

# Robust Adaptive Beamforming Based on Low-Rank and Cross-Correlation Techniques

Hang Ruan and Rodrigo C. de Lamare

*Abstract*—This work presents cost-effective low-rank techniques for designing robust adaptive beamforming (RAB) algorithms. The proposed algorithms are based on the exploitation of the cross-correlation between the array observation data and the output of the beamformer. Firstly, we construct a general linear equation considered in large dimensions whose solution yields the steering vector mismatch. Then, we employ the idea of the full orthogonalization method (FOM), an orthogonal Krylov subspace based method, to iteratively estimate the steering vector mismatch in a reduced-dimensional subspace, resulting in the proposed orthogonal Krylov subspace projection mismatch estimation (OKSPME) method. We also devise adaptive algorithms based on stochastic gradient (SG) and conjugate gradient (CG) techniques to update the beamforming weights with low complexity and avoid any costly matrix inversion. The main advantages of the proposed low-rank and mismatch estimation techniques are their cost-effectiveness when dealing with high dimension subspaces or large sensor arrays. Simulations results show excellent performance in terms of the output signal-to-interference-plus-noise ratio (SINR) of the beamformer among all the compared RAB methods.

*Keywords*—robust adaptive beamforming, low-rank techniques, low complexity algorithms.

## I. INTRODUCTION

Adaptive beamforming has been one of the most important research areas in sensor array signal processing. It has also been recognized that traditional adaptive beamformers are extremely sensitive to environmental uncertainties or steering vector mismatches, which may be caused by many different factors (e.g., imprecise antenna size calibration, signal pointing errors or local scattering). Furthermore, some radar systems in advanced applications require antenna arrays with a very large number of sensor elements in highly dynamic environments, which leads to the increase of computational complexity and the decrease of the convergence rate for computing the parameters of the beamformer.

### A. Prior and Related Work

In order to mitigate the effects of uncertainties on adaptive beamformers, robust adaptive beamforming (RAB) techniques have been developed. Popular approaches include worst-case optimization [2], [9], diagonal loading [4], and eigen-subspace decomposition and projection techniques [6], [8], [11]. However, these RAB approaches have some limitations such as their ad hoc nature, high probability of subspace swap at low signal-to-noise ratio (SNR) [3] and high computational cost due to on-line optimization or subspace decomposition techniques.

Furthermore, in the case of large sensor arrays the above mentioned RAB methods may encounter problems for their application. This is because in these RAB algorithms, a cubic or greater computational cost is required to compute the beamforming parameters. Therefore, dimensionality reduction (or rank-reduction) methods ([15]-[31], [33]-[36]) have been employed and developed to reduce the complexity and improve the convergence rate.

In the recent years, great efforts have been devoted to the investigation of robust dimensionality reduction techniques for RAB. The beamspace approach of [23] projects the data onto a lower dimension subspace by using a beamspace matrix, whose columns are determined by linearly independent constrained optimization problems. A more effective approach (i.e., [16]-[20],[24],[25]) is based on preprocessing the array observation data using a Krylov subspace. However, there are different ways to generate the Krylov subspace and the choice usually depends on the cost and the performance. The Arnoldi method [12], [13], [21] and the Lanczos iterations [12], [13], [15] are typical approaches used to generate orthogonal Krylov subspaces, whereas [19] also introduces a method to generate non-orthogonal ones. However, the main challenge in these techniques is the model order determination. Specifically, the model order must be properly chosen to ensure robustness to over-determination of the system model order [17]. Another effective approach to dimensionality reduction is the joint iterative optimization (JIO) [26]-[31] techniques, [33]-[36], which employ a subspace projection matrix and jointly and iteratively optimize the bases of the subspace and the beamformer weights. The work in [27] has developed a recursive least squares (RLS) adaptive algorithm based on widely-linear processing using the JIO technique. The study in [29] has devised efficient stochastic gradient (SG) and RLS RAB algorithms from a modified JIO (MJIO) scheme.

### B. Contributions

In this work, we propose and study novel RAB algorithms that are based on low-rank and cross-correlation techniques. In the proposed techniques, we exploit the prior knowledge that the steering vector mismatch of the desired signal is located within an angular sector which is assumed known. The proposed algorithms are based on the exploitation of the cross-correlation between the array observation data and the output of the beamformer, which avoids costly optimization procedures. We firstly construct a linear system (considered in high dimension) involving the mismatched steering vector and the statistics of the sampled data. Then we employ an iterative full orthogonalization method (FOM) [12], [13] to compute an orthogonal Krylov subspace whose model order is determined by both the minimum sufficient rank [17], which ensures no information loss when capturing the signal of interest (SoI) with interferers, and the execute-and-stop criterion of FOM [12], [13], which automatically avoids overestimating the number of bases of the computed subspace. The estimated vector that contains the cross-correlation between the array observation data and the beamformer output is projected onto the Krylov subspace, in order to update the steering vector mismatch, resulting in the pro-

posed orthogonal Krylov subspace projection mismatch estimation (OKSPME) method.

Furthermore, based on the OKSPME method, we have also devised adaptive stochastic gradient (SG), conventional conjugate gradient (CCG) and modified conjugate gradient (MCG) algorithms derived from the proposed optimization problems to reduce the cost for computing the beamforming weights, resulting in the proposed OKSPME-SG, OKSPME-CCG and OKSPME-MCG RAB algorithms. We remark that the steering vector is also estimated and updated using the CG-based recursions to produce an even more precise estimate. Derivations of the proposed algorithms are presented and discussed along with an analysis of their computational complexity.

Moreover, we develop an analysis of the mean squared error (MSE) between the estimated and the actual steering vectors for the general approach of using a presumed angular sector associated with subspace projections. This analysis mathematically describes how precise the steering vector mismatch can be estimated. Upper and lower bounds are derived and compared with the approach in [6]. Another analysis on the computational complexity of the proposed and existing algorithms is also provided.

In the simulations, we consider local scattering scenarios (both coherent and incoherent) to model the mismatch effects. We also study the performance of the proposed algorithms by testing the output signal-to-interference-plus-noise ratio (SINR) of the beamformer with respect to training snapshots and different input SNRs. The number of sensor elements and interferers is also varied and compared in each scenario to provide a comprehensive performance study. In summary, the contributions of this work are:

- The proposed OKSPME RAB method.
- The development of the modified SG and CG type OKSPME RAB algorithms.
- An analysis of the computational complexity and the MSE performance of the proposed and existing RAB algorithms.

The remaining sections of this paper are organized as follows: The system model and problem statement are described in Section II. Section III introduces the proposed OKSPME method, whereas Section IV introduces the proposed robust adaptive algorithms. Section V provides the MSE analysis of the steering vector estimation and the complexity analysis. Section VI presents and discusses the simulation results. Section VII gives the conclusion.

## II. SYSTEM MODEL AND PROBLEM STATEMENT

Let us consider a linear antenna array of  $M$  sensors and  $K$  narrowband signals which impinge on the array. The data received at the  $i$ th snapshot can be modeled as

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

where  $\mathbf{s}(i) \in \mathbb{C}^{K \times 1}$  are uncorrelated source signals,  $\boldsymbol{\theta} = [\theta_1, \dots, \theta_K]^T \in \mathbb{R}^K$  is a vector containing the directions of arrival (DoAs) and  $[\cdot]^T$  denotes the transpose,  $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1) + \mathbf{e}, \dots, \mathbf{a}(\theta_K)] = [\mathbf{a}_1, \dots, \mathbf{a}_K] \in \mathbb{C}^{M \times K}$  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 circular Gaussian noise with zero mean and variance  $\sigma_n^2$ . The beamformer output is given by

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

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

$$SINR = \frac{\sigma_1^2 |\mathbf{w}^H \mathbf{a}_1|^2}{\mathbf{w}^H \mathbf{R}_{I+N} \mathbf{w}}, \quad (3)$$

where  $\sigma_1^2$  is the desired signal power,  $\mathbf{R}_{I+N}$  is the interference-plus-noise covariance (INC) matrix. The problem of maximizing the SINR in (3) can be cast as the following optimization problem:

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && \mathbf{w}^H \mathbf{R}_{I+N} \mathbf{w} \\ & \text{subject to} && \mathbf{w}^H \mathbf{a}_1 = 1, \end{aligned} \quad (4)$$

which is known as the MVDR beamformer or Capon beamformer [1], [4]. 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 sample covariance matrix (SCM) of the received data as

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

Using the SCM for directly computing the weights will lead to 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 [2], [3]. As previously mentioned, most of the conventional and existing RAB algorithms are computationally costly especially when encountering arrays with a very large number of sensors. Therefore, the RAB design problem we are interested in solving includes the following aspects:

- To design cost-efficient algorithms that are robust against uncertainties and values of SNRs and interferers in the presence of uncertainties in the steering vector of a desired signal.
- The proposed algorithms must preserve their robustness and low-complexity features for large sensor arrays.

## III. PROPOSED OKSPME METHOD

In this section, the proposed OKSPME method is introduced. This method aims to construct a linear system involving only known or estimated statistics and then projects an estimated cross-correlation vector between the array observation data and the beamformer output onto an orthogonal Krylov subspace, in order to update the steering vector mismatch with reduced complexity. The SCM of the array observation data is estimated by (5). The cross-correlation vector between the array observation data and the beamformer output can be expressed as  $\mathbf{d} = E[\mathbf{x}y^*]$  (where  $[\cdot]^*$  denotes complex conjugation) or equivalently as

$$\mathbf{d} = E[(\mathbf{A}\mathbf{s} + \mathbf{n})(\mathbf{A}\mathbf{s} + \mathbf{n})^H \mathbf{w}]. \quad (6)$$

Assuming that  $|\mathbf{a}_k^H \mathbf{w}| \ll |\mathbf{a}_1^H \mathbf{w}|$  for  $k = 2, \dots, K$  and all signals have zero mean, the cross-correlation vector  $\mathbf{d}$  can be rewritten as

$$\mathbf{d} = E[(\mathbf{A}\mathbf{s} + \mathbf{n})(s_1^* \mathbf{a}_1^H \mathbf{w} + \mathbf{n}^H \mathbf{w})]. \quad (7)$$

~~By also assuming that the desired signal is statistically independent from the interferers and the noise so that the interferers are sufficiently canceled (i.e.  $E[s_k s_1^*] = 0$  and  $E[s_k \mathbf{a}_k s_1^* \mathbf{a}_1^H \mathbf{w}] = 0$  for  $k = 2, \dots, K$ ) such that they fall much below the noise floor and the desired signal power is not affected by the interference, (7) can be rewritten as~~

$$\mathbf{d} = E[\sigma_1^2 \mathbf{a}_1^H \mathbf{w} \mathbf{a}_1 + \mathbf{n} \mathbf{n}^H \mathbf{w}], \quad (8)$$

which can be estimated by the sample cross-correlation vector (SCV) given by

$$\hat{\mathbf{d}}(i) = \frac{1}{i} \sum_{k=1}^i \mathbf{x}(k) y^*(k). \quad (9)$$

#### A. Desired Signal Power Estimation

In this subsection, we describe an iterative method for the desired signal power ( $\sigma_1^2$ ) estimation based on our prior work in [32], which can be accomplished by directly using the desired signal steering vector. **Alternatively, a designer can employ a maximum likelihood (ML) or a minimum variance (MV) estimator for computing the desired signal power. However, the approach described is recommended as it has lower complexity than the ML and the MV estimators.** In the adopted method, we need to choose an initial guess for the steering vector mismatch within the presumed angular sector, say  $\hat{\mathbf{a}}_1(0)$  and set  $\hat{\mathbf{a}}_1(1) = \hat{\mathbf{a}}_1(0)$ . By adding the snapshot index  $i$ , we can rewrite the array observation data as

$$\mathbf{x}(i) = \hat{\mathbf{a}}_1(i) s_1(i) + \sum_{k=2}^K \mathbf{a}_k(i) s_k(i) + \mathbf{n}(i), \quad (10)$$

where  $\hat{\mathbf{a}}_1(0)$  and  $\hat{\mathbf{a}}_1(i)$  ( $i = 1, 2, \dots$ ) designate the initial guess of the steering vector and its estimate at the  $i$ th snapshot, respectively.

Pre-multiplying the above equation by  $\hat{\mathbf{a}}_1^H(i)$  we have

$$\hat{\mathbf{a}}_1^H(i) \mathbf{x}(i) = \hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i) s_1(i) + \sum_{k=2}^K \hat{\mathbf{a}}_1^H(i) \mathbf{a}_k(i) s_k(i) + \mathbf{n}(i). \quad (11)$$

Here we assume that each of the interferers is orthogonal or approximately orthogonal to the desired signal. Specifically, the steering vector of each of the interferers is orthogonal ( $\hat{\mathbf{a}}_1^H(i) \mathbf{a}_k(i) = 0$ ,  $k = 2, 3, \dots, K$ ), or approximately orthogonal ( $\hat{\mathbf{a}}_1^H(i) \mathbf{a}_k(i) \ll \hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i)$ ,  $k = 2, 3, \dots, K$ ) to the desired signal steering vector (i.e.,  $\hat{\mathbf{a}}_1(i)$ ), so that  $\hat{\mathbf{a}}_1^H(i) \mathbf{a}_k(i)$  ( $k = 2, 3, \dots, K$ ) approaches zero and the term  $\sum_{k=2}^K \hat{\mathbf{a}}_1^H(i) \mathbf{a}_k(i) s_k(i)$  in (11) can be neglected, resulting in

$$\hat{\mathbf{a}}_1^H(i) \mathbf{x}(i) = \hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i) s_1(i) + \hat{\mathbf{a}}_1^H(i) \mathbf{n}(i). \quad (12)$$

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

$$E[|\hat{\mathbf{a}}_1^H(i) \mathbf{x}(i)|^2] = E[(\hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i) s_1(i) + \hat{\mathbf{a}}_1^H(i) \mathbf{n}(i))^* (\hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i) s_1(i) + \hat{\mathbf{a}}_1^H(i) \mathbf{n}(i))]. \quad (13)$$

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

$$E[|\hat{\mathbf{a}}_1^H(i) \mathbf{x}(i)|^2] = |\hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i)|^2 E[|s_1(i)|^2] + \hat{\mathbf{a}}_1^H(i) E[\mathbf{n}(i) \mathbf{n}^H(i)] \hat{\mathbf{a}}_1(i), \quad (14)$$

where  $E[\mathbf{n}(i) \mathbf{n}^H(i)]$  represents the noise covariance matrix  $\mathbf{R}_n(i)$  that can be replaced by  $\sigma_n^2 \mathbf{I}_M$ , where the noise variance  $\sigma_n^2$  can be easily estimated by a specific estimation method. A possible approach is to use a Maximum Likelihood (ML) based method as in [14]. Replacing the desired signal power  $E[|s_1(i)|^2]$  and the noise variance  $\sigma_n^2$  by their estimates  $\hat{\sigma}_1^2(i)$  and  $\hat{\sigma}_n^2(i)$ , respectively, we obtain

$$\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i) \mathbf{x}(i)|^2 - |\hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i)| \hat{\sigma}_n^2(i)}{|\hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i)|^2}. \quad (15)$$

The expression in (15) has a low complexity ( $\mathcal{O}(M)$ ) and can be directly implemented if the desired signal steering vector and the noise level are accurately estimated.

#### B. Orthogonal Krylov Subspace Approach for Steering Vector Mismatch Estimation

An orthogonal Krylov subspace strategy is proposed in order to estimate the mismatch with reduced cost and deal with situations in which the model order is time-varying. Our idea is based on constructing a linear system, which considers the steering vector mismatch as the solution, and solving it by using an iterative Krylov subspace projection method. To this end, consider a general high-dimensional linear system model given by

$$\mathbf{B} \mathbf{a}_1 = \mathbf{b}, \quad (16)$$

where  $\mathbf{B} \in \mathbb{C}^{M \times M}$  and  $\mathbf{b} \in \mathbb{C}^{M \times 1}$ . Then we need to express  $\mathbf{B}$  and  $\mathbf{b}$  only using available information (known statistics or estimated parameters), so that we can solve the linear system with the Krylov subspace of order  $m$  ( $m \ll M$ ) described by

$$\mathbf{K}_m = \text{span}\{\mathbf{b}, \mathbf{B}\mathbf{b}, \mathbf{B}^2\mathbf{b}, \dots, \mathbf{B}^{m-1}\mathbf{b}\}. \quad (17)$$

Taking the complex conjugate of (12), we have

$$\mathbf{x}^H(i) \hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i) s_1^*(i) + \mathbf{n}^H(i) \hat{\mathbf{a}}_1(i). \quad (18)$$

Pre-multiplying both sides of (18) by the terms of (10), **then adding an extra term  $\delta \mathbf{I} \hat{\mathbf{a}}_1(i)$  (where  $\delta$  is a small positive number defined by the user) and simplifying the terms, we obtain**

$$(\mathbf{x}(i) \mathbf{x}^H(i) + \delta \mathbf{I}) \hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1(i) \hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i) s_1(i) s_1^*(i) + \mathbf{n}(i) \mathbf{n}^H(i) \hat{\mathbf{a}}_1(i) + \delta \hat{\mathbf{a}}_1(i). \quad (19)$$

Replacing  $\mathbf{x}(i) \mathbf{x}^H(i) + \delta \mathbf{I}$  by  $\hat{\mathbf{R}}(i)$ ,  $s_1(i) s_1^*(i)$  by  $\hat{\sigma}_1^2(i)$  and  $\mathbf{n}(i) \mathbf{n}^H(i)$  by  $\hat{\sigma}_n^2(i) \mathbf{I}_M$ , we obtain

$$\hat{\mathbf{R}}(i) \hat{\mathbf{a}}_1(i) = \underbrace{\hat{\mathbf{a}}_1(i) \hat{\mathbf{a}}_1^H(i) \hat{\mathbf{a}}_1(i) \hat{\sigma}_1^2(i)}_{\hat{\mathbf{b}}(i)} + (\hat{\sigma}_n^2(i) + \delta) \hat{\mathbf{a}}_1(i), \quad (20)$$

TABLE I  
ARNOLDI-MODIFIED GRAM-SCHMIDT ALGORITHM

---

For  $j = 1, 2, \dots$  do:  
  Compute  $\mathbf{u}_j = \hat{\mathbf{R}}\mathbf{t}_j$   
  For  $l = 1, 2, \dots, j$ , do:  
     $h_{l,j} = \langle \mathbf{u}_j, \mathbf{t}_l \rangle$   
     $\mathbf{u}_j = \mathbf{u}_j - h_{l,j}\mathbf{t}_l$   
  End do.  
  Compute  $h_{j,j+1} = \|\mathbf{u}_j\|$ .  
  If  $h_{j,j+1} = 0$  or  $j \geq K + 1$ ,  
    set  $m = j$ ;  
    break;  
  Else compute  $\mathbf{t}_{j+1} = \frac{\mathbf{u}_j}{h_{j,j+1}}$ .  
End do.

---

in which by further defining the expression on the right-hand side as  $\hat{\mathbf{b}}(i)$ , we can rewrite (20) as

$$\hat{\mathbf{R}}(i)\hat{\mathbf{a}}_1(i) = \hat{\mathbf{b}}(i). \quad (21)$$

As can be seen (21) shares the same form as the linear system of equations in (16) and  $\hat{\mathbf{b}}(i)$  can be expressed in terms of  $\hat{\mathbf{a}}_1(i)$ ,  $\hat{\sigma}_1^2(i)$  and  $\hat{\sigma}_n^2(i)$  whereas  $\hat{\mathbf{R}}(i)$  can be estimated by (5). In the following step, we employ the Arnoldi-modified Gram-Schmidt algorithm from the FOM method [12], [13] associated with the minimum sufficient rank criterion discussed in [17] to compute an orthogonal Krylov subspace. We define a residue vector to represent the estimation error in the  $i$ th snapshot as

$$\hat{\mathbf{r}}(i) = \hat{\mathbf{b}}(i) - \hat{\mathbf{R}}(i)\hat{\mathbf{a}}_1(i), \quad (22)$$

and let

$$\mathbf{t}_1(i) = \frac{\hat{\mathbf{r}}(i)}{\|\hat{\mathbf{r}}(i)\|}. \quad (23)$$

Then the Krylov subspace bases can be computed using the modified Arnoldi-modified Gram-Schmidt algorithm as in Table I (the snapshot index  $i$  is omitted here for simplicity).

In Table I,  $\langle, \rangle$  denotes the inner product and the parameters  $h_{l,j}$  ( $l, j = 1, 2, \dots, m$ ) are real-valued coefficients, the model order is determined once if one of the following situations is satisfied:

- The execute-and-stop criterion of the original Arnoldi-modified Gram-Schmidt algorithm is satisfied (i.e.,  $h_{j,j+1} = 0$ ).
- The minimum sufficient rank for addressing the SoI and interferers is achieved (i.e.,  $j \geq K + 1$ , where  $K$  is the number of signal sources), so that no more subspace components are necessary for capturing the SoI from all the existing signal sources.

Now by inserting the snapshot index, we have

$$\hat{\mathbf{T}}(i) = [\mathbf{t}_1(i), \mathbf{t}_2(i), \dots, \mathbf{t}_m(i)], \quad (24)$$

and the Krylov subspace projection matrix is computed by

$$\hat{\mathbf{P}}(i) = \hat{\mathbf{T}}(i)\hat{\mathbf{T}}^H(i). \quad (25)$$

It should be emphasized that the Krylov subspace matrix  $\hat{\mathbf{T}}(i)$  obtained here is constructed by starting with the residue vector

TABLE II  
PROPOSED OKSPME METHOD

---

Initialization:  
 $\hat{\mathbf{w}}(1) = \mathbf{1}$ ;  
Choose an initial guess  $\hat{\mathbf{a}}_1(0)$  within the sector and set  $\hat{\mathbf{a}}_1(1) = \hat{\mathbf{a}}_1(0)$ ;  
For each snapshot  $i = 1, 2, \dots$ :  
   $\hat{\mathbf{R}}(i) = \frac{1}{i} \sum_{k=1}^i \mathbf{x}(k)\mathbf{x}^H(k)$   
   $\hat{\mathbf{d}}(i) = \frac{1}{i} \sum_{k=1}^i \mathbf{x}(k)y^*(k)$   
  **Step 1. Compute the desired signal power**  
 $\hat{\sigma}_1^2(i) = \frac{|\hat{\mathbf{a}}_1^H(i)\mathbf{x}(i)|^2 - |\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|\hat{\sigma}_n^2(i)}{|\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)|^2}$   
  **Step 2. Determine the Krylov subspace**  
 $\hat{\mathbf{b}}(i) = \hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)\hat{\sigma}_1^2(i) + \hat{\sigma}_n^2(i)\hat{\mathbf{a}}_1(i)$   
 $\hat{\mathbf{r}}(i) = \hat{\mathbf{b}}(i) - \hat{\mathbf{R}}(i)\hat{\mathbf{a}}_1(i)$   
 $\mathbf{t}_1(i) = \frac{\hat{\mathbf{r}}(i)}{\|\hat{\mathbf{r}}(i)\|}$   
Apply the algorithm in Table I to determine  $m$  and  $\mathbf{t}_1(i), \dots, \mathbf{t}_m(i)$   
 $\hat{\mathbf{T}}(i) = [\mathbf{t}_1(i), \mathbf{t}_2(i), \dots, \mathbf{t}_m(i)]$   
  **Step 3. Update the steering vector**  
 $\hat{\mathbf{P}}(i) = \hat{\mathbf{T}}(i)\hat{\mathbf{T}}^H(i)$   
 $\hat{\mathbf{a}}_1(i+1) = \hat{\mathbf{a}}_1(i) + \frac{\hat{\mathbf{P}}(i)\hat{\mathbf{d}}(i)}{\|\hat{\mathbf{P}}(i)\hat{\mathbf{d}}(i)\|}$   
 $\hat{\mathbf{a}}_1(i+1) = \hat{\mathbf{a}}_1(i+1)/\|\hat{\mathbf{a}}_1(i+1)\|$   
  **Step 4. Compute the weight vector**  
 $\hat{\mathbf{R}}_{I+N}(i) = \hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)$   
 $\hat{\mathbf{w}}(i) = \frac{\hat{\mathbf{R}}_{I+N}^{-1}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{R}}_{I+N}^{-1}(i)\hat{\mathbf{a}}_1(i)}$   
End snapshot

---

$\hat{\mathbf{r}}(i)$ . In other words,  $\hat{\mathbf{T}}(i)$  is constructed with the estimation error of the steering vector. In order to extract the estimation error information and use it to update the steering vector mismatch, we can project the SCV  $\hat{\mathbf{d}}(i)$  in (9) onto  $\hat{\mathbf{P}}(i)$  and add the estimation error to the current estimate of  $\hat{\mathbf{a}}_1(i)$  as

$$\hat{\mathbf{a}}_1(i+1) = \hat{\mathbf{a}}_1(i) + \frac{\hat{\mathbf{P}}(i)\hat{\mathbf{d}}(i)}{\|\hat{\mathbf{P}}(i)\hat{\mathbf{d}}(i)\|}. \quad (26)$$

### C. INC Matrix and Beamformer Weight Vector Computation

Since we have estimated both the desired signal power  $\hat{\sigma}_1^2(i)$  and the mismatched steering vector in the previous subsections, the INC matrix can be obtained by subtracting the desired signal covariance matrix out from the SCM as

$$\hat{\mathbf{R}}_{I+N}(i) = \hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i). \quad (27)$$

The beamformer weight vector is computed by

$$\hat{\mathbf{w}}(i) = \frac{\hat{\mathbf{R}}_{I+N}^{-1}(i)\hat{\mathbf{a}}_1(i)}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{R}}_{I+N}^{-1}(i)\hat{\mathbf{a}}_1(i)}, \quad (28)$$

which has a computationally costly matrix inversion  $\hat{\mathbf{R}}_{I+N}^{-1}(i)$ . The proposed OKSPME method is summarized in Table II. In the next section, we will introduce adaptive algorithms to avoid matrix inversions and reduce the complexity.

## IV. PROPOSED ADAPTIVE ALGORITHMS

This section presents adaptive strategies based on the OKSPME robust beamforming method, resulting in the proposed OKSPME-SG, OKSPME-CCG and OKSPME-MCG algorithms, which are especially suitable for dynamic scenarios.

In the proposed adaptive algorithms, we estimate the desired signal power and its steering vector with the same recursions as in OKSPME, whereas the estimation procedure of the beamforming weights is different. In particular, we start from a reformulated optimization problem and use SG and CG-based adaptive recursions to derive the weight update equations, which reduce the complexity by an order of magnitude as compared to that of OKSPME.

#### A. OKSPME-SG Adaptive Algorithm

We resort to an SG adaptive strategy and consider the following optimization problem:

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

where  $\hat{\mathbf{R}}(i)$  can be written as  $\mathbf{x}(i)\mathbf{x}^H(i)$  and  $\hat{\mathbf{R}}_1(i)$  represents the desired signal covariance matrix and can be written as  $\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)$ .

Then we can express the SG recursion as

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

where  $\mathcal{L} = \mathbf{w}^H(i)(\mathbf{x}(i)\mathbf{x}^H(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{w}(i) + \lambda_{\mathcal{L}}(\mathbf{w}^H(i)\hat{\mathbf{a}}_1(i) - 1)$  and  $\mu$  is the step size.

By substituting  $\mathcal{L}$  into the SG equation (30) and letting  $\mathbf{w}^H(i+1)\hat{\mathbf{a}}_1(i+1) = 1$ ,  $\lambda_{\mathcal{L}}$  is obtained as

$$\lambda_{\mathcal{L}} = \frac{2(\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i) - y(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i))}{\hat{\mathbf{a}}_1^H(i)\hat{\mathbf{a}}_1(i)}. \quad (31)$$

By substituting  $\lambda_{\mathcal{L}}$  in (30) again, the weight update equation for OKSPME-SG is obtained as

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

The adaptive SG recursion circumvents a matrix inversion when computing the weights using (28), which is unavoidable in OKSPME. Therefore, the computational complexity is reduced from  $\mathcal{O}(M^3)$  in OKSPME to  $\mathcal{O}(M^2)$  in OKSPME-SG. It is also important that the step size  $\mu$  should satisfy  $0 < \mu < \frac{1}{\hat{\sigma}_1^2(i)}$  to guarantee that  $\mathbf{I} - \mu\hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i)$  is always a positive-definite matrix so that (32) is ensured converging to a solution. To implement OKSPME-SG, **Step 1**, **Step 2** and **Step 3** from Table II and (32) are required.

#### B. OKSPME-CCG Adaptive Algorithm

In this subsection, the proposed OKSPME-CCG algorithm is introduced. In CG-based approaches, we usually employ a forgetting factor (e.g.  $\lambda$ ) to estimate the second-order statistics of the data or the SCM [1], [10], which can be expressed by

$$\hat{\mathbf{R}}(i) = \lambda\hat{\mathbf{R}}(i-1) + \mathbf{x}(i)\mathbf{x}^H(i), \quad (33)$$

whereas the SCV  $\hat{\mathbf{d}}(i)$  can be estimated with the same forgetting factor as described by

$$\hat{\mathbf{d}}(i) = \lambda\hat{\mathbf{d}}(i-1) + \mathbf{x}(i)y^*(i). \quad (34)$$

The proposed optimization problem that leads to the OKSPME-CCG algorithm is described by

$$\underset{\hat{\mathbf{a}}_1(i), \mathbf{v}(i)}{\text{minimize}} \mathcal{J} = \mathbf{v}^H(i)(\hat{\mathbf{R}}(i) - \hat{\mathbf{R}}_1(i))\mathbf{v}(i) - \hat{\mathbf{a}}_1^H(i)\mathbf{v}(i), \quad (35)$$

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

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

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

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

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

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

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

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

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

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 the cost function with respect to  $\alpha_{\hat{\mathbf{a}}_1,n}(i)$  and  $\alpha_{\mathbf{v},n}(i)$ , respectively. The solutions are given by

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

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

The parameters  $\beta_{\hat{\mathbf{a}}_1,n}(i)$  and  $\beta_{\mathbf{v},n}(i)$  should be chosen to provide conjugacy for direction vectors [10], which results in

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

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

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

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

The computational cost of OKSPME-CCG algorithm is  $\mathcal{O}(NM^2)$ , which is higher than the cost required in OKSPME-SG due to the inner iterations at every snapshot. The proposed OKSPME-CCG is summarized in Table III.

TABLE III  
PROPOSED OKSPME-CCG ALGORITHM

Initialization:

$$\hat{\mathbf{w}}(1) = \mathbf{v}_0(1) = \mathbf{1}; \lambda;$$

Choose an initial guess  $\hat{\mathbf{a}}_1(0)$  within the sector and set  $\hat{\mathbf{a}}_1(1) = \hat{\mathbf{a}}_1(0)$ ;

For each snapshot  $i = 1, 2, \dots$ :

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

$$\hat{\mathbf{d}}(i) = \frac{1}{i} \sum_{k=1}^i \mathbf{x}(k) y^*(k)$$

Step 1 from Table II

Step 2 from Table II

Step 3 from Table II

Steering Vector and Weight Vector Estimations

$$\hat{\mathbf{a}}_{1,0}(i) = \hat{\mathbf{a}}_1(i)$$

$$\mathbf{g}_{\hat{\mathbf{a}}_1,0}(i) = \hat{\sigma}_1^2(i) \mathbf{v}_0(i) \mathbf{v}_0^H(i) \hat{\mathbf{a}}_{1,0}(i) + \mathbf{v}_0(i)$$

$$\mathbf{g}_{\mathbf{v},0}(i) = \hat{\mathbf{a}}_{1,0}(i) - \hat{\mathbf{R}}(i) \mathbf{v}_0(i)$$

$$\mathbf{P}_{\hat{\mathbf{a}}_1,0}(i) = \mathbf{g}_{\hat{\mathbf{a}}_1,0}(i); \mathbf{P}_{\mathbf{v},0}(i) = \mathbf{g}_{\mathbf{v},0}(i)$$

For each iteration index  $n = 1, 2, \dots, N$ :

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

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

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

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

$$\mathbf{g}_{\hat{\mathbf{a}}_1,n}(i) = \hat{\sigma}_1^2(i) \mathbf{v}_n(i) \mathbf{v}_n^H(i) \hat{\mathbf{a}}_{1,n}(i) + \mathbf{v}_n(i)$$

$$\mathbf{g}_{\mathbf{v},n}(i) = \mathbf{g}_{\mathbf{v},n-1}(i) - \alpha_{\mathbf{v},n}(i) (\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i) \mathbf{x}(i) \mathbf{x}^H(i)) \mathbf{P}_{\mathbf{v},n}(i)$$

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

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

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

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

End iteration

$$\mathbf{v}_0(i+1) = \mathbf{v}_N(i)$$

$$\mathbf{w}(i) = \frac{\mathbf{v}_N(i)}{\hat{\mathbf{a}}_{1,N}^H(i) \mathbf{v}_N(i)}$$

End snapshot

### C. OKSPME-MCG Adaptive Algorithm

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

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

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

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 [10] given by

$$0 \leq \mathbf{P}_{\hat{\mathbf{a}}_1}^H(i) \mathbf{g}_{\hat{\mathbf{a}}_1}(i) \leq 0.5 \mathbf{P}_{\hat{\mathbf{a}}_1}^H(i) \mathbf{g}_{\hat{\mathbf{a}}_1}(i-1), \quad (49)$$

$$0 \leq \mathbf{P}_{\mathbf{v}}^H(i) \mathbf{g}_{\mathbf{v}}(i) \leq 0.5 \mathbf{P}_{\mathbf{v}}^H(i) \mathbf{g}_{\mathbf{v}}(i-1). \quad (50)$$

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 re-express them in one snapshot as

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

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

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

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

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

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

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

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

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

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 (53) into (49) gives

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

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

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

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

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

$$\mathbf{p}_v(i+1) = \mathbf{g}_v(i) + \beta_v(i)\mathbf{p}_v(i), \quad (61)$$

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

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

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

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 MCG approach employs the forgetting factor  $\lambda$  and constant  $\eta$  for estimating  $\alpha(i)$ , which means its performance may depend on a suitable choice of these parameters. The proposed OKSPME-MCG algorithm requires a complexity of  $\mathcal{O}(M^2)$ . However, the cost is usually much lower compared to CCG approach for the elimination of inner recursions and it presents a similar performance in most studied scenarios. From an implementation point of view, the choice of using the CCG and MCG algorithms is based on the stationarity of the system: the CCG algorithm is more suitable for scenarios in which the system is stationary and we can compute the beamformer with  $K$  iterations while the MCG algorithm is suggested for non-stationary scenarios as we only run one iteration per snapshot and can track variations in the environment. Table IV summarizes the OKSPME-MCG algorithm.

## V. ANALYSIS

In this section, we present an analysis of the following aspects of the proposed and existing algorithms:

- An analysis of the MSE between the estimated and actual steering vectors for the general approach that employs a presumed angular sector.
- MSE analysis results of the proposed OKSPME method and the SQP method in [6] and their relationships and differences.
- A complexity analysis for the proposed and existing algorithms.

### A. MSE analysis

Firstly, we present the MSE analysis of the general approach that employs a presumed angular sector. Since we have the steering vector estimate  $\hat{\mathbf{a}}_1(i)$  in the  $i$ th snapshot, by denoting the true steering vector as  $\mathbf{a}_1$ , we can express the MSE of the estimate  $\hat{\mathbf{a}}_1(i)$  as

$$\begin{aligned} \text{MSE}\{\hat{\mathbf{a}}_1(i)\} &= \text{tr}\{E[(\hat{\mathbf{a}}_1(i) - \mathbf{a}_1)(\hat{\mathbf{a}}_1(i) - \mathbf{a}_1)^H]\} \\ &= E\{[(\hat{\mathbf{a}}_1(i) - \mathbf{a}_1)^H(\hat{\mathbf{a}}_1(i) - \mathbf{a}_1)].\} \end{aligned} \quad (65)$$

In the approach that employs an angular sector, we usually choose an initial guess (i.e.,  $\hat{\mathbf{a}}_1(0)$ ) from the presumed sector. Let us express the accumulated estimation error as

$$\hat{\mathbf{e}}(i) = \hat{\mathbf{a}}_1(i) - \hat{\mathbf{a}}_1(0), \quad (66)$$

then (65) can be rewritten as

$$\text{MSE}\{\hat{\mathbf{a}}_1(i)\} = E\{[(\hat{\mathbf{a}}_1(0) + \hat{\mathbf{e}}(i) - \mathbf{a}_1)^H(\hat{\mathbf{a}}_1(0) + \hat{\mathbf{e}}(i) - \mathbf{a}_1)].\} \quad (67)$$

TABLE IV  
PROPOSED OKSPME-MCG ALGORITHM

---

Initialization:  
 $\hat{\mathbf{w}}(1) = \mathbf{v}(0) = \mathbf{1}$ ;  $\lambda$ ;  $\eta_v = \eta_{\hat{\mathbf{a}}_1}$ ;  
 Choose an initial guess  $\hat{\mathbf{a}}_1(0)$  within the sector and set  $\hat{\mathbf{a}}_1(1) = \hat{\mathbf{a}}_1(0)$ ;  
 $\mathbf{g}_v(0) = \mathbf{p}_v(1) = \hat{\mathbf{a}}_1(1)$ ;  $\mathbf{g}_{\hat{\mathbf{a}}_1}(0) = \mathbf{p}_{\hat{\mathbf{a}}_1}(1) = \mathbf{v}(0)$ ;  
 For each snapshot  $i = 1, 2, \dots$ :

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

$$\hat{\mathbf{d}}(i) = \frac{1}{i} \sum_{k=1}^i \mathbf{x}(k)y^*(k)$$

**Step 1** from Table II  
**Step 2** from Table II  
**Step 3** from Table II  
**Steering Vector and Weight Vector Estimations**  
 $\alpha_{\hat{\mathbf{a}}_1}(i) = [\lambda(\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)) - \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i) + \mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i) + \eta_{\hat{\mathbf{a}}_1}\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)] / [\hat{\sigma}_1^2(i)\mathbf{p}_{\hat{\mathbf{a}}_1}^H(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)]$   
 $\alpha_v(i) = \frac{\lambda(\mathbf{p}_v^H(i)\mathbf{g}_v(i-1) - \mathbf{p}_v^H(i)\hat{\mathbf{a}}_1(i)) - \eta_v\mathbf{p}_v^H(i)\mathbf{g}_v(i-1)}{\mathbf{p}_v^H(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_v(i)}$   
 $\hat{\mathbf{a}}_1(i) = \hat{\mathbf{a}}_1(i-1) + \alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$   
 $\mathbf{v}(i) = \mathbf{v}(i-1) + \alpha_v(i)\mathbf{p}_v(i)$   
 $\mathbf{g}_{\hat{\mathbf{a}}_1}(i) = (1-\lambda)\mathbf{v}(i) + \lambda\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1) + \hat{\sigma}_1^2(i)\alpha_{\hat{\mathbf{a}}_1}(i)\mathbf{v}(i)\mathbf{v}^H(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i) - \mathbf{x}(i)\mathbf{x}^H(i)\hat{\mathbf{a}}_1(i)$   
 $\mathbf{g}_v(i) = (1-\lambda)\hat{\mathbf{a}}_1(i) + \lambda\mathbf{g}_v(i-1) - \alpha_v(i)(\hat{\mathbf{R}}(i) - \hat{\sigma}_1^2(i)\hat{\mathbf{a}}_1(i)\hat{\mathbf{a}}_1^H(i))\mathbf{p}_v(i) - \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{v}(i-1)$   
 $\beta_{\hat{\mathbf{a}}_1}(i) = \frac{[\mathbf{g}_{\hat{\mathbf{a}}_1}(i) - \mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)]^H \mathbf{g}_{\hat{\mathbf{a}}_1}(i)}{\mathbf{g}_{\hat{\mathbf{a}}_1}^H(i-1)\mathbf{g}_{\hat{\mathbf{a}}_1}(i-1)}$   
 $\beta_v(i) = \frac{[\mathbf{g}_v(i) - \mathbf{g}_v(i-1)]^H \mathbf{g}_v(i)}{\mathbf{g}_v^H(i-1)\mathbf{g}_v(i-1)}$   
 $\mathbf{p}_{\hat{\mathbf{a}}_1}(i+1) = \mathbf{g}_{\hat{\mathbf{a}}_1}(i) + \beta_{\hat{\mathbf{a}}_1}(i)\mathbf{p}_{\hat{\mathbf{a}}_1}(i)$   
 $\mathbf{p}_v(i+1) = \mathbf{g}_v(i) + \beta_v(i)\mathbf{p}_v(i)$   
 $\mathbf{w}(i) = \frac{\mathbf{v}(i)}{\hat{\mathbf{a}}_1^H(i)\mathbf{v}(i)}$

---

End snapshot

The initial guess  $\hat{\mathbf{a}}_1(0)$  can be described as the true steering vector plus a guess error vector (i.e.,  $\epsilon$ ):

$$\hat{\mathbf{a}}_1(0) = \mathbf{a}_1 + \epsilon. \quad (68)$$

Taking expectation of both sides of the above, we have

$$E\{\hat{\mathbf{a}}_1(0)\} = \mathbf{a}_1 + E\{\epsilon\}. \quad (69)$$

Substituting (68) into (67), taking into account that the accumulated estimation error is uncorrelated with the initial guess error and simplifying the expression, we obtain

$$\begin{aligned} \text{MSE}\{\hat{\mathbf{a}}_1(i)\} &= E\{\epsilon^H\epsilon\} + E\{\epsilon^H\}E\{\hat{\mathbf{e}}(i)\} + E\{\hat{\mathbf{e}}^H(i)\}E\{\epsilon\} + E\{\hat{\mathbf{e}}^H(i)\hat{\mathbf{e}}(i)\}. \end{aligned} \quad (70)$$

Furthermore, it should be emphasized that both  $\epsilon$  and  $\hat{\mathbf{e}}(i)$  are in vector forms, which means that their second-order statistics can be re-expressed in terms of their first-order statistics of their Euclidean norms. Then we can re-express (70) as

$$\begin{aligned} \text{MSE}\{\hat{\mathbf{a}}_1(i)\} &= E\{\|\epsilon\|^2\} + E\{\|\hat{\mathbf{e}}(i)\|^2\} + 2E\{\epsilon^H\}E\{\hat{\mathbf{e}}(i)\}. \end{aligned} \quad (71)$$

Since both  $\|\epsilon\|$  and  $\|\hat{\mathbf{e}}(i)\|$  are scalars we have

$$E\{\|\epsilon\|^2\} = \text{Var}\{\|\epsilon\|\} + E^2\{\|\epsilon\|\}, \quad (72)$$

$$E[\|\hat{\epsilon}(i)\|^2] = \text{Var}[\|\hat{\epsilon}(i)\|] + E^2[\|\hat{\epsilon}(i)\|]. \quad (73)$$

At this stage, we can employ Popoviciu's inequality [37] to obtain the upper bounds for the expectations of the variances of the random vectors  $\epsilon$  and  $\hat{\epsilon}(i)$ , which are given by

$$\text{Var}[\|\epsilon\|] \leq \frac{(\sup\|\epsilon\| - \inf\|\epsilon\|)^2}{4}, \quad (74)$$

$$\text{Var}[\|\hat{\epsilon}(i)\|] \leq \frac{(\sup\|\hat{\epsilon}(i)\| - \inf\|\hat{\epsilon}(i)\|)^2}{4}. \quad (75)$$

However, the last term in (71) is not analytical when conducting a norm analysis. Actually,  $E[\epsilon]$  depends on how the presumed sector is chosen: if the sector is chosen in an unbiased manner (i.e., the true steering vector lies in the centre of the sector), then we have  $E[\epsilon] = \mathbf{0}$  by symmetry criterion, in which case we can omit the last terms of (71). For convenience of carrying out the norm analysis as the next step, we focus on the unbiased case only, so that the MSE only depends on the expectation, the infimum and the supremum of  $\|\epsilon\|$  and  $\|\hat{\epsilon}(i)\|$ . In Fig. 1, we utilize Euclidean geometry to illustrate the relationships among the norms of the errors and the norm of the steering vector, which is a fixed parameter due to the re-normalization procedure after it is estimated each time.

According to Fig. 1, we can use  $\theta$  (i.e., half of the angular sector, assumed less than  $\pi/4$ ) and  $\|\mathbf{a}_1\|$  to obtain  $E[\|\epsilon\|]$  by the following (any angular parameter appeared in the equations should be measured in radians rather than degrees):  $\|\epsilon\|$  is equivalent to the chord length which corresponds to the arc of a variable  $\tau$ , which can be any value from 0 to  $\theta$  with equal probability, in other words, the choice of  $\tau$  is uniformly distributed within  $[0, \theta]$ . If the sample size of the selected  $\epsilon$  is large enough, we can approximately describe its probability density function (pdf) as a continuous function given by

$$f(\tau) = \frac{1}{\theta}. \quad (76)$$

Meanwhile, we are also able to calculate the chord length  $\|\epsilon\|$  from a simple geometric criterion as

$$\|\epsilon\| = 2\|\mathbf{a}_1\| \sin \frac{\tau}{2}. \quad (77)$$

Then the expectation of  $\|\epsilon\|$  can be computed by

$$E[\|\epsilon\|] = \int_0^\theta \|\epsilon\| f(\tau) d\tau, \quad (78)$$

from which after simplification we obtain

$$E[\|\epsilon\|] = \frac{8\|\mathbf{a}_1\| \sin^2 \frac{\theta}{4}}{\theta}. \quad (79)$$

At this point, we can also compute the variance of  $\|\epsilon\|$  by using (79) as

$$\text{Var}[\|\epsilon\|] = \int_0^\theta (\|\epsilon\| - E[\|\epsilon\|])^2 f(\tau) d\tau, \quad (80)$$

from which after simplification we obtain

$$\text{Var}[\|\epsilon\|] = 2\|\mathbf{a}_1\|^2 \left(1 - \frac{\sin \theta}{\theta} - \frac{32 \sin^4 \frac{\theta}{4}}{\theta^2}\right). \quad (81)$$

In addition, it is clear that we have  $\inf\|\epsilon\| = 0$  and  $\sup\|\epsilon\| = 2\|\mathbf{a}_1\| \sin \frac{\theta}{2}$ , which can be substituted in (74) and result in

$$\text{Var}[\|\epsilon\|] \leq \|\mathbf{a}_1\|^2 \sin^2 \frac{\theta}{2}. \quad (82)$$

We can see that the right-hand side of (81) satisfies the inequality in (82). After substituting (79) and (81) in (72), we obtain

$$E[\|\epsilon\|^2] = 2\|\mathbf{a}_1\|^2 \left(1 - \frac{\sin \theta}{\theta}\right). \quad (83)$$

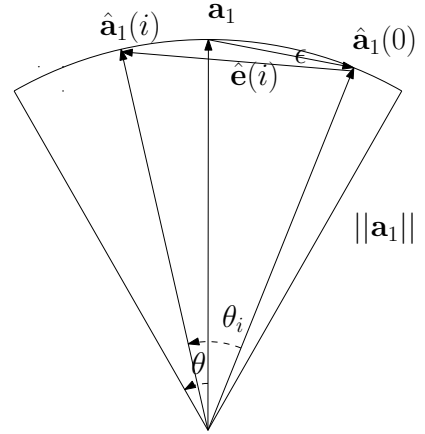


Fig. 1. Euclidean norm interpretation of the MSE

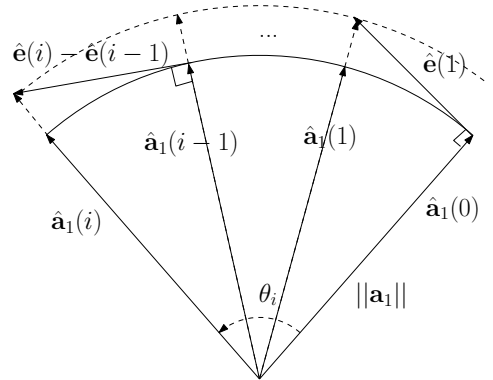


Fig. 2. update scheme of the SQP method

Regarding the computation of the norm of the accumulated estimation error  $\|\hat{\epsilon}(i)\|$ , we need to emphasize that even though the steering vector is always re-normalized each time after it is updated, the piecewise estimation error in each snapshot does not directly update the steering vector to its normalized version, which means it is inappropriate to calculate the estimation error by geometric methods directly from Fig. 1 because the accumulated estimation error partially comes from the unnormalized steering vectors. However, we can obtain the infimum and supremum values for  $\|\hat{\epsilon}(i)\|$  if we assume the update scheme is



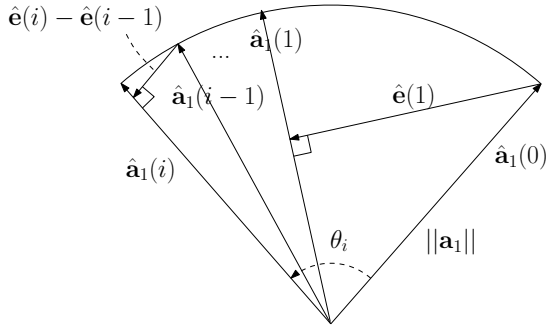


Fig. 3. update scheme of the OKSPME method

unidirectional (i.e., the steering vector is updated from  $\hat{\mathbf{a}}_1(0)$  to  $\hat{\mathbf{a}}_1(i)$  in a single direction within the sector), with the unnormalized steering vectors considered.

We firstly look at the SQP method scenario in [6]. The steering vector update scheme is shown in Fig. 2. It is necessary to emphasize that now we focus on the angular sector range of  $\theta_i$  (i.e., the angle difference between the initially guessed steering vector and its estimate in the  $i$ th snapshot) rather than  $\theta$ . In [6], an online-optimization program was used to iteratively solve for the piecewise estimation error in every snapshot, which was always orthogonal to the current steering vector estimate. Let us consider that at each time instant the steering vector is updated, its direction changes by  $\theta_{i,k}$ , where  $i$  is the snapshot index and  $k$  ( $1 \leq k \leq i$ ) is the index for the  $k$ th update. Since the total direction change in a snapshot is  $\theta_i$ , then we have

$$\theta_i = \sum_{k=1}^i \theta_{i,k}, \quad (84)$$

and the norm of the accumulated estimation error is no greater than the sum of the norms of all the piecewise estimation errors, which is given by the inequality

$$\|\hat{\mathbf{e}}(i)\| \leq \sum_{k=1}^i \|\mathbf{a}_1\| \tan \theta_{i,k}. \quad (85)$$

If we assume  $\theta_i$  is less than  $\pi/2$ , then the right-hand side of (85) achieves its maximum value when  $\theta_{i,k} = \tan \theta_i$ , which is also the supremum of  $\|\hat{\mathbf{e}}(i)\|$  and equals

$$\|\hat{\mathbf{e}}(i)\|_{max} = \|\mathbf{a}_1\| \tan \theta_i. \quad (86)$$

On the other hand, we notice that the piecewise estimation error vector can never enter into the angular sector, but at most move along with the arc if the number of iterations is large enough. In this case, we can approximately and geometrically illustrate the arc length corresponding with  $\theta_i$  as the lower bound by taking the limit  $i \rightarrow \infty$ , i.e.,

$$\lim_{i \rightarrow \infty} \|\hat{\mathbf{e}}(i)\| = \theta_i \|\mathbf{a}_1\|, \quad (87)$$

which is actually the infimum of  $\|\hat{\mathbf{e}}(i)\|$  and cannot be achieved since the number of snapshots or iterations are always limited in practical situations. By combining (86) and (87),  $\|\hat{\mathbf{e}}(i)\|$  is

bounded by

$$\inf \|\hat{\mathbf{e}}(i)\| = \theta_i \|\mathbf{a}_1\| < \|\hat{\mathbf{e}}(i)\| \leq \|\mathbf{a}_1\| \tan \theta_i = \sup \|\hat{\mathbf{e}}(i)\|. \quad (88)$$

Different from the SQP method, the proposed OKSPME method utilizes the Krylov subspace and the cross-correlation vector projection approach to extract the error information then use it to update the steering vector. From (9) we have

$$\begin{aligned} \hat{\mathbf{d}}(i) &= \frac{1}{i} \sum_{k=1}^i \mathbf{x}(k) y^*(k) = \frac{1}{i} \sum_{k=1}^i \mathbf{x}(k) (\mathbf{w}^H(k) \mathbf{x}(k))^* \\ &= \frac{1}{i} \sum_{k=1}^i \mathbf{x}(k) \mathbf{x}^H(k) \mathbf{w}(k) = \frac{1}{i} \sum_{k=1}^i \hat{\mathbf{R}}(k) \mathbf{w}(k). \end{aligned} \quad (89)$$

Note that an initialization for vector  $\hat{\mathbf{d}}$  or matrix  $\hat{\mathbf{R}}$  should be considered to ensure  $\hat{\mathbf{R}}$  is full-rank and invertible, which can be done by either setting  $\hat{\mathbf{d}}(0) = \delta \mathbf{I} \mathbf{w}(0)$  or  $\hat{\mathbf{R}}(0) = \delta \mathbf{I}$ . We also know that

$$\mathbf{w}(k) = \frac{\hat{\mathbf{R}}^{-1}(k) \hat{\mathbf{a}}_1(k)}{\hat{\mathbf{a}}_1^H(k) \hat{\mathbf{R}}^{-1}(k) \hat{\mathbf{a}}_1(k)} = \frac{\hat{\mathbf{R}}^{-1}(k) \hat{\mathbf{a}}_1(k)}{\hat{\sigma}_1^2(k)}. \quad (90)$$

Pre-multiplying (90) by  $\hat{\mathbf{R}}(k)$  on both sides we obtain

$$\hat{\mathbf{R}}(k) \mathbf{w}(k) = \frac{\hat{\mathbf{a}}_1(k)}{\hat{\sigma}_1^2(k)}, \quad (91)$$

which is then substituted in (89) and results in

$$\hat{\mathbf{d}}(i) = \frac{1}{i} \sum_{k=1}^i \frac{\hat{\mathbf{a}}_1(k)}{\hat{\sigma}_1^2(k)}, \quad (92)$$

where  $\hat{\sigma}_1^2(k)$  is a scalar, which means the SCV contains the direction of the desired signal steering vector. Projecting  $\hat{\mathbf{d}}(i)$  onto the Krylov subspace represented by  $\hat{\mathbf{P}}(i)$  is therefore similar to projecting  $\hat{\mathbf{a}}_1(i)$ . In our method, the estimation of  $\hat{\mathbf{d}}(i)$  is separate from the update of  $\hat{\mathbf{a}}_1(i)$ , which means the steering vector estimation error used for the updates is obtained from  $\hat{\mathbf{d}}(i)$ , so that in the  $k$ th ( $1 \leq k < i$ ) snapshot, the error does not have to be orthogonal to  $\hat{\mathbf{a}}_1(k)$ , but should be orthogonal to another potentially better estimate  $\hat{\mathbf{a}}_1(j)$  ( $1 \leq k < j \leq i$ ), resulting in a situation where the error is located inside the sector (see Fig. 3). There are two benefits in the case which the error is inside the sector: faster convergence rate and smaller estimation error. We can obtain the infimum and supremum values in a similar way. By applying the inequality that the norm of the accumulated estimation error is no greater than the sum of the norms of all the piecewise estimation errors, we have

$$\|\hat{\mathbf{e}}(i)\| \leq \sum_{k=1}^i \|\mathbf{a}_1\| \sin \theta_{i,k}, \quad (93)$$

where the parameters  $\theta_{i,k}$  ( $k = 1, 2, \dots, i$ ) satisfy the constraint in (84). However, the right-hand side of (93) achieves its maximum value when all these parameters are equal (i.e.,  $\theta_{i,1} = \theta_{i,2} = \dots = \theta_{i,i} = \frac{\theta_i}{i}$ ) and it is given by

$$\|\hat{\mathbf{e}}(i)\|_{max} = i \|\mathbf{a}_1\| \sin \frac{\theta_i}{i}, \quad (94)$$

The right-hand side of (94) can be treated as a function of  $i$  which is an increasing function on  $i = 1, 2, \dots, \infty$ . Therefore, we can take the limit of it to obtain the upper bound of  $\|\hat{\epsilon}(i)\|_{max}$ , and so as to  $\|\hat{\epsilon}(i)\|$ . In fact, when  $i \rightarrow \infty$ , the piecewise estimation error moves along the arc corresponding with  $\theta_i$ , resulting in the upper bound obtained is the same as the lower bound of the SQP method case, which is given by the right-hand side expression of (87) and defines the supremum of  $\|\hat{\epsilon}(i)\|$  in this case. Since we have already assumed that  $\theta_i$  is less than  $\pi/2$  so that  $\hat{\epsilon}(i)$  must be inside of the angular sector but its Euclidean norm cannot be smaller than the orthogonal distance between  $\hat{\mathbf{a}}_1(0)$  to  $\hat{\mathbf{a}}_1(i)$ , so this orthogonal distance can define the lower bound of  $\|\hat{\epsilon}(i)\|$ , which is actually the infimum and calculated by  $\|\mathbf{a}_1\| \sin \theta_i$ . Then, in the OKSPME method,  $\|\hat{\epsilon}(i)\|$  is bounded by

$$\inf \|\hat{\epsilon}(i)\| = \|\mathbf{a}_1\| \sin \theta_i \leq \|\hat{\epsilon}(i)\| < \theta_i \|\mathbf{a}_1\| = \sup \|\hat{\epsilon}(i)\|. \quad (95)$$

By taking expectations of both (88) and (95), we obtain

$$E[\theta_i] \|\mathbf{a}_1\| < \{E[\|\hat{\epsilon}(i)\|]\}_{SQP} \leq \|\mathbf{a}_1\| \tan(E[\theta_i]), \quad (96)$$

$$\|\mathbf{a}_1\| \sin(E[\theta_i]) \leq \{E[\|\hat{\epsilon}(i)\|]\}_{OKSPME} < E[\theta_i] \|\mathbf{a}_1\|. \quad (97)$$

On the other side, by substituting (88) and (95) in (75), we obtain

$$0 \leq \{\text{Var}[\|\hat{\epsilon}(i)\|]\}_{SQP} \leq \frac{\|\mathbf{a}_1\|^2 (\tan \theta_i - \theta_i)^2}{4}, \quad (98)$$

$$0 \leq \{\text{Var}[\|\hat{\epsilon}(i)\|]\}_{OKSPME} \leq \frac{\|\mathbf{a}_1\|^2 (\theta_i - \sin \theta_i)^2}{4}. \quad (99)$$

Substituting (96), (98) and (97), (99) in (73), respectively, we have

$$\begin{aligned} E^2[\theta_i] \|\mathbf{a}_1\|^2 &< \{E[\|\hat{\epsilon}(i)\|^2]\}_{SQP} \\ &\leq \frac{\|\mathbf{a}_1\|^2 (\tan \theta_i - \theta_i)^2}{4} + \|\mathbf{a}_1\|^2 \tan^2(E[\theta_i]), \end{aligned} \quad (100)$$

$$\begin{aligned} \|\mathbf{a}_1\|^2 \sin^2(E[\theta_i]) &\leq \{E[\|\hat{\epsilon}(i)\|^2]\}_{OKSPME} \\ &< \frac{\|\mathbf{a}_1\|^2 (\theta_i - \sin \theta_i)^2}{4} + E^2[\theta_i] \|\mathbf{a}_1\|^2. \end{aligned} \quad (101)$$

However,  $E[\theta_i]$  also has its lower and upper bounds. Since our analysis focuses on the unbiased case only as mentioned, the true steering vector is located in the center of the angular sector and the estimate  $\hat{\mathbf{a}}_1(i)$  is always closer to the center than  $\hat{\mathbf{a}}_1(0)$ . Let us assume that even if the estimate  $\hat{\mathbf{a}}_1(i)$  always happens to be very close to either edge of the sector, no matter how  $\hat{\mathbf{a}}_1(0)$  is chosen within the sector,  $\theta_i$  will vary from 0 to  $2\theta$  with equal probability, or equivalently, uniformly distributed within  $[0, 2\theta)$ , in which case we can obtain the upper bound for  $E[\theta_i]$  by taking the average between 0 to  $2\theta$ , which is obtained as  $\theta$ . On the other hand, if we assume that the estimate  $\hat{\mathbf{a}}_1(i)$  always happens to be exactly at the center of the sector, resulting in that  $\theta_i$  can only vary from 0 to  $\theta$ , or uniformly distributed within  $[0, \theta]$  in which case  $E[\theta_i] = \theta/2$ , resulting in the lower bound of  $E[\theta_i]$  is  $\theta/2$ . Therefore, the upper and lower bounds for  $\text{MSE}\{\hat{\mathbf{a}}_1(i)\}$  can be further obtained by substituting  $E[\theta_i]_{max} \rightarrow \theta$ ,  $[\theta_i]_{max} \rightarrow 2\theta$

and  $E[\theta_i]_{min} = \theta/2$ ,  $[\theta_i]_{min} = 0$  into the upper and lower bounds of (100) and (101) respectively, resulting in

$$\begin{aligned} \frac{\theta^2}{4} \|\mathbf{a}_1\|^2 &< \{E[\|\hat{\epsilon}(i)\|^2]\}_{SQP} \\ &< \frac{\|\mathbf{a}_1\|^2 (\tan 2\theta - 2\theta)^2}{4} + \|\mathbf{a}_1\|^2 \tan^2 \theta, \end{aligned} \quad (102)$$

$$\begin{aligned} \|\mathbf{a}_1\|^2 \sin^2 \frac{\theta}{2} &\leq \{E[\|\hat{\epsilon}(i)\|^2]\}_{OKSPME} \\ &< \frac{\|\mathbf{a}_1\|^2 (2\theta - \sin 2\theta)^2}{4} + \theta^2 \|\mathbf{a}_1\|^2. \end{aligned} \quad (103)$$

Finally, by combining the expectation of the mean-squared initial guess error  $E[\|\epsilon\|^2]$  in (83) with (102) and (103), we obtain the bounds for the MSE of the steering vector estimate  $\text{MSE}\{\hat{\mathbf{a}}_1(i)\}$  as

$$\begin{aligned} (2 - \frac{2 \sin \theta}{\theta} + \frac{\theta^2}{4}) \|\mathbf{a}_1\|^2 &< \{\text{MSE}\{\hat{\mathbf{a}}_1(i)\}\}_{SQP} \\ &< (2 - \frac{2 \sin \theta}{\theta} + \frac{(\tan 2\theta - 2\theta)^2}{4} + \tan^2 \theta) \|\mathbf{a}_1\|^2, \end{aligned} \quad (104)$$

$$\begin{aligned} (2 - \frac{2 \sin \theta}{\theta} + \sin^2 \frac{\theta}{2}) \|\mathbf{a}_1\|^2 &\leq \{\text{MSE}\{\hat{\mathbf{a}}_1(i)\}\}_{OKSPME} \\ &< (2 - \frac{2 \sin \theta}{\theta} + \frac{(2\theta - \sin 2\theta)^2}{4} + \theta^2) \|\mathbf{a}_1\|^2. \end{aligned} \quad (105)$$

From (104) and (105), we can see that the MSEs now only depend on two parameters: the norm of the true steering vector and the angular sector spread. The lower and upper bounds of the proposed OKSPME method are lower than those of the SQP method. As mentioned before, it is important that the presumed angular sector spread  $2\theta$  must be less than  $\pi/2$  (i.e.,  $90^\circ$ ) to ensure the previous assumption of  $\theta_i < \pi/2$  is always valid.

## B. Complexity Analysis

The computational complexity analysis is discussed in this subsection. We measure the total number of additions and multiplications (i.e., flops) in terms of the number of sensors  $M$  performed for each snapshot for the proposed algorithms and the existing ones and list them in Table V. Note that the SQP method in [6] has a highly-variant computational complexity in different snapshots, due to the online optimization program based on random choices of the presumed steering vector. However, it is usually in  $\mathcal{O}(M^{3.5})$ . The complexity of the LCWC algorithm in [9] often requires a much larger  $n$  than that in the proposed LOCSME-CCG algorithm. It is obvious that all of the proposed algorithms have their complexity depending on the Krylov subspace model order  $m$ , which is determined from Table I and is no larger than  $K + 1$ . For the convenience of comparison, we eliminate all parameters except  $M$  by setting them to common values (the values of  $n$  in LCWC and OKSPME-CCG is set to 50 and 5 respectively,  $m = K + 1$  where  $K = 3$ ) and illustrate their complexity with  $M$  varying from 10 to 100 as shown in Fig. 4. As can be seen that the proposed OKSPME-SG and OKSPME-MCG algorithms have lower complexity than the other algorithms.

TABLE V  
COMPLEXITY COMPARISON

RAB Algorithms	Flops
LOCSME [32]	$4M^3 + 3M^2 + 20M$
RCB [4]	$2M^3 + 11M^2$
SQP [6]	$\mathcal{O}(M^{3.5})$
LOCME [7]	$2M^3 + 4M^2 + 5M$
LCWC [9]	$2nM^2 + 7nM$
OKSPME	$M^3 + (4m + 11)M^2 + (3m^2 + 5m + 20)M$
OKSPME-SG	$(4m + 7)M^2 + (3m^2 + 5m + 33)M$
OKSPME-CCG	$(4m + 8n + 8)M^2 + (3m^2 + 5m + 33n + 29)M$
OKSPME-MCG	$(4m + 14)M^2 + (3m^2 + 5m + 86)M$

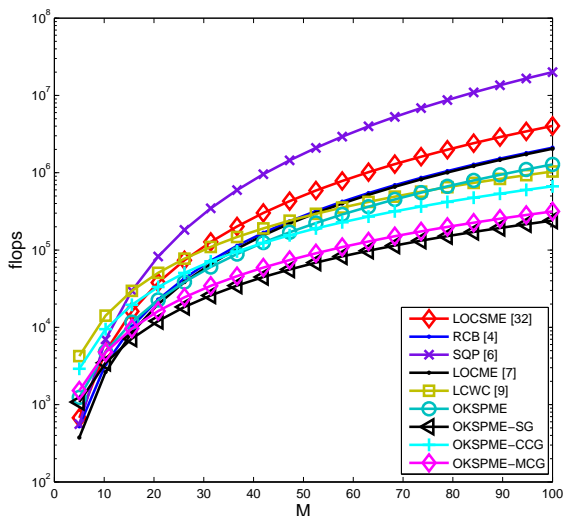


Fig. 4. Complexity Comparison

## VI. SIMULATIONS

In this section, we present and discuss the simulation results of the proposed RAB algorithms by comparing them to some of the existing RAB techniques. We consider uniform linear arrays (ULA) of omnidirectional sensors with half wavelength spacing. To produce all the figures (if unspecified in a few scenarios), 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  $\theta_1 = 10^\circ$ . The signal-to-interference ratio (SIR) is fixed at 0dB. As the prior knowledge, the angular sector in which the desired signal is assumed to be located is chosen as  $[\theta_1 - 5^\circ, \theta_1 + 5^\circ]$ . The results focus on the beamformer output SINR performance versus the number of snapshots, or a variation of input SNR ( $-10$ dB to  $30$ dB) and both coherent and incoherent local scattering mismatch [5] scenarios are considered.

### A. Mismatch due to Coherent Local Scattering

All simulations in this subsection consider coherent local scattering. With time-invariant coherent local scattering, the

steering vector of the desired signal is modeled as

$$\mathbf{a}_1 = \mathbf{p} + \sum_{k=1}^4 e^{j\varphi_k} \mathbf{b}(\theta_k), \quad (106)$$

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  $10^\circ$  and standard deviation  $2^\circ$ . The angles  $\varphi_k$  ( $k = 1, 2, 3, 4$ ) are independently and uniformly taken from the interval  $[0, 2\pi]$  in each simulation run. Both  $\theta_k$  and  $\varphi_k$  change from trials while remaining constant over snapshots.

We firstly compare our proposed methods with some classical RAB methods (i.e., standard diagonal loading method with a fixed loading factor equal to 10 times the noise variance, the RCB method in [4] which estimates the loading factor iteratively, and the method that solves an online quadratic optimization programming, which refers to the SQP method [6]). The numbers of sensors and signal sources (including the desired signal) are set to  $M = 10$  and  $K = 3$ , respectively. For this case only, we set the INR (interferences-to-noise ratio) to 20dB and illustrate the SINR performance versus snapshots within 100 snapshots in Fig. 5. The two interferers are arranged to be in the directions of  $\theta_2 = 30^\circ$  and  $\theta_3 = 50^\circ$ , respectively. The other user-defined parameters, if unspecified, (e.g. the step size  $\mu$  and the forgetting factor  $\lambda$ ) are manually optimized to give the best algorithm performance, which is also applied for the other simulation scenarios.

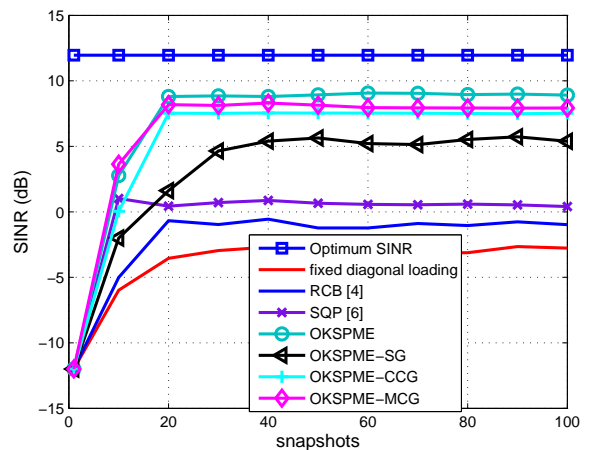


Fig. 5. Coherent local scattering, SINR versus snapshots,  $M = 10$ ,  $K = 3$ , INR = 20dB

We then set the number of sensors to  $M = 12$ , the number of signal sources as (including the desired signal)  $K = 3$  and illustrate the SINR versus snapshots and the SINR versus input SNR performance in Fig. 6 and Fig. 7 respectively. The two interferers are arranged to be in the directions of  $\theta_2 = 30^\circ$  and  $\theta_3 = 50^\circ$ , respectively. In either Fig. 6 or Fig. 7, we can see that the proposed OKSPME method has a very similar or slightly better performance compared to the LOCSME algorithm of [32] and both of them have the best performance. Furthermore, the

TABLE VI  
CHANGES OF INTERFERERS

Snapshots	Number of Interferers ( $K - 1$ )	DoAs
0 – 150	2	$\theta_2 = 30^\circ, \theta_3 = 50^\circ$ .
151 – 300	5	$\theta_2 = 20^\circ, \theta_3 = 30^\circ, \theta_4 = 40^\circ,$ $\theta_5 = 50^\circ, \theta_6 = 60^\circ$ .

proposed OKSPME-CCG and OKSPME-MCG algorithms also achieve very close performance to OKSPME.

In Fig. 8, we assess the SINR performance versus snapshots of those selected algorithms in a specific time-varying scenario which encounters a halfway redistribution of the interferers at a certain snapshot. In this case, the number of sensors is kept at  $M = 12$ , whereas the details of the interferers are given in Table VI.

In Figs. 9 and 10, we set the number of signal sources to  $K = 3$ , but increase the number of sensors from  $M = 12$  to  $M = 40$  and study the SINR versus snapshots and the SINR versus input SNR performance of the selected and proposed dimensionality reduction RAB algorithms, respectively. We set the reduced-dimension as  $D = 4$  for the beamspace based algorithm [23] in all simulations. This time, it is clear that the proposed OKSPME, OKSPME-SG, OKSPME-CCG and OKSPME-MCG algorithms all have a certain level of performance degradation compared to the scenario where  $M = 12$ . The proposed OKSPME based algorithms achieve better performances than the beamspace approach.

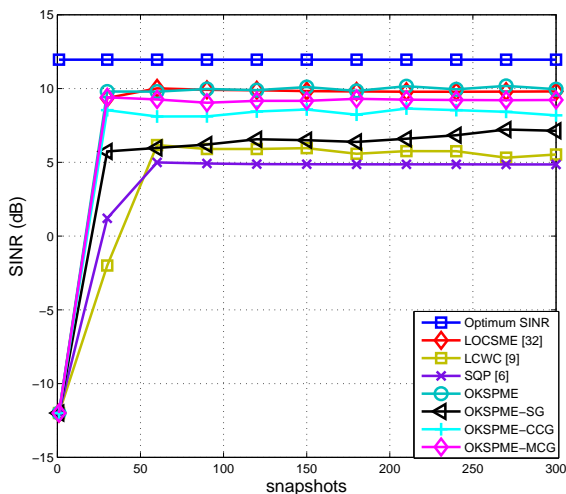


Fig. 6. Coherent local scattering, SINR versus snapshots,  $M = 12$ ,  $K = 3$

### B. Mismatch due to Incoherent Local Scattering

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

$$\mathbf{a}_1(i) = s_0(i)\mathbf{p} + \sum_{k=1}^4 s_k(i)\mathbf{b}(\theta_k), \quad (107)$$

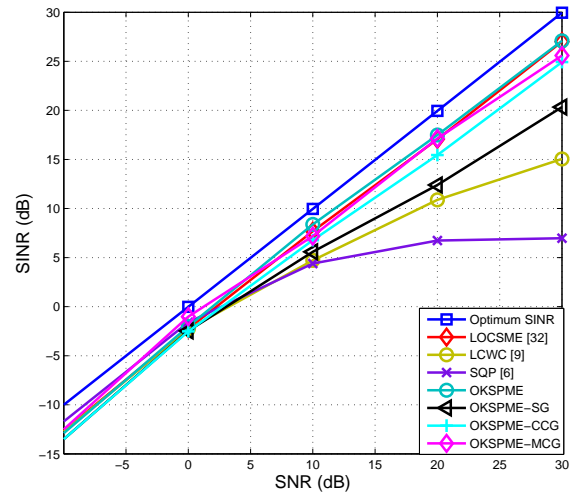


Fig. 7. Coherent local scattering, SINR versus SNR,  $M = 12$ ,  $K = 3$

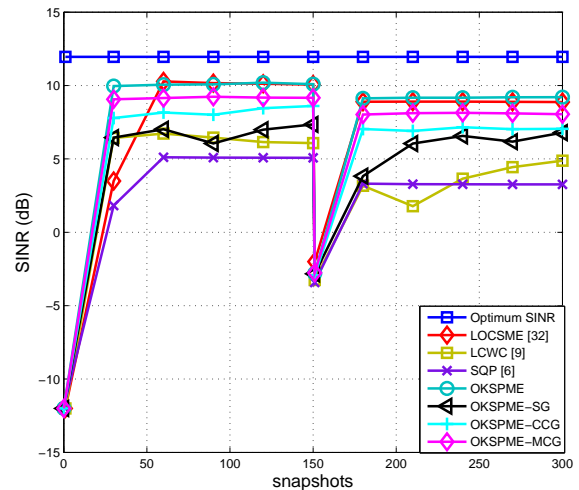


Fig. 8. Coherent local scattering, SINR versus snapshots,  $M = 12$

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  $10^\circ$  and standard deviation  $2^\circ$ . At this time,  $s_k(i)$  changes both from run to run and from snapshot to snapshot. In order to show the effects caused by incoherent scattering only, we set the parameters  $M = 40$  and  $K = 3$ , study the SINR versus SNR performance of the selected algorithms in Fig. 11 and compare the results with Fig. 10. As a result, a performance degradation is observed for all the studied algorithms. This is because the time-varying nature of incoherent scattering results in more dynamic and environmental uncertainties in the system, which increases the steering vector mismatch.

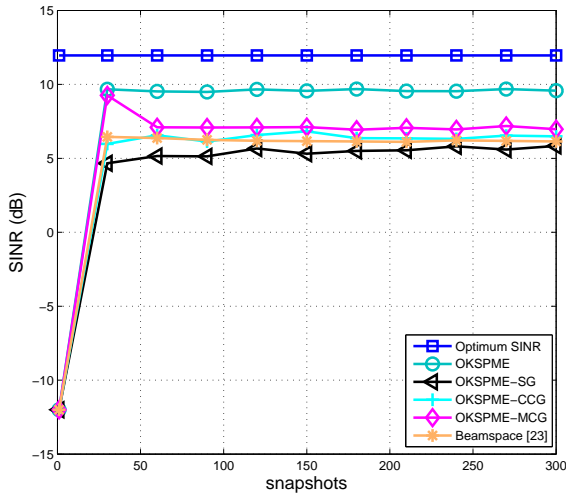


Fig. 9. Coherent local scattering, SINR versus snapshots,  $M = 40$ ,  $K = 3$

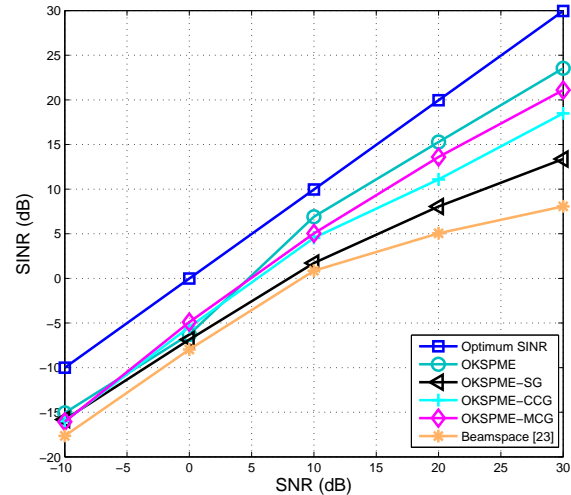


Fig. 11. Incoherent local scattering, SINR versus SNR,  $M = 40$ ,  $K = 3$

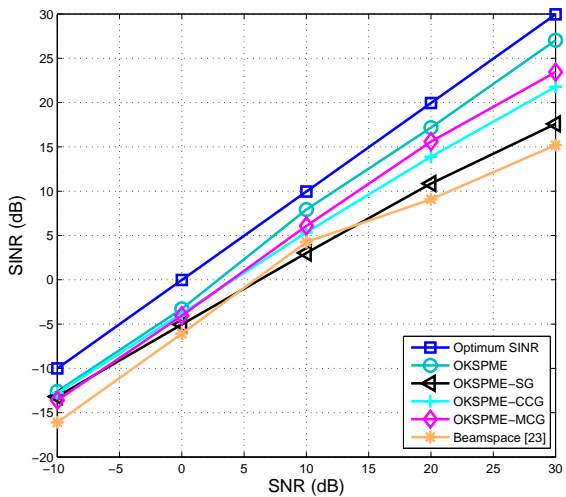


Fig. 10. Coherent local scattering, SINR versus SNR,  $M = 40$ ,  $K = 3$

## VII. CONCLUSION

We have proposed the OKSPME algorithm based on the exploitation of cross-correlation mismatch estimation and the orthogonal Krylov subspace. In addition, low complexity RAB algorithms, OKSPME-SG, OKSPME-CCG and OKSPME-MCG have been developed to enable the beamforming weights to be updated recursively without matrix inversions. A detailed steering vector estimation MSE analysis for the general RAB design approach that relies on a presumed angular sector as prior knowledge has been provided. The computational complexity of the proposed and some of the existing algorithms have been compared and discussed. Simulation results have shown that the proposed algorithms have robustness against different choices of user-defined parameters and environmental effects, and achieved excellent output SINR performance especially in medium-high input SNR values.

## REFERENCES

- [1] H. L. Van Trees, *Optimum Array Processing*, New York: Wiley, 2002.
- [2] S. A. Vorobyov, A. B. Gershman and Z. Luo, "Robust Adaptive Beamforming using Worst-Case Performance Optimization: A Solution to the Signal Mismatch Problem," *IEEE Transactions on Signal Processing*, Vol. 51, No. 4, pp 313-324, Feb 2003.
- [3] A. Khabbazi-basmenj, S. A. Vorobyov and A. Hassaniien, "Robust Adaptive Beamforming Based on Steering Vector Estimation with as Little as Possible Prior Information," *IEEE Transactions on Signal Processing*, Vol. 60, No. 6, pp 2974-2987, June 2012.
- [4] J. Li, P. Stoica and Z. Wang, "On Robust Capon Beamforming and Diagonal Loading," *IEEE Transactions on Signal Processing*, Vol. 57, No. 7, pp 1702-1715, July 2003.
- [5] D. Astely and B. Ottersten, "The effects of Local Scattering on Direction of Arrival Estimation with Music," *IEEE Transactions on Signal Processing*, Vol. 47, No. 12, pp 3220-3234, Dec 1999.
- [6] A. Hassaniien, 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.
- [7] L. Landau, R. de Lamare, M. Haardt, "Robust Adaptive Beamforming Algorithms Using Low-Complexity Mismatch Estimation," *Proc. IEEE Statistical Signal Processing Workshop*, 2011.
- [8] J. Zhuang and A. Manikas, "Interference cancellation beamforming robust to pointing errors," *IET Signal Process.*, Vol. 7, No. 2, pp. 120-127, April 2013.
- [9] 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.
- [10] 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. 686-697, Feb 2010.
- [11] 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.
- [12] A. Filimon, "Krylov Subspace Iteration Methods," May 2008.
- [13] H. A. van der Vorst, "Iterative Krylov Methods for Large Linear Systems," Cambridge University Press, 2003.
- [14] M. Morelli, L. Sanguietti and U. Mengali, "Channel Estimation for Adaptive Frequency-Domain Equalization," *IEEE Transactions on Wireless Communications*, Vol. 4, No. 5, pp. 53-57, Sep 2005.
- [15] Z. Bai, "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems," Applied Numerical Mathematics, Department of Computer Science, University of California, 2002.
- [16] R. C. de Lamare, M. Haardt and R. Sampaio-Neto, "Blind Adaptive Constrained Reduced-Rank Parameter Estimation Based on Constant Modulus Design for CDMA Interference Suppression," *IEEE Transactions on Signal Processing*, Vol. 56, No. 6, June 2008.
- [17] H. Ge, I. P. Kirsteins and L. L. Scharf, "Data Dimension Reduction Us-

- ing Krylov Subspaces Making Adaptive Beamformers Robust to Model Order-Determination,” in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 4, 2006.
- [18] I. P. Kirsteins and H. Ge, “Performance Analysis of Krylov Space Adaptive Beamformers,” *Sensor Array and Multichannel Processing, Fourth IEEE Workshop*, pp. 16-20, July 2006.
- [19] S. D. Somasundaram, N. H. Parsons, P. Li and R. C. de Lamare, “Data-Adaptive Reduced-Dimension Robust Capon Beamforming,” *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4159-4163, May 2013.
- [20] S. D. Somasundaram, N. H. Parsons, P. Li and R. C. de Lamare, “Reduced-Dimension Robust Capon Beamforming Using Krylov-Subspace Techniques,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 1, pp. 270-289, Jan 2015.
- [21] V. Druskin and V. Simoncini, “Adaptive Rational Krylov Subspaces for Large-Scale Delymical Systems,” *Systems and Control Letters*, Vol. 60, No. 8, pp. 546-560, 2011.
- [22] K. Carlberg and C. Farhat, “An Adaptive POD-Krylov Reduced-Order Model for Structural Optimization,” *8th World Congress on Structural and Multidisciplinary Optimization*, June 2009.
- [23] A. Hassaniien and S. A. Vorobyov, “A Robust Adaptive Dimension Reduction Technique With Application to Array Processing,” *IEEE Sig. Proc. Letters*. Vol. 16, No. 1, pp. 22-25, Jan 2009.
- [24] M. Yukawa, R. C. de Lamare and I. Yamada, “Robust Reduced-Rank Adaptive Algorithm Based on Parallel Subgradient Projection and Krylov Subspace,” *IEEE Transactions on Signal Processing*, Vol. 57, No. 12, Dec 2009.
- [25] C. Dumard and T. Zemen, “Double Krylov Subspace Approximation for Low Complexity Iterative Multi-User Decoding and Time-Variant Channel Estimation,” *6th IEEE Workshop on Signal Processing Advances in Wireless Communications*, pp. 328-332, June 2005.
- [26] R. C. de Lamare and R. Sampaio-Neto, “Reduced-Rank Adaptive Filtering Based on Joint Iterative Optimization of Adaptive Filters,” *IEEE Sig. Proc. Letters*. Vol. 14, No. 12, Dec 2007.
- [27] N. Song, W. U. Alokozai, R. C. de Lamare and M. Haardt, “Adaptive Widely Linear Reduced-Rank Beamforming Based on Joint Iterative Optimization,” *IEEE Sig. Proc. Letters*. Vol. 21, No. 3, Mar 2014.
- [28] S. D. Somasundaram, “A Framework for Reduced Dimension Robust Capon Beamforming,” *IEEE Statistical Signal Processing Workshop (SSP)*, 2011.
- [29] P. Li and R. C. de Lamare, “Low-Complexity Robust Data-Dependant Dimensionality Reduction Based on Joint Iterative Optimization of Parameters,” *IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 49-52, Dec 2013.
- [30] R. C. de Lamare and R. Sampaio-Neto, “Adaptive Reduced-Rank Processing Based on Joint and Iterative Interpolation, Decimation, and Filtering,” *IEEE Transactions on Signal Processing*, Vol. 57, No. 7, July 2009.
- [31] L. Wang, R. C. de Lamare and M. Haardt, “Direction finding Algorithms with Joint Iterative Subspace Optimization,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 4, pp. 2541-2553, Oct 2014.
- [32] 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, Nov 2013.
- [33] C. Zhang, R. Tian, X. Tan, “A Leakage-Based Dimensionality Reduction Beamforming for MIMO Cognitive Radio Networks,” *8th International Conference on Communications and Networking in China (CHINACOM)*, 2013.
- [34] R. Fa, R. C. de Lamare and D. Zanatta-Filho, “Reduced-Rank STAP Algorithm for Adaptive Radar Based on Joint Iterative Optimization of Adaptive Filters,” *42nd Asilomar Conference on Signals, Systems and Computers*, pp. 533-537, Oct 2008.
- [35] R. Fa, R. C. de Lamare and P. Clarke, “Reduced-Rank STAP for MIMO Radar Based on Joint Iterative Optimization of Knowledge-Aided Adaptive Filters,” *Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp. 496-500, Nov 2009.
- [36] S.D. Somasundaram, “Reduced dimension robust Capon beamforming for large aperture passive sonar arrays,” *IET Radar, Sonar and Navigation*, Vol. 5, No. 7, pp. 707-715, 2011.
- [37] Popoviciu, “Sur Les Equations Algebriques Ayant Toutes Leurs Racines Relles,” *Mathematica* 9, pp. 129-145, 1935.