# Multi-sensor signal separation, localization and classification by exploring sparsity and smoothness

Research Thesis

Submitted in Partial Fulfillment of The
Requirements for the Degree of
Master of Science in Electrical Engineering

Dmitri Model

Submitted to the Senate
The Technion—Israel Institute of Technology

Ya'ar, 5765 Haifa June, 2005

# Acknowledgements

The Research Thesis Was Done Under

The Supervision of Dr.Michael Zibulevsky

in the Faculty of Electrical Engineering.

THE GENEROUS FINANCIAL HELP OF TECHNION IS GRATEFULLY ACKNOWLEDGED.

I would like to express my sincere gratitude to my supervisor, Dr. Michael Zibulevsky, for his excellent guidance, patience and continued support throughout all the stages of this research.

# Contents

# List of Figures

# List of Tables

# Abstract

This thesis consists of two contributions in the field of multi-channel signal processing. The first topic of this thesis is a developing of multiple-source reconstruction and localization technique for wide-band signals in sensor arrays. The proposed approach may handle both straight-path-only and reverberant model of signal propagation, based on the knowledge of impulse responses for all location-sensor pairs. We assume that source signals are spatially sparse, as well as have sparse (say, wavelet-type) representation in time domain. This prior is expressed via a large scale convex optimization problem, which involves $l_1$ and non-squared $l_2$ norm minimization. The optimization is carried out by the Truncated Newton method, using preconditioned Conjugate Gradients in inner iterations. Presented numerical experiments demonstrate an advantage of our approach comparing to existing source localization methods in such aspects as super-resolution, robustness to noise and very limited data size. Our approach doesn't require accurate initialization, and also can recover correlated sources.

The second contribution of this thesis is the developing of Electro-Encephalography (EEG) signals analysis methods. This is done in the framework of

Brain-Computer interface (BCI), in which our goal is to distinguish between two mental tasks a person is concentrating on. In order to improve the classification performance, we use two stage preprocessing of multi-sensor data. At first, we perform *spatial* filtering, by taking weighted linear combination of sensors. At the second step, we perform *time-domain* filtering. Filter coefficients are learned from the data by maximizing the between class discrimination and minimizing the total variation of result average or, alternatively, suppressing the signal at the windows, where it is known to be absent. No other information on signals of interest is assumed to be available. This leads to a constrained optimization problem, which involves $l_1$-$l_2$ norm minimization. We also suggest solution using eigen-value decomposition approach. We evaluate our method both on synthetic and real EEG data, and demonstrate the advantage of our approach comparing to the existing methods.

# List of symbols

## Chapter 2

| | |
|---|---|
| $*$ | convolution. |
| $\mathcal{A}$ | forward operator. |
| $\mathcal{A}^*$ | adjoint operator. |
| BSS | Blind Source Separation. |
| $C$ | matrix containing signal coefficients in its rows. |
| $D_{km}$ | the delay of the $k$-th source toward the $m$-th sensor. |
| DOA | Direction of Arrival. |
| $G$ | gradient of objective function. |
| $\mathcal{H}$ | Hessian of objective function. |
| $h_{km}$ | the transfer function from the $k$-th source toward the $m$-th sensor. |
| ICA | Independent Component Analysis. |
| $K$ | number of sources. |
| $L$ | number of possible locations. |
| $l_1$ | first norm. |
| $M$ | number of sensors. |
| $N$ | noise matrix. |
| $n_i(t)$ | noise registered by the $i$-th sensor. |
| $S$ | matrix containing signals from all grid nodes in its rows. |
| $s_k(t)$ | $k$-th source signal. |
| TDOA | Time Delay Of Arrival. |

| | |
|---|---|
| $v$ | propagation speed. |
| $\|X\|_F$ | Frobenius norm of matrix $X$. |
| $Y$ | sensors' measurement matrix. |
| $y_i(t)$ | $i$-th sensor output. |
| $\alpha$ | slowness vector. |
| $\delta(t)$ | Dirac delta function. |
| $\mu_1, \mu_2$ | trade-off parameters. |
| $\rho(t)$ | interpolation kernel. |
| $\Phi$ | matrix containing the elements of basis in its rows. |
| $\psi(x)$ | smooth approximation of the second norm. |

## Chapter 3

| | |
|---|---|
| $a$ | mixing vector. |
| BCI | Brain-Computer Interface. |
| CSP | Common Spatial Patterns. |
| CSSP | Cross Spatial-Spectral Patterns. |
| CV | Cross Validation. |
| EEG | Electro-Encephalography. |
| EMG | Electro-myography. |
| EVD | Eigenvalue Decomposition. |
| $h[n]$ | time-domain filter. |
| k-NN | k Nearest Neighbors classifier. |
| $L$ | number of trials which belong to the first class. |

| | |
|---|---|
| $M$ | number of trials which belong to the second class. |
| NM | Nearest Mean classifier. |
| $S$ | number of sensors. |
| $s.t.$ | subject to. |
| $\hat{s}^j_{avg}$ | over-trials averaged estimated signal. |
| $\hat{s}^j_i$ | single-trial estimated signal. |
| SVM | Support Vector Machines classifier. |
| $T$ | number of time samples. |
| TV | Total Variation. |
| $w$ | spatial integration vector. |
| $X^j_{avg}$ | over-trials averaged matrix. |
| $X^j_i$ | *single-trial matrix.* |
| $Y$ | matrix of differences (used for calculation of TV). |

# Chapter 1

# Introduction

In this thesis we address the problem of improving multi-channel data processing, exploring prior knowledge of some specific properties of signals, such as sparsity or smoothness. We present two algorithms. The first is in the field of source localization and reconstruction with sensor array. The second is in a field of single-trial Electro-Encephalography (EEG) signals classification. The purpose of this chapter is to introduce the problems addressed in this thesis, motivate the need for a new approach, and describe the main contribution and the organization of the thesis.

## 1.1 Overview of the Problems Addressed in this Thesis

Two problems addressed in this thesis have completely different application: the problem of source localization and reconstruction using sensor

arrays is of great importance in the fields of wireless communication [1], radar [2],sonar [3], and exploration seismology [4], while the problem of EEG signals classification by spatial and spectral filtering is of great interest for neuroscience community [5] and Brain-Computer Interface (BCI) systems [6]. However, in both cases the goal is to process the noisy multi-channel data in order to reconstruct single or multiple sources. As byproduct of source reconstruction, this may yield the source localization in the first case, and improvement of classification accuracy in the second. For the solution of both problems we adopt regularized numerical optimization approach, which ends up in similar problem formulations in both cases. Thus we find it appropriate to include both algorithms in the same thesis.

## Sparse Multi-Channel Signal Reconstruction and Localization

Among the desired properties of source reconstruction method are the robustness to the measurement and ambient noise, limited amount of data samples and the ability to distinguish closely located sources. We suggest the technique which improves these properties for the special case, when the sources can be sparsely represented in time domain in some know basis or frame.

Many different approaches address the problem of detecting multiple wide-band sources and estimating their angle of arrival (locations), based on the signals received by a sensor array. The maximum-likelihood estimation [7] is potentially the most precise technique; it assumes however that the

number of sources and the source spectral density matrix are known. Moreover, the likelihood function is generally non-convex, and may have spurious local solutions. Another approach for multiple source localization combines a special ARMA parameter estimation method with a non-linear optimization procedure to estimate the relative time delays [8]. However, this approach cannot effectively treat correlated sources and requires prior knowledge of the number of sources.

The signal-subspace processing approach, was first proposed for the narrow-band case [9]. Under the condition that the observation period is long and signal-to-noise ratio is not too low, this approach has been shown to have substantially higher resolution in estimating the directions of arrival of the signals, than conventional beamformer [10], Capon's MLM [11], and autoregressive spectral estimators [12].

The concept of signal-subspace processing can also be used in the wide-band case. The technique given in [13] can be referred to as *incoherent* signal-subspace processing: the angle estimation is first done with each narrow-band component individually, followed by combination of these estimates for the final result. As is common in any detection and estimation system, at low signal-to-noise ratios the threshold effect prevents the final combination to be effective [14]. Another problem with the incoherent signal-subspace processing is its inability to handle completely correlated sources even if SNR is infinitely high and the observation time is infinitely long. Several techniques have been developed based on *coherent* signal-subspace processing [14]. They demonstrate better performance than corresponding incoherent techniques, but still require rather long observation time and high SNR ratio for for good

estimation of covariance matrices.

A conventional framework for source reconstruction is Blind Source Separation (BSS) using Independent Component Analysis (ICA) (see for example [15] for review). Conventional ICA of instantaneous mixtures exploits the non-Gaussianity of source signals to perform separation. Similar methods for ICA have been developed from a number of different view points: minimizing Kulback-Liebler divergence, Infomax or Maximum Likelihood estimation [16, 17, 18, 19]. Some methods look for the most non-Gaussian components, using kurtosis [20] or negentropy as non-Gaussian measures [21, 22]. Other approach estimates the unmixing matrix by performing approximate diagonalization of a cumulant tensor of the mixtures [23]. A more advanced methods (e.g. [24, 25]) exploit the idea of sparse source representation for a very efficient blind source separation.

However above methods assume instantaneous mixture model, i.e. assuming that simple multiplicative term may describe the propagation from source to sensor. This is applicable only for systems without propagation delays or narrow-band signals in frequency domain.

Moreover, the instantaneous BSS model leads to non-convex optimization problem. Hence the optimization procedure can converge to spurious local minima. This problem becomes especially challenging when a noise is incorporated explicitly into statistical model [26].

The convolutional ICA problem, on the contrary to instantaneous one, assumes convolutive model of signal propagation. Many methods have been proposed to solve it. Some of them suggested to work directly in the time-domain [27]. Working in the time domain has the disadvantage of being

rather computational hard, due to large number of variables. Other approaches suggested moving to the frequency domain in order to transform the convolution into multiplication and apply ICA methods for instantaneous mixtures for each frequency bin [28], [29]. However, there is an inherent permutation problem in all frequency-domain ICA methods, which doesn't exist in time-domain methods.

The method introduced in this thesis also assumes multipath (convolutive) model of signal propagation but differs from conventional ICA techniques, because it assumes that the forward (mixing) operator is known i.e. impulse responses for all location-sensor pairs are assumed to be known[1]. It uses the approach similar to [30] by assuming several point sources, which can be sparsely represented in space, by dividing the area of interest into a discrete grid of potential source locations.

Since the number of possible source locations is often much greater than the number of sensors, the corresponding inverse-problem is ill-posed. The technique described in [30] suggests to regularize the solution by enforcing spatial sparsity using non-squared $l_2$ norm regularization. Since we explore the case, when the signals can be also sparsely represented in time domain, we suggest to impose temporal sparsity in addition to spatial one. This can be done using the $l_1$ norm, which is commonly used to enforce sparsity [31, 24].

In order to solve the problems of source reconstruction and localization, we estimate the source matrix $S$, which contains estimated signals from all possible locations (not just actual sources) in it rows. By calculating the energy of signals in all location (rows of $S$), we receive the spatial spectra, i.e.

[1] which may be unpractical for some reverberation environment.

an estimate of source locations. The rows, which correspond to dominating energy peaks will actually contain the reconstructed sources.

We conduct numerical experiments in which we compare the performance of our technique with the source localization method [30]. This comparison demonstrates the advantages of enforcing temporal sparsity along with spatial sparsity, such as improved robustness to noise and to very limited data size.

## Learning Spatial and Temporal Filters for Single-Trial EEG Classification

The second contribution of this thesis addresses the problem of multi-sensor EEG signal classification, in order to facilitate Brain-Compute Interface (BCI). People have speculated that EEG might be used as alternative communication channel, which allows the brain to act bypassing peripheral nerves and muscles, since electroencephalography was first described by Hans Berger in 1929 [32]. First simple communication systems, that were driven by electrical activity recorded from the head, appeared about three decades ago [6]. In the past years, it has been shown that it is possible to recognize distinct mental precesses from online EEG (see, for example [33, 34, 35, 36]). By associating certain EEG patterns to simple commands, it is possible to control a computer, creating an alternative communication channel, which is usually called *Brain-Computer Interface* (BCI) [6, 37].

One of the most complicated problem of the BCI is classifying very noisy EEG signals, obtained by registering the brain activity of the subject. The

first approach suggests dealing with this problem by requiring extensive training, in order to teach the subject to acquire self-control over a certain EEG components, such as sensorimotor $\mu$-rhythm [37] or slow cortical potentials [38]. This ability to create certain EEG patterns *at will* is translated by BCI system to cursor movement [37, 39] or selection of letters or words on computer monitor [40, 38].

The second approach suggests developing *subject-specific* classifiers to recognize different cognitive processes from EEG signals [34, 35, 41]. In this case, the typical BCI procedure consists of two stages. First, the person trains the system by concentrating on predefined mental tasks. Usually two different tasks are used in the training. BCI registers several EEG samples of each task. Then, the training data is being processed in order to build a classifier. In the second stage, the subject concentrates on one of the tasks again, and the system *automatically* classifies the EEG signals. The key for successful classification is a good preprocessing of raw data. The objective of this chapter is to develop a preprocessing methods based on spatial and spectral filtering, which will improve the classification accuracy.

The use of spatial filtering in order to improve the classification is not a new discovery. The method introduced in [42] propose to treat each time sample individually, and find the spatial integration weight by logistic regression. However this approach does not take into account the time courses of EEG signal, making implicit assumption that the coupling vector between the source and sensors is constant over the time of the response. Moreover, this method tries to maximally discriminate between signals of two classes, using no regularization. Thus, the resulting spatial filter is very prone to

noise.

A more advanced method for learning spatial filters is CSP [43, 44]. It suggests to find several projections, that maximally discriminate the variances of projected data of two classes. The CSSP [45] method suggest an improvement to CSP, by incorporating a spectral filter in addition to spatial one. However, both approaches take no usage of time-course of signals, and as result discriminate some measure of sources, rather than sources themselves.

We propose yet another method for spatial and spectral filtering of multichannel EEG signals. The proposed approach is based on the assumption that the response to the mental task is temporary *smooth* (e.g. has limited total variation) and/or is expected to be small in certain time windows, where the task is not performed. Coefficients of both spatial and spectral filters are learned by optimizing the between class discrimination and the smoothness of result average. No other information on signals of interest is assumed to be available.

We have evaluated proposed method on several data sets. Our simulations show, that the proposed preprocessing significantly improves the classification rate, with respect to unprocessed data (simple sum of channels or choosing the best sensor) as well as data preprocessed by CSP [44] and CSSP [45] methods (which are currently considered the state-of-the-art in the field).

## 1.2 Outline and Contributions

The main contribution of this thesis is two algorithms. The first is a method for source localization and reconstruction, which is presented in Chapter 2. The second is a method for spatial and spectral filtering of multi-channel EEG signals. It is introduced in Chapter 3.

### Chapter 2

In this chapter we present the method which is intended to solve multiple source reconstruction and localization problems by exploring both temporal and spatial sparsity. We follow the approach of [30] by assuming several point sources, which can be sparsely represented in space, by dividing the area of interest into a discrete grid of potential source locations. In addition, we assume that impinging signals can be sparsely represented in an appropriate basis or frame [46, 47] (e.g., via the short time Fourier transform, Wavelet transform, Wavelet Packets, etc.). The combination of spatial and temporal sparsity assumptions leads to an improved performance, as demonstrated by our simulations.

### Chapter 3

In this chapter we propose several methods for spatial and spectral filtering of multi-channel EEG signals, which is intended to improve classification of EEG signals corresponding to different mental tasks. The proposed approach is based on the assumption that the response to the mental task is temporary *smooth* (i.e. has limited total variation) and/or is expected to be small

in certain time windows, where the task is not performed. Coefficients of both spatial and spectral filters are learned by optimizing the between class discrimination and the smoothness of result average. No other information on signals of interest is assumed to be available.

We have evaluated proposed method on several data sets. Our simulations shows, that the proposed preprocessing significantly improves the classification rate, with respect to unprocessed data (simple sum of channels or choosing the best sensor) as well as data preprocessed by CSP [44] and CSSP [45] methods (which are currently considered the state-of-the-art in the field).

We have also developed misclassification rate lower bound, which is applicable for the experiments with synthesized signals. This bound shows how well can we perform signal reconstruction (and further classification) based on spatial integration only. Our simulations shows, that in majority of cases we reach the bound or stay very close to it. If we use *time-domain* filtering in addition to spatial integration, then we perform even better.

# Chapter 2

# Sparse Multi-Channel Signal Reconstruction

In this chapter we present a method of multiple wideband source localization and reconstruction. We suggest the technique which improves the robustness to noise, limited data and the ability to distinguish closely located sources by modifying the algorithm proposed in [30] for the case, when the sources can be sparsely represented in time domain in some know basis or frame.

This chapter is organized as follows. We start with a short introduction to Sensor Array Processing in Section 2.1. Then we present the observation model in Section 2.2. The algorithm, which is the main contribution of this chapter, is introduced in Section 2.3. And, finally, we conduct computation experiments and make the conclusions in Section 2.4.

## 2.1 Overview of Sensor Array Processing

The usage of sensor arrays instead of single sensor provides numerous benefits, such as improvement in signal-to-noise ratio, possibility of electronic steering and jammer suppression among others. But more important, the task of source localization with single omnidirectional sensor is impossible. We can estimate the source locations only when several sensors are used. Another task, which becomes feasible when sensor array is used, is a reconstruction of multiple sources. The above tasks of source localization and reconstruction are in the main focus of this thesis. But, before we describe our method of Sensor Array processing, it is necessary to present the mathematical model for the problem. This is done in the current section.

### 2.1.1 Sources in the far-field of the array

We start with the most basic case, assuming sources in the far-field of a uniform linear array, which consists of M omnidirectional sensors, lined up along the $x$-coordinate axis, with equal spacing $L$, see Fig. 2.1.

For the sources in the farfield of the array, the curvature of the wavefront is insignificant across the aperture of the array, and the plane wavefront approximation holds. The propagation of the source signal $s_k(t)$ satisfies the equation $s_k(t-\mathbf{p}^T\alpha)$, where $\mathbf{p}$ is the position, and $\alpha$ is the so called slowness vector aligned with the direction of propagation of the wavefront. The vector $\alpha$ has a magnitude equal to the inverse of propagation speed, i.e. $\|\alpha\|_2 = 1/v$. The distance attenuation factor is not considered in the farfield model, since it will be almost constant across the array.

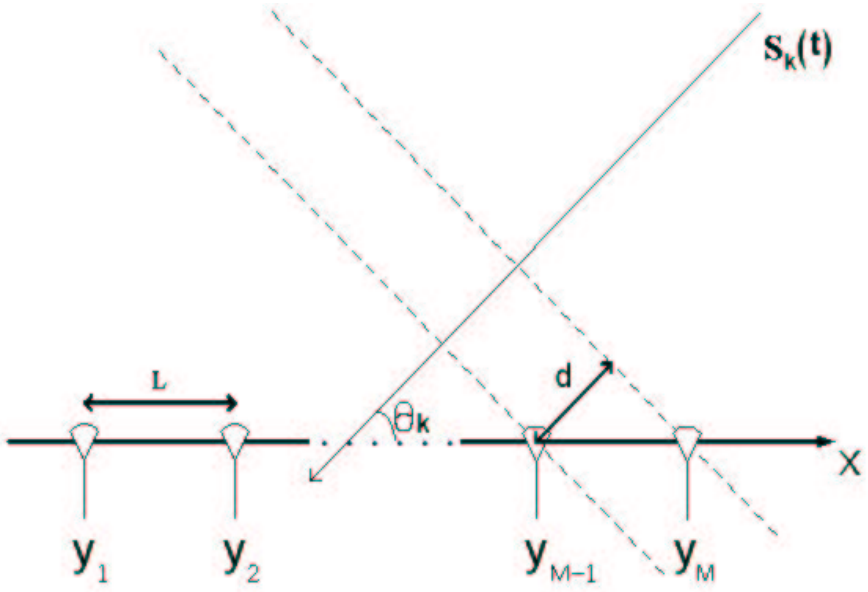



Figure 2.1: An illustration of uniform linear array. A source $s_k$(t) is impinging on the array at angle $\Theta_k$. The $i$-th sensor produces the output $y_i$

The wavefront $s_k(t)$ arrives from the direction that makes an angle of $\Theta_k$ with an $x$-axis (Fig. 2.1). When the wavefront reaches the $M$-th sensor, it has to travel the distance $d_k = L\cos\Theta_k$, before it appears at the next sensor. If no noise is present, and there is only one active source, then the output of the $(M-1)$-th sensor, $y_{M-1}$, is delayed[1] by $D_k = d_k/v$ with respect to $y_M$, the $M$-th sensor output , i.e. $y_{M-1}(t) = y_M(t-D)$. Furthermore, taking into consideration that the sensors are equally spaced, we can conclude that the first sensor output, $y_1$, will be delayed by $(M-1)D_k$ with respect to the last sensor output, i.e. $y_1(t) = y_M(t-(M-1)D_k)$. Note, that we are not interested in an absolute delay from a source to a sensor, but rather in a relative delay between different sensors.

[1] We are dealing with a general wideband signals, hence we are working with time delays and not with a phase shifts as in narrowband case.

It is more convenient to calculate all delays with respect to the first sensor, hence if we say a delay of the $m$-th sensor output, we mean a delay of the $m$-th sensor output with respect to the first sensor output. In order to formalize and summarize all of the above, we provide the equation for calculation of the delay, $D_{km}$, of $m$-th sensor output for the $k$-th source signal, which impinges on the array from the $\Theta_k$ direction:

$$D_{km} = -(m-1)\frac{L\cos\Theta_k}{v} \tag{2.1}$$

The above equation holds for the linear equally spaced sensor array, but it doesn't mean that our model is limited to this case only. Actually, the only thing required, is that the array geometry should be known. Once we know the sensor array structure, we are able to calculate the relative Time-Delays-Of-Arrival (TDOA).

The source localization task is not limited to a single source. Actually, it is assumed that several source signals $\{s_1(t), \ldots, s_K(t)\}$ imping on the array from the directions $\{\Theta_1, \ldots, \Theta_K\}$. Due to the linearity of the system the superposition principle holds, and the output of the $m$-th sensor can be expressed as:

$$y_m(t) = \sum_{k=1}^{K} \delta(t - D_{km}) * s_k(t) + n_m(t) \tag{2.2}$$

where $\delta(t)$ is a Dirac delta function, $*$ denotes convolution, and $n_m(t)$ is an additive noise - which is present in every practical system.

Note, that in the farfield case, the source location is characterized by the Direction-of-Arrival (DOA) only, which should be estimated by solving source localization problem.

### 2.1.2 Sources in the nearfield of the array

The generalization of the model to the case where the sources lie in the nearfield of the array has a number of applications, for example audio source localization using microphone array. In the nearfield case the plane-wave approximation no longer holds, and we use the spherical wave equation instead. At the distance $r$ from a single source $s_k(t)$ the wavefront is equal to $(1/r)s_k(t-r/v)$. If we denote $r_{km}$ the distance from the $k$-th source to the $m$-th sensor, then the $m$-th sensor output will be $y_m(t) = (1/r_{km})s_k(t - r_{km}/v)$. Since we are interested in the relative TDOA, the delay of the $k$-th source to $m$-th sensor will be $D_{km} = (r_{km}/v) - (r_{k1}/v)$. Now, the nearfield model will look pretty similar to the farfield model, except for the attenuation factor $1/r_{km}$. If $K$ signals $\{s_1(t), \ldots, s_K(t)\}$ impinge on the array, using superposition we receive the following equation for the output of the $m - th$ sensor:

$$y_m(t) = \sum_{k=1}^{K} (1/r_{km})\delta(t - D_{km}) * s_k(t) + n_m(t) \tag{2.3}$$

Note, that in the nearfield case, the source locations are characterized both by the DOA and by range[2], and thus our task is to reveal both of these parameters.

### 2.1.3 Reverberant Environment

Till now we supposed that the sources arrive to the sensors at the straight line and no reflections are present. However, this assumption is not always valid, for example for acoustic signals in closed environment. In the presence

[2]In polar coordinate system. In the cartesian coordinate system, the source location is characterized by $(x, y)$

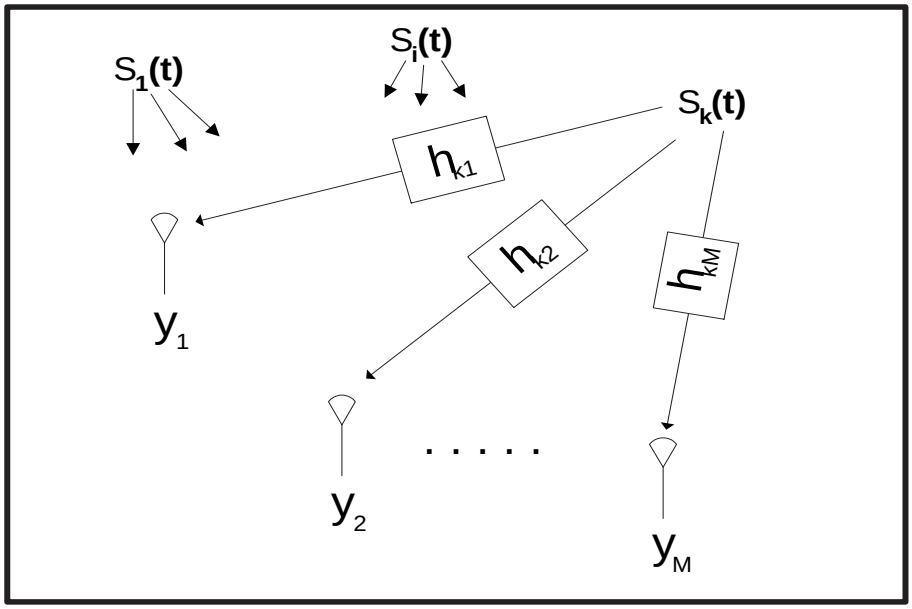



Figure 2.2: An illustration of reverberant environment. The source signal $s_k$ is captured by the $i$-th sensor as a convolution of the source signal $s_k$ with thr FIR filter $h_{ki}$

of multi-path, the propagation of the $k$-th source to the $m$-th sensor can no longer be described by the delay $D_{km}$ and the attenuation $1/r_{km}$ only. Rather, the signal captured by the sensor can be well represented by a convolution of the source signal with the FIR filter, modelling the transfer function between the source and the sensor (see for example [48])(Fig. 2.2).

Let us denote $h_{km}(t)$ the transfer function between the $k$-th source and the $m$-th sensor. Then, the $k$-th source will appear by the $m$-th sensor as $h_{km}(t) * s_k(t)$. We can now rewrite the equations (2.2) and (2.3) as:

$$y_m(t) = \sum_{k=1}^{K} h_{km}(t) * s_k(t) + n_m(t) \tag{2.4}$$

Note, that the above equation fits both the farfield and the nearfield models, if the transfer functions $h_{km}(t)$ are appropriately chosen.

## 2.2 Problem Formulation

### 2.2.1 Observation Model

Assume $K$ discrete time signals $s_k[n]$ impinge on a sensor array, consisting of $M$ sensors. The multipath propagation of $k$-th source toward $m$-th sensor can be represented by a convolution with the transfer function $h_{km}[n]$ (see for example [48])(Fig. 2.2). Thus, we can describe the output of the $m$-th sensor, $y_m[n]$, as:

$$y_m[n] = \sum_{k=1}^{K} h_{km}[n] * s_k[n] + n_m[n] \tag{2.5}$$

where $*$ denotes convolution, and $n_m[n]$ is an additive noise registered by the sensor. The above equation is also valid for far-field direct path model if we set $h_{km}[n] = \delta(n - D_{km})$:

$$y_m[n] = \sum_{k=1}^{K} \delta(n - D_{km}) * s_k[n] + n_m[n] \tag{2.6}$$

Here, $D_{km}$ is a delay of the $k$-th source toward the $m$-th sensor. The delay $D_{km}$ is *relative* to the first sensor, i.e. $D_{k1} = 0 \quad \forall k$.

### 2.2.2 Discretized Spatial Model

In our approach, we assume that the properties of the environment (signal propagation model, sensors positions) are known. Given the geometry of the problem, we can divide the whole area of interest in some discrete set of potential locations. It could be a set of pixels/voxels in the near field case, or a grid of Directions-of-Arrival angles in the far field case. We will usually

have much more potential locations than active sources, and our task is to identify which locations do actually contain sources.

Denote the potential source signal at $l$-th location as $s_l[n],\ 1 \leqslant n \leqslant T$. Suppose, we have $L$ potential locations, and the signals are limited in time to $T$ samples. Let introduce a $L \times T$ matrix $S$, which contains in its rows source signals at all potential locations. The matrix $S$ is the unknown we wish to estimate. When the solution will be achieved, we expect only few rows of $S$, corresponding to the active sources, to be significantly large. The energies of signals at each location will serve us as the spatial spectra estimation.

Assume we have $M$ sensors in our array. We introduce the *sensors measurement* matrix $Y$, which contains in the $m$-th row the output signal $y_m[n]$ of the $m$-th sensor.

Since we know the positions of sensors and the wave propagation model, we can pre-calculate the transfer functions $h_{lm}[n]$ from any grid node $l$ to any sensor $m$.

Let us define the *Forward Operator* $\mathcal{A}$ by its *action* $U = \mathcal{A}S$ on an arbitrary matrix $S$ :

$$u_m[n] = \sum_{l=1}^{L} h_{lm}[n] * s_l[n] \tag{2.7}$$

where $u_m[n]$ is the $m$-th row of $U$. Note, that we treat all locations (rows of $S$) equally, as if they all contain an active source. This representation of $\mathcal{A}$ is sufficient for our computations, and we do not need its explicit matrix form.

Our problem is to find $S$ given the observation

$$Y = \mathcal{A}S + N \tag{2.8}$$

where $N$ is an additive noise matrix, $m$-th row of which contains the noise registered by $m$-th sensor. Despite the operator $\mathcal{A}$ is known, the problem cannot be solved without additional priors, because we have much more grid locations than sensors.

The adjoint operator $\mathcal{A}^*$, which will be needed for the gradient computation, is given by its action $X = \mathcal{A}^*Y$: the $i$-th row of $X$ is

$$x_i[n] = \sum_{j=1}^{m} (h_{ji}[-k] * y_j[k])[n] \tag{2.9}$$

Note the "minus" sign near the argument of $h_{ji}$.

### 2.2.3 Interpolation

As mentioned above, we work with the discrete-time signals. Therefore, a problem arises when $D_{ji}$ is not integer. A straightforward solution is to replace the fractional delays with the rounded ones. However, this approach significantly limits the spatial resolution. A better approach suggests upsampling of signals prior to applying the $\mathcal{A}$ operator. The upsampling may be produced using some interpolation kernel.

Let $\mathcal{I}_{N_{up}}$ denote upsampling by factor $N_{up}$ operator, and if $S$ is an $L \times T$ matrix, then $S_{up} = \mathcal{I}_{N_{up}} S$ will be $L \times TN_{up}$ matrix. Note, that the $1 + N_{up}(i-1)$ -th column of $S_{up}$ is equal to the $i$-the column of $S$ $(1 \leq i \leq T)$. Other columns should be calculated using interpolation.

Suppose, we want to calculate the $j$-th column, $S^j_{up}$, of the matrix $S_{up}$. This column corresponds to the time point, laying between the samples $k = \lceil \frac{j}{N_{up}} \rceil$ and $k+1$ of the original signal ($\lceil\ \rceil$ is the ceiling operator). The distances between the above time point and the closest samples of original

signal are $d^- = (j-(k-1)N_{up}-1)/N_{up}$ to the left sample and $d^+ = 1-d^-$ to the right sample (measured in sampling periods $T_s$). Finally, if $\rho$ is the interpolation kernel, $N_{io}$ is an interpolation order and $S^k$ is the $k$-th column of $S$, then:

$$S_{up}^k = \sum_{l=-N_{io}+1}^{N_{io}} \rho(l-d^-)S^{k+l} \tag{2.10}$$

We also need to calculate the adjoint operator $\mathcal{I}_{N_{up}}^*$, which translates an $L \times TN_{up}$ matrix $S_{up}$ into $L \times T$ matrix $S_r = \mathcal{I}_{N_{up}}^* S_{up}$.

Using the above notations, we can write the following formula for the $S_r^k$ - the $k$-th column of matrix $S_r$:

$$S_r^k = \sum_{l=-N_{io}*N_{up}}^{N_{io}*N_{up}} \rho\left(\frac{l}{N_{up}}\right) S_{up}^{N_{up}(k-1)+l} \tag{2.11}$$

Now, in our model we will use the modified operators

$$\hat{\mathcal{A}} = \mathcal{A} \cdot \mathcal{I}_{N_{up}} \qquad \hat{\mathcal{A}}^* = \mathcal{I}_{N_{up}}^* \cdot \mathcal{A}^* \tag{2.12}$$

instead of $\mathcal{A}$ and $\mathcal{A}^*$, but for simplicity, we will continue to denote the modified operators as $\mathcal{A}$ and $\mathcal{A}^*$. Note, that after upsampling, we should adjust $D_{ji}$ to be $D_{ji}N_{up}$. We will still need to round $D_{ji}N_{up}$ to the closest integer, but now the rounding error is $N_{up}$ times less.

## 2.3 Algorithm Description

### 2.3.1 Sparse Regularization

We solve the problems of source separation and localization in the inverse problem framework: we want to find source matrix $S$, such that after applying to it the forward operator $\mathcal{A}$, the result will be as much close as possible to the actual sensors measurement matrix $Y$, i.e. $Y \approx \mathcal{A}S$. In another words, we want to find minimizer $\hat{S} = arg\min_S \|Y - \mathcal{A}S\|$. This measure of proximity is connected to maximum-likelihood model. The choice of particular norm is done according to noise model. We assume white gaussian noise, thus we use Frobenius matrix norm, defined by: $\|X\|_F = \sqrt{\sum_{ij} X_{ij}^2}$.

However, the direct solution of above problem does not lead to good estimate of source matrix $S$, since the problem is ill-possed: there much more possible locations than sensors. In order to regularize a solution, we use a sparse prior: we assume that the sources $S$ are sparsely representable in some basis or overcomplete system of functions e.g. Gabor, wavelet, wavelet packet, etc. (see for example [31]). Particularly, there exists some operator $\Phi$ and the *sparse* matrix of coefficients, $C$, such that:

$$S = C\Phi \tag{2.13}$$

The matrix $\Phi$ contains elements of the chosen basis in its rows. The rows of matrix $C$ will contain the coefficients of decomposition of time-domain source signals in a chosen basis.

In addition to temporal sparsity, we will enforce a spatial sparsity, as proposed in [49]. This sparsity indeed exists in our problem formulation.

Recall, that we've divided the space into a set of possible locations, and there are much more locations than active sources.

All of the above leads to the following objective function, which has to be minimized in $C$:

$$F(C) = \frac{1}{2} \|Y - \mathcal{A}(C\Phi)\|_F^2 + \mu_1 \sum_{i,j} |c_{ij}| + \mu_2 \sum_{i=1}^{m} \|c_i\|_2 \tag{2.14}$$

where $c_i$ denotes the $i$-th row of the matrix C (the $i$-th source' coefficients), and $c_{ij}$ is the $j$-th element in $c_i$. The scalars $\mu_1$ and $\mu_2$ are used to regulate the weight of each term.

The first term in (2.14) is the Frobenius-norm-based data fidelity. The second term intends to prefer sparsely representable signals in time; it is based on the $l_1$-norm, which had been previously proven to be effective in enforcing sparsity [31, 24]. The third one is the spatial sparsity regularizing term, which is intended to prefer solutions with the source signals concentrated in a small number of locations. It is easy to see that moving some coefficient from an active source location to an empty one, will strictly increase this term.

Note, that we have chosen $C$ (and not $S$), to be the variable of proposed objective function. This choice is intentional: the transformation from $C$ to $S$ always exists, and it is defined in (2.13). However the inverse transformation is not always defined (for example when the chosen system of functions is *overcomplete*).

Note also, that quite often matrix $\Phi$ does not need to be stored explicitly. Multiplication by $\Phi$ and $\Phi^*$ corresponds to the synthesis and analysis operation in some signal dictionary, and may be performed very efficiently (like

for example fast wavelet or wavelet packet transform), see [31] for details and more examples.

In order to minimize the objective (2.14) numerically, we use a smooth approximation of the $l_2$-norm, having the following form[3]:

$$\psi(x) = \sqrt{\sum_i x_i^2 + \epsilon} \approx \|x\|_2 \tag{2.15}$$

the approximation becomes more precise as $\epsilon \to 0$. It can be easily seen, that if $\psi$ is applied to a single element of $x$ - it becomes the smooth approximation of absolute value:

$$\psi(x_i) = \sqrt{x_i^2 + \epsilon} \approx |x| \tag{2.16}$$

Using (2.15) and (2.16), we obtain the following objective function:

$$F(C) = \frac{1}{2}\|Y - \mathcal{A}(C\Phi)\|_F^2 + \mu_1 \sum_{i,j} \psi(c_{ij}) + \mu_2 \sum_{i=1}^{m} \psi(c_i) \tag{2.17}$$

### 2.3.2 Choosing Optimization Technique

We can efficiently calculate both the $\mathcal{A}S$ and the $\mathcal{A}^*Y$ products, which enables us to calculate the gradient matrix $G$ and the product of the Hessian operator $\mathcal{H}$ with an arbitrary matrix $X$ (see subsection 2.3.3). Hence, the objective (2.17) can be minimized by one of the numerical optimization methods, for example the *Quasi Newton* method. A problem arises when the dimension of the problem grows. The memory consumption and iteration cost grow as $(mT)^2$. This circumstance leads us to the usage of the *Truncated Newton* method [50],[51]. In the *Truncated Newton* method the

[3] An alternative would be to solve the problem in the Conic Programming framework; we leave this option for the future

Newton direction $D$ is found by the approximate solution of the system of linear equations $\mathcal{H}D = -G$. This is done by the *linear Conjugate-Gradients* method. We use diagonal preconditioning in order to further speed up the optimization [52]. Note that in *Truncated Newton* method, the memory consumption grows linearly with the number of variables. This enables us to solve large problems with fair performance. See next subsection for detailed calculation of objective function derivatives and Appendix A.2 for detailed description of *Truncated Newton* and preconditioned *Conjugate-Gradients* algorithms.

### 2.3.3 Objective Function Derivatives

In order to use the *Truncated Newton* method, we need to calculate the gradient $G$ of the objective (2.17), as well as to implement the product of the Hessian $\mathcal{H}$ with an arbitrary matrix $X$. We also derive multiplication by the diagonal of $H$, required for preconditioned *Conjugate-Gradients*. Note that $\mathcal{H}$ is a tensor, but if we parse the matrix variable $C$ into a long vector, then a Hessian will be represented by a matrix $H$. We will use these notations throughout this section. We will also use $G_i$ and $\mathcal{H}_i$ to denote gradient and Hessian of respective terms of the objective function (2.17).

Let us start with the first term in (2.17). We will define a new operator $\mathcal{B}$ in the following way:

$$\mathcal{B}C = \mathcal{A}(C\Phi) \qquad \mathcal{B}^*X = (\mathcal{A}^*X)\Phi^* \tag{2.18}$$

This enables us to write the first term in (2.17) as: $F_1 = \frac{1}{2}\left\|\mathcal{B}C - Y\right\|_F^2$. If we introduce new variable $U = \mathcal{B}C - Y$, then $F_1 = \frac{1}{2}\left\|U\right\|_F^2 = \frac{1}{2}Tr(U^TU)$.

Hence, $dF_1 = \frac{1}{2}\left(Tr(U^T dU) + Tr(dU^T U)\right) = Tr(U^T dU)$. Substituting $U$ and $dU = \mathcal{B}dC$ yields $dF_1 = Tr\left((\mathcal{B}C - Y)^T \mathcal{B}dC\right) = \langle \mathcal{B}C - Y, \mathcal{B}dC\rangle = \langle \mathcal{B}^*(\mathcal{B}C - Y), dC\rangle$. Recall that $dF = \langle G, dC\rangle$, and we get the gradient

$$G_1(C) = \mathcal{B}^*(\mathcal{B}C - Y) \tag{2.19}$$

In order to calculate the multiplication of the Hessian operator $\mathcal{H}$ by an arbitrary matrix $X$ we need to recall that $dG(C) = \mathcal{H}dC$. By (2.19) $dG_1(C) = \mathcal{B}^*(\mathcal{B}dC)$, and thus for an arbitrary $X$

$$\mathcal{H}_1 X = \mathcal{B}^*(\mathcal{B}X) \tag{2.20}$$

parentheses are used to ensure correct order of multiplications.

In order to proceed with the second and the third terms of objective (2.17), we need to use the gradient and Hessian of (2.15):

$$\nabla\psi(x) = \frac{1}{\psi(x)}x \tag{2.21}$$

$$\begin{aligned}\left(\nabla^2\psi(x)\right)_{ii} &= -\frac{1}{\psi^3(x)}x_i^2 + \frac{1}{\psi(x)} \\ \left(\nabla^2\psi(x)\right)_{ij} &= -\frac{1}{\psi^3(x)}x_i x_j \qquad (i \neq j)\end{aligned} \tag{2.22}$$

where $(\nabla^2\psi(x))_{ii}$ and $(\nabla^2\psi(x))_{ij}$ are diagonal and off diagonal elements elements of $\nabla^2\psi(x)$ respectively. Now, by straightforward calculations we can write down the gradients of the second and the third term in (2.17):

$$(G_2)_{ij} = \mu_1 \frac{1}{\psi(c_{ij})}c_{ij} \tag{2.23}$$

$$(G_3)_{ij} = \mu_2 \frac{1}{\psi(c_i)}c_{ij} \tag{2.24}$$

note, that the gradient of (2.17) is a matrix, because our variable $C$ is also a matrix (hence $G_1$,$G_2$ and $G_3$ are also matrices). It can be noticed in (2.23), that all elements of $G_2$ are independent, and thus the $H_2$ matrix will be diagonal. It is convenient to 'pack' the diagonal of $\mathcal{H}_2$ into a matrix with the same size as $C$ row by row. Let us denote the packed matrix as $\tilde{H}_2$:

$$\tilde{H}_{2_{ij}} = \mu_1 \left( -\frac{1}{\psi^3(c_{ij})} c_{ij}^2 + \frac{1}{\psi(c_{ij})} \right). \tag{2.25}$$

It is obvious, that

$$\mathcal{H}_2 X = \tilde{H}_2 \odot X \tag{2.26}$$

where $\odot$ is element-wise multiplication.

In order to define the multiplication $\mathcal{H}_3 X$ we need to rewrite the equation (2.22):

$$\nabla^2 \psi(c_i^T) = \frac{1}{\psi^3(c_i^T)} c_i^T c_i + \frac{1}{\psi(c_i^T)} I \tag{2.27}$$

where $I$ represents the identity matrix. Now it is easy to define the $i$-th row of $\mathcal{H}_3 X$:

$$(\mathcal{H}_3 X)_i = \mu_2 \left( -\frac{1}{\psi^3(c_i^T)} c_i (c_i x_i^T) + \frac{1}{\psi(c_i^T)} x_i \right) \tag{2.28}$$

where $x_i$ is the $i$-th row of matrix $X$.

This calculus is sufficient for the *Truncated Newton* method. However, in order to use *Preconditioned* Conjugate Gradients method for inner iterations, we need to define the diagonal of the Hessian of (2.17).

We will calculate the elements in the diagonal of $\mathcal{H}_1$ in the following manner: let $E$ be a zero matrix with only one non-zero element equal to 1 at an arbitrary location - $i$-th row and $j$-th column. Then:

$$\left( \tilde{H}_1 \right)_{ij} = \langle E, \mathcal{H}_1 E \rangle \tag{2.29}$$

where $\tilde{H}_1$ is a diagonal of $\mathcal{H}_1$ packed in the same manner as a diagonal of $H_2$ in (2.25).

It follows from (2.20) that $\langle E, \mathcal{H}_1 E\rangle = \langle E, \mathcal{B}^*(\mathcal{B}E)\rangle = \langle \mathcal{B}E, \mathcal{B}E\rangle = \|\mathcal{B}E\|_F^2$.

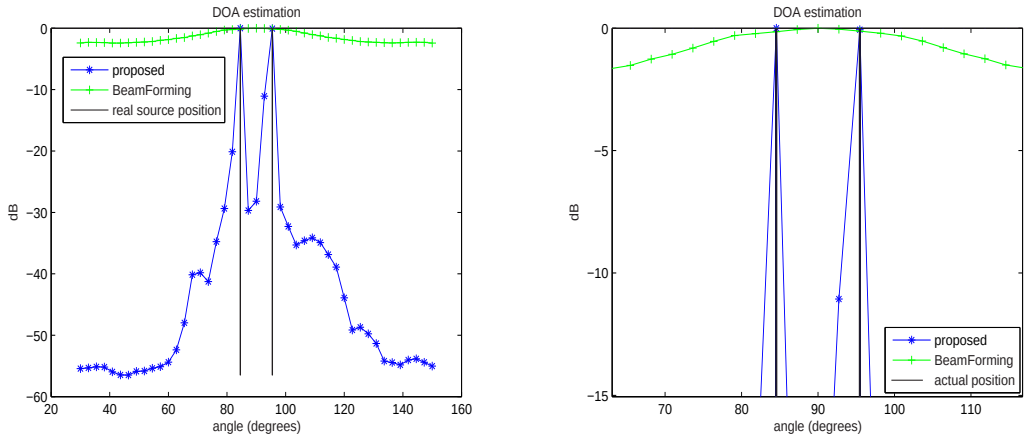
The diagonal of $\mathcal{H}_2$ is already defined in (2.25). Finally, the diagonal of $\mathcal{H}_3$, packed in the same manner as a diagonal of $\mathcal{H}_2$, is given by:

$$(\tilde{H}_3)_{ij} = \mu_2 \left( -\frac{1}{\psi^3(c_i)} c_{ij}^2 + \frac{1}{\psi^3(c_i)} \right) \tag{2.30}$$

## 2.4 Computational Experiments

In this section we evaluate the proposed *SMSR* algorithm. Conducted simulations demonstrate the feasibility of our method. Moreover, we show that the proposed approach is robust to noise and very limited data size. In addition, it doesn't require an accurate initialization. Another strength of our algorithm is that it is able to resolve closely spaced sources, i.e. it is able to achieve supper-resolution. However, the disadvantage of our approach is a high computation complexity.

In our simulations we did not use signals recorded in a real-word environment, rather we have generated the source signals $S$ and sensors' measurement matrix $Y$. However, we have tried to keep our simulations as close as possible to the real-word problems. Thus, we've chosen the following generation procedure: first, we have generated the sparse coefficients matrix $C$. Next, the source signals were created by $S = C\Phi$. In this way an existence of sparse representation of source signals (the assumption of our approach) is guaranteed by generation. In our simulations we have used Symlet8 Wavelets. Finally, sensors' measurement matrix is created by: $Y = \mathcal{A}S + N$, where $\mathcal{A}$ defined in (2.12) and $N$ represents the additive zero-mean white Gaussian noise. In this generation procedure, we have set parameters $N_{up}$,$N_{io}$ and $\rho(t)$ in (2.10) to have different values, from those used later in reconstruction procedure. This is done in order to introduce the "model error".




(a) (b)

Figure 2.3: (a) Noise Free Case. The proposed method correctly identifies the location of sources.(b) Zoom-in picture, in which it can be clearly seen that beamforming fails to separate closely spaced sources.

### 2.4.1 Far-Field Model

We start our simulations with basic scenario: we assume far field $2D$ model and sensors lined up with constant distances. In this scenario, we divided the space into discrete grid of angular locations. The delay of the $j$-th source location toward the $i$-th sensor is easy to calculate, given the geometrical position of each sensor and assuming that the source is far enough, so that signal arrives as a planar wave (far field assumption).

#### 2.4.1.1 Noise-Free Case

First, we decided to demonstrate the feasibility of our approach, starting with noiseless environment. The experimental setup is as following: 4 sensors

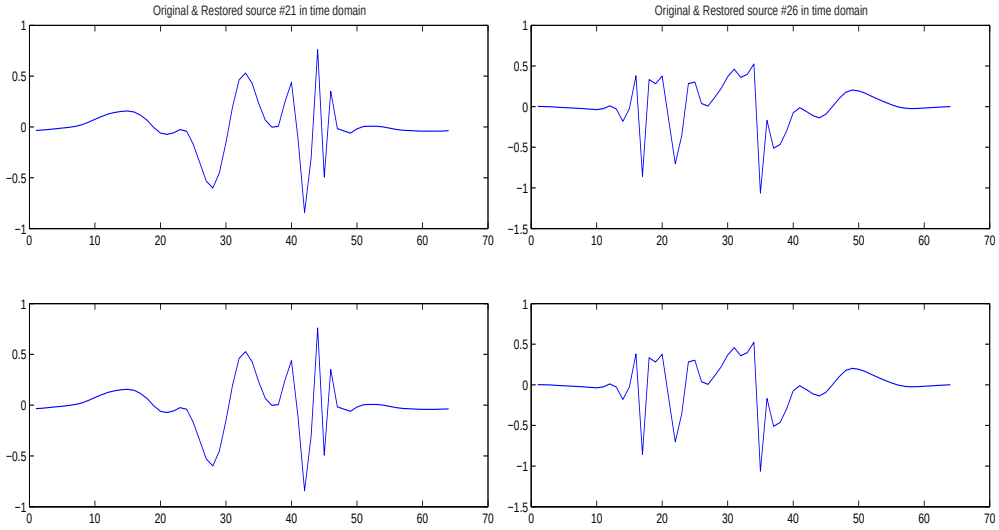



Figure 2.4: Source Separation (no noise). Top: sources from 2 active directions, bottom: restored sources. The normalized reconstruction error is less than 1e-3.

are lined up with $\frac{\lambda_{min}}{2} = \frac{1}{2}\frac{\mathcal{C}}{f_{max}}$ distance (we assume our signal to be band limited, and $f_{max}$ denoting the highest frequency). Signals are arriving from 45 possible directions, and they are 64 time samples-long.

We've chosen to have only 2 active sources, located very close to each other - within $10^o$. In these conditions conventional methods, such as beamforming and MUSIC fail to super-resolve them (as shown in [53],[54]). The experiment was successful. Using proposed algorithm, we have correctly estimated the source positions (Figure 2.3(a)), while the conventional delay-sum beamforming has failed to resolve the closely spaced sources (Figure 2.3(b)). We have also correctly reconstructed the sources. The normalized reconstruction error was less than $5 * 10^{-3}$ (Figure 2.4) (the error was calculated according to $\left\| \frac{s_{init}}{\|s_{init}\|_2} - \frac{s_{rec}}{\|s_{rec}\|_2} \right\|_2$).

#### 2.4.1.2 Noisy Environment

In the next experiment, we have added white Gaussian noise to the matrix $Y$. The SNR at each sensor was $5dB$. The contaminated by the noise matrix $Y$ was used as an input to our algorithm.

We have compared results based on our approach (based on spatial and temporal sparsity), with the method based on spatial sparsity only, in spirit of ([53],[54]). This can be done by setting $\mu_1 = 0$ in (2.17). We have also done the DOA estimation by conventional delay-sum beamforming. Then we have computed the energy of the restored signals at each direction for all methods and compared the results. It can be seen from Figure 2.5 that proposed method has correctly identified the active source directions, however if we optimize on the spatial sparsity only, we fail to correctly detect the source positions. The conventional delay-sum beamforming has also failed in DOA estimation task. In the first experiment, it has failed to resolve the closely-spaced sources (Figure 2.5(b)). In the second experiment, one of the signals has about twice larger energy than the other. In this case, the beamforming has found only the first source (Figure 2.5(d)).

We have also checked the signal reconstruction performance of our algorithm in the noisy environment. We have compared original vs. reconstructed signals from active directions. As one can see in Figure 2.6 the active sources were restored rather accurately. The reconstruction error was about 5e-2. This is a very good result, particularly if we take into consideration a high noise level (SNR=5dB) of the sensor signals.

Figure 2.7 demonstrates the convergence of the algorithm vs. number

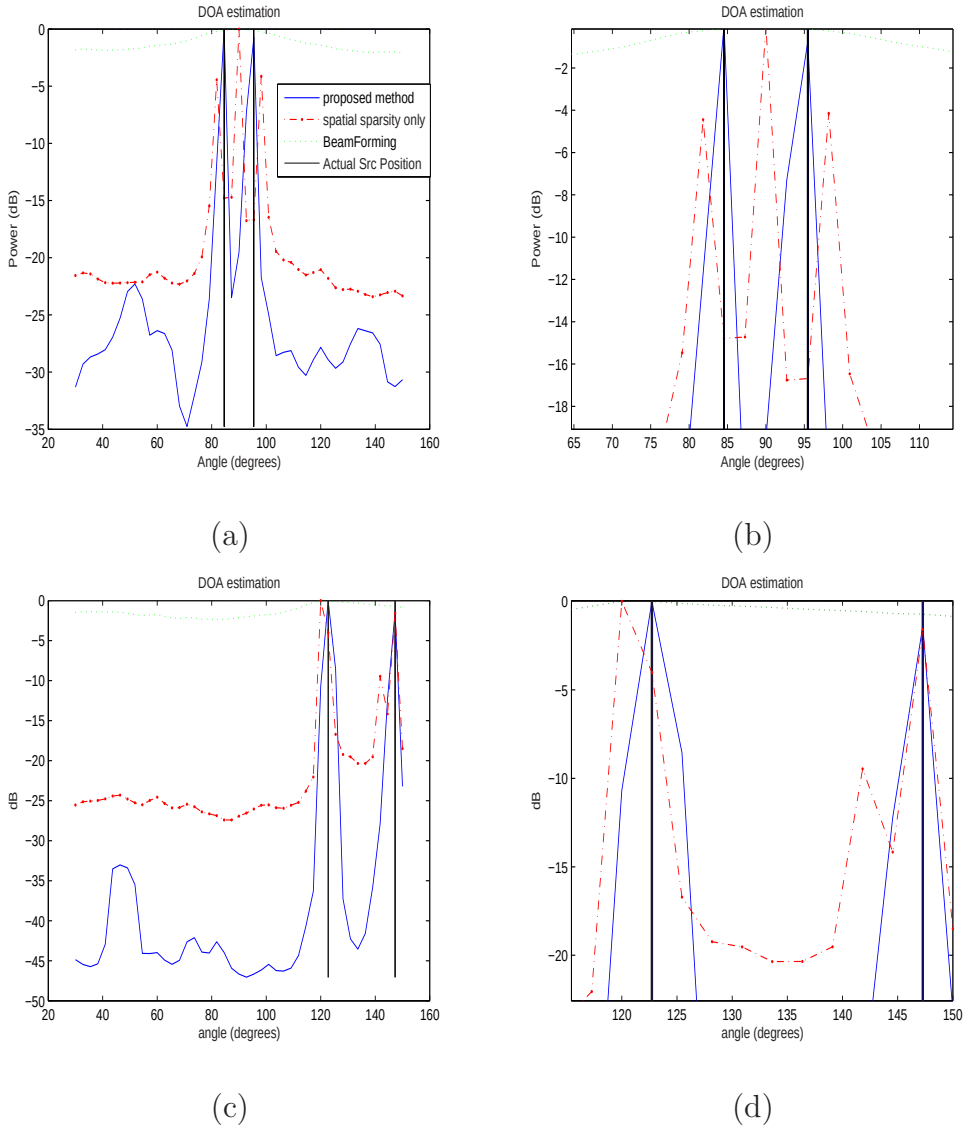



Figure 2.5: Examples of DOA estimation by proposed and alternative methods (SNR=5 db). (a),(c) Solid line represents DOA estimation by proposed method. Two peaks coincide with actual source positions (marked by solid vertical line). However, DOA estimation based on spatial sparsity only (dash-dot line) fails to correctly identify the sources. The beamformer (dotted line) also fails to resolve the sources. (b),(d) Zoom-in pictures, in which details can be seen more clearly.

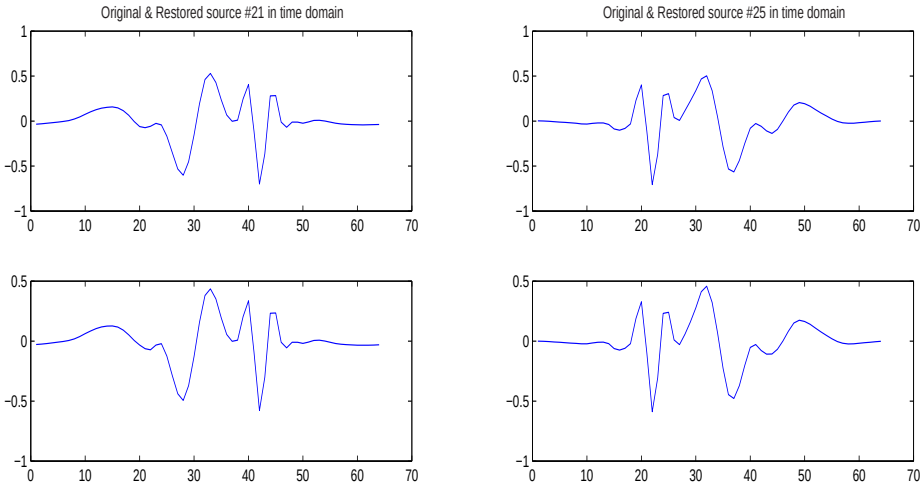



Figure 2.6: Source Separation (SNR=5dB). Top: sources from 2 active directions, bottom: restored sources. The normalized reconstruction error is about 5e-2.

of iterations. We have compared the computational load of the algorithm for two cases: with and without preconditioning of Conjugate Gradients (CG) (see Appendix A.2 for description of Truncated Newton (TN) and CG algorithms). The number of Truncated Newton (outer) iterations was slightly less for un-preconditioned version. However with preconditioning the total number of CG iterations was reduced by a factor of ≈45 and the total computation time by a factor of ≈28 (5.3 min vs. 150 min).

### 2.4.2 Near-Field Model

In this experiment we have evaluated the source localization and reconstruction capabilities of our algorithm in the near field scenario.

The problem setup was as following: we assume nearfield, 2D environ-

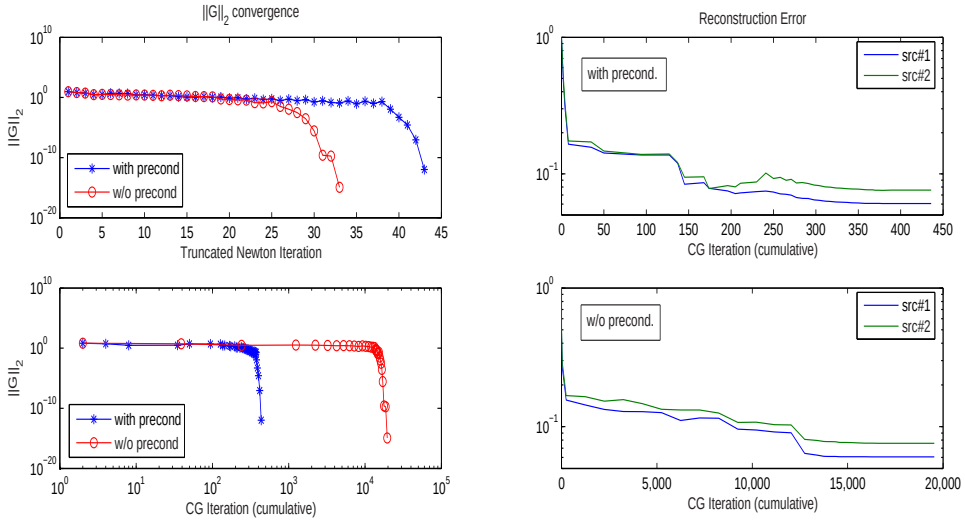



Figure 2.7: Optimization convergence. The usage of preconditioning in CG drastically reduces the computational load. The total number of iterations were reduced by a factor of ≈45. The computation time was reduced by a factor of ≈28 due to preconditioning.

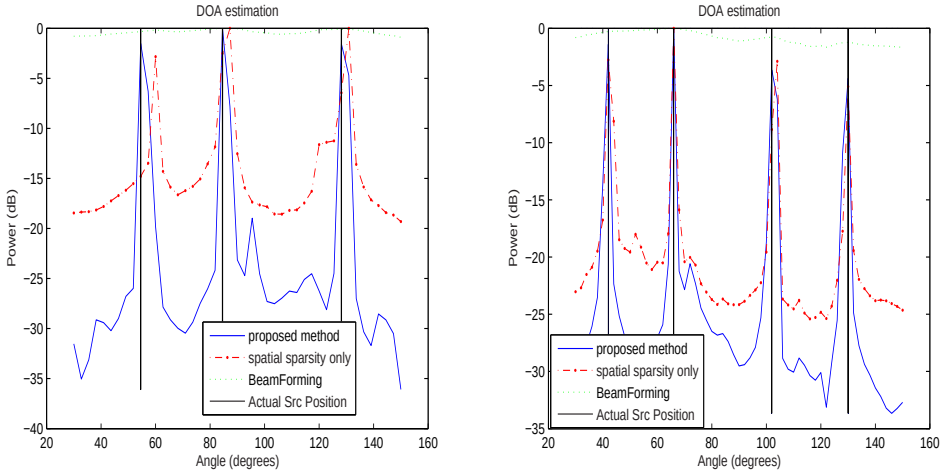



Figure 2.8: More examples of DOA estimation by proposed and alternative methods (SNR=10 db). The proposed method has correctly identified the source locations. However, the method based on spatial sparsity only has a bias with respect to some source locations.

ment. The "2D room" was chosen to have 1mX1m dimensions, and was divided into equally spaced 400 locations. Then, we have randomly chosen locations for 4 sensors and 3 active sources. Transfer functions from each location to each sensor was assumed to reflect the direct path propagation and 3 reflections from the walls. The generation of transfer functions was done using the code available at public website [55]. The remaining setup procedure was similar to previous cases.

First, we have checked the feasibility of our approach on the noiseless case. We have successfully identified the source locations. The results can be seen in the figure 2.9.

We have also tried to solve the problem in the noisy environment. When applying our method for noisy environment, we have to make sure, that the

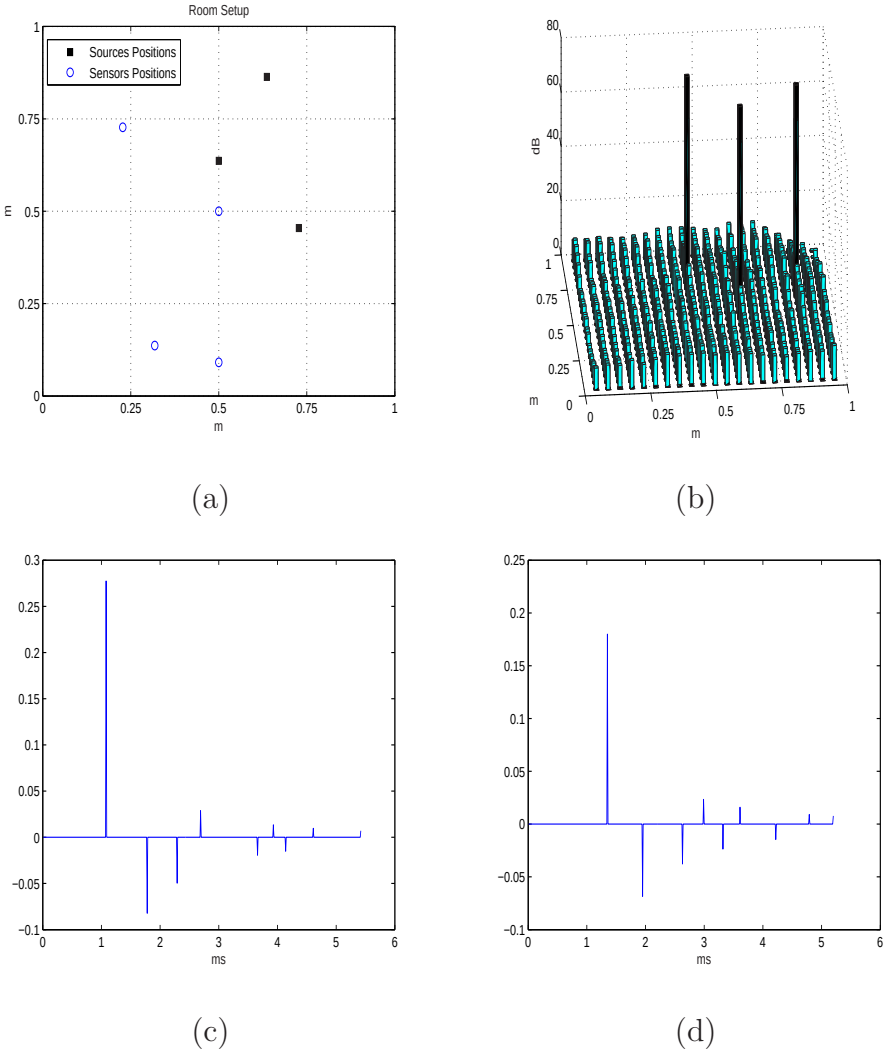



Figure 2.9: 2D near-field, noiseless environment. (a) Sensors and active sources positions. Four sensors are depicted by circles. Three active sources are depicted by squares. (b) Estimated signal energy at each possible location. The proposed algorithm has correctly identified locations of active sources. (c),(d) example of impulse responses used to simulate the propagation of signals

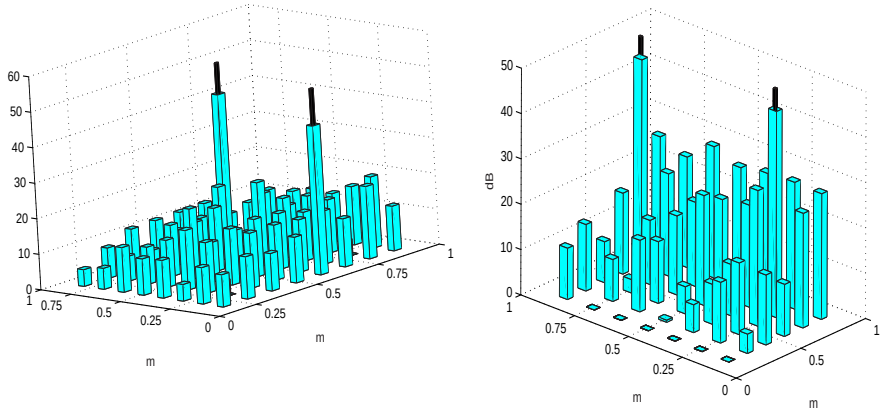



Figure 2.10: Two examples of source localization in near-field noisy environment (SNR=10dB). In both cases the true source locations were identified correctly.

position of sensors and the noise level are appropriate: i.e. each of source signals reaches at least two sensors with amplitudes larger than a noise. This is necessary to avoid ambiguity in source localization.

When the above restriction is met, proposed *SMSR* method has fairly good performance. Two examples of source localization in near-field noisy environment are shown in Figure 2.10. The signal-to-noise ratio was 10dB. The locations of active sources were identified correctly in both cases. The source signals themselves were reconstructed with (normalized) error of about 6e-2.

### 2.4.3 Robustness to Noise and Model Errors

The proposed enforcement of temporal sparsity along with spatial sparsity was shown by our simulation to have additive value. In most cases it helped

to achieve lower side-lobes level in DOA estimation. In some cases it facilitated correct localization of sources, that the spatial-sparsity-only method has failed to identify correctly. In this section, we will compare the robustness of two approaches to model errors and noise.

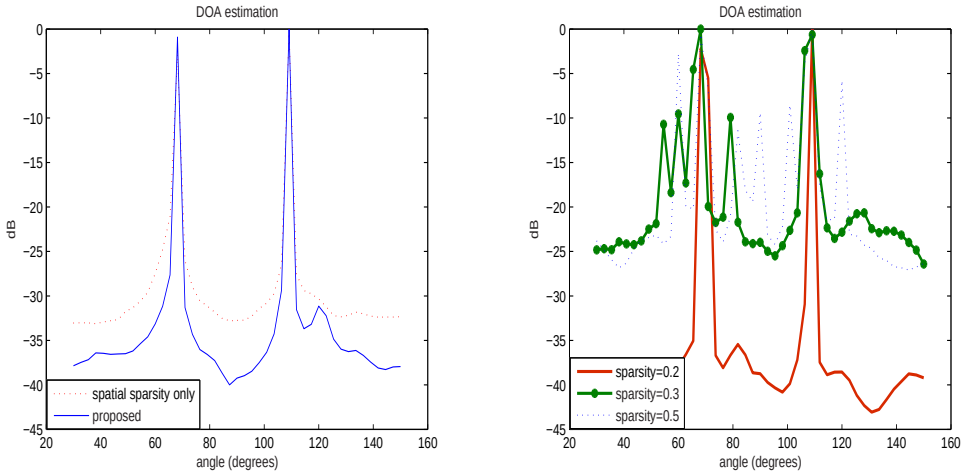
#### 2.4.3.1 Robustness to temporal non-sparsity

Although it is obvious, that aforementioned additive value of enforcement of temporal sparsity along with spatial sparsity is achieved only when the assumption of temporal sparsity of sources is valid, it is interesting to estimate the robustness of proposed algorithm to this assumption.

In order to estimate the robustness of proposed algorithm to temporal sparsity assumption, we have evaluated our algorithm in the basic scenario of far-field, linear array of 4 sensors and two sources. The sources were chosen each time with different degree of temporal sparsity (fraction of non-zero elements).

First, we have estimated the location of sources by proposed and spatial-sparsity-only methods. The sources were chosen to have temporal sparsity equal to 0.1. As one can see on Fig. 2.11(a), proposed method achieves lower side-lobe levels. Then, we have kept parameters $\mu_1$ and $\mu_2$ in (2.17) unchanged[4] and evaluated the proposed method on sources with different degree of sparsity. As one can see on Fig. 2.11(b), the performance of proposed algorithm is still good for temporal sparsity of 0.2. However, as temporal sparsity growth to 0.5, the performance gradually degrades and we receive

[4]We did not change optimization parameters on purpose, since by setting $\mu_1 = 0$ we can actually obtain spatial-sparsity-only DOA estimation




(a) (b)

Figure 2.11: Far-field scenario, 4 sensors, 2 sources. Illustration of DOA estimation by proposed algorithm for sources with different degree of temporal sparsity (fraction of non-zero elements) (a) Doa estimation by proposed and spatial-sparsity-only method. Temporal sparsity of sources is 0.1. Proposed algorithm exhibits lower side-lobe levels. (b) Doa estimation by proposed for different values of temporal sparsity of sources. As number of non-zero elements grows, the assumption of temporal sparsity becomes invalid and consequently the performance of proposed method degrades.

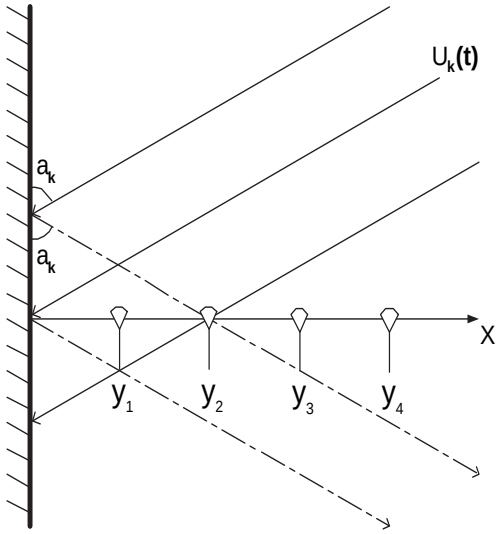



Figure 2.12: Illustration of far field model with reflection. Infinite wall is placed to the left of the linear sensor array. Signals arrive to the array in the straight path, and as reflection from the wall.

spurious peaks. This experiment demonstrates that proposed algorithm has some degree of robustness to incorrect assumption of temporal sparsity.

#### 2.4.3.2 Robustness to incorrect estimation of impulse response

In order to estimate the robustness of proposed algorithm to incorrect estimation of impulse responses, we have conducted the following experiment: we still stick to the far-field $2D$ model and linear array, but this time we place an infinite "wall" to the left of the sensor array (figure 2.12). Signals that arrive from angles $0 < \alpha < 90$ reach the array in the straight path and as reflection from the wall. Signals that originate from directions $90 < \alpha < 180$ do not reach the array. We have generated the sensor measurement matrix using "full" transfer function (which takes into account the reflections from

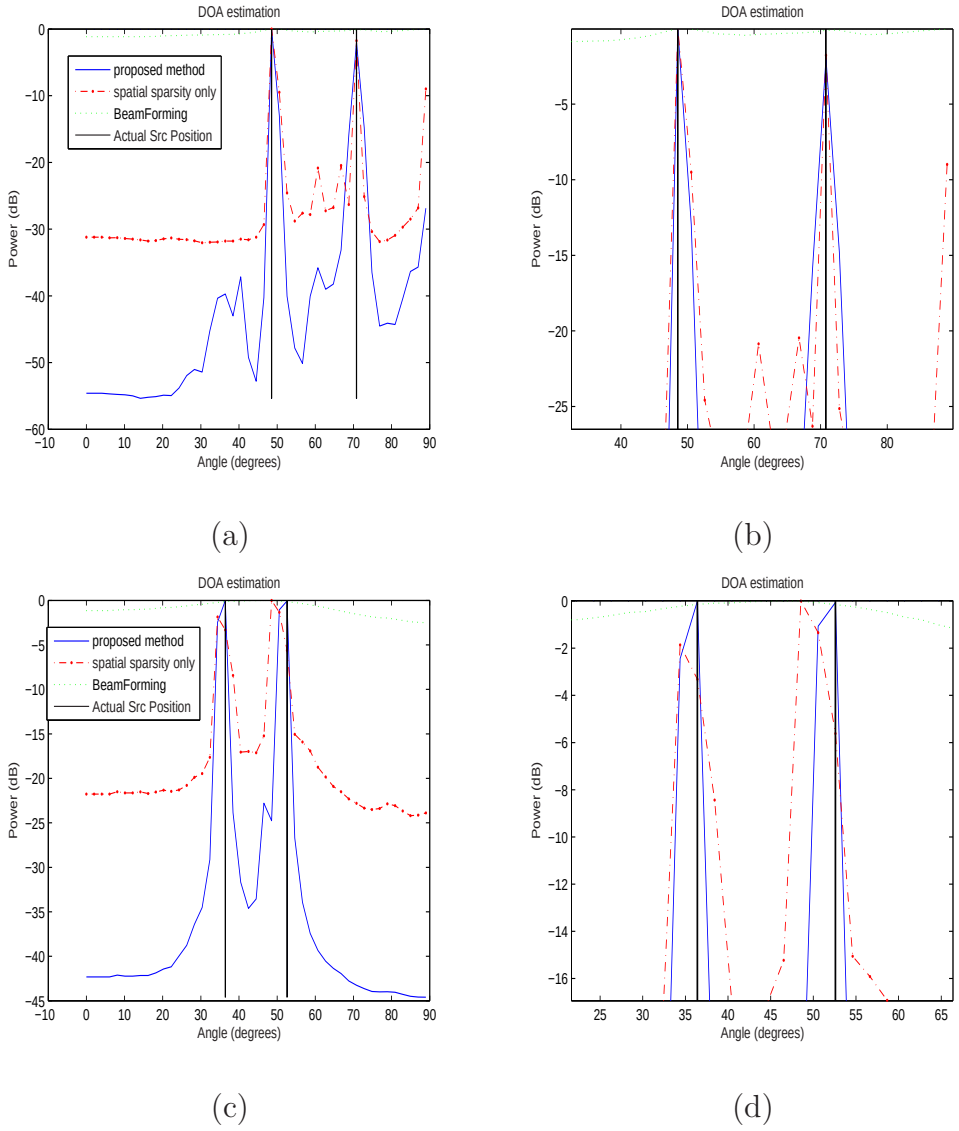



Figure 2.13: Examples of DOA estimation in reverberant environment. (a) *Convolutive* reconstruction (SNR=5dB, reflection factor= 0.9). The proposed algorithm has correctly estimated the DOAs. The side-lobes level is about 15dB lower, with comparison to DOA estimation based on spatial sparsity only (b) *Convolutive* reconstruction, zoom-in picture (c) *straight-path only* reconstruction (SNR=5dB, reflection factor= 0.3). Source locations were correctly identified by proposed method. However the DOA estimation based on spatial sparsity only has failed to correctly locate the sources. (d) *straight-path only* reconstruction, zoom-in picture.

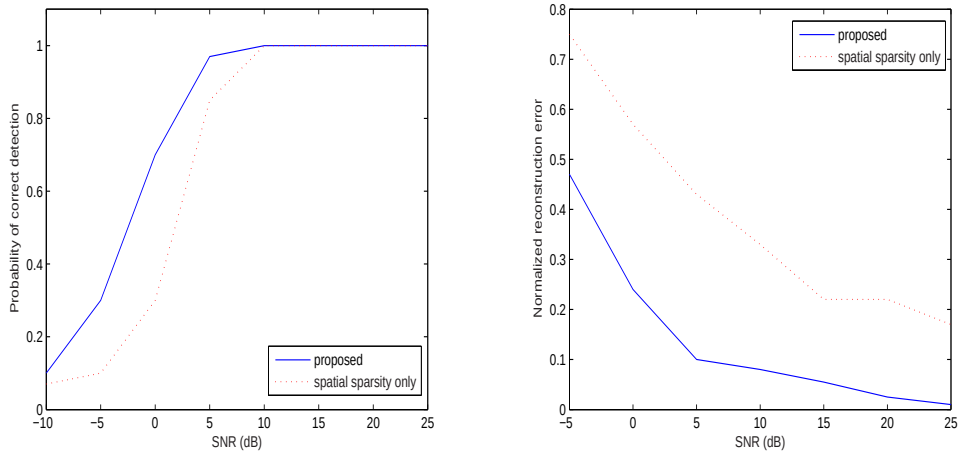
the wall). Then, we have performed the reconstruction in two ways: using the "full" transfer function (*Convolutive* reconstruction) and using the *straight-path only* approximation of the transfer function, which doesn't include the reflections from the wall (*straight-path only* reconstruction).

Our experiments show, that using *Convolutive* reconstruction, we are able to correctly identify the source locations even in the case, when the echo has the same amplitude as the straight-path signal (reflection factor equal to 1). However, using the *straight-path only* technique, we are able to restore sources only in the light reverberation conditions (reflection factor equal or less than 0.3), see Fig. 2.13. This results shows, that proposed algorithm was able to deal with incorrect estimation of transfer function, when the omitted reflection was 10dB lower than the direct path signal. The algorithm based on spatial sparsity only, was able to correctly estimate location of sources when the reflection factor was less then 0.15 (equivalent to omitted reflection energy is lower by 16.5dB or more, with respect to the energy of direct path signal)

#### 2.4.3.3 Robustness to Noise

We have estimated the robustness to noise of proposed and spatial-sparsity-only based methods by probability of correct estimation of source locations and sources reconstruction error.

We have chosen the basic scenario: far-field, linear array of 4 sensors, 2 sources at $80^o$ and $100^o$. We have run 20 simulation for different SNR values. Figure 2.14(a) shows, that both algorithms accurately estimate DOA for high SNR values. However for low SNR values, the enforcement of both temporal




(a) (b)

Figure 2.14: Far-field scenario, 4 sensors, 2 sources at $80^o$ and $100^o$. Comparison of proposed and spatial-sparsity-only methods by (a) Probability of correct DOA estimation. (b) average normalized source reconstruction error

and spatial sparsity results in higher probability of correct DOA estimation.

Moreover, one can observe on Fig. 2.14(b), that even at high SNR values, spatial-sparsity-only based algorithm has poor performance for source reconstruction. The proposed approach, on the contrary, estimates the sources rather accurately even at moderate SNR values.

### 2.4.4 Conclusions

Conducted simulations demonstrates the additive value of enforcement of temporal sparsity along with spatial sparsity. In most cases, the proposed *SMSR* algorithm achieves lower side-lobe levels, than the method based on spatial sparsity only. In some cases, it is able to resolve sources, which the spatial sparsity only method fails to resolve.

In addition, the proposed enforcement of both temporal and spatial sparsity makes the proposed approach capable to accurately reconstruct the source signals even at high noise levels. Moreover, the proposed problem formulation allows incorporation of reverberant ("*full*") transfer functions in the model, which leads to *convolutive* reconstruction. The later has much better performance than the original *straight-path only* approach.

The weaknesses of our approach are high computational complexity and the need to subjectively assess the trade-off parameters. These issues can serve as challenging topics for further research.

# Chapter 3

# Learning Spatial and Spectral Filters for Single-Trial EEG Classification

In this chapter, we present our method for spatial and spectral multi-sensor EEG signal filtering. Proposed method is based on the assumption that the "response" to the mental task is smooth, and is contained in several sensor channels. We find both *spatial* and *spectral* filters "blindly", by optimizing the between class discrimination and the smoothness of result average. We assume no other information on signals of interest is available.

This chapter is organized as follows. In Section 3.1 we describe the first stage of our preprocessing method, which is based on spatial integration. In Section 3.2 we develop additional method for spatial filtering based on Eigenvalue Decomposition. In Section 3.3 we develop a bound, which shows how well can we perform signal reconstruction (and further classification)

based on spatial integration only. In section 3.4 we describe the second stage of our algorithm, which is based on time-domain filtering. Section 3.5 is devoted to computational experiments. Finally, we discuss the results in Section 3.6.

## 3.1 Spatial Integration Method

### 3.1.1 Data Description

In this section we provide the general description of the data format, which we use in our preprocessing method.

Suppose EEG data was recorded using $S$ channels. Single trial signals, corresponding to one of the two possible mental tasks were taken from the raw data, synchronized by some external stimuli or cue, and they are $T$ samples long. Each single trial is stored in the $T \times S$ matrix. Let us denote $X_l^1, 1 \leq l \leq L$ trials that belong to the first class, and $X_m^2$ , $1 \leq m \leq M$ trials that belong to the second class.

If we produce the averaging among the trials, we obtain

$$X_{avg}^1 = \frac{1}{L}\sum_{l=1}^{L} X_l^1 \qquad X_{avg}^2 = \frac{1}{M}\sum_{m=1}^{M} X_m^2$$

where $X_{avg}^1$ and $X_{avg}^2$ are $T \times S$ matrices.

### 3.1.2 The Method

In our model, we assume that **each sensor** records the following signal:

$$x_i(t) = a_i s^j(t) + n_i(t) \tag{3.1}$$

where $a_i$ is the coupling coefficient for sensor $i$, $n_i(t)$ denotes the noise and background activity recorded by the sensor and $s^j(t)$ , $j \subset \{1, 2\}$ is the response to one of the two possible mental tasks.

We will use a linear estimate of single trial signals

$$\hat{s}_i^1 = X_i^1 w \ , \ \hat{s}_j^2 = X_j^2 w \tag{3.2}$$

where $w$ is the $S \times 1$ *weighting vector*.

The average of the estimated signals is:

$$\hat{s}_{avg}^1 = X_{avg}^1 w \ , \ \hat{s}_{avg}^2 = X_{avg}^2 w \tag{3.3}$$

Using the above notation, we can formulate our objective as finding the weighting vector $w$ such, that will maximally discriminate between the average estimated signals $\hat{s}_{avg}^1$ and $\hat{s}_{avg}^2$, while keeping single trial estimated signals $\hat{s}_l^1$ and $\hat{s}_m^2$ $(1 \leq l \leq L \, , \, 1 \leq m \leq M)$ smooth. The smoothness can be measured e.g. by *total variation*, defined as

$$\Phi(\hat{s}) = \sum_{l=1}^{L} \sum_{t=1}^{T-1} z_l^1(t) + \sum_{m=1}^{M} \sum_{t=1}^{T-1} z_m^2(t) \tag{3.4}$$

$$\text{where} \ \ z_l^1(t) = |\hat{s}_l^1(t+1) - \hat{s}_l^1(t)| \ , \ 1 \leq t \leq T-1$$

$$\text{and} \ \ z_m^2(t) = |\hat{s}_m^2(t+1) - \hat{s}_m^2(t)| \ , \ 1 \leq t \leq T-1$$

This leads to following objective function:

$$\min_w - \left\| \hat{s}_{avg}^1 - \hat{s}_{avg}^2 \right\|_2^2 + \mu \Phi(\hat{s}) \tag{3.5}$$

$$s.t. \left\| w \right\|_2 = 1$$

where $\mu$ is a tradeoff parameter, which is intended to balance between smoothness of signals and between class discrimination. We are forcing the norm of

weighting vector $w$ to remain constant in order to prevent degenerate solution of $\|w\| \to \infty$ or $\|w\| \to 0$.

If we substitute expressions for $\hat{s}^1_{avg}$,$\hat{s}^2_{avg}$ and $\Phi(\hat{s})$ from equations (3.2),(3.3) and (3.4), after few simple algebraic steps the objective function (3.5) becomes:

$$\begin{aligned} &\min_{w} - \|X_{avg} w\|_2^2 + \mu \left( \|Y w\|_1 \right) \\ &s.t. \, \|w\|_2 = 1 \end{aligned} \tag{3.6}$$

where $\|\cdot\|_1$ is the first norm, $X_{avg} = X^1_{avg} - X^2_{avg}$ and $Y$ is a block-matrix, composed of matrices $Y^1_l, 1 \leq l \leq L$ and $Y^2_m, 1 \leq m \leq M$ placed one under another; i.e. if matrices $Y^1_l$ and $Y^2_m$ have dimensions of $T-1 \times S$, then a matrix $Y$ will have dimensions of $(L+M)(T-1) \times S$.

Matrices $Y^1_l$ and $Y^2_m$ are defined as:

$$Y^1_l(t,i) = X^1_l(t+1,i) - X^1_l(t,i) \ , \ 1 \leq t \leq T-1$$

$$Y^2_m(t,i) = X^2_m(t+1,i) - X^2_m(t,i) \ , \ 1 \leq t \leq T-1$$

note that $z^1_l = Y^1_l w$, $z^2_m = Y^2_m w$.

### 3.1.3 Close look at the Objective Function

One can ask the following question: Why should we measure discrimination between classes on average signals, while smoothness ($TV$) is measured on single trial signals? Why couldn't it be vise versa, for example? In order to answer this question, we need to analyze what we want to achieve by minimizing the objective function.

#### 3.1.3.1 Between Class Discrimination

Let's start with between class discrimination. We can measure it on average signals (as we do in objective (3.6)), alternatively it could be done on single trials. We have two reason to prefer our choice.

First of all, we should remember that EEG signals are extremely noisy. Therefore, over trial averaging, which improves the SNR, is more reliable for estimation of between class discrimination. On the contrary, if it is done on signal trials, we are more likely to start measuring discrimination between noise and noise.

Second, if we choose single trials, we should match one trial of the first class to some trial of the second class. This matching can be arbitrary. Thus we may choose different permutations, which may lead to different results. This is undesired feature for the stability of the algorithm.

#### 3.1.3.2 Total Variation

The Total Variation is measured in objective (3.6) on the single trials signals, which is preferable to over-trials average, which smoothes the signals, and doesn't give a real estimate of $TV$.

This claim can be demonstrated by the following example. Let's assume that all sensors contain pure signal, except for the sensor $\#k$, which registered desired signal plus some noise $n(t)$. If we choose to measure $TV$ on over-trials averaged signals, $X^1_{avg}$ and $X^2_{avg}$, and if $n(t)$ is zero mean, the SNR of the sensor $\#k$ will go to infinity as number of trials increases. Hence, no care of noisy sensor will be taken. However, if we choose to measure $TV$ on single

trail signals, the noise of sensor $\#k$ will influence the optimization procedure, and that sensor will receive a zero weight in spatial integration.

### 3.1.4 Getting Rid of the Tradeoff Parameter

In objective (3.6) there is a need to choose a value for a tradeoff parameter $\mu$. Although we have found out by our simulations, that the optimization result is quite robust to the change of value of $\mu$, the need to subjectively asses the tradeoff parameter is still an essential drawback. In this subsection, we rewrite the objective function in such a way, that it will contain no parameter any more.

For beginning, let us note, that the norm and the sign of the vector $w$ have no significance. We are interested only in *relative to each other* values of its elements. In other words, we want to find such $w$, which will satisfy two conditions. *First*, it will minimize a value of the second term of (3.6), when a value of the first term is constant. *Second*, it will minimize a value of the first term, when a value of the second term is constant. Now, let us write the objective function which will satisfy the above conditions[1]:

$$\begin{aligned} &\min_{w} \|Yw\|_1 \\ &s.t. \|X_{avg}w\|_2^2 = 1 \end{aligned} \tag{3.7}$$

One can note, that above objective function satisfies the first condition at the solution point by definition. The second condition is also satisfied. This can be proved in the following way. Suppose $w_{TV}$ is a solution of (3.7). Let us assume by contradiction, that there exist $w_{new}$, such that

[1] Alternatively, we may switch role of main term and constraint of objective (3.7)

$\|Yw_{new}\|_1 = \|Yw_{TV}\|_1$ and $\|X_{avg}w_{new}\|_2 = c^2 < 1$. In such a case, $w = \frac{1}{c}w_{new}$ would satisfy the constraint, while $\|Yw\|_1 < \|Yw_{TV}\|_1$. This contradicts the assumption, that $w_{TV}$ is a solution of (3.7). Thus, the second condition also holds.

Although the problems (3.6) and (3.7) are not completely equivalent, the problem (3.7) can be viewed as such, that optimally (and automatically) chooses the tradeoff parameter $\mu$. Indeed, if $w_\mu$ is a solution of (3.6) for some value of $\mu$ and $w_{TV}$ is a solution of (3.7), then, after re-scaling of $w_{TV}$, we have proven that: $\|Yw_{TV}\|_1 < \|Yw_\mu\|_1$ if $\|X_{avg}w_{TV}\|_2^2 = \|X_{avg}w_\mu\|_2^2$ and $\|X_{avg}w_{TV}\|_2^2 > \|X_{avg}w_\mu\|_2^2$ if $\|Yw_{TV}\|_1 = \|Yw_\mu\|_1$. This is true for any value of $\mu$, thus $w_{TV}$ is really the optimal solution. In Section 3.1.5 we provide derivatives of objective function (3.7), which will help to minimize it using numeric optimization.

An alternative to objective (3.7) can be obtained from Basis Pursuit perspective [31]. The matrix $Y$ in (3.7) may contain the coefficients of signal representation in some basis or "overcomplete" dictionary $\Psi$ (i.e. short-time Fourier transform, wavelet transform, etc.), which is expected to be sparse: $Y = \Psi^T X$ (see for example [56]). Additional alternative, is to build the matrix $Y$ from signals at predefined time windows, where their energy is expected to be small.

### 3.1.5 Numerical Optimization

Since the objective function (3.7) is a constrained optimization problem, it is convenient to minimize it by Lagrange Multipliers technique. For this purpose, we will need to calculate the gradient of the main and the constraint

terms.

Before we proceed, we should notice, that the main term of (3.7) is not differentiable. Hence, for optimization, we will use the smooth approximation of absolute value function.

$$\psi(t) = c(\frac{|t|}{c} - \log(1 + \frac{|t|}{c})) \tag{3.8}$$

Note that $\psi'(t)$ is defined at $t = 0$:

$$\psi'(t) = \frac{t}{c + |t|} \tag{3.9}$$

The approximation becomes more accurate, when $c \to 0$.

The modified problem will receive the following form:

$$\begin{aligned} &\min_{w} \sum_{l=1}^{L} 1^T \psi\left(Y_l^1 w\right) + \sum_{m=1}^{M} 1^T \psi\left(Y_m^2 w\right) \\ &s.t. \left\|X_{avg}^1 w - X_{avg}^2 w\right\|_2 = 1 \end{aligned} \tag{3.10}$$

where 1 is a vector of ones, and the application of $\psi(\cdot)$ to a vector is element-wise.

Let's denote the the main term of (3.10) as $f(w)$, and rewrite the constraint to be

$$g(w) = \|X_{avg} w\|_2^2 = w^T X_{avg}^T X_{avg} w$$

where $X_{avg} = X_{avg}^1 - X_{avg}^2$.

Now, we can easily calculate the gradients of $f(w)$ and $g(w)$, using the matrix derivations and the chain rules:

$$\nabla f(w) = \sum_{l=1}^{L} \left(Y_l^1\right)^T \psi'\left(Y_l^1 w\right) + \sum_{m=1}^{M} \left(Y_m^2\right)^T \psi'\left(Y_m^2 w\right) \tag{3.11}$$

$$\nabla g(w) = X_{avg}^T X_{avg} w \tag{3.12}$$

This calculus is sufficient for minimization of the objective function (3.10).

## 3.2 Learning Spatial Integration weights through Eigenvalue Decomposition

In this section, we propose a solution of a problem, which is similar somehow to (3.7):

$$\max_{x} \|Ax\|_2^2 \tag{3.13}$$
$$s.t. \|x\|_2^2 = 1$$

where $A$ is matrix, and $x$ is a vector. If we rewrite the main term as $\|Ax\|_2^2 = x^T A^T Ax = x^T \left(A^T A\right) x$, then it can be easily seen, that a solution for above problem is an eigenvector, which corresponds to the largest eigenvalue of matrix $B = A^T A$.

Now, let us return to the problem (3.7), more precisely to its alternative[2] :

$$\max_{w} \|X_{avg} w\|_2^2 \tag{3.14}$$
$$s.t. \|Yw\|_2^2 = 1$$

We can produce change of variables in (3.14) in order to make it look like (3.13). Let us rewrite the constraint term: $\|Yw\|_2^2 = w^T Y^T Yw$. If matrix Y is a full rank (which is very likely for the noisy data), the matrix $C = Y^T Y$ has a Cholesky factorization[3] $C = U^T U$. Now, the constraint

[2]Note, that objective (3.14) is quadratic, quadratically constrained. A $\Phi(\hat{s})$ term also appears with a $l_2$ norm, and thus can not represent *Total Variation* any more.

[3]If matrix $Y$ is not full rank, we may use a regularization $C = Y^T Y + \alpha I$, where $\alpha$ is a small constant, and $I$ is an identity matrix

can be written as $\|Yw\|_2^2 = w^T Y^T Y w = w^T U^T U w$. If we introduce a new variable $x = Uw$ ($w = U^{-1}x$), then the objective (3.14) can be written as:

$$\begin{aligned} &\max_x \left\| X_{avg} U^{-1} x \right\|_2^2 \\ &s.t. \, \|x\|_2^2 = 1 \end{aligned} \tag{3.15}$$

which exactly resembles the problem (3.13). Thus we know, that solution of (3.15) is an eigenvector $\nu_{max}$, which corresponds to the largest eigenvalue, $\lambda_{max}$, of matrix $B = \left(X_{avg} U^{-1}\right)^T X_{avg} U^{-1} = U^{-1T} X_{avg}^T X_{avg} U^{-1}$. Hence, the solution of (3.14) is $w = U^{-1}\nu_{max}$.

An important advantage of this approach, is that it doesn't require iterative optimization, but needs only a few simple algebraic steps. However, the difference of objective (3.14) is that in the $\Phi(\hat{s})$ term, the $l_1$ norm was replaced by $l_2$ norm. Thus it doesn't measure *Total Variation* any more. Nonetheless, our simulations shows, that using approach described in this subsection, we achieve similar results, with comparison to those, achieved using objective (3.14).

## 3.3 Theoretically Best Spatial Filtering

In some (unpractical) situations, we can evaluate the bound for the best theoretically achievable separations of signal from noise, using spatial filtering. This can be done if the 'sensor measurement' data is synthesized in the following way:

$$X = s_{art} a^T + N \tag{3.16}$$

In above formula, the 'sensor measurement' $T \times S$ matrix $X$ is obtained by mixing artificial signal, represented in $T \times 1$ vector $s_{art}$ with $S \times 1$ coupling vector $a$, and and adding $T \times S$ noise matrix $N$.

If we know both background noise covariance matrix $N$ and the mixing vector $a$ in (3.16), we can find the best weighting vector $w$ solving the following problem[4]:

$$\begin{aligned} &\min_w \|Nw\|_2^2 = w^T R w \\ &s.t. \left\|s_{art} a^T w\right\|_2^2 = 1 \end{aligned} \tag{3.17}$$

where $R = N^T N$ is the noise covariance matrix.

The artificial signal $s_{art}$ can be normalized $\|s_{art}\|_2 = 1$, hence the constraint in (3.17) can be simplified:

$$\begin{aligned} &\min_w w^T R w \\ &s.t.\ a^T w = 1 \end{aligned} \tag{3.18}$$

Note, that in above equation, we want to find $w$, which maximally suppress the noise, while keeping the norm of unmixing vector constant. And one can state, that this task differs from the task of the objective function (3.10). But actually in both cases we want to perform the de-noising of the signal of interest. In this sense, the solution of (3.18) can be viewed as theoretically best achievable limit and thus a good reference point for comparison.

We can solve the problem (3.18) using Lagrange Multipliers:

$$\min_w w^T R w - \lambda \left(a^T w - 1\right)$$

[4]Note, that we want to perform the de-noising by weighted sum of channels. Thus, the idea of subtracting the noise matrix is not relevant.

The gradient of above objective is given by: $g(w) = Rw - \lambda a^T$. The solution is obtained if $g(w) = 0 \Rightarrow w_{th} = \lambda R^{-1} a$. Note, that $w_{th}$ can be found up to scaling and sign, hence every real $\lambda \neq 0$ can be chosen. If we take $\lambda = 1$ we get the final formula for $w_{th}$:

$$w_{th} = R^{-1} a \tag{3.19}$$

## 3.4 *Time-Domain* Filtering

Signals, reconstructed by one of the spatial filtering methods, will still suffer from noise contamination, which can be further reduced by the second stage of preprocessing - *time-domain* filtering of the estimated signals (3.2).

The problem in applying filtering is that we do not know in advance which filter to use, because the signals of interest as well as background activity noise are unknown. Thus, we propose to find a suitable filter by learning, based on the same criteria used for finding spatial filter: maximize between class discrimination, while keeping the resulting signal smooth (or, alternatively, small in predefined time windows).

So, we want to find filter $h[n]$, $1 \leqslant n \leqslant N_{filt}$, which will further discriminate between reconstructed signals $\hat{s}^1_{avg}[n]$ and $\hat{s}^2_{avg}[n]$:

$$\begin{aligned} &\max_{h[n]} \left\| \left( \hat{s}^1_{avg}[n] - \hat{s}^2_{avg}[n] \right) * h[n] \right\|_2^2 \\ &s.t. \sum_{l=1}^{L} \left\| \hat{z}^1_l * h[n] \right\|_2 + \sum_{m=1}^{M} \left\| \hat{z}^2_m * h[n] \right\|_2 = 1 \end{aligned} \tag{3.20}$$

where $*$ denotes convolution.

Since, we are working with discrete time, time-limited signals, let us build $(T - N_{filt} + 1) \times N_{filt}$ convolution matrix $\tilde{X}^1_{avg}$, $j$-th column of which will

contain $\hat{s}^1_{avg}[n]$, $j \leqslant n \leqslant (T - N_{filt} + j)$, (i.e. the $j$-th column of $\tilde{X}$ contains a shifted by $(j-1)$ signal $\hat{s}^1_{avg}[n]$, which is also truncated by $(j-1)$ taps at the beginning and $(N_{filt} - j)$ taps at the end). In the same manner we can define convolution matrix $\tilde{X}^2_{avg}$, columns of which will contain shifted replicas of $\hat{s}^2_{avg}[n]$. And finally, matrix $\tilde{X}_{avg} = \tilde{X}^1_{avg} - \tilde{X}^2_{avg}$.

Similarly, we can build convolution matrices $\tilde{X}^1_l$,$\tilde{X}^2_m$ for single trials and construct from them matrix $\tilde{Y}$ in the same manner as matrix $Y$ is constructed from $X^1_l$,$X^2_m$ in (3.6).

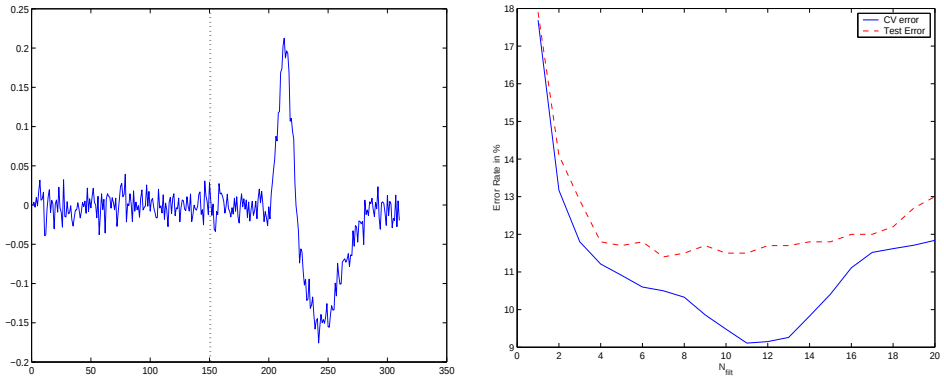
After those preparations, we can rewrite the problem (3.20) in an familiar form:

$$\begin{aligned} &\max_{w_{filt}} \left\| \tilde{X}_{avg} w_{filt} \right\|_2^2 \\ &s.t. \left\| \tilde{Y} w_{filt} \right\|_2^2 = 1 \end{aligned} \tag{3.21}$$

which exactly resemble the problem (3.14). Thus, the solution developed in section 3.2 can be used to solve the problem (3.21). Note, that the solution of (3.21), $w_{filt}$, is actually temporal filter, rather than spatial one.

There is an alternative choice of the matrix $\tilde{Y}$. It can represent a background activity noise[5], which we also want to minimize, instead of Total Variation. This idea may be even more appealing, because we minimize the background noise directly, and not some measure of it - Total Variation. This is really a good approach, if the background activity is stationary. Our simulations shows, that an alternative choice of the matrix $\tilde{Y}$ is better for real EEG recordings, while on the synthetic data, we have preferred the initial

[5]it can be obtained, for example, if we start to register the response well in advance (Fig. 3.1(a)).




(a) (b)

Figure 3.1: (a) Illustration of (single-channel) EEG recording, which contains a background activity only. If we start to register the response well in advance, then at the beginning we will record a background activity only. Signal left to the dashed vertical line can be treated as a background noise. The actual response appear after the dashed vertical line. (b)Cross Validation error rate for different values of $N_{filt}$. One can notice, that real test error (dashed line) is highly correlated with CV error. This enables us to choose the optimal order of FIR filter.

choice.

Now, the only open question left, is how to choose the optimal order $N_{filt}$ of FIR filter $h[n]$. We have no closed solution for this issue. In our experiments, we have chosen its value based on cross validation on the training data: we have calculated the $CV$ error for different values of $N_{filt}$, and then have chosen the one, which gives the lowest error rate. Figure 3.1(b) illustrates, that the $CV$ error rate and test error rate are highly correlated.

## 3.5 Computational Experiments

We have conducted several experiments, using both synthetic and real signals in order to show the feasibility of the proposed approaches and make a comparison between them and other spatial integration methods: CSP [44] and CSSP [45]. Also the comparison to direct use of unprocessed data, namely simple sum of channels ("$\sum$") and choosing the best channel ("best ch."), was also performed to emphasize the contribution of proper spatial integration. In this section we provide a results of our simulations.

### 3.5.1 Real EEG Signals

In this experiment, we used the data obtained from two BCI competitions, held in years 2002 [57] and 2003 [58]. The goal of above competitions was to validate signal processing and classification methods for Brain Computer Interface. Data set consists of trials of spontaneous EEG activity, one part labelled (training data) and another part unlabelled (test data). The goal was to infer labels for the test set, by prepossessing the training data. Inferred

test labels should have maximally fit the true (but unknown to participants) test labels.

#### 3.5.1.1 BCI competition 2002

This data set [57], consists of EEG signals, that were recorded from one subject in sessions with few minute's breaks in between. The subject was sitting in a normal chair, relaxed arms resting on the table, fingers in a standard typing position at the computer keyboard (index fingers at 'f', 'j' and little fingers at 'a',';'). The task was to press two chosen keys with the corresponding fingers in a self-chosen order and timing ('self-paced key typing'). A total of 516 keystrokes was done at an average speed of 1 key every 2.1 seconds. Brain activity was measured with 27 Ag/AgCl electrodes at 1000 Hz sampling rate using a band-pass filter from 0.05 to 200 Hz.

Further, windows 1500 ms long were cut out of the continuous raw signals each ending at 120 ms **before** the respective keystroke. The reason for choosing the endpoint at $-120$ ms is that before this point the classification based on measuring EMG activity only is still close to chance. 100 trials equally spaced over the whole experiment were defined to be the test set, leaving 413 labelled trials for training. For classification we used only last 700 ms of a trail, the first 800 ms were treated as background noise used in calculation of $w_{filt}$.

We used only the training data for learning spatial and temporal filters by our methods described in previous sections - $w_{TV}$, $w_{EVD}$ and $w_{TV} + w_{filt}$. Then, we have classified the filtered test data. Figure 3.2 illustrates the classification process. We have tried several classifiers for classification us-

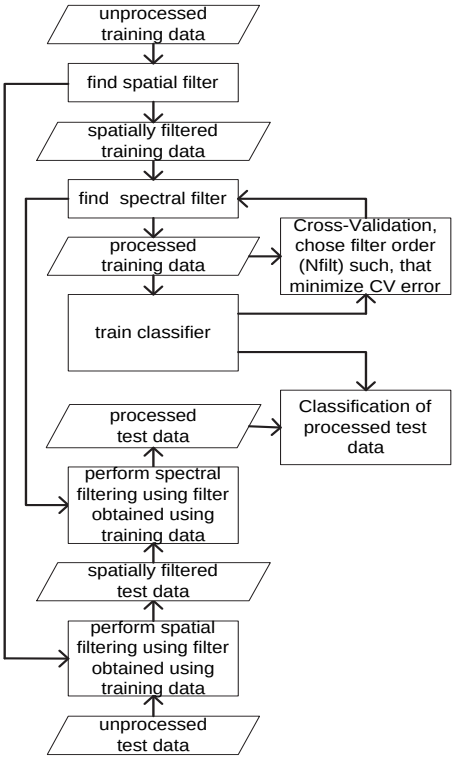



Figure 3.2: Flow chart of classification process. The temporal filtering stage is omitted for $w_{TV}$ and $w_{EVD}$ methods.

| | CSP | CSSP | $w_{TV}$ | $w_{EVD}$ | $w_{TV}+w_{filt}$ | $\sum$ | best ch. |
|---|---|---|---|---|---|---|---|
| 10-fold CV | 21(4.1) | 12(3.3) | 12(3.8) | 12(3.5) | 13(3.9) | 51(11) | 32(10) |
| Test | 27 | 21 | 5 | 5 | 5 | 53 | 30 |

Table 3.1: Classification results (error rate in %) of BCI competition 2002 data set. Each column corresponds to a different method of preprocessing. Both*10-fold Cross-Validation*(std is given in parenthesis) and test error results are provided. For CSP method we used $m = 4$, for CSSP $m = 8$,$\tau = 16$, $N_{filt} = 100$ was used for proposed $w_{TV} + w_{filt}$ method. All these parameters were chosen such that minimize cross-validation error.

ing "pr-tools" classification toolbox [59]), including nearest mean, k nearest neighbor (k-nn) [60] and Support Vector Machines (SVM) with different kernels [61]. All classifiers have provided similar results with BCI2002/3 data sets, therefore we have preferred to use the simplest one - nearest mean classifier. The result of classification error, both of *10-fold Cross-Validation* [62] and test error[6], are summarized in Table 3.1. The best result reported by competition organizers was 4% error rate. The classification results using for CSP and CSSP methods using Fisher Linear Discriminant are 27% and 21% test error rate respectively (are similar to Nearest-Mean classification results).

#### 3.5.1.2 BCI competition 2003

The data set for that competition [58] is similar to the previous one (self-paced key typing). This time the average typing rate was 1 key per second.

[6]Test error was calculated when real labels were published by competition organizers.

| | CSP | CSSP | $w_{TV}$ | $w_{EVD}$ | $w_{TV} + w_{filt}$ | $\sum$ | best ch. |
|---|---|---|---|---|---|---|---|
| 10-fold CV | 18(4.9) | 10(4.9) | 21(4) | 24(4.3) | 22(4) | 43(14) | 37(9) |
| Test | 30 | 22 | 22 | 22 | 22 | 41 | 39 |

Table 3.2: Classification results (error rate in %) of BCI competition 2003 data set. Each column corresponds to a different method of preprocessing. Both*10-fold Cross-Validation* (std is given in parenthesis) and test error results are provided. For CSP method we used $m = 6$, for CSSP $m = 6$,$\tau = 22$, $N_{filt} = 110$ was used for proposed $w_{TV}$+$w_{filt}$ method. All these parameters were chosen such that minimize cross-validation error.

Totally, there are 416 epochs of 500 ms length each ending 130 ms before a key press. 316 epochs are labelled (training set), the remaining 100 epoches are unlabelled (test set).For classification we used only last 360 ms of data, the first 140 ms were treated as background noise which were used in calculation of $w_{filt}$.

We used the training data for preprocessing, trying out all proposed approaches - $w_{TV}$, $w_{EVD}$ and $w_{TV}$+$w_{filt}$. Again, we used nearest mean classifier for classification. The result of classification error of *10-fold Cross-Validation* and test error are summarized in Table 3.2. The best result reported by competition organizers was error rate of 16%. The classification results using for CSP and CSSP methods using Fisher Linear Discriminant are 29% and 19% test error rate respectively (are similar to Nearest-Mean classification results).

#### 3.5.1.3 Response to Visual Stimuli

This EEG data was obtained during the following procedure[7]. The subject was shown a sequence of 3 different images, at some predefined and constant over-trials pace. Afterwards, the subject had to respond, by pressing a button. The trials were repeated with periodicity of 7 seconds. The delay between the first and the second images in the sequence was 1.5 seconds, and 2.5 seconds between the second and the third image. Then, the subject was given 3 seconds for respond. Each session consisted of approximately 30 trials. There were several sessions, with few minutes break between them. The EEG data was recorded by 23 electrodes, with sampling rate of 256 samples per second.

In this experiment, we were interested in distinguishing a response to visual stimuli from the absence of response, i.e. regular background activity. We have built two classes of signals from the row data: the first class represented the response to visual stimuli (image was shown), and the second class represented the absence of visual stimuli (regular background activity). In order to build the first class, we have cut from the row data the segments, which start at the times when the first image is shown and are 180 time samples in length. The second class was built from segments, started 300 time samples before the third image is shown, and ended 120 time samples

[7]For this experiment we have used the data, that was recorded in the laboratory for Evoked Potentials in the Technion - Israel Institute of Technology. We have not conducted the new experiment for our purposes, but rather we have used the data that were already recorded for some other research [63]. We are grateful to Hillel Pratt for providing us with these EEG recordings.

| | CSP | CSSP | $w_{TV}$ | $w_{EVD}$ | $w_{TV}$+$w_{filt}$ | $\sum$ | best ch. |
|---|---|---|---|---|---|---|---|
| NM | 8.9(2.1) | 5.8(1.5) | 2.2(0.7) | 2.2(0.6) | 2.3(0.6) | 39(5.5) | 26(4.5) |
| k-nn | 8.3(2) | 5.6(1.4) | 2.2(0.7) | 1.7(0.6) | 3.3(0.7) | 27(6) | 16(5) |
| FLD/SVM | 7.3(1.8) | 3.9(1.7) | 2.2(0.6) | 2.6(0.9) | 2.9(0.8) | 41(13) | 29(14) |

Table 3.3: *10-fold Cross-Validation* error rate in % (std is given in parenthesis) on Visual Stimuli data set. Each column corresponds to a different methods of preprocessing. Results of applying 3 different classifiers are shown: Nearest Mean (NM), k-nn (with k=3) and FLD/SVM with exponential kernel. Fisher Linear Discriminant (FLD) is applied on CSP and CSSP methods as proposed in original papers. Since FLD is not applicable if signals of both classes are zero-mean, in remaining methods we use SVM instead. For CSP method we used $m = 4$, for CSSP $m = 6$,$\tau = 12$, $N_{filt} = 40$ was used for proposed $w_{TV}$+$w_{filt}$ method. All these parameters were chosen such that minimize cross-validation error. The same parameters were used later in the test stage.

before the time when the third image is displayed.

We have randomly chosen 180 trials to be the training set and the remaining 60 trials were made to be the test set. Then, we have applied all our approaches. We provide the classification results by NM, k-NN and SVM with exponential kernel classifiers demonstrated different performance. The results of 10-fold cross validation are summarized in the Table 3.3. The test error is provided in Table 3.4. Reconstructed signals are shown in Figure 3.3.

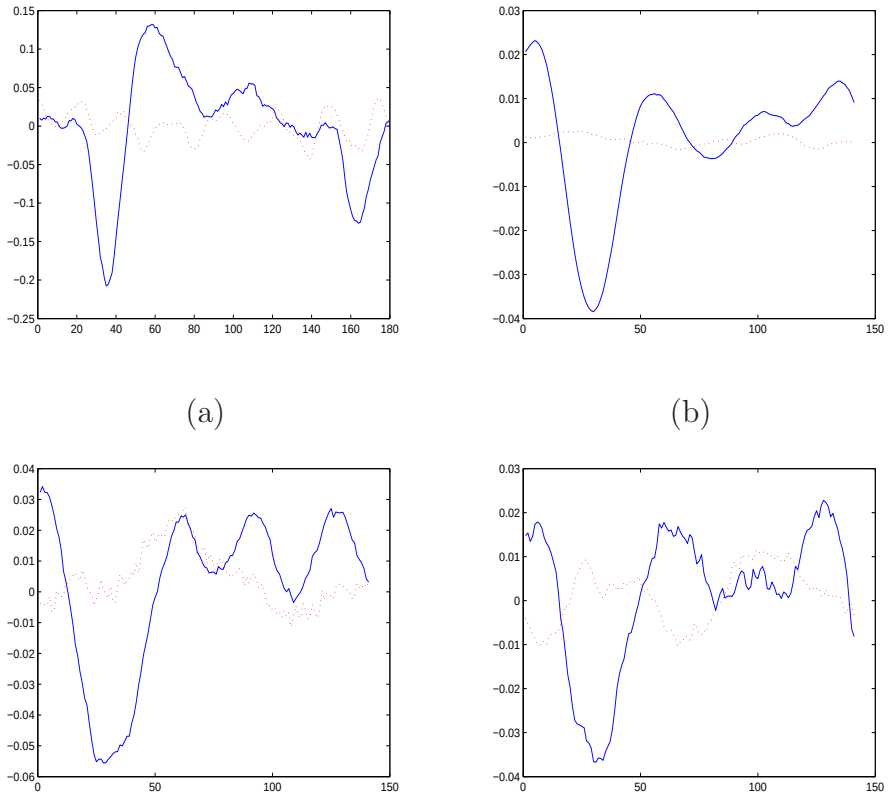



Figure 3.3: Experiment with visual stimuli data set. One can easily tell, that the dotted line represents the signal with background activity only, while the solid line corresponds to the signal which contains the response. (a) - averaged signals reconstructed by $w_{TV}$; (b) - averaged signals reconstructed by $w_{TV}$ and further filtered by $w_{filt}$; (c),(d) - examples of single-trial signals reconstructed by $w_{TV}$ and further filtered by $w_{filt}$. Note, that two classes are easily distinguishable even on single trials. Also note the similarity between single-trial and averaged signals

| | CSP | CSSP | $w_{TV}$ | $w_{EVD}$ | $w_{TV}$+$w_{filt}$ | $\sum$ | best ch. |
|---|---|---|---|---|---|---|---|
| NM | 3.3 | 3.3 | 3.3 | 1.7 | 0.0 | 40 | 35 |
| k-nn | 10 | 8.3 | 0.0 | 1.7 | 3.3 | 35 | 20 |
| FLD/SVM | 6.7 | 5 | 1.7 | 1.7 | 3.3 | 45 | 42 |

Table 3.4: Test error rate (in %) on Visual Stimuli data set. Each column corresponds to a different methods of preprocessing. Results of applying 3 different classifiers are shown: Nearest Mean (NM), k-nn (with k=3) and FLD/SVM with exponential kernel. See table 3.3 for further details

### 3.5.2 Artificial Signals

We decided not to stop the evaluation of our approach after two previous experiments, and generated synthetic data set, which should resemble the real experiments. The idea was to mix some smooth signal (two different signals, $s^1_{art}$ and $s^2_{art}$, for two different classes) into background noise $N$.

We modelled our signals to be $T$ time samples in length. Thus, artificial signals, $s^1_{art}$ and $s^2_{art}$, are $T \times 1$ vectors. Moreover, we assumed $S$ channel data, hence the dimensions of single trial and noise matrices ($X_i$ and $N_i$ respectfully) are $T \times S$. The weights, with which the artificial signal $s_{art}$ reaches each channel (column of $X_i$), were randomly chosen and organized in $S \times 1$ *mixing vector* $a$. This leads to the following data synthesis model:

$$\begin{aligned} X^1_l &= s^1_{art} a^T + N_l \\ X^2_m &= s^2_{art} a^T + N_m \end{aligned} \tag{3.22}$$

#### 3.5.2.1 White Gaussian Background Noise

In this experiment, the noise matrix $N_i$ was generated as white gaussian noise ($150 \times 25$ matrix, generated independently for each trial). The mixing vector $a$ was randomly generated (each element in $a$ is uniformly distributed between $[-1; 1]$). This experiment setup stands up for "real" problem of 25 sensors and 150 time samples in each trial. We have generated 1200 trials. First 200 trials (approximately 100 of each class) were used for preprocessing (finding unmixing vector $w$). Remaining 1000 trials were used for classification by the nearest mean algorithm [62].

We have compared results of classification of data preprocessed by different methods - $w_{TV}$, $w_{EVD}$ and $w_{filt}$ and unprocessed data (simple sum over all channels "$\sum$" and choosing the best channels "best ch."). In addition, in experiments with the synthesized data, we have the good reference point for comparison - results obtained by applying $w_{th}$ (3.19). As shown in Section 3.3, this method serves as an upper bound of signal denoising by spatial filtering. Classification results (by Nearest Mean Classifier) are displayed in the Table 3.5. Three rows refer to three different SNR[8].

#### 3.5.2.2 Real EEG Background Noise

In the second experiment, we have used real EEG signals as the background noise $N$ ($150 \times 22$ matrix for each trial). The rest of the setup is identical to the previous case. Classification results are displayed in the Table 3.6.

[8]SNR refer to average signal-to-noise ratio at each sensor in single trial. Since mixing weights $a_i$ of artificial signal $s_{art}$ are randomly generated, we have taken the average value of $a_i = 0.5$

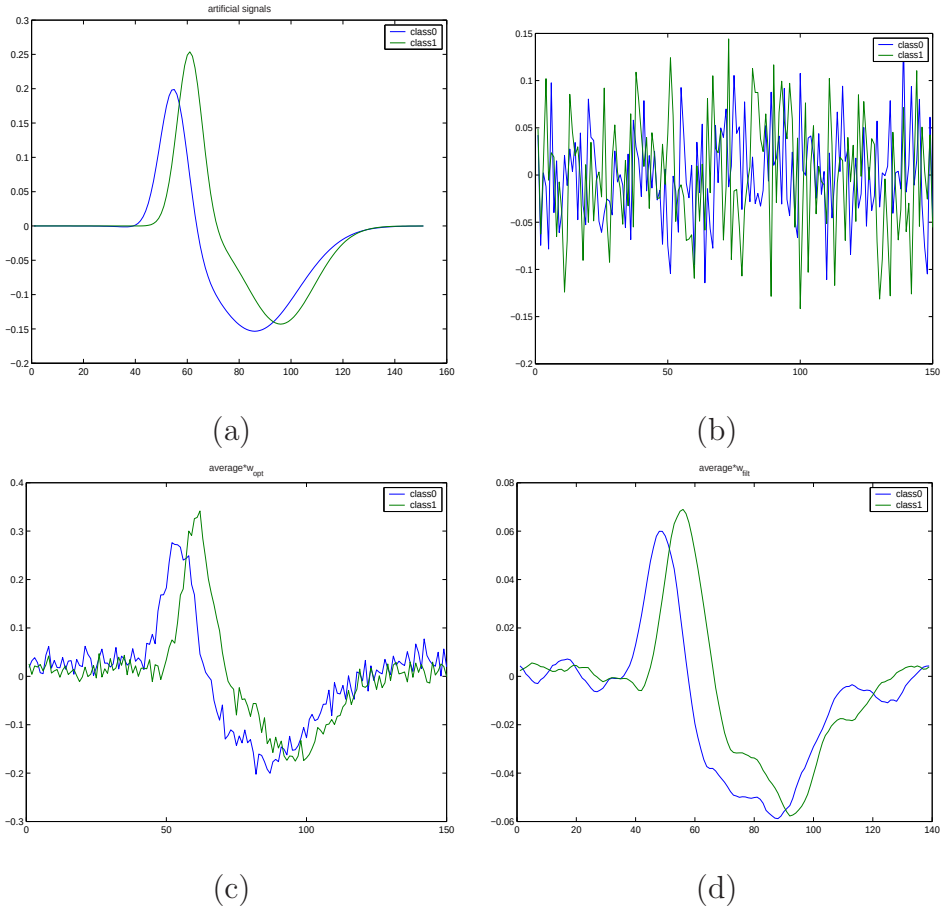



Figure 3.4: Experiment with artificial signals and Random Gaussian background noise: (a) - artificial signals; (b) - signals restored by simple sum of channels; (c) - signals restored by $w_{TV}$; (d) - signals restored by $w_{TV}$ and further filtered by $w_{filt}$.

| | $w_{TV}$ | $w_{EVD}$ | $w_{TV} + w_{filt}$ | $w_{th}$ | $\sum$ | best ch. |
|---|---|---|---|---|---|---|
| SNR=-10dB | 0.0% | 0.0% | 0.0% | 0.0% | 43.7% | 14.0% |
| SNR=-15dB | 3.5% | 3.3% | 1.4% | 2.7% | 44.1% | 36.7% |
| SNR=-20dB | 20.3% | 20.4% | 13.7% | 17.1% | 49.0% | 41.7% |

Table 3.5: Experiment with artificial signals and White Gaussian background noise: Classification Error Rate in % . The first column shows an average SNR, measured at each sensor.

| | $w_{TV}$ | $w_{EVD}$ | $w_{TV} + w_{filt}$ | $w_{th}$ | $\sum$ | best ch. |
|---|---|---|---|---|---|---|
| SNR=-20dB | 2.5% | 1.3% | 1.1% | 0.0% | 51.6% | 43.3% |
| SNR=-25dB | 10.5% | 10.3% | 10.0% | 0.7% | 50.5% | 50.1% |
| SNR=-30dB | 25.7% | 22.6% | 21.9% | 1.3% | 52.9% | 49.5% |

Table 3.6: Experiment with artificial signals and real EEG recordings as a background noise: Classification Error Rate in %. The first column shows an average SNR, measured at each sensor.

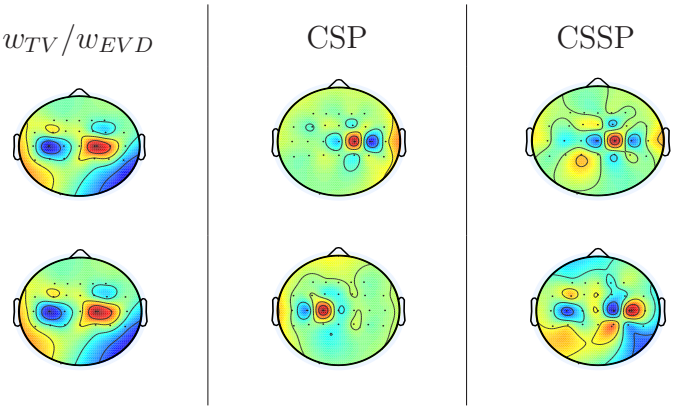



Figure 3.5: Imagined hand movement data set (BCI competition 2002). Spatial filters received by $w_{TV}$,$w_{EVD}$ (left column) and CSP,CSSP methods (two patterns per each). Spatial filters received by $w_{TV}$,$w_{EVD}$ are almost identical. Spatial filters received using CSP,CSSP are different, but concentrate over the same area

## 3.6 Discussion

Classification results presented in previous chapter leave no doubt that spatial integration improves classification accuracy. An interesting issue is a comparison of the spatial filters obtained by proposed $w_{TV}$ and $w_{EVD}$ techniques with spatial patterns obtained by CSP and CSSP methods.

Figures 3.5 and 3.6 show the spatial patterns obtained by different methods on imagined hand movement data sets. One can notice, that $w_{TV}$ and $w_{EVD}$ are similar in both data sets. Spatial filters received by CSP and CSSP methods are different, however they concentrate on the same area. If we take into account the classification results on these data sets (which for proposed methods are at least as good as for CSP and CSSP methods) it hints that $w_{TV}/w_{EVD}$ were able to concentrate all relevant spatial information in one

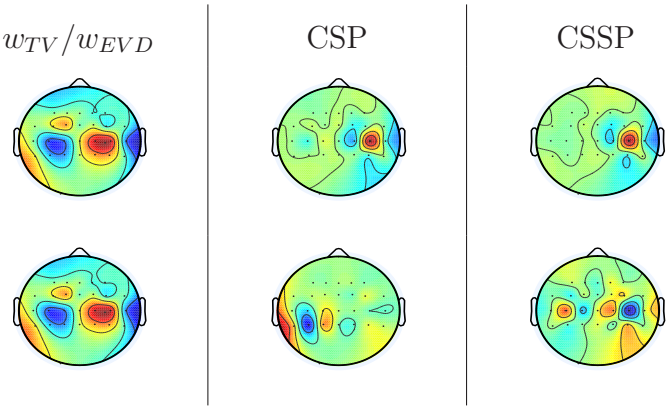



Figure 3.6: Imagined hand movement data set (BCI competition 2003). Spatial filters received by $w_{TV}$,$w_{EVD}$ (left column) and CSP,CSSP methods (two patterns per each). Spatial filters received by $w_{TV}$,$w_{EVD}$ are almost identical. Spatial filters received using CSP,CSSP are different, but concentrate over the same area. Note the similarity to filters in figure 3.5

single filter. It may also hint, that after reconstructing the signal using spatial filter, a better strategy for classification is using it time course, rather than some measure of it - variance as proposed by CSP/CSSP methods. Indeed, the response to two different tasks may have different time courses, while having the same variance.

The spatial filters for the visual stimuli data set, can be observed in Figure 3.7. This time, all the methods have produced very similar filters[9]. If we look at the classification results and compare $w_{TV}/w_{EVD}$ vs. CSP and $w_{TV} + w_{filt}$ vs. CSSP[10], we can observe that proposed methods have similar (or even better) performance. Thus, we can again argue, that $w_{TV}/w_{EVD}$

[9]for CSP/CSSP methods we have chosen the most similar filter, among first $2m$ which participate in classification

[10]this comparison is the most fair since both $w_{TV}+w_{filt}$ and CSSP use spatial-temporal filtering

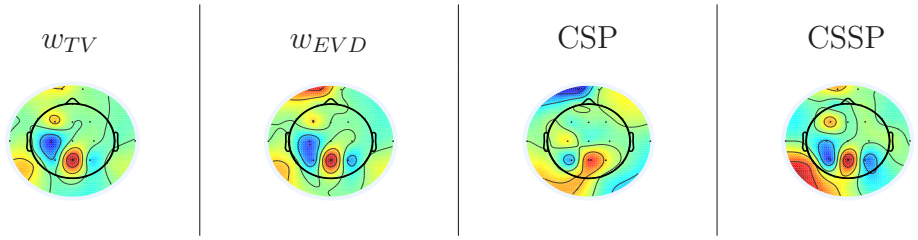
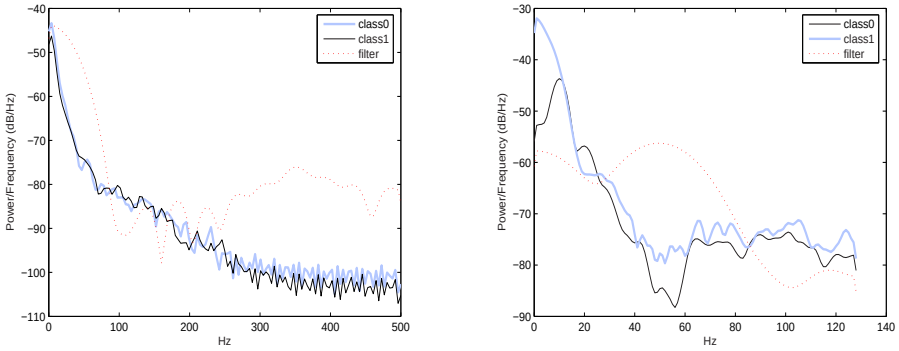



Figure 3.7: Visual stimuli data set. Spatial filters received by $w_{TV}$,$w_{EVD}$ (left column) and CSP,CSSP methods. Note the similarity of all spatial filters.




(a) (b)

Figure 3.8: Power spectra of estimated signals (solid lines) and resulting temporal filter $w_{filt}(dashedline)$: (a) - imagined hand movement data set (BCI competition 2003); (b) - response to visual stimuli data set.

contains the most relevant spatial information in a single filter.

Another interesting issue for discussion is the contribution of temporal filtering to the classification accuracy. One can notice, that in imagined hand movement data sets, temporal filter did not contribute to the classification accuracy. However, on visual data set it has improved the classification considerably.

This phenomena is easy to explain. In the imagined hand movement data

sets, signals of both classes correspond to the movement task and has similar power spectra (Fig. 3.8(a)). In this case, a resulting temporal filter turns to be a simple low-path which doesn't contribute to the further discrimination of classes (it may, however, improve the quality of estimation by rejecting high-frequency noise).

In the visual data set, signals of different classes has different origin: the first contains background activity, while the second is a response to the visual stimuli. As one can see on Fig. 3.8(b) those signals have different frequency components. In this case, a power spectra of resulting temporal filter has dominant peaks over frequency regions, were power spectra of two signals differ the most. Thus, such filtering contribute to the further between-class discrimination and classification improvement.

# Chapter 4

# Conclusion

Both algorithms presented in this thesis, have been shown by conducted simulations to overperform the existing techniques by exploring the sparsity of the signals in the first case, and the smoothness in the second.

## Sparse Multi-Channel Signal Reconstruction and Localization

The first algorithm, presented in this thesis, is a method of multiple wide-band source localization and reconstruction. It is designed for the case, when the sources can be sparsely represented in time domain in some know basis or frame. It improves the algorithm proposed in [30] by exploring the temporal sparsity along with spatial one. It was formulated as a regularized optimization problem. Thus it does not require accurate initialization and the convergence to global minima is guaranteed.

Conducted simulations demonstrate the advantage of enforcement of temporal sparsity along with spatial sparsity. In most cases, the proposed *SMSR* algorithm achieves lower side-lobe levels, than the method based on spatial sparsity only. In some cases, it is able to resolve sources, which the spatial sparsity only method fails to resolve.

In addition, the proposed enforcement of both temporal and spatial sparsity makes the proposed approach capable to accurately reconstruct the source signals even at high noise levels. Moreover, the proposed problem formulation allows to use the algorithm in the presence of multi-path. This requires, however, the knowledge of impulse responses for all location-sensor pairs, which may be unpractical for some reverberation environment.

The weaknesses of our approach are high computational complexity and the need to subjectively assess the trade-off parameters. These issues can serve as challenging topics for further research. The computational load may be reduced, for example, by introducing more efficient optimization procedure, which is tailored for the problem formulation (2.17) and a structure of the forward operator $\mathcal{A}$.

An automatic choice of trade-off parameters may be based on discrepancy principle: choose the trade-off parameter such, that the residual of the solution obtained using it will fit some known statistics of noise. The choice may be done iteratively. However, it is preferably to develop a closed solution. As a starting point one may use a procedure developed in [49] for a similar problem with single trade-off parameter.

# Learning Spatial and Spectral Filters for Single-Trial EEG Classification

We have presented a two-stage preprocessing algorithm. It extracts the desired response from multi-channel data, by means of spatial integration in the first stage, and time-domain filtering at the second stage. This preprocessing is essential for the classification. Our experiments shows, that the miss-classification rate achieved on the preprocessed data is significantly lower, than an error rate obtained by classifying unprocessed signals (simple sum of channels, best channel) as well as data preprocessed by CSP [44] and CSSP [45] methods.

In addition, we show in our simulations on synthetic data, that the error rate achieved after the first stage of preprocessing reaches (or is very close to) the lower bound developed for spatial integration methods. Moreover, if we apply the second stage of proposed algorithm - time-domain filtering, we receive the error rate even lower than an above bound.

The comparison of spatial filters obtained by the proposed method to those obtained by CSP and CSSP method shows, that all three filters concentrate in the same cortical area. Moreover, the proposed method has found almost identical spatial filters for two different data set, both containing EEG of imagined hand movement. This result illustrates that proposed method may be valuable for neurology, and not only for improving classification accuracy.

The application of proposed method for more data sets and exploring the resulting spatial and spectral filters is an exciting subject for further

research. In particular, it is of a great interest to explore how does a choice of mental tasks influence a classification performance, to compare spatial filters of different subjects for the same mental tasks, and finally, to observe an effect of feedback on spatial filters, when proposed method is used in online BCI system.

Another topic for further research may be an extension of proposed method for multi-class problem (more than 2 classes). This may be done, for example, by modifying the objective function (3.7) to contain a sum of all combinations of pair-wise signal differences.

# Appendix A

## A.1 Adjoint Operator

We want to calculate the *adjoint* operator $\mathcal{A}^*$ to the forward operator $\mathcal{A}$ defined in equations (2.7) and (2.9).

The adjoint operator can be defined using the inner product: suppose we have vectors[1] $x$ and $y$, then the adjoint operator is one, that satisfies the equation:

$$\langle y\mathcal{A}^*, x\rangle = \langle y, \mathcal{A}x\rangle \tag{A.1}$$

for all $x$ and $y$.

Now, we will prove that:

[1]Here we do not really distinguish between vectors and matrices, since it affects only the definition of inner product

*Proposition*

If the forward operator $\mathcal{A}$ is defined as:

$$x[n] = \mathcal{A}s[n] = h[n] * s[n] \tag{A.2}$$

where $*$ denotes convolution. Then, the adjoint operator $\mathcal{A}^*$ is defined as

$$y[n] = x[n]\mathcal{A}^* = h[-n] * x[n] \tag{A.3}$$

Note, that the only difference between forward and adjoint operators (A.2) and (A.3) is a "$-$" in the argument of $h[n]$.

*Proof*

In order to show, that equations (A.2) and (A.3) satisfy the definition of adjoint operator, we will calculate two inner products[2] $\langle y\mathcal{A}^*, x\rangle$ and $\langle y, \mathcal{A}x\rangle$, and show that they are equal for all $x$ and $y$.

$$\langle y, \mathcal{A}x\rangle = \sum_n y[n]\left(\mathcal{A}x\right)[n] = \sum_n y[n]\left(h * x\right)[n] \tag{A.4}$$

$$\begin{aligned}\langle y\mathcal{A}^*, x\rangle &= \sum_n \left(y\mathcal{A}^*\right)[n]x[n] = \sum_n \left(h[-m] * y[m]\right)[n]x[n] = \\ &= \sum_n x[n] \sum_m h[m-n]y[m] = \sum_m y[m] \sum_n x[n]h[m-n] = \\ &= \sum_m y[m]\left(h * x\right)[m] \end{aligned} \tag{A.5}$$

One can notice, that expressions (A.4) and (A.5) are equal for all $x$ and $y$. Thus (A.3) is adjoint operator for (A.2).

[2]here we use the ordinary Euclidean inner product $< a, b >= \sum_{ij} a_i b_j$

Now, it is straight forward to extend the proposition to the case, when the forward operator is defined as a *sum of convolutions*:

$$x[n] = \mathcal{A}s[n] = \sum_i h_i[n] * s[n] \tag{A.6}$$

In this case, the adjoint operator is given by:

$$y[n] = x[n]\mathcal{A}^* = \sum_i h_i[-n] * x[n] \tag{A.7}$$

The proof is pretty trivial: if we define $h_t \triangleq \sum_i h_i[n]$, then we receive the following definition of $\mathcal{A}$:

$$x[n] = \mathcal{A}s[n] = h_t[n] * s[n]$$

which by the first proposition gives the following definition of $\mathcal{A}^*$:

$$y[n] = x[n]\mathcal{A}^* = h_t[-n] * x[n] = \sum_i h_i[-n] * x[n]$$

which proves that operator (A.7) is adjoint to (A.6).

## A.2 Truncated Newton Method

Here we provide a description of *Truncated Newton* algorithm. For more details please refer to [51] and [52].

In the algorithm description we will use the following notations: $f(C)$ - the objective function. $G$ and $\mathcal{H}$ are the gradient and the Hessian of $f(C)$, respectively. The *Truncated Newton* method applied to the objective has the following iterative scheme:

1. Start with an initial estimate $C_0$ of source coefficients
2. For $k = 1, 2, ...$ until convergence
   (a) Compute the current direction $D_k$ by approximate solution of system of linear equations $\mathcal{H}D_k = -G_k$
   (b) Compute the step size $\alpha_k$ by exact or inexact line search: $\alpha_k = arg\min_\alpha f(C_k + \alpha D_k)$
   (c) $C_{k+1} = C_k + \alpha_k D_k$
3. End of loop

The step 2a is performed by the **preconditioned** *linear Conjugate Gradients*. We use the diagonal operator $\mathcal{W}$ for preconditioning. $\mathcal{W}$ has the same size and the diagonal as $\mathcal{H}$ - the Hessian of $f(C)$. Since $\mathcal{W}$ is diagonal, the calculation of $\mathcal{W}^{-1}$ is straightforward. Moreover, the optimization algorithm doesn't differ much from the regular *CG*:

1. Start with $D_0$, $R_0 = \mathcal{H}D_0 + G_k$, $\beta_0 = 0$, $P_0 = 0$

2. For $k = 1, 2, ...$

   (a) $P_k = -\mathcal{W}^{-1} R_k + \beta_{k-1} P_{k-1}$

   (b) $\gamma_k = \frac{\langle R_k, \mathcal{W}^{-1} R_k \rangle}{\langle P_k, \mathcal{H} P_k \rangle}$

   (c) $D_{k+1} = D_k + \gamma_k P_k$

   (d) $R_{k+1} = R_k + \gamma_k \mathcal{H} P_k$

   (e) $\beta_k = \frac{\langle R_{k+1}, \mathcal{W}^{-1} R_{k+1} \rangle}{\langle R_k, \mathcal{W}^{-1} R_k \rangle}$

3. End of loop

where $\langle A, B \rangle = Tr\left(A^T B\right) = \sum_{ij} a_{ij} b_{ij}$ is an inner product of two matrices $A$ and $B$.

Note, that we are not looking for the exact solution of step 2a of *Truncated Newton* algorithm. Hence, we should stop our *CG* algorithm when we are close enough to the solution. One of the stop criteria may be a fixed number of steps. Other possible criteria is when the $\frac{\|R_k\|_2}{\|R_0\|_2}$ is low enough - say $10^{-3}$.

# Bibliography


[1] T. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 1998.

[2] M. Skolnik, *Introduction to Radar Systems*. McGraw-Hill, 2000.

[3] R. Nielsen, *Sonar Signal Processing*. Artech House, 1991.

[4] E. Robinson and S. Treitel, *Geophysical Signal Analysis*. Prentice Hall, 1980.

[5] L. Parra, C. Spence, A. Gerson, and P. Sajda, "Recipes for the linear analysis of eeg," *Neuroimage*, vol. 28, pp. 326–341, 2005.

[6] J. Vidal, "Toward direct brain-computer communication," *Annu Rev Biophys Bioeng*, pp. 157–180, 1973.

[7] M. Wax and T. Kailath, "Optimum localization of multiple sources by passive arrays," in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1210–1218, Sept. 1983.

[8] A. Nehorai, G. Su, and M. Morf, "Estimation of time differences of arrival by pole decomposition," in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1478–1491, Dec. 1983.

[9] R. Schmidt, *A Signal Subspace Approach to Multiple Emmiter Location and Spectral Estimation*. PhD thesis, Stanford university, 1981.

[10] D. Johnson and D. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[11] J. Capon, "High-resolution frequency wavenumber spectrum analysis," in *Proc. IEEE*, vol. 8, pp. 1408–1418, 1969.

[12] J. E. Evans, J. R. Johnson, and D. F. Sun, "Application of advanced signal processing techniques to angle of arrival estimation in atc navigation and surveillance systems," tech. rep. 582, MIT Lincoln Laboratory, 1982.

[13] G. Su and M. Morf, "The signal subspace approach for multiple wide-band emitter location," in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1502–1522, Dec. 1983.

[14] H. Wang and M. Kaveh, "Coherent signal-subspace processing for the detection and estimation of angle of arrival of multiple wide-band sources," in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 823–831, Aug. 1985.

[15] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 9, pp. 2009–2025, Oct. 1998.

[16] P. Comon, "Independent component analysis: A new concept," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[17] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[18] D. Pham, P. Garrat, and C. Jutten, "Separation of a mixture of independent sources through a maximum likelihood approach," in *European Signal Processing Conference*, pp. 771–774, 1992.

[19] B. A. Pearlmutter and L. C. Parra, "Maximum likelihood blind source separation: A context-sensitive generalization of ICA," in *Advances in Neural Information Processing Systems 9*, MIT Press, 1997.

[20] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non gaussian signals," in *Proc. Inst. Elect.*, vol. 140, pp. 362–370, 1993.

[21] A. Hyvarinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, Oct. 1997.

[22] A. Hyvarinen, "The fixed-point algorithm and maximum-likelihood estimation of independent component analysis," *Neural Processing Letters*, vol. 10, no. 1, pp. 1–5, 1999.

[23] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 3017–3030, 1996.

[24] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computations*, vol. 13, no. 4, pp. 863–882, 2001.

[25] M. Zibulevsky, B. A. Pearlmutter, P. Bofill, and P. Kisilev, "Blind source separation by sparse decomposition," in *Independent Components Analysis: Princeiples and Practice* (S. J. Roberts and R. M. Everson, eds.), Cambridge University Press, 2001.

[26] J.-F. Cardoso, H. Snoussi, J. Delabrouille, and G. Patanchon, "Blind separation of noisy Gaussian stationary sources. Application to cosmic microwave background imaging," in *Proc. EUSIPCO*, vol. 1, pp. 561–564, 2002.

[27] K. Torkkola, "Blind separation of convolved sources based on information maximization," in *Neural Networks for Signal Processing VI*, (Kyoto, Japan), IEEE Press, Sept. 4–6 1996.

[28] P. Smaragdis, "Information theoretic approaches to source separation," Master's thesis, MIT Media Lab., 1997.

[29] N. Mitianoudis and M. Davies, "Audio source separation of convolutive mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 11, pp. 489–497, Sept. 2003.

[30] D. M. Malioutov, M. Çetin, and A. S. Willsky, "Sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, pp. 3010–3022, Aug. 2005.

[31] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[32] H. Berger, "Uber das electrenkephalogramm des Menschen," *Arch Psychiat Nervenkr*, vol. 87, pp. 527–570, 1929.

[33] J. Kalcher, D. Flotzinger, C. Neuper, S. Golly, and G. Pfurtscheller, "Graz brain-computer interface ii: Toward communication between humans and computers based on online classification of three different EEG patterns," *Med Biol Eng Comp*, vol. 34, pp. 383–388, 1996.

[34] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer, "EEG-based discrimination between imagination of right and left hand movement," *Electroenceph. clin. Neurophysiology*, vol. 103, no. 6, pp. 642–651, 1997.

[35] C. Anderson, E. Stolz, and S. Shamsunder, "Multivariate autoregressive models for classification of spontaneous electroencephalogram during mental tasks," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 3, pp. 277–286, 1998.

[36] B. Obermaier, C. Neuper, C. Guger, and G. Pfurtscheller, "Information transfer rate in a five-classes brain-computer interface," *IEEE Trans. Neural Sys. Rehab. Eng.*, vol. 9, no. 3, pp. 283–288, 2001.

[37] J. R. Wolpaw, D. J. McFarland, D. J. Neat, and C. A. Forneris, "An EEG-based brain-computer interface for cursor control," *Clinical Neurophysiology*, vol. 78, no. 3, pp. 252–259, 1991.

[38] A. Kubler, B. Kotchoubey, T. Hinterberger, N. Ghanayim, J. Perelmouter, M. Schauer, C. Fritsch, E. Taub, and N. Birbaumer, "The thought translation device: a neurophysiological approach to commincation in total motor paralysis," *Experimental Brain Research*, vol. 124, pp. 223–232, 1999.

[39] J. R. Wolpaw, D. Flotzinger, G. Pfurtscheller, and D. F. McFarland, "A timing of EEG-based cursor control," *Clinical Neurophysiology*, vol. 146, pp. 529–538, 1997.

[40] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kubler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Nature*, vol. 398, pp. 297–298, 1999.

[41] B. Blankertz, G. Curio, and K.-R. Müller, "Classifying single trial EEG: Towards brain computer interfacing," in *Advances in Neural Information Processing Systems* (S. B. T.G. Dietterich and Z. Ghahramani, eds.), vol. 14, (Cambridge, MA), pp. 157–164, MIT Press, 2002.

[42] L. Parra, C. Alvino, A. Tang, B. Pearlmutter, N. Yeung, A. Osman, and P. Sajda, "Linear spatial integration for single trial detection in encephalography," *NeuroImage*, vol. 17, no. 1, pp. 223–230, 2002.

[43] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg, "Designing optimal spatial filters for single-trial EEG classification in a movement task," *Clinical Neurophysiology*, vol. 110, pp. 787–798, 1998.

[44] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 441–446, 2000.

[45] S. Lemm, B. Blankertz, G. Curio, and K.-R. Müller, "Spatio-spectral filters for improving the classification of single trial EEG," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 9, pp. 1541–1548, 2005.

[46] D. Model and M. Zibulevsky, "Signal reconstruction in sensor arrays using temporal-spatial sparsity regularization," in *ICA'04 Proceedings, Lecture Notes in Computer Science, Volume 3195, Oct 2004*, pp. 319–326.

[47] D. Model and M. Zibulevsky, "Signal reconstruction in sensor arrays using sparse representations," *Signal Processing*, 2005, In Press.

[48] K. B. Christensen, "The application of digital signal processing to large-scale simulation of room acoustics: frequency responce modeling and optimization software for multichannel dsp engine," *J. AES*, vol. 40, pp. 260–276, Apr. 1992.

[49] D. M. Malioutov, "A sparse signal reconstruction perspective for source localization with sensor arrays," Master's thesis, MIT, July 2003.

[50] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact newton methods," *SIAM Journal on Numerical Analysis*, vol. 19, pp. 400–408, 1982.

[51] S. Nash, "A survey of truncated-newton methods," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 45–59, 2000.

[52] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. New York: Academic Press, 1981.

[53] M. Çetin, D. M. Malioutov, and A. S. Willsky, "A variational technique for source localization based on a sparse signal reconstruction perspective," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 2965–2968, May 2002.

[54] D. M. Malioutov, M. Çetin, J. W. FisherIII, and A. S. Willsky, "Superresolution source localization through data-adaptive regularization," *IEEE Sensor Array and Multichannel Signal Processing Workshop*, pp. 194–198, Aug. 2002.

[55] [online] ICA'99 Synthetic Benchmarks, http://sound.media.mit.edu/ica-bench/.

[56] M. Zibulevsky and Y. Y. Zeevi, "Extraction of a single source from multichannel data using sparse decomposition," *Neurocomputing*, 2002. accepted for publication.

[57] P. Sajda, A. Gerson, K.-R. Müller, B. Blankertz, and L. Parra, "A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces," *IEEE Trans. Neural Sys. Rehab. Eng.*, vol. 11, no. 2, pp. 184–185, 2003. Datasets and details about the results available at: http://newton.bme.columbia.edu/competition.htm.

[58] B. Blankertz, K.-R. Müller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, and N. Birbaumer, "The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1044–1051, 2004. Datasets and details about the results available at: http://ida.first.fraunhofer.de/projects/bci/competition/.

[59] R. P. Duin, "Prtools, a pattern recognition toolbox for matlab." Pattern Recognition Group, Delft University of Technology, 2000. Download from http://www.prtools.org/.

[60] T. Cover and P. Hart., "Nearest neighbor pattern classification," *IEEE Trans. Info. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[61] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[62] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York, USA: John Wiley & Sons, second ed., 2001.

[63] Z. Bigman and H. Pratt, "Time course and nature of stimulus evaluation in category induction as revealed by visual event-related potentials," *Biol. Psychol.*, vol. 66, pp. 99–128, 2004.