On the Least-Squares Performance of a Novel Efficient Center Estimation Method for Clustering-Based MLSE Equalization

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

***Abstract*—Recently, a novel maximum-likelihood sequence estimation (MLSE) equalizer was reported that avoids the explicit estimation of the channel impulse response. Instead, it is based on the fact that the (noise-free) channel outputs, needed by the Viterbi algorithm, coincide with the points around which the received (noisy) samples are clustered and can thus be estimated directly with the aid of a supervised clustering method. Moreover, this is achieved in a computationally efficient manner that exploits the channel linearity and the symmetries underlying the transmitted signal constellation. The resulting computational savings over the conventional MLSE equalization scheme are significant even in the case of relatively short channels where MLSE equalization is practically applicable. It was demonstrated, via simulations, that the performance of this algorithm is close to that using a least-squares (LS) channel estimator, although its computational complexity is even lower than that of the least-mean squares (LMS)-trained MLSE equalizer. This paper investigates the relationship of the center estimation (CE) part of the proposed equalizer with the LS method. It is proved that, when using LS with the training sequence employed by CE, the two methods lead to the same solution. However, when LS is trained with random data, it outperforms CE, with the performance difference being proportional to the channel length. A modified CE method, called MCE, is thus developed, that attains the performance of LS with perfectly random data, while still being much simpler computationally than classical LS estimation. Through the results of this paper, CE is confirmed as a methodology that combines high performance, simplicity, and low computational cost, as required in a practical equalization task. An alternative, algebraic viewpoint on the CE method is also provided.**

***Index Terms*—Center estimation (CE), channel equalization, clustering, least-squares (LS), maximum-likelihood sequence estimation (MLSE).**

I. Introduction

ONE of the major problems encountered in the receiver design of any communication system is that of combatting intersymbol interference (ISI) arising due to limited channel bandwidth or multipath propagation. The part of the receiver used to mitigate ISI is the equalizer, and the related literature is very rich (see, e.g., [20] and [23]).

The equalizers based on the maximum-likelihood sequence estimation (MLSE) scheme [5], [20] are commonly implemented via the Viterbi algorithm (VA) [6], and they require the channel impulse response (CIR) to be known. For this purpose, one may resort to any appropriate identification method [11], [12]. Once the CIR has been identified, its inner products (convolution) with all possible channel input vectors (associated with the states) are computed and subsequently used in the metrics computations for the VA.

Recently, a novel MLSE equalizer was proposed, that circumvents the problem of *explicit* CIR parametric modeling, leading to substantial computational savings [13]–[17]. It belongs to the class of the so-called clustering-based sequence equalizers (CBSEs) (e.g., [7], [8], and [24]), since it is based on the idea that the set of all possible channel output values, needed at the Viterbi stage, are simply the *centers* of the clusters formed by the received observations at the receiver front end and can thus be estimated via a supervised clustering approach. In contrast to earlier CBSE methods, however, which appeal to clustering in a high-dimensional space, defined by successive observations,[1] the novel algorithm operates in a one-dimensional space [13]. Furthermore, it uses an efficient cluster center estimation (CE) technique that exploits the structural symmetries underlying the generation mechanism of the clusters of the received samples. This leads to a considerable reduction of the number of cluster centers required to be estimated directly from the training data. It turns out that the centers of all the M^L clusters, formed by the noisy output of a channel of length L with an input alphabet of size M, can be determined on the basis of estimates of *only* L properly selected ones. This has a twofold advantage. First, since only L clusters need to be learned, a considerably shorter sequence suffices for training, compared to previously proposed CBSE receivers. It is constructed so as to generate a *cyclic repetition* of only L input vectors, corresponding to the selected L clusters. Second, the computational complexity is drastically reduced.

It has been observed, via simulations [16], [17], that the proposed CE technique exhibits a similar to least-squares (LS) performance, despite its low computational complexity. This paper investigates this issue. It is shown that *CE yields the same estimates for the cluster centers that would result from computing*

Manuscript received June 3, 2004; revised May 3, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Franz Hlawatsch.

E. Kofidis is with the Department of Statistics and Insurance Science, University of Piraeus, 185 34 Piraeus, Greece (e-mail: kofidis@unipi.gr).

Y. Kopsinis is with the Institute for Digital Communications, School of Engineering and Electronics, University of Edinburgh, Edinburgh EH9 3JL, U.K. (e-mail: y.kopsinis@ed.ac.uk).

S. Theodoridis is with the Department of Informatics and Telecommunications, University of Athens, Panepistimioupolis, Ilissia, 157 84 Athens, Greece (e-mail: stheodor@di.uoa.gr).

Digital Object Identifier 10.1109/TSP.2006.870582

[1]This is also the case in symbol-by-symbol equalizers that are based on clustering (see, e.g., [2], [3], [19]).

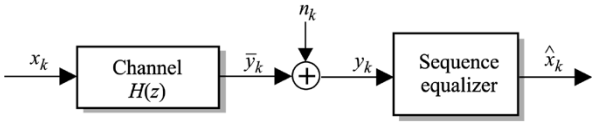


Fig. 1. Communication system model.

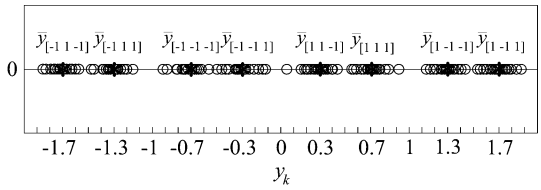


Fig. 2. Plot of the clusters formed by the received samples when BPSK symbols are transmitted, the transfer function of the channel is $H(z) = 1 - 0.5z^{-1} + 0.2z^{-2}$ and white Gaussian noise of SNR = 20 dB is present. Stars denote cluster centers, while circles correspond to noisy channel output samples.

them as convolution sums using the channel estimate provided by the LS method, trained on the same data with CE. Moreover, the computational requirements of the two methods are compared, clearly demonstrating the computational advantage of CE over the LS method. This comes from the fact that CE estimates the cluster centers directly, in an efficient manner that exploits constellation symmetries, and not via convolutions. On the other hand, simulation results show that if the LS method is employed with random, instead of periodic training data, it performs better compared to the CE method and this improvement increases as the CIR becomes longer. A mean-squared error performance analysis is carried out to theoretically justify this fact. The improvement resulting from the use of random data in the LS method is shown to be proportional to the channel length, L. It is shown that, in order to minimize this difference in performance, the CE method has to be appropriately modified so that it uses a periodic repetition of a longer (than L) training sequence, enjoying perfect randomness. Although this increases the computational demands, the new version of the CE algorithm [modified CE (MCE)] is still more economic than direct LS estimation in terms of computational complexity.

The rest of this paper is organized as follows. Section II describes the discrete-time model for the communication system and states the problem. The CE method is briefly reviewed in Section III. In Section IV a proof of least-squares optimality of the solution provided by CE is presented. An analysis of the observed performance hysteresis of CE with respect to LS, when the latter employs random training data, is provided in Section V. The modified CE method is presented and analyzed in Section VI. The methods are compared in terms of their computational complexity in Section VII. Section VIII presents the conclusions.

Notation. Vectors and matrices will be denoted by bold lower and upper case letters, respectively. The superscripts * and ^{T} will, respectively, stand for complex conjugation and transposition.

II. Description of the Communication System

A block diagram of the adopted discrete-time model for the communication system is depicted in Fig. 1. A sequence of independent and identically distributed (i.i.d.) symbols, x_k, drawn from a finite alphabet S of size M is transmitted through the channel, assumed to be invariant during the transmission of the training data and modeled as a finite impulse response filter of length L and transfer function $H(z)$. The input signal constellation is assumed *symmetric*. That is, M is even and S contains both the symbols $a_i, i = 1,2,\ldots,M/2$, and their negatives. This includes both real (e.g., M-PAM [17]) and complex (e.g., M-PSK ($M > 2$) and M-QAM [16]) constellations. For the sake of generality, the results will be presented for the complex case. In the examples, we will assume 2-PAM [binary phase-shift keyed (BPSK)] and 4-QAM [quadrature phase-shift keyed (QPSK)] signaling. The received sequence is given by

$$y_k = \sum_{i=0}^{L-1} h_i x_{k-i} + n_k = \boldsymbol{h}^T \boldsymbol{x}_k + n_k \equiv \bar{y}_k + n_k \qquad (1)$$

where $\boldsymbol{h} = [h_0, h_1, \ldots, h_{L-1}]^T$ is the vector of the (generally complex) symbol-spaced L taps of the CIR, $\boldsymbol{x}_k = [x_k, x_{k-1}, \ldots, x_{k-L+1}]^T$ is the input data vector, $\bar{y}_k$ is the noiseless channel output, and n_k is additive white noise, uncorrelated with $\bar{y}_k$. The noise is assumed zero-mean with variance σ^2. If it is complex, its real and imaginary components are assumed to be uncorrelated with each other and each one of variance $\sigma^2/2$. The noisy observations y_k are fed to the sequence estimator, whose aim is to provide estimates $\hat{x}_k$ of the input symbol sequence.

III. Center Estimation

As it is well known [20], in MLSE equalization one has to estimate the CIR first and then use this estimate in the VA [6] (or one of its variants) to obtain the required estimate of the symbol sequence $\boldsymbol{x}_k$, based on metric computations of the form $D_{k,\boldsymbol{x}} = |y_k - \boldsymbol{h}^T \boldsymbol{x}|^2$, with $\boldsymbol{x}$ ranging over the set of all possible L-tuples of input symbols.[2] However, it can be readily seen that what one really needs, instead, are the noiseless observations $\bar{y}_{\boldsymbol{x}} \equiv \boldsymbol{h}^T \boldsymbol{x}$, since $D_{k,\boldsymbol{x}} = |y_k - \bar{y}_{\boldsymbol{x}}|^2$. Hence, *the CIR estimation step can be bypassed* [13] by estimating the quantities $\bar{y}_{\boldsymbol{x}}$ directly. This has been introduced in [14]–[17], and its computational advantages have been demonstrated. The basic idea stems from the fact that all possible values which $\bar{y}_{\boldsymbol{x}}$ can take are simply the points (centers) around which the received samples (observations) y_k are clustered (due to the presence of the noise). These centers can be estimated via any supervised clustering technique. An example is given in Fig. 2, where the received samples for a 3-tap channel with transfer function $H(z) = 1 - 0.5z^{-1} + 0.2z^{-2}$ and BPSK input are shown. The noise is assumed to be white Gaussian, corresponding to an SNR of 20 dB. The notation $\bar{y}_{[x_k, x_{k-1}, x_{k-2}]}$ denotes the cluster center which is associated with the transmitted symbol sequence $\boldsymbol{x} = [x_k, x_{k-1}, x_{k-2}]^T$ at time k. The spread of the clusters depends on the power of the noise. The number of clusters as well as their position on the real line depend on the number and the values of the CIR taps.

In total, there are M^L cluster centers that have to be estimated. This number evaluates to $2^3 = 8$ for the example of

[2]Forney's MLSE scheme [5] is adopted here.

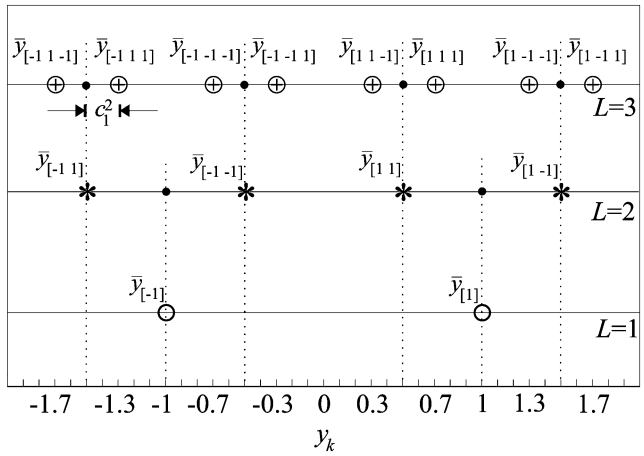


Fig. 3. $\oplus$'s denote cluster centers that correspond to the 3-tap channel $H(z) = 1 - 0.5z^{-1} + 0.2z^{-2}$ with BPSK input. The centers of the clusters that are due only to the first (and the second) tap are shown as o's ($*$'s).

Fig. 2 and can easily grow very large in practical situations (e.g., $M^L = 1\,024$ for a 5-tap channel with 4-PAM input). Nevertheless, a deeper thought reveals that there is a lot of redundant information shared by the cluster centers, which can be exploited for the benefit of the computational complexity. This is due to the intrinsic dependency between the locations of the cluster centers, caused by the linearity of the channel as well as the symmetry of the symbol alphabet. Fig. 3 depicts the centers of the clusters formed by the output samples of the channels consisting of only the first tap (o's), the two first taps ($*$'s) and all 3 taps ($\oplus$'s) of the channel of Fig. 2. The pair of the centers due to the first tap is symmetrically located around zero. The four centers resulting from the combination of the first and the second taps can be grouped in two pairs, each of them being symmetrically located around the (previous) centers associated with the first tap. This rationale carries on for the centers generated when all 3 taps of the channel act together.[3] In other words, *the locations of the observations follow a hierarchical pattern of symmetries*, depending on the channel length.

It has been shown [15], [17] that this structure in the cluster center constellation, demonstrated by the above example, implies that *only* L (out of the M^L) properly selected cluster centers need to be estimated directly from the training sequence. The rest $M^L - L$ of them can then be easily determined using the obtained estimates of these L selected centers. To explain the method in some more detail, observe that

$$c_x^m = xh_m, \quad x \in S \tag{2}$$

is the *contribution* of the mth tap of the CIR in the summation part of (1). That is, c_x^m is the contribution of the mth tap to the generation of a cluster center. Using this notation, the term $\bar{y}_k$ in (1) can be rewritten as

$$\bar{y}_{[x_k,x_{k-1},\ldots,x_{k-L+1}]} = \sum_{m=0}^{L-1} c_{x_{k-m}}^m \tag{3}$$

[3]The reader is referred to [16] for an analogous example of a complex channel with two-dimensional (QPSK) input.

where $\bar{y}_{[x_k,x_{k-1},\ldots,x_{k-L+1}]}$ is the cluster center associated with the transmitted L-tuple $[x_k, x_{k-1}, \ldots, x_{k-L+1}]$. Furthermore, it is easy to realize that, for each tap h_m, only one of the M possible values of c_x^m needs to be computed; all the others can be obtained via a simple multiplication, as follows:

$$c_{x'}^m = \left(\frac{x'}{x}\right) \cdot c_x^m. \tag{4}$$

For BPSK signaling, i.e., $S = \{-1, 1\}$, the above translates to a simple sign change ($c_{-1}^m = -c_1^m$), whereas in the QPSK case $\pi/2$ rotations are needed, e.g., $c_{1-j}^m = -jc_{1+j}^m, c_{-1-j}^m = j^2 c_{1+j}^m = -c_{1+j}^m, c_{-1+j}^m = jc_{1+j}^m$. Therefore, *the computation of all the cluster centers* (via (3)) *requires the estimation of only* L tap contributions c_x^m *one for each channel tap*, which, as it will be seen shortly, are in turn computed via the estimates of only L properly selected centers.

A method for the appropriate selection of the centers which have to be estimated directly from the training data was proposed in [14]–[17]. First, choose any one of the M^L centers, say $\bar{y}_{[x_0,x_1,\ldots,x_{L-1}]}$. We call it the *basic center*, $\bar{y}_{\text{basic}}$, and the corresponding L-tuple the *basic sequence*, $\boldsymbol{x}_{\text{basic}} = [x_0, x_1, \ldots, x_{L-1}]$.[4] Then the L centers to be estimated directly are chosen as those which correspond to the basic sequence with a sign change in one of its entries: $\bar{y}^0 \equiv \bar{y}_{[-x_0,x_1,\ldots,x_{L-1}]}, \bar{y}^1 \equiv \bar{y}_{[x_0,-x_1,\ldots,x_{L-1}]}, \ldots, \bar{y}^{L-1} \equiv \bar{y}_{[x_0,x_1,\ldots,-x_{L-1}]}$. These L centers can be estimated via any supervised clustering algorithm [25]. For example, a simple averaging of all the observations $y_k^{(m)}$ that belong to the corresponding cluster m was proposed in [14]–[17]

$$\hat{\bar{y}}^m = \frac{1}{N^{(m)}} \sum_{k=0}^{N^{(m)}-1} y_k^{(m)} \tag{5}$$

where $N^{(m)}$ denotes the number of observations associated with the mth cluster center $\bar{y}^m$. Once estimates $\hat{\bar{y}}^m$ for $\bar{y}^m, m = 0, 1, \ldots, L-1$ have been computed, $\bar{y}_{\text{basic}}$ is estimated as [14], [15]

$$\hat{\bar{y}}_{\text{basic}} = \frac{\sum_{m=0}^{L-1} \hat{\bar{y}}^m}{L-2}, \quad L > 2. \tag{6}$$

The computation of the tap contributions is then straightforward [14], [15]

$$\hat{c}_{x_m}^m = \frac{\hat{\bar{y}}_{\text{basic}} - \hat{\bar{y}}^m}{2}, \quad m = 0, 1, \ldots, L-1. \tag{7}$$

Using (6) in (7), an equivalent formula for $\hat{c}_{x_m}^m$ results as follows:

$$\hat{c}_{x_m}^m = \frac{1}{2(L-2)} \left[\sum_{i \neq m} \hat{\bar{y}}^i - (L-3)\hat{\bar{y}}^m \right]. \tag{8}$$

[4]The index in x_i here does not represent (absolute) time.

For the example of Fig. 3, one can set, e.g., $\boldsymbol{x}_{\text{basic}} = [1,1,1]$. Then, it suffices to estimate the $L = 3$ centers $\bar{y}^0 = \bar{y}_{[-1,1,1]}, \bar{y}^1 = \bar{y}_{[1,-1,1]}$, and $\bar{y}^2 = \bar{y}_{[1,1,-1]}$. The tap contributions are then determined as[5] $\hat{c}_1^0 = (\hat{\bar{y}}^1 + \hat{\bar{y}}^2)/2, \hat{c}_1^1 = (\hat{\bar{y}}^0 + \hat{\bar{y}}^2)/2$, and $\hat{c}_1^2 = (\hat{\bar{y}}^0 + \hat{\bar{y}}^1)/2$. For a 3-tap channel with QPSK input, a possible choice for the basic sequence is $\boldsymbol{x}_{\text{basic}} = [1+j, 1+j, 1+j]$. Then the centers that have to be estimated are $\bar{y}^0 = \bar{y}_{[-1-j,1+j,1+j]}, \bar{y}^1 = \bar{y}_{[1+j,-1-j,1+j]}$, and $\bar{y}^2 = \bar{y}_{[1+j,1+j,-1-j]}$, and the corresponding tap contributions are again computed as above.

The above procedure does not apply when $L = 2$ (cf. (6)). A different approach must be taken in this case. This is presented in the Appendix, where the 2-tap channel case is separately treated. In the rest of the paper, it will be assumed that $L > 2$.

The above method for estimating the M^L cluster centers, based on an averaging procedure of direct estimation of only L of them, will be referred to, simply, as the *CE algorithm*. If the employed training sequence is to be as short and effective as possible, it has to be chosen so that it "visits" the selected clusters (i.e., generates the corresponding input vectors) as many times as possible and equally often. It is readily seen that, if *only* the L-tuples corresponding to the centers $\bar{y}^m$ are to appear in the training sequence, the symbols in the basic sequence should coincide, i.e.,

$$x_0 = x_1 = \cdots = x_{L-1} = x. \tag{9}$$

Such a choice of training sequence for the BPSK scheme, and for $\boldsymbol{x}_{\text{basic}} = [1,1,\ldots,1]$, can be the periodic repetition[6] of the sequence (for $L > 2$) $[\underbrace{1,1,1,\ldots,1}_{L-1},-1]$ [15]. This gives rise to L possible input data vectors, which appear at the rows of the matrix

$$\boldsymbol{X}_1^T = \begin{bmatrix} -1 & 1 & \cdots & 1 \\ 1 & -1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & -1 \end{bmatrix}. \tag{10}$$

In practice, when input symbols are drawn from an alphabet S, one may construct a training sequence by simply multiplying the above sequence, used for the BPSK scheme, with a selected symbol, $x \in S$.[7] The corresponding data matrix will then equal x times the matrix given in (10). In the sequel, it will be seen that, as expected, x should be chosen so that its modulus is as high as possible.

[5]When $L = 3$, only two of the three centers are used for the estimation of each tap contribution. This point needs special attention and will be further elaborated upon in Section VI.

[6]A periodic training sequence is employed here with period equal to the channel length, as in the *periodic training* approach [21], [22] followed in *cyclic equalization* [1, pp. 380–383]. However, the special structure and role (visiting particular centers) of the sequence used here, as well as the clustering flavor of the method, differentiate CE from the periodic training approach.

[7]For example, with QPSK signaling and $x = 1 + j$, the training sequence corresponding to $\boldsymbol{x}_{\text{basic}} = [1+j, 1+j, \ldots, 1+j]$ will be constructed by periodically repeating the sequence $[\underbrace{1+j, 1+j, 1+j, \ldots, 1+j}_{L-1}, -1-j]$.

IV. Equivalence With LS Estimation

A. CE as a Channel Identification Method

More generally, if the basic sequence is $\boldsymbol{x}_{\text{basic}} = [x_0, x_1, \ldots, x_{L-1}]$ (with $L > 2$), the corresponding $L \times L$ matrix keeping the training L-tuples at its rows is of the form

$$\begin{aligned} \boldsymbol{X}^T &= \begin{bmatrix} -x_0 & x_1 & \cdots & x_{L-1} \\ x_0 & -x_1 & \cdots & x_{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_0 & x_1 & \cdots & -x_{L-1} \end{bmatrix} \\ &= \boldsymbol{X}_1^T \operatorname{diag}(x_0, x_1, \ldots, x_{L-1}) \end{aligned} \tag{11}$$

where $\boldsymbol{X}_1$ is given by (10) and $\operatorname{diag}(\boldsymbol{x})$ denotes the diagonal matrix with the vector $\boldsymbol{x}$ on its main diagonal. Then the L centers that are estimated directly from the training data are given by

$$\bar{\boldsymbol{y}} \equiv [\bar{y}^0 \quad \bar{y}^1 \quad \cdots \quad \bar{y}^{L-1}]^T = \boldsymbol{X}^T \boldsymbol{h}. \tag{12}$$

Denote by $\hat{\bar{\boldsymbol{y}}} \equiv [\hat{\bar{y}}^0 \quad \hat{\bar{y}}^1 \quad \cdots \quad \hat{\bar{y}}^{L-1}]^T$ the estimates for these L centers, resulting from the CE algorithm. It will be shown that they are LS optimal. In other words, the result is the same as if one had first identified the CIR via the LS method, using the data in (11) as the training sequence, and then use the estimated CIR to estimate all possible (noise-free) channel outputs, $\bar{y}_k$, via convolutions. For the purposes of this analysis, and only in order to have a common testbed for the comparison of the CE and LS methods, we shall view CE as a method of channel identification and consider the equivalent CIR estimate (although this is not needed when using the CE method[8]). This is indeed possible, in view of the relation between the tap contributions (estimated by CE) and the channel taps, namely, $c_{x_m}^m = x_m h_m$ (cf. (2)). If

$$\hat{\boldsymbol{c}} = [\hat{c}_{x_0}^0, \hat{c}_{x_1}^1, \ldots, \hat{c}_{x_{L-1}}^{L-1}]^T$$

is the vector of tap contribution estimates, an estimate for the CIR vector can be computed as

$$\hat{\boldsymbol{h}} = \operatorname{diag}(x_0^{-1}, x_1^{-1}, \ldots, x_{L-1}^{-1}) \hat{\boldsymbol{c}}. \tag{13}$$

Equation (12) implies that the estimated taps will be related to the estimated centers via the relation

$$\hat{\bar{\boldsymbol{y}}} = \boldsymbol{X}^T \hat{\boldsymbol{h}} \tag{14}$$

which, in view of (11) and (13), can be written as

$$\hat{\bar{\boldsymbol{y}}} = \boldsymbol{X}_1^T \hat{\boldsymbol{c}}$$

or equivalently ($\boldsymbol{X}_1$ is nonsingular)

$$\hat{\boldsymbol{c}} = \boldsymbol{X}_1^{-T} \hat{\bar{\boldsymbol{y}}}. \tag{15}$$

[8]CE estimates directly the cluster centers and not the CIR taps. The channel taps enter only *implicitly* in the tap contributions that are computed from the L directly estimated centers as a means for the computation of the remaining $M^L - L$ ones.

Noting that

$$\boldsymbol{X}_1^{-T} = \frac{1}{2(L-2)} \times \begin{bmatrix} -(L-3) & 1 & \cdots & 1 \\ 1 & -(L-3) & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & -(L-3) \end{bmatrix} \quad (16)$$

it is seen that (8) is, in fact, a computationally efficient manner of implementing the matrix-vector multiplication in (15).

We will now show how the mean-squared error (MSE) in the estimation of the centers is related to that of the channel taps. If $\bar{\boldsymbol{X}}^T$ is the $M^L \times L$ matrix containing at its rows all of the M^L L-tuples of symbols from S, the vector of all M^L center estimates, call it $\hat{\bar{\boldsymbol{y}}}_{M^L}$, will be given by

$$\hat{\bar{\boldsymbol{y}}}_{M^L} = \bar{\boldsymbol{X}}^T \hat{\boldsymbol{h}}$$

and its covariance matrix will be equal to

$$\mathrm{cov}(\hat{\bar{\boldsymbol{y}}}_{M^L}) = \bar{\boldsymbol{X}}^T \mathrm{cov}(\hat{\boldsymbol{h}}) \bar{\boldsymbol{X}}^*.$$

Recalling that $S = \{a_i; i = 1, 2, \ldots, M/2\} \cup \{-a_i; i = 1, 2, \ldots, M/2\}$, it is readily verified that $\bar{\boldsymbol{X}}$ satisfies the relation[9]

$$\bar{\boldsymbol{X}}^* \bar{\boldsymbol{X}}^T = 2M^{L-1} \sum_{i=1}^{M/2} |a_i|^2 \boldsymbol{I}_L = M^L P_S \boldsymbol{I}_L$$

where $\boldsymbol{I}_n$ denotes the nth-order identity matrix and

$$P_S \equiv \frac{1}{M/2} \sum_{i=1}^{M/2} |a_i|^2 \quad (17)$$

is the input signal power. Hence, since the rows of $\bar{\boldsymbol{X}}^T$ are equally probable, the average MSE for the estimation of a center $\hat{\bar{y}}$ will be given by[10]

$$\begin{aligned} E[\mathrm{var}(\hat{\bar{y}})] &= \frac{1}{M^L} \mathrm{Trace}[\mathrm{cov}(\hat{\bar{\boldsymbol{y}}}_{M^L})] \\ &= \frac{1}{M^L} \mathrm{Trace}[\bar{\boldsymbol{X}}^T \mathrm{cov}(\hat{\boldsymbol{h}}) \bar{\boldsymbol{X}}^*] \\ &= \mathrm{Trace}\left[\mathrm{cov}(\hat{\boldsymbol{h}}) \frac{\bar{\boldsymbol{X}}^* \bar{\boldsymbol{X}}^T}{M^L}\right] \\ &= P_S \mathrm{Trace}[\mathrm{cov}(\hat{\boldsymbol{h}})] \\ &= P_S \sum_{m=0}^{L-1} \mathrm{var}(\hat{h}_m). \end{aligned} \quad (18)$$

[9]Note that each of the symbols $a_i, -a_i, i = 1, 2, \ldots, M/2$ appears $M^L/M = M^{L-1}$ times in each of the columns of $\bar{\boldsymbol{X}}^T$. Moreover, the columns of this matrix are orthogonal.

[10]Note that both CE and LS provide unbiased estimates, i.e., $E[\hat{\boldsymbol{h}}] = \boldsymbol{h}$.

B. Proof of Equivalence of CE With the LS Method

Let $y_0, y_1, \ldots, y_{N-1}$ be the received samples (observations) when the rows of $\boldsymbol{X}^T$ are periodically used as input data vectors, with y_k resulting from row $k \bmod L$.[11] Write the number of observations as $N = n_0 L + n_1$, where $0 \le n_1 < L$. This implies that centers $\bar{y}^m, m = 0, 1, \ldots, n_1 - 1$ have used one more sample for their estimation than the rest (which have used n_0 samples each). Since $\hat{\bar{y}}^m$ is determined as the average of those observations that belong to the mth cluster, it can be written as

$$\hat{\bar{y}}^m = \begin{cases} \frac{1}{n_0+1} \sum_{n=0}^{n_0} y_{nL+m}, & 0 \le m \le n_1 - 1 \\ \frac{1}{n_0} \sum_{n=0}^{n_0-1} y_{nL+m}, & n_1 \le m \le L-1 \end{cases}$$

or, equivalently, in matrix-vector form

$$\hat{\bar{\boldsymbol{y}}} = \left[\underbrace{\boldsymbol{\mathcal{I}} \quad \boldsymbol{\mathcal{I}} \quad \cdots \quad \boldsymbol{\mathcal{I}}}_{n_0} \quad \vdots \quad \begin{matrix} \frac{1}{n_0+1} \boldsymbol{I}_{n_1} \\ \boldsymbol{0} \end{matrix} \right] \boldsymbol{y} \equiv \boldsymbol{\mathcal{A}} \boldsymbol{y} \quad (19)$$

where

$$\boldsymbol{\mathcal{I}} \equiv \begin{bmatrix} \frac{1}{n_0+1} \boldsymbol{I}_{n_1} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{n_0} \boldsymbol{I}_{L-n_1} \end{bmatrix} \quad (20)$$

and $\boldsymbol{y} \equiv [y_0, y_1, \cdots, y_{N-1}]^T$. The estimate for the tap vector, call it $\hat{\boldsymbol{h}}^{\mathrm{CE}}$, results then via (11), (13), and (15), as follows:

$$\hat{\boldsymbol{h}}^{\mathrm{CE}} = \boldsymbol{X}^{-T} \hat{\bar{\boldsymbol{y}}} \quad (21)$$

$$= \boldsymbol{X}^{-T} \boldsymbol{\mathcal{A}} \boldsymbol{y}. \quad (22)$$

The corresponding problem for the LS estimation of the CIR can be formulated as

$$\min_{\hat{h}} \|\boldsymbol{y} - \boldsymbol{\mathcal{X}}^T \hat{\boldsymbol{h}}\|_2^2 \quad (23)$$

where the training data matrix $\boldsymbol{\mathcal{X}}^T$ is built as

$$\boldsymbol{\mathcal{X}}^T \equiv [\underbrace{\boldsymbol{X} \quad \boldsymbol{X} \quad \cdots \quad \boldsymbol{X}}_{n_0} \boldsymbol{X}_{n_1}]^T \quad (24)$$

with $\boldsymbol{X}_{n_1}$ denoting the first n_1 columns of $\boldsymbol{X}$. The solution to (23) is [11], [4]

$$\hat{\boldsymbol{h}}^{\mathrm{LS}} = (\boldsymbol{\mathcal{X}}^* \boldsymbol{\mathcal{X}}^T)^{-1} \boldsymbol{\mathcal{X}}^* \boldsymbol{y} \equiv \boldsymbol{\Phi}^{-1} \boldsymbol{z} \quad (25)$$

where

$$\boldsymbol{\Phi} \equiv \boldsymbol{\mathcal{X}}^* \boldsymbol{\mathcal{X}}^T \quad (26)$$

is the sampled data autocorrelation matrix and

$$\boldsymbol{z} \equiv \boldsymbol{\mathcal{X}}^* \boldsymbol{y} \quad (27)$$

[11]y_k's are not necessarily contiguous in time. Such would be the case if only the particular L-tuples appeared in the training sequence. Then (9) would hold and the training sequence should be constructed by periodically repeating $[\underbrace{x, x, \ldots, x}_{L-1}, -x]$.

denotes the sampled cross-correlation vector. Noting that

$$\begin{aligned}\boldsymbol{\Phi} &= \boldsymbol{X}^* \begin{bmatrix} (n_0+1)\boldsymbol{I}_{n_1} & \boldsymbol{0} \\ \boldsymbol{0} & n_0\boldsymbol{I}_{L-n_1} \end{bmatrix} \boldsymbol{X}^T \\ &= \boldsymbol{X}^*\boldsymbol{\mathcal{I}}^{-1}\boldsymbol{X}^T \end{aligned} \tag{28}$$

(25) becomes

$$\begin{aligned}\hat{\boldsymbol{h}}^{\mathrm{LS}} &= \boldsymbol{X}^{-T}\boldsymbol{\mathcal{I}}(\boldsymbol{X}^*)^{-1}\boldsymbol{\mathcal{X}}^*\boldsymbol{y} \\ &= \boldsymbol{X}^{-T}\boldsymbol{\mathcal{I}}\begin{bmatrix} \boldsymbol{I}_L & \boldsymbol{I}_L & \cdots & \boldsymbol{I}_L & \begin{matrix}\boldsymbol{I}_{n_1} \\ \boldsymbol{0}\end{matrix} \end{bmatrix}\boldsymbol{y} \\ &= \boldsymbol{X}^{-T}\boldsymbol{\mathcal{A}}\boldsymbol{y} \end{aligned} \tag{29}$$

which is identical with the equivalent CIR estimate that would result from the CE method (cf. (22).

It is of interest to note that the above proof applies to any $L\times L$ matrix $\boldsymbol{X}^T$ with linearly independent rows, not necessarily of the form of (11).[12] It thus provides an alternative viewpoint on the LS method through its relationship with averaging for cluster center estimation.

V. Mean-Squared Error Performance Analysis

The results of the previous section show that CE is equivalent with the LS method when L input vectors of the specific form (11) are repeatedly presented to the channel. Note that this type of training sequence is required in order that the CE method have its very attractive computational simplicity [16], [17].[13] However, the CE method is outperformed by LS when the latter employs randomly selected L-tuples of training symbols. Examples are shown in Fig. 4(a) and (b) for BPSK input and channels of length $L = 5$ and $L = 7$, respectively. White Gaussian noise of SNR $= 20$ dB was added to the channel output. In each case, estimates of 500 independent, randomly generated CIR's were computed using both the CE algorithm with the training data of the form of (10) and LS with random training data and the results were ensemble averaged. The average mean-squared tap-estimation error (MSE), $E[\|\hat{\boldsymbol{h}} - \boldsymbol{h}\|^2]/L$, is plotted.[14] For comparison purposes, the MSE learning curve of the least-mean squares (LMS) algorithm [11], [12] trained with random data and using a step size of[15] 0.01, is also included, along with the curve for the theoretically minimum MSE, i.e., σ^2/N [4]. It is apparent that the discrepancy in the obtained performance of the LS method between the two training data sets grows with the CIR length. To explain this phenomenon, let us have a look at the MSE incurred in the two cases. As it is well known, this is given by [11]

$$\mathrm{cov}(\hat{\boldsymbol{h}}) = \sigma^2 E[\boldsymbol{\Phi}^{-1}] \tag{30}$$

[12]It is only for matrices of this form, however, that (21) corresponds to the CE method.

[13]Observe that all but one of the entries in each row of the matrix in (16) coincide. This results in significant computational savings.

[14]Note that, as before and for the sake of the comparison, CE is viewed here as a channel estimation method.

[15]This value was chosen so that an acceptable MSE is obtained in a reasonable number of iterations. A smaller step size, chosen so as to minimize the misadjustment at the end of the training data block, would result in a poor MSE (compared to LS) for 50, 100 or even 150 training samples.

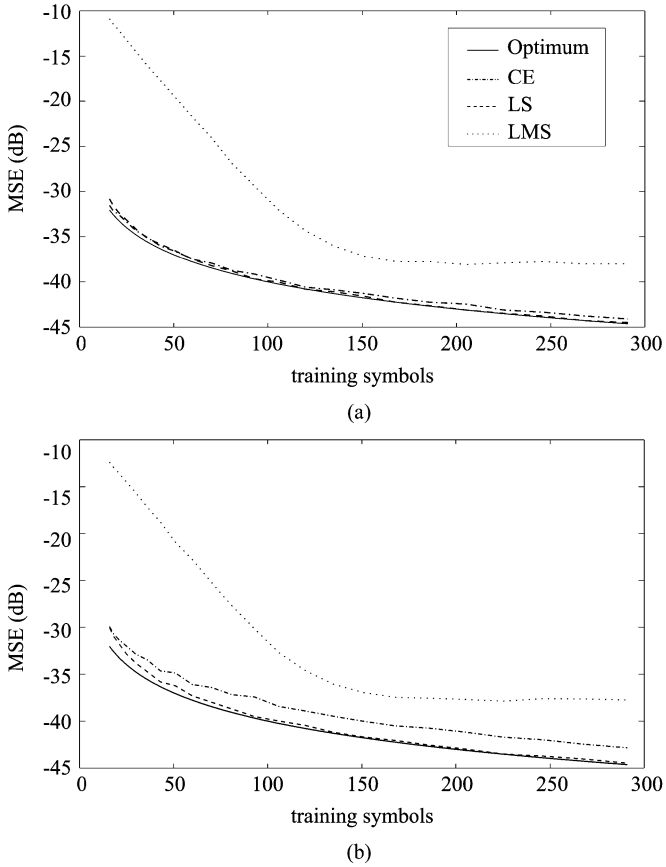


Fig. 4. Mean-squared tap-estimation error (MSE) curves resulting from CE (training with data (10)) and LS (random training sequence) for channels of length (a) $L = 5$ and (b) $L = 7$. The LMS and optimal (σ^2/N) curves are also included.

where σ^2 is the noise variance. For random data, $\boldsymbol{\Phi}$ tends to $NP_S\boldsymbol{I}_L$ as the number, N, of observations grows sufficiently large. In this case, (30) becomes $\mathrm{cov}(\hat{\boldsymbol{h}}) = (\sigma^2/NP_S)\boldsymbol{I}_L$. Hence, the MSE (i.e., variance) for each tap estimated by the LS method with random data is

$$\mathrm{var}\left(\hat{h}_m^{\mathrm{LS}}\right) = \frac{\sigma^2}{NP_S}. \tag{31}$$

To compute the MSE resulting from the CE method using the data in (10), one may use (8), recalling that $\hat{c}_{x_m}^m = x_m\hat{h}_m^{\mathrm{CE}}$ and using the fact that the center estimates $\hat{\bar{y}}^m, m = 0, 1, \ldots, L-1$ are uncorrelated with each other due to the noise being white. Thus

$$\mathrm{var}\left(\hat{c}_{x_m}^m\right) = \frac{1}{4(L-2)^2}\left[\sum_{i\neq m}\mathrm{var}(\hat{\bar{y}}^i) + (L-3)^2\mathrm{var}(\hat{\bar{y}}^m)\right]. \tag{32}$$

Moreover, since each center is estimated by averaging the (uncorrelated) observations associated with it, we can write

$$\mathrm{var}(\hat{\bar{y}}^i) = \begin{cases} \frac{\sigma^2}{n_0+1}, & 0 \le i \le n_1 - 1 \\ \frac{\sigma^2}{n_0}, & n_1 \le i \le L-1 \end{cases}.$$

Using the latter in (32) yields the first equation shown at the bottom of the page; hence, we have (33), shown at the bottom of the page. Note that these can also be derived via (30) by computing the diagonal entries of the matrix $\mathbf{\Phi}^{-1}$, with the aid of (28), for the training data of the form of (11).

When N is an integer multiple of L, i.e., $n_1 = 0$ and $n_0 = N/L$ (or when N is sufficiently large so that n_1 can be neglected), (33) becomes

$$\text{var}\left(\hat{h}_m^{\text{CE}}\right) = \frac{1}{|x_m|^2}\frac{[(L-3)^2+L-1]L}{4(L-2)^2}\cdot\frac{\sigma^2}{N} \tag{34}$$

$$= \left(\frac{P_S}{|x_m|^2}\cdot\frac{L}{4}\right)\cdot\frac{(L-3)^2+L-1}{(L-2)^2} \times \text{var}\left(\hat{h}_m^{\text{LS}}\right), \quad m = 0,1,\ldots,L-1. \tag{35}$$

Notice that the tap MSE that would result from CE is, in general, different for different taps, depending on the modulus squared of the corresponding symbol in the basic sequence. In the more practical case where all x_m are equal (cf. (9)), all taps are estimated with the same accuracy. *In the rest of the paper, we will only consider training sequences consisting of a symbol x, and its negative*. From the last equation, it follows that the use of cyclically repeating L training data vectors, as in (9) and (11), although sufficient to visit all L selected clusters N/L times, yields estimation performance roughly $(P_S/|x|^2)\cdot(L/4)$ times lower than that achievable with randomly chosen training data. This suggests that symbols of the highest possible modulus must be used in the training sequence. Then the factor $P_S/|x|^2$ will, in general, be less than one. For example, for 4-PAM input, where $S = \{-3,-1,1,3\}, x$ should be chosen as 3 or -3, in which case $P_S/|x|^2$ will equal $(5/9)$. With 16-QAM, where $S = \{\pm 1 \pm j, \pm 1 \pm 3j, \pm 3 \pm j, \pm 3 \pm 3j\}, x$ should be one of $\pm 3 \pm 3j$, corresponding again to $P_S/|x|^2 = (5/9)$.[16] The MSE of the LS method can, however, be reduced further if a random sequence consisting only of the symbols x and $-x$ is employed. Then (31) takes the form $\text{var}(\hat{h}_m^{\text{LS}}) = \sigma^2/(N|x|^2)$ and the factor in parentheses in (35) becomes $L/4$. To be more precise, it can be shown that $(L-1)/4$ is an asymptote, as $L \to \infty$, of $\text{var}(\hat{h}_m^{\text{CE}})/\text{var}(\hat{h}_m^{\text{LS}})$. Nevertheless, it must be noted that *the loss in performance with respect to LS is little*, for the range of CIR lengths and the number of training symbols used in practice with MLSE equalizers, as this is also verified by Fig. 4(a) and (b), especially in the range of less than 50 training symbols.

[16]In practice, of course, one would choose x to be a symbol of the highest possible modulus while at the same time not exceeding the transmission power constraints.

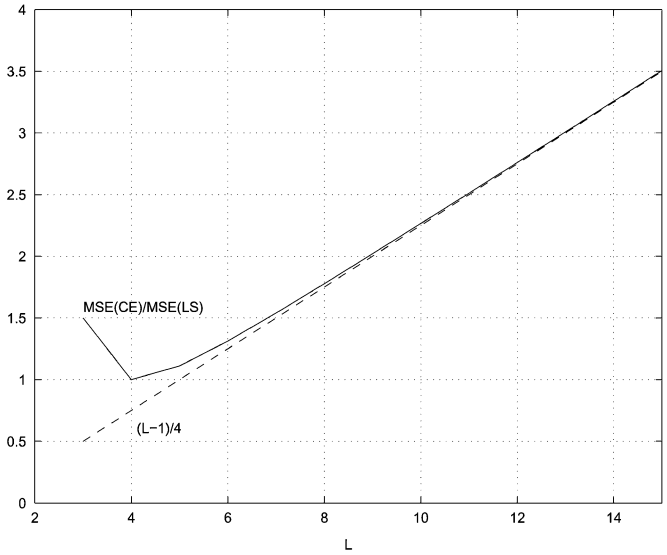


Fig. 5. Ratio $\text{var}(\hat{h}_i^{\text{CE}})/\text{var}(\hat{h}_i^{\text{LS}})$ as a function of L (solid line). CE is using a periodically repeated short (length L) training sequence, whereas random data are employed by the LS method. The number of observations is a multiple of L. The asymptote $(L-1)/4$ is also shown (dashed line).

Fig. 5 summarizes the results of the above analysis. Observe that CE performs as well as LS with random data when $L = 4$. In this case, (8) becomes: $\hat{c}_x^m = (\sum_{i\neq m}\hat{\bar{y}}^i - \hat{\bar{y}}^m)/4$. That is, each tap contribution is computed as the average of all four centers $\hat{\bar{y}}^i, i = 0,1,2,3$, with a sign change in one of them. When the number of observations N is a multiple of the number of centers, i.e., $N = 4n_0$, this results in the same MSE, for the corresponding channel tap, as the LS method would result with random data: $\text{var}(\hat{h}_m) = (1/|x|^2)\text{var}(\hat{c}_x^m) = (1/|x|^2)(4(\sigma^2/n_0)/4^2) = \sigma^2/(|x|^2N)$. Another way to see this is by observing that, in this case, the training data used in the CE method are *perfectly random* [4], in the sense that the corresponding sample autocorrelation matrix $\mathbf{\Phi}$ is *diagonal*, $\mathbf{\Phi} = N|x|^2\boldsymbol{I}_4$, when N is a multiple of 4. This is because, for $L = 4$, the matrix (10) is orthogonal,[17] $\boldsymbol{X}_1\boldsymbol{X}_1^T = 4\boldsymbol{I}_4$. Hence, (15) becomes $\hat{\boldsymbol{c}}_x = (1/4)\boldsymbol{X}_1\hat{\bar{\boldsymbol{y}}}$, yielding the *tap contributions as averages of the estimates of the L centers $\bar{y}^i$ or their negatives*.

[17]This conclusion can also be drawn by inspection of (16).

$$\text{var}\left(\hat{c}_{x_m}^m\right) = \begin{cases} \frac{1}{4(L-2)^2}\left[(n_1-1)\frac{\sigma^2}{n_0+1} + (L-n_1)\frac{\sigma^2}{n_0} + (L-3)^2\frac{\sigma^2}{n_0+1}\right], & 0 \le m \le n_1-1 \\ \frac{1}{4(L-2)^2}\left[n_1\frac{\sigma^2}{n_0+1} + (L-n_1-1)\frac{\sigma^2}{n_0} + (L-3)^2\frac{\sigma^2}{n_0}\right], & n_1 \le m \le L-1 \end{cases}$$

$$\text{var}(\hat{h}_m^{\text{CE}}) = \frac{1}{|x_m|^2}\begin{cases} \frac{\sigma^2}{4(L-2)^2}\left[\frac{(L-3)^2+n_1-1}{n_0+1} + \frac{L-n_1}{n_0}\right], & 0 \le m \le n_1-1 \\ \frac{\sigma^2}{4(L-2)^2}\left[\frac{n_1}{n_0+1} + \frac{(L-3)^2+L-n_1-1}{n_0}\right], & n_1 \le m \le L-1 \end{cases} \tag{33}$$

When $L = 3$, as it can be seen from (8), each tap contribution is also computed as the average of centers that are estimated directly from the training observations. However, only two of the three centers are employed for each contribution. This explains why the CE method performs in this case worse than for $L = 4$.

VI. Modified CE (MCE) Approach

It is clear from the above discussion that the orthogonality of the data matrix is crucial for the CE method to attain the performance offered by LS with random data. However, it can be easily verified that only for $L = 4$ the matrix $\boldsymbol{X}_1^T$ in (10) is orthogonal.[18] Orthogonality of (the columns of) the data matrix can be achieved for other channel lengths as well if it is allowed to be *tall*, that is, if more than L distinct cluster centers are estimated directly from the training data. Thus, a training sequence, consisting of the symbols x and -x, is needed that will "visit" more than L clusters and generate a (corresponding) data matrix[19] $\boldsymbol{X}^T$ with orthogonal columns. Then, whenever the number of observations N is equal to a multiple of the number of directly estimated cluster centers, the autocorrelation matrix will be diagonal: $\boldsymbol{\Phi} = N|x|^2\boldsymbol{I}_L$. As shown in [4], diagonality of $\boldsymbol{\Phi}$ results in optimal LS CIR estimation performance for a given number of training samples and CIR taps. *Exact* diagonality of $\boldsymbol{\Phi}$ for a *finite* number of observations can be achieved, e.g., by using training sequences derived from m-sequences of appropriate order [4]. A method for constructing such a training sequence is as follows: Given a (binary) m-sequence of order $L-1$, i.e., $m = 2^{L-1} - 1$, circularly shift it so that it begins with $L-1$ 1's (or -1's). Then insert a 1 (respectively, -1) at its beginning. This is done in order to include the L-tuple of all 1's (respectively, -1's).[20] Finally, multiply the sequence with x. The resulting sequence is of length 2^{L-1} and its periodic repetition can be used as a training sequence. An example for $L = 3$ is the sequence of length $2^{3-1} = 4$

$$x, x, x, -x.$$

Note that this is the same sequence used in the CE method for the case of $L = 4$. However, here it is employed for $L = 3$. Another example, for $L = 5$, is given by the following sequence of length $2^{5-1} = 16$: $x, x, x, x, x, -x, x, -x, x, x, -x, -x, x, -x, -x, -x$.[21]

It follows from the properties of m-sequences [10], [18] that the use of such a training sequence will "visit" half of the cluster centers associated with combinations of x and $-x$, i.e., 2^{L-1}, with the remaining 2^{L-1} being simply their negatives. For example, the periodic repetition of the above sequence for $L = 3$ will cyclically generate the following four 3-tuples:

$$\begin{matrix} x & x & x \\ -x & x & x \\ x & -x & x \\ x & x & -x. \end{matrix}$$

Thus, averaging the observations associated with the L-tuples appearing in such a training sequence will result, in effect, in *direct* estimates of the centers of *all* 2^L clusters that correspond to $\pm x$ L-tuples.

Note that a center $\bar{y}_{\boldsymbol{x}}$ can use for its estimation the negatives of the observations corresponding to the center $\bar{y}_{-\boldsymbol{x}}$. Thus, in the above example, one can use the observations corresponding to, e.g., the input L-vector $[x, -x, x]$ (3rd tuple above) to estimate the center $\bar{y}_{[-x,x,-x]}$ instead of $\bar{y}_{[x,-x,x]}$, by simply changing the signs of these observations. Hence, without loss of generality and for the purposes of deriving the new algorithm, we will use as training L-tuples not those generated directly by the above sequence but those 2^{L-1} L-tuples of $\pm x$ whose first entry is x. These result simply via a sign change and can be arranged, in a specific order, at the rows of a $2^{L-1} \times L$ data matrix, which, for the example of $L = 3$, is given by

$$\boldsymbol{X}^T = x \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \equiv x\boldsymbol{X}_1^T \tag{36}$$

with $\boldsymbol{X}_1^T$ again denoting the corresponding matrix for the $x = 1$ case.

Observe that the columns of $\boldsymbol{X}^T$ above are orthogonal; in general

$$\boldsymbol{X}^*\boldsymbol{X}^T = 2^{L-1}|x|^2\boldsymbol{I}_L. \tag{37}$$

Let $\hat{\bar{\boldsymbol{y}}}$ denote the $2^{L-1} \times 1$ vector of center estimates for the clusters associated with the rows of $\boldsymbol{X}^T$, computed by averaging the corresponding observations. It then follows from the relation $\hat{\bar{\boldsymbol{y}}} = \boldsymbol{X}^T\hat{\boldsymbol{h}}$ that the corresponding CIR estimate will be given by

$$\hat{\boldsymbol{h}} = \frac{1}{2^{L-1}|x|^2}\boldsymbol{X}^*\hat{\bar{\boldsymbol{y}}} \tag{38}$$

and the vector of tap contribution estimates

$$\hat{\boldsymbol{c}}_x \equiv x\hat{\boldsymbol{h}} = \frac{1}{2^{L-1}}\boldsymbol{X}_1\hat{\bar{\boldsymbol{y}}}. \tag{39}$$

The center estimation scheme, which is based on the computation of the tap contributions via (39), namely as averages of the estimates of the centers of the 2^{L-1} clusters (or their negatives) generated by the above training sequence, will be henceforth referred to as MCE. The corresponding CIR estimate, given by (38), will be denoted by $\hat{\boldsymbol{h}}^{\mathrm{MCE}}$. Note that (39) extends the way

[18] Of course, one could think of other $L \times L$ matrices of ± 1 that are orthogonal for other (even) values of L as well. However, the matrix of the form of (10) (or a permuted version thereof) is the only choice if only the L-tuples corresponding to the rows of the matrix are to appear in the training sequence (no discarded observations).

[19] Note that, although some matrices and vectors in this section are of a higher dimension than their counterparts in previous sections, the same notation will be employed here for the sake of continuity.

[20] The result is a so-called *de Bruijn* sequence [10].

[21] The presence of a DC level in these sequences (as well as in those used in CE), whenever undesirable, can be addressed with the aid of appropriate modulation techniques (see, e.g., [9, Ch. 6 and 11]).

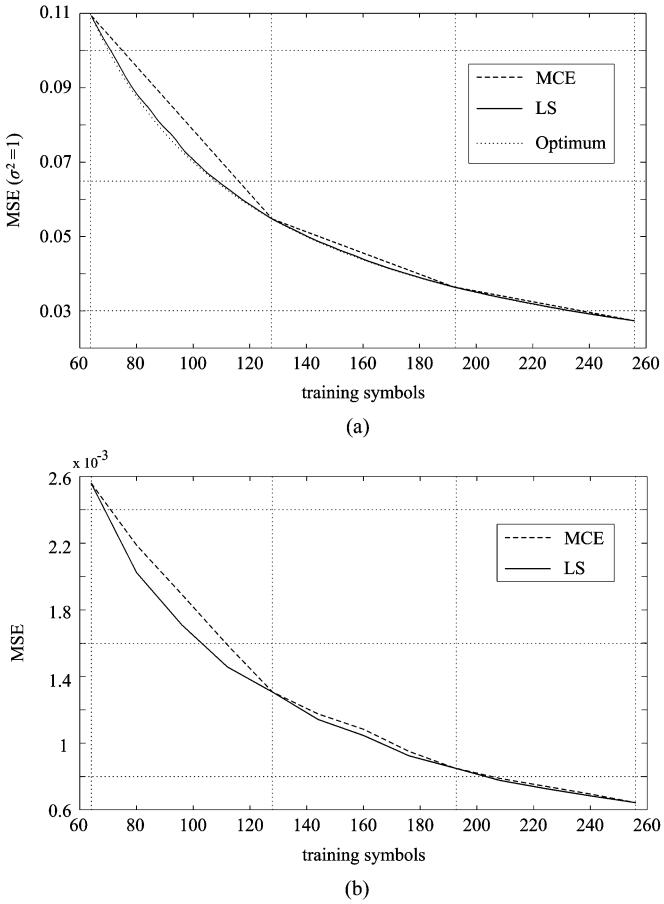


Fig. 6. Total mean-squared tap-estimation error (MSE) curves resulting from MCE and LS, trained with BPSK m-sequence derived sequences, for channels of length $L = 7$: (a) Theoretical ($\sigma^2 = 1$); (b) experimental (SNR = 20 dB). The dotted curve in (a) corresponds to the optimal total variance ($L\sigma^2/N$).

tap contributions are estimated in CE for $L = 4$ to other values of L.

To evaluate the MSE performance of MCE as compared to that of LS, let us, as before, write the number of observations as $N = n_0 2^{L-1} + n_1$, with $0 \le n_1 < 2^{L-1}$. Noting that the 2^{L-1} center estimates in $\hat{\bar{\boldsymbol{y}}}$ are uncorrelated with each other and n_1 of them have used one more sample for their computation than the rest, the following expression for the variance of each tap estimate is obtained:

$$\begin{aligned}\mathrm{var}\left(\hat{h}_m^{\mathrm{MCE}}\right) &= \frac{1}{|x|^2}\cdot\frac{n_1\frac{\sigma^2}{n_0+1}+(2^{L-1}-n_1)\frac{\sigma^2}{n_0}}{(2^{L-1})^2}\\ &= \frac{\sigma^2}{n_0 2^{L-1}|x|^2}\left[1-\frac{n_1}{2^{L-1}(n_0+1)}\right]. \quad (40)\end{aligned}$$

The corresponding total variance $\mathrm{Trace}(\mathrm{cov}(\hat{\boldsymbol{h}}^{\mathrm{MCE}})) = L\mathrm{var}(\hat{h}_m^{\mathrm{MCE}})$ is plotted in Fig. 6(a) for BPSK input signaling ($x = 1$), $L = 7$ and $\sigma^2 = 1$, along with that of the LS estimate, computed as in [4]. As expected from (40), MCE attains LS performance when the number of observations is an integer multiple of the number of directly estimated centers, 2^{L-1}, i.e., when $n_1 = 0$. Similar to the LS performance is also achieved for a sufficiently large N, in which case n_1 can be considered negligible. For comparison purposes, Fig. 6(a) also shows the minimum total MSE, namely $L\sigma^2/N$. Observe that the MSE resulting from both the LS and the MCE method coincides with the optimum one for values of N that are multiples of 2^{L-1}, since these cases correspond to perfectly random data [4]. The corresponding experimental results for [randomly generated as in Fig. 4(b)] CIRs of length 7 and SNR = 20 dB are depicted in Fig. 6(b) and are seen to be in good agreement with the theoretical ones.

It thus turns out that, for $N = n_0 2^{L-1}$, $\hat{\boldsymbol{h}}^{\mathrm{MCE}}$ is LS-optimal. This can also be seen by noting that the autocorrelation matrix $\boldsymbol{\Phi}$, defined as in (26) with $\boldsymbol{\mathcal{X}}$ built as in (24), equals

$$\boldsymbol{\Phi} = n_0 2^{L-1}|x|^2\boldsymbol{I}_L + \boldsymbol{X}_{n_1}^*\boldsymbol{X}_{n_1}^T$$

in view of (37). For $n_1 = 0$ this becomes $\boldsymbol{\Phi} = N|x|^2\boldsymbol{I}_L$. Moreover, the corresponding $\boldsymbol{\mathcal{A}}$ matrix (cf. (19)) is then given by

$$\boldsymbol{\mathcal{A}} = \frac{1}{n_0}[\underbrace{\boldsymbol{I}_{2^{L-1}} \quad \boldsymbol{I}_{2^{L-1}} \quad \cdots \quad \boldsymbol{I}_{2^{L-1}}}_{n_0}].$$

Thus, the LS solution is (cf. (29)

$$\begin{aligned}\hat{\boldsymbol{h}}^{\mathrm{LS}} &= \frac{1}{n_0 2^{L-1}|x|^2}\boldsymbol{\mathcal{X}}^*\boldsymbol{y}\\ &= \frac{1}{2^{L-1}|x|^2}\boldsymbol{X}^*\frac{1}{n_0}[\boldsymbol{I}_{2^{L-1}} \quad \boldsymbol{I}_{2^{L-1}} \quad \cdots \quad \boldsymbol{I}_{2^{L-1}}]\boldsymbol{y}\\ &= \frac{1}{2^{L-1}|x|^2}\boldsymbol{X}^*\boldsymbol{\mathcal{A}}\boldsymbol{y} = \frac{1}{2^{L-1}|x|^2}\boldsymbol{X}^*\hat{\bar{\boldsymbol{y}}}\\ &= \hat{\boldsymbol{h}}^{\mathrm{MCE}}. \quad (41)\end{aligned}$$

A. Efficient Implementation of MCE

Consider (39), which, for the example of (36), takes the form

$$\begin{aligned}\hat{c}_x^2 &= \frac{\overbrace{\hat{\bar{y}}^0}^{\alpha_0^2} - \overbrace{\hat{\bar{y}}^1}^{\alpha_1^2} + \overbrace{\hat{\bar{y}}^2}^{\alpha_2^2} - \overbrace{\hat{\bar{y}}^3}^{\alpha_3^2}}{4}\\ &= \frac{\overbrace{(\hat{\bar{y}}^0+\hat{\bar{y}}^2)}^{\beta_0^2} - \overbrace{(\hat{\bar{y}}^1+\hat{\bar{y}}^3)}^{\beta_1^2}}{4}\\ \hat{c}_x^1 &= \frac{\overbrace{(\hat{\bar{y}}^0+\hat{\bar{y}}^1)}^{\beta_0^1=\alpha_0^1} - \overbrace{(\hat{\bar{y}}^2+\hat{\bar{y}}^3)}^{\beta_1^1=\alpha_1^1}}{4}\\ \hat{c}_x^0 &= \frac{\overbrace{\overbrace{(\hat{\bar{y}}^0+\hat{\bar{y}}^1)}^{\alpha_0^1} + \overbrace{(\hat{\bar{y}}^2+\hat{\bar{y}}^3)}^{\alpha_1^1}}^{\alpha_0^0}}{4}.\end{aligned}$$

It is apparent from the above expressions that some of the additions performed for calculating a tap contribution estimate are also needed to calculate subsequent estimates. This suggests that

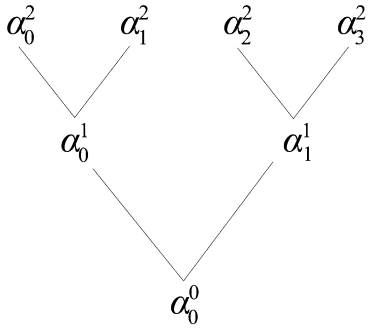


Fig. 7. Graphical representation of the recursive computation of α_i^{L-k}'s for the example of $L = 3$.

a careful grouping of the required additions may considerably reduce the operations count. Thus, for the above example, we have

$$\begin{aligned}
&\alpha_i^2 = \hat{\bar{y}}^i, \quad i = 0,1,2,3\\
&\beta_0^2 = \alpha_0^2 + \alpha_2^2, \quad \beta_1^2 = \alpha_1^2 + \alpha_3^2\\
&\hat{c}_x^2 = \frac{\beta_0^2 - \beta_1^2}{4}\\
&\alpha_0^1 = \alpha_0^2 + \alpha_1^2, \quad \alpha_1^1 = \alpha_2^2 + \alpha_3^2\\
&\beta_0^1 = \alpha_0^1, \quad \beta_1^1 = \alpha_1^1\\
&\hat{c}_x^1 = \frac{\beta_0^1 - \beta_1^1}{4}\\
&\alpha_0^0 = \alpha_0^1 + \alpha_1^1\\
&\hat{c}_x^0 = \frac{\alpha_0^0}{4}.
\end{aligned}$$

We can now observe that α's can be calculated in a recursive manner, and this is represented graphically via the hierarchical tree structure of Fig. 7. Based on (39) and the structure of the matrix $\boldsymbol{X}^T$ as exemplified by (36), one can easily see that, for the general case of an L-taps channel, the above procedure can be stated as follows:

0. $\alpha_i^{L-1} = \hat{\bar{y}}^i, \quad i = 0,1,\ldots,2^{L-1}-1$
1. For $k = 1,2,\ldots,L-1$
   $\alpha_i^{L-k-1} = \alpha_{2i}^{L-k} + \alpha_{2i+1}^{L-k}, \quad i = 0,1,\ldots,2^{L-k-1}-1$
   $\beta_l^{L-k} = \sum_{i=0}^{2^{L-k-1}-1} \alpha_{2i+l}^{L-k}, \quad l = 0,1$
   $\hat{c}_x^{L-k} = \frac{\beta_0^{L-k} - \beta_1^{L-k}}{2^{L-1}}$
2. $\hat{c}_x^0 = \frac{\alpha_0^0}{2^{L-1}}$.

Note that α's and β's need not be available for all k at the same time. Once α_i^{L-k-1} and β_l^{L-k-1} have been computed, α_i^{L-k} and β_l^{L-k} are not needed anymore. This reduces considerably the memory requirements of the above procedure.

VII. Computational Complexity Considerations

We have shown that the CE method is LS optimal when a periodic repetition of L properly selected input vectors is used for training. When used with random data, the LS method exhibits an improved performance over CE. The improvement factor was computed to $L/4$. This performance gain, however, is obtained at a substantially higher computational cost.

We shall now compare the computational requirements of the two methods in the context of MLSE equalization.[22] In the case of real data, CE requires $N + L - 1$ additions and $2L + 1$ divisions in order to compute the tap contributions $\hat{c}_x^m$, $m = 0,1,\ldots,L-1$. Taking into account the symmetry of the input alphabet and the fact that $\hat{c}_x^m$ and $\hat{c}_{-x}^m, m = 0,1,2,\ldots,L-1$ are known, it follows that, for the computation of the rest of the tap contributions, $(M/2-1)L$ multiplications are required (cf. (4)). $(M^L/2)(L-1)$ additions are then needed in order to compute the M^L center estimates. The computational requirements for complex input data are similarly derived. In that case, and if real operations are considered, the operations counts given above for the computation of the $\hat{c}_x^m$'s, $m = 0,1,\ldots,L-1$, have to be doubled. Moreover, for the M-QAM scheme, $4(M/4-1)L$ multiplications and $2(M^L/4)(L-1) + 2(M/4-1)L$ additions are required to compute the rest of the tap contributions and the cluster centers. Observe that, when BPSK (QPSK) data are used, the rest of the tap contributions need no operations for their calculation, since they can simply result via sign changes (resp. rotations). It must be emphasized that *the number of multiplications/divisions required by CE is independent of the size of the training data set*.

The normal procedure followed in the LS method is to first compute the vector $\boldsymbol{z}$ (cf. (27)) and then multiply it with the inverse of the matrix $\boldsymbol{\Phi}$ to determine an estimate for the CIR.[23] For the training data employed here, some simplifications to this procedure are possible and will be adopted here in order to make the fairest possible comparison. Noting that the training sequence consists of a symbol x and its negative, we can write the matrix $\boldsymbol{\mathcal{X}}$ in (24) in the form $\boldsymbol{\mathcal{X}} = x\boldsymbol{\mathcal{X}}_1$, where $\boldsymbol{\mathcal{X}}_1$ is again defined as in (24) but with $\boldsymbol{X}$ replaced by the corresponding matrix, $\boldsymbol{X}_1$, of ± 1's. Then the autocorrelation matrix can be expressed as $\boldsymbol{\Phi} = \boldsymbol{\mathcal{X}}^*\boldsymbol{\mathcal{X}}^T = |x|^2\boldsymbol{\mathcal{X}}_1\boldsymbol{\mathcal{X}}_1^T$, whereby it is seen to be real. The CIR vector will therefore be written as

$$\begin{aligned}
\hat{\boldsymbol{h}}^{\mathrm{LS}} &= \boldsymbol{\Phi}^{-1}\boldsymbol{z}\\
&= \frac{1}{|x|^2}\left(\boldsymbol{\mathcal{X}}_1\boldsymbol{\mathcal{X}}_1^T\right)^{-1} x^*\boldsymbol{\mathcal{X}}_1\boldsymbol{y}.
\end{aligned}$$

Hence, if one computes the vector of tap contributions $\hat{\boldsymbol{c}}_x \equiv [\hat{c}_x^0, \hat{c}_x^1, \ldots, \hat{c}_x^{L-1}]^T$ instead, the required computations are implied by the relation

$$\begin{aligned}
\hat{\boldsymbol{c}}_x &= x\hat{\boldsymbol{h}}^{\mathrm{LS}}\\
&= x\frac{1}{|x|^2}\left(\boldsymbol{\mathcal{X}}_1\boldsymbol{\mathcal{X}}_1^T\right)^{-1} x^*\boldsymbol{\mathcal{X}}_1\boldsymbol{y}\\
&= \left(\boldsymbol{\mathcal{X}}_1\boldsymbol{\mathcal{X}}_1^T\right)^{-1}\boldsymbol{\mathcal{X}}_1\boldsymbol{y}\\
&\equiv \boldsymbol{\Phi}_1^{-1}\boldsymbol{z}_1. \qquad (42)
\end{aligned}$$

In the real case, the vector $\boldsymbol{z}_1 = \boldsymbol{\mathcal{X}}_1\boldsymbol{y}$ requires $L(N-1)$ additions and LN multiplications for its computation. Its multiplication with the matrix $\boldsymbol{\Phi}_1^{-1} = (\boldsymbol{\mathcal{X}}_1\boldsymbol{\mathcal{X}}_1^T)^{-1}$ requires another

[22] Note that in the LS-based MLSE equalizer, although the CIR vector $\hat{\boldsymbol{h}}^{\mathrm{LS}}$ is first estimated, it is the quantities $x\hat{h}_m^{\mathrm{LS}}$ that will ultimately be needed in computing the convolutions (centers) $(\hat{\boldsymbol{h}}^{\mathrm{LS}})^T\boldsymbol{x}$.

[23] This paper is only concerned with batch estimation; hence, no reference is made to fast recursive LS (FLS) algorithms [11]. Besides, as shown in [16], CE is far more computationally efficient even than LMS, with respect to the total number of operations required per training session.

TABLE I
COMPUTATIONAL REQUIREMENTS OF CE AND LS METHODS (REAL DATA)

Method	Add/Sub	Mult/Div
CE	$N+L-1$	$2L+1$
LS	$L(N-1)+L(L-1)$	$LN+L^2$

$L(L-1)$ additions and L^2 multiplications. Once the tap contributions in (42) have been calculated, the remaining computation for the estimation of the rest of the tap contributions and subsequently the cluster centers is identical to that for the CE method. The computational requirements for the complex case are derived in an analogous manner. A summary of the operations counts of the two methods for a channel of length L, a real symmetric input constellation of size M, and a sample set of size N is provided in Table I. The complexity for the complex case can be found by just doubling these counts (this is because all multiplications/divisions performed in the two methods are between a *real* and a complex number). Only the operations involved in calculating the tap contributions for a symbol x are included in Table I; the rest of the computation in the context of MLSE equalization is the same in both methods. Observe that the counts given for the LS method in the above table, as well as in those to follow, do not include the computations required to compute and invert $\mathbf{\Phi}_1$ ($\mathbf{\Phi}_1^{-1}$ can be assumed to have been pre-computed [4][24]). Nonetheless, as it can be seen from the above results, even with no matrix computation and inversion taken into account, the computational burden for the LS method, corresponding to the estimation of the tap contribution vector $\hat{\mathbf{c}}_x$, raises to $O(NL)$, for $N > L$, as compared to only $O(N)$ for the CE approach. For example, in the realistic case of $L = 5$ taps and $N = 30$ observations, with real input data, the LS method requires 175 multiplications and 165 additions, whereas only 11 multiplications/divisions and 34 additions are needed in the CE method. The computational advantage of the CE method over LS is apparent, particularly in the required multiplications/divisions.

To complete the comparison of the two methods, it must also be emphasized that, in practical wireless communications scenarios, where channels are rather short (for example, L is of the order of 5 in GSM), the performance degradation factor $(L/4)$, pointed out above, can hardly be of any significant effect. Furthermore, as it was shown in Section V, there is no such degradation for channels of length 4. This holds also true (with a negligible increase in computational complexity) for the cases of $L = 3$ (see Section VI) and $L = 2$ (see Appendix).

Regarding MCE, the need to estimate 2^{L-1} cluster centers directly instead of only L seems to diminish the computational advantage of the center estimation method over classical LS. However, if the efficient implementation described above is adopted, an operations count that is still lower than that of direct LS can be attained. Let us count the operations required, for the real case, in the MCE procedure, outlined in Section VI. Step 0 involves the estimation of the 2^{L-1} centers. For each of the first n_1 centers n_0 additions and one division are required, whereas the rest of them need n_0-1 additions and one division each. Hence, Step 0 is completed with $n_1 n_0 + (2^{L-1} - n_1)(n_0 - 1) = N -$ 2^{L-1} additions and 2^{L-1} divisions. In Step 1, $2^{L-2} + 2^{L-3} + \cdots + 1 = 2^{L-1} - 1$ additions are required for computing the α's. The computation of the β's will require $\sum_{k=1}^{L-1} 2(2^{L-k-1}-1) = 2(2^{L-1}-1) - 2(L-1)$ additions. We need one more addition and one division for each of the last $L-1$ tap contributions. The first tap contribution $\hat{c}_x^0$ needs one division only (Step 2). The operations counts required for the calculation of the $\hat{c}_x^m$'s are summarized in Table II for both the MCE and LS methods. Again, these have to be doubled in the complex case. As seen in the above table, MCE is considerably more efficient than LS. It must be noted that, similarly to CE, *the number of multiplications/divisions required in MCE is independent of the size of the training data set*. To make the computational advantages of MCE over LS more apparent, the operations counts of the two methods for several concrete values of L and N are given in Table III. For the operations counts of the LS method, it has been taken into account that Φ_1 equals a scalar times the identity matrix whenever N is a multiple of 2^{L-1}.

[24]The computational savings resulting from having pre-computed the $L \times N$ matrix $\mathbf{\Phi}_1^{-1}\boldsymbol{\mathcal{X}}_1$ instead are only of the order of L^2, which is negligible for large N.

TABLE II
COMPUTATIONAL REQUIREMENTS OF MCE AND LS METHODS (REAL DATA)

Method	Add/Sub	Mult/Div
MCE	$N+2^L-L-2$	$2^{L-1}+L$
LS	$L(N-1)+L(L-1)$	$LN+L^2$

TABLE III
COMPUTATIONAL REQUIREMENTS OF MCE AND LS METHODS (REAL DATA)

		Add/Sub		Mult/Div	
L	N	MCE	LS	MCE	LS
3	10	13	33	7	39
3	15	18	48	7	54
3	20	23	57	7	60
6	64	120	378	38	384
6	80	136	504	38	516
6	106	162	660	38	672
8	320	566	2608	136	2624
8	384	630	3064	136	3072
8	448	694	3632	136	3648

VIII. CONCLUSION

The relationship of the CE technique, used in the clustering-based MLSE equalizer of [16] and [17], with the LS method was studied, both in the context of channel identification and cluster center estimation for MLSE equalization. It was proved that, when using a properly constructed short (periodically repeated) training sequence, the two methods lead to the same solution. The LS method, when trained with random data, was shown to outperform CE. To address this problem, a modified CE method, called MCE, was proposed that attains LS performance when perfectly random data are employed. We demonstrated that LS performance is attained by CE/MCE at a computational cost substantially lower than that of classical LS estimation. The results of this paper bring out the importance of CE as a methodology that combines high performance, simplicity, and low computational cost, as required in a practical equalization task. Moreover, an alternative, algebraic viewpoint on the CE technique is provided, while at the same time leading to a

new interpretation of the LS method in terms of averaging for cluster center estimation.

APPENDIX
CASE OF $L = 2$ TAPS

When $L = 2$, an approach different from that presented in Section III must be taken since the centers suggested therein, namely $\bar{y}^0 = \bar{y}_{[-x,x]}$ and $\bar{y}^1 = \bar{y}_{[x,-x]} = -\bar{y}^0$, do not carry sufficient information for the estimation of two tap contributions. Said in different words, the 2-tuples $[-x, x]$ and $[x, -x]$ are not linearly independent. A way out of this is to estimate instead the centers $\bar{y}_{[x,x]}$ and $\bar{y}_{[x,-x]}$. Then the tap contributions are computed as $\hat{c}_x^0 = (\hat{\bar{y}}_{[x,x]} + \hat{\bar{y}}_{[x,-x]})/2$ and $\hat{c}_x^1 = (\hat{\bar{y}}_{[x,x]} - \hat{\bar{y}}_{[x,-x]})/2$. A training sequence for estimating these two centers might be constructed as the periodic repetition of $[x, x, -x]$. Note that the 2-tuple $[-x, x]$ will also appear in that sequence. Since this is the negative of the tuple $[x, -x]$, the corresponding observation can be used for the estimation of the center $\bar{y}_{[x,-x]}$ once its sign has been changed. Note that the two centers are not "visited" equally often when using the above sequence. Within the N received samples, with N assumed to be a multiple of 3, $N/3$ samples correspond to the tuple $[x, x]$ and the rest $2N/3$ to $[x, -x]$. The resulting MSE is thus given by

$$\begin{aligned}\mathrm{var}(\hat{c}_x^m) &= \frac{\mathrm{var}\left(\hat{\bar{y}}_{[x,x]}\right) + \mathrm{var}\left(\hat{\bar{y}}_{[x,-x]}\right)}{2^2} \\ &= \frac{\frac{\sigma^2}{N/3} + \frac{\sigma^2}{2N/3}}{4} \\ &= \frac{9}{8} \cdot \frac{\sigma^2}{N}\end{aligned}$$

which is only slightly higher than that of the LS method with random data. One can do better than that by simply using a training sequence that will visit the two clusters the same number of times. Such a sequence can be constructed by periodically repeating $[x, x, x, -x]$.[25] Although longer than the above, this sequence generates each of the two tuples, $[x, x]$ and $[x, -x]$, $N/2$ times, where N is a multiple of 4. The corresponding MSE is then easily verified to be the optimum one, as follows:

$$\mathrm{var}\,(\hat{c}_x^m) = \frac{\frac{\sigma^2}{N/2} + \frac{\sigma^2}{N/2}}{4} = \frac{\sigma^2}{N}.$$

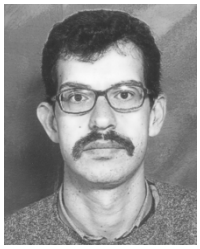
ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions that helped improve the quality of the manuscript.

REFERENCES

[1] S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[2] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 570–590, Jul. 1993.
[3] ——, "Adaptive Bayesian equalizer with decision feedback," *IEEE Trans. Signal Process.*, vol. 41, no. 9, pp. 2918–2927, Sep. 1993.
[4] S. N. Crozier, D. D. Falconer, and S. A. Mahmoud, "Least sum of squared errors (LSSE) channel estimation," *Proc. Inst. Elect. Eng.—Part F*, vol. 138, no. 4, pp. 371–378, Aug. 1991.
[5] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 3, pp. 363–378, May 1972.
[6] ——, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
[7] K. Georgoulakis and S. Theodoridis, "Efficient clustering techniques for channel equalization in hostile environments," *Signal Process.*, vol. 58, pp. 153–164, 1997.
[8] ——, "Channel equalization for coded signals in hostile environments," *IEEE Trans. Signal Process.*, vol. 47, no. 6, pp. 1783–1787, Jun. 1999.
[9] I. A. Glover and P. M. Grant, *Digital Communications*, 2nd ed. Essex, U.K.: Pearson Education, Ltd., 2004.
[10] S. W. Golomb, *Shift Register Sequences*. San Francisco, CA: Holden-Day, 1967.
[11] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2001.
[12] N. Kalouptsidis and S. Theodoridis, *Adaptive System Identification and Signal Processing Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[13] Y. Kopsinis and S. Theodoridis, "Reduced-complexity clustering techniques for nonlinear channel equalization," presented at the Proc. 5th Int. Conf. Signal Processing(ICSP)/16th World Computer Congr. (WCC), Beijing, China, Aug. 2000.
[14] ——, "A novel low complexity cluster based MLSE equalizer for QPSK signaling scheme," presented at the Digital Signal Processing (DSP) Conf., Santorini, Greece, Jul. 2002.
[15] ——, "A novel cluster based MLSE equalizer for 2-PAM signaling scheme," presented at the European Signal Processing Conf. (EUSIPCO), Toulouse, France, Sep. 2002.
[16] ——, "An efficient low complexity technique for MLSE equalizers for linear and nonlinear channels," *IEEE Trans. Signal Process.*, vol. 51, no. 12, pp. 3236–3248, Dec. 2003.
[17] ——, "A novel cluster based MLSE equalizer for M-PAM signaling schemes," *Signal Process.*, vol. 83, pp. 1905–1918, 2003.
[18] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers, Vol. 2: Synchronization, Channel Estimation, and Signal Processing*, 2nd ed. New York: Wiley, 1997.
[19] J. R. Montalvão Filho, B. Dorizzi, and J. C. M. Mota, "Channel estimation by symmetrical clustering," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1459–1469, Jun. 2002.
[20] J. G. Proakis, *Digital Communications*, 3rd ed: McGraw-Hill, 1995.
[21] S. Qureshi, "Fast start-up equalization with periodic training sequences," *IEEE Trans. Inf. Theory*, vol. 23, no. 5, pp. 553–563, Sep. 1977.
[22] S. Qureshi, "Correction to 'Fast start-up equalization with periodic training sequences'," *IEEE Trans. Inf. Theory*, vol. 24, no. 1, p. 130, Jan. 1978.
[23] ——, "Adaptive equalization," *Proc. IEEE*, vol. 73, no. 9, pp. 1349–1386, Sep. 1985.
[24] S. Theodoridis, C. F. N. Cowan, C. P. Callender, and C. M. S. See, "Schemes for equalization of communication channels with nonlinear impairments," *Proc. Inst. Elect. Eng.—Communications*, vol. 142, no. 3, pp. 165–171, Jun. 1995.
[25] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 2nd ed. New York: Academic, 2003.

[25]Observe that this is the sequence that would be used in the $L = 4$ case.



Eleftherios Kofidis (S'94–M'97) was born on September 6, 1967, in Trikala, Greece. He received the diploma (hons.) degree and the Ph.D. degree from the Department of Computer Engineering and Informatics, University of Patras, Patras, Greece, in 1990 and 1996, respectively.

From 1996 to 1998, he was with the Greek Army. From 1998 to 2000, he was a Postdoctoral Research Fellow with the Communications, Image and Information Processing Department, Institut National des Télécommunications, Evry, France. From 2001 to 2004, he held Adjunct Professor positions with the University of Piraeus, Piraeus, Greece, and the University of Peloponnese, Peloponnese, Greece. He is currently a Lecturer of informatics in the Department of Statistics and Insurance Science, University of Piraeus. His current research interests are in signal processing for communications.

Dr. Kofidis is a Member of the Technical Chamber of Greece.

Yannis Kopsinis (S'97–M'03) was born in Patras, Greece, on May 1, 1973. He received the B.Sc. degree in informatics and Ph.D. degree from the Department of Informatics and Telecommunications, University of Athens, Athens, Greece, in 1998 and 2003, respectively.

Since 2004, he has been a Research Fellow with the Institute for Digital Communications, University of Edinburgh, Edinburgh, U.K. His research interests include channel equalization, adaptive algorithms, and electronic signal processing and performance evaluation for lightwave communications.

Sergios Theodoridis (SM'03) received the B.Sc. (hons.) degree in physics from the University of Athens, Athens, Greece, and the M.Sc. and Ph.D. degrees from the Department of Electronics and Electrical Engineering, Birmingham University, U.K.

He is currently Professor of signal processing and communications in the Department of Informatics and Telecommunications, University of Athens. For four years, he was also a Member of the Board of Directors of COSMOTE (the Greek DCS-1800 mobile phone operating company). His research interests lie in the areas of adaptive algorithms, channel equalization, pattern recognition, signal processing for music, and OCR systems. He has extensively published in prestigious international journals and refereed conferences. He is the coeditor of the book *Adaptive System Identification and Signal Processing Algorithms* (Englewood Cliffs, NJ: Prentice-Hall, 1993) and coauthor of the book *Pattern Recognition* (New York: Academic, 2nd ed., 2003) and of three books in Greek, two of which were for the Greek Open University.

Dr. Theodoridis is currently the President of the EURASIP. He was the General Chairman of EUSIPCO 1998, and he is the Technical Program Co-Chair for the International Symposium on Circuits and Systems (ISCAS) 2006. He has served as Vice-Chairman of the Greek Pedagogical Institute. He is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE SIGNAL PROCESSING MAGAZINE and a member of the editorial boards of the EURASIP journals *Signal Processing* and *Applied Signal Processing*.