Efficient FFT-based Circulant Embedding Method and Subspace Algorithms for Wideband Single-carrier Equalization

Raymond Guan and Thomas Strohmer
Department of Mathematics, University of California,
Davis, CA 95616-8633, USA.
Email: rguan, strohmer@math.ucdavis.edu

***Abstract*— In single-carrier transmission systems, wider bandwidth allows higher data rate by transmitting narrower pulses. For wireless applications, however, this means that the effective channel response is longer and the number of significant taps increases. This results in higher computational burden at the receiver. In this paper, we first review some well-known equalization methods and their shortcomings. Then we will propose the use of the circulant embedding method and the CG algorithm as efficient equalizers that are specifically well suited in dealing with long delay spread channels. These methods take advantage of the low computational complexity of the FFT algorithm, resulting in the same overall computational complexity of $N\log(N)$. Furthermore, when the CG algorithm is used correctly, it may perform better than the MMSE equalizer at a much lower cost.**

I. Introduction

In recent years, driven by military application as well as consumer products, the demand for high-data rate wireless communication systems has been increasing at a dazzling pace. This leads to wider bandwidth communication systems such as ultra wideband system with bandwidth in the order of hundreds megahertz. While the use of multiple antennas is known to increase capacity, it comes at a significant increase in hardware costs. Thus in practice, the best way to achieve higher data rates in wireless communications is still to increase the bandwidth of the wireless channel, since the channel capacity grows linearly with the channel bandwidth. However, simply increasing the channel bandwidth is only half the story – there are problems associated with this approach. For instance, channel information (CI) is harder to obtain for wideband channels and multipath propagation plays an increasingly dominant role which makes equalization a challenging task.

There exist essentially two competing wireless transmission schemes: single-carrier and multi-carrier transmission [1], [2]. A prominent example for a multi-carrier communication system is Orthogonal Frequency Division Multiplexing (OFDM). Comparing the two schemes, OFDM [3] has the advantage of low receiver complexity while achieving ML decoding. The disadvantage is paying a higher price for the transmitter with a wide dynamic range power amplifier to handle the peak-to-average power ratio (PAPR) problem that is intrinsic to the OFDM scheme. Also, the ML performance is achieved by cyclic prefix (or postfix) which effectively reducing data rate by transmitting redundant information. If reducing data rate is tolerable, single-carrier with zero-padding or cyclic extension [4] may achieve ML performance as well with frequency domain equalization [5] while avoiding the use of expensive power amplifiers. Furthermore, single-carrier systems are less sensitive to channel estimation error and to carrier frequency offset.

This work was partially supported by NSF DMS grant No. 0511461 and NSF VIGRE grant No. DMS-0135345.

In this paper, our interest is wideband real-time high data rate applications with single-carrier systems. Due to the long delay spread encountered in wideband communication systems, there is a severe penalty in terms of loss of data rate when employing single carrier block transmission as is done for instance when using guard intervals or a cyclic prefix. On the other hand the complexity of equalization can become significant for non-block transmission schemes, which raises the problem of how to construct numerically efficient equalizers for this case. Therefore in this paper we focus on designing efficient equalization algorithms for high-data rate single carrier non-block transmissions. The proposed algorithms are sufficiently fast to be used for (near) real-time applications such as video streaming.

II. Problem Formulation

Continuous and discrete models are introduced in this section while establishing the notation that will be used throughout the rest of this paper.

A. Communication Model

To transmit the set of discrete information symbols $\{s_k\}_{k=-\infty}^{\infty}$, it is first converted to a continuous-time signal by a sum of weighted copies of a finite duration pulse shaping function $p(t)$ via

$$c(t) = \sum_k s_k p(t - kT). \tag{1}$$

Let $h_c(t)$ represent the impulse response of the communication channel. Throughout the paper, exact CI at the receiver is assumed. At the receiver, the continuous-time received signal, $r(t) = (h_c * c)(t)$ is filtered with a finite duration matched filter $q(t)$ and then sampled to produce a digital signal so that it may further processes digitally with DSP chips to reduce cost.

Mathematically, define $p_k = p(t-kT)$, we are discretizing the continuous-time function

$$b(t) = (r * q)(t) = \sum_k s_k (h_c * p_k * q)(t) \tag{2}$$

to, depending on the sampling rate, produce discrete sequences $b(t_{i,j})$ and $h_d(i,j) = (h_c * p_k * q)(t_{ij})$ where $t_{i,j}$ is the sampling time that takes on values from the set $\{..., -2T/a - jP, -T/a - jP, 0 - jP, T/a - jP, ..., iT/a - jP, ...\}$. We have in general three cases: Nyquist sampling (if $a = 1, P = 0$), Q-fold integer over-sampling (if $a = 1$, $P = T/Q$ and $j = 0, 1, ..., (Q-1)$ for some natural number Q) and fractional sampling with a sampling period of P (if $a > 0$ is not an integer, $P = T/(aQ)$ and $j = 0, 1, ..., (Q-1)$ for some natural number Q).

B. Algebraic Formulation

The following notational conventions are used throughout the rest of the paper: (1) capital italic letters such as H denote finite dimensional matrices. (2) Lower case italic letters such as x or h are finite dimensional vectors. (3) For matrices, subscript like x in H_x refers to the x-th row of the matrix H if x is a number, otherwise it is a naming of the matrix, and H^j is the j-th column of H. Similarly, h_i is either the i-th entry in the vector h or an element from the set $\{h_i\}$, it should be clear from the context. (4) Superscript with parenthesis like $H^{(k)}$ (or $x^{(k)}$) refers to different matrices (or vectors) for different integer k. (5) Bold letter such as $\mathbf{H}$ is the infinite dimensional counterparts of H and other conventions, such as subscript or superscript, for finite dimensional matrix or vector hold for the infinite dimensional one as well. (6) For random variables in matrix or vector form, bold slanted letters such as $\boldsymbol{n}$ are used.

To develop numerical algorithms, it is often convenient to formulate the problem in terms of algebraic equations. Discretizing $b(t)$ enables us to do just that. For the Nyquist sampling case, we have the bi-infinite dimensional system

$$\mathbf{H}x = b \tag{3}$$

where

$$\mathbf{H} = \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & h_L & h_{L-1} & ... & h_1 & 0 & ... & 0 & \ddots \\ \ddots & 0 & h_L & h_{L-1} & ... & h_1 & \ddots & \vdots & \ddots \\ \ddots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \ddots \\ \ddots & 0 & ... & 0 & h_L & h_{L-1} & ... & h_1 & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

is a bi-infinite band Toeplitz matrix formed by the length L discrete equivalent channel $h = [h_d(1,0), h_d(2,0), ..., h_d(L,0)]^T$ and $\mathbf{x} = [..., s_{-1}, s_0, s_1, ...]^T$. Since the above linear system is infinite dimensional, we need to trunacte it before we can deal with it numerically. However, it remains rank deficient regardless of how large the system is in finite dimensions. In the absence of noise, a sampling rate higher than the Nyquist rate, i.e., with $a > 1$, is required for perfect (FIR) equalization [6]. For practical purposes, we consider integer oversampling only in this paper. For the case of Q-fold oversampling, the finite channel matrix H has block Toeplitz structure with a block size of Q rows:

$$H_{(Q)} = \begin{bmatrix} h_{(L)} & h_{(L-1)} & ... & h_{(1)} & 0 & ... & 0 \\ 0 & h_{(L)} & h_{(L-1)} & ... & h_{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & ... & 0 & h_{(L)} & h_{(L-1)} & ... & h_{(1)} \end{bmatrix} \tag{4}$$

where $h_{(l)} = [h_d(l,0), h(l,1), ..., h(l, Q-1)]^T$. For the rest of the paper we tacitly assume two-fold [1] oversampling, drop the $H_{(2)}$ notation and simply use H.

To be more accurate, a two-fold oversampled single-carrier system that also models noise and interference with a random vector $\boldsymbol{n}$ is

$$Hx + \boldsymbol{n} = b. \tag{5}$$

For simplicity, throughout the rest of the paper, the following assumption are made: (1) $x_k = 1$ or -1, i.e., we consider BPSK modulation scheme. (2) When the transmitting symbols are considered as random variables, we consider them as independent and identically distributed (i.i.d) with zero mean and equally likely. (3) The entries of the noise vector $\boldsymbol{n}$ is normally i.i.d. with zero mean and variance σ^2.

Incidentally, the formulation for the integer oversampling case is the same as for a single-input-multiple-output (SIMO) communication system [7]. For the Q-fold oversampled case,

$$h^{(m)} = [h_d(1, m-1), h_d(2, m-1), ..., h_d(L, m-1)]^T \tag{6}$$

where m = 1, 2, ..., $(Q-1)$ or Q is the equivalent m-th receiver discrete channel. Therefore, the results in this paper are applicable directly to SIMO single carrier transmission systems.

III. Single Carrier Equalizers

Any left matrix inverse of H, denoted by $H^\dagger$, may be used as an equalizer for the single-carrier system (5). And any row from the matrix $H^\dagger$, termed an *FIR equalizer*, may be used to decode symbols one at a time by taking inner product of the time synchronized received vector and the FIR equalizer. Particulary for wideband wireless single-carrier system, the matter of which FIR equalizer to choose among the rows of $H^\dagger$ is relevant; we will come back to this point in section V-B. In this section we will discuss some well-known and some not so well-known equalizers and point out some of their properties.

A. State of the Art Equalizers

Various types of equalizers are known, notably linear equalizers such as zero-forcing (ZF) equalizer and minimum-mean

[1]However we note that the results hold true with trivial modifications for any integer oversampling rate.

square error (MMSE) equalizer and they are respectively

$$H_{ZF}^{\dagger} = (H^*H)^{-1}H^* \tag{7}$$

and

$$H_{MMSE}^{\dagger} = (H^*H + \sigma^2 I)^{-1}H^*. \tag{8}$$

Non-linear equalizers like decision feedback equalizers (DEFs) are effective and simple to implement when the channel delay spread is short or sparse with few significant taps. However, for channels with many significant taps, error propagation limits the use of them. In terms of performance, the best equalizer is the maximum-likelihood (ML) equalizer given by

$$\underset{x\in\mathbb{S}}{\operatorname{argmin}}\, ||b - Hx||, \tag{9}$$

where $\mathbb{S}$ is the set of all possible solutions. In general, ML is too costly or even impossible to implement in practice due to its high computational complexity.

B. Practical Solutions

The main disadvantage of the MMSE solution is the cubic computational complexity in inverting the matrix $(H^*H) + \sigma^2 I$. For the Nyquist rate sampling case, sub-optimal solution via the efficient FFT are known [2] and the basic idea is to approximate the baud rate Toeplitz matrix H_b by a circulant matrix C_b. In matrix notation, the solution is

$$H_b^{\dagger} = (C_b^* C_b + \sigma^2 I)^{-1} C_b^* \tag{10}$$

However, in the oversampling case, the structure of H is block Toeplitz and the FFT approach in [2] does not apply. For the case of overampling, we introduce the class of *circulant-embedding* (CE) *equalizers*. These are the ideas of approximating not H but (H^*H) by a circulant matrix C, thus the cost of matrix inversion may still be in the order of $N\log N$ due to the efficient FFT algorithm. Variances of the CE equalizers deriving from approximating the ZF and MMSE equalizers are the *zero-forcing circulant-embedding* (ZFCE) *equalizer*

$$H_{ZFCE}^{\dagger} = C^{-1}H^* \tag{11}$$

and the *regularized circulant-embedding* (RCE) *equalizer*

$$H_{RCE}^{\dagger} = (C + \sigma^2 I)^{-1}H^* \tag{12}$$

Now, before we get to the convergence properties of the solution by the ZFCE equalizer to the bi-infinite model ZF solution, the circulant matrix construction needs to be addressed since the solution will depend on it. Recalling that H is a bi-infinite block-Toeplitz matrix, it is interesting to analyze the structure of the finite model matrix H^*H which can be represented as

$$(H^*H) = \begin{bmatrix} * & * & 0 \\ * & T_{H^*H} & * \\ 0 & * & * \end{bmatrix},$$

where T_{H^*H} is a $(2L-1)$ band hermitian positive definite (PD) Toeplitz matrix. The positive-definiteness of T_{H^*H} follows from the Cauchy interlacing theorem and the fact that $\mathbf{H}^*\mathbf{H}$ is PD [8]. From this observation, we can conclude that, although both $\mathbf{Hx} = \mathbf{b}$ and $\mathbf{H}^*\mathbf{Hx} = \mathbf{H}^*\mathbf{b}$ are bi-infinite linear system, we may truncate the systems into $Hx = b$ and $T_{H^*H}x = b_T$ where b_T is the appropriate truncation of H^*b so that equality holds. Thus, instead of comparing the infinite model directly, we may compare the finite model solution. Also, having a finite Toeplitz matrix T_{H^*H} to work with we are in a better position to discuss the construction of the circulant matrix C. There are various ways to do this, in this paper we consider the *embedding method*. That is, C is constructed first by embedding T_{H^*H} into a larger matrix and then modifying only the lower-left and upper-right corner entries appropriately so that it becomes a circulant matrix [9]. Therefore we may write $C = T_{H^*H} + W$ for some W that depends on T_{H^*H}. Although H is not Toeplitz, T_{H^*H} is and to compute its row or (column) which is needed to construct C, FFT may still apply and the overall computational complexity of ZFCE or RCE equalizer remains in the order of $N\log N$.

The following theorem shows that, in the absence of noise, the solution of the ZFCE equalizer converges exponentially fast to the solution of the bi-infinite ZF equalizer. A detailed proof of this result (as well as other theoretical results) will be given in the journal version of this paper. To be concise, for $N \in \mathbb{N}$ and $\mathbf{y} \in l^2(\mathbb{Z})$ or $l^2(2N+1)$ let's define the orthogonal projection operator $P_{(N)}$ by

$$P_{(N)}\mathbf{y} = [\cdots\ 0\ y_{-N}\ \cdots\ y_N\ 0\ \cdots]^T \tag{13}$$

and the truncation operator $T_{(N)}\mathbf{y} = [\ y_{-N}\ \cdots\ y_N]^T$.

Theorem 3.1: Suppose $\mathbf{T}$ is a band bi-infinite hermitian PD Toeplitz matrix. Let $\mathbf{x} \in l^2(\mathbb{Z})$, $\mathbf{y} = \mathbf{Tx}$, $\mathbf{T}^{(N)} = P_{(N)}\mathbf{T}P_{(N)}$ and $C^{(N)}$ is a size $(2N+1)$ circulant matrix constructed from the $-N$-th to the N-th rows and columns of $\mathbf{T}^{(N)}$ by the embedding method. Define approximate solution $\mathbf{x}^{(N)} = P_{(N)}\left(C^{(N)}\right)^{-1}T_{(N)}\mathbf{y}$. Then there exists N such that for all $M > N$

$$||P_{(M)}\mathbf{x} - \mathbf{x}^{(M)}|| \le c_1 e^{-c_2 M} \tag{14}$$

is true for some constant $c_1, c_2 > 0$ that are independent of M and depend only on the condition number and the bandwidth of the matrix $\mathbf{T}$.

IV. Subspace Methods in Theory

The solution to the noisy linear system is well established in the theory of regularization. The methods may be categorized into direct and iterative schemes. Direct methods cost as much as inverting a matrix so we will only focus on the efficient iterative methods that are fast enough to be used in real time. In particular, we will consider Krylov subspace methods based on the conjugate-gradient (CG) algorithm.

A. The CG Algorithm

There are various algorithms that are considered as Krylov subspace methods; the first of these is the by now classical CG algorithm, cf. [10] for a detailed discussion of CG. Due to its many nice properties, CG is also a natural candidate as efficient algorithm for equalization for single-carrier communication systems. While the CG method has become a standard tool in numerical analysis, its practical use in the context of equalization is not straightforward at all and requires some careful modifications and adaptations. And this is the topic for most of the rest of this paper.

The CG method, applying to a symmetric positive definite (SPD) linear system $Ax = y$, produces a sequence of iterates $x^{(k)}$, $k = 1, 2, \ldots$, which converge monotonically to the true solution in the noise-free case, if the matrix A is invertible, in finite number of iterations. The computational complexity of the algorithm is determined by a matrix vector multiplication in each iteration. When the matrix is not SPD, as is the case in (5), the CG algorithm can be applied to the associated normal equations and there are variations of CG like CGNE (CG algorithm applied to normal equations) that avoid the explicit computation of the matrix product H^*H; instead it only costs another matrix vector multiplication with H^*y for some vector y in each iteration[10]. Throughout the rest of the paper, we will refer both CG and CGNE algorithm as CG algorithm; it should be clear from the corresponding linear systems which one is meant.

B. Stopping Criteria

The CG algorithm is widely used today due to its nice convergent properties [10]. If the eigenvalues of the matrix A are clustered or bounded away from zero, it converges fast. However the convergence properties are more delicate in the presence of noise. As mentioned, in the noise-free case CG produces a sequence $x^{(k)}$ that converges to the true solution x. Unfortunately, in the presence of noise the monotone convergence of this sequence to the true solution is no longer guaranteed, this fact is also true for CGNE [10]. The iterates $x^{(k)}$ may first converge, but later diverge from the true solution. A rule is needed to determine which one in the sequence is the best choice. Furthermore, the evaluation of such a rule must also be efficient so that it may be applied in real time applications. One such efficient rule is the so-called *Discrepancy principle* [11]. This principle depends on a parameter ϵ and the solution index k is the smallest integer such that

$$||r^{(k)}||_2 \leq \epsilon \tag{15}$$

is satisfied. The advantage of this rule is that it is simple to compute. However, it requires to chose a sensitive parameter ϵ. This parameter should be a function of the noise power for example. We found that ϵ as a simple linear function of noise power is not a robust SC to use across a wide range of signal-to-noise ratio (SNR) values. Figure 1 demonstrates the difficulty in using this principle as well as the importance of a good stopping rule. Another well-known stopping criterion, the L-curve [12], has also turned out to be unreliable in the context of equalization.

C. Two Different Uses of the CG Algorithms

Before discussing optimal stopping criterion for the CG algorithm, we want to point out that by properly formulating the problems, the CG algorithm can be used to compute both the matrix system and the FIR solution and different solutions require a different SCs.

Applying the CG algorithm in the straightforward manner, we may use it to solve x in the noisy linear system (5) directly.

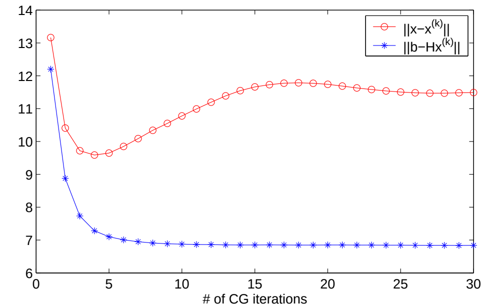

Fig. 1. Actual and residual error.

On the other hand, we may first solve for an FIR equalizer g and then find the desired solution via decoding ($g^T b$). The latter approach motivates the use of the CG algorithm to solve the following linear equation

$$H^T g = a^{(k)} \tag{16}$$

where $a^{(k)}$ is a vector consisting only of zeros entries except at the k-th entry, which is equal to 1.

D. SC for the CG Algorithms

Given full knowledge of the channel information H, for any FIR equalizer g, the BER may be analyzed. Suppose, without lost of generality, x_1 is the desired decoding symbol. Define a vector $d = g^T H$ and assuming $d_1 \neq 0$, otherwise, the BER for x_1 is 0.5. By normalizing the vector g as $g^{(1)} = (1/d_1)g$, we may expand

$$\left(g^{(1)}\right)^T b = x_1 + (1/d_1)\sum_{k=2}^{N} d_k x_k + \sum_{k=1}^{M} g_k^{(1)} \boldsymbol{n}_k. \tag{17}$$

Thus, the BER may be computed by analyzing the total noise distribution that consists of interference noise due to the neighbor symbols x_2, ... x_N (the second term in the above equation) and the equalized noise (the third term). The interference noise may be modeled with a sum of Bernoulli distributions $\mathbf{x}_1$, ..., $\mathbf{x}_{N-1}$ that have zero-mean and variance $|d_2/d_1|^2$, ..., $|d_{N-1}/d_1|^2$. The equalized noise is a sum of normal distribution with zero mean and variance $(|g_1|\sigma)^2$, ... $(|g_M|\sigma)^2$, thus it is still normally distributed with zero mean and variance $(||g||\sigma)^2$. The total noise is then modeled as

$$\sum_{k=1}^{N-1} \mathbf{x}_k + ||g||\sigma\overline{\boldsymbol{n}}, \tag{18}$$

where $\overline{\boldsymbol{n}}$ is normally distributed RV with zero-mean and unit variance. Therefore an optimal SC is possible, at least in theory, for the CG algorithm when it is used to compute FIR equalizers. The actual computational complexity is 2^N, due to mainly the interference noise. For large N, the exponential complexity renders the optimal SC impractical. A suboptimal SC based on second order statistics, discussed in the next section, is much cheaper to compute and thus may be used in practice.

For direct computation of x by the CG algorithm, from the ML solution, it is reasonable to minimize the metric

$$||b - Hx^{(k)}||^2 \tag{19}$$

as a SC. In practice however, the above expression requires modification, which is also discussed in the next section.

V. Subspace Methods in Practice

Applying the CG algorithm in practice requires some additional consideration such as finding a low cost SC with acceptable performance and implementing the same algorithm more efficiently. These are the topics of this section.

A. MMSE-SC for FIR Equalizers

For the application of wireless communications, we know that the solution x belongs to a finite alphabet set and we should take advantage of this information when developing SCs. Now, consider the case that the entries of $\boldsymbol{x}$ be uncorrelated zero-mean unit energy RV with Bernoulli distribution again. Assume the transmitted symbols and the receiver noise are uncorrelated as well, i.e., $\mathcal{E}\{\boldsymbol{x}_i\boldsymbol{n}_j\} = 0, \forall i,j \in \mathbb{Z}$. When the CG algorithm is applied to the linear system in (16), a sequence of approximate FIR equalizers, $\{g^{(k)}\}_{k\in\mathbb{N}}$, is produced. For each of these solutions, say $g^{(n)}$, written as G_n^T for notational clarity, we may analyze them, similar to (17), by separating the signal and noise components as

$$G_n b = (G_nH)_k\boldsymbol{x}_k + \sum_{i\neq k}(G_nH)_i\boldsymbol{x}_i + G_n\boldsymbol{n}. \tag{20}$$

By formulation, we know $(G_nH)_k$ is supposed to converge to 1. We may initialize $g^{(0)}$ to be the matched filter, i.e., the conjugate of the k-th row of H^T. Thus, it is reasonable to make the following approximation

$$(G_nH)_k \approx 1. \tag{21}$$

Therefore, minimizing the (total) noise power may be used as a SC. This suggests the following noise power metric

$$\begin{aligned} P^{(1)}_{FIR-SC} &= \mathcal{E}\left\{\left|\left|\sum_{i\neq k}(G_nH)_i\boldsymbol{x}_i + G_n\boldsymbol{n}\right|\right|^2\right\} \\ &= ||G_nH||^2 - |(G_nH)_k|^2 + (||G_n||\sigma)^2. \end{aligned} \tag{22}$$

By the approximation in (21) and the definition of the residual, (22) may be approximated by

$$P^{(2)}_{FIR-SC} = ||a^{(k)} - r^{(n)}||^2 + (||G_n||\sigma)^2 - 1. \tag{23}$$

Computationally, this may still be further simplified since the value $||r^{(n)}||^2$ is known in each iteration and $a^{(k)}$ is non zero only at the k-th entry which is one. An efficient SC is then

$$P^{(3)}_{FIR-SC} = ||r^{(n)}||^2 + (||G_n||\sigma)^2. \tag{24}$$

Consider the case that the approximation in (21) is not good due to for example taking the initial approximation to be the zero vector, which is often done in practice so that the residual converges monotonically. In this case, we want to maximize signal minus noise power. This leads to another SC rule

$$\begin{aligned} P^{(4)}_{FIR-SC} &= \mathcal{E}\left\{|(G_nH)_k\boldsymbol{x}_k|^2\right\} \\ &\quad - \mathcal{E}\left\{\left|\left|\sum_{i\neq k}(G_nH)_i\boldsymbol{x}_i + G_n\boldsymbol{n}\right|\right|^2\right\} \\ &= 2|(G_nH)_k|^2 - ||G_nH||^2 - (||G_n||\sigma)^2 \end{aligned} \tag{25}$$

The above SCs are different due to the quality of approximation of $(G_nH)_k$ by one. When normalizing the FIR equalizer G_n such that

$$(G_nH)_k = 1 \tag{26}$$

we found that the the expected error of the the k-th entry of x is

$$\mathcal{E}\left\{|\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k|^2\right\} = (||G_nH||^2 - 1) + (||G_n||\sigma)^2. \tag{27}$$

The proof is a straight forward computation. Note, SCs (22) and (25) will give a same solution as SC (27) when (26) is true. Therefore, if the SNR is known, the above mean-square error (MSE) expression may be used as a practical SC. In fact, we found that it is the best SCs among the ones presented in this paper for the CG algorithm for computing FIR equalizers. Such SC actually determines the MSEs from a set of FIR equalizers that are computed by the CG algorithm, thus it is the minimum mean-square error SC or MMSE-SC.

From the two terms in the above metric we may see that the 'best' equalizer, in the presence of noise is the one that balances between equalizing the channel (the interference in the first term) and not amplifying the noise too much (the last term). The above SC is only suboptimal since it is possible to have two different (total) noise distributions with the one corresponding to the smaller BER may actually have a larger variance.

One last point to be addressed to complete the discussion on FIR equalizer computation with the CG algorithm is that which symbol (entry) in x is the best to be equalized. That is, what value should we choose for k in $a^{(k)}$ from equation (16). We will provide a partial answer to this question in the next subsection since it is related to the observation from our experiment that the BER for the symbols in the middle entries of x is much better than the ones on the boundary.

B. Matrix Solution

As mentioned in section III, the ML metric may be used as a SC for the CG algorithm when computing x directly. In practice, due to the finite model approximation to the infinite model, the truncation of the finite model leads to unequal MSE for different entries in the solution vector x. The full analysis of this BER behavior includes the effect of the noise and it is too involved. For this reason, we will only analyze the loss of signal energy due to truncation in this paper. In the absence of noise, let us consider any one particular transmission of the signal x. In the finite model formulation, refer to Eq. (4), the energy in the first transmitted symbol x_1 is spread out by the channel by the taps $h_L^{(1)}$ and $h_L^{(2)}$ and arrives at the receiver, captured by the samples b_1 and b_2. To capture the maximum energy of this symbol, the match filter $g_{(1)} = [h_L^{(1)}, h_L^{(2)}, 0, \cdots]^*$ is applied which gives

$$g_{(1)}^T b = \left(|h_L^{(1)}|^2 + |h_L^{(2)}|^2\right)x_1 + \sum_{k=2}^{L} f_k x_k. \tag{28}$$

Similarly, the analysis for the symbol, WLOG, x_L is

$$g_{(L)}^T b = \left(||h^{(1)}||^2 + ||h^{(2)}||^2\right)x_L + \sum_{k\neq L}\tilde{f}_k x_k. \tag{29}$$

The first terms in the right hand side of equations (28) and (29) represent the equalized signal while the second terms are interference noises due to neighbor symbols which we are ignoring. Therefore, we see that unless the energy of each sub-channel is well concentrated in the last taps, the equalized signal energy for the entries in the middle of the vector x, e.g., x_L is larger than the boundary entries such as x_1.

Now, we are at a better position to answer the question from the last sub-section – which symbol in x should we equalize? For the matched filter case, we see that it is possible to extract the maximum signal energy for x_L, x_{L+1}, ..., x_{N-L+1}. Thus, equalizing any one of this symbol is equally good. Taking noise and interference into consideration, the matter is not so straight forward. Equation (27) offers an elegant but expensive solution if H is large. For a general guideline, we recommend equalizing exactly the middle entries, though, this is not always the best choice.

Applying the CG algorithm to the noisy linear system (5), with enough number of iterations, the algorithm will produce the ZF solution. However, this is not a good solution due to the amplification of noise. There are two different ways to deal with the noise enhancement problem and produce a regularized solution similar to the MMSE equalizer. One way to do this is stopping the CG algorithm earlier with a good stopping rule. For real time application, the process of solution selection should be done automatically at the same time would stop the algorithm as soon as the 'best' solution is produced. The above analysis suggests not to used the boundary entries in the solution $x^{(k)}$ for the SC since they are naturally worst solution than the ones in the middle. A 'safer' way to avoid too much noise amplification, however, is by explicitly regularizing the linear system with a regularization parameter η, i.e., we are applying the CG algorithm to the regularized linear system

$$(H + \eta^2 I)x = b. \tag{30}$$

When η is chosen to be equal to the noise power σ, the algorithm converges to the MMSE solution instead of the ZF one. An even better solution, we found, is that in each step the regularization parameter is chosen from a set of values and it may vary from one CG iteration to the next. By implementing the (modified) CG algorithm correctly, it is still a very low cost equalizer. Notice the similarity in the two different uses of the CG (or CGNE) algorithm, section IV-C, they all require the matrix vector multiplications Tu and T^*v for some block Toeplitz matrix T and some vector u and v. In the theory regularization, it is well-known that these may be computed efficiently with the FFT algorithm [9]. Thus, the efficient FFT algorithm reduces the over all CG algorithm computational complexity to $N \log(N)$, which is a still low even when N is large. An efficient implementation of the CG algorithm for the regularized linear system (30) may be found in [13].

VI. Simulation Results

The BER simulations of a 64-tap length sub-channel are shown in Fig. 2. We are comparing the performance of the MMSE, RCE, GC-FIR and CG-direct methods. The MMSE and RCE curves are overlapped due to good approximation of the CE method. The CG-FIR performance is slightly worst than the MMSE one. Linear equalizers such as the MMSE equalizer use only the channel and noise power information. In contrast, when solving x directly, the CG algorithm make uses all the information available, including the received vector b, it is able to outperform even the MMSE solution. From our simulations with various channels, we observed that the number of CG iterations needed to compute the FIR equalizer or approximate x directly is small – two to seven iterations across a wide range of SNR values. Many researchers also observe this rapid convergent behaviour of the CG algorithm that is due to its (still not fully understood) inherent regularization property that the algorithm iterates extract first the dominant directions/subspace(s) of the matrix, and only later those directions that are associated with the samll singular values [12]. Due to limited space constraint, other simulation results will be presented in the journal version of this paper.

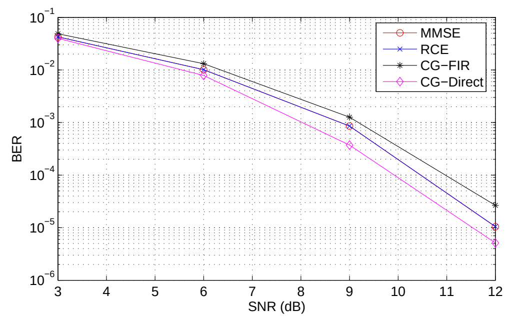

Fig. 2. Equalizer performance comparison of an 64-tap sub-channel.

References

[1] T. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 2002.

[2] G. B. Giannakis, Y. Hua, P. Stoica, and L. Tong, *Signal Processing Advances in Wireless and Mobile Communications - Volume I, Trends in Channel Estimation and Equalization*. Prentice-Hall, September 2000.

[3] A. Scaglione, G. B. Giannakis, and S. Barbarossa, "Redundant filterbank precoders and equalizers. parts I and II," *IEEE Transactions on Signal Processing*, vol. 47, pp. 1988–2022, July 1999.

[4] L. Deneire, B. Gyselinckx, and M. Engels, "Training sequence vs. cyclic prefix : a new look on single carrier communication," *IEEE Communication Letters*, vol. 5, no. 7, pp. 292–294, 2001.

[5] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Communications Magazine*, pp. 58–66, April 2002.

[6] G. Harikumar and Y. Bresler, "FIR perfect signal reconstruction from multiple convolutions: Minimum deconvolver orders," *IEEE Transactions on Signal Processing*, vol. 46, JANUARY 1998.

[7] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge University Press, 2003.

[8] R. Horn and C. Johnson, *Matrix analysis*. Cambridge: Cambridge University Press, 1990.

[9] R. H. Chan and M. K. Ng, "Conjugate gradient methods for Toeplitz systems," *SIAM Review*, vol. 38, pp. 427–482, 1996.

[10] M. Hanke, *Conjugate gradient type methods for ill-posed problems*. Harlow: Longman Scientific & Technical, 1995.

[11] S. Pereverzev and E. Schock, "Morozov's discrepancy principle for tikhonov regularization of severely ill-posed problems in finite-dimensional subspaces.."

[12] P. C. Hansen, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. Society for Industrial and Applied Mathematics, 1999.

[13] A. Frommer and P. Maass, "Fast CG-based methods for Tikhonov–Phillips regularization," *SIAM Journal on Scientific Computing*, vol. 20, no. 5, pp. 1831–1850, 1999.