Efficient Channel Shortening Equalizer Computation

Richard K. Martin and C. Richard Johnson, Jr.[1]
School of Electrical & Computer Engineering
Cornell University
Ithaca, NY 14853
e-mail: {frodo,johnson}@ece.cornell.edu

Abstract — **Channel shortening is often necessary for demodulation of multicarrier signals, complexity reduction of maximum likelihood sequence estimation (MLSE), and can be used to suppress interference in multiuser detection. The Maximum Shortening SNR (MSSNR) and Minimum Mean Squared Error (MMSE) designs for channel shortening are popular in the literature due to their ease of implementation and near-optimality. This paper proposes a method to reduce the complexity of computing the matrices in the MSSNR design by a factor of 140 (for typical ADSL parameters) relative to the methods of Wu, Arslan, and Evans, for a total reduction of a factor of 4000 relative to the brute force approach, without degrading performance. A similar technique is presented for the MMSE design, reducing the matrix computation by a factor of 16 (for typical ADSL parameters).**

I. Introduction

Channel shortening first became an issue in receivers employing maximum likelihood sequence estimation (MLSE) [1]. MLSE is the optimal estimation method in terms of minimizing the error probability of a sequence. However, for an alphabet of size $\mathcal{A}$ and an effective channel length of $L_c + 1$, the complexity of MLSE grows exponentially as $\mathcal{A}^{L_c}$. One method of reducing this enormous complexity is to employ a prefilter to shorten the channel to a manageable length [2].

More recently, channel shortening has been proposed for use in multiuser detection [3]. Consider a direct-sequence code division multiple access (DS-CDMA) system that has L users, with a flat fading channel for each user. The optimum multiuser detector is the MLSE, yet complexity grows exponentially with the number of users. "Channel shortening" can be employed to suppress $L - K$ of the scalar channels and retain the other K channels, effectively reducing the number of users from L to K. Then the MLSE can be employed to recover the signals of the remaining K users [3]. In this context, "channel shortening" means reducing the number of scalar channels rather than reducing the number of channel taps.

Channel shortening has found its most widespread use in systems employing multicarrier modulation (MCM) [4]. MCM techniques like orthogonal frequency division multiplexing (OFDM) and discrete multi-tone (DMT) have been deployed in applications such as the wireless LAN standards IEEE 802.11a and HIPERLAN/2, Digital Audio Broadcast (DAB) and Digital Video Broadcast (DVB) in Europe, and asymmetric and very-high-speed digital subscriber loops (ADSL, VDSL). MCM can easily combat channel dispersion, provided the channel delay spread is not greater than the length of the cyclic prefix (CP). However, if the cyclic prefix is not long enough, inter-carrier interference (ICI) and inter-symbol interference (ISI) will be present.

A well-known technique to combat the ICI and ISI caused by an inadequate CP length is the use of a time-domain equalizer (TEQ) in the receiver front end. The TEQ is a finite impulse response (FIR) filter that shortens the channel in such a manner that the delay spread of the combined channel-equalizer impulse response is not longer than the CP length. The TEQ design problem has been extensively studied in the literature [2] – [19]. In [2], Falconer and Magee proposed a minimum-mean-square-error (MMSE) method for channel shortening, which was designed to reduce the complexity in maximum likelihood sequence estimation. More recently, Melsa, Younce, and Rohrs [7] proposed the maximum shortening SNR (MSSNR) method, which attempts to minimize the energy outside the window of interest while holding the energy inside fixed. The MSSNR method is essentially a zero-forcing version of the MMSE equalizer [12]. The MSSNR approach was generalized to the min-ISI method in [14], which allows the residual ISI to be shaped in the frequency domain. A blind, adaptive algorithm that searches for the TEQ maximizing the SSNR cost function was proposed in [17].

This paper examines the MSSNR and MMSE methods of channel shortening. The structure of each solution is exploited to dramatically reduce the complexity of computing the TEQ. Previous work on reducing the complexity of the MSSNR design was presented in [9]. This work observed that the matrices involved are almost Toeplitz, so the $(i+1, j+1)$ element can be computed efficiently from the (i, j) element. Our proposed method makes use of this, but focuses rather on determining the matrices and eigenvector for a given delay based on the matrices and eigenvector computed for the previous delay.

The remainder of this paper is organized as follows. Section II presents the system model and notation. Section III reviews the MSSNR and MMSE designs. Section IV discusses methods of reducing the computation of each design without a performance loss. Section V provides a complexity comparison to illustrate the gains of the proposed methods, and Section VI concludes the paper.

II. System Model and Notation

The multicarrier system model is shown in Fig. 1, and the notation is summarized in Table 1. The input stream is separated into blocks of length N, then the N symbols are placed in parallel bins. Each bin is viewed as a QAM signal that will be modulated by a different carrier. An efficient means of implementing the modulation in discrete time is to use an inverse fast Fourier transform (IFFT), which converts each bin (which acts as one of the frequency components) into a

[1]This work was supported in part by Applied Signal Technology (Sunnyvale, CA).

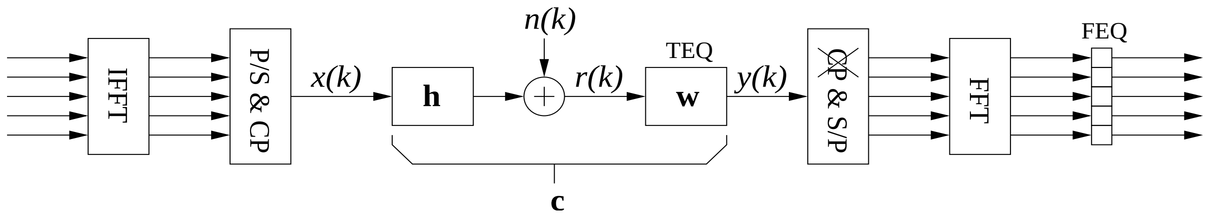

Figure 1: Traditional multicarrier system model. (I)FFT: (inverse) fast Fourier transform, P/S: parallel to serial, S/P: serial to parallel, CP: add cyclic prefix, and xCP: remove cyclic prefix.

Table 1: Channel shortening notation

Notation	**Meaning**
$x(k)$	transmitted signal (IFFT output)
$n(k)$	channel noise
$r(k)$	received signal
$y(k)$	signal after TEQ
N	FFT (block) size
ν	CP length
Δ	delay of effective channel
N_Δ	number of possible values of Δ
$\mathbf{h} = [h_0, \cdots, h_{L_h}]$	channel impulse response
$\mathbf{w} = [w_0, \cdots, w_{L_w}]$	TEQ impulse response
$\mathbf{c} = [c_0, \cdots, c_{L_c}]$	effective channel ($\mathbf{c} = \mathbf{h} \star \mathbf{w}$)
$\mathbf{b} = [b_0, \cdots, b_\nu]$	target impulse response
$\tilde{L}_h = L_h + 1$	channel length
$\tilde{L}_w = L_w + 1$	TEQ length
$\tilde{L}_c = L_c + 1$	length of the effective channel
$\mathbf{H}$	$\tilde{L}_c \times \tilde{L}_w$ channel convolution matrix
$\mathbf{H}_{win}$	middle $\nu + 1$ rows of $\mathbf{H}$
$\mathbf{H}_{wall}$	$\mathbf{H}$ with middle $\nu + 1$ rows removed
$\mathbf{A} = \mathbf{H}_{wall}^T \mathbf{H}_{wall}$	$\tilde{L}_w \times \tilde{L}_w$ real matrix
$\mathbf{B} = \mathbf{H}_{win}^T \mathbf{H}_{win}$	$\tilde{L}_w \times \tilde{L}_w$ real matrix
$\mathbf{I}_N$	$N \times N$ identity matrix
$[\mathbf{A}]_{(i,j)}$	Element i, j of matrix $\mathbf{A}$
$\mathbf{A}^*$, $\mathbf{A}^T$, $\mathbf{A}^H$	conjugate, transpose, and Hermitian

time-domain signal. After transmission, the receiver can use an FFT to recover the data within a bit error rate tolerance, provided that equalization has been performed properly.

In order for the subchannels to not interfere with one another, the convolution of the signal and the channel must be a circular convolution. It is actually a linear convolution, so it is made to appear circular by adding a cyclic prefix to the start of each data block. The cyclic prefix is obtained by prepending the last ν samples of each block to the beginning of the block. If the CP is at least as long as the channel, then the output of each subchannel is equal to the input times a scalar complex gain factor. The signals in the bins can then be equalized by a bank of complex gains, referred to as a frequency domain equalizer (FEQ) [20].

Transmitting the CP wastes time that could be used to transmit data, reducing the throughput by a factor of $\frac{N}{N+\nu}$. Thus, the CP is usually set to a reasonably small value, and a TEQ is used to shorten the channel to this length. In ADSL and VDSL, the CP length is $\frac{1}{16}$ of the block (symbol) length. As discussed in Section I, TEQ design methods have been well explored [2] – [19].

III. Review of the MSSNR and MMSE designs

This section reviews the MSSNR and MMSE designs for channel shortening.

A. The MSSNR solution

Consider the maximum shortening SNR (MSSNR) TEQ design [7], which attempts to maximize the ratio of the energy in a window of the effective channel over the energy in the remainder of the effective channel. Following [7], we define

$$\mathbf{H}_{win} = \begin{bmatrix} h(\Delta) & h(\Delta-1) & \cdots & h(\Delta-\tilde{L}_w+1) \\ \vdots & & \ddots & \vdots \\ h(\Delta+\nu) & h(\Delta+\nu-1) & \cdots & h(\Delta+\nu-\tilde{L}_w+1) \end{bmatrix} \quad (1)$$

and

$$\mathbf{H}_{wall} = \begin{bmatrix} h(0) & 0 & \cdots & 0 \\ \vdots & \ddots & & \\ h(\Delta-1) & h(\Delta-2) & \cdots & h(\Delta-\tilde{L}_w) \\ h(\Delta+\nu+1) & h(\Delta+\nu) & \cdots & h(\Delta+\nu-\tilde{L}_w+2) \\ \vdots & \ddots & & \\ 0 & \cdots & 0 & h(L_h) \end{bmatrix} \quad (2)$$

Thus, $\mathbf{c}_{win} = \mathbf{H}_{win}\mathbf{w}$ yields a length $\nu + 1$ window of the effective channel, and $\mathbf{c}_{wall} = \mathbf{H}_{wall}\mathbf{w}$ yields the remainder of the effective channel. The MSSNR design problem can be stated as "minimize $\|\mathbf{c}_{wall}\|$ subject to the constraint $\|\mathbf{c}_{win}\| = 1$," as in [7]. This reduces to

$$\min_{\mathbf{w}} \left(\mathbf{w}^T \mathbf{A} \mathbf{w} \right) \quad \text{subject to} \quad \mathbf{w}^T \mathbf{B} \mathbf{w} = 1, \quad (3)$$

where

$$\mathbf{A} = \mathbf{H}_{wall}^T \mathbf{H}_{wall}, \quad \mathbf{B} = \mathbf{H}_{win}^T \mathbf{H}_{win}. \quad (4)$$

$\mathbf{A}$ and $\mathbf{B}$ are real, symmetric $\tilde{L}_w \times \tilde{L}_w$ matrices. However, $\mathbf{A}$ is invertible, but $\mathbf{B}$ may not be. An alternative formulation that addresses this is to "maximize $\|\mathbf{c}_{win}\|$ subject to the constraint $\|\mathbf{c}_{wall}\| = 1$," which works well even when $\mathbf{B}$ is not invertible [8]. The alternative formulation reduces to

$$\max_{\mathbf{w}} \left(\mathbf{w}^T \mathbf{B} \mathbf{w} \right) \quad \text{subject to} \quad \mathbf{w}^T \mathbf{A} \mathbf{w} = 1, \quad (5)$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined in (4). Solving (3) leads to a TEQ that satisfies the generalized eigenvector problem,

$$\mathbf{A}\mathbf{w} = \tilde{\lambda}\mathbf{B}\mathbf{w}, \quad (6)$$

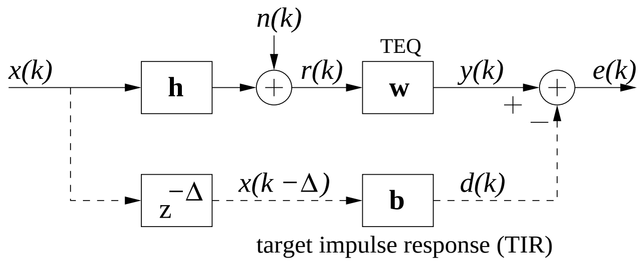

Figure 2: MMSE system model: $\mathbf{h}$, $\mathbf{w}$, and $\mathbf{b}$ are the impulse responses of the channel, TEQ, and target, respectively. Here, Δ represents transmission delay. The dashed lines indicate a virtual path, which is used only for analysis.

and the alternative formulation in (5) leads to a related generalized eigenvector problem,

$$\mathbf{B}\mathbf{w} = \lambda \mathbf{A}\mathbf{w}. \tag{7}$$

The solution for $\mathbf{w}$ will be the generalized eigenvector corresponding to the smallest (largest) generalized eigenvalue $\tilde{\lambda}$ (λ). Section IV shows how to obtain most of $\mathbf{B}(\Delta+1)$ from $\mathbf{B}(\Delta)$, how to obtain $\mathbf{A}(\Delta)$ from $\mathbf{B}(\Delta)$, and how to initialize the eigensolver for $\mathbf{w}(\Delta+1)$ based on the solution for $\mathbf{w}(\Delta)$.

B. The MMSE solution

The system model for the minimum mean-squared error (MMSE) solution [2] is shown in Fig. 2. It creates a virtual target impulse response (TIR) $\mathbf{b}$ of length $\nu+1$ such that the MSE, which is measured between the output of the effective channel and the output of the TIR, is minimized. In the absence of noise, if the input signal is white, then the optimal MMSE and MSSNR solutions are identical [12].

The MMSE design uses a target impulse response (TIR) $\mathbf{b}$ that must satisfy [2], [5], [11]

$$\mathbf{R}_{rx}\mathbf{b} = \mathbf{R}_r \mathbf{w}, \tag{8}$$

where

$$\mathbf{R}_{rx} = \mathrm{E}\left[\begin{bmatrix} r(k) \\ \vdots \\ r(k-L_w)\end{bmatrix}\begin{bmatrix} x(k-\Delta) & \cdots & x(k-\Delta-\nu)\end{bmatrix}\right] \tag{9}$$

is the channel input-output cross-correlation matrix and

$$\mathbf{R}_{r} = \mathrm{E}\left[\begin{bmatrix} r(k) \\ \vdots \\ r(k-L_w)\end{bmatrix}\begin{bmatrix} r(k) & \cdots & r(k-L_w)\end{bmatrix}\right] \tag{10}$$

is the channel output autocorrelation matrix. Typically, $\mathbf{b}$ is computed first, and then (8) is used to determine $\mathbf{w}$. The goal is that $\mathbf{h} \star \mathbf{w}$ approximates a delayed version of $\mathbf{b}$. The target impulse response is the eigenvector corresponding to the minimum eigenvalue of [2], [5], [11]

$$\mathbf{R}(\Delta) = \mathbf{R}_x - \mathbf{R}_{xr}\mathbf{R}_r^{-1}\mathbf{R}_{rx}.$$

Section IV addresses how to determine most of $\mathbf{R}(\Delta+1)$ from $\mathbf{R}(\Delta)$, and how to use the solution for $\mathbf{b}(\Delta)$ to initialize the eigensolver for $\mathbf{b}(\Delta+1)$.

IV. Efficient computation

There is a tremendous amount of redundancy involved in the brute force calculation of the MSSNR design. Wu, Arslan, and Evans [9] proposed a method to exploit some of the redundancy to greatly decrease the cost of computing $\mathbf{A}$ and $\mathbf{B}$. This approach was based on exploiting the fact that the matrices are nearly Toeplitz, so the $(i+1, j+1)$ element can be obtained from the (i, j) element. However, there are also two other forms of redundancy: redundancy over delay, and redundancy between $\mathbf{A}$ and $\mathbf{B}$. This section discusses methods to exploit this, reusing even more of the computations to dramatically decrease the required complexity. Specifically, for a given delay Δ,

- $\mathbf{A}(\Delta)$ can be computed from $\mathbf{B}(\Delta)$ almost for free.
- $\mathbf{B}(\Delta+1)$ can be computed from $\mathbf{B}(\Delta)$ almost for free.
- A shifted version of the optimal MSSNR TEQ $\mathbf{w}(\Delta)$ can be used to initialize the generalized eigenvector solution for $\mathbf{w}(\Delta+1)$ to decrease the number of iterations needed for the eigenvector computation.
- $\mathbf{R}(\Delta+1)$ can be computed from $\mathbf{R}(\Delta)$ almost for free.
- A shifted version of the optimal MMSE TIR $\mathbf{b}(\Delta)$ can be used to initialize the generalized eigenvector solution for $\mathbf{b}(\Delta+1)$ to decrease the number of iterations needed for the eigenvector computation.

We now discuss each of these points in turn.

A. Computing $\mathbf{A}(\Delta)$ from $\mathbf{B}(\Delta)$

Let $\mathbf{C} = \mathbf{H}^T\mathbf{H}$, and recall that $\mathbf{A} = \mathbf{H}_{wall}^T\mathbf{H}_{wall}$ and $\mathbf{B} = \mathbf{H}_{win}^T\mathbf{H}_{win}$. Note that

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \hline \mathbf{H}_{win} \\ \hline \mathbf{H}_2 \end{bmatrix}, \quad \mathbf{H}_{wall} = \begin{bmatrix} \mathbf{H}_1 \\ \hline \mathbf{H}_2 \end{bmatrix}. \tag{11}$$

Thus,

$$\begin{aligned} \mathbf{C} &= \mathbf{H}_1^T\mathbf{H}_1 + \mathbf{H}_{win}^T\mathbf{H}_{win} + \mathbf{H}_2^T\mathbf{H}_2 && (12) \\ &= \underbrace{\left(\mathbf{H}_1^T\mathbf{H}_1 + \mathbf{H}_2^T\mathbf{H}_2\right)}_{\mathbf{A}} + \underbrace{\left(\mathbf{H}_{win}^T\mathbf{H}_{win}\right)}_{\mathbf{B}}. && (13) \end{aligned}$$

To emphasize the dependence on the delay Δ, we write

$$\boxed{\mathbf{C} = \mathbf{A}(\Delta) + \mathbf{B}(\Delta)} \tag{14}$$

Since $\mathbf{H}$ is a tall convolution matrix, $\mathbf{C}$ is symmetric and Toeplitz. Thus, it is fully determined by its first row or column:

$$\mathbf{C}_{(0:L_w,0)} = \mathbf{H}^T \left[\mathbf{h}^T, \mathbf{0}_{(1\times L_w)}\right]^T = \left(\mathbf{H}_{(0:L_h,0:L_w)}\right)^T \mathbf{h}. \tag{15}$$

$\mathbf{C}$ can be computed using $\tilde{L}_h\tilde{L}_w$ multiply adds, and its first column can be stored using $\tilde{L}_w$ memory words. Since $\mathbf{C}$ is independent of Δ, we only need to compute it once. Then each time Δ is incremented and the new $\mathbf{B}(\Delta)$ is computed, $\mathbf{A}(\Delta)$ can be computed from $\mathbf{A}(\Delta) = \mathbf{C} - \mathbf{B}(\Delta)$ using only $\tilde{L}_w^2$ additions and no multiplications. In constrast, the "brute force" method requires $\tilde{L}_w^2(L_h-\nu)$ multiply-adds to compute $\mathbf{A}$ for each delay, and the method of [9] requires about $\tilde{L}_w(L_w+L_h-\nu)$ multiply-adds per delay.

B. Computing $\mathbf{B}(\Delta+1)$ from $\mathbf{B}(\Delta)$

Recall that $\mathbf{B}(\Delta) = \mathbf{H}_{win}^T(\Delta)\mathbf{H}_{win}(\Delta)$, where

$$\mathbf{H}_{win}(\Delta) = \begin{bmatrix} h(\Delta) & h(\Delta-1) & \cdots & h(\Delta-\tilde{L}_w+1) \\ \vdots & & \ddots & \vdots \\ h(\Delta+\nu) & h(\Delta+\nu-1) & \cdots & h(\Delta+\nu-\tilde{L}_w+1) \end{bmatrix} \quad (16)$$

The key observation is that

$$[\mathbf{H}_{win}(\Delta+1)]_{(0:\nu,1:L_w)} = [\mathbf{H}_{win}(\Delta)]_{(0:\nu,0:L_w-1)}. \quad (17)$$

This means that

$$\boxed{[\mathbf{B}(\Delta+1)]_{(1:L_w,1:L_w)} = [\mathbf{B}(\Delta)]_{(0:L_w-1,0:L_w-1)}} \quad (18)$$

so most of $\mathbf{B}(\Delta+1)$ can be obtained without requiring any computations. Now partition $\mathbf{B}(\Delta+1)$ as

$$\mathbf{B}(\Delta+1) = \left[\begin{array}{c|c} \alpha & \mathbf{g}^T \\ \hline \mathbf{g} & \hat{\mathbf{B}} \end{array}\right], \quad (19)$$

where $\hat{\mathbf{B}}$ is obtained from (18). Since $\mathbf{B}(\Delta+1)$ is almost Toeplitz, α and all of the elements of $\mathbf{g}$ save the last can be efficiently determined from the first column of $\hat{\mathbf{B}}$ [9]. Computing each of these L_w elements requires two multiply-adds. Finally, to compute the last element of $\mathbf{g}$,

$$\mathbf{g}_{(\nu-1)} = \left([\mathbf{H}_{win}]_{(0:\nu,L_w)}\right)^T [\mathbf{H}_{win}]_{(0:\nu,0)}, \quad (20)$$

requiring $\nu+1$ multiply-adds.

Thus, a fast MSSNR design algorithm is as follows:

1. Compute $\mathbf{B} = \mathbf{H}_{win}^T(\Delta_{min})\,\mathbf{H}_{win}(\Delta_{min})$.
2. $\mathbf{C}_{(0:L_w,0)} = \left(\mathbf{H}_{(0:L_h,0:L_w)}\right)^T \mathbf{h}$.
3. Fill in the rest of the symmetric, Toeplitz matrix $\mathbf{C}$.
4. $\mathbf{A} = \mathbf{C} - \mathbf{B}$.
5. Solve $\mathbf{Bw} = \lambda\mathbf{Aw}$ for the generalized eigenvector corresponding to the largest eigenvalue, as in [7].
6. For $\Delta = \Delta_{min}+1 : \Delta_{max}$, do the following:
   (a) $[\mathbf{B}]_{(1:L_w,1:L_w)} = [\mathbf{B}]_{(0:L_w-1,0:L_w-1)}$
   (b) $[\mathbf{B}]_{(0:L_w-1,0)} = [\mathbf{B}]_{(1:L_w,1)} + h(\Delta+\nu)\cdot h(\Delta+\nu-[0:L_w-1]) - h(\Delta-1)\cdot h(\Delta-1-[0:L_w-1])$
   (c) $[\mathbf{B}]_{(L_w,0)} = \left([\mathbf{H}_{win}]_{(0:\nu,L_w)}\right)^T [\mathbf{H}_{win}]_{(0:\nu,0)}$
   (d) $[\mathbf{B}]_{(0,1:L_w)} = [\mathbf{B}]^T_{(1:L_w,0)}$
   (e) $\mathbf{A} = \mathbf{C} - \mathbf{B}$.
   (f) Solve $\mathbf{Bw} = \lambda\mathbf{Aw}$ for the generalized eigenvector corresponding to the largest eigenvalue.
   (g) If this delay produces a larger shortening SNR λ than the previous delay, save $\mathbf{w}$.

Note that in step (b), the indices may become negative, in which case the corresponding elements are zero. It should be stressed that the gains will not be as apparent in an environment such as Matlab, since the brute force method is matrix based, and the proposed approach is an element-by-element approach. Matlab is optimized for the former, but embedded DSPs may not be.

C. Computing $\mathbf{R}(\Delta+1)$ from $\mathbf{R}(\Delta)$

Recall that for the MMSE design, we must compute

$$\mathbf{R}(\Delta) = \mathbf{R}_x - \mathbf{R}_{xr}\mathbf{R}_r^{-1}\mathbf{R}_{rx},$$

where

$$\mathbf{R}_x = \mathrm{E}\left[\begin{bmatrix} x(k-\Delta) \\ \vdots \\ x(k-\Delta-\nu) \end{bmatrix} \begin{bmatrix} x(k-\Delta) & \cdots & x(k-\Delta-\nu) \end{bmatrix}\right] \quad (21)$$

and

$$\mathbf{R}_{rx} = \mathrm{E}\left[\begin{bmatrix} r(k) \\ \vdots \\ r(k-L_w) \end{bmatrix} \begin{bmatrix} x(k-\Delta) & \cdots & x(k-\Delta-\nu) \end{bmatrix}\right] \quad (22)$$

Note that $\mathbf{R}_x$ does not depend on Δ, and that it is Toeplitz. Thus,

$$\begin{aligned} [\mathbf{R}_x(\Delta+1)]_{(0:\nu-1,0:\nu-1)} &= [\mathbf{R}_x(\Delta)]_{(0:\nu-1,0:\nu-1)} \\ &= [\mathbf{R}_x(\Delta)]_{(1:\nu,1:\nu)}. \end{aligned} \quad (23)$$

Let $\mathbf{P}(\Delta) = \mathbf{R}_{xr}\mathbf{R}_r^{-1}\mathbf{R}_{rx}$. Observing that

$$[\mathbf{R}_{rx}(\Delta+1)]_{(0:L_w,0:\nu-1)} = [\mathbf{R}_{rx}(\Delta)]_{(0:L_w,1:\nu)}, \quad (24)$$

we see that

$$[\mathbf{P}(\Delta+1)]_{(0:\nu-1,0:\nu-1)} = [\mathbf{P}(\Delta)]_{(1:\nu,1:\nu)}. \quad (25)$$

Combining (23) and (25),

$$\boxed{[\mathbf{R}(\Delta+1)]_{(0:\nu-1,0:\nu-1)} = [\mathbf{R}(\Delta)]_{(1:\nu,1:\nu)}} \quad (26)$$

The matrix $\mathbf{R}_r$ is symmetric and Toeplitz. However, the inverse of a Toeplitz matrix is, in general, not Toeplitz [21]. This means that $\mathbf{R}(\Delta)$ has no further structure that can be easily exploited, so the first row and column of $\mathbf{R}(\Delta+1)$ cannot be obtained from the rest of $\mathbf{R}(\Delta+1)$ using the tricks in [9]. Even so, (26) allows us to obtain most of the elements of each $\mathbf{R}(\Delta)$ for free, so only $\nu+1$ elements must be computed rather than $(\nu+1)(\nu+2)/2$ elements. In ADSL, $\nu=32$; in VDSL, ν can range up to 512; and in DVB, ν can range up to 2048. Thus, the proposed method reduces the complexity of calculating $\mathbf{R}(\Delta)$ by factors of 17, 257, and 1025 (respectively) for these standards.

D. Intelligent eigensolver initialization

Let $\mathbf{w}(\Delta)$ be the MSSNR solution for a given delay. If we were to increase the allowable filter length by 1, then it follows that

$$\hat{\mathbf{w}}(\Delta+1) = z^{-1}\mathbf{w}(\Delta) = \left[0, \mathbf{w}^T(\Delta)\right]^T \quad (27)$$

should be a near-optimum solution, since it produces the same value of the shortening SNR as for the previous delay. Experience suggests that the TEQ coefficients are small near the

Table 2: Computational complexity of various MSSNR implementations. MACs are real multiply-and-accumulates and adds are real additions (or subtractions). For the example, $\tilde{L}_h = 512$, $\tilde{L}_w = 32$, $\tilde{L}_c = 543$, $\nu = 32$, and $N_\Delta = 511$.

	brute force	**Wu, et al. [9]**
step	MACs	MACs
$\mathbf{C}$	0	0
$\mathbf{B}(\Delta_{min})$	$\tilde{L}_w^2(\nu+1)$	$\tilde{L}_w(L_w+\nu)$
$\mathbf{A}(\Delta_{min})$	$\tilde{L}_w^2(L_h-\nu)$	$\tilde{L}_w(L_c-\nu)$
Each $\mathbf{B}(\Delta)$	$\tilde{L}_w^2(\nu+1)$	$\tilde{L}_w(L_w+\nu)$
Each $\mathbf{A}(\Delta)$	$\tilde{L}_w^2(L_h-\nu)$	$\tilde{L}_w(L_c-\nu)$
Total:	$\tilde{L}_w^2\tilde{L}_h N_\Delta$	$\tilde{L}_w(L_w+L_c)N_\Delta$
Example:	267,911,168	9,369,696

	proposed	
step	MACs	adds
$\mathbf{C}$	$\tilde{L}_h\tilde{L}_w$	0
$\mathbf{B}(\Delta_{min})$	$\tilde{L}_w(L_w+\nu)$	0
$\mathbf{A}(\Delta_{min})$	0	$\tilde{L}_w^2$
Each $\mathbf{B}(\Delta)$	$2L_w+\nu+1$	0
Each $\mathbf{A}(\Delta)$	0	$\tilde{L}_w^2$
Total:	$\left(2\tilde{L}_w+\nu\right)(N_\Delta-1)$ $+\tilde{L}_h\tilde{L}_w$	$\tilde{L}_w^2 N_\Delta$
Example:	66,850	523,264

edges, so the last tap can be removed without drastically affecting the performance. Therefore,

$$\hat{\mathbf{w}}(\Delta+1) = \left[0, \left[\mathbf{w}^T(\Delta)\right]_{(0:L_w-1)}\right]^T \tag{28}$$

is a fairly good solution for the delay $\Delta+1$, so this should be the initialization for the generalized eigenvector solver for the next delay. Similarly, for the MMSE TIR,

$$\hat{\mathbf{b}}(\Delta+1) = \left[\left[\mathbf{b}^T(\Delta)\right]_{(1:\nu)}, 0\right]^T \tag{29}$$

should be the initialization for the eigenvector solver for the next delay.

V. Complexity comparison

Table 2 shows the (approximate) number of computations for each step of the MSSNR method, using the "brute force" approach, the method in [9], and the proposed approach. Note that N_Δ refers to the number of values of the delay that are possible (usually equal to the length of the effective channel minus the CP length). For a typical downstream ADSL system, the parameters are $\tilde{L}_w = L_w+1 = 32$, $\tilde{L}_h = L_h+1 = 512$, $L_c = L_w + L_h = 542$, $\nu = 32$, and $N_\Delta = \tilde{L}_c - \nu = 511$. The "example" lines in Table 2 show the required complexity for computing all of the $\mathbf{A}$'s and $\mathbf{B}'s$ for these parameters using each approach. Observe that [9] beats the brute force method by a factor of 29, the proposed method beats [9] by a factor of 140, and the proposed method beats the brute force method by a factor of 4008.

Table 3 shows the (approximate) computational requirements of the "brute force" approach and the proposed approach for computing the MMSE matrices $\mathbf{R}(\Delta), \Delta \in \{\Delta_{min}, \cdots, \Delta_{max}\}$. The "example" line shows the required complexity for computing the $\mathbf{R}(\Delta)$ matrices using each method for the same parameter values as the example in Table 2. The proposed method yields a decrease in complexity by a factor of approximately $\frac{\nu+2}{2}$, which in this case is a factor of 16.

Table 3: Computational complexity of various MMSE implementations. MACs are real multiply-and-accumulates. For the example, $\tilde{L}_w = \nu = 32$, and $N_\Delta = 511$.

	brute force	**proposed**
step	MACs	MACs
$\mathbf{R}(\Delta_{min})$	$\frac{(\nu+1)(\nu+2)\tilde{L}_w^2}{2}$	$\frac{(\nu+1)(\nu+2)\tilde{L}_w^2}{2}$
Each $\mathbf{R}(\Delta)$	$\frac{(\nu+1)(\nu+2)\tilde{L}_w^2}{2}$	$(\nu+1)\tilde{L}_w^2$
Total:	$\frac{N_\Delta(\nu+1)(\nu+2)\tilde{L}_w^2}{2}$	$\tilde{L}_w^2(\nu+1)\left((N_\Delta-1)+\frac{1}{2}(\nu+2)\right)$
Example:	293,551,104	17,808,384

VI. Conclusions

The computational complexity of two popular channel shortening algorithms, the MSSNR and MMSE methods, has been addressed. A method was proposed which reduces the complexity of computing the $\mathbf{A}$ and $\mathbf{B}$ matrices in the MSSNR design by a factor of 140 (for typical ADSL parameters) relative to the methods of Wu, Arslan, and Evans [9], for a total reduction of a factor of 4000 relative to the brute force approach, without degrading performance. A similar technique was proposed to reduce the complexity of computing the $\mathbf{R}(\Delta)$ matrix used in the MMSE design by a factor of 16 (for typical ADSL parameters).

Acknowledgments

The authors wish to thank Dr. Brian Evans and Mr. Ming Ding of The University of Texas at Austin for their comments.

References

[1] G. D. Forney, "Maximum-likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. on Info. Theory*, vol. 18, pp. 363–378, May 1972.

[2] D. D. Falconer and F. R. Magee, "Adaptive Channel Memory Truncation for Maximum Likelihood Sequence Estimation," *Bell Sys. Tech. Journal*, pp. 1541–1562, Nov. 1973.

[3] I. Medvedev and V. Tarokh, "A Channel-Shortening Multiuser Detector for DS-CDMA Systems," in *Proceeding of the 53rd Vehicular Technology Conference*, Rhodes, Greece, May 2001, vol. 3, pp. 1834–1838.

[4] A. N. Akansu, "Orthogonal Transmultiplexers in Communication – A Review," *IEEE Transactions on Signal processing*, vol. 46, no. 4, pp. 979–995, Apr. 1998.

[5] N. Al-Dhahir and J. M. Cioffi, "Efficiently Computed Reduced-Parameter Input-Aided MMSE Equalizers for ML Detection: A Unified Approach," *IEEE Trans. on Info. Theory*, vol. 42, no. 3, pp. 903–915, May 1996.

[6] N. Al-Dhahir and J. M. Cioffi, "Optimum Finite-Length Equalization for Multicarrier Transceivers," *IEEE Trans. on Comm.*, vol. 44, no. 1, pp. 56–64, Jan. 1996.

[7] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse Response Shortening for Discrete Multitone Transceivers," *IEEE Trans. on Comm.*, vol. 44, pp. 1662–1672, Dec. 1996.

[8] C. Yin and G. Yue, "Optimal Impulse Response Shortening for Discrete Multitone Transceivers," *Electronics Letters*, vol. 34, pp. 35–36, Jan. 1998.

[9] J. Wu, G. Arslan, and B. L. Evans, "Efficient Matrix Multiplication Methods to Implement a Near-Optimum Channel Shortening Method for Discrete Multitone Transceivers," in *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2000, vol. 1, pp. 152–157.

[10] K. Van Acker, G. Leus, M. Moonen, O. van de Wiel, and T. Pollet, "Per Tone Equalization for DMT-Based Systems," *IEEE Trans. on Comm.*, vol. 49, no. 1, pp. 109–119, Jan. 2001.

[11] B. Farhang-Boroujeny and M. Ding, "Design Methods for Time-Domain Equalizers in DMT Transceivers," *IEEE Trans. on Comm.*, vol. 49, no. 3, pp. 554–562, Mar. 2001.

[12] D. Daly, C. Heneghan, and A. D. Fagan, "A Minimum Mean-Squared Error Interpretation of Residual ISI Channel Shortening for Discrete Multitone Transceivers," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 2001, vol. 4, pp. 2065–2068.

[13] R. Schur and J. Speidel, "An Efficient Equalization Method to Minimize Delay Spread in OFDM/DMT Systems," in *IEEE International Conference on Communications*, June 2001.

[14] G. Arslan, B. L. Evans, and S. Kiaei, "Equalization for Discrete Multitone Receivers To Maximize Bit Rate," *IEEE Trans. on Signal Processing*, vol. 49, no. 12, pp. 3123–3135, Dec. 2001.

[15] A. Tkacenko and P. P. Vaidyanathan, "Noise Optimized Eigenfilter Design of Time-domain Equalizers for DMT Systems," in *Proc. IEEE Int. Conf. on Comm.*, New York, NY, Apr.–May 2002, vol. 1.

[16] J. Balakrishnan, R. K. Martin, and C. R. Johnson, Jr., "Blind, Adaptive Channel Shortening by Sum-squared Auto-correlation Minimization (SAM)," in *Proc. 36th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2002.

[17] R. K. Martin, J. Balakrishnan, W. A. Sethares, and C. R. Johnson, Jr., "A Blind, Adaptive TEQ for Multicarrier Systems," *IEEE Signal Processing Letters*, vol. 9, no. 11, pp. 341–343, Nov. 2002.

[18] M. Milosevic, L. F. C. Pessoa, B. L. Evans, and R. Baldick, "Optimal Time Domain Equalization Design for Maximizing Data Rate of Discrete Multi-tone Systems," *IEEE Trans. on Signal Processing*, under revision.

[19] K. Vanbleu, G. Ysebaert, G. Cuypers, M. Moonen, and K. Van Acker, "Bitrate Maximizing Time-Domain Equalizer Design for DMT-based Systems," Submitted to *IEEE Trans. on Comm.*

[20] J. M. Cioffi, "A Multicarrier Primer," http://www-isl.stanford.edu/people/cioffi/pdf/multicarrier.pdf.

[21] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, Inc., New York, NY, 1996.