Adaptive Channel Equalisation Using Minimum BER Gradient-Newton Algorithms

Rodrigo C. de Lamare and Raimundo Sampaio-Neto
CETUC - PUC-RIO, 22453-900, Rio de Janeiro - Brazil
e-mails: delamare@infolink.com.br, raimundo@cetuc.puc-rio.br

Abstract—

In this paper we investigate the use of adaptive minimum bit error rate (MBER) Gradient-Newton algorithms for channel equalisation applications. The proposed algorithms approximate the bit error rate (BER) from training data using linear transversal and decision feedback (DFE) equaliser structures. A comparative analysis of linear and DFE equalisers, employing minimum mean squared error (MMSE), previously reported MBER and the proposed MBER algorithms is carried out. Computer simulation experiments show that the MBER Gradient-Newton approaches outperform other analysed algorithms and can operate with shorter training sequences.

I. Introduction

Channel equalisers employing the minimum mean square error (MMSE) [1],[2] criterion have become rather successful, since they usually show good performance and have simple adaptive implementation [2],[3]. However, it is well known that the MSE cost function is not optimal in digital communications applications, and the most appropriate cost function is the bit error rate (BER) [4],[5]. The approximate minimum bit error rate (AMBER) [4] and the least bit error rate (LBER) [5] are two of the most successful and suitable algorithms for adaptive implementation. However, these minimum bit error rate (MBER) algorithms usually require long training sequences to converge to lower bit error rates than those achieved by the techniques that employ the MSE cost function. In this work, we investigate MBER Gradient-Newton algorithms that can speed up the convergence of the equaliser, requiring shorter training data. The proposed algorithms, denoted Gradient-Newton-AMBER and Gradient-Newton-LBER, are similar to the well known LMS-Newton algorithm and employ the error functions used in the AMBER and the LBER, respectively. We perform a comparative analysis of linear and decision feedback (DFE) equalisers, employing the LMS [3], the AMBER [4], the LBER [5] the LMS-Newton [3], the Gradient-Newton-AMBER and the Gradient-Newton-LBER adaptive algorithms. Computer simulation experiments show that the MBER Gradient-Newton approaches outperform other analysed algorithms and can operate with shorter training sequences, even though they require higher computational complexity.

This paper is organised as follows. Section II briefly describes the communication system model. The adaptive equalisers structures and stochastic gradient algorithms are presented in Sections III and IV. Section V is dedicated to the Gradient-Newton based algorithms. Section VI presents and discusses the simulation results and Section VII gives the concluding remarks of this work.

II. System Model

We assume a BPSK communication system that transmits modulated symbols through a radio-type communication channel, which is followed by an adaptive equaliser and a symbol detector, as shown in Fig. 1. Consider a discrete time communication channel, where $x(k)$ is the binary transmitted symbol, $h(k)$ is the channel impulse response with memory M and $n(k)$ is additive white gaussian noise (AWGN) with power spectrum density σ^2. The output signal $r(k)$ of the channel is expressed by:

$$r(k) = s(k) + n(k) = \sum_{i=0}^{M} h(i)x(k-i) + n(k) \qquad (1)$$

where $s(k)$ is the channel output without noise.

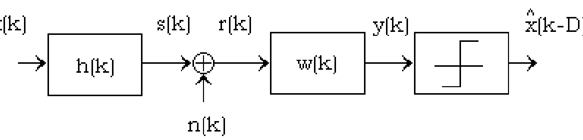

Fig. 1. Communication channel and receiver.

The channel output vector $\mathbf{r}(k) = \big[r(k) \dots r(k-N)\big]^T$ is given by:

$$\mathbf{r}(k) = \mathbf{s}(k) + \mathbf{n}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{n}(k) \qquad (2)$$

where $\mathbf{x}(k) = \big[x(k) \dots x(k-M-N)\big]^T$ is the vector with the channel inputs, $\mathbf{n}(k) = \big[n(k) \dots n(k-N)\big]^T$ is the noise sample vector, $\mathbf{s}(k) = \big[s(k) \dots s(k-N)\big]^T$ is the vector without noise and $\mathbf{H}$ is a $(N+1) \times (M+N+1)$ Toeplitz convolution matrix expressed by:

$$\mathbf{H} = \begin{pmatrix} h(0) & \dots & h(M) & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & h(0) & \dots & h(M) \end{pmatrix} \qquad (3)$$

III. Equalisers structures

The adaptive channel equalisation problem involves the application of a receiving filter, that adjusts its coefficients in order to minimise a given objective function [1],[2]. The equaliser must be adaptive in order to track the signal variations imposed by the channel, however, it requires a desired signal taken from a training sequence to adjust its parameters, as shown in Fig. 2.

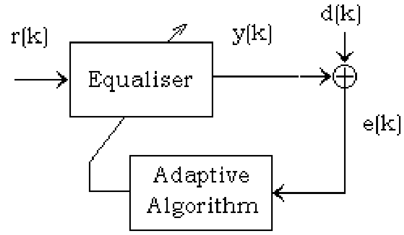

Fig. 2. Block diagram of an adaptive equaliser.

The linear transversal equaliser consists of a linear filter with $N+1$ taps described by the vector $\mathbf{w} = \big[w_0 \; ... \; w_N\big]^T$. The linear equaliser output is given by:

$$y(k) = \mathbf{w}^T \mathbf{r}(k) \tag{4}$$

where $\mathbf{r}(k) = \big[r(k) \; \ldots \; r(k-N)\big]$ is the observed output signal vector of the channel. The decision $\hat{x}(k-D)$ on the transmitted symbol $x(k-D)$ is determined by the equaliser output signal, $\hat{x}(k-D) = sgn(y(k))$, where D corresponds to the delay imposed by the channel and the equaliser.

The decision feedback (DFE) equaliser employs the received samples and past decisions to estimate the received symbols. The output of the DFE equaliser is described by:

$$y(k) = \mathbf{w_f}^T \mathbf{r}(k) - \mathbf{b}^T \hat{\mathbf{x}}(k) \tag{5}$$

where $\mathbf{r}(k) = [r(k) \; \ldots \; r(k-N_1)]$ is the observed output signal vector of the channel, the past detected symbol vector is $\hat{\mathbf{x}}(k) = [\hat{x}(k-D-1) \; \ldots \; \hat{x}(k-D-N_2)]^T$, $\mathbf{w_f} = [w_0 \; ... \; w_{N_1}]^T$ is the feedforward coefficient vector and $\mathbf{b} = [b_1 \; ... \; b_{N_2}]^T$ is the feedback coefficient vector. Alternatively the DFE can be expressed by:

$$y(k) = \mathbf{w}^T \mathbf{u}(k) \tag{6}$$

where $\mathbf{u}(k) = \big[r(k) \ldots r(k-N_1-1)\hat{x}(k-D-1)\ldots\hat{x}(k-D-N_2)\big]$ is the observation vector for the DFE structure, $\mathbf{w} = \big[\mathbf{w}_f^T \; -\mathbf{b}^T\big]^T = \big[w_0 \; ... \; w_{N_1+N_2+1}\big]^T$ is the coefficient vector with N_1+1 and N_2 taps in the feedforward and feedback sections, respectively.

IV. Stochastic gradient algorithms

In this section, we describe stochastic gradient algorithms that adjust the parameters of the receivers based on the minimisation of the mean square error (MSE) and the bit error rate (BER) cost functions.

A. The LMS algorithm

The adaptive equalisation solution for the linear equaliser via the LMS algorithm [1] is based on the MMSE error criterion formed by the error signal $e(k) = d(k) - y(k)$, and is described by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{r}(k) \tag{7}$$

In the case of the decision feedback (DFE) equaliser, the solution is given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{u}(k) \tag{8}$$

where $d(k) = x(k-D)$ is the desired signal taken from the training sequence, $\mathbf{r}(k)$ is the observation vector for the linear equaliser, $\mathbf{u}(k)$ is the observation vector for the DFE structure and μ is the algorithm step size.

B. The AMBER algorithm

Given a transmitted training sequence $\mathbf{d}$, the bit error probability $P(\epsilon|\mathbf{d})$, for the linear and the DFE receivers, is expressed by:

$$P\big(\epsilon|\mathbf{d}\big) = P\big(d(k)sgn(y(k)) = -1\big)$$

$$P\big(\epsilon|\mathbf{d}\big) = P\big(sgn(d(k)y(k)) = -1\big) = P\big(y_s(k) < 0\big) \tag{9}$$

where $y(k)$ is given by (4), for the linear equaliser, and expressed by (6), in the case of the DFE equaliser and $d(k)$ is the desired symbol taken from the training sequence.

The equaliser solution that minimises the BER criterion via the AMBER algorithm [4] for linear structures employs the vector function $g(\mathbf{w}(k))$ [3] to approximate an expression for a coefficient vector $\mathbf{w}(k)$ that achieves a MBER performance with linear receiver structures, as described by:

$$g(\mathbf{w}(k)) = E\left[Q\left(\frac{d(k)\mathbf{w}^T(k)\mathbf{s}(k)}{\parallel \mathbf{w}(k) \parallel \sigma}\right)d(k)\mathbf{s}(k)\right] \tag{10}$$

where $d(k)$ is the desired transmitted symbol taken from the training sequence and $Q(.)$ is the Gaussian error function. A simple stochastic solution for $\mathbf{w}$ can be derived by using $g(\mathbf{w}(k))$ and adjusting the receiver weights by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu g(\mathbf{w}(k)) \tag{11}$$

Note that for linear receiver structures the quantity $Q\Big(\frac{d(k)\mathbf{w}^T(k)\mathbf{s}(k)}{\|\mathbf{w}(k)\|\sigma}\Big)$ inside the expected value operator in (10) corresponds to the conditional bit error probability given the product $d(k)\mathbf{s}(k)$. This quantity can be replaced in (10) by an error indicator function $i_d(k)$ given by:

$$i_d(k) = \frac{1}{2}(1 - sgn(d(k)y(k))) \tag{12}$$

where $y(k)$ is the estimated symbol and $d(k)$ is the desired signal provided by the training sequence.

Following this approach, the AMBER algorithm, as devised for linear equalisers [4], is described by the following equalities:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu E\left[Q\left(\frac{d(k)\mathbf{w}^T(k)\mathbf{s}(k)}{\parallel \mathbf{w}(k) \parallel \sigma}\right)d(k)\mathbf{s}(k)\right]$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu E\Big[E\big[i_d(k) \mid d(k)\mathbf{s}(k)\big]d(k)\mathbf{s}(k)\Big]$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu E\big[i_d(k)d(k)\mathbf{s}(k)\big]$$

Since $\mathbf{s}(k) = \mathbf{r}(k) - \mathbf{n}(k)$, and $i_d(k)$ and $d(k)$ are statistically independent, we have $E[i_d(k)d(k)\mathbf{n}(k)] = E[d(k)]E[i_d(k)\mathbf{n}(k)] = 0$ and thus:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu E\Big[i_d(k)d(k)\mathbf{r}(k)\Big] \tag{13}$$

The AMBER stochastic gradient update equation for the linear equaliser is given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu i_d(k)d(k)\mathbf{r}(k) \tag{14}$$

And the AMBER approach applied to the DFE equaliser is expressed by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu i_d(k)d(k)\mathbf{u}(k) \tag{15}$$

Note that the expressions in (7) and (8) equal (14) and (15), respectively, if we replace $e(k)$ by $i_d(k)d(k)$. In practice, a modified error indicator function $i_d(k) = \frac{1}{2}(1 - sgn(d(k)y(k) - \tau))$ is employed, where the threshold τ is responsible for increasing the algorithm rate of convergence. This algorithm updates when an error is made and also when an error is almost made, becoming a smarter choice for updating the filter coefficients.

C. The LBER algorithm

The equaliser BER depends on the distribution of the decision variable $y(k)$, which is a function of the weights of the equaliser. The sign-adjusted decision variable for the DFE equaliser $y_s(k) = sgn(x(k-D))y(k)$ is drawn from a Gaussian mixture, described by:

$$y_s(k) = sgn(x(k-D))\left(\mathbf{w}^T\mathbf{H}\mathbf{x}(k) - \mathbf{b}^T\hat{\mathbf{x}}(k) + \mathbf{w}^T\mathbf{n}(k)\right)$$

$$y_s(k) = sgn(x(k-D))y'(k) + n'(k) \tag{16}$$

where the first term of (16) is the noise free sign-adjusted equaliser output.

Consider that K samples of the transmitted symbols $x(k)$ and K samples of the received symbols $r(k)$ are available from the samples $d(k) = x(k-D)$ of a training sequence. A kernel density estimate [5] of the p.d.f. of y_s is given by:

$$p_{y_s}(y_s) = \frac{1}{K\sqrt{2\pi}\rho(\mathbf{w_f^T w_f})^{1/2}}$$

$$\sum_{k=1}^{K} exp\left(\frac{-(y_s - sgn(d(k))y(k))^2}{2\rho^2\mathbf{w_f^T w_f}}\right) \tag{17}$$

where ρ is the radius parameter of the kernel density estimate [5].

Substituting the expected value of the gradient with a single point estimate, we have:

$$\hat{p}_{y_s}(y_s(k)) = \frac{1}{K\sqrt{2\pi}\rho(\mathbf{w_f^T w_f})^{1/2}}$$

$$exp\left(\frac{-(y_s - sgn(d(k))y(k))^2}{2\rho^2\mathbf{w_f^T w_f}}\right) \tag{18}$$

The probability of error is estimated by:

$$P_\epsilon = P(y_s < 0) = \int_{-\infty}^{0} \hat{p}_{y_s}(y_s)dy_s = Q\left(\frac{sgn(d(k))y(k)}{\rho(\mathbf{w_f^T w_f})^{1/2}}\right) \tag{19}$$

The gradient terms of P_ϵ are:

$$\frac{\partial P_\epsilon}{\partial \mathbf{w_f}} = \frac{exp\left(\frac{-y(k)^2}{2\rho^2\mathbf{w_f^T w_f}}\right)sgn(d(k))}{\sqrt{2\pi}\rho}\left(\frac{-\mathbf{r}(k)}{(\mathbf{w_f^T w_f})^{1/2}} + \frac{\mathbf{w_f}y(k)}{(\mathbf{w_f^T w_f})^{3/2}}\right) \tag{20}$$

and

$$\frac{\partial P_\epsilon}{\partial \mathbf{b}} = \frac{1}{\sqrt{2\pi}\rho(\mathbf{w_f^T w_f})^{1/2}} exp\left(\frac{-y(k)^2}{2\rho^2\mathbf{w_f^T w_f}}\right)sgn(d(k))\hat{\mathbf{x}}(k) \tag{21}$$

An algorithm similar to the LMS was devised in [5] by substituting the exact pdf by its instantaneous estimate and adjusting the receiver weights such that $\mathbf{w_f}^T(k)\mathbf{w_f}(k) = 1$:

$$\mathbf{w_f}(k+1) = \mathbf{w_f}(k) - \mu\left[\frac{\partial P_\epsilon}{\partial \mathbf{w}}\right]_k \tag{22}$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) - \mu\left[\frac{\partial P_\epsilon}{\partial \mathbf{b}}\right]_k \tag{23}$$

The LBER algorithm for the linear equaliser ($\mathbf{w}_f = \mathbf{w}$ *and* $\mathbf{b} = \mathbf{0}$) is given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\frac{1}{\sqrt{2\pi}\rho}exp\left(\frac{-(y(k))^2}{2\rho^2}\right)sgn(d(k))$$

$$\times\left(\mathbf{I} - \mathbf{w}(k)\mathbf{w}^T(k)\right)\mathbf{r}(k) \tag{24}$$

The LBER algorithm for the DFE equaliser is expressed by:

$$\mathbf{w_f}(k+1) = \mathbf{w_f}(k) + \mu\frac{1}{\sqrt{2\pi}\rho}exp\left(\frac{-(y(k))^2}{2\rho^2}\right)sgn(d(k))$$

$$\times\left(\mathbf{r}(k) - \mathbf{w}(k)y(k)\right) \tag{25}$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) - \mu\frac{1}{\sqrt{2\pi}\rho}exp\left(\frac{-(y(k))^2}{2\rho^2}\right)sgn(d(k))\hat{\mathbf{x}}(k) \tag{26}$$

Rearranging the expressions for the DFE receiver, we can rewrite them in a single vector $\mathbf{w}(k)$ format as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\frac{1}{\sqrt{2\pi}\rho}exp\left(\frac{-(y(k))^2}{2\rho^2}\right)sgn(d(k))$$

$$\times\left(\mathbf{I} - [\mathbf{w_f}^T(k)\ \mathbf{0}^T]^T\mathbf{w}^T(k)\right)\mathbf{u}(k) \tag{27}$$

where $\mathbf{w_f}(k) = [w_0\ \ldots\ w_{N_1}]$ is the feedforward coefficient vector, $d(k) = x(k-D)$ is the desired signal taken from the training sequence, μ is the algorithm step size and ρ the radius parameter which is related to the noise standard deviation σ. Whilst in the AMBER, a non-zero τ defines a region boundary where the algorithm will continue to update, in the LBER, the effect of the distance from the decision boundary is controlled by an exponential term [5].

V. Gradient-Newton Based Algorithms

Gradient-Newton algorithms [3] incorporate second-order statistics of input signals, increasing their convergence rate. They usually have a faster convergence rate than gradient techniques, although they require a higher computational complexity. The update equation of Newton´s method is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{1}{2}\mathbf{R_u}^{-1} g_{\mathbf{w}}(k) \tag{28}$$

where $\mathbf{R_u}$ is the autocorrelation matrix of the observation vector $\mathbf{u}$ and $g_{\mathbf{w}}(k)$ is the gradient vector.

In practice, only estimates of the autocorrelation matrix $\mathbf{R_u}$ and the gradient vector $g_{\mathbf{w}}(k)$ are available. These estimates can be applied to Newton´s formula to devise an update rule give by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu\hat{\mathbf{R_u}}^{-1}(k)\hat{g}_{\mathbf{w}}(k) \tag{29}$$

The convergence factor μ is introduced to protect the algorithm from divergence, which is originated by the use of noisy estimates of $\mathbf{R_u}$ and $g_{\mathbf{w}}(k)$.

To obtain an unbiased estimate of the observation matrix $\mathbf{R_u}$, we employ the following weighted sum:

$$\hat{\mathbf{R_u}}(k) = \alpha\mathbf{u}(k)\mathbf{u}^T(k) + (1-\alpha)\hat{\mathbf{R_u}}(k-1) \tag{30}$$

where α is a small factor chosen in the range $0 < \alpha \leq 0.1$ and $\mathbf{u}(k)$ is the observation vector.

To avoid the required inversion of $\hat{\mathbf{R_u}}(n)$, we use the matrix inversion lemma, described by:

$$[\mathbf{A}+\mathbf{BCD}]^{-\mathbf{1}} = \mathbf{A}^{-\mathbf{1}} - \mathbf{A}^{-\mathbf{1}}\mathbf{B}[\mathbf{DA}^{-\mathbf{1}}\mathbf{B}+\mathbf{C}^{-\mathbf{1}}]^{-\mathbf{1}}\mathbf{DA}^{-\mathbf{1}} \tag{31}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are matrices with appropriate dimensions and $\mathbf{A}$ and $\mathbf{C}$ are non-singular. Choosing $A = (1-\alpha)\hat{\mathbf{R_u}}(k-1)$, $\mathbf{B} = \mathbf{D}^T = \mathbf{r}(k)$ and $\mathbf{C} = \alpha$, it can be shown that:

$$\hat{\mathbf{R_u}}^{-1}(k) = \frac{1}{1-\alpha}\Bigg[\hat{\mathbf{R_u}}^{-1}(k-1) - \frac{\hat{\mathbf{R_u}}^{-1}(k-1)\mathbf{u}(k)\mathbf{u}^T(k)\hat{\mathbf{R_u}}^{-1}(k-1)}{\frac{1-\alpha}{\alpha} + \mathbf{u}^T(k)\hat{\mathbf{R_u}}^{-1}\mathbf{u}(k)}\Bigg] \tag{32}$$

The resulting equation for the computation of $\hat{\mathbf{R_u}}^{-1}(k)$ is less complex to update ($O(N^2)$) than its direct inversion ($O(N^3)$).

A. LMS-Newton algorithm

The LMS-Newton [3] algorithm employs the error signal $e(k) = d(k) - y(k)$, which corresponds to the MMSE solution. Thus, the estimate of the gradient $\hat{g}_{\mathbf{w}}(k)$ is replaced by $e(k)\mathbf{r}(k)$ to yield the expression of the LMS-Newton algorithm for the linear equaliser as given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\hat{\mathbf{R_r}}^{-1}(k)e(k)\mathbf{r}(k) \tag{33}$$

In the case of the decision feedback (DFE) equaliser, the solution is expressed by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\hat{\mathbf{R_u}}^{-1}(k)e(k)\mathbf{u}(k) \tag{34}$$

where $d(k) = x(k-D)$ is the desired signal taken from the training sequence, $\mathbf{r}(k)$ is the observation vector for the linear equaliser, $\mathbf{u}(k)$ is the observation vector for the DFE structure and μ is the algorithm step size. Note that the LMS-Newton algorithms differs from the LMS algorithm by the use of the observation matrix $\hat{\mathbf{R_u}}^{-1}(k)$ to increase its rate of convergence.

B. Gradient-Newton-AMBER algorithm

An approach similar to LMS-Newton can be used to devise a Gradient-Newton based algorithm that minimises a given objective function $g(\mathbf{w}(k))$, as expressed by:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\hat{\mathbf{R_u}}^{-1}\hat{g}(\mathbf{w}(k)) \tag{35}$$

We chose the objective function $g(\mathbf{w}(k))$ used in the AMBER algorithm [4] as an approximation to an MBER function. Then, we use the observation matrix $\hat{\mathbf{R_u}}^{-1}(k)$ to speed up the convergence rate of the algorithm and obtain the Gradient-Newton AMBER update equation for the linear equaliser:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\hat{\mathbf{R_r}}^{-1}(k)i_d(k)d(k)\mathbf{r}(k) \tag{36}$$

The decision feedback (DFE) equaliser solution is expressed by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\hat{\mathbf{R_u}}^{-1}(k)i_d(k)d(k)\mathbf{u}(k) \tag{37}$$

where $d(k) = x(k-D)$ is the desired signal taken from the training sequence, $\mathbf{r}(k)$ is the observation vector for the linear equaliser, $\mathbf{u}(k)$ is the observation vector for the DFE structure and μ is the algorithm step size. Note that the Gradient-Newton algorithm only differs from the AMBER by the addition of the inverse correlation matrix $\hat{\mathbf{R_u}}^{-1}(k)$ in (14) and (15).

C. Gradient-Newton-LBER algorithm

An algorithm similar to the LMS-Newton can be devised employing an approach analogous to the LBER algorithm. Using Newton´s update rule we have:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu\mathbf{R}^{-1}(k)\left[\frac{\partial P_\epsilon}{\partial \mathbf{w}}\right]_k \tag{38}$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) - \mu\mathbf{R}^{-1}(k)\left[\frac{\partial P_\epsilon}{\partial \mathbf{b}}\right]_k \tag{39}$$

The Gradient-Newton-LBER algorithm for the linear receiver is given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\frac{1}{K\sqrt{2\pi}\rho}exp\left(\frac{-(y(k))^2}{2\rho^2}\right)sgn(d(k))$$

$$\times \hat{\mathbf{R}_{\mathbf{r}}}^{-1}(k)\left(\mathbf{I}-\mathbf{w}(k)\mathbf{w}^T(k)\right)\mathbf{r}(k) \qquad (40)$$

The decision feedback (DFE) equaliser solution in a single vector $\mathbf{w}(k)$ format is expressed by:

$$\mathbf{w}(k+1)=\mathbf{w}(k)+\mu\frac{1}{\sqrt{2\pi}\rho}exp\left(\frac{-(y(k))^2}{2\rho^2}\right)sgn(d(k))\hat{\mathbf{R}_{\mathbf{u}}}^{-1}(k)$$

$$\times\left(\mathbf{I}-[\mathbf{w_f}^T(k)\ \mathbf{0}^T]^T\mathbf{w}^T(k)\right)\mathbf{u}(k) \qquad (41)$$

where $\mathbf{w_f}(k)=[w_0\ \ldots\ w_{N_1}]$ is the feedforward coefficient vector, $d(k)=x(k-D)$ is the desired signal taken from the training sequence, $\mathbf{r}(k)$ is the observation vector for the linear equaliser, $\mathbf{u}(k)$ is the observation vector for the DFE structure and μ is the algorithm step size. Note that the Gradient-Newton algorithm only differs from the LBER for the DFE structure by the use of the inverse correlation matrix $\hat{\mathbf{R}_{\mathbf{u}}}^{-1}(k)$ in (27), which increases its rate of convergence.

VI. Simulations

In this section, we conduct simulation experiments to assess the convergence and the BER performance of the linear and DFE equalisers operating with the algorithms described and perform a comparative analysis of them. To evaluate the receivers, we have simulated their operation under different radio-type communication channels.

A. Convergence Analysis

The simulation experiments, conducted to assess the convergence of the different structures and algorithms, employed 1000 training data bits averaged over 100 independent experiments. Furthermore, we use a small fixed threshold $\tau=0.1$ for the AMBER based algorithms and $\rho=8\sigma^2$ for the LBER based methods. In the first situation, the linear equalisers have 8 taps, the stochastic gradient algorithms operate with $\mu=0.0075$ and the gradient-newton techniques operate with $\alpha=0.001$ and $\mu=0.0001$. We consider a linear channel with transfer function $H(z)=1-0.25z^{-1}+0.4z^{-2}$. The bit error rate (BER) was measured at each received symbol for the different receivers, as shown in Fig. 3.

According to the curves in Fig. 3, the linear transversal equaliser operating with the Gradient-Newton-LBER algorithm achieved the best convergence performance, outperforming the Gradient-Newton-AMBER, the LMS-Newton, the AMBER, the LBER and the LMS techniques. Indeed, the gradient-newton algorithms have shown faster convergence than their gradient counterparts.

In the second experiment, we have designed DFE equalisers with 6 taps in the feedforward section and 2 taps in the feedback one. The stochastic gradient algorithms were tuned to operate with $\mu=0.01$ and the gradient-newton techniques operate with $\alpha=0.001$ and $\mu=0.0001$. We consider a linear channel with transfer function $H(z)=1.1+1.2z^{-1}-0.2z^{-2}$. As occurs in the first experiment the BER is measured at each received symbol for the different receivers, as depicted in Fig. 4.

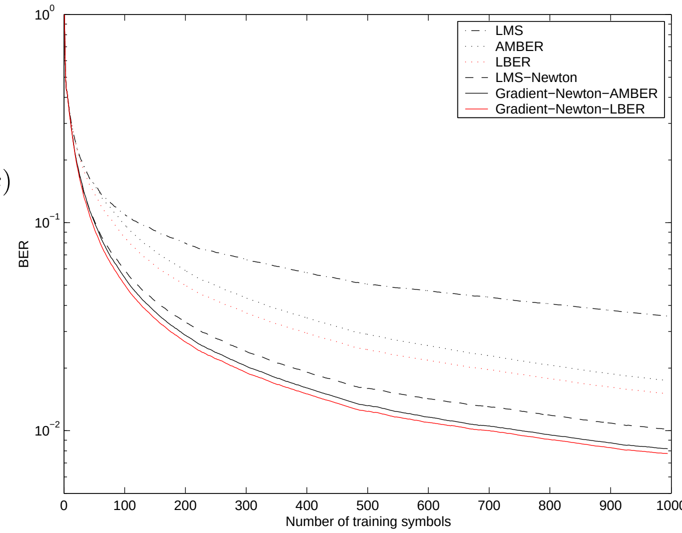

Fig. 3. Convergence performance for the linear equalisers for channel $H(z)=1-0.25z^{-1}+0.4z^{-2}$ with $\frac{E_b}{N_0}=16dB$.

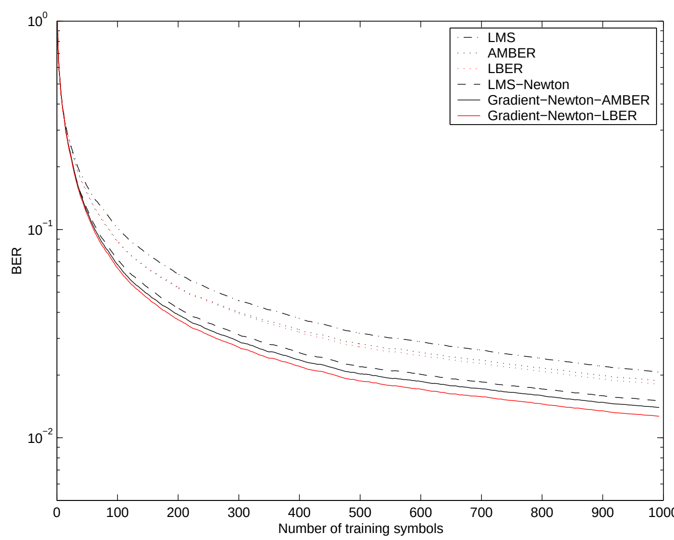

Fig. 4. Convergence performance for the DFE equalisers for channel $H(z)=1.1+1.2z^{-1}-0.2z^{-2}$ with $\frac{E_b}{N_0}=15dB$.

For DFE equalisers, the Gradient-Newton-LBER algorithm also achieved the best convergence performance, slightly outperforming the Gradient-Newton-AMBER technique, followed by the LMS-Newton, the LBER, the AMBER and the LMS algorithms.

B. BER Performance

The BER simulation results were obtained with 400 training symbols and 10^4 data bits averaged over 100 independent experiments. All equalisers operate with a step size μ during training and no adaptation occurs in data mode. We use a small fixed threshold $\tau=0.1$ for the AMBER type algorithms and $\rho=8\sigma^2$ for the LBER based methods.

In the first experiment, the equalisers have 8 taps and operate with $\mu = 0.0075$ and $\alpha = 0.001$ We consider a linear channel with transfer function $H(z) = 1 - 0.25z^{-1} + 0.4z^{-2}$. Fig. 5 shows the BER performance of the receivers.

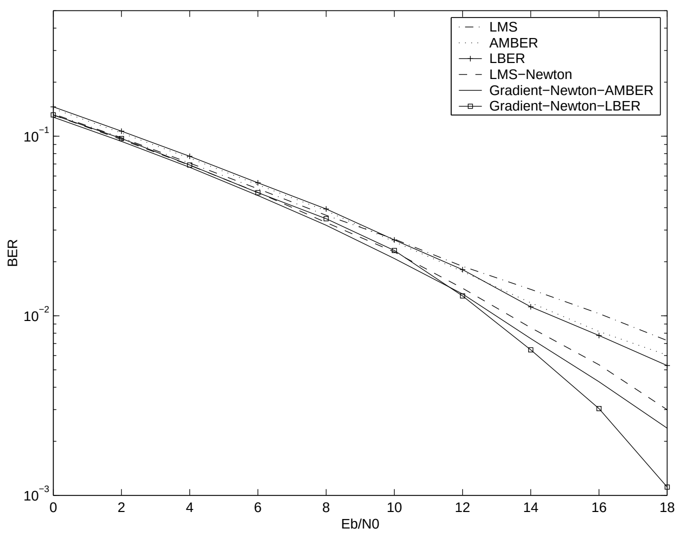

Fig. 5. BER performance using linear equalisers with channel $H(z) = 1 - 0.25z^{-1} + 0.4z^{-2}$.

According to Fig. 5 the linear equaliser operating with the Gradient-Newton-LBER algorithm has the best BER performance amongst the examined systems, followed by the Gradient-Newton-AMBER, the LMS-Newton, the AMBER, the LBER and the LMS algorithms.

In the second experiment, we have used DFE equalisers with 6 taps in the feedforward section and 2 taps in the feedback one. The stochastic gradient algorithms operate with $\mu = 0.01$ and the gradient-newton techniques operate with $\alpha = 0.001$ and $\mu = 0.0001$. We consider a linear channel with transfer function $H(z) = 1.1 + 1.2z^{-1} - 0.2z^{-2}$. Fig. 6 shows the BER performance of the receivers.

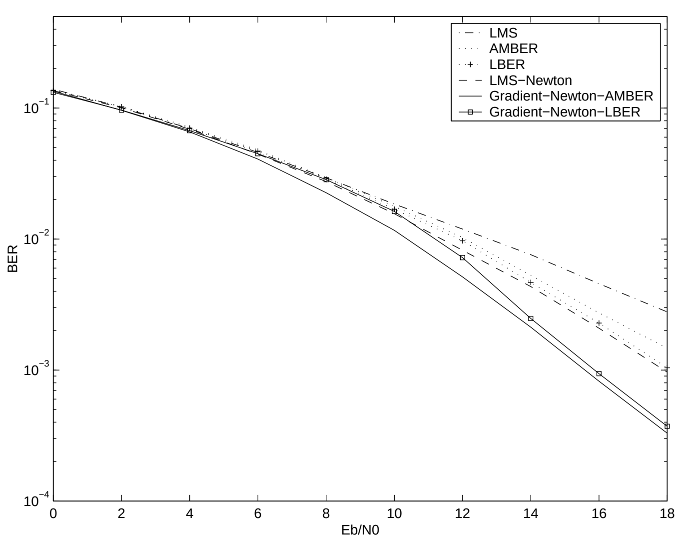

Fig. 6. BER performance using DFE equalisers for channel $H(z) = 1.1 + 1.2z^{-1} - 0.2z^{-2}$.

In the case of decision feedback equalisers, the Gradient-Newton-AMBER algorithm has achieved the best BER performance amongst the examined systems, followed by the Gradient-Newton-LBER, the LMS-Newton, the LBER, the AMBER and the LMS algorithms, as depicted in Fig. 6.

Note that the Gradient-Newton-LBER method performs well at high $\frac{E_b}{N_0}$, whilst the Gradient-Newton-AMBER shows good performance for both high and low $\frac{E_b}{N_0}$. In addition, the proposed Gradient-Newton type algorithms are superior to the well known LMS-Newton technique, whilst requiring a lower computational complexity and shorter training sequences.

VII. Concluding Remarks

We investigated the use of adaptive minimum bit error rate (MBER) Gradient-Newton algorithms for channel equalisation applications. The proposed algorithms approximate the bit error rate (BER) from training data using linear transversal and decision feedback (DFE) equaliser structures. A comparative analysis of linear and DFE equalisers, employing minimum mean squared error (MMSE), MBER and the proposed MBER algorithms was carried out. Computer simulation experiments have shown that the Gradient-Newton-AMBER and Gradient-Newton-LBER approaches outperform other analysed algorithms and can operate with shorter training sequences, even though they require a higher computational complexity.

References

[1] S. U. H. Qureshi, "Adaptive equalization", *Proceedings of IEEE*, vol. 73, pp. 1349-1387, 1985.

[2] J.K. Tugnait, Lang Tong and Zhi ding,"Single-user channel estimation and equalization", *IEEE Signal Processing Magazine* , vol. 17, pp. 17 -28, May 2000

[3] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*, Kluwer Academic Publishers, Boston, MA, 1997.

[4] C. Yeh and J. R. Barry, "Approximate minimum bit-error rate equalization for binary signaling", *IEEE Proc International Conference on Communications*, vol. 1, 1998, pp. 16-20.

[5] B. Mulgrew and S. Chen, "Stochastic Gradient Minimum BER Decision Feedback Equalisers," *IEEE Proc Adaptive Systems for Signal Processing, Communications, and Control Symposium.*, Canada, October 2000, pp. 93-98.