ON THE LEAST SQUARES PERFORMANCE OF A NOVEL EFFICIENT CENTER ESTIMATION METHOD FOR CLUSTERING-BASED MLSE EQUALIZATION

Eleftherios Kofidis, Yannis Kopsinis, and Sergios Theodoridis*

Department of Informatics and Telecommunications,
University of Athens,
Panepistimioupoli, Ilissia,
157 84 Athens, Greece
email:{kofidis,kopsinis,stheodor}@di.uoa.gr

ABSTRACT

Recently, a novel Maximum Likelihood Sequence Estimation (MLSE) equalizer was reported, that avoids the explicit estimation of the channel impulse response. Instead, the centers of the clusters which are formed by the received samples are estimated, in a computationally efficient manner, that exploits the channel linearity and the symmetries underlying the transmitted signal constellation. This paper investigates the relationship of the center estimation (CE) part of the proposed equalizer with the Least Squares (LS) method, demonstrating that it can attain LS performance at a substantially lower computational cost. The importance of CE is thus confirmed, as a methodology that combines high performance, simplicity and low computational cost, as required in a practical equalization task. The results of this paper provide also an alternative, algebraic viewpoint on the CE method, while at the same time leading to a new interpretation of the LS, in terms of averaging for cluster center estimation.

1. INTRODUCTION

One of the major problems encountered in the receiver design of any communication system is that of combatting Inter-Symbol 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., [12]).

The equalizers based on the Maximum Likelihood Sequence Estimation (MLSE) scheme [12] are implemented via the Viterbi algorithm (VA) and they require the channel impulse response (CIR) to be known. For this purpose, one may resort to any appropriate identification method [4]. 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 metric computations for the VA.

Recently, a novel MLSE equalizer was proposed, that circumvents the problem of *explicit* CIR parametric modelling, leading to substantial computational savings [6]–[10]. It belongs to the class of the so-called Clustering-Based Sequence Equalizers (CBSE) (e.g., [3]), since it is based on the idea that the set of all possible (noiseless) 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 from the noisy observations via a 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 [6]. Furthermore, the algorithm uses an efficient cluster center estimation (CE) technique, that exploits the structural symmetries underlying the generation mechanism of the clusters of the received samples to considerably reduce the number of cluster centers required to be estimated directly. 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 [9, 10] that the proposed CE technique exhibits a similar to Least Squares (LS) performance, despite its low computational complexity, which is even lower than that of the LMS-based method commonly employed in standard MLSE equalizers [9]. These computational savings are due to the fact that the new setting allows for an efficient exploitation of the symmetries in the input constellation. This paper investigates this issue. It is shown that CE yields the same estimates for the cluster centers with those that would result from computing them as convolution sums using the channel estimate provided by the LS method, trained on the same data with CE. The computational requirements of the two methods are compared, clearly demonstrating the computational advantage of CE over the classical LS method.

2. 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

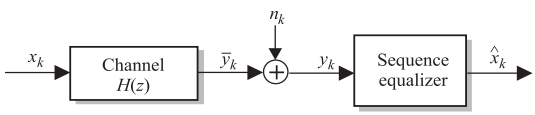

Figure 1: Communication system model

and identically distributed (i.i.d.) symbols, x_k, drawn from a finite alphabet, S, of size M, is transmitted through the channel, modelled 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 [10]) and complex (e.g., M-QAM [9]) constellations. For the sake of generality, the results will be presented for the complex case. In the examples, we will assume 2-PAM (BPSK) and 4-QAM (QPSK) signaling. The received sequence is given by:

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

where the superscripts * and ^{H} denote complex conjugation and Hermitian transposition, respectively, $\underline{h} = [h_0,\ h_1,\ \ldots,\ h_{L-1}]^T$ is the vector of the (generally complex) L taps of the CIR, $\underline{x}_k =$

*Currently with the School of Engineering and Electronics, University of Edinburgh, UK (e-mail: y.kopsinis@ed.ac.uk).

[1]This is also the case in symbol-by-symbol equalizers that are based on clustering; see, e.g., [1, 11].

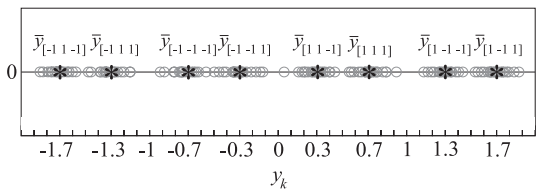

Figure 2: Plot of the clusters formed by the received samples when BPSK symbols are transmitted, the transfer function of the channel is $H(z) = 1 - 0.5z^{-1} + 0.2z^{-2}$ and white Gaussian noise of SNR=20 dB is present. Stars denote cluster centers, while gray circles correspond to noisy channel output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

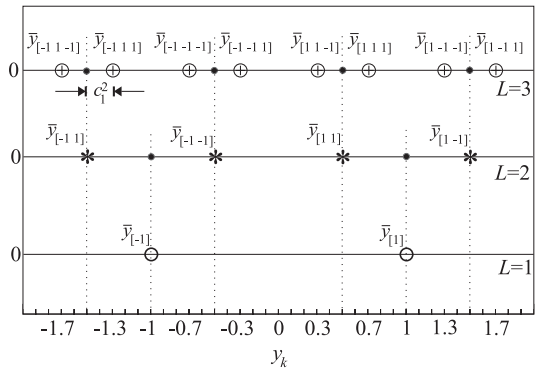

Figure 3: $\oplus$'s denote cluster centers which correspond to the 3-tap channel $H(z) = 1 - 0.5z^{-1} + 0.2z^{-2}$ with BPSK input. The centers of the clusters that are due only to the first (and the second) tap are shown as $\circ$'s ($*$'s).

$[x_k,\ x_{k-1},\ \ldots,\ x_{k-L+1}]^T$ is the input data vector, $\bar{y}_k$ is the noiseless channel output, and n_k is additive, zero-mean white noise, uncorrelated with $\bar{y}_k$. The noisy observations, y_k, are fed to the sequence equalizer whose aim is to provide estimates, $\hat{x}_k, \hat{x}_{k-1}, \ldots, \hat{x}_{k-L+1}$, of the input symbol sequence.

3. CENTER ESTIMATION (CE) METHOD

As it is well known [12], the MLSE equalizer has first to estimate the CIR and then apply the VA (or one of its variants) to estimate the symbol sequence $\underline{x}_k$, based on metric computations of the form $D_{k\underline{x}} = |y_k - \underline{h}^H \underline{x}|^2$, with $\underline{x}$ ranging over the set of all possible L-tuples of input symbols (associated with the states). However, it can be readily seen that what one really needs, instead, are the noiseless observations $\bar{y}_{\underline{x}} \equiv \underline{h}^H \underline{x}$, since $D_{k,\underline{x}} = |y_k - \bar{y}_{\underline{x}}|^2$. Hence, the CIR estimation step can be bypassed [6]. The possible values that $\bar{y}_{\underline{x}}$ can take are simply the points (centers) around which the received samples (observations) y_k are clustered, due to the presence of the noise. Fig. 2 shows the received samples for a 3-tap channel with transfer function $H(z) = 1 - 0.5z^{-1} + 0.2z^{-2}$ and BPSK input, when white Gaussian noise, corresponding to an SNR of 20 dB, is also present. The notation $\bar{y}_{[x_k, x_{k-1}, x_{k-2}]}$ denotes the cluster center which is associated with the transmitted symbol sequence $\underline{x} = [x_k, x_{k-1}, x_{k-2}]$ 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 Fig. 2. However, the information contained in the cluster centers is highly redundant. This is due to the intrinsic dependency between the locations of the cluster centers, caused by the linearity of the channel and the symmetry of the symbol alphabet. Fig. 3 depicts the centers of the clusters formed by the output samples of the first tap ($\circ$'s), the two first taps ($*$'s) and the 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 4 centers due to the first and the second tap can be grouped in two pairs with each of them being symmetrically located around the centers associated with the first tap. This rationale carries on for the centers generated by all 3 taps of the channel. In a way, the locations of the observations follow a hierarchical pattern of symmetries, depending on the channel length.

It has been shown [8, 10] that this structure in the cluster center constellation, exhibited in the above examples, implies that only L (out of the M^L) properly selected cluster centers need to be directly estimated. The rest of them ($M^L - L$) can then be easily determined on the basis of the estimates of these L centers. To precisely describe the estimation procedure, let

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

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

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

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:

$$c_{x'}^m = \frac{x'}{x} c_x^m \tag{4}$$

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 eq. (3)) requires the estimation of only L *tap contributions* c_x^m, 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 that have to be directly estimated was proposed in [7]–[10]. First choose any one of the M^L centers, say $\bar{y}_{[x_0, x_1, \ldots, x_{L-1}]}$, and call it *basic center*, C_{basic}, and the corresponding L-tuple *basic sequence*, $\underline{x}_{\text{basic}} = [x_0, x_1, \ldots, x_{L-1}]$.[2] Then the L centers to be directly estimated are chosen as those which correspond to the basic sequence with a sign change in one of its entries: $C_0 \equiv \bar{y}_{[-x_0, x_1, \ldots, x_{L-1}]}$, $C_1 \equiv \bar{y}_{[x_0, -x_1, \ldots, x_{L-1}]}, \ldots, C_{L-1} \equiv \bar{y}_{[x_0, x_1, \ldots, -x_{L-1}]}$. These L centers can be estimated via any supervised clustering algorithm [13]. For example, a simple averaging of the observations, $y_k^{C_m}$, belonging to the corresponding cluster, m, was proposed in [7]–[10]:

$$\hat{C}_m = \frac{1}{N_{C_m}} \sum_{k=0}^{N_{C_m}-1} y_k^{C_m}, \tag{5}$$

where N_{C_m} denotes the number of observations associated with C_m. Once estimates, $\hat{C}_m$, for C_m, $m = 0, 1, \ldots, L-1$ have been computed, C_{basic} is determined as [7, 8]:

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

The computation of the tap contributions is then straightforward [7, 8]:

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

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

$$\hat{c}_{x_m}^m = \frac{\sum_{i \neq m} \hat{C}_i}{2(L-2)} - \frac{L-3}{2(L-2)} \hat{C}_m \tag{8}$$

[2] The index in x_i here does not represent time.

For the example of Fig. 3, one can set, e.g., $\underline{x}_{\text{basic}} = [1,\ 1,\ 1]$. Then it suffices to estimate the $L = 3$ centers $C_0 = \bar{y}_{[-1,1,1]}$, $C_1 = \bar{y}_{[1,-1,1]}$, and $C_2 = \bar{y}_{[1,1,-1]}$. The tap contributions are then determined as $\hat{c}_1^0 = \frac{\hat{C}_1+\hat{C}_2}{2}$, $\hat{c}_1^1 = \frac{\hat{C}_0+\hat{C}_2}{2}$, and $\hat{c}_1^2 = \frac{\hat{C}_0+\hat{C}_1}{2}$.[3] Clearly, the above procedure does not apply when $L = 2$. A different approach must be taken in this case [5]. 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 directly estimating only L of them, will be referred to here simply as the *CE algorithm*. If the employed training sequence is to be as short and effective as possible, it has to be so chosen as to visit the selected clusters as many times as possible and equally often. It is readily seen that, if *only* the L-tuples corresponding to the centers C_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 can be the periodic repetition of the sequence (for $L > 2$) $[\underbrace{x, x, x, \ldots, x}_{L-1}, -x]$ [8]. This gives rise to L possible input data vectors which appear at the rows of the matrix:

$$X^T = xX_1^T = x \begin{bmatrix} -1 & 1 & \cdots & 1 \\ 1 & -1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & -1 \end{bmatrix} \tag{10}$$

4. EQUIVALENCE WITH LEAST SQUARES (LS) ESTIMATION

4.1 CE as a Channel Identification Method

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

$$\begin{aligned} X^T &= \begin{bmatrix} -x_0 & x_1 & \cdots & x_{L-1} \\ x_0 & -x_1 & \cdots & x_{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_0 & x_1 & \cdots & -x_{L-1} \end{bmatrix} \\ &= X_1^T \operatorname{diag}(x_0, x_1, \ldots, x_{L-1}) \end{aligned} \tag{11}$$

Let, moreover, $\bar{y}^m$, $m = 0, 1, \ldots, L-1$, denote the center corresponding to the mth row of X^T (It was denoted as C_m above.). That is:

$$\underline{\bar{y}} \equiv \begin{bmatrix} \bar{y}^0 & \bar{y}^1 & \cdots & \bar{y}^{L-1} \end{bmatrix}^T = X^T \underline{h}^* \tag{12}$$

It will be shown in this section that the center estimates that result from the CE algorithm, using the training data (11), are LS optimal. In other words, the result is the same as if one had first resorted to the LS method, with the data (11), to identify the CIR and then use it to estimate all possible (noise-free) channel outputs, $\bar{y}_k$, via convolutions. For the purposes of this analysis, and in order to have a common basis of comparison of the CE and LS methods, we shall view CE as a method of channel identification. 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

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

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

$$\underline{\hat{h}} = \left[\operatorname{diag}(x_0^{-1}, x_1^{-1}, \ldots, x_{L-1}^{-1})\underline{\hat{c}}\right]^* \tag{13}$$

[3] When $L = 3$, only two of the three centers are used for the estimation of each tap contribution, leading to a sub-optimal performance. An appropriate modification for this case is elaborated in [5].

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

$$\underline{\hat{\bar{y}}} \equiv \begin{bmatrix} \hat{\bar{y}}^0 & \hat{\bar{y}}^1 & \cdots & \hat{\bar{y}}^{L-1} \end{bmatrix}^T = X^T \underline{\hat{h}}^* \tag{14}$$

which, in view of (11) and (13), can be written as $\underline{\hat{\bar{y}}} = X_1^T \underline{\hat{c}}$ or equivalently (X_1 is nonsingular)

$$\underline{\hat{c}} = X_1^{-T} \underline{\hat{\bar{y}}} \tag{15}$$

Noting that

$$X_1^{-T} = \begin{bmatrix} -\frac{L-3}{2(L-2)} & \frac{1}{2(L-2)} & \cdots & \frac{1}{2(L-2)} \\ \frac{1}{2(L-2)} & -\frac{L-3}{2(L-2)} & \cdots & \frac{1}{2(L-2)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2(L-2)} & \frac{1}{2(L-2)} & \cdots & -\frac{L-3}{2(L-2)} \end{bmatrix} \tag{16}$$

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

In order to be allowed to carry over the results of the comparison of the CE and LS methods from the CIR identification context to that of cluster center estimation (ultimately needed in the VA), we must also note that the mean-squared error (MSE) for the estimation of the centers is directly related to that for the estimation of the channel taps. Indeed, taking into account the special structure of the alphabet S, namely $S = \{a_i; i = 1, 2, \ldots, M/2\} \cup \{-a_i; i = 1, 2, \ldots, M/2\}$, it can be shown [5] that the average MSE for the estimation of a center is given by $E[\operatorname{var}(\hat{\bar{y}})] = P_S \sum_{m=0}^{L-1} \operatorname{var}(\hat{h}_m)$, where $P_S \equiv \frac{1}{M}\sum_{i=1}^{M} |a_i|^2$ is the input signal power.

4.2 Proof of Equivalence with the LS Method

Let $y_0, y_1, \ldots, y_{N-1}$ be the received samples (observations) when the rows of X^T are periodically used as input data vectors, with y_k resulting from row $k \bmod L$. 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:

$$\underline{\hat{\bar{y}}} = \begin{bmatrix} \underbrace{\mathscr{I} \quad \mathscr{I} \quad \cdots \quad \mathscr{I}}_{n_0} & \vdots & \begin{matrix} \frac{1}{n_0+1} I_{n_1} \\ \mathbf{0} \end{matrix} \end{bmatrix} \underline{y} \equiv \mathscr{A}\underline{y} \tag{17}$$

where I_n denotes the nth-order identity matrix,

$$\mathscr{I} \equiv \begin{bmatrix} \frac{1}{n_0+1} I_{n_1} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n_0} I_{L-n_1} \end{bmatrix}, \tag{18}$$

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

$$\begin{aligned} \underline{\hat{h}}^{\text{CE}} &= X^{-H} \underline{\hat{\bar{y}}}^* && (19) \\ &= X^{-H} \mathscr{A} \underline{y}^* && (20) \end{aligned}$$

The corresponding LS problem can be formulated as

$$\min_{\hat{h}} \|\underline{y} - \mathscr{X}^T \underline{\hat{h}}^*\|_2^2 \tag{21}$$

Table 1: Computational requirements of CE and LS methods (Real data).

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

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

$$\mathscr{X}^T \equiv [\underbrace{X \quad X \quad \cdots \quad X}_{n_0} \; X_{n_1}]^T \tag{22}$$

with X_{n_1} denoting the first n_1 columns of X. The solution to (21) is ([4, 2]):

$$\underline{\hat{h}}^{\mathrm{LS}} = (\mathscr{X}\mathscr{X}^H)^{-1}\mathscr{X}\underline{y}^* \equiv \Phi^{-1}\underline{z} \tag{23}$$

where $\Phi \equiv \mathscr{X}\mathscr{X}^H$ is the sampled data autocorrelation matrix and $\underline{z} \equiv \mathscr{X}\underline{y}^*$ denotes the cross-correlation vector. Noting that

$$\Phi = X \begin{bmatrix} (n_0+1)I_{n_1} & \mathbf{0} \\ \mathbf{0} & n_0 I_{L-n_1} \end{bmatrix} X^H = X\mathscr{I}^{-1}X^H, \tag{24}$$

eq. (23) becomes

$$\begin{aligned} \underline{\hat{h}}^{\mathrm{LS}} &= X^{-H}\mathscr{I}X^{-1}\mathscr{X}\underline{y}^* \\ &= X^{-H}\mathscr{I}\begin{bmatrix} I_L & I_L & \cdots & I_L & \begin{matrix} I_{n_1} \\ \mathbf{0} \end{matrix} \end{bmatrix}\underline{y}^* \\ &= X^{-H}\mathscr{A}\underline{y}^*, \end{aligned} \tag{25}$$

which is recognized as the estimate resulting from the CE method (cf. (20)).

It is of interest to note that the above proof applies to any $L \times L$ matrix X^T with linearly independent rows, not necessarily of the form of (11).[4] It thus provides an alternative viewpoint on the LS method through its relationship with averaging for cluster estimation.

5. COMPUTATIONAL COMPLEXITY CONSIDERATIONS

We have shown that the CE method yields the LS solution when a periodic repetition of L properly selected input vectors is used for training. In fact, CE attains LS performance at a computational cost substantially lower than that of direct LS estimation. 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 1. The practical case of (9) is considered here. Only the operations involved in calculating the tap contributions for a symbol x are included; the rest of the computation is the same in both methods. Just double these counts to find the complexity (in terms of real operations) for the complex case.

The normal procedure followed in the LS method is to first compute the vector $\underline{z}$ and then multiply it with the inverse of the matrix Φ. Noting that the training sequence consists of a symbol x and its negative, some simplifications to this procedure are possible and are adopted here. Indeed, if $\mathscr{X}_1$ is defined as in (22) but with X replaced by the corresponding matrix, X_1, of ± 1's, then $(\underline{\hat{h}}_{\mathrm{LS}})^* = \frac{1}{|x|^2}(\mathscr{X}_1\mathscr{X}_1^T)^{-1}x^*\mathscr{X}_1\underline{y}$ and for the vector of tap contributions $\underline{\hat{c}}_x \equiv [\hat{c}_x^0, \hat{c}_x^1, \ldots, \hat{c}_x^{L-1}]^T$ the required computations are implied by the relation $\underline{\hat{c}}_x = (\mathscr{X}_1\mathscr{X}_1^T)^{-1}\mathscr{X}_1\underline{y} \equiv \Phi_1^{-1}\underline{z}_1$. The counts given for the LS method in the above table do not include the computations required to compute and invert Φ_1 (Φ_1^{-1} can be assumed to have been pre-computed [2][5]). Even so, the computational burden for the LS method raises to $O(NL)$, for $N > L$, as compared to only $O(N)$ for the CE approach. It must be emphasized that *the number of multiplications/divisions required by CE is independent of the data size.* Table 2 gives the operations counts required by the two methods for the realistic case of $L = 5$ taps and $N = 30$ observations, with real input data. The computational advantage of the CE method over LS is apparent, particularly in the required multiplications/divisions.

Table 2: Computational requirements of CE and LS methods (Real data, $L = 5$, $N = 30$).

Method	Add/Sub	Mult/Div
CE	34	11
LS	165	175

[4] It is only for matrices of this form, however, that eq. (19) corresponds to the CE method.

[5] The computational savings resulting from having pre-computed the $L \times N$ matrix $\Phi_1^{-1}\mathscr{X}_1$ instead are only of the order of L^2, which are negligible for large N.

REFERENCES

[1] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks,* vol. 4, no. 4, pp. 570–590, July 1993.

[2] S. N. Crozier, D. D. Falconer, and S. A. Mahmoud, "Least sum of squared errors (LSSE) channel estimation," *Proc. IEE–Part F,* vol. 138, no. 4, pp. 371–378, Aug. 1991.

[3] K. Georgoulakis and S. Theodoridis, "Efficient clustering techniques for channel equalization in hostile environments," *Signal Processing,* vol. 58, pp. 153–164, 1997.

[4] S. Haykin, *Adaptive Filter Theory,* 4th ed., Prentice-Hall, 2001.

[5] E. Kofidis, Y. Kopsinis, and S. Theodoridis, "On the least-squares performance of a novel efficient center estimation method for clustering-based MLSE equalization," under review in *IEEE Trans. Signal Processing.*

[6] Y. Kopsinis and S. Theodoridis, "Reduced-complexity clustering techniques for nonlinear channel equalization," in *Proc. WCC-2000 (ICSP),* Beijing, China, Aug. 2000.

[7] ——, "A novel low complexity cluster based MLSE equalizer for QPSK signaling scheme," *Proc. DSP-2002,* Santorini, Greece, July 2002.

[8] ——, "A novel cluster based MLSE equalizer for 2-PAM signaling scheme," in *Proc. EUSIPCO-2002,* Toulouse, France, Sept. 2002.

[9] ——, "An efficient low complexity technique for MLSE equalizers for linear and nonlinear channels," *IEEE Trans. Signal Processing,* vol. 51, no. 12, pp. 3236–3248, Dec. 2003.

[10] ——, "A novel cluster based MLSE equalizer for M-PAM signaling schemes," *Signal Processing,* vol. 83, pp. 1905–1918, 2003.

[11] J. R. Montalvão Filho, B. Dorizzi, and J. C. M. Mota, "Channel estimation by symmetrical clustering," *IEEE Trans. Signal Processing,* vol. 50, no. 6, pp. 1459–1469, June 2002.

[12] J. G. Proakis, *Digital Communications,* 3rd ed., McGraw-Hill, 1995.

[13] S. Theodoridis and K. Koutroumbas, *Pattern Recognition,* 2nd ed., Academic Press, 2003.