



Politecnico di Torino

Master Thesis

Master degree in Electronic Engineering

Control of a Myoelectric Prosthesis

Student:

Maxime Bourges

Supervisor:

Prof. Luca Mesin

A.A. 2020/2021

Abstract

Starting from the electromyography, usually recorded on the surface of the skin, a prosthesis can be controlled following a classical step-flow. By adding a real-time pre-processing to a single-channel surface electromyography, a new signal is obtained: the deconvolution, which provides an estimation of the firing rates of the muscle units involved in the motion. In this work, after giving an overview of the classical pattern control strategies, we resolved a classification problem with both the electromyography and the associated deconvolution. The problem is: 10 motion classes, 2 recording channels on 10 healthy subjects. Classical time-domain features are extracted from the signals and reduced by Mutual Component Analysis, and the classification is done by both a Support Vector Machine and a k-Nearest Neighbours. The overall results of this work are that the classification results are better and more robust when using the estimation of the firing rates than the classical signal. Even if deeper insight in this new technic is required to state a clear advantage over the electromyography, this preliminary work gives promising outcomes for a further use of the deconvolution as prosthesis control technic.

Table of Contents

Abstract.....	1
Glossary	5
Introduction	7
I Muscle Contraction, EMG and detection.....	10
1. Anatomy	10
2. Muscle contraction and electrical command	12
a. MU recruitment.....	12
b. Action Potential creation	16
c. Propagation of the AP.....	22
d. Propagation to the skin.....	25
3. EMG detection.....	27
a. Intramuscular EMG.....	27
b. Surface EMG	28
Conclusion.....	33
II Pre-processing of data.....	34
1 Windowing.....	34
a. Why dividing the EMG in several time periods?.....	34
b. Disjoint windows.....	35
c. Sliding Windows.....	36
2 Filtering.....	38
III Features extraction and reduction	40
1 Features extraction	40
a. Time Domain features.....	40
b. Frequency Domain Features	42
c. Time-Frequency Domain Features	42
2. Reduction of the feature set	44
a. Why do we need to reduce the feature set?.....	44
b. Principal Component Analysis (PCA)	45

c. Linear Discriminant Analysis	47
d. Mutual Component Analysis	49
IV. Classifiers	52
1. Linear Discriminant Analysis	52
2. K Nearest Neighbours	54
3. Support Vector Machine	56
4. Artificial Neural Networks	60
5. Classification trees	65
6. Gaussian Mixture Model	68
V. Post-processing	71
1. Majority Voter	71
2. Bayesian Fusion	72
VI. Beyond EMG classification: using deconvolution as a new technic.	76
1. Other signals used in the literature	76
2. Deconvolution signal	78
a. What is it?	78
b. Uses of MU's discharges in the frame of a classification problem	79
c. Single channel deconvolution	80
3. Classification	83
a. Method	83
b. Results	86
4. Perspectives of this technic	91
Conclusion	95
Appendix	96
MCA algorithm	96
Références	98

Table of figures

Figure 1-Classical flowchart of the resolution of a pattern classification problem for limb movements.....	8
Figure 2-Recruitment Threshold Excitation of a pool of neurons according to Fuglevand's model	14
Figure 3-Firing rates of the neurons from a pool of 6 neurons, depending on the excitation level.....	15
Figure 4-Hodgkin-Huxley model for transmembrane voltage.....	18
Figure 5-Model of a muscle fibre in the cable theory.....	22
Figure 6-Tripole current along muscle fibre in cable theory	25
Figure 7-Electrical model of a surface electrode	29
Figure 8-Temporal partition of a signal according to the disjoint windows scheme.....	36
Figure 9-Temporal partition of a signal according to the sliding windows scheme.....	37
Figure 10-Division of the Time-Frequency space in the different TFR.....	43
Figure 11-Representation of a PCA in a two-features space.....	46
Figure 12-Partition of the space with 10 training points for a kNN (k=1), known as Voronoi graph	55
Figure 13-Support Machine Vector for a two-classes problem in a two-features space.....	57
Figure 14-Representation of a simple Artificial Neural Network, with a simple forward topology and two hidden layers.	61
Figure 15-Representation of an exemple of a Classification Tree.....	66
Figure 16-Motions considered in the pattern classification problem considered in this Master Thesis	84
Figure 17 - Boxplots and mean classification error rate for TD features, SVM and MV with 11 decisions, depending on the number of features and the signal used	88
Figure 18 - Number of occurrences of every feature over 10 sets of features (one for each subject).	90

Glossary

ANN: Artificial Neural Network

AP: Action Potential

BSS: Blind Source Separation

DoF: Degrees of Freedom

EMG: Electromyogram

FD: Frequency Domain

GMM: Gaussian Mixture Model

HH (model): Hodgkin-Huxley (model)

IAV: Integrate Absolute Value

ISI: Inter-Spike Interval

IZ: Innervation Zone

kNN: k Nearest Neighbours

LDA: Linear Discriminant Analysis

MAV: Mean Absolute Value

MCA: Mutual Component Analysis

MU: Motor Unit

MV: Majority Voter

PCA: Principal Components Analysis

RMS: Root Mean Square

RTE: Recruitment Threshold Excitation

SSC: Slope Sign Change

STFT: Short Time Fourier Transform

SVM: Support Vector Machine

TD: Time Domain

TFR: Time-Frequency Representation

ULDA: Uncorrelated LDA

WL: Waveform Length

WPT: Wavelet Packet Transform

WT: Wavelet Transform

ZC: Zero Crossings

Introduction

For several decades now, myoelectric prostheses have been under investigation. Indeed, these devices allow people who lost a limb to partially recover motion capabilities, such as grasping for arm amputees or walking for leg amputees, and so to improve their quality of life. Body-powered mechanical prostheses, which are control by the power of the patient's residual limb, can be used by patients to retrieve some functions [1] [2]. These devices are simple but cannot reproduce a big range of motions: for example, the hand is usually modelled by a hook that perform only grasping functions. On the other hand, myoelectric artificial limbs can be controlled by recording the muscle activity, and the motions are done thanks to actuators and external power sources [2]. In 1955, Batty showed the possibility to use the electromyogram (EMG) to control an artificial upper limb [3]. Indeed, even in the case of a patient who had to undergo an amputation, the central nervous system still "controls" the phantom limb by sending set of impulses. Recording them thanks to electrodes allows to get necessary information to control a device.

If the first models of prosthesis were simple on/off devices (the only motion available usually being the grasp for upper-limb prostheses) controlled by the amplitude of the recorded EMG [2] [3] [4], pattern classification with multiple degrees-of-freedom (DoF) have been developed for the last decades [5]: many different patterns can be reproduced today. The motivation is simple: covering a wider range of motions leads to a better user experience, since it tends to mimic an actual limb and everyday life tasks can be performed [6]. Researchers can now control each finger individually with a marvellous success rate (the rate of correctly classified motions) [7] [8]. Nevertheless, the output of the devices

becoming more complex, the classification algorithm could not be based only on an amplitude threshold anymore, and more sophisticated methods are needed. In this scope, even if several technics to decipher the intended movement and to control a prosthesis from the EMG have been proposed [2], we will focus on pattern recognition classification which follows a classical step flow [2] [9]:

1. Pre-processing of raw EMG
2. Extraction of features
3. Selection/reduction of the features set
4. Classification
5. Post-processing

This succession of computations is shown on Figure 1.

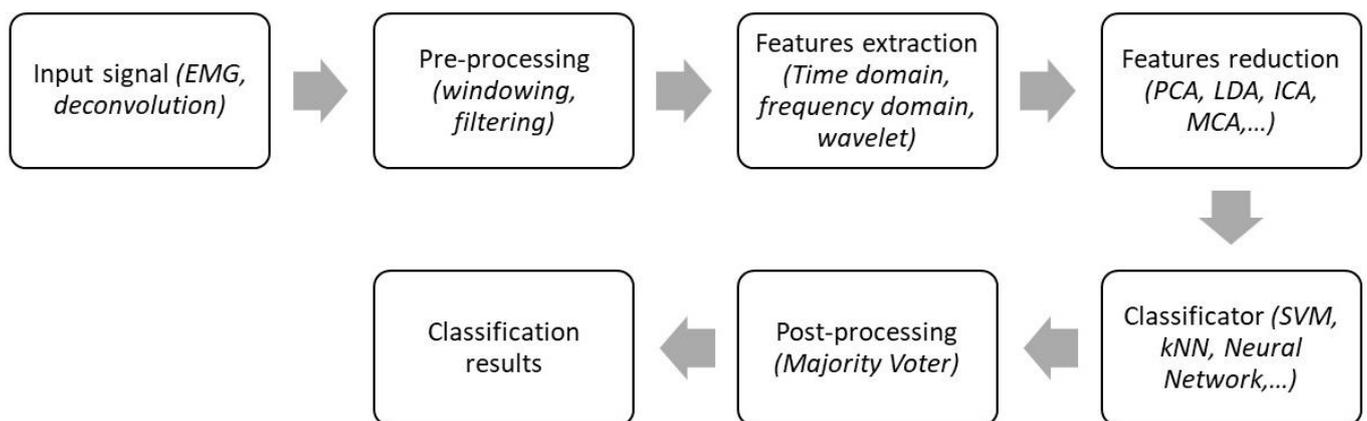


Figure 1-Classical flowchart of the resolution of a pattern classification problem for limb movements

Even if good performances are achieved in the frame of pattern control, many improvements can be done, and many researchers are currently working to propose technics that could enhance the prosthetic control. The specialists advocate for few improvement axes. The first one is continuing to increase the controllable DoF with a low error, to further improve the user experience by approximating better an actual limb, without confusing the different motions

[10] [11]. The control should also be as robust as possible toward non-ideal conditions and the changes of EMG [11]: in the presence of sweat or when the user is tired, the recorded EMG can vary from the recording in laboratory conditions [12] [13] and so degrades the performances of the algorithm. A good control should not be sensitive to these changes and should overcome these problems that will surely occur during the use of the device. Finally, the prosthesis should provide to the user a sensory feedback. Most of devices only integrate the feed-forward, i.e. sending the control to the prosthesis, with the simplest feedback (the vision of the motion). Closing the loop by adding a sensory feedback (adding an applied pressure to the residual limb for example) could lead to more efficient control, learning and adaptation [6] [10] [11].

In this work, we will give a non-exhaustive overview of the prosthetic pattern control, with the strategies most used. We will first recall the basis of muscle anatomy, how the contraction works and how we can record this activity. Then, for each stage of the classical pattern control, we will explain the purpose of the step and present the most used technics in the literature. Finally, a promising method of control, based on the deconvolution of the EMG, will be introduced.

I Muscle Contraction, EMG and detection

Before diving into the description of the classification, it is essential to understand the mechanisms of a limb gesture. The movements done by an individual can be divided in two categories: reflexes and intentional movements. The first ones are rapid responses of the body to an external stimulus, and they are done “automatically” by the subject: the command comes from the peripheral nervous system (the spinal cord), not from the central nervous system. Since the prosthesis aims at reproducing the movement intended by the user, the reflexes will be kept out of this subject and we will only focus on voluntary contraction.

Understanding how a limb moves, what happens in the muscle during a movement, is a key to understand the control of a prosthesis. We will first look at the anatomy of a muscle, then we will present how a voluntary movement is commanded, and we will finish this part by examining how a monitoring of the activity can be achieved. Most information presented here are taken from [14], when not specified otherwise.

1. Anatomy

The organs responsible for the movements are the skeletal muscles. By contracting, either the length of the muscle change (isotonic contraction) or a tension is created (isometric contraction). In either case, the tension or the movement is transmitted to the bones and makes the concerned limb move in the case of an isotonic contraction, as bones and muscles are attached to each

other thanks to tendons. Note that other types of muscles (not attached to any bone) exist, the cardiac muscle for example is not responsible for a motion, but we will only focus on the skeletal muscles here.

A muscle can be described as several muscular fibres aligned in the longitudinal direction of the organ. They are the parts of the muscle that contract, and so induce the motion of the limb. Indeed, at the end of the fibres, there are tendons fixed to the bones which put them into motion. These fibres can be classified in two categories [15]:

- Type I, or red fibres: rich in blood and oxygen, they can contract for a long period of time, but can only produce little force twitches.
- Type II, or white fibres: poor in blood and oxygen, they can produce high force twitches, but only for short periods of time.

The fibres are also linked to the central nervous system (where the order of movement comes from) by the neuromuscular junctions in the innervation zone (IZ), i.e. where the nerves who propagate the information of the wanted motion are linked to muscular fibres.

The contraction of a given fibre is induced by an electrical signal propagating along it, called Action Potential (AP) [15]. This is a brief depolarization of the membrane of a muscle fibre or of a neuron, and it causes the aperture of voltage-sensitive and selective channels allowing the flow of certain ions across the membrane of the cell. Especially, calcium ions Ca^{2+} flow into the fibre cytoplasm, which induces the release of more calcium ions, and eventually the concentration of these ions allows actin-myosin activity which leads to the contraction [16].

The smallest functional unit in the muscle is named motor unit (MU). Each muscle is usually formed by several MU, themselves formed by a neuron and the

muscular fibres innervated by it [15]. For a given motion wanted by the patient, the nerves activate the needed units that will contract according to the command (to the force for example). The MU can also be divided in two categories: the small ones, involved in all contractions, and the large ones only involved in intensive contractions.

Now we have reviewed the main physiological elements involved in a movement, we can go deeper in the explanations of what happens when a contraction is commanded by the patient.

2. Muscle contraction and electrical command

Let us imagine a patient who wants to perform a gesture. Thus, the central nervous system, especially the brain, sends an order to contract the concerned muscles through nerves. This order is under the form of a bio-electrical signal. As we said before, a muscle is divided in several MU, and not all of them are concerned with every contraction: it depends on the force level required. Once the concerned MUs get excited, an AP is created and propagates along the muscle fibres. We will see here the most accepted models explaining these phenomena.

a. MU recruitment

The force developed during contraction depends on two parameters: the size of the involved MU and their firing rate, which is the frequency at which the neurons stimulate the fibres. The Henneman's size principle states that the

smallest MU, able to furnish only small force levels, are the ones recruited first and as the commanded force increases, the bigger units are recruited in addition [17]. Fuglevand et al. [18] proposed a model to describe the activation of the MU and the behaviour of the fire rate. In this model, a recruitment threshold is assigned to each MU of the considered pool. If the command, related to the wanted force exceeds, the recruitment threshold of a unit, the MU starts firing (i.e. being excited by the neuron and contracts). Researchers have noted that there were many small units always (or almost always) active, so with a low threshold, and as the force level (and so the command) increase, only few big units become active. To fit these observations, a model of threshold repartition was proposed, which distributes the threshold of each MU with an exponential law:

$$\text{RTE}(i) = e^{\alpha i} \quad (\text{I. 2.1})$$

With:

- $\text{RTE}(i)$: recruitment threshold excitation of the i^{th} motoneuron
- i : index identifying the motor neuron
- α : coefficient to set the range of threshold value. It is given by: $\alpha = \frac{\ln(R)}{n}$
with R the desired range of threshold and n the number of motoneurons.

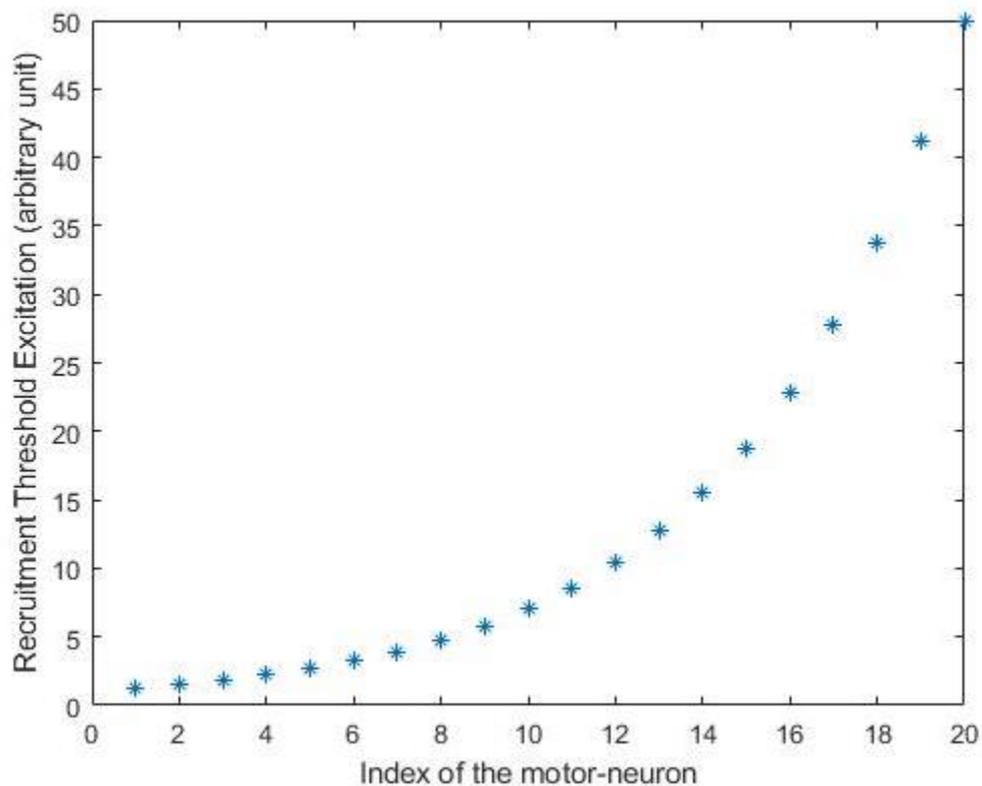


Figure 2-Recruitment Threshold Excitation of a pool of neurons according to Fuglevand's model. The Range of Recruitment is arbitrarily set to 50 (arbitrary unit), and the RTE is given in an arbitrary unit.

Once we established which units are involved in the motion, we still must determine at which frequency they will discharge (i.e. get excited by the neuron, contract and return to idle state). When the excitation level reaches the threshold of a unit, the MU starts firing at its minimum rate. Studies tend to show that this minimum frequency is the same for all MU, regardless their size or their recruitment threshold [19] [20]. It is approximately 8 impulses/s [20]. As the input level increases, the firing rate increases. The best input-output relationship is a single linear model [21], with a gain uncorrelated with the size of the units, until it reaches the maximum frequency of the unit and the output rate saturates. The saturation rate depends on the MU and is ranged between 20 and 45 impulses per seconds [19].

This model can be mathematically written under the form:

$$FR_i(t) = \begin{cases} 0 & \text{if } E(t) < RTE(i) \\ g_E * [E(t) - RTE(i)] + FR_{\min} & \text{if } RTE(i) < E(t) < RTE_{SAT}(i) \\ FR_{\max} & \text{if } E(t) > RTE_{SAT}(i) \end{cases} \quad (I. 2.2)$$

With:

- FR_i : firing rate of the i^{th} motor neuron
- $E(t)$: excitation level
- g_E : gain
- $RTE_{SAT}(i)$: threshold for which the maximum firing rate is reached for the i^{th} motor neuron

The last part of the model considers that the time between two consecutive

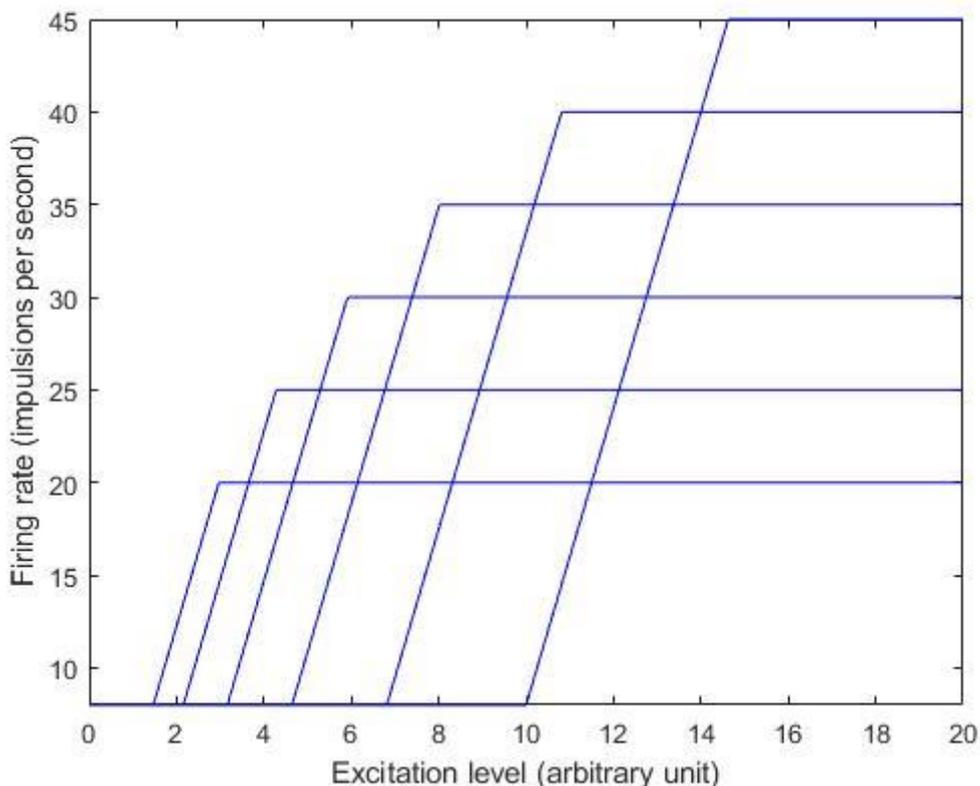


Figure 3-Firing rates of the neurons from a pool of 6 neurons, depending on the excitation level. Their saturation firing rates are between 20 and 45 impulsions per second. Their RTE are given by Fuglevand's model, with $R=10$. The gain is 8imp/s.

discharges is variable (a Gaussian noise is added to the inter-spike interval (ISI)).

So, MU are being activated and AP are created in these at a certain frequency. We will now try to understand the behaviour of the action potential in the fibres.

b. Action Potential creation

In an excitable cell, a neuron or a muscle fibre, the membrane at rest is polarized: in the case of muscle cells, the potential at rest inside the cell is negative compared to the extracellular fluid. When the cell is excited, a brief depolarization of the membrane occurs: that is the action potential.

The voltage between the two side of the membrane is induced by the concentrations and the fluxes through the membrane of two main ions, sodium ions Na^+ and potassium ions K^+ . At rest, Na^+ concentration is higher outside the cell than inside, and K^+ concentration is higher inside than outside. This condition is kept by the activity of the $\text{Na}^+ - \text{K}^+$ pump. Selective channels, allowing only one type of ion to pass through, are present across the membrane and the concentration gradient of each species determines the flux of them in their respective channels, by diffusion. This flux is however counteracted by another flux in the opposite direction, as a voltage difference is generated by the motion of the ions. This flux opposes the motions of ions. The equilibrium is reached when these two fluxes are equal [22]. However, when the cell gets excited, the permeabilities of the membrane toward ions change for a brief time, which leads to the depolarization of the membrane.

Ions channels can be either open or closed. Let a selective channel be open. The total flux is obtained with addition, with their respective directions taken into

account, of the diffusive flux, given by Fick's law, and the electrodynamic one, given by Planck's law. The Nernst-Planck equation is obtained as a result:

$$J = -D * \left(\nabla c + \frac{zC}{RT} F \nabla \phi \right) \quad (\text{I. 2.3})$$

With:

- J: diffusion flux density ($\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
- D: diffusion coefficient ($\text{m}^2 \cdot \text{s}^{-1}$)
- c: concentration of the species ($\text{mol} \cdot \text{m}^{-3}$)
- z: valence of the ionic species
- R: the universal gas constant ($\text{J} \cdot \text{K}^{-1} \cdot \text{mol}^{-1}$)
- T: temperature
- F: Faraday's constant ($\text{C} \cdot \text{mol}^{-1}$)
- ϕ : Electrical potential (V)

We established the equilibrium is reached when the flux is 0. If we consider the problem in a 1D version, we get as an equilibrium condition:

$$J = -D * \left(\frac{dc}{dx} + \frac{zF}{RT} c \frac{d\phi}{dx} \right) = 0 \quad (\text{I. 2.4})$$

That can be solved through a separation of variables methods, and the solution is the Nernst potential [22]:

$$V_{\text{eq}} = V_i - V_e = \frac{RT}{zF} \ln \left(\frac{c_e}{c_i} \right) \quad (\text{I. 2.5})$$

Nevertheless, this would be the voltage at the equilibrium if only one species of ions was present. In the presence of several ion types, if the Nernst potential of an ion is reached, the other ions are not in their equilibrium state. So, the actual rest potential is a weighted sum of each Nernst potential, no ions are in

equilibrium and they are still flowing through the membrane, but nonetheless the system reaches a dynamic equilibrium. Many models have been proposed to describe this state: a complete one considering a system of coupled equations (Nernst-Planck and Poisson equations), the one proposed by Goldman, Hodgkin and Katz under few assumptions [23] [24] and the linear model proposed by Hodgkin and Huxley (HH model) [25], the simplest.

In the HH model, the membrane of the excitable cell is modelled a parallel association of a capacitor (keeping ions on each side of it), conductors (modelling the channels by a conductance, so one conductance for each ion), voltage generators (representing the gradient of concentration, modelled by the Nernst potential for each ion) and two current sources (modelling the ion pumps).

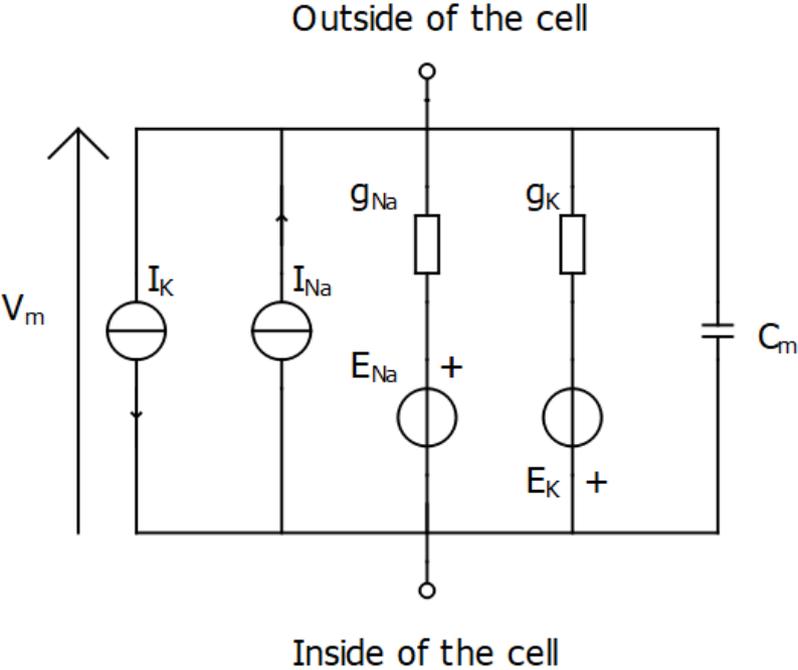


Figure 4-Hodgkin-Huxley model for transmembrane voltage.

The mathematical model is:

$$I = C_m \frac{dV_m}{dt} + g_K(V_m - V_K) + g_{Na}(V_m - V_{Na}) \quad (\text{I. 2.6})$$

With:

- V_m : voltage across the membrane
- V_X : Nernst voltage of the X species
- g_X : channel conductance for the X species
- I : total membrane current per unit area

The potential difference between the outside and the inside at equilibrium is the weighted sum of the Nernst potentials of each ion:

$$V_m^{eq} = \frac{\sum_i g_i V_i^{eq}}{\sum_i g_i} \quad (\text{I. 2.7})$$

At rest, this potential is around -70mV [15].

As we said, channels can be either closed or open. The kinetics of aperture and closure of the channels change the permeability of the channels, thus the conductance in the HH model. Therefore, it is essential to understand the dynamics of the channels to understand how the AP works.

Sodium and potassium channels can both be described by sequences of four subunits. A sodium channel is formed by three subunits of type m , which are activation gates, and one subunit of type h , which is an inhibition gate [25]. Each of these subunits are independent and can be either closed or open. The conductance of the sodium is denoted:

$$g_{Na} = \bar{g}_{Na} m^3 h \quad (\text{I. 2.8})$$

With:

- m, h : variables describing the state of their respective channel (0 when closed, and increasing as the unit closes)
- \bar{g}_{Na} : a constant

The potassium channel is, on the other hand, formed by four independent subunits of type n . The conductance of the potassium is given by:

$$g_K = \bar{g}_K n^4 \quad (\text{I. 2.9})$$

With:

- n : variable describing the state of the n units
- \bar{g}_K : a constant

The dynamics of these subunits can be described by the following equations:

$$\dot{m} = \frac{m_\infty(V) - m}{\tau_m(V)} \quad (\text{I. 2.10})$$

$$\dot{h} = \frac{h_\infty(V) - h}{\tau_h(V)} \quad (\text{I. 2.11})$$

$$\dot{n} = \frac{n_\infty(V) - n}{\tau_n(V)} \quad (\text{I. 2.12})$$

The steady-state values of these variables, as well as the time constants for the aperture/closure, depend on the voltage across the membrane. We will not give the analytical dependency [25] here, but we can give a qualitative behaviour:

- m_{∞} and n_{∞} , representing steady states of activation gates, increases with the voltage. So, the m and n subunits are closed in rest conditions and tend to open when V rises (so during the depolarization).
- h_{∞} , representing steady states of inhibition gate, is high at rest and decreases when the voltage increases. The h subunit is so open during rest and tends to close during depolarization.
- τ_m is much smaller than τ_h and τ_n . The consequence is that the sodium channels open rapidly (m open before h close), faster than the potassium ones.

At rest, in a muscle fibre, more potassium channels than sodium channels are open (for low V , $n_{\infty}^4 > m_{\infty}^3 h_{\infty}$). Hence, adopting the HH model, the voltage across the membrane is negative since the Nernst potential of potassium is negative (see equation (I. 2.7)).

Let a depolarization, i.e. an increase in the membrane voltage due to the excitation from the neuron, occurs. If the depolarization stays below a given threshold, the conductances of the channels do not change much, and the system returns to its equilibrium soon. On the other hand, if the threshold is reached by the depolarization, the channels begin to open or to close depending to their nature. The m subunits of the sodium channel, the fastest ones, begin to open while the h subunit remains open and the n subunits remain closed. During this phase, the conductivity of the sodium channel increases, leading to a further increase of the voltage as the sodium can flow inside the cell and the potassium

cannot flow outside (positive feedback). After a sufficient time, the slower subunits begin to change state: the inhibition h units close and the potassium n units open. The effect is a return to rest state, as the chemical equilibrium can be restored [22].

This is the mechanism of the action potential in the muscle fibre. Note that, due to the slow dynamic of the gates, a hyperpolarization (voltage below the rest potential) occurs after the repolarization. As long as the rest conditions are not restored, the cell remains insensitive to a second excitation (the n and h units are still open), which explains the maximum firing rate of a fibre [25].

This is what happens locally during an AP. Nevertheless, it does not occur at the same time everywhere in the fibre: it propagates, as other physical signals.

c. Propagation of the AP

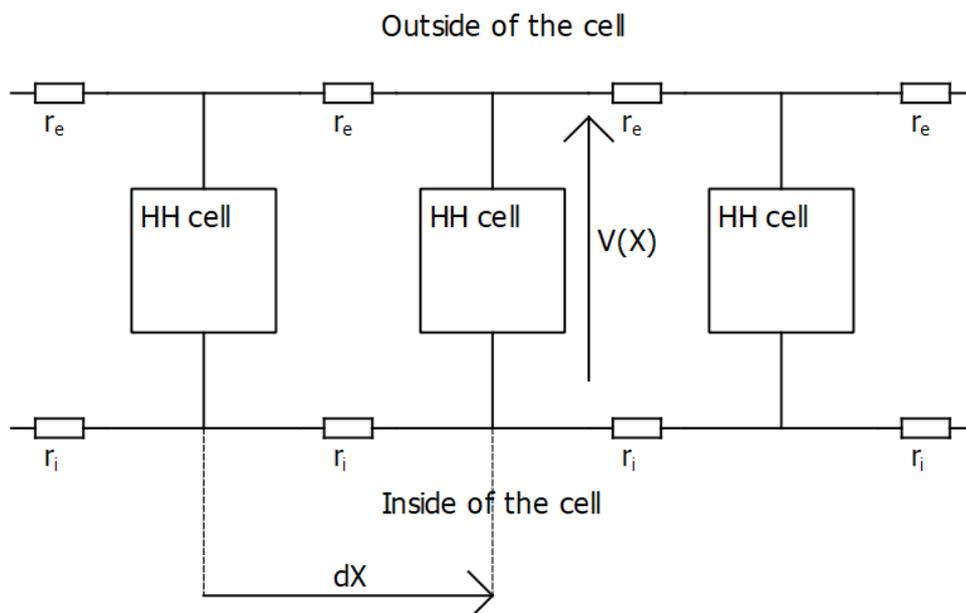


Figure 5-Model of a muscle fibre in the cable theory

The cable theory has been developed [26] to model the propagation of the AP along the muscle fibres. This theory aims at computing the voltage and the

current wherever in the fibre, by modelling it with an electrical circuit. The fibre is discretized in several units, that we will call cells here, linked to each other with distributed resistances defined along the axon of the fibre, inside and outside it. Each cell represents the current flowing through the membrane of the fibre. HH model can be used.

When a portion of the fibre is depolarized, i.e. the voltage at the poles of a cell changes sign, a current starts flowing through this given cell. Consequently, return currents appear in the surrounding cells, creating circles of flowing currents and depolarizing these cells. Thus, the AP propagates to the neighbouring cells, except the ones which has been previously depolarized (because of the long time-constants of the channels we already mentioned in *1.2.b-Action Potential creation*). So, the AP goes only toward one direction.

In the cable theory, the distance between each cell is dX , the resistances per unit length are denote r_i and r_e for respectively the interior and the exterior of the fibre, and the capacitance per unit length of the membrane is c_m . If we denote p the perimeter of the axon, I_{Ion} a term describing the I-V relation of the ion channels, and V the transmembrane voltage at position X , the cable equation is given by:

$$\frac{1}{r_i + r_e} \frac{\partial^2 V}{\partial X^2} = p \left(c_m \frac{dV}{dt} + I_{Ion}(V) \right) \quad (I. 2.13)$$

The voltage evolution of the action potential along the fibre axis can be modelled as an exponential decay. The action potential is initiated at the IZ and propagates to the tendons: the space variable along the fibre, z , is used:

$$V_m(z) = \begin{cases} Az^3 e^{-z} + B & \text{for } z \in [z_{min}; z_{max}] \\ 0 & \text{otherwise} \end{cases} \quad (I. 2.14)$$

With:

- $A = 96\text{mV}\cdot\text{mm}^3$ a constant
- $B = -90\text{mV}$ a constant
- z_{\min} and z_{\max} the bounds' coordinates of the membrane

The transmembrane current by unit area is, according to cable equation, proportional to the second derivative of the transmembrane voltage. The solution can be approximated by 3 Dirac functions, thus leading to the tripole current model, proposed by Rosenfalck [27]. In the depolarized zone, the tripole current models the currents entering and exiting the fibres with respectively a delta δ Dirac function of amplitude I (current sink), and two delta Dirac function with respective amplitudes of $\frac{2}{3}I$, 2mm ahead of the sink, and $\frac{1}{3}I$, 4mm behind the sink. The two last currents are called current sources. These three currents propagate toward the end of the fibre, conserving the distance between them and their amplitudes. When the first pole arrives to the tendon, it stops propagating while the two other poles continue moving toward the tendon. When the second pole reaches the end of the fibre, the two poles superimpose: the outcome is a delta Dirac function of $\frac{1}{3}I$. This resulting pole does not move, and the tripole becomes a dipole. The last pole continues its propagation until arriving to the tendon. It superimposes with the other pole, and the AP extinguishes.

The speed at which the AP propagates in the fibre is called conduction velocity (CV). It is a value of interest, as it is related to the type of fibre (fast fibres have larger CV than slower fibre) and the fatigue. The fatigue is characterized by some

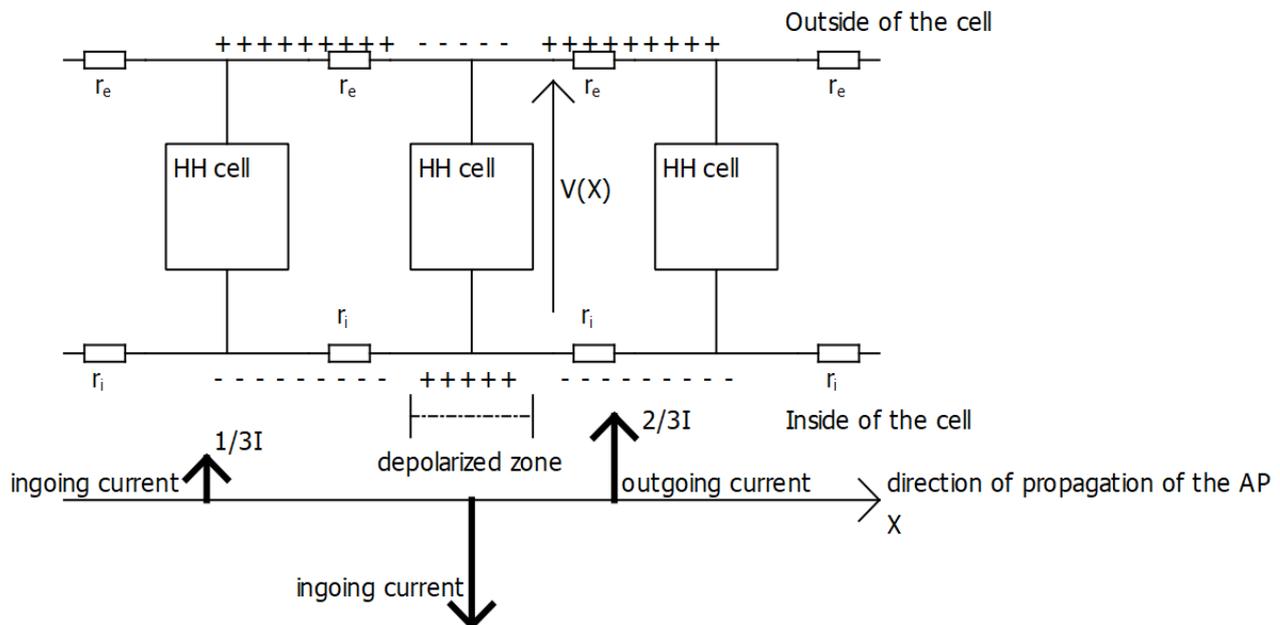


Figure 6-Tripole current along muscle fibre in cable theory

changes in the properties of the contraction, including the decrease of the maximal developed force and of the contraction velocity. These changes induce variations in the EMG.

d. Propagation to the skin

The trigger for the muscle contraction is the propagation of an electric signal along its concerned fibres. The current arising in the muscle fibres leads to the generation of a potential in the surrounding tissues, up to the skin surface: it acts like a source in an electromagnetic problem. The problem in the frame of bio-tissues is called Volume Conductor problem, and we will describe it here.

The electromagnetic equations can describe the problem. Given low frequencies, a quasi-static approximation can be done. Thus, the dielectric

properties are neglected, and the volume conductor can be described with only the conductivity tensor. Especially, we are in presence of a source, the electrical behaviour of the volume can be described by Poisson's equation [28]:

$$\nabla \cdot (\underline{\underline{\sigma}} \nabla \phi) = -I \quad (\text{I. 2.15})$$

With:

- ϕ : the electrical potential
- I : the current density of the source
- $\underline{\underline{\sigma}}$: conductivity tensor

The conductivity tensor can have different properties given how the problem is modeled. In a homogenous, isotropic, and infinite medium, the tensor is constant in time and space [29] (simplest model possible). More realistic models include planar [30] or cylindrical [31] layers (layers of muscle, fat, or other physiological tissues). In the case of several media, a conductivity tensor is defined for each layer.

To get a full model of the volume conductor, boundary conditions and the geometry of the system must be considered together with Poisson's equation. The usual boundary condition is given in (I.2.16), as the volume is considered insulated and no current can exit it.

$$J = -\underline{\underline{\sigma}} \nabla \phi = 0 \text{ on } \partial\Omega \quad (\text{I. 2.16})$$

With:

- $\partial\Omega$: the boundary of the volume conductor

Moreover, if several layers are considered, we must set the conditions at the interfaces in order to have a matching solution for all media. These conditions

are the continuity of the potential and of the current flow (see (I.2.17) and (I.2.18)).

$$\phi|_- = \phi|_+ \quad (\text{I.2.17})$$

$$\underline{\underline{\sigma}} \nabla \phi \cdot \vec{n}|_- = \underline{\underline{\sigma}} \nabla \phi \cdot \vec{n}|_+ \quad (\text{I.2.18})$$

With:

- -,+: designate each side of the interface
- \vec{n} : normal vector at the interface

An analytical solution cannot always be derived from this problem. We can compute one only under specific conditions [29] [30] [32]. Otherwise, numerical methods, such as finite element modelling, are used [28] [33].

3. EMG detection

Muscle activity is initiated by an electrical signal, triggered by the nerves, which propagates along muscle fibres and a potential is created in the whole conductive volume around the concerned muscle to the skin. Measuring this electrical signal gives an insight of the muscle activity and thus allows medical diagnosis [34] and prosthesis control.

The objective of this section is to give an overview of the recording technics and their inherent problems.

a. Intramuscular EMG

Since the bioelectrical signal, initiator and witness of the contraction of a muscle, originates inside the muscle fibres, the first idea is to record it directly from the source.

To do so, it is needed to place the electrode inside the muscle. The sensor tip is placed near the muscular fibres of interest and directly record the AP propagating along it. The sensors are usually on the end of thin wires or of a needle. The recorded signal is very selective [35], since only one muscle is under investigation (the detection volume, i.e. the volume for which a reliable potential can be obtained, is very small and so only local tissues are monitored). The pattern classification based on intramuscular EMG gives good results compared to surface EMG [36].

Nevertheless, the technic is invasive and requires a surgery to place the electrode inside the muscle of interest. This is the main drawback of the intramuscular EMG, and it is why most of researchers use surface EMG.

b. Surface EMG

As we said in the description of the volume conductor problem, under the effect of the source, namely the AP, a potential is created inside the tissues, from the muscular fibres to the skin surface. Therefore, with a proper equipment, it is possible to record the activity of the muscle on the surface of the skin, placed on the limb in motion. We will explore several aspects of this technic in this section. The information we present here are taken from [37] when not specified.

Electrode structure

There are different types of surface electrodes. The structural factors that can vary between them are their shapes, their physical dimensions, the technology, and the materials used. Thus, there are many ways to classify the available electrodes on the market (wet vs dry electrodes for example, based on either if the electrode includes a layer of conductive gel or not). In the frame of the recording of bio-signals such as surface EMG, the usual distinction is done between polarizable and non-polarizable electrodes. The first ones are characterized by a strong capacitive behaviour. This feature of polarizable electrodes is due to a double layer of charges at the metal-electrolyte interface. Consequently, there is no charge flow through the electrode-skin interface. With this kind of device, a displacement of the metal surface with respect to the electrolyte solution, due to the contraction of the muscle for example, induce a movement artifact (i.e. a change in the surface potential). Thus, they are not

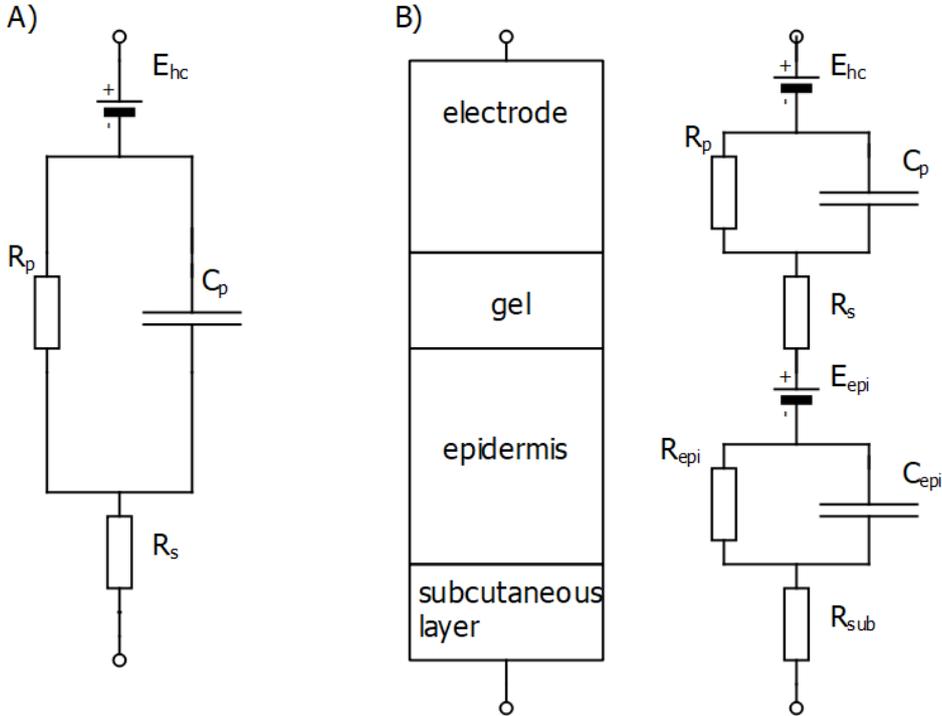


Figure 7-A) Electrical model of a surface electrode plus the conductive gel. B) Model of the whole electrode-skin interface.

suitable for the recording of EMG. So non-polarizable electrodes are used.

The interface between the non-polarizable electrode and the skin can be modelled by a non-linear RC circuit, whose components depend on both frequency and current (see Figure 7). This whole model is obtained by considering separately the electrode-electrolyte and the electrolyte-skin interfaces, and then serially connect them.

Disposition of the electrodes

We established earlier that the AP generates a potential on the skin. A reference potential is defined as the potential at a point far enough to be out of the source's influence. The simplest way to record the surface EMG at a given point is to measure the voltage between the chosen point and the reference point: this is the monopolar configuration. This configuration contains the whole information that can be got from the detection volume.

Other configurations are possible. These alternatives use more than one detection electrode (i.e. several monopolar signals), and the resulting signal is a linear combination of the inputs. The distance between two electrodes is the same everywhere and is called interelectrode distance (denoted IED). Doing so, the output acts like a spatial filter. The most popular dispositions are the Single Differential (SD), the Double Differential (DD) and the Laplacian. They respectively act like the first spatial derivative, the second spatial derivative and the double derivative along two normal directions. The linear combination of the signals from the electrodes is represented in matrix, in which non-zero elements represent the place of an electrode and the coefficient associated to its signal. Two adjacent numbers mean the two electrodes are away of the IED distance. The representations of the three filters we mentioned are given by:

Single differential: $(-1 \ 1)$

Double differential: $(-1 \ 2 \ -1)$

$$\text{Laplacian: } \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

The motivation for using such filters is the poor spatial selectivity of surface recording: indeed, with a monopolar configuration, the detection volume is large, so the contributions of many MU are included in the recording and it is hard to distinguish the different contributions. Using a spatial filter enhances the selectivity of the device by reducing the detection volume, and the different sources become more separable.

More recent researches use a further improvement of the network of electrodes by recording high density EMG (by using arrays of close, small electrodes). It provides temporal and spatial information about the electrical activity of the muscle, in particular a precise location of the IZ [38], estimation of AP's properties [39], or estimation of MU location [40].

Amplification

Given the low level of the EMG, an amplification stage is needed. A differential amplifier is used. The usual requirements for the amplification after recording are [41]:

- A high gain (typically between 1000 and 100 000)

- Low input noise density. Both this noise and the wanted signal are amplified, thus a noisy input will contaminate the whole chain. Typical wanted values are $1 - 60\text{nV}/\sqrt{\text{Hz}}$ and $1 - 100\text{fA}/\sqrt{\text{Hz}}$.
- High input impedance. It must be at least 100 times higher than the largest impedance of the electrode-skin interface. This is needed to get a reliable measure of the potential by reducing the effect of the electrode-skin impedance on the measure.
- High common mode ratio rejection (CMRR). An ideal differential amplifier amplifies only the difference of the two input signals (with a gain A_d). Nevertheless, a real amplifier does not fully reject signals present on both inputs (in particular, a capacitive coupling with the power line creates an interference voltage) and the common part is also amplified, with a gain A_c . The presence of this common components should be kept as low as possible. The CMRR is given by: $\text{CMRR} = 20 \log_{10} \left(\frac{A_d}{A_c} \right)$, and it should be around 100dB to limit power line interference.

Usual problems arising

One of the main problems of surface EMG is the low selectivity due to the large detection volume. So, the signal recorded by one electrode contains the contributions of several MU, and not only the ones from the targeted muscle. Furthermore, when several recording channels are placed on the limb of a patient, crosstalk between them arises. It means that signals from different channels have components in common and that they are not independent. Therefore, the information extracted from these signals will not independent

neither. As we said, spatial filtering can counteract the low selectivity, and crosstalk can also be decreased, by increasing the IED for example [42].

Moreover, there is an intrinsic noise in the electrode. Indeed, every electrical equipment creates electrical noise, and the electrode used to record the EMG is no exception. This is due to the charge carriers' flux at the metal-electrolyte interface and to the change of impedance of the electrode-skin interface. The impedance depends on the design of the device (on the surface notably [43]) and on time (drying of the gel, change in the properties of the skin or of the interface [44] [45]). Thus, the measure will be noisy in any case. This noise has a frequency range from 0Hz to several kHz with a level of $1 - 4\mu V_{RMS}$.

Conclusion

A voluntary contraction is accompanied by a bio-electrical signal along the muscle fibres, the Action Potential, that creates a potential in the volume surrounding the contracted muscle. This potential goes to the surface of the skin and can be measure with the proper equipment. The recorded signal gives an insight of muscle activity and can be used in the frame of prosthesis control.

The simpler models of artificial limb implement a single DoF controlled by a threshold on the EMG level. To control more than one DoF, a more complex process must be implemented.

II Pre-processing of data

Once the EMG is recorded, the classification process can begin. The first step is to prepare the data to ensure good results. This pre-processing is usually the windowing of the EMG and the filtering.

1 Windowing

The windowing of the time data is simply the division of it into a subset of smaller time series. In this way, the data will not be considered instantly or as the whole recording, but as a succession of smaller time series of a given length.

For example, let us imagine we recorded a 5s EMG from a subject doing a given motor task. We divide it into a subset of windows, of 200ms for example. Thus, the classifier will not give results for a continuous time, but there will instead be an output motion class (i.e. an assigned motion, among the motions that can be reproduced by the device) for each discrete window. We will now see why this windowing is applied, and what are the main procedure used.

a. Why dividing the EMG in several time periods?

The EMG is, by nature, a stochastic [46]. This random nature makes the instantaneous value of the record unusable. If instead of using the value of the EMG an instant t we take the record during a period, we can compute properties over this time span and thus overcome this problem.

Some researchers have described the real-time constraint of the prosthesis: the computational delay of the prosthesis should be kept under 300ms, otherwise the user feels the delay and the reproduced motion seems unnatural [47]. To have a response time of the device under this time value, the length of the time-windows must be less than 300ms: having a greater window would violate the real-time requirement, leading to a disturbing user experience.

b. Disjoint windows

The most natural division of the raw signal is surely the one called “disjoint windows” in the literature. With this method, each time window begins when the previous ends, thus there is absolutely no overlapping between the different time divisions. This explanation will make more sense with an example.

Let us choose a window length, for example 200ms, and an EMG recording starting at $t = 0$ s. The first window will be from $t = 0$ s to $t = 200$ ms; the second window will start at $t = 200$ ms and ends at $t = 400$ ms; the third will start at $t = 400$ ms and so on and so forth.

This strategy is clearly the simplest way to divide the time span of our initial full signal. Nevertheless, it is generally suboptimal: indeed, if the required process time, τ , is smaller than the window size, w_{insize} , then there is a time span of $(w_{\text{insize}} - \tau)$ where the computing capacity remains unutilized [47]. A solution to better use the capacity of the computational unit is to use a sliding window scheme.

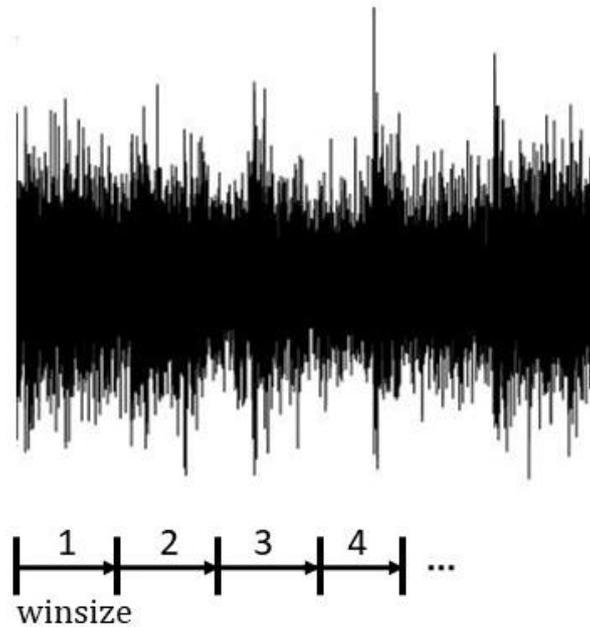


Figure 8-Temporal partition of a signal according to the disjoint windows scheme. The numbers represent the different time spans of each window.

c. Sliding Windows

As we said, the main motivation for the choice of using sliding windows is the optimization of the computational capacity of the device. In this strategy, unlike the previous one, the windows overlap each other, to have a denser stream of data. Doing so, the classification can be optimized to have almost no time without any computation.

Englehart defined the critical parameters of the windowing [47]: the window cannot be slid by less than the process time for the classification of one time unit. For example, if the required classification time of a 200ms window is 20ms, the minimum time for which the window can be slid is 20ms. In that case, the first window will start at $t = 0s$ and end at $t = 200ms$. The second window will start at $t = 20ms$ and end $t = 220ms$. The third time window is $t = 40ms$ and end at $t = 240ms$, and so on. In that example, the stream of data is optimized: a new

process starts as soon as the previous process ends, and the computational unit is always working. The stream is as dense as possible, and the classifier can provide an output every 20ms instead of every 200ms with disjoint windows. Note that the increment of the window is not necessarily equal to the process time and can be between this lower bound (20ms in our example) and the upper bound that is the size of the windows (200ms here).

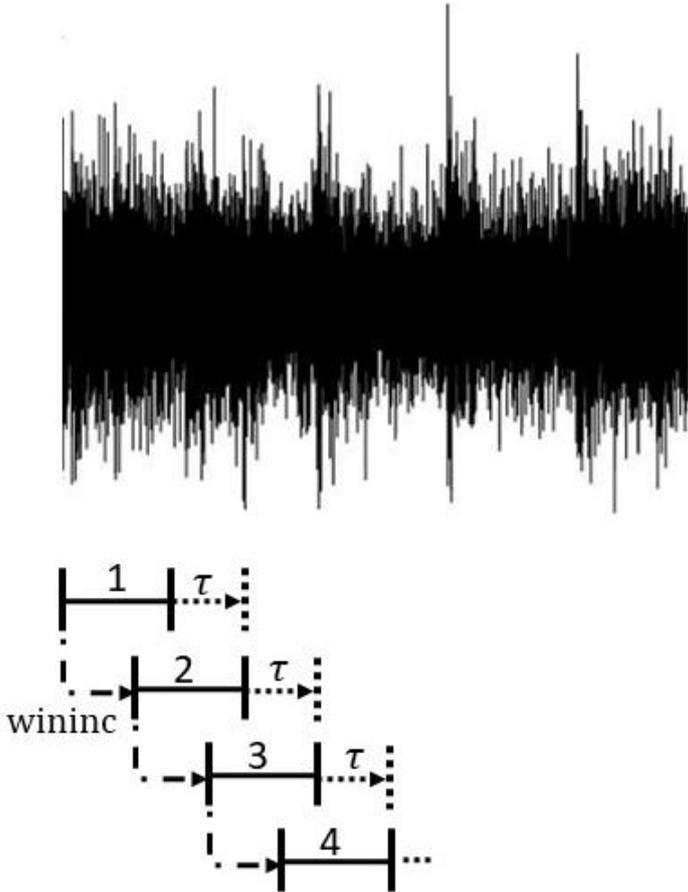


Figure 9-Temporal partition of a signal according to the sliding windows scheme

This method leads to better classification results, denser stream of data, but also increase the computational cost [47] as we could have expected. The parameters can be chosen in the way to get the best compromise between these different aspects.

2 Filtering

The raw recorded EMG is usually noisy and cannot be processed as it is. Many artifacts appear in the spectrum and contaminate the signal. Chowdhury and her colleagues listed the main sources of noise in the EMG [48]. We report them here.

- Inherent noise in the sensor: see *1.3.b-Surface EMG*.
- Movement artifact: when the subject moves, the motion can induce a noise in the recording. De Luca et al. noted two origins to this noise [49]: the first one is when a muscle is contracted, it moves, its length changes, changing the relative placement of the electrode above it and inducing a voltage change; the second source is the movement of the skin that affects the skin-electrode interface. The frequency range of this noise is of the order of 1 – 10Hz.
- Electromagnetic noise: the human body receives electromagnetic radiations and behaves like an antenna. As a result, other signals add to the recording. The most important artifact to add is the interference from the power line, which add an important component at 50Hz (or 60Hz, depending on the power grid frequency).
- Crosstalk: when monitoring the activity of a muscle with a surface electrode, the record can capture an EMG coming from a muscle we did not intend to capture. The wanted EMG is contaminated by the unwanted one, and the first becomes more difficult to interpret. The crosstalk depends on physiological parameters, such as fat layer thickness [50] or muscle shortening [51].

- Internal noise: degradation of the signal due to several factors (anatomical, biochemical, and physiological) that occurs because of the number, the depth, and the location of the active muscle fibres.
- Inherent instability of the signal: because the amplitude of the EMG is quasi-random, the components between 0 and 20Hz are considered unstable (they depend on the firing rate of the muscle unit, which is quasi-random in nature).

To eliminate most of these noises, researchers usually apply two filters. The first one is a notch filter, with a cut-off frequency at 50 or 60Hz (depending on the frequency of the power line). The second is a band-pass filter to remove the part of the signal where the noise prevails. The usual bounds are 20Hz for the lower bound, and between 400 and 500Hz for the upper bound.

Once the data are prepared after these two steps, the classification can begin.

III Features extraction and reduction

The aim of the pattern classification is to assign a class (i.e. the motion to reproduce by the prosthesis, in the case of the prosthetic classification) to each window according to the properties, the attributes, of the signal during this time span. These attributes are called “features”, and they are the input of the classifier, which will assign a class of membership to every window according to the features of the signal during the time window. Researches tend to show that the choice of the features is a predominant parameter for the classification results [52], so this step is crucial.

We will see here what are the features that are traditionally used in the literature, and then we will see the solutions to keep the dimensionality under control.

1 Features extraction

Many features sets have been proposed in the literature. The most used are surely the Time Domain (TD) features since they require rather simple processing with no transformation of the signal. Some researchers also use signal properties from the Frequency Domain (FD) and Time-Frequency Domain, such as Wavelet transforms (WT).

a. Time Domain features

TD features are properties of the signal related to the time-series representing the signal, calculated over the observed window. They are the most used in research papers focusing on myoelectric control, due to their simplicity (they are directly computed from the time-series, without needing transform, unlike the FD features involving a Fourier Transform or the WT features) and because their meanings are more intuitive. Many different features have been used, it is difficult to tell a specific set with features always used, but we will introduce here some of the most recurrent ones [5] [53] [54] [55].

- Zero-Crossing (ZC): this is how many times the zero-amplitude axis is crossed during the time-window.
- Mean Absolute Value (MAV): the mean of the absolute value of the signal.
- Integrate Absolute Value (IAV): the sum of the absolute value of the signal.
- Root Mean Square (RMS): square root of the mean of the signal values to the square.
- Slope Sign Change (SSC): how many times the slope of the signal changes sign (i.e. how many times the derivative of the signal crosses zero).
- Waveform Length (WL): it is the cumulative length of the signal. It also gives a measure of the complexity of the signal [5].
- Willison Amplitude: how many times the change in the signal exceed a predefined threshold. It is an indicator of muscle contraction level.
- Auto-Regressive Model: this states that in a time series, the current output can be interpreted as a linear combination of the previous outputs plus a stochastic term. The coefficients can be estimated for an EMG, and the coefficient can thus be used as classification features.

b. Frequency Domain Features

Unlike the previous set, these features require a transformation of the time-series recorded: the computation of the Power Spectral Density [9] [56] (which is the power distribution over the frequency, i.e. the power of each frequency component). This can be by several numerical technics, such as Welch's Method.

The features that can be computed are:

- Median Frequency: frequency for which half of the power is distributed above and the other is below.
- Mean Frequency: the mean frequency of the power spectrum.

c. Time-Frequency Domain Features

The time-series and its Fourier transform are representations of the signal in respectively the time domain and the frequency domain. There is a one-to-one relation between them. Nevertheless, with them both, we cannot easily observe where in time the spectral components are, which is interesting in a non-stationary signal. The time-frequency representations (TFR) of the signal are a way to overcome this issue. Indeed, while the previous representations were only in "one dimension", the TFRs are in "two dimensions", so we can monitor how the energy of the signal distributes in time and frequency simultaneously.

Three kinds of TFR are usually used in the frame of the prosthesis control: Short-Time Fourier Transform (STFT), the Wavelet Transform (WT) and the Wavelet Packet Transform (WPT) [9] [52] [57]. All these representations are linear discrete TFRs. Indeed, the quadratic TFRs and the continuous TFRs are not

recommended for real-time applications [52] as they are less efficient, and this is a threat to real-time capabilities of the prosthesis.

The main differences between these three is how the time-frequency plane is divided (see Figure 10). For the STFT, the tiling is the same for the entire plane, the fixed aspect-ratio is the same all over the space (thus the time and frequency resolutions are the same everywhere). In the WT, the tiling is variable, and so the resolutions adapt according of where we are in the plane. For example, to study high-frequency components, short time windows are enough, while the windows need to be longer for low-frequency components. The frequency resolution is proportional to the centre frequency, so the aspect-ratio is variable. This variable, yet fixed, tiling was shown to best fits physiological signals. The WPT, on the other hand, has an adaptative tiling: the tiling is set to best fits the aimed application. The tiling is usually set by minimizing the reconstruction error, using a class separability cost function [58], to have the best tiling for classification.

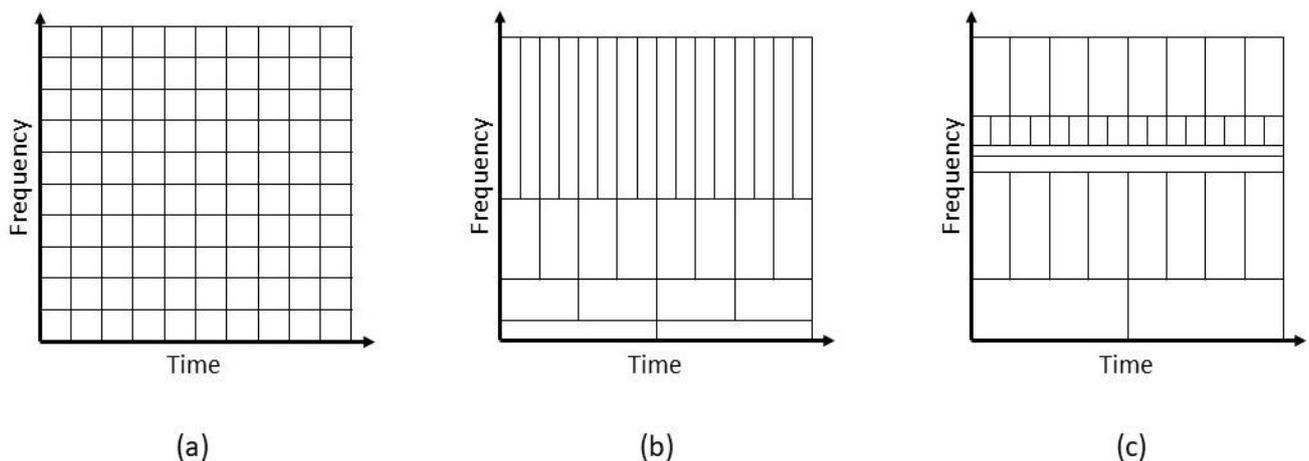


Figure 10-Division of the Time-Frequency space in the different TFR. (a) STFT: the tiling is fixed, with the same aspect ratio everywhere. (b)WT: the tilling is fixed, but the resolution depends on the center frequency. (c)WPT: the tiling adapts to the signal.

Many parameters can be chosen for each transformation: the window type, length and overlap for STFT; the mother wavelet for WT; the mother wavelet and the cost function for WPT. Englehart et al. gave the parameters which seemed to lead to best classification results [52]. We report them in Table 1.

Type of TFR	STFT	WT	WPT
<p>Best parameters according to [52]</p>	<ul style="list-style-type: none"> • Window width: 64ms • Window Overlap: 50% • Window type: Hamming 	<p>Mother Wavelet: Coiflet-4</p>	<ul style="list-style-type: none"> • Mother Wavelet: Symmlet-5 • Cost Function: Euclidean Distance

Table 1-Parameters for Time-Frequency Representations leading to the lowest error rates in the frame of a myoelectric pattern classification problem [52].

2. Reduction of the feature set

In most papers, the feature set obtained after extraction is reduced. It means the classifier does not take as an input the whole set, but only a part and/or a different projection of it. We will see here the motivations of such a step, and then some methods that can be used.

a. Why do we need to reduce the feature set?

One of the first motivations is to avoid the curse of dimensionality. That is a phenomenon that can occur in many numerical domains, especially in Machine Learning. The root of this problem is when we work in a high-dimension space

(i.e. many features are sent to the classifier, a feature being one dimension), the data become sparse in the space, so a huge number of observations is needed to have a reliable result (this number grows exponentially with dimensionality). More data implies a bigger computational cost, and we want to avoid that.

Another possible problem is the overfitting. This is the fact that too many parameters are used to describe the data, leading to extremely specific models to discriminate the different classes, and the classification becomes less robust.

Finally, the original features do not necessarily provide the best separability between the classes and the best clustering of them. The transformation of the set into another equivalent one, like a projection, can overcome this problem.

b. Principal Component Analysis (PCA)

Let us imagine a p -space (a p -feature set for example). The principal components of this space are the p direction vectors that best fits the data, with the i^{th} vector being orthogonal to the $(i - 1)$ previous vectors. We determine the directions that best fits the data by minimizing the Euclidian distance from the points to the directions. The new vectors after this transformation are a combination of the original vector set and is an orthogonal basis, with the maximum variance laying in the first direction, and the minimum variance in the last direction. The new directions are uncorrelated [52]. The user of the PCA can nevertheless decide to only use the k first principal component rather than the whole set, thus reducing the dimensionality. The algorithm to get the new basis is the following.

Let z be the original space, with M observations (columns) of p variables (rows). The first step is to subtract to z the mean of each variable, thus leading to a new vector x [59]:

$$x = z - E[z] \quad (\text{III. 2.1})$$

Then we compute the $p \times p$ covariance matrix of x :

$$C_x = E[xx^T] \quad (\text{III. 2.2})$$

We can then get the principal component of the space thanks to the unit-length eigenvectors $(e_1 \dots e_p)$ of the covariance matrix. The projection matrix W contains the eigen vectors, and $S = Wx$ is the projection of the observations of the original features along the principal components. Note that we can use only the k first rows of S (i.e. use the projection of the original feature set only along the k first principal components), with $1 \leq k \leq p$, for the classification, thus reducing the feature set.

It has been shown by many studies that using the PCA on the original set of features rather than using the raw data improves the classification results [52] [59].

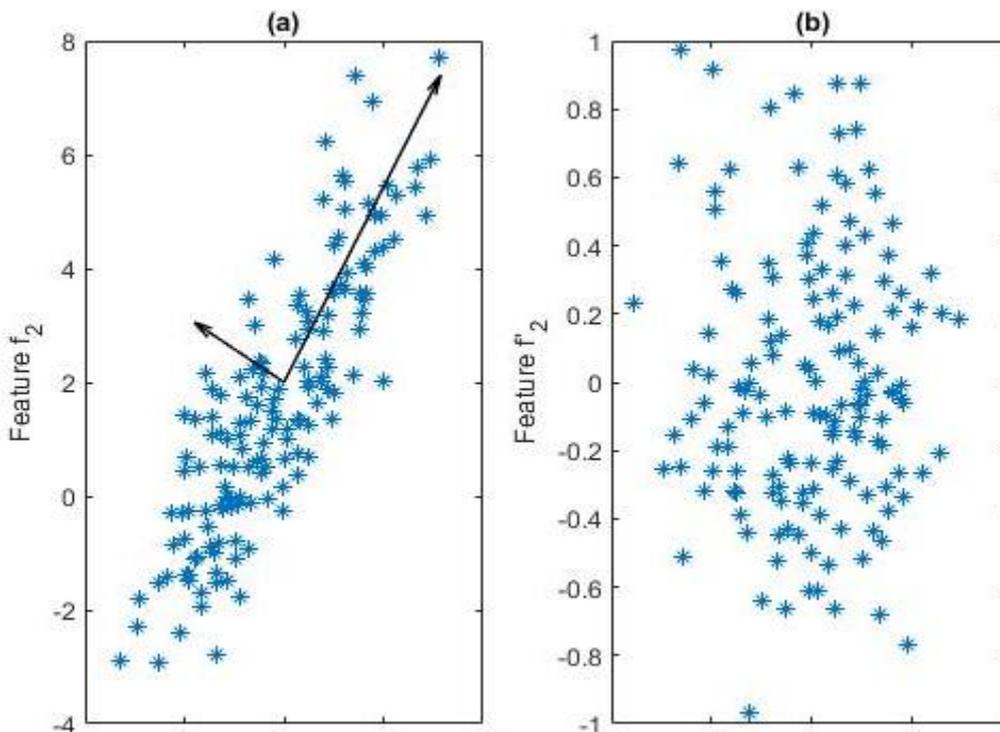


Figure 11-Representation of a PCA in a two-features space. (a) Repartition of the samples in the original feature space. The arrows represent the principal components. (b) Projection of the samples along the principal components.

c. Linear Discriminant Analysis

The Linear Discriminant Analysis (LDA) is a technic to change the original feature space in another one where the different classes are more separated. The main difference with the PCA is that the LDA is a supervised method, i.e. it takes as an input, in addition to the observation of the features, the classes in which fall every observation [60]. This allows to have a projection class-wise, improving the separability of the different patterns. Nevertheless, training values (i.e. observations of the feature space for which the output class is known) are required.

Let us consider a $p \times M$ matrix X , each column being an observation and each row being a feature. The transformation matrix is denoted G , of dimensions $p \times k$, with $k < p$ the number of desired features. This transformation matrix maps each observation x_i in X (each column) to a y_i vector in a k -dimension space. We so get a new feature matrix in $\mathbb{R}^{k \times M}$: $Y = G^T X$. The features in the new space are linear combinations of the features in the original space. The aim of LDA is to get the G matrix, ensuring a good clustering of the different classes, and it is done through the scattering matrixes.

Assume there are N classes in which the data can be classified. The scattering matrixes are defined as follow:

$$\text{Within - class scatter matrix: } S_w = \sum_{i=1}^N P(c_i) E[(X - \mu_i)(X - \mu_i)^T] \quad (\text{III. 2.3})$$

$$\text{Between - class scatter matrix: } S_b = \sum_{i=1}^N P(c_i) (\mu_i - \mu_0)(\mu_i - \mu_0)^T \quad (\text{III. 2.4})$$

With:

- μ_i : the mean of the i^{th} class
- μ_0 : the global mean vector $\mu_0 = \sum_{i=1}^N P(c_i)\mu_i$
- $P(c_i)$: the a priori probability of the class c_i . $P(c_i) \approx \frac{n_i}{N}$, where n_i the number of samples of the class.

The within-class matrix is related to how well the classes are clustered: the trace of it is a measure of the average of the variances of the features in the classes. The between-class matrix on the other hand is related to how the different classes are separated, and its trace is a measure of the average distance of the mean of the classes to the global mean. They both are square matrixes, with as many rows and columns as there are features in the space.

The matrix G must be chosen in order to maximize the separability of the classes, while minimizing the variance inside each class. This can be resumed as maximizing the following function:

$$J = \text{trace}(S_w^{-1}S_b) \quad (\text{III. 2.5})$$

Now, since $Y = G^T X$, we can define other scattering matrixes:

$$S_w^Y = GS_w G^T \quad (\text{III. 2.6})$$

$$S_b^Y = GS_b G^T \quad (\text{III. 2.7})$$

So, we can express J as:

$$J(G) = \text{trace}[(GS_w G^T)^{-1}(GS_b G^T)] \quad (\text{III. 2.8})$$

And G is given by:

$$G = \arg \left[\max_G J(G) \right] \quad (\text{III. 2.9})$$

This optimization problem can be solved by resolving an eigenvalue problem on $S_w^{-1}S_b$, assuming that S_w is non-singular.

The rank of the between-class scattering matrix is bounded by $(N - 1)$, so the reduced space cannot have a higher dimension in LDA.

Some variations of the classical LDA have been proposed to improve the algorithm, the pseudoinverse based LDA to handle singular scatter matrixes [61], or uncorrelated LDA (ULDA) to get uncorrelated features as an output [62], which is the one mostly used in the frame of classification problems and provides better results than when the original raw set is used [55] [62]. Note that the LDA can also be used as a classifier (see *III.2.c-Linear Discriminant Analysis*).

d. Mutual Component Analysis

If the PCA set the information in a more effective way, the main limitation of this technic is that the class labels are not taken into account, and so the re-mapping of the features is not done in order to increase the separability of the possible patterns. A modification has been studied, using the concept of mutual information (MI). This new reduction method, called Mutual Component Analysis (MCA), was introduced by Khushaba [7] and requires training data, such as the LDA.

In information theory, we can quantify the information of a random variable through its entropy. This concept was introduced by Shannon [63]. Let us take a random variable X , its entropy is noted $H(X)$ and is given by:

$$H(X) = -E[\log P(X)] \quad (\text{III. 2.10})$$

From this definition, we can express the conditional entropy of the random variable X given the knowledge of random variable Y :

$$H(X|Y) = -E[\log P(X|Y)] = -P(X, Y)\log P(X|Y) \quad (\text{III. 2.11})$$

And so, we get:

$$H(X|Y) = H(X, Y) - H(Y) \quad (\text{III. 2.12})$$

The MI of two variables measures how mutually dependent they are, i.e. the information they both have in common:

$$I(X, Y) = E \left[\log \frac{P(X, Y)}{P(X)P(Y)} \right] \quad (\text{III. 2.13})$$

In the frame of our problem, the random variables can be replaced by the features. Thus, $I(f_1, f_2)$ represent the information in common brought by the two features f_1 and f_2 , thus how redundant they are. If two features have a huge MI, removing one of them from the feature set would not change much the result of the classification, since most of the information it would have brought to the set is already covered by another feature. So, the point of the MCA is to choose properly the features to reduce the redundancy between them.

Alternatively, we can replace one of the features by the class labels of the training data, getting a new value $I(C, f_1)$. This quantifies the relevance of the feature toward the classification problem: if $I(C, f_1)$ is higher than $I(C, f_2)$, it means that the information carried by the feature f_1 is more relevant toward the class labels (since the values taken by f_1 depends more on the class it falls into than f_2), i.e. it discriminates better the classes. The features should be chosen in order to increase this value.

The algorithm proposed in [7] is thought to fit these two requirements, i.e. it follows “Minimal Redundancy-Maximum Relevance” criterion to select the

features with the most interesting information for the problem under investigation and with the less information in common, followed by a PCA to project along the principal component of the feature space and thus get a new space where the features are uncorrelated. An algorithm of this kind ensures a minimal loss of information. We give the algorithm in the *Appendix*.

The results obtained by this reduction are compared to more traditional ones, such as PCA, Linear Discriminant Analysis (LDA) and Uncorrelated Linear Discriminant Analysis (ULDA). The conclusions are that there is a statistically significant improvement in the classification results when using MCA instead of PCA, while they are slightly worse than the results with LDA and ULDA (without being statistically different with a p-value set to 0.05). Nevertheless, the computation time is way smaller for MCA than for the LDAs, so it seems to be an interesting, yet simple in terms of computation, alternative to the usual methods.

IV. Classifiers

Once we have a proper set of features, with characteristics that can ensure a good separability of the classes, we use them as an input for the classifier.

All classifiers work in two phases. The first one is the training phase. In this one, we use observations for which we know the pattern in which the data fall (so the expected output classes, or the motion the subject wants to do) and the corresponding class labels. This phase is for calibrating the classifier, to “teach” it how the class are separated in the features space. The second phase is the testing one. In this, we send to the trained classifier new data as observations of features (the output patterns are considered unknown). The classifier outputs the class labels of such observations, namely the corresponding pattern according to its calibration.

In this section, we will present some of the most used classifiers in the literature, how they work, their advantages and limitations.

1. Linear Discriminant Analysis

The LDA classifier is based on the Bayes classification rule [64] [65], which states that an observation x of the feature set belongs to a given class w_i among the N possible output classes if:

$$P(w_i|x) > P(w_j|x) \text{ for } j = 1, \dots, N \quad (\text{IV. 1.1})$$

These posterior probabilities can be computed using the Bayes formula, that we recall:

$$P(w_i|x) = \frac{P(w_i)P(x|w_i)}{P(x)} \quad (\text{IV. 1.2})$$

With:

- $P(w_i)$: the prior probability for class i
- $P(x|w_i)$: the class distribution

If we assume all classes are equally probable, i.e. there is an equal number of observations for each class in the training set, $P(w_i)$ is the same for all i and the classification rule can be reduced as:

$$P(x|w_i) > P(x|w_j) \text{ for } j = 1, \dots, N \quad (\text{IV. 1.3})$$

In the LDA classification, we assume that inside each class the distribution of the observations along each feature follows a gaussian distribution. This hypothesis is named the Multivariate Normality. Under this assumption, the probability density functions follow the rule:

$$P(x|w_i) = \frac{1}{\sqrt{(2\pi)^p \det(C)}} \exp\left(-\frac{1}{2}(x - \mu_i)^T C^{-1}(x - \mu_i)\right) \quad (\text{IV. 1.4})$$

With:

- x : the observation of the feature set to classify
- p : the dimension of x (number of features)
- C : the covariance matrix for all classes
- μ_i : the mean value of the class i

The point of the training is to get the parameters μ_i and C from the known values. For the classical LDA, the parameters are fixed: the classifier is called static. This version is simple, but it can encounter difficulties when there are changes in the EMG over time (due to fatigue for example). To compensate the effects of these

changes, some authors proposed a modification of the algorithm to create an adaptative LDA classifier (ALDA).

The difference between the LDA and the ALDA is the update of the parameters with time. In the ALDA, at every step, the L last observations are used to update the values of the model. The classifier is kept up to date by always using the last possible training set in a sense. The ALDA is more robust to noise and to the changes in the EMG (that can be caused by the fatigue, electrode shifting, sweating for example), and globally shows better performances than the classical static LDA [64] [65].

2. K Nearest Neighbours

The k -nearest neighbours (kNN) classifier (k being an integer) is probably the most intuitive and the simplest classifier. It is based on the idea that a new point in the features space is likely to belong to the same class as the surrounding points.

As training values, we send to the classifier L observations of the feature space (where a distance is defined, Euclidean distance for example) with their corresponding L output classes. When a new observation of the feature set arrives to be classified, the distance of this new value to each of the L training values is computed, and then these distances are ranked from the nearest to the furthest. The output value of the new observation is the class that appears the most among the classes of the k observations for which the distances are the smallest [66].

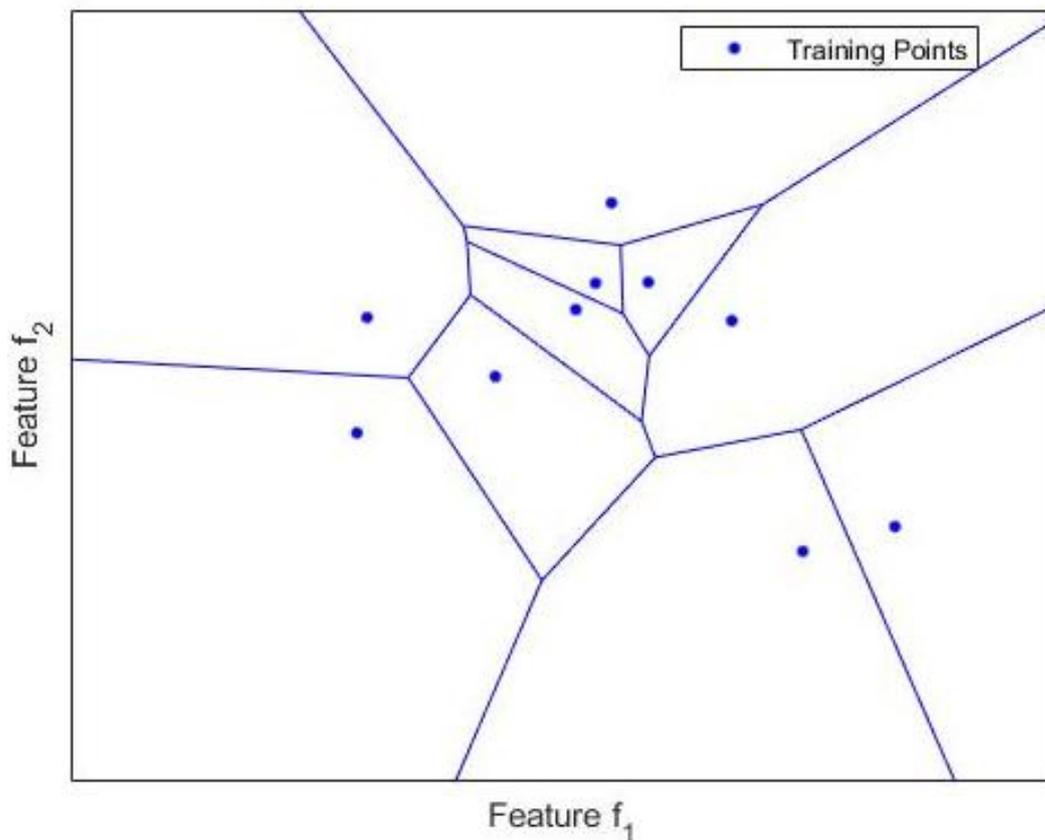


Figure 12-Partition of the space with 10 training points for a k NN ($k=1$), known as Voronoi graph. The lines represent the “frontier” where the closest training point changes.

The parameter k has to be chosen. A large k value, thus when more neighbours are taken into account, reduces the noise on the classification but, on the other hand, makes the separation between the different classes less clear (the classes become less clustered). An optimal value can be sought using some optimization technics [67].

An improvement of the method by weighting the distances to the neighbours has been proposed [68]. The idea is that if a neighbour is close to the observation to classify, it should matter more than the furthest training value. So, we give a weight to each of the k nearest neighbours, to give more importance to the closest ones. This method is more robust to noise.

Note that the kNN is high computational technic since L distances are computed for each observation to classify.

3. Support Vector Machine

The Support Machine Vector (SVM) is a classifier based on the research of a hyperplane that best separates two classes in the feature space. Thus, the observation in this space on one side of the hyperplane will be assigned to one class, and the observations on the other side will be assigned to the other class. The construction of the classifier was described by Cortes and Vapnik [69].

Let L training observations of the feature space, x_i , and the corresponding classes, y_i . So, the training set is L duets (x_i, y_i) . The class y_i is either +1 or -1 depending on the class.

A hyperplane is described as:

$$w^T x + w_0 = 0 \quad (\text{IV. 3.1})$$

With:

- w : normal vector to the plane
- w_0 : the position of the plane

The hyperplane that separates the two classes is not unique, the algorithm selects the one that offers the maximum margin (see Figure 13). It means that the chosen hyperplane will be the furthest possible to the closest training points, to minimise the misclassifications.

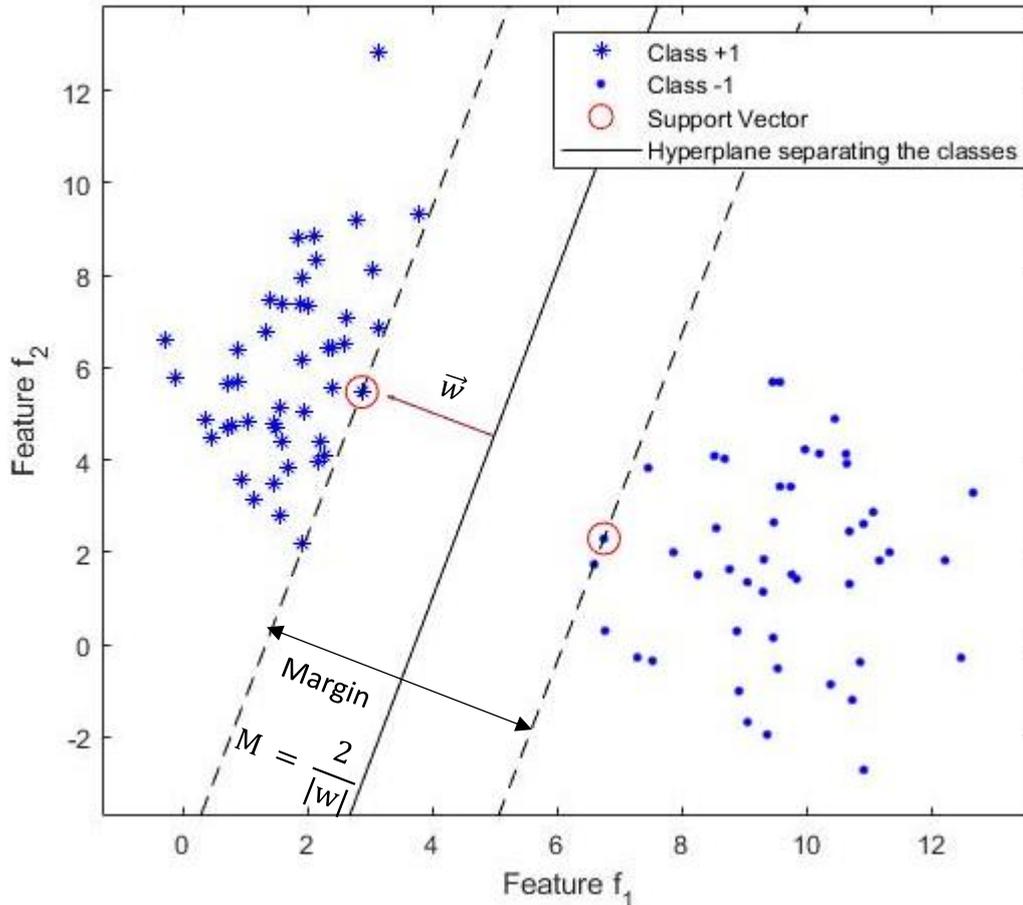


Figure 13-Support Machine Vector for a two-classes problem in a two-features space. The stars are the training points for the class "+1", while the dots are the training points for the class "-1". The support vector are circled. The normal vector w is shown before scaling. The continuous line is the hyperplane separating the classes, and the dashed lines are shown for the margin.

The classes are said linearly separable if there are a vector w and a scalar w_0 such as:

$$y_i(w^T x_i + w_0) \geq 1 \text{ for } i = 1, \dots, L$$

In that case, w and w_0 are rescaled to get a unitary distance for the closest training point for each class. These points are noted x^+ and x^- , belonging respectively to the class "+1" and "-1".

$$\begin{cases} w^T x^+ + w_0 = +1 \\ w^T x^- + w_0 = -1 \end{cases} \quad (\text{VI. 3.2})$$

Let the optimal hyperplane, that separates the classes with the maximal margin. The direction of w is the one where the distance between the projections of the vectors of the training set is the highest (since it is the hyperplane with the maximal margin). We defined the margin as:

$$M = \frac{w^T}{|w|} \cdot (x^+ - x^-) \quad (\text{IV. 3.3})$$

The margin is simply the minimum distance between the projections a vector from the class “ + 1” and a vector from the class “ – 1” along the direction of w . By the equation (IV.3.2), we obtain:

$$M = \frac{2}{|w|} \quad (\text{IV. 3.4})$$

And the research of the optimal hyperplane is reduced to the optimization problem:

$$\begin{aligned} &\text{minimize } J(w, w_0) = \frac{1}{2} \|w\|^2 \\ &\text{under the constraint: } y_i(w^T x_i + w_0) \geq 1 \text{ for } i = 1, \dots, L \end{aligned}$$

The vectors for which $y_i(w \cdot x_i + w_0) = 1$ are called support vectors. It has been shown the vector w_0 , that gives the position of the optimal hyperplane, can be written as a linear combination of support vectors [69].

This linear classifier applies only in the case of separable classes, so no sample “crosses” the separation of the hyperplane and falls in the “wrong part” of the space. Nevertheless, it may happen that data do not validate this condition. In that case, modifications must be done in the algorithm. In particular we introduce slack variables ξ [69], such as:

$$y_i(w^T x_i + w_0) \geq 1 - \xi_i \text{ for } i = 1, \dots, L$$

The value of ξ_i , depends on either if the pattern is on the correct side of the plane: $\xi_i = 0$ if it is outside the margin and on the right side, $\xi_i > 0$ if inside the margin or on the wrong side. The point of the algorithm in this case is to find the margin to have the sum of the ξ_i as small as possible. Thus, the optimization problem become:

$$\text{minimize } J(w, w_0, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^p \xi_i$$

under the constraint: $y_i(w^T x_i + w_0) \geq 1 - \xi_i$, for $i = 1, \dots, L$ and $\xi_i > 0$

For both cases, Lagrangian functions of the problem can be introduced, and the Karush-Kuhn-Tucker (KKT) conditions can be applied. Note that the classifier remains linear after the addition of slack variables. Nevertheless, if this not enough for the considered problem, one can use the kernel method to remap the data in the new space where they will be separable [70] [71]: this is the non-linear SVM.

The SVM is defined only for a two-classes problem. Nevertheless, it can be generalized to a multiclass problem by calculating several classifiers. Two procedures can be followed to extend to $N > 2$ output patterns:

- One-versus-all. In this case, we consider each class individually and all the other classes are considered a single class. A SVM is computed to separate the class and the aggregation of all the other classes. So, in the end, we get N classifiers. When an observation to be classified arrives, the N classifications are done and the one with the highest output function assigns the output class (winner takes all).
- One-versus-one. In that case, the classes are considered two-by-two and a classifier is computed for every pair of classes. $N(N - 1)/2$ binary

classifiers are computed. The new observations are classified by each classifier, and the class with the maximum number of wins is designed as the output class (max-wins voting strategy).

4. Artificial Neural Networks

The Artificial Neural Networks (ANN) are classifiers for which the structure is inspired by the organization of actual brain cells, especially on how each neuron interacts with the others.

These networks are organized in 3 or more layers (an input layer, one or more hidden layers and an output layer) of “neurons”, which are the nodes of the network. Each neuron of a given layer is linked to all neurons of the previous and following layers (if they exist), the connections modelling the synapses. The job of these synapses is to link the output of the neurons on the $(n - 1)^{\text{th}}$ layer to the input of the neurons on the n^{th} layer.

The neurons are the computational units of the network. When a neuron gets the information (i.e. the signals) from the previous neural layer, it processes the output (sent to the next layer) from a linear combination of inputs. For the i^{th} neuron of a given layer, the output is given by:

$$y_i = \varphi_i \left(\sum_{j=1}^k (\omega_{ij}x_j) + b_i \right) \quad (\text{IV. 4.1})$$

With:

- y_i : output of the i^{th} neuron of the given layer

- x_j : input coming from the j^{th} neuron of the previous layer
- ω_{ij} : weight of the input x_j for the i^{th} neuron
- b_i : a bias
- $\varphi_i(\cdot)$: activation function of the i^{th} neuron

This weighted sum plus a bias is taken as the argument of the activation function, that will state the information the neuron will send to the following layer. The most used activation function is the sigmoid:

$$\varphi(x) = \frac{1}{1 + \exp(-ax)} \quad (\text{IV. 4.2})$$

With:

- a : the slope parameter

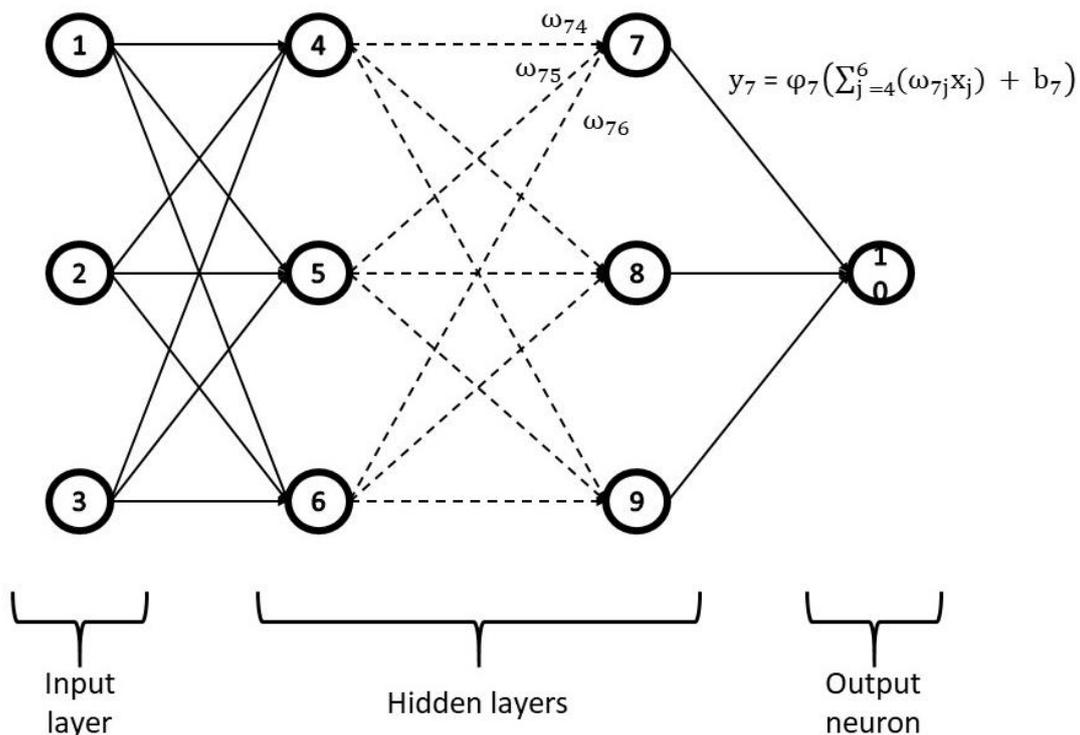


Figure 14-Representation of a simple Artificial Neural Network, with a simple forward topology and two hidden layers.

So according to the inputs of the neuron, a value between 0 and 1 is outputted. There are in the input layer one neuron by feature used. The value sent to the

second layer is the observation of that feature during the time window. The output of the neuron of the output layer finally gives the class assigned to the observation of the features.

This is the basic scheme of the neural networks. Nevertheless, the connection topology can change, with synapses that could skip a layer, return to a previous layer, or be connected to the same layer.

The training of the net in supervised learning involves tuning the weights, the bias, and the slope parameters of the neurons in order to best fit to the expected output. Unsupervised learning is also possible [72], but we will not focus on it here. The optimization of the net is done by minimizing a cost function, which measures the error between the computed output and the expected one for the training set. The usual training of the ANN is based on the gradient of the cost function, and the update of the parameters is done in the direction of it. The most popular algorithms are the back propagation [73] and the Levenberg-Marquardt algorithm [74] [75].

The back-propagation algorithm is called like this because the error is propagated from the output to the nodes inside the hidden layers. *The first step* is setting the weights to initial small random value.

Then, the *forward computation*: for every training observation, every argument in the net and the output are computed with the current parameters. The cost function related to the parameters under investigation is calculated.

Backward computation: the dependency of the cost function toward the parameters is computed.

Update: the weights and biases are updated to reduce the cost (in the direction of the gradient). We go back to the forward computation and we go on until the error reaches a minimum.

The Levenberg-Marquardt algorithm is designed to solve non-linear least-square problems. The sum of square errors is considered:

$$E(\omega) = \sum_{i=1}^L (d_i - y(x_i, \omega))^2 \quad (\text{IV. 4.3})$$

With:

- d_i : expected class of the i^{th} observation
- x_i : i^{th} observation of the feature space
- ω : weights of the network
- $y(x_i, \omega)$: classification of the i^{th} observation of the feature space given the weights of the network

At each iteration step, the weights are updated as follows: $\omega \rightarrow \omega + \vec{\delta}$. To estimate the update vector, $\vec{\delta}$, we consider the linearization of the output of the network:

$$y(x_i, \omega + \vec{\delta}) \approx y(x_i, \omega) + J_i \vec{\delta} \quad (\text{IV. 4.4})$$

With:

- J_i : the gradient of the y : $J_i = \frac{\partial y(x_i, \omega)}{\partial \omega}$

So, the error function can be approximated by:

$$E(\omega + \vec{\delta}) \approx \sum_{i=1}^L (d_i - y(x_i, \omega) - J_i \vec{\delta})^2 = \|\vec{d} - \vec{y}(\vec{x}, \omega) - J_i \vec{\delta}\|^2 \quad (\text{IV. 4.5})$$

With:

- \vec{d} : a vector containing the values d_i
- $\vec{y}(\vec{x}, \omega)$: a vector containing the values $y(x_i, \omega)$

We choose the value of the update vector with the pseudoinverse, $J^\#$, of the matrix J . We get the optimal value of $\overrightarrow{\delta_{opt}}$ to minimize the error function (derivative set to 0):

$$\overrightarrow{\delta_{opt}} = J^\# (\vec{d} - \vec{y}(\vec{x}, \omega)) \quad (\text{IV. 4.6})$$

With:

- $J^\# = (J^H J)^{-1} J^H = (J^T J)^{-1} J^T$ in the case of a real matrix

This update is called the Gauss-Newton algorithm. In Levenberg's algorithm, on the other hand, a damping term λ is added:

$$(J^T J + \lambda I) \overrightarrow{\delta_{opt}} = J^T (\vec{d} - \vec{y}(\vec{x}, \omega)) \quad (\text{IV. 4.7})$$

This term is updated at every iteration and is used to control the speed of convergence. If the convergence is fast, λ is reduced, and the algorithm gets closer to the Gauss-Newton result. On the other hand, if the reduction of the error is slow, λ is increased to follow the gradient-descent direction a step further. The problem of this feature is when λ is large: in this case, the approximation of the Hessian matrix $J^T J$ is not used, and the convergence is slow in direction of small gradients. To overcome this problem, Marquardt modified the algorithm: the idea is scaling gradient's components according to the curvature, so the movements direction of small gradients become larger. He replaced the identity by $\text{diag}(J^T J)$ [75]:

$$(J^T J + \lambda \text{diag}(J^T J)) \overrightarrow{\delta_{opt}} = J^T (\vec{d} - \vec{y}(\vec{x}, \omega)) \quad (\text{IV. 4.8})$$

5. Classification trees

Decision trees are predictive models, in which the classes are the leaves, and the branches are splits based on a certain threshold of a given feature. After a split, the algorithm goes to another node according to the evaluation of the feature. For example, let us take a node i for which the split is based on the MAV. If the MAV is smaller than a given α , the next step in the algorithm will be the node j which is linked to node i by the transition $MAV < \alpha$. Each observation begins at the top of the tree, passes every split (thus classes are rejected sequentially), to finally arrive to a leaf which is the assigned output class. A schematic of a classification tree is displayed on Figure 15. The main issue is to decide which features should be used on the splits, in which orders and what are the corresponding thresholds. These decisions are usually taken adopting a top-down approach: at each step, we choose the feature that best splits the set.

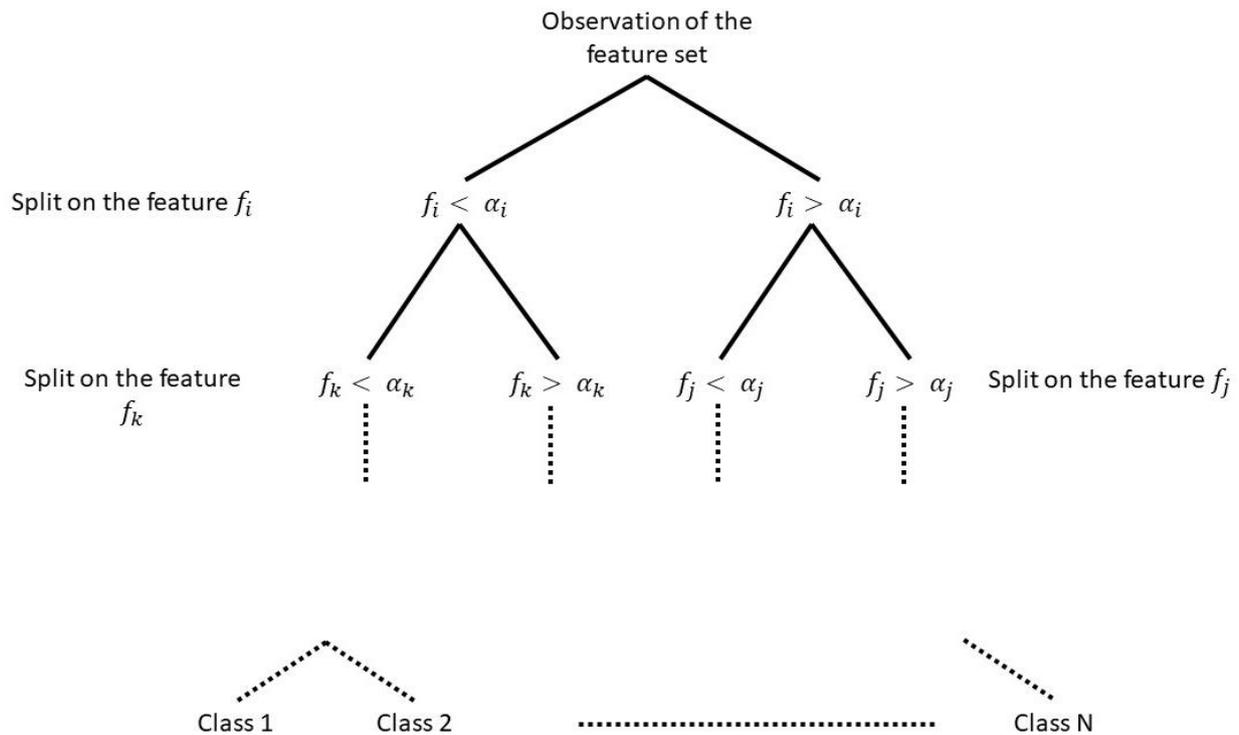


Figure 15-Representation of an exemple of a Classification Tree. Each split is based on the value of a feature, and the leaves are the target classes.

Several strategies to construct the tree have been proposed, we will review some of them here.

The **Iterative Dichotomiser 3** (ID3) is an algorithm that uses the entropies of the features to construct the splits. It has been introduced by Quinlan [76].

The entropy of a feature was given in the *Mutual Component Analysis* part of this work. As we stated there, the entropy is a notion linked to the information gain: the lower the entropy, the higher the information brought by the feature. Using this value allows to state which features are the most interesting in the frame of the classification, and thus we can say which ones most discriminate the data set. This is computed by the information gain, given by:

$$IG(S, f) = H(S) - \sum_{t \in T} p(t)H(t) = H(S) - H(S|f) \quad (IV. 5.1)$$

With:

- f : feature on which the split is done
- S : feature set
- T : space of the subsets t created by the splitting of S on f . $S = \bigcup_{t \in T} t$
- $p(t)$: proportion of the number of elements in t to the number of elements in S .

The information gain is basically a measure of how the uncertainty is reduced after a split on the feature f .

The ID3 algorithm is:

1. Compute the entropy and information gain after a split on every feature of the data set.
2. Split the data set by using the feature for which the entropy will be minimized after splitting (the threshold is chosen to have an optimally divided set).
3. Create one node of the decision tree with this given test on this given feature.
4. Iterate the algorithm on each subset with the unused features.

The **Classification And Regression Tree (CART)** [77] is a decision tree in which the splits are computed by the mean of a cost function, the chosen splits being the ones minimizing this function. For each node, all different features and splits are tested, and the one for which the cost is the lowest is chosen. Usually, the Gini Index is chosen as the cost function for classification problems. This index is a measure of the “impurity” of a given value. In the frame of a classification problem, it states the disparity of the classes after a split. If the Gini coefficient

is 0, it means all the data belong to a single class, and so this split discriminates perfectly. It is given by:

$$I_G(p) = \sum_{i=1}^N p_i \sum_{j \neq i}^N p_j = \sum_{i=1}^N p_i (1 - p_i) = \sum_{i=1}^N p_i - \sum_{i=1}^N p_i^2 = 1 - \sum_{i=1}^N p_i^2 \quad (\text{IV.5.2})$$

With:

- N: number of classes
- p_i : proportion of samples in the i^{th} class

To improve the accuracy and to decrease the complexity of the algorithm, a modification has been proposed: creating several trees, and aggregate their results to get the final class. The classifier is called Random Forest [78] [79]. For the j^{th} tree, only a part of the feature set is used (usually, for p features in the original set, only \sqrt{p} features are used for each tree). This subset of features is chosen randomly. This is a solution to the dimensionality curse, the trees are simple and stable, and leads to good classification results.

6. Gaussian Mixture Model

In statistics, a mixture model is a method to represent a population containing several subpopulations. Starting from it, we can give for every observation, every element of the population, the probability that it belongs to each subpopulation. If we assimilate every possible output class to a subpopulation, we can clearly see how interesting such models can be of interest to classify the EMG. The Gaussian Mixture Model (GMM) is one of these representations.

Let a N class classification problem, and a p -dimensional feature set. X is an observation of the feature space to be classified. The GMM of this problem, if we suppose a k -order model, is for each class n ($n = 1, \dots, N$), with p features:

$$\lambda_n = (w_i^n, \vec{\mu}_i^n, C_i^n)_{i=1, \dots, k} \quad (\text{IV. 6.1})$$

With:

- w_i^n : weights. They are under the constraints: $\sum_{i=1}^k w_i^n = 1$ and $w_i^n > 0$
- $\vec{\mu}_i^n$: mean vectors, of dimension $p \times 1$
- C_i^n : covariance matrices, of dimension $p \times p$

These parameters of the model can be computed using an estimation-maximization algorithm [80].

The mixture density x for the n^{th} class, i.e. the likelihood of the observation x to belong to the n^{th} class, is given by:

$$P(\vec{x}|\lambda_n) = \sum_{i=1}^k w_i^n \rho_i^n(\vec{x}) \quad (\text{IV. 6.2})$$

With:

- $\rho_i^n(\vec{x})$: Gaussian density functions of the n^{th} class. It is defined by:

$$\rho_i^n(\vec{x}) = \frac{1}{(2\pi)^{p/2} |C_i^n|^{1/2}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu}_i^n)'(C_i^n)^{-1}(\vec{x} - \vec{\mu}_i^n)\right)$$
, p being the dimension of the feature space

Once all the probabilities are estimated for an arriving observation (the likelihoods that this observation belongs to each class), the pattern with the maximum probability is selected as the output of the classifier.

This algorithm still has some issue. Firstly, the order of the model is crucial. Selecting too few mixture components can leads to a model that does not

describe accurately the space while when selecting too many, the model cannot be reliably trained by the training set and thus leads to reduced classification performances (plus an increase of the required computation). So an optimal value has to be found. Huang [81] found that each subject (so each problem) has a given optimal order, for which the performances are the best (better than a “universal” value of this order).

Secondly, during the training of the GMM, it is usual that the variances become small (especially for high orders). These low variances produce singularities and reduce the classification results. The solution to avoid this problem is to set a Variance Limiting (VL) constraint [82]: the lower bound of the covariance matrixes σ_{min}^2 is chosen. This value should not be chosen too high, otherwise there is a risk that all elements are the same, drastically changing the performances of the model; and it should not be set too low neither, as it could not resolve the initial problem. The value of σ_{min}^2 is usually set empirically.

Finally, the form of the covariance matrixes is important. The matrixes can either be full or diagonal. The diagonal ones seem to outperform the full ones, and the computations required are less intensive [83].

V. Post-processing

The final step of the classification is to apply a post-processing to smoothen the results, thus eliminating the possible spurious assigned classes. The main method is the majority voter (MV), but other technics, exploiting a Bayesian approach, are proposed.

1. Majority Voter

The majority voter takes as an input the output of the classifier, which is the assigned class for each time window. The aim of the MV is to smoothen the decisions of the classifier by averaging it over a certain number of windows. This is based on the idea that a different result occurring in the middle of a rather homogenous stream of output is likely to be misclassified, and so the MV corrects it. The parameter of the MV is the number of windows to consider. For example, for the n^{th} window with 5 decisions, the majority voter takes the five decisions between the $(n - 2)^{\text{th}}$ and the $(n + 2)^{\text{th}}$, and the output class for the n^{th} window is the class that appears the most in this span. Doing so, the MV eliminates spurious decisions. As a concrete example, if the flow of classification results, i.e. the output of the classifier, is:

Window	$(n - 2)^{\text{th}}$	$(n - 1)^{\text{th}}$	n^{th}	$(n + 1)^{\text{th}}$	$(n + 2)^{\text{th}}$
Assigned class	1	1	2	1	3

The classifier first classified the n^{th} window as part of the second motion class. But, if we apply a MV with 5 decisions, this result is transformed. Indeed, as we can see, there three occurrences of the class “1” in the decisions we consider, so the MV will output the n^{th} window as part of the first motion class. The spurious decision has been eliminated; the results are smoother.

The requirements for a real-time application are explained in [47]: basically, the controller delay must be kept under 300ms, and the optimal delay for the patient is in the 100 – 125ms range [84]. So, the number of decisions we can include in the MV will depends on the delay we want for the device. Let us state that $2m + 1$ decisions are included in the majority voter (the decision of the window under observation, m decisions before, m decisions after), that wininc is the window increment (so the increment must be between the computation time for the classification of one window and the length of the windows, see *II.1.c-Sliding Windows*), and T_d is the delay. Englehart and Hudgins stated the inequality:

$$m * \text{wininc} < T_d$$

The parameters wininc depends on the length of the window and on the processing we apply to it [47]. Generally, the greater the number m , the lower the error rate, but it increases the total delay, making it possibly suboptimal. A compromise must be found between the classification results and the delay, especially by changing the window length.

2. Bayesian Fusion

The MV has shown its effectiveness despite its simplicity. Nevertheless, researchers as Khushaba address criticism to this method because it approaches the output of the classifier “in a naïve manner without consider the actual probabilities of misclassification” [8]. A new approach is so described to overcome this problem, based on a Bayesian fusion.

The Bayesian fusion assumes the statistical independence of the results we combine. Khushaba uses a disjoint windows scheme to fulfil this requirement. Indeed, the EMG is by nature random, and taking two disjoint windows (thus no overlapping), the samples can be considered weakly correlated. Nevertheless, some researchers seem to show that the Bayesian fusion can work even if the statistical independence is not respected [85].

Let a n-classes problem be. We will denote the class set C_i , with $i = 1, 2, \dots, n$. We will denote the k^{th} time window w_k . The probability the data in the k^{th} window belongs to the i^{th} class is denoted $p(C_i|w_k)$. We consider the n classes the only possible output, so the sum of conditional probabilities is 1: $\sum_{i=1}^n p(C_i|w_k) = 1$.

After the classification of the first window, we express the probability that the data belongs to a given class C_i by $p(C_i|w_1)$. After the classification of the second window, we express the probability as $p(C_i|w_2)$ and so on so forth. We can also express another probability using the Bayes rule: the probability the class is C_i given the data of the two first windows:

$$p(C_i|w_1, w_2) = \frac{p(w_1|C_i, w_2)p(C_i|w_2)}{p(w_1|w_2)} \quad (\text{V. 2.1})$$

With the data from the different windows independent, we can simplify some expression as following:

$$p(w_1|C_i, w_2) = p(w_1|C_i) \quad (V. 2.2)$$

$$p(w_1|w_2) = p(w_1) \quad (V. 2.3)$$

Reducing the Bayes expression to:

$$p(C_i|w_1, w_2) = \frac{p(w_1)p(C_i|w_2)}{p(w_1)} \quad (V. 2.4)$$

From the expansion:

$$p(w_1|C_i) = \frac{p(C_i|w_1)p(w_1)}{p(C_i)} \quad (V. 2.5)$$

we finally get:

$$p(C_i|w_1, w_2) = \frac{p(C_i|w_1)p(C_i|w_2)}{p(C_i)} \quad (V. 2.6)$$

We notice the result is simply equal to the product of the conditional probabilities of the given class C_i for all time window, divided by a term to normalize the sum of probabilities for all classes. If we generalize for M time-windows:

$$p(C_i|w_1, w_2, \dots, w_M) = \Delta \prod_{m=1}^M p(C_i|w_m) \quad (V. 2.7)$$

With Δ the normalization term to have a proper and valid probability density function. The class for which the probability is the highest when the M time-windows are taken into account is considered the best classification.

An improvement can however still be done. Indeed, if one of the estimated probabilities is zero, the probability $p(C_i|w_1, w_2, \dots, w_M)$ becomes null. Moreover, the L previous probabilities, still stored in the memory, weight as

much as the probabilities of the current window, which can cause delay in the transition between two classes: indeed, for the first time windows after a change of motion, the most ancient probabilities designate the previous motion as more probable as long as they are considered in the post-processing. These problems can be resolved by adding a weighting factor k_j as:

$$p(C_i|w_1, w_2, \dots, w_M) = \Delta \prod_{m=1}^M [p(C_i|w_m) + k_j] \quad (\text{V. 2.8})$$

With:

- j : the position in the queue. $j = 1, 2, \dots, L + 1$, with $j = 1$ for the last value inserted in the queue and increasing as it moves to older value.

This factor must decrease as the concerned window becomes “further away” to the current window. The higher weights must be assigned to the more recent windows, to reduce the bias toward the previous results. We express it as:

$$k_j = 10 \frac{\exp(-0.5 \times j/(L + 1))}{\sum_{l=1}^{L+1} \exp(-0.5 \times l/(L + 1))} \quad (\text{V. 2.9})$$

VI. Beyond EMG classification: using deconvolution as a new technic.

Surface EMG has been widely used as the signal in the classification research. The main axes of improvement for the classification are usually increasing the number of detection channels [57], finding new technics for one stage instead of using more “traditional” ones [86], or optimizing one or several parameters [87]. However, increasing the number of surface electrodes could make the prosthesis expensive, cumbersome and uncomfortable. Indeed, as the required performances of the prosthesis are constantly rising, many works aim at using only few detection channels [88]. Moreover, further optimizing the processing of the EMG could be prone to overfitting or to the enlargement of the processing time, which should be kept low to allow real time control and to be accepted by users. Indeed, the processing delay should be kept under 300ms to ensure a good user experience [47], with an optimal delay between 100 and 125ms [84]. Thus, an alternative is to get better information from the muscles.

We will here give an overview of the other signals used in the literature, and then we will explore the possibility of using a deconvolution of the EMG.

1. Other signals used in the literature.

The first signal that comes to mind is the intramuscular EMG (see *1.3.a-Intramuscular EMG*) for its high selectivity. But as mentioned earlier, it is not used due to the invasiveness of the technic.

It has also been proposed to use a different measure from EMG, such as the force myography (FMG). FMG is a measure of the surface pressure that occurs during a muscle contraction. The FMG can so be recorded by force sensors, and it can be used in pattern classification [89] or regression control [90] that continuously maps the position of the artificial limb under control. The advantage of the FMG is the natural mechanical filtering, which makes the necessary signal conditioning less complex than for EMG, due to its stochastic nature. Therefore, the output of the force sensor can directly be used for classification, unlike the raw EMG [90].

Nevertheless, these alternatives usually require a different hardware that is still less widespread than surface EMG amplifiers, which are instead available in most labs studying prosthetic control, since it is the most widely used signal. Thus, an innovative pre-processing of surface EMG to extract the firing pattern of the MU involved in the motion could provide a step forward in the field of myoelectric control. This would keep the benefits of surface EMG technology (which is non-invasive and widespread), possibly providing a better input (e.g., in terms of selectivity of information, stability to noise or crosstalk) to the classification machine.

We will here give a brief overview of the deconvolution of the EMG and the methods used in the literature to obtain it and to use it in a control frame. Then, we will prove the possibility to control several patterns using a simpler deconvolution technic (only one channel, in real-time) than the “more traditional” ones.

2. Deconvolution signal

a. What is it?

The surface EMG that we record on the surface of the skin of the patient can be modelled as a convolutive mixture of the muscle units' discharges involved in the movement. Indeed, the movement is created by the activation of muscle units, i.e. the propagation of action potentials along the muscle fibres, which is then filtered by the conduction volume before arriving to the electrode (the convolution being the mathematical representation of filtering in the time domain) [91]. For each MU involved in the motion, a contribution appears in the surface EMG: the firing pattern of the MU, convoluted with a kernel.

We recall that a convolution is a mathematical operation that associates to a couple of functions, f and g , a third function, $f * g$, called the convolution of f and g . Mathematically, it is given by:

$$(f * g)(t) = \int_{\tau = -\infty}^{\tau = +\infty} f(t - \tau)g(\tau)d\tau = \int_{\tau = -\infty}^{\tau = +\infty} f(\tau)g(t - \tau)d\tau \quad (\text{VI. 2.1})$$

Visually, it can be seen as how the function f overlaps with a “sliding” function g . The convolution generalizes the idea of moving average.

So, the EMG is related to these firing patterns, but it is not a direct measure of it. Several methods have been established to access to this signal starting from the recorded EMG and are reviewed in [92]. One of the most interesting methods is certainly the Blind Source Separation (BSS) based Convolution Kernel

Compensation [93], which is robust and offers good estimation of the firing rate of the MU [92].

b. Uses of MU's discharges in the frame of a classification problem.

A new information, which gives a more precise insight of the muscle activity than the classical EMG, is available. Specifically, the timings of MU recruitment and discharges are related to muscle force [94] and velocity [95]. Thus, researchers have tried to use this new signal for prosthetic purposes.

Farina et al. [96] used a BSS to get the MU's discharge timing from a high-density recording device (more than 50 channels) EMG, after muscle reinnervation. They then resolved a pattern classification problem, with either 7, 9 or 11 motions to classify. On one hand, the classical EMG classification was done (extracting TD features), and on the other hand the neuron discharge timings were used to classify the motions. The use of the decomposition into MU's firing patterns outperformed the classical classification technics (average rate of classification superior to 97%, against 85%).

The decomposition is also used to predict wrist [97] or fingers [98] kinematics (regression control), i.e. get an estimation of the joint angle of the given degree of freedom. The correlation between the predicted and the actual angles allow authors of respective papers to assess that this kind of prosthetic control is promising.

Nevertheless, in this brief overview of the possible control procedures using the MU discharge timings, a high number of sensors is needed (usually more than

50), and so the needed computations can be huge, which are drawbacks from the point of view of a real prosthesis.

c. Single channel deconvolution

While most estimations of the firing patterns technics require either computationally intensive methods or dense sensor arrays, Luca Mesin proposed to compute it with a single channel, called the Single Channel Deconvolution [99].

A Single Differential EMG can be interpreted as a noisy convolution of the firing rate of the MU with a kernel. This kernel is the first derivative of a Gaussian function, with a standard deviation set to optimally fit the Power Spectral Density. This function models the average MU Action Potential (MUAP) and can be rescaled in amplitude. The model is:

$$s(t) = K(t) * f(t) + n(t) \quad (\text{VI. 2.2})$$

With:

- $s(t)$: the EMG
- $K(t)$: the kernel
- $f(t)$: the firing pattern of the involved MU
- $n(t)$: the noise

Note that in the model above, all errors (approximations in the convolution due to noise, differences between the MUAP shape and the chosen kernel, ...) are aggregated in the noise term.

The aim of the deconvolution is to get the signal f . To do so, we must resolve an unstable inverse problem. This problem must be regularized with the Tykhonov approach [100]:

$$\operatorname{argmin}_{\hat{f}(t)} = \|s(t) - K(t) * \hat{f}(t)\|_2^2 + \alpha \|\hat{f}(t)\|_2^2 \quad (\text{VI. 2.3})$$

With:

- α : a term set to get a compromise between an optimal fit of the data and the stability of the solution.
- $\|\cdot\|_2$: the L_2 -norm in the space.

To fit the discrete signals (digital records of the sEMG), we discretize the problem [99]. The deconvolution operation becomes the multiplication with a matrix A , which contains delayed versions of the kernel in its columns: the kernel is discretized according to the sampling frequency and delayed by multiple of the sampling time. α is set to 1% of the maximum eigenvalue of $A^T A$. We must take care of the computational cost: indeed, the number of elements of $A^T A$ increases as the square of the number of samples in the time frame (the i^{th} column of A is a version of the kernel delayed of $i * t_s$, or i samples, so there are, for a signal of T seconds, $(\lfloor T/t_s \rfloor + 1)$ columns in A). Thus, to limit the needed computation, the time frame is divided in several smaller epochs. The deconvolution for each epoch is calculated separately, and then recombined to get the whole deconvolution signal.

The problem established earlier is a mean square error (L_2 norm), which allows to get an analytical solution of the problem [101]. Nevertheless, the square function of the L_2 norm exacerbates the high values and reduces the small values. The solution of problem involving the square of the functions would be

too sensitive to values exceptionally high (outliers), while being too tolerant toward small values. A more stable solution would be obtained using the L_1 norm, which involves the amplitude of the errors and not the energy. This solution would also be sparse, as small values are put to zero if useless.

However, a least- L_1 problem is more difficult to solve than a least-square problem since no analytical solution is available in this case. It has been proposed to use an Iterative Reweighted Least Square (IRLS) method [101]. This is a method to solve L_p norm problems iteratively, in which the problem is approximated by a weighted least square problem at each step, for which an analytical solution exists. So, for each step, the initial problem:

$$\min_{\beta} \sum_{i=1}^k |y_i - f_i(\beta)|^p \quad (\text{VI. 2.4})$$

Is approximated by:

$$\beta^{(t+1)} = \min_{\beta} \sum_{i=1}^k \omega_i^{(t)} |y_i - f_i(\beta)|^2 \quad (\text{VI. 2.5})$$

With:

- $\beta = \beta^{(1)}, \beta^{(2)}, \dots$: the parameters to find.
- y_i : goal function.
- $\omega_i^{(t)}$: weights to be updated at each iteration. $\omega_i^{(0)} = 1$, and $\omega_i^{(t)} = |y_i - f_i(\beta)|^{p-2}$

Notice that before each iteration, the solution is set to 0 when it is negative since the firing pattern should be positive.

The proposed deconvolution can provide the firing rate of the muscle units. Moreover, the method can be done in real-time [99], which is a requirement for the possible application in the frame of prosthetic control.

3. Classification

We will now show the interest of using the single channel deconvolution in the frame of a prosthetic control.

a. Method

Acquisition

The classification problem we will try to resolve using the single channel deconvolution is the one solved by Khushaba in his paper [8], with the data he made available on his website [102]. Ten motions patterns are to classify, and two recording channels are used.

In brief, the dataset includes EMGs from 10 subjects (2 were excluded in Khushaba's paper), aged between 20 and 35 years old. No subject suffers from limb disability, nor from any neurological or muscular disorder. They have been seated on an armchair, allowing to support and fix their arm. The subjects were asked to perform the following ten classes of movement (individual and combined fingers flexions): Thumb (TT); Index (II); Middle (MM); Ring (RR); Little (LL); Thumb and Index (TI); Thumb and Middle (TM); Thumb and Ring (TR); Thumb and Little (TL); Hand Closed (HC). These motions are shown in Figure 16.

The subjects were instructed to contract their muscles from rest position and hold the flexion for 5 seconds (the transition being included in the data). Each movement was performed six times, with a resting time of 3-to-5 seconds in-between. Four of these trials are used to train the classifier, while the other two form the test set. Thus, the training set for each subject is constituted of 4 trials for 10 motor tasks, therefore 40 recordings of 5 s each, and the test set for each subject is constituted of the remaining 20 recordings.

Surface EMG was recorded using two bipolar channels (Delsys DE 2.x series EMG sensors) and processed by the Bagnoli Desktop EMG Systems from Delsys Inc. A reference electrode has been attached on the wrist of each subject, and the two channels were near the elbow. The EMG was then amplified by a Delsys-Bagnoli-8 amplifier (total gain equal to 1000), sampled at 4000 Hz by a 12-bit analog-to-digital converter (National Instruments, BNC-2090), and acquired using Delsys EMGWorks Acquisition software.

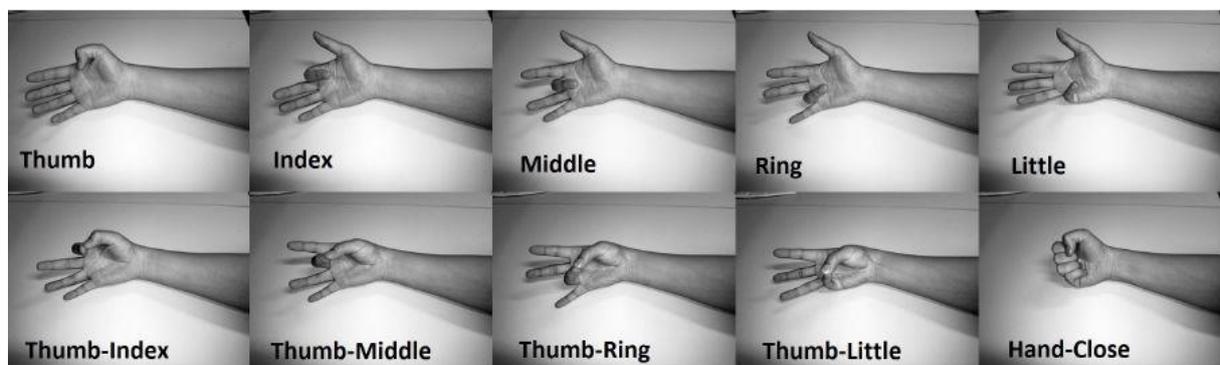


Figure 16-Motions considered in the pattern classification problem considered in this Master Thesis. These are the motion considered in the paper [8]. This figure extracted from [8] and downloaded from [102].

Signal Processing

The classical steps of EMG classification have been reminded in this work (see *Introduction*). These steps were applied either to the raw EMG or to the signals obtained after deconvolution, to understand if there is any advantage in using such a pre-processing. The details of the classification are described below.

The following signals have been considered.

1. The raw EMG, notch filtered at 50Hz and band-pass filtered between 20Hz and 400Hz. The attenuation is at least 30dB in the stopband.
2. The signal obtained by deconvolution of the EMG filtered by a notch at 50Hz and band-pass filtered between 1 and 400Hz (referred to as “deconvolution signal” in the following).
3. Deconvolution signal band-pass filtered between 20Hz and 400Hz with the same filter as for the raw EMG.

Notice that, since we deconvoluted a previously filtered EMG, some noise is already excluded from the deconvolution we will use for the classification. We decided to apply the classification to both the original deconvolution and a band-pass filtered version. This is to see if a further filtering could be useful, i.e., if the deconvolution signal is also noisy, or if adding the components out of the 20 – 400Hz band (such as the low frequency peak, which reflects the average firing rate [103]) can improve the classification results.

The signals were divided in several time windows of 250ms, slided by 50ms. Using sliding instead of disjoint windows leads to better performances but requires more computation time. The minimum window increment for windows of this size is 16ms [47], so we chose an increment of 50ms to find a compromise between computation time and performances. We have extracted, for each window, some classical TD features from each channel: MAV, RMS, ZC, SSC, WL

and IAV. The features set was then reduced using the Mutual Component Analysis. Different reduced sets have been considered, containing respectively either 4, 6 or 8 features.

We have then applied two different classifiers: SVM and the kNN. For the SVM, a “one-versus-one” strategy has been used to extend the algorithm to our multiclass problem. Concerning the kNN, the parameter k was empirically set to 5 after a rough tuning on a few preliminary tests.

Finally, a classical majority voter is applied as a post-processing. We set the number of windows in the MV to 11, which is compatible with the maximum number of decisions (13) established in [47]. Note that the maximum number of decisions, and so the maximum delay, is not the optimal delay for the average patient (100 – 125ms). But we found after a preliminary check that sticking to this optimal delay (i.e. using only 5 decisions in the MV) would increase the error rate, so we kept 11 decisions.

The classification was run on Matlab 2019b, using some codes and toolboxes available on the internet [7] [55] [104] [105].

b. Results

The error rate in classification as a function of the number of features used after reduction with the SVM approach, is displayed in Figure 17. Since the results for kNN follow the same trend, we decided to not show them for the sake of simplicity. Notice that the results shown here are not comparable to the ones obtained in [8], since different features and technics are used here, and two patients had been excluded from the previous study while we used them all here.

We observe that, for every tested configuration, using the unfiltered deconvolution leads to better results in average than using the raw EMG filtered. Specifically, the mean error rate was reduced of 4-6% (for example, using 8 features and the SVM, the error rate is 19.06% with the features extracted from the EMG while it is 13.74% when using the deconvolution). We did not plot it on the figures, but the use of the filtered deconvolution signal leads to far worse classification performances than the other two signals when it is filtered in the usual band (20-400 Hz). Indeed, the classification error rate in that case is between 4 and 10% higher than when using the EMG (for example, using 6 features and the SVM, the error rate is 19.95% with the features extracted from the EMG while it is 27.45% when using the filtered deconvolution). For this reason, this signal will be excluded from most of the following results and considerations.

As suspected, most of the classification errors come from the misinterpretation of motions involving the same fingers. For example, most of the errors when the subject makes RR are misclassification as TR. Using the deconvolution, we reduce these misinterpretation (as the mean error diminishes), but the same trend is observable. Thus, the classification based on the deconvolution signal does not solve completely the confusion between “similar motions”.

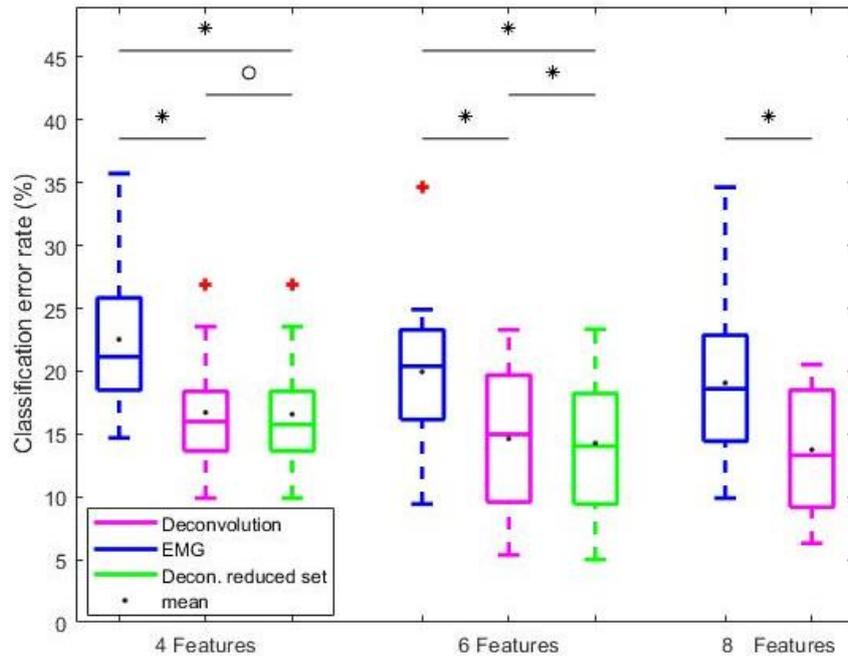


Figure 17 - Boxplots and mean classification error rate for TD features, SVM and MV with 11 decisions, depending on the number of features and the signal used, namely the EMG, the deconvolution, and the deconvolution with a reduced feature set. The bars above indicate a statistically significant difference between the classification outputs obtained from two different signals. The p-values were computed with McNemar test [106]. A circle indicates a p-value < 0.05, and an asterisk indicates a p-value < 0.01.

		EMG	Deconvolution
kNN	4 feat	7.23	4.92
	6 feat	8.83	5.20
	8 feat	8.22	5.46
SVM	4 feat	5.91	5.38
	6 feat	7.17	5.81
	8 feat	7.36	4.93

Table 2 - Standard Deviation of the error rate among all participants depending on the signal used for classification: either the EMG or the deconvolution. The standard deviation has been computed for the kNN and the SVM methods with 4, 6 or 8 features used after features reduction.

The boxplots (Figure 17) and the standard deviation of the error rates (Table 2 **Erreur ! Source du renvoi introuvable.**) allow us to say that using the deconvolution also provides more robust classification outputs toward the different subjects than with the raw EMG. Indeed, the standard deviation is smaller when using the deconvolution: the error rate is less keen to deviate from

the mean. When using the EMG to classify the motion task of the subject, some aberrant results can occur (subject 3 is by far the most relevant example, with an error rate higher than 35% for all configurations).

We can further improve the classification algorithm changing the features set. When we look at the features mostly selected by the MCA algorithm (i.e., the most informative/lowest redundant) when classifying starting from the deconvolution signals (Figure 18) we realize, in general, the same ones are used across all subjects. These features are the SSC, the WL, and the RMS. On the contrary, the ZC features (estimated after removing the mean value) were not considered for the deconvolution signal, since this signal is always positive. The IAV is rarely used, and the MAV is used a little bit more than the previous two. We can use a reduced set of features, computing only the three dominant features for both deconvolution signals (coming from the two channels), halving the number of features and so reducing the required computations. Such a distinction in the occurrences of the features was not clear when using the EMG, so this reduced feature set was considered only for the deconvolution signal. Moreover, this could remove overfitting problems. We observe on Figure 17 that the performances are slightly better using only these 6 features (3 per channel) instead of the classical 12 (6 per channel). It seems to offer a great compromise between reduced computation and great classification results.

The different approaches when using deconvolution signals, either including all temporal features or only a reduced set, are compared to the classification obtained using the EMG in Table 3 and Table 4, showing some statistically significant improvements.

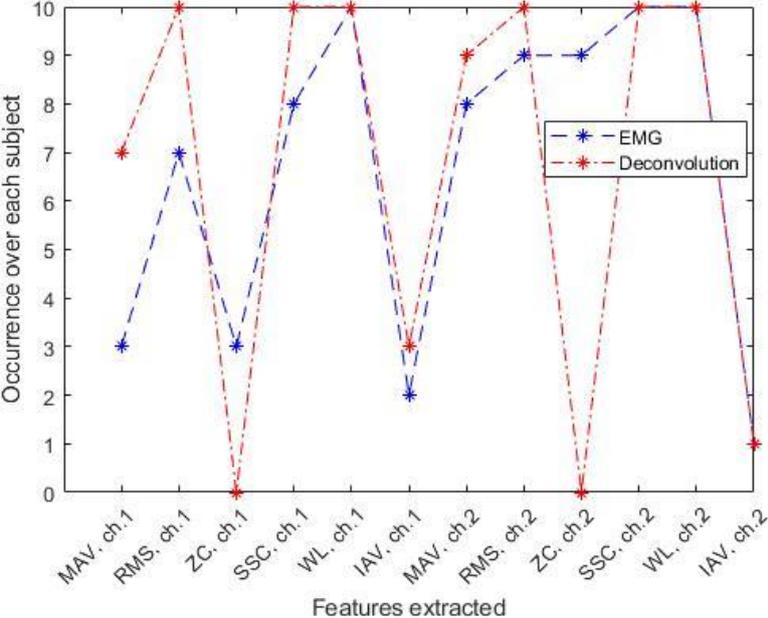


Figure 18 - Number of occurrences of every feature over 10 sets of features (one for each subject). These sets have been constructed by extracting the TD features introduced in the “Methods” section and by extracting 8 features by MCA for each subject.

Classifier	Number of features	EMG vs. Deconvolution	EMG vs. Reduced Decon.	Deconvolution vs. Reduced Decon.
kNN	4	p = 0.106	p = 0.0371	p = 0.300
	6	p = 0.0840	p = 0.0273	p = 0.412
	8	p = 0.0645	/	/
SVM	4	p = 0.0371	p = 0.0195	p = 0.500
	6	p = 0.0371	p = 0.0273	p = 0.133
	8	p = 0.0098	/	/

Table 3-Statistical significance (Wilcoxon sign rank test) between the classification results when using either the EMG or the deconvolution signal. Different classifiers, number of features selected by MCA and Time Domain (TD) sets (either complete or reduced) are considered. Significant improvements of performance when using the deconvolution signal instead of the original EMG (with a p-value lower than 0.05) are indicated in grey.

Classifier	Number of features	EMG vs. Deconvolution	EMG vs. Reduced Decon.	Deconvolution vs. Reduced Decon.
kNN	4	p < 0.001	p < 0.001	p < 0.001
	6	p < 0.001	p < 0.001	p < 0.001
	8	p < 0.001	/	/
SVM	4	p < 0.001	p < 0.001	p = 0.0405
	6	p < 0.001	p < 0.001	p = 0.00216
	8	p < 0.001	/	/

Table 4 - Statistical significance (McNemar test) between the classification results of each time window (either '1' when correctly classified or '0' when misclassified) when using either the EMG or the deconvolution signal. Different classifiers, the number of features selected by MCA, and Time Domain (TD) sets (either complete or reduced) are considered. Significant improvements of performance when using the deconvolution signal instead of the original EMG (with a p-value lower than 0.05) are indicated in grey.

4. Perspectives of this technic

The deconvolution provides an estimation of the MU cumulative firing rate. Thus, it allows to emphasize the information provided by MU firing patterns. This preliminary processing, which can be performed in real-time with only one single-differential detection channel [99], leads to better classification results than using the raw data. Indeed, deconvoluting the EMG gives access to “purer” information, more related to the MU activities, which discriminates better the different motor tasks. Other methods to get this information and to use it in a prosthesis control frame had been used in the literature before as we said earlier, and we mentioned the problems they had (they require computationally intensive processing and many detection channels). On the other hand, the estimation of the cumulative firing pattern can be achieved from a single detection channel and in real-time. Thus, the control method introduced in this Master Thesis seems to be simpler, faster and cheaper than the pre-existing ones. Moreover, filtering the deconvolution signal leads to worse results than when using the EMG, suggesting that the components out of the [20Hz –

400Hz] band are relevant and carry information useful for the classification, unlike the EMG for which the out-of-band components are essentially noise. Therefore, the whole deconvoluted signal seems to be interesting in the scope of motion classification.

Another advantage of this new technic compared to the raw EMG is the robustness toward the different subjects, as shown by a lower standard deviation on classification error rates. The high variation of the classification performances over the patients when using the raw EMG can be due to many factors, e.g., a misplacement of an electrode, an unusual noise or differences in the anatomy of the patients, reflecting on different MUAP shapes. On the contrary, when using the deconvolution (which ideally compensates for different MUAP shapes, preserving only information on firings), we obtained more stable results across subjects.

We performed two statistical tests to explore if the improvements in performances when using the deconvolution signals instead of the EMGs are statistically significant or due to chance: a Wilcoxon sign rank test between the error rate of each subject; and a McNemar test [106] between the classification output (either 'good' or 'wrong' classification) for each time window. The tests were carried out with the deconvolution signals and the EMG, for both considered classifiers (i.e., kNN and SVM) and for 4, 6 and 8 features extracted by MCA (starting from either the original or a reduced feature set). As shown in Table 3 and Table 4, statistically significant differences are asserted in most cases, namely all the classification results from SVM and the classification from kNN between the EMG and the deconvolution with reduced TD set. So, when using the SVM classifier and/or the reduced TD set, the null hypothesis can be rejected with a 5% level of confidence, and we can conclude there are statistically significant differences when using the deconvolution for

classification instead of the EMG. These outputs of the Wilcoxon sign rank test tend to confirm the advantages of the methods introduced in this section from a statistical point-of-view. In the case of the Wilcoxon test carried on the error rates of the kNN, the p-values are, in some cases (see Table 3), too high to draw a conclusion at the chosen level of confidence. Nevertheless, still lower median classification rates were consistently obtained when using the deconvolution signal, independently of the number of features included. Note that no statistical improvement between the deconvolution and the deconvolution with a reduced feature set can be asserted at the given level of confidence. Nevertheless, the reduction of the set has advantages we explained earlier.

Even if our new methods based on single-channel deconvolution seems to be of interest for prosthetic control, our study did not cover the whole aspect of this field and some points still must be answered. Firstly, as we mentioned in the *VI.3.b-Results* section, the confusion between the classes were found when the same fingers were in motion (e.g., between TL and LL, or TR and RR). Despite the reduction of the confusion, we did not completely solve it when using the pre-processing by deconvolution. We can think that the cumulative firing rate to bend the ring finger alone or to bend the ring with the thumb will be essentially the same, and that is why using the deconvolution still leads to an important confusion. Further information could be needed to discriminate those motions. Some classifiers have also been introduced to classify simultaneous motions [107]: applying such approaches to the deconvolution signal could possibly improve the classification performances.

Secondly, only one domain of features, the time-domain, has been considered. As we mentioned earlier, many others are used in the literature, some of them showing excellent classification results [52] [57], that we did not test on this preliminary work. Could we provide, starting from the deconvolution, other kind

of features that lead to even better classification results? Or is it only efficient for TD?

Thirdly, the data we used here were recorded in laboratory conditions, and researchers have noted how different can be the results between these conditions and the daily life [13]. The arm of the subjects was on a support, in a certain position that was the same for all trials. We cannot draw any conclusion on the robustness toward an electrode misplacement or a different arm position. Moreover, the effect of the fatigue was not investigated. Both central and peripheral myoelectric manifestations of fatigue have been documented [108]. As the deconvolution signal ideally reflects central control, compensating for MUAPs shapes (changing as a peripheral manifestation of fatigue), we can speculate that it should be affected to a lesser extent with respect to the raw EMG (which is affected by both central and peripheral fatigue). These aspects are surely to be explored if a commercial use is considered. The data were also processed offline, and the method was not tested for a continuous flow of data. Testing it would give us the confirmation that our proposal meets the real-time requirements. Finally, no prototype was built: we could consider it the final test, to ensure a good user experience through getting the output expected by the user in a reasonable delay.

Nevertheless, beside the limitations of this master thesis, our method is promising because it overcomes the EMG in the classification process, without needing other complex detection system and by keeping the computational cost low. It seems to be a path to explore for the future of the prosthesis control and further researches in that field, beginning by testing the most advanced methods used for the EMG in the literature, such as force and pattern classification [109].

Conclusion

Myoelectric prostheses are key devices to help people with disabilities or who had to undergo surgery to recover motion capabilities. Nowadays, the state-of-the-art indicates that most of the proposed strategies follow the same scheme of pattern classification, that we described here. For each stage, many different processing techniques, among which there are classical algorithms of machine learning, have been described and shown to be effective.

In most papers in the field, the surface EMG is used because of its simplicity and its non-invasiveness. However, due to the filtering effect of the volume conductor, the EMG does not give a precise insight of the MU activity. Works have been done with numerical methods extracting the firing pattern of the MU, and the obtained signal was used to control a prosthesis. These methods were proved to be interesting in spite of the limitations (dense detection arrays and expensive computation). We proved that simpler deconvolution methods (single differential channel and real-time processing) could be used for prosthesis control and leads to great results compared to classical EMG processing. This could be an interesting alternative for prosthetic pattern classification in the future, but many questions still need to be answered (they are essentially the same as for EMG), concerning the robustness and the parallel classification, to fully consider this effective for real devices.

Appendix

MCA algorithm

Let F be the original feature set, of dimension p . The aim of the MCA is to get a feature set L of dimension $k < p$, with $L \subset F$, conserving the learning information. We define $g(L)$ an evaluation function for the new feature subset. The algorithm is directly taken from [81].

- Initialisation: $F = \{f_i, i = 1, \dots, p\}$, $L = \emptyset$, $g(L) = 0$;
- Step 1: For each $f_i \in F$, compute $I(C, f_i)$. We choose the feature f_j that produces the highest information toward the class labels, remove it from F and add it to the new subset: $F \leftarrow F \setminus \{f_j\}$, $L \leftarrow \{f_j\}$ and $g(L) \leftarrow I(C, f_j)$.
- Step 2: For each feature $f_i \in F$ and $f_i \notin L$, we compute $m(f_i) = g(L) + \lambda I(C, f_i)$, where λ is the information gain. The feature f_j that maximizes m is added to L , removed from F , and $g(L) = m(f_j)$.

The information gain λ is given by: $\lambda = \frac{2}{1 + \exp(-\alpha D)} - 1$, with:

○ $D =$

$$\min_{f_j \in L} \left[\frac{H(f_i) - I(f_i; f_j)}{H(f_i)} \right] \frac{1}{|L|} \sum_{f_j \in L} \exp \left(\frac{I(C; f_i) + I(C; f_j) - I(C; \{f_i, f_j\})}{H(C)} \right)$$

○ $\alpha = 0.3$ a small value empirically set to make $\lambda \approx 0$

○ $|L|$ the cardinal of L

and approximates the amount of information added to the set by adding f_i . If f_i is completely redundant with the features already in L , $H(f_i) - I(f_i; f_j) = 0$ and thus $\lambda = 0$. On the other hand, if f_i brings a new information in L , $\frac{H(f_i) - I(f_i; f_j)}{H(f_i)} = 1$ and $\lambda \approx 1$ since $I(C; \{f_i, f_j\}) = 0$.

- Step 3: if $|L| < p$, return to step 2.

- Step 4: normalize the gains in $g(L)$ by dividing it by the size of the subset to remove the size effect of the estimated gain: $m(f_i) \leftarrow m(f_i)/i$ with $i = 1, \dots, p$.
- Step 5: the k first features ($k < p$) with a normalized value of m such as $m(f_i) > \frac{1}{|L|} \sum_i m(f_i)$ are selected to form the new feature set.
- Step 6: Apply PCA to the new set.

Références

- [1] M. A. LeBlanc et M. CP, «Innovation and Improvement of Body-Powered Arm Prostheses: A First Step,» 1985.
- [2] P. Geethanjali, «Myoelectric control of prosthetic hands: state-of-the-art review,» *Medical devices (Auckland, N.Z.)*, vol. 9, p. 247—255, 2016.
- [3] C. K. Battye, A. Nightindale et J. Whillis, «The use of myo-electric currents in the operation of prostheses,» *The Journal of bone and joint surgery*, vol. 37B, p. 506—510, 1955.
- [4] D. McKenzie, «The Russian Myoelectric Arm,» *The Journal of bone and joint surgery. British volume*, vol. 47, p. 418—420, August 1965.
- [5] B. Hudgins, P. Parker et R. Scott, «A new strategy for multifunction myoelectric control,» *IEEE transactions on bio-medical engineering*, vol. 40, pp. 82-94, 1993.
- [6] F. Cordella, A. Ciancio, R. Sacchetti, A. Davalli, A. Cutti, E. Guglielmelli et L. Zollo, «Literature Review on Needs of Upper Limb Prosthesis Users,» *Frontiers in Neuroscience*, vol. 10, 2016.
- [7] R. Khushaba et S. Kodagoda, «Electromyogram (EMG) feature reduction using Mutual Components Analysis for multifunction prosthetic fingers control,» 2012.
- [8] R. Khushaba, S. Kodagoda, M. Takruri et G. Dissanayake, «Toward Improved Control of Prosthetic Fingers Using Surface Electromyogram (EMG) Signals,» *Expert Systems with Applications*, vol. 39, p. 10731—10738, 2012.
- [9] N. Parajuli, N. Sreenivasan, P. Bifulco, M. Cesarelli, S. Savino, V. Niola, D. Esposito, T. Hamilton, G. Naik, U. Gunawardana et G. Gargiulo, «Real-Time EMG Based Pattern Recognition Control for Hand Prostheses: A Review on Existing Methods, Challenges and Future Implementation,» *Sensors (Basel)*, vol. 19, p. 4596, 2019.
- [10] P. Kyberd, C. Wartenberg, L. Sandsjö, S. Jönsson, D. Gow, J. Frid, C. Almström et L. Sperling, «Survey of Upper-Extremity Prosthesis Users in Sweden and the United Kingdom,» *PO Journal of Prosthetics and Orthotics*, vol. 19, pp. 55-62, 2007.
- [11] N. Jiang, S. Dosen, K. Muller et D. Farina, «Myoelectric Control of Artificial Limbs—Is There a Need to Change Focus?,» *IEEE Signal Processing Magazine*, vol. 29, pp. 152-150, 2012.
- [12] O. Samuel, M. Asogbon, Y. Geng, A. Al-Timemy, S. Pirbhulal, N. Ji, S. Chen, P. Fang et G. Li, «Intelligent EMG Pattern Recognition Control Method for Upper-Limb Multifunctional Prostheses: Advances, Current Challenges, and Future Prospects,» *IEEE Access*, vol. 7, pp. 10150-10165, 2019.
- [13] E. Campbell, A. Phinyomark et E. Scheme, «Current Trends and Confounding Factors in Myoelectric Control: Limb Position and Contraction Intensity,» *Sensors (Basel)*, 2020.
- [14] L. Mesin, *Neuromuscular System Engineering*, 2019.

- [15] R. Enoka, «Motor Unit,» *Wiley Encyclopedia of Biomedical Engineering*, 2006.
- [16] S. Ebashi et M. Endo, «Calcium and muscle contraction,» *Progress in Biophysics and Molecular Biology*, vol. 18, pp. 123-183, 1968.
- [17] E. Henneman, G. Somjen et D. Carpenter, «Functional Significance of Cell Size in Spinal Motoneurons,» *Journal of neurophysiology*, vol. 28, pp. 560-580, 1965.
- [18] A. J. Fuglevand, D. A. Winter et A. E. Patla, «Models of recruitment and rate coding organization in motor-unit pools,» *Journal of neurophysiology*, vol. 70, pp. 2470-2488, 1993.
- [19] C. De Luca, R. LeFever, M. McCue et A. Xenakis, «Behaviour of human motor units in different muscles during linearly varying contractions,» *Journal of Physiology*, vol. 329, pp. 113-28, 1982.
- [20] H. Milner-Brown, R. Stein et R. Yemm, «Changes in firing rate of human motor units during linearly changing voluntary contractions,» *Journal of Physiology*, vol. 230, pp. 371-90, 1973.
- [21] R. Granit, D. Kernell et G. K. Shortess, «The behaviour of mammalian motoneurons during long-lasting orthodromic, antidromic and trans-membrane stimulation,» *The Journal of physiology*, vol. 169, pp. 743-754, 1963.
- [22] A. L. Hodgkin et A. F. Huxley, «Currents carried by sodium and potassium ions through the membrane of the giant axon of Loligo,» *The Journal of physiology*, vol. 116, p. 449-472, 1952.
- [23] D. Goldman, «Potential, impedance, and rectification in membranes,» *Journal of General Physiology*, vol. 27, p. 37-60, 1943.
- [24] A. Hodgkin et B. Katz, «The effect of sodium ions on the electrical activity of the giant axon of the squid,» *Journal of Physiology*, vol. 108, pp. 37-77, 1949.
- [25] A. L. Hodgkin et A. F. Huxley, «A quantitative description of membrane current and its application to conduction and excitation in nerve,» *The Journal of physiology*, vol. 117, pp. 500-544, 1952.
- [26] A. Hodgkin et W. Rushton, «The electrical constants of a crustacean nerve fibre,» *Proc R Soc Med*, vol. 134, pp. 444-79, 1946.
- [27] P. Rosenfalck, «Intra- and extracellular potential fields of active nerve and muscle fibres. A physico-mathematical analysis of different models,» *Acta Physiol Scand Suppl*, vol. 321, pp. 1-168, 1969.
- [28] D. Farina, L. Mesin et S. Martina, «Advances in surface electromyographic signal simulation with analytical and numerical descriptions of the volume conductor,» *Medical & biological engineering & computing*, vol. 42, p. 467-476, 2004.
- [29] J. Clark et R. Plonsey, «The extracellular potential field of the single active nerve fiber in a volume conductor,» *Biophysical journal*, vol. 8, p. 842-864, 1968.
- [30] L. Mesin, «Analytical generation model of surface electromyogram for multi-layer volume conductors,» *chez Modelling in Medicine and Biology IV*, 2005.

- [31] D. Farina, L. Mesin, S. Martina et R. Merletti, «A surface EMG generation model with multilayer cylindrical description of the volume conductor,» *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 415-426, 2004.
- [32] D. Farina et R. Merletti, «A novel approach for precise simulation of the EMG signal detected by surface electrodes,» *IEEE Transactions on Biomedical Engineering*, vol. 48, pp. 637-646, 2001.
- [33] M. Lowery, N. Stoykov, J. Dewald et T. Kuiken, «Volume Conduction in an Anatomically Based Surface EMG Model,» *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 2138 - 2147, 2005.
- [34] S. Pullman, D. Goodin, A. Marquinez, S. Tabbal et M. Rubin, «Clinical utility of surface EMG,» *Neurology*, vol. 55, pp. 171-177, 2000.
- [35] R. F. Weir, P. R. Troyk, G. DeMichele et T. Kuiken, «Implantable Myoelectric Sensors (IMES) for Upper-Extremity Prosthesis Control - Preliminary Work,» *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, vol. 2, pp. 1562-1565, 2003.
- [36] T. Farrell et R. Weir, «Pilot comparison of surface vs. implanted EMG for multifunctional prosthesis control,» chez *IEEE 9th International Conference on Rehabilitation Robotics*, 2005.
- [37] R. Merletti, A. Botter, A. Troiano, E. Merlo et M. Minetto, «Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art,» *Clinical biomechanics*, vol. 24, pp. 122-134, 2008.
- [38] Y. Liu, C. Zhang, N. Dias, Y. Chen, S. Li, P. Zhou et Y. Zhang, «Transcutaneous innervation zone imaging from high-density surface electromyography recordings,» *Journal of Neural Engineering*, vol. 17, 2020.
- [39] D. Farina et R. Merletti, «Estimation of average muscle fiber conduction velocity from two-dimensional surface EMG recordings,» *Journal of Neuroscience Methods*, vol. 134, pp. 199-208, 2004.
- [40] K. Roeleveld et D. Stegeman, «What do we learn from motor unit action potentials in surface electromyography?,» *Muscle Nerve Suppl*, vol. 11, pp. 92-97, 2002.
- [41] C. Metting van Rijn, A. Peper et C. Grimbergen, «Amplifiers for bioelectric events: A design with a minimal number of parts,» *Medical & biological engineering & computing*, vol. 32, pp. 305-310, 1994.
- [42] J. Mogk et P. Keir, «Crosstalk in surface electromyography of the proximal forearm during gripping tasks,» *Journal of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology*, vol. 13, pp. 63-71, 2003.
- [43] W. Franks, I. Schenker, P. Schmutz et A. Hierlemann, «Impedance Characterization and Modeling of Electrodes for Biomedical Applications,» *IEEE transactions on bio-medical engineering*, vol. 52, pp. 1295-302, 2005.

- [44] D. Hewson, J. Hogrel, Y. Langeron et J. Duchêne, «Evolution in impedance at the electrode-skin interface of two types of surface EMG electrodes during long-term recordings,» *J Electromyogr Kinesiol*, vol. 13, pp. 273-9, 2003.
- [45] K.-P. Hoffmann, R. Ruff et W. Poppendieck, «Long-Term Characterization of Electrode Materials for Surface Electrodes in Biopotential Recording,» chez *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006.
- [46] T. Farrell, *Multifunctional Prosthesis Control: The Effects of Targeting Surface vs. Intramuscular Electrodes on Classification Accuracy and the Effect of Controller Delay on Prosthesis Performance*, NORTHWESTERN UNIVERSITY, 2007.
- [47] K. Englehart et B. Hudgins, «A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control,» *IEEE transactions on bio-medical engineering*, vol. 50, pp. 848-54, 2003.
- [48] R. Chowdhury, «Surface Electromyography Signal Processing and Classification Techniques,» *Sensors*, vol. 13, pp. 12431-12466, 2013.
- [49] C. De Luca, L. Gilmore, M. Kuznetsov et S. Roy, «Filtering the surface EMG signal: Movement artifact and baseline noise contamination,» *Journal of Biomechanics*, vol. 43, pp. 1573-1579, 2010.
- [50] M. Lowery, N. Stoykov et T. Kuiken, «A simulation study to examine the use of cross-correlation as an estimate of surface EMG cross talk,» *Journal of applied physiology*, vol. 94, pp. 1324-34, 2003.
- [51] S. Viljoen, T. Hanekom et D. Farina, «Effect of characteristics of dynamic muscle contraction on crosstalk in surface electromyography recordings,» *SAIEE Africa Research Journal*, vol. 98, pp. 18-28, 2007.
- [52] K. Englehart, B. Hudgins, P. Parker et M. Stevenson, «Classification of the myoelectric signal using time-frequency based representations,» *Medical engineering & physics*, vol. 21, pp. 431-8, 1999.
- [53] M. Zardoshti-Kermani, B. C. Wheeler, K. Badie et R. M. Hashemi, «EMG feature evaluation for movement control of upper extremity prostheses,» *IEEE Transactions on Rehabilitation Engineering*, vol. 3, pp. 324-333, 1995.
- [54] C. Li, J. Ren, H. Huang, B. Wang, Y. Zhu et H. Hu, «PCA and deep learning based myoelectric grasping control of a prosthetic hand,» *BioMedical Engineering OnLine*, vol. 17, 2018.
- [55] A. Chan et G. Green, «Myoelectric control development toolbox,» chez *30th Conference of the Canadian Medical & Biological Engineering Society*, Toronto, Canada, 2007.
- [56] K. A. Farry, I. D. Walker et R. G. Baraniuk, «Myoelectric teleoperation of a complex robotic hand,» *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 775-788, 1996.
- [57] K. Englehart, B. Hudgins et P. Parker, «A wavelet-based continuous classification scheme for multifunction myoelectric control,» *IEEE transactions on bio-medical engineering*, vol. 48, pp. 302-11, 2001.

- [58] N. Saito et R. Coifman, «Local discriminant bases and their applications,» *Journal of Mathematical Imaging and Vision*, vol. 5, pp. 337-358, 1995.
- [59] L. Hargrove, L. Guanglin, K. Englehart et B. Hudgins, «Principal Component Analysis Preprocessing for Improved Classification Accuracy in Pattern-Recognition-Based Myoelectric Control,» *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 1407-1414, 2009.
- [60] M. Welling, *Fisher Linear Discriminant Analysis*, Department of Computer Science, University of Toronto, 2006.
- [61] J. Ye, R. Janardan, C. Park et H. Park, «An Optimization Criterion for Generalized Discriminant Analysis on Undersampled Problems,» *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, pp. 982-94, 2004.
- [62] J. Ye, R. Janardan, Q. Li et H. Park, «Feature extraction via generalized uncorrelated linear discriminant analysis,» chez *Twenty-first International Conference on Machine learning (ICML '04)*, New-York, NY, USA, 2004.
- [63] C. E. Shannon, «A mathematical theory of communication,» *The Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.
- [64] J. Blumberg, J. Rickert, S. Waldert, A. Schulze-Bonhage, A. Aertsen et C. Mehring, «Adaptive classification for brain computer interfaces,» chez *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, 2007.
- [65] H. Zhang, Y. Zhao, F. Yao, L. Xu, P. Shang et P. Li, «An adaptation strategy of using LDA classifier for EMG pattern recognition,» chez *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2013.
- [66] T. Cover et P. Hart, «Nearest Neighbor Pattern Classification,» *IEEE Transactions on Information Theory*, vol. 13, pp. 21-27, 1967.
- [67] P. Hall, B. Park et R. Samworth, «Choice of neighbor order in nearest-neighbor classification,» *The Annals of Statistics*, vol. 36, 2008.
- [68] K. Hechenbichler et K. Schliep, «Weighted k-Nearest-Neighbor Techniques and Ordinal Classification,» 2004.
- [69] C. Cortes et V. Vapnik, «Support-Vector Networks,» *Machine Learning*, vol. 20, pp. 273--297, 1995.
- [70] T. Hofmann, B. Schölkopf et A. Smola, «Kernel methods in machine learning,» *The Annals of Statistics*, vol. 36, 2007.
- [71] B. E. Boser, I. M. Guyon et V. N. Vapnik, «A Training Algorithm for Optimal Margin Classifiers,» chez *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 1992.
- [72] T. Kohonen, «The self-organizing map,» *Proceedings of the IEEE*, vol. 78, pp. 1464-1480, 1990.

- [73] D. Rumelhart, G. Hinton et R. J. Williams, «Learning representations by back-propagating errors,» *Nature*, vol. 323, pp. 533-536, 1986.
- [74] K. Levenberg, «A method for the solution of certain non-linear problems in least squares.,» *Quarterly of Applied Mathematics*, vol. 2, pp. 164-168, 1944.
- [75] D. Marquardt, «An Algorithm for Least-Squares Estimation of Nonlinear Parameters,» *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, p. 431–441, 1963.
- [76] J. R. Quinlan, «Induction of decision trees,» *Machine Learning*, p. 81–106 , 1986.
- [77] L. Breiman, J. Friedman, R. Olshen et C. J. Stone, *Classification and Regression Trees*, 1983.
- [78] T. K. Ho, «Random decision forests,» chez *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Montreal, 1995.
- [79] L. Breiman, «Random Forests,» *Machine Learning*, vol. 45, pp. 5-32, 2001.
- [80] G. Yu, G. Sapiro et S. Mallat, «Solving Inverse Problems With Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity,» *IEEE Transactions on Image Processing*, vol. 21, 2010.
- [81] Y. Huang, K. Englehart, B. Hudgins et A. Chan, «A Gaussian Mixture Model Based Classification Scheme for Myoelectric Control of Powered Upper Limb Prostheses,» *IEEE transactions on bio-medical engineering*, vol. 52, pp. 1801-11, 2005.
- [82] D. & R. R. Reynolds, «Robust text-independent speaker identification using Gaussian Mixture speaker models,» *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 72 - 83, 1995.
- [83] D. Reynolds, T. Quatieri et R. Dunn, «Speaker Verification Using Adapted Gaussian Mixture Models,» *Digital Signal Processing*, vol. 10, pp. 19-41, 2000.
- [84] T. Farrell et R. Weir, «The Optimal Controller Delay for Myoelectric Prostheses,» *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 15, pp. 111 - 118, 04 2007.
- [85] P. Domingos et M. Pazzani, «Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier,» *Machine Learning*, pp. 105-112, 1996.
- [86] M. Khezri et M. Jahed, «Real-time intelligent pattern recognition algorithm for surface EMG signals,» *Biomedical Engineering Online*, vol. 6, p. 45, 2007.
- [87] M. R. Ahsan, M. Ibrahimy et O. Khalifa, «Optimization of Neural Network for Efficient EMG Signal Classification,» chez *8th International Symposium on Mechatronics and its Applications*, 2012.
- [88] M. Tavakoli, C. Benussi et J. Lourenco, «Single Channel Surface EMG Control of Advanced Prosthetic Hands: A Simple, Low Cost and Efficient Approach,» *Expert Systems with Applications*, vol. 79, 2017.

- [89] A. Radmand, E. Scheme et K. Englehart, «High-density force myography: A possible alternative for upper-limb prosthetic control,» *Journal of Rehabilitation Research and Development*, vol. 53, pp. 443-456, 2016.
- [90] A. Belyea, K. Englehart et E. Scheme, «FMG vs EMG: A Comparison of Usability for Real-time Pattern Recognition Based Control,» *IEEE Transactions on Biomedical Engineering*, vol. PP, pp. 1-1, 2019.
- [91] D. Farina, A. Holobar, R. Merletti et R. Enoka, «Decoding the neural drive to muscles from the surface electromyogram,» *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, vol. 121, pp. 1616-23, 2010.
- [92] D. Farina, R. Merletti et R. Enoka, «The extraction of neural strategies from the surface EMG: An update,» *Journal of applied physiology*, vol. 96, pp. 1486-95, 2004.
- [93] A. Holobar et D. Zazula, «Multichannel Blind Source Separation Using Convolution Kernel Compensation,» *IEEE Transactions on Signal Processing*, vol. 55, pp. 4487 - 4496.
- [94] R. Enoka et J. Duchateau, «Rate Coding and the Control of Muscle Force,» *Cold Spring Harbor Perspectives in Medicine*, vol. 7, 2017.
- [95] A. Vecchio, F. Negro, A. Holobar, A. Casolo, J. P. Folland, F. Felici et D. Farina, «You are as fast as your motor neurons: speed of recruitment and maximal discharge of motor neurons determine the maximal rate of force development in humans,» *The Journal of Physiology*, vol. 597, 2019.
- [96] D. Farina, I. Vujaklija, M. Sartori, T. Kapelner, F. Negro, N. Jiang, K. Bergmeister, A. Andalib, J. Principe et O. Aszmann, «Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation,» *Nature Biomedical Engineering*, vol. 1, 2017.
- [97] T. Kapelner, F. Negro, O. C. Aszmann, Farina et D., «Decoding Motor Unit Activity From Forearm Muscles: Perspectives for Myoelectric Control,» *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, pp. 244-251, 2018.
- [98] C. Chen, G. Chai, W. Guo, X. Sheng, D. Farina et X. Zhu, «Prediction of finger kinematics from discharge timings of motor units: implications for intuitive control of myoelectric prostheses,» *Journal of Neural Engineering*, vol. 16, 2018.
- [99] L. Mesin, «Single channel surface electromyogram deconvolution to explore motor unit discharges,» *Medical & Biological Engineering & Computing*, pp. 1-10, 2019.
- [100] C. Groetsch, *The theory of Tikhonov regularization for Fredholm equations of the first kind*, Pitman, 1984.
- [101] C. Burrus, «Iterative Reweighted Least Squares *,» 2014.
- [102] R. Khushaba. [En ligne]. Available: <https://www.rami-khushaba.com>. [Accès le 02 03 2020].

- [103] L. Mesin, «Non-Propagating Components of Surface Electromyogram Reflect Motor Unit Firing Rates,» *IEEE*, vol. 7, pp. 106155--106161, 2019.
- [104] R. N. Khushaba, A. Al-Jumaily et A. Al-Ani, «Novel Feature Extraction Method based on Fuzzy Entropy and Wavelet Packet Transform for Myoelectric Control,» chez *7th International Symposium on Communications and Information Technologies ISCIT2007*, Sydney, Australia, 2007.
- [105] I. Nabney, «MATLAB Central File Exchange-Netlab,» [En ligne]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/2654-netlab>. [Accès le September 2020].
- [106] G. Cardillo, «McNemar test,» 2021. [En ligne]. Available: <https://github.com/dnafinder/mcnemar>.
- [107] A. J. Young, L. H. Smith, E. J. Rouse et L. J. Hargrove, «Classification of Simultaneous Movements Using Surface EMG Pattern Recognition,» *IEEE Transactions on Biomedical Engineering*, vol. 60, pp. 1250-1258, 2013.
- [108] L. Mesin, D. Dardanello, A. Rainoldi et G. Boccia, «Motor unit firing rates and synchronisation affect the fractal dimension of simulated surface electromyogram during isometric/isotonic contraction of vastus lateralis muscle,» *Medical Engineering & Physics*, vol. 38, pp. 1530-1533, 2016.
- [109] C. Castellini et P. van der Smagt, «Surface EMG in advanced hand prosthetics,» *Biological cybernetics*, vol. 100, pp. 35-47, 2008.