TRANSFORM DOMAIN QUASI-NEWTON ALGORITHMS FOR ADAPTIVE EQUALIZATION IN BURST TRANSMISSION SYSTEMS

Kostas Berberidis [*] *Stelios Rantos*[*] *Jacques Palicot* [♭]

Dept. of Computer Engineering & Informatics, and CTI
University of Patras, 26500 Rio-Patras, Greece
berberid@ceid.upatras.gr, rantos@ceid.upatras.gr

DMR/DDH FT R&D
BP 59, 35512 Cesson Sevigne, France
jacques.palicot@rd.francetelecom.fr

ABSTRACT

In this paper two new adaptive equalizers are proposed which belong to the quasi-Newton (QN) algorithmic family. The first algorithm is a Linear Equalizer (LE) and the second one is a Decision Feedback Equalizer (DFE). In the LE case the involved inverse Hessian matrix is approximated by a proper expansion consisting of powers of a Toeplitz matrix. Due to this formulation the algorithm can be efficiently implemented in the transform domain (TD) using FFT. The same idea is applied to the Feedforward part of the DFE. The derived algorithms enjoy the advantages of QN algorithms, that is, they exhibit faster convergence than their stochastic gradient counterparts and less computational complexity as compared to other Newton-type algorithms. These advantages are further enhanced due to TD implementation.

1. INTRODUCTION

The proposed paper is concerned with adaptive equalization in mobile wireless communication systems in which the transmission is done in bursts of data. A major cause of performance degradation in wireless mobile systems is the so-called multipath phenomenon. The introduced Intersymbol Interference (ISI) may reduce dramatically the probability of a correct decision in the receiver [1, 2]. Therefore, equalization turns out to be a major task in these cases. Moreover, since the channel may change significantly during the inter-bursts period, and even within a burst, the involved equalizer should be able to track the channel variations, and also have a fast convergence so as to need a reduced training sequence. This latter requirement implies that a corresponding saving in bandwidth may be achieved. Furthermore, since the adaptive equalizer is to operate in real time it should require a low computational burden.

Most of the existing symbol-by-symbol adaptive equalization schemes (either linear or DFE) belong to the stochastic gradient algorithmic family and they are implemented using variations of the well-known Least Mean Squares (LMS) algorithm. However, when the equalizer's input is colored the LMS-based DFE has slow convergence [3], hence does not lend itself for burst transmission systems with a short training sequence. The colored input results in an ill-conditioned autocorrelation matrix and the situation deteriorates as the filter order increases. A possible solution would be an RLS-based equalizer incorporating the Recursive Least Squares algorithm (or a fast version) to update the feedforward (FF) and feedback (FB) filters. However, the complexity of such a scheme is several times more than that of the LMS based schemes [1]. As for the fast-RLS schemes [4], their numerical behavior in certain implementation platforms is still an issue under investigation.

The algorithm proposed in this paper belongs to the QN family of algorithms. This family of algorithms lies between the LMS and RLS algorithms. They exhibit faster convergence rate than LMS and lower complexity than RLS. The characteristic of QN algorithms is that an approximation of the inverse Hessian matrix is involved in the filter updating relations in an attempt to whiten the input process. With appropriate choices of the inverse Hessian, the LMS and RLS algorithms can be considered as special cases of QN algorithms. Thus, under the same settings, the performance of a QN algorithm (in terms of convergence and tracking) depends mainly on the choice of the inverse Hessian matrix [4, 5].

In this paper the Hessian matrix is initially approximated by a Toeplitz matrix which is constructed by estimates of the autocorrelation sequence of the channel output. However, its inversion at each time step requires $O(M^2)$ operations, where M is the equalizer's order. To overcome this problem, the inverse Hessian matrix is expanded to a proper series, allowing fast, FFT-based, computation of the involved terms. Both the LE and DFE cases are treated in the paper. The resulting algorithms enjoy the advantages of QN algorithms, i.e. they exhibit faster convergence than their stochastic gradient counterparts and less computational complexity as compared to most Newton-type algorithms. These advantages are further enhanced due to transform domain implementation.

The following notation is used throughout the paper. In the time domain, vectors and matrices are denoted by bold lower case and bold upper case letters, respectively. In the frequency domain, vectors are denoted by calligraphic upper case letters.

2. PROBLEM FORMULATION AND PRELIMINARIES

The channel is assumed to be a discrete-time finite impulse response channel corrupted by additive white gaussian noise. The transmitted information sequence $\{d\}$ consists of i.i.d. symbols taken from a finite alphabet with zero mean and variance σ_d^2. The channel output is sampled at the symbol rate and the resulting sequence is denoted as $\{x\}$.

The algorithm proposed for estimating the LE filter is

$$y(n) = \boldsymbol{w}_M^H(n)\boldsymbol{x}_M(n) \quad (1)$$

$$e(n) = d(n) - y(n) \quad (2)$$

$$\boldsymbol{w}_M(n+1) = \boldsymbol{w}_M(n) + \mu \boldsymbol{R}^{-1}\boldsymbol{x}_M(n)e^*(n) \quad (3)$$

[*♭] This work was supported by FT R&D under contract no. 981B457 with Computer Technology Institute (CTI).

where $\boldsymbol{x}_M(n) = [x(n+M_1)\cdots x(n+M_1-M+1)]^T$ is the current input vector (M_1 is a proper delay of the LE), $\boldsymbol{w}_M(n)$ contains the LE coefficients and $y(n)$ is the output of the equalizer. $\boldsymbol{R}$ is a Toeplitz matrix constructed by a time-averaged estimator of the autocorrelation sequence $r_0, r_1, \ldots, r_{M-1}$ of $\{x\}$, i.e. $r_k(n) = \sum_{i=1+k}^{n} \beta^{n-i} x(i) x^*(i-k)$. The inversion of the (Toeplitz) autocorrelation matrix $\boldsymbol{R}$ requires $O(M^2)$ operations per time step if a Levinson-type algorithm is used. To alleviate this problem we propose, as an alternative to [6], the expansion of $\boldsymbol{R}$ as described in the subsection below.

2.1. Inverting the autocorrelation matrix

Introducing a real constant $\gamma > 0$ we consider the matrix

$$\frac{1}{\gamma}\boldsymbol{R} = \boldsymbol{I} + \boldsymbol{A} \tag{4}$$

where we have defined the matrix

$$\boldsymbol{A} = \begin{bmatrix} \left(\frac{r_0}{\gamma} - 1\right) & \frac{r_1}{\gamma} & \cdots & \frac{r_{M-1}}{\gamma} \\ \frac{r_1^*}{\gamma} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{r_1}{\gamma} \\ \frac{r_{M-1}^*}{\gamma} & \cdots & \frac{r_1^*}{\gamma} & \left(\frac{r_0}{\gamma} - 1\right) \end{bmatrix} \tag{5}$$

We will proceed further by exploiting a matrix analysis result. Let $\zeta_1, \ldots, \zeta_M$ be the eigenvalues of a matrix $\boldsymbol{S}$. Then $\varrho(\boldsymbol{S}) = \max_i |\zeta_i|$ denotes its spectral radius. Then according to a well-known theorem [7]: *The series* $\sum_{k=0}^{\infty} \boldsymbol{S}^k$ *converges iff* $\varrho(\boldsymbol{S}) < 1$. *Under this condition,* $\boldsymbol{I} - \boldsymbol{S}$ *is nonsingular and the limit of the series is equal to* $(\boldsymbol{I} - \boldsymbol{S})^{-1}$

Note that in our case, $\boldsymbol{S} = -\boldsymbol{A}$ is a Hermitian matrix and $\|\boldsymbol{S}\|_2 = \varrho(\boldsymbol{S})$. Using this theorem we may invert the autocorrelation matrix $\boldsymbol{R}$ as

$$\left(\frac{1}{\gamma}\boldsymbol{R}\right)^{-1} = (\boldsymbol{I} + \boldsymbol{A})^{-1} \tag{6}$$

$$\boldsymbol{R}^{-1} = \frac{1}{\gamma}(\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{A}^2 - \boldsymbol{A}^3 + \boldsymbol{A}^4 - \cdots) \tag{7}$$

Obviously we may approximate the inverse matrix as close as desirable using only the first q terms of the infinite sum, i.e.

$$\boldsymbol{R}^{-1} \approx \frac{1}{\gamma} \sum_{i=0}^{q-1} (-1)^i \boldsymbol{A}^i \tag{8}$$

The role of the constant γ is to ensure that $\|\boldsymbol{A}\|_2 < 1$ so as the above theorem to be applicable. Let $\lambda_1 \geq \cdots \geq \lambda_M > 0$ be the eigenvalues of $\boldsymbol{R}$. Then $(\frac{\lambda_1}{\gamma} - 1)\cdots(\frac{\lambda_M}{\gamma} - 1)$ are the eigenvalues of $\boldsymbol{A}$. Since $\|\boldsymbol{A}\|_2 = \varrho(\boldsymbol{A})$ and $\|\boldsymbol{R}\|_2 = \varrho(\boldsymbol{R}) = \lambda_1$ it is readily found that we must divide $\boldsymbol{R}$ with a constant

$$\gamma > 0.5\|\boldsymbol{R}\|_2 \tag{9}$$

but not too large so that the eigenvalues of $\boldsymbol{A}$ stay away from 1 in magnitude for the series to converge faster. Based on the hermitian symmetry of R and the well known inequality, [8], $\|\boldsymbol{R}\|_2 \leq \sqrt{\|\boldsymbol{R}\|_1 \|\boldsymbol{R}\|_\infty}$ we get that $\|\boldsymbol{R}\|_2 \leq |r_0| + 2\sum_{k=1}^{M-1} |r_k|$. Thus we may choose for γ the non-optimal value

$$\gamma = 0.5|r_0| + \sum_{k=1}^{M-1} |r_k| \tag{10}$$

3. THE LINEAR EQUALIZATION CASE

To end up with a practical scheme let us approximate $\boldsymbol{R}^{-1}$ as

$$\hat{\boldsymbol{R}}^{-1} = \frac{1}{\gamma}(\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{A}^2) \tag{11}$$

This yields the following update scheme

$$\boldsymbol{w}_M(n+1) = \boldsymbol{w}_M(n) + \frac{\mu e^*(n)}{\gamma}(\boldsymbol{x}_M(n) - \boldsymbol{A}\boldsymbol{x}_M(n) + \boldsymbol{A}^2\boldsymbol{x}_M(n)) \tag{12}$$

An important thing to notice is that matrix $\boldsymbol{A}$ is Toeplitz. So we may implement the product $\boldsymbol{A}\boldsymbol{x}$ efficiently in the frequency-domain by embedding $\boldsymbol{A}$ into the $2M \times 2M$ circulant matrix

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{A} \end{bmatrix} \tag{13}$$

where it is sufficient here to define the first column of $\boldsymbol{C}$ as $\boldsymbol{c} = [\boldsymbol{a}^*; 0; \boldsymbol{a}^R_{1:M-1}]$ with the help of some Matlab notation. The column vector $\boldsymbol{a}$ denotes the first row of $\boldsymbol{A}$. Vector $\boldsymbol{a}^R_{1:M-1}$ consists of the $M-1$ last elements of $\boldsymbol{a}$ in reverse order. If $\boldsymbol{F}$ is the DFT matrix of order $2M$ then $\boldsymbol{F}\boldsymbol{C}\boldsymbol{F}^{-1} = diag(\boldsymbol{F}\boldsymbol{c}) = \boldsymbol{D}$ and

$$\begin{bmatrix} \boldsymbol{A}\boldsymbol{x}_M(n) \\ \boldsymbol{0}_M \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_M & \boldsymbol{O}_M \\ \boldsymbol{O}_M & \boldsymbol{O}_M \end{bmatrix} \boldsymbol{F}^{-1}\boldsymbol{D}\boldsymbol{F} \begin{bmatrix} \boldsymbol{x}_M(n) \\ \boldsymbol{0}_M \end{bmatrix}$$

In other words, three DFTs and a pointwise vector-by-vector product suffice to carry out the matrix-vector multiplication.

So we may augment the update scheme as

$$\begin{bmatrix} \boldsymbol{w}_M(n+1) \\ \times \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_M(n) \\ \times \end{bmatrix} + \frac{\mu e^*(n)}{\gamma}\left(\begin{bmatrix} \boldsymbol{x}_M(n) \\ \times \end{bmatrix} - \begin{bmatrix} \boldsymbol{A}\boldsymbol{x}_M(n) \\ \times \end{bmatrix} + \begin{bmatrix} \boldsymbol{A}^2\boldsymbol{x}_M(n) \\ \times \end{bmatrix}\right).$$

We denote by $\times$, a "don't care", $M \times 1$ vector. Taking the DFT of both sides we have

$$\mathcal{W}(n+1) = \mathcal{W}(n) + \frac{\mu e^*(n)}{\gamma}(\mathcal{Q}_0(n) - \mathcal{Q}_1(n) + \mathcal{Q}_2(n)) \tag{14}$$

where $\mathcal{W}(n) = \boldsymbol{F}\begin{bmatrix} \boldsymbol{w}_M(n) \\ \times \end{bmatrix}$ and $\mathcal{Q}_i(n) = \boldsymbol{F}\begin{bmatrix} \boldsymbol{A}^i\boldsymbol{x}_M(n) \\ \times \end{bmatrix}$ for $i = 0, 1, 2$. We also define the frequency-domain vector $\mathcal{X}(n) = \boldsymbol{F}\begin{bmatrix} \boldsymbol{x}_M(n) \\ \boldsymbol{0}_M \end{bmatrix}$.

Let us now assume that the $\{r_k\}$ sequence is fixed from step to step. Then at time n we just have to update the frequency-domain vectors $\mathcal{Q}_i(n)$ in terms of the already computed vectors $\mathcal{Q}_i(n-1)$. More specifically we define that $\mathcal{Q}_0(n) = \mathcal{X}(n)$. From the DFT definition we note that, for $k = 0, 1, \ldots, 2M-1$,

$$\mathcal{X}_k(n) = \mathcal{X}_k(n-1)e^{-j\frac{\pi k}{M}} + x(n+M_1) - (-1)^k x(n+M_1-M)$$

If we define the corresponding vector-operation as

$$\mathcal{X}(n) = recdft(\mathcal{X}(n-1), x(n+M_1), x(n+M_1-M)),$$

where $recdft$ stands for recursive DFT, then we may write

$$\mathcal{Q}_0(n) = recdft(\mathcal{Q}_0(n-1), x(n+M_1), x(n+M_1-M)) \tag{15}$$

Furthermore, defining

$$\mathcal{Q}_1(n) = \boldsymbol{F}\begin{bmatrix} \boldsymbol{A}\boldsymbol{x}_M(n) \\ \boldsymbol{0}_M \end{bmatrix}, \quad (16)$$

it can be readily shown that $\boldsymbol{D}\mathcal{Q}_1(n) = \boldsymbol{F}\begin{bmatrix} \boldsymbol{A}^2\boldsymbol{x}_M(n) \\ \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}_M(n) \end{bmatrix}$. So, it is legal, according to the definition of $\mathcal{Q}_i(n)$'s, to compute $\mathcal{Q}_2(n)$ simply as

$$\mathcal{Q}_2(n) = \boldsymbol{D}\mathcal{Q}_1(n) \quad (17)$$

Finally, the computation of $\mathcal{Q}_1(n)$ is left. Defining $\boldsymbol{z}(n-1) = \boldsymbol{A}\boldsymbol{x}_M(n-1)$, we observe that

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x}_M(n) &= \begin{bmatrix} \boldsymbol{a}^T\boldsymbol{x}_M(n) \\ \boldsymbol{z}_{0:M-2}(n-1) \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{a}^*_{1:M-1} \end{bmatrix} x(n+M_1) \\ &\quad - \begin{bmatrix} 0 \\ \boldsymbol{a}^R_{1:M-1} \end{bmatrix} x(n+M_1-M) \end{aligned}$$

So, if we define the $2M \times 1$ frequency-domain vectors $\mathcal{A}_1 = \boldsymbol{F}\begin{bmatrix} 0 \\ \boldsymbol{a}^*_{1:M-1} \\ \boldsymbol{0}_M \end{bmatrix}$ and $\mathcal{A}_2 = \boldsymbol{F}\begin{bmatrix} 0 \\ \boldsymbol{a}^R_{1:M-1} \\ \boldsymbol{0}_M \end{bmatrix}$, then we may write in a compact way that

$$\begin{aligned} \mathcal{Q}_1(n) &= recdft(\mathcal{Q}_1(n-1), \boldsymbol{a}^T\boldsymbol{x}_M(n), \boldsymbol{a}^{RH}\boldsymbol{x}_M(n-1)) \\ &\quad + \mathcal{A}_1 x(n+M_1) - \mathcal{A}_2 x(n+M_1-M) \quad (18) \end{aligned}$$

To finish with the update equation we observe that the time-domain error signal can be computed by already available frequency domain vectors as

$$\begin{aligned} e(n) &= d(n) - \boldsymbol{w}_M^H(n)\boldsymbol{x}_M(n) \\ &= d(n) - \begin{bmatrix} \boldsymbol{w}_M(n) \\ \times \end{bmatrix}^H \begin{bmatrix} \boldsymbol{x}_M(n) \\ \boldsymbol{0}_M \end{bmatrix} \\ &= d(n) - \begin{bmatrix} \boldsymbol{w}_M(n) \\ \times \end{bmatrix}^H \frac{\boldsymbol{F}^H\boldsymbol{F}}{2M} \begin{bmatrix} \boldsymbol{x}_M(n) \\ \boldsymbol{0}_M \end{bmatrix} \\ &= d(n) - \frac{1}{2M}\mathcal{W}^H(n)\mathcal{X}(n) \quad (19) \end{aligned}$$

The resulting QN-LE scheme is summarized below:

Quantities defined at initialization: μ, λ, $\mathcal{W}(0)$, $P_k(-1)$
Quantities updated en block: r_k, γ, $\boldsymbol{a}$, $\boldsymbol{D}$, $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{Q}_1$
For $n = 0, 1, \ldots$ *(sample by sample)*
$\mathcal{Q}_0(n) = \mathcal{X}(n)$
$\mathcal{Q}'_1(n) = recdft(\mathcal{Q}_1(n-1), \boldsymbol{a}^T\boldsymbol{x}_M(n), \boldsymbol{a}^{RH}\boldsymbol{x}_M(n-1))$
$\mathcal{Q}_1(n) = \mathcal{Q}'_1(n) + \mathcal{A}_1 x(n+M_1) - \mathcal{A}_2 x(n+M_1-M)$
$\mathcal{Q}_2(n) = \boldsymbol{D}\mathcal{Q}_1(n)$
$\mathcal{Q}(n) = \mathcal{Q}_0(n) + \mathcal{Q}_1(n) + \mathcal{Q}_2(n)$
$e(n) = d(n) - \frac{1}{2M}\mathcal{W}^H(n)\mathcal{X}(n)$
$P_k(n) = \lambda P_k(n-1) + (1-\lambda)|\mathcal{X}_k(n)|^2$
$\boldsymbol{M}(n) = diag\left(P_0^{-1}(n), \ldots, P_{2M-1}^{-1}(n)\right)$
$\mathcal{W}(n+1) = \mathcal{W}(n) + \frac{\mu e^*(n)}{\gamma}\boldsymbol{M}(n)\mathcal{Q}(n)$
End

We note that we have used a matrix step size $\mathbf{M}(n)$ instead of the classic μ following [9, 10].

It can be shown that the above algorithm requires no more that $16M$ complex multiplications and a comparable number of complex additions per time step, plus 4 FFTs each time the autocorrelation sequence based quantities are updated (once per block, which block could be even the whole burst in a transmission system).

4. EXTENSION TO THE DFE CASE

The output $y(n)$ of a DFE equalizer at time n is given by

$$y(n) = \boldsymbol{c}^H_{M_f}(n)\boldsymbol{x}_{M_f}(n) - \boldsymbol{b}^H_{M_b}(n)\boldsymbol{d}_{M_b}(n) \quad (20)$$

where $\boldsymbol{x}_{M_f}(n)$ is the taps input vector of the FF filter at time n and $\boldsymbol{d}_{M_b}(n)$ is the taps input vector of the FB filter at time n. We may write the last equation as follows

$$\begin{aligned} y(n) &= \begin{bmatrix} \boldsymbol{c}^H_{M_f}(n) & -\boldsymbol{b}^H_{M_b}(n) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{M_f}(n) \\ \boldsymbol{d}_{M_b}(n) \end{bmatrix} \quad (21) \\ &\equiv \boldsymbol{w}^H_M(n)\boldsymbol{u}_M(n) \quad (22) \end{aligned}$$

The task here is to develop a quasi-Newton type adaptive algorithm for the updating of filter $\boldsymbol{w}_M = [\boldsymbol{c}_{M_f}; -\boldsymbol{b}_{M_b}]$. Unfortunately the derivation steps followed in the previous section cannot be directly extended to the DFE case. The reason is that the autocorrelation matrix of the input vector in the filter $\boldsymbol{w}$ loses its Toeplitz structure. Let us denote by $\boldsymbol{R}_u$ this matrix assuming for the moment stationarity where necessary. Then

$$\begin{aligned} \boldsymbol{R}_u &= E[\boldsymbol{u}_M(n)\boldsymbol{u}^H_M(n)] \quad (23) \\ &= E[\begin{bmatrix} \boldsymbol{x}_{M_f}(n) \\ \boldsymbol{d}_{M_b}(n) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^H_{M_f}(n) & \boldsymbol{d}^H_{M_b}(n) \end{bmatrix}] \quad (24) \\ &= E[\begin{bmatrix} \boldsymbol{x}_{M_f}(n)\boldsymbol{x}^H_{M_f}(n) & \boldsymbol{x}_{M_f}(n)\boldsymbol{d}^H_{M_b}(n) \\ \boldsymbol{d}_{M_b}(n)\boldsymbol{x}^H_{M_f}(n) & \boldsymbol{d}_{M_b}(n)\boldsymbol{d}^H_{M_b}(n) \end{bmatrix}] \quad (25) \\ &= \begin{bmatrix} \boldsymbol{R}_x & \boldsymbol{R}^H_{dx} \\ \boldsymbol{R}_{dx} & \boldsymbol{R}_d \end{bmatrix} \quad (26) \end{aligned}$$

wherefrom we see that the full autocorrelation matrix is no longer Toeplitz but block Toeplitz with $\boldsymbol{R}_d$ a diagonal matrix assuming i.i.d. information symbols $d(n)$.

As we have just mentioned, matrix $\boldsymbol{R}_x$ is only the upper left part of the full autocorrelation matrix $\boldsymbol{R}_u$. Moreover, we have assumed that $\boldsymbol{R}_d = \sigma_d^2\boldsymbol{I}$, so its inversion is trivial. What make the inversion of matrix $\boldsymbol{R}_u$ difficult to handle are the (Toeplitz) cross-terms $\boldsymbol{R}_{dx}$ and $\boldsymbol{R}^H_{dx}$. It can be shown that the matrix $\boldsymbol{R}_{dx}$ is composed of the causal coefficients (excluding the main echo h_0) of the channel's impulse response and so, for typical multipath channels, it is sparse with small-valued elements.

Thus, at a first approach, we may set these blocks equal to zero matrices and investigate the loss in performance in comparison with the ideal Newton/LMS algorithm. In that case, we have the approximation

$$\hat{\boldsymbol{R}}_u = \begin{bmatrix} \hat{\boldsymbol{R}}_x & \boldsymbol{O}_{M_f,M_b} \\ \boldsymbol{O}_{M_b,M_f} & \sigma_d^2\boldsymbol{I} \end{bmatrix} \quad (27)$$

and it is straightforward to apply the proposed scheme for the $\hat{\boldsymbol{R}}_x$. That is, we update the FF filter by using the QN-LE scheme as described in the previous section, while we update the FB filter by applying the TD-LMS technique.

5. SIMULATIONS AND DISCUSSION

To illustrate the performance of the algorithms we provide some simulation results. Two typical wireless channels, named as channel A and channel B, have been used [11]. Channel A has the values [0.9333, 0.5012, 0.5129, 0.5370] at taps [1,8,15,22] and channel B has the values [0.7490, 1, 0.2290, 0.3160, 0.0550, 0.1580] at

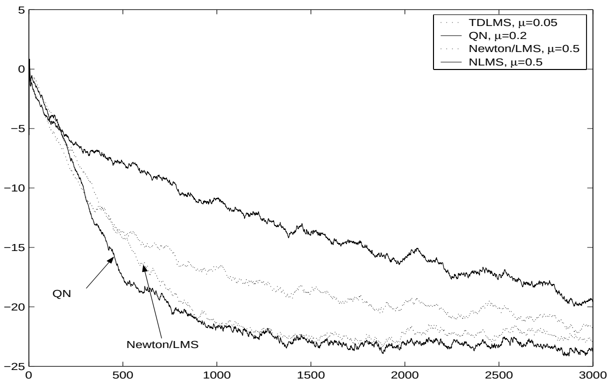

Fig. 1. channel A - LE(128,16)

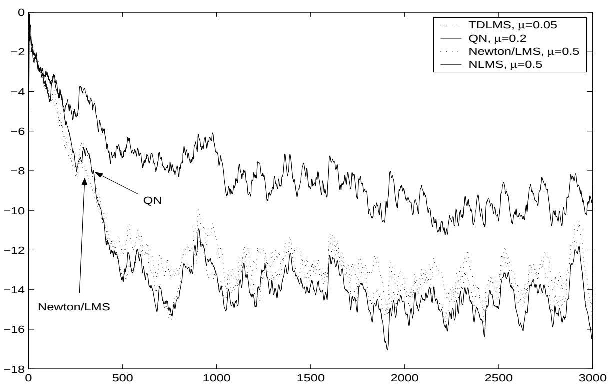

Fig. 2. channel B - LE(128,16)

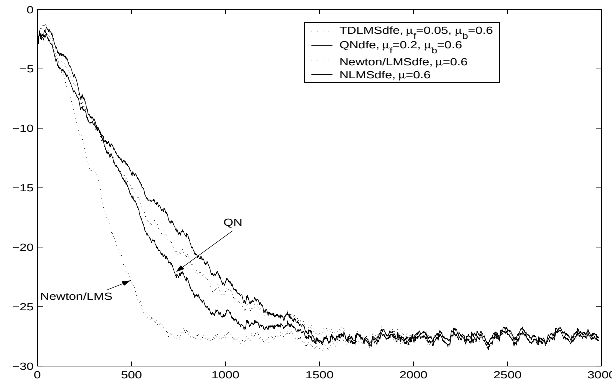

Fig. 3. channel A - DFE(16,64)

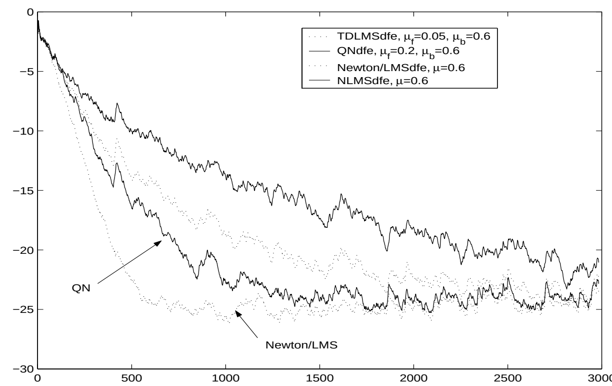

Fig. 4. channel B - DFE(16,64)

taps [1, 2, 18, 26, 34, 40]. Clearly, channel B results in more serious ISI than channel A. We study the performance using a $M = 128$-order LE (with a $M_1 = 16$ taps noncausal length), and a DFE stucture with a noncausal FF filter of order $M_f = 16$ and a causal FB filter of order $M_b = 64$. The results are shown on figures 1-4 comparing the Quasi-Newton algorithm with Newton/LMS, TD-LMS and NLMS algorithms. We update the matrix step-size for the TD-LMS and QN algorithms as suggested in [10]. A suitable matrix step-size initialization/update is still under investigation for the QN approach. For λ the value 0.9995 was chosen.

By inspecting figures 1-2, we conclude that the new QN-LE algorithm has a performance which is better as compared to TD-LMS and approaches that of the ideal Newton/LMS. From figures 3-4 we deduce that the same comment holds for the QN-DFE algorithm as compared to the TD-LMS-DFE.

6. REFERENCES

[1] J.G. Proakis, "Adaptive Equalization for TDMA Digital Mobile Radio", *IEEE Transactions on Vehicular Technology*, vol.40, no.2, pp. 333-341, May 1991.

[2] T.S. Rappaport, "Wireless Communications: Principles and Practice", Prentice-Hall, 1996.

[3] Simon Haykin, "Adaptive Filter Theory", *Prentice Hall*, 1996.

[4] G. Glentis, K. Berberidis, S. Theodoridis, "Efficient Least Squares Adaptive Algorithms for FIR Transversal Filtering: A Unified View", *IEEE Signal Processing Magazine*, vol.16, no.4, pp. 13-41, July 1999.

[5] Kostas Berberidis, Jacques Palicot, "A frequency domain quasi-Newton algorithm", *Signal Processing*, vol. 47, pp. 235-238, 1995.

[6] Daniel F. Marshall, W. Kenneth Jenkins, "A Fast Quasi-Newton Adaptive Filtering Algorithm", *IEEE Transactions on Signal Processing*, vol.40, no.7, pp. 1652-1662, July 1992.

[7] Yousef Saad, "Iterative Methods for Sparse Linear Systems", *Pws Publishing Company*, 1996.

[8] Gene H. Golub, Charles F. Van Loan, "Matrix Computations", *Johns Hopkins*, 1996.

[9] S. Shankar Narayan, Allen M. Peterson, Madihally J. Narasihma "Transform Domain LMS Algorithm", *IEEE Trans. ASSP*, vol. ASSP-31, pp 609-615, June 1983.

[10] Jae Chon Lee, Chong Kwan Un, "Performance of Transform-Domain LMS Adaptive Digital Filters", *IEEE Trans. ASSP*, vol. ASSP-34, pp 499-510, June 1986.

[11] ETSI, UMTS Technical Report 101 112, version 3.2.0, 1998-04.