8-1-1994

Blind equalization

Daniel Diguele

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation


This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Abstract

An equalizer is an adaptive filter that compensates for the non-ideal characteristics of a communication channel by processing the received signal. The adaptive algorithm searches for the inverse impulse response of the channel, and it requires knowledge of a training sequence, in order to generate an error signal necessary for the adaptive process. There are practical situations where it would be highly desirable to achieve complete adaptation without the use of a training sequence, hence the term “blind”. Examples of these situations are multipoint data networks, high-capacity line-of-sight digital radio, and reflection seismology. A blind adaptive algorithm has been developed, based on simplified equalization criteria. These criteria are that the second- and fourth-order moments of the input and output sequences are equalized. The algorithm is entirely driven by statistics, only requiring knowledge of the variance of the input signal. Because of the insensitivity of higher-order statistics to Gaussian processes, the algorithm performs well when additive white Gaussian noise is present in the channel. Simulations are presented in which the new blind equalizer developed is compared to other equalization algorithms.
Acknowledgements

I would like to express my appreciation of Dr. S. Dianat, my Thesis Advisor, who oversaw this project from the beginning. Without his keen insight and guidance I would have never finished this work, and I am truly grateful for his making time for me in his very busy schedule. I would also like to thank Dr. M. R. Raghuveer for the opportunity he gave me to start studying communications channels and adaptive equalization under his guidance. Thanks also to Janel Brennan for her encouragement and for the editing job.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iv</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Chapter 2: Adaptive Equalization</td>
<td>4</td>
</tr>
<tr>
<td>Intersymbol Interference</td>
<td>5</td>
</tr>
<tr>
<td>Linear Transversal Equalizers</td>
<td>9</td>
</tr>
<tr>
<td>Zero-forcing criterion</td>
<td>10</td>
</tr>
<tr>
<td>Mean-Square-Error criterion</td>
<td>12</td>
</tr>
<tr>
<td>The LMS algorithm</td>
<td>15</td>
</tr>
<tr>
<td>Decision-directed equalization</td>
<td>18</td>
</tr>
<tr>
<td>Decision-Feedback Equalizers</td>
<td>19</td>
</tr>
<tr>
<td>Chapter 3: Higher-Order Statistics</td>
<td>21</td>
</tr>
<tr>
<td>Definitions:</td>
<td></td>
</tr>
<tr>
<td>Moments and cumulants of stochastic signals</td>
<td>22</td>
</tr>
<tr>
<td>Moments and cumulants of stochastic processes</td>
<td>24</td>
</tr>
<tr>
<td>Properties of moments and cumulants</td>
<td>28</td>
</tr>
<tr>
<td>Equalization with higher-order statistics</td>
<td>30</td>
</tr>
<tr>
<td>The Bussgang deconvolution techniques</td>
<td>31</td>
</tr>
<tr>
<td>Special cases of the Bussgang algorithm:</td>
<td></td>
</tr>
<tr>
<td>The Sato algorithm</td>
<td>32</td>
</tr>
<tr>
<td>The Godard algorithm</td>
<td>34</td>
</tr>
</tbody>
</table>
A simplified set of conditions for equalization .......... 36
Constrained criterion ....................................... 38
Unconstrained criteria ...................................... 42

**Chapter 4**: Blind equalization .................................................. 46

**Chapter 5**: Simulations and results ........................................... 55
Case 1: One-pole channel ....................................................... 57
Case 2: One-pole channel with additive colored noise ...... 63
Case 3: One-pole time-varying channel ................................. 66
Case 4: Two-pole channel ...................................................... 69
Case 5: Three-pole channel .................................................... 78
Case 6: Three-pole, two-zero channel ................................. 82

**Conclusions** ......................................................................... 86

**References** ........................................................................... 89

**Appendix** ............................................................................. 92
Chapter 1

Introduction

Considerable effort has been devoted in the past three decades to the study of data-transmission systems which make efficient use of the available power and channel bandwidth. Increasingly, we rely on computer communications for transmission of vast amounts of data. The need for high-speed data transmissions over analog telephone channels has primarily been met by the appearance of fast modems which carry digital data over these voice-bandwidth channels.

Analog channels deliver distorted versions of their input signals. The transmission of digital data over such channels is limited by the non-ideal transformations performed by the channel on the signals being transmitted. For bandwidth-limited channels (such as voice-grade telephone channels), the chief determining factor in the design of high-speed transmission systems is Intersymbol Interference (ISI), which is caused by time-dispersion in the transmit filter, the transmission medium, and the receive filter. Other limiting factors on the channel performance are the possible additions of background thermal noise, impulse noise, and channel fading.

Equalization dates back to the use of loading coils to improve the characteristics of twisted-pair telephone cables. An equalizer compensates for the non-ideal characteristics of a communications channel by processing the
received signal. In most cases, the channel characteristics are not known beforehand, so the equalizer in fact consists of an adaptive filter that searches for the inverse impulse response of the channel. An adaptive filtering algorithm requires knowledge of a training sequence in order to form an error signal necessary for the adaptive process. Since the transmitter and the receiver are usually physically separated, there are two common ways to generate a replica of the desired response at the receiver. One such way is by using a training sequence known to both the transmitter and the receiver. A decision-directed method can also be used which does not require a training sequence, provided a good replica of the transmitted sequence is being produced at the output of a decision device at the receiver.

There are practical situations where it would be highly desirable to achieve complete adaptation without the use of a training sequence. In this context, blind deconvolution algorithms have received much attention recently. Blind equalizers are adaptive filtering algorithms designed such that they do not need an externally-supplied desired response to generate an error signal. Instead, an estimate of the training sequence is generated by applying a non-linear transformation on the data sequences passing through the channel.

One example of a system in which blind equalization can be very helpful is in a multipoint data network, where common problems involve severe variations in channel characteristics, or simply that a receiver was not powered on during initial synchronization of the network. In a heavily loaded large multipoint network, data throughput is increased and the burden of monitoring the network is eased if some form of blind equalization is built into the receiver design [9]. Other examples include high-capacity line-of-sight digital
radio, where multipath fading is a big problem; and reflection seismology, where the traditional method of linear-predictive deconvolution ignores valuable phase information contained in the reflection seismogram, unlike the blind deconvolution method \[^{10}\].

In many practical situations the channel impulse response many not be minimum-phase (i.e. not all poles and zeros are inside the unit circle). Examples of such non-minimum phase systems include telephone channels and fading radio channels. Equalization of such channels requires the identification of both the magnitude and the phase of the system's transfer function. The magnitude can be identified using second-order statistics of the output signal. The phase information, however, is more complicated to extract, and it involves the calculation of higher-order statistics \[^{24}\].

The purpose of this work is to develop a new blind deconvolution algorithm well suited for problems where tracking of higher-order statistical variations is needed. The convergence behavior of the algorithm will be investigated, and its performance will be compared to other well-known adaptive equalization schemes.

Chapter 2 provides a complete review of the field of adaptive equalization. Chapter 3 introduces higher-order statistics and some of the concepts behind their use in equalization. In Chapter 4, the development of the new blind equalization algorithm is shown. Chapter 5 presents the results of using this new algorithm with a variety of communications channels, and compares these results to those obtained with other algorithms. Finally, the conclusion and suggestions for further study can be found at the end of Chapter 5.
Chapter 2

Adaptive Equalization

An equalizer compensates for the non-ideal characteristics of a communication channel. The term equalization is synonymous with deconvolution and inverse modeling. The waveform of the transmitted signal arrives at the receiver convolved with the impulse response of the channel (Fig. 1), hence the use of the term deconvolution, to express the operation of restoring the original signal. In order to deconvolve a channel, it is necessary to model the inverse of its frequency response, wherefrom the term inverse modeling is derived.

The inverse model of an unknown system is, itself, a system with a frequency response approximating as much as possible the reciprocal of the unknown frequency response. In the case of communications channels, time dispersion is the most common type of distortion, giving rise to Intersymbol Interference (ISI). A dispersive channel is one in which signals at different frequencies travel with different velocities, or different group delays. Other types of transformations performed by the channel are frequency translation and nonlinear or harmonic distortion. Also, corruption of the input waveform (usually statistical) may be additive and/or multiplicative, and due to various
factors such as background thermal noise processes, impulsive noise and channel fading.

![Diagram](image)

**Figure 1.** Data transmission system

**Intersymbol Interference (ISI)**

Intersymbol interference appears in all pulse-modulation systems: frequency-shift keying (FSK), phase-shift keying (PSK), quadrature amplitude modulation (QAM) and pulse-amplitude modulation (PAM).

Figure 1 represents a generalized equivalent model of a digital communication system. For the sake of simplicity, it will be assumed that the "channel" includes the effects of the transmitter filter, the modulator, the actual transmission channel, and the demodulator (see Fig. 2). The output signal $y(t)$ is the superposition of the impulse response of the channel $h(t)$ to each symbol in the input sequence $x(n)$ (i.e. convolution) plus additive white Gaussian noise $N(t)$:
\[ y(t) = x(t)^\text{transmit} - kTs + N(t) \] (1)

where:

- \( Ts \) is the signaling interval (seconds), and thus \( \frac{1}{Ts} \) is the data transmission rate (symbols/second).

Sampling the output \( y(t) \) every \( Ts \) seconds:

\[ y(nTs) = x(k)h(nTs - kTs) + N(nTs) \] (2)

Making the substitution \( nTs = n \), the following expression in the discrete-time domain is obtained:

\[ y(n) = x(n) + x(k)h(n - k) + N(n) \] (3)

The summation term in equation 3 is the interference from neighboring symbols (ISI), and it can be seen to consist of past samples of the input data sequence \( x(n) \), weighted by samples of the channel impulse response \( h(t) \). The first term is the desired signal \( x(n) \), and the last is the additive white Gaussian noise \( N(n) \).
The ISI is zero if and only if the channel impulse response has zero crossings at $T_s$-spaced intervals, that is, if $h[(n-k)T_s]=0$ for $k\neq n$ in equation 3. When the impulse response has such uniformly-spaced zero crossings (see Figure 3 below), it satisfies Nyquist's first criterion, namely that the channel have no response beyond twice the Nyquist bandwidth $f_n=1/2T_s$. In channels which exhibit time dispersion (and thus ISI) the zero crossings in the impulse response have moved so that they do not occur at regular $T_s$-spaced intervals.

![Figure 3](image)

(a) Impulse response $h(t)$

(b) Frequency response $H(z)$

**Figure 3.** Characteristics of a channel with no ISI.
The effect of Intersymbol Interference can be seen in practice from a trace of the received signal, on an oscilloscope with its time base synchronized to the symbol rate. Such a trace is called an "eye pattern". Figure 4 shows an eye pattern for a binary PAM system with no ISI. If a channel satisfies the zero ISI condition, the "eye" is then fully open and there are only two distinct levels (for the binary PAM case at hand) at the sampling time. The peak distortion (also shown in Fig. 4) is the ISI that occurs when the data pattern is such that all the intersymbol interference terms in Eq. 3 add up to produce the maximum deviation from the desired signal at the sampling time.

![Figure 4. Outline of a binary eye pattern.](image)

The purpose of an equalizer is to minimize the ISI, which would also minimize the probability of an incorrect decision at the receiver. Without the equalizer, the eye pattern would not be open as in Fig. 4, but instead it would show a wide disparity in the positive and negative sinc pulses, indicating the presence of distortion in the sinc pulses, which is associated with ISI.
Linear Transversal Equalizers

The most commonly used filter structure in channel equalization is the transversal filter, also known as tapped-delay line filter, shown in Fig. 5. In such a filter the current and past values of the received signal $y(n)$ are linearly weighted by the equalizer coefficients $w_k$, and then summed to produce the output $z(n)$.

![Transversal Filter Diagram]

**Figure 5.** Transversal equalizer.

The equalizer output is given by:

$$ z(n) = \sum_{k=0}^{M-1} w_k y(n - k) $$  \hspace{1cm} (4)

where:

$M := \text{Number of equalizer weights (taps)}.$

$w_k := \text{Equalizer weights}.$
Eq. 4 is called a finite convolution sum, because it convolves the finite impulse response of the equalizer filter \( \{w_k\} \) with the filter input \( \{y(n)\} \) to produce the filter output \( \{z(n)\} \).

Zero-forcing criterion

If the equalizer coefficients \( w_k, k=0,1,\ldots,M-1 \) are chosen to force the samples of the combined channel and equalizer impulse response to zero at all but one of the \( M \) \( T_s \)-spaced instants in the space of the equalizer, then we call such an equalizer a zero-forcing (ZF) equalizer.

Let \( S_n \) be the convolution sum of \( h_n \) and \( w_n \) (see Fig. 7), that is, the impulse response of the combined channel plus equalizer impulse responses:

\[
S_n = \sum_{k=0}^{\infty} w_k \delta(n-kT_s)
\]

The equalizer \( w_k \) is assumed to have an infinite number of taps. Its output at the \( n^{th} \) sampling instant is
\[ z(n) = s_0x(n) + x(k) sn_ji + \sum_{k=-\infty}^{\infty} w_k^N(n - k) \]

The first term in Eq. 6 represents a scaled version of the input (which is the desired symbol at the output). The second term is the intersymbol interference, and the third is the contribution of the additive white Gaussian noise. The peak distortion, which is the peak value of the second term in (6) is:

\[ D = \max_{n} |s_n| \]

Hence the peak distortion \( D \) is a function of the equalizer weights \( w_k \). If the equalizer has an infinite number of taps, it is possible to choose the tap weights such that \( D = 0 \). This would mean that the combined channel plus equalizer impulse response \( S_n = 0 \) for all \( n \) except \( n = 0 \). That is, the intersymbol interference can be completely eliminated. The values of the equalizer weights that would achieve this are determined by the condition:

\[ \sum_{j=0}^{\infty} j w_k = 0 \quad \text{for} \quad n = 0 \]

\[ \sum_{j=0}^{\infty} j w_k = s_n \quad \text{for} \quad n \neq 0 \]
Taking the z-transform of Eq. 9:

\[ S(z) = W(z)H(z) = 1 \]

Solving for \( W(z) \):

\[ W(z) = \frac{1}{H(z)} \]

So complete elimination of the intersymbol interference requires the use of an inverse filter to the channel \( H(z) \). Such an inverse filter is called a zero-forcing filter.

Notice, however, that such perfect inverse filter requires the use of an infinite number of tap weights, which is not feasible in practice.

Mean-Square-Error (MSE) criterion

In this criterion, the equalizer weight coefficients \( w_k \) are adjusted to minimize the mean-square value of the error, defined as:

\[ e(n) = x(n) - z(n) \]

Where \( x(n) \) is the input (desired signal) to the communications system, and \( z(n) \) is the output of the equalizer (Fig. 7).

Let us define the cost function \( J \) to be minimized for the MSE criterion as:

\[ J = \{|e(n)|^2\} \]

Where \( E{} \) stands for the expectation operation. By substituting (12) into (13):

\[ J = E|x(n) - z(n)|^2 \]

and (4), modified for an equalizer with an infinite number of taps, into (14):
So the cost function $J$ is a quadratic function of the equalizer coefficients \{w_k\}. This function can be easily minimized with respect to the \{w_k\} to yield an infinite set of linear equations for the \{w_k\} coefficients $^{20}$.

Another way to obtain the set of linear equations for the equalizer coefficients \{w_k\} is to invoke the orthogonality principle in mean-square estimation, i.e. to select the weights \{w_k\} such that the error $e(n)$ is orthogonal to the input signal sequence to the equalizer \{y(n)\}$^{20}$:

\[
E\{e(n)y^*(n-l)\} = 0 \quad -\infty < l < \infty \tag{16}
\]

where the $y^*$ denotes the conjugate of the sequence \{y(n)\}, in case the data is complex-valued. Substituting $e(n)$ into (16):

\[
E\left\{\left(x(n) - \sum_{k=-\infty}^{\infty} w_k y(n-k)\right)y^*(n-l)\right\} = 0 \tag{17}
\]

Taking the summation and the equalizer coefficients out of the expectation:

\[
\sum_{k=-\infty}^{\infty} w_k E\{y(n-k)y^*(n-l)\} = 0 = E\{x(n)y^*(n-l)\} \quad \text{for} \quad -\infty < l < \infty \tag{18}
\]

In order to evaluate the expectations in (18), an expression is needed for $y(n)$. Suppose the channel has a frequency response $H(z)$, where $H(z)$ is a polynomial of degree $L$. Thus:

\[
y(n) = \sum_{j=0}^{L} h_j x(n-j) + N(n) \tag{19}
\]

where (refer to Fig. 7):
y(n) := Data sequence going into the equalizer.

h_j := Impulse response of the communications channel.

x(n) := Input data sequence into the channel.

N(n) := Additive white Gaussian noise sequence.

Substituting (19) into (18), we obtain [20]:

\[
E\{y(n-k)y^*(n-l)\} = \sum_{j=0}^{L} h_j^* h_{j+l-k} + N_0 \delta_{lk}
\]

\[
= \begin{cases}
  r_{hh}(l-k) + N_0 \delta_{lk} & \text{for } |l-k| \leq L \\
  0 & \text{otherwise}
\end{cases}
\]  

(20)

and,

\[
E\{x(n)y^*(n-l)\} = \begin{cases}
  h_n^* & \text{for } -L \leq l \leq 0 \\
  0 & \text{otherwise}
\end{cases}
\]  

(21)

where:

\[
\delta_{lk} := \begin{cases}
  1, & l = k = 0 \\
  0, & l, k \neq 0
\end{cases} \text{ (Kronecker delta)}
\]

N_0 := Spectral density of the additive white Gaussian noise.

r_{hh} := Autocorrelation function of the channel impulse response

\[
= E\{h^*(n)h(n+k)\} = \sum_{k=0}^{L-n} h_k^* h_{k+n} \quad k = 0, 1, \ldots, L
\]  

(22)

L := Number of roots of the polynomial H(z), the frequency response of the channel.

Substituting Eqs. (20) and (21) into (19), and taking the z-transform:

\[
W(z)\left[H(z)H^*\left(z^{-1}\right) + N_0\right] = H^*\left(z^{-1}\right)
\]  

(23)

And solving for W(z) we obtain the transfer function of the equalizer based on the MSE criterion:
\[ W(z) = \frac{H^*(z^{-1})}{H(z)H^*(z^{-1}) + N_0} \]  

(24)

Equalizers built using the MSE criterion are more robust than ZF equalizers, because the equalizer coefficients are chosen to minimize the mean-square-error (the sum of the squares of all the ISI terms plus the noise power at the output of the equalizer). As we saw, the ZF criterion neglects the effect of noise. A finite-length ZF equalizer will minimize the peak distortion (worst case ISI) only if the peak distortion before equalization is less than 100%, i.e. if the binary eye pattern is initially open \(^{[14]}\). This condition is often not met, especially at high speeds on bad channels, which is why most current high-speed voice-band modems use MSE equalizers.

**The Least-Mean-Square (LMS) algorithm**

The most common equalizer (Fig. 8) update method involves updating each weight coefficient every time a symbol comes through the communication system. What makes this possible is that the MSE is a quadratic function of the equalizer coefficients \( \{w_k\} \), as seen in the previous section.

In order to minimize the error signal (Eq. 12), a cost function was defined (Eq. 13) which represented the mean-square value (energy signal) of the error. Why choose the square of the error as the objective function? The fourth power of the error would have been just as valid. The reason is that we can use a gradient-search algorithm to find the lowest point in the bowl-shaped quadratic performance surface, and to find the gradient the derivative has to be taken. Using a quadratic, differentiation leads to linear equations, which are easy to
solve, especially when compared to the non-linear higher-order equations we would obtain otherwise.

Additive Gaussian noise \( N(n) \) Input \( x(k) \), Channel \( H(z) \)

Desired signal \( x(n) \)

Figure 8. LMS equalizer.

Thus, finding the optimum equalizer weights involves finding the point along the performance surface where the gradient is zero. Let the vector of equalizer weights be defined as:

\[
W = [w_0, w_1, w_2] \tag{25}
\]

The cost function \( J = E[e(n)^2] \) could be approximated by time-averaging, but that method is too time and memory consuming. Instead, a coarse estimate will be used, simply replacing the expectation operation by its current realization. Hence at the \( k+1 \) realization:

\[
J_k = x(k) - z(k) \tag{26}
\]
But $z(k)$ is the output of the equalizer, and it can be expressed in vector form as:

$$z(k) = Y^T(k)W(k)$$  \hspace{1cm} (27)

where:

$$Y^T(k) = [y(k) \ y(k-1) \ y(k-2) \ldots \ y(k-L+1)]$$  \hspace{1cm} (28)

and

$$W(k) = [w_0(k) \ w_1(k) \ w_2(k) \ldots \ w_{L-1}(k)]^T$$  \hspace{1cm} (29)

$L := \text{Number of equalizer tap weights.}$

So substituting (27) into (26):

$$J_k = |x(k) - Y^T(k)W(k)|^2$$  \hspace{1cm} (30)

Then the gradient can be expressed as:

$$\nabla_k J = \frac{\partial \left[ e^2(k) \right]}{\partial W(k)}$$  \hspace{1cm} (31)

$$= 2e(k) \frac{\partial e(k)}{\partial W(k)} = 2e(k) \frac{\partial \left[ x(k) - Y^T(k)W(k) \right]}{\partial W(k)}$$

Thus:

$$\nabla_k J = -2e(k)Y^T(k)$$  \hspace{1cm} (32)

To initialize the algorithm, start with an initial guess $W_0$, and move in the direction of decreasing gradient magnitude. The search algorithm has the general form:

$$W(k+1) = W(k) - \mu \nabla_k J$$  \hspace{1cm} (33)

where:

$$\mu := \text{Step size.}$$

$$\nabla_k J := \text{Gradient at the } k\text{th iteration.}$$
And so the LMS algorithm is:

\[ W(k + 1) = W(k) + 2\mu e(k)Y^T(k) \]  

The LMS algorithm is very important and widely used because of its simplicity and ease of computation. It does not require off-line gradient estimations or repetitions of data. However, at every step during the equalization process, the desired response (usually in the form of a training sequence) must be known in order to generate the error signal \( e(k) \) in (34).

**Decision-directed equalization**

Another type of equalization uses an estimate of the error signal different from that of the LMS algorithm. Instead of using a training sequence to generate an error signal, it uses an estimate of the input.

The algorithm used is the same as the LMS algorithm, but the error signal differs as follows (Fig. 9):

\[ e(k) = \bar{x}(k) - z(k) \]  

**Figure 9.** Decision-directed equalizer.
The error signal is derived from the receiver estimate, which is not necessarily correct. For this reason, it is best to use this type of equalizer with low-noise, low-distortion channels. But the most common use of the decision-directed equalizer is after the channel has already been equalized (i.e. the eye pattern is already open) using another, more robust, algorithm. Then, this equalizer can track slow variations in the channel characteristics.

**Decision-Feedback (DFE) equalizers**

This type of equalizer is very useful for channels with severe amplitude distortion. It has a nonlinear structure that uses decision feedback to cancel the interference from symbols which have already been detected. Assuming past detected values to be correct, the ISI contributed by these symbols can be canceled exactly by subtracting an appropriately weighted version of these symbols from the output of the equalizer.

![Diagram of Decision-Feedback (DFE) equalizer](image_url)

**Figure 10.** Decision-Feedback (DFE) equalizer.
The DFE equalizer consists of two sections (see Fig. 9): a feedforward section, which is like the linear transversal equalizer discussed earlier; and a feedback section, which is the one used to remove that part of ISI caused by the past detected symbols from the equalizer output. The equalizer output can be expressed as:

\[ z(k) = Y_T(k)W_F(k) + \tilde{X}_T(k)W_B(k) \]  

(36)

where the equalizer taps are updated as follows:

\[ W_F(k+1) = W_F(k) + \mu e(k)Y_T(k) \]  

(37)

and

\[ W_B(k+1) = W_B(k) + \mu e(k)\tilde{X}_T(k) \]  

(38)
Chapter 3

Higher-Order Statistics

There are three main reasons for using higher-order statistics in signal processing [18]:

1. To extract information due to deviations from Gaussianity (normality).
2. To estimate the phase of non-Gaussian parametric signals.
3. To detect and characterize the non-linear properties of mechanisms which generate time series via phase relations of their harmonic components.

The first two motivations presented above apply directly to this work on adaptive equalization. The first one is due to the fact that for Gaussian processes, all polyspectra of order higher than two are identically zero. In signal processing applications, any periodic or quasi-periodic signal can be characterized as non-Gaussian, while most additive noise processes are white Gaussian. Hence working with higher-order statistics has the advantage of not being affected by Gaussian noise.

The second motivation is based on the fact that higher-order spectra preserve the phase information of non-Gaussian signals. In the previous chapter we saw equalization methods based on least-squares criteria. They are widely used because they yield linear equations that are easy to solve. But the autocorrelation domain (second-order moment) suppresses the phase
information, and hence those approaches cannot cope with systems that are non-minimum phase.

**Definitions**

**Moments and cumulants of stochastic signals**

Let's define a set of $n$ random variables $\{x_1, x_2, x_3, \ldots, x_n\}$. Their joint moments of order $r = k_1 + k_2 + \ldots + k_n$ are given by\(^{[19]}\):

$$Mom[x_1^{k_1}, x_2^{k_2}, \ldots, x_n^{k_n}] \equiv E[x_1^{k_1} x_2^{k_2} \ldots x_n^{k_n}]$$

$$= (-j)^r \frac{\partial^r \Phi(\omega_1, \omega_2, \ldots, \omega_n)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2} \ldots \partial \omega_n^{k_n}} \bigg|_{\omega_1 = \omega_2 = \ldots = \omega_n = 0}$$ (39)

where

$$\Phi(\omega_1, \omega_2, \ldots, \omega_n) \equiv E[e^{j(\omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n)}]$$ (40)

is their joint characteristic function.

Another form of the joint characteristic function is defined as the natural logarithm of $\Phi(\omega_1, \omega_2, \ldots, \omega_n)$:

$$\Psi(\omega_1, \omega_2, \ldots, \omega_n) \equiv \ln[\Phi(\omega_1, \omega_2, \ldots, \omega_n)]$$ (41)

The joint cumulants of order $r$ of the same set of random variables are defined as the coefficients in the Taylor expansion of the second characteristic function about zero:
\[ \text{Cum}_x \times \text{h}_x \times \text{K}^{-n-j}' \text{dr}(\text{co}_1, \text{co}_2, \ldots, \text{co}_n) \]

\[ M_{\text{dco}^n}^n \]

\[ \text{fl} \times 1 = \text{fl} \times 2 = \ldots = C_{\text{On} = 0} \]

Thus the joint cumulants can be expressed in terms of the joint moments of a set of random variables. The relationships can be obtained by substituting:

\[ \langle \text{fl} \rangle_1 = l + j \alpha m_1 - m_2 + m_k + \ldots \]

(43)

into (39), (41), (42), and working out differentiations about zero. For example, the moments of the random variable \{x_i\}:

\[ m_1 = \text{Mom}_x = \sum x \]

\[ m_2 = \text{Mom}[x_i x_i] = \sum x^2 \]

\[ m_3 = \text{Mom}[x_i x_i x_i] = \sum x^3 \]

\[ m_4 = \text{Mom}[x_i x_i x_i x_i] = \sum x^4 \]

are related to its cumulants by:

\[ c_1 = \text{Cum}[x_i] = m_1 \]

\[ c_2 = \text{Cum}[x_i x_i] = m_2 - m_1 \]

\[ c_3 = \text{Cum}[x_i x_i x_i] = m_3 - 3m_2 + 2m_1 \]

\[ c_4 = \text{Cum}[x_i x_i x_i x_i] = m_4 - 4m_3 + 6m_2 - 6m_1 + m_1 \]

The general relationship between moments of \( x^1, x^2, x^3, \ldots, x^n \) and joint cumulants of order \( r = n \) is given by:

\[ f_{17}^{23} \]
where the summation extends over all partitions \((S_1, S_2, \ldots, S_p)\) such that 

\[ P = 1, 2, \ldots, n, \]

of the set of integers \((1, 2, \ldots, n)\). For example, the set of integers \((1, 2, 3)\) can be partitioned into:

- \(P = 1\): \(S_1 = \{1, 2, 3\}\)
- \(P = 2\): \(S_1 = \{1\}, S_2 = \{2, 3\}\)
- \(P = 3\): \(S_1 = \{2\}, S_2 = \{1, 3\}\)
- \(P = 3\): \(S_1 = \{3\}, S_2 = \{1, 2\}\)

And hence the third order cumulant of the random variables \(x_1, x_2, x_3\) is:

\[
C_{x_1 x_2 x_3} = \mathbb{E}\{x_1 x_2 x_3\} - \mathbb{E}\{x_1\}\mathbb{E}\{x_2 x_3\} - \mathbb{E}\{x_2\}\mathbb{E}\{x_1 x_3\} - \mathbb{E}\{x_3\}\mathbb{E}\{x_1 x_2\} + 2\mathbb{E}\{x_1\}\mathbb{E}\{x_2\}\mathbb{E}\{x_3\}\]

Moments and cumulants of stochastic processes

Let \(\{X(k)\}\), where \(k = 0, 1, 2, \ldots\) be a stationary random process. If the moments of \(\{X(k)\}\) up to order \(n\) exist, then:

\[ \text{Mom}[X(k), X(k + r_1), \ldots, X(k + n - 1)] = \mathbb{E}\{X(k)X(k + r_1)\ldots X(k + n - 1)\} \]

The moments depend only on the time differences \(T_1, T_2, \ldots, T_{n - 1}\).
$T_j = 0, 1, 2, \ldots$ for all $i$. Hence, simplifying the notation, the moments of a stationary random process can be written as:

$$m^{(r_1 t_2 t_\ldots t_{n-1})} = E\{X(k)X(k+r_1)\ldots X(k+T_{n-1})\} \quad (46)$$

and the $n-1$ order cumulants:

$$C^{(T_1, T_2, \ldots, T_{n-1})} = \text{Cum}[X(k), X(k+T_1), \ldots, X(k+T_{n-1})] \quad (47)$$

Combining (44), (46) and (47) the following relationships between moments and cumulants of a stationary random process $\{X(k)\}$ are obtained:

First-order cumulants:

$$C^{X} = E\{X(k)\} = \mu$$

Mean value \quad (48)

Second-order cumulants:

Set of integers: $(0, 1)$

Partitions: $p=1$ \quad $S^1 = \{0, 1\}$

$p=2$ \quad $S^2 = \{0\}$ $S^2 = \{1\}$

$c^{(T_1)} = E\{X(k)X(k+T_1)\} - E\{X(k)\}E\{X(k+r_1)\} \quad (49)$

And in terms of moments:

$c^2(T_1) = m^2(T_1) - 2\mu^2$\quad (50)

$\text{Covariance sequence}$

$= m^2(-T_1) - [m_1] = c^2(T_1)$

where:

$m^0(T_i) = E\{X(k)X(k+T_i)\}$ is the autocorrelation sequence.
Third-order cumulants:

Set of integers: \( (0,1,2) \)

Partitions:
- \( p=1 \):
  \[ S_1 = \{0,1,2\} \]
- \( p=2 \):
  \[ S_1 = \{0\}, S_2 = \{1,2\} \]
  \[ S_1 = \{1\}, S_2 = \{0,2\} \]
  \[ S_1 = \{2\}, S_2 = \{0,1\} \]
- \( p=3 \):
  \[ S_1 = \{0\}, S_2 = \{1\}, S_3 = \{2\} \]

Fourth-order cumulants:

Set of integers: \( (0,1,2,3) \)

Partitions:
- \( p=1 \):
  \[ S_1 = \{0,1,2,3\} \]
- \( p=2 \):
  \[ S_1 = \{0,1\}, S_2 = \{2,3\} \]
  \[ S_1 = \{0,2\}, S_2 = \{1,3\} \]
  \[ S_1 = \{0,3\}, S_2 = \{1,2\} \]
  \[ S_1 = \{1\}, S_2 = \{0,2,3\} \]
  \[ S_1 = \{2\}, S_2 = \{0,1,3\} \]
  \[ S_1 = \{3\}, S_2 = \{0,1,2\} \]

\[ CX(T_i, T_2) = E\{X(k)X(k+T_i)X(k+T_2)\} - E\{X(k)\}E\{X(k+T_i)X(k+T_2)\} - E\{X(k+T_i)\}E\{X(k)X(k+T_2)\} - E\{X(k+T_2)\}E\{X(k)X(k+T_i)\} + 2E\{X(k)\}E\{X(k+T_i)\}E\{X(k+T_2)\} \] (51)

So in terms of moments:

\[ c_3(\rho_i, T_2) = mx(T_i, T_2) - mx mx(T_2 - T_1) - mx mx(T_2) - mx mx(T_i) + 2(mx)^2 \] (52)
\[ c_4^x(\tau_1, \tau_2, \tau_3) = E\{X(k)X(k + \tau_1)X(k + \tau_2)X(k + \tau_3)\} \]

\[-E\{X(k)X(k + \tau_1)\}E\{X(k + \tau_2)X(k + \tau_3)\} \]

\[-E\{X(k)X(k + \tau_2)\}E\{X(k + \tau_1)X(k + \tau_3)\} \]

\[-E\{X(k)X(k + \tau_3)\}E\{X(k + \tau_1)X(k + \tau_2)X(k + \tau_3)\} \]

\[-E\{X(k + \tau_1)\}E\{X(k)X(k + \tau_2)X(k + \tau_3)\} \]

\[-E\{X(k + \tau_2)\}E\{X(k)X(k + \tau_1)X(k + \tau_3)\} \]

\[-E\{X(k + \tau_3)\}E\{X(k)X(k + \tau_1)X(k + \tau_2)\} \]

\[+2E\{X(k)X(k + \tau_1)\}E\{X(k + \tau_2)\}E\{X(k + \tau_3)\} \]

\[+2E\{X(k)X(k + \tau_2)\}E\{X(k + \tau_1)\}E\{X(k + \tau_3)\} \]

\[+2E\{X(k)X(k + \tau_3)\}E\{X(k + \tau_1)\}E\{X(k + \tau_2)\} \]

\[+2E\{X(k + \tau_1)X(k + \tau_2)\}E\{X(k)\}E\{X(k + \tau_3)\} \]

\[+2E\{X(k + \tau_1)X(k + \tau_3)\}E\{X(k)\}E\{X(k + \tau_2)\} \]

\[+2E\{X(k + \tau_2)X(k + \tau_3)\}E\{X(k)\}E\{X(k + \tau_1)\} \]

\[-6E\{X(k)\}E\{X(k + \tau_1)\}E\{X(k + \tau_2)\}E\{X(k + \tau_3)\} \] (53)

In terms of moments:
\[ c_4^x(\tau_1, \tau_2, \tau_3) = m_4^x(\tau_1, \tau_2, \tau_3) - m_2^x(\tau_1)m_2^x(\tau_3 - \tau_2) \]

\[-m_2^x(\tau_2)m_2^x(\tau_3 - \tau_1) - m_2^x(\tau_3)m_2^x(\tau_2 - \tau_1) \]

\[-m_1^x m_3^x(\tau_2 - \tau_1, \tau_3 - \tau_1) - m_1^x m_3^x(\tau_2, \tau_3) \]
The kurtosis is defined as the fourth-order cumulant of a zero-mean stochastic process (i.e., $m_T = E\{x(k)\} = 0$) for which $T_1 = T_2 = T_3 = 0$:

$$yx = \mathcal{C}_1(0,0,0) = E\{x^4(k)\} - 3E\{x^2(k)\}^2$$

If $T_1 = T_2 = T_3 = 1$ and $T_2 = 0$, the fourth-order cumulant is:

$$c\mathcal{C}_1(1,1,1) = E\{x(k)X^3(k) + 1\} - 3E\{x(k)X(k) + 1\}E\{x^2(k) + 1\}$$

Properties of moments and cumulants:

1. Moments and cumulants are symmetric functions in their arguments:
   $$\mathrm{Mom}\{x_1, x_2, x_3\} = \mathrm{Mom}\{x_2, x_1, x_3\} = \mathrm{Mom}\{x_3, x_2, x_1\}$$

2. Moments and cumulants are functions of the constants $a_1, a_2, ..., a_n$.
3. If the random variables \( \{x_1, x_2, \ldots, x_n\} \) can be divided into two or more groups which are statistically independent, their \( n \)th-order cumulant is identical to zero; i.e., \( \text{Cum}[x_1, x_2, \ldots, x_n] = 0 \) whereas, in general, \( \text{Mom}[x_1, x_2, \ldots, x_n] \neq 0 \).

4. If the sets of random variables \( \{x_1, x_2, \ldots, x_n\} \) and \( \{y_1, y_2, \ldots, y_n\} \) are independent, then:
\[
\text{Cum}[x_1+y_1, x_2+y_2, \ldots, x_n+y_n] = \text{Cum}[x_1, \ldots, x_n] + \text{Cum}[y_1, \ldots, y_n]
\]

whereas in general:
\[
\text{Mom}[x_1+y_1, x_2+y_2, \ldots, x_n+y_n] = E\{(x_1+y_1)(x_2+y_2)\ldots(x_n+y_n)\}
\]
\[
\neq \text{Mom}[x_1, \ldots, x_n] + \text{Mom}[y_1, \ldots, y_n]
\]

However, for the random variables \( \{y_1, x_1, x_2, \ldots, x_n\} \) we have that:
\[
\text{Cum}[x_1+y_1, x_2, \ldots, x_n] = \text{Cum}[x_1, x_2, \ldots, x_n] + \text{Cum}[y_1, x_2, \ldots, x_n]
\]
and
\[
\text{Mom}[x_1+y_1, x_2, \ldots, x_n] = \text{Mom}[x_1, x_2, \ldots, x_n] + \text{Mom}[y_1, x_2, \ldots, x_n]
\]

5. If the set of random variables \( \{x_1, x_2, \ldots, x_n\} \) is jointly Gaussian, then all the information about their distribution is contained in the moments of order \( n \leq 2 \). Therefore, all moments of order greater than two \( (n > 2) \) have no new information to provide. This leads to the fact that all joint cumulants of order \( n > 2 \) are identical to zero for Gaussian random vectors. So the cumulants of order \( n > 2 \) in a sense measure the non-Gaussianity of a time series.
Equalization with higher-order statistics

The purpose of blind equalization is to identify the inverse of an unknown linear time-invariant (possibly non-minimum phase) system without any physical access to the system input signal. Such operation requires the identification of both the magnitude and the phase of the system's transfer function. Identification of the magnitude can be accomplished with second-order statistics alone, but finding the phase involves the higher-order statistics of the received signal. In this sense, in order to find the phase of the system's transfer function, some form of nonlinearity must be used. Depending on where the nonlinear transformation is being applied on the data, three important families of blind equalization algorithms have appeared [17]:

1. The Bussgang algorithms, where the nonlinearity is in the output of the adaptive equalization filter.

2. The Polyspectra algorithms, where the nonlinearity is in the input of the adaptive equalizer filter, and

3. The algorithms where the nonlinearity is inside the equalization filter, i.e., nonlinear filter (e.g. Volterra) or neural network.

The Bussgang algorithms are generally implemented with LMS-based approaches. Of the three families mentioned above, the Bussgang algorithms have by far the lowest computational complexity, which is only slightly greater than that of a conventional adaptive equalizer equipped with a training phase. Hence, for the rest of this work I will concentrate only on Bussgang-type approaches.
The Bussgang deconvolution techniques

These algorithms are iterative deconvolution schemes that utilize memoryless nonlinear transformation at the output of the equalizer to generate a "desired" signal (an estimate of the input signal) at each iteration. Fig. 10 shows a block diagram of a blind deconvolution scheme:

\[ y(n) \rightarrow \text{Transversal filter } W(z) \rightarrow z(n) \rightarrow \text{Zero-memory nonlinear estimator } g(-) \rightarrow \text{Desired signal } \hat{x}(k) \]

Figure 11. Blind equalizer.

The key component in this scheme is the zero-memory nonlinear estimator, which allows us to estimate the input data sequence \( x(n) \) given the deconvolved sequence \( z(n) \). The mean-square error (MSE) may be used to determine the best estimate of \( x(n) \) given \( z(n) \). The choice of this optimization criterion yields a conditional mean estimator that is both sensible and robust [12].

Given the observation \( z(n) \), the conditional mean estimate \( \hat{x}(n) \) of the random variable (input signal) \( x(n) \) is written as \( E\{x|z\} \) (dropping the time indexes for convenience of presentation), where \( E\{ \} \) denotes the expectation operation. Hence:
\[ \bar{x} = E\{ x | z \} = \int_{-\infty}^{\infty} x p_x(x | z) dx \]  

(57)

where:

\( p_x(x | z) \) is the conditional probability density function of \( x \) given \( z \)

(a posteriori density)

Using Bayes' rule on the conditional pdf, (57) becomes:

\[ \bar{x} = \int_{-\infty}^{\infty} x \frac{p_z(z|x)p_x(x)}{p_z(z)} dx \]  

(58)

\[ \bar{x} = \frac{1}{p_z(z)} \int_{-\infty}^{\infty} x p_z(z|x)p_x(x) dx \]  

(59)

where:

\( p_z(z|x) \sim N(x(n), \sigma_N^2) \), i.e., \( p_z(z|x) \) is normally distributed with mean \( x(n) \) (the input signal), and variance \( \sigma_N^2 \), the variance of the additive Gaussian noise \( N(n) \).

\( p_x(x) \) is the probability density function of the input signal, \( x(n) \).

If \( x(n) \) is zero-mean Gaussian, with variance \( \sigma_x^2 \), i.e. \( p_x(x) \sim N(0, \sigma_x^2) \),

then (59) reduces to [17]:

\[ \bar{x}(n) = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_N^2} z(n) \]  

(60)

In general, convergence of the Bussgang algorithm is not guaranteed. A proof of its convergence for the case of an infinite length equalizer has been
provided by Beneviste et al.\textsuperscript{[3]}, but unfortunately this infinite length equalizer assumption is unrealistic and unattainable in practice. To date, no zero-memory nonlinear function has been found which guarantees global convergence of the blind equalizer, and hence, the problem remains open.

**Special cases of the Bussgang algorithm**

**The Sato algorithm:**

The Sato algorithm\textsuperscript{[23]} consists of minimizing a non-convex cost function of the form:

\[ J(n) = E\left\{ \left[ \tilde{x}(n) - z(n) \right]^2 \right\} \]  \hspace{2cm} (61)

where (refer to Fig. 11):

- \( z(n) \): Output of the transversal filter.
- \( \tilde{x}(n) \): Estimate of the transmitted input signal.

The estimate of the transmitted input signal is obtained using the following zero-memory nonlinearity:

\[ \tilde{x}(n) = \gamma \text{sgn}[z(n)] \]  \hspace{2cm} (62)

where \( \text{sgn}[\_] \) refers to the signum function, and \( \gamma \) is a constant which sets the gain of the equalizer, and is defined by:

\[ \gamma = \frac{E\{x^2(n)\}}{E\{x(n)\}} \]  \hspace{2cm} (63)

So the gain of the equalizer is obtained from the statistics of the input signal \( x(n) \). The Sato algorithm for blind equalization was initially introduced to deal with the one-dimensional M-ary PAM signals. It is a robust algorithm, and it is superior to the decision-directed equalizer, although its rate of convergence is slower. In fact, the nonlinearity defined in (62) is very similar to that in the
decision-directed algorithm (35), except for the gain factor $\gamma$, which is
dependent on the input data.

Beneviste et al.\textsuperscript{[3]} have proved that the Sato algorithm can achieve
global convergence if the probability density function of the transmitted data
sequence can be approximated by a sub-Gaussian function such as the
uniform distribution \textsuperscript{[12]}. However, this result has been disputed by other
researchers, reporting poor performance of Sato's algorithm.

Just as in the case of the decision-directed equalizer, it has been
established that almost always Sato's algorithm converges to the correct
solution once the eye pattern has been opened.

**The Godard algorithm:**

The Godard algorithm \textsuperscript{[9]} consists of minimizing a nonconvex cost
function of the form:

$$J(n) = E\left\{ \left[ |z(n)|^p - R_p \right]^2 \right\}$$

where $p$ is a positive integer, and $R_p$ is a positive real constant defined as:

$$R_p = \frac{E\left\{ |x(n)|^{2p} \right\}}{E\left\{ |x(n)|^p \right\}}$$

The Godard algorithm is designed to penalize deviations of the blind
equalizer output $z(n)$ from a constant modulus \textsuperscript{[12]}. Godard was the first to
propose a family of *constant modulus* blind equalizers for use in two-
dimensional digital communication systems. The constant $R_p$ is chosen such
that the cost function in (64) is zero when perfect equalization is achieved.
The equalizer tap weights \{w_n\} in Godard's algorithm are updated using an LMS-type algorithm \cite{9}:

\[
W(k + 1) = W(k) + \mu e^*(k) Y^T(k) \tag{66}
\]

where:

\(\mu\) := Step size parameter.

\(e^*(k)\) := Conjugate of the error signal.

\(Y^T(k)\) := Input vector to the equalizer.

The error signal \(e(k)\) is generated as follows:

\[
e(k) = z(k)|z(k)|^{p-2}\left(R_p - |z(k)|^p\right) \tag{67}
\]

It can be seen from the above definition of error signal (67) and from the definition of the cost function in (64) that this adaptation algorithm does not require carrier phase recovery. Consequently, it runs slower, although it presents the advantage of decoupling the problems of carrier phase recovery and ISI cancellation from each other \cite{12}.

Two particular choices of the integer \(p\) above yield important cases:

- **Case 1**: \(p = 1\)

  In this case the cost function in (64) becomes:

  \[
  J(n) = E\left[\left|z(n) - R_1\right|^2\right] \tag{68}
  \]

  where:

  \[
  R_1 = \frac{E\{|x(n)|^2\}}{E\{|x(n)|\}} \tag{69}
  \]

  which can be viewed as a modification of the Sato algorithm.

- **Case 2**: \(p = 2\)

  The cost function in (64) becomes:
\[ J(n) = E \left\{ \left| z(n) \right|^2 - R_2 \right\}^2 \]  

(70)

where:

\[ R_2 = \frac{E\left\{ |x(n)|^4 \right\}}{E\left\{ |x(n)|^2 \right\}} \]  

(71)

This case is referred to in the literature as the constant modulus algorithm (CMA) \(^{[26]}\), and is the most widely used in practice and the most widely investigated blind equalization algorithm.

Although Godard showed in his original paper that the algorithm would converge to the global minimum and achieve perfect equalization, provided it was initialized in a special manner, there are conflicting reports in the literature since then. Some have demonstrated that it is possible for the Godard algorithm to exhibit ill convergence due to the existence of local (i.e. false) minima.

**A simplified set of conditions for equalization**

A sufficient condition for equalization is that the probability distribution of the individual recovered symbols at the equalizer output be equal to the probability distribution of the individual transmitted symbols at the channel input. This condition led to the formulation of a general class of criteria that converge to the desired response under the assumption that the input distribution belongs to a certain family of continuous-type distributions \(^{[3]}\). Note, however, that in digital communications the input distributions are of discrete type.
A new simplified set of conditions for equalization has been derived by Shalvi and Weinstein showing that the necessary and sufficient conditions for equalization are that the second- and fourth-order moments of the individual input and output symbols be equal.

The objective is to set the taps \( \{ w_k \} \) of the equalizer so that the output sequence \( z(n) \) is identical to the input sequence \( x(n) \), up to a constant delay and possibly a constant phase shift.

The impulse response of the combined system \( S(z) \) (see Fig. 12) can be expressed as the convolution sum of the impulse responses of the channel \( H(z) \) and the equalizer \( W(z) \):

\[
S(z) = h_k * w_k = \sum_{l=-\infty}^{\infty} h_{k-l} w_l
\]

Equalization condition: We want to set the vector of equalizer weights \( W = [w_0, w_1, w_2, \ldots] \) so that the combined system \( S = [s_0, s_1, s_2, \ldots] \) is a vector having only one nonzero component in which the magnitude equals one:

\[
S = \begin{bmatrix} e^{j0} & 0 & 0 \end{bmatrix}^T
\]

where \( \theta \) is the phase shift, and the nonzero component is delayed an unknown number of samples.
In this section a criterion for equalization is developed assuming that the input sequence is independent and identically distributed (i.i.d.).

Start by expressing the input-output relationship as:

\[ z(k) = x(k) * s_k = \sum_{i=0}^{\infty} x(k-i) s_i \quad (74) \]

By squaring both sides of (74) and taking expected values an expression is obtained for the variance of the output sequence in terms of the variance of the input and the impulse response of the combined system:

\[ E[z^2(k)] = E[x^2(k)] s^2_k \quad (75) \]

And similarly, by raising both sides of (74) to the fourth power, taking expected values and using (75), an expression is obtained for the kurtosis of the output sequence in terms of the variance of the input and the impulse response of the combined system:

\[ K(z) = K(x)^s_k \quad (76) \]

Where \( K(\cdot) \) represents the kurtosis as defined in Eq. (55).

Eqs. (75) and (76) form the basis of the following theorem:

**Theorem:** If \( J[z^2(k)] \equiv J[x^2(k)] \), then

\[ a) |K(z)| \leq |K(x)| \]

\[ b) |i'_(z(k))| = |i'_(x(k))| \]

if and only if the impulse response of the combined system is of the form of (73).
Proof: Let $S = [s_0 s_1 s_2]$ be a vector of complex variables such that $Y_i - V_i$.

Hence if $X_i = I_i = 1$ then:

$$D = M^i$$

where equality holds if and only if $S$ has at most one nonzero component.

And so the proof follows immediately from recalling (75) and (76).

Hence, by the above theorem, a necessary and sufficient condition for equalization is that $E[z^{(k)}] = E[x^{(k)}]$ and $K(z) = K(x)$, which is much simpler than having to equalize all moments of the probability distributions.

It can be shown as well that if the input and output sequences are real-valued, or if the input sequence is complex-valued such that $E|x^{(k)}| = 0$ (e.g. when the real and imaginary parts of $x^{(k)}$ are statistically uncorrelated with the same variance), then the condition $|i(z)| = |i_{<x;k}|$ can be replaced by $E\{z^{(k)}\} = E\{x^{(k)}\}$.
This follows from the above theorem, since by equating the variances of the input and output sequences we are constraining the impulse response of the combined system to \( X = 1 \) (see Eq-75). And if that is the case, then the kurtosis of the output sequence will always be smaller than or equal to the kurtosis of the input sequence. Hence by maximizing \(|i^(_j)|\) in the adaptation algorithm, perfect equalization will be achieved.

The criterion function is then written as follows:

\[
J = |K(z)| = \text{sgn}[K(z)] K(z) \tag{79}
\]

Substituting (55) into (79) for the general case of a complex sequence:

\[
J = \text{sgn}[K(x)] E\{z(k)^2 - 2Ez(k)^2 - Ez^2(k)} \tag{80}
\]

If the average power is constrained (i.e., the vector of equalizer weights is normalized after each iteration), the term \(Elz(k)\) is constant, and therefore can be ignored.

The algorithm requires spectral prewhitening of the channel output. The output sequence can then be expressed as:

\[
z(k) = y(k) * w_k = w_i y(k-1) \tag{81}
\]

where \(y(k)\) is the output of the channel, after the prewhitening operation. And so, performing straightforward differentiation of the criterion function in (80) with respect to each equalizer weight \(w_i\) we obtain an explicit expression for the gradient.
\[
\frac{\partial J}{\partial w_l} = 4 \text{sgn}[K(x)] \left[ E\{ |z(k)|^2 z(k) y^* (k - l) \} - E\{ z^2(k) \} E\{ z^* (k) y^* (k - l) \} \right]
\]  

(82)

Substituting the expected value by its current realization, the adaptive algorithm is obtained:

\[
W(k + 1) = W(k) + \delta \cdot \text{sgn}[K(x)] \left[ |z(k)|^2 z(k) - \langle z^2(k) \rangle z^* (k) \right] Y^T(k)
\]

(83)

where:

\[
\delta := \text{Step size.}
\]
\[
\langle z^2(k) \rangle := \text{Estimate of the variance of the output signal } E\{ z^2(k) \}.
\]

The variance of the output \( z(k) \) can be estimated by empirical averaging:

\[
\langle z^2(k) \rangle = (1 - \delta_c) \langle z^2(k - 1) \rangle + \delta_c z^2(k)
\]

(84)

where \( \delta_c \) is the step size used for the estimation.

A normalization operation is required at every iteration in order to satisfy the average power constraint:

\[
W'(k) = \frac{W(k)}{\sqrt{\sum_l |w_l(k)|^2}}
\]

(85)

This algorithm can be applied to both the sub-Gaussian case (\( K(x) < 0 \)) as well as the super-Gaussian case (\( K(x) > 0 \)). Recall, however, that the input sequence has to be i.i.d., which limits the amount of problems it can be applied to. More importantly, it requires spectral prewhitening of the channel output, which in some applications may be prohibitive in terms of added complexity.
In this section, criteria for blind equalization will be presented free of the restrictions that were needed in the previous section. Start by choosing a cost function that is a function of the unit impulse response of the combined system.

\[
J(s) = \frac{1}{2} \int_{\mathbb{R}} |s(t)|^2 dt + \int_{\mathbb{R}} f(t) dt 
\]

where \( s(t) \) is a measurable piecewise continuous real-valued function such that the cost function \( J(s) \), having the form:

\[
g(t) = t^2 + f(t) 
\]

monotonically increases in the interval \( 0 < t < 1 \), and monotonically decreases for \( t > 1 \), having a unique maximum at \( t_1 \).

Claim: The cost function \( J(s) \) obtains its maximum if and only if the impulse response of the combined system \( S \) is of the form of Eq. (73) (i.e. has only one nonzero component in which the magnitude equals one).

Proof: Using the inequality in (77), we can write (86) as:

\[
J(s) \leq \frac{1}{2} \int_{\mathbb{R}} |s(t)|^2 dt + \int_{\mathbb{R}} f(t) dt 
\]

where equality holds if and only if \( S \) has at most one nonzero component. By the definition of \( g(t) \) in (87) above:

\[
g(s) = \sum_{i=1}^{n} s(i) 
\]

where equality holds if and only if \( s(i) \leq 1 \). Hence both (88) and (89) are satisfied with equality if and only if \( S \) is of the form of Eq. (73).
To specify a criterion function, consider the following choice for a function $g(t)$:

$$g(t) = 2\alpha t - \alpha t^2, \quad \alpha > 0 \quad (90)$$

This function increases monotonically in $0 \leq t < 1$, and decreases monotonically for $t > 1$, as required. The criterion function will then have the same form as $g(t)$:

$$J(s) = \left[ \sum_l |s_l|^4 - \left( \sum_l |s_l|^2 \right)^2 \right] + 2\alpha \sum_l |s_l|^2 - \alpha \left( \sum_l |s_l|^2 \right)^2 \quad (91)$$

This term becomes zero when the impulse response of the combined system, $S$, is of the form of Eq. (73).

Using (75) and (76), the criterion function above (91) can be expressed in terms of the statistics of the input $x(k)$ and the output $z(k)$. Hence:

$$J = \frac{K(z)}{K(x)} - (1 + \alpha) \frac{E^2 \left\{ |z(k)|^2 \right\}}{E^2 \left\{ |x(k)|^2 \right\}} + 2\alpha \frac{E \left\{ |z(k)|^2 \right\}}{E \left\{ |x(k)|^2 \right\}} \quad (92)$$

Which can be rewritten in the form:

$$J = \text{sgn}[K(x)] \left[ E \left\{ |z(k)|^4 \right\} - E \left\{ |z(k)|^2 \right\} \right]^2$$

$$+ \gamma_1 E^2 \left\{ |z(k)|^2 \right\} + 2 \gamma_2 E \left\{ |z(k)|^2 \right\} \quad (93)$$
where:
\[
71 = -2 + (l+a)K(x)
\]
\[E2l\]
\[
E_2\]
\[w/2\]
\]
\[(94)\]

and,
\[
72 = \& K(x)E^{k}[x(k)]
\]
\]
\[(95)\]

Now substituting (81) into (93) (notice that no prewhitening of the observed signal \(y(k)\) is needed this time), and performing straightforward differentiation with respect to the equalizer weights, we obtain an expression for the gradient:
\[
dJ = 4\text{sgn}[K(x)E^z(k)]y^*(k-l) + y_2E\{z(k)y^*(k-l)}
\]
\[(96)\]

Empirical averaging can be used to estimate the values of \(E_1(z(k))\) and \(E_2(z(k))\). Approximating the rest of the expectation by their current realizations we obtain the following adaptive algorithm:
\[
W(k+1) = W(k) + 8\text{sgn}[K(x)]|z(\ddagger)|^2 + r_1(|z(\ddagger)|^2) + r_2(\ddagger - \ddagger^2W)YT(k)
\]
\[(97)\]
where:

\[
\langle z^2(k) \rangle = (1 - \delta_\varepsilon)\langle z^2(k - 1) \rangle + \delta_\varepsilon z^2(k)
\]  \hspace{1cm} (98)

and,

\[
\langle |z(k)|^2 \rangle = (1 - \delta_\rho)\langle |z(k - 1)|^2 \rangle + \delta_\rho |z(k)|^2
\]  \hspace{1cm} (99)

are the empirical estimates of the expected values mentioned before, and \( \delta_\varepsilon \) and \( \delta_\rho \) are the step sizes used.

In the next chapter, a new algorithm for blind equalization is developed, using as the starting point the same unconstrained criteria just presented here. The cost function used in this new approach is simpler than the one in Eq. (91), resulting in a faster and simpler algorithm.
Chapter 4

Blind equalization

In the previous chapters, simplified criteria were presented for blind channel equalization. The development of this new algorithm follows along similar lines to that of the unconstrained criteria previously presented. We will start by defining all the variables involved in the problem. Referring to Fig. 12 in the previous chapter, we have the following vectors:

The equalizer tap weights (L taps):

\[ \mathbf{w} = [w_1, w_2, \ldots, w_L] \]  

(100)

The impulse response of the combined system (length M):

\[ \mathbf{s} = [s_0, s_1, s_2, \ldots, s_M] \]  

(101)

And the input sequence to the equalizer at time \( k \) (L samples):

\[ \mathbf{y}(k) = [y(k), y(k-1), y(k-2), \ldots, y(k-L)] \]  

(102)

The equalization condition will basically be the same as the one given in Eq. (73), although a slight simplification is introduced. We still want to set the equalizer weights so that the combined system impulse response in (101) is a vector having only one nonzero component in which the magnitude equals one:

46
\[ S = e^{j\theta} [1 \ 0 \ 0 \ \ldots \ 0]^T \]  \hspace{1cm} (103)

where \( \theta \) is the phase shift, and the possibility of a delay in the nonzero component is not taken into account for simplicity’s sake.

The **criterion function** (or cost function) will be chosen following the general model of the function \( g(t) = t^2 + f(t) \) (see Eq. 87). Consider, then, the following simple function:

\[ g(t) = -\alpha t^2 \ ; \alpha > 0 \]  \hspace{1cm} (104)

which is an inverted parabola with a maximum at \( t = 0 \). The parameter \( \alpha \) can be viewed as sort of a step size parameter, since it determines the speed of the approach to the perfect equalization point \( (t = 0) \):

**Figure 13.** Mesh surface of the model function g(t).
Since the equalization condition is $E\{z^2(k)\} = E\{x^2(k)\}$ and $K(z) = K(x)$, these two equations will be included into the cost function, following the form of the model function $g(t)$. Hence the proposed cost function is:

$$J = \left[ K(z) - K(x) \right] - \alpha \left[ E\{z^2(k)\} - E\{x^2(k)\} \right]^2$$ \hspace{1cm} (105)

This term becomes zero when the impulse response of the combined system, $S$, is of the form of Eq. (103). Note that this term does not appear in the general model function $g(t)$ (Eq. 104), but it is included so that the kurtosis equalization criteria above forms part of the cost function. This follows the same approach used by Shalvi & Weinstein in going from Eq. (90) to Eq. (91).

Next, we want to show that the cost function in (105) has no spurious local maxima. If that is the case, then a gradient-search algorithm will be expected to converge to the desired response. Eq. (105) can be written in terms of the impulse response of the combined system by using Eqs. (75) and (76):

$$J = \left[ K(x) \sum_l |s_l|^4 - K(x) \right] - \alpha \left[ E\{x^2(k)\} \sum_l |s_l|^2 - E\{x^2(k)\} \right]^2$$ \hspace{1cm} (106)

The impulse response of the combined system, $S$, can be expressed as the convolution sum of the impulse responses of the channel and the equalizer (see Eq. 72). The first term of the impulse response of the combined system, $s_0$, is:

$$s_0 = h_0 w_0$$ \hspace{1cm} (107)
Thus, if the first tap of the equalizer is kept equal to one during the equalization process, and assuming $h_0 = 1$ (if it is not so, we can make it so by means of an automatic gain control, AGC), we have:

$$s_0 = 1 \quad (108)$$

Using (108) in (106):

$$J = K(x) J_{\text{sys}} - a E^2 \{x^2(k)\}$$

$$J = K(x) J_{\text{sys}} - a E^2 \{x^2(k)\} Z^*_{0}$$

$$J = K(x) J_{\text{sys}} - a E^2 \{x^2(k)\} Z^*_{0} X_k z_l$$

$$J = K(x) J_{\text{sys}} - a E^2 \{x^2(k)\} Z^*_{0} X_k z_l$$

Using the inequality in (77), we obtain:

$$n^2 = K(x) J_{\text{sys}} X_k z_l$$

The above inequality achieves equality if and only if $S$ has at most one nonzero component. Thus, we can see that the cost function has a global maximum. Let us now see if it has any other local maxima as well. Taking the derivative of (110) with respect to each coefficient $S_j$ of the combined system impulse response, the gradient of the cost function is:

$$\frac{\partial J}{\partial S_i} = 4 K(x) s_i - 2 a E^2 \{x^2(k)\} Z^*_{0}$$

And equating it to zero to find the maxima:

$$4 K(x) s_i = 2 a E^2 \{x^2(k)\} Z^*_{0}$$

Hence, the two solutions to Eq. (113) are:
Solution 1:

Since we already know that \( s_0 = 1 \) (Eq. 108), this solution corresponds to the perfect equalization criterion (Eq. 103).

Notice that Eq. (116) only yields a valid solution if \( K(x) > 0 \), that is, when the input distribution is super-Gaussian. If we make that assumption, we can continue by writing expressions for all \( M-1 \) components of \( S_j \):

\[
0 = s_1^2 - 2 \xi_{\{x_j\}}(117)
\]

So, in order to obtain a solution to (116), we have to solve a system of \( 2^{M-1} \) equations.

Adding all \( s \), we obtain:

\[
\sum_{i=0}^{M-1} s_i = 2^{M-1} \xi_{\{x_j\}}(118)
\]

The two summation terms cancel out, and we can solve for \( a \):

\[
a = t_{pr,0} - T(119)
\]
In practice, this is not a valid solution. For all practical purposes, it is impossible to choose for $\alpha$ precisely the value in (119). Thus, even if the input distribution is super-Gaussian, as long as the value for $\alpha$ is not the one in Eq. (119), the algorithm will converge to the desired global maximum. Hence the proposed cost function $J$ in Eq. (105) is a suitable one for blind equalization.

Next I will derive an LMS-type, gradient-search algorithm using the cost function in (105). The adaptive algorithm will have the general form:

$$W(k + 1) = W(k) - \delta \frac{\partial J}{\partial W} \quad (120)$$

Substituting (55) into (105), and expanding the square of the difference of expected values:

$$J = \left[ E\left\{ z^4(k) \right\} - 3E^2\left\{ z^2(k) \right\} - K(x) \right]$$
$$- \alpha \left[ E^2\left\{ z^2(k) \right\} + E^2\left\{ x^2(k) \right\} - 2E\left\{ z^2(k) \right\} E\left\{ x^2(k) \right\} \right] \quad (121)$$

And now write the output sequence $z(k)$ in terms of the input vector to the equalizer $Y$ and the vector of equalizer weights $W$, both at time $k$:

$$z(k) = W^T(k) \cdot Y(k) = Y^T(k) \cdot W(k) \quad (122)$$

The variance of the input sequence $x(k)$ will be denoted by:

$$\sigma_x^2(k) = E\left\{ x^2(k) \right\} \quad (123)$$

For simplicity's sake, the time index $k$ will be dropped from (107) and (108).
Substituting (122) and (123) into (121), the cost function can be written entirely in terms of the statistics of the input sequence \( x(k) \), the received sequence \( y(k) \), and the equalizer coefficients. Hence:

\[
J = \left[ E\left\{W^T Y Y^T W W^T Y Y^T W\right\} - 3E^2\left\{W^T Y Y^T W\right\} - K(x) \right] \\
- \alpha \left[ E^2\left\{W^T Y Y^T W\right\} + \left(\sigma_x^2\right)^2 - 2\sigma_x^2 E\left\{W^T Y Y^T W\right\} \right]
\]  

(124)

In order to find the maximum of the cost function, its derivative has to be taken with respect to the vector of equalizer coefficients. Recall from vector calculus:

\[
\frac{\partial}{\partial W} \left(W^T Y Y^T W\right) = 2 Y Y^T W
\]

(125)

Thus, using the chain rule of differentiation, and assuming that the order of expectation and differentiation can be interchanged, the derivative of the cost function is:

\[
\frac{\partial J}{\partial W} = E\left\{[2 Y Y^T W][W^T Y Y^T W] + [2 Y Y^T W][W^T Y Y^T W]\right\} \\
- (3)(2)E\left\{W^T Y Y^T W\right\}E\left\{2 Y Y^T W\right\} \\
- \alpha(2)E\left\{W^T Y Y^T W\right\}E\left\{2 Y Y^T W\right\} + \alpha(2)\sigma_x^2 E\left\{2 Y Y^T W\right\}
\]

(126)

The statistics of the input (variance and kurtosis) are obviously not functions of the equalizer weights, and so as constants they drop out due to the differentiation. Pulling all the constants out of the expectations, and substituting (122) back into (126) yields:
\[
\frac{\partial J}{\partial W} = 4E\left\{z^3(k)\mathbf{Y}(k)\right\} - 12E\left\{z^2(k)\right\}E\left\{z(k)\mathbf{Y}(k)\right\} \\
- 4\alpha E\left\{z^2(k)\right\}E\left\{z(k)\mathbf{Y}(k)\right\} + 4\alpha \sigma_x^2 E\left\{z(k)\mathbf{Y}(k)\right\} 
\]  
(127)

If we denote the variance of the output sequence \(z(k)\) by:
\[
\sigma_z^2(k) = E\left\{z^2(k)\right\} 
\]  
(128)

Then, substituting (128) into (127) and grouping like terms:

\[
\frac{\partial J}{\partial W} = 4E\left\{z^3(k)\mathbf{Y}(k)\right\} - \left(12 + 4\alpha\right)\sigma_z^2(k)E\left\{z(k)\mathbf{Y}(k)\right\} \\
+ 4\alpha \sigma_x^2 E\left\{z(k)\mathbf{Y}(k)\right\} 
\]  
(129)

In order to use the above gradient (129) in a gradient-search algorithm, an estimate of the expected values has to be obtained. The simplest type of estimation will be used: replacing the expectations by their current realizations. The gradient can then be approximated by:

\[
\frac{\partial J}{\partial W} \approx 4\left[z^2(k) - \sigma_z^2(k)(3 + \alpha) + \alpha \sigma_x^2\right]z(k)\mathbf{Y}(k) 
\]  
(130)

And hence, the new blind equalization algorithm is:

\[
\mathbf{W}(k+1) = \mathbf{W}(k) - 4\delta\left[z^2(k) - \sigma_z^2(k)(3 + \alpha) + \alpha \sigma_x^2\right]z(k)\mathbf{Y}^T(k) 
\]  
(131)

where:

\(\mathbf{W}(k)\): = Vector of equalizer tap weights at iteration \(k\).
$Y(k) := \text{Observed sequence vector (input to the equalizer) at time } k.$

$\delta := \text{Algorithm step size.}$

$z(k) := \text{Output of the equalizer at time } k.$

$\alpha := \text{Cost function step size parameter.}$

$\sigma_x^2 := \text{Variance of the input distribution.}$

$\sigma_z^2(k) := \text{Estimate of the variance of the output distribution at iteration } k.$

The variance of the output distribution $z(k)$ will be estimated by performing empirical averaging:

$$\sigma_z^2(k + 1) = (1 - \delta_{\epsilon})\sigma_z^2(k) + \delta_{\epsilon}z^2(k) \quad (132)$$

where $\delta_{\epsilon}$ is the estimation step size.
Chapter  5

Simulations and results

In order to test the performance of the blind equalization algorithm derived in Chapter 4 (Eq. 131), MATLAB simulations were carried out comparing its performance to the LMS algorithm (Eq. 34) and the Shalvi and Weinstein algorithm (Eq. 97). Plots will be provided here comparing the convergence behavior and the probability of symbol error of the algorithms when equalizing several different channels.

It is important to point out that the results shown here are neither absolute nor definitive. Blind equalization algorithms are extremely sensitive to the choice of step size and other key parameters. Due to the nonlinear nature of these algorithms, choosing a wrong value for one of these parameters often means unbound growth of the equalizer coefficients, in other words, the algorithm becomes unstable. But in trying to avoid this, choosing a step size that is too small will result in very slow convergence, and many thousands of samples may be needed in order to reach a steady-state solution. The parameter values used in these simulations are not optimal, but rather empirical values based on the outcomes of many trials.
All the simulations were run using real-valued signals, although they work just the same for complex-valued signals. Computational time and complexity was an issue while running the simulations. Because of that, the number of samples used for equalization was kept at a minimum, as were the number of equalizer taps. As can be seen in the simulations, sometimes the weights have not fully converged yet, and that results in a larger probability of symbol error. For those cases, equalization was performed until the tap weights unequivocally showed that they were converging. Using longer equalization periods and more taps results in total convergence (perfect equalization), and lower error probabilities.

Monte-Carlo simulations were also performed to obtain some measure of the average performance of the algorithms at hand. For a given channel, the equalization process (for a range of signal-to-noise ratios) was repeated 50 times, and the probabilities of symbol error averaged over the 50 trials. In all, six different channels were simulated:

1. One-pole channel (simple RC network channel model).
   Algorithms: Chapter 4 blind equalizer, Shalvi & Weinstein blind equalizer, and decision-directed LMS.

2. One-pole channel with colored Gaussian noise added.
   Algorithms: Ch. 4 blind equalizer, decision-directed LMS.

3. One-pole time-varying channel.
   Algorithms: Ch. 4 blind equalizer, decision-directed LMS.

4. Two-pole channel.
   Algorithms: Ch. 4 blind equalizer, Shalvi & Weinstein blind equalizer, and LMS with training.
5. Three-pole channel. Algorithms: Ch. 4 blind equalizer and LMS with training.

6. Three-pole, two-zero channel. Algorithms: Ch. 4 blind equalizer and LMS with training.

Case 1: One-pole channel A simple low-pass filter is used as the model for the channel (Fig. 14). The input sequence is generated via a binary source outputting uniformly distributed +1's and -1's. White Gaussian noise $N(k)$ is added at the output of the channel. In the discrete-time domain, the output of the channel is given by:

$$y(k+1) = x(k+1) + e^{-aTb}y(k) + N(k)$$  (133)

where:

- $a$, $b$ are the 3 dB bandwidth.
- $RC/T_0$: Bit duration (signaling period).
- The product $aT_0$ is a measure of the intersymbol interference (ISI). The greater this product is, the smaller the ISI. For $aT_0 > 2$ the ISI is nonexistent or negligible. In this case I chose $aT_0 = 1$ (see Fig. 14d).

Substituting this value of $aT_0$ into (133), we obtain:

$$y(k+1) = x(k+1) + e^{-X}y(k) + N(k)$$  (134)

And taking the z-transform of both sides of (134), the transfer function is:

$$H(z) = r$$  (135)

$$1 - 0.3679z^{-1}$$
The ideal equalizer, if the additive Gaussian noise was to be ignored, would have a transfer function $W(z) = \ldots$

The error probability that $H(z)$ results from using this ideal equalizer will be compared to that of the other three equalizers.

Finally, the step sizes used for this particular channel are:

- Blind equalizer: $\delta = 0.0005$.
- Shalvi & Weinstein blind equalizer: $\delta_{sw} = 0.05$.
- Decision-directed LMS equalizer: $\mu = 0.05$.

Input sequence: $x(n)$

Output sequence: $y(n)$

**Block diagram.**

**Figure 14. One-pole (RC network) channel model.**
**Figure 15.** Convergence behavior of the blind equalizer for a one-pole channel (SNR = 10 dB).

<table>
<thead>
<tr>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.3552</td>
<td>0.0005</td>
<td>-0.0194</td>
<td>0.0469</td>
<td>-0.0059</td>
</tr>
</tbody>
</table>
Figure 16. Convergence behavior of the Shalvi & Weinstein blind equalizer for a one-pole channel (SNR = 10 dB).

<table>
<thead>
<tr>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2038</td>
<td>-0.0710</td>
<td>-0.0012</td>
<td>-0.0078</td>
<td>0.0084</td>
<td>-0.0016</td>
</tr>
</tbody>
</table>
Figure 17. Convergence behavior of the decision-directed LMS equalizer for a one-pole channel (SNR = 10 dB).

**Final decision-directed LMS equalizer weights**

<table>
<thead>
<tr>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9339</td>
<td>-0.3586</td>
<td>0.0139</td>
<td>-0.0365</td>
<td>0.0976</td>
<td>-0.0824</td>
</tr>
</tbody>
</table>
Figure 18. Probability of error for equalization of a one-pole channel. (Accuracy: ± 0.0003)
**Case 2: One-pole channel with additive colored noise**

This channel is the very same simple RC model used in the previous case. However, the additive white Gaussian noise in this case will be passed through a simple filter. The resulting colored Gaussian noise is then added to the output of the channel, and equalized.

The noise filter \( A(z) \) in Fig. 19 is chosen such that the variance of the input noise signal \( N_w(k) \) is the same as the variance of the output noise signal \( N_c(k) \). The colored noise signal can be written in terms of the white noise signal as:

\[
N_c(k) = a_1 N_w(k) + a_2 N_w(k - 1) \tag{136}
\]

From the above, the relationship between the input and output variances is:

\[
\sigma_{N_c}^2 = a_1^2 \sigma_{N_w}^2 + a_2^2 \sigma_{N_w}^2 = \left( a_1^2 + a_2^2 \right) \sigma_{N_w}^2 \tag{137}
\]

And hence, if \( \sigma_{N_c}^2 = \sigma_{N_w}^2 \), then the filter coefficients have to be:

\[
a_1^2 + a_2^2 = 1 \tag{138}
\]

That is,

\[
a_1 = a_2 = 0.7071 \tag{139}
\]

**Figure 19.** Channel with additive colored noise block diagram.
Figure 20. Convergence behavior of the blind equalizer for a one-pole channel with colored noise added (SNR = 20 dB)

<table>
<thead>
<tr>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.3479</td>
<td>-0.0045</td>
<td>0.0206</td>
<td>-0.0128</td>
<td>-0.0148</td>
</tr>
</tbody>
</table>
**Figure 21.** Convergence behavior of an LMS equalizer with training for a one-pole channel with colored noise added (SNR = 20 dB).

<table>
<thead>
<tr>
<th></th>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.9991</td>
<td>-0.3709</td>
<td>-0.0030</td>
<td>-0.0067</td>
<td>-0.0016</td>
<td>-0.0084</td>
</tr>
</tbody>
</table>
**Case 3: One-pole time-varying channel**

The simple one-pole channel used in case 1 will be modified here so that the pole changes as the equalization process is taking place. To that effect, the parameter $aT_b$ in Eq. (133) will vary as follows (see Fig. 22):

$$aT_b(k) = \begin{cases} 
-\frac{0.2}{1000}k + 1 & \text{for } 0 \leq k < 1000 \\
0.8 & \text{for } k \geq 1000 
\end{cases}$$

(140)

![Graph of $aT_b(k)$](image)

**Figure 22.** Time-varying channel.

The purpose of simulating this time-varying case is to see if the blind equalizer developed in Chapter 4 can track variations in time of the channel characteristics. So for this case, no probability of error calculations were performed, and only the second tap weight of the equalizer, $w_1$, is plotted. Recall from Eqs. (134) and (135) that changing the value of $aT_b$ results in the second tap weight of the channel changing as well.

Two blind algorithms are compared here, the blind equalizer developed in Ch. 4 and the decision-directed LMS equalizer. Plotted on the same graph with the second tap weights for the equalizers, is the actual variation of the channel pole.
Figure 23. Convergence behavior of the second tap weight of the blind equalizer and the decision-directed LMS equalizer for a one-pole time-varying channel (SNR = 10 dB).

(Note: The straight line segment is the ideal equalizer pole variation. The trace for the decision-directed LMS algorithm is the one that hovers above the ideal pole, while the trace for the blind equalizer is, on the average, much closer to the ideal pole.)

<table>
<thead>
<tr>
<th>Final blind equalizer weights</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_0 )</td>
<td>1</td>
<td>(-0.4140)</td>
<td>0.0283</td>
<td>(-0.0160)</td>
<td>(-0.0257)</td>
<td>0.0236</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final decision-directed LMS equalizer weights</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_0 )</td>
<td>0.9029</td>
<td>(-0.3691)</td>
<td>0.0353</td>
<td>(-0.0193)</td>
<td>(-0.0213)</td>
<td>0.0244</td>
</tr>
</tbody>
</table>
**Figure 24.** Convergence behavior of the second tap weight of the blind equalizer and the decision-directed LMS equalizer for a one-pole time-varying channel (SNR = 20 dB).

*(Note: The straight line segment is the ideal equalizer pole variation. The trace for the decision-directed LMS algorithm is the more constant one, while the trace for the blind equalizer shows more variation during convergence.)*

<table>
<thead>
<tr>
<th>Final blind equalizer weights</th>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>-0.4552</td>
<td>-0.0151</td>
<td>-0.0036</td>
<td>-0.0098</td>
<td>-0.0089</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final decision-directed LMS equalizer weights</th>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.9029</td>
<td>-0.4483</td>
<td>-0.0170</td>
<td>-0.0005</td>
<td>-0.0113</td>
<td>-0.0072</td>
</tr>
</tbody>
</table>
Case 4: Two-pole channel

The transfer function of the channel is:

\[ H(z) = \frac{1}{1 - 1.2z^{-1} + 0.32z^{-2}} = \frac{1}{(1 - 0.8z^{-1})(1 - 0.4z^{-1})} \]  

The input to the channel is an uniformly distributed 8-PAM (8-symbol Pulse Amplitude Modulation) sequence, with the PAM levels shown in Fig. 25:

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.7</td>
<td>-0.5</td>
<td>-0.3</td>
<td>-0.1</td>
<td>+0.1</td>
<td>+0.3</td>
<td>+0.5</td>
<td>+0.7</td>
</tr>
</tbody>
</table>

**Figure 25.** Pulse Amplitude Modulation (PAM) levels.

Three algorithms were simulated, with the following parameters:

- Blind equalizer: \( \vartheta = 0.005 \); \( \alpha = 0.1 \)
- Shalvi & Weinstein blind equalizer: \( \vartheta = 0.005 \); \( \alpha = 0.01 \); \( w_0 = 1 \)
- LMS equalizer with training: \( \mu = 0.005 \)

For the Shalvi & Weinstein blind equalizer, the first tap weight had to be kept equal to one throughout the equalization process, in order to ensure convergence of the rest of the weights.

Since the input distribution is known, the variance and the kurtosis of the 8-PAM signal can be calculated. Hence the values used for the blind equalizers are:

- Variance of the input signal: \( \sigma_x^2 = 0.21 \)
- Kurtosis of the input signal: \( K(x) = -0.0546 \)
**Figure 26.** Convergence behavior of the blind equalizer for a two-pole channel with additive white Gaussian noise. (SNR = 25 dB).

<table>
<thead>
<tr>
<th>Final blind equalizer weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0$</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
Figure 27. Convergence behavior of the Shalvi & Weinstein blind equalizer for a two-pole channel with additive white Gaussian noise (SNR = 25 dB).

<table>
<thead>
<tr>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.0826</td>
<td>0.1099</td>
<td>0.1098</td>
<td>-0.0027</td>
<td>-0.0295</td>
</tr>
</tbody>
</table>
Figure 28. Convergence behavior of the LMS equalizer with training for a two-pole channel with additive white Gaussian noise. (SNR = 25 dB)

Final LMS equalizer with training weights

<table>
<thead>
<tr>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9568</td>
<td>-1.1032</td>
<td>0.2300</td>
<td>0.0565</td>
<td>-0.0178</td>
<td>-0.0036</td>
</tr>
</tbody>
</table>
Figure 29. Error density function of the blind equalizer for a two-pole channel with additive white Gaussian noise.
(SNR = 25 dB)
Figure 30. Error density function of the Shalvi & Weinstein blind equalizer for a two-pole channel with additive white Gaussian noise (SNR = 25 dB).
Figure 31. Error density function of the LMS equalizer with training for a two-pole channel with additive white Gaussian noise (SNR = 25 dB)
Figure 32. Error function of the ideal (channel inverse) equalizer, for a two-pole channel with additive white Gaussian noise. (SNR = 25 dB)
Figure 33. Probability of error for equalization of a two-pole channel with additive white Gaussian noise.
(Accuracy = ± 0.0001)
Case 5: Three-pole channel

The transfer function of the channel is:

\[
H(z) = \frac{1}{1 - 0.7z^{-1} - 0.3z^{-2} + 0.16z^{-3}}
\]

\[
= \frac{1}{(1 - 0.8z^{-1})(1 - 0.4z^{-1})(1 + 0.5z^{-1})}
\]  \hspace{1cm} (142)

The input sequence is the same 8-PAM distribution as in the previous case. Two algorithms were run, with the following parameters:

- Blind equalizer: \( \vartheta = 0.0015 \); alpha = 0.1
- LMS equalizer with training: \( \mu = 0.01 \)
Figure 34. Convergence behavior of the blind equalizer for a three-pole channel with additive white Gaussian noise. (SNR = 15 dB)

<table>
<thead>
<tr>
<th>( w_0 )</th>
<th>( w_1 )</th>
<th>( w_2 )</th>
<th>( w_3 )</th>
<th>( w_4 )</th>
<th>( w_5 )</th>
<th>( w_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.6746</td>
<td>-0.2963</td>
<td>0.1242</td>
<td>0.0225</td>
<td>0.0195</td>
<td>-0.0078</td>
</tr>
</tbody>
</table>
Figure 35. Convergence behavior of the LMS equalizer with training for a three-pole channel with additive white Gaussian noise (SNR = 15 dB).

Final LMS equalizer with training weights

<table>
<thead>
<tr>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>$w_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9810</td>
<td>-0.6790</td>
<td>-0.2981</td>
<td>0.1463</td>
<td>0.0031</td>
<td>-0.0018</td>
<td>-0.0011</td>
</tr>
</tbody>
</table>
Figure 36. Probability of error for equalization of a three-pole channel with additive white Gaussian noise.
(Accuracy = ± 0.0001)
Case 6: Three-pole, two-zero channel

The transfer function of the channel is:

\[ H(z) = \frac{1 + 0.85z^{-1} - 0.095z^{-2}}{1 - 0.7z^{-1} - 0.3z^{-2} + 0.16z^{-3}} \]

\[ = \frac{(1 - 0.95z^{-1})(1 + 0.1z^{-1})}{(1 - 0.8z^{-1})(1 - 0.4z^{-1})(1 + 0.5z^{-1})} \]

(143)

And the transfer function of the ideal filter \( W^*(z) = \frac{1}{H(z)} \) is:

\[ W^*(z) = 1 - 1.55z^{-1} + 1.1125z^{-2} - 0.9329z^{-3} + 0.8987z^{-4} - 0.6753z^{-5} + \ldots \]

(144)

The input sequence is an 8-PAM distribution, as in the previous two cases. Two algorithms were run, with the following parameters:

- Blind equalizer: \( \vartheta = 0.0008 \); \( \text{alpha} = 0.1 \)
- LMS equalizer with training: \( \mu = 0.00075 \)
Figure 37. Convergence behavior of the blind equalizer for a three-pole, two-zero channel with additive white Gaussian noise. (SNR = 25 dB)

<table>
<thead>
<tr>
<th>$w_0$</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>$w_6$</th>
<th>$w_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1.2861</td>
<td>0.4291</td>
<td>-0.0980</td>
<td>0.0647</td>
<td>-0.0587</td>
<td>0.0264</td>
<td>0.0031</td>
</tr>
</tbody>
</table>
Figure 38. Convergence behavior of the LMS equalizer with training for a three-pole, two-zero channel with additive white Gaussian noise (SNR = 25 dB).

Final LMS equalizer with training weights

<table>
<thead>
<tr>
<th>( w_0 )</th>
<th>( w_1 )</th>
<th>( w_2 )</th>
<th>( w_3 )</th>
<th>( w_4 )</th>
<th>( w_5 )</th>
<th>( w_6 )</th>
<th>( w_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7772</td>
<td>-1.0041</td>
<td>0.3971</td>
<td>-0.1493</td>
<td>0.1010</td>
<td>-0.0656</td>
<td>0.0377</td>
<td>-0.0145</td>
</tr>
</tbody>
</table>
Figure 39. Probability of error for equalization of a three-pole, two-zero channel, with additive white Gaussian noise. (Accuracy = ± 0.00004)
Conclusions

The simulations performed have clearly shown that the equalization algorithm developed in Chapter 4 (Eq. 131) works well for a variety of communications channels, and that it also compares favorably with other well-known equalization algorithms.

The basic advantage to the algorithm developed in this thesis is that it is actually "blind" to the input signals, and hence it does not require that a training sequence be supplied in order to converge to the right solution. The blind equalizer is entirely driven by statistics. The only information that needs to be supplied \textit{a priori} to the algorithm is the variance of the input signal which, of course, is always known.

For simple channels, such as the one-pole examples driven by binary input signals shown in Cases 1-3, all the equalization algorithms used performed very well. Even when the channel characteristics were varied in time, as in Case 3, the algorithms were able to track the variations and still converge to the right solution.

When more complicated channels were simulated, with an 8-PAM input sequence (Cases 4-6), some of the algorithms performed poorly. The Shalvi & Weinstein blind equalizer, which formed the basis for the development of the new blind equalizer, did not compare well with the new blind equalizer and the LMS equalizer with training. For the number of samples used in the equalization process, a significantly higher probability of symbol error was obtained. The decision-directed LMS equalizer was also tried, but it does not achieve convergence at all. The only equalizers that performed reasonably well
were the new blind equalizer (from Chapter 4) and the LMS equalizer with training.

Although it is not really appropriate to compare a blind equalizer with an equalizer that makes use of a training sequence, the fact that the blind equalizer developed here holds its own in the comparison is a measure of success. Of course the LMS with training does a little better for the same equalization period, but that is to be expected. While the LMS algorithm has complete knowledge of the signals to expect at the output, the blind equalizer only has knowledge of the variance of the input signal.

The drawbacks of the blind equalization algorithm developed, when compared to the LMS algorithm, are the increased computational complexity and the need for a longer equalization period. Compared to other blind equalizers in the literature, the one developed here is significantly less complex, and it also converges much faster. In fact, performance-wise, the blind equalizer developed here is much closer to the LMS algorithm than to other blind deconvolution schemes published so far.

Practical adaptive equalizers for telephone channels have typical filter sizes of 32 to 64 weights, and use tens of thousands of samples during the equalization process. In light of this, the fact that the blind equalizer needs more samples than the ones used for the simulations does not make it any less feasible in practice. When performing the simulations, there were severe limitations in both the number of filter taps and the length of the equalization process. Without these limitations, the simulations would have shown nearly perfect equalization, with symbol error rates significantly lower than those obtained.
Due to the nonlinear nature of the blind deconvolution problem, stability is a very important, and problematic, issue. With equalizers based on the MSE criteria (such as LMS), eigenvalues and poles can be used to study the stability of the algorithms. But in the blind equalization case, these methods are not available, and the problem becomes a very difficult one.

A possible area for future research is to study how Liapunov criteria may be useful in establishing stability criteria for the blind deconvolution problem. This would help solve the biggest problem encountered while simulating blind equalization algorithms: how to choose the step sizes such that the equalizer does not become unstable.

Another area for further study would be to investigate and simulate blind equalizers applied to nonminimum-phase channels. Although the theory introduced in this thesis establishes the feasibility of equalizing such channels, more time is needed to further pursue such possibility. Other channels, such as the digital magnetic recording channel, could also benefit from using blind equalizers.

Finally, a detailed noise analysis would be necessary, in order to prove that the blind equalizer developed here performs better than other equalizers, in a high additive white Gaussian noise environments.
References


Appendix
% The following MATLAB program simulates a random 8-PAM input sequence
% with additive white Gaussian noise. This sequence is passed through a
% channel modelled by a digital filter. The output of this channel is then
% equalized using three different algorithms: (1) The blind equalizer developed
% in Chapter 4 of this thesis; (2) The Shalvi & Weinstein blind equalizer;
% and (3) The LMS algorithm with training. Once the equalization process is
% completed, the final tap weights of the equalizer filters are used to filter out
% a different input sequence than the one used for equalization, but which has
% also been passed through the channel filter. After the output of the channel
% is filtered, the output of the equalizers is compared with the original input
% sequence, and the probability of symbol error is computed.
%
% The input signal is produced by a pulse amplitude modulation
% (PAM) source, in which the symbols are discrete random variables
% admitting the eight equiprobable values +/- 0.7, 0.5, 0.3 and 0.1
% The kurtosis of the input is known, since we know its probability
% distribution. \( \Gamma(x) = E[x(n)^4] - 3\{E[x(n)^2]\}^2 = 0.0777 - 3(0.21)^2 \)
% Hence \( \Gamma(x) = -0.0546 \)
%
% Define the 8 symbols used:
%
% \( S1 = -0.7; \)
% \( S2 = -0.5; \)
% \( S3 = -0.3; \)
% \( S4 = -0.1; \)
% \( S5 = 0.1; \)
% \( S6 = 0.3; \)
% \( S7 = 0.5; \)
% \( S8 = 0.7; \)
%
% Generate two different random input sequences, one used for equalization
% and the other one used for computation of the symbol error.
%
% \( tlength = 5000; \)
% \( ilength = 10000; \)
% \( errhos=zeros(50,16); \)
% \( errsw=zeros(50,16); \)
% \( errlms=zeros(50,16); \)
% \( errideal=zeros(50,16); \)

% Start the Monte-Carlo simulation (50 iterations)
%
for mc=1:1:50
%
% Generate the input symbols
%
    rand('seed',sum(100*clock));
t = rand(1, ilength);  % Equalization input sequence.
x = rand(1, ilength);  % Error computation sequence.
for i = 1:1:ilength
    if x(i) < 0.125
        x(i) = S1;
    elseif x(i) < 0.25
        x(i) = S2;
    elseif x(i) < 0.375
        x(i) = S3;
    elseif x(i) < 0.5
        x(i) = S4;
    elseif x(i) < 0.625
        x(i) = S5;
    elseif x(i) < 0.75
        x(i) = S6;
    elseif x(i) < 0.875
        x(i) = S7;
    else
        x(i) = S8;
    end
    if t(i) < 0.125
        t(i) = S1;
    elseif t(i) < 0.25
        t(i) = S2;
    elseif t(i) < 0.375
        t(i) = S3;
    elseif t(i) < 0.5
        t(i) = S4;
    elseif t(i) < 0.625
        t(i) = S5;
    elseif t(i) < 0.75
        t(i) = S6;
    elseif t(i) < 0.875
        t(i) = S7;
    else
        t(i) = S8;
    end
end
xt = t(1:tlength);
clear t;

% The input data will pass through a channel with frequency response
% H(Z)= 1/[(1-0.4*Z^(-1))*(1-0.8*Z^(-1))*(1+0.5*Z^(-1))]

b = [1];
a = [1 -0.7 -0.3 0.16];
st = filter (b, a, xt);
s = filter (b, a, x);

% Add Gaussian noise to the signal coming out of the channel
% The SNR will go from 15 to 30 dB.

E = 0.21;  % Variance of the input signal
% rand('normal');
randn('seed',sum(100*clock));
Nt = randn(1, tlength);
N = randn(1, ilength);
SNR = 15;

% Start SNR loop

for count = 1:1:16
    SNR
%
% Add the white Gaussian noise to the output of the channel, with the
% appropriate SNR.

    N0 = 10 ^ (-0.1 * SNR);
sigma = sqrt(N0 * E);
NS = Nt * sigma;
yt = st + NSt;
NS = N * sigma;
y = s + NS;
clear s NS st NSt
%
% Now the output of the channel will pass through the equalizer w
%
% Initialize the HOS algorithm variables
%
taps = 6;
% w = zeros (taps,taps+1);
w = zeros (taps,1);
% w(1,taps+1) = 1;
w(1) = 1;
delt = 0.005;
as = 1e-4;
varz = 0;
alpha = 0.1;

% Initialize the Shalvi & Weinstein HOS algorithm variables.
%
%wsw = zeros(taps,taps+1);
ww = zeros(taps,1);
% wsw(1,taps+1) = 1;
ww(1) = 1;
deltsw = 5e-3;
asw= 1e-3;
varzsw = 0;
alphasw = 0.01;
kurtosis = -0.0546;
gamm1 = (2 + ((1 + alphasw) * kurtosis) / (E)^2);
gamm2 = alphasw * (kurtosis) / (E)^2;
%
%
% Initialize the LMS algorithm variables
%
%wlms = zeros(taps,taps+1);
wlms = zeros(taps,1);
mu = 0.005;
%
% Run the HOS adaptive algorithm
%
for n = (taps + 1):1:tlength
    yc = yt(n:-1:n - taps + 1)';
    % z=w(:,n)'*yc;
    z=w'*yc;
    % w(:,n+1)=w(:,n)-4*delt*z*yc*(z^2+alpha*E-
    % varz*(3+alpha));
    w=w-4*delt*z*yc*(z^2+alpha*E-varz*(3+alpha));
    % w(1,n+1)=1;
    w(1)=1;
    if (isnan(w(2)))
        if (isnan(w(2,n+1)))
            error('The algorithm has blown up')
    end
    varz = (1-as) * varz + as * z^2;
%
% Run the Shalvi & Weinstein HOS algorithm
%
%zsw = wsw(:,n)' * yc;
zsww = wsw' * yc;
% wsw(:,n+1)=wsw(:,n)-deltsw*(zsw^2+(gamm1-1)
    *varzsw+gamm2)*zsw*yc;
wsw=wsw-deltsw*(zsw^2+(gamm1-1)
    *varzsw+gamm2)*zsw*yc;
% wsw(1,n+1)=1;
    wsw(1)=1;
    varzsw=(1-asw)*varzsw+asw*zsw^2;
%
% And the LMS algorithm with training,
%
% zlms = wlms(:,n) * yc;
  zlms = wlms' * yc;
  p = zlms - xt(n);
% wlms(:,n+1) = wlms(:,n) - 2 * mu * p * yc;
  wlms = wlms - 2 * mu * p * yc;
end
clear z zlms p yt;
%
% Take the last value of w and wlms as the channel's equalizer weights
%
%W(count,:) = w(:,tlength)'
W(count,:) = w';
%Wsw(count,:) = wsw(:,tlength)'
Wsw(count,:) = wsw';
%Wlms(count,:) = wlms(:,tlength)'
Wlms(count,:) = wlms';
if (mc ==1 & count == 1)
  wdig=w;
  wshw=wsw;
  wlmst=wlms;
end
clear w wsw wlms;
%
% Now pass the whole sequence y(n) -coming out of the channel-
% through the equalizers W, Wsw, Wlms and ideal (channel inverse).
%
  zout_hos = filter(W(count,:),b,y);
  zout_sw = filter(Wsw(count,:),b,y);
  zout_lms = filter(Wlms(count,:),b,y);
  zout_ideal = filter(a,b,y);
  clear y;
%
% Plot the HOS-equalizer weights
for l = 1:1:taps
  clg
  plot(w')
  grid
title('Daniel Diguele -Master's Thesis- HOS-based adaptive equalizer')
xlabel('Number of samples')
ylabel('Evolution of the equalizer''s weights')
pause
end

% Plot the Shalvi & Weinstein HOS-equalizer weights
for l = 1:1:taps
    clg
    plot(wsw')
    grid
title('Daniel Diguele -Master''s Thesis- S&W HOS-based adaptive equalizer')
xlabel('Number of samples')
ylabel('Evolution of the equalizer''s weights')
pause
end

% Plot the LMS-equalizer weights
for l = 1:1:taps
    clg
    plot(wlms(1,:))
    grid
    ylabel(['Evolution of the equalizer''s Wlms(',num2str(l),')'])
pause
end

% Quantize zout to PAM levels
for k = 1:1:length
    if zout_hos(k) <= -0.6
        q_hos(k) = S1;
    elseif zout_hos(k) <= -0.4
        q_hos(k) = S2;
    elseif zout_hos(k) <= -0.2
        q_hos(k) = S3;
    elseif zout_hos(k) < 0.0
        q_hos(k) = S4;
    elseif zout_hos(k) < 0.2
        q_hos(k) = S5;
    elseif zout_hos(k) < 0.4
        q_hos(k) = S6;
    elseif zout_hos(k) < 0.6
        q_hos(k) = S7;
    else
        q_hos(k) = S8;
end
if zout_sw(k) <= -0.6
    q_sw(k) = S1;
elseif zout_sw(k) <= -0.4
    q_sw(k) = S2;
elseif zout_sw(k) <= -0.2
    q_sw(k) = S3;
elseif zout_sw(k) < 0.0
    q_sw(k) = S4;
elseif zout_sw(k) < 0.2
    q_sw(k) = S5;
elseif zout_sw(k) < 0.4
    q_sw(k) = S6;
elseif zout_sw(k) < 0.6
    q_sw(k) = S7;
else
    q_sw(k) = S8;
end

if zout_lms(k) <= -0.6
    q_lms(k) = S1;
elseif zout_lms(k) <= -0.4
    q_lms(k) = S2;
elseif zout_lms(k) <= -0.2
    q_lms(k) = S3;
elseif zout_lms(k) < 0.0
    q_lms(k) = S4;
elseif zout_lms(k) < 0.2
    q_lms(k) = S5;
elseif zout_lms(k) < 0.4
    q_lms(k) = S6;
elseif zout_lms(k) < 0.6
    q_lms(k) = S7;
else
    q_lms(k) = S8;
end

if zoutIdeal(k) <= -0.6
    q_ideal(k) = S1;
elseif zoutIdeal(k) <= -0.4
    q_ideal(k) = S2;
elseif zoutIdeal(k) <= -0.2
    q_ideal(k) = S3;
elseif zoutIdeal(k) < 0.0
    q_ideal(k) = S4;
elseif zoutIdeal(k) < 0.2
    q_ideal(k) = S5;
\[ q_{\text{ideal}}(k) = S5; \]
\[ \text{elseif } z_{\text{out,ideal}}(k) < 0.4 \]
\[ q_{\text{ideal}}(k) = S6; \]
\[ \text{elseif } z_{\text{out,ideal}}(k) < 0.6 \]
\[ q_{\text{ideal}}(k) = S7; \]
\[ \text{else} \]
\[ q_{\text{ideal}}(k) = S8; \]
\[ \text{end} \]
\[ \text{end} \]
\[ \text{clear } z_{\text{out,hos}} z_{\text{out,sw}} z_{\text{out,lms}} z_{\text{out,ideal}}; \]

\% Find the number and quantity of the errors
\% 
\[ \text{ehos} = x - q_{\text{hos}}; \]
\[ \text{esw} = x - q_{\text{sw}}; \]
\[ \text{elms} = x - q_{\text{lms}}; \]
\[ \text{eideal} = x - q_{\text{ideal}}; \]
\[ \text{clear } q_{\text{hos}} q_{\text{sw}} q_{\text{lms}} q_{\text{ideal}}; \]
\[ \text{errhos}(mc,\text{count}) = \text{length(find(ehos))}/\text{ilength}; \]
\[ \text{errsw}(mc,\text{count}) = \text{length(find(esw))}/\text{ilength}; \]
\[ \text{errlms}(mc,\text{count}) = \text{length(find(elms))}/\text{ilength}; \]
\[ \text{errideal}(mc,\text{count}) = \text{length(find(eideal))}/\text{ilength}; \]
\[ \text{if } (mc == 1) \]
\[ \text{Wdig}=W; \]
\[ \text{Wshw}=Wsw; \]
\[ \text{Wlmst}=Wlms; \]
\[ \text{end} \]

\% Generate the probability density functions of the error signals
\% There are 15 possible error levels for the 8-PAM signal
\%
\[ \text{level} = -1.4; \]
\[ \text{phos} = \text{zeros}(1,15); \]
\[ \text{psw} = \text{zeros}(1,15); \]
\[ \text{plms} = \text{zeros}(1,15); \]
\[ \text{pideal} = \text{zeros}(1,15); \]
\[ \text{for } l = 1:1:15 \]
\[ \text{phos}(l) = \text{length(find((ehos<level+0.01)&(ehos>level-0.01))))/\text{ilength};} \]
\[ \text{psw}(l) = \text{length(find((esw<level+0.01)&(esw>level-0.01))))/\text{ilength};} \]
\[ \text{plms}(l) = \text{length(find((elms<level+0.01)&(elms>level-0.01))))/\text{ilength};} \]
\[ \text{pideal}(l) = \text{length(find((eideal<level+0.01)&(eideal>level-0.01))))/\text{ilength};} \]
\[ \text{level} = \text{level} + 0.2; \]
\[ \text{end} \]
clear ehos esw elms eideal;

% Plot the error pdf's
%
bar(-1.4:0.2:1.4,phos)
title('Daniel Diguele - Master's Thesis: HOS-based adaptive equalizer')
xlabel('Error')
ylabel('Probability of error')
    text(0.6,0.85,['PAM source, SNR= ',num2str(SNR),', dB'], 'sc')
    text(0.6,0.8,['Equalizer taps = ',num2str(taps)], 'sc')
    text(0.6,0.75,['Probability of error= %',num2str(e(count))], 'sc')
pause

clear phos plms psw pideal;

SNR = SNR + 1;
end
save 3pmonte errhos errsw errlms errideal Wdig Wshw Wblms delt
deltsw mu wdig wshw wlmst;
end