Adaptive Reduced-Rank Interference Suppression Based on the Multistage Wiener Filter

Michael L. Honig, *Fellow, IEEE,* and J. Scott Goldstein, *Fellow, IEEE*

Abstract—A class of adaptive reduced-rank interference suppression algorithms is presented based on the multi-stage Wiener filter (MSWF). The performance is examined in the context of direct-sequence (DS) code division multiple access (CDMA). Unlike the Principal Components method for reduced-rank filtering, the algorithms presented can achieve near full-rank performance with a filter rank much less than the dimension of the signal subspace. We present batch and recursive algorithms for estimating the filter parameters, which do not require an eigen-decomposition. Algorithm performance in a heavily loaded DS-CDMA system is characterized via computer simulation. Results show that the reduced-rank algorithms require significantly fewer training samples than other reduced- and full-rank algorithms.

Index Terms—Adaptive filters, code-division multiple access (CDMA), interference suppression.

I. INTRODUCTION

REDUCED-RANK linear filtering has been proposed for array processing and radar applications to enable accurate estimation of filter coefficients with a relatively small amount of observed data (e.g., see [1], [2] and the references therein). Other applications of reduced-rank filtering include equalization [3] and interference suppression in direct-sequence (DS) code-division multiple access (CDMA) communications systems [4]–[8]. In this paper we present reduced-rank adaptive filtering algorithms which are based on the multi-stage Wiener filter (MSWF) [9], [10]. Algorithm performance is studied in the context of DS-CDMA.

Reduced-rank interference suppression for DS-CDMA was originally motivated by situations where the processing gain N is much larger than the dimension of the signal subspace (e.g., [4] and [5]). This is relevant for some applications where a large processing gain is desired for covertness. If an N-tap adaptive filter is used to suppress interference (e.g., see [6]), then large N implies slow response to changing interference and channel conditions.

Much of the work on reduced-rank interference suppression for DS-CDMA has been based on "principal components (PC)" in which the received vector is projected onto an estimate of the lower dimensional signal subspace with largest energy (e.g., [4], [7]). This technique can improve convergence and tracking performance when N is much larger than the signal subspace. This assumption, however, does not hold for a heavily loaded commercial cellular system. Furthermore, in that application N can still be relatively large (i.e., >100).

Two reduced-rank methods that do not require the dimension of the projected subspace to be greater than that of the signal subspace are the "cross-spectral (CS)" method, presented in [11] (see also [12]), and the MSWF, presented in [10]. Unlike the CS and PC methods, the MSWF does not rely on an explicit estimate of the signal subspace, but rather generates a set of basis vectors by means of a successive refinement procedure [10]. (See also [8], [13]. An "Auxiliary Vector" filter is presented, which generates the same subspace as the reduced-rank MSWF.) This technique can attain near full-rank minimum mean squared error (MMSE) performance with a filter rank which is much smaller than the dimension of the signal subspace [14]. As will be demonstrated, this low rank enables a substantial reduction in the number of training samples needed to obtain an accurate estimate of the filter parameters.

We present a class of adaptive filtering algorithms, which are motivated by the MSWF. These algorithms do not require an eigen-decomposition, and are relatively simple (especially for small filter rank). Both batch and recursive algorithms are presented in this paper, along with training-based, or decision-directed, and blind versions of each. The blind algorithms require knowledge of the desired user's spreading code and associated timing (i.e., see [6]). We will also assume that timing information is available for the training-based algorithms. The performance of the adaptive MSWF techniques are illustrated numerically, and are compared with other adaptive reduced-rank techniques.

The next section presents the DS-CDMA model, Sections III and IV review reduced-rank MMSE filtering and the MSWF, and Section V presents the adaptive MSWF algorithms. Numerical results are presented in Section VI, and adaptive rank selection is discussed in Section VII.

II. CDMA SYSTEM MODEL

An asynchronous CDMA system model is considered in which the kth user, $1 \leq k \leq K$, transmits a baseband signal

$$x_k(t) = \sum_i A_k b_k(i) p_k(t - iT - \tau_k) \tag{1}$$

where $b_k(i)$ is the ith symbol transmitted by user k, $p_k(t)$ is the spreading waveform associated with user k, and τ_k and A_k are,

M. L. Honig is with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: mh@ece.nwu.edu).

J. S. Goldstein is with SAIC, Chantilly, VA 20151-3707 USA (e-mail: sgoldstein@trg1.saic.com).

respectively, the delay and amplitude associated with user k. We assume binary signaling, so that $b_k(i) \in \{\pm 1\}$. For DS-CDMA,

$$p_k(t) = \sum_{i=1}^{N-1} a_k[i]\Psi(t - iT_c) \tag{2}$$

where $a_k[i] \in \{\pm 1/\sqrt{N}\}$, $i = 0, \ldots, N-1$, is the real-valued spreading sequence, $\Psi(t)$ is the chip waveform, normalized to have unit energy, T_c is the chip duration, and $N = T/T_c$ is the processing gain. It is assumed that the same spreading code is repeated for each symbol. The numerical results in Section VI assume rectangular chip shapes.

Let $\mathbf{y}(i)$ be the N-vector containing samples at the output of a chip-matched filter during the ith transmitted symbol, assuming that the receiver is synchronized to the desired user. Letting $k = 1$ correspond to the user to be detected, we can write

$$\mathbf{y}(i) = A_1 b_1(i)\mathbf{p}_1 + \sum_{k=2}^{K} A_k[b_k(i)\mathbf{p}_k^+ + b_k(i-1)\mathbf{p}_k^-] + \mathbf{n}(i) \tag{3}$$

where $\mathbf{p}_1$ is the spreading sequence associated with the desired user, $\mathbf{p}_k^-$ and $\mathbf{p}_k^+$ are the two N-vectors associated with the kth interferer due to asynchronous transmission, and $\mathbf{n}(i)$ is the vector of noise samples at time i, assumed to be white with covariance $\sigma^2\mathbf{I}$. In what follows, we will use the more convenient notation

$$\mathbf{y}(i) = \mathbf{P}^-\mathbf{A}\mathbf{b}(i-1) + \mathbf{P}^+\mathbf{A}\mathbf{b}(i) + \mathbf{n}(i) \tag{4}$$

where $\mathbf{P}^\pm$ is the $N \times K$ matrix with columns given by the corresponding signal vectors, $\mathbf{b}(i)$ is the vector of transmitted symbols across users, and $\mathbf{A}$ is the diagonal matrix of amplitudes. (Since the receiver is synchronized to the desired user, if $\mathbf{P}^+$ contains $\mathbf{p}_1$, then the column of $\mathbf{P}^-$ corresponding to user 1 contains all zeros.)

III. REDUCED-RANK LINEAR MMSE FILTERING

The MMSE receiver consists of the vector $\mathbf{c}$, which is chosen to minimize the MSE

$$\mathcal{M} = E\{|b_1(i) - \mathbf{c}^\dagger\mathbf{y}(i)|^2\} \tag{5}$$

where $\dagger$ represents Hermitian transpose. For simplicity, we assume that $\mathbf{c}$ contains N coefficients and spans a single symbol interval, which is suboptimal for asynchronous DS-CDMA [6]. The following discussion is easily generalized to the case where the vector $\mathbf{c}$ spans multiple symbol intervals.

The vector $\mathbf{c}$ can be estimated from received data via standard stochastic gradient or least squares estimation techniques [6]. However, large N implies slow convergence. A reduced-rank algorithm reduces the number of adaptive coefficients by projecting the received vectors onto a lower dimensional subspace. Specifically, let $\mathbf{S}_D$ be the $N \times D$ matrix with column vectors which are an orthonormal basis for a D-dimensional subspace, where $D < N$. The projected received vector corresponding to symbol i is then given by

$$\tilde{\mathbf{y}}(i) = \mathbf{S}_D^\dagger\mathbf{y}(i) \tag{6}$$

where, in what follows, all D-dimensional quantities are denoted with a "tilde."

The sequence of projected received vectors $\{\tilde{\mathbf{y}}(i)\}$ is the input to a tapped-delay line filter, represented by the D-vector $\tilde{\mathbf{c}}(i)$ for symbol i. The filter output corresponding to the ith transmitted symbol is

$$z(i) = \tilde{\mathbf{c}}^\dagger(i)\tilde{\mathbf{y}}(i). \tag{7}$$

Assuming coherent detection, the vector $\tilde{\mathbf{c}}(i)$ which minimizes the mean squared error (MSE) $E(|e(i)|^2)$, where $e(i) = b_1(i) - \tilde{\mathbf{c}}^\dagger(i)\tilde{\mathbf{y}}(i)$, is

$$\tilde{\mathbf{c}}_{mmse} = \tilde{\mathbf{R}}^{-1}\tilde{\mathbf{p}}_1 \tag{8}$$

where

$$\tilde{\mathbf{R}} = E[\tilde{\mathbf{y}}(i)\tilde{\mathbf{y}}^\dagger(i)] = \mathbf{S}_D^\dagger\mathbf{R}\mathbf{S}_D \tag{9}$$

$$\begin{aligned}\mathbf{R} &= E[\mathbf{y}(i)\mathbf{y}^\dagger(i)] \\ &= \mathbf{P}^-\mathbf{A}^2\mathbf{P}^{-\dagger} + \mathbf{P}^+\mathbf{A}^2\mathbf{P}^{+\dagger} + \sigma^2\mathbf{I}\end{aligned} \tag{10}$$

$$\tilde{\mathbf{p}}_1 = \mathbf{S}_D^\dagger E[b_1^*(i)\mathbf{y}(i)] = \mathbf{S}_D^\dagger\mathbf{p}_1. \tag{11}$$

The associated MMSE for a rank D filter is given by

$$\mathcal{M}_D = 1 - \tilde{\mathbf{p}}_1^\dagger\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{p}}_1. \tag{12}$$

Before presenting the MSWF, we briefly mention other reduced-rank filters, which have been previously proposed. The performance of the adaptive MSWF algorithms to be described will be compared with the performance of these other methods in Section VI. A simulation study of the adaptive eigen-decomposition and partial despreading methods is presented in [5].

A. Eigen-Decomposition Techniques

PC reduced-rank filtering is based on the eigen-decomposition

$$\mathbf{R} = \mathbf{V}\Lambda\mathbf{V}^\dagger \tag{13}$$

where $\mathbf{V}$ is the orthonormal matrix whose columns are eigenvectors of $\mathbf{R}$, and Λ is the diagonal matrix of eigenvalues. If we assume the eigenvalues are ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$, then for given subspace dimension D, the projection matrix for PC is $\mathbf{S}_D = \mathbf{V}_{1:D}$, the first D columns of $\mathbf{V}$. This technique can allow a significant reduction in rank when the dimension of the signal subspace is much less than N. If this is not the case, then projecting onto the subspace $\mathbf{S}_D$ for small D is likely to reduce the desired signal component. This is especially troublesome in a near-far environment where the energy associated with the interference subspace is greater than that for the desired user.

If the spreading code for the desired user $\mathbf{p}_1$ is known, then combining the PC method with the Generalized Sidelobe Canceler (GSC) structure [15], [16] maintains the desired signal energy. Specifically, the filter $\mathbf{c}$ can be expressed as $\mathbf{c} = \mathbf{p}_1 - \mathbf{B}\mathbf{w}$, where $\mathbf{B}$ is a *blocking matrix*, and satisfies $\mathbf{B}^\dagger\mathbf{p}_1 = 0$. Selecting $\mathbf{w}$ to minimize the output MSE gives $\mathbf{w} = \check{\mathbf{R}}^{-1}\check{\mathbf{p}}_1$, where $\check{\mathbf{R}} = \mathbf{B}^\dagger\mathbf{R}\mathbf{B}$, and $\check{\mathbf{p}}_1 = \mathbf{B}^\dagger\mathbf{R}\mathbf{p}_1$. A reduced-rank GSC is then obtained by projecting the output of $\mathbf{B}$ onto a smaller subspace, spanned by the columns of $\mathbf{S}_D$. A rank-D approximation for $\mathbf{w}$ is given by $\mathbf{S}_D\tilde{\mathbf{w}}$, where $\tilde{\mathbf{w}} = (\mathbf{S}_D^\dagger\check{\mathbf{R}}\mathbf{S}_D)^{-1}(\mathbf{S}_D^\dagger\check{\mathbf{p}}_1)$. For the PC method, the columns of $\mathbf{S}_D$ are the eigenvectors of $\check{\mathbf{R}}$ corresponding to the D largest eigenvalues [17]–[19].
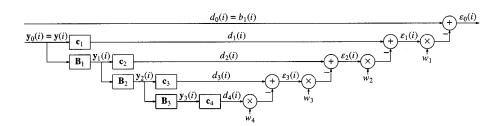
Fig. 1.   Multistage Wiener filter.

An alternative to PC is to choose the set of D eigenvectors for the projection matrix which minimizes the MSE. Specifically, if $\mathbf{S}_D$ consists of D eigenvectors of $\mathbf{R}$, then the MSE can be written in terms of projected variables as

$$\mathcal{M}_D = 1 - \|\Lambda_D^{-1/2}\tilde{\mathbf{p}}_1\|^2 \qquad (14)$$

where Λ_D is the $D \times D$ diagonal matrix of associated eigenvalues. To minimize $\mathcal{M}_D$, the basis vectors should be the eigenvectors of $\mathbf{R}$ associated with the D largest values of $\|\tilde{\mathbf{p}}_{1,k}\|^2/\lambda_k$, where $\tilde{\mathbf{p}}_{1,k} = \mathbf{V}_k^\dagger \mathbf{p}_1$ is the kth component of $\tilde{\mathbf{p}}_1$, and $\mathbf{V}_k$ is the kth column of $\mathbf{V}$. (Note the inverse weighting of λ_k in contrast with PC.)

This technique, called "cross-spectral (CS)" reduced-rank filtering, was presented in [11]. Prior to that work, a similar CS metric for ordering the eigenvalues in a GSC was presented in [12]. The CS reduced-rank filter can perform well for $D < K$ without the GSC structure since it takes into account the energy in the subspace contributed by the desired user. Unlike PC, the estimated subspace for CS requires either training, or *a priori* knowledge of the desired user's spreading code $\mathbf{p}_1$. A disadvantage of eigen-decomposition techniques in general is the complexity associated with estimation of the signal subspace.

B. Partial Despreading

In this method, proposed in [20], the received DS-CDMA signal is *partially despread* over consecutive segments of m chips, where m is a parameter. The partially despread vector has dimension $D = \lceil N/m \rceil$, and is the input to the D-tap filter. Consequently, $m = 1$ corresponds to the full-rank MMSE filter, and $m = N$ corresponds to the matched filter. The columns of $\mathbf{S}_D$ in this case are nonoverlapping segments of $\mathbf{p}_1$, where each segment is of length m.

Specifically, if $N/m = D$, the jth column of $\mathbf{S}_D$ is

$$[\mathbf{S}_D]_j' = [0 \cdots 0 \ \mathbf{p}_{1,\,[(j-1)m+1:jm]}' \ 0 \cdots 0] \qquad (15)$$

where $1 \leq j \leq D$, the subscript $(l:m)$ denotes components l through m of the corresponding vector, and there are $(j-1)m$ zeros on the left and $(D-j)m$ zeros on the right. This is a simple reduced-rank technique that allows the selection of MMSE performance between the matched and full-rank MMSE filters by adjusting the number of adaptive filter coefficients.

IV. THE MULTISTAGE WIENER FILTER (MSWF)

The MSWF was presented in [10] for the known statistics case, i.e., known covariance matrix $\mathbf{R}$ and steering vector $\tilde{\mathbf{p}}_1$. A block diagram of a four-stage MSWF is shown in Fig. 1. The stages are associated with the sequence of nested filters

$\mathbf{c}_1, \ldots, \mathbf{c}_D$, where D is the order of the filter. If $D = N$, then the filter is the full-rank MMSE (Wiener) filter. Let $\mathbf{B}_m$ denote a *blocking matrix*, i.e.,

$$\mathbf{B}_m^\dagger \mathbf{c}_m = \mathbf{0}. \qquad (16)$$

In what follows, we will sometimes write $\mathbf{B}_m = \text{null}(\mathbf{c}_m)$, which is $M \times (M-1)$, and other times write $\mathbf{B}_m = \mathbf{I} - \mathbf{c}_m \mathbf{c}_m^\dagger$, which is $M \times M$, but has rank $M - 1$.

Referring to Fig. 1, let $d_m(i)$ denote the output of the filter $\mathbf{c}_m$, and $\mathbf{y}_m(i)$ denote the output of the blocking matrix $\mathbf{B}_m$, both at time i. The $(m+1)$st multi-stage filter is determined by

$$\mathbf{c}_{m+1} = E[d_m^* \mathbf{y}_m]. \qquad (17)$$

For $m = 0$, we have $d_0(i) = b_1(i)$ (the desired input symbol), $\mathbf{y}_0(i) = \mathbf{y}(i)$, and $\mathbf{c}_1$ is the matched filter $\mathbf{p}_1$. As in [10], it will be convenient to normalize the filters $\mathbf{c}_1, \ldots, \mathbf{c}_D$ so that $\|\mathbf{c}_m\| = 1$.

The filter output is obtained by linearly combining the outputs of the filters $\mathbf{c}_1, \ldots, \mathbf{c}_D$ via the weights $w_1, \ldots, w_{D-1}$. This is accomplished stage-by-stage. Referring to Fig. 1, let

$$\epsilon_m(i) = d_m(i) - w_{m+1}\epsilon_{m+1}(i) \qquad (18)$$

for $1 \leq m \leq D$ and $\epsilon_D(i) = d_D(i)$. Then w_{m+1} is selected to minimize $E[|\epsilon_m|^2]$.

The rank-D MSWF is given by the following set of recursions.

Initialization:

$$d_0(i) = b_1(i), \qquad \mathbf{y}_0(i) = \mathbf{y}(i). \qquad (19)$$

For $n = 1, \ldots, D$ *(Forward Recursion):*

$$\mathbf{c}_n = E[d_{n-1}^* \mathbf{y}_{n-1}(i)]/\|E[d_{n-1}^* \mathbf{y}_{n-1}]\| \qquad (20)$$

$$d_n(i) = \mathbf{c}_n^\dagger \mathbf{y}_{n-1}(i) \qquad (21)$$

$$\mathbf{B}_n = \text{null}(\mathbf{c}_n) \qquad (\text{if } n < D) \qquad (22)$$

$$\mathbf{y}_n = \mathbf{B}_n^\dagger \mathbf{y}_{n-1} \qquad (\text{if } n < D). \qquad (23)$$

Decrement $n = D, \ldots, 1$ *(Backward Recursion):*

$$w_n = E[d_{n-1}^*(i)\epsilon_n(i)]/E[|\epsilon_n(i)|^2] \qquad (24)$$

$$\epsilon_{n-1}(i) = d_{n-1}(i) - w_n^*\epsilon_n(i) \qquad (25)$$

where $\epsilon_D(i) = d_D(i)$. The estimate of d_0 is $w_1^*\epsilon_1$.

At stage n the filter generates a desired sequence $\{d_n(i)\}$ and an "observation" sequence $\{\mathbf{y}_n(i)\}$. Replacing $\mathbf{c}_n$ in the MSWF by the MMSE filter $\mathbf{c}_n^{(mmse)}$ for estimating $\{d_{n-1}(i)\}$ from $\{\mathbf{y}_{n-1}(i)\}$ gives the full-rank MMSE filter. The MSWF is "self-similar" in the sense that the MMSE filter $\mathbf{c}_n^{(mmse)}$ is

replaced by the associated MSWF. The covariance matrix for the projected vector $\tilde{\mathbf{y}} = [d_0, d_1, \ldots, d_{D-1}]$ is *tri-diagonal* [10].

It is shown in [14] that MSWF has the following properties.

1) Let $\mathcal{S}_D$ denote the D-dimensional subspace associated with the rank D MSWF. Then

$$\mathcal{S}_D = \text{span}\{\mathbf{c}_1, \ldots, \mathbf{c}_D\} \tag{26}$$

$$= \text{span}\{\mathbf{p}_1, \mathbf{R}\mathbf{p}_1, \mathbf{R}^2\mathbf{p}_1, \ldots, \mathbf{R}^{D-1}\mathbf{p}_1\} \tag{27}$$

where the first set of basis vectors is an orthonormal set, and the basis vectors in the second set are not orthogonal. That is, a reduced-rank MSWF projects the received signal onto $\mathcal{S}_D$, the Krylov subspace defined by (27), and optimizes the filter within that subspace.

2) The rank D needed to achieve full-rank performance does not scale with system size (K and N). This is shown by computing the large system output SINR for the reduced-rank MSWF, defined by letting $K \to \infty$ and $N \to \infty$ with fixed K/N. For the ideal synchronous CDMA model, as D increases, this large system output SINR converges to the full-rank large system SINR as a continued fraction. As a consequence, full-rank performance is essentially achieved with rank $D \leq 8$ for a wide range of loads and signal-to-noise ratios (SNRs).

We remark that, as D increases, the set of basis vectors in (27) can become nearly linearly dependent even for $D \ll K$. In that case, the transformed covariance matrix $\mathbf{S}_D^\dagger \mathbf{R} \mathbf{S}_D$ becomes ill-conditioned, which creates numerical problems with computing the reduced-rank filter $\tilde{\mathbf{c}}$. This indicates that fewer than D basis vectors essentially span the projection subspace, and that D can be decreased without significantly increasing the MSE. This observation is used in Section VII to formulate an adaptive rank selection method.

V. ADAPTIVE REDUCED-RANK ALGORITHMS

In this section, we present a family of adaptive algorithms which are related to the MSWF. A straightforward way to derive such an adaptive algorithm is to replace statistical averages by sample averages. This has the geometric interpretation of changing the metric space in which variables are defined [21]. Namely, for the known statistics case, we define the inner product between two random variables X and Y as $\langle X, Y \rangle = E[X^*Y]$, which leads to an MMSE cost criterion (minimize $\|b_1 - \mathbf{c}^\dagger \mathbf{y}\|^2$ for random b_1 and $\mathbf{y}$).

For the given data case, inner product between two vectors is defined in the standard way. Given a sequence of M received vectors and M training (or estimated) symbols

$$\mathbf{Y} = [\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(M)] \tag{28}$$

$$\overline{\mathbf{b}} = [b_1(1), \ldots, b_1(M)] \tag{29}$$

the ($M \times 1$) vector of errors is defined as

$$\mathbf{e} = \overline{\mathbf{b}} - \mathbf{c}^\dagger \mathbf{Y} \tag{30}$$

Initialization:
$$\mathbf{d}_0 = \overline{\mathbf{b}} \qquad \mathbf{Y}_0 = \mathbf{Y} \tag{35}$$

For $n = 1, \ldots, D$ (Forward Recursion):

$$\hat{\mathbf{p}}_n = \mathbf{Y}_{n-1}\mathbf{d}_{n-1}^\dagger \quad (N - n + 1) \times 1 \tag{36}$$
$$\delta_n = \|\hat{\mathbf{p}}_n\| \tag{37}$$
$$\mathbf{c}_n = \hat{\mathbf{p}}_n/\delta_n \tag{38}$$
$$\mathbf{d}_n = \mathbf{c}_n^\dagger \mathbf{Y}_{n-1} \quad 1 \times M \tag{39}$$

If $n < N$,

$$\mathbf{B}_n = \text{null}(\mathbf{c}_n) \quad (N - n + 1) \times (N - n) \tag{40}$$
$$\mathbf{Y}_n = \mathbf{B}_n^\dagger \mathbf{Y}_{n-1} \quad (N - n) \times M \tag{41}$$

Decrement $n = D, \ldots, 1$ (Backward Recursion):

$$w_n = (\epsilon_n \mathbf{d}_{n-1}^\dagger)/\|\epsilon_n\|^2 = \delta_n/\|\epsilon_n\|^2 \tag{42}$$
$$\epsilon_{n-1} = \mathbf{d}_{n-1} - w_n^* \epsilon_n \tag{43}$$

where $\epsilon_D = \mathbf{d}_D$. The symbol estimate is $w_1^* \epsilon_1$.

Fig. 2. Algorithm 1. Batch adaptive MSWF with training.

and our objective is to minimize $\|\mathbf{e}\|^2$, which is the standard least squares (LS) cost function. For rank $D < N$, the cost function becomes

$$\mathcal{C}_D = \sum_{i=1}^{M} \|b_1(i) - \tilde{\mathbf{c}}^\dagger \tilde{\mathbf{y}}(i)\|^2 \tag{31}$$

where $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{y}}$ are the associated projected variables. Specifically

$$\tilde{\mathbf{y}}(i) = \hat{\mathbf{S}}_D^\dagger(i)\mathbf{y}(i) \tag{32}$$

where the columns of $\hat{\mathbf{S}}_D(i)$ are the estimated basis vectors for the subspace $\mathbf{S}_D$ at time i.

A. Batch Algorithms

Here we consider estimation of the MSWF parameters given $\mathbf{Y}$ and $\overline{\mathbf{b}}$ in (28) and (29). The approach just described leads to Algorithm 1 (see Fig. 2, the batch adaptive MSWF with training (35)–(43). Following the approach in [10], it is straightforward to show that this algorithm tri-diagonalizes the $(N+1) \times (N+1)$ extended sample covariance matrix

$$\hat{\overline{\mathbf{R}}} = \overline{\mathbf{Y}}\,\overline{\mathbf{Y}}^\dagger \tag{33}$$

where

$$\overline{\mathbf{Y}} = \begin{bmatrix} \overline{\mathbf{b}} \\ --- \\ \mathbf{Y} \end{bmatrix}. \tag{34}$$

In what follows, we assume that the rows of each blocking matrix $\mathbf{B}_n$, $n = 1, \ldots, D$, are orthonormal, so that the performance is independent of the specific choice of blocking matrix. In general, the performance does depend on the choice of blocking matrices when they are not constrained to be orthonormal [22]. Note that $\epsilon_0 = \min_{\tilde{\mathbf{c}}} \mathcal{C}_D$, the minimized LS cost function in (31). When used in decision-directed mode, the estimate of the block of transmitted symbols $\overline{\mathbf{b}}$ is $w_1^* \epsilon_1$, where ϵ_n is computed from (43).

Initialization:

$$\mathbf{T}_1 = \hat{\bar{\mathbf{R}}}(i) \qquad (44)$$

For $n = 1, \ldots, D$ (Forward Recursion):

$$\hat{\mathbf{p}}_n = [\mathbf{T}_n]_{n;(n+1:N+1)} \qquad (45)$$
$$\delta_n = \|\hat{\mathbf{p}}_n\| \qquad \mathbf{c}_n = \hat{\mathbf{p}}_n/\delta_n \qquad (46)$$
$$\mathbf{B}_n = \text{null}(\mathbf{c}_n) \qquad (47)$$

$$\mathbf{L}_n = \begin{bmatrix} \mathbf{I}_{n,n} & | & \mathbf{0}_{n,N+1-n} \\ & & \mathbf{c}_n^\dagger \\ \mathbf{0}_{N+1-n,n} & | & - - - - \\ & & \mathbf{B}_n^\dagger \end{bmatrix} \qquad (48)$$

$$\mathbf{T}_{n+1} = \mathbf{L}_n \mathbf{T}_n \mathbf{L}_n^\dagger \qquad (49)$$

Decrement $n = D, \ldots, 1$ (Backward Recursion):

$$w_n = \delta_n/\xi_n \qquad (50)$$
$$\xi_{n-1} = [\mathbf{T}_{D+1}]_{(n+1,n+1)} - w_n^* \delta_n \qquad (51)$$

where $\xi_n = \|\epsilon_n\|^2$ in (43), and $\xi_D = [\mathbf{T}_{D+1}]_{(D+1,D+1)}$.

Fig. 3. Algorithm 2. Adaptive MSWF based on tri-diagonalization of the sample covariance matrix.

A nontraining based, or blind version of the preceding algorithm can be obtained simply by substituting $\mathbf{p}_1$ (spreading code for the desired user) for $\hat{\mathbf{p}}_1$ in the preceding algorithm. The resulting set of forward recursions does not exactly tri-diagonalize the extended sample covariance matrix, and the associated output SINR tends to converge more slowly to the optimum value relative to a training-based algorithm. An illustrative example is given in Section VI.

An alternative set of computations to Algorithm 1 for estimating the MSWF parameters is Algorithm 2, given by (44)–(51) (see Fig. 3). Algorithm 2 tri-diagonalizes the extended sample covariance matrix $\hat{\bar{\mathbf{R}}}$ [10]. Specifically, let $\overline{\mathbf{D}} = [\mathbf{d}_0; \mathbf{d}_1; \mathbf{d}_2; \ldots; \mathbf{d}_D]$, where $\mathbf{d}_n$ is defined by (39), and ";" separates rows, so that $\overline{\mathbf{D}}$ is $(D+1) \times M$. Then Algorithm 2 computes the tri-diagonal matrix $\tilde{\mathbf{T}}_{D+1} = \overline{\mathbf{D}}\,\overline{\mathbf{D}}^\dagger$. Namely, $\tilde{\mathbf{T}}_{D+1}$ is the $(D+1) \times (D+1)$ matrix, which occupies the upper left corner of $\mathbf{T}_{D+1}$, computed in Algorithm 2. In Algorithm 2, $\mathbf{X}_{l;(k:m)}$ denotes the row vector containing the kth through mth components of the lth row of the matrix $\mathbf{X}$. The MSWF recursions (19), (21)–(23), and (25) are then used to compute the filter output.

We remark that the computational requirements of the preceding algorithm for small D are modest in comparison with reduced-rank techniques that require the computation of eigenvectors of the sample covariance matrix $\hat{\mathbf{R}} = \mathbf{Y}\mathbf{Y}^\dagger$.

B. Recursive Algorithms

A recursive update for the extended sample covariance matrix is given by

$$\hat{\bar{\mathbf{R}}}(i) = (1 - \mu) * \hat{\bar{\mathbf{R}}}(i-1) + \mu \bar{\mathbf{y}}(i)\bar{\mathbf{y}}^\dagger(i) \qquad (52)$$

where μ is a forgetting factor which discounts past data. The preceding algorithms can, in principle, be used to update the MSWF parameters at each i, although this would be computationally intensive. A somewhat simpler recursive algorithm,

Initialization:

$$d_0(i) = b_1(i) \qquad \mathbf{y}_0(i) = \mathbf{y}(i) \qquad (53)$$

At each i for $n = 1, \ldots, D$:

$$\hat{\mathbf{p}}_n(i) = (1 - \mu)\hat{\mathbf{p}}_n(i-1) + \mu d_{n-1}^*(i)\mathbf{y}_{n-1}(i) \qquad (54)$$
$$\delta_n(i) = \|\hat{\mathbf{p}}_n(i)\|, \qquad \mathbf{c}_n(i) = \hat{\mathbf{p}}_n(i)/\delta_n(i) \qquad (55)$$
$$d_n(i) = \mathbf{c}_n^\dagger(i)\mathbf{y}_{n-1}(i) \qquad (56)$$
$$\qquad (57)$$

If $n < N$,

$$\mathbf{B}_n(i) = \text{null}[\mathbf{c}_n^\dagger(i)] \qquad (58)$$
$$\mathbf{y}_n(i) = \mathbf{B}_n^\dagger(i)\mathbf{y}_{n-1}(i) \qquad (59)$$

Decrement $n = D, \ldots, 1$:

$$\xi_n(i) = (1 - \mu)\xi_n(i-1) + \mu|\epsilon_n(i)|^2 \qquad (60)$$
$$w_n(i) = \delta_n(i)/\xi_n(i) \qquad (61)$$
$$\epsilon_{n-1}(i) = d_{n-1}(i) - w_n^*(i)\epsilon_n(i) \qquad (62)$$

where $\epsilon_D(i) = d_D(i)$.
Filter output (estimate of $b_1(i)$) $= w_1^*(i)\epsilon_1(i)$

Fig. 4. Algorithm 3. Stochastic gradient (SG) MSWF.

which is equivalent to the adaptive MSWF, is based on computing powers of $\hat{\mathbf{R}}$, to be described in the next section.

Rather than perform an exact tri-diagonalization of the sample covariance matrix at each iteration, it is also possible to approximate the MSWF parameters via sample averages. This leads to Algorithm 3, given by (53)–(62) (see Fig. 4), the "Stochastic Gradient (SG)" MSWF algorithm. This algorithm is computationally simpler than recursive versions of Algorithms 1 and 2, but does not exactly tri-diagonalize the extended sample covariance matrix $\hat{\bar{\mathbf{R}}}(i)$ at each iteration. Consequently, Algorithm 3 does not perform as well as the "exact" Algorithms 1 and 2, as the results in Section VI illustrate.

C. Algorithms Based on Powers of $\hat{\mathbf{R}}$

An alternative set of adaptive algorithms can be derived based on the second representation for $\mathcal{S}_D$ given in (27). For the given data case with training, we replace the matrix of basis vectors $\mathbf{S}_D$ by

$$\hat{\mathbf{S}}_D(i) = [\hat{\mathbf{p}}_1(i), \hat{\mathbf{R}}(i)\hat{\mathbf{p}}_1(i), \hat{\mathbf{R}}^2(i)\hat{\mathbf{p}}_1(i), \ldots, \hat{\mathbf{R}}^{D-1}(i)\hat{\mathbf{p}}_1(i)] \qquad (63)$$

where

$$\hat{\mathbf{p}}_1(i) = (1 - \mu)\hat{\mathbf{p}}_1(i-1) + \mu d^*(i)\mathbf{y}(i) \qquad (64)$$

and $\hat{\mathbf{R}}(i)$ is updated according to (52). Let

$$\hat{\gamma}_m = \hat{\mathbf{p}}_1^\dagger \hat{\mathbf{R}}^m \hat{\mathbf{p}}_1 \qquad (65)$$
$$\hat{\boldsymbol{\gamma}}_{l:m} = [\hat{\gamma}_l \ \hat{\gamma}_{l+1} \cdots \hat{\gamma}_m]' \qquad (66)$$
$$\hat{\Gamma}_{l:l+m} = [\hat{\boldsymbol{\gamma}}_{l:l+m} \ \hat{\boldsymbol{\gamma}}_{l+1:l+m+1} \cdots \hat{\boldsymbol{\gamma}}_{l+m:l+2m}] \qquad (67)$$

$$\hat{\mathbf{R}} = \mathbf{Y}\mathbf{Y}^{\dagger}, \quad \hat{\mathbf{p}}_1 = \mathbf{Y}\bar{\mathbf{b}}^{\dagger} \quad (69)$$

For $m = 0, \ldots, D$, $\hat{\mathbf{v}}_m = \hat{\mathbf{R}}^m \hat{\mathbf{p}}_1$
For $m = 0, \ldots, 2D - 1$, $\hat{\gamma}_m = \hat{\mathbf{v}}_j^{\dagger} \hat{\mathbf{v}}_l$, $j + l = m$.

$$\tilde{\mathbf{c}}_D = \hat{\Gamma}_{1:D}^{-1} \hat{\gamma}_{0:D-1} \quad (70)$$

For $i = 1, \ldots, M$:

$$\tilde{\mathbf{y}}(i) = \hat{\mathbf{S}}_D^{\dagger} \mathbf{y}(i) \qquad \text{Symbol estimate} = \tilde{\mathbf{c}}^{\dagger} \tilde{\mathbf{y}}(i) \quad (71)$$

Fig. 5. Algorithm 4. Batch adaptive algorithm based on ppowers of the sample covariance matrix.

where the dependence on i is not shown for convenience. Note that $\hat{\Gamma}_{l:l+m}$ is an $(m + 1) \times (m + 1)$ matrix. Selecting $\tilde{\mathbf{c}}$ to minimize (31), where $\tilde{\mathbf{y}}$ is given by (32), gives

$$\tilde{\mathbf{c}}_D = (\hat{\mathbf{S}}_D^{\dagger} \hat{\mathbf{R}} \hat{\mathbf{S}}_D)^{-1} \hat{\mathbf{S}}_D^{\dagger} \hat{\mathbf{p}}_1 = \hat{\Gamma}_{1:D}^{-1} \hat{\gamma}_{0:D-1}. \quad (68)$$

Given $\mathbf{Y}$ and $\bar{\mathbf{b}}$ in (28) and (29), a reduced-rank batch algorithm with training is Algorithm 4 given by (69)–(71) (see Fig. 5). If $\mathbf{p}_1$ is known, then in the absence of training, $\hat{\mathbf{p}}_1(i)$ in (63) and (65) can be replaced by $\mathbf{p}_1$.

Following the same argument used to prove [14, Theorem 2], it can be shown that Algorithm 4 is equivalent to Algorithm 1 if the blocking matrix in (40) is replaced by $\mathbf{B}_n = \mathbf{I} - \mathbf{c}_n \mathbf{c}_n^{\dagger}$, and the dimensions of the other variables are adjusted accordingly. That is, both algorithms produce the same filter output. Of course, the preceding algorithm can be implemented recursively, where the variables $\hat{\mathbf{v}}_m$ and $\hat{\gamma}_m$ are recomputed for each i.
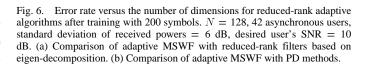
VI. NUMERICAL RESULTS

Fig. 6 shows plots of error rate versus the number of dimensions for reduced-rank adaptive algorithms after training with 200 symbols. Parameters for all numerical examples are $N = 128$, $K = 42$, and the received powers are log-normal with standard deviation 6 dB. The top graph shows results for the following algorithms: MSWF, CS, PC with the GSC structure (PC-GSC), and the matched filter (MF). For the adaptive CS method, $\mathbf{R}$ and $\mathbf{p}_1$ in (13) and (14) are replaced by $\hat{\mathbf{R}}$ and $\hat{\mathbf{p}}_1$, respectively. The simulated MSWF and CS filters require a training sequence, and do not require knowledge of $\mathbf{p}_1$. In contrast, the simulated PC-GSC does not require a training sequence, but is assumed to know $\mathbf{p}_1$.

The bottom graph in Fig. 6 shows results for three partial despreading (PD) methods, which correspond to the way the filter $\tilde{\mathbf{c}}$ is updated given the sequence of training symbols $\{b_1(i)\}$ and the projected (partially despread) vectors $\{\tilde{\mathbf{y}}(i)\}$. Stochastic Gradient with PD (SG-PD) indicates that the vector $\tilde{\mathbf{c}}$ is updated according to a normalized Stochastic Gradient algorithm. LS-PD and MMSE-PD correspond to LS and MMSE solutions for $\tilde{\mathbf{c}}$. The adaptive PD algorithms require both a training sequence *and* knowledge of $\mathbf{p}_1$.

The error rate P_e in Fig. 6 is computed assuming that the residual interference plus noise at the output of the adaptive filter is Gaussian. Specifically,

$$P_e \approx Q\left(\sqrt{\frac{A_1^2 |\tilde{\mathbf{c}}^{\dagger} \mathbf{p}_1|^2}{\tilde{\mathbf{c}}^{\dagger} \mathbf{R}_I \tilde{\mathbf{c}}}}\right) \quad (72)$$
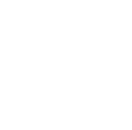

Fig. 6. Error rate versus the number of dimensions for reduced-rank adaptive algorithms after training with 200 symbols. $N = 128$, 42 asynchronous users, standard deviation of received powers = 6 dB, desired user's SNR = 10 dB. (a) Comparison of adaptive MSWF with reduced-rank filters based on eigen-decomposition. (b) Comparison of adaptive MSWF with PD methods.

where $\mathbf{R}_I$ is the covariance matrix for the interference plus noise [i.e., (10) without the desired signal $\mathbf{p}_1$], and $\tilde{\mathbf{c}}$ is the reduced-rank filter, which must be computed from the estimated MSWF parameters (see [10]), or equivalently, from (68). Results are averaged over random spreading codes, delays, and powers.

Fig. 6 shows that the adaptive reduced-rank techniques generally achieve optimum performance when $D < N$. Namely, when D is large, insufficient data is available to obtain an accurate estimate of the filter coefficients, whereas for small D, there are insufficient degrees of freedom to suppress interference. The minimum error rate for the adaptive MSWF is achieved with only eight stages (dimensions), which is much smaller than the minimizing order for the other reduced-rank techniques. Furthermore, this minimum error rate for the MSWF is substantially lower than the error rate for the matched filter receiver, and is not very far from the full-rank MMSE error rate. Additional simulations with only 100 training samples show that the minimum error rate for the adaptive MSWF is again achieved with $D = 8$.
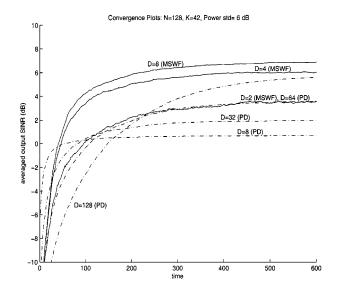
Fig. 7 shows output SINR versus time, or number of training symbols, for the "exact" MSWF algorithm given by (44)–(51).

Fig. 7. Output SINR versus time (training symbols) for recursive MSWF and RLS-PD algorithms. Parameters are the same as in Fig. 6.

Fig. 8. Output SINR versus time (number of received vectors) for blind adaptive MSWF algorithms. Parameters are the same as in Fig. 6.

Curves corresponding to different ranks D are shown. Analogous curves for the RLS algorithm with PD are also shown. System parameters are the same as in Fig. 6. The figure shows that a low-rank adaptive MSWF (e.g., $D = 4$) can converge significantly faster than the full-rank RLS, and has nearly the same asymptotic SINR. As expected, for the RLS with PD, as the dimension decreases, convergence speed increases, but asymptotic SINR decreases.

Fig. 8 compares the convergence of blind MSWF algorithms (i.e., $\hat{\mathbf{p}}_1 = \mathbf{p}_1$). Plots are shown for the exact MSWF with $D = 2$ and $D = 4$, and for the gradient MSWF with $D = 4$. The rank $D = 4$ filters perform best over a wide range of training intervals. Also shown are plots for the full-rank blind SG algorithm [6], the full-rank blind RLS algorithm (i.e., $\mathbf{c} = \hat{\mathbf{R}}^{-1}\mathbf{p}_1$), and the MSWF with training with $D = 8$. These results show that for the parameters selected, the reduced-rank algorithms converge significantly faster than the analogous full-rank algorithms. However, we remark that the full-rank blind RLS algorithm was found to be sensitive to the initialization of $\hat{\mathbf{R}}$, and the choice of exponential weight μ in (52). (The full-rank RLS algorithm with training is much less sensitive to the choice of these parameters.) Specifically, for the exact algorithms shown in Fig.8, $\mu = 1$ and $\hat{\mathbf{R}}(0) = 0.01 \times \mathbf{I}$. For the SG algorithms, $\mu = 0.98$. Reducing μ significantly improves the convergence speed of the full-rank RLS algorithm over relatively short training intervals, but this is traded off against degraded steady-state performance. In contrast, the performance of the blind MSWF is relatively insensitive to these parameters. These results also show that there is a noticeable degradation in performance in going from the training-based to blind to SG MSWF algorithms for the case considered. Still, these latter algorithms perform significantly better than the full-rank SG algorithm.

The initial degradation in performance shown for the blind algorithms (especially prominent for the full-rank RLS algorithm) occurs because the estimated covariance matrix $\hat{\mathbf{R}}$ is ill-conditioned for very short training intervals. This behavior has been verified analytically in [23]. Increasing the diagonal weights in

the initial estimate $\hat{\mathbf{R}}(0)$ reduces this initial degradation at the expense of somewhat slower convergence to steady-state.

VII. RANK ADAPTATION

Fig. 6 indicates that the performance of the adaptive MSWF can be a sensitive function of the rank D. Here we provide two adaptive methods for selecting the rank of the filter. Related work on rank selection for the Auxiliary Vector method is presented in [24]. The first method is based on the observation that the basis vectors $\hat{\mathbf{v}}_0, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_{D-1}$, where $\hat{\mathbf{v}}_m = \hat{\mathbf{R}}^m\hat{\mathbf{p}}_1$, are linearly dependent, or nearly dependent, for relatively small values of D. Furthermore, it is easily shown that if $\hat{\mathbf{v}}_D$ is in $\mathcal{S}_D$, the subspace spanned by $\hat{\mathbf{v}}_0, \ldots, \hat{\mathbf{v}}_{D-1}$, then $\hat{\mathbf{v}}_n \in \mathcal{S}_D$ for all $n \geq D$. This leads to the *stopping rule*

$$D = \max\left\{ n: \frac{\|\mathcal{P}_{\mathcal{S}_n}^{\perp}(\hat{\mathbf{v}}_n)\|}{\|\hat{\mathbf{v}}_n\|} > \delta^* \right\} \tag{73}$$

where $\mathcal{P}_{\mathcal{S}}^{\perp}(\mathbf{x})$ is the orthogonal projection of the vector $\mathbf{x}$ onto the subspace $\mathcal{S}$, and δ^* is a small positive constant.

For the powers of $\hat{\mathbf{R}}$ method, the stopping rule (73) prevents the matrix $\Gamma_{1:D}$ in (70) from being ill-conditioned. In the Appendix it is shown that

$$\|\mathcal{P}_{\mathcal{S}_n}^{\perp}(\hat{\mathbf{v}}_n)\| = \prod_{l=1}^{n+1} \delta_l \tag{74}$$

where δ_n is given by (37). We have not found an analogous expression for $\|\hat{\mathbf{v}}_n\|$ in terms of MSWF parameters, which is easily computable. Consequently, we do not have an equivalent stopping rule which can be conveniently used with Algorithms 1–3.

The second method for selecting the filter rank is based on estimating the MSE from the *a posteriori* LS cost function

$$C_D'(i) = \sum_{m=1}^{i} \mu^{i-m} |b_1(m) - \tilde{\mathbf{c}}_D^{\dagger}(m-1)\tilde{\mathbf{y}}_D(m)|^2 \tag{75}$$
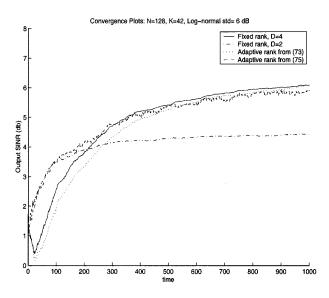
Fig. 9. Output SINR versus number of symbols for the blind adaptive MSWF with rank adaptation. Parameters are the same as in Fig. 6.

where the subscript D denotes the rank associated with the variable. For each i, we can select the D which minimizes $\mathcal{C}'_D(i)$. The exponential weighting factor μ is needed since the optimal rank D can change as a function of training interval i.

The preceding rank selection techniques were simulated for the same system model and parameters used to generate Fig. 6. For rank selection based on (73), we chose $\delta^* = 0.01$. Further simulations indicate that performance is insensitive to this choice over a reasonable range (i.e., between 10^{-3} and 10^{-2}). For the MSWF with training, the results essentially coincide with those shown for rank $D = 8$ in Fig. 7, although the second method, based on the *a posteriori* LS cost function, performs slightly worse than the first method. Further simulations and analysis indicate that rank $D = 8$ appears to be optimal, or nearly optimal, for a wide range of system parameters and training intervals [23]. This observation is consistent with the results in [14] (for synchronous CDMA), which show that the MSWF achieves essentially full-rank performance with rank $D = 8$.

For the blind adaptive MSWF, the optimal rank generally changes with the training interval, as shown in Fig. 8. For very short training intervals ($i < N$), $D = 1$ or $D = 2$ is best. The optimal D increases with training, but is generally ≤ 5, which is typically less than the optimal D for the MSWF with training. Fig. 9 shows output SINR versus training interval for the blind adaptive MSWF with rank selected according to (73), and rank selected by minimizing $\mathcal{C}'_D(i)$ in (75) with $\mu = 0.99$ for each i. Also shown are curves corresponding to fixed ranks $D = 4$ and $D = 2$. For the case simulated, the latter rank adaptation method is able to track the optimal rank fairly closely, whereas the former method tracks the performance with $D = 4$.

VIII. CONCLUSION

Adaptive reduced-rank linear filters have been presented based on the MSWF. These algorithms can be used in any adaptive filtering application, although the performance has been examined in the context of interference suppression for DS-CDMA. For large filter lengths, the MSWF allows a substantial reduction in rank, relative to other reduced-rank filters, such as those based on an eigen-decomposition of the sample covariance matrix. Numerical results show that the adaptive MSWF achieves near full-rank performance with fewer training samples than what is required by other full- and reduced-rank techniques. For the examples considered, an adaptive MSWF with rank eight achieves near full-rank performance with significantly less than $2N$ training samples, where N is the number of filter coefficients. Methods for tracking the optimal rank as a function of training interval have also been presented.

APPENDIX
DERIVATION OF (74)

It is shown in [14] that

$$\mathbf{c}_{n+1} = \frac{1}{\delta_{n+1}} \left(\mathbf{I} - \sum_{l=1}^{n} \mathbf{c}_l \mathbf{c}_l^\dagger \right) \mathbf{R}\mathbf{c}_n$$

$$= \frac{1}{\delta_{n+1}} \mathcal{P}_{\mathcal{S}_n}^\perp (\mathbf{R}\mathbf{c}_n) \tag{76}$$

where $\mathbf{c}_n$ is given by (20) for the MSWF and $\delta_{n+1} = \|\mathcal{P}_{\mathcal{S}_n}^\perp(\mathbf{R}\mathbf{c}_n)\|$ is a normalization constant. For the given data (unknown statistics) case, δ_n is given by (37).

From (27) and (76), we can write

$$\mathbf{c}_n = \frac{1}{\delta_n} \sum_{l=0}^{n-1} a_{l;n-1} \mathbf{v}_l \tag{77}$$

where $\mathbf{v}_l = \mathbf{R}^l \mathbf{p}_1$, and the $a_{l;n-1}$s are constants, so that

$$\mathbf{R}\mathbf{c}_n = \frac{1}{\delta_n} \sum_{l=0}^{n-1} a_{l;n-1} \mathbf{v}_{l+1}. \tag{78}$$

Combining (78) with (76) gives

$$\mathbf{c}_{n+1} = \frac{1}{\delta_{n+1}\delta_n} a_{n-1;n-1} \mathcal{P}_{\mathcal{S}_n}^\perp(\mathbf{v}_n). \tag{79}$$

To evaluate $a_{n-1;n-1}$, we combine (76) and (77) which gives

$$\sum_{l=0}^{n-1} a_{l;n-1} \mathbf{v}_l = \mathcal{P}_{\mathcal{S}_{n-1}}^\perp (\mathbf{R}\mathbf{c}_{n-1})$$

$$= \mathcal{P}_{\mathcal{S}_{n-1}}^\perp \left[\mathbf{R} \left(\frac{1}{\delta_{n-1}} \sum_{l=0}^{n-2} a_{l;n-2} \mathbf{v}_l \right) \right]$$

$$= \frac{1}{\delta_{n-1}} \mathcal{P}_{\mathcal{S}_{n-1}}^\perp \left(\sum_{l=0}^{n-2} a_{l;n-2} \mathbf{v}_{l+1} \right)$$

$$= \frac{1}{\delta_{n-1}} a_{n-2;n-2} \mathcal{P}_{\mathcal{S}_{n-1}}^\perp (\mathbf{v}_{n-1}). \tag{80}$$

Writing $\mathcal{P}_{\mathcal{S}_{n-1}}^\perp(\mathbf{v}_{n-1}) = \mathbf{v}_{n-1} - \mathcal{P}_{\mathcal{S}_{n-1}}(\mathbf{v}_{n-1})$, and equating right- and left-hand coefficients of $\mathbf{v}_{n-1}$ shows that

$$a_{n-1;n-1} = \frac{1}{\delta_{n-1}} a_{n-2;n-2} = \prod_{l=1}^{n-1} \frac{1}{\delta_l} \tag{81}$$

since $a_{0;0} = 1$. Combining with (79) establishes (74) for the known statistics case. The preceding derivation also applies to the given data case, where statistical averages are replaced by sample averages.

ACKNOWLEDGMENT

The authors thank a careful reviewer for detailed comments which helped to improve the paper significantly.

REFERENCES

[1] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[2] P. A. Zulch, J. S. Goldstein, J. R. Guerci, and I. S. Reed, "Comparison of reduced-rank signal processing techniques," in *Proc. 32nd Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 1998.

[3] L. Tong and S. Perreau, "Multichannel blind channel estimation: From subspace to maximum likelihood methods," *Proc. IEEE*, vol. 86, pp. 1951–1968, Oct. 1998.

[4] X. Wang and H. V. Poor, "Blind multiuser detection: A subspace approach," *IEEE Trans. Inform. Theory*, vol. 44, pp. 677–690, Mar. 1998.

[5] M. L. Honig, "A comparison of subspace adaptive filtering techniques for DS-CDMA interference suppression," in *Proc. IEEE MILCOM*, Monterey, CA, Nov. 1997, pp. 836–840.

[6] M. L. Honig and H. V. Poor, "Adaptive interference suppression," in *Wireless Communications: Signal Processing Perspectives*, H. V. Poor and G. W. Wornell, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1998, ch. 2, pp. 64–128.

[7] Y. Song and S. Roy, "Blind adaptive reduced-rank detection for DS-CDMA signals in multipath channels," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1960–1970, Nov. 1999.

[8] D. A. Pados and S. N. Batalama, "Joint space–time auxiliary-vector filtering for DS/CDMA systems with antenna arrays," *IEEE Trans. Commun.*, vol. 47, pp. 1406–1415, Sept. 1999.

[9] J. S. Goldstein and I. S. Reed, "Multidimensional Wiener filtering using a nested chain of orthogonal scalar Wiener filters," University of Southern California, Los Angeles, CA, USC Tech. Rep. CSI-96-12-04, Dec. 1996.

[10] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A multistage representation of the Wiener filter based on orthogonal projections," *IEEE Trans. Inform. Theory*, vol. 44, Nov. 1998.

[11] J. S. Goldstein and I. S. Reed, "Reduced rank adaptive filtering," *IEEE Trans. Signal Processing*, vol. 45, pp. 492–496, Feb. 1997.

[12] K. A. Byerly and R. A. Roberts, "Output power based partial adaptive array design," in *Twenty-Third Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Oct.r 1989, pp. 576–580.

[13] D. A. Pados and G. N. Karystinos, "An iterative algorithm for the computation of the MVDR filter," *IEEE Trans. Signal Processing*, vol. 49, pp. 290–300, Feb. 2001.

[14] M. L. Honig and W. Xiao, "Performance of reduced-rank linear interference suppression," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1928–1946, July 2001.

[15] S. P. Applebaum and D. J. Chapman, "Adaptive arrays with main beam constraints," *IEEE Trans. Antennas Propagat.*, vol. AP-24, pp. 650–662, Sept. 1976.

[16] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 27–34, Jan. 1982.

[17] L. L. Scharf and D. W. Tufts, "Rank reduction for modeling stationary signals," *IEEE Trans. Signal Processing*, vol. SP-35, pp. 350–355, Mar. 1987.

[18] B. D. Van Veen, "Eigenstructure based partially adaptive array design," *IEEE Trans. Antennas Propagat.*, vol. 36, pp. 357–362, Mar. 1988.

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

[20] R. Singh and L. B. Milstein, "Interference suppression for DS/CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 446–453, Mar. 1999.

[21] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*. Boston, MA: Kluwer Academic, 1985.

[22] J. S. Goldstein, "The dynamic behavior of constrained adaptive array sensor processors," USAF Rome Lab. Tech. Rep. RL-TR-92-327, Dec. 1992.

[23] W. Xiao and M. L. Honig, "Convergence analysis of adaptive full-rank and multistage reduced-rank interference suppression," in *Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2000.

[24] H. Qian and S. N. Batalama, "Data-record-based criteria for the selection of an auxiliary-vector estimator of the MVDR filter," in *Proc. 34th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 2000, pp. 802–807.

Michael L. Honig (S'80–M'81–SM'92–F'97) received the B.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1977 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1978 and 1981, respectively.

He subsequently joined Bell Laboratories in Holmdel, NJ, where he worked on local area networks and voiceband data transmission. In 1983, he joined the Systems Principles Research Division at Bellcore, where he worked on digital subscriber lines and wireless communications. He was a Visiting Lecturer at Princeton University, Princeton, NJ, during the fall of 1993. Since the fall of 1994, he has been with Northwestern University, Evanston, IL, where he is a Professor in the Electrical and Computer Engineering Department. He was a Visiting Mackay Professor at the University of California, Berkeley, during the fall of 2000.

Dr. Honig has served as an Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and the IEEE TRANSACTIONS ON COMMUNICATIONS, and was a Guest Editor for *European Transactions on Telecommunications* and *Wireless Personal Communications*. He was also a member of the Digital Signal Processing Technical Committee for the IEEE Signal Processing Society. He is a member of the Board of Governors for the IEEE Information Theory Society.

J. Scott Goldstein (S'87–M'88–SM'96–F'01) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1997.

He is an Assistant Vice President and Senior Scientist at SAIC, Chantilly, VA, where he serves as the Manager for Adaptive Signal Exploitation. His responsibilities include program development, program management and the technical leadership of over twenty staff members in the area of advanced signal processing for detection, estimation, classification and recognition within the fields of radar, sonar, spectral sensing, communications and navigation. In addition, he is an Adjunct Professor in the Department of Electrical and Computer Engineering at the Virginia Polytechnic Institute and State University, Blacksburg. He also serves as a reserve officer in the U.S. Air Force, where he is involved in research and development efforts for advanced sensor programs. Previously, he was a staff member at MIT Lincoln Laboratory, Lexington, MA, served as Vice-President and Chief Scientist of Adaptronics, Inc., was a Shackelford Fellow with the Radar Systems Division of the Sensors and Electromagnetic Applications Laboratory at the Georgia Tech Research Institute, Atlanta, and was a staff member at the Institute for Defense Analyzes. Prior to this he served in the U.S. Army and currently has nearly 20 years of military service. He has also served as a consultant to the Army Research Laboratory and Adaptive Sensors, Inc. He has published over 100 technical papers.

Dr. Goldstein is a member of Sigma Xi, Tau Beta Pi, and Eta Kappa Nu. He is currently serving on the IEEE Fellow Committee, as the Assistant Administrative Editor for the IEEE Aerospace and Electronic Systems Society, and is a member of the IEEE Radar Systems Panel. He is a co-recipient of the IEE Clarke-Griffith Memorial Paper Award and Premium for best paper in the IEE Proceedings—Radar, Sonar and Navigation. He is also a recipient of the IEEE AESS EASCON Award, the Japanese Okawa Foundation Research Grant and the AFCEA Postgraduate Fellowship. In addition, he has received over 15 Air Force Awards for Scientific Achievement due to his research in radar and communications.