Adaptive Reduced-Rank LCMV Beamforming Algorithms Based on Joint Iterative Optimization of Filters: Design and Analysis

R. C. de Lamare, L. Wang and R. Fa

Abstract

This paper presents reduced-rank linearly constrained minimum variance (LCMV) beamforming algorithms based on joint iterative optimization of filters. The proposed reduced-rank scheme is based on a constrained joint iterative optimization of filters according to the minimum variance criterion. The proposed optimization procedure adjusts the parameters of a projection matrix and an adaptive reduced-rank filter that operates at the output of the bank of filters. We describe LCMV expressions for the design of the projection matrix and the reduced-rank filter. We then describe stochastic gradient and develop recursive least-squares adaptive algorithms for their efficient implementation along with automatic rank selection techniques. An analysis of the stability and the convergence properties of the proposed algorithms is presented and semi-analytical expressions are derived for predicting their mean squared error (MSE) performance. Simulations for a beamforming application show that the proposed scheme and algorithms outperform in convergence and tracking the existing full-rank and reduced-rank algorithms while requiring comparable complexity.

Keywords

Adaptive filters, beamforming, constrained optimization, iterative methods.

I. Introduction

In recent years, adaptive beamforming techniques have attracted considerable interest and found applications in radar, wireless communications and sonar [1], [2]. The adaptive beamforming techniques are used in systems equipped with antenna arrays and usually have a trade-off between performance and computational complexity which depends on the designer's choice of the adaptation algorithm [3], [6], [7]. The optimal linearly constrained minimum variance (LCMV) beamformer is designed in such a way that it attempts to minimize the array output power while maintaining a constant response in the direction of a signal of interest (SoI) [1], [2], [3]. However, this technique requires the computation of the inverse of the input data covariance matrix and the knowledge of the cross-correlation vector, rendering the method very complex for practical applications when the system is large. Adaptive versions of the LCMV beamformer were subsequently reported with stochastic gradient (SG) [4] and recursive least squares (RLS) [8] algorithms.

These algorithms require estimates of the input data covariance matrix, which is a task that may become challenging in large systems and in highly dynamic situations such as those found in wireless communications and radar applications. This is because the convergence speed and tracking properties of adaptive filters depend on the number of sensor

elements M [3] and on the eigenvalue spread of the input data covariance matrix. Given this dependency on the number of sensor elements M, it is thus intuitive to reduce M while simultaneously extracting the key features of the original signal via an appropriate transformation.

A cost-effective technique in short-data record scenarios and, in particular, with systems containing a large number of parameters is reduced-rank signal processing. The advantages are their superior convergence properties and enhanced tracking performance when compared with full-rank schemes operating with a large number of parameters, and their ability to exploit the low-rank nature of the signals encountered in beamforming applications. Several reduced-rank methods have been proposed to generate the signal subspace [3]-[16]. They range from computationally expensive eigen-decomposition techniques [9]-[11] to alternative approaches such as the auxiliary-vector filter (AVF) [12],[13], [18], the multistage Wiener filter (MSWF) [13], [14], [16], [17] which are based on the Krylov subspace, and joint optimization approaches [15], [19]. Despite the improved convergence and tracking performance achieved with Krylov methods [12]-[14], [16]-[18] they are relatively complex to implement and can suffer from numerical problems. The joint optimization techniques reported in [15], [19] outperform the eigen-decomposition- and Krylov-based methods and are amenable to efficient adaptive implementations. However, the design and analysis of adaptive LCMV reduced-rank algorithms based on joint optimization approaches have not been considered so far.

This work proposes LCMV reduced-rank algorithms based on constrained joint iterative optimization of filters for antenna-array beamforming. The proposed scheme, whose initial results were reported in [20], [21], jointly optimizes a projection matrix and a reduced-rank filter that operates at the output of the projection matrix. The essence of the proposed approach is to change the role of adaptive LCMV filters. The bank of adaptive filters is responsible for performing dimensionality reduction, whereas the reduced-rank filter effectively forms the beam in the direction of the SoI. We describe LCMV expressions for the design of the projection matrix and the reduced-rank filter and present SG and RLS algorithms for efficiently implementing the method. We also introduce an automatic rank estimation algorithm for determining the most adequate rank for the proposed algorithms. An analysis of the stability and the convergence properties of the proposed algorithms is presented and semi-analytical expressions are derived for predicting their performance.

This paper is organized as follows. The system model is described in Section II. The full-rank and the reduced-rank LCMV filtering problems are formulated in Section III. Section IV is dedicated to the proposed method, whereas Section V is devoted to the derivation of the adaptive SG and RLS algorithms and the rank adaptation technique. Section VI focuses on the analysis of the proposed algorithms. Section VII presents and discusses the simulation results and Section VIII gives the concluding remarks.

II. SYSTEM MODEL

Let us consider a smart antenna system equipped with a uniform linear array (ULA) of M elements, as shown in Fig. 1. Assuming that the sources are in the far field of the array, the signals of K narrowband sources impinge on the array ($K < M$) with unknown directions of arrival (DOA) θ_l for $l = 1, 2, \ldots, K$.

The input data from the antenna array can be organized in an $M \times 1$ vector expressed by

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

where

$$\boldsymbol{A}(\theta) = [\boldsymbol{a}(\theta_1), \dots, \boldsymbol{a}(\theta_K)]$$

is the $M \times K$ matrix of signal steering vectors. The $M \times 1$ signal steering vector is defined as

$$\boldsymbol{a}(\theta_l) = \left[1, e^{-2\pi j \frac{d_s}{\lambda_c}\cos\theta_l}, \dots, e^{-2\pi j(M-1)\frac{d_s}{\lambda_c}\cos\theta_l}\right]^T \tag{2}$$

for a signal impinging at angle θ_l, $l = 1, 2, \dots, K$, where $d_s = \lambda_c/2$ is the inter-element spacing, λ_c is the wavelength and $(.)^T$ denotes the transpose operation. The vector $\boldsymbol{n}(i)$ denotes the complex vector of sensor noise, which is assumed to be zero-mean and Gaussian with covariance matrix $\sigma^2\boldsymbol{I}$.

III. PROBLEM STATEMENT

In this section, we formulate the problems of full-rank and reduced-rank LCMV filters. In order to perform beamforming with a full-rank LCMV filter, we linearly combine the data vector $\boldsymbol{r}(i) = [r_1^{(i)} \ r_2^{(i)} \ \dots \ r_M^{(i)}]^T$ with the full-rank filter $\boldsymbol{w} = [w_1 \ w_2 \ \dots \ w_M]^T$ to yield

$$x(i) = \boldsymbol{w}^H\boldsymbol{r}(i) \tag{3}$$

The optimal LCMV filter is the $M \times 1$ vector $\boldsymbol{w}$, which is designed to solve the following optimization problem

$$\begin{aligned} \text{minimize } E[|\boldsymbol{w}^H\mathbf{r}(i)|^2] &= \boldsymbol{w}^H\boldsymbol{R}\boldsymbol{w} \\ \text{subject to } \boldsymbol{w}^H\boldsymbol{a}(\theta_k) &= 1 \end{aligned} \tag{4}$$

The solution to the problem in (4) is given by [3], [4]

$$\boldsymbol{w}_{\text{opt}} = \frac{\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_k)}{\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_k))} \tag{5}$$

where $\boldsymbol{a}(\theta_k)$ is the steering vector of the SoI, $\boldsymbol{r}(i)$ is the received data, the covariance matrix of $\boldsymbol{r}(i)$ is described by $\boldsymbol{R} = E[\boldsymbol{r}(i)\boldsymbol{r}^H(i)]$, $(\cdot)^H$ denotes Hermitian transpose and $E[\cdot]$ stands for expected value. The filter $\boldsymbol{w}(i)$ can be estimated via SG or RLS algorithms [3]. However, the laws that govern their convergence and tracking behaviors imply that they depend on M and on the eigenvalue spread of $\boldsymbol{R}$.

A reduced-rank algorithm must extract the most important features of the processed data by performing dimensionality reduction. This mapping is carried out by a $M \times D$ projection matrix $\boldsymbol{S}_D$ on the received data as given by

$$\bar{\boldsymbol{r}}(i) = \boldsymbol{S}_D^H\boldsymbol{r}(i) \tag{6}$$

where, in what follows, all D-dimensional quantities are denoted with a "bar". The resulting projected received vector $\bar{\boldsymbol{r}}(i)$ is the input to a filter represented by the D vector $\bar{\boldsymbol{w}} = [\bar{w}_1 \ \bar{w}_2 \ \dots \bar{w}_D]^T$. The filter output is

$$\bar{x}(i) = \bar{\boldsymbol{w}}^H\bar{\boldsymbol{r}}(i) \tag{7}$$

In order to design the reduced-rank filter $\bar{\boldsymbol{w}}$ we consider the following optimization problem

$$\text{minimize } E\big[|\bar{\boldsymbol{w}}^H \bar{\boldsymbol{r}}(i)|^2\big] = \bar{\boldsymbol{w}}^H \bar{\boldsymbol{R}} \bar{\boldsymbol{w}}$$

$$\text{subject to } \bar{\boldsymbol{w}}^H \bar{\boldsymbol{a}}(\theta_k) = 1 \tag{8}$$

The solution to the above problem is

$$\bar{\boldsymbol{w}}_{\text{opt}} = \frac{\bar{\boldsymbol{R}}^{-1} \bar{\boldsymbol{a}}(\theta_k)}{\bar{\boldsymbol{a}}^H(\theta_k) \bar{\boldsymbol{R}}^{-1} \bar{\boldsymbol{a}}(\theta_k)} \tag{9}$$

where the reduced-rank covariance matrix is $\bar{\boldsymbol{R}} = E[\bar{\boldsymbol{r}}(i)\bar{\boldsymbol{r}}^H(i)] = \boldsymbol{S}_D^H \boldsymbol{R} \boldsymbol{S}_D$ and the reduced-rank steering vector is $\bar{\boldsymbol{a}}(\theta_k) = \boldsymbol{S}_D^H \boldsymbol{a}(\theta_k)$. The associated minimum variance (MV) for a LCMV filter with rank D is

$$\text{MV} = \frac{1}{\boldsymbol{a}(\theta_k)^H \boldsymbol{S}_D (\boldsymbol{S}_D^H \boldsymbol{R} \boldsymbol{S}_D)^{-1} \boldsymbol{S}_D^H \boldsymbol{a}(\theta_k)} \tag{10}$$

The above development shows that the main problem is how to cost-effectively design $\boldsymbol{S}_D$ to perform dimensionality reduction on $\boldsymbol{r}(i)$, resulting in improved convergence and tracking performance over the full-rank filter. In the Appendix, we provide a necessary and sufficient condition for $\mathbf{S}_D$ to preserve the MV of optimal full-rank filter and discuss the existence of multiple solutions. In the following, we detail our proposed reduced-rank method.

IV. PROPOSED REDUCED-RANK METHOD

In this section, we introduce the principles of the proposed reduced-rank scheme. The proposed scheme, depicted in Fig. 2, employs a matrix $\boldsymbol{S}_D(i)$ with dimensions $M \times D$ to perform dimensionality reduction on a data vector $\boldsymbol{r}(i)$ with dimensions $M \times 1$. The reduced-rank filter $\bar{\boldsymbol{w}}(i)$ with dimensions $D \times 1$ processes the reduced-rank data vector $\bar{\boldsymbol{r}}(i)$ in order to yield a scalar estimate $\bar{x}(i)$. The projection matrix $\mathbf{S}_D(i)$ and the reduced-rank filter $\bar{\boldsymbol{w}}(i)$ are jointly optimized in the proposed scheme according to the MV criterion subject to a constraint that ensures that the reduced-rank array response is equal to unity in the direction of the SoI.

In order to describe the proposed method, let us first consider the structure of the $M \times D$ projection matrix

$$\boldsymbol{S}_D(i) = [\, \boldsymbol{s}_1(i) \mid \boldsymbol{s}_2(i) \mid \, \ldots \, \mid \boldsymbol{s}_D(i) \,] \tag{11}$$

where the columns $\boldsymbol{s}_d(i)$ for $d = 1, \, \ldots, \, D$ constitute a bank of D full-rank filters with dimensions $M \times 1$ as given by

$$\boldsymbol{s}_d(i) = [s_{1,d}(i) \; s_{2,d}(i) \; \ldots \; s_{M,d}(i)]^T$$

The output $\bar{x}(i)$ of the proposed reduced-rank scheme can be expressed as a function of the input vector $\mathbf{r}(i)$, the projection matrix $\mathbf{S}_D(i)$ and the reduced-rank filter $\bar{\mathbf{w}}(i)$:

$$\bar{x}(i) = \bar{\boldsymbol{w}}^H(i) \boldsymbol{S}_D^H(i) \boldsymbol{r}(i) = \bar{\boldsymbol{w}}^H(i) \bar{\boldsymbol{r}}(i) \tag{12}$$

It is interesting to note that for $D = 1$, the proposed scheme becomes a conventional full-rank LCMV filtering scheme with an addition weight parameter w_D that provides an amplitude gain. For $D > 1$, the signal processing tasks are changed and the full-rank LCMV filters compute a subspace projection and the reduced-rank filter provides a unity gain

in the direction of the SoI. This rationale is fundamental to the exploitation of the low-rank nature of signals in typical beamforming scenarios.

The LCMV expressions for the filters $\mathbf{S}_D(i)$ and $\bar{\mathbf{w}}(i)$ can be computed via the proposed optimization problem

$$\text{minimize } E\big[|\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{r}(i)|^2\big] = \bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{R}\boldsymbol{S}_D(i)\bar{\boldsymbol{w}}(i)$$
$$\text{subject to } \bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) = 1 \tag{13}$$

In order to solve the above problem, we resort to the method of Lagrange multipliers [3] and transform the constrained optimization into an unconstrained one expressed by the Lagrangian

$$\mathcal{L}(\boldsymbol{S}_D(i), \bar{\boldsymbol{w}}(i)) = E\big[|\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{r}(i)|^2\big] + 2\Re[\lambda(\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) - 1)], \tag{14}$$

where λ is a scalar Lagrange multiplier, $*$ denotes complex conjugate and the operator $\Re[\cdot]$ selects the real part of the argument. By fixing $\bar{\boldsymbol{w}}(i)$, minimizing (14) with respect to $\boldsymbol{S}_D(i)$ and solving for λ, we get

$$\boldsymbol{S}_D(i) = \frac{\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_k)}, \tag{15}$$

where $\boldsymbol{R} = E[\boldsymbol{r}(i)\boldsymbol{r}^H(i)]$ and $\bar{\boldsymbol{R}}_{\bar{w}} = E[\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)]$. By fixing $\boldsymbol{S}_D(i)$, minimizing (14) with respect to $\bar{\boldsymbol{w}}(i)$ and solving for λ, we arrive at the expression

$$\bar{\boldsymbol{w}}(i) = \frac{\bar{\boldsymbol{R}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_k)}{\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{R}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_k)}, \tag{16}$$

where $\bar{\boldsymbol{R}}(i) = E[\boldsymbol{S}_D^H(i)\boldsymbol{r}(i)\boldsymbol{r}^H(i)\boldsymbol{S}_D(i)] = E[\bar{\boldsymbol{r}}(i)\bar{\boldsymbol{r}}^H(i)]$, $\bar{\boldsymbol{a}}(\theta_k) = \boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k)$. The associated MV is

$$\text{MV} = \frac{1}{\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{R}}^{-1}(i)\bar{\boldsymbol{a}}(\theta_k)}. \tag{17}$$

Note that the filter expressions in (15) and (16) are not closed-form solutions for $\bar{\boldsymbol{w}}(i)$ and $\boldsymbol{S}_D(i)$ since (15) is a function of $\bar{\boldsymbol{w}}(i)$ and (16) depends on $\boldsymbol{S}_D(i)$. Thus, it is necessary to iterate (15) and (16) with initial values to obtain a solution. An analysis of the optimization problem in (13) is given in Appendix II. Unlike existing approaches based on the MSWF [16] and the AVF [18] methods, the proposed scheme provides an iterative exchange of information between the reduced-rank filter and the projection matrix and leads to a much simpler adaptive implementation. The projection matrix reduces the dimension of the input data, whereas the reduced-rank filter yields a unity response in the direction of the SoI. The key strategy lies in the joint optimization of the filters. The rank D must be set by the designer to ensure appropriate performance or can be estimated via another algorithm. In the next section, we seek iterative solutions via adaptive algorithms for the design of $\boldsymbol{S}_D(i)$ and $\bar{\boldsymbol{w}}(i)$, and automatic rank adaptation algorithms.

V. ADAPTIVE ALGORITHMS

In this section we present adaptive SG and RLS versions of the proposed scheme for efficient implementation. We also consider the important issue of automatically determining the rank of the scheme via the proposal of an adaptation technique. We then provide the computational complexity in arithmetic operations of the proposed reduced-rank algorithms.

A. Stochastic Gradient Algorithm

In this part, we present a low-complexity SG adaptive reduced-rank algorithm for efficient implementation of the proposed method. These algorithms were reported in [20], [21] and are reproduced here for convenience. By computing the instantaneous gradient terms of (14) with respect to $\boldsymbol{S}_D^*(i)$ and $\bar{\boldsymbol{w}}^*(i)$, we get

$$
\begin{aligned}
\nabla \mathcal{L}_{MV\,\boldsymbol{S}_D^*(i)} &= \bar{x}^*(i)\boldsymbol{r}(i)\bar{\boldsymbol{w}}^H(i) + 2\lambda^* \mathbf{a}(\theta_k)\bar{\boldsymbol{w}}^H(i) \\
\nabla \mathcal{L}_{MV\,\bar{\boldsymbol{w}}^*(i)} &= \bar{x}^*(i)\boldsymbol{S}_D^H(i)\mathbf{r}(i) + 2\lambda^* \boldsymbol{S}_D^H(i)\mathbf{a}(\theta_k)
\end{aligned}
\tag{18}
$$

By introducing the positive step sizes μ_s and μ_w, using the gradient rules $\boldsymbol{S}_D(i+1) = \boldsymbol{S}_D(i) - \mu_s \nabla \mathcal{L}_{MV\,\boldsymbol{S}_D^*(i)}$ and $\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{w}}(i) - \mu_w \nabla \mathcal{L}_{MV\,\bar{\boldsymbol{w}}^*(i)}$, enforcing the constraint and solving the resulting equations, we obtain

$$
\boldsymbol{S}_D(i+1) = \boldsymbol{S}_D(i) - \mu_s \bar{x}^*(i)\big[\boldsymbol{r}(i)\bar{\boldsymbol{w}}^H(i) - \big(\boldsymbol{a}^H(\theta_k)\boldsymbol{a}(\theta_k)\big)^{-1}\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\boldsymbol{a}^H(\theta_k)\boldsymbol{r}(i)\big],
\tag{19}
$$

$$
\bar{\boldsymbol{w}}(i+1) = \bar{\boldsymbol{w}}(i) - \mu_w \bar{x}^*(i)\big[\boldsymbol{I} - \big(\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{a}}(\theta_k)\big)^{-1}\bar{\boldsymbol{a}}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k)\big]\bar{\boldsymbol{r}}(i),
\tag{20}
$$

where $\bar{x}(i) = \bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{r}(i)$. The proposed scheme trades-off a full-rank filter against one projection matrix $\boldsymbol{S}_D(i)$ and one reduced-rank adaptive filter $\bar{\boldsymbol{w}}(i)$ operating simultaneously and exchanging information.

B. Recursive Least Squares Algorithms

Here we derive an RLS adaptive reduced-rank algorithm for efficient implementation of the proposed method. To this end, let us first consider the Lagrangian

$$
\mathcal{L}_{\mathrm{LS}}(\boldsymbol{S}_D(i), \bar{\boldsymbol{w}}(i)) = \sum_{l=1}^{i} \alpha^{i-l}\big|\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{r}(l)\big|^2 + 2\Re[\lambda\big(\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) - 1\big)]
\tag{21}
$$

where α is the forgetting factor chosen as a positive constant close to, but less than 1.

Fixing $\bar{\boldsymbol{w}}(i)$, computing the gradient of (21) with respect to $\boldsymbol{S}_D(i)$, equating the gradient to a null vector and solving for λ, we obtain

$$
\boldsymbol{S}_D(i) = \frac{\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)}
\tag{22}
$$

where $\boldsymbol{R}(i) = \sum_{l=1}^{i} \alpha^{i-l}\boldsymbol{r}(l)\boldsymbol{r}^H(l)$ is the input covariance matrix, and $\bar{\boldsymbol{R}}_{\bar{w}}(i) = \bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)$ is the reduced-rank weight matrix at time instant i. The computation of (22) includes the inversion of $\boldsymbol{R}(i)$ and $\bar{\boldsymbol{R}}_{\bar{w}}(i)$, which may increase significantly the complexity and create numerical problems. However, the expression in (22) can be further simplified using the constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) = 1$. The details of the derivation of the proposed RLS algorithms and the simplification are given in Appendix III. The simplified expression for $\boldsymbol{S}_D(i)$ is given by

$$
\boldsymbol{S}_D(i) = \frac{\boldsymbol{P}(i)\boldsymbol{a}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k)}{\boldsymbol{a}^H(\theta_k)\boldsymbol{P}(i)\boldsymbol{a}(\theta_k)}
\tag{23}
$$

where we defined the inverse covariance matrix $\boldsymbol{P}(i) = \boldsymbol{R}^{-1}(i)$ for convenience of presentation. Employing the matrix inversion lemma [3], we obtain

$$
\boldsymbol{k}(i) = \frac{\alpha^{-1}\boldsymbol{P}(i-1)\boldsymbol{r}(i)}{1 + \alpha^{-1}\boldsymbol{r}^H(i)\boldsymbol{P}(i-1)\boldsymbol{r}(i)}
\tag{24}
$$

$$\boldsymbol{P}(i) = \alpha^{-1}\boldsymbol{P}(i-1) - \alpha^{-1}\boldsymbol{k}(i)\boldsymbol{r}^H(i)\boldsymbol{P}(i-1) \tag{25}$$

where $\boldsymbol{k}(i)$ is the $M \times 1$ Kalman gain vector. We set $\boldsymbol{P}(0) = \delta\boldsymbol{I}_M$ to start the recursion of (25), where δ is a positive constant and $\boldsymbol{I}_M$ is an $M \times M$ identity matrix.

Assuming $\boldsymbol{S}_D(i)$ is known and taking the gradient of (21) with respect to $\bar{\boldsymbol{w}}(i)$, equating the terms to a null vector and solving for λ, we obtain the $D \times 1$ reduced-rank filter

$$\bar{\boldsymbol{w}}(i) = \frac{\bar{\boldsymbol{P}}(i)\bar{\boldsymbol{a}}(\theta_k)}{\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{P}}(i)\bar{\boldsymbol{a}}(\theta_k)} \tag{26}$$

where $\bar{\boldsymbol{P}}(i) = \bar{\boldsymbol{R}}^{-1}(i)$ and $\bar{\boldsymbol{R}}(i) = \sum_{l=1}^{i}\alpha^{i-l}\bar{\boldsymbol{r}}(l)\bar{\boldsymbol{r}}^H(l)$ is the reduced-rank input covariance matrix. In order to estimate $\bar{\boldsymbol{P}}(i)$, we use the matrix inversion lemma [3] as follows

$$\bar{\boldsymbol{k}}(i) = \frac{\alpha^{-1}\bar{\boldsymbol{P}}(i-1)\bar{\boldsymbol{r}}(i)}{1 + \alpha^{-1}\bar{\boldsymbol{r}}^H(i)\bar{\boldsymbol{P}}(i-1)\bar{\boldsymbol{r}}(i)} \tag{27}$$

$$\bar{\boldsymbol{P}}(i) = \alpha^{-1}\bar{\boldsymbol{P}}(i-1) - \alpha^{-1}\bar{\boldsymbol{k}}(i)\bar{\boldsymbol{r}}^H(i)\bar{\boldsymbol{P}}(i-1) \tag{28}$$

where $\bar{\boldsymbol{k}}(i)$ is the $D \times 1$ reduced-rank gain vector and $\bar{\boldsymbol{P}}(i) = \bar{\boldsymbol{R}}^{-1}(i)$ is referred to as the reduced-rank inverse covariance matrix. Hence, the covariance matrix inversion $\bar{\boldsymbol{R}}^{-1}(i)$ is replaced at each step by the recursive processes (27) and (28) for reducing the complexity. The recursion of (28) is initialized by choosing $\bar{\boldsymbol{P}}(0) = \bar{\delta}\bar{\boldsymbol{I}}_D$, where $\bar{\delta}$ is a positive constant and $\bar{\boldsymbol{I}}_D$ is a $D \times D$ identity matrix.

The proposed RLS algorithm trade-off a full-rank filter with M coefficients against one projection matrix $\boldsymbol{S}_D(i)$, given in (23)-(25) and one $D \times 1$ reduced-rank adaptive filter $\bar{\boldsymbol{w}}(i)$, given in (26)-(28), operating simultaneously and exchanging information.

C. Complexity of Proposed Algorithms

Here, we evaluate the computational complexity of the proposed and analyzed LCMV algorithms. The complexity expressed in terms of additions and multiplications is depicted in Table I. We can verify that the proposed reduced-rank SG algorithm has a complexity that grows linearly with DM, which is about D times higher than the full-rank SG algorithm and significantly lower than the MSWF-SG [16]. If $D << M$ (as we will see later) then the additional complexity can be acceptable provided the gains in performance justify them. In the case of the proposed reduced-rank RLS algorithm the complexity is quadratic with M^2 and D^2. This corresponds to a complexity slightly higher than the one observed for the full-rank RLS algorithm, provided D is significantly smaller than M, and comparable to the cost of the MSWF-RLS [16] and the AVF [18].

In order to illustrate the main trends in what concerns the complexity of the proposed and analyzed algorithms, we show in Fig. 3 the complexity in terms of additions and multiplications versus the number of input samples M. The curves indicate that the proposed reduced-rank RLS algorithm has a complexity lower than the MSWF-RLS algorithm [16] and the AVF [18], whereas it remains at the same level of the full-rank RLS algorithm. The proposed reduced-rank SG algorithm has a complexity that is situated between the full-rank RLS and the full-rank SG algorithms.

D. Automatic Rank Selection

The performance of the algorithms described in the previous subsections depends on the rank D. This motivates the development of methods to automatically adjust D on the basis of the cost function. Unlike prior methods for rank selection which utilize MSWF-based algorithms [16] or AVF-based recursions [18], we focus on an approach that jointly determines D based on the LS criterion computed by the filters $\boldsymbol{S}_D(i)$ and $\bar{w}_D(i)$, where the subscript D denotes the rank used for the adaptation. In particular, we present a method for automatically selecting the ranks of the algorithms based on the exponentially weighted *a posteriori* least-squares type cost function described by

$$\mathcal{C}(\boldsymbol{S}_D(i-1), \bar{\boldsymbol{w}}_D(i-1)) = \sum_{l=1}^{i} \alpha^{i-l} |\bar{\boldsymbol{w}}_D^H(i-1)\boldsymbol{S}_D(i-1)\boldsymbol{r}(l)|^2, \tag{29}$$

where α is the forgetting factor and $\bar{\mathbf{w}}_D(i-1)$ is the reduced-rank filter with rank D. For each time interval i, we can select the rank D_{opt} which minimizes $\mathcal{C}(\boldsymbol{S}_D(i-1), \bar{\boldsymbol{w}}_D(i-1))$ and the exponential weighting factor α is required as the optimal rank varies as a function of the data record. The key quantities to be updated are the projection matrix $\boldsymbol{S}_D(i)$, the reduced-rank filter $\bar{\boldsymbol{w}}_D(i)$, the associated reduced-rank steering vector $\bar{\boldsymbol{a}}(\theta_k)$ and the inverse of the reduced-rank covariance matrix $\bar{\boldsymbol{P}}(i)$ (for the proposed RLS algorithm). To this end, we define the following extended projection matrix $\boldsymbol{S}_D(i)$ and the extended reduced-rank filter weight vector $\bar{\boldsymbol{w}}_D(i)$ as follows:

$$\boldsymbol{S}_D(i) = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,D_{\min}} & \cdots & s_{1,D_{\max}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{M,1} & s_{M,2} & \cdots & s_{M,D_{\min}} & \cdots & s_{M,D_{\max}} \end{bmatrix} \text{ and } \bar{\boldsymbol{w}}_D(i) = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{D_{\min}} \\ \vdots \\ w_{D_{\max}} \end{bmatrix} \tag{30}$$

The extended projection matrix $\boldsymbol{S}_D(i)$ and the extended reduced-rank filter weight vector $\bar{\boldsymbol{w}}_D(i)$ are updated along with the associated quantities $\bar{\boldsymbol{a}}(\theta_k)$ and $\bar{\boldsymbol{P}}(i)$ (only for the RLS) for the maximum allowed rank $D_{\max}$ and then the proposed rank adaptation algorithm determines the rank that is best for each time instant i using the cost function in (29). The proposed rank adaptation algorithm is then given by

$$D_{\text{opt}} = \arg \min_{D_{\min} \leq d \leq D_{\max}} \mathcal{C}(\boldsymbol{S}_D(i-1), \bar{\boldsymbol{w}}_D(i-1)) \tag{31}$$

where d is an integer, $D_{\min}$ and $D_{\max}$ are the minimum and maximum ranks allowed for the reduced-rank filter, respectively. Note that a smaller rank may provide faster adaptation during the initial stages of the estimation procedure and a greater rank usually yields a better steady-state performance. Our studies reveal that the range for which the rank D of the proposed algorithms have a positive impact on the performance of the algorithms is limited, being from $D_{\min} = 3$ to $D_{\max} = 8$ for the reduced-rank filter recursions. These values are rather insensitive to the system load (number of users), to the number of array elements and work very well for all scenarios and algorithms examined. The additional complexity of the proposed rank adaptation algorithm is that it requires the update of all involved quantities

with the maximum allowed rank $D_{\max}$ and the computation of the cost function in (29). This procedure can significantly improve the convergence performance and can be relaxed (the rank can be made fixed) once the algorithm reaches steady state. Choosing an inadequate rank for adaptation may lead to performance degradation, which gradually increases as the adaptation rank deviates from the optimal rank. A mechanism for automatically adjusting $D_{\min}$ and $D_{\max}$ based on a figure of merit and the processed data would be an important technique to be investigated. For example, this mechanism could in principle adjust $D_{\min}$ and $D_{\max}$ in order to address the needs of the model and the performance requirements. This remains a topic for future investigation.

One can also argue that the proposed rank adaptation may not be universally applied to signal processing problems, even though it has been proven highly effective to the problems we dealt with. Another possibility for rank adaptation is the use of the cross-validation (CV) method reported in [18]. This approach selects the lengths of the filters that minimize a cost function that is estimated on the basis of data that have not been used in the process of building the filters themselves. This approach based on the concept of "leave one out" can be used to determine the rank without requiring any prior knowledge or the setting of a range of values [18]. A drawback of this method is that it may significantly increase the length of the filters, resulting in higher complexity. Other possible approaches for rank selection may rely on some prior knowledge about the environment and the system for inferring the required rank for operation. The development of cost-effective methods for rank selection remains an interesting area for investigation.

VI. ANALYSIS OF ALGORITHMS

In this section, we present the stability and the MSE convergence analyses of the proposed SG algorithms. Specifically, we consider the joint optimization approach and derive conditions of stability for the proposed SG algorithms. We then assume that the algorithms will converge and carry out the MSE convergence analysis in order to semi-analytically determine the MSE upon convergence. The RLS algorithms are expected to converge to the optimal LCMV filter and this has been verified in our studies. A discussion on the preservation of the MV performance, the existence of multiple solutions and an analysis of the optimization of the proposed scheme valid for both SG and RLS algorithms is included in the Appendices I and II.

A. Stability Analysis

In order to establish conditions for the stability of the proposed SG algorithms, we define the error matrices at time i as

$$\boldsymbol{e_{S_D}}(i) = \boldsymbol{S}_D(i) - \boldsymbol{S}_{D,\mathrm{opt}}$$

and

$$\boldsymbol{e_{\bar{w}}}(i) = \bar{\boldsymbol{w}}(i) - \bar{\boldsymbol{w}}_{\mathrm{opt}},$$

where $\bar{\boldsymbol{w}}_{\mathrm{opt}}$ and $\boldsymbol{S}_{D,\mathrm{opt}}$ are the optimal parameter estimators. Since we are dealing with a joint optimization procedure, both filters have to be considered jointly. By substituting the expressions of $\boldsymbol{e_{S_D}}(i)$ and $\boldsymbol{e_{\bar{w}}}(i)$ in (19) and (20),

respectively, and rearranging the terms we obtain

$$
\begin{aligned}
\boldsymbol{e_{S_D}}(i+1) = {} & \left\{\boldsymbol{I} - \mu_s[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_k)\boldsymbol{a}(\theta_k))^{-1}\boldsymbol{a}(\theta_k)\boldsymbol{a}^H(\theta_k)]\boldsymbol{r}(i)\boldsymbol{r}^H(i)\right\}\boldsymbol{e_{S_D}}(i) \\
& - \mu_s[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_k)\boldsymbol{a}(\theta_k))^{-1}\boldsymbol{a}(\theta_k)\boldsymbol{a}^H(\theta_k)]\boldsymbol{r}(i)\bar{\boldsymbol{w}}^H(i)\boldsymbol{r}^H(i)\boldsymbol{S}_D(i)\boldsymbol{e_{\bar{w}}}(i) \\
& + \mu_s[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_k)\boldsymbol{a}(\theta_k))^{-1}\boldsymbol{a}(\theta_k)\boldsymbol{a}^H(\theta_k)]\boldsymbol{r}(i)\boldsymbol{r}^H(i)[\boldsymbol{S}_D(i)(\boldsymbol{I} - \bar{\boldsymbol{w}}_{\mathrm{opt}}\bar{\boldsymbol{w}}^H(i)) - \boldsymbol{S}_{D,\mathrm{opt}}]
\end{aligned}
\tag{32}
$$

$$
\begin{aligned}
\boldsymbol{e_w}(i+1) = {} & \left\{\boldsymbol{I} - \mu_w[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{a}}(\theta_k))^{-1}\bar{\boldsymbol{a}}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k)]\bar{\boldsymbol{r}}(i)\bar{\boldsymbol{r}}^H(i)\right\}\boldsymbol{e_w}(i) \\
& - \mu_w[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{a}}(\theta_k))^{-1}\bar{\boldsymbol{a}}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k)]\bar{\boldsymbol{r}}(i)\boldsymbol{r}^H(i)\boldsymbol{e_{S_D}}(i) \\
& + \mu_w[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{a}}(\theta_k))^{-1}\bar{\boldsymbol{a}}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k))\boldsymbol{S}_D^H(i)]\bar{\boldsymbol{r}}(i)\bar{\boldsymbol{r}}^H(i)(\boldsymbol{S}_D(i)(\boldsymbol{I} - \bar{\boldsymbol{w}}_{\mathrm{opt}}) - \boldsymbol{S}_{D,\mathrm{opt}})
\end{aligned}
\tag{33}
$$

Taking expectations and simplifying the terms, we obtain

$$
\begin{bmatrix} E[\boldsymbol{e_{S_D}}(i+1)] \\ E[\boldsymbol{e_{\bar{w}}}(i+1)] \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} E[\boldsymbol{e_{S_D}}(i)] \\ E[\boldsymbol{e_{\bar{w}}}(i)] \end{bmatrix} + \boldsymbol{T}
\tag{34}
$$

where

$$
\boldsymbol{P} = \begin{bmatrix} \left\{\boldsymbol{I} - \mu_s[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_k)\boldsymbol{a}(\theta_k))^{-1}\boldsymbol{a}(\theta_k)\boldsymbol{a}^H(\theta_k)]\boldsymbol{r}(i)\boldsymbol{r}^H(i)\right\} & -\mu_s[\boldsymbol{I} - \boldsymbol{a}(\theta_k)\boldsymbol{a}^H(\theta_k)]\boldsymbol{r}(i)\bar{\boldsymbol{w}}^H(i)\boldsymbol{r}^H(i)\boldsymbol{S}_D(i) \\ -\mu_w[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{a}}(\theta_k))^{-1}\bar{\boldsymbol{a}}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k)]\bar{\boldsymbol{r}}(i)\boldsymbol{r}^H(i) & \left\{\boldsymbol{I} - \mu_w[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{a}}(\theta_k))^{-1}\bar{\boldsymbol{a}}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k)]\bar{\boldsymbol{r}}(i)\bar{\boldsymbol{r}}^H(i)\right\} \end{bmatrix},
$$

$$
\boldsymbol{T} = \begin{bmatrix} \mu_s[\boldsymbol{I} - (\boldsymbol{a}^H(\theta_k)\boldsymbol{a}(\theta_k))^{-1}\boldsymbol{a}(\theta_k)\boldsymbol{a}^H(\theta_k)]\boldsymbol{r}(i)\boldsymbol{r}^H(i)[\boldsymbol{S}_D(i)(\boldsymbol{I} - \bar{\boldsymbol{w}}_{\mathrm{opt}}\bar{\boldsymbol{w}}^H(i)) - \boldsymbol{S}_{D,\mathrm{opt}}] \\ \mu_w[\boldsymbol{I} - (\bar{\boldsymbol{a}}^H(\theta_k)\bar{\boldsymbol{a}}(\theta_k))^{-1}\bar{\boldsymbol{a}}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k))\boldsymbol{S}_D^H(i)]\bar{\boldsymbol{r}}(i)\bar{\boldsymbol{r}}^H(i)(\boldsymbol{S}_D(i)(\boldsymbol{I} - \bar{\boldsymbol{w}}_{\mathrm{opt}}) - \boldsymbol{S}_{D,\mathrm{opt}}) \end{bmatrix}.
$$

The previous equations imply that the stability of the algorithms depends on the spectral radius of $\boldsymbol{P}$. For convergence, the step sizes should be chosen such the eigenvalues of $\boldsymbol{P}^H\boldsymbol{P}$ are less than one. Unlike the stability analysis of most adaptive algorithms [3], in the proposed approach the terms are more involved and depend on each other as evidenced by the equations in $\boldsymbol{P}$ and $\boldsymbol{T}$.

B. MSE Convergence Analysis

Let us consider in this part an analysis of the MSE in steady state. This follows the general steps of the MSE convergence analysis of [3] even though novel elements will be introduced in the proposed framework. These novel elements in the analysis are the joint optimization of the two adaptive filters $\bar{\boldsymbol{w}}(i)$ and $\boldsymbol{S}_D(i)$ of the proposed scheme and a strategy to incorporate the effect of the step size of the recursions in (19) and (20).

Let us define the MSE at time $i + 1$ using the relations

$$
\boldsymbol{e_w}(i+1) = \boldsymbol{w}(i+1) - \boldsymbol{w}_{opt}
$$

and

$$
\xi(i) = E[\boldsymbol{w}^H(i)\boldsymbol{r}(i)\boldsymbol{r}(i)\boldsymbol{w}(i)],
$$

where the filter $\boldsymbol{w}(i) = \boldsymbol{S}_D(i)\bar{\boldsymbol{w}}(i)$ with M coefficients is the D-rank approximation of a full-rank filter obtained with an inverse mapping performed by $\boldsymbol{S}_D(i)$.

The MSE of the proposed scheme can be expressed by:

$$\text{MSE}(i) = E[|d(i) - \boldsymbol{w}^H(i)\boldsymbol{r}(i)|^2]$$
$$= \epsilon_{min} + \xi(i) - \xi_{min} - E[\boldsymbol{e}_{\boldsymbol{w}}^H(i)]\boldsymbol{a}(\theta_k) - \boldsymbol{a}^H(\theta_k)E[\boldsymbol{e}_{\boldsymbol{w}}(i)] \quad (35)$$
$$= \epsilon_{min} + \xi_{ex}(i) - E[\boldsymbol{e}_{\boldsymbol{w}}^H(i)]\boldsymbol{a}(\theta_k) - \boldsymbol{a}^H(\theta_k)E[\boldsymbol{e}_{\boldsymbol{w}}(i)]$$

where $d(i)$ corresponds to the desired signal, $\xi(i) = E[\boldsymbol{w}^H(i)\boldsymbol{r}(i)\boldsymbol{r}^H(i)\boldsymbol{w}(i)]$, $\epsilon_{min} = E[|d(i) - \boldsymbol{w}_{opt}^H\boldsymbol{r}(i)|^2]$ is the MSE with

$$\boldsymbol{w}_{opt} = \xi_{min}\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_k), \quad (36)$$

where $\xi_{min} = 1/(\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_k))$ is the minimum variance, and $\xi_{ex}(i) = \xi(i) - \xi_{min}$ is the excess MSE due to the adaptation process at the time instant i. Since $\lim_{i\to\infty} E[\boldsymbol{e}_{\boldsymbol{w}}(i)] = 0$ we have

$$\lim_{i\to\infty} \text{MSE}(i) = \epsilon_{min} + \lim_{i\to\infty} \xi_{ex}(i) \quad (37)$$

where the $\xi_{ex}(\infty)$ term in (37) is the steady-state excess MSE resulting from the adaptation process. The main difference here from prior work lies in the fact that this refers to the excess MSE produced by a D-rank approximation filter $\boldsymbol{w}(i)$. In order to analyze the trajectory of $\xi(i)$, let us rewrite it as

$$\xi(i) = E[\boldsymbol{w}^H(i)\boldsymbol{r}(i)\boldsymbol{r}^H(i)\boldsymbol{w}(i)]$$
$$= E[\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{r}(i)\boldsymbol{r}^H(i)\boldsymbol{S}_D(i)\bar{\boldsymbol{w}}(i)] \quad (38)$$
$$= tr\, E[\boldsymbol{R}_{\boldsymbol{w}}(i)\boldsymbol{R}]$$

where $\boldsymbol{R}_{\boldsymbol{w}}(i) = E[\boldsymbol{w}(i)\boldsymbol{w}^H(i)] = \boldsymbol{w}_{opt}\boldsymbol{w}_{opt}^H + E[\boldsymbol{e}_w(i)]\boldsymbol{w}_{opt}^H + \boldsymbol{w}_{opt}E[\boldsymbol{e}_{\boldsymbol{w}}^H(i)] + \boldsymbol{R}_{\boldsymbol{e}_w}(i)$ [5].

To proceed with the analysis, we must define the quantities $\boldsymbol{R} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^H$, where the columns of $\boldsymbol{\Phi}$ are the eigenvectors of the symmetric and positive semi-definite matrix $\boldsymbol{R}$ and $\boldsymbol{\Lambda}$ is the diagonal matrix of the corresponding eigenvalues, $\boldsymbol{R}_{\boldsymbol{e}_w}(i) = E[\boldsymbol{e}_{\boldsymbol{w}}(i)\boldsymbol{e}_{\boldsymbol{w}}^H(i)]$, the rotated tap error vector $\tilde{\boldsymbol{e}}_{\boldsymbol{w}}(i) = \boldsymbol{\Phi}^H\boldsymbol{e}_{\boldsymbol{w}}(i)$, the rotated signal vectors $\tilde{\boldsymbol{r}}(i) = \boldsymbol{\Phi}^H\boldsymbol{r}(i)$, $\tilde{\boldsymbol{a}}(\theta_k) = \boldsymbol{\Phi}^H\boldsymbol{a}(\theta_k)$ and $\boldsymbol{R}_{\tilde{\boldsymbol{e}}_w}(i) = E[\tilde{\boldsymbol{e}}_{\boldsymbol{w}}(i)\tilde{\boldsymbol{e}}_{\boldsymbol{w}}^H(i)] = \boldsymbol{\Phi}^H\boldsymbol{R}_{\boldsymbol{e}_w}(i)\boldsymbol{\Phi}$. Rewriting (38) in terms of the above transformed quantities we have:

$$\xi(i) = tr\, E[\boldsymbol{\Lambda}\boldsymbol{\Phi}^H\boldsymbol{R}_{\boldsymbol{w}}\boldsymbol{\Phi}]$$
$$= \xi_{min} + tr[E[\tilde{\boldsymbol{e}}_{\boldsymbol{w}}(i)]\tilde{\boldsymbol{a}}^H(\theta_k) + \tilde{\boldsymbol{a}}(\theta_k)E[\tilde{\boldsymbol{e}}_{\boldsymbol{w}}^H(i)] \quad (39)$$
$$+ \boldsymbol{\Lambda}\boldsymbol{R}_{\tilde{\boldsymbol{e}}_w}(i)]$$

Since $\lim_{i\to\infty} E[\tilde{\boldsymbol{e}}_{\boldsymbol{w}}(i)] = 0$, then $\lim_{i\to\infty} \xi(i) = \xi_{min} + tr[\Lambda\boldsymbol{R}_{\tilde{\boldsymbol{e}}_w}]$. Thus, it is evident that to assess the evolution of $\xi(i)$ it is sufficient to study $\boldsymbol{R}_{\tilde{\boldsymbol{e}}_w}(i)$.

Using $\boldsymbol{e}_{\boldsymbol{S}_D}(i)$ and $\boldsymbol{e}_{\bar{\boldsymbol{w}}}(i)$ and combining them to compute $\boldsymbol{e}_{\boldsymbol{w}}(i)$, we get

$$\boldsymbol{e}_{\boldsymbol{w}}(i) = \boldsymbol{w}(i) - \boldsymbol{w}_{\text{opt}}$$
$$= \boldsymbol{S}_D(i)\bar{\boldsymbol{w}}(i) - \boldsymbol{S}_{D,\,\text{opt}}\bar{\boldsymbol{w}}_{\text{opt}} \quad (40)$$
$$= \boldsymbol{e}_{\boldsymbol{S}_D}(i)\boldsymbol{e}_{\bar{\boldsymbol{w}}}(i) + \boldsymbol{S}_{D,\text{opt}}\boldsymbol{e}_{\bar{\boldsymbol{w}}}(i) + \boldsymbol{e}_{\boldsymbol{S}_D}(i)\bar{\boldsymbol{w}}_{\text{opt}}$$

Substituting the expressions for $e_{\boldsymbol{S}_D}(i+1)$ and $e_{\bar{\boldsymbol{w}}}(i+1)$ in (32) and (33), respectively, to compute $e_{\boldsymbol{w}}(i+1)$, we get

$$
\begin{aligned}
e_{\boldsymbol{w}}(i+1) = e_{\boldsymbol{w}}(i) &- \mu_w \bar{x}^*(i) \boldsymbol{S}_D(i) \bar{\boldsymbol{r}}_p(i) - \mu_s \bar{x}^*(i) \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{w}}(i) \\
&+ \mu_s \mu_w (\bar{x}^*(i))^2 \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{r}}_p(i) + \boldsymbol{S}_{D,\text{opt}} e_{\bar{\boldsymbol{w}}}(i) + e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{opt}
\end{aligned}
\tag{41}
$$

where

$$
\bar{x}(i) = \bar{\boldsymbol{w}}^H(i) \boldsymbol{S}_D^H(i) \boldsymbol{r}(i) = \boldsymbol{w}^H(i) \boldsymbol{r}(i)
$$

$$
\boldsymbol{S}_{\boldsymbol{r}_p}(i) = \big(\boldsymbol{I} - (\boldsymbol{a}^H(\theta_k) \boldsymbol{a}(\theta_k))^{-1} \boldsymbol{a}(\theta_k) \boldsymbol{a}^H(\theta_k) \big) \boldsymbol{r}(i) \bar{\boldsymbol{w}}^H(i)
$$

$$
\bar{\boldsymbol{r}}_p(i) = (\boldsymbol{I} - (\boldsymbol{S}_D(i) \boldsymbol{a}^H(\theta_k) \boldsymbol{S}_D^H(i) \boldsymbol{a}(\theta_k))^{-1} \boldsymbol{S}_D(i) \boldsymbol{a}(\theta_k) \boldsymbol{a}^H(\theta_k) \boldsymbol{S}_D^H(i) \boldsymbol{S}_D(i) \boldsymbol{r}(i)
$$

We can further rewrite the expressions above in order to obtain a more compact and convenient representation as

$$
e_{\boldsymbol{w}}(i+1) = (\boldsymbol{I} - \boldsymbol{A}) e_{\boldsymbol{w}}(i) + \boldsymbol{B}\boldsymbol{C} + \mu_s \mu_w (\bar{x}^*(i))^2 \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{r}}_p(i) + e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{opt}
\tag{42}
$$

where

$$
\boldsymbol{A} = \mu_w \boldsymbol{S}_D(i) \bar{\boldsymbol{r}}_p(i) \boldsymbol{r}^H(i) + \mu_s \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{w}}(i) \boldsymbol{r}^H(i) - \boldsymbol{S}_{D,\text{opt}}
$$

$$
\boldsymbol{B} = -\mu_w \boldsymbol{S}_D(i) \bar{\boldsymbol{r}}_p(i) \boldsymbol{r}^H(i) - \mu_s \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{w}}(i) \boldsymbol{r}^H(i)
$$

$$
\boldsymbol{C} = e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{\text{opt}} + \boldsymbol{S}_{D,\text{opt}} e_{\bar{\boldsymbol{w}}}(i) + e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{\text{opt}}.
$$

Now, we need to compute $\boldsymbol{R}_{e_{\boldsymbol{w}}}(i+1) = E[e_{\boldsymbol{w}}(i+1) e_{\boldsymbol{w}}^H(i+1)]$ by using the result in (42), which yields

$$
\begin{aligned}
\boldsymbol{R}_{e_{\boldsymbol{w}}}(i+1) = (\boldsymbol{I} - \boldsymbol{A}) \boldsymbol{R}_{e_{\boldsymbol{w}}}(i)(\boldsymbol{I} - \boldsymbol{A})^H &+ (\boldsymbol{I} - \boldsymbol{A}) e_{\boldsymbol{w}}(i) \boldsymbol{C}^H \boldsymbol{B}^H \\
&+ \mu_s \mu_w (\bar{x}(i))^2 (\boldsymbol{I} - \boldsymbol{A}) e_{\boldsymbol{w}}(i)(\bar{\boldsymbol{r}}_p^H(i) \boldsymbol{S}_{\boldsymbol{r}_p}^H(i)) \\
&+ (\boldsymbol{I} - \boldsymbol{A}) e_{\boldsymbol{w}}(i) \bar{\boldsymbol{w}}_{\text{opt}}^H \boldsymbol{S}_{D,\text{opt}}^H + \boldsymbol{B}\boldsymbol{C} e_{\boldsymbol{w}}^H(i)((\boldsymbol{I} - \boldsymbol{A})^H \\
&+ \boldsymbol{B}\boldsymbol{C}\boldsymbol{C}^H \boldsymbol{B}^H + \mu_s \mu_w (\bar{x}(i))^2 \boldsymbol{B}\boldsymbol{C} \bar{\boldsymbol{r}}_p^H(i) \boldsymbol{S}_{\boldsymbol{r}_p}^H(i) \\
&+ \boldsymbol{B}\boldsymbol{C} \boldsymbol{w}_{\text{opt}}^H e_{\boldsymbol{S}_D}^H(i) + \mu_s \mu_w (\bar{x}^*(i))^2 \boldsymbol{S}_{\boldsymbol{r}_p}(i) \boldsymbol{r}_p(i) e_{\boldsymbol{w}}^H(i)(\boldsymbol{I} - \boldsymbol{A})^H \\
&+ \mu_s \mu_w (\bar{x}^*(i))^2 \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{r}}_p(i) \boldsymbol{C}^H \boldsymbol{A}^H \\
&+ (\mu_s \mu_w)^2 |\bar{x}(i)|^4 \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{r}}_p(i) \bar{\boldsymbol{r}}_p^H(i) \boldsymbol{S}_{\boldsymbol{r}_p}(i) \\
&+ \mu_s \mu_w (\bar{x}(i))^2 e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{\text{opt}} \bar{\boldsymbol{r}}_p(i) \boldsymbol{S}_{\boldsymbol{r}_p}(i) \\
&- e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{opt} e_{\boldsymbol{w}}^H(i)(\boldsymbol{I} - \boldsymbol{A})^H + e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{opt} \boldsymbol{C}^H \boldsymbol{B}^H \\
&+ e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{\text{opt}} \bar{\boldsymbol{w}}_{\text{opt}}^H e_{\boldsymbol{S}_D}^H(i)
\end{aligned}
\tag{43}
$$

Since $E[e_{\boldsymbol{w}}(i)] = \boldsymbol{0}$ and $E[e_{\boldsymbol{S}_D}(i)] = \boldsymbol{0}$, we can simplify the previous expression and obtain

$$
\begin{aligned}
\boldsymbol{R}_{e_{\boldsymbol{w}}}(i+1) = (\boldsymbol{I} - \boldsymbol{A}) \boldsymbol{R}_{e_{\boldsymbol{w}}}(i)(\boldsymbol{I} - \boldsymbol{A})^H &\\
&+ \boldsymbol{B}\boldsymbol{C}\boldsymbol{C}^H \boldsymbol{B}^H + \mu_s \mu_w (\bar{x}(i))^2 \boldsymbol{B}\boldsymbol{C} \bar{\boldsymbol{r}}_p^H(i) \boldsymbol{S}_{\boldsymbol{r}_p}^H(i) \\
&+ \mu_s \mu_w (\bar{x}^*(i))^2 \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{r}}_p(i) \boldsymbol{C}^H \boldsymbol{A}^H \\
&+ (\mu_s \mu_w)^2 |\bar{x}(i)|^4 \boldsymbol{S}_{\boldsymbol{r}_p}(i) \bar{\boldsymbol{r}}_p(i) \bar{\boldsymbol{r}}_p^H(i) \boldsymbol{S}_{\boldsymbol{r}_p}(i) \\
&+ e_{\boldsymbol{S}_D}(i) \bar{\boldsymbol{w}}_{\text{opt}} \bar{\boldsymbol{w}}_{\text{opt}}^H e_{\boldsymbol{S}_D}^H(i)
\end{aligned}
\tag{44}
$$

Solving for $\boldsymbol{R}_{e_w}$, the MSE can be computed by

$$\text{MSE}(i+1) = \epsilon_{min} + tr[\Lambda \boldsymbol{R}_{\tilde{e}_w}(i)]$$
$$= \epsilon_{min} + tr[\Lambda \boldsymbol{\Phi} \boldsymbol{R}_{e_w}(i) \boldsymbol{\Phi}^H] \tag{45}$$

It should be remarked that the expression for $\boldsymbol{R}_{e_w}(i)$ is quite involved and requires a semi-analytical approach with the aid of computer simulations for its computation. This is because the terms resulting from the joint adaptation create numerous extra terms in the expression of $\boldsymbol{R}_{e_w}(i)$, which are very difficult to isolate. We found that using computer simulations to pre-compute the terms of $\boldsymbol{R}_{e_w}(i)$ as a function of the step sizes was more practical and resulted in good match between the semi-analytical and simulated curves. In the following section, we will demonstrate that it is able to predict the performance of the proposed SG algorithm.

VII. SIMULATIONS

In this section we evaluate the performance of the proposed and the analyzed beamforming algorithms via computer simulations. We also verify the validity of the MSE convergence analysis of the previous section. A smart antenna system with a ULA containing M sensor elements is considered for assessing the beamforming algorithms. In particular, the performance of the proposed scheme and SG and RLS algorithms is compared with existing techniques, namely, the full-rank LCMV-SG [4] and LCMV-RLS [8], and the reduced-rank algorithms with $\boldsymbol{S}_D(i)$ designed according to the MSWF [16], the AVF [18] and the optimal linear beamformer that assumes the knowledge of the covariance matrix [2]. In particular, the algorithms are compared in terms of the mean-squared error (MSE) and the signal-to-interference-plus-noise ratio (SINR), which is defined for the reduced-rank schemes as

$$\text{SINR}(i) = \frac{\bar{\boldsymbol{w}}^H(i) \boldsymbol{S}_D^H(i) \boldsymbol{R}_s \boldsymbol{S}_D(i) \bar{\boldsymbol{w}}(i)}{\bar{\boldsymbol{w}}^H(i) \boldsymbol{S}_D^H(i) \boldsymbol{R}_I \boldsymbol{S}_D(i) \bar{\boldsymbol{w}}(i)}, \tag{46}$$

where $\boldsymbol{R}_s$ is the autocorrelation matrix of the desired signal and $\boldsymbol{R}_I$ is the cross-correlation matrix of the interference and noise in the environment. Note that for the full-rank schemes the $\text{SINR}(i)$ assumes $\boldsymbol{S}_D^H(i) = \boldsymbol{I}_M$, where $\boldsymbol{I}_M$ is an identity matrix with dimensionality M. For each scenario, 200 runs are used to obtain the curves. In all simulations, the desired signal power is $\sigma_d^2 = 1$, and the signal-to-noise ratio (SNR) is defined as $\text{SNR} = \frac{\sigma_d^2}{\sigma^2}$. The filters are initialized as $\bar{\boldsymbol{w}}(0) = [1\ 0\ \dots\ 0]$ and $\boldsymbol{S}_D(0) = [\boldsymbol{I}_D^T\ \boldsymbol{0}_{D \times (M-D)}^T]$, where $\boldsymbol{0}_{D \times M-D}$ is a $D \times (M-D)$ matrix with zeros in all experiments.

A. MSE Analytical Performance

In this part of the section, we verify that the results in (43) and (45) of the section on MSE convergence analysis of the proposed reduced-rank SG algorithms can provide a means of estimating the MSE upon convergence. The steady state MSE between the desired and the estimated symbol obtained through simulation is compared with the steady state MSE computed via the expressions derived in Section VI. In order to illustrate the usefulness of our analysis we have carried out some experiments. To semi-analytically compute the MSE for the SG recursion, we have used (36) and assumed the knowledge of the data covariance matrix $\mathbf{R}$. We consider 5 interferers ($K = 6$ users in total - the SoI and

the interferers) at -60°, -30°, 0°, 45°, 60° with powers following a log-normal distribution with associated standard deviation 3 dB around the SoI's power level, which impinges on the array at 15°.

We compare the results obtained via simulations with those obtained by the semi-analytical approach presented in Section VI. In particular, we consider two sets of parameters in order to check the validity of our approach. One of the sets has larger step sizes ($\mu_s = 0.0025$ and $\mu_w = 0.01$), whereas the other set employs smaller step sizes ($\mu_s = 0.001$ and $\mu_w = 0.001$) for the recursions. The results shown in Fig. 4 indicate that the curves obtained with the semi-analytical approach agrees with those obtained via simulations for both sets of parameters, verifying the validity of our analysis. Note that the algorithms with smaller step sizes converge slower than the algorithms equipped with larger step sizes. However, the proposed algorithms with smaller step sizes converge to the same level of MSE as the optimal LCMV, whereas the proposed algorithms with larger step sizes exhibit a higher level of misadjustment. In what follows, we will consider the convergence rate of the proposed reduced-rank algorithms in comparison with existing algorithms.

B. SINR Performance

In the first two experiments, we consider 7 interferers at -60°, -45°, -30°, -15^0, 0°, 45°, 60° with powers following a log-normal distribution with associated standard deviation 3 dB around the SoI's power level. The SoI impinges on the array at 30°. The parameters of the algorithms are optimized.

We first evaluate the SINR performance of the analyzed algorithms against the rank D using optimized parameters (μ_s, μ_w and forgetting factors λ) for all schemes and $N = 250$ snapshots. The results in Fig. 5 indicate that the best rank for the proposed scheme is $D = 4$ (which will be used in the second scenario) and it is very close to the optimal full-rank LCMV filter. Our studies with systems with different sizes show that D is relatively invariant to the system size, which brings considerable computational savings. In practice, the rank D can be adapted in order to obtain fast convergence and ensure good steady-state performance and tracking after convergence.

We show another scenario in Fig. 6 where the adaptive LCMV filters are set to converge to the same level of SINR. The parameters used to obtain these curves are also shown. The SG version of the MSWF is known to have problems in these situations since it does not tridiagonalize its covariance matrix [16], being unable to approach the optimal LCMV. The curves show an excellent performance for the proposed scheme which converges much faster than the full-rank-SG algorithm, and is also better than the more complex MSWF-RLS and AVF schemes.

In the next experiment, we consider the design of the proposed adaptive reduced-rank LCMV algorithms equipped with the automatic rank selection method described in Section V.D. We consider 5 interferers at -60°, -30°, 0°, 45°, 60° with equal powers to the SoI, which impinges on the array at 15°. Specifically, we evaluate the proposed rank selection algorithms against the use of fixed ranks, namely, $D = 3$ and $D = 8$ for both SG and RLS algorithms. The results show that the proposed automatic rank selection method is capable of ensuring an excellent trade-off between convergence speed and steady-state performance, as illustrated in Fig 7. In particular, the proposed algorithm can achieve a significantly faster convergence performance than the scheme with fixed rank $D = 8$, whereas it attains the same steady state performance.

In the last experiment, we consider a non-stationary scenario where the system has 6 users with equal power and the environment experiences a sudden change at time $i = 800$. The 5 interferers impinge on the ULA at -60°, -30°, 0°, 45°, 60° with equal powers to the SoI, which impinges on the array at 15°. At time instant $i = 800$ we have 3 interferers with 5 dB above the SoI's power level entering the system with DoAs -45°, -15° and 30°, whereas one interferer with DoA 45° and a power level equal to the SoI exits the system. The proposed and analyzed adaptive beamforming algorithms are equipped with automatic rank adaptation techniques and have to adjust their parameters in order to suppress the interferers. We optimize the step sizes and the forgetting factors of all the algorithms in order to ensure that they converge as fast as they can to the same value of SINR. The results of this experiment are depicted in Fig. 8. The curves show that the proposed reduced-rank algorithms have a superior performance to the existing algorithms.

VIII. Conclusions

We proposed reduced-rank LCMV beamforming algorithms based on joint iterative optimization of filters. The proposed reduced-rank scheme is based on a constrained joint iterative optimization of filters according to the minimum variance criterion. We derived LCMV expressions for the design of the projection matrix and the reduced-rank filter and developed SG and RLS adaptive algorithms for their efficient implementation along with an automatic rank selection technique. An analysis of the stability and the convergence properties of the proposed algorithms was presented and semi-analytical expressions were derived for predicting the MSE performance. The numerical results for a digital beamforming application with a ULA showed that the proposed scheme and algorithms outperform in convergence and tracking the existing full-rank and reduced-rank algorithms at comparable complexity. The proposed algorithms can be extended to other array geometries and applications .

Appendix

I. Preservation of MV and Existence of Multiple Solutions

In this Appendix we discuss the conditions for which the MV obtained for the full-rank filter is preserved and the existence of multiple solutions in the proposed optimization method. Given an $M \times D$ projection matrix $\boldsymbol{S}_D(i)$, where $D \leq M$, the MV is achieved if and only if $\boldsymbol{w}$ which minimizes (4) belongs to the $\mathrm{Range}\{\boldsymbol{S}_D(i)\}$, i.e. $\boldsymbol{w}(i)$ lies in the subspace generated by $\boldsymbol{S}_D(i)$. In this case, we have

$$\mathrm{MV}(\bar{\boldsymbol{w}}(i)) = (\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_k))^{-1}. \tag{47}$$

For a general $\boldsymbol{S}_D(i)$, we have

$$\mathrm{MV}(\bar{\boldsymbol{w}}(i)) \geq (\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_k))^{-1}. \tag{48}$$

From the above relations, we can conclude that there exists multiple solutions to the proposed optimization problem.

II. Analysis of The Optimization of the Proposed Scheme

In this appendix, we carry out an analysis of the proposed reduced-rank method and its optimization. Our approach is based on expressing the output of the proposed scheme and the proposed constraint in a convenient form that renders itself to analysis. Let us rewrite the proposed constrained optimization method in (13) using the method of Lagrange multipliers and express it by the Lagrangian

$$\mathcal{L} = E\big[|\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{r}(i)|^2\big] + 2\Re[\lambda(\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) - 1)], \tag{49}$$

In order to proceed, let us express $\bar{x}(i)$ in an alternative and more convenient form as

$$\begin{aligned}
\bar{x}(i) &= \bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{r}(i) = \bar{\boldsymbol{w}}^H(i)\sum_{d=1}^{D}\boldsymbol{s}_d^H(i)\boldsymbol{r}(i)\boldsymbol{q}_d \\
&= \bar{\boldsymbol{w}}^H(i)
\begin{bmatrix}
\boldsymbol{r}(i) & 0 & 0 & \dots & 0 \\
0 & \boldsymbol{r}(i) & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \dots & 0 & 0 & \boldsymbol{r}(i)
\end{bmatrix}^T
\begin{bmatrix}
\boldsymbol{s}_1^*(i) \\
\boldsymbol{s}_2^*(i) \\
\vdots \\
\boldsymbol{s}_D^*(i)
\end{bmatrix} \\
&= \bar{\boldsymbol{w}}^H(i)\boldsymbol{\Re}^T(i)\boldsymbol{s}_v^*(i)
\end{aligned} \tag{50}$$

where $\boldsymbol{\Re}(i)$ is a $DM \times D$ block diagonal matrix with the input data vector $\boldsymbol{r}(i)$, $\boldsymbol{q}_d$ is a $D \times 1$ vector with a 1 in the d-th position and $\boldsymbol{s}_v^*(i)$ is a $DM \times 1$ vector with the columns of $\boldsymbol{S}_D(i)$ stacked on top of each other.

In order to analyze the proposed joint optimization procedure, we can rearrange the terms in $\bar{x}(i)$ and define a single $D(M+1) \times 1$ parameter vector $\boldsymbol{f}(i) = [\bar{\boldsymbol{w}}^T(i)\ \boldsymbol{s}_v^T(i)]^T$. We can therefore further express $\bar{x}(i)$ as

$$\begin{aligned}
\bar{x}(i) &= \boldsymbol{f}^H(i)
\begin{bmatrix}
\boldsymbol{0}_{D\times D} & \boldsymbol{0}_{D\times DM} \\
\boldsymbol{\Re}(i) & \boldsymbol{0}_{DM\times DM}
\end{bmatrix}
\boldsymbol{f}(i) \\
&= \boldsymbol{f}^H(i)\boldsymbol{G}(i)\boldsymbol{f}(i)
\end{aligned} \tag{51}$$

where $\boldsymbol{G}(i)$ is a $D(M+1) \times D(M+1)$ matrix which contains $\boldsymbol{\Re}(i)$. Now let us perform a similar linear algebra transformation with the proposed constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) = 1$ and express it as

$$\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) = \boldsymbol{f}^H(i)\boldsymbol{A}(\theta_k)\boldsymbol{f}(i) \tag{52}$$

where the $D(M+1) \times D(M+1)$ matrix $\boldsymbol{A}(\theta_k)$ is structured as

$$\boldsymbol{A}(\theta_k) =
\begin{bmatrix}
\boldsymbol{0}_{D\times D} & \boldsymbol{0}_{D\times DM} \\
\boldsymbol{\Re}_{\boldsymbol{a}(\theta_k)} & \boldsymbol{0}_{DM\times DM}
\end{bmatrix}$$

and the $DM \times D$ block diagonal matrix $\boldsymbol{\Re}_{\boldsymbol{a}(\theta_k)}(i)$ with the steering vector $\boldsymbol{a}(\theta_k)$ constructed as

$$\boldsymbol{\Re}_{\boldsymbol{a}(\theta_k)} = \begin{bmatrix} \boldsymbol{a}(\theta_k) & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{a}(\theta_k) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \boldsymbol{a}(\theta_k) \end{bmatrix} \tag{53}$$

At this point, we can alternatively express the Lagrangian in (57) as

$$\mathcal{L} = E\left[|\boldsymbol{f}^H(i)\boldsymbol{G}(i)\boldsymbol{f}(i)|^2\right] + 2\Re[\lambda(\boldsymbol{f}^H(i)\boldsymbol{A}(\theta_k)\boldsymbol{f}(i) - 1)]. \tag{54}$$

We can examine the convexity of the above Lagrangian by computing the Hessian ($\boldsymbol{H}$)with respect to $\boldsymbol{f}(i)$ using the expression [22]

$$\boldsymbol{H} = \frac{\partial}{\partial \boldsymbol{f}^H(i)} \frac{\partial(\mathcal{L})}{\partial \boldsymbol{f}(i)} \tag{55}$$

and testing if the terms are positive semi-definite. Specifically, $\boldsymbol{H}$ is positive semi-definite if $\boldsymbol{v}^H\boldsymbol{H}\boldsymbol{v} \geq 0$ for all nonzero $\boldsymbol{v} \in \boldsymbol{C}^{D(M+1) \times D(M+1)}$ [23]. Therefore, the optimization problem is convex if the Hessian $\boldsymbol{H}$ is positive semi-definite.

Evaluating the partial differentiation in the expression given in (55) yields

$$\begin{aligned} \boldsymbol{H} = E\big[&\boldsymbol{f}^H(i)\boldsymbol{G}(i)\boldsymbol{f}(i)\boldsymbol{G}(i) + \boldsymbol{G}(i)\boldsymbol{f}(i)\boldsymbol{f}^H(i)\boldsymbol{G}(i) \\ &+ \boldsymbol{G}(i)\boldsymbol{f}^H(i)\boldsymbol{G}(i)\boldsymbol{f}(i) + \boldsymbol{f}^H(i)\boldsymbol{G}(i)\boldsymbol{G}(i)\boldsymbol{f}(i) + 2\lambda\boldsymbol{A}(\theta_k)\big] \end{aligned} \tag{56}$$

By examining $\boldsymbol{H}$, we verify that the second and fourth terms are positive semi-definite, whereas the first and the third terms are indefinite. The fifth term depends on the constraint, which is typically positive in the proposed scheme as verified in our studies, yielding a positive semi-definite matrix. Therefore, the optimization problem can not be classified as convex. It is however important to remark that our studies indicate that there are no local minima and there exists multiple solutions (which are possibly identical).

In order to support this claim, we have checked the impact on the proposed algorithms of different initializations . This study confirmed that the algorithms are not subject to performance degradation due to the initialization although we have to bear in mind that the initialization $\boldsymbol{S}_D(0) = \boldsymbol{0}_{M \times D}$ annihilates the signal and must be avoided. We have also studied a particular case of the proposed scheme when $M = 1$ and $D = 1$, which yields the Lagrangian $\mathcal{L}(\bar{w}, \boldsymbol{S}_D) = E\left[|\bar{w}S_D r|^2\right] + 2\Re\left[\lambda(\bar{w}S_D a(\theta_k) - 1)\right]$. Choosing S_D (the "scalar" projection) fixed with D equal to 1, it is evident that the resulting function $\mathcal{L}(\bar{w}, S_D = 1, r) = |w^* r|^2 + 2\Re\left[\lambda(\bar{w}a(\theta_k) - 1)\right]$ is a convex one. In contrast to that, for a time-varying projection S_D the plots of the function indicate that the function is no longer convex but it also does not exhibit local minima. This problem can be generalized to the vector case, however, we can no longer verify the existence of local minima due to the multi-dimensional surface. This remains as an interesting open problem.

III. DERIVATION AND SIMPLIFICATION OF $\boldsymbol{S}_D(i)$

In this appendix, we detail the derivation of the filter $\boldsymbol{S}_D(i)$ and the simplification shown in (23) for reducing the computational complexity. Let us consider the derivation of $\boldsymbol{S}_D(i)$ obtained from the minimization of the Lagrangian

$$\mathcal{L}(\boldsymbol{S}_D(i), \bar{\boldsymbol{w}}(i)) = \sum_{l=1}^{i} \alpha^{i-l} |\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{r}(l)|^2 + 2\Re[\lambda(\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) - 1)], \tag{57}$$

Taking the gradient terms of the above expression with respect to $\boldsymbol{S}_D^*(i)$, we get

$$\begin{aligned} \nabla\mathcal{L}(\boldsymbol{S}_D(i), \bar{\boldsymbol{w}}(i))_{\boldsymbol{S}_D^*(i)} &= \sum_{l=1}^{i} \alpha^{i-l}\boldsymbol{r}(l)\boldsymbol{r}^H(l)\boldsymbol{S}_D(i)\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i) + 2\lambda\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i) \\ &= \boldsymbol{R}(i)\boldsymbol{S}_D(i)\bar{\boldsymbol{R}}_{\bar{w}}(i) + 2\lambda\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i). \end{aligned} \tag{58}$$

Making the above gradient terms equal to zero yields

$$\boldsymbol{S}_D(i) = \boldsymbol{R}^{-1}(i)(-2\lambda)\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}. \tag{59}$$

Using the proposed constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) = 1$ and substituting the above filter expression, we obtain the Lagrange multiplier $\lambda = -1/2(\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k))^{-1}$. Substituting λ into (59), we get

$$\boldsymbol{S}_D(i) = \frac{\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)} \tag{60}$$

The above expression for the matrix filter $\boldsymbol{S}_D(i)$ can be simplified by observing the quantities involved and making use of the proposed constraint $\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) = 1$. Let us consider the term $\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i)$ in the denominator of (60) and multiply it by the proposed constraint as follows:

$$\begin{aligned} \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i) &= \bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) \\ &= \bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) = 1. \end{aligned} \tag{61}$$

Now let us consider the term $\boldsymbol{a}^H(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)$ and rewrite it as follows:

$$\begin{aligned} \boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i) &= \boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}^H(i)\boldsymbol{S}_D^H(i)\boldsymbol{a}(\theta_k) \\ &= \boldsymbol{a}(\theta_k)\boldsymbol{a}^H(\theta_k)\boldsymbol{S}_D(i)\bar{\boldsymbol{w}}(i)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i) \\ &= \boldsymbol{a}(\theta_k)\boldsymbol{a}^H(\theta_k)\boldsymbol{S}_D(i) = \boldsymbol{a}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k). \end{aligned} \tag{62}$$

Using the relations obtained in (61) and (62) into the expression in (60), we can get a simpler expression for the projection matrix as given by

$$\boldsymbol{S}_D(i) = \frac{\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)}{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)} = \frac{\boldsymbol{R}^{-1}(i)\overbrace{\boldsymbol{a}(\theta_k)\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)}^{\boldsymbol{a}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k)}}{\underbrace{\bar{\boldsymbol{w}}^H(i)\bar{\boldsymbol{R}}_{\bar{w}}^{-1}(i)\bar{\boldsymbol{w}}(i)}_{1}\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)} \tag{63}$$

$$= \frac{\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)\bar{\boldsymbol{a}}^H(\theta_k)}{\boldsymbol{a}^H(\theta_k)\boldsymbol{R}^{-1}(i)\boldsymbol{a}(\theta_k)}$$

This completes the derivation and the simplification.

REFERENCES

[1] H. L. Van Trees, "Detection, Estimation, and Modulation Theory, Part IV, Optimum Array Processing", John Wiley & Sons, 2002.

[2] J. Li and P. Stoica, "Robust adaptive Beamforming", *Wiley* , 2006.

[3] S. Haykin *Adaptive Filter Theory*, Prentice Hall, 4rd ed, 2002.

[4] O. L. Frost III, "An algorithm for linearly constrained adaptive array processing" *Proceedings of IEEE*, vol. AP-30, pp. 27 - 34, Jan 1972.

[5] M. Honig, U. Madhow and S. Verdu, "Blind adaptive multiuser detection," *IEEE Trans. Inform. Theory,* vol. 41, pp. 944-960, July 1995.

[6] B. D. Van Veen, "Adaptive covergence of linearly constrained beamformers based on the sample covariance matrix", *IEEE Transactions on Signal Processing*, vol. 39, pp. 1470-1473, 1991.

[7] D. D. Feldman and L. J. Griffiths, "A projection approach for robust adaptive beamforming", *IEEE Transactions on Signal Processing*, vol. 42, pp. 867-876, 1994.

[8] L. S. Resende, J. M. T. Romano and M. G. Bellanger "A fast least-squares algoirthm for linearly constrained adaptive filtering" *IEEE Transactions on Signal Processing*, vol. 44, pp. 1168 - 1174, May 1996.

[9] L. L. Scharf and D. W. Tufts, "Rank reduction for modeling stationary signals," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-35, pp. 350-355, March 1987.

[10] L. L. Scharf and B. van Veen, "Low rank detectors for Gaussian random vectors," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-35, pp. 1579-1582, November 1987.

[11] A. M. Haimovich and Y. Bar-Ness, "An eigenanalysis interference canceler," *IEEE Trans. on Signal Processing*, vol. 39, pp. 76-84, Jan. 1991.

[12] D. A. Pados and S. N. Batalama "Joint space-time auxiliary vector filtering for DS/CDMA systems with antenna arrays" *IEEE Transactions on Communications*, vol. 47, no. 9, pp. 1406 - 1415, 1999.

[13] S. Burykh and K. Abed-Meraim "Reduced-rank adaptive filtering using Krylov subspace" *EURASIP Journal on Applied Signal Processing*, vol. 12, pp. 1387 - 1400, 2002

[14] J. S. Goldstein, I. S. Reed and L. L. Scharf "A multistage representation of the Wiener filter based on orthogonal projections" *IEEE Transactions on Information Theory*, vol. 44, no. 7, 1998.

[15] Y. Hua, M. Nikpour and P. Stoica, "Optimal reduced rank estimation and filtering," IEEE Transactions on Signal Processing, pp. 457-469, Vol. 49, No. 3, March 2001.

[16] M. L. Honig and J. S. Goldstein, "Adaptive reduced-rank interference suppression based on the multistage Wiener filter," *IEEE Transactions on Communications*, vol. 50, no. 6, June 2002.

[17] E. L. Santos and M. D. Zoltowski, "On Low Rank MVDR Beamforming using the Conjugate Gradient Algorithm", *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004.

[18] Q. Haoli and S.N. Batalama, "Data record-based criteria for the selection of an auxiliary vector estimator of the MMSE/MVDR filter", *IEEE Transactions on Communications*, vol. 51, no. 10, Oct. 2003, pp. 1700 - 1708.

[19] R. C. de Lamare and R. Sampaio-Neto, "Reduced-Rank Adaptive Filtering Based on Joint Iterative Optimization of Adaptive Filters", *IEEE Signal Processing Letters*, Vol. 14, no. 12, December 2007.

[20] R. C. de Lamare and M. J. Lowe, "A Reduced-Rank Approach to Adaptive Linearly Constrained Minimum Variance Beamforming Based on Joint Iterative Optimization of Adaptive Filters", *Proc. IEEE 9th Workshop on Signal Processing Advances in Wireless Communications*, July 2008.

[21] R. C. de Lamare, "Adaptive Reduced-Rank LCMV Beamforming Algorithms Based on Joint Iterative Optimisation of Filters", *Electronics Letters*, vol. 44, no. 9, 2008.

[22] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2nd Ed., 1999.

[23] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, Md, 1996.
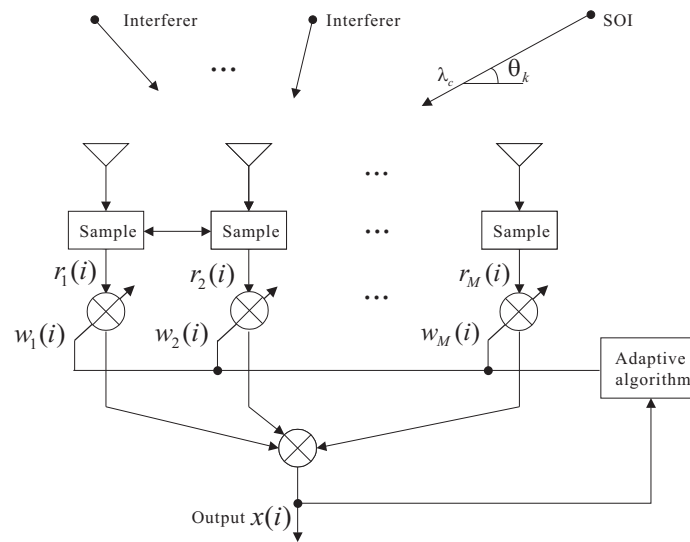
Fig. 1. Schematic of a linear antenna array system with interferers.
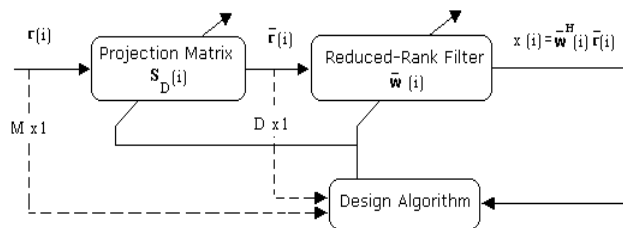

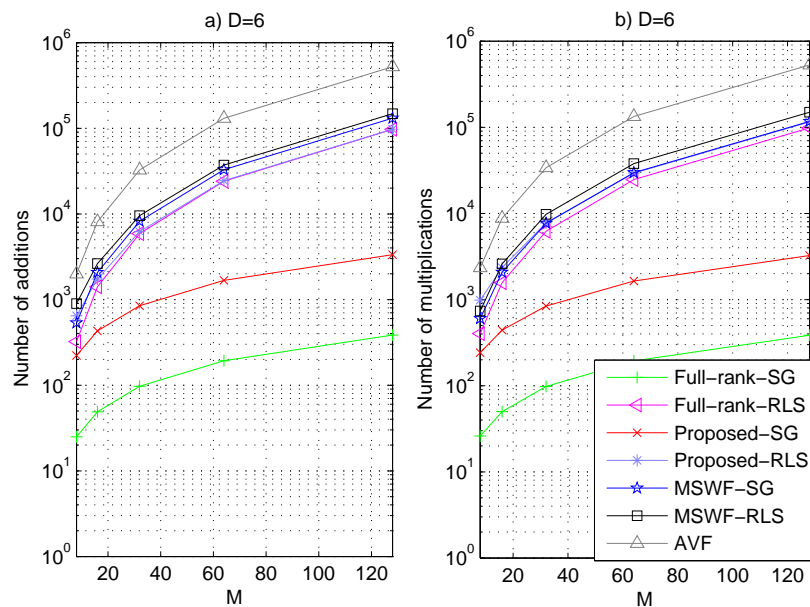Fig. 2. Schematic of the proposed reduced-rank scheme.

Fig. 3. Complexity in terms of arithmetic operations against M.

TABLE I

Computational complexity of LCMV algorithms.

Algorithm	Additions	Multiplications
Full-rank-SG [4]	$3M + 1$	$3M + 2$
Full-rank-RLS [8]	$3M^2 - 2M + 3$	$6M^2 + 2M + 2$
Proposed-SG [21]	$3DM + 2M$ $+2D - 2$	$3DM + M$ $+5D + 2$
Proposed-RLS	$3M^2 - 2M + 3$ $+3D^2 - 8D + 3$	$7M^2 + 2M$ $+7D^2 + 9D$
MSWF-SG [16]	$DM^2 - M^2$ $+3D - 2$	$DM^2 - M^2$ $+2DM + 4D + 1$
MSWF-RLS [16]	$DM^2 + M^2 + 6D^2$ $-8D + 2$	$DM^2 + M^2$ $+2DM + 3D + 2$
AVF [18]	$D((M)^2 + 3(M - 1)^2) - 1$ $+D(5(M - 1) + 1) + 2M$	$D(4M^2 + 4M + 1)$ $+4M + 2$

Fig. 4. MSE analytical versus simulated performance for the proposed reduced-rank SG algorithm.
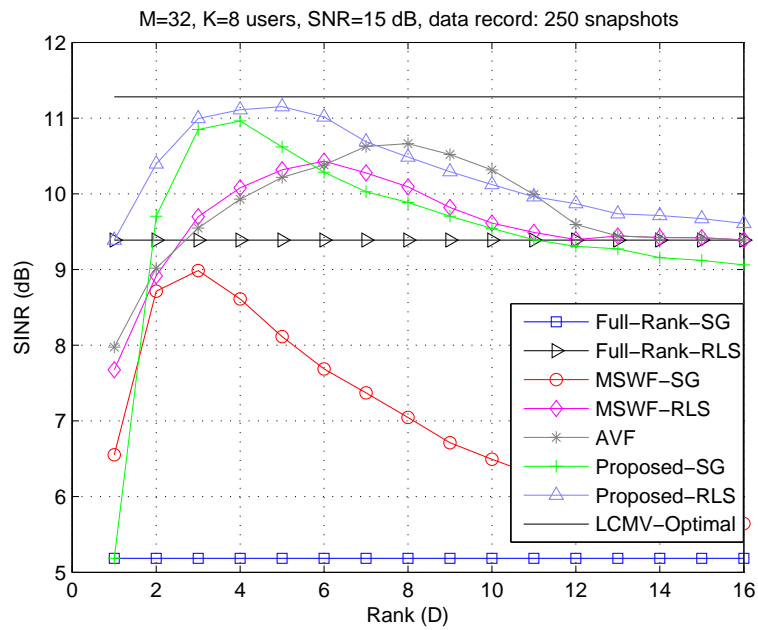
Fig. 5. SINR performance of LCMV algorithms against rank (D) with $M = 32$, $SNR = 15$ dB, $N = 250$ snapshots.
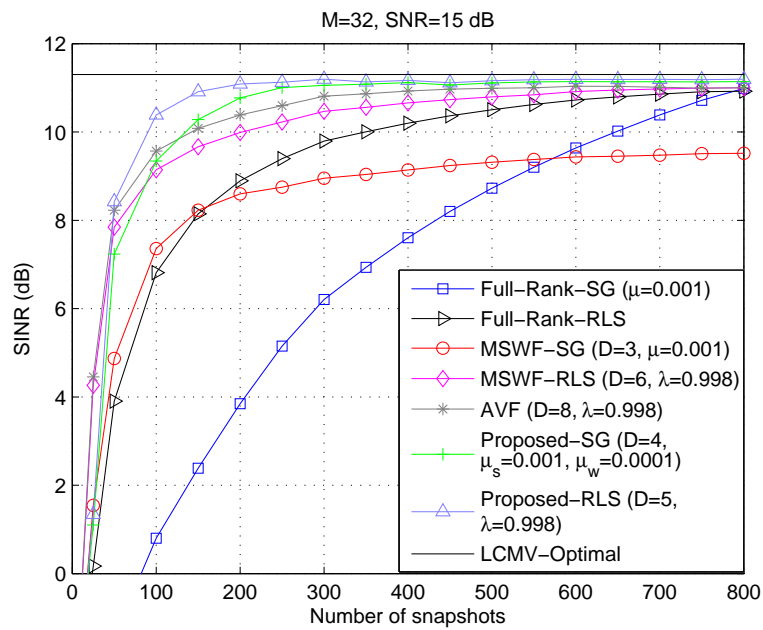

Fig. 6. SINR performance of LCMV algorithms against snapshots with $M = 32$, $SNR = 15$ dB.
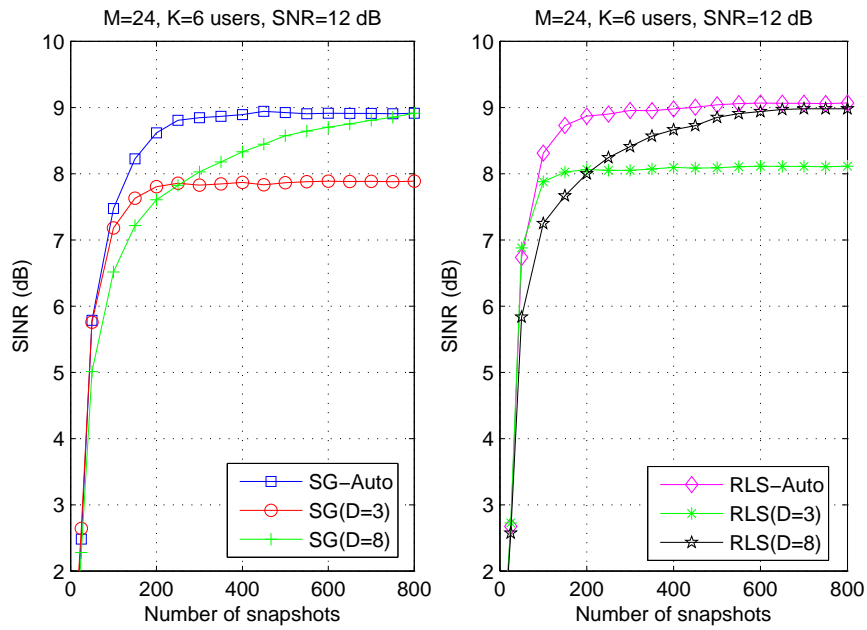
Fig. 7.   SINR performance of LCMV (a) SG and (b) RLS algorithms against snapshots with $M = 24$, $SNR = 12$ dB with automatic rank selection.
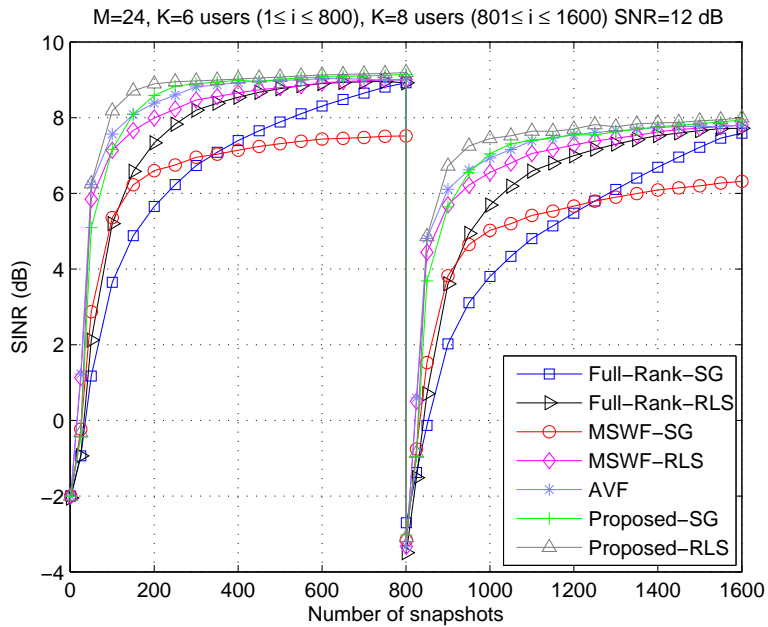

Fig. 8.   SINR performance of LCMV algorithms against snapshots with $M = 24$, $SNR = 12$ dB in a non-stationary scenario.