Adaptive minimum-BER decision feedback equalisers for binary signalling

B. Mulgrew† *and S. Chen*‡

† Department of Electronics Electrical Engineering The University of Edinburgh Edinburgh EH9 3JL, Scotland, U.K. (corresponding author)
‡ Department of Electronics and Computer Science University of Southampton, Highfield Southampton SO17 1BJ, U.K.

Abstract

The problem of constructing adaptive minimum bit error rate (MBER) decision feedback equalisers (DFE's) for binary signalling is considered. Gradient and Gauss-Newton algorithms are considered for both conventional and state (or space) translation forms of the DFE. The Hessian matrix for the Gauss-Newton algorithm is introduced for the first time. Kernel density estimation is demonstrated to provide a convenient mechanism for approximating the BER as a smooth function of the available data. This leads to the development of a number of block and serial adaptive algorithms. Computer simulation is used to assess the performance of these algorithms.

Keywords

minimum bit error rate, decision feedback equaliser, binary, state-translation, kernel density estimation, Gauss-Newton

I. Introduction

In developing training strategies for linear combiner decision feedback equalisers (DFE's), it is convenient to adopt a minimum mean squared error (MMSE) cost function as this facilitates the use of standard adaptive filter techniques such as the least mean squares (LMS) and recursive least squares (RLS) algorithms. However it has long been understood that the MMSE cost function is not optimal in this application - the minimum bit error rate (MBER) cost function being the more appropriate choice [2]. Further, the BER rate of a DFE optimised using a MMSE criterion can be distinctly inferior to the true optimum solution [1].

Two methods currently exist which can be classified as training algorithms for minimum BER DFE's in binary signalling. These are: the space or state translation algorithm of [8] and [1]; the approximate minimum bit error rate (AMBER) algorithm of [5]. The former is a batch or block adaptive filter in that the channel is estimated and this estimate used in turn to form an estimate of the theoretical BER. A gradient search technique is used to find the weights that minimise the estimated BER. While this algorithm has been demonstrated to work reliably in the presence of errors in the estimate of the channel impulse response, it is not truly adaptive in the sense that it does not operate directly on the data but rather employs the intermediate step of channel impulse response estimation. Further the gradient search algorithm can take many iterative steps to converge to the solution. The latter is a stochastic gradient algorithm which is identical to the signed error LMS algorithm except in the vicinity of the decision boundary where it is modified to continue updating the weights when the signed LMS algorithm would not. The algorithm is appealing due to its computational simplicity and straightforward extension to the complex signalling case [4]. The difficulty with it is in selection of the step-size: a small step size is required to ensure convergence to the min BER solution with the associated problem of slow initial convergence. A variant of the dual sign algorithm [7] has been applied in an attempt to overcome this problem [4].

In this study the problem of constructing adaptive MBER DFE's for binary signalling is addressed. Section II provides the necessary background and definition of terms. In Section III gradient and Gauss-Newton algorithms are developed for the conventional DFE rather the translated form of [1]. In particular the Hessian matrix for the Gauss-Newton algorithm has not appeared before. However the formulation is such that it can be applied to both conventional and state-translation architectures. Kernel density estimation is employed is section IV to approximate the bit error rate as a smooth function of the data. The advantage of this approach is that an error does not need to be observed to guarantee an estimate of the error rate and the smooth function is a convenient route to gradient and Gauss-Newton algorithms. Block and serial adaptive algorithms in both conventional and state-translation form are presented. Section V provides an assessment of the various algorithms based on computer simulations. Finally conclusions are drawn in section VI.

II. Background

The channel is modelled as a finite impulse response filter with an additive noise source, and the received signal at sample k is

$$r(k) = \bar{r}(k) + e(k) = \sum_{i=0}^{n_a-1} a_i s(k-i) + e(k) \tag{1}$$

where $\bar{r}(k)$ denotes the noiseless channel output; n_a is the channel length and a_i are the channel tap weights; the Gaussian white noise $e(k)$ has zero mean and variance σ_e^2; the symbol sequence $\{s(k)\}$ is independently identically distributed and has a 2-PAM constellation. The signal to noise ratio (SNR) of the system is defined as

$$SNR = E[\bar{r}^2(k)]/E[e^2] = \sigma_s^2 (\sum_{i=0}^{n_a-1} a_i^2)/\sigma_e^2 \tag{2}$$

where $\sigma_s^2 = E[s^2(k)]$ is the symbol variance.

For a conventional linear-combiner DFE the decision variable z at time k is a linear combination of received samples and past decisions:

$$z(k) = \mathbf{w}^T \mathbf{r}(k) - \mathbf{b}^T \hat{\mathbf{s}}_b(k) \tag{3}$$

where $\mathbf{r}(k) = [r(k)\, r(k-1) \ldots r(k-m+1)]^T$ is the channel observation vector, $\hat{\mathbf{s}}_b(k) = [\hat{s}(k-d-1)\, \hat{s}(k-d-2) \ldots \hat{s}(k-d-n)]^T$ is the past detected symbol vector, $\mathbf{w} = [w_0\, w_1 \ldots w_{m-1}]^T$ is the feedforward coefficient vector and $\mathbf{b} = [b_1\, b_2 \ldots b_n]^T$ is the feedback coefficients vector. The integers d, m and n will be referred to as the decision delay, the feedforward delay and feedback orders respectively. Without loss of generality, $d = n_a - 1$, $m = n_a$ and $n = n_a - 1$ will be used as this choice of DFE structure parameters is sufficient to guarantee the linear separability of the subsets of the channel states related to the different decisions [1]. Alternatively the linear-combiner DFE can be expressed in state translated form [9]:

$$\begin{aligned} z(k) &= \mathbf{w}^T (\mathbf{r}(k) - \mathbf{F}_2 \hat{\mathbf{s}}_b(k)) \\ &= \mathbf{w}^T \mathbf{r}'(k) \end{aligned} \tag{4}$$

where $\mathbf{F}_2$ is constructed by partitioning the channel impulse response matrix $\mathbf{F} = [\mathbf{F}_1\, \mathbf{F}_2]$, where:

$$\mathbf{F}_1 = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n_a-1} \\ 0 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ 0 & \cdots & 0 & a_0 \end{bmatrix} \tag{5}$$

$$\mathbf{F}_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_{n_a-1} & 0 & \ddots & \vdots \\ a_{n_a-2} & a_{n_a-1} & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ a_1 & \cdots & a_{n_a-2} & a_{n_a-1} \end{bmatrix} \tag{6}$$

In the case of 2-PAM, the reconstructed symbols with lag d are obtained from the sign of the decision variable $z(k)$ i.e.

$$\hat{s}(k-d) = \mathrm{sgn}(z(k)) \tag{7}$$

III. Minimum bit error rate equalisation

The bit error rate (BER) observed at the output of the equaliser is dependent on the distribution of the decision variable $z(k)$ which in turn is a function of the equaliser tap weights. To be more specific, the probability of error, P_E, is:

$$P_E = P(\mathrm{sgn}(s(k-d))z(k) < 0) \tag{8}$$

The sign adjusted decision variable $z_s(k) = \mathrm{sgn}(s(k-d))z(k)$ is drawn from a Gaussian mixture. From the definition of $z(k)$,

$$z_s(k) = \mathrm{sgn}(s(k-d))(\mathbf{w}^T\mathbf{F}\mathbf{s}(k) - \mathbf{b}^T\hat{\mathbf{s}}_b(k)) + \mathrm{sgn}(s(k-d))\mathbf{w}^T\mathbf{e}(k) \tag{9}$$

$$= \mathrm{sgn}(s(k-d))z'(k) + e'(k) \tag{10}$$

$\mathbf{e}(k) = [e(k)\, e(k-1) \ldots e(k-d-n)]^T$ is the vector of noise samples; $\mathbf{s}(k) = [s(k)\, s(k-1) \ldots s(k-d-n_{n_a})]^T$ is the vector of transmitted symbols. The first term on the right hand side of (10), $\mathrm{sgn}(s(k-d))z'(k)$, is the noise-free sign-adjusted equaliser output and is a member of a finite set with N_z elements - these are the local means of the Gaussian mixture. Without noise the combination of channel and DFE is a finite state machine whose state is defined by the vector $\mathbf{s}(k)$. Thus if $\mathbf{s}(k) \in \{\mathbf{s}_1 \cdots \mathbf{s}_i \cdots \mathbf{s}_{N_z}\}$, the state $\mathbf{s}_i$ uniquely defines the state of $z'(k)$, $\mathbf{r}(k)$, $s(k-d)$ and $\hat{\mathbf{s}}_b(k)$ - label these z_i, $\mathbf{r}_i$, s_i and $\hat{\mathbf{s}}_{bi}$ respectively. Note that while $\mathbf{s}(k)$ has N_z states, $s(k-d)$ has 2 possible values (2-PAM). However since $s(k-d)$ is a component of the vector $\mathbf{s}(k)$, the state of $\mathbf{s}(k)$ uniquely defines the value of $s(k-d)$. The second term $e'(k)$ is a zero mean Gaussian white noise process with variance $\sigma_e^2\mathbf{w}^T\mathbf{w}$ - defining the distribution about the local means.

The probability density function $p_z(z_s)$ is thus:

$$p_z(z_s) = \frac{1}{N_z\sqrt{2\pi}\sigma_e\sqrt{\mathbf{w}^T\mathbf{w}}}\sum_{i=1}^{N_z}\exp\left(-\frac{(z_s - \mathrm{sgn}(s_i)z_i)^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right) \tag{11}$$

and the probability of error is:

$$P_E = \int_{-\infty}^{0} p_z(z_s)dz_s \tag{12}$$

As in [1] the weights that minimise P_E can be obtained using gradient search techniques. Here the approach is "equaliser output referenced" i.e. in terms of $z(k)$ rather than the "equaliser input referenced" technique discussed in [1] that is in terms of the noise free input vectors. There is a natural computational advantage in dealing with scalars rather than vectors. Further the development is in terms of the conventional feedforward and feedback coefficients rather than the channel estimate associated with [1]. Thus the gradient terms are:

$$\begin{aligned}\frac{\partial P_E}{\partial \mathbf{w}} &= \frac{1}{N_z\sqrt{2\pi}\sigma_e}\left(\frac{\mathbf{w}\mathbf{w}^T}{(\mathbf{w}^T\mathbf{w})^{3/2}} - \frac{\mathbf{I}}{(\mathbf{w}^T\mathbf{w})^{1/2}}\right)\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\operatorname{sgn}(s_i)\mathbf{r}_i \\ &\quad -\frac{\mathbf{w}\mathbf{b}^T}{N_z\sqrt{2\pi}\sigma_e(\mathbf{w}^T\mathbf{w})^{3/2}}\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\operatorname{sgn}(s_i)\hat{\mathbf{s}}_{bi} \qquad (13)\end{aligned}$$

and

$$\frac{\partial P_E}{\partial \mathbf{b}} = \frac{1}{N_z\sqrt{2\pi}\sigma_e\sqrt{\mathbf{w}^T\mathbf{w}}}\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\operatorname{sgn}(s_i)\hat{\mathbf{s}}_{bi} \qquad (14)$$

A gradient search technique would thus be defined:

$$\mathbf{w}_{j+1} = \mathbf{w}_j - \mu\left[\frac{\partial P_E}{\partial \mathbf{w}}\right]_j \qquad (15)$$

and

$$\mathbf{b}_{j+1} = \mathbf{b}_j - \mu\left[\frac{\partial P_E}{\partial \mathbf{b}}\right]_j \qquad (16)$$

where μ is the step-size.

As usual with gradient based algorithms the convergence can be slow, taking many iterations of the algorithm to reach a solution. A Gauss-Newton algorithm is one way to improve the convergence performance. However the Gauss Newton technique requires evaluation of the Hessian. The Hessian for the minimum BER linear DFE has four distinct terms. These are:

$$\begin{aligned}\frac{\partial^2 P_E}{\partial \mathbf{w}^T\partial \mathbf{w}} &= \frac{1}{N_z\sqrt{2\pi}\sigma_e}[\left(\frac{\mathbf{I}}{(\mathbf{w}^T\mathbf{w})^{3/2}} - 3\frac{\mathbf{w}\mathbf{w}^T}{(\mathbf{w}^T\mathbf{w})^{5/2}}\right)\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\operatorname{sgn}(s_i)z_i \\ &\quad +\frac{\mathbf{w}}{(\mathbf{w}^T\mathbf{w})^{3/2}}\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\operatorname{sgn}(s_i)\mathbf{r}_i^T\right\} \\ &\quad +\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\operatorname{sgn}(s_i)\mathbf{r}_i\right\}\frac{\mathbf{w}^T}{(\mathbf{w}^T\mathbf{w})^{3/2}}\end{aligned}$$

$$
\begin{aligned}
&+\frac{\mathbf{w}\mathbf{w}^T}{\sigma_e^2(\mathbf{w}^T\mathbf{w})^{7/2}}\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\mathrm{sgn}(s_i)z_i^3\right\} \\
&-\frac{\mathbf{w}}{\sigma_e^2(\mathbf{w}^T\mathbf{w})^{5/2}}\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\mathrm{sgn}(s_i)z_i\mathbf{r}_i^T\right\} \\
&-\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\mathrm{sgn}(s_i)z_i\mathbf{r}_i\right\}\frac{\mathbf{w}^T}{\sigma_e^2(\mathbf{w}^T\mathbf{w})^{5/2}} \\
&+\frac{1}{\sigma_e^2}\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\mathrm{sgn}(s_i)z_i\mathbf{r}_i\mathbf{r}_i^T\right\}]
\end{aligned}
\tag{17}
$$

$$
\begin{aligned}
\frac{\partial^2 P_E}{\partial\mathbf{b}^T\partial\mathbf{w}} = \frac{1}{N_z\sqrt{2\pi}\sigma_e}[&-\frac{\mathbf{w}}{(\mathbf{w}^T\mathbf{w})^{3/2}}\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\mathrm{sgn}(s_i)\hat{\mathbf{s}}_{bi}^T\right\} \\
&+\frac{\mathbf{w}}{\sigma_e^2(\mathbf{w}^T\mathbf{w})^{5/2}}\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\mathrm{sgn}(s_i)z_i^2\hat{\mathbf{s}}_{bi}^T\right\} \\
&-\frac{1}{\sigma_e^2(\mathbf{w}^T\mathbf{w})^{3/2}}\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\mathrm{sgn}(s_i)\mathbf{r}_i z_i\hat{\mathbf{s}}_{bi}^T\right\}]
\end{aligned}
\tag{18}
$$

$$
\frac{\partial^2 P_E}{\partial\mathbf{w}^T\partial\mathbf{b}} = \left(\frac{\partial^2 P_E}{\partial\mathbf{b}^T\partial\mathbf{w}}\right)^T \tag{19}
$$

$$
\frac{\partial^2 P_E}{\partial\mathbf{b}^T\partial\mathbf{b}} = \frac{1}{N_z\sqrt{2\pi}\sigma_e}\frac{1}{\sigma_e^2(\mathbf{w}^T\mathbf{w})^{3/2}}\left\{\sum_{i=1}^{N_z}\exp\left(-\frac{z_i^2}{2\sigma_e^2\mathbf{w}^T\mathbf{w}}\right)\mathrm{sgn}(s_i)z_i\hat{\mathbf{s}}_{bi}\hat{\mathbf{s}}_{bi}^T\right\} \tag{20}
$$

The Gauss-Newton algorithm is thus:

$$
\begin{bmatrix}\mathbf{w}_{j+1}\\ \mathbf{b}_{j+1}\end{bmatrix} = \begin{bmatrix}\mathbf{w}_j\\ \mathbf{b}_j\end{bmatrix} - \mu\begin{bmatrix}\frac{\partial^2 P_E}{\partial\mathbf{w}^T\partial\mathbf{w}} & \frac{\partial^2 P_E}{\partial\mathbf{b}^T\partial\mathbf{w}}\\ \frac{\partial^2 P_E}{\partial\mathbf{w}^T\partial\mathbf{b}} & \frac{\partial^2 P_E}{\partial\mathbf{b}^T\partial\mathbf{b}}\end{bmatrix}_j^{-1}\begin{bmatrix}\frac{\partial P_E}{\partial\mathbf{w}}\\ \frac{\partial P_E}{\partial\mathbf{b}}\end{bmatrix}_j \tag{21}
$$

It is well known that equalisation is a classification problem. The coefficient vector of the equaliser $[\mathbf{w}^T\,\mathbf{b}^T]^T$ is orthogonal to the hyperplane that is used to separate the two classes in hyperspace. Since it is the orientation of this vector that defines the decision boundary, the size of the vector has no effect on the BER. Thus it is not necessary to constrain the size of this vector apart from avoiding the degenerative case where its size is zero and hence no hyperplane can be defined. The size of the vector is thus a nuisance parameter in the optimisation procedure. One convenient way to remove it is to re-scale it after every iteration of the algorithm. From consideration of the expressions for the gradient and the Hessian it is obvious that rescaling by the size of the vector $\mathbf{w}$ simplifies many of the expressions. Thus:

$$\mathbf{w}'_{j+1} = \frac{\mathbf{w}_{j+1}}{\sqrt{\mathbf{w}_{j+1}^T \mathbf{w}_{j+1}}} \tag{22}$$

$$\mathbf{b}'_{j+1} = \frac{\mathbf{b}_{j+1}}{\sqrt{\mathbf{w}_{j+1}^T \mathbf{w}_{j+1}}} \tag{23}$$

The convergence performance of the gradient (G) and Gauss-Newton (GN) algorithms is illustrated in Figure 1 for a 3-tap channel at a signal/noise ratio of 15dB. Both algorithms are initialised with the coefficients of the Wiener minimum mean squared error (MMSE) DFE. It is clear that the Gauss-Newton learning strategy offers a better convergence performance than the gradient technique both when the channel impulse response is known a priori and when the channel is estimated from the data - in this case 64 samples.

IV. Adaptive algorithms

The key to developing adaptive algorithms is the p.d.f. $p_z(z_s)$ of the signed decision variable. Several techniques are available for estimating this scalar p.d.f., the simplest of which is box counting [10]. However it is well known that box counting requires large amounts of data for reliable estimates. Further it does not exploit the a priori knowledge of the structure of the p.d.f. that is available in this case. Kernel density estimation is known to produce far more reliable estimates with short data records and in particular is extremely natural when dealing with Gaussian mixtures.

Consider the case where K training samples of the transmitted symbols $\{s(k)\}$ and K associated received samples $\{r(k)\}$ are available (ignoring end effects). Given a set of weights $\mathbf{w}$ and $\mathbf{b}$, a kernel density estimate of the p.d.f. is:

$$\hat{p_z}(z_s) = \frac{1}{K\sqrt{2\pi}\rho_e\sqrt{\mathbf{w}^T\mathbf{w}}} \sum_{k=1}^{K} \exp\left(-\frac{(z_s - \mathrm{sgn}(s(k-d))z(k))^2}{2\rho_e^2\mathbf{w}^T\mathbf{w}}\right) \tag{24}$$

The usual problem associated with kernel density estimation is in selecting the radius parameter ρ_e. However in this case it is clearly related to the noise r.m.s. σ_e which is usually known roughly or can be estimated on line. In [10] a lower bound of:

$$\rho_e = \left(\frac{4}{3K}\right)^{\frac{1}{5}} \sigma_e \tag{25}$$

is suggested.

To illustrate the power of kernel density estimation in this application consider Figure 2 which shows the p.d.f. of the signed decision variable at the output of a DFE for a channel at *SNR* of 15 dB. The p.d.f. estimates are constructed using 100 received samples. It is clear that while the box counting method gives a good estimate of the underlying p.d.f. at the points at which it is evaluated, it provides little information about the tails of the distribution. In particular it is the p.d.f. in the range $z_s < 0$ which is needed in order to evaluate the BER c.f. equation (12). The kernel density estimate provides better estimates of the tails of the distribution than the box counting method. A second and compelling reason for using equation (24) is that it provides an estimate of the p.d.f. in the form of smooth differentiable function - a necessity for developing gradient and Gauss Newton optimisation techniques.

Starting with an estimate of the p.d.f. a block (or batch) adaptive filter algorithm follows in an analogous manner to the steepest decent or Gauss-Newton algorithms of section III. Two strategies are possible depending on whether a conventional or state translation DFE is employed. For the former, the summations over the states in (13), (14), (17), (18), (19) and (20) are replaced with summations over available data as in (24). In the latter an estimate of the channel impulse response is first formed from the data using a standard least squares method. Then, using this channel estimate as the actual channel impulse response, the state-translated output states $\{\mathbf{r'}_i = \mathbf{r}_i - \mathbf{F}_2\hat{\mathbf{s}}_{bi}\}$ are generated. The gradient algorithm reduces to (13) with the output vectors $\mathbf{r}_i$ replaced with the state-translated output vectors $\mathbf{r'}_i$. The Hessian reduces to equation (17) alone and the Gauss-Newton further reduces to:

$$\mathbf{w}_{j+i} \quad = \quad \mathbf{w}_j - \mu \left[\frac{\partial^2 P_E}{\partial \mathbf{w}^T \partial \mathbf{w}} \right]_j^{-1} \left[\frac{\partial P_E}{\partial \mathbf{w}} \right]_j \tag{26}$$

The same concepts can also be employed to develop an LMS-style update algorithm to train a MBER DFE. The key to developing the LMS algorithm from its related steepest decent algorithm is to replace the ensemble average of the gradient with a single point estimate of the gradient. In a similar manner, at sample k a point estimate of the p.d.f.

is simply:

$$\hat{p_z}(z_s(k)) = \frac{1}{\sqrt{2\pi}\rho_e\sqrt{\mathbf{w}^T\mathbf{w}}}\exp\left(-\frac{(z_s - \mathrm{sgn}(s(k-d))z(k))^2}{2\rho_e^2\mathbf{w}^T\mathbf{w}}\right) \tag{27}$$

An LMS style update follows from section III by replacing the exact p.d.f. with this crude estimate - assuming that the weights are rescaled after each update such that $\mathbf{w}^T(k)\mathbf{w}(k) = 1$ we have:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) \\ &\quad + \frac{\mu}{\sqrt{2\pi}\rho_e}(\mathbf{r}(k) - \mathbf{w}(k)z(k))\exp\left(-\frac{z^2(k)}{2\rho_e^2}\right)\mathrm{sgn}(s(k-d)) \end{aligned} \tag{28}$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) - \frac{\mu}{\sqrt{2\pi}\rho_e}\exp\left(-\frac{z^2(k)}{2\rho_e^2}\right)\mathrm{sgn}(s(k-d))\hat{\mathbf{s}}_b(k) \tag{29}$$

For a state-translation DFE, equation (28) is modified by replacing $\mathbf{r}(k)$ with $\mathbf{r}'(k)$.

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) \\ &\quad + \frac{\mu}{\sqrt{2\pi}\rho_e}(\mathbf{r}'(k) - \mathbf{w}(k)z(k))\exp\left(-\frac{z^2(k)}{2\rho_e^2}\right)\mathrm{sgn}(s(k-d)) \end{aligned} \tag{30}$$

where:

$$\mathbf{r}'(k) = \mathbf{r}(k) - \mathbf{F}_2(k)\hat{\mathbf{s}}_b(k) \tag{31}$$

Equation (29) is not used, rather an LMS algorithm is used to form an estimate of the impulse response vector $\mathbf{a} = [a_0\ a_1\ ...\ a_{n_a-1}]^T$:

$$\mathbf{a}(k+1) = \mathbf{a}(k) + 2\mu\begin{bmatrix} s(k-d) \\ \hat{\mathbf{s}}_b(k) \end{bmatrix}\left(r(k-d) - \mathbf{a}^T(k)\begin{bmatrix} s(k-d) \\ \hat{\mathbf{s}}_b(k) \end{bmatrix}\right) \tag{32}$$

This estimate of the impulse response provides the elements of the matrix $\mathbf{F}_2$. Note that equation (31) can be implemented efficiently using the state-translation structure considered in reference [8]. For the sake of discussion the above techniques based on point estimates of the p.d.f. will be called “least bit error rate” (LBER) algorithms.

An alternative approach is the approximate minimum-BER (AMBER) algorithm of [5]. It's DFE form for 2-PAM can be expressed as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu I(k)\mathrm{sgn}(e(k))\mathbf{r}(k) \tag{33}$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) - 2\mu I(k)\mathrm{sgn}(e(k))\mathbf{s}_b\hat{(}k) \tag{34}$$

where the error is:

$$e(k) = s(k-d) - \mathbf{w}^T(k)\mathbf{r}(k) + \mathbf{b}^T(k)\hat{\mathbf{s}}_b(k) \tag{35}$$

and the indicator function is:

$$I(k) = \frac{1}{2}(1 - \mathrm{sgn}(s(k-d)z(k) - \tau)) \tag{36}$$

The parameter τ is the nonnegative threshold which permits the weights to be updated in the region of the decision boundary.

Although the LBER and AMBER have been developed using different philosophies it is useful to compare them with respect to the parameter update mechanism. In the simplest form of AMBER [5], the threshold parameter τ is zero and the algorithm only updates when a decision error is observed. When the algorithm is initialised the equaliser is unlikely to separate the noise free states and hence the indicator function will be on most of the time in which case AMBER is equivalent to the signed error LMS algorithm c.f. [6]. When the algorithm has converged to a point where it separates the noise free states correctly its BER performance will be similar to the Wiener MMSE DFE. However in this region errors will be predominately caused by thermal noise. Thus the probability of the algorithm updating may be low and hence further convergence may be slow. Introducing the variable τ essentially defines a region around the decision boundary where the algorithm will continue to update even when errors do not occur. This region is defined by:

$$|\mathbf{w}^T\mathbf{r} - \mathbf{b}^T\hat{\mathbf{s}}_b| < \tau \tag{37}$$

In the LBER algorithm the effect of the distance from the decision boundary is controlled by the exponential term $\exp(-\mathrm{z}^2/(2\rho_\mathrm{e}^2))$. This can be viewed as a soft distance measure. The size of an update is a continuous and decreasing function of the distance from the boundary. The distance is scaled by the kernel radius ρ_e which in turn is a function of the noise r.m.s. σ_e.

V. Results

The convergence performance of a conventional DFE (block adaptive) and the state-translation DFE using a block of 128 training samples is illustrated in Figure 3 . Both gradient and Gauss-Newton forms are included. The algorithms are initialised using a least squares estimate of the Wiener MMSE DFE. For reference the performance with exact knowledge of the channel is included. These results are typical of observations over a range of channels and signal/noise conditions and can be summarised as follows:

(i) The gradient version of the state-translation DFE is the most robust and converges to the minimum BER solution is all case observed.

(ii) The gradient version of the conventional DFE does not converge as fast as the equivalent state-translation DFE and is not as robust in that divergence is occasionally observed.

(iii) The Gauss-Newton version of the state-translation DFE is the fastest converging of all. However instances of divergence are occasionally observed.

(iv) The Gauss-Newton version of the conventional DFE generally diverges. When it does converge it is faster than the gradient form.

The conclusions can be summarised as follows: if a gradient based block adaptive filter is required then the state-translation form provides an attractive solution whereas if the convergence of Gauss-Newton algorithm is required it is better to estimate the channel first and proceed as in section III.

The convergence performance of 4 LMS-style adaptive DFE's is compared in Figures 4 and 5. These are: the LBER conventional DFE of equations (28) and (29); a state translation LBER DFE of equations (30) (31), (32); the AMBER DFE of equations (33), (34), (35) and (36) with $\tau = 0.1$; an LMS conventional DFE.

In Figure 4 the channel impulse response is $\{0.5,\ 1.0\}$ with a SNR of 16 dB. The BER's are evaluated using equations (11) and (12) and averaged over an ensemble of 20 runs. The step size μ for all algorithms was set at 1/18 to ensure fairly fast convergence for the bench mark LMS algorithm. Figure 4 demonstrates the promise of the LBER approach in that both the the LBER-conventional-DFE and the LBER-state-translation-DFE outperforms the LMS and AMBER algorithms. However this result is atypical of the LBER conventional DFE. In many cases the LBER-conventional-DFE is no better than AMBER and since AMBER is extremely simple computationally it would be the preferred choice. A more typical example is illustrated in Figure 5 where the channel impulse response is $\{0.25,\ 0.5,\ 1.0\}$ with a SNR of 17 dB. Here the step size for all algorithms was set at 1/30. Under these conditions there is little to distinguish the LBER-conventional-DFE from the AMBER DFE and the LMS DFE. It should be noted however that in both Figures the LBER-state-translation-DFE outperforms all the other algorithms.

VI. Conclusions

The problem of constructing adaptive MBER DFE's for binary signalling has been considered. Both gradient and Gauss-Newton algorithms have been formulated in such a manner as to facilitate application to conventional and state-translation architectures. The Hessian matrix for the Gauss-Newton algorithm has not appeared before. Kernel density estimation has been demonstrated to provide a convenient mechanism for approximating the BER as a smooth function of the available data. Block and serial adaptive algorithms in both conventional and state-translation form have been developed from this premise. A computer simulation study of the various algorithms leads to following conclusions: for block processing a combination of channel estimation and GN BER minimisation provides a robust and fast converging solution; for LMS-style training the LBER state translation DFE outperforms existing algorithms.

References

[1] S. Chen, B. Mulgrew, E. S. Chng and G. J. Gibson "Space translation properties and the minimum-BER linear-combiner DFE," *IEE Proc. Communications*, vol. 145, no. 5, pp. 316-322, 1998.

[2] E. Sharmash and K. Yao, "On the structure and performance of a linear decision feedback equalizer based on the minimum error probability criterion," *IEEE International Conference on Communications, ICC 74*, 1974.

[3] P. Galko and S. Pasupathy, "Optimal linear receiver filters for binary digital signals," *IEEE International Conference on Communications, ICC 82*, 1982.

[4] C. C. Yeh and J. R. Barry, "Approximate minimum bit error rate equalization for pulse-amplitude and quadrature-amplitude modulation," *IEEE International Conference on Communications, ICC 98*, vol. 1, pp. 16-20, 1998.

[5] C. C. Yeh and J. R. Barry, "Approximate minimum-bit error rate equalization for binary signaling," *IEEE International Conference on Communications, ICC 97*, pp. 1095-1099, 1997.

[6] R. Sharma, W. A. Sethares and J. A. Bucklew, "Asymptotic analysis of stochastic gradient-based adaptive filtering algorithms with general cost functions", *IEEE Trans. Signal Processing*, vol. 44, no. 9, pp. 2186-2194, Sept 1996.

[7] C. P. Kwong, "Dual sign algorithm for adaptive filtering," *IEEE Trans. Communications*, pp. 1272-1275, 1986.

[8] S. Chen, E. S. Chng, B. Mulgrew, G. J. Gibson, "Minimum BER Linear Combiner DFE," *IEEE International Conference on Communications, ICC-96*, pp. 1173-1177, 1996.

[9] A. P. Clark, L. H. Lee and R. S. Marshall, "Developments of the conventional non-linear equaliser," *IEE Proceedings Part F*, vol. 129, no. 2, pp. 85-94, 1982.

[10] B. W. Silverman, *Density Estimation*, London, Chapman Hall, 1996.

[11] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis*, Oxford University Press, 1997.

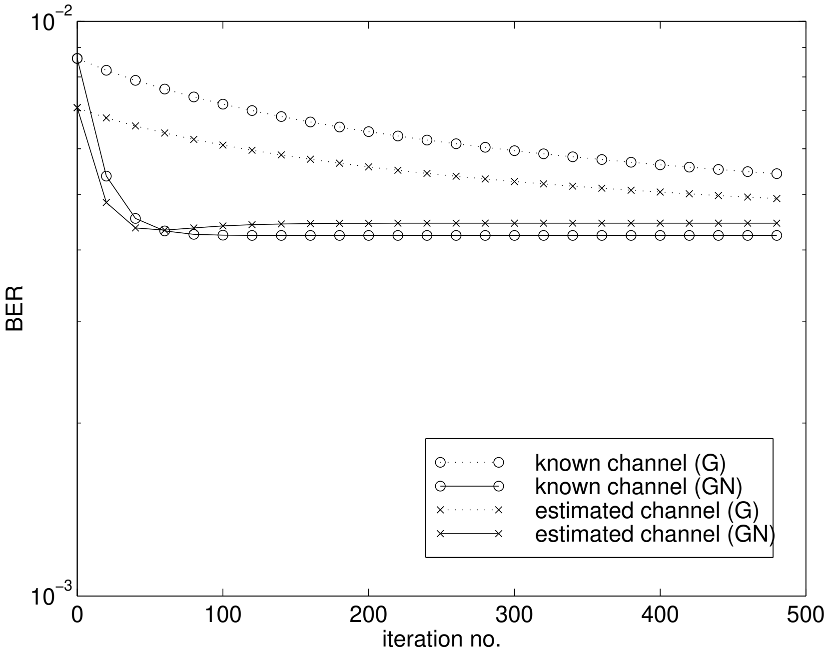

Fig. 1. Convergence comparison of gradient (G) and Gauss-Newton (GN) algorithms for known channel and estimated channel: channel impulse response $\{0.25,\ 0.5,\ 1.0\}$; $SNR = 15$dB

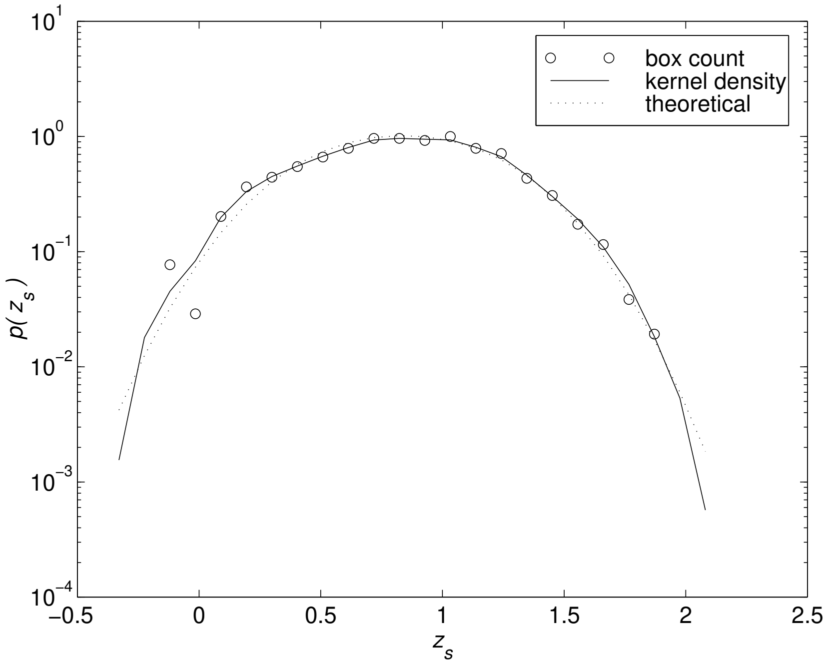

Fig. 2. Distribution of signed decision variable

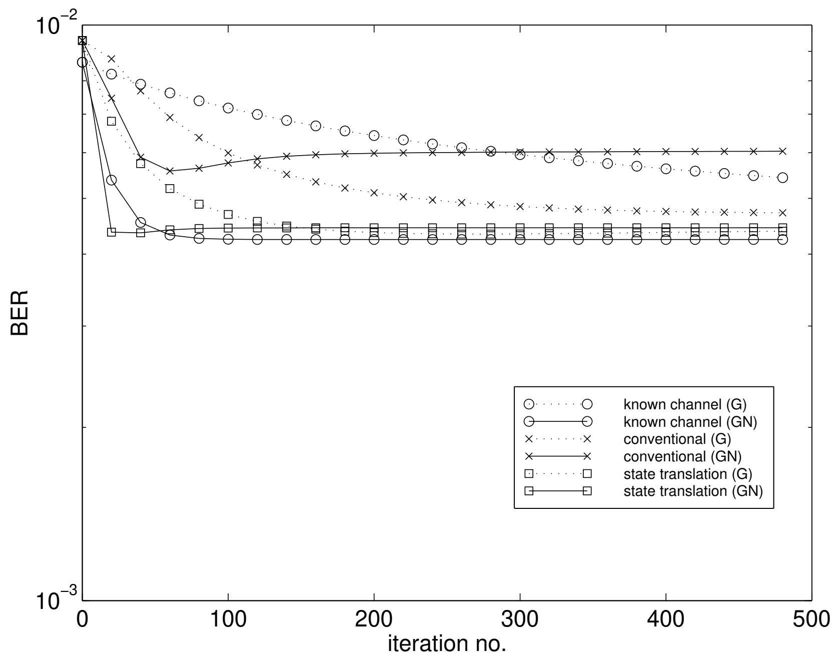

Fig. 3. Convergence comparison of gradient (G) and Gauss-Newton (GN) block adaptive algorithms: channel impulse response $\{0.25,\ 0.5,\ 1.0\}$; $SNR = 15$dB

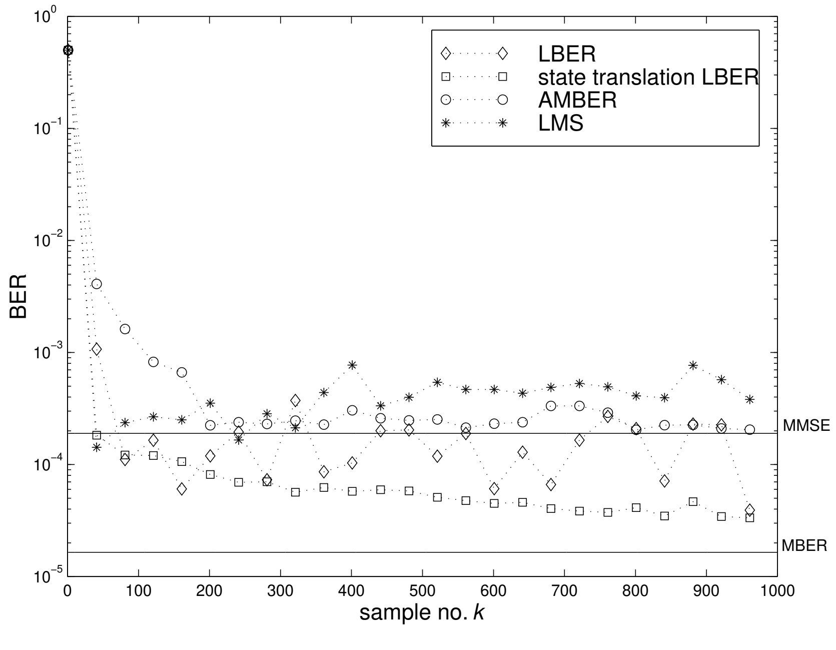

Fig. 4. Convergence comparison of LMS-DFE, AMBER-DFE, LBER conventional DFE and LBER state translation DFE: channel impulse response $\{0.5,\ 1.0\}$; $SNR = 16$dB

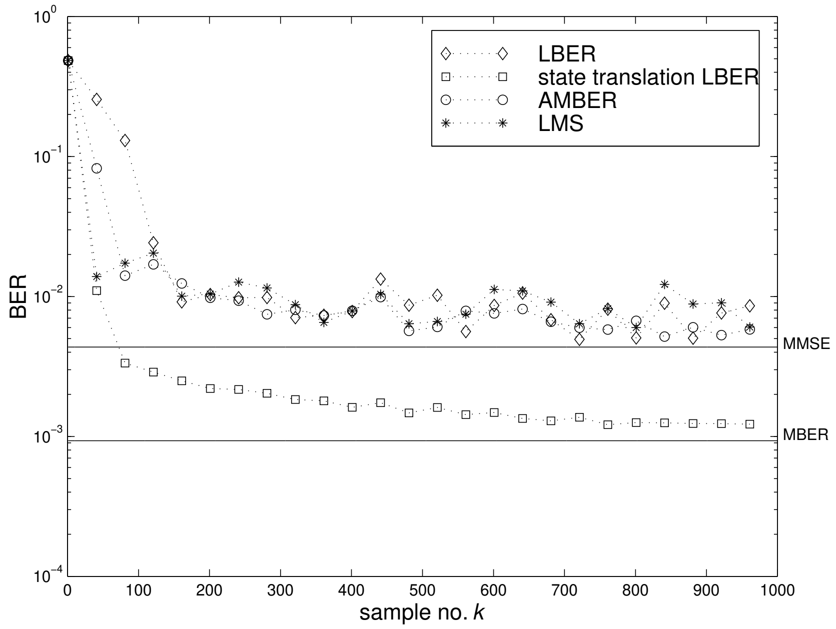

Fig. 5. Convergence comparison of LMS-DFE, AMBER-DFE, LBER conventional DFE and LBER state translation DFE: channel impulse response $\{0.25,\ 0.5,\ 1.0\}$; $SNR = 17$dB