Fast-Converging Blind Adaptive Channel-Shortening and Frequency-Domain Equalization

Richard K. Martin, Member, IEEE

Abstract—Orthogonal frequency-division multiplexing (OFDM) is a popular transmission format for emerging wireless communication systems, including satellite radio, various wireless local area network (LAN) standards, and digital broadcast television. Single-carrier cyclic-prefixed (SCCP) modulation is similar to OFDM, but with all frequency-domain operations performed at the receiver. Systems employing OFDM and SCCP perform well in the presence of multipath provided that the channel delay spread is shorter than the guard interval between transmitted blocks. If this condition is not met, a channel-shortening equalizer can be used to shorten the channel to the desired length. In modestly time-varying environments, an adaptive channel shortener is of interest. All existing adaptive channel shorteners require renormalization to restrain the channel shortener away from zero. In this paper, we study the use of a unit-tap constraint rather than a unit-norm constraint on the adaptive channel shortener. We use this constraint to manipulate existing algorithms into a framework analogous to the recursive least squares algorithm, and we develop adaptation rules for blind and semiblind frequency domain equalizers for SCCP receivers. Simulations of the proposed algorithms show an order of magnitude improvement in convergence speed, as well as a reduced asymptotic bit error rate.

Index Terms—Adaptive, blind, channel shortening, cyclic prefix, equalization, multicarrier.

I. INTRODUCTION

Cyclic-prefix fixed communication systems are block-based transmission systems that prefix each block with a copy of the end of the block. This transforms the linear convolution of the data and the channel into a circular convolution, enabling simple frequency-domain equalization (FDE). The most common type of cyclic-prefixed systems are those employing multicarrier modulation, which can take on two forms: in wireless systems, the modulation is called orthogonal-frequency-division multiplexing (OFDM), and in wireline systems, the modulation is called discrete multitone (DMT). Examples of wireless multicarrier systems include wireless local area networks (IEEE 802.11a/g, HIPERLAN/2, MMAC) [1], wireless metropolitan area networks (IEEE 802.16) [2], digital video and audio broadcasting in Europe [3], [4], satellite radio (Sirius and XM Radio) [5], and the proposed standard for multiband ultra wideband (IEEE 802.15.3a). Examples of wireline multicarrier systems include power line communications (HomePlug) [6] and digital subscriber lines (DSLs) [7].

Most channel-shortening designs make use of a norm constraint as part of the means of maximization of a generalized Rayleigh quotient [11]. All existing adaptive or iterative channel-shortening designs enforce the constraint by normalizing the filter after each update [13], [15], [22], [24]–[26]. In [27], an iterative algorithm was proposed to perform a generalized eigendecomposition; and this algorithm was applied to the channel-shortening problem in [28] and [29] in order to avoid the renormalization inherent to most iterative and A related emerging modulation format that also employs a cyclic prefix (CP) is single-carrier cyclic-prefixed (SCCP) modulation [8]–[10], also known as single-carrier frequency-domain equalization (SC-FDE). SCCP systems have many of the advantages of multicarrier systems without the high peak-to-average power ratio. Due to being proposed more recently, SCCP systems do not yet have the wide deployment of OFDM systems, but they are gaining popularity in the literature.

Cyclic-prefixed systems are very robust to multipath, provided that the delay spread of the transmission channel is less than the CP length. If the channel is short, then equalization of the channel can be done pointwise in the frequency domain by a bank of complex scalars. This is called a frequency-domain equalizer (FEQ). However, if the channel is longer than the CP, additional equalization is required. Typically, this takes the form of a channel-shortening equalizer (CSE), which is a filter at the front end of the receiver, designed with the intent that the effective channel is shorter than the CP (but not necessarily a single impulse) [11]. This paper considers adaptive design of the CSE and FEQ for both multicarrier and SCCP systems.

Channel shortening has a long history. It was applied to maximum-likelihood sequence estimation (MLSE) during the 1970s [12]–[14]. Starting in the mid-1990s, it was used to shorten the long impulse responses of twisted pair telephone lines in order to enable frequency-domain equalization in DSL [15]–[18]. While the initial designs were based on heuristic cost functions, more recent designs have focussed on maximizing the bit rate for a given bit error rate (BER) [19]–[21], which is the appropriate measurement of performance in wireline multicarrier systems that allow bit allocation across the subcarriers.

Since most of the channel-shortening literature has been developed in the context of DSL, most designs have high complexity and are not blind or adaptive. As wireless cyclic-prefixed systems become more popular, adaptive channel shorteners will become necessary. Adaptive channel shortening has been studied in [13], [22]–[25]. However, these designs are stochastic gradient descent algorithms similar to the least-mean-squares (LMS) algorithm, hence they are slow to converge. This paper proposes a method to improve the convergence time by several orders of magnitude.
Adaptive channel shorteners. However, the convergence rate of the algorithm is still slow [29].

This paper addresses two issues. First, we develop a blind adaptive recursive least squares (RLS)-like implementation of the Multicarrier Equalization by Restoration of Redundancy (MERRY) channel shortener that converges much faster than the previously proposed LMS-like MERRY algorithm [24]. Second, we develop blind adaptive FDEs for SCCP systems, which have not been investigated in the literature to date. Adaptive algorithms are of interest in wireless applications, which have not been investigated in the literature to date. Blind algorithms do not make use of knowledge of a transmitted training sequence. This is required in noncooperative environments (surveillance). Moreover, in time-varying environments, a training sequence would need to be frequently repeated, which reduces the bandwidth available for data.

The remainder of this paper is organized as follows. In Section II, we present the multicarrier and SCCP system models. In Section III, we propose a rapidly converging blind adaptive channel shortener. In Section IV, we propose several adaptation rules for the FEQ for SCCP systems. In Section V, we first investigate the asymptotic performance of the proposed CSE algorithm, then we compare the convergence speed of the proposed CSE algorithm to existing algorithms, and finally investigate the performance of the proposed FEQ adaptation rules. Section VI concludes the paper.

II. SYSTEM MODEL

In this section, we describe the multicarrier system model and the SCCP system model. We first discuss the two types of transmitters, then we discuss the channel model (which is the same for both cases), then we discuss the two types of receivers. In both cases, we assume a single-input multiple-output (SIMO) channel model, which can be achieved by employing multiple receive antennas and/or oversampling the received signal.

A typical multicarrier system is shown in Fig. 1. The complex finite-alphabet data symbols (usually multilevel QAM data) are blocked into groups of size \( N \). The \( k \)th such block is denoted \( X(k) \), and it is referred to as the frequency-domain transmitted data. Then, an inverse discrete Fourier transform (IDFT) is taken to get \( N \) time-domain samples. In practical systems, the DFT is implemented by the fast Fourier transform (FFT). A cyclic prefix is appended by copying the last \( \nu \) samples of the block to the beginning of the block, and then the \( M = N + \nu \) samples are transmitted serially. The \( i \)th transmitted data sample is denoted \( x(i) \). Note that \( k \) is the block index and \( i \) is the sample index.

The SCCP system model is shown in Fig. 2. The complex finite-alphabet data symbols are blocked into groups of size \( N \), with the \( k \)th block denoted \( X(k) \). Then, a cyclic prefix of length \( \nu \) is inserted at the start of each block, and the resulting data sequence \( x(i) \) is transmitted serially. Note that there is no IDFT at the transmitter in SCCP modulation, but otherwise the transmitter is identical to a multicarrier transmitter.

The method by which \( x(k) \) is generated (multicarrier or SCCP) does not matter, so long as the CP is present and so long as \( x(k) \) is otherwise uncorrelated over time. For both multicarrier and SCCP systems, once the CP is inserted, the transmitted signal obeys the relation

\[
x(Mk+i) \equiv x(Mk+i+N), \quad i \in \{1, \ldots, \nu\}, -\infty < k < \infty.
\]

(1)

The transmission channels \( h_1, h_2, \ldots, h_p \) are modeled as finite-impulse-response (FIR) filters of length \( L_h \), and the data received on the \( p \)th path \( r_p \) is given by

\[
r_p(i) = \sum_{j=0}^{L_h-1} h_p(j)x(j-i) + n_p(i).
\]

(2)

The noise sequences are assumed to be independent of each other and of the transmitted data, but each noise sequence is not necessarily white or Gaussian. The CSEs \( w_1, w_2, \ldots, w_p \)
are FIR filters of length $L_w$, hence $y_p$, the output of CSE $p$, is obtained via

$$ y_p(i) = \sum_{j=0}^{L_w-1} w_p(j)r_p(i - j) $$

(3)

$$ = \sum_{j=0}^{L_w-1} c_p(j)x(i - j) + \sum_{j=0}^{L_w-1} w_p(j)n_p(i - j). $$

(4)

The final output is obtained by

$$ y(i) = \sum_{p=1}^{P} y_p(i) $$

(5)

$$ = \sum_{j=0}^{L_w-1} c(j)x(i - j) + \sum_{j=0}^{L_w-1} \sum_{p=1}^{P} w_p(j)n_p(i - j). $$

(6)

where the effective channel (including all $P$ receive paths) of length $L_c$ is

$$ c(j) = \sum_{p=1}^{P} c_p(j), \quad j \in \{0, \ldots, L_c - 1\}. $$

(7)

After channel shortening, the cyclic prefix is removed from each block, a DFT is used to return the data to the frequency domain, and the FEQ is used to invert the channel in the frequency domain. At this point, multicarrier and SCCP systems differ. In multicarrier systems, the FEQ output is the estimate of the data $\hat{X}(k)$. In SCCP systems, one additional step is required: A DFT must be taken of the equalized frequency domain data to get the estimated data block $\hat{X}(k)$.

Note that for multicarrier systems, the channel input and output are not finite-alphabet data. For an SCCP system, the channel input is finite-alphabet, but even when the channel is short, the channel output is not finite alphabet, due to inter-symbol interference.

III. RLS-LIKE ADAPTIVE ALGORITHMS

In [24], the blind adaptive MERRY channel-shortening algorithm was proposed. The original implementation of MERRY was an LMS-like algorithm, although with a unit-norm constraint on the channel shortener. In this section, we reformulate the constrained minimization of the MERRY cost function from an eigenvector problem into a least-squares problem, and we introduce a pseudo-"desired" signal for use in creating an RLS-like algorithm to adaptively minimize the MERRY cost. However, unlike the traditional RLS algorithm, this algorithm is blind.

A note on Terminology: In this paper, the algorithm of [24] will be referred to as "MERRY-UNC," to emphasize the fact that it uses a unit-norm constraint (UNC). Although it was not explicitly considered in [24], it is also possible to extend MERRY to use a unit-tap constraint (UTC), and the variant of the method of [24] under this constraint will be termed "MERRY-UTC." The algorithm proposed in this paper will be referred to as "RLS-MERRY," or sometimes as "RLS-MERRY-UTC," to emphasize its use of the UTC.

A. RLS-MERRY

The MERRY algorithm [24], [29] is a blind adaptive channel shortener that exploits the redundancy in the cyclic prefix. Due to the presence of the CP, there are $\nu$ pairs of identical samples in the beginning and end of each transmitted block. Thus, convolution of a length $\nu$ channel with this data would lead to one pair of identical outputs. However, if the channel is longer than $\nu$ taps, this redundancy is corrupted. MERRY attempts to restore the redundancy by minimizing the MSE between the two samples which would be identical if the channel was short. Formally, the cost function is

$$ J_{\text{MERRY}} = E \left[ |y(Mk + \nu + \Delta) - y(Mk + \nu + N + \Delta)|^2 \right], $$

$$ \Delta \in \{0, \ldots, M - 1\}. $$

(8)

The parameter $\Delta$ is the transmission delay, and choosing $\Delta$ is equivalent to estimating the locations of the boundaries between successive blocks. (See [29] for a blind method for choosing $\Delta$ for the MERRY algorithm.)

The cost function (8) has an undesirable local minimum at $w = 0$, so it must be minimized subject to a constraint. Possible constraints are the unit-norm constraint (UNC) $w^H w = 1$ or the UTC $w_{00} = 1$. Al-Dhahir and Cioffi [17] compared the UNC and the UTC for the minimum mean-square error (MMSE) channel shortener, and proved that the MMSE channel shortener always has a superior (i.e., lower) MSE under the UNC. For this reason, all adaptive channel shorteners in the literature make use of the UNC rather than the UTC. However, this paper aims to show that the UTC has advantages for an adaptive implementation.

Define the difference vectors

$$ \tilde{r}_p(k) = \begin{bmatrix} r_p(Mk + \nu + \Delta) \\ \vdots \\ r_p(Mk + \nu + N + \Delta) - L_w \\ \vdots \\ r_p(Mk + \nu + N + \Delta - L_w) \end{bmatrix} $$

(9)

for $1 \leq p \leq P$, and define the "stacked" vectors:

$$ w^T = [w_1^T, w_2^T, \ldots, w_P^T], $$

$$ \tilde{r}^T(k) = [\tilde{r}_1^T(k), \tilde{r}_2^T(k), \ldots, \tilde{r}_P^T(k)]. $$

(10)

(11)

Note that

$$ y(Mk + \nu + \Delta) - y(Mk + \nu + N + \Delta) = w^T \tilde{r}(k), $$

(12)

Thus, MERRY under the UNC minimizes the Rayleigh quotient

$$ \frac{w^H A w}{w^H w}, \quad A = E \left[ \tilde{r}(k) \tilde{r}^T(k) \right]. $$

(13)

Thus, the MERRY algorithm (which alternates between gradient descent of (8) and projection onto the constraint space) computes the eigenvector of $A$ with minimum eigenvalue. As such, in its UNC form, it is unsuitable for least-squares algorithm acceleration techniques.
The difficulty with formulating a RLS-like algorithm is that there is no “desired” or “training” signal to compare to. However, consider the use of the UTC instead of the UNC. Truncate the vector \( \mathbf{w} \) by removing element \( i_0 \)

\[
\mathbf{w}_t = [w_0, w_1, \ldots, w_{i_0-1}, w_{i_0+1}, \ldots, w_{PL_w-1}]^T
\]

(14)

and similarly for \( \hat{\mathbf{r}}_t(k) \). For simplicity of notation, assume that of the \( P \) CSes, the one that has a tap constrained to unity is \( p = 1 \), so that 0 \( \leq i_0 \leq L_w - 1 \). There is no loss of generality here since the order of the channel numbering is arbitrary. The cost function under the UTC becomes

\[
J_{\text{MERRY}} = E\left[\mathbf{w}^T \hat{\mathbf{r}}(k) \tilde{H}(k) \mathbf{w}^*\right] = E\left[\mathbf{w}_t^T \tilde{\mathbf{r}}_t(k) + 1 \cdot \hat{\mathbf{r}}_t(i_0(k))\right]^2
\]

(16)

\[
\hat{J}_e = E\left[\mathbf{w}_t^T \tilde{\mathbf{r}}_t(k) - d(k)\right]^2
\]

(17)

where we have defined the “desired” signal

\[
d(k) = -\hat{\mathbf{r}}_t(k)
\]

\[
= r_1(Mk + \nu + N + \Delta - i_0) - r_1(Mk + \nu + \Delta - i_0).
\]

(18)

When the free parameter is \( \mathbf{w}_t \) rather than \( \mathbf{w} \), the cost function (17) is no longer constrained. Minimization of (17) over \( \mathbf{w}_t \) thus becomes a least-squares problem.

By substituting the difference signal (18) and the truncated vectors into the desired signal, data vector, and regressor vector in a standard RLS algorithm [30, pp. 120–124], we obtain an RLS update to the unconstrained portion of the CSE, as follows:

\[
e(k) = (r_1(Mk + \nu + N + \Delta - i_0)
\]

\[
- r_1(Mk + \nu + \Delta - i_0))
\]

\[
\mathbf{z}(k) = \mathbf{R}^{-1}(k)\tilde{\mathbf{r}}(k)
\]

(19)

\[
\hat{\mathbf{z}}(k) = \frac{\mathbf{z}(k)}{\rho + \tilde{\mathbf{r}}^H(k) \hat{\mathbf{r}}(k)}
\]

(20)

\[
\mathbf{w}(k+1) = \mathbf{w}(k) + \epsilon(k) \hat{\mathbf{z}}(k)
\]

(21)

\[
\mathbf{R}^{-1}(k+1) = \frac{1}{\rho} \left( \mathbf{R}^{-1}(k) - \hat{\mathbf{z}}(k) \hat{\mathbf{z}}^H(k) \right)
\]

(22)

The scalar \( \rho \) is the “forgetting factor”: \( \rho \approx 1 \) places equal emphasis on all data up to the present, whereas the smaller \( \rho \) becomes, the more emphasis is placed on recent data. The matrix \( \mathbf{R}^{-1}(0) \) is typically initialized as a scaled identity matrix \( \eta \mathbf{1}_{PL_w-1} \), where \( \eta \) is a large positive constant. Since the complete filter \( \mathbf{w} \) has one tap constrained to unity, the remaining coefficients will be initialized to the zero vector \( \mathbf{w}_0 = \mathbf{0}_{PL_w-1} \).

The computational cost of this algorithm is approximately \( 2(PL_w)^2 \) complex multiply-and-accumulate (MAC) operations per update, if the symmetry in (23) is exploited. The bulk of the complexity comes from computing \( \mathbf{z}(k) \) by (20) and (23). This complexity can be reduced to approximately 10(PL_w) to 20(PL_w) complex MACs per update; see [30, pp. 125–128], for example.

### B. Asymptotic Solution

Under the UTC, the optimization problem becomes a least-squares problem. In this section we derive the formula for the filter weights that RLS-MERRY will converge to as it adapts to solve this least-squares problem.

Regardless of the constraint, the MERRY cost function is a quadratic form as given in (13). The matrix \( \mathbf{A} \) can be shown to consist of \( P \times P \) blocks, with block \((p,q)\) of size \( L_w \times L_w \) given by [29]

\[
A_{p,q} = H_{\text{wall},p}^H H_{\text{wall},q}
\]

(24)

where \( H_{\text{wall},p} \) consists of rows \( \Delta \) to \( \Delta + \nu - 1 \) of the full convolution matrix for subchannel \( p \).

Once we consider the constraint, the product \( \mathbf{w}^H \mathbf{A} \mathbf{w} \) can be divided into portions coming from the constrained tap and from the unconstrained tap. Define \( \mathbf{w}_c \) as in (14), and similarly define \( \mathbf{A}_c \) to be \( \mathbf{A} \) with row \( i_0 \) and column \( i_0 \) removed. Let \( \mathbf{a}_c \) be column \( i_0 \) of \( \mathbf{A} \) with element \( i_0 \) removed, and let \( \alpha \) be element \((i_0, i_0)\) of \( \mathbf{A} \). Thus, \( \mathbf{A} \) has been partitioned into \( \mathbf{A}_c, \mathbf{a}_c, \mathbf{a}_c^H \), and \( \alpha \). Thus,

\[
J = \mathbf{w}^H \mathbf{A} \mathbf{w} = \mathbf{w}_c^H \mathbf{A}_c \mathbf{w}_c + \mathbf{a}_c^H \mathbf{w}_c + \mathbf{w}^H \mathbf{a}_c + \alpha
\]

(25)

Setting the gradient with respect to \( \mathbf{w}_c \) to zero and solving yields

\[
\mathbf{w}_c^{\text{opt}} = -A_c^{-1} \mathbf{a}_c
\]

(26)

and the resulting value of the MERRY cost is

\[
J_{\min} = \alpha - \mathbf{a}_c^H A_c^{-1} \mathbf{a}_c.
\]

(27)

Equation (26) is useful for computing asymptotic performance of the algorithm (which we will do in Section V-A), but it does not yield additional physical insight. The physical interpretation, which comes from (24), is that the energy in the “wall” portion of the SIMO channel (i.e., the excess length beyond the length of the CP) is being suppressed, subject to the constraint.

### C. RLS-MMSE

Although we are primarily interested in blind channel shorteners, for purposes of comparison we now consider an RLS implementation of the trained, adaptive MMSE channel shortener [13] operating under the UTC. The optimal nonadaptive solution for this structure is given by the MMSE channel shortener in [31] under the identity tap constraint (ITC) with the number of transmit antennas set to \( n_t = 1 \), but we focus here on an adaptive implementation via the RLS algorithm.

The time-domain portion of the system model for a SIMO MMSE channel shortener is depicted in Fig. 3. The idea is that the effective SIMO channel should approximate a target impulse response (TIR) of length \( \nu + 1 \), with a bulk delay of \( \Delta \) samples. The design parameters are the CSes \( \mathbf{w}_1, \ldots, \mathbf{w}_P \) and the TIR \( \mathbf{b} \). In [13], an LMS-like update was presented, which required three steps per update: gradient descent of \( \mathbf{w} \) for fixed \( \mathbf{b} \), gradient descent of \( \mathbf{b} \) for fixed \( \mathbf{w} \), and renormalization of \( \mathbf{b} \) to maintain the UNC. To implement an RLS-like algorithm, the problem needs to be cast as a least-squares problem with a suitable choice of “desired” signal. Although the TIR output could
be considered to be the desired signal, the gradient descent of $b$ and renormalization of $b$ will affect the correlation between the training signal $x$ and the desired signal. However, if the UTC is used, then the desired signal can be defined to be the contribution to the error signal from the tap that is constrained to unity. This leads to the desired signal being simply a delayed version of the transmitted signal $x$.

Mathematically, we wish to adaptively minimize the cost function

$$J_{\text{MMSE}} = E \left[ e(i)^2 \right]$$

where $e(i) = \left[ x(i - \Delta), \cdots, x(i - \Delta - \nu) \right] b$

subject to the constraint $b_0 = 1$ for some $i_0$. Define the vectors

$$v = \left[ w_{0}^T, \cdots, w_{M}^T \right]^T$$

$$s(i) = \left[ r_{0}^T(i), \cdots, r_{L}^T(i), -x(i - \Delta), \cdots, -x(i - \Delta - i_0 - 1), -x(i - \Delta - i_0 + 1), \cdots, -x(i - \Delta - \nu) \right]^T$$

and the desired signal

$$d(i) = x(i - \Delta - \nu).$$

Then, an RLS implementation is obtained by substituting (30), (31), and (32), into the standard RLS update as the parameter vector, regressor, and desired signal, respectively, leading to

$$e(i) = x(i - \Delta - \nu) - v^T s(i)$$

$$z(i) = R^{-1}(i)s(i)$$

$$\hat{z}(i) = \frac{z(i)}{\rho + s^T(i)z(i)}$$

$$v(i+1) = v(i) + e(i)\hat{z}(i)$$

$$R^{-1}(i+1) = \frac{1}{\rho} (R^{-1}(i) - \hat{z}(i)\hat{z}^H(i)).$$

Note that the MMSE channel shortener can update as frequently as once per sample, rather than once per block.

IV. ADAPTIVE FREQUENCY-DOMAIN EQUALIZATION

In a multicarrier system, the immediate output of each individual FEQ coefficient is expected to be from a finite-alphabet. Thus, adaptive FEQ algorithms could be derived as an LMS algorithm, a decision-directed LMS (DD-LMS) algorithm, or a constant modulus algorithm (CMA), with the update taking the form of a one-tap filter (i.e., a scalar) adapting based in its immediate output. Moreover, for 4-QAM signal constellations, the FEQ in a multicarrier system can be omitted entirely if differential encoding is used, since magnitude is not an issue for 4-QAM constellations and only phase differences matter when differential encoding is used.

In contrast, in an SCCP system, the FEQ output has not yet been completely processed, hence it is not expected to be from a finite-alphabet and the update rules are not as clear. For the same reason, the use of differential encoding will not remove the need for an FEQ. However, after taking an FFT of the entire FEQ output $N$-vector, we obtain an $N$-vector of data that should be finite-alphabet (assuming equalization has been accomplished). Thus, in an SCCP system, each individual FEQ tap must adapt based on some function of all $N$ post-FFT outputs. The goal of this section is to derive these blind and semiblind adaptation rules for the FEQ in an SCCP system.

If training is available, the FEQ can be adapted via LMS or RLS. If training is unavailable, the desired finite-alphabet nature of the final FFT output can be exploited to adapt the FEQ via a DD-LMS algorithm [32] or CMA [33]. In all cases, the training signal or the signal that should be finite alphabet does not occur at the FEQ output, but later on, after the IFFT, which complicates the update rules. Moreover, the equalizer is partially an element-by-element structure rather than a filter, which must be taken into account when deriving the update rules.

First, consider an LMS algorithm, which can be used to aid initial adaptation in a semiblind implementation. Denote the FEQ vector as $D$, with input $u(k)$ and output $\hat{u}(k)$. Let $F_N$ be the matrix that computes the FFT. Recall that $\hat{X}(k) = F_N^H \hat{u}(k)$ is the output vector after the final IFFT. Let $e_{\text{out}}(k) = X(k) - \hat{X}(k)$ be the $N \times 1$ error vector at the output of the entire system, temporarily assuming that training is available. Let the cost function for FEQ adaptation be the expected norm squared of this vector

$$J_{\text{MMSE}} = E \left[ e_{\text{out}}^H(k) e_{\text{out}}(k) \right]$$

$$= E \left[ (X(k) - \hat{X}(k))^H (X(k) - \hat{X}(k)) \right].$$

A stochastic gradient descent of the squared error with respect to the FEQ coefficient vector $D$ leads to

$$D_{\text{blur}}(k+1) = D_{\text{blur}}(k) + \mu u^*(k) \odot (F_N e_{\text{out}}(k))$$

where $\odot$ denotes element-by-element multiplication. The update can be simplified by noting that

$$F_N e_{\text{out}}(k) = (F_N X(k)) - \hat{u}(k)$$

Authorized licensed use limited to: IEEE Xplore. Downloaded on February 4, 2009 at 05:26 from IEEE Xplore. Restrictions apply.
so no FFTs are required if the training signal $F_N X(k)$ is stored in the frequency domain at the receiver.

An RLS implementation can be derived as well. Define the diagonal matrix $U(k)$ to have $u(k)$ as its main diagonal and zeros elsewhere

$$U(k) = \text{diag}[u(k)].$$

Then, the cost function (37) can be expanded as

$$J_{nse} = D^H E \left[ U^H(k) F_N F_N^H U(k) \right] D$$

$$- D^H E \left[ U^H(k) F_N X(k) \right] - E \left[ X(k) F_N^H U(k) \right] D$$

$$+ E \left[ X^H(k) X(k) \right].$$

Note that $R$ is diagonal. The FEQ that minimizes this cost is

$$D_{opt} = R^{-1} P$$

which can be computed recursively as

$$R(k) = \rho_{eq} \left( R(k-1) + \text{diag}[u^*(k) \odot u(k)] \right)$$

$$P(k) = \rho_{eq} \left( P(k-1) + u^*(k) \odot (F_N X(k)) \right)$$

$$D_{rls} = R^{-1}(k) P(k)$$

where $\rho_{eq}$ is a “forgetting factor” that is slightly less than one in general. Note that since $R$ is diagonal, (45) requires $N$ divisions and $N$ multiplications rather than a matrix inversion and matrix-vector multiply. $R$ and $P$ can be initialized to all zeros, but $D$ should be initialized to all ones.

When training is not available, the time-domain training data can be replaced by decisions on the output data, in both the LMS and RLS algorithms. For LMS, this leads to

$$D_{dlms}(k+1) = D_{dlms}(k) + \mu u^*(k)$$

$$\odot (F_N \left( Q \left[ \hat{X}(k) \right] - \hat{X}(k) \right))$$

where $Q[k]$ quantizes the data to the nearest constellation point. For RLS, this leads to replacing (44) by

$$P_{dl}(k) = \rho_{eq} \left( P_{dl}(k-1) + u^*(k) \odot (F_N Q \left[ \hat{X}(k) \right]) \right).$$

Decision direction generally requires a good initialization to roughly separate the constellation points. Alternatively, consider the constant modulus cost function

$$J_{cm} = \frac{1}{2} \sum_{i=0}^{N} \left[ \left| \hat{X}_i(k) \right|^2 - \gamma_i \right]^2$$

where $\gamma$ is a vector of real positive constants representing the desired moduli on the collection of tones. Each modulus can be independently chosen to match the source statistics on that tone, via the method of [34]. For example, for unit-power 4-QAM constellations, $\gamma = [1, 1, \ldots, 1]^T$. A gradient descent of (48) leads to a constant modulus algorithm

$$D_{cma}(k+1) = D_{cma}(k) - \mu u^*(k)$$

$$\odot (F_N \left( \left( \hat{X}_i^*(k) \odot \hat{X}(k) - \gamma_i \right) \odot \hat{X}(k) \right)).$$

In summary, we have five FEQ adaptation rules: LMS, RLS, DD-LMS, DD-RLS, and CMA. The first two require training and the last three do not. However, the author has found that a purely blind implementation of the FEQ usually leads to convergence to a poor setting, even if differential encoding is used and the final eye diagrams are “open eye.” Thus, we consider semiblind FEQs in the simulations in Section V, in which DD-LMS and CMA are initialized by a small number of iterations of LMS, and DD-RLS is initialized by a small number of iterations of RLS.

V. SIMULATIONS

In this section, we provide simulation results to demonstrate the performance of the proposed algorithms. In Section V-A, we investigate the asymptotic (nonadaptive) performance of the CSE when the two different constraints (UNC or UTC) are used. In Section V-B, we investigate the convergence speed of the proposed RLS algorithms. In Section V-C, we investigate the performance of the proposed adaptive FEQs for SCCP systems.

A. Asymptotic Performance of Channel Shortening

The main reason why the UTC has not been implemented in adaptive channel shorteners is that for the MMSE channel shortener, it leads to a higher MSEN than the UNC [17]. This suggests that, in general, the UTC has an inferior performance than the UNC, at least in terms of the optimal nonadaptive designs. In this section, we characterize the performance of the channel shorteners that minimize the MERRY cost (rather than the MSE) under the UNC versus the UTC. Specifically, we evaluate the designs in terms of the BER for a broadcast system with a fixed bit allocation across the tones. The goal is to show that the advantages of adaptive algorithm formulation under the UTC do not come with an associated asymptotic performance penalty.

Consider a multicarrier system with FFT size $N = 64$ and a CP length of $\nu = 16$. Note that although this system is comparable to the IEEE 802.11a and HIPERLAN wireless LAN standards, we are not implementing any particular standard to the letter. The channel model for this simulation is a Rayleigh-fading channel with an exponential delay profile, with approximately 32 nonzero taps (about double the CP length). There are $P = 2$ receive antennas, and each of the two CSEs has $LoS = 10$ taps. Fig. 4 shows the uncoded BER for designs that minimize the MERRY cost function using each of the constraints in a broadcast communication system that transmits 4-QAM constellations on each of the tones. The BER was measured directly, and was averaged over 100 realizations of the Rayleigh-fading channel. Differential encoding was assumed to eliminate the need for an FEQ. The performance of the UTC and the UNC are very similar, hence the use of the UTC will not adversely affect the asymptotic performance of the adaptive algorithm.

Note that the BER performance is not as low as one might expect. This is not due to a fault in the MERRY cost function, as many standard designs (such as the MMSE design [13] and the MSSNR design [18]) exhibit BER performance that is very similar to that in Fig. 4. (See, for example, [36, Fig. 12], which shows four popular channel shorteners with BER values of $10^{-2}$, even for high SNR.)
B. Convergence Speed of Channel Shortening

Again, consider a multicarrier system with FFT size $N = 64$ and a CP length of $\mu = 16$, an SNR of 30 dB, and $P = 2$ receive antennas, with each of the two channels modeled as 32-tap Rayleigh-fading channels with exponential delay profiles. Each of the CSEs has $L_{\text{CP}} = 20$ taps, and both LMS-like adaptive algorithms use a step size $\mu_0$ of $0.05$. The delay parameter $\Delta$ was blindly estimated using the method in [29], and this same delay was used for all algorithms.

Figs. 5 and 6 show the MERRY cost and BER as the algorithms adapt. Each performance metric was measured over 1000 realizations of the channel, input sequence, and noise sequence. The algorithms considered are the basic (LMS-like) MERRY algorithm with a unit-norm constraint [24], the basic (LMS-like) MERRY algorithm modified to use a unit-tap constraint, and the proposed (RLS-like) implementation of the MERRY algorithm. These are denoted MERRY-UNC, MERRY-UTC, and RLS-MERRY-UTC, respectively. The update rule for MERRY-UNC can be found in [24]; and the update rule for MERRY-UTC is similar, but the projection to restore a unit norm is removed and instead the tap held to unity is simply not updated. The update rule for RLS-MERRY-UTC is given by (19) to (23).

For reference, we also compare with the trained adaptive MMSE design of [13], and the trained nonadaptive closed-form solution of the MMSE cost function with a unit-norm constraint [17], [31] with perfect channel state information (CSI). For the adaptive MMSE design, we assume one out of every 80 symbols is available for training, which is realistic and also lets the MMSE algorithm update at the same rate as MERRY (which can only update once per block, which is once per 80 samples in this case). The MMSE step sizes on the channel shortener and target impulse response [13] were set as high as possible without causing divergence of the algorithm.

Since the MERRY cost in Fig. 5 considers algorithms with different constraints, the MERRY cost for the UNC was normalized by the power in tap $p_0$ of $\mathbf{w}$ to allow for fair comparison to the two UTC algorithms. Observe that the MERRY cost is monotonically decreasing for all algorithms, indicating that the gradient descent is operating as expected. All three algorithms appear to be approaching comparable asymptotic values of the cost, but the RLS-like implementation converges in several orders of magnitude fewer iterations.

By comparing Figs. 5 and 6, we see that after 1000 iterations, by both metrics the RLS-MERRY algorithm has the best performance of all the adaptive algorithms, but the relative performance of MERRY-UTC and MERRY-UNC is not consis-

---

1 Ideally, these algorithms would be compared in lower SNR ranges as well. However, many channel shorteners (trained, blind, adaptive, and nonadaptive alike) perform poorly at low SNR values [36], and investigating and addressing this is beyond the scope of this paper.

2 Both LMS-like algorithms became unstable when the step size was about twice this value. Since the filters are constrained, “unstable” in this context refers to large, chaotic variations in the tap values that would have led to divergence in the absence of the constraints.
tent. Specifically, MERRY-UTC has achieved a slightly better (lower) MERRY cost, yet it has a slightly worse (higher) BER. The reason for this is that optimizing the MERRY cost is a heuristic approach and will not necessarily lead to optimization of the BER. As an analogy, this is similar to using the constant modulus cost to minimize the mean squared error in a traditional adaptive equalizer. One equalizer may have a lower constant modulus cost than another equalizer but may still have a higher mean squared error, as shown in [33, Fig. 15].

Since not all channels are Rayleigh fading, we also include a simulation of a Ricean-fading channel. All parameters are as they were in Fig. 6, except the channel model. Now each channel tap is independently Ricean fading with a Rice factor of 3, and with an exponentially decaying delay profile. The resulting BER curves are shown in Fig. 7, which is qualitatively similar to Fig. 6, although the gains of the RLS implementation are more pronounced.

C. Performance of Adaptive FEQs

In this section, we study the proposed update rules for the FEQ in an SCCP system. We assume that the CSE updates using the RLS-like implementation of Section III. The parameters are an FFT size of $N = 64$ and a CP of length $\nu = 16$, with $P = 2$ channels and a CSE of length $L_w = 10$ for each channel. The channels are Rayleigh fading as before, with an SNR of 25 dB. The performance metric is the BER, as measured at the output of the final IFFT. Differential encoding of the 4-QAM source data is assumed. The reason for the use of differential encoding is that in any blind equalization scheme, there will be a phase ambiguity in the equalized data, since constant modulus and decision-directed costs are invariant to 90° rotations of the symbol estimates. Using differential encoding avoids this ambiguity, although at the cost of a BER increase.

Fig. 8 shows the BER for four FEQ adaptation algorithms: DD-LMS, CMA, RLS, and DD-RLS. In all cases, the CSE was allowed to adapt for 100 iterations before the FEQ was turned on. The DD-LMS and CMA update rules were initialized by 25 iterations of LMS, and the DD-RLS update rule was initialized by two iterations of RLS. Convergence of DD-LMS and CMA is very slow, but RLS and DD-RLS lead to much faster convergence.

VI. CONCLUSION

We have proposed a fast, RLS-like implementation of the blind, adaptive MERRY channel shortener. We also proposed several trained and blind adaptation rules for the frequency-domain equalizer in SCCP systems, which do not have finite-alphabet outputs until after the final IFFT. The performance of the proposed algorithms was demonstrated via simulations in exponentially-decaying Rayleigh-fading channels with additive white Gaussian noise. The proposed channel-shortening algorithm converges an order of magnitude faster than existing blind, adaptive channel shorteners and leads to a lower BER, making implementation practical for wireless multicarrier and SCCP systems. The proposed FEQs for SCCP systems converge very quickly as well, provided they begin adapting after the channel shortener nears convergence.

REFERENCES


Richard K. Martin (M’04) received dual B.S. degrees (summa cum laude) in physics and electrical engineering from the University of Maryland, College Park, in 1999 and the M.S. and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, in 2001 and 2004, respectively.

Since August 2004, he has been an Assistant Professor at the Air Force Institute of Technology (AFIT), Dayton, OH, where he is the Signal Processing Curriculum Chair and the Graduate Electrical Engineering Program Chair. His research interests include multicarrier equalization; single-carrier cyclic-prefix systems; blind, adaptive algorithms; reduced complexity equalizer design; and sparse adaptive filters. He has authored ten journal papers, 20 conference papers, and the book Theory and Design of Adaptive Filters Answer Book (Upper Saddle River, NJ: Prentice-Hall, 2002); and he has two patents and two more patents pending.

Dr. Martin was twice elected “Instructor of the Quarter” for the Electrical and Computer Engineering Department by the AFIT Student Association.