*International Journal of Innovative Computing, Information and Control*
Volume **8**, Number **5(B)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3755–3768

# DESIGN OF DISTRIBUTED BEAMFORMING SYSTEM USING SEMI-DEFINITE PROGRAMMING

Ka Fai Cedric Yiu[1], Ming Jie Gao[1] and Zhi Guo Feng[1,2]

[1]Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong, P. R. China
macyiu@polyu.edu.hk

[2]College of Mathematics
Chongqing Normal University
No. 12, Tianchen Road, Shapingba, Chongqing 400047, P. R. China

Received October 2010; revised March 2011

Abstract. *In this paper, the design of distributed broadband beamforming system is studied. In the configuration, we assume that each microphone is equipped with wireless communications capability. Once their mutual distance information is collected, localization techniques can be used to estimate the microphone locations. A broadband beamformer can then be designed such that the error between the actual response and the desired response is minimized. However, due to variations in the estimated microphone locations, robust design with uncertainties must be considered. This problem is formulated as a minimax optimization problem, which is then transformed into a semi-definite programming problem so that interior point algorithms can be applied. We illustrate the proposed method by several designs and show that the algorithm is robust and efficient.*
**Keywords:** Distributed beamforming, Sensor network, Semi-definite programming

1. **Introduction.** Current advancement of wireless communications has made its deployment in wider perspective. This facilitates the development of distributed systems, such as a microphone network, which overcomes some of the technical problems from a wired system by providing greater freedom of movements for the speaker and avoidance of cabling problems common with wired microphones caused by constant moving and stressing the cables. There are numerous applications that can be built on a microphone network. Speech is the preferred natural interface for controlling equipment in households or factories. However, signal degradation poses a serious problem in many environments, which affects the accuracy of the speech recognition and voice control system. As a result, beamforming techniques are required to enhance the received signals.

Broadband beamformers [1, 2, 3, 4] have been studied extensively due to their wide applications in many areas such as radar, sonar, wireless communications, biomedicine, speech and acoustics. When microphone arrays are deployed, many beamforming algorithms exist (for example, [5, 6, 7]) to reduce the level of localized and ambient noise signals from the desired direction via spatial filtering, which plays an important role in noise reduction and speech enhancement. For many applications, such as video conference and mobile telephony, the speaker does not stand very far from the array. There are various algorithms dedicated for the design of this kind of beamformer in the literature. In [8], the near-field-far-field reciprocity relationship is derived and applied to designing near-field beamformers via far-field design techniques. An interesting approach is presented in [9]. It makes use of a signal propagation vector representing an ideal point source of acoustic radiation. When the desired frequency response is known, multidimensional filter design techniques can be applied. In [10], the minimax problem is formulated as a quadratic programming problem and the SQP method is applied. A penalty function method is developed in [11] to formulate the problem as an unconstrained nonlinear optimization problem. This method is modified in [12] by replacing the penalty function with a root-catching method. In [13], the $l_1$ norm measure and the real rotation theorem are applied to formulating the problem as a semi-infinite linear programming problem.

For many applications, the design problem can be formulated as a minimax optimization problem. Similar to many filter design problems [14, 15, 16], large-scale linear programming techniques (for example, [17, 18]) are often used. When the problem size increases as a result of an increase in the number of filters as well as the filter lengths, or a refinement in the discretization of the frequency-space domain, the number of constraints will be very large and these problems will be very expensive to solve if the methods above are applied. Hence, an efficient algorithm is necessary. Semi-definite programming (SDP) is a generalization of linear programming (LP) where the decision variables are arranged in a symmetric matrix instead of a vector, and the non-negative orthant is replaced by the cone of positive semi-definite matrices [19, 20, 21]. Since interior point algorithms can be employed, semi-definite programming has polynomial time computational complexity and can be solved efficiently. It has also been successfully applied to many signal processing problems, such as frequency response masking filter design [22], antenna design [23], filter bank design [24] and sensor network [25], achieving very good performance. Hence, in this paper, we will apply the SDP method to solving the broadband beamformer design problem.

One of the assumptions in applying the aforementioned design techniques is that the locations of the microphones are required to be measured exactly. In practice, the microphones could be scattered around and could even be moving around occasionally. If very precise measurements are needed every time, this will make the design process very tedious and repetitive. If wireless microphones are deployed instead, we just need to make use of the accurate positions of a few anchor nodes in the network together with the pairwise distance measurements between any two nodes to estimate the locations of the wireless microphones. Since the distance measurements always contain noise and the effect of the measurement uncertainty usually depends on the geometrical relationship between sensors which is not known *a priori*, optimization techniques are often deployed to find the best estimates. Here, we also adopt the SDP method for solving the problem [26, 27]. The basic idea behind the technique is to convert the nonconvex quadratic distance constraints into linear constraints by introducing a relaxation to remove the quadratic term in the formulation. The performance of this technique is highly satisfactory compared with other techniques [28, 29]. Very few anchor nodes are required to accurately estimate the position of all the unknown nodes in a network. Also the estimation errors are minimal even when the anchor nodes are placed arbitrarily within the network.

Owing to perturbations in the estimated sensor locations, a robust formulation is required to allow for a certain amount of errors. In fact, the designed beamformers turn out to be very sensitive to errors in the microphone locations. In this paper, the sensor network technology will be employed and incorporated into broadband beamforming design. In particular, an appropriate robust formulation is proposed to give more flexibility in the designs. We will demonstrate by examples that the proposed robust approach is essential to regain the accuracy in the designs if microphone locations are indeed erroneous.

The rest of the paper is organized as follows. In Section 2, we formulate the wireless beamformer design problem and the localization problem of microphones. Then, we introduce its corresponding robust problem. In Section 3, we transform all the problems into equivalent semi-definite programming problems. For illustration, several examples are solved in Section 4. Conclusions are presented in Section 5.

## 2. **Formulation.**

The structure of a wireless near-field broadband beamformer can be found in Figure 1, where the positions of microphones can be arbitrary and the sound signal is received by the microphone array and processed by the FIR filters behind.

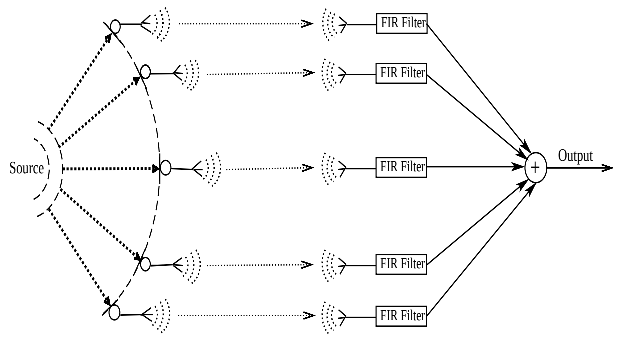


FIGURE 1. The structure of a wireless near-field beamformer

We assume that there are $N$ elements in the array. Using a simple spherical model, the transfer function from the source point $\boldsymbol{r}$ to the $i$-th element of the broadband beamformer is given by

$$A_i(\boldsymbol{r}, f) = \frac{1}{\|\boldsymbol{r} - \boldsymbol{r}_i\|} e^{-j2\pi f \|\boldsymbol{r} - \boldsymbol{r}_i\|/c}, \tag{2.1}$$

where $\boldsymbol{r}$ is the position vector of the source signal, $\boldsymbol{r}_i$ is the position vector of the $i$-th microphone, $f$ is the frequency, and $c$ is the sound speed. Then, the array response vector is therefore given by

$$\boldsymbol{a}(\boldsymbol{r}, f) = (A_1(\boldsymbol{r}, f), \ldots, A_N(\boldsymbol{r}, f))^\intercal. \tag{2.2}$$

Let each microphone signal be sampled at a rate of $f_s$, and suppose that each FIR filter has $L$ taps. Denote the filter response vector by

$$\boldsymbol{d}_0(f) = \left(1, e^{-j2\pi f/f_s}, \ldots, e^{-j2\pi f(L-1)/f_s}\right)^\intercal \tag{2.3}$$

and the filter coefficients by

$$\boldsymbol{w} = (\boldsymbol{w}_1^\intercal, \ldots, \boldsymbol{w}_N^\intercal)^\intercal, \tag{2.4a}$$

where

$$\boldsymbol{w}_i = (w_i(0), \ldots, w_i(L-1))^\intercal, \quad i = 1, \ldots, N, \tag{2.4b}$$

then the actual response of the broadband beamformer is given by

$$G(\boldsymbol{r}, f) = \boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) \tag{2.5}$$

with

$$\boldsymbol{d}(\boldsymbol{r}, f) = \boldsymbol{a}(\boldsymbol{r}, f) \otimes \boldsymbol{d}_0(f), \tag{2.6}$$

where $\otimes$ denotes the Kronecker product and the dimension of $\boldsymbol{w}$ is $n = N \times L$.

Let $G_d(\boldsymbol{r}, f)$ be the specified desired response of the broadband beamformer, and consider a region $\Omega = \cup_{i=1}^{m}\Omega_i$ in the space-frequency domain where each $\Omega_i$ is a convex set and $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$. Then, the minimax design problem can be formulated as

$$\min_{\boldsymbol{w}\in\mathbb{R}^n} \max_{(\boldsymbol{r},f)\in\Omega} |\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)|.$$

Obviously, if the term $|\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)|$ above is replaced by $|\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)|^2$, the optimal solution will not be changed. Hence, we can formulate the filter design problem as

**Problem 1.** *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters to minimize the following cost function*

$$\max_{(\boldsymbol{r},f)\in\Omega} |\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)|^2. \tag{2.7}$$

For wireless microphones, since they can be placed anywhere, it is more practical if we can estimate the locations spontaneously. In fact, the locations of the microphones can be estimated by a method whose principle is the same as the localization problem in sensor network. That is, to estimate the locations of the microphones, we need to have some points whose locations are known. These points are called anchors and can be denoted by $\boldsymbol{a} = \{\boldsymbol{a}_k :\in \mathbb{R}^h, k \in M^{(1)}\}$, where $M^{(1)}$ is the index set of the anchors and $h$ is the dimension which can be 1, 2 or 3, depending on the structure of the anchors. The unknown microphones are called sensors, which can be denoted by $\boldsymbol{r} = \{\boldsymbol{r}_j :\in \mathbb{R}^h, j \in M^{(2)}\}$, where $M^{(2)}$ is the index set of the sensors. An example of sensors and anchors can be seen in Figure 2, where three diamond points are the anchors and two circle points are the sensors.

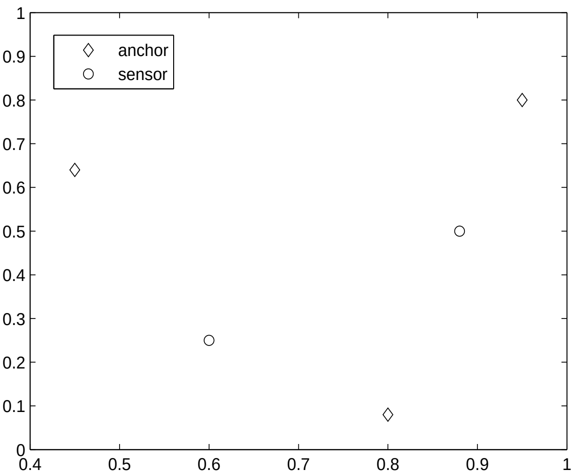


Figure 2. An example of sensor network

For every pair of points, we can estimate the distance. That is, we have Euclidean distance measures $\hat{d}_{kj}$ between anchor $\boldsymbol{a}_k$ and sensor $\boldsymbol{r}_j$ for some $k$, $j$, and $\hat{d}_{ij}$ between sensor $\boldsymbol{r}_i$ and sensor $\boldsymbol{r}_j$ for some $i < j$. Denoting $N_a = \{(k, j) : k \in M^{(1)}, j \in M^{(2)}\}$ and $N_r = \{(i, j) : i < j, i \in M^{(1)}, j \in M^{(1)}\}$, we have

$$\begin{aligned} \|\boldsymbol{a}_k - \boldsymbol{r}_j\|^2 &= \hat{d}_{kj}^2 \quad \forall (k, j) \in N_a, \\ \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 &= \hat{d}_{ij}^2 \quad \forall (i, j) \in N_r. \end{aligned} \tag{3.1}$$

From the information of distances, we can then estimate the locations of the sensors. The localization problem is to find the sensor coordinates $\boldsymbol{r}$ such that (3.1) is satisfied. The localization problem is equivalent to the optimization problem below:

**Problem 2.** *Find the locations $\boldsymbol{r}$ of the sensors to minimize*

$$F(\boldsymbol{r}) = \sum_{(i,j)\in N_r} \left| \|r_i - r_j\|^2 - \hat{d}_{ij}^2 \right| + \sum_{(k,j)\in N_a} \left| \|a_k - r_j\|^2 - \hat{d}_{kj}^2 \right|. \tag{3.2}$$

Thus, we can find the locations $\boldsymbol{r}$ of the sensors by optimizing the cost function (3.2). If the optimal value $F^*$ is zero, then the solution obtained is the exact locations. However, in most cases, the optimal value $F^*$ is strictly greater than zero. To see this, we suppose that the number of $\boldsymbol{a}$ is $m$. Then, the number of $N_a$ is $mN$ and the number of $N_r$ is $N(N-1)/2$. Hence, the total number of the equalities in (3.1) is $mN + N(N-1)/2$. On the other hand, the number of decision variables is $hN$. Then, when $mN + N(N-1)/2 > hN$, that is, $m + (N-1)/2 > h$, this problem is over-determined. Since $h \le 3$, this condition is satisfied in most cases. Hence, the optimal value of Problem 2 is strictly greater than zero and the obtained locations are not exact in most cases. Since the locations $\boldsymbol{r}$ is not exact and the performance of the designed beamformer is very sensitive to the errors in the locations, a robust design is needed.

Similar to Problem 1, we consider a corresponding robust problem where the location vector contains certain uncertainties. Denote the position vector by $\tilde{\boldsymbol{r}} = p(\boldsymbol{r}, \theta)$, where $\boldsymbol{r}$ is a position vector in Problem 1 and $\theta \in [-\eta, \eta]$ is the parameter for uncertainty. Without loss of generality, we define $p(\boldsymbol{r}, 0) = \boldsymbol{r}$. We can formulate the robust filter design problem as

**Problem 3.** *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters to minimize the following cost function*

$$\max_{\theta\in[-\eta,\eta]} \max_{(\boldsymbol{r},f)\in\Omega} |\boldsymbol{w}^\intercal \boldsymbol{d}(\tilde{\boldsymbol{r}}, f) - G_d(\tilde{\boldsymbol{r}}, f)|^2. \tag{2.8}$$

Both Problem 1 and Problem 3 are nonlinear minimax optimization problems. After the discretization of the space-frequency domain $\tilde{\Omega} = [-\eta, \eta] \times \Omega$, gradient-based methods can be applied to solving for numerical solutions. However, if the discretization of $\tilde{\Omega}$ is very large, these problems become very expensive to solve. Thus, an algorithm with polynomial time computational complexity is desirable.

## 3. **Methodology.**

### 3.1. **Robust broadband beamformer design.**

The cost functions in Problem 1 and Problem 3 are quadratic. They can be rearranged as SDP problems as follows. Expanding the complex functions

$$\boldsymbol{d}(\boldsymbol{r}, f) = \boldsymbol{d}_1(\boldsymbol{r}, f) + j\boldsymbol{d}_2(\boldsymbol{r}, f), \tag{2.9}$$

$$G_d(\boldsymbol{r}, f) = G_{d_1}(\boldsymbol{r}, f) + jG_{d_2}(\boldsymbol{r}, f), \tag{2.10}$$

and denoting

$$u(\boldsymbol{r}, f) = (\boldsymbol{w}^\intercal \boldsymbol{d}_1(\boldsymbol{r}, f) - G_{d_1}(\boldsymbol{r}, f)), \tag{2.11}$$

$$v(\boldsymbol{r}, f) = (\boldsymbol{w}^\intercal \boldsymbol{d}_2(\boldsymbol{r}, f) - G_{d_2}(\boldsymbol{r}, f)), \tag{2.12}$$

by adding an additional variable $z$, Problem 1 becomes

$$\begin{aligned} &\min_{\boldsymbol{w}\in\mathbb{R}^n, z\in\mathbb{R}} \quad z \\ &\text{s.t.} \quad u(\boldsymbol{r}, f)^2 + v(\boldsymbol{r}, f)^2 \le z, \quad \forall (\boldsymbol{r}, f) \in \Omega. \end{aligned} \tag{2.13}$$

We will make use of the following theorem proven in [30]:

**Theorem 3.1.** *Let $A$ be an $n \times n$ real symmetric matrix of rank $r$. Then, the following statements are equivalent:*

1. *$A$ is positive semi-definite.*
2. *All eigenvalues of $A$ are nonnegative.*
3. *There exists an $n \times r$ matrix $S$ such that $A = SS^T$.*
4. *All principal minors of $A$ are nonnegative.*

By Theorem 3.1, the constraint in the above problem holds if and only if

$$\Phi(z, \boldsymbol{w}, \boldsymbol{r}, f) = \begin{pmatrix} z & u(\boldsymbol{r}, f) & v(\boldsymbol{r}, f) \\ u(\boldsymbol{r}, f) & 1 & 0 \\ v(\boldsymbol{r}, f) & 0 & 1 \end{pmatrix} \succeq 0, \quad \forall (r, f) \in \Omega, \tag{2.14}$$

where " $\succeq$ " denotes the positive semi-definite symbol. Denote

$$G(z, \boldsymbol{w}) = diag\{\Phi(z, \boldsymbol{w}, \boldsymbol{r}^1, f^1), \ldots, \Phi(z, \boldsymbol{w}, \boldsymbol{r}^k, f^k)\}, \tag{2.15}$$

where $\Omega_d = \{(\boldsymbol{r}^1, f^1), \ldots, (\boldsymbol{r}^k, f^k)\} \subset \Omega$ is a set of dense grid points. Then, we transformed Problem 1 into a SDP optimization problem:

**Problem 4.** *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters and $z$, such that $z$ is minimized, subject to the constraint*

$$G(z, \boldsymbol{w}) \succeq 0. \tag{2.16}$$

Similarly to Problem 1, Problem 3 can also be transformed into an SDP optimization problem. Denote

$$\tilde{G}(z, \boldsymbol{w}) = diag\{\Phi(z, \boldsymbol{w}, p(\boldsymbol{r}^1, \theta^1), f^1), \ldots, \Phi(z, \boldsymbol{w}, p(\boldsymbol{r}^k, \theta^k), f^k)\}, \tag{2.17}$$

where $\tilde{\Omega}_d = \{(\theta^1, \boldsymbol{r}^1, f^1), \ldots, (\theta^k, \boldsymbol{r}^k, f^k)\} \subset \tilde{\Omega}$ is a set of dense grid points. Then, Problem 3 is transformed into an SDP optimization problem:

**Problem 5.** *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters and $z$, such that $z$ is minimized, subject to the constraint*

$$\tilde{G}(z, \boldsymbol{w}) \succeq 0. \tag{2.18}$$

Basically, since there is a discretization of the interval $[-\eta, \eta]$, Problem 5 is more expensive to solve than Problem 4. However, if $\eta$ is small, it is not necessary to do the whole discretization of the interval $[-\eta, \eta]$ and Problem 5 can be simplified. This can be seen in the next theorem.

**Theorem 3.2.** *Suppose that $\eta$ is small. Then, for any given coefficients vector $\boldsymbol{w} \in \mathbb{R}^n$ and frequency $f$, we have*

$$\max_{\theta \in [-\eta, \eta]} |G(p(\boldsymbol{r}, \theta), f) - G_d(p(\boldsymbol{r}, \theta), f)|^2 =$$
$$\max\left\{|G(p(\boldsymbol{r}, -\eta), f) - G_d(p(\boldsymbol{r}, -\eta), f)|^2, |G(p(\boldsymbol{r}, \eta), f) - G_d(p(\boldsymbol{r}, \eta), f)|^2\right\} + |o(\eta)|. \tag{2.19}$$

**Proof:** Denote $H(f) = (H_1(f), \ldots, H_N(f))^\intercal$, where $H_i(f) = \boldsymbol{w}_i^\intercal \boldsymbol{d}_0(f)$. Then, the actual frequency response $G(p(\boldsymbol{r}, \theta), f)$ can be reformulated as

$$G(p(\boldsymbol{r}, \theta), f) = H^\intercal(f) \boldsymbol{b}(p(\boldsymbol{r}, \theta), f). \tag{2.20}$$

Denote the gradient of $\boldsymbol{b}(p(\boldsymbol{r}, \theta), f)$ with respect to the parameter $\theta$ as

$$\frac{\partial \boldsymbol{b}(p(\boldsymbol{r}, \theta), f)}{\partial \theta} = \left(\frac{\partial B_1(p(\boldsymbol{r}, \theta), f)}{\partial \theta}, \ldots, \frac{\partial B_N(p(\boldsymbol{r}, \theta), f)}{\partial \theta}\right)^\intercal. \tag{2.21}$$

Then, since $p(\boldsymbol{r},0)=\boldsymbol{r}$ and $\eta$ is small, we can rewrite $G(p(\boldsymbol{r},\theta),f)$ as

$$G(p(\boldsymbol{r},\theta),f)=H^{\intercal}(f)\left(\boldsymbol{b}(\boldsymbol{r},f)+\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}\theta\right)+o(\theta). \tag{2.22}$$

Similarly, $G_d(p(\boldsymbol{r},\theta),f)$ can be rewritten as

$$G_d(p(\boldsymbol{r},\theta),f)=G_d(\boldsymbol{r},f)+\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\theta+o(\theta). \tag{2.23}$$

Then, we have

$$\begin{aligned}
&|G(p(\boldsymbol{r},\theta),f)-G_d(p(\boldsymbol{r},\theta),f)|^2\\
&=\left|G(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)+\left(H^{\intercal}(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}-\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\theta+o(\theta)\right|^2\\
&=\left|G(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)+\left(H^{\intercal}(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}-\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\theta\right|^2+|o(\theta)|.
\end{aligned} \tag{2.24}$$

Note that the first term of the right hand side of (2.24) is convex with respect to $\theta$ and the maximum of a convex function exists when $\theta$ is in the boundary. Then, we have

$$\begin{aligned}
&\max_{\theta\in[-\eta,\eta]}\left|G(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)+\left(H^{\intercal}(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}-\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\theta\right|^2\\
&=\max\left\{\left|G(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)+\left(H^{\intercal}(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}-\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\eta\right|^2,\right.\\
&\qquad\left.\left|G(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)-\left(H^{\intercal}(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}-\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\eta\right|^2\right\}\\
&=\max\left\{|G(p(\boldsymbol{r},-\eta),f)-G_d(p(\boldsymbol{r},-\eta),f)+o(\eta)|^2,\right.\\
&\qquad\left.|G(p(\boldsymbol{r},\eta),f)-G_d(p(\boldsymbol{r},\eta),f)+o(\eta)|^2\right\}\\
&=\max\left\{|G(p(\boldsymbol{r},-\eta),f)-G_d(p(\boldsymbol{r},-\eta),f)|^2,\right.\\
&\qquad\left.|G(p(\boldsymbol{r},\eta),f)-G_d(p(\boldsymbol{r},\eta),f)|^2\right\}+|o(\eta)|.
\end{aligned} \tag{2.25}$$

This completes the proof.

By Theorem 3.2, if $\eta$ is small, Problem 5 can be simplified to

**Problem 6.** *Find a coefficient vector $\boldsymbol{w}\in\mathbb{R}^n$ of the beamformer filters and $z$, such that $z$ is minimized, subject to the constraint*

$$\hat{G}(z,\boldsymbol{w})\succeq 0, \tag{2.26}$$

*where*

$$\begin{aligned}
\hat{G}(z,\boldsymbol{w})=diag\{&\Phi(z,\boldsymbol{w},p(\boldsymbol{r}^1,\eta),f^1),\Phi(z,\boldsymbol{w},p(\boldsymbol{r}^1,-\eta),f^1),\\
&\ldots,\Phi(z,\boldsymbol{w},p(\boldsymbol{r}^k,\eta),f^k),\Phi(z,\boldsymbol{w},p(\boldsymbol{r}^k,-\eta),f^k)\}.
\end{aligned} \tag{2.27}$$

3.2. **Localization of microphones.** Let $\boldsymbol{R}(=[\boldsymbol{r}_1,\boldsymbol{r}_2,\ldots,\boldsymbol{r}_n])\in\mathbb{R}^{h\times N}$ be the unknown matrix. Then, we have

$$\|\boldsymbol{r}_i-\boldsymbol{r}_j\|^2=\boldsymbol{e}_{ij}^{\intercal}\boldsymbol{R}^{\intercal}\boldsymbol{R}\boldsymbol{e}_{ij}$$

$$\|\boldsymbol{a}_k-\boldsymbol{r}_j\|^2=\begin{pmatrix}\boldsymbol{a}_k^{\intercal} & \boldsymbol{e}_j^{\intercal}\end{pmatrix}\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{R}^{\intercal}\end{bmatrix}[\boldsymbol{I}\ \boldsymbol{R}]\begin{pmatrix}\boldsymbol{a}_k\\ \boldsymbol{e}_j\end{pmatrix},$$

where $\boldsymbol{e}_{ij}$ is the vector with 1 at the $i$-th position, $-1$ at the $j$-th position and 0 elsewhere, $\boldsymbol{e}_j$ is the vector with $-1$ at the $j$-th position and 0 elsewhere. Let $\boldsymbol{Y}=\boldsymbol{R}^{\intercal}\boldsymbol{R}$, then Problem

2 is equivalent to find a symmetric matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times N}$ and a matrix $\boldsymbol{R} \in \mathbb{R}^{h\times N}$ such that the following equations are satisfied:

$$\begin{aligned}
&\boldsymbol{e}_{ij}^{\intercal} Y \boldsymbol{e}_{ij} = \hat{d}_{ij}^2, && \forall (i,j) \in N_r \\
&\left(\boldsymbol{a}_k^{\intercal}\ \boldsymbol{e}_j^{\intercal}\right) \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^{\intercal} & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_k \\ \boldsymbol{e}_j \end{pmatrix} = \hat{d}_{kj}^2, && \forall (k,j) \in N_a \\
&\boldsymbol{Y} = \boldsymbol{R}^{\intercal}\boldsymbol{R}.
\end{aligned} \tag{3.3}$$

To relax the sensor network localization problem, we relax $\boldsymbol{Y} = \boldsymbol{R}^{\intercal}\boldsymbol{R}$ to $\boldsymbol{Y} \succeq \boldsymbol{R}^{\intercal}\boldsymbol{R}$ which is equivalent to [31]:

$$\boldsymbol{Z} := \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^{\intercal} & \boldsymbol{Y} \end{pmatrix} \succeq 0.$$

Then, the relaxed version of the problem (3.3) can be represented as a standard semi-definite programming model, that is, we need to find a symmetric matrix $\boldsymbol{Z} \in \mathbb{R}^{(h+N)\times(h+N)}$ such that the following equations are satisfied:

$$\begin{aligned}
&(\boldsymbol{b}^{\intercal}\ \boldsymbol{0}^{\intercal})\boldsymbol{Z}\begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{pmatrix} = \boldsymbol{b}^{\intercal}\boldsymbol{b}, \quad \text{for some vectors } \boldsymbol{b} \in \mathbb{R}^h \\
&(\boldsymbol{0}^{\intercal}\ \boldsymbol{e}_{ij}^{\intercal})\boldsymbol{Z}\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{e}_{ij} \end{pmatrix} = \hat{d}_{ij}^2, \quad \forall (i,j) \in N_r \\
&\left(\boldsymbol{a}_k^{\intercal}\ \boldsymbol{e}_j^{\intercal}\right)\boldsymbol{Z}\begin{pmatrix} \boldsymbol{a}_k \\ \boldsymbol{e}_j \end{pmatrix} = \hat{d}_{kj}^2, \quad \forall (k,j) \in N_a \\
&\boldsymbol{Z} \succeq 0.
\end{aligned} \tag{3.4}$$

The first set of equations in (3.4) is to assure that the first $h \times h$ submatrix of $\boldsymbol{Z}$ is $\boldsymbol{I}$. The number of the vector $\boldsymbol{b}$ depends on the dimension $h$. If $h = 1$, there is only one element in $\boldsymbol{I}$ and the minimum number of $b$ is 1. If $h = 2$, since $\boldsymbol{I}$ is symmetric, there are three elements to be determined and the minimum number of $\boldsymbol{b}$ is 3. If $h = 3$, there are six elements in the symmetric matrix $\boldsymbol{I}$ to be determined and the minimum number of $\boldsymbol{b}$ is 6. There are many choices for $\boldsymbol{b}$. An example can be seen in Table 1.

TABLE 1. A typical choice of the vector $\boldsymbol{b}$

| $h$ | $\boldsymbol{b}$ |
|---|---|
| 1 | 1 |
| 2 | $(1\ 0)^{\intercal}$, $(0\ 1)^{\intercal}$, $(1\ 1)^{\intercal}$ |
| 3 | $(1\ 0\ 0)^{\intercal}$, $(0\ 1\ 0)^{\intercal}$, $(0\ 0\ 1)^{\intercal}$, $(1\ 1\ 0)^{\intercal}$, $(1\ 0\ 1)^{\intercal}$, $(0\ 1\ 1)^{\intercal}$ |

A relaxed solution $\boldsymbol{Z}$ can be obtained by solving Equations (3.4). However, as we discuss in Section 2, the second and third set of equations in (3.4) are not satisfied in most cases and the solution does not exist. Therefore, we need to consider Problem 2. To transform Problem 2 into a semi-definite programming problem, we add some nonnegative slack variables as $\boldsymbol{\alpha} = \{\alpha_{ij}^+, \alpha_{ij}^-, \alpha_{kj}^+, \alpha_{kj}^- :\geq 0, \forall (i,j) \in N_r, \forall (k,j) \in N_a\}$. Then, Problem 2 is reformulated as

**Problem 7.** *Find $\boldsymbol{\alpha}$ and the locations $\boldsymbol{r}$ of the sensors, such that the cost function*

$$\sum_{(i,j)\in N_r} (\alpha_{ij}^+ + \alpha_{ij}^-) + \sum_{(k,j)\in N_a} (\alpha_{kj}^+ + \alpha_{kj}^-)$$

*is minimized, subject to the constraints*

$$\begin{aligned}
\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 - \hat{d}_{ij}^2 &= \alpha_{ij}^+ - \alpha_{ij}^-, \quad \forall (i,j) \in N_r \\
\|\boldsymbol{a}_k - \boldsymbol{r}_j\|^2 - \hat{d}_{kj}^2 &= \alpha_{kj}^+ - \alpha_{kj}^-, \quad \forall (k,j) \in N_a.
\end{aligned} \tag{3.5}$$

With the introduced relaxed matrix $\boldsymbol{Z}$, Problem 7 is transformed into a standard SDP problem:

**Problem 8.** *Find $\boldsymbol{\alpha}$ and the symmetric matrix $\boldsymbol{Z}$, such that*

$$\sum_{(i,j)\in N_r}(\alpha_{ij}^{+}+\alpha_{ij}^{-})+\sum_{(k,j)\in N_a}(\alpha_{kj}^{+}+\alpha_{kj}^{-})$$

*is minimized, subject to the constraints*

$$\begin{aligned}
&(\boldsymbol{b}^{\intercal}\ \boldsymbol{0}^{\intercal})\boldsymbol{Z}\begin{pmatrix}\boldsymbol{b}\\ \boldsymbol{0}\end{pmatrix}=\boldsymbol{b}^{\intercal}\boldsymbol{b}, && \textit{for some vectors } \boldsymbol{b}\in\mathbb{R}^{h}\\
&(\boldsymbol{0}^{\intercal}\ \boldsymbol{e}_{ij}^{\intercal})\boldsymbol{Z}\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{e}_{ij}\end{pmatrix}-\alpha_{ij}^{+}+\alpha_{ij}^{-}=\hat{d}_{ij}^{2}, && \forall(i,j)\in N_r\\
&\left(\boldsymbol{a}_k^{\intercal}\ \boldsymbol{e}_j^{\intercal}\right)\boldsymbol{Z}\begin{pmatrix}\boldsymbol{a}_k\\ \boldsymbol{e}_j\end{pmatrix}-\alpha_{kj}^{+}+\alpha_{kj}^{-}=\hat{d}_{kj}^{2}, && \forall(k,j)\in N_a\\
&\boldsymbol{Z}\succeq 0.
\end{aligned}\tag{3.6}$$

Problem 8 is a semi-definite programming problem which can be solved by any SDP software. Note that any solution of Problem 8 has at least rank $h$. For a localizable system, we need to impose certain conditions on the rank of $\boldsymbol{Z}$ and the relaxation of $\boldsymbol{Y}$. This is summarized in the following.

**Definition 3.1.** *The localization problem is localizable if there is a unique localization in $\mathbb{R}^h$ and there is no $\boldsymbol{r}_j\in\mathbb{R}^{h'}$, $j=1,\ldots,n$, where $h'>h$, such that*

$$\|\boldsymbol{r}_i-\boldsymbol{r}_j\|^2=d_{ij}^2,\quad\forall(i,j)\in N_r$$

$$\left\|\begin{pmatrix}\boldsymbol{a}_k\\ \boldsymbol{0}\end{pmatrix}-\boldsymbol{r}_j\right\|^2=d_{kj}^2,\quad\forall(k,j)\in N_a.$$

*The latter says that the problem cannot be localized in a higher dimension space where the locations of the anchors are augmented to $(\boldsymbol{a}_k^{\intercal}\ \boldsymbol{0}^{\intercal})^{\intercal}\in\mathbb{R}^{h'}$, $k\in M^{(1)}$.*

Then, we have the following theorems (proven in [27]):

**Theorem 3.3.** *The following statements are equivalent:*

1. *The problem is localizable.*
2. *The max rank of the solution $\boldsymbol{Z}$ has rank $h$.*
3. *The solution $\boldsymbol{Z}$ satisfy $\boldsymbol{Y}=\boldsymbol{R}^{\intercal}\boldsymbol{R}$ or Trace$(\boldsymbol{Y}-\boldsymbol{R}^{\intercal}\boldsymbol{R})=0$.*

**Theorem 3.4.** *If a problem contains a subproblem that is localizable, then the submatrix solution corresponding to the subproblem in the SDP solution has rank $h$. That is, the SDP relaxation computes a solution that localizes all possibly localizable unknown sensor points.*

From these two theorems, we can see that the solution to the SDP problem provides the first and second moment information on $\boldsymbol{R}$ [32]. After we find a solution $\boldsymbol{Z}$ by solving Problem 8, $\boldsymbol{r}_j$ will be the estimated position of $j$-th microphone and $Y_{jj}-\|\boldsymbol{r}_j\|^2$ will be used as its perturbation. The total perturbation of the microphones is then given by

$$Trace(\boldsymbol{Y}-\boldsymbol{R}^{\intercal}\boldsymbol{R})=\sum_{j=1}^{N}\left(Y_{jj}-\|\boldsymbol{r}_j\|^2\right).$$

4. **Illustrative Examples.** In solving the formulated linear SDP problems (Problem 4, Problem 5, Problem 6 and Problem 8), interior point algorithms can be applied. There are several software packages available, such as LMI control toolbox [33], SDPA-M [34], SDPSOL [35] and SeDuMi [36]. All these software packages can be applied. In this section, we use SDPA-M [34] and the computation was performed in Matlab.

The proposed method is first used to design several broadband beamformers with different target performances. At the same time, we will study the performances of the designs towards errors in speaker and microphone locations. We focus on multimedia applications and the desired frequency response function will include the frequency range of human voice together with a range of positions that the speaker is located. We choose the desired response function as

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

where $\boldsymbol{r}_c$ is the coordinate for the center element, the sound speed is $c = 340.9m/s$ and the sample increment is $T = 125\mu s$, that is, the sampling rate is set as 8kHz.

In the first example, we consider an equispaced linear array with five elements. To avoid spatial aliasing for the frequency of interest, the element spacing is 5cm. That is, they are located at the coordinates $\{(-0.1, 0), (-0.05, 0), \ldots, (0.1, 0)\}$. A seven-tap FIR filter behind each element is used. The passband region and stopband region are specified on an $x$-axis parallel with, and $y = 1$ meter in front of, the array. The passband region is defined as

$$\{(\boldsymbol{r}, f) : -0.4m \le x \le 0.4m,\ y = 1m,\ 0.5kHz \le f \le 1.5kHz\}$$

while the stopband region is the union of several parts as

$$\{(\boldsymbol{r}, f) : -0.4m \le x \le 0.4m,\ y = 1m,\ 2.5kHz \le f \le 4kHz\},$$
$$\{(\boldsymbol{r}, f) : 1.5m \le |x| \le 2.5m,\ y = 1m,\ 0.5kHz \le f \le 1.5kHz\},$$
$$\{(\boldsymbol{r}, f) : 1.5m \le |x| \le 2.5m,\ y = 1m,\ 2.5kHz \le f \le 4kHz\}.$$

The complexity of the implementation depends on the discretization of the space-frequency domain $\Omega$ in this problem. Suppose that the number of discretization of $\Omega$ is given by $m_x \times m_f$. Then, for different numbers of discretization, the comparison of our method with the SIP method [13] is given in Table 2.

Table 2. Comparison of the running times (seconds)

| $m_x \times m_f$ | LP | SDP |
|---|---|---|
| $40 \times 40$ | $17.66s$ | $8.98s$ |
| $80 \times 80$ | $299.63s$ | $36.97s$ |
| $120 \times 120$ | $1490.25s$ | $85.04s$ |
| $130 \times 130$ | $2192.06s$ | $102.89s$ |

From Table 2, we see that our method is more efficient than SIP method [13], especially when the number of discretization becomes very large. The amplitude of the actual response $G(\boldsymbol{r}, f)$ is shown in Figure 3. We consider some perturbations to the design parameters. When the speaker location changes, we found that the optimized performance is very similar with or without robustness in the formulation. This again confirms the findings in [13] that the optimal design is not too sensitive to the movement of speaker.

We then consider possible errors in the first and the last microphones. We set $x_1 \in [-0.15, -0.07]$ and $x_5 \in [0.07, 0.15]$. When the $x$-coordinate of the first and the last

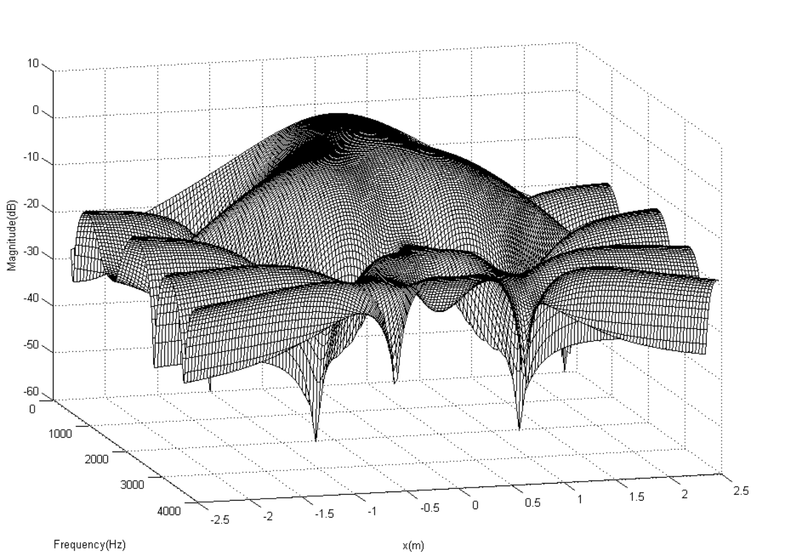


FIGURE 3. Amplitude of $G(\boldsymbol{r}, f)$ in Example 1 where $N = 5$, $L = 7$ and $y = 1m$

microphone move to $-0.15$m and $0.15$m, respectively, the performance is summarized in Table 3. From the results, it is clear that the optimized beamformer is very sensitive to perturbations in microphone locations. With the use of the robust formulation, the optimized performance is recovered in spite of the perturbations.

TABLE 3. Optimized designs with errors in microphone locations

| Methods | Passband gain | Passband ripple (dB) | Stopband ripple (dB) |
|---|---|---|---|
| Robust design | 1.03146 | 0.20281 | $-14.23836$ |
| Normal design | 1.05639 | 0.36851 | $-6.87912$ |

In the second example, we demonstrate how to incorporate the sensor localization method together with the robust design formulation. We consider a 5m $\times$ 5m classroom and the speaker stands in the middle of the room. At the corner of the room, there are two anchors with coordinates $\{(-2.5, 2.5), (2.5, 2.5)\}$. Seven microphones are located at the coordinates $\{(-0.15, 0.7), (-0.1, 0.8), (-0.05, 0.9), (0, 1), (0.05, 0.9), (0.1, 0.8), (0.15, 0.7)\}$. We assume the distances between the nodes can be estimated and there exist errors in the estimated distances. We simulate the estimated distances similar to [29]. That is, we add a random error to the estimated distance:

$$\hat{d_{ij}} = d_{ij} \cdot (1 + \epsilon \times N_f)$$

where $N_f$ is a given noisy factor between $[0, 1]$ and $\epsilon$ is a standard normal random variable. For this example, we further assume that the microphone with coordinate $(0, 1)$ is also an anchor, and all the other microphones are sensors. The noisy factor is chosen as 0.005 and the distances between the nodes can be estimated. In the first stage, we estimate the microphones' positions by solving Problem 8. The estimated positions are illustrated in Figure 4 and the perturbations of these six sensors are $0.0139, 0.0007, 0.0004, 0.0005, 0.0004$ and $0.0018$, respectively. Since the perturbations of three sensors with coordinates $(-0.05, 0.9)$, $(0.05, 0.9)$ and $(0.1, 0.8)$ are very small, they are neglected and we just consider the uncertainties of the other three sensors.

In the second stage, with the estimated positions and the perturbations, we design a robust beamformer with a seven-tap filter behind each microphone. The passband

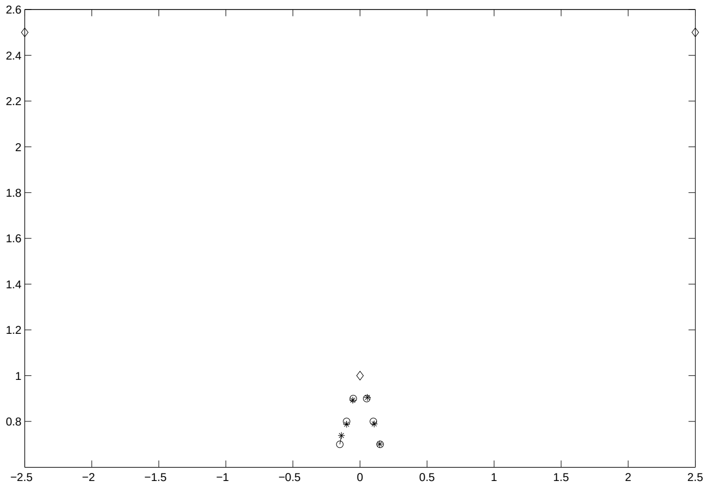

Figure 4. Estimated positions via solving the localization problem

region, stopband region and the desired frequency response function are chosen in the same manner as in Example 1. By solving Problem 6, we obtain the optimal design. The amplitude of the actual response $G(r, f)$ is shown in Figure 5.

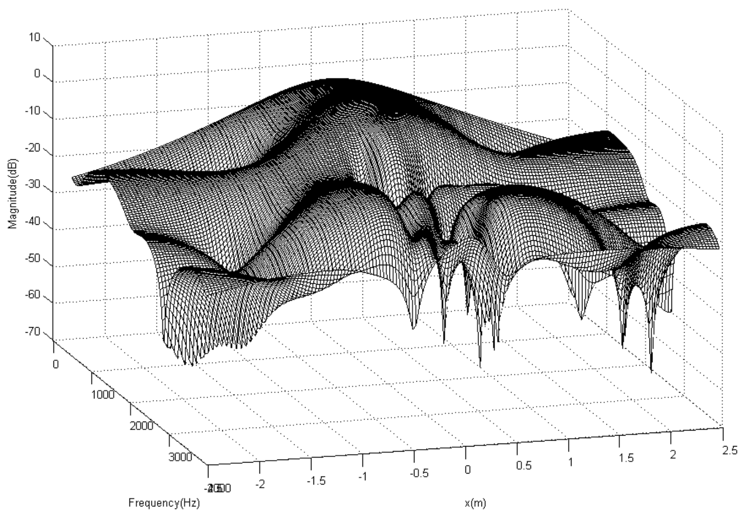


Figure 5. Amplitude of $G(\boldsymbol{r}, f)$ in Example 2

5. **Conclusions.** In this paper, we have formulated the design problem of distributed broadband beamformers with wireless microphones. The sensor network technology has been used to estimate sensor microphone locations and was incorporated into the design process. We have studied the performance of the optimized designs and found that it was very sensitive to perturbations in microphone locations. We have proposed a suitable robust formulation as a remedy to regain the performance. From the examples, we demonstrated that this approach is essential to regain accuracy in the optimized designs.

**Acknowledgment.** This paper is supported by RGC Grant PolyU. (5365/09E) and Research Committee of the Hong Kong Polytechnic University. The first and last authors would also like to thank the support of NSFC Grant (10901170). Finally, we wish to especially thank Prof. Ye Yinyu for encouragement and fruitful discussion.

## REFERENCES

[1] B. V. Veen and K. Buckley, Beamforming: A versatile approach to spatial filtering, *IEEE ASSP Magazine*, vol.5, no.2, pp.4-24, 1988.
[2] H. L. V. Trees, *Optimum Array Processing*, Wiley, NY, 2002.
[3] M. Khosravy, M. R. Asharif and K. Yamashita, A pdf-matched short-term linear predictability approach to blind source separation, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3677-3690, 2009.
[4] Z. Zhang, K. I. T. Miyake and T. Imamura, Three-dimension sound localization by binaural model using self organizing map, *International Journal of Innovative Computing, Information and Control*, vol.6, no.1, pp.361-371, 2010.
[5] O. Hoshuyama, A. Sugiyama and A. Hirano, A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters, *IEEE Trans. on Signal Processing*, vol.47, no.10, pp.2677-2684, 1999.
[6] S. Gannot, D. Burshtein and E. Weinstein, Signal enhancement using beamforming and nonstationarity with applications to speech, *IEEE Trans. on Signal Processing*, vol.49, no.8, pp.1614-1626, 2001.
[7] S. C. Chan and H. H. Chen, Uniform concentric circular arrays with frequency-invariant characteristics-theory, design, adaptive beamforming and DOA estimation, *IEEE Trans. on Signal Processing*, vol.55, no.1, pp.165-177, 2007.
[8] R. Kennedy, T. Abhayapala and D. Ward, Broadband nearfield beamforming using radial beampattern transformation, *IEEE Trans. on Signal Processing*, vol.46, pp.2147-2156, 1998.
[9] J. Ryan and R. Goubran, Array optimization applied in the near field of a microphone array, *IEEE Trans. on Speech Audio Processing*, vol.8, no.2, pp.173-178, 2000.
[10] S. Nordebo, I. Claesson and S. Nordholm, Weighted chebyshev approximation for the design of broadband beamformers using quadratic programming, *IEEE Signal Processing Letters*, vol.1, pp.103-105, 1994.
[11] S. Nordholm, V. Rehbock, K. L. Teo and S. Nordebo, Chebyshev approximation for the design of broadband beamformers in the near field, *IEEE Trans. on Circuits and Systems II*, vol.45, no.1, pp.141-143, 1998.
[12] B. Lau, Y. Leung, K. L. Teo and V. Sreeram, Minimax filters for microphone arrays, *IEEE Trans. on Circuits and Systems II*, vol.46, no.12, pp.1522-1525, 1999.
[13] K. F. C. Yiu, X. Q. Yang, S. Nordholm and K. L. Teo, Near-field broadband beamformer design via multidimensional semi-infinite linear programming techniques, *IEEE Trans. on Speech and Audio Processing*, vol.11, no.6, 2003.
[14] H. Dam, K. Teo, S. Nordebo and A. Cantoni, The dual parametrization approach to optimal least square fir filters design subject to maximum error constraints, *IEEE Trans. on Signal Processing*, vol.48, pp.2314-2320, 2000.
[15] Y. Yu, K. Teo, Y. Lim and G. Zhao, Extrapolated impulse response filter and its application in the synthesis of digital filters, *Signal Processing*, vol.85, no.3, pp.581-590, 2005.
[16] W. Lee, L. Caccetta, K. Teo and V. Rehbock, Optimal design of complex fir filters with arbitrary magnitude and group delay responses, *IEEE Trans. on Signal Processing*, vol.54, pp.1617-1627, 2006.
[17] Y. C. Lim and Y. Lian, The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique, *IEEE Trans. on Circuits and Systems II*, vol.40, no.2, pp.88-95, 1993.
[18] J. Choi and C. Lim, A cholesky factorization based approach for blind FIR channel identification, *IEEE Trans. on Signal Processing*, vol.56, no.4, pp.1730-1735, 2008.
[19] H. Wolkowicz, R. Saigal and L.Vandenberghe, *Handbook of Semidefinite Programming, Theory, Algorithms and Applications*, Kluwer Academic, 2003.
[20] S. Li, X. Yang, K. Teo and S. Wu, A solution method for combined semi-infinite and semi-definite programming, *ANZIAM Journal*, vol.45, pp.477-494, 2003.
[21] X. Huang, X. Yang and K. Teo, A sequential quadratic penalty method for nonlinear semidefinite programming, *Optimization*, vol.52, pp.715-738, 2003.

[22] W. S. Lu and T. Hinamoto, Optimal design of frequency-response-masking filters using semidefinite programming, *IEEE Trans. on Circuits and Systems – I: Fundamental Theory and Applications*, vol.50, no.4, 2003.

[23] F. Wang, V. Balakrishnan, Y. Zhou, J. Chen, R. Yang and C. Frank, Optimal array pattern synthesis using semidefinite programming, *IEEE Trans. on Signal Processing*, vol.51, no.5, 2003.

[24] H. H. Kha, H. D. Tuan, B.-N. Vo and T. Q. Nguyen, Symmetric orthogonal complex-valued filter bank design by semidefinite programming, *IEEE Trans. on Signal Processing*, vol.55, no.9, pp.4405-4414, 2007.

[25] P. Biswas and Y. Ye, Semidefinite programming for ad hoc wireless sensor network localization, *Proc. of the 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, USA, pp.46-54, 2004.

[26] P. Biswas, T. Lian, T. Wang and Y. Ye, Semidefinite programming based algorithms for sensor network localization, *ACM Trans. on Sensor Networks*, vol.2, no.2, pp.188-120, 2006.

[27] A. So and Y. Ye, Theory of semidefinite programming for sensor network localization, *Mathematical Programming*, vol.109, no.2-3, pp.367-384, 2007.

[28] P. Biswas, T. Liang, K. Toh, Y. Ye and T. Wang, Semidefinite programming approaches for sensor network localization with noisy distance measurements, *IEEE Trans. on Automation Science and Engineering*, vol.3, no.4, pp.360-371, 2006.

[29] Z. Wang, S. Zheng, S. Boyd and Y. Ye, Further relaxations of the SDP approach to sensor network localization, *SIAM J. Optimization*, vol.19, no.2, pp.655-673, 2008.

[30] Z. Chen, *Higher Algebra and Analytic Geometry*, China Higher Education Press Beijing and Springer-Verlag Berlin Heidelberg, 2001.

[31] S. Boyd, L. Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, 1994.

[32] D. Bertsimas and Y. Ye, Semidefinite relaxations, multivariate normal distributions, and order statistics, *Hankbook of Combinatorial Optimization*, vol.3, pp.1-19, 1998.

[33] P. Gahinet, A. Nemirovski, A. J. Laub and M. Chilali, LMI *Control Toolbox for Use with Matlab User's Guide Version 1*, 1995.

[34] K. Fujisawa, Y. Futakata, M. Kojima, S. Matsuyama, S. Nakamura, K. Nakata and M. Yamashita, Seidefinite programming algorithm in matlab user's manual – Version 6.2.0, *Research Reports on Mathematical and Computing Sciences*, 2005.

[35] S.-P. Wu and S. Boyd, Sdpsol: A parser/solver for semidefinite programs with matrix structure, in *Advances in Linear Matrix Inequality Methods in Control*, L. El Ghaoui and S.-L. Niculescu (eds.), SIAM, 1996.

[36] J. F. Sturm, *Using Sedumi 1.02, a Matlab Toolbox for Optimization over Symmertric Cones*, Department of Econometrics, Tilburg University, 1998.