



Signal reconstruction in sensor arrays using sparse representations[☆]

Dmitri Model, Michael Zibulevsky*

Technion—Israel Institute of Technology, Electrical Engineering Department, Haifa, Israel

Received 9 November 2004; received in revised form 11 May 2005; accepted 30 May 2005
Available online 3 August 2005

Abstract

We propose a technique of multisensor signal reconstruction based on the assumption, that source signals are spatially sparse, as well as have sparse representation in a chosen dictionary in time domain. This leads to a large scale convex optimization problem, which involves combined l_1 - l_2 norm minimization. The optimization is carried by the truncated Newton method, using preconditioned conjugate gradients in inner iterations. The byproduct of reconstruction is the estimation of source locations.

© 2005 Published by Elsevier B.V.

Keywords: Source localization; Source reconstruction; Wideband; Multipath

1. Introduction

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 [1] is potentially the most precise technique; it assumes however that the number of sources and the source spectral density matrix are known. Also 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 [2]. 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 [3]. Under the condition that the observation period is long

[☆]This research has been supported in parts by the “Dvorah” fund of the Technion and by the HASSIP Research Network Program HPRN-CT-2002-00285, sponsored by the European Commission. The research was carried out in the Ollendorff Minerva Center, funded through the BMBF.

*Corresponding author. Tel.: +972 4 829 4724; fax: +972 4 829 4799.

E-mail addresses: dmm@tx.technion.ac.il (D. Model), mzib@ee.technion.ac.il (M. Zibulevsky).

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, Capon's MLM [4], and autoregressive spectral estimators [5]. The concept of signal-subspace processing can also be used in the wide-band case. The technique given in [6] 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 [7]. 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 [7]. They demonstrate better performance than corresponding incoherent techniques, but still require rather long observation time and high SNR ratio for good estimation of covariance matrices.

The method presented in our work is very general, it is applicable for both narrow-band and wide-band signals in both near-field and far-field scenarios. Both localization of the sources and estimation of their time courses is achieved simultaneously. We assume that incoming signals can be sparsely represented in an appropriate basis or frame (e.g., via the short time Fourier transform, wavelet transform, wavelet packets, etc.). This idea was exploited, for example, in [8,9] for a very efficient blind source separation. We also divide the space into a discrete grid of potential source locations and assume that the sources are sparsely located there (similarly to [10–12]). Since the number of possible source locations is often much greater than the number of sensors, the corresponding inverse-problem is ill-posed. In order to regularize the solution, both spatial and temporal sparsity are enforced using l_1 -norm and non-squared l_2 norm regularization. The use of l_1 norm in order to achieve sparsity is well known in signal representation community, see for example [8,13].

Our method deals with the sensor array model in time domain, and thus is applicable for both narrowband and wideband signals. It also treats the multipath (convolutive) model of signal propagation. The combination of assumptions of spatial and temporal sparsity leads to an improved performance, as demonstrated by our simulations.

2. Problem formulation

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 [14]) (Fig. 1(a)). Thus, we can describe the output of the m th sensor, $y_m[n]$, as

$$y_m[n] = \sum_{k=1}^K h_{km}[n] * s_k[n] + n_m[n], \quad (1)$$

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 $h_{km}[n] = \delta(n - D_{km})$:

$$y_m[n] = \sum_{k=1}^K \delta(n - D_{km}) * s_k[n] + n_m[n]. \quad (2)$$

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. $D_{k1} = 0 \forall k$.

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.

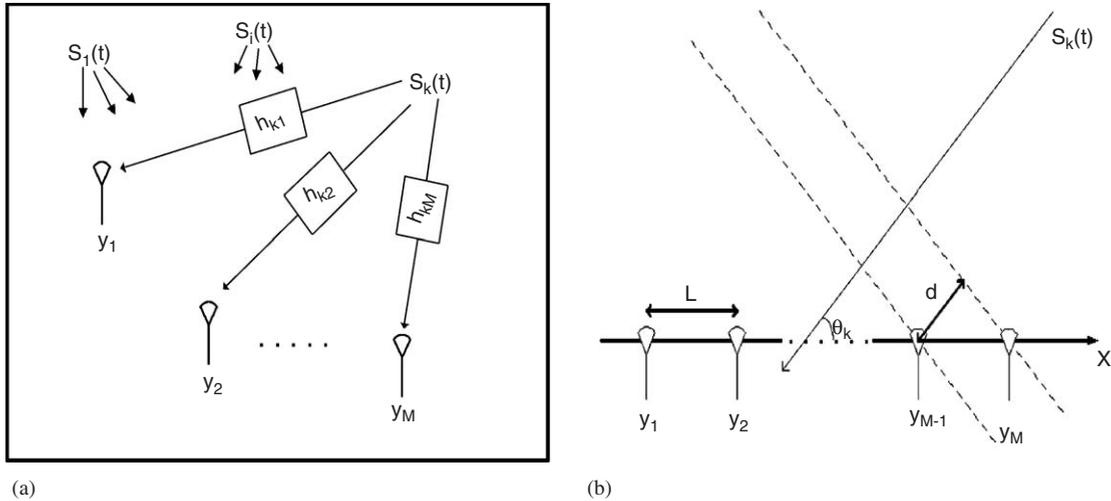


Fig. 1. (a) In a reverberant environment a propagation of the signal $s_k[n]$ toward the i th sensor is modelled as a convolution with a transfer function $h_{ki}[n]$. (b) An illustration of uniform linear array. The signal $s_k[n]$ impinges on the array from far field.

Denote the potential source signal at l th location as $s_l[n]$, $1 \leq l \leq T$. Suppose, we have L potential locations, and the signals are limited in time to T samples. Let us introduce an $L \times T$ 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 is 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

$$u_m[n] = \sum_{l=1}^L h_{lm}[n] * s_l[n], \tag{3}$$

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

$$Y = \mathcal{A}S + N, \tag{4}$$

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 adjoint¹ 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

$$x_i[n] = \sum_{j=1}^m (h_{ji}[-k] * y_j[k])[n]. \tag{5}$$

Note the “minus” sign near the argument of h_{ji} .

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

¹Linear operators L and L^* are adjoint if the inner products $\langle L^*x, y \rangle = \langle x, Ly \rangle$ for any x and y .

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 $\mathcal{I}_{N_{\text{up}}}$ denote upsampling by factor N_{up} operator, and if S is an $L \times T$ matrix, then $S_{\text{up}} = \mathcal{I}_{N_{\text{up}}} S$ will be $L \times TN_{\text{up}}$ matrix. Note, that the $1 + N_{\text{up}}(i - 1)$ th column of S_{up} is equal to the i th 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_{up}^j , of the matrix S_{up} . This column corresponds to a time point, laying between the samples $k = \lceil j/N_{\text{up}} \rceil$ and $k + 1$ of the original signal ($\lceil \cdot \rceil$ is the ceiling function). The distances between the above time point and the closest samples of original signal are $d^- = (j - (k - 1)N_{\text{up}} - 1)/N_{\text{up}}$ to the left sample and $d^+ = 1 - d^-$ to the right sample (measured in sampling periods T_s). Finally, if ρ is the interpolation kernel, N_{io} is an interpolation order and S^k is the k th column of S , then

$$S_{\text{up}}^j = \sum_{l=-N_{\text{io}}+1}^{N_{\text{io}}} \rho(l - d^-) S^{k+l}. \quad (6)$$

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

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

$$S_r^k = \sum_{l=-N_{\text{io}}*N_{\text{up}}}^{N_{\text{io}}*N_{\text{up}}} \rho\left(\frac{l}{N_{\text{up}}}\right) S_{\text{up}}^{N_{\text{up}}(k-1)+l}. \quad (7)$$

Now, in our model we will use the modified operators

$$\hat{\mathcal{A}} = \mathcal{A} \mathcal{I}_{N_{\text{up}}}, \quad \hat{\mathcal{A}}^* = \mathcal{I}_{N_{\text{up}}}^* \mathcal{A}^* \quad (8)$$

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 $D_{ji} N_{\text{up}}$. We will still need to round $D_{ji} N_{\text{up}}$ to the closest integer, but now the rounding error is N_{up} times less.

3. Algorithm description

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 other 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-posed: 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 [13]). Particularly, there exists some operator Φ and the *sparse* matrix of coefficients, C , such that

$$S = C\Phi. \quad (9)$$

The matrix Φ 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 [12]. This sparsity indeed exists in our problem formulation. Recall, that we have 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 :

$$F(C) = \frac{1}{2} \|Y - \mathcal{A}(C\Phi)\|_F^2 + \mu_1 \sum_{ij} |c_{ij}| + \mu_2 \sum_{i=1}^m \|c_i\|_2, \quad (10)$$

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 μ_1 and μ_2 are used to regulate the weight of each term.

The first term in (10) 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 [8,13]. 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 (9). However the inverse transformation is not always defined (for example when the chosen signal dictionary is *overcomplete*).

Note also, that quite often matrix Φ does not need to be stored explicitly. Multiplication by Φ and Φ^* 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 [13] for details and more examples.

In order to minimize the objective (10) numerically, we use a smooth approximation of the l_2 -norm, having the following form:²

$$\psi(x) = \sqrt{\sum_i x_i^2 + \varepsilon} \approx \|x\|_2 \quad (11)$$

the approximation becomes more precise as $\varepsilon \rightarrow 0$. It can be easily seen, that if ψ is applied to a single element of x —it becomes the smooth approximation of absolute value

$$\psi(x_i) = \sqrt{x_i^2 + \varepsilon} \approx |x|. \quad (12)$$

²An alternative would be to solve the problem in the Conic Programming framework; we leave this option for the future.

Using (11) and (12), we obtain the following objective function:

$$F(C) = \frac{1}{2} \|Y - \mathcal{A}(C\Phi)\|_F^2 + \mu_1 \sum_{i,j} \psi(c_{ij}) + \mu_2 \sum_{i=1}^m \psi(c_i). \quad (13)$$

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 Appendix A). Hence, the objective (13) 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* (TN) method [15,16]. In the *TN* method the 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* (CG) method. We use diagonal preconditioning in order to further speed up the optimization [17]. Note that in *TN* method, the memory consumption grows linearly with the number of variables. This enables us to solve large problems with fair performance. See Appendix A for detailed calculation of objective function derivatives and Appendix B for detailed description of *TN* and preconditioned (CG) algorithms.

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 does not 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

super-resolution. However, the disadvantage of our approach is a high computation complexity.

In our simulations we did not use signals recorded in a real-world 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-world problems. Thus, we have 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: $Y = AS + N$, where A defined in (8) 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 (6) to have different values, from those used later in reconstruction procedure. This is done in order to introduce the “model error”.

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).

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 are lined up with $\lambda_{min}/2 = \frac{1}{2}c/f_{max}$ 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 have chosen to have only 2 active sources, located very close to each other—within 10° . In these conditions conventional methods, such as beamforming [18] and MUSIC fail to super-resolve them (as shown in [10,11]). The experiment was successful. Using proposed algorithm, we have correctly estimated the source positions (Fig. 2(a)), while the conventional delay-sum beamforming has failed to resolve the closely spaced sources (Fig. 2(b)). We have also correctly reconstructed the sources. The normalized reconstruction error was less

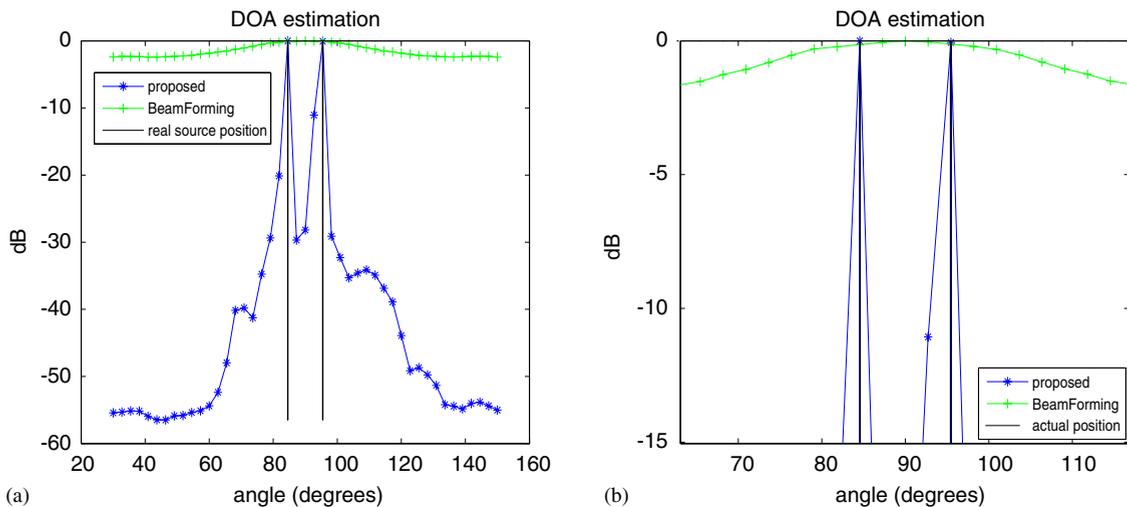


Fig. 2. (a) Noise-free case. The proposed method correctly identifies the location of sources. (b) Zoom-in picture, in which it can be clearly seen that beamforming fails to separate closely spaced sources.

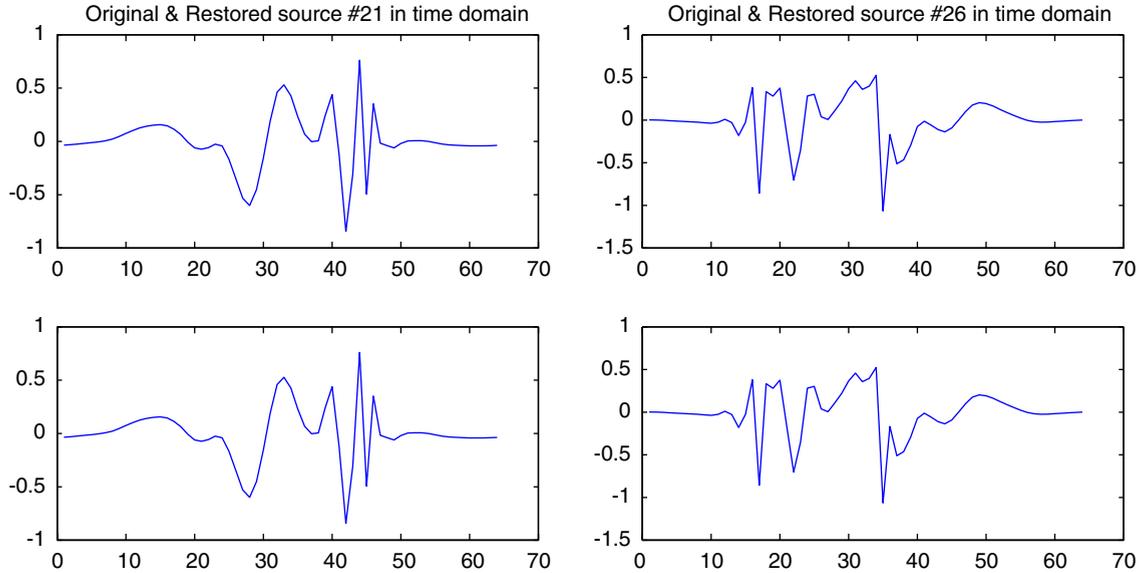


Fig. 3. Source separation (no noise). Top: sources from 2 active directions, bottom: restored sources. The normalized reconstruction error is less than $1e-3$.

than 5×10^{-3} (Fig. 3) (the error was calculated according to $\|s_{\text{init}}/\|s_{\text{init}}\|_2 - s_{\text{rec}}/\|s_{\text{rec}}\|_2\|_2$).

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 5 dB. 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 [10,11]. This can be done by setting $\mu_1 = 0$ in (13). 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 Figs. 4 and 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 (Fig. 4(a)), it has failed to resolve the closely-spaced sources. In the

second experiment (Fig. 4(c)), one of the signals has about twice larger energy than the other. In this case, the beamforming has found only the first source.

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 Fig. 6, the active sources were restored rather accurately. The reconstruction error was about 5×10^{-2} . This is a very good result, particularly if we take into consideration a high noise level (SNR = 5 dB) of the sensor signals.

Fig. 7 demonstrates the convergence of the algorithm vs. number of iterations. We have compared the computational load of the algorithm for two cases: with and without preconditioning of CG (see Appendix B for description of TN and CG algorithms). The number of TN (outer) iterations was slightly less for unpreconditioned 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 vs. 150 min).

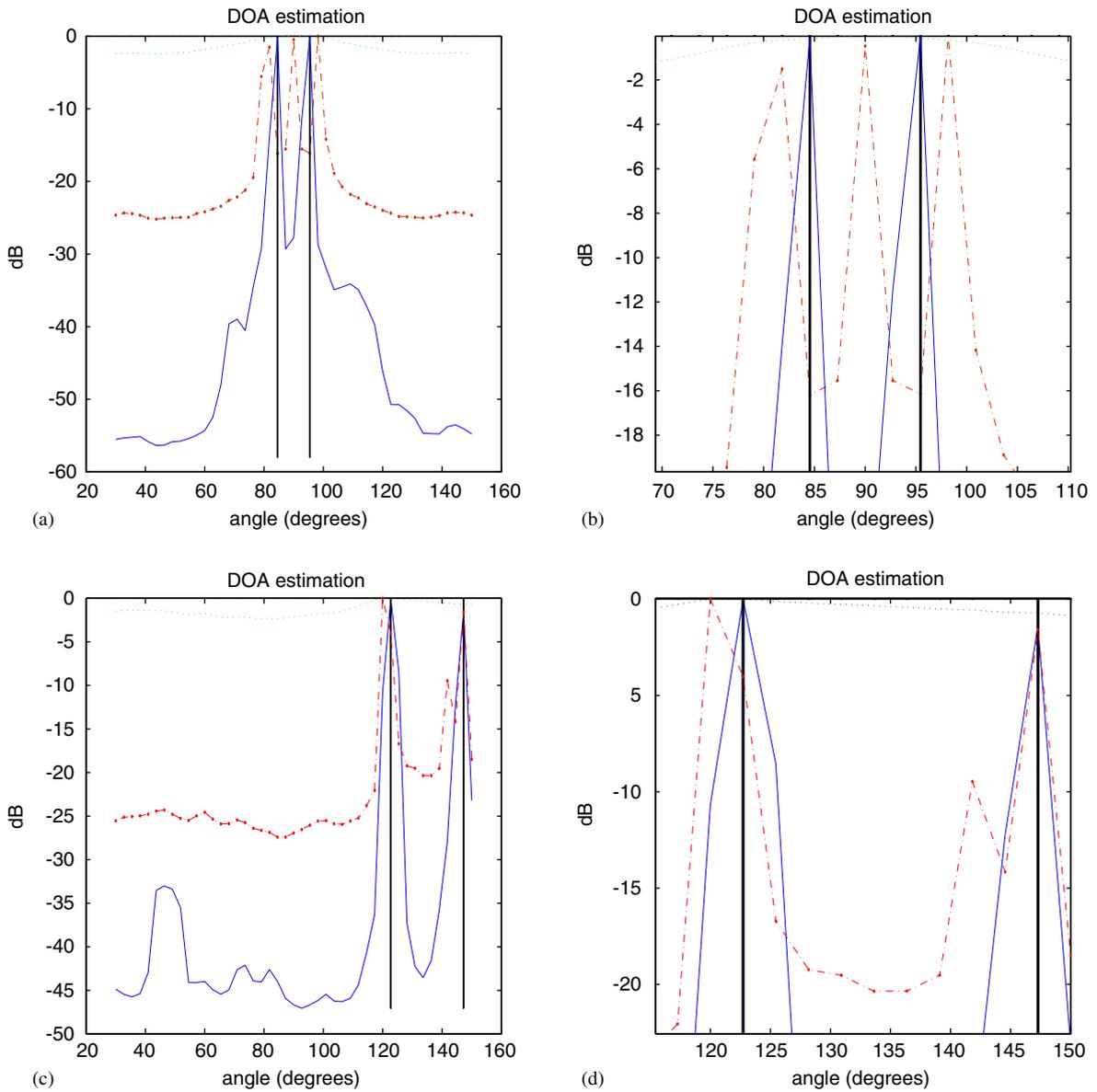


Fig. 4. Examples of DOA estimation by proposed and alternative methods (SNR = 5 db). (a), (c) Solid line represents DOA estimation by proposed method. Two peaks coincide with actual source positions (marked by solid vertical line). However, DOA estimation based on spatial sparsity only (dash-dot line) 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.

4.1.3. Reconstruction of signals with echo

The problem setup is as follows: 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 (Fig. 8), i.e. signals which arrive from angles $0 < \alpha < 90$ arrive to the array in the straight

path and as an reflection from the wall. However signals which 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 reverberations from the wall). Then, we have

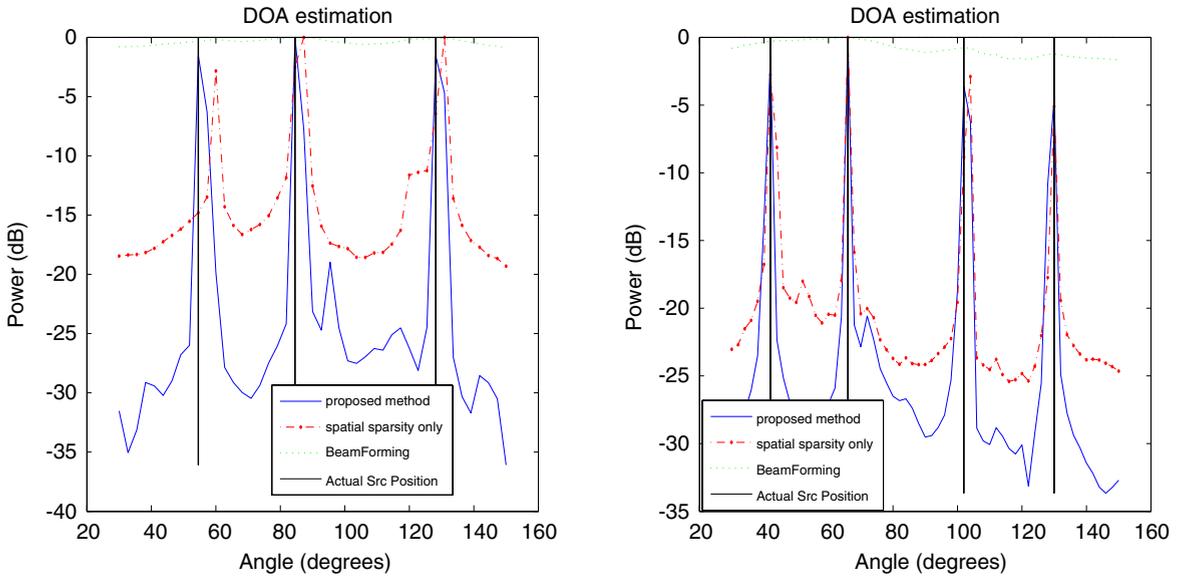


Fig. 5. More examples of DOA estimation by proposed and alternative methods (SNR = 10 db). The proposed method has correctly identified the source locations. However, the method based on spatial sparsity only has a bias with respect to some source locations.

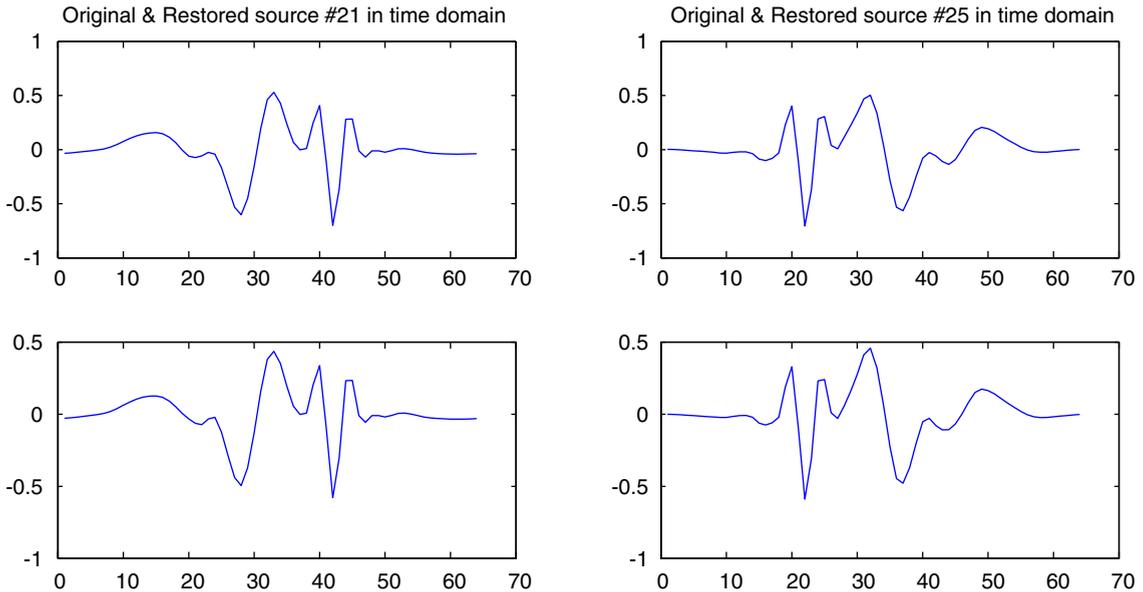


Fig. 6. Source separation (SNR = 5 dB). Top: sources from 2 active directions, bottom: restored sources. The normalized reconstruction error is about 5×10^{-2} .

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 does not

include the reverberations from the wall (*straight-path only* reconstruction).

Our experiments show, that using *convolutive* reconstruction, we are able to correctly identify the

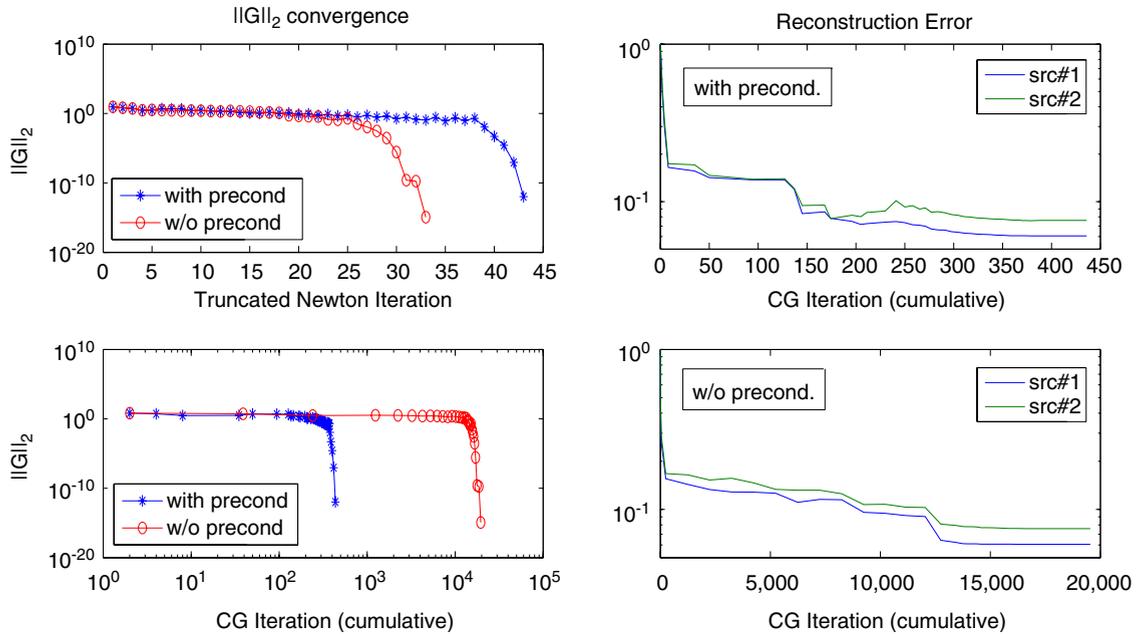


Fig. 7. Optimization convergence. The usage of preconditioning in CG drastically reduces the computational load. The total number of iterations were reduced by a factor of ≈ 45 . The computation time was reduced by a factor of ≈ 28 due to preconditioning.

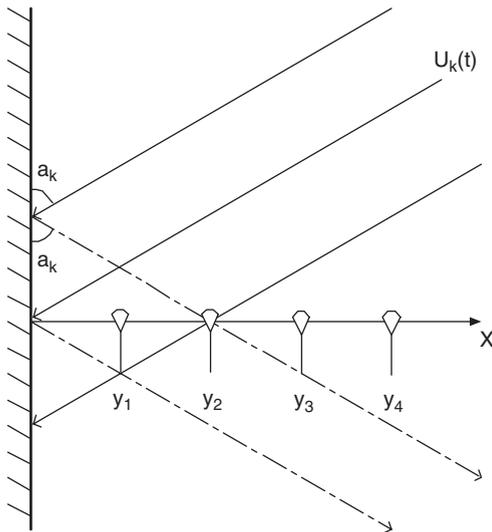


Fig. 8. Illustration of far-field model with reverberation. Infinite wall is placed to the left of the linear sensor array. Signals arrive to the array in the straight path, and as reverberation from the wall.

source' locations even in the case, when the echo has the same amplitude as the straight-path signal (reverberation factor equal to 1). However, using

the *straight-path only* technique, we are able to restore sources only in the light reverberation conditions (reverberation factor equal or less than 0.3). In both cases, we have evaluated the algorithm in the noisy environment. Some results are shown in Fig. 9.

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 near-field, 2D environment. The “2D room” was chosen to have 1 m \times 1 m 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 reverberations from the walls. The generation of transfer functions was done using the code available at public website [19]. The remaining setup procedure was similar to previous cases.

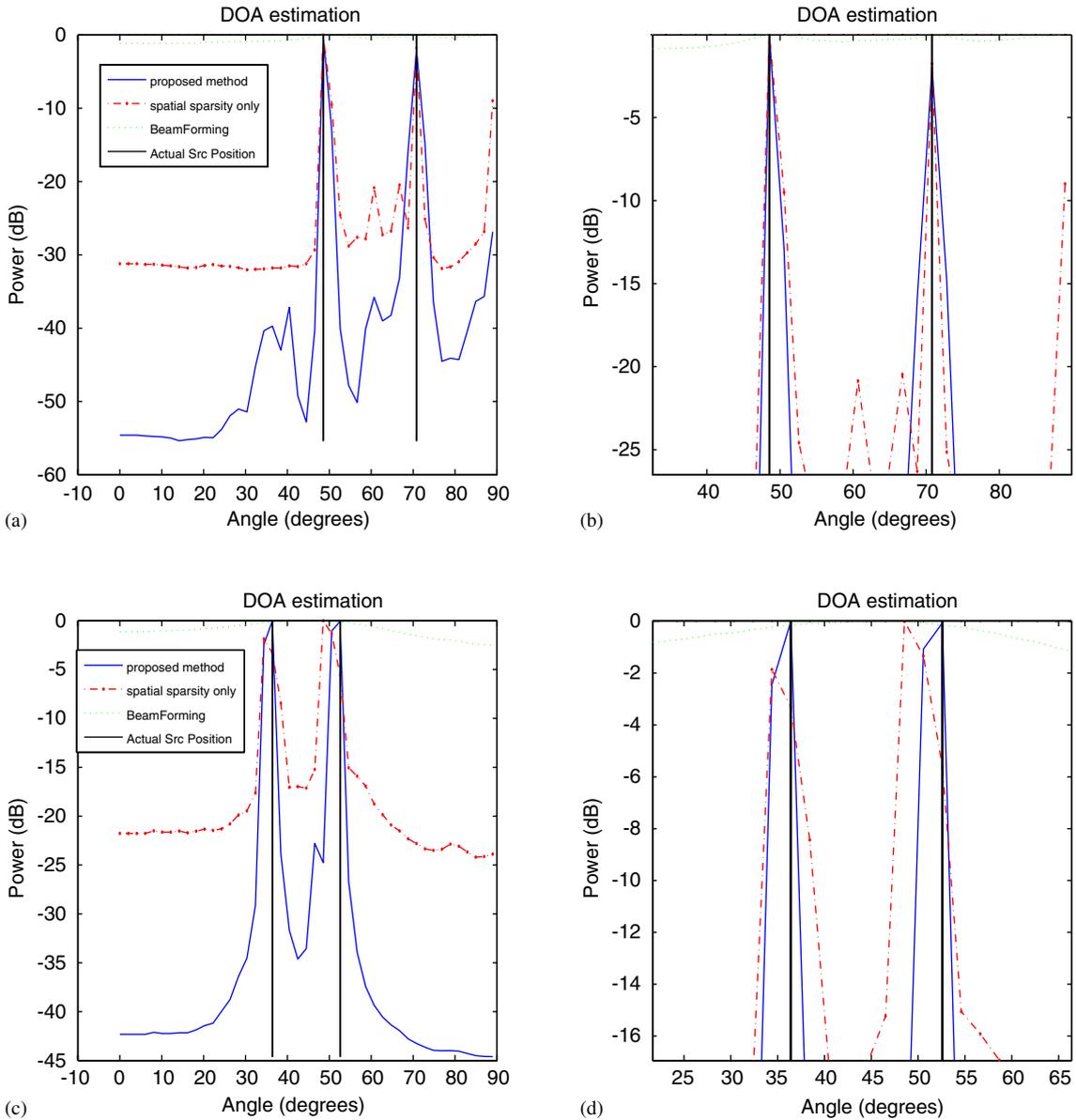


Fig. 9. Examples of DOA estimation in reverberant environment. (a) *Convolutional* reconstruction (SNR = 5 dB, reverberation factor = 0.9). The proposed algorithm has correctly estimated the DOAs. The side-lobes level is about 15 dB lower, with comparison to DOA estimation based on spatial sparsity only (b) *convolutional* reconstruction, zoom-in picture (c) *straight-path only* reconstruction (SNR = 5 dB, reverberation 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.

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 Fig. 10.

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 position of sensors and the noise level are appropriate: i.e. each

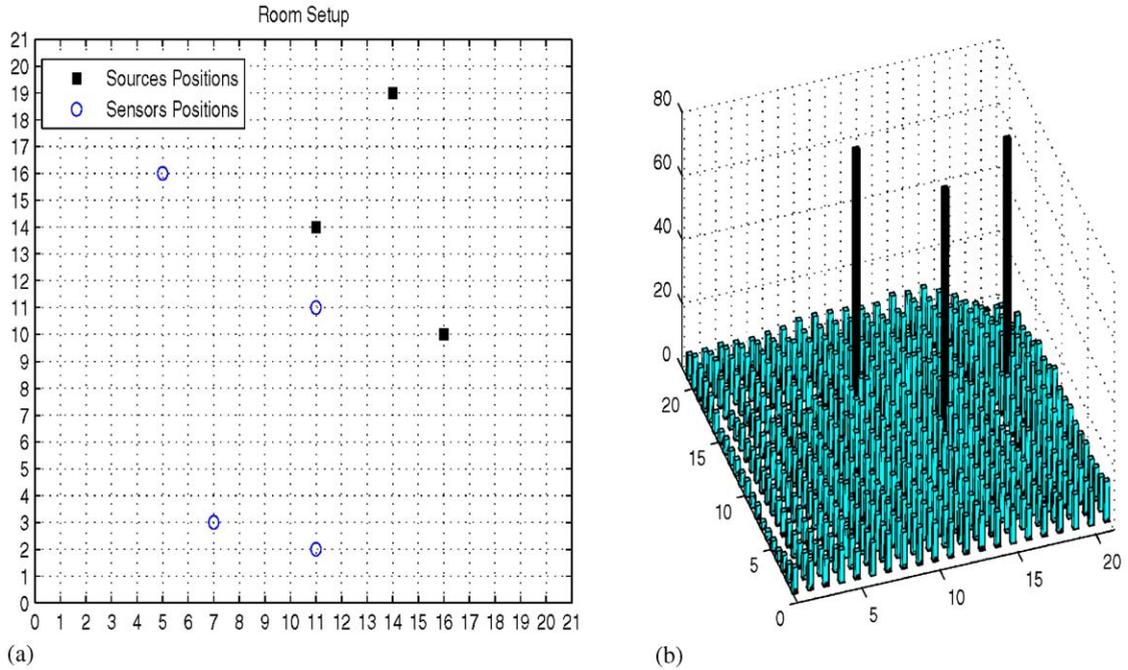


Fig. 10. 2D near-field, noiseless environment. (a) Sensors and active sources positions. Four sensors are depicted by circles. Three active sources are depicted by squares. (b) Estimated signal energy at each possible location. The proposed algorithm has correctly identified locations of active sources.

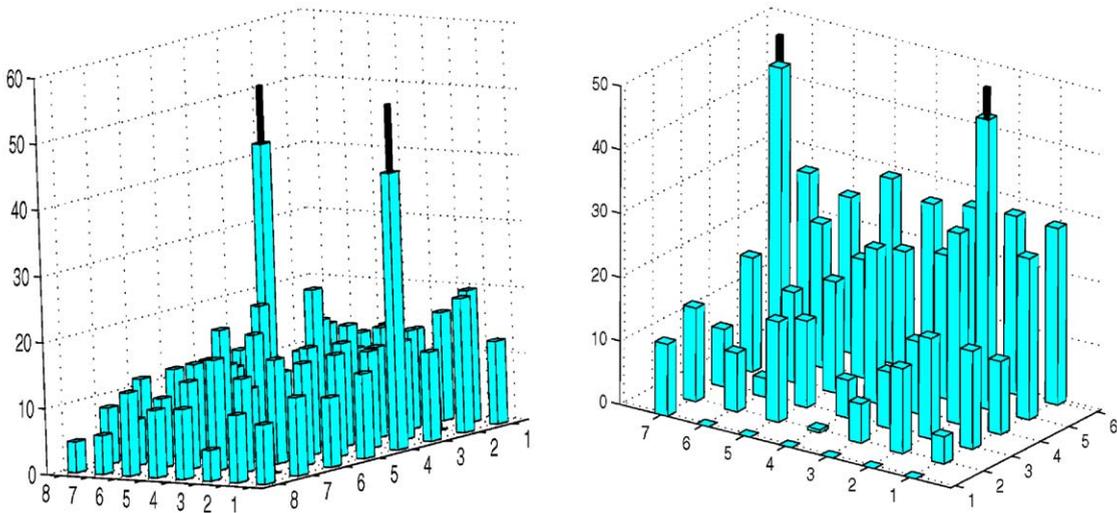


Fig. 11. Two examples of source localization in near-field noisy environment. In both cases the true source locations were identified correctly.

of source signals reaches at least two sensors with amplitudes larger than 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 Fig. 11. The signal-to-noise ratio was 10 dB. The locations of active sources were identified correctly in both cases. The source signals themselves were reconstructed with (normalized) error of about 6×10^{-2} .

5. Conclusions

Conducted simulations demonstrate 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 latter 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.

Appendix A. Objective function derivatives

In order to use the *TN* method, we need to calculate the gradient G of the objective (13), 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 CG . 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 appendix. We will also use G_i and \mathcal{H}_i to denote gradient and Hessian of respective terms of the objective function (13).

Let us start with the first term in (13). We will define a new operator \mathcal{B} in the following way:

$$\mathcal{B}C = \mathcal{A}(C\Phi), \quad \mathcal{B}^*X = (\mathcal{A}^*X)\Phi^*. \tag{A.1}$$

This enables us to write the first term in (13) as: $F_1 = \frac{1}{2} \|\mathcal{B}C - Y\|_F^2$. If we introduce new variable $U = \mathcal{B}C - Y$, then $F_1 = \frac{1}{2} \|U\|_F^2 = \frac{1}{2} \text{Tr}(U^T U)$. Hence, $dF_1 = \frac{1}{2} (\text{Tr}(U^T dU) + \text{Tr}(dU^T U)) = \text{Tr}(U^T dU)$. Substituting U and $dU = \mathcal{B} dC$ yields $dF_1 = \text{Tr}((\mathcal{B}C - Y)^T \mathcal{B} dC) = \langle \mathcal{B}C - Y, \mathcal{B} dC \rangle = \langle \mathcal{B}^*(\mathcal{B}C - Y), dC \rangle$. Recall that $dF = \langle G, dC \rangle$, and we get the gradient

$$G_1(C) = \mathcal{B}^*(\mathcal{B}C - Y). \tag{A.2}$$

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 (A.2) $dG_1(C) = \mathcal{B}^*(\mathcal{B} dC)$, and thus for an arbitrary X

$$\mathcal{H}_1 X = \mathcal{B}^*(\mathcal{B}X) \tag{A.3}$$

parentheses are used to ensure correct order of multiplications.

In order to proceed with the second and the third terms of objective (13), we need to use the gradient and Hessian of (11)

$$\nabla \psi(x) = \frac{1}{\psi(x)} x, \tag{A.4}$$

$$\begin{aligned} (\nabla^2 \psi(x))_{ii} &= -\frac{1}{\psi^3(x)} x_i^2 + \frac{1}{\psi(x)}, \\ (\nabla^2 \psi(x))_{ij} &= -\frac{1}{\psi^3(x)} x_i x_j, \quad (i \neq j), \end{aligned} \tag{A.5}$$

where $(\nabla^2 \psi(x))_{ii}$ and $(\nabla^2 \psi(x))_{ij}$ are diagonal and off diagonal 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 (13)

$$(G_2)_{ij} = \mu_1 \frac{1}{\psi(c_{ij})} c_{ij}, \tag{A.6}$$

$$(G_3)_{ij} = \mu_2 \frac{1}{\psi(c_i)} c_{ij}, \tag{A.7}$$

note, that the gradient of (13) 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 (A.6), 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

$$\tilde{H}_{2_{ij}} = \mu_1 \left(-\frac{1}{\psi^3(c_{ij})} c_{ij}^2 + \frac{1}{\psi(c_{ij})} \right). \quad (\text{A.8})$$

It is obvious, that

$$\mathcal{H}_2 X = \tilde{H}_2 \odot X, \quad (\text{A.9})$$

where \odot is element-wise multiplication.

In order to define the multiplication $\mathcal{H}_3 X$ we need to rewrite the Eq. (A.5)

$$\nabla^2 \psi(c_i^T) = \frac{1}{\psi^3(c_i^T)} c_i^T c_i + \frac{1}{\psi(c_i^T)} I, \quad (\text{A.10})$$

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

$$(\mathcal{H}_3 X)_i = \mu_2 \left(-\frac{1}{\psi^3(c_i^T)} c_i (c_i x_i^T) + \frac{1}{\psi(c_i^T)} x_i \right), \quad (\text{A.11})$$

where x_i is the i th row of matrix X .

This calculus is sufficient for the *TN* method. However, in order to use *preconditioned CG* method for inner iterations, we need to define the diagonal of the Hessian of (13).

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

$$(\tilde{H}_1)_{ij} = \langle E, \mathcal{H}_1 E \rangle, \quad (\text{A.12})$$

where \tilde{H}_1 is a diagonal of \mathcal{H}_1 packed in the same manner as a diagonal of H_2 in (A.8).

It follows from (A.3) 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 (A.8). Finally, the diagonal of \mathcal{H}_3 , packed in the same manner as a diagonal of \mathcal{H}_2 , is given by

$$(\tilde{H}_3)_{ij} = \mu_2 \left(-\frac{1}{\psi^3(c_i)} c_{ij}^2 + \frac{1}{\psi^3(c_i)} \right). \quad (\text{A.13})$$

Appendix B. TN method

In the algorithm description we will use the following notations: $f(C)$ —the objective function (13). G and \mathcal{H} are the gradient and the Hessian of $f(C)$, respectively. The *TN* method applied to the objective (13) has the following iterative scheme:

- (1) Start with an initial estimate C_0 of source coefficients.
- (2) For $k = 1, 2, \dots$ 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 α_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 CG**. We use the diagonal operator \mathcal{W} for preconditioning. \mathcal{W} has the same size and the diagonal as \mathcal{H} —the Hessian of (13). Since \mathcal{W} is diagonal, the calculation of \mathcal{W}^{-1} is straightforward. Moreover, the optimization algorithm does not 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 $k = 1, 2, \dots$
 - (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 $\langle A, B \rangle = \text{Tr}(A^T B) = \sum_{ij} a_{ij} b_{ij}$ is an inner product of two matrices A and B .

Note, that we are not looking for the exact solution of step 2a of *TN* 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 $\|R_k\|_2 / \|R_0\|_2$ is low enough—say 10^{-3} .

References

- [1] M. Wax, T. Kailath, Optimum localization of multiple sources by passive arrays, *IEEE Trans. Acoust. Speech Signal Process. ASSP-31* (September 1983) 1210–1218.
- [2] A. Nehorai, G. Su, M. Morf, Estimation of time differences of arrival by pole decomposition, *IEEE Trans. Acoust. Speech Signal Process. ASSP-31* (December 1983) 1478–1491.
- [3] R.O. Schmidt, A signal subspace approach to multiple emitter location and spectral estimation, Ph.D. Thesis, Stanford University, 1981.
- [4] J. Capon, High-resolution frequency wavenumber spectrum analysis, *Proc. IEEE* 8 (1969) 1408–1418.
- [5] J.E. Evans, J.R. Johnson, D.F. Sun, Application of advanced signal processing techniques to angle of arrival estimation in atc navigation and surveillance systems, Technical Report 582, MIT Lincoln Laboratory, 1982.
- [6] G. Su, M. Morf, The signal subspace approach for multiple wide-band emitter location, *IEEE Trans. Acoust. Speech Signal Process. ASSP-31* (December 1983) 1502–1522.
- [7] H. Wang, M. Kaveh, Coherent signal-subspace processing for the detection and estimation of angle of arrival of multiple wide-band sources, *IEEE Trans. Acoust. Speech Signal Process. ASSP-33* (August 1985) 823–831.
- [8] M. Zibulevsky, B.A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, *Neural Computations* 13 (4) (2001) 863–882.
- [9] M. Zibulevsky, B.A. Pearlmutter, P. Bofill, P. Kisilev, Blind source separation by sparse decomposition, in: S.J. Roberts, R.M. Everson (Eds.), *Independent Components Analysis: Principles and Practice*, Cambridge University Press, Cambridge, 2001.
- [10] M. Çetin, D.M. Malioutov, A.S. Willsky, A variational technique for source localization based on a sparse signal reconstruction perspective, in: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, May 2002, pp. 2965–2968.
- [11] D.M. Malioutov, M. Çetin, J.W. Fisher III, A.S. Willsky, Superresolution source localization through data-adaptive regularization, in: *IEEE Sensor Array and Multichannel Signal Processing Workshop*, August 2002, pp. 194–198.
- [12] D.M. Malioutov, A sparse signal reconstruction perspective for source localization with sensor arrays, Master's Thesis, MIT, July 2003.
- [13] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM J. Sci. Comput.* 20 (1) (1998) 33–61.
- [14] K.B. Christensen, The application of digital signal processing to large-scale simulation of room acoustics: frequency response modeling and optimization software for multi-channel dsp engine, *J. AES* 40 (April 1992) 260–276.
- [15] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, *SIAM J. Numer. Anal.* 19 (1982) 400–408.
- [16] S. Nash, A survey of truncated-Newton methods, *J. Comput. Appl. Math.* 124 (2000) 45–59.
- [17] P.E. Gill, W. Murray, M.H. Wright, *Practical Optimization*, Academic Press, New York, 1981.
- [18] D. Johnson, D. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [19] ICA'99 Synthetic benchmarks, <http://sound.media.mit.edu/ica-bench/> (online).