A BLOCK ADAPTIVE DFE IN THE FREQUENCY DOMAIN BASED ON TENTATIVE DECISIONS

Kostas Berberidis and *Panos Karaivazoglou*

Department of Computer Engineering & Informatics
School of Engineering, University of Patras
26500 Patras, GREECE
Fax: +30-61-991909 e-mails: berberid@cti.gr and karaivap@ceid.upatras.gr

ABSTRACT

In this paper a new block adaptive DFE implemented in the frequency domain is derived. The new algorithm is suitable for rejecting ISI due to multipath echoes, especially to wireless transmission systems which involve channels with long impulse response. The novel idea is to use tentative decisions properly derived by minimizing a frequency domain criterion. The algorithm has a steady-state performance which is practically identical to that of the symbol-by-symbol DFE. At the same time it offers a faster convergence rate and substantial savings in complexity. Additionally, the level of the steady-state MSE achieved by the algorithm is insensitive to the length of the block. Thus the new algorithm allows a trade off between complexity and processing delay without affecting performance.

1 INTRODUCTION

An important problem encountered in many wireless transmission systems is the so-called multipath phenomenon. In most cases this phenomenon introduces severe intersymbol interference (ISI) which must be drastically eliminated in order to permit any useful transmission at all. The problem becomes even more important in cases where the involved channel has a relatively long impulse response. Typical applications of the kind are multipoint multichannel distribution systems (MMDS), very high speed local distributions systems, digital TV terrestial broadcasting systems etc. In all these applications the impulse response of the multipath channel spans a time interval equal to several symbol periods. In some cases the involved channel impulse response is very long, in particular the causal part which may last up to several hundreds of symbol periods.

It is well established in the literature (e.g. [1], [4]) that ISI introduced by channels of the above type can be effectively rejected using adaptive Decision Feedback Equalizers (DFE). However the implementation of long DFEs in today's available hardware is a very difficult task. A possible way to cope with this difficulty is to use block adaptive equalizers preferably implemented in the frequency domain. Frequency domain adaptive (FDA) filters have been extensively studied in the literature (e.g. see [2], [3]). As compared to the sample-by-sample adaptive filters they exhibit lower complexity, faster convergence and equivalent steady state performance. However most of the existing and well-known FDA filters are suitable only for linear equalization (e.g. FLMS algorithm in [2]). As shown in [4] linear equalizers may exhibit poor performance in channels of the above type and hence they are disqualified from being used in the corresponding applications.

An attempt for a frequency domain block DFE was presented in [4] and the technique developed there enjoys the advantages of FDA schemes, i.e. ease of implementation, modularity, very low complexity and fast convergence. However in the algorithm of [4], for reasons of computational efficiency, the heading part of the involved feedback filter is truncated by a fraction of the block length. Due to this fact, if strong near echoes are present in the causal part of the channel impulse response, the performance of the algorithm is inferior to that of the symbol by symbol DFE.

In this paper a new block adaptive DFE implemented in the frequency domain is developed. The main novelty in the developed algorithm is to use properly derived tentative decisions. These tentative decisions, corresponding to the current block, are provided beforehand by a modified version of the block DFE minimizing a cost function in the frequency domain. The tentative decisions enter the filtering and the updating part of a full length block DFE implemented in the frequency domain. Note that most part of the computations needed in this part have already been done in the first part. The algorithm has a steady-state performance practically identical to that of the symbol-by-symbol DFE and remarkably faster convergence rate. It offers a flexibility in the choice of the block length and its complexity is substantially lower as compared to the symbol by symbol DFE.

The new Tentative Decisions based Block DFE (TDB-DFE) is derived in Section 2 while simulation results verifying its performance are presented in Section 3. The notations used in the paper is as follows. In the time domain: scalar variables, vectors, and matrices, are denoted respectively by lower case letters, lower case bold, and upper case letters. In the frequency domain: vectors are given in bold upper case letters, and matrices are denoted by calligraphic upper case letters.

2 THE TDB-DFE ALGORITHM

2.1 Problem Description

Due to the multipath phenomenon the received signal consists of several components with each one being a scaled, delayed and phase shifted version of the original transmitted signal. Notice that in most of the applications at hand the precursor part of the channel's impulse response is in general much shorter and of less energy as compared to the postcursor part. The same holds for the impulse response of the inverse channel. Indeed, as it was shown in [4], this response is mostly extented to the postcursor direction spanning a time interval much longer than the respective part of the impulse response of the direct channel. Note also that quite often the direct channel spectrum exhibits very deep nulls (as in cases of strong far echoes). The above explain intuitively why linear equalizers are in general unsuitable for the applications of interest. On the contrary DFEs match perfectly with the above characteristics. Indeed, the ISI caused by the long causal part of the impulse response can be drastically eliminated by a long Feedback (FB) filter. On the other hand the small remaining amount of ISI due to the anticausal part may be effectively reduced using a relatively short Feedforward (FF) filter.

To proceed to our derivation let us first recall the conventional symbol-by-symbol DFE described by the following equations

$$y(n) = \boldsymbol{a}_M^T(n)\boldsymbol{x}_M(n+M-1) + \boldsymbol{b}_N^T(n)\boldsymbol{d}_N(n-1) \quad (1)$$
$$d(n) = f\{y(n)\} \quad (2)$$
$$e(n) = y(n) - d(n) \quad (3)$$
$$\boldsymbol{a}_M(n+1) = \boldsymbol{a}_M(n) + 2\mu^a \boldsymbol{x}_M^*(n+M-1)e(n) \quad (4)$$
$$\boldsymbol{b}_N(n+1) = \boldsymbol{b}_N(n) + 2\mu^b \boldsymbol{d}_N^*(n-1)e(n) \quad (5)$$

where $\{x\}$ and $\{d\}$ denote the equalizer's input and decision sequences respectively. Vector $\boldsymbol{d}_N(n-1) = [d(n-1)\ldots d(n-N)]^T$ and vector $\boldsymbol{x}_M(n+M-1)$ is defined in a similar way. Vectors $\boldsymbol{a}_M(n)$ and $\boldsymbol{b}_N(n)$ denote the estimates at time n of the $M-th$ order FF filter and the $N-th$ order FB filter respectively. $f\{\cdot\}$ in Eq. (2) stands for the decision device function.

Our aim is to get a block of decisions at a time, say Q decisions, where $Q \leq N$. The FF and FB filters will be updated every Q steps. Let us first derive a block formulation of the DFE equations, i.e.

$$\boldsymbol{y}_Q(n+Q-1) = X_{Q\times M}\boldsymbol{a}_M(n) + D_{Q\times N}\boldsymbol{b}_N(n) \quad (6)$$
$$\boldsymbol{d}_Q(n+Q-1) = f\{\boldsymbol{y}_Q(n+Q-1)\} \quad (7)$$
$$\boldsymbol{e}_Q(n+Q-1) = \boldsymbol{y}_Q(n+Q-1) - \boldsymbol{d}_Q(n+Q-1) \quad (8)$$
$$\boldsymbol{a}_M(n+Q-1) = \boldsymbol{a}_M(n) + 2\mu^a X_{Q\times M}^H \boldsymbol{e}_Q(n+Q-1) \quad (9)$$
$$\boldsymbol{b}_N(n+Q-1) = \boldsymbol{b}_N(n) + 2\mu^b D_{Q\times N}^H \boldsymbol{e}_Q(n+Q-1) \quad (10)$$

where

$$X_{Q\times M} = \begin{bmatrix} x(n+M+Q-2) & \ldots & x(n+Q-1) \\ \vdots & \ddots & \vdots \\ x(n+M-1) & \ldots & x(n) \end{bmatrix}$$

and

$$D_{Q\times N} = \begin{bmatrix} D_{Q\times Q-1}^1 & D_{Q\times N-Q+1}^2 \end{bmatrix} \quad (11)$$

with

$$D_{Q\times Q-1}^1 = \begin{bmatrix} d(n+Q-2) & \ldots & d(n) \\ \vdots & \ddots & \vdots \\ d(n-1) & \ldots & d(n-Q+1) \end{bmatrix}$$

$$D_{Q\times N-Q+1}^2 = \begin{bmatrix} d(n-1) & \ldots & d(n+Q-N-1) \\ \vdots & \ddots & \vdots \\ d(n-Q) & \ldots & d(n-N) \end{bmatrix}$$

If $\boldsymbol{y}_Q(n+Q-1)$ were available then, according to (7), vector $\boldsymbol{d}_Q(n+Q-1)$ would be the one with the shortest euclidean distance from $\boldsymbol{y}_Q(n+Q-1)$ and would contain the Q detected symbols (decisions) of the current block. However vector $\boldsymbol{y}_Q(n+Q-1)$ cannot be computed directly from (6) since matrix $D_{Q\times N}$ contains unknown decisions, specifically all $d(k)$ for $k = n,\ldots,n+Q-2$. All these unknown decisions lie in the upper left $Q-1\times Q-1$ triangular part of matrix $D_{Q\times Q-1}^1$. Due to this fact the well studied block adaptive techniques [3], which apply perfectly for linear equalizers, are not directly applicable in the DFE case.

2.2 Derivation of the TDB-DFE Algorithm

As mentioned in Section 1 the main idea is to use tentative decisions in place of the unknown decisions. The tentative decisions will be derived via a minimization procedure in the frequency domain. To proceed further let us partition the FB filter as $\boldsymbol{b}_N(n) = [\boldsymbol{b}_{Q-1}^{1T}\ \boldsymbol{b}_{N-Q+1}^{2T}]^T$. Using this partitioned form in (6) we obtain

$$\boldsymbol{y}_Q(n+Q-1) = \boldsymbol{y}_Q^p(n+Q-1) + D_{Q\times Q-1}^1\boldsymbol{b}_{Q-1}^1 \quad (12)$$

where

$$\boldsymbol{y}_Q^p(n+Q-1) = X_{Q\times M}\boldsymbol{a}_M(n) + D_{Q\times N-Q+1}^2\boldsymbol{b}_{N-Q+1}^2 \quad (13)$$

is the part of $\boldsymbol{y}_Q(n+Q-1)$ which is based on known quantities only and therefore it can be computed. Note that tentative decisions could be obtained by simply passing $\boldsymbol{y}_Q^p(n+Q-1)$ through the decision device. However in such a case the available information is not fully exploited. Next the error vector corresponding to the current block can be written as

$$\boldsymbol{e}_Q(n+Q-1) = \boldsymbol{y}_Q^p(n+Q-1) + B_{Q\times 2Q-1}\boldsymbol{d}_{2Q-1}(n+Q-1) - \boldsymbol{d}_Q(n+Q-1) \quad (14)$$

where

$$B_{Q\times 2Q-1} = \begin{bmatrix} 0 & b_1 & \ldots & b_{Q-1} & 0 & \ldots & 0 \\ 0 & 0 & b_1 & \ldots & b_{Q-1} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & b_1 & \ldots & b_{Q-1} \end{bmatrix}$$

Note that the matrix-by-vector product of the right hand side of (12) is now written in an equivalent representation which reveals the involved linear convolution operation. Before getting into the frequency domain we append Q zeros at the end of vectors $\boldsymbol{e}_Q(n+Q-1)$ and $\boldsymbol{y}_Q^p(n+Q-1)$ (changing their size to $2Q$) and we define the following circulant matrix

$$B_{2Q}^c = circulant\left\{[\mathbf{0}_{Q+1}^T \; b_{Q-1} \; \ldots \; b_1]^T\right\} \tag{15}$$

i.e. matrix B_{2Q}^c is a $2Q \times 2Q$ circulant matrix whose first column is the one given in (15). Now we can write (14) as

$$\left[\begin{array}{c} \boldsymbol{e}_Q(n+Q-1) \\ \mathbf{0}_Q \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y}_Q^p(n+Q-1) \\ \mathbf{0}_Q \end{array}\right] + C_{2Q}^1\left[B_{2Q}^c \boldsymbol{d}_{2Q}(n+Q-1) - \boldsymbol{d}_{2Q}(n+Q-1)\right] \tag{16}$$

where C_{2Q}^1 is a $2Q \times 2Q$ constraint matrix which is imposed in order to keep only the linear convolution part, and is defined as

$$C_{2Q}^1 = \left[\begin{array}{cc} \boldsymbol{I}_{Q\times Q} & \mathbf{0}_{Q\times Q} \\ \mathbf{0}_{Q\times Q} & \mathbf{0}_{Q\times Q} \end{array}\right]$$

From (16) we can easily get

$$\left[\begin{array}{c} \boldsymbol{e}_Q(n+Q-1) \\ \mathbf{0}_Q \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y}_Q^p(n+Q-1) \\ \mathbf{0}_Q \end{array}\right] + C_{2Q}^1 B_{2Q}^1 \boldsymbol{d}_{2Q}(n+Q-1) \tag{17}$$

where $B_{2Q}^1 = B_{2Q}^c - I_{2Q}$. Next applying the Fourier operator (denoted as F) to both sides of (17) we obtain

$$\boldsymbol{E}_{2Q} = \boldsymbol{Y}_{2Q}^p + F \cdot C_{2Q}^1 \cdot F^{-1} \cdot \mathcal{B}_{2Q}^1 \cdot \boldsymbol{D}_{2Q} \tag{18}$$

Next we minimize the total squared error within the block with respect to the decision vector and then we use the derived "decisions" in order to compute the unknown part of equation (12). Setting $\frac{\partial(\boldsymbol{E}_{2Q}^H \boldsymbol{E}_{2Q})}{\partial \boldsymbol{D}_{2Q}} = 0$ and after some algebra we obtain

$$\boldsymbol{D}_{2Q}^H \cdot [\mathcal{B}_{2Q}^{1H} F^{-H} C_{2Q}^1 F^H \mathcal{B}_{2Q}^1] = -\boldsymbol{Y}_{2Q}^{pH} F C_{2Q}^1 F^{-1} \mathcal{B}_{2Q}^1$$

If the constraint matrix C_{2Q}^1 was invertible we could write

$$\boldsymbol{D}_{2Q} = -\mathcal{B}_{2Q}^1 \boldsymbol{Y}_{2Q}^p \tag{19}$$

The constraint matrix can be made invertible by substituting every zero element in its main diagonal with a very small number ϵ. Note that ϵ introduces only a negligible error since we are free to choose a very small value for it. Thus from (19) and taking inverse FFT of $\boldsymbol{D}_{2Q}$ we obtain Q decisions for the current block, which could be used as tentative ones. Note however that at the same time we get Q decisions corresponding to the previous block. However these decisions are already known. To exploit this knowledge we split the $2Q$-length decision vector into two components as

$$\boldsymbol{d}_{2Q}(n+Q-1) = \left[\begin{array}{c} \boldsymbol{d}_Q(n+Q-1) \\ \mathbf{0}_Q \end{array}\right] + \left[\begin{array}{c} \mathbf{0}_Q \\ \boldsymbol{d}_Q(n-1) \end{array}\right] \tag{20}$$

By substituting (20) in (17) and performing the above minimization procedure with respect to the unknown component of the decision vector in (20) we get

$$\hat{\boldsymbol{D}}_{2Q} = -\mathcal{B}_{2Q}^1 \hat{\boldsymbol{Y}}_{2Q} \tag{21}$$

where $\hat{\boldsymbol{Y}}_{2Q}$ is the IFFT of vector

$$\left[\begin{array}{c} \boldsymbol{y}_Q^p(n+Q-1) \\ \mathbf{0}_Q \end{array}\right] + C_{2Q}^1 \cdot B_{2Q}^1 \cdot \left[\begin{array}{c} \mathbf{0}_Q \\ \boldsymbol{d}_Q(n-1) \end{array}\right] \tag{22}$$

Taking inverse FFT of $\hat{\boldsymbol{D}}_{2Q}$ and using the resulting decisions in (12) we can compute $\boldsymbol{y}_Q(n+Q-1)$. Passing this vector through the decision device we obtain the decisions which are considered tentative and enter the filtering and updating part of the full length TDB-DFE.

2.3 Summary of the TDB-DFE algorithm

Filtering Part:

1. Compute:
$\boldsymbol{y}_Q^p(n+Q-1) = X_{Q\times M}\boldsymbol{a}_M + D_{Q\times N-Q+1}^2 \boldsymbol{b}_{N-Q+1}^2$
2. Define: $B_{2Q}^1 = diag\left\{F[-1 \; \mathbf{0}_Q^T \; b_{Q-1} \; \ldots \; b_1]\right\}$
Compute: $\hat{\boldsymbol{D}}_{2Q} = -\mathcal{B}_{2Q}^1 \cdot \hat{\boldsymbol{Y}}_{2Q}$
with $\hat{\boldsymbol{Y}}_{2Q}$ as defined in (22).
Compute: $\hat{\boldsymbol{d}}_Q(n+Q-1) = [I_Q \; 0_Q]F^{-1}\hat{\boldsymbol{D}}_{2Q}$
3. Compute the tentative decisions $\tilde{\boldsymbol{d}}_Q(n+Q-1)$ by passing the sum $\boldsymbol{y}_Q^p(n+Q-1) + \hat{D}_{Q\times Q-1}^1 \boldsymbol{b}_{Q-1}^1$ through the decision device.
4. Compute the final decisions $\boldsymbol{d}_Q(n+Q-1)$ by passing the sum $\boldsymbol{y}_Q^p(n+Q-1) + \tilde{D}_{Q\times Q-1}^1 \boldsymbol{b}_{Q-1}^1$ through the decision device.

The filtering part can be implemented in the frequency domain using either *Overlap-Save* or *Overlap-Add* sectioning methods. It should be noticed that the tentative decisions derived using the suggested minimization can be considered as initial values for a nonlinear recurrent procedure. This issue is further investigated in a forthcoming paper.

Updating Part:

1. Compute: $\boldsymbol{E}_{2Q}(k) = F\{[\mathbf{0}_Q^T \; \boldsymbol{e}_Q^T(n+Q-1)]^T\}$
and $\boldsymbol{E}_{2N}(k) = F\{[\mathbf{0}_{2N-Q}^T \; \boldsymbol{e}_Q^T(n+Q-1)]^T\}$
2. Compute: $\mathcal{X}_{2Q}(k) =$
$diag\{F[x(n-Q+M-1) \; \ldots \; x(n+Q+M-2)]\}$
and $\mathcal{D}_{2N}(k) =$
$diag\{F \cdot [d(n-2N+Q-1) \; \ldots \; d(n+Q-2)])\}$
3. Compute: $\boldsymbol{P}_{2Q}^x(k) =$
$\lambda_1 \boldsymbol{P}_{2Q}^x(k-1) + (1-\lambda_1)\mathcal{X}_{2Q}^*(k)\mathcal{X}_{2Q}(k) \cdot \mathbf{1}_{2Q}$
and $\boldsymbol{P}_{2N}^d(k) =$
$\lambda_2 \boldsymbol{P}_{2N}^d(k-1) + (1-\lambda_2)\mathcal{D}_{2N}^*(k)\mathcal{D}_{2N}(k) \cdot \mathbf{1}_{2N}$
4. Compute the matrix step size for each filter
$\mathcal{M}_{2Q}^a(k) = \mu_1 \cdot diag\{[p_{2Q,1}^{-x}(k) \; \ldots \; p_{2Q,2Q}^{-x}(k)]\}$ and
$\mathcal{M}_{2N}^a(k) = \mu_2 \cdot diag\{[p_{2N,1}^{-d}(k) \; \ldots \; p_{2N,2N}^{-d}(k)])\}$
5. Update the FF and FB filters
$\boldsymbol{A}_{2Q}(k+1) =$
$\boldsymbol{A}_{2Q}(k) + 2FC_{2Q}^a F^{-1}\mathcal{M}_{2Q}^a(k)\mathcal{X}_{2Q}^*(k)\boldsymbol{E}_{2Q}(k)$

$$\boldsymbol{B}_{2N}(k+1) = \boldsymbol{B}_{2N}(k) + 2FC^{b}_{2N}F^{-1}\mathcal{M}^{b}_{2N}(k)\mathcal{D}^{*}_{2N}(k)\boldsymbol{E}_{2N}(k)$$

The updating part above has been given in a more detailed form in order to show the use of the involved matrix step sizes. k denotes the current block. Constraint matrices C^{a}_{2Q} and C^{b}_{2N} are defined in a similar way as matrix C^{1}_{2Q} in the previous subsection. They have I_M and I_N respectively in their upper left parts and zeros elsewhere. $\mathbf{1}_{2Q}$ is a $2Q$ length vector with unity elements and $\mathbf{1}_{2N}$ is defined accordingly. A good choice for $\lambda_i,\ i = 1,2$ is $1-\mu_i$. The initial vectors for $\boldsymbol{P}^{x}_{2Q}$ and $\boldsymbol{P}^{d}_{2N}$ are vectors with unity elements.

3 Simulation Results

The experiments for testing the TDB-DFE algorithm were carried out on a multipath channel with an impulse response consisting of 7 echoes. The amplitudes of the echoes were $-21dB$, $-16dB$, $-10dB$, $-14dB$, $-18dB$, $-20dB$, and $-8dB$ and their corresponding time delays were equal to $-17T_s$, $-11T_s$, $25T_s$, $93T_s$, $151T_s$, $180T_s$, and $235T_s$ respectively, where T_s was the symbol period. The phases of the echoes were chosen randomly. The input to the channel was a QPSK sequence while at the output white complex gaussian noise was added.

In Fig. 1 the TDB-DFE is compared with the conventional LMS algorithm and the symbol-by-symbol DFE (as given by Eqs. (1)-(5)). The lengths of the FF and FB filters were equal to 32 and 256 respectively. The block length for TDB-DFE was 64. A common step size was used for all algorithms. The two upper MSE curves correspond to LMS 256 and 384 coefficients. We can see that there is an improvement in performance if we use 384 instead of 256 coefficients, but LMS still cannot achieve the performance of DFE. It should be noticed is that the curves corresponding to TDB-DFE and the conventional DFE are practically the same. In this experiment TDB-DFE was implemented with fixed step sizes in the updating part.

In Fig. 2 the symbol-by-symbol DFE is compared with the TDB-DFE (with block length equal to 64 and matrix-step size in the updating part). As expected TDB-DFE exhibits a faster convergence rate as compared to DFE for the same misadjustment.

Finally in Fig. 3 we have the MSE curves of TDB-DFE for three different block sizes, namely 64, 128, and 224. The curves are practically the same showing that the size of the block does not affect the performance of the new algorithm.

References

[1] J.G. Proakis, M. Salehi "Communication Systems Engineering", *Prentice Hall International Editions.*, 1994.

[2] E.R. Ferrara, "Frequency-domain Adaptive Filtering", in *"Adaptive Filters"*, C.F.N. Cowan and P.M. Grant, Eds., *Prentice Hall*, 1985, ch. 6, pp. 145-179.

[3] J.J. Shynk, "Frequency-domain and Multirate Adaptive Filtering", *"IEEE Signal Processing Magazine"*, Jan. 1992, pp. 14-37.

[4] K. Berberidis, J. Palicot, "A New Frequency-Domain Equalizer for Channels with Long Impulse Response", *EUSIPCO-96*, Trieste, Sep. 1996.

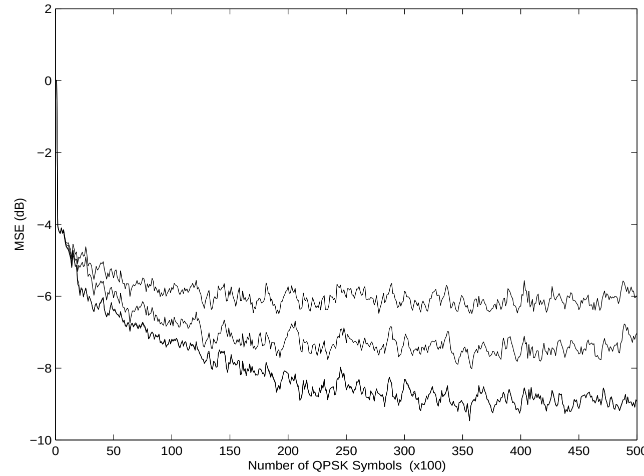

FIGURE 1

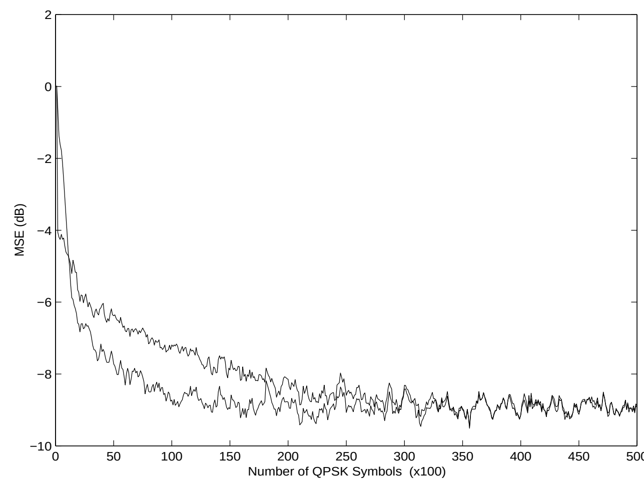

FIGURE 2

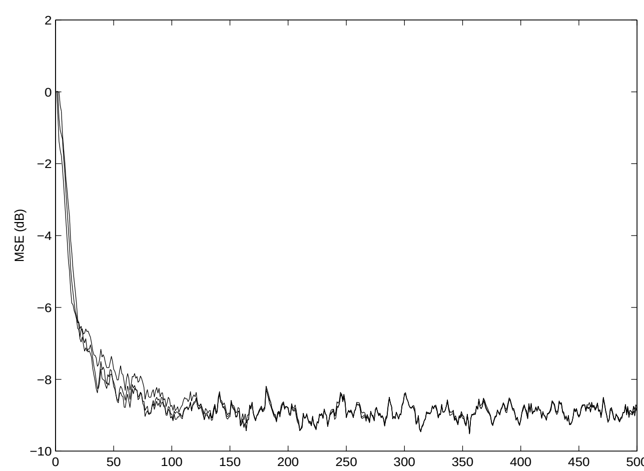

FIGURE 3