [36]).

During the information decoding, control schemes has been used which use one dataset for training the system and build a way to map the synergies between the EMG signals and their effect on human limb. Most of the literature deals with the correct classification and execution of a certain number of pre-determined movements with linear (Linear Discriminant Analysis, Linear time invariant models, Non-negative matrix factorization) or not linear controller (Support vector machines, Artificial neural networks, Gaussian mixture models and K-nearest neighbors). These kind of techniques reach very accurate results in determining the actions performed by individuals from their EMG. Despite the good results, with the classification techniques it is only possible to determine the type of movement, not the trajectory. The majority of the studies focus on classification problems, yet there are some researches regarding regression techniques. Usually, the regression is used for a continuous and proportional control of humanoid robots. Complications due to cross-talk of the sensors and amplitude cancellation make this type of model generally used in test with a single degree of freedom, however recent studies have begun to use this kind of models also to simultaneously control more degrees of freedom.

The aim of the creation of subject-independent models is the extraction of specific characteristic of the EMG signal which let the merging of data coming from multiple subjects in order to obtain a general model and not one built ad-hoc for a certain subject. This goal isn't easy to achieve because of the great variability of EMG signals. Furthermore, the differences between different people give even worst results than the natural deterioration of the signal. In the last years has been proposed a few attempts for the realization of a subject-independent model which have shown promising results. Orabona et al. [57] proposed a way to give the patients a pre-trained model, which will be subsequently refined and adapted to the specific subject so as to shorten the length of the train phase. Castellini et al. [14] wanted to reach the same goal as well, but tried to reach it doing an analysis of cross-subject models: the model was built on data from a subject and then tested on data from another. Even Gibson et al. [29] have dealt with the problem of shorten the training phase required when a new user wants to use the system. Also in this case, the objective was to create a classifier able to recognize different movements. The results are very promising, the classifier is able to obtain an accuracy of 79 $\pm$ 6.6 %. Matsubara et al. [47] have developed a multi-user interface which can classify different movements using a bilinear model, achieving an accuracy of 73 %. Khushaba [37] wanted to obtain a multi-user interface capable of adapting to new users while maintaining good performances so he built a framework to implement style-independent transformations by using the canonical correlation analysis (CCA). The studies in this field are few and relatively recent, furthermore they have focused almost solely on the classification of the signal, but it still has been done no concrete study based on regression. This fact leaves chances for many other promising solutions.

The goal of this thesis is to develop a subject-independent probabilistic framework based on Robot-Learning techniques in order to obtain the continuous regression of the motion trajectories of the robot starting directly from human physiological data (sEMG) of several subjects. It has been built a model based on Gaussian Mixture, created from the EMG signals collected by different people, the goal is to create a general model based on the data of different subjects, which can be used by every person without any train or with a very short one. Once the model was built, has been tested the ability of the framework in the estimation of a joint bending angle by using as input EMG signals coming from several channels with a Gaussian Mixture Regression (Gaussian Mixture Regression (GMR)) algorithm. For the feature extraction has been used two widespread techniques: at first full wave rectification, filtering and normalization for the realization of a model which will be used offline, considering the whole signal representing the movement and at each instant associating the corresponding bending angle and the EMG values for each channel. Then, the model has been tested also with an alternative approach to overcome the limitation of the previous framework so as it is possible to use it also online: Wavelet Transform [18], in fact, Wavelet Transform allows to consider only a small window of the whole signal. By using this technique, the model is able to compute the synthesis value representing the signal just after a single raw sample has been collected.

A probabilistic framework based on Mixture of Gaussians (MoG) distributions only require a reduced number of parameters to be kept, resulting in lightweight models. Furthermore, a GMM/GMR probabilistic framework requires low training data to achieve good results. It also provides fast regression that perfectly matches with the use on an online application. This model has been tested on three different datasets involving different joints, from both upper and lower limbs.

Chapter 2

Physiological Signals

2.1 Electromyography

The Electromyography (EMG) is a widely used technique for recording the electrical activity produced by skeletal muscles. Skeletal muscles are a form of striated muscle tissue which is under the control of the somatic nervous system; that is to say, it is voluntarily controlled. It is one of three major muscle types, the others being cardiac and smooth muscle. As their name suggests, most skeletal muscles are attached to bones by bundles of collagen fibers known as tendons. The EMG measures muscle response or electrical activity in response to a nerve's stimulation of the muscle. Muscles are stimulated by signals from nerve cells called Motor Neurons (MN). This stimulation generates electrical activity in the muscle, which in turn causes contraction. The number of fibers per motor unit varies by the kind of movement for which it is intended. EMG signals derive from potential generated through muscular unit activation.

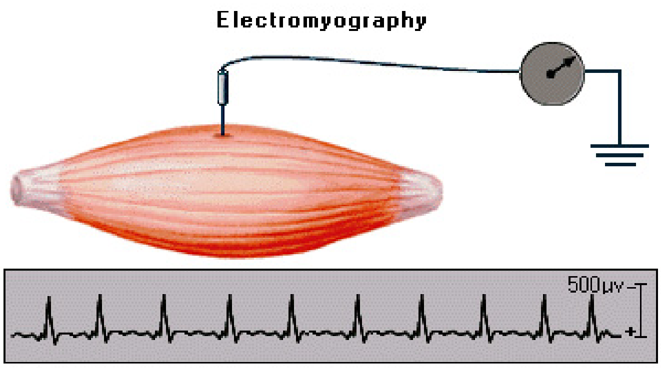

Figure 2.1: EMG acquisition

EMG testing has a variety of clinical and biomedical applications. EMG is used as a diagnostics tool for identifying neuromuscular diseases, or as a research tool for studying kinesiology and disorders of motor control. EMG signals are also used as a control signal for prosthetic devices such as prosthetic hands, arms and lower limbs.

The EMG can be recorded basically in two different ways: invasive (intramuscular EMG) or non-invasive (sEMG). Surface EMG assesses muscle function by recording muscle activity from the surface above the muscle on the skin. Surface electrodes are able to provide only a limited assessment of the muscle activity. sEMG can be recorded by a pair of electrodes or by a more complex array of multiple electrodes. More than one electrode is needed because EMG recordings display the potential difference (voltage difference) between two separate electrodes.

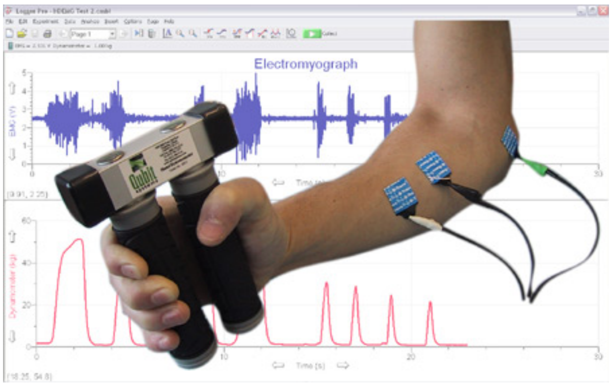

Figure 2.2: Surface EMG

Intramuscular EMG can be performed using a variety of different types of recording electrodes. The simplest approach is a monopolar needle electrode. This can be a fine wire inserted into a muscle with a surface electrode as a reference or two fine wires inserted into muscle referenced to each other as it is possible to see on Figure 2.1. After assessing resting and insertional activity, the electromyographer assess the activity of muscle during voluntary contraction. Each electrode track gives only a very local picture of the activity of the whole muscle. Because skeletal muscles differ in the inner structure, the electrode has to be placed at various locations to obtain an accurate study.

As previously said, EMG signals could be used to control prosthetic devices. In this case, the EMG signals are used to build a model of a certain movement. The intramuscular EMG is very invasive and could be extremely constraining in everyday life to control a prosthesis, so for this kind of application is better to use surface EMG (sEMG), although they give less accurate

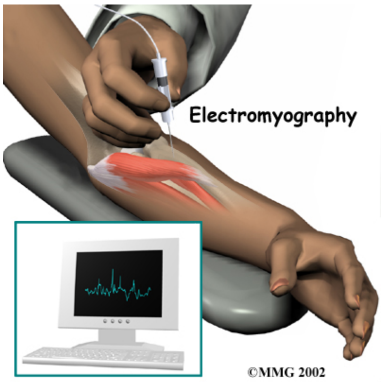

Figure 2.3: Intramuscular EMG

results than intramuscular EMG.

In biomechanics, three applications dominate the use of the surface EMG signal: its use as an indicator of the initiation of muscle activation, its relationship to the force produced by a muscle, and its use as an index of fatigue processes occurring within a muscle [22]. As an indicator of the initiation of muscle activity, the signal can provide the timing sequence of one or more muscles performing a task, such as during gait or in the maintenance of erect posture. Another important application of the EMG signal is to provide information about the force contribution of individual muscles as well as groups of muscles. Use in the individual muscle provides the greater attraction. The resultant muscular moment acting on a joint during a specific task is only in exceptionally rare cases due to one muscle. Thus, in the vast majority of cases of interest, the ability to determine noninvasively the force contribution of individual muscles provides an enormous advantage, particularly when biomechanical models are developed to describe the workings of a segment of the musculoskeletal system.

Electromyography is a seductive muse because it provides easy access to physiological processes that cause the muscle to generate force, produce movement and accomplish the countless functions that allow us to interact with the world around us. Moreover, if it is chosen to use the sEMG, the signal can be collected in a noninvasive way. The current state of surface electromyography is enigmatic. It provides many important and useful applications, but it has many limitations that must be understood, considered, and eventually removed. Nonetheless, judicious applications of known facts can ensure the fidelity of the EMG signal, reduce crosstalk, and provide sufficient stationarity in the signal; normalization of the signal amplitude may remove the influence of many other variables.

Possible limitations in the use of EMG signals are due to the fact that this kind of signals are very sensitive to variations, even in the same person, such as muscle fatigue, small shifts of the sensors, physiological (such as the skin impedance or the amount of fat), and psychological factors. The recorded signal from a certain subject deteriorates and changes considerably over time, so it is even harder build a subject-independent model.

The figure shows a schematic diagram of the factors that affect the EMG signals and how these factors interact among them and influences one another.

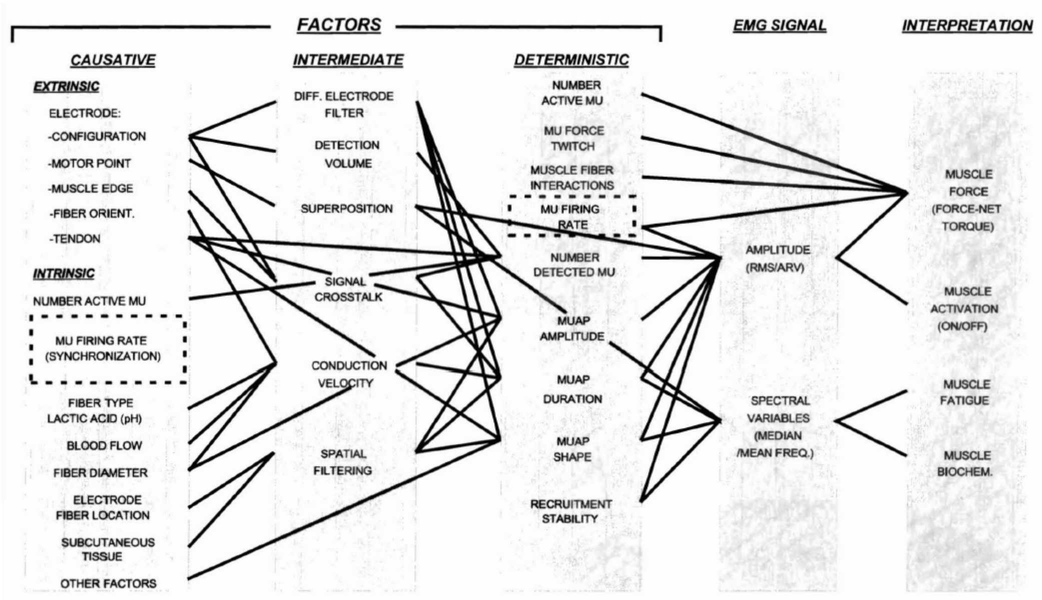

Figure 2.4: Factors that affect the EMG signals

EMG signals have to be pre-processed before they can be used. First of all the signal is decomposed, since EMG signals are essentially made up of superimposed motor unit action potentials (MUAPs) from several motor units, for a thorough analysis, the measured EMG signals can be decomposed into their constituent MUAPs. MUAPs from different motor units tend to have different characteristic shapes, while MUAPs recorded by the same electrode from the same motor unit are typically similar. Notably MUAP size and shape depend on where the electrode is located with respect to the fibers and so can appear to be different if the electrode moves position. Then the signal is rectified, rectification is the translation of the raw EMG signal to a single polarity frequency (usually positive). The purpose of rectifying a signal is to ensure the raw signal does not average zero, due to the raw EMG signal having positive and negative components. It facilitates the signals and process and calculates the mean, integration and the Fast Fourier Transform (FFT).

2.2 Dataset

The model has been tested with three different dataset. The first two included data from the knee joint, while the second one included data from the fingers and the wrist. Working with data from both lower and upper limbs gave us the possibility of testing our model and his goodness in different situations. Furthermore, the structure of the two dataset is quite different, in the first one few subjects performed many repetitions of the same movement, while in the latter many subjects repeated some movement a much smaller number of times. This let us analyze different aspects, for example for the first dataset we studied the number of repetitions needed for obtaining a good model, for the second one we observed the variability of the subjects.

2.2.1 First Dataset

In the first dataset [52], three healthy subjects (S1 - S3; age 30 $\pm$ 4; one female) were asked to naturally kick a ball from a sitting position (Figure 2.5). EMG signals were acquired with an active 8-channel wireless EMG system at 1000 Hz. The eight EMG electrodes were placed on the left leg of each subject in order to cover the principal muscular groups active during the kick task. The recorded muscles were: *Rectus femoris*, *Vastus lateralis*, *Vastus medialis*, *Tibialis anterior*, *Gastrocnemius lateralis*, *Gastrocnemius medialis*, *Biceps femoris caput longus*, *Peroneus longu*s (Figure 2.6). Synchronously to the EMG signals, it has been recorded the kinematics of the left leg by means of an optoelectronic system. Six retro reflecting markers were placed on the subjects leg and six infrared digital cameras recorded the marker positions at 60 Hz during the whole recording session. The kinematic of the knee-joint angle was computed from the position of the markers placed on the leg at each time instant t. Each person repeated the same movement about 70 times.

2.2.2 Second Dataset

This database contains 11 samples of subject with some abnormality in the knee previously diagnostic and 11 of healthy subjects. These data was collected with the equipment Biometrics DataLOG MWX8 electromiography and goniometry. The data were collected at the Batallón de sanidad

Figure 2.5: Performing of the kick movement

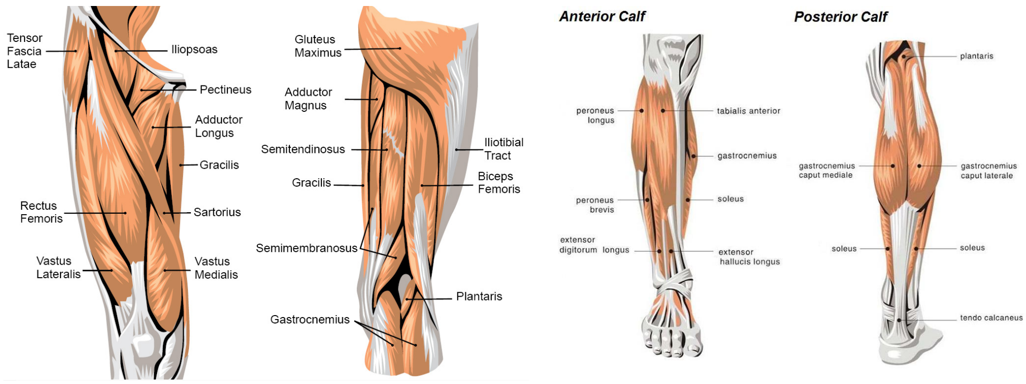

Figure 2.6: Muscles of the leg

(BASAN) Universidad Militar Nueva Granada, Bogotá.

22 male subjects, 11 with different abnormalities in the knee previously diagnostic has been studied in three different exercises for analysis in muscular behavior in relation with the knee, march, extension of the leg from sitting position and the flexion of the knee standing up.

The acquisition process was made with 4 electrodes on following muscles: *vastus internus*, *semitendinosus*, *biceps femoris* and *rectus femoris*. For recording the data has been used datalog MWX8 of Biometrics company, with 8 analog channels and 4 digital, 4 for sEMG samples and 1 for gonimetry. They were acquired directly from the equipment of MWX8, stored on internal memory and transmitted in real time to software Datalog on computer through bluetooth, with 14 bits of resolution and frequency sample of 1000Hz.

Has been used four electrodes for each data series. Each set contains 3 to 5 repetitions of each exercise proposed.

2.2.3 Third Dataset

The third dataset comes from the Ninapro (Non Invasive Adaptive Prosthetics) database [8] [30]. The goal of the Ninapro project is to build a robust and complete dataset, made with data collected from many different subjects, which perform several hand and wrist movements. The database aims at allowing worldwide research groups to study the relationship between sEMG, hand/arm kinematics, dynamics and clinical parameters using the same dataset, so as it is possible to compare the classification and regression performances with various techniques.

The database contains data obtained from 40 intact subjects (28 males, 12 females; 34 right handed, 6 left handed; age 29.9 $\pm$ 3.9). Each subject performed 50 different movements shown in Figure 2.9, each one was repeated six times. Different kind of movement were performed, including the flexion of the fingers, the movement of the wrist and several every day activities, like the grasping of different objects. In this study we focused on the wrist flexion and extension, because of his analogy with the knee bending.

During the exercises performed using the Cyberglove II, the intact subjects were asked to mimic movies of movement shown on the screen of the laptop with their right hand. Hand kinematics was measured using a 22-sensor CyberGlove II (CyberGlove Systems LLC, www.cyberglovesystems.com), a motion capture data glove, instrumented with joint-angle measurements. In addition to the CyberGlove, a standard commercially available 2-axis IS40 inclinometer (Fritz Kübler GmbH, www.kuebler.com) was fixed to the subject's wrist to measure the wrist orientation. This inclinometer has a range

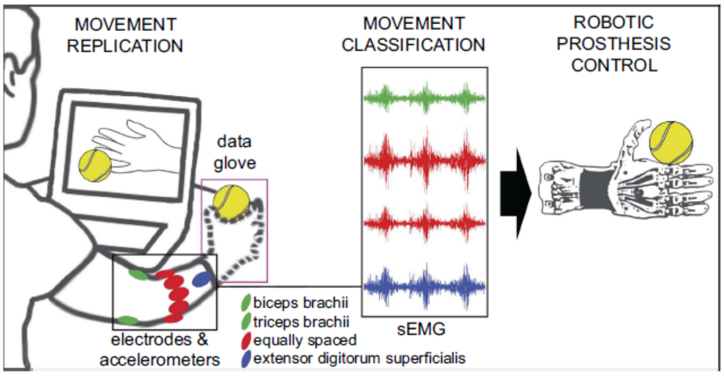

Figure 2.7: Acquisition procedure scheme. The subjects are asked to mimic movies of movement shown on the screen of the laptop. The sEMG signal is recorded through up to 12 electrodes and can be used to test methods to control robotic hand prostheses naturally.

(a) Movement Exercise

(b) Force Exercise

Figure 2.8: Placement of the 12 electrodes on the arm and Acquisition setup for the (a) discrete movement and (b) force exercises.

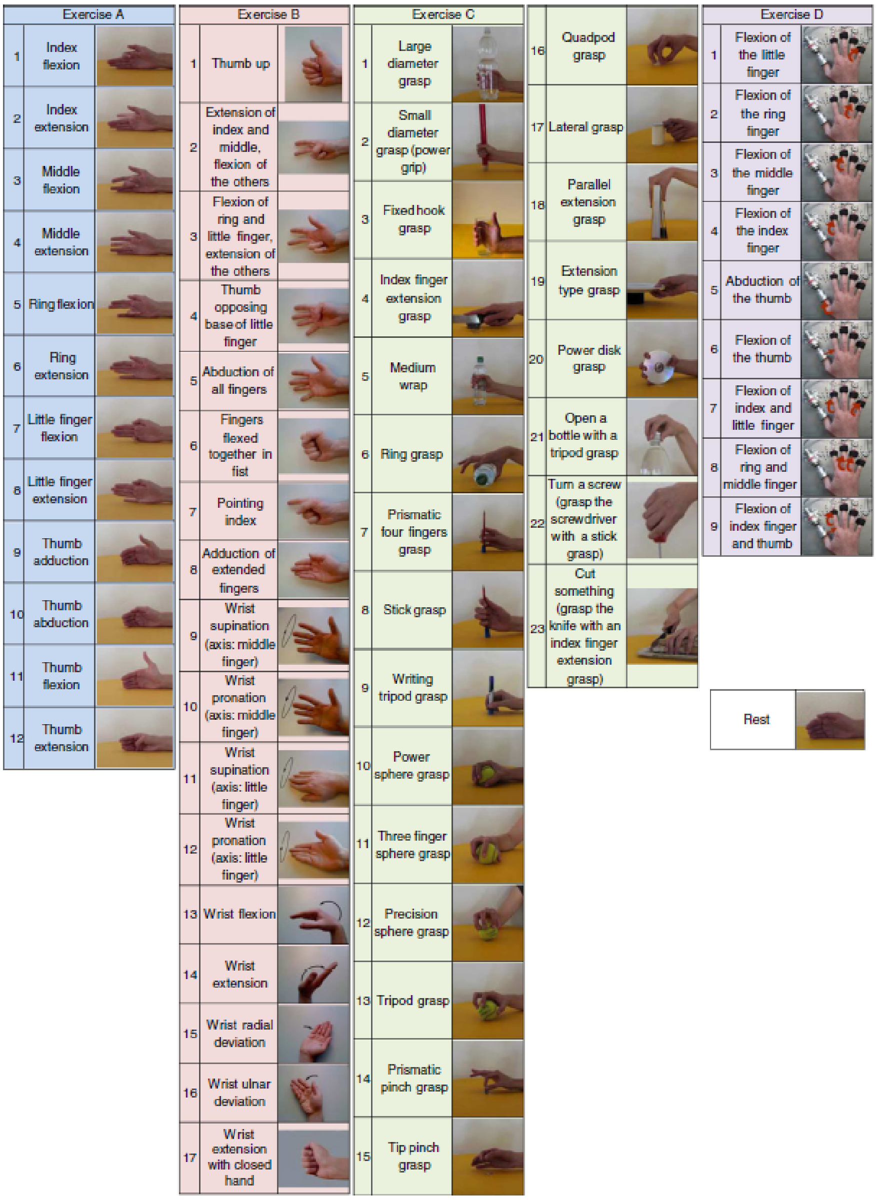

Figure 2.9: Movements and force patterns divided by exercise. Exercise A (light blue): 12 basic movements of the fingers; Exercise B (red): 8 isometric and isotonic hand configurations and 9 basic movements of the wrist; Exercise C (green): 23 grasping and functional movements (everyday objects are presented to the subject for grasping, in order to mimic daily-life actions); Exercise D (purple): 9 force patterns; Rest position (white).

of 120 and a resolution of less than 0.15. Hand dynamics was measured using the Finger-Force Linear Sensor (FFLS), employing strain gauge sensors to detect flexion and extension forces of all fingers, plus abduction and adduction of the thumb. Muscular activity was measured using Delsys double differential sEMG electrodes. sEMG signals were sampled at a rate of 2 kHz with a baseline noise of less than 750 nV RMS. These electrodes also integrate a 3-axes accelerometer sampled at 148 Hz. Eight electrodes were equally spaced around the forearm at the height of the radio-humeral joint; two electrodes were placed on the main activity spots of the *flexor digitorum superficialis* and of the *extensor digitorum superficialis*, two electrodes were also placed on the main activity spots of the *biceps brachii* and of the *triceps brachii*. Data from the CyberGlove were transmitted over a Bluetooth-tunneled serial port at slightly less than 25 Hz; data from the inclinometer and the FFLS were acquired at 100 Hz using a National Instruments data acquisition card (NI-DAQ PCMCIA 6024E, 12-bit resolution); the Delsys base station received the sEMG and accelerometer streams via an ad-hoc wireless network and relayed the data via a standard USB connection to the laptop. Each data sample provided by each sensor was associated to an accurate timestamp using Windows performance counters.

Chapter 3

Signal Analysis

In order to be processed, the EMG signals have to be elaborated so as to highlight the muscular activation during the kick tasks. For the online and offline EMG elaboration has been used two different approaches to process EMG signals. While the elaboration for the offline model is quite standard and extensively used and described in literature, for the online elaboration the feature extraction has been performed through Wavelet Transform [18].

3.1 Basic Signal Analysis

For the stationary model, EMG data was processed by means of signal rectification and smoothing in order to highlight the muscular activation during the kick tasks. This method is widely exploited in literature to denoise EMG signals and extract useful information and features for classification purposes. In brief:the raw 8-channel EMG signals were high-pass filtered (zero-phase digital filter, 4 th order Butterworth, 30 Hz cut-off frequency) to remove artifacts; then, signals were full-wave rectified and low-pass filtered (4 th order Butterworth, 6 Hz cut-off frequency). Signal processing was performed in MATLAB 8.0. For the second dataset, the data were synchronized (all streams were supersampled to the highest sampling frequency 2 kHz or 100 Hz, using linear interpolation or nearest-neighbour interpolation), relabelled (erroneous movement labels have been corrected by applying a generalized likelihood ratio algorithm offline, which realigns the movement boundaries by maximizing the likelihood of a rest-movement-rest sequence) and filtered (shift the signal to a "standard" one and apply a dynamic time warping to adjust the amplitude of the various signals with a dynamic time warping algorithm). The results of this operation are shown in Figure 3.1.

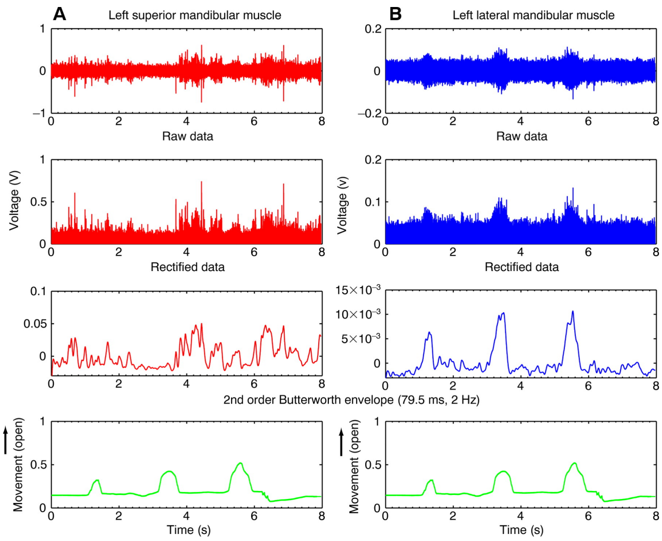

Figure 3.1: Example of a basic signal analysis on a EMG signal.

3.2 Wavelet

The offline elaboration of the EMG signals is very limiting, because in this way the model is not usable for real life applications. It is very interesting the possibility of elaborating the signal online, so as to let the exoskeleton moving in an integral way with the subject leg, or let the prosthesis follow almost immediately the human thought.

Regrettably, training a Gaussian Mixture Model only with the EMG signals and the angle of the joint without using the time, yields to weak results. The major drawback is that both slow and fast changing properties of sEMG signals are relevant to the analyzing task. Slow variations provide information related to the body movements and tissue properties while the fast variations of signal are useful to understand the muscle activity and motor recruitment itself.

Some widespread techniques adopted for pattern recognition are Fourier Transform [73], Integral Absolute Value (IAV), variance and zero crossing [43], Mean Average Value (MAV) [46], Rooted Mean Square (RMS), Mean Power Frequency (MPF) [41], or as proposed in [62] full wave rectification, filtering and normalization. The major drawback of these transformation methods, especially fast and short-term Fourier Transform, is that they assume signal to be stationary [35]. Alternative approaches based on Wavelet Transform have more effective results since EMG signals are nonstationary. Daubechies adopted Wavelet Transform to analyze time series that contain non-stationary power at many different frequencies [21]. Laterza [42] showed thatWavelet Transform is a valuable alternative to represent time frequency signals. His work highlighted several advantages. In fact, Wavelet Transform are a linear multiresolution representation of the original signal without crossterms affections. While Guglielminotti [31] found out the good matching properties between an EMG signal and its Wavelet shapes. Recent works reinforced advantages in using Wavelet Transform for EMG analysis [61] [58]. In particular, Chowdhury [15] emphasized the good results obtained when adopting Daubechies functions by investigating and analyzing various research studies on Wavelet Transform. Moreover, Wavelet information can be synthesized to obtain a more compact representation by using statistical features as stated by Subasi [68] [67]. Advantages of this technique are linearity, multiresolution representation and being unaffected by cross terms. Better results are obtained when wavelet shape matches the shape of the EMG as it achieves a good energy localization in the time-scale plane.

3.2.1 Wavelet Transform

A wavelet is a wave-like oscillation with an amplitude that begins at zero, increases, and then decreases back to zero. It can typically be visualized as a "brief oscillation" like one might see recorded by a seismograph or heart monitor. Generally, wavelets are purposefully crafted to have specific properties that make them useful for signal processing. Wavelets can be combined, using a *reverse, shift, multiply and integrate* technique called convolution, with portions of a known signal to extract information from the unknown signal.

Wavelet Transform [18], in a similar way as Fourier Transform, has been used in numerous applications about processing and analysis of signals. Both these tools are used to look at the frequency components, anyway Wavelet Transform preserves the temporal aspect of the signal without resolution limits in frequency. Therefore, Wavelet analysis is able to extract signal information regarding both time and frequency. From a mathematical point of view, Wavelet transform is similar to Fourier transform with the exception that instead of using a basis composed by sine and cosine it uses particular functions that satisfy certain mathematical rules. Moreover, wavelet analysis allows to locate signal information both in time and in frequency and can be applied in a useful manner in a wide window of time as well as in a closer one.

Wavelet transform is a spectral estimation technique in which any general function can be expressed as an infinite series of wavelets. The basic idea underlying wavelet analysis consists of expressing a signal as a linear combination of a particular set of functions (wavelet transform, Wavelet Transform (Wavelet Transform)) 3.1, obtained by shifting and dilating one single function called a *mother wavelet* ($\psi(t)$) by means of a scaling function ($\phi(t)$).

$$f[n] = \frac{1}{\sqrt{M}} \sum_{k} W_{\phi}[j_0, k]\phi_{j_0,k}[n] + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_{k} W_{\psi}[j, k]\psi_{j,k}[n] \qquad (3.1)$$

The decomposition of the signal leads to a set of coefficients called wavelet coefficients. Therefore the signal can be reconstructed as a linear combination of the wavelet functions weighted by the wavelet coefficients. In order to obtain an exact reconstruction of the signal, adequate number of coefficients must be computed. The Wavelet Transform can be thought of as an extension of the classic Fourier transform, except that, instead of working on a single scale (time or frequency), it works on a multi-scale basis. This multi-scale feature of the Wavelet Transform allows the decomposition of a signal into a number of scales, each scale representing a particular coarseness of the signal under study.

The mother wavelet is scaled and translated to provide multi-resolution analysis: wide wavelets are used for low frequencies, while narrow wavelets do the work at high frequencies. Wavelet transform method is divided into two types: Discrete Wavelet Transform (DWT) and continuous wavelet transform (Continuous Wavelet Transform (CWT)). DWT was selected in this study because of the concentration in real-time engineering applications. DWT is a technique that iteratively transforms an interested signal into multi-resolution subsets of coefficients. Like the conventional time-frequency analysis, the DWT transforms the EMG signal with a suitable wavelet basis function (WF).

Wavelet transform is considered a flexible and general technique because there is a wide variety of wavelet functions which could be used. Moreover, anyone can design his own wavelet function according to his needs, provided that some mathematical constraints are satisfied. Depending on the kind of signal, one can use Discrete Wavelet Transform (DWT) or Continuous Wavelet Transform (CWT).

As a mathematical tool, wavelets can be used to extract information from many different kinds of data. Sets of wavelets are generally needed to analyze data fully. A set of "complementary" wavelets will decompose data without gaps or overlap so that the decomposition process is mathematically reversible. Thus, sets of complementary wavelets are useful in wavelet based compression/decompression algorithms where it is desirable to recover the original information with minimal loss.

Wavelet Transform lead us different frequency components getting the most significant information of EMG signal spending less time and computation.

In the feature extraction stage, numerous different methods can be used so that several different features can be extracted from the same raw data.

All wavelet transforms may be considered forms of time-frequency representation for continuous-time (analog) signals and so are related to harmonic analysis.

A complete and detailed description of Wavelet Transform and its supporting theory can be found in [18] and [64].

3.2.2 Choice of the best mother wavelet

A previous study [72] shows that the selection of the mother wavelet is particularly important because every mother wavelet yields to different results even when applied to the same signal. Chowdhury [15] successfully used Daubechies family (db) function to analyze sEMG signals. His research focused on the processing of sEMG and its use in different applications. The signal was process by means of some specific functions (db2, db4, db6, db44 and db45) at decomposition level 4 in order to maintain the maximum amount of information. In a similar study, Phinyomark [58] was able to find good results by using db7 as mother wavelet. Wavelet Transform leads to a series of values representing the considered signal through a specific mother wavelet. In this study, we would like to synthesize the information provided by Wavelet Transform in a single value representing the wavelet decomposition as already tested in several works [26][11][13]. Mean Average Value (MAV) and Rooted Mean Square (RMS) methods have been selected due the good results reached in the literature.

By looking at the good performances in both accuracy and time obtained for subject-specific cases by Valentini *et al* [72], has been selected the db2 mother wavelet from the Daubechies family for representing the input EMGs through a series of M coefficients. Synthesizing the coefficients provided by Wavelet Transform (Wavelet Transform) to a single value representing the wavelet decomposition allows us to compare different signals. The synthesis function should guarantee a certain level of smoothness in order to avoid suddenly changes from one instant to another and being fast enough to be computed online. MAV 3.2 represents a good candidate since the results achieved in [72].

$$MAV = \frac{1}{M}\sum_{k=1}^{M}|x_k| \tag{3.2}$$

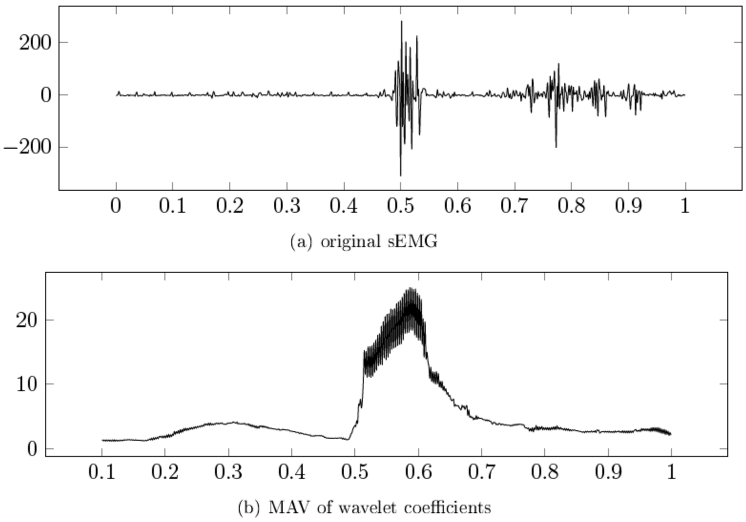

Figure 3.2: Row EMG signal and after the feature extraction by means of Wavelet Transform

3.3 Smoothing and Normalization

Usually the wavelet transform applied to an EMG channel are very jagged. A model built on this kind of data gives poor results because the great variability of the signals results in poor model performances. The Wavelet Transform of the EMG channels are smoothed and normalized in order to obtain better and more robust models.

The smoothing function used is provided by MATLAB which smooths the data using a moving average filter. The average of the S = 10% of the data points available within the windows is computed to smooth the data at the instant t. This process is equivalent to lowpass filtering with the response of the smoothing given by

$$y_s(t) = \frac{1}{2N+1}\sum_{k=1}^{S} y(t-k) \tag{3.3}$$

Where $y_s(t)$ is the smoothed value for the ith data point, N is the number of neighboring data points on either side of $y_s(t)$, and $2N+1$ is the span. A moving average filter smooths data by replacing each data point with the average of the neighboring data points defined within the span.

The smoothing function used smooths the data using the loess method. It performs a local regression using weighted linear least squares and a 2nd degree polynomial model.

In the Figure 3.3 is shown the Wavelet Transform of a EMG channel before and after the smoothing.

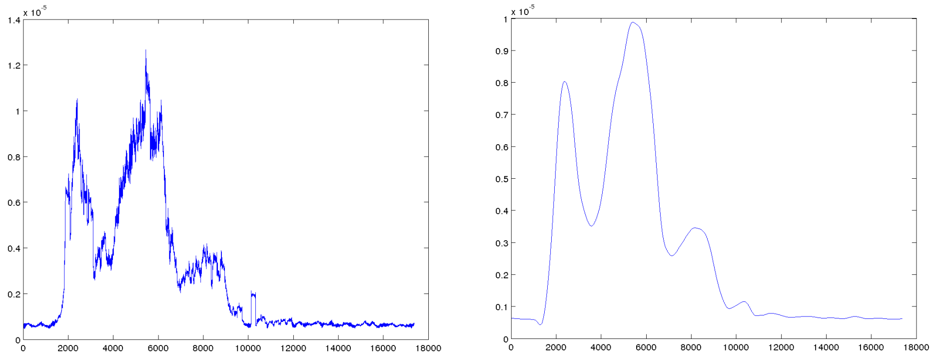

Figure 3.3: Wavelet Transform of a EMG channel before and after the smoothing.

Often the signals of a certain subject have different amplitude than the others, a normalization process has been introduced to regularize the EMG signals. The normalization phase has been executed offline for the training dataset by using the relative maximum within the processed trial. During the testing procedure, the mean of the maximums collected during the training has been used as normalization factor, when no data were available on a subject. While we used the mean of the maximums within the trials of the specific subject, with at least 10 attempts collected.

3.4 Dynamic Time Warping

Shift is a common occurrence in data analysis. Many analytical techniques yield data where the same phenomena may yield variations at different positions or may have different durations depending on the specific analytical conditions. Analogously, the measurements for the single samples can have different time scales or axes, or the sample vectors may have different lengths (e.g. different batch lengths in industrial processes). Warping is one of the numerous pretreatment methods that have been proposed to correct for shifts, conditioning data for multilinear models for exploratory purposes as well as quantitative determination by alignment of the shifted variables. If data are not brought to a form where the observed variables of the samples under analysis express similar attributes, the required assumption for using bi- and multilinear modeling, namely that like variables represent similar phenomena in all samples, is violated.

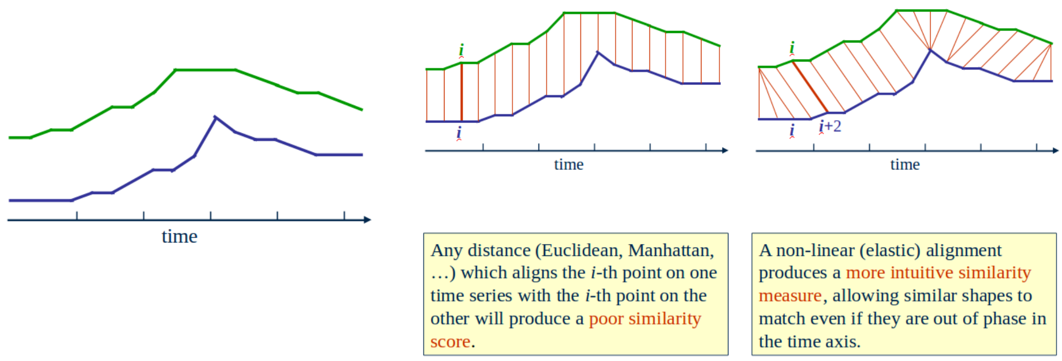

Figure 3.4: Graphic example of the DTW algorithm.

Two different warping algorithms have received much attention in recent years for the alignment of time trajectories. The first method, termed DTW, was initially devised for aligning frequency spectra of words pronounced by different speakers for recognition purposes. The more recent approach for aligning signals, termed Correlation Optimized Warping (COW), was proposed in 1998 as a means to correct chromatograms for shifts in the time axis prior to multivariate modeling [71].

3.4.1 DTW Algorithm

Dynamic time warping nonlinearly warps the two trajectories in such a way that similar events are aligned and a minimum distance between them is obtained. The algorithm in recent years it has found application in chromatography, in batch process monitoring and in gene expression studies. In time series analysis, DTW is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. For instance, similarities in walking patterns could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and graphics data. Indeed, any data which can be turned into a linear sequence can be analyzed with DTW. A well known application has been automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. Also it is seen that it can be used in partial shape matching application.

In general, DTW is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restrictions. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in time series classification. Although DTW measures a distance-like quantity between two given sequences, it doesn't guarantee the triangle inequality to hold.

Computing the DTW requires $O(N^2)$ in general and, unfortunately, cannot be used in real time, because it needs all the data at the beginning of the computation.

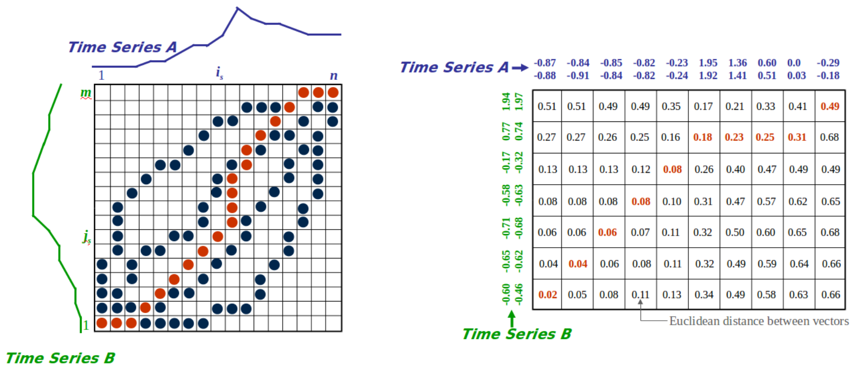

Figure 3.5: A graphic and a numeric example of the DTW algorithm.

In our case the DTW algorithm has been tested with several combination of parameters to find out the best result. An example is shown in Figure 3.6

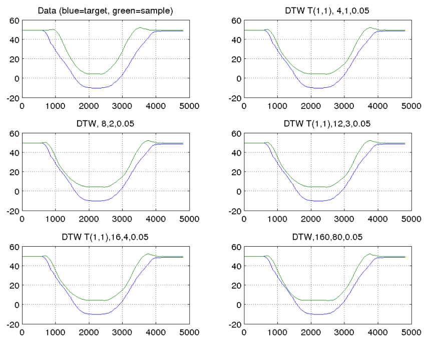

Figure 3.6: DTW on the angle of the knee with different parameters.

Chapter 4

Gaussian Mixture

At the IAS-Lab (Intelligent Autonomous System Laboratory of University of Padua), Michieletto et al. [52] developed a probabilistic framework which estimates the knee bending angle based on a GMM-GMR. This framework was able to correlate EMG signals to the corresponding joint angles matching them through acquisition time.

This thesis aims to explore the use of a Gaussian Mixture Model (GMM) for estimation of single-joint angle with data collected from multiple subjects. A Gaussian Mixture Regression (GMR) technique is then used to retrieve the data from the trained model. This approach enables an autonomous extraction of the task-related information encoded in EMG signals, without loss of generality. Moreover, such a probabilistic framework based on Mixture of Gaussians (MoG) distributions only require a reduced number of parameters to be kept, resulting in lightweight models. In the past, several groups exploited MoG, and in particular GMM, as a method to encode EMG information. Chu et al. [17] were able to recognize EMG patterns related to hand motion in ten different subjects by means of a GMM classifier. Suresh et al. [69] exploited a combination of EMG and GMM as a signature for people identification. Furthermore, a GMM/GMR probabilistic framework requires less training data to achieve good results and provides faster regression with respect to other echniques usually used in the field of EMG pattern recognition (e.g., Neural Networks (NN) [75]). In particular, Fukuda et al. [27] proposed a Feedforward Neural Network (FNN) that achieves performances comparable to GMM, but with a large manual tuning effort and with the drawback of high sensitivity to the specific case study. For these reasons, a GMM probabilistic framework has been already widely adopted in robotic applications [50], Robot Learning from Demonstration (RLfD) [10] [13], sound categorization [59], grasp modeling [60] and action recognition [44].

4.1 Gaussian Mixture Model

Gaussian Mixture Model is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) [23] algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.

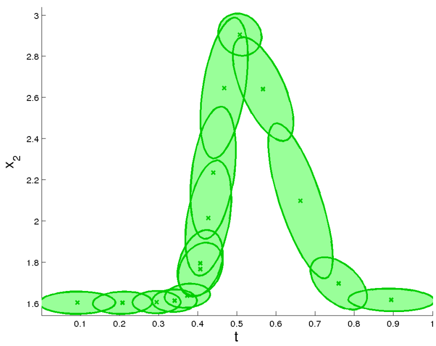

Figure 4.1: Example of GMM. Every green oval is a n-dimensional gaussian

The stochastic approach of the GMM is the ideal to address the high variability of the input EMG signals. Information extracted from EMG was used as input of a Gaussian Mixture Model (GMM) to estimate its correlation with the knee bending angle α.

The aim of GMM is to obtain the weighted sum of K Gaussian components which best approximates the input dataset representing the set of kick trials used for the training. For example, in the first dataset the total number of data samples was $N = nT$, where n is the number of trials used to train the system, and $T = 2000$ is the number of observations acquired during each trial. A single data in input at the framework in the offline case is described in Equation 4.1.

$$\zeta_j = \{t, \xi(t), \alpha(t)\} \in \mathbb{R}^D \quad \xi(t) = \{\xi_c, f(t)\}_{c=1,\ldots,C} \tag{4.1}$$

where:

- $t \in \mathbb{R}$ is the time elapsed from the beginning of the trial (ms);
- $\xi_c(t) \in \mathbb{R}$ is the c^{th} EMG channel at the time instant t and f is one of MAV,RMS or VAR;

- $\xi(t) \in \mathbb{R}^C$ is the set of considered channels, $1 \leq |\xi| \leq 8$, at the time instant t;
- $\alpha(t) \in \mathbb{R}$ is the knee bending angle at the time instant t;
- $3 \leq D \leq 10$ is the dimensionality of the problem.

In the online case, a single data in input at the framework is described in Equation 4.2.

$$\zeta_j = \{t, \xi, \alpha\} \in \mathbb{R}^D \quad \xi = \{\xi_c\}_{c=1,\dots,C} \tag{4.2}$$

where:

- $\xi_c \in \mathbb{R}$ is the c^{th} EMG channel;
- $\xi \in \mathbb{R}^C$ is the set of considered channels, $1 \leq |\xi| \leq 8$;
- $\alpha \in \mathbb{R}$ is the knee bending angle;
- $2 \leq D \leq 9$ is the dimensionality of the problem.

4.2 Expectation Maximization

The GMM was trained through the Expectation-Maximization (EM) algorithm, resulting in a probability distribution of the train dataset later used to perform the regression of the knee angle.

In statistics,EM algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

The EM algorithm is used to find (locally) maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Typically these models involve latent variables in addition to unknown parameters and known data observations. That is, either there are missing values among the data, or the model can be formulated more simply by assuming the existence of additional unobserved data points. For example, a mixture model can be described more simply by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component that each data point belongs to.

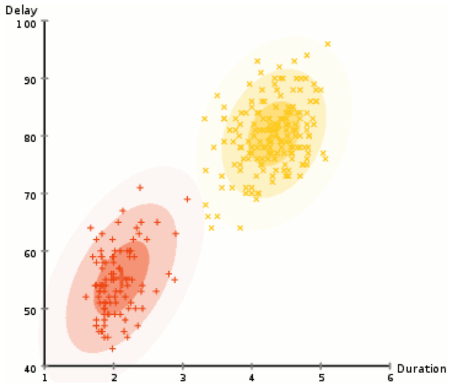

Figure 4.2: Example of the EM algorithm. The red and yellow ovals show how the algorithm adapt to the data (the red and yellow crosses)

The EM algorithm proceeds from the observation that the following is a way to solve these two sets of equations numerically. One can simply pick arbitrary values for one of the two sets of unknowns, use them to estimate the second set, then use these new values to find a better estimate of the first set, and then keep alternating between the two until the resulting values both converge to fixed points. It's not obvious that this will work at all, but in fact it can be proven that in this particular context it does, and that the derivative of the likelihood is (arbitrarily close to) zero at that point, which in turn means that the point is either a maximum or a saddle point. In general there may be multiple maxima, and there is no guarantee that the global maximum will be found. Some likelihoods also have singularities in them, i.e. nonsensical maxima. For example, one of the "solutions" that may be found by EM in a mixture model involves setting one of the components to have zero variance and the mean parameter for the same component to be equal to one of the data points.

The Expectation-Maximization algorithm iteratively estimates the optimal parameters $\theta = (\pi_k, \mu_k, \Sigma_k)$ that characterizes the K mixtures composing the GMM. The algorithm can be separated in two cyclic phases: Expectation and Maximization. The EM loop stops when the increment of the log-likelihood $\mathcal{L} = \sum_{j=1}^{N} \log\left(p\left(\zeta_j|\theta\right)\right)$ at each iteration becomes smaller than a defined threshold ϵ, given by $\frac{\mathcal{L}(t+1)}{\mathcal{L}(t)} < \epsilon$.

The Expectation step (E-step) is described as follows:

$$p_{k,j}(t+1) = \frac{\pi_k(t)\mathcal{N}(\zeta_j; \mu_k(t), \Sigma_k(t))}{\sum_{i=1}^{K} \pi_i(t)\mathcal{N}(\zeta_j; \mu_i(t), \Sigma_i(t))} \tag{4.3}$$

and the Maximization step (M-step) is computed according to:

$$\pi_k(t+1) = \frac{1}{N}\sum_{i=1}^{N} p_{k,j}(t+1) \tag{4.4}$$

$$\mu_k(t+1) = \frac{\sum_{i=1}^{N} p_{k,j}(t+1)\xi_j}{\sum_{i=1}^{N} p_{k,j}(t+1)}$$

$$\Sigma_k(t+1) = \frac{\sum_{i=1}^{N} p_{k,j}(t+1)(\zeta_j - \mu_k(t+1))(\zeta_j - \mu_k(t+1))^\top}{\sum_{i=1}^{N} p_{k,j}(t+1)}$$

The algorithm optimizes the parameters of the K Gaussian components by maintaining a monotone increasing likelihood during the local search of the maximum. This approach enables an autonomous extraction of the kick characteristic EMG signal while still maintaining an appropriate generalization.

Finally, the resulting probability density function is computed:

$$p(\zeta_j) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\zeta_j; \mu_k, \Sigma_k) \tag{4.5}$$

where:

- π_k are priors probabilities;
- $\mathcal{N}(\zeta_j; \mu_k, \Sigma_k)$ are Gaussian distribution defined by μ_k and Σ_k, respectively mean vector and covariance matrix of the k-th distribution.

The main drawback in the learning process lies in the EM requirement of a prior specification for the model complexity (i.e., the number of components K). On one hand, an overestimation of this parameter might lead to overfitting and, consequently, to a poor generalization; on the other hand, an underestimation will result to poor regression performances. To deal with this issue we introduced an entropy based selection of the best number of components, K, in the GMM.

Several entropy based model selection techniques has been proposed in literature (e.g. Bayesian Information Criterion (BIC) [63], Akaike Information Criterion (AIC) [4], Minimum Description Length (MDL) [9] and Minimum Message Length (MML) [74]). Although, in [16] the authors proposed a specific criteria to estimate the value of the K parameter in the case of EMG signals, in this work we preferred a more standard approach based on BIC. In our experiments the whole learning process has been repeated with different

GMM complexities by using BIC (Equation 4.6) as index of model quality with respect to the number of components K.

$$S_{BIC} = -2\mathcal{L} + n_p \log N \tag{4.6}$$

where:

- $\mathcal{L} = \sum_{j=1}^{N} \log\left(p\left(\zeta_j|\theta\right)\right)$ is the log-likelihood for the considered model θ;
- $n_p = (K-1) + K(D + \frac{1}{2}D(D+1))$ is the number of free parameters required for a mixture of K components with full covariance matrix.

The log-likelihood measures how well the model fits the data, while the second term is introduced to avoid data overfitting and maintain the model general enough. In our experiments the best BIC value was obtained with $K = 15$ components.

4.3 Regression

In statistics, regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables. In all cases, the estimation target is a function of the independent variables called the regression function. In regression analysis, it is also of interest to characterize the variation of the dependent variable around the regression function which can be described by a probability distribution.

Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships.

Many techniques for carrying out regression analysis have been developed. Familiar methods such as linear regression and ordinary least squares regression are parametric, in that the regression function is defined in terms of a

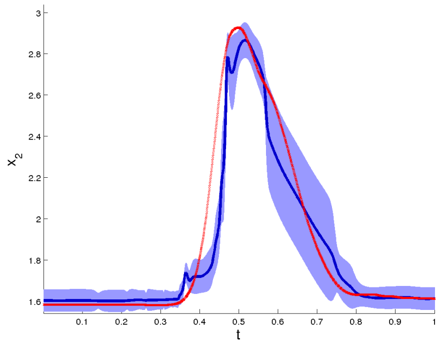

Figure 4.3: Example of regression on data. The red line represent the input data, while the blue line represent the smooth estimated movement computed starting from EMG signals.

finite number of unknown parameters that are estimated from the data. Nonparametric regression refers to techniques that allow the regression function to lie in a specified set of functions, which may be infinite-dimensional.

The performance of regression analysis methods in practice depends on the form of the data generating process, and how it relates to the regression approach being used. Since the true form of the data-generating process is generally not known, regression analysis often depends to some extent on making assumptions about this process. These assumptions are sometimes testable if a sufficient quantity of data is available. Regression models for prediction are often useful even when the assumptions are moderately violated, although they may not perform optimally. However, in many applications, especially with small effects or questions of causality based on observational data, regression methods can give misleading results.

In a narrower sense, regression may refer specifically to the estimation of continuous response variables, as opposed to the discrete response variables used in classification. Typical applications of the analysis of EMG signals have the purpose of classify different movements. The Gaussian Mixture Regression (GMR) has been used to retrieve a smooth generalized version of the signal encoded in the associated GMM. So that, the conditional expectation of the joint angle $\hat{\alpha}$ is calculated from the consecutive temporal value t and the EMG signals ξ known a priori. As we already said, the k-th Gaussian component is defined by the parameters (π_k, μ_k, Σ_k), where:

$$\mu_k = \{\mu_{p,k}\, \mu_{\alpha,k}\} \qquad \Sigma_k = \begin{bmatrix} \Sigma_{p,k} & \Sigma_{p\alpha,k} \\ \Sigma_{\alpha p,k} & \Sigma_{\alpha,k} \end{bmatrix} \tag{4.7}$$

with μ_p and Σ_p respectively the mean and the covariance of the known a priori information. The conditional expectation and its covariance can be estimated offline respectively using Equation 4.8 and 4.9.

$$\hat{\alpha} = E\left[\alpha\left|t,\xi\right.\right] = \sum_{k=1}^{K} \beta_k \hat{\alpha}_k \tag{4.8}$$

$$\hat{\Sigma}_s = Cov\left[\alpha\left|t,\xi\right.\right] = \sum_{k=1}^{K} \beta_k^2 \hat{\Sigma}_{\alpha,k} \tag{4.9}$$

where:

- $\beta_k = \frac{\pi_k \mathcal{N}\left(t,\xi_c\middle|\mu_{p,k},\Sigma_{p,k}\right)}{\sum_{j=1}^{K}\mathcal{N}\left(t,\xi_c\middle|\mu_{p,j},\Sigma_{p,j}\right)}$ is the weight of the k-th Gaussian component through the mixture;
- $\hat{\alpha}_k = E\left[\alpha_k\left|t,\xi\right.\right] = \mu_{\alpha,k} + \Sigma_{\alpha p,k}\left(\Sigma_{p,k}\right)^{-1}\left(\{t,\xi\} - \mu_{p,k}\right)$ is the conditional expectation of α_k given $\{t,\xi\}$;
- $\hat{\Sigma}_{\alpha,k} = Cov\left[\alpha_k\left|t,\xi\right.\right] = \Sigma_{\alpha,k} + \Sigma_{\alpha p,k}\left(\Sigma_{p,k}\right)^{-1}\Sigma_{p\alpha,k}$ is the conditional covariance of α_k given $\{t,\xi\}$.

While, for the online analysis of the movement, the conditional expectation and its covariance can be estimated respectively using Equation 4.10 and 4.11.

$$\hat{\alpha} = E\left[\alpha\left|\xi\right.\right] = \sum_{k=1}^{K} \beta_k \hat{\alpha}_k \tag{4.10}$$

$$\hat{\Sigma}_s = Cov\left[\alpha\left|\xi\right.\right] = \sum_{k=1}^{K} \beta_k^2 \hat{\Sigma}_{\alpha,k} \tag{4.11}$$

where:

- $\beta_k = \frac{\pi_k \mathcal{N}\left(\xi_c\middle|\mu_{p,k},\Sigma_{p,k}\right)}{\sum_{j=1}^{K}\mathcal{N}\left(\xi_c\middle|\mu_{p,j},\Sigma_{p,j}\right)}$ is the weight of the k-th Gaussian component through the mixture;
- $\hat{\alpha}_k = E\left[\alpha_k\left|\xi\right.\right] = \mu_{\alpha,k} + \Sigma_{\alpha p,k}\left(\Sigma_{p,k}\right)^{-1}\left(\{\xi\} - \mu_{p,k}\right)$ is the conditional expectation of α_k given $\{\xi\}$;
- $\hat{\Sigma}_{\alpha,k} = Cov\left[\alpha_k\left|\xi\right.\right] = \Sigma_{\alpha,k} + \Sigma_{\alpha p,k}\left(\Sigma_{p,k}\right)^{-1}\Sigma_{p\alpha,k}$ is the conditional covariance of α_k given $\{\xi\}$.

Thus, the generalized form of the motions $\hat{\zeta} = \{t,\xi,\hat{\alpha}\}$ ($\hat{\zeta} = \{\xi,\hat{\alpha}\}$ for the online analysis) required only the weight, mean and covariance of the Gaussian components calculated through the EM algorithm.

4.4 Error estimation

In order to evaluate the effectiveness of the GMM-based system has been used two different techniques. Originally we exploited the Normalized Mean Square Error (NMSE) in order to evaluate the effectiveness of the GMM-based system. The selected function measures the Goodness of Fit (GoF) by using as metric the NMSE between test and reference data, in our case $\hat{\alpha}$ (the data estimated through the GMR) and α (the angle calculated by means of the motion capture system):

$$\mathrm{GoF}_{NMSE}(t) = 1 - \frac{MSE(\hat{\alpha}(t))}{MSE(E[\alpha(t)])} \tag{4.12}$$

with:

- t, temporal instant from the beginning of the trial (ms);
- $\hat{\alpha}(t)$, estimated angle at the instant t;
- $MSE(x(t)) = \|\alpha(t) - x(t)\|^2$, where $\alpha(t)$ angle calculated through the motion capture at the instant t;
- $E[\alpha(t)]$, mean along the time of the angles given by the motion capture.

By using this formula, the GoF costs vary between $-\infty$ (bad fit) to 1 (perfect fit). The idea is to obtain a value representing the goodness of fit and not the error from the real value. Moreover, in this case, zero represents the value reached from a straight line in fitting the reference.

Later we changed metric, using the correlation coefficient, also known as the Pearson product-moment correlation coefficient. It is a measure of the linear correlation between two variables X and Y, giving a value between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation. It is widely used in the sciences as a measure of the degree of linear dependence between two variables.

Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations.

Pearson's correlation coefficient when applied to a population is commonly represented by the Greek letter ρ (rho). The formula for ρ is:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{4.13}$$

where:

- cov is the covariance;

- σ_X is the standard deviation of X.

The formula for ρ can be expressed in terms of mean and expectation. Since

$$cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] \tag{4.14}$$

Then the formula for ρ can also be written as

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X sigma_Y} \tag{4.15}$$

where:

- cov and σ_X are defined as above;
- μ_X is the mean of X;
- E is the expectation.

The correlation coefficient ranges from -1 to 1. A value of 1 implies that a linear equation describes the relationship between X and Y perfectly, with all data points lying on a line for which Y increases as X increases. A value of -1 implies that all data points lie on a line for which Y decreases as X increases. A value of 0 implies that there is no linear correlation between the variables.

More generally, the correlation coefficient is positive if X_i and Y_i tend to be simultaneously greater than, or simultaneously less than, their respective means. The correlation coefficient is negative if X_i and Y_i tend to lie on opposite sides of their respective means. Moreover, the stronger is either tendency, the larger is the absolute value of the correlation coefficient.

4.5 Incremental Learning of Gaussian Mixture Model

EM algorithm has been widely used for learning Gaussian Mixture Models (GMM) in applications involving static data where GMM learning is a one-time process only and incremental learning is not much of a desired feature. In many practical applications, learning GMM from dynamic streams of data requires incremental learning for practicality and efficiently. It is in such scenarios where traditional approaches towards EM clustering have significant shortcomings. The requirement that number of clusters be specified has serious problems associated with it in dynamic data scenarios. In those scenarios there is no way by which one could identify number of clusters beforehand. Furthermore, accuracy of EM algorithm is greatly dependent on the way it is initialized. Most of the traditional approaches for initialization can not insure optimal accuracy within specific time constraints.

This is the so called unsupervised incremental learning, which considers building a model, seen as a set of concepts of the environment describing a data flow, where each data point is just instantaneously available to the learning system. In this case, the learning system needs to take into account these instantaneous data to update its model of the environment. An important issue in unsupervised incremental learning is the stability-plasticity dilemma, i.e., whether a new presented data point must be assimilated in the current model or cause a structural change in the model to accommodate the new information that it bears, i.e., a new concept.

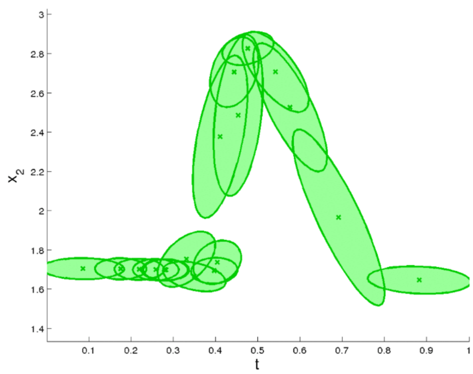

Figure 4.4: Example of the IGMM. The gaussians are not estimated ex-novo at every repetition of the algorithm, but they are adapted to new data.

The EM algorithm, for instance, follows the mixture distribution approach for probabilistic modeling. This algorithm proceeds in two steps: an estimation step (E) that computes the probabilistic membership (the posterior probability) of every data to each component of the mixture model based on a current hypothesis (a set of parameters), followed by a maximization step (M) that updates the parameters of the current hypothesis based on the maximization of the likelihood of the data. Here the number of concepts is fixed and must be known at the start of the learning process. The parameters of each distribution are computed through the usual statistical point estimators, a batch-mode approach which considers that the complete training set is previously known and fixed. As the EM algorithm, IGMM also follows the mixture distribution modeling [12] [20] [2] [40]. However, its model can be effectively expanded with new components (i.e. concepts) as new relevant information is identified in the data flow. Moreover, IGMM adjusts the parameters of each distribution after the presentation of every single data point according to recursive equations that are approximate incremental counterparts of the batch-mode update equations used by the EM algorithm.

Two main issues arise in the attempt to solve the problem of unsupervised incremental learning: how to tackle the stability-plasticity dilemma and how to update the values of distribution parameters as new data points are sequentially acquired. IGMM handles the stability-plasticity dilemma by means of a novelty criterion based on the likelihood of the mixture components. The IGMM algorithm converges after the presentation of few training samples and does not require a predefined number of Gaussian distributions.

4.5.1 Incremental Gaussian Mixture Model

IGMM assumes that the probability density of the input data $p(x)$ can be modeled by a linear combination of component densities $p(x|j)$ corresponding to independent probabilistic processes, in the form

$$p(x) \sum_{j=1}^{M} p(x|j)p(j) \tag{4.16}$$

This representation is called a mixture model and the coefficients $p(j)$ are called the mixing parameters, related to the prior probability of x having been generated from component j of the mixture. The priors are adjusted to satisfy the constraints

$$\sum_{j=1}^{M} p(j) = 10 \leq p(j) \leq 1 \tag{4.17}$$

Similarly, the component density functions p(x|j) are normalized so that

$$\int p(x|j)dx = 1 \tag{4.18}$$

The probability of observing vector $x = (x_1, ..., x_i, ..., x_D)$ belonging to the jth mixture component, is computed by a multivariate normal Gaussian, with mean μ_j and covariance matrix C_j:

$$p(x|j) = \frac{1}{(2\pi)^(D/2)\sqrt{|C_j|}} exp\{-\frac{1}{2}(x-\mu_j)^T C_j^{-1}(x-\mu_j)\} \tag{4.19}$$

IGMM adopts an incremental mixture distribution model, having special means to control the number of mixture components that effectively represent the so far presented data. The goal is modeling environments whose overall dynamics can be described by a set of persistent concepts which will be incrementally learned and represented by a set of mixture components. So, it has been introduced a novelty criterion to overcome the problem of the model complexity selection, related to the decision whether a new component should be added to the current model. The mixture model starts with a single component with unity prior, centered at the first input data, with a baseline covariance matrix specified by default. New components are added by demand. IGMM uses a minimum likelihood criterion to recognize a vector x as belonging to a mixture component. For each incoming data point the algorithm verifies whether it minimally fits any mixture component. A data point x is not recognized as belonging to a mixture component j if its probability $p(x|j)$ is lower than a previously specified novelty threshold. In this case, $p(x|j)$ is interpreted as a likelihood function of the jth mixture component. If x is rejected by all density components, it means that it bears new information, a new component is added to the model, appropriately adjusting its parameters.

A new mixture component is created when the instantaneous data point $x = (x_1, ..., x_i, ..., x_D0)$ matches the novelty criterion written as

$$p(x|j) < \frac{\tau_{nov}}{(2\pi)^(D/2)\sqrt{|C_j|}} \forall j \tag{4.20}$$

An instantaneous data point that does not match the novelty criterion needs to be assimilated by the current mixture distribution, causing an update in the values of its parameters due to the information it bears. IGMM follows an incremental version for the usual iterative process to estimate the parameters of a mixture model based on two steps: an estimation step (E) and a maximization step (M). The update process begins computing the posterior probabilities of component membership for the data point, $p(j|x)$, the estimation step. These can be obtained through Bayes' theorem.

The posterior probabilities can then be used to compute new estimates for the values of the mean vector μ_i^{new} and covariance matrix C_j^{new} of each component density $p(x|j)$, and the priors $p^{new}(j)$ in the maximization step. Next, will be derived the recursive equations used by IGMM to successively estimate these parameters.

The parameters $\theta = (\theta_1, ..., \theta_M)^T$, corresponding to the means, μ_j, covariances matrices, C_j, and priors $p(j)$ of a mixture model involving D-dimensional Gaussian distributions $p(x|j)$, can be estimated from a data sequence of t vectors, $X = x_1, ..., x_n, ..., x_t$ assumed to be drawn independently from this mixture distribution. The estimates of the parameters are random vectors whose statistical proprieties are obtained from their joint density function. Starting from an initial "guess", each observation vector is used to update the estimates according to a successive estimation procedure, basing on the maximization of the likelihood of the data. The likelihood of θ for the given X, $L(\theta)$, is the joint probability density of the whole data stream X. The technique of maximum likelihood sets the value of θ by maximizing $L(\theta)$.

The mean vector and the covariance matrix are estimated following the natural conjugate technique. Moreover when the probability density of the input data is a Gaussian Mixture Model with M components, an observation x^t is probabilistic assigned to a distribution j by the corresponding posterior probability $p(j|x^t)$. In this case, the equivalent number of samples used to compute the parameter estimates of the jth distribution component corresponds to the sum of posterior probabilities that the data presented so far were generated from component j, the *0th-order data moment for j*. IGMM stores this summation as the variable sp_j which is periodically restarted to avoid an eventual saturation. The recursive equations used by IGMM to update the model distributions are:

$$sp_j = sp_j + p(j|x) \tag{4.21}$$

$$\mu_j = mu_j + \frac{p(j|x)}{sp_j}(x - mu_j) \tag{4.22}$$

$$C_j = C_j - (\mu_j - \mu_j^{old})(\mu_j - \mu_j^{old})^T + \frac{p(j|x)}{sp_j}(x - mu_j)[(x - mu_j)(x - mu_j)^T - C_j] \tag{4.23}$$

$$p(j) = \frac{sp_j}{\sum_{q=1}^{M} sp_q} \tag{4.24}$$

where $p(j|x)$ μ_j^{old} refers to the value of μ_j at time $t-1$ (i.e., before updating). These update equations continuously compute an instantaneous approximation of the parameters that represent the mixture distribution.

The IGMM algorithm has just two configuration parameters, σ_{ini} and τ_{nov}. The σ_{ini} parameter is not critical, the τ_{nov} parameter, on the other hand, is more critical and must be defined carefully. It indicates how distant x must be from μ_j to be consider a non-member of j. For instance, $\tau_{nov} = 0.01$ indicates that $p(x|j)$ must be lower than one percent of the probability in the center of the Gaussian for x be considered a non-member of j. If

$\tau_{nov} < 0.01$, few pattern units will be created and the regression will be coarse. If $\tau_{nov} > 0.01$, more pattern units will be created and consequently the regression will be more precise. In the limit, if $\tau_{nov} = 1$ one unit per training pattern will be created.

Chapter 5

Interfacing with the real robot

The framework described has been tested in a real situation controlling two different robots using a Robot Operating System (ROS) [51] based system, namely al Aldebaran NAO [54] and a Comau Smart5 SiX [53]. The software is able to simulate generation of EMG, to compute signal analysis and to tell robot the position to set. For each step there is a dedicated ROS node which performs its task through reception and dispatch of ROS messages. Figure 4.1 shows blocks which form the framework, starting from EMG signals, passing through features extraction, until robot motion. The chart shows how it is possible to modify and separate each step according to needs; for example if we want to change the humanoid robot, or if we want to use a different kind of model. When robots are used to replicate human movements it should be carefully checked whether motors and joints allow to perform motion the same way. In the case of knee joint there are not particular problems since NAO limits almost match ones of a human being, but could arise some if for example are considered also hip and ankle joint. Comau Smart5 SiX has been used for testing the third dataset, i. e. the one which involved wrist motion. The limits of the Smart5 SiX are less strict than the limits of a human wrist, so there is no need of controls to check if the motion overcome the human constraints.

5.1 NAO Robot

NAO is a 58-cm tall humanoid robot producted by Aldebaran, intended to be a friendly companion around the house, the great interactivity of the robot makes him really endearing and loveable.

While waiting to be ready for home use, NAO became a star in the world of education. In more than 70 countries, he was used in computer and science classes, from primary school through to university. Thanks to NAO, students can learn programming in a fun and practical way. They can program him to walk, catch small objects and even dance! He then conquered communities of developers, who recognised him as a powerful and incredibly expressive medium for creating applications.

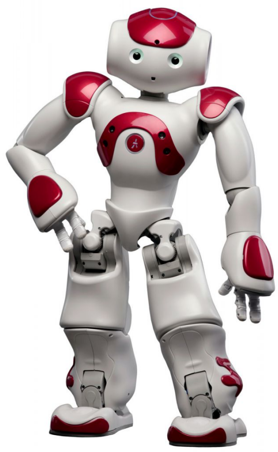

Figure 5.1: Aldebaran Nao

NAO has unique combination of hardware and software: he consists of sensors, motors and software driven by NAOqi, its dedicated operating system. Movement libraries are available through graphics tools such as Choregraphe and other advanced programming software. They allow users to create elaborate behaviors, access the data acquired by the sensors and control the robot.

NAO has:

- Body with 25 degrees of freedom (DOF) whose key elements are electric motors and actuators;
- Sensor network: two cameras, four directional microphones, sonar rangefinder, two IR emitters and receivers, one inertial board, nine tactile sensors and eight pressure sensors;
- Various communication devices, including voice synthesizer, LED lights and 2 high-fidelity speakers
- Intel ATOM 1,6ghz CPU (located in the head) that runs a Linux kernel and supports Aldebaran's proprietary middleware (NAOqi);

- Second CPU (located in the torso);
- 48.6-watt-hour battery that provides NAO with 1.5 or more hours of autonomy, depending on usage.

This combination of technologies gives NAO the ability to detect its surroundings. Now it must interpret what it detected. This is where the embedded software in NAO's head comes in. Aldebaran created a dedicated operating system, NAOqi, allowing the small humanoid to understand and interpret the data received by its sensors.

5.1.1 Omnidirectional walking

NAO's walking uses a simple dynamic model (linear inverse pendulum) and quadratic programming. It is stabilized using feedback from joint sensors. This makes walking robust and resistant to small disturbances and torso oscillations in the frontal and lateral planes are absorbed. NAO can walk on a variety of floor surfaces, such as carpeted, tiled and wooden floors. NAO can transition between these surfaces while walking.

5.1.2 Whole body motion

NAO's motion module is based on generalized inverse kinematics, which handles Cartesian coordinates, joint control, balance, redundancy and task priority. This means that when asking NAO to extend its arm, it bends over because its arms and leg joints are taken into account. NAO will stop its movement to maintain balance.

5.1.3 Fall Manager

The Fall Manager protects NAO when it falls. Its main function is to detect when NAO's center of mass (CoM) shifts outside the support polygon. The support polygon is determined by the position of the foot or feet in contact with the ground. When a fall is detected, all motion tasks are killed and, depending on the direction, NAO's arms assume protective positioning, the CoM is lowered and robot stiffness is reduced to zero.

5.2 Smart SiX

Smart5 SiX robots, the smallest within the Comau range, boast an exceptionally compact design particularly suitable for all operations that require

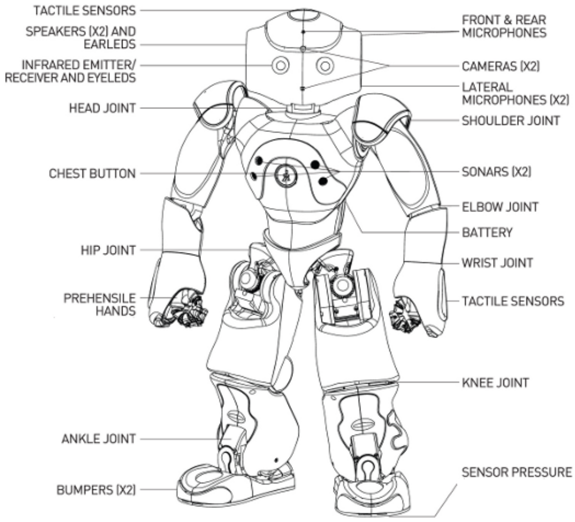

Figure 5.2: Aldebaran Nao

fast movement and a high degree of repeatability. Smart5 SiX range features a model exclusively dedicated to arc welding: the Smart5 SiX ARC, designed to optimise movement specifically for arc welding. This model features an integrated fitting solution which involves the passage of the cables through the robot base.

Figure 5.3: Comau Smart5 SiX

The main robot characteristics are listed below:

- Pre-engineered for use with a variety of optional devices;

- Oil lubrication for all the reducers, with the exception of axes 5 and 6 which are lubricated with grease;
- Possibility of connecting electrical and pneumatic services to the forearm;
- Reduced wrist dimensions enable high capacity orientation in small spaces;
- High repeatability;
- Robot protection level is IP65;
- no specific devices for axis compensation.

The handling of the axes is controlled by brushless motors with direct transmission of the motion to axes 1-2-3-4, by means of mechanical geared reduction units, whereas for axes 5-6 the transmission is by belt to a Harmonic Drive type reduction unit. The main robot fittings include:

- A specific welding dressing;
- An internal pneumatic line with upper connection on the back of the forearm;
- Wiring that comprises a service line with a connector on the upper plate next to the pneumatic connection;
- Flat surfaces and threaded holes on the upper part of the forearm that can be used to assemble fixtures (servovalves, transformer, etc.).

A specific outfitting is available with the Smart SiX robot for arc welding, including the welding wire coil, wire-puller, torch and equipment on the robot

The robot consists of an anthropomorphic structure with 6 degrees of freedom. It is anchored to the floor by means of a steel plate and bolts. The robot has a fixed base on which the column with the axis 2 gear reducer rotates around the vertical axis (axis 1). An arm connects axis 2 to the forearm and includes the reducers of axes 3-4-5-6; the wrist is located at the end of the forearm. The robot axes are equipped with software limit stop (programmable) and /or mechanical shock absorber stops supplied as standard or on request. The strokes of the main axes (axes 1-2-3) can be limited by means of additional mechanical shock absorber stops, according to specific application requirements

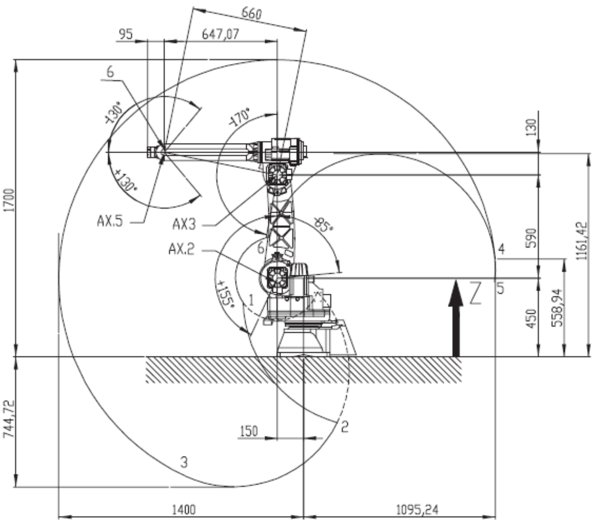

Figure 5.4: Comau Smart5 SiX operating area

5.3 ROS

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. Creating truly robust, general-purpose robot software is hard. From the robot's perspective, problems that seem trivial to humans often vary wildly between instances of tasks and environments. Dealing with these variations is so hard that no single individual, laboratory, or institution can hope to do it on their own.

As a result, ROS was built from the ground up to encourage collaborative robotics software development. For example, one laboratory might have experts in mapping indoor environments and could contribute a world-class system for producing maps. Another group might have experts at using maps to navigate and yet another group might have discovered a computer vision approach that works well for recognizing small objects in clutter. ROS was designed specifically for groups like these to collaborate and build upon each other's work.

ROS is a large project with many ancestors and contributors. The need for an open-ended collaboration framework was felt by many people in the robotics research community and many projects have been created towards this goal. Various efforts at Stanford University in the mid-2000s involving integrative, embodied AI, such as the STanford AI Robot (STAIR) and the Personal Robots (PR) program, created in-house prototypes of flexible, dy-

Figure 5.5: ROS logo

namic software systems intended for robotics use. In 2007, Willow Garage, a nearby visionary robotics incubator, provided significant resources to extend these concepts much further and create well-tested implementations. The effort was boosted by countless researchers who contributed their time and expertise to both the core ROS ideas and to its fundamental software packages. Throughout, the software was developed in the open using the permissive BSD open-source license and gradually has become a widely-used platform in the robotics research community.

From the start, ROS was developed at multiple institutions and for multiple robots, including many institutions who received PR2 robots from Willow Garage. Although it would have been far simpler for all contributors to place their code on the same servers, over the years, the "federated" model has emerged as one of the great strengths of the ROS ecosystem. Any group can start their own ROS code repository on their own servers and they maintain full ownership and control of it. They don't need anyone's permission. If they choose to make their repository publicly available, they can receive the recognition and credit they deserve for their achievements and benefit from specific technical feedback and improvements like all open source software projects.

The ROS ecosystem now consists of tens of thousands of users worldwide, working in domains ranging from tabletop hobby projects to large industrial automation systems.

ROS was designed to be as distributed and modular as possible, so that users can use as much or as little of ROS as they desire, the modularity of ROS allows the user to pick and choose which parts are useful and which parts have to be implemented.

The distributed nature of ROS also fosters a large community of user-contributed packages that add a lot of value on top of the core ROS system. At last count there were over 3,000 packages in the ROS ecosystem and that is only the ROS packages that people have taken the time to announce to the public. These packages range in fidelity, covering everything from proof-of-concept implementations of new algorithms to industrial-quality drivers and capabilities. The ROS user community builds on top of a common infrastructure to provide an integration point that offers access to hardware drivers, generic robot capabilities, development tools, useful external libraries and more.

Over the past several years, ROS has grown to include a large community of users worldwide. Historically, the majority of the users were in research labs, but increasingly we are seeing adoption in the commercial sector, particularly in industrial and service robotics.

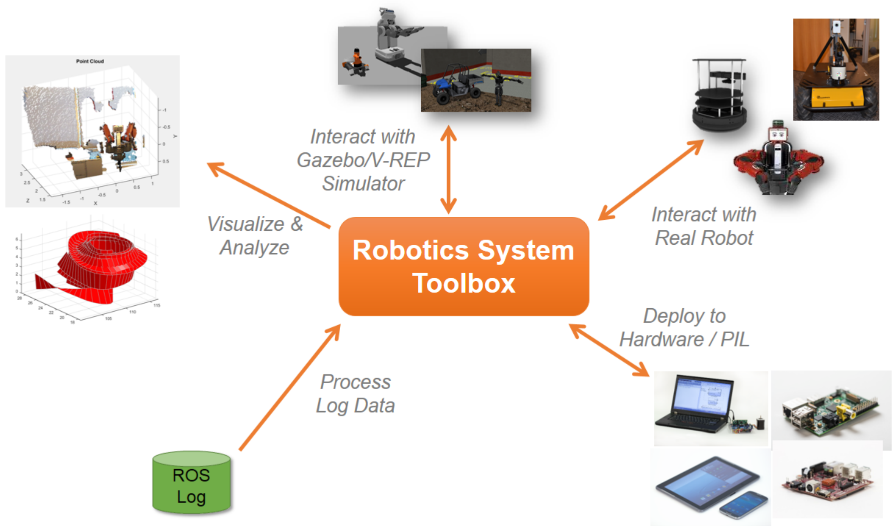

Figure 5.6: What ROS can do

5.3.1 Communications Infrastructure

At the lowest level, ROS offers a message passing interface that provides inter-process communication and is commonly referred to as a middleware.

The ROS middleware provides these facilities:

- Publish/subscribe anonymous message passing;
- Recording and playback of messages
- Request/response remote procedure calls
- Distributed parameter system

5.3.1.1 Message Passing

A communication system is often one of the first needs to arise when implementing a new robot application. ROS's built-in and well-tested messaging system saves time by managing the details of communication between distributed nodes via the anonymous publish/subscribe mechanism as shown in 5.7. Another benefit of using a message passing system is that it forces the user to implement clear interfaces between the nodes in the system, thereby improving encapsulation and promoting code reuse. The structure of these message interfaces is defined in the message IDL (Interface Description Language).

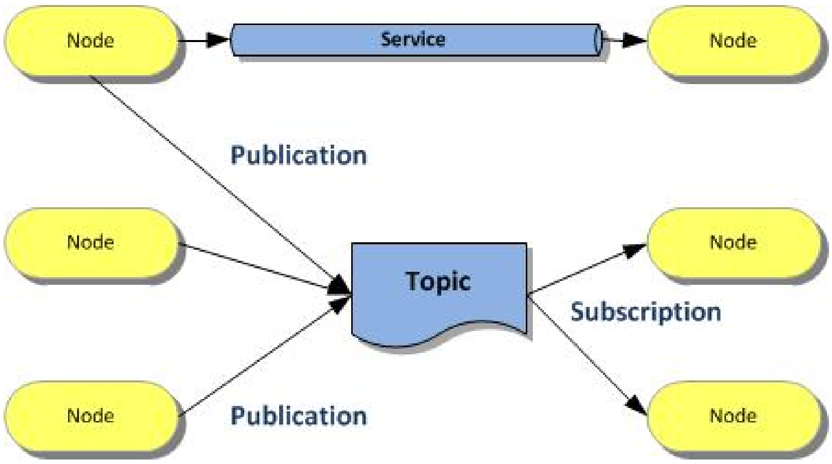

Figure 5.7: Message passing in ROS

5.3.1.2 Recording and Playback of Messages

Because the publish/subscribe system is anonymous and asynchronous, the data can be easily captured and replayed without any changes to code. Say you have Task A that reads data from a sensor and you are developing Task B that processes the data produced by Task A. ROS makes it easy to capture the data published by Task A to a file and then republish that data from the file at a later time. The message-passing abstraction allows Task B to be agnostic with respect to the source of the data, which could be Task A or the log file. This is a powerful design pattern that can significantly reduce the development effort and promote flexibility and modularity in the system.

5.3.1.3 Remote Procedure Calls

The asynchronous nature of publish/subscribe messaging works for many communication needs in robotics, but sometimes is needed a synchronous request/response interactions between processes. The ROS middleware provides this capability using services. Like topics, the data being sent between processes in a service call are defined with the same simple message IDL.

The ROS middleware also provides a way for tasks to share configuration information through a global key-value store (Distributed Parameter System). This system allows the user to easily modify the task settings and even allows tasks to change the configuration of other tasks.

5.3.2 Robot-Specific Features

In addition to the core middleware components, ROS provides common robot-specific libraries and tools that will get the robot up and running quickly. Here are just a few of the robot-specific capabilities that ROS provides:

- Standard Message Definitions for Robots;
- Robot Geometry Library;
- Robot Description Language;
- Preemptable Remote Procedure Calls;
- Diagnostics;
- Pose Estimation;
- Localization;
- Mapping;
- Navigation.

5.3.2.1 Standard Robot Messages

Years of community discussion and development have led to a set of standard message formats that cover most of the common use cases in robotics. There are message definitions for geometric concepts like poses, transforms and vectors; for sensors like cameras, IMUs and lasers; and for navigation data like odometry, paths and maps; among many others. By using these standard messages in the application, the code will interoperate seamlessly with the rest of the ROS ecosystem, from development tools to libraries of capabilities.

5.3.2.2 Robot Geometry Library

A common challenge in many robotics projects is keeping track of where different parts of the robot are with respect to each other. For example, if you want to combine data from a camera with data from a laser, you need to know where each sensor is, in some common frame of reference. This issue is especially important for humanoid robots with many moving parts. This problem has been addressed in ROS with the tf (transform) library, which will keep track of where everything is in the robot system.

Designed with efficiency in mind, the tf library has been used to manage coordinate transform data for robots with more than one hundred degrees of freedom and update rates of hundreds of Hertz. The tf library allows the user to define both static transforms, such as a camera that is fixed to a mobile base and dynamic transforms, such as a joint in a robot arm. It is possible to transform sensor data between any pair of coordinate frames in the system. The tf library handles the fact that the producers and consumers of this information may be distributed across the network and the fact that the information is updated at varying rates.

5.3.2.3 Robot Description Language

Another common robotics problem that ROS solves is how to describe the robot in a machine-readable way. ROS provides a set of tools for describing and modeling the robot so that it can be understood by the rest of the ROS system, including tf, robot state publisher and rviz. The format for describing the robot in ROS is URDF (Unified Robot Description Format), which consists of an XML document in which the user describes the physical properties of the robot, from the lengths of limbs and sizes of wheels to the locations of sensors and the visual appearance of each part of the robot.

Once defined in this way, the robot can be easily used with the tf library, rendered in three dimensions for nice visualizations and used with simulators and motion planners.

5.3.2.4 Preemptable Remote Procedure Calls

While topics (anonymous publish/subscribe) and services (remote procedure calls) cover most of the communication use cases in robotics, sometimes the user need to initiate a goal-seeking behavior, monitor its progress, be able to preempt it along the way and receive notification when it is complete. ROS provides actions for this purpose. Actions are like services except they can report progress before returning the final response and they can be preempted by the caller. So, for example, you can instruct your robot to navigate to some location, monitor its progress as it attempts to get there, stop or redirect it along the way and be told when it has succeeded (or failed). An action is a powerful concept that is used throughout the ROS ecosystem.

5.3.2.5 Diagnostics

ROS provides a standard way to produce, collect and aggregate diagnostics about the robot so that, at a glance, the user can quickly see the state of the robot and determine how to address issues as they arise.

5.3.2.6 Pose Estimation, Localization and Navigation

ROS also provides some "batteries included" capabilities that help the users get started on their robotics project. There are ROS packages that solve basic robotics problems like pose estimation, localization in a map, building a map and even mobile navigation.

5.3.3 Tools

One of the strongest features of ROS is the powerful development toolset. These tools support introspecting, debugging, plotting and visualizing the state of the system being developed. The underlying publish/subscribe mechanism allows the user to spontaneously introspect the data flowing through the system, making it easy to comprehend and debug issues as they occur. The ROS tools take advantage of this introspection capability through an extensive collection of graphical and command line utilities that simplify development and debugging.

5.3.3.1 rviz

Perhaps the most well-known tool in ROS, rviz provides general purpose, three-dimensional visualization of many sensor data types and any URDF-described robot.

rviz can visualize many of the common message types provided in ROS, such as laser scans, three-dimensional point clouds and camera images. It also uses information from the tf library to show all of the sensor data in a common coordinate frame, together with a three-dimensional rendering of the robot. Visualizing all of the data in the same application not only looks impressive, but also allows the user to quickly see what the robot sees and identify problems such as sensor misalignments or robot model inaccuracies.

5.3.3.2 rqt

ROS provides rqt, a Qt-based framework for developing graphical interfaces for the robot. Anyone can create custom interfaces by composing and configuring the extensive library of built-in rqt plugins into tabbed, split-screen and other layouts. The users can also introduce new interface components by writing their own rqt plugins.

The *rqt_graph* plugin provides introspection and visualization of a live ROS system, showing nodes and the connections between them and allowing the user to easily debug and understand their running system and how it is structured.

With the rqt *plot_plugin*, it is possible to monitor encoders, voltages, or anything that can be represented as a number that varies over time.

For monitoring and using topics, there are the *rqt_topic* and *rqt_publisher* plugins. The former lets the user monitor and introspect any number of topics being published within the system. The latter allows to publish messages to any topic, facilitating ad hoc experimentation with the system.

For data logging and playback, ROS uses the bag format. Bag files can be created and accessed graphically via the *rqt_bag* plugin. This plugin can record data to bags, play back selected topics from a bag and visualize the contents of a bag, including display of images and plotting of numerical values over time.

5.4 Robot Simulation

In order to test the goodness of the model on a real robot, the first step is connecting to it. For the NAO robot communication is done through TCP/IP, so the IP address of the robot must be set after switching on the robot.

After connecting the computer to the robot all the sensors will activate and all the joints will became stiff. Then NAO will be placed in a certain stable, initial position (in this case a sitting position).

At this point we could start to communicate with the robot.

In this simulation we will not have a real person to which are connected sEMG sensors in order to collect electromyographic data. This would be a very interesting simulation, but building up the whole system would be too complicated, since in the IAS Lab there is not the necessary equipment. Furthermore this is not the main goal of the thesis. To simulate the online collection of the data, a ROS node will simulate generation of signals as if they were instantly acquired. The node, named *emg_generator*, reads from a configuration file the path where the file containing signals is stored, it loads it and publishes a message with EMG values at a frequency rate equal to the original acquisition one. The message will contain the EMG signals from the previously selected channels. When there are no more values to be published it sends a message with fake values to tell listening node that the signal is finished.

There is an other node which listen to EMG messages and connect to robot. It is named *model2motion_wlt*. It listens to EMG messages, compute wavelet transform and then publish to the robot the bending angle of the selected joint.

model2motion_wlt also handles the load of the GMM model. Every time a new sEMG value is received by the callback, the new portion of signal is elaborated through the function *computeRegression* which returns the corresponding angle. Due to normal imperfections of estimation, values resulting from regression must be managed in order to eliminate noise and oscillations. The position sent to the robot is the mean of the last twenty angles computed.

Humanoid robots are built to resemble human beings but mechanical components like motors and materials cause some limits to behavior replication. Human and NAO capabilities are different depending on the joint considered. For example the motion of the knee joint is almost identical and the only difference is about the minimum and maximum reachable angles. Instead, for the motion of the Comau Smart5 SiX, the limits in the wrist movement exceeds the limits of the human wrist, so there is no need of verifying them. However, is possible to store the limits of the robot joints in a YAML Ain't Markup Language (YAML) file and, during the execution, check that they are not exceeded.

Parameters for both nodes *model2motion* and *emg_generator* can be set through a YAML file.

Once NAO is ready to perform actions, the node *model2motion_wlt* can be run. When robot is in position, the simulation of the EMG signals can be started.

Chapter 6

Experimental Results

The three dataset described in the previous chapters have been analyzed through different aspects, depending on their characteristics.

6.1 Dataset 1

The first dataset, as previously said, is composed by many repetitions of the same movement (about 60) performed by three subjects. This configuration is ideal to study how the model adapts to a new subject and after how many repetitions the model fits to a new subject.

The sEMG signals were collected from eight different channels and for every movement were considered 2000 samples. Building a model takes time and, of course, more trials and channels are used, more time is needed to build the model. These two variables have to been examined in order to detect the best trade off between the goodness of the model and the time needed to build it.

The main goal of this part of the project was to detect the best combination of the parameters to build the most robust model. First the dataset has been used to build a model offline, with the classic Gaussian Mixture Model. Then has been tried the incremental GMM to see if the results were comparable with the classic version. Finally has been checked the goodness of the online GMM with the feature extraction performed by wavelet transform.

In all the cases the results were satisfying, though the better results were obtained with the offline classic version of GMM.

For sake of simplicity, the three subjects studied in this dataset are referred to as S1 for the first subject (a man), S2 for the second subject (a man) and S3 for the third subject (a woman).

The sEMG signals were recorded with sensors placed in correspondence of eight muscles:

Muscle	Short Name
Rectus femoris	Ch1
Vastus lateralis	Ch2
Vastus medialis	Ch3
Tibialis anterior	Ch4
Gastrocnemius lateralis	Ch5
Gastrocnemius medialis	Ch6
Biceps femoris caput longus	Ch7
Peroneus longus	Ch8

Table 6.1: Abbreviation of the muscles whence are collected sEMG signals

6.1.1 Offline analysis [66]

GMM have been trained with data from couple of subjects (S1+S2, S1+S3, S2+S3). For every couple, different sizes of training set have been considered (10, 30, 60, 120), half from the first subject and half from the second one. For the testing phase, has been used 10 trials coming from the remaining subject in order to verify the generality of the model. The described procedure has been applied to all the collected EMG channels, so as to detect which channels bring the most information.

Figure 6.1 represents the GoF computed for the couple S1+S2 and tested on S3 varying the EMG channel and the number of data used for training, while Figure 6.2 shows the same thing for the subjects S1+S3 tested on S2. Figure 6.3 shows a comparison of the results of GoF of all the possible combination of subjects with the model built with 60 trials used as train.

Results showed a good estimation even with few input data for almost all the channels. Increasing the cardinality of the training set we generally obtained similar or better performance with some exceptions. In fact, it was worth to use a number of trials varying between 30 and 60 in order to obtain more stable results during the testing phase, while using a greater number of trials raised significantly the time for model training without any apparent benefit in efficiency.

As it is possible to see, the results are all quite good. For the model build on S1 and S2 the poor results for Ch2, Ch5 and Ch6 are due to acquisition problem for those particular channels.

Looking at the whole set of results, they were generally quite similar, except for specific channels in some rare cases. As regards the best EMG channels, generally Ch1, Ch2, Ch3 and Ch7 were the muscles bringing most

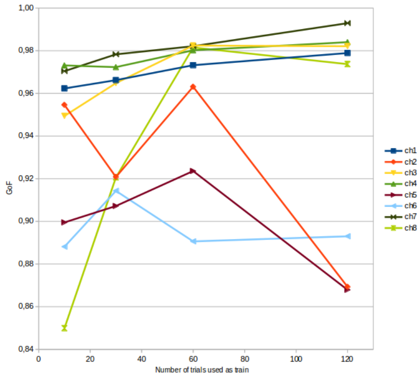

Figure 6.1: GoF values from every channel related to the number of trials used as train for subjects S1+S2

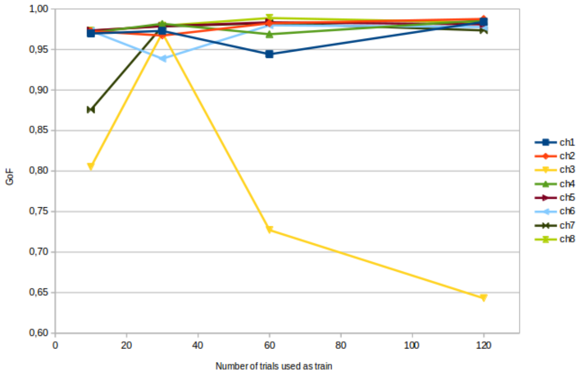

Figure 6.2: GoF values from every channel related to the number of trials used as train for subjects S1+S3

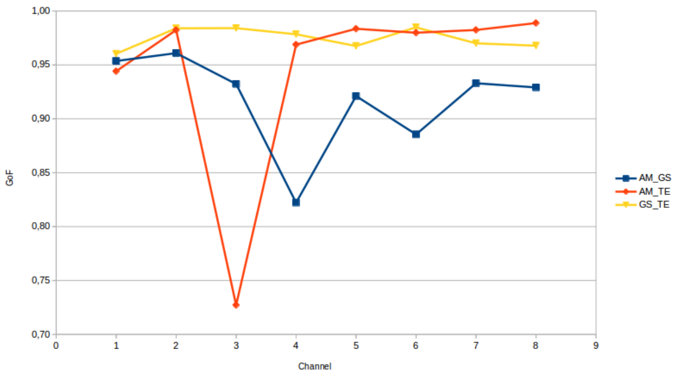

Figure 6.3: Comparison of the GoF values from every channel for all the subjects with 60 trials used as train

information. In fact, the GoF for the EMG channels related to these muscles resulted between the most informative even in subject-specific models for all the three subjects. Moreover, the cited muscles are the principal actors of the considered movement from a biological point of view.

Coherently, tests showed that the best trade off in terms of both stability and efficiency has been obtained using three out four of the already cited EMG channels (GoF = 0.9257 for Ch1+Ch2+Ch3 and GoF = 0.9409 for Ch1+Ch2+Ch7). Using more than three channels is unnecessary because it leads to an overfitting of the system. On the other hand using less than three channels gives results not enough safe, in fact a little problem on acquisition of just one channel could compromise the final result, as shown for the model build on S1 and S2 and tested on S3.

6.1.1.1 Leave One Out

The model has then been studied in a leave-one-out way (Leave One Out (LOO)), in which it has been built on $n-1$ subjects and tested on the nth subject. First the model is built with 60 observations, half from the first subject and half from the second one. The model is then tested on 10 trials of the third, unknown, subject. Then, these observations are added to the train data and a new model is built with 70 trials, 30 from the first subject, 30 from the second subject and 10 from the third subject. The new model is then tested on 10 new trials of the third subject and so on. In this way is possible to simulate the fitting of the model on a new subject and and study after how many repetitions the model is able to adapt to the data of a new person.

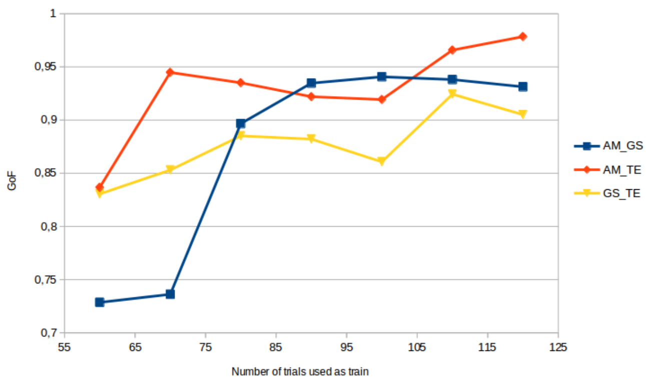

Figure 6.4: LOO test of the model while adapting to a new subject

As it is possible to see on Figure 6.4 after about 20 repetitions the model is able to fit quite well to a new subject.

6.1.1.2 Comparison between subject-independent and multi-subject model

6.2 compare the results between subject-specific and subject independent models regarding the most informative EMG channels. The subject-independent model for the subject Sn has been trained on all subjects except Sn and tested on Sn (n = 1, 2, 3), using a leave-one-out approach.

Sometimes the subject-independent model shows even better results than the single subject one. In other cases the results were poorer in the subject-independent model, this happens when the same channel gives different performances for the two single subjects.

Subject	S1	S2	S3
Specific	0.9238	0.9700	0.9570
Independent	0.8887	0.9214	0.8733

Table 6.2: GoF values comparing results from subject-specific and subject-independent models.

6.1.1.3 IGMM

The same study described on the previous section has been repeated building the Gaussian Mixture Model in a incremental way. In this case the results were a few poorer but the time needed to build it is reduced considerably.

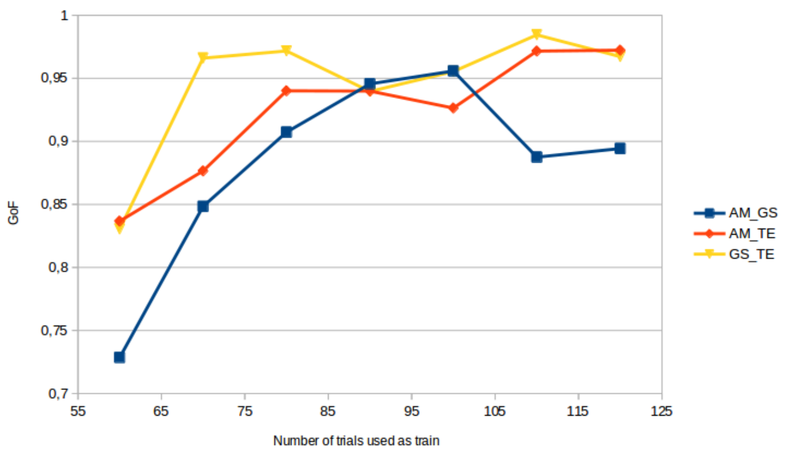

Figure 6.5: LOO test of the IGMM while adapting to a new subject

6.1.2 Online analysis [65]

In this dataset, the high number of repetitions of the same movement (about 60) performed by three subjects gave us the possibility to study the adaptation performances of the model adapts to a novel person. The 3 most significant channels have been selected, according to the results coming from the preliminary study. The selected channels (1, 2 and 8) recorded the activity of the muscles *Rectus femoris*, *Vastus lateralis* and *Peroneus longus*.

The db2 mother wavelet and MAV synthesis feature have been applied to the raw signal provided from every single channel. The resulting values have been associated to the corresponding bending angle along time. A model for each channel has been trained and GMR have been used to retrieve the estimated bending angle to be compared with a testing set. The 3 channels offering the best performances have been selected in order to obtain similar models. The number of channels has been defined by looking at minimum set of channels resulting with a significant correlation coefficient for the specific dataset.

Again, a leave-one-out approach has been adopted by building the model on 30 trials coming from 2 subjects and tested on the remaining one. We obtained 3 models on the movement trained by using a total of 60 repetitions. For analyzing the model adaptation, the data of every subject has been divided in blocks of 10 repetitions. In fact, 10 repetitions bring a good amount of information to the system, good enough to add a substantial contribute to the previous model. In the first part of the analysis, the different blocks of movements of a certain subject are used to test the model built on the other two subjects. This is represented in the 6.6 by the red lines. The blue line, instead, shows a model tested, respectively, on the same data of the previous case, but using an updated model with the previous testing data added to the training set. In this way the model is updated with the data from the third subject. As the new data are added to the model, the results improved, giving generally better results with respect to the first part of the analysis.

When the testing data showed more variability, the model generally decrease its performances, but in a long term perspective the adaptation characteristics of the proposed framework could solve the problems demonstrated with some subject with the NinaPro dataset.

6.1.3 Final comparison

In the Figure 6.7 is possible to see a comparison between all the techniques described. As it is possible to see, the classic GMM offline approach gives

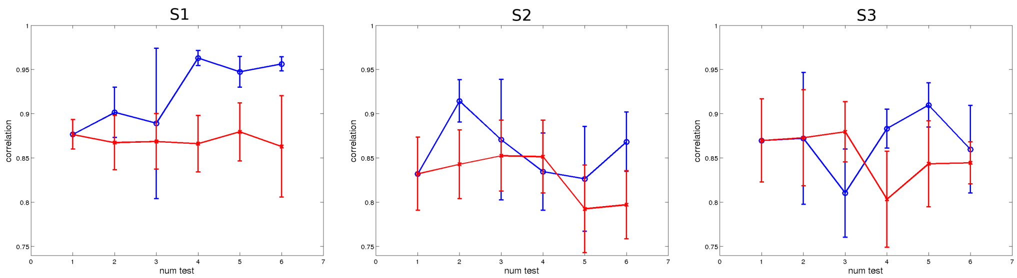

Figure 6.6: Correlation and Standard Deviation for the model of a kicking movement. The red line represent the results of the model built on two subjects and tested on different data from the third subject, without updating the model. The blue line represent the results of the model tested on the same data than the previous case, but updating the model with the data of the third person.

the best results, while the online approach gives poorer results. This is not surprising given that in the wavelet approach time is not considered and, of course, time brings a lot of information, which are useful in the building of the model.

The worsening of the results with 110 trials used as train is due to poor data added to the model. The classic version on the GMM is more resistant to rapid deterioration and adapts less quickly to new data, while the wavelet approach is more sensitive to changes in the train data.

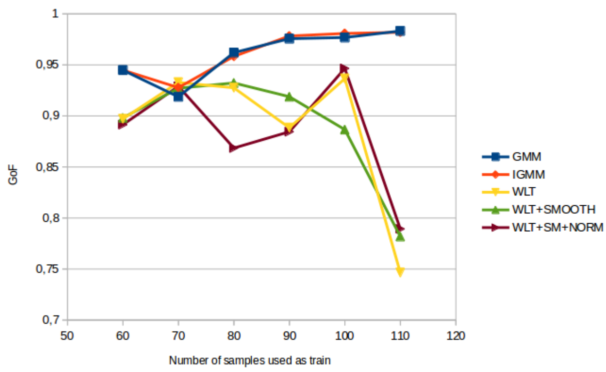

Figure 6.7: Comparison of the various techniques on the model built on S1 and S2 with LOO approach

6.2 Dataset 2

The dataset contains data of three movements, but only two has been analyzed. This dataset has been chosen because its movements involved the same joints of the first dataset. It also involved more subjects than the previous dataset. The dataset has been tested in a leave-one-out way, in which the first 40 trials are used for training the model and the last three ones are used for testing the model. Unfortunately this dataset is not as good as the other one, in fact the movements were performed with different velocities and this led to very poor performances as is shown in Figure 6.9.

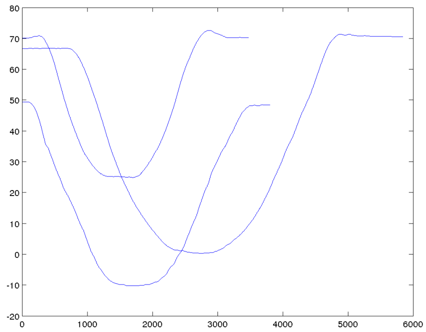

Figure 6.8: Different speeds for the same movement

To improve the results it is possible to use the DTW algorithm described in a previous chapter. After applying the algorithm the results change considerably as it is possible to see on Figure 6.9.

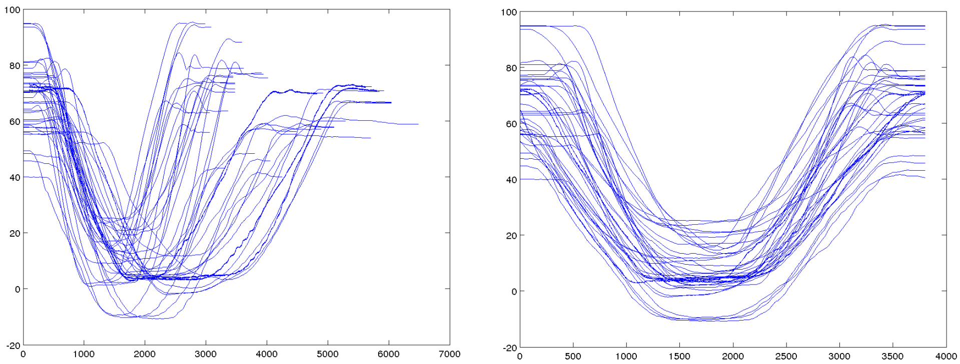

Figure 6.9: Knee angle before and after the application of the DTW algorithm

The model built on row data gave really poor performances, which increase impressively after the application of the DTW algorithm as it is possible to see on Figure 6.10.

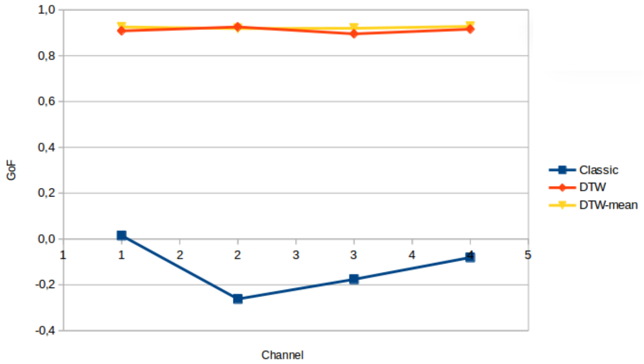

Figure 6.10: GoF of the model built on row data and after applying DTW algorithm

6.3 Dataset 3

In order to obtain comparable results between the considered dataset has been applied a series of standardizing approaches. A similar number of samples ($\simeq$ 2000) for trial has been considered by down-sampling of one tenth the information available in the NinaPro database, while the other two databases already had a similar number of samples. We look at the most informative EMG channels by conducting a preparatory study. The objective of the study was to select an equal number of muscles connected to the performed movement by means of a quantitative measure of the influence in the movement of each considered channel. As for the first dataset, to the signal of every channel has been applied the db2 mother wavelet and MAV synthesis feature, associating the resulting values to the corresponding bending angle along time.

The 3 channels offering the best performances have been selected in order to obtain similar models. The number of channels has been defined by looking at minimum set of channels resulting with a significant correlation coefficient for the specific dataset. It is worth to notice that more channels could be considered for this dataset, resulting in more accurate estimation. Anyway, a subset of the significant channels have been selected to simplify the comparison with the models produced with the first dataset, for which only 3 channels were actually correlated to the motion.

The high number of subjects involved in this dataset is ideal to analyze the robustness of the proposed framework. Two different movements of upper limbs, and in particular the wrist, i.e. flexion (Movement 13) and extension (Movement 14), have been selected between the different movement contained in the dataset. They are simple and involve a single joint movement. The purpose of this choice is to focus on subject-independent model more than on the complexity of the motion. The 3 most significant channels have been selected, according to the results coming from the preliminary study. The selected channels (3, 5 and 7) brought information from muscles around the forearm. A leave-one-out approach has been adopted by building the model on 39 subjects and tested on the remaining one. We obtained 40 models for each movement to be tested on the 6 repetitions of the testing subject. The mean and the standard deviation of the correlation coefficient computed from the comparison of 6 estimated bending trajectories with the actual measured angles have been plotted in Fig. 2 and Fig. 3.

The results showed a good correlation resulting from the created GMM/GMR framework. Both the movements reached a statistically significant mean correlation coefficient ($\rho_\alpha, \hat{\alpha} \geq 0.7$), with a lower result for Movement 14 (6.12) ($\rho_\alpha, \hat{\alpha} = 0.7117$) and quite consistent result for Movement 13 (6.11) ($\rho_\alpha, \hat{\alpha} = 0.8224$).

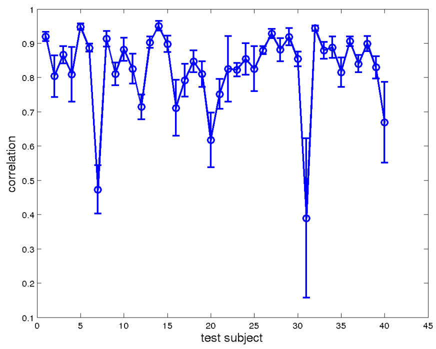

Figure 6.11: Correlation and Standard Deviation for the model of a wrist flexion movement. The model was built on $n - i$ subjects and tested on the ith

The model have showed low performances on some specific subjects, probably due to a different interpretation of the task with respect to the remainder of the users. This problem could probably be solved by considering an higher number of repetitions performed by the same individual, leading to a less mechanical movement.

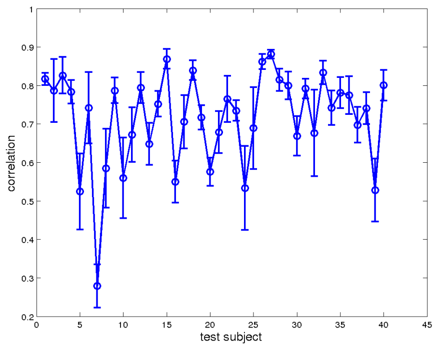

Figure 6.12: Correlation and Standard Deviation for the model of a wrist extension movement. The model was built on $n - i$ subjects and tested on the ith

Chapter 7

Conclusions

7.1 Discussion

This thesis proposed a method to estimate online a single joint angle for both upper and lower limbs. A Gaussian Mixture Model was built on Surface Electromyography signals from different dataset, while joint angles related to new, unseen Surface Electromyography data has been estimated by means of Gaussian Mixture Regression. The online version of the proposed framework used Wavelet Transform to estimate the bending angle starting only from a small portion of the whole signal. An offline version of the framework considering the entire evolution of the signal has been used as basis of our work.

The model was able to obtain significant results on new, unseen data, with great variability of subjects and few repetitions of the movements. Moreover, we obtained good results even with different movements of the same joint (Correlation Coefficient 0.8224 on average for wrist flexion and 0.7117 on average for wrist extension). Furthermore, we prove that is possible to improve the model adapting it to the data of a new subject, since this gave better results than without updating the model.

This means that it is possible to extract common features from data coming from multiple subjects. Of course the results obtained on the subject-independent model are not as good as the subject-specific case, but there are many possible ways to improve the quality of the signals. In fact, data collected from different subjects might be noisier than single-subject's signals, but useful processing tools such as DTW algorithm, smoothing and normalization could significantly improve the goodness of our model.

The fact that this approach give good results with both subject-specific and subject-independent models created online and offline tells us that it is a good way to estimate this kind of physiological data.

The estimated joint angles have been remapped onto a robotic joint in order to online control two kind of robots: a manipulator for testing upper limbs and a humanoid for testing lower limbs. Testing the model on different dataset involving different movements and with different kind of robots definitely proves that the regression of a single joint bending angle is effectively achievable via Gaussian Mixture Model.

7.2 Future work

Despite the good results obtained in a quite new field which and with few proposed solutions, there are a lot of possible improvements. An interesting study will involve data of amputee subjects or with mobility disorders in order to test how the model change, since prostheses are not destined to healthy subjects and, of course, non-healthy people have even different ways of performing simple tasks like kicking a ball or walking.

Furthermore, part of this study regarded the selection of the channels and their number. Little changes in the position of the sensors could affect the goodness of the signal and, consequently, the goodness of the model. As further work, it would be interesting to perform an adaptive and dynamic selection of the features, adapting them to the performance obtained by the model.

We also want to study new and more fast methods to increase the reliability of the data with better smoothing and normalization techniques.

Moreover, controlling just one joint is very limited, it will be of great interest increase the control framework to multiple joints and perform a more depth study about the potentials of the incremental version of GMM. It will be very challenging studying the number of training trials needed by a new subject when using the system, to let the model fit to him.

As the robotics devices are intended to work with the humans, it is important to manage the safety of the users, handling all the potentially dangerous situations in which, for example, the prosthesis stops working or make a mistake computing the model.

A lot of work can be done about this interesting topic and, since this is just the beginning of the researches on this field, a lot of improvements can be reached. Let the robot learning from human demonstrations seams quite a science fictional topic, but it is not, and the fact that this research could help people with some kind of disability to live in a better way is even more satisfying and incredible.

References

[1] "Mobiligence program," http://www.race.u-tokyo.ac.jp/~ota/mobiligence/index_e.html.

[2] W. Ahmad, "Incremental learning of gaussian mixture models."

[3] A. Ajiboye and R. Weir, "A heuristic fuzzy logic approach to emg pattern recognition for multifunctional prosthesis control," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 13, no. 3, pp. 280–291, Sept 2005.

[4] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *2nd International Symposium on Information Theory*, 1973, pp. 267–281.

[5] P. Artemiadis and K. Kyriakopoulos, "A bio-inspired filtering framework for the emg-based control of robots," pp. 1155–1160, 2009.

[6] P. Artemiadis and K. Kyriakopoulos, "Emg-based control of a robot arm using low-dimensional embeddings," *Robotics, IEEE Transactions on*, vol. 26, no. 2, pp. 393–398, April 2010.

[7] P. Artemiadis and K. Kyriakopoulos, "An emg-based robot control scheme robust to time-varying emg signal features," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 3, pp. 582–588, May 2010.

[8] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific data*, vol. 1, 2014.

[9] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, 1998.

[10] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, “Robot programming by demonstration,” in *Springer handbook of robotics.* Springer, 2008, pp. 1371–1394.

[11] S. Calinon, F. Guenter, and A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, 2007.

[12] S. Calinon and A. Billard, “Incremental learning of gestures by imitation in a humanoid robot,” in *Proceedings of the ACM/IEEE international conference on Human-robot interaction.* ACM, 2007, pp. 255–262.

[13] S. Calinon and A. Billard, “A probabilistic programming by demonstration framework handling constraints in joint space and task space,” in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on.* IEEE, 2008, pp. 367–372.

[14] C. Castellini, A. Fiorilla, and G. Sandini, “Multi-subject/daily-life activity emg-based control of mechanical hands,” *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, p. 41, 2009. [Online]. Available: http://www.jneuroengrehab.com/content/6/1/41

[15] R. H. Chowdhury, M. B. I. Reaz, M. A. B. M. Ali, A. A. A. Bakar, K. Chellappan, and T. G. Chang, “Surface Electromyography Signal Processing and Classification Techniques,” *Sensors*, vol. 13, no. 9, pp. 12 431–12 466, 2013.

[16] J. Chu and Y. J. Lee, “Conjugate prior penalized learning of Gaussian mixture models for EMG pattern recognition,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[17] J.-U. Chu and Y.-J. Lee, “Conjugate-prior-penalized learning of gaussian mixture models for multifunction myoelectric hand control,” *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 3, pp. 287–297, 2009.

[18] L. Chun-Lin, “A tutorial of the wavelet transform,” *NTUEE, Taiwan*, 2010.

[19] E. A. Clancy, E. L. Morin, and R. Merletti, “Sampling, noise-reduction and amplitude estimation issues in surface electromyography,” *Journal of Electromyography and Kinesiology*, pp. 1–16, Feb. 2002. [Online]. Available: http://dx.doi.org/10.1016/S1050-6411(01)00033-5

[20] A. C. da Rocha Costa, R. M. Vicari, and F. Tonidandel, "Advances in artificial intelligence–sbia 2010," 2010.

[21] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *Information Theory, IEEE Transactions on*, vol. 36, no. 5, pp. 961–1005, Sep 1990.

[22] C. J. De Luca, "The use of surface electromyography in biomechanics," *Journal of applied biomechanics*, vol. 13, pp. 135–163, 1997.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[24] S. Du and M. Vuskovic, "Temporal vs. spectral approach to feature extraction from prehensile emg signals," in *Information Reuse and Integration, 2004. IRI 2004. Proceedings of the 2004 IEEE International Conference on*. IEEE, 2004, pp. 344–350.

[25] K. Englehart, B. Hudgin, and P. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *Biomedical Engineering, IEEE Transactions on*, vol. 48, no. 3, pp. 302–311, March 2001.

[26] M. Erkilinc and F. Sahin, "Camera control with emg signals using principal component analysis and support vector machines," in *Systems Conference (SysCon), 2011 IEEE International*, April 2011, pp. 417–421.

[27] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, "A human-assisting manipulator teleoperated by emg signals and arm motions," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 2, pp. 210–222, 2003.

[28] P. Gerus, M. Sartori, T. Besier, B. Fregly, S. Delp, S. Banks, M. Pandy, D. D'Lima, and D. Lloyd, "Subject-specific knee joint geometry improves predictions of medial tibiofemoral contact forces," *Journal of biomechanics*, vol. 46, no. 16, pp. 2778–2786, 2013.

[29] A. E. Gibson, M. Ison, and P. Artemiadis, "User-independent hand motion classification with electromyography," in *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2013, pp. V002T26A002–V002T26A002.

[30] A. Gijsberts, M. Atzori, C. Castellini, H. Muller, and B. Caputo, “Movement error rate for evaluation of machine learning methods for semg-based hand movement classification,” *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 22, no. 4, pp. 735–744, July 2014.

[31] P. Guglielminotti and R. Merletti, “Effect of electrode location on surface myoelectric signal variables: a simulation study,” in *9th Int. Congress of ISEK*, vol. 188, 1992.

[32] N. Hogan and R. W. Mann, “Myoelectric signal processing: Optimal estimation applied to electromyography - part i: Derivation of the optimal myoprocessor,” *Biomedical Engineering, IEEE Transactions on*, vol. BME-27, no. 7, pp. 382–395, July 1980.

[33] G. Huang, Z. Zhang, D. Zhang, and X. Zhu, “Spatio-spectral filters for low-density surface electromyographic signal classification,” vol. 51, no. 5, pp. 547–555, 2013.

[34] B. Hudgins, P. Parker, and R. Scott, “A new strategy for multifunction myoelectric control,” *Biomedical Engineering, IEEE Transactions on*, vol. 40, no. 1, pp. 82–94, Jan 1993.

[35] A. Ismail and S. Asfour, “Continuous wavelet transform application to EMG signals during human gait,” in *Signals, Systems amp; Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, vol. 1, Nov 1998, pp. 325–329 vol.1.

[36] M. Ison and P. Artemiadis, “Beyond user-specificity for emg decoding using multiresolution muscle synergy analysis,” in *ASME 2013 Dynamic Systems and Control Conference.* American Society of Mechanical Engineers, 2013, pp. V001T08A006–V001T08A006.

[37] R. Khushaba, “Correlation analysis of electromyogram signals for multiuser myoelectric interfaces,” *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 22, no. 4, pp. 745–755, July 2014.

[38] R. Khushaba, A. Al-Ani, and A. Al-Jumaily, “Orthogonal fuzzy neighborhood discriminant analysis for multifunction myoelectric hand control,” *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 6, pp. 1410–1419, June 2010.

[39] R. Khushaba, M. Takruri, J. Miro, and S. Kodagoda, "Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features," *Neural Network*, vol. 55, pp. 42–58, 2014.

[40] M. Kristan, D. Skocaj, and A. Leonardis, "Incremental learning with gaussian mixture models," in *Proc. Computer Vision Winter Workshop, Moravske Toplice, Slovenia*, 2008, pp. 25–32.

[41] T. Lalitharatne, Y. Hayashi, K. Teramoto, and K. Kiguchi, "A study on effects of muscle fatigue on EMG-based control for human upper-limb power-assist," in *Information and Automation for Sustainability (ICIAfS), 2012 IEEE 6th International Conference on*, Sept 2012, pp. 124–128.

[42] F. Laterza and G. Olmo, "Analysis of EMG signals by means of the matched wavelet transform," *Electronics Letters*, vol. 33, no. 5, pp. 357–359, Feb 1997.

[43] S. Lee and G. N. Saridis, "The control of a prosthetic arm by emg pattern recognition," *Automatic Control, IEEE Transactions on*, vol. 29, no. 4, pp. 290–302, 1984.

[44] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 2010, pp. 9–14.

[45] D. Lloyd and T. Besier, "An emg-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo," *Journal of biomechanics*, vol. 36, no. 6, pp. 765–776, 2003.

[46] C. Loconsole, S. Dettori, A. Frisoli, C. A. Avizzano, and M. Bergamasco, "An emg-based approach for on-line predicted torque control in robotic-assisted rehabilitation," in *Haptics Symposium (HAPTICS), 2014 IEEE*. IEEE, 2014, pp. 181–186.

[47] T. Matsubara and J. Morimoto, "Bilinear modeling of emg signals to extract user-independent features for multiuser myoelectric interface," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 8, pp. 2205–2213, Aug 2013.

[48] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[49] S. Micera, A. Sabatini, P. Dario, and B. Rossi, "A hybrid approach to emg pattern analysis for classification of arm movements using statistical and fuzzy techniques," *Medical engineering & physics*, vol. 21, no. 5, pp. 303–311, 1999.

[50] S. Michieletto, N. Chessa, and E. Menegatti, "Learning how to approach industrial robot tasks from natural demonstrations," in *Advanced Robotics and its Social Impacts (ARSO), 2013 IEEE Workshop on.* IEEE, 2013, pp. 255–260.

[51] S. Michieletto, S. Ghidoni, E. Pagello, M. Moro, and E. Menegatti, "Why teach robotics using ros?" *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 8, no. 1, pp. 60–68, 2014.

[52] S. Michieletto, L. Tonin, M. Antonello, R. Bortoletto, F. Spolaor, E. Pagello, and E. Menegatti, "Gmm-based single-joint angle estimation using emg signals," in *The 13th Int. Conf. on Intelligent Autonomous Systems*, 2014.

[53] S. Michieletto, E. Tosello, F. Romanelli, V. Ferrara, and E. Menegatti, "Ros-i interface for comau robots," in *Simulation, Modeling, and Programming for Autonomous Robots.* Springer International Publishing, 2014, pp. 243–254.

[54] S. Michieletto, D. Zanin, and E. Menegatti, "Nao robot simulation for service robotics purposes," in *Modelling Symposium (EMS), 2013 European.* IEEE, 2013, pp. 477–482.

[55] J. Millan, P. Ferrez, F. Galan, E. Lew, and R. Chavarriaga, "Non-invasive brain-machine interaction," 2008.

[56] S. Muceli, N. Jiang, and D. Farina, "Extracting signals robust to electrode number and shift for online simultaneous and proportional myoelectric control by factorization algorithms," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 22, no. 3, pp. 623–633, May 2014.

[57] F. Orabona, C. Castellini, B. Caputo, A. Fiorilla, and G. Sandini, "Model adaptation with least-squares svm for adaptive hand prosthetics," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* IEEE, 2009, pp. 2897–2903.

[58] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "Application of Wavelet Analysis in EMG Feature Extraction for Pattern Classification," *Measurement Science Review*, vol. 11, no. 2, pp. 45–52, 2011.

[59] D. Reynolds, R. C. Rose, *et al.*, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[60] J. Romero, T. Feix, H. Kjellstrom, and D. Kragic, "Spatio-temporal modeling of grasping actions," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on.* IEEE, 2010, pp. 2103–2108.

[61] U. Sahin and F. Sahin, "Pattern recognition with surface emg signal based wavelet transformation," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on.* IEEE, 2012, pp. 295–300.

[62] M. Sartori, M. Reggiani, E. Pagello, and D. Lloyd, "Modeling the Human Knee for Assistive Technologies," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 9, pp. 2642–2649, Sept 2012.

[63] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[64] Y. Sheng, "Wavelet transform," *The transforms and applications handbook*, pp. 747–827, 1996.

[65] F. Stival, S. Michieletto, and E. Pagello, "Online subject-independent modeling of semg signals for the motion of a single robot joint," in *IEEE International Conference on Robotics and Automation, 2016 (ICRA2016). (submitted).*

[66] F. Stival, S. Michieletto, and E. Pagello, "Subject-independent modeling of semg signals for the motion of a single robot joint," in *Workshop of Robotics: Science and Systems 2015 on Combining AI Reasoning and Cognitive Science with Robotics*, 2015.

[67] A. Subasi, "Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction," *Computers in Biology and Medicine*, vol. 37, no. 2, pp. 227–244, 2007.

[68] A. Subasi, "EEG signal classification using wavelet feature extraction and a mixture of expert model," *Expert Systems with Applications*, vol. 32, no. 4, pp. 1084–1093, 2007.

[69] M. Suresh, P. Krishnamohan, and M. Holi, "Gmm modeling of person information from emg signals," in *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*. IEEE, 2011, pp. 712–717.

[70] D. Tkach, H. Huang, and T. Kuiken, "Study of stability of time-domain features for electromyographic pattern recognition," *Journal of NeuroEngineering and Rehabilitation*, vol. 7, no. 1, p. 21, 2010. [Online]. Available: http://www.jneuroengrehab.com/content/7/1/21

[71] G. Tomasi, F. van den Berg, and C. Andersson, "Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data," *Journal of Chemometrics*, vol. 18, no. 5, pp. 231–241, 2004.

[72] R. Valentini, S. Michieletto, F. Spolaor, Z. Sawacha, and E. Pagello, "Processing of semg signals for online motion of a single robot joint through gmm modelization," 2015.

[73] M. Vitor-Costa, L. Pereira, R. Oliveira, R. Pedro, T. Camata, T. Abrao, M. Brunetto, and L. Altimari, "Fourier and wavelet spectral analysis of emg signals in maximal constant load dynamic exercise," in *Conference Proceedings (IEEE Engineering in Medicine and Biology Society Conf)*, vol. 1, 2010, pp. 4622–4625.

[74] C. S. Wallace and D. L. Dowe, "Minimum message length and kolmogorov complexity," *The Computer Journal*, vol. 42, no. 4, pp. 270–283, 1999.

[75] L. Wang and T. S. Buchanan, "Prediction of joint moments using a neural network model of muscle activations from emg signals," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 10, no. 1, pp. 30–37, 2002.

[76] F. E. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control." *Critical reviews in biomedical engineering*, vol. 17, no. 4, pp. 359–411, 1989. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/2676342

[77] M. Zardoshti-Kermani, B. Wheeler, K. Badie, and R. Hashemi, "Emg feature evaluation for movement control of upper extremity prostheses," *Rehabilitation Engineering, IEEE Transactions on*, vol. 3, no. 4, pp. 324–333, Dec 1995.