

Implementation Complexity and Communication Performance Tradeoffs in Discrete Multitone Modulation Equalizers

Richard K. Martin

Air Force Institute of Technology, AFIT/ENG
Dept. of Electrical & Computer Engineering
Wright-Patt AFB, OH 45433-7765, USA
richard.martin@afit.edu

Koen Vanbleu

Broadcom UK Ltd.
2800 Mechelen, Belgium
koen.vanbleu@broadcom.com

Ming Ding

Bandspeed, Inc.
4301 Westbank Drive, Bldg. B, Suite 100
Austin, TX 78746, USA
mding@bandspeed.com

Geert Ysebaert

Alcatel Bell
Francis Wellesplein, 1
2018 Antwerpen, Belgium
geert.ysebaert@alcatel.be

Milos Milosevic

Schlumberger
Sugar Land, TX 77478, USA
mmilosevic@austin.rr.com

Brian L. Evans

The University of Texas at Austin
Dept. of Electrical & Computer Engineering
Engineering Science Building, Room 433B
1 University Station C0803
Austin, TX 78712-1084, USA
bevans@ece.utexas.edu

Marc Moonen

ESAT-SISTA K.U. Leuven
Kasteelpark Arenberg 10, Office: 01.69
B-3001 Leuven-Heverlee, Belgium
Marc.Moonen@esat.kuleuven.ac.be

C. Richard Johnson, Jr.

Cornell University, 390 Rhodes Hall
School of Electrical & Computer Engineering
Ithaca, NY 14853, USA
johnson@ece.cornell.edu

© 2006 IEEE. To appear in *IEEE Transactions on Signal Processing*, August 2006. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact:

IEEE Intellectual Property Rights Office
IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331, USA.
Telephone: + Intl. 732-562-3966.
Fax: + Intl. 732-981-8062.

Implementation Complexity and Communication Performance Tradeoffs in Discrete Multitone Modulation Equalizers

Richard K. Martin, *Member, IEEE*, Koen Vanbleu, *Member, IEEE*,
 Ming Ding, *Member, IEEE*, Geert Ysebaert, *Member, IEEE*,
 Milos Milosevic, *Member, IEEE*, Brian L. Evans, *Senior Member, IEEE*,
 Marc Moonen, *Member, IEEE*, C. Richard Johnson, Jr., *Fellow, IEEE*

Abstract—Several high-speed communication standards modulate encoded data on multiple carrier frequencies using the inverse Fourier transform (FFT). The real part of the quantized inverse FFT samples form a symbol. The symbol is periodically extended by prepending a copy of its last few samples, a.k.a. a cyclic prefix. When the cyclic prefix is longer than the channel order, amplitude and phase distortion can be equalized entirely in the frequency domain. In the receiver, prior to the FFT, a time-domain equalizer, in the form of a finite impulse response filter, shortens the effective channel impulse response. Alternately, a bank of equalizers tuned to each carrier frequency can be used. In earlier work, we unified optimal multicarrier equalizer design algorithms as a product of generalized Rayleigh quotients. In this paper, we convert the unified theoretical framework into a framework for fast design algorithms. The relevant literature is reviewed and classified according to this framework. We analyze the achieved bit rate vs. implementation complexity (in terms of multiply-and-accumulate operations) tradeoffs in the original and fast design algorithms. The comparison includes multiple implementations of each of 16 different equalizer structures and design algorithms using synthetic and measured discrete multitone modulated data.

Index Terms: Multicarrier Equalization, Channel Shortening, Real-time Implementation.

EDICS Designation: 2-IMPL – algorithm implementation in hardware and software

This work was supported in part by NSF grant CCR-0310023; Applied Signal Technology (Sunnyvale, CA); Texas Instruments (Dallas, TX); the Belgian State, Prime Minister's Office – Federal Office for Scientific, Technical and Cultural Affairs – Interuniversity Poles of Attraction Programme (2002–2007) – IUAP P5/22 and P5/11; the Concerted Research Action GOA-MEFISTO-666 of the Flemish Government; Research Project FWO nr. G.0196.02; and The State of Texas Advanced Technology Program under project 003658-0614-2001. The views expressed in this paper are those of the authors, and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government. This document has been approved for public release; distribution unlimited. The scientific responsibility is assumed by the authors.

R. K. Martin is with the Dept. of Elec. and Comp. Eng., The Air Force Inst. of Technology, WPABF, OH 45433 (richard.martin@afit.edu). M. Ding is with Bandspeed, Inc., Austin, TX (mding@bandspeed.com). K. Vanbleu is with Broadcom UK Ltd., Mechelen, Belgium (koen.vanbleu@telenet.be). M. Milosevic is with Schlumberger, Sugar Land, TX 77478 USA (mmilosevic@austin.rr.com). G. Ysebaert is with Alcatel Bell, Antwerpen, Belgium. Brian Evans is with the Dept. of Elec. and Comp. Eng. at The University of Texas at Austin (bevans@ece.utexas.edu). M. Moonen is with the Katholieke Universiteit Leuven – ESAT-SCD/SISTA, 3001 Leuven–Heverlee, Belgium (moonen@esat.kuleuven.ac.be). C. R. Johnson, Jr., is with the School of Elec. and Comp. Eng., Cornell University, Ithaca, NY 14853 (johnson@ece.cornell.edu).

I. INTRODUCTION

Multicarrier (MC) modulation is currently enjoying a boom in popularity, largely due to the fact that it allows an efficient receiver implementation that achieves high throughput [1]. Discrete multitone (DMT) has been implemented in wireline MC applications such as various digital subscriber line (DSL) standards [2] and in power line communications standards. Orthogonal frequency division multiplexing (OFDM) has been adopted in wireless MC standards such as in IEEE 802.11a [3] and HIPERLAN2 [4] local area networks, digital video and audio broadcast (DVB/DAB) [5], [6], and satellite radio [7]. We focus on the wireline case, but the main difference is that wireless implementations assume a complex-valued baseband model, whereas wireline implementations use a real-valued baseband format.

One of the main advantages of MC modulation (relative to single carrier modulation) is the ease with which equalization can be performed. If the channel delay spread is shorter than the guard interval between the transmitted blocks, then the frequency-selective channel appears as a bank of adjacent flat fading channels, and equalization can be efficiently performed in the frequency domain by a bank of scalars. If the channel delay spread is longer than this guard interval, then a prefilter is needed at the receiver to shorten the effective channel to the appropriate length. This prefilter is called a time-domain equalizer (TEQ). A review of optimal TEQ designs is given in [1]. An alternative to the TEQ structure is to use a bank of filters or linear combiners, one per tone, to remove the intersymbol and intercarrier interference (ISI, ICI) caused by a long channel. The filters can be placed in the time or frequency domain, leading to the TEQ filter bank (TEQFB) [8] or the Per-Tone Equalizer (PTEQ) [9], respectively.

Many equalizer designs are computationally intensive, requiring multiple matrix inversions, eigendecompositions, and Cholesky decompositions. However, the matrices involved often have such a structure that many computations can be reused. Moreover, it is sometimes possible to transform the problem into a mathematically equivalent problem that requires fewer computations, and sometimes removes the matrix decompositions and matrix inverses altogether. The goals of this paper are:

- i) to survey the complexity reduction techniques in the

multicarrier equalization literature,

- ii) to categorize these techniques based on their assumptions and the possible loss of optimality involved,
- iii) to compare the computational cost of the original and the efficient implementations, and
- iv) to demonstrate the tradeoff between achievable bit rate and the complexity of the efficient implementations for synthetic and measured ADSL channels.

The performance will be assessed in an identical manner for all designs. Computational cost will be in terms of real multiply-and-accumulate (MAC) operations. We will generally ignore terms that are significantly smaller than the leading term. Note that MAC comparisons are only valid in an “order of” sense, because there may be some variation in complexity depending on exactly how each design is implemented (e.g. depending on what eigen decomposition algorithm is used). Moreover, the computational cost also involves, e.g., the number of comparisons and data transfers (memory accesses), while registers or extra memory are needed to store intermediate results. The figures at the end of the paper will provide numbers based on typical parameter settings, valid in an “order of” sense. Whether a specific algorithm is suited for implementation on a certain platform then also depends on a large number of implementation aspects (such as word lengths, the degree of parallelism and operator sharing) and the chosen platform and technology. The used platforms for DSL communication vary significantly, from a flexible software-based solution on a (dedicated) DSP to a more strict (but more cost-effective) ASIC-based strategy.

This paper is a companion paper to [1]. In [1], it was shown that almost all TEQ designs take the form of maximizing a product of generalized Rayleigh quotients, and the maximum attainable bit rate of each design was assessed. In this paper, we survey computational complexity reduction techniques, and compare bit rate vs. complexity for these efficient implementations. The remainder of this paper is organized as follows. General complexity reduction techniques and fixed-point implementation issues are described in Section II. Techniques for single Rayleigh quotient designs are discussed in Sections III and IV, with a single filter or multiple filters, respectively. Techniques for designs that maximize a product of Rayleigh quotients are discussed in Section V. Section VI shows the tradeoffs between computational complexity and achievable bit rate, and Section VII concludes the paper. The notation will be:

- N is the (ID)DFT size, ν is the prefix length, $s = N + \nu$ is the symbol size, N_u is the number of used tones, \mathcal{S} is the set of used tones, N_z is the number of unused (“null”) tones, i is the tone index, k is the DMT symbol index, n is the sample index, Δ is the synchronization delay, and N_Δ is the number of values of Δ that are considered in a given TEQ design.
- For iterative designs, $I_{algorithm}$ is the number of iterations for that algorithm.
- \mathcal{F}_N and \mathcal{I}_N are the N -point DFT and IDFT matrices, respectively; \mathbf{f}_i is the i^{th} DFT row.
- The transmitted (QAM) frequency domain symbol vector at time k is X^k ; its i^{th} entry is X_i^k ; vectors \mathbf{x}^k , \mathbf{y}^k , and

\mathbf{u}^k contain the transmitted time domain samples, received samples (before the TEQ), and TEQ output samples, respectively.

- The vectors \mathbf{w} , \mathbf{h} , and $\mathbf{c} = \mathbf{h} \star \mathbf{w}$ contain the TEQ, channel, and effective channel impulse responses of orders L_w , L_h , and L_c , respectively, where \star denotes linear convolution.
- $\mathbf{0}_{m \times n}$ is the all zero matrix of size $m \times n$; \mathbf{I}_n is the identity matrix of size $n \times n$.
- $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^*$, $\mathcal{E}\{\cdot\}$ denote transpose, Hermitian, complex conjugate, and expectation, respectively.

II. COMPLEXITY REDUCTION TECHNIQUES AND FIXED-POINT ISSUES

Almost all TEQ designs can be classified as maximizing a cost function in the form of a product of generalized Rayleigh quotients [1],

$$\hat{\mathbf{w}}^{opt}(\Delta) = \arg \max_{\hat{\mathbf{w}}} \prod_{j=1}^M \frac{\hat{\mathbf{w}}^T \mathbf{B}_j(\Delta) \hat{\mathbf{w}}}{\hat{\mathbf{w}}^T \mathbf{A}_j(\Delta) \hat{\mathbf{w}}} \quad (1)$$

(or minimization of its inverse), where $\hat{\mathbf{w}}$ is usually the TEQ and where the synchronization delay Δ is a design parameter. Many TEQ designs reduce to the case of a single generalized Rayleigh Quotient ($M = 1$), which can be maximized by solving a generalized eigenvalue problem. For the more difficult case when multiple generalized Rayleigh quotients are involved ($M > 1$), numerical methods must be applied to search for the best solution. However, solutions for both the $M = 1$ and $M > 1$ cases are usually computationally expensive, and some are infeasible for a cost-effective real-time implementation, especially on programmable fixed-point DSPs. Recent literature has therefore contained much work on computationally efficient methods for calculating the optimum equalizer coefficients. This section proposes a classification scheme of these techniques, and discusses other issues relevant to fixed-point implementation of multicarrier equalizers.

A. Classification of complexity reduction techniques

Some complexity reduction techniques entail no loss of performance, whereas others use heuristics or approximations with a possible loss of performance with respect to the designs they are approximating. We categorize the various techniques as follows:

- (a) exploitation of the structure of the \mathbf{A}_j and \mathbf{B}_j matrices in (1), with no loss of performance
- (b) reuse of computations between different values of the synchronization delay (without affecting performance), or reduction of the number of delays considered (possibly degrading performance)
- (c) approximation of the \mathbf{A}_j and \mathbf{B}_j matrices (as Toeplitz, persymmetric, or circulant, for example), with an expected loss of performance
- (d) use of iterative algorithms to approximate an optimal design, with an expected performance degradation.

When \mathbf{A}_j and \mathbf{B}_j are structured, type (a) techniques exploit this structure when performing certain matrix operations. For

example, \mathbf{A}_j and \mathbf{B}_j are often constructed using correlation matrices of the transmitted and/or received signals. In [10] it was pointed out that correlation matrices are block-Toeplitz matrices and therefore some Toeplitz-based algorithms [11] could be applied to efficiently compute their inverses. Another more complicated approach is to re-use computations when computing the elements of \mathbf{A}_j and \mathbf{B}_j , as in [12] for the minimum intersymbol interference (Min-ISI) design [13], as in [14] for the maximum shortening SNR (MSSNR) design [15], as in [16] for the minimum interblock interference (MinIBI) design [17], and as in [16], [18] for the minimum delay spread (MDS) design [19].

The \mathbf{A}_j and \mathbf{B}_j generally depend on the synchronization delay Δ , and it is common to optimize designs over a range of values of Δ . Type (b) complexity reduction techniques simplify the search for the delay corresponding to optimal performance. Most designs require the solution of (1) separately for each delay, thereby making complexity proportional to the number of possible delays. If $\mathbf{A}_j(\Delta_o)$ and $\mathbf{B}_j(\Delta_o)$ depend on a delay Δ_o and only change slightly as the delay is incremented, then it may be possible to derive $\mathbf{A}_j(\Delta_o + 1)$ and $\mathbf{B}_j(\Delta_o + 1)$ from $\mathbf{A}_j(\Delta_o)$ and $\mathbf{B}_j(\Delta_o)$, rather than by recomputing the matrices entirely [14], [16]. Another approach is to re-formulate a given design to be less delay dependent, e.g. by making either \mathbf{A}_j or \mathbf{B}_j independent of the delay [19], [20], [21], [22], [23]. Heuristic approaches may also be adopted. Some equalizer designs (particularly those that explicitly optimize bit rate) show a performance which is smooth and optimal for a number of consecutive delays [8], [9], [24]; i.e. there exists a flat region on the bit rate versus synchronization delay curve. One could design the equalizer for a single delay within the expected flat region (as many vendors do), or search over a small number of possible delays [10], [18]. The expected flat region is typically near the delay of the transmission channel itself.

Type (c) complexity reduction techniques make approximations in \mathbf{A}_j or \mathbf{B}_j that may induce an acceptable performance loss. One example is to approximate a Toeplitz matrix by a circulant matrix [25], [26], which has discrete Fourier transform basis vectors as eigenvectors [27]. Using the FFT and IFFT operations, the matrix computations can be carried out very efficiently.

As another example, \mathbf{A}_j and \mathbf{B}_j can be assumed or forced to be persymmetric [28] or Toeplitz [29], leading to a linear phase (symmetric or skew-symmetric) solution for $\hat{\mathbf{w}}$ in (1). Forcing a TEQ to have linear phase leads to a substantial decrease in implementation complexity at the cost of a limited loss in bit rate [22], [28], [29], [30]. Other parameter reduction techniques (besides forcing a TEQ to have linear phase) include the reparameterization of a long FIR channel or TEQ as a pole-zero filter with fewer parameters [10], [31], and the use of the same filter (up to a scalar) for several adjacent tones in a per-tone equalizer (PTEQ) [9] or TEQ filter bank (TEQFB) [8], leading to “per group” schemes. The dual-path TEQ structure [32] can be thought of as an extreme example of a tone-grouped TEQFB, in which one TEQ is designed for all of the tones and a second TEQ is designed to maximize bit rate on a subset of tones.

In some cases, finding the solution of (1) is computationally too expensive. As a consequence, some authors resort to iterative and adaptive algorithms to obtain the solution. This is what we call a type (d) complexity reduction technique. For instance, when the equalizer design problem can be described as an eigenvalue problem, candidates to find a specific eigenvector include the generalized power method [13], gradient descent algorithms with projections [33], [34], and stochastic gradient descent algorithms with projections [35]. In addition, least-squares problems, e.g. with the PTEQ, can efficiently be solved recursively [36], [37].

Sections III, IV, and V give explicit details regarding the types (a), (b), (c), and (d) approaches described above for the cases $M = 1$ for a single filter, $M = 1$ for multiple filters, and $M > 1$ for a single filter, respectively, with M as in (1).

B. Fixed-point implementation issues

Any fixed-point number can be represented with m bits for the integer part and n bits for the fractional part. One example is the Q-format notation in Texas Instruments’ C6000 DSPs. The dynamic range of the problem determines m and the required precision determines n , although the nature of the underlying DSP induces a practical restriction on the total number of bits ($m + n$) that can be used. Commonly, the need for the integer part is eliminated via appropriate normalization of the data, which ensures that multiplication will not change the dynamic range.

In the TEQ design problem, attention should be paid to some special matrix operations. To solve (1) with $M = 1$, which requires a generalized eigendecomposition, one standard method involves computing the Cholesky factorization of the matrix \mathbf{B} ; see [1]. However, a fixed-point implementation produces $\mathbf{A} + \Delta\mathbf{A}$ and $\mathbf{B} + \Delta\mathbf{B}$ instead of \mathbf{A} and \mathbf{B} . The error of the computed eigenvalues is bounded by a multiple of $\kappa(\mathbf{B})\mu$, where $\kappa(\mathbf{B})$ is the condition number of \mathbf{B} and μ is the unit round-off [38]. When \mathbf{B} is ill-conditioned, numerical stability can be lost in the Cholesky factorization. The condition number of \mathbf{B} is often large, so even with careful choices of the binary data format, the accuracy of Cholesky factorization can be unacceptable when the dimension of \mathbf{B} (usually the TEQ length) is large.

The effect of round-off errors, called the digital noise floor, can be incorporated into the noise model explicitly, as in [8], or implicitly, as in [24].

III. SINGLE QUOTIENT CASES

This section considers reduced-complexity implementations of TEQ designs for the specific case of maximizing a single generalized Rayleigh quotient.

A. Methods for eigenvector computation

The maximization of a single generalized Rayleigh quotient requires computation of the generalized eigenvector corresponding to the largest generalized eigenvalue of the matrix pair (\mathbf{B}, \mathbf{A}) , as discussed in [1]. This section details general techniques for this math problem, and subsequent sections discuss details specific to particular TEQ designs.

One common iterative eigensolver is the generalized power method [11], which iterates

$$\mathbf{B} \hat{\mathbf{w}}_{k+1} = \mathbf{A} \mathbf{w}_k \quad (2)$$

$$\mathbf{w}_{k+1} = \frac{\hat{\mathbf{w}}_{k+1}}{\|\hat{\mathbf{w}}_{k+1}\|}, \quad (3)$$

which requires a square root and division at each step for the normalization, as well as an LU factorization [11] of \mathbf{B} to solve (2) for $\hat{\mathbf{w}}_{k+1}$. A similar approach is to alternate between gradient descent of $\mathbf{w}^T \mathbf{A} \mathbf{w}$ and renormalization to maintain $\mathbf{w}^T \mathbf{B} \mathbf{w} = 1$:

$$\hat{\mathbf{w}}_{k+1} = \mathbf{w}_k - \mu \mathbf{A} \mathbf{w}_k \quad (4)$$

$$\mathbf{w}_{k+1} = \frac{\hat{\mathbf{w}}_{k+1}}{\|\hat{\mathbf{w}}_{k+1}\|_{\mathbf{B}}}, \quad (5)$$

where $\|\mathbf{w}\|_{\mathbf{B}}^2 \triangleq \mathbf{w}^T \mathbf{B} \mathbf{w}$ and μ is a small user-defined step size.

The expensive renormalization in (3) and (5) can be avoided through the use of a Lagrangian constraint, as in [17], [39], which leads to an iterative eigensolver of the form

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu (\mathbf{B} \mathbf{w}_k - \mathbf{A} \mathbf{w}_k (\mathbf{w}_k^T \mathbf{B} \mathbf{w}_k)), \quad (6)$$

where μ is a small user-defined step size. If stochastic rank-one approximations of \mathbf{B} and \mathbf{A} are available, as in [35], then the generalized eigensolver in (6) requires $\mathcal{O}(L_w)$ multiply-adds per update. If the matrices \mathbf{A} and \mathbf{B} are used explicitly, (6) requires $\mathcal{O}(L_w^2)$ multiply-adds per update. In either case, (6) is amenable to fixed-point calculation. For comparison, an LU factorization or a Cholesky decomposition requires $\mathcal{O}(L_w^3)$ floating point operations, including many divisions.

B. The MMSE family

There are several flavors of MMSE TEQ designs, which are distinguished based on the constraint used to avoid the trivial solution $\mathbf{b} = \mathbf{w} = \mathbf{0}$. See [1] for details on the different constraints. For any MMSE method, the correlation matrices \mathbf{R}_{xx} , \mathbf{R}_{xx}^{-1} , \mathbf{R}_{xy} , \mathbf{R}_{yx} , \mathbf{R}_{yy} , and \mathbf{R}_{yy}^{-1} (definitions in [1] and [40]) must be computed. We now explain how to efficiently compute these matrices.

Typically, \mathbf{R}_{xx} is delay invariant and can be approximated as a diagonal matrix, trivializing the computation of \mathbf{R}_{xx}^{-1} . In downstream ADSL, e.g., tones 33–256 are used [2], which makes \mathbf{R}_{xx} almost the identity. The channel output autocorrelation \mathbf{R}_{yy} is also delay invariant (since it is an autocorrelation matrix), Toeplitz, and symmetric, but not diagonal. Computing the inverse of such a matrix, i.e. \mathbf{R}_{yy}^{-1} , requires only $\mathcal{O}(3L_w^2)$ instead of $\mathcal{O}(L_w^3)$ operations [11, Section 4.7.4]. Moreover, when \mathbf{R}_{yy}^{-1} is approximated by a circulant matrix [27], its inverse can be performed by means of DFTs at the cost of $\mathcal{O}((L_w + 1) \log_2(L_w + 1))$ multiply-adds [25], [26], assuming that the TEQ length $(L_w + 1)$ is a power of 2.

If the channel is known explicitly, then the matrices \mathbf{R}_{xy} , \mathbf{R}_{yx} and \mathbf{R}_{yy} can be written in terms of the channel coefficients, as in [41]. Otherwise, computation of \mathbf{R}_{xy} and \mathbf{R}_{yx} can be simplified by re-using computations from one delay Δ to the next. Note that

$$[\mathbf{R}_{yx}(\Delta + 1)]_{(0:L_w, 0:\nu-1)} = [\mathbf{R}_{yx}(\Delta)]_{(0:L_w, 1:\nu)}, \quad (7)$$

which provides the bulk of $\mathbf{R}_{yx}(\Delta + 1)$ for free. Moreover, the matrix [41], [42], [40]

$$\mathbf{R}(\Delta) = \mathbf{R}_{xx} - \mathbf{R}_{xy} \mathbf{R}_{yy}^{-1} \mathbf{R}_{yx}, \quad (8)$$

is used for a unit norm constraint on \mathbf{b} , for example. This matrix must also be computed for every delay. Using (7), we have [14]

$$[\mathbf{R}(\Delta + 1)]_{(0:\nu-1, 0:\nu-1)} = [\mathbf{R}(\Delta)]_{(1:\nu, 1:\nu)}. \quad (9)$$

In fact, (9) holds for all MMSE designs, not just for the unit norm constraint on \mathbf{b} . For each new delay, only the last column of $\mathbf{R}(\Delta + 1)$ must be computed, and the last row is obtained by symmetry. Moreover, the speed of the computation of the eigenvector of $\mathbf{R}(\Delta + 1)$ can be increased by using a shifted version of the target impulse response (TIR) for delay Δ to initialize the eigensolver for delay $\Delta + 1$ [14].

Approximations can be made to further simplify the computations. For instance, [42] first proposes the use of a representative class of channels, and then pre-computes the desired TIR for each channel. When an actual channel is measured, the TIR is selected as the one corresponding to the pre-defined channel that best matches the actual channel [42]. The TEQ is then computed to match the given channel to the precomputed TIR.

Impulse responses can also be approximated as symmetric. For an infinite length TEQ, the finite length MMSE TIR will be symmetric or skew-symmetric [29], despite the fact that the physical transmission channels are generally not symmetric. Thus, it is reasonable to enforce a finite-length symmetric TIR. This reduces the complexity of the eigensolver by a factor of 4, at a loss of about 10% of the bit rate for a 20-tap TEQ [14].

An alternate approach is to avoid the matrix computation and eigenvector solver altogether via an iterative algorithm. The MMSE design was originally proposed in a form similar to (4) and (5), except with simultaneous gradient descent on both the TIR and the TEQ [41]. However, this approach is often slow to converge [2], [43]. Moreover, this adaptive algorithm requires time-domain training, which is only available if there is training on all of the frequency bins in a given symbol. This is not the case in many multicarrier standards, for example Digital Video Broadcast [5]; and in ADSL, the training is only available during the initial start-up phase and every 69th symbol thereafter. This can in principle be remedied by using decision-direction if one is willing to tolerate a delay of an entire block before decisions can be made, perhaps by updating at the symbol rate rather than at the sample rate.

The computational complexity for designs in the MMSE family is summarized in Tables II and III at the end of Section III.

C. Chow's TEQ training algorithm

In [43], Chow *et al.* describe an efficient TEQ training algorithm. It is meant as a computationally inexpensive iterative algorithm (by reusing the available hardware such as FFT/IFFT blocks) that approximates the MMSE TEQ with unit-norm constraint on \mathbf{b} while avoiding expensive matrix inversions

[2]. However, the algorithm does not ensure convergence to the MMSE TEQ.

Each iteration consists of 4 steps: an update of the TIR \mathbf{b} , a windowing of \mathbf{b} to $\nu + 1$ taps, an update of the TEQ \mathbf{w} , and a windowing of \mathbf{w} to $L_w + 1$ taps. The updates are performed in the frequency domain ($B_i^{win} = \mathcal{F}_N[\mathbf{b}^T | \mathbf{0}^T]^T$, $W_i^{win} = \mathcal{F}_N[\mathbf{w}^T | \mathbf{0}^T]^T$, where the $\mathbf{0}$ vectors extend the filters to length N), either by an instantaneous zero forcing update or a frequency domain LMS update:

$$B_i = \frac{W_i^{win} Y_i^k}{X_i^k} \quad \text{or} \quad (10)$$

$$B_i = B_i^{win} + \mu (X_i^k)^* (W_i^{win} Y_i - B_i^{win} X_i) \quad \text{and} \quad (11)$$

$$W_i = \frac{B_i^{win} X_i^k}{Y_i^k} \quad \text{or} \quad (12)$$

$$W_i = W_i^{win} + \mu (Y_i^k)^* (B_i^{win} X_i - W_i^{win} Y_i) \quad (13)$$

The time-domain windowing is performed on the inverse FFT of W and B such that only the $L_w + 1$ and $\nu + 1$ samples with highest total energy are retained. An algorithm outline and the computational complexity for Chow's algorithm are given in Table I.

TABLE I

OUTLINE AND COMPLEXITY (PER ITERATION) OF CHOW'S ALGORITHM, USING DIVISION FOR B_i IN (10) AND LMS FOR W_i IN (12). MACS ARE REAL MULTIPLY-AND-ACCUMULATE OPERATIONS, AND N IS THE FFT SIZE.

Operation	Complexity per iteration
1. update B	$4N + N \log_2(N)$ MACs
2. window \mathbf{b}	$2N + N \log_2(N)$ MACs
3. normalize \mathbf{b}	1 square root & 1 division
4. update W	$4N + N \log_2(N)$ MACs
5. window \mathbf{w}	$2N + 2N \log_2(N)$ MACs
total:	$N(12 + 5 \log_2(N))$ MACs + 1 sqrt + 1 division

D. The MSSNR family

This section discusses the MSSNR TEQ design [15] and its extensions, including symmetric and skew-symmetric MSSNR TEQs [28], [30] and related methods such as the Minimum Inter-symbol Interference (Min-ISI) method [13], the Minimum Inter-Block Interference (Min-IBI) method [17], and Minimum Delay Spread (MDS) methods [19], [20].

First, consider the standard MSSNR design. Following [15], we define \mathbf{H} as the channel convolution matrix of size $(L_c + 1) \times (L_w + 1)$, \mathbf{H}_{win} as rows Δ through $\Delta + \nu$ of \mathbf{H} (with row indexing starting at zero), and \mathbf{H}_{wall} as the remaining rows of \mathbf{H} . Details can be found in [1]. The MSSNR design problem can be stated as [44]

$$\begin{aligned} \max_{\mathbf{w}} \quad & \left(\mathbf{w}^T \underbrace{\mathbf{H}_{win}^T \mathbf{H}_{win}}_{\mathbf{B}} \mathbf{w} \right) \\ \text{subject to} \quad & \mathbf{w}^T \underbrace{\mathbf{H}_{wall}^T \mathbf{H}_{wall}}_{\mathbf{A}} \mathbf{w} = 1. \end{aligned} \quad (14)$$

It has been shown that maximizing the energy of the “windowed” portion of the effective channel with respect to the energy of the “walled” portion leads to the same TEQ as

maximizing the energy of the “windowed” portion of the effective channel with respect to the “window” energy plus the “wall” energy, i.e. the energy of the entire channel [23], [45]. Thus, (14) is equivalent to

$$\max_{\mathbf{w}} \left(\mathbf{w}^T \underbrace{\mathbf{H}_{win}^T \mathbf{H}_{win}}_{\mathbf{B}} \mathbf{w} \right) \quad \text{subject to} \quad \mathbf{w}^T \underbrace{\mathbf{H}^T \mathbf{H}}_{\mathbf{C}} \mathbf{w} = 1. \quad (15)$$

The solution for \mathbf{w} will be the generalized eigenvector of the matrix pair (\mathbf{B}, \mathbf{C}) corresponding to the largest generalized eigenvalue λ ; note that \mathbf{C} takes on the role of \mathbf{A} in (1). Since \mathbf{C} is not a function of delay Δ , it only needs to be computed once, and since it is symmetric and Toeplitz, it can be computed in its entirety by computing only the first column. Moreover, (14) requires a Cholesky decomposition of \mathbf{A} or \mathbf{B} for each Δ , but since \mathbf{C} is not delay dependent, only one Cholesky decomposition is needed for (15). Thus, we will refer to (15) rather than (14). A similar implementation, with a generalization to reduce noise gain, was proposed in [21].

To solve (15), the $(L_w + 1) \times (L_w + 1)$ matrix \mathbf{B} must be computed for each of the possible values of Δ , and for each Δ a generalized eigenvector must be computed. Reducing the complexity can be accomplished by reducing the computation of \mathbf{B} , or by reducing the computation of the eigenvectors. One way to re-use computations is to obtain all but the first row and column of $\mathbf{B}(\Delta + 1)$ by shifting in all but the last row and column of $\mathbf{B}(\Delta)$ [14],

$$[\mathbf{B}(\Delta + 1)]_{(1:L_w, 1:L_w)} = [\mathbf{B}(\Delta)]_{(0:L_w-1, 0:L_w-1)} \quad (16)$$

in a manner similar to (9). The first column of $\mathbf{B}(\Delta + 1)$ can then be quickly obtained as follows. Since \mathbf{B} is nearly Toeplitz, instead of computing a full $(\nu + 1)$ -length dot product to get each element, only two multiply-adds are needed [12]:

$$\begin{aligned} \mathbf{B}_{(m,n)} = & \mathbf{B}_{(m+1,n+1)} + h(\Delta + \nu + 1 - m) h(\Delta + \nu + 1 - n) \\ & - h(\Delta - m) h(\Delta - n). \end{aligned} \quad (17)$$

The first row can then be obtained by transposing the first column. The \mathbf{B} for the first delay considered can also be computed almost entirely via (17).

Further reductions in complexity can be obtained by reducing the number of delay values that are searched (possibly creating sub-optimal performance), or by using a shifted version of the TEQ for delay Δ to initialize the eigensolver for the TEQ for delay $\Delta + 1$ [14].

Similar complexity reduction techniques can be applied to MSSNR variants such as the Min-ISI method [13], the Min-IBI method [17], and the Minimum Delay Spread (MDS) method [19]. For example, in the Min-ISI method, the ISI is weighted in the frequency domain, leading to a more complicated \mathbf{B} matrix (see [1]). The above techniques still apply, although (17) must be modified as in [12]. For the Min-IBI method, instead of simply updating the \mathbf{A} matrix, one can form a delay-dependent matrix \mathbf{E} of the difference of two successive \mathbf{A} matrices, and then update \mathbf{E} efficiently [16]. For the MDS method, a similar technique applies, but a first order *and* a second order error matrix must be updated for each delay increment. This is discussed in [16] with a global delay search,

TABLE II

COMPLEXITY OF OPTIMAL SINGLE RAYLEIGH QUOTIENT DESIGNS. L_h , L_w , AND L_c ARE THE LENGTHS OF THE CHANNEL, TEQ, AND EFFECTIVE CHANNEL; ν IS THE CP LENGTH; N IS THE FFT SIZE; AND N_Δ IS THE NUMBER OF DELAYS SEARCHED OVER.

Design	Original implementation	Efficient implementation
MMSE [40]	$(\frac{2}{3}\nu^3 + L_w^2\nu + L_w\nu^2) N_\Delta$	$(\frac{2}{3}\nu^3 + L_w^2 + L_w\nu) N_\Delta + 2\nu L_w^2$
Sym-MMSE [14]	$(\frac{1}{12}\nu^3 + L_w^2\nu + L_w\nu^2) N_\Delta$	$(\frac{1}{12}\nu^3 + L_w^2 + L_w\nu) N_\Delta + 2\nu L_w^2$
MSSNR [15]	$(\frac{11}{3}L_w^3 + L_cL_w^2) N_\Delta$	$\frac{8}{3}L_w^3N_\Delta + L_hL_w + 2L_w^2$
Sym-MSSNR [30], [48]	$(\frac{1}{3}L_w^3 + L_cL_w^2) N_\Delta$	$\frac{1}{3}L_w^3N_\Delta + L_hL_w + 2L_w^2$
MinISI [13]	$(\frac{11}{3}L_w^3 + \frac{1}{2}sL_w^2 + 5NL_w) N_\Delta$	$(\frac{8}{3}L_w^3 + 3NL_w + 5L_w^2 - 2\nu L_w) N_\Delta$
Sym-MinISI	$(\frac{1}{3}L_w^3 + \frac{1}{2}sL_w^2 + 5NL_w) N_\Delta$	$(\frac{1}{3}L_w^3 + 3NL_w + 5L_w^2 - 2\nu L_w) N_\Delta$
MinIBI [17]	$(\frac{11}{3}L_w^3 + L_cL_w^2) N_\Delta$	$\frac{8}{3}L_w^3N_\Delta + L_cL_w^2$
MDS [19]	$(\frac{2}{3}L_w^3 + L_cL_w^2) N_\Delta$	$\frac{2}{3}L_w^3N_\Delta + 2L_cL_w^2$
Dual Path [32]	$(10L_w^3 + 2L_cL_w^2 + \frac{1}{2}sL_w^2 + 5NL_w) N_\Delta$	$(8L_w^3 + \frac{2}{3}\nu^3 + 3NL_w) N_\Delta + 2L_cL_w^2$

and a similar (and more efficient) method is discussed in [18] wherein the updates can be performed while only computing the \mathbf{A} matrix for selected delays.

The Min-IBI and MDS designs are part of a larger class defined in [21]. Consider minimizing

$$J = \alpha \frac{\sum_n f(n - n_{mid}) |\mathbf{c}_n|^2}{\sum_n |\mathbf{c}_n|^2} + (1 - \alpha) \frac{\sigma_n^2}{\sigma_x^2 \|\mathbf{c}\|^2}, \quad (18)$$

where n_{mid} is the desired “middle” of the non-zero portion of the effective channel and $f(\cdot)$ is an arbitrary function. The case $\alpha = 1$ and $f(n) = n^2$ leads to an algorithm that minimizes the delay spread (MDS) of the effective channel [19]. The case $\alpha = 1$ and

$$f(n) = \begin{cases} 0, & -\frac{\nu}{2} \leq n \leq \frac{\nu}{2} \\ 1, & \text{otherwise} \end{cases} \quad (19)$$

leads to an algorithm which minimizes $\mathbf{w}^T \mathbf{A} \mathbf{w}$ while keeping $\mathbf{w}^T \mathbf{C} \mathbf{w} = 1$ [with \mathbf{A} and \mathbf{C} as in (14) and (15)]. For general values of α (“Noise-limited MSSNR,” or NL-MSSNR), (17) still applies, since the noise term does not change the near-Toeplitz structure of the matrices.

Hitherto, the MSSNR complexity reduction techniques that we have discussed have focused on finding the same SSNR-maximizing solution at a lower cost. An alternate philosophy is to use approximations or iterative algorithms to find nearly the same solution at reduced cost. Symmetric MSSNR (Sym-MSSNR) constrains the impulse response to have linear phase (symmetric or skew-symmetric), so only half of the TEQ coefficients need to be computed. This reduces the complexity of the eigensolver by a factor of 4. However, the bit rates of the constrained MSSNR solution drop by about 3% for ADSL and VDSL systems [14], [28], [30].

One iterative method of solving (15) is the generalized power method of (2) and (3). Other iterative/adaptive MSSNR techniques have been proposed in [33] and [35]. These techniques are similar to the power method, but perform a gradient descent of a cost function (rather than a matrix multiply) with a periodic renormalization. Alternatively, (6) can be used to avoid the renormalization.

The computational complexity for designs in the MSSNR family is summarized in Tables II and III at the end of Section III.

E. The CNA adaptive equalizer

In many multicarrier standards [3], [4], [5], [6], the frequency-domain input signal is zero-padded before transmission, so some frequency bins X_i are null (zero). In the absence of ISI, each corresponding receiver FFT output U_i is expected to also be zero; whereas in the presence of ISI, it may not be zero. The carrier-nulling algorithm (CNA) [46], [47] performs a stochastic gradient descent of the output energy in the set of N_z null carriers, where a periodic renormalization is used to avoid $\mathbf{w} = \mathbf{0}$. This constrained minimization problem is in fact an eigenvector problem, and the CNA algorithm is a low-complexity adaptive eigenvector estimator which equalizes the channel to an impulse, rather than shortening it to a window [46]. The computational complexity of CNA is given in Table III at the end of Section III.

F. Complexity comparison

Table II compares the computational complexity of the optimal single-Rayleigh quotient designs considered in this section. Formulas are given for the designs as originally proposed, as well as for the more efficient (yet mathematically equivalent) implementations discussed in this section. Table III compares the computational complexity of approximate iterative designs that attempt to maximize a single generalized Rayleigh quotient. Each table entry was determined by going through the designs in Sections III-A through III-E line by line and adding up numbers of operations. The “efficient implementation” column assumes the use of techniques discussed in this section such as (16), whereas the “original implementation” column computes each design as it was originally presented.

IV. MULTIPLE FILTERS, EACH WITH A SINGLE QUOTIENT

The per tone equalizer (PTEQ) and time domain equalizer filter bank (TEQFB) designs treated in this section discontinue the practice of using only one filter to equalize the channel across the entire bandwidth, and instead assign each sub-channel a potentially different equalizing filter. Both methods use the achievable bit rate as their objective function, thus breaking away from the practice of earlier methods (e.g. in Sections II and III) that maximized objective functions that were not necessarily related to the bit rate of the system. Both methods were reviewed in [1], focusing on the equalizer

TABLE III

COMPLEXITY OF ITERATIVE SINGLE RAYLEIGH QUOTIENT DESIGNS. L_h , L_w , AND L_c ARE THE LENGTHS OF THE CHANNEL, TEQ, AND EFFECTIVE CHANNEL; ν IS THE CP LENGTH; N IS THE FFT SIZE; N_Δ IS THE NUMBER OF DELAYS SEARCHED OVER; AND I_x IS THE NUMBER OF ITERATIONS FOR METHOD "X."

Design	Complexity
Adaptive MMSE [41]	$(4\nu + 2L_w) I_{mmse} N_\Delta$
Chow [43]	$(12N + 5N \log_2(N)) I_{chow}$
MDS as in [18]	$2L_c L_w^2 + \frac{2}{3} L_w^3 I_{m ds}$
MSSNR via (2)	$(L_c L_w^2 + \frac{3}{2} L_w^2 I_{power}) N_\Delta$
MSSNR via (6)	$L_w^2 (L_c + 2I_{mssnr}) N_\Delta$
MERRY [35]	$4L_w I_{merry} N_\Delta$
SAM [49]	$4L_w (L_c - \nu) I_{sam}$
Nafie & Gatherer [33]	$2L_w I_{ng} N_\Delta$
CNA [46]	$N (N_z + L_w) I_{cna} N_\Delta$

architecture and design premises. This section describes the implementation of these methods, with emphasis on the computational complexity encountered during equalizer coefficient initialization and data transmission.

A. Per-tone equalizer

The PTEQ architecture [9] allows one equalizer in the *frequency domain* for each subchannel. PTEQ moves the equalization after the FFT block and incorporates the functions of the FEQ as well. The PTEQ derivation (details in [9]) starts from the conventional single time domain architecture and uses the linearity of all operations to arrive at the frequency domain equalizer \mathbf{w}_i for subchannel i . We can write the equalized output on tone i as [36]

$$\tilde{X}_i^k = \bar{\mathbf{v}}_i^T \mathbf{F}_i \mathbf{y}^k \quad (20)$$

where \tilde{X}_i^k is the estimate of the transmitted symbol X_i^k in subchannel i , $\bar{\mathbf{v}}_i^T$ are PTEQ equalizer coefficients for the i^{th} tone [9], \mathbf{y}^k is a vector of $N + L_w$ samples in symbol k , and

$$\mathbf{F}_i = \left[\begin{array}{c|c|c} \mathbf{I}_{L_w} & \mathbf{0} & -\mathbf{I}_{L_w} \\ \hline \mathbf{0} & & \mathbf{f}_i \end{array} \right] \quad (21)$$

Here, \mathbf{f}_i computes the i^{th} output of the N -point DFT. The optimal coefficients are then arrived at by minimizing

$$J(\mathbf{v}_i) = E[|\bar{\mathbf{v}}_i^T \mathbf{F}_i \mathbf{y}^k - X_i^k|^2] \quad (22)$$

The cost function (22) can be minimized using various direct methods for solving least-squares or MMSE problems, either with or without the knowledge of the channel state information and noise and signal statistics. Direct methods require a transmission of a training sequence of K symbols and a large number of computations, although an adaptive method would have lower numerical complexity.

An adaptive PTEQ method minimizing (22) based on recursive least squares (RLS) with inverse updating is given in [36]. This RLS-based method estimates the covariance matrix of the equalizer input $\mathbf{R}_i^k = \sum_{j=1}^k (\mathbf{F}_i \mathbf{y}^j)^* (\mathbf{F}_i \mathbf{y}^j)^T$ and decomposes it into $(\mathbf{R}_i^k)^{-1} = (\mathbf{L}_i^k)^H \mathbf{L}_i^k$ where \mathbf{L}_i^k is a lower triangular matrix. The algorithm then for K iterations directly improves the estimate of \mathbf{L}_i^k [without recomputing $(\mathbf{R}_i^k)^{-1}$] and uses the byproduct of that refinement in an RLS-based adaptation

for the equalizer coefficients \mathbf{v}_i^k . The reader should see [36] for further details. Most important, the inclusion of the sliding FFT difference terms induces a special structure in \mathbf{L}_i^k where the matrix $\mathbf{L}_i^k(0 : L_w - 1, 0 : L_w - 1)$ is real and equal for all subchannels and only the last row of \mathbf{L}_i^k is different and complex. A combined RLS-LMS initialization technique is described in [37].

The RLS initialization complexity, assuming that all of the available subchannels are used, is $\frac{N}{2}(20L_w + 30) + 3L_w^2 + 7L_w$ MACs/iteration, while the RLS-LMS complexity under the same assumptions is $\frac{N}{2}(4L_w + 13) + 3L_w^2 + 7L_w$ MACs/iteration [37].

Note, that in contrast to direct PTEQ initialization methods, the RLS PTEQ does not need knowledge of the channel state and the noise statistics. The simulation results reported in [36] show that for the given examples, the RLS-based initialization algorithm achieves a data rate similar to the direct methods for the same number of training symbols.

B. Time domain equalizer filter bank

A per tone method with a *time domain* equalizer for each subchannel is the TEQ Filter Bank (TEQFB) [8]. The method models the subchannel SNR as a single generalized Rayleigh quotient

$$\text{SNR}_i = \frac{\mathbf{w}^T \tilde{\mathbf{B}}_i \mathbf{w}}{\mathbf{w}^T \tilde{\mathbf{A}}_i \mathbf{w}}, \quad (23)$$

where the complex-valued Hermitian symmetric $(L_w + 1) \times (L_w + 1)$ matrices are

$$\begin{aligned} \tilde{\mathbf{A}}_i &= 2S_{x,i} \left(\underbrace{\mathbf{H}_{wall,1}^T \mathbf{V}_i \mathbf{V}_i^H \mathbf{H}_{wall,1}}_{\mathbf{A}_{i,h}} + \underbrace{\mathbf{H}_{wall,2}^T \mathbf{W}_i \mathbf{W}_i^H \mathbf{H}_{wall,2}}_{\mathbf{A}_{i,t}} \right) \\ &\quad + \underbrace{\mathbf{Q}_i^{\text{noise}} \mathbf{R}_n [\mathbf{Q}_i^{\text{noise}}]^H}_{\mathbf{A}_{i,awgn} + \mathbf{A}_{i,next} + \mathbf{A}_{i,adc}} + \frac{\sigma_{\text{DNF}}^2}{\mathbf{w}^T \mathbf{w}} \mathbf{I}_{L_w+1}, \quad (24) \\ \tilde{\mathbf{B}}_i &= S_{x,i} \mathbf{H}^T \mathbf{Q}_i^{\text{circ}} [\mathbf{Q}_i^{\text{circ}}]^H \mathbf{H}. \quad (25) \end{aligned}$$

$\mathbf{H}_{wall,1}$ and $\mathbf{H}_{wall,2}$ are convolution matrices composed of the head and tail portions of the channel, $\mathbf{h}(0 : \Delta - 1)$ and $\mathbf{h}(\Delta + \nu + 1 : N)$, respectively; \mathbf{V}_i and \mathbf{W}_i are upper and lower triangular Hankel matrices made from the i^{th} row of the DFT matrix, \mathbf{f}_i ; $\mathbf{Q}_i^{\text{noise}}$ and $\mathbf{Q}_i^{\text{circ}}$ are Hankel matrices made from \mathbf{f}_i that account for the DMT symbol structure; \mathbf{R}_n is the noise (AWGN, crosstalk and finite precision of analog-to-digital converter) covariance matrix; and σ_{DNF}^2 is the power of the noise due to the fixed-point arithmetic [8]. The TEQFB design involves computing $\tilde{\mathbf{A}}_i$ and $\tilde{\mathbf{B}}_i$ as in (24) and (25), then maximizing a generalized Rayleigh quotient for each subchannel. The derivations of $\tilde{\mathbf{A}}_i$ and $\tilde{\mathbf{B}}_i$ are given in [8], but only the final equations (23), (24), and (25), are needed for implementation. The efficient TEQFB initialization procedure in [50] exploits the structure of these matrices to reduce the number of computations necessary for their initialization compared to a straight multiply-update approach that would be taken if no such structure existed.

1) *Subchannel SNR model numerator*: Element k, j of $\tilde{\mathbf{B}}_i$ can be written as

$$\tilde{\mathbf{B}}_i[k, j] = N \mathbf{f}_i[k - j] \underbrace{\left(\sum_{m=0}^{N+L_w-2-k} \mathbf{h}[m] \mathbf{f}_i[m] \right)}_{\mathbf{t}_i[k]} \times \underbrace{\left(\sum_{l=0}^{N+L_w-2-j} \mathbf{h}[l] \mathbf{f}_i[-l] \right)}_{\mathbf{t}_i^*[k]} \quad (26)$$

where $0 \leq k \leq L_w$. A recursive formula for the computation of elements $\mathbf{t}_i[k]$ is given in [50]. Computation of the lower triangle elements of $\tilde{\mathbf{B}}_i$ requires order $\mathcal{O}(\max(L_w^2, N))$ real multiply-accumulate (MAC) operations.

2) *Subchannel SNR model denominator*:

a) *AWGN and ADC component*: The AWGN and ADC contribution is captured in $\mathbf{A}_{i,awgn} + \mathbf{A}_{i,adc}$, which is a Hermitian symmetric and Toeplitz matrix. Thus, it is only necessary to compute its first column. The remaining elements are then defined by the Hermitian Toeplitz structure.

b) *Near-end crosstalk component*: The matrix $\mathbf{A}_{i,next} = \mathbf{Q}_i^{\text{noise}} \mathbf{R}_{next} [\mathbf{Q}_i^{\text{noise}}]^H$ where the noise covariance matrix \mathbf{R}_{next} is symmetric and Toeplitz. Hence,

$$\mathbf{A}_{i,next}[k, j] = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \mathbf{R}_{next}[|n - m + i - j|, 0] \mathbf{f}_i[m - n] \quad (27)$$

The dependence of the element $\mathbf{A}_{i,next}[k, j]$ on the index $i - j$ of matrix \mathbf{R}_{next} means that $\mathbf{A}_{i,next}$ also is symmetric and Toeplitz and only the first column needs to be calculated. The algorithm requires $\mathcal{O}(4N + 15L_w)$ real MACs [50].

c) *Channel tail component*: Define the temporary Hankel matrix $\mathbf{X}_i = \mathbf{H}_{wall,2}^T \mathbf{W}_i$. It is shown in [50] that the element $\mathbf{A}_{i,t}^{k,j}$ is recursively defined as

$$\mathbf{A}_{i,t}[L_w - 1, j] = \sum_{g=L_w-1}^{\Delta-1} \mathbf{X}_i[0, g] \sum_{s=0}^{\Delta-L_w+j} \mathbf{X}_i^H[0, s], \quad (28)$$

$$\mathbf{A}_{i,t}[k, j] = \mathbf{A}_{i,t}[k + 1, j + 1] + \mathbf{X}_i[0, k] \mathbf{X}_i^H[0, j]. \quad (29)$$

Computation of the lower triangle half of $\mathbf{A}_{i,t}$ requires exactly $7L_w^2 + 4L_w\Delta + 5\Delta - 3L_w$ MACs.

d) *Channel head component*: Define $\mathbf{Z}_i = \mathbf{H}_{wall,1}^T \mathbf{V}_i$. A recursive relationship can be defined between the elements of the k^{th} row of \mathbf{Z}_i :

$$\mathbf{Z}_i[k, j+1] = \mathbf{f}_i[1] \mathbf{Z}_i[k, j] + \mathbf{h}[(N+L_w-2)-k-(j+1)]. \quad (30)$$

This algorithm for calculation of $\mathbf{A}_{i,h}$ will update the value of all of the matrix elements with the contribution of the product the j^{th} column of \mathbf{Z}_i and the j^{th} row of \mathbf{Z}_i^H for $0 < j < N - \nu - \Delta + L_w - 1$. The algorithm requires order $\mathcal{O}(NL_w^2)$ MACs.

The term of $\mathcal{O}(NL_w^2)$ dominates the complexity of the subchannel SNR calculation. Once the subchannel SNRs are calculated, a generalized eigenvector problem must be solved for each of the N_u used tones. Solving a symmetric generalized eigenvector problem of dimension L_w takes approximately

$14L_w^3$ operations [11, p.464]. The initialization complexity of a TEQ, a PTEQ, and a TEQFB are compared in Table IV. (In Section VI, specifically in Fig. 4, typical parameter values are substituted into Table V to provide a graphical comparison.)

TABLE IV

EQUALIZER INITIALIZATION COMPUTATIONAL COMPLEXITY, ASSUMING $\frac{N}{2}$ DATA-CARRYING SUBCHANNELS. L_h and L_w ARE THE LENGTHS OF THE CHANNEL AND TEQ; N IS THE FFT SIZE; $s = N + \nu$ IS THE SYMBOL SIZE; N_Δ IS THE NUMBER OF DELAYS SEARCHED OVER; AND I_x IS THE NUMBER OF ITERATIONS FOR METHOD "X."

Architecture	MACs
TEQ (via MSSNR)	$\frac{8}{3}L_w^3 N_\Delta + L_h L_w + 2L_w^2$
PTEQ	$\frac{1}{2}N(9L_w s^2 + 8L_w^2 s)$
PTEQ (via RLS)	$\left(\frac{N}{2}(20L_w + 30) + 3L_w^2\right) I_{rls}$
PTEQ (via RLS+LMS)	$\left(\frac{N}{2}(4L_w + 13) + 3L_w^2\right) I_{lms}$
TEQFB	$\frac{1}{2}N(NL_w^2 + 14L_w^3)$

C. Data transmission complexity

The computational complexity and memory requirements during data transmission (as opposed to initialization) for the TEQ, TEQFB, and PTEQ architectures are shown in Table V. (In Section VI, specifically in Fig. 4, typical parameter values are substituted into Table V to provide a graphical comparison.) Memory requirements depend more on the equalizer architecture that is used (i.e. TEQ, TEQ-FB, PTEQ) than on the algorithm used to design the equalizer, hence this is the only section of the paper in which we compare memory use. Thus, a TEQFB can have lower memory needs than a PTEQ;

TABLE V

DATA TRANSMISSION COMPUTATIONAL COMPLEXITY FOR SAMPLE RATE $f_s = 2.208$ MHz, SYMBOL RATE $f_{\text{sym}} = 4$ kHz, AND ASSUMING $\frac{N}{2}$ DATA-CARRYING SUBCHANNELS. L_w IS THE TEQ LENGTH, N IS THE FFT SIZE, AND ν IS THE CP LENGTH.

TEQ	MACs / s	Storage Words
convolution	$N(L_w + 1)f_{\text{sym}}$	$(L_w + 1)$
FFT	$2N \log_2 N f_{\text{sym}}$	$4N + 2\nu$
FEQ	$2N f_{\text{sym}}$	N
PTEQ	MACs / s	Storage Words
FFT	$2N \log_2 N f_{\text{sym}}$	$4N + 2\nu$
Difference terms	$L_w f_{\text{sym}}$	L_w
Combiner	$N(L_w + 2)f_{\text{sym}}$	$N(L_w + 1)$
TEQFB	MACs / s	Storage Words
TEQ FB	$\frac{N}{2}L_w f_s$	$\frac{N}{2}(L_w + 1)$
Goertzel FB	$(N^2 + N)f_{\text{sym}}$	$4N$
FEQ	$2N f_{\text{sym}}$	N

however, a TEQFB has significantly higher computational requirements during data transmission that make it too expensive for cost-effective embedded implementation today. If equalization should take at most 5% of the processor time and 17-tap subchannel equalizers are used, a TEQFB becomes feasible for single-core processors running at 240 MHz (multiple core processors can do with a lower speed due to the possibility of a highly parallelized TEQFB implementation).

V. MORE THAN ONE RAYLEIGH QUOTIENT

Although the most popular single TEQ design methods are based on solving a generalized eigenvalue problem (see

Section III), they are in general not optimal in the sense of bit rate maximization. Several attempts have been made to design a TEQ that maximizes bit rate. As in Section IV, the bit rate is the underlying objective function, but in this section we are focusing on *single* TEQ design. These designs vary in nature, but they can all be described in a common way as a maximization of a product of multiple Rayleigh quotients, and hence lead to a non-linear optimization problem [1]. In this section, each method is briefly reviewed, some non-linear optimization procedures are presented, and complexity is tabulated.

A. Maximum geometric signal-to-noise ratio (MGSNR) method

The geometric SNR for a DMT system [40], [51], [52], [53] is defined as

$$\text{SNR}_{\text{geom}} = \left(\prod_{i \in \mathcal{S}} \text{SNR}_i \right)^{\frac{1}{N_u}}, \quad (31)$$

where \mathcal{S} is the set of tones that carry data and N_u is the number of tones in that set. For high SNR and fixed transmission bandwidth, maximizing the bit rate is equivalent to maximizing the geometric SNR in (31) [40], [52]. Assuming equal power distribution in all subchannels and that the TEQ \mathbf{w} and the TIR \mathbf{b} are related through $\mathbf{R}_{yx}\mathbf{b} = \mathbf{R}_{yy}\mathbf{w}$, maximization of (31) can be rewritten as [40], [52]

$$\mathbf{b}_{\text{GSNR}}^{\text{opt}} = \arg \max_{\mathbf{b}} \sum_{i \in \mathcal{S}} \ln \mathbf{b}^T \mathbf{G}_i \mathbf{b} \quad (32)$$

$$\text{s.t. } \mathbf{b}^T \mathbf{b} = 1 \text{ and } \mathbf{b}^T \mathbf{R}_{\Delta} \mathbf{b} \leq \text{MSE}_{\text{max}}, \quad (33)$$

where \mathbf{R}_{Δ} is given in (8), \mathbf{G}_i is a matrix with DFT coefficients related to tone i , and the second constraint avoids equalization to a single spike.

The optimization problem in (32) is a constrained nonlinear optimization problem and does not have a closed form solution. In [40], [52], the MATLAB[®] optimization toolbox was used to solve (32). Recently, Laskarian and Kiaei proposed

TABLE VI

ALGORITHM OUTLINE AND COMPLEXITY TO SOLVE (32) [34]. L_w IS THE TEQ LENGTH, N IS THE FFT SIZE, AND ν IS THE CP LENGTH.

Operation	Complexity (flops)
1. Initialization:	
1.1 Calculation of \mathbf{R}_{Δ}	$\frac{13}{4}L_w^2 + L_w^2\nu + L_w\nu^2$
1.2 Eigenvalue decomposition of \mathbf{R}_{Δ}	$(\nu + 1)^3$
2. Per iteration:	
2.1 Gradient computation	$3N_u\nu^2 + 2N_u\nu$
2.2 Descent update	$\nu + 1$
2.3 Projection onto convex set	$2(\nu + 1)$

to use the gradient projection method in conjunction with projection onto convex sets as a means to find the solution of (32) [34], [54]. First, they remove the unit-norm constraint on the TIR since the origin is not a trivial solution of the problem. Then they observe that the second constraint represents a closed convex set in the $(\nu + 1)$ -dimensional Euclidean space: $\{\mathbf{b} \in \mathbb{R}^{\nu+1} | \mathbf{b}^T \mathbf{R}_{\Delta} \mathbf{b} \leq \text{MSE}_{\text{max}}\}$. Using the convexity property of the constraint set along with a suitable iterative

descent method efficiently leads them to a stationary point. A feasible descent direction is obtained by taking a step along the negative gradient of the cost function followed by a projection on the constraint set. Based on [34], an outline of the algorithm and its complexity are given in Table VI.

B. Maximum bit rate (MBR) method

In [13] and [55] the Maximum Bit Rate (MBR) TEQ design method was presented to maximize the bit rate *at the TEQ output*. The approximate subchannel SNR model is given by

$$\widetilde{\text{SNR}}_i(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{A}_i \mathbf{w}}{\mathbf{w}^T \mathbf{B}_i \mathbf{w}}, \quad (34)$$

where \mathbf{A}_i and \mathbf{B}_i describe the signal and noise components for tone i respectively (see [13], [55] for details). The approximate bit allocation is then given by

$$b_{\text{DMT}}(\mathbf{w}) = \sum_{i \in \mathcal{S}} \log_2 \left(1 + \frac{\widetilde{\text{SNR}}_i(\mathbf{w})}{\Gamma} \right), \quad (35)$$

and the bit rate is $f_{\text{sym}} \cdot b_{\text{DMT}}(\mathbf{w})$, where $f_{\text{sym}} = 4$ kHz is the symbol rate. Arslan, Evans and Kiaei [13] propose to maximize this non-linear bit rate equation by using an advanced iterative Newton-like optimization algorithm, such as the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton algorithm [56] in the MATLAB[®] optimization toolbox. The authors conclude that the MBR procedure is computationally expensive and not well-suited for real-time implementation on a programmable digital signal processor.

C. Bit rate maximizing TEQ (BM-TEQ) and maximum data rate TEQ (MDR-TEQ)

The TEQ procedures of Sections V-A and V-B contain many approximations. Vanbleu *et al.* and Milosevic *et al.* independently suggested very similar TEQ design procedures for bit rate maximization, referred to as the bit rate maximizing TEQ (BM-TEQ) in [24] and the maximum data rate (MDR) TEQ in [57]. In both cases, the bit rate maximization problem can be written as

$$\arg \max_{\mathbf{w}} b_{\text{DMT}}(\mathbf{w}) = \arg \min_{\mathbf{w}} \sum_{i \in \mathcal{S}} \log_2 \frac{\mathbf{w}^T \mathbf{B}_i \mathbf{w}}{\mathbf{w}^T \mathbf{A}_i \mathbf{w}}, \quad (36)$$

where \mathbf{A}_i and \mathbf{B}_i are tone dependent matrices (see [8], [24] for details). In [24], the bit rate is maximized *at the FEQ output* while the authors of [8] have chosen to maximize a slightly differently defined bit rate *at the FFT output* (or FEQ input).

Minimizing (36) is an unconstrained nonlinear optimization problem. Due to the difficulty of solving such a problem, in this section we focus on solving the optimization problem rather than on efficiently computing \mathbf{A}_i and \mathbf{B}_i . When channel knowledge and noise statistics are available, standard non-linear iterative optimization algorithms, such as iterative (quasi-)Newton and simplex algorithms, can be applied to solve (36). Milosevic *et al.* [8] use the Almqvist-Levin iteration [58] to find a root of the gradient of (36) corresponding to the closest *local* maximum to the initial point. The initial \mathbf{w} can

be the TEQFB subchannel equalizer that results in the highest value of (36) for all subchannels of interest.

In [24], the authors observed (without proof) that, although the BM-TEQ cost function is often multimodal, the different local minima yield nearly optimal performance. Therefore, a *recursive* TEQ update based on a Gauss-Newton-like search direction was proposed to solve (36),

$$\mathbf{w}^k = \mathbf{w}^{k-1} - (\mathbf{H}^k(\mathbf{w}^{k-1}))^\dagger \mathbf{g}^k(\mathbf{w}^{k-1}), \quad (37)$$

where $\mathbf{g}^k(\mathbf{w}^{k-1})$ is the gradient of the cost function, $(\cdot)^\dagger$ is the pseudo-inverse, and $\mathbf{H}^k(\mathbf{w}^{k-1})$ is a positive semidefinite approximation of the Hessian of the cost function. The algorithm is recursive (or adaptive) since the TEQ update is based on continuously incoming data and not on noise statistics nor channel knowledge. The gradient and Hessian in (37) are obtained by [59]

- recursively estimating the Cholesky factor of $\mathcal{E}\{[Y_i^k \quad \Delta^T \mathbf{y}^k]^T [(Y_i^k)^* \quad \Delta^T \mathbf{y}^k]\}$ and the crosscorrelations $\mathcal{E}\{\Delta \mathbf{y}^k (X_i^k)^*\}$, $\mathcal{E}\{Y_i^k (X_i^k)^*\}$ for $\forall i \in \mathcal{S}$, where Y_i^k is the DFT output of the received signal for tone i , $\Delta \mathbf{y}^k$ are L_w difference terms of the received time-domain signal, and X_i^k is the transmitted frequency domain symbol for tone i ,
- evaluating the expressions for \mathbf{H}^k and \mathbf{g}^k as functions of \mathbf{w}^{k-1} .

An outline of the algorithm of [59] and its complexity are given in Table VII. The algorithm converges very fast (less than 100 iterations) and allows for further adaptation and tracking during data transmission.

TABLE VII

ALGORITHM OUTLINE AND COMPLEXITY TO SOLVE (36) [59]. L_w IS THE TEQ LENGTH, N IS THE FFT SIZE, AND ν IS THE CP LENGTH.

Operation	Complexity (flops/iteration)
1. Statistics update	$L_w^2 + 2N_u L_w$
2. Gradient and Hessian computation	$2N_u L_w^2 + N_u L_w$
3. Pseudo-inverse calculation of Hessian	L_w^3
4. TEQ update	L_w^2

VI. COMMUNICATIONS PERFORMANCE EVALUATION

This section presents a performance comparison of popular optimal designs and the low-complexity implementations presented in this paper. Section VI-A describes the communications performance measure, Section VI-B describes the synthetic data environment, and Section VI-C describes the data sets from which we extracted channel measurements, Section VI-D discusses the bit rate results, and Section VI-E compares the computational complexity of optimal and efficient designs.

A. Performance and complexity measures

The performance measure adopted in this paper is the achievable bit rate for a fixed probability of error (10^{-7}). Bit allocation on subcarrier i is calculated by

$$b_i = \log_2 \left(1 + \frac{\text{SNR}_i}{\Gamma_{sim}} \right) \quad (38)$$

where SNR_i is the *measured* SNR at the i th subcarrier, obtained by averaging the output signal to interference and noise ratio at the FEQ output, and

$$\Gamma_{sim} \text{ (in dB)} = \Gamma_{gap} + \text{system margin} - \text{coding gain}. \quad (39)$$

Here, the ‘‘SNR gap’’ $\Gamma_{gap} = 9.8$ dB corresponds to 10^{-7} bit error rate, the system margin is 6 dB, and the coding gain is 5 dB [60]. The achievable bit rate is then $R = f_{sym} \sum_i b_i$, where $f_{sym} = 4$ kHz is the symbol rate and $\sum_i b_i$ is the number of bits per DMT symbol.

We will assess computational complexity in terms of the number of multiply-and-accumulate operations required. The comparison is only valid in an ‘‘order-of’’ sense, since there will be some variation in the numbers depending on exactly how the design is implemented. The goal of this section is to quantify the trade-offs between these performance and complexity measures.

B. Synthetic data simulation environment

The physical media for ADSL channels are telephone lines, which are metallic twisted pairs of wires. This paper uses a group of eight loops widely used in research simulations, called the carrier serving area (CSA) loops, which were proposed by Bell Systems in the early 1970s. The impulse responses of these test loops can be obtained by using the *LINEMOD* software [61], which is based on two-port network transmission line theory [2, Sec. 3.5]. The simulations use the 8 CSA loops (available in [62]) in series with a 5th order Chebyshev Type I high-pass filter with cut-on frequency at 4.8 KHz and a high-pass filter with cut-on at 138 kHz, which serve to filter out the telephone voiceband signal and to filter out the upstream signals, respectively.

Sources of DSL noise can be classified as impulse noise, background noise, and crosstalk between wires. Impulse noise consists of impulses occurring at random times, and background noise is usually modelled as additive white Gaussian noise (AWGN). Crosstalk is further divided into near-end crosstalk (NEXT) and far-end crosstalk (FEXT). NEXT tends to be dominant in ADSL transmissions [2]. Our simulations use NEXT corresponding to 5 ISDN disturbers plus AWGN at -140 dBm/Hz, distributed over the entire bandwidth (relative to 23 dBm input signal power).

C. AST data set

Applied Signal Technology has generously provided the authors with several measured ADSL data signals. The voltage signal from a telephone line was recorded, sampled at 2.5 MHz, and digitized. The signal was frequency-duplexed so that the upstream and downstream channels lay in two distinct frequency bands [2].

We resampled the data to exactly 2.208 MHz, and then used the C-REVERB2 training sequence to perform a (downstream) channel estimate. The estimated channel is given by

$$\hat{\mathbf{h}} = \mathcal{F}_N^{-1} \left(\frac{1}{1000} \sum_{k=1}^{1000} \frac{\mathcal{F}_N \mathbf{x}^k}{\mathcal{F}_N \mathbf{y}^k} \right), \quad (40)$$

where vector division is performed pointwise. Here, \mathcal{F}_N is the DFT matrix, \mathbf{x}^k is the k^{th} period of the chosen C-REVERB2 signal, and \mathbf{y}^k is the corresponding received signal over the same period. The C-REVERB2 signal is generated according to [63, Sec. 10.4.5]. Fig. 1 shows the impulse responses of the estimated channels for two sets of recorded data, including all transmit and receive filters. The x-axis is the sample number. There are $246.4\mu\text{s}$ per symbol and 544 samples per symbol, hence one sample is approximately $0.45\mu\text{s}$. The y-axis is the amplitude of the samples of the channel impulse response.

D. Bit rate assessment

The FFT size is $N = 512$ and the CP length is $\nu = 32$, as in the G.DMT standard for downstream transmission. Delay optimization has been applied to all methods. The TEQ length is 17 taps, which is a common choice in practice. Fig. 2(a) compares the bit rate, averaged over the eight CSA test loops, for 16 common equalizer designs, and Fig. 2(b) shows the bit rates for the measured channels from Fig. 1.

The dual path TEQ computed the MMSE, MSSNR, and MDS TEQs, and picked the best one for one path, then designed a Min-ISI TEQ optimized over a subset of tones for the second path. As such, it outperforms the other designs in Fig. 2. The designs that make use of approximations usually induce a small loss in bit rate with respect to their optimal counterparts. The SAM algorithm seems to become stuck in false local (but not global) minima of the SAM cost function, leading to a performance loss. The last six algorithms, which explicitly attempt to maximize the bit rate, are listed in order of fewer approximations and more general structures; hence performance is expected to (and generally does) increase as we move left to right on the bar chart.

E. Complexity comparison

In this section, we plot the computational complexity of various equalizer designs, architectures, and implementations, for typical parameter values. The FFT size is $N = 512$, the CP length is $\nu = 32$, the DMT symbol length is $s = N + \nu = 544$, the equalizer length is $L_w = 16$, the number of tones used for data transmission is $N_u = 256$, the length of the channel estimate is $L_h = 512$, and the symbol rate is $f_s = 2.208 \cdot 10^6$ symbols/second. The figures in this section simply plot the values obtained by substituting these parameter values into the tables throughout Sections III to V.

All of the single Rayleigh quotient designs search over 64 delay values, except for Lopez-Valcarce's MDS implementation (which explicitly reduces the number of delays), Chow's algorithm (which includes a delay search as part of the iteration), and SAM (which automatically selects a delay without a search). All of the algorithms which explicitly attempt to maximize the bit rate are relatively insensitive to the delay (within a certain range), hence only one delay is tested for these algorithms. The number of iterations for the iterative designs are given in Table VIII. These numbers were chosen based on the convergence rates reported when the algorithms were proposed.

The left side of Fig. 3 gives the complexity of TEQ designs that maximize a single generalized Rayleigh quotient: the

TABLE VIII
NUMBERS OF ITERATIONS FOR ITERATIVE DESIGNS.

Design	Iterations
Iterative MDS	10
Iterative MGSNR	20
MSSNR (power method & iterative method)	20
Chow, BM-TEQ, RLS and RLS-LMS PTEQ	100
MMSE, MERRY, CNA, Nafie & Gatherer	500
SAM	2000

MMSE design [40], the Symmetric MMSE design [14], the MSSNR design [15], the Symmetric MSSNR design [30], [48], the MinISI design [13], the Symmetric MinISI design, the MinIBI design [17], the MDS design [19], and the dual-path TEQ [32]. Values are also given for efficient versions of these designs that make use of the techniques in [12], [16], [64]. The MMSE design is cheap because it involves a standard eigenvector problem rather than a generalized eigenvector problem. The dual-path TEQ is the most expensive because it computes the MMSE, MSSNR, MinISI, and MDS TEQs as part of its design process.

The right side of Fig. 3 gives the complexity of iterative and adaptive TEQ designs: the adaptive MMSE design [41], Chow's algorithm [43], Lopez-Valcarce's MDS implementation [18], the MSSNR design via the power method as in (2), the MSSNR design via iterating (6), MERRY [35], SAM [49], Nafie & Gatherer's design [33], and CNA [46]. The cheapest by far is the MDS design, since it only searches about 10 delays and each iteration is cheap. However, it does not explicitly consider the size of the cyclic prefix; rather, it simply minimizes the delay spread, which is a heuristic approach.

Fig. 4 gives the complexity of equalizer designs that explicitly attempt to maximize the bit rate. The TEQ designs considered are the iterative MGSNR implementation in [34] and the BM-TEQ [24]. The filter bank designs are the TEQ-FB [8], the PTEQ [9], and the RLS implementation of the PTEQ [36], [37]. The basic implementation of the PTEQ has the highest initialization complexity, although this can be fixed by using RLS. The TEQ-FB architecture has the highest complexity during data transmission.

VII. CONCLUSION

Equalizer design for multicarrier systems can be a computationally intensive procedure. We have surveyed the TEQ design literature for complexity reduction techniques, and categorized them in terms of whether or not (and how) the designs depart mathematically from the optimal design. We have tabulated the complexity requirements of the most popular algorithms for their original and reduced-complexity implementations. This tabulation was used to demonstrate the trade-offs between the bit rate performance and the complexity of these efficient implementations, for both synthetic and measured ADSL channels. For ADSL channels, most designs yield bit rates that only differ by about 10%, but computational requirements can vary by several orders of magnitude. By using complexity reduction techniques, the complexity of these designs can be reduced by several orders of magnitude for many of the designs.

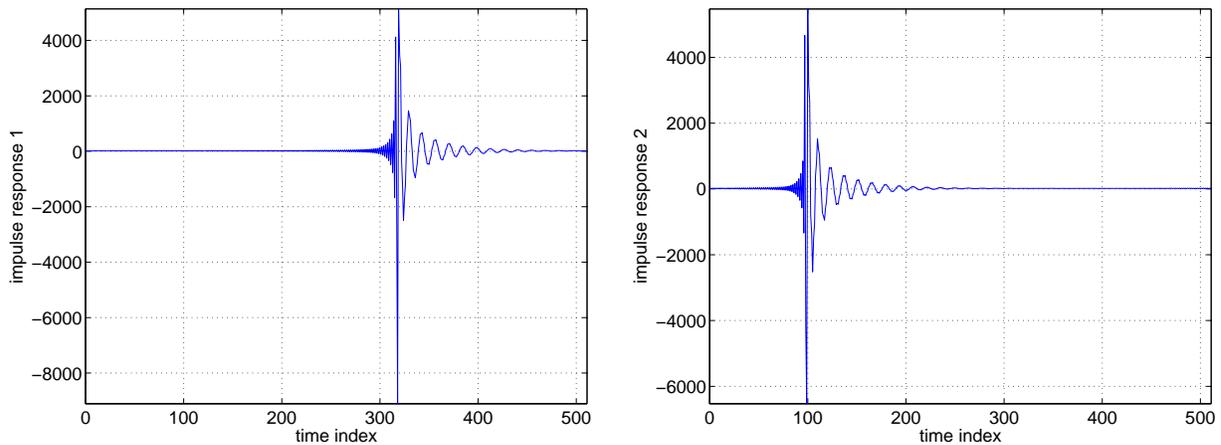


Fig. 1. Estimated channel impulse responses for measured signals 1 and 2 from the Applied Signal Technology data set.

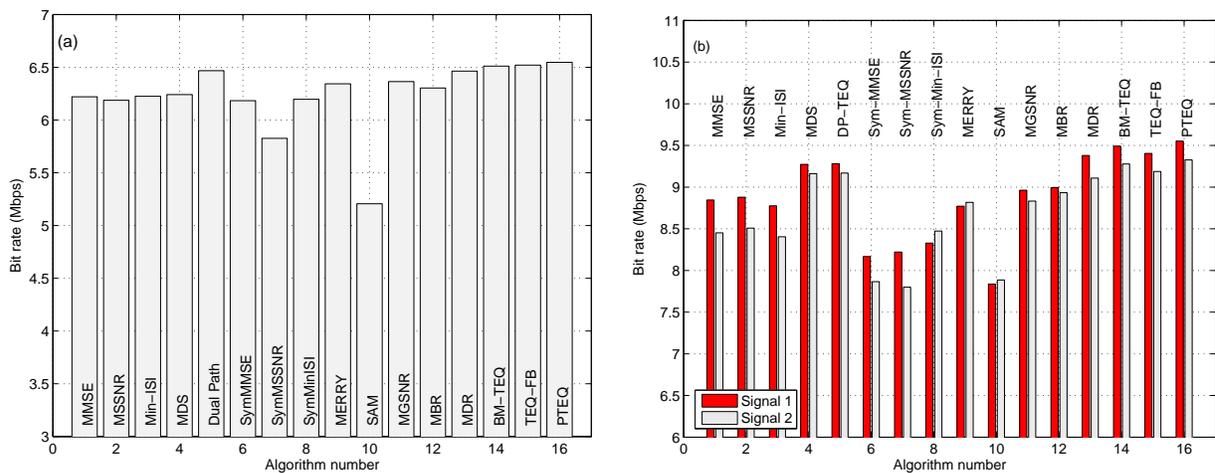


Fig. 2. (a) Bit rate comparison, averaged over the eight CSA test loops. (b) Bit rate for the two measured ADSL channels. The channel impulse responses are given in Fig. 1. This data was reported in [1] and is summarized here in order to provide a performance vs. complexity tradeoff in the next section.

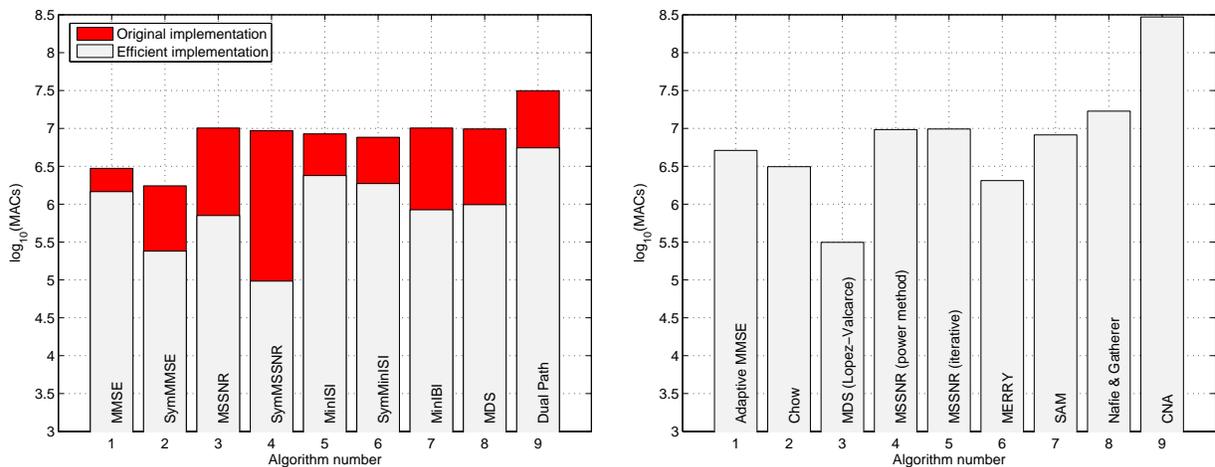


Fig. 3. Left: computational complexity of single Rayleigh quotient TEQ designs, for equalizer initialization. (Complexity during data transmission is the same for all single TEQ designs, hence it does not vary within this figure.) Values are given for the original implementation of each design as well as for the most computationally efficient implementation which is mathematically equivalent. Right: complexity of iterative single Rayleigh quotient designs.

Choosing the best equalizer design depends on your bit rate and complexity targets. The PTEQ has the highest achievable bit rate, yet its implementation cost is higher than that of most other designs. The iterative MDS TEQ has the lowest

implementation cost, with average performance. The dual-path TEQ has nearly optimal performance with an implementation cost that is slightly above average.

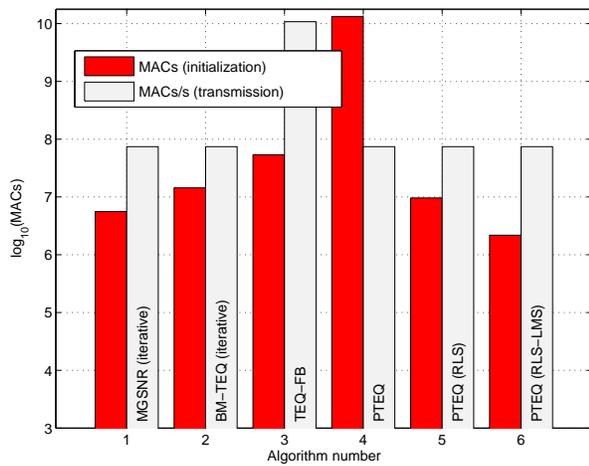


Fig. 4. Computational complexity of designs that explicitly attempt to maximize the bit rate. The complexity of initializing the equalizer is in MACs, and the complexity during data transmission is in MACs per second.

ACKNOWLEDGEMENTS

The authors wish to thank Applied Signal Technology (Sunnyvale, CA) for providing the measured ADSL data, and Mike Woodhall (formerly at Applied Signal Technology) for answering our many questions about the manner in which the ADSL data was measured.

REFERENCES

- [1] R. K. Martin, K. Vanbleu, M. Ding, G. Ysebaert, M. Milosevic, B. L. Evans, M. Moonen, and C. R. Johnson, Jr., "Unification and Evaluation of Equalization Structures and Design Algorithms for Discrete Multitone Modulation Systems," *IEEE Trans. on Signal Processing*, vol. 53, no. 10, pp. 3880–3894, Oct. 2005.
- [2] T. Starr, J.M. Cioffi, and P.J. Silvermann, *Understanding Digital Subscriber Line Technology*, Prentice Hall, 1999.
- [3] The Inst. of Electrical and Electronics Engineers, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 802.11a," 1999 Edition.
- [4] The European Telecomm. Standards Inst., "Broadband Radio Access Networks (BRAN); High Performance Radio Local Area Networks (HIPERLAN) Type 2; System Overview," ETR101 683 114, 1999.
- [5] The European Telecomm. Standards Inst., "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television," ETSI EN 300 744 V1.4.1, 2001 Edition.
- [6] The European Telecomm. Standards Inst., "Radio Broadcasting System, Digital Audio Broadcasting (DAB) to Mobile, Portable, and Fixed Receivers," ETS 300 401, 1995–1997.
- [7] D. H. Layer, "Digital Radio Takes to the Road," *IEEE Spectrum*, vol. 38, pp. 40–46, July 2001.
- [8] M. Milosevic, L. F. C. Pessoa, B. L. Evans, and R. Baldick, "DMT Bit Rate Maximization with Optimal Time Domain Equalizer Filter Bank Architecture," in *Proc. IEEE Asilomar Conf. on Signals, Systems, and Comp.*, Pacific Grove, CA, Nov. 2002, vol. 1, pp. 377–382.
- [9] 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.
- [10] 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.
- [11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1996.
- [12] 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.
- [13] 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.
- [14] R. K. Martin, M. Ding, B. L. Evans, and C. R. Johnson, Jr., "Efficient Channel Shortening Equalizer Design," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 13, pp. 1279–1290, Dec. 2003.
- [15] 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.
- [16] R. K. Martin and C. R. Johnson, Jr., "Fast Channel Shortening with Polynomial Weighting Functions," in *Proc. Conf. on Information Sciences and Systems*, Princeton, NJ, Mar. 2004.
- [17] S. Celebi, "Interblock Interference (IBI) Minimizing Time-Domain Equalizer (TEQ) for OFDM," *IEEE Signal Processing Letters*, vol. 10, no. 8, pp. 232–234, Aug. 2003.
- [18] R. López-Valcarce, "Minimum Delay Spread TEQ Design in Multi-carrier Systems," *IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 682–685, Aug. 2004.
- [19] R. Schur and J. Speidel, "An Efficient Equalization Method to Minimize Delay Spread in OFDM/DMT Systems," in *Proc. IEEE Int. Conf. on Comm.*, Helsinki, Finland, June 2001, vol. 5, pp. 1481–1485.
- [20] A. Tkachenko 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.
- [21] A. Tkachenko and P. P. Vaidyanathan, "A Low-Complexity Eigenfilter Design Method for Channel Shortening Equalizers for DMT Systems," *IEEE Trans. on Comm.*, vol. 51, no. 7, July 2003.
- [22] M. Ding, B. L. Evans, R. K. Martin, and C. R. Johnson, Jr., "Minimum Intersymbol Interference Methods for Time Domain Equalizer Design," in *Proc. IEEE Global Comm. Conf.*, San Francisco, CA, Dec. 2003, vol. 4, pp. 2146–2150.
- [23] I. Ilani, "Time Domain Equalizer for DMT Transceivers – A Geometric Approach," US patent no. 6341298, Jan. 2002.
- [24] K. Vanbleu, G. Ysebaert, G. Cuyper, M. Moonen, and K. Van Acker, "Bitrate Maximizing Time-Domain Equalizer Design for DMT-based Systems," *IEEE Trans. on Comm.*, vol. 52, no. 6, pp. 871–876, June 2004.
- [25] I. Lee, J. S. Chow, and J. M. Cioffi, "Performance Evaluation of a Fast Computation Algorithm for the DMT in High-speed Subscriber Loop," *IEEE Journal on Selected Areas in Comm.*, vol. 13, no. 9, pp. 1564–1570, Dec. 1995.
- [26] N. Lashkarian and S. Kiaei, "Fast Algorithm for Finite-length MMSE Equalizers with Application to Discrete Multitone Systems," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Mar. 1999, vol. 5, pp. 2753–2756.
- [27] R. M. Gray, "Toeplitz and Circulant Matrices: A Review," [Online]. Available: <http://ee.stanford.edu/~gray/toeplitz.pdf>.
- [28] R. K. Martin, C. R. Johnson, Jr., M. Ding, and B. L. Evans, "Exploiting Symmetry in Channel Shortening Equalizers," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc.*, Hong Kong SAR, China, Apr. 2003, vol. 5, pp. V–97–V–100.
- [29] R. K. Martin, C. R. Johnson, Jr., M. Ding, and B. L. Evans, "Infinite Length Results for Channel Shortening Equalizers," in *Proc. IV IEEE Sig. Proc. Workshop on Sig. Proc. Advances in Wireless Comm.*, Rome, Italy, June 2003, pp. 518–522.
- [30] C. Ribeiro, V. Silva, and P. S. R. Diniz, "Linear Phase Impulse Response Shortening for xDSL DMT Modems," in *IEEE Int. Telecom. Symp.*, Brazil, Sept. 2002, pp. 368–371.
- [31] J. S. Chow, J. C. Tu, and J. M. Cioffi, "A Discrete Multitone Transceiver System for HDSL Applications," *IEEE Journal on Selected Areas in Comm.*, vol. 9, no. 6, pp. 895–907, Aug. 1991.
- [32] M. Ding, A. J. Redfern, and B. L. Evans, "A Dual-path TEQ Structure for DMT-ADSL Systems," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, San Francisco, CA, May 2002, vol. 3, pp. 2573–2576.
- [33] M. Nafie and A. Gatherer, "Time-Domain Equalizer Training for ADSL," in *Proc. IEEE Int. Conf. on Comm.*, Montreal, Canada, June 1997, vol. 2, pp. 1085–1089.
- [34] N. Lashkarian and S. Kiaei, "Optimum Equalization of Multicarrier Systems: A Unified Geometric Approach," *IEEE Trans. on Comm.*, vol. 49, pp. 1762–1769, Oct. 2001.
- [35] 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.
- [36] K. Van Acker, G. Leus, M. Moonen, and T. Pollet, "RLS-Based Initialization for Per-Tone Equalizers in DMT Receivers," *IEEE Trans. on Comm.*, vol. 51, no. 6, pp. 885–889, June 2003.

- [37] G. Ysebaert, K. Vanbleu, G. Cuypers, M. Moonen, and T. Pollet, "Combined RLS-LMS Initialization for Per Tone Equalizers in DMT- Receivers," *IEEE Trans. on Signal Processing*, vol. 51, no. 7, pp. 1916–1927, July 2003.
- [38] P. I. Davis, N. J. Higham, and F. Tisseur, "Analysis of the Cholesky Method With Iterative Refinement For Solving the Symmetric Definite Generalized Eigenproblem," *SIAM Journal Matrix Anal. Appl.*, vol. 23, no. 2, pp. 472–493, 2001.
- [39] C. Chatterjee, V. P. Roychowdhury, J. Ramos, and M. D. Zoltowski, "Self-Organizing Algorithms for Generalized Eigen-Decomposition," *IEEE Trans. on Neural Networks*, vol. 8, no. 6, pp. 1518–1530, Nov. 1997.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] J. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "Equalizer Training Algorithms for Multicarrier Modulation Systems," in *Proc. IEEE Int. Conf. on Comm.*, Geneva, Switzerland, May 1993, pp. 761–765.
- [44] C. Yin and G. Yue, "Optimal Impulse Response Shortening for Discrete Multitone Transceivers," *IEE Electronics Letters*, vol. 34, pp. 35–36, Jan. 1998.
- [45] R. K. Martin, J. M. Walsh, and C. R. Johnson, Jr., "Low Complexity MIMO Blind, Adaptive Channel Shortening," *IEEE Trans. on Signal Processing*, vol. 53, no. 4, pp. 1324–1334, Apr. 2005.
- [46] M. de Courville, P. Duhamel, P. Madec, and J. Palicot, "Blind equalization of OFDM systems based on the minimization of a quadratic criterion," in *Proc. IEEE Int. Conf. on Comm.*, Dallas, TX, June 1996, pp. 1318–1321.
- [47] F. Romano and S. Barbarossa, "Non-Data Aided Adaptive Channel Shortening for Efficient Multi-Carrier Systems," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Hong Kong SAR, China, Apr. 2003, vol. 4, pp. 233–236.
- [48] R. K. Martin, M. Ding, B. L. Evans, and C. R. Johnson, Jr., "Infinite Length Results and Design Implications for Time-Domain Equalizers," *IEEE Trans. on Signal Processing*, vol. 52, no. 1, pp. 297–301, Jan. 2004.
- [49] J. Balakrishnan, R. K. Martin, and C. R. Johnson, Jr., "Blind, Adaptive Channel Shortening by Sum-squared Auto-correlation Minimization (SAM)," *IEEE Trans. on Signal Processing*, vol. 51, no. 12, pp. 3086–3093, Dec. 2003.
- [50] M. Milosevic, *Maximizing Data Rate of Discrete Multitone Systems using Time Domain Equalization Design*, Ph.D. thesis, The University of Texas at Austin, May 2003.
- [51] N. Al-Dhahir and J. Cioffi, "A Band-optimized Reduced-complexity Equalized Multicarrier Transceiver," *IEEE Trans. on Comm.*, vol. 45, pp. 948–956, Aug. 1997.
- [52] N. Al-Dhahir and J. M. Cioffi, "Optimum Finite-Length Equalization for Multicarrier Transceivers," in *Proc. IEEE Global Comm. Conf.*, San Francisco, CA, Nov. 1994, pp. 1884–1888.
- [53] N. Al-Dhahir and J. Cioffi, "The Combination of Finite-length Geometric Equalization and Bandwidth Optimization for Multicarrier Transceivers," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, May 1995, pp. 1201–1204.
- [54] N. Lashkarian and S. Kiaei, "Optimum Equalization of Multicarrier Systems Via Projection Onto Convex Set," in *IEEE Int. Conf. on Comm.*, Vancouver, June 1999, vol. 2, pp. 968–972.
- [55] G. Arslan, *Equalization Techniques for Multicarrier Modulation*, Ph.D. thesis, The Univ. of Texas at Austin, Dec. 2000.
- [56] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, New York, NY, 2 edition, 1992.
- [57] M. Milosevic, L. F. C. Pessoa, and B. L. Evans, "Simultaneous Multichannel Time Domain Equalizer Design Based On The Maximum Composite Shortening SNR," in *Proc. IEEE Asilomar Conf. on Signals, Systems, and Comp.*, Pacific Grove, CA, Nov. 2002, vol. 2, pp. 1895–1899.
- [58] Y. Almogly and O. Levin, "A Class of Fractional Programming Problems," *Operations Research*, vol. 19, pp. 57–67, 1971.
- [59] K. Vanbleu, G. Ysebaert, G. Cuypers, and M. Moonen, "On time-domain and frequency-domain MMSE-based TEQ design for DMT transmission," accepted for publication in *IEEE Trans. Signal Processing*, Sept. 2004, available at <ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/vanbleu/reports/04-02.pdf>.
- [60] J. M. Cioffi, "A Multicarrier Primer," [Online]. Available: <http://www-isl.stanford.edu/people/cioffi/pdf/multicarrier.pdf>.
- [61] D. G. Messerschmitt and A. Salvekar, "Linemod software for transmission line analysis," Stanford University. [Online]. Available: <http://www.stanford.edu/~cioffi/linemod/linemod.html>.
- [62] G. Arslan, M. Ding, B. Lu, M. Milosevic, Z. Shen, and B. L. Evans, "MATLAB DMTEQ Toolbox 3.1," The University of Texas at Austin, May 10, 2003. [Online.] Available: <http://www.ece.utexas.edu/~bevans/projects/adsl/dmteq/dmteq.html>.
- [63] International Telecommunications Union, "Asymmetrical Digital Subscriber Line (ADSL) Transceivers," ITU, 1999.
- [64] R. K. Martin and C. R. Johnson, Jr., "Efficient Channel Shortening Equalizer Computation," in *Proc. Conf. on Information Sciences and Systems*, Baltimore, MD, Mar. 2003.