An Efficient Low Complexity Cluster-Based MLSE Equalizer for Frequency-Selective Fading Channels

Yannis Kopsinis, Member, IEEE, Sergios Theodoridis, Senior Member, IEEE, and Eleftherios Kofidis, Member, IEEE

Abstract—Recently, a novel MLSE equalizer was reported, that does not require the explicit estimation of the channel impulse response. Instead, it utilizes, in an efficient manner, the estimates of the centers of the clusters formed by the received observations. In this paper, a novel cluster tracking scheme is presented, which extends the application of this equalizer in time-varying transmission environments. The proposed algorithm is shown to be equivalent in tracking performance with the classic LMS-based MLSE equalizer, yet much simpler computationally. This is a consequence of the fact that the new method allows for an efficient exploitation of the symmetries underlying the signaling scheme.

Index Terms—Bayesian decision-feedback equalization, channel tracking, center estimation, channel equalization, clustering, fading channels, Least Mean Squares (LMS) algorithm, Maximum Likelihood Sequence Estimation (MLSE).

I. INTRODUCTION

TWO OF THE major problems encountered in mobile communication systems are those of InterSymbol Interference (ISI) and channel time variation. ISI can be mitigated by the maximum likelihood sequence estimation (MLSE) equalizer, which is effectively implemented by the Viterbi algorithm (VA) [1]. The MLSE equalizer utilizes the channel impulse response (CIR), which is estimated during the start up period, prior to the processing of each data block, based on an a priori known training sequence. However, under time varying conditions, it is possible for the CIR to change significantly during the transmission of a data block, leading to serious performance degradation. For this reason, the CIR has to be continuously re-estimated utilizing the decisions obtained by the VA, in order to track the channel variations.

Unfortunately, the VA provides reliable decisions only for symbols at time instants prior to \( k - D, D \approx 5L \), where \( L \) is the CIR length and \( k \) the time of the most recent received observation [1]. This decision delay is inherent in the VA and leads to poor performance since the channel taps used at the most recent time, \( k \), are estimates computed \( D \) time instances earlier, i.e., at time \( k - D \), and in the mean time the channel may have changed. Many methods have been proposed in order to address this problem associated with the decision delay. These have evolved around the following main directions:

- Track the channel by utilizing tentative decisions corresponding to a small fixed delay, \( d \), usually as long as \( L \) [2].
- Use tentative decisions that correspond to a long enough fixed delay, \( d \), in conjunction with an appropriately defined channel predictor [3], [4] to account for the channel variations.
- Adopt a Per Survivor Processing (PSP) philosophy, which results in channel estimates with no delay [5], [6].

The estimates of the channel taps are commonly adapted via the Least-Mean Squares (LMS) algorithm [3].

The main drawback of all of the above MLSE equalizers is their high computational load, which may limit their practical use. In the present paper a novel MLSE equalizer is presented, which can offer substantial computational savings in all the required processing stages, namely: a) the start up phase, b) the tracking phase, and c) the distance metrics computation in the VA. The proposed equalizer belongs to the family of Cluster-Based Sequence Equalizers (CBSE) [7], [8], [9], [10] since it utilizes the clusters formed by the received observations, circumventing the problem of explicit CIR parametric modeling. It is based on a recently reported algorithm [9], [10], called 1-D CBSE, originally developed for stationary channels. Initialization is achieved via an effective cluster center estimation technique [9], which exploits the structural symmetries underlying the generation mechanism of the clusters of the received samples. To extend 1-D CBSE in time varying environments, a novel cluster tracking scheme, which is equivalent in performance to the LMS algorithm, is introduced in this paper. The total computational complexity of the proposed equalizer is much lower compared to the complexity of the standard MLSE.

All the above mentioned approaches to the mitigation of the channel estimation delay problem can be employed with this new cluster-based scheme, in a straightforward manner. A number of these variants of the 1-D CBSE equalizer are developed in this paper and compared with the corresponding versions of the MLSE-LMS and the finite memory Bayesian decision feedback (DF) equalizers [11], [12] in terms of performance and computational complexity.

II. DESCRIPTION OF THE COMMUNICATION SYSTEM

In our baseband communication system model, the following notation is used. \( x_k \) denotes the \( k \)th transmitted symbol,
which can take $M$ distinct values, $n_k$ is the white complex-valued additive noise at the channel output, and $y_k$ is the $k$th received observation. The received signal, sampled at $t = kT$, with $T$ being the transmission period of the symbols, is given by

$$y_k = \sum_{i=0}^{L-1} h_{k,i}x_{k-i} + n_k = h_k^T x_k + n_k = \bar{y}_k + n_k \quad (1)$$

where $h_k = [h_{k0}, h_{k1}, \ldots, h_{k,L-1}]^T$ is the vector of the $L$, $T$-spaced, taps of the CIR at time $k$ and $\bar{y}_k$ indicates the noiseless observation associated with the transmitted sequence of symbols $x_k = [x_k, x_{k-1}, \ldots, x_{k-L+1}]^T$. The input to the channel is assumed i.i.d. The variance of the noise is $\sigma^2$ and the signal-to-noise ratio is given by $\text{SNR} = \left(\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} E[|\bar{y}_i|^2]\right) / \sigma^2$.

### III. The One-Dimensional CBSE

The MLSE equalizer is efficiently implemented with the VA, which estimates the transmitted symbol sequence optimally, based on metric computations of the form $D_{k,x} = |y_k - h_k^T x|^2$, with $x$ ranging over the set of the $M^L$ possible $L$-tuples of input symbols. In the case of the standard MLSE, the CIR estimate is needed only for the computation of the $M^L$ convolutions, $h_k^T x$, at every stage of the trellis diagram. However, this operation poses high computational demands, which can be rather unacceptable for many practical communication systems. Significant computational savings, without any loss in performance, can be achieved with the recently published *One-Dimensional Cluster-Based Sequence Equalizer (1-D CBSE)* $^3$ $^9$. The latter equalizer avoids the explicit estimation of the CIR by estimating, instead, the noiseless observations $\bar{y}_{k,x} = h_k^T x$, which are uniquely determined by the vectors $x$, directly. Thus, the distance metric in the VA becomes $D_{k,x} = |y_k - \bar{y}_{k,x}|^2$.

In fact, all the possible values that the noiseless observations $\bar{y}_{k,x}$ can take are simply the points (centers) on the complex plane around which the received samples $y_k$ are clustered due to the noise. Thus, the problem of the parametric estimation of the CIR, as it is commonly required in MLSE equalizers, can be circumvented and all that is needed is to estimate the $M^L$ centers $\bar{y}_{k,x}$ of the clusters formed on the complex plane.

When the channel varies with time, a center tracking scheme has to be included in order to allow for the tracking of the cluster centers while they move away from their initial positions. This is the main contribution of this paper and is developed in the next section. The equalizer needs also a center estimation scheme, able to provide reliable initial estimates of the cluster centers per data block. Recently, a new method for the cluster center estimation has been presented $^9$, which requires neither the direct information from all the clusters $^{13}$ nor the CIR estimate as an intermediate step before the center computation $^{11}, ^{12}$. In contrast, the new center estimation (CE) method exploits the symmetries in the constellation of the cluster centers in the complex plane and in this way it succeeds in estimating all the $M^L$ centers based on the direct estimates of only $L$ properly selected centers, thus speeding up the training period dramatically. An extensive discussion of the CE technique can be found in $^9$. We briefly present here the specific structure formed by the cluster centers, since it is a key point for the understanding of the proposed center tracking method. The necessary definitions and notations are also given.

Both the cluster center estimation technique and the novel center tracking scheme are applicable to any symmetric constellation $^{10}, ^9$. However, for convenience, the presentation will adopt the QPSK signaling scheme, where each data symbol can take one among $M = 4$ distinct values: $1 + j, 1 - j, -1 + j, -1 - j$. Let us assume a general $L$-taps channel with impulse response vector $h = [h_0, h_1, \ldots, h_m, \ldots, h_{L-1}]^T$, which is considered to be approximately constant during the transmission of a short training sequence. We define the contribution, $c^m_x$, of the $m$th tap, $h_m$, to the generation of a cluster center as the quantity

$$c^m_x = x h_m, \quad x \in \{1 + j, 1 - j, -1 + j, -1 - j\} \quad (2)$$

In other words, this is the contribution of the $h_m$ tap to the convolution sum in (1). Using this notation, (1) can be rewritten as

$$\bar{y}[x_k, x_{k-1}, \ldots, x_{k-L+1}] = \sum_{m=0}^{L-1} c^m_{x_k-m} \quad (3)$$

where $\bar{y}[x_k, x_{k-1}, \ldots, x_{k-L+1}]$ is the cluster center associated with the transmitted $L$-tuple $[x_k, x_{k-1}, \ldots, x_{k-L+1}]$. We can observe that $c^m_x$ can take one out of four possible values, depending on the value of the symbol $x$. It is easy to realize that, for each $h_m$, only one of these four possible values, say $c_{1+j}^m$, needs to be known, and the rest can be obtained by simple $\frac{\pi}{2}$ rotations in the complex plane $^9$. (A discussion of

---

$^1$Superscript $T$ denotes transposition.

$^2$Forney’s MLSE scheme $^1$ is assumed here.

$^3$The 1-D CBSE algorithm is a reduced complexity version of the previously proposed CBSE equalizers $^7, ^8$. 

![Fig. 1. Cluster center constellation for a 2-taps channel.](image-url)
how is this generalized for the case of $M > 4$ is deferred to Section VI).

The specific way that the centers are positioned on the complex plane is exposed in Fig. 1, via a 2-taps example. The $4^L = 16$ centers, denoted by filled circles ($\bullet$), form 4 similar squares, whose size and angle of rotation are determined by the contribution $c^L_1$ of the second tap. These squares are centered on the corners of a fifth central square, drawn in dashed line, which is associated with the contribution $c^L_2$ of the first tap.

This structure was efficiently exploited in [9] in order to estimate the $M^L$ cluster centers based on the direct estimates of only $L$ of them, using training sequences that are formed as periodic repetitions of the symbol string

$$[x, x, \ldots, x, -x]^\top$$

where $x$ can be any of the symbols of the adopted alphabet. It was shown in [14] that, when such a training sequence is used, the centers estimated with the proposed method are Least Squares (LS) optimal, i.e., they coincide with those that would have been computed if the LS estimate for the CIR were used in (2) and (3). Furthermore, for channels up to 6 taps long, the CE method performs similarly with the LS method (even if the latter is trained with random training data) and much better than the LMS algorithm.

**IV. CLUSTER CENTER TRACKING TECHNIQUE**

In the tracking mode, the MLSE equalizers exploit the delayed decisions, provided by the VA, to adapt the channel estimates. Similarly, the novel tracking scheme utilizes the same decisions to update the center estimates in the complex plane. Previously proposed center tracking methods, e.g., [13], update only one center per received sample. This results in very poor performance, because many centers remain constant for a long period of time even if, in the meantime, the transmission environment has significantly changed. In order to deal with this difficulty, it was proposed to track the CIR instead of the cluster centers themselves. However, such an approach overlooks all the advantages offered by cluster-based methods, increasing the computational complexity due to the need of convolution calculations per detected symbol. In contrast to these methods, our technique manages to efficiently adapt all the centers simultaneously, each time a transmitted symbol is detected, avoiding the convolution calculation.

Even though the cluster centers are moving in the complex plane, their general structure remains the same, as it has been exposed in Section III. The centers are in the corners of $4^{L-1}$ similar squares, whose sizes and angles of rotation are determined by the contribution of the $L$th tap. Their position is determined by the contributions of the first $L - 1$ taps.

In order to track all the cluster centers, what is needed is to rotate and resize the corresponding squares toward the direction which is defined by each detected symbol. Let us take a 2-taps example under noiseless transmission, for illustration purposes. In Fig. 2 the geometry of the centers and the associated squares as they have been formed at time instant $k$ are illustrated, with solid lines. As we mentioned in the Introduction, the VA provides, for tracking purposes, tentative decisions with a delay $d$. So, at the next processing time instance, $k+1$, the VA can decide on which transmitted vector $x$ generated the received sample $y_{k-d+1}$ (denoted in Fig. 2 by $\times$). Let us say that the VA decides $x = [1-j, 1-j]$, then, after the adaptation, the center $\bar{y}_{[1-j, 1-j]}$ should be moved to coincide with the point $\times$. To this end, we shall assume here that the channel variation affects equally all the tap contributions (all taps change equally fast). For the adaptation we can adopt a two-step procedure:

**Step 1:** The central square is resized and rotated in accordance with the four small arrows, of length $a/2$, as it is shown with dotted lines in Fig. 2 ($a/L$ for an $L$-taps channel). The four external squares are then “re-centered” on the corners of the adapted central square. As a consequence of this re-location, the center $\bar{y}_{[1-j, 1-j]}$ is moved towards the point $\times$, covering half of the required distance. Moreover, the specific structure of the squares is also preserved. The result is shown in Fig. 3 with solid lines.

**Step 2:** Next, following a similar rationale, the four external squares are resized and rotated, as it is shown in Fig. 3 with dotted lines. The final positions of the centers, after the adaptation procedure, are denoted by open circles ($\circ$).

Taking into account the fact that both, the size and the angle of rotation of the squares, are determined by the corresponding tap contributions, the above procedure is efficiently implemented as follows:

$^4$This channel length covers a wide range of contemporary wireless communication systems.

---

**Fig. 2.** Adaptation of the square associated with the contribution of the first channel tap. The central square, drawn with solid line, is turned into the dotted one.
1) Compute an estimate of the center $\hat{y}_x$ based on the estimates $c^i_{k-d,x_i}$, $0 \leq i \leq L-1$, of the tap contributions at time instance $k-d$, where $x = [x_0, \ldots, x_i, \ldots, x_{L-1}]$ consists of the latest tentative decisions of the VA and corresponds to the received sample $y_{k-d+1}$:

$$\hat{y}_x = \sum_{i=0}^{L-1} c^i_{k-d,x_i}. \quad (5)$$

2) Compute the error between the received sample $y_{k-d+1}$ and the corresponding center estimate:

$$a = y_{k-d+1} - \hat{y}_x. \quad (6)$$

3) Adapt the tap-contribution estimates according to:

$$c^i_{k-d+1,x_i} = c^i_{k-d,x_i} + \lambda a, \quad i = 0, 1, \ldots, L - 1, \quad (7)$$

where $\lambda$ takes positive values less than or equal to $1/L$. As it is common in this type of adaptation schemes, the choice of $\lambda$ depends on the noise power. In the noiseless case, one should choose $\lambda = \frac{1}{L}$ since, as it was demonstrated in the above example, this will update $\hat{y}_x$ so that it coincides with $y_{k-d+1}$. When noise is present, smaller values of $\lambda$ have to be employed. More on choosing $\lambda$ will be said later on in this section.

4) Compute the other values of the tap-contributions by $\frac{\pi}{2}$ rotations.

In fact, the above adaptation strategy favors equally all the tap-contributions ignoring their specific value,\(^6\) $c^i_{k,x_i}$, which is determined by the symbol $x_i$. This is reasonable in the case of signaling schemes, like QPSK, in which all the symbols have equal energies. However, in the general case of $M$-QAM signaling, the various symbols could be treated differently, since lower energy ones are expected to be affected more by the noise. This suggests that one should trust more the symbols of higher energy. Hence, it is reasonable that the required distance $a$ be covered by resizing more (by appropriate weighting) the squares corresponding to tap-contributions with higher values. This allows us to better exploit the immunity to noise of the high energy symbols. Therefore, the adaptation equation of the center tracking method (7) is modified to include a symbol energy dependent adaptation factor as shown below:

$$c^i_{k-d+1,x_i} = c^i_{k-d,x_i} + \lambda E_{x_i} a, \quad i = 0, 1, \ldots, L - 1, \quad (8)$$

For example, $E_{x_i}$ can be chosen to be the energy of the $i$th symbol, $E_{x_i} = |x_i|^2$. This variant of the center tracking scheme is expected to perform better than that of eq. (7), since it exploits more efficiently the information about the symbol energy.

Another alternative would be to choose the weighting factor as

$$E_{x_i} = \frac{|x_i|^2}{\|x_i\|^2}, \quad (9)$$

i.e., each contribution is adapted in accordance with the percentage of the energy of the respective symbol with respect to the total energy of $x = [x_0, \ldots, x_{L-1}]$. It is not difficult to see geometrically that the use of the normalized form of eq. (9) guarantees that the updated center $\hat{y}_x$ coincides with the point $y_{k-d+1}$, in the noiseless case. This is not the case for the unnormalized version of eq. (8) with $E_{x_i} = |x_i|^2$.

A. Equivalence with the LMS-based tracking

We will now show that the above cluster tracking techniques are strongly related to the LMS tracking algorithm. More specifically, the cluster tracking scheme of eq. (8) with $E_{x_i} = |x_i|^2$ is equivalent to the standard LMS algorithm and its variant with $E_{x_i}$ chosen as in eq. (9) is equivalent to normalized LMS [15]. The channel tracking method of eq. (7) is equivalent to the LMS algorithm in the case of signaling schemes with equal symbol energies. It is important, though, to stress out the point that “equivalent to LMS” here means that the end result of the tracking procedure will be the same. Nevertheless, LMS acts on the channel taps and the new method on the clusters directly. This distinct difference is what provides to the cluster method substantial computational advantages, as we will soon see.

Proof: Eq. (8) can be rewritten as $\hat{h}_{k-d+1,i} x_i = \hat{h}_{k-d,i} x_i + \lambda E_{x_i} a$ or equivalently $\hat{h}_{k-d+1,i} x_i = \hat{h}_{k-d,i} x_i + \lambda E_{x_i} a$, taking into account that $\frac{1}{x_i} = \frac{1}{|x_i|^2} x_i^*$, the latter equation becomes

$$\hat{h}_{k-d+1,i} x_i = \hat{h}_{k-d,i} x_i + \frac{\lambda}{|x_i|^2} E_{x_i} a, \quad (10)$$

where, from (1) and (3),

$$a = y_{k-d+1} - \hat{h}_{k-d}^T x, \quad (11)$$

with $x$ being the lately detected transmitted vector.

Eqs. (11), (10) define the adaptation of the $i$th channel tap via the standard LMS or the normalized LMS algorithm, depending on whether $E_{x_i}$ equals $|x_i|^2$ or is given by eq. (9), respectively. The corresponding step size is $\mu = \lambda$ in both
cases. This remark provides us with guidelines on how $\lambda$ should be chosen in eq. (8). With QPSK signaling, where $|x_i|^2 = 2$ for all symbols, the center tracking technique of eq. (7) is also equivalent to the LMS algorithm, with step size $\mu = \lambda/2$. Note that the upper bound of $1/L$ for $\lambda$, mentioned with respect to eq. (7), agrees with the upper bound of $1/(2L)$ that is known to hold for $\mu$ with QPSK data [15].

V. PERFORMANCE EVALUATION

Fig. 4 shows the tracking performance of the new cluster tracking scheme, together with that of the LMS algorithm. The simplest variant of the cluster tracking scheme has been adopted, i.e., that of eq. (7), since QPSK signaling is employed. The curves are the result of the ensemble averaging of 1500 independent runs, where a Rayleigh fading channel was simulated consisting of 3 complex taps and corresponding to a vehicle speed of 120 km/h was employed. The symbol rate and carrier frequency were 200 Kbauds and 900 MHz, respectively. The root mean square (RMS) of the real and imaginary component of each path was set equal to 1. Both algorithms were initialized using 15 training symbols and the adaptation procedure was realized using the correct information symbols. The step sizes $\lambda$ and $\mu$ were set equal to their optimum values, 0.08 and 0.08/2=0.04, respectively. As expected (see Section IV-A), the steady state performance in both algorithms is the same. However, the cluster tracking technique exhibits better initialization behavior, due to the superior convergence performance (LS-like) of the center estimation (CE) method [9], [14].

The various approaches to the elimination of the estimation delay problem, outlined in the introductory section, can be directly incorporated in the CBSE equalizer. Thus, as far as the symbol error rate (SER) performance is concerned, we tested the 1-D CBSE equalizer with the center tracking being performed using 5 symbol delayed tentative decisions without predictor, with predictor, and with the PSP technique. Moreover, the finite memory Bayesian-DFE (FMBE) [12] was used as a benchmark. The latter equalizer was implemented with the aid of a radial basis function (RBF) neural network. The transmission was realized in data blocks comprising 200 information and 15 training symbols and the training symbols were positioned in the beginning of each data block.

VI. COMPUTATIONAL REQUIREMENTS

In general, when the MLSE equalizers are used with packet transmission, the probability of erroneous detection of the last $D-1$ symbols of the packet, where $D$ is the decision delay of the VA, is higher than that of the other symbols. This is due to the limited decision delay. This phenomenon is more intense in the case of frequency selective fading conditions, since it is possible for the first CIR taps to exhibit deep fading resulting in non-minimum phase channels. For the elimination of this effect, we have added $L-1$ guard symbols in the beginning of each data block.

Fig. 5 shows the performance curves of the various versions of the 1-D CBSE equalizer, as well as the Bayesian one and the 1-D CBSE with tracking based on tentative decisions with no guard symbols. The fading channel parameters were the same with those mentioned in the tracking example of Fig. 4 and the fading channel was left to evolve for 215 sec or equivalently for a time period of 43 million transmitted symbols. The performance of the LMS-based MLSE equalizer is not shown, since it is slightly worse than that of the 1-D CBSE, due to the better start up provided by the center estimation algorithm used for the initialization. In all the cases, except that of tentative decisions without guard symbols, the 1-D CBSE outperforms the FMBE. Moreover, the performance improvement due to the guard symbols is remarkable. However, the advantage of the guard interval adoption vanishes in the case of the tentative decisions tracking with higher vehicle speeds, e.g., 300 km/h. This happens because tentative decision tracking is inadequate with such a high vehicle speed, leading, in any case, to many erroneous decisions.

7The channel taps were simulated so that they vary independently of one another, via Smith’s fading simulator [16].

8Two guard symbols were used.
Table I summarizes the operations counts corresponding to (a) the propagation characteristics of the VA.

(Note: The quantities $A, B, C$ in (a) are zero for $M = 4$, whereas, for $M > 4$, $A = L(M + 4)$, $B = L(M/2 + 1)$, $C = 2L$.

<table>
<thead>
<tr>
<th>Method</th>
<th>MUL/DIV</th>
<th>ADD/SUB</th>
<th>$(\cdot \cdot)^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LMS-MLSE</td>
<td>1-D CBSE</td>
<td>LMS-MLSE</td>
</tr>
<tr>
<td>Initialization</td>
<td>$N_n(8L+2)$</td>
<td>$2(2L+1)$</td>
<td>$8N_nL$</td>
</tr>
<tr>
<td>Tentative decisions</td>
<td>$N(8L+2)$</td>
<td>$N(A+2)$</td>
<td>$8NL$</td>
</tr>
<tr>
<td>PSP</td>
<td>$N(8L+2)M^d-1$</td>
<td>$N(A+2)M^d-1$</td>
<td>$8NLM^d-1$</td>
</tr>
<tr>
<td>VA</td>
<td>$4NLM^d$</td>
<td>$0$</td>
<td>$2N(2L+1)M^d$</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Method</th>
<th>MUL/DIV</th>
<th>ADD/SUB</th>
<th>$(\cdot \cdot)^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LMS-MLSE</td>
<td>1-D CBSE</td>
<td>LMS-MLSE</td>
</tr>
<tr>
<td>Initialization</td>
<td>$1,260$</td>
<td>$22$</td>
<td>$1,200$</td>
</tr>
<tr>
<td>Tentative decisions</td>
<td>$8,400$</td>
<td>$400$</td>
<td>$8,000$</td>
</tr>
<tr>
<td>PSP</td>
<td>$2,150,400$</td>
<td>$102,400$</td>
<td>$2,048,000$</td>
</tr>
<tr>
<td>VA</td>
<td>$4,096,000$</td>
<td>$0$</td>
<td>$4,505,600$</td>
</tr>
</tbody>
</table>

(b)

estimation using the LMS algorithm or the tap-contributions estimation using the CE technique, b) the adopted tracking scheme and c) the computational requirements of the VA. Table Ia summarizes the operations counts corresponding to the above three parts in terms of real multiplications and additions for a data block consisting of $N_t$ training and $N$ information observations, with M-QAM signaling. The cluster tracking method of eq. (8) is employed, with $E_{x_i} = 1$ for the case of $M = 4$ and $E_{x_i} = |x_i|^2$ for $M > 4$. Note that the quantities $A, B, C$ in Table Ia only appear for larger than QPSK constellations ($M > 4$), where an extra computational effort is needed in order to compute the $M$ possible values of the tap contributions per channel tap. Indeed, in such a case, we need first to compute the factors $E_{x_i}$ for $i = 0, 1, \ldots, L-1$ (although these could be assumed to have been precomputed for all possible symbols $x_i$). Second, two more real multiplications per channel tap are required, for the products $E_{x_i} \cdot \lambda_a$ in eq. (8). Finally, for larger than QPSK constellations, simple $\pi/2$ rotations do not suffice to obtain, from one of the $M$ possible values of a tap contribution, all the rest. An efficient manner of computing these contribution values is suggested by the fact that the $M$ different QAM symbols can be grouped in $M/4$ subsets, each containing four symbols that are positioned at the corners of a square centered at the origin.\footnote{For example, if $M = 16$, then the $M/4 = 4$ subsets are: $S_1 = \{1 + j, 1 - j, -1 + j, -1 - j\}$, $S_2 = \{3 + 3j, 3 - 3j, -3 + 3j, -3 - 3j\}$, $S_3 = \{1 + 3j, 3 - j, -1 - 3j, -3 + j\}$, and $S_4 = \{1 - 3j, -3 - j, -1 + 3j, 3 + j\}$.} Hence, for each group of four, only one contribution needs to be computed for each tap, $h_i$, and the remaining three can be obtained by simple $\pi/2$ rotations. That is, one only needs to compute $M/4$ contribution values per tap. Once one of the $M/4$ contributions, $c_{x_i}^d$, has been obtained via eq. (8), the rest $M/4 - 1$ can be computed via the formula $c_{x_i}^d = x' \cdot (c_{x_i}^d / |x_i|^2)$. The complex division is needed only once per channel tap and the complex multiplication $M/4 - 1$ times per channel tap. Note that the quantities $|x_i|^2$ that are required in these complex divisions have already been computed as $E_{x_i}$. The above amounts to $A = L(M + 4)$ real multiplications, $B = L(M/2 + 1)$ real additions, and $C = 2L$ squaring operations per sample, in addition to those needed in the $M = 4$ case.

It should be emphasized that the number of multiplications and divisions required when the initialization is realized with the CE method is independent of the number of training symbols and is much lower than that required by LMS. Moreover, with respect to the third part of computations, the
1-D CBSE is a multiplications-free technique because of its emancipation from the need of computing $M^L$ convolutions per data symbol. Finally, the computational complexity of the tracking part, in all versions discussed in Section IV, is less than that of the equivalent LMS variants, in the QPSK case. When $M > 4$, the proposed tracking scheme is more complex computationally than the LMS tracking. However, this is counterbalanced by the drastic computational savings in the VA. For example, in the case of 256-QAM, with $L = 5$ taps and $N_{tr}$, $N_{tr}$ equal to 30 and 200 samples, respectively, the standard MLSE, with tentative decisions tracking, needs more than $4 \times 10^{15}$ multiplications and $4.8 \times 10^{15}$ additions, while at the same time the proposed equalizer needs only 260,422 multiplications per data packet and about the two thirds of the additions of the standard MLSE. Table Ib shows the number of operations required for the above values of $L$, $N_{tr}$, $N_{tr}$, and QPSK input. It can be seen that the 1-D CBSE needs, in the case of 256-QAM, much less multiplications than the standard MLSE requires for QPSK signaling.

VII. CONCLUSION

In this paper, a cluster-based sequence equalizer for time-varying transmission environments was proposed. The tracking of the cluster centers is realized via a novel scheme, which is equivalent in performance with the LMS tracking algorithm. However, now, this performance is obtained at a significantly reduced overall computational cost. The reduced complexity of the new equalization method is achieved by acting on the cluster centers directly, bypassing in this way the computation of convolutions and, moreover, by exploiting the symmetries of the symbol constellation.

REFERENCES


