Low-Complexity Robust Adaptive Beamforming Algorithms Exploiting Shrinkage for Mismatch Estimation

Hang Ruan and Rodrigo C. de Lamare

Abstract— **This paper proposes low-complexity robust adaptive beamforming (RAB) techniques based on shrinkage methods. We firstly review a Low-Complexity Shrinkage-Based Mismatch Estimation (LOCSME) batch algorithm to estimate the desired signal steering vector mismatch, in which the interference-plus-noise covariance (INC) matrix is also estimated with a recursive matrix shrinkage method. Then we develop low complexity adaptive recursive versions of stochastic gradient (SG) and conjugate gradient (CG) to update the beamforming weights, resulting in low-cost robust adaptive algorithms. An analysis of the effect of shrinkage on the estimation procedure is developed along with a computational complexity study of the proposed and existing algorithms. Simulations are conducted in local scattering scenarios and comparisons to existing RAB techniques are provided.**

Keywords—**robust adaptive beamforming, shrinkage methods, low complexity methods.**

I. INTRODUCTION

Sensor array signal processing techniques and their applications to wireless communications, sensor networks and radar have been widely investigated in recent years. Adaptive beamforming is one of the most important topics in sensor array signal processing which has applications in many fields. However, adaptive beamformers may suffer performance degradation due to small sample data size or the presence of the desired signal in the training data. In practical environments, desired signal steering vector mismatch problems like signal pointing errors [16], imprecise knowledge of the antenna array, look-direction mismatch or local scattering may even lead to more significant performance loss [4].

A. Prior and Related Work

In order to address these problems, robust adaptive beamforming (RAB) techniques have been developed in recent years. Popular approaches include worst-case optimization [4], diagonal loading [5], [6], [25], and eigen-decomposition [15], [16]. However, general RAB designs have some limitations such as their ad hoc nature, high probability of subspace swap at low SNR and high computational cost [7].

Further recent works have looked at approaches based on combined estimation procedures for both the steering vector mismatch and interference-plus-noise covariance (INC) matrix to improve RAB performance. The worst-case optimization methods in [4], [21], [22], [23] solve an online semi-definite programming (SDP) while using a matrix inversion to estimate the INC matrix. The method in [10] estimates the steering vector mismatch by solving an online Sequential Quadratic Program (SQP) [8], while estimating the INC matrix using a shrinkage method [10]. Another similar method which jointly estimates the steering vector using SQP and the INC matrix using a covariance reconstruction method [11], presents outstanding performance compared to other RAB techniques. However, their main disadvantages include the high computational cost associated with online optimization programming, the matrix inversion or reconstruction process, and slow convergence.

Our recent work in [14] has introduced a Low-Complexity Shrinkage-Based Mismatch Estimation (LOCSME) algorithm, which implements an efficient iterative robust beamforming method with precise estimation of the steering vector mismatch. In this method, an extension of the Oracle Approximating Shrinkage (OAS) method [12] is employed to perform vector shrinkage estimation of the cross-correlation vector between the sensor array received data and the beamformer output. The mismatched steering vector is efficiently estimated without any costly optimization procedure in a low-complexity sense. Then, we estimate the desired signal power based on the desired signal steering vector and the received data. In a subsequent step, we perform matrix shrinkage to the sample covariance matrix (SCM), from which the covariance matrix of the desired signal is computed and subtracted to obtain a further estimated INC matrix. Then the output signal-to-interference-plus-noise ratio (SINR) can be computed directly.

B. Contributions

In this work, we firstly develop a stochastic gradient (SG) adaptive version of the LOCSME technique in [14], denoted LOCSME-SG, which does not require matrix inversions or costly recursions to update the beamforming weights adaptively. In particular, the SCM is estimated only once using a knowledge-aided (KA) shrinkage [19], [20] algorithm along with the computation of the beamforming weights based on the estimated steering vector through SG recursions. Secondly, we also develop an adaptive LOCSME technique based on the conjugate gradient (CG) adaptive algorithm, resulting in CG type algorithms, denoted LOCSME-CCG and LOCSME-MCG. Different from LOCSME-SG, the CG type algorithms not only updates the beamforming weights, but can also estimate the mismatched steering vector, which sequentially performs the estimation of the mismatched vector by LOCSME in every snapshot. An analysis shows that both LOCSME-SG and LOCSME-CG achieve one degree lower complexity than the original LOCSME. Simulations also show an excellent performance which benefits from the precise estimation provided by the shrinkage approach. Our contributions are summarized as follows:

- The development of LOCSME type SG and CG algorithms.
- An investigation of the effect of shrinkage on the estimation accuracy of the algorithms.
- A study of the performance and the complexity of the pro-

posed and existing algorithms.

The paper is organized as follows. The system model and problem statement are described in Section II. A review of the LOCSME algorithm is provided in Section III whereas Section IV presents the proposed adaptive LOCSME-SG, LOCSME-CCG and LOCSME-MCG algorithms. Section V provides the shrinkage and complexity analyses. Section VI presents the simulation results. Section VII gives the conclusion.

II. System Model and Problem Statement

Consider a linear antenna array of M sensors and K narrowband signals which impinge on the array. The data received at the ith snapshot can be modeled as

$$\mathbf{x}(i) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(i) + \mathbf{n}(i), \tag{1}$$

where $\mathbf{s}(i) \in \mathbb{C}^{K\times 1}$ are uncorrelated source signals, $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_K]^T \in \mathbb{R}^K$ is a vector containing the directions of arrival (DoAs), $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1) + \mathbf{e}, \cdots, \mathbf{a}(\theta_K)] \in \mathbb{C}^{M\times K}$ is the matrix which contains the steering vector for each DoA and $\mathbf{e}$ is the steering vector mismatch of the desired signal, $\mathbf{n}(i) \in \mathbb{C}^{M\times 1}$ is assumed to be complex Gaussian noise with zero mean and variance σ_n^2. The beamformer output is

$$y(i) = \mathbf{w}^H\mathbf{x}(i), \tag{2}$$

where $\mathbf{w} = [w_1, \cdots, w_M]^T \in \mathbb{C}^{M\times 1}$ is the beamformer weight vector, where $(\cdot)^H$ denotes the Hermitian transpose. The optimum beamformer is computed by maximizing the signal-to-interference-plus-noise ratio (SINR) given by

$$SINR = \frac{\sigma_1^2|\mathbf{w}^H\mathbf{a}_1|^2}{\mathbf{w}^H\mathbf{R}_{i+n}\mathbf{w}}. \tag{3}$$

where σ_1^2 is the desired signal power, $\mathbf{R}_{i+n}$ is the INC matrix. Assuming that the steering vector $\mathbf{a}_1$ is known precisely ($\mathbf{a}_1 = \mathbf{a}(\theta_1)$), then problem (3) can be cast as an optimization problem

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} \quad \mathbf{w}^H\mathbf{R}_{i+n}\mathbf{w} \\ &\text{subject to} \quad \mathbf{w}^H\mathbf{a}_1 = 1, \end{aligned} \tag{4}$$

which is known as the MVDR beamformer or Capon beamformer [1], [2]. The optimum weight vector is given by $\mathbf{w}_{opt} = \frac{\mathbf{R}_{i+n}^{-1}\mathbf{a}_1}{\mathbf{a}_1{}^H\mathbf{R}_{i+n}^{-1}\mathbf{a}_1}$. Since $\mathbf{R}_{i+n}$ is usually unknown in practice, it can be estimated by the SCM of the received data as

$$\hat{\mathbf{R}}(i) = \frac{1}{i}\sum_{k=1}^{i}\mathbf{x}(k)\mathbf{x}^H(k), \tag{5}$$

which results in the Sample Matrix Inversion (SMI) beamformer $\mathbf{w}_{SMI} = \frac{\hat{\mathbf{R}}^{-1}\mathbf{a}_1}{\mathbf{a}_1{}^H\hat{\mathbf{R}}^{-1}\mathbf{a}_1}$. However, the SMI beamformer requires a large number of snapshots to converge and is sensitive to steering vector mismatches [10], [11]. The problem we are interested in solving is how to design low-complexity robust adaptive beamforming algorithms that can preserve the SINR performance in the presence of uncertainties in the steering vector of a desired signal.

III. LOCSME Robust Beamforming Algorithm

In this section, the LOCSME algorithm [14] is briefly reviewed. The basic idea of LOCSME [14] is to obtain a precise estimate of the desired signal steering vector by exploiting cross-correlation vector between the beamformer output and the array observation data and then computing the beamforming weights.

A. Steering Vector Estimation

The cross-correlation between the array observation data and the beamformer output can be expressed as $\mathbf{d} = E\{\mathbf{x}y^*\}$. With assumptions that $|\mathbf{a}_m\mathbf{w}| \ll |\mathbf{a}_1\mathbf{w}|$ for $m = 2, \cdots, K$ and that the signal sources and that the system noise have zero mean while the desired signal is independent from the interferers and the noise, $\mathbf{d}$ can be rewritten as $\mathbf{d} = E\{\sigma_1{}^2\mathbf{a}_1^H\mathbf{w}\mathbf{a}_1 + \mathbf{n}\mathbf{n}^H\mathbf{w}\}$. By projecting $\mathbf{d}$ onto a predefined subspace [9], which collects all possible information from the desired signal, the unwanted part of $\mathbf{d}$ can be eliminated. LOCSME also exploits prior knowledge which amounts to providing an angular sector in which the desired signal is located, say $[\theta_1 - \theta_e, \theta_1 + \theta_e]$. The subspace projection matrix $\mathbf{P}$ is given by

$$\mathbf{P} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_p][\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_p]^H, \tag{6}$$

where $\mathbf{c}_1, \cdots, \mathbf{c}_p$ are the p principal eigenvectors of the matrix $\mathbf{C}$ (where p is chosen by the user), which is defined by [8]

$$\mathbf{C} = \int_{\theta_1-\theta_e}^{\theta_1+\theta_e}\mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta. \tag{7}$$

In order to achieve a better estimation of the steering vector, we employ the OAS shrinkage technique to obtain a more accurate estimate of the vector $\mathbf{d}$. Let us define the sample correlation vector (SCV) in snapshot i as

$$\hat{\mathbf{l}}(i) = \frac{1}{i}\sum_{k=1}^{i}\mathbf{x}(k)y^*(k), \tag{8}$$

and its mean value as

$$\hat{\nu}(i) = \sum\hat{\mathbf{l}}(i)/M. \tag{9}$$

Then we aim to shrink the SCV towards its mean value $\hat{\nu}(i)$, which yields

$$\hat{\mathbf{d}}(i) = \hat{\rho}(i)\hat{\nu}(i) + (1 - \hat{\rho}(i))\hat{\mathbf{l}}(i), \tag{10}$$

where $\hat{\rho}(i)$ represents the shrinkage coefficient ($\hat{\rho}(i) \in (0, 1)$). To find out the optimum $\hat{\rho}(i)$, we minimize the mean square error (MSE) of $E[\|\hat{\mathbf{d}}(i) - \hat{\mathbf{d}}(i-1)\|^2]$, which leads to

$$\hat{\rho}(i) = \frac{(1-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1) + \sum\hat{\mathbf{d}}(i-1)\sum^*\hat{\mathbf{d}}(i-1)}{(i-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1) + (1-\frac{i}{M})\sum\hat{\mathbf{d}}(i-1)\sum^*\hat{\mathbf{d}}(i-1)}. \tag{11}$$

Once the correlation vector $\hat{\mathbf{d}}$ is obtained, the steering vector is estimated by

$$\hat{\mathbf{a}}_1(i) = \frac{\mathbf{P}\hat{\mathbf{d}}(i)}{\|\mathbf{P}\hat{\mathbf{d}}(i)\|_2}. \tag{12}$$

B. Desired Signal Power Estimation

This subsection will introduce a method to estimate the desired signal power σ_1^2. This can be accomplished by directly using the desired signal steering vector. Let us rewrite the received data as

$$\mathbf{x}(i) = \hat{\mathbf{a}}_1(i)s_1 + \sum_{k=2}^{K} \mathbf{a}_k s_k + \mathbf{n}(i). \quad (13)$$

Pre-multiplying the above equation by $\hat{\mathbf{a}}_1^H(i)$ and assuming $\hat{\mathbf{a}}_1(i)$ is uncorrelated with the interferers, we obtain

$$\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i) = \hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1 + \hat{\mathbf{a}}_1^H(i)\mathbf{n}(i). \quad (14)$$

Taking the expectation of $|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2$, we obtain

$$E[|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2] = E[(\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1 + \hat{\mathbf{a}}_1^H(i)\mathbf{n}(i))^* (\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)s_1 + \hat{\mathbf{a}}_1^H(i)\mathbf{n}(i))]. \quad (15)$$

If the noise is statistically independent from the desired signal, then we have

$$E[|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2] = |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2 E[|s_1|^2] + \hat{\mathbf{a}}_1^H(i)E[\mathbf{n}(i)\mathbf{n}^H(i)]\hat{\mathbf{a}}_1(i), \quad (16)$$

where $E[\mathbf{n}(i)\mathbf{n}^H(i)]$ represents the noise covariance matrix $\mathbf{R}_n$ which can be replaced by $\sigma_n^2\mathbf{I}_M$, where σ_n^2 is assumed known here for convenience, otherwise it can be easily estimated by a specific estimation method. A proper approach is to use a Maximum Likelihood (ML) based method as in [24]. A specialized comparison between the cases when the noise power is assumed known or estimated is also given in the simulations. Replacing the desired signal power $E[|s_1|^2]$ by its estimate $\hat{\sigma}_1^2(i)$, the desired signal power estimate is computed as

$$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2 - |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\sigma_n^2}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}. \quad (17)$$

Equation (17) has a low complexity ($\mathcal{O}(M)$) and can be directly implemented if the desired signal steering vector is accurately estimated and the noise level is known.

C. Estimation of the INC matrix

In this subsection, we describe a method to estimate the INC matrix that is based on the OAS matrix shrinkage method [12] and used in LOCSME. First of all, we need the SCM in (5) as a preliminary estimate for the INC matrix. Then we define $\hat{\mathbf{F}}_0 = \hat{\nu}_0\mathbf{I}$, where $\hat{\nu}_0 = \mathrm{tr}(\hat{\mathbf{R}})/M$. By minimizing the MSE described by $E[\|\tilde{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2]$, the following recursion is employed:

$$\tilde{\mathbf{R}}(i) = \hat{\rho}_0(i)\hat{\mathbf{F}}_0(i) + (1 - \hat{\rho}_0(i))\hat{\mathbf{R}}(i), \quad (18)$$

$$\hat{\rho}_0(i) = \frac{(1-\frac{2}{M})\mathrm{tr}(\tilde{\mathbf{R}}(i-1)\hat{\mathbf{R}}(i-1)) + \mathrm{tr}^2(\tilde{\mathbf{R}}(i-1))}{(i-\frac{2}{M})\mathrm{tr}(\tilde{\mathbf{R}}(i-1)\hat{\mathbf{R}}(i-1)) + (1-\frac{i}{M})\mathrm{tr}^2(\tilde{\mathbf{R}}(i-1))}, \quad (19)$$

where $\hat{\rho}_0(0)$ must be initialized between 0 and 1 to guarantee convergence [12]. To exclude the information of the desired signal from the covariance matrix of the sensor array observation data, a simple subtraction is considered

$$\tilde{\mathbf{R}}_{i+n}(i) = \tilde{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i). \quad (20)$$

D. Computation of Beamforming Weights

The beamforming weights of LOCSME are computed directly by

$$\hat{\mathbf{w}}(i) = \frac{\tilde{\mathbf{R}}_{i+n}^{-1}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\tilde{\mathbf{R}}_{i+n}^{-1}(i)\hat{\mathbf{a}}_1(i)}, \quad (21)$$

which has a computationally costly matrix inversion $\tilde{\mathbf{R}}_{i+n}^{-1}(i)$. to reproduce the LOCSME algorithm, whose complexity is $\mathcal{O}(M^3)$, equations (9)-(12) and (17)-(21) are required. In comparison to previously reported RAB algorithms in [7], [8], [10], [11] with costly online optimization procedures and complexity $\mathcal{O}(M^3)$ or higher, LOCSME requires lower cost.

IV. Proposed Adaptive Algorithms

In this section, we develop adaptive strategies based on the LOCSME robust beamforming technique, resulting in the proposed LOCSME-SG, LOCSME-CCG and LOCSME-MCG algorithms. These algorithms are developed for implementation purposes and are especially suitable for dynamic scenarios. In these adaptive algorithms, we employ the same recursions as in LOCSME to estimate the steering vector and the desired signal power, whereas the estimation procedures of the INC matrix and the beamforming weights are different. In particular, LOCSME-SG employs a low-cost KA shrinkage method to estimate the INC matrix. For LOCSME-SG, LOCSME-CCG and LOCSME-MCG, the weight vector update equation is derived from a reformulated optimization problem.

A. LOCSME-SG Adaptive Algorithms

With the estimate of the desired signal power we subtract unwanted information of the interferences out from the array received data to obtain a modified array observation (MAO) vector. Consider a simple substraction step as

$$\mathbf{x}_{i+n}(i) = \mathbf{x}(i) - \hat{\sigma}_1(i)\hat{\mathbf{a}}_1(i). \quad (22)$$

Then the INC matrix can be estimated by

$$\hat{\mathbf{R}}_{i+n}(i) = \mathbf{x}_{i+n}(i)\mathbf{x}_{i+n}^H(i). \quad (23)$$

Now, we employ the idea of KA shrinkage method [19], [20] to help with our INC estimation. By applying a linear shrinkage model to the INC matrix, we have

$$\breve{\mathbf{R}}_{i+n}(i) = \eta(i)\mathbf{R}_0 + (1 - \eta(i))\hat{\mathbf{R}}_{i+n}(i), \quad (24)$$

where $\mathbf{R}_0$ is an initial guess for the INC matrix, $\eta(i)$ is the shrinkage parameter and $\eta(i) \in (0, 1)$. Here the shrinkage parameter is expected to be adaptively estimated. Employing an idea of adaptive filtering [19], [20], it is possible to set the overall filter output $y_f(i)$ equal to $[\breve{\mathbf{R}}_{i+n}(i)\hat{\mathbf{a}}_1(i)]^H\mathbf{x}(i)$

which is the linear combination of the outputs from two filter elements which are $y_{0f}(i) = [\mathbf{R}_0\hat{\mathbf{a}}_1(i)]^H\mathbf{x}(i)$ and $\hat{y}_f(i) = [\hat{\mathbf{R}}_{i+n}(i)\hat{\mathbf{a}}_1(i)]^H\mathbf{x}(i)$, which leads to

$$y_f(i) = \eta(i)y_{0f}(i) + (1-\eta(i))\hat{y}_f(i). \tag{25}$$

To restrict $\eta(i)$ to a value greater than 0 and less than 1, a sigmoidal function is employed:

$$\eta(i) = \mathrm{sgm}[\epsilon(i)] = \frac{1}{1+e^{-\epsilon(i)}}, \tag{26}$$

where $\epsilon(i)$ is updated as

$$\begin{aligned}\epsilon(i+1) = \epsilon(i) - \frac{\mu_\epsilon}{(\sigma_\epsilon + q(i))}(\eta(i)|y_{0f}(i) - \hat{y}_f(i)|^2 \\ + \mathcal{R}\{(y_{0f}(i) - \hat{y}_f(i))\hat{y}_f^*(i)\})\eta(i)(1-\eta(i)),\end{aligned} \tag{27}$$

where μ_ϵ is the step size while σ_ϵ is a small positive constant, and $q(i)$ is updated as

$$q(i+1) = \lambda_q(i)(1-\lambda_q)|y_{0f}(i) - \hat{y}_f(i)|^2, \tag{28}$$

where λ_q is a forgetting factor.

Now we resort to an SG adaptive strategy to reduce the complexity required by the matrix inversion. The optimization problem (4) can be re-expressed as

$$\begin{aligned}\underset{\mathbf{w}(i)}{\text{minimize}} \quad & \mathbf{w}^H(i)(\mathbf{x}(i)\mathbf{x}^H(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i) \\ \text{subject to} \quad & \mathbf{w}^H(i)\hat{\mathbf{a}}_1(i) = 1.\end{aligned} \tag{29}$$

Then we can express the SG recursion as

$$\mathbf{w}(i+1) = \mathbf{w}(i) - \mu\frac{\partial\mathcal{L}}{\partial\mathbf{w}(i)}, \tag{30}$$

where $\mathcal{L} = \mathbf{w}^H(i)(\mathbf{x}(i)\mathbf{x}^H(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i) + \lambda(\mathbf{w}^H(i)\hat{\mathbf{a}}_1(i) - 1)$. By substituting $\mathcal{L}$ into the SG equation (30) and letting $\mathbf{w}^H(i+1)\hat{\mathbf{a}}_1(i+1) = 1$, λ is obtained as

$$\lambda = \frac{2(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i) - y(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i))}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)}. \tag{31}$$

By substituting λ back into (30) again, the weight update equation for LOCSME-SG is obtained as

$$\begin{aligned}\mathbf{w}(i+1) = (\mathbf{I} - \mu\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i) \\ - \mu(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i) + y^*(i)(\mathbf{x}(i) - \frac{\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)})).\end{aligned} \tag{32}$$

The adaptive SG recursion circumvents a matrix inversion when computing the weights using (21), which is unavoidable in LOCSME. Therefore, the computational complexity is reduced from $\mathcal{O}(M^3)$ in LOCSME to $\mathcal{O}(M^2)$ in LOCSME-SG. The proposed LOCSME-SG algorithm is summarized in Table I.

TABLE I
PROPOSED LOCSME-SG ALGORITHM

Initialize:
$\mathbf{C} = \int_{\theta_1-\theta_e}^{\theta_1+\theta_e} \mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta$
$[\mathbf{c}_1, \cdots, \mathbf{c}_p]$: p princical eigenvectors of $\mathbf{C}$
$\mathbf{P} = [\mathbf{c}_1, \cdots, \mathbf{c}_p][\mathbf{c}_1, \cdots, \mathbf{c}_p]^H$
$\hat{\mathbf{l}}(0) = \mathbf{0}$; $\mathbf{w}(0) = \mathbf{1}$; $\hat{\rho}(1) = \rho(0) = 1$;
For each snapshot index $i = 1, 2, \cdots$:
$\hat{\mathbf{l}}(i) = \frac{1}{i}\sum_{k=1}^{i}\mathbf{x}(k)y^*(k)$
Steering vector mismatch estimation
$\hat{\nu}(i) = \sum\hat{\mathbf{l}}(i)/M$
$\hat{\mathbf{d}}(i) = \hat{\rho}(i)\hat{\nu}(i) + (1-\hat{\rho}(i))\hat{\mathbf{l}}(i)$
$\hat{\rho}(i) = \frac{(1-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1)+\sum\hat{\mathbf{d}}(i-1)\sum^*\hat{\mathbf{d}}(i-1)}{(i-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1)+(1-\frac{i}{M})\sum\hat{\mathbf{d}}(i-1)\sum^*\hat{\mathbf{d}}(i-1)}$
$\hat{\mathbf{a}}_1(i) = \frac{\mathbf{P}\hat{\mathbf{d}}(i)}{\Vert\mathbf{P}\hat{\mathbf{d}}(i)\Vert_2}$
Desired signal power estimation
$\hat{\sigma}_1^2(i) = \frac{\lvert\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)\rvert^2 - \lvert\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)\rvert\sigma_n^2}{\lvert\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)\rvert^2}$
Computation of INC matrix
$\mathbf{x}_{i+n}(i) = \mathbf{x}(i) - \hat{\sigma}_1(i)\hat{\mathbf{a}}_1(i)$
$\hat{\mathbf{R}}_{i+n}(i) = \mathbf{x}_{i+n}(i)\mathbf{x}_{i+n}^H(i)$
$\breve{\mathbf{R}}_{i+n}(i) = \eta(i)\mathbf{R}_0 + (1-\eta(i))\hat{\mathbf{R}}_{i+n}(i)$
$y_{0f}(i) = [\mathbf{R}_0\hat{\mathbf{a}}_1(i)]^H\mathbf{x}_{i+n}(i)$
$\hat{y}_f(i) = [\hat{\mathbf{R}}_{i+n}(i)\hat{\mathbf{a}}_1(i)]^H\mathbf{x}_{i+n}(i)$
$y_f(i) = \eta(i)y_{0f}(i) + (1-\eta(i))\hat{y}_f(i)$
$\eta(i) = \frac{1}{1+e^{-\epsilon(i)}}$
$\epsilon(i+1) = \epsilon(i) - \frac{\mu_\epsilon}{(\sigma_\epsilon+q(i))}(\eta(i)\lvert y_{0f}(i) - \hat{y}_f(i)\rvert^2$
$+\mathcal{R}\{(y_{0f}(i) - \hat{y}_f(i))\hat{y}_f^*(i)\})\eta(i)(1-\eta(i))$
$q(i+1) = \lambda_q(i)(1-\lambda_q)\lvert y_{0f}(i) - \hat{y}_f(i)\rvert^2$
Computation of beamformer weights
$\mathbf{w}(i+1) = (\mathbf{I} - \mu\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i)$
$-\mu(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i) + y^*(i)(\mathbf{x}(i) - \frac{\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)}))$
End snapshot

B. LOCSME-CCG Adaptive Algorithm

In order to introduce CG-based adaptive algorithms, we specifically divide them into two different algorithms, namely, LOCSME-CCG and its modified version LOCSME-MCG. In the approach of LOCSME-CCG, the SCV $\hat{\mathbf{l}}(i)$ is replaced by an estimate with a forgetting factor λ, which is a constant scalar less than and close to 1 as

$$\hat{\mathbf{l}}(i) = \lambda\hat{\mathbf{l}}(i-1) + \mathbf{x}(i)y^*(i), \tag{33}$$

before we employ it into the vector shrinkage method. The INC matrix is also estimated directly with this forgetting factor as

$$\hat{\mathbf{R}}(i) = \lambda\hat{\mathbf{R}}(i-1) + \mathbf{x}(i)\mathbf{x}^H(i). \tag{34}$$

In order to derive CG-based recursions we need to reformulate the cost function that needs to be minimized as follows

$$\begin{aligned}\underset{\hat{\mathbf{a}}_1(i),\mathbf{v}(i)}{\text{minimize}}\mathcal{J} = \mathbf{v}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{v}(i) \\ - \mathcal{R}\{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)\},\end{aligned} \tag{35}$$

where $\mathbf{v}(i)$ is the CG-based weight vector. In LOCSME-CCG, we require a run of N iterations in each snapshot. In the nth iteration, $\hat{\mathbf{a}}_{1,n}(i)$ and $\mathbf{v}_n(i)$ are updated as follows

$$\hat{\mathbf{a}}_{1,n}(i) = \hat{\mathbf{a}}_{1,n-1}(i) + \alpha_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i), \tag{36}$$

$$\mathbf{v}_n(i) = \mathbf{v}_{n-1}(i) + \alpha_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i), \tag{37}$$

where $\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)$ and $\mathbf{p}_{\mathbf{v},n}(i)$ are direction vectors updated by

$$\mathbf{p}_{\hat{\mathbf{a}}_1,n+1}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) + \beta_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i), \tag{38}$$

$$\mathbf{p}_{\mathbf{v},n+1}(i) = \mathbf{g}_{\mathbf{v},n}(i) + \beta_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i), \tag{39}$$

where $\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)$ and $\mathbf{g}_{\mathbf{v},n}(i)$ are the negative gradients of the cost function in terms of $\hat{\mathbf{a}}_1(i)$ and $\mathbf{v}(i)$, respectively, which are expressed as

$$\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) = -\frac{\partial \mathcal{J}}{\partial \hat{\mathbf{a}}_{1,n}(i)} = \hat{\sigma}_1^2(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\hat{\mathbf{a}}_{1,n}(i) + \mathbf{v}_n(i), \tag{40}$$

$$\begin{aligned}\mathbf{g}_{\mathbf{v},n}(i) &= -\frac{\partial \mathcal{J}}{\partial \mathbf{v}_n(i)} \\ &= \mathbf{g}_{\mathbf{v},n-1}(i) - \alpha_{\mathbf{v},n}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\mathbf{x}(i)\mathbf{x}^H(i))\mathbf{p}_{\mathbf{v},n}(i).\end{aligned} \tag{41}$$

The scaling parameters $\alpha_{\hat{\mathbf{a}}_1,n}(i)$, $\alpha_{\mathbf{v},n}(i)$ can be obtained by substituting (36) and (37) into (35) and minimizing with respect to $\alpha_{\hat{\mathbf{a}}_1,n}(i)$ and $\alpha_{\mathbf{v},n}(i)$, respectively. The solutions are given by

$$\alpha_{\hat{\mathbf{a}}_1,n}(i) = -\frac{\mathbf{g}^H_{\hat{\mathbf{a}}_1,n-1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}{\hat{\sigma}_1^2(i)\mathbf{p}^H_{\hat{\mathbf{a}}_1,n}(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}, \tag{42}$$

$$\alpha_{\mathbf{v},n}(i) = \frac{\mathbf{g}^H_{\mathbf{v},n-1}(i)\mathbf{p}_{\mathbf{v},n}(i)}{\mathbf{p}^H_{\mathbf{v},n}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_{1,n}(i)\hat{\mathbf{a}}^H_{1,n}(i))\mathbf{p}_{\mathbf{v},n}(i)}. \tag{43}$$

The parameters $\beta_{\hat{\mathbf{a}}_1,n}(i)$ and $\beta_{\mathbf{v},n}(i)$ should be chosen to provide conjugacy for direction vectors [17], [18] which results in

$$\beta_{\hat{\mathbf{a}}_1,n}(i) = \frac{\mathbf{g}^H_{\hat{\mathbf{a}}_1,n}(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)}{\mathbf{g}^H_{\hat{\mathbf{a}}_1,n-1}(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}(i)}, \tag{44}$$

$$\beta_{\mathbf{v},n}(i) = \frac{\mathbf{g}^H_{\mathbf{v},n}(i)\mathbf{g}_{\mathbf{v},n}(i)}{\mathbf{g}^H_{\mathbf{v},n-1}(i)\mathbf{g}_{\mathbf{v},n-1}(i)}. \tag{45}$$

After $\hat{\mathbf{a}}_{1,n}(i)$ and $\mathbf{v}_n(i)$ are updated for N iterations, the beamforming weight vector $\mathbf{w}(i)$ can be computed by

$$\mathbf{w}(i) = \frac{\mathbf{v}_N(i)}{\hat{\mathbf{a}}^H_{1,N}(i)\mathbf{v}_N(i)}, \tag{46}$$

while the estimated steering vector is also updated to $\hat{\mathbf{a}}_{1,N}(i)$. Table II summarizes the LOCSME-CCG algorithm.

C. LOCSME-MCG Adaptive Algorithm

In LOCSME-MCG, we let only one iteration be performed per snapshot[17], [18], which further reduces the complexity compared to LOCSME-CCG. Here we denote the CG-based weights and steering vector updated by snapshots rather than inner iterations as

$$\hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1(i-1) + \alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i), \tag{47}$$

$$\mathbf{v}(i) = \mathbf{v}(i-1) + \alpha_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i). \tag{48}$$

TABLE II
PROPOSED LOCSME-CCG ALGORITHM

Initialize:
$\mathbf{C} = \int_{\theta_1-\theta_e}^{\theta_1+\theta_e} \mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta$
$[\mathbf{c}_1, \cdots, \mathbf{c}_p]$: p princical eigenvectors of $\mathbf{C}$
$\mathbf{P} = [\mathbf{c}_1, \cdots, \mathbf{c}_p][\mathbf{c}_1, \cdots, \mathbf{c}_p]^H$
$\hat{\mathbf{l}}(0) = \mathbf{0}$; $\hat{\mathbf{R}}(0) = \mathbf{I}$; $\mathbf{w}(1) = \mathbf{v}_0(1) = \mathbf{1}$; $\hat{\rho}(1) = \rho(0) = 1$; $\lambda = 0.98$;

For each snapshot index $i = 1, 2, \cdots$:
$\hat{\mathbf{l}}(i) = \lambda\hat{\mathbf{l}}(i-1) + \mathbf{x}(i)y^*(i)$
$\hat{\mathbf{R}}(i) = \lambda\hat{\mathbf{R}}(i-1) + \mathbf{x}(i)\mathbf{x}^H(i)$

Steering vector mismatch estimation
$\hat{\nu}(i) = \sum \hat{\mathbf{l}}(i)/M$
$\hat{\mathbf{d}}(i) = \hat{\rho}(i)\hat{\nu}(i) + (1-\hat{\rho}(i))\hat{\mathbf{l}}(i)$
$\hat{\rho}(i) = \frac{(1-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1) + \sum \hat{\mathbf{d}}(i-1)\sum^* \hat{\mathbf{d}}(i-1)}{(i-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1) + (1-\frac{i}{M})\sum \hat{\mathbf{d}}(i-1)\sum^* \hat{\mathbf{d}}(i-1)}$
$\hat{\mathbf{a}}_1(i) = \frac{\mathbf{P}\hat{\mathbf{d}}(i)}{\|\mathbf{P}\hat{\mathbf{d}}(i)\|_2}$

Desired signal power estimation
$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2 - |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\sigma_n^2}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}$

CCG-based estimations of steering vector mismatch and beamformer weights
$\hat{\mathbf{a}}_{1,0}(i) = \hat{\mathbf{a}}_1(i)$
$\mathbf{g}_{\hat{\mathbf{a}}_1,0}(i) = \hat{\sigma}_1^2(i)\mathbf{v}_0(i)\mathbf{v}_0^H(i)\hat{\mathbf{a}}_{1,0}(i) + \mathbf{v}_0(i)$
$\mathbf{g}_{\mathbf{v},0}(i) = \hat{\mathbf{a}}_{1,0}(i) - \hat{\mathbf{R}}(i)\mathbf{v}_0(i)$
$\mathbf{p}_{\hat{\mathbf{a}}_1,0}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,0}(i)$; $\mathbf{p}_{\mathbf{v},0}(i) = \mathbf{g}_{\mathbf{v},0}(i)$

For each iteration index $n = 1, 2, \cdots, N$:
$\alpha_{\hat{\mathbf{a}}_1,n}(i) = -\frac{\mathbf{g}^H_{\hat{\mathbf{a}}_1,n-1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}{\hat{\sigma}_1^2(i)\mathbf{p}^H_{\hat{\mathbf{a}}_1,n}(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)}$
$\alpha_{\mathbf{v},n}(i) = \frac{\mathbf{g}^H_{\mathbf{v},n-1}(i)\mathbf{p}_{\mathbf{v},n}(i)}{\mathbf{p}^H_{\mathbf{v},n}(i)(\hat{\mathbf{R}}(i)-\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_{1,n}(i)\hat{\mathbf{a}}^H_{1,n}(i))\mathbf{p}_{\mathbf{v},n}(i)}$
$\hat{\mathbf{a}}_{1,n}(i) = \hat{\mathbf{a}}_{1,n-1}(i) + \alpha_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)$
$\mathbf{v}_n(i) = \mathbf{v}_{n-1}(i) + \alpha_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i)$
$\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) = \hat{\sigma}_1^2(i)\mathbf{v}_n(i)\mathbf{v}_n^H(i)\hat{\mathbf{a}}_{1,n}(i) + \mathbf{v}_n(i)$
$\mathbf{g}_{\mathbf{v},n}(i) = \mathbf{g}_{\mathbf{v},n-1}(i) - \alpha_{\mathbf{v},n}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\mathbf{x}(i)\mathbf{x}^H(i))\mathbf{p}_{\mathbf{v},n}(i)$
$\beta_{\hat{\mathbf{a}}_1,n}(i) = \frac{\mathbf{g}^H_{\hat{\mathbf{a}}_1,n}(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i)}{\mathbf{g}^H_{\hat{\mathbf{a}}_1,n-1}(i)\mathbf{g}_{\hat{\mathbf{a}}_1,n-1}(i)}$
$\beta_{\mathbf{v},n}(i) = \frac{\mathbf{g}^H_{\mathbf{v},n}(i)\mathbf{g}_{\mathbf{v},n}(i)}{\mathbf{g}^H_{\mathbf{v},n-1}(i)\mathbf{g}_{\mathbf{v},n-1}(i)}$
$\mathbf{p}_{\hat{\mathbf{a}}_1,n+1}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) + \beta_{\hat{\mathbf{a}}_1,n}(i)\mathbf{p}_{\hat{\mathbf{a}}_1,n}(i)$
$\mathbf{p}_{\mathbf{v},n+1}(i) = \mathbf{g}_{\mathbf{v},n}(i) + \beta_{\mathbf{v},n}(i)\mathbf{p}_{\mathbf{v},n}(i)$
End iteration

computation of beamformer weights
$\mathbf{v}_0(i+1) = \mathbf{v}_N(i)$
$\mathbf{w}(i) = \frac{\mathbf{v}_N(i)}{\hat{\mathbf{a}}^H_{1,N}(i)\mathbf{v}_N(i)}$
End snapshot

As can be seen, the subscripts of all the quantities for inner iterations are eliminated. Then, we employ the degenerated scheme to ensure $\alpha_{\hat{\mathbf{a}}_1}(i)$ and $\alpha_{\mathbf{v}}(i)$ satisfy the convergence bound [17] given by

$$0 \leq \mathbf{p}^H_{\hat{\mathbf{a}}_1}(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i) \leq 0.5\mathbf{p}^H_{\hat{\mathbf{a}}_1}(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1), \tag{49}$$

$$0 \leq \mathbf{p}^H_{\mathbf{v}}(i)\mathbf{g}_{\mathbf{v}}(i) \leq 0.5\mathbf{p}^H_{\mathbf{v}}(i)\mathbf{g}_{\mathbf{v}}(i-1). \tag{50}$$

Instead of updating the negative gradient vectors $\mathbf{g}_{\hat{\mathbf{a}}_1}(i)$ and $\mathbf{g}_{\mathbf{v}}(i)$ in iterations, now we utilize the forgetting factor to reexpress them in one snapshot as

$$\mathbf{g}_{\hat{\mathbf{a}}_1}(i) = (1-\lambda)\mathbf{v}(i) + \lambda\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1) + \hat{\sigma}_1^2(i)\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i), \quad (51)$$

$$\mathbf{g}_{\mathbf{v}}(i) = (1-\lambda)\hat{\mathbf{a}}_1(i) + \lambda\mathbf{g}_{\mathbf{v}}(i-1) - \alpha_{\mathbf{v}}(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{v}(i-1). \quad (52)$$

Pre-multiplying (51) and (52) by $\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)$ and $\mathbf{p}_{\mathbf{v}}^H(i)$, respectively, and taking expectations we obtain

$$E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i)] = E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)(\mathbf{v}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1)(i)] + \lambda E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)] - \lambda E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)] + E[\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\hat{\sigma}_1^2(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)], \quad (53)$$

$$E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i)] = \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)] - E[\alpha_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)], \quad (54)$$

where in (54) we have $E[\hat{\mathbf{R}}(i)\mathbf{v}(i-1)] = E[\hat{\mathbf{a}}_1(i)]$. After substituting (54) back into (50) we obtain the bounds for $\alpha_{\mathbf{v}}(i)$ as follows

$$\frac{(\lambda-0.5)E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)]}{E[\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)]} \leq E[\alpha_{\mathbf{v}}(i)] \leq \frac{\lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)] - \lambda E[\mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)]}{E[\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)]}. \quad (55)$$

Then we can introduce a constant parameter $\eta_{\mathbf{v}} \in [0, 0.5]$ to restrict $\alpha_{\mathbf{v}}(i)$ within the bounds in (55) as

$$\alpha_{\mathbf{v}}(i) = \frac{\lambda(\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1) - \mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)) - \eta_{\mathbf{v}}\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)}{\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)}. \quad (56)$$

Similarly, we can also obtain the bounds for $\alpha_{\hat{\mathbf{a}}_1}(i)$. For simplicity let us define $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)] = A$, $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)] = B$, $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i)] = C$ and $E[\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\hat{\sigma}_1^2(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)] = D$. Substituting equation (53) into (49) gives

$$\frac{\lambda(B-A) - B + C}{D} \leq E[\alpha_{\hat{\mathbf{a}}_1}(i)] \leq \frac{\lambda(B-A) - B + C + 0.5A}{D}, \quad (57)$$

in which we can introduce another constant parameter $\eta_{\hat{\mathbf{a}}_1} \in [0, 0.5]$ to restrict $\alpha_{\hat{\mathbf{a}}_1}(i)$ within the bounds in (57) as

$$E[\alpha_{\hat{\mathbf{a}}_1}(i)] = \frac{\lambda(B-A) - B + C + \eta_{\hat{\mathbf{a}}_1}A}{D}, \quad (58)$$

or

$$\alpha_{\hat{\mathbf{a}}_1}(i) = [\lambda(\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) + \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i) + \eta_{\hat{\mathbf{a}}_1}\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)] / [\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)]. \quad (59)$$

Then we can update the direction vectors $\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$ and $\mathbf{p}_{\mathbf{v}}(i)$ by

$$\mathbf{p}_{\hat{\mathbf{a}}_1}(i+1) = \mathbf{g}_{\hat{\mathbf{a}}_1}(i) + \beta_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i), \quad (60)$$

$$\mathbf{p}_{\mathbf{v}}(i+1) = \mathbf{g}_{\mathbf{v}}(i) + \beta_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i), \quad (61)$$

where $\beta_{\hat{\mathbf{a}}_1}(i)$ and $\beta_{\mathbf{v}}(i)$ are updated by

$$\beta_{\hat{\mathbf{a}}_1}(i) = \frac{[\mathbf{g}_{\hat{\mathbf{a}}_1}(i) - \mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]^H\mathbf{g}_{\hat{\mathbf{a}}_1}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1}^H(i-1)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)}, \quad (62)$$

$$\beta_{\mathbf{v}}(i) = \frac{[\mathbf{g}_{\mathbf{v}}(i) - \mathbf{g}_{\mathbf{v}}(i-1)]^H\mathbf{g}_{\mathbf{v}}(i)}{\mathbf{g}_{\mathbf{v}}^H(i-1)\mathbf{g}_{\mathbf{v}}(i-1)}. \quad (63)$$

Finally we can update the beamforming weights by

$$\mathbf{w}(i) = \frac{\mathbf{v}(i)}{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)}, \quad (64)$$

The LOCSME-MCG algorithm is summarized in Table III. The MCG approach employs the forgetting factor λ and constant η for estimating $\alpha(i)$, which means its performance may depend on a suitable choice of these parameters. However, it requires much lower complexity for the elimination of inner recursions compared to CCG and presents a similar performance in the simulations.

V. Analysis: Shrinkage and Complexity

This section investigates the effects of shrinkage approaches and the computational complexity of the proposed algorithms. Firstly we rewrite the vector shrinkage recursions into a matrix shrinkage recursion. Then we employ an eigen-decomposition approach to examine the eigenvalues dispersion for the vector shrinkage and matrix shrinkage cases by exploring the MSE [3] of their eigenvalues, and give reasons why shrinkage gives an important contribution to the performance. Then we present a complexity analysis for the proposed algorithms and comparisons to the existing RAB algorithms. It is clear that the proposed algorithms achieve one degree lower complexity than most of the existing ones.

A. Effects of Shrinkage

First of all, we modify the vector shrinkage formula (10) to the following full rank matrix form

$$\hat{\mathbf{D}}(i) = \hat{\rho}(i)\hat{\mathbf{V}}(i) + (1-\hat{\rho}(i))\hat{\mathbf{L}}(i), \quad (65)$$

where $\hat{\mathbf{V}}(i)$, $\hat{\mathbf{D}}(i)$ and $\hat{\mathbf{L}}(i)$ are all diagonal matrix, having each of their diagonal entries identical to $\hat{\nu}(i)$, elements of the optimal shrinkage estimator $\hat{\mathbf{d}}(i)$ and elements of the SCV $\hat{\mathbf{l}}(i)$, respectively, whereas all the three matrices have their other entries equal to zero. Associated with (18), it can be seen they share the same linear shrinkage formula. Now, we carry out eigenvalue decompositions for every matrix in (10). Since the eigenvalues of a diagonal matrix are simply its diagonal entries, the eigenvalues of $\hat{\mathbf{D}}(i)$, $\hat{\mathbf{V}}(i)$ and $\hat{\mathbf{L}}(i)$ can be expressed as

$$\{\hat{d}_1(i), \cdots, \hat{d}_M(i)\}, \quad (66)$$

$$\{\hat{\nu}(i), \cdots, \hat{\nu}(i)\}, \quad (67)$$

TABLE III

PROPOSED LOCSME-MCG ALGORITHM

Initialize:

$\mathbf{C} = \int_{\theta_1-\theta_e}^{\theta_1+\theta_e} \mathbf{a}(\theta)\mathbf{a}^H(\theta)d\theta$

$[\mathbf{c}_1, \cdots, \mathbf{c}_p]$: p princical eigenvectors of $\mathbf{C}$

$\mathbf{P} = [\mathbf{c}_1, \cdots, \mathbf{c}_p][\mathbf{c}_1, \cdots, \mathbf{c}_p]^H$

$\hat{\mathbf{l}}(0) = \mathbf{0}$; $\hat{\mathbf{R}}(0) = \mathbf{I}$; $\mathbf{w}(1) = \mathbf{v}(0) = \mathbf{1}$; $\hat{\rho}(1) = \rho(0) = 1$;

$\lambda = 0.95$; $\eta_{\mathbf{v}} = \eta_{\hat{\mathbf{a}}_1} = 0.1$;

$\mathbf{g}_{\mathbf{v}}(0) = \mathbf{p}_{\mathbf{v}}(1) = \hat{\mathbf{R}}(0)\mathbf{v}(1)$; $\mathbf{g}_{\hat{\mathbf{a}}_1}(0) = \mathbf{p}_{\hat{\mathbf{a}}_1}(1) = \mathbf{v}(0)$;

For each snapshot index $i = 1, 2, \cdots$:

$\hat{\mathbf{l}}(i) = \lambda\hat{\mathbf{l}}(i-1) + \mathbf{x}(i)y^*(i)$

$\hat{\mathbf{R}}(i) = \lambda\hat{\mathbf{R}}(i-1) + \mathbf{x}(i)\mathbf{x}^H(i)$

Steering vector mismatch estimation

$\hat{\nu}(i) = \sum \hat{\mathbf{l}}(i)/M$

$\hat{\mathbf{d}}(i) = \hat{\rho}(i)\hat{\nu}(i) + (1-\hat{\rho}(i))\hat{\mathbf{l}}(i)$

$\hat{\rho}(i) = \frac{(1-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1) + \sum \hat{\mathbf{d}}(i-1) \sum^* \hat{\mathbf{d}}(i-1)}{(i-\frac{2}{M})\hat{\mathbf{d}}^H(i-1)\hat{\mathbf{l}}(i-1) + (1-\frac{i}{M}) \sum \hat{\mathbf{d}}(i-1) \sum^* \hat{\mathbf{d}}(i-1)}$

$\hat{\mathbf{a}}_1(i) = \frac{\mathbf{P}\hat{\mathbf{d}}(i)}{\|\mathbf{P}\hat{\mathbf{d}}(i)\|_2}$

Desired signal power estimation

$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2 - |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\sigma_n^2}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}$

MCG-based estimations of steering vector mismatch and beamformer weights

$\alpha_{\hat{\mathbf{a}}_1}(i) = [\lambda(\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)$

$+\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i) + \eta_{\hat{\mathbf{a}}_1}\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]$

$/[\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)]$

$\alpha_{\mathbf{v}}(i) = \frac{\lambda(\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1) - \mathbf{p}_{\mathbf{v}}^H(i)\hat{\mathbf{a}}_1(i)) - \eta_{\mathbf{v}}\mathbf{p}_{\mathbf{v}}^H(i)\mathbf{g}_{\mathbf{v}}(i-1)}{\mathbf{p}_{\mathbf{v}}^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i)}$

$\hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1(i-1) + \alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$

$\mathbf{v}(i) = \mathbf{v}(i-1) + \alpha_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i)$

$\mathbf{g}_{\hat{\mathbf{a}}_1}(i) = (1-\lambda)\mathbf{v}(i) + \lambda\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)$

$+\hat{\sigma}_1^2(i)\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i)$

$\mathbf{g}_{\mathbf{v}}(i) = (1-\lambda)\hat{\mathbf{a}}_1(i) + \lambda\mathbf{g}_{\mathbf{v}}(i-1) - \alpha_{\mathbf{v}}(i)(\hat{\mathbf{R}}(i)$

$-\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_{\mathbf{v}}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{v}(i-1)$

$\beta_{\hat{\mathbf{a}}_1}(i) = \frac{[\mathbf{g}_{\hat{\mathbf{a}}_1}(i) - \mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]^H\mathbf{g}_{\hat{\mathbf{a}}_1}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1}^H(i-1)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)}$

$\beta_{\mathbf{v}}(i) = \frac{[\mathbf{g}_{\mathbf{v}}(i) - \mathbf{g}_{\mathbf{v}}(i-1)]^H\mathbf{g}_{\mathbf{v}}(i)}{\mathbf{g}_{\mathbf{v}}^H(i-1)\mathbf{g}_{\mathbf{v}}(i-1)}$

$\mathbf{p}_{\hat{\mathbf{a}}_1}(i+1) = \mathbf{g}_{\hat{\mathbf{a}}_1}(i) + \beta_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$

$\mathbf{p}_{\mathbf{v}}(i+1) = \mathbf{g}_{\mathbf{v}}(i) + \beta_{\mathbf{v}}(i)\mathbf{p}_{\mathbf{v}}(i)$

Computation of beamformer weights

$\mathbf{w}(i) = \frac{\mathbf{v}(i)}{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)}$

End snapshot

$$\{\hat{l}_1(i), \cdots, \hat{l}_M(i)\}, \tag{68}$$

respectively. Since we have

$$\begin{aligned} E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{V}}(i)\|^2] &= E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1) + \hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\|^2] \\ &= E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)\|^2] + E[\|\hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\|^2] \\ &\quad + 2E[\langle \hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1), \hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\rangle] \\ &= E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)\|^2] + \|\hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\|^2 \\ &\quad + 2\langle E[\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)], \hat{\mathbf{D}}(i-1) - \hat{\mathbf{V}}(i)\rangle, \end{aligned} \tag{69}$$

where $\langle , \rangle$ denotes the inner product and we have $E[\hat{\mathbf{L}}(i)] = \hat{\mathbf{D}}(i-1)$, then the inner product term in the above equation equals 0, which yields the following

$$E[\|\hat{\mathbf{L}}(i) - \hat{\nu}(i)\mathbf{I}\|^2] - \|\hat{\mathbf{D}}(i-1) - \hat{\nu}(i)\mathbf{I}\|^2 = E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)\|^2]. \tag{70}$$

Equation (70) can be interpreted in terms of the eigenvalues of the matrices if we rewrite it as

$$E[\frac{1}{M}\sum_{m=1}^{M}(\hat{l}_m(i) - \hat{\nu}(i))^2] - \frac{1}{M}\sum_{m=1}^{M}(\hat{d}_m(i-1) - \hat{\nu}(i))^2 = E[\|\hat{\mathbf{L}}(i) - \hat{\mathbf{D}}(i-1)\|^2]. \tag{71}$$

Note that in (71), $\hat{\nu}(i)$ actually represents the mean value of the SCV $\hat{\mathbf{l}}(i)$ or the diagonal entries of matrix $\hat{\mathbf{V}}(i)$. Similarly to the matrix shrinkage in (18), we can process the same analysis even though the matrices are no longer diagonal but will lead to a more general result. Assuming the eigenvalues of the matrices $\tilde{\mathbf{R}}(i)$, $\hat{\mathbf{F}}_0(i)$, and $\hat{\mathbf{R}}(i)$ are

$$\{\lambda_1(i), \cdots, \lambda_M(i)\}, \tag{72}$$

$$\{f_1(i), \cdots, f_M(i)\}, \tag{73}$$

$$\{\gamma_1(i), \cdots, \gamma_M(i)\}, \tag{74}$$

respectively. Then we have

$$\begin{aligned} E[\|\hat{\mathbf{R}}(i) - \hat{\mathbf{F}}_0(i)\|^2] &= E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1) + \tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\|^2] \\ &= E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2] + E[\|\tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\|^2] \\ &\quad + 2E[\langle \hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1), \tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\rangle] \\ &= E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2] + \|\tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\|^2 \\ &\quad + 2\langle E[\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)], \tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\rangle, \end{aligned} \tag{75}$$

where the inner product term equals 0 because of $E[\hat{\mathbf{R}}(i)] = \tilde{\mathbf{R}}(i-1)$, which results in

$$E[\|\hat{\mathbf{R}}(i) - \hat{\mathbf{F}}_0(i)\|^2] - \|\tilde{\mathbf{R}}(i-1) - \hat{\mathbf{F}}_0(i)\|^2 = E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2]. \tag{76}$$

Noting that $\hat{\mathbf{F}}_0(i) = \hat{\nu}_0(i)\mathbf{I}$, then (76) is equivalent to

$$E[\|\hat{\mathbf{R}}(i) - \hat{\nu}_0(i)\mathbf{I}\|^2] - \|\tilde{\mathbf{R}}(i-1) - \hat{\nu}_0(i)\mathbf{I}\|^2 = E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2], \tag{77}$$

which can be rewritten in an alternative form as

$$E[\frac{1}{M}\sum_{m=1}^{M}(\gamma_m(i) - \hat{\nu}_0(i))^2] - \frac{1}{M}\sum_{m=1}^{M}(\lambda_m(i-1) - \hat{\nu}_0(i))^2 = E[\|\hat{\mathbf{R}}(i) - \tilde{\mathbf{R}}(i-1)\|^2]. \tag{78}$$

Because the expectation on the right hand side of equation (71) and (78) are always non-negative, so we have their left hand side always equal or larger than 0, which yields

$$E[\frac{1}{M}\sum_{m=1}^{M}(\hat{l}_m(i)-\hat{\nu}(i))^2] \geq \frac{1}{M}\sum_{m=1}^{M}(\hat{d}_m(i-1)-\hat{\nu}(i))^2, \quad (79)$$

$$E[\frac{1}{M}\sum_{m=1}^{M}(\gamma_m(i)-\hat{\nu}_0(i))^2] \geq \frac{1}{M}\sum_{m=1}^{M}(\lambda_m(i-1)-\hat{\nu}_0(i))^2. \quad (80)$$

Since we also know that

$$E[\hat{\nu}(i)] = \frac{1}{M}\sum_{m=1}^{M}\hat{d}_m(i-1), \quad (81)$$

$$E[\hat{\nu}_0(i)] = \frac{1}{M}\sum_{m=1}^{M}\lambda_m(i-1), \quad (82)$$

which express the expected mean of the eigenvalues of the sampled matrix $\hat{\mathbf{L}}(i)$ and $\hat{\mathbf{R}}(i)$ in snapshot i, respectively. Then equations (79) and (80) indicate that the expected MSE of the eigenvalues of $\hat{\mathbf{L}}(i)$ or $\hat{\mathbf{R}}(i)$ in snapshot i is always larger or equal to those of the optimal shrinkage estimator $\hat{\mathbf{D}}(i-1)$ or $\tilde{\mathbf{R}}(i-1)$ obtained from the previous snapshot. In other words, the eigenvalues of the sampled matrix are more dispersedly distributed (here we should have $\hat{d}_1(i-1) > \hat{l}_1(i) > 0$, $\hat{d}_m(i-1) < \hat{l}_m(i)$ and $\lambda_1(i-1) > \gamma_1(i) > 0$, $\lambda_m(i-1) < \gamma_m(i)$) based on their expected mean value than those of the optimal shrinkage estimator from the last snapshot. Shrinking the sampled matrix to a matrix with less dispersed eigenvalues can lead to an improved covariance matrix estimator as reported in [13].

B. Complexity Analysis

In this part, we analyze the computational complexity in terms of flops (total number of additions and multiplications) required by the proposed RAB algorithms. The proposed RAB algorithms avoid costly matrix inversion and multiplication procedures, which are unavoidable in the existing RAB algorithms. The complexity comparison among different algorithms are listed in Table IV. It should be noted that LOCSME-CCG has its complexity dependent on the number of inner iterations N, which can be properly selected within the range of $5-10$. However, the LCWC algorithm of [15] also requires N inner iterations per snapshots, which significantly varies in different snapshots and is usually much larger than the value of N in the proposed LOCSME-CCG algorithm. It is clear that our proposed algorithms have one degree lower complexity in terms of the number of sensors M, which are dominated by $\mathcal{O}(M^2)$, resulting in great advantages when M is large. Fig. 1 gives illustrations of the complexity comparison of the listed algorithms, where the values of N for [15] and the proposed LOCSME-CCG are selected as 50 and 10, respectively.

TABLE IV
COMPLEXITY COMPARISON

RAB Algorithms	Flops
LOCSME [14]	$4M^3+3M^2+20M$
RCB [5]	$2M^3+11M^2$
Algorithm of [10]	$M^{3.5}+7M^3+5M^2+3M$
LOCME [9]	$2M^3+4M^2+5M$
LCWC [15]	$N(2M^2+7M)$
LOCSME-SG	$15M^2+30M$
LOCSME-CCG	$5M^2+21M+N(8M^2+32M)$
LOCSME-MCG	$13M^2+77M$

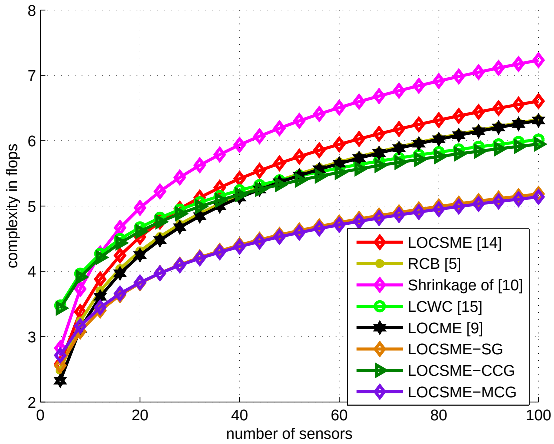

Fig. 1. complexity versus number of sensors

VI. SIMULATION RESULTS

The simulations are carried out under both coherent and incoherent local scattering mismatch [6] scenarios. A uniform linear array (ULA) of $M = 12$ omnidirectional sensors with half wavelength spacing is considered. 100 repetitions are executed to obtain each point of the curves and a maximum of $i = 300$ snapshots are observed. The desired signal is assumed to arrive at $\theta_1 = 10°$ while there are other two interferers impinging on the antenna array from directions $\theta_2 = 30°$ and $\theta_3 = 50°$. The signal-to-interference ratio (SIR) is fixed at 0dB. For the optimum scenario in each of the comparisons, we do not consider the existences of interferers and assume the DoA of the desired signal is known. However, the noise is still considered. For our proposed algorithms, the angular sector in which the desired signal is assumed to be located is chosen as $[\theta_1 - 5°, \theta_1 + 5°]$ and the number of eigenvectors of the subspace projection matrix p is selected manually with the help of simulations. The results focus on the beamformer output SINR performance versus the number of snapshots, or a variation of input SNR (-10dB to 30dB).

A. Mismatch due to Coherent Local Scattering

The steering vector of the desired signal affected by a time-invariant coherent local scattering effect is modeled as

$$\mathbf{a}_1 = \mathbf{p} + \sum_{k=1}^{4} e^{j\varphi_k}\mathbf{b}(\theta_k), \quad (83)$$

where $\mathbf{p}$ corresponds to the direct path while $\mathbf{b}(\theta_k)(k = 1,2,3,4)$ corresponds to the scattered paths. The angles $\theta_k(k = 1,2,3,4)$ are randomly and independently drawn in each simulation run from a uniform generator with mean $10°$ and standard deviation 2°. The angles $\varphi_k (k = 1, 2, 3, 4)$ are independently and uniformly taken from the interval $[0, 2\pi]$ in each simulation run. Notice that θ_k and φ_k change from trials while remaining constant over snapshots.

Fig. 2 and Fig. 3 illustrate the performance comparisons of SINR versus snapshots and SINR versus SNR, respectively, in terms of the mentioned RAB algorithms in the last section under coherent scattering case. Specifically to obtain Fig. 2, we assume the noise power is known and select $\mu = 0.2$, $\mu_\epsilon = 1$, $\sigma_\epsilon = 0.001$, $\lambda_q = 0.99$, $\mathbf{R}_0 = 10\mathbf{I}$ for LOCSME-SG, $\lambda = 0.95$ for LOCSME-CCG and $\lambda = 0.95$, $\eta = 0.2$ for LOCSME-MCG. However, selection of these parameters may vary according to different input SNR as in Fig. 3. The proposed algorithms outperform the other algorithms and are very close to the standard LOCSME, especially for LOCSME-CCG and LOCSME-MCG.

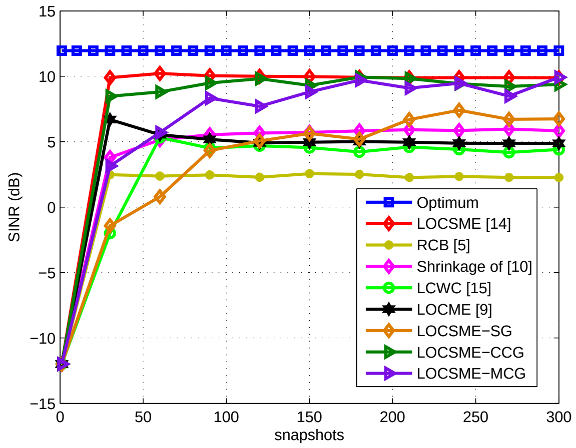

Fig. 2. coherent local scattering, SINR versus snapshots

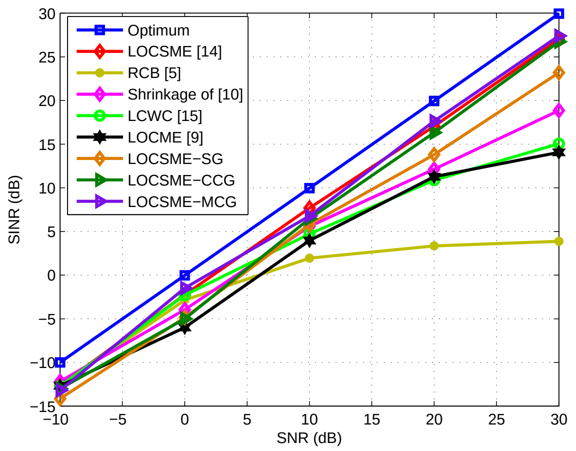

Fig. 3. coherent local scattering, SINR versus SNR

In Fig. 4, we use an ML-based method to estimate the noise power in LOCSME, LOCSME-SG, LOCSME-CCG and LOCSME-MCG in the same scenario of Fig. 2. It is clear that no noticeable differences between their performance can be observed by comparing Fig. 2 and Fig. 4.

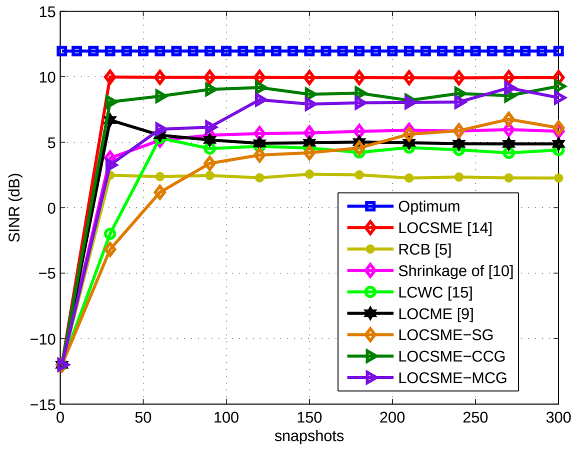

Fig. 4. coherent local scattering, SINR versus snapshots

B. Mismatch due to Incoherent Local Scattering

In the incoherent local scattering case, the desired signal has a time-varying signature and the steering vector is modeled by

$$\mathbf{a}_1(i) = s_0(i)\mathbf{p} + \sum_{k=1}^{4} s_k(i)\mathbf{b}(\theta_k), \qquad (84)$$

where $s_k(i)(k = 0, 1, 2, 3, 4)$ are i.i.d zero mean complex Gaussian random variables independently drawn from a random generator. The angles $\theta_k (k = 0, 1, 2, 3, 4)$ are drawn independently in each simulation run from a uniform generator with mean 10° and standard deviation 2°. This time, $s_k(i)$ changes both from run to run and from snapshot to snapshot.

Fig. 5 and Fig. 6 illustrate the performance comparisons of SINR versus snapshots and SINR versus SNR, respectively, in terms of the mentioned RAB algorithms in the last section under incoherent scattering case. To obtain Fig. 5, we select $\mu = 0.1$, $\mu_\epsilon = 5$, $\sigma_\epsilon = 0.001$, $\lambda_q = 0.99$, $\mathbf{R}_0 = 50\mathbf{I}$ for LOCSME-SG, $\lambda = 0.99$ for LOCSME-CCG and $\lambda = 0.95$, $\eta = 0.3$ for LOCSME-MCG. However, we have optimized the parameters to give the best possible performance at different input SNRs.

Different from the coherent scattering results, all the algorithms have a certain level of performance degradation due to the effect of incoherent local scattering model, in which case we have the extra system dynamics with the time variation, contributing to more environmental uncertainties in the system. However, over a wide range of input SNR values, the proposed algorithms are still able to outperform the other RAB algorithms. One point that needs to be emphasized is, most of the existing RAB algorithms experience significant performance degradation when the input SNR is high (i.e. around or more than 20dB), which is explained in [11] that the desired signal always presents in any kind of diagonal loading technique. However, the proposed algorithms have improved the estimation accuracy, so that the high SNR degradation is successfully avoided as can be seen in Fig. 5 and Fig. 6.

We assess the SINR performance versus snapshots of the selected algorithms in a specific time-varying scenario which encounters moving source signals that are captured with different

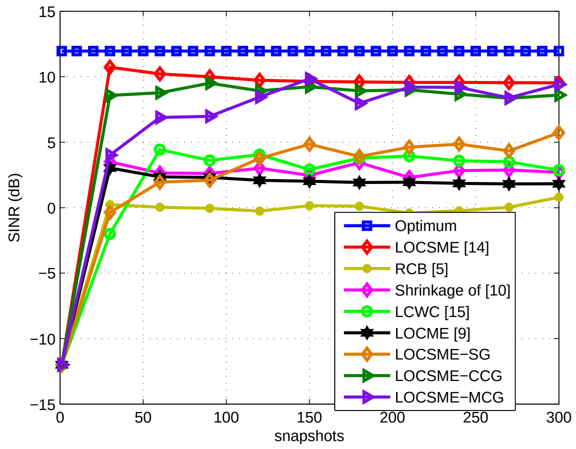

Fig. 5. incoherent local scattering, SINR versus snapshots

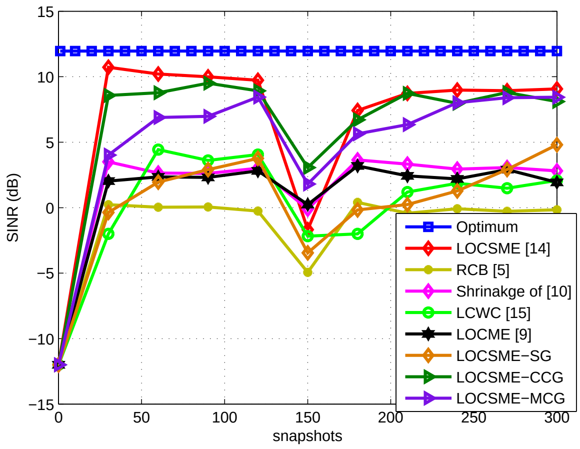

Fig. 7. incoherent local scattering, time-varying scenario

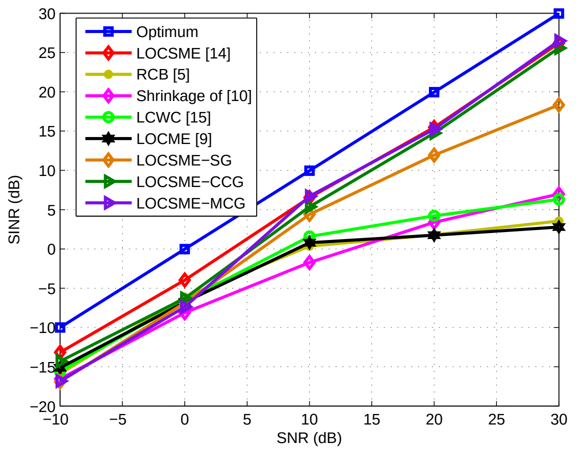

Fig. 6. incoherent local scattering, SINR versus SNR

DoAs in certain snapshots as introduced in Table V. The result of this scenario is shown in Fig. 7.

In addition, it should also be emphasized that performance comparisons with the conventional adaptive algorithms (i.e. SG, CCG or MCG without combined to LOCSME) are not included, as they are not recognized as RAB algorithms and have much worse performance in the presence of uncertainties. Actually, as mentioned in the introduction, it has already been shown that conventional adaptive beamforming algorithms are extremely sensitive to the statistical characteristics of the sampled data (i.e. data size and data accuracy). Especially, when these algorithms suffer environment uncertainties (i.e. steering vector mismatch), significant further performance degradation is unavoidable.

TABLE V

CHANGES OF INTERFERERS

Snapshots	DoAs
$0-150$	$\theta_1=10°, \theta_2=30°, \theta_3=50°.$
$150-300$	$\theta_1=15°, \theta_2=25°, \theta_3=35°.$

VII. CONCLUSION

This work proposed low-complexity adaptive RAB algorithms developed from the LOCSME RAB method. In each of these algorithms, we have derived recursions for the weight vector update and exploited effective shrinkage methods, both of which require low complexity without losing any noticeable performance. Additionally, in the CG-based RAB algorithms we have enabled the estimation for the mismatch steering vector inside the CG recursions to enhance the robustness. Both complexity and performance comparisons are provided and analyzed. Simulation results have shown that the proposed algorithms achieved excellent output SINR performance and are suitable for operation in high input SNR.

REFERENCES

[1] H. L. Van Trees, *Optimum Array Processing*, New York: Wiley, 2002.

[2] B. Van Veen and K. M. Buckley, “Beamforming Techniques for Spatial Filtering,” 2000 CRC Press LLC.

[3] S. Li, R. C. de Lamare and M. Haardt, “Adaptive Frequency-Domain Group-Based Shrinkage Estimators for UWB Systems”, *IEEE Trans. Veh. Technol.*, Vol. 62, No. 8, pp. 3639-3652, Oct 2013.

[4] S. A. Vorobyov, A. B. Gershman and Z. Luo, “Robust Adaptive Beamforming using Worst-Case Performance Optimization: A Solution to Signal Mismatch Problem,” *IEEE Trans. Sig. Proc.*, Vol. 51, No. 4, pp 313-324, Feb 2003.

[5] J. Li, P. Stoica and Z. Wang, “On Robust Capon Beamforming and Diagonal Loading,” *IEEE Trans. Sig. Proc.*, Vol. 57, No. 7, pp 1702-1715, July 2003.

[6] D. Astely and B. Ottersten, “The effects of Local Scattering on Direction of Arrival Estimation with Music,” *IEEE Trans. Sig. Proc.*, Vol. 47, No. 12, pp 3220-3234, Dec 1999.

[7] A. Khabbazibasmenj, S. A. Vorobyov and A. Hassanien, “Robust Adaptive Beamforming Based on Steering Vector Estimation with as Little as Possible Prior Information,” *IEEE Trans. Sig. Proc.*, Vol. 60, No. 6, pp 2974-2987, June 2012.

[8] A. Hassanien, S. A. Vorobyov and K. M. Wong, “Robust Adaptive Beamforming Using Sequential Quadratic Programming: An Iterative Solution to the Mismatch Problem,” *IEEE Sig. Proc. Letters.*, Vol. 15, pp 733-736, 2008.

[9] L. Landau, R. de Lamare, M. Haardt, “Robust Adaptive Beamforming Algorithms Using Low-Complexity Mismatch Estimation,” *Proc. IEEE Statistical Signal Processing Workshop*, 2011.

[10] Y. Gu and A. Leshem, “Robust Adaptive Beamforming Based on Jointly Estimating Covariance Matrix and Steering Vector,” *Proc. IEEE International Conference on Acoustics Speech and Signal Processing*, pp 2640-2643, 2011.

[11] Y. Gu and A. Leshem, "Robust Adaptive Beamforming Based on Interference Covariance Matrix Reconstruction and Steering Vector Estimation," *IEEE Trans. Sig. Proc.*, Vol. 60, No. 7, July 2012.

[12] Y. Chen, A. Wiesel and A. O. Hero III, "Shrinkage Estimation of High Dimensional Covariance Matrices," *Proc. IEEE International Conference on Acoustics Speech and Signal Processing*, pp 2937-2940, 2009.

[13] O. Ledoit and M. Wolf, "A well-conditioned estimator for large dimensional covariance matrix," Journal of Multivariate Analysis, vol. 88 (2004) 365-411, Feb 2001.

[14] H. Ruan and R. C. de Lamare, "Robust Adaptive Beamforming Using a Low-Complexity Shrinkage-Based Mismatch Estimation Algorithm," *IEEE Sig. Proc. Letters.*, Vol. 21, No. 1, pp 60-64, 2013.

[15] A. Elnashar, "Efficient implementation of robust adaptive beamforming based on worst-case performance optimization," IET Signal Process., Vol. 2, No. 4, pp. 381-393, Dec 2008.

[16] J. Zhuang and A. Manikas, "Interference cancellation beamforming robust to pointing errors," IET Signal Process., Vol. 7, No. 2, pp. 120-127, April 2013.

[17] L. Wang and R. C. de Lamare, "Constrained adaptive filtering algorithms based on conjugate gradient techniques for beamforming," IET Signal Process., Vol. 4, No. 6, pp. 686697, Feb 2010.

[18] L. Wang, "Array Signal Processing Algorithms for Beamforming and Direction Finding," (University of York, Ph.D. Thesis, Department of Electronics, University of York, 2009).

[19] R. Fa, R. de Lamare and V. H. Nascimento, "Knowledge-aided STAP algorithm using convex combination of inverse covariance matrices for heterogeneous clutter," *Proc. IEEE International Conference on Acoustics Speech and Signal Processing*, pp 2742-2745, 2010.

[20] P. Stoica, J. Li, X. Zhu, and J. R. Guerci, "On Using a priori Knowledge in Space-Time Adaptive Processing, IEEE Transactions on Signal Processing, Vol. 56, No. 6, pp. 2598-2602, June 2008.

[21] S. E. Nai, W. Ser, Z. L. Yu and H. Chen, "Iterative Robust Minimum Variance Beamforming," *IEEE Transactions on Signal Processing*, Vol. 59, NO. 4, April 2011.

[22] Z. L. Yu, Z. Gu, J. Zhou, Y. Li, S. Wee, M. H. Er, "A Robust Adaptive Beamformer Based on Worst-Case Semi-Definite Programming," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5914-5919, 2010.

[23] J. P. Lie, W. Ser and C. M. S. See, "Adaptive Uncertainty Based Iterative Robust Capon Beamformer Using Steering Vector Mismatch Estimation," *IEEE Transactions on Signal Processing*, Vol. 59, No. 9, Sep 2011.

[24] M. Morelli, L. Sangunietti and U. Mengali, "Channel Estimation for Adaptive Frequency-Domain Equalization," *IEEE Transactions on Wireless Communications*, Vol. 4, No. 5, pp. 53-57, Sep 2005.

[25] B. Liao, S. C. Chan, and K. M. Tsui, "Recursive Steering Vector Estimation and Adaptive Beamforming Under Uncertainties," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 489-501, Jan 2013.