ICT for Health
Laboratory # 6
ICA and EEG

Monica Visintin

Politecnico di Torino

2024/25

Table of Contents

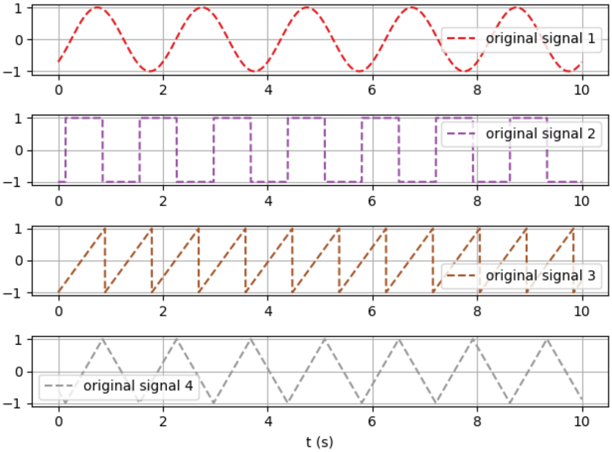
Table of Contents

FastICA in Python [1]

1. In this artificial example, 4 signals are generated: a sinusoidal signal, a square wave, a sawtooth wave, a triangular wave.
2. Note that the probability density function (pdf) of both sawtooth and triangular waves is uniform between the minimum and the maximum; the pdf of the square wave is discrete (two possible values: the minimum and the maximum), the pdf of the sinusoidal signal is $f(x) = \frac{1}{\pi\sqrt{x(1-x)}}$. Thus the pdfs are far from being Gaussian.
3. Each sample of the four signals is multiplied by a 4×4 square matrix (that is randomly generated), to generate four new mixed signals. Task of FastICA is to estimate the original signals from the new mixed signals.
4. In the Python code that you find in DropBox you have the generation of the original and mixed signals, with the measured/estimated pdfs.
5. You have the results obtained with `FastICA`, with plots.
6. Moreover, you have the results obtained with PCA, with plots.

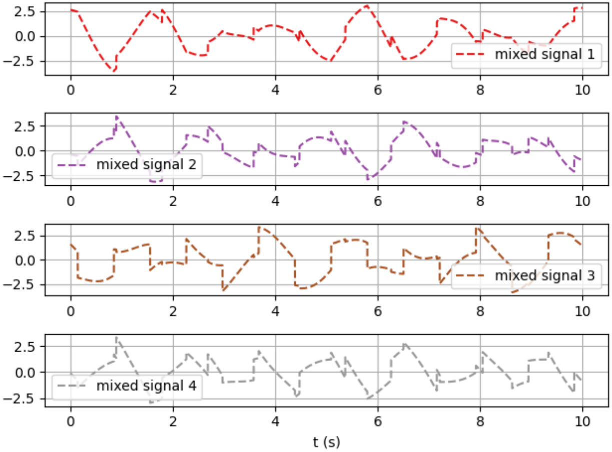
FastICA in Python [2]

Original signals



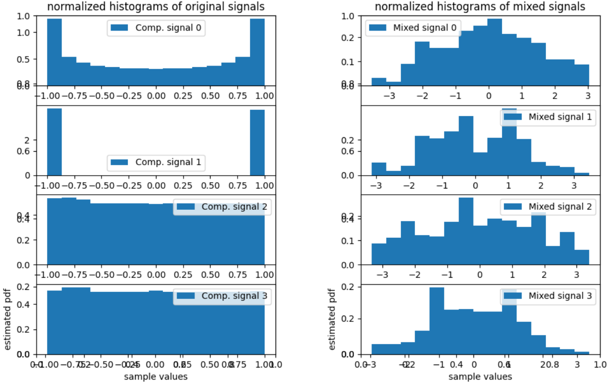
FastICA in Python [3]

Mixed signals



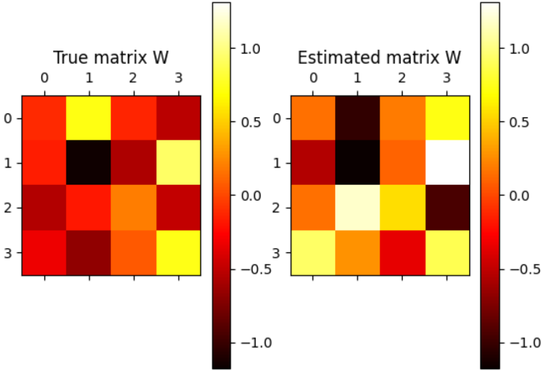
FastICA in Python [4]

Histograms



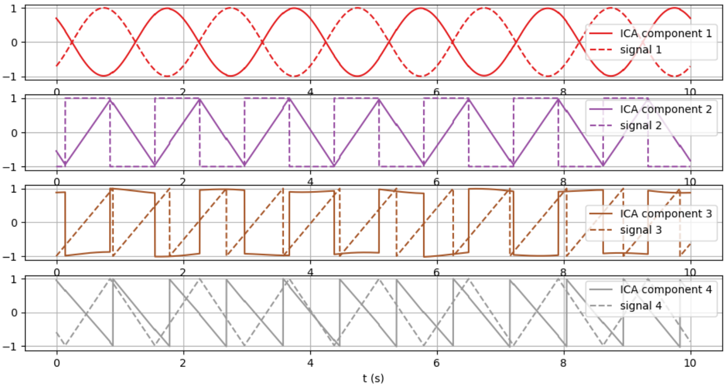
FastICA in Python [5]

FastICA estimation of matrix W



FastICA in Python [6]

Fast-ICA



FastICA in Python [7]

PCA

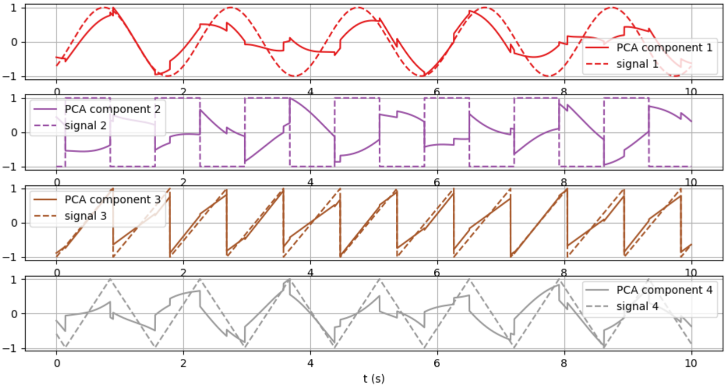


Table of Contents

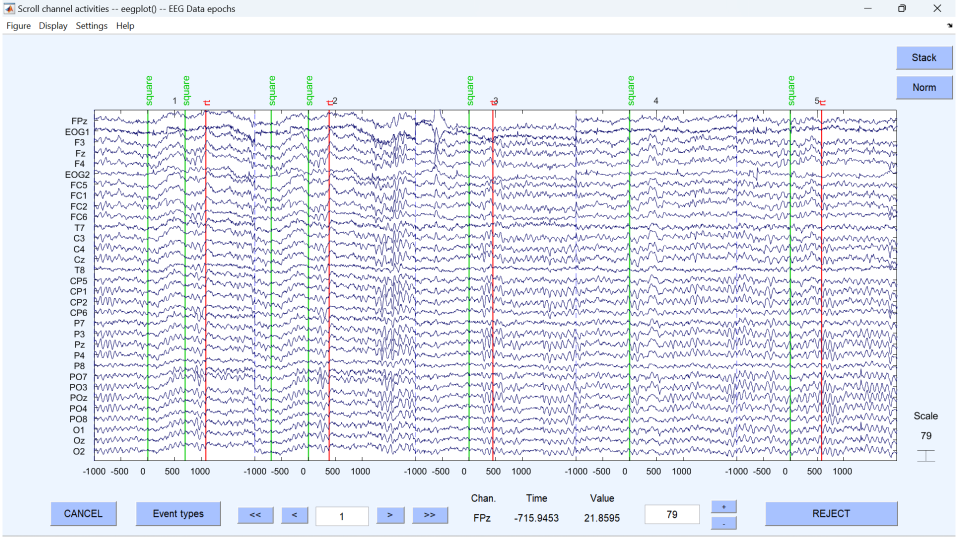
EEGLAB installation

EEGLAB is an add-on of MATLAB, therefore first you need to have MATLAB installed with the Signal Processing Toolbox installed.
You can in principle install EEGLAB from the home menu of MATLAB, by selecting Add-Ons, Get Add-Ons and searching for EEGLAB. However in 2023 there is a wrong package on MATLAB site and this procedure gives problems.
The safest way is to download EEGLAB from the EEGLAB website `https://sccn.ucsd.edu/eeglab/downloadtoolbox.php`, unzip the downloaded file in a folder, move with MATLAB in that folder (icon browse for folder in the home menu).

Starting EEGLAB [1]

1. From the MATLAB command window, simply write `eeglab` to start. A window will appear, with all the commands necessary to run EEGLAB.
2. The first step is to upload an EEG file. For this demo, we will use one of the available datasets: in the menu select File/Load existing dataset move to folder `sample_data` and then select `eeglab_data_epochs_ica.set`. The experiment was the following: the person was in front of a screen and had to click a button when he/she saw a square in one of 5 possible locations on the screen. In the plot of the next slide (Plot/Channel data (scroll)), you see the 32 EEG signals and green vertical lines corresponding to the times at which the square appeared and red vertical times when the button was pressed. Actually, what you see is the juxtaposition of 3 second clips: from one second before to two seconds after the appearance of the square on the screen. The number of experiments (epochs) is 80.

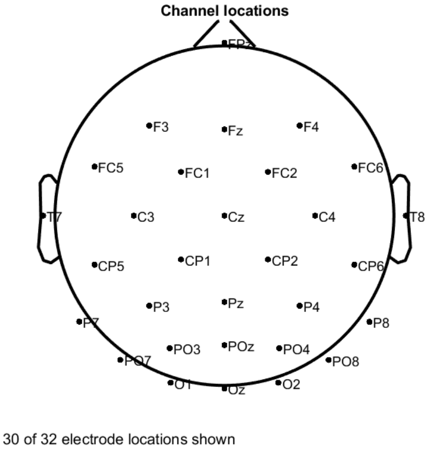
Starting EEGLAB [2]



Starting EEGLAB [3]

This dataset includes the locations of the electrodes, for other datasets locations should be provided separately. There are 32 electrodes, whose position you can see using Plot/Channel locations/By name

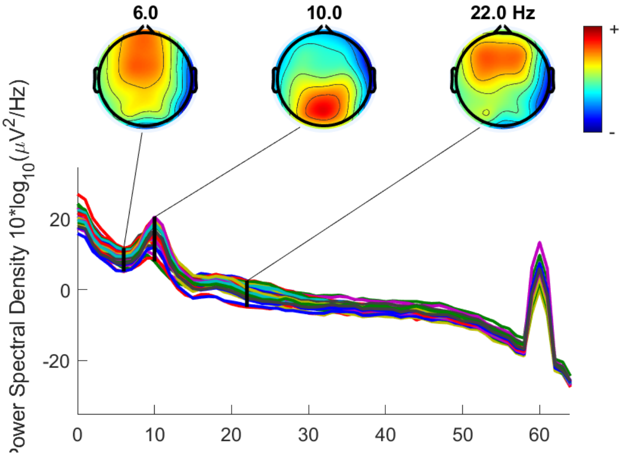
Starting EEGLAB [4]



Starting EEGLAB [5]

3. It is necessary to remove the linear trends and the interference from the mains, at 60 Hz. To see the presence of the interference, see the power spectra (i.e. how the signal power is distributed in frequency) using Plot/Channel Spectra and Maps and setting the frequency range to plot (5th parameter) as `0 64`. In fact, the EEG samples were taken at sampling frequency 128 Hz and the maximum frequency that can be seen in the power spectrum is 128/2=64 Hz. Note that each EEG signal in each “epoch” of 3 seconds corresponds to 384 samples. Power spectra are shown in dBs: $10 \log_{10}$ of the power spectrum. Students with no background on Fourier transforms are invited to listen to the videos in `https://eeglab.org/tutorials/08_Plot_data/Time-Frequency_decomposition.html`

Starting EEGLAB [6]



The meaning of the figure will be explained later on.

Starting EEGLAB [7]

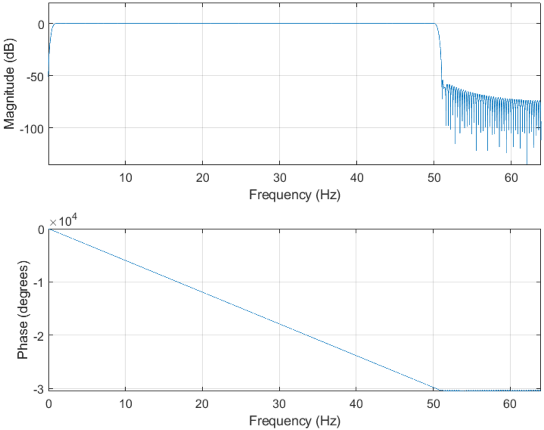
To remove the linear trend, a high pass filter with corner frequency equal to 1 Hz can be used. To remove the mains interference you can use a notch filter (that only removes the 60 Hz component) or a low pass filter with bandwidth for example equal to 50 Hz. Note that neurologists identify (and are interested in):

- δ waves (deep sleeping, not in this case) at 0.5-4 Hz Hz
- θ waves (drowsiness) at 4-7 Hz
- α waves (relaxation) at 8-15 Hz
- β waves (active thinking) at 16-31 Hz

Therefore, the resulting band pass filter, that allows to pass frequencies from 1 to 50 Hz, is correct.
To perform the filtering, use Tools/Filter the data/Basic FIR filter, and insert 1 and 50 as the first two parameters. Note for the students with a signal processing background: since reaction times are fundamental, it is necessary to use a filter that introduces a delay that is constant over the entire frequency range, and a linear phase FIR filter is therefore the correct choice. A new dataset is generated and a new name is asked for, for example `EEG Data epochs Filtered`

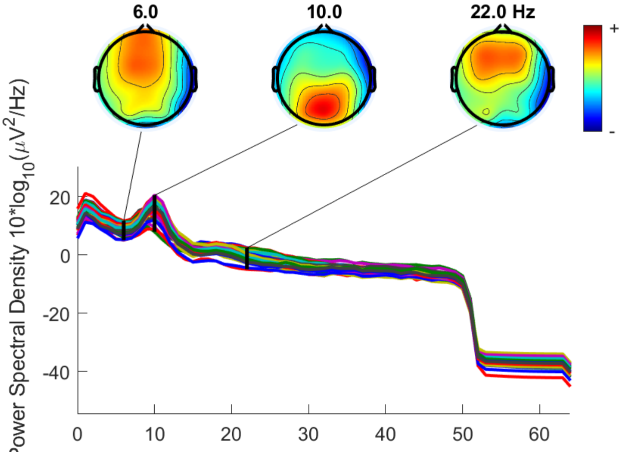
Starting EEGLAB [8]

Filter transfer function



Starting EEGLAB [9]

Power spectra of the signals after filtering.



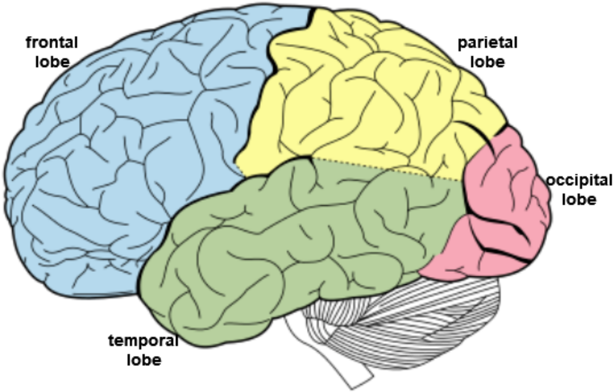
Starting EEGLAB [10]

Meaning of the the figure with power spectra:

- For each of the electrodes (at a known position on the skull) you measure the power spectrum.
- You plot the value of the power spectrum at selected frequencies (6, 10, 22 in the figure) in a 2-D image of the skull and using a color scale (you show, as color, the power spectrum value for each electrode, whose position is known), then you interpolate these values so that you color the entire 2D skull.
- θ waves (6 Hz) are more concentrated on the frontal lobe.
- α waves (10 Hz) are more concentrated on the parietal and occipital lobes.
- β waves (22 Hz) are more concentrated on the frontal lobe.

However, remember that the electrodes capture the sums of the signals generated inside the brain in different locations and therefore there is not such a strict correspondence between the locations shown in the power spectra plot and the true brain lobes involved in the task.

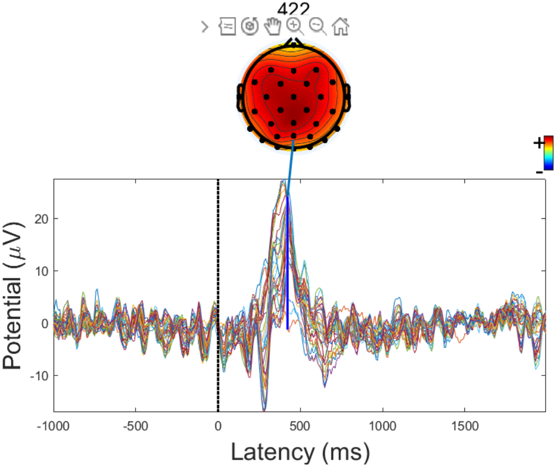
Starting EEGLAB [11]



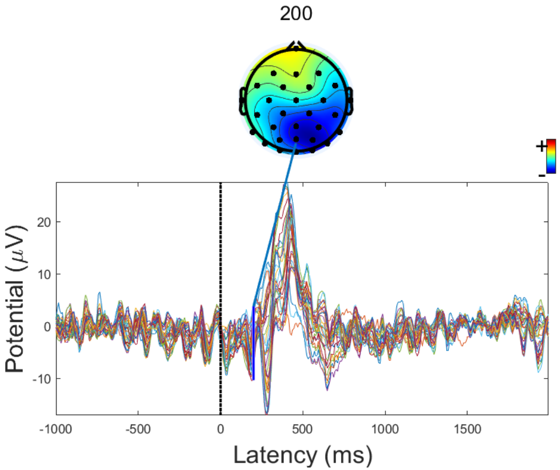
Other plots [1]

1. It is possible to view the ERPs (Event Related Potentials) that are obtained for each of the 32 electrodes by averaging the signals in each epoch (from -1 to 2 seconds around the appearance of the square on the screen): Plot/Channel ERPs/With scalp maps. In this case, the plot on the 2D scalp shows (in color scale) the amplitude of the averaged signal at a given latency (422 ms) from the appearance of the square on the screen. It is possible to change the latency by modifying the second input parameter from NaN to 200 (for example).

Other plots [2]



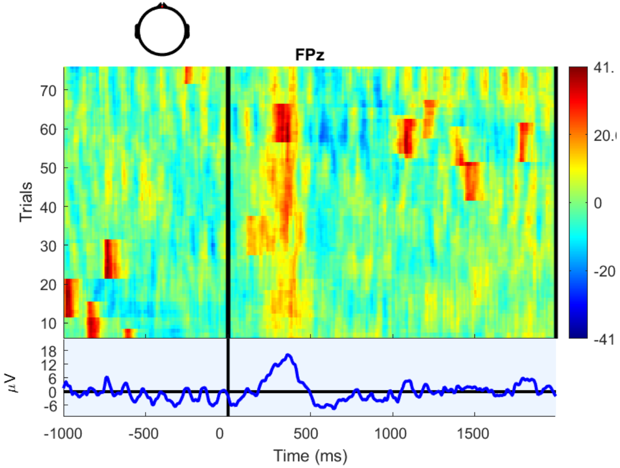
Other plots [3]



Other plots [4]

2. With Plot/Channel ERP Image, it is possible to see, for the selected electrode, the average ERP signal and, in color scale, each of the 80 signals in the experiment (out of which the average is evaluated). The electrode in the figure is number 1, middle front.

Other plots [5]



Other plots [6]

3. With Plot/Channel time-frequency the time-frequency plot is obtained for the selected sensor (again number 1). More details are given in the videos at `https://eeglab.org/tutorials/08_Plot_data/Time-Frequency_decomposition.html`, but, in a nutshell, the time-frequency plot is obtained by selecting a time window of samples (for example from sample 10 to sample 110), evaluate the Fast Fourier Transform (FFT) and plot the absolute values of the FFT as colors at the time corresponding to sample 10, then repeat the process on a subsequent time window (from sample 11 to sample 111), etc.
   In the next figure you see that at about 300 ms from the appearance of the square, signal for electrode FPz has mainly components at low frequencies (below 10 Hz, θ waves).

Other plots [7]

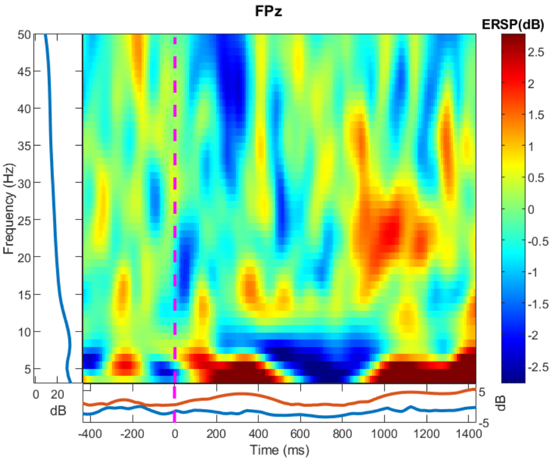


Table of Contents

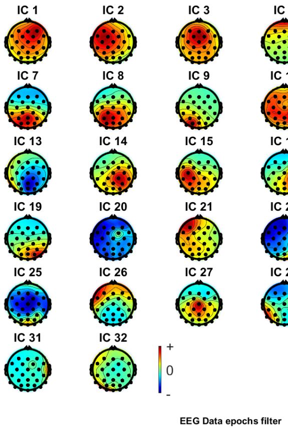
Independent Component Analysis [1]

1. Go to Tools/Decompose Data by ICA and leave the parameters unchanged. Independent Component Analysis starts (note that it is not FastICA, but Infomax ICA). All the 32 components (since there are 32 sensors) are generated in parallel using the gradient algorithm and maximizing the joint entropy (which corresponds to minimize the mutual information) after a nonlinear transformation (sigmoid) and quantization, more details in the videos: `https://www.youtube.com/watch?v=kWAjhXr7pT4&list=PLXc9qfVbMMN2uDadxZ_0EsHjzcRtlLNxc`
   Note that, while the algorithm runs, MATLAB shows in the command window the learning rate (which is decreased during the optimization process), the norm of the difference between the current matrix $\mathbf{W}$ and the previous one, and the angle between the current $\mathbf{W}$ and the previous one (basically, you evaluate the inner product, which can also be obtained as product of the norms times the cosine of the angle). If the angle is 180 degrees, it means that the algorithm is moving around the optimum solution, but with a too large learning coefficient and it is necessary to decrease it. In the implemented algorithm, if the angle is larger than 60 degrees, then the learning rate is multiplied by 0.98.

Independent Component Analysis [2]

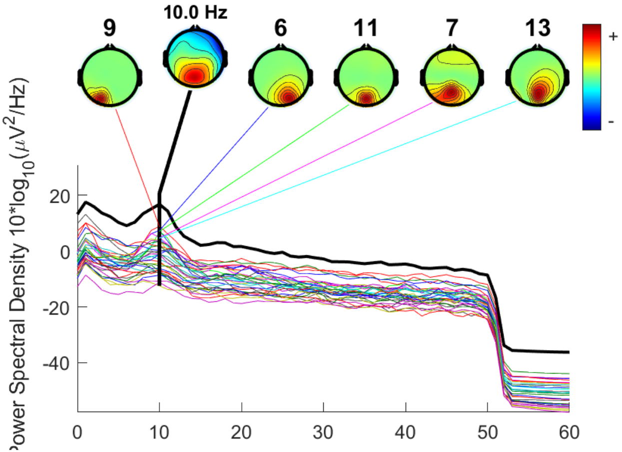
2. Once the components are found, it is possible to generate the plots.
   - Using Plot/Components maps/In 2D you can see the rows of matrix $\mathbf{W}$ that allow to generate each component from the original signals (remember that each original signal is associated with a location on the scalp).

Independent Component Analysis [3]



Independent Component Analysis [4]

- Using Plot/Components spectra and maps you can see how the overall 10 frequency power is divided into the various ICA components with associated map.



Independent Component Analysis [5]

- The most important point is that you can select components that are artifacts (like heart rate, eye blinking, etc). In the example, note component 4, which is almost always equal to zero, but shows a peak at time -750 ms in the middle of the plot (probably eye blinking). By removing artifacts, it is easier for the neurologist to interpret the EEG and diagnose illnesses.