Robust Reduced-Rank Adaptive Processing Based on Parallel Subgradient Projection and Krylov Subspace Techniques

Masahiro Yukawa, Member, IEEE, Rodrigo C. de Lamare, Member, IEEE, and Isao Yamada, Senior Member, IEEE

Abstract—In this paper, we propose a novel reduced-rank adaptive filtering algorithm by blending the idea of the Krylov subspace methods with the set-theoretic adaptive filtering framework. Unlike the existing Krylov-subspace-based reduced-rank methods, the proposed algorithm tracks the optimal point in the sense of minimizing the ‘true’ mean square error (MSE) in the Krylov subspace, even when the estimated statistics become erroneous (e.g., due to sudden changes of environments). Therefore, compared with those existing methods, the proposed algorithm is more suited to adaptive filtering applications. The algorithm is analyzed based on a modified version of the adaptive projected subgradient method (APSM). Numerical examples demonstrate that the proposed algorithm enjoys better tracking performance than the existing methods for the interference suppression problem in code-division multiple-access (CDMA) systems as well as for simple system identification problems.

Index Terms—reduced-rank adaptive filtering, Krylov subspace, set-theory, subgradient methods

I. INTRODUCTION

Reduced-rank adaptive filtering has attracted significant attention over several research communities including signal processing; e.g., [1]-[36]. Whereas early works were motivated by the so-called overmodeling problem, many of the recent works were motivated mainly by computational constraints and slow-convergence problems due to a large number of parameters. Specifically, a Krylov subspace associated with the input autocorrelation matrix and the cross-correlation vector between input and output has been used in several methods: Cayley-Hamilton receiver [18], multistage Wiener filter (MSWF) [19], [21], [25], auxiliary-vector filtering (AVF) [23], [24], Powers of R (POR) receiver [21], and the conjugate gradient reduced-rank filter (CGRRF) [31], [32] (see [34] for their connections). All of those previous studies focus on minimizing a mean square error (MSE) within the Krylov subspace. However, in the erroneous case (i.e., in cases where there is a mismatch in estimates of the autocorrelation matrix and the cross-correlation vector), the methods minimize an ‘erroneous’ MSE function in the Krylov subspace. Therefore, the solution obtained at each iteration is no longer ‘optimal’ in the sense of minimizing the ‘true’ MSE within the Krylov subspace.

In this paper, we propose an adaptive technique, named Krylov reduced-rank adaptive parallel subgradient projection (KRR-APSP) algorithm, tracking directly the ‘optimal’ solution in the Krylov subspace. The KRR-APSP algorithm firstly performs dimensionality reduction with an orthonormal basis of the Krylov subspace, followed by adjustments of the coefficients of a lower-dimensional filter based on the set-theoretic adaptive filtering framework [1]. As a result, in cases where the environment changes dynamically (which makes the estimates of the statistics erroneous), the KRR-APSP algorithm realizes better tracking capability than the existing Krylov-subspace-based methods (The computational complexity is comparable to the existing methods).

The rest of the paper is organized as follows. In Section II, the motivation and the problem statement are presented, in which it is shown that, in a low-dimensional Krylov subspace, (i) the achievable MSE is close to the minimum MSE (MMSE) and (ii) system identification of high accuracy is possible, provided that the condition number of the autocorrelation matrix is close to unity. In Section III, we present the proposed reduced-rank algorithm, and discuss its tracking property and computational complexity. The KRR-APSP algorithm (i) designs multiple closed convex sets consistent with the recently arriving data, and (ii) moves the filter toward the intersection of the convex sets (to find a feasible solution) by means of parallel subgradient projection at each iteration. Because the noise is taken into account in the set design, KRR-APSP is intrinsically robust. In Section IV, to prove important properties (monotonicity and asymptotic optimality) of the proposed algorithm, we firstly present an alternative derivation of the algorithm from a modified version of the adaptive projected subgradient method (APSM) [2], [3], and then present an analysis of the modified APSM. It is revealed that, in the (original) high dimensional vector space, the proposed algorithm performs parallel subgradient projection in a series of Krylov subspaces. In Section V, numerical examples are presented to verify the advantages of the proposed algorithm over CGRRF, followed by the conclusion in Section VI.

1A related approach called set-membership adaptive filtering has independently been developed, e.g., in [7], [7].
2APSM has proven a promising tool to derive efficient algorithms in many applications [7], [7], [7], [7], [7], [7], [7], [7].
II. MOTIVATION AND PROBLEM STATEMENT

Let $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{N}^*$ denote the sets of all real numbers, nonnegative integers, and positive integers, respectively. We consider the following linear model:

$$d_k := u_k^T h^* + n_k, \forall k \in \mathbb{N},$$

where $u_k := [u_{k1}, u_{k2}, \cdots, u_{kN}]^T \in \mathbb{R}^N$ $(N \in \mathbb{N}^*)$ denotes the input vector, $h^* \in \mathbb{R}^N$ the unknown system, $n_k$ the additive noise, and $d_k$ the output $(k$: sample index, $(\cdot)^T$: transposition). The MMSE filter in the whole space $\mathbb{R}^N$ is well-known to be characterized by the so-called Wiener-Hopf equation $Rh^*_\text{MMSE} = p$ (see, e.g., [45]), where $R := E\{u_k u_k^T\}$ and $p := E\{u_k d_k\}$ $(E\{\cdot\}$: expectation). For simplicity, we assume that $R$ is invertible and the input and the noise are (statistically) orthogonal; i.e., $E\{n_k u_k\} = 0$. In this case, $p = E\{u_k (u_k^T h^* + n_k)\} = Rh^*$, and the MSE function $f : \mathbb{R}^N \rightarrow [0, \infty)$ is given as

$$f(h) := E\{(d_k - h^T u_k)^2\} = h^T Rh - 2h^T p + \sigma^2_d^2.$$  

(2)

Here, $\sigma^2_d := E\{d_k^2\}$ and $\|\cdot\|_R$ is the R-norm$^3$ defined for any vector $a \in \mathbb{R}^N$ as $\|a\|_R := \sqrt{a^T Ra}$. From (2), it is seen that $h^* = h_{\text{MMSE}}(= R^{-1} p)$.

Let us now consider, for $D \in \{1, 2, \ldots, N\}$, the MMSE filter within the following Krylov subspace:

$$K_D(R, p) := \text{span}\{p, Rp, \cdots, R^{D-1} p\}.$$  

(3)

$$= \text{span}\{Rh^*, R^2 h^*, \cdots, R^D h^*\} \subset \mathbb{R}^N.$$  

(4)

Referring to (2), the MMSE solution in $K_D(R, p)$ is characterized by

$$f_{K_D(R, p)}(h^*) = \arg \min_{h \in K_D(R, p)} \|h^* - h\|_R,$$  

(5)

where we denote by $P_C(x)$ the metric projection of a vector $x$ onto a closed convex set $C$ in the $A$-norm sense. In particular, the metric projection in the sense of Euclidean norm is denoted simply by $P_C(x)$. In words, the MMSE filter in the subspace is the best approximation, in the $R$-norm sense, of $h^*$ in $K_D(R, p)$. Noting that $f_{K_D(R, p)}(h^*)$ coincides with the vector obtained through $D$ steps of the conjugate gradient (CG) method with its initial point being the zero vector, the MSE is bounded as follows [46, Theorem 10.2.6]:

$$f_{K_D(R, p)}(h^*) \leq \left[ 4 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2D} - 1 \right] \|h^*\|_R^2 + \sigma^2_d,$$  

(6)

where $\kappa := \|R\|_2 \|R^{-1}\|_2 \geq 1$ is the condition number of $R$. System identifiability in $K_D(R, p)$ is discussed below.

Remark 1: How accurately can the system $h^*$ be identified in the subspace $K_D(R, p)$? In the system identification problem, we wish to minimize the Euclidean norm $\|h^* - h\|$ rather than the $R$-norm $\|h^* - h\|_R$. To clarify the difference between the MSE minimization and the system identification over $K_D(R, p)$, the projections in the different senses are illustrated in Fig. 1. By the Rayleigh-Ritz theorem [2], it is readily verified that $\lambda_{\text{max}}^{-1/2} \|x\|_R \leq \|x\| \leq \lambda_{\text{min}}^{-1/2} \|x\|_R$ for any $x \in \mathbb{R}^N$, where $\lambda_{\text{max}} > 0$ and $\lambda_{\text{min}} > 0$ denote the maximum and minimum eigenvalues of $R$, respectively. It is thus verified that $\|P_{K_D(R, p)}(h^*) - P_{K_D(R, p)}(h^*)\| \leq \|h^* - P_{K_D(R, p)}(h^*)\| \leq \lambda_{\text{min}}^{-1/2} \|h^* - P_{K_D(R, p)}(h^*)\|_R \leq 2\lambda^{-1/2} \|h^*\|_R \alpha^D(\kappa)$, where $\alpha(\kappa) := (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1) \in (0, 1)$. Here, the first inequality is due to the basic property of projection, and the second one is verified by [46, Theorem 10.2.6]. This suggests that system identification of high accuracy would be possible for a small $D$ when $\kappa \approx 1$ (if $\kappa \gg 1$, preconditioning$^4$ should be performed).

In reality, $R$ and $p$ are rarely available, thus should be estimated from observed measurements. Let $\hat{R}$ and $\hat{p}$ be estimates of $R$ and $p$, respectively, and $\hat{h}^*$ be characterized by $\hat{R}\hat{h}^* = \hat{p}$. CGRRF [?, [7], [2]] computes, at each iteration, the

$^3$ The $R$-norm is also called the energy norm induced by $R$. The same norm is used in [?] to derive the CG method.

$^4$The importance of preconditioning is well-known in numerical linear algebra; see, e.g., [47], [2] and the references therein. Also the importance is mentioned in [?] for an application of the conjugate gradient method to the adaptive filtering problem. Different types of CG-based adaptive filtering algorithms have also been proposed, e.g., in [?], [2].
best approximation of \( \hat{h} \) in \( K_D(\tilde{R}, \tilde{p}) \) in the \( \tilde{R} \)-norm sense; i.e., \( p^*_k(\tilde{R}_{D}(\tilde{R}, \tilde{p})) \). This realizes significantly fast convergence and reasonable steady-state performance as long as good estimates are available; i.e., \( \hat{R} \approx R \) and \( \hat{p} \approx p \). However, once those estimates become unreliable (which happens when the environments change suddenly), \( p^*_k(\tilde{R}_{D}(\tilde{R}, \tilde{p})) \) makes little sense, and CGRRF (or the other existing Krylov-subspace-based methods) should wait until a certain amount of data arrive to recapture reasonable estimates.

The goal of this paper is to propose an alternative to the existing Krylov-subspace-based methods to address this restriction. To be specific, the main problem in this work is stated as follows. Given that the Krylov subspace is employed for dimensionality reduction, the problem is to design an efficient algorithm that can always track \( p^*_k(\tilde{R}_{D}(\tilde{R}, \tilde{p})) \), which minimizes the true MSE \( f(h) \) over \( K_D(\tilde{R}, \tilde{p}) \) [see [2]]. Such an algorithm should have better tracking capability than the existing methods after dynamic changes of environments, because \( p^*_k(\tilde{R}_{D}(\tilde{R}, \tilde{p})) \) does not minimize the true MSE as long as the estimates \( \hat{R} \) and \( \hat{p} \) are erroneous. This is illustrated in Fig. [2] in which the estimates are assumed to become erroneous. Note in the figure that the difference between \( f(h) \) and \( |h - \hat{h}|_2^2 \) is a constant in terms of \( h \), which makes no difference in the equal error contours. In the following section, we present an adaptive algorithm that achieves the goal.

III. PROPOSED REDUCED-RANK ADAPTIVE FILTER

We firstly present a reduced-rank version of the set-theoretic adaptive filtering algorithm named adaptive parallel subgradient projection (APSP) algorithm [2]. The proposed algorithm is called Krylov Reduced-Rank Adaptive Parallel Subgradient Projection (KRR-APSP). We then show, for its simplest case, that the proposed algorithm tracks \( p^*_k(\tilde{R}_{D}(\tilde{R}, \tilde{p})) \), and discuss its computational complexity.

A. Proposed KRR-APSP Algorithm

Let \( \tilde{R}_k \) and \( \tilde{p}_k \) be estimates of \( R \) and \( p \) at time \( k \in \mathbb{N} \), respectively, and \( S_k \) an \( N \times D \) matrix whose column vectors form an orthonormal basis in the sense of the standard inner product) of the subspace \( K_D(\tilde{R}_k, \tilde{p}_k) \). For dimensionality reduction, we force the adaptive filter \( h_k \in \mathbb{R}^N \) to lie in \( K_D(\tilde{R}_k, \tilde{p}_k) \subset \mathbb{R}^N \) at each time instance \( k \). Thus, with a lower dimension \( \dim h_k \in \mathbb{R}^D \), the adaptive filter is characterized as \( h_k = S_k h_k \). In the following, a tilde will be used for expressing a \( D \)-dimensional vector (or a subset of \( \mathbb{R}^D \)).

The output of the adaptive filter is given by
\[
h_k^T u_k = \tilde{h}_k^T S_k^T u_k = \tilde{h}_k^T \bar{u}_k \quad (\bar{u}_k := S_k^T u_k \in \mathbb{R}^D).
\]

The reduced-rank adaptive filtering scheme is illustrated in Fig. 3.

The idea of set-theoretic adaptive filtering is as follows:

1) construct (possibly multiple) closed convex sets containing a desired set, i.e. \( p^*_k(\tilde{R}_{D}(\tilde{R}, \tilde{p})) \) in this case, with high probability; and

2) approach the intersection of those sets at each iteration. Let us present the design of the closed convex sets. Given \( r \in \mathbb{N} \), we define
\[
U_k := [u_k, u_{k-1}, \ldots, u_{k-r+1}] \in \mathbb{R}^{N \times r}
d_k := [d_k, d_{k-1}, \ldots, d_{k-r+1}] \in \mathbb{R}^r
e_k(h) := U_k^T h - d_k \in \mathbb{R}^r, \forall h \in \mathbb{R}^N.
\]

Then, with a simple restriction on \( h \in \mathbb{R}^N \) in the stochastic property set proposed in [7], the closed convex sets in \( \mathbb{R}^N \) are given as
\[
C_k(\rho) := \left\{ h \in \mathcal{R}(S_k) : g_k(h) := \|e_k(h)\|^2 - \rho \leq 0 \right\},
\]

\[k \in \mathbb{N}, \] (8)

where \( \rho \geq 0 \), \( \mathcal{R}(\cdot) \) stands for range, and \( \|\cdot\| \) denotes the Euclidean norm. Intuitively, \( C_k(\rho) \) is a set of filtering vectors consistent with the data observed at time \( k \) in the sense that the norm of the error-vector is bounded by a small constant \( \rho \). If \( \rho \) is too small, there could be no consistent solution; for an extreme example, if \( \rho = 0 \) and we have the data sets \( (u_k, d_k) \) and \( (u_{k+1}, d_{k+1}) \) such that \( u_k = u_{k+1} \) and \( d_k \neq d_{k+1} \), then \( C_k(\rho) \cap C_{k+1}(\rho) = \emptyset \). Note however, that even in such an infeasible case, the proposed algorithm is guaranteed to move the filter closer to all the points that minimize a weighted sum of the distances to the convex sets \( (C_k(\rho))_{k \in \mathbb{N}} \), as will be shown in Theorem [1]a in Section IV-B.

Let \( I_k \) be the control sequence at the \( k \)-th iteration; i.e., the set of indices used at time \( k \) (a typical example is \( I_k := \{k, k-1, \ldots, k-q+1\} \) for \( q \in \mathbb{N}^+ \)). Replacing \( h \) in \( C_i(\rho) \), \( r \in I_k \), by \( S_i h \), the stochastic property set in \( \mathbb{R}^D \) is obtained as follows:
\[
\tilde{C}_i(\rho) := \left\{ \tilde{h} \in \mathbb{R}^D : g_i^{(\rho)}(\tilde{h}) := \|e_i^{(\rho)}(\tilde{h})\|^2 - \rho \leq 0 \right\},
\]

\[r \in I_k, \rho \in \mathbb{R} \] (9)

Here, \( e_i^{(\rho)}(\tilde{h}) := U_i^T S_i \tilde{h} - d_i \in \mathbb{R}^r, \forall \tilde{h} \in \mathbb{R}^D \). The projection onto \( \tilde{C}_i^{(\rho)}(\rho) \) is approximated by the projection onto the simple closed half-space \( \tilde{H}_{i,k} = \tilde{C}_i^{(\rho)}(\rho) \) defined as
\[
\begin{align*}
\tilde{H}_{i,k}(\tilde{h}) := \left\{ \tilde{h} \in \mathbb{R}^D : \tilde{h} + \tilde{h}_k + \tilde{S}_i + \tilde{g}_i^{(\rho)}(\tilde{h}) \leq 0 \right\},
\end{align*}
\]

\[\forall r \in I_k, \rho \in \mathbb{R} \] (10)
where $\tilde{s}_k^{(k)} := \nabla g_k^{(k)}(\tilde{h}_k) := 2S_k^T U_k e_k^{(k)}(\tilde{h}_k) \in \mathbb{R}^D$. An important property is $\tilde{h}_k \not\in C_k^{(k)}(\rho) \Rightarrow \tilde{h}_k \not\in H_k^{(k)}(\tilde{h}_k)$ [2, Lemma 2], thus the boundary of $H_k^{(k)}(\tilde{h}_k)$ is a separating hyperplane between $\tilde{h}_k$ and $C_k^{(k)}(\rho)$. The projection of $\tilde{h}_k$ onto $H_k^{(k)}(\tilde{h}_k)$ is given as

$$P_{H_k^{(k)}(\tilde{h}_k)}(\tilde{h}_k) = \begin{cases} \tilde{h}_k & \text{if } g_k^{(k)}(\tilde{h}_k) \leq 0, \\ \tilde{h}_k - \frac{g_k^{(k)}(\tilde{h}_k)}{\|s_k^{(k)}\|} s_k^{(k)} & \text{otherwise,} \end{cases}$$

(11)

which is also referred to as the subgradient projection relative to $g_k^{(k)}$ (see Appendix A). Let $w_i^{(k)} \in \{0, 1\}$, $i \in I_k$, $k \in \mathbb{N}$, denote the weight satisfying $\sum_{i \in I_k} w_i^{(k)} = 1$; see [2] for a strategic design of the weights. Then, the proposed KRR-APS proximal algorithm is presented in what follows.

Given an arbitrary initial vector $\tilde{h}_0 \in \mathbb{R}^D$, the sequence $(\tilde{h}_k)_{k \in \mathbb{N}} \subset \mathbb{R}^D$ is inductively generated as follows. Given $\tilde{h}_k$ and $I_k$ at each time $k \in \mathbb{N}$, $\tilde{h}_{k+1}$ is defined as

$$\tilde{h}_{k+1} = \tilde{h}_k + \lambda_k \mathcal{M}_k \left( \sum_{i \in I_k} w_i^{(k)} P_{H_k^{(k)}(\tilde{h}_k)}(\tilde{h}_k) - \tilde{h}_k \right),$$

(12)

where $\lambda_k \in [0, 2]$, $\tilde{H}_k^{(k)}(\tilde{h}_k)$ is defined as in (10), and

$$\mathcal{M}_k := \begin{cases} 1 & \text{if } g_k^{(k)}(\tilde{h}_k) \leq 0, \forall i \in I_k, \\ \sum_{i \in I_k} w_i^{(k)} \left\| P_{H_k^{(k)}(\tilde{h}_k)}(\tilde{h}_k) - \tilde{h}_k \right\|^2 & \text{otherwise.} \end{cases}$$

For convenience, efficient implementation of the proposed algorithm is given in TABLE I (For computational efficiency, we introduce a parameter $m$ to control how frequently $S_k$ is updated). We mention that, although the condition for updating $\tilde{\delta}_k$ is similar to the one used in the set-membership affine projection algorithm [2], the major differences are that (i) the update is based on the subgradient projection, (ii) multiple closed convex sets are employed at each iteration (each set is indexed by an element of $I_k$), and (iii) no matrix inversion is required.

We shall finish up this subsection by summarizing the parameters used in the proposed algorithm:

- $r$: the dimension of the orthogonal complement of the underlying subspace of $C_k(0)$ (see the definition of $U_k$, and $d_k$ before (3)),
- $q$: the number of projections computed at each iteration,
- $\rho$: the error bound (controlling the ‘volume’ of $C_k(\rho)$),
- $m$: the frequency of updating $S_k$.

Although the function $g_k^{(k)}$ is differentiable, the subgradient projection can be defined also for non-differentiable functions. Note that $\text{lev}_{\leq 0}(g_k^{(k)}) := \{ \tilde{h} \in \mathbb{R}^D : g_k^{(k)}(\tilde{h}) \leq 0 \} \neq \emptyset$.

### TABLE I

**Efficient Implementation of the Proposed Algorithm.**

- **Requirements:** Initial transformation matrix $S_0$, inputs $(U_k)_{k \in \mathbb{N}}$, outputs $(d_k)_{k \in \mathbb{N}}$, control sequence $\lambda_k$, step size $\lambda_k \in [0, 2]$, weights $w_i^{(k)}$, $\forall i \in I_k$, initial vector $\tilde{h}_0 \in \mathbb{R}^D$, constant $\rho \geq 0$, $m \in \mathbb{N}^*$
- **1. Filter output:** $y_k := \hat{a}_k^T \tilde{h}_k = (U_k^T S_k \tilde{h}_k)$
- **2. Filter update:**
  - (a) For $i \in I_k$, do the following:
    - $U_i^{(k)} := S_k^T U_k \in \mathbb{R}^{D \times r}$
    - $e_i^{(k)} := (U_i^{(k)})^T \tilde{h}_k - d_i \in \mathbb{R}^r$
    - If $\|e_i^{(k)}\|^2 \leq \rho$
      - $\hat{\delta}_i^{(k)} := 0 \in \mathbb{R}^D$, $\hat{c}_i^{(k)} := 0$
    - else
      - $a_i^{(k)} := U_i^{(k)} e_i^{(k)} \in \mathbb{R}^D$
      - $c_i^{(k)} := \|a_i^{(k)}\|^2 \in [0, \infty)$
      - $d_i^{(k)} := \rho - \|w_i^{(k)}\|^2 \in (-\infty, \rho]$  
      - $\hat{\delta}_i^{(k)} := w_i^{(k)} d_i^{(k)} a_i^{(k)} / (2c_i^{(k)}) \in \mathbb{R}^D$
      - $\hat{c}_i^{(k)} := \left( \|\hat{\delta}_i^{(k)}\|^2 / \|w_i^{(k)}\|^2 \right) (w_i^{(k)} d_i^{(k)})^2 / 4c_i^{(k)} \in (0, \infty)$
  - end;
  - (b) If $\|e_i^{(k)}\|^2 \leq \rho$ for all $i \in I_k$, 
    - $\tilde{h}_{k+1} := \hat{h}_k \in \mathbb{R}^D$
  - else
    - $\tilde{\delta}_k := \sum_{i \in I_k} \hat{\delta}_i^{(k)} \in \mathbb{R}^D$
    - $M_k := \|\tilde{\delta}_k\|^2 \in \mathbb{R}$
    - $\hat{h}_{k+1} := \hat{h}_k + \lambda_k M_k \tilde{\delta}_k \in \mathbb{R}^D$
  - end;
- 3: if $k \equiv 1 \text{ mod } m$
  - Compute $S_{k+1} \in \mathbb{R}^{N \times D}$, an orthonormalized version of $K_D (R_k, \tilde{p}_k)$; see Section III-B
  - else
    - $S_{k+1} := S_k$
  - end;

Intuitively, the convex set $C_k(\rho)$ is obtained by ‘ballooning’ the linear variety used in the affine projection algorithm (APA) [53], [54], and $r$ corresponds to the ‘order’ of APA [45].

The tracking property and the computational complexity of the proposed algorithm are discussed in the following subsection.

### B. Tracking Property and Computational Complexity

As explained in the final paragraph in Section II, an algorithm that tracks $P_{K_D}(R_k, \tilde{p}_k)(H^*)$ is expected to enjoy better tracking capability than the existing Krylov-subspace-based reduced-rank methods. In this subsection, we firstly show that the proposed algorithm (or the vector $h_k = S_k \tilde{h}_k$, $k \in \mathbb{N}$, generated by the proposed algorithm) has such a property for its simplest case: $r = 1$, $\rho = 0$, $I_k = \{k\}$ (i.e., $q = 1$). In this case, the proposed algorithm is reduced to

$$\tilde{h}_{k+1} = \tilde{h}_k + \lambda_k d_k - \tilde{h}_k^T u_k u_k,$$

(13)
where \( \hat{\lambda}_k := \lambda_k / 2 \in [0, 1] \). The update equation in (13) is nothing but the NLMS algorithm (It should be mentioned that the step-size range of \( \lambda_k \) is a half of that of NLMS). Thus, (13) is a stochastic gradient algorithm for the following problem:

\[
\min_{\mathbf{h} \in \mathbb{R}^d} \mathbb{E}\{(d_k - \mathbf{h}^T \mathbf{u}_k)^2\}.
\]

This implies that \( \tilde{h}_k \) generated by (13) tracks the minimizer of (14): for details about the tracking performance of NLMS, see [7] and the references therein. Hence, noting that \( \mathbf{u}_k = \mathbf{S}_k \tilde{h}_k \), it is seen that \( \mathbf{h}_k := \mathbf{S}_k \tilde{h}_k \) tracks the solution to the following problem (which is equivalent to (14)):

\[
\min_{\mathbf{h} \in \mathbb{R}^d} \{d_k - \mathbf{h}^T \mathbf{u}_k\}^2.
\]

Referring to (2) and (5), the minimizer of (15) is generated by (13) tracks \( \mathbf{P}_R^{(R)}(\hat{\mathbf{R}}_k, \hat{\mathbf{p}}_k)(\mathbf{h}_k^*) \). This verifies that \( \mathbf{h}_k := \mathbf{S}_k \hat{h}_k \) generated by (13) tracks \( \mathbf{P}_R^{(R)}(\hat{\mathbf{R}}_k, \hat{\mathbf{p}}_k)(\mathbf{h}_k^*) \).

Now, let us move to the discussion about the computational complexity (i.e., the number of multiplications per iteration) of the proposed algorithm. For simplicity, we let \( \mathbf{R}_k : = \{k, k-1, \ldots, k-\tau + 1\} \), which is used in Section [V] We assume that, given \( \hat{\mathbf{R}}_k \) and \( \hat{\mathbf{p}}_k \), the complexity to construct the matrix \( \mathbf{S}_k \) is the same as that of CGRRF. As \( \mathbf{S}_k \) is computed every \( m \) iterations (see TABLE II, the average complexity for computing \( \mathbf{S}_k \) is \( (D-1)N^2/m + (5D-4)/m + 2(D-1)/m \).

What about the complexity to update \( \hat{\mathbf{R}}_k \) and \( \hat{\mathbf{p}}_k \)? For the system model presented in Section II the autocorrelation matrix \( \mathbf{R} \) is known to have a Toeplitz structure, provided that the input process is stationary. Hence, it is sufficient to estimate \( \mathbb{E}\{\mathbf{u}_k \mathbf{u}_k^T\} \in \mathbb{R}^N \), which can be done by \( \bar{r}_{k+1} := \gamma \bar{r}_k + u_k u_k^T, k \in \mathbb{N} \), with the forgetting factor \( \gamma \in (0, 1) \). Similarly, the vector \( \hat{\mathbf{p}}_k \) is updated as \( \hat{\mathbf{p}}_{k+1} := \gamma \hat{\mathbf{p}}_k + d_k \mathbf{u}_k \).

The rest is the complexity for the filter update. One of the distinguished advantages of the APSP algorithm is its inherently parallel structure [7], [7], [7], [7], [7], [7]. We start by considering the case where only a single processor is available. Because the matrices \( (\mathbf{U}_\ell)_\ell \in \mathbb{T}_k \) used at time \( k \) have only \( q + r - 1 \) distinct column vectors \( (\mathbf{u}_k, \mathbf{u}_{k-1}, \cdots, \mathbf{u}_{k-q+r-2}) \), the complexity to compute \( \mathbf{U}_\ell^{(k)} \) for all \( \ell \in \mathbb{T}_k \) is \( (q + r - 1)DN \). Fortunately, however, this is only required when \( \mathbf{S}_k \) is updated (every \( m \) iterations), and, when \( \mathbf{S}_k \) is not updated, only the first column of \( \mathbf{U}_\ell^{(k)} \) (i.e., \( \mathbf{S}_k^{(k)} \mathbf{u}_k \)) should be computed. This is because, when \( \mathbf{S}_k \) is not updated, it holds that \( \mathbf{U}_\ell^{(k)} = \mathbf{U}_\ell^{(k-1)} \) for \( \ell \in \mathbb{T}_k \backslash \{k\} \) and \( \mathbf{U}_k^{(k)} = [\mathbf{U}_{k-1}^{(k-1)}]_{r=1}^{r-1} \), where \( [\mathbf{A}]_{o,b} \) designates the submatrix of \( \mathbf{A} \) consisting of the \( o \) to \( b \)th column vectors. Thus, the average complexity for \( \mathbf{U}_\ell^{(k)} \) is \( (q + r - 1)DN + (m - 1)DN/m \). For the same reason as \( \mathbf{U}_\ell^{(k)} \in \mathbb{T}_k \), the matrices \( (\mathbf{U}_\ell^{(k)})_\ell \in \mathbb{T}_k \) also have only \( q + r - 1 \) distinct column vectors, hence the complexity to compute \( \mathbf{e}_\ell^{(k)} \) is no more than \( 2(q + r - 1)D \). Overall, the total complexity for the filter update is \( \alpha(q, r, m)DN + (4q + 2r)D + (r + 7)q + 2, \) where \( \alpha(q, r, m) := (q + m - 2)/m \).

TABLE II

| Algorithm       | Number of multiplications per iteration |
|-----------------|----------------------------------------|
| NLMS            | \( 3N^2 + 2 \)                        |
| RLS             | \( 4N^2 + 4N + 1 \)                    |
| CGRRF           | \( (D - 1)N^2/m + (5D - 4)/m + 4N + 2(D - 1) \) |
| KRR-APSP        | \( (D - 1)N^2/m + 2 + (4q + 2r)D + (r + 7)q + 2 \) |
| KRR-APSP        | \( (D - 1)N^2/m + 2 + (4q + 2r)D + (r + 7)q + 2 \) |
| (single processor) | \( (D - 1)N^2/m + 2 + (4q + 2r)D + (r + 7)q + 2 \) |
| (q processors)  | \( (D - 1)N^2/m + 2 + (4q + 2r)D + (r + 7)q + 2 \) |

Finally, we consider the case where \( q \) parallel processors are available. In this case, the computation of the variables corresponding to each \( \ell \in \mathbb{T}_k \) is naturally assigned to each processor. We consider the complexity imposed on each processor at each iteration. The complexity to compute \( \mathbf{U}_\ell^{(k)} \) is \( rDN \), when \( \mathbf{S}_k \) is updated, and \( DN \), when \( \mathbf{S}_k \) is not updated. The average complexity is thus \( \beta(r, m)DN \), where \( \beta(r, m) := (r + m - 1)/m \). Overall, the per-processor complexity for the filter update is \( \beta(r, m)DN + (2r + 4)D + r + 9 \). For \( D = 5, m = 10, r = 1, \) and an arbitrary \( q \), the complexity for the filter update is \( 5N + 40 \).

In TABLE III the overall complexity of the proposed algorithm is summarized with those of the NLMS algorithm, the RLS algorithm [45, Table 9.1], and CGRRF [7]; we assume for fairness that CGRRF updates the filter every \( m \) iterations. Figure 4 plots the number of multiplications against the filter length \( N \) for \( D = 5, m = 10, r = 1, \) and \( q = 5 \) (which are used in Section [V-B]). We can see that the complexity of the proposed algorithm is much lower than that of RLS (due to the factor \( m \)), and marginally higher than that of CGRRF; in particular, for a large value of \( N \), the difference between

---

7 The Lanczos method, which is essentially equivalent to the CG method [46], can also be used for constructing \( \mathbf{S}_k \).

8 If, for example, the system model presented in Section [V-C] is to be considered, then \( \mathbf{R} \) is not Toeplitz in general. In such a case, at least the upper triangular portion of \( \mathbf{R} \) should be estimated (Note that \( \mathbf{R} \) is always symmetric).
the proposed and CGRRF methods is negligible. Moreover, compared with NLMS, the proposed algorithm requires higher complexity for realizing better performance. However, the difference can be significantly reduced by increasing \( m \); in our experiments, the use of \( m = 100 \) gives almost the same performance as the use of \( m = 10 \). It should be mentioned that the difference (in computational complexity) between CGRRF and KRR-APSP can be further reduced by taking into account the update date of the vector \( \bar{h}_k \) (i.e., the rate in which it happens that \( \| e_k \|_2^2 \leq \rho \)). If we choose \( \rho \) appropriately, the update rate is typically less than 10%.

In conclusion, the proposed algorithm is highly expected to realize, with comparable computational complexity, superior tracking performance to the existing Krylov-subspace-based reduced-rank methods, as will be verified by simulations in Section IV. Moreover, the algorithm has a fault tolerance nature thanks to its inherently parallel structure; i.e., even if some of the engaged concurrent processors are crashed, the lack of information from the crashed processors would not cause any serious degradation in performance. This is because the direction of update is determined by taking into account all the directions suggested by each input data vector little by little.

In the following section, we present an analysis of the proposed algorithm.

IV. ANALYSIS OF THE PROPOSED ALGORITHM

In the adaptive filtering or learning, the observed measurements are mostly corrupted by noise and the environments are nonstationary in many scenarios. Under such uncertain situations, it is difficult (or nearly impossible) to guarantee that the adaptive filter approaches the optimal one monotonically at every iteration. Thus, a meaningful and realistic property desired for an adaptive algorithm would be to approach every point in an appropriately designed set of filtering vectors monotonically at each iteration. How can such a set, say \( \Omega_k \subset \mathbb{R}^N \), be designed?

In our analysis, we let \( \Theta_k : \mathbb{R}^N \to [0, \infty) \) be a (continuous and convex) objective function, and \( \Omega_k \) is defined as a set of all the vectors that achieve the infimum of \( \Theta_k \) over a certain constraint set. (The constraint is associated with the requirements that the filter should lie in the Krylov subspace.) Then, the desired monotone approximation property is expressed as follows:\(^9\)

\[
\| h_{k+1} - h^*_k \| \leq \| h_k - h^*_k \|, \quad \forall h^*_k \in \Omega_k, \quad k \in \mathbb{N}. \quad (16)
\]

We stress that (16) insists that the monotonicity holds for all the elements of \( \Omega_k \).

What about ‘optimality’ in terms of the objective function \( \Theta_k \)? Is it possible to prove ‘optimality’ in any sense? As you might notice, the objective function \( \Theta_k \) depends on \( k \). Namely, what we should ‘minimize’ is not a fixed objective function but is a sequence of objective functions \( \{ \Theta_k \}_{k \in \mathbb{N}} \). This is the major difference from the normal optimization problems, and this formulation naturally fits the adaptive signal processing because the objective function should be changing in conjunction with changing environments. Thus, a meaningful ‘optimality’ to show would be that \( (h_k)_{k \in \mathbb{N}} \) minimizes \( (\Theta_k)_{k \in \mathbb{N}} \) asymptotically; i.e.,

\[
\lim_{k \to \infty} \Theta_k(h_k) = 0,
\]

which is called asymptotic optimality \(^7\), \(^7\).

The goal of this section is to prove that the proposed algorithm enjoys the two desired properties (16) and (17). To this end, we firstly build, with the objective function \( \Theta_k \), a unified framework named reduced-rank adaptive projected subgradient method (R-APSM), and derive the proposed algorithm from R-APSM with a specific design of \( \Theta_k \). We then prove that R-APSM, including the proposed algorithm as its special case, has the desired properties under some mild conditions.

A. Alternative Derivation of the Proposed Algorithm

Recall here that \( h_k \) is forced to lie in \( \mathcal{R}(S_k) \) at each iteration \( k \in \mathbb{N} \). For an analysis of the proposed algorithm, we define

\[
\Phi_k := S_{k+1} S_k^T \in \mathbb{R}^{N \times N}.
\]

Given an arbitrary \( h_0 \in \mathbb{R}^N \) and a sequence of continuous convex objective functions \( \Theta_k : \mathbb{R}^N \to [0, \infty), \quad k \in \mathbb{N} \), R-APSM \(^10\) generates a sequence \( (h_k)_{k \in \mathbb{N}} \subset \mathbb{R}^N \) by

\[
h_{k+1} := \begin{cases} 
\Phi_k \left[ h_k - \lambda_k \left( \Theta_k(h_k) - \Theta_k'(h_k) h_k \right) \right], & \text{if } \Theta_k'(h_k) \neq 0, \\
\Phi_k h_k, & \text{otherwise,}
\end{cases}
\]

where \( \lambda_k \in [0, 2], \quad k \in \mathbb{N} \), and \( \Theta_k'(h_k) \in \partial \Theta_k(h_k) \) is a subgradient of \( \Theta_k \) at \( h_k \) (see Appendix A). Suppose that \( \text{lev}_{\leq 0} \Theta_k := \{ h \in \mathbb{R}^N : \Theta_k(h) \leq 0 \} \neq \emptyset \) (\( \Leftrightarrow \min_{h \in \mathbb{R}^N} \Theta_k(h) = 0 \)). Then, removing \( \Phi_k \) in (19) for \( \lambda_k = 1 \)

\(^9\)To ensure \( \Omega_k \)’s closedness and convexity of \( \Omega_k \) are essential.

\(^{10}\)The original APSM \(^7\), \(^7\) is obtained by replacing \( \Phi_k \) in (19) by a projection operator onto a closed convex set of an absolute constraint.
is the subgradient projection relative to $\Theta_k$ [cf. (11)], which is denoted by $T_{sp}(\Theta_k) (h_k)$ (see Fig. 5). The update equation in (19) can be expressed as

$$h_{k+1} := \Phi_k \left[ h_k + \lambda_k \left( T_{sp}(\Theta_k) (h_k) - h_k \right) \right].$$  \hspace{1cm} (20)

Noticing that the thick arrow in Fig. 5 expresses $T_{sp}(\Theta_k) (h_k) - h_k$, the figure with (20) provides a geometric interpretation of R-APSM.

Let us now derive the proposed algorithm from R-APSM. Let $I_k$ be the control sequence, and $u_{i}^{(k)} \in (0, 1)$, $i \in I_k$, $k \in \mathbb{N}$, the weight, both of which are defined in the same way as in Section III-A. An outer approximating closed half-space $H^-_i (h_k) \supset C_i (\rho_k)$ is defined as [see (8)]

$$H^-_i (h_k) := \left\{ h \in \mathbb{R}^n : \left< h - h_k, s_i^{(k)} \right> + g_i (h_k) \leq 0 \right\},$$  \hspace{1cm} (21)

where $s_i^{(k)} := \nabla g_i (h_k) := 2U_i e_i (h_k) \in \mathbb{R}^n$. Because

1) $H^-_i (h_k)$, $i \in I_k$, contains favorable vectors because of the definition of $C_i (\rho_k)$,

2) $h_k$ should lie in $R (S_k) = \mathcal{K}_D (\tilde{R}_k, \tilde{p}_k)$, the distance to $H^-_i (h_k) \cap R (S_k)$ is a natural candidate of objective function.

For more weight to assigning a larger weight to a farther set, the weight $d (h_k, H^-_i (h_k) \cap R (S_k))$ is given to the distance function $d (h_k, H^-_i (h_k) \cap R (S_k))$.

With a normalization factor $L_k := \sum_{i \in I_k} u_i^{(k)}$, the resulting objective function is given as follows:

$$\Theta_k (h_k) := \begin{cases} \frac{1}{L_k} \sum_{i \in I_k} u_i^{(k)} d (h_k, H^-_i (h_k) \cap R (S_k)) & \text{if } L_k \neq 0, \\ \times d (h_k, H^-_i (h_k) \cap R (S_k)) & \text{otherwise.} \end{cases}$$

An application of R-APSM to $\Theta_k (h_k)$ in (21) yields (cf. [?])

$$h_{k+1} = \Phi_k \left[ h_k + \lambda_k M_k \left( \sum_{i \in I_k} u_i^{(k)} P_{H^-_i (h_k) \cap R (S_k)} (h_k) - h_k \right) \right],$$

where $\lambda_k \in [0, 2]$, $k \in \mathbb{N}$, and

$$M_k := \begin{cases} 1 \text{ if } g_i (h_k) \leq 0, \forall i \in I_k, \\ \sum_{i \in I_k} u_i^{(k)} \left\| P_{H^-_i (h_k) \cap R (S_k)} (h_k) - h_k \right\|^2 & \text{if } g_i (h_k) \leq 0, \\ \left\| \sum_{i \in I_k} u_i^{(k)} P_{H^-_i (h_k) \cap R (S_k)} (h_k) - h_k \right\|^2 & \text{otherwise.} \end{cases}$$

Noticing $h_k \in R (S_k)$ and defining $Q_k := S_k S_k^T$, the projection of $h_k$ onto $H^-_i (h_k) \cap R (S_k)$ is given as follows:

$$P_{H^-_i (h_k) \cap R (S_k)} (h_k) = \begin{cases} h_k - g_i (h_k) \frac{Q_k s_i^{(k)}}{Q_i s_i^{(k)}}, & \text{if } g_i (h_k) \leq 0, \\ h_k, & \text{otherwise.} \end{cases}$$

Letting $h_k = S_k \tilde{h}_k$, we obtain $e_i (h_k) = e_i \left( \tilde{h}_k \right)$, $g_i (h_k) = g_i \left( \tilde{h}_k \right)$, and $S_k^T s_i^{(k)} = e_i^{(k)}$, from which and $P_{H^- (h_k) \cap R (S_k)} (h_k) \in R (S_k)$ we can verify

$$P_{H^-_i (h_k) \cap R (S_k)} (h_k) = S_k P_{H^-_i (h_k) \cap R (S_k)} (\tilde{h}_k).$$ \hspace{1cm} (24)

Substituting (24) and $h_k = S_k \tilde{h}_k$ into (22), and left-multiplying both sides of (22) by $S_k^T$, we obtain the proposed algorithm. Taking a look at the update equation in (22), it is seen that it has the same form as the linearly constrained adaptive filtering algorithm [7] except for the mapping $\Phi_k$ from $R (S_k)$ to $R (S_{k+1})$. Hence, viewing the behavior of the proposed algorithm in $\mathbb{R}^N$, it performs parallel subgradient projection in a series of (constraint) Krylov subspaces $R (S_k) \in \mathbb{R}^n$.

B. Analysis of R-APSM

We prove that sequence $(h_k)_{k \in \mathbb{N}}$ generated by R-APSM satisfies the desired properties (15) and (17). In the analysis, the fixed point set of the ‘mapping’ $\Phi_k := S_k S_k^T : \mathbb{R}^n \rightarrow R (S_{k+1})$, $a \mapsto \Phi_k a$, plays an important role. What is the fixed point set? Given a mapping $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$, a point $x \in \mathbb{R}^n$ satisfying $T (x) = x$ is called a fixed point of $T$. Moreover, the set of all such points, i.e. the set $\text{Fix} (T) := \{ x \in \mathbb{R}^n : T (x) = x \}$, is called the fixed point set of $T$. The set $\text{Fix} (\Phi_k)$ is characterized as below.

Proposition 1: (Characterizations of $\text{Fix} (\Phi_k)$)

(a) $0 \in \text{Fix} (\Phi_k)$.

(b) $\text{Fix} (\Phi_k) \subset R (S_k) \cap R (S_{k+1}).$

(c) $\text{Fix} (\Phi_k) = \left\{ S_k \tilde{z} = S_{k+1} \tilde{z} : \tilde{z} \in \text{Fix} \left( S_k^T S_{k+1} \right) \subset \mathbb{R}^D \right\}.$

Proof: See Appendix \[\Box\]

Define

$$\Theta_k^* := \inf_{x \in \text{Fix} (\Phi_k)} \Theta_k (x), \quad k \in \mathbb{N},$$

and

$$\Omega_k := \{ h \in \text{Fix} (\Phi_k) : \Theta_k (h) = \Theta_k^* \}, \quad k \in \mathbb{N}. \hspace{1cm} (28)$$

(As mentioned before [19], the constraint set $\text{Fix} (\Phi_k)$ is associated with the requirements $h_k \in R (S_k)$ for any $k \in \mathbb{N}$.) Then, the following theorem holds.

Theorem 1: The sequence $(h_k)_{k \in \mathbb{N}}$ generated by R-APSM satisfies the following.

(a) (Monotone Approximation)

Assume $\Omega_k \neq \emptyset$. Then, for any $\lambda_k \in \left( 0, 2 \left( 1 - \Theta_k^* / \Theta_k (h_k) \right) \right)$, (16) holds.

(b) (Assume in addition $\Theta_k (h_k) > \inf_{x \in \mathbb{R}^N} \Theta_k (x) \geq 0$)

Then, for any $\lambda_k \in \left( 0, 2 \left( 1 - \Theta_k^* / \Theta_k (h_k) \right) \right)$,

$$\left\| h_{k+1} - h^*_k \right\| < \left\| h_k - h^*_k \right\|, \quad \forall h^*_k \in \Omega_k.$$
(29)

(b) (Boundedness, Asymptotic Optimality) Assume
\[ \exists K_0 \in \mathbb{N} \text{ s.t. } \begin{cases} \Theta_k = 0, \forall k \geq K_0, \text{ and} \\ \Omega := \bigcap_{k \geq K_0} \Omega_k \neq \emptyset. \end{cases} \]  

Then \((h_k)_{k \in \mathbb{N}}\) is bounded. In particular, if there exist \(\varepsilon_1, \varepsilon_2 > 0\) such that \(\lambda_k \in ]\varepsilon_1, 2 - \varepsilon_2[ \subset (0, 2)\), then (17) holds, provided that \((\Theta_k'(h_k))_{k \in \mathbb{N}}\) is bounded.

**Proof:** See Appendix[11].

Finally, for the \(\Theta_k\) specified by (21), we discuss the assumptions made in Theorem[11]. First, it is worth mentioning that \(S_k\) tends to stop moving when the estimates of \(\hat{R}\) and \(p\) become reliable, and, in such a case, Proposition[11] implies \(\text{Fix}(\Phi_k) = \mathcal{R}(S_k)\). Hence, we assume \(\text{Fix}(\Phi_k) = \mathcal{R}(S_k)\) for simplicity here. Moreover, it mostly holds that \(\bigcap_{k \in \mathcal{I}_k} H_k^{-1}(h_k) \cap \mathcal{R}(S_k) \neq \emptyset\) at each \(k \in \mathbb{N}\), unless the observed data are highly inconsistent. In this case, \((\Theta_k^*, 0)\) holds. We remark that, under \(\text{Fix}(\Phi_k) = \mathcal{R}(S_k)\), the condition \(\bigcap_{k \in \mathcal{I}_k} H_k^{-1}(h_k) \cap \mathcal{R}(S_k) \neq \emptyset\) is sufficient but not necessary for (16) to hold. (In fact, \(\Omega_k\) can be nonempty even if \(\bigcap_{k \in \mathcal{I}_k} H_k^{-1}(h_k) = \emptyset\).)

Under \(\text{Fix}(\Phi_k) = \mathcal{R}(S_k)\), the conditions in (30) are satisfied when \(\bigcap_{k \geq K_0} \bigcap_{l \in \mathcal{I}_k} H_l^{-1}(h_k) \cap \mathcal{R}(S_k) \neq \emptyset\), which mostly holds if the observed data are consistent for \(k \geq K_0\). We mention that \((\Theta_k'(h_k))_{k \in \mathbb{N}}\) for the \(\Theta_k\) in (21) is automatically bounded[2].

In dynamic environments, it is hardly possible to ensure \(\text{Fix}(\Phi_k) = \mathcal{R}(S_k)\) for all \(k \geq K_0\), since \(S_k\) will move when the environments change. In this case, the asymptotic optimality is difficult to be guaranteed. However, it is possible that the monotone approximation is guaranteed, because the environments would be nearly static in some (short) periods and, within such periods, \(S_k\) may stop moving.

**V. Numerical Examples**

This section provides numerical examples to verify the advantages of the proposed algorithm over the CGRRF method[2] (Note: we omit a comparison with the RLS algorithm, because it is known that CGRRF provides convergence comparable to RLS with lower computational complexity and it does not suffer from any numerical instability problems[2], [7]). In the current study, weakly correlated input signals are employed in order to avoid preconditioning for conciseness. In simple system identification problems, we firstly examine the performance of the proposed algorithm for different values of \(D\) and \(q\), and then compare the proposed algorithm with CGRRF. We finally apply the two methods to a multiple access interference suppression problem in code-division multiple-access (CDMA) wireless communication systems. In all the simulations, we set \(\mathcal{I}_k := \{k, k - 1, \cdots, k - q + 1\}\), and the matrix \(S_k\) is updated every \(m = 10\) iterations with \(\hat{R}_0 := O\), \(\hat{p}_0 := 0\), and \(\gamma = 0.999\).

**A. Performance of the Proposed Algorithm for System Identification**

To compute arithmetic averages of MSE and system mismatch, i.e. \(\|h^* - \hat{h}_k\|^2 / \|h^*\|^2\), 300 independent experiments are performed. In each experiment, \(h^*\) is generated randomly for \(N = 50\), and the input signal is generated by passing a white Gaussian signal through a length-30 finite impulse response (FIR) filter whose coefficients are chosen randomly (the resulting input signal has weak autocorrelation). The signal to noise ratio (SNR) is set to \(\text{SNR} := 10 \log_{10} \left( E \left\{ z_k^2 \right\} / E \left\{ n_k^2 \right\} \right) = 15\) dB, where \(z_k := \langle u_k, h^* \rangle\).

The parameters are set \(\lambda_0 = 0.03, \rho = 0.15, q = 4, r = 1, \hat{h}_0 = 0, D = 3, 5, 8\). The results are depicted in Fig. 6. It is seen that, from \(D = 3\) to \(D = 5\), an increase of \(D\) leads to better steady-state performance both in system mismatch and MSE. However, from \(D = 5\) to \(D = 8\), the gain in MSE is slight, although a significant gain is obtained in system mismatch. This is because the value of \(\|h_k - h^*\|\) at the steady state is still not small enough in the case of \(D = 5\), but the value of \(\|h_k - h^*\|_R\) is already small enough (see Section[II]).

Next we fix the value of \(D = 8\), and change the value of \(q\) as \(q = 1, 2, 3, 5, 8\). The rest of the parameters are the same as in Fig. 6. The results are depicted in Fig. 7. As a benchmark, the performance curves of NLMS for step size \(\lambda_k = 0.03\)
are also drawn. It is seen that an increase of \( q \) (the number of parallel projections computed at each iteration) raises the speed of convergence significantly.

### B. Proposed versus CGRRF for System Identification

We compare the performance of the proposed algorithm with CGRRF and the NLMS algorithm. The \( h^* \) and the input signals are generated in the same way as in Section V-A and the SNR is set to \( \text{SNR} = 20 \text{ dB} \). We consider the situation where \( h^* \) changes dynamically at 1000th iteration; the input statistics are unchanged, which means that only the crosscorrelation vector \( p \) is changed. For all the algorithms (except for CGRRF), the step size is set to \( \lambda_k = 0.05 \), and for the proposed algorithm, we set \( \rho = 0.1 \), \( q = 1,5 \), \( r = 1 \), \( h_0 = 0 \), and \( D = 5 \). For CGRRF, the Krylov subspace dimension is set also to \( D = 5 \), and the initial vector at each time instant is set to the zero vector.

Figure 8 plots the results. As expected from the discussion in Section II, the tracking speed of CGRRF after the sudden change of \( h^* \) is slow, although its convergence speed at the initial phase is fast. On the other hand, the proposed algorithm for \( q = 5 \) achieves fast initial convergence and good tracking performance simultaneously.

### C. Proposed versus CGRRF for Interference Suppression Problem in CDMA Systems

We apply the proposed algorithm and CGRRF to the multiple access interference suppression problem occurring in
the CDMA systems (see, e.g., [?]). The received data vector, corresponding to the input vector \( u_k \), is given as

\[
    u_k := S A b_k + w_k. \tag{31}
\]

Here, letting \( K \) denote the number of users accessing the same channel, \( S \in \mathbb{R}^{N \times K} \) is the signature matrix (each column corresponds to each user), \( A \in \mathbb{R}^{K \times K} \) a diagonal matrix with the amplitudes from the \( K \) users, \( b_k \in \{1, -1\}^K \) the data symbol vector of the \( K \) users, and \( w_k \in \mathbb{R}^N \) the vector of additive white Gaussian noise with zero mean. The output \( d_k \) in Fig. 3 corresponds to the element of \( S \) associated with the desired user. For simplicity, we assume chip-synchronous but code-asynchronous systems, as usual in the literature on this problem, and fading of the channels is not considered. Also we assume that the training sequence is available to adapt the filter \( h_k \). For the spreading codes, the length-31 Gold sequences are employed (i.e., \( N = 31 \)).

In the first simulation, we assume static environments with \( K = 8 \) users having equal amplitudes under \( \text{SNR} = 15 \text{ dB} \). We set \( D = 5 \) for both CGRRF and the proposed algorithm, and \( \lambda_k = 0.02, \rho = 0.01, r = 1, \) and \( q = 1, 5 \) for the proposed algorithm. At the iteration \( k = 0 \), the rank-reduction matrix \( S_1 \in \mathbb{R}^{N \times D} \) is firstly computed, and then the lower-dimensional adaptive filter \( \tilde{h}_k \) is initialized as \( \hat{h}_0 := S_1^T s \), where \( s \in \mathbb{R}^N \) is the signature vector of the desired user. For CGRRF, the initial vector at each time instant is set to \( s \). The results are depicted in Fig. 9.

In the second simulation, we assume dynamic environments under \( \text{SNR} = 10 \text{ dB} \). At the beginning, there are \( K = 4 \) users accessing the same channel simultaneously, and, at the bit number 1000, all the interfering users stop their access and another interfering user establishes a new connection to the channel (i.e., the total number of accessing users after the bit number 1000 is \( K = 2 \)). All the interfering signals have twice larger amplitudes than the desired one. For the proposed algorithm, we set \( \rho = 0.1 \) and the other parameters are the same as in the first simulation. The parameters for CGRRF are the same as in the first simulation. The results are depicted in Fig. 10.

From Fig. 9 it is seen that the proposed algorithm (for \( q = 5 \)) performs similarly to CGRRF in the static environments. From Fig. 10 on the other hand, it is seen that the proposed algorithm exhibits better tracking performance than CGRRF. This is consistent with the results in Fig. 8 and also with the discussion in Section II.

VI. CONCLUSION

This paper has presented a robust reduced-rank adaptive filtering algorithm based on the Krylov subspace and the theoretic adaptive filtering method. The proposed algorithm provides excellent tradeoff between performance (in particular, tracking capability) and computational complexity. The valuable properties (monotone approximation and asymptotic optimality) of the proposed algorithm have been proven within the framework of the modified APSM. It would be worth repeating that the algorithm has a fault tolerance nature due to its inherently parallel structure. The numerical examples have demonstrated that the proposed algorithm exhibits much better tracking performance than CGRRF (with comparable computational complexity). This suggests that the proposed algorithm should perform better than the existing Krylov-subspace-based reduced-rank methods in nonstationary environments. We finally mention that the proposed algorithm has no numerical problems, since it requires no matrix inversion, which implies that the algorithm is easy to implement.

APPENDIX A

MATHEMATICAL DEFINITIONS

Let \( \mathcal{H} \) denote a real Hilbert space equipped with an inner product \((\cdot, \cdot)\) and its induced norm \(\|\cdot\|\). We introduce some mathematical definitions used in this paper.

(a) A set \( C \subseteq \mathcal{H} \) is said to be convex if \( \nu x + (1 - \nu) y \in C \),\( \forall x, y \in C, \forall \nu \in (0, 1). \) A function \( \Theta : \mathcal{H} \to \mathbb{R} \) is said to be convex if \( \Theta(\nu x + (1 - \nu) y) \leq \nu \Theta(x) + (1 - \nu) \Theta(y), \forall x, y \in \mathcal{H}, \forall \nu \in (0, 1) \); the inequality is sometimes called Jensen’s inequality [?].

(b) A mapping \( T : \mathcal{H} \to \mathcal{H} \) is said to be (i) nonexpansive if \( \|T(x) - T(y)\| \leq \|x - y\|, \forall x, y \in \mathcal{H} \); (ii) attracting nonexpansive if \( T \) is nonexpansive with \( \text{Fix}(T) \neq \emptyset \) and \( \|T(x) - f\|^2 \leq \|x - f\|^2, \forall x, f \in \mathcal{H} \text{ Fix}(T) \times \text{ Fix}(T) \); and (iii) strongly or \( \eta \)-attracting nonexpansive if \( T \) is nonexpansive with \( \text{Fix}(T) \neq \emptyset \) and there exists \( \eta > 0 \) s.t. \( \eta \|x - T(x)\|^2 \leq \|x - f\|^2 - \|T(x) - f\|^2, \forall x \in \mathcal{H}, \forall f \in \text{ Fix}(T) \).

(c) Given a continuous convex function \( \Theta : \mathcal{H} \to \mathbb{R} \), the subdifferential of \( \Theta \) at any \( y \in \mathcal{H} \), defined as \( \partial \Theta(y) := \{a \in \mathcal{H} : (x - y, a) + \Theta(y) \leq \Theta(x), \forall x \in \mathcal{H} \} \), is nonempty. An element of the subdifferential \( \partial \Theta(y) \) is called a subgradient of \( \Theta \) at \( y \).

(d) Suppose that a continuous convex function \( \Theta : \mathcal{H} \to \mathbb{R} \) satisfies \( \text{lev}_{\leq 0} \Theta := \{x \in \mathcal{H} : \Theta(x) \leq 0\} \neq \emptyset \). Then, for a subgradient \( \Theta'(x) \in \partial \Theta(x) \), a mapping \( T_{sp(\Theta)} : \mathcal{H} \to \mathcal{H} \) defined by

\[
    T_{sp(\Theta)}(x) := \begin{cases} 
    x - \frac{\Theta(x)}{\|\Theta'(x)\|^2} \Theta'(x) & \text{if } \Theta(x) > 0 \\
    x & \text{if } \Theta(x) \leq 0 
    \end{cases}
\]
is called a subgradient projection relative to $\Theta$ (see, e.g., [?]).

APPENDIX B

PROPERTIES OF $\Phi_k$ AND PROOF OF PROPOSITION 1

This appendix presents basic properties of $\Phi_k$, the proof of Proposition 1 and some results regarding the attracting nonexpansivity of $\Phi_k$ (see Appendix A).

**Lemma B.1:** (Basic properties of $\Phi_k$)

(a) $\Phi_k x = S_{k+1} \bar{x}$ for all $\bar{x} \in \mathbb{R}^D$ and $x = S_k \bar{x}$.

(b) For any $x \in \mathbb{R}^N$, $\|\Phi_k x\| \leq \|x\|$; the equality holds if and only if $x \in \mathcal{R}(S_k)$. Moreover, the mapping $\Phi_k$ is nonexpansive (cf. Appendix A).

**Proof of Lemma B.1:**

For all $\bar{x} \in \mathbb{R}^D$, we have $\Phi_k x = S_{k+1} S_k \bar{x} = S_{k+1} \bar{x}$.

**Proof of Lemma B.1:**

Then, for any $x \in \mathbb{R}^N$,

$$\|\Phi_k x\| = \left\|S_{k+1} S_k x\right\| = \left\|S_k S_k^T x\right\| = \left\|P_{\mathcal{R}(S_k)}(x)\right\| \leq \|x\| \quad \text{(B.1)}$$

The inequality is verified by the nonexpansivity of the projection operator; the equality holds if and only if $x \in \mathcal{R}(S_k)$.

**Proof of Proposition 1**

**Proof of Proposition 1a:** $\Phi_k 0 = 0$ implies $0 \in \text{Fix}(\Phi_k)$.

**Proof of Proposition 1b:** Suppose $h \in \text{Fix}(\Phi_k)$. Then, $h = \Phi_k h \implies h \in \mathcal{R}(S_{k+1})$. Moreover, by Lemma B.1b, $\Phi_k h = h \implies h \in \mathcal{R}(S_k)$. Hence $h \in \mathcal{R}(S_k) \cap \mathcal{R}(S_{k+1})$, implying that $\text{Fix}(\Phi_k) \subseteq \mathcal{R}(S_k) \cap \mathcal{R}(S_{k+1})$. 

**Proof of Proposition 1c:** To prove (26), it is sufficient to show

$$S_k^T S_{k+1} \bar{z} = \tilde{z} \iff S_{k+1} \bar{z} = S_k \tilde{z}. \quad \text{(B.2)}$$

Assume $S_k^T S_{k+1} \bar{z} = \tilde{z}$. Then, we have

$$S_k^T S_{k+1} \bar{z} = S_k \tilde{z} \quad \text{(B.3)}$$

Alternatively, the equivalence between (B.4) and (B.5) is verified by the well-known Pythagorean theorem. From (B.3) and (B.6), we obtain $S_{k+1} \bar{z} = S_k \tilde{z}$. The converse is obvious, which verifies (B.2).

By Proposition 1b, any element $z \in \text{Fix}(\Phi_k)$ can be expressed as $z = S_{k+1} \bar{z}$, $\exists \bar{z} \in \mathbb{R}^D$. Then, we have

$$S_{k+1} \bar{z} \in \text{Fix}(\Phi_k) \iff S_{k+1} S_{k+1} \bar{z} = S_{k+1} \bar{z} \iff S_{k+1} \bar{z} = \bar{z} \iff \bar{z} \in \text{Fix}(S_{k+1}^T S_k). \quad \text{(B.7)}$$

which with (26) verifies (25).

**Proof of Proposition 7d:**

The orthonormality of $S_k$ and $S_{k+1}$ imply that $\Phi_k = P_{\mathcal{R}(S_k)}$ [?]. Moreover, due to the basic property of projection, we obtain Fix ($\Phi_k$) = Fix ($P_{\mathcal{R}(S_k)}$).

Finally, thanks to Proposition 1 we can show that $\Phi_k$ is attracting nonexpansive if and only if $S_k = S_{k+1}$, as described below.

**Lemma B.2:** (On attracting nonexpansivity of $\Phi_k$):

(a) If $S_k = S_{k+1}$, then $\Phi_k$ is the projection matrix thus 1-attracting nonexpansive.

(b) If $S_k \neq S_{k+1}$, then $\Phi_k$ is nonexpansive but not attracting nonexpansive.

**Proof of Lemma B.2a:** By Proposition 1d, $S_k = S_{k+1} \Rightarrow \Phi_k = P_{\mathcal{R}(S_k)} \Rightarrow \Phi_k = P_{\mathcal{R}(S_k)}$.

Hence, by the Pythagorean theorem, we have

$$\|x - \Phi_k x\|^2 = \|x - f\|^2 - \|\Phi_k x - f\|^2, \quad \forall x \in \mathbb{R}^N, \forall f \in \text{Fix}(\Phi_k). \quad \text{(B.8)}$$

This means that the mapping $\Phi_k$ is 1-attracting nonexpansive.

**Proof of Lemma B.2b:** By $S_k \neq S_{k+1}$, there exists $\tilde{z} \in \mathbb{R}^D$ s.t. $S_{k+1} \tilde{z} \neq S_k \tilde{z}$.

For such a $\tilde{z}$, it holds that $\Phi_k S_k \tilde{z} = S_{k+1} S_k \tilde{z} \neq S_{k+1} \tilde{z}$, implying $S_{k+1} \tilde{z} \notin \text{Fix}(\Phi_k)$. Hence, we obtain

$$\|\Phi_k \tilde{z} - 0\| = \|S_{k+1} \tilde{z}\| = \|S_k \tilde{z}\| = \|\tilde{z}\| \quad \text{(B.9)}$$

This verifies that $\Phi_k$ is not attracting nonexpansive.

APPENDIX C

**Proof of Theorem 1**

**Proof of (a)-(I):** If $\Theta_k'(h_k) = 0$, then, $\forall h_k^* \in \Omega_k$, $\|h_{k+1} - h_k^*\|^2 = \|\Phi_k h_k - \Phi_k h_k^*\|^2 \leq \|h_k - h_k^*\|^2. \quad \text{(C.1)}$

Assume now $\Theta_k'(h_k) \neq 0$. In this case, we have

$$\|h_{k+1} - h_k^*\|^2 = \Phi_k \left[ h_k - \lambda \frac{\Theta_k(h_k)}{||\Theta_k'(h_k)||} \Theta_k'(h_k) \right] - \Phi_k h_k^*$$

$$\leq \left[ h_k - h_k^* - \lambda \frac{1}{||\Theta_k'(h_k)||} \Theta_k'(h_k) \right] \Theta_k'(h_k)$$

$$= \left[ h_k - h_k^* \right] - \lambda \frac{1}{||\Theta_k'(h_k)||} \Theta_k'(h_k)$$

$$+ \lambda \frac{1}{||\Theta_k'(h_k)||} \left[ \Theta_k'(h_k) \right]$$

$$\leq \left[ h_k - h_k^* \right] - \lambda \frac{1}{||\Theta_k'(h_k)||} \left[ \Theta_k'(h_k) \right] + \lambda \frac{1}{||\Theta_k'(h_k)||} \left[ \Theta_k'(h_k) \right]$$

$$= \left[ h_k - h_k^* \right] - \lambda \frac{1}{||\Theta_k'(h_k)||} \Theta_k'(h_k)$$

$$\leq \left[ h_k - h_k^* \right] + \lambda \frac{1}{||\Theta_k'(h_k)||} \Theta_k'(h_k)$$

which verifies (16). Here, the first and second inequalities are verified by the nonexpansivity of $\Phi_k$ and the definition of subgradient (see Lemma B.1 and Appendix A), respectively.
Proof of (a)-(II): Noting that $\Theta_k(h_k) > \inf_{x \in \mathbb{R}^N} \Theta_k(x)$ implies $\Theta_k'(h_k) \neq 0$, we can readily verify (29) by (C.2).

Proof of (b): From Theorem 11.1, we see that the nonnegative sequence $(\|h_k - \omega\|)_{k \geq K_0}$ for any $\omega \in \Omega$ is convergent, hence $(h_k)_{k \in \mathbb{N}}$ is bounded. Moreover, since $0 \in \partial\Theta_k(h_k)$ implies $\Theta_k(h_k) = 0$, it is sufficient to check the case $\Theta_k'(h_k) \neq 0$. In this case, by (C.2), we have

$$\|h_k - \omega\|^2 - \|h_{k+1} - \omega\|^2 \geq \varepsilon \|h_{k+1} - h_k\|^2 \geq 0. \quad (C.3)$$

Therefore, the convergence of $(\|h_k - \omega\|)_{k \geq K_0}$ implies

$$\lim_{k \to \infty} \Theta_k'(h_k) = 0, \quad (C.4)$$

hence the boundedness of $\Theta_k'(h_k)_{k \geq N}$ ensures

$$\lim_{k \to \infty} \Theta_k'(h_k) \neq 0.$$

\[\square\]

REFERENCES

[1] D. W. Tufts, R. Kumaresan, and I. Kistrens, Data adaptive signal estimation by singular value decomposition of a data matrix, Proc. IEEE, vol. 70, pp. 684865, Jun. 1982.

[2] W. F. Gabriel, Using spectral estimation techniques in adaptive processing antenna systems, IEEE Trans. Antennas Propagat., vol. AP-34, pp. 291300, Mar. 1986.

[3] L. L. Scharf and D. W. Tufts, Rank reduction for modeling stationary signals, IEEE Trans. Acoustics, Speech and Signal Processing, vol. ASSP-35, no. 3, pp. 350355, Mar. 1987.

[4] B. D. Van Veen and R. A. Roberts, Partially adaptive beamformer design via output power minimization, IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, pp. 15241532, Nov. 1987.

[5] D. G. Stro and L. S. Miller, Properties of the single-bit single-user MMSE receiver for DS-CDMA system, IEEE Trans. Commun., vol. 47, pp. 416425, Mar. 1999.

[6] X. Wang and H. V. Poor, “Blind multiuser detection: A subspace approach,” IEEE Trans. Inform. Theory, vol. 44, no. 2, pp. 677690, Mar. 1998.

[7] D. G. Stro and L. S. Miller, “Properties of the single-bit single-user MMSE receiver for DS-CDMA system,” IEEE Trans. Commun., vol. 47, pp. 416425, Mar. 1999.

[8] Y. Song and S. Roy, “Blind adaptive reduced-rank detection for DS/CDMA in multipath channels,” IEEE J. Selected Areas in Commun., vol. 17, no. 11, pp. 19061970, Nov. 1999.

[9] R. C. de Lamare and R. Sampaio-Neto, “Adaptive reduced-rank MMSE filtering with interpolated FIR filters and adaptive interpolators,” IEEE Signal Processing Lett., vol. 12, no. 3, pp. 177180, Mar. 2005.

[10] R. C. de Lamare and R. Sampaio-Neto, “Adaptive reduced-rank MMSE parameter estimation based on an Adaptive Diversity Combined Decimation and Interpolation Scheme,” Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, 2007, vol. 3, pp. III-1317-III-1320.

[11] R. C. de Lamare, R. Sampaio-Neto, “Space-time adaptive reduced-rank processor for interference mitigation in DS-CDMA systems,” IET communications, vol. 2, no. 2, pp. 388-397, 2008.

[12] 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, pp. 2503 2514.

[13] R.C. de Lamare, R. Sampaio-Neto and M. Haardt, “Blind Adaptive Constrained-Constant-Modulus Reduced-Rank Interference Suppression Algorithms Based on Interpolation and Switched Decimation,” IEEE Trans. on Signal Processing, vol.59, no.2, pp. 681-695, Feb. 2011.
[40] R. C. de Lamare and P. S. R. Diniz, “Set-Membership Adaptive Algorithms based on Time-Varying Error Bounds for CDMA Interference Suppression”, IEEE Trans. on Vehicular Technology, vol. 58, no. 2, February 2009, pp. 644 - 654.

[41] P. Clarke and R. C. de Lamare, "Low-Complexity Reduced-Rank Linear Interference Suppression based on Set-Membership Joint Iterative Optimization for DS-CDMA Systems”, IEEE Trans. on Vehicular Technology, vol. 60, no. 9, November 2011, pp. 4324-4337.

[42] T. Wang, R. C. de Lamare and P. D. Mitchell, “Low-complexity set-membership channel estimation for cooperative wireless sensor networks”, IEEE Transactions on Vehicular Technology, vol. 60, no. 6, pp. 2594-2607, 2011.

[43] M. Yukawa and I. Yamada, “Pairwise optimal weight realization Acceleration technique for set-theoretic adaptive parallel subgradient projection algorithm,” IEEE Trans. Signal Processing, vol. 54, no. 12, pp. 4557-4571, December 2006.

[44] M. Yukawa, K. Slavakis, and I. Yamada, “Adaptive parallel quadratic-metric projection algorithms,” IEEE Trans. Audio, Speech and Language Processing, vol. 15, no. 5, pp. 1665-1680, July 2007.

[45] S. Haykin, Adaptive Filter Theory, Prentice Hall, New Jersey, 4th edition, 2002.

[46] G. H. Golub and C. F. Van Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

[47] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.

[48] O. Axelsson, “A survey of preconditioned iterative methods for linear systems of algebraic equations,” BIT, vol. 25, pp. 166-187, 1985.

[49] Y. Saad, Iterative methods for sparse linear systems, SIAM, Philadelphia, PA, 2nd edition, 2003.

[50] A. W. Hull and W. K. Jenkins, “Preconditioned conjugate gradient methods for adaptive filtering,” in Proc. IEEE Int. Symp. Circuits Syst., Jun. 1991, pp. 5408-5411.

[51] G. K. Boray and M. D. Srinath, “Conjugate gradient techniques for adaptive filtering,” IEEE Trans. Circuits Syst. I, vol. 39, no. 1, pp. 110-117, Jan. 1992.

[52] P. S. Chang and A. N. Willson, Jr., “Analysis of conjugate gradient algorithms for adaptive filtering,” IEEE Trans. Signal Processing, vol. 48, no. 2, pp. 409-418, Feb. 2000.

[53] T. Hinamoto and S. Maekawa, “Extended theory of learning identification,” Trans. IEE Japan, vol. 95, no. 10, pp. 2272-234, 1975, in Japanese.

[54] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” IEICE Trans., vol. E76-A, no. 5, pp. 126-132, 1984, in Japanese.

[55] P. L. Combettes, “The foundations of set theoretic estimation,” Proc. IEEE, vol. 81, no. 2, pp. 182-208, Feb. 1993.

[56] H. H. Bauschke and J. M. Borwein, “On projection algorithms for solving convex feasibility problems,” SIAM Review, vol. 38, no. 3, pp. 367-426, 1996.

[57] Y. Censor and S. A. Zenios, Parallel Optimization: Theory, Algorithm, and Optimization, Oxford University Press, 1997.

[58] D. Butnariu, Y. Censor, and S. Reich, Eds., Inherently parallel algorithms in feasibility and optimization and their applications, New York: Elsevier, 2001.

[59] U. Madhow and M. L. Honig, “MMSE interference suppression for direct-sequence spread-spectrum CDMA,” IEEE Trans. Commun., vol. 42, no. 12, pp. 3178-3188, Dec. 1994.

[60] S. Boyd and L. Vandenberghe, Convex optimization, Cambridge University Press, Cambridge, 2004.

[61] G. Strang, Linear algebra and its applications, Saunders College Publishing, 3rd edition, 1988.