



THE UNIVERSITY
of EDINBURGH

**Independent Component Analysis (ICA) and Signal Decomposition
for surface Electromyography (sEMG) Signals**

Andreas Christou

Master of Science
Operational Research with Data Science
School of Mathematics
University of Edinburgh
August 2020

Supervised by Dr. Timothy Cannings

Own work declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

Andreas Christou
Edinburgh, 21/08/2020

Abstract

Independent Component Analysis (ICA) has received a lot of attention in statistical as well as in biomedical signal processing. It is widely used in blind source separation (BSS) problems, as it is a convenient method to separate signals from different sources, without any prior information about them or the mixing process. In the first part of the dissertation we report a theoretical background of ICA, analyzing what are the preprocessing steps that are needed and how ICA works, and then giving more details on the two algorithms that are compared, fastICA and ProDenICA. In the second part we present experimental results in a simulation environment to see what ICA achieves and what are the merits and drawbacks of the two ICA algorithms while in the third part we consider a real surface Electromyography (sEMG) dataset. sEMG is one type of bioelectrical signals produced by the human body and contain significant information about muscle activity. ICA is applied to sEMG signals in order to recover the original signals originating from each muscle. Besides, a post-ICA method that overcomes the independent component ordering ambiguity is proposed, allowing them to be related to the suitable corresponding muscles. ICA and the post-ICA steps that are described, manage to reach more than 79% accuracy on three hand gesture classification problems. The experimental results in both simulation and sEMG dataset indicate that ICA is an appropriate method for signal recovering and identification of hand gestures using sEMG signals.

Keywords: Independent Component Analysis, ICA, Electromyography, sEMG, signal processing

Acknowledgements

I would like to deeply thank my supervisor Dr. Timothy Cannings for his support and guidance during the dissertation. He was always available and eager to help and give valuable advice since the very first day until the end. His continuous feedback and recommendations were substantial for the dissertation. I would also like to thank him for his patience during our weekly online meetings, and his willingness to help solving any problem faced during these three months.

I would also like to thank my parents Christos and Elena, as well as my brother Yiannis and my sister Rebecca for their encouragement and support during this intense academic year.

Finally, a big thank you to my girlfriend Semina for being patient, supportive and caring through my master's degree, as well as my friends for being always next to me.

Table of Contents

1. Introduction	2
1.1 Aim	3
1.2 Structure	3
2. Background	4
2.1 Independent Component Analysis	4
2.1.1 ICA Restrictions	5
2.1.2 Ambiguities of ICA	6
2.1.3 Preprocessing	6
2.1.4 Solving ICA	8
2.1.5 ICA Algorithms	9
2.2 Surface Electromyography (sEMG)	11
2.2.1 Limitations	12
2.2.2 sEMG and ICA.....	12
3. Simulation	13
3.1 First Steps	13
3.2 Time Series Signals	14
3.3 More Time Series Signals	15
3.4 fastICA vs ProDenICA	17
4. sEMG Dataset	19
4.1 Dataset Description	19
4.2 Dataset Analysis	20
4.2.1 Exploration	20
4.2.2 Baseline	22
4.2.3 fastICA	23
4.2.4 Bins	25
4.2.5 Windows	26
4.2.6 One for All	27
4.2.7 Correlation Matrix with mixtures	28
4.2.8 Results Summary	30
4.2.9 Limitations	31
5. Conclusion	34
5.1 Review	34
5.2 Further Steps	35
References	36

1 . Introduction

Analysing signal recordings is challenging for numerous reasons as observed signals are affected by several circumstances. Noise from the recording equipment, but also delay of signal recording between the electrodes that record the signal, are responsible for obscuring the recorded signal. Moreover, recording techniques between experiments influence the measured signals. Often, signal recordings describe a human body activity, and analyzing such signals is a very challenging task. Human characteristics such as muscle anatomy and physiology, as well as nerve disorders or neuromuscular problems, lead to more complicated signal recordings.

Nevertheless, these reasons are not the only ones that can affect the signal. Some artifacts from other parts of the body, such as heartbeats, are also responsible for obscuring the signal. Furthermore, when a person performs a gesture, more muscles are activated. As a result, electrodes do not record only the closest muscle's activity but signals from other neighbouring muscles as well. This is called cross-talk and it is more obvious when electrodes are placed on muscles that are very closed to each other like in forearm. Consequently, a signal recorded at the surface of a muscle is a superposition of activity generated by the nearby muscles.

Blind Source Separation (BSS) can be used to recover individual signals without any prior information of them or the mixing procedure. One of the most widely used BSS approaches to tackle this problem is Independent Component Analysis (ICA), and the goal is to recover the original component signals from the mixture signals. In ICA, observed signals are assumed to be linear mixtures of some latent variables (source signals) which are considered to be statistically independent and have non-Gaussian probability distributions. Although there are many good reasons to consider this method as a solution to such problems, ICA introduces also some unavoidable ambiguities. Thus, these problems should be taken into account when applying this approach. ICA is an unsupervised learning method, and consequently, the evaluation of this approach is not trivial.

Surface Electromyography (sEMG) is a non-invasive technique of recording muscle activity. A significant property of sEMG signals is that a signal recorded from a muscle can be considered to be independent of other signals originating from other neighbouring muscles, as well as of other bioelectrical signals (ECG, EOG, etc.). This assumption gives the opportunity for ICA to be used in such applications.

1.1 Aim

The goal of the dissertation is to exploit the benefits that ICA provides in order to extract interpretable signals from the observed mixtures. Two very widely known ICA algorithms, called fastICA and ProDenICA, are applied. The performance of the two algorithms, as well as their computational complexity, are studied in a simulation environment. As stated, ICA is an unsupervised learning method. However, for the simulations, a supervised learning technique is analysed, using predetermined generated source signals. Apart from the simulations, real surface Electromyography signals dataset is used. The dataset contains mixed sEMG signals recorded within an experiment. Each subject participated in this experiment performed a series of hand gestures and the signals were recorded using a bracelet with 8 sensors placed on the forearm. Since there is no information about the source signals, this is an unsupervised learning problem. It is difficult to evaluate the quality of signal separation, so the measure that is used to evaluate the performance of the method is how accurately the system identifies the hand gesture correctly. During this unsupervised problem, a significant problem is faced. This is the independent components' ordering ambiguity. Each time ICA runs, it gives the estimated independent components in a different order. This permutation problem should be solved and we propose a post-ICA method in order to give a solution to this.

1.2 Structure

This dissertation consists of five chapters. The current chapter contains a small introduction on the necessity of signal analysis and decomposition, as well as the aim of the dissertation. In chapter 2, a deep analysis of a BSS technique called ICA is made. Besides, ICA has some limitations and ambiguities when it is applied, and these are presented as well. The preprocessing steps required for ICA as well as two techniques to minimize the non-Gaussianity are also reviewed. Two of the most widely known and used ICA algorithms are presented. This chapter closes with the introduction on sEMG signals, their limitations and how ICA can be used on sEMG signals. In the next chapter, an effort is made to simulate time series signals, then mix them and then try to decompose them. This is done, in order to be able to evaluate how ICA performs when decomposing the mixture signals. The two algorithms presented in chapter 2 are compared in terms of signal recovering performance, as well as their computational complexity. In chapter 4, a real sEMG dataset for gestures is explained. Then there is a presentation of all the post-ICA steps that were followed in order to overcome some ambiguities ICA introduced, and build a model that will be efficient in terms of gesture classification accuracy. Finally, the dissertation is completed in chapter 5, with some conclusions and some suggestions for further research.

2. Background

One can imagine a situation that he is in a cocktail party, where he can hear many different people talking while a song is played in the background. The problem is that he is trying to understand what his friends are telling when talking to him, but he is unable to do it because of the different talks and sounds that are heard at the same time. The solution to this problem is given by the Independent Component Analysis (ICA). Similar but more practical problems to the cocktail party problem are audio separation and noise cancelling, where ICA tries to detect the noise and remove it from audio signals. Moreover, ICA can be used in financial data to reveal some hidden properties of parallel time series (currency exchange rates). Besides, ICA is used in biomedical signal processing to detect disturbances that occur in signals and such example will be analysed in chapter 4.

2.1 Independent Component Analysis

Independent Component Analysis (Bell & Sejnowski, 1995; Comon, 1994; Hyvärinen, Karhunen, & Oja, 2001; Hyvärinen & Oja, 2000; Jutten & Herault, 1991) gives the solution to problems similar to the Cocktail Party problem. Different signals are emitted by different sources, and some sensors observe a mixture of these signals. An important assumption that the ICA does, is that the observed signals \mathbf{x} taken by some sensors are linear mixtures of the original signals \mathbf{s} . Each sensor observes a different linear combination of these signals. The observed signals are said to be weighted sums of the original ones, where the weights depend on the distance of the sensors from the sources.

To make this more clear, someone can think of some signals s_1, s_2 and s_3 and some observed mixtures of these signals, x_1, x_2 and x_3 . Each one of the observed signals can be thought as linear combinations of the initial signals as below, where the coefficients a_{ij} are the mixing weights.

$$\begin{aligned}x_1(t) &= a_{11} * s_1(t) + a_{12} * s_2(t) + a_{13} * s_3(t) \\x_2(t) &= a_{21} * s_1(t) + a_{22} * s_2(t) + a_{23} * s_3(t) \\x_3(t) &= a_{31} * s_1(t) + a_{32} * s_2(t) + a_{33} * s_3(t)\end{aligned}$$

The above can be written in a matrix form as below.

$$\mathbf{x}(t) = \mathbf{As}(t)$$

In the general case of the ICA, the signals are not time dependent, so the time index can be dropped, giving the following form.

$$\mathbf{x} = \mathbf{As}$$

The original signals \mathbf{s} are unknown, since they cannot be observed independently, and they are what the ICA is trying to find. That is why they are defined as latent variables in the equation. The problem is that the mixing matrix \mathbf{A} , containing the signal coefficients (weights), is also unknown. For simplicity, it is assumed that the matrix \mathbf{A} is squared and invertible. Since both the mixing matrix \mathbf{A} and the source signals \mathbf{s} are both unknown, they should be estimated only by the observed mixed signals \mathbf{x} .

ICA tries to recover the initial signals given only the observed mixtures. The goal is to find an unmixing matrix \mathbf{W} such that $\mathbf{W} = \mathbf{A}^{-1}$. Since it is impossible to find the exact matrix \mathbf{W} , ICA tries to find as better approximation as possible for the unmixing matrix, and consequently for the original signals, and thus the following holds for the estimated independent components $\tilde{\mathbf{s}}$.

$$\tilde{\mathbf{s}} = \mathbf{Wx}$$

The schematic representation of ICA is presented in figure 1 below.

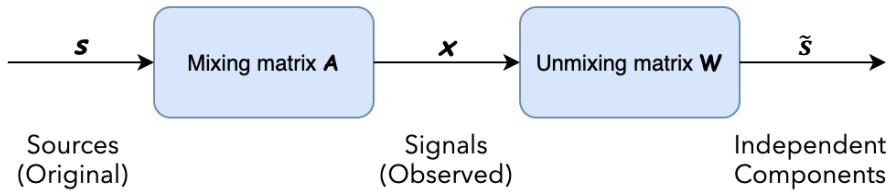


Figure 1: Schematic representation of ICA

2.1.1 ICA Restrictions

One way to recover the original signals from the mixtures, is to assume that the source signals are statistically independent. This straightforward assumption most of the times is realistic, since the signals are emitted by different (independent) sources, and provides a simple solution to the problem of the signal separation. This is one key assumption for the source signals.

Another major assumption which is made for the source signals is that they have a non-gaussian distribution. It is a fact that gaussian distributions are very simple, since the third and higher order cumulants are zero. But since the higher order cumulants are vital for the ICA, gaussian distribution cannot be used. Another way to show that gaussian variables are not suitable for solving the ICA, is to see what happens to the model if they were gaussian. Since the mixing matrix is orthogonal, then the mixtures (observed signals) would be also gaussian, and the joint probability density function would be symmetric, making it impossible to estimate the mixing matrix \mathbf{A} . Therefore non-gaussian distributions should be used, so the mixing matrix \mathbf{A} can be estimated. While it is known that non-gaussian

distributions should be used, these distributions are still unknown and they need to be estimated by the ICA.

2.1.2 Ambiguities of ICA

While ICA can be used to recover the original signals, it still suffers from some non-negligible ambiguities. The first one is that it cannot define the magnitudes of the independent components. This is due to the fact that both mixing matrix \mathbf{A} and signals \mathbf{s} are unknown, and thus they can be multiplied and divided respectively by the same scalar and give the same result for the mixtures. A simple solution to this ambiguity is to restrict the variance of the independent components \mathbf{s} to be equal to 1, making the matrix \mathbf{A} to be defined uniquely. The second ambiguity, is the sign of the independent components, which can be the opposite of the original source signals. Finally, the third ambiguity and the most important one is that the ICA cannot assign which calculated independent component corresponds with which source signal. That happens because each time ICA is initialized randomly and then each time it estimates the independent components in a different order depending on the initialization. The above ambiguities can be formulated using the following equation.

$$\tilde{\mathbf{s}} = \mathbf{D}\mathbf{P}\mathbf{s}$$

where $\tilde{\mathbf{s}}$ is the independent components executed by the ICA model, \mathbf{D} is a diagonal matrix that contains nonzero elements and it express the first two arbitrariness. In fact, the elements of the matrix \mathbf{D} give the magnitude ratio between the independent components recovered by the ICA and the original signals. The matrix \mathbf{P} is a permutation matrix that contains only one unit element in each row, while the other elements are very close to zero, and this matrix expresses the third ambiguity described above. Since ICA cannot correspond the initial signals to the independent components, the matrix \mathbf{P} is used to show this ambiguity.

2.1.3 Preprocessing

The first thing that should be done, before starting solving for the ICA, is to center the observed data \mathbf{x} . This step is done in order to reduce the calculations and make the algorithms simpler. Centering the observed data is done by just removing the mean, so the mixture signals have zero mean. This will make also the independent components \mathbf{s} to have zero mean. Once the independent components are calculated the mean multiplied by the matrix \mathbf{W} should be added to them.

As mentioned above, solving ICA is a challenging problem since it needs to calculate the two unknown multivariate variables, mixing matrix \mathbf{A} and original signals \mathbf{s} , while knowing only the mixture signals \mathbf{x} . The problem can be separated in more minor problems. The calculation of the original signals is left aside for the moment, and the focus is on the calculation of the mixing matrix \mathbf{A} . This matrix is assumed to be squared and invertible at the moment, and thus it can be divided into smaller parts. A way to do that is to use the Singular Value Decomposition (SVD) to divide the matrix \mathbf{A} into three smaller matrices as shown below, where each one of them performs an operation on the data.

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \Rightarrow \mathbf{W} = \mathbf{A}^{-1} = \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^T$$

The first step made for the calculation of the unmixing matrix \mathbf{W} is the calculation of the covariance matrix. This is a simple step, since the covariance matrix can be calculated as the expected value of the outer product of the data, resulting to a $d \times d$ matrix which contains all the correlations between the data. Considering that the rotation matrices \mathbf{V} and \mathbf{U} are orthogonal and making the assumption that the independent components \mathbf{s} have an identity covariance matrix, it can be shown that the covariance matrix is given by the follow.

$$\begin{aligned} E[\mathbf{x}\mathbf{x}^T] &= E[(\mathbf{As})(\mathbf{As})^T] = E[(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{s})(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{s})^T] \\ &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T E[\mathbf{s}\mathbf{s}^T] \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T \end{aligned}$$

Observing the above result, one can see that the covariance of the data is independent of the rotation matrix \mathbf{V} and the original signals \mathbf{s} . In addition, since the matrix $\boldsymbol{\Sigma}$ is diagonal and thus a symmetric matrix, it can be diagonalized by its eigenvectors. So the above result can be written as follow.

$$E[\mathbf{x}\mathbf{x}^T] = \mathbf{E}\mathbf{D}\mathbf{E}^T$$

where \mathbf{D} is a diagonal matrix containing the eigenvalues, and \mathbf{E} is an orthogonal matrix containing the eigenvectors of the covariance of the data. This also gives a unique solution to the problem. So, this gives the unmixing matrix \mathbf{W} written as below.

$$\mathbf{W} = \mathbf{V}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T$$

where \mathbf{D} and \mathbf{E} contain the eigenvalues and the eigenvectors respectively of the covariance of the data, resulting to a form where only the rotation matrix \mathbf{V} is unknown.

The first step that is needed in order to be able to calculate the unmixing matrix \mathbf{W} and the independent components is to whiten the data, or in other words to remove all the linear dependencies from the data and then normalize the variance along all dimensions. The procedure of removing the dependencies from the data is called Principal Component Analysis (PCA) and it is operated by the matrix \mathbf{E} which contains all the eigenvectors of the covariance of the data. When the data is projected on these eigenvectors, the linear correlations are removed. The normalization of the variance is done by the $\mathbf{D}^{-\frac{1}{2}}$. Therefore, prewhitening the data simplifies the problem reducing it to just finding the rotation matrix \mathbf{V} . This can be seen more clear below, where the operation of whiten is done on the observed data \mathbf{x} .

$$\mathbf{x}_w = \left(\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T\right)\mathbf{x}$$

Therefore the calculation of the independent components is reduced to the following.

$$\mathbf{s} = \left(\mathbf{V}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T\right)\mathbf{x} \Rightarrow \mathbf{s}_w = \mathbf{V}\mathbf{x}_w$$

2.1.4 Solving ICA

Observing the last equation from above, one can see that in order to find the independent components, only the rotation matrix \mathbf{V} is left to be calculated. To do that, the statistics of independence of the components should be considered. The ICA problem has been reduced to just finding a rotation \mathbf{V} such that \mathbf{s}_w is statistically independent. However, it is often not possible to find such a matrix. So, the ICA problem can be seen as the problem of finding a rotation \mathbf{V} such that \mathbf{s}_w is as much as statistically independent as possible.

In order to do that, the matrix \mathbf{W} should be a matrix that maximizes the non-Gaussianity of \mathbf{Wx} . There are several ways to do that, as they will be presented below.

a. Kurtosis

As mentioned above, higher order statistics (3rd order and above) can be used as a measure for non-Gaussianity. Kurtosis is such a statistic which represents a fourth order cumulant. It is given by the following equation

$$kurt(\mathbf{y}) = E(\mathbf{y}^4) - 3[E(\mathbf{y}^2)]^2 = E(\mathbf{y}^4) - 3$$

Since the data are whitened, they have zero mean and unit variance giving the above result. Kurtosis is zero for a gaussian random variable and for the most non-gaussian ones it is either positive or negative. The absolute value of the kurtosis can be seemed as a measure of non-Gaussianity, so the goal is to maximize the absolute value of the kurtosis of $\mathbf{w}^T \mathbf{x}$.

In practice, a gradient algorithm can be used to obtain the maximum kurtosis. An arbitrary initial matrix \mathbf{W} is selected, and then, the direction that the absolute value of the kurtosis is growing most strongly is computed. Then, \mathbf{W} is moving in that direction, The computation of the gradient and matrix \mathbf{W} is done until convergence.

While using kurtosis to maximize the non-Gaussianity is a very simple method, however, it lacks of robustness, since it is very sensitive to outliers (Huber, 1985).

b. Negentropy

Another measure of non-Gaussianity, which is also more robust than the kurtosis is the negentropy. Entropy of a random variable is defined as the uncertainty of an event, so in order to maximize the entropy, the randomness of the event should be as highest as possible. Gaussian distribution is the least structured of all distributions and it has the largest entropy. So, the negentropy can be defined as a measure which is zero for gaussian variables. So, entropy can be used in order to measure the non-Gaussianity, and it is formulated as below.

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$$

where \mathbf{y}_{gauss} and \mathbf{y} have the same mean and covariance matrix. Negentropy is a very good measure of non-Gaussianity, however it is very computationally expensive. For this reason,

it is important that it should be approximated, in order to need less computational time. It can be very well approximated by the following.

$$J(\mathbf{y}) \propto [E\{G(\mathbf{y})\} - E\{G(\mathbf{v})\}]^2$$

where \mathbf{v} is a gaussian variable and \mathbf{y} a non-Gaussian variable. They both have zero mean and unit variance. G is a nonquadratic function and should be chosen carefully, so it gives good approximations of negentropy. It should also not grow too fast, so it can be more robust than kurtosis. The most widespread functions used for G are the following.

$$\begin{aligned} G_1(\mathbf{y}) &= \frac{1}{a_1} \log \cosh a_1 \mathbf{y} \quad \text{where } 1 \leq a_1 \leq 2, \text{ often equal to 1} \\ G_2(\mathbf{y}) &= -\exp\left(-\frac{\mathbf{y}^2}{2}\right) \end{aligned}$$

With the above approximation, the estimation of the negentropy becomes easier and less computationally expensive. The most significant improvement is that it is a more robust measure than kurtosis.

2.1.5 ICA Algorithms

a. FastICA

FastICA (Hyvärinen & Oja, 2000) is a learning algorithm that is used in order to maximize the non-Gaussianity of the projection of $\mathbf{w}^T \mathbf{x}$, and uses approximation of negentropy described above as a measure of non-Gaussianity. The first step in the algorithm is doing the preprocessing steps, eg. centering the data, removing their mean, and whiten them. The second step, is to select the number of independent components needed from the algorithm to estimate. After that, all vectors \mathbf{w}_i of the matrix \mathbf{W} should be randomly initialized.

The matrix \mathbf{W} should be orthogonalized in order to prevent different vectors of the matrix to converge to the same maxima, or in other words, for whitened data, to decorrelate them. There are two ways of orthogonalization (Hyvärinen et al., 2001; Tharwat, 2018). Deflationary (or sequential) orthogonalization estimates the independent components one by one, while symmetric one estimates them in parallel. The symmetric orthogonalization is used since it has the ability to estimate the vectors \mathbf{w}_i in parallel, making the procedure much faster. Another major advantage of the parallel computation of the weight vectors is that since the calculation is done in parallel, the errors are not extended from one weight vector to the other. The symmetric orthogonalization can be achieved by the following

$$\mathbf{W} \leftarrow (\mathbf{W} \mathbf{W}^T)^{-\frac{1}{2}} \mathbf{W}$$

The next step of the FastICA algorithm is the change of each weight vector of the matrix \mathbf{W} as shown below.

$$\mathbf{w}_i \leftarrow E\{\mathbf{x}G(\mathbf{w}_i^T \mathbf{x})\} - E\{G'(\mathbf{w}_i^T \mathbf{x})\} \mathbf{w}_i$$

Where G can be any of the two functions mentioned in the previous part used for the approximation of the negentropy. The next step is to symmetrically orthogonalize the new matrix \mathbf{W} using the new weight vectors calculated, using the same method as above.

The final step of the algorithm, is to check if the weight vectors converge. The weight vectors converge if the dot product of the old and new values of \mathbf{w}_i is very close to 1, or in other words, if they point to the same direction. If all the weight vectors converge then the algorithm ends. If not, new weight vectors are calculated as shown above, and the same procedure is followed.

FastICA is known as the fastest algorithm used for ICA. As mentioned, it is distributed since it does the calculations in parallel. While almost all ICA methods converge linear, FastICA converges cubic, making it way much faster than the others. It is also very computationally simple, not involving any learning parameters. Finally, it is also a generalized algorithm, since it finds the independent components of any non-Gaussian distribution.

b. ProDenICA

Product Density Independent Component Analysis (ProDenICA) (Hastie, Tibshirani, & Friedman, 2009; Tibshirani & Hastie, 2003) uses semi-parametric density estimate based on cubic splines. Independent components as their name indicates have by definition a joint product density

$$f_{\mathbf{s}}(\mathbf{s}) = \prod_{j=1}^p f_j(s_j)$$

Each density f_j is represented by a tilted Gaussian density (Efron & Tibshirani, 1996) as

$$f_j(s_j) = \phi(s_j) e^{g_j(s_j)}$$

where ϕ is the standard normal density and g is a smooth function, estimated with cubic splines and satisfies the normalization condition so that the integral of f_j is equal to 1. The goal is to maximize the log-likelihood of the data, but since it contains too many parameters, a regularized version is used. Two penalties are introduced. The first is the latter constraint explained, and the second one is just a roughness penalty. The objective function that needs to be maximized is the following.

$$\sum_{j=1}^p \left[\frac{1}{N} \sum_{i=1}^N [\log\{\phi(a_j^T x_i)\} + g_j(a_j^T x_i)] - \int \phi(t) e^{g_j(t)} dt - \lambda_j \int \{g_j''(t)\}^2 dt \right]$$

Increasing the parameters λ_j makes the solutions to approach the standard Gaussian ϕ . Some approximations are made for the first integral. The weights a_j are estimated using the fixed point algorithm used in FastICA. For comparison of the original matrix \mathbf{A} and the orthogonal matrix estimated by the algorithm, the Amari metric is used (Amari, 1998).

It is a fact the ProDenICA adapts to the independent components distributions with the minimal dependencies. It also outperforms the FastICA algorithm in terms of the Amari metric (Tibshirani & Hastie, 2003), meaning that it converges better to the initial mixing matrix, giving more accurate independent components. However, it is way much computationally expensive than FastICA algorithm, making it essential for the user to make a trade-off between a very fast but not that much accurate algorithm, and a computationally expensive, slower but more accurate one.

2.2 Surface Electromyography (sEMG)

Electromyography is a process that measures the electrical activity of the muscles when muscle fibers contract. These signals are measured by electromyograph and the generated signal is called an electromyogram.

There are two ways to measure electrical activity. The first one is using needles to be attached to the muscles of the patient and is called Intramuscular Electromyography. This process is painful for the patient and requires a significant amount of time. The second approach to measure the muscle electrical activity is non-invasive and painless. Electrodes are affixed on the surface of the skin above muscles. Electromyography signals are measured from these electrodes on the skin providing a quicker and easier way than Intramuscular EMG to measure these signals. The latter procedure is called Surface Electromyography (sEMG).

Electromyograms have a variety of applications in biomedicine. Some of the applications are detection of medical abnormalities and determination of human and animal movements. EMG is also used in gesture recognition with extensions to command and control of prosthetic parts.

On figure 2a, one can see how the surface electrodes are placed on the skin surface, in order to measure the sEMG, while on figure 2b it is presented how these signals look like.

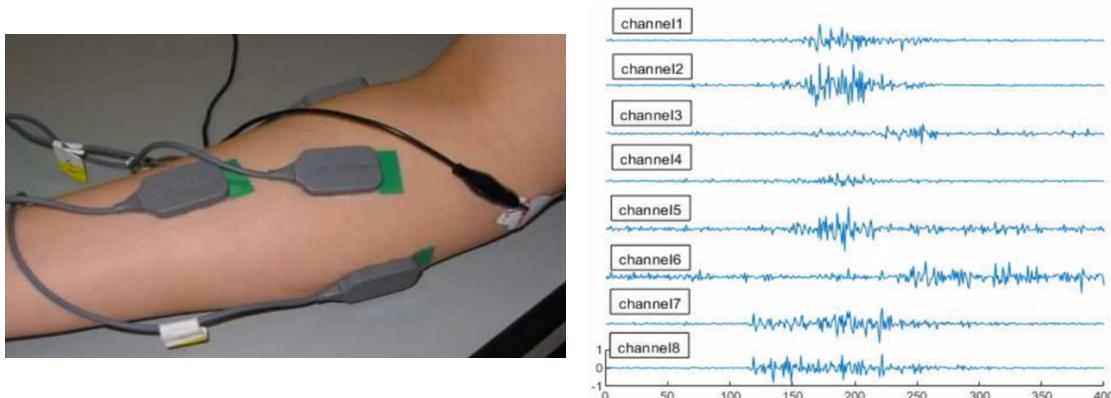


Figure 2: a) Electrodes are placed on the skin surface to measure the muscles' activity. (G. Naik, Kumar, Arjunan, & Palaniswami, 2008) b) Measured signals are transferred to a PC and this is the output on the screen. (Zhang, Yang, Qian, & Zhang, 2019)

2.2.1 Limitations

While sEMG signals have numerous applications, in some cases they are not very reliable since they may be affected by numerous circumstances. Some faulty measurements may be a result of the electrodes' properties or the recording method. Another factor that influences the sEMG signals is due to the fact that humans are very complex creatures. Consequently, muscle anatomy and physiology can lead to unreliable results, as some of these characteristics are different for each person. Besides some nerve disorders or neuromuscular problems may lead to bad measurements.

One of the most important drawbacks of sEMG comes from the fact that electrodes record noise and some signals originating from other parts of the body. When a gesture is performed, many muscles are activated. So, electrodes record signals from other neighboring muscles too, and not only from the one that they are supposed to. That is called cross-talk and may lead to bad and unreliable measurements of sEMG signals. Moreover, electrocardiography (ECG) artifacts may contribute to the electrode recording.

2.2.2 sEMG and ICA

All these limitations stated above, introduce some difficulty on the identification of the source of the signal. Many techniques are used to remove cross-talk and artifacts, and to reduce noise. One of the most widespread techniques in the research is ICA. ICA is used to isolate and remove noise and artifacts as well as the muscles' cross-talk. These assets can be exploited to make ICA an advantageous technique for sEMG signal separation.

As described in the previous chapter, ICA requires very little information about the mixed signals to be separated and of the sources. Besides, a signal coming from one muscle can be assumed to be statistically independent of signals originating from the neighboring muscles or noise and ECG artifacts. Moreover, signals measured by electrodes have non-Gaussian distribution (G. R. Naik, K. Singh, & Palaniswami, 2006). Based on the above, there is a confidence that ICA performs nicely on EMG signals, and that is something that will be examined in chapter 4.

3. Simulation

In this chapter some simulated data will be used in order to see what ICA achieves. It is important to use simulated data, so the evaluation will be easy, since the source signals will be known, making it a supervised learning problem. The fastICA algorithm will be used to estimate the independent components using only the mixtures and then the results will be evaluated by comparing them with the known source signals. Finally, a comparison between fastICA and ProDenICA is presented, and the merits and drawbacks of each of the two ICA algorithms are stated.

3.1 First steps

First, two independent variables s_1 and s_2 that follow uniform distribution will be generated. Variable independence is important in this procedure since ICA has a prerequisite that the components are independent of each other in order to perform well. Then a random mixture matrix \mathbf{A} , which is generated using values that follow uniform distribution, is multiplied with the initial variables \mathbf{s} and it gives two new variables that can be defined as mixtures. These mixtures are the signals that are truly observed by the sensors that are in different positions and receive both of the original signals but with different weights. In figure 3a, one can see the initial variables \mathbf{s} in the range [-0.5,0.5], and in figure 3b the resulting mixture signals after these initial variables \mathbf{s} being multiplied with the random matrix \mathbf{A} .

As mentioned above, mixture signals \mathbf{x} are the ones that are observed and measured by the sensors. In a real case, original signals are unknown, however, in this simulation case, it is assumed that they are known in order to validate the results in the end. Given only the mixture signals \mathbf{x} , the aim is to calculate the independent components \mathbf{s} and the matrix \mathbf{A} that was used to mix the initial signals. A way to do that is to introduce Independent Component Analysis (ICA) in the simulation. As stated in a previous section, ICA is used to unmix the observed mixed signals and give the best estimation for the original signals. A simple and fast algorithm that is described above will try to recover the initial signals given only the mixtures.

FastICA (section 2.1.5.b) is used on the mixed signals. ICA tries to calculate the matrix used to mix the signals and then uses its inverse to estimate the initial signals. The results of the fastICA algorithm are presented in figure 3c. At a first glance, it seems that the independent components calculated are not the same as the initial ones, and not even close to them. In

the following sections of this chapter, it will be explained why even if the recovered signals do not appear to be close to the initial ones, they really are.

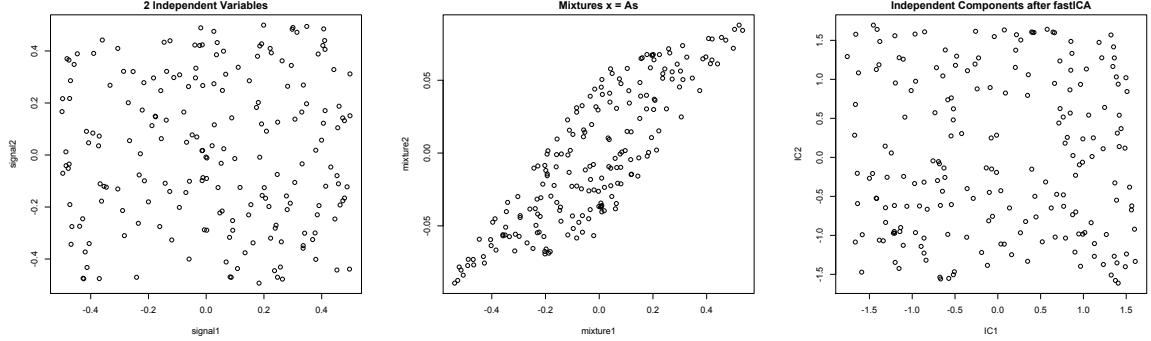


Figure 3: a) Two independent variables follow uniform distribution. b) Observed signals after multiplication with the mixing matrix \mathbf{A} . c) The fastICA algorithm give these independent components.

3.2 Time Series Signals

In the previous experiment, it was unclear if the fastICA achieved recovering the initial signals. In this section, two independent variables following a uniform distribution will be generated again, but they will be assumed to be time series signals, like the real case scenarios. Each signal has a duration of 200 observations and some noise is added to seem more realistic. Then, a random mixing matrix \mathbf{A} is used to simulate the mixing procedure, resulting in new mixture signals \mathbf{x} . Initial and mixed signals are illustrated in figures 4a and 4b respectively.

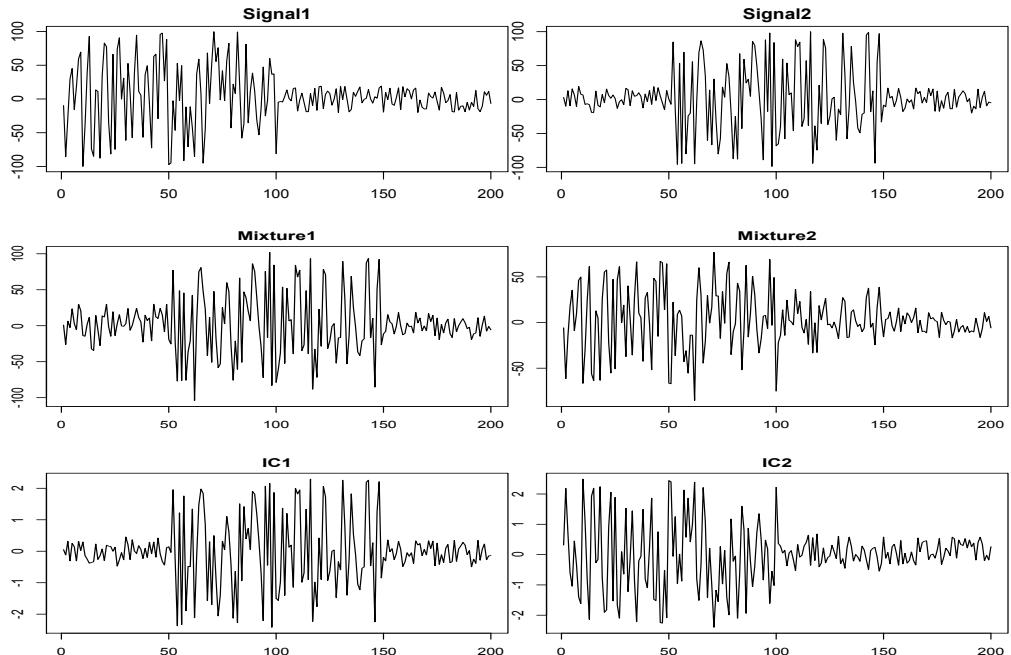


Figure 4: a) Source time series signals. b) Mixtures after multiplication of source signals with mixing matrix \mathbf{A} . c) Recovered signals (independent components) using the fastICA algorithm

Then, the fastICA algorithm is used to recover the initial signals, given only the mixed ones. The recovered signals are presented in figure 4c. The ambiguities that ICA introduces are

illustrated here. The first one is that there is no clear correspondence between the initial signals and the independent components. The second independent component estimated by the algorithm corresponds to the first generated signal. Another ambiguity is observed by looking at the amplitudes of the initial and estimated signals, which do not have the same magnitudes. One other major ambiguity of the ICA is the sign of the signal. While the first independent component IC_1 seems to have the same sign as its respective signal s_2 , that does not hold for the second one (IC_2), which seems to have a different sign than the initial signal s_1 . Assuming that fastICA recovers the signals well, the above ambiguities can be formulated as following.

$$\begin{aligned} IC_1 &\approx c_1 * s_2 \\ IC_2 &\approx -c_2 * s_1 \end{aligned}$$

where c_1 and c_2 are constants used to explain the amplitude changes.

When there are only two estimated independent components, someone can observe, compare and decide if the ICA managed to recover the source signals. In this case, observing and comparing the signals and the independent components as well as taking into account the ambiguities explained, one could say that the level of signal recovery is good. However, comparing the signals and independent components is not easy, especially when more signals should be recovered. Therefore, it is important that the level of signal recovery be quantified, and this is presented in the next section, where more signals are used, making it difficult to match the source signals and independent components manually.

3.3 More Time Series Signals

In the previous section two mixtures were used as an input and the ICA algorithm estimated the two independent components. As shown, it was possible to compare and correspond them manually to the source signals, solving the ordering ambiguity. In this section, four independent signals are used, and the difficulties of matching the signals manually becomes more clear. Again, all initial time series signals follow a uniform distribution, and some noise is added to each of them as presented in figure 5a. A random mixing matrix is used to mix the signals and the result is shown in figure 5b. FastICA algorithm is used and the estimated independent components are shown in figure 5c.

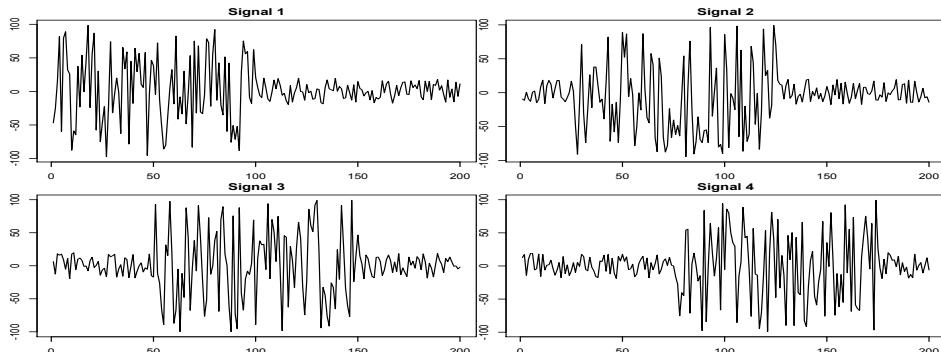
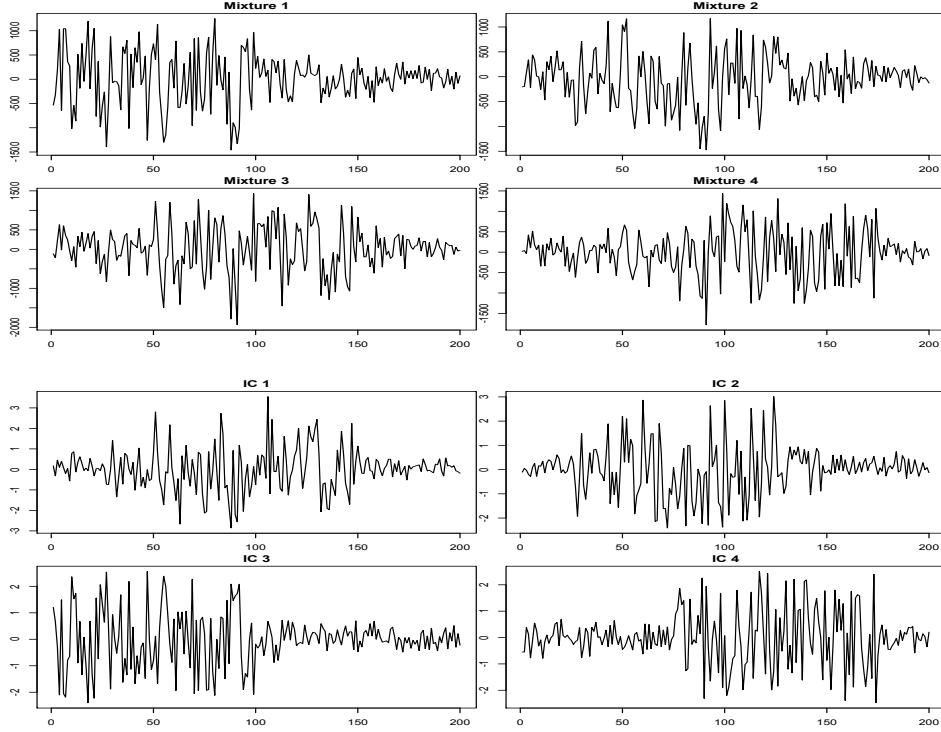


Figure 5: a) Source time series signals



*Figure 5: b) Mixtures after multiplication of source signals with mixing matrix A .
c) Recovered signals (independent components) using the fastICA algorithm*

Considering the ambiguities that were analysed previously and observing the initial signals and the independent components calculated by the algorithm, it is not possible to correspond them. Having more signals to compare, it is a difficult job to do manually, like before. Moreover, observing the recovered signals and comparing them to the source signals, it can be said that they do not look exactly the same, making the matching process even harder. As explained in the previous section, the level of signal recovery should be quantified, as well. Thus, a new tool should be introduced to give a clear summary of the level of signal recovery as well as of the signal matching process. This new tool is the correlation matrix (Tesfayesus, Yoo, Moffitt, & Durand, 2004). Since the initial signals are known in this simulation, they can be used to be corresponded and compared with the estimated ones. The correlation matrix between the initial and estimated signals is presented below. The absolute values of the correlations are used, so the sign ambiguity does not play any role.

	IC1	IC2	IC3	IC4
signal1	0.014	0.101	0.986	0.133
signal2	0.421	0.907	0.031	0.012
signal3	0.838	0.538	0.079	0.036
signal4	0.050	0.046	0.181	0.981

Table 1: Correlation Matrix of source signals with independent components estimated using the fastICA algorithm

Using the above method, the ordering ambiguity is solved and the level of signal recovery is quantified. The source and recovered signals with the highest correlation coefficients are corresponded as highlighted in the matrix above. In addition, observing just the correlation

coefficients between the source and recovered signals is enough to quantify the level of signal recovery during the ICA. From the correlation matrix above, it can be seen that some signals were well recovered, while some others not that much.

The above procedure can also be followed in the case that there are more mixture signals than the initial ones and the mixing matrix is not invertible (under-complete problem). This happens when the number of sensors is greater than the number of initial signals, so more mixtures are measured. ICA algorithm will give more independent components but some of them may be white noise. Correlation matrix will be used again, but in this case, some columns will have very low correlation coefficients with the initial signals, and thus, they will not correspond to any of them. These components are not useful and can be removed (Tesfayesus et al., 2004). Another solution is to delete some mixtures using a dimensionality reduction technique such as Principal Component Analysis (PCA) to decrease the number of mixtures (Tharwat, 2018).

3.4 fastICA vs ProDenICA

As observed in table 1, the fastICA algorithm recovered some of the signals accurately, while other signals less accurately. A more complex algorithm that was described in a previous chapter will be used in order to check if it can achieve better results than the fastICA algorithm. This algorithm is called ProDenICA (section 2.1.5.b). The same initial and mixed signals are used. These mixed signals are used as input to the ProDenICA algorithm. The independent components calculated by the ProDenICA algorithm are shown in figure 6 below and they appear to be closer to the initial signals than the ones that were calculated using the fastICA algorithm.

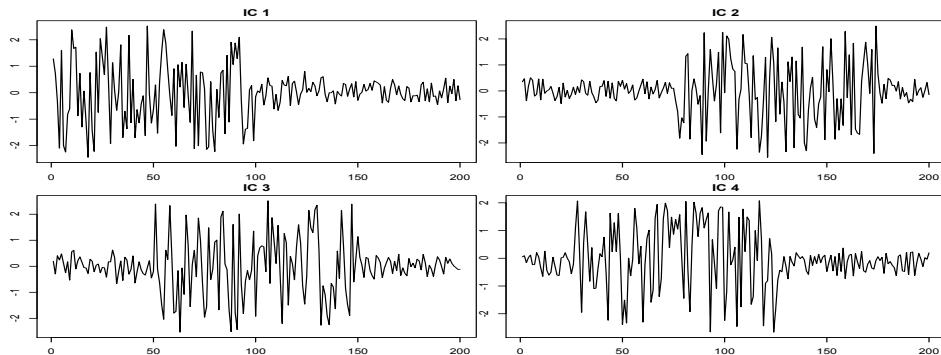


Figure 6: Independent components estimated by the ProDenICA algorithm for mixtures shown in figure 4b).

A correlation matrix will be used again to evaluate the calculated independent components. It seems that this algorithm performs better, but it still cannot overcome the ICA ambiguities. So, the correlation matrix is used to correspond the independent components to the initial signals and show the correlation between them. Again, the highest correlation is assumed to give the correspondence, and it is highlighted in table 2.

	IC1	IC2	IC3	IC4
signal1	0.991	0.016	0.031	0.128
signal2	0.009	0.079	0.020	0.997
signal3	0.053	0.011	0.992	0.114
signal4	0.029	0.999	0.044	0.011

Table 2: Correlation Matrix of source signals with independent components estimated using the ProDenICA algorithm

The correlation matrix above verifies the intuition that was gotten by observing the plots, that the ProDenICA performs very well in this simulation. The correlation coefficient of each signal with the corresponding independent component is extremely high, near to 1, meaning that the ProDenICA algorithm achieves recovering the initial signals highly accurate.

While the ProDenICA algorithm appears to perform better than fastICA, it is important to investigate the time needed for these two algorithms to run and give the calculated independent components. It is known from the previous chapter that the ProDenICA algorithm is more complex than fastICA, but it is important to have some metric on the time taken by the algorithms to run. The average computational times for both algorithms after 20 repetitions are stated below.

$$\begin{aligned} \textbf{fastICA}: & 0.056 \text{ seconds} \\ \textbf{ProDenICA}: & 4.697 \text{ seconds} \end{aligned}$$

The difference in computational time is apparent. fastICA algorithm appears to be extremely fast, compared to the ProDenICA algorithm that it is much slower since it takes almost 5 seconds to calculate the independent components. While some may be more interested in the computational time of each of the two algorithms, some others may prefer the accuracy they provide. In terms of accuracy, as presented above, it is clear that ProDenICA performs better. However, this improvement in accuracy comes with a price. Depending on the application on which ICA is run, the appropriate algorithm should be selected. When signal decomposition is required to be done quickly, like in real-time applications, then the fastICA algorithm should be used. On the other hand if the application does not require fast computations, then ProDenICA has the edge, as it can be more accurate on signal recovering.

4. sEMG Dataset

4.1 Dataset Description

In the experiment (Lobov, Krilova, Kastalskiy, Kazantsev, & V.A., 2019; Makarov, Lobov, Krilova, Kastalskiy, & Kazantsev, 2018) 36 subjects participated. Each of them performed two series of static hand gestures, one after the other. In most of the cases, each of the series consists of six gestures, while a small minority of subjects performed seven gestures. Hand gestures were operated for 3 seconds with a pause between them of a duration of 3 seconds.

For the pattern recordings, a MYO Thalmic bracelet (Thalmic Labs) was worn by the subjects on the forearm. The MYO bracelet is equipped with eight equidistant sensors as illustrated in figure 7a below. Sensors simultaneously acquire myographic signals and these signals are sent via Bluetooth to a PC. In figure 7b, one can see the signal output of each sensor on the screen during the gesture.



Figure 7: a) MYO Thalmic bracelet. b) Signal output of each sensor on the screen during a hand gesture.

The dataset contains 36 raw sEMG signals, one for each subject participated in the experiment. Each file consists of ten columns. The first one is the time, the following eight columns are the eight EMG channels of MYO Thalmic bracelet, and the tenth column contains the gesture performed. There are eight different labels as stated below. Gesture 7 (extended palm) was performed by only two subjects. Each column contains at least 30.000 instances, with most of the subjects consisting of 40.000 - 50.000 instances.

-
- 0** - unmarked data
1 - hand at rest
2 - hand clenched in a fist
3 - wrist flexion
4 - wrist extension
5 - radial deviations
6 - ulnar deviations
7 - extended palm
-

ICA provides unsupervised learning to signal separation. The input of ICA is the sEMG signals. These signals contain noise, ECG artifacts, and signals originating from the neighboring muscles. There is no information about the signal source and therefore, it is not attainable to evaluate the quality of the separation by comparing the independent components to the original signals. Therefore, the system's accuracy to identify the hand gestures correctly is used to evaluate the model.

4.2 Dataset Analysis

4.2.1 Exploration

The first step that should be done is dataset exploration. As described above, the data for each subject consists of ten columns. The first one is the time passed from the beginning of the experiment for each subject, the next eight columns contain the data for raw sEMG signals measured by the MYO bracelet, and the last one contains the gesture label at each time unit. On figure 8, one can see the raw sEMG signals for a random subject. Each plot corresponds to one of the eight MYO bracelet's sensors.

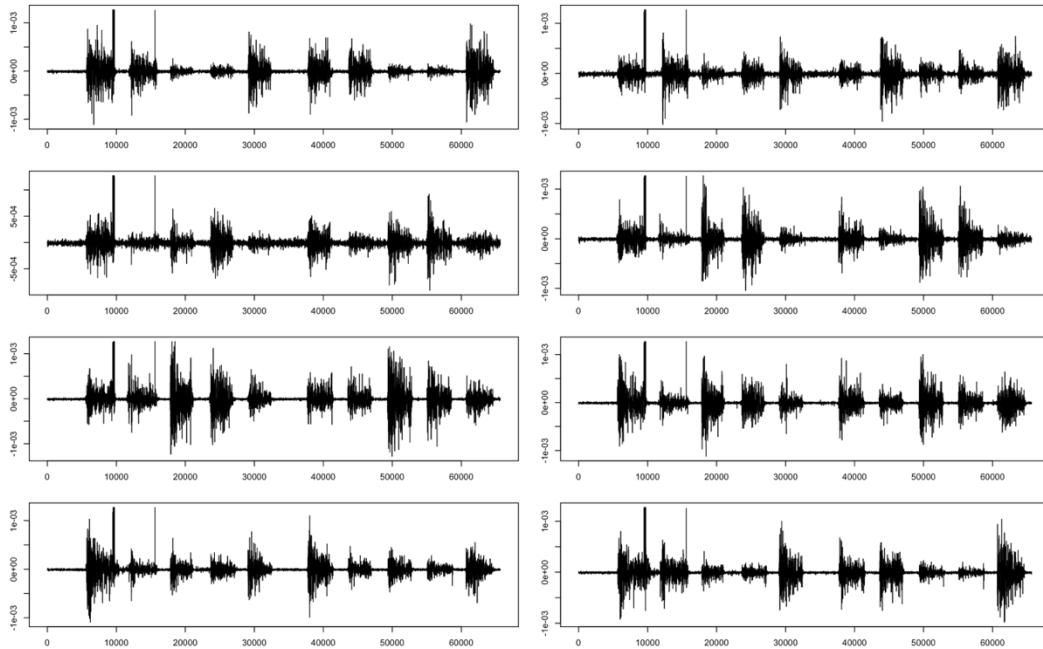


Figure 8: Raw sEMG signals measured by the eight MYO bracelet's sensors.

Raw sEMG signals cannot be used blindly, and thus, some preprocessing steps are required to bring the data in a usable form. Each raw signal should be broken into pieces, where each piece will be a gesture. The gesture label is used and each raw signal is divided into 24-26 smaller signals (12-13 pauses between the gestures that occur 2 times each).

One step that needed to be done before proceeding further was to consider class labels 0 and 1 as a unitary class. As mentioned in the previous section, class 0 corresponds to unlabeled data, or in other words to the pauses that happened after each gesture, while class 1 tally to the gesture "hand at rest". These classes are identical, so it is reasonable to consider them as a single one.

The class distribution is presented in figure 9a. As anyone can observe, the class containing pauses and "hand at rest" gesture (class 1) has the vast majority of gestures in the dataset. Thus, it should be expected that results may be affected by this imbalance. On figure 9b all class labels that imply some gestures are combined in order to create a new one. Classes are more balanced in this case and this classification problem will be examined as well. The training set consists of 75% of the data, while the testing set contains the rest 25%.

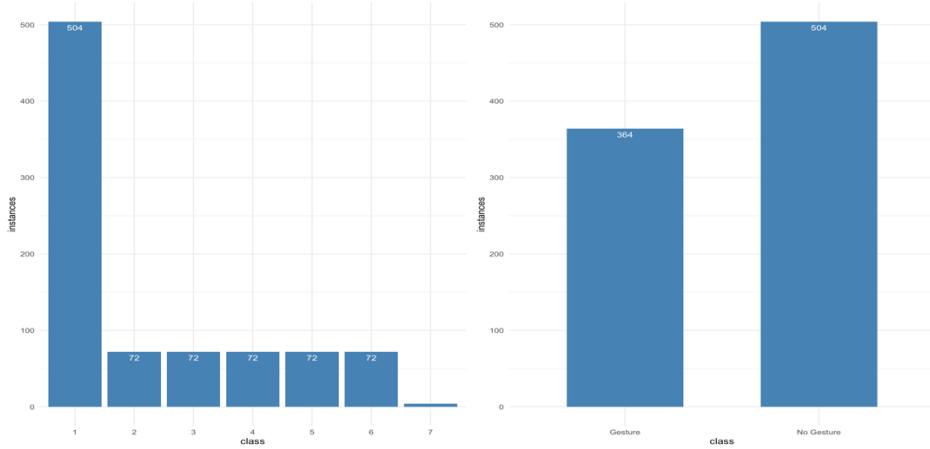


Figure 9: a) Original problem's class distribution. b) Binary problem's class distribution.

On figure 10 the gesture with label 2 (the first active gesture from raw signals - hand clenched in a fist) is illustrated. The problem discussed in the previous chapters is obvious. Sensors record signals originating from other parts of the body, as well as from other neighbouring muscles. While for a certain gesture only some of the muscles are activated, the recording signals indicate that more muscles are activated giving very identical measured signals. This is clearly something that should be investigated since for various gestures performed, different muscles are activated.

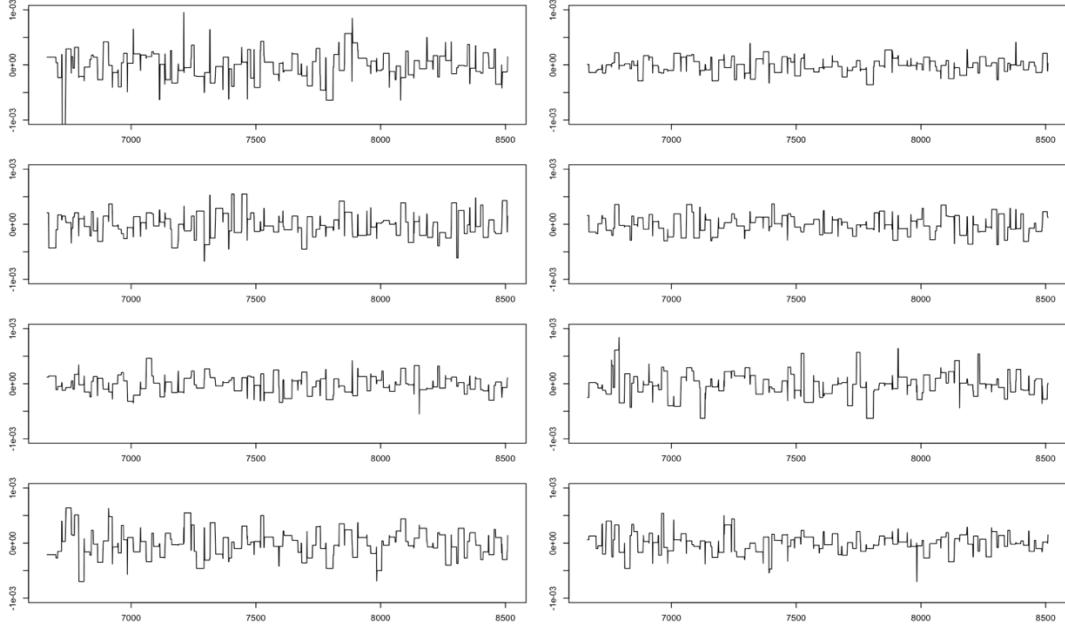


Figure 10: "Hand clenched in a fist" gesture signals on eight channels.

There are various ways to process a time-series signal. The most widely used methods in the time domain through the research (Djuwari, Kumar, Arjunan, & Naik, 2007; Garcia & Vieira, 2011; G. Naik et al., 2008; G. R. Naik et al., 2006) are Root Mean Squares (RMS) and Mean Absolute Values (MAV). The former represents the mean power of an sEMG signal and reflects the muscle activity, while the latter describes the average of the absolute values of the amplitude of an sEMG signal, and provides information about the muscle contraction level. They are given by the following formulas

$$RMS = \sqrt{\frac{1}{N} \sum_{k=1}^N s_k^2} \quad MAV = \frac{1}{N} \sum_{k=1}^N |s_k|$$

where N is the duration of the signal in time units, and s_k is the measurement of the signal at time k . In the following analysis, only RMS is used.

4.2.2 Baseline

The question following the previous section is if using only the mixtures is enough to classify the gestures correctly. Considering that sensors measure almost identical signals as presented in figures 8 and 10 then the average signal can be used for the classification task.

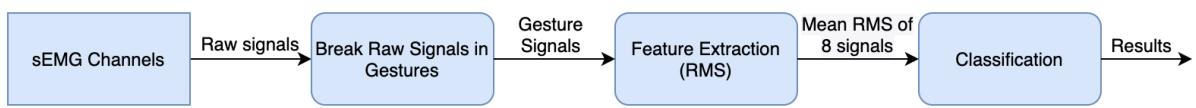


Figure 11: Baseline Method Diagram

Using the mean RMS of the 8 channels and using a multinomial (multiclass) logistic regression classifier, gives the confusion matrix presented below. The accuracy achieved

by the classifier is 0.507, however, this accuracy is a result of the dataset imbalance as explained above. The classifier performs very badly, as it classifies all gesture signals as class 1. This can also be seen in confusion matrix 1b, where again classifier cannot distinguish when there is a gesture or not. Although the accuracy seems to be acceptable as a baseline, the true accuracy is much less due to the high proportion the gesture 1 has in the dataset. To investigate how raw gesture signals are truly classified, majority class 1 will be removed. On confusion matrix 1c, one can see the gestures' classification that gives an accuracy of 0.243. It can be observed that the vast majority of instances is classified as class 3 and 4, while no predictions for classes 2, 6 and 7 are made.

		True class						
Predicted class		1	2	3	4	5	6	7
	1	110	24	21	17	23	21	1
	2	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0

		True class						
Predicted			NG	G				
	NG		110	107				
	G		0	0				

		True class						
Predicted class		2	3	4	5	6	7	
	2	0	0	0	0	0	0	0
	3	9	9	2	9	8	0	0
	4	13	7	15	12	12	1	0
	5	2	5	0	2	1	0	0
	6	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0

Confusion Matrix 1: Raw signals - a) Original Classification Problem. b) Binary Classification Problem.
c) Original Classification Problem without "No Gesture".

4.2.3 fastICA

Raw sEMG signals contain noise, ECG artifacts, and signals originating from other neighbouring muscles. ICA is used to remove noise and ECG artifacts, as well as the muscles' cross-talk. The reasons why ICA can work on such problems were explained in the previous chapter. The fastICA algorithm which was analysed in section 2.1.5.a, is employed. The algorithm takes as an input the raw sEMG signals of each subject and after the analysis, it gives its best estimation about the initial signals. These estimations are called independent components and are presented in figure 12.

A first glance in figure 12 gives the intuition that fastICA works well since it is observed less noise and artifacts during some gestures. Comparing the results of the fastICA algorithm to the raw sEMG signals for the subject (figure 8), it can be seen that some noise and ECG artifacts in the form of spikes on the signal are removed. However, the most important result of fastICA is the fact that each independent component appears to be indeed independent of each other. This can be noticed if one takes a look at figure 8 and compares it to figure 12. While in figure 8 almost all the eight signals appear to be indistinguishable during some gestures, in figure 12 there is a discrimination of the signals taken from the eight sensors.

Although the improvement in the signal is apparent, ICA introduces some other problems. The first one is the change in the signal amplitude. Another ambiguity introduced by ICA is the change in the sign of some independent components compared to raw signals. However, these two ambiguities do not appear to be serious, as during the feature selection using RMS, the sign is negligible and amplitudes are proportional to each other. The third

and most important problem is the change in the order of independent components when compared to the eight sensors' signals. This is an unsupervised learning problem, and thus initial source signals cannot be used as it happened in the simulation, since they are unknown. Therefore, this ambiguity should be investigated more.

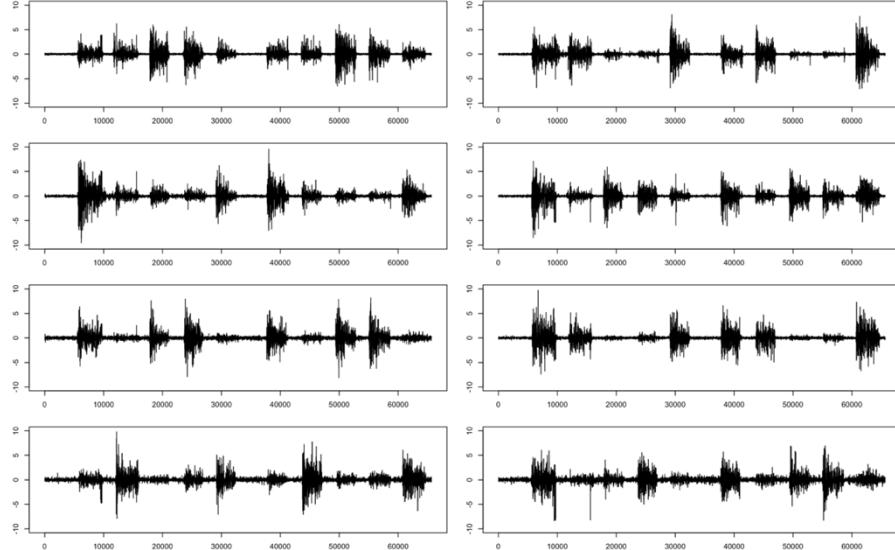


Figure 12: Independent Components estimated using the fastICA algorithm.

Before proceeding to further analysis, independent components should be broken into pieces, where each piece will be a certain gesture. Figure 13a demonstrates gesture 2, where some independent components have high amplitudes but some others have near to 0. It is examined if any of the estimated signals can be white noise, in order to be removed and then less but more important signals to be used. For a signal to be white noise, it should have an autocorrelation function equal to zero (or close enough to zero) for all lags except of lag 0. Here, it appears that none of the independent components can be assumed to be white noise since the autocorrelation function is still high after some lags (figure 13b). Thus all eight signals should be considered, avoiding losing important information.

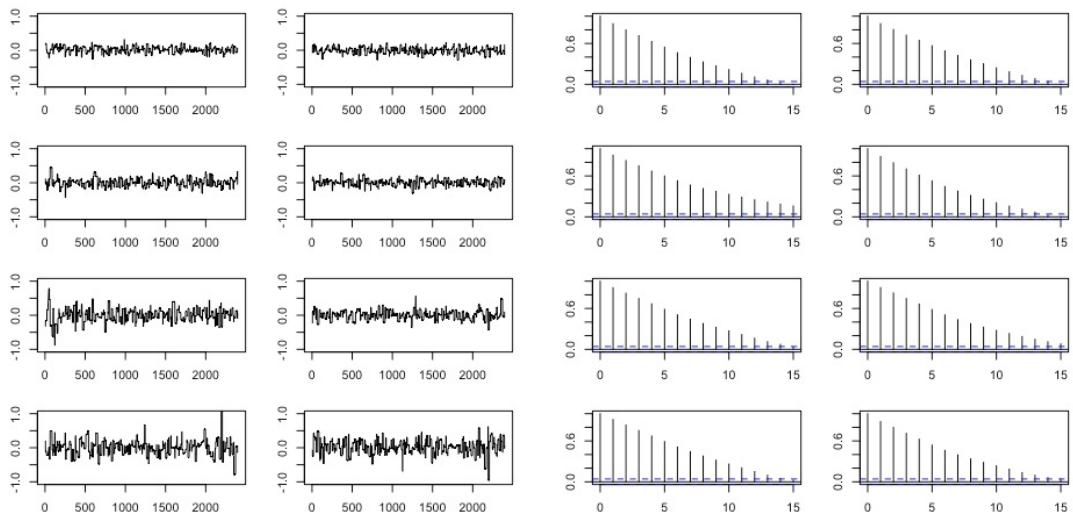


Figure 13: a) "Hand clenched in a fist" gesture after fastICA run.
b) Autocorrelation for eight gesture signals (after fastICA).

4.2.4 Bins

Each gesture is defined by the measured signal in both eight sensors. Contrary to raw sEMG signals, independent components resulting from ICA differ more in terms of amplitude with each other. The problem as explained above is that ICA introduces the ordering ambiguity, so it is not possible to correspond the raw signals to the independent components given by the fastICA algorithm.

The RMS values are calculated for each independent component, in order to get the muscle activity during the gesture. Since the order of the signals is unknown and for each subject is different, it is not reasonable for a classifier to have as input the signals in the given order. A simple assumption that can be made is that each gesture's amplitude is similar amongst all subjects participating in the experiment. So, the RMS values of the independent components can be used in order to produce something that can be used.

The first method we propose to solve the problem is the following. The idea is to design the features, so they represent the distribution of the RMS values. Lowest and highest RMS values amongst all gestures from all subjects were taken and used to set the lower and upper bounds. Ten bins were created in this range. For instance, if the lowest and highest values are 0 and 3 respectively, then the ten bins are [0, 0.3], [0.3, 0.6] ... [2.7, 3.0]. Then, it is checked in which bins the RMS values lie for each gesture, and the new features are created and are used as the classifier input. An example is presented in figure 14.

	RMS Values	0.4	2.4	2.0	2.3	0.8	1.6	1.7	2.8	
Range	Bin1	Bin2	Bin3	Bin4	Bin5	Bin6	Bin7	Bin8	Bin9	Bin10
RMS Bins Distribution	[0, 0.3]	[0.3, 0.6]	[0.6, 0.9]	[0.9, 1.2]	[1.2, 1.5]	[1.5, 1.8]	[1.8, 2.1]	[2.1, 2.4]	[2.4, 2.7]	[2.7, 3]
	0	1	1	0	0	2	1	2	0	1

Figure 14: Bins used for RMS values distribution

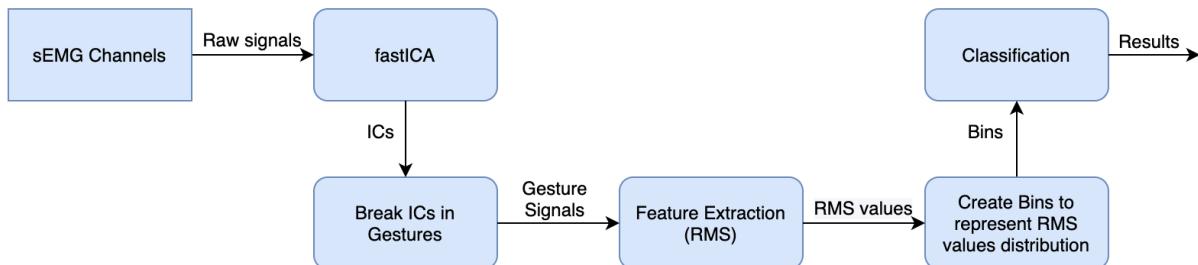


Figure 15: Bins Method Diagram

Using the new features as the classifier input and running a multinomial logistic regression model, an accuracy of 0.562 is achieved. However, as seen in the confusion matrix 2a, the vast majority of predictions are made for class 1. Running a binary logistic regression to see if there is a gesture or not, the accuracy is 0.760, showing that using this method, there is an improvement in the discrimination on whether a gesture is performed or not. When class 1 is removed then the accuracy achieved by the classifier is 0.402. All accuracy results are

improved when compared to the baseline classifiers. The confusion matrices 2b and 2c show more balanced predictions for binary and “only gesture” problems.

		True class						
Predicted class		1	2	3	4	5	6	7
	1	104	11	15	8	17	3	1
	2	4	8	3	1	0	1	0
	3	1	1	2	0	1	2	0
	4	1	3	1	7	5	4	0
	5	0	1	0	0	0	0	0
	6	0	0	0	1	0	1	0
	7	0	0	0	0	0	0	0

		True class		
Predicted		NG	G	
	NG	95	37	
	G	15	70	

		True class						
Predicted class		2	3	4	5	6	7	
	2	11	4	3	0	3	0	
	3	2	7	0	5	4	0	
	4	5	3	11	6	5	0	
	5	4	6	2	8	3	1	
	6	2	1	1	4	6	0	
	7	0	0	0	0	0	0	

Confusion Matrix 2: Bins after the fastICA algorithm - a) Original Classification Problem.

b) Binary Classification Problem. c) Original Classification Problem without "No Gesture".

4.2.5 Windows

Taking a closer look at some gestures, some abnormalities are observed in the signals. Gesture labels were set during the experiment for each gesture a subject made, but gestures were not made in a strict window. This is illustrated in figure 16. These signals correspond to class 1, where no gesture was performed by the subject. In an ideal case, all signals would be very close to 0. However, it appears that previous and next gestures affect the signal, so this is something that should be considered.

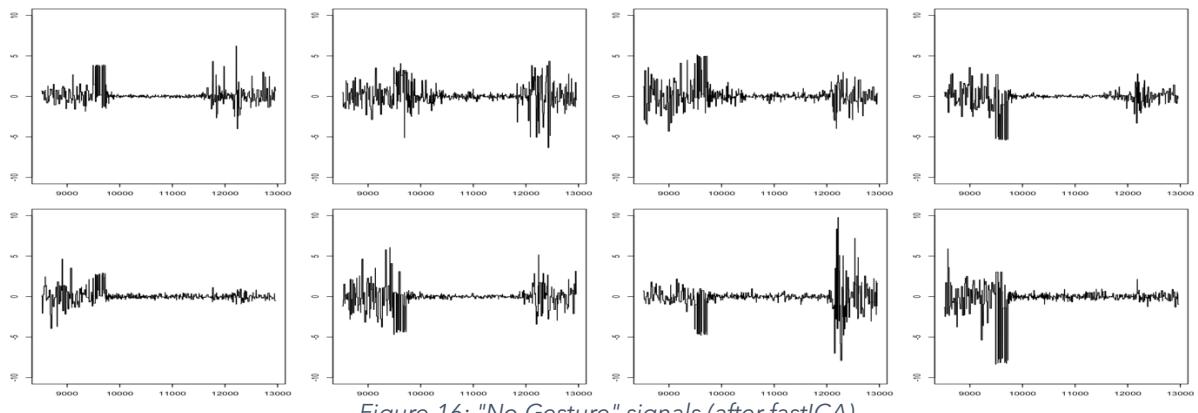


Figure 16: "No Gesture" signals (after fastICA).

To limit the effect of the previous and next signal on the current one, the window technique for feature extraction is applied. It is assumed that the main part of each gesture happens in the middle of the gesture signal. Thus, signals are cut by 25% on left and right, and keeping the middle 50%. The idea of using windows to get the main part of the gesture was taken from Zhang, Yang, Qian and Zhang (2019) where sliding windows were used to improve feature extraction. In this case, a single window of 50% of the signal was used, as described above.

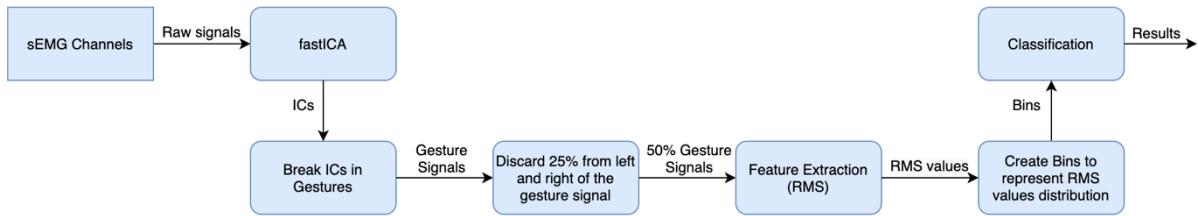


Figure 17: Bins with Windows Method Diagram

The above technique seems to improve classification accuracy. The new model seems to understand better when there is a gesture or not, improving the accuracy of the binary problem to 0.824. Slight improvements are also observed in the general problem, where the classification accuracy is 0.608, and in the problem where “no gesture” is absent where the accuracy is 0.458. Again, it appears that there is some improvement on the prediction balance. Confusion matrices are presented below.

		True class						
Predicted class		1	2	3	4	5	6	7
	1	106	7	13	1	16	9	1
	2	2	11	3	1	1	3	0
	3	0	1	0	0	0	0	0
	4	0	4	3	11	5	4	0
	5	0	1	1	2	1	2	0
	6	2	0	1	1	0	3	0
	7	0	0	0	0	0	0	0

		True class		
Predicted		NG	G	
	NG	97	25	
	G	13	82	

		True class						
Predicted class		2	3	4	5	6	7	
	2	13	3	2	1	2	0	
	3	2	7	0	6	2	0	
	4	4	4	11	4	5	0	
	5	4	3	4	10	4	1	
	6	1	4	0	2	8	0	
	7	0	0	0	0	0	0	

Confusion Matrix 3: Bins with windows after fastICA - a) Original Classification Problem.
b) Binary Classification Problem. c) Original Classification Problem without "No Gesture".

4.2.6 One for all

Although the above methods improved the classification accuracy, it seems that this improvement has some limits due to the information lost during the future redesigning to represent the distribution of the RMS values. So, something else should be done in order to improve the classification results.

The problem that should be resolved is the ordering ambiguity ICA introduces to independent components. What ICA does is trying to estimate the mixing matrix and then inverse it to calculate the unmixing matrix. After these steps, it multiplies the observed signals (mixtures) with the unmixing matrix to give the independent components. However, each time an ICA algorithm runs, it can give a different permutation of the mixing matrix.

Since the subjects participated in the experiment were healthy, we can assume that their muscles act similarly when they perform a certain gesture and see how this method works. This assumption is made, so the mixing matrix to be calculated only once, and then multiplied by all the mixed signals in the experiment. This method will leave aside any permutation ambiguity ICA introduces since it will calculate independent components for all subjects based on the mixing matrix of a random one.

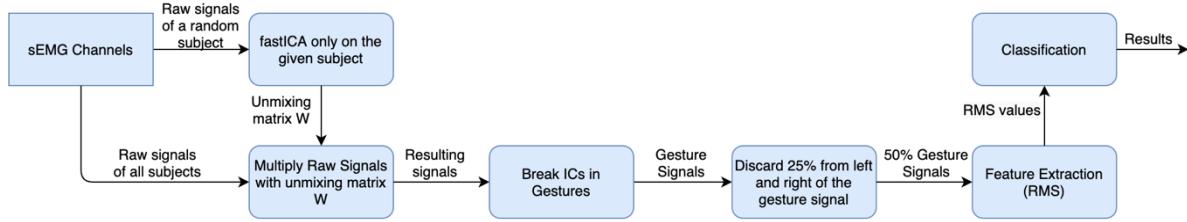


Figure 18: One for All Method Diagram

Using this method, there is no need to use RMS values distribution (bins) to build the classifier, since all independent components have the same order. So, the job is much easier now. While the method described above is not ideal since it does not take advantage of ICA on all signals but only on one subject, it achieves a noticeable improvement in classification accuracy. The classification was performed twice. The first time no feature redesign other than RMS was made, while the second time windows were used. Without windows, classification accuracy is 0.664 for the original problem, 0.785 when removing “no gesture” class, and 0.677 for the binary problem. Using windows the respective accuracies are 0.733, 0.785, and 0.797. The confusion matrices for the latter model are presented below, showing that predictions are more balanced than before, even for the original problem.

		True class						
Predicted class	1	2	3	4	5	6	7	
	1	104	12	9	1	9	8	0
	2	4	8	0	0	0	0	0
	3	0	4	12	0	0	2	0
	4	0	0	0	15	4	1	0
	5	0	0	0	1	10	0	0
	6	2	0	0	0	0	10	1
	7	0	0	0	0	0	0	0

		True class						
Predicted		2	3	4	5	6	7	
	NG	102	36					
	G	8	71					

		True class						
Predicted class	2	3	4	5	6	7		
	2	18	0	0	1	0	0	
	3	5	19	0	0	3	0	
	4	0	0	14	5	1	0	
	5	0	0	3	16	0	0	
	6	1	2	0	1	17	1	
	7	0	0	0	0	0	0	

Confusion Matrix 4: One for all with windows - a) Original Classification Problem.
b) Binary Classification Problem. c) Original Classification Problem without “No Gesture”.

4.2.7 Correlation Matrix with Mixtures

From all the above analyses, it appears that the independent components’ order is very crucial for the gesture classification. While the previous step is not ideal, as explained above, it achieves a great improvement using the same order for all independent components. So, steps towards this direction should be made.

As stated, the problem of the ordering ambiguity is due to the absence of the original signals. So independent components and source signals cannot be compared to give the right order. A good question would be if the independent components could be compared to the observed signals to help with the signal order. ICA “cleans” mixture signals from noise, artifacts, and signals from the neighbouring muscles. Thus, it can be assumed that the mixed signal taken from one sensor will be correlated to the “cleaned” signal calculated after ICA. The method we propose is going to be analysed below.

The first steps are the same as previous methods. Raw sEMG signals are used as input to the fastICA algorithm, which exports the estimates about the initial source signals, called independent components. These signals should be in the same order for all subjects in the experiment. The correlation between observed mixtures and “cleaned” signals is made using a correlation matrix. On table 3 one can see how a correlation matrix (absolute values) looks for a random subject. It appears that for each independent component there is a mixture that correlates the most to. The highest correlations for each independent component are highlighted and the matchings are stated in figure 19. It is very important the fact that the correlation coefficients for the matching signals are high. Note, that this correlation matrix and correspondence hold only for this certain participant and for this ICA run. The method was used in 3 more subjects to check manually if the results in the table 3 and figure 19 were a coincidence. Both results showed a 1-1 correspondence, and matching correlation coefficients of a similar level as the ones below.

	IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8
Ch1	0.036	0.334	0.050	0.064	0.533	0.074	0.019	0.769
Ch2	0.050	0.983	0.029	0.031	0.032	0.165	0.022	0.002
Ch3	0.390	0.292	0.073	0.011	0.043	0.868	0.033	0.013
Ch4	0.968	0.017	0.052	0.042	0.022	0.157	0.178	0.025
Ch5	0.176	0.007	0.095	0.009	0.002	0.055	0.978	0.016
Ch6	0.107	0.022	0.947	0.145	0.013	0.015	0.264	0.006
Ch7	0.078	0.046	0.277	0.945	0.068	0.063	0.092	0.076
Ch8	0.067	0.061	0.133	0.192	0.959	0.060	0.020	0.117

Ch1
Ch2
Ch3
Ch4
Ch5
Ch6
Ch7
Ch8

IC8
IC2
IC6
IC1
IC7
IC3
IC4
IC5

Table 3 (Left): Correlation matrix of mixed signals with independent components.

Figure 19 (Right): Observed mixtures are corresponded to the independent components with the highest correlation coefficient.

Then, the permutation matrix is created. It is an 8x8 square matrix, where rows correspond to channels’ (sensors) signals and columns to independent components. All values in the matrix are 0 except for the ones that have a correspondence as illustrated in figure 19. Then independent components are multiplied with the permutation matrix. Doing this, independent components are ordered based on their correlation with the observed signals. Then, the window method is applied to the signals as before. RMS values are calculated for each of the eight gesture signals. These RMS values are used as the multinomial logistic regression classifier input.

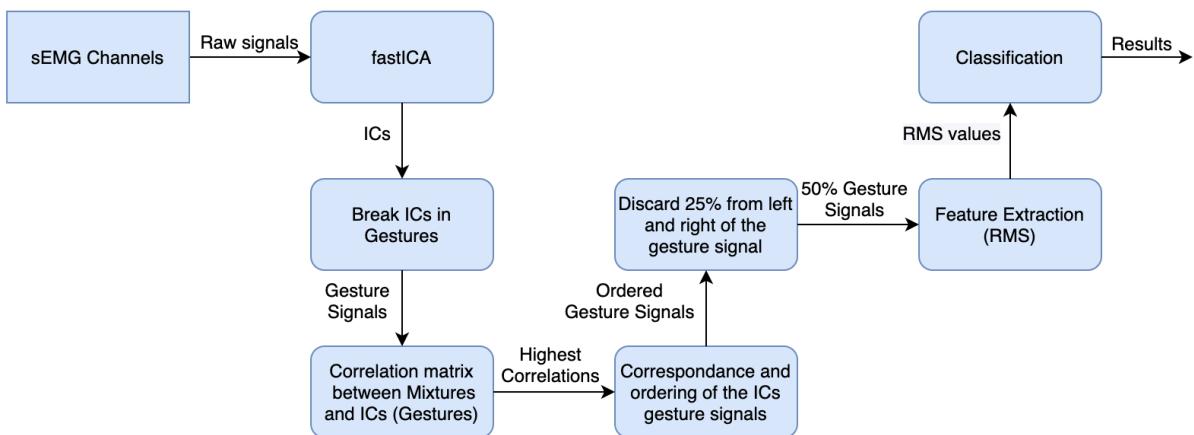


Figure 20: Correlation Matrix Method Diagram

The accuracy for the original problem is 0.811, for the binary problem 0.834, and for the problem where “no gesture” is absent the accuracy is 0.798. The confusion matrices are presented below. In the first confusion matrix, one can see that the model classifies accurately most of the gestures, overcoming well enough the class distribution problem, where class 1 has the vast majority of instances, making it more difficult problem for the classifier. Accurate and balanced predictions are observed in the other two confusion matrices, 5b and 5c as well.

		True class						
Predicted class		1	2	3	4	5	6	7
	1	104	8	8	1	4	3	0
	2	1	16	0	0	0	0	0
	3	4	0	13	0	0	3	0
	4	0	0	0	13	3	0	0
	5	0	0	0	3	16	0	0
	6	1	0	0	0	0	14	1
	7	0	0	0	0	0	1	0

		True class						
Predicted			NG	G				
	NG	103	29					
	G	7	78					
		2	3	4	5	6	7	
2	21	1	0	0	1	0		
3	0	16	0	0	3	0		
4	0	0	13	4	0	0		
5	1	2	4	18	1	0		
6	2	1	0	0	16	0		
7	0	1	0	1	0	1		

Confusion Matrix 5: Correlation Matrix with windows - a) Original Classification Problem.
b) Binary Classification Problem. c) Original Classification Problem without “No Gesture”.

4.2.8 Results Summary

On the following table, one can see the accuracy summary for all the different methods described above and used.

	Original	Binary	Only Gestures
Baseline	0.507	0.507	0.243
Bins	0.562	0.760	0.402
Bins and Windows	0.608	0.824	0.458
One for All	0.664	0.677	0.785
One for All – Windows	0.733	0.797	0.785
Correlation - Windows	0.811	0.834	0.798

Table 4: Accuracy for all the different methods described for the three different problems (Summary).

The first method used was the baseline, where mixtures were used for the classification. Since the eight mixtures were similar to each other, the mean RMS value was used as the classifier input. Although classification accuracy seemed to be acceptable for a baseline solution, the truth is that this number was because the majority of the instances were in an individual class. So, removing this class and checking the accuracy only on the other classes, much less accuracy is observed.

The second method used was the Bins. After running the fastICA algorithm, the independent components’ order ambiguity was introduced. Source signals are unknown, in contrast to the simulation. So, Bins was a method to overcome this obstacle and try to redesign the features so that they represent the distribution of the RMS values. This method achieved a noticeable improvement in accuracy, especially on the binary problem. This improvement was even bigger when windows were used to get the main part of the

gesture. However, this method had some drawbacks. The most important thing was that although it achieved overcoming the ordering ambiguity and features meaning something, much information is lost because of the feature redesign.

Thus, something else had to be done. The assumption that muscles perform similarly for all humans led to the next method. Taking the assumption as a given, the ordering ambiguity of ICA could be eliminated. Running the fastICA on only one participant and assume that the unmixing matrix is the same for all the other subjects helped to overcome the barriers explained. After that, all independent components had the same order for all participants, without losing too much information like in the previous method. The windows method also gave a great boost to the classification results of the “One for all” method, since it achieved an accuracy improvement of more than 7% in two of the problems compared to the same method but without windows.

However, a more general method that would take advantage of all the participants needed to be used. The fastICA algorithm ran for all subjects in the experiment and gave independent components in an abstract order. A correlation matrix was used in order to try to order them. Since the source signals were unknown, the correlation matrix included the correlation coefficients of the independent components with the observed signals. That happened since ICA “cleaned” the signals from noise, artifacts, and signals from the neighbouring muscles. So the “cleaned” signal (independent component s_i) may have some correlation with the closest observed signal x_j . With this method, independent components are ordered and the ambiguity is surpassed. This method outperformed all other methods achieving a classification accuracy of more than 0.79 in the three classification problems.

4.2.9 Limitations

As mentioned above, the accuracy that the system identifies the hand gesture correctly was the measure that was used to evaluate the performance of the method. The proposed method implemented in section 4.2.7 achieved a classification accuracy of more than 79% in the three different problems, showing that the method performs well. Especially, when this method is compared to the baseline solution or the other implementations, it is more clear that it performs well, improving the classification accuracy significantly.

In this section, the implementation analysed above will be used on the simulation data illustrated in section 3.3 in order to see whether or not the proposed method solves the ordering ambiguity. This is done since the source signals are known in the simulation environment and the order of the signals is known.

The four source signals presented in figure 5a were used to produce the mixed signals by being multiplied with a random matrix A, which was generated with values that follow a uniform distribution. Then the fastICA algorithm estimates the independent components illustrated in figure 5c, using as input only the mixtures (figure 5b). As described in section 4.2.7, the correlation coefficients between mixtures and independent components are calculated giving the correlation matrix in table 5. The correlation matrix in table 6 shows

the correlations between source and observed signals, showing that the signal ordering was retained using this particular mixing matrix.

	IC1	IC2	IC3	IC4
mixture1	0.276	0.161	0.940	0.115
mixture2	0.547	0.590	0.427	0.287
mixture3	0.761	0.321	0.461	0.325
mixture4	0.350	0.057	0.401	0.844

Table 5: Correlation coefficients of mixtures and independent components

	mixture1	mixture2	mixture3	mixture4
signal1	0.924	0.434	0.368	0.273
signal2	0.290	0.807	0.040	0.098
signal3	0.224	0.248	0.859	0.387
signal4	0.262	0.300	0.378	0.886

Table 6: Correlation coefficients of source signals and mixtures

Observing the table 6, it appears that the mixing matrix did not change the order of the signals. This could happen, but it would not be a big problem, as a reordering could be done using the highest correlations of the table 6. Besides, it appears that there is 1-1 matching of the independent components and the observed signals (table 5). However, repeating the above experiment 50 times using the same source signals but generating each time a different random mixing matrix, it appears that only in 58% (29 times) of the repetitions there was 1-1 correspondence (table 7). For the rest repetitions, more than one signals corresponded to a single independent component. Note, that this is also a result of the random simulation data and it can get worse or better depending on the randomly generated data.

	Times
1 IC – 4 Signals	3
2 ICs – 4 Signals	5
3 ICs – 4 Signals	13
4 ICs – 4 Signals	29

Table 7: Independent components and source signals matching after 50 repetitions with different random mixing matrices

When data are not random but they are structured and meaningful (real data), then the method works better. In addition, in simulation data, a random matrix was used as explained. Since the weights for each of the source signals were given randomly, then it could be the case that a source signal is dominant over the others and it is present in all the mixtures, or signals have very similar weights. Thus, fastICA could face some difficulties in separating the signals accurately. Moreover, the random mixing matrix may have values that give a different order to the mixtures. However, this can be solved by using a correlation matrix similar to table 6, and reorder the mixtures. All these show that data and mixing procedure play a crucial role in how well the above method performs. It is important

for the proposed method that the mixing procedure is reasonable, or in other words, the weights of the mixing matrix to be logical, for instance to represent distances of sensors to the sources.

The proposed method described in section 4.2.7 appears to work well when the mixing procedure is good, and worse when the weights on the mixing matrix are not suitable. An example of a good mixing procedure is the sEMG signals, where appropriate weights are given to each muscle's activity. The distance between a muscle and a sensor define a reasonable weight. This gives some good enough observed signals, making it easier to be analysed and estimate the independent components accurately and make the above method work properly. The importance of a good mixing matrix and real data is also observed if anyone compares the correlation coefficients in table 3 (section 4.2.7) and table 5 (section 4.2.9). In the latter table (simulation data) although there is 1-1 matching, the correlation coefficients appear to be lower than those in table 3 (sEMG data), giving the conclusion that the random mixing procedure can cause problems if it is not proper. In contrast, the correlation coefficients in table 3 are high, indicating that the mixing procedure was good and that when the proposed method is used on real data with suitable mixing procedure, it works well.

Finally, the assumption that led to the proposed method, that the mixed signal taken from one sensor will be correlated to the "cleaned" signal (independent component) calculated by ICA, is confirmed by comparing the correlation coefficients between the recovered and the observed signals in these two cases, as analysed above. Therefore when a reasonable mixing matrix (weights are proportional to the distance of source from the sensor and to the signal amplitudes) and not random data are used, the proposed method can be a solution to the signal separation problem.

5. Conclusion

5.1 Review

In this dissertation, a BSS technique called ICA was demonstrated. The theoretical background, as well as the application of ICA in a simulation environment with generated signals and in a real sEMG signals dataset, were presented. The complications, and particularly the independent components' ordering ambiguity ICA introduces, were analysed as well.

On simulation data, the limitations are less, since it is a supervised learning problem, with source signals being known. Consequently, ordering ambiguity is handled easier, since a correlation matrix between source signals and independent components estimated is enough to correspond initial signals to the recovered ones. Correlation coefficients between the corresponded initial and estimated signals define the performance of ICA during signal recovery. It was shown that the fastICA algorithm performs well, and a comparison to the more complex algorithm ProDenICA was made. Results showed that ProDenICA performs better in terms of signal recovery, especially when there are more signals to be separated. However, its computational complexity and the computation time it needs to estimate the independent components make it restrictive in cases when fast computations are needed.

In practice, source signals are unknown, making the signal recovery an unsupervised learning problem. There is not an indication of whether or not ICA performed well, as independent components cannot be correlated with the source signals. For this reason, a real sEMG dataset with hand gestures was used. The measure of the quality of signal recovery was the accuracy of the system implemented to identify hand gestures correctly. The most difficult problem faced and needed to be resolved was the ordering ambiguity of the estimated independent components. Some implementations were presented and compared. As it was explained, ICA performs some "cleaning" from noise, artifacts and activity originating from neighbouring muscles on the mixture signals. The intuition that an observed signal from a certain sensor would have the highest correlation coefficient with the corresponding "cleaned" signal, was proven to be right. This technique combined with some post-ICA steps, appeared to perform the best amongst the methods examined, having the highest gesture classification accuracy. The proposed approach achieved a gesture classification accuracy of more than 79%, showing how useful ICA is for signal separation problems as well as the importance of resolving the independent components' ordering ambiguity. Finally, the importance of a good mixing procedure was shown,

comparing the real dataset results with the ones that occurred when simulation data with random mixing matrices were used.

5.2 Further Steps

Although some methods that were analysed and implemented during the dissertation worked well, there are still many improvements that can be done and worth further research on this field.

First, ICA has some margins for improvement. While there are a bunch of algorithms available, only a few of them stand out. Two of them are fastICA and ProDenICA, both having many merits but also some crucial drawbacks. There is a need for new methods and algorithms that will perform well in the signal separation task, and at the same time, they will not need much computational resources and time to estimate the independent components. Some research has been made on this field the recent years, with Samworth and Yuan (2018) paper on ICA with non-parametric maximum likelihood estimation to give promising results in terms of accuracy on signal recovery.

On sEMG signal recovering, there are wider margins for development. This field of biomedical signal processing has seen much research the recent years with the rise of artificial intelligence. The hand gesture sEMG signals that were covered in the dissertation have many useful applications, with one of the most important ones being the Prosthetic technology. Very high accuracy on gesture classification is vital, so the sEMG signals measured from a patient's muscles are recognized and the patient being able to control the prosthetic part. Much work on preprocessing and post-ICA steps need to be done, so the accuracy will reach acceptable enough for applications like this. For this, some more accurate classification techniques could be used like Artificial Neural Networks instead of the logistic regression classifier used for this dissertation. Additionally, more methods other than RMS could be used in order to contain more signal characteristics, and then build a more reliable classifier. Apart from the accuracy needed, computation time is a necessary condition. There is a need for very fast data processing and gesture recognition since for real-time applications the total time from signal acquisition to pattern recognition should not exceed 300ms, making the need of fast and accurate ICA algorithms more essential.

References

- Amari, S. (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, 276, 251-276.
- Bell, A. J., & Sejnowski, T. I. (1995). An Information-Maximization Approach to Blind Separation and Blind Deconvolution. 1159(1994), 1129-1159.
- Comon, P. (1994). Independent Component Analysis, a new concept? *Signal Processing, Elsevier*, 36, 287-314. [https://doi.org/10.1016/0165-1684\(94\)90029-9](https://doi.org/10.1016/0165-1684(94)90029-9)
- Djuwari, D., Kumar, D., Arjunan, S., & Naik, G. (2007). Limitations and Applications of ICA for Surface Electromyogram for Identifying Hand Gestures. *International Journal of Computational Intelligence and Applications*, 7(3), 281-300.
- Efron, B., & Tibshirani, R. (1996). Using Specially Designed Exponential Families for Density Estimation. *The Annals of Statistics*, 24(January 1995), 2431-2461.
- Garcia, M. A. C., & Vieira, T. M. M. (2011). Medicina del Deporte. *Rev Andal Med Deporte*, 4(1), 17-28.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised Learning. In *The Elements of Statistical Learning*. https://doi.org/https://doi.org/10.1007/978-0-387-84858-7_14
- Huber, P. J. (1985). Projection Pursuit. *The Annals of Statistics*, 13(2), 435-475.
- Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons, Inc.
- Hyvärinen, A., & Oja, E. (2000). *Independent Component Analysis : Algorithms and Applications*. 1(1).
- Jutten, C., & Herault, J. (1991). *Blind separation of sources, Part I" An adaptive algorithm based on neuromimetic architecture*. 24, 1-10.
- Lobov, S., Krilova, N., Kastalskiy, I., Kazantsev, V., & V.A., M. (2019). EMG data for gestures Data Set. Retrieved from <https://archive.ics.uci.edu/ml/datasets/EMG+data+for+gestures>
- Makarov, V. A., Lobov, S., Krilova, N., Kastalskiy, I., & Kazantsev, V. (2018). Latent Factors Limiting the Performance of sEMG-Interfaces. *MDPI Sensors*, 1-19. <https://doi.org/10.3390/s18041122>
- Naik, G., Kumar, D., Arjunan, S., & Palaniswami, M. (2008). Independent Component Approach to the Analysis of Hand Gesture sEMG and Facial sEMG. *Biomedical Engineering: Applications, Basis and Communications*, 20(2), 83-93.
- Naik, G. R., K. K. D., Singh, V. P., & Palaniswami, M. (2006). Hand gestures for HCI using ICA of EMG. *Australian Computer Society, Inc.*, 56(VisHCI), 67-72.
- Samworth, R. J., & Yuan, M. (2018). *Independent component analysis via nonparametric maximum likelihood estimation*. 1-28.
- Tesfayesus, W., Yoo, P., Moffitt, M., & Durand, D. M. (2004). *Blind Source Separation of Nerve Cuff Recordings*. 1-4.
- Thalmic Labs. (n.d.). Myo Gesture Control Armband. Retrieved from <https://www.robotshop.com/uk/myo-gesture-control-armband-black.html>
- Tharwat, A. (2018). Independent Component Analysis : an Introduction. *Applied Computing and Informatics*, (August 2018). <https://doi.org/10.1016/j.aci.2018.08.006>
- Tibshirani, R., & Hastie, T. (2003). Independent Components Analysis through Product Density Estimation. *MIT Press, Cambridge, MA*, pp. 649-656.
- Zhang, Z., Yang, K., Qian, J., & Zhang, L. (2019). Real-Time Surface EMG Pattern Recognition for Hand Gestures Based on an Artificial Neural Network. *MDPI Sensors*, 1-15.