J. Nonlinear Var. Anal. 7 (2023), No. 4, pp. 467-485
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.7.2023.4.02

SPARSE BROADBAND BEAMFORMER DESIGN VIA PROXIMAL OPTIMIZATION TECHNIQUES

YONGXIN CHEN[1], ZHIBAO LI[2], XINGJU CAI[3], DEREN HAN[1,4,*]

[1]*LMIB, School of Mathematical Sciences, Beihang University, Beijing 100191, China*
[2]*School of Mathematics and Statistics, Central South University, Changsha 410083, China*
[3]*School of Mathematical Sciences, Jiangsu Key Laboratory for NSLSCS, Nanjing Normal University, Nanjing 210023, China*
[4]*Jiangsu Key Laboratory for NSLSCS, Nanjing Normal University, Nanjing 210023, China*

Abstract. Beamforming is one of the most important techniques to enhance the quality of signal in array sensor signal processing, and the performance of a beamformer is usually related to the design of array configuration and beamformer weight. Recently, it was realized that the sparsity of the filter coefficients can reduce the cost of signal acquisition and communication, and as a consequence, the sparse broadband beamformer design attracts more and more attentions. In this paper, we first propose a proximal sparse beamformer design model which obtains the sparse and robust filter coefficients through solving a composite optimization problem. The objective function of the model is the sum of a least squares term, a proximal term, and an ℓ_1-regularization term. The least squares term reflects the data fidelity; the proximal term, whose center is predetermined via a simple least squares, enhances the robustness; while the ℓ_1 term ensures the sparsity of the solution. This model not only maintains the authenticity of the least squares solution, but also ensures the sparsity of the filter coefficients. A significant feature of the model is that we use ‘partial’ data to obtain the least squares solution and use another ‘partial’ data to construct the data fidelity term, which can evidently decrease the computational cost. For solving the composite optimization problem, we tailor several popular algorithms, such as the alternating direction method of multipliers, the forward-backward splitting method, and the Douglas-Rachford splitting method. Numerical results observably exhibit the improvements of the proposed approach over existing works in both effectiveness and efficiencies.

Keywords. Alternating direction method of multipliers; Douglas-Rachford splitting method; Filter coefficients; Foward-backward splitting method; Sparse beamformer.

1. INTRODUCTION

Sensor array beamforming is a powerful technique to extract the signal of interest (SOI) in the desired direction while suppressing the interferences and noise in other directions. It is widely used in radar, sonar, wireless communications, satellite navigation, radio astronomy, speech enhancement, and other areas [1, 2]. The classic methods of beamformer design include the

*Corresponding author.
Email addresses: chenyongxin@buaa.edu.cn (Y. Chen), zblimath@csu.edu.cn (Z. Li), caixingju@njnu.edu.cn (X. Cai), handr@buaa.edu.cn (D. Han).

Received May 18, 2022; Accepted June 20, 2022.

minimum variance distortionless response (MVDR) beamformer design, the linearly constrained minimum variance (LCMV) beamformer design [3, 4], and the generalized sidelobe canceller (GSC) [5, 6]. Generally, the performance of beamforming is greatly affected by both the filter coefficients and the array configuration [7, 8, 9, 10]. Recently, the sparsity of array configuration attracts more and more attentions, and various strategies have been developed to devise sparse arrays for different tasks including direction finding, adaptive beamforming, and beampattern synthesis [11, 12, 13, 14]. With such a sparse array configuration, the beamformer can achieve increased array aperture and degrees of freedom while reducing the computational complexity compared to conventional uniform arrays [15, 16].

On the other hand, sparsity also plays an important role in filter design. It was realized that higher order sparse filter has the ability to improve the beamforming performance over lower order filters, offering considerable savings in hardware and data storage [17, 18]. A series of studies on the sparse filter design were investigated by numerous researchers in the past decades. For instance, the heuristic successive thinning and the ℓ_1-norm approximation approaches for sparse filter design in [19]; The branch-and-bound algorithm for quadratically-constrained sparse filter design in [20]; The joint sparsity and order optimization problems for sparse finite-duration impulse response (FIR) filter design in [21, 22, 23].

The mathematical model related to the sparse filter design is aimed essentially to minimize the ℓ_0-norm of filter coefficients, which is hard to handle both from the theoretical and the numerical points of view. One of the most common techniques to deal with such problems is the ℓ_p-norm relaxation, which was proved to be equivalent to the original problem under certain conditions [24, 25, 26]. In particular, the ℓ_1-norm minimization problem has attracted wide interest in the field of optimization [27, 28, 29, 30]. The advantage of ℓ_1-norm sparsity term is that it is convex, which is important from an optimization point of view; essentially, convexity, and non-convexity are the watersheds in optimization [31]. Hence in our model, we propose to use the ℓ_1-norm to enhance the sparsity of the solution.

Theoretically and intuitively, the beamformer design can be completed via solving a system of linear equations whose coefficient is a tall matrix, i.e., its number of rows is much larger than that of columns. However, due to the noise in data acquisition, the least-squares solution of the linear system is usually far away from the desired one. Specially, the solution loses its sparsity. At the same time, the system is usually ill-conditioned, making the numerical approach unstable.

In this paper, we propose a proximal sparse beamformer design model. To improve the sparsity, we combine the ℓ_1 term with the data fidelity term. Moreover, we also introduce a proximal term to enhance the stability of the model. Hence, the proposed model is a composite optimization problem whose objective function is the sum of a data fidelity term (least squares term), a proximal term whose center is the solution of a linear system from partial data and an ℓ_1-regularization term. A significant feature of the model is that we use 'partial' data to obtain the least squares solution and use another 'partial' data to construct the data fidelity term, which can evidently decrease the computational cost. The idea is borrowed from cross validation in modeling and other fields [32, 33], while here the main purpose is to reduce the computational burden via utilizing the data structure. That is, since the dimension of data collected by the signal is much larger than the number of parameters, the least squares solution of 'partial' data is not too far away from that of the whole data set. Moreover, using 'partial' data can also avoid over-fitting of the solution.

The structure of the composite optimization problem is very suitable for the alternating direction method of multipliers (ADMM). ADMM is an augmented Lagrangian based method, which generates the iterations via minimizing one block variable while the other block variables are fixed, i.e., we can understand the ADMM as a Gauss-Seidel implementation for solving linear equality constrained optimization problems with separable objective functions approximately with a single iteration [34]. Meanwhile, it can also be understood as the Douglas-Rachford splitting method applying to the dual problem [35]. One of the main reasons that ADMM performs reasonably well in dealing with the modern application problems arising from big data and artificial intelligence is that the subproblems in ADMM is very easy to solve and even possess closed-form solution. With the rapid development of ADMM, there have been several survey papers [34, 36, 37, 38] and we refer the interested readers to these papers and the references therein. In this paper, by introducing some auxiliary variables, we first reformulate the model into a separable optimization problem with linear equality constraints, which falls into the framework that is suitable for ADMM. The two subproblems are very simple. The first subproblem is the soft-thresholding, which just cost $O(n)$ flops, where n is the dimension of the first block variable. The second subproblem is a least-squares problem with a positive definite coefficient matrix, which is also low-cost, since the matrix possesses good structures.

ADMM belongs to the operator splitting algorithms. Similar algorithms include the Douglas-Rachform splitting methods [39, 40, 41] and forward-backward splitting methods [42]. We numerically compare the performance of these methods.

The rest of this paper is organized as follows. In Section 2, we formulate the mathematical model for the beamformer design problem and convert it into a linear equation system. In Section 3, we propose the proximal optimization problem on the design of sparse beamformer, as well as some algorithms. In Section 4, several numerical experiments are presented to verify the performance of the proximal sparse beamformer design methods. Section 5 ends the paper by giving some conclusions.

2. Beamformer Design Problem

We use i to represent the imaginary unit, i.e., $i=\sqrt{-1}$. Frequency is denoted as f and sampling frequency is denoted as f_s. Consider an M-element microphone with K-tap FIR filter. The filter coefficients are denoted as $w=[w_1^T,w_2^T,\cdots,w_M^T]^T$ and $w_j=[w_j(1),w_j(2),\cdots,w_j(K)]$ represents the coefficient of the j-th filter. Denote by $H_j(\boldsymbol{r},f)$ the transfer function from space point $\boldsymbol{r}=\{x,y,z\}$ to the j-th microphone element. The beamformer response can be expressed as

$$G(\boldsymbol{r},f)=\sum_{j=1}^{M}W_j(w,f)H_j(\boldsymbol{r},f), \tag{2.1}$$

where $W_j(w,f)=w_j^T d(f), j=1,2,\cdots,M$ and $d(f)=[1,\mathrm{e}^{\frac{-2\pi i f}{f_s}},\cdots,\mathrm{e}^{\frac{-2\pi i f(K-1)}{f_s}}]$. Let $G_d(\boldsymbol{r},f)$ denote the desired response. The beamforming target is to find a group of filter coefficients w, such that the beamformer response meets the desired response, i.e.,

$$\sum_{j=1}^{M}W_j(w,f)H_j(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)=0, \quad \forall\ (\boldsymbol{r},f)\in\Omega, \tag{2.2}$$

where Ω is the space-frequency domain of interest.

Since the space-frequency region Ω is continuous, (2.2) is semi-infinite. To solve it numerically, the discretization methods and reduction based approaches are usually introduced to transform it into the finite numerical problem approximately [43]. We approximate the space-frequency domain Ω by Ω_N, which is a multi-dimensional grid region with a uniform grid containing N mesh points in each dimension. For each of $(\boldsymbol{r}, f) \in \Omega_N$, we rearrange the beamformer response (2.1) as

$$G(\boldsymbol{r},f)=\sum_{j=1}^{M}W_j(w,f)H_j(\boldsymbol{r},f)=\sum_{j=1}^{M}(H_j(\boldsymbol{r},f)d(f))^T w_j = a(\boldsymbol{r},f)^T w,$$

where

$$a(\boldsymbol{r},f)=[H_1(\boldsymbol{r},f)d(f)^T, H_2(\boldsymbol{r},f)d(f)^T,\ldots,H_M(\boldsymbol{r},f)d(f)^T]^T \in \mathbb{R}^{MK}.$$

By expanding the complex vectors

$$a(\boldsymbol{r},f)=a_1(\boldsymbol{r},f)+ia_2(\boldsymbol{r},f),$$
$$G_d(\boldsymbol{r},f)=G_{d_1}(\boldsymbol{r},f)+iG_{d_2}(\boldsymbol{r},f),$$

where $a_1(\boldsymbol{r},f)$, $a_2(\boldsymbol{r},f)$, $G_{d_1}(\boldsymbol{r},f)$ and $G_{d_2}(\boldsymbol{r},f)$ are the real and imaginary parts of $a(\boldsymbol{r},f)$ and $G_d(\boldsymbol{r},f)$, respectively, beamformer design problem (2.2) can be divided into

$$\begin{cases} a_1(\boldsymbol{r},f)^T w - G_{d_1}(\boldsymbol{r},f)=0, \\ a_2(\boldsymbol{r},f)^T w - G_{d_2}(\boldsymbol{r},f)=0, \end{cases} \text{for all } (\boldsymbol{r},f)\in\Omega_N.$$

In matrix formulation, the above beamformer design problem will be written as a system of linear equations

$$Aw-b=0, \tag{2.3}$$

where the composite matrix

$$A=\begin{bmatrix} a_1(\boldsymbol{r}_1,f_1)^T \\ a_2(\boldsymbol{r}_1,f_1)^T \\ a_1(\boldsymbol{r}_2,f_2)^T \\ a_2(\boldsymbol{r}_2,f_2)^T \\ \vdots \\ a_1(\boldsymbol{r}_N,f_N)^T \\ a_2(\boldsymbol{r}_N,f_N)^T \end{bmatrix} \in \mathbb{R}^{2N\times MK},$$

and vector

$$b=[G_{d_1}(\boldsymbol{r}_1,f_1),\ G_{d_2}(\boldsymbol{r}_1,f_1),\ G_{d_1}(\boldsymbol{r}_2,f_2),\ G_{d_2}(\boldsymbol{r}_2,f_2),\ldots,\ G_{d_1}(\boldsymbol{r}_N,f_N),\ G_{d_2}(\boldsymbol{r}_N,f_N)]^T \in \mathbb{R}^{2N}.$$

In the following, for notation simplicity, we let $m=2N$ and $n=MK$. Usually, $m \gg n$. Hence, the matrix A in (2.3) is a tall matrix, and we assume that it has full column rank. Generally, (2.3) does not have an exact solution one resorts to the least squares solution w^*, i.e.,

$$w^*=(A^TA)^{-1}A^Tb,$$

which is the solution to the optimization problem

$$\min_{w\in\mathbb{R}^n} \frac{1}{2}\|Aw-b\|^2. \tag{2.4}$$

The least squares solution is not sparse and this complicates the filter design. To obtain a sparse solution, another term is introduced into the model, resulting to the composite optimization problem

$$\min_{w\in\mathbb{R}^n} \frac{1}{2}\|Aw-b\|^2+\gamma\|w\|_1, \tag{2.5}$$

where $\|w\|_1 = \sum_{i=1}^{n}|w_i|$. The beamformer design with the sparse FIR filters can simplify the arithmetic operations and speed up the time of output. [43] also proposed $\ell_2-\ell_p(0<p<1)$ minimization model

$$\min_{w\in\mathbb{R}^n} \frac{1}{2}\|Aw-b\|^2+\gamma\|w\|_p, \tag{2.6}$$

which was solved by the smoothing Barzilai-Borwein step gradient (SBBG) method. However, this model is a nonconvex optimization problem. Compared with convex optimization algorithms, nonconvex optimization algorithms lack convergence theoretical properties and robustness.

3. Proximal Sparse Beamformer Design

Though the solution of (2.5) is sparse, it may be not stable, due to the noise in getting the elements data of A and b, and the ill-condition of the matrix A. Moreover, since $m \gg n$, some rows are 'redundant'. Hence, we propose to separate the data into two groups, i.e., we draw some rows of A to form $A_1\in\mathbb{R}^{m_1\times n}$ and some rows to form $A_2\in\mathbb{R}^{m_2\times n}$, where $m_1\geq n$, $m_2\geq n$, and $m_1+m_2\geq m$. The first group is used to 'predict' a proximal center which will be used later to enhance the stability of the model, denoted as

$$\overline{w} = (A_1^TA_1)^{-1}A_1^Tb, \tag{3.1}$$

i.e., the solution of the 'partial' least squares

$$\min_{w\in\mathbb{R}^n} \frac{1}{2}\|A_1w-b\|^2.$$

Combining the proximal term to sparse model (2.5), we have the whole model

$$\min_{w\in\mathbb{R}^n} F(w)=\frac{1}{2}\|A_2w-b\|^2+\frac{\alpha}{2}\|w-\overline{w}\|^2+\gamma\|w\|_1, \tag{3.2}$$

where $\alpha>0$ and $\gamma>0$ are parameters, balancing the three terms. Model (3.2) is a composite optimization problem, whose objective function consists of three terms. The first one is a least squares term with 'partial' data, which reflects the data fidelity. The second term is a proximal term, which is used to enhance the stability of the model. The last term is the ℓ_1 term, which is used to improve the sparsity of the solution. Note also that the objective function is strongly convex optimization, and many algorithms can be used to solve it. In the following, we introduce three splitting type algorithms which are favorable for large-scale problems.

The first algorithm is the alternating direction method of multiplier, known as ADMM, which is efficient for solving the linearly constrained convex optimization problem with two blocks of variables [36]. To make the model (3.2) suitable for using ADMM, we first reformulate it into

$$\begin{aligned}\min_{w,x\in\mathbb{R}^n}\ & f(w)+g(x)\\ \text{s.t.}\quad & x=w,\end{aligned} \tag{3.3}$$

where $f(w)=\frac{1}{2}\|A_2w-b\|^2$ and $g(x)=\frac{\alpha}{2}\|x-\overline{w}\|^2+\gamma\|x\|_1$. The augmented Lagrangian function of (3.3) is

$$\mathscr{L}_\beta(x,w,\lambda) \; = \; \gamma\|x\|_1+\tfrac{\alpha}{2}\|x-\overline{w}\|^2-\lambda^T(x-w)+\frac{\beta}{2}\|x-w\|^2+\frac{1}{2}\|A_2w-b\|^2,$$

and the iterative scheme of ADMM is

$$\begin{cases} x^{k+1}=\underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \;\mathscr{L}_\beta(x,w^k,\lambda^k), \\ w^{k+1}=\underset{w\in\mathbb{R}^n}{\operatorname{argmin}} \;\mathscr{L}_\beta(x^{k+1},w,\lambda^k), \\ \lambda^{k+1}=\lambda^k-\beta(x^{k+1}-w^{k+1}). \end{cases} \tag{3.4}$$

The x-subproblem in (3.4) is just the soft-thresholding operation $S_{\frac{\gamma}{\alpha+\beta}}$, which is defined as

$$S_{\frac{\gamma}{\alpha+\beta}}(x)=\begin{cases} x-\dfrac{\gamma}{\alpha+\beta}, & x>\frac{\gamma}{\alpha+\beta}; \\ 0, & |x|\le\frac{\gamma}{\alpha+\beta}; \\ x+\dfrac{\gamma}{\alpha+\beta}, & x<-\frac{\gamma}{\alpha+\beta}. \end{cases}$$

The w-subproblem in (3.4) is the least squares

$$w^{k+1}=\underset{w\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|A_2w-b\|^2+\frac{\beta}{2}\|w-x^{k+1}\|^2+w^T\lambda^k,$$

or equivalently,

$$w^{k+1}=(\beta I+A_2^TA_2)^{-1}(-\lambda^k+\beta x^{k+1}+A_2^Tb),$$

which is well-defined since the matrix $\beta I+A_2^TA_2$ is positive definite. When the matrix A_2 has some properties, e.g., it is sparse or circulated, this problem can be settled down at low cost. In a word, the explicit iterative scheme of ADMM for solving (3.2) is

$$\begin{cases} x^{k+1}=S_{\frac{\gamma}{\alpha+\beta}}(\dfrac{\beta}{\alpha+\beta}w^k+\dfrac{1}{\alpha+\beta}\lambda^k+\dfrac{\alpha}{\alpha+\beta}\overline{w}); \\ w^{k+1}=(\beta I+A_2^TA_2)^{-1}(-\lambda^k+\beta x^{k+1}+A_2^Tb); \\ \lambda^{k+1}=\lambda^k-\beta(x^{k+1}-w^{k+1}), \end{cases} \tag{3.5}$$

and we describe the detailed steps in Algorithm 3.1.

The second splitting type method we introduced here, the forward-backward splitting algorithm (FBSM) [40, 42], can also be applied for solving (3.2). Let

$$g(w)=\gamma\|w\|_1,$$

and

$$f(w)=\frac{1}{2}\|A_2w-b\|^2+\frac{\alpha}{2}\|w-\overline{w}\|^2.$$

FBSM generates the iterative sequence via the recursion

$$w^{k+1}=(I+c_k\partial g)^{-1}(I-c_k\nabla f)(w^k),$$

where I is identity operator and c_k is a constant. We describe the details of FBSM in Algorithm 3.2.

Algorithm 3.1 Alternating Direction Method of Multipliers

1: Give matrix $A \in \mathbb{R}^{m\times n}$, vector b, and maximum iterations MaxIter. Given parameters $\beta, \gamma, \alpha \geq 0$, absolute error tolerance AbsTOL, and relative error tolerance RelTOL. Form A_1 and A_2 from A, and let (w^0, λ^0) be any initial point and set $k := 0$.

2: Compute $\overline{w} = (A_1^T A_1)^{-1} A_1^T b$.

3: **while** $k <$ MaxIter **do**

4: Compute

$$x^{k+1} = S_{\frac{\gamma}{\alpha+\beta}}\left(\frac{\beta}{\alpha+\beta} w^k + \frac{1}{\alpha+\beta}\lambda^k + \frac{\alpha}{\alpha+\beta}\overline{w}\right),$$

where S is the soft thresholding operator.

5: Compute

$$w^{k+1} = (\beta I + A_2^T A_2)^{-1}(-\lambda^k + \beta x^{k+1} + A_2^T b).$$

6: Compute

$$\lambda^{k+1} = \lambda^k - \beta(x^{k+1} - w^{k+1}).$$

7: Compute p^{k+1} as the primal residual and d^{k+1} as the dual residual at iteration $k+1$

$$\begin{aligned} p^{k+1} &= x^{k+1} - w^{k+1}, \\ d^{k+1} &= -\beta(w^{k+1} - w^k). \end{aligned}$$

8: Compute eps^P as primal feasibility tolerance and eps^D as dual feasibility tolerance

$$\begin{aligned} eps^P &= \sqrt{m}\ \text{AbsTOL} \\ &\quad + \max\{\|x^{k+1}\|, \|w^{k+1}\|, \|b\|\}\ \text{RelTOL}, \\ eps^D &= \sqrt{n}\ \text{AbsTOL} + \|\lambda^{k+1}\|\ \text{RelTOL}. \end{aligned}$$

9: **if** $\|p^{k+1}\| < eps^P$ and $\|d^{k+1}\| < eps^D$ **then** STOP and RETURN w^{k+1}.

10: **end if**

11: Set $k := k+1$.

12: **end while**

Algorithm 3.2 Forward-backward Splitting Method

1: Give matrix $A \in \mathbb{R}^{m\times n}$, vector b, and maximum iterations MaxIter. Give parameters $\beta, \gamma, \alpha \geq 0$, and error tolerance TOL. Form A_1 and A_2 from A, and let w^0 be any initial point and set $k := 0$.

2: Compute $\overline{w} = (A_1^T A_1)^{-1} A_1^T b$

3: **while** $k <$ MaxIter **do**

4: Compute

$$\begin{aligned} u &= w^k - c_k A_2^T (A_2 w^k - b) - \alpha c_k (w^k - \overline{w}), \\ w^{k+1} &= S_{\gamma c_k}(u), \end{aligned}$$

where S is a soft thresholding operator.

5: **if** $\|w^{k+1} - w^k\| <$ TOL **then** STOP and RETURN w^{k+1}.

6: **end if**

7: Set $k := k+1$.

8: **end while**

The Peaceman-Rachford splitting algorithm (PRSM) [40, 44] is also an effective algorithm for the minimization of the sum of two functions and the iterative scheme is $w^{k+1} = Pw^k$, where

$$P = (2(I + c_k\nabla f)^{-1} - I)(2(I + c_k\partial g)^{-1} - I).$$

Note that, without further assumptions, P is only guaranteed to be nonexpansive. So the iteration of PRSM is not necessarily convergent. Although PRSM is usually more efficient, we introduce the Douglas-Rachford splitting algorithm (DRSM) [39, 40] here as the third splitting type method. DRSM generates the iterative sequence via $w^{k+1} = Qw^k$, where

$$Q \;=\; (I + c_k\nabla f)^{-1}(2(I + c_k\partial g)^{-1} - I) + \left(I - (I + c_k\partial g)^{-1}\right).$$

Algorithm 3.3 Douglas-Rachford Splitting Method

Give matrix $A \in \mathbb{R}^{m\times n}$, vector b, and maximum iterations MaxIter, and give parameters $\beta, \gamma, \alpha \geq 0$. Form A_1 and A_2 from A and let w^0 be any initial point and set $k := 0$.
Compute $\overline{w} = (A_1^T A_1)^{-1} A_1^T b$.
while $k <$ MaxIter **do**
  Compute
$$u^k = S_{c_k\gamma}\left(w^k\right)$$
  where S is a soft thresholding operator.
  if $0 \in \nabla f(u^k) + \partial g(u^k)$ **then** STOP and RETURN u^k.
  end if
  Compute
$$v^k = \left(c_k A_2^T A_2 + c_k\alpha I + I\right)^{-1} p^k,$$
  where
$$p^k = c_k A_2^T b + c_k\alpha\overline{w} + 2u^k - w^k.$$
  Compute
$$w^{k+1} = v^k + w^k - u^k,$$
  and set $k := k+1$.
end while

The convergence of Algorithm 3.2 and Algorithm 3.3 are summarized in Appendix B.

Note that all these splitting algorithms can make full use of the structure of the problem (3.2). Algorithm 3.1, Algorithm 3.2 and Algorithm 3.3 are certainly the most well-known splitting algorithms and the splitting procedures of these two methods are interesting [40].

4. Numerical Experiments

In this section, the microphone array system setting is similar to the setting in [43].

4.1. Microphone array system.

The passband region is defined as

$$\Omega_P \;=\; \{(\boldsymbol{r}, f)|x = 1m, |y-4| \leq 0.4m, z = 1.5m, 0.5\text{kHz} \leq f \leq 2.0\text{kHz}\}.$$

We define the stopband regions as $\Omega_S = \Omega_1 \cup \Omega_2 \cup \Omega_3$, where

$$\begin{aligned}
\Omega_1 &= \{(\boldsymbol{r}, f)|x = 1m, |y-4| \leq 0.4m, z = 1.5m, 2.5\text{kHz} \leq f \leq 4.0\text{kHz}\},\\
\Omega_2 &= \{(\boldsymbol{r}, f)|x = 1m, 1.5m \leq |y-4| \leq 3m, z = 1.5m, 0.5\text{kHz} \leq f \leq 2.0\text{kHz}\},\\
\Omega_3 &= \{(\boldsymbol{r}, f)|x = 1m, 1.5m \leq |y-4| \leq 3m, z = 1.5m, 2.5\text{kHz} \leq f \leq 4.0\text{kHz}\}.
\end{aligned}$$

In the passband region Ω_P, we define the desired response function as

$$G_d(r,f) = \exp\left\{-2\pi i f\left(\frac{\|r-r_0\|}{c} + \frac{K-1}{2f_s}\right)\right\},$$

where r_0 is the space point of the microphone and let $f_s = 8000$Hz. We discrete Ω_P and Ω_S into a grid of 30×30 frames, and we use a 120×120 grid to evaluate the beamforming performance.

4.2. **The performance of algorithms for solving model** (3.2)**.** Model (3.2) is an unconstrained optimization problem. Many algorithms for solving this model have been mentioned above. In this subsection, some of these algorithms are applied to solve the proximal sparse beamformer design model. We first use the full information of A in this subsection, that is to say, $A_1 = A_2 = A$.

We compare three different algorithms and give the detailed iterative scheme. They are alternating direction method of multipliers, forward-backward splitting method, and Douglas-Rachford splitting method. In Figure 1, we show the changes of $\log(\|x^{k+1}-x^k\|)$ and $\log(F(x^k - F^*))$ with the iteration, and we can conclude that ADMM performs better numerically than DRSM and FBSM on the number of iterations. Meanwhile, from Table 1, FBSM is the best in terms of time. This is due to the simplicity of solving subproblems in FBSM. We also compare these three methods to SBBG method mentioned before which is adopted for solving the sparse beamformer design model (2.6). All these methods are used to solve the models with the optimal parameters. Due to the nonconvexity of the model, the SBBG method is far inferior to the tailed methods proposed in this paper. The performance of the SBBG method in detail can be found in [43].

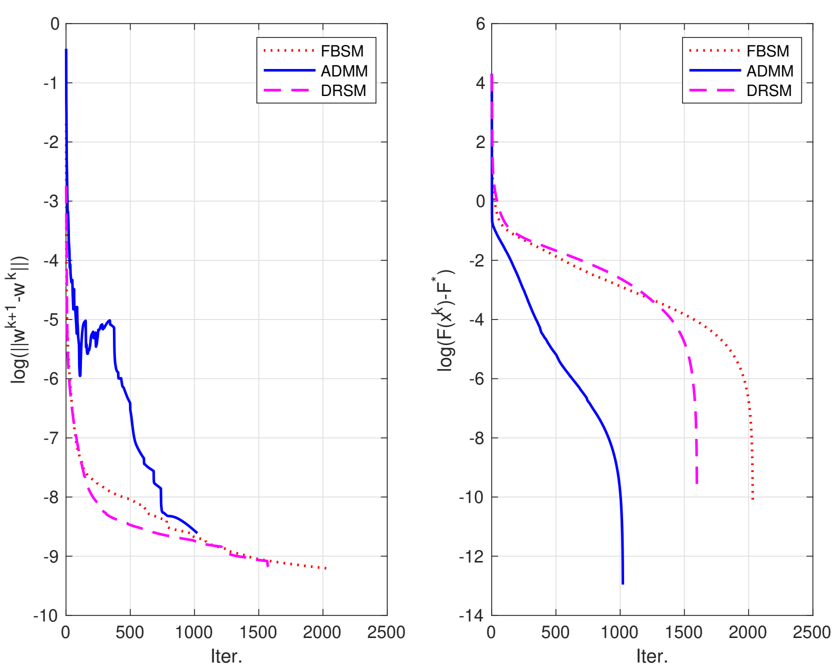

FIGURE 1. The performance of three different algorithms for solving the model.

TABLE 1. The CPU time spent by different methods.

Method	ADMM	FBSM	DRSM	SBBG
Time(s)	4.3730	1.9483	4.4862	17.293

4.3. **Overall performance of proximal sparse beamformer design.** In the following numerical experiments, we set the number of microphone elements $M = 7$ and the length of FIR filter $K = 20$.

The overall performance of least squares beamformer design is shown in Figure 2. The performance of sparse beamformer design in [43] and proximal sparse beamformer design are plotted in Figure 3 and Figure 4, respectively. From the results shown in the above figures, we can conclude that the performance of proximal sparse beamformer design is more close to the performance of least squares beamformer design than the one of sparse beamformer design. Also, the distribution of filter coefficients of proximal sparse beamformer design is more close to that of least squares beamformer design.

We use the number of zeros of the coefficient, the passband gain, the passband ripple, and the stopband ripple to evaluate the performance of the beamformer design. We say that the design performs well numerically when the passband gain is large and the passband ripple and the stopband ripple are small. Our proposed model not only maintains the effect which is close to that of the least squares solution, but also ensures the sparsity. In the tables, ”N_0” denotes the number of zeros, ”PG” denotes the passband gain, ”PR” denotes the passband ripple, and ”SR” denotes the stopband ripple. The results have been shown in Table 2. When $\gamma = 1, \alpha = 0$, the solution is the solution of sparse beamformer design. When $\gamma = 0, \alpha = 0$, the solution is just the least squares solution. The number of zeros of filter in proximal sparse beamformer design is same as that in proximal sparse beamformer design. However, both the passband ripples and stop ripples of proximal sparse beamformer design are close to the least squares beamformer design.

TABLE 2. Performance evaluation of enhanced signal with $\beta = 1$.

γ	α	N_0	PG	PR	SR
1	0	45	0.9836	0.3013	-13.0039
0	0	0	1.0033	0.1155	-19.4824
1.781	0.01	46	0.9768	0.3085	-12.2133
1.091	0.001	48	0.9821	0.3014	-12.9140
0.144	0.001	45	1.0039	0.1776	-18.1781
0.147	0.001	45	1.0028	0.1683	-17.0112
0.1464	0.001	45	1.0030	0.1709	-17.1657

In Figure 5, we give the comparison results between sparse beamformer design and least squares beamformer design when $\alpha = 0$. The case when $\alpha = 0.001$ has been shown in Figure 6. From the figures, we can conclude that the proximal sparse design has the best performance when $\alpha = 0.001$ and $\gamma = 0.14$ probably.

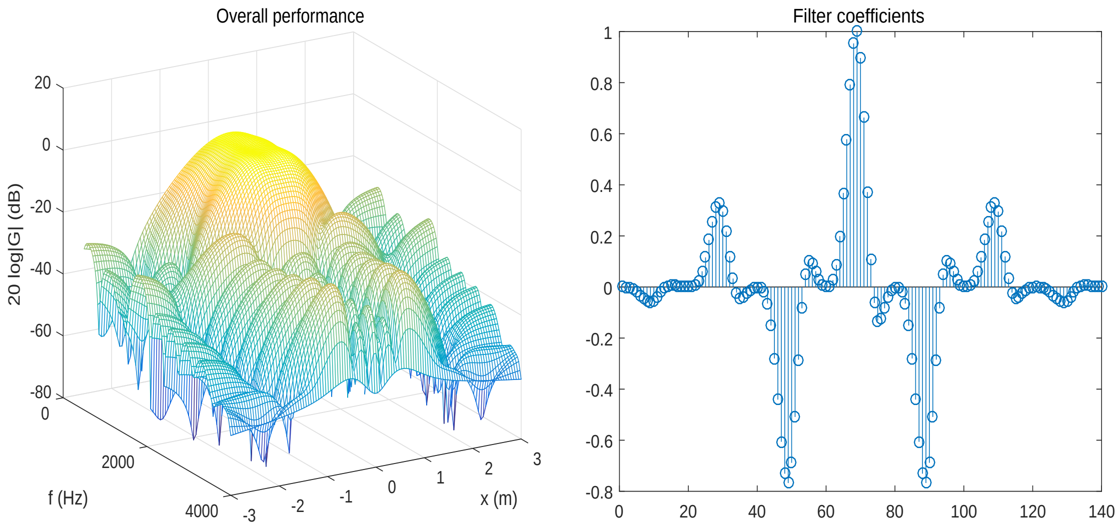

FIGURE 2. Overall performance of the least squares beamformer design.

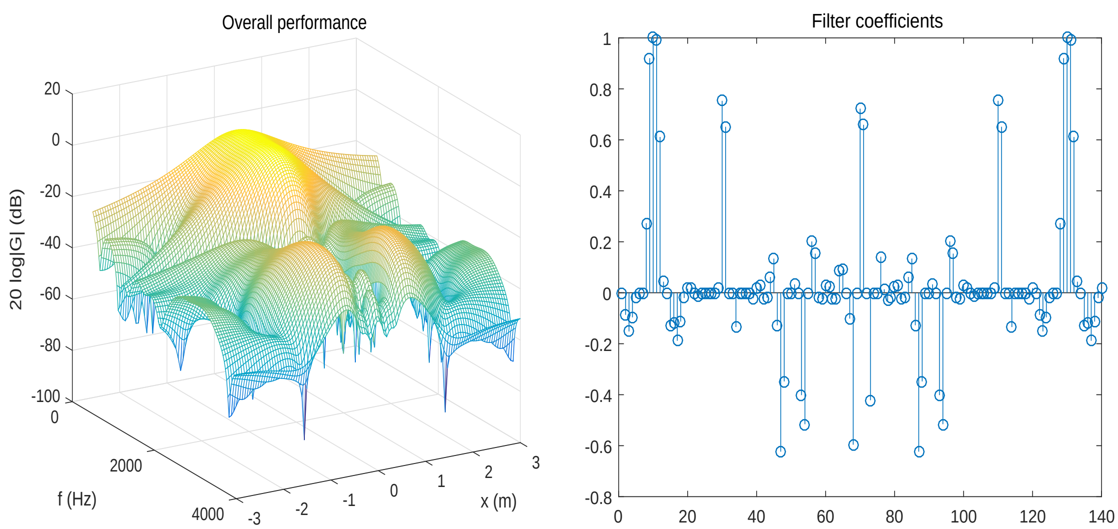

FIGURE 3. Overall performance of the sparse beamformer design.

4.4. **An improvement to the model.** The proximal sparse beamformer design model concludes two optimal problems. The solution of model (2.4) is just the prediction point of model (3.2), so we do not need to obtain the exact solution of (2.4). Moreover, $A \in \mathbb{R}^{m\times n}$ in (2.4) where $m \gg n$ and it has column full rank, in other words, A has a mass of linearly dependent rows. It will take a lot of unnecessary memory and time cost if we use all information of A. If we take $A_1 = A$, the least squares model will be easy to overfit. A natural idea to improve the whole model is to use the part information of A for preventing this. We obtain A_1 by choosing some rows of A and obtain the least squares solution as the prediction point. And then what we care about is how many rows we pick and how to pick. In our experiment, $A \in \mathbb{R}^{1800\times 140}$ and we try a couple of ways to pick the rows. The performance of different selection strategies has been shown below. The performance of the four strategies has been shown below.

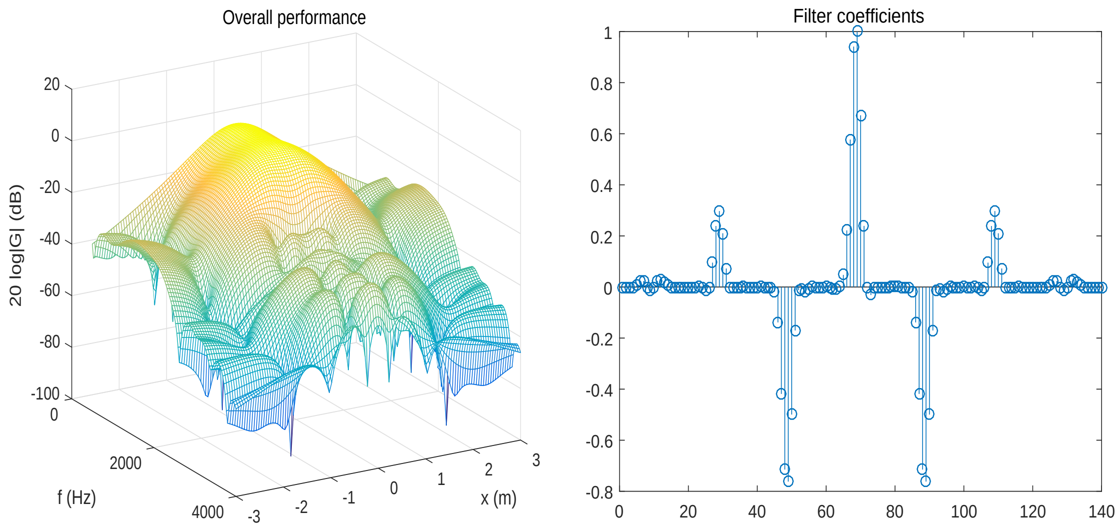

FIGURE 4. Overall performance of the proximal sparse beamformer design.

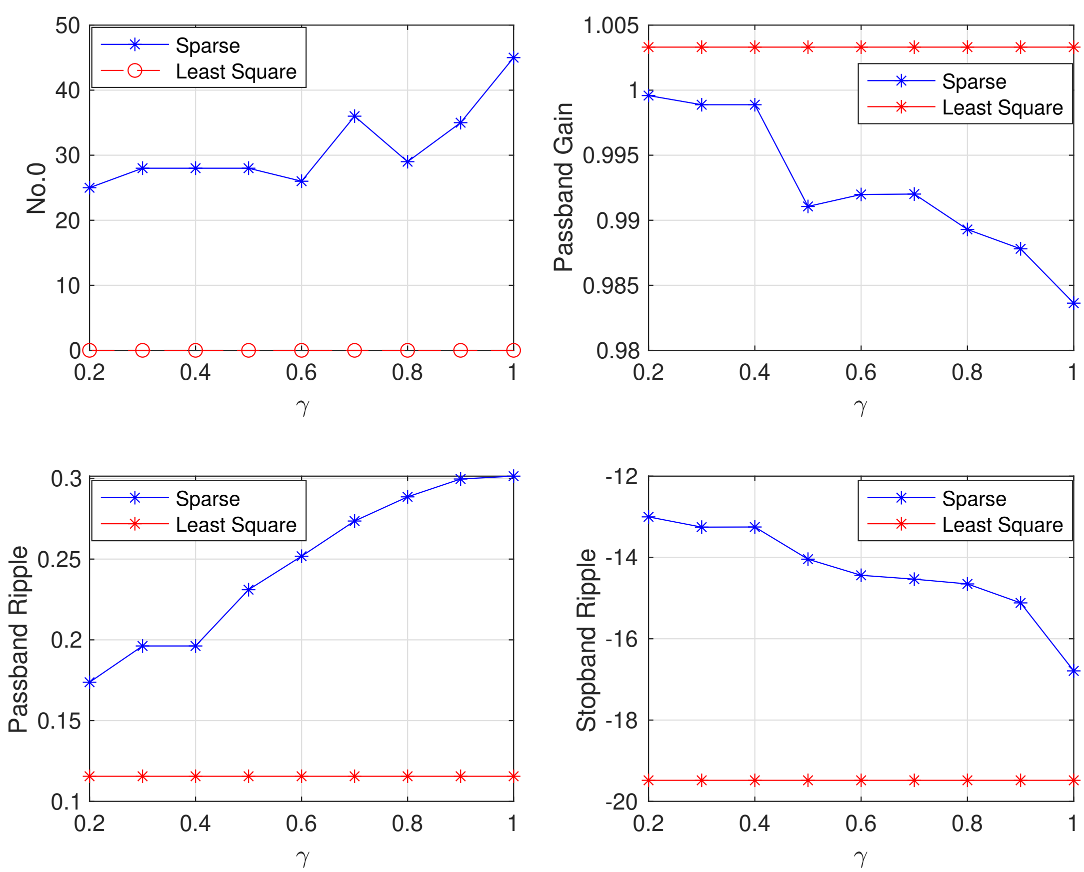

FIGURE 5. The performance comparison between sparse model and least squares model.

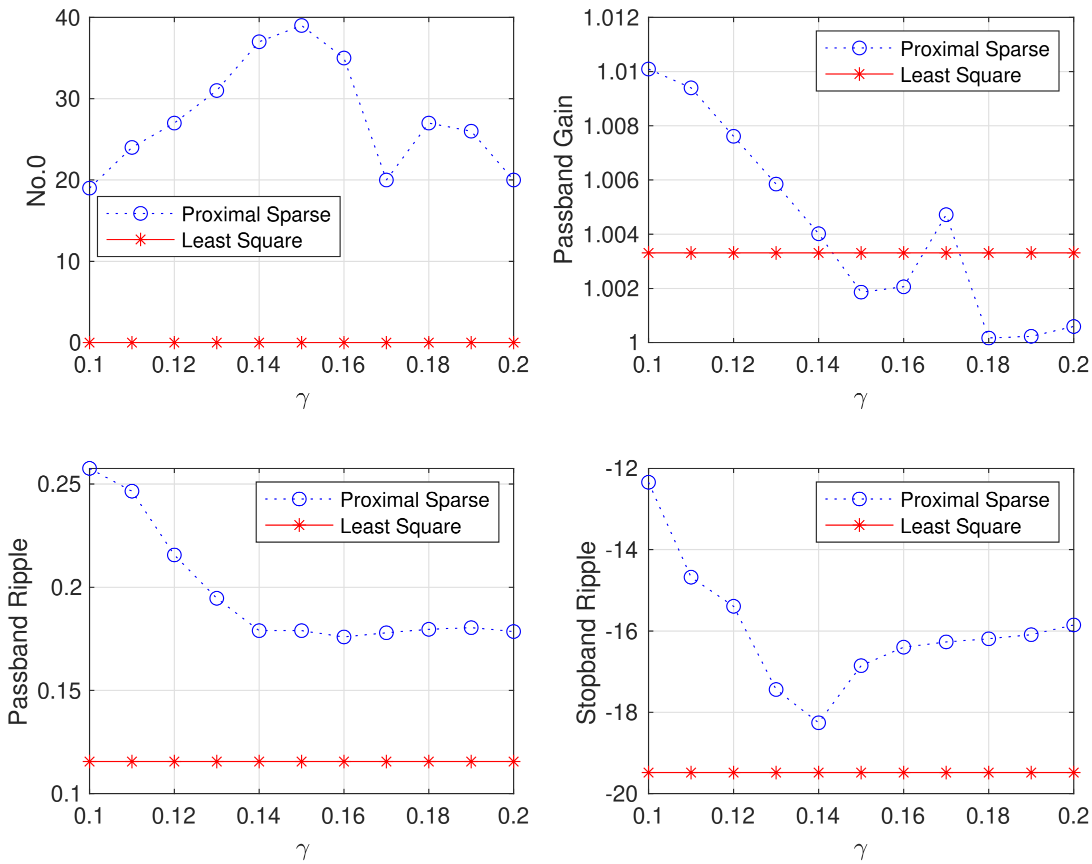

FIGURE 6. The performance comparison between proximal sparse model and least squares model when $\alpha = 0.001$.

We first consider some simple cases in that we choose 900 rows of A. Let us take the first 900 rows, the last 900 rows, odd rows, and even rows, respectively. The performance of the above four strategies has been demonstrated in Table 3. From Table 3, the strategy of selecting partial columns in model (2.4) does reduce the solution time. And as the scale of the problem grows, the improvement in time will be more obvious.

TABLE 3. Performance evaluation of enhanced signal with $\gamma = 0.144$ and $\alpha = 0.001$.

rows	N_0	PG	PR	SR	Time(s)
1800	45	1.0039	0.1776	-18.1781	1.948
first 900	26	1.0016	0.1775	-16.4650	1.763
last 900	45	1.0028	0.1689	-17.0608	1.788
odd 900	30	1.0049	0.1909	-18.5799	1.840
even 900	32	1.0042	0.1847	-18.4217	1.809

From Table 3, the choice of the last 900 rows performs best among the four strategies. The effect of picking the last 900 rows of A is similar to that using full rows and picking the last 900 rows takes less cost of memory and time in arithmetic operations.

Then we choose some rows of A randomly to obtain A_1. We pick 500 rows and 1000 rows respectively and Table 4 shows their performance.

TABLE 4. Performance evaluation of enhanced signal with $\gamma = 0.144$ and $\alpha = 0.001$.

A_1	N_0	PG	PR	SR
500 rows	40	1.0034	0.1777	-17.7567
500 rows	35	1.0070	0.2116	-15.8923
500 rows	40	1.0038	0.1781	-17.9637
500 rows	39	1.0022	0.1761	-16.9803
500 rows	37	1.0026	0.1738	-17.1818
1000 rows	38	1.0022	0.1749	-17.0794
1000 rows	39	1.0059	0.2061	-17.4086
1000 rows	44	1.0039	0.1789	-17.6979
1000 rows	44	1.0039	0.1784	-17.5778
1000 rows	37	1.0045	0.1872	-18.5273

From Table 4, we can conclude that picking part rows of A is a viable strategy. It makes a great performance similar to using the full rows and saves a lot of memory and computation time. However, due to randomness, the effect of each experiment is also different. Overall, picking the part rows of the full rank matrix A is a choice to improve the efficiency of solving the model.

5. CONCLUSION

In this paper, we first proposed the proximal sparse beamformer design model which consists of two stages. We just used ℓ_1-regularization to characterize sparsity rather than the other nonconvex models. We also added a proximal term to the sparse beamformer design model so that the solution is sparse and robust. Compared to the existing work, our work greatly improved the efficiencies and effectiveness. The numerical experiments indicated that the proximal sparse beamformer design has a good performance in applications.

Acknowledgements
Xingju Cai was supported by the NSFC grant Nos. 12131004, 11871279; Deren Han was supported by the NSFC grant Nos. 12126603, 12131004. Zhibao Li was supported by the NSFC grant No. 12271526, Natural Science Foundation of Hunan Province (Nos. 2021JJ30167, 2022JJ30675), and the Natural Science Foundation of Changsha (No. kq2202068).

APPENDIX A. ITERATIVE SCHEME

In the numerical experiments, we compare the performance of different algorithms. In this section, we present the explicit iterative scheme of these algorithms in detail.

In (3.5), the x-minimization subproblem is

$$\begin{aligned} x^{k+1} &= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ \mathscr{L}_\beta(x, w^k, \lambda^k) \\ &= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \left\{ \gamma\|x\|_1 + \frac{\alpha}{2}\|x-\overline{w}\|^2 - (\lambda^k)^T(x-w^k) + \frac{\beta}{2}\|x-w^k\|^2 \right\} \\ &= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \left\{ \|x\|_1 + \frac{\alpha+\beta}{2\gamma}\|x - \frac{\beta}{\alpha+\beta}w^k - \frac{1}{\alpha+\beta}\lambda^k - \frac{\alpha}{\alpha+\beta}\overline{w}\|^2 \right\} \\ &= S_{\frac{\gamma}{\alpha+\beta}}\left(\frac{\beta}{\alpha+\beta}w^k + \frac{1}{\alpha+\beta}\lambda^k + \frac{\alpha}{\alpha+\beta}\overline{w} \right). \end{aligned}$$

The the w-minimization subproblem is

$$\begin{aligned} w^{k+1} &= \underset{w\in\mathbb{R}^n}{\operatorname{argmin}}\ \mathscr{L}_\beta(x^{k+1}, w, \lambda^k), \\ &= \underset{w\in\mathbb{R}^n}{\operatorname{argmin}}\ \left\{ (\lambda^k)^T w + \frac{\beta}{2}\|x^{k+1}-w\|^2 + \frac{1}{2}\|A_2 w - b\|^2 \right\}. \end{aligned} \tag{5.1}$$

Form the optimality condition of (5.1), we have

$$\lambda^k + \beta(w^{k+1} - x^{k+1}) + A_2^T(Aw^{k+1} - b) = 0.$$

Clearly, the iterative scheme of w-minimization subproblem is

$$w^{k+1} = (\beta I + A_2^T A_2)^{-1}(-\lambda^k + \beta x^{k+1} + A_2^T b).$$

In Algorithm 3.2 and Algorithm 3.3, we set

$$f(w) = \frac{1}{2}\|A_2 w - b\|^2 + \frac{\alpha}{2}\|w - \overline{w}\|,$$

and $g(w) = \gamma\|w\|_1$. Then

$$\nabla f(w) = A_2^T(A_2 w - b) + \alpha(w - \overline{w}),$$

and $(I + c_k\partial g)^{-1}(w) = S_{c_k\gamma}(w)$, where S is the soft thresholding operator. The explicit form of $(I + c_k\nabla f)^{-1}$ is obtained by the following discuss

$$(I + c_k\nabla f)^{-1}(w) = \underset{t\in\mathbb{R}^n}{\operatorname{argmin}} \left\{ c_k f(t) + \frac{1}{2}\|t - w\|^2 \right\}.$$

Form the above optimality condition and the definition of f, we obtain

$$t = \left(c_k A_2^T A_2 + c_k\alpha I + I\right)^{-1}\left(c_k A_2^T b + c_k\alpha\overline{w} + w\right).$$

That is to say

$$(I + c_k\nabla f)^{-1}(w) = \left(c_k A_2^T A_2 + c_k\alpha I + I\right)^{-1}\left(c_k A_2^T b + c_k\alpha\overline{w} + w\right).$$

So the explicit iterative scheme of FBSM is

$$\begin{aligned} w^{k+1} &= (I + c_k\partial g)^{-1}(I - c_k\nabla f)(w^k) \\ &= (I + c_k\partial g)^{-1}\left(w^k - c_k A_2^T(A_2 w^k - b) - c_k\alpha(w^k - \overline{w})\right) \\ &= S_{c_k\gamma}\left(w^k - c_k A_2^T(A_2 w^k - b) - c_k\alpha(w^k - \overline{w})\right). \end{aligned}$$

The explicit iterative scheme of DRSM is

$$\begin{aligned} w^{k+1} &= \Big((I+c_k\nabla f)^{-1}\left(2\left(I+c_k\partial g\right)^{-1}-I\right)+\left(I-\left(I+c_k\partial g\right)^{-1}\right)\Big)\left(w^k\right) \\ &= \Big((I+c_k\nabla f)^{-1}\left(2\left(I+c_k\partial g\right)^{-1}-I\right)\Big)\left(w^k\right)+w^k-\left(I+c_k\partial g\right)^{-1}\left(w^k\right) \\ &= (I+c_k\nabla f)^{-1}\left(2S_{c_k\gamma}(w^k)-w^k\right)+w^k-S_{c_k\gamma}(w^k) \\ &= \left(c_kA_2^TA_2+c_k\alpha I+I\right)^{-1}\left(c_kA_2^Tb+c_k\alpha\overline{w}+2S_{c_k\gamma}(w^k)-w^k\right)+w^k-S_{c_k\gamma}(w^k). \end{aligned}$$

Appendix B. Convergence Proof

In this section, we present the convergence proof of the algorithms adopted in this paper. The convergence of Algorithm 3.1 is well understood. Here we cite a result from [36].

Theorem 5.1. *Suppose that $\{w^k\}$ is the iterative sequence generated by Algorithm 3.1. Then $\{w^k\}$ converges to some w^* which is the solution of the proximal sparse model if the solution set is not empty.*

In fact, due to the structure of the objective function in (3.2), the sequence generated by the algorithm converges in a linear rate [45].

Theorem 5.2. *Let $\{w^k\}$ be the sequence generated by the ADMM scheme for (3.2). When the iterative w^k is close enough to the solution set, we can conclude that the algorithm has a local linear convergence [45].*

Lemma 5.1. [46] *Let the operator $T:\mathbb{R}^n\to\mathbb{R}^n$. If T is firmly nonexpansive, then, for all $x,y\in\mathbb{R}^n$,*

$$\|Tx-Ty\|^2+\|(I-T)x-(I-T)y\|^2\le\|x-y\|^2.$$

Theorem 5.3. *Let S^* be the solution set of the proximal sparse model, and assume that $S^*\neq\emptyset$. Suppose that $\{w^k\}$ is the iterative sequence generated by Algorithm 3.2, and $f:\mathbb{R}^n\to\mathbb{R}^n$ has L-Lipschitz gradient. If $c_k=c$ and $0<c<\frac{2}{L}$, then $\{w^k\}$ converges to some w^* which is the solution of the proximal sparse model.*

Proof. The iterative scheme of FBSM is

$$w^{k+1}=(I+c\partial g)^{-1}(I-c\nabla f)(w^k).$$

For any $w^*\in S^*$, we have $w^*=(I+c\partial g)^{-1}(I-\nabla f)(w^*)$. The proximal operator $(I+c\partial g)^{-1}$ is firmly nonexpansive. From the fact that $J=(I+c\partial g)^{-1}$ is firmly nonexpansive operator and Lemma 5.1, we can obtain

$$\begin{aligned} &\|w^{k+1}-w^*\|^2 \\ =&\|J(I-c\nabla f)w^k-J(I-c\nabla f)w^*\|^2 \\ \le&\|(I-c\nabla f)w^k-(I-c\nabla f)w^*\|^2-\|(I-J)(I-c\nabla f)w^k-(I-J)(I-c\nabla f)w^*\|^2. \end{aligned}$$

Because f has L-Lipschitz gradient, we can conclude that ∇f is $\frac{1}{L}$-co-coercive, i.e., for any $x,y\in\mathbb{R}^n$,

$$\langle x-y,\nabla f(x)-\nabla f(y)\rangle\ge\frac{1}{L}\|\nabla f(x)-\nabla f(y)\|^2.$$

From this, we have

$$\begin{aligned}&\|(I-c\nabla f)w^k-(I-c\nabla f)w^*\|^2\\ &=\|w^k-w^*\|^2-2c\langle w^k-w^*,\nabla f(w^k)-\nabla f(w^*)\rangle+c^2\|\nabla f(w^k)-\nabla f(w^*)\|^2\\ &\leq\|w^k-w^*\|^2-c\left(\frac{2}{L}-c\right)\|\nabla f(w^k)-\nabla f(w^*)\|^2.\end{aligned}$$

From the above statement, we can obtain

$$\begin{aligned}&\|w^{k+1}-w^*\|^2\\ &\leq\|w^k-w^*\|^2-c\left(\frac{2}{L}-c\right)\|\nabla f(w^k)-\nabla f(w^*)\|^2-\|w^k-c\nabla f(w^k)-w^{k+1}+c\nabla f(w^*)\|^2.\end{aligned} \tag{5.2}$$

Thus $\{w^k\}$ is Fejér monotone with respect to S^* [46]. For any $w^*\in S^*$, $\lim_{k\to+\infty}\|w^k-w^*\|$ exists, and hence $\{w^k\}$ is bounded. From (5.2), we have

$$\lim_{k\to+\infty}\nabla f(w^k)=\nabla f(w^*),$$

and

$$\lim_{k\to+\infty}\left\|w^{k+1}-w^k+c\nabla f(w^k)-c\nabla f(w^*)\right\|=0.$$

From the iterative scheme, we obtain $w^k-w^{k+1}-c\nabla f(w^k)\in c\partial g(w^{k+1})$. Suppose that the bounded sequence $\{w^k\}$ has a convergent subsequence $\{w^{k_j}\}$ which converges to $\overline{w}$. From the continuity of ∇f, we have

$$\nabla f(\overline{w})=\nabla f(w^*). \tag{5.3}$$

From the monotonicity of ∂g, for any $u\in\partial g(w)$, we obtain

$$\left\langle\frac{1}{c}\left(w^{k_j-1}-w^{k_j}-c\nabla f(w^{k_j-1})\right)-u,w^{k_j}-w\right\rangle\geq 0.$$

Taking the limit of the above inequality as $j\to\infty$, we have

$$\langle-\nabla f(w^*)-u,\overline{w}-w\rangle\geq 0. \tag{5.4}$$

From the maximal monotonicity of ∂g and (5.4), we have $(\overline{w},-\nabla f(w^*))\in\text{gra}\,\partial g$, in other words,

$$-\nabla f(w^*)\in\partial g(\overline{w}). \tag{5.5}$$

Combining (5.3) and (5.5), we can conclude that $0\in\nabla f(\overline{w})+\partial g(\overline{w})$, so $\overline{w}\in S^*$ and $\lim_{k\to+\infty}\|w^k-\overline{w}\|$ exists. Because the subsequence $\{w^{k_j}\}$ converges to $\overline{w}$, we can conclude that

$$\lim_{k\to+\infty}\|w^k-\overline{w}\|=0.$$

Thus we complete the proof. □

Next we show the convergence of DRSM. In fact, the iterative scheme of DRSM can be written as $z^{k+1}=\frac{1}{2}(I+R_1R_2)z^k$, where $R_1=2(I+c\partial g)^{-1}-I$, $R_2=2(I+c\nabla f)^{-1}-I$, and $z^k=(I+c\nabla f)w^k$. Let $T=\frac{1}{2}(I+R_1R_2)$, where R_1R_2 is a nonexpansive operator. Thus T is firmly nonexpansive.

Theorem 5.4. *Let S^* be the solution set of the proximal sparse model, and assume that $S^* \neq \emptyset$. Suppose that $\{w^k\}$ is the iterative sequence generated by Algorithm 3.3. Let $z^k = (I + c\nabla f)w^k$. Then $\{z^k\}$ converges to z^* which belongs to the set of zeros of T. Furthermore, $w^k = (I + c\nabla f)^{-1} z^k$ converges to some w^* which is the solution of the proximal sparse model.*

Proof. The proof is similar to Theorem 5.3. Thus it is omitted. □

References

[1] S. Applebaum, Adaptive arrays, IEEE Trans. Antennas Propagation 24 (1976), 585-598.

[2] B.D. Van Veen, K.M. Buckley, Beamforming: A versatile approach to spatial filtering, IEEE ASSP Magazine 5 (1998) 4-24.

[3] O.L. Frost, An algorithm for linearly constrained adaptive array processing, Proc. IEEE 60 (1972), 926-935.

[4] M. Wax, Y. Anu, Performance analysis of the minimum variance beamformer, IEEE Trans. Signal Process. 44 (1996), 928-937.

[5] L. Griffiths, C. Jim, An alternative approach to linearly constrained adaptive beamforming, IEEE Trans. Antennas Propagation 30 (1982), 27-34.

[6] I. Cohen, Analysis of two-channel generalized sidelobe canceller (GSC) with post-filtering, IEEE Trans. Speech Audio Process. 11 (2003), 684-699.

[7] Z.G. Feng, K.F.C. Yiu, S.E. Nordholm, Placement design of microphone arrays in near-field broadband beamformers, IEEE Trans. Signal Process. 60 (2012), 1195-1204.

[8] Z. Li, K.F.C. Yiu, Z. Feng, A hybrid descent method with genetic algorithm for microphone array placement design, Appli. Soft Computing 13 (2013), 1486-1490.

[9] Z. Li, K.F.C. Yiu, Beamformer configuration design in reverberant environments, Engineering Applications of Artificial Intelligence, 47 (2016), 81-87.

[10] Q. Wang, S. Y. Low, Z. Li, et al., Sensor placement optimization of blind source separation in a wireless acoustic sensor network via hybrid descent methods, Appl. Acoustics 188 (2022), 108509.

[11] X. Wang, M. Amin, X. Cao, Analysis and design of optimum sparse array configurations for adaptive beamforming, IEEE Trans. Signal Process. 66 (2017), 340-351.

[12] S.A. Hamza, M.G. Amin, Hybrid sparse array beamforming design for general rank signal models, IEEE Trans. Signal Process. 67 (2019), 6215-6226.

[13] Z. Zheng, W. Wang, Y.D. Zhang, MISC array: A new sparse array design achieving increased degrees of freedom and reduced mutual coupling effect, IEEE Trans. Signal Process. 67 (2020), 1728-1741.

[14] Z. Zheng, T. Yang, W.Q. Wang, et al., Robust adaptive beamforming via coprime coarray interpolation, 169 (2020), 107382.

[15] P. Pal, P.P. Vaidyanathan, Nested arrays: A novel approach to array processing with enhanced degrees of freedom, IEEE Trans. Signal Process. 58 (2010), 4167-4181.

[16] M.G. Amin, X. Wang, Y.D. Zhang, et al., Sparse arrays and sampling for interference mitigation and DOA estimation in GNSS, Proceedings of the IEEE, 104 (2016), 1302-1317.

[17] D. Wei, Non-convex optimization for the design of sparse FIR filters, IEEE/SP 15th Workshop on Statistical Signal Processing, pp. 117-120, 2009.

[18] A. Jiang, H.K. Kwan, Y. Zhu, Peak-error-constrained sparse FIR filter design using iterative SOCP, IEEE Trans. Signal Process. 60 (2012), 4035-4044.

[19] T. Baran, D. Wei, A.V. Oppenheim, Linear programming algorithms for sparse filter design, IEEE Trans. Signal Process. 58 (2010), 1605-1617.

[20] D. Wei, A. V. Oppenheim, A branch-and-bound algorithm for quadratically-constrained sparse filter design, IEEE Trans. Signal Process. 61 (2013), 1006-1018.

[21] C. Rusu, B. Dumitrescu, Iterative reweighted L_1 design of sparse FIR filters, Signal Process. 92 (2012), 905-911.

[22] A. Jiang, H.K. Kwan, Y. Zhu, et al., Design of sparse FIR filters with joint optimization of sparsity and filter order, IEEE Trans. Circuits Systems I: Regular Papers, 62 (2015), 195-204.

[23] R. Matsuoka, S. Kyochi, S. Ono, et al., Joint sparsity and order optimization based on ADMM with non-uniform group hard thresholding, IEEE Transactions on Circuits and Systems I: Regular Papers 65 (2018), 1602-1613.
[24] E.J. Candes, The restricted isometry property and its implications for compressed sensing, Comptes Rendus Mathematique, 346 (2008), 589-592.
[25] E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies, IEEE Trans. Info. Theory 52 (2006), 5406-5425.
[26] E.J. Candes, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete, IEEE Trans. Info. Theory 52 (2006), 489-509.
[27] P.L. Combettes, J.C. Pesquet, Proximal thresholding algorithm for minimization over orthonormal bases, SIAM J. Optim. 18 (2007), 1351-1376.
[28] E.T. Hale, W. Yin, Y. Zhang, Fixed-point continuation for ℓ_1-minimization: Methodology and convergence, SIAM J. Optim. 19 (2008), 1107-1130.
[29] W. Yin, S. Osher, D. Goldfarb, et al., Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing, SIAM J. Imaging Sci. 1 (200), 143-168.
[30] J. Yang, Y. Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing, SIAM J. Sci. Comput. 33 (2011), 250-278.
[31] R.T. Rockafellar, Lagrange multipliers and optimality, SIAM Rev. 35 (1993), 183-238.
[32] P. Refaeilzadeh, L. Tang, H. Liu, Cross-validation, Encyclopedia of Database Systems 5 (2009), 532-538.
[33] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. 4 (2010), 40-79.
[34] R. Glowinski, On alternating direction methods of multipliers: A historical perspective. In: Fitzgibbon, W., Kuznetsov, Y., Neittaanmäki, P., Pironneau, O. (eds) Modeling, Simulation and Optimization for Science and Technology, Computational Methods in Applied Sciences, vol 34. Springer, Dordrecht, 2014.
[35] J. Eckstein, D.P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, Math. Program. 55 (1992), 293-318.
[36] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends in Mach. Learn. 3 (2011), 1-122.
[37] J. Eckstein, W. Yao, Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives, Pacific J. Optim. 11 (2015), 619-644.
[38] D.R. Han, A survey on some recent developments of alternating direction method of multipliers, J. Oper. Res. Soc. China 10 (2022), 1-52.
[39] J. Douglas, H.H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, Trans. Amer. Math. Soc. 82 (1956), 421-439.
[40] F. Facchinei, J.S. Pang, Finite-dimensional variational inequalities and complementarity problems, Springer, 2003.
[41] X.J. Cai, K. Guo, Z.M. Wu, D.R. Han, The Developments of proximal point algorithms, J. Oper. Res. Soc. China 10 (2022), 197-239.
[42] P.L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, SIAM J. Numer. Anal. 16 (1979), 964-979.
[43] Z. Li, K.F.C. Yiu, Y. Dai, On sparse beamformer design with reverberation, Appl. Math. Model. 58 (2018), 98-110.
[44] D.W. Peaceman, H.H. Rachford, The numerical solution of parabolic and elliptic differential equations, J. Soc. Indust. Appl. Math. 3 (1995), 28-41.
[45] D. Han, X. Yuan, Local linear convergence of the alternating direction method of multipliers for quadratic programs, SIAM J. Numer. Anal. 51 (2013), 3446-3457.
[46] H.H. Bauschke, P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, Springer, Cham, 2011.