Adaptive Chip-Rate Equalization of Downlink Multirate Wideband CDMA

Adam R. Margetts, *Student Member, IEEE,* and Philip Schniter, *Member, IEEE*

***Abstract*—We consider a downlink direct sequence-code division multiple access (DS-CDMA) system in which multirate user signals are transmitted via synchronous orthogonal short codes overlaid with a common scrambling sequence. The transmitted signal is subjected to significant time- and frequency-selective multipath fading.**

In response to this scenario, a novel two-mode receiver is proposed that accomplishes chip-rate adaptive equalization aided by filtering and/or cancellation of multiaccess interference (MAI). In the acquisition mode, a code-multiplexed pilot is used to adapt the equalizer from cold start or loss-of-lock. The use of MAI filtering results in a third-order least mean squares (LMS) algorithm, which has significant advantages over standard (i.e., first-order) LMS in nonstationary environments. In the tracking mode, decision-direction facilitates MAI-cancellation in the equalizer update, which enhances performance. The receiver monitors pilot decision quality as a means of switching between the two modes. The performance of the adaptive receiver is studied through analysis and simulation.

***Index Terms*—Adaptive equalization, averaged-error averaged-regressor LMS (AEAR-LMS), code division multiple access (CDMA), decision-direction (DD), scrambling, variable step-size.**

I. Introduction

In third generation mobile direct sequence-code division multiple access (DS-CDMA) systems, downlink multirate symbol streams are multiplexed using orthogonal short codes and then scrambled by a cell-specific long code prior to synchronous transmission, as shown in Fig. 1. The multipath propagation channel has a time-dispersive effect that destroys the orthogonality among user codes, substantially degrading the performance of matched-filter (MF)-based detectors. The usual methods of multipath mitigation in CDMA (e.g., "blind minimum output energy" techniques [2]) rely on received signal cyclostationarity. In our application, however, the scrambling code destroys the cyclostationarity; therefore, an alternative means of multipath mitigation is necessary. We focus on adaptive chip-level equalization as a means of restoring orthogonality and hence reducing multiaccess interference (MAI) in a time- and frequency-selective fading environment. Since the mobile terminals in these systems are cost- and power-limited, we desire a low-complexity solution.

Adaptive receivers for the scrambled CDMA downlink can be classified as follows: those that estimate the channel response for subsequent estimation of equalizer coefficients and those that adapt the equalizer coefficients directly. Several receivers of the channel-estimating variety are discussed and compared in [3] and [4]. Reduced-rank methods have been proposed in [5]. The receiver we propose adapts the equalizer coefficients directly. Ultimately, the linear minimum mean squared error (MMSE) equalizer (with perfect channel knowledge) forms a reference against which any linear receiver can be compared [6].

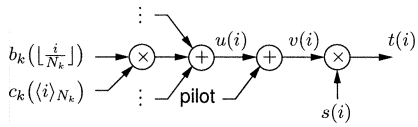


Fig. 1. Scrambled-CDMA transmitter model.

Frank and Visotsky [7] first proposed the use of a code-multiplexed pilot for equalizer adaptation in the scrambled CDMA downlink. (A more thorough treatment is found in [8].) Though Petre *et al.* [9] later extended this pilot-aided scheme to incorporate chip-fractional sampling, both [7] and [9] update the equalizer at the symbol rate. A straightforward chip-rate Normalized-LMS (NLMS) equalizer that uses pilot chips as a training signal was proposed and studied in [3] and [10]. Because short-code orthogonality manifests only at *symbol*-synchronous instants, however, the presence of MAI can severely degrade the chip-rate updates at off-symbol instants. Thus, small adaptation step sizes are needed to provide sufficient averaging [3], which leads to reduced tracking ability. In response, we develop a modified LMS algorithm that incorporates a novel "error filtering" mechanism. The mean transient response of the resulting algorithm corresponds to that of a third-order dynamical system (in contrast to standard LMS, which behaves as a first-order dynamical system [11]). Such higher order LMS algorithms have demonstrated tracking performance superior to standard LMS [12].

The aforementioned pilot-aided scheme is intended for cold-start or loss-of-lock situations; when it yields adequately reliable symbol estimates, our receiver switches to a decision-directed (DD) equalizer update algorithm, for which we assume that all active user codes are known. The chip-rate DD algorithm alleviates the MAI problem faced by the pilot-aided algorithm and, consequently, yields better performance.

In Section II, we outline the system model and perform a preliminary analysis, and in Section III, we describe several optimal equalizer designs. In Section IV, we describe pilot-aided adaptive equalization algorithms and analyze their mean-convergence, and in Section V, we outline a DD algorithm and analyze its robustness to decision errors. In Section VI, we numerically evaluate the performance of our adaptive receiver in fading

Manuscript received August 4, 2003; revised July 7, 2004. This work was supported by the Ohio Space Grant Consortium. Portions of this work appeared at the 2002 Asilomar Conference on Signals, Systems, and Computers [1]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Constantinos B. Papadias.

The authors are with the Department of Electrical Engineering, The Ohio State University, Columbus OH 43210 USA (e-mail: schniter.1@osu.edu).

Digital Object Identifier 10.1109/TSP.2005.847848

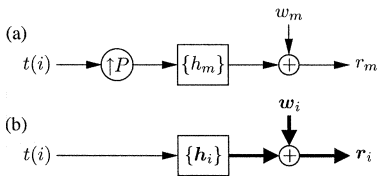


Fig. 2. Channel model at (a) chip-fractional-rate and (b) chip-rate.

multipath for comparison to an adaptive Rake receiver and the max-SINR receiver, and in Section VII, we draw conclusions.

II. System Model

A. Model Specification

Our transmitted signal model is illustrated in Fig. 1 with the following definitions. K denotes the number of users; N_k the kth user's spreading gain; $\{b_k(n), n \in \mathbb{Z}\}$ the kth user's symbol stream; $\{c_k(i), i = 0 \ldots N_k - 1\}$ the kth user's short code (where i is the chip index); $\{u(i)\}$ the multiuser sequence; $\{v(i)\}$ the multiuser-plus-pilot sequence; $\{s(i)\}$ the scrambling sequence; and $\{t(i)\}$ the transmitted sequence. The function $\langle M \rangle_N$ denotes $M \bmod N$, and $\lfloor \cdot \rfloor$ denotes the floor function. In the sequel, we will use $\{\delta_i\}$ to denote the Kronecker delta sequence and $\boldsymbol{e}_\nu$ to denote a vector of zeros with a one in the νth position[1] (for $\nu \geq 0$). We use $(\cdot)^T$ to denote the transpose and $(\cdot)^H$ the Hermitian transpose.

Fig. 2(a) describes the discrete-time chip-fractionally spaced channel model using P samples per chip, where $\{h_m\}$ denotes the chip-fractional impulse response of the channel and pulse-shaping filters, and where $\{w_m\}$ denotes additive channel noise. In Sections II–V, we assume that the channel response varies slowly enough (relative to the chip rate) to justify this linear time-invariant (LTI) model; the effect of time-variation is discussed in more detail in Section VI. The chip-fractional received signal can be written as (see, e.g., [13] for a detailed development of this multirate fractionally sampled channel representation)

$$r_m = \sum_{\ell=-\infty}^{\infty} t(\ell) h_{m-\ell P} + w_m. \tag{1}$$

A chip-spaced model follows from definition of the vector-valued quantities $\boldsymbol{h}_i := \left[h_{iP+P-1}, \ldots, h_{iP}\right]^T$, $\boldsymbol{w}_i := \left[w_{iP+P-1}, \ldots, w_{iP}\right]^T$, and $\boldsymbol{r}_i := \left[r_{iP+P-1}, \ldots, r_{iP}\right]^T$, as in Fig. 2(b). Then

$$\boldsymbol{r}_i = \sum_{\ell=0}^{L_h} \boldsymbol{h}_\ell t(i-\ell) + \boldsymbol{w}_i. \tag{2}$$

Since our equalizer will employ a window of $L_f + 1$ chip-spaced samples, we define $\boldsymbol{r}(i) := \left[\boldsymbol{r}_i^T, \ldots, \boldsymbol{r}_{i-L_f}^T\right]^T$,

[1]The subscript "ν" on e_ν refers to the relative delay between the descrambling code $s^*(i-\nu)$ and the equalizer output $x(i)$ and will be used in this context throughout the paper.

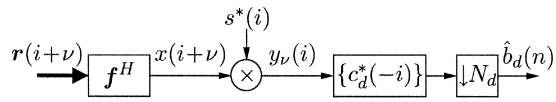


Fig. 3. Equalization, descrambling, and matched-filtering.

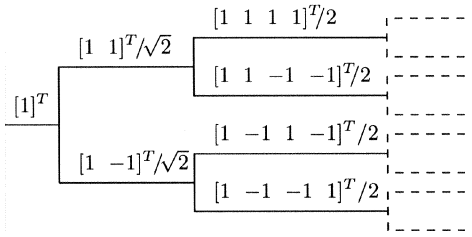


Fig. 4. Orthonormal multirate spreading code tree.

$\boldsymbol{t}(i) := \left[t_i, \ldots, t_{i-L_h-L_f}\right]^T$, $\boldsymbol{w}(i) := \left[\boldsymbol{w}_i^T, \ldots, \boldsymbol{w}_{i-L_f}^T\right]^T$, and the block-Toeplitz matrix

$$\boldsymbol{H} := \begin{bmatrix} \boldsymbol{h}_0 & \cdots & \boldsymbol{h}_{L_h} & & \\ & \ddots & & \ddots & \\ & & \boldsymbol{h}_0 & \cdots & \boldsymbol{h}_{L_h} \end{bmatrix}$$

so that

$$\boldsymbol{r}(i) = \boldsymbol{H}\boldsymbol{t}(i) + \boldsymbol{w}(i). \tag{3}$$

Collecting the $P(L_f + 1)$ equalizer coefficients into the vector $\boldsymbol{f}$, the equalizer produces the chip-rate output sequence $\{x(i)\}$ (see Fig. 3[2]):

$$\begin{aligned} x(i) &= \boldsymbol{f}^H \boldsymbol{H}\boldsymbol{t}(i) + \boldsymbol{f}^H \boldsymbol{w}(i) && (4)\\ &= \boldsymbol{q}^H \boldsymbol{t}(i) + \tilde{w}(i) && (5)\\ &= \sum_{\ell=0}^{L_f+L_h} q_\ell^* t(i-\ell) + \tilde{w}(i) && (6) \end{aligned}$$

where $\boldsymbol{q} := \boldsymbol{H}^H \boldsymbol{f} = [q_0, \ldots, q_{L_f+L_h}]^T$ will be referred to as the "chip-spaced system response", and $\tilde{w}(i) := \boldsymbol{f}^H \boldsymbol{w}(i)$.

We make the following assumptions about our system with $\mathrm{E}[\cdot]$ denoting the expectation operator.

A1) Circular, i.i.d., zero-mean, PSK scrambling:

$$\forall\, i, \quad |s(i)| = 1; \quad \mathrm{E}[s(i)s^*(i+j)] = \delta_j.$$

A2) Multirate orthonormal Walsh codes:

The code-tree for the users' short codes is shown in Fig. 4 [14]. Codes branching from an active code are not valid in order to preserve orthonomality. For the active code-set $\mathcal{A}$, the following relationship holds:

$$\forall\, k, \quad \ell \in \mathcal{A}, \text{ s.t. } N_\ell \geq N_k, \quad \forall\, m \in \left\{0, \ldots, \frac{N_\ell}{N_k} - 1\right\}$$

$$\forall\, j : \delta_{\ell-k} = \sum_{i=0}^{N_k-1} c_k^*(i) c_\ell(i + mN_k), \quad |c_k(j)| = \frac{1}{\sqrt{N_k}}.$$

[2]The system in Fig. 3 can be made causal by the substitution $i \to i - \nu$. The noncausal labeling simplifies the derivations of (8) and (9).

A3) Constant pilot at user index $k = 0$:

$$\forall\, n, \quad b_0(n) = b_0;\ \ c_0(i) = \begin{cases} \frac{1}{\sqrt{N_0}}, & 0 \le i \le N_0 - 1 \\ 0, & \text{else.} \end{cases}$$

A4) Circular, independent, zero-mean user symbols:

$$\forall\, n, \quad \forall\, k \neq 0 : \mathrm{E}[b_k(n) b_\ell^*(n+m)] = P_k \delta_m \delta_{\ell - k}$$

where P_k denotes the kth user's symbol energy, and from A2), the kth user's chip amplitude is $\sqrt{P_k/N_k}$. $\{b_k(n)\}$ is independent of $\{s(i)\}$.

A5) Zero-mean, circular, white, Gaussian noise $\{w_m\}$ with variance σ_w^2, independent of $\{b_k(n)\}$ and $\{s(i)\}$.

B. Preliminary Analyzes

From Fig. 1 and A1)–A4), the transmitted signal is zero-mean and uncorrelated with variance

$$\begin{aligned} \sigma_t^2 &= \mathrm{E}\,|v(i)|^2\, \mathrm{E}\,|s(i)|^2 \\ &= \frac{|b_0|^2}{N_0} + \sigma_u^2 \end{aligned} \tag{7}$$

where $\sigma_u^2 := \sum_{k=1}^{K} P_k/N_k$ since $v(i) = (b_0/\sqrt{N_0}) + u(i)$, and $u(i) = \sum_{k=1}^{K} c_k\big(\langle i \rangle_{N_k}\big) b_k\big(\lfloor i/N_k \rfloor\big)$. Note that $\sigma_v^2 = \sigma_t^2$ and N_0 is the pilot spreading factor.

From Fig. 3, (6), and A1), the equalized descrambled signal $\{y_\nu(i)\}$ can be written as

$$\begin{aligned} y_\nu(i) = s^*(i)\tilde{w}(i+\nu) + q_\nu^* v(i) \\ + \sum_{\ell \neq \nu} q_\ell^* s^*(i) s(i+\nu-\ell) v(i+\nu-\ell). \end{aligned} \tag{8}$$

The first term on the right of (8) represents additive noise, the second term represents the desired signal plus "coherent" MAI and pilot, and the third represents "randomized" interference.

From Fig. 3 and A2), the symbol estimates for the dth user are obtained by matched-filtering (8) using the dth user's code.

$$\begin{aligned} \hat{b}_d(n) &= \sum_j c_d^*(-j) y_\nu(nN_d - j) \\ &= \sum_{i=0}^{N_d-1} c_d^*(i) y_\nu(nN_d + i) \\ &= \sum_{i=0}^{N_d-1} c_d^*(i) s^*(nN_d + i) \tilde{w}(nN_d + i + \nu) + q_\nu^* b_d(n) \\ &\quad + \sum_{\ell \neq \nu} q_\ell^* \sum_{i=0}^{N_d-1} s^*(nN_d + i) s(nN_d + i + \nu - \ell) \\ &\quad \times c_d^*(i) v(nN_d + i + \nu - \ell). \end{aligned} \tag{9}$$

From (9), we see that $\hat{b}_d(n)$ is composed of a noise term, a signal term, and an MAI term. The MAI term vanishes if and only if the channel is perfectly equalized, i.e., $q_i = \delta_{i-\nu}$ for some ν. Thus, while equalization yields the possibility to suppress MAI, Rake receivers (which avoid equalization and combine the MF outputs from descramblers with different chip-offsets) are inherently MAI-corrupted [15]. Equalization does, however, introduce the potential for noise gain, i.e., an increase in the variance of $\tilde{w}(i)$. Each equalizer design criterion (e.g., MMSE, zero-forcing) strikes a particular balance between residual MAI and noise variance.

A useful measure of system performance is the signal to interference-plus-noise ratio (SINR) of the symbol estimate $\hat{b}_d(n)$. Below, we calculate SINR for a particular coefficient set $\boldsymbol{f}$, assuming the channel is known. Without loss of generality, we consider the 0th symbol of the dth user in (9), so that we have (10), shown at the bottom of the page. Using A5), $\mathrm{E}|\tilde{w}(i+\nu)|^2 = \mathrm{E}|\boldsymbol{f}^H \boldsymbol{w}(i+\nu)|^2 = \sigma_w^2 \|\boldsymbol{f}\|^2$. Since A1) implies

$$\begin{aligned} &\mathrm{E}[s^*(i) s(i+\nu-\ell) s(j) s^*(j+\nu-n)] \\ &\qquad = \begin{cases} 1, & \text{if } \ell = n, i = j \\ 1, & \text{if } \ell = n = \nu, i \neq j \\ 0, & \text{else} \end{cases} \end{aligned} \tag{11}$$

the interference power is

$$\mathrm{E}\left| \sum_{\ell \neq \nu} q_\ell^* \sum_{i=0}^{N_d-1} c_d^*(i) s^*(i) t(i+\nu-\ell) \right|^2 = \sigma_t^2 \sum_{\ell \neq \nu} |q_\ell|^2. \tag{12}$$

Since the noise and interference in the denominator of (10) are uncorrelated

$$\begin{aligned} \mathrm{SINR} &= \frac{P_d |q_\nu|^2}{\sigma_w^2 \|\boldsymbol{f}\|^2 + \sigma_t^2 \sum_{\ell \neq \nu} |q_\ell|^2} \\ &= \frac{P_d |\boldsymbol{h}_\nu^H \boldsymbol{f}|^2}{\boldsymbol{f}^H \big(\sigma_w^2 \boldsymbol{I} + \sigma_t^2 \boldsymbol{H}\boldsymbol{H}^H - \sigma_t^2 \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H\big) \boldsymbol{f}} \end{aligned} \tag{13}$$

where P_d is the dth user's symbol energy, and $\boldsymbol{h}_\nu$ denotes the νth column of $\boldsymbol{H}$.

III. Optimal Equalizers

Several optimal (nonadaptive) equalizers are reviewed and compared in the sections below. These criteria are (stochastically) minimized by the adaptive algorithms discussed in the sequel.

$$\mathrm{SINR} := \frac{\mathrm{E}\,|q_\nu^* b_d(0)|^2}{\mathrm{E}\left| \sum_{i=0}^{N_d-1} c_d^*(i) s^*(i) \left(\tilde{w}(i+\nu) + \sum_{\ell \neq \nu} q_\ell^* t(i+\nu-\ell) \right) \right|^2}. \tag{10}$$

A. Max-SINR Single-User Equalizer

Using a Rayleigh-quotient argument with (13), we find that the equalizer maximizing the SINR of $\hat{b}_d(n)$ is

$$\boldsymbol{f}_{s,*}^{(\nu)} = \xi \left(\boldsymbol{H}\boldsymbol{H}^H - \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H + \frac{\sigma_w^2}{\sigma_t^2}\boldsymbol{I} \right)^{-1} \boldsymbol{h}_\nu \tag{14}$$

where $\xi \in \mathbb{C}$ is an arbitrary scaling. Applying the matrix inversion lemma to (14) yields an equivalent expression

$$\boldsymbol{f}_{s,*}^{(\nu)} = \xi' \left(\boldsymbol{H}\boldsymbol{H}^H + \frac{\sigma_w^2}{\sigma_t^2}\boldsymbol{I} \right)^{-1} \boldsymbol{h}_\nu \tag{15}$$

where $\xi' \in \mathbb{C}$ is an arbitrary scaling.

B. MMSE Single-User Equalizer

Although $\boldsymbol{f}$ equalizes the chip-rate signal, we ultimately want that the descrambled/despread equalizer output yields accurate *symbol* estimates. The corresponding MSE cost is

$$\hat{J}_u^{(d,\nu)} = \mathrm{E}\left| \hat{b}_d(n) - b_d(n) \right|^2$$

which is minimized [7] by

$$\boldsymbol{f}_u^{(d,\nu)} = \frac{P_d}{\eta(P_d - \sigma_t^2)} \left(\boldsymbol{H}\boldsymbol{H}^H + \frac{\sigma_w^2}{\sigma_t^2}\boldsymbol{I} \right)^{-1} \boldsymbol{h}_\nu \tag{16}$$

where $\eta = \boldsymbol{h}_\nu^H \left(\boldsymbol{H}\boldsymbol{H}^H + (\sigma_w^2/\sigma_t^2)\boldsymbol{I} \right)^{-1} \boldsymbol{h}_\nu$. Note that (16) is max-SINR according to (15).

C. MMSE Transmitted-Sequence Equalizer

The MMSE coefficient vector for estimation of the ν-delayed chip-rate transmitted sequence $\{t(i-\nu)\}$ is given by

$$\boldsymbol{f}_{t,*}^{(\nu)} = \left(\boldsymbol{H}\boldsymbol{H}^H + \frac{\sigma_w^2}{\sigma_t^2}\boldsymbol{I} \right)^{-1} \boldsymbol{H}\boldsymbol{e}_\nu \tag{17}$$

following a straightforward minimization of the ν-delay MSE cost [16]

$$J_t^{(\nu)} = \mathrm{E}\left| \boldsymbol{f}^H \boldsymbol{r}(i) - t(i-\nu) \right|^2. \tag{18}$$

Note that $\boldsymbol{f}_{t,*}^{(\nu)}$ is also max-SINR according to (15).

If $\{t(i)\}$ was known to the receiver (say, via decision direction), an empirical version of (18) could be constructed to optimize the equalizer coefficients. We revisit this idea in Section V.

D. MMSE Pilot Equalizer

From A3), a chip-spaced pilot-referenced error signal can be constructed as the difference between the descrambled equalizer output and a constant reference γ, yielding the cost

$$J_p^{(\nu)} = \mathrm{E}\left| s^*(i-\nu)\boldsymbol{f}^H \boldsymbol{r}(i) - \gamma \right|^2 \tag{19}$$

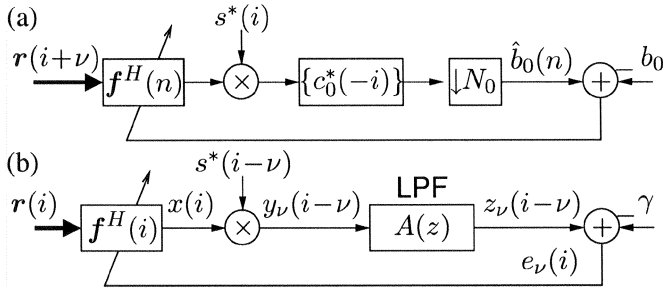


Fig. 5. Equalizer adaptation at (a) symbol-rate using matched-filtering and (b) chip-rate using MAI filtering.

which has the MMSE solution [17]

$$\boldsymbol{f}_{p,*}^{(\nu)} = \gamma^* \frac{b_0}{\sigma_t^2 \sqrt{N_0}} \left(\boldsymbol{H}\boldsymbol{H}^H + \frac{\sigma_w^2}{\sigma_t^2}\boldsymbol{I} \right)^{-1} \boldsymbol{H}\boldsymbol{e}_\nu. \tag{20}$$

Note that $\boldsymbol{f}_{p,*}^{(\nu)}$ is max-SINR according to (15) and that $\boldsymbol{f}_{p,*}^{(\nu)} = \boldsymbol{f}_{t,*}^{(\nu)}$ when $\gamma = \sigma_t^2\sqrt{N_0}/b_0^*$.

IV. Pilot-Aided Adaptive Equalization

In typical *symbol*-rate pilot-aided adaptive equalization schemes (e.g., [7], [9]), the descrambled equalizer output is matched-filtered according to the pilot code, and the result is compared to the pilot symbol b_0, as shown in Fig. 5(a). The error signal can then be used to adjust the equalizer at the symbol rate. Because the pilot code is orthogonal to all user codes, this error signal will be free of MAI when perfect equalization (i.e., $q_i = \delta_{i-\nu}$) is achieved. Without perfect equalization, however, the multipath-corrupted user signals will be nonorthogonal, and the equalizer updates will be MAI-corrupted.

When temporal channel variation is significant, it may be advantageous to update the equalizer at the *chip* rate rather than the symbol rate. Doing so would eliminate the possibility for matched filtering, however, which means that the error signal could become significantly MAI-corrupted. Because the chip-rate pilot signal is constant and the MAI is zero mean, though, this MAI could be attenuated through lowpass filtering [as in Fig. 5(b)]. Such lowpass filtering, however, could slow the reaction of the update algorithm to variations in the channel response. Thus, the optimal lowpass filter design will be a function of fading rate and user load. In the sequel, we perform a detailed study of these issues and outline an update algorithm that automatically adjusts to its operating environment.

A. Averaged-Error/Averaged-Regressor LMS

From Fig. 5(b), the error signal $e_\nu(i) = z_\nu(i-\nu) - \gamma$ yields the (delayed) instantaneous squared-error cost

$$\hat{J}_a(i-\nu) = |e_\nu(i)|^2 = |z_\nu(i-\nu) - \gamma|^2. \tag{21}$$

Stochastic gradient descent of the mean-squared cost $\mathrm{E}[\hat{J}_a(i)]$ can be accomplished using the LMS algorithm [11] with chip-rate update equation

$$\boldsymbol{f}(i+1) = \boldsymbol{f}(i) - \mu \cdot \nabla_{\boldsymbol{f}(i)} \hat{J}_a(i-\nu). \tag{22}$$

We derive $\nabla_{\boldsymbol{f}(i)}\hat{J}_a(i-\nu)$ below for two lowpass filter structures and coin the resulting algorithm "averaged-error/averaged-regressor LMS" (AEAR-LMS).[3] To reduce notation, we omit the subscripts on $z_\nu(i)$, $y_\nu(i)$, and $e_\nu(i)$.

1) Recursive Averaging: Suppose $A(z) = \zeta/1-G(z)$, where ζ is a constant, where $G(z)=\sum_{j=1}^{L_g} g_j z^{-j}$, and where $\{g_j\}$ and ζ are real-valued. Then, $z(i)$ can be obtained recursively:

$$z(i)=\zeta\sum_{\ell=0}^{L_f}\boldsymbol{f}_\ell^H(i+\nu)\boldsymbol{r}_{i+\nu-\ell}s^*(i)+\sum_{j=1}^{L_g}g_j z(i-j).$$

Next, we realize from (21) that

$$\nabla_{\boldsymbol{f}_\ell(i+\nu)}\hat{J}_a(i)=\big(z(i)-\gamma\big)^*\nabla_{\boldsymbol{f}_\ell(i+\nu)}z(i)$$

where $\nabla_{\boldsymbol{f}}$ denotes the complex-valued gradient operator with respect to the vector $\boldsymbol{f}$. For convenience, we define

$$\boldsymbol{\alpha}_\ell(i+\nu):=\nabla_{\boldsymbol{f}_\ell(i+\nu)}z(i),\quad \ell\in\{0,\dots,L_f\}.$$

If we assume (e.g., due to small μ) that $\boldsymbol{f}_\ell^*(i+\nu)\approx\boldsymbol{f}_\ell^*(i+\nu-j)$, for $j\in\{1,\dots,L_g\}$, then

$$\begin{aligned}\nabla_{\boldsymbol{f}_\ell(i+\nu)}z(i-j)&\approx\nabla_{\boldsymbol{f}_\ell(i+\nu-j)}z(i-j)\\&=\boldsymbol{\alpha}_\ell(i+\nu-j)\quad j\in\{1,\dots,L_g\}\end{aligned}$$

which results in the recursion

$$\boldsymbol{\alpha}_\ell(i+\nu)=\zeta\boldsymbol{r}_{i+\nu-\ell}s^*(i)+\sum_{j=1}^{L_g}g_j\boldsymbol{\alpha}_\ell(i+\nu-j).\qquad(23)$$

Practically speaking, $\boldsymbol{\alpha}_\ell(i)$ can be computed by delaying the received signal $\{\boldsymbol{r}_i\}$ by ℓ, descrambling with $\{s^*(i-\nu)\}$, and filtering the result with $A(z)$. With $\boldsymbol{\alpha}(i):=[\boldsymbol{\alpha}_0^T(i),\dots,\boldsymbol{\alpha}_{L_f}^T(i)]^T$, we can write the equalizer update

$$\boldsymbol{f}(i+1)=\boldsymbol{f}(i)-\mu\cdot\boldsymbol{\alpha}(i)\big(z(i-\nu)-\gamma\big)^*.\qquad(24)$$

Adopting the single-pole lowpass characteristic $A(z)=1-\rho/1-\rho z^{-1}$ reduces the filter design to a single parameter ρ and simplifies the updating procedure to

$$\boldsymbol{\alpha}(i)=(1-\rho)\boldsymbol{r}(i)s^*(i-\nu)+\rho\boldsymbol{\alpha}(i-1)\qquad(25)$$

$$\begin{aligned}e(i)=&(1-\rho)\big(\boldsymbol{f}^H(i)\boldsymbol{r}(i)s^*(i-\nu)-\gamma\big)\\&+\rho e(i-1)\end{aligned}\qquad(26)$$

$$\boldsymbol{f}(i+1)=\boldsymbol{f}(i)-\mu\boldsymbol{\alpha}(i)e^*(i).\qquad(27)$$

As is evident from (25)–(27), the incorporation of recursive MAI filtering yields an update algorithm that falls in the filtered-error/filtered-regressor LMS family [19]. In Appendix B, we show that the mean transient response of the single-pole algorithm (25)–(27) corresponds to that of a third-order dynamical system (in contrast to standard LMS, which exhibits the exponential decay of a first-order system [11]). Such higher order

[3]The derivation is similar to that of the filtered-X LMS algorithm [18] used in active noise control; however, the error filter of AEAR-LMS is intentionally inserted to enhance system performance.

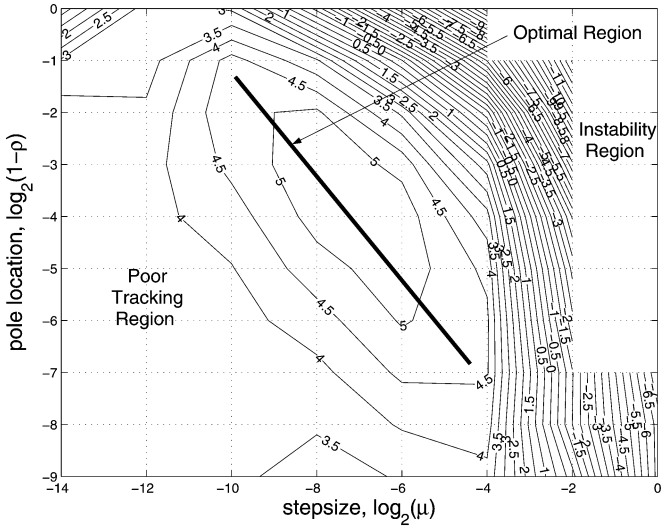


Fig. 6. SINR of symbol-estimates versus pole location ρ and step-size μ for AEAR-LMS. Mobile velocity is 5 km/hr, SNR $=$ 10 dB with system parameters as in Section VI.

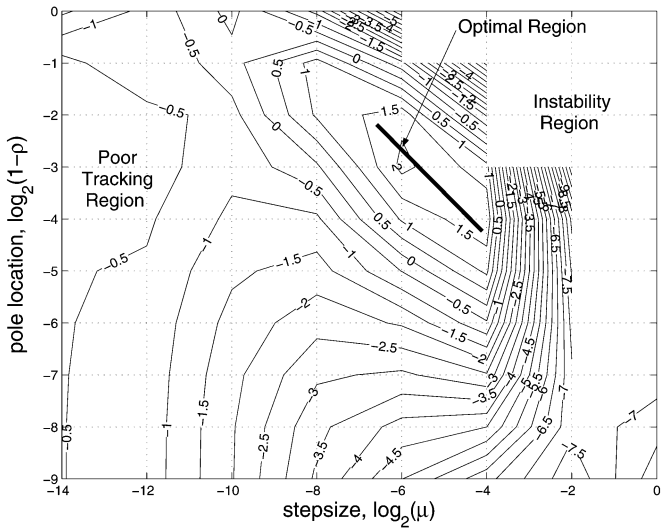


Fig. 7. SINR of symbol-estimates versus pole location ρ and step-size μ for AEAR-LMS. Mobile velocity is 120 km/hr, SNR $=$ 10 dB with system parameters as in Section VI.

LMS algorithms have demonstrated superior tracking performance in Rayleigh-fading channels [12].

The tracking performance of single-pole AEAR-LMS is a function of two adjustable parameters: μ and ρ. Simulation studies under various operating conditions suggest that fixing ρ (at a suitable value) and adjusting μ yields performance very close to that obtained through joint optimization of both parameters. (See Figs. 6 and 7.[4]) In Section IV-B, we derive a variable step-size algorithm that automatically adjusts μ to changes in user load or fading rate.

2) Sliding-Window Averaging: Using arguments similar those in the previous section, it can be shown that for the case

[4]Note that the performance of chip-rate LMS with no error filtering (upper border of Figs. 6 and 7, $\rho=0$) is lower than AEAR-LMS with optimal choice of μ and ρ.

of causal finite impulse response (FIR) $A(z) = \sum_{j=0}^{L_a} a_j z^{-j}$, (24) defines the equalizer update with $\boldsymbol{\alpha}(i)$ computed via

$$\boldsymbol{\alpha}(i+\nu) = \sum_{j=0}^{L_a} a_j \boldsymbol{r}(i+\nu-j)s^*(i-j). \tag{28}$$

B. Variable Step-Size Single-Pole Aear-LMS

In this section, we derive a variable step-size [20] version of the single-pole AEAR-LMS algorithm (25)–(27). The benefit of automatic step-size adjustment, as outlined in Section IV-AI, is automatic algorithm optimization under varying user load and/or fading rate.

We now introduce a time-dependent step-size μ_i. From (21), the gradient of the instantaneous cost with respect to μ_i is

$$\begin{aligned}\nabla_{\mu_i}\hat{J}_a(i-\nu) &= e(i)\nabla_{\mu_i}e^*(i) + e^*(i)\nabla_{\mu_i}e(i)\\ &= 2\Re\{\varepsilon(i)e^*(i)\}\end{aligned}$$

where $\varepsilon(i) := \nabla_{\mu_i} e(i)$ and $\Re\{\cdot\}$ returns the real part of the argument. If we assume $\mu_i \approx \mu_{i-1}$ and define $\boldsymbol{\psi}(i) := \nabla_{\mu_i}\boldsymbol{f}(i)$, then (26) implies that

$$\varepsilon(i) = (1-\rho)\boldsymbol{\psi}^H(i)\boldsymbol{r}(i)s^*(i-\nu) + \rho\varepsilon(i-1).$$

Similarly, (24) implies that

$$\begin{aligned}\boldsymbol{\psi}(i+1) &= \boldsymbol{\psi}(i) - \nabla_{\mu_i}\left[\mu_i\boldsymbol{\alpha}(i)e^*(i)\right]\\ &= \boldsymbol{\psi}(i) - \boldsymbol{\alpha}(i)e^*(i) - \mu_i\boldsymbol{\alpha}(i)\varepsilon^*(i).\end{aligned}$$

Hence, the step-size update equations become

$$\begin{aligned}\varepsilon(i) &= (1-\rho)\boldsymbol{\psi}^H(i)\boldsymbol{r}(i)s^*(i-\nu) + \rho\varepsilon(i-1)\\ \boldsymbol{\psi}(i+1) &= \boldsymbol{\psi}(i) - \boldsymbol{\alpha}(i)\left(e(i) + \mu_i\varepsilon(i)\right)^*\\ \mu_{i+1} &= \mu_i - \epsilon\Re\{\varepsilon(i)e^*(i)\}\end{aligned}$$

where ϵ is the step-size of the variable step-size algorithm and should be chosen much smaller than the operating range of $\{\mu_i\}$.

C. Steady-State Behavior in Slow Fading

In this section, we solve for the mean stationary points of AEAR-LMS, assuming a fixed (or slowly varying) channel. This will be accomplished by taking the expectation of (24) and examining the case $i \to \infty$. The results hold for either recursive or nonrecursive $A(z)$.

Consider $\boldsymbol{f}(\infty) := \lim_{i\to\infty} \mathrm{E}[\boldsymbol{f}(i)]$; then, from (24)

$$\boldsymbol{f}(\infty) = \boldsymbol{f}(\infty) - \lim_{i\to\infty} \mu\mathrm{E}[\boldsymbol{\alpha}(i+\nu)(z(i)-\gamma)^*]. \tag{29}$$

It is shown in Appendix A that, under suitable approximations

$$\boldsymbol{f}(\infty) = \frac{\gamma^* b_0 \sum_j a_j}{\sqrt{N_0}(1+\beta\eta)\sum_j a_j^2}\left(\sigma_t^2\boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2\boldsymbol{I}\right)^{-1}\boldsymbol{h}_\nu \tag{30}$$

where

$$\eta = \frac{1}{\sum_j a_j^2}\boldsymbol{h}_\nu^H\left(\sigma_t^2\boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2\boldsymbol{I}\right)^{-1}\boldsymbol{h}_\nu. \tag{31}$$

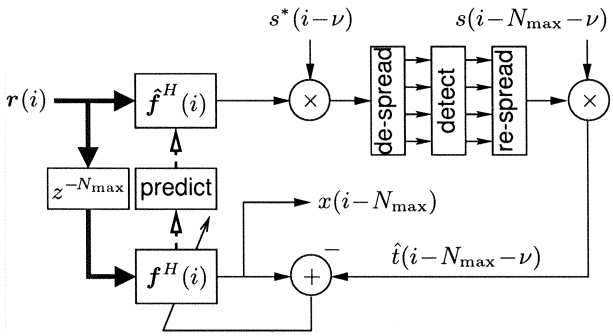


Fig. 8. Decision-directed chip-rate adaptive equalization.

From (30), we conclude that AEAR-LMS exhibits mean convergence to the max-SINR equalizer (15). Furthermore, with suitable choice of γ, AEAR-LMS ensures mean convergence to any of the MMSE equalizers discussed in Section III.

V. Decision-Directed Adaptive Equalization

The principle feature of the previously discussed AEAR-LMS algorithm is the use of lowpass filtering for error-signal-MAI attenuation. While a properly constructed lowpass filter significantly enhances equalizer performance in fading channels (recall Fig. 7), performance could be increased further if the MAI could be altogether *removed* from the error signal.

With this in mind, we propose a second mode of adaptation based on DD. In DD mode, the receiver makes hard tentative decisions on all active users' symbols and uses them to construct a (delayed, approximate) copy of the transmitted sequence. The transmitted sequence is then used to update the adaptive filter at the chip rate according to criterion $J_t^{(\nu)}$ from (18). Since it will not be possible to make reliable symbol estimates without a properly adjusted equalizer, we intend to engage the DD mode only after the pilot-trained mode (i.e., AEAR-LMS) has converged. Likewise, if the equalizer misconverges during DD mode (e.g., during a deep fade), the receiver switches back to pilot training. By misconverges, we mean that the equalizer loses lock and reconverges to a rotated signal constellation; the SINR is unaffected, but the BER approaches unity. The details of the DD mode and mode-switching mechanism are discussed below.

A. DD Equalizer Update

Because we are interested in a low-complexity receiver, we propose a simple symbol estimation procedure: 1) Descramble the equalizer output, 2) compute a matched-filter output for each active user, and 3) quantize the matched-filter outputs. The latter two operations are denoted by "despread" and "detect" in Fig. 8. The hard symbol estimates and spreading codes are then used to regenerate a (delayed, approximate) copy of the multiuser sequence $\hat{v}(i-N_{\max}-\nu)$, which is rescrambled to yield $\hat{t}(i-N_{\max}-\nu)$.

Note that the joint decision making process incurs a delay of $N_{\max}$ chips, where $N_{\max}$ denotes the spreading gain of the lowest-rate user. We saw in Section III that MMSE equalization using reference signal $\hat{t}(i-\nu)$ has the same solution (up

to a scaling) as pilot-based adaptation using the descrambling signal $\{s^*(i-\nu)\}$. Thus, to be consistent with pilot adaptation, we must delay $\boldsymbol{r}(i)$ by $N_{\max}$ chips when performing DD adaptation based on $\hat{t}(i-N_{\max}-\nu)$. This choice implies the need for two equalizers: an $N_{\max}$-delayed equalizer $\boldsymbol{f}(i-N_{\max})$, which is adaptively updated, and a "tentative" equalizer $\hat{\boldsymbol{f}}(i)$, which is used to generate the symbol estimates (see Fig. 8). $\hat{\boldsymbol{f}}(i)$ can be computed using an $N_{\max}$-step forward prediction of $\boldsymbol{f}(i-N_{\max})$. Arguing that, for typical mobile velocities, the equalizer taps experience relatively little change over $N_{\max}$ chips, the prediction can be approximated by simply copying $\boldsymbol{f}(i-N_{\max})$ to $\hat{\boldsymbol{f}}(i)$. The final symbol estimates should be computed from the output of $\boldsymbol{f}(i-N_{\max})$ if the complexity/delay can be tolerated.

The DD update can be constructed from standard LMS:

$$\begin{aligned} e(i) &= \boldsymbol{f}^H(i)\boldsymbol{r}(i) - \hat{t}(i-\nu) \\ \boldsymbol{f}(i+1) &= \boldsymbol{f}(i) - \mu \boldsymbol{r}(i)e^*(i) \\ \hat{\boldsymbol{f}}(i+N_{\max}+1) &= \boldsymbol{f}(i+1) \end{aligned}$$

with the variable step-size option [20] (after $\mu \to \mu_i$):

$$\begin{aligned} \mu_{i+1} &= \mu_i - \epsilon \mathrm{Re}\{\boldsymbol{\psi}^H(i)\boldsymbol{r}(i)e^*(i)\} \\ \boldsymbol{\psi}(i+1) &= \left[\boldsymbol{I} - \mu_i \boldsymbol{r}(i)\boldsymbol{r}^H(i)\right]\boldsymbol{\psi}(i) - \boldsymbol{r}(i)e^*(i). \end{aligned}$$

B. Mode-Switching Mechanism

To propose a switching mechanism, we must first understand the effect of decision errors on the DD algorithm. Let us denote the kth-user's symbol-error signal by $\Delta_k(n) = \hat{b}_k(n) - b_k(n)$ (for $k \neq 0$). As a simple example, we consider the case of binary phase shift keying (BPSK) symbols and uniform user power $P_k = 1$ so that $|b_k(n)| = 1$, and $|\Delta_k(n)| \in \{0, 2\}$.

With 25% symbol error rate, the nonzero case of $|\Delta_k(n)|$ occurs 25% of the time, implying $\mathrm{E}[|\Delta_k(n)|^2] = 1$. In this case, lack of user correlation A4) and the code normalization A2) imply that the kth user's symbol errors contribute an MAI energy of N_k^{-1} to the chip-rate DD error signal. In the pilot-aided algorithm and under the same assumptions, the kth user's signal also contributes MAI energy N_k^{-1} to the chip-rate error signal prior to lowpass filtering (and less after lowpass filtering). Roughly speaking, then, the DD algorithm should have about the same level of MAI in its error signal as the pilot-aided algorithm when BER ≈ 0.25; at lower BER, we expect the DD algorithm to have a cleaner error signal. This suggests that BER ≈ 0.25 forms a rough switching threshold between the pilot-aided and DD algorithms. Numerical simulations in Section VI confirm that the BER-optimal switching threshold is not far from BER $= 0.25$.

In practice, the code-multiplexed pilot symbols are used by the receiver to generate the BER switching signal. The receiver divides the number of pilot-symbol errors during the last slot by the number of pilot symbols to calculate the pilot-BER estimate, which is then compared to the switching threshold [see Fig. 9(b)].

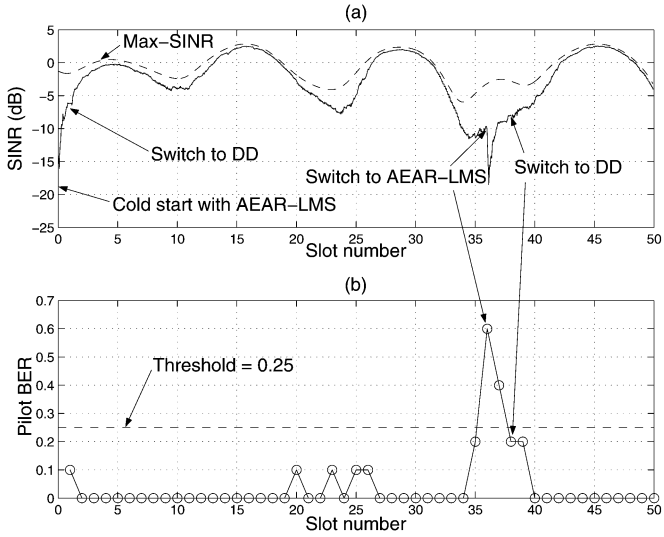


Fig. 9. (a) SINR trajectory with 0 dB SNR and threshold set at BER $= 25\%$. (b) Pilot BER during slot. Max SF $= 256$.

VI. Simulation Results

In all simulations, we assume a 1/2-chip spaced, 1/2-loaded, scrambled DS-CDMA downlink. In Figs. 9–13, the downlink consists of one user at each of the following spreading factors: $\{4, 8, 16, 32, 64, 128, 256\}$; users transmit with unit symbol energy, and the pilot has spreading factor of 256 with symbol energy four times that of a single user. In Figs. 14 and 15, the downlink consists of one user at each of the following spreading factors: $\{4, 8, 16, 32\}$; users and pilot transmit with unit symbol energy, and the pilot has spreading factor 32. Spreading factors and codes are chosen to mimic the WCDMA standard [14], and the transmitted chip sequence is overlaid with random quadrature phase shift keying (QPSK) scrambling. A Rayleigh-fading channel—generated according to Jake's model [21]—is used where the chip-spaced rays are uncorrelated and have power profile $\{0, -3, -6, -9\}$ dB with total average power equal to one. Delay spread of the fractionally spaced channel is approximately 20 chips and the equalizers have 50 taps to span 25 chips. The system delay ν is set to 21 chips. The variable step-size algorithm update weights ϵ were BER-optimized experimentally and are 2^{-15}, 2^{-18}, and 2^{-20} for AEAR-LMS, DD, and symbol-rate LMS, respectively. Unless otherwise noted, velocity is 60 km/hr, chipping rate is 3.84 Mcps, carrier frequency is 2 GHz, and the square-root raised-cosine chip waveform has excess bandwidth 0.22. The performances shown in Figs. 10–15 are averaged across users.[5].

Fig. 9 shows a typical trajectory of the SINR and BER across data slots (with ten pilot symbols per slot). Pilot training was employed at startup, and after a relatively rapid convergence, the receiver switched to DD mode. The deep fade beginning at slot 34 caused the DD algorithm to lose lock and reconverge to a phase-rotated constellation. The resulting spike in pilot BER cued the receiver to switch back to pilot training, causing a brief drop in SINR, while the equalizer readjusted its phase to finally suppress pilot BER and thereby allow a switch back to

[5]Although we target single-user reception, performance varies with spreading factor.

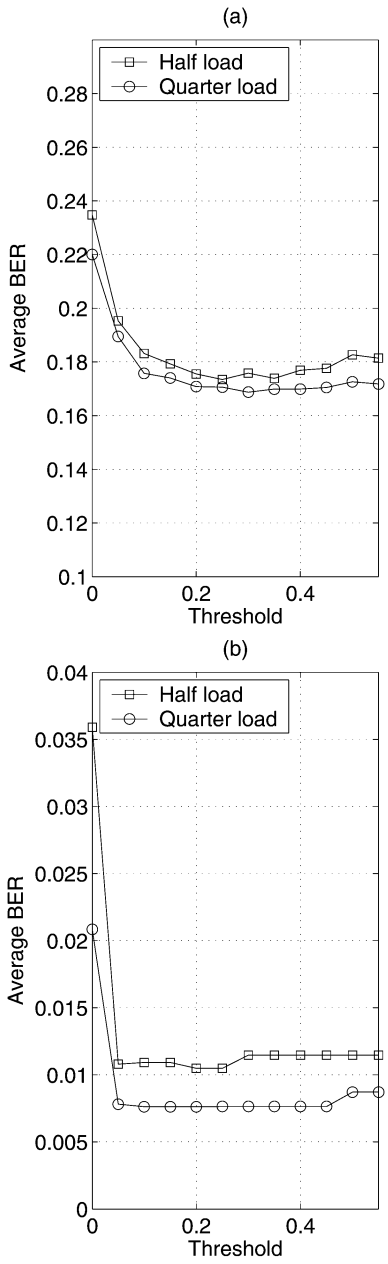


Fig. 10. Pilot BER versus switching threshold for 1/4- and 1/2-loaded cell in (a) SNR $= 0$ dB and (b) SNR $= 10$ dB. Max SF $= 256$.

DD mode. The near-optimality of the 0.25 pilot-BER threshold suggested in Section V is verified numerically in Fig. 10, where BER performance is plotted versus threshold. (The 1/4-loaded system was obtained by removing the user with spreading factor 4.)

Figs. 11 and 12 show SINR and BER performance, respectively, versus SNR at a fixed mobile speed of 60 km/hr. Fig. 13 shows both SINR and BER versus mobile speed at a fixed SNR of 10 dB. From these figures, we make the following observations. The symbol-rate LMS algorithm outlined in Fig. 5(a), although implemented with an adjustable step-size [20], performed poorly at all but the lowest velocities because of the low rate of adaptation (once every 256 chips—the maximum spreading factor). The same is true for symbol-rate recursive least squares (RLS) updating, where the RLS algorithm has

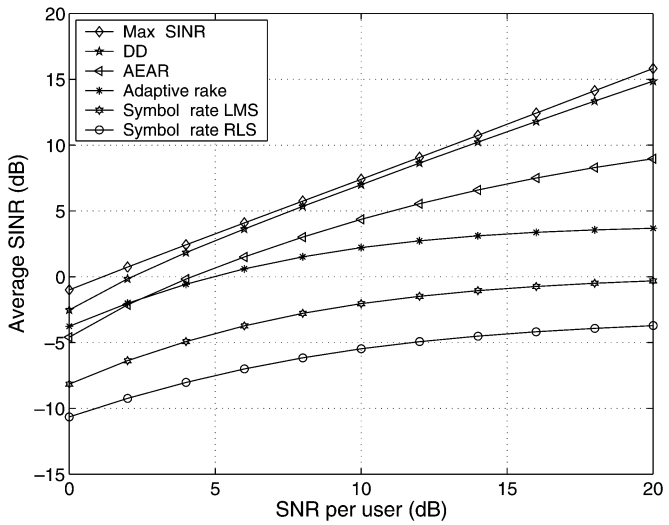


Fig. 11. Average SINR versus SNR (60 km/hr). Max SF $= 256$.

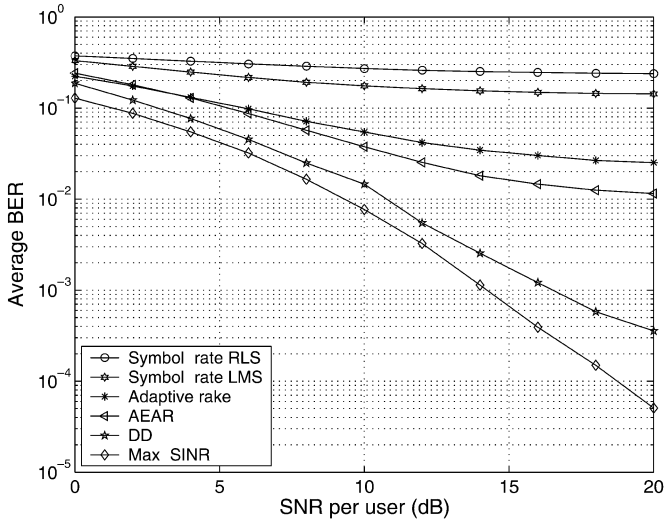


Fig. 12. Average uncoded BER versus SNR (60 km/hr). Max SF $= 256$.

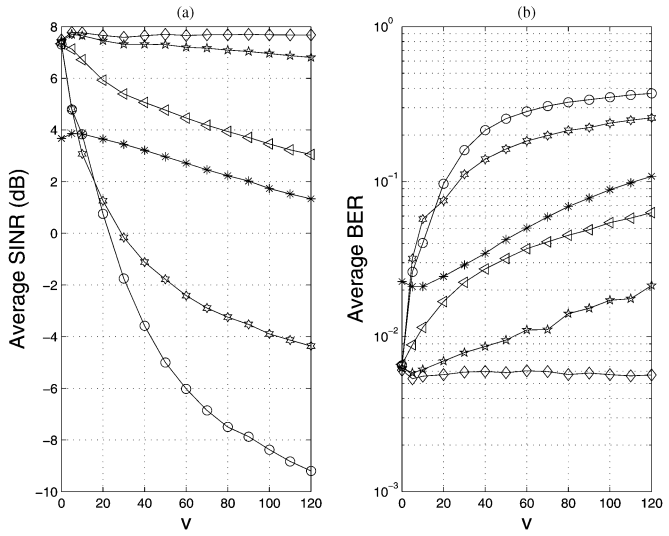


Fig. 13. (a) Average SINR and (b) Average BER versus velocity. SNR $= 10$ dB. Max SF $= 256$. (Legend as in Fig. 11).

BER-optimized forgetting factor and trains with the same error signal as the symbol-rate LMS algorithm. We note that the

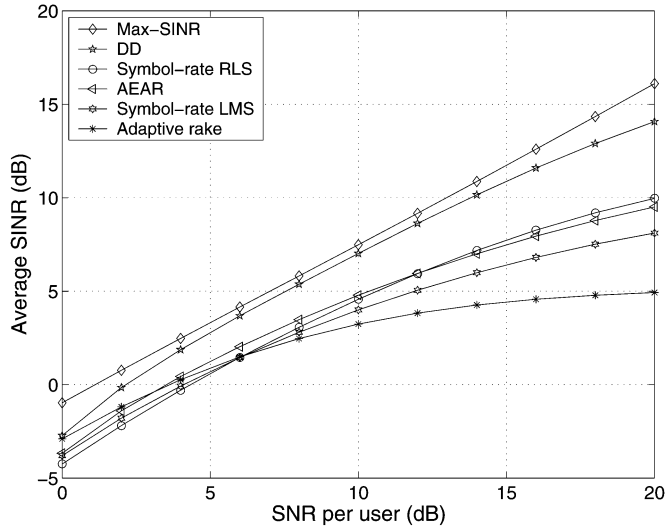


Fig. 14. Average SINR versus SNR (60 km/hr). Max SF $= 32$.

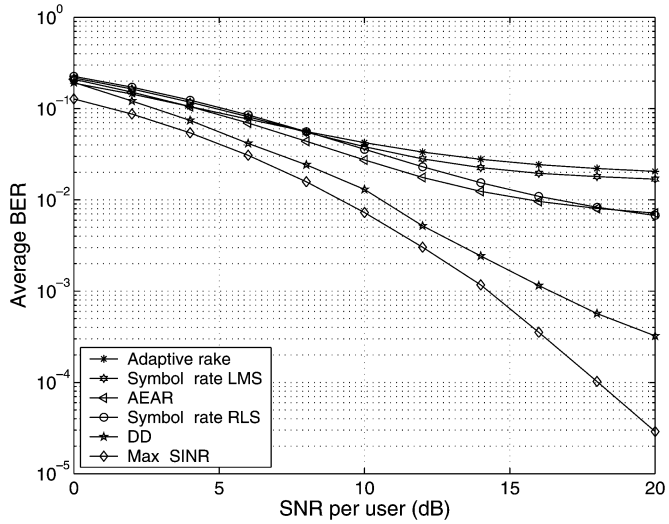


Fig. 15. Average uncoded BER versus SNR (60 km/hr). Max SF $= 32$.

pilot code[6] length used in symbol-rate updating must be set to the maximum spreading factor to remain orthogonal to all possible user codes. Thus, a lower maximum spreading factor would allow faster symbol-rate updates to enhance tracking performance and thereby improve BER and SINR performances. (This is illustrated in Figs. 14 and 15, where both symbol-rate LMS and RLS show marked improvement when symbol-rate updates come every 32 chips.) The adaptive Rake receiver[7] exhibited a more graceful degradation with mobile speed than did symbol-rate LMS. However, at all but the lowest SNRs, it performed far from optimal due to inherently high levels of MAI [recall (9)]. The variable step-size single-pole AEAR-LMS from Section IV-B (using $\rho = 1 - 2^{-4}$) performs on par with the adaptive Rake receiver at low SNR but much better at moderate and high SNRs. Still, there is a sizeable gap between AEAR-LMS and max-SINR performance. DD adaptation (with a pilot-BER switching-threshold of 0.25) performs significantly better than AEAR-LMS and, in fact, approaches the performance of max-SINR equalization over a wide range of SNRs and velocities. Recall that the max-SINR equalizer assumes knowledge of the actual time-varying channel coefficients, whereas the adaptive algorithms do not.

[6]Recall from A3) that the pilot is spread with a normalized all 1's code.

[7]The adaptive Rake employed pilot-based channel estimation in which descrambled pilot-matched-filter outputs were averaged using single-pole filters, whose pole locations were BER-optimized through simulation.

VII. Conclusion

We proposed a two-mode adaptive linear equalizer that updates at the chip rate to enhance tracking of time-varying parameters. The first mode (AEAR-LMS) is pilot trained and intended for cold-start and loss-of-lock situations. It uses a novel filtering mechanism to reduce MAI in the equalizer update. We focused on the case of single-pole filters for reasons of complexity and showed that these give rise to a third-order LMS algorithm, which is known for improved tracking in nonstationary environments. Optimum performance of the AEAR-LMS algorithm depends jointly on the pole placement and step-size value; however, experiments have shown that carefully choosing a fixed pole location and automatically adjusting the step-size can lead to near-optimal performance. Analysis showed that AEAR-LMS converges in the mean to the max-SINR equalizer.

After convergence, the receiver switches to the second (DD) mode. The DD algorithm improves tracking by canceling MAI in the error signal used to update the equalizer. Pilot decision quality is monitored as a means of switching between AEAR-LMS and DD.

We are currently investigating nonlinear processing stages that further improve on the linear processing reported in this paper, as well as algorithms that detect the set of currently active user codes (for applications in which this information is not known by the receiver).

Appendix A
Steady-State Behavior of AEAR-LMS

Making use of (11) and $\mathrm{E}[s^*(i-j)s(i-\ell)] = \delta_{j-\ell}$, it can be shown that[8]

$$\mathrm{E}\{\boldsymbol{\alpha}(i+\nu)z^*(i)\} = \sum_j \sum_\ell a_j a_\ell \Big(\sigma_t^2 \boldsymbol{H}\boldsymbol{H}^H \delta_{j-\ell} + \sigma_w^2 \boldsymbol{I}\delta_{j-\ell} + (1-\delta_{j-\ell})\boldsymbol{h}_\nu \boldsymbol{h}_\nu^H \mathrm{E}[v(i-j)v^*(i-\ell)] \Big) \mathrm{E}[\boldsymbol{f}(i+\nu-\ell)]. \quad (32)$$

From A3) and A4), it can be shown that

$$\sum_j \sum_\ell a_j a_\ell \mathrm{E}[v(i-j)v^*(i-\ell)] = \frac{P_0}{N_0}\Big(\sum_j a_j\Big)^2 + \mathrm{E}\,|\tilde{u}(i)|^2 \quad (33)$$

where $\tilde{u}(i) := \sum_j a_j u(i-j)$. Although the multiuser signal $\{u(i)\}$ is nonwhite and cyclostationary with period $N_{\max} := \max_k N_k$, filtering $\{u(i)\}$ with lowpass $A(z)$ tends to smooth the variations in $\mathrm{E}|\tilde{u}(i)|^2$ over the cyclostationarity

[8]Some derivation details have been omitted due to space limitations; for a more complete development, see [22].

period so that $\mathrm{E}|\tilde{u}(i)|^2 \approx \sigma_{\tilde{u}}^2$ for some fixed $\sigma_{\tilde{u}}^2$. In addition, we have observed (experimentally) that $\mathrm{E}|\tilde{u}(i)|^2$ is small relative to $(P_0/N_0)\left(\sum_j a_j\right)^2$, which suggests that errors in the approximation of $\mathrm{E}|\tilde{u}(i)|^2$ should be negligible. If we substitute (33) into (32) and consider $i \to \infty$, we obtain

$$\lim_{i\to\infty} \mathrm{E}\{\boldsymbol{\alpha}(i+\nu)z^*(i)\} \approx \left(\boldsymbol{B} + \beta \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H\right)\boldsymbol{f}(\infty) \tag{34}$$

where $\boldsymbol{f}(\infty) := \lim_{i\to\infty} \mathrm{E}[\boldsymbol{f}(i)]$, and

$$\boldsymbol{B} := \sum_j a_j^2 \left(\sigma_t^2 \boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2 \boldsymbol{I}\right) \tag{35}$$

$$\beta := \frac{b_0}{\sqrt{N_0}}\Big(\sum_j a_j\Big)^2 + \sigma_{\tilde{u}}^2 - \sigma_t^2 \sum_j a_j^2. \tag{36}$$

Next, it is straightforward to show that

$$\mathrm{E}[\boldsymbol{\alpha}(i+\nu)] = \sum_j a_j \frac{b_0}{\sqrt{N_0}} \boldsymbol{h}_\nu. \tag{37}$$

Using (34)–(37), we can evaluate (29) for $i \to \infty$:

$$\begin{aligned}\boldsymbol{f}(\infty) &= \boldsymbol{f}(\infty) - \mu \lim_{i\to\infty} \mathrm{E}[\boldsymbol{\alpha}(i+\nu)(z(i)-\gamma)^*] \\ &= \boldsymbol{f}(\infty) - \mu\left[\left(\boldsymbol{B} + \beta \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H\right)\boldsymbol{f}(\infty) - \frac{\gamma^* b_0 \sum_j a_j}{\sqrt{N_0}} \boldsymbol{h}_\nu\right]\end{aligned}$$

which can be written

$$\left(\boldsymbol{B} + \beta \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H\right)\boldsymbol{f}(\infty) = \frac{\gamma^* b_0 \sum_j a_j}{\sqrt{N_0}} \boldsymbol{h}_\nu. \tag{38}$$

Applying the matrix inversion lemma to (38), we arrive at

$$\boldsymbol{f}(\infty) = \frac{\gamma^* b_0 \sum_j a_j}{\sqrt{N_0}(1+\beta\eta)\sum_j a_j^2} \left(\sigma_t^2 \boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2 \boldsymbol{I}\right)^{-1} \boldsymbol{h}_\nu \tag{39}$$

where

$$\eta = \frac{1}{\sum_j a_j^2} \boldsymbol{h}_\nu^H \left(\sigma_t^2 \boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2 \boldsymbol{I}\right)^{-1} \boldsymbol{h}_\nu. \tag{40}$$

Appendix B
Transient Behavior of AEAR-LMS

In this Appendix, we show that the AEAR-LMS algorithm behaves in the mean as a third-order system. Defining the mean parameter error vector $\tilde{\boldsymbol{f}}(i) := \mathrm{E}\{\boldsymbol{f}(i)\} - \boldsymbol{f}(\infty)$, we have

$$\begin{aligned}\tilde{\boldsymbol{f}}(i+\nu+1) = \tilde{\boldsymbol{f}}(i+\nu) &- \mu \mathrm{E}\{\boldsymbol{\alpha}(i+\nu)z^*(i)\} \\ &+ \mu\gamma^* \mathrm{E}\{\boldsymbol{\alpha}(i+\nu)\}.\end{aligned}$$

From (32), (37), and (38), we can write

$$\begin{aligned}\tilde{\boldsymbol{f}}(i+\nu+1) = \tilde{\boldsymbol{f}}(i+\nu) &- \mu\left(\sigma_t^2 \boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2 \boldsymbol{I} - \sigma_t^2 \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H\right) \\ &\times \sum_j a_j^2 \tilde{\boldsymbol{f}}(i+\nu-j) \\ &- \mu \frac{P_0}{N_0} \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H \sum_j a_j \sum_\ell a_\ell \tilde{\boldsymbol{f}}(i+\nu-\ell) \\ &- \mu \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H \sum_j a_j \sum_\ell a_\ell \mathrm{E}\{u(i-j)u^*(i-\ell)\} \\ &\times \tilde{\boldsymbol{f}}(i+\nu-\ell).\end{aligned} \tag{41}$$

The last term of (41) is simplified as follows:

$$\begin{aligned}&\sum_j a_j \sum_\ell a_\ell \mathrm{E}\{u(i-j)u^*(i-\ell)\}\tilde{\boldsymbol{f}}(i+\nu-\ell) \\ &\quad= \sum_j \sum_{\ell\neq j} a_j a_\ell \mathrm{E}\{u(i-j)u^*(i-\ell)\}\tilde{\boldsymbol{f}}(i+\nu-\ell) \\ &\qquad+ \sum_j a_j^2 \sigma_u^2 \tilde{\boldsymbol{f}}(i+\nu-j) \\ &\quad\approx \sum_j a_j^2 \sigma_u^2 \tilde{\boldsymbol{f}}(i+\nu-\ell)\end{aligned} \tag{42}$$

since the off-diagonal terms of the double-sum approximately cancel. Taking the z-transform of (41) and defining $\tilde{\boldsymbol{f}}(z) := \mathcal{Z}\{\tilde{\boldsymbol{f}}(i+\nu+1)\}$ yields

$$\begin{aligned}\tilde{\boldsymbol{f}}(z) = \tilde{\boldsymbol{f}}(z)z^{-1} &- \mu \frac{P_0}{N_0} \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H \sum_j a_j A(z)\tilde{\boldsymbol{f}}(z)z^{-1} \\ &- \mu\left(\sigma_t^2 \boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2 \boldsymbol{I} - \frac{P_0}{N_0} \boldsymbol{h}_\nu \boldsymbol{h}_\nu^H\right) A_2(z)\tilde{\boldsymbol{f}}(z)z^{-1}\end{aligned} \tag{43}$$

where $A(z) := \sum_j a_j z^{-j}$, $A_2(z) := \sum_j a_j^2 z^{-j}$, and where we have used the approximation (42). If we choose a single-pole averaging filter $A(z) = (1-\rho/1-\rho z^{-1})$, then $A_2(z) = ((1-\rho)^2/1-\rho^2 z^{-1})$. Clearing the denominators of (43) by multiplying both sides by $(1-\rho z^{-1})(1-\rho^2 z^{-1})$, we obtain, after rearranging terms

$$\left((1-\rho z^{-1})(1-\rho^2 z^{-1})(1-z^{-1})\boldsymbol{I} + \mu \boldsymbol{D} z^{-1}\right)\tilde{\boldsymbol{f}}(z) = \boldsymbol{0}$$

where $\boldsymbol{D} = \left(\sigma_t^2 \boldsymbol{H}\boldsymbol{H}^H + \sigma_w^2 \boldsymbol{I} - (P_0/N_0)\boldsymbol{h}_\nu \boldsymbol{h}_\nu^H\right)(1-\rho)^2 + (1-\rho)(P_0/N_0)\boldsymbol{h}_\nu \boldsymbol{h}_\nu^H \sum_j a_j$. Noting that $\boldsymbol{D}$ has eigendecomposition $\boldsymbol{D} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H$, we define the transformation $\tilde{\boldsymbol{g}}(z) := \boldsymbol{U}^H \tilde{\boldsymbol{f}}(z)$; thus, the characteristic equation of the mth coordinate of the transformation $\tilde{\boldsymbol{g}}(z)$ is

$$(1-\rho z^{-1})(1-\rho^2 z^{-1})(1-z^{-1}) + \mu\lambda_m z^{-1} = 0 \tag{44}$$

which is clearly third order.

REFERENCES

[1] P. Schniter and A. Margetts, "Adaptive chip-rate equalization of downlink multirate wideband CDMA," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, vol. 2, Nov. 2002, pp. 1228–1232.

[2] M. Honig, U. Madhow, and S. Verdu, "Blind adaptive multiuser detection," *IEEE Trans. Inf. Theory*, vol. 41, no. 4, pp. 944–960, Jul. 1995.

[3] K. Hooli, M. Juntti, M. J. Heikkila, P. Komulainen, M. Latva-Aho, and J. Lilleberg, "Chip-level channel equalization in WCDMA downlink," *EURASIP J. Applied Signal Process.*, vol. 2002, pp. 757–770, Aug. 2002.

[4] O. Prätor, C. Unger, A. Zoch, and G. P. Fettweis, "Performance of adaptive chip equalization for the WCDMA downlink in fast changing environments," in *Proc. IEEE Int. Symp. Spread Spectrum Techn. Applicat.*, vol. 1, Sep. 2002, pp. 273–277.

[5] S. Chowdhury and M. D. Zoltowski, "Adaptive MMSE equalization for wideband CDMA forward link with time-varying frequency selective channels," in *Proc. IEEE Int.. Conf. Acoust., Speech, Signal Process.*, vol. 3, 2002, pp. 2605–2608.

[6] T. P. Krauss and M. D. Zoltowski, "Downlink specific linear equalization for frequency selective CDMA cellular systems," *J. VLSI Signal Process.*, vol. 30, pp. 143–161, Jan. 2002.

[7] C. D. Frank and E. Visotsky, "Adaptive interference suppression for direct-sequence CDMA systems with long spreading codes," in *Proc. Allerton Conf. Commun., Contr., Comput.*, Sep. 1998.

[8] C. D. Frank, E. Visotsky, and U. Madhow, "Adaptive interference suppression for the downlink of a direct sequence CDMA system with long spreading sequences," *VLSI*, vol. 30, pp. 273–291, Jan. 2002.

[9] F. Petre, M. Moonen, M. Engels, B. Gyselinckx, and H. D. Man, "Pilot-aided adaptive chip equalizer receiver for interference suppression in DS-CDMA forward link," in *Proc. IEEE Veh. Technol. Conf.*, vol. 1, Sep. 2000, pp. 303–308.

[10] K. Hooli, M. Latva-Aho, and M. Juntti, "Performance evaluation of adaptive chip-level channel equalizers in WCDMA downlink," in *Proc. IEEE Int. Conf. Commun.*, vol. 6, 2001, pp. 1974–1979.

[11] S. Haykin, Ed., *Unsupervised Adaptive Filtering*. New York: Wiley, 1999.

[12] S. Gazor, "Prediction in lms-type adaptive algorithms for smoothly time varing environments," *IEEE Trans. Signal Process.*, vol. 47, no. 6, pp. 1735–1739, Jun. 1999.

[13] C. R. Johnson Jr, P. Schniter, T. Endres, J. Behm, D. Brown, and R. Casas, "Blind equalization using the constant modulus criterion: A review," *Proc. IEEE—Special Issue on Blind System Identification and Estimation*, vol. 86, no. 10, pp. 1927–50, Oct. 1998.

[14] "Third Generation Partnership Project,", TS GRAN 25.213, 2003.

[15] G. E. Bottomley, "Optimizing the rake receiver for the CDMA downlink," in *Proc. IEEE Veh. Technol. Conf.*, 1993, pp. 742–745.

[16] D. T. M. Slock and I. Ghauri, "Blind maximum SINR receiver for the DS-CDMA downlink," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, 2000, pp. 2485–2488.

[17] T. P. Krauss, W. J. Hillery, and M. D. Zoltowski, "MMSE equalization for the forward link in 3G CDMA: Symbol-level versus chip-level," in *Proc. IEEE Workshop Statist. Signal Array Process.*, Aug. 2000, pp. 18–22.

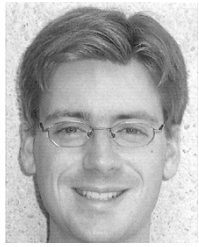
[18] S. M. Kuo and D. R. Morgan, *Active Noise Control: Algorithms and DSP Implementations*. New York: Wiley, 1996.

[19] W. A. Sethares, B. D. O. Anderson, and C. R. Johnson Jr, "Adaptive algorithms with filtered regressor and filtered error," *Math. Contr., Signals, Syst.*, vol. 2, pp. 381–403, Jul. 1989.

[20] A. Benveniste, M. M'etivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. Paris, France: Springer-Verlag, 1990.

[21] W. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: IEEE, 1993.

[22] A. R. Margetts, "Adaptive Chip-Rate Equalization of Downlink Multirate Wideband CDMA," Master's thesis, The Ohio State Univ., Columbus, OH, 2002.



Adam R. Margetts (S'98) received the B.S. degree (magna cum laude) in electrical engineering and mathematics from Utah State University, Logan, in 2000 and the M.S. degree in electrical engineering from The Ohio State University, Columbus, OH, in 2002, where is currently pursuing the Ph.D. degree.

His areas of research include adaptive equalization for wireless communications and signal processing for mobile ultra-wideband communication systems.

Mr. Margetts received a fellowship from the Dayton Area Graduate Studies Institute (DAGSI) while pursuing the M.S. degree and is supported in part during his Ph.D. studies by a fellowship from the Ohio Space Grant Consortium (OSGC).

Philip Schniter (M'92) received the B.S. and M.S. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1992 and 1993, respectively, and the Ph.D. degree in electical engineering from Cornell University, Ithaca, NY, in 2000.

From 1993 to 1996, he was with Tektronix Inc., Beaverton, OR, as a systems engineer. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus. His areas of research include signal processing for wireless communication systems and sensor networks.

Dr. Schniter received a Schlumberger Fellowship and an Intel Foundation Fellowship while pursuing the Ph.D. degree. He received the 1999 Prize Paper Award from the IEEE Energy Development and Power Generation Committee for work relating to his M.S. thesis. In 2003, he received a National Science Foundation CAREER Award.