# Reduced-Rank Adaptive Filtering Using Localized Processing for CDMA Systems

Rebecca Y. M. Wong and Raviraj Adve, *Senior Member, IEEE*

***Abstract*—An integrated beamforming (spatial processing) and multiuser-detection (temporal processing) scheme is an effective approach to increase system capacity but is also impractical due to the high associated computational costs. To overcome this problem, researchers have developed reduced-rank approaches. Adding to this class of algorithms, this paper introduces a joint-domain adaptive algorithm, which processes spatial and temporal data within a localized region after transformation to a "beamspace." This new joint-domain-localized (JDL) adaptive algorithm is developed for single-cell uplink code-division multiple-access (CDMA) systems with a receive-antenna array. Given a rank constraint on the JDL transformation, this paper develops the optimal choice of transformation matrix and justifies a simpler suboptimal choice. The JDL algorithm is shown to have relatively low computation load. We also introduce the JDL-Z algorithm that combines JDL processing with zero forcing for multicell uplink CDMA systems. At the cost of higher computational complexity, this new scheme provides better performance and a faster convergence rate than the JDL algorithm. However, the key contribution is a framework wherein adaptive processing can be followed by yet another stage of adaptive processing. Simulations are used to illustrate the efficacy of the two algorithms.**

***Index Terms*—Beamforming, code-division multiple access (CDMA), joint-domain processing, multiuser detection, zero forcing.**

## I. Introduction

COMBINING space-division multiple access with temporal multiple-access techniques, such as code-division multiple access (CDMA), which is sometimes referred to as 2-D-CDMA, helps maximize system capacity without sacrificing bandwidth [1]. However, both forms of multiple-access schemes are interference-limited, which requires interference suppression to achieve their potential. In this regard, based on the minimum mean-squared error (MMSE) criterion, joint-domain adaptive processing that integrates receive beamforming (spatial processing) and multiuser detection (temporal processing) outperforms all possible linear processing [2]. For a system with $N$ array elements and a spreading gain of $G$, the jointly optimal MMSE (OMMSE) algorithm finds an adaptive weight for all $NG$ spatio-temporal degrees of freedom. Unfortunately, this processor is prohibitively expensive computationally and is also inefficient in terms of the required training symbols.

Manuscript received August 23, 2004; revised June 2, 2005, January 12, 2006, July 8, 2006, and October 27, 2006. The review of this paper was coordinated by Prof. E. Sourour.

R. Y. M. Wong is with Redknee Inc., Mississauga, ON, L4W 4Y9 Canada.

R. Adve is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada.

Digital Object Identifier 10.1109/TVT.2007.898398

To overcome the drawbacks of the OMMSE processor, researchers have proposed suboptimal schemes with fewer adaptive unknowns. There are now several such "reduced-rank" schemes. Principal components with generalized sidelobe cancellation [3] and cross-spectral metric [4], [5] are reduced-rank filtering schemes based on an eigendecomposition of the interference covariance matrix. These filters reduce the number of adaptive weights by projecting the received signal onto a lower dimensional signal subspace. However, the disadvantage of these filters is the high computational load of the eigendecomposition. The recently introduced auxiliary vector (AV) filter [6], [7] does not involve any eigendecomposition or matrix inversion. It generates a sequence of linear filters that converges to the OMMSE filter. Recent work has solved the problem of choosing the optimal sequence length, which makes the algorithm robust to limited training [8]. The authors show that the revised AV scheme outperforms the popular multistage-Weiner-filter (MSWF) algorithm with limited training. The algorithm presented here is an alternative approach with low computation load that also requires only limited training.

Other popular reduced-rank algorithms are the iterative constrained OMMSE (COMMSE) processor [2] and the MSWF [9], [10]. The MSWF obtains the adaptive weights by a multistage decomposition. It provides satisfactory results but with relatively high complexity and slow convergence rate in terms of required training. With iterative cascaded spatial and temporal processing, the COMMSE filter only yields additive gains, while the OMMSE processor yields multiplicative gains. Clearly, a data-efficient joint-domain processor with reduced computational load and near-optimal performance would be a significant advance over the state of the art.

This paper introduces joint-domain-localized (JDL) adaptive processing for uplink CDMA systems with low computational load and fast convergence rate in terms of the required training. The algorithm draws inspiration from an efficient joint-domain technique developed for radar systems [11], [12]. Some of the terminology used in this paper is drawn from this literature. The JDL algorithm adaptively processes the spatio-temporal data in a localized region after matched filtering to obtain data in a "beamspace," although this beamspace does not have the conventional physical interpretation in CDMA systems. Localization, here, refers to choosing a subset of users for adaptive processing after the matched filter. This paper states the optimum choice, in terms of MSE, of this subset and justifies two significantly simpler suboptimal choices. This is an important consideration in implementing the proposed scheme. As with other schemes [1], [2], [6], the algorithm is based on assumed knowledge of the spatio-temporal signatures of all users.

Simulations show that, given limited training, the JDL algorithm performs better than the "fully optimal" MMSE, MSWF, and COMMSE processors with significantly lower computational load.

Building on the JDL algorithm, we introduce a new algorithm, designated JDL-Z, which integrates the JDL algorithm with zero forcing for multicell CDMA environments with significant intercell interference. The JDL-Z algorithm assumes knowledge of channels and spreading codes of intracell users only. The numerical examples show that the JDL-Z algorithm achieves better performance than the JDL algorithm with faster convergence rate. However, this is at the cost of significantly higher computational load.

In summary, the contributions of this paper are as follows:

1) design of a low-computation-load algorithm that adaptively processes received signals after selective matched filtering;
2) an effective selection process with low complexity that plays an important role in the success of the algorithm;
3) extension of the algorithm to replace the matched filter with an adaptive first stage for scenarios with significant intercell interference.

Before introducing the JDL and JDL-Z algorithms, Section II presents the model for an uplink direct-sequence CDMA (DS-CDMA) system with a receive-antenna array. Section III describes the JDL algorithm, followed by numerical examples illustrating its performance in Section IV. Section V presents the JDL-Z algorithm, including examples illustrating its performance. This section also provides a complexity analysis of JDL and JDL-Z algorithms and compares these complexities to the MSWF and COMMSE reduced-rank processors. Section VI ends this paper with some conclusions.

## II. System Model

Consider a synchronous CDMA uplink cellular system with $N$ receive antennas and $K$ users, of which $M$ users are intracell (within the cell of interest), and $P$ users are intercell (outside the cell of interest). Each user is assigned a random binary short-spreading code with processing gain $G$ and transmits using a single antenna. Assuming the channels are slow and flat, the uplink signal within a single-symbol period $0 < t \leq T_s$ at the receive-antenna array is a length-$N$ vector, which is given by

$$\mathbf{x}(t) = \sum_{k=1}^{K}\sum_{j=0}^{G-1} a_k b_k s_k^j \psi(t - jT_c)\mathbf{h}_k + \mathbf{n}(t) \qquad (1)$$

where $a_k$, $b_k$, and $\mathbf{h}_k$ are the received amplitude, data symbol, and channel vector, of length-$N$, of user $k$, respectively. The sequence $\{s_k^j = \pm 1, j = 0, \ldots, G-1\}$ represents the length-$G$ spreading code $\mathbf{s}_k$ of user $k$ with chip waveform $\psi(t)$ and lasting chip period $T_c$. Both temporal and spatial signatures of the users are assumed to have unit energy, i.e., $\int_0^{T_c} |\psi(t)|^2 dt = 1/G$ and $E\{\mathbf{h}_k^{\mathrm{H}}\mathbf{h}_k\} = 1$ for $\forall k$, where $^{\mathrm{H}}$ denotes the Hermitian and $E\{\cdot\}$ the statistical expectation operator. The receiver noise $\mathbf{n}(t)$ is modeled as white and Gaussian.

After matched filtering to the chip waveform, the received spatio-temporal data signal at the base station for a symbol period can be represented as a length-$NG$ vector $\mathbf{x}$, given by

$$\mathbf{x} = \sum_{k=1}^{K} a_k b_k (\mathbf{s}_k \otimes \mathbf{h}_k) + \mathbf{n} \qquad (2)$$

$$= \sum_{k=1}^{K} a_k b_k \mathbf{z}_k + \mathbf{n} \qquad (3)$$

where $\otimes$ denotes the Kronecker product, and $\mathbf{s}_k = [s_k^0, s_k^1, \ldots, s_k^{G-1}]^{\mathrm{T}}$ is the temporal signature of user $k$, making $\mathbf{z}_k = \mathbf{s}_k \otimes \mathbf{h}_k$ the spatio-temporal channel (spatio-temporal signature) of user $k$.

In this paper, the processor is assumed to have knowledge of the spatial and temporal channels $\mathbf{h}_k$ and $\mathbf{s}_k$ of all $M$ intracell users but no knowledge of the channels of the $P$ intercell (interfering) users.

It is worth justifying the data model in (1) as future generations of wireless communication systems must deal with multipath fading, which would significantly complicate the channel-estimation process (number of paths, their amplitudes and delays). Channel estimation is, however, beyond the scope of this paper. Here, we assume that the channels of all intercell users are known. In the case of multipath, each path would be treated a separate known user (to be combined later in a RAKE receiver) with all other parts treated as additional interference. Furthermore, a popular approach in dealing with multipath channels is to use orthogonal frequency-division multiplexing. In this case, the above data model can be thought of as being applied to each subcarrier. Multipath channels are, however, beyond the scope of this paper and may be considered in future work.

## III. JDL Processing

The JDL algorithm is a two-stage beamspace scheme. The first stage transforms the spatio-temporal received signal $\mathbf{x}$ into spatio-temporal "beamspace" by correlating the received signals with $\eta$-selected spatio-temporal "beams" (effectively matched filtering with $\eta$ spatio-temporal channels). The physical interpretation of this beamspace deserves some consideration. In line-of-sight (LOS) channels, there is a one-to-one correspondence, captured by the steering vector, between the signal emanating from a specific angle and the signals received at the antenna array. The steering vector is the spatial signature associated with the user from that angle. Clearly, in fading channels, this notion of a steering vector does not hold. On the other hand, we know that in LOS channels, the beamspace is formed with an inner product with the steering vector. Therefore, while the physical interpretation of beamspace does not hold, we use here an inner product with the spatio-temporal signature of each user $\mathbf{z}_k$ for user $k$.

For each desired user, $\eta$ spatio-temporal "beams" are formed by correlating the received signal with the spatio-temporal signatures of $\eta$ users. The crucial issue of how to choose these users is left for Section III-A. This first stage provides some interference suppression by decorrelating the signals of the interferers. The beamspace data can be obtained using a transformation matrix $\mathbf{T}$. The transformation process is given by

$$\tilde{\mathbf{x}} = \mathbf{T}^{\mathrm{H}}\mathbf{x} \tag{4}$$

where the tilde ( $\tilde{}$ ) denotes the beamspace domain, and $\mathbf{T}$ is the $NM \times \eta$ transformation matrix. In [11], this set of data in the beamspace is called the localized processing region (LPR). This term arises because the postmatched-filter data chosen for further processing are closest (in terms of cross correlation) to that of the desired user. The size of the LPR, which is $\eta$, is an implementation issue and represents a tradeoff between performance, required training, and computational load. The details regarding choosing an optimal $\eta$ are presented in Section III-B.

Having obtained beamspace data, the second stage of processing suppresses residual interference and noise. The localized beamspace data $\tilde{\mathbf{x}}$ are adaptively combined, in terms of MMSE, to produce a soft-decision statistic for the data symbol. The MMSE weights are found in beamspace by [13]

$$\tilde{\mathbf{w}} = \tilde{\mathbf{R}}^{-1}\tilde{\mathbf{v}} \tag{5}$$

$$\tilde{\mathbf{R}} = E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathrm{H}}\} \tag{6}$$

$$\tilde{\mathbf{v}} = E\{\tilde{\mathbf{x}}d^*\} \tag{7}$$

where $^*$ denotes complex conjugation, and $d$ is the desired information symbol.

In practice, the MMSE weights may be obtained using the sample matrix inversion (SMI) by finding the sample-averaged estimate $\tilde{\mathbf{R}}$ or using a training scheme, such as least mean squares (LMS) or recursive least squares [13]. The weights could also be estimated using a blind scheme, since the transformation process does not require training data and is known to the receiver. A blind scheme, which is based on data correlation, would use $\tilde{\mathbf{R}}$ instead of the original $\mathbf{R} = E\{\mathbf{x}\mathbf{x}^{\mathrm{H}}\}$. Similarly, a blind scheme, which is based on the channel-code combination $\mathbf{z}$, would use $\tilde{\mathbf{z}} = \mathbf{T}^{\mathrm{H}}\mathbf{z}$ instead. However, here we focus on training-based schemes. Given $L$ training symbols, the SMI weights are given by

$$\tilde{\mathbf{w}}_{\mathrm{SMI}} = \tilde{\mathbf{R}}_{\mathrm{SMI}}^{-1}\tilde{\mathbf{v}}_{\mathrm{SMI}} \tag{8}$$

$$\tilde{\mathbf{R}}_{\mathrm{SMI}} = \frac{1}{L}\sum_{l=1}^{L}\tilde{\mathbf{x}}_l\tilde{\mathbf{x}}_l^{\mathrm{H}} \tag{9}$$

$$\tilde{\mathbf{v}}_{\mathrm{SMI}} = \frac{1}{L}\sum_{l=1}^{L}\tilde{\mathbf{x}}_l d_l^* \tag{10}$$

where $d_l$ is the $l$th training symbol, and $\mathbf{x}_l$ is the received signal over the corresponding symbol period. This is the traditional MMSE process, with the beamspace data replacing the usual spatio-temporal data. The computation load per user is reduced from finding $NG$ adaptive weights to $\eta$ adaptive weights. The issue of computation load is explored further in Section III-D. The reduction in number of adaptive weights also results in significant gains in required sample support. As shown in [14], estimation of high dimensional correlation matrices with limited-sample support results in significant loss in terms of signal-to-interference-plus-noise ratio (SINR) and bit-error rate (BER). Similarly, reducing the number of unknowns results in corresponding gains in terms of required training in iterative schemes, such as LMS [15].

The final soft-decision statistic to estimate the transmitted symbol is

$$y = \tilde{\mathbf{w}}^{\mathrm{H}}\tilde{\mathbf{x}} = [\mathbf{T}\tilde{\mathbf{w}}]^{\mathrm{H}}\mathbf{x}. \tag{11}$$

The JDL adaptive process therefore results in equivalent spatio-temporal weights given by $\mathbf{w} = \mathbf{T}\tilde{\mathbf{w}}$.

The JDL algorithm is, therefore, a two-stage process for each user of interest. For each user, the other $K-1$ users act as interference. The first stage, given by (4), represents matched filtering with a limited set of available spatio-temporal channels. The columns of the transformation matrix $\mathbf{T}$ are the channels of $\eta$ users. The key remaining question, which is addressed in Section III-A, is the choice of the $\eta$ users.

In the second stage, which is summarized by (5)–(7), interference is further reduced using an MMSE criterion. In practice, this second stage could be implemented using (8)–(10) or an iterative algorithm such as LMS. The number of adaptive unknowns is reduced to $\eta$. By carefully choosing the transformation matrix in the first stage, this two-stage process still results in effective interference suppression.

Another interpretation of the JDL algorithm comes from the recognition that choosing $\eta = K$ makes JDL equivalent to the postmatched filter, optimal, MMSE multiuser detection scheme when all users' spreading codes are known [16]. In this regard, the JDL algorithm may be interpreted as the postmatched-filter processor, however, after spatio-temporal matching with only a subset of (carefully chosen) users. The other users that do not contribute to the transformation matrix are effectively treated as additional noise. The two-stage interpretation, however, allows for the extension to the JDL-Z algorithm of Section V, where the nonadaptive first-stage transformation is replaced by an adaptive zero-forcing transformation, although potentially by any transformation of interest.

### A. *Choice of Transformation Matrix*

A central issue in the performance of the JDL algorithm is the choice of the transformation matrix in (4), specifically the choice of the users whose channels make up the columns of the matrix. This section addresses this important issue of presenting the optimal but impractical choice and then justifying a simpler choice with low associated computation load. We first introduce the optimal choice of transformation matrix based on an MMSE criterion, followed by two suboptimal methods with lower computation load.

*1) Optimal Method Based on MMSE:* The optimal choice of $\eta$ spatio-temporal channels would minimize the overall MSE. Since, using (5), [4]

$$\begin{aligned}\mathrm{MSE} &= 1 - \tilde{\mathbf{w}}^{\mathrm{H}}\tilde{\mathbf{R}}\tilde{\mathbf{w}} \\ &= 1 - \tilde{\mathbf{v}}^{\mathrm{H}}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{v}}\end{aligned} \tag{12}$$

the $\eta$ spatio-temporal channels should be chosen, such that $\tilde{\mathbf{v}}^{\mathrm{H}}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{v}}$ is maximized. However, this would entail evaluating this term for all ${}^{K}C_{\eta} = K!/(K-\eta)!\eta!$ possible combinations of $\eta$ spatio-temporal channels. This optimal method is clearly prohibitively expensive computationally and impractical.

*2) Suboptimal Method Based on MMSE:* To overcome the problem with computation load associated with the optimal approach, we introduce an MMSE-based recursive suboptimal method, which has a lower computational load. The first column of $\mathbf{T}$ is chosen by a direct evaluation of $\tilde{\mathbf{v}}^{\mathrm{H}}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{v}}$ obtained after evaluating the term for all $M$ possible intracell candidates. Since this corresponds to $\eta = 1$, $\tilde{\mathbf{R}}$ is only a number. The other $(\eta - 1)$ users are chosen one at a time recursively, based on the previously chosen channels to minimize MSE at each choice. Without loss of generality, user 1 is the desired user.

*Proposition:* At the $(n+1)$th choice, given $\mathbf{T}_n = [\mathbf{t}_1, \ldots, \mathbf{t}_n]$, the best choice of $\mathbf{t}_{n+1}$ to form $\mathbf{T}_{n+1} = [\mathbf{T}_n \ \mathbf{t}_{n+1}]$ in the MMSE sense is given by

$$\mathbf{t}_{n+1} = \arg\max_{\mathbf{z}_i} \frac{\left|\tilde{\mathbf{z}}_1^{\mathrm{H}}\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i - \zeta_{1,i}\right|^2}{\mathbf{z}_i^{\mathrm{H}}\left(\mathbf{R} - \mathbf{R}\mathbf{T}_n\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\right)\mathbf{z}_i} \tag{13}$$

where $\mathbf{z}_i$ is not included in $\mathbf{T}_n$ to avoid making $\tilde{\mathbf{R}}_{n+1}$ singular. $\tilde{\mathbf{R}}_n = \mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{T}_n$ and $\tilde{\mathbf{z}}_1 = \mathbf{T}_n^{\mathrm{H}}\mathbf{z}_1$ are, respectively, the autocorrelation matrix and channel of the desired user in the transform domain with LPR size of $n$. $\zeta_{i,j} = \mathbf{z}_i^{\mathrm{H}}\mathbf{z}_j$ represents the cross correlation between users $i$ and $j$.

*Proof:* See Appendix A. ■

The $\eta$ spatio-temporal channels that form the columns of the transformation matrix $\mathbf{T}$ are, therefore, found by the following recursive scheme.

Step 1) Initialization:

$$\mathbf{t}_1 = \arg\max_{\mathbf{z}_i} \ \tilde{\mathbf{v}}^{\mathrm{H}}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{v}}, \qquad 1 \le i \le K \tag{14}$$

where $(\tilde{\ })$ represents the transform domain with $\mathbf{T} = \mathbf{z}_i$. At this stage, $\tilde{\mathbf{R}}$ is a single number.

Step 2) For $n = 1, \ldots, \eta - 1$, set $\mathbf{T}_{n+1} = [\mathbf{T}_n \, \mathbf{t}_{n+1}]$, where $\mathbf{t}_{n+1}$ is chosen using the criterion in (13).

Although suboptimal, the computational advantage of this iterative method over the optimal method is clear. This iterative approach requires only $(\eta - 1)$ as opposed to ${}^{K}C_{\eta} = K!/(K-\eta)!\eta!$ matrix inversions.

*3) Suboptimal Method Using Correlation:* An even simpler choice for the users' channels, making up the columns of the transformation matrix, is based on the idea that the MMSE process, in the second stage, should focus on suppressing the most interfering users. We define the amplitude-weighted cross correlation

$$\rho_{i,j} = a_i^* a_j \mathbf{z}_i^{\mathrm{H}}\mathbf{z}_j. \tag{15}$$

The chosen columns of the transformation matrix are the desired user's channel and the channels of the $(\eta - 1)$ users with the largest amplitude-weighted cross correlation. Note that this approach assumes knowledge of the users' amplitudes and channels. If amplitudes are unavailable, the cross correlation $\zeta_{i,j}$ may be used. Clearly, this approach has extremely low computation load, even when compared to the recursive approach aforementioned.

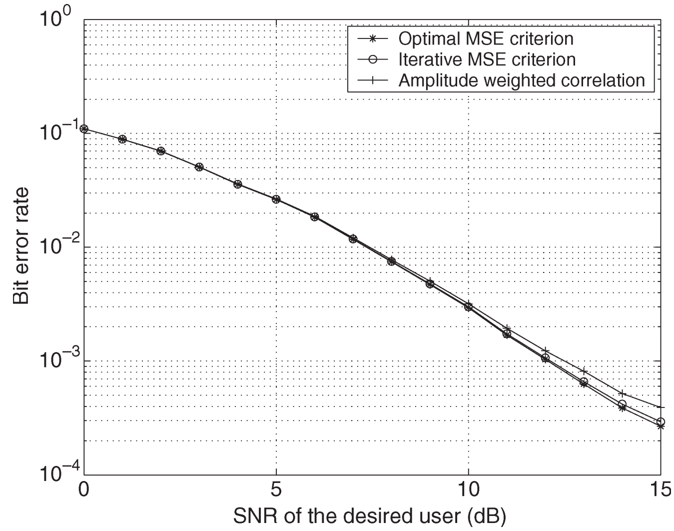


Fig. 1. Performance comparison of using different choices of transformation matrices.

There are, therefore, three proposed choices of the transformation matrix $\mathbf{T}$. With decreasing computation load, the matrix can be chosen optimally using the technique in Section III-A1 or, suboptimally, using either (13) or (15). Fig. 1 plots the BER versus signal-to-noise ratio (SNR) using these three constructions of $\mathbf{T}$. The system uses $N = 4$ receive antennas, a processing gain of $G = 8$ with $K = 8$ equal power users. Each user's transmission uses an independent and identically distributed (i.i.d.) uncorrelated Rayleigh fading channel. Ideal weights are used by using the true values of the autocorrelation matrix $\tilde{\mathbf{R}}$ and cross-correlational vector $\tilde{\mathbf{v}}$. The LPR size $\eta$ is fixed at five. The figure shows that using the most amplitude-weighted cross-correlated channels is a good choice for $\mathbf{T}$ because it achieves near-optimal performance with a very low computation load. For the rest of this paper, we, therefore, use this criterion to choose the users whose channels form the transformation matrix $\mathbf{T}$.

### B. Optimal Size of LPR in JDL

If the ideal JDL weights are obtained using the true autocorrelation matrix $\tilde{\mathbf{R}}$ and cross-correlational vector $\tilde{\mathbf{v}}$, then performance improves with increasing LPR size up to $\eta = K$. In practice, the adaptive weights must be estimated, possibly using training as in (8)–(10). This section illustrates the tradeoff between performance and required training as related to the choice of $\eta$.

Given a fixed number of training symbols $L$, the ideal choice of $\eta$ would minimize the output MSE

$$\mathrm{MSE}(\eta, L) = E\left\{\left|\tilde{\mathbf{w}}_{\mathrm{SMI}}(\eta, L)^{\mathrm{H}}\tilde{\mathbf{x}}(\eta) - d\right|^2\right\}. \tag{16}$$

Unfortunately, this MSE must be estimated for all feasible values of $\eta$ and $L$, which is impossible in real time.

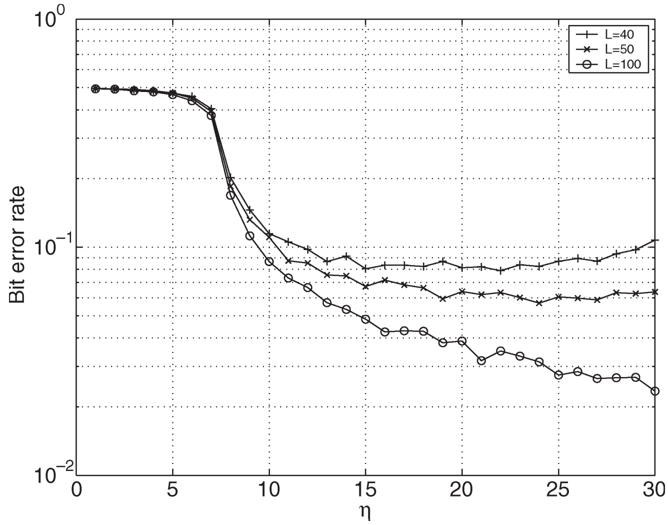


Fig. 2. BER of JDL algorithm using SMI with $L = 40, 50,$ and 100 training bits, $N = 4$, $K = 30$, and $G = 12$.

Fig. 2 illustrates the performance of the JDL algorithm with varying $\eta$. SMI is used to estimate the JDL weights with $L = 40, 50,$ and 80 training bits. The single-cell DS-CDMA uplink system has $N = 4$ receive antennas, $K = 30$ users, and processing gain $G = 12$. The SNR of the desired user and 22 interferers is set at 10 dB. To simulate a severe near–far situation, the SNR of seven interferers is fixed at 40 dB above the other users. This case is chosen to illustrate a drawback of the JDL algorithm—the fact that the LPR must account for all strong interferers. BPSK is used for data modulation. The BER is the average over 1000 simulations, in which the uncorrelated Rayleigh channels are kept constant for 1000 bits.

For all three curves, the SER $\approx 0.5$ for $\eta \leq 7$, which is the number of strong interferers, i.e., at minimum, the processing region must cover the most relevant spatio-temporal beamspace for processing. As expected, given a relatively large number of training symbols ($L = 100$), the error rate falls with growing $\eta$. However, even for this case, the error rate is not very sensitive to the choice of $\eta$. Given limited training ($L = 40$ or 50), the performance actually worsens for large $\eta$. This is consistent with the results in [14]. With limited training and large $\eta$, it is not possible to estimate the filter coefficients accurately. With small $\eta$, there are insufficient degrees of freedom to suppress interference.

### C. Implementation Details

This section presents a methodology to implement the JDL algorithm developed. Given the receive data $\mathbf{x}$ and a choice of $\eta$, the steps in the implementation are as follows.

1) For each of the $K$ users, form $\tilde{x}_k = \mathbf{z}_k^{\mathrm{H}}\mathbf{x}$, which is the matched-filter estimate of the user's signal.
2) Designate user 1 to be the user of interest. Choose $(\eta - 1)$ "most interfering users" users with the greatest amplitude-weighted cross correlation using (15).
3) Form the length-$\eta$ vector $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_{(2)}, \ldots, \tilde{x}_{(\eta)}]^{\mathrm{T}}$ using the matched-filter estimates of the desired user and the $(\eta - 1)$ users chosen in step 2).

TABLE I
PER-USER COMPUTATIONAL COMPLEXITY ANALYSIS OF JDL

| JDL algorithm step | Complexity |
|---|---|
| Matched filtering | $\mathcal{O}[NG]$ |
| Amplitude weighted cross-correlation | $\mathcal{O}[(K-1)NG/2]$ |
| Weight calculation | $\eta^3$ |

4) Estimate the adaptive weights $\tilde{\mathbf{w}}$ in MMSE sense using (8)–(10) or any other approach of choice (e.g., LMS, RLS, etc.).
5) Apply the weights to obtain a soft-decision statistic $y = \tilde{\mathbf{w}}^{\mathrm{H}}\tilde{\mathbf{x}}$ for further processing.
6) Repeat steps 2)–5) for all other users of interest. Note that the choice of the $(\eta - 1)$ most interfering users in step 2) changes for each user of interest.

### D. Complexity Analysis

Compared to the optimal processor, the computational advantage of the JDL algorithm is clear. In the general case of multicell CDMA, the OMMSE processor requires $NG$ adaptive weights. Solving the size-$NG$ linear system of equations, results in complexity $\mathcal{O}[(NG)^3]$ (here, $\mathcal{O}$ is used to indicate complexity "on the order of"). Only in the special case of a single-cell system, with $K$-known spatio-temporal channels, can this OMMSE processor be implemented after matched filtering with all users, resulting in complexity $\mathcal{O}[K^3]$.

The analysis for the JDL algorithm assumes the likely scenario that the receiver will eventually decode all users' signals. The matched-filter process is then $\mathcal{O}[NG]$ per user. Obtaining the amplitude-weighted cross correlation requires $\mathcal{O}[K(K-1)NG/2]$ computations overall, i.e., $\mathcal{O}[(K-1)NG/2]$ per user. Finally, obtaining the weights is an $\mathcal{O}[\eta^3]$ process. The complexities of the individual steps are summarized in Table I. The overall complexity is, therefore, $\mathcal{O}[\eta^3 + (K+1)NG/2]$.

The computational complexities of other reduced-rank approaches are significantly higher than that of JDL. The MSWF [9] with $D$ stages has complexity $\mathcal{O}[D(NG)^2]$ per user. The COMMSE [2] processor with $Q$ iterations to convergence has complexity $\mathcal{O}[Q(KN^2 + KG^2 + N^3 + G^3)]$ per user. The proof of the complexities of MSWF and COMMSE are presented in Appendices B and C, respectively.

In summary, as compared to the optimal algorithm and some other suboptimal reduced-rank algorithms, the JDL algorithm has significantly lower computational complexity. The simulations in the next section show that this lower complexity does not result in degraded performance. In practical situations with relatively few training symbols, the performance is superior to the optimal and other reduced-rank approaches. The examples also illustrate the flexibility provided by the choice of $\eta$.

## IV. SIMULATION RESULTS

This section presents results of simulations to illustrate the performance of the JDL algorithm as compared to the OMMSE, COMMSE [2], and MSWF [9] algorithms. The system is

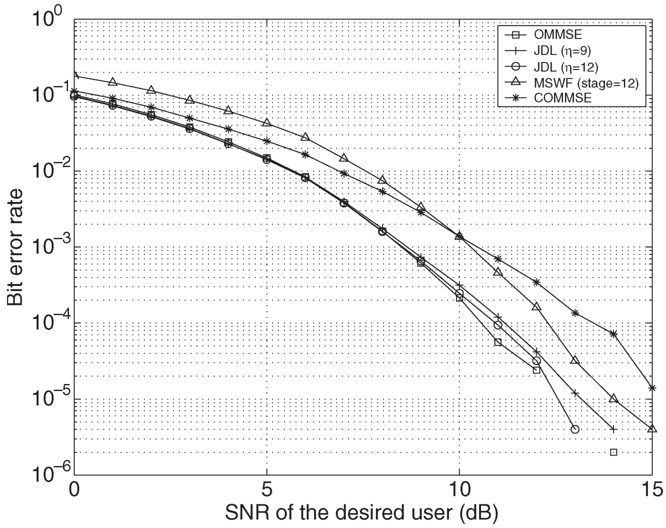


Fig. 3. Performance comparison of joint-domain filters, $N = 11$, $K = 20$, and $G = 16$ with 500 training bits.

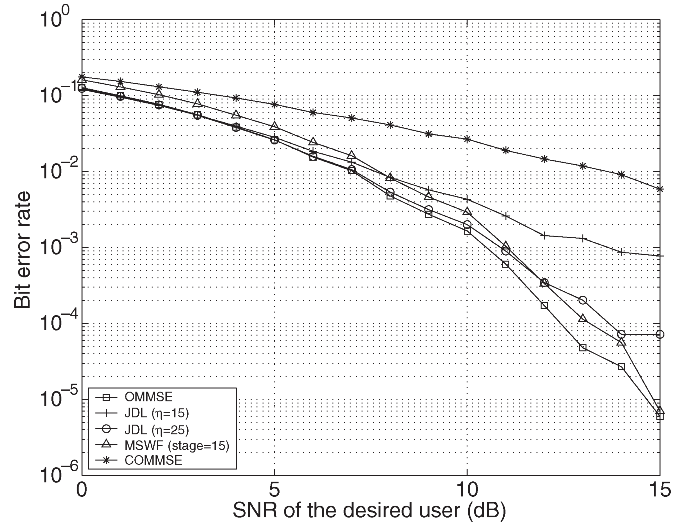


Fig. 4. Performance comparison of joint-domain filters, $N = 8$, $G = 12$, and $K = 30$ with 500 training bits.

modeled as a single-cell CDMA system with slow flat i.i.d. Rayleigh fading. All users' channels are assumed to be known. The OMMSE algorithm is therefore implemented as a postmatched-filter multiuser detector with $K$ adaptive weights. The channels are modeled as flat and fixed over the time period used to estimate the autocorrelation matrix $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{v}}$. In all examples, each user is assigned a random binary spreading code. The signal is sampled at the chip rate. In this example, BPSK is used for data modulation, and the algorithms are compared in terms of BER. Clearly, with the increasing data rates in modern communication systems, the choice of a flat channel is losing validity. However, since the users' are all assigned random codes, a multipath channel could be treated as separate users whose signals are to be combined in a RAKE receiver, or the multiple paths can be treated as additional interference.

The first example compares the JDL and other joint-domain algorithms in a lightly loaded system with power control. As expected, all algorithms work fairly well. However, a crucial feature of the JDL algorithm is its effectiveness in near–far situations. The next example uses a heavily loaded situation with a large number of training bits with a significant near–far problem. The third example presents the performance advantage of the JDL algorithm with limited training in highly loaded situations. In all three examples, the adaptive weights are estimated using SMI. An example illustrating the convergence rates of the JDL algorithm is delayed to Section V-D, including the extended JDL-Z algorithm.

### A. Lightly Loaded System With Power Control

This example is based on a DS-CDMA system with $N = 11$ receive antennas and a spreading gain of $G = 16$ serving $K = 20$ users. The system uses power control, i.e., all users' signals arrive at the receiver with the same power. Fig. 3 compares the performance of the JDL algorithm with $\eta = 9$ and 12, with the OMMSE, MSWF, and COMMSE algorithms. A large number of training symbols $(L = 500)$ is used to obtain weights close to their theoretical values. As expected, in a relatively stress-free scenario, all algorithms provide good performance with the OMMSE algorithm providing the best. Note that the JDL algorithm with $\eta = 9$ (which is less than half of $K$) or $\eta = 12$ provides near-optimal performance, emphasizing the computation advantages of the JDL algorithm. Note also the improvement in performance over the MSWF and COMMSE algorithms.

### B. Heavily Loaded System in a Near–Far Scenario

This example uses a DS-CDMA system with $N = 8$ receive antennas and a processing gain $G = 12$ supporting $K = 30$ users. All users are equal in power, except for four interferers at 40 dB above the desired user representing a severe near–far effect. This example uses $L = 500$ training bits, making the estimated weights very close to their ideal values. Fig. 4 compares the BER of two implementations of JDL ($\eta = 15$ and 25), OMMSE, MSWF $(D = 15)$, and COMMSE algorithms. The figure shows that, as expected, OMMSE always outperforms other algorithms. However, with reduced computational load, JDL processing provides good performance. Increasing the LPR size significantly improves performance but at the cost of higher computational load. In a single-cell system, the JDL algorithm with $\eta = K = 30$ achieves optimal performance. The JDL algorithm, with lower computational load, performs better than the COMMSE processor, while at the expense of higher computational load, the MSWF algorithm results in near-optimal performance. However, as shown in the next example, which is limited to a small number of training bits, JDL performs better than the MSWF algorithm while retaining lower computational complexity.

### C. Heavily Loaded System With Limited Training

When limited by the number of training bits, the JDL algorithm can even outperform the theoretically fully OMMSE algorithm. This example uses the same setting as the earlier example, with the exception that only 40 training bits are used.

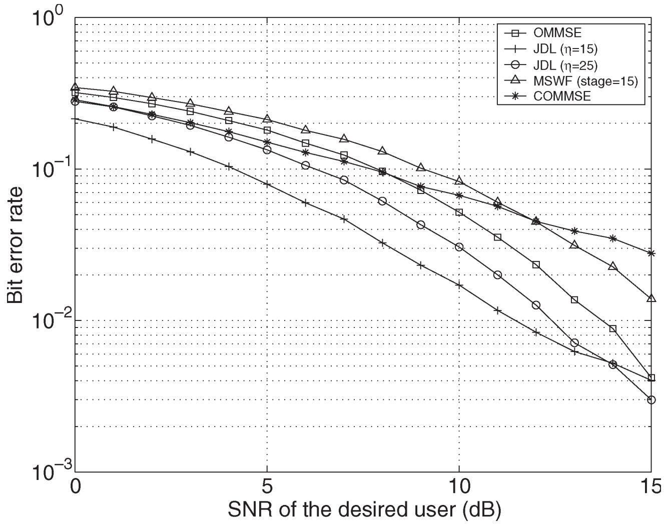


Fig. 5. Heavily loaded system with 40 training bits, $N = 8$, $G = 12$, and $K = 30$.

Fig. 5 compares the performance of the JDL ($\eta = 15$ and $\eta = 25$), OMMSE, MSWF $(D = 15)$, and COMMSE algorithms. Fig. 5 shows that the JDL algorithm achieves better performance than the other processors over a wide range in SNR. In fact, due to the limited training, choosing $\eta = 15$ results in better performance than choosing $\eta = 25$. Again, this is consistent with the results in [14].

In summary, the JDL processor described above processes signals after matched filtering to their spatio-temporal signatures. The term "localized processing" arises from use of only $\eta$ matched-filtered signals for further processing. The suboptimal choice of $(\eta - 1)$ "most interfering users" works very well with limited complexity. Note that, other than the choice of users, this matching process is nonadaptive. The next section details an interesting extension of the JDL concept with an adaptive transform to beamspace.

## V. JDL Processing Using Zero Forcing

The JDL with the zero-forcing algorithm is based on the realization that matched filtering obtains a nonadaptive estimate of the desired signal. The adaptive second stage (MMSE) combines $\eta$ such nonadaptive estimates to further suppress residual interference. In this section, we replace the nonadaptive first stage with an adaptive processor at the cost of higher computational complexity. Here, the first-stage matched filter is replaced by a zero-forcing receiver [13]. The choice of zero forcing in the first stage is just one of many possibilities but serves to illustrate the concept of following adaptive processing with another stage of adaptive "beamspace" processing to suppress residual interference. The key contribution here, therefore, is the framework created for two-stage adaptive processing. Section V-D presents an example of using the MSWF algorithm instead of zero forcing in the first stage.

### *A. Formulation*

The JDL-Z algorithm is presented here in terms of a single desired user. As in (1), the received data in a multicell CDMA system is the sum of the signals of $K$ users, of which $M$ are intracell users. Only the channels of the $M$ intracell users are assumed to be known. The transformation matrix $\mathbf{T}$ is constructed using $\eta$ zero-forcing weight vectors, as opposed to the channels of $\eta$ users. In the second stage, $\eta$ beamspace data samples are adaptively combined in the sense of MMSE. As with JDL, the transformation matrix is constructed on the basis of the $\eta$ most correlated spatio-temporal channels.

The transformation matrix $\mathbf{T}$ consists of $\eta$ zero-forcing weight vectors. For example, if $\eta = 3$, we assume that $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$ are the desired and the $(\eta - 1)$ most correlated intracell spatio-temporal channels. In this case

$$\mathbf{T} = [\mathbf{t}_1 \ \mathbf{t}_2 \ \mathbf{t}_3] \tag{17}$$

where

$$\mathbf{Z}_{1:M}^{\mathrm{H}} \mathbf{t}_i = \mathbf{e}_i, \qquad i = 1, 2, 3. \tag{18}$$

The matrix $\mathbf{Z}_{1:M} = [\mathbf{z}_1 \ \mathbf{z}_2, \ldots, \mathbf{z}_M]$ contains the channels of all $M$ intracell users, and $\mathbf{e}_i$ is a length-$M$ vector of zeros with a single one in the $i$th position. The weights $\mathbf{t}_i$ have unity response to the $i$th channel and null out all other users' channels. This stage, therefore, eliminates all intracell interference.

As in (4) with the original JDL algorithm, the transformed received signal $\tilde{\mathbf{x}}$ is defined as

$$\tilde{\mathbf{x}} = \mathbf{T}^{\mathrm{H}} \mathbf{x}. \tag{19}$$

The second stage of the JDL-Z algorithm, which suppresses residual interference and noise, is the same as that in the original JDL algorithm. The MMSE weights $\tilde{\mathbf{w}}$ in zero-forced beamspace are obtained by training, using (8)–(10), with the final soft-decision statistic given by (11). Since the receiver does not possess information of the intercell users, the MMSE weights can only be estimated using SMI or using an iterative scheme, such as the LMS algorithm. The second MMSE stage also eliminates the problem of noise enhancement associated with zero forcing.

As described above, it appears that the JDL-Z algorithm is extremely complex, requiring $\eta$ zero-forcing solutions per user. However, since, in any reasonable system, all intracells users are desired, zero forcing can first be executed for all $M$ intracell users. The second stage of JDL can then be implemented by choosing $\eta$ of these $M$ users for further processing. Furthermore, we emphasize that zero forcing is just one choice of adaptive processing in the first stage. The additional complexity due to the JDL MMSE stage is, therefore, restricted in estimating an $\eta \times \eta$ matrix and in solving the corresponding equation.

### *B. Optimal Size of LPR in JDL-Z*

The JDL-Z weights can only be estimated because of the unknown intercell interferers. This section shows that, in terms of performance, the JDL-Z algorithm has characteristics similar to the original JDL algorithm. As with JDL, the optimal choice of LPR is prohibitively expensive to compute, and we illustrate the choice using a numerical example.

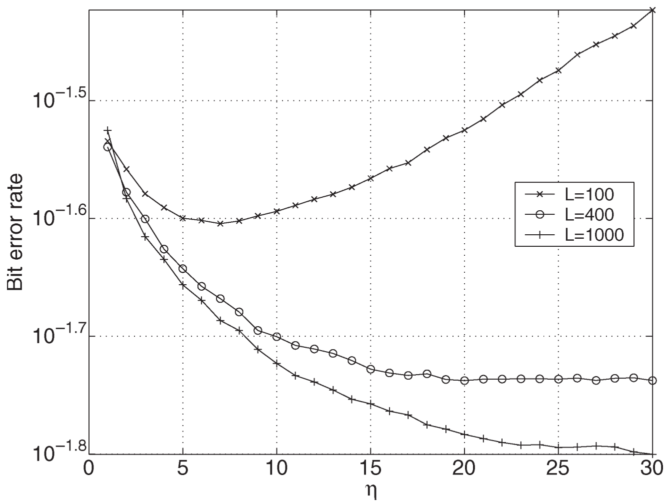


Fig. 6. BER of JDL-Z algorithm using SMI with 100, 400, and 1000 training bits.

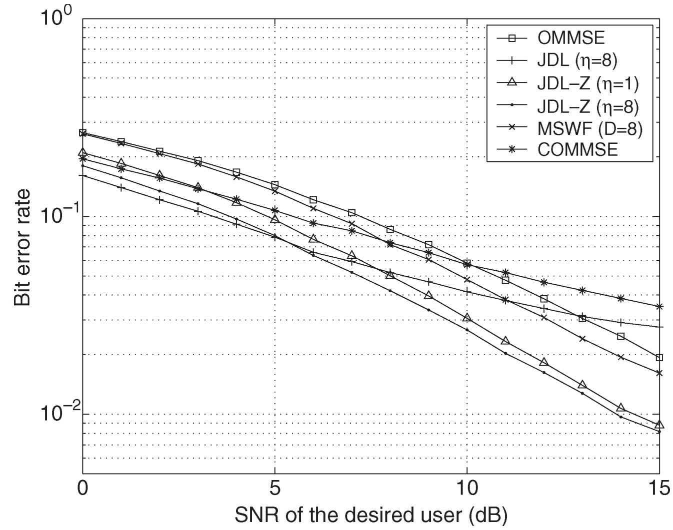


Fig. 7. Performance comparison of joint-domain filters in multicell environments, $N = 4$, $M = 30$, $P = 80$, and $G = 12$.

Consider a multicell DS-CDMA uplink system consisting of $N = 4$ receive antennas, $M = 30$ intracell and $P = 80$ intercell users, and processing gain $G = 12$. The SNR of the intracell users is set to 10 dB. There are 30 and 50 intercell interferers with SNR of $-2$ and $-10$ dB, respectively. The performance of JDL-Z with varying $\eta$ is shown in Fig. 6. The estimated weights are obtained by SMI using $L = 100, 400,$ and $1000$ training bits. Fig. 6 shows that, as expected, the performance of the JDL-Z algorithm is sensitive to $\eta$, as well as the number of training bits. One can find the optimal rank from the graph, however, as expected, with a large number of training bits; the performance improves with larger $\eta$. The figure also shows that using a second stage of processing potentially halves the BER—the case of $\eta = 1$ in the figure corresponds to using zero forcing only.

## C. Implementation Details

This section presents a scheme to implement the JDL-Z algorithm developed. The implementation closely follows the implementation of the JDL algorithm presented in Section III-C. Given the receive data $\mathbf{x}$ and a choice of $\eta$, the steps in the implementation are as follows.

1) For each of the $K$ users, form $\tilde{x}_k = \mathbf{t}_k^{\mathrm{H}}\mathbf{x}$, which is the zero-forcing estimate of the user's signal.
2) Designate user 1 to be the user of interest. Choose $(\eta - 1)$ "most interfering users" with the greatest amplitude-weighted cross correlation using (15).
3) Form the length-$\eta$ vector $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_{(2)}, \ldots, \tilde{x}_{(\eta)}]^{\mathrm{T}}$ using the zero-forcing estimates of the desired user and the $(\eta - 1)$ users chosen in step 2).
4) Estimate the adaptive weights $\tilde{\mathbf{w}}$ in the MMSE sense using (8)–(10) or any other approach of choice (e.g., LMS, RLS, etc.).
5) Apply the weights to obtain a soft-decision statistic $y = \tilde{\mathbf{w}}^{\mathrm{H}}\tilde{\mathbf{x}}$ for further processing.
6) Repeat steps 2)–5) for all other users of interest. Note that the choice of the $(\eta - 1)$ most interfering users in step 2) changes for each user of interest.

## D. Numerical Examples

This section presents representative simulations to illustrate the performance of the JDL-Z algorithm. All examples simulate multicell CDMA systems. The first example illustrates the performance advantages of JDL-Z in heavily loaded situations. The second example illustrates the faster convergence rate of JDL-Z compared with other reduced-rank processors. The final example illustrates a JDL–MSWF algorithm: one that uses the MSWF instead of zero forcing in the first stage.

We assume that the channels of all intracell users are known but that the receiver does not have any knowledge of the intercell interference. SMI is used to estimate the adaptive weights $\tilde{\mathbf{w}}$ in the second JDL stage. Slow, flat, and uncorrelated Rayleigh fading channels are modeled as constant over the time period used to estimate the autocorrelation matrix $\tilde{\mathbf{R}}$ and cross-correlation vector $\tilde{\mathbf{v}}$ in each simulation. The channel-fading coefficients are modeled as i.i.d. zero-mean complex Gaussian random variables with unit variance. BPSK is used for data modulation. It is noted that matched filtering cannot be used to obtain OMMSE weights because not all channels of users are known—the postmatched-filter weights are optimal in terms of MMSE only if the matched filter matches the received signal to all $K$ users' spatio-temporal signatures.

*1) Heavily Loaded System With Power Control and Limited Training:* This example uses a multicell CDMA system with $N = 4$ receive antennas, $M = 30$ equal power intracells, and $P = 80$ intercell users, each with processing gain $G = 12$. Of the 80 intercell interferers, 30 are at a power level $-12$ dB, and 50 are at $-20$ dB with respect to the desired user. There are 100 training bits used to estimate the required weights.

Fig. 7 compares the BER of two implementations of JDL-Z ($\eta = 1$ and 8), JDL $(\eta = 8)$, OMMSE, MSWF $(D = 8)$, and COMMSE algorithms. JDL-Z with $\eta = 1$ is equivalent to using a single zero-forcing stage. Note the significantly worsened performance of the OMMSE (optimal with infinite training), JDL, MSWF, and COMMSE filters in this multicell system. Among filters introduced earlier, the JDL algorithm

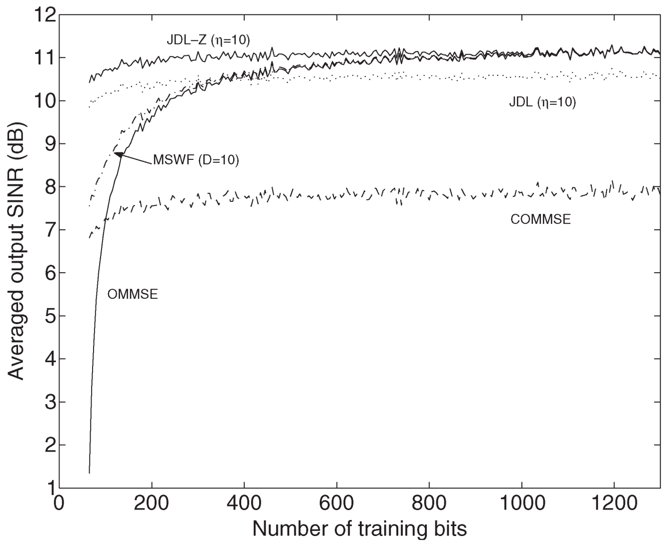


Fig. 8. Convergence rate comparison of joint-domain processors. $N = 4$, $K = 80$, and $G = 16$.

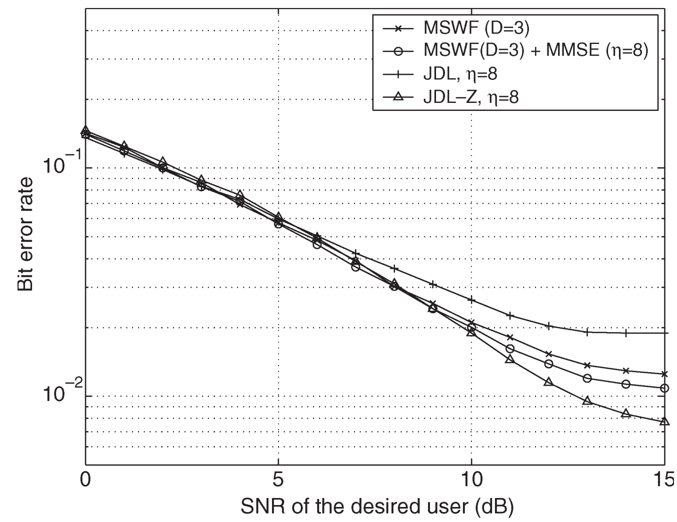


Fig. 9. Performance of a JDL–MSWF filter.

performs the best. On the other hand, JDL-Z processing achieves extremely good performance, even with small rank. Note that increasing $\eta$ from one to eight results in only slightly better performance. This is because the zero-forcing first stage eliminates all intracell interference. The second stage serves to further suppress interference. Note that the computational overhead in implementing this second stage is only $\mathcal{O}[\eta^3]$, since any zero-forcing receiver would detect all $M$ intracell users.

*2) Comparing Convergence Rates:* The JDL and JDL-Z algorithms reduce the number of adaptive weights to $\eta$. In this regard, when using training, these algorithms have significantly improved convergence rate. This example serves to illustrate this important characteristic of the JDL-based algorithms.

This example uses a multicell CDMA system with power control, $N = 4$ receive antennas, processing gain of $G = 16$ with $M = 20$ equal power intracells, and $P = 60$ intercell users. All intracell users have SNR of 10 dB. Of the intercell users, 20 are at a power level of $-2$ dB, and 40 are at $-10$ dB. Fig. 8 plots the resulting output SINR of the JDL, JDL-Z, MSWF, theoretically OMMSE, and COMMSE algorithms as a function of the number of training bits used. The JDL and JDL-Z algorithms use $\eta = 10$, while the MSWF uses $D = 10$ stages. The results are obtained by averaging over random spreading codes and channels.

The figure shows that the OMMSE scheme has the slowest convergence rate as it has the largest number of adaptive weights (i.e., $NG = 64$). With the same number of adaptive weights $(\eta = 10)$, JDL-Z converges slightly faster than JDL. This comes at the cost of significant additional computational complexity in the first zero-forcing stage. The number of iterations to convergence are 350, 450, and 1200 for the JDL-Z, JDL, and OMMSE algorithms, respectively. Therefore, the JDL-Z algorithm would be most effective in relatively fast fading channels.

Another interesting characteristic of Fig. 8 is the level of the output SINR. As expected, the OMMSE and MSWF filters converge to the optimal SINR. Interestingly, the JDL-Z filter also results in the same optimal SINR, even though it requires significantly reduced training.

*3) Illustrating a JDL–MSWF Algorithm:* This final example illustrates the flexibility of the JDL concept. In the first stage, the zero forcing is replaced by the MSWF algorithm, i.e., the data vector $\tilde{\mathbf{x}}$ are the adaptive soft-decision statistics for the desired user and the $\eta - 1$ most correlated users using the MSWF algorithm. As with the JDL-Z algorithm, this algorithm can be implemented by choosing the estimates corresponding to $\eta$ users after all $M$ intracell users have been processed using the MSWF filter.

The example uses a system with $N = 4$ antennas and processing gain of $G = 8$ supporting equal power $M = 20$ intracell users. $P = 60$ intercell users act as interference with 20 users at a power level of $-12$ dB and 40 users at a power level of $-20$ dB with respect to the desired user. There are 1000 training bits used to estimate the required adaptive weights, hence, achieving near-optimal performance.

Fig. 9 plots the BER results of using a JDL–MSWF filter. The figure also plots the performance of the JDL, MSWF, and JDL-Z algorithms. The JDL, JDL-Z, and JDL–MSWF algorithms use $\eta = 8$ in the second stage. The MSWF algorithm is implemented with $D = 3$ stages. As expected, given the large amount of training, the MSWF filter outperforms the JDL algorithm. However, due to the additional stage of processing, the JDL–MSWF performs better than the MSWF by itself. At the cost of additional computational complexity in the zero-forcing stage, the JDL-Z filter outperforms all other filters.

## VI. Conclusions

This paper has presented a new two-stage JDL adaptive processing. Borrowing from the radar literature, the algorithm is said to transform the received data to "beamspace," that is, the postmatched filter data. In fading channels, this beamspace has little physical meaning. However, the presence of fading and the use of spreading codes make the algorithm presented here very different from the original JDL algorithm [11], [12]. In particular, this paper addresses the crucial issue of how to form the transformation matrix by introducing three criteria for choosing the users that form this matrix. This issue does not arise in the radar literature.

In the second stage of processing, the JDL algorithm adaptively combines $\eta$ beamspace data samples based on an MMSE criterion. The JDL algorithm is then shown to have low complexity while providing better performance than other reduced-rank processors, particularly with limited training. This is mainly because of the reduction in the number of adaptive unknowns to $\eta$. The numerical examples show that the JDL processor outperforms the fully OMMSE processor and other reduced-rank filters, such as the MSWF and COMMSE filters with short training sequences.

Building on the JDL algorithm, this paper also introduced a new JDL-Z processor that integrates JDL and zero forcing. The nonadaptive first stage in JDL processing is replaced by an adaptive zero-forcing technique. Such a processor is particularly useful for multicell CDMA systems, effectively suppressing inter- and intracell interference. The use of the second stage eliminates the problem of noise enhancement associated with zero forcing. This processor also serves to illustrate the flexibility of the JDL concept, and zero forcing is just one possible choice for the first stage. An example in Section V-D3 illustrates the use of the less-complex MSWF algorithm in the first stage.

## APPENDIX A
## PROOF OF THE OPTIMAL TRANSFORMATION MATRIX FOR JDL

The proposition in Section III-A2 states that the best choice, in the MMSE sense, to augment the rank-$N$ transformation matrix $\mathbf{T}_n$ is to choose the spatio-temporal signature vector $\mathbf{z}_i$, which is given by

$$\mathbf{t}_{n+1} = \arg\max_{\mathbf{z}_i} \frac{\left|\tilde{\mathbf{z}}_1^{\mathrm{H}}\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i - \zeta_{1,i}\right|^2}{\mathbf{z}_i^{\mathrm{H}}\left(\mathbf{R} - \mathbf{R}\mathbf{T}_n\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\right)\mathbf{z}_i}. \qquad (20)$$

*Proof:* Define the following symbols:

$\tilde{\mathbf{R}}_n$ is the transformed autocorrelation matrix of the received signal with $\eta = n$.

$\zeta_{ij} = \mathbf{z}_i^{\mathrm{H}}\mathbf{z}_j$ is the correlation between user $i$ and $j$. $\mathbf{T}_n$ is the transformation matrix with $\eta = n$.

Without loss of generality, we assume that user 1 is the desired user, and the LPR size is $\eta$. Since minimizing the MSE of the detected symbol is equivalent to maximizing $\tilde{\mathbf{v}}\tilde{\mathbf{R}}_{n+1}^{-1}\tilde{\mathbf{v}}$, the $(n+1)$th column of $\mathbf{T}_{n+1} = [\mathbf{T}_n \ \mathbf{z}_i]$, $\mathbf{z}_i$ is chosen such that $\tilde{\mathbf{v}}\tilde{\mathbf{R}}_{n+1}^{-1}\tilde{\mathbf{v}}$ is maximized. We have

$$\begin{aligned}\tilde{\mathbf{R}}_{n+1}^{-1} &= E\left[\mathbf{T}_{n+1}\mathbf{x}\mathbf{x}^{\mathrm{H}}\mathbf{T}_{n+1}^{\mathrm{H}}\right]^{-1}\\ &= \begin{pmatrix}\tilde{\mathbf{R}}_{11} & \tilde{\mathbf{R}}_{12}\\ \tilde{\mathbf{R}}_{12}^{\mathrm{H}} & \tilde{\mathbf{R}}_{22}\end{pmatrix}\end{aligned}$$

TABLE II
COMPUTATIONAL COMPLEXITY ANALYSIS OF MSWF

| **MSWF Algorithm** | **Complexity** |
|---|---|
| Forward Recursion for $n = 1, ..., D$ | |
| $\mathbf{c}_n = \mathrm{E}[d_{n-1}^*(i)\mathbf{x}_{n-1}(i)]/\lVert\mathrm{E}[d_{n-1}^*(i)\mathbf{x}_{n-1}(i)]\rVert$ | $\mathcal{O}[NG]$ |
| $d_n(i) = \mathbf{c}_n^H\mathbf{x}_{n-1}(i)$ | $\mathcal{O}[NG]$ |
| $\mathbf{B}_n = \mathbf{I} - \mathbf{c}_n\mathbf{c}_n^H$ | $\mathcal{O}[(NG)^2]$ |
| $\mathbf{x}_n(i) = \mathbf{B}_n^H\mathbf{x}_{n-1}(i)$ | $\mathcal{O}[(NG)^2]$ |
| Backward Recursion for $n = D, ..., 1$ | |
| $w_n = \mathrm{E}[d_{n-1}^*(i)e_n(i)]/\mathrm{E}[\lvert e_n(i)\rvert^2]$ | $\mathcal{O}[1]$ |
| $e_{n-1}(i) = d_{n-1}(i) - w_n^*e_n(i)$ | $\mathcal{O}[1]$ |
| Total Complexity | $\mathcal{O}[D(NG)^2]$ |

where

$$\begin{aligned}\tilde{\mathbf{R}}_{11} &= \left[\tilde{\mathbf{R}}_n - \mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i\left(\mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{z}_i\right)^{-1}\mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{T}_n\right]^{-1}\\ \tilde{\mathbf{R}}_{12} &= \frac{\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i}{\mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{T}_n\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i - \mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{z}_i}\\ \tilde{\mathbf{R}}_{22} &= \left[\mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{z}_i - \mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{T}_n\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n\mathbf{R}\mathbf{z}_i\right]^{-1}.\end{aligned}$$

After algebraic simplification and using the matrix-inversion lemma

$$\begin{aligned}&\tilde{\mathbf{v}}^{\mathrm{H}}\tilde{\mathbf{R}}_{n+1}^{-1}\tilde{\mathbf{v}}\\ &= |a_1|^2\left[\tilde{\mathbf{z}}_1^{\mathrm{H}}\left(\tilde{\mathbf{R}}_n - \mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i\left(\mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{z}_i\right)^{-1}\mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{T}_n\right)^{-1}\tilde{\mathbf{z}}_1\right.\\ &\qquad\left. + \frac{\zeta_{1,i}\mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{T}_n\tilde{\mathbf{R}}_n^{-1}\tilde{\mathbf{z}}_1 + \zeta_{i,1}\tilde{\mathbf{z}}_1^{\mathrm{H}}\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i - |\zeta_{1,i}|^2}{\mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{T}_n\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i - \mathbf{z}_i^{\mathrm{H}}\mathbf{R}\mathbf{z}_i}\right]\\ &= |a_1|^2\tilde{\mathbf{z}}_1^{\mathrm{H}}\tilde{\mathbf{R}}_n^{-1}\tilde{\mathbf{z}}_1 + |a_1|^2\frac{\left|\tilde{\mathbf{z}}_1^{\mathrm{H}}\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\mathbf{z}_i - \zeta_{1,i}\right|^2}{\mathbf{z}_i^{\mathrm{H}}\left(\mathbf{R} - \mathbf{R}\mathbf{T}_n\tilde{\mathbf{R}}_n^{-1}\mathbf{T}_n^{\mathrm{H}}\mathbf{R}\right)\mathbf{z}_i}.\end{aligned} \qquad (21)$$

Since the first term is constant, the best choice of $\mathbf{z}_i$ is that given in (20). ■

## APPENDIX B
## PROOF OF COMPUTATIONAL COMPLEXITIES

### A. Complexity of MSWF

The steps involved in the MSWF algorithm [9] with $D$ stages and the associated computational complexity are shown in Table II, where $d_n(i)$ is the $i$th desired symbol at the $n$th stage, and $x_n(i)$ is the $i$th filtered output at the $n$th stage. The statistic of the desired bit is $w_1 e_1$. To initialize the parameters, $d_0(i) = b_1(i)$, $\mathbf{x}_0(i) = \mathbf{x}(i)$, and $e_D(i) = d_D(i)$. As shown in the table, the computational complexity of MSWF is, therefore, $\mathcal{O}[D(NG)^2]$. Note also that additional real-time computation is necessary to obtain the statistics for the desired symbols.

### B. Complexity of COMMSE Algorithm

For the COMMSE algorithm [2], let $\hat{\mathbf{w}}_t$ and $\hat{\mathbf{w}}_s$ be the temporal and spatial filters that require updating. The resulting $[\hat{\mathbf{w}}_t, \hat{\mathbf{w}}_s]$ pair yields the matrix filter

$$\mathbf{w}_{\mathrm{COMMSE}} = \hat{\mathbf{w}}_t \otimes \hat{\mathbf{w}}_s. \tag{22}$$

Without loss of generality, we assume that user 1 is the desired user; the two update equations for $\hat{\mathbf{w}}_t$ and $\hat{\mathbf{w}}_s$ become

$$\hat{\mathbf{w}}_t = a_1 \left(\tilde{\mathbf{w}}_s^{\mathrm{H}} \mathbf{h}_1\right) \left( \sum_{j=1}^{K} |a_j|^2 \left|\tilde{\mathbf{w}}_s^{\mathrm{H}} \mathbf{h}_j\right|^2 \mathbf{s}_j \mathbf{s}_j^{\mathrm{H}} + \sigma^2 |\tilde{\mathbf{w}}_s|^2 \mathbf{I} \right)^{-1} \mathbf{s}_1 \tag{23}$$

$$\hat{\mathbf{w}}_s = a_1 \left(\tilde{\mathbf{w}}_t^{\mathrm{H}} \mathbf{s}_1\right) \left( \sum_{j=1}^{K} |a_j|^2 \left|\tilde{\mathbf{w}}_t^{\mathrm{H}} \mathbf{s}_j\right|^2 \mathbf{h}_j \mathbf{h}_j^{\mathrm{H}} + \sigma^2 |\tilde{\mathbf{w}}_t|^2 \mathbf{I} \right)^{-1} \mathbf{h}_1. \tag{24}$$

This set of coupled equations is executed iteratively until the iteration converges. We denote the number of iterations to convergence as $Q$. It can be shown easily that the complexity order of updating $\hat{\mathbf{w}}_t$ is $\mathcal{O}[KN + KG^2 + G^3]$. Similarly, the complexity of updating $\hat{\mathbf{w}}_s$ is $\mathcal{O}[KG + KN^2 + N^3]$. These two update equations are executed $Q$ times until the results converge. The Kronecker product of $\hat{\mathbf{w}}_s$ and $\hat{\mathbf{w}}_t$ has the complexity of $\mathcal{O}[NG]$. Therefore, the total order of the complexity of the COMMSE filter is $\mathcal{O}[Q(KG^2 + KN^2 + G^3 + N^3)]$.

### C. Complexity of JDL-Z Algorithm

The JDL-Z algorithm requires the solution of (18) for each of the $M$ intracell users. The matrix $\mathbf{Z}_{1:M}$ is an $NG \times M$ matrix and is common to all $M$ equations. The complexity of this stage, which is the same as the complexity of any zero-forcing filter, is, therefore, $\mathcal{O}[(NG)^3]$. The second stage of the JDL-Z is identical in computation to the JDL algorithm and has complexity $\mathcal{O}[\eta^3]$. The overall complexity is, therefore, $\mathcal{O}[(NG)^3]$. The performance advantages of the JDL-Z algorithm are, therefore, at the cost of computational complexity. However, it is worth repeating that the additional complexity of implementing the JDL MMSE stage is only $\mathcal{O}[\eta^3]$.

## REFERENCES


[1] X. Bernstein and A. M. Haimovich, "Space-time optimum combining for increased capacity of wireless CDMA," in *Proc. IEEE Int. Conf. Commun.*, Jun. 1996, vol. 1, pp. 597–601.

[2] A. Yener, R. D. Yates, and S. Ulukus, "Combined multiuser detection and beamforming for CDMA systems: Filter structures," *IEEE Trans. Veh. Technol.*, vol. 51, no. 5, pp. 1087–1095, Sep. 2002.

[3] A. M. Haimovich and Y. Bar-Ness, "An eigenanalysis interference canceler," *IEEE Trans. Signal Process.*, vol. 39, no. 1, pp. 76–84, Jan. 1991.

[4] J. S. Goldstein and I. S. Reed, "Reduced-rank adaptive filtering," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 492–496, Feb. 1997.

[5] N. L. Owsley, *Array Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[6] S. N. Batalama, M. J. Medley, and D. A. Pados, "Robust adaptive recovery of spread-spectrum signals with short data records," *IEEE Trans. Commun.*, vol. 48, no. 10, pp. 1725–1731, Oct. 2000.

[7] D. A. Pados and G. N. Karystinos, "An iterative algorithm for the computation of the MVDR filter," *IEEE Trans. Signal Process.*, vol. 49, no. 2, pp. 290–300, Feb. 2001.

[8] H. Qian and S. N. Batalama, "Data record-based criteria for the selection of an auxiliary vector estimator of the MMSE/MVDR filter," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1700–1708, Oct. 2003.

[9] M. Honig and J. S. Goldstein, "Adaptive reduced-rank interference suppression based on the multistage Wiener filter," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 986–994, Jun. 2002.

[10] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A multistage representation of the Wiener filter based on orthogonal projections," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2943–2959, Nov. 1998.

[11] H. Wang and L. Cai, "On adaptive spatial-temporal processing for airborne surveillance radar systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 3, pp. 660–670, Jul. 1994.

[12] R. S. Adve, T. B. Hale, and M. C. Wicks, "Practical joint domain localised adaptive processing in homogeneous and nonhomogeneous environments. Part 1: Homogeneous environments," *Proc. Inst. Electr. Eng.—Radar, Sonar Navigation*, vol. 147, no. 2, pp. 57–65, Apr. 2000.

[13] L. C. Godara, "Application of antenna arrays to mobile communications, Part II: Beam-forming and direction-of-arrival considerations," *Proc. IEEE*, vol. 85, no. 8, pp. 1195–1245, Aug. 1997.

[14] I. N. Psaromiligkos and S. N. Batalama, "Data record size requirements for adaptive space-time DS/CDMA signal detection," *IEEE Trans. Commun.*, vol. 52, no. 9, pp. 1538–1546, Sep. 2004.

[15] T. J. Lim, Y. Gong, and B. Farhang-Boroujeny, "Convergence analysis of chip- and fractionally spaced LMS adaptive multiuser CDMA detectors," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 2219–2228, Aug. 2000.

[16] S. Verdu, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.



**Rebecca Y. M. Wong** was born in Hong Kong in 1978. She received the B.A.Sc. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and the M.A.Sc. degree in electrical engineering from the University of Toronto, Toronto, ON, in 2004. Her research interest during her graduate studies was joint-domain adaptive processing in code-division multiple-access systems.

She is currently a System Design Analyst with Redknee Inc., Mississauga, ON.



**Raviraj Adve** (S'88–M'91–SM'03) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, in 1990 and the Ph.D. degree from Syracuse University, Syracuse, NY, in 1996, where his dissertation received a Doctoral Prize.

Between 1997 and August 2000, he was with Research Associates for Defense Conversion Inc., on contract with the Air Force Research Laboratory, Rome, NY. Since August 2000, he has been with the Faculty of the University of Toronto, Toronto, ON, Canada. His research interests include practical signal-processing algorithms for smart antennas with applications in wireless communications and radar. He is currently focused on linear precoding in wireless communications, cooperative communications, and augmenting space-time adaptive processing with a waveform dimension.