Independent component analysis

Table of Contents

1) Independent component analysis (ICA)

1.1 Introduction

Independent component analysis (ICA) is a signal processing tool used to separate an instantaneous mix of independent, non-Gaussian random variables. This is used mainly when we have different sensors that are picking up different signals and we want to decompose the signal into its original components to uncover the different sources or to denoise the original mixtures. The underlying assumption is that the sources are linearly mixed, do not have a Gaussian distribution and are independent one from another. The mixture can be represented with the following mixing model:

$$X = AS$$

Where:

- **X** is the mixture, that is the signal that is captured. Its rows are equal to the number of sensors that the signal is captured from.

- **A** is the mixing matrix. Each column of the mixing matrix contains the mixing weights for a defined signal source.

- **S** is the source matrix. Each row of this source matrix contains an independent component that is to be estimated. The number of components that can be derived cannot exceed the number of available sensors.

Thus, the estimation of the sources can be simplified by the following equation:

$$Y = WX$$

Where **Y** is our signal estimation (**Y≈S).** and **W** is the inverse or pseudo-inverse of the mixing matrix **A** also known as the de-mixing matrix **($W \approx A^{-1}$).** In other words, if we know the linear combination by which the sources have been mixed, we can recover the original sources present in the signal. The steps by which it achieves that are summarized as follows:

1) **Whitening:** Before separating the independent components, the mixture **X** is whitened so that its different components are made uncorrelated to each other. Its aim is to recover the original shape of the data that was altered during the mixing process. This is usually done with projecting the data unto its principal components as to obtain components that are uncorrelated with one with the other (covariance matrix of unity). Whitening the data renders the estimation of the independent components much easier to derive.

2) **FastICA algorithm:** After obtaining the uncorrelated sources. We are left with the task of reducing the Gaussianity of the data by maximizing the negentropy of the whitened data. In other words, we want to obtain the estimated sources **Y** that are statistically independent

from each other. This follows the central limit theorem dictates that any mixture of random variables is more Gaussian that the original variables. At each iteration of the algorithm, the values of **W** are updated as to make the sources more independent from each other (less Gaussian). When **W** converges, we obtain the estimated independent source. [1]

In this exercise session, we will apply ICA techniques for two different settings. We will first explore different ICA properties by mixing four different mixtures with different types of mixtures and noises and try to denoise specific sources from our mixture. The aim is to get acquainted with the way ICA works and specifically the FastICA algorithm in that context. Afterwards, we will use the obtained knowledge on an EEG signal to remove eye blink artefact that is present in the different channels.

1.2 Programming Exercises

1.2.1 Exploration and analysis of ICA (properties) on a mixture from ECG

1.2.1.1 Measuring 4 mixtures

We were provided 4 different signal sources (ECG, Sawtooth, PPG and white Gaussian noise). The four sources **S** were first normalized and then mixed with a provided mixing matrix **A** simulating a scenario where a mixed signal is captured from four different sources. FastICA was then applied to obtain the estimated sources **Y** and the estimated mixing matrix **W.** The estimated sources were then normalized as to have their variance one and their mean equal to zero.

Question 1: *You normalized the estimated sources. Why is this important? Is this necessary for ICA?*

Normalization of the estimated sources is needed to account for the ambiguity of the scaling. That is because we can multiply the mixing vector **A** and divide the sources **S** when we are trying to recover them. That is why the sources need to be normalized after recovering them.

We then proceeded to match the original sources **S** with the estimated sources **Y** through correlation. The corresponding signals were then plotted next to each other **(Figure 1.1).**

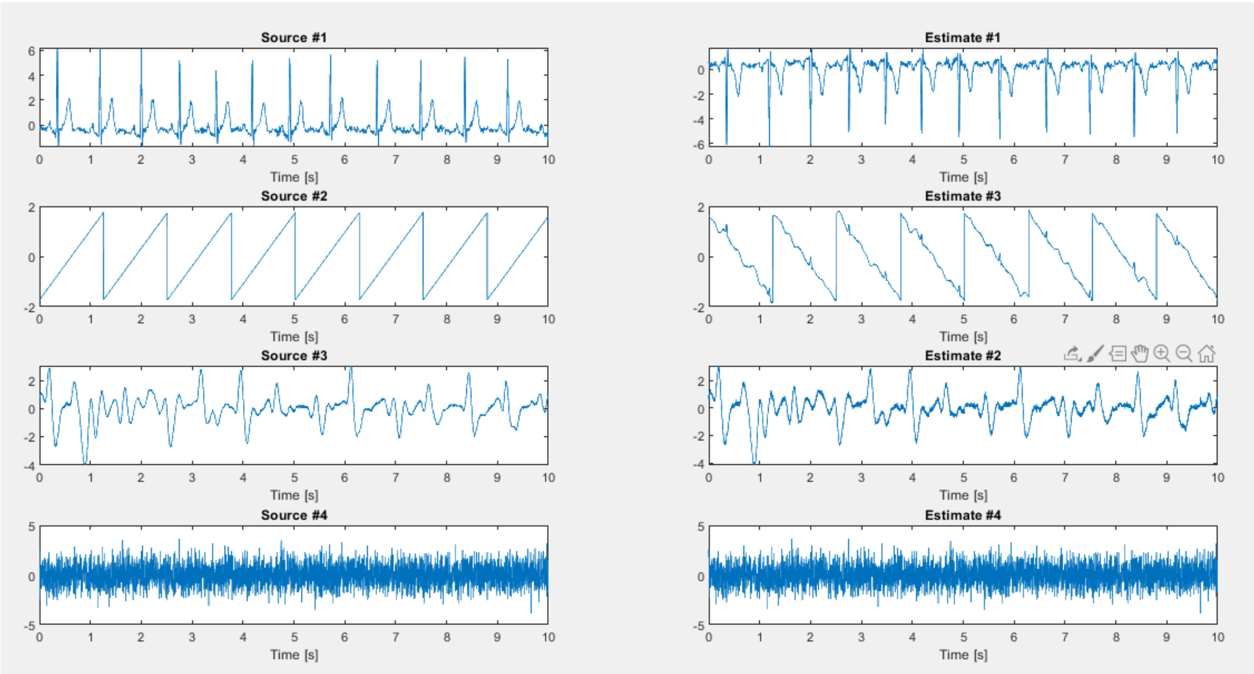

Figure 1.1 *Original sources vs matching estimated components (4 mixtures)*

__**Question 2:**__ Y*ou matched the estimated sources with the original sources. Why do we need to do this? Could we not directly use the output of the fastICA-algorithm?*

Since the order of sources are not fixed (the de-mixing matrix can have any order of columns on each iteration of FastICA) we needed to account for the ambiguity of the order by which the estimated signals **Y** are outputted. That is why we reorder them through correlation with the known sources.

Now that we have the matching signals plotted next to each other, we wanted to estimate the performance of our ICA output for 4 mixtures. We calculated the Root Mean Squared Error (RMSE) between each source signals and corresponding signals **(Table 1.1).**

__**Question 3:**__ *When computing the RMSE values in 1.1.1d, you sometimes needed to change the sign of the estimated sources. Why is this?*

As mentioned earlier, there is an ambiguity in sign and scale since both the source and mixing matrix are unknown. This can be seen in **Figure 1.1** where the first and second estimated sources flipped with respect to the original sources. To account for that, we calculated two RMSE values for each original source, one with the original outputted sign and one with the flipped sign (-**Y**). The chosen RMSE was the one with the lowest RMSE that corresponds to the most exact match.

We were then tasked to remove two artefact sources (Gaussian and sawtooth) from the original mixture by modifying the mixture matrix. We were given both noises as templates. The way we approached this task is as if we have no knowledge about the original mixing matrix. We treated the mixture as the only data that was provided to us. We estimated the sources through FastICA and obtained the estimated mixing matrix **W.** We then proceeded to match the noise templates with the estimated sources as to obtain the index to the columns of the mixing matrix that correspond to those noises. We then zeroed out the components of those columns to obtain our modified estimated mixing matrix $\mathbf{W}_{\mathbf{mod\ (4x2)}}$**.** We reconstructed our denoised mixture by multiplying $\mathbf{W}_{\mathbf{mod}}$ with the estimated sources.

$$X_{denoised} = W_{mod}\, Y$$

The obtained estimation contains a mixture of both the PPG and ECG without the influence of both noises. The mixed signal and its denoised version can be visualized in **(Figure 1.2).** The influence of the Gaussian noise is no longer there as the signals look smoother. The effect of removal of sawtooth can also be visualized in the denoised signal since there is no longer a horizontal line that characterized the sawtooth in the signal.

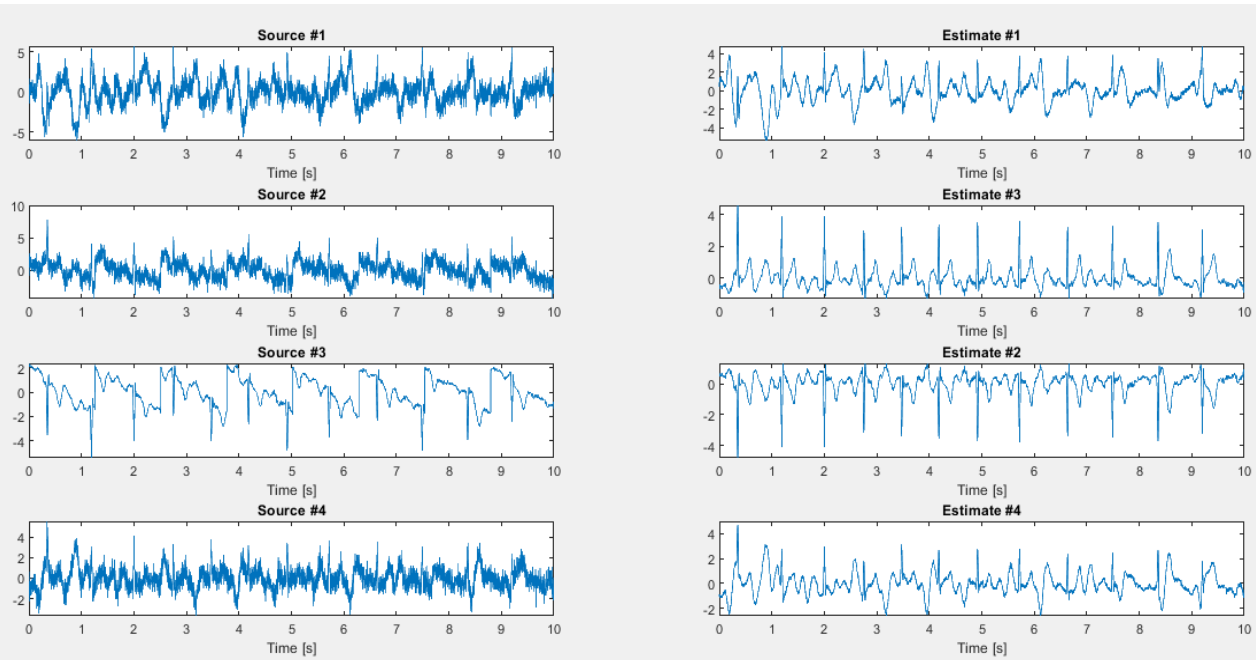

Figure 1.2 *Mixed sources vs denoised sources (4 mixtures)*

<u>Question 4:</u> *Should the estimated sources be normalized? Why (not)?*

It is not necessary to normalize the estimated sources with the modified mixing matrix the noisy original sources that are outputted from ICA are already normalized. The fact that **Y** is modified means that **X** is normalized as well.

.

1.2.1.2 Measuring 30 mixtures

We now obtain a new mixture of 30 signals through a modified mixing matrix consisting of 30 rows as to obtain a mixed signal in 30 dimensions (sensors). We repeat the same step as before (computing the estimated sources, matching plots together and RMSE computation). The original and plotted are plotted in **Figure 1.3.**

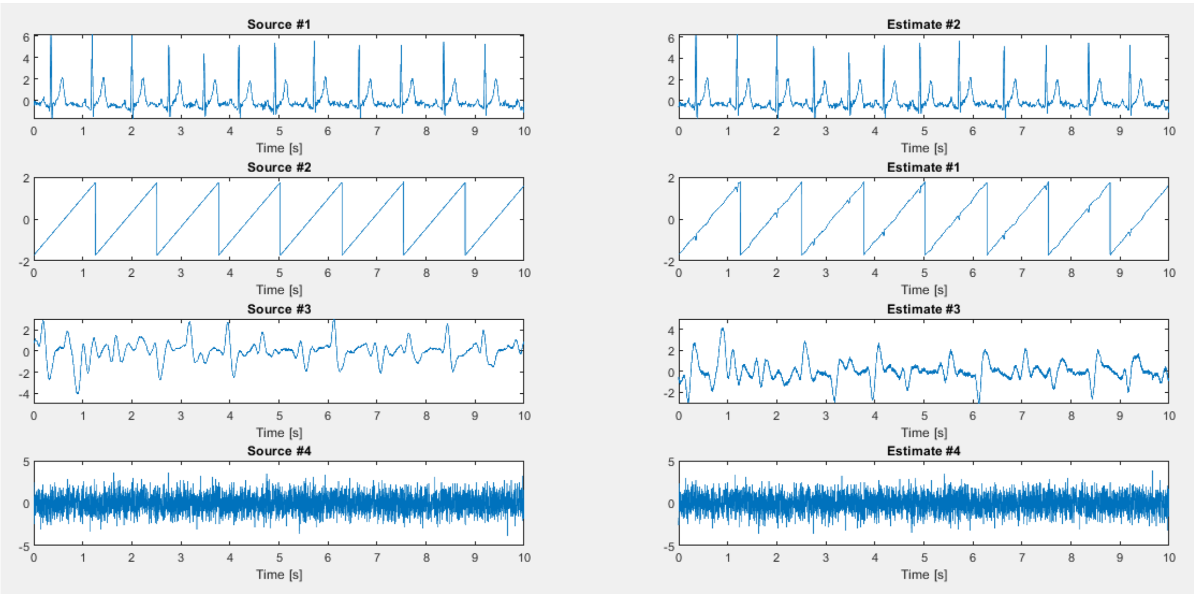

Figure 1.3 *Original sources vs matching estimated components (30 mixtures)*

<u>Question 5:</u> *Is the fastICA algorithm able to identify the number of sources in? How does it work?*

The algorithm can identify the number of independent sources. After centring the data around the mean, it calculates the eigen decomposition of the covariance matrix of the mixture. The dimensionality is reduced by projecting the dataset onto the principal axes that are orthogonal to each other. Those axes are oriented in the direction of highest variance in the data (explain the most variability in the mixture). These are the eigenvectors corresponding to the non-zero eigenvalues. In our case, the 30 dimensions of our mixture is reduced to 4 dimensions and we obtain uncorrelated components. The data is then whitened as to obtain a variance of 1 so that all components would contribute equally to the output of ICA result (we obtain a covariance matrix close to unity).

We now add random uniform random noise to our 30 mixtures. When applying ICA to the noisy mixture we obtain 30 estimates of independent noises. We correlated our original source noises with the 30 obtained estimated sources to identify the matches and plot them **(Figure 1.4).**

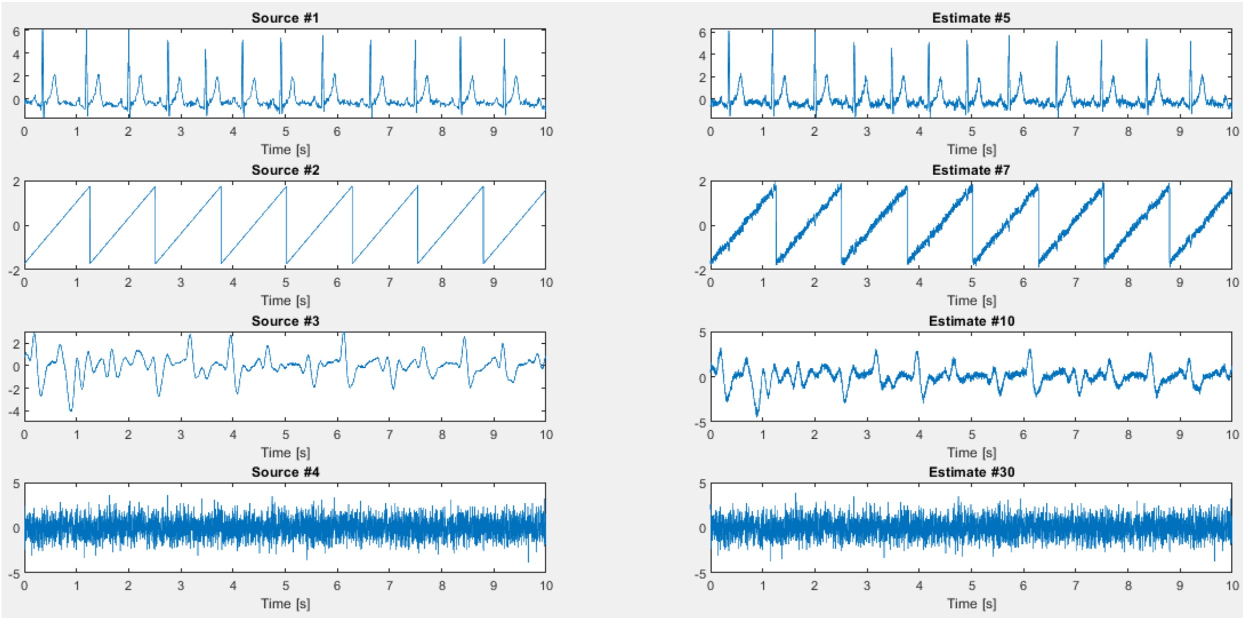

Figure 1.4 *Original sources vs best matching estimated components (30 mixtures with added uniform noise)*

We can clearly notice the effect of added noise when compared to **Figure 1.3.** In addition, our algorithm did not perform optimally since it outputted 30 estimates instead of 4. The 30 estimates were visualized (not shown here) and we could notice that only besides the 4 estimated sources. The rest corresponds to random noisy signal.

<u>**Question 6:**</u> *Is the fastICA algorithm able to identify the number of sources? How does it work?*

The algorithm is still able to recover the estimated 4 sources. However, it is outputting additional 26 sources that correspond to the noise. Unlike the previous case where the dimensions were reduced to 4 for the clean 30 mixtures, the algorithm retained all the non-zero eigenvalues so that no dimensionality reduction occurred in the whitening phase. This is due to the presence of the underlying noise since they account for many non-zero components -albeit very low in magnitude- that gives information about the small variance in the dataset. In PCA dimensionality reduction, those components mainly correspond to the noise. The default setting for the ICA algorithm is to retain all non-zero eigenvalues when reducing dimensions.

To improve on the source separation and reduce the noise of the estimated. We will perform a projection on a selected component by implementing PCA as a pre-processing for ICA. We used the MATLAB command PCA to obtain eigenvalues and eigenvectors of the covariance matrix of our noisy mixture. To estimate the number of components to choose. We plotted the eigenvalues in descending order **(Figure 1.5).**

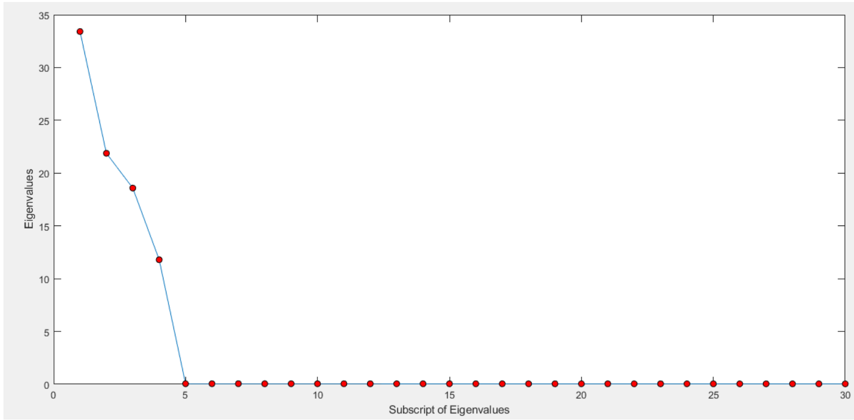

Figure 1.5 *Eigenvalues of the covariance matrix for the 30 mixtures signal with added uniform noise*

We can clearly see that most of the variability of the data can be expressed with 4 components whilst the rest correspond to the underlying noise. We projected our data onto the first 4 principal components with the highest eigenvalues. We projected our data back to the original sensor space. FastICA was applied to the denoised projection consisting of 30 dimensions. The algorithm outputted 4 sources this time which was expected since we the input was denoised. **(Figure 1.6)**

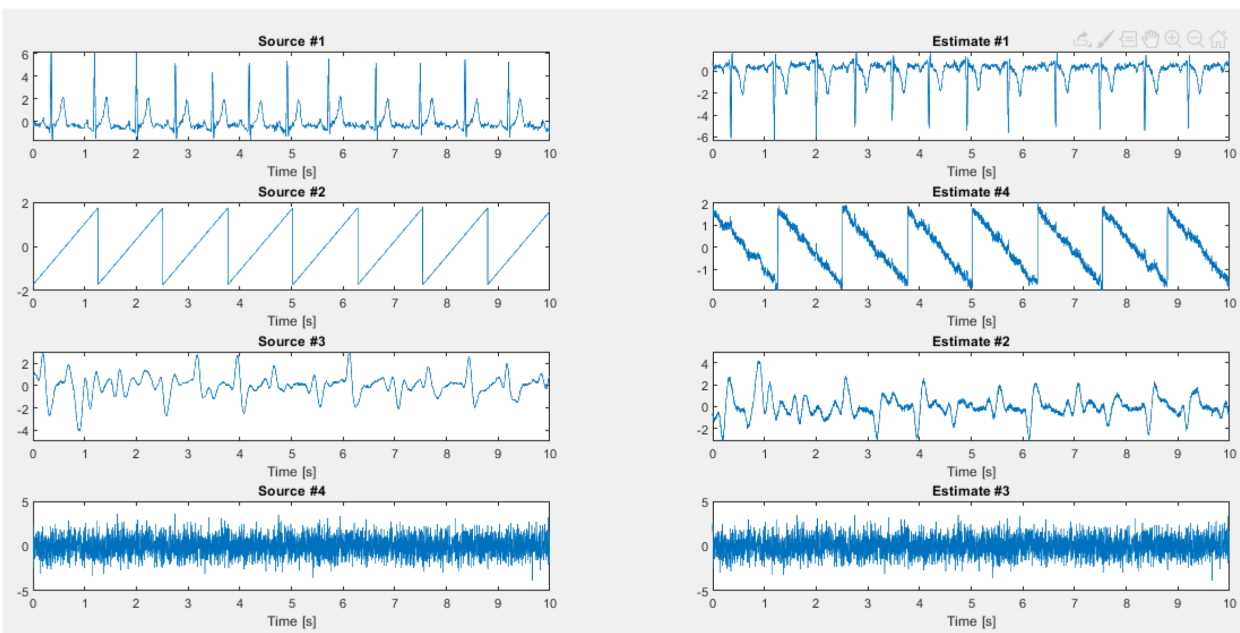

Figure 1.6 *Original sources vs best matching estimated components (30 mixtures with added uniform noise with reduced dimensions)*

Comparing with **Figure 1.5**, we can visually inspect that the noise has been substantially reduced. Implementing PCA is not only an important pre-processing tool to reduce dimensions and improve computational efficiency. But it can also be used as a tool for noise reduction. The effect of noise reduction when applying PCA for the 30 mixture is further confirmed by RMSE estimations in **Table 1.3**.

Sources	Type of mixture			
	4 mixtures	30 mixtures	30 mixtures with noise	30 mixtures with noise (Dimensions reduced)
ECG	0.067	0.056	0.124	0.083
Sawtooth	0.099	0.043	0.103	0.134
PCG	0.104	0.076	0.187	0.118
Gaussian noise	0.056	0.055	0.200	0.090

Table 1.1 *RMSE estimation between original signal sources and the signal sources derived from different types of mixtures*

Comparing all the different source estimations. We could see that the best estimate for all the cases was for the 30 mixtures (without additional noise). It is closer to the actual signal than when we estimate the sources from a mixture of 4. This indicates that FastICA is better able to estimate sources with an increased number of sensors. However, the sources from 4 mixtures where still better than for the 30 mixtures even when applying PCA to reduce the noise. That is because PCA reduces noises but does not eliminate them altogether.

Question 7: *Suppose an additional Gaussian noise source is present. Would fastICA be able to separate both? Why (not)?*

Two Gaussian sources will have a joint density that is rotationally symmetric. Its principal component can be thought of to be equal in all directions. Therefore, PCA won't be able to identify a distinct principal component for that sources. ICA will face difficulties when trying to maximize the negentropy of two Gaussian sources. So, the sources cannot be separated.

1.2.2 ICA to remove eye blink artefacts from EEG

We will now apply ICA for a removal of eyeblink artefact in an EEG signal consisting of 12 channels. The recordings are of 20s long of an auditory attention experiment. There is a presence of eye blink artefacts that need to be removed since it does not contain any relevant information for the experiment. **Figure 1.7** demonstrates the acquired signal from 12 channels, the present eye blink artefacts are marked in red.

Question 1: *In what EEG channels are the eye blinks best noticeable? Why?*

We can observe that the eyeblinks are most prominent in the the electrodes associated with the frontal lobe. That is because they are located closely to the eye. The muscle activity of the blinks generates electrical activity that is picked up mainly by the closest electrodes

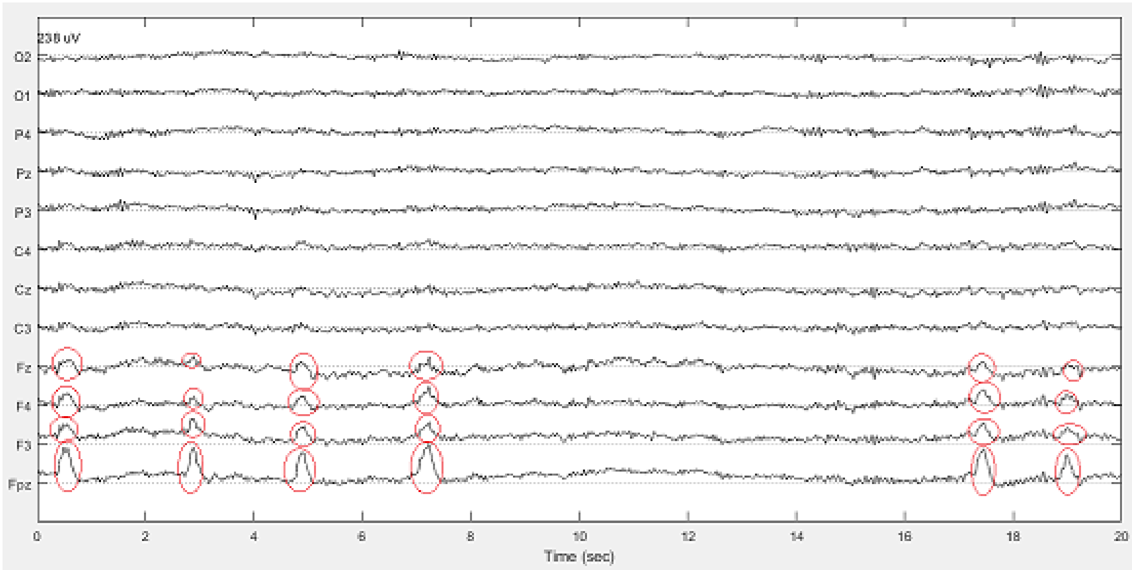

Figure 1.7 *EEG signal for 12 channels (eye blinks are indicated in red)*

We then apply FastICA to our signal to obtain the different estimated sources (**Figure 1.8).**

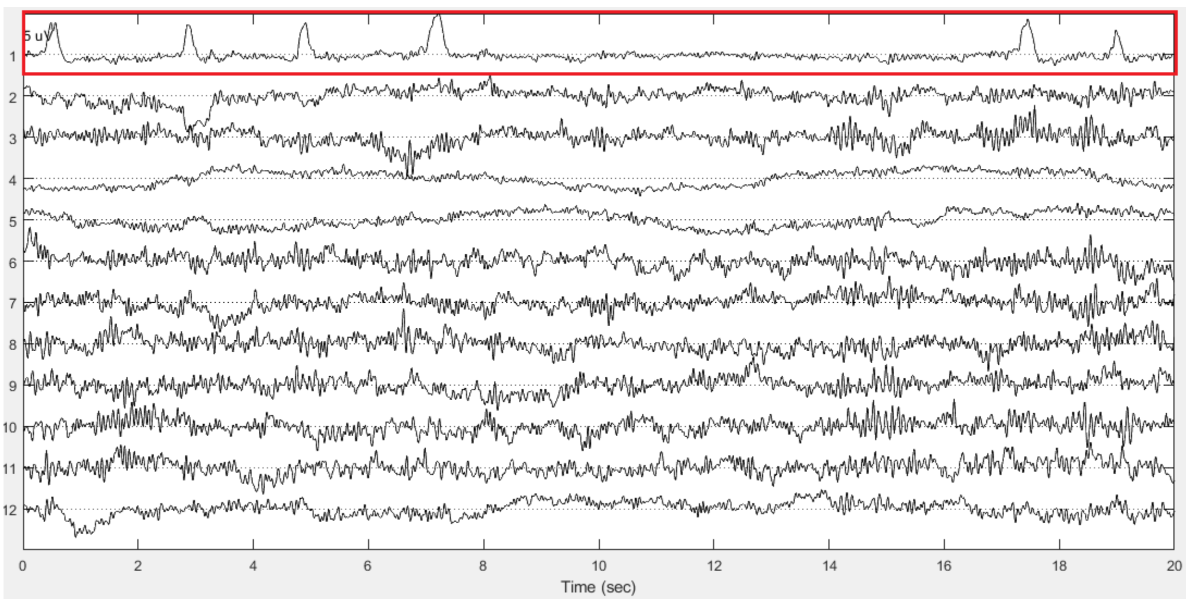

Figure 1.8 *ICA decomposition of the EEG signal (eye blink component indicated in red)*

Based on the output of the mixing matrix, we can map out a topographic representation of the contribution of each individual component **(Figure 1.9)**

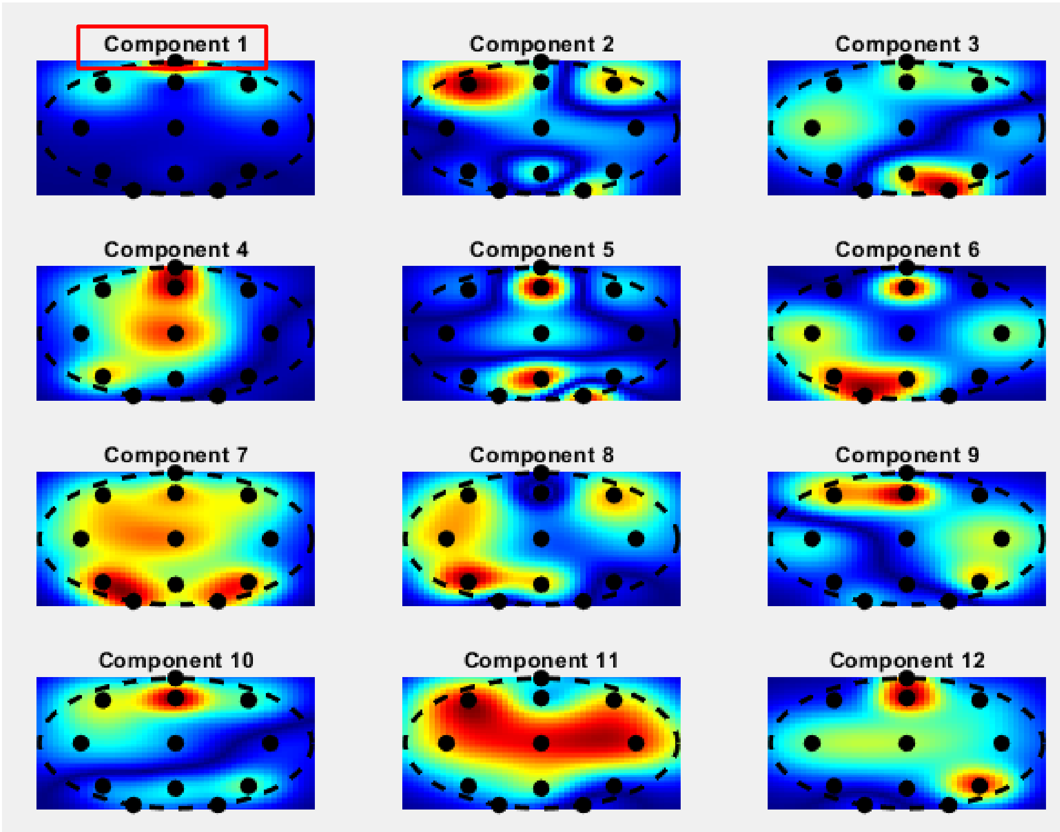

Figure 1.9 *Topographic representation of the contribution of each component (eye blink component indicated in red)*

From the decomposition of fastICA and the topographic representation. It is quite evident that the first component is the one that is the most associated with the artefact.

Given the fact that ICA always output the sources up to a permutation. We decided to extract the signal in the FPZ channel (located above the bridge of the noise) to be used as a template to correlate with the sources estimated from ICA.

Following the same approach as the previous exercise, we modified the mixing matrix by zeroing out the components associated with the eyeblink (in the plotted case, column 1). The sources were mixed again with the modified mixing matrix to obtain the denoised EEG signals **Figure 1.10.**

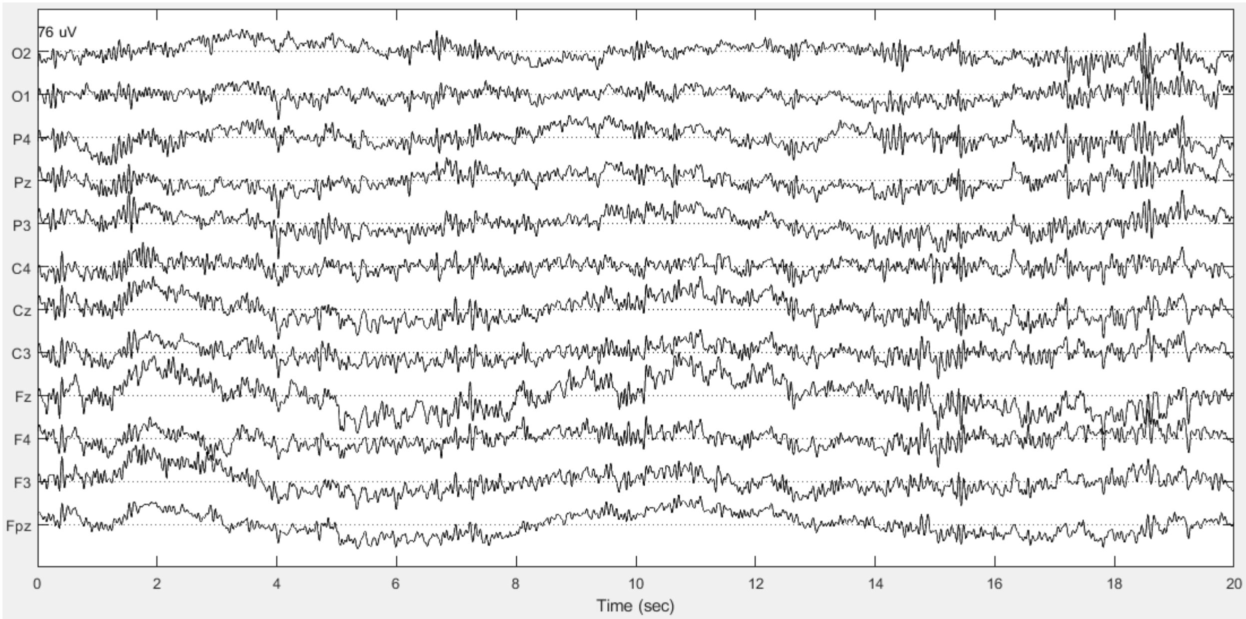

Figure 1.10 *EEG signal for 12 channels (eye blink artefact removed)*

1.3 Conclusions

In this exercise session, we learned about the application of FastICA as a useful tool to decompose a signal and obtain the underlying sources. In general, ICA can separate any mixture of signals if they are linearly mixed and that they do not contain more than two Gaussianily distributed sources. Therefore, it is important to have an estimate of the potential sources contributing to a signal before applying ICA.

FastICA is a simple, robust method that can be used to detect, separate and remove noisy activity in an EEG recording. This can substantially improve the readability of the results and help provide valuable diagnostic information.

<u>References</u>

[1] Aapo Hyvrinen, Erkki Oja. **“Independent Component Analysis A Tutorial”.** April,1999.