sensors

MDPI

Article

On the Sparse Beamformer Design

Mingjie Gao [1], **Ka Fai Cedric Yiu** [2,*] **and Sven Nordholm** [3]

[1] School of Mathematical Sciences, South China Normal University, Guangzhou 510000 , China; mjgao@m.scnu.edu.cn
[2] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong, China
[3] Department of Electrical and Computer Engineering, Curtin University, Perth 6102, Australia; s.nordholm@curtin.edu.au
* Correspondence: macyiu@polyu.edu.hk; Tel.: +86-852-3400-8981

Received: 28 July 2018; Accepted: 27 September 2018; Published: 19 October 2018

Abstract: In designing acoustic broadband beamformers, the complexity can grow significantly when the number of microphones and the filter length increase. It is advantageous if many of the filter coefficients are zeroes so that the implementation can be executed with less computation. Moreover, the size of the array can also be pruned to reduce complexity. These problems are addressed in this paper. A suitable optimization model is proposed. Both array pruning and filter thinning can be solved together as a two-stage optimization problem to yield the final sparse designs. Numerical results show that the complexity of the designed beamformers can be reduced significantly with minimal effect on performance.

Keywords: beamformer design; signal enhancement; array pruning; sparse filters

1. Introduction

Beamforming is a spatial filtering technique used to enhance the required signal via a sensor array for directional signal transmission or reception [1,2]. With a fixed configuration, the steering vectors of the desired signals can be estimated together with their directions-of-arrival [3]. More elaborated physical signal propagation models have also be employed to describe complicated wave phenomena [4]. As a result, the beamformer design problem can be formulated as an optimization problem similar to the design of multidimensional digital filters. Various optimization methods, such as linear programming techniques [5], quadratic programming techniques [6], and second-order cone programming [7] have been applied. If the beamformers are applied in the near-field of the speaker, it becomes a broadband design problem and several optimization methods have been developed. These include the use of quadratic programming [8], multicriteria formulation [9], linear programming [10], and semi-definite programming [11,12]. We have also analytically investigated the performance limit of the optimization when the filter length is long and the number of microphones is large [13]. As the dimension of the beamformer increases, the optimization problems become large-scale and are difficult to handle, even with state-of-the-art optimization software. Additionally, the complexity of the beamforming system becomes very high.

In designing beamformers, it is advantageous to have filters with many zeroes. In this way, the implementation complexity can be reduced significantly. Therefore, the design of sparse beamforming filters is of great interest. In solving this l_0-norm problem, an often subproblem is to employ a l_1-norm as a linear relaxation of the original problem, and iterate on the number of zeroes via a successive thinning technique [14,15]. When the l_1-norm is employed, the problem becomes a non-strictly convex semi-infinite programming problem. For fast convergence and at the same time to reduce the magnitudes of unnecessary filter coefficients, a perturbation exchange algorithm is proposed. The idea is to add a small perturbation to the objective function, making the problem strictly convex. Under the influence of a small perturbation to the objective function, we obtain similar frequency response to the traditional method [10]. Moreover, an efficient exchange method was proposed for a convex semi-infinite programming problem in [16] which has low complexity. During the exchange process, global optima of the exchange points are not required for addition, but only those points with constraint violations. In dropping active points, the Lagrange multipliers are used to remove all inactive constraints. The new add–drop rule saves much computational time. This is particularly important for higher-dimensional design problems after discretization with a huge dimension in the constraint set.

In addition to the sparseness in the filters, it is also important to prune unnecessary microphones in order to reduce the size of the array without affecting the performance very much. Indeed, in the beamforming design problem, the array response changes significantly when the microphone positions change [17]. This problem has been addressed partially in the array thinning technique [18,19], which is essentially one-dimensional with application mainly in antenna design. A multi-dimensional array configuration design problem has been proposed in [20,21] which allows the microphone positions to be optimized in the multi-dimensional spatial domain. As an extension, in order to prune unnecessary microphones, it is possible to formulate the sparse configuration design problem by adding a penalty term to the configuration design problem in order to drive redundant microphones together for successive pruning.

In this paper, we formulate and propose a novel algorithm for the sequential pruning of a microphone array, and at the same time tackle the sparse design of beamforming filters through successive thinning. We formulate the optimization problem based on the minimax approach, although the method should work equally well for the least-squares or the minimum mean output energy approaches [22,23]. Numerical results have shown that it is possible to have a region of approximately equal performance when the sparsity increases, in both filters and microphones. Therefore, the proposed methods can indeed reduce the complexity of the beamformer array and filters significantly without performance degradation. In the following sections, the beamformer design problem is first described in Section 2. Then, the formulation of the microphone array pruning problem is presented in Section 3 together with the perturbation exchange algorithm. Finally, numerical results are given in Section 4.

2. Beamformer Design Formulation

In a typical environment, a beamformer contains a series of microphones placed in pre-defined locations. Behind each microphone, there is an FIR filter attached for processing the received sound signals [9]. However, in the beamforming design problem, the array response may change significantly when the microphone positions change. Thus, we should not ignore the configuration of the microphone array. Let the beamformer have N microphones and let each FIR filter have L taps. Denote the position vector of the source signal by $\boldsymbol{s}$ and the position vector of the i-th microphone by $\boldsymbol{r}_i$. Let $\gamma = (\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_N)$, and the region of all possible locations is given by $\Gamma^N = \Gamma \times \cdots \times \Gamma$.

The transfer function from the source to the i-th microphone is given by

$$A_i(\boldsymbol{s}, f, \gamma) = \frac{1}{\|\boldsymbol{s} - \gamma_i\|} e^{-j2\pi f \|\boldsymbol{s} - \gamma_i\|/c}. \tag{1}$$

The array response is therefore given by

$$\boldsymbol{a}(\boldsymbol{s}, f, \gamma) = (A_1(\boldsymbol{s}, f, \gamma), \ldots, A_N(\boldsymbol{s}, f, \gamma))^\intercal, \tag{2}$$

and the beam response is

$$G(\boldsymbol{s}, f, \gamma) = \boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{s}, f, \gamma)$$

with

$$\boldsymbol{d}(\boldsymbol{s},f,\gamma)=\boldsymbol{a}(\boldsymbol{s},f,\gamma)\otimes\boldsymbol{d}_0(f),$$

where $\otimes$ is the Kronecker product, and $\boldsymbol{d}_0(f)=\left(1,e^{-j2\pi f/f_s},\ldots,e^{-j2\pi f(L-1)/f_s}\right)^{\mathsf{T}}$ is the filter response vector. For a given array configuration, this beamforming design problem can be formulated as a minimax problem:

$$\min_{\boldsymbol{w}\in\mathbb{R}^{NL},\gamma\in\Gamma^N}\ \max_{(\boldsymbol{s},f)\in\Omega}|\boldsymbol{w}^{\mathsf{T}}\boldsymbol{d}(\boldsymbol{s},f,\gamma)-G_d(\boldsymbol{s},f,\gamma)|, \tag{3}$$

where $G_d(\boldsymbol{s},f,\gamma)$ is the specified desired response of the broadband beamformer.

Following [10], we expand the complex functions as

$$\boldsymbol{d}(\boldsymbol{s},f,\gamma)=\boldsymbol{d}_1(\boldsymbol{s},f,\gamma)+j\boldsymbol{d}_2(\boldsymbol{s},f,\gamma),$$

$$G_d(\boldsymbol{s},f,\gamma)=G_{d_1}(\boldsymbol{s},f,\gamma)+jG_{d_2}(\boldsymbol{s},f,\gamma),$$

and denote

$$u(\boldsymbol{s},f,\gamma)=(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{d}_1(\boldsymbol{s},f,\gamma)-G_{d_1}(\boldsymbol{s},f,\gamma)),$$
$$v(\boldsymbol{s},f,\gamma)=(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{d}_2(\boldsymbol{s},f,\gamma)-G_{d_2}(\boldsymbol{s},f,\gamma)).$$

By introducing a slack variable δ

$$\delta=\max_{(\boldsymbol{s},f)\in\Omega}|u(\boldsymbol{s},f,\gamma)+jv(\boldsymbol{s},f,\gamma)|,$$

the above minimax problem can be further written as

$$\begin{cases}\displaystyle\min_{\boldsymbol{w}\in\mathbb{R}^{NL},\gamma\in\Gamma^N,\delta} & \delta\\ \qquad\text{s.t.} & |u(\boldsymbol{s},f,\gamma)+jv(\boldsymbol{s},f,\gamma)|\le\delta\quad\forall(\boldsymbol{s},f)\in\Omega.\end{cases} \tag{4}$$

We can actually control the real part and the imaginary part separately. Using the l_1 norm as a linear relaxation, and introducing two new variables as

$$z_1=\max_{(\boldsymbol{s},f)\in\Omega}|u(\boldsymbol{s},f,\gamma)|,\quad z_2=\max_{(\boldsymbol{s},f)\in\Omega}|v(\boldsymbol{s},f,\gamma)|,$$

we convert the above problem into the following problem:

$$\begin{cases}\displaystyle\min_{\boldsymbol{w}\in\mathbb{R}^{NL},\gamma\in\Gamma^N,z_1,z_2} & z_1+z_2\\ \qquad\text{s.t.} & |u(\boldsymbol{s},f,\gamma)|\le z_1\quad\forall(\boldsymbol{s},f)\in\Omega\\ & |v(\boldsymbol{s},f,\gamma)|\le z_2\quad\forall(\boldsymbol{s},f)\in\Omega\end{cases} \tag{5}$$

which is equivalent to

$$\begin{cases}\displaystyle\min_{\boldsymbol{w}\in\mathbb{R}^{NL},\gamma\in\Gamma^N,z_1,z_2} & z_1+z_2 & \\ \qquad\text{s.t.} & \boldsymbol{w}^{\mathsf{T}}\boldsymbol{d}_1(\boldsymbol{s},f,\gamma)-G_{d_1}(\boldsymbol{s},f,\gamma)\le z_1 & \forall(\boldsymbol{s},f)\in\Omega\\ & -\boldsymbol{w}^{\mathsf{T}}\boldsymbol{d}_1(\boldsymbol{s},f,\gamma)+G_{d_1}(\boldsymbol{s},f,\gamma)\le z_1 & \forall(\boldsymbol{s},f)\in\Omega\\ & \boldsymbol{w}^{\mathsf{T}}\boldsymbol{d}_2(\boldsymbol{s},f,\gamma)-G_{d_2}(\boldsymbol{s},f,\gamma)\le z_2 & \forall(\boldsymbol{s},f)\in\Omega\\ & -\boldsymbol{w}^{\mathsf{T}}\boldsymbol{d}_2(\boldsymbol{s},f,\gamma)+G_{d_2}(\boldsymbol{s},f,\gamma)\le z_2 & \forall(\boldsymbol{s},f)\in\Omega.\end{cases} \tag{6}$$

To summarize, the design problem can be formulated as the following semi-infinite programming problem:

$$\begin{cases} \min\limits_{\boldsymbol{z}\in\mathbb{R}^{NL+2},\gamma\in\Gamma^N} & \boldsymbol{b}^T\boldsymbol{z} \\ \quad\text{s.t.} & \boldsymbol{H}(\boldsymbol{s},f,\gamma)\boldsymbol{z}-\boldsymbol{G}(\boldsymbol{s},f,\gamma)\le 0 \qquad \forall(\boldsymbol{s},f)\in\Omega, \end{cases} \tag{7}$$

where $\boldsymbol{z}=(\boldsymbol{w},z_1,z_2)^T$, $\boldsymbol{b}=(\boldsymbol{0},1,1)^T$,

$$\boldsymbol{H}(\boldsymbol{r},f,\gamma)=\begin{pmatrix} \boldsymbol{d}_1(\boldsymbol{s},f,\gamma)^T & -1 & 0 \\ -\boldsymbol{d}_1(\boldsymbol{s},f,\gamma)^T & -1 & 0 \\ \boldsymbol{d}_2(\boldsymbol{s},f,\gamma)^T & 0 & -1 \\ -\boldsymbol{d}_2(\boldsymbol{s},f,\gamma)^T & 0 & -1 \end{pmatrix}, \quad \boldsymbol{G}(\boldsymbol{r},f,\gamma)=\begin{pmatrix} G_{d_1}(\boldsymbol{s},f,\gamma) \\ -G_{d_1}(\boldsymbol{s},f,\gamma) \\ G_{d_2}(\boldsymbol{s},f,\gamma) \\ -G_{d_2}(\boldsymbol{s},f,\gamma) \end{pmatrix}.$$

Define

$$g(\boldsymbol{z},(\boldsymbol{s},f,\gamma))=\boldsymbol{H}(\boldsymbol{s},f,\gamma)\boldsymbol{z}-\boldsymbol{G}(\boldsymbol{s},f,\gamma).$$

Then the above problem (2) can be represented by

$$\begin{cases} \min\limits_{\boldsymbol{z}\in\mathbb{R}^{NL+2},\gamma\in\Gamma^N} & \boldsymbol{b}^T\boldsymbol{z} \\ \quad\text{s.t.} & g(\boldsymbol{z},(\boldsymbol{s},f,\gamma))\le 0 \qquad \forall(\boldsymbol{s},f)\in\Omega. \end{cases} \tag{8}$$

3. Microphone Array Pruning Formulation

For microphone array configuration design, we formulate the problem as

$$\min_{\gamma\in\Gamma^N}\boldsymbol{b}^T\boldsymbol{z}(\gamma), \tag{9}$$

where $\boldsymbol{z}(\gamma)$ is the optimal value of the beamformer filter design problem

$$\begin{cases} \min\limits_{\boldsymbol{z}\in\mathbb{R}^{NL+2}} & \boldsymbol{b}^T\boldsymbol{z}(\gamma) \\ \quad\text{s.t.} & g(\boldsymbol{z},(\boldsymbol{s},f,\gamma))\le 0 \qquad \forall(\boldsymbol{s},f)\in\Omega. \end{cases}$$

However, if the number of microphones is very large, it will increase the implementation complexity significantly, but not necessarily improve the performance. Thus, we should find a way to prune unnecessary microphones in order to reduce the size of the microphone array without substantially affecting the performance. Let the initial number of microphones be N. A penalty term can be added to (9) in order to drive redundant microphones together for subsequent pruning. The final sparse configuration design problem can be formulated as

$$\min_{\gamma\in\Gamma^N}\boldsymbol{b}^T\boldsymbol{z}(\gamma)+\delta\sum_{i,j}\|\gamma_i-\gamma_j\|_2^2,$$

where $\boldsymbol{z}(\gamma)$ is the optimal value of the beamformer filter design problem

$$\begin{cases} \min\limits_{\boldsymbol{z}\in\mathbb{R}^{NL+2}} & \boldsymbol{b}^T\boldsymbol{z}(\gamma)+\epsilon\|\boldsymbol{z}\|_2^2 \\ \quad\text{s.t.} & g(\boldsymbol{z},(\boldsymbol{s},f,\gamma))\le 0 \qquad \forall(\boldsymbol{s},f)\in\Omega. \end{cases}$$

In the above optimization problem, the added penalty function $\delta\sum_{i,j}\|\gamma_i-\gamma_j\|_2^2$ in the objective function plays the role of driving redundant microphones together. Clearly, when we increase δ, the redundant microphones will move closer and faster. After solving the above optimization problem, for every pair of microphones, we can calculate the Euclidean distances $d_{ij}=\|\gamma_i-\gamma_j\|_2$ between every pair of microphones γ_i and γ_j. Denote the smallest distance by d_{ab}. Using d_{ab}, we prune the microphone γ_a, so the number of microphones is reduced to $N-1$. We in turn solve the above optimization problem using $N-1$ microphones with the designed configuration. We prune one microphone in each iteration until the stopband ripple has increased unacceptably. The final algorithm can be summarized as follows:

Step 1: Set $k = 0$ and solve the following two-stage optimization problem:

$$\min_{\gamma \in \Gamma^{N-k}} \boldsymbol{b}^T z(\gamma) + \delta \sum_{i,j} \|\gamma_i - \gamma_j\|_2^2,$$

where $z(\gamma)$ is the optimal value of the beamformer filter design problem with a small perturbation

$$\begin{cases} \min\limits_{\boldsymbol{z} \in \mathbb{R}^{(N-k)L+2}} & \boldsymbol{b}^T \boldsymbol{z}(\gamma) + \epsilon \|\boldsymbol{z}\|_2^2 \\ \quad\text{s.t.} & g(\boldsymbol{z}, (\boldsymbol{r}, f, \gamma)) \leq 0 \qquad \forall (\boldsymbol{r}, f) \in \Omega. \end{cases}$$

If stopband ripple increases significantly, stop.

Step 2: The microphones a and b are chosen to correspond to the smallest Euclidean distance

$$d_{ab} = \min \|\gamma_i - \gamma_j\|_2, i, j = 1, \cdots, N-k.$$

Step 3: Prune the microphone γ_a, set $k = k+1$, and return to Step 1.

In solving the sub-problem (10), discretization is employed to convert the solution space into a finite set of points. Note that we also add $\epsilon\|z\|_2^2$ in the objective function of the beamformer filter design subproblem so that the objective function becomes strictly convex, which enables the use of modified exchange method [16,24] with a rather low complexity, even for the multi-dimensional design problem. When ϵ is sufficiently small, the approximate solution of the perturbed problem will be very close to the true solution.

Assume that there exists a finite subset Ω_0 such that $\boldsymbol{b}^T \boldsymbol{z}$ is level bounded on the feasible set, that is, for every $a \in \mathbb{R}$, the set

$$\mathcal{L}_a^0 := \{\boldsymbol{z} \in \mathbb{R}^n; \boldsymbol{b}^T \boldsymbol{z} \leq a \text{ and } g(\boldsymbol{z}, (\boldsymbol{s}, f, \gamma) \leq 0 \text{ for all } (\boldsymbol{s}, f) \in \Omega_0\}$$

is bounded when it is nonempty. For a given finite set $\mathcal{R} = \{(\boldsymbol{s}_j, f_j), j = 1, \cdots, m\} \subset \Omega$, we consider the finite problem denoted by $(BP_\epsilon(\mathcal{R}^{(i)}))$:

$$\begin{cases} \min\limits_{\boldsymbol{z} \in \mathbb{R}^{(N-k)L+2}} & \boldsymbol{b}^T \boldsymbol{z}(\gamma) + \epsilon \|\boldsymbol{z}(\gamma)\|_2^2 \\ \quad\text{s.t.} & g(\boldsymbol{z}, (\boldsymbol{s}_j, f_j, \gamma)) \leq 0 \qquad \text{for any } j = 1, \cdots, m \end{cases} \tag{10}$$

We use a modified exchange algorithm [24] for this subproblem. The algorithm can be summarized as follows:

Step 0: Choose a finite reference set $\mathcal{R}^{(0)} = \{(\boldsymbol{s}_j^{(0)}, f_j^{(0)}), j = 1, \cdots, m_0\} \subset \Omega$ such that $\Omega_0 \subset \mathcal{R}^0$. Let $\boldsymbol{z}^{(0)}(\gamma)$ be an optimal solution to $(BP_\epsilon(\mathcal{R}^{(0)}))$ and let $\{\lambda(\boldsymbol{s}_j^{(0)}, f_j^{(0)}), j = 1, \cdots, m_0\} \in \mathbb{R}^m$ be the set of associated multipliers. Set $k = 0$.

Step 1: Find a set $\{(\boldsymbol{s}_{new}^{(k)}, f_{new}^{(k)}), new = 1, \cdots, n\} \subset \Omega$ such that

$$g(\boldsymbol{z}^{(k)}, (\boldsymbol{s}_{new}^{(k)}, f_{new}^{(k)}, \gamma)) > \eta.$$

If such a point does not exist, then stop. Otherwise, put $\bar{\mathcal{R}}^{(k+1)} = \mathcal{R}^{(k)} \cup \{(\boldsymbol{s}_{new}^{(k)}, f_{new}^{(k)})\}$.

Step 2: Let $\boldsymbol{z}^{(k+1)}(\gamma)$ be an optimal solution to $(BP_\epsilon(\bar{\mathcal{R}}^{(k+1)}))$ and let $\{\lambda^{(k+1)}(\boldsymbol{s}, f), (\boldsymbol{s}, f) \in \bar{\mathcal{R}}^{(k+1)}\}$ be the set of associated multipliers.

Step 3: Let

$$\mathcal{R}^{(k+1)} := \left\{ (\boldsymbol{s}, f) \in \bar{\mathcal{R}}^{(k+1)}; (\boldsymbol{s}, f) \in \Omega_0 \text{ or } \lambda^{(k+1)}(\boldsymbol{s}, f) > 0 \right\}.$$

Set $k = k + 1$, and return to Step 1.

One important point in Step 1 is that we choose multiple points satisfying the adding rule instead of just choosing one. This will reduce the number of iterations and shorten the computational time. Furthermore, convergence of the algorithm can be found in [24].

Using the final pruned array, we can then perform the sparse filter design problem. The overall sparse design problem can be tackled in two stages. In the first stage, the microphone array is first pruned. Then, we can initiate stage two of the method to design the sparse beamforming filters using the algorithm presented in [24].

4. Numerical Examples

In this section, we provide examples to demonstrate the performance of the algorithm which was implemented in MATLAB. We chose the desired response function as

$$G_d(\boldsymbol{r}, f) = \begin{cases} e^{-j2\pi f \left(\frac{||\boldsymbol{r}-\boldsymbol{r}_c||}{c} + \frac{L-1}{2} T \right)}, & \text{if } (\boldsymbol{r}, f) \text{ is in passband region,} \\ 0, & \text{if } (\boldsymbol{r}, f) \text{ is in stopband region,} \end{cases}$$

where $\boldsymbol{r}_c$ is the reference central microphone location. In this example, we considered the microphone array pruning problem combined with the design of sparse filters. Assume first that there are nine microphones located at the coordinates {(−0.08, 1.2), (−0.04,1.2), (0,1.2), (0.04,1.2), (0.08,1.2), (0,1.4), (0,1.3), (0,1.1), (0,1)} (Figure 1), with a 26-tap FIR filter behind each element. Note that in the Figures, the speaker is represented by a red diamond while microphones are represented by blue circles. The passband region is defined as

$$\{(x, f) : -0.4\,\text{m} \le x \le 0.4\,\text{m},\ 0.5\,\text{kHz} \le f \le 1.5\,\text{kHz}\},$$

while the stopband region is simplified as the union of

$$\{(x, f) : 1.8\,\text{m} \le |x| \le 3\,\text{m},\ 0.5\,\text{kHz} \le f \le 1.5\,\text{kHz}\},$$

$$\{(x, f) : -3\,\text{m} \le |x| \le 3\,\text{m},\ 2\,\text{kHz} \le f \le 4,\,\text{kHz}\}.$$

In the following, we used the MATLAB functions *fminsearch* and quadprog to solve the microphone pruning problem. For the performance, since stopband ripple is an important indicator of noise reduction, we employed it to demonstrate the effect of sparsity in the designs. After solving the problem, the number of microphones could be decreased to 4 (Figure 2). The amplitude of the actual response $G(\boldsymbol{r}, f)$ is shown in Figure 3, in which the passband is represented by red color. From Figure 4, we can see that the designed beamformer size could be reduced significantly. Using the final pruned array, we could then perform the sparse filter design problem. The amplitude of the actual response G(r,f) is shown in Figure 5. It is observed in Figure 6 that the performance was not affected greatly even when the sparseness increased to about 38% of zero elements for filter coefficient.

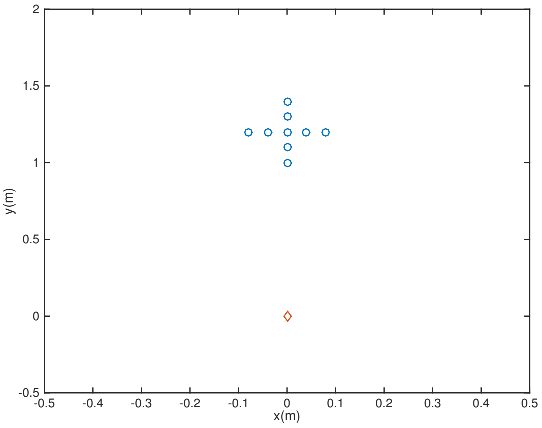

Figure 1. Initial array configuration (Ex1).

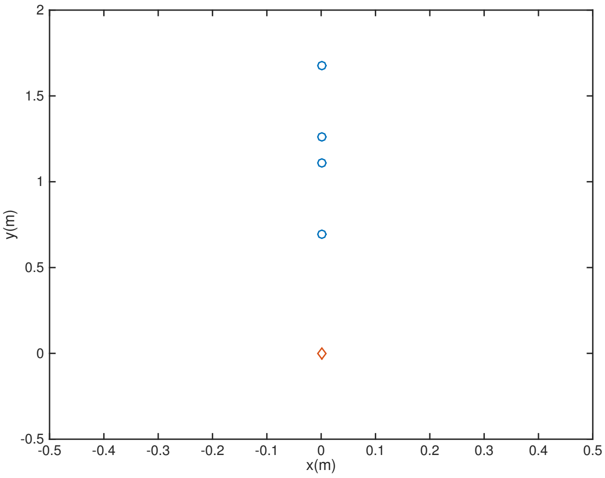

Figure 2. Final array configuration (Ex1).

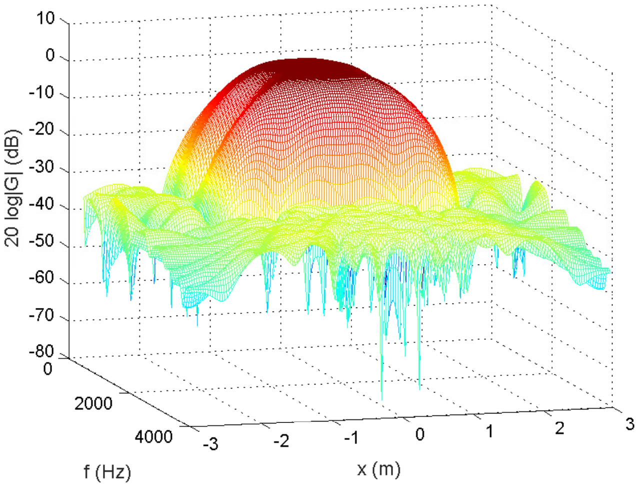

Figure 3. Amplitude of $G(\mathbf{r}, f)$, 55.56% of zeroes for microphone arrays (Ex1).

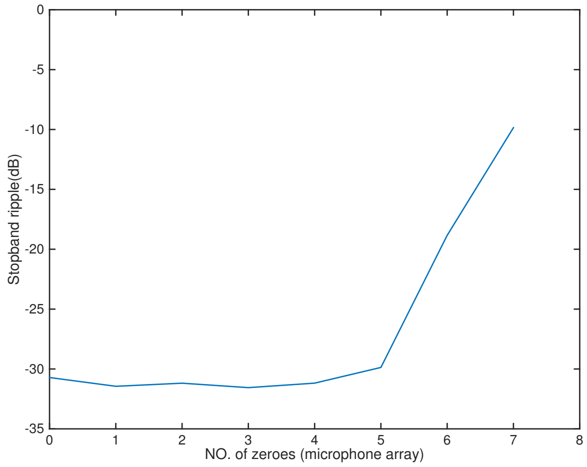

Figure 4. Stopband ripple for microphone arrays (Ex1).

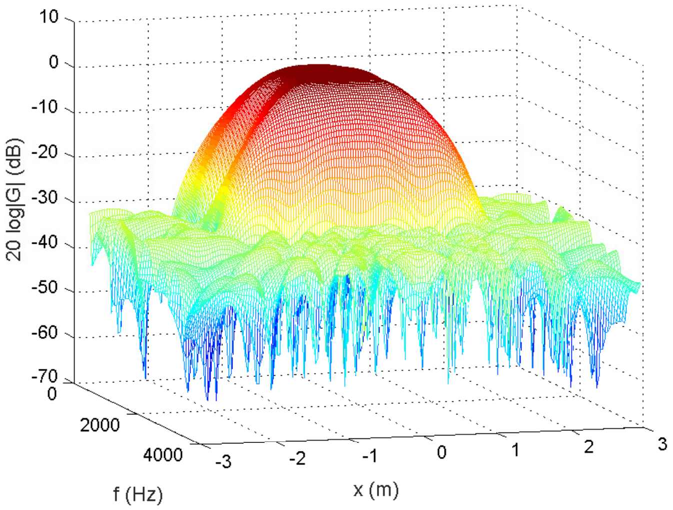

Figure 5. Amplitude of $G(\boldsymbol{r}, f)$, 38.46% of zeroes for filters (Ex1).

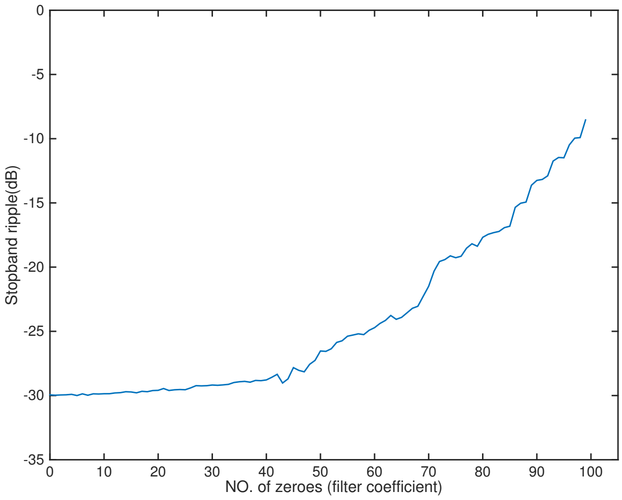

Figure 6. Stopband ripple for filters (Ex1).

In the next example, we considered the microphone array pruning problem using a randomly generated microphone array (Figure 7). The passband region was defined as

$$\{(x, f) : -0.4\,\text{m} \leq x \leq 0.4\,\text{m},\ 0.5\,\text{kHz} \leq f \leq 1.5\,\text{kHz}\},$$

while the stopband region was the union of several parts, as

$$\{(x, f) : -0.4\,\text{m} \leq x \leq 0.4\,\text{m},\ 2\,\text{kHz} \leq f \leq 4\,\text{kHz}\},$$

$$\{(x, f) : 1.8\,\text{m} \leq |x| \leq 3\,\text{m},\ 0.5\,\text{kHz} \leq f \leq 1.5\,\text{kHz}\},$$

$$\{(x, f) : 1.8\,\text{m} \leq |x| \leq 3\,\text{m},\ 2\,\text{kHz} \leq f \leq 4\,\text{kHz}\}.$$

A 12-tap FIR filter behind each element was used. The number of microphones could be decreased to 5 (Figure 8). The amplitude of the actual response $G(\boldsymbol{r}, f)$ is shown in Figure 9. From Figure 10, we can see that the designed beamformer array could be simplified significantly without greatly affecting the performance. After pruning the microphone array, we could then carry out the sparse filter design problem. The amplitude of the actual response $G(\boldsymbol{r}, f)$ is shown in Figure 11. It is observed in Figure 12 that the performance was not affected greatly, even when the sparseness increased to about 41% of zero elements for the filter coefficient.

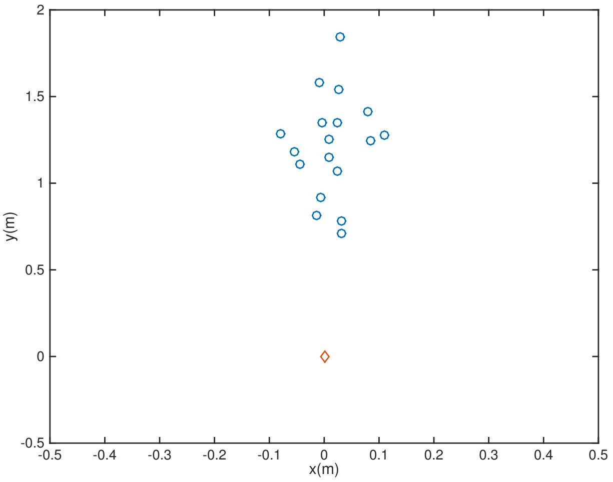

Figure 7. Initial array configuration (Ex2).

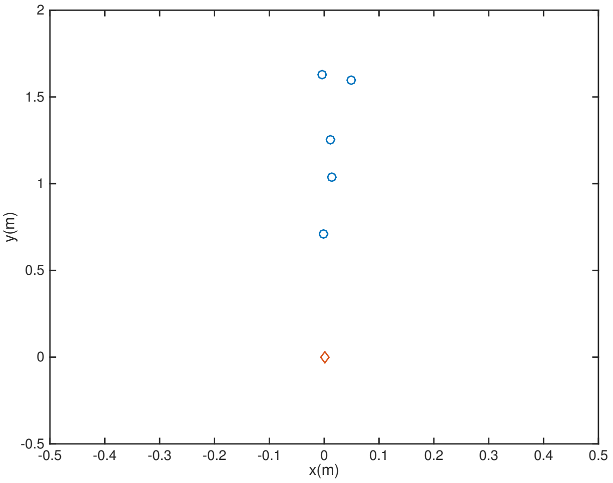

Figure 8. Final array configuration (Ex2).

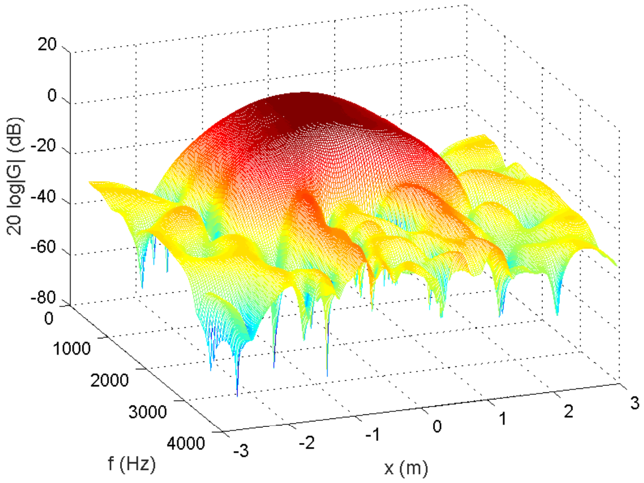

Figure 9. Amplitude of $G(\boldsymbol{r}, f)$, 75% of zeroes for microphone arrays (Ex2).

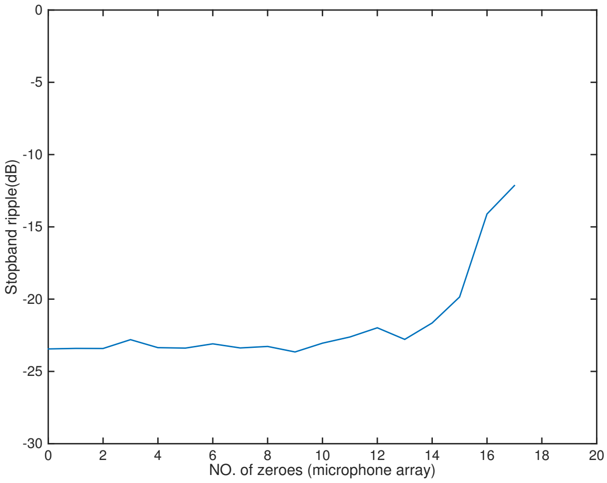

Figure 10. Stopband ripple for microphone arrays (Ex2).

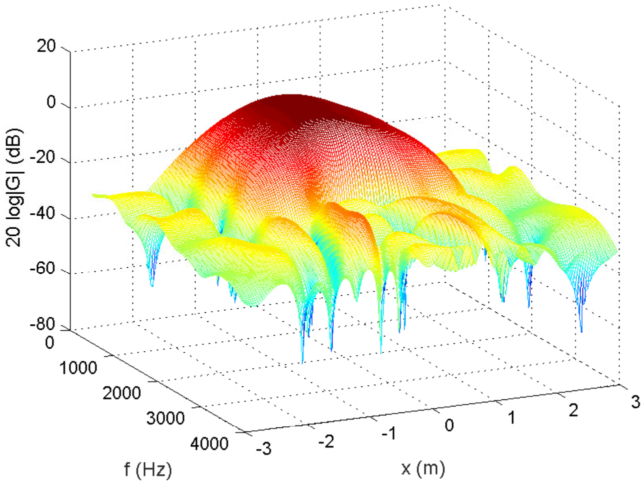

Figure 11. Amplitude of $G(\boldsymbol{r}, f)$, 41.67% of zeroes for filters (Ex2).

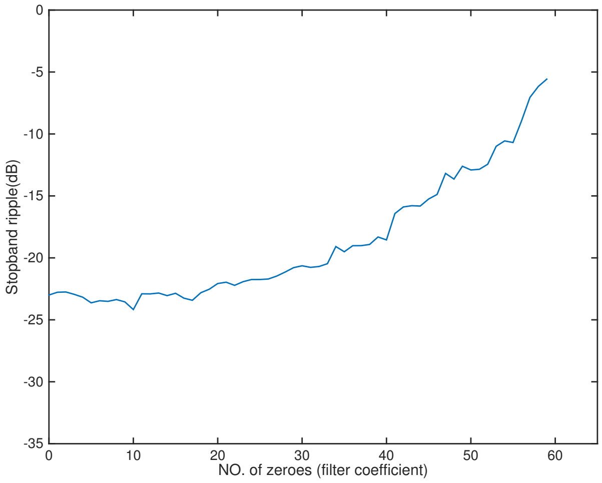

Figure 12. Stopband ripple for filters (Ex2).

5. Conclusions

In this paper, we studied the sparse beamformer design problem for both array and filters. A novel array pruning problem was formulated and an optimization algorithm was proposed to prune the microphones sequentially. In finding sparse beamformer filters, a perturbation exchange algorithm was proposed to solve the sparse design problem with the successful thinning technique. We studied the performance of the optimized designs with several examples. We demonstrated that there was a region of approximately equal performance for both array pruning and sparse filter design. As a future extension, it would be of interest to study the convergence properties of the overall optimization algorithm.

Author Contributions: Writing—original draft, M.G. and K.a F.C.Y.; Writing—review & editing, K.F.C.Y. and S.N.

Funding: This paper is supported by the National Natural Science Foundation of China (NSFC) Grant 11801184 and 11771157, GRF Grant PolyU. 152200/14E and PolyU grant 4-ZZGS and G-YBVQ.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brandstein, M.; Ward, D. *Microphone Arrays: Signal Processing Techniques and Applications*; Springer: Berlin, Germany, 2001.
2. Kellermann, W. Beamforming for speech and audio signals. In *Handbook of Signal Processing in Acoustics*; Havelock, D., Kuwano, S., Vorlander, M., Eds.; Springer: New York, NY, USA, 2008.
3. Ryan, J.; Goubran, R. Array optimization applied in the near field of a microphone array. *IEEE Trans. Speech Audio Process.* **2000**, *8*, 173–178. [CrossRef]
4. Kennedy, R.; Ward, D.; Abhayapala, T. Nearfield beamforming using radial reciprocity. *IEEE Trans. Signal Process.* **1999**, *47*, 33–40. [CrossRef]
5. Lim, Y.C.; Lian, Y. The optimum design of one and two-dimensional FIR filters using the frequency response masking technique. *IEEE Trans. Circuits Syst. II* **1993**, *40*, 88–95. [CrossRef]
6. Lau, B.K.; Leung, Y.H.; Teo, K.L.; Sreeram, V. Minimax filters for microphone arrays. *IEEE Trans. Circuits Syst. II* **1999**, *46*, 1522–1525. [CrossRef]
7. Chan, S.C.; Chen, H.H. Uniform concentric circular arrays with frequency-invariant characteristics-theory, design, adaptive beamforming and DOA estimation. *IEEE Trans. Signal Process.* **2007**, *55*, 165–177. [CrossRef]
8. Nordholm, S.; Rehbock, V.; Teo, K.L.; Nordebo, S. Chebyshev approximation for the design of broadband beamformers in the near field. *IEEE Trans. Circuits Syst. II* **1998**, *45*, 141–143. [CrossRef]
9. Yiu, K.F.C.; Grbic, N.; Teo, K.L.; Nordholm, S. A new design method for broadband microphone arrays for speech input in automobiles. *IEEE Signal Proc. Lett.* **2002**, *9*, 222–224.

10. Yiu, K.F.C.; Yang, X.Q.; Nordholm, S.; Teo, K.L. Near-field broadband beamformer design via multidimensional semi-infinite linear programming techniques. *IEEE Trans. Speech Audio Process.* **2003**, *11*, 725–732.
11. Feng, Z.G.; Yiu, K.F.C.; Nordholm, S. A two-stage method for the design of near-field broadband beamformer. *IEEE Trans. Signal Process.* **2011**, *59*, 3647–3656. [CrossRef]
12. Feng, Z.G.; Yiu, K.F.C. The design of multi-dimensional acoustic beamformers via window functions. *Digit. Signal Process.* **2014**, *29*, 107–116. [CrossRef]
13. Feng, Z.G.; Yiu, K.F.C.; Nordholm, S. Performance limit of broadband beamformer designs in space and frequency. *J. Optim. Theory Appl.* **2015**, *164*, 316–341. [CrossRef]
14. Baran, T.; Wei, D.; Oppenheim, A.V. Linear programming algorithms for sparse filter design. *IEEE Trans. Signal Process.* **2010**, *58*, 1605–1617. [CrossRef]
15. Jiang, A.; Kwan, H.K. Recent advances in sparse FIR filter design using l_0 and l_1 optimization techniques. In *Trends in Digital Signal Processing*; Lim, Y., Kwan, H.K., Siu, W.C., Eds.; Pan Stanford Publishing: Singapore, 2016.
16. Zhang, L.P.; Wu, S.Y.; Lopez, M.A. Optimizing floating point units in hybrid FPGAs. *SIAM J. Optim.* **2010**, *20*, 2959–2977. [CrossRef]
17. Yiu, K.F.C.; Gao, M.J.; Feng, Z.G. Design of near-field broadband beamformer using semi-definite programming. *Int. J. Innov. Comput. Inf. Control* **2012**, *8*, 3755–3768.
18. Oliveri, G.; Donelli, M.; Massa, A. Linear array thinning exploiting almost difference sets. *IEEE Trans. Antennas Propag.* **2009**, *57*, 3800–3812. [CrossRef]
19. Fernandez-Delgado, M.; Rodriguez-Gonzalez, J.; Iglesias, R.; Barro, S.; Ares-Pena, A. Fast array thinning using global optimization methods. *J. Electromagn. Waves Appl.* **2010**, *24*, 2259–2271. [CrossRef]
20. Feng, Z.G.; Yiu, K.F.C.; Nordholm, S. Placement design of microphone arrays in near-field broadband beamformers. *IEEE Trans. Signal Process.* **2012**, *60*, 1195–1204. [CrossRef]
21. Li, Z.B.; Yiu, K.F.C.; Feng, Z.G. A hybrid descent method with genetic algorithm for microphone array placement design. *Appl. Soft Comput.* **2013**, *13*, 1486–1490. [CrossRef]
22. Honig, M.; Madhow, U.; Verdu, S. Blind adaptive multiuser detection. *IEEE Trans. Inf. Theory* **1995**, *41*, 944–960. [CrossRef]
23. Darsena, D.; Verde, F. Minimum-mean-output-energy blind adaptive channel shortening for multicarrier simo transceivers. *IEEE Trans. Signal Process.* **2007**, *55*, 5755–5771. [CrossRef]
24. Gao, M.J.; Yiu, K.F.C.; Wu, S.Y. A perturbation exchange algorithm for convex semi-infinite programming with applications in sparse beamformer. *Pac. J. Optim.* **2018**, *14*, 15–30.