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

Michael L. Honig† J. Scott Goldstein‡

Abstract

A class of adaptive reduced-rank interference suppression algorithms is presented based on the multistage Wiener filter introduced by Goldstein & Reed. 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 perform well when the filter rank is much less than the dimension of the signal subspace. We present batch and recursive algorithms for estimating the filter parameters, which do not require matrix inversion or 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.

*Manuscript submitted to *IEEE Transactions on Communications*, March 2000. This work was presented in part at the *1998 Asilomar Conference on Signals, Systems, and Computing.*

†Dept. of Electrical & Computer Engineering, Northwestern University, Evanston, IL 60208; Tel/Fax: (847) 491-7803/467-3550, e-mail: mh@ece.nwu.edu. Supported by U. S. Army Research Office under grant DAAD19-99-1-0288.

‡SAIC, 4001 N. Fairfax Drive, Suite 400, Arlington, VA 22203; e-mail: sgoldstein@trg1.saic.com.

1 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, 5, 6, 7, 8, 7]. In this paper we present reduced-rank adaptive filtering algorithms which are based on the multistage Wiener filter of Goldstein and Reed [9]. The performance of these techniques 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 [10]), 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" 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" method, proposed in [11] (see also [12]), and the Multistage Wiener (MSW) filter, presented in [9]. Unlike the Cross-Spectral and principal components methods, the MSW filter 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 [13, 9]. (See also [8], in which an equivalent algorithm is presented.) This technique can attain near full-rank Minimum Mean Squared Error (MMSE) performance with a filter order 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 estimate the filter parameters.

We present a class of adaptive filtering algorithms which are motivated by the MSW filter. These algorithms do not require an eigen-decomposition or matrix inversion, 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 [10]). The performance of the adaptive MSW techniques are illustrated numerically, and are compared with other adaptive reduced-rank techniques.

The next section presents the DS-CDMA model, Sections 3 and 4 review reduced-rank MMSE filtering and the MSW filter, and Section 5 presents the adaptive MSW algorithms. Numerical results are presented in Section 6, and adaptive rank selection is discussed in Section 7.

2 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, 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, 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 6 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

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

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.

3 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 [10]. 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 [10]. However, large N implies slow convergence speed. 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

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

where

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

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

and

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

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 MSW filter, we briefly mention other reduced-rank filters, which have been previously proposed. The performance of the adaptive MSW algorithms to be described will be compared with the performance of these other methods in Section 6. A simulation study of the adaptive eigen-decomposition and partial despreading methods is presented in [5].

3.1 Eigen-decomposition Techniques

Principal Components (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 $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$, 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.

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

$$\mathcal{M}_D = 1 - \|\Lambda^{-1}\tilde{\mathbf{p}}_1\|^2 \tag{14}$$

To minimize $\mathcal{M}_D$, the basis vectors should be the eigenvectors of $\mathbf{R}$ associated with the D largest values of $|\tilde{\mathbf{p}}_{1,k}/\lambda_k|^2$, 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 $|\lambda_k|^2$ in contrast with PC.)

This technique, called "Cross-Spectral (CS)" reduced-rank filtering, was proposed in [12] and [11]. This technique can perform well for $D < K$ since it takes into account the energy in the subspace contributed by the desired user. Unlike PC, the projection subspace for CS requires 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.

3.2 Partial Despreading

In this method, proposed in [15], 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 $\mathbf{M}_D$ 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 $\mathbf{M}_D$ is

$$[\mathbf{M}_D]_j{}' = [0 \ldots 0 \; \mathbf{p}_{1,[(j-1)m+1:jm]}{}' \; 0 \ldots 0] \tag{15}$$

where $1 \leq j \leq D$, 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.

4 The Multistage Wiener (MSW) Filter

The MSW filter was presented in [9] for the known statistics case, i.e., known covariance matrix $\mathbf{R}$ and steering vector $\mathbf{p}_1$. A block diagram of a four-stage MSW filter is shown in Figure 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.,

$$\mathbf{B}_m^\dagger \mathbf{c}_m = \mathbf{0}. \tag{16}$$

Referring to Figure 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

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

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 [9], 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 Figure 1, let

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

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

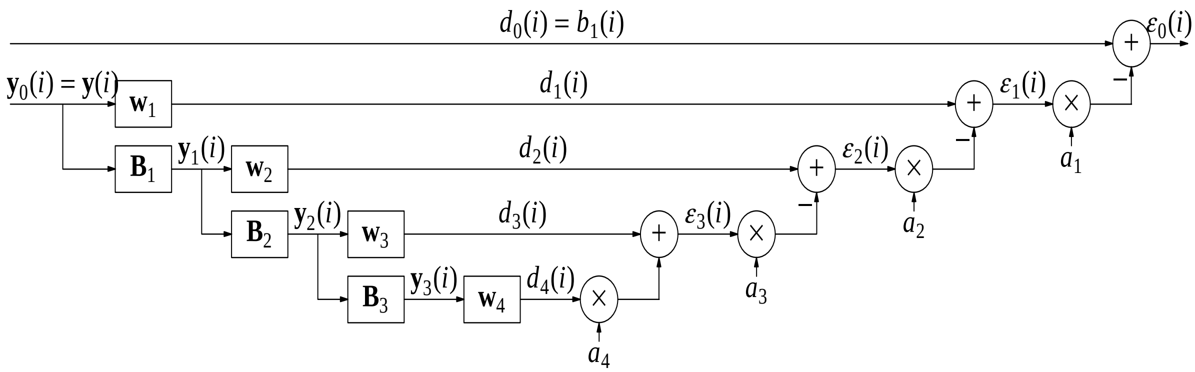

Figure 1: Multi-stage Wiener filter.

The rank D MSW filter is given by the following set of recursions.

Initialization:

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

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

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

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

$$\mathbf{B}_n = \mathbf{I} - \mathbf{c}_n \mathbf{c}_n^\dagger \tag{22}$$

$$\mathbf{y}_n = \mathbf{B}_n^\dagger \mathbf{y}_{n-1} \tag{23}$$

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

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

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

where $\epsilon_D(i) = \mathbf{d}_D(i)$.

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 MSW filter by the MMSE filter $\mathbf{c}_n^{(mmse)}$ for estimating $\{d_n(i)\}$ from $\{\mathbf{y}_n(i)\}$ gives the full-rank MMSE filter. The MSW is "self-similar" in the sense that the MMSE filter $\mathbf{c}_n^{(mmse)}$ is replaced by the associated MSW filter. The covariance matrix for the projected vector $\tilde{\mathbf{y}} = [d_0,\ d_1, \ldots, d_{D-1}]$ is *tri-diagonal* [9].

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

1. Let $\mathcal{S}_{\mathcal{D}}$ denote the D-dimensional subspace associated with the rank D MSW filter. Then

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

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

   where the first set of basis vectors is an orthonormal set, and the basis vectors in the second set are not orthogonal.

2. The rank D needed to achieve full-rank performance does not scale with system size (K and N). In particular, full-rank performance is essentially achieved with rank $D \leq 8$ for a wide range of loads and Signal-to-Noise Ratios (SNRs).

5 Adaptive Reduced-Rank Algorithms

In this section, we present a family of adaptive algorithms which are related to the MSW filter. 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 [16]. Namely, for the known statistics case, we define the inner product between two random variables X and Y as $< X, Y >= E[X^*Y]$, 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,

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

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

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

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

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.

5.1 Batch Algorithms

Here we consider estimation of the MSW filter parameters given $\mathbf{Y}$ and $\bar{\mathbf{b}}$ in (28) and (29). The approach just described leads to Algorithm 1, the batch adaptive MSW filter with training (35-43). Following the approach in [9], it is straightforward to show that this algorithm tri-diagonalizes the $(N+1) \times (N+1)$ extended sample covariance matrix

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

where

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

Algorithm 1 Batch adaptive MSW algorithm with training.

Initialization:

$$\mathbf{d}_0 = \bar{\mathbf{b}} \qquad \mathbf{Y}_0 = \mathbf{Y} \tag{35}$$

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

$$\hat{\mathbf{p}}_n = \mathbf{Y}_{n-1}\mathbf{d}_{n-1}^{\dagger} \quad (N-n+1)\times 1 \tag{36}$$

$$\delta_n = \|\hat{\mathbf{p}}_n\| \tag{37}$$

$$\mathbf{c}_n = \hat{\mathbf{p}}_n/\delta_n \tag{38}$$

$$\mathbf{d}_n = \mathbf{c}_n^{\dagger}\mathbf{Y}_{n-1} \quad 1\times M \tag{39}$$

$$\mathbf{B}_n = \text{null}(\mathbf{c}_n) \quad (N-n+1)\times(N-n) \tag{40}$$

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

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 choice of blocking matrices $\mathbf{B}_n$, $n = 1, \ldots, D$, can affect performance for a specific data record, although asymptotically, as $M \to \infty$, the corresponding MSE is independent of this choice. The numerical results in Section 6 assume

$$\mathbf{B}_n = \mathbf{I}_{N-n+1,N-n} - \mathbf{c}_n \mathbf{c}_{n,(1:N-n)}^{\dagger} \tag{44}$$

where $\mathbf{I}_{m,n}$ is the $m \times n$ identity matrix and the subscript $(1 : m)$ denotes the first m components of the corresponding vector. Note that $\epsilon_0 = \min\ \mathcal{C}_D$, the minimized LS cost function in (31). When used in decision-directed mode, the estimate of the block of transmitted symbols $\bar{\mathbf{b}}$ is $w_1^* \epsilon_1$, where ϵ_n is computed from (43).

A non-training 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 hence does not perform as well as a training-based algorithm. An illustrative example is given in Section 6.

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 $\hat{\mathbf{R}} = \mathbf{Y}\mathbf{Y}^{\dagger}$.

5.2 Recursive Algorithms

A recursive version of the preceding batch adaptive MSW algorithm tri-diagonalizes the sample covariance matrix $\hat{\mathbf{R}}$ at each iteration. A recursive update for the extended sample covariance matrix is:

$$\hat{\bar{\mathbf{R}}}(i) = (1 - \mu) * \hat{\bar{\mathbf{R}}}(i - 1) + \mu \bar{\mathbf{y}}(i) \bar{\mathbf{y}}^{\dagger}(i), \tag{57}$$

where μ is a smoothing factor which discounts past data. Algorithm 2, given by (45-56), tri-diagonalizes the sample covariance matrix $\hat{\mathbf{R}}$ at each i [9]. Specifically, let $\bar{\mathbf{D}} = [\mathbf{d}_1; \mathbf{d}_2; \ldots; \mathbf{d}_D]$, where $\mathbf{d}_n$ is defined by (39), and ";" separates rows, so that $\bar{\mathbf{D}}$ is $D \times M$. Then Algorithm 2

Algorithm 2 Adaptive MSW algorithm based on tri-diagonalization of $\hat{\mathbf{R}}$.

Initialization:

$$\mathbf{T}_1 = \hat{\hat{\mathbf{R}}}(i) \qquad \hat{\mathbf{p}}_1 = \hat{\hat{\mathbf{R}}}_{1;(2:N,1)}(i) \tag{45}$$

$$\delta_1 = \|\hat{\mathbf{p}}_1\| \qquad \mathbf{c}_1 = \hat{\mathbf{p}}_1/\delta_1 \tag{46}$$

$$\mathbf{y}_0 = \mathbf{y}(i) \qquad d_0 = b_1(i) \tag{47}$$

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

$$\mathbf{B}_n = \mathrm{null}(\mathbf{c}_n^\dagger) \tag{48}$$

$$\mathbf{L}_n = \begin{bmatrix} \mathbf{I}_{n,n} & | & \mathbf{0}_{n,N-n} \\ & & \mathbf{c}_n^\dagger \\ \mathbf{0}_{N-n,n} & | & ---- \\ & & \mathbf{B}_n^\dagger \end{bmatrix} \tag{49}$$

$$\mathbf{T}_{n+1} = \mathbf{L}_n \mathbf{T}_n \mathbf{L}_n^\dagger \tag{50}$$

$$\hat{\mathbf{p}}_{n+1} = \mathbf{T}_{n+1;(n+2:N,n+1)} \tag{51}$$

$$\delta_{n+1} = \|\hat{\mathbf{p}}_{n+1}\| \qquad \mathbf{c}_{n+1} = \hat{\mathbf{p}}_{n+1}/\delta_{n+1} \tag{52}$$

$$\mathbf{y}_n = \mathbf{B}_n^\dagger \mathbf{y}_{n-1} \qquad d_n = \mathbf{c}_n^\dagger \mathbf{y}_n \tag{53}$$

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

$$\xi_n = \mathbf{T}_{D;(n+1,n+1)} - w_{n+1}^* \delta_{n+1} \tag{54}$$

$$w_n = \delta_n/\xi_n \tag{55}$$

$$\epsilon_{n-1} = d_{n-1} - w_n^* \epsilon_n \tag{56}$$

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

Symbol estimate $= w_1^*(i)\epsilon_1(i)$

computes the tri-diagonal matrix $\mathbf{T}_D = \bar{\mathbf{D}}\bar{\mathbf{D}}^\dagger$. 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 $\mathbf{X}$.

Rather than perform an exact tri-diagonalization of the sample covariance matrix at each iteration, it is also possible to approximate the MSW filter parameters via sample averages. This leads to Algorithm 3, given by (58-66), the MSW "stochastic gradient" algorithm. This algorithm is computationally simpler than the preceding algorithm (45-56), but does not exactly tridiagonalize the extended sample covariance matrix $\hat{\hat{\mathbf{R}}}(i)$ at each iteration. Consequently, Algorithm 3 does not perform as well as the preceding "exact" Algorithm 2, as the results in Section 6 illustrate.

5.3 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

$$\hat{\mathbf{S}}_D(i) = \left[\hat{\mathbf{p}}_1(i),\ \hat{\mathbf{R}}(i)\hat{\mathbf{p}}_1(i),\ \hat{\mathbf{R}}^2(i)\hat{\mathbf{p}}_1(i), \ldots, \hat{\mathbf{R}}^{D-1}(i)\hat{\mathbf{p}}_1(i)\right] \tag{67}$$

where

$$\hat{\mathbf{p}}_1(i) = (1-\mu)\hat{\mathbf{p}}_1(i-1) + \mu d^*(i)\mathbf{y}(i) \tag{68}$$

and $\hat{\mathbf{R}}(i)$ is updated according to (57).

Let

$$\hat{\gamma}_m = \hat{\mathbf{p}}_1^\dagger \hat{\mathbf{R}}^m \hat{\mathbf{p}}_1 \tag{72}$$

$$\hat{\gamma}_{l:m} = [\hat{\gamma}_l\ \hat{\gamma}_{l+1} \ldots \hat{\gamma}_m]' \tag{73}$$

$$\hat{\Gamma}_{l:l+m} = [\hat{\gamma}_{l:l+m}\ \hat{\gamma}_{l+1:l+m+1} \cdots \hat{\gamma}_{l+m:l+2m}] \tag{74}$$

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

$$\tilde{\mathbf{c}}_D = (\hat{\mathbf{S}}_D^\dagger \hat{\mathbf{R}} \hat{\mathbf{S}}_D)^{-1} \hat{\mathbf{S}}_D^\dagger \hat{\mathbf{p}}_1 = \hat{\Gamma}_{1:D}^{-1} \hat{\gamma}_{0:D-1} \tag{75}$$

Algorithm 3 Stochastic gradient adaptive MSW algorithm.

Initialization:

$$d_0(i) = b_1(i) \qquad \mathbf{y}_0(i) = \mathbf{y}(i) \tag{58}$$

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

$$\hat{\mathbf{p}}_n(i) = (1-\mu)\hat{\mathbf{p}}_n(i-1) + \mu d^*_{n-1}(i)\mathbf{y}_{n-1}(i) \tag{59}$$

$$\delta_n(i) = \|\hat{\mathbf{p}}_n(i)\|, \qquad \mathbf{c}_n(i) = \hat{\mathbf{p}}_n(i)/\delta_n(i) \tag{60}$$

$$\mathbf{B}_n(i) = \mathrm{null}[\mathbf{c}_n^\dagger(i)] \tag{61}$$

$$d_n(i) = \mathbf{c}_n^\dagger(i)\mathbf{y}_{n-1}(i) \tag{62}$$

$$\mathbf{y}_n(i) = \mathbf{B}_n^\dagger(i)\mathbf{y}_{n-1}(i) \tag{63}$$

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

$$\xi_n(i) = (1-\mu)\xi_n(i-1) + \mu|\epsilon_n|^2 \tag{64}$$

$$w_n(i) = \delta_n(i)/\xi_n(i) \tag{65}$$

$$\epsilon_{n-1}(i) = d_{n-1}(i) - w_n^*(i)\epsilon_n(i) \tag{66}$$

where $\epsilon_D(i) = d_D(i)$.

Filter output (estimate of $b_1(i)$) $= w_1^*(i)\epsilon_1(i)$

Algorithm 4 Batch adaptive algorithm based on powers of $\hat{\mathbf{R}}$.

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

For $m = 0, \ldots, D-1$, $\hat{\mathbf{v}}_m = \hat{\mathbf{R}}^m \hat{\mathbf{p}}_1$

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

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

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

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

Given $\mathbf{Y}$ and $\bar{\mathbf{b}}$ in (28) and (29), a reduced-rank batch algorithm with training is given by Algorithm 4 (69-71). If $\mathbf{p}_1$ is known, then in the absence of training, $\hat{\mathbf{p}}_1(i)$ in (67) and (72) can be replaced by $\mathbf{p}_1$.

Following the same argument used to prove Theorem 2 in [14], it can be shown that Algorithm 4 is equivalent to Algorithm 1 with blocking matrix $\mathbf{B}_n = \mathbf{I} - \mathbf{c}_n \mathbf{c}_n^\dagger$. 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.

6 Numerical Results

Figure 2 shows error rate vs. number of dimensions for reduced-rank adaptive algorithms after training with 200 symbols. In this plot $N = 128$, $K = 42$, and the received powers are log-normal with standard deviation 6 dB. Results are shown for the following algorithms: MSW, Cross-Spectral (CS), 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. Three curves are shown for Partial Despreading (PD), corresponding to the way the vector $\tilde{\mathbf{c}}$ 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 error rate P_e in Figure 1 is computed via the Gaussian approximation for residual interference plus noise at the output of the adaptive filter. Specifically,

$$P_e \approx Q\left(\sqrt{\frac{|\tilde{\mathbf{c}}^\dagger \mathbf{p}_1|^2}{\tilde{\mathbf{c}}^\dagger \mathbf{R}_I \tilde{\mathbf{c}}}}\right) \tag{76}$$

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 MSW filter parameters (see [9]), or equivalently, from (75). Results are averaged over random spreading codes, delays, and powers.

Figure 2 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 MSW algorithm 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 MSW algorithm 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 MSW algorithm is again achieved with $D = 8$.

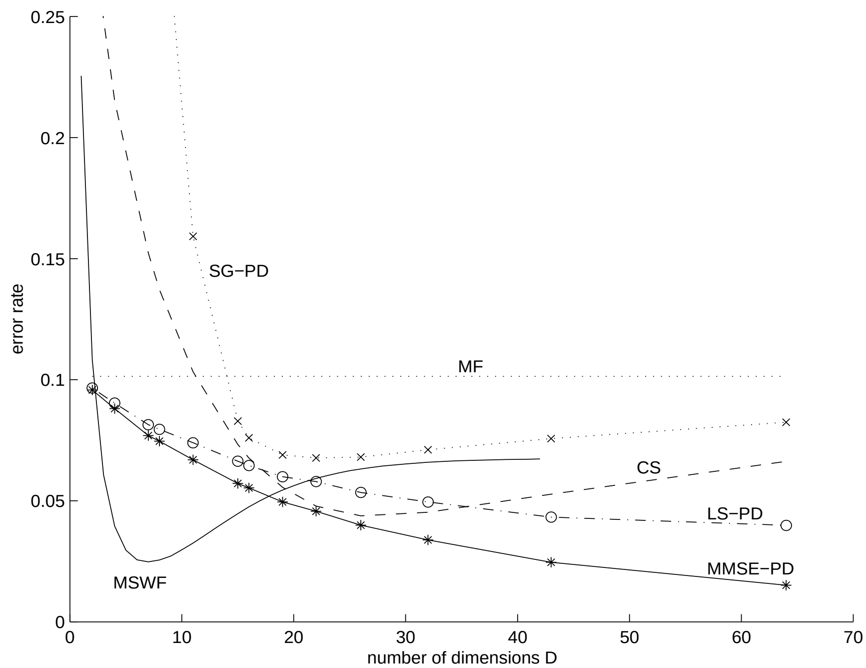

Figure 2: Error rate vs. number of dimensions for reduced-rank adaptive algorithms after training with 200 symbols. $N = 128$, 42 asynchronous users, power standard deviation= 6 dB

Figure 3 shows output SINR vs. time for the "exact" recursive MSW algorithm given by (45-56). 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 Figure 2. The figure shows that a low-rank MSW algorithm (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.

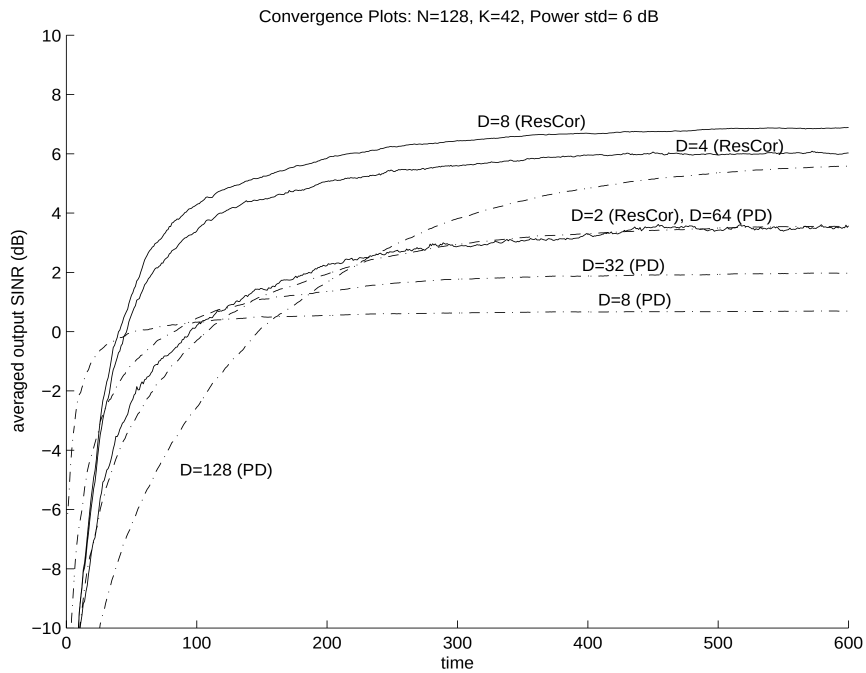

Figure 3: Output SINR vs. time for recursive MSW and RLS-PD algorithms.

Figure 4 compares the convergence of the different MSW algorithms described in Section 5 for $D = 8$. In this case $N = 32$, $K = 16$, and the power of each interferer is 6 dB greater than the power of the desired user. Also shown are convergence curves for the full-rank RLS algorithm with training and the stochastic gradient (LMS) algorithm. These results show that there is a noticeable degradation in performance in going from training-based to blind to gradient MSW algorithms. Still, these latter algorithms perform significantly better than the full-rank LMS algorithm.

7 Rank Adaptation

Figure 2 indicates that the performance of the adaptive MSW filter can be a sensitive function of the rank D. Here we provide two adaptive methods for selecting the rank of the filter. The

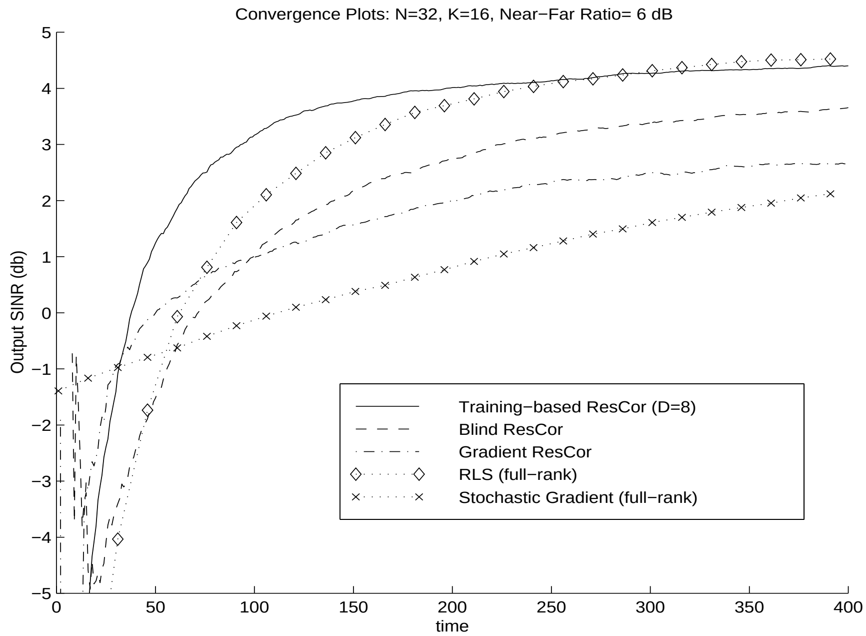

Figure 4: Output SINR vs. time for different adaptive MSW algorithms.

first method is based on the observation that for small sets of training data, the basis vectors $\hat{\mathbf{v}}_1,\ \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_{D-1}$, where $\hat{\mathbf{v}}_m = \hat{\mathbf{R}}^m \hat{\mathbf{p}}_1$, are linearly dependent (or nearly dependent). Furthermore, it is easily shown that if $\hat{\mathbf{v}}_{D+1}$ is in $\mathcal{S}_D$, the subspace spanned by $\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_D$, then $\hat{\mathbf{v}}_n \in \mathcal{S}_D$ for all $n > D$. This leads to the *stopping rule*:

$$D = \max \left\{ n : \frac{\|\mathcal{P}^{\perp}_{\mathcal{S}_{n-1}}(\hat{\mathbf{v}}_n)\|}{\|\hat{\mathbf{v}}_n\|} > \delta^* \right\} \tag{77}$$

where $\mathcal{P}^{\perp}_{\mathcal{S}}(\mathbf{x})$ 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 (77) prevents the matrix , $_{1:D}$ in (70) from being ill-conditioned. In the Appendix it is shown that

$$\|\mathcal{P}^{\perp}_{\mathcal{S}_{n-1}}(\hat{\mathbf{v}}_n)\| = \prod_{l=1}^{n} \delta_l \tag{78}$$

where δ_n is given by (37). We have not found an analogous expression for $\|\hat{\mathbf{v}}_n\|$ in terms of MSW filter 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

$$\mathcal{C}'_D(i) = \sum_{m=1}^{i} |b_1(m) - \tilde{\mathbf{c}}_D^\dagger(m-1)\tilde{\mathbf{y}}_D(m)|^2 \tag{79}$$

For each i, we can select the D which minimizes $\mathcal{C}'_D(i)$.

The preceding rank selection techniques were simulated for the same system model and parameters used to generate Figure 3. The results essentially coincide with those shown for rank $D = 8$ in the figure, 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 [17]. This observation is consistent with the results in [14] (for synchronous CDMA), which show that the MSW filter achieves essentially full-rank performance with rank $D = 8$.

8 Conclusions

Adaptive reduced-rank linear filters have been presented based on the MSW filter. 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 MSW filter allows a large reduction in rank, relative to other reduced-rank filters, such as those based on an eigen-decomposition of the sample covariance matrix. The adaptive MSW filter can therefore achieve near full-rank performance with many fewer training samples than what is required by other full- and reduced-rank techniques. For the examples considered, an adaptive MSW filter 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 have also been presented. These methods successfully identify the optimal filter rank for the cases considered.

A Derivation of (78)

It is shown in [14] that

$$\begin{aligned} \mathbf{c}_{n+1} &= \frac{1}{\delta_{n+1}} \left(\mathbf{I} - \sum_{l=1}^{n} \mathbf{c}_l \mathbf{c}_l^{\dagger} \right) \mathbf{R}\mathbf{c}_n \\ &= \frac{1}{\delta_{n+1}} \mathcal{P}^{\perp}_{\mathcal{S}_n}(\mathbf{R}\mathbf{c}_n) \end{aligned} \tag{80}$$

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

From (27) and (80), we can write

$$\mathbf{c}_n = \frac{1}{\delta_n} \sum_{l=1}^{n} a_{l;n} \mathbf{v}_l \tag{81}$$

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

$$\mathbf{R}\mathbf{c}_n = \frac{1}{\delta_n} \sum_{l=1}^{n} a_{l;n} \mathbf{v}_{l+1}. \tag{82}$$

Combining (82) with (80) gives

$$\mathbf{c}_{n+1} = \frac{1}{\delta_{n+1}\delta_n} a_{n;n} \mathcal{P}^{\perp}_{\mathcal{S}_n}(\mathbf{v}_{n+1}) \tag{83}$$

To evaluate $a_{n;n}$, we combine (80) and (81) which gives

$$\begin{aligned} \sum_{l=1}^{n} a_{l;n} \mathbf{v}_l &= \mathcal{P}^{\perp}_{\mathcal{S}_{n-1}}(\mathbf{R}\mathbf{c}_{n-1}) \\ &= \mathcal{P}^{\perp}_{\mathcal{S}_{n-1}} \left(\mathbf{R} \frac{1}{\delta_{n-1}} \sum_{l=1}^{n-1} a_{l;n-1} \mathbf{v}_l \right) \\ &= \frac{1}{\delta_{n-1}} \mathcal{P}^{\perp}_{\mathcal{S}_{n-1}} \left(\sum_{l=1}^{n-1} a_{l;n-1} \mathbf{v}_{l+1} \right) \\ &= \frac{1}{\delta_{n-1}} a_{n-1;n-1} \mathcal{P}^{\perp}_{\mathcal{S}_{n-1}}(\mathbf{v}_n) \end{aligned} \tag{84}$$

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

$$a_{n;n} = \frac{1}{\delta_{n-1}} a_{n-1;n-1} = \prod_{l=1}^{n-1} \frac{1}{\delta_l}, \tag{85}$$

since $a_{1;1} = 1/\delta_1$. Combining with (83) establishes (78) for the known statistics case. The preceding derivation also applies to the given data case, where statistical averages are replaced by sample averages.

References

[1] D. H. Johnson and D. E. Dudgeon. *Array Signal Processing: Concepts and Techniques.* Prentice-Hall, Englewood Cliffs, NJ, 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, November 1998.

[3] L. Tong and S. Perreau. Multichannel blind channel estimation: from subspace to maximum likelihood methods. *Proceedings of the IEEE*, 86(10):1951–1968, October 1998.

[4] X. Wang and H.V. Poor. Blind multiuser detection: A subspace approach. *IEEE Trans. Inform. Theory*, 44(2):677–690, March 1998.

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

[6] H.V. Poor and G.W. Wornell, editors. *Wireless Communications: Signal Processing Perspectives.* Prentice-Hall, Englewood Cliffs, NJ, 1998.

[7] Y. Song and S. Roy. Blind adaptive reduced-rank detection for DS-CDMA signals in multipath channels. *IEEE Journal on Selected Areas in Communications*, 17(11):1960–1970, November 1999.

[8] D. A. Pados and S. N. Batalama. Joint space-time auxiliary-vector filtering for DS/CDMA systems with antenna arrays. *IEEE Transactions on Communications*, 47(9):1406–1415, September 1999.

[9] 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*, 44(7), November 1998.

[10] M.L. Honig and H.V. Poor. Adaptive interference suppression. In H.V. Poor and G.W. Wornell, editors, *Wireless Communications: Signal Processing Perspectives*, chapter 2, pages 64–128. Prentice-Hall, Englewood Cliffs, NJ, 1998.

[11] J.S. Goldstein and I.S. Reed. Reduced rank adaptive filtering. *IEEE Trans. Signal Processing*, 45(2):492–496, February 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*, pages 576–580, Pacific Grove, CA, October 1989.

[13] J.S. Goldstein and I.S. Reed. A new method of Wiener filtering and its application to interference mitigation for communications. In *Proc. IEEE MILCOM*, volume 3, pages 1087–1091, Monterey, CA, November 1997.

[14] M. L. Honig and W. Xiao. Performance of reduced-rank linear interference suppression for DS-CDMA. submitted to *IEEE Trans. Inform. Theory*, 1999.

[15] R. Singh and L. B. Milstein. Interference suppression for DS/CDMA. *IEEE Transactions on Communications*, 47(3):446–453, March 1999.

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

[17] W. Xiao and M. L. Honig. Convergence analysis of adaptive full-rank and multistage reduced-rank interference suppression. In *Conference on Information Sciences and Systems*, Princeton, NJ, March 2000.