OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

medbo®

Correction of TMS-evoked artifacts in the EEG

Bachelor thesis

At the
Ostbayerischen Technischen Hochschule Regensburg
and Medbo Zentrum für Neuromodulation

by
Guillem Boada Gardenyes

Faculty:	Informatik und Mathematik
Studies:	Biomedical Engineering
Examiner & supervisor:	Prof. Dr.-Ing. Axel Doering
External supervisor:	PD Dr. Martin Schecklmann
Second examiner:	Prof. Dr. Christoph Palm

August 2017

Statement

1. I know that this copy of the bachelor thesis passes into the ownership of the Regensburg University of Applied Sciences as an exam performance.

2. I hereby declare that I composed this bachelor thesis self-dependent and I did not use the thesis for other exam scores, I did not use other sources and tools which are not specified and I marked literal and corresponding quotations.

_____

Place, Date and Signature

Student: Guillem Boada Gardenyes
Student ID: 2906874

Work time: 06.03.2017 – 15.08.2017

“Mientras el cerebro sea un misterio, el universo continuará siendo un misterio también”

Santiago Ramón y Cajal (1852-1934), Nobel prize in Medicine

Thanks to:

Prof. Dr.-Ing. Axel Doering for his constant supervision and guiding.

PD Dr. Martin Schecklmann for his easy-going way of facing problems.

Timo Fuchs for his frequent feedback and advising.

Víctor López and Max Schmaußer for offering themselves as subjects for the measurements.

TMS-Labor for his friendly reception and daily positive working environment.

Family and friends for their unconditional support.

Regensburg, August 2017 Guillem Boada Gardenyes

Abstract

In the research of the brain, the most complex organ of the human body, its function can be studied through the analysis of Evoked Potentials (EP). This evoked activity can be reproduced in a diverse way and recorded with an Electroencephalogram (EEG). One technique to evoke potentials is the Trancranial Magentic Stimulation (TMS), but the action of it results in many artifacts in the EEG that overlap the EP and difficult its proper analysis. The aim of this thesis is to analyze the line noise artifact in TMS-EEG data by researching its origins and, then, introduce and evaluate the existing correction methods. Finally, an alternative correction algorithm based on a subtraction using an extracephalic reference signal and averaging is proposed.

Table of contents

1. Introduction

In the research of the brain, the Electroencephalogram (EEG) is well-known as a basic and efficient technique to record the neural activity. Apart of this standard diagnosis method, techniques aiming to stimulate the brain have appeared, like the Transcranial Magnetic Stimulation (TMS) which is able to generate a magnetic field close to the scalp that induces a current in a specific part of the brain. The concurrent use of both techniques (EEG-TMS) is becoming popular as a powerful tool to study brain connections as it enables to induce changes in the neural activity and to record those changes. However, this combination of techniques is challenging because the TMS causes interferences to the EEG, this undesired electrical activity hides the neural activity. In this study, the origin and characteristics of the line noise artifact in EEG-TMS recordings are analyzed. Then, after a review on the existing methods for correction, a new line noise artifact correction algorithm is presented.

2. Background

In order to understand this work, it is necessary to first comprehend some basic concepts of the elements involved in this topic: Brain, Electroencephalogram (EEG), Transcranial Magnetic Stimulation (TMS), Artifacts and Signal Processing. The interrelation between these elements is clarified in the next Figure 1.

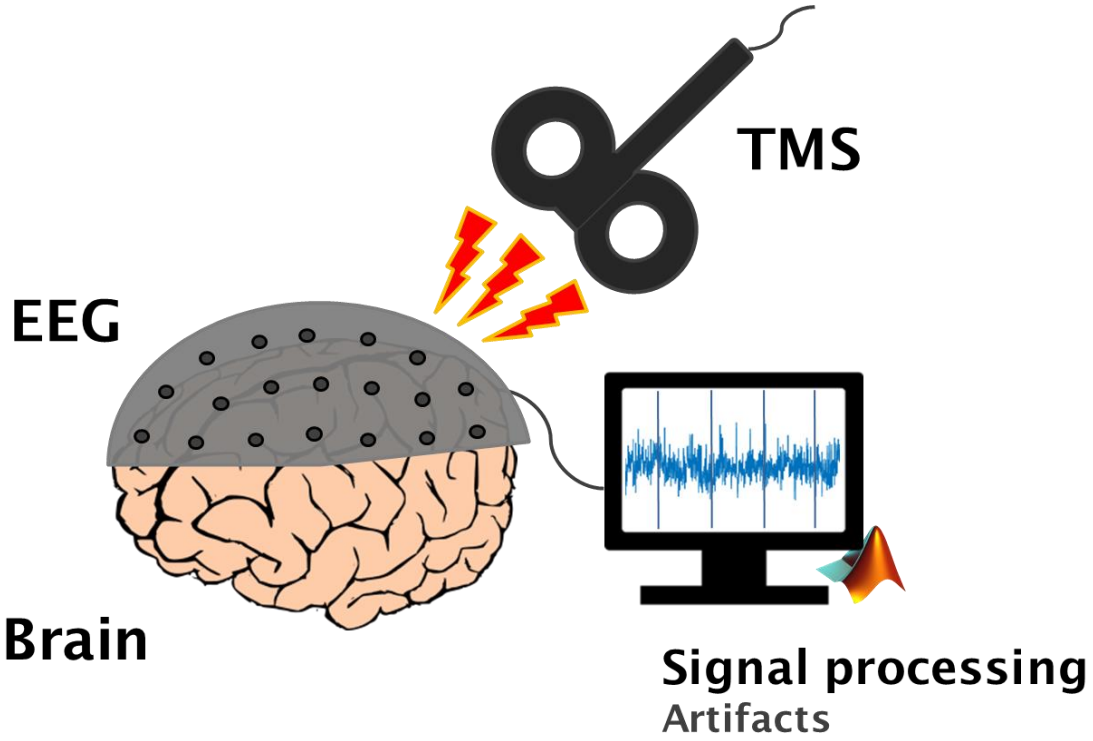

Figure 1: Overview of the main elements involved in the EEG-TMS topic.

The EEG records the neural activity from the brain mixed with the activity induced by the TMS function. This TMS induced activity called artifacts can be faced with Signal Processing tools.

2.1 The brain

The brain is the most important organ of the body as it controls the vital functions and rises up consciousness. The exact way in which this is managed is still unknown, however, it has been proved that the neurons produce and transmit electrical impulses, the so-called action potentials. These currents move from one cell to another across the synapse, regulated by the neurotransmitters. As the EEG is placed on the scalp or external surface of this organ, it records all the neural activity of the brain, but, mostly, the activity from the cortical pyramidal cells found in the cortex, these are called post-synaptic dendritic currents.

Every region of the brain has some specific functions, for example, the motor cortex (Figure 2, in red) is in charge of the planning, control and execution of voluntary movements.

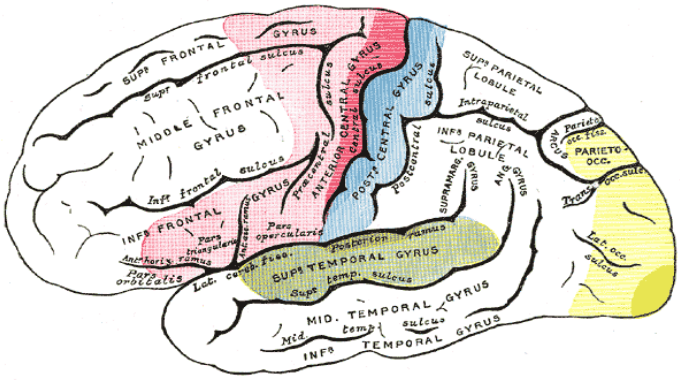

Figure 2: Motor cortex in red, sensory cortex in blue, auditory cortex in green and visual cortex in yellow. Image 756 from Gray (1918).

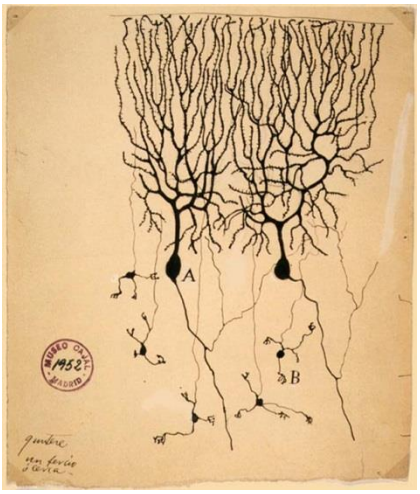

Figure 3: First neurons drawing by Santiago Ramón y Cajal (1899).

2.2 Electroencephalogram (EEG)

The electroencephalogram or electroencephalography (EEG) is a continuous monitoring method for recording the brain electrical activity on the scalp, it is typically non-invasive. These electrical waves are mostly originated due to the neurons found in the most superficial layer of the brain. Despite the major part of the activity comes from the cerebral cortex, all the other structures of the brain underneath the cortex are also affecting the record. As a single neuron potential cannot be perceived, the received signal actually comes from the activity of a group of neurons, this activity is called Local Field Potential (LFP). The EEG is registered on the scale of microvolts (μV).

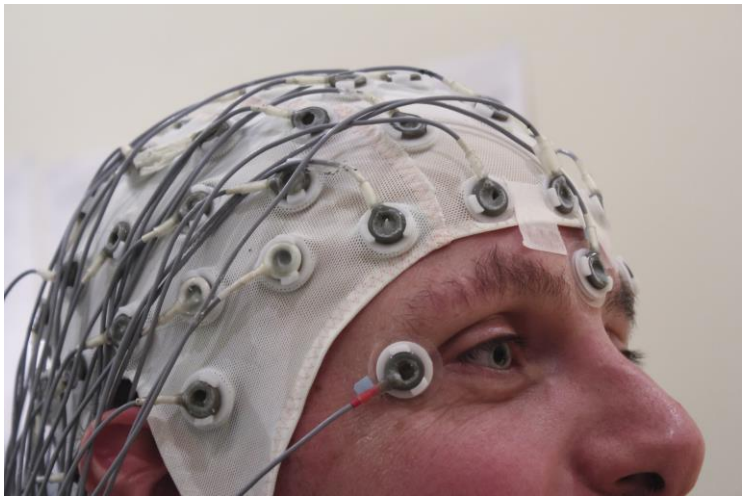

Figure 4: Photo by C. Hope retrieved from T. Sheerman-Chase (2012).

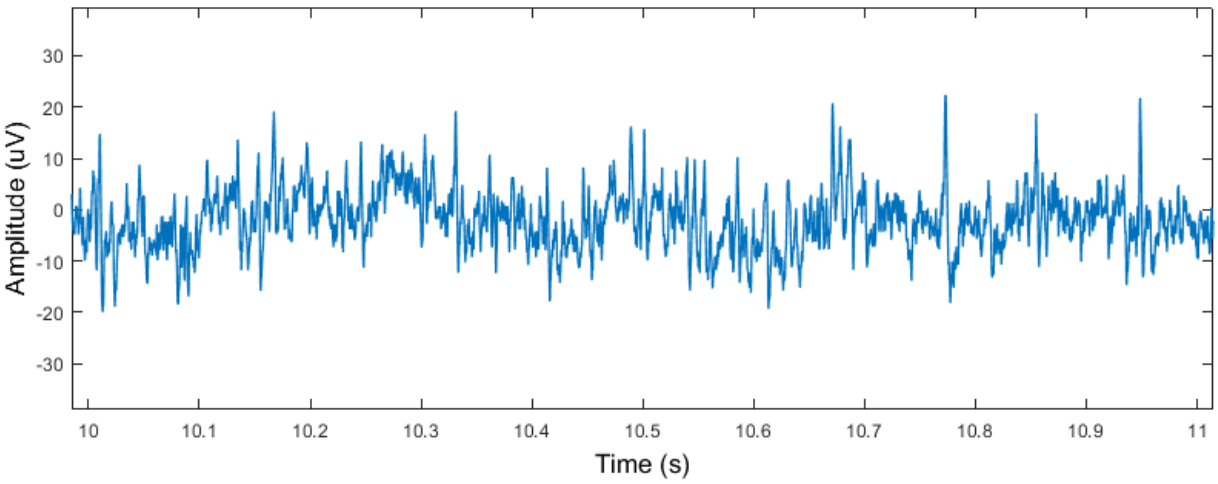

Figure 5: Standard EEG plot from own experiments, channel C3.

The technique to record the EEG consists of placing several electrodes over the scalp, in a non-invasive version, and grouping them into pairs, each pair is denominated derivation. Afterwards, the electrical potential between the electrodes of each derivation is obtained using a differential amplifier, as shown in Figure 6, and graphed on a moving paper using a polygraph or recorded digitally in a computer.

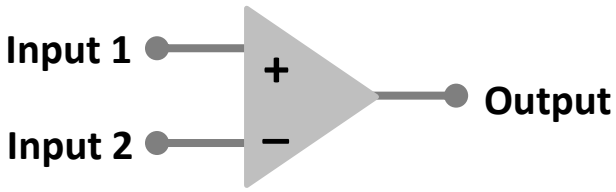

Figure 6: Differential amplifier.

The input 1 and input 2 are the potentials received by the electrodes and the output is their difference. The next Figure 7 shows how the differential amplifier performs its action.

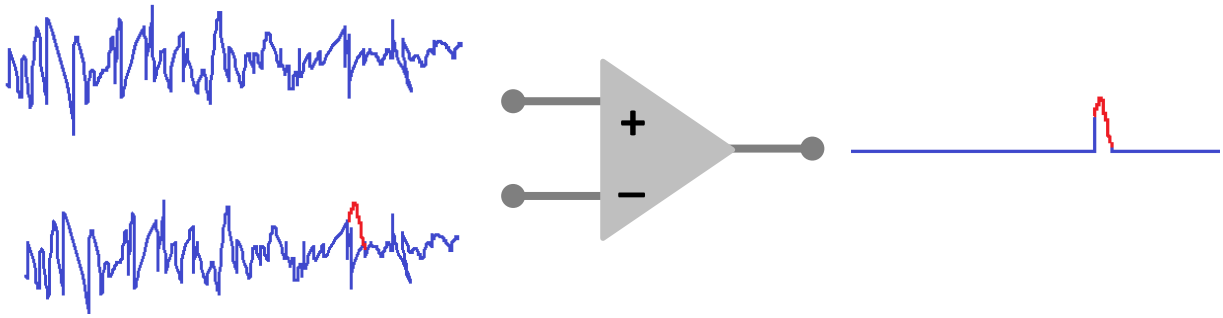

Figure 7: Differential amplifier function.

The EEG technique is well-known for its speed, being able to record voltage samples at a lower rate than milliseconds, and for its passivity, as it does not apply any current or action to the body in its performance. However, it is limited by its poor spatial resolution that does not allow the user to know where the activity is exactly coming from. This handicap can be tackled with Signal Processing.

In the medical field, the EEG is commonly known as a diagnostic tool for diseases like epilepsy, but in this work it is of interest to focus on its use for research. Thanks to its easiness of practice, the EEG has become the most popular tool in all the fields that involve the brain function.

This tool is used with scientific purposes to study the electrical activity within the brain and discover what role does every region of the brain play when controlling the rest of the human functions, from the motor actions to the psychological activity. This kind of research is done by studying the Event-Related Potentials (ERP); those are the reactions of the brain induced by an external stimulus that can be identified on the EEG. The stimuli can be from any sensorial source, like visual or auditory; TMS manages a stimulus as it induces current to the brain. The EEG is also involved in research studies with further objectives as understanding the phenomena of intelligence and subjective consciousness.

2.2.1 Electrode locations

In order to describe the location of the scalp electrodes there is a standardized pattern named International 10/20 System that allows comparing different EEG recordings. This arrangement of electrodes is based on the relation between the positions of the different areas in the cerebral cortex and four anatomical landmarks: the intersection between the frontal bone and the two nasal bones, called nasion; the highest point of the external occipital protuberance found on the back of the head, called inion; and the two pre-auricular points.

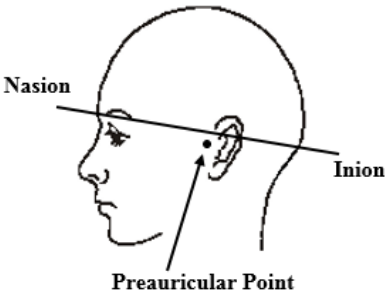

Figure 8: Reference points used in the International 10/20 system.

From those cephalometric landmarks, it is possible to draw lines and establish points relatively to those distances; these are separated by steps of 10 or 20% of the total distance, this is shown in Figure 9.

It exists a nomenclature that, by using letters and numbers, it eases the task of referring to each specific location, as shown in Table 1.

Electrode	Reference location
F	Frontal lobe
T	Temporal lobe
C	Central position
P	Parietal lobe

Table 1: Nomenclature in the 10/20 System.

Every point is referred to a region of the cerebral cortex. Furthermore, the odd numbers are found on the left side and the even ones on the right. The “z” stands for “zero”, those are the locations found on the midline.

It is possible to determine more positions be adding locations in between the existing ones.

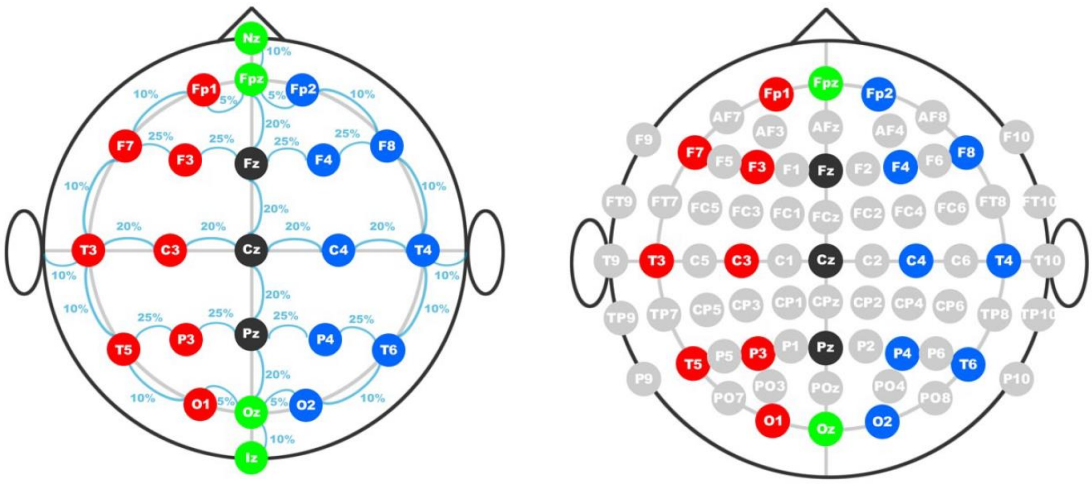

Figure 9: International 10/20 System locations.

In the experiments performed in the TMSLab for this work, 64 locations are used.

2.3 Transcranial Magnetic Stimulation (TMS)

A plastic-enclosed coil of wire is placed close to the scalp and electrical pulses are sent through it generating, according to the Biot-Savart law, a magnetic field oriented orthogonally to the plane of the coil. Then, the magnetic field passes through the skin and skull, inducing an oppositely directed current in the brain, as proved by the Faraday's law. This current causes a change in the transmembrane of the neurons, which leads to the depolarization or hyperpolarization of the neurons and the firing of action potentials.

This effect is achieved by quickly discharging a large capacitor into the coil to produce pulsed magnetic fields between 2 and 3 T, similar strength as Magnetic Resonance Imaging (MRI). The path of this current in the brain is difficult to model because the brain is irregularly shaped and electricity and magnetism are not conducted uniformly throughout its tissues.

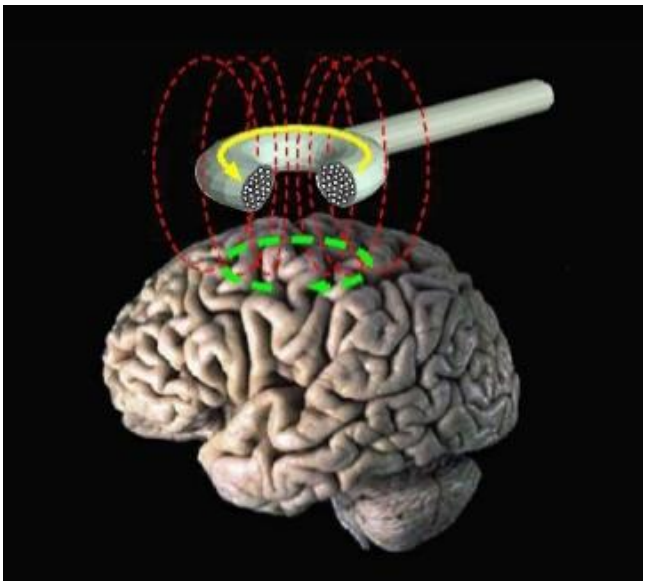

Figure 10: TMS magnetic function with a circular coil. Image from Wasserman (2013).

On one hand, TMS is used for treating neurological disorders like depression, tinnitus and multiple sclerosis. On the other hand, it is widely used to research the brain connection by stimulating a region and analyzing its effects in the patient. The effects can be studied in the EEG, however, it also records the TMS-induced artifacts which complicates its reading.

In the TMS-Labor, a MedTronic MagPro system is used with a figure-eight coil.

2.4 Artifacts

EEG artifacts are understood as unwanted electrical activity arising from other sources different than the brain. These interferences hide the signal of interest, in this case, the neural activity.

The artifacts can be classified depending on their origin. If they come from the subject's body, they are considered physiological artifacts, as an example, there is the eye-induced artifact and the ECG artifact. When the artifact arises from outside the subject, like external equipment and the environment, it is considered an extra-physiologic artifact. The line noise artifact coming from the main line through the lighting and other systems and devices belongs to this group.

The TMS action induces many artifacts in the EEG as muscle-evoked artifact and Auditory Evoked Potentials (AEP), one of them is the TMS-induced line noise artifact.

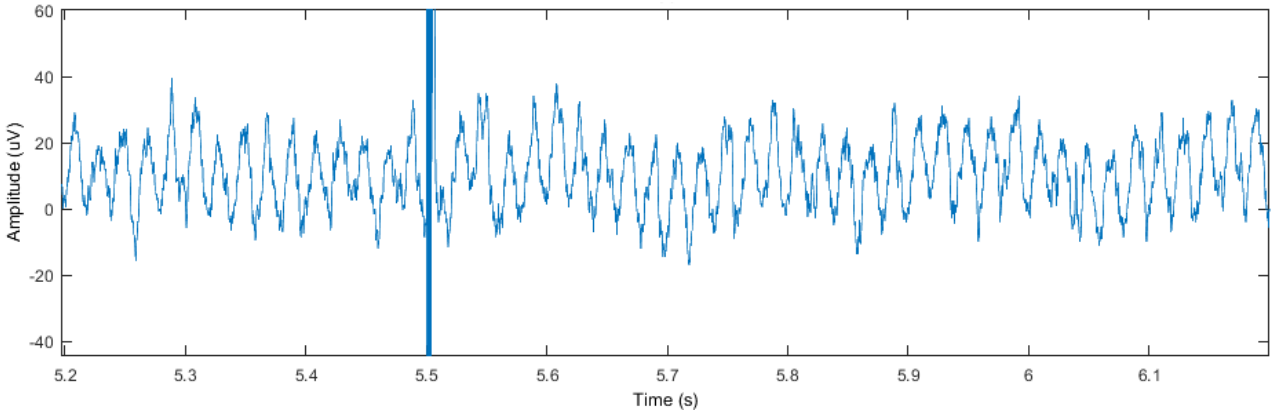

Figure 11: EEG signal with TMS-induced line noise artifact of 50Hz.

This work aims to correct the line noise artifact from EEG signal combined with TMS, so that the neural activity can be read from such a recorded EEG as the one in Figure 11. There exist Signal Processing tools and algorithms that make this correction possible.

2.5 Signal processing

Signal Processing is the science in charge of analyzing and modifying signals, in our case, signals coming from the brain activity and recorded with EEG.

When the first EEGs were measured, it was only possible to write the raw signal on a paper thanks to a galvanometer chart recorder. However, the digitalization enabled many other ways of visualizing and opened the doors to the Signal Processing, which appeared to be a very useful tool for knowing more information of the EEG.

Assuming that the EEG is sampled and saved in a digital format, there are some concepts that are of interest to understand the work done in this thesis. After some background in Signal Processing, the used software and toolboxes will be presented.

2.5.1 Basic concepts

As mentioned, nowadays, EEG systems convert the continuous analog signals to discrete signals which can be saved in data files and, later, manipulated with Signal Processing. The frequency at which samples are recorded is defined by the sampling frequency; for the recordings of this work, the sampling frequency has been 5000Hz.

2.5.1.1 Nyquist–Shannon sampling theorem

The sampling theorem defines a condition for which it is possible to determine a limit sampling frequency for which the discrete signal can save all the information of the continuous signal with a specific bandwidth. This sampling frequency is named Nyquist frequency and computed with the next formula.

$$f_{Nyquist} = \frac{f_s}{2}$$

2.5.1.2 Fourier transform

It is the operation which decomposes a time domain signal in its frequency content. It is a powerful tool to analyze the amount of a specific frequency in a signal. The resulting signal from the transformation expressing in the frequency domain is named spectrum. The spectrum shown on the next Figure 12 reveals a big amount of 50Hz frequency in the example analyzed signal.

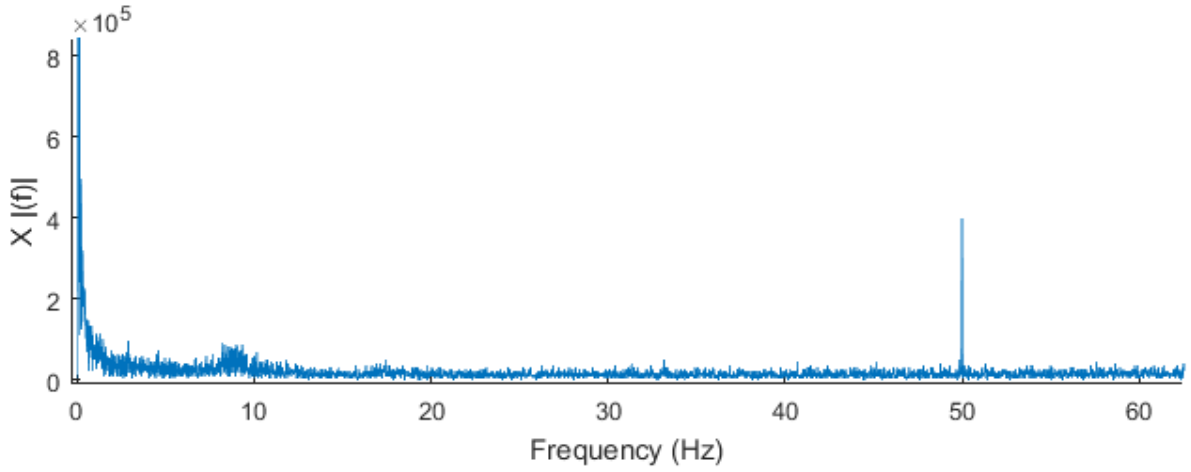

Figure 12: Fourier transform of an EEG signal.

2.5.1.3 Frequency filters

This tool enables the removal or modification of the amplitude of specific frequencies in the signal.

Types depending on its range

By defining the so-called cutoff frequencies, it is possible to determine which specific range of frequencies is desired to be removed or maintained.

- **Lowpass:** The frequency components higher than the cutoff frequency are removed.
- **Highpass:** The frequency components lower than the cutoff frequency are removed.
- **Bandpass:** The range between the two selected cutoff frequencies is maintained.
- **Bandstop:** The range between the two selected cutoff frequencies is removed.
- **Notch filter:** Similar to a bandstop filter, it aims to remove a single frequency component. It is commonly used for removing the line noise at is only found in a punctual frequency.

It is important to highlight that the cutoff frequency does not define a perfect exact limit from which the change is applied, there is a transition range.

Types depending on its response

The impulse response of a filter enables us to classify them into two groups.

- **Finite Impulse Response filter (FIR):** As its name indicates, the impulse response of a FIR filter has a finite duration in time. It can be said that it has a finite settling time.
- **Infinite Impulse Response filter (IIR):** Oppositely, its impulse response has infinite duration.

Each type has its advantages and drawbacks, being a better choice for one case or another.

2.5.2 Tools

In order to apply the Signal Processing methods and analyze the EEG, the software Matlab® assisted by some toolboxes was used.

2.5.2.1 Matlab®

It is a powerful numerical computing program used widely throughout science and engineering. It allows the manipulation of data in matrices and the plotting of it in a very diverse way.

2.5.2.2 EEGLAB - Delorme & Makeig (2004)

EEGLAB is an open source environment for EEG signal processing that runs in Matlab as a GUI.

2.5.2.3 TMSEEG - Atluri et. al. (2016)

TMSEEG is a toolbox for Matlab that provides a defined but adaptable pipeline for TMS-EEG data.

2.5.2.4 TMS EEG Signal Analyser (TESA) - Rogasch et. al. (2016)

TESA is an extension of the EEGLAB that provides useful functions for TMS-EEG data processing.

3. Line noise artifact

The methodology to study the line noise artifacts consists of a first analysis of the problem, with a research of its origins and characteristics. Later, the existing correction pipelines are presented and, finally, the proposed algorithm is described.

3.1 Analysis

In order to analyze the line noise artifact, a couple of experiments were performed. With these experiments, it was aimed to record data with specific conditions and an extracephalic reference signal which will be later used for the proposed algorithm.

3.1.1 Origin of the line noise

3.1.1.1 Hypothesis

It is assumed that the line noise artifact might be originated from different sources:

- **Environmental line noise** coming from the AC power lines, lighting and equipment.
- **Antenna effect** caused by the TMS coil shape.
- **Current leaks** from the TMS device.
- **TMS device function** when generating and emitting the pulse.

3.1.1.2 Experimental procedure

The experimental procedure is described, including the information regarding the tools and conditions that were involved.

First of all, the EEG system (Brain Amp DC amplifier, Brain Products) using a compatible EasyCap of the proper size for the patient with 64 ring electrodes (62 EEG + 1 Ref on FCz + 1 GND on AFz) and the TMS system (MedTronic MagPro) with a figure-eight coil are installed; then, the skin under the electrodes is cleaned by applying isopropyl-alcohol 70% using wooden cotton swabs. Later, EEG paste (ABRALYT HiCl) is introduced through the holes of the ring electrodes to fill up the electrode-skin space; this is done with syringes. Afterwards, the impedances are checked and brought underneath 5kΩ by scratching the skin with the wooden part of a swab. Finally, an additional electrode is attached with tape on the backward side of the coil and connected to the Iz position input. The exact place in which it is placed and the way it is placed can be seen in Figure 14.

The recordings are performed to a 21 years old male subject who is sitting comfortably on a chair. During the recording time, his eyes are opened looking forward to a fixed point and he stays quiet. Each recording lasts 2.5 minutes. Taking into account that there is a total of 10 recordings, the recording time was about 25 minutes and the whole preparation and procedure lasted around 1 hour and 30 minutes. The sampling rate is 5000Hz. During the recording, the subject is listening to white noise thanks to the anti-radiation earplugs, this way, the Auditory Evoked Potentials (AEP) are avoided. Regarding the recordings in which the coil is placed on the head, it always aims the position C3. Finally, for the recordings in which there is stimulation, the TMS pulse is done in a nearly periodic way between 4 and 6 seconds. So, that the recordings end up by containing 30 TMS pulses. The way in which the coil is placed on the head of the subject can be seen in Figure 13.

Figure 13: Experimental montage.

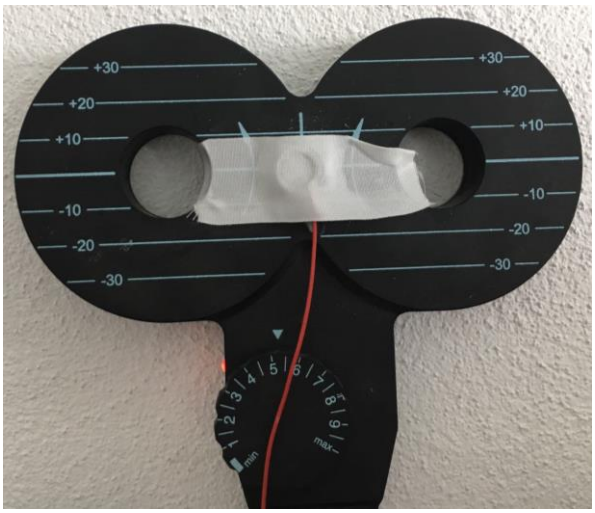

Figure 14: Extracephalic reference electrode location.

3.1.1.3 Recorded data

For every source, there has been selected a set of conditions that could give a meaningful difference to prove the influence of each source to the artifact.

- **Environmental line noise:** As this is an already proved source, there is no need to prove its evidence. It will be recorded in this first recording and will be used as a control recording to be compared with the other ones.

#	On head/air	plugged / unplugged	Power on/off	abled / disabled	Stimulation yes/no	Intensity (%)
1	air	unplugged	off	disabled	no	-

Table 2: Conditions showing evidence of the environmental line noise source.

- **Antenna effect:** The meaningful condition is the position of the coil, whether on the head or far from the patient, “air”.

#	On head/air	plugged / unplugged	Power on/off	abled / disabled	Stimulation yes/no	Intensity (%)
1	air	unplugged	off	disabled	no	-
2	head	unplugged	off	disabled	no	-

Table 3: Conditions showing evidence of the antenna effect line noise source.

- **Current leaks:** Considering that these leaks come from the device, there are many conditions that affect the function of the device, hence, the resulting current leaks. With the device unplugged, it is possible to affirm that there are no leaks, with plugged, they may appear. With power on, they should appear more, and with abled, they may appear even more. “Abled” is called to the condition of turning on the device from the front screen. For abled, different intensity selections are done to see if there is any change even though there is no stimulation.

#	On head/air	plugged / unplugged	Power on/off	abled / disabled	Stimulation yes/no	Intensity (%)
2	head	unplugged	off	disabled	no	-
3	head	plugged	off	disabled	no	-
4	head	plugged	on	disabled	no	0
5	head	plugged	on	abled	no	0
6	head	plugged	on	abled	no	30
7	head	plugged	on	abled	no	60

Table 4: Conditions showing evidence of the current leaks line noise source.

- **Stimulation effect:** Different stimulation intensities will be tested and compared to the recording without stimulation.

#	On head/air	plugged / unplugged	Power on/off	abled / disabled	Stimulation yes/no	Intensity (%)
5	head	plugged	on	abled	no	0
8	head	plugged	on	abled	yes	0
9	head	plugged	on	abled	yes	30
10	head	plugged	on	abled	yes	60

Table 5: Conditions showing evidence of the stimulation effect line noise source.

All the recordings performed in the experiment are summarized in the next Table 6 indicating their exact conditions and given file name. This data can be found in the folder “Files/LN_analysis/Raw”.

#	File name	On head/air	plugged / unplugged	Power on/off	abled / disabled	Stimulation yes/no	Intensity (%)
1	exp2_air_unplug_powerOff_noStim	air	unplugged	off	disabled	no	-
2	exp2_head_unplug_powerOff_noStim	head	unplugged	off	disabled	no	-
3	exp2_head_plug_powerOff_noStim	head	plugged	off	disabled	no	-
4	exp2_head_plug_powerOn_disabled	head	plugged	on	disabled	no	0
5	exp2_head_plug_powerOn_noStim0	head	plugged	on	abled	no	0
6	exp2_head_plug_powerOn_noStim30	head	plugged	on	abled	no	30
7	exp2_head_plug_powerOn_noStim60	head	plugged	on	abled	no	60
8	exp2_head_plug_powerOn_yesStim0	head	plugged	on	abled	yes	0
9	exp2_head_plug_powerOn_yesStim30	head	plugged	on	abled	yes	30
10	exp2_head_plug_powerOn_yesStim60	head	plugged	on	abled	yes	60

Table 6: Summary of the recordings with their corresponding file names.

3.1.1.4 Analysis procedure

In order to know how a certain condition affects to the magnitude of the line noise, the presence of the 50Hz component from every recording will be measured and compared. This will be done in the frequency domain, as it gives a clear result for the overall signal. In every case, an equal length of 2 minutes and 30 seconds was recorded. However, for the procedure, only 1 minute is analyzed because of computational memory reasons.

First of all, the data will be managed and preprocessed using EEGLAB to get the variables that can be used in Matlab. Then, some more preprocessing will be applied to remove the baseline of the data. Later, the recordings with stimulation pulses will be processed for removing the pulses so that proper spectrums can be computed. Finally, the line noise peak maximum for each spectrum will be found. The code *algorithm_analysis.m* in charge of the line noise artifact analysis can be found in Annex II and in "Files/LN_analysis".

3.1.1.4.1 Load data and preprocessing

The used data are the experimental recordings presented in Table 6. All the recordings from "Files/LN_analysis/Raw" are imported with EEGLAB toolbox following the steps explained on the script. For the recordings with stimulation #9 and #10, epochs are extracted from -1 second to 3.5 seconds with respect to the pulse trigger location. So, the pulse is located at the second 1. The recording #8 with stimulation but intensity zero did not show the TMS typical pulses, so it will not be epoched. The instructions for extracting epochs with EEGLAB are also included in the script.

Then, the variable *ALLEEG* including all the recordings managed in EEGLAB is saved as *data_analysis.mat* and it is also found in "Files/LN_analysis".

As another preprocessing step, the baseline from all channels is removed using the function *detrend()*, as it is thought that this may show a nicer display of the data when being compared.

3.1.1.4.2 Removing TMS pulses

The pulses are removed using the own written function *removePulses()*, this function can be found at the end of the Annex II, section of "Auxiliary functions" and in "Files/Auxiliary_functions". When applied, it substitutes the TMS pulses by zeros, doing the so-called zero-padding.

```
[EEG_noPulses] = removePulses(EEG,locPulses,rangeLeft,rangeRight,ch)
```

The inputs enable the user to select the range around the pulse that is desired to be zero-padded. After selecting a range from 10 milliseconds before the pulse to 20 milliseconds after the pulse and introducing the location of the pulses found in the variable ALLEEG in "locPulses", the result of this removal is shown in Figure 15.

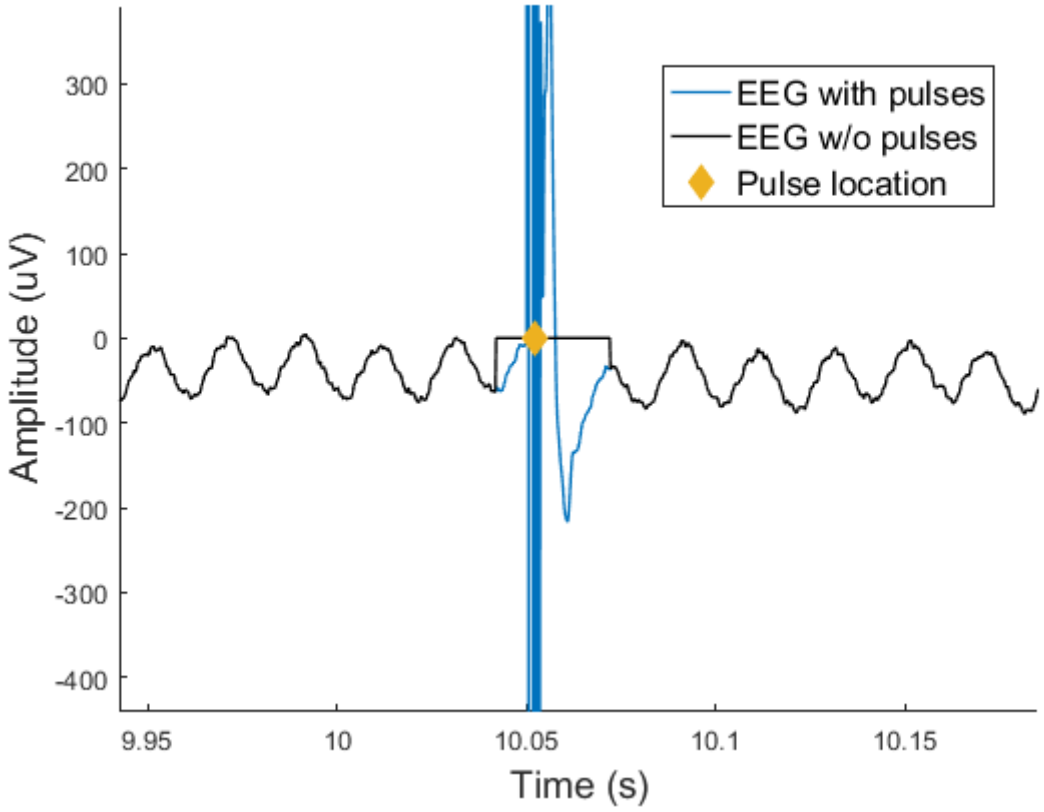

Figure 15: Pulses removal using the *removePulses()* function.

3.1.1.4.3 Computing spectrums

In order to facilitate the computation of a big amount of spectrums, the function *fft2plot()* is written; it can also be found in the Annex II and in "Files/Auxiliary_functions".

```
[X] = fft2plot(x)
```

When a signal is input to this function, it simply returns the half of the absolute FFT which is of interest for being plotted.

Finally, for every computed spectrum, the height of the line noise artifact peak is found using the command *max()* over the segment of FFT around 50Hz. These values will be useful to compare the line noise presence in the recordings and display it in a visual way as bar plots.

3.1.1.5 Analysis results

In this section, the spectrum of every recording and some displays in the temporal domain will be shown and commented. The hypothetical and real results will be compared.

After overviewing several channels of every recording, the channel C3 has been selected to be studied as it is the one aimed by the coil, hence, the one which will show a clearer influence of each condition to the line noise artifact. Anyway, the spectral analysis has been done for all channels, so that it is possible to review it if needed. As an exception, the channel corresponding to FC5 should not be analyzed as it was damaged and it was not possible to get a correct impedance under 5kΩ for the recording.

The recordings are analyzed in the groups previously specified in the study of each hypothetical source.

3.1.1.5.1 Antenna effect

As previously shown in Table 3, the two selected recordings only differ in the condition “air/head”. It is expected that the line noise artifact is higher in the recording in which the coil is placed in the head as the antenna effect should occur in this condition. The Figure 16 shows the spectrum of both signals. The influence of the antenna effect is undeniable.

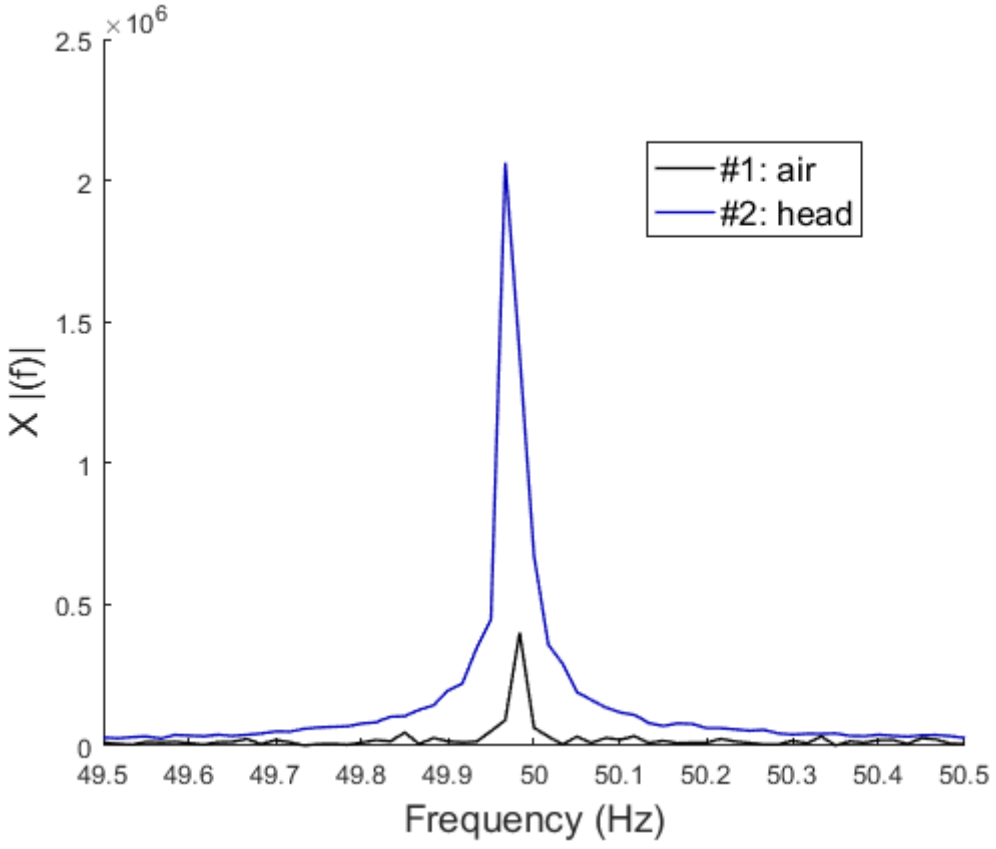

Figure 16: Spectrum of recordings #1 and #2.

3.1.1.5.2 Current leaks effect

For this hypothetical source, there are many conditions which could affect if the device was transmitting current leaks and in which amount would it be doing it. The previous Table 4, collects the set of conditions that define each recording.

The recording #2 with the unplugged TMS device will be used as a control recording as it can be affirmed that, if the device is not connected to the general line, it will not receive any current and, hence, transmit any current.

Figure 17 shows the different results of the line noise peak height for each recording.

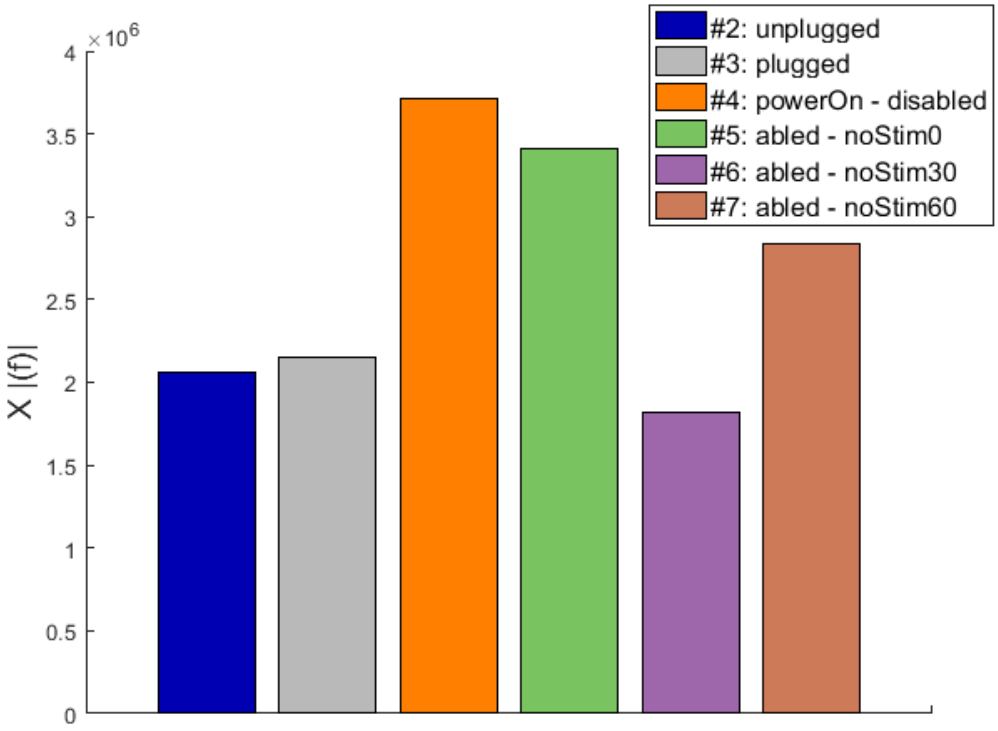

Figure 17: Bar display of the line noise peak heights for the current leaks meaningful recordings.

Firstly, the graph shows that the fact of plugging the device does not allow any current leaks because the height of the line noise between #2 and #3 can be said to keep constant. When powering on the device, it is clearly shown in the recording #4 that the line noise increases considerably. It can be affirmed that this change allows the current to get into the system, resulting in some leaks that end up reaching the EEG electrodes. Abling the TMS device in and selecting a null intensity #5 does not seem to change much the state of the device as the line noise presence is very similar to the recording in which the device is disabled. Finally, for the recordings without stimulation #6 and #7 in which the intensity selection is 30% and 60%, respectively, the analyzed line noise peaks show lower amplitude. For now, it is thought that this may be explained with some internal filtering of the TMS device.

It can be affirmed that there are current leaks and that they occur when powering on the device.

3.1.1.5.3 Stimulation effect

The control recording consists of the recording #5 without stimulation and with selected intensity zero. For this experiment, it has been thought to record stimulation in different intensities and observe how the results vary.

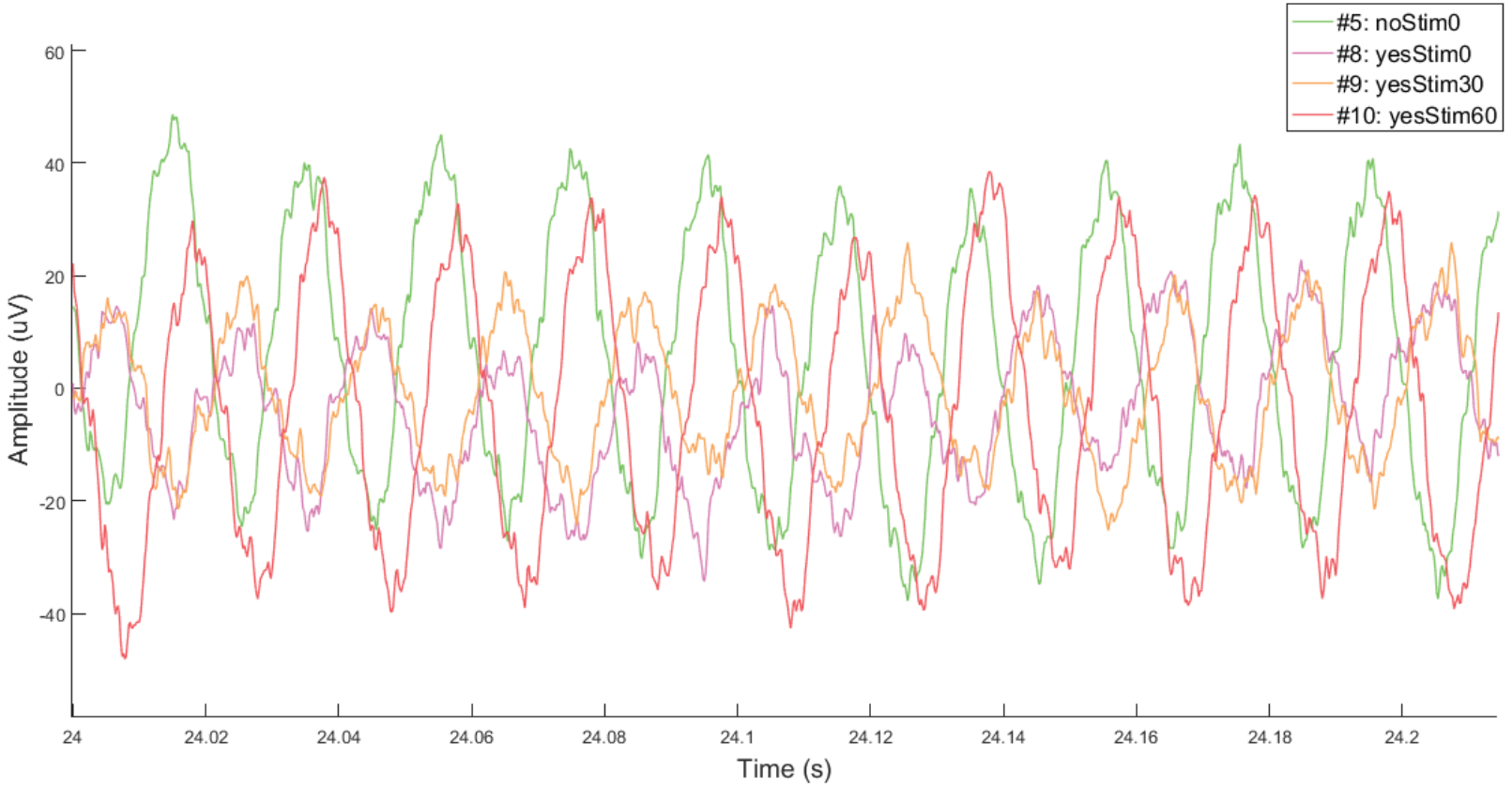

Figure 18: Temporal display of the recordings related with the stimulation analysis.

At first look, it can be observed that the signal with higher line noise artifact seems to be the control recording #5, the one without stimulation, together with the stimulation recording #10 with intensity 60%.

Clearly, Figure 19 shows how the recordings with stimulation suffered changes that resulted in unexpected line noise artifact magnitudes.

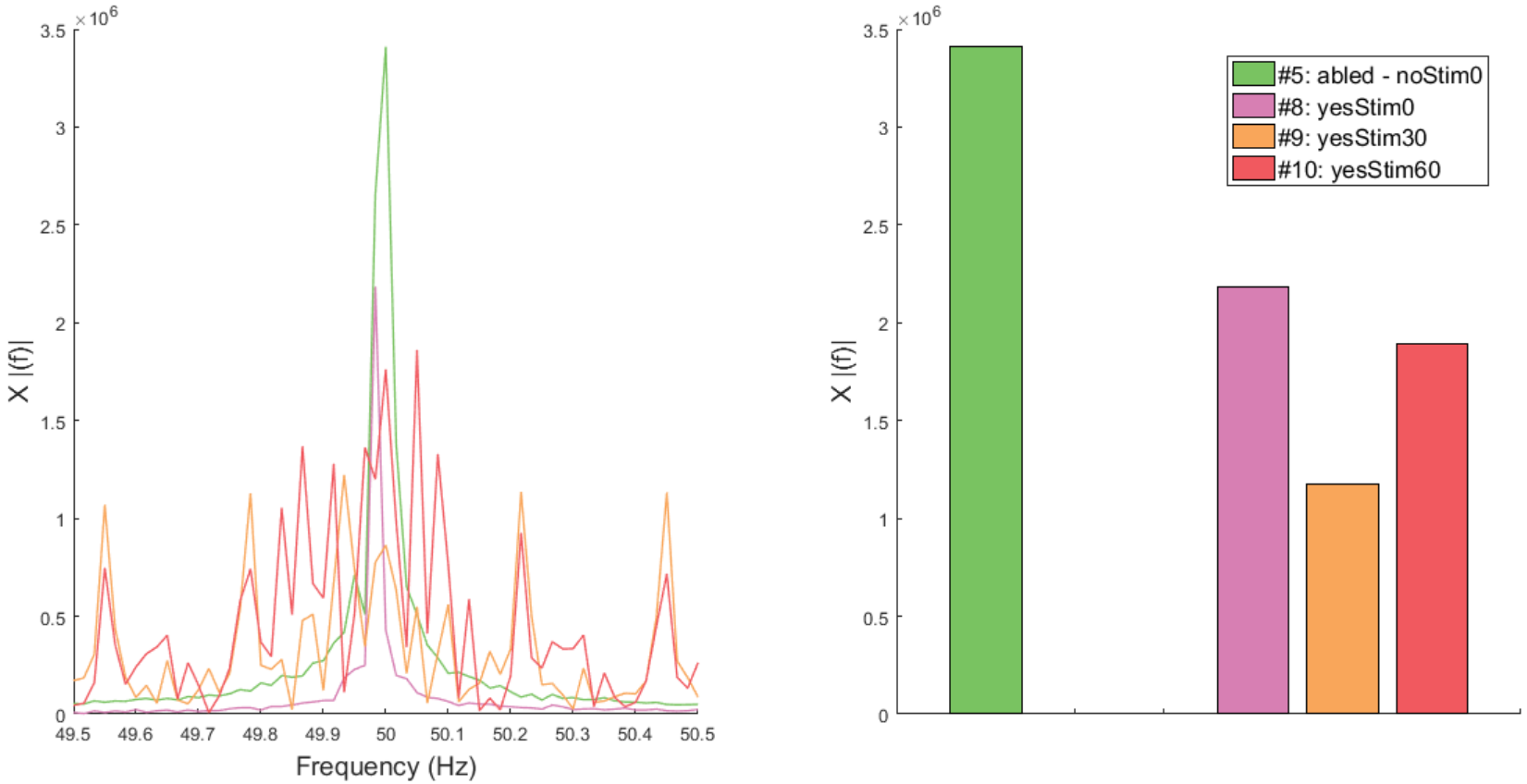

Figure 19: Figure 19: Spectrum (left) and bar plot the line noise peaks heights for the stimulation analysis.

Firstly, it can be observed that just by doing stimulation, even with intensity zero, the function of the device changed resulting in a lower line noise peak. Then, for the recordings #9 and #10 with stimulation and intensity 30% and 60%, the spectrum shows these unknown and symmetric peaks around the 50Hz frequency. This could be explained by some internal filtering that the TMS device may apply.

3.1.1.6 Experimental conclusions

This experiment analysis clarified the origin of the line noise; it proved that it mainly comes with a similar proportion from the antenna effect caused by the coil shape and from the current leaks consequence of powering on the TMS device (see Figure 20).

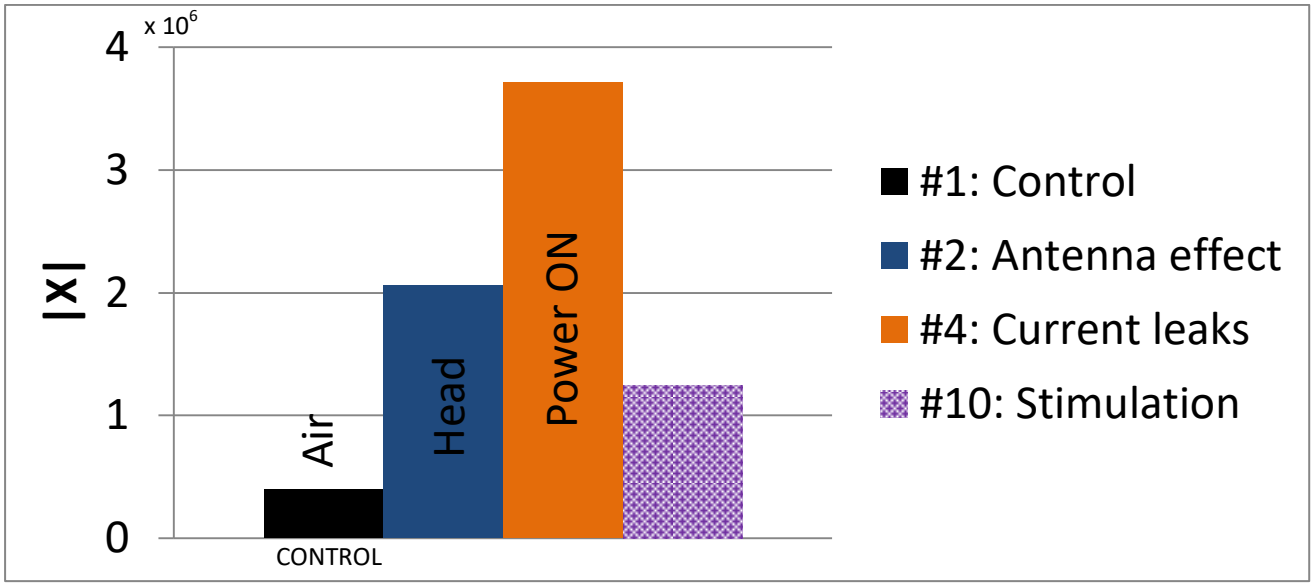

Figure 20: Line noise spectral peak heights of the most meaningful recordings.

3.1.2 Location

Once the origin of the artifact is clarified, it is also of interest to analyze how it is presented over the scalp, so if there are some locations that are more affected than others. This characteristic of the line noise artifact will be useful to proceed with the correction proposed algorithm

Hereunder, the scalp map is shown with the positions of the electrodes used in the measurements, using the well-known International 10–20 system. In our trials, 64 electrodes were used: 62 are the channels shown in the data; another is the reference electrode in the position FCz; and the left one is the ground located in AFz.

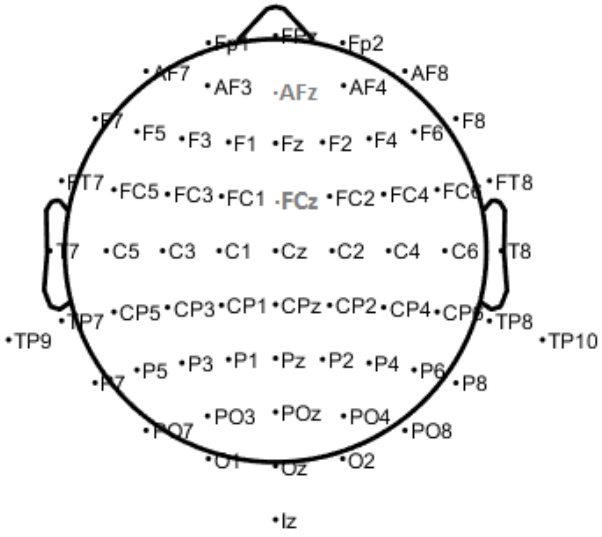

Figure 21. Scalp map showing the 64 electrode positions used in the measurements.

Five different measurements have been analyzed and for all of them the conclusion was the same. The next results correspond to the measurement with stimulation over the temporal muscle, between the electrodes C5-CP5, with 60% output power.

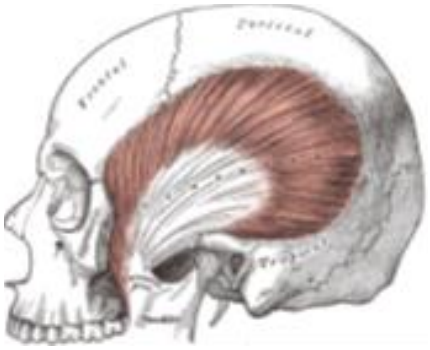

Figure 23: Anatomical representation of the temporal muscle, illustration from Gray (1918).

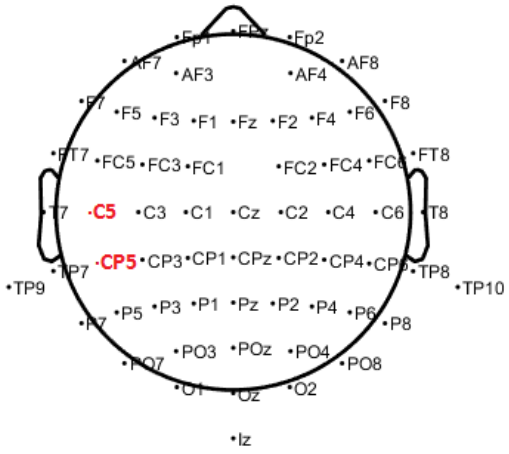

Figure 22: Stimulation target location.

In the next Figure 24, it can be observed the spectrum of every channel complemented with the scalp maps showing the locations at which some frequencies of interest are stronger. In our case, we will pay more attention on the last scalp map showing where the 50Hz frequency predominates, the one on the right.

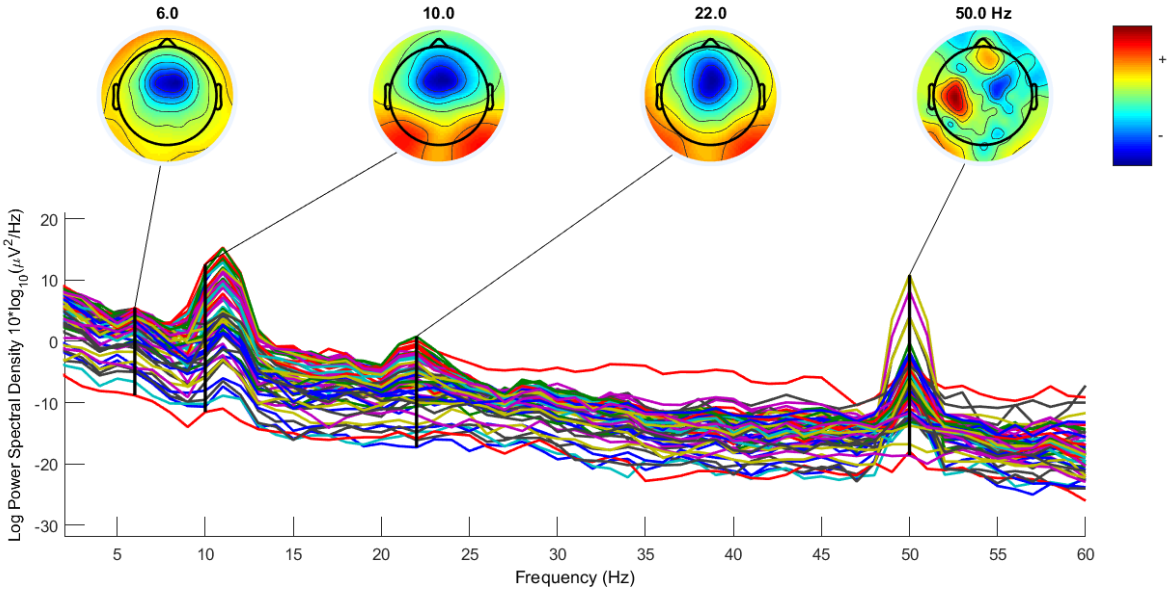

Figure 24: Spectrum with scalpmaps showing the location of specific frequencies.

After analyzing the five measurements, we can affirm that the 50Hz line noise mostly appears in the regions close to the stimulation target, near the space where the TMS device is working.

3.2 Correction

3.2.1 Existing methods

On one hand, there are pre-experimental methods which consist on assuring the low electrode impedances and a proper cable arrangement. However, these methods are not enough to prevent the line noise. That is why in this work the post-experimental methods are discussed. At this moment, there are a couple of toolboxes that introduce a method to correct the line noise artifact.

One of them is the TESA (Rogasch et al., 2016) which consist of an extension of the EEGLAB and adds several functions and proposes pipelines to process TMS-EEG data. Its method for correcting the line noise artifact is a first removal of the TMS pulse artifacts to, afterwards, filter the data with a Butterworth bandstop filter. The other one is the TMSEEG (Atluri et al., 2016), this is a toolbox supported also for Matlab but with its own GUI. This toolbox offers a pipeline which suggests to correct the line noise artifact by applying a Notch filter once the TMS pulses have been removed, so the same idea than TESA toolbox.

This correction method works well as it manages to remove the line noise artifact, however, it implies eliminating the segments of data where the pulses are found and losing the information that might be enclosed there. Furthermore, it eliminates the entire activity at 50Hz, including the one that it is not due to the line noise but could be from the neural activity.

3.2.2 Proposed method

After understanding the existing methods, an alternative solution which does not imply removing segments of data and applying frequency filtering is presented. The proposed algorithm is based on the combination of a subtraction filter using a reference extracephalic reference and an averaging of epochs.

3.2.2.1 Procedure

The best way to present this approach will be by explaining its procedure and getting into detail in each step. The Matlab code can be found in the Annex II and in "Files/LN_correction" as *algorithm_correction.m*.

3.2.2.1.1 Load data and preprocessing

The used data is TMS-EEG data with extracted epochs using EEGLAB. The example data can be found in "Files/LN_correction/Raw". In the worked data, there is a total of 20 epochs and each epoch is 4.5 seconds long. The pulse is located at the second 1 of every epoch. The followed steps for importing the data, extracting the epochs and saving the variable ALLEEG containing the data to be managed in Matlab Command window are indicated at the beginning of the script. As a preprocessing, the baseline of all channels is removed using the *detrend()* function.

3.2.2.1.2 Subtraction

Extracephalic reference analysis

To use the extracephalic reference for the subtraction, it is necessary to modify its parameters so that it fits each EEG channel. To get to know what parameters should be modified and in which way, a first view of the reference signal compared with an EEG channel is shown in Figure 25.

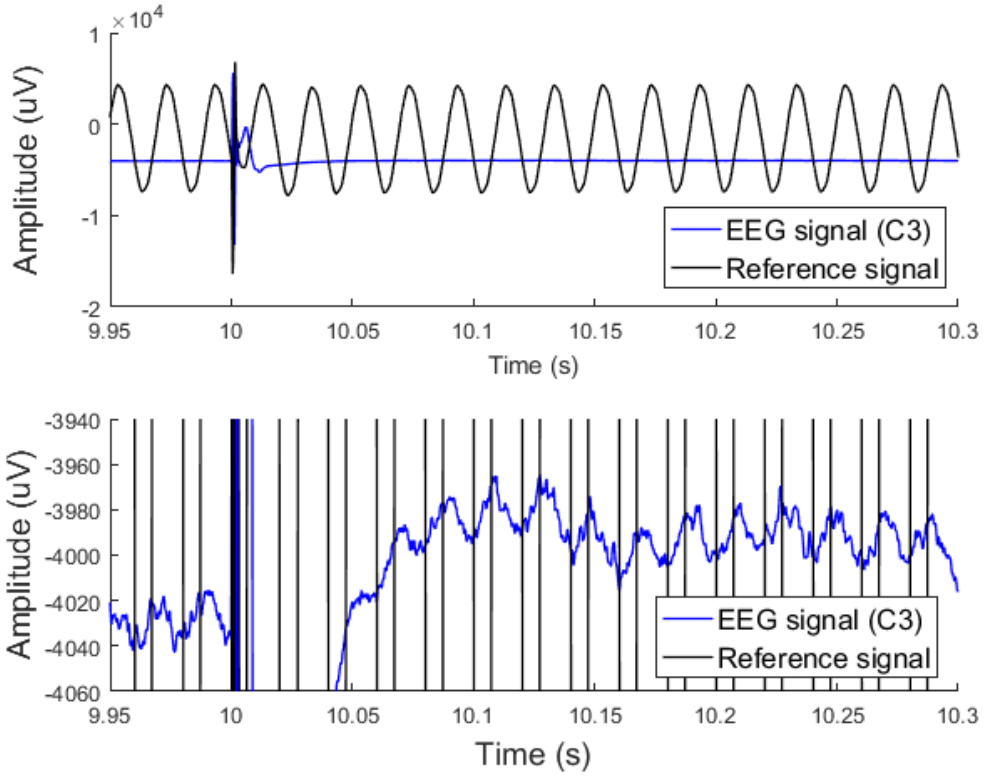

Figure 25: Temporal first view of EEG channel and the reference signal.

The amplitude of the reference signal is much higher than the one from the EEG and the signals are shifted in time. Before applying subtraction, it is needed to fit the reference signal to the EEG. So, an amplitude fitting and a phase shift correction must be done.

Amplitude fitting

In order to correct the amplitude of the reference electrode, a rate will be computed and, later, multiplied by the whole reference signal. This rate will be obtained by dividing the amplitude of the EEG channel and the reference channel. Then, there will be a different correction rate for each channel.

$$Rate = \frac{Amplitude_{EEG}}{Amplitude_{ref}}$$

$$Ref_{modified} = Rate \cdot Ref$$

Computing the amplitudes is done by dividing the signals in cycle units of 100 samples (20ms) and finding the maximum and minimum in each cycle. Then, their subtraction will result in the cycle amplitudes. Finally, as it is known that the line noise is constant in amplitude, all the computed amplitudes of the same signal are averaged and a final unique amplitude is found for each signal.

Define cycle units

The next Figure 26 shows how the signal is divided into the cycle units.

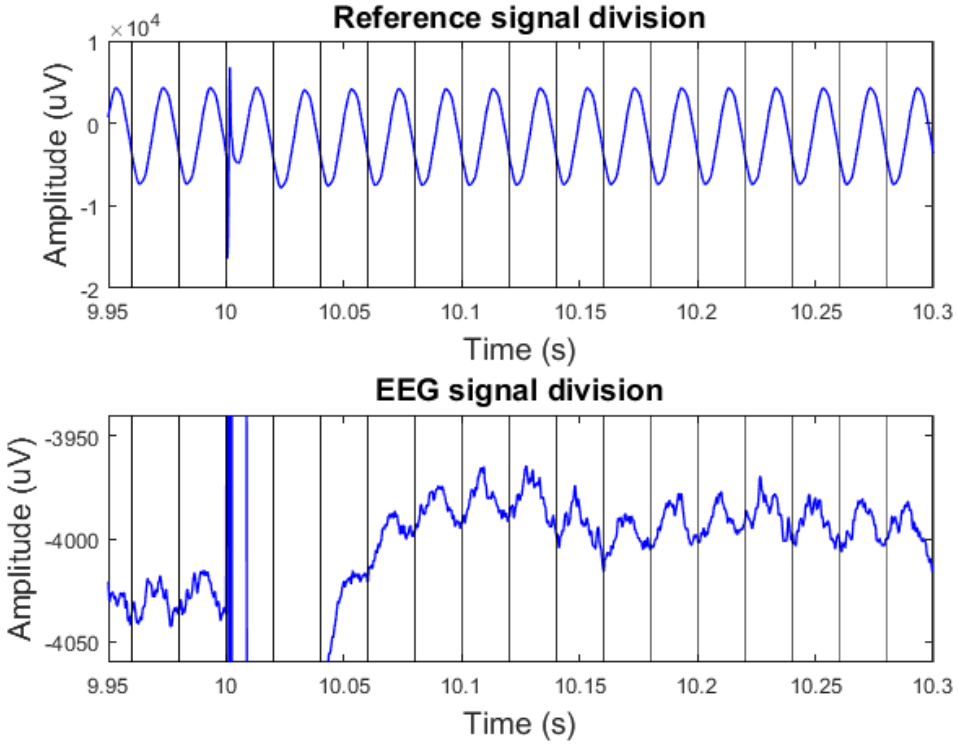

Figure 26: Division of the signals in cycle units.

This division of every channel into units is done using the function *reshape()*. There is a total of 62 channels, 3000 units per channel and 100 samples per unit.

Find maximums & minimums

For each unit, it is possible to find its maximum and minimum. This is done by using the function *max()* and *min()* in a loop that includes every unit of every channel.

Compute amplitude

The amplitude is computed by simply subtracting the minimums from the maximums. So, a different amplitude for every unit of every channel is obtained.

The next Figure 27 shows the amplitude of every unit from the channel C3. These amplitudes look correct as their values are in between 20 and 40uV. The dots that are far from the horizontal cloud correspond to the amplitudes on the pulse units, these have to be corrected later because if not the averaging of unit rates would be spoilt.

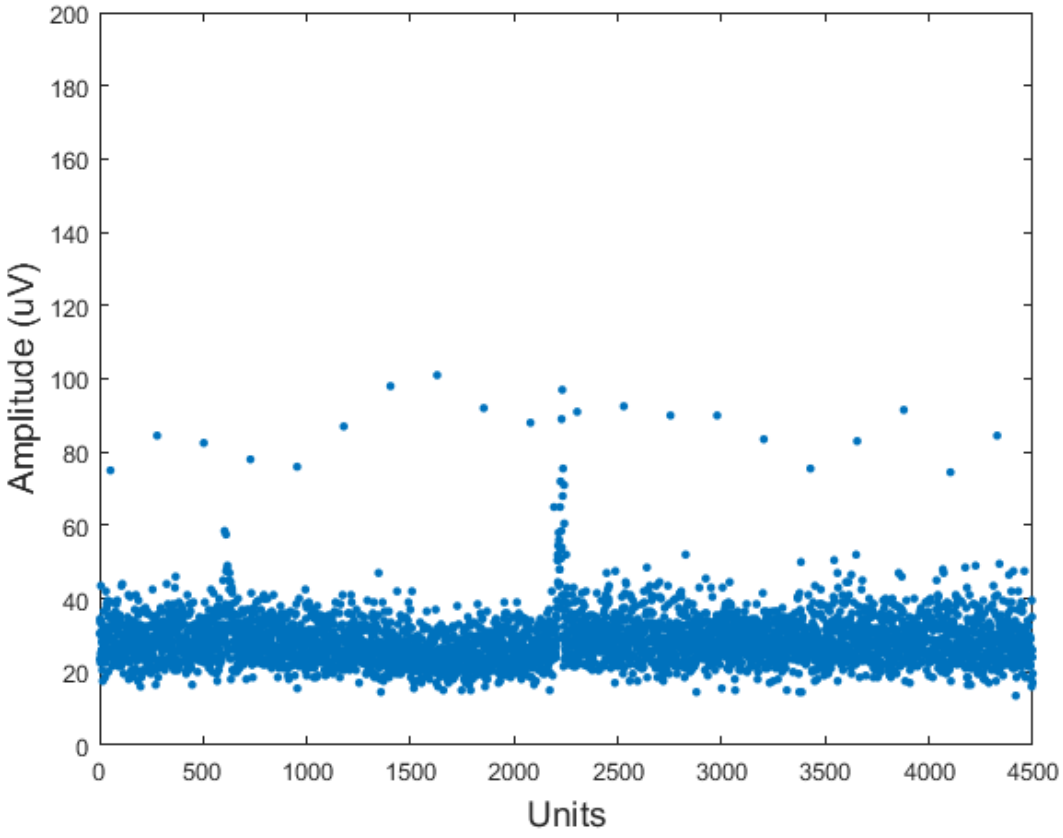

Figure 27: Computed amplitudes of the units from the channel C3.

Compute rate

In the next step, the rate that will be multiplied per the reference signal is computed. Next Figure 28 shows around what values is the rate found. In the case of C3, they are found around 0.002 and 0.004. Later, the average will be computed to know exactly what the rate for this specific channel is.

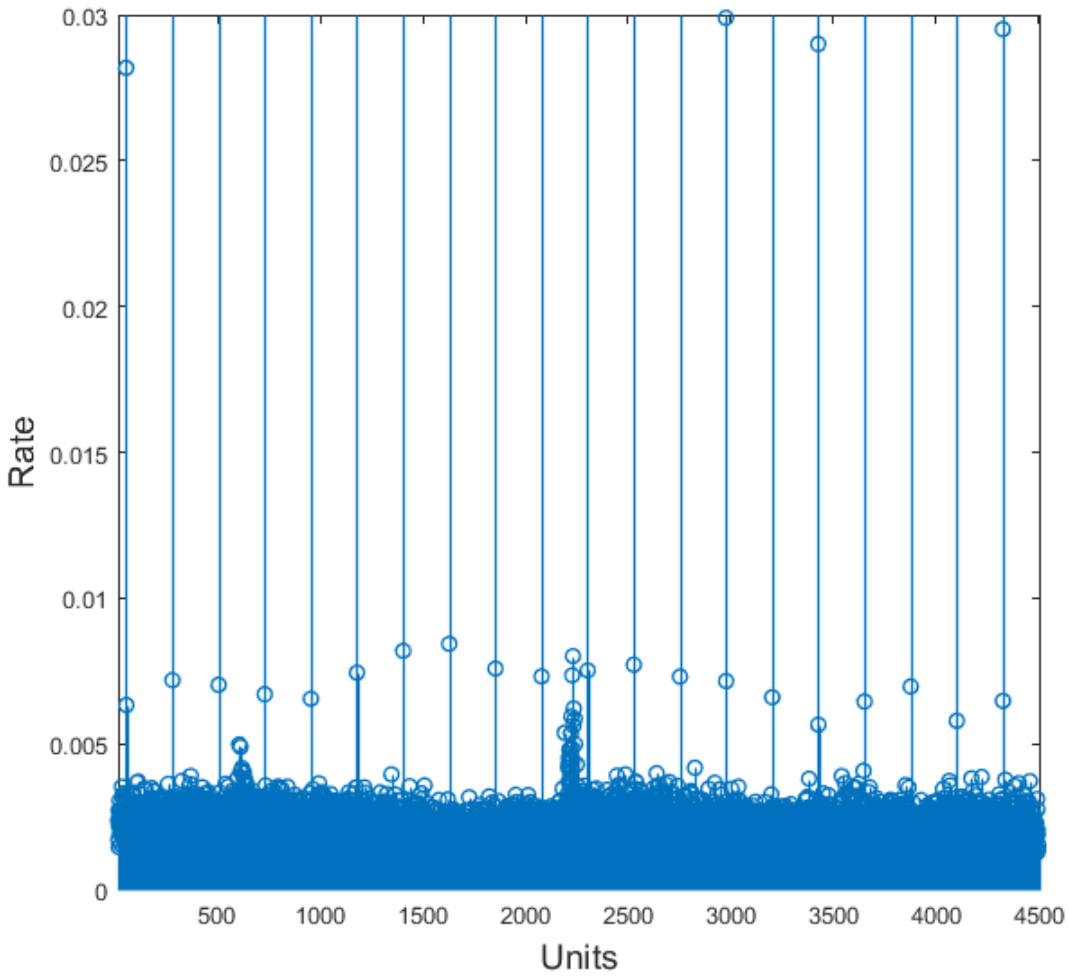

Figure 28: Computed rate values from the channel C3.

But as it can be observed, the units where the pulse was found were spoilt and now their rate is not correct. This is corrected by setting the three samples were the pulse is found to the average of the sample before and the fourth sample with respect to the pulse event, where the pulse decay has finished. The next Figure 29 shows the result of this correction.

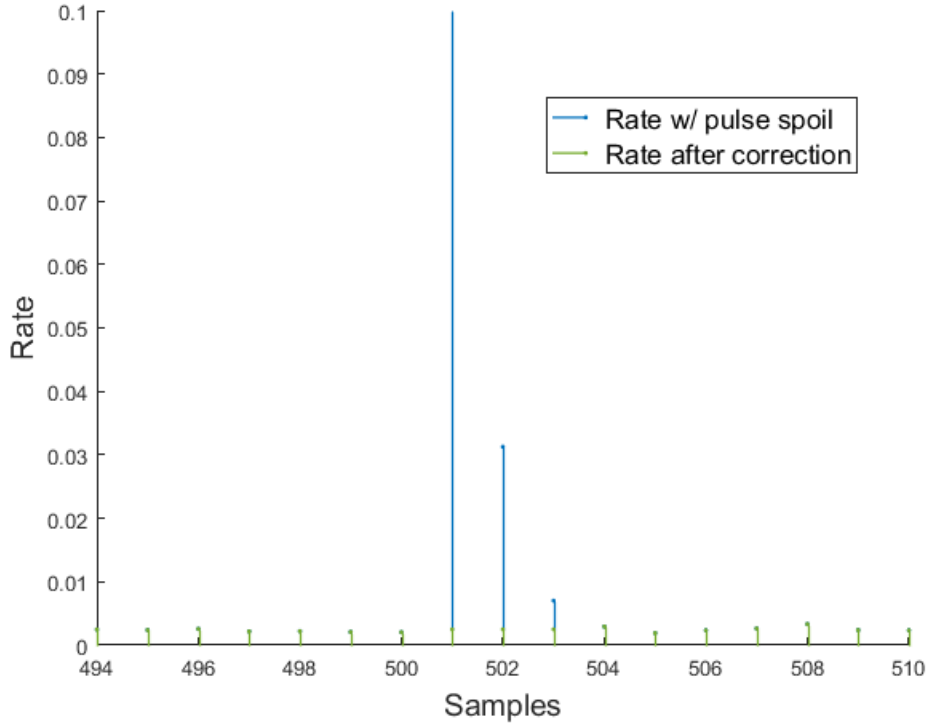

Figure 29: Pulse spoil correction in the rate values for C3.

Even though these were only three samples per pulse, it was important to correct them because they would spoil the computation of the average of rates.

To continue, the average is computed using the function *mean()*. The obtained values will be the ones multiplied by the reference signal to fit the reference signal amplitude. Hereunder, the rate values are shown, the bold rate 1.0000 corresponds to the rate of the same reference signal; it proves that the procedure is working well.

$$Rate = \frac{Amplitude_{EEG}}{Amplitude_{ref}}$$

```
avg_rate_amp                                                    =

    0.0014    0.0013    0.0036    0.0008    0.0040
    0.0013    0.0014    0.0010    0.0047    0.0014
    0.0012    0.0012    0.0021    0.0011    0.0025
    0.0010    0.0008    0.0013    0.0011    0.0012
    0.0023    0.0008    0.0009    0.0011    0.0013
    0.0011    0.0011    0.0007    0.0021    0.0012
    0.0013    1.0000    0.0010    0.0011    0.0013
    0.0012    0.0009    0.0010    0.0029    0.0010
    0.0013    0.0009    0.0010    0.0012    0.0012
    0.0013    0.0011    0.0011    0.0012    0.0012
    0.0028    0.0010    0.0010    0.0010
    0.0014    0.0032    0.0009    0.0020
    0.0032    0.0013    0.0025    0.0013
```

Compute modified reference signal

As explained, the rate is different for each channel. So now, it is time to compute the modified reference signal for each single channel though the shown expression.

$$Ref_{modified} = Rate \cdot Ref$$

As the reference electrode also recorded the TMS pulses, they are removed from the modified reference signals using *removePulses()* from -10 to 20 milliseconds with respect to the trigger pulse event, so that when applying the subtraction the genuine pulses of each recording is not changed. Figure 30 shows what part of the signal is zero-padded so that it does not affect to the subtraction.

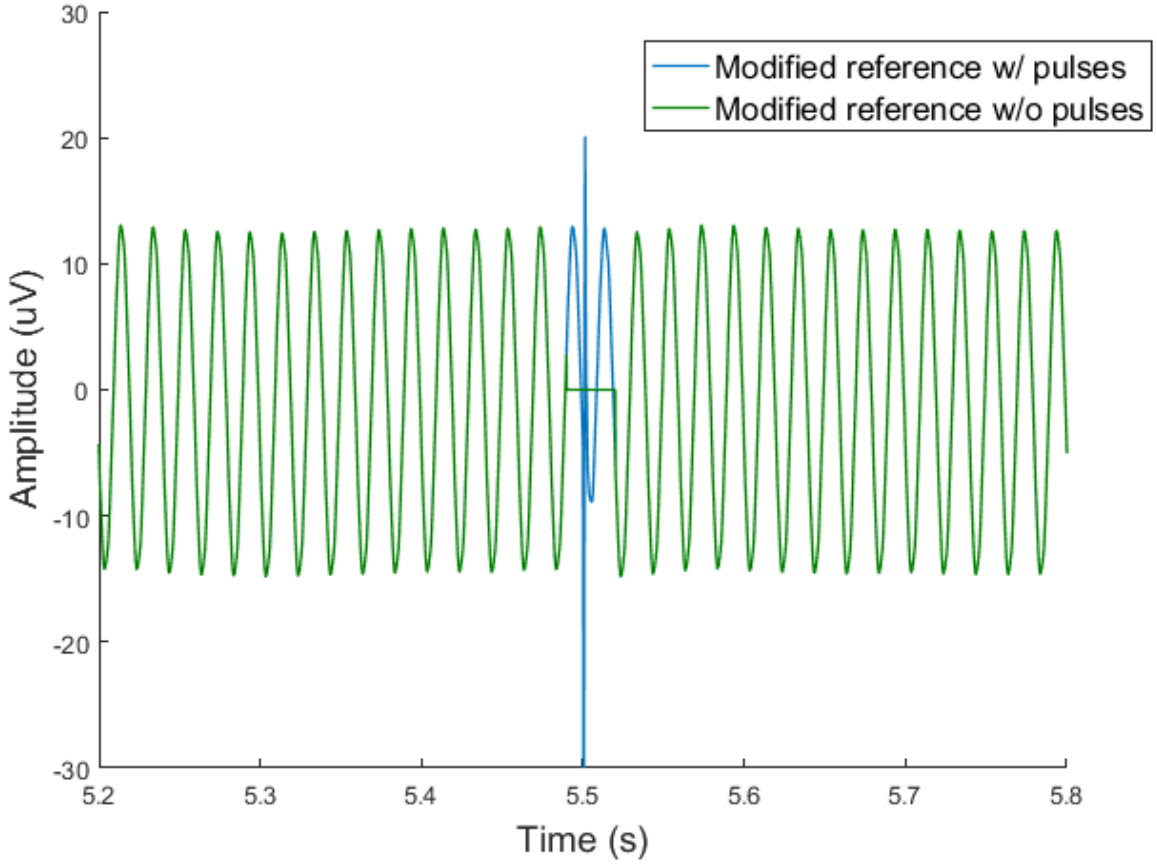

Figure 30: Pulses removal from the modified reference signals.

Finally, it is checked that the amplitude was properly modified by plotting together an EEG channel and its modified reference channel. In this Figure 31, it is clear how this is managed.

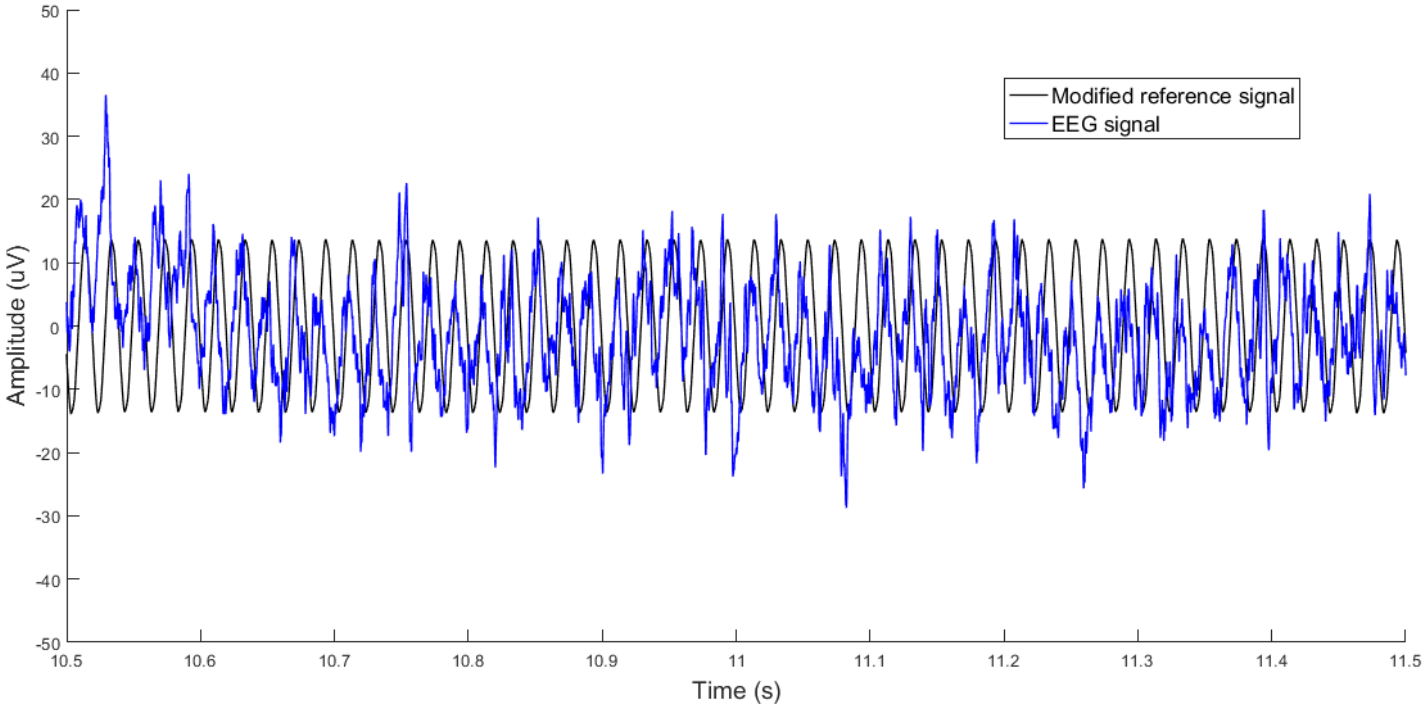

Figure 31: Fitted amplitude of the reference signal for channel C3.

It is possible to see that the next step will consist of correcting the phase shift as the 50Hz oscillations of the two signals do not coincide.

Phase shift correction

Once the amplitude of the reference signal was fitted to each EEG channel, the lag between the modified reference signals and the EEG signals is corrected following the next procedure.

First of all, segments without pulse will be extracted from every signal. Later, they will be cross-correlated in order to find the exact number of samples that the segments are lagged between them. It is assumed that the lag found for the segment will be the lag of the entire signal; later, this is checked. Finally, the lag will be corrected by removing an amount of samples equal to the lag, so that one of the signals is displaced and gets aligned with the other.

Extract segments

A segment lasting 1 second, from the second 7 to the 8, is selected. Any other segment without pulse would also work. The next Figure 32 shows the extracted segment from an EEG channel and from the reference channel.

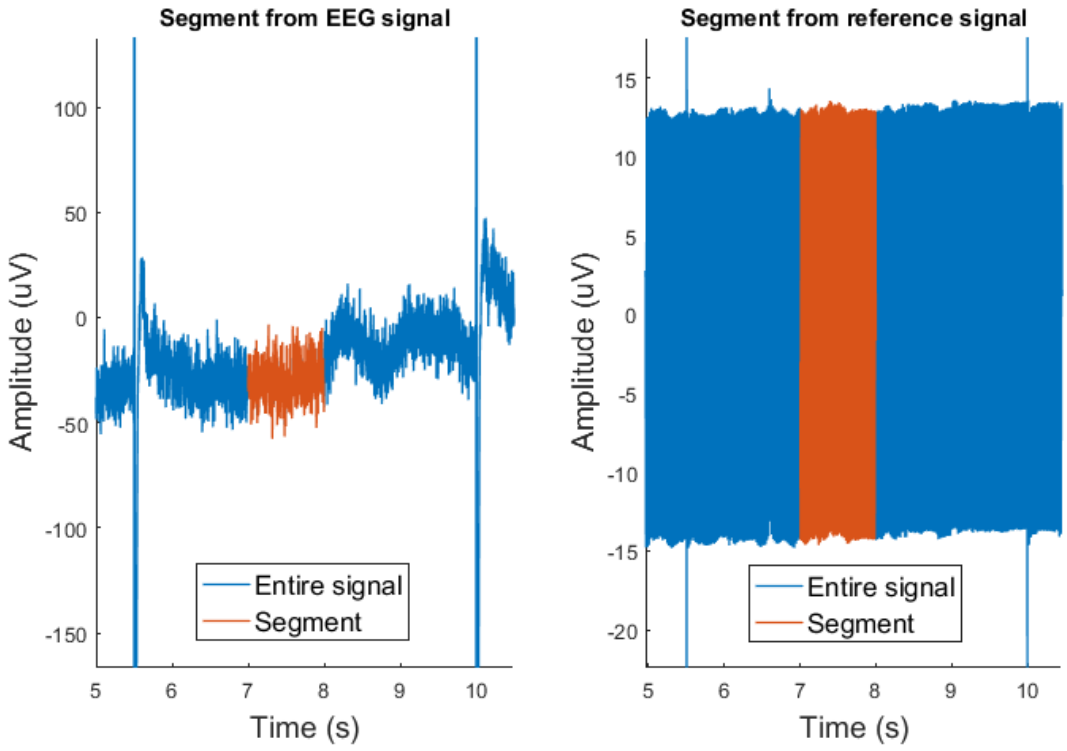

Figure 32: Extracted segment from the signals, EEG (left) and reference (right).

Find the lag

Next, the cross-correlation is applied and the lag is computed by finding the location of the maximum peak in the cross-correlation. From the next Figure 33, the location of the maximum will be equal to the number of samples that the EEG signal is shifted from the reference signal.

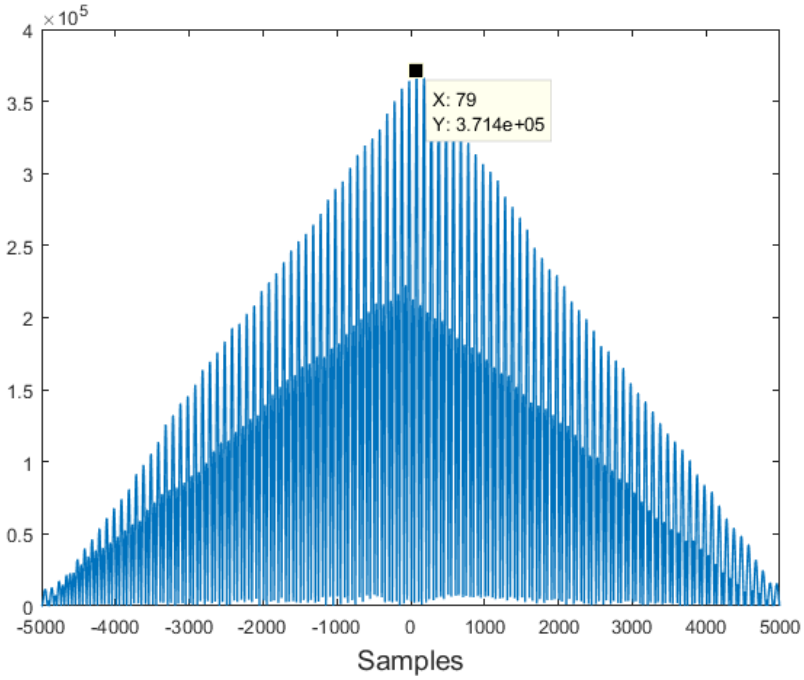

Figure 33: Cross-correlation of C3 and its corresponding reference signal.

As the 50Hz wave is having a periodicity of 100 samples, we know that the lag range will be from -100 to +100. The next Figure 34 shows a zoom of the correlation for the range in which the peak is expected to be found.

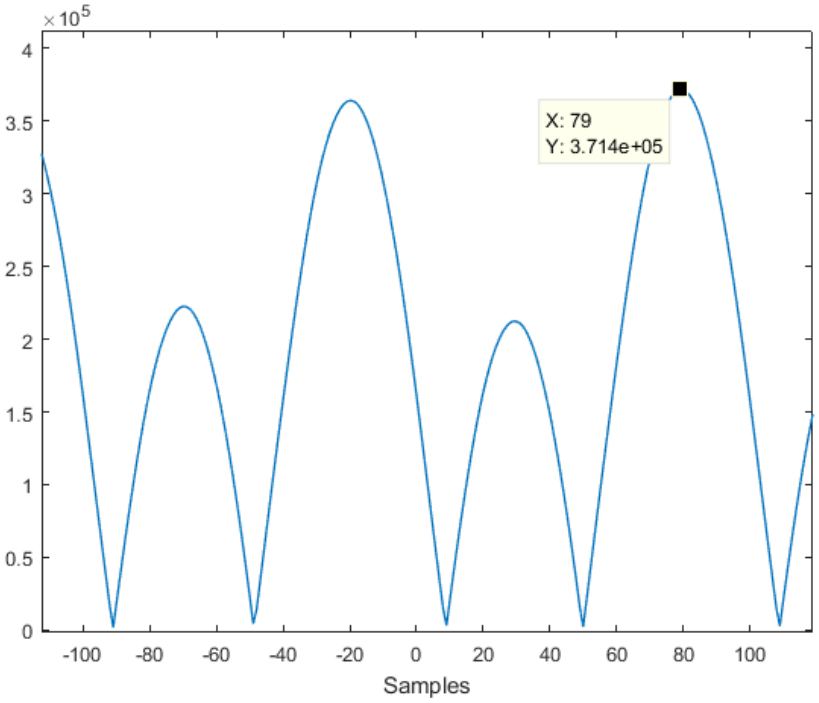

Figure 34: Zoomed cross-correlation of C3 and its corresponding reference signal.

The next variable shows the found lag for each channel:

```
lagDiff =

    76    32    30    33   336
    29    28    32    40    81
   178    30    32   -20    29
   -20    78    30   562    30
    79    79   178    29    26
    30    79    80    79    29
    78     0    76    32    30
   -20    16    29    79    78
    80    29   -22    30   -19
   -20    78    29    74    78
    30    79    77    31
    30    78    27    30
    26    31   128    32
```

Values look quite reasonable because most of them are below the maximum expected lag of 100 samples, because the 50Hz cycle is 100 samples long. However, there are some in which the delay was wrongly found because the signals may be too different and the line noise too small in the EEG signal.

Correcting the lag

Next, the signals are aligned taking into account the found lag. Some of them are expected to be wrong aligned as not all the found delays are correct. It is assumed that from now on the pipeline will only work for some channels.

After applying the procedure, it is possible to check if the alignment worked well. The first channel to be checked is C3 as it is the one with more line noise artifact. It will be checked for the region where the segment was extracted and also for another region (see Figure 35); this way, it will be proved that the whole signal has been aligned and not only the segment. Then, the subtraction will make sense.

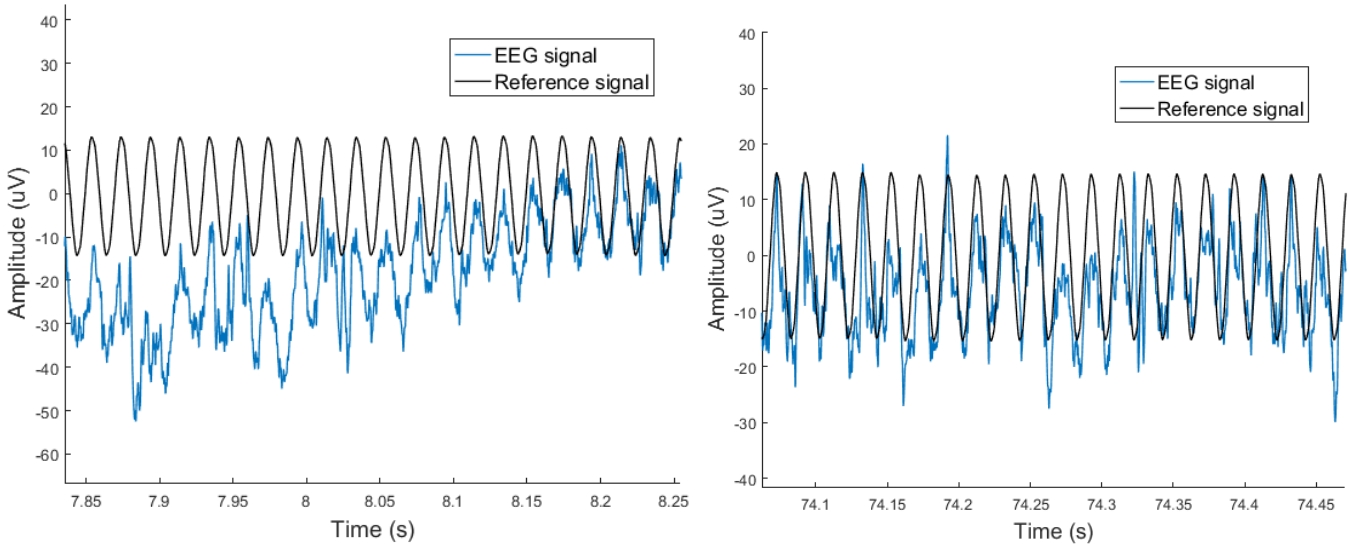

Figure 35: Aligned signals for the selected segments (left) and for a further time window (right), for channel C3.

Both last figures show aligned signals. This means that the subtraction is expected to be successful for channel C3. Next, let us check if the alignment also worked for another channel. Hereunder, Figure 36 shows a channel far from the TMS stimulation region.

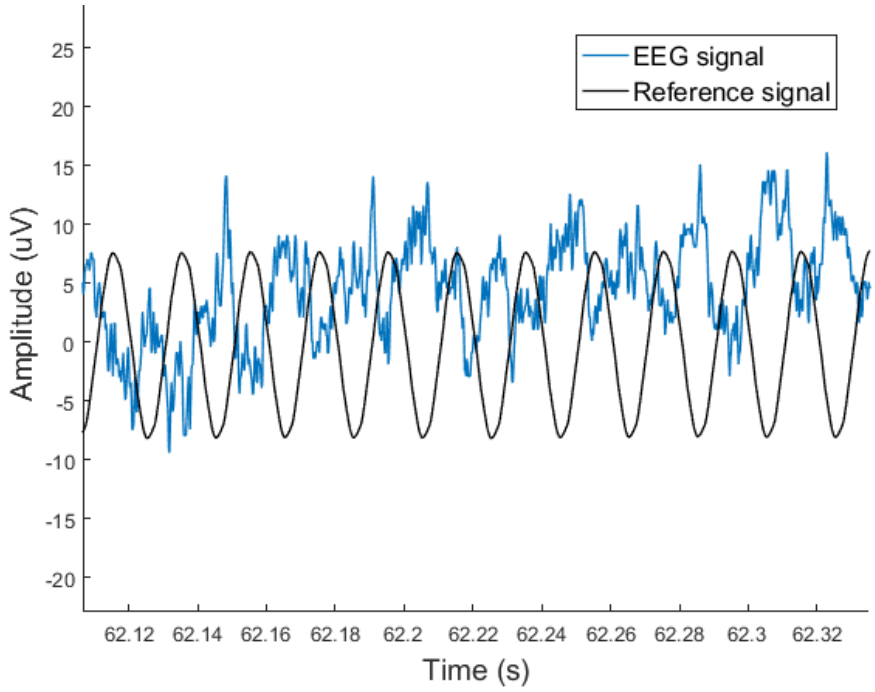

Figure 36: Signal alignment of C6, a channel far from C3.

It can be seen that the line noise wave in the EEG channel is so low that it cannot be observed in the time domain. Hence, it does not make much sense to apply the method of subtraction because the reference signal was not fitted in amplitude properly and corrected in shifting without any temporal presence of the 50Hz in the EEG signal.

For the current algorithm, the subtraction is only ready to be applied to the channels with high line noise artifact, these are the ones close to the target stimulation location.

Subtraction operation

Finally, the subtraction operation can be done for the channels in which the extracephalic reference was managed to be fitted in amplitude and phase. It is just about taking the EEG channels and subtracting their respective modified reference signal from them.

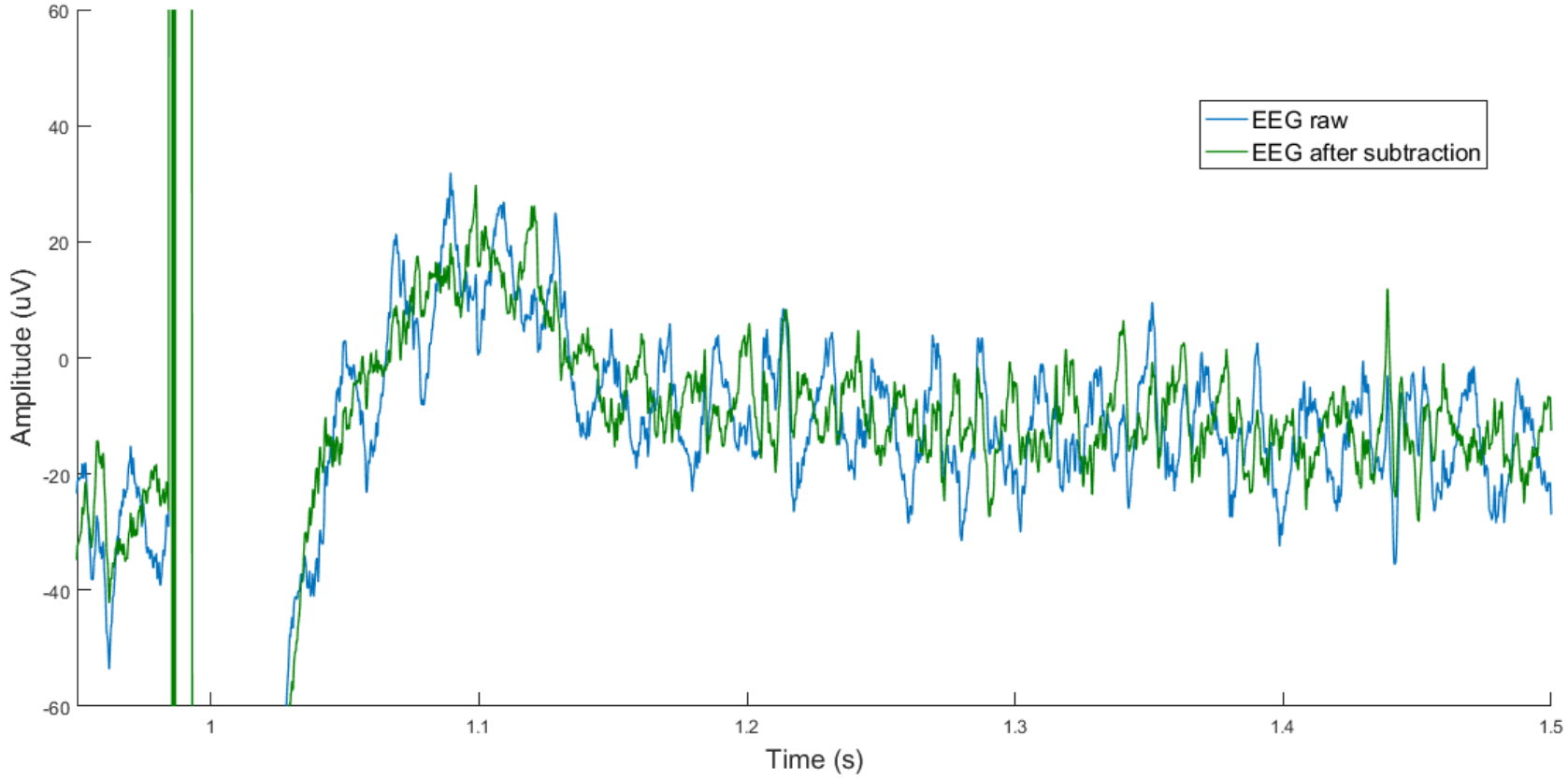

Figure 37: Temporal comparison of the signal before and after subtraction.

As it can be observed in Figure 37, the data after subtraction is showing less 50Hz wave. Later, this will be checked in the spectrum.

3.2.2.1.3 Averaging

Once the subtraction is done, this operation will consist of taking all epochs for a single channel and averaging them. It will be done for all channels so that, at the end, we obtain a variable containing the epoch average for every channel.

Firstly, it is needed to separate again the signal in epochs. Then, the averaging formula can be applied.

$$AVG\ Epochs = \frac{\sum Epochs}{N_{epochs}}$$

Figure 38 shows how different do the averaged epochs and a single epoch looks like regarding the line noise artifact.

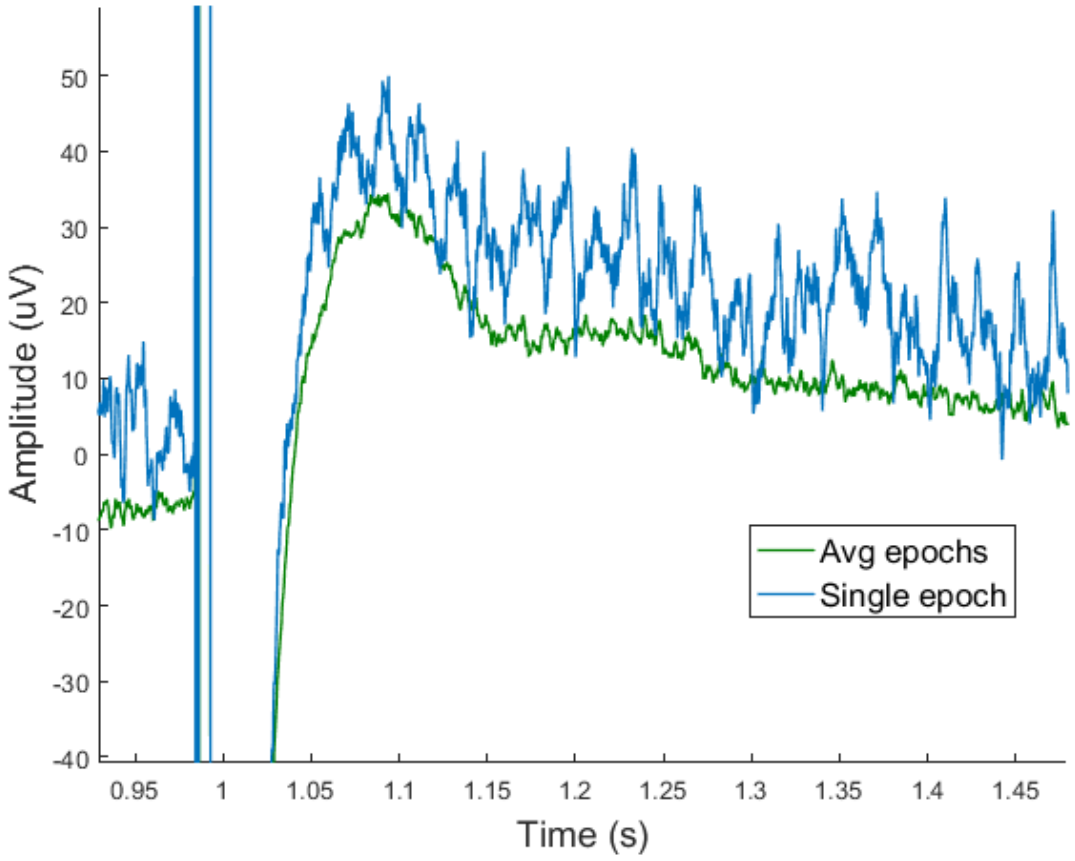

Figure 38: Averaged epochs result for C3.

In the temporal domain, it is already possible to see how the signal has improved a lot. As there were 20 epochs, using the known formula $1/\sqrt{N}$ we can confirm that the line noise has been reduced by the factor 0.2236 from the signal after subtraction.

Once we have obtained this result, it is time to analyze it in the frequency domain to quantify how much the line noise has been reduced and confirm if the subtraction is really improving the correction.

3.2.2.2 Results analysis

In order to be able to perform the spectral analysis, it is needed to remove the TMS pulses with zero-padding so that they do not spoil the spectrum computation. Pulses from four different cases are removed to do a further proper comparison. The four cases are:

1. Single epoch without subtraction
2. Single epoch with subtraction
3. Averaged epochs without subtraction
4. Averaged epochs with subtraction

The zero-padding is decided to go from 30 milliseconds before the TMS event until 300 milliseconds after the event, so that the whole pulse is removed. Next Figure 39 shows how the pulse has been removed.

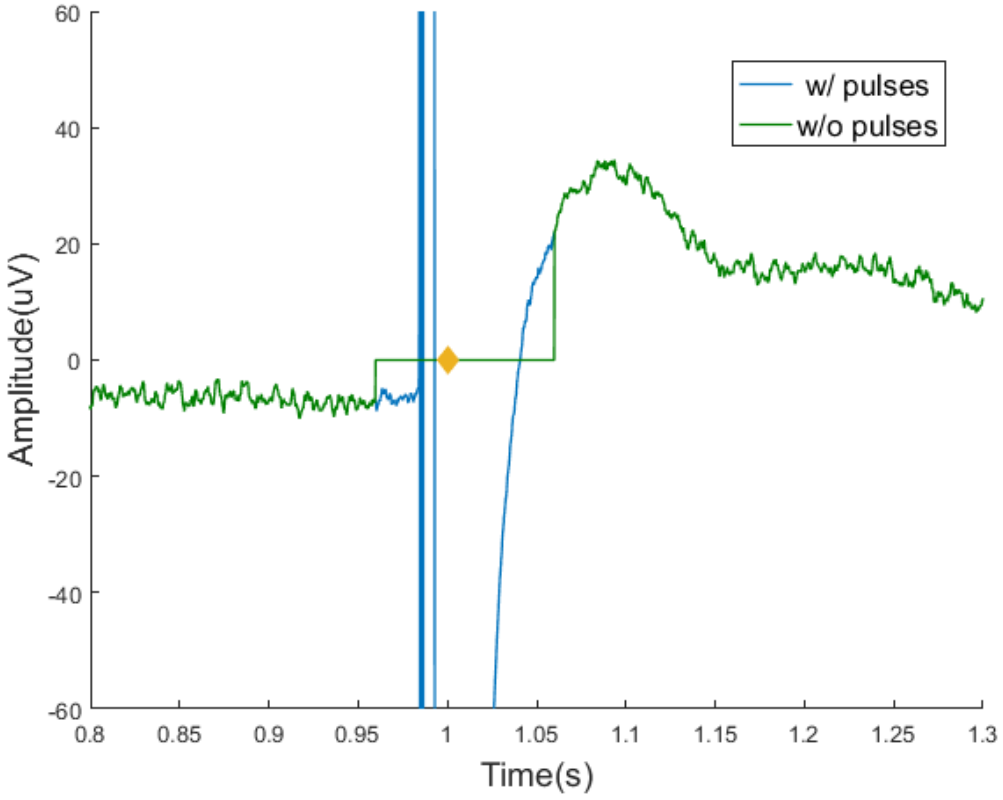

Figure 39: Pulse removal from the signal of averaged epochs with subtraction.

Then, the Fourier transform can be computed using *fft2plot()*. Next, Figure 40 and 41 show the resulting spectrum.

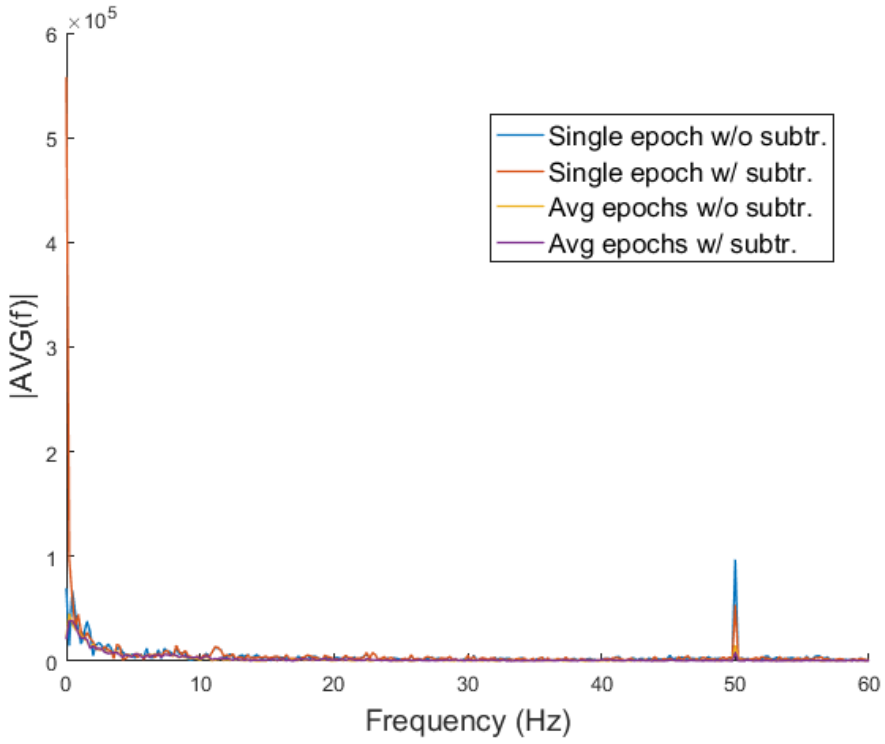

Figure 40: Spectrums of the studied cases.

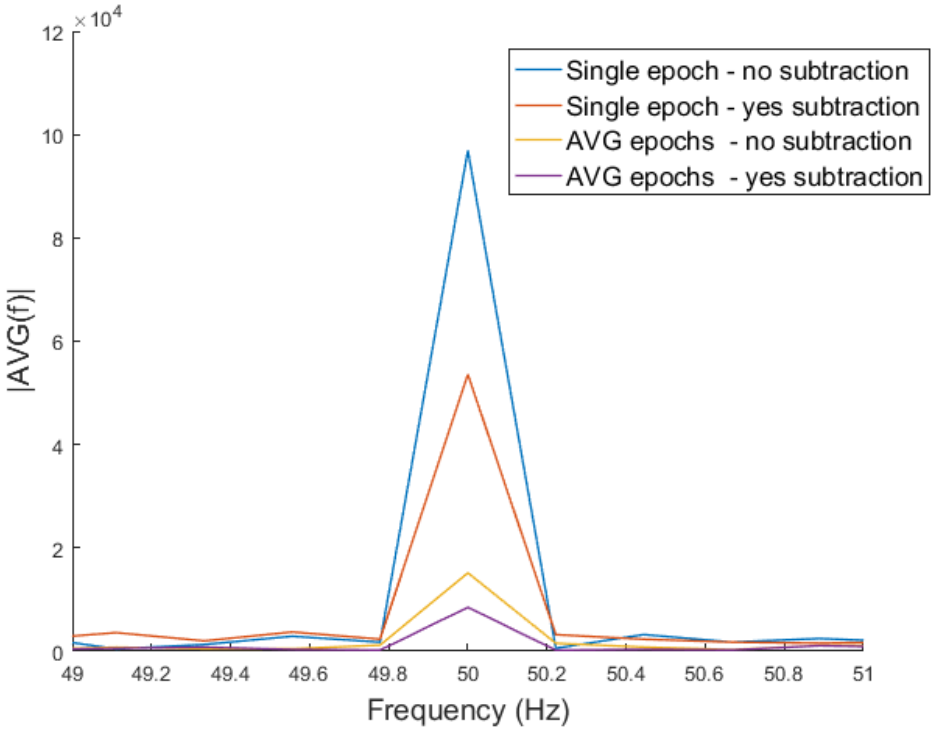

Figure 41: Closer view to the line noise peaks of the studied spectrums.

The last Figure 41 confirms how the proposed algorithm worked positively on the removal of the line noise artifact as it is the one having the smallest peak at 50Hz. The peak is getting close to the height of the surrounding frequencies which are assumed to have a correct magnitude.

Here, there are shown the 4 previous cases in the temporal domain. It is possible to observe how averaging truly reduces the line noise. However, the difference between the two averaged signals, with and without subtraction, is not possible to be observed and this is why we needed the spectrum analysis, which proved that the averaged signal after subtraction is giving the best result.

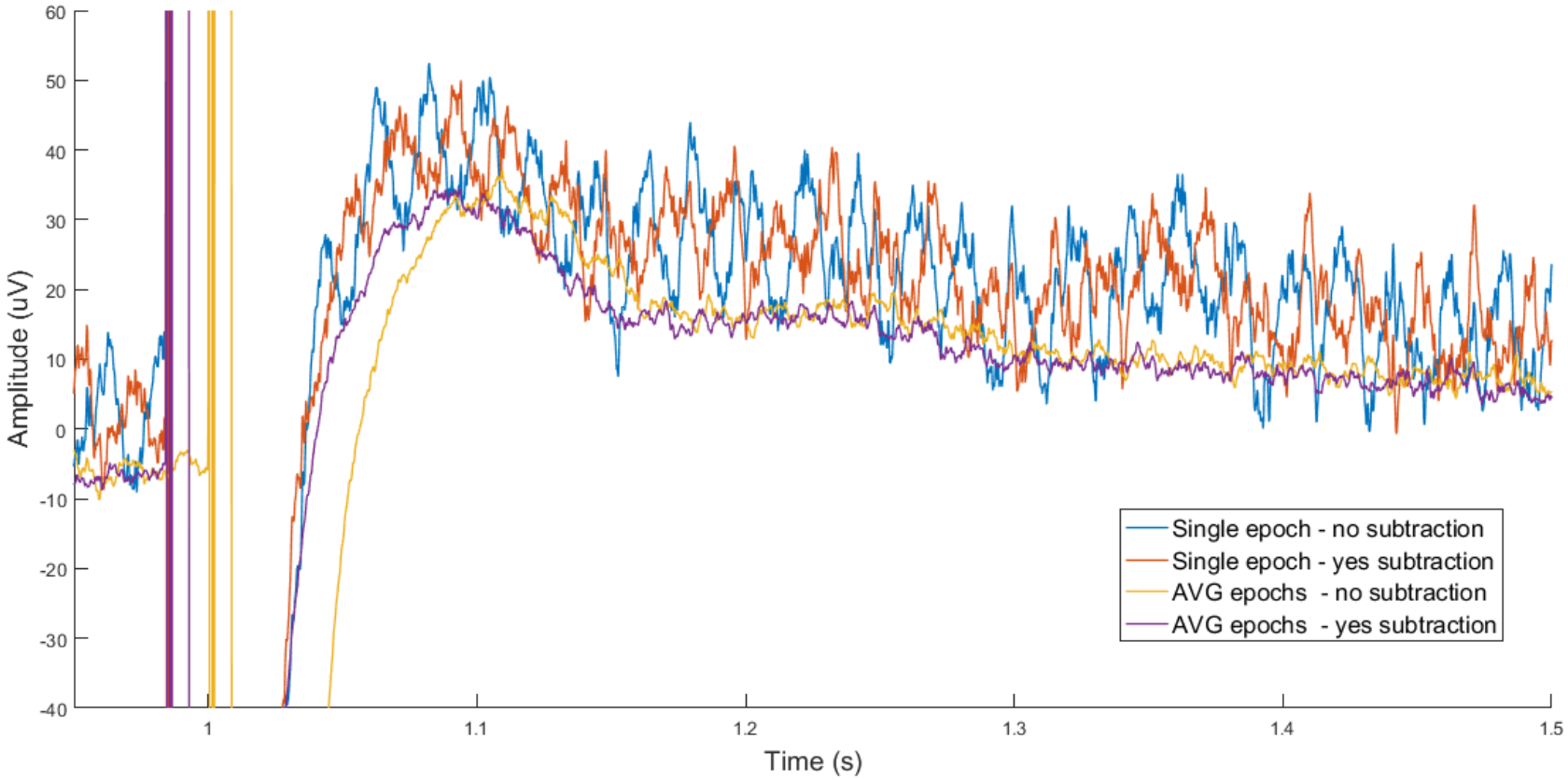

Figure 42: Results in the temporal domain of the studied cases.

4. Conclusions

In the combination of TMS with EEG, the line noise artifact appears to be a problem as it overlaps the neural activity complicating the analysis of TMS-EEG data.

In this study, the main origin of this interference was found to be the antenna effect caused by the TMS-coil shape and the current leaks coming from the TMS device when powered on. Both sources seemed to affect the line noise artifact magnitude in the same proportion. Once known the characteristics of the problem, the existing methods were understood to, later, propose an alternative pipeline which could improve the weak points of the existing algorithms.

The presented method applied the combination of subtraction with an extracephalic reference and averaging, showing a positive result in which the line noise has been managed to be minimized. As an advantage to the existing methods, it does not remove the segments of the signal containing the pulses, so the information contained there is not lost. Furthermore, it does not completely remove the entire frequency component at 50Hz, because it only acts subtracting the line noise wave. Oppositely, this method can only be applied to the channels where there is high line noise, so the ones where the TMS stimulation is aimed. This might be solved with some further improvement on the procedure of fitting the amplitude based on a regression approach. As another drawback, there is an obvious need of recording this special extracephalic electrode which is not present in the current standardized EEG procedure. To avoid this electrode need, an idea could be to model the reference signal from the EEG electrodes, but this is already a different approach that the one shown in this work.

With this work, a new approach for the correction of the line noise artifact in TMS-EEG data has been described. I believe that this will be useful for either improving the existing methods or building new pipelines based on the presented ideas.

5. Bibliography

[Proakis & Manolakis 2017] PROAKIS, JOHN G.; MANOLAKIS G., DIMITRIS. *Tratamiento digital de señales 4 Edición.* Pearson Educación S.A., 2007

[Delorme & Makeig 2004] DELORME, ARNAUD; MAKEIG, SCOTT: *EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis.* In: Journal of Neuroscience Methods, 2004

[Rogasch et al. 2016] ROGASCH, NIGEL C.; SULLIVAN, CALEY; THOMSON, RICHARD H.; ROSE, NATHAN S.; BAILEY, NEIL W.; FITZGERALD, PAUL B.; FAR- ZAN, FARANAK; HERNANDEZ-PAVON, JULIO C.: *Analysing concurrent transcranial magnetic stimulation and electroencephalographic data: a review and introduction to the open-source TESA software*. In: NeuroImage, 2016

[Alturi et al. 2016] ATLURI, SRAVYA; FREHLICH, MATTHEW; MEI, YE; DOMINGUEZ, LUIS G.; ROGASCH, NIGEL C.; WONG, WILLY; DASKALAKIS, ZA_RIS J.; FARZAN, FARANAK: *TMSEEG: A MATLAB-Based Graphical User Interface for Processing Electrophysiological Signals during Transcranial Magnetic Stimulation.* In: Frontiers in Neural Circuits 10, April 2017

[Weißbach & Welfonder 2019] WEIßBACH, TOBIAS; WELFONDER, ERNST: *High Frequency Deviations within the European Power System - Origins and Proposals for Improvement.* In: VGB PowerTech 6/2009, June 2019

[Swissgrid AG] Swissgrid AG: Netzfrequenz im Fachportal. Internet. https://www.swissgrid.ch/swissgrid/de/home/experts/topics/frequency.html [March 2017]

[Fuchs 2017] FUCHS, T.: *TMS-evozierte Signale : Artefakterkennung und Korrektur.*In: OTH, 2017

[Delorme] DELORME, ARNAUD: *ICA (Independent Component Analysis) for dummies.* https://sccn.ucsd.edu/~arno/indexica.html [April 2017]

[Gray 1918] GRAY, HENRY: *Anatomy of the human body*. Philadelphia and New York: Lea and Febiger, 1918

[Ramón y Cajal 1899] RAMÓN Y CAJAL, SANTIAGO: *Drawing of Purkinje cells (A) and granule cells (B) from pigeon cerebellum*. Instituto Santiago Ramón y Cajal, 1899

[Wassermann 2013] WASSERMANN, ERIC: *Transcranial Brain Stimulation representation*. National Institute of Neurological Disorders and Stroke, National Institutes of Health, United States Department of Health and Human Services, 2013

[Schomer et al. 2010] SCHOMER, D.; SILVA, F.; SILVA, F.: *Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Williams & Wilkins, 2010

[Ilmoniemi & Kicic 2010] ILMONIEMI, RISTO J.; KICIC, DUBRAVKO: *Methodology for Combined TMS and EEG*. In: Brain Topography 22, 2010

[Doren et al. 2015] DOREN, J. V.; LANGGUTH, B.; SCHECKLMANN, M.: *TMS-related potentials and artifacts in combined TMS-EEG measurements: Comparison of three different TMS devices.* In: Neurophysiologie Clinique/Clinical Neurophysiology 45, 2015

[ter Braack et al. 2015] TER BRAACK E.M.; DE VOS, C.C.; VAN PUTTEN. M.J.: *Masking the Auditory Evoked Potential in TMS-EEG: A Comparison of Various Methods.* In: Brain Topography 28, 2015

[Dornhoffer & Mennemeier 2010] DORNHOFFER, JOHN L.; MENNEMEIER, MARK: *Using repetitive transcranial magnetic stimulation for the treatment of tinnitus*. In: The Hearing Journal 63, 2010

[Perera et al. 2016] PERERA, TARIQUE; GEORGE, MARK S.; GRAMMER, GEO_REY; JA- NICAK, PHILIP G.; PASCUAL-LEONE, ALVARO; WIRECKI, THEODORE S*.: The Clinical TMS Society Consensus Review and Treatment Recommendations for TMS Therapy for Major Depressive Disorder*. In: Brain Stimulation 9, 2016

[Simpson & Macdonell 2015] SIMPSON, MARION; MACDONELL, RICHARD: The use of transcranial magnetic stimulation in diagnosis, prognostication and treatment evaluation in multiple sclerosis. In: Multiple Sclerosis and Related Disorders 4, 2015

ANNEX I: Lists

List of figures

List of tables

ANNEX II: Code

Line noise analysis

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This script aims to analyse the experimental recordings to study
% what are the origens of the line noise artifact. For every recording, the
% 50Hz component will be analysed in both temporal and frequency domain.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Load data

```
% The raw recordings found on the Files/LN_correction/Raw folder are
% imported to EEGLAB and epochs from the recordings with stimulation are
% extracted. It is done the next way:
%
%   1. Start EEGLAB         >>eeglab
%   2. Import recordings    File>Import data>Using EEGLAB functions and
%                           plugins>From Brain Vis. Rec. .vhdr file
%
% Once the all recordings are imported in EEGLAB, epochs are extracted
% from the recordings with stimulation.
%
%   3. Extract epochs         Tools>Extract epochs
%                           In the pop-up, select the triggers where the
%                           pulses are located as the reference event.
%                           Then, a range from 1 second before the event
%                           until 3.5 seconds after the event.
%
% Finally, the variable ALLEEG containing all the datasets is saved
%
%   4. Save ALLEEG variable >>save('ALLEEG','-v7.3');
%
% This is the variable containing all the data used in this script.

load ('data_analysis.mat');

fs=5000; % Sampling rate

% A length is selected to analyse the same data amount for each recording
N=300000; % The selected length is 300000 samples (1 minute)
t=0:1/fs:(N-1)*(1/fs); % Time vector for a later plotting of the data

% Save every recording in an independent variable
ch=1:62;
air_unplug_powerOff_noStim=ALLEEG(1).data(ch,1:N);
head_unplug_powerOff_noStim=ALLEEG(2).data(ch,1:N);
head_plug_powerOff_noStim=ALLEEG(3).data(ch,1:N);
head_plug_powerOn_disabled=ALLEEG(4).data(ch,1:N);
head_plug_powerOn_noStim0=ALLEEG(5).data(ch,1:N);
head_plug_powerOn_noStim30=ALLEEG(6).data(ch,1:N);
head_plug_powerOn_noStim60=ALLEEG(7).data(ch,1:N);
head_plug_powerOn_yesStim0=ALLEEG(8).data(ch,1:N);
```

```
% Recording 9 and 10 is data with stimulation but not epoched,
% it is not of interest to be used.
head_plug_powerOn_yesStim30_epochs=ALLEEG(11).data(ch,1:N);
head_plug_powerOn_yesStim60_epochs=ALLEEG(12).data(ch,1:N);

%Group all the data in a sigle variable for easing its manipulation
data=cat(3,air_unplug_powerOff_noStim,...
            head_unplug_powerOff_noStim,...
            head_plug_powerOff_noStim,...
            head_plug_powerOn_disabled,...
            head_plug_powerOn_noStim0,...
            head_plug_powerOn_noStim30,...
            head_plug_powerOn_noStim60,...
            head_plug_powerOn_yesStim0,...
            head_plug_powerOn_yesStim30_epochs,...
            head_plug_powerOn_yesStim60_epochs,...
            zeros(62,N),...    % Leave place for the "yesStim30_noPulse"
            zeros(62,N));      % Leave place for the "yesStim60_noPulse"

% Set short names for an easy identification and reference to the datasets
names_rec={'#1: air',...
    '#2: head',...
    '#3: plugged',...
    '#4: powerOn - disabled',...
    '#5: abled - noStim0',...
    '#6: abled - noStim30',...
    '#7: abled - noStim60',...
    '#8: yesStim0',...
    '#9: yesStim30',...
    '#10: yesStim60'};

% Create a variable that defines a specific color RGB for each recording
color_rec=[0 0 0; 0 0 178; 185 185 185; 255 127 0; 121 195 97; 158 102 171;...
    205 122 88; 215 127 179; 249 166 90; 241 89 95]'/255;
```

Processing

```
% Remove baseline. The data w/o BSL is saved in the new variable data_noBSL
data_noBSL=zeros(62,N,12);
for rec=1:10
    for ch=1:62
        data_noBSL(ch,:,rec)=detrend(data(ch,:,rec));
    end
end

    % Check - Remove baseline
    figure
    hold on
    plot(t,data(5,:,1))
    plot(t,data_noBSL(5,:,1))
    xlabel('Time (s)')
    ylabel ('Amplitude (uV)')
    title('Checking baseline removal')
    legend('EEG raw',...
        'EEG after baseline removal')
```

```
% Remove pulses from the recordings with "yesStim", rec #9 and #10
load('locPulses_analysis.mat'); % Obtained from the EEGLAB variable ALLEEG.
                            % It is found in "ALLEEG(11).event.latency"
locPulses=locPulses+260; % Correct latency. As the found event does not
                         % exactly coincide with the pulse, a latency is
                         % determined arbitrarily and the locations of the
                         % pulses are corrected.
locPulses=locPulses(1:14); % Select only the pulses in the analyzed data
rangeLeft=50;
rangeRight=100;
ch=1:62;

data_noBSL(:,:,11)=removePulses(data_noBSL(:,:,9),locPulses,rangeLeft,rangeRight,ch);
data_noBSL(:,:,12)=removePulses(data_noBSL(:,:,10),locPulses,rangeLeft,rangeRight,ch);

    % This is the new saved data:
    % data_noBSL(:,:,11) --> yesStim30_noBSL_noPulses
    % data_noBSL(:,:,12) --> yesStim60_noBSL_noPulses

    % Check - Remove pulses
    t_locPulses=locPulses./5000;
    figure
    hold on
    plot(t,data_noBSL(5,:,10))
    plot(t,data_noBSL(5,:,12))
    plot(t_locPulses,zeros(1,14),'*')
    xlabel('Time (s)')
    ylabel ('Amplitude (uV)')
    title('Checking pulses removal')
    legend('EEG with pulses',...
        'EEG after removing pulses',...
        'Pulse location')

% Compute FFTs. All FFTs are saved in the variable X.
f=(0:N/2-1)*fs/N;
X=zeros(12,N/2);
for i=1:12
    X(i,:)=fft2plot(data_noBSL(5,:,i));
end
% The FFTs of #9 and #10 are useless because they were spoilt by the
% pulses, the signals #11 and #12 w/o pulses are the ones to be used.

    % Check - Compute FFTs
    figure
    plot(f,X(1,:))
    xlabel('Frequency (Hz)');
    ylabel('X |(f)|');
    xlim([0 60]);

    % Find the maximum of the line noise peaks for the bar display
        rec_fft=[1:8 11 12]; % Recordings 9 and 10 are not of interest
                             % because they contain pulses

        % Range in which the maximum is searched
        f1=45; n1=(N/2)*f1/(fs/2);
        f2=55; n2=(N/2)*f2/(fs/2);
```

```
    % The variable "X_50" will contain the maximums values
    X_50=zeros(1,12);
    loc_50=zeros(1,12);
    for i=rec_fft
        [max_50,l_50]=max(X(i,n1:n2));
        X_50(i)=max_50;
        loc_50(i)=l_50;
    end
        % Reorganise the variable
        X_50(9:10)=X_50(11:12);
        X_50(11:12)=[];

    % Check - Maximums
    display(X_50')
```

Results

```
% The data is only analysed for channel C3 because is the one aimed for the
% TMS stimulation, hence, the one in which the line noise variation will be
% more clear.
```

Antenna effect

```
% The variable "rec" selects the recordings that are gonna be
% conclusive regarding the studied line noise source.
rec_antenna=[1 2];

% Temporal analysis
    % A segment of 1 second is selected, between second 24 and 25. The
    % baseline is removed so that the data can be compared in the plot.
    figure
    hold on
    t_sec=24:1/fs:25;
    range=24*5000:25*5000; %seconds*5000=samples
    for i=rec_antenna
       plot(t_sec,detrend(data_noBSL(5,range,i)),'Color', color_rec(:,i))
    end
    xlabel('Time (s)')
    ylabel('Amplitude (uV)')
    title('Antenna effect')
    legend(names_rec(rec_antenna));

% Comment: We can already observe a clear line noise in the "head"
% condition. In the "air" condition I observe some periodic pulses that
% I don't understand. These are spike artifacts with not clear explanation.

% Spectral analysis
figure
hold on
for i=rec_antenna
    plot(f,X(i,:),'Color', color_rec(:,i))
end
title('Spectrums comparison');
xlabel('Frequency (Hz)');
ylabel('X |(f)|');
```

```
%     xlim([0 70]);
    xlim([49.5 50.5]);
    legend(names_rec(rec_antenna))

    % Comment: It is very clear how there is antenna effect because the
    % line noise highly increases by just placing the coil on the scalp.
```

Current leaks

```
    rec_leaks=2:7;

    % Temporal analysis
        figure
        hold on
        t_sec=24:1/fs:25;
        range=24*5000:25*5000; %seconds*5000=samples
        for i=rec_leaks
           plot(t_sec,detrend(data_noBSL(5,range,i)),'Color', color_rec(:,i))
        end
        xlabel('Time (s)')
        ylabel('Amplitude (uV)')
        title('Current leaks effect')
        legend(names_rec(rec_leaks))

    % Comment: It is difficult to observe the difference in the line noise
    % between the recordings, this is why a further spectral analysis is
    % done.

    % Spectral analysis
    figure
        % Spectrum
        subplot(1,2,1)
        hold on
        for i=rec_leaks
            plot(f,X(i,:),'Color', color_rec(:,i))
        end
        title('Current leaks effect');
        xlabel('Frequency (Hz)');
        ylabel('X |(f)|');
        xlim([49.5 50.5]);
        legend(names_rec(rec_leaks))
        % Bar plot
        subplot(1,2,2)
        hold on
        for i=rec_leaks
            bar(i,X_50(i),'FaceColor',color_rec(:,i));
        end
        % set(gca,'XTickLabel',names_rec_new(rec_fft)); %Si faig noms mes curts potser
        ylabel('X |(f)|');
        set(gca,'XTickLabel',{' ','',' ','',' ','',' ',''});
        legend(names_rec(rec_leaks));

    % Comment:
        % Unplugged and plugged are almost the same
        % Setting "powerOn" increases a lot the line nosie (leaks)
```

```
    % Setting "abled" keeps the line noise quite similar
    % Selecting intensities causes strange behaviour
```

Stimulation effect

```
rec_stim=[5 8 9 10];
    % 5 is head_plug_powerOn_noStim0
    % 8 is head_plug_powerOn_yesStim0
    % 9 10 are the ones with stim 30% and 60% and noPulses

% Temporal analysis
figure
hold on
t_sec=24:1/fs:25;
range=24*5000:25*5000; %seconds*5000=samples
for i=rec_stim
    plot(t_sec,detrend(data_noBSL(5,range,i)),'Color', color_rec(:,i))
end
xlabel('Time (s)')
ylabel('Amplitude (uV)')
title('Stimulation effect')
legend(names_rec(rec_stim));

% Spectral analysis
figure
    % Spectrum
    subplot(1,2,1)
    hold on
    for i=rec_stim
        plot(f,X(i,:),'Color', color_rec(:,i))
    end
    title('Stimulation effect');
    xlabel('Frequency (Hz)');
    ylabel('X |(f)|');
    xlim([49.50 50.5]);
    legend(names_rec(rec_stim))
    % Bar plot
    subplot(1,2,2)
    set(gcf,'name','Stimulation effect(C3)','numbertitle','off')
    hold on
    for i=rec_stim
        bar(i,X_50(i),'FaceColor',color_rec(:,i));
    end
    ylabel('X |(f)|');
    set(gca,'XTickLabel',{' ','',' ','',' ','',' ',''});
    legend(names_rec(rec_stim));


% Comment: Performing stimulation results in a lower line noise peak,
% even when the selected intensity is zero. However, the time domain
% shows that the line noise is not that small as expressed by the peak
% and the spectrum shows some symmetrical subpeaks with no clear
% explanation. It can be that the device is applying some filtering and
% that the resulting recorded signal is the result of that filtering.
```

Summary of results

```
rec_summary=1:10;

figure
set(gcf,'name','Line noise peak amplitude(C3)','numbertitle','off')
hold on
for i=rec_summary
    bar(i,X_50(i),'FaceColor',color_rec(:,i));
end
title('Line noise peak amplitude')
ylabel('|X(f)|');
set(gca,'XTickLabel',{' ','',' ','',' ','',' ',''});
legend(names_rec(rec_summary));
```

Conclusions

```
% The origin of the TMS-evoked line noise artifact has been mostly
% clarified. It is due to:
%
%     - Antenna effect caused by the coil placed on the head
%     - Current leaks that occur when powering on the device
%
% Even though the stimulation recordings show unexplainable results, it can
% be assumed that the recorded line noise comes from the two mentioned
% sources.
```

Line noise correction

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This script aims to show the procedure in which the EEG line noise
% artifact can be corrected by combining subtraction using an extracephalic
% reference electrode and averaging.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Load data

```
% The algorithm is applied to EEG-TMS epoched data. Once the data is
% recorded, it is imported to EEGLAB and epochs are there extracted.
% The raw recording found on Files/LN_correction/Raw is managed with EEGLAB
% in the next way:
%
%   1. Start EEGLAB         >>eeglab
%   2. Import recordings    File>Import data>Using EEGLAB functions and
%                           plugins>From Brain Vis. Rec. .vhdr file
%
% Once the all recording is imported in EEGLAB, epochs are extracted.
%
%   3. Extract epochs         Tools>Extract epochs
%                           In the pop-up, select the triggers where the
%                           pulses are located as the reference event.
%                           Then, a range from 1 second before the event
%                           until 3.5 seconds after the event.
%
% Finally, the variable ALLEEG with the two datasets is saved.
%
%   4. Save ALLEEG variable >>save('ALLEEG','-v7.3');
%
% This is the variable containing all the data used in this script.

load ('data_correction.mat');
fs=5000; % Sampling frequency

% Decide a time length to be processed
N=450000; % 450000 samples (1.5 minutes)
t=0:1/fs:(N-1)*(1/fs);

% Select the dataset containing the epoched stim data
x=ALLEEG(2).data(:,1:N);

% Remove the baseline
x_noBSL=zeros(62,N);
for ch=1:62
    x_noBSL(ch,:)=detrend(x(ch,:));
end

    % Check - Baseline removal
    figure
    hold on
    plot(t,x(5,:))
    plot(t,x_noBSL(5,:))
    xlim([0 20])
    ylim([-5000 1000])
```

```
    xlabel('Time(s)')
    ylabel('Amplitude(uV)')
    title('Baseline removal')
    legend('EEG ',...
        'EEG no baseline')
```

Subtraction

Extracephalic reference analysis

```
% A first view is shown to decide what changes must to be done to the
% reference signal so that it can be used for subtraction.

% Channel 20 (coil) is selected because it recorded the ref signal.
% Channel 5 (C3) is selected as example, it is the one just under the coil.

Ref=x(20,:);    %Extracephalic reference channel
EEG_C3=x(5,:);  %EEG channel

figure
subplot(2,1,1)
hold on
plot(t,EEG_C3,'b')
plot(t,Ref,'k')
xlim([9.95 10.3])
title('Observing the amplitude...')
xlabel('Time (s)')
ylabel('Amplitude (uV)')
legend('EEG signal (C3)','Reference signal');

subplot(2,1,2)
hold on
plot(t,EEG_C3,'b')
plot(t,Ref,'k')
xlim([9.95 10.3])
ylim([-4060 -3940])
title('Observing the phase...')
xlabel('Time (s)')
ylabel('Amplitude (uV)')
legend('EEG signal (C3)','Reference signal');

% Amplitude: Much different. Higher in the Ref electrode.
% Phase: There is phase shift between them.
```

```
% The reference signal must look as similar as possible to the line noise
% component of the aimed EEG channel to be properly subtracted.
%
% Amplitude -----> Reescale ther reference
% Phase shift ---> Align the signals
```

Amplitude fitting

```
% The reference signal will be reescaled by multiplying it by the rate
% resulting from dividing the aimed EEG channel amplitude and the reference
% signal amplitude.

        % Rate = EEG_amplitude/Ref_amplitude;
        % Ref_modified = Rate*Ref;

% Finding the amplitudes is done by dividing the signals in cycle unit of
% 100 samples (20ms) finding the maximum and minimum in each cycle. Then,
% their subtraction will result in the cycle amplitudes. Finally, as it is
% known that the line noise is constant in amplitude, all the computed
% amplitudes of the same signal are averaged and a final unique amplitude
% is found for each signal.
```

Define cycle units

```
% The next figures show this division
figure
    %Ref channel
    subplot(2,1,1)
    plot(t,Ref,'b')
    xlim([9.95 10.3])
    ylim([-2e4 1e4])
    title('Reference signal division')
    xlabel('Time (s)')
    ylabel('Amplitude (uV)')
    seg_start=0:0.02:60;
    vline_mod(seg_start, 'k')

    %EEG channel (C3)
    subplot(2,1,2)
    plot(t,EEG_C3,'b')
    xlim([9.95 10.3])
    ylim([-4060 -3940])
    title('EEG signal division')
    xlabel('Time (s)')
    ylabel('Amplitude (uV)')
    seg_start=0:0.02:60;
    vline_mod(seg_start, 'k')

% There will be a total of 62 channels, 3000 units/ch, 100samples/unit.
u_data=zeros(62,100,N/100);
for i=1:62
    u_data(i,:,:)=reshape(x(i,:),[100,N/100]);
end

    % Check - Defining cycle units
    figure
    hold on
    plot(u_data(10,:,6),'*') %(ch,samples,unit)
    plot(x(10,501:600))   %(ch,samples)
    xlabel('Samples')
    ylabel('Amplitude (uV)')
```

```
    title('Checking reshape()')
    legend('Reshaped data','Original data');
```

Find maximums & minimums

```
maxs=zeros(62,N/100);
mins=zeros(62,N/100);
for i=1:62
    for j=1:N/100
        maxs(i,j)=max(u_data(i,:,j));
        mins(i,j)=min(u_data(i,:,j));
    end
end
```

Compute amplitudes

```
amps=maxs-mins;

    % Check - Compute amplitudes
    figure
    plot(amps(5,:),'.')
    ylim([0 200])
    xlabel('Units')
    ylabel('Amplitude (uV)')
    title('Amplitudes of each unit of an EEG channel (C3)')
```

Compute rate

```
% In a same channel, there is a different rate for every unit. However, we
% will compute the average and use a unique rate for each channel.
rate_amp=zeros(62,N/100); %Ch 20 should result in 1 and will not be used.
for i=1:62
    rate_amp(i,:)=amps(i,:)./amps(20,:);
end

    % Check - Compute rate, reasonable rate values around 1-2.
    figure
    stem(rate_amp(5,:))
    ylim([0 0.03])
    xlabel('Units')
    ylabel('Rate')
    title('Rate of each unit of an EEG channel (C3)')

% Clean rate from pulse spoil:
    % Take the pulse sample and the 2 after samples, then, substitute them by the
    % average of the sample just before and after these 3 samples.

pulseSpoil=51:225:4550; % Location of the pulses in the units
                        % It is looked manually in the plotted data
rate_amp_clean=rate_amp;
for ch=1:62
    for i=1:length(pulseSpoil)
        rate_amp_clean(ch,pulseSpoil(i):pulseSpoil(i)+2)=(rate_amp(ch,pulseSpoil(i)-
1)+rate_amp(ch,pulseSpoil(i)+3))/2;
    end
```

```
end

    % Check - Clean rate from pulse spoil
    figure
    hold on
    stem(rate_amp(5,:))
    stem(rate_amp_clean(5,:))
    xlim([494 510])
    ylim([0 0.1])
    title('Rate correction')
    xlabel('Samples')
    ylabel('Rate')
    legend('Rates w/ pulse spoil',...
       'Rates after correction')

% Compute a unique averaged rate for every channel
avg_rate_amp=mean(rate_amp_clean,2);

    % Check - Averaged rates, OK because the Ref rate is 1.
    display(avg_rate_amp)
    display(avg_rate_amp(20))
```

Compute modified reference signal

```
% For every channel there will be a different reference signal with a
% different amplitude.

        % Ref_modified = Rate*Ref;

Ref_mod=zeros(62,N);
for ch=1:62
    Ref_mod(ch,:)=x_noBSL(20,:)*avg_rate_amp(ch);
end

% Remove pulses from Ref_mod so that when subtracting the pulses don't change
load('locPulses_correction.mat') % Obtained from the EEGLAB variable ALLEEG
                                 % It is found in "ALLEEG(2).event.latency"
locPulses=locPulses';
rangeLeft=50;
rangeRight=100;
ch=1:62;
[Ref_mod_noPulses] = removePulses(Ref_mod,locPulses,rangeLeft,rangeRight,ch);

    % Check - Remove pulses from Ref_mod
    figure
    hold on
    plot(t,Ref_mod(5,:))
    plot(t,Ref_mod_noPulses(5,:))
    xlim([5.2 5.8])
    ylim([-30 30])
    title('Pulse removal from Iz mod')
    xlabel('Time (s)')
    ylabel('Amplitude (uV)')
    legend('Modified reference w/ pulses',...
        'Modified reference w/o pulses')
```

```
    % Check - Amplitude fitting, the amplitudes are similar for C3 and
    % some other neraby channels, for the rest there is no visible 50Hz.
    ch=5;
    figure
    hold on
    plot(t,Ref_mod_noPulses(ch,:))
    plot(t,x_noBSL(ch,:))
    xlim([10.5 11.5])
    ylim([-50 50])
    legend('Modified reference signal','EEG signal')
    title('Checking amplitude fitting')
    xlabel('Time (s)')
    ylabel('Amplitude (uV)')
```

Phase shift correction

```
% In order to find the lag between the EEG signals and their corresponding
% reference signal, a segment without pulse will be extracted from every
% signals and they will be used in the cross-correlation function xcross()
% which will give the lag in samples. After, the lag will be corrected by
% cutting a segment at the start long as the lag is.

% Select the same time-located segment for every channel
ch=5; % Channel C3
figure
hold on
plot(t,x_noBSL(ch,:))
plot(t,Ref_mod(ch,:))
xlim([7 8]) %This segment looks fine because there is no pulse
xlabel('Time (s)')
ylabel('Amplitude (uV)')
title('Selecting the segment')
legend('EEG signal','Ref signal')

t_Left=7;
t_Right=8;
rangeLeft=t_Left*fs;
rangeRight=t_Right*fs;

N_seg=rangeRight-rangeLeft;
t_seg=(rangeLeft-1)/fs:1/fs:(rangeRight-2)/fs;

% Extract segments
x_seg=zeros(62,N_seg);
Iz_seg=zeros(62,N_seg);
for ch=1:62
    x_seg(ch,:)=x_noBSL(ch,rangeLeft:rangeRight-1);
    Iz_seg(ch,:)=Ref_mod(ch,rangeLeft:rangeRight-1);
end

    % Check - Extract segments
    ch=5;
    figure

    subplot(1,2,1)
    hold on
```

```
    plot(t,x_noBSL(ch,:))
    plot(t_seg,x_seg(ch,:))
    xlim([t_Left-2 t_Right+2.5])
    ylim([-150 150])
    xlabel('Time (s)')
    ylabel('Amplitude (uV)')
    title('Extracting segment for EEG signal')
    legend('Entire signal','Segment')

    subplot(1,2,2)
    hold on
    plot(t,Ref_mod(ch,:))
    plot(t_seg,Iz_seg(ch,:))
    xlim([t_Left-2 t_Right+2.5])
    ylim([-20 20])
    xlabel('Time (s)')
    ylabel('Amplitude (uV)')
    title('Extracting segment for reference signal')
    legend('Entire signal','Segment')

% Find the lag: Cross-correlation
corr=zeros(62,9999);
loc=zeros(1,62);
lagDiff=zeros(1,62);
timeDiff=zeros(1,62);
for ch=1:62
    [corr(ch,:),lag]=xcorr(x_seg(ch,:),Iz_seg(ch,:),'none'); % "none" scaling
    [~,loc(ch)] = max(abs(corr(ch,:)));
    lagDiff(ch)=lag(loc(ch));
    timeDiff(ch)=lagDiff(ch)/fs;    %lag in time
end

    % Check - Cross-correlation
    ch=5;
    figure
    plot(lag,abs(corr(ch,:)))
%     xlim([-110 110])
    xlabel('Samples')
    title('Cross-correlation EEG-RefSignal')

    % Check - Find the lag
    display(lagDiff')

% Correcting the lag
lagDiff_pos=lagDiff;
x_a=x_noBSL;
Iz_a=Ref_mod_noPulses;
for ch=1:62
    % For positive lag
    if (lagDiff(ch)>0)
        % Correct lag
        x_i=x_noBSL(ch,lagDiff(ch)+1:end);
        % Get the oringinal length with zero-padding
        x_i=[x_i zeros(1,N-length(x_i))];
        x_a(ch,:)=x_i;
    % For negative lag
    else
```

```
        % Correct lag
        Iz_i=Ref_mod_noPulses(ch,-lagDiff(ch)+1:end);
        % Get the oringinal length with zero-padding
        Iz_i=[Iz_i zeros(1,N-length(Iz_i))];
        Iz_a(ch,:)=Iz_i;
    end
end

    % Check - Corrected phase shift, fitted reference signal
    ch=5;
%     ch=48; % C6, far from the TMS aimed location
    display(lagDiff(ch))
    figure
    hold on
    plot(t,x_a(ch,:))
    plot(t,Iz_a(ch,:))
    xlabel('Time (s)')
    ylabel('Amplitude (uV)')
    title('Phase shift correction')
    legend('EEG signal','Reference signal')


% Comment: The EEG signal C3 and the ones close to the TMS aimed location
% look aligned, however, the channels far from these locations could not be
% aligned because their line noise is too low.

% This reference signal fitting does not work for all channels, but only
% for the ones close to the TMS aimed region. Only those can be corrected
% by subtraction
```

Compute subtraction

```
% Taking the EEG channels and subtracting their respective fitted reference
% This will only be done in C3 as it has the better fitting results
y=zeros(62,N);

% Select the channel to apply subtraction
ch_subtr=5;
y(ch_subtr,:)=x_a(ch_subtr,:)-Iz_a(ch_subtr,:);

    % Check - Subtraction
        figure
        hold on
        plot(t,x_a(ch_subtr,:))
        plot(t,y(ch_subtr,:))
        xlim([0.95 1.5])
        ylim([-60 60])
        xlabel('Time (s)')
        ylabel('Amplitude (uV)')
        title('Subtraction result')
        legend('EEG raw','EEG after subtraction')
```

Averaging

```
% First, the data will be separated again in epochs, then, all the epochs
% will be summed and its result will be divided by the number of epochs.

            % AVG Epochs = sum(epochs)/N_epochs

% Epoch channels that were not subtracted
y_ep=zeros(62,22500,20);
for ch=1:62
    y_ep(ch,:,:)=reshape(x_noBSL(ch,:),[22500 20]);
end

    % Check - Epoch non-subtracted channels
    figure
    hold on
    plot(y_ep(5,:,10),'*') %(ch,samples,unit)
    plot(x_noBSL(5,202501:225000))   %(ch,samples)
    xlim([6000 6500])
    ylim([0 50])
    xlabel('Samples')
    ylabel('Amplitude (uV)')
    title('Checking reshape()')
    legend('Reshaped data','Original data');

% Epoch the subtracted channel and
y_ch=y(ch_subtr,:);
y_ep(ch_subtr,:,:)=vec2mat(y_ch,22500)';
% Add it to the whole filtered EEG variable.
y_ep_ch=y_ep(ch_subtr,:,:);
y_ep_ch=squeeze(y_ep_ch);

    % Check - Epoch subtracted channel
    figure
    hold on
    plot(y_ep(ch_subtr,:,10),'*') %(ch,samples,unit)
    plot(y_ch(202501:225000))   %(samples)
    xlim([3000 3500])
    ylim([-20 40])
    xlabel('Samples')
    ylabel('Amplitude (uV)')
    title('Checking reshape()')
    legend('Reshaped data','Original data');

% Averaging
avg_y=zeros(62,22500);
for ch=1:62
    avg_y(ch,:)=sum(y_ep(ch,:,:),3)/size(y_ep,3);
end

    t_ep=0:1/fs:(22500-1)*(1/fs);

    figure
    hold on
    plot(t_ep,avg_y(5,:)) %averaged epochs after subtraction
    plot(t_ep,y_ch(202501:225000)) %single epoch after subtraction
    title('Averaged epochs result');
```

```
    xlabel('Time (s)');
    ylabel('avg(uV)');
    xlim([0.95 1.5])
    ylim([-40 60]);
    legend('Avg epochs with subtraction',...
        'Single epoch with subtraction')

    %Comment: It is undeniable that the line noise has decreased.
```

Results

```
%Spectral analysis
% Remove pulses using zero-padding
locPulses=5000; % It corresponds to 1 seconds, the location of the unique
                % pulse of a single epoch or the averaged epochs.
rangeLeft=200;
rangeRight=300;
ch=1; %Because only 1 channel is input

% 1. C3 ep1 without subtraction
C3_ep1_woSubtr=detrend(ALLEEG(2).data(5,:,1));
[C3_ep1_woSubtr_noPulse] = removePulses(C3_ep1_woSubtr,locPulses,rangeLeft,rangeRight,ch);

% 2. C3 ep1 after subtraction
C3_ep1_wSubtr=y_ep_ch(:,1)';
[C3_ep1_wSubtr_noPulse] = removePulses(C3_ep1_wSubtr,locPulses,rangeLeft,rangeRight,ch);

% 3. Avg C3 without subtraction
% Get the epoched data
C3_woSubtr=ALLEEG(2).data(5,:,:);
C3_woSubtr=squeeze(C3_woSubtr);
C3_woSubtr=detrend(C3_woSubtr);
avg_C3_woSubtr=sum(C3_woSubtr,2)/size(C3_woSubtr,2);
avg_C3_woSubtr_noPulse=avg_C3_woSubtr;
avg_C3_woSubtr_noPulse(locPulses-rangeLeft:locPulses+rangeRight)=0;

% 4. Avg C3 after subtraction
avg_C3_wSubtr=avg_y(5,:);
[avg_C3_wSubtr_noPulse] = removePulses(avg_C3_wSubtr,locPulses,rangeLeft,rangeRight,ch);

    % Check - Remove pulses
    figure
    hold on
    plot(t_ep,avg_C3_wSubtr)
    plot(t_ep,avg_C3_wSubtr_noPulse)
    plot(locPulses/fs,0,'*')
    xlim([0.8 1.3])
    ylim([-60 60])
    xlabel('Time(s)')
    ylabel('Amplitude(uV)')
    legend('with pulse','without pulse')
    title('Pulse removal');

% Spectral analysis
L=length(avg_C3_wSubtr_noPulse);
```

```
A=fft2plot(C3_ep1_woSubtr_noPulse);
B=fft2plot(C3_ep1_wSubtr_noPulse);
C=fft2plot(avg_C3_woSubtr_noPulse);
D=fft2plot(avg_C3_wSubtr_noPulse);

f=(0:L/2-1)*fs/L;

figure
hold on
plot(f,A);
plot(f,B);
plot(f,C);
plot(f,D);
title('Spectrums comparison');
xlabel('Frequency (Hz)');
ylabel('|AVG(f)|');
xlim([0 60]);
% ylim([0 12e4]);
lgd=legend('Single epoch - no subtraction','Single epoch - yes subtraction',...
    'AVG epochs  - no subtraction','AVG epochs  - yes subtraction')

figure
hold on
plot(f,A);
plot(f,B);
plot(f,C);
plot(f,D);
title('Spectrums comparison');
xlabel('Frequency (Hz)');
ylabel('|AVG(f)|');
xlim([49 51]);
 ylim([0 12e4]);
legend('Single epoch - no subtraction','Single epoch - yes subtraction',...
    'AVG epochs  - no subtraction','AVG epochs  - yes subtraction')

% Temporal analysis
figure
hold on
plot(t_ep,x_a(5,202501:225000))
plot(t_ep,y_ch(202501:225000)) %single epoch after subtraction
plot(t_ep,avg_C3_woSubtr)
plot(t_ep,avg_C3_wSubtr)
title('Comparison in the temporal domain');
xlabel('Time (s)');
ylabel('avg(uV)');
xlim([0.95 1.5])
ylim([-40 60]);
legend('Single epoch - no subtraction','Single epoch - yes subtraction',...
    'AVG epochs  - no subtraction','AVG epochs  - yes subtraction')
```

Auxiliary functions

removePulses()

```
function [EEG_noPulses] = removePulses(EEG,locPulses,rangeLeft,rangeRight,ch)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function removes the TMS pulses of a signal using zero-padding.
% Inputs:   EEG ---------> Signal with pulses
%           locPulses ---> Location of the pulses
%           rangeLeft ---> Samples to be removed before the pulse
%           rangeRight --> Samples to be removed after the pulse
%           ch ----------> Channels to be cleaned
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for channels=ch;
    for i=locPulses
        EEG(channels,i-rangeLeft:i+rangeRight)=0;
    end
    EEG_noPulses=EEG;
end

end
```

fft2plot()

```
function [X] = fft2plot(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Returns the meaningful half of the absolute spectrum of the input signal
% This output is ready to be plotted.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

L=length(x);

A=fft(x);
Aabs=abs(A);
X=Aabs(1:L/2);

end
```