LOW-RANK COVARIANCE MATRIX TAPERING FOR ROBUST ADAPTIVE BEAMFORMING

Michael Rübsamen *Christian Gerlach* *Alex B. Gershman*

Communication Systems Group
Technische Universität Darmstadt
Merckstr. 25, 64283 Darmstadt, Germany

ABSTRACT

Covariance matrix tapering (CMT) is a popular approach to improve the robustness of adaptive beamformers against moving or wideband interferers. In this paper, we develop a computationally efficient on-line implementation of the CMT technique based on a low-rank approximation of the taper matrix and the recursive least squares (RLS) algorithm. It is demonstrated that the performance of the proposed low-rank CMT approach is very close to that of the conventional CMT technique.

Index Terms— Covariance matrix tapering, low-rank approximations, robust adaptive beamforming

1. INTRODUCTION

Narrowband adaptive beamforming techniques can degrade severely in scenarios with rapidly moving or wideband interferers [1]. Therefore, several approaches have been proposed to improve the robustness of narrowband adaptive beamformers in such cases. For example, a robust modification of the Hung-Turner adaptive beamformer has been developed in [2], where data-dependent derivative constraints (DDCs) have been used to broaden the null areas of the adaptive array beampattern and to improve the beamformer robustness. In [3], the DDC approach has been extended to several other popular adaptive beamforming techniques such as the sample matrix inversion (SMI) algorithm [4]-[5] and the diagonally loaded SMI (LSMI) technique [6]-[8].

Another conceptually similar approach to robust adaptive beamforming in the presence of moving interferers has been proposed in [9]-[11]. This approach is commonly referred to as the covariance matrix tapering (CMT) technique and is based on point rather than derivative data-dependent constraints. The CMT technique amounts to tapering the array sample covariance matrix by means of a judiciously chosen matrix taper to widen the adapted beampattern nulls. In [12], it has been shown that the DDC and CMT approaches are closely related and, in particular, that the DDC technique can be interpreted and implemented as a particular form of the CMT method.

A substantial shortcoming of the conventional CMT technique is that in its general form, it is useful for batch processing but is unsuitable for on-line (adaptive) implementation. To overcome this problem, we develop a new computationally efficient on-line algorithm to implement the CMT method. Our approach is based on a low-rank approximation of the taper matrix.

2. BACKGROUND

Assume that L sources impinge on a uniform linear array (ULA) of N omnidirectional sensors with known manifold. The array steering vector is given by

$$\boldsymbol{a}(\theta) = \begin{bmatrix} 1 & e^{j\frac{2\pi d}{\lambda}\sin(\theta)} & \dots & e^{j(N-1)\frac{2\pi d}{\lambda}\sin(\theta)} \end{bmatrix}^T$$

where d is the interelement spacing, λ is the signal wavelength, and $(\cdot)^T$ denotes the transpose. The output of a narrowband beamformer is given by

$$y(k) = \boldsymbol{w}^H \boldsymbol{x}(k)$$

where k is the time index, $\boldsymbol{x}(k) = [x_0(k), \dots, x_{N-1}(k)]^T$ is the $N \times 1$ complex snapshot vector, $\boldsymbol{w} = [w_0, \dots, w_{N-1}]^T$ is the $N \times 1$ complex vector of beamformer weights, and $(\cdot)^H$ denotes the Hermitian transpose. The snapshot vector can be written as

$$\begin{aligned} \boldsymbol{x}(k) &= \boldsymbol{s}(k) + \boldsymbol{i}(k) + \boldsymbol{n}(k) \\ &= s_s(k)\boldsymbol{a}_s + \boldsymbol{i}(k) + \boldsymbol{n}(k) \end{aligned}$$

where $\boldsymbol{s}(k)$, $\boldsymbol{i}(k)$, and $\boldsymbol{n}(k)$ are the signal, interference, and noise components, respectively, $s_s(k)$ is the signal waveform, and $\boldsymbol{a}_s$ is the signal steering vector.

The optimal weight vector can be found by means of maximizing the signal-to-interference-plus-noise ratio (SINR)

$$\text{SINR} = \frac{\sigma_s^2 |\boldsymbol{w}^H \boldsymbol{a}_s|^2}{\boldsymbol{w}^H \boldsymbol{R}_{i+n} \boldsymbol{w}}$$

where $\sigma_s^2 = \mathrm{E}\{|s_s(k)|^2\}$ is the signal power,

$$\boldsymbol{R}_{i+n} \triangleq \mathrm{E}\{(\boldsymbol{i}(k) + \boldsymbol{n}(k))(\boldsymbol{i}(k) + \boldsymbol{n}(k))^H\}$$

is the $N \times N$ interference-plus-noise covariance matrix, and $\mathrm{E}\{\cdot\}$ denotes the statistical expectation. The maximization of the SINR is equivalent to solving the following minimum variance distortionless response (MVDR) problem:

$$\min_{\boldsymbol{w}} \boldsymbol{w}^H \boldsymbol{R}_{i+n} \boldsymbol{w} \qquad \text{s.t.} \qquad \boldsymbol{w}^H \boldsymbol{a}_s = 1. \tag{1}$$

The solution to (1) is given by [5]

$$\boldsymbol{w}_{\text{opt}} = \alpha \boldsymbol{R}_{i+n}^{-1} \boldsymbol{a}_s,$$

where $\alpha = (\boldsymbol{a}_s^H \boldsymbol{R}_{i+n}^{-1} \boldsymbol{a}_s)^{-1}$ is a normalization constant that does not affect the SINR and thus will be omitted in the sequel.

In the on-line mode, an exponential time window may be used to update the covariance matrix $\boldsymbol{R}$ as

$$\hat{\boldsymbol{R}}(k) = \left(1 - \frac{1}{K}\right) \hat{\boldsymbol{R}}(k-1) + \frac{1}{K} \boldsymbol{x}(k) \boldsymbol{x}^H(k). \tag{2}$$

This leads to the SMI beamformer

$$\boldsymbol{w}_{\text{SMI}}(k) = \hat{\boldsymbol{R}}^{-1}(k)\boldsymbol{a}_s, \quad (3)$$

where $\boldsymbol{w}_{\text{SMI}}(k)$ can be computed from $\boldsymbol{w}_{\text{SMI}}(k-1)$ with the complexity $O(N^2)$ using the RLS algorithm.

The essence of the CMT method is to use the tapered covariance matrix

$$\hat{\boldsymbol{R}}_{\text{T}}(k) = \hat{\boldsymbol{R}}(k) \odot \boldsymbol{T} \quad (4)$$

instead of $\hat{\boldsymbol{R}}(k)$ in (3). Here, $\boldsymbol{T}$ is the $N \times N$ taper matrix and $\odot$ denotes the Schur-Hadamard (elementwise) matrix product. Different taper matrices have been suggested in the literature; see [9]-[11] and [13].

In the on-line mode, the matrix $\hat{\boldsymbol{R}}_{\text{T}}(k)$ can be updated as

$$\hat{\boldsymbol{R}}_{\text{T}}(k) = \left(1 - \frac{1}{K}\right)\hat{\boldsymbol{R}}_{\text{T}}(k-1) + \frac{1}{K}(\boldsymbol{x}(k)\boldsymbol{x}^H(k)) \odot \boldsymbol{T}. \quad (5)$$

In contrast to (2), each update of the weight vector using (5) requires $O(N^3)$ operations in the general case.

The most commonly used taper matrix is [9], [10]

$$[\boldsymbol{T}_1]_{l,m} = \frac{\sin(\pi(l-m)\gamma)}{\pi(l-m)\gamma} \quad (6)$$

where γ determines the width of the beampattern nulls. It has been shown in [13] that

$$\boldsymbol{R} \odot \boldsymbol{T}_1 = \text{E}\left\{(\boldsymbol{x}(k) \odot \boldsymbol{e}(\Omega))\,(\boldsymbol{x}(k) \odot \boldsymbol{e}(\Omega))^H\right\} \quad (7)$$

where the random vector $\boldsymbol{e}(\Omega)$ is statistically independent from the snapshot vector $\boldsymbol{x}(k)$ and is defined as

$$\boldsymbol{e}(\Omega) \triangleq \begin{bmatrix} 1 & e^{j\Omega} & e^{j2\Omega} & \dots & e^{j(N-1)\Omega} \end{bmatrix}^T. \quad (8)$$

Here, the random variable Ω is uniformly distributed in the interval $-\gamma\pi \leq \Omega \leq \gamma\pi$. Furthermore, it follows from (7) that

$$\boldsymbol{T}_1 = \text{E}\{\boldsymbol{e}(\Omega)\boldsymbol{e}(\Omega)^H\}. \quad (9)$$

The DDC methods of [3] are based on the observation that the m-th derivative of the array steering vector of a linear array can be written as

$$\frac{\partial^m \boldsymbol{a}(\theta)}{\partial \theta^m} = \alpha_m(\theta)\boldsymbol{D}^m\boldsymbol{a}(\theta)$$

where $\alpha_m(\theta)$ is a scalar and

$$\boldsymbol{D} \triangleq \text{diag}\left\{ 0 \quad 1 \quad \dots \quad N-1 \right\}$$

does not depend on θ. Using M data-dependent derivative constraints, the snapshot covariance matrix should be replaced by

$$\hat{\boldsymbol{R}}_{\text{DDC}}(k) = \hat{\boldsymbol{R}}(k) + \sum_{m=1}^{M} \xi_m \boldsymbol{D}^m \hat{\boldsymbol{R}}(k)\boldsymbol{D}^m \quad (10)$$

where the coefficients $\xi_1, \dots, \xi_M$ adjust the weights of the derivative constraint terms. It has been shown in [12] that (10) can be written as

$$\hat{\boldsymbol{R}}_{\text{DDC}}(k) = \hat{\boldsymbol{R}}(k) \odot \boldsymbol{T}_2,$$

where

$$\boldsymbol{T}_2 = \boldsymbol{1}_N + \sum_{m=1}^{M} \xi_m \boldsymbol{d}^m {\boldsymbol{d}^m}^H,$$

$\boldsymbol{1}_N$ denotes an $N \times N$ matrix of ones, the vector $\boldsymbol{d}$ stacks the diagonal entries of $\boldsymbol{D}$, and the vector $\boldsymbol{d}^m$ results from taking the m-th power of each element of $\boldsymbol{d}$. In what follows, we assume that the weights are chosen as $\xi_m = N/\|\boldsymbol{d}^m {\boldsymbol{d}^m}^H\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm.

3. LOW-RANK IMPLEMENTATION OF THE CMT METHOD

As mentioned before, the computational complexity of the CMT algorithm in the on-line mode is dominated by the computation of

$$\boldsymbol{w}_{\text{T}}(k) = \hat{\boldsymbol{R}}_{\text{T}}(k)^{-1}\boldsymbol{a}_s$$

where $O(N^3)$ operations are required in general to compute $\boldsymbol{w}_{\text{T}}(k)$ from $\boldsymbol{w}_{\text{T}}(k-1)$.

A lower computational complexity may be achieved by means of using low-rank matrix tapers that will be introduced below. As all practically relevant matrix tapers are Hermitian positive-definite matrices, any taper matrix can be eigendecomposed as

$$\boldsymbol{T} = \sum_{i=1}^{N} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^H \quad (11)$$

where λ_i ($i = 1, \dots, N$) are the eigenvalues sorted in descending order and $\boldsymbol{u}_i$ ($i = 1, \dots, N$) are the corresponding eigenvectors. Then, a low-rank approximation of $\boldsymbol{T}$ yields

$$\boldsymbol{T} \simeq \sum_{i=1}^{J} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^H \quad (12)$$

where $J < N$ (and typically, $J \ll N$). Inserting (12) into (4), we have

$$\begin{aligned}\hat{\boldsymbol{R}}_{\text{T}}(k) &= \left(1 - \frac{1}{K}\right)\hat{\boldsymbol{R}}_{\text{T}}(k-1) \\ &\quad + \frac{1}{K}\left(\boldsymbol{x}(k)\boldsymbol{x}^H(k)\right) \odot \left(\sum_{i=1}^{J} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^H\right) \\ &= \left(1 - \frac{1}{K}\right)\hat{\boldsymbol{R}}_{\text{T}}(k-1) + \frac{1}{K}\sum_{i=1}^{J} \lambda_i \boldsymbol{v}_i(k)\boldsymbol{v}_i^H(k)\end{aligned}$$

where

$$\boldsymbol{v}_i(k) \triangleq \boldsymbol{x}(k) \odot \boldsymbol{u}_i.$$

Applying the matrix inversion lemma and using the RLS algorithm, we have that in the low-rank taper case, the computational complexity of updating the weight vector is $O(JN^2)$ per step. Therefore, a significantly more efficient on-line implementation of the CMT method can be achieved if the taper matrix is approximated by a low-rank matrix. Note that the taper matrix $\boldsymbol{T}_2$ is inherently low-rank for $M \ll N$, whereas the taper matrix $\boldsymbol{T}_1$ is full-rank. However, as will be shown in Section 4, the latter matrix can be approximated by a low-rank matrix without any significant performance degradation.

There are several ways to obtain low-rank taper matrices from their full-rank counterparts. One approach is to take the dominant terms of the eigendecomposition, as in (12), while another physically motivated approach for the particular case of the matrix $\boldsymbol{T}_1$ follows from (9):

$$\boldsymbol{T}_1 \simeq \frac{1}{J}\sum_{i=1}^{J} \boldsymbol{e}(\Omega_i)\boldsymbol{e}^H(\Omega_i) \quad (13)$$

where $\boldsymbol{e}(\Omega_i)$ denotes the vector of (8) evaluated at $\Omega = \Omega_i$. The user-defined values Ω_i ($i = 1, \dots, J$) are samples of the interval $-\gamma\pi \leq \Omega \leq \gamma\pi$.

If the CMT approach is combined with diagonal loading, the repeated multiplication of the covariance matrix with the factor $1 - 1/K$ in the exponential window case leads to suppression of the diagonal loading term. Therefore, in the case of diagonal loading, a rectangular window has to be employed, that is, the covariance matrix has to be updated as

$$\hat{\boldsymbol{R}}(k) = \hat{\boldsymbol{R}}(k-1) + \frac{1}{K}\boldsymbol{x}(k)\boldsymbol{x}^H(k) - \frac{1}{K}\boldsymbol{x}(k-K)\boldsymbol{x}^H(k-K). \quad (14)$$

In the rectangular window case, the use of low-rank CMT yields the following update of the matrix $\hat{\boldsymbol{R}}_{\mathrm{T}}(k)$:

$$\hat{\boldsymbol{R}}_{\mathrm{T}}(k) = \hat{\boldsymbol{R}}_{\mathrm{T}}(k-1) + \frac{1}{K}\sum_{i=1}^{J}\lambda_i(\boldsymbol{v}_i(k)\boldsymbol{v}_i^H(k) - \boldsymbol{v}_i(k-K)\boldsymbol{v}_i^H(k-K)).$$

Therefore, as in the case of exponential window, the complexity of updating the weight vector with the rectangular window is $O(JN^2)$ per step.

4. SIMULATION RESULTS

We assume a ULA of $N = 20$ sensors spaced half a wavelength apart.

Fig. 1 displays the magnitude of the 10 largest eigenvalues of the 20×20 taper matrix $\boldsymbol{T}_1$ of (6). The values are normalized with respect to the eigenvalue with the largest magnitude. This figure demonstrates that the number of negligible eigenvalues increases when decreasing the parameter γ.

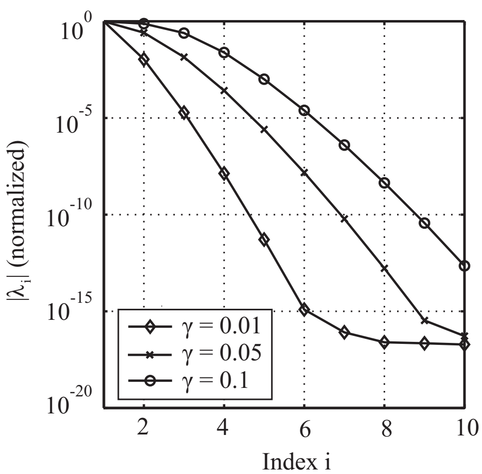

Fig. 1. Magnitude of the 10 largest eigenvalues of the 20×20 taper matrix $\boldsymbol{T}_1$ for different values of γ.

Let us assume that the signal of interest impinges on the array from the direction $\theta_s = 0°$ with the signal-to-noise ratio (SNR) of 3 dB, and that there is one interferer impinging from the direction-of-arrival (DOA) $\theta_i = -10°$ with the interference-to-noise ratio (INR) of 30 dB. For the taper matrix $\boldsymbol{T}_1$, $\gamma = 0.05$ is assumed. The following CMT-based SMI beamformers are compared:

- BF #1: The beamformer using the taper matrix $\boldsymbol{T}_1$.
- BF #2: The beamformer using the taper matrix $\boldsymbol{T}_2$ with 2 DDC constraints. This results in the rank of the taper matrix equal to three.
- BF #3: The beamformer using the low-rank approximation (12) of $\boldsymbol{T}_1$ with $J = 3$.
- BF #4: The beamformer using the low-rank approximation (13) with uniform sampling grid in Ω and $J = 3$.

In our first example, the weight vectors of all the beamformers tested are computed using the exact $\boldsymbol{R}_{i+n}$ and the resulting beampatterns are shown in Fig. 2. From this figure, it can be observed that there is a certain similarity between the beampatterns that correspond to the low-rank and full-rank CMT cases.

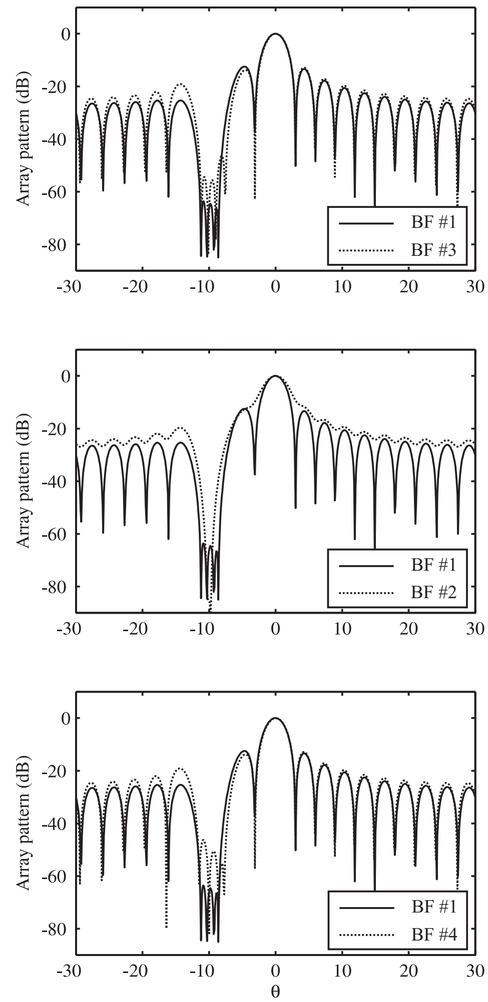

Fig. 2. Adaptive array beampatterns.

In our second example, the matrix $\boldsymbol{R}_{i+n}$ is estimated using the exponential window with $K = 50$. For the signal-of-interest, we use the same settings as in our first example, but now assume three equal-power interferers with INR = 30 dB and the DOAs $\theta_{i,1} = 20° + 5° \sin(k/10)$, $\theta_{i,2} = -40° + 10° \cos(k/12)$, and $\theta_{i,3} = -25° + 12° \sin(k/15)$. We also assume that the training snapshots contain no signal component. Fig. 3 displays the output SINR of the beamformers tested versus the time index k. The optimum SINR curve is also shown in this figure.

From the first plot of Fig. 3, it follows that the beamformers BF #1 and BF #3 have nearly the same performance, i.e., the low-rank approximation of $\boldsymbol{T}_1$ does not lead to any visible performance degradation. From the second plot of Fig. 3, it can be seen that the performances of beamformers BF #1 and BF #2 are quite similar as well, even though these beamformers use different types of constraints and different ranks of the taper matrices. The third plot of Fig. 3 demonstrates that beamformer BF #4 has somewhat better performance than BF #1. Interestingly, this implies that low-rank CMT techniques in some scenarios can perform even better than their full-rank counterparts.

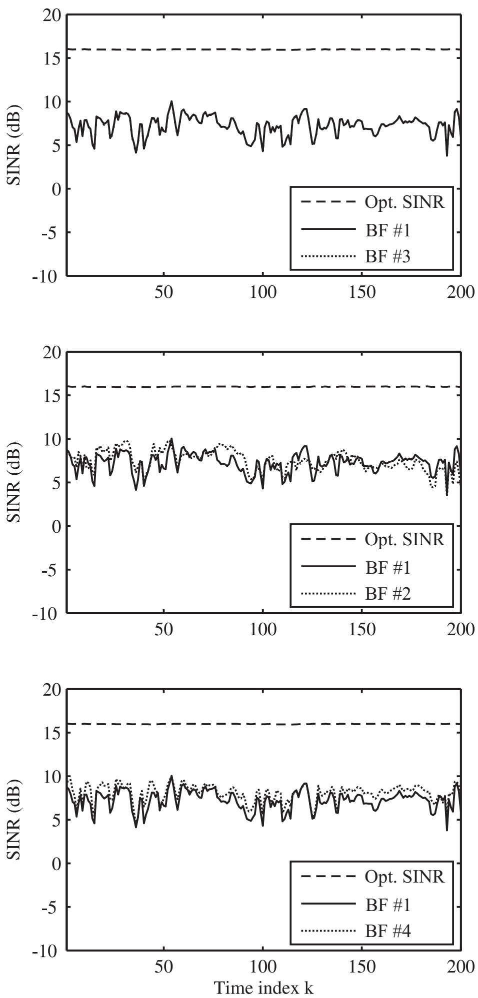

Fig. 3. Output SINRs versus time index.

5. CONCLUSIONS

A novel low-rank approach has been developed for a computationally efficient on-line implementation of covariance matrix tapering based robust beamforming techniques. The proposed low-rank covariance matrix tapering approach is shown not to suffer any performance degradation as compared to its full-rank counterpart.

6. REFERENCES

[1] A. B. Gershman, "Robust adaptive beamforming in sensor arrays," *AEÜ – Int. Journal of Electronics and Communications*, vol. 53, no. 6, pp. 305-314, Dec. 1999.

[2] A. B. Gershman, G. V. Serebryakov, and J. F. Böhme, "Constrained Hung-Turner adaptive beam-forming algorithm with additional robustness to wideband and moving jammers," *IEEE Trans. Antennas and Propagation*, vol. 44, pp. 361–367, March 1996.

[3] A. B. Gershman, U. Nickel, and J. F. Böhme, "Adaptive beamforming algorithms with robustness against jammer motion," *IEEE Trans. Signal Processing*, vol. 45, pp. 1878-1885, July 1997.

[4] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Trans. Aerospace and Electron. Syst.*, vol. 10, pp. 853–863, Nov. 1974.

[5] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*, Wiley, New York, 1980.

[6] Y. I. Abramovich, "Controlled method for adaptive optimization of filters using the criterion of maximum SNR," *Radio Engineering and Electronic Physics*, vol. 26, pp. 87–95, March 1981.

[7] B. D. Carlson, "Covariance matrix estimation errors and diagonal loading in adaptive arrays," *IEEE Trans. Aerospace and Electron. Syst.*, vol. 24, pp. 397–401, July 1988.

[8] H. Cox, R. M. Zeskind, and M. H. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, pp. 1365–1376, Oct. 1987.

[9] R. J. Mailloux, "Covariance matrix augmentation to produce adaptive array pattern troughs," *IEE Electron. Letters*, vol. 31, no. 10, pp. 771–772, 1995.

[10] M. Zatman, "Production of adaptive array troughs by dispersion synthesis," *IEE Electron. Lett.*, vol. 31, no. 25, pp. 2141–2142, 1995.

[11] J. Riba, J. Goldberg, and G. Vazquez, "Robust beamforming for interference rejection in mobile communications", *IEEE Trans. Signal Processing*, vol. 45, pp. 271–275, Jan. 1997.

[12] M. Zatman, "Comments on "Theory and application of covariance matrix tapers for robust adaptive beamforming," *IEEE Trans. Signal Processing*, vol. 48, pp. 1796–1800, June 2000.

[13] J. R. Guerci, "Theory and application of covariance matrix tapers for robust adaptive beamforming," *IEEE Trans. Signal Processing*, vol. 47, pp. 977–985, Apr. 1999.