Low-Complexity Channel-Estimate Based Adaptive Linear Equalizer

Teyan Chen, Yuriy V. Zakharov, *Senior Member, IEEE*, and Chunshan Liu,

***Abstract*—In this paper, we propose a low-complexity channel-estimate based adaptive linear equalizer. The equalizer exploits coordinate descent iterations for computation of equalizer coefficients. The proposed technique has as low complexity as $\mathcal{O}(N_u(K+M))$ operations per sample, where K and M are the equalizer and channel estimator length, respectively, and N_u is the number of iterations such that $N_u \ll K$ and $N_u \ll M$. Moreover, with dichotomous coordinate descent iterations, the computation of equalizer coefficients is multiplication-free and division-free, which makes the equalizer attractive for hardware design. Simulation shows that the proposed adaptive equalizer performs close to the minimum mean-square-error equalizer with perfect knowledge of the channel.**

***Index Terms*—Adaptive equalization, linear equalizer, channel estimation, dichotomous coordinate descent, DCD**

I. Introduction

Equalization is a well known method for combatting the inter-symbol interference in communication channels [1]. Coefficients of an adaptive linear equalizer (LE) can be computed without explicit channel estimation using the channel output and known pilot signal [1]. However, channel-estimate (CE) based equalizers can outperform LEs with the direct adaptation [2]. The CE based adaptive equalizers re-compute equalizer coefficients for every update of the channel estimate, preferably for every sample of a received signal. This requires generation and inversion of a $K \times K$ channel autocorrelation matrix, where K is the equalizer length. In general, it results in a complexity of $\mathcal{O}(K^3)$ operations per sample. Exploiting structural properties of the matrix, the complexity can be reduced down to $\mathcal{O}(K^2)$ operations [3]. Recursive least squares (RLS) adaptive channel estimators have a complexity $\mathcal{O}(M^2)$, where M is the channel estimator length [4]. It is usual that $K > M$, thus the complexity of computing the equalizer coefficients determines the total complexity. Moreover, recently computationally efficient iterative adaptive algorithms of complexity $\mathcal{O}(N_u M)$ have been proposed, where $N_u << M$, with a performance close to that of the RLS algorithm [5]. Thus, adaptive channel estimation can be significantly simpler than CE based computation of equalizer coefficients. To reduce the whole complexity, computation of equalizer coefficients should be simplified.

In this paper, we propose a novel CE based adaptive LE. The proposed equalizer is applicable for using together with channel estimators based on adaptive algorithms with partial update (see [6] and references therein), including adaptive algorithms with coordinate descent iterations [7], [8], [5]. Moreover, we show that when using the dichotomous coordinate descent (DCD) iterations [9], computation of equalizer coefficients can be multiplication-free and division-free. When using the DCD algorithm in both the channel estimator and equalizer, the overall complexity of the equalization is as low as $\mathcal{O}(N_u(K+M))$ operations per sample.

Notations: In this paper, we use capital and small bold fonts to denote matrices and vectors, respectively. For example, $\mathbf{G}$ and $\mathbf{r}$ represent a matrix and a vector, respectively. Elements of the matrix and vector are denoted as $G_{p,n}$ and r_n. A pth column of $\mathbf{G}$ is denoted as $\mathbf{G}^{(p)}$. $\mathbf{G}^T$ denotes transpose of matrix $\mathbf{G}$, and $\mathbf{I}_K$ is a $K \times K$ identity matrix. The variable n is used as a time index and i is iteration index. $E\{\cdot\}$ denotes the expectation. Only the real-valued case is considered in this paper; the extension to the complex-valued case is straightforward.

II. Preliminaries

We consider that the received signal $y(n)$ is given by

$$y(n) = \mathbf{x}^T(n)\mathbf{h}(n) + \nu(n), \tag{1}$$

where $\mathbf{x}(n) = [x(n)\ x(n-1)\dots\ x(n-M+1)]^T$, $x(n)$ is the transmitted signal, $\mathbf{h}(n) = [h_1(n)\ h_2(n)\dots\ h_M(n)]^T$ is the channel impulse response, and $\nu(n)$ is the white noise with zero mean and variance σ_ν^2. At time instant n, a K-length LE with the tap weight coefficient vector $\mathbf{f}(n)$ estimates the transmitted signal as $\hat{x}(n) = \mathbf{y}^T(n)\mathbf{f}(n)$, where $\mathbf{y}(n) = [y(n)\ y(n-1)\dots y(n-K+1)]^T$. The equalizer vector $\mathbf{f}(n)$ is adjusted to minimize the mean square error (MSE) $E\{[x(n)-\hat{x}(n)]^2\}$. For CE based equalization, minimizing the MSE requires solving the normal equations [1]

$$\mathbf{G}(n)\mathbf{f}(n) = \boldsymbol{\xi}(n), \tag{2}$$

where $\mathbf{G}(n) = \overline{\mathbf{H}}^T(n)\overline{\mathbf{H}}(n) + \sigma_\nu^2\mathbf{I}_K$, $\boldsymbol{\xi}(n) = \overline{\mathbf{H}}^T(n)\mathbf{e}_l$, $\mathbf{e}_l$ is a $(K+M-1)\times 1$ vector of all zeros except the lth element, which equals one and corresponds to the equalizer delay, and $\overline{\mathbf{H}}(n)$ is a $(K+M-1)\times K$ time-varying channel convolution matrix. In practice, as the time-varying channel is unknown, estimates $\hat{\mathbf{h}}(n-j), j = 0,\dots,K-1$, of the channel impulse

The authors are with the Department of Electronics, University of York, York YO10 5DD, UK. e-mail: (tc512@ohm.york.ac.uk; yz1@ohm.york.ac.uk; cl563@ohm.york.ac.uk).

response are used to form $\overline{\mathbf{H}}(n)$ as given by

$$\overline{\mathbf{H}}(n) = \begin{bmatrix} \hat{\mathbf{h}}(n) & 0 & \cdots & 0 \\ 0 & \hat{\mathbf{h}}(n-1) & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \hat{\mathbf{h}}(n-K+1) \end{bmatrix}. \quad (3)$$

III. Low-complexity CE based adaptive LE

A. Assumptions

We use the following assumptions:

1) For every time sample n, the channel estimate can be updated N_u times. We will be using index $i = (n-1)N_u + k$, where $k = 1, \ldots, N_u$, to indicate such an update. Correspondingly, the sequence of the normal equations to be solved in the MMSE LE is now given by

$$\mathbf{G}(i)\mathbf{f}(i) = \boldsymbol{\xi}(i). \quad (4)$$

2) For every i, the channel estimator updates only one, $p(i)$th, element in $\hat{\mathbf{h}}(i)$ as $\hat{h}_{p(i)}(i) = \hat{h}_{p(i)}(i-1) + \Delta\hat{h}(i)$.

3) For every i, only one, $q(i)$th, equalizer coefficient in $\hat{\mathbf{f}}(i)$ is updated as $\hat{f}_{q(i)}(i) = \hat{f}_{q(i)}(i-1) + \Delta\hat{f}(i)$. Here, $\hat{\mathbf{f}}(i)$ denotes an approximation to the MMSE solution $\mathbf{f}(i)$ at iteration i.

4) The convolution matrix (3) can be approximated for each i as

$$\mathbf{H}(i) = \begin{bmatrix} \hat{\mathbf{h}}(i) & 0 & \cdots & 0 \\ 0 & \hat{\mathbf{h}}(i) & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \hat{\mathbf{h}}(i) \end{bmatrix}. \quad (5)$$

The number of iterations for computing the equalizer coefficients after an update of the channel estimate can be made greater than one. This is a straightforward extension of the algorithm described below. However, our simulation (not presented here) has shown little improvement in the equalizer performance compared to the case of one iteration (as given by assumption 3).

B. Derivation

Equations (4) can be transformed into a sequence of auxiliary normal equations $\mathbf{G}(i)\Delta\mathbf{f}(i) = \boldsymbol{\xi}_0(i)$ [5]. A recursive approach for solving the equations is described in Table I [5], where: $\mathbf{r}(i)$ is the residual vector $\mathbf{r}(i) = \boldsymbol{\xi}(i) - \mathbf{G}(i)\hat{\mathbf{f}}(i)$; $\Delta\mathbf{G}(i) = \mathbf{G}(i) - \mathbf{G}(i-1)$; and $\Delta\boldsymbol{\xi}(i) = \boldsymbol{\xi}(i) - \boldsymbol{\xi}(i-1)$.

In Table I, step 1 requires finding $\Delta\mathbf{G}(i)$ which involves computation of the matrix $\mathbf{G}(i) = \mathbf{H}^T(i)\mathbf{H}(i)$ with a complexity of $\mathcal{O}(M^2)$. Step 2 requires $\mathcal{O}(MK)$ operations to compute $\Delta\mathbf{G}(i)\hat{\mathbf{f}}(i-1)$. These are the most computationally demanding operations and below we show how these operations can be simplified when using our assumptions.

Computation of $\Delta\mathbf{G}(i)\hat{\mathbf{f}}(i-1)$:

Let $\mathbf{H}(i) = \mathbf{H}(i-1) + \boldsymbol{\Delta}(i)$, then we have

$$\Delta\mathbf{G}(i)\hat{\mathbf{f}}(i-1) = \boldsymbol{\Delta}^T(i)\mathbf{H}(i-1)\hat{\mathbf{f}}(i-1) + \mathbf{H}^T(i-1)\boldsymbol{\Delta}(i)\hat{\mathbf{f}}(i-1) + \boldsymbol{\Delta}^T(i)\boldsymbol{\Delta}(i)\hat{\mathbf{f}}(i-1). \quad (6)$$

TABLE I
Recursively solving a sequence of equations

Step	Equation
	Initialization: $\mathbf{r}(0) = \mathbf{0}$, $\boldsymbol{\xi}(0) = \mathbf{0}$, $\hat{\mathbf{f}}(0) = \mathbf{0}$
	for $i = 1, 2, \ldots$
1	Find $\Delta\mathbf{G}(i)$ and $\Delta\boldsymbol{\xi}(i)$
2	$\boldsymbol{\xi}_0(i) = \mathbf{r}(i-1) + \Delta\boldsymbol{\xi}(i) - \Delta\mathbf{G}(i)\hat{\mathbf{f}}(i-1)$
3	Solve $\mathbf{G}(i)\Delta\mathbf{f} = \boldsymbol{\xi}_0(i) \Rightarrow \Delta\hat{\mathbf{f}}(i),\ \mathbf{r}(i)$
4	$\hat{\mathbf{f}}(i) = \hat{\mathbf{f}}(i-1) + \Delta\hat{\mathbf{f}}(i)$

Denoting $\mathbf{b}(i-1) = \mathbf{H}(i-1)\hat{\mathbf{f}}(i-1)$, we obtain

$$\mathbf{b}(i-1) = [\mathbf{H}(i-2) + \boldsymbol{\Delta}(i-1)][\hat{\mathbf{f}}(i-2) + \Delta\hat{\mathbf{f}}(i-1)],$$

which gives a recursion for $\mathbf{b}(i-1)$:

$$\mathbf{b}(i-1) = \mathbf{b}(i-2) + \mathbf{H}(i-2)\Delta\hat{\mathbf{f}}(i-1) + \boldsymbol{\Delta}(i-1)\hat{\mathbf{f}}(i-1). \quad (7)$$

Note that $\boldsymbol{\Delta}(i-1)$ is a Toeplitz matrix whose first column is $\Delta\hat{h}(i-1)\mathbf{e}_{p(i-1)}$. We also have $\Delta\hat{\mathbf{f}}(i-1) = \Delta\hat{f}(i-1)\mathbf{e}_{q(i-1)}$. Then (7) can be rewritten as

$$\mathbf{b}(i-1) = \mathbf{b}(i-2) + \Delta\hat{f}(i-1)\hat{\mathbf{h}}^{[q(i-1)]}(i-2) + \Delta\hat{h}(i-1)\hat{\mathbf{f}}^{[p(i-1)]}(i-1),$$

where $\hat{\mathbf{h}}^{[q(i-1)]}(i-2)$ is a $(K+M-1)\times 1$ vector obtained by shifting elements of $\hat{\mathbf{h}}(i-2)$ by $q(i-1)$ positions down, and the other elements of $\hat{\mathbf{h}}^{[q(i-1)]}(i-2)$ are zeros. Definition for $\hat{\mathbf{f}}^{[p(i-1)]}(i-1)$ is similar to that of $\hat{\mathbf{h}}^{[q(i-1)]}(i-2)$. Thus, the first term on the right hand side of (6) is given by

$$\boldsymbol{\Delta}^T(i)\mathbf{H}(i-1)\hat{\mathbf{f}}(i-1) = \boldsymbol{\Delta}^T(i)\mathbf{b}(i-1) = \Delta\hat{h}(i)\mathbf{b}_{p(i):p(i)+K-1}(i-1), \quad (8)$$

where $\mathbf{b}_{p(i):p(i)+K-1}(i-1)$ is a $K\times 1$ vector whose elements are obtained by extracting the $p(i)$th to $p(i)+K-1$th elements from the vector $\mathbf{b}(i-1)$. After some algebra, we find that the second term on the right hand side of (6) can be expressed as

$$\mathbf{H}^T(i-1)\boldsymbol{\Delta}(i)\hat{\mathbf{f}}(i-1) = \boldsymbol{\Delta}^T(i)\mathbf{c}(i-1) = \Delta\hat{h}(i)\mathbf{c}_{M-p(i)+1:M-p(i)+K}(i-1), \quad (9)$$

where, for the vector $\mathbf{c}(i-1)$ we obtain a recursion similar to that for $\mathbf{b}(i-1)$:

$$\mathbf{c}(i-1) = \mathbf{c}(i-2) + \Delta\hat{f}(i-1)\hat{\mathbf{u}}^{[q(i-1)]}(i-2) + \Delta\hat{h}(i-1)\hat{\mathbf{f}}^{[M-p(i-1)+1]}(i-1),$$

where elements of the vector $\hat{\mathbf{u}}(i-2)$ are given by

$$\hat{u}_m(i-2) = \hat{h}_{M-m+1}(i-2), m = 1, \ldots, M.$$

Since $\boldsymbol{\Delta}^T(i)\boldsymbol{\Delta}(i) = \Delta\hat{h}^2(i)\mathbf{I}_K$, the third term on the right hand side of (6) is given by

$$\boldsymbol{\Delta}^T(i)\boldsymbol{\Delta}(i)\hat{\mathbf{f}}(i-1) = \Delta\hat{h}^2(i)\hat{\mathbf{f}}(i-1). \quad (10)$$

From (8), (9) and (10), denoting $\mathbf{z}(i) = \Delta\mathbf{G}(i)\hat{\mathbf{f}}(i-1)$, we finally obtain a simplified expression for (6):

$$\mathbf{z}(i) = \Delta\hat{h}(i)\left[\mathbf{b}_{p(i):p(i)+K-1}(i-1) + \mathbf{c}_{M-p(i)+1:M-p(i)+K}(i-1) + \Delta\hat{h}(i)\hat{\mathbf{f}}(i-1)\right].$$

TABLE II
LOW-COMPLEXITY COMPUTATION OF CE BASED EQUALIZER COEFFICIENTS

Step	Equation	×	+
	Initialization: $i=0$, $\mathbf{G}(0)=\sigma_\nu^2\mathbf{I}_K$, $\hat{\mathbf{f}}(0)=\mathbf{0}$, $\mathbf{r}(0)=\mathbf{0}$, $\mathbf{b}(0)=\mathbf{0}$, $\mathbf{c}(0)=\mathbf{0}$		
	for $n=1,2,\ldots$		
	for $k=1,\ldots,N_u$		
1	$i=i+1$		
2	Obtain $\hat{\mathbf{h}}(i)$, $\Delta\hat{h}(i)$ and position $p(i)$ from a channel estimator		
3	$\mathbf{z}(i)=\Delta\hat{h}(i)[\mathbf{b}_{p(i):p(i)+K-1}(i-1)+\mathbf{c}_{M-p(i)+1:M-p(i)+K}(i-1)+\Delta\hat{h}(i)\hat{\mathbf{f}}(i-1)]$	$2K$	$2K$
4	$\boldsymbol{\xi}_0=\mathbf{r}(i-1)+\Delta\hat{h}(i)\mathbf{e}_{l+p(i)}-\mathbf{z}(i)$	$-$	$2K+1$
5	Compute $\Delta\mathbf{G}^{(1)}(i)$ using (11) and update $\mathbf{G}^{(1)}(i)=\mathbf{G}^{(1)}(i-1)+\Delta\mathbf{G}^{(1)}(i)$	M	$2M$
6	Use one iteration to solve $\mathbf{G}\Delta\mathbf{f}=\boldsymbol{\xi}_0$ and obtain $\Delta\hat{f}(i)$, $q(i)$, and $\mathbf{r}(i)$	P_{mu}	P_{ad}
7	$\hat{f}_{q(i)}(i)=\hat{f}_{q(i)}(i-1)+\Delta\hat{f}(i)$	$-$	1
8	$\mathbf{b}(i)=\mathbf{b}(i-1)+\Delta\hat{f}(i)\hat{\mathbf{h}}^{[q(i)]}(i-1)+\Delta\hat{h}(i)\hat{\mathbf{f}}^{[p(i)]}(i)$	$K+M$	$K+M$
9	$\mathbf{c}(i)=\mathbf{c}(i-1)+\Delta\hat{f}(i)\hat{\mathbf{u}}^{[q(i)]}(i-1)+\Delta\hat{h}(i)\hat{\mathbf{f}}^{[M-p(i)+1]}(i)$	$K+M$	$K+M$
	Total for each sample n: $N_u(4K+3M+P_{mu})$ mult. and $N_u(6K+4M+1+P_{ad})$ adds		

Computation of $\Delta\mathbf{G}(i)$:

The matrix $\mathbf{G}(i)$ is a symmetric Toeplitz matrix and $\Delta\mathbf{G}(i)$ is also a symmetric Toeplitz matrix. For each update of the channel estimate, only the first column of $\Delta\mathbf{G}(i)$ needs to be updated. In this column, only the first M elements are nonzero, which are given by

$$\Delta G_{1,1}(i)=\Delta\hat{h}(i)\left[\hat{h}_{p(i)}(i)+\hat{h}_{p(i)}(i-1)\right], \quad (11)$$
$$\Delta G_{1,m}(i)=\Delta\hat{h}(i)\left[\hat{h}_{p(i)-m+1}(i-1)+\hat{h}_{p(i)+m-1}(i-1)\right],$$

where $m=2,\ldots,M$.

The proposed technique for computing the equalizer coefficients is now summarized in Table II. Here, we assume that the noise variance σ_ν^2 is known. Table II also shows the complexity of the computation steps in terms of multiplications and additions. The complexity of computing the LE coefficients will depend on the iterative technique used for solving the equation $\mathbf{G}\Delta\mathbf{f}=\boldsymbol{\xi}_0$ at step 6, where P_{mu} and P_{ad} denote the number of multiplications and additions, respectively.

C. DCD iterations

We propose to use the DCD iteration described in Table III, which is simple for implementation and shows fast convergence to optimal performance [5]. When using the DCD iteration, it is assumed that the equalizer coefficients are represented as M_b-bit fixed-point numbers within an interval $[-A,A]$, where A is preferably a power-of-two number. The step-size parameter α is $\alpha=2^{-a}A$, i.e. also a power-of-two number. With such settings, operations required in the DCD algorithm are only additions as all multiplications and divisions are replaced by bit-shifts; see more details on the parameter choice in [5]. If, in addition, the adaptive channel estimator is implemented using the RLS-DCD adaptive filter of complexity $\mathcal{O}(N_uM)$ [5], the increments $\Delta\hat{h}(i)$ will be power-of-two numbers. Therefore, all multiplications in Table II can be replaced by bit-shift operations. With the DCD iteration, step 6 in Table II is multiplication-free and requires no more than $2K+M_b+1$ additions.

TABLE III
DCD ALGORITHM WITH ONE UPDATE

Step	Equation	+
	Initialization: $\mathbf{r}=\boldsymbol{\xi}_0$, $\alpha=A/2$, $a=1$	
1	$q=\arg\max_{j=1,\ldots,K}\{\lvert r_j\rvert\}$, go to step 4	$K-1$
2	$a=a+1$, $\alpha=\alpha/2$	$-$
3	if $a>M_b$, the algorithm stops	$-$
4	if $\lvert r_q\rvert\le(\alpha/2)G_{q,q}$, then go to step 2	1
5	$\Delta\hat{f}=\text{sign}(r_q)\alpha$	1
6	$\mathbf{r}=\mathbf{r}-\text{sign}(r_q)\alpha\mathbf{G}^{(q)}$	K
	$\Delta\hat{f}(i)=\Delta\hat{f}$, $q(i)=q$, $\mathbf{r}(i)=\mathbf{r}$	
	Total: $P_{mu}=0$ and $P_{ad}\le 2K+M_b+1$	

IV. SIMULATION RESULTS

In this section, we compare the performance of seven LEs: 1) MMSE LE. For every time sample n, the convolution matrix $\overline{\mathbf{H}}(n)$ is formed using the perfectly known channel response $\mathbf{h}(n-j), j=0,\ldots,K-1$, instead of its estimate as in (3). The equalizer vector is found by solving (2); 2) RLS CE based adaptive LE. The time-varying channel is estimated using the classical RLS algorithm with a forgetting factor λ, and for every sample n, the convolution matrix $\overline{\mathbf{H}}(n)$ is formed using (3). The equalizer vector is obtained by solving (2); 3) LMS CE based adaptive LE. This is similar to the RLS CE based adaptive LE except that the time-varying channel is estimated using the classical LMS algorithm [4]; 4) RLS CE based adaptive LE (K samples). This is the RLS CE based adaptive LE which estimates the time-varying channel for every sample n, while the equalizer coefficients are computed once for K samples; 5) RLS directly adaptive (DA) LE. The equalizer coefficients are directly computed for every sample n using the RLS algorithm [1]; 6) LMS DA LE. The equalizer coefficients are directly computed for every sample n using the LMS algorithm [1]; 7) Proposed LE. The time-varying channel is estimated using the RLS-DCD algorithm from [5] with a forgetting factor λ and for every i, the leading index $p(i)$ is chosen according to the position of the maximum in the residual vector (see [5]). The choice of N_u for the DCD

algorithm is investigated in [5], [10]. The equalizer vector is obtained using the algorithm in Table II.

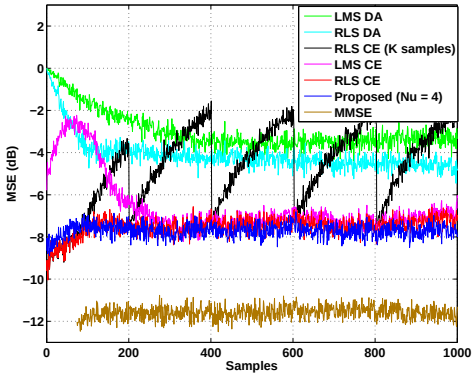

Fig. 1. MSE performance of LEs: SNR $= 20$ dB, $\upsilon = 1 - 10^{-3}$, $M = 51$, $K = 201$, $M_b = 16$, $A = 1$, forgetting factor is 0.9804 for the RLS CE and the proposed LE and 0.995 for the RLS DA LE, step size is 0.005 for the LMS DA LE and 0.02 for the LMS CE.

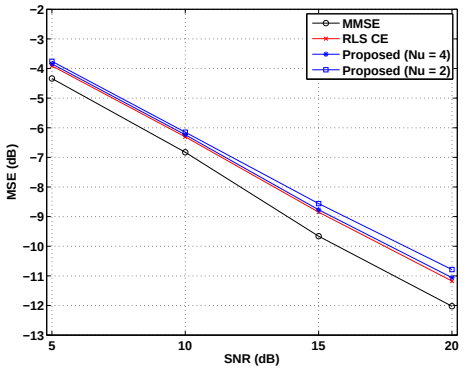

Fig. 2. Steady-state MSE performance of the three LEs at different SNRs: $\upsilon = 1 - 10^{-4}$, $M = 51$, $K = 201$, $M_b = 16$, $A = 1$; forgetting factor is 0.9951 for SNR $= 5$ and 10 dB, 0.9902 for SNR $= 15$ and 20 dB, and 0.9804 for SNR $= 25$ dB.

To simulate the time-varying channel impulse response $\mathbf{h}(n)$, we adopt the first order autoregressive model given by $\mathbf{h}(n) = \sqrt{\upsilon}\ \mathbf{h}(n-1) + \sqrt{1-\upsilon}\ \boldsymbol{\omega}(n)$ [11], where $\sqrt{\upsilon}$ is the autoregressive factor and $\boldsymbol{\omega}(n)$ are zero-mean independent random Gaussian vectors, whose elements have variance $1/M$. The channel length is $M = 51$ and the equalizer length is $K = 201$. We use $l = (K+M)/2 = 126$ as the equalizer delay [12]. Different signal to noise ratios (SNRs) are considered, and for each SNR, simulation results are obtained by averaging over 500 independent simulation trials. For each trial, 1000 pilot symbols of unit power are transmitted.

Fig.1 compares the MSE performance of the seven LEs for SNR $= 20$ dB and the time-varying channel with $\upsilon = 1 - 10^{-3}$. For computing the MSE for each n, a 1000-length data sequence independent of the pilot is filtered with the equalizer vector $\hat{\mathbf{f}}(n)$ derived using the pilot. It is seen that the proposed LE performs very close to the RLS CE based adaptive LE and outperforms the other LEs.

Fig.2 compares the MSE performance of LEs at different SNRs for a time-varying channel with $\upsilon = 1 - 10^{-4}$. For a simulation trial, the steady-state MSE is evaluated as MSE $= \frac{1}{926}\sum_{n=75}^{1000}[x(n) - \mathbf{y}^T(n)\hat{\mathbf{f}}(n)]^2$. It is seen that with $N_u = 4$ and even $N_u = 2$, the proposed LE provides performance very close to that of the RLS CE based adaptive LE, and close to that of the MMSE LE.

TABLE IV
NUMBER OF MULTIPLICATIONS PER SAMPLE ($M = 51$)

		$K = 51$	$K = 101$	$K = 201$	$K = 401$
MMSE		3.9×10^5	2.5×10^6	1.8×10^7	1.3×10^8
RLS CE		4.1×10^5	2.6×10^6	1.8×10^7	1.3×10^8
RLS CE (K samples)		8113	2.6×10^4	9.1×10^4	3.4×10^5
LMS CE		3.9×10^5	2.5×10^6	1.8×10^7	1.3×10^8
RLS DA		1.6×10^4	6.2×10^4	2.4×10^5	9.7×10^5
LMS DA		153	303	603	1203
Proposed	$N_u = 2$	1020	1470	2370	4170
	$N_u = 4$	1734	2584	4284	7684

Table IV compares the number of multiplications required in different LEs at each sample for different equalizer length. It is seen that the proposed LE has much lower computational complexity than the other CE based LEs. Its complexity is also significantly lower than that of the RLS DA LE.

V. CONCLUSIONS

In this work, we have proposed a channel-estimate based adaptive LE with a complexity as low as $\mathcal{O}(N_u(K+M))$ operations per sample, where $N_u \ll K$ and $N_u \ll M$. The proposed technique exploits coordinate descent iterations for computing the equalizer coefficients. Moreover, when using the dichotomous coordinate descent iterations, computation of the equalizer coefficients is multiplication-free and division-free, which makes it attractive for hardware design. Simulation results show that, with only a few updates per sample, the proposed LE performs very close to the RLS CE based adaptive LE and close to the MMSE LE with perfect knowledge of the channel.

REFERENCES

[1] J. Proakis, *Digital communications*. McGraw-Hill New York, 1995.
[2] R. A. Ziegler, M. W. Al-Dhahir, and J. M. Cioffi, "Nonrecursive adaptive decision-feedback equalization from channel estimates," in *IEEE 42nd Vehicular Technology Conference*, 1992, pp. 600–603.
[3] N. Al-Dhahir and J. Cioffi, "Fast computation of channel-estimate based equalizers in packet data transmission," *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2462–2473, Nov. 1995.
[4] S. Haykin, *Adaptive Filter Theory (4th Edition)*. Prentice Hall, 2001.
[5] Y. Zakharov, G. White, and J. Liu, "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Transactions on Signal Processing*, vol. 56, no. 7 Part 2, pp. 3150–3161, 2008.
[6] K. Dogancay and O. Tanrikulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 8, pp. 762–769, 2002.
[7] G. Xu, T. Bose, and J. Schroeder, "The Euclidean direction search algorithm for adaptive filtering," in *Proceedings of IEEE Int. Symp. Circuits and Systems, ISCAS'99.*, vol. 3, 1999, pp. 146–149.
[8] G. Xu, T. Bose, and J. Schroeder, "Channel equalization using an Euclidean direction search based adaptive algorithm," in *Global Telecommunications Conference, GLOBECOM 98.*, vol. 6, 1998, pp. 3479–3484.
[9] Y. Zakharov and T. Tozer, "Multiplication-free iterative algorithm for LS problem," *Electronics Letters*, vol. 40, pp. 567–569, April 2004.
[10] J. Liu, Y. Zakharov, and B. Weaver, "Architecture and FPGA design of dichotomous coordinate descent algorithms," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 56, no. 11, pp. 2425–2438, 2009.

[11] R. Otnes and M. Tuchler, "Low-complexity turbo equalization for time-varying channels," in *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th*, vol. 1, 2002, pp. 140–144.
[12] P. Butler and A. Cantoni, "Noniterative automatic equalization," *IEEE Transactions on Communications*, vol. 23, no. 6, pp. 621–633, 1975.