ROBUST ADAPTIVE BEAMFORMING USING SEQUENTIAL QUADRATIC PROGRAMMING

Aboulnasr Hassanien *Sergiy A. Vorobyov*

University of Alberta
Electrical and Computer Engineering
Edmonton, AB, T6G 2V4 Canada

Kon Max Wong

McMaster University
Electrical and Computer Engineering
Hamilton, ON, L8S4K1 Canada

ABSTRACT

In this paper, a new algorithm for robust adaptive beamforming is developed. The basic idea of the proposed algorithm is to estimate the difference between the actual and presumed steering vectors and to use this difference to correct the erroneous presumed steering vector. The estimation process is performed iteratively where a quadratic convex optimization problem is solved at each iteration. Unlike other robust beamforming techniques, our algorithm does not assume that the norm of the mismatch vector is upper bounded, and hence it does not suffer from the negative effects of over/under estimation of the upper bound. Simulation results show the effectiveness of the proposed algorithm.

Index Terms— Array signal processing, adaptive arrays, parameter estimation, robustness, mathematical programming.

1. INTRODUCTION

Adaptive beamforming plays an important role in many applications such as radar, sonar, speech processing, seismology, and wireless communications. Conventional adaptive beamforming techniques assume that the steering vector associated with the signal of interest (SOI) is precisely known. However, in practice the information may not be perfectly known resulting in a mismatch between the presumed and the actual steering vectors. Such a mismatch arises due to imprecisely known wavefield propagation conditions, imperfectly calibrated arrays, array perturbations, and/or signal pointing errors [1], [2].

In order to provide robustness against such mismatches, several techniques have been recently proposed in the literature [2]–[5]. These techniques assume that the norm of the mismatch vector is upper-bounded. In practice, neither the mismatch vector nor its upper bound is known. If the upper bound is overly estimated, then the aforementioned robust beamforiming techniques become too conservative. On the other hand, under estimation of the upper bound may result in self-nulling of the SOI. In both cases, these techniques may suffer from performance degradation.

In this paper, we propose a new algorithm for robust adaptive beamforming which is based on estimating the difference between the actual and presumed steering vectors. The estimation process is performed iteratively. A quadratic convex optimization problem is solved at each iteration. Then, the presumed steering vector is updated and used to obtain the beamformer weights using any adaptive beamforming technique. In our approach we do not assume that the mismatch vector is upper bounded and, hence, we avoid the need for estimating the upper bound. Our algorithm shows improvements in performance compared to existing adaptive beamforming techniques.

2. ARRAY SIGNAL MODEL

The output of a narrowband adaptive beamformer is given by

$$y(t) = \mathbf{w}^H \mathbf{x}(t) \tag{1}$$

where t is the time index, $\mathbf{x}(t) = [x_1(t), \dots, x_M(t)]^T \in \mathcal{C}^M$ is the array observation vector, $\mathbf{w} = [w_1, \dots, w_M]^T \in \mathcal{C}^M$ is the complex vector of beamforming weights, and $(\cdot)^T$ and $(\cdot)^H$ stand for the transpose and Hermitian transpose, respectively. The complex vector of array observation can be written as

$$\mathbf{x}(t) = \mathbf{x}_s(t) + \mathbf{x}_i(t) + \mathbf{n}(t) \tag{2}$$

where $\mathbf{x}_s(t)$, $\mathbf{x}_i(t)$, and $\mathbf{n}(t)$ are the statistically independent components of the desired signal, interference, and sensor noise, respectively. We consider the case of narrowband desired signal that can be written as

$$\mathbf{x}_s(t) = s(t)\mathbf{a} \tag{3}$$

Supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada and the Alberta Ingenuity Foundation, Alberta, Canada.

S.A. Vorobyov is on leave from the Joint Research Institute, Heriot-Watt University and the University of Edinburgh, Edinburgh, Scotland, UK.

where $s(t)$ is the desired signal waveform and $\mathbf{a}$ is the actual steering vector (actual spatial signature) associated with the desired signal.

3. BACKGROUND

In the absence of steering vector mismatch, the optimal solution for the adaptive beamforming problem is given by solving the following optimization problem [6]

$$\min_{\mathbf{w}} \ \mathbf{w}^H \mathbf{R}_{\mathbf{x}} \mathbf{w} \quad \text{subject to} \ \mathbf{w}^H \mathbf{a} = 1. \tag{4}$$

The solution to the above optimization problem is given by the well-known Capon beamformer [6]

$$\mathbf{w}_{\text{opt}} = \frac{\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}}{\mathbf{a}^H \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}}. \tag{5}$$

However, in the presence of any mismatch the above beamformer is no longer optimal and it might lead to a catastrophic self-nulling of the desired signal. An elegant approach for robust adaptive beamforming using worst-case performance optimization was introduced in [2]. The basic idea of this approach is to impose a constraint that the absolute value of the array response is greater or equal to unity for all vectors that belong to the neighborhood of the presumed vector. This was formulated in the following optimization problem

$$\min_{\mathbf{w}} \ \mathbf{w}^H \mathbf{R}_{\mathbf{x}} \mathbf{w} \quad \text{subject to} \min_{\|\mathbf{e}\| \le \epsilon} |\mathbf{w}^H(\tilde{\mathbf{a}} + \mathbf{e})| \ge 1 \tag{6}$$

where $\tilde{\mathbf{a}}$ is the presumed steering vector and $\mathbf{e}$ is the mismatch vector. The above optimization problem is hard to solve because it involves an infinite number of nonconvex constraints. However, it can be transformed into the following convex second-order cone programming problem [2]

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^H \mathbf{R}_{\mathbf{x}} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^H \tilde{\mathbf{a}} \ge \epsilon \|\mathbf{w}\| + 1. \end{aligned} \tag{7}$$

The above optimization problem can be solved using interior point methods with a complexity of $O(M^{3.5})$. However, it is assumed in (7) that the value of ϵ is known. Over or under estimation of ϵ leads to performance degradation. Hence, more work on this problem is required.

4. THE PROPOSED ALGORITHM

In this section, we propose a new robust adaptive beamforming algorithm which makes use of sequential quadratic programming. Our algorithm is based on estimating the mismatch vector and it allows forming the beam using the corrected steering vector.

Recall that the power spectrum of the Capon beamformer (as a function of the array response vector) is given by [6]

$$P(\mathbf{a}) = \frac{1}{\mathbf{a}\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{a}}. \tag{8}$$

Due to the mismatch between the actual steering vector $\mathbf{a}$ and the presumed steering vector $\tilde{\mathbf{a}}$, we have $P(\tilde{\mathbf{a}}) \le P(\mathbf{a})$ which means that the error vector can be estimated by maximizing $P(\tilde{\mathbf{a}} + \mathbf{e})$. Equivalently, the mismatch vector can be estimated via minimizing the denominator of (8). Hence, the mismatch vector can be obtained by solving the following optimization problem

$$\begin{aligned} \min_{\mathbf{e}} \quad & (\tilde{\mathbf{a}} + \mathbf{e})^H \mathbf{R}_{\mathbf{x}}^{-1} (\tilde{\mathbf{a}} + \mathbf{e}) \\ \text{subject to} \quad & (\tilde{\mathbf{a}} + \mathbf{e})^H \mathbf{P}_{\mathbf{a}}^{\perp} = 0 \\ & \|\tilde{\mathbf{a}} + \mathbf{e}\| = \sqrt{M} \end{aligned} \tag{9}$$

where $\mathbf{P}_{\mathbf{a}}^{\perp}$ is a projection matrix onto a subspace that is orthogonal to the actual steering vector $\mathbf{a}$. The first constraint in (9) is used to prevent the corrected steering vector from converging to a steering vector associated with interfering signals. Therefore, imposing this constraint enforces the corrected steering vector to remain in the vicinity of the presumed steering vector. For this purpose, we build two orthogonal subspaces. The first subspace contains the steering vector associated with the desired signal while the other subspace contains all steering vectors associated with interfering signals. Towards this end, we build a positive definite matrix

$$\mathbf{C} \triangleq \int_{\Theta} \mathbf{c}(\theta)\mathbf{c}^H(\theta)\, d\theta \tag{10}$$

where $\mathbf{c}(\theta)$ is a steering vector that is associated with a hypothetical source that originates at the array from direction θ and $\Theta = [\theta_1 \ \theta_2]$ is a spatial sector that represents the range of the angular location of SOI[1]. We assume that Θ is centered at the presumed direction of arrival (DOA) of SOI and, hence, it can be estimated at the same stage where the presumed DOA is estimated. Then, we form the column orthogonal matrix

$$\mathbf{U} \triangleq [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K] \tag{11}$$

where $\{\mathbf{u}_k\}_{k=1}^K$ are K principal eigenvectors of $\mathbf{C}$. In (11), K is the number of dominant eigenvalues of $\mathbf{C}$. By definition, the actual steering vector $\mathbf{a}$ belongs to the subspace spanned by the columns of $\mathbf{U}$. Hence, the projection matrix $\mathbf{P}_{\mathbf{a}}^{\perp}$ can be found as

$$\mathbf{P}_{\mathbf{a}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^H. \tag{12}$$

It is also natural to impose the equality constraint in (9) in order to force the updated steering vector to have the same norm as the presumed and the actual steering vectors. However, this equality constraint represents a non-convex set. Therefore, the optimization problem (9) is not convex, and hence it is not easy to solve in a computationally efficient manner. In the following subsection, we propose an iterative solution for (9) where the equality constraint is relaxed and the non-convex problem is transformed into a convex one.

[1]The sector Θ is assumed to be distinguishable from the general angular locations of all interfering signals.

4.1. Sequential Quadratic Programming Based Implementation

The mismatch vector $\mathbf{e}$ consists of two components, one is orthogonal to the presumed vector $\tilde{\mathbf{a}}$ and the other one is parallel to it. One meaningful approach for estimating $\mathbf{e}$ is to search for the orthogonal component and to add it to the presumed steering vector to get an updated version of it. The updated version of the presumed steering vector is then scaled so that it has a $\sqrt{M}$ norm. Then, the component of the mismatch vector that is orthogonal to the updated version of the presumed steering vector can be obtained and used to update the presumed steering vector again. These iterations are repeated until the algorithm converges. The process of iterative searching, updating, and scaling is illustrated in Figure 1.

Using the notation $\mathbf{e}_\perp$ for the component of $\mathbf{e}$ that is orthogonal to $\tilde{\mathbf{a}}$, the optimization problem (9) can be reformulated as follows

$$\begin{aligned} \min_{\mathbf{e}_\perp} \quad & (\tilde{\mathbf{a}}+\mathbf{e}_\perp)^H \mathbf{R}_{\mathbf{x}}^{-1} (\tilde{\mathbf{a}}+\mathbf{e}_\perp) \\ \text{subject to} \quad & (\tilde{\mathbf{a}}+\mathbf{e}_\perp)^H \mathbf{P}_{\mathbf{a}}^{\perp} = 0 \\ & \|\tilde{\mathbf{a}}+\mathbf{e}_\perp\| \leq \sqrt{M} \\ & \tilde{\mathbf{a}}^H \mathbf{e}_\perp = 0 \end{aligned} \tag{13}$$

where the orthogonality between $\mathbf{e}$ and $\tilde{\mathbf{a}}$ is imposed by the additional constraint. Note that in the above optimization problem, the equality in the second constraint is replaced with an inequality. This relaxation does not change the optimization problem because $\|\tilde{\mathbf{a}}+\mathbf{e}_\perp\| \geq \|\tilde{\mathbf{a}}\| = \sqrt{M}$, and the only value for $\mathbf{e}_\perp$ that satisfies this constraint is zero because the norm of $\tilde{\mathbf{a}}$ is equal to $\sqrt{M}$. We can also relax the upper bound on the second constraint in (13) by adding a small number δ (the value of δ is of user choice) to the right hand side of the constraint in order to allow a space for the algorithm to search for $\mathbf{e}_\perp$. Hence, the optimization problem (13) can be modified as

$$\begin{aligned} \min_{\mathbf{e}_\perp} \quad & (\tilde{\mathbf{a}}+\mathbf{e}_\perp)^H \mathbf{R}_{\mathbf{x}}^{-1} (\tilde{\mathbf{a}}+\mathbf{e}_\perp) \\ \text{subject to} \quad & (\tilde{\mathbf{a}}+\mathbf{e}_\perp)^H \mathbf{P}_{\mathbf{a}}^{\perp} = 0 \\ & \|\tilde{\mathbf{a}}+\mathbf{e}_\perp\| \leq \sqrt{M} + \delta \\ & \tilde{\mathbf{a}}^H \mathbf{e}_\perp = 0. \end{aligned} \tag{14}$$

The optimization problem (14) is a convex quadratic programming problem that can be efficiently solved using interior point methods. After the value of the orthogonal component that minimizes the objective function of (14) is found, the updated steering vector can be projected to the sphere again, i.e., the norm of the updated steering vector can be scaled back to the value of $\sqrt{M}$. It is worth mentioning that the value of δ does not affect the value of the final solution, but rather it affects the convergence rate. Specifically, if a small value of δ is used, the number of iterations required for the algorithm to converge will be very large. On the other hand, if the value of δ is chosen to be large, then the number of iterations required for the algorithm to converge will be small.

The proposed algorithm is summarized as follows:

Step 1: Estimate $\mathbf{e}_\perp$ by solving (14).

Step 2: If $\mathbf{e}_\perp{}^H \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{e}_\perp + 2\Re\{\tilde{\mathbf{a}}^H \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{e}_\perp\} \geq -10^{-3}$, go to Step 5, where $\Re\{\cdot\}$ stands for the real part. This condition interrupts the algorithm if the inequality $(\tilde{\mathbf{a}}+\mathbf{e}_\perp)^H \mathbf{R}_{\mathbf{x}}^{-1} (\tilde{\mathbf{a}}+\mathbf{e}_\perp) \leq \tilde{\mathbf{a}}^H \mathbf{R}_{\mathbf{x}}^{-1} \tilde{\mathbf{a}}$ is not satisfied, i.e., if the objective function of (14) is not reduced at the current iteration.

Step 3: Update the presumed steering vector by setting $\tilde{\mathbf{a}} = \tilde{\mathbf{a}} + \mathbf{e}_\perp$.

Step 4: Set $\tilde{\mathbf{a}} = \sqrt{M} \frac{\tilde{\mathbf{a}}}{\|\tilde{\mathbf{a}}\|}$, then go to Step 1.

Step 5: Calculate the robust adaptive beamformer weights as

$$\mathbf{w}_{\text{SQP}} = \frac{\hat{\mathbf{R}}_{\mathbf{x}}^{-1} \tilde{\mathbf{a}}}{\tilde{\mathbf{a}}^H \mathbf{R}_{\mathbf{x}}^{-1} \tilde{\mathbf{a}}}.$$

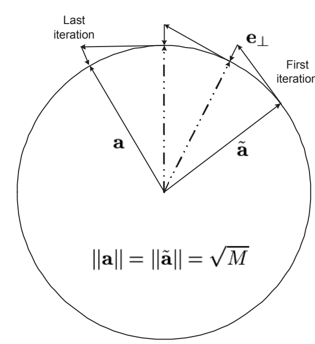

Fig. 1. Convergence trajectory of the SQP based robust adaptive beamforming algorithm.

5. SIMULATION RESULTS

In our simulations, we assume a uniform linear array of $M = 10$ omnidirectional sensors spaced half a wave length apart. The additive noise is modeled as a complex Gaussian zero-mean spatially and temporally white process that has identical variances in each array sensor. We assume two interfering sources with plane wavefronts and directions of arrival $-50°$ and $-20°$, respectively. The interference-to-noise ratio (INR) in a single sensor is equal to 30 dB. The desired signal is assumed to be a plane-wave that impinges on the array from direction $\theta = 5°$. The sample covariance matrix is computed based on $N = 1000$ data snapshots. All results are calculated based on 200 independent simulation runs.

In all examples, the proposed beamforming algorithm is compared to the the sample matrix inversion (SMI) beamformer and to the robust adaptive beamformer of [2] in terms

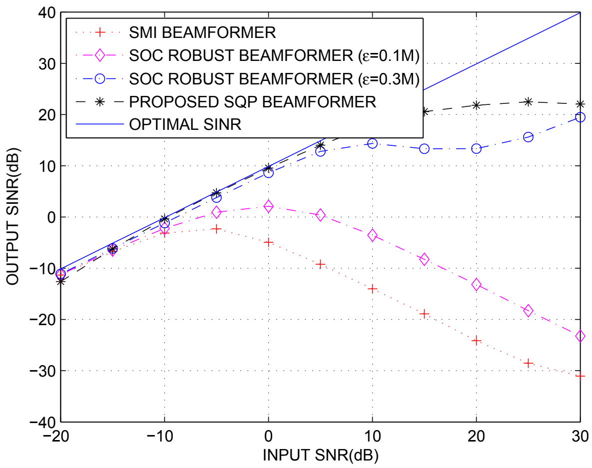

Fig. 2. Output SINR versus SNR; first example.

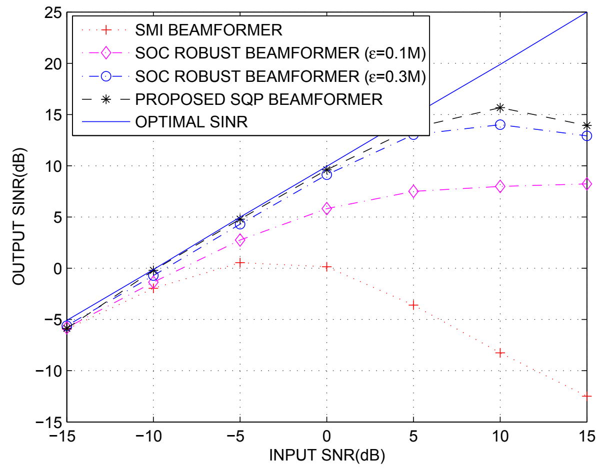

Fig. 3. Output SINR versus SNR; second example.

of the output signal-to-noise-plus-interference ratio (SINR) versus SNR. The method of [2] is referred to as the second-order cone (SOC) robust beamformer. In the proposed algorithm, the general angular location of the desired signal is assumed to be within the interval $\Theta = [0° \; 10°]$ and the value $\delta = 0.1$ is used. The number of dominant eigenvalues of the matrix $\mathbf{C}$ is taken to be equal to 4. The SeDuMi MATLAB toolbox is used to solve (14) and to compute the weight vector of the SOC robust beamformer. The proposed algorithm has always converged in all of our numerical experiments.

In the first example, we assume a look direction mismatch of 3°, i.e., the presumed steering vector is calculated at $\theta = 8°$. Two different values of $\epsilon = \{0.1M, \; 0.3M\}$ are used in the SOC beamformer. The first value is an under estimate of ϵ while the second value equals the exact norm of the mismatch vector. The performance of all methods is shown in Fig. 2 which depicts that the proposed algorithm has better performance compared to other beamformers tested.

In the second example, we consider both random sensor position errors and random look direction mismatch. Each sensor is assumed to be randomly displaced from its original location and the displacement is drawn uniformly from the set $[-0.1, \; 0.1]$ measured in wavelength. Also, the look direction mismatch is assumed to be random and uniformly distributed in $[-4°, \; 4°]$. The SOC beamformer is again computed for $\epsilon = 0.1M$ and for $\epsilon = 0.3M$. Note that the exact norm of the error vector changes from one simulation run to another. Hence, neither the value of $\epsilon = 0.1M$ nor the value of $\epsilon = 0.3M$ is an ideal choice. The performance of all methods is shown in Fig. 3. It is clear from the figure that the proposed method outperforms other techniques.

6. CONCLUSIONS

A new algorithm for robust adaptive beamforming that estimates the difference between the actual and presumed steering vectors has been developed. The estimation process has been performed iteratively where a quadratic convex optimization problem is solved at each iteration in order to update the presumed steering vector. The corrected steering vector has been used to obtain the beamformer weights. The proposed algorithm does not assume that the norm of the mismatch vector is upper bounded, and hence it does not suffer from the negative effects of over/under estimation of the upper bound. The effectiveness of the proposed algorithm has been shown using simulation results.

7. REFERENCES

[1] A. M. Vural, "Effects of perturbations on the performance of optimum/adaptive arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 15, pp. 76–87, Jan. 1979.

[2] S. A. Vorobyov, A. B. Gershman, and Z. Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem," *IEEE Trans. Signal Processing*, vol. 51, pp. 313–324, Feb. 2003.

[3] S. A. Vorobyov, A. B. Gershman, Z. Q. Luo, and N. Ma, "Adaptive beamforming with joint robustness against mismatched signal steering vector and interference nonstationarity," *IEEE Signal Processing Letters*, vol. 11, pp. 108–111, Feb. 2004.

[4] R. G. Lorenz and S. P. Boyd, "Robust minimum variance beamforming," *IEEE Trans. Signal Processing*, vol. 53, pp. 1684–1696, May 2005.

[5] A. B. Gershman, Z-Q. Luo, and S. Shahbazpanahi, "Robust adaptive beamforming based on worst-case performance optimization, in *Robust Adaptive Beamforming*, P. Stoica and J. Li, Eds., John Wiley & Sons, Hoboken, NJ, 2006, pp. 49-89.

[6] H. L. Van Trees, *Optimum Array Processing*, Wiley, NY, 2002.