A Modem without a Training Signal: A New Approach

Bandula Dahanayake
Farmfield Crescent, Ottawa, Ontario, Canada K2M 2S9
Bandula_Dahanayake@yahoo.com

***Abstract*—In transmitting digital information, the Modulator-Demodulators (Modems) and the Digital Communication Systems, in general, depend on equalizers in order to eliminate or minimize inter symbol interference. Equalizers used in practice require a training signal that is known to the receiver to initialize a communication system so that it can eliminate the inter symbol interference. Here, a new approach for eliminating the need for a training signal in initializing a modem or a communication system is introduced based on the fractional sampling. Unlike the existing block processing equalizers based on fractional sampling, the proposed equalizer is data adaptive. It is a fast learning Blind Adaptive Equalizer that does not require a training signal; a self learning algorithm. The Blind Adaptive Equalizer is capable of estimating the channel parameters and the transmitted symbols adaptively irrespective of the phase properties of the channel. The learning characteristic of the Blind Adaptive Equalizer is independent of the signal to noise ratio (snr) at high snr. The fast decaying characteristic of the learning curves indicates its fast learning capability. At higher signal to noise ratios, the variance of the channel parameter estimates of the Blind Adaptive Equalizer is negligibly small. In addition, the Blind Adaptive Equalizer achieves zero probability of error at high signal to noise ratios. It can also be used in signal interception due to its blind adaptive nature.**

***Keywords*—Adaptive-Equalizer; Channel-Estimation; Communication; Blind-Equalizer;**

I. INTRODUCTION

All the practical communication channels are band limited. When a channel is band limited in frequency domain, its resulting impulse response in time domain will be unbounded. This causes a transmitted symbol to be spread out into the adjacent symbols causing inter symbol interference. When the signal is received at the receiver, its first task before any attempt to estimate the symbols, would be to eliminate or minimize the effect of the channel from the received signal so that the inter symbol interference will be eliminated or minimized. In order to do that, it is necessary to estimate the channel before start receiving the information. In general, digital communication system starts transmitting a signal known to the receiver so that the receiver can estimate the channel and eliminate the effect of the channel; the device achieve this process is known as a equalizer. Equalizers can be block processors where the block of received known signal is used to estimate the channel, or they can be data adaptive where parameters of the equalizer are updated based on each data point. Adaptive equalizers are able to respond to changing channel environments.

In order to avoid the use of a training signal, a fractional sampling based block processing equalizer was proposed in [1]. In fractional sampling, a symbol is oversampled M times to convert signal channel into a M-multichannel system. If the regular sampling rate is f_s then, the fractional sampling is achieved by sampling at Mf_s. Then, the cross correlation characteristic of the multichannel outputs are used to estimate the channel parameters based on the generalized eigenvalue-eigenvector decomposition. This eliminates the need for a training signal. It was also assumed the channel to be cyclo-stationary. In spite of it good performance characteristic, it hasn't been practical due to it block processing nature. Now, the question is how we can use the cross correlation characteristics of the multichannel outputs to design an adaptive equalizer? If we can answer this question, we will have a Blind Adaptive Equalizer.

II. SIGNAL MODEL

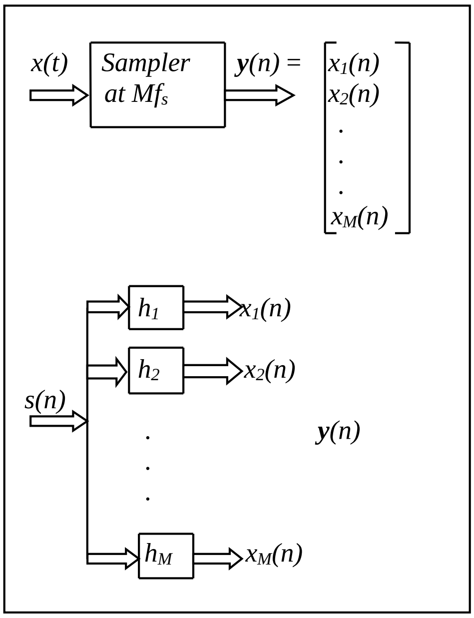
We start with multi-channel signal model of a communication system obtained by using fractional sampling. Let us assume that we sample the signal $x(t)$, the received baseband signal, M times the symbol rate. Here, M is arbitrary and has to be greater than the order of the channel, L. Both the order of the channel, L and the rate of fractional sampling M are unknown. We present a methodology for determining proper L and M later.

When we sample the signal $x(t)$, the received baseband signal, M times the symbol rate, each symbol interval contains M samples. Let the m^{th} sample of the n^{th} received symbol be denoted by,

$$x_m(n),\ m=1, 2, ...,M,\ n=1, 2, ...,$$

n denotes the symbol identifier, and m denotes the fractional sampling identifier in a symbol.

Single to multi-channel model can be illustrated as follows:



x(t) is the received baseband signal, *t* denotes time,
$\mathbf{y}(n)$ is a vector containing M samples of n^{th} symbol,

$\mathbf{y}(n)=[x_1(n), x_2(n), \ldots, x_M(n)]^T$,

f_s is the symbol frequency,
$s(n)$ is the transmitted n^{th} symbol,

$\mathbf{h}_m=[h_m(0), h_m(1), \ldots, h_m(L-1)]^T$,

m denotes the m^{th} channel, $m=1, 2, \ldots M$,
M=the rate of fractional sampling or the number of samples per symbol, M also denotes the number of channels in the multi-channel model,
L=the maximum of the order of the channels, $[.]^T$ denotes the vector or matrix transpose.

If the input symbol sequence is $s(n), n=1, 2, \ldots$, we can write $x_m(n)$ as,

$$x_m(n)=\sum_{i=0}^{L-1} h_m(i)s(n-i), \quad m=1, 2, \ldots, M$$

$$x_m(n) = h_m * s(n)$$

where, * denotes the convolution operator.

Let,
$\mathbf{h}_m=[h_m(0), h_m(1), \ldots, h_m(L-1)]^T$,
$\mathbf{s}(n)=[s(n), s(n-1), \ldots, s(n-(L-1))]^T$,
$\mathbf{y}(n)=[x_1(n), x_2(n), \ldots, x_M(n)]^T$,
$n=1, 2, \ldots,$.

Then, $x_m(n)=\mathbf{h}_m^T\mathbf{s}(n)$. (1)

Further, $\mathbf{y}(n)=\mathbf{H}^T\mathbf{s}(n)$ (2)

where,

$$\mathbf{H} = \begin{bmatrix} h_1(0) & h_2(0) & \ldots & h_M(0) \\ h_1(1) & h_2(1) & \ldots & h_M(1) \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ h_1(L-1) & h_2(L-1) & \ldots & h_M(L-1) \end{bmatrix}$$

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_M] \qquad (3)$$

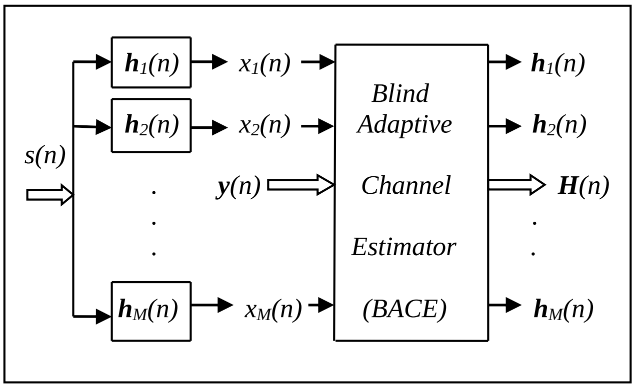
Here, $s(n)$ has to be estimated, $\mathbf{H}$ is unknown and to be estimated in order to obtain $s(n)$. The received signal $x(t)$ is known to us. We want to estimate $s(n)$ adaptively from $x(t)$. We sample $x(t)$ at the rate of M times the symbol rate f_s to obtain the vector,

$$\mathbf{y}(n)=[x_1(n), x_2(n), \ldots, x_M(n)]^T, n=1, 2, \ldots, .$$

We are going to estimate $s(n)$ from $\mathbf{y}(n)$ adaptively. In order to achieve that, we first formulate a methodology for estimating $\mathbf{H}$ adaptively.

III. A NEW APPROACH FOR BLIND ADAPTIVE CHANNEL ESTIMATION (BACE)

We now formulate an adaptive algorithm for estimating the parameters for the multi-channel system matrix $\mathbf{H}$. We want to achieve that without any training signal, and hence we call it a Blind Adaptive Channel Estimator (BACE). Let us consider the multi-channel model:



M=the number of channels or the number of samples per symbol; this has to be chosen appropriately.
L=the maximum of the order of the channels; this is unknown and has to be estimated.

We consider the estimation of M and L later. We can now write,

$$s(n)*h_i(n)=x_i(n) \qquad (4)$$
$$s(n)*h_j(n)=x_j(n) \qquad (5)$$

By convolving Equation (4) with h_j, we obtain,

$x_i(n)*h_j(n)=h_j(n)*s(n)*h_i(n)$

Substituting from Equation (5), we get,

$$x_i(n)*h_j(n)= x_j(n)*h_i(n) \quad (6)$$

Let,

$$\mathbf{X}_m(L)=\begin{bmatrix} x_m(L) & x_m(L-1) & \ldots & x_m(1) \\ x_m(L+1) & x_m(L) & \ldots & x_m(2) \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ x_m(L+(n-1)) & x_m(L+(n-2)) & \ldots & x_m(n) \end{bmatrix}$$

$$\mathbf{X}_m(L)=\begin{bmatrix} \mathbf{x}_m(L)^T \\ \mathbf{x}_m(L+1)^T \\ . \\ . \\ . \\ \mathbf{x}_m(L+(n-1))^T \end{bmatrix} \quad (7)$$

where,

$\mathbf{x}_m(L+n)=[x_m(L+n), x_m(L+n-1), ..., x_m(n+1)]^T$.

Then, Equation (6) can be written as,

$$\mathbf{x}_i(n)^T\mathbf{h}_j=\mathbf{x}_j(n)^T\mathbf{h}_i \quad (8)$$

where, $\mathbf{h}_m=[h_m(0), h_m(1), ..., h_m(L-1)]^T$.

Now, we define an objective function Ξ as

$\Xi = E [\Xi(n)]$

where, E denotes the expectation operator, and the instantaneous objective function $\Xi(n)$ is given by,

$$\Xi(n)=\frac{1}{2}\sum_{i=1}^{M-1}\sum_{j=i+1}^{M} e_{ij}^2 \quad (9)$$

where,
$e_{ij}(n)= \mathbf{x}_i(n)^T\mathbf{h}_j - \mathbf{x}_j(n)^T\mathbf{h}_i$

The important error relationship at n^{th} symbol that our blind adaptive algorithm is based on is,

$$e_{ij}(n)= \mathbf{x}_i(n)^T\mathbf{h}_j - \mathbf{x}_j(n)^T\mathbf{h}_i \quad (10)$$

From this relationship, we get,

$$\partial e_{ij}(n)/\partial \mathbf{h}_i=-\mathbf{x}_j(n) \quad (11)$$

$$\partial e_{ij}(n)/\partial \mathbf{h}_j= \mathbf{x}_i(n) \quad (12)$$

Now, we concatenate them vectors $\mathbf{x}_1(n)$, $\mathbf{x}_2(n)$, ..., $\mathbf{x}_M(n)$ to obtain the vector $\mathbf{x}(n)$,

$$\mathbf{x}(n)=\begin{bmatrix} \mathbf{x}_1(n) \\ \mathbf{x}_2(n) \\ . \\ . \\ . \\ \mathbf{x}_M(n) \end{bmatrix} \quad (13)$$

$$\mathbf{h}=\begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ . \\ . \\ . \\ \mathbf{h}_M \end{bmatrix} \quad (14)$$

$\mathbf{x}_m(n)$ and $\mathbf{h}_m$ are vectors of length L,
$\mathbf{x}(n)$ and $\mathbf{h}$ are vectors of length ML.

Since $e_{ij}(n)=\mathbf{x}_i(n)^T\mathbf{h}_j-\mathbf{x}_j(n)^T\mathbf{h}_i$,

the complement symmetry gives us,

$$e_{ij}(n)= - e_{ji}(n). \quad (15)$$

We now define the matrix $\boldsymbol{\Gamma}(n)$ so that,

$$\boldsymbol{\Gamma}(n)=\begin{bmatrix} e_{11}(n)\mathbf{I} & e_{12}(n)\mathbf{I} & \ldots & e_{1M}(n)\mathbf{I} \\ e_{21}(n)\mathbf{I} & e_{22}(n)\mathbf{I} & \ldots & e_{2M}(n)\mathbf{I} \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ e_{M1}(n)\mathbf{I} & e_{M2}(n)\mathbf{I} & \ldots & e_{MM}(n)\mathbf{I} \end{bmatrix} \quad (16)$$

where, $\mathbf{I}$ is an identity matrix of order $(L\times L)$ and $\boldsymbol{\Gamma}(n)$ is of order $(ML\times ML)$.

The matrix $\boldsymbol{\Gamma}(n)$ is used in the algorithm for channel estimation. Once the algorithm is learned, all the elements in the matrix become zero or negligibly small.

Since $e_{ii}(n)=0$ and $e_{ij}(n) = - e_{ji}(n)$, we get,

$$\boldsymbol{\Gamma}(n) = \begin{bmatrix} 0\mathbf{I} & e_{12}(n)\mathbf{I} & e_{13}(n)\mathbf{I} & \dots & e_{1M}(n)\mathbf{I} \\ -e_{12}(n)\mathbf{I} & 0\mathbf{I} & e_{23}(n)\mathbf{I} & \dots & e_{2M}(n)\mathbf{I} \\ . & . & & \dots & . \\ . & . & & \dots & . \\ . & . & & \dots & . \\ -e_{1M}(n)\mathbf{I} & -e_{2M}(n)\mathbf{I} & -e_{3M}(n)\mathbf{I} & \dots & 0\mathbf{I} \end{bmatrix} \quad (17)$$

Now, the $\frac{\partial \Xi(n)}{\partial \boldsymbol{h}}$ can be written as,

$$\frac{\partial \Xi(n)}{\partial \boldsymbol{h}} = - \boldsymbol{\Gamma}(n)\mathbf{x}(n) \quad (18)$$

We can now minimize $\Xi(n)$ using the gradient search, to obtain the parameter update relationship,

$$\mathbf{h}(n) = \mathbf{h}(n-1) - \alpha \frac{\partial \Xi(n)}{\partial \boldsymbol{h}} \quad (19)$$

where, α is the learning constant, $\mathbf{h}(n)$ is the parameter estimate after n symbols.

The Equations (18) and (19) together provide the parameter update for the learning algorithm.

The Learning Algorithm:

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \alpha \boldsymbol{\Gamma}(n-1)\mathbf{x}(n) \quad (20)$$

$$\mathbf{x}(n) = \begin{bmatrix} \mathbf{x}_1(n) \\ \mathbf{x}_2(n) \\ . \\ . \\ . \\ \mathbf{x}_M(n) \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ . \\ . \\ . \\ \mathbf{h}_M \end{bmatrix}$$

$$\boldsymbol{\Gamma}(n) = \begin{bmatrix} 0\mathbf{I} & e_{12}(n)\mathbf{I} & e_{13}(n)\mathbf{I} & \dots & e_{1M}(n)\mathbf{I} \\ -e_{12}(n)\mathbf{I} & 0\mathbf{I} & e_{23}(n)\mathbf{I} & \dots & e_{2M}(n)\mathbf{I} \\ . & . & & \dots & . \\ . & . & & \dots & . \\ . & . & & \dots & . \\ -e_{1M}(n)\mathbf{I} & -e_{2M}(n)\mathbf{I} & -e_{3M}(n)\mathbf{I} & \dots & 0\mathbf{I} \end{bmatrix}$$

$e_{ij}(n)= \mathbf{x}_i(n)^T\mathbf{h}_j(n-1)-\mathbf{x}_j(n)^T\mathbf{h}_i(n-1)$,

which is innovation at symbol n.

$\mathbf{x}_m(n)=[x_m(n), x_m(n-1), \dots, x_m(n-(L-1))]^T$,

$m=1,2, \dots, M$.
$\mathbf{h}_m$ *is a vector of order* L,
$\mathbf{h}$ *is a vector of order* ML,
$\mathbf{x}_m(n)$ *is a vector of order* L,
$\mathbf{x}(n)$ *is a vector of order* ML
$\boldsymbol{\Gamma}$ *is a matrix of order* $ML\times ML$,
$\mathbf{I}$ *is an identity matrix of order* $L\times L$.

Note that $\boldsymbol{\Gamma}(n)$ is not available at symbol n. Hence, we have replaced $\boldsymbol{\Gamma}(n)$ by the best available estimate, $\boldsymbol{\Gamma}(n-1)$, to obtain the learning algorithm given in Equation (20).

IV. ESTIMATION OF THE ORER OF THE CHANNEL AND THE RATE OF FRACTIONAL SAMPLING

The information required in estimating the order of the channel, L, and the rate of fractional sampling or the number of samples per bit, M, are inherent in the channel parameter matrix $\boldsymbol{H}$ obtained for arbitrary L and M,

$\boldsymbol{H}=[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_M]$,

where $\mathbf{h}_m$ is a vector of order L, and $\boldsymbol{H}$ is a matrix or order $L\times M$, $L\leq M$.

Initially, we choose M arbitrarily and set $L=M$. Then, we estimate the channel parameter matrix $\boldsymbol{H}$. Using singular value decomposition of matrix $\boldsymbol{H}$, we obtain,

$$\boldsymbol{H=U\Lambda V} \quad (21)$$

where, $\boldsymbol{\Lambda}$ is a diagonal matrix.

Let $\boldsymbol{\lambda}=diag(\boldsymbol{\Lambda})$
$=[\lambda_1, \lambda_2, \dots, \lambda_M]$ (22)

where, $\lambda_1\geq \lambda_2\geq\lambda_3, \dots, \geq\lambda_M$ (23)

If the order of the channel is less than the fractional sampling M, then the number of the dominant singular values will be equivalent to the order of the channel L.

***Proper value for L and M can be found in the following steps*:**

Step-1: If all the singular values of matrix $\boldsymbol{H}$ are dominant, then increase the rate of fractional sampling M and restart the learning.

Step-2: If the number of dominant singular values of matrix $\boldsymbol{H}$ is less than M, then, the proper choice of the fractional sampling rate M has been made. Then, we have,
L= effective rank of $\boldsymbol{H}$,
where, the effective rank is the number of dominant singular values $\boldsymbol{H}$.

Step-3: Once the proper values for L and M are found, restart the learning process and estimate the parameter matrix $\boldsymbol{H}$.

V. ESTIMATION OF BITS

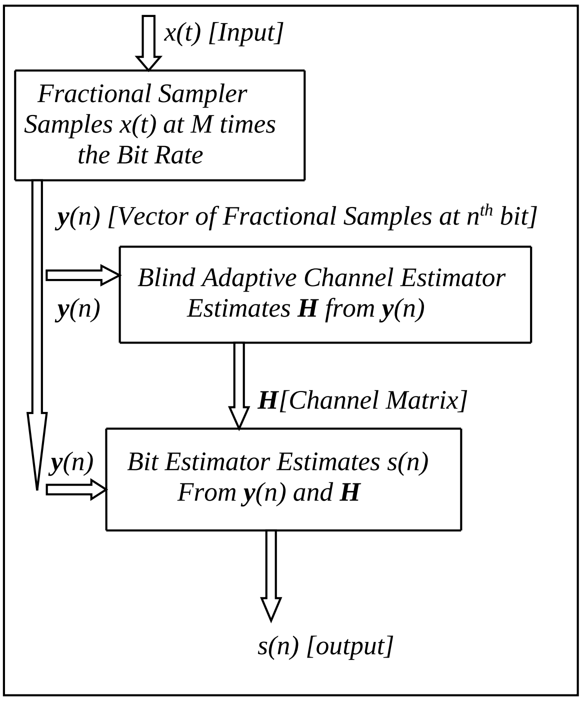
From the Equation (2), we have the single channel to multi-channel model given by,

$$\boldsymbol{y}(n)=\boldsymbol{H}^T\boldsymbol{s}(n) \qquad (24)$$

where, $\boldsymbol{h}_m=[h_m(0), h_m(1), ..., h_m(L-1)]^T$,
$\boldsymbol{s}(n)=[s(n), s(n-1), ..., s(n-(L-1))]^T$,
$\boldsymbol{y}(n)=[x_1(n), x_2(n), ..., x_M(n)]^T, n=1, 2, \ldots, .$

$\boldsymbol{H}=[\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_M]$.

The process of estimating s(n) can be illustrated as follow:



We can solve the equation (24) to obtain,

$$\boldsymbol{s}(n)=(\boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{H}\boldsymbol{y}(n) \qquad (25)$$

From the proper choice of L, the matrix $\boldsymbol{H}$ is full rank. Therefore, the solution of equation (25) is unique.
Now, let,

$\boldsymbol{P}=(\boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{H}$.

In the event the matrix $\boldsymbol{H}$ is nearly singular, pseudo-inverse is chosen in place of $(\boldsymbol{H}\boldsymbol{H}^T)^{-1}$.
Now, we can write,

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}^T \\ --- \\ \boldsymbol{Q} \end{bmatrix}$$

where, $\boldsymbol{p}^T$ is the first row of the matrix $\boldsymbol{P}$.
We can now obtain the n^{th} symbol $s(n)$,

$$s(n)=\boldsymbol{p}^T\boldsymbol{y}(n) \qquad (26)$$

In order to recover symbols accurately, it is necessary to select the rate of fractional sampling M such that, $M \geq L$. We have already found a way to achieve this in the previous section.

VI. PERFORMANCE EVALUATION

We have seen that the Blind Adaptive Equalizer performs its operations in two stages:

1. Estimation of the channel
2. Estimation of the symbols

We want to find out how well the Blind Adaptive Equalizer carryout these tasks. We want to see how good the channel estimator is, and how the performance of the channel estimator varies with the signal to noise ratio (snr).

We also want to know how often an estimated symbol differs from the transmitted symbol. In other words, we want to know the probability of error. In fact, the probability of error provides us the overall performance of the equalizer.

In the case of a data adaptive learning algorithm, it is important to know how fast the algorithm learns. In order to find out the speed of the algorithm, we also need to define the learning curves.

We know that the bias and variance are two very useful performance evaluators for any parameter estimation algorithm. Therefore, we use the bias and variance of the channel parameter estimates as performance evaluators in the channel estimation part of the Blind Adaptive Equalizer. However, we have to modify the conventional bias and variance to incorporate the multi-channel situation so that we have one consolidated bias and one consolidated variance. To achieve that, we define normalized multichannel parameter matrix $\boldsymbol{H}_{nor}$ so that the i^{th} column $(\boldsymbol{h}_i)_{nor}$ of matrix $\boldsymbol{H}_{nor}$ is given by,

$(\boldsymbol{h}_i)_{nor}=(1/h_i(0))\boldsymbol{h}_i$.

In other words, we have normalized each channel parameter vector with respect to the first element of each channel. Now, instead of $\boldsymbol{H}$, we deal with $\boldsymbol{H}_{nor}$. This will eliminate any ambiguity regarding the presence of any multiplication factor.

Let the true channel parameter matrix be $\boldsymbol{H}$, and the estimated channel parameter matrix be $est(\boldsymbol{H})$. Then, the error matrix is given by,

$$\Delta\boldsymbol{H}_{nor}= \boldsymbol{H}_{nor} - est(\boldsymbol{H}_{nor}).$$

Definition: Bias

We define bias as,

$$Bias=mean(abs(E[\Delta\boldsymbol{H}_{nor}]))$$

where, for any matrix $\boldsymbol{A}$,
$mean(\boldsymbol{A})$ is a scalar quantity where it is the mean of all the elements in matrix $\boldsymbol{A}$,
$abs(\boldsymbol{A})$ is a matrix where each element of matrix $\boldsymbol{A}$ is replaced by the absolute value of each element, and
E is the expectation operator.

Definition: Variance

We define Variance as,

$$Variance=E[mean(\Delta\boldsymbol{H}_{nor}\otimes\Delta\boldsymbol{H}_{nor})]$$

where, E denotes the expectation operator, and $\otimes$ denotes the element by element product between two matrices.

Performance Bounds:

The curves *Bias* and *Variance* against signal to noise ratio *snr* provide the performance bounds of the algorithm. These curves provide the lowest signal to noise ratio that the algorithm could be used successfully in practice.

Probability of Error:

Let us assume that we have transmitted n symbols over a channel, and out of that n_{error} number of symbols is in error. Then, the probability of error is defined as,

$$Probability\ of\ Error = E[\lim_{n\to\infty}\{n_{error}/n\}],$$

where, E denotes the expectation operator.

The *Probability of Error* versus *snr* gives us the overall performance of the communication system.

VII. THE IMPLEMENTATION OF THE BLIND ADAPTIVE EQUALIZER

Step-1: Choose M, the rate of fractional sampling for the number of samples per bit arbitrarily.

Step-2: Set $L=M$, L is the maximum length of channels in the multi-channel system.

Step-3: Initialize $\boldsymbol{h}(0)$ using a small random sequence, where,

$$\boldsymbol{h}(0) = \begin{bmatrix} \boldsymbol{h}_1(0) \\ \boldsymbol{h}_2(0) \\ \vdots \\ \boldsymbol{h}_M(0) \end{bmatrix}$$

Step-4: Update $\boldsymbol{h}$ using the algorithm in Equation (20),

$$\boldsymbol{h}(n) = \boldsymbol{h}(n-1) + \alpha\,\boldsymbol{\Gamma}(n-1)\boldsymbol{x}(n)$$

where α is a small value.
We choose α=0.01.

$$\boldsymbol{h}(n) = \begin{bmatrix} \boldsymbol{h}_1(n) \\ \boldsymbol{h}_2(n) \\ \vdots \\ \boldsymbol{h}_M(n) \end{bmatrix}$$

$\boldsymbol{h}_m(n)$ is the estimate of $\boldsymbol{h}_m$ after m symbols.

$$\boldsymbol{\Gamma}(n) = \begin{bmatrix} e_{11}(n)\boldsymbol{I} & e_{12}(n)\boldsymbol{I} & \cdots & e_{1M}(n)\boldsymbol{I} \\ e_{21}(n)\boldsymbol{I} & e_{22}(n)\boldsymbol{I} & \cdots & e_{2M}(n)\boldsymbol{I} \\ \vdots & \vdots & \cdots & \vdots \\ e_{M1}(n)\boldsymbol{I} & e_{M2}(n)\boldsymbol{I} & \cdots & e_{MM}(n)\boldsymbol{I} \end{bmatrix}$$

$\boldsymbol{I}$ is an identity matrix of order $L\times L$, and $\boldsymbol{\Gamma}$ is of order $LM\times LM$,

$$e_{ij}(n)= \boldsymbol{x}_i(n)^T\boldsymbol{h}_j(n-1) - \boldsymbol{x}_j(n)^T\boldsymbol{h}_i(n-1),$$

which is innovation at symbol n,

$$\boldsymbol{x}(n)= \begin{bmatrix} \boldsymbol{x}_1(n) \\ \boldsymbol{x}_2(n) \\ \vdots \\ \boldsymbol{x}_M(n) \end{bmatrix}$$

$\boldsymbol{x}_m(n)=[x_m(n), x_m(n-1), ..., x_m(n-(L-1))]^T$,
$m=1,2, ..., M$.

Step-5: Obtain the matrix $\boldsymbol{H}$, where
$\boldsymbol{H}=[\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_M]$.

Step-6: Evaluate the singular values of $\boldsymbol{H}$.

Step-7: If the $rank(\boldsymbol{H})=M$, then, choose a higher rate and restart the algorithm from Step-2.

Step-8: If the $rank(\boldsymbol{H})<M$, then set $L=rank(\boldsymbol{H})$ and restart learning from Step-2.

Step-9: Estimate symbol $s(n)$ using the relationships,

$$s(n)=\boldsymbol{p}^T\boldsymbol{y}(n)$$

$$\boldsymbol{P}=(\boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{H}$$

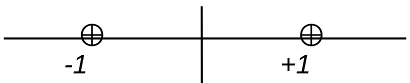
$$\boldsymbol{P}=\begin{bmatrix} \boldsymbol{p}^T \\ --- \\ \boldsymbol{Q} \end{bmatrix}$$

From the choice of L, the matrix $\boldsymbol{H}$ is full rank. Further, $L \leq M$. Therefore, the solution is unique. In the event $\boldsymbol{H}$ in nearly singular, $\boldsymbol{P}$ is chosen as the pseudo-inverse of $\boldsymbol{H}^T$.

Note: The algorithm can readily be extendable into complex space.

VIII. SIMULATION RESULTS

In order to test the performance of the algorithm, we use a binary signal constellation given below:



(Signal Constellation)

The Symbols are chosen randomly from the constellation for transmission.

For the purpose of simulation, we choose $\boldsymbol{H}$ given by,

$$\boldsymbol{H}=\begin{bmatrix} 1 & 0.2 & 0.6 & 0.2 & -0.3 & -0.9 \\ 0.5 & 0.7 & 0.1 & -0.1 & 0.5 & 0.6 \\ 0.3 & 0.1 & 0.5 & 0.2 & 0.9 & -0.5 \end{bmatrix}$$

Now that we have both input and the channel parameter matrix, we can obtain the channel output. We add appropriate amount of random Gaussian noise to obtain a noise corrupted channel output for a given signal to noise ratio, snr.

We implement the algorithm and repeat it for *100* independent realizations to obtain *Bias, Variance, Learning Curves, and the Probability of Error*. We use *2000* symbols in estimating $\boldsymbol{H}$. The learning constant $\alpha=0.01$.

DIAGRAM-1: Learning Curves, Bias, and Variance

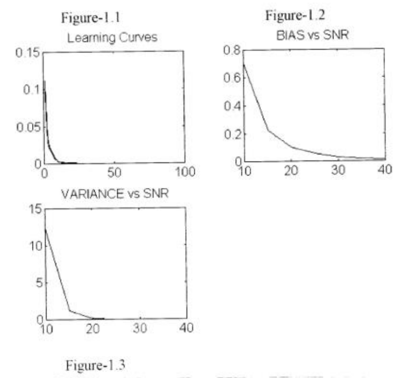
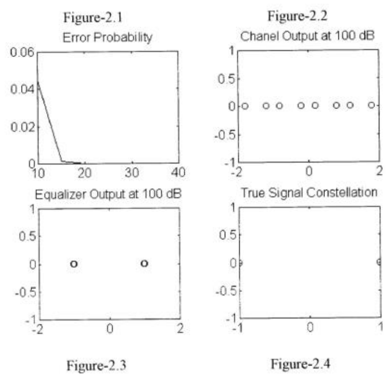


DIAGRAM-2: *Error Probability* Versus *snr*, Signal Constellation



A. Learning Curves, Bias and Variance

The simulation results for the Learning Curves, the Bias and the Variance are given in the Diagram-1.

Figure-1.1: Learning Curves

This shows the learning curves for the Blind Adaptive Equalizer. They are obtained by averaging 100 independent realizations. We have shown learning curves for snr=10, 15, 20, 25, 30, 35, and 40 dB. Since all the curves are overlapping, it appears that the learning curves are independent of the signal to noise ratio (snr) for snr≥10. This indicates that the speed of the learning algorithm is independent of the signal to noise ratio for higher snr values. The algorithm learns in as few as 20 symbols.

Figure-1.2: Bias vs. snr

The bias against signal to noise ratio (snr) is shown. As expected, bias decreases as the signal to noise ratio increases.

Figure-1.3: Variance vs. snr

The variance against signal to noise ratio is shown. It is clear that for snr>20 dB, the algorithm performs with a negligible variance. The Blind Adaptive

Equalizer estimates parameters of the channel reliably at high signal to noise ratio.

B. Error Probability and Signal constellation

The simulation results for the error probabilities are given in the Diagram-2.

Figure-2.1: Probability of Error

The error probability against signal to noise ratio (snr) is plotted. The error probabilities are calculated using 10^6 transmitted symbols. The error probability provides us the overall performance of the digital communication system. As we increase the signal to noise ratio beyond 20 dB, the algorithm performs with zero probability of error.

Figure-2.2: Channel Output
(Signal Constellation) for snr=100 dB

We have shown here the channel output for almost no noise environment, snr=100 dB. Since we have considered 2-point signal constellation and a channel of length 3, that is a channel containing three parameters, the number of output levels would be 2^3, which is evident from the Figure.

The effect of the channel on the input constellation is very evident from the Figure. It shows what exactly channel does to an input signal. The channel has dispersed the input signal. At the receiver, what we have is this dispersed signal plus noise. Our effort is to recover the true input from the dispersed signal received at the receiver. The results we obtained using the algorithm presented here is shown in Figure-2.3: Blind Adaptive Equalizer Output for snr=100 dB

This illustrates the recovered transmitted signal or the output of the adaptive equalizer. What is shown here is the output of the Blind Adaptive Equalizer before the slicer In fact, the algorithm recovered the transmitted symbol accurately. Since it is evident here that the Blind Adaptive Equalizer recovers signal accurately for almost no noise environment, we can now consider the performance under noisy conditions.

Figure-2.4: Input Signal Constellation

Here, we have shown the input signal constellation for comparison.

C. Signal Constellation

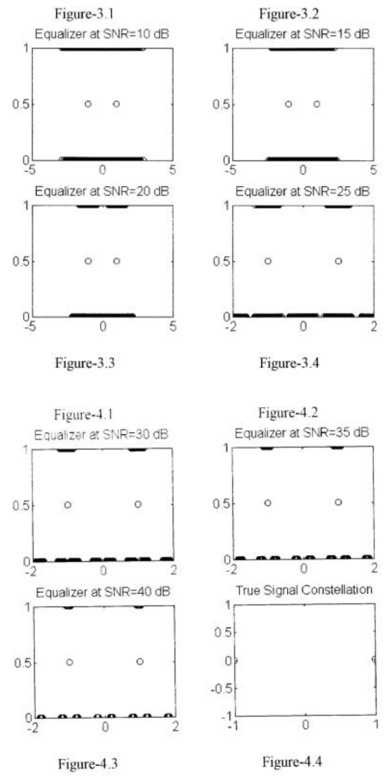
The simulation results for the signal constellation are given in the Diagram-3. Here, we present the following for different signal to noise ratios (snr):

- *s(n)*, input signal
- *x(n)*, output of the channel for the received baseband signal
- *est(s(n))*, estimated symbols (before the slicer).

In each Figure in Diagram-3:

- The lower level represents *x(n)*
- The middle level represents *s(n)*
- The upper level represents *est(s(n))*

DIAGRAM-3: Signal Constellation



Figures 3.1-4.3 shows *s(n), x(n)*, and *est(s(n))* for different signal to noise ratio (snr) values. Figure-4.4 illustrates the true input signal constellation *s(k)*.

From Figure-3.1, it is clear that the Blind Adaptive Equalizer was not able to estimate the symbols accurately at snr=10 dB. However, as we increase the snr, the Blind Adaptive Equalizer starts to distinguish symbols accurately.

As we see from Figure-3.3, the signal constellation of the output of the channel is completely overlapping at snr=20 dB. However, the adaptive Blind Equalizer clearly brings out two clusters representing binary signal constellation in its output.

At higher snr values, snr≥20 dB, the Blind Adaptive Equalizer recovers the transmitted symbols with no error.

D. Estimation of L and M

In order to estimate the length of the channel, *L* and the rate of fractional sampling *M*, we have to obtain the singular value vector $\boldsymbol{\lambda}$ of the estimated channel parameter matrix *est(**H**))* for arbitrary *M*.

Let the normalized singular value vector $\boldsymbol{\lambda}_{nor}$ be,

$\boldsymbol{\lambda}_{nor}=[1, \lambda_2/\lambda_1, \lambda_3/\lambda_1, \ldots, \lambda_M/\lambda_1]$,
where, $\boldsymbol{\lambda}=[\lambda_1, \lambda_2, \ldots, \lambda_M]$.

The normalized singular vector $\boldsymbol{\lambda}_{nor}$(snr) for different snr values are given below:

$\boldsymbol{\lambda}_{nor}$(10)=[1, 0.7641, 0.1606, 0.0490, 0.0197, 0.0002]
$\boldsymbol{\lambda}_{nor}$(15)=[1, 0.6911, 0.2516, 0.0138, 0.0077, 0.0016]
$\boldsymbol{\lambda}_{nor}$(20)=[1, 0.6929, 0.0661, 0.0088, 0.0050, 0.0013]
$\boldsymbol{\lambda}_{nor}$(25)=[1, 0.5892, 0.0963, 0.0055, 0.0032, 0.0001]
$\boldsymbol{\lambda}_{nor}$(30)=[1, 0.5177, 0.1620, 0.0038, 0.0023, 0.0008]
$\boldsymbol{\lambda}_{nor}$(35)=[1, 0.4408, 0.2914, 0.0144, 0.0013, 0.0001]
$\boldsymbol{\lambda}_{nor}$(40)=[1, 0.7175, 0.2268, 0.0070, 0.0018, 0.0000]
$\boldsymbol{\lambda}_{nor}$(100)=[1, 0.5067, 0.4703, 0.0000, 0.0000, 0.0000]

The number of dominant singular values is equal to the true order of the model, *L*. This also gives us a clue in determining *M*, the fractional sampling required. If the singular values of the matrix *est(**H**)* do not tail off, we have to increase the fractional sampling until we see the tailing off on the singular values of the *est(**H**).*

Therefore, singular values of *est(**H**)* not only help us to determine the order of the channel *L*, but also in choosing the proper rate of fractional sampling *M*.

IX. CONCLUSIONS

In a communication system, modem has to be initialized before the actual data transfer takes place. During the initialization period, the system will estimate the channel so that the unwanted effects of the channel can be taken out so that the transmitted symbols can be estimated while removing the inter symbol interference brought by the channel. The estimation of the channel is usually achieved adaptively by transmitting a training signal that is known to the receiver. The Blind Adaptive Equalizer achieves the estimation of the channel and the transmitted information without any training signal. The estimation of the channel and the estimation of the transmitted symbols are done using the transmitted information itself adaptively.

The Blind Adaptive Equalizer is based on the conversion of the single channel into a multichannel system by using the fractional sampling. The multi-channel parameter matrix can be used to estimate the order of the channel, to determine the fractional sampling required, and to estimate the information transmitted. Its data adaptive nature allows it to be used in time varying channels. Since it does not require a training signal that is known to the receiver, it can be used in the signal interception environment.

The Blind Adaptive Equalizer's learning characteristic indicates that it is a fast learner and the speed of the learning is independent of the signal to noise ratio (snr) for higher snr values. At higher snr values, the Blind Adaptive Equalizer is capable of eliminating the inter-symbol interference providing the estimate of the symbol with zero probability of error. The Blind Adaptive Equalizer provides an elegant and speedy learning mechanism for Modems in digital communication systems without the need of a training signal.

REFERENCES

[1] Xu G., H. Liu, L. Tong and T. Kailath, "*A Least Square Approach for Blind Channel Identification*", IEEE Transaction of Signal Processing, vol. 43, No. 12, December 1995.