RESEARCH

Open Access

Design of broadband beamformers with low complexity

Zhiguo Feng[1], Cedric Ka Fai Yiu[1*], Kok-Lay Teo[2] and Sven Nordholm[3]

Abstract

In this article, we consider the design of broadband beamformers with low complexity. In fact, the design problem is multi-objective in nature, trading off between speech distortion and noise suppression. Finding a balance between these two objectives is important in order to achieve a desired sound quality. These measures are introduced as the objectives here. The design can then be obtained via a bi-objective integer programming problem, where the coefficients of the filters are expressed as sums of signed powers-of-two terms. We study two different integer spaces and penalty functions for solving the problem. Then, an algorithm based on a discrete filled function is developed for finding the optimal design. In order to illustrate the effectiveness of the algorithm, real data is used and two broadband beamformers are demonstrated.

1 Introduction

The increased popularity of wireless cellular telephones and their uses in a variety of occasions has motivated the development of handsfree communication devices. In this particular acoustic environment, the microphone array is required to suppress the car noise as well as the interference from the handsfree loudspeaker, while keeping the distortion of the speech low. Since the mathematical model of this problem is very difficult to construct, sequences of calibration signals are used instead for the design of beamformers [1].

Under this signal model, the least-squares technique (LS) and the signal-to-noise plus interference ratio (SNIR) are often used [2,3] to optimize the performance of the beamformer. However, evaluation results [3] have shown that beamformers designed by LS have very good distortion controls, while the ones designed by SNIR have better suppression levels of both noise and interference when compared to LS. Both approaches cannot control directly the individual level of speech distortion, noise suppression and interference suppression. This problem is partially overcome in [4], where it was proposed to use nonlinear programming to design beamformers with multi-criteria and demonstrated the set of Pareto optimal. Indeed, trade-offs between different performance indicators in different

beamforming systems are important and have been studied in the literature [5-10]. However, after achieving the required performances, coefficients of the designed beamformers are of very high precision and hence require significant efforts in order to preserve the performance and trade off the computational efficiency. Indeed, in many designs, the truncation method is still widely employed and quantization errors play a significant role in the dynamic range of filter gain and increase with filter order [11,12]. Furthermore, an efficient implementation of the beamformers in fixed-point arithmetic hardware (such as FPGA) is essential [13] in the production stage. However, there is very little result for finding the finite precision beamformers subject to achieving a certain speech quality.

One way to achieve low complexity is to express the coefficients of the filters as sums of signed powers-of-two (SPT) terms, and minimize the number of SPT terms required. This problem has received a great attention. For the design of high-pass or low-pass finite impulse response (FIR) filters, the use of SPT terms via the least squares criterion or the minimax criterion have been widely studied in the literature. Several optimization methods, such as branch and bound [14], simulated annealing [15], and searching techniques [16,17] have been proposed to tackle this class of problems. For the design of the filter bank, several algorithms have been proposed, such as the genetic algorithm [18], and the tree search algorithm [19]. However, these approaches have

* Correspondence: macyiu@polyu.edu.hk
[1]Department of Applied Mathematics, The Hong Kong Polytechnic
University, Kowloon, Hong Kong, PR China
Full list of author information is available at the end of the article

not been applied for broadband beamformer designs yet, especially when we consider the model of optimizing signal distortion and noise suppression directly. Effective algorithms are required to tackle this new constrained optimization problem.

The problem of designing FIR filters with low complexity is always formulated as a constrained optimization problem, where the variables take values in -1, 0, and 1. However, the number of variables for this formulation is very large and the proposed methods, such as branch and bound and the tree search algorithm, are very expensive. The heuristic methods such as the genetic algorithm and simulated annealing are not only expensive but also unpredictable. To reduce the number of variables, it is necessary to find a fast conversion between an integer and its minimum combination of signed power of two terms. In [20], a method is proposed to allocate the number of SPT terms to each coefficient value, but the optimum assignment scheme is not guaranteed. In this article, a fast and optimal conversion based on a recursive function is introduced and the problem is transformed into an integer programming problem with the number of variables greatly reduced. For this problem, the optimal solution should be close to the infinite precision solution, which can easily be obtained by gradient-based methods. Hence, the truncation of the infinite precision solution can be set as a good initial point and we are required to search the neighborhood of the initial point in the integer space. A suitable technique is the filled function method, which was first introduced in [21] for global optimization with continuous variables employed in a hybrid setting similar to [22]. It searches for a better minimizer among local minimizers by means of a function, which is called a filled function. A discrete filled function method was later developed in [23] for solving discrete global optimization problems. In this article, a novel method is proposed for the finite precision beamformer design using a discrete filled function. We formulate the design problem and transform it into an unconstrained integer programming problem. By incorporating a procedure for choosing initial points and using a discrete filled function, we develop an efficient algorithm to tackle the problem.

The rest of the article is organized as follows. In Section 2, with the calibration signal, we present the signal model and formulate the beamformer design problem. In Section 3, we transform the original problem into an unconstrained integer programming problem. An algorithm is then developed in Section 4, which is then applied for the off-line design of beamformers. Two examples are given in Section 5 and the numerical results obtained are compared with other methods.

2 Problem formulation

Unlike the single-channel case, multi-channel beamforming techniques exploit the properties of spatial and temporal distributions of both the speech and noise sources to enhance the performance. The structure of a linear FIR beamformer with several channels is shown in Figure 1.

We assume that there are M elements in the microphone array. In general, the signals received by the ith microphone element can be represented as

$$x^i(n) = x^i_s(n) + x^i_N(n) + x^i_I(n), \quad i = 1, 2, \ldots, M, \quad (2.1)$$

where $x^i_s(n)$, $x^i_N(n)$, and $x^i_I(n)$ are the source signal, the noise signal and the interference signal, respectively, and n is the discrete time index. Assume that known calibration sequence observations are used for each of these signals. The output of the beamformer is given by

$$y(n) = \sum_{i=1}^{M} \sum_{j=0}^{L-1} w^i(j) x^i(n-j), \quad (2.2)$$

where L is the length of the filters and $w = ((w^1)^\top, \ldots, (w^M)^\top)^\top$, where

$$w^i = (w^i(0), w^i(1), \ldots, w^i(L-1))^\top, \quad i = 1, \ldots, M,$$

are the weight vectors. The coefficients $w^i(j)$, $i = 1, \ldots, M$, $j = 0, \ldots, L - 1$, are expressed by

$$w^i(j) = \sum_{k=1}^{b} s_{k,i,j} 2^{-k}, \quad i = 1, \ldots, M, j = 0, \ldots, L - 1 \quad (2.3)$$

where b is the wordlength and $s_{k,i,j} \in \{-1, 0, 1\}$. To reduce the hardware complexity in real applications, the coefficients need to satisfy the constraints
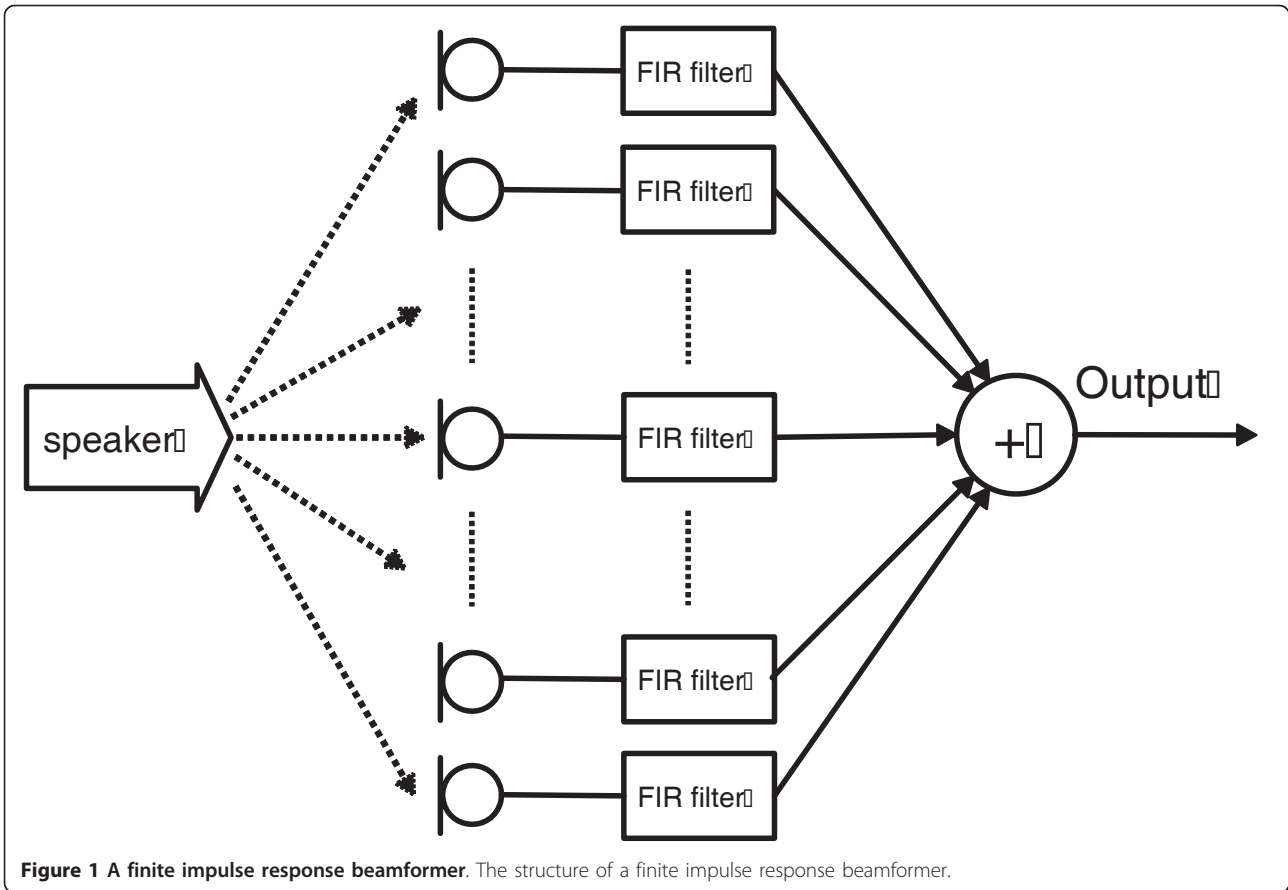
$$\sum_{k=1}^{b} \sum_{i=1}^{M} \sum_{j=0}^{L-1} |s_{k,i,j}| \le N_1, \quad (2.4)$$

where N_1 is the total allowable number of the SPT terms used, and

$$\sum_{k=1}^{b} |s_{k,i,j}| \le N_2, \quad i = 1, \ldots, M, j = 0, \ldots, L - 1, \quad (2.5)$$

where N_2 is the allowable number of SPT terms for each coefficient $w^i(j)$. Let $\mathcal{W}$ denote the set of all those w such that the constraints (2.4) and (2.5) are satisfied.

Basically, the LS formulation and the SNIR formulation are two different kinds of methods for determining the weight matrix. In both approaches, there is no direct control over the level of distortion and the level of noise and interference suppression. In particular, LS is very good at

Figure 1 A finite impulse response beamformer. The structure of a finite impulse response beamformer.

the distortion level, but poor in the suppression level. On the other hand, SNIR is always very good at the suppression level but left a consistently high level of distortion.

For beamformers, it is desired to maximize the noise and interference suppression, while keeping the distortion caused by the beamforming filters at the minimum. In order to measure various quantities, calibration signals are used. Define $\hat{P}_{x_s^r x_s^r}(\omega)$ as the power spectrum estimate of the source signal without filtering, where the rth microphone is chosen as reference, and $\hat{P}_{y_s y_s}(\omega)$ is the power spectrum estimate of the beamformer output with filtering when the source signal is active alone. The normalized distortion measure can be defined as:

$$D_r(\boldsymbol{w}) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|C_d\hat{P}_{y_s y_s}(\omega) - \hat{P}_{x_s^r x_s^r}(\omega)\right|d\omega = \varepsilon\left\{\left|C_d\hat{P}_{y_s y_s}(\omega) - \hat{P}_{x_s^r x_s^r}(\omega)\right|\right\}, \quad (2.6)$$

where $\varepsilon\{\bullet\}$ denotes the mean value and ω is the frequency. The constant C_d is defined as:

$$C_d = \frac{\int_{-\pi}^{\pi}\hat{P}_{x_s^r x_s^r}(\omega)d\omega}{\int_{-\pi}^{\pi}\hat{P}_{y_s y_s}(\omega)d\omega} = \frac{\varepsilon\{\hat{P}_{x_s^r x_s^r}\}}{\varepsilon\{\hat{P}_{y_s y_s}\}}, \quad (2.7)$$

Similarly, the normalized noise suppression measure and interference suppression measure are, respectively, given by

$$S_N(\boldsymbol{w}) = C_s\frac{\int_{-\pi}^{\pi}\hat{P}_{x_N^r x_N^r}(\omega)d\omega}{\int_{-\pi}^{\pi}\hat{P}_{y_N y_N}(\omega)d\omega} = C_s\frac{\varepsilon\{\hat{P}_{x_N^r x_N^r}\}}{\varepsilon\{\hat{P}_{y_N y_N}\}}, \quad (2.8)$$

and

$$S_I(\boldsymbol{w}) = C_s\frac{\int_{-\pi}^{\pi}\hat{P}_{x_I^r x_I^r}(\omega)d\omega}{\int_{-\pi}^{\pi}\hat{P}_{y_I y_I}(\omega)d\omega} = C_s\frac{E\{\hat{P}_{x_I^r x_I^r}\}}{E\{\hat{P}_{y_I y_I}\}}, \quad (2.9)$$

where $C_s = \frac{1}{C_d}$. Note that in (2.8), $\hat{P}_{y_N y_N}(\omega)$ and $\hat{P}_{x_N^r x_N^r}(\omega)$ are power spectrum estimates of the beamformer output with and without filtering, when the surrounding noise is active alone. In the same manner, $\hat{P}_{y_I y_I}(\omega)$ and $\hat{P}_{x_I^r x_I^r}(\omega)$ are power spectrum estimates when the interference signals are active alone. Both the noise and the interference suppression measures are normalized in accordance with the amplification/attenuation caused by the beamformer to the reference observation

when the source signal is active alone, i.e., when the beamformer attenuates the source signal by a specific amount, the noise and interference suppression quantities are reduced by the same amount.

In order to control the suppression and the distortion simultaneously, the filter design problem is formulated as a nonlinear programming problem given below.

Problem 1. *Find a $w \in \mathcal{W}$ such that*

$$F_1(w) \equiv -10 * (\log_{10} S_N(w) + \log_{10} S_I(w)) \qquad (2.10)$$

is minimized, subject to

$$G_1(w) \equiv 10 * \log_{10} D_r(w) - c_1 \leq 0, \qquad (2.11)$$

where c_1 is a pre-defined distortion level in dB scale.

Similar to Problem 1, we can also formulate the constrained nonlinear programming problem as:

Problem 2. *Find a $w \in \mathcal{W}$ such that*

$$F_2(w) \equiv 10 * \log_{10} D_r(w), \qquad (2.12)$$

is minimized, subject to

$$G_2(w) \equiv -10 * (\log_{10} S_N(w) + \log_{10} S_I(w)) + c_2 \leq 0, (2.13)$$

where c_2 is a pre-defined suppression level in dB scale.

Remark 1. *In some situations, the influence of interference noise can be ignored. In such cases, the term S_I in both Problems 1 and 2 can be removed.*

3 Infinite precision solution

Both Problems 1 and 2 are discrete optimization problems. There is a lack of gradient information and hence they are much more difficult to be solved than continuous optimization problems. Here, we will first solve the infinite precision solution of the continuous version of the optimization problem, and use it as an initial guess to search for a finite precision solution in the second stage.

3.1 Equivalent form

Basically, the computation of $\hat{P}_{y_s y_s}$, $\hat{P}_{y_N y_N}$ and $\hat{P}_{y_I y_I}$ are very expensive if we calculate them by using the filtered signals y_s, y_N, and y_I for every w [4]. To simplify the computation, we take the discrete time Fourier transform of both sides of (2.2) yielding

$$Y(\omega) = \sum_{k=1}^{M} W_k(\omega X_k(\omega), \qquad (3.1)$$

where $W_k(\omega)$, $X_k(\omega)$, and $Y(\omega)$ are the Fourier transforms of w^k, x^k, and y, respectively. Denote $^-$ as the conjugate symbol, we have

$$\hat{P}_{yy}(\omega) = |Y(\omega)|^2 = (\sum_{k=1}^{M} W_k(\omega) X_k(\omega))(\sum_{j=1}^{M} \bar{W}_j(\omega) \bar{X}_j(\omega))$$

$$= \sum_{k=1}^{M} \sum_{j=1}^{M} W_k(\omega) \bar{W}_j(\omega) X_k(\omega) \bar{X}_j(\omega) = \sum_{k=1}^{M} \sum_{j=1}^{M} W_k(\omega) \bar{W}_j(\omega) \hat{\bar{P}}_{x^k x^j}(\omega), \qquad (3.2)$$

where $\hat{P}_{x^k x^j}(\omega)$ is the cross power spectrum of x^k and x^j. Since for each k, $W_k(\omega)$ can be expressed as

$$W_k(\omega) = \sum_{j=0}^{L-1} w^k(j) e^{-ij\omega} = (w^k)^\mathsf{T} \xi(\omega),$$

where $\xi(\omega) = (1, e^{-i\omega}..., e^{-i(L-1)\omega})^\top$, it is clear that

$$W_k(\omega) \bar{W}_j(\omega) = (w^k)^\mathsf{T} \xi(\omega)(\bar{\xi}(\omega))^\mathsf{T}(w)^j, \qquad (3.3)$$

where $\xi(\omega)(\bar{\xi}(\omega))^\mathsf{T}$ can be computed directly as a Toeplitz matrix function:

$$\xi(\omega)(\bar{\xi}(\omega))^\mathsf{T} = \begin{pmatrix} 1 & e^{i\omega} & \ldots & e^{i(L-1)\omega} \\ e^{-i\omega} & 1 & \ldots & e^{i(L-2)\omega} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-i(L-1)\omega} & e^{-i(L-2)} & \ldots & 1 \end{pmatrix}. \quad (3.4)$$

Let $R_\xi(\omega)$ and $I_\xi(\omega)$ be, respectively, the real part and imaginary part of $\xi(\omega)(\bar{\xi}(\omega))^\mathsf{T}$ given by

$$R_\xi(\omega) = \begin{pmatrix} 1 & \cos\omega & \ldots & \cos(L-1)\omega \\ \cos\omega & 1 & \ldots & \cos(L-2)\omega \\ \vdots & \vdots & \ddots & \vdots \\ \cos(L-1)\omega & \cos(L-2)\omega & \ldots & 1 \end{pmatrix} \quad (3.5)$$

$$I_\xi(\omega) = \begin{pmatrix} 0 & \sin\omega & \ldots & \sin(L-1)\omega \\ -\sin\omega & 0 & \ldots & \sin(L-2)\omega \\ \vdots & \vdots & \ddots & \vdots \\ -\sin(L-1)\omega & -\sin(L-2)\omega & \ldots & 0 \end{pmatrix} \quad (3.6)$$

and let $R_{x^k x^j}(\omega)$ and $I_{x^k x^j}(\omega)$ be the real part and imaginary part of $\hat{\bar{P}}_{x^k x^j}(\omega)$, respectively. Then, (3.2) can be rewritten as

$$\hat{P}_{yy}(\omega) = \mathrm{Re} \left\{ \sum_{k=1}^{M} \sum_{j=1}^{M} W_k(\omega) \bar{W}_j(\omega) \hat{\bar{P}}_{x^k x^j}(\omega) \right\}$$

$$= \sum_{k=1}^{M} \sum_{j=1}^{M} \mathrm{Re} \left\{ (w^k)^\mathsf{T} \left[R_\xi(\omega) + i I_\xi(\omega) \right] w^j \left[R_{x^k x^j}(\omega) + i I_{x^k x^j}(\omega) \right] \right\} \quad (3.7)$$

$$= \sum_{k=1}^{M} \sum_{j=1}^{M} (w^k)^\mathsf{T} \left[R_{x^k x^j}(\omega) R_\xi(\omega) - I_{x^k x^j}(\omega) I_\xi(\omega) \right] w^j.$$

Let $\Xi(\omega)$ be a $\mathbb{R}^{ML \times ML}$ matrix function given by

$$\Xi_{xx}(\omega) = \begin{pmatrix} \Xi_{x^1 x^1}(\omega) & \Xi_{x^1 x^2}(\omega) & \ldots & \Xi_{x^1 x^M}(\omega) \\ \Xi_{x^2 x^1}(\omega) & \Xi_{x^2 x^2}(\omega) & \ldots & \Xi_{x^2 x^M}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ \Xi_{x^M x^1}(\omega) & \Xi_{x^M x^2}(\omega) & \ldots & \Xi_{x^M x^M}(\omega) \end{pmatrix}, \quad (3.8)$$

where

$$\Xi_{x^k x^j}(\omega) = R_{x^k x^j}(\omega) R_\xi(\omega) - I_{x^k x^j}(\omega) I_\xi(\omega). \qquad (3.9)$$

Then, by (3.7)-(3.9), the constant C_d defined in (2.7) can be simplified as

$$C_d = \frac{\varepsilon\left\{\hat{P}_{x_s^r x_s^r}\right\}}{\varepsilon\left\{w^\mathsf{T}\,\Xi_{x_s x_s}w\right\}}. \tag{3.10}$$

The normalized distortion measure defined in (2.6) can then be rewritten as

$$\begin{aligned}
D_r(w) &= \varepsilon\left\{\left\|\frac{\varepsilon\{\hat{P}_{x_s^r x_s^r}\}}{\varepsilon\{w^\mathsf{T}\Xi_{x_s x_s}w\}}\hat{P}_{y_s y_s}(\omega) - \hat{P}_{x_s^r x_s^r}(\omega)\right\|\right\} \\
&= \frac{\varepsilon\left\{\left\|w^\mathsf{T}\left(\varepsilon\{\hat{P}_{x_s^r x_s^r}\}\Xi_{x_s x_s} - \hat{P}_{x_s^r x_s^r}\varepsilon\{\Xi_{x_s x_s}\}\right)w\right\|\right\}}{\varepsilon\{w^\mathsf{T}\Xi_{x_s x_s}w\}},
\end{aligned} \tag{3.11}$$

and the normalized noise suppression measure and interference suppression measure defined in (2.8) and (2.9) can be, respectively, simplified as

$$S_N(w) = \frac{\varepsilon\left\{\hat{P}_{x_N^r x_N^r}\right\}}{\varepsilon\{\hat{P}_{x_s^r x_s^r}\}} \bullet \frac{w^\mathsf{T}\varepsilon\left\{\Xi_{x_s x_s}\right\}w}{w^\mathsf{T}\varepsilon\left\{\Xi_{x_N x_N}\right\}w}, \tag{3.12}$$

$$S_I(w) = \frac{\varepsilon\left\{\hat{P}_{x_I^r x_I^r}\right\}}{\varepsilon\left\{\hat{P}_{x_s^r x_s^r}\right\}} \bullet \frac{w^\mathsf{T}\varepsilon\left\{\Xi_{x_s x_s}\right\}w}{w^\mathsf{T}\varepsilon\left\{\Xi_{x_I x_I}\right\}w}. \tag{3.13}$$

With (2.6), (2.8), and (2.9) replaced by (3.11)-(3.13), the computation of distortion and suppression values has been greatly simplified.

3.2 Properties

From (3.11)-(3.13), we have the following lemma.

Lemma 1. *For any w and any $\lambda \neq 0$, it holds that*

$$D_r(\lambda w) = D_r(w),\ S_N(\lambda w) = S_N(w),\ S_I(\lambda w) = s_I(w).$$

By Lemma 1, we can restrict each coefficient $w^i(j)$ in a bounded open interval (-1,1). Let $\overline{\mathcal{W}} = (-1,1)^{ML}$, the respective infinite precision solutions of Problems 1 and 2 can be solved with the coefficients w confined in $\overline{\mathcal{W}}$.

For the suppression given by (3.12) (respectively, (3.13)), if the constraint of distortion level is ignored, we obtain the optimal solution of (3.12) (respectively, (3.13)). Denote

$$\tilde{w} = \left(\varepsilon\left\{\Xi_{x_N x_N}\right\}\right)^{1/2}w,\quad \tilde{R} = \left(\varepsilon\left\{\Xi_{x_N x_N}\right\}\right)^{-1/2}\varepsilon\left\{\Xi_{x_s x_s}\right\}\left(\varepsilon\left\{\Xi_{x_N x_N}\right\}\right)^{-1/2}.$$

Then, maximizing (3.12) is equivalent to maximizing

$$\tilde{S}_N(\tilde{w}) = \frac{\tilde{w}^\mathsf{T}\tilde{R}\tilde{w}}{\|\tilde{w}\|^2}. \tag{3.14}$$

Considering (3.14), the optimal value is given by the maximal eigenvalue of $\tilde{R}$ and the optimal solution $\tilde{w}^*$ is given by the respective eigenvector. Then, the optimal solution w^* maximizing (3.12) is given by

$$w^* = \left(\varepsilon\left\{\Xi_{x_N x_N}\right\}\right)^{-1/2}\tilde{w}^*.$$

We obtain the optimal solution w^* by maximizing (3.13) using the same principle.

The continuous versions of the optimization problems of Problems 1 and 2 are nonlinear programming problems. One well-known method for solving this type of problems is the sequential quadratic programming (SQP) which will be employed here. An overview of SQP can be found in [24].

4 Finite precision solution

For the coefficients of beamformers, it is advantageous to express each one as the sum of signed power-of-two terms. In general, we hope to use less wordlength and less SPT terms to obtain a satisfying performance when compared with the infinite precision solution. However, if the wordlength is not long enough, the truncated finite precision solution will produce very poor performance. As a result, we need to develop a method to find a satisfying finite precision solution by using a relatively small number of SPT terms.

4.1 Problem transformation

Let us first construct a transformation to convert Problem 1 to an equivalent integer programming problem.

Ignoring the constraints (2.4) and (2.5), it is easy to see that the set of all $w^i(j)$ is $\{2^{-b}k,\ k = 1-2^b,\ldots,2^b-1\}$. That is, for any $\tilde{w} \in \left\{2^{-b}k, k = 1 - 2^b,\ldots,2^b - 1\right\}$, there exists a vector $\bar{s} = (\bar{s}_k)$, $\bar{s}_k \in \{-1, 0, 1\}$, such that

$$\bar{w} = \sum_{k=1}^{b}\bar{s}_k 2^{-k}. \tag{4.1}$$

Then, when the wordlength is taken as b-bit, the minimal number of SPT terms for $\bar{w}$ is defined as:

$$P_w(\bar{w}, b) = \min_{\bar{s}}\sum_{k=1}^{b}|\bar{s}_k| \tag{4.2}$$

$$s.t.\qquad \bar{w} = \sum_{k=1}^{b}\bar{s}_k 2^{-k}.$$

We now introduce an integer vector $z = ((z^1)^\mathsf{T}\ \ldots,\ (z^M)^\mathsf{T})^\mathsf{T}$, where

$$z^i = \left(z^i(0), z^i(1),\ldots,z^i(L-1)\right)^\mathsf{T},\quad i = 1,\ldots,M,$$

such that

$$z^i(j) = 2^b w^i(j),\quad i = 1,\ldots,M, j = 0,\ldots,L-1.$$

Then, for each i and j, $z^i(j) \in \{1 - 2^b,\ldots,2^b - 1\}$. Thus, when the wordlength is taken as b-bit, the minimal

number of SPT terms for some integer $\bar{z}$ can be defined by

$$P_z(\bar{z}, b) = P_w(2^{-b}\bar{z}, b). \tag{4.3}$$

Remark 2. *The computation of $P_z(\bar{z}, b)$ and the conversion from an integer to signed digit code can be implemented by recursive functions. These are given in the Appendix.*

Clearly, the constraints (2.4) and (2.5) are equivalent to

$$\sum_{i=1}^{M} \sum_{j=0}^{L-1} P_z(z^i(j), b) \leq N_1 \tag{4.4}$$

$$P_z(z^i(j), b) \leq N_2, \quad \forall i, j. \tag{4.5}$$

The cost functions (2.10) and (2.12) are equivalent to

$$\bar{F}_1(z) = -10 * (\log_{10} S_N(2^{-b}z) + \log_{10} S_I(2^{-b}z)), \tag{4.6}$$

and

$$\bar{F}_2(z) = 10 * \log_{10} D(2^{-b}z). \tag{4.7}$$

To deal with the constraints, we use the penalty function method. Let Q_1, Q_2, Q_3, Q_4 and Q_5 denote five sufficiently large positive real numbers. The constraints (4.4), (4.5), (2.11), and (2.13) are replaced with

$$g_1(z) = \max \left\{ 0, Q_1 \left(\sum_{i=1}^{M} \sum_{j=0}^{L-1} P_z(z^i(j), b) - N_1 \right) \right\} \tag{4.8}$$

$$g_{2,i,j}(z) = \max\{0, Q_2(P_z(z^i(j), b) - N_2)\}, \quad i = 1, \ldots, M, j = 0, \ldots, L-1, \tag{4.9}$$

$$g_3(z) = \begin{cases} Q_3 G_1(2^{-b}z) + Q_5 & \text{if } G_1(2^{-b}z) > 0 \\ 0 & \text{if } G_1(2^{-b}z) \leq 0' \end{cases} \tag{4.10}$$

and

$$g_4(z) = \begin{cases} Q_4 G_2(2^{-b}z) + Q_5 & \text{if } G_2(2^{-b}z) > 0 \\ 0 & \text{if } G_2(2^{-b}z) \leq 0 \end{cases} \tag{4.11}$$

respectively. Then, by adding the penalized terms, the objective functions (4.6) and (4.7) become

$$E_1(z) = \bar{F}_1(z) + g_1(z) + \sum_{i=1}^{M} \sum_{j=0}^{L-1} g_{2,i,j}(z) + g_3(z), \tag{4.12}$$

and

$$E_2(z) = \bar{F}_2(z) + g_1(z) + \sum_{i=1}^{M} \sum_{j=0}^{L-1} g_{2,i,j}(z) + g_4(z), \tag{4.13}$$

respectively. Let $\mathcal{Z} = \{1 - 2^b, \ldots, 2^b - 1\}^{ML}$ denote the space of all possible z. Thus, Problem 1 is transformed into:

Problem 3. *Find a $z \in \mathcal{Z}$ such that $E_1(z)$, which is defined by (4.12), is minimized.*

Similarly, Problem 2 is transformed into:

Problem 4. *Find a $z \in \mathcal{Z}$ such that $E_2(z)$, which is defined by (4.13), is minimized.*

Problems 3 and 4 are unconstrained integer programming problems. To solve Problems 3 and 4, we will develop a three-step algorithm. The first step is the selection of an initial point, the second step is a local search, while the third step is a global approach.

4.2 Initial point

We will develop an efficient computational method to search for the optimal solutions of Problems 3 and 4. But first, we need to find a good initial point.

In general, it is much more difficult to solve the discrete optimization problem than to solve the continuous version of the optimization problem. In order to select good initial points for Problems 3 and 4, we first find the solutions (infinite precision solutions) of the continuous version of Problems 1 and 2 and then round them to the nearest discrete solutions.

Then, after the optimal solution $\hat{w}$ of the continuous version of Problem 1 or 2 is obtained, we denote $\hat{z} = 2^b \hat{w}$ and select the initial point z_0 as the nearest point in $\mathcal{Z}$ to $\hat{z}$, that is,

$$z_0^i(j) = T(\hat{z}^i(j)), \quad i = 1, \ldots, M, j = 0, \ldots, L-1, \tag{4.14}$$

where T is a function defined by

$$T(\bar{z}) = \begin{cases} [\bar{z} + 0.5], & \text{if } |\bar{z}| \leq 2^b - 1, \\ 2^b - 1, & \text{if } \bar{z} > 2^b - 1, \\ 1 - 2^b, & \text{if } \bar{z} < 1 - 2^b, \end{cases} \tag{4.15}$$

in which $[\bar{z}]$ denotes the largest integer less than or equal to $\bar{z}$.

4.3 Local search

With an initial point, we can start to search for a local minimizer. The steepest descent algorithm is applied here to find a local minimizer by selecting the point which produces the largest reduction in the value of the objective function over the current point's neighborhood. The definition of neighborhood is given in

Definition 1. *For any $z \in \mathcal{Z}$, the neighborhood of z is defined by*

$$\mathcal{N}(z) = \{z, z \pm e_i : i = 1, \ldots, ML\} \cap \mathcal{Z}, \tag{4.16}$$

where e_i is the ith unit vector (the $\mathbb{R}^{ML}$ vector with the ith component equal to one and all other components equal to zero).

If we have found a point which minimizes the objective function over its neighborhood, then the local search stops and the point obtained is called a local

minimizer. The precise definition of local minimizers is given as follows:

Definition 2. *A point z^* is called a local minimizer of E_i over $\mathcal{Z}$ if $E_i(z^*) \leq E_i(z)$, $\forall z \in \mathcal{N}(z^*)$.*

Based on the two definitions given above, we present a discrete steepest descent algorithm, which will be used as a subroutine, to search for a local minimizer of the cost function E_i in the following algorithm.

Algorithm 1. (Subroutine)

1. Start from an initial minimum z^* and an initial point $z' = z_0$. Compute the objective function value E_i.
2. For each point $z \in \mathcal{N}(z')\backslash\{z'\}$, compute the corresponding objective function value $E_i(z)$. Suppose z' is such that $E_i(z')$ is the minimum. If $E_i(z') \geq E_i(z^*)$, then z^* is a local minimizer of E_i and stop, else goto Step 3.
3. Set $z^* = z'$. Goto Step 2.

4.4 Global approach

With Algorithm 1, we can find a local minimizer from any initial point. For this problem, there exist many local minimizers, and not all of them are useful in practice. Thus, we shall derive a discrete filled function method to search for the best local minimizer.

We introduce the following function based on the one constructed in [23].

$$F_{\mu,\rho}(z;z^*) = \mu[E_i(z) - E_i(z^*)]^2 - \rho\|z - z^*\|^2, \quad \text{if } E_i(z) \geq E_i(z^*), \quad (4.17)$$

where $\|\cdot\|$ denotes the usual Euclidean norm. When $\rho > 0$, $0 < \mu < \rho/K$ (K is a sufficiently large real number), (4.17) is called a discrete filled function.

It is not necessary to define the function $F_{\mu,\rho}$ when $E_i(z) < E_i(z^*)$. In this case, we can use the discrete steepest descent method directly with z as the initial point and will obtain a local minimizer, which is better than z^* with reference to the objective function.

The search according to this discrete filled function (4.17) takes place as follows. With the starting point z^*, the $\mu[E_i(z) - E_i(z^*)]^2$ term favors a solution with lower objective function value while the $-\rho\|z - z^*\|^2$ term favors a solution far away from z^*. Combining the effects, the discrete filled function favors a solution whose objective function value is not too much greater than that of z^* and at a considerable distance away from z^*. The idea is to direct its search towards the direction with the least increase in the objective function value.

To address the situation when we fail to find a point z such that $E_i(z) < E_i(z^*)$ using the discrete filled function (4.17) as the objective function, we choose a positive integer number n_s. When the number of searching steps is greater than n_s, we stop and return the current local minimizer. This is because for the problem considered in this article, the optimal solution should be in the neighborhood of the continuous solution. Therefore, when the number of searching steps is greater than n_s, we can consider that the best local minimizer has been obtained and it is not necessary to continue searching.

Then, with the current local minimizer z^* of E_i, we present a discrete steepest descent algorithm to search for a point better than z^* in the following algorithm.

Algorithm 2. (Subroutine)

1. Set the current local minimizer z^* as the initial point z_0. Set $l = 0$ be the number of searching steps.
2. For each point $z \in \mathcal{N}(z_l)\backslash\{z_l\}$, compute the corresponding objective function value $F_{\mu,\rho}(z; z^*)$. Choose z' such that $F_{\mu,\rho}(z'; z^*)$ is the minimum. Go to Step 3.
3. If $E_i(z') < E_i(z^*)$, then stop and return z' which is better than z^*, else set $l = l + 1$ and goto Step 4.
4. If $l < n_s$, go to Step 2, else stop and return $z' = z^*$, which means there is no point better than z^*.

4.5 Algorithm

The proposed algorithm solving Problem 3 (or Problem 4) is summarized in the following:

Algorithm 3. (Main program)

1. Calculate the infinite precision solution $\hat{w}$ of Problem 1 (or Problem 2) by using the SQP method. Compute the initial point z_0.
2. Apply Algorithm 1 to obtain a local minimizer of the objective function E_i with the initial point z_0. Let the local minimizer be denoted as z^*.
3. Apply Algorithm 2 to find a point z' better than z^*. If $z' \neq z^*$, set $z_0 = z'$ and go to Step 2, else stop search and obtain the solution z^*, go to Step 4.
4. If the best value $E_i(z^*)$ is sufficiently large, it means some constraints are not satisfied and the solution is infeasible, else return the solution z^* and its value $E_i(z^*)$. Stop.

It can be seen that Algorithm 3 includes two searches: the search of local minimizer and the search for switching to the better local minimizer. Both searches are based on the discrete steepest descent algorithm.

5 Simulation results

The proposed method has been used to solve several examples. The results obtained are consistently favorable when compared with results obtained by the truncation method. In this section, we illustrate the results for two examples. The computation was performed in Matlab, where the coefficients in this article are set as

$\mu = 10^{-6}$, $\rho = 1$, $n_s = 10$, $Q_1 = Q_2 = Q_5 = 100$, $Q_3 = Q_4 = 10$.

In the first example, the measurements were performed in a car environment. The calibration signals were recorded with a sample rate of 12 kHz and with a 300-3400 Hz bandwidth. A linear microphone array with four elements is considered. Noise calibration signals were used, which are emitted individually from the artificial talker and the handsfree loudspeaker as the source and the interference calibration signals, respectively. Interference signals were recorded by emitting an independent sequence of white noise, from the handsfree loudspeaker alone, within the bandwidth. This recording serves as the point source interference calibration signal. Recordings with real speech signals were recorded both individually and while driving. In order to gather background noise signals, the car was driving at a speed of 110 km/h on a paved road. The duration of these signals was 8 s [3].

We consider Problem 1 with the filter length given as 8. The total allowable number of SPT terms is constrained by $N_1 = 40$ and the allowable number for each coefficient is constrained by $N_2 = 3$. To see how our method studies, we take an example when the distortion level is taken as $c_1 = -25$ dB and the wordlength is taken as $b = 5$. We use the function fmincon in Matlab to solve the infinite precision solution, whose objective value is -23.8079 dB. The truncation solution is obtained, where the objective value is -20.7160 and the number of SPT terms is 60. For the proposed method, we first obtain a local minimizer by Algorithm 1, where the objective value is -21.9350 dB. By the discrete filled function, we apply Algorithm 1 to search for a solution better than the current local minimizer. Then, setting this solution as the initial point, we apply Algorithm 1 to obtain a better local minimizer whose objective value is -22.1598 dB. Again, by applying the discrete filled function four times, we obtain the local minimizer whose objective value is -23.4359 dB. This is then the best solution we found. Hence, we can see that from the truncation method to the proposed method, the total improvement of performance is 2.72 dB and the improvement of the number of SPT terms is 20. Furthermore, the improvement of performance from local search to global search is 1.50 dB.

Next, by choosing $N_1 = 50$, $N_2 = 3$, and $b = 6$, we apply the proposed method to solve the infinite precision solution with the distortion level taken as $c_1 = -20$ dB, -22 dB, and -25 dB, respectively. The infinite and finite precision solutions and their values are reported in Table 1.

The power spectrum estimates of the unprocessed and processed noise and interference signal using Welch's periodogram in the case of $c_1 = -20$ dB are depicted in

Table 1 Results for Example 1

Solutions	Distortion level	Sum of SPT	Constraint G_1	$F_1(w^*)$
Infinite	-20 dB	–	0	-31.9979
Precision	-22 dB	–	0	-26.7198
Solutions	-25 dB	–	0	-23.8079
	-20 dB	75	0.0052	-31.9196
Truncated	-22 dB	79	0.0254	-26.5992
Solutions	-25 dB	77	-0.1208	-21.9489
	-20 dB	50	-0.0022	-31.8630
Proposed	-22 dB	50	-0.0260	-26.6194
Method	-25 dB	50	-0.0206	-23.6929

Figures 2 and 3. In the suppression figure, if the line of beamformer output is lower than that of the single sensor observation, then the noise signal or interference signal is suppressed. The much the dashed line is lower than the real line, the much the signal is suppressed, especially at some frequency when the intensity is strong (dB value is high). It can be seen from Figures 2 and 3 that the noise or interference signal has been suppressed.

In the second example, the calibration signals were recorded with a two-element microphone arrays in an anechoic environment. A sample rate of 6 kHz is used. The duration of the signals was 5 s. We consider Problem 2 with the filter length given as 20. The total allowable number of SPT terms is constrained by $N_1 = 50$ and the allowable number for each coefficient is constrained by $N_2 = 3$. The wordlength is taken as $b = 6$. We use the function fmincon in Matlab to solve the infinite precision solution with the suppression level taken as $c_2 = -20$ dB, -22 dB, and -25 dB, respectively. The infinite and finite precision solutions and their values are reported in Table 2. The power spectrum estimates of the unprocessed and processed noise and interference signal using Welch's periodogram in the case of $c_2 = -20$ dB are shown in Figure 4. As the dashed line is lower than the real line for all frequencies in Figure 4, the noise signal has been suppressed remarkably.

We can see from Tables 1 and 2 that the sums of SPT terms of the directly truncated solutions are always very large, and neither their values are satisfactory when compared with the infinite precision solutions, nor are the constraints. For the solutions obtained by the discrete filled function method, the sums of SPT terms are reduced remarkably and their values are very close to that of the infinite precision solutions.

6 Conclusion

In this article, a novel design method is developed for the finite precision beamformers. The design is based on a bi-criteria formulation trading off speech distortion against noise suppression, and the method is based on
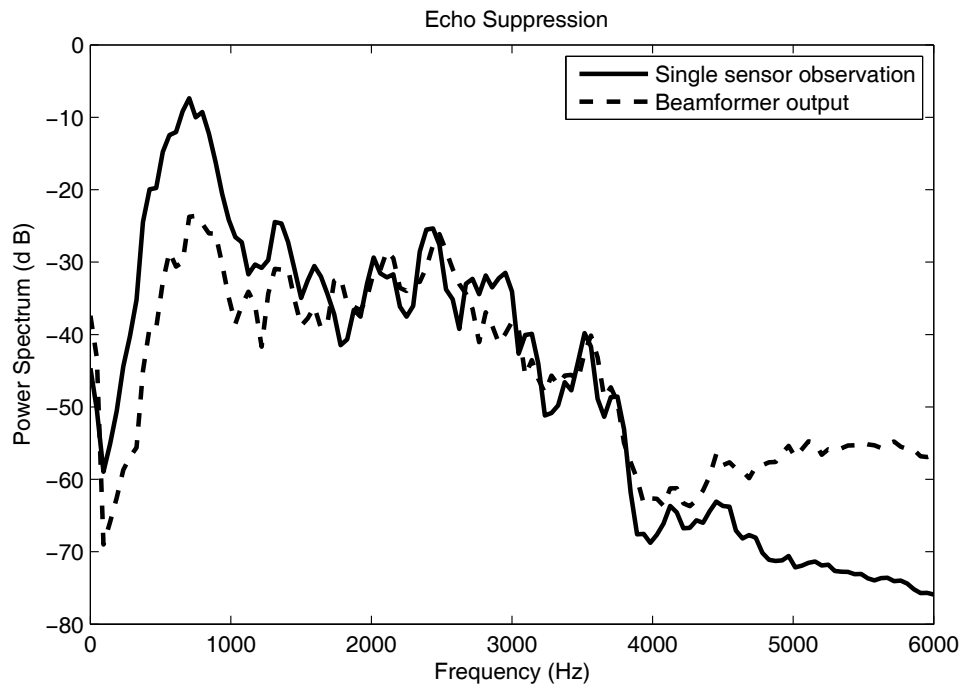
Figure 2 Noise suppression in Example 1. Spectrum estimates using Welch's pe-riodogram of an unprocessed single microphone observation and the beamformer output signal, when the distortion level is -20 dB.
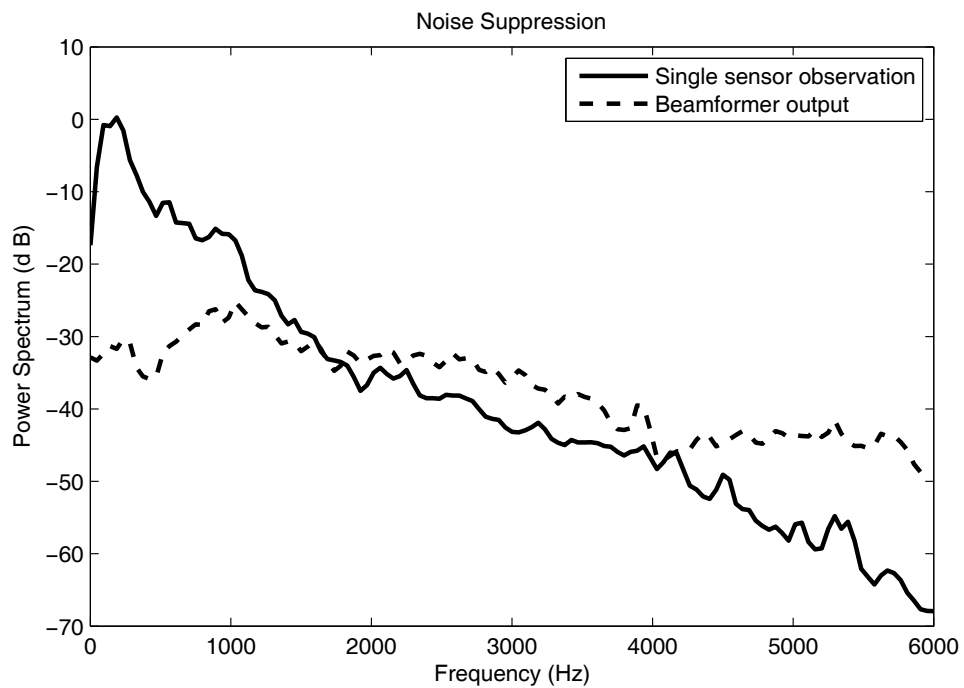

Figure 3 Interference suppression in Example 1. Spectrum estimates using Welch's periodogram of an unprocessed single microphone observation and the beamformer output signal, when the distortion level is -20 dB.

Table 2 Results for Example 2

Solutions	Suppression level	Sum of SPT	Constraint G_2	$F_2(w^*)$
Infinite	-20 dB	–	0	-26.7387
Precision	-22 dB	–	0	-26.2833
Solutions	-25 dB	–	0	-25.3942
	-20 dB	67	0.3268	-26.7499
Truncated	-22 dB	93	0.0981	-26.2938
Solutions	-25 dB	73	0.3541	-25.4031
	-20 dB	50	-0.0047	-26.7228
Proposed	-22 dB	50	-0.0188	-26.2194
Method	-25 dB	50	-0.0057	-25.3521

the concept of discrete filled functions, where filter coefficients are expressed as sums of SPT terms. An algorithm is derived using gradient descent techniques together with a procedure. From our numerical studies, the proposed method is highly effective, producing superior performance and reasonable design complexity.

Appendix 1: computation of $P_z(\bar{z}, b)$

When the wordlength is taken as b-bit, the minimal number of SPT terms for $\bar{z}$, which is defined as $P_z(\bar{z}, b)$ in (4.3), is computed as a recursive function:

Recursive Function $(P_z(\bar{z}, b))$.

1. If $|\bar{z}| \geq 2^b$, then $P_z(\bar{z}, b) = +\infty$. Stop and return $P_z(\bar{z}, b)$.
2. If $\bar{z} = \pm 1$, then $P_z(\bar{z}, b) = 1$. Stop and return $P_z(\bar{z}, b)$.
3. If $|\bar{z}| > 1$ and $\bar{z}$ is odd, then call the functions $P_z(\bar{z} + 1, b)$ and $P_z(\bar{z} + 1, b)$ respectively. Then, let $P_z(\bar{z}, b) = 1 + \min\{P_z(\bar{z} - 1, b), P_z(\bar{z} + 1, b)\}$. Stop and return $P_z(\bar{z}, b)$.
4. If $\bar{z}$ is even, then call the function $P_z(\bar{z}/2, b - 1)$. Then, $P_z(\bar{z}, b) = P_z(\bar{z}/2, b - 1)$. Stop and return $P_z(\bar{z}, b)$.

For this recursive function, the first two terms are the terminal conditions and the last two terms are the recursive rules. It is not difficult to see that the recursive rules preserve the property that $P_z(\bar{z}, b)$ is the minimal number of SPT terms for $\bar{z}$. For example, in the third term, if $P_z(\bar{z} - 1, b)$ is the minimal number of SPT terms for $\bar{z} - 1$ and $P_z(\bar{z} + 1, b)$ is the minimal number of SPT terms for $\bar{z} + 1$, then $P_z(\bar{z}, b)$ is the minimal
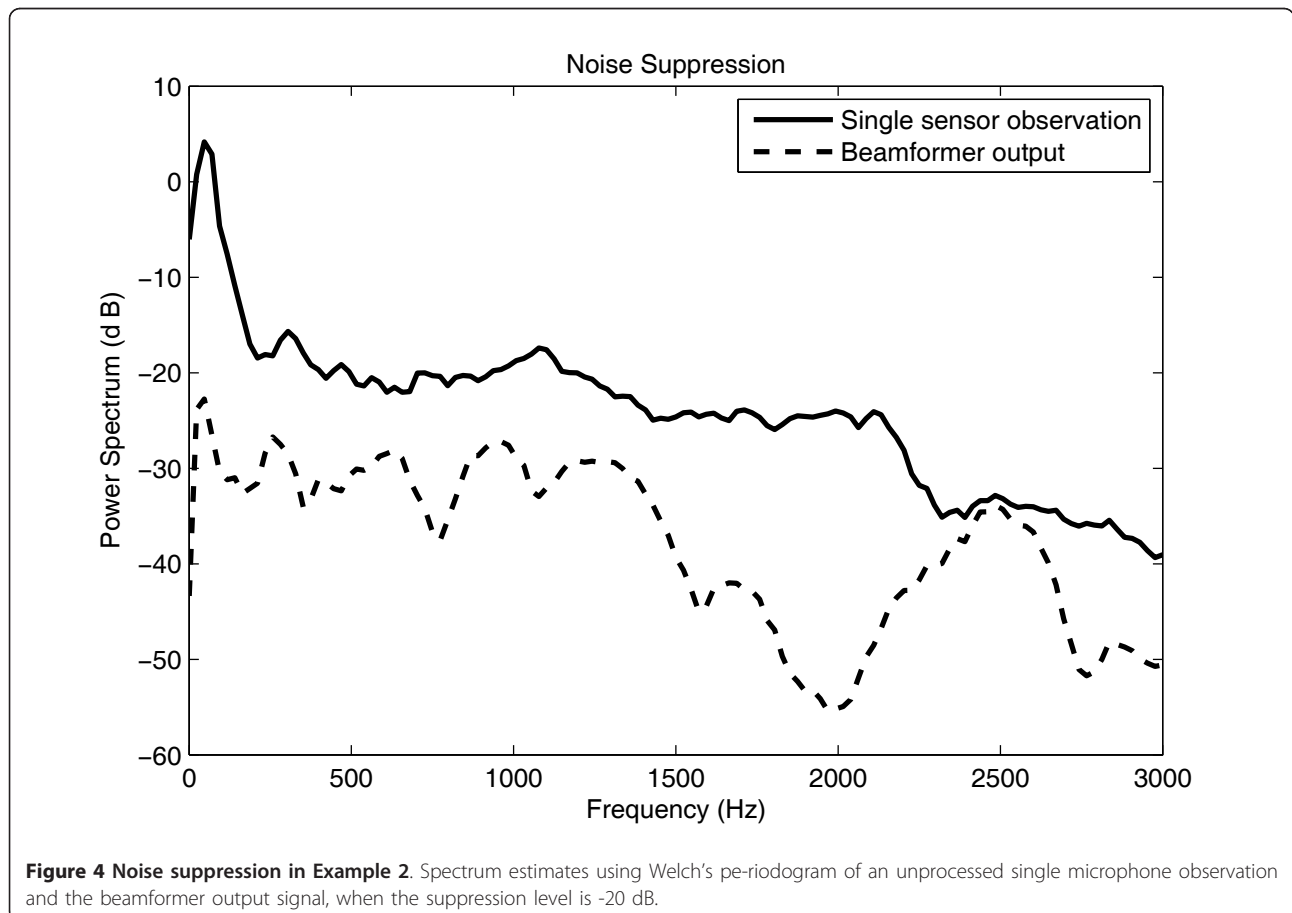

Figure 4 Noise suppression in Example 2. Spectrum estimates using Welch's pe-riodogram of an unprocessed single microphone observation and the beamformer output signal, when the suppression level is -20 dB.

number of SPT terms for $\bar{z}$. Hence, we can compute $P_z(\bar{z}, b)$ by applying this recursive function.

Appendix 2: converting an integer into signed digit code

Let $\Lambda(\bar{z}, b)$ denotes the minimal number of signed powers of two terms of $\bar{z}$ when the wordlength is taken as b-bit. Generally, $\Lambda(\bar{z}, b)$ is not unique.

Recursive Function ($\Lambda(\bar{z}, b)$).

> *1. If $|\bar{z}| \geq 2^b$, then $\Lambda(\bar{z}, b) = \emptyset$. Stop.*
>
> *2. If $\bar{z} = 1$, then $\Lambda(\bar{z}, b) = \{2^0\}$. If $\bar{z} = -1$, then $\Lambda(\bar{z}, b) = \{-2^0\}$. Stop and return $\Lambda(\bar{z}, b)$.*
>
> *3. If $|\bar{z}| > 1$ and $\bar{z}$ is odd, then call the functions $\Lambda(\bar{z} - 1, b)$, $\Lambda(\bar{z} + 1, b)$ and $\Lambda(\bar{z} - 1, b)$, $\Lambda(\bar{z} + 1, b)$. if $P_z(\bar{z}, b) = 1 + P_z(\bar{z} + 1, b)$ then let $\Lambda(\bar{z}, b) = \Lambda(\bar{z} + 1, b) \cup \{-2^0\}$, else if $P_z(\bar{z}, b) = 1 + P_z(\bar{z} - 1, b)$, then let $\Lambda(\bar{z}, b) = \Lambda(\bar{z} - 1, b) \cup \{2^0\}$. Stop and return $\Lambda(\bar{z}, b)$.*
>
> *4. If $\bar{z}$ is even, then call the function $\Lambda(\bar{z}/2, b-1)$. Suppose that $\Lambda(\bar{z}/2, b-1) = \{s_{i_k} 2^{i_k}, k = 1, \ldots\}$, then let $\Lambda(\bar{z}, b) = \{s_{i_k} 2^{i_k+1}, k = 1, \ldots\}$. Stop and return $\Lambda(\bar{z}, b)$.*

Author details
[1]Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong, PR China [2]Department of Mathematics and Statistics, Curtin University, Perth, WA, Australia [3]School of Electrical and Computer Engineering, Curtin University, Perth, Australia

Competing interests
The authors declare that they have no competing interests.

References
1. S Nordholm, I Claesson, M Dahl, Adaptive microphone array employing calibration signals: an analytical evaluation. IEEE Trans Speech Audio Process. **7**(3), 241–252 (1999). doi:10.1109/89.759030
2. M Dahl, I Claesson, Acoustic noise and echo canceling with microphone array. IEEE Trans Veh Technol. **48**(5), 1518–1526 (1999). doi:10.1109/25.790527
3. S Nordholm, I Claesson, N Grbić, M Brandstein, D Ward, Optimal and adaptive microphone arrays for speech input in automobiles, Microphone Arrays, (Springer Verlag, Berlin Heidelberg New York, 2001), pp. 307–330
4. KFC Yiu, N Grbić, KL Teo, S Nordholm, A new design method for broadband microphone arrays for speech input in automobiles. IEEE Signal Process Lett. **9**(7), 222–224 (2002)
5. S Weiss, IK Proudler, in *Comparing efficient broadband beamforming architectures and their performance trade-offs*, vol. 1. (International Conference on Digital Signal Processing, Santorini, Greece, 2002), pp. 417–423
6. Y Zhang, L Milstein, P Siegel, in *The tradeoff between diversity gain and interference suppression via beamforming in a CDMA system*, vol. II.

7. (Proceedings of the 2003 IEEE conference on Military communications, Boston, USA, 2003), pp. 1132–1137
7. M Lanne, A Lundgren, M Viberg, in *Optimized beamforming calibration in the presence of array imperfections*, vol. II. (Proceedings of the ICASSP, Honolulu, Hawaii, USA, 2007), pp. 973–976
8. D Guenther, W Walker, Robust finite impulse response beamforming applied to medical ultrasound. IEEE Trans Ultrason Ferroelectron Freq Control. **56**, 1168–1188 (2009)
9. YP Liu, Q Wan, Total difference based partial search LCMV beamformer. Progress Electromag Res Lett. **18**, 97–103 (2010)
10. N Sharma, J Buck, J Simmons, Trading detection for resolution in active sonar receivers. J Acoustical Soc Am. **130**, 1272–1281 (2011). doi:10.1121/1.3614547
11. SV Gerven, DV Compernolle, P Wauters, W Verstraeten, K Eneman, K Delaet, Multiple beam broadband beamforming: filter design and real-time implementation, (IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, New York, 1995), pp. 173–176
12. S Srinivasan, A Pandharipande, K Janse, Beamforming under quantization errors in wireless binaural hearing aids. EURASIP J Audio Speech Music Process Article ID 824797 (2008)
13. KFC Yiu, CH Ho, N Grbric, Y Lu, XX Shi, W Luk, Reconfigurable acceleration of microphone array algorithms for speech enhancement, in *19th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP 2008)*. Leuven, Belgium, pp. 203–208 (2008)
14. YC Lim, SR Parker, FIR filter design over a discrete powers-of-two coefficient space. IEEE Trans Acoust Speech Signal Process. **31**, 583–591 (1983). doi:10.1109/TASSP.1983.1164085
15. N Benvenuto, M Marchesi, A Uncini, Applications of simulated annealing for the design of special digital filters. IEEE Trans Signal Process. **40**, 323–332 (1992). doi:10.1109/78.124942
16. DN Li, YC Lim, Y Lian, JJ Song, A polynomial-time algorithm for designing FIR filters with power-of-two coefficients. IEEE Trans Signal Process. **50**, 1935–1941 (2002). doi:10.1109/TSP.2002.800385
17. ZG Feng, KL Teo, A discrete filled function method for the design of FIR filters with signed-powers-of-two coefficients. IEEE Trans Signal Process. **56**, 134–139 (2008)
18. YJ Yu, YC Lim, A novel genetic algorithm for the design of a signed power-of-two coefficient quadrature mirror filter lattice filter bank. Circ Syst Signal Process. **21**(3), 263–276 (2002). doi:10.1007/s00034-004-7043-z
19. YC Lim, YJ Yu, A width-recursive depth-first tree search approach for the design of discrete coefficient perfect reconstruction lattice filter bank. IEEE Trans Circ Syst II. **50**(6), 257–266 (2003). doi:10.1109/TCSII.2003.812912
20. YC Lim, R Yang, DN Li, JJ Song, Signed power-of-two term allocation scheme for the design of digital filters. IEEE Trans Circ Syst II. **46**(5), 577–584 (1999). doi:10.1109/82.769806
21. RP Ge, A filled function method for finding a global minimizer of a function of several variables. Math Program. **46**, 191–204 (1990). doi:10.1007/BF01585737
22. KFC Yiu, YQ Liu, KL Teo, A hybrid descent method for Global optimization. J global Optim. **28**(2), 229–238 (2004)
23. CK Ng, LS Zhang, D Li, WW Tian, Discrete filled function method for discrete global optimization. Comput Optim Appl. **31**, 87–115 (2005). doi:10.1007/s10589-005-0985-7
24. R Fletcher, Practical Methods of Optimization, (John Wiley and Sons, Great Britain, 1987)