BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

ARTEFACTS REMOVAL FROM BRAIN EEG SIGNALS USING ADAPTIVE ALGORITHMS

NÁZEV PRÁCE

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — **JURAJ HATALA**
AUTOR PRÁCE

SUPERVISOR — **Dr. SADIA SHAKIL, M.Sc.**
VEDOUCÍ PRÁCE

BRNO 2021

Abstract

This thesis covers the problem of artifacts in electroencephalography (EEG) data and the methods used to remove them with a focus on adaptive filtering. Artifacts are an unavoidable part of the EEG method and they have a negative impact on the analysis of the results by covering the brain signals of interest. Adaptive filtering is a versatile method that can be used for removal of these artifacts if the reference signal correlated with the artifact is provided. The primary goal of this thesis is a proposal and implementation of the framework that can be used to apply methods of adaptive filtering on EEG data. The secondary goal is to examine the effectiveness of a novel Q-LMS algorithm on the task of removal of artifacts from EEG as it was not yet used in this scenario. The work is introducing a library in a Python environment for EEG adaptive filtering and shows and evaluates experiments for EEG artifact removal scenarios with a Q-LMS filter implemented in the proposed library. In this library, a user is able to construct customizable filtering pipelines. The library offers a variety of adaptive filters and reference-building methods with a focus on processing neurological data in BIDS format. However, the user is able to share his custom filters with the framework as well as use his own input data and reference signals. The experiments with Q-LMS showed that it is a well-functioning adaptive algorithm yet the filtering results were moderate in contrast to results obtained by other standard adaptive algorithms.

Abstrakt

Tato práce se zabývá problémem artefaktů ve záznamech elektroencefalografie (EEG) a metodami jejich odstranění s důrazem na adaptivní filtrace. Artefakty jsou neodmyslitelnou součástí metody EEG a negativně ovlivňují analýzu výsledků tím, že překrývají zájmové mozkové signály. Adaptivní filtrace je všestrannou metodou, kterou lze použít pro odstranění těchto artefaktů, pokud je k dispozici referenční signál korelovaný s artefaktem. Hlavním cílem této práce je návrh a implementace frameworku, který umožní aplikaci metod adaptivní filtrace na EEG data. Druhotným cílem je posouzení účinnosti nového algoritmu Q-LMS při odstraňování artefaktů z EEG, protože dosud nebyl v tomto scénáři použit. Práce představuje knihovnu v prostředí Python pro adaptivní filtrace EEG a ukazuje a hodnotí experimenty pro scénáře odstraňování artefaktů s použitím Q-LMS filtru implementovaného v navržené knihovně. V této knihovně je uživatel schopen vytvářet přizpůsobitelné filtrační pipeliny. Knihovna nabízí různé adaptivní filtry a metody vytváření referenčního signálu s důrazem na zpracování neurologických dat ve formátu BIDS. Uživatel však může sdílet vlastní filtry s frameworkem a také používat vlastní vstupní data a referenční signály. Experimenty s Q-LMS algoritmem ukázaly, že se jedná o dobře fungující adaptivní algoritmus, avšak výsledky filtrace byly průměrný ve srovnání s výsledky dosaženými jinými standardními adaptivními algoritmy

Keywords

EEG, electroencephalography, adaptive filtering, adaptive algorithm, noise cancelation, EEG artifact, neurology, Python framework, LMS, Q-LMS, reference signal, RLS, human speech signal, cascade filtering, signal processing, EMG, ECG, EOG

Klíčová slova

EEG, elektroencefalografie, adaptivní filtrace, adaptivní algoritmus, potlačení šumu, EEG artefakt, neurologie, Python framework, LMS, Q-LMS, referenční signál, RLS,signál lidské řeči, kaskádová filtrace, zpracování signálů, EMG, EKG, EOG

Reference

HATALA, Juraj. *Artefacts Removal from Brain EEG Signals Using Adaptive Algorithms.* Brno, 2021. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Dr. Sadia Shakil, M.Sc.

Artefacts Removal from Brain EEG Signals Using Adaptive Algorithms

Declaration

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana X... Další informace mi poskytli... Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

....
Juraj Hatala
May 9, 2023

Acknowledgements

I want to thank Dr. Sadia Shakil, M.Sc for taking me under her supervision after my original supervisor Dr. Muhammad Arif had to travel abroad. I am grateful for Dr. Sadia Shakil's guidance and support throughout my work. I also want to thank Dr. Muhammad Arif for his valuable guidance in the first semester and his continued willingness to assist me and share his knowledge, even after no longer being my supervisor.

Contents

Chapter 1

Introduction

With the emersion of Neurological disciplines, the need to analyze brain activity has risen to prominence in the last few decades. Electroencephalography(EEG) is one of the tools used for this purpose that has seen wide employment in medical and scientific circles. Its advantage is its noninvasive nature and relatively cheap setup. However, EEG comes with its own set of problems and one of the biggest is the introduction of artifacts to the recorded data. Adaptive filtering is one of the many methods used for the removal of these artifacts.

The quality of adaptive filtering depends on many factors. There is a large number of adaptive algorithms and the effectiveness of each may differ depending on the situation. To evaluate the effectiveness of adaptive filters researchers need to explore a number of different scenarios. This work proposes a framework in a Python environment that would help researchers construct adaptive filtering scenarios on EEG data. A secondary goal is to use this proposed framework for the evaluation of the novel adaptive algorithm Quantum least mean squared(Q-LMS) for EEG artifact removal purposes and compare its capability with other algorithms.

Removal of EEG artifacts is a meaningful and interesting process. EEG has been used for many years to diagnose medical patients and effective filtering of artifacts is an essential part of this process. Also, after the recent progress in the development of the Brain-computer interface(BCI) that can be seen for example in projects of Neuralink company, brain signal processing may see new applications for commercial use in the near future.

In the next chapter, the EEG method is examined in detail. There is a review of history and motivation, then the practical challenges and examination of neural-based waves and their implications in experiments. After that, the problem of EEG artifacts is introduced and the artifacts are then divided into different types. Lastly, there is a review of methods that are currently commonly in use for artifact removal together with their possible limitation.

In the third chapter, there is a detailed introduction to the adaptive filtering method. Then there is an explanation of four of the adaptive algorithms, that will be part of the framework.

In the fourth chapter, a proposal for adaptive filtering is made based on the information gained from the proceeding chapters and reviewed literature. Individual functions of the framework are explained in detail using pseudocodes and then the internal structure of framework implementation is described in one of the sections.

In the fifth chapter, a comparison with existing frameworks in the Python environment is made. These frameworks overlap in function with the proposed framework and the differences between them are pointed out.

In this chapter two experiments are made with the help of the proposed framework. These experiments aim to validate the novel Q-LMS algorithm and explore its capabilities. Also, these experiments aim to test the proposed framework in practice and show its ability to play a part in meaningful experiments.

In the last chapter, the final conclusion about this work is made. The upsides and downsides are critically evaluated and the next possible steps for research are suggested.

Chapter 2

Introduction to Electroencephalography and to the problem of artifacts

This chapter serves as an introduction to the field of Electroencephalography(EEG) and the problem of artifacts in EEG signals. First, there is a short overview of Electroencephalography as a scientific field, next, there is an examination of known EEG artifacts, and after that, a list of standard methods that are regularly used to deal with EEG artifacts.

2.1 Electroencephalography overview

Electroencephalography(EEG) is an old method of studying brain activity. For the first time, it was described in 1929 in a paper by Hans Berger. Berger believed that the human brain is able to send telepathic signals so he started to examine electrical and thermal fluctuation around the scalp.[39] He did not find proof of a telepathic connection yet the method he developed found its way into neurological practice. In the present day, EEG is used primarily for diagnosing epilepsy and sleep disorders [2] but also in many BCI applications.

EEG is recorded using small electrodes distributed around the skull. The number of electrodes varies, one of the standards adopted by The International Federation of Clinical Neurophysiology is an international 10 - 20 electrode placement protocol that describes the placement of 21 electrodes[24], but there are many applications using 35 - channel, 125-channel, or even high-density 256 - channel. The effects of different numbers of electrodes on EEG focused on recording during mobile activities have been explored in an article in the Journal of Behavioral and Brain Science.[19] With an increased number of electrodes quality of captured EEG rises but so does the cost and setup becomes more complex and time-consuming.

It's important to understand that when we are displaying and analyzing EEG the measured voltage is defined by the difference in electric potentials between a reference electrode and an active electrode. After the measurement, we can make different interpretations of the measured data by choosing different references for electrodes. The arrangement of reference electrodes and active electrodes is called montage 2.1.[2]

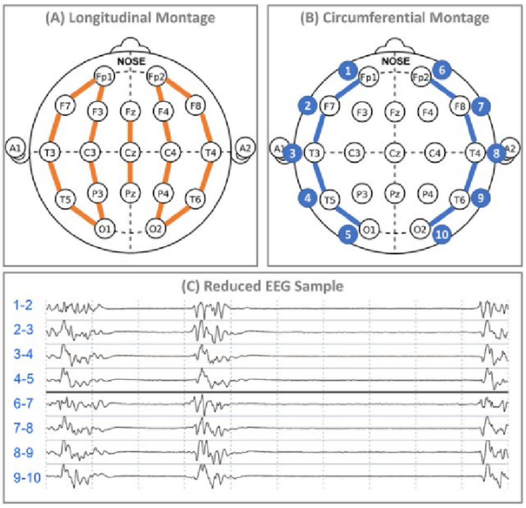


Figure 2.1: (A) and (B) show the positional arrangement of electrodes on the scalp i.e. montage. (C) then show the captured EEG signals. The figure was taken from paper [37].

Researchers recognize multiple neural-based waves that can occur in EEG signals. These waves contain a lot of important pieces of information and are the main components for deriving results from experiments. Table 2.1 shows frequency ranges and functions of various frequency bands in EEG.

Name	Frequency	Function
Gamma	30 - 100 Hz	Rare in the human brain. occurs during the process of combining different senses such as sound and sight.
Beta	14 - 26 Hz	Found only in healthy adults during active thinking, paying attention, and solving critical problems.
Alpha	8 - 13 Hz	Associated with wakefulness, closing the eye, effortless alertness, and creativity.
Mu	8 - 13 Hz	Shows synchronous firing of motor neurons. They are overlapping with other brain waves.
Theta	4 - 8 Hz	Normal for young children. In older children and adults, they are observed during arousal or in a sleepiness state. Also observed during meditation.
Delta	0.5 - 4.0 Hz	Associated with deep sleep.

Table 2.1: Brain waves categorization as described in the book by Nidal, K. and Malik, A. S.[26].

2.2 EEG artifacts

One of the most prominent problems in EEG analysis is dealing with EEG artifacts. EEG is by nature a very weak signal so the measurement requires sensitive tools. These tools can pick many signals from the body, or the environment around them. This means that signals from different sources may cover and obscure EEG signals[28], that the researchers want to observe. Therefore, by EEG artifact, we understand any potential fluctuations of non-neural origin[26]. It is important to understand and study them, in order to prevent their negative effects on EEG analysis by removing them. We can distinguish two main classes of artifacts, Physiological and Non-Physiological[32].

Physiological artifacts

Artifacts that are related to internal body functions.

- Electromyographic artifacts(EMG) - Artifacts that are produced by contractions of muscles. Muscle movement generates a small electrical current that can be picked up by EEG electrodes and pollute the signal. These artifacts are hard to avoid, even if the subject is asked to do as little movement as possible, they won't be able to completely avoid it. In the book by Nidal, Kamel, and Malik, Aamir Saeed [26], in section 1.4.1.2, the EMG artifacts are shown on such a subject. In some real-world applications, the artifacts may get exponentially bigger so dealing with them is of big importance. EMG of skeletal and facial muscles affects EEG signals directly because their sources are located close to measuring electrodes. The frequency of skeletal EMG is in the range of 0 to 200 Hz and is built of more distinct components. We can see that so-called EMG beta components that are in the range of 20 to 30 Hz closely resemble EEG beta waves [11] which makes their filtration without losing important EEG information a challenging task. Other EMG components affect alpha and delta waves as well. A big problem in EMG artifact removal is the inability to establish a reference channel because the noise source is not localized like in EOG or ECG so there is no way of using an additional reference diode effectively. This makes many filtering methods, like adaptive filtering or regression dependent on reference channel ineffective.

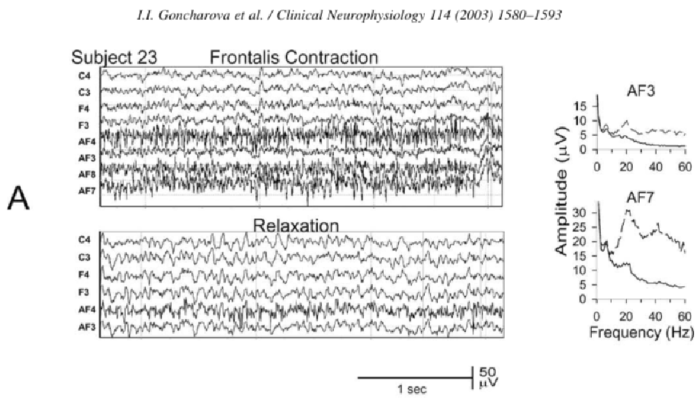


Figure 2.2: This recorded data introduced in [11] show the true impact of frontalis muscle contraction on EEG data. Frontalis muscles are muscles of the forehead.

- Ocular artifacts (EOG) - Artifacts that arise from blinks, eye movement, or other visual stimuli. Eye movement itself produces multiple artifacts of multiple mechanical causes with different amounts of disturbance to EEG so removing a simple change of eye directions may be a very complex task. These artifacts interact with neural signals from the frontal parts of the brain and they are characterized as waves of higher amplitude and lower frequency than EEG signals.

  During eye blink, the eyelid is sliding over the cornea, the transparent front part of the eye, which is positively charged with respect to the forehead. The change of resistance between the forehead and the cornea produces spikes of electric signal that contaminate EEG measurement. During larger eye movements, the change of orientation of an eyeball causes interaction between the positively charged cornea and negatively charged retina, part of an eye that covers the back part of an eyeball, this results in a small electrical signal that will be too picked by EEG electrodes.[28]

  In 2.3 we can see a depiction of standard EOG artifacts and their influence on EEG signal. Artifact has been extracted from the HEOG(horizontal Electrooculogram), a channel that some datasets include. This channel should contain a strong presence of horizontal EOG artifacts for artifact removal purposes. The clear EEG has been approximated by filtration.

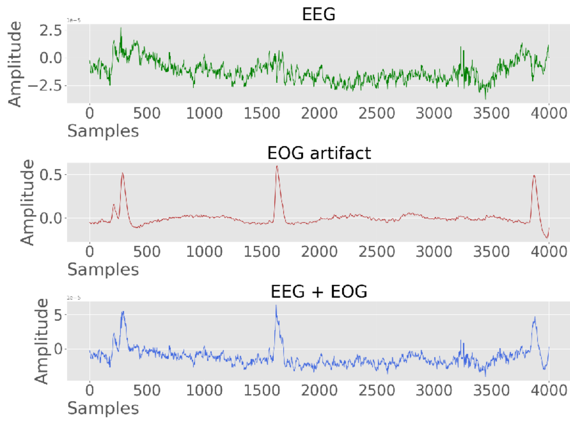


Figure 2.3: Depiction of EOG artifact in EEG signal.

- Electrocardiogram artifacts (ECG) - The heart can produce EEG artifacts as a result of a rhythmic contraction of cardiac muscles that generates an electric field that can affect potentials on the surface of the skull. ECG artifacts have periodic characters and it is possible to generate an artificial ECG signal and use it for reference[17]

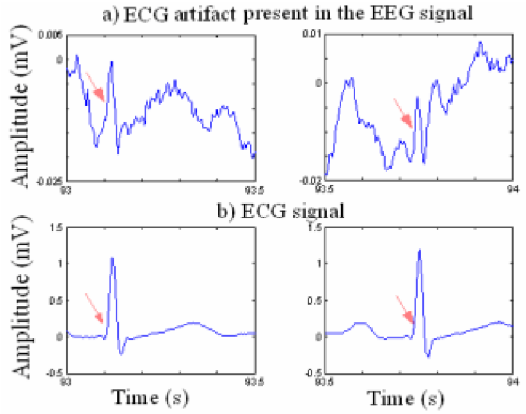


Figure 2.4: In the figure we can see captured ECG signal and subsequent artifact in the EEG signal. The artifact was shown in[6].

Name	Frequency
EOG	delta and theta bands
EMG	0 - 200 Hz
ECG	30 - 100 Hz

Table 2.2: Artifacts categorization by frequency.

Non-physiological artifacts

Artifacts are caused by external sources of the environment of the experimental setup.

- Electrode and equipment artifacts - Any small movement of an electrode during measurement can cause big disturbances in a signal, the problem occurs when the EEG electrodes partially lose contact with the skull. Therefore, it is important to avoid any manipulation with electrodes during an experiment by the researcher or patient. Unequal or too high impedance on electrodes poses a problem, too. The impedance should be held lower than 5000 ohms. When the bipolar placement of electrodes is used, it is important to place electrodes properly. When not the resulting signal will contain additional artifacts. It is good to use standard well-tested equipment. It is important to keep it in good condition and check it for possible malfunctions before every use.[26]

- Artifacts from the environment - Electrical devices like laptops and phones present can introduce electromagnetic waves that may interfere with recording. Movements and sounds around the room where the experiment is conducted may also introduce potential noises. Possible causes of environmental artifacts are listed with examples in the book by Sazgar, Mona, and Young, Michael G.[32]. These artifacts may be filtered out, but it is better to keep the original signal as clean as possible, therefore it is advised to follow strict protocols during experiments that are designed to prevent any unnecessary corruption of the measured signal. An example of such protocol is Recording Protocol for Cognitive and Affective Human Neuroscience Research.[8]

2.3 Common methods of EEG artifact removal

EEG is a widely used method, especially in the medical field and artifact filtering has always been a concern. Over the years many methods were proposed and tested. Different applications desire different qualities, some need very good results in terms of filtering, and others need to be as fast as possible. Also, cost and setup difficulty plays a role. This section presents a few standard methods that see wide usage over the field.

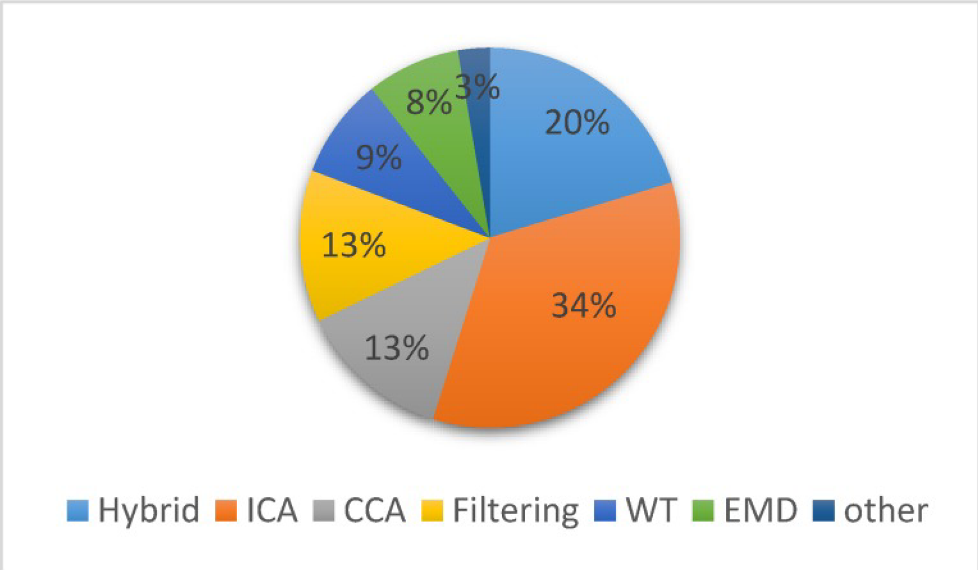


Figure 2.5: Diagram that shows a comparison in the usage of different methods for artifact removal in EEG research. Diagram was presented in [17] where it is shown in Figure 1.

ICA(Independent Component Analysis)

Independent component analysis(ICA) of the most widely used methods for the removal of EEG artifacts. The ICA method was developed to solve the Blind Source Separation problem which describes the need to separate multiple sound (signal) sources recorded simultaneously on the recording device. ICA can solve this problem very well as long as there are as many recording devices as there are different sources of signals. The method assumes that signal sources are instantaneously linear mixtures of cerebral and artifactual sources and it decomposes them into statistically independent components. This approach works well in EEG analysis[17], as we have electrodes capturing signals from many different sources (ideally neuron groups) at once, where some of them may be sources of artifacts. In standard practice, researchers run recorded data through the ICA algorithm, which decomposes EEG signals into a set of individual components. Researchers inspect them manually and if they see that some component resembles an artifact, they remove it from the set. After the inspection, a new signal is reversely built from the remaining components.

The approach is suitable for clinical practice yet it has some shortcomings. The staff reviewing results needs to be highly trained and experienced. Inspecting waves of EEG signals with their chaotic nature is not an intuitive task especially when we are working with many channeled devices. There are also many opportunities for errors. Programs for displaying EEG datasets offer many tools like low-pass or high-pass filters, amplitude multiplication, time axis control, or notch filters. Simple changes in these configurations can greatly impact data presentation; if they go unnoticed, they can easily cloud the researcher's judgment. For example, if somebody changes the size of a displayed chunk of data from 10 seconds to 12 seconds all displayed frequencies would appear lower, subsequently, the normal alpha waves could be mistaken for muscle artifacts and removed. The practices of manually analyzing EEG data by researchers and medical personnel were covered in video series by Dr. Jeremy J. Moeller[22].

CCA(Canonical Correlation Analysis)

The method is similar to the ICA yet different in some key ways. CCA is using second-order statistics instead of the higher-order statistics that are used by ICA and it separates components from uncorrelated sources instead of statistically independent sources. The use of second-order statistics brings lower computational complexity which means faster calculation of results than the ICA method.

The CCA for EEG artifact removal showed very good results. The paper about the removal of EMG artifacts[7] shows that for this task CCA outperformed the ICA method. In the paper the hypothesis is made that the EMG signal has a broad frequency range, approximating white noise, thus its autocorrelation will be very low, in opposition to EEG which has much larger autocorrelation. The decomposition was made and the components with low autocorrelation were removed.

The method has similar downsides to ICA but the computation time is shortened.

WT(Wavelet Transform)

Wavelet transform is based on a similar principle as the Fourier transform. Fourier transform represents the analyzed signal as the sum of sine functions but unfortunately, these functions are not determined in time[26]. Therefore when we analyze signals using Fourier transform we can tell what frequencies are represented but cannot tell when these frequencies occur on time axes. Wavelets are functions that are well localized in both time and frequency. There are multiple wavelet functions each preferable for different applications.

For EEG artifact removal, we decompose the channel recording into wavelets. After that thresholding is applied to remove the signals containing artifacts. A clear EEG signal is reconstructed from the remaining data.

Artifact removal using this technique has mixed results and the removed signal will often overlap with spectral properties. For that reason, it is often used in combination with other methods[17].

EMD(Empirical Mode Decomposition)

Empirical mode decomposition(EMD) is very well suited for processing non-static and non-linear signals. It is a data-driven method that decomposes the signal to IMF functions based on information about the signal's amplitude and frequency.

It is problematic to remove artifacts based on intrinsic mode functions(IMF) components due to the overlapping at higher frequencies with EEG but when there is an especially strong presence of artifacts it can outperform other methods like ICA and wavelet transform.[3]

Adaptive Filtering

The method that is closely examined in this paper is one of the regularly used methods for filtering EEG artifacts. Single-channel filtering is done with the help of a reference signal that should be correlated with the noise contained in the input signal. [6]

For EEG filtration firstly the quality reference signal needs to be obtained. Reference signals are normally acquired from other devices for signal capturing like electrodes around the eyes for EOG reference or on muscles for EMG reference. Then one of the many adaptive filtering algorithms has to be chosen and a filtering pipeline created.

Chapter 3

Adaptive Filtering

This chapter is about adaptive filtering. A general adaptive filter will be introduced as well as methods of building reference signals and a few adaptive algorithms. Examined methods will be in later chapters used for artifact filtering.

3.1 General adaptive filter

The function of an adaptive filter is to remove noise from an input, where the filtered noise is a signal correlated to the provided reference. The filter is adapting its coefficients by the chosen algorithm in a way that it is able to generate a signal similar to noise in a filtered signal based on reference. The generated signal is afterward subtracted from the input signal and thus is the noise removed. The difference between filtered signal and filter output is known as error value and it is then also used for adaptation.

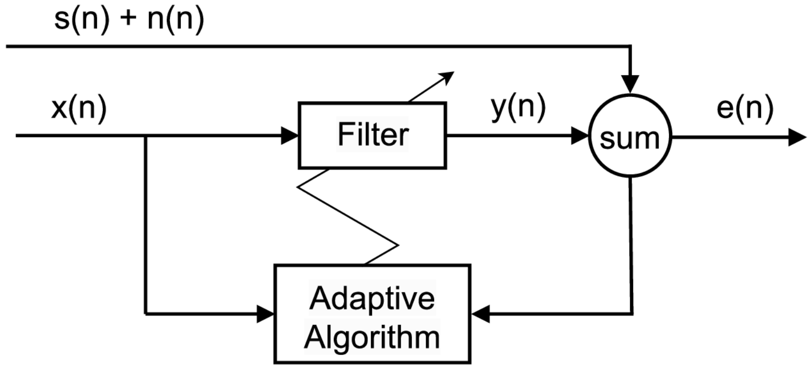


Figure 3.1: Block diagram of the adaptive filter. The diagram is presented as Figure 1 in the article by Mustafa, R., et al.[23]. d(n) is signal composed from s(n), that is the looked for base signal, and n(n),that is the noise that x(n) is referencing.

Adaptive filtering works under the assumption that noise in the filtered signal is correlated to the reference and the rest of the signal is not. If there are parts of the signal, apart from the noise, that is correlated to reference, the filter will filter them out and important data could be lost in the process.

There are two ways to use adaptive filters for noise cancellation and deciding between them is based on the reference signal that is in disposition. Either there is a reference for the looked-for base signal or there is a reference for the noise signal that needs to be filtered out. If the reference is for the base signal,y(n) will be the approximation of the base signal, and e(n) will be the noise, this approach is called adaptive signal enhancement [9]. In the second case, the y(n) will be noise approximation and the e(n) error will be the base signal as shown in Figure 3.1. For filtering EEG artifacts the second approach is usually used[6].

A classic example of the usage of adaptive filters is for filtering echoes in audio calls. When the calls are made from inside of buildings, the received voice of the speaker on the other side will bounce off the walls and return back to them through the listener's microphone leading to an unpleasant effect. For clear communication, this effect needs to be eliminated. The echoes are correlated with the incoming audio signals, so they can be used as reference signals for the adaptive filter that will filter our microphone input. The adaptive algorithm will take care of adjusting the reference signal in a way that it matches the echos and the output will be subsequently subtracted from the audio we are sending.

A similar approach can be used for filtering EEG signals. If there is a need to filter certain types of EEG artifacts a reliable reference signal needs to be provided. Reference signals should represent the source of noise as much as possible in order to establish correlation and establishing the reference is one of the big challenges in EEG adaptive filtering. There are many methods for capturing reference signals.

An advantage of adaptive filtering is that the signal can be filtered in real-time, as soon as it is measured. Computational intensity is relatively small compared to some other methods used for EEG artifact filtering like ICA section 2.3 and ICC section 2.3 but still considerable. The possibility of real-time filtering can be useful in applications like BCI(Brain Computer Interface)[33] where the EEG signals must be processed rapidly to ensure quick response and good user experience.

3.2 Choosing reference signal

Choosing the right reference signal is an essential part of adaptive filtering. It is important that the reference signal will correlate as much as possible with the artifact we want to filter out yet that it does not correlate with other parts of the original EEG signal so the important pieces of information will not be lost. Every artifact is different in nature and comes from different sources and some are easier to capture or define than others. Therefore multiple methods for obtaining different reference signals are being used each suitable for filtering different types of noise[6].

- Capturing reference from the source - When we can reliably identify the artifact source a special device can be selected to capture signals close to the source of an artifact. It is an analogy to noise filtering in audio signals when one microphone is localized near the source of noise to capture it and filter it from the second microphone localized nearby that is used for communication. For EEG referencing electrodes can be placed near the eyes of the subject to capture EOG artifacts. With the introduction of new devices also comes additional costs and setup difficulties.

- Reference separated from original EEG - If the artifacts are predictable and have frequencies that do not overlap with important EEG waves we can separate them directly from captured EEG signals using a bandwidth filter. When we isolate the electrode where the artifact is strongest and capture it we can use it as a reference for filtering it from all other diodes. The problem is that capturing the artifact precisely in its whole form and without capturing some other parts of EEG using bandwidth filters is nearly impossible, so compromises have to be made and a lot of experimentation is required.

- Artificially created reference signal - When the artifact is mathematically well-defined we can generate an artificial signal resembling the actual artifact. The quality of filtering will then depend on the quality of the mathematical definition and its ability to describe real-world noise.

3.3 Adaptive algorithms

There is a large number of proposed adaptive algorithms that are working on various principles. Algorithms that will be part of this work are presented in this section.

LMS(Least Mean Squares)

This most common and simple adaptive algorithm is extensively used for its computational speed, easy implementation, and good performance.

Algorithms adapt all weights in every iteration in a way that will minimize the squared error of the filter. Each weight is updated based on its previous value and order of change of squared error in reaction to the previous value. The learning coefficient μ represents the strength of reaction to changes in squared error, if the value of this constant is too big the filter will be unstable as it will react violently to any change in squared error, on the contrary, if the value is too small the adaptation will be minimal.

Already in 1996, the algorithm has been used for filtering EEG artifacts and it proved to be very effective [25]. Multiple filters can be attached in cascade schema where each is used for filtering different types of artifacts using different references.[6] This approach can be used with any adaptive filter not just LMS.

$$y(n) = \sum_{k=n}^{L} w_k x(n-k) \tag{3.1}$$

The representation of standard FIR that is used in LMS, is given in 3.1. In equations, w_k represents weights that will be updated by the adaptive algorithm.

$$e(n) = d(n) - y(n) \tag{3.2}$$

Error is a difference between the desired signal and filter output in previous iterations.

$$e(n) = d(n) - \sum_{k=n}^{L} w_k x(n-k) \tag{3.3}$$

After subversion of $y(n)$

$$w_k(n-1) = w_k(n) + \mu(-\nabla_k) \tag{3.4}$$

Adaptation of weights

$$\nabla_k = \frac{\partial\{e^2\}}{\partial w_k(n)} \tag{3.5}$$

Representation of the change in squared error.

$$w_k(n-1) = w_k(n) - \mu \frac{\partial\{e^2(n)\}}{\partial w_k(n)} \tag{3.6}$$

Equation 3.4 after substitution

$$w_k(n-1) = w_k(n) - 2\mu e(n) \frac{\partial\{e(n)\}}{\partial w_k(n)} \tag{3.7}$$

Derivation with respect to $w_k(n)$

$$w_k(n-1) = w_k(n) - 2\mu e(n) \frac{\partial\{d(n) - \sum_{k=n}^{L} w_k x(n-k)\}}{\partial w_k(n)} \tag{3.8}$$

Substitution of e(n)

$$w_k(n-1) = w_k(n) - 2\mu e(n) x(n-k) \tag{3.9}$$

Another derivation with respect to $w_k(n)$

Equation 3.9 is the final representation of LMS adaptation. These equations were presented in a paper on cascade filtering of EEG.[6]

RLS(Recursive Least Squares)

RLS shows better results than other methods like LMS when the reference signal was derived from EEG as it displays minimal loss of filtered signal. It also performs well with other forms of the reference signal, therefore, it is considered to be one of the best adaptive algorithms for EEG filtering as stated in [29].

$R(n)$ is an autocorrelation matrix. This represents the relationship between the current value and past values of the series. $R^{-1}(n)$ is transposed value of matrix. $r(n)$ represents the mutual correlation between our reference signal and filtered signal. Lambda is known as the forgetting factor.

$$w_k(n) = R^{-1}(n) r(n) \tag{3.10}$$

$$R^{-1}(n) = \sum_{i=0}^{n} \lambda x(i) x(i)^T \tag{3.11}$$

$$r(n) = \sum_{i=0}^{n} \lambda x(i) d(i)^T \tag{3.12}$$

An iterative approach is needed for an easy transition into a computer program. The algorithm makes use of Woodbury matrix identity $R^{-1}(n)$ where k(n) is a gain vector.

$$R^{-1}(n) = \frac{1}{\lambda}[R^{-1}(n-1) - k(n) x(n)^T R^{-1}(n-1)] \tag{3.13}$$

$$k(n) = \frac{R^{-1}(n-1) x(n)}{\lambda + x(n)^T R^{-1}(n-1) x(n)} \tag{3.14}$$

Adaptation of weight for every iteration will look as follows.

$$w(n) = w(n-1) + k(n) d(n) \tag{3.15}$$

Equations were taken from Padasip library documentation [4].

FLMS(Fraction Least Mean Squares)

In this algorithm, fractional order derivative accompanies classical integer order derivative. This approach has shown promising results in many applications. Convergence is increased together with computational complexity.

The first-order gradient is in this algorithm used together with the fractional order gradient

$$w_k(n) = w_k(n-1) - \frac{\mu_(1)}{2}(\frac{\partial J(n)}{\partial w}) + \mu_(f)(\frac{\partial^f}{\partial w^f}J(n)) \tag{3.16}$$

Caputo and Riemann-Liouville rule for fractional order derivative $g(t) = t^n$. Where D^f is the fractional order operator

$$D^f g(t) = \frac{\Gamma(n+1)}{\Gamma(n-f+1)t^(n-f)} \tag{3.17}$$

Gamma function

$$\Gamma(n) = (n-1)! \tag{3.18}$$

$$\frac{\partial^f}{\partial w^f}J(n) = -2(e(n)u(n))(\frac{\partial^f}{\partial w^f}w(n)) \tag{3.19}$$

After substitution, we get a weight update function

$$w_k(n) = w_k(n-1) + \mu_1 e(n)u(n) + \frac{\mu_f}{\Gamma(2-f)W^(1-f)(n)} \tag{3.20}$$

Equations and information about FLMS were presented in [38]

Q-LMS(Quantum Least Mean Squares)

Q-LMS is a novel algorithm based on a q-derivative concept. Firstly, to avoid possible confusion, it should be stated that there is an algorithm similar in name called Quaternion LMS(QLMS). This is an algorithm intended for the filtration of three and four-dimensional processes and more information about its functioning of it can be found in this paper [34].

Standard LMS algorithm implements a learning rate value μ, If we choose a large learning rate value the rate of convergence will be faster but the steady state will be higher which means that results will be less precise during the steady state. For better results, a smaller learning rate has to be chosen and this will on the other hand negatively affect the rate of convergence. Q-LMS introduces the q parameter that will be correcting the learning rate value during the operation of adaptive filtering. At the start, the learning rate will be pushed to higher values for faster convergence and when approaching the steady state learning rate will be lowered for better results.

New weight update function with the q parameter that is aiming to control the learning rate

$$w_k(n) = w_k(n-1) + \mu\frac{q+1}{2}e(n)u(n) \tag{3.21}$$

The time-varying rule that is used for q parameter update

$$\psi(n+1) = \beta\psi(n) + \gamma e(n)^2 \tag{3.22}$$

The update of the q parameter is defined as this distribution function

$$q(n+1) = \begin{cases} q_{upper} & \text{if } \psi(n+1) > q_{upper} \\ 1 & \text{for } x\psi(n+1) < 1 \\ \psi(n+1) & \text{otherwise} \end{cases} \tag{3.23}$$

q_{upper} is chosen to fulfill the stability condition

$$q_{upper} = \frac{2}{\mu\lambda_{max}} \tag{3.24}$$

λ_{max} is the max value of the correlation matrix that represents correlations between samples of the reference signal in each iteration.

Equations were assembled based on the paper [1].

Few adaptive algorithms have been examined in this section. These adaptive algorithms have been chosen for this work for different reasons. LMS algorithm is standard for adaptive filtering of any kind so it should not be missing in a framework focused on adaptive filtering. RLS algorithm as noted before has already shown very good results on EEG artifact filtering so it should be included also in the framework. Q-LMS algorithm, on the other hand, is a novel algorithm that has, to my knowledge, not yet been tested for EEG artifact removal so I wanted to explore its capabilities in this work using the framework.

3.4 Data

All datasets used in this work are available on OpenNEURON, a free and open platform for validating and sharing BIDS-compliant MRI, PET, MEG, EEG, and EEG data.

For working with datasets we are using MNE-Python, an open-source Python package for visualizing and analyzing human neurophysiological data. Package offers tools for processing the dataset as well as many tools for standard experiments.

- Clear EEG[21] - Dataset contains data from patients diagnosed with Alzheimer's disease and Frontotemporal Dementia (FTD group) as well as patients in full health. Recordings were acquired from the 2nd Department of Neurology of AHEPA General Hospital of Thessaloniki. Part of the dataset are processed data clear of EEG artifacts. Artifact-free data will be used as base EEG data in the experiment on adaptive filtering.

- ECG data [10] - This dataset is combining human-participant high-density EEG with physiological and continuous behavioral metrics. One of these metrics is electrocardiogram(ECG) recording conducted simultaneously with EEG recording. This ECG data will be used in the experiment on adaptive filtering as a reference channel and for the construction of ECG artifacts.

- EOG data[16]- Data are collected on 122 collage-aged-participants that scored reliably high or low in Beck Depression Inventory circa 2008-2010. Experiments were carried out in John J.B. Allen's lab at U Arizona and the dataset contains more than 16 GB of EEG recordings. The EOG channel from this dataset was used in the experiment on adaptive filtering for squaring reference and building EOG artifacts.

- EMG data[27] - Dataset includes synchronized 128-channel EEG, lower leg EMG, neck EMG, EOG, and motion capture data. Data was collected at the University of Michigan by Steven Peterson in the lab of Daniel Ferris. Again EMG channels were used in the experiment for the construction of the EMG artifact and at the same time as a reference channel for the filtration of this artifact for this artifact.
- EOG-ECG data[31] - These data consist of EEG and pupilogram. The dataset should also contain ECG this data however was not found so a different dataset had to be used. Recordings were done on a group of young and a group of older adults that were engaged in auditory-cued reaction time tasks or passively listening to the auditory stimulus.

Chapter 4

Preposal of a framework for adaptive filtering of EEG artifacts

This chapter will include the design of the proposed framework for adaptive filtering of EEG artifacts from EEG signals based on information acquired in earlier sections.

4.1 General design

The aim is to design a framework that can be used to test adaptive algorithms on EEG data to determine their quality, their ability to remove different kinds of EEG artifacts, and their damaging effects on the original EEG. The idea is to find an algorithm you are interested in for your application, explore its capabilities, and compare it to other algorithms all in the span of a few hours. Because of that Framework should be intuitive to use with straightforward concepts.

Design should be simple and easily scalable. After the upload of data user can choose what channels of the dataset he wants to work on. For each of the chosen channels will be subsequently created an object that will contain all data during its lifecycle. Users can then choose from a database of accessible adaptive filters and reference-building options and assign them to channels. Filtering and reference preparation will be executed by run command at the end of a script. Assigning and all managing are done by a controller that is the basis of the framework and encapsulates most of the functionality.

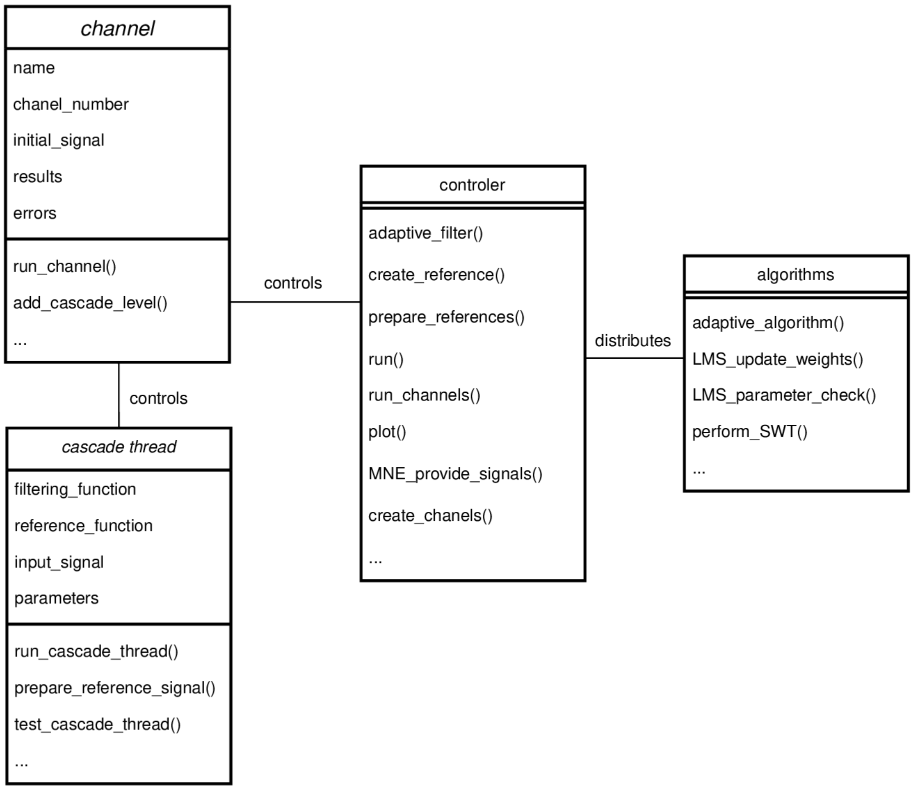


Figure 4.1: Diagram showing application architecture.

For every EEG channel in the dataset that the user wants to work with, one `channel` object is initialized with stored signal data. This object is created with the desired number of `cascade_thread` objects that can each run filtration on their input data. The `channel` object is running each of these threads managing their inputs and storing their results. All algorithms that are used by cascade threads for filtering or reference building are defined in `algorithms` object. The main object of the framework is then assigning references to cascade threads, managing what channels should be run, and carrying out the visualization of results from channels.

4.2 Functions

One channel one filtration

The most basic operation this framework should manage is to use an adaptive filter of choice on one channel of an EEG dataset. There are many recorded channels in one EEG session and we can get information about them with an informative helper function that will print information about the uploaded dataset.

After we know all the channel names we can specify our selection with `MNE_chosen_channels()` function. In this example, one object channel will be created after selection which we can then target with `chosen_channels()` parameter in the succeeding functions for adaptive filter and reference builder specification.

```
MNE_provide_signals(filepath)

```

```
MNE_choose_chanels(chosen_chanels=[FP1})
adaptive_filter(type='LMS', chosen_chanels=[0], parameters)
create_reference(type = 'wavelet', chosen_chanels=[0])

run()
```

Listing 4.1: Python example

Diversification

When we talk about EEG datasets we talk about recordings on headsets that may contain tens or hundreds of electrodes. Each electrode has a different placement on the headset and that is important in the context of artifact filtering where different electrodes may be vulnerable to different types of artifacts. For example, electrodes around the eyes will contain much stronger EOG than electrodes on the head and we may want to adjust our filters accordingly. For these reasons and others, our framework should offer a choice to adjust any channel or group of channels directly in terms of chosen filter, its parameters, or its reference.

```
MNE_provide_signals(filepath)

MNE_choose_chanels(chosen_chanels=[FP1,FP2,AF3])
adaptive_filter(type='LMS', chosen_chanels=[0], parameters)
adaptive_filter(type='LMS', chosen_chanels=[1], parameters)
adaptive_filter(type='RLS', chosen_chanels=[2], parameters)

create_reference(type='wavelet', chosen_chanels=[1])
create_reference(type='wavelet', chosen_chanels=[2,3])

run()
```

Listing 4.2: Python example

Cascade filtration

Implementing adaptive filters in cascade is very common in practical use. Each filter is designed specifically to filter one type of artifact and the reference signal and parameters are chosen accordingly. Each step will be separated by `cascade_step()` which will introduce a new `cascade_thread` object to chosen channels. Each channel will then prepare the correct paths for data.

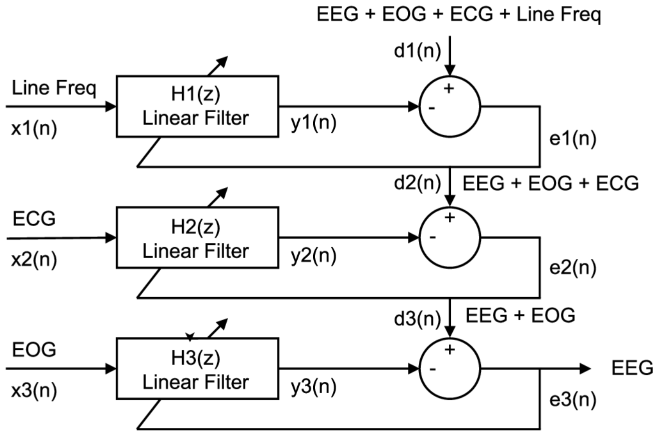


Figure 4.2: Block diagram of cascade filtering based on the diagram proposed in paper [6].

```
MNE_provide_signals(filepath)
MNE_choose_chanels(chosen_chanels)


adaptive_filter(type, chosen_chanels, parameters)
create_reference(type, chosen_chanels)

cascade_step()

adaptive_filter(type, chosen_chanels, parameters)
create_reference(type, chosen_chanels)

cascade_step()

adaptive_filter(type, chosen_chanels, parameters)
create_reference(type = 'wawelet', chosen_chanels)

run()
```

Listing 4.3: Python example

Adding custom algorithms

Adaptive filtering is a wide field that is constantly evolving as new algorithms are being developed regularly. That is why this framework should support adding custom algorithms as simply as possible so the user won't be limited to the already implemented algorithms. As this framework is developed for algorithm testing, users should be able to use it on a wide range of algorithms. This function is aimed at researchers that want to quickly test their new algorithm without the need to build code infrastructure around it or for developers that want to easily explore new possibilities for their products.

There can be many differences between adaptive algorithms but we can find many similarities too. Our three build-in algorithms for example differ only in weight updates, parameters, and parameter updates. That is because all are variations of LMS. These are mathematically significant changes yet in implementation, the changes can be expressed in a single function. If the user has a basic knowledge of Python programming, we can let him define this function and its parameters. Then we can secure that the framework can reliably receive this function through a function pointer and work with it as with any build-in algorithm.

```python
def my_weight_update(weights, samples_vector, error, desired_sample, parameters):
    for i in range(len(weights):
        weight_delta = parameters['learning_rate'] * error * samples_vector[i]
        weights[i] = weights[i] + weight_delta

    return weights, error

-- --

MNE_provide_signals(filepath)
MNE_choose_chanels(chosen_chanels)

param = {'learning_rate':0.1}
custom_LMS_weight_update(fuction_pointer=my_weight_update,
                         chosen_chanels, parameters=param)

create_reference(type, chosen_chanels)

filter_pipeline.run()
```

Listing 4.4: Python example

Another option is to implement the whole algorithm. Adaptive filters have by definition two inputs, a reference signal, and a filtered signal. Users can build a function that will have these two arguments plus parameters arguments, a dictionary containing all needed parameters. If the function can return output and error framework will be able to implement such filter function seamlessly into its workflow.

Reference building

The preparation of reference signal is a big part of the adaptive filtering method as discussed in earlier sections. The results of adaptive filtration will depend largely on the quality of the reference signal and there are many methods of acquiring one.

One of the best ways to get high-quality references is to use a measurement device to record the reference signal near the source of the artifact during the EEG session. This signal will be very good for artifact removal as the artifact and reference will surely have a high correlation as far as the source of the artifact is identified correctly. There are studies that explore reference measuring from different sources and these reference channels are shared in their datasets. Also, a large number of datasets include EOG reference channels because they can be easily recorded with a few more electrodes. In the proposed framework the reference channels that are part of the datasets can be simply used with one function.

If reference channels are not available in the dataset, the framework offers the possibility to use single-channel component decomposition methods for the preparation of reference signals. This method should be used mostly for the evaluation of the filtering process because such a reference signal won't guarantee a good correlation with artifacts and big parts of actual EEG may be filtered out. There is an option to use Wavelet transform for input signal decomposition. The framework uses Stationary Wavelet Transform because the length of the output is the same as the length of the input signal so it can be directly used as a reference. The PyWavelets[20] library is used for this implementation. Another option is to use EMD transform. That is implemented using the EMD Python library[30]. Of course, after computing these decompositions simplest way to get rid of unwanted components is to reconstruct the signal without them, this option is present only for observing functions of adaptive filters without the need to look for reference signals.

There is also an option to add a custom reference signal. This is most useful while designing experiments with custom data and not with data from datasets. This function also adds the possibility to add preprocessing steps on reference signals that are not part of the framework.

4.3 Implementation details

Used tools

Numpy[13], MNE-python[12], Padasip[35], Scipy[36], PyWavelets[20], Matplotlib[14], openneuro[15], Scipy[36], pytest[18],EMD-python[30]

Implementation

The Main Class, users will be working with, is class `framework()`. Here are present all the methods that are listed in 4.2. Inputting dataset is managed by the function `MNE_provide_signals(filepath)` that will read BIDS files and create object Python can work with. Reading BIDS is done with help from MNE-Python library[12], which will create an MNE object that contains signal data that can be extracted into Python lists. Class `channel` manages the filtration of one channel that represents the signal captured from one electrode. Instances of this class will create one or more instances of `cascade_threat` class. `cascade_threat` is the smallest component that takes care of the filtration of one signal. There can be multiple `cascade_threat` instances in one `channel` instance, `channel` manages their inputs and results to create a functional cascade filtering body.

Every adaptive filter is represented by two functions working together, one represents the filtering part, and the other updating of weights. For example, every LMS-based algorithm (LMS, QLMS, FLMS) has the same filter function but a different weight update function. Parameters to these algorithms are given in dictionary `parameters`. Correct parameters are defined in the check functions specific to each algorithm. These check functions are then run before every filtration and make sure the right parameters are chosen.

When a filter is defined by the user using function `adaptive_filter()`, `channel` takes as an input function pointer to the filtering functions, reflecting the choice of filter made by the user, and set this function pointer for his last `cascade_thread`. When the filtering starts `cascade_thread` will be using the defined filtering function. The reference building is done in the same way. When the existing reference signal is chosen the reference building function simply reads the reference. Available algorithms for adaptive filtering and reference building are defined in class `algorithms`.

In a standard experiment with one adaptive filter on each channel, there is only one `cascade_thread` object present in each `channel`. If the user wants to link another filter in succession to the first one he can use the function `cascade_step()`. This function will create a new `cascade_thread` for selected channels that will be added to the `cascade_threads` list in `channel`. The user must then, define the adaptive algorithm and reference that will be used by this thread. Each thread has implemented a self-check that will be run before every filtration. This guarantees that the `cascade_thread` is correctly configured and error messages will inform the user about possible problems.

When the filtration is started by the function `run()` multiple processes are created with the help of the python-multiprocessing library. Each process manages filtration on one `channel` by running function `channel.run_channel()`. This function manages `cascade_threads` of this channel by running each in succession making sure that results from one are used as inputs to another. Filtration of one cascade step is run by the function `cascade_thread.run_cascade_thread()`. After the one process finishes, the changed `channel` object is retrieved from it. New `channel` object contains the results of the filtration and changed state. `stderr` output is also captured after the finish of the process. If any error occurred during filtration, all results are abandoned and an error message is propagated to the user. Only the first captured error message is displayed.

Chapter 5

Comparison with other frameworks

In this chapter, we will look at other options for using adaptive filters and working with EEG datasets. There will be an introduction to other libraries and frameworks, a look at their ability to complete functions of the implemented framework, and their advantages and disadvantages.

5.1 MNE-Python

MNE-Python is a large open-source project that offers a variety of tools for working with neurophysiological data. It doesn't implement LMS-based filters yet it implements many methods for analyzing and visualization of EEG data. The library also implements methods that can be used for the preparation of reference signals for adaptive filters such as independent components analysis. More information can be found in the canonical journal article [12]. If you wanted to use your own filter design you would have to follow these steps:

- Get EEG data from a raw object into Python lists or Numpy arrays
- Organized data based on raw object info
- Design and implement your adaptive filter
- Organize your results into raw object
- Review your results

We can see that is a lot of side management and that's why implemented framework does all these steps for you but the algorithm implementation.

If you would like to use cascade filtering you would have to use the same approach for designing your filters and then you would have to manage the connection between them as one's results are input for another.

Observed advantages:

- Great visualization options
- Options for most state of art operations on neurological data
- Huge community of experts

- Maintenance and constant bug repair and evolving
- Well documented
- Open source license

Observed disadvantages:

- Lacking options for adaptive filtering.
- Rather complicated way to implement adaptive filters into the MNE workflow.
- High complexity.
- Assumed level of expertise in the field of neurology and signal processing for the understanding of documentation and usage.

The MNE-python was not implemented with adaptive filtering in mind, but it is standard for working with neurological data. Any researcher that wants to work with EEG datasets in a Python environment would benefit from the ability to use this library for its wide range of options, active community, and extensive documentation [12].

5.2 Padasip

Padasip [35] is a Python library that offers tools for filtering, prediction, reconstruction, and classification of signals with its main focus on adaptive filtering. Implemented are standard adaptive filters as well as some newer methods in the field. Filters are well-optimized and they are implemented in the Numpy library for better performance.

In order to use your adaptive filter for the EEG dataset by BIDS standard you would have to use some other software to prepare your data like MNE-python and the implement visualization of results.

Observed advantages:

- Better performance
- Good documentation with many examples of usage
- Larger amount of in-build adaptive filtering options
- Available options for data processing
- Option for real-time adaptive filtering applications

Observed disadvantages:

- Need to build visualization manually
- Need to prepare your data manually

Padasip was built to simplify adaptive signal processing tasks within the Python environment. It should be used as a modular part of implementations that are requiring adaptive filtering and these would provide the infrastructure for obtaining data and visualization of results.

Padasip is state-of-the-art adaptive filtering in Python and the implemented framework can use its function of including custom algorithms to leverage Padasip filters. Padasip implementations of LMS and RLS algorithms have been used to visually validate Frameworks implementations. Codes can be found in `experiments/validations` directory of the project repository. We can see some minor differences in RLS algorithm operations that can be caused by the implementation details choice of delta value in the initialization of the identity matrix. Also with the RLS algorithm, we can see the great optimization of the Padasimp library where its implementation provides significantly better performance.

Chapter 6

Experiments

6.1 Noise cancellation in human speech signal

In the first experiment, the effectiveness of 3 algorithms is examined in a noise cancellation scenario on an audio recording of human speech data. The aim of the experiment is to first compare the effectiveness of different algorithms on the less challenging task before moving to EEG artifact filtering as the experiment will focus on filtering simple sinusoidal noise. This noise is much less complex than a standard EEG artifact so the results should be easier to interpret. This experiment also works as the validation of the used adaptive filtering method Q-LMS and showcases of basic functions of the designed framework. Because Q-LMS is a novel algorithm, there are not any freely available implementations that could be used for validation like in the case of LMS and RLS algorithms as discussed in section 5.2. For this reason, it is important to show the function of Q-LMS in an experiment. This experiment was based on a paper examining Q-LMS noise cancellation capabilities[1].

Methodology

Recording of human speech will be used as a base signal. The sinusoid with a frequency of 50 Hz will be then introduced to the base signal as noise. The resulting signal will serve as an input to the adaptive filter. As a reference signal, the same sinusoid will be used but this time with a shifted phase to slightly reduce correlation. Signals will be filtered by three different adaptive algorithms for comparison.

To evaluate results square root differences and RMSE(root mean square error)[5] will be computed. The square root differences will be calculated between the base signal(audio recording) and the output of the adaptive filter. This calculation will show the quality of approximated signal and the process of adaptive filtration. The results of every adaptive filter will be plotted on graphs simultaneously with LMS results as LMS is the simplest of implemented adaptive algorithms and it is well examined by the science community. The differences between the base signal and the output of adaptive filters will also be expressed by RMSE value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{6.1}$$

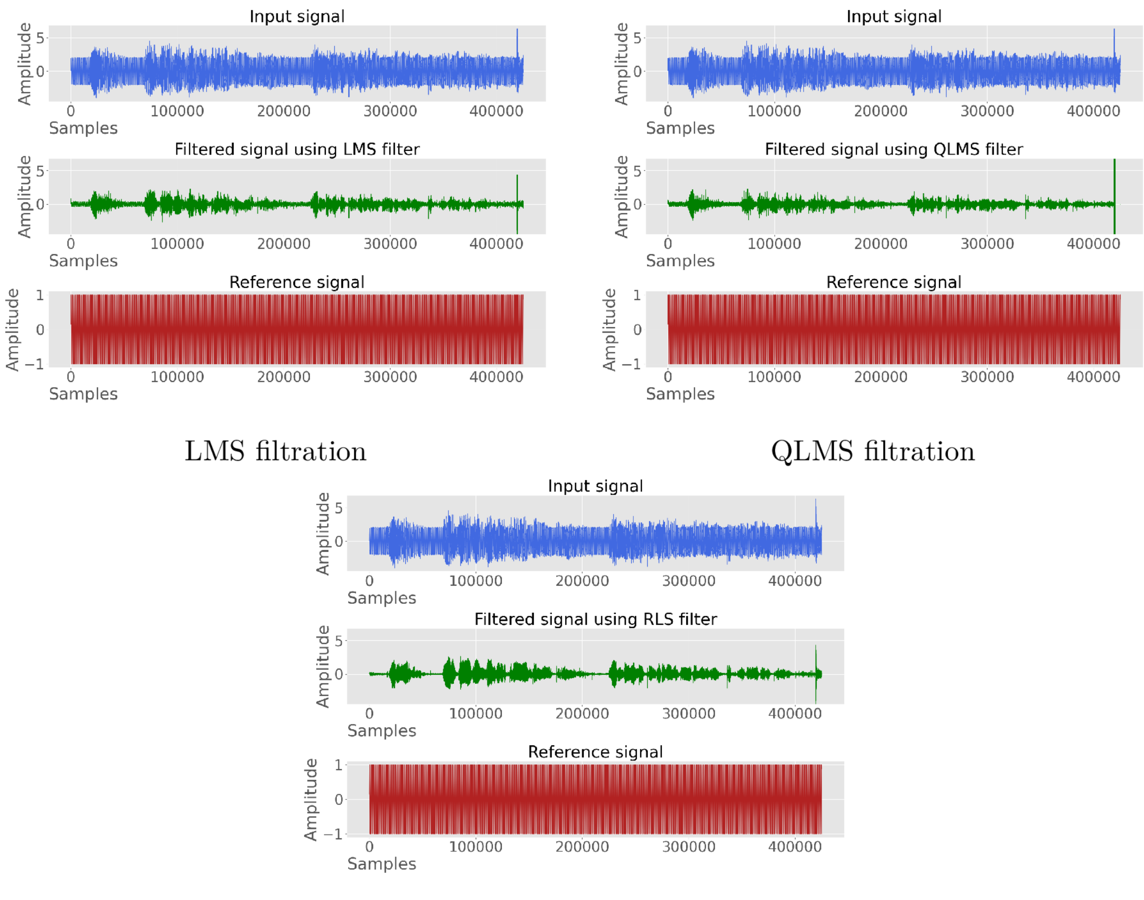
$$e_i = y_i - x_i \tag{6.2}$$

e_i stands for the difference in two signals we want to compare and n stands for the number of samples. e_i^2 then stands for the square difference between two values.

For LMS and Q-LMS algorithms same learning rate of 0.005 is used for a fair comparison. The effects of learning rate values differ between LMS-based and RLS algorithms and RLS would not work properly with the value of 0.005. That is why RLS was used with a learning rate of 0.99. Q-LMS upper q bound is set for 500. This value was determined experimentally where the q parameter doesn't seem to be limited from up and at the same time algorithm retains stability. Gamma and beta values were chosen close to their lower bounds to keep the q parameter closer to its minimal value. Gamma and Beta are 10 and 0.05 respectively.

Results

In figure:6.1 the graphs of raw filtering output are plotted. These graphs were given by filtering framework and there can be seen that especially RLS captured more speech data.



LMS filtration

QLMS filtration

RLS filtration

Figure 6.1: Framework visualization output of adaptive filtering of human speech data.

The next two graphs represent a difference in squared error between LMS and QLMS, and LMS and RLS. We can see right away that the error of RLS is negligible in contrast to LMS and QLMS. The error is the squared difference between the base, noise-less speech signal, and filter output.

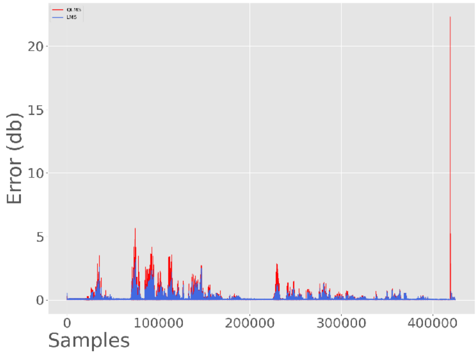


Figure 6.2: LMS against QLMS

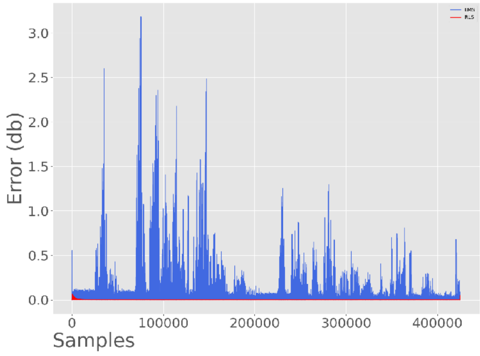
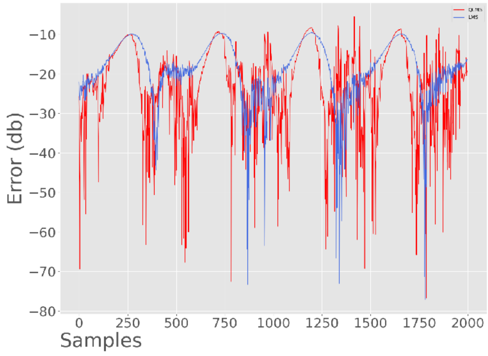


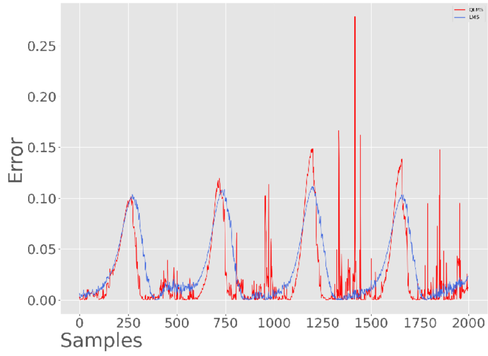
Figure 6.3: LMS against RLS

Figure 6.4: Comparisons of squared error after filtration by different algorithms.

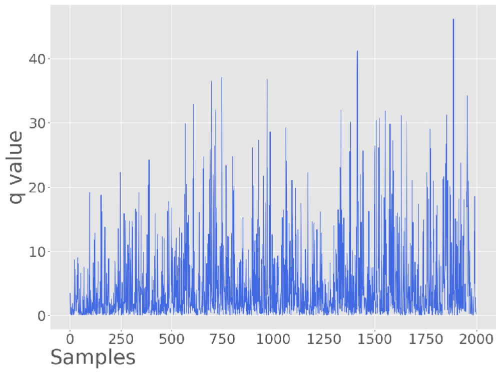
Another two graphs depict also the same squared difference but this time also on a logarithmical scale, which allows for smoother visualization. The last third graph depicts the activity of the q parameter during the filtration as shown in equation 3.24. This time is depicted only a cutout window of 2000 samples for enabling a closer look at Q-LMS functioning.



Error in logarithmic scale



Squared error



q parameter activity

Figure 6.5: Closer look on Q-LMS filtration. Cutout window of 2000 samples.

Table 6.1 Shows the values of RMSE after filtration. These values again represent the difference between the base signal and the filter.

LMS	QLMS	RLS
0.2934	0.3323	0.0195

Table 6.1: Table depicts RMSE values after filtration, which shows the similarity between the filtration result and the actual noiseless signal. A smaller RMSE value means that signals are more similar and thus filtration was more effective.

Discussion

Based on the assumption that the Q-LMS is LMS variant that is looking to improve on the algorithm, better performance was expected in comparison to LMS. However, in this configuration, the results were very close to LMS with a tendency for weaker performance. From the graphs, in 6.5 we can see that in error peaks q parameter works as expected and

QLMS is able to outperform LMS. However, when the filtration reaches a steady state, QLMS seems to not be able to stabilize. This causes spikes in learning rate value which then introduces unnecessary errors.

RLS filtration performs exceptionally well as expected on the bases of information in section 3.3.

With all that said all after the examination of visualized results and audio recordings before and after filtering, the conclusion is that all algorithms successfully removed the noise from the speech data and significantly improved the quality of the audio recording.

6.2 Filtering of EEG artifacts in cascade

In the second experiment, we will examine the effectiveness of different algorithms on the EOG ECG and EMG artifact removal from EEG data. As discussed in earlier section 2.2, EEG artifacts are sources of complex noise and their removal is a difficult task. Every type of artifact is different and their response to filtration may vary from method to method, one of the aims of the experiment is to see if adaptive filtering methods respond strongly to different artifact types or are some methods just better all around than others. Another aim is to inspect the effectiveness of the QLMS algorithm explored in an earlier section on EEG artifact removal as this algorithm was not been used on EEG artifact removal before.

Methodology

For this experiment, four types of signals will be needed. Unfortunately, it is uncommon to find all of these in an average dataset therefore data from multiple datasets had to be used.

- Clear EEG channel - Downloaded from the dataset that had also contained results of artifact removal methods.
- EOG reference channel - Commonly found in EEG datasets usually collected by three electrodes placed on the face around the eyes.
- ECG reference channel - Also commonly found in EEG datasets. Collected by an electrode or other recording device placed on the chest near the heart.
- EMG reference channel - Not commonly found in EEG datasets. Collected by electrodes placed on the muscles.

In this experiment, EEG artifacts are derived from reference signals.

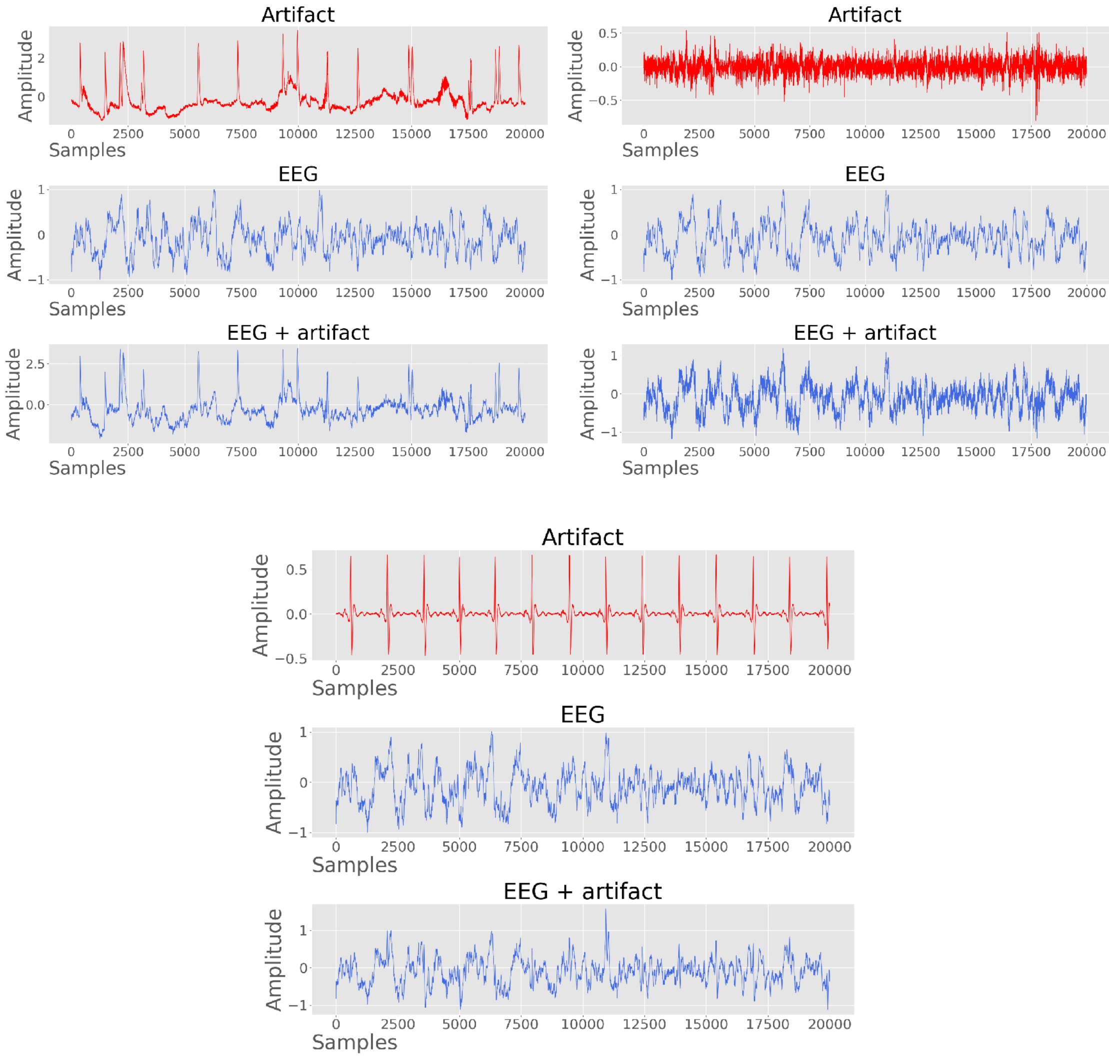


Figure 6.6: Graphs are showing impacts of individual artifacts on clear EEG signal

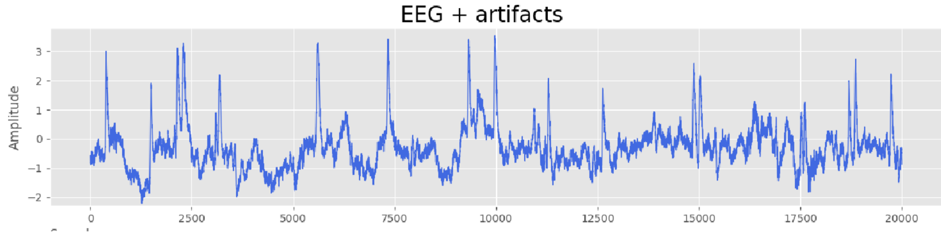
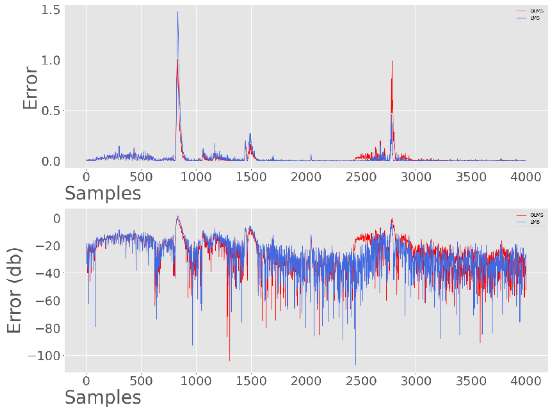


Figure 6.7

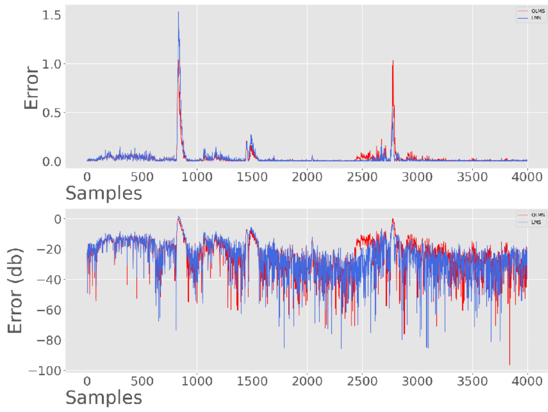
Figure 6.8: Graphs show the mean square error after filtration using three different adaptive algorithms each compared to the LMS algorithm with blue error.

Results

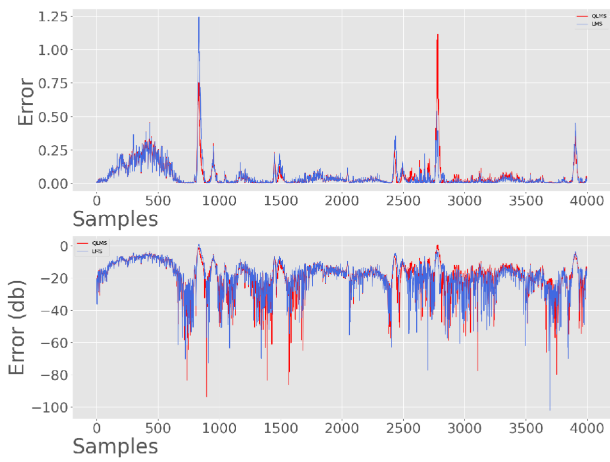
These first graphs are showing the squared error and squared error in logarithmic scale for each cascade step during Q-LMS filtration. In the first step, the EOG artifact is filtered, in the second EMG and in the last ECG. This squared error is compared to the squared error of LMS algorithm.



Step 1: EOG filtration



Step 2: EMG filtration



Step 3: ECG filtration

Figure 6.9: Graphs showing the squared error with squared error in logarithmic scale during every cascade step of Q-LMS filtration

Here we can see the resulting filtered EEG. when we make a comparison with figure 6.13 we can clearly see that many artifacts were successfully filtered.

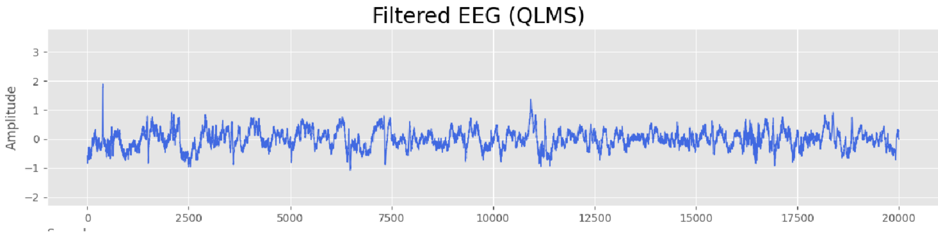
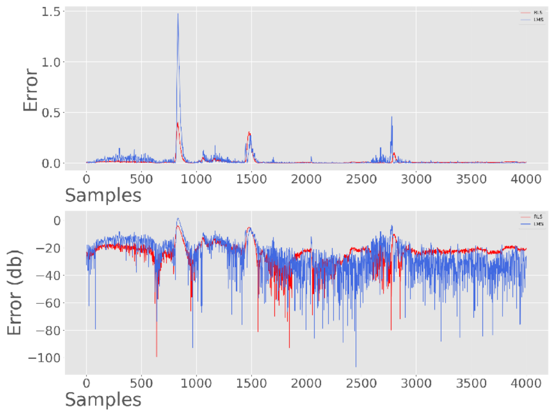


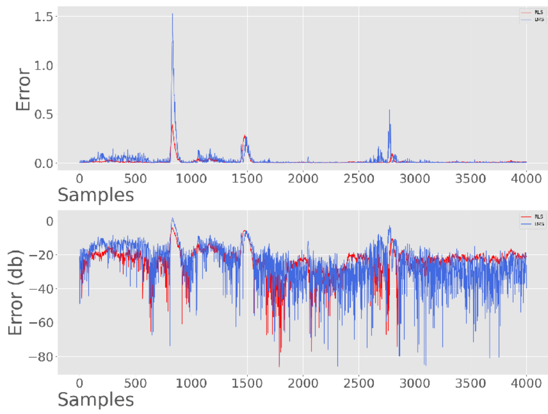
Figure 6.10

Figure 6.11: EEG signal after removal of artifacts using Q-LMS adaptive filter

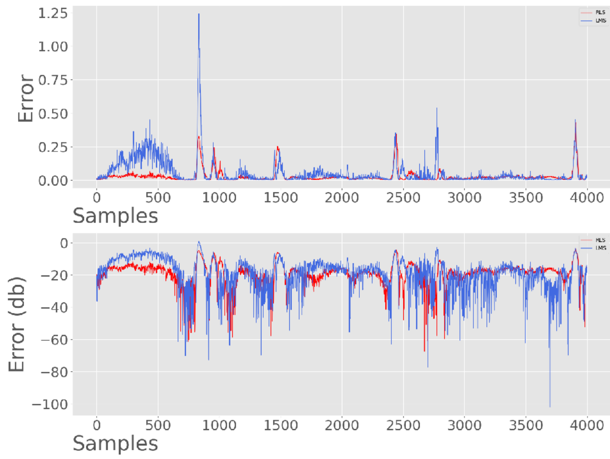
In this figure, we can see the squared errors depicted in the same way as in 6.9 but during RLS filtration. At first glance, we can see that in comparison to Q-LMS, the error stays in considerably lower values.



Step 1: EOG filtration



Step 2: EMG filtration



Step 3: ECG filtration

Figure 6.12: Graphs are showing impacts of individual artifacts on clear EEG signal

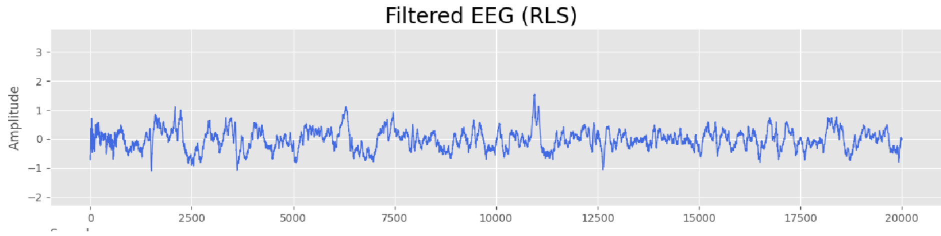


Figure 6.13

Figure 6.14: EEG signal after removal of artifacts using RLS adaptive filter

Table 6.2 is showing RMSE values of each algorithm for each cascade step. The final third step should be taken as the most significant value representing the final filtration results. The other values should be looked at carefully if trying to determine the quality of certain artifact filtration, as in cascade, the error from previous filtration will be propagated to current filtration.

Stage	Artifact type	LMS	QLMS	RLS
1	EOG	0.1654	0.1564	0.1058
2	EMG	0.0563	0.0620	0.0431
3	ECG	0.1832	0.1800	0.1043

Table 6.2: Table depicting RMSE between cascade stage filtration result and EEG free of the artifacts that were targeted by current and preceding stages.

Discussion

We can read a similar pattern from the results, as in the first experiment 6.2 where the q parameter works as expected in some parts of the filtering process, increasing the learning rate in order to return to a steady state when the error is high. In other parts, the q parameter seems to unexpectedly rise only to increase the error while the LMS with a static learning rate retains good performance. This time the results from table 6.2 seem to indicate that this time Q-LMS outperformed LMS although only by a little margin. The results of filtering an EMG artifact suggest that for this couple of reference signal and artifact LMS still outperformed Q-LMS. This may show that the LMS works better for cases when the reference and noise have a high correlation as this artifact was least modified while built from the reference signal. This hypothesis can be supported also by the first experiment 6.2 where the noise was also very similar to the reference and the results were in favor of LMS.

Once again the RLS shows remarkably better results although this time not as dominantly as in the first experiment 6.2.

Results show that the multiple types of artifacts were successfully removed from EEG signals using cascade filtering.

Chapter 7

Conclusion

The primary goal of this thesis was to propose and implement a framework with a toolkit capable of adaptive filtration on EEG data. The secondary goal was to explore the capabilities of a novel Q-LMS algorithm for filtering of EEG artifacts as this algorithm was not yet used for this type o scenario. Both goals were successfully fulfilled and the functions of the framework were shown in experiments with Q-LMS.

In the first part of the thesis EEG artifacts were carefully studied together with standard methods of their removal. Internal processes of these methods were discussed as well as their possible limitations in real-world applications.

In the next part, a close examination of the adaptive filtering method was performed. The general adaptive filter has been introduced together with specific adaptive algorithms.

Based on the studied material the framework for adaptive filtering was proposed and implemented in a Python environment.

The other existing frameworks that are covering similar problems were put forward and a discussion about their similarities and differences to the proposed framework was held. The results of adaptive filtration of the Padasip framework were compared to the results of the implemented framework in order to gain validation for LMS and RLS algorithms.

After that, the experiments aim to examine the capabilities of the Q-LMS adaptive algorithm for artifact removal from EEG. In experiments, the filtration was done using the proposed framework. These experiments showed that the Q-LMS algorithm has moderate results for EEG artifact scenario as opposed to the RLS method which showed excellent performance.

During the work on the thesis, I had to go through a large number of scientific publications and that helped me to improve my own scientific writing and my reading comprehension of written scientific material. This work also gave me a good introduction to handling brain signals and the interpretation of experiments tested my newly obtained knowledge.

For future work, I would suggest implementing the ability of real-time filtering of the data. Such an approach would need to deal with interesting problems like managing delays and communicating between processes conducting the filtering at least in the case of filters in cascade. Also, the possibility to connect the software to real-world EEG recording devices could be investigated together with new possibilities for the visualization of results. Another area with a place for more studies would be the problem of obtaining a reference signal. Here the methods for the generation of custom reference signals could be examined and implemented into the framework.

Bibliography

[1] Arif, M., Naseem, I., Moinuddin, M., Khan, S. S. and Ammar, M. M. Adaptive noise cancellation using q-LMS. In: *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*. April 2017, p. 1–4 [cit. 26.4.2023]. DOI: 10.1109/ICIEECT.2017.7916527.

[2] Beniczky, S. and Schomer, D. L. Electroencephalography: basic biophysical and technological aspects important for clinical applications. *Epileptic Disorders* [online]. Arcueil (France): John Libbey Eurotext. december 2020, vol. 22, no. 6, p. 697–715. DOI: 10.1684/epd.2020.1217. ISSN 1294-9361. Available at: https://pubmed.ncbi.nlm.nih.gov/33270023/.

[3] Bono, V., Das, S., Jamal, W. and Maharatna, K. Hybrid wavelet and EMD/ICA approach for artifact suppression in pervasive EEG. *Journal of Neuroscience Methods* [online]. Amsterdam (Netherlands): Elsevier. july 2016, vol. 267, p. 89–107, [cit. 2023-04-11]. DOI: https://doi.org/10.1016/j.jneumeth.2016.04.006. ISSN 0165-0270. Available at: https://www.sciencedirect.com/science/article/pii/S0165027016300437.

[4] C., M. *Padasip.filters.rls*. 2016 [cit. 9.5.2023]. Available at: https://matousc89.github.io/padasip/sources/filters/rls.html.

[5] Chai, T. and Draxler, R. R. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development* [online]. Göttingen (Germany): Copernicus Publications on behalf of the European Geosciences Union. june 2014, vol. 7, no. 3, p. 1247–1250, 27 May 2014, [cit. 26.4.2023]. DOI: https://doi.org/10.5194/gmd-7-1247-2014. License: Creative Commons Attribution 3.0 License. Available at: https://gmd.copernicus.org/articles/7/1247/2014/.

[6] Correa, A. G., Laciar, E., Patiño, H. D. and Valentinuzzi, M. E. Artifact removal from EEG signals using adaptive filters in cascade. *Journal of Physics: Conference Series* [online]. Bristol (United Kingdom): IOP Publishing. nov 2007, vol. 90, no. 1, p. 012081. DOI: 10.1088/1742-6596/90/1/012081. ISSN 17426588. Available at: https://dx.doi.org/10.1088/1742-6596/90/1/012081.

[7] De Vos, M., Riès, S., Vanderperren, K., Vanrumste, B., Alario, F.-X. et al. Removal of muscle artifacts from EEG recordings of spoken language production. *Neuroinformatics* [online]. Berlin (Germany): Spinger. june 2010, vol. 8, no. 2, p. 135–150. DOI: 10.1007/s12021-010-9071-0. Available at: https://doi.org/10.1007/s12021-010-9071-0.

[8] Farrens, J. L., Simmons, A. M., Luck, S. J. and Kappenman, E. S. Electroencephalogram (EEG) Recording Protocol for Cognitive and Affective Human Neuroscience Research. june 2020. DOI: 10.21203/rs.2.18328. Creative Commons Attribution 4.0 International License: https://creativecommons.org/licenses/by/4.0/. Available at: https://assets.researchsquare.com/files/pex-779/v3/53604cc3-6b01-4c5b-a7ae-82dbc9e0b050.pdf?c=1631843061.

[9] Ferrara, E. and Widrow, B. Multichannel adaptive filtering for signal enhancement. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. Kuala Lumpur (Malaysia): IEEE. june 1981, vol. 29, no. 3, p. 766–770. DOI: 10.1109/TASSP.1981.1163589. ISSN 0096-3518.

[10] Gebodh, N., Esmaeilpour, Z., Datta, A. and Bikson, M. *„Dataset of Concurrent EEG, ECG, and Behavior with Multiple Doses of transcranial Electrical Stimulation - BIDS“*. OpenNeuro, 29. may 2021 [cit. 29.4.2023]. DOI: 10.18112/openneuro.ds003670.v1.1.0. License CC0. Available at: https://openneuro.org/datasets/ds003670/versions/1.1.0.

[11] Goncharova, I., McFarland, D., Vaughan, T. and Wolpaw, J. EMG contamination of EEG: spectral and topographical characteristics. *Clinical Neurophysiology* [online]. 2003, vol. 114, no. 9, p. 1580–1593, [cit. 28.4.2023]. DOI: https://doi.org/10.1016/S1388-2457(03)00093-2. ISSN 1388-2457. Available at: https://www.sciencedirect.com/science/article/pii/S1388245703000932.

[12] Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D. et al. MEG and EEG Data Analysis with MNE-Python. *Frontiers in Neuroscience* [online]. Lausanne (Switzerland): Frontiers Media SA. december 2013, vol. 7, no. 267, p. 1–13. DOI: 10.3389/fnins.2013.00267.

[13] Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P. et al. Array programming with NumPy. *Nature*. Springer Science and Business Media LLC. september 2020, vol. 585, no. 7825, p. 357–362. DOI: 10.1038/s41586-020-2649-2. Available at: https://doi.org/10.1038/s41586-020-2649-2.

[14] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. IEEE COMPUTER SOC. 2007, vol. 9, no. 3, p. 90–95. DOI: 10.1109/MCSE.2007.55.

[15] Höchenberger, R., Larson, E., Rockhill, A., Gramfort, A., Vliet, M. van et al. *Openneuro-py* [https://github.com/hoechenberger/openneuro-py]. v2022.4.0. GitHub, 2021. Version 0.3.6, Licensed under GPL.

[16] jcavanagh@unm.edu, J. F. C. *„EEG: Probabilistic Selection and Depression“*. OpenNeuro, 15. january 2021 [cit. 9.5.2023]. DOI: 10.18112/openneuro.ds003474.v1.1.0. License: CC0. Available at: https://openneuro.org/datasets/ds003474/versions/1.1.0.

[17] Jiang, X., Bian, G.-B. and Tian, Z. Removal of Artifacts from EEG Signals: A Review. *Sensors* [online]. MDPI. february 2019, vol. 19, no. 5, p. 6, [cit. 2023-04-11].

DOI: 10.3390/s19050987. ISSN 1424-8220. Available at: http://www.mdpi.com/1424-8220/19/5/987.

[18] Krekel, H., Oliveira, B., Pfannschmidt, R., Bruynooghe, F., Laugher, B. et al. *Pytest:7.2.2*. 2004. Available at: https://github.com/pytest-dev/pytest.

[19] Lau, T. M., Gwin, J. T. and Ferris, D. P. How Many Electrodes Are Really Needed for EEG-Based Mobile Brain Imaging? *Journal of Behavioral and Brain Science*. 2012, vol. 02, no. 03, p. 387–393. DOI: 10.4236/jbbs.2012.23044. ISSN 2160-5866. Available at: http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jbbs.2012.23044.

[20] Lee, G. R., Gommers, R., Waselewski, F., Wohlfahrt, K. and O’Leary, A. PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*. The Open Journal. 2019, vol. 4, no. 36, p. 1237. DOI: 10.21105/joss.01237. Available at: https://doi.org/10.21105/joss.01237.

[21] Miltiadous, A., Tzimourta, K. D., Afrantou, T., Ioannidis, P., Grigoriadis, N. et al. *„A dataset of 88 EEG recordings from: Alzheimer's disease, Frontotemporal dementia and Healthy subjects"*. OpenNeuro, 17. february 2023 [cit. 29.4.2023]. DOI: doi:10.18112/openneuro.ds004504.v1.0.2. License: CC0. Available at: https://openneuro.org/datasets/ds004504/versions/1.0.4.

[22] Moeller, J. *EEG Basics* [online]. August 2014 [cit. 28.4.2023]. Accessed: 28.4.2023. Available at: https://www.youtube.com/playlist?list=PLxaiR6teSdjoEZWaDWm28A9QjFN7eguAp.

[23] Mustafa, R., Ali, M. A. M., Umat, C. and Al Asady, D. Design and implementation of least mean square adaptive filter on Altera Cyclone II Field Programmable Gate Array for active noise control. In: Taib, M. N. and Hamzah, M. K., ed. *2009 IEEE Symposium on Industrial Electronics & Applications*. 1st ed. Kuala Lumpur (Malaysia): IEEE, October 2009, vol. 1, p. 479–484 [cit. 2023-04-11]. DOI: 10.1109/ISIEA.2009.5356420. ISBN 978-1-4244-4681-0. Available at: http://ieeexplore.ieee.org/document/5356420/.

[24] &NA;. Guideline Thirteen. *Journal of Clinical Neurophysiology*. 1994, vol. 11, no. 1, p. 111–113. DOI: 10.1097/00004691-199401000-00014. ISSN 0736-0258. Available at: http://journals.lww.com/00004691-199401000-00014.

[25] Narasimhan, S. and Dutt, D. Application of LMS adaptive predictive filtering for muscle artifact (noise) cancellation from EEG signals. *Computers & Electrical Engineering* [online]. Amsterdam (Netherlands): Elsevier. january 1996, vol. 22, no. 1, p. 13–30, [cit. 2023-04-11]. DOI: https://doi.org/10.1016/0045-7906(95)00030-5. ISSN 0045-7906. Available at: https://www.sciencedirect.com/science/article/pii/0045790695000305.

[26] Nidal, K. and Malik, A. S. *EEG/ERP analysis: methods and applications*. 1st ed. Boca Raton: Crc Press, december 2014.

[27] Peterson, S. and Ferris, D. OpenNeuro. 11. [cit. 29.4.2023]. DOI: doi:10.18112/openneuro.ds003739.v1.0.3. Lisense: CC0. Available at: https://openneuro.org/datasets/ds003739/versions/1.0.3.

[28] Plöchl, M., Ossandón, J. P. and König, P. Combining EEG and eye tracking: identification, characterization, and correction of eye movement artifacts in electroencephalographic data. *Frontiers in Human Neuroscience*. 2012, vol. 6. DOI: 10.3389/fnhum.2012.00278. ISSN 1662-5161. Available at: http://journal.frontiersin.org/article/10.3389/fnhum.2012.00278/abstract.

[29] Queiroz, C. M. M., Silva, G. M. da, Walter, S., Peres, L. B., Luiz, L. M. D. et al. Single channel approach for filtering electroencephalographic signals strongly contaminated with facial electromyography. *Frontiers in Computational Neuroscience* [online]. Lausanne (Switzerland): Frontiers Media SA. july 2022, vol. 16, p. 5. DOI: 10.3389/fncom.2022.822987. ISSN 1662-5188. Available at: https://www.frontiersin.org/articles/10.3389/fncom.2022.822987.

[30] Quinn, A. J., Santos, V. Lopes-dos, Dupret, D., Nobre, A. C. and Woolrich, M. W. EMD: Empirical Mode Decomposition and Hilbert-Huang Spectral Analyses in Python. *Journal of Open Source Software*. The Open Journal. 2021, vol. 6, no. 59, p. 2977. DOI: 10.21105/joss.02977. License: GPL. Available at: https://doi.org/10.21105/joss.02977.

[31] Ribeiro, M. J. and Castelo Branco, M. *„EEG, ECG and pupil data from young and older adults: rest and auditory cued reaction time tasks"*. OpenNeuro, 10. june 2021 [cit. 29.4.2023]. DOI: 10.18112/openneuro.ds003690.v1.0.0. License: CC0. Available at: https://openneuro.org/datasets/ds003690/versions/1.0.0.

[32] Sazgar, M. and Young, M. G. EEG Artifacts. In: *Absolute Epilepsy and EEG Rotation Review: Essentials for Trainees*. Cham (Switzerland): Springer International Publishing, 2019, p. 149–162 [cit. 28.4.2023]. DOI: 10.1007/978-3-030-03511-2_8. ISBN 978-3-030-03511-2. Available at: https://doi.org/10.1007/978-3-030-03511-2_8.

[33] Thomas, K. P., Guan, C., Tong, L. C. and Prasad, V. A. An adaptive filter bank for motor imagery based Brain Computer Interface. In: *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2008, p. 1104–1107. DOI: 10.1109/IEMBS.2008.4649353.

[34] Took, C. C. and Mandic, D. P. The Quaternion LMS Algorithm for Adaptive Filtering of Hypercomplex Processes. *IEEE Transactions on Signal Processing*. Kuala Lumpur (Malaysia): IEEE. april 2009, vol. 57, no. 4, p. 1316–1327. DOI: 10.1109/TSP.2008.2010600. ISSN 1941-0476.

[35] Černý, M., Pikálek, T. and Mareček, J. *Padasip: Python Adaptive Signal Processing* [https://github.com/matousc89/padasip]. GitHub, 2021. Version 0.3.6, Licensed under the MIT License.

[36] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020, vol. 17, p. 261–272. DOI: 10.1038/s41592-019-0686-2.

[37] Westover, M. B., Gururangan, K., Markert, M., Blond, B., Lai, S. et al. Diagnostic Value of Electroencephalography with Ten Electrodes in Critically Ill Patients. *Neurocritical Care* [online]. Lausanne (Switzerland): Frontiers Media SA.

february 2020, vol. 33, p. 1–13. DOI: 10.1007/s12028-019-00911-4. License: CC BY 4.0.

[38] Zubair, S., Chaudhary, N. I., Khan, Z. A. and Wang, W. Momentum fractional LMS for power signal parameter estimation. *Signal Processing* [online]. Amsterdam (Netherlands): Elsevier. january 2018, vol. 142, p. 441–449. DOI: https://doi.org/10.1016/j.sigpro.2017.08.009. ISSN 0165-1684. Available at: https://www.sciencedirect.com/science/article/pii/S0165168417302918.

[39] İnce, R., Adanir, S. S. and Sevmez, F. The inventor of electroencephalography (EEG): Hans Berger (1873–1941). *Child's Nervous System* [online]. Berlin (Germany): Spinger. september 2021, vol. 37, no. 9, p. 2723–2724, [cit. 28.4.2023]. DOI: 10.1007/s00381-020-04564-z. ISSN 1433-0350. Available at: https://doi.org/10.1007/s00381-020-04564-z.