Recursive single-most-likely-replacement channel equaliser

W.-T. Chen
C.-Y. Chi

Indexing terms: Channel equalisation, Block signal processing, Maximum-likelihood sequence estimation, Single-most-likely-replacement algorithm

Abstract: The paper proposes a recursive single-most-likely-replacement (SMLR) equaliser, that is a fixed-lag block signal processing algorithm indexed by the block size L and the number of decisions $N \leqslant L$ at each recursion, for channels in the presence of intersymbol interference of finite or infinite length and additive white Gaussian noise. Both computational load and storage required by the proposed recursive SMLR equaliser are linearly proportional to the block size. Two simulation examples illustrate the performance of the proposed recursive SMLR equaliser.

1 Introduction

In digital communication systems the principal sources resulting in transmission error are intersymbol interference and additive noise. A great many channel equalisers have been reported to detect discrete symbols transmitted over a communication channel with intersymbol interference, and basically they can be classified into two major categories. Briefly, the first category consists of linear equalisers and decision-feedback equalisers (DFE) [1–3]; these detect digital symbols in a symbol-by-symbol manner. In general, equalisers of this category require low computational complexity at the expense of high symbol error rate (SER).

The second category includes various equalisation algorithms [4–15] based on maximum-likelihood sequence estimation (MLSE). Their SER is generally much lower than the SER of the first category. When the channel has a finite length v, Forney's maximum-likelihood sequence estimation via Viterbi algorithm (MLSE–VA) [4] provides the optimum solution for MLSE. However, both computational load and storage required by the Viterbi algorithm are proportional to the total number of states of trellis, which grows exponentially with the channel length. When the channel length becomes large, the algorithm becomes impractical. To reduce the complexity of Viterbi algorithm and retain much of the performance advantage of MLSE, various suboptimum algorithms have been reported. For instance, the complexity of Viterbi algorithm can be reduced by directly truncating the channel impulse response [5]. The computational complexity can also be reduced by shaping the channel into a short impulse response through prefiltering such as applying a linear (or decision-feedback) equaliser [6–8] prior to applying the Viterbi algorithm. However, either neglecting the residual interference terms or prefiltering operations may result in significant error propagation, and therefore can lead to high SER. The delayed decision-feedback sequence estimation (DDFSE) [9] is another method to reduce the number of states. The complexity of DDFSE is controlled by a finite positive integer ζ, the reduced memory (for Viterbi algorithm) of the channel. When $\zeta = 0$, the DDFSE algorithm reduces to a conventional DFE in the first class; when the memory v of channel is finite and $\zeta = v$, the DDFSE is equivalent to MLSE–VA; when $\zeta < v$, the larger ζ, the closer is the performance of the DDFSE algorithm to the performance of MLSE. However, as in Viterbi algorithm, the number of states of trellis of the DDFSE algorithm too increases exponentially with the parameter ζ. Recently, Williamson *et al.* [10] proposed a block decision feedback equaliser (block DFE) structure which is indexed by the block length L and the number of decisions $N \leqslant L$ made at each recursion. With the structure of block DFE, they also developed a family of equalisers, called (L, N)-DFE, based on maximum a posteriori decision criterion and another family of equalisers, called high SNR (L, N)-DFE for the case of high signal-to-noise ratio (SNR) based on the same criterion. The (1, 1)-DFE emulates a conventional DFE; the (L, N)-DFE can have performance arbitrarily close to that of MLSE as both L and N approach infinity. Although the block size L can be taken to be small even if the channel impulse response is long because, as in DDFSE, the tail of the channel could be significantly cancelled by the use of decision feedback, the complexity of the optimum decision procedure for both (L, N)-DFE and high SNR (L, N)-DFE increases exponentially with L, and therefore, the block length L can not be chosen large.

Channel equalisation in digital communications is similar to deconvolution problems in other areas such as seismology, speech processing, radio astronomy and ultrasonic nondestructive evaluation except for different characteristics of the channel input and the channel itself. Next, let us briefly review the concept of the well-known single-most-likely-replacement (SMLR) detection algorithm [16–18] which is the kernel of many blind maximum-likelihood deconvolution algorithms [18, 19]

Paper 1086K (E5), first received 21st July 1993 and in revised form 4th January 1994

The authors are with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, Republic of China

The research described in this paper was supported by the National Science Council under grant NSC82-0404-E-007-115.

because the new equaliser to be presented later is based on the same concept.

The SMLR algorithm proposed by Kormylo and Mendel is an offline iterative suboptimum maximum-likelihood algorithm which detects spike locations of a sparse spike train with random amplitudes modelled as a Bernoulli–Gaussian (B–G) signal. At each iteration, for a reference Bernoulli (binary) sequence, it computes a set of log-likelihood ratios of test sequences to a reference sequence where each test sequence differs from the reference sequence only at a single time location, and then replaces the reference sequence with the test sequence associated with the largest log-likelihood ratio. This algorithm performs well and is computationally efficient. Chi and Chen [20] further proposed a recursive B–G model based SMLR algorithm, a block signal processing algorithm, which not only is computationally efficient but also inherits all the performance advantages of the offline SMLR algorithm. Moreover, both computational load and total storage required by the recursive B–G model based SMLR algorithm are linearly proportional to the block size.

In this paper we propose a recursive SMLR equaliser for channels with intersymbol interference of finite or infinite length in the presence of additive white Gaussian noise. Basically, the proposed recursive SMLR equaliser can be thought of as a high SNR (L, N)-DFE which uses a suboptimum SMLR detection algorithm instead of the optimum decision procedure used in the block DFE [10]. However, both computational load and total storage required by the proposed recursive SMLR equaliser are proportional to $M \cdot L$ rather than M^L as required by the block DFE where M is the number of different symbols transmitted through the channel.

2 A recursive SMLR detection algorithm for channel equalisation

Consider the following baseband discrete-time channel model

$$z(j) = \mu(j) * v(j) + n(j) = \sum_{i=0}^{\infty} v(i)\mu(j-i) \tag{1}$$

where $z(j)$ is the noisy channel output, $v(j)$ is the channel impulse response, $\mu(j)$ is the channel input and $n(j)$ is additive channel noise. Assume that $\mu(j)$ is a sequence of discrete values drawn from a finite set $U = \{u_1, u_2, \ldots, u_M\}$ with equal probability $1/M$ and that $n(j)$ is white Gaussian with variance σ_n^2. The new recursive SMLR equaliser has the same basic structure as the block DFE [10] which is indexed by the block size L and the number of decisions $N \leqslant L$ made at each recursion. At the recursion k, the data block $\boldsymbol{z}_k = (z(k), z(k+1), \ldots, z(k+L-1))^T$ is processed to iteratively search for the optimum estimate $\hat{\boldsymbol{\mu}}_k = (\hat{\mu}(k), \hat{\mu}(k+1), \ldots, \hat{\mu}(k+L-1))^T$ of $\boldsymbol{\mu}_k = (\mu(k), \mu(k+1), \ldots, \mu(k+L-1))^T$. When the detection algorithm converges, the N elements of $\tilde{\boldsymbol{\mu}}_k = [I_N : O_{(L-N)}]\hat{\boldsymbol{\mu}}_k$ are the detected $(\mu(k), \mu(k+1), \ldots, \mu(k+N-1))^T$ where I_N is an $N \times N$ identity matrix, $O_{(L-N)}$ is an $N \times (L-N)$ matrix of zeros. Then the next block $\boldsymbol{z}_{k+N}$ is processed for the next recursion $k+N$. Next, let us present the new recursive SMLR equaliser in detail.

Assume that the convolutional model (eqn. 1) can also be represented in a pth-order state-variable form [10] as

$$\boldsymbol{x}(j) = \Phi \boldsymbol{x}(j-1) + \boldsymbol{\gamma}\mu(j) \tag{2}$$

$$z(j) = \boldsymbol{h}^T \boldsymbol{x}(j) + n(j) \tag{3}$$

where $\boldsymbol{x}(j)$, $\boldsymbol{\gamma}$ and $\boldsymbol{h}$ are $p \times 1$ vectors, Φ is a $p \times p$ matrix. Note that

$$v(j) = \begin{cases} \boldsymbol{h}^T \Phi^j \boldsymbol{\gamma} & j \geqslant 0 \\ 0 & j < 0 \end{cases} \tag{4}$$

and that given $v(j)$, there exist many $(\Phi, \boldsymbol{\gamma}, \boldsymbol{h})$s [21].

Let $\boldsymbol{e}_k = (e(k), e(k+1), \ldots, e(k+L-1))^T$ where

$$e(j) = z(j) - \hat{z}(j) = z(j) - \hat{\mu}(j) * v(j) \tag{5}$$

The new recursive SMLR equaliser tries to search for $\boldsymbol{\mu}_k = \hat{\boldsymbol{\mu}}_k$ such that the unconditional likelihood function [10]

$$S_k\{\boldsymbol{\mu}_k \,|\, \boldsymbol{z}_k\} = \frac{1}{(2\pi\sigma_n^2)^{L/2}} \exp\left\{-\frac{\boldsymbol{e}_k^T \boldsymbol{e}_k}{2\sigma_n^2}\right\} \times \left(\frac{1}{M}\right)^L \tag{6}$$

is maximum at the recursion k under the assumption that $\mu(j)$ for all $j < k$ have been correctly detected.

The signal processing procedure of the proposed recursive SMLR equaliser is shown in Fig. 1 which

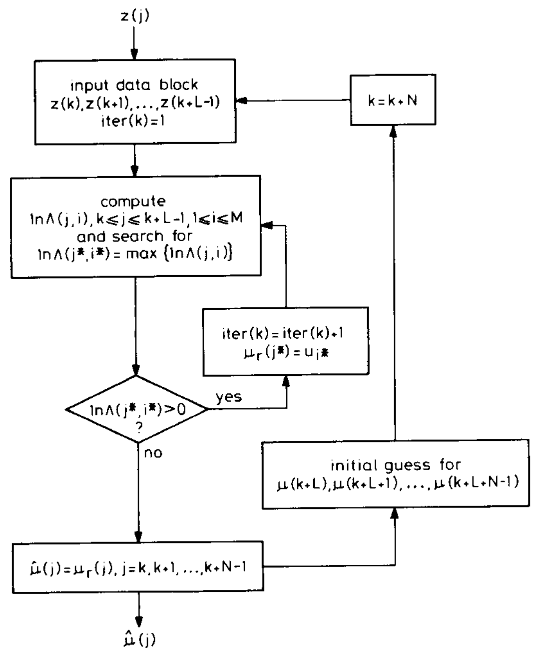

Fig. 1 *Signal processing procedure of the proposed recursive SMLR equaliser.*

includes an inner loop and an outer loop. The inner loop is an iterative detection procedure based on a set of log-likelihood ratios $\ln \Lambda(j, i)$ computed from the input data block $\boldsymbol{z}_k$. When the inner loop converges, only N elements of $\tilde{\boldsymbol{\mu}}_k = [I_N : O_{(L-N)}]\hat{\boldsymbol{\mu}}_k$ are the desired optimal estimates of $(\mu(k), \mu(k+1), \ldots, \mu(k+N-1))^T$. The outer loop is nothing but a reset procedure for the next recursion associated with S_{k+N} including initial guesses for $\boldsymbol{\mu}_{k+N}$ as well as a new data block $\boldsymbol{z}_{k+N}$. Next, let us present how the detection is performed in the inner loop.

Let $\Lambda(j, i)$ denote the likelihood ratio

$$\Lambda(j, i) = \frac{S_k\{\boldsymbol{\mu}_i^* \,|\, \boldsymbol{z}_k\}}{S_k\{\boldsymbol{\mu}_r^* \,|\, \boldsymbol{z}_k\}} \qquad \text{for } k \leqslant j \leqslant k+L-1 \quad \text{and } 1 \leqslant i \leqslant M \tag{7}$$

where $\boldsymbol{\mu}_r^* = (\mu_r(k), \mu_r(k+1), \ldots, \mu_r(k+L-1))^T$ is a reference sequence and $\boldsymbol{\mu}_i^* = (\mu_i(k) = \mu_r(k), \ldots, \mu_i(j-1) =$

$\mu_r(j-1),\ \mu_t(j)=u_i,\ \mu_t(j+1)=\mu_r(j+1),\ \ldots,\ \mu_t(k+L-1)=\mu_r(k+L-1))^T$ is a test sequence which differs from $\boldsymbol{\mu}_r^*$ only at a single time location j. During the recursion k, the proposed recursive SMLR equaliser shown in Fig. 1 searches for the optimal $\hat{\boldsymbol{\mu}}_k$ through the following procedure:

(*a*) Compute $\ln \Lambda(j, i)$ for all $(j, i) \in \{(j, i) \mid k \leqslant j \leqslant k+L-1, 1 \leqslant i \leqslant M\}$.

(*b*) Assume that $\ln \Lambda(j^*, i^*) = \max\{\ln \Lambda(j, i) \mid k \leqslant j \leqslant k+L-1,\ 1 \leqslant i \leqslant M\}$; if $\ln \Lambda(j^*, i^*) > 0$, update $\mu_r(j^*)$ by u_{i^*} and to to (*a*).

Note that $\ln \Lambda(j, i) = 0$ when $u_i = \mu_r(j)$. When $\ln \Lambda(j, i) \leqslant 0$ for all $(j, i) \in \{(j, i) \mid k \leqslant j \leqslant k+L-1, 1 \leqslant i \leqslant M\}$, the detection procedure is finished and the N elements of $\tilde{\boldsymbol{\mu}}_k = [I_N : O_{(L-N)}]\boldsymbol{\mu}_r^*$ are the desired estimates $\hat{\mu}(k), \hat{\mu}(k+1), \ldots, \hat{\mu}(k+N-1)$ while $\mu_r(k+N), \mu_r(k+N+1), \ldots, \mu_r(k+L-1)$ together with initial guesses for $\mu(k+L), \mu(k+L+1), \ldots, \mu(k+L+N-1)$ will be used as the initial values for $\boldsymbol{\mu}_{k+N}$ for the next recursion associated with S_{k+N}. For $N = 1$, one can arbitrarily select a $u_i \in U$ as the initial guess for $\mu(k+L)$, and for $N > 1$, any conventional DFE [1, 2, 9, 10] can be used to obtain initial values for $\mu(k+L)$, $\mu(k+L+1)$, $\ldots$, $\mu(k+N+L-1)$. The proposed recursive SMLR detection algorithm differs from Chi and Chen's recursive SMLR detection algorithm [20] in that the channel input $\mu(k)$ is an M-ary sequence for the former and a B–G sequence $\mu(j) = q(j) \cdot r(j)$ for the latter where $q(j)$ is a Bernoulli sequence with probability λ for $q(j) = 1$ and probability $1 - \lambda$ for $q(j) = 0$ and $r(j)$ is a white Gaussian sequence, and that the expressions needed for computing the key log-likelihood ratios $\ln \Lambda(j, i)$ are totally different for both. Next, let us present how to compute $\ln \Lambda(j, i)$.

The formula for computing $\ln \Lambda(j, i)$ which is needed in the detection procedure (*a*) can be shown to be

$$\ln \Lambda(j, i) = \{f(j) - \tfrac{1}{2}[u_i - \mu_r(j)]a(j)\}[u_i - \mu_r(j)] \tag{8}$$

where

$$f(j) = \boldsymbol{\gamma}^T \boldsymbol{w}(j) \tag{9}$$

and

$$a(j) = \boldsymbol{\gamma}^T C_w(j)\boldsymbol{\gamma} \tag{10}$$

in which $\boldsymbol{w}(j)$ is a $p \times 1$ vector and $C_w(j)$ is a $p \times p$ matrix. The vector $\boldsymbol{w}(j)$ and matrix $C_w(j)$ can be obtained by running the causal filter (in state-variable form)

$$\hat{\boldsymbol{x}}(j) = \Phi\hat{\boldsymbol{x}}(j-1) + \boldsymbol{\gamma}\mu_r(j) \tag{11}$$

$$\hat{z}(j) = \boldsymbol{h}^T\hat{\boldsymbol{x}}(j) \tag{12}$$

$$\tilde{z}(j) = z(j) - \hat{z}(j) \tag{13}$$

forwards from $j = k$ to $k + L - 1$ and then running the anticausal filter (in state-variable form)

$$\boldsymbol{w}(j) = \Phi^T\boldsymbol{w}(j+1) + \boldsymbol{h}\tilde{z}(j)/\sigma_n^2 \tag{14}$$

$$C_w(j) = \Phi^T C_w(j+1)\Phi + \boldsymbol{h}\boldsymbol{h}^T/\sigma_n^2 \tag{15}$$

backwards from $j = k + L - 1$ to k. The proof of eqns. 8–15 is given in Appendix 6. The initial condition $\hat{\boldsymbol{x}}(k-1)$ for eqn. 11 is associated with S_{k-N}, and thus is available prior to the time point k. The initial conditions for eqns. 14 and 15 are $\boldsymbol{w}(k+L) = \boldsymbol{0}$ (zero vector) and $C_w(k+L) = [0]$ (zero matrix), respectively. Next, let us discuss the computational efficiency for computing the quantities $f(j)$ and $a(j)$ needed to compute $\ln \Lambda(j, i)$ and their physical meaning.

Note that the state-variable equations given by eqns. 11 and 12 are the same as the state-variable equations given by eqns. 2 and 3 and that $\hat{z}(j)$ is nothing but the predicted $z(j)$ and $\tilde{z}(j)$ given by eqn. 13 is the associated prediction error. Therefore, when $\mu_r(j) = \mu(j)$ for all $k \leqslant j \leqslant k+L-1$ and $\mu(j)$ for all $j < k$ have been correctly detected by assumption, $\tilde{z}(j) = n(j)$ is a white Gaussian sequence with variance σ_n^2. Furthermore, eqns. 14 and 9 form an anticausal filter with the impulse response $v_b(j)$ given by

$$v_b(j) = \begin{cases} \boldsymbol{\gamma}^T(\Phi^T)^{-j}\boldsymbol{h} & \text{for } j \leqslant 0 \\ 0 & \text{for } j > 1 \end{cases} \tag{16}$$

Together with eqn. 4 this implies

$$v_b(j) = v(-j) \tag{17}$$

Therefore, $f(j)$ is the output of an anticausal matched filter associated with the channel $v(j)$ driven by $\tilde{z}(j)/\sigma_n^2$. On the other hand, $a(j)$ given by eqn. 10 is the variance of $f(j)$ because $C_w(j)$ given by eqn. 15 can be easily shown to be the covariance matrix of $\boldsymbol{w}(j)$ given by eqn. 14 when $\tilde{z}(j)$ is a white sequence with variance σ_n^2. Moreover, $\boldsymbol{a}_k = (a(k), a(k+1), \ldots, a(k+L-1))^T$ are the same for all k because they do not depend on the data block $\boldsymbol{z}_k$ and therefore can be computed and stored ahead of time as long as the channel parameters Φ, $\boldsymbol{\gamma}$ as well as $\boldsymbol{h}$ and noise variance σ_n^2 are known *a priori*. Therefore, the computational efficiency of the proposed recursive SMLR equaliser is due to the fact that only $f(j)$ for $j = k, k+1, \ldots, k+L-1$ need be computed in order to obtain the $L \times M$ log-likelihood ratios $\ln \Lambda(j, i)$ for $j = k, k+1, \ldots, k+L-1$ and $i = 1, 2, \ldots, M$ at each iteration. On the other hand, Chi and Chen's recursive SMLR detection algorithm [20] computes the corresponding $f(j)$ and $a(j)$ needed to compute the corresponding $\ln \Lambda(j, i)$ for $M = 2$ by an optimal Kalman smoother [17, 18, 22, 23] which is much more complicated than the linear filter $v(j)$ and the matched filter $v_b(j)$ used by the proposed recursive SMLR equaliser. Next, it is appropriate to address the distinctions between the proposed recursive SMLR equaliser and the block DFE as well as Viterbi algorithm based equalisers from the computation and storage points of view.

When $(L, N) = (1, 1)$, the proposed recursive SMLR equaliser reduces to a conventional DFE (equivalent to the DDFSE for $\zeta = 0$ and the (1, 1)-DFE) [1, 2, 9, 10]. The proposed recursive SMLR equaliser can also be thought of as a high SNR (L, N)-DFE with the optimum decision procedure replaced by the SMLR detection procedure. As previously discussed, the proposed recursive SMLR equaliser computes $L \times M$ log-likelihood ratios $\ln \Lambda(j, i)$ for all $(j, i) \in \{(j, i) \mid k \leqslant j \leqslant k+L-1, 1 \leqslant i \leqslant M\}$ by processing the data block $\boldsymbol{z}_k$ with a causal filter (same as the channel) (forward processing) followed by the associated anticausal matched filter (backward processing). However, computational load for the detection of $\boldsymbol{\mu}_k$ should be proportional to iter$(k) \times L$ where iter(k) is the number of iterations spent in the inner loop in the SMLR detection procedure shown in Fig. 1. Hence, both computational load and storage required by the proposed recursive SMLR equaliser are linearly proportional to the block size L. Moreover, the parameter L associated with the proposed recursive SMLR equalizer plays the role similar to the reduced memory ζ in DDFSE, the block length L in block DFE and the truncated channel length v in MLSE–VA. As previously mentioned, both storage and computational load required by equalisers based on Viterbi algorithm [4, 9] are proportional to the number of states of trellis (e.g. M^{v+1} in MLSE–VA, $M^{\zeta+1}$ in DDFSE). Hence, the proposed SMLR equaliser is much more practical than channel equalisers which use Viterbi algorithm for the detection of $\mu(j)$ from the view points of both computational load and storage, and therefore, it is particularly suitable for the case when the block length L must be chosen large in order to retain satisfactory performance.

3 Simulation

In this Section two simulation examples are provided to illustrate the performance of the recursive SMLR equaliser. In the simulation, a binary random sequence of $\{1, -1\}$ was generated which was then convolved with a selected channel impulse response (infinite-length channel in example 1 and finite-length channel in example 2) to obtain the simulation data of length more than 500000 for SNR equal to 8, 10, 12, 14 and 16 (in dB), respectively. Then $\mu(j)$ was estimated using the proposed SMLR equaliser. With the same simulation data for performance comparison, $\mu(j)$ was also estimated using high SNR (L, N)-DFE, the DDFSE algorithm, MLSE–VA and a conventional DFE [1, 2, 9, 10] which is equivalent to the DDFSE algorithm for $\zeta = 0$, (1, 1)-DFE as well as the proposed recursive SMLR equaliser for $(L, N) = (1, 1)$.

Example 1: (infinite-length channel)
A one-pole channel with the infinite impulse response $v(j) = 0.9^j$ (taken from Reference 9), which can be expressed as the following state-variable model:

$$x(j) = 0.9x(j-1) + \mu(j) \quad (18)$$

$$z(j) = x(j) + n(j) \quad (19)$$

was used in this example. Then the unknown $\mu(j)$ was estimated using both the proposed recursive SMLR equaliser and the DDFSE algorithm. Notice that the block size L for the former corresponds to the reduced memory $\zeta = L - 1$ for the latter. The simulation results are shown in Fig. 2, where crosses and plusses denote

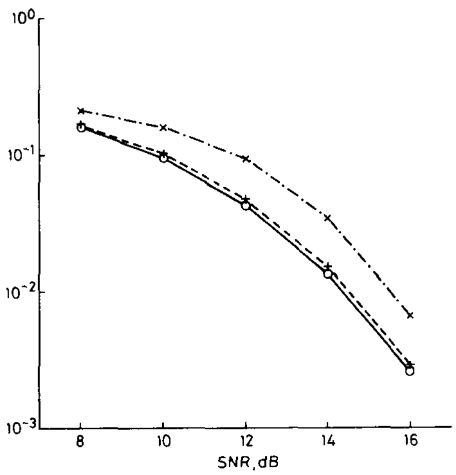

Fig. 2 *Simulation results for the infinite one-pole channel with impulse response $v(j) = 0.9^j$ for $j > 0$*
+—+ and ×—× SERs associated with the proposed recursive SMLR equaliser for $(L, N) = (6, 1)$ and $(L, N) = (1, 1)$, respectively
O—O SERs associated with the DDFSE algorithm for the reduced channel memory $\zeta = 5$ (corresponding to $L = 6$)

SERs associated with the proposed recursive SMLR equaliser for $(L, N) = (1, 1)$ and $(L, N) = (6, 1)$, respectively, and circles denote SERs associated with the DDFSE algorithm for $\zeta = 5$. Note, from this figure, that the results obtained by the proposed recursive SMLR equaliser for $(L, N) = (6, 1)$ are quite close to those obtained by the DDFSE algorithm for $\zeta = 5$ and much better than the results for $(L, N) = (1, 1)$ which, as previously mentioned, are equivalent to the results associated with the above mentioned conventional DFE. These results indicate that the performance of the proposed recursive SMLR equaliser is comparable to the performance of the DDFSE algorithm for the same block length and both of them outperform the conventional DFE. On the other hand, the average of iter(k) over $k = 1, 2, \ldots, 500000$ lies between 1.5 and 1.6 for each result (associated with the proposed recursive SMLR equaliser) shown in Fig. 2. A qualitative explanation for this is that when the initial values of $\mu(k), \mu(k+1), \ldots, \mu(k+L-2)$ are all correct, the number of iterations spent for estimating $\mu(k+L-1)$ is equal to 1.5 on average, and that only a few percent of the initial values for $\mu(j)$, for $j = k, k+1, \ldots, k+L-2$ (taken from the previous recursion $k - N$) are incorrect on average. Therefore, the computational load required by the proposed recursive SMLR equaliser is proportional to iter$(k) \times L \leqslant 1.6 \times L < 2^L$ for this case.

Example 2: (finite-length channel)
A channel with a finite-length impulse response $v(j) = 0.9^j$ for $j = 0, \ldots, 5$ was used in this example. The simulation results are shown in Fig. 3, where crosses and

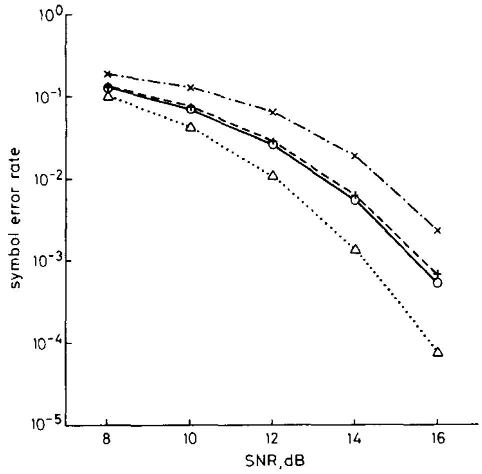

Fig. 3 *Simulation results for the channel with finite impulse response $v(j) = 0.9^j$ for $j = 0, 1, \ldots, 5$*
+—+ and ×—× SERs associated with the proposed recursive SMLR equaliser for $(L, N) = (6, 1)$ and $(L, N) = (1, 1)$, respectively
O—O SNRs associated with high SNR(6, 1)-DFE
△··△ SERs associated with the MLSE–VA

plusses denote SERs associated with the proposed recursive SMLR equaliser for $(L, N) = (1, 1)$ (a conventional DFE) and $(L, N) = (6, 1)$, respectively, circles denote SERs associated with the high SNR (6, 1)-DFE and triangles denote SERs associated with MLSE–VA [4]. One can observe, from this Figure, that the results obtained by the proposed recursive SMLR equaliser for $(L, N) = (6, 1)$ are quite close to those obtained by the high SNR (6, 1)-DFE and much better than the results obtained by the conventional DFE. However, the results associated with MLSE–VA are much better than those associated with the proposed recursive SMLR equaliser and those associated with the high SNR (6, 1)-DFE. The average of iter(k) too lies between 1.5 and 1.6 for each result associated with the proposed SMLR equaliser, which, again, indicates that the computational complexity required by the proposed recursive SMLR equaliser is proportional to $\text{iter}(k) \times L \leqslant 1.6 \times L \leqslant 2^L$ for this case as well.

4 Conclusions

We have presented a recursive SMLR equaliser for channels in the presence of intersymbol interference of finite or infinite length and additive white Gaussian noise. The proposed recursive SMLR equaliser has a structure similar to Chi and Chen's recursive SMLR deconvolution algorithm [20] although they are designed for different purposes. The proposed recursive SMLR equaliser is a block signal processing algorithm shown in Fig. 1 which processes the data block z_k to decide the first N estimates of $\boldsymbol{\mu}_k$ based on the $L \times M$ log-likelihood ratios $\ln \Lambda(j, i)$ given by eqn. 8 for $k \leqslant j \leqslant k + L - 1$ and $1 \leqslant i \leqslant M$ where M is the number of different symbols for $\mu(j) \in U$. The computation of these $L \times M$ log-likelihood ratios $\ln \Lambda(j, i)$ can be done quite efficiently by running once a causal filter (same as the channel) given by eqns. 11 and 12 followed by an anticausal matched filter given by eqns. 14 and 9 associated with the channel.

The proposed recursive SMLR equaliser can also be viewed as a high SNR (L, N)-DFE with the optimum decision procedure replaced by the SMLR detection procedure. The presented two simulation examples showed that for an infinite-length one-pole channel, the performance of the proposed recursive SMLR equaliser is comparable to the performance of the DDFSE algorithm for the same block size, and that, for a finite-length channel, the performance of the former is quite close to the suboptimum high SNR block DFE while the optimum MLSE–VA performs best. However, the computational load and storage required by the proposed recursive SMLR equaliser are linearly proportional to the block size L, while those required by the block DFE, DDFSE and Viterbi algorithm based channel equalisers increase exponentially with block size, reduced channel length, and channel length (or truncated channel length), respectively.

5 References

1 AUSTIN, M.E.: 'Decision-feedback equalization for digital communication over dispersive channels' (MIT Lincoln Laboratory, Lexington, MA, Aug. 1967), Technical report 437

2 CLARK, A.P.: 'Equalizers for digital modems' (John Wiley & Sons, New York, 1985)

3 PROAKIS, J.G.: 'Digital communications' (McGraw-Hill Book Company, New York, 1989)

4 FORNEY, G.D. Jr.: 'Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference', *IEEE Trans. Inform. Theory*, May 1972, **IT-18**, pp. 363–378

5 McLANE, P.J.: 'A residual interference error bound for truncated-state Viterbi detectors', *IEEE Trans. Inform. Theory*, Sept. 1980, **IT-16**, pp. 549–553

6 FALCONER, D.D., and MAGEE, F.R. Jr.: 'Adaptive channel memory truncation for maximum likelihood sequence estimation', *Bell Syst. Tech. J.*, Nov. 1973, **52**, pp. 1541–1562

7 BEARE, C.T.: 'The choice of the desired impulse response in combined linear-Viterbi algorithm equalizers', *IEEE Trans. Commun.*, Aug. 1978, **COM-26**, pp. 1301–1307

8 LEE, W.U., and HILL, F.S.: 'A maximum-likelihood sequence estimator with decision-feedback equalization', *IEEE Trans. Commun.*, Sept. 1977, **COM-25**, pp. 971–979

9 DUEL, A., and HEEGARD, C.: 'Delayed decision feedback sequence estimation', *IEEE Trans. Commun.*, May 1989, **37**, pp. 428–436

10 WILLIAMSON, D., KENNEDY, R.A., and PULFORD, G.W.: 'Block decision feedback equalization', *IEEE Trans. Commun.*, Feb. 1992, **40**, (2), pp. 255–264

11 WOZENCRAFT, J.M.: 'Sequential decoding for reliable communications'. 'MIT-RLE technical report', TR325, 1957

12 HACCOUN, D., and FERGUSON, M.J.: 'Generalized stack algorithm for decoding convolutional codes', *IEEE Trans. Inform. Theory*, Nov. 1975, **IT-21**, pp. 638–651

13 CHEVILLAT, P.R., and COSTELLO, D.J. Jr.: 'A multiple stack algorithm for erasurefree decoding of convolutional codes', *IEEE Trans. Commun.*, Dec. 1990, **COM-25**, (12), pp. 1460–1470

14 XIONG, F., ZERIK, A., and SHWEDYK, E.: 'Sequential sequence estimation for channels with intersymbol interference of finite or infinite length', *IEEE Trans. Commun.*, June 1990, **38**, (6), pp. 795–804

15 VITERBI, A.J.: 'Error bounds for convolutional codes and an asymptotically optimum decoding algorithm', *IEEE Trans. Inform. Theory*, Apr. 1967, **IT-13**, pp. 260–269

16 KORMYLO, J., and MENDEL, J.M.: 'Maximum-likelihood detection and estimation of Bernoulli–Gaussian processes', *IEEE Trans. Inf. Theory*, 1982, **IT-28**, pp. 482–488

17 MENDEL, J.M.: 'Optimal seismic deconvolution: an estimation-based approach' (Academic Press, New York, 1983)

18 MENDEL, J.M.: 'Maximum-likelihood deconvolution: a journey into model-based signal processing' (Springer-Verlag, New York, 1990)

19 CHI, C.-Y., MENDEL, J.M., and HAMPSON, D.: 'A computationally-fast approach to maximum-likelihood deconvolution', *Geophysics*, May 1984, **49**, (5), pp. 550–565

20 CHI, C.-Y., and CHEN, W.-T.: 'A recursive SMLR deconvolution algorithm for Bernoulli–Gaussian signals', *IEE Proc. F, Radar Signal Process.*, June 1991, **138**, (3), pp. 263–266

21 TRETTER, S.A.: 'Introduction to discrete-time signal processing' (Wiley, New York, 1976)

22 MENDEL, J.M.: 'Lessons in digital estimation theory' (Prentice-Hall, Englewood Cliffs, NJ, 1987)

23 CHI, C.-Y., and CHEN, W.-T.: 'An adaptive maximum-likelihood deconvolution algorithm', *Signal Process.*, Aug. 1991, **24**, (2), pp. 149–163

6 Appendix: Proof of eqns. 8–15 for computing the log-likelihood ratio ln $\Lambda(j, i)$

Since the log function is a monotonically increasing function, maximizing $S_k\{\boldsymbol{\mu}_k \mid z_k\}$ is equivalent to maximising the log-likelihood function as follows

$$\ln(S_k\{\boldsymbol{\mu}_k \mid z_k\}) = -\frac{L}{2}[\ln(2\pi) + 2\ln(\sigma_n) + 2\ln(M)] - \frac{1}{2\sigma_n^2} \boldsymbol{e}_k^T \boldsymbol{e}_k \quad (20)$$

where

$$\boldsymbol{e}_k = z_k - V\boldsymbol{\mu}_k^+ \quad (21)$$

in which

$$\boldsymbol{\mu}_k^+ = (\mu(0), \mu(1), \ldots, \mu(k + L - 1))^T \quad (22)$$

and V is an $L \times (k + L)$ matrix

$$V = \begin{bmatrix} v(k) & v(k-1) & \cdots & v(0) & 0 & \cdots & 0 \\ v(k+1) & v(k) & \cdots & v(1) & v(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ v(k+L-1) & v(k+L-2) & \cdots & v(L-1) & v(L-2) & \cdots & v(0) \end{bmatrix} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k+L-1}] \quad (23)$$

where $\boldsymbol{v}_i$ is the ith column of V.

From eqns. 7, 20 and 21, we have

$$\ln \Lambda(j, i) = \ln(S_k\{\boldsymbol{\mu}_i^* \mid z_k\}) - \ln(S_k\{\boldsymbol{\mu}_r^* \mid z_k\}) = \frac{1}{2\sigma_n^2}(\boldsymbol{\eta}_r^T \boldsymbol{\eta}_r - \boldsymbol{\eta}_i^T \boldsymbol{\eta}_i) \quad (24)$$

where

$$\boldsymbol{\eta}_r = (\tilde{z}(k), \tilde{z}(k+1), \ldots, \tilde{z}(k+L-1))^T = \boldsymbol{z}_k - V\boldsymbol{\mu}_r^+ \quad (25)$$

and

$$\boldsymbol{\eta}_t = \boldsymbol{z}_k - V\boldsymbol{\mu}_t^+ \quad (26)$$

in which

$$\boldsymbol{\mu}_r^+ = (\hat{\mu}(0), \ldots, \hat{\mu}(k-1), \mu_r(k), \mu_r(k+1), \ldots, \mu_r(k+L-1))^T \quad (27)$$

and

$$\boldsymbol{\mu}_t^+ = (\hat{\mu}(0), \ldots, \hat{\mu}(k-1), \mu_r(k), \ldots, \mu_r(j-1), u_i, \mu_r(j), \ldots, \mu_r(k+L-1))^T \quad (28)$$

One can see, from eqns. 21, 25 and 5, that $\tilde{z}(j)$ can also be expressed as eqn. 13 where $\hat{z}(j)$ is nothing but the output of the causal filter with the same impulse response as the channel. Therefore, $\hat{z}(j)$ can be computed with the same state-variable equations given by eqns. 11 and 12 as those for the channel.

The log-likelihood ratio $\ln \Lambda(j, i)$ given by eqn. 24 can further be expressed as

$$\ln \Lambda(j, i) = \frac{1}{2\sigma_n^2}[(\boldsymbol{\eta}_r - \boldsymbol{\eta}_t)^T\boldsymbol{\eta}_r + (\boldsymbol{\eta}_t - \boldsymbol{\eta}_r)^T \times (\boldsymbol{\eta}_r - \boldsymbol{\eta}_t) + \boldsymbol{\eta}_r^T(\boldsymbol{\eta}_r - \boldsymbol{\eta}_t)] \quad (29)$$

From eqns. 23, 25 and 26, we have

$$\boldsymbol{\eta}_r - \boldsymbol{\eta}_t = \boldsymbol{z}_k - V\boldsymbol{\mu}_r^+ - \boldsymbol{z}_k + V\boldsymbol{\mu}_t^+ = \boldsymbol{v}_j[\mu_t(j) - \mu_r(j)] = \boldsymbol{v}_j[u_i - \mu_r(j)] \quad (30)$$

since $\boldsymbol{\mu}_t^+$ differs from $\boldsymbol{\mu}_r^+$ only in the $(j+1)$th component. Now, substituting eqn. 30 into eqn. 29, we obtain

$$\ln \Lambda(j, i) = \frac{1}{2\sigma_n^2}\{[u_i - \mu_r(j)]\boldsymbol{v}_j^T\boldsymbol{\eta}_r - [u_i - \mu_r(j)]^2\boldsymbol{v}_j^T\boldsymbol{v}_j + [u_i - \mu_r(j)]\boldsymbol{\eta}_r^T\boldsymbol{v}_j\}$$
$$= \{f(j) - \tfrac{1}{2}[u_i - \mu_r(j)]a(j)\}[u_i - \mu_r(j)] \quad (31)$$

where

$$f(j) = \boldsymbol{v}_j^T\boldsymbol{\eta}_r/\sigma_n^2 \quad \text{for } j = k, k+1, \ldots, k+L-1 \quad (32)$$

and

$$a(j) = \boldsymbol{v}_j^T\boldsymbol{v}_j/\sigma_n^2 \quad \text{for } j = k, k+1, \ldots, k+L-1 \quad (33)$$

Therefore, we have proven that the log-likelihood ratio $\ln \Lambda(j, i)$ given by eqn. 8 is true. Next, let us show that $f(j)$ given by eqn. 32 can be obtained by eqns. 9 and 14 and that $a(j)$ given by eqn. 33 can be obtained by eqns. 10 and 15.

It can be easily shown, from eqns. 14 and 15, that

$$\boldsymbol{w}(j) = \sum_{i=j}^{k+L-1} (\Phi^T)^{i-j}\boldsymbol{h}\tilde{z}(i)/\sigma_n^2 \quad (34)$$

and

$$C_w(j) = \sum_{i=j}^{k+L-1} (\Phi^T)^{i-j}\boldsymbol{h}\frac{1}{\sigma_n^2}\boldsymbol{h}^t\Phi^{i-j} \quad (35)$$

From eqns. 32, 25, 23, 4 and 34, we have

$$f(j) = \sum_{i=j}^{k+L-1} \tilde{z}(i)v(i-j)/\sigma_n^2 = \sum_{i=j}^{k+L-1} \boldsymbol{h}^T\Phi^{i-j}\boldsymbol{\gamma}\tilde{z}(i)/\sigma_n^2$$
$$= \gamma^T \sum_{i=j}^{k+L-1} (\Phi^T)^{i-j}\boldsymbol{h}\tilde{z}(i)/\sigma_n^2 = \gamma^T\boldsymbol{w}(j) \quad (36)$$

which implies that $f(j)$ can be obtained from the anticausal state-variable model given by eqns. 14 and 9. Finally, from eqns. 33, 23, 4 and 35, we have

$$a(j) = \sum_{i=j}^{k+L-1} v^2(i-j)/\sigma_n^2$$
$$= \gamma^T\left[\sum_{i=j}^{k+L-1} (\Phi^T)^{i-j}\boldsymbol{h}\frac{1}{\sigma_n^2}\boldsymbol{h}^t\Phi^{i-j}\right]\gamma$$
$$= \gamma^T C_w(j)\gamma \quad (37)$$

which implies that $a(j)$ can be obtained from eqns. 15 and 10.